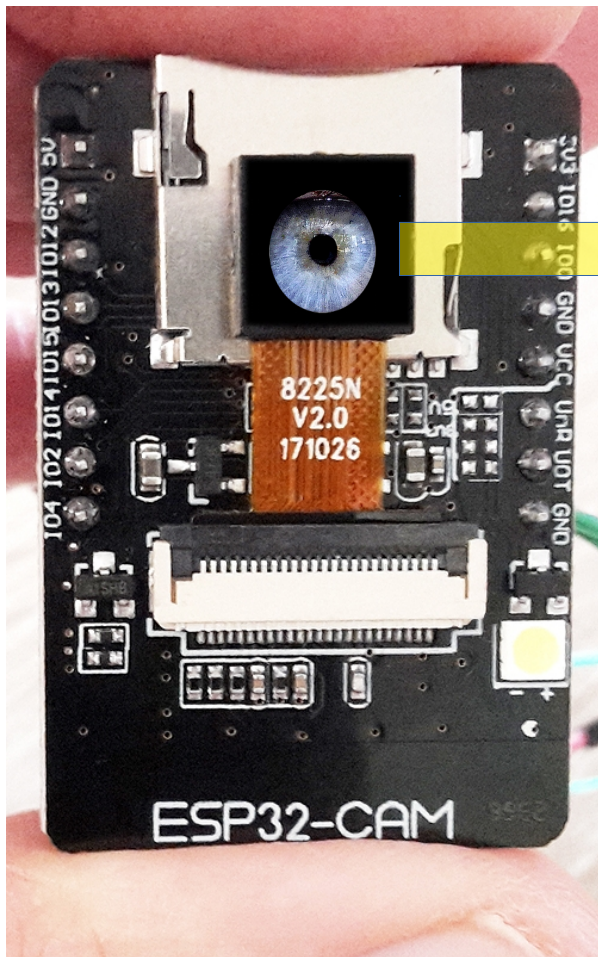


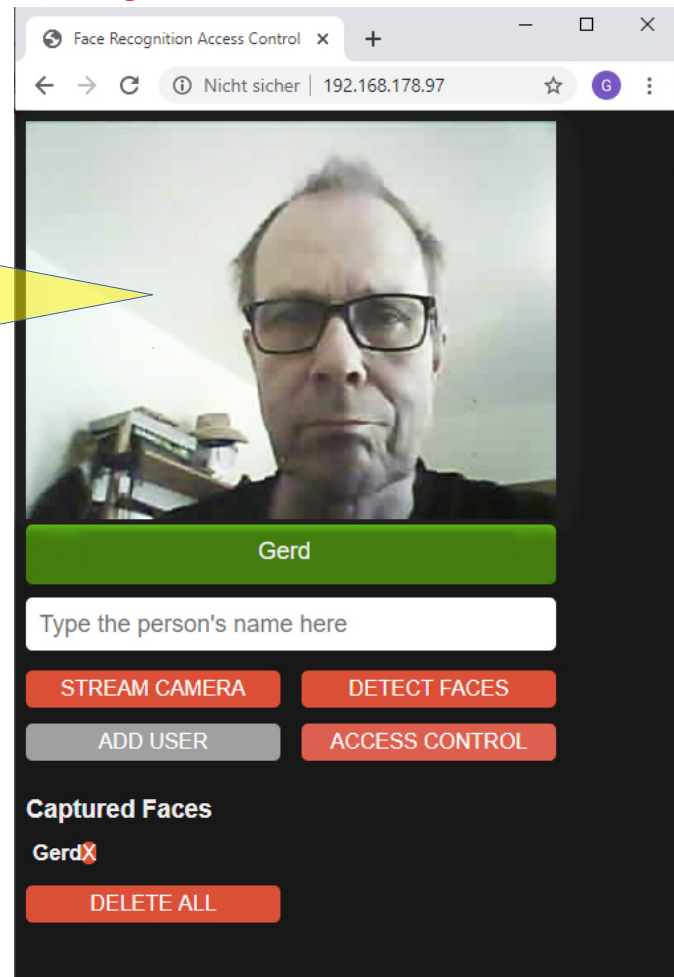
ESP32 mit Kamera: Von der Webcam für 6 € bis zum Gesichtserkennungs-Türöffner - bei Bedarf mit Sprachausgabe über Raspberry Pi



```
ESP32_open_door | Arduino 1.8.12
Datei Bearbeiten Sketch Werkzeuge Hilfe

ESP32_open_door camera_index.h camera_pins.h

#include <ArduinoWebsockets.h>
#include "esp_http_server.h"
#include "esp_timer.h"
#include "esp_camera.h"
#include "camera_index.h"
#include "Arduino.h"
#include "fd_forward.h"
#include "fr_forward.h"
#include "fr_flash.h"
const char* ssid = "hegburk"; const char* password = " ";
#define ENROLL_CONFIRM_TIMES 5
#define FACE_ID_SAVE_NUMBER 7
#define CAMERA_MODEL_AI_THINKER
#include "camera_pins.h"
using namespace websockets;
WebsocketsServer socket_server; camera_fb_t * fb = NULL;
long current_millis; long last_detected_millis = 0; long last_detected_millis1 = 0;
void app_facenet_main(); void app_httpserver_init();
```



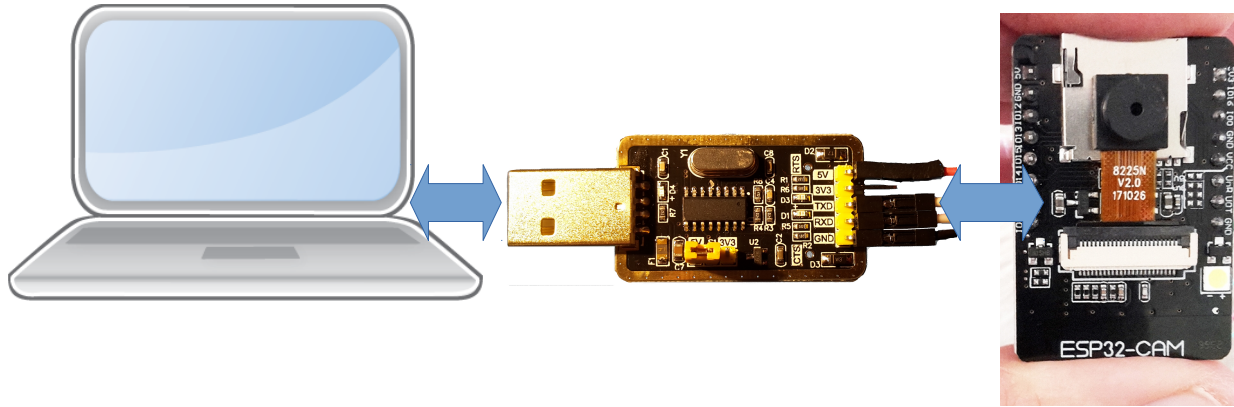
Das ESP32-Board mit Kamera kann als Webcam oder WLAN-Kamera nur zur Übertragung eines Videostreams benutzt werden – das Ganze für 6€. In den Baustein lässt sich eine Gesichtserkennung und -identifizierung integrieren, mit der Schaltimpulse für das Öffnen einer Tür abgegeben werden können.

Wird zusätzlich ein entsprechend programmierter Raspberry Pi angeschlossen, können identifizierte Personen sogar akustisch namentlich begrüßt werden.

Soll nur der Videostream bereitgestellt werden, kostet solch ein System knapp 6 € – vorausgesetzt eine 5-V-Stromversorgung existiert. Für die Schaltimpulse wären zusätzlich ca. 3 € für ein Relais notwendig und ein Raspberry Pi mit Zubehör zur Sprachausgabe kostet 30 bis 40 €.

Nur während der Programmierung wird ein USB-Seriell-Wandler benötigt, der etwa 5 € kostet, nach der Programmierung aber wieder abgebaut und anders verwendet werden kann.

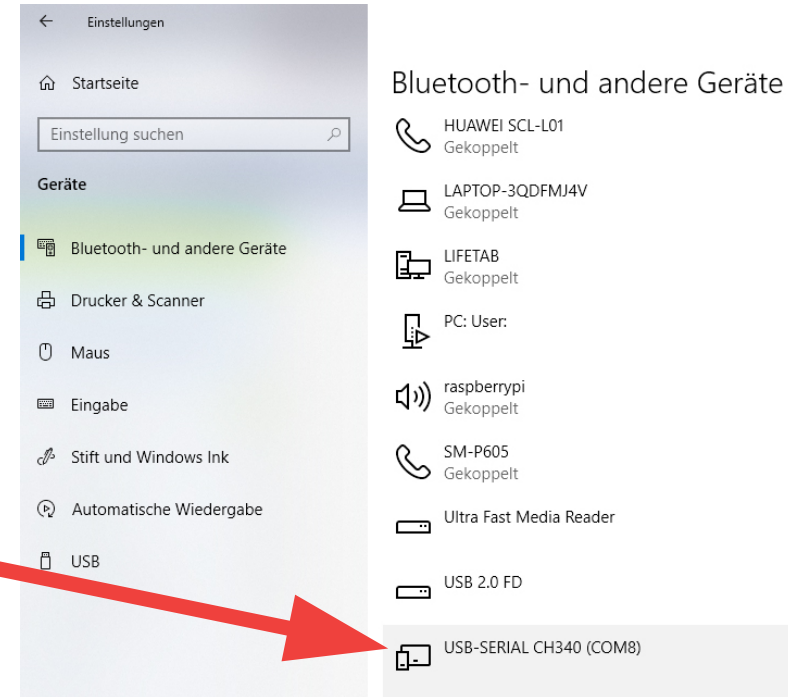
Bevor es richtig losgeht: Der USB-to-Serial-Adapter



Bevor man sich um die Anschlüsse am ESP32 selbst kümmert, sollte man sicherstellen, dass man einen funktionierenden US-Serial-Adapter hat und für diesen auch der Treiber installiert ist, denn er muss für die Programmierung zwischen Rechner und ESP32 eingebaut werden.

Überprüfung: Wenn man in die Einstellungen bei Windows 10 schaut, muss unter „Bluetooth- und andere Geräte“ ein Eintrag wie im Bild erscheinen.

Ggf. Notwendige Hilfestellungen findest du auf der folgenden Seite.



https://praxistipps.chip.de/usb-serial-converter-treiber-installieren-so-gehts_51213 – 23.05.2020

USB Serial Converter: Treiber installieren - so geht's

09.01.2017 16:43 von Tim Aschermann



Möchten Sie eine Verbindung zwischen USB- und RS232-Anschluss herstellen, müssen Sie zuvor den Treiber für den USB Serial Converter installieren. Wie das geht, zeigen wir Ihnen in diesem Praxistipp.

Anleitung: Treiber für USB Serial Converter installieren

Im besten Fall haben Sie den Treiber auf CD oder als EXE-Datei auf Ihrem Computer vorliegen. Starten Sie in diesem Fall einfach die CD oder die jeweilige Datei. Verbinden Sie den USB Serial Converter noch nicht mit dem Computer und folgen Sie den Anweisungen der Installation. Wurde die Installation abgeschlossen, können Sie den USB Serial Converter an den PC anschließen und nutzen.

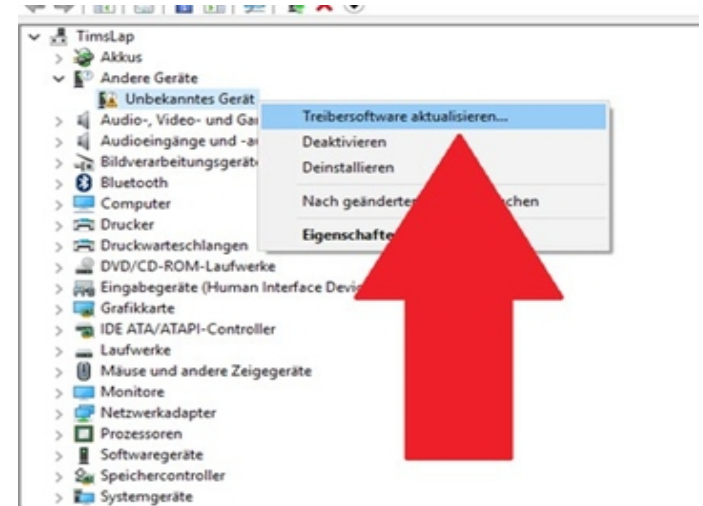
Anleitung: Treiber für USB Serial Converter installieren

Im besten Fall haben Sie den Treiber auf CD oder als EXE-Datei auf Ihrem Computer vorliegen. Starten Sie in diesem Fall einfach die CD oder die jeweilige Datei. Verbinden Sie den USB Serial Converter noch nicht mit dem Computer und folgen Sie den Anweisungen der Installation. Wurde die Installation abgeschlossen, können Sie den USB Serial Converter an den PC anschließen und nutzen.

USB Serial Converter: Treiber manuell installieren

Liegt kein Treiber vor, ist das halb so schlimm. In vielen Fällen kann Windows den Treiber auch alleine finden. Gehen Sie dazu wie folgt vor:

1. Verbinden Sie den USB Serial Adapter mit dem PC und öffnen Sie den .
2. Wählen Sie hier das "unbekannte Gerät" mit der rechten Maustaste aus.
3. Über die Option "Treibersoftware aktualisieren" kann Windows "Automatisch nach aktueller Treibersoftware suchen".
4. Wird Windows fündig, wird der Treiber heruntergeladen und anschließend installiert.
5. Klappt das nicht, klicken Sie erneut mit der rechten Maustaste auf das Gerät und wählen Sie die "Eigenschaften" aus.
6. Wechseln Sie oben zur Registerkarte "Details" und wählen Sie die Option "Hardware-IDs".
7. Mit einer der Hardware-IDs können Sie nun über Google nach Treibern für den USB Serial Converter suchen.



Damit sich das Board programmieren lässt:
Zusätzliche Bordverwalter-URL eingeben!

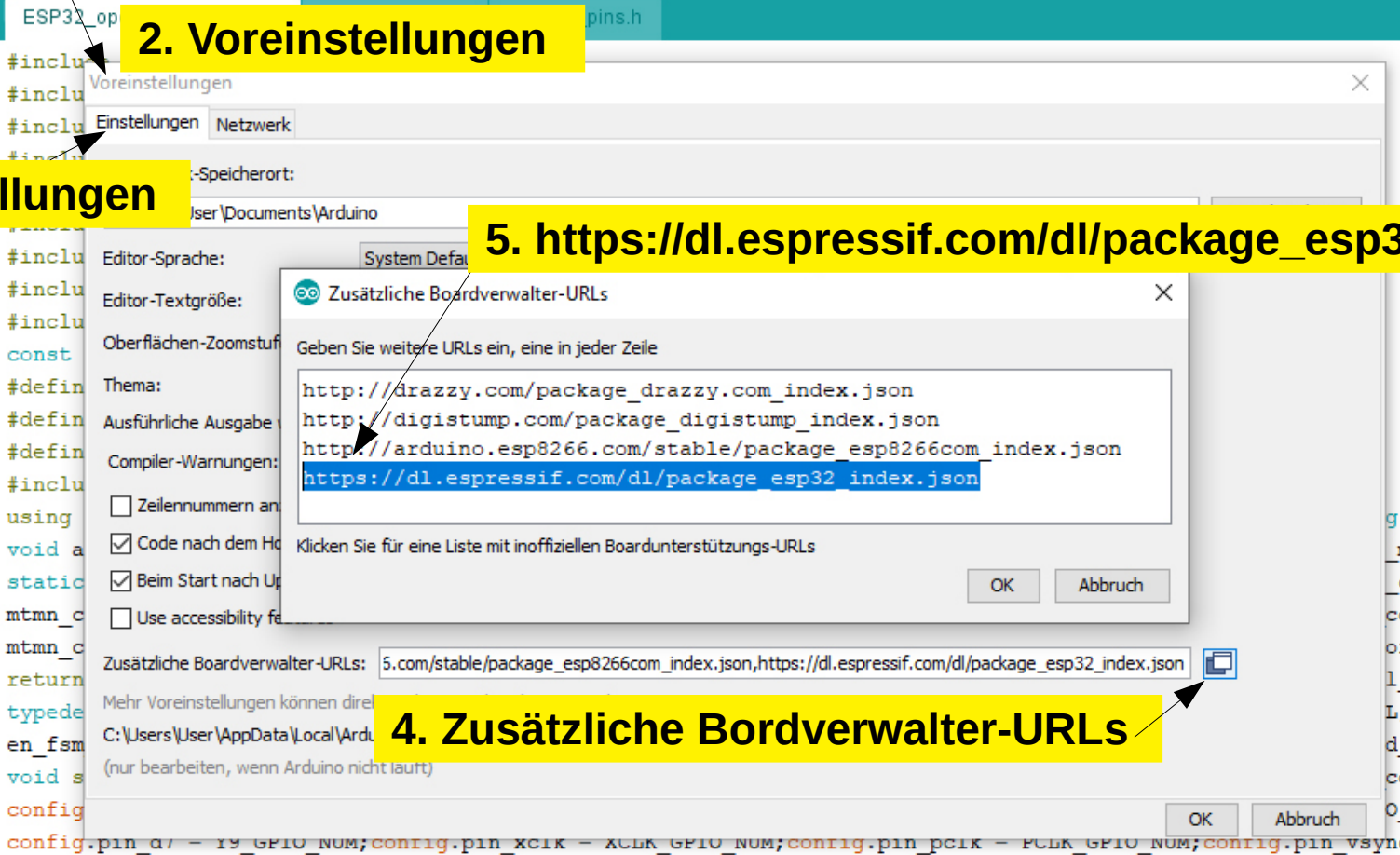
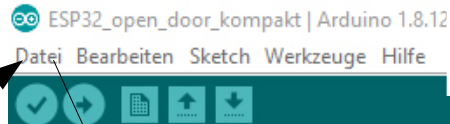
1. Datei

2. Voreinstellungen

3. Einstellungen

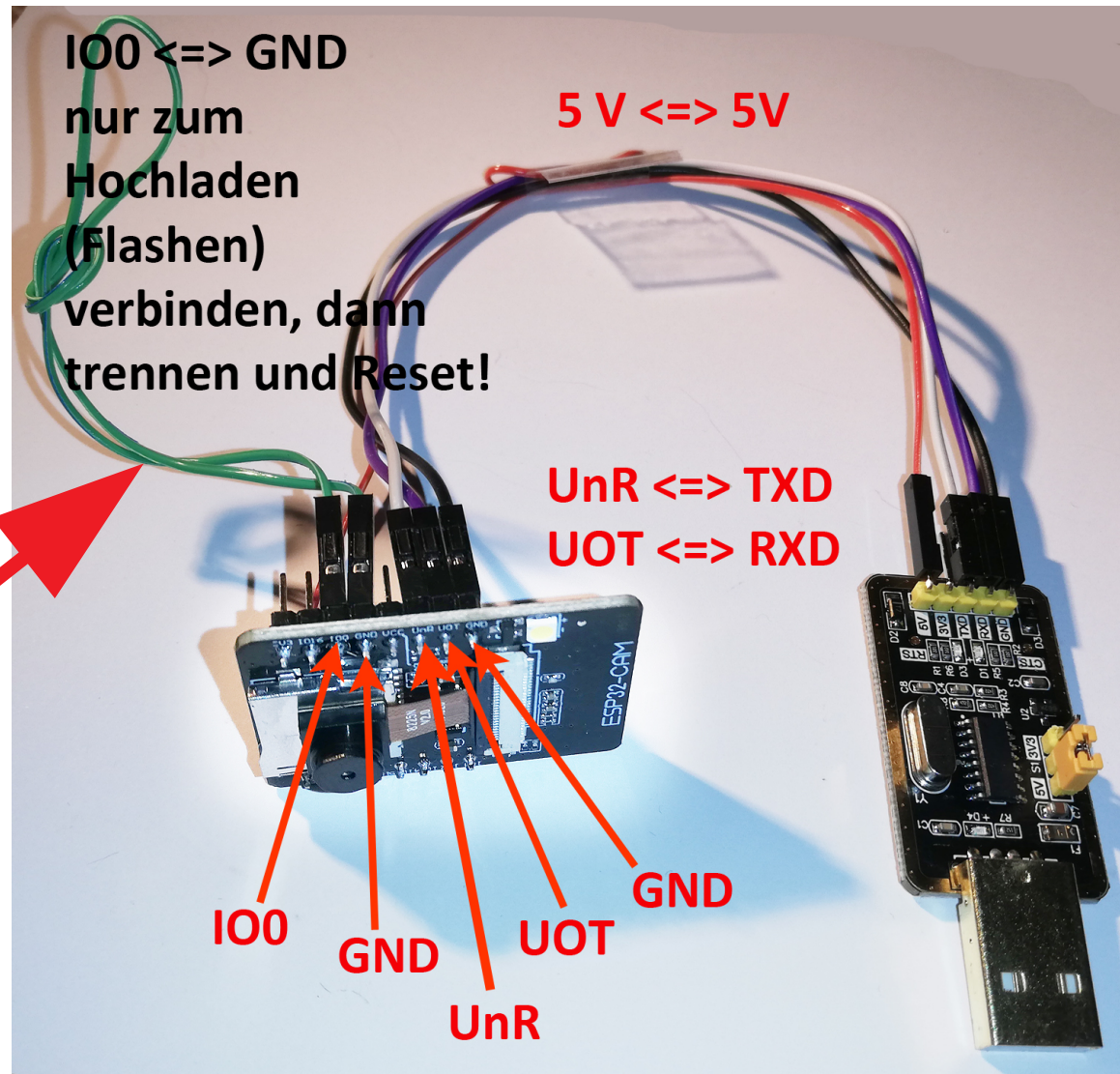
5. https://dl.espressif.com/dl/package_esp32_index.json

4. Zusätzliche Bordverwalter-URLs



Die Verbindungen zwischen USB-Seriell-Adapter und ESP32

Überbrückung zwischen
IO0 und GND nur zum
Sketch-Hochladen
(Flashen) anbringen,
anschließend trennen.



Unter „Werkzeuge“ das richtige Board wählen!

sketch_may24a | Arduino 1.8.12

Datei Bearbeiten Sketch **Werkzeuge** Hilfe

sketch_may24a

```
void setup() {  
  // put your set  
  
}  
  
void loop() {  
  // put your mai  
  
}
```

Werkzeuge:

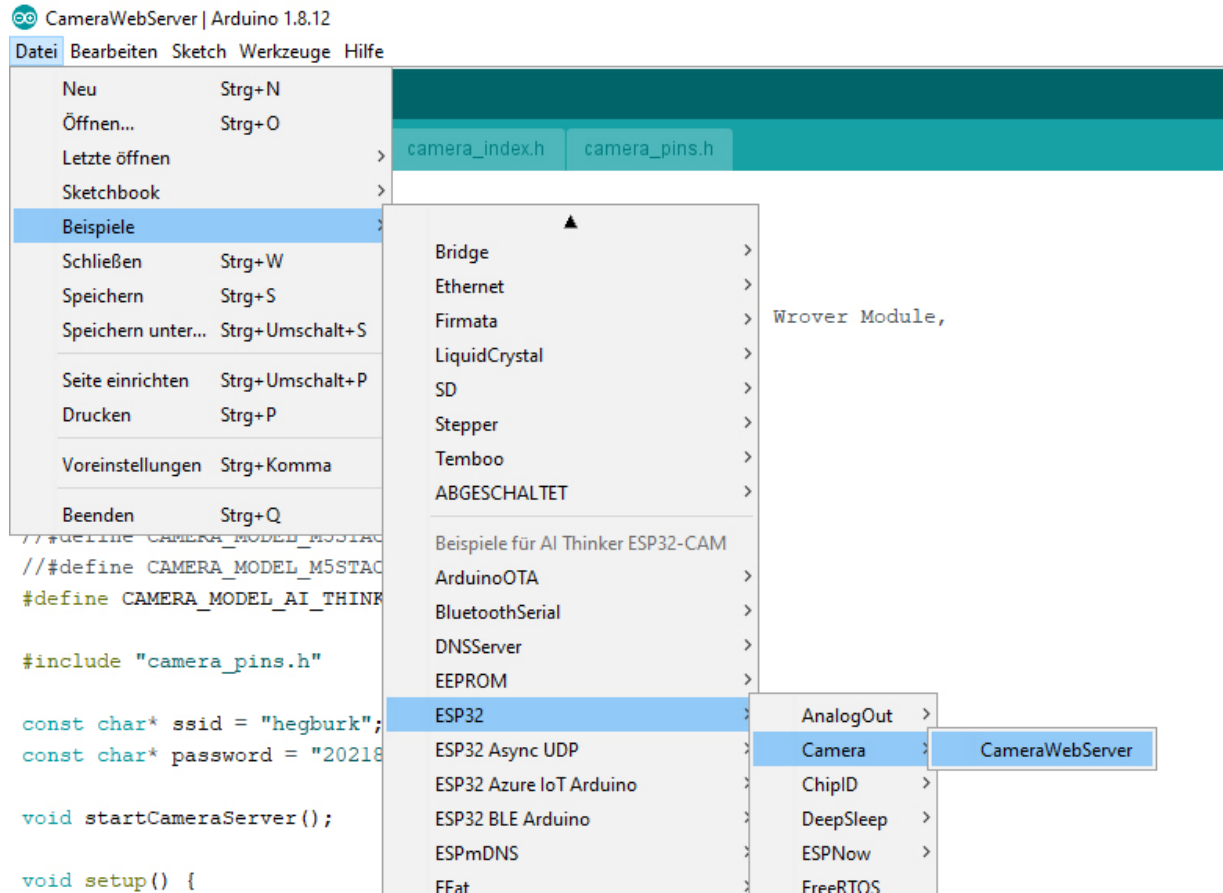
- Automatische Formatierung (Strg+T)
- Sketch archivieren
- Kodierung korrigieren & neu laden
- Bibliotheken verwalten... (Strg+Umschalt+I)
- Serieller Monitor (Strg+Umschalt+M)
- Serieller Plotter (Strg+Umschalt+L)
- WiFi101 / Wi-Fi NINA Firmware Updater
- ArduBlock
- Board: "AI Thinker ESP32-CAM"**
- Partition Scheme: "Face Recognition (2621440 bytes with OTA)"
- Port
- Boardinformationen holen
- Programmer: "ArduinoISP"
- Bootloader brennen

Boardverwalter...

- ▲
- AI Thinker ESP32-CAM**
- SparkFun LoRa Gateway 1-Channel
- TTGO T-Watch
- WEMOS D1 MINI ESP32
- Pycom GPy
- VintLabs ESP32 Devkit
- Robot Zero One

ESP8266 Boards (2 of 2)

Aus den Beispielen für ESP32 den CameraWebServer laden!



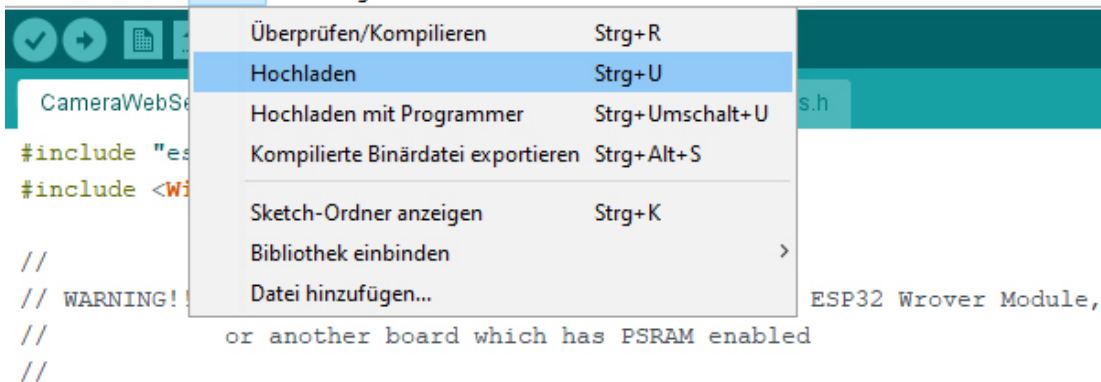
2. eigene WLAN-Zugangsdaten eingeben!



CameraWebServer-Sketch anpassen!

**Zur Kontrolle: Diese drei Dateien müssen mit
im Arduino-Sketch-Ordner sein!**





Sketch hochladen - das dauert etwas - schließlich Brücken zum Flashen entfernen und den kleinen Reset Knopf auf der ESP32-Rückseite drücken!

```
// Select camera model
// #define CAMERA_MODEL_WROVER
// #define CAMERA_MODEL_ESP_E
// #define CAMERA_MODEL_M5STA
// #define CAMERA_MODEL_M5STA
```

Hochladen...

Hash of data verified.

Compressed 2100768 bytes to 1661711...

Writing at 0x00010000... (0 %)

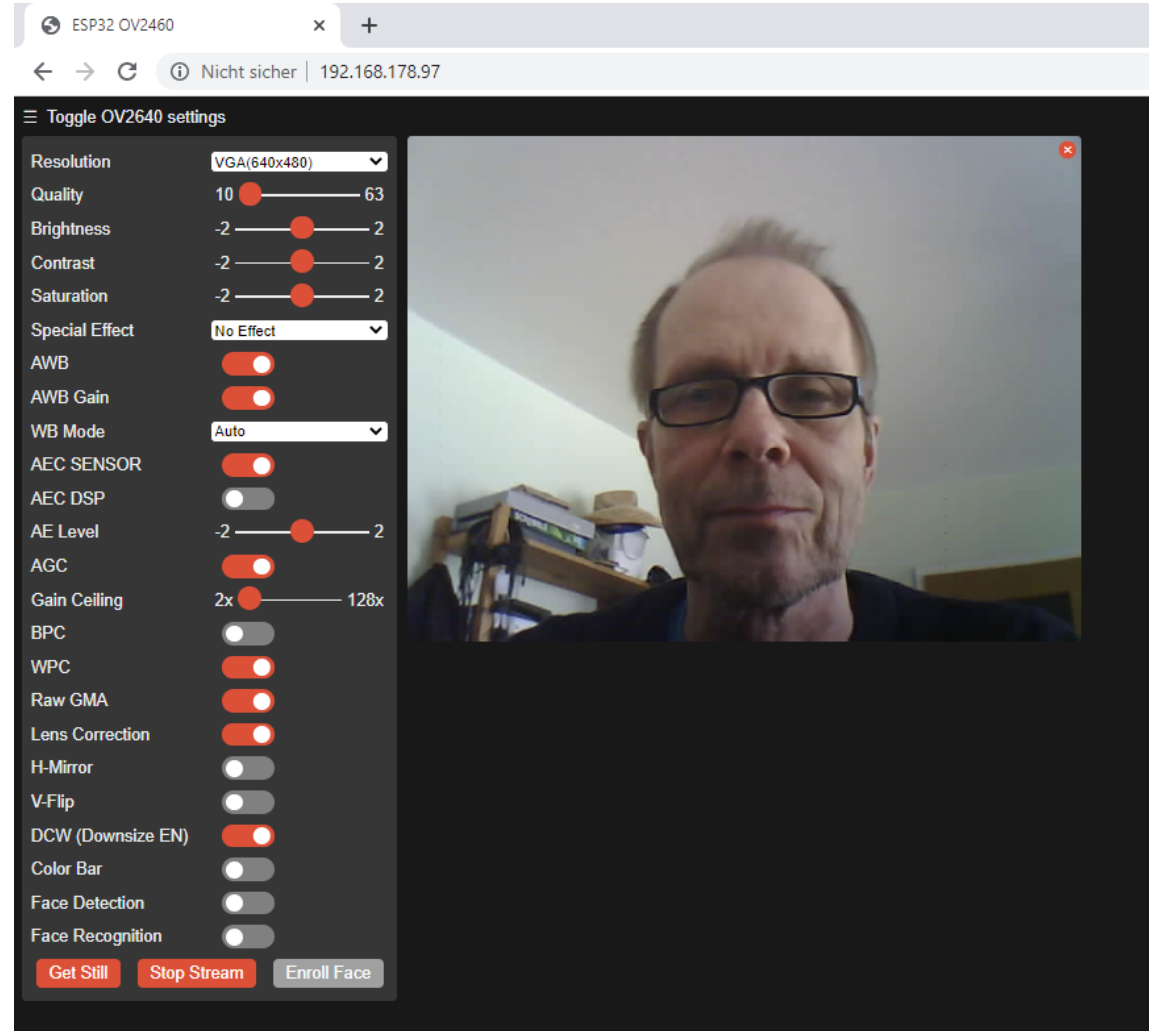
1

Hochladen abgeschlossen.

Hard resetting via RTS pin...

IP-Adresse in Browser eingeben, nochmals **Reset-Knopf drücken!**

IO0-GND-Brücke muss entfernt worden sein. Die IP-Adresse wird nach Reset im seriellen Monitor der Arduino-Entwicklungsumgebung angezeigt.



Gesichter auch nach dem Abschalten gespeichert halten!

<https://robotzero.one/esp32-face-door-entry/> - 22.05.2020

Persistent Storage Partition Scheme

A new partition scheme with persistent storage on the on-board flash is needed. You can download a scheme I created from here:

Add this file to the directory containing the other partition schemes. This is found in one of two places, depending on how you installed the Arduino IDE.

Arduino IDE installed from the Windows Store:

```
C > Users > *your-user-name* > Documents > ArduinoData > packages > esp32 > hardware > esp32 > 1.0.1 > tools > partitions
```

Arduino IDE installed from the Arduino website:

```
C > Users > *your-user-name* > AppData > Local > Arduino15 > packages > esp32 > hardware > esp32 > 1.0.1 > tools > partitions
```

The new scheme has to be added to your ESP device in the boards manager configuration file – boards.txt. Again this is found in one of two places.

Arduino IDE installed from the Windows Store:

```
C > Users > *your-user-name* > Documents > ArduinoData > packages > esp32 > hardware > esp32 > 1.0.1
```

Arduino IDE installed from the Arduino website:

```
C > Users > *your-user-name* > AppData > Local > Arduino15 > packages > esp32 > hardware > esp32 > 1.0.1
```

Add the following three lines below the existing partitionScheme options for the esp32wrover board in this boards.txt file.

```
esp32wrover.menu.PartitionScheme.rzo_partition=Face Recognition (2621440 bytes with OTA)
esp32wrover.menu.PartitionScheme.rzo_partition.build.partitions=rzo_partitions
esp32wrover.menu.PartitionScheme.rzo_partition.upload.maximum_size=2621440
```

Close and reopen the IDE to confirm the new 'Face Recognition' partition scheme is available in the Tools menu.

There's an article here that explains in much more detail how to set up a new scheme and duplicate a board definition:

Capture Face Data to Persistent Storage

The CameraWebServer example in the IDE doesn't save enrolled faces in a way that will survive power loss. To modify it to use the new partition a few changes need to be made to the code. In the Arduino IDE, make a copy of your working CameraWebServer Sketch from the previous tutorial by saving it with a new file name such as CameraWebServerPermanent.

You should see three tabs in the Arduino IDE similar to the image [...]:

In the second tab (app_httpd.cpp) make the following changes.

After `#include "fr_forward.h"` (around line 24) add:

```
#include "fr_flash.h";
```

Change `int8_t left_sample_face = enroll_face(&id_list, aligned_face);(&id_list, aligned_face);` (around line 178) to:

```
int8_t left_sample_face = enroll_face_id_to_flash(&id_list, aligned_face);
```

After `face_id_init(&id_list, FACE_ID_SAVE_NUMBER, ENROLL_CONFIRM_TIMES);` (around line 636) add:

```
read_face_id_from_flash(&id_list);
```

Flash and run this Sketch in the same way as before. Enrolled face data is now being saved to the new partition on the flash memory. [...]



Hast du deine Gesichtserkennung auf einer Reihen von Personen trainiert, dann wirst du enttäuscht sein, wenn der ESP32 sie alle vergessen hat, wenn er nach ein Abschaltung wieder hochfährt. Das lässt sich ändern, erfordert aber eine intensive Beschäftigung mit dem System. „Robertzero“ hat eine sehr gute Anleitung dazu geschrieben, die ich hier zitiere. Seine Beschreibung ist so gut, dass ich sie nicht übersetze. Wer die Möglichkeiten nutzen will, muss sich hinein vertiefen. Ich denke es lohnt sich.



ESP32_open_door_kompakt\$

camera_index.h

camera_pins.h

```
#include <ArduinoWebsockets.h>
#include "esp_http_server.h"
#include "esp_timer.h"
#include "esp_camera.h"
#include "camera_index.h"
#include "Arduino.h"
#include "fd_forward.h"
#include "fr_forward.h"
#include "fr_flash.h"
const char* ssid = "hegburk"; const char* password = "2021871308777";
#define ENROLL_CONFIRM_TIMES 5
#define FACE_ID_SAVE_NUMBER 7
#define CAMERA_MODEL_AI_THINKER
#include "camera_pins.h"

using namespace websockets; WebsocketsServer socket_server; camera_fb_t * fb = NULL; long current_millis; long last_detected_millis = 0; long last_detected_millis = 0;

void app_facenet_main(); void app_https_server_init(); typedef struct {uint8_t *image; box array_t *net_boxes; d1_matrix3d_t *face_id; } http_img_process_result;
static inline mtmm_config_t app_mtmm_config() {mtmm_config_t mtmm_config = {0}; mtmm_config.type = FAST; mtmm_config.min_face = 80; mtmm_config.pyramid = 0.707;
mtmm_config.pyramid_times = 4; mtmm_config.p_threshold.score = 0.6; mtmm_config.p_threshold.nms = 0.7; mtmm_config.p_threshold.candidate_number = 20; mtmm_config.r_threshold.score = 0.7;
mtmm_config.r_threshold.nms = 0.7; mtmm_config.r_threshold.candidate_number = 10; mtmm_config.o_threshold.score = 0.7; mtmm_config.o_threshold.nms = 0.7; mtmm_config.o_threshold.candidate_number = 1;
return mtmm_config; } mtmm_config_t mtmm_config = app_mtmm_config(); face_id_name_list_t st_face_list; static d1_matrix3du_t *aligned_face = NULL; httpd_handle_t camera_httpd = NULL;

typedef enum {START_STREAM, START_DETECT, SHOW_FACES, START_RECOGNITION, START_ENROLL, ENROLL_COMPLETE, DELETE_ALL, }
on_fsm_state; on_fsm_state g_state; typedef struct {char enroll_name[ENROLL_NAME_LEN]; } httpd_resp_value; httpd_resp_value st_name;

void setup() {pinMode(2, OUTPUT); Serial.begin(115200); Serial.setDebugOutput(true); Serial.println(); camera_config_t config; config.ledc_channel = LEDC_CHANNEL_0; config.ledc_timer = LEDC_TIMER_0;
config.pin_d0 = Y2_GPIO_NUM; config.pin_d1 = Y3_GPIO_NUM; config.pin_d2 = Y4_GPIO_NUM; config.pin_d3 = Y5_GPIO_NUM; config.pin_d4 = Y6_GPIO_NUM; config.pin_d5 = Y7_GPIO_NUM; config.pin_d6 = Y8_GPIO_NUM;
config.pin_d7 = Y9_GPIO_NUM; config.pin_xclk = XCLK_GPIO_NUM; config.pin_pclk = PCLK_GPIO_NUM; config.pin_vsync = VSYNC_GPIO_NUM; config.pin_href = HREF_GPIO_NUM; config.pin_sccb_sda = SIOD_GPIO_NUM;
config.pin_sccb_scl = SIOC_GPIO_NUM; config.pin_pwdn = PWDN_GPIO_NUM; config.pin_reset = RESET_GPIO_NUM; config.xclk_freq_hz = 20000000; config.pixel_format = PIXFORMAT_JPEG;
config.frame_size = FRAMESIZE_SVGA; config.jpeg_quality = 12; config.fb_count = 1; esp_err_t err = esp_camera_init(&config); if (err != ESP_OK) {Serial.printf("Camera init failed with error 0x%x", err); return;}
sensor_t * s = esp_camera_sensor_get(); s->set_frame_size(s, FRAMESIZE_SVGA); WiFi.begin(ssid, password); while (WiFi.status() != WL_CONNECTED) {delay(500); Serial.print("."); } Serial.println("WiFi connected");
app_https_server_init(); app_facenet_main(); socket_server.listen(82); Serial.print("Camera Ready! Use 'http://192.168.1.1'");
static esp_err_t index_handler(httpd_req_t *req) {httpd_resp_set_type(req, "text/html"); httpd_resp_set_hdr(req, "Content-Encoding", "gzip");
return httpd_resp_send(req, (const char *)index_ov2640_html_gz_len); } httpd_uri_t index_uri = { .uri = "/", .method = HTTP_GET, .handler = index_handler, .user_ctx = NULL };

void app_https_server_init() {httpd_config_t config = HTTPD_DEFAULT_CONFIG();
if (httpd_start(&camera_httpd, &config) == ESP_OK) Serial.println("httpd start"); (httpd_register_uri_handler(camera_httpd, &index_uri)); }

void app_facenet_main() {face_id_name_init(&st_face_list, FACE_ID_SAVE_NUMBER, ENROLL_CONFIRM_TIMES); aligned_face = d1_matrix3du_alloc(1, FACE_WIDTH, FACE_HEIGHT, 3);
read_face_id_from_flash_with_name(&st_face_list); static inline int do_enrollment(face_id_name_list_t *face_list, d1_matrix3du_t *new_id)
{ESP_LOGD(TAG, "START ENROLLING"); int left_sample_face = enroll_face_id_to_flash_with_name(face_list, new_id, st_name.enroll_name);
ESP_LOGD(TAG, "Face ID %s Enrollment: Sample %d", st_name.enroll_name, ENROLL_CONFIRM_TIMES - left_sample_face); return left_sample_face; }

static esp_err_t send_face_list(WebsocketsClient &client) {client.send("delete faces"); face_id_name_t *head = st_face_list.head; char add_face[64];
for (int i = 0; i < st_face_list.count; i++) {sprintf(add_face, "list:face:%s", head->id_name); client.send(add_face); head = head->next; }
static esp_err_t delete_all_faces(WebsocketsClient &client) {delete_face_all_in_flash_with_name(&st_face_list); client.send("delete faces"); }

void handle_message(WebsocketsClient &client, WebsocketsMessage msg) {if (msg.data() == "stream") {g_state = START_STREAM; client.send("STREAMING"); }
if (msg.data() == "detect") {g_state = START_DETECT; client.send("DETECTING"); } if (msg.data().substring(0, 8) == "capture:")
{g_state = START_ENROLL; char person[FACE_ID_SAVE_NUMBER * ENROLL_NAME_LEN] = {0}; msg.data().substring(8).toCharArray(person, sizeof(person));
memcpy(st_name.enroll_name, person, strlen(person) + 1); client.send("CAPTURING"); if (msg.data() == "recognise") {g_state = START_RECOGNITION; client.send("RECOGNISING"); } if (msg.data().substring(0, 7) == "remove:")
{char person[ENROLL_NAME_LEN * FACE_ID_SAVE_NUMBER]; msg.data().substring(7).toCharArray(person, sizeof(person)); delete_face_id_in_flash_with_name(&st_face_list, person);
send_face_list(client); if (msg.data() == "delete all") {delete_all_faces(client); last_detected_millis = millis(); } void loop() {auto client = socket_server.accept(); client.onMessage(handle_message);
d1_matrix3du_t *image_matrix = d1_matrix3du_alloc(1, 320, 240, 3); http_img_process_result out_res = {0}; out_res.image = image_matrix->item; send_face_list(client); client.send("STREAMING");
while (client.available()) {client.poll(); fb = esp_camera_fb_get(); if (g_state == START_DETECT || g_state == START_ENROLL || g_state == START_RECOGNITION)
{out_res.net_boxes = NULL; out_res.face_id = NULL; fmt2rgb888(fb->buf, fb->len, fb->format, out_res.image); out_res.net_boxes = face_detect(image_matrix, &mtmm_config);
if (out_res.net_boxes) {if (align_face(out_res.net_boxes, image_matrix, aligned_face) == ESP_OK) {out_res.face_id = get_face_id(aligned_face); last_detected_millis = millis(); }
if (g_state == START_DETECT) {client.send("FACE DETECTED"); } if (g_state == START_ENROLL)
{int left_sample_face = do_enrollment(&st_face_list, out_res.face_id); client.send(enrolling_message[64]); printf(enrolling_message, "SAMPLE NUMBER %d FOR %s",
ENROLL_CONFIRM_TIMES - left_sample_face, st_name.enroll_name); client.send(enrolling_message); if (left_sample_face == 0) {ESP_LOGI(TAG, "Enrolled Face ID: %s", st_face_list.tail->id_name);
g_state = START_STREAM; char captured_message[64]; sprintf(captured_message, "FACE CAPTURED FOR %s", st_face_list.tail->id_name);
client.send(captured_message); send_face_list(client); } } if (g_state == START_RECOGNITION && (st_face_list.count > 0))
{face_id_name_t *f = recognise_face_with_name(&st_face_list, out_res.face_id); if (f) {char recognised_message[64]; sprintf(recognised_message, "%s", f->id_name); Serial.println(recognised_message);
client.send(recognised_message); digitalWrite(2, HIGH); delay(5000); digitalWrite(2, LOW); } else {client.send("FACE NOT RECOGNISED"); } d1_matrix3du_free(out_res.face_id); } else {if (g_state == START_DETECT) {client.send("NO FACE DETECTED"); }
if (g_state == START_DETECT && millis() - last_detected_millis > 5000) {client.send("DETECTING"); } client.sendBinary((const char *)fb->buf, fb->len); esp_camera_fb_return(fb); fb = NULL; } }
```

Der Code – eng an „Robotzero“ angelehnt und kompakt dargestellt – ist in dieser Form lauffähig, wenn sich im Sketchordner auch die beiden Dateien „camera_index.h“ und „camera_pins.h“ enthalten sind. Diese beiden Dateien werden auf der nächsten Seite noch erläutert und stehen über die Seiten von „Robotzero“ zum Download bereit.


```

#include <ArduinoWebsockets.h>
#include "esp_http_server.h"
#include "esp_timer.h"
#include "esp_camera.h"
#include "camera_index.h"
#include "Arduino.h"
#include "fd_forward.h"
#include "fr_forward.h"
#include "fr_flash.h"
const char* ssid = "                ";const char* password = "                ";
#define ENROLL_CONFIRM_TIMES 5
#define FACE_ID_SAVE_NUMBER 7
#define CAMERA_MODEL_AI_THINKER
#include "camera_pins.h"
using namespace websockets; WebsocketsServer socket_server;camera_fb_t *fb = NULL;long current_millis;long last_detected_millis = 0;long last_detected_millis1 = 0;
void app_facenet_main();void app_httpsrvr_init();typedef struct{uint8_t *image;box_array_t *net_boxes;dl_matrix3d_t *face_id;http_img_process_result;
static inline mtmn_config_t app_mtmn_config(){mtmn_config_t mtmn_config = {0};mtmn_config.type = FAST;mtmn_config.min_face = 80;mtmn_config.pyramid = 0.707;
mtmn_config.pyramid_times = 4;mtmn_config.p_threshold.score = 0.6;mtmn_config.p_threshold.nms = 0.7;mtmn_config.p_threshold.candidate_number = 20;mtmn_config.r_threshold.score = 0.7;
mtmn_config.r_threshold.nms = 0.7;mtmn_config.r_threshold.candidate_number = 10;mtmn_config.o_threshold.score = 0.7;mtmn_config.o_threshold.nms = 0.7;mtmn_config.o_threshold.candidate_number = 1;
return mtmn_config;}mtmn_config_t mtmn_config = app_mtmn_config();face_id_name_list_t face_list;static dl_matrix3du_t *aligned_face = NULL;httpd_handle_t camera_httpd = NULL;
typedef enum{START_STREAM,START_DETECT,SHOW_FACES,START_RECOGNITION,START_ENROLL,ENROLL_COMPLETE,DELETE_ALL;}
en_fsm_state_t en_fsm_state;typedef struct{char enroll_name[ENROLL_NAME_LEN];httpd_resp_value_t httpd_resp_value;httpd_resp_value_t st_name;
void setup() {pinMode(2, OUTPUT);Serial.begin(115200);Serial.setDebugOutput(true);Serial.println();camera_config_t config;config.ledc_channel = LEDC_CHANNEL_0;config.ledc_timer = LEDC_TIMER_0;
config.pin_d0 = Y2_GPIO_NUM;config.pin_d1 = Y3_GPIO_NUM;config.pin_d2 = Y4_GPIO_NUM;config.pin_d3 = Y5_GPIO_NUM;config.pin_d4 = Y6_GPIO_NUM;config.pin_d5 = Y7_GPIO_NUM;config.pin_d6 = Y8_GPIO_NUM;
config.pin_d7 = Y9_GPIO_NUM;config.pin_xclk = XCLK_GPIO_NUM;config.pin_pclk = PCLK_GPIO_NUM;config.pin_vsync = VSYNC_GPIO_NUM;config.pin_href = HREF_GPIO_NUM;config.pin_sscb_sda = SIOD_GPIO_NUM;
config.pin_sscb_scl = SIOC_GPIO_NUM;config.pin_pwdn = PWDN_GPIO_NUM;config.pin_reset = RESET_GPIO_NUM;config.xclk_freq_hz = 20000000;config.pixel_format = PIXFORMAT_JPEG;
config.frame_size = FRAMESIZE_SVGA;config.jpeg_quality = 12;config.fb_count = 1;esp_err_t err = esp_camera_init(&config);if (err != ESP_OK) {Serial.printf("Camera init failed with error 0x%x", err);return;}
sensor_t *s = esp_camera_sensor_get();s->set_framesize(s, FRAMESIZE_QVGA);WiFi.begin(ssid, password);while (WiFi.status() != WL_CONNECTED) {delay(500);Serial.print(".");};Serial.println("WiFi connected");
app_httpsrvr_init();app_facenet_main();socket_server.listen(82);Serial.print("Camera Ready! Use 'http://'");Serial.println(WiFi.localIP());
static esp_err_t index_handler(httpd_req_t *req){httpd_resp_set_type(req, "text/html");httpd_resp_set_hdr(req, "Content-Encoding", "gzip");
return httpd_resp_send(req, (const char *)index_ov2640_html_gz, index_ov2640_html_gz_len);}httpd_uri_t index_uri={.uri="/",.method=HTTP_GET,.handler=index_handler,.user_ctx=NULL};
void app_httpsrvr_init () {httpd_config_t config = HTTPD_DEFAULT_CONFIG();
if (httpd_start(&camera_httpd, &config) == ESP_OK)Serial.println("httpd_start");httpd_register_uri_handler(camera_httpd, &index_uri);}
void app_facenet_main(){face_id_name_init(&st_face_list, FACE_ID_SAVE_NUMBER, ENROLL_CONFIRM_TIMES);aligned_face = dl_matrix3du_alloc(1, FACE_WIDTH, FACE_HEIGHT, 3);
read_face_id_from_flash_with_name(&st_face_list);static inline int do_enrollment(face_id_name_list_t *face_list, dl_matrix3d_t *new_id)
{ESP_LOGD(TAG, "START ENROLLING");int left_sample_face = enroll_face_id_to_flash_with_name(face_list, new_id, st_name.enroll_name);
ESP_LOGD(TAG, "Face ID %s Enrollment: Sample %d", st_name.enroll_name, ENROLL_CONFIRM_TIMES - left_sample_face);return left_sample_face;}
static esp_err_t send_face_list(WebsocketsClient &client){client.send("delete_faces");face_id_node_t head = st_face_list.head;char add_face[64];
for (int i = 0; i < st_face_list.count; i++){sprintf(add_face, "listface:%s", head->id_name);client.send(add_face);head = head->next;}}
static esp_err_t delete_all_faces(WebsocketsClient &client){delete_face_all_in_flash_with_name(&st_face_list);client.send("delete_faces");}
void handle_message(WebsocketsClient &client, WebsocketsMessage msg){if (msg.data() == "stream"){g_state = START_STREAM;client.send("STREAMING");}
if (msg.data() == "detect"){g_state = START_DETECT;client.send("DETECTING");if (msg.data().substring(0, 8) == "capture:"){
g_state = START_ENROLL;char person[FACE_ID_SAVE_NUMBER * ENROLL_NAME_LEN] = {0};msg.data().substring(8).toCharArray(person, sizeof(person));
memcpy(st_name.enroll_name, person, strlen(person) + 1);client.send("CAPTURING");if (msg.data() == "recognise"){g_state = START_RECOGNITION;client.send("RECOGNISING");if (msg.data().substring(0, 7) == "remove:"){
char person[ENROLL_NAME_LEN * FACE_ID_SAVE_NUMBER];msg.data().substring(7).toCharArray(person, sizeof(person));delete_face_id_in_flash_with_name(&st_face_list, person);
send_face_list(client);if (msg.data() == "delete_all"){delete_all_faces(client);last_detected_millis1=millis();}void loop(){auto client = socket_server.accept();client.onMessage(handle_message);
dl_matrix3du_t *image_matrix = dl_matrix3du_alloc(1, 320, 240, 3);http_img_process_result out_res = {0};out_res.image = image_matrix->item;send_face_list(client);client.send("STREAMING");
while (client.available()){client.poll();fb = esp_camera_fb_get();if (g_state == START_DETECT || g_state == START_ENROLL || g_state == START_RECOGNITION)
{out_res.net_boxes = NULL;out_res.face_id = NULL;fmt2rgb888(fb->buf, fb->len, fb->format, out_res.image);out_res.net_boxes = face_detect(image_matrix, &mtmn_config);
if (out_res.net_boxes){if (align_face(out_res.net_boxes, image_matrix, aligned_face) == ESP_OK){out_res.face_id = get_face_id(aligned_face);last_detected_millis = millis();}
if (g_state == START_DETECT) {client.send("FACE DETECTED");}if (g_state == START_ENROLL)
{int left_sample_face = do_enrollment(&st_face_list, out_res.face_id);char enrolling_message[64];sprintf(enrolling_message, "SAMPLE NUMBER %d For %s",
ENROLL_CONFIRM_TIMES - left_sample_face, st_name.enroll_name);client.send(enrolling_message);if (left_sample_face == 0){ESP_LOGI(TAG, "Enrolled Face ID: %s", st_face_list.tail->id_name);
g_state = START_STREAM;char captured_message[64];sprintf(captured_message, "FACE CAPTURED FOR %s", st_face_list.tail->id_name);
client.send(captured_message);send_face_list(client);} if (g_state == START_RECOGNITION && (st_face_list.count > 0))
{face_id_node_t * = recognize_face_with_name(&st_face_list, out_res.face_id);if (!){char recognised_message[64];sprintf(recognised_message, "%s", f->id_name);Serial.println(recognised_message);
client.send(recognised_message);digitalWrite(2,HIGH);delay(5000);digitalWrite(2,LOW);}else{client.send("FACE NOT RECOGNISED");}dl_matrix3d_free(out_res.face_id);}else{if (g_state!=START_DETECT){client.send("NO FACE DETECTED");}
if (g_state == START_DETECT && millis()-last_detected_millis>500) {client.send("DETECTING");}client.sendBinary((const char *)fb->buf, fb->len);esp_camera_fb_return(fb);fb = NULL;}}

```

ESP32_open_door_kompakt - camera_index.h | Arduino 1.8.12

Datei Bearbeiten Sketch Werkzeuge Hilfe

```
ESP32_open_door_kompakt camera_index.h camera_pins.h

//File: index_ov2640.html.gz, Size: 4316
#define index_ov2640_html_gz_len 4316
const uint8_t index_ov2640_html_gz[] = {
0x1f, 0x8b, 0x08, 0x08, 0x76, 0x8e, 0x5e, 0x5d, 0x00, 0xff, 0x69, 0x6e, 0x64, 0x65, 0x78, 0x2e, 0x68, 0x74, 0x6d, 0x6c, 0x2e
};
```

Dies ist ein kompakte verschlüsselt, Google-Stichwort „Compress HTML“.

Erläuterungen zu beiden zusätzlichen Dateien im Sketch-Ordner

ESP32_open_door_kompakt - camera_pins.h | Arduino 1.8.12

Datei Bearbeiten Sketch Werkzeuge Hilfe

```
ESP32_open_door_kompakt camera_index.h camera_pins.h

#define Y6_GPIO_NUM 39
#define Y5_GPIO_NUM 5
#define Y4_GPIO_NUM 34

#elif defined(CAMERA_MODEL_AI_THINKER)
#define PWDN_GPIO_NUM 32
#define RESET_GPIO_NUM -1
#define XCLK_GPIO_NUM 0
#define SIOD_GPIO_NUM 26
#define SIOC_GPIO_NUM 27

#define Y9_GPIO_NUM 35
#define Y8_GPIO_NUM 34
#define Y7_GPIO_NUM 39
#define Y6_GPIO_NUM 36
#define Y5_GPIO_NUM 21
#define Y4_GPIO_NUM 19
#define Y3_GPIO_NUM 18
#define Y2_GPIO_NUM 5
#define VSYNC_GPIO_NUM 25
#define HREF_GPIO_NUM 23
#define PCLK_GPIO_NUM 22

#else
#error "Camera model not selected"
#endif
```

Hier holt sich der Sketch die richtigen Pin-Zuordnungen.

Namentliche akustische Ansprache identifizierter Personen über Raspberry Pi

Wenn man die namentlich Identifikation zu einer sprachlichen Begrüßung nutzen will, dann man über den seriellen Adapter die Namen einem Raspberry -Pi zuführen, der sie dann in in eine Audio-Datei umwandelt, diese abspielt und über den Lautsprecher-Ausgang ausgibt. Bei meinen Versuch hat es sich als notwendig erwiesen die, Geschwindigkeit der seriellen Schnittstelle auf 9600 Baud einzustellen Außerdem ließen sich die Lautsprecher-Signale nur rauschfrei wiedergeben, wenn Verstärker und Raspberry Pi nicht an der gleichen Stromquelle angeschlossen waren. Hier habe ich mit zwei Powerbanks gearbeitet.

SerialText2Voice.py - E:\CH@PH\RS+\SmartHome\Face\SerialText2Voice.py

File Edit Format Run Options Window Help

```
#!/usr/bin/env python3
#!/usr/bin/python3
from gtts import gTTS
import pygame
import serial
pygame.mixer.init()
line2="Hi!";line3="Hallo";message="Hi"
ser = serial.Serial('/dev/ttyUSB0',9600)
pygame.mixer.init()
while True:
    try:
        line = ser.readline()
    except ValueError:
        pass
    try:
        line2 = line.decode("utf-8")
        line3 = line2[:-1]
    except ValueError:
        pass
    print(line3)
    message = "Hi "+line3
    tts = gTTS(text=message, lang='en')
    tts.save("test.mp3")
    try:
        pygame.mixer.music.load("test.mp3")
        pygame.mixer.music.play()
    except ValueError:
        pass
    message == ""
```

