

esp32_sendet_emails_vom_bme280

Diesen Beitrag gibt es auch als [PDF-Dokument](#).

Um aus der Pampa mit dem ESP32/ESP8266 Nachrichten als SMS versenden zu können, braucht es ein [GSM-Modul](#). Es steht ja dort selten ein LAN zur Verfügung. Darüber, dass es von daheim aus mit einem WLAN-Zugang der ESPs auf vielfältige Weise über [TCP/HTTP](#), [MQTT](#) oder [UDP](#) auch klappt, habe ich auch schon geschrieben. Die Links sind nur eine kleine Auswahl. Mit TCP wird eine gesicherte bidirektionale Verbindung aufgebaut. So kann man auch von Auswärts auf einen Webserver auf den ESPs zugreifen, wenn auf dem DSL-Router eine Portweiterleitung eingerichtet ist. Allerdings braucht es bei HTTP eine Anfrage vom Browser eines Clients, um die Verbindung aktiv herzustellen. Mit MQTT brauche ich einen Broker auf einem Heimserver, zum Beispiel einem Raspi, der den Datenaustausch koordiniert, Informationen sammelt und weitergibt.

In manchen Fällen ist es aber besser, wenn sich der ESP32 von sich aus beim Client, also einem Handy oder Tablet meldet, um eine Nachricht zu hinterlassen. Deren Eingang kann mit einem Signal verbunden werden. Die Rede ist von e-Mails. Ja, und richtig, der ESP32 und der ESP8266, können beide e-Mail-Nachrichten aus einem LAN heraus versenden. Wie das geht, das erfahren Sie in dieser neuen Folge aus der Reihe

MicroPython auf dem ESP32 und ESP8266

heute

e-Mail-Nachrichten von ESP32 und ESP8266 – Teil 1

Gleich vorneweg: e-Mails versenden können beide, der ESP32 wie auch der ESP8266, denn beide verfügen über ein WLAN-Interface. Das Testprogramm **e-mail.py** läuft ohne Änderung auf beiden Systemen. Das erforderliche Modul micro-Mail in **umail.py** ist mit seinen 126 Programmzeilen recht überschaubar.

Anders sieht es mit dem Modul **bme280** aus. Die Anwendung **bme280-monitor.py**, die auf das Modul zurückgreift, ist aus Platzgründen nur auf einem ESP32 lauffähig.

Als Anwendung bietet sich aber jedes Ereignis an, das sich mittels Sensoren erfassen und vom Controller auswerten lässt – Land unter im Keller, Mausefalle hat zugeschnappt, Einbrecher in der Wohnung, und so weiter. Was wir also für dieses Projekt in jedem Fall brauchen, ist ein Controller. Ob es ein ESP8266 sein kann oder ein ESP32 sein muss, das hängt letztlich von der Art und Vielfalt der weiteren Komponenten ab. Für jeden Wasserstandssensor reicht ein GPIO-Pin. Das gilt auch für einen akustischen Alarmgeber. Für ein Display brauche ich zwei bis drei, je nachdem die Anzeige am I2C- oder am SPI-Bus hängt. Beim ESP8266 sind die digitalen Eingänge mit 9 Stück nicht gerade im Überfluss vorhanden und darüber hinaus durch andere Funktionen, meist den Start des Systems betreffend, in der Einsetzbarkeit stark eingeschränkt. Das zeigt die Tabelle 1.

| Label | GPIO | Input | Output | Notes |
|-------|--------|---------------|-------------------------|--|
| D0 | GPIO16 | Kein IRQ | Kein PWM or I2C support | HIGH at boot wake up from deep sleep |
| D1 | GPIO5 | OK | OK | SCL bei I2C-Nutzung |
| D2 | GPIO4 | OK | OK | SDA bei I2C-Nutzung |
| D3 | GPIO0 | pulled up | OK | FLASH button, wenn LOW Normales Booten, wenn HIGH |
| D4 | GPIO2 | pulled up | OK | Muss beim Booten HIGH sein verbunden mit der On-Board-LED, LOW aktiviert LED |
| D5 | GPIO14 | OK | OK | SPI (SCLK) |
| D6 | GPIO12 | OK | OK | SPI (MISO) |
| D7 | GPIO13 | OK | OK | SPI (MOSI) |
| D8 | GPIO15 | pulled to GND | OK | SPI (CS) Muss beim Booten LOW sein |
| RX | GPIO3 | OK | RX pin | Muss beim Booten HIGH sein |
| TX | GPIO1 | TX pin | OK | Muss beim Booten HIGH sein Debugausgang beim Booten |
| A0 | ADC0 | Analog Input | X | |

Tabelle 1: Pinbelegung und Systemfunktionen beim ESP8266

Das gilt es bei der Auswahl des Controllers zu beachten. Der ESP32 ist dagegen in dieser Hinsicht wesentlich anspruchsloser.

Hardware

Allein zum Versenden von e-Mails ist im Prinzip jeder der angeführten Typen einsetzbar. Die Auflistung der Bauteile enthält neben dem Controller auch bereits die Teile, die in der nächsten Folge zum Einsatz kommen werden.

Um den Zustand der Schaltung jederzeit auch direkt vor Ort einsehen zu können, habe ich dem ESP ein kleines Display spendiert, das über den I2C-Bus angesteuert wird. Es ist sogar grafikfähig und könnte daher auch zeitliche Änderungen des Messsignals als Kurve darstellen. Über die Flash-Taste wäre eine Umschaltung zwischen Text- und Grafikmodus machbar und bei längerem Drücken ein geordneter Abbruch des Programms, falls zum Beispiel Aktoren sicher ausgeschaltet werden müssen.

Als Messanwendung habe ich mich für einen Klimamonitor mit dem BME280 entschieden. Der Bosch-Sensor kann Luftdruck, relative Luftfeuchte und Temperatur erfassen. Der Chef in meiner Schaltung wird daher ein ESP32 sein.

| | |
|----------|---|
| 1 | D1 Mini NodeMcu mit ESP8266-12F WLAN Modul oder D1 Mini V3 NodeMCU mit ESP8266-12F oder NodeMCU Lua Amica Modul V2 ESP8266 ESP-12F WIFI oder NodeMCU Lua Lolin V3 Module ESP8266 ESP-12F WIFI oder ESP32 Dev Kit C unverlötet oder ESP32 Dev Kit C V4 unverlötet oder ESP32 NodeMCU Module WLAN WiFi Development Board mit CP2102 oder NodeMCU-ESP-32S-Kit oder ESP32 Lolin LOLIN32 WiFi Bluetooth Dev Kit |
| 1 | 0,91 Zoll OLED I2C Display 128 x 32 Pixel |
| 1 | GY-BME280 Barometrischer Sensor für Temperatur, Luftfeuchtigkeit und Luftdruck |
| 1 | MB-102 Breadboard Steckbrett mit 830 Kontakten |
| diverse | Jumper Wire Kabel 3 x 40 STK. je 20 cm M2M/ F2M / F2F evtl. auch 65Stk. Jumper Wire Kabel Steckbrücken für Breadboard |
| optional | Logic Analyzer |

Die Software

Fürs Flashen und die Programmierung des ESP32:

[Thonny](#) oder
[uPyCraft](#)

Verwendete Firmware für den ESP32:

[v1.19.1 \(2022-06-18\) .bin](#)

Verwendete Firmware für den ESP8266:

[v1.19.1 \(2022-06-18\) .bin](#)

Die MicroPython-Programme zum Projekt:

[ssd1306.py](#) Hardwaretreiber für das OLED-Display

[oled.py](#) API für das OLED-Display

[umail.py](#) Micro-Mail-Modul

[e-mail.py](#) Demoprogramm für den e-Mailversand

MicroPython - Sprache - Module und Programme

Zur Installation von Thonny finden Sie hier eine [ausführliche Anleitung \(english version\)](#). Darin gibt es auch eine Beschreibung, wie die [Micropython-Firmware](#) (Stand 18.06.2022) auf den ESP-Chip [gebrannt](#) wird.

MicroPython ist eine Interpretersprache. Der Hauptunterschied zur Arduino-IDE, wo Sie stets und ausschließlich ganze Programme flashen, ist der, dass Sie die MicroPython-Firmware nur einmal zu Beginn auf den ESP32 flashen müssen, damit der Controller MicroPython-Anweisungen versteht. Sie können dazu Thonny, µPyCraft oder esptool.py benutzen. Für Thonny habe ich den Vorgang [hier](#) beschrieben.

Sobald die Firmware geflasht ist, können Sie sich zwanglos mit Ihrem Controller im Zwiegespräch unterhalten, einzelne Befehle testen und sofort die Antwort sehen, ohne vorher ein ganzes Programm kompilieren und übertragen zu müssen. Genau das stört mich nämlich an der Arduino-IDE. Man spart einfach enorm Zeit, wenn man einfache Tests der Syntax und der Hardware bis hin zum Ausprobieren und Verfeinern von Funktionen und ganzen Programmteilen über die Kommandozeile vorab prüfen kann, bevor man ein Programm daraus strickt. Zu diesem Zweck erstelle ich auch gerne immer wieder kleine Testprogramme. Als eine Art Makro fassen sie wiederkehrende Befehle zusammen. Aus solchen Programmfragmenten entwickeln sich dann mitunter ganze Anwendungen.

Autostart

Soll das Programm autonom mit dem Einschalten des Controllers starten, kopieren Sie den Programmtext in eine neu angelegte Blankodatei. Speichern Sie diese Datei unter `boot.py` im Workspace ab und laden Sie sie zum ESP-Chip hoch. Beim nächsten Reset oder Einschalten startet das Programm automatisch.

Programme testen

Manuell werden Programme aus dem aktuellen Editorfenster in der Thonny-IDE über die Taste F5 gestartet. Das geht schneller als der Mausklick auf den Startbutton, oder über das Menü **Run**. Lediglich die im Programm verwendeten Module müssen sich im Flash des ESP32 befinden.

Zwischendurch doch mal wieder Arduino-IDE?

Sollten Sie den Controller später wieder zusammen mit der Arduino-IDE verwenden wollen, flashen Sie das Programm einfach in gewohnter Weise. Allerdings hat der ESP32/ESP8266 dann vergessen, dass er jemals MicroPython gesprochen hat. Umgekehrt kann jeder Espressif-Chip, der ein kompiliertes Programm aus der Arduino-IDE oder die AT-Firmware oder LUA oder ... enthält, problemlos mit der MicroPython-Firmware versehen werden. Der Vorgang ist immer so, wie [hier](#) beschrieben.

Eine e-Mail vom Controller

Es ist aus verschiedenen Gründen keine gute Idee, vom eigenen e-Mail-Konto aus Nachrichten durch einen ESP **versenden** zu lassen. Dagegen sprechen eindeutig Sicherheitsaspekte und auch eine mögliche Sperrung des Kontos, wenn der ESP Mist baut und zum Beispiel sehr viele Mails innerhalb kurzer Zeit versendet oder andere Fehler verursacht, die der Provider ahndet. Deshalb beginne ich hier mit der Einrichtung eines neuen Google-Kontos, über das der ESP dann arbeiten kann. Als Empfänger können Sie ruhig Ihr normales Mail-Konto benutzen, beim gleichen Provider oder einem anderen, das ist egal.

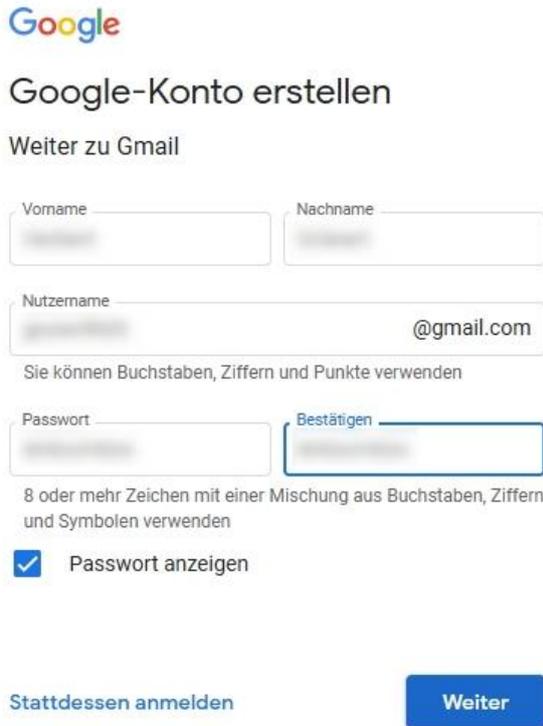
Ein Google-Konto einrichten

Mit einem Handy, Tablet oder PC kann/muss man sich heute mit 2-Weg-Authentifizierung bei einem e-Mail-Provider anmelden, das ist der sichere Weg. Beim Anmelden anderer Geräte, wie zum Beispiel einem Microcontroller, funktioniert das nicht. Sie können den Authentifizierungslink oder die SMS ja nicht empfangen und bestätigen. Es ist schon einige Zeit her, als in solchen Fällen nur der Ausweg über die unsichere einfache Authentifizierung über Benutzername und Passwort möglich war. Das wurde von einigen Providern aber nicht mehr unterstützt.

Ich stelle Ihnen heute eine Möglichkeit vor, mit Ihrem ESP trotzdem eine Mail zu versenden. Das gelingt mit Hilfe eines App-Passworts, einer Kombination von 16 Zeichen, die Sie einer bestimmten Anwendung oder einem Gerät zuordnen können.

Beginnen wir mit der [Erstellung eines neuen Google-Kontos](#). Folgen Sie dem Link und klicken Sie auf **Konto erstellen**.

Folgen Sie jetzt der Benutzerführung. Kleiner Tipp, Es ist nicht nötig, Ihre tatsächlichen persönlichen Daten wie Name und Geburtsdatum anzugeben, Sie können sie auch faken. Nur müssen Sie spätestens im nächsten Kapitel über die Telefon- oder Handynummer erreichbar sein, um Ihre e-Mailadresse zu bestätigen.



The screenshot shows the Google account creation interface. At the top is the Google logo. Below it is the heading 'Google-Konto erstellen' and the sub-heading 'Weiter zu Gmail'. The form consists of several input fields: 'Vorname' (First name) and 'Nachname' (Last name) are side-by-side; 'Nutzername' (Username) is a single wide field with '@gmail.com' pre-filled; 'Passwort' (Password) and 'Bestätigen' (Confirm) are side-by-side. Below the password fields is a checkbox labeled 'Passwort anzeigen' (Show password) which is checked. At the bottom left is a link 'Stattdessen anmelden' (Sign in instead) and at the bottom right is a blue button labeled 'Weiter' (Next).

Abbildung 1: Name und Passwort erfassen

Im ersten Schritt geben Sie einen Namen an, Google generiert daraus einige Mailadressen. Suchen Sie eine aus oder geben Sie eine ein, aber bitte **nicht Ihre Haupt-Mailadresse!** Bauen Sie ein Passwort. - Weiter



Willkommen bei Google

grunerth05@gmail.com

Google nutzt diese Nummer nur zum Schutz Ihres Kontos. Die Nummer ist für andere Personen nicht sichtbar. Sie können später entscheiden, ob Sie die Nummer auch zu anderen Zwecken nutzen möchten.

Damit schützen wir Ihr Konto

Tag Monat Jahr

Ihr Geburtsdatum

Geschlecht



Ihre personenbezogenen Daten sind bei uns sicher und geschützt

Warum wir nach diesen Informationen fragen

[Zurück](#)

[Weiter](#)

Abbildung 2: Geburtsdatum und Geschlecht

Das Geburtsdatum kann wie der Name gefakt werden. Wählen Sie am besten eins aus den Jahren vor 2004. – Weiter



Personalisierungseinstellungen auswählen

- Express-Personalisierung (1 Schritt)**
Nutzen Sie diese Personalisierungseinstellungen, um Inhalte und Werbung zu erhalten, die auf Ihre Interessen abgestimmt sind. Wir erinnern Sie in einigen Wochen daran, Ihre Einstellungen zu überprüfen.
- Manuelle Personalisierung (5 Schritte)**
Legen Sie Ihre Personalisierungseinstellungen Schritt für Schritt fest. Sie entscheiden, welche Einstellungen aktiviert oder deaktiviert sind, damit die Inhalte und die Werbung, die Ihnen angezeigt werden, Ihren Vorstellungen entsprechen.

Unter account.google.com können Sie Ihre Einstellungen jederzeit ändern

[Weiter](#)



Sie haben Kontrolle über die von Google erhobenen Daten und deren Nutzung

Abbildung 3: Personalisierungseinstellungen

Damit Sie erfahren, was alles zu Personalisierung gehört wählen Sie Manuelle Personalisierung, auch wenn das länger dauert.



Web- & App-Aktivitäten für eine noch schnellere Suche speichern

Schritt 1 von 5

Wählen Sie aus, ob Sie Web- & App-Aktivitäten speichern möchten

- Aktivitäten speichern, bis ich sie manuell lösche
- Aktivitäten 18 Monate lang speichern mit der Option, sie jederzeit manuell zu löschen
- Web- & App-Aktivitäten nicht in meinem Konto speichern



Sie haben Kontrolle über die von Google erhobenen Daten und deren Nutzung

Abbildung 4: Aktivitäten nicht speichern

Sie werden ihr Mailkonto selten abrufen, da ist es Ihnen sicher egal, was so alles gespeichert werden kann. – Weiter



YouTube-Verlauf für eine bessere YouTube-Startseite speichern

Schritt 2 von 5

Wählen Sie aus, ob Sie den YouTube-Verlauf speichern möchten

- Aktivitäten speichern, bis ich sie manuell lösche
- Aktivitäten 36 Monate lang speichern mit der Option, sie jederzeit manuell zu löschen
- YouTube-Verlauf nicht in meinem Konto speichern



Sie haben Kontrolle über die von Google erhobenen Daten und deren Nutzung

Abbildung 5: YouTube-Verlauf nicht speichern

Auch der YouTube-Verlauf kann Ihnen wurscht sein, auch was an Werbung reinflattert. – Weiter - Weiter



Wenn Sie mehr individuelle Werbeanzeigen erhalten möchten, aktivieren Sie "Personalisierte Werbung"

Schritt 3 von 5

Wählen Sie aus, ob personalisierte Werbung aktiviert werden soll

- Personalisierte Werbung einblenden**
Werbung wird dann anhand Ihrer Aktivitäten und anderer Daten auf Sie zugeschnitten und Sie können Werbetreibende oder Werbethemen sperren, an denen Sie nicht interessiert sind
-
- Allgemeine Werbung einblenden**
Es wird nach wie vor Werbung eingeblendet, allerdings ist diese für Sie möglicherweise weniger relevant. Sie basiert auf allgemeinen Faktoren wie der Tageszeit, Ihrem allgemeinen Standort oder den Inhalten der Website, auf der Sie sich gerade befinden.

Abbildung 6: Allgemeine Werbung



Sie haben Kontrolle über die von Google erhobenen Daten und deren Nutzung



Gelegentlich Erinnerungen für den Privatsphärecheck erhalten

Schritt 4 von 5

Legen Sie fest, ob Sie gelegentlich daran erinnert werden möchten, einen Privatsphärecheck durchzuführen und wichtige Datenschutzeinstellungen zu prüfen

- Auswählen, welche Arten von Daten wir speichern dürfen
 - Aktualisieren, was Sie mit Freunden teilen oder veröffentlichen
 - Festlegen, was für Werbung für Sie eingeblendet werden soll
- Erinnerungen für den Privatsphärecheck**
Gelegentlich per E-Mail Erinnerungen zu diesen Einstellungen erhalten

Unter account.google.com können Sie Ihre Einstellungen jederzeit ändern

Abbildung 7: Keine Erinnerungen



Sie haben Kontrolle über die von Google erhobenen Daten und deren Nutzung

Weil Sie ja vermutlich nix an den Einstellungen ändern wollen, brauchen Sie auch keine Erinnerungen. – Weiter – Und dann war's das auch schon.



Einstellungen für Personalisierung und Cookies bestätigen

Schritt 5 von 5

- Web- & App-Aktivitäten
Diese Einstellung wird deaktiviert
- YouTube-Verlauf
Diese Einstellung wird deaktiviert
- Personalisierte Werbung
Diese Einstellung wird deaktiviert
- Erinnerungen für den Privatsphärecheck
Gelegentliche Erinnerungen per E-Mail sind nicht
eingrichtet



Sie haben Kontrolle über die von
Google erhobenen Daten und
deren Nutzung

Abbildung 8: Zusammenfassung

Ganz unten finden Sie einen Weiter-Button und dann kommen Sie mit Linksklick auf GoogleKonto links oben zur Startseite ihres neuen Accounts. Melden Sie sich jetzt ab.

Erstellen eines App-Passworts

Nach der Neuansmeldung bekommen Sie wahrscheinlich eine Meldung eingeblendet.

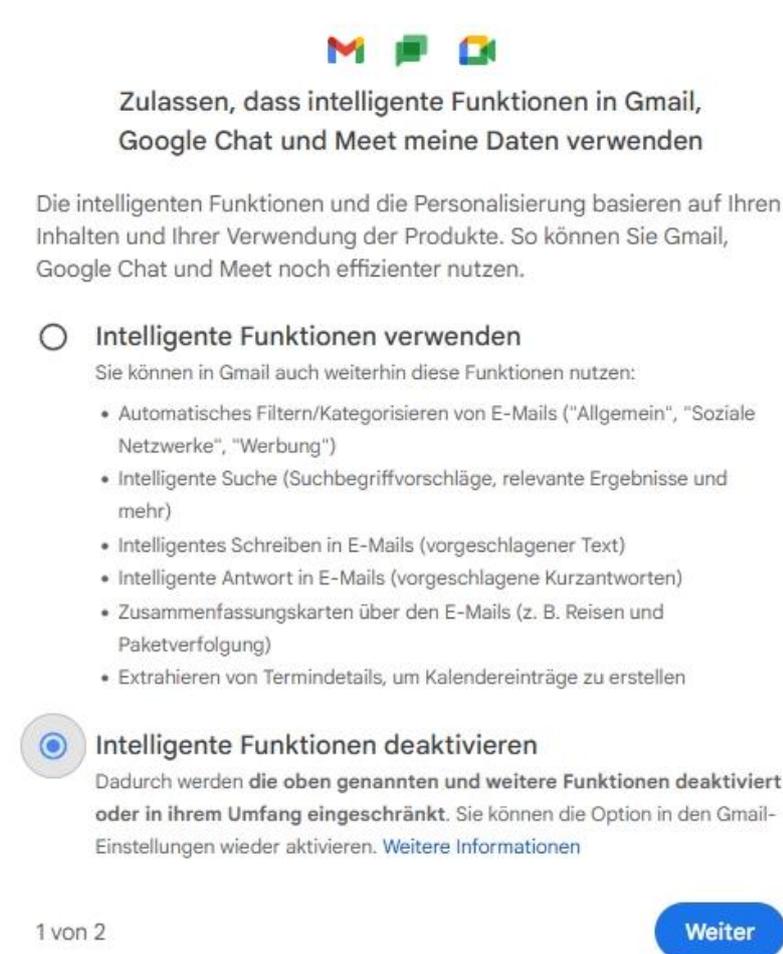
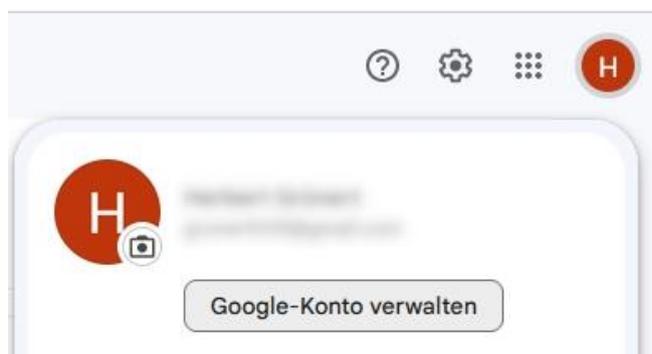


Abbildung 9: Intelligente Funktionen deaktivieren

Die intelligenten Funktionen können Sie deaktivieren, weil der ESP32/ESP8266 die eh nicht nutzen wird.

Auf der Startseite rechts oben klicken Sie auf dem Button mit Ihrem Initial. Mit Linksklick auf **Google-Konto verwalten** wird links ein Menü eingeblendet. Klicken Sie dort auf **Sicherheit**.



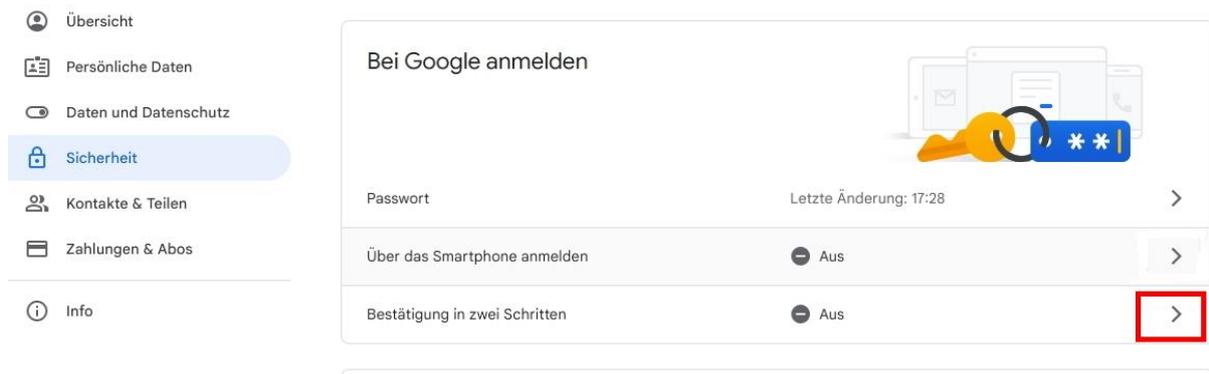


Abbildung 11: Bei Google anmelden

Im Fenster Sicherheit scrollen Sie bis **Bei Google anmelden**. Bevor ein App-Passwort vergeben werden kann, muss **Bestätigung in zwei Schritten** aktiviert werden. Starten Sie mit Klick auf das Größer-Zeichen.

← Bestätigung in zwei Schritten

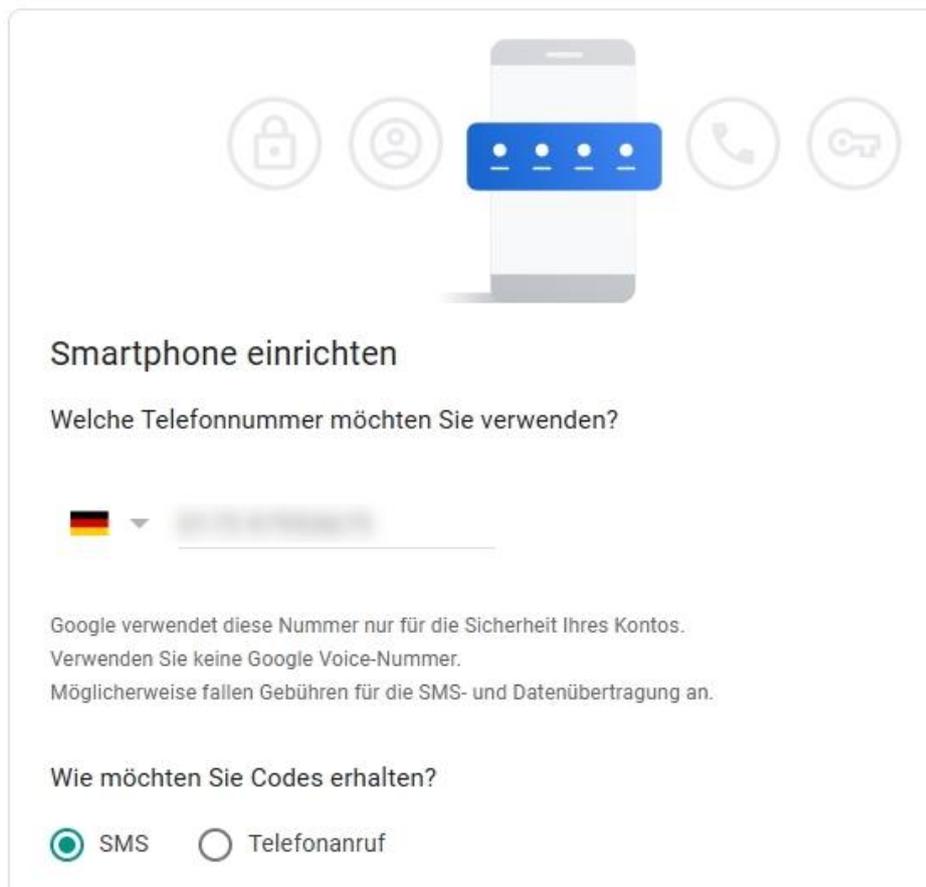
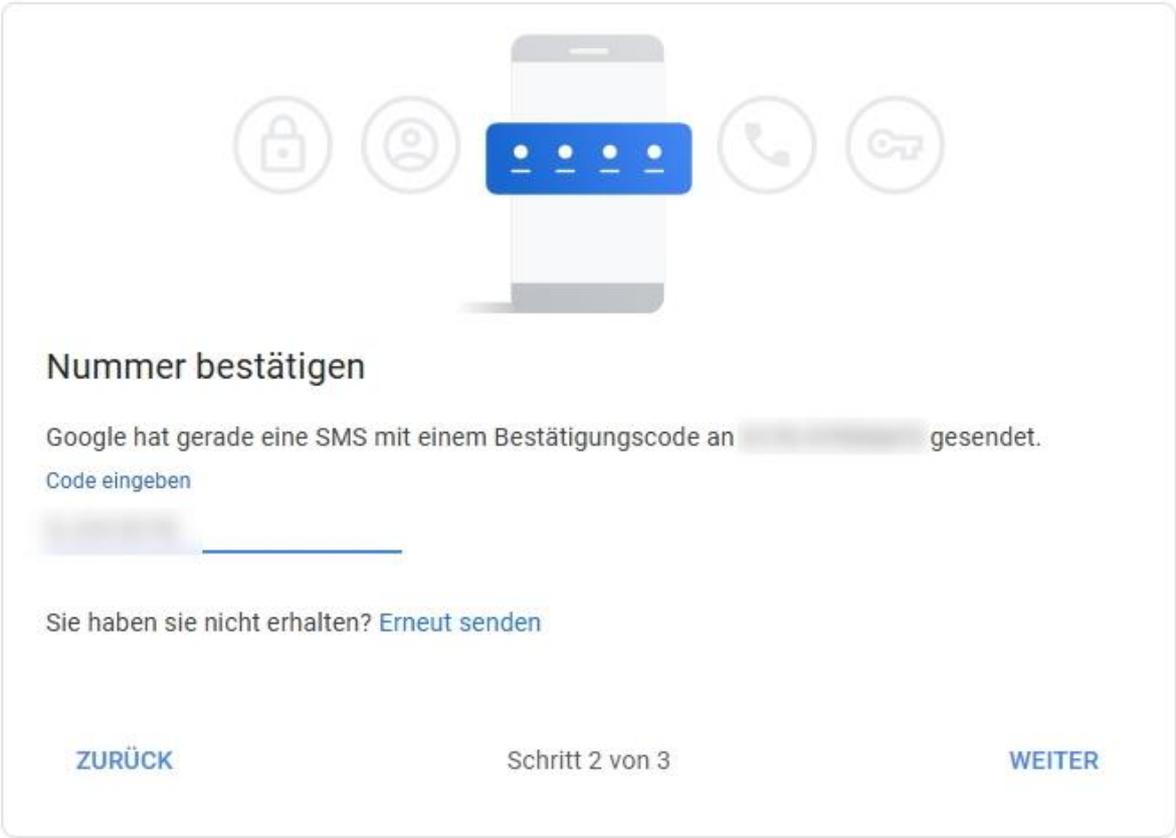


Abbildung 12: Smartphone einrichten

Jetzt brauchen Sie die Handynummer. Hier müssen Sie leider Ihre eigene angeben, weil Google Ihnen einen Bestätigungscode schickt.

← Bestätigung in zwei Schritten



Nummer bestätigen

Google hat gerade eine SMS mit einem Bestätigungscode an [blurred] gesendet.

Code eingeben

[blurred] _____

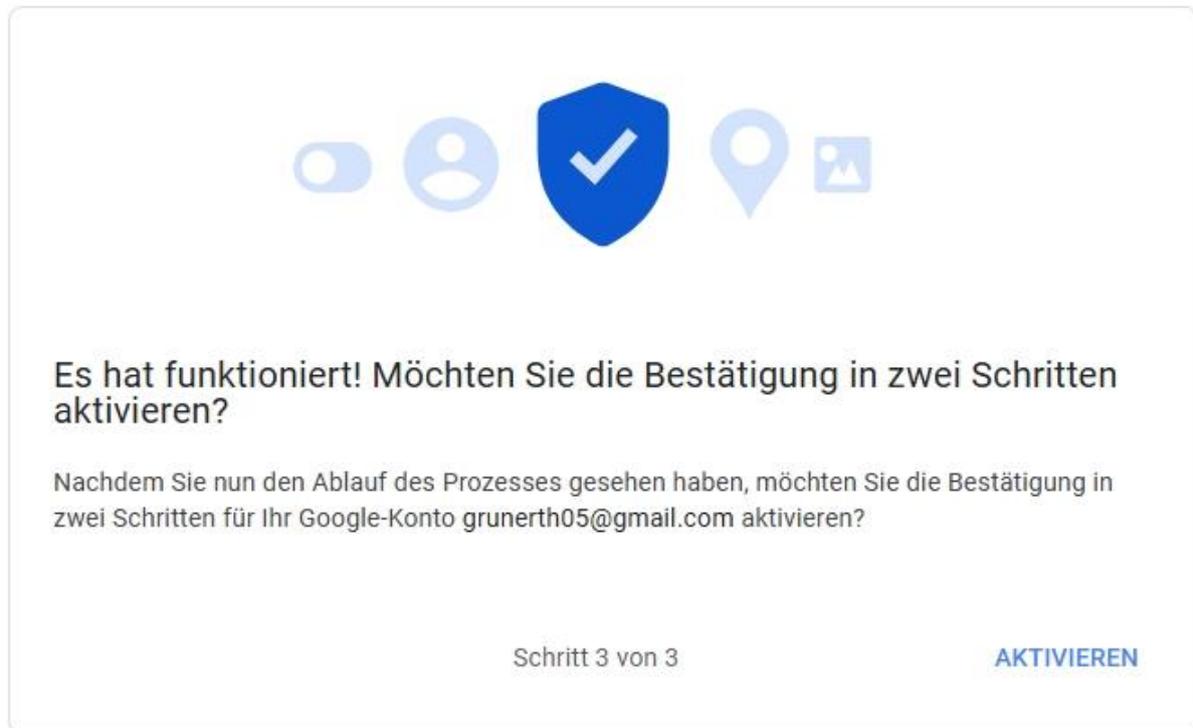
Sie haben sie nicht erhalten? [Erneut senden](#)

[ZURÜCK](#) Schritt 2 von 3 [WEITER](#)

Abbildung 13: Nummer bestätigen

Per SMS erhalten Sie einen sechsstelligen Bestätigungscode. – Weiter – Aktivieren

← Bestätigung in zwei Schritten



The screenshot shows a confirmation screen for enabling two-step verification. At the top, there are five icons: a toggle switch, a person icon, a shield with a checkmark, a location pin, and a photo icon. The main text asks if the user wants to activate two-step verification for their Google account. Below this, it provides the email address 'grunerth05@gmail.com'. At the bottom, it indicates 'Schritt 3 von 3' and has an 'AKTIVIEREN' button.

Es hat funktioniert! Möchten Sie die Bestätigung in zwei Schritten aktivieren?

Nachdem Sie nun den Ablauf des Prozesses gesehen haben, möchten Sie die Bestätigung in zwei Schritten für Ihr Google-Konto grunerth05@gmail.com aktivieren?

Schritt 3 von 3 AKTIVIEREN

Abbildung 14: Aktivieren

← Bestätigung in zwei Schritten



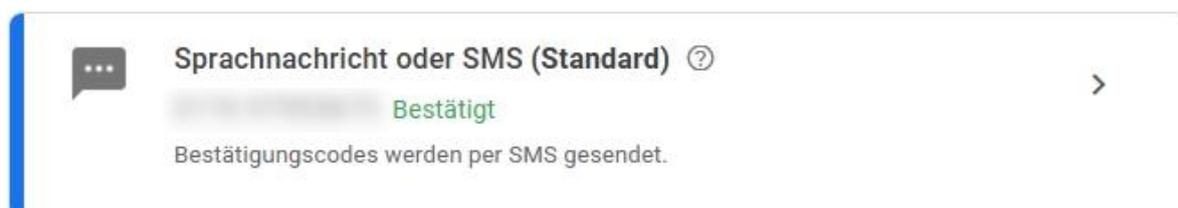
The screenshot shows a status bar for two-step verification. It displays 'Bestätigung in zwei Schritten – AKTIVIERT' and a 'DEAKTIVIEREN' button.

Bestätigung in zwei Schritten – AKTIVIERT DEAKTIVIEREN

Mögliche zweite Schritte

Nachdem Sie Ihr Passwort eingegeben haben, verwenden Sie einen zweiten Schritt, um Ihre Identität für die Anmeldung zu bestätigen. [Weitere Informationen](#)

Hinweis: Sie erhalten auf kompatiblen Smartphones, auf denen Sie sich anmelden, als weitere Option zur Bestätigung in zwei Schritten auch Aufforderungen von Google.



The screenshot shows a notification for two-step verification. It has a speech bubble icon, the title 'Sprachnachricht oder SMS (Standard)', a status 'Bestätigt', and a right arrow. Below the notification, it says 'Bestätigungs-codes werden per SMS gesendet.'

... Sprachnachricht oder SMS (Standard) ? >

Bestätigt

Bestätigungs-codes werden per SMS gesendet.

Abbildung 15: Bestätigung aktiviert

Mit Klick auf **GoogleKonto** kommen Sie zurück zur Verwaltungsseite. Scrollen Sie erneut in **Sicherheit** bis **Bei Google anmelden**.



Abbildung 16: Sicherheit - App-Passwörter

Es ist noch kein App-Passwort eingerichtet, also erzeugen wir ein erstes. Sie können auch mehrere erstellen, für jeden Client ein eigenes. Klick aufs Größer-Zeichen.



Abbildung 17: Erneute Authentifizierung

Google fordert eine erneute Authentifizierung mit Username und Passwort. – Weiter

← App-Passwörter

Mithilfe von App-Passwörtern können Sie sich auf Geräten, die die Bestätigung in zwei Schritten nicht unterstützen, über Apps in Ihrem Google-Konto anmelden. Sie müssen das Passwort nur einmal eingeben und es sich daher nicht merken. [Weitere Informationen](#)

Sie haben keine App-Passwörter.

Wählen Sie die App und das Gerät aus, für die Sie das App-Passwort erstellen möchten.

App auswählen

- E-Mail
- Kalender
- Kontakte
- YouTube
- Andere (*benutzerdefinierter Name*)

Gerät auswählen

GENERIEREN

Abbildung 18: App auswählen

Als App wählen Sie **e-Mail** aus, als Gerät **Andere**.

← App-Passwörter

Mithilfe von App-Passwörtern können Sie sich auf Geräten, die die Bestätigung in zwei Schritten nicht unterstützen, über Apps in Ihrem Google-Konto anmelden. Sie müssen das Passwort nur einmal eingeben und es sich daher nicht merken. [Weitere Informationen](#)

Sie haben keine App-Passwörter.

Wählen Sie die App und das Gerät aus, für die Sie das App-Passwort erstellen möchten.

E-Mail

Gerät auswählen

- iPhone
- iPad
- BlackBerry
- Mac
- Windows Phone
- Windows-Computer
- Andere (*benutzerdefinierter Name*)

GENERIEREN

Abbildung 19; Gerät auswählen

Dann geben Sie einen beliebigen Gerätenamen an. - Generieren

← App-Passwörter

Mithilfe von App-Passwörtern können Sie sich auf Geräten, die die Bestätigung in zwei Schritten nicht unterstützen, über Apps in Ihrem Google-Konto anmelden. Sie müssen das Passwort nur einmal eingeben und es sich daher nicht merken. [Weitere Informationen](#)

Sie haben keine App-Passwörter.

Wählen Sie die App und das Gerät aus, für die Sie das App-Passwort erstellen möchten.

ESP32

Abbildung 20: Gerätenamen angeben

Generiertes App-Passwort

Email

Password

Ihr App-Passwort für Ihr Gerät



Verwendung

Öffnen Sie in der App oder auf dem Gerät, die bzw. das Sie einrichten möchten, die Einstellungen Ihres Google-Kontos. Ersetzen Sie Ihr Passwort durch das oben angezeigte 16-stellige Passwort.

Wie mit Ihrem normalen Passwort haben Sie auch mit diesem App-Passwort vollen Zugriff auf Ihr Google-Konto. Sie müssen es sich jedoch nicht merken und sollten es weder aufschreiben noch an Dritte weitergeben.

FERTIG

Abbildung 21: Generiertes App-Passwort merken

Auch wenn es bei **Verwendung** anders steht, **merken Sie sich in jedem Fall das erzeugte Passwort**, eine Folge von 16 Kleinbuchstaben. Sie brauchen diese später beim Programm. Bewahren Sie den Code sicher auf. - Fertig

← App-Passwörter

Mithilfe von App-Passwörtern können Sie sich auf Geräten, die die Bestätigung in zwei Schritten nicht unterstützen, über Apps in Ihrem Google-Konto anmelden. Sie müssen das Passwort nur einmal eingeben und es sich daher nicht merken. [Weitere Informationen](#)

Meine App-Passwörter

| Name | Erstellt | Zuletzt verwendet |
|-------|----------|-------------------|
| ESP32 | 17:56 | - |

Wählen Sie die App und das Gerät aus, für die Sie das App-Passwort erstellen möchten.

App auswählen Gerät auswählen

GENERIEREN

Abbildung 22: Das app-Passwort ist gespeichert

Damit ist das App-Passwort erzeugt, und Sie können loslegen.

Sicherheit

- Kontakte & Teilen
- Zahlungen & Abos
- Info

Bei Google anmelden

Passwort Letzte Änderung: 17:28

Bestätigung in zwei Schritten Ein

App-Passwörter 1 Passwort

Abbildung 23: Jetzt können Sie loslegen

Wir versenden e-Mails

Das SMTP-Protokoll (Simple Mail Transfer Protocol), das zum Versand von e-Mails eingesetzt wird, ist ein menschenlesbares Protokoll, das auf TCP-Datenströmen aufsetzt. Für die Übertragung wird also zwischen Client und Server eine gesicherte, bidirektionale Verbindung ausgehandelt. "gesichert" bezieht sich hier auf die Eigenschaften verlustfrei und fehlerkorrigierend, nicht auf abhörsicher. Gleichwohl wird eine Art Verschlüsselung zum Transfer von Username und Passwort angewandt, [Base64](#). Bei diesem Coding werden aus drei normalen Bytes vier ASCII-Zeichen aus dem Bereich A-Z, a-z, 0-9,+ und /. Anhänge, wie zum Beispiel Bilder, werden auch in Base64 codiert. Das erklärt, warum das Volumen bei der Bildübertragung zunimmt.

Laden Sie sich jetzt als Erstes die MicroPython-Datei [umail.py](#) herunter. Speichern Sie sie in Ihrem Arbeitsverzeichnis (`_workspace` von Thonny). Laden Sie die Datei auf den ESP32/ESP8266 hoch. Das Original stammt von [Shawwnn](#), für Debugging-Zwecke habe ich einige print-Zeilen eingebaut, die auch die Struktur einer Übertragung offenbaren. Die Ausgabe im Terminal erfolgt, wenn Sie beim Konstruktoraufruf den Parameter `debug = True` setzen.

```
smtp = umail.SMTP('smtp.gmail.com', 465, ssl=True, debug=True)
```

Aber gehen wir der Reihe nach vor.

```
import umail
import network
import sys
from time import sleep
from machine import SoftI2C, Pin

# Geben Sie hier Ihre eigenen Zugangsdaten an
mySSID = 'EMPIRE_OF_ANTS'
myPass = 'nightingale'
```

Einige Importe versorgen uns mit den nötigen Zutaten, `umail` ist hier die wichtigste. Für `mySSID` und `myPass` geben Sie bitte die [Credentials](#) für Ihren WLAN-Router an.

```
class MailError(Exception):
    pass

class UnkownPortError(MailError):
    def __init__(self):
        print("System-Fehler\nunbekannter Port")
        print("Nur ESP32, ESP8266 werden unterstützt")
```

Die beiden Exception-Klassen dienen der Fehlerbehandlung bei der Bestimmung des Ports.

```
if sys.platform == "esp8266":
    i2c=SoftI2C(scl=Pin(5),sda=Pin(4))
elif sys.platform == "esp32":
    i2c=SoftI2C(scl=Pin(22),sda=Pin(21),freq=100000)
else:
    raise UnkownPortError()
```

Eine `UnkownPortError`-Exception wird geworfen, wenn sich der Controller weder als ESP32 noch als ESP8266 identifiziert.

```
sender_email = 'ernohub@gmail.com'
sender_name = 'ESP32' #sender name
sender_app_password = 'xxxxxxxxxxxxxxxxxxx'
='meine@mail.org'
email_subject ='Test e-Mail'
```

sender_email ist der Google-Account mit dem zugehörigen App-Passwort, das wir oben erzeugt haben. Bei **recipient_email** geben Sie Ihre Mailadresse an, unter der Sie die Meldungen empfangen möchten.

```
connectStatus = {
    1000: "STAT_IDLE",
    1001: "STAT_CONNECTING",
    1010: "STAT_GOT_IP",
    202: "STAT_WRONG_PASSWORD",
    201: "NO AP FOUND",
    5: "UNKNOWN"
}
```

Das [Dict](#) **connectStatus** übersetzt die Statuscodes des Station-Interfaces in Klartext.

```
def hexMac(byteMac):
    """
    Die Funktion hexMAC nimmt die MAC-Adresse im Bytecode und
    bildet daraus einen String fuer die Rueckgabe
    """
    macString = ""
    for i in range(0, len(byteMac)):
        macString += hex(byteMac[i])[2:]
        if i < len(byteMac) - 1:
            macString += "-"
    return macString
```

Die Funktion **hexMac()** verrät uns die MAC-Adresse der WLAN-Schnittstelle.

```
>>> %Run -c $EDITOR_CONTENT

OLED API OK
Size:128x32
BME280 initialized @ 0x76
STATION MAC:    f0-8-d1-d2-1e-94
```

Abbildung 24: MAC-Abfrage

Diese muss im Router eingetragen werden, damit der Controller vom Türsteher (MAC-Filter) Einlass erhält. Bei einem TP-LINK-Router sieht das so aus.

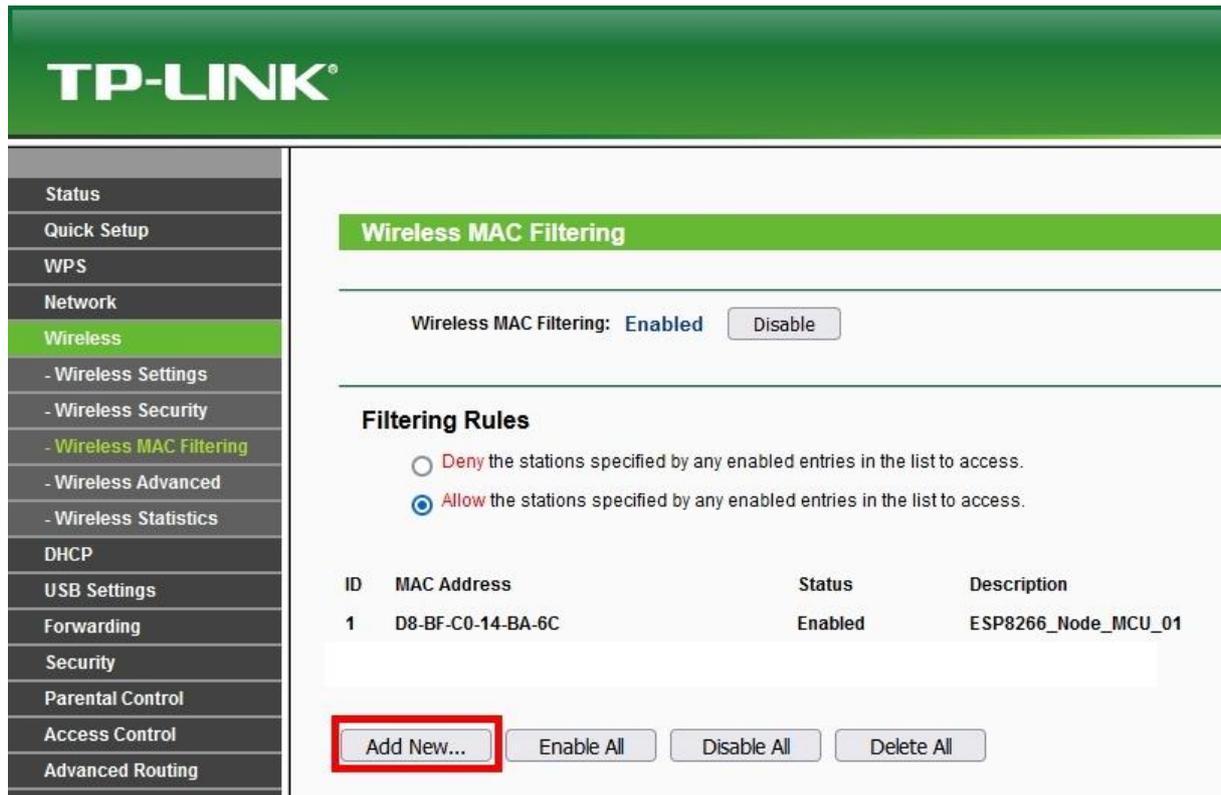


Abbildung 25: MAC-Filtering

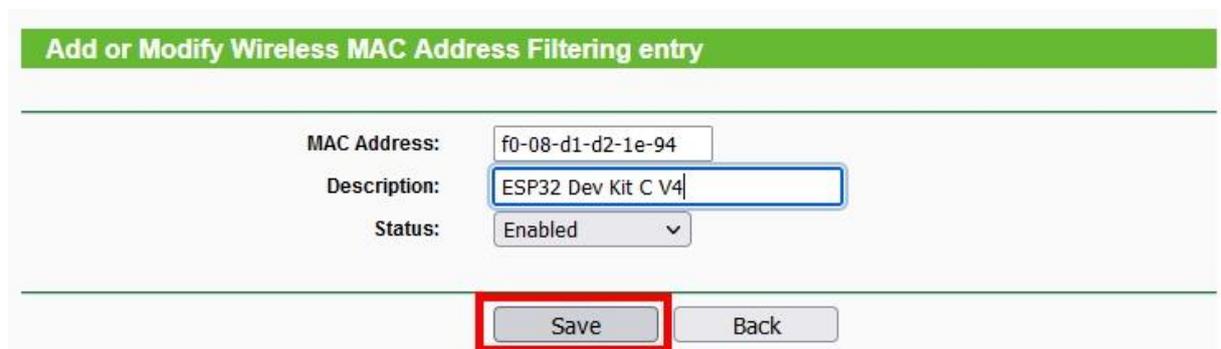


Abbildung 26: MAC-Filtering - Eintrag

Dann bauen wir eine Verbindung zum Router auf. Während der Vorgang läuft, werden im Sekundenabstand Punkte im Terminal ausgegeben.

Eine Mail zu versenden ist Dank des Moduls **umail** kein großer Act.

```
# ***** Eine Mail versenden *****
#
smtp = umail.SMTP('smtp.gmail.com', 465, ssl=True, debug=True)
smtp.login(sender_email, sender_app_password)
smtp.to(recipient_email)
smtp.write("From:" + sender_name + "<" + sender_email + ">\n")
smtp.write("Subject:" + email_subject + "\n")
smtp.write("Greetings from ESP32/ESP8266")
smtp.send()
smtp.quit()
```

Der Konstruktoraufruf erzeugt ein SMTP-Objekt. Wir übergeben die URL des Servers und die Portnummer des SMTP-Portals. Dann senden wir unsere G-Mail-Kennung und das App-Passwort. Der Empfänger wird übertragen, danach der Name und die Mailadresse des Absenders. Es folgt ein Betreff und dann senden wir den Text der Nachricht. Längere Texte können in mehrere **write**-Anweisungen aufgeteilt werden. Bisher haben wir die Daten nur an den G-Mail-Server übertragen, der sie fleißig aufgesaugt hat. Mit **send** geben wir den Auftrag für die Weiterleitung an uns als Empfänger, bevor wir die Verbindung zum Server kappen.

In der nächsten Blogfolge werden wir den BME als Messwertaufnehmer für Druck, rel. Feuchte und Temperatur in Dienst stellen. Im Programm können wir eine zeitliche Steuerung vorsehen sowie eine Alarmfunktion, die sofort eine Mail sendet, wenn zum Beispiel der Luftdruck in kurzer Zeit stark fällt, was bei einem anziehenden Gewitter der Fall ist.

Bis dann!