

Anleitung ESP32

Treiber installieren:

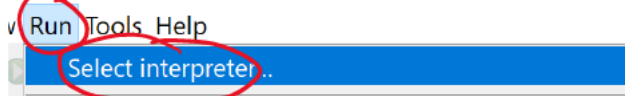
1. auf Moodle die Treiber herunterladen (CP210x_Universal_Windows_Driver.zip)
2. ZIP-Datei entpacken (Rechtsklick → Alle Extrahieren)
3. Im entpackten Ordner den Installer starten (z.B. die Datei CP210xVCPInstaller_x64.exe auf 64bit Windows)
Achtung: Wer NICHT Administrator auf dem PC ist: mit gedrückter [Shift]-Taste („Großmach“-Taste) einen Rechtsklick auf die .EXE-Datei → als Administrator ausführen
4. Setup durchführen
5. Nach dem Setup den ESP32 via USB anstecken und den Geräte Manager öffnen (im Startmenü nach Geräte-Manager suchen oder [Win]+[R] und dann „devmgmt.msc“ eingeben)
6. Im Geräte-Manager sollte in der Kategorie „Anschlüsse (COM & LPT)“ ein Silicon Labs Gerät erscheinen



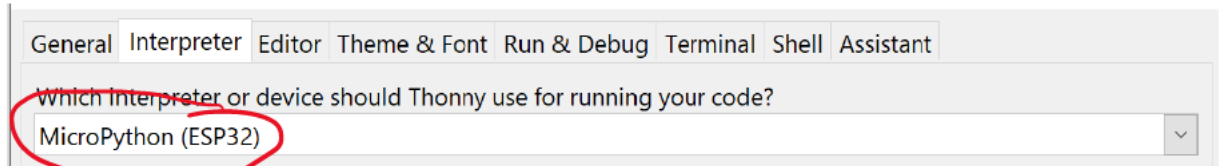
Den COM-Port (hier COM3) merken für später ...

MicroPython (über Thonny) auf dem ESP32 installieren:

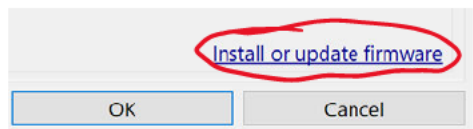
1. Thonny von der Webseite herunterladen (<https://thonny.org/>)
2. Das Installationsprogramm starten und installieren
3. Thonny öffnen
4. Interpreter-Einstellungen öffnen:



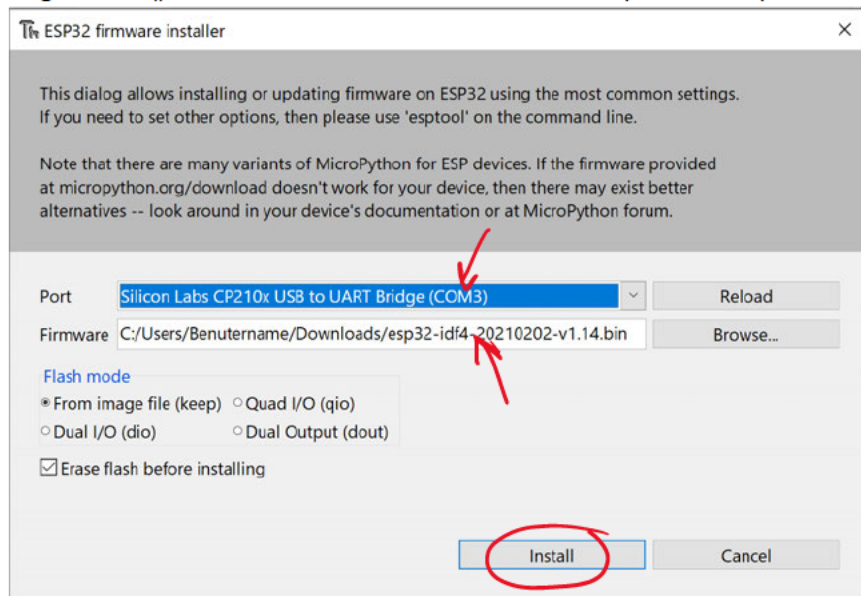
5. Dort MicroPython (ESP32) auswählen



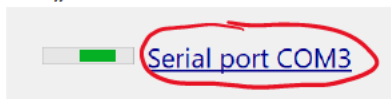
6. Falls der MicroPython Interpreter nicht am ESP läuft, muss es installiert/geflasht werden
(= wenn keine Verbindung hergestellt werden kann
... oder der ESP ist ganz neu
... oder es wurde vorher mit PlatformIO, Arduino o.ä. gearbeitet ...)
Dazu unten in der rechten unteren Ecke „Install or update firmware“ auswählen:



7. Es öffnet sich ein neues Dialogfenster; dort den entsprechenden Port auswählen (Name beginnt mit „Silicon Labs ...“ und das Firmware-File (*.bin Datei) suchen:

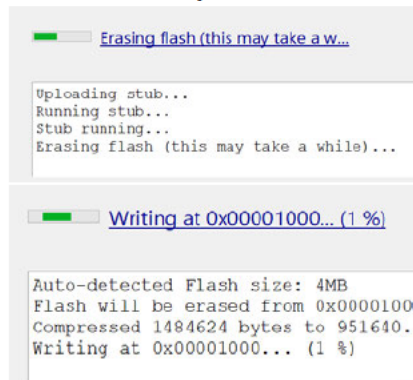


8. Auf „Install“ drücken → Es erscheint „Starting“, „Serial port ...“ usw.:

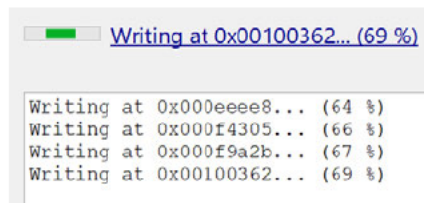


Auf die blaue Schrift drücken → es öffnet sich ein Textfeld

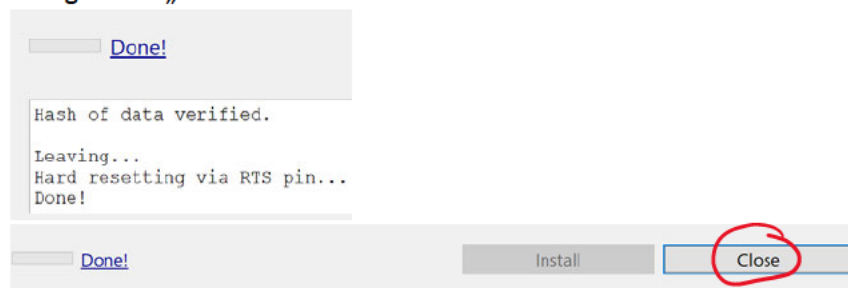
- a. Falls dort der Upload mit %-Zahlen kommt → alles ist OK



es dauert ...



Fertig → auf „Close“ drücken

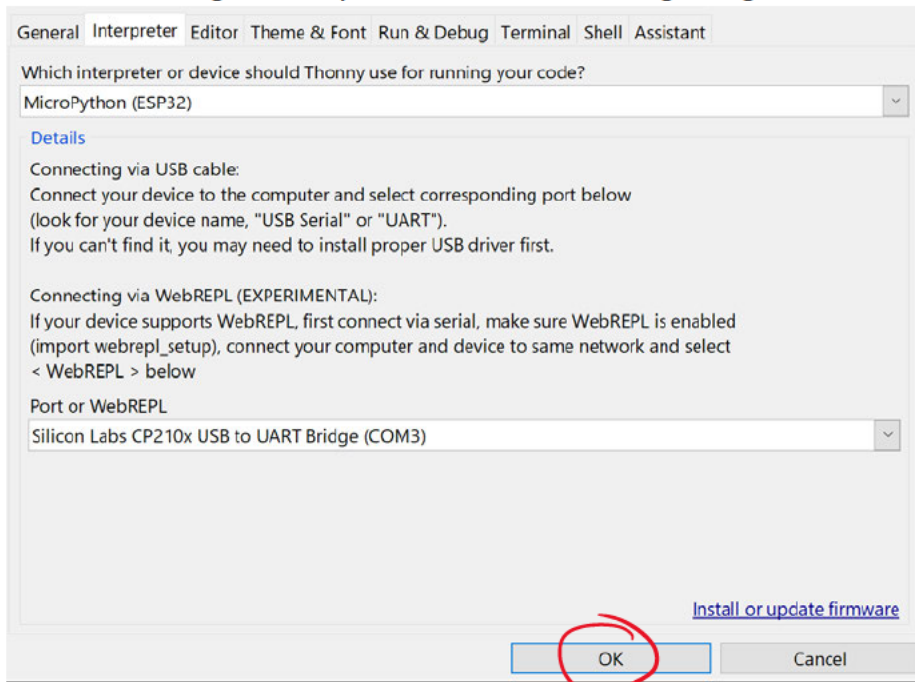


b. Falls dort nach einiger Zeit eine Fehlermeldung kommt:

```
A fatal error occurred: Failed to connect to Espressif device: Timed out waiting for
packet header
Erase command returned with error code 2
```

Den Upload einfach neu starten und nach dem drücken des „Install“-Buttons muss auf dem ESP32 der BOOT-Button (neben dem USB-Anschluss) für mehrere Sekunden gedrückt werden (etwa 2-3 Sekunden sollten idR reichen)

9. Nach einem erfolgreichen Upload sollten die Einstellungen ungefähr so aussehen:



auf OK drücken

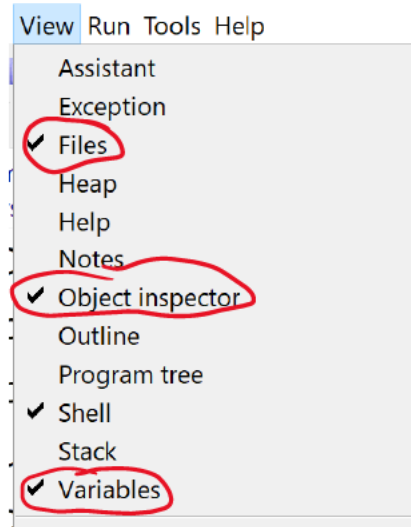
10. In Thonny gibt es unten die Konsole, dort sollte folgendes erscheinen:



Wenn kein Text / nur der rote Text da ist in der Menüleiste oben folgendes drücken:



11. Jetzt kann Thonny verwendet werden; damit man produktiv arbeiten kann wird empfohlen folgende Zusatzfenster zu aktivieren:



Wichtiger Thonny-Tipp:

Falls das Python-Programm „hängt“, beispielsweise in einer Schleife wo die ganze Zeit `input()` aufgerufen wird o.ä., kann man in der Shell unten einfach `[Strg]+[C]` drücken und beendet das ausgeführte Programm → man springt damit in den Interpreter-Mode zurück → es sollte `>>>` erscheinen.

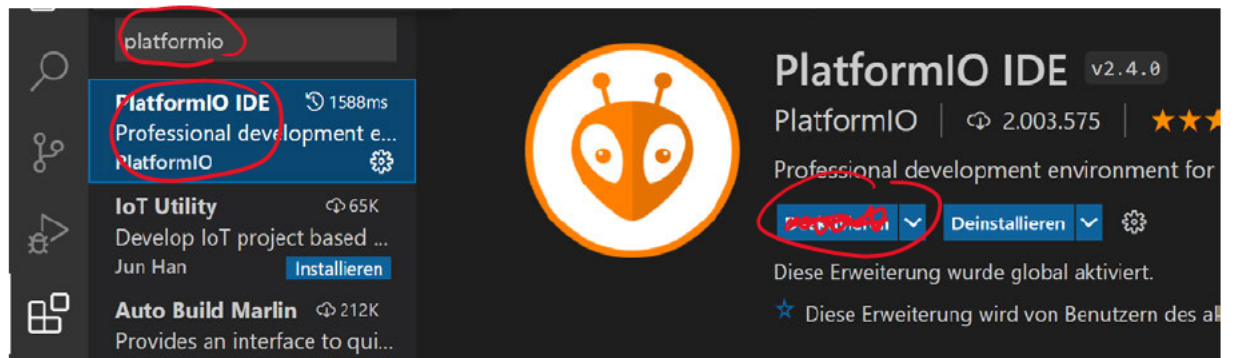
Tastenkombination	Verhalten
CTRL-A	enter raw REPL mode (no <code>>>></code> prompt, no echo)
CTRL-B	enter normal REPL mode
CTRL-C	interrupt a running program
CTRL-D	soft reset
CTRL-E	paste mode

PlatformIO in VSCode installieren:

1. VSCode sollte schon installiert sein, wenn nicht herunterladen und installieren!
2. In VSCode die Erweiterungen suchen:



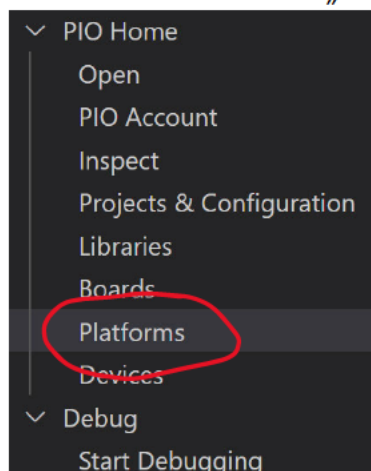
3. PlatformIO IDE suchen und installieren:



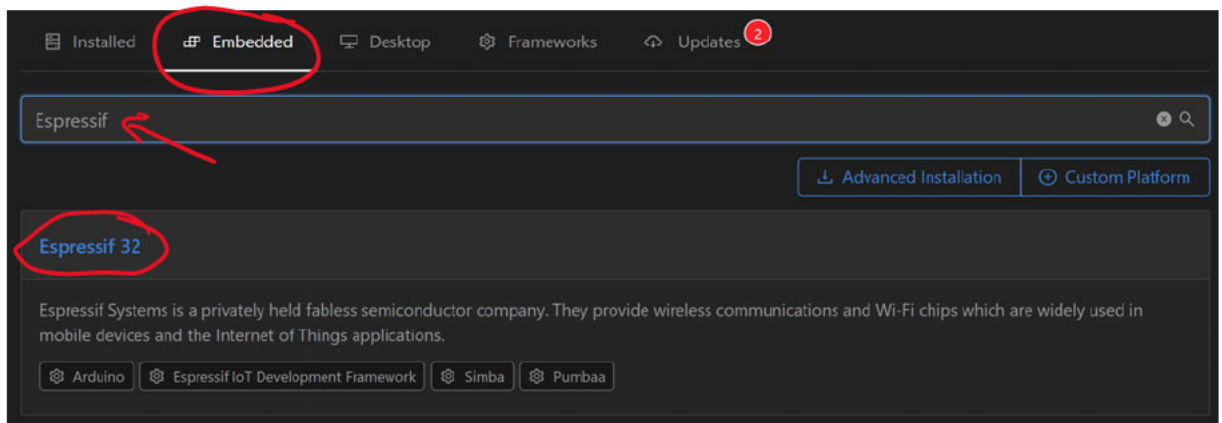
4. Nach der Installation erscheint ein „Alien“-Icon in der Seitenleiste:



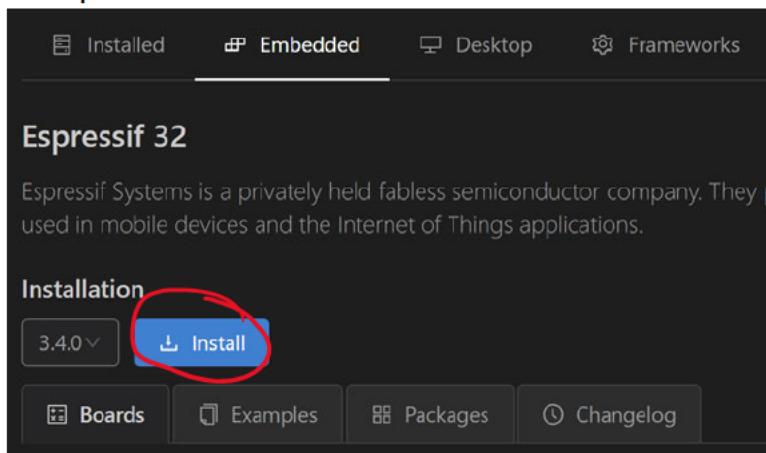
5. Dort klicken und im Menü „Platforms“ auswählen:



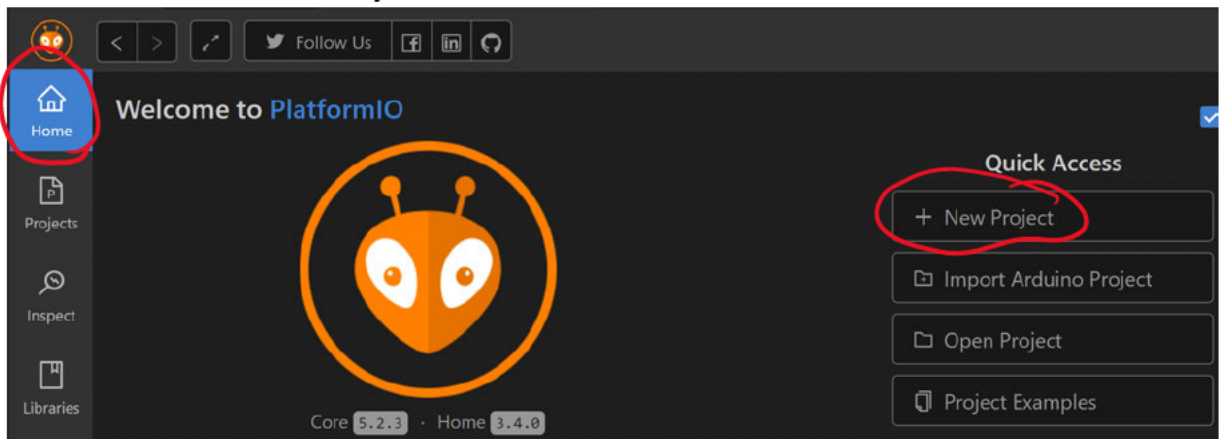
6. Im Dialog auf den Reiter „Embedded“ klicken, nach „Espressif“ suchen und auf „Espressif 32“ klicken:



7. Die Espressif 32 Plattform installieren:



8. Auf Home drücken → New Project



9. Im Dialogfenster den Projektnamen eingeben, als Board den NodeMCU-32S auswählen, Plattform Arduino und Finish drücken:

Project Wizard ✕

This wizard allows you to **create new** PlatformIO project or **update existing**. In the last case, you need to uncheck "Use default location" and specify path to existing project.

Name:

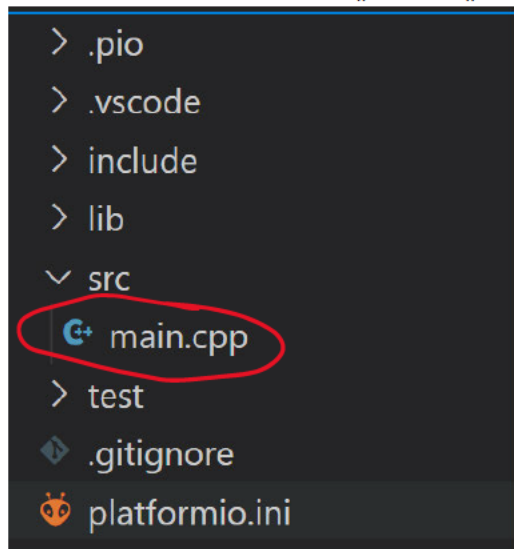
Board:

Framework:

Location: ☒ Use default location ?

Nach einiger Zeit ist alles abgeschlossen und wir landen im Code-Bereich ...

10. Dort befindet sich im Ordner „src“ die „main.cpp“:



In der main.cpp befinden sich die relevanten Funktionen:

```
src > main.cpp > setup()
1  #include <Arduino.h>
2
3  void setup() {
4      // put your setup code here, to run once:
5  }
6
7  void loop() {
8      // put your main code here, to run repeatedly:
9  }
```

→ setup wird 1x zu Beginn ausgeführt, danach springt das Programm in loop()

→ loop wird in einer Endlosschleife ausgeführt, hier kommt der eigentliche Programmcode rein

11. Codeauslagerung und die Verwendung von Bibliotheken (Libraries) in weiteren Dateien abseits von main.cpp ist erwünscht!
12. Happy Coding!

13. In der Statusleiste befinden sich mehrere Icons:



... Build = Programm kompilieren, auf Compilerfehler checken



... Upload = kompilieren und auf den ESP hochladen; falls der Upload nicht geht, muss auch hier wieder der Boot-Button neben dem USB-Anschluss gedrückt werden!



... Serial Monitor = öffnet die serielle Schnittstelle, damit kann der Computer mit dem ESP32 über USB (USB-Kabel) kommunizieren

Standardmäßig verwendet PlatformIO eine Geschwindigkeit bei der seriellen Kommunikation (baudrate) von 9600 baud; man erkennt das an:



drücken und dann öffnet sich ...

```
--- Available filters and text transformations: colorize, debug, default,
rintable, send_on_enter, time
--- More details at https://bit.ly/pio-monitor-filters
--- Miniterm on COM3 9600.8,N,1 ---
--- Quit: Ctrl+C | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H ---
```

Falls höhere Übertragungsraten benötigt werden:

platformio.ini öffnen und `monitor_speed = 115200` hinzufügen → Speichern



platformio.ini

```
10
11 [env:nodemcu-32s]
12 platform = espressif32
13 board = nodemcu-32s
14 framework = arduino
15 monitor_speed = 115200
```

Jetzt sollte die Geschwindigkeit höher sein:

```
--- Available filters and text transformations: colorize, debug, default,
rintable, send_on_enter, time
--- More details at https://bit.ly/pio-monitor-filters
--- Miniterm on COM3 115200.8,N,1 ---
--- Quit: Ctrl+C | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H ---
```


Weitere Softwarealternativen

Pymakr in VS Code:

Mit Pymakr kann man MicroPython in VS Code programmieren. Eine Anleitung ist unter folgendem Link zu finden:

<https://alepycom.gitbooks.io/pycom-documentation/content/chapter/pymakr/installation/vscode.html>

Arduino IDE mit ESP32:

Der ESP32 kann auch über die Arduino IDE (C/C++) programmiert werden. Eine Anleitung ist unter folgendem Link zu finden:

<https://espressif-docs.readthedocs-hosted.com/projects/arduino-esp32/en/latest/installing.html>