Wasserbox V2 - Flashen

1. Visual Studio Code (VS Code)

Um für den ESP32 geschriebene Programme hochzuladen (zu kompilieren und flashen) wird Visual Studio Code verwendet.
Das IDE (Integrated Development Environment) von Arduino würde sich ebenfalls anbieten. In der neuen Version 2 fehlt jedoch bisher der SPIFFS-Upload für den ESP32, so dass es für uns nutzlos ist. Aus unbekannten Gründen bereitet auch die alte Arduino-Version V1.8.3 im Zusammenhang mit der WebServer-Bibliothek Probleme.

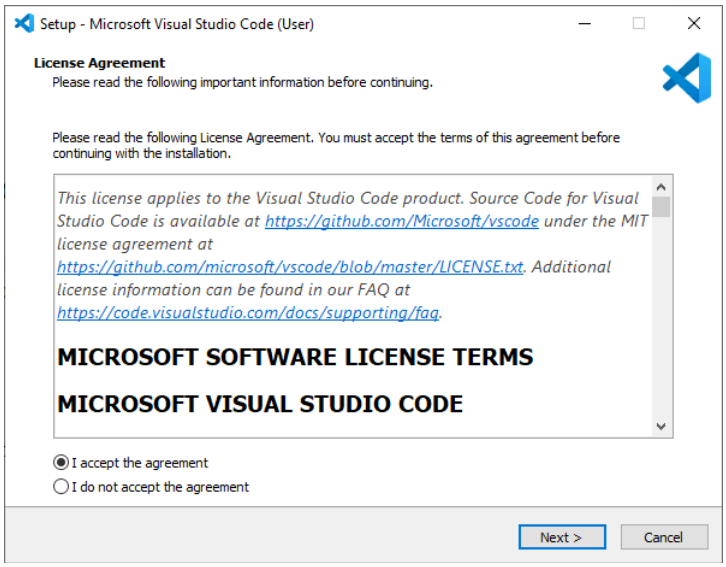
VS Code ist einfach zu installieren und kompiliert zudem um vieles schneller als die Arduino IDE.

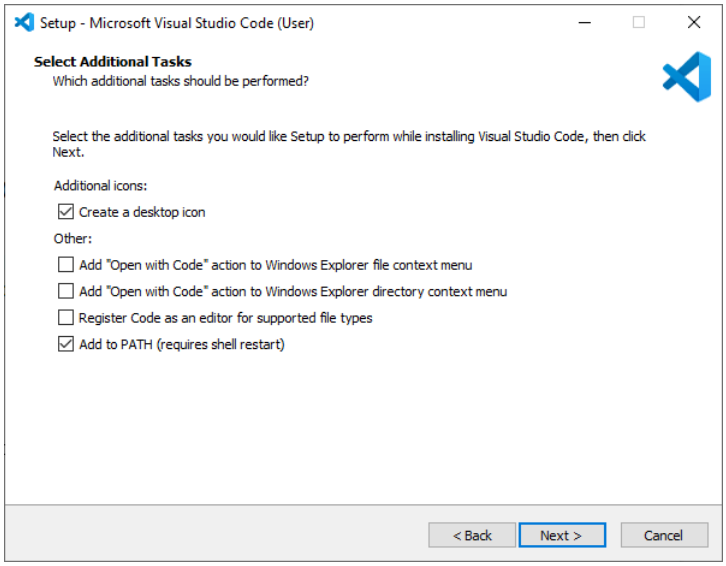
Eine Beschreibung zur Installation von VS Code findet sich hier:
https://randomnerdtutorials.com/vs-code-platformio-ide-esp32-esp8266-arduino/#2

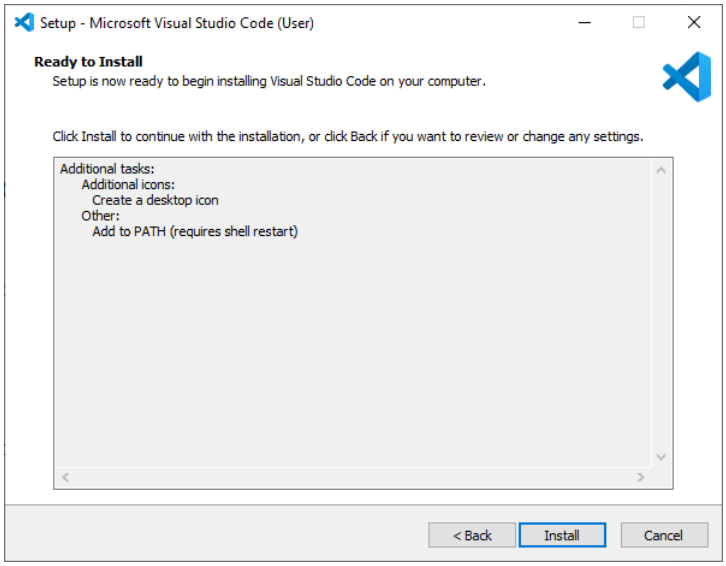
2. Kurzanleitung zur Installation von Visual Studio Code (Windows)

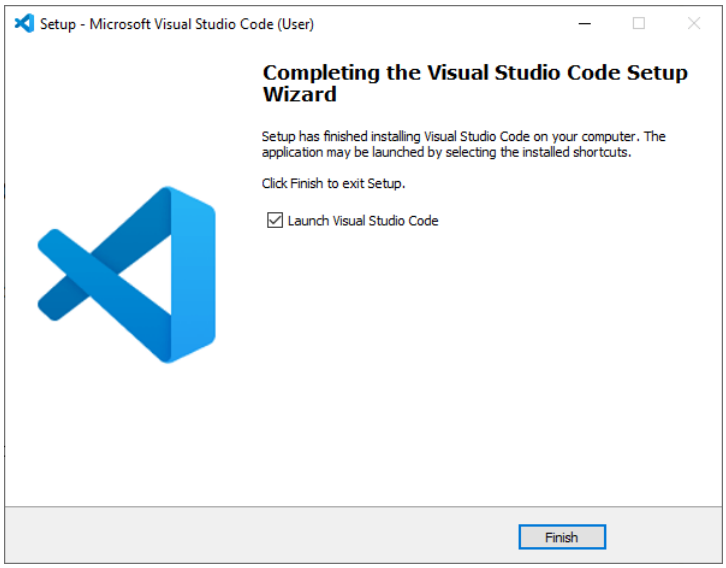
Die Software fur VS Code kann hier heruntergeladen werden:
https://code.visualstudio.com/

Mit dem Wizard sind folgende Schritte zu durchlaufen:

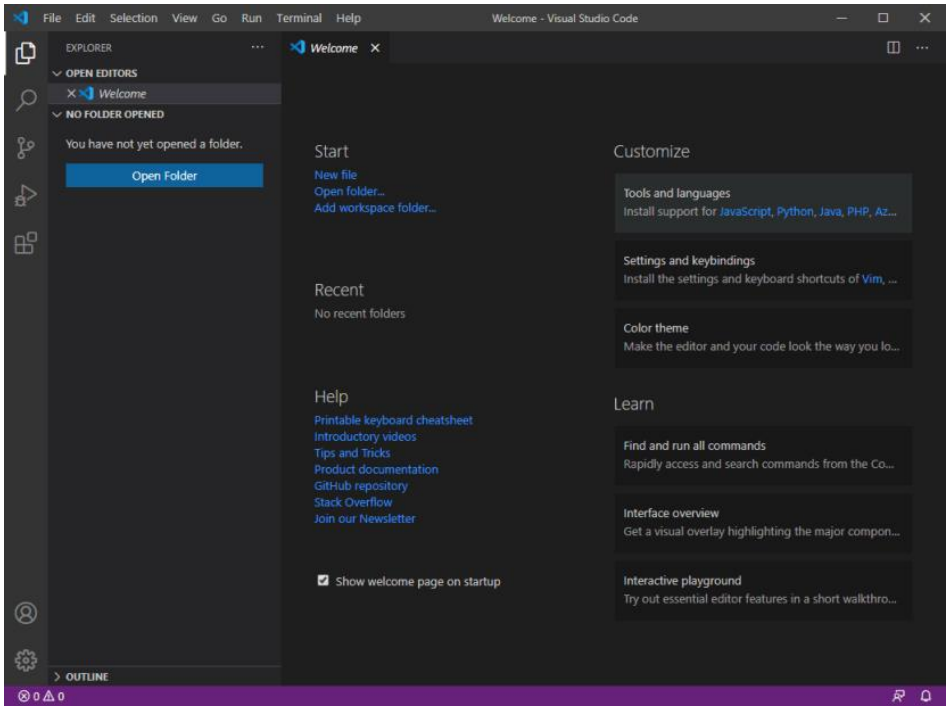






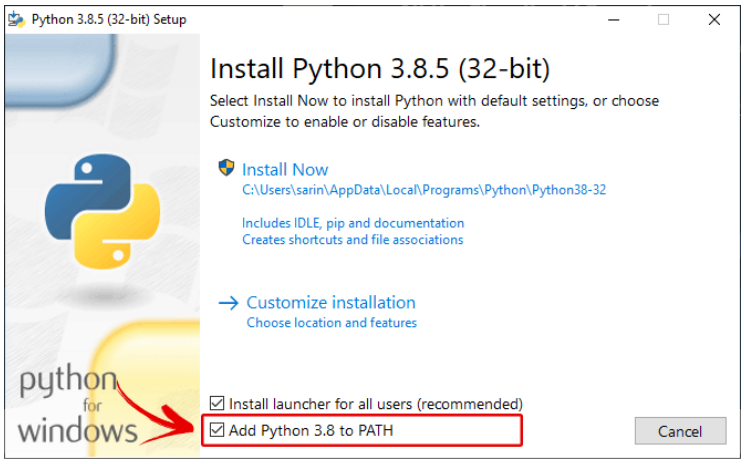


Nach der Installation zeigt VS Code beim Starten folgendes Fenster:

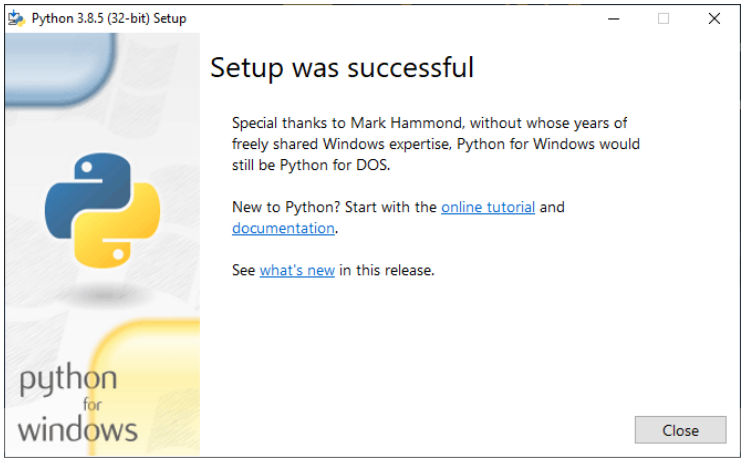


3. Installation von Python

Zur Programmierung des ESP32 mit der PlatformIO IDE mussPython 3.5 oder höher installiert werden.
Python kann hier heruntergeladen werden: https://python.org/download
Folgende Schritte sind bei der Installation durchzuführen:

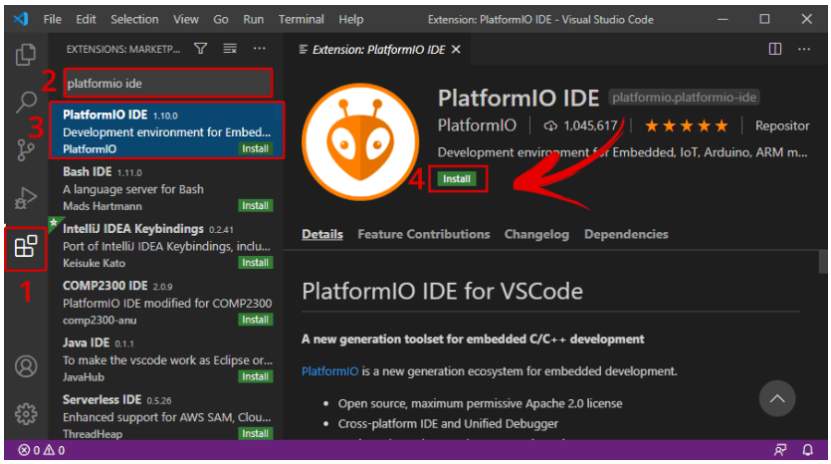


Pfad hinzufügen!



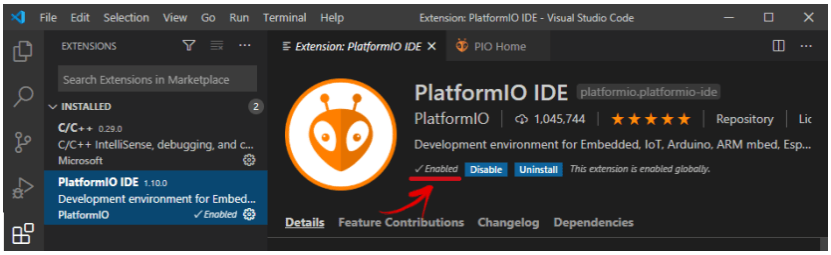
4. Installation der PlatformIO IDE Erweiterung in VS Code

VS Code soll mit der Platform IDE erweitert werden. Dazu sind nach Öffnen von VS Code folgende Schritte nötig:

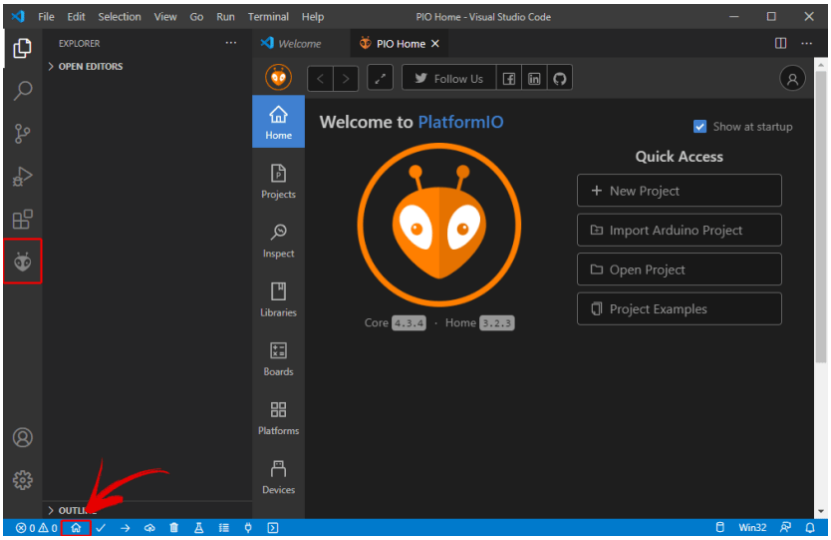


1. Extensions anklicken
2. 'PlatformIO IDE' suchen
3. Angezeigte Option wählen
4. Auf Install-Schaltfläche klicken. Die Installation dauert eine Weile.

Nach der Installation muss Platform IDE enabled sein.



Jetzt muss das Icon PlatformIO links sichtbar sein sowie das Symbol Home unten angezeigt werden.



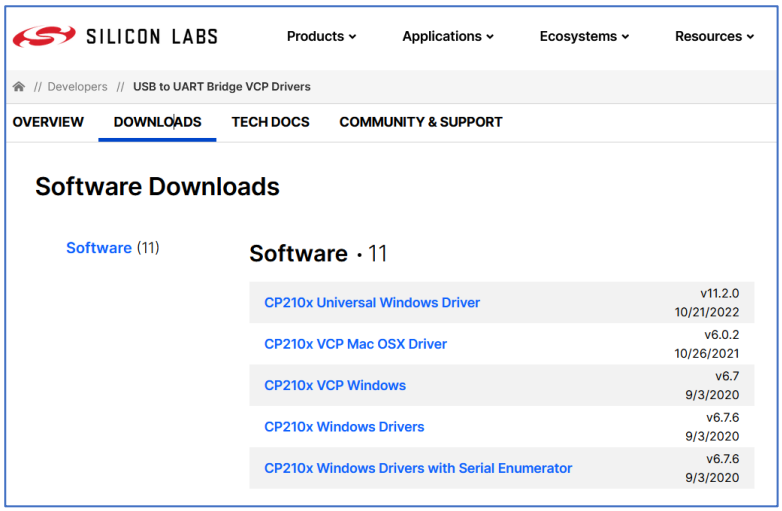
Vorzugsweise wird jetzt VS Code nochmals neu gestartet.

5. Installation des USB-Treibers für das Adafruit ESP32 Feather Board

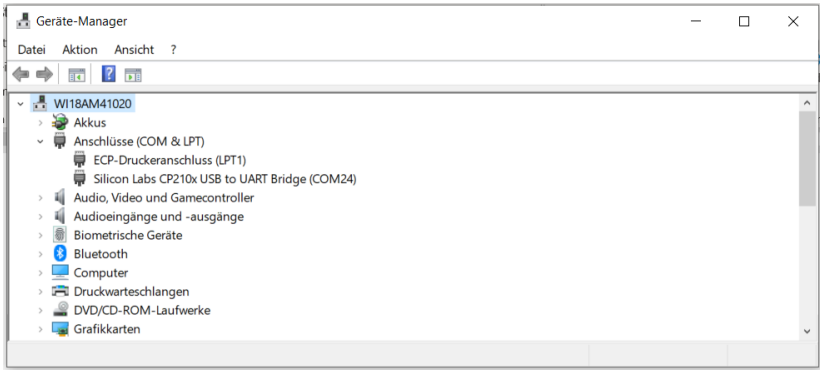
Das verwendete Board von Adafruit verwendet einen Kommunikations-Chip von SiLabs, welcher einen speziellen USB-Treiber benötigt. Die Beschreibung von Adafruit:
https://learn.adafruit.com/adafruit-huzzah32-esp32-feather/using-with-arduino-ide

Hier wird in Kürze die Installation beschrieben.

Der Treiber kann hier heruntergeladen werden:
http://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers



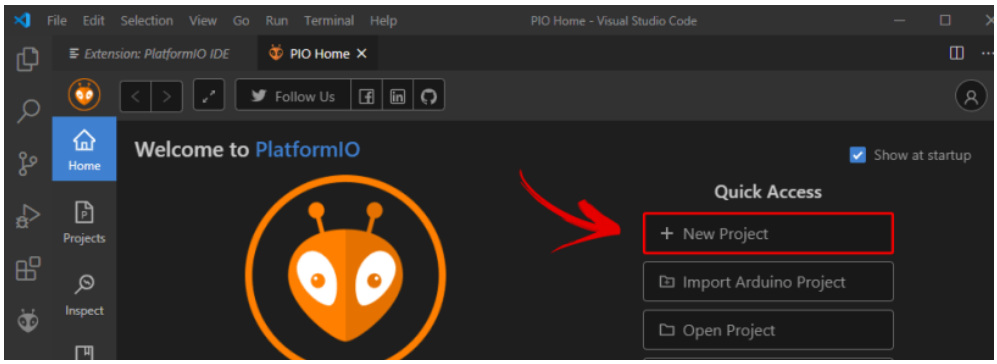
Gewählt wird 'CP210x VCP Windows'. Nach Download wird an einen gewünschten Ort entpackt und schliesslich die Datei 'CP210xVCPInstaller_x64.exe' ausgeführt.
Nach erfolgreicher Installation kann ein ESP32 an die USB-Schnittstelle angeschlossen werden. Der Geräte-Manager zeigt dann den ESP32 wie folgt an:



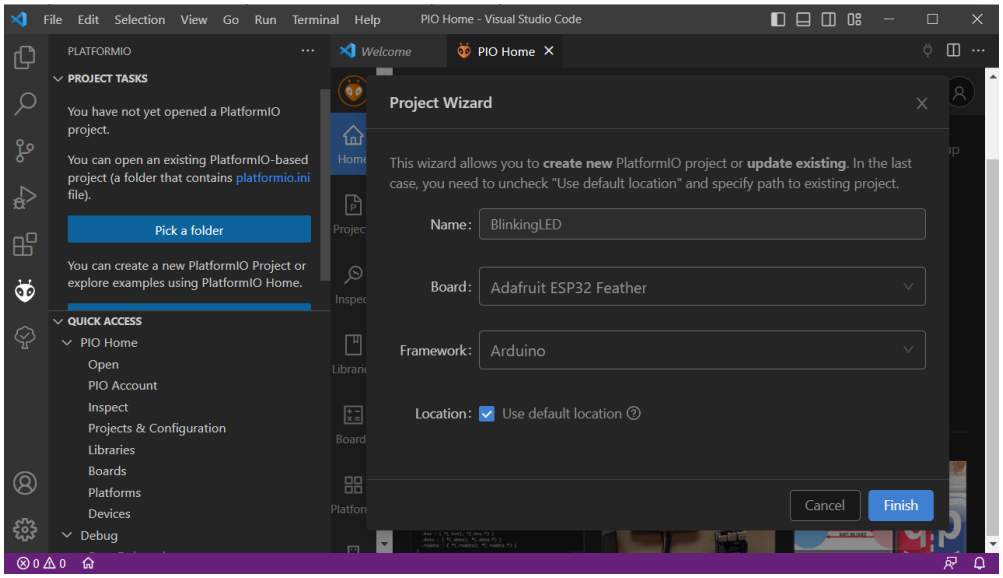
6. Funktionstest

PC und esp32 mit einem USB-Kabel verbinden. Auf dem ESP32-Board blinkt eine orange LED, der PC hat ein Device am USB-Port erkannt (Signalton).

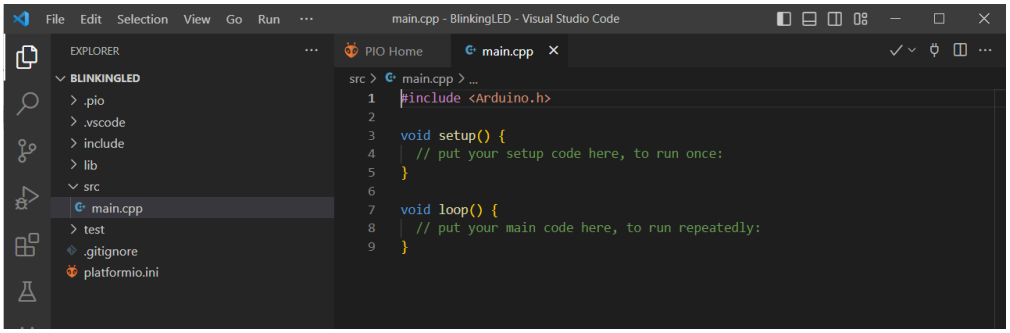
Untenstehend wird ein neues Projekt angelegt, um das Programmieren zu testen.
In VS Code auf PlatformIO Home (allenfalls ganz unten auf der Statuszeile) klicken, danach auf +New Project.



Im Project Wizard 'BlinkingLED' eingeben und das Board 'Adafruit ESP32 Feather' wählen, danach 'Finish' klicken.

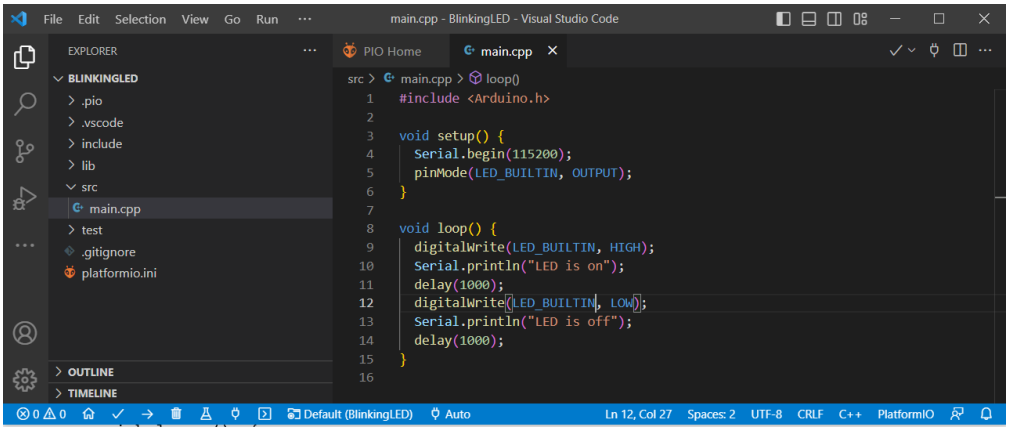


Dem (eigenen) Code vertrauen, 'src' öffnen und auf 'main.cpp' doppelklicken.

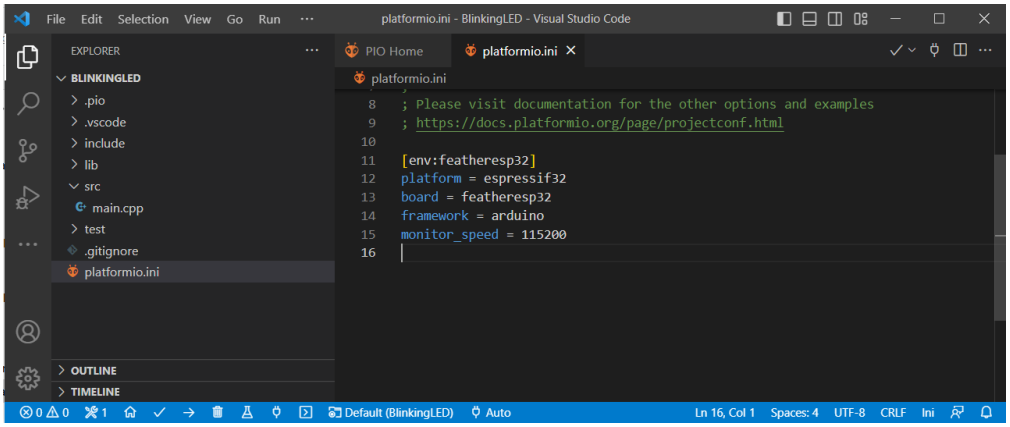


Folgenden Code ins Editorfenster 'main.cpp' kopieren:

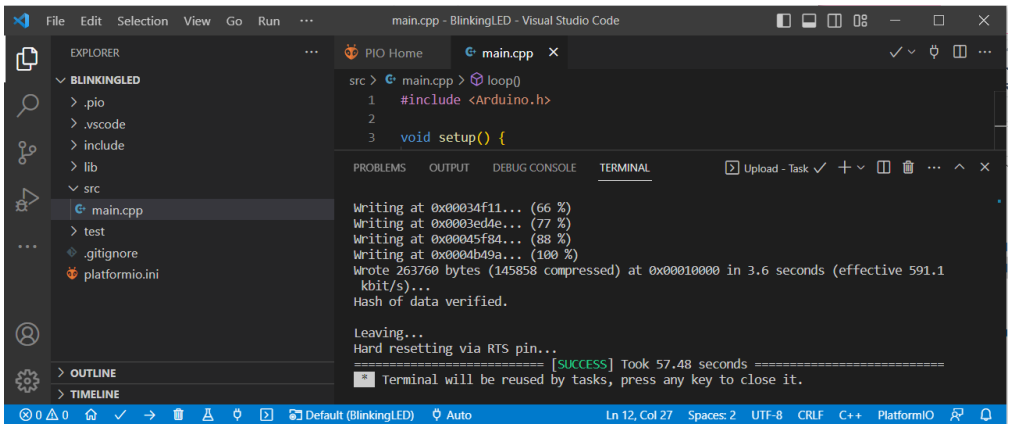
```
#include <Arduino.h>

void setup() {
  Serial.begin(115200);
  pinMode(LED, OUTPUT);
}

void loop() {
  digitalWrite(LED, HIGH);
  Serial.println("LED is on");
  delay(1000);
  digitalWrite(LED, LOW);
  Serial.println("LED is off");
  delay(1000);
}
```



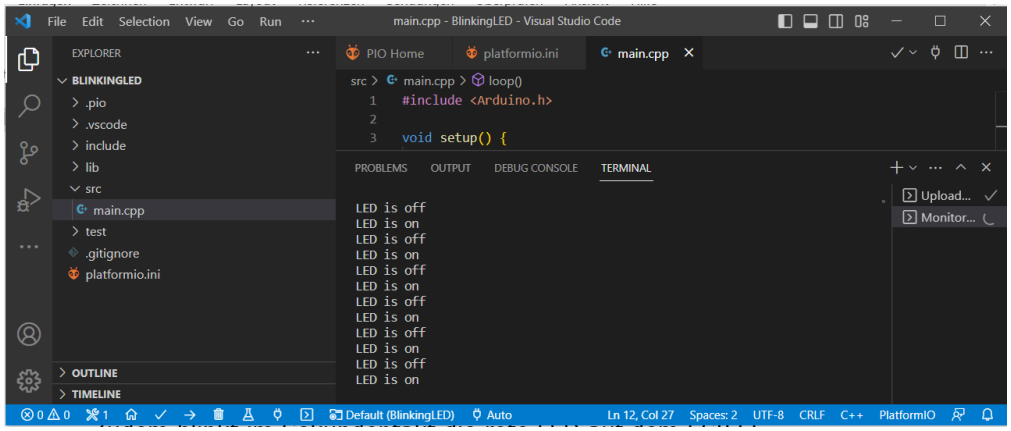
Damit die Baudrate des Monitors richtig gesetzt wird, links auf 'platformio.ini' doppelklicken und auf Zeile 15 'monitor_speed = 115200' eintragen.



In der Statuszeile auf den 'Pfeil nach rechts' drücken, um zu kompilieren und das Kompilat in den ESP32 zu flashen. Nach erfolgreichen Operationen sieht das Fenster TERMINAL wie folgt aus:

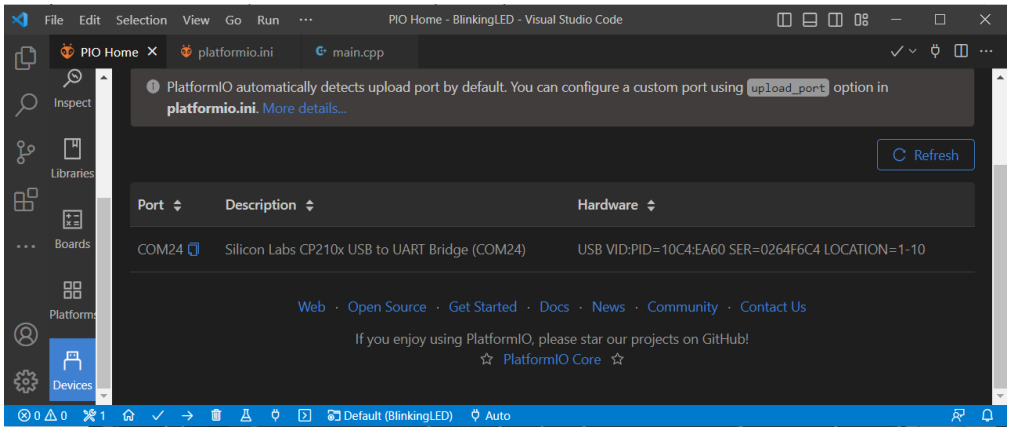


Zudem blinkt im Sekundentakt die rote LED auf dem ESP32.
Mit Klicken auf das Stecker-Symbol auf der Statuszeile wird der Serial Monitor aktiviert:



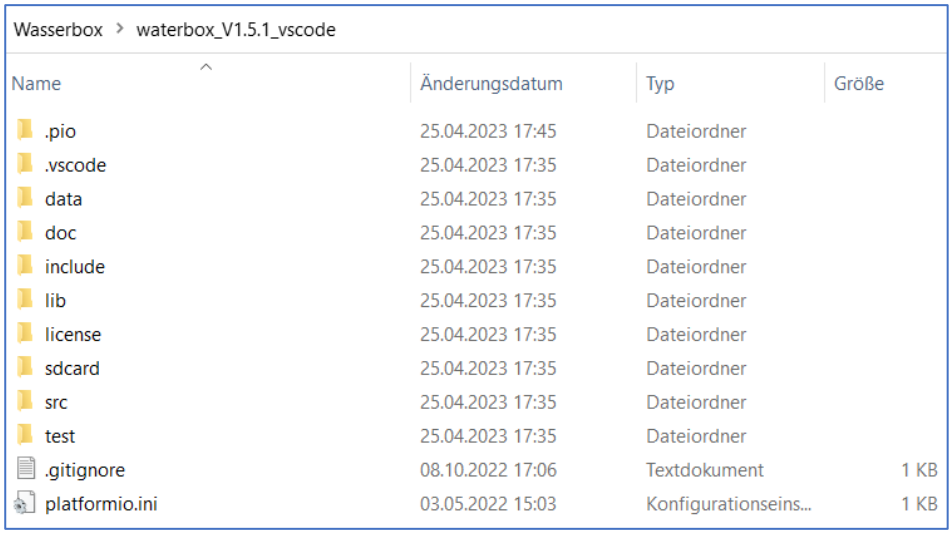
Die Ausgabe an den Serial Monitor wird im TERMINAL sichtbar.
Übrigens: VS Code detektiert den USB-Anschluss automatisch (und korrekt, solange nicht mehrere USB-Anschlüsse besetzt sind).

Allenfalls nützlich zu wissen:
Bei Wahl des Reiters 'PIO Home' und Klicken auf 'Devices' kann der gefundene USB-Port ersehen werden:



7. Projekt Wasserbox anlegen

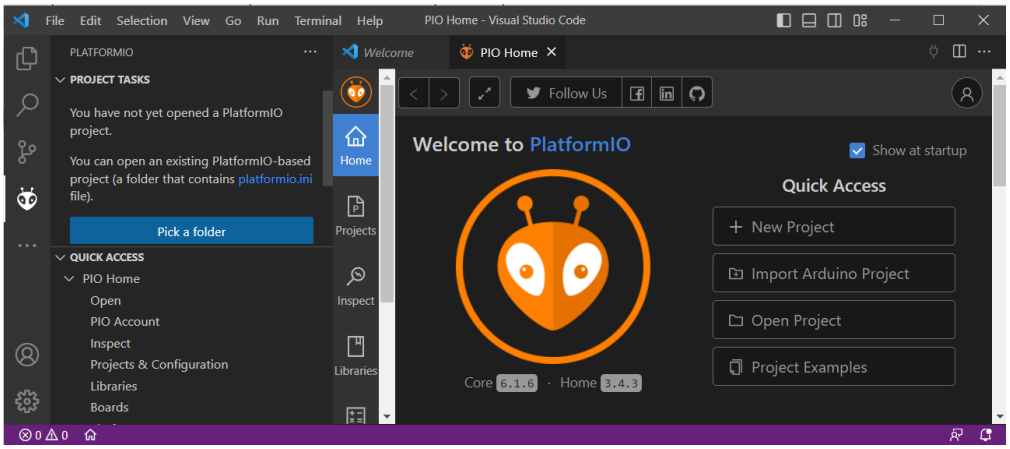
An geeigneter Stelle (frei wählbar) legen wir auf dem PC den Ordner 'Wasserbox' an. In diesen Ordner wird via switchdrive die heruntergeladene Datei 'waterbox_Vx.x.x_vscode.zip ' entpackt, d.h. der Ordner 'waterbox_Vx.x.x_vscode'.
In diesem Ordner befinden sich das gesamte Projekt Wasserbox.
Beispiel:



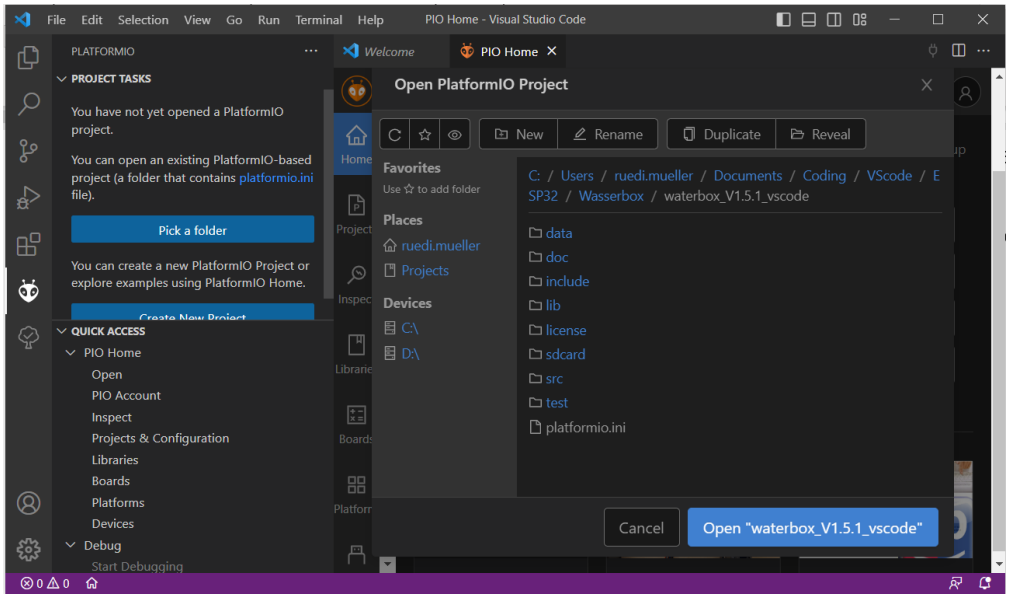
8. Projekt Wasserbox öffnen

VS Code starten. Sollte noch ein altes Projekt angezeigt werden, kann dieses mit 'File>Close Folder' entladen werden.

Links auf Insektensymbol klicken, dann in Statuszeile auf Home-Symbol drücken.



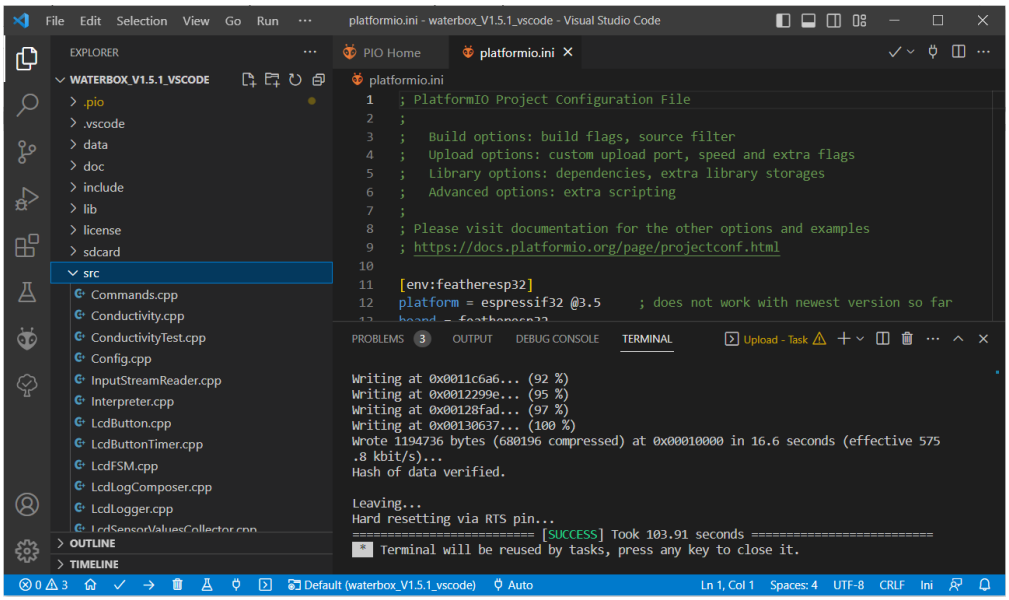
Jetzt 'Open Project' betätigen und den Pfad zu obigem Projekt suchen.



'Open "waterbox_V1.5.1_vscode"' aktivieren und vertrauen.

9. Programmcode kompilieren und flashen

'Pfeil-nach-rechts' in Statuszeile drücken. um zu kompilieren und das Kompilat zu flashen.

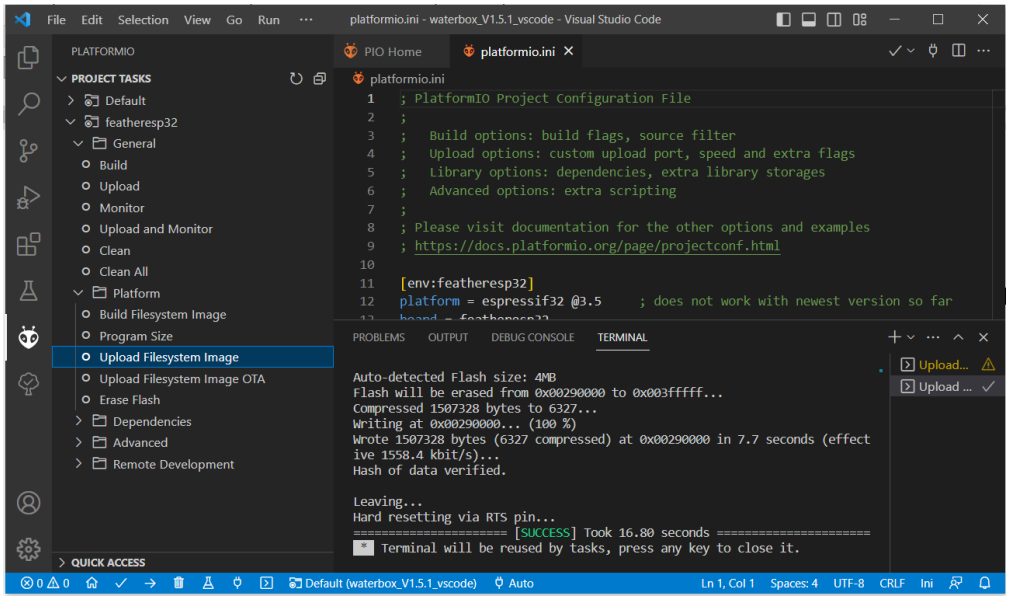


Damit ist der ESP32 mit dem Programm geladen.

10. SPIFFS flashen

Der ESP32 bietet die Möglichkeit, im Flash-Memory ein Filesystem zu erstellen und zu verwalten. Die Wasserbox macht davon Gebrauch. Daher muss auch das SPIFFS hochgeladen werden.

Links auf Insektensymbol klicken, dann 'Upload Filesystem Image' aktivieren.

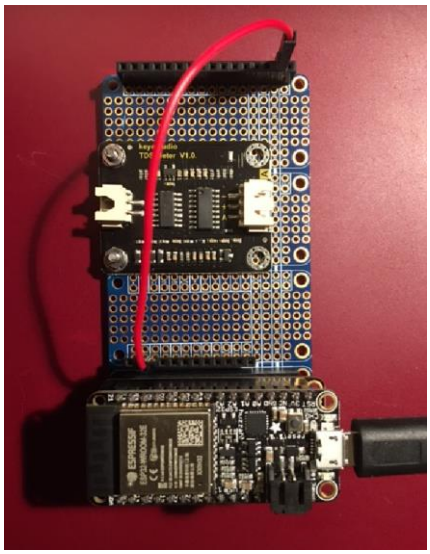


Mit der Erfolgsmeldung ist nun auch das SPIFFS im ESP32 hochgeladen.

11. Default-Werte für die Sensorkalibrierung aktivieren (ab V2.1.0)

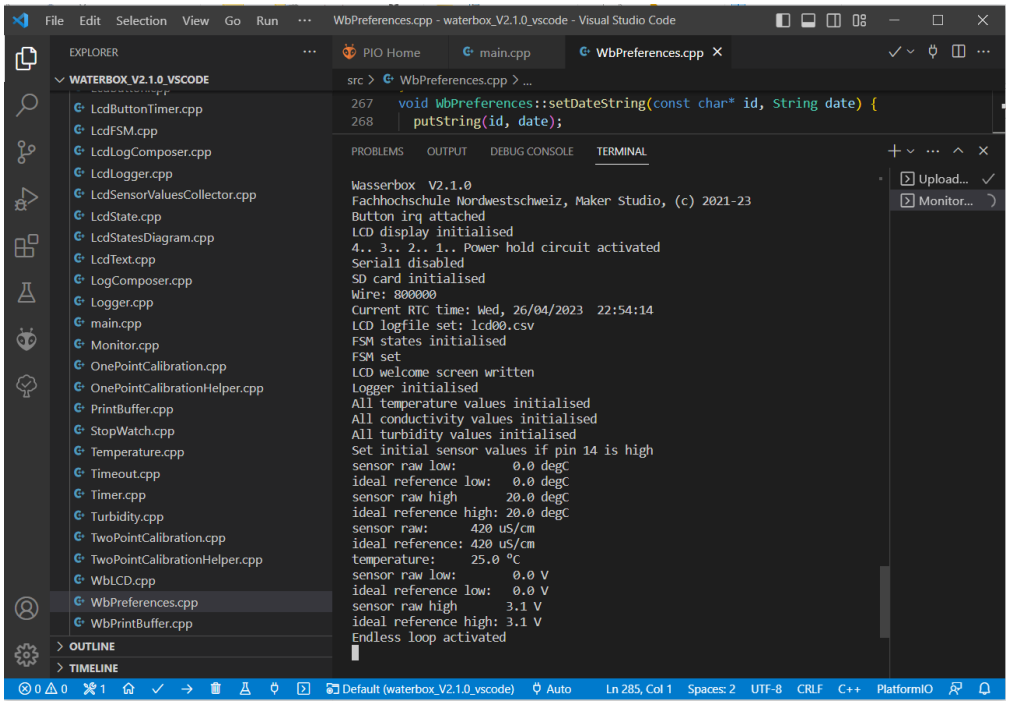
Die Sensoren der Wasserbox können via Befehle kalibriert werden. Um eine erste Annäherung an die Kalibrierung schnell vorzunehmen, kann der Eingang Pin 14 des ESP32 nach Flashen (gemäss 9.) hochgesetzt werden. Nach einem Reset werden dann entsprechende Werte der Präferenzen 'prefs' (vgl. Wasserbox-Befehle) der Sensoren gesetzt. Das Setzen wird auch am Serial Monitor gemeldet.

Vorgehen:
Geflashten ESP32 mit adalogger Breakout in eine Grundplatine einstecken.
Pin 14 mit 3.3 V verbinden: Verbindung von zweiter Buchse rechts der oberen Leiste zu dritter Buchse links der unteren Leiste vornehmen.



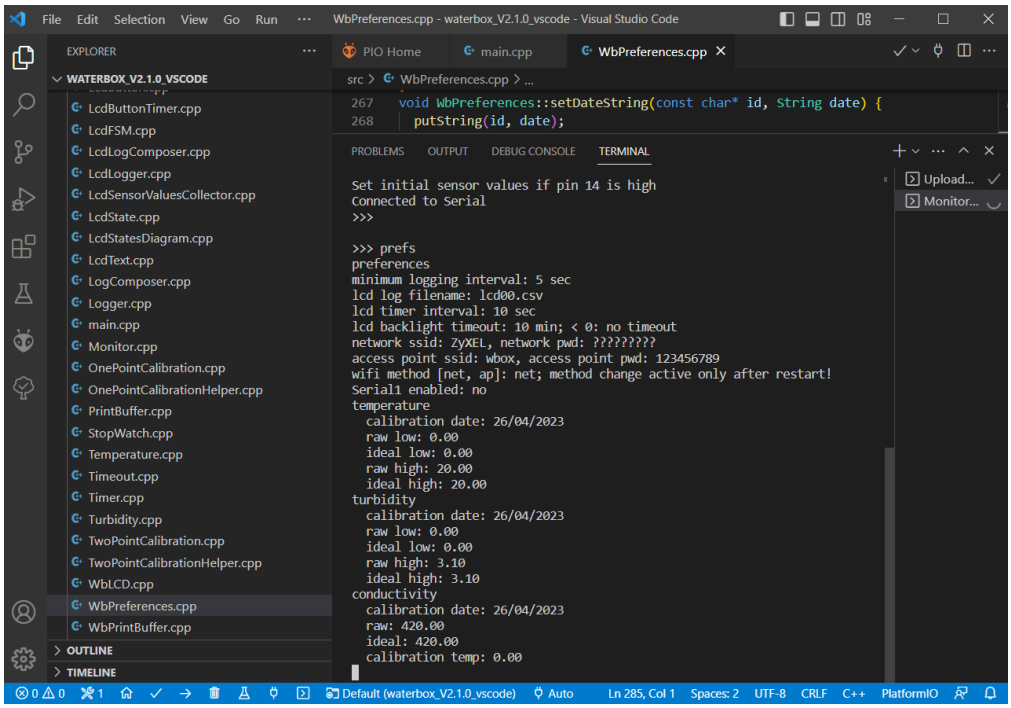
ESP32 an USB-Anschluss des PC bei aktivem VS Code einstecken.

Im TERMINAL von VS Code erscheint folgenden Ausgabe:



Kabel entfernen.

Eine zweite Kontrolle ist wie folgt möglich:
Kabel muss entfernt sein. Reset am ESP32 vornehmen und in TERMINAL klicken. 'prefs' und Return eingeben.



Die Werte sind gespeichert, wie ersehen werden kann.

12. Programm-Kurztest

ESP32 inkl. adalogger Breakout in die Wasserbox-Hardware eingestecken.
Programmier-USB-Kabel lösen und Akku an Grundplatine verbinden. Mit Drücken der Bedientaste soll das LCD-Display aufleuchten und entsprechende Meldungen darstellen. Damit ist zusätzlich sichergestellt, dass der Programm-Upload funktioniert hat.

13. Händisches Definieren der Parameter

a. Einrichten des Access Points
   >>> setap
   wifi method [net, ap]: ap; method change active only after restart!
   Dann Reset

b. Setzen von Datum und Zeit
   >>> setdate i4i4i2022
   Mon, 4/04/2022 21:08:26

   >>> settime i21i9i50
   Mon, 4/04/2022 21:09:50

c. Setzen der Kalibrierwerte für den Temperatursensor
   Eiswasser
   >>> settempcaliblow f0
   Wasser mit Umgebungstemperatur. An Referenzthermometer Wert ablesen und eingeben (z.B. 22.4°C)
   >>> settempcalibhigh f22.4

d. Setzen der Kalibrierwerte für den Turbidity-Sensor
   Sichttiefe messen mit der Secchi-Scheibe
   // TODO
   >>> setturbiditycaliblow f   in cm
   // TODO
   >>> setturbiditycalibhigh f   in cm

e. Setzen des Kalibrierwertes für den Conductivity-Sensor
   Referenzwert von Kalibrierflüssigkeit (bei 25°C) bei eingetauchtem Sensor übertragen (z.B. 200µS/cm).
   >>> setconductivitycalib f200

f. Setzen des Namens der LCD-Datei, falls Defaultname nicht erwünscht
   >>> setlcdfilename "aabach.csv"

g. Setzen des Logging Intervalls im LCD Display Mode, z.B 30 Minuten
   >>> setlcdloggerinterval i30i0

h. Setzen der LCD-Display-Hintergrundbeleuchtungsdauer, z.B. 3 Minuten
   >>> setbacklighttimeout i3

i. Überblick der eingestellten Werte:
   >>> prefs