

Wasserbox ESP32 V2 - Expert-Modus

1. Einführung

Die Wasserbox baut auf dem Adafruit HUZZAH32 – ESP32 Feather

(<https://learn.adafruit.com/adafruit-huzzah32-esp32-feather>) auf. Mit verwendet wird ein Adafruit Adalogger FeatherWing (<https://learn.adafruit.com/adafruit-adalogger-featherwing>) mit Uhr (RTC) und SD-Card-Holder.

Der ESP32 ist WLAN-fähig, was in der Wasserbox insbesondere für das Downloading von Datenfiles von der SD-Card genutzt wird.

2. Expert-Modus

Der Expert-Modus ist für erweiterten Einsatz der Wasserbox als Logger für Temperatur, Wasserleitfähigkeit und Wassertrübung gedacht. Insbesondere erlaubt er, Parameter zu setzen und zu ändern, was im Bedientastenbetrieb (auch 'LCD Display Mode' genannt) nicht möglich ist. Daher ist der Expert-Modus auch notwendig, um die Wasserbox vor dem ersten Einsatz zu kalibrieren.

Im Expert-Modus kann auf zwei Arten gearbeitet werden: Entweder via USB-Kabel zwischen ESP32 und PC oder via WLAN.

Da für das WLAN auf Seite des ESP32 die SSID und das Passwort gesetzt werden müssen, kommt man allerdings im allerersten Schritt nicht um eine Kabelverbindung herum. Nach Setzen von SSID und Passwort steht der Wahl der Verbindungsart nichts mehr entgegen.

3. Expert-Modus im Kabelbetrieb

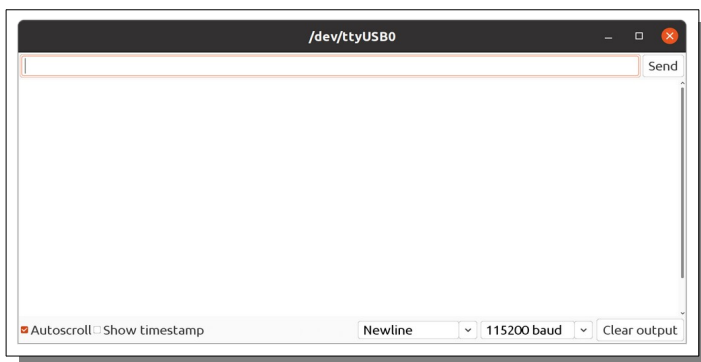
Bei Verwendung eines USB-Kabels zwischen ESP32 und PC ist darauf zu achten, dass ein allfälliges Kabel zwischen Wasserbox und Batterie/Akku entfernt ist.

Nach Aufstarten und Erscheinen des Begrüßungstextes am LCD Display – die Bedientaste muss entgegen dem ersten Text nicht gedrückt werden – ist die Wasserbox für die Kommunikation im Expert-Modus bereit.

Hinweis:

Sollte die Hintergrundbeleuchtung erlöschen, reicht es, die Bedientaste zu berühren (nicht zu drücken!).

Es wird empfohlen, den Serial Monitor der Arduino-IDE zu verwenden. Dabei reicht es, vorerst im Menü 'Tool' den Port zu wählen, der vom ESP32 verwendet wird. Mit Ctrl+Shift+M wird der Serial Monitor aus dem Arduino-IDE gestartet. Unten rechts im Monitorfenster ist die Baudrate auf '115200' einzustellen.



Mit Klicken in die Zeile links neben der Send-Taste kann jetzt 'help' eingegeben werden:



Return oder Betätigen der Send-Schaltfläche schickt den Befehl ab. Darauf muss die Wasserbox mit einer Befehlsübersicht antworten. Vgl unten bei 4.

Zudem erscheint auf dem LCD-Display die Meldung, dass die Bedientaste nicht mehr aktiv ist.

Im folgenden werden die Expert-Modus-Befehle besprochen.

4. Bedienung

Wir beginnen mit Eingabe von **'help'** und Drücken von **'send'**. Gross-/Kleinschreibung wird unterschieden.

Die Hilfebefehle (in Gruppen unterteilt) werden an den Monitorbildschirm ausgegeben:

```
=== HELP ===
help ..... show help commands
helpsensors .... show sensor commands
helpdatetime ... show date and time commands
helplogfile .... show logfile commands
helplogger ..... show logger commands
helpmonitor .... show monitoring commands
helplcdmode .... show lcd display mode commands
helpmisc ..... show miscellaneous commands
helpdevel ..... show developer commands
```

4.1. helpsensors

Mit Eingabe von **'helpsensors'** werden weitere Befehle angezeigt. Dies gilt für alle Befehle, die mit **'help'** beginnen (Mehr dazu in den folgenden Unterkapiteln).

```
=== SENSORS ===
temp ..... show the current temperature
tempcalib ..... show temperature calibration values
settempcaliblow f<ideal> ..... set ideal temp calibration low value
settempcalibhigh f<ideal> ..... set ideal temp calibration high value
conductivity ..... show the current electrical conductivity
conductivitycalib ..... show conductivity calibration value
setconductivitycalib f<ideal> .... set ideal conductivity calibration value
turbidity ..... show the current turbidity
turbiditycalib ..... show turbidity calibration values
setturbiditycaliblow f<ideal> .... set ideal turbidity calibration low value
setturbiditycalibhigh f<ideal> ... set ideal turbidity calibration high value
```

'temp' zeigt die momentan gemessene, kalibrierte Temperatur am Bildschirm. Diese wird auf ein Zehntel Grad gerundet dargestellt.

'tempcalib' zeigt vier pausable Temperaturwerte, die nicht-flüchtig im ESP32 gespeichert sind, vorausgesetzt, es wurde vorgängig kalibriert. Vgl. weiter unten.

Der Temperatursensor - als **'sensor'** bezeichnet - zeigt hier 0.4°C und 29.3°C an. Diese Werte wurden mit nachfolgend beschriebenen Befehlen gespeichert. Der untere Wert (raw low) wurde gemessen, als beim idealen Referenzthermometer der Wert 0.1°C (ideal reference low) abgelesen und eingegeben wurde.

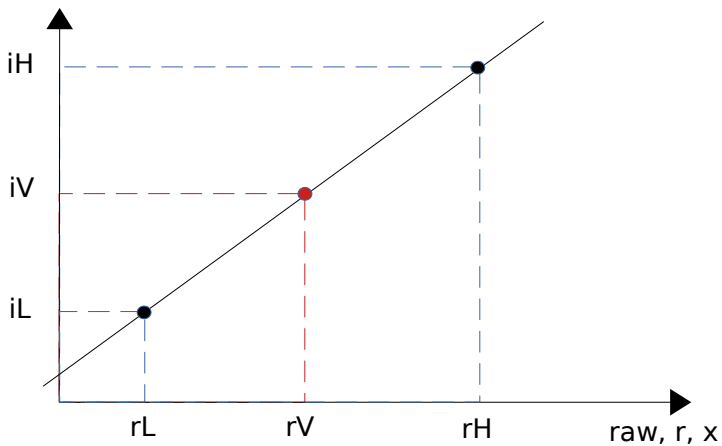
Der obere Werte (sensor raw high) von 29.3°C wurde gespeichert, als beim idealen Thermometer der Wert (ideal reference high) 30.1°C abgelesen und eingegeben wurde.

```
temp calibration, last changed: 3/03/2021
sensor raw low:      0.4°C
ideal reference low: 0.1°C
sensor raw high     29.3°C
ideal reference high: 30.1°C
```

Mit diesen Angaben wird eine Zweipunkt-Kalibrierung des Temperatursensors vorgenommen.

Allgemein:
ideal: Referenzsensor
raw: Arduinosensor

ideal, i, y



$$y = c0 + m * (x - c1)$$

$$y = i, x = r$$

$$c0 = \text{offset} = iL$$

$$c1 = rL$$

$$m = \text{slope} = (iH - iL) / (rH - rL)$$

eingesetzt:

$$i = iL + \frac{iH - iL}{rH - rL} * (r - rL)$$

Punkt auf der Kalibriergeraden:

$$iV = (rV - rL) * (iH - iL) / (rH - rL) + iL$$

Der Befehl '**settempcaliblow f<ideal>**' dient zur Messung und Eingabe des tiefen Kalibrierwertes. Zur Kalibrierung des tiefen Punktes wird die Sonde z.B. in Eiswasser gehalten. Die Temperatur wird zusätzlich mit einem genauen (idealen) Referenzthermometer gemessen. Die ideale Temperatur wird in °C abgelesen und als <ideal>-Wert eingegeben, z.B. 0.1
Die Befehlszeile unten zeigt dies.

```
settempcaliblow f0.1|
```

'f' kennzeichnet eine Float-Zahl (1 Nachkommastelle, Komma als Punkt geschrieben). Zwischen Befehl und f-Parameter muss mindestens ein Leerschlag stehen.

Der Befehl '**settempcalibhigh f<ideal>**' dient zur Messung und Eingabe der hohen Kalibrierwerte. Zur Kalibrierung des hohen Punktes wird die Sonde z.B. in auf 30°C erhitztes Wasser eingetaucht. Die Temperatur wird zusätzlich mit dem genauen (idealen) Referenzthermometer gemessen. Die ideale Temperatur wird wiederum abgelesen und als <ideal>-Wert eingegeben, z.B. 30.2

```
settempcalibhigh f30.2
```

Mit obigem 'tempcalib' können die Werte zusammengestellt angesehen werden.

'**conductivity**' zeigt die momentan gemessene, kalibrierte Leitfähigkeit am Bildschirm. Die Leitfähigkeit wird mit einer Einpunktkalibrierung eingestellt, die durch die Wasserbox temperaturkompensiert wird.

'**conductivitycalib**' zeigt zwei Leitfähigkeitswerte, die nicht-flüchtig im ESP32 gespeichert sind. Mit einer Kalibrierflüssigkeit bekannter Leitfähigkeit wird der Wert (idealerweise bei 25 °C) eingestellt.

Einpunktkalibrierung und Temperaturkompensation:

TODO

Der Befehl '**setconductivitycalib f<ideal>**' dient zur Messung und Eingabe des Kalibrierwertes.

'**turbidity**' zeigt die momentan gemessene, kalibrierte Trübung am Bildschirm. Es wird von einer Zweipunktkalibrierung ausgegangen, wie sie für die Temperatur beschrieben ist. Mit zwei Kalibrierflüssigkeiten bekannter Trübung werden die Werte eingestellt.

TODO: Ist dies die korrekte Vorgehensweise?

'**turbiditycalib**' zeigt vier Trübungswerte, die nicht-flüchtig im ESP32 gespeichert sind.

Der Befehl '**set**turbiditycaliblow f<ideal>' dient zur Messung und Eingabe des tiefen Kalibrierwertes.

Der Befehl '**set**turbiditycalibhigh f<ideal>' dient zur Messung und Eingabe des hohen Kalibrierwertes.

4.2. helptimedate

'**helptimedate**' zeigt die Befehle an, mit welchen Zeit und Datum der Echtzeituhr (Real Time Clock, RTC) angezeigt und eingestellt werden können.

Voraussetzung ist eine funktionierende Uhr. Die Stütz-Knopfzelle muss eingesetzt sein und genügend Ladung aufweisen. Die Zellenladung reicht sicher für ein Jahr oder mehr. Adafruit empfiehlt, immer eine Knopfzelle eingesetzt zu haben, auch wenn diese entladen sein sollte. Ansonsten kann Arduino bei Zeitabfragen hängen.

```
=== DATE AND TIME ===
date ..... show the current RTC date
time ..... show the current RTC time
setdate i<dd> i<mm> i<yyyy> ... set the RTC date
settime i<hh> i<mm> i<ss> .... set the RTC time
```

'**date**' stellt das aktuell eingestellte Datum inkl. Zeit dar:

```
Wed, 3/03/2021 12:29:50
```

'**time**' kann synonym zu 'date' verwendet werden.

'**setdate** i<dd> i<mm> i<yyyy>' richtet das Datum. Die 'i' (integer, d.h. Ganzzahlenwerte) sind von der angegebenen Anzahl Ziffer gefolgt. Fehleingaben werden nicht überprüft. Beispiel für die Eingabe des 13. März 2021:

```
setdate i13 i03 i2021
```

Ob die Parameter untereinander mit Leerschlag abgetrennt sind oder nicht, spielt keine Rolle.

Analog wird die Zeit mit '**settime** i<dd> i<mm> i<yyyy>' eingestellt, z.B. 7 Uhr 12 Minuten 33 Sekunden:

```
settime i07 i12 i33
```

4.3. helplogfile

'**helplogfile**' zeigt:

```
=== LOGFILE ===
logfile ..... show name of logfile
setlogfile "<name>" ... set name of logfile
logfilecontents ..... show logfile (partial) contents
```

'**logfile**' zeigt den mit der nächsten Logging-Messung verbundenen Dateinamen. In diese Datei werden die gemessenen Werte geschrieben. Dies sind Datum, Zeit, Temperatur, Leitfähigkeit und Trübung, falls aktiviert (vgl. weiter unten).

Wurde kein Name eingegeben, verwendet die Wasserbox den Default-Namen 'noname.csv'.

```
noname.csv
```

Der Name wird mit einer Nummer modifiziert, sobald geloggt wird, damit mit diesem Namensteil bis 100 unterschiedliche Dateien abgelegt werden könnten: noname00.csv .. noname99.csv .

Wird geloggt, gelten folgende Namensregeln:

- Der Dateiname muss das FAT-Format 8.3 aufweisen. D.h. der Name (hier 'noname') darf maximal 8 Zeichen lang sein. Optional kann eine Extension (hier 'csv' für comma-separated values) durch einen Punkt getrennt angegeben werden. Maximale Extension-Länge sind 3 Zeichen.
- Ein Dateiname mit 8 Zeichen wird um 2 Zeichen gekürzt und mit einer zweistelligen Nummer beginnend ab '00' ergänzt. Sollte dieser Name auf der SD-Karte bereits vorhanden sein, wird hochgezählt.
- Ein Dateiname mit 7 Zeichen wird um 1 Zeichen gekürzt und um zwei Ziffern ergänzt.
- Alle andern Dateien mit mindestens 1 Zeichen Länge werden um zwei Ziffern ergänzt. Daher wird im Falle von 'noname.csv' bei einer leeren SD-Karte die Datei 'noname00.csv' erstellt, sobald das Logging gestartet wird.

'setlogfile **name** "<name>"' erlaubt das Definieren eines eigenen Namens. Obige Regeln sind zu beachten.

'logfile **contents**' zeigt den (Teil-)Inhalt der aktuellen Datei. Der Inhalt einer anderen Datei kann angezeigt werden, wenn vorgängig mit 'setlogfile **name** "<name>"' der Dateiname einer existierenden Datei angegeben wird.

Beispiel:

```
setlogfile "noname04.csv"
```

```
logfilecontents
```

```
contents of noname04.csv:
date [dd.mm.yyyy],time [hh:mm:ss],temperature [degree C]
24.06.2022,22:47:25,26.8
24.06.2022,22:47:35,27.2
24.06.2022,22:47:45,28.3
... (last lines)
24.06.2022,22:48:54,34.4
24.06.2022,22:48:56,34.2
24.06.2022,22:49:05,33.9
eof
```

Falls möglich, werden die ersten und die letzten drei Zeilen angezeigt.

4.4. helplogger

'helplogger' liefert:

```
=== LOGGER ===
setloggerstarttime i<hh> i<mm> ..... set logging start time for date today
setloggerstartdate i<dd> i<mm> i<yyyy> ... set logging start date if not today
setloggerstoptime i<hh> i<mm> ..... set logging stop time for date today
setloggerstopdate i<dd> i<mm> i<yyyy> ... set logging stop date if not today
setloggerinterval i<hh> i<mm> i<ss> ..... set logging interval
setloggerintervaldays i<dd> ..... set days of logging interval
resetlogger ..... reset all logging times and dates
loggersettings ..... show logging settings
setloggertemp on ..... include temp in logging
setloggertemp off ..... exclude temp from logging
setloggerconductivity on ..... include conductivity in logging
setloggerconductivity off ..... exclude conductivity from logging
setloggerturbidity on ..... include turbidity in logging
setloggerturbidity off ..... exclude turbidity from logging
loggerrun ..... run logging
loggerhalt ..... halt logging
```

Mit den Loggerbefehlen wird das Logging (Daten-Schreiben auf SD-Karte) vorbereitet und schliesslich gestartet.

'setloggerstarttime i<hh> i<mm>' definiert, ab welchem Zeitpunkt das Loggen gestartet wird. Der Parameter 'hh' entspricht den Stunden (24-Stunden-Format), der Parameter 'mm' entspricht den Minuten.

```
setloggerstartdate i16i45
```

Auf dem Monitor wird quittiert:

```
logger start: 03/03/2021 16:45:00
```

Mit `'setloggerstoptime i<hh> i<mm>'` wird definiert, wann das Loggen beendet wird. In keinem der Fälle wird getestet, ob die Zeiten plausibel sind.

`'setloggerinterval i<hh> i<mm> i<ss>'` erlaubt die Definition des Logging-Intervalls. Per Default ist der kleinste Wert 5 Sekunden festgelegt, der sich auf diese Weise überschreiben lässt.

```
setloggerinterval i00 i02 i30
```

Die vorgenommenen Einstellungen lassen sich mit `'loggersettings'` überprüfen:

```
log file: noname.csv
logger start: 10/09/2022 16:45:00
logger stop: 10/09/2022 22:30:00
logger interval: 00:02:30
logger running: no
temperature logging on: no
conductivity logging on: no
turbidity logging on: no
```

Sollen die Messungen über mehrere Tage vorgenommen werden, stehen die beiden Befehle `'setloggerstartdate i<dd> i<mm> i<yyyy>'` und `'setloggerstopdate i<dd> i<mm> i<yyyy>'` zur Verfügung.

Auch das Logging-Intervall kann grösser als 24 Stunden gesetzt werden: `'setloggerintervaldays i<dd>'`

Mit `'resetlogger'` werden alle Zeiten zurückgesetzt.

Beim Loggen kann die Temperatur etc. ein- oder ausgeschlossen werden. Dazu dienen die Befehle

`'setloggertempon'` bzw. `'setloggertempoff'`,
`'setloggerconductivityon'` bzw. `'setloggerconductivityoff'` und
`'setloggerturbidityon'` bzw. `'setloggerturbidityoff'`

Schliesslich kann das Logging mit `'loggerrun'` gestartet werden. Hierbei wird ein neues Logfile erstellt und auf den Zeitpunkt gewartet, der mit dem Startzeitpunkt fürs Loggen übereinstimmt. Wird der Endzeitpunkt erreicht, stoppt das Logging, was angezeigt wird. Während des Loggens können weiterhin viele Befehle benutzt werden. Insbesondere kann das Loggen auch vorzeitig abgebrochen werden. Dazu dient `'loggerhalt'`.
Läuft das Logging, sind einige Befehle blockiert, um die Messprotokollierung konsistent zu halten.

Das Format der Daten, die geloggt werden, ist geeignet, um diese als csv-Datei in Excel zu überführen.

4.5. helpmonitor

`'helpmonitor'` stellt folgende Befehle dar:

```
=== MONITOR ===
setmonitorstarttime i<hh> i<mm> ..... set monitoring start time for date today
setmonitorstartdate i<dd> i<mm> i<yyyy> ... set monitoring start date if not today
setmonitorstoptime i<hh> i<mm> ..... set monitoring stop time for date today
setmonitorstopdate i<dd> i<mm> i<yyyy> ... set monitoring stop date if not today
setmonitorinterval i<hh> i<mm> i<ss> ..... set monitoring interval
addmonitorintervaldays i<dd> ..... add days to monitoring interval
resetmonitor ..... reset all monitoring times and dates
monitorsettings ..... show monitoring settings
setmonitortempon ..... include temp in monitoring
setmonitortempoff ..... exclude temp from monitoring
setmonitorconductivityon ..... include conductivity in monitoring
setmonitorconductivityoff ..... exclude conductivity from monitoring
setmonitorturbidityon ..... include turbidity in monitoring
setmonitorturbidityoff ..... exclude turbidity from monitoring
monitorrun ..... run monitoring
monitorhalt ..... halt monitoring
```

Diese Monitor-Befehle finden ihr Pendant beim Logger.

Die Daten werden hier am Monitor dargestellt, statt dass sie in eine Log-Datei geschrieben werden.

Logger und Monitor können unabhängig voneinander verwendet werden, auch gleichzeitig oder zeitlich überlappend.

4.6. helplcdmode

'**helplcdmode**' ist im Zusammenhang mit der 4-Zeilen-Anzeige und der Bedientaste wichtig. Die Hilfe zeigt folgende Befehle:

```
=== LCD DISPLAY MODE ===
setlcdfilename "<name>" ..... set logfile name for lcd display mode
setlcdloggerinterval i<mm> i<ss> ... set logging interval for lcd display mode (>= 10 sec)
setbacklighttimeout i<min> ..... set timeout period for lcd backlight; -1: no timeout
lcdsettings ..... show settings for lcd display mode
```

'**setlcdfilename** "<name>"' definiert, welcher Dateiname beim automatischen Dateianlegen verwendet wird. <name> unterliegt den Regeln wie in 4.3. beschrieben. Dieser Name wird im ESP32 gespeichert und steht auch bei späteren Messwernerfassungen als Stammmname zur Verfügung. Es können bis 100 Dateien mit einem Stammmnamen erzeugt werden.

Beispiel: Mit 'setlcdfilename "lcd.csv"' werden im LCD Display Mode in der Folge die Dateien 'lcd00.csv', 'lcd01.csv', 'lcd02.csv' etc. automatisch angelegt. Nach Erreichen von 'lcd99.csv' wird an der Wasserbox eine Fehlermeldung ausgegeben. Dann muss hier der Name geändert oder eine andere SD-Karte eingesetzt werden.

'**setlcdloggerinterval** i<mm> i<ss>' konfiguriert das Zeitintervall zwischen den Messwernerfassungen, die in die Datei geschrieben werden. Kleinster Wert ist 10 Sekunden, grösster Wert 59 Minuten 59 Sekunden. Das Zeitintervall wird im ESP32 gespeichert und bleibt bis zur Neukonfiguration erhalten.

'**setbacklighttimeout** i<min>' definiert in Minuten, wie lange die Hintergrundbeleuchtung des LCD Displays leuchtet, falls die Bedientaste nicht mehr berührt wird. Jedes Berühren der Bedientaste setzt das Timeout zurück.

Mit '**lcdsettings**' werden die Einstellungen im Zusammenhang mit dem LCD Display Modus aufgelistet:

```
lcd display mode settings
file name: lcd00.csv
logging interval: 1 min 0 sec
lcd backlight timeout: 5 min, < 0: no timeout
button short-press: < 1 sec (not settable)
button long-press: > 3 sec (not settable)
```

Der aktuelle Stammmname (allenfalls mit einer Nummer versehen) für das automatische Bilden des Dateinamens wird angezeigt. Das Logging Intervall ist ersichtlich sowie Zeitangaben zum kurzen wie zum langen Tastendruck werden angezeigt. Diese beiden Werte können nicht verändert werden. Zudem lässt sich das eingestellte Backlight Timeout ansehen.

4.7. helpmisc

'**helpmisc**' zeigt verschiedene weitere Befehle:

```
=== MISCELLANEOUS ===
version ..... show version
cardinfo ..... show SD card info
setechoon ..... set echo input in monitor window
setechooff ..... suppress echo in monitor window
serial ..... communicate via Serial
wifi ..... show wifi settings
setnet ..... set wifi connection to local network
setap ..... set wifi as access point
setnetcredentials "<ssid>" "<pwd>" ... set network credentials
setapcredentials "<ssid>" "<pwd>" .... set access point credentials
license ..... show license text
poweroff ..... power off the system
```

'**version**' zeigt die Version der Software an.

'**cardinfo**' liefert Angaben zur verwendeten SD-Karte.

'**setechoon**' bewirkt nach Ausführung, dass Eingabetext als Ausgabe - mit '>>>' vorangestellt - wiedergegeben wird. Dies ist dort nützlich, wo das Kommunikationsprogramm kein Echo ausgibt.

'**setechooff**' hebt die Ausgabe des Eingabetexts auf. Dies ist die Defaulteinstellung der Wasserbox.

'**serial**' leitet die Eingaben eines Terminal-Programms via USB-Kabel zum Serial-Port des ESP32 bzw. leitet ESP32-Ausgaben vom Serial-Port zum USB-Port des Terminal-Programms um. Ueber diesen Port wird auch der Upload neuer Software vorgenommen.

'**wifi**' zeigt WLAN-Einstellungen an. Es ist entweder Netzwerk- oder Access-Point-Betrieb mit entsprechender SSID und entsprechendem Passwort möglich.

```
network ssid: ZyXEL, network pwd: ??????????
access point ssid: wbox-ap, access point pwd: 123456789
wifi method [net, ap]: ap; method change active only after restart!
```

Nach Aenderung mittels '**setnet**' oder '**setap**' ist jeweils ein Reset des ESP32 nötig.

SSID und Passwort sind mit '**setnetcredentials** "<ssid>" "<pwd>"' bzw. '**setapcredentials** "<ssid>" "<pwd>"' vorzunehmen.

'**license**' zeigt die MIT Lizenz für die Software an.

'**poweroff**' löst die Selbsthaltung: Die Wasserbox wird stromlos geschaltet, sofern sie mit dem Akku betrieben wird.

4.8. helpdevel

Mit '**helpdevel**' werden Befehle angezeigt, die für die Entwicklung, beim Testen und z.T. bei der Fehlersuche nützlich sind.

```
=== DEVELOPMENT ===
setmonitorsettings ..... set test times for monitor testing
setloggersettings ..... set test times for logger testing
testrunmonitor ..... run monitor with all sensors on
testrunlogger ..... run logger with all sensors on
copymonitordatestologger ..... copy monitor dates and times to logger
fakesettempcaliblow f<sensor> f<ideal> ..... fake and set temp calib low values
fakesettempcalibhigh f<sensor> f<ideal> ..... fake and set temp calib high values
fakesetconductivitycalib f<sensor> f<ideal> f<temp> ... fake and set conductivity calib values
fakesetturbiditycaliblow f<sensor> f<ideal> ..... fake and set turbidity calib low values
fakesetturbiditycalibhigh f<sensor> f<ideal> ..... fake and set turbidity calib high values
prefs ..... show preferences
clearprefs ..... CAUTION: clear all preferences
enableserial1 ..... enable Serial1 (after restart)
disableserial1 ..... disable Serial1 (after restart)
serial1 ..... communicate via Serial1
startlcdmode ..... enable lcd display mode
stoplcdmode ..... disable lcd display mode
startwifimode ..... connect to wifi
stopwifimode ..... deactivate wifi
directory ..... show sd card root files
test0 ..... show temperature values
test1 ..... show conductivity values
test2 ..... show turbidity values
test3 ..... set LCD backlight on
test4 ..... set LCD backlight off
test5 ..... show system (esp32) date and time
test6 ..... not used
test7 ..... not used
test8 ..... not used
test9 i<param> ..... test parameter passing
```


'**setmonitorsettings**' definiert eine 2-Minuten-Messung mit 10-Sekunden-Intervall für Testzwecke im Monitorbetrieb.

'**setloggersettings**' definiert eine 2-Minuten-Messung mit 10-Sekunden-Intervall für Testzwecke im Loggerbetrieb.

'**testrunmonitor**' testet den Monitor mit Setzen von Zeiten, Wahl aller Sensoren und Starten des Monitors. Nach 1 bis 2 Minuten werden die Werte für 2 Minuten auf dem Monitor angezeigt.

'**testrunlogger**' testet den Logger mit Setzen von Zeiten, Wahl aller Sensoren und Starten des Monitors. Nach 1 bis 2 Minuten werden die Werte für 2 Minuten auf die SD-Karte gespeichert. Defaultname ist 'logtst.csv'.

Mit '**copymonitordatestologger**' werden Start-, Stop- und Intervallzeiten vom Monitor auf den Logger kopiert.

'**fakesettempcaliblow f<sensor> f<ideal>**' und '**fakesettempcalibhigh f<sensor> f<ideal>**' erlauben es, Kalibriertemperaturen zu setzen, ohne dass gemessen wird: Praktisch für Testzwecke.

'**fakesetconductivitycalib f<sensor> f<ideal> f<temp>**', '**fakesetturbiditycaliblow f<sensor> f<ideal>**' und '**fakesetturbiditycalibhigh f<sensor> f<ideal>**' sind die Pendanten für die weiteren Sensoren.

'**prefs**' stellt die gespeicherten Präferenzen, also die Parameter, dar.

'**clearprefs**' löscht ALLE Präferenzen.

'**serial1**' stellt einen Zusatzport zur Verfügung. Hier kann ein zweites Terminalprogramm zum Einsatz kommen. Vgl. auch 6.

'**enableserial1**' und '**disableserial1**' aktivieren/deaktivieren die Serial1-Schnittstelle, an der ein FTDI-Breakout angeschlossen sein muss, welches per USB-Kabel an einen PC angeschlossen ist. Vgl. auch 6.

'**startlcdmode**' startet die 4-Zeilenanzeige. Nützlich vor allem nach Verwendung von 'stoplcdmode'.

'**stoplcdmode**' stoppt die 4-Zeilenanzeige. Die Anzeige steht, Aktionen sind unterbunden. Nützlich, um Zeit- und Synchronisationsprobleme zu vermeiden, wenn andere Befehle via Serial ausgeführt werden.

'**startwifimode**' startet das Verbinden via WLAN.

'**stopwifimode**' beendet die WLAN-Verbindung.

'**directory**' zeigt alle Dateien im Root-Verzeichnis (ohne Unterverzeichnisse) an.

'**test0**' bis '**test9**' sind Befehle, die für rasches Testen und Aendern von Testfunktionen gedacht sind. Diese können rasch umdefiniert werden, ohne dass an den Befehlsdateistrukturen Aenderungen vorgenommen werden müssen.

5. Expert-Modus via WLAN

5.1. Voraussetzung

Im Kabelmodus müssen SSID und Passwort gesetzt worden sein. Nötigenfalls kann dies für Access Point wie folgt nachgeholt werden:

Nach Verbinden und Kontaktaufnahme der Wasserbox ist z.B. einzugeben:

```
>>> setapcredentials "wbox" "123456789"  
access point ssid: wbox, access point pwd: 123456789
```

Hinweis:

Das Passwort muss mindestens 9 Zeichen lang sein.

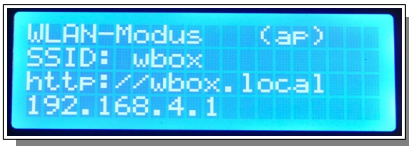
Danach ist Access Point zu wählen:

```
>>> setap  
wifi method [net, ap]: ap; method change active only after restart!
```

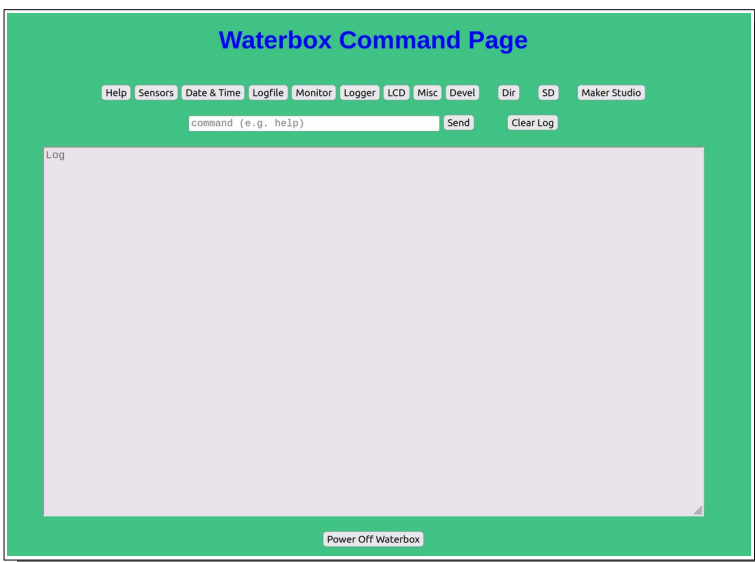
Schliesslich ist die Resettaste des ESP32 (die kleine schwarze Taste gleich bei der USB-Buchse auf dem ESP32 HUZZAH) zu drücken.

5.2. Verbinden via Access Point (ohne USB-Kabel zum PC)

Nach Abwarten der ersten 5 Sekunden nach Reset ist die Bedientaste lang (> 10 Sekunden) zu drücken. Nach 4 Sekunden erscheint oben rechts im Display ein blinkendes Radio-Symbol, schliesslich zeigt das Display den WLAN-Modus an.



Auf dem PC, dem Tablet oder dem Handy ist das Netzwerk 'wbox' zu wählen und das Passwort '123456789' einzugeben, um zu verbinden. Danach wird ein Browser geöffnet und die URL '<http://192.168.4.1>' bzw. '192.168.4.1' eingegeben. '<https://192.168.4.1>' funktioniert nicht. Es wird nun die Seite 'Waterbox Command Page' angezeigt.



Mit Eingabe von 'help' oder einfacher mit Drücken der Help-Schaltfläche werden die Befehlsgruppen wie unter '4. Bedienung' angezeigt.

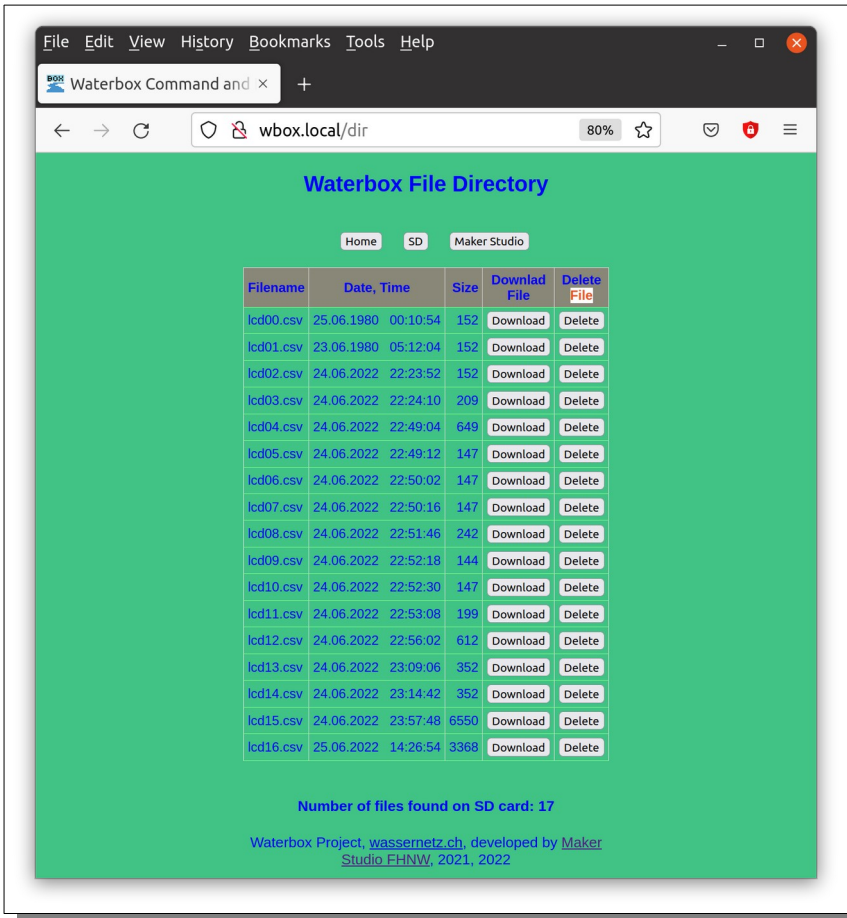
Hinweis:

Oftmals, leider nicht zuverlässig regelmässig, kann die Seite statt mit '192.168.4.1' mit '<http://wbox.local>' bzw. 'wbox.local' aufgerufen werden.

Hinweis: '<https://wbox.local>' funktioniert nicht!

5.3. Log-Dateien auf der SD-Karte

Mit Klicken auf 'Dir' wird auf einer neuen Seite das 'Waterbox File Directory' mit den Dateien auf dem Root-Verzeichnis dargestellt. Unterverzeichnisse werden nicht dargestellt.



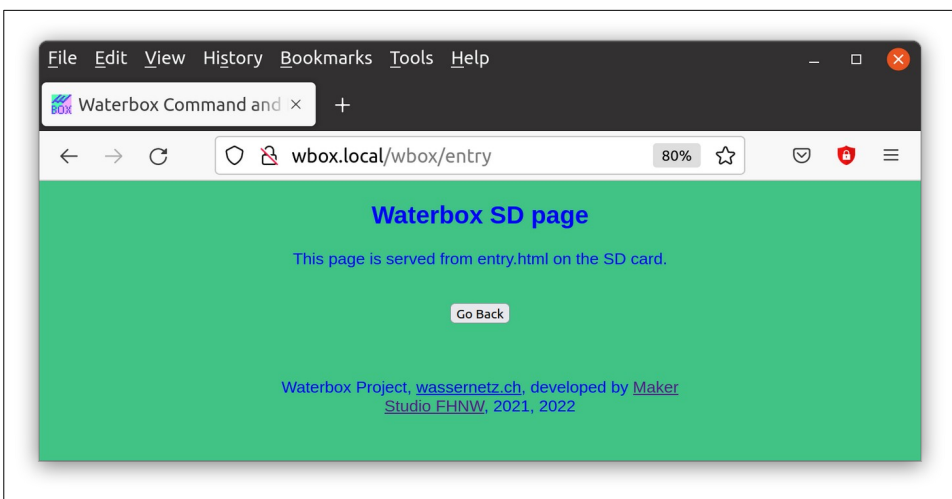
Hier lassen sich einzelne Files nacheinander herunterladen oder auch löschen. Mit 'Home' gelangt man auf die Homepage zurück.

Hinweis:

Die Schaltfläche 'MakerStudio' und die Links unten auf der Seite funktionieren nur, falls 'net'-statt 'ap'-Betrieb gewählt wurde.

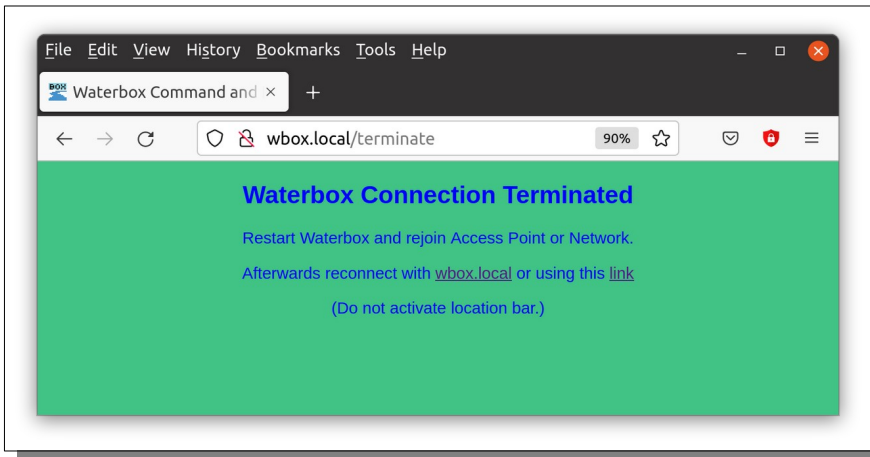
5.4. SD-Schaltfläche

'SD' stellt eine neue Seite dar, die mit Code aus dem Unterverzeichnis 'wbox' der SD-Karte dargestellt wird. Sie ist als Hook-up für allfällige spätere HTML-Erweiterungen ab SD-Karte gedacht.



5.5. Ausschalten nach Aufruf des Expert-Modus

Wurde der Expert-Modus regulär, also bei Verwendung eines Akku bei geschlossener Box aufgerufen, wo kein Zugang zu Reset und Stecker besteht, wird die Wasserbox auf der Homepage mit der Schaltfläche 'Power Off Waterbox' ausgeschaltet.



Eine Kommunikation via Serial1 ist programmiert, jedoch nicht aktiv. Bei Bedarf muss der Quellcode neu kompiliert werden. Danach ist Serial1 gleichberechtigt wie Serial. Allerdings muss hardware-seitig ein Serial-USB-Wandler angeschlossen werden.

6. SERIAL1

Für EntwicklerInnen und Testende:

Für Testzwecke besteht die Möglichkeit, die Kommunikation via 'serial1' auf ein anderes Terminalprogramm umzuleiten, um erweiterte Textausgaben zu erhalten.

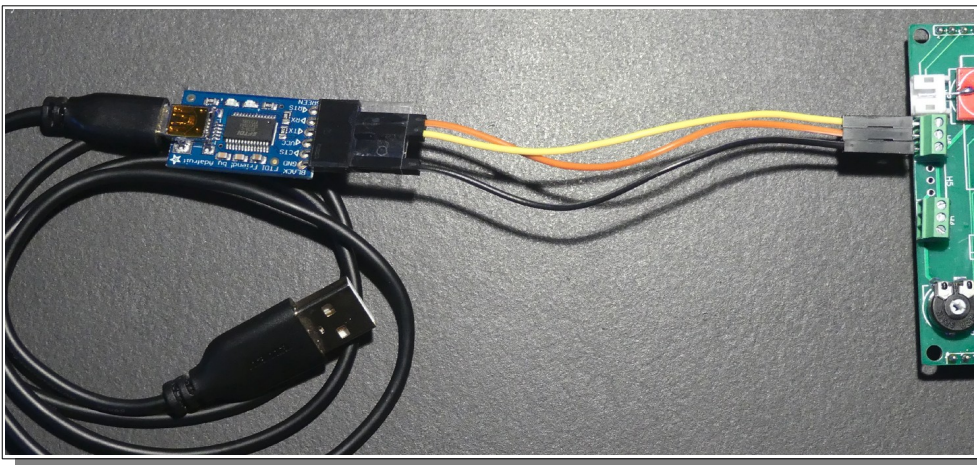
Dies Terminalprogramm kann der Serial Monitor der Arduino-IDE sein oder ein anderes, z.B.

PuTTY, miniterm.py für Linux (Ubuntu) oder

PuTTY, TeraTerm für Windows oder

PuTTY für iOS

Auf der ESP32-Seite muss dazu ein Terminalanschluss bei H6 eingelötet und ein Adafruit FTDI-Friend mit Kabel zum Gerät mit dem Monitorprogramm verbunden werden. Zudem muss Serial1 enabled sein. Vgl. bei 4.



7. Fehlermeldungen

7.1. SD 'card error'

Ist die Micro-Karte (vollständig) eingesteckt/ingerastet?
Ist die Knopfbatterie vorhanden und gut platziert?

Hinweise:

Die Kopfbatterie hat eine Lebensdauer von einigen Jahren.
Die Wasserbox ist grundsätzlich nicht einsetzbar, wenn die SD-Karte fehlt.

7.2. Fehler im LCD Display Modus

Hat die automatische Nummerierung des Dateinamens die Nummer 99 erreicht, wird beim nächsten Versuch eine Datei anzulegen, eine Fehler angezeigt. Es muss ein neuer Name gewählt werden, oder es müssen Dateien gelöscht werden, oder die SD-Karte muss neu formatiert werden.

Hinweis:

Auf der SD-Karte befindet sich in Unterverzeichnis/sen HTML-Code. Dieser muss wieder auf die Karte kopiert werden.

7.3. 'wire transmission error'

Die Kommunikation mit dem RTC (der Uhr) oder dem Temperatur-Sensor funktioniert nicht.
Ist der Adafruit Logger gut eingesteckt (Kontakte)?
Ist die Knopfbatterie vorhanden und gut platziert?

Hinweis:

Die Kopfbatterie hat eine Lebensdauer von einigen Jahren.

7.4. Nichtplausible Sensorwerte

Sind die Sensoren korrekt angeschlossen? Werden die Sensor-Breakouts mit Spannung versorgt (leuchtende LEDs)?