

Das PiLogger One Handbuch



Inhaltsverzeichnis

1 Einleitung	5
2 Sicherheitshinweise	6
3 Auswahl des Gastgeber-Rechners (Host)	7
3.1 Raspberry Pi Familie	7
3.2 ESP32 Microcontroller	9
4 Installation WebMonitor auf Raspberry Pi	10
4.1 Raspberry Pi OS installieren	10
4.1.1 Auswahl Betriebssystem	10
4.1.2 Voraussetzungen für die SD-Kartenerstellung	10
4.1.3 Download Raspberry Pi Imager	11
4.1.4 Raspberry Pi Imager verwenden	12
4.1.5 Erstes Booten, SSH Verbindung	18
4.1.6 System einrichten	20
4.1.6.1 Raspi-Config updaten	21
4.1.6.2 System Options	22
4.1.6.3 Display Options	23
4.1.6.4 Interface Options	23
4.1.6.4.1 4 2C	24
4.1.6.5 Performance Options	
4.1.0.0 Euclaisation Options	25
4.1.6.6.2 3 Keyboard	
4.1.6.7 Advanced Options	29
4.1.6.8 Raspi-Config beenden	29
4.1.7 Allgemeines Update	30
4.1.7.1 Firmware Update	30
4.1.7.2 Distributions-Update	30
4.2 Montage und Inbetriebnahme	32
4.2.1 Montage PiLogger One Typ A	32
4.2.1.1 Vorbereitung	32
4.2.1.2 Modul aufsetzen	35
4.2.2 Anschluss Pillogger One Typ B	
4.2.3 Erstes Einschalten.	40
4.3 PILogger WebMonitor Installieren	41
4.3.1 Router als NTP-Server einrichten	41
4.3.2 Installer stanen	42
5 Installation WebMonitor auf ESP32 Modul	46
5.1 Adapter und Modul verlöten	47
5.1.1 Adapter mit NodeMCU ESP32 Modul	49
5.1.2 Belegung optionale Stiftfelder	50
5.2 Installation Thonny	51
5.3 Uberprüfung/Aktualisierung 'esptool'	52
5.4 Vorbereiten Modul mit Adapter	53
5.5 Flashen MicroPython	53
5.6 Kopieren der Archiv-Dateien	56

5.7 Anpassen der WiFi Zugangsdaten 5.8 Netzwerknamen ändern	.58 .59
5.9 PiLogger One montieren	.60
5.10 Erstes Einschalten	.62
 6 Bedienungsanleitung WebMonitor Software	.63 .64 .66 .67 .68 .69 .70 .71 .72 .72 .74 .75 .78 .82
6.8 Die Seite Download	.84
 7.1 Gleichspannungs- und Strommessung 7.2 Impulszähleingang 7.3 Temperaturfühlereingang 	.86 .87 .88
8 Anschlussbelegung GPIO-Leiste	.90
9 Anwendungs-Programmierschnittstelle 9.1 Konfigurationsregister 9.2 Messwertregister 9.3 Wertebereiche und Skalierung	.91 .91 .93 .94
10 Temperaturfühler-Tabellen	.96
11 Technische Daten	.99
12 Kalibrierung. 1 12.1 Kalibrierung Temperaturmessung. 1 12.2 Kalibrierung Impulszählung. 1 12.3 Kalibrierung Spannungsmessung. 1 12.4 Kalibrierung Strommessung. 1	L 00 L00 L02 L02 L02
13 Einbindung in Smart Home Systeme1 13.1 Home Assistant1	L06 L06
14 Anhang 1 14.1 Die I ² C Tools 1 14.2 Konzept PiLogger WebMonitor 1	L10 110 111

Handbuch PiLogger One

Raspberry Pi ® is a trademark of the Raspberry Pi Foundation.

 $\ensuremath{\text{Linux}}\xspace$ is a registered trademark of Linus Torvalds.

Windows ® is a registered trademark of Microsoft Corporation.

All other trademarks are the property of their respective owners.

1 Einleitung

Der PiLogger One ist eine Erweiterung für den Raspberry Pi. Mit dem PiLogger One wird ein eigenständiges 4-Kanal Messsystem über den I²C Bus an den Raspberry Pi angebunden.

Mit der Variante 'Typ B' ist der Anschluss über ein kurzes 5-adriges Kabel an den Gastrechner möglich – damit bleibt die GPIO-Leiste größtenteils weiter verfügbar. Außerdem gibt es einen Adapter für einen ESP32 Microcontroller und eine speziell hierfür angepasste Software-Version.

Der Schwerpunkt liegt auf der Gleichstrom-Leistungsmessung (DC power meter), ergänzt um einen Impulszähler mit dem Drehraten gemessen werden können, wie zum Beispiel für Windgeschwindigkeit oder Durchflussmenge, sowie einem Messeingang für Temperaturfühler. Genaueres hierzu findet sich im Kapitel 'Technische Daten'.

Diese Kombination eignet sich besonders für die Überwachung kleiner alternativer Stromerzeuger, wie Windräder oder Solar-Anlagen und insbesondere deren Akkus.

Der Clou ist das Konzept einer eigenständig arbeitenden Peripherie-Einheit. Der PiLogger hat seine eigene Zeitbasis, mit welcher er präzise Messintervalle einhalten kann, unabhängig von den Software-Aktivitäten auf dem Gastrechner. Der PiLogger übernimmt die periodischen Messungen, die Leistungsberechnung und die gleitende Mittelwertbildung. Der Raspberry Pi muss nur zu den gewünschten Log-Zeitpunkten eine Abfrage durchführen und kann dann je nach Wunsch weiter auswerten und speichern. Die Programmier-Schnittstelle wird in einem eigenen Kapitel beschrieben.

Mit der Vielseitigkeit des Raspberry Pi und insbesondere mit der Möglichkeit sich über LAN und WLAN zu vernetzen, eröffnen sich tolle Möglichkeiten – zum Beispiel ein kleiner Server mit grafischen Auswertungen, der freien Software 'WebMonitor'.

Dieser Web-Server stellt im heimischen Netzwerk die Messwerte des PiLogger One zur Verfügung. Diese können dann in Smart Home Systemen, wie zum Beispiel 'Home Assistant' verwendet werden oder direkt über das Web-Interface mit einem Browser auf jedem Endgerät angezeigt werden.

Dieses Handbuch versucht eine Schritt für Schritt Anleitung auch für Einsteiger zu bieten. Fortgeschrittene Benutzer können die ersten Kapitel überspringen. Die letzten Kapitel dienen als technische Referenz und Dokumentation für die erfolgreiche Einbindung des PiLoggers in die eigenen Projekte.

Viel Spaß und Erfolg mit dem PiLogger !

2 Sicherheitshinweise

Die folgenden Sicherheits- und Gefahrenhinweise dienen nicht nur zum Schutz Ihrer Gesundheit, sondern auch zum Schutz des Geräts. Lesen Sie sich bitte die folgenden Punkte aufmerksam durch:

- Vor Berührung und beim Anschluss der Platine sind geeignete Schutzmaßnahmen gegen statische Aufladung zu treffen (z.B. Erdungsband, nichtleitende Unterlage usw.).
- Montage und Anschluss dürfen nur im unbestromten Zustand vorgenommen werden. Alle Stromversorgungen sind auszuschalten.
- Das Produkt ist nur für trockene, geschlossene Innenräume geeignet.
 Es darf nicht feucht oder nass werden, andernfalls kann es zu Beschädigungen kommen.
- Schützen Sie das Produkt vor Kälte, Hitze, direkter Sonneneinstrahlung, Staub und Schmutz.
- Gehen Sie vorsichtig mit dem Produkt um, durch Stöße, Schläge oder dem Fall aus bereits geringer Höhe kann es beschädigt werden.
- Das Produkt ist kein Spielzeug und gehört nicht in Kinderhände. Platzieren Sie das Produkt so, dass es von Kindern nicht erreicht werden kann.
- Lassen Sie das Gerät, sowie das Verpackungsmaterial nicht achtlos liegen, dieses könnte für Kinder zu einem gefährlichen Spielzeug werden.

Achten Sie beim Anbringen der Messkabel auf gute Isolierung und Berührungsschutz. Stromschlaggefahr !



Achten Sie beim Anbringen der Messkabel auf Zugentlastung und einen stabilen Aufbau um Kurzschlüsse zu vermeiden !

3 Auswahl des Gastgeber-Rechners (Host)

Der PiLogger One ist im Grunde ein Multi-Sensor am I²C Bus. Somit kommen zunächst alle Rechner in Betracht, die einen I²C Anschluss zur Verfügung stellen. Traditionell sind dies die Raspberry Pi's mit ihren frei zugänglichen GPIO-Leisten (**G**eneral **P**urpose Input **O**utput – allgemein verwendbare Eingangs-/Ausgangs-Pins).

Der PiLogger One selber speichert keine Messwerte, deshalb ist für einen Logger (Datenschreiber) die Verfügbarkeit eines schnellen permanenten Speichermediums ein wichtiges Kriterium – sprich : eine SD-Karte. Alle bisherigen Raspberry Pi Modelle bieten diese. Bei ESP32 basierten Rechnermodulen sind sie aber nicht Standard. Mit unserer Adapterplatte kann ein ESP32 WROOM Modul mit einem Micro SD Kartenhalter aufgerüstet werden.

Ein weiteres Kriterium ist die Abwägung zwischen Stromverbrauch und Rechenleistung. Ist der PiLogger One lediglich eine zusätzliche Erweiterung für einen Rechner der weitere Aufgaben erfüllt, ist der Stromverbrauch zweitrangig. Dafür müssen aber die Sensorleitungen an den Ort des Rechners geführt werden. Ist der PiLogger One die Hauptaufgabe, zum Beispiel in einer Inselanlage zur Batterieüberwachung, ist ein kleiner stromsparender Rechner mit Netzwerkanbindung die bessere Wahl. Insbesondere weil die 'PiLogger WebMonitor' Software darauf basiert, dass Anzeige und Bedienung über einen Browser erfolgen – also nicht unbedingt lokal auf dem Host-Rechner, kann der Host-Rechner für den PiLogger One auf Display und Tastatur verzichten und stattdessen nahe am Messort montiert werden.

3.1 Raspberry Pi Familie

Die Grundversion des PiLogger One ist mit einer 26-poligen Buchsenleiste ausgestattet, die direkt auf die GPIO-Leiste eines Raspberry Pi 1 passt. Sie passt auch auf die 40-poligen Stiftleisten der nachfolgenden Generationen des Raspberry Pi – dabei wird nur der rückwärts kompatible Teil der Leiste benutzt.

Es kommen also tatsächlich alle Versionen, einschließlich der Zero's in Betracht. Die Rechenleistung auch der kleinsten Versionen, die mit einem Single-Core-Prozessor ausgestattet sind, wie der Zero 1 und die vier 1er (A, B und A+, B+), ist mehr als ausreichend für den Betrieb des PiLogger One mit dem WebMonitor.

Da alle Raspberry Pi's mit einem vollständigen Linux-Betriebssystem betrieben werden, ist das Anschließen von Standardperipherie in der Regel kein Problem. Damit können auch Pi's die kein LAN oder WLAN an Bord haben beliebig nachgerüstet werden.

Beispielsweise gibt es LAN-Adapter mit Micro-USB-Stecker, sodass auch ein

Raspberry Pi Zero (ohne W, wie WLAN) damit ins Heimnetzwerk eingebunden werden kann. Die gibt es sogar mit POE (**P**ower **O**ver **E**thernet), sodass der Raspi direkt auch mit Strom versorgt werden kann.

Oder ein WLAN-Adapter mit Stab-Antenne, oder ... oder

Das heißt letztlich, dass die Entscheidung für einen bestimmten Raspberry Pi von der geplanten Anwendung und den örtlichen Gegebenheiten abhängt: Direkt am Messort eher ein kleiner, sparsamer Raspi und im Wohnzimmer am Monitor vielleicht doch Raspi 3 oder 4.



Bild 1: PiLogger One auf Raspberry Pi Zero W



Bild 2: PiLogger One auf Raspberry 4

Die Einrichtung unter Raspberry Pi OS wird in Kapitel 4 beschrieben.

3.2 ESP32 Microcontroller

MicroPython ist eine spezielle Implementierung von Python für Microcontroller mit geringem Ressourcen-Bedarf. Es ist ein freies und offenes Software-Projekt von Damien George unter MIT Lizenz (<u>https://micropython.org/</u>).

Da die Software 'PiLogger WebMonitor' in Python geschrieben ist und weder Tastatur noch Display benötigt, ist MicroPython das Hauptkriterium für die Auswahl eines noch sparsameren Host-Rechners. Das zweite Kriterium ist die Netzwerkfähigkeit. Damit kommen zunächst alle Rechner-Module in die engere Wahl, die von einer MicroPython Portierung profitieren und etwa mit einem WLAN-Modul ausgestattet sind.

Die meisten dieser Module verfügen nicht über einen SD Kartenhalter und wenn, dann typischerweise nur mit einer SPI Anbindung mit nur einer Datenleitung. Das bedeutet eine weitere Geschwindigkeitseinbusse. Wir haben uns daher entschieden für ein 36-poliges ESP32 WROOM Modul eine Adapterplatte mit microSD Kartenhalter zu entwerfen, wobei die SD Karte mit der SDIO Schnittstelle mit 4 Datenleitungen angesteuert wird.

Für diese Kombination haben wir den 'PiLogger WebMonitor' erfolgreich portiert.

Der Strombedarf des WebMonitors ist damit fast 4 mal so sparsam wie mit einem Raspberry Pi Zero W : 230 mW zu 900 mW bei 5V.

Die Kehrseite ist eine höhere Antwortzeit bei Web-Anfragen. Das bedeutet eine etwas trägere Reaktion beim Seitenaufruf im Browser.





Bild 3: PiLogger One auf ESP32 WROOM

Die Einrichtung für die ESP32 Variante wird in Kapitel 5 beschrieben.

4 Installation WebMonitor auf Raspberry Pi

4.1 Raspberry Pi OS installieren

In diesem Kapitel wird Schritt für Schritt beschrieben wie der Raspberry Pi soweit vorbereitet wird, dass anschließend der PiLogger WebMonitor installiert werden kann.

4.1.1 Auswahl Betriebssystem

Der PiLogger WebMonitor baut auf der Raspberry-Pi-Seite vollständig auf Software-Komponenten auf Python-Basis. Damit ist prinzipiell der Einsatz auf jedem Betriebssystem (OS) mit Python-Unterstützung möglich. Lediglich die spezielle Hardware-Unterstützung für die Raspberry Pi GPIOs unter Python wird noch benötigt.

Das Ziel der weiteren Anleitung ist die einfache und direkte Beschreibung einer sogenannten 'headless'-Installation – also der Betrieb des Raspberry Pi ohne angeschlossenes Display und Eingabegerät. Damit läuft der Raspberry im Normalbetrieb ganz allein mit minimalem Hardware-Einsatz und somit geringstmöglichem Stromverbrauch als web-gesteuerter Logger.

Deshalb ist hier die Empfehlung : Raspberry Pi OS in der Lite-Version.

Es kann aber natürlich auch eine der Desktop Varianten verwendet werden, wobei die Installation grundsätzlich gleich läuft, nur das die Bedienung direkt an Bildschirm und Tastatur des Raspberry Pi's läuft.

Raspberry Pi OS ist das offizielle, von der Raspberry Pi Foundation bereitgestellte Betriebssystem, welches auf Debian Linux basiert und breit unterstützt wird.

4.1.2 Voraussetzungen für die SD-Kartenerstellung

Um eine SD-Karte für einen Raspberry Pi vorzubereiten ist natürlich ein Computer mit einem SD-Karten-Laufwerk notwendig – egal, ob intern oder extern. Außerdem muss eine Image-Datei des Betriebssystems per Internet heruntergeladen werden. Dafür muss der Rechner natürlich eine Internetverbindung haben.

(Ein Image – zu deutsch: Abbild – ist eine große Datei, die den vollständigen Inhalt eines Datenträgers als 1:1 Kopie enthält.)

Seit der Version des Raspberry Pi OS vom 4.4.2022 (Debian 11, bullseye) gibt es keinen vorinstallierten Standardbenutzer 'Pi' mehr.

Insbesondere für eine 'headless' Installation muss also für den ersten Systemstart ein Benutzer angelegt werden, damit ein Zugang über das Netzwerk überhaupt möglich ist.

Eine manuelle Installation ist zwar noch weiterhin möglich, erfordert aber unter Anderem einen weiteren Linux-Rechner (z.B. bereits laufender Raspberry) zur Erzeugung des Passwort-Hash (Fingerprint des verschlüsselten Passwortes) – sprich: Es ist nicht mehr wirklich einfach.

Für eine einfache Vorbereitung empfiehlt sich deshalb die Verwendung des Hilfsprogramms 'Raspberry Pi Imager', das von der Raspberry Pi Foundation bereitgestellt wird.

4.1.3 **Download** Raspberry Pi Imager

Auf der Software-Seite von <u>www.raspberrypi.org</u> gibt es die Möglichkeit ein komfortables Programm (Tool) namens 'Raspberry Pi Imager' für 3 verschiedene Betriebssysteme eines Hilfsrechners herunterzuladen oder mittels Befehlszeile auf einem bereits laufenden Raspberry Pi zu installieren :

taspberry Pi Imager	🚜 Raspberry Pi
aspberry Pi Imager is the quick and easy way to install aspberry Pi OS and other operating systems to a microSD ard, ready to use with your Raspberry Pi.	Rapberry Pl Device Operating System Biorage CHOOSE DEVICE CHOOSE OS CHOOSE STORAGE
lownload and install Raspberry Pi Imager to a computer vith an SD card reader. Put the SD card you'll use with our Raspberry Pi into the reader and run Raspberry Pi nager.	NOT
Download for Ubuntu for x86	
Iownload for Windows Iownload for macOS	
o install on Raspberry Pi OS , type	

Nach dem Download für das zutreffende System installieren wir die Software auf unserem Hilfsrechner.

Das hängt etwas von dem vorhandenen Betriebssystem ab, passiert aber in der Regel durch einen Doppelklick auf die gerade heruntergeladene Datei.

4.1.4 Raspberry Pi Imager verwenden

Wir starten den Raspberry Pi Imager und sehen dieses Startfenster:

8		Raspberry Pi Ima	ager v1.8.5	~ ^ X
	👸 Rasp	berry Pi		
	Raspberry Pi Modell	Betriebssystem (OS)	SD-Karte	
	MODELL WARLEN	US WAHLEN	SD-KARTE WARLEN	

Als Erstes müssen wir unter 'MODELL WÄHLEN' unseren vorhandenen Raspberry Pi auswählen. Das filtert für die nächste Auswahl nicht passende Optionen weg und beeinflusst die Empfehlung.

Im nächsten Auswahlfenster wählen wir nun eine der Varianten des Betriebssystems aus. Diese Auswahl erfolgt in Stufen:



Als erstes erscheint die Empfehlung für unser Modell (hier ein Raspi 3). Darunter, unter 'Raspberry Pi OS (other)' können wir die Alternativen finden:

	Betriebssystem (OS)	x
<	Zurück	
	Zuruck zum Hauptmenu	
	Raspberry Pi OS (Legacy, 32-bit) Lite	
X	A port of Debian Bullseye with security updates and no desktop environment	
W	Veröffentlicht: 2024-03-12	
	Online - 0.4 GB Download	
	Raspberry Pi OS (Legacy, 32-bit) Full	
8	A port of Debian Bullseye with security updates, desktop environment and recommended applications	
Y	Veröffentlicht: 2024-03-12	
	Online - 2.4 GB Download	
	Raspberry Pi OS (Legacy, 64-bit)	
X	A port of Debian Bullseye with security updates and desktop environment	

Da hier im Weiteren beispielhaft ein System ohne die grafische Oberfläche benutzt werden soll, ist die Wahl hier nicht der empfohlene erste Eintrag, sondern aus dem Untermenü darunter der Eintrag 'Raspberry Pi OS Lite (32-bit)'. Die 64 Bit Variante wäre auch möglich, braucht aber bei geringem Geschwindigkeitsvorteil mehr Arbeitsspeicher und etwas mehr Strom – Geschmackssache...

Natürlich lässt sich der PiLogger WebMonitor auch auf einem Desktop-System verwenden - also mit grafischer Oberfläche - das soll hier aber nicht das Ziel dieser Anleitung sein.

Der Raspberry Pi Imager lädt die neueste Version live von raspberrypi.org herunter – es wird also eine Internetverbindung benötigt.

Das heruntergeladene Image wird für weitere zu flashende SD Karten auf dem Rechner gespeichert.

Nun müssen wir noch das *richtige* Laufwerk mit der Micro-SD-Karte für den Raspi unter 'SD-KARTE WÄHLEN'.

Wenn wir nun auf 'WEITER' drücken, kommt diese Rückfrage:

Handbuch PiLogger One

8.	Raspberry Pi Imager v1.8.5	~ ^ X
8 Raspbe	erry Pi	
	OS Anpassungen anwenden?	x
Möchten Sie die	vorher festgelegten OS Anpassung	en anwenden?
EINSTELLUNGEN BEARBEITEN	NEIN, EINSTELLUNGEN LÖSCHEN	JA NEIN
		WEITER



Diese Einstellungen sollten immer bearbeitet / verwendet werden um einen Benutzer anzulegen und den WLAN Zugang einzurichten ! Insbesondere bei einer 'headless' Installation ist sonst kein Zugang möglich !

Wir gehen also auf 'EINSTELLUNGEN BEARBEITEN' und kommen in diese Maske:

All Hostname: pilo	local	
🖌 Benutzername u	nd Passwort festlegen	
Benutzername:	karl	_
Passwort:	•••••	
Wifi einrichten		
SSID: Ho	omenet	
Passwort:		
Passwort	anzeigen 🗌 Verborgene SS	ID
Wifi-Land: D	E 👻	
Spracheinstellun	gen festlegen	
Zeitzone:	Europe/Berlin 👻	
Tastaturlayout:	de 🗸	

Ausgabe V2.2 , 29.06.2024

Unter dem Reiter 'Allgemein' sind hier Netzwerkname (Hostname), Benutzer, Wifi (WLAN) Zugang und Spracheinstellungen festlegbar. Bitte alles mit den eigenen Werten ausfüllen :)

Unter dem Reiter 'Dienste' ist dann noch SSH (**S**ecure **Sh**ell) zu aktivieren, damit insbesondere bei einem 'headless' System der Remote-Zugriff möglich ist:

ALLGEMEIN	DIENSTE	OPTIONEN
SSH aktivieren		
Passwort zu	r Authentifizierung verwend	den
Authentifizie authorized_	rung via Public-Key keys für 'karl':	
SSH-KEYG	EN AUSFÜHREN	
	SPEICHERN	

Anschließend drücken wir auf 'SPEICHERN' und gelangen zurück zur Frage:

Aspberry Pi	
OS Anpassungen anwenden?	X
Möchten Sie die vorher festgelegten OS Anpassungen anwender EINSTELLUNGEN BEARBEITEN NEIN, EINSTELLUNGEN LÖSCHEN JA	NEIN
WEITER	

Ausgabe V2.2 , 29.06.2024

Die wir jetzt mit 'JA' beantworten. Nun werden wir noch gewarnt, dass alle Daten auf der SD Karte gelöscht werden – wenn es sich um den richtigen Speicher handelt, antworten wir nochmal mit 'JA' :



Daraufhin startet nun endlich der Schreibvorgang :

🕉 Raspberry Pi		
Raspberry Pi Modell RASPBERRY PI 3	Betriebssystem (OS) RASPBERRY PI OS LITE (32-BIT)	SD-Karte
	Schreiben 15%	SCHREIBEN ABBRECHEN

Nach dem Schreiben der Daten wird automatisch die Überprüfung gestartet :



Der erfolgreiche Abschluss wird mit diesem Fenster gemeldet :

Raspberry Pi OS Lite (32-bit) wurde auf I (bootfs, rootfs) geschrieben	Interner SD-Kartenleser
Sie können die SD-Karte nun aus dem Le	segerät entfernen
WEITER	

Die SD Karte kann nun in den Raspberry Pi eingesetzt werden.

4.1.5 Erstes Booten, SSH Verbindung

Nachdem nun die SD-Karte in den Raspberry Pi eingesetzt ist, ist es Zeit den Strom einzuschalten, um dem Raspberry Pi beim ersten Hochfahren (Booten) eine Weile zuzusehen – die SD-Karten-LED muss einige Zeit ganz hektisch blinken.

Der Raspberry meldet sich nun also beim ersten Booten am WLAN-Router (oder Access-Point) an. Dabei verwendet er den Netzwerk-Namen den wir in den besonderen Einstellungen im Imager angegeben haben.

Wir starten jetzt ein SSH-fähiges Terminal-Programm auf unserem anderen Rechner - z.B. das bewährte 'Putty'.

Dort geben wir für den zu kontaktierenden Raspberry Pi (jetzt von hier gesehen, der Host-Rechner) als Adresse den Netzwerknamen und Port 22 an. Die IP-Adresse, die der Router dem Raspberry zugewiesen hat, wird automatisch ermittelt. Es macht aber durchaus Sinn im Router einzustellen, dass der Raspberry immer genau diese Adresse erhalten soll.

Category:	Basic options for your PuTTY see	ssion
 Session 	Specify the destination you want to connect	t to
Logging	Host <u>N</u> ame (or IP address)	Port
 Terminal 	pilogger-test	22
Keyboard	Connection type:	
Bell	○ Ra <u>w</u> ○ <u>T</u> elnet ○ Rlog <u>i</u> n ● <u>S</u> SH	⊖ Se <u>r</u> ia
Features	Load, save or delete a stored session	
 Window 	Saved Sessions	
Appearance		
Behaviour	Default Settings	Load
Translation		
 Selection 		Save
Colours		Delete
Fonts		
 Connection 		
Data		
Proxy	Close window on e <u>x</u> it:	
Telnet	Always Never Only on cle	ean exit
Rlogin		
About	<u>O</u> pen	Cancel

Der Raspberry Pi sendet nun ein Zertifikat, das wir in Putty jetzt annehmen müssen, um sicher zu stellen, dass der entfernte Rechner (in unserem Fall der Raspberry) wirklich der echte gewünschte Verbindungspartner ist:

Handbuch PiLogger One



Diese Nachfrage ist wie das Überprüfen eines Personalausweises, eine sogenannte Authentifizierung. Sie taucht immer auf, wenn ein neuer oder geänderter SSH-Server-Schlüssel unter einer bestimmten MAC-Adresse auftaucht (Netzwerkgeräte-Schnittstellen-ID – nicht zu verwechseln mit der IP-Adresse). Das passiert beim ersten Kontakt mit einem bestimmten Gerät oder wenn das System auf diesem Gerät neu installiert wurde – wie in unserem Fall. Wir akzeptieren also mit 'Accept' und die Verbindung wird aufgebaut. Wir bekommen jetzt ein Terminal-Fenster mit Kommando-Zeilen-Eingabe (CLI command line interface).

Jetzt müssen wir uns einloggen:



Hier verwenden wir nun natürlich den Nutzernamen und das Passwort, die wir bei der SD-Kartenkonfiguration angegeben haben.

4.1.6 System einrichten

Normalerweise muss ein frisches Raspberry Pi OS nun einmal grundlegend konfiguriert werden. Weil wir aber schon einiges über die Konfiguration im 'Raspberry Pi Imager' erledigt haben, ist jetzt nicht mehr so viel zu tun. Zum Beispiel die Ländereinstellung ist bereits in den besonderen Einstellungen gesetzt worden – hat aber die Linux 'locale' nicht vollständig korrekt gesetzt. Für die letzten notwendigen Aktionen und um die Möglichkeiten zu zeigen, führen wir das Tool 'raspi-config' aus.

Dazu geben wir in der Kommando-Zeile folgendes ein:

```
sudo raspi-config
```

Das Programm startet dann mit diesem Bildschirm:

Raspberry Pi	3 Model B Plus Rev 1. Raspberry Pi Softw	4 ware Configuration Tool (raspi-config)	
1 2 3 4 5 6 8 9	System Options Display Options Interface Options Performance Options Localisation Options Advanced Options Update About raspi-config	Configure system settings Configure display settings Configure connections to peripherals Configure performance settings Configure language and regional settings Configure advanced settings Update this tool to the latest version Information about this configuration tool	
	<select></select>	<finish></finish>	

Die Bedienung dieses Tools funktioniert zunächst mit den Tasten 'Pfeil abwärts' und 'Pfeil hoch' um die Listenpunkte anzusteuern. Der ausgewählte Punkt kann dann mit der Taste 'Enter' (oder 'Return') aktiviert werden.

Mit der Taste 'Tab' können die Optionspunkte <Select> (Auswählen) und <Finish> (Beenden) angesteuert werden und dann wieder mit 'Enter' aktiviert werden.

4.1.6.1 Raspi-Config updaten

Als Erstes machen wir einen Abstecher zur Option 8 und versuchen ein Update des Tools selbst.

Dazu manövrieren wir mit der Taste 'Pfeil abwärts' auf die Zeile 8:

Raspberry Pi 3 Model B Plus Rev 1.	.4
Raspberry Pi Soft	ware Configuration Tool (raspi-config)
1 System Options 2 Display Options 3 Interface Options 4 Performance Options 5 Localisation Options 6 Advanced Options 8 Update 9 About raspi-config	Configure system settings Configure display settings Configure connections to peripherals Configure performance settings s Configure language and regional settings Configure advanced settings Update this tool to the latest version Information about this configuration tool
<select></select>	<finish></finish>

Hierfür ist eine funktionierende Internet-Verbindung notwendig. Nach einem 'Enter' springt die Anzeige wieder auf die Kommandozeile und das Update wird durchgeführt:



Ist das Update beendet, wartet das Script 5 Sekunden und startet 'raspi-config' erneut, so dass wir wieder im Hauptmenü auf Punkt 1 stehen.

4.1.6.2 System Options

Da wir jetzt wieder auf dem Punkt 1 'System Options' im Hauptmenü stehen, gehen wir mit einem Druck auf die 'Enter'-Taste in das Untermenü:

Raspberry Pi So	ftware Configuration Tool (raspi-config)
S1 Wireless LAN	Enter SSID and passphrase
S2 Audio	Select audio out through HDMI or 3.5mm jack
S3 Password	Change password for the 'pi' user
S4 Hostname	Set name for this computer on a network
S5 Boot / Auto Login	Select boot into desktop or to command line
S6 Splash Screen	Choose graphical splash screen or text boot
S7 Power LED	Set behaviour of power LED
S8 Browser	Choose default web browser
<select< td=""><td>> <back></back></td></select<>	> <back></back>

• S1 'Wireless LAN'

gibt uns die Möglichkeit die Log-In-Daten für das WLAN zu ändern. Da wir offenbar erfolgreich eingeloggt sind, ist dies jetzt nicht notwendig.

S2 'Audio'

gibt die Möglichkeit bei Raspberry Pi Modellen mit analoge Audiobuchse den Audioausgang zu wählen (oder über HDMI). Beim Modell Zero gibt es die nicht.

• S3 'Password'

gibt die Möglichkeit für den aktuellen Benutzer das Passwort zu ändern. Sollte zur Zeit nicht notwendig sein.

- S4 'Hostname' ist der Netzwerkname des Raspi. Hatten wir ja auch frisch festgelegt.
- S5 'Boot / Auto Login' ist eigentlich nur für Systeme mit Display und Tastatur interessant.
- S6 'Splash Screen' ist eine Einstellmöglichkeit für Systeme mit Display.
- S7 'Power LED' ermöglicht auf einigen Raspberry Pi Modellen das Verhalten der Power LED zu ändern.
- S8 'Browser'

lässt uns wählen, ob wir statt 'Chromium' lieber 'Firefox' als Standard-Browser setzen möchten.

Also ein paar interessante Optionen, die nach Wunsch geändert werden können, aber für den PiLogger WebMonitor jetzt nicht geändert werden brauchen.

4.1.6.3 Display Options

Im Hauptmenü wählen wir nun die Zeile 2 'Display Options' und bekommen dieses Untermenü:

 Raspberry Pi Softwar	re Configuratior	n Tool (raspi-config)	
D1 Underscan D2 Screen Blanking D3 VNC Resolution D4 Composite	Remove black bo Enable/disable Set resolution Enable/disable	order around screen screen blanking for headless use composite output	
<select></select>		<back></back>	

Diese Optionen betreffen allesamt mögliche Display-Einstellungen.

Da wir hier eine 'Lite' Variante (ohne Desktop) aufgesetzt haben, sind diese Optionen hier nicht relevant.

Mit der 'Tab' Taste gehen wir auf <Back> und mit 'Enter' kehren wir in das Hauptmenü zurück.

4.1.6.4 Interface Options

Im Hauptmenü wählen wir den Punkt 3 'Interface Options' aus. Nach dem Bestätigen mit der 'Enter'-Taste öffnet sich das Untermenü:

Ra	spberry Pi Software	e Configuration Tool (raspi-config)
I1 SSH	Enable/disabl	le remote command line access using SSH
I2 VNC	Enable/disabl	le graphical remote desktop access
I3 SPI	Enable/disabl	le automatic loading of SPI kernel module
I4 I2C	Enable/disabl	le automatic loading of I2C kernel module
I5 Serial	Port Enable/disabl	le shell messages on the serial connection
I6 1-Wire	Enable/disabl	le one-wire interface
I7 Remote	GPIO Enable/disabl	le remote access to GPIO pins
	<select></select>	<back></back>

Von den vielen unterschiedlichen Optionen hier interessieren uns im Rahmen dieser Anleitung zunächst zwei. Unter I1 'SSH' kann der SSH Server aktiviert werden. Das haben wir aber bereits erfolgreich mit der SD-Karten-Vorbereitung gemacht.

Wir müssen aber noch mit I4 'I2C' die I²C Schnittstelle aktivieren. Diese ist für den PiLogger notwendig.

4.1.6.4.1 I4 I2C

Zum Aktivieren der I²C Schnittstelle navigieren wir in die Zeile I4:



Nach einem Druck auf 'Enter' werden wir gefragt ob wir das 'ARM I2C interface' aktivieren wollen:

Would you like the	e ARM I2C interface	e to be enabled?
<ye:< td=""><td>5<mark>2</mark></td><td><no></no></td></ye:<>	5 <mark>2</mark>	<no></no>

Die Voreinstellung ist <No>, weil diese Schnittstelle standardmäßig deaktiviert ist. Wir wechseln also mit der 'Pfeil links'-Taste auf <Yes> und drücken auf 'Enter'. Danach bekommen wir die Bestätigungsmeldung:



Nach einem Druck auf 'Enter' werden wir wieder in das Hauptmenü versetzt.

4.1.6.5 Performance Options

Unter dem Menüpunkt 4 können Einstellungen für Leistung des Raspberry vorgenommen werden.

Wir navigieren also auf die Zeile 4 und drücken die 'Enter'-Taste bekommen wir dieses Untermenü:

	Raspberry Pi	Software	Configuration	Tool	(raspi-config)	
	2 Overlay File	e System H	Enable/disable	read-	-only file system	m
-	-	-				
	<sele< td=""><td>ect></td><td></td><td><</td><td>(Back></td><td></td></sele<>	ect>		<	(Back>	

Dieses Menü ermöglicht je nach Modell verschiedene Einstellungen des Raspberry Pi rund um das Thema 'Leistung'. Tatsächlich haben wir hier im Moment nichts zu tun.

Mit zweimal 'Tab' springen wir zu <Back> und mit 'Enter' springen wir wieder zurück zum Hauptmenü.

4.1.6.6 Localisation Options

Wir gehen nun auf die Zeile 5 'Localisation Options'. Nach dem Auswählen mit der 'Enter'-Taste bekommen wir dieses Untermenü:



Wie oben bereits erwähnt, funktioniert die Länderauswahl im 'Imager' nicht vollständig. Die Punkte 'Timezone' und 'WLAN Country' sind allerdings erledigt. Wir müssen uns hier also um L1 und L3 kümmern.

4.1.6.6.1 L1 Locale

Die Option L1 ermöglicht das Einrichten der länderspezifischen Einstellungen des Systems, also die Auswahl des Zeichensatzes, der Sprache und der Anzeigenformate wie etwa Datum und Uhrzeit. Diese Ländereinstellungen werden 'Locales' genannt.

Standardmäßig ist lediglich Englisch sowie ein Basissatz eingerichtet. Nach einem Druck auf 'Enter' bekommen wir eine lange Liste von möglichen Ländereinstellungen zur Auswahl angeboten:

Locales are a framework to sw language, country, characters Please choose which locales t particularly for new installa compatibility with older syst Locales to be generated: All locales All locales aDJ.UTF-8 UTF-8 aER UTF-8 bER	Configuring locales itch between multiple languages and allow users to use their , collation order, etc. o generate. UTF-8 locales should be chosen by default, tions. Other character sets may be useful for backwards ems and software.
[] anp_IN UTF-8 [] ar_AE.UTF-8 UTF-8 [] ar_BH.UTF-8 UTF-8 [] ar_DZ.UTF-8 UTF-8 [] ar_EG.UTF-8 UTF-8 [] ar_IN UTF-8	
<0k>	<cancel></cancel>

Grundsätzlich können entweder alle (All locales) oder einzelne zur weiteren Einrichtung ausgewählt werden. Alle auszuwählen würde nicht nur eine lange Einrichtungszeit bedeuten, es würde auch im Betrieb unnötig Ressourcen binden und einige Vorgänge verlangsamen.

Hier empfiehlt es sich genau eine weitere Landeseinstellung hinzuzufügen – in unserem Fall 'de_DE.UTF-8 UTF-8'.

Das ist die Sprache 'Deutsch' für das Land 'Deutschland' mit der Zeichensatz-Kodierung UTF-8.

Zu diesem Eintrag navigieren wir mit der 'Pfeil abwärts'-Taste:

Handbuch PiLogger One



Das Auswählen geschieht hier mit der Leertaste (Space), woraufhin der Eintrag ein Sternchen für den Status 'gewählt' bekommt:

Genauso können jetzt noch weitere Landessätze bei Bedarf ausgewählt werden. Dies ist, wie gesagt, in unserem Fall nicht notwendig.

Wir gehen also nun mit der **'Tab'-Taste** auf das Aktionsfeld <Ok>, welches dabei rot hinterlegt wird:

Configuring locales Locales are a framework to switch between multiple languages and allow users to use their language, country, characters, collation order, etc. Please choose which locales to generate. UTF-8 locales should be chosen by default, particularly for new installations. Other character sets may be useful for backwards compatibility with older systems and software. Locales to be generated:	ackage configuration	
[] crh_UA UTF-8 [] csb_CZ.UTF-8 UTF-8 [] csb_PL UTF-8 [] cv_RU UTF-8 [] cv_RU UTF-8 [] da_DK.UTF-8 UTF-8 [] da_AT.UTF-8 UTF-8 [] da_CH.UTF-8 UTF-8 [] da_LI.UTF-8 UTF-8 [] da_LI.UTF-8 UTF-8 [] da_LI.UTF-8 UTF-8 [] da_LI.UTF-8 UTF-8 [] da_DDE UTF-8 [] doi_IN UTF-8 [] doi_IN UTF-8 [] dx_MV UTF-8 [] dx_MV UTF-8 [] dx_MV UTF-8	Locales are a framework to sw language, country, characters Please choose which locales t particularly for new installa compatibility with older syst Locales to be generated:	Configuring locales itch between multiple languages and allow users to use their , collation order, etc. o generate. UTF-8 locales should be chosen by default, tions. Other character sets may be useful for backwards ems and software.
< <u>Ck></u> <cancel></cancel>	<pre>[] crh_UA UTF-8 [] cs_CZ.UTF-8 UTF-8 [] csb_PL UTF-8 [] cv_RU UTF-8 [] cy_GB.UTF-8 UTF-8 [] da_DK.UTF-8 UTF-8 [] de_AT.UTF-8 UTF-8 [] de_BE.UTF-8 UTF-8 [] de_CH.UTF-8 UTF-8 [] de_LI.UTF-8 UTF-8 [] de_LI.UTF-8 UTF-8 [] de_LI.UTF-8 UTF-8 [] doi_IN UTF-8 [] doi_NUTF-8 [] dv_MV UTF-8 [] dv_MV UTF-8</pre>	
	KOk>	<cancel></cancel>

Mit einem Druck auf 'Enter' wird diese Eingabe nun abgeschlossen.

Das nächste Fenster fordert uns nun auf die bevorzugte Ländereinstellung unter den aktivierten auszuwählen:



Wir wählen also 'de_DE.UTF-8' mit der 'Pfeil abwärts'-Taste, gehen auf <Ok> mit der 'Pfeil rechts'-Taste und Drücken 'Enter'.

Jetzt wechselt die Anzeige wieder auf das Konsolenfenster und die Verarbeitung, das heißt das Einrichten, dauert eine kleine Weile.

Danach springt die Anzeige wieder in das Hauptmenü von 'Raspi-Config'.

4.1.6.6.2 L3 Keyboard

Wir gehen also noch einmal in das Menü 5 'Localisation Options'. Mit dem Menüpunkt L3 wird das Tastatur-Layout den bisher gemachten Ländereinstellungen angepasst. Dazu müssen wir diese Option einmal ausführen. Wir navigieren also wie gewohnt auf den Punkt L3:



Wir starten diese Aktion also mit einem 'Enter'-Tastendruck und werden auf die Konsolen-Anzeige geschaltet, wo die Meldung 'Reloading keymap. This may take a short while' um etwas Geduld bittet. Danach werden wir wieder auf das Hauptmenü zurückgesetzt.

4.1.6.7 Advanced Options

Auch wenn wir in diesem Untermenü derzeit keine Notwendigkeit haben etwas zu tun, werfen wir einen Blick auf die Möglichkeiten - zur Information:

Raspberry P	Pi Software Configuration Tool (raspi-config)
A1 Expand Filesyster A2 Network Interface A3 Network Proxy Set A6 Wayland	em Ensures that all of the SD card is available ce Names Enable/disable predictable network i/f names ettings Configure network proxy settings Switch between X and Wayland backends
A7 Audio Config	Set audio control system
<sele< td=""><td>ect> <back></back></td></sele<>	ect> <back></back>

Der Punkt A1 'Expand Filesystem' war bei älteren Raspi OS sinnvoll – sollte aber jetzt während des ersten Boot-Vorgangs automatisch erledigt worden sein. Die anderen Optionen beschäftigen sich mit Desktop-Grafik und Netzwerk-Einstellungen, die in Spezialfällen wichtig sein können.

Wir sind hier fertig und wechseln mit 'Tab' auf <Back> und kehren mit 'Enter' zurück ins Hauptmenü.

4.1.6.8 Raspi-Config beenden

Da wir jetzt alle notwendigen Einstellungen vorgenommen haben, können wir das Konfigurationsprogramm beenden. Dazu wechseln wir mit der 'Pfeil rechts'-Taste auf das Aktionsfeld <Finish> und drücken 'Enter'.

Für die weiteren Aktionen ist es sinnvoll einen Neustart durchzuführen. Falls nicht durch die vorgenommenen Änderungen von Raspi-Config selber ein Neustart angeboten wird, machen wir das jetzt selber direkt. Wir geben also im Konsolenfenster ein:

sudo reboot now

Die Eingabe schließen wir mit 'Enter' ab. Da der Raspberry jetzt herunterfährt um neu zu starten, wird die Verbindung zu Putty getrennt:

Wir müssen jetzt eine Weile warten, bis der Raspberry neu gestartet ist und können uns dann mit Putty wieder neu verbinden.

4.1.7 Allgemeines Update

Um die neue Raspberry Pi OS-Installation komplett zu machen, sollte noch ein normales Update durchgeführt werden. Je nachdem, wie alt das Image ist, sind in der Zwischenzeit weitere Verbesserungen durchgeführt worden, die teilweise auch sicherheitsrelevant sein können. Diese werden so schnell wie möglich verteilt, also auch außer der Reihe von kompletten Installations-Images.

Die folgenden Aktionen setzen eine funktionierende Internet-Verbindung voraus.

4.1.7.1 Firmware Update

Ein Firmware Update wird bei den neueren Versionen von Raspberry Pi OS *nicht* mehr empfohlen.

Das Firmware Update soll nur noch in Ausnahmefällen durchgeführt werden:



Tatsächlich wird bei einem Distributions-Update nun auch die Firmware upgedatet, falls notwendig.

Wir überspringen also diesen Schritt und fahren fort mit dem Distributions-Update.

4.1.7.2 Distributions-Update

Nun kann ein allgemeines Upgrade der Raspberry Pi OS-Distribution vorgenommen werden. Dazu muss die lokale Datenbank der installierten und verfügbaren Software-Pakete aufgefrischt werden.

Wir geben also folgende Zeile ein:

```
sudo apt-get update
```

Hier wird mit 'super user'-Rechten das Tool 'apt-get' aufgerufen und angewiesen

ein Update der Paket-Datenbank auszuführen. Dieser Vorgang dauert nicht lange. Die Ausgabe sieht etwa so aus:



Nun geben wir die folgende Textzeile ein:

sudo apt-get dist-upgrade

Damit wird jetzt das tatsächliche Update der Raspberry Pi OS-Installation gestartet. Zunächst wird ermittelt, welche Pakete geändert werden müssen. Dann wird abgefragt, ob diese Änderungen tatsächlich installiert werden sollen:



Nach der Eingabe von 'J' mit Abschluss durch 'return' beginnt der Hauptteil des Updates. Je nachdem wie frisch das verwendete Raspberry Pi OS Image ist, müssen eventuell etliche Pakete upgedatet werden. Das kann dann schon mal bis zu einer Stunde dauern. Das gezeigte Beispiel ist mit einem 1 Monat alten Image gemacht, mit relativ überschaubarem Umfang (30 zu aktualisierende Pakete).

Das Update ist vollständig durchgeführt, wenn der Kommando-Zeilen-Prompt wieder erscheint. Es sollte nun einmal neu gestartet werden:

sudo reboot

Damit ist der Raspberry Pi grundsätzlich eingerichtet.

4.2 Montage und Inbetriebnahme

Der PiLogger One ist ursprünglich genau passend für einen Raspberry Pi 1 entwickelt worden. Er hat deshalb eine 26 polige Buchsenleiste und passt neben die Cinch-Buchse für Audio des Raspberry Pi 1.

Die neueren Raspberry Pi bekamen dann eine 40 polige Stiftleiste für die GPIO's. Da die Stifte, die der PiLogger One benötigt, auch weiterhin kompatibel belegt zur Verfügung stehen, ist der PiLogger One auch für diese jüngeren Raspberry Pi's verwendbar.

Seit kurzem gibt es eine Variante des PiLogger One, die auf die 26 polige Buchsenleiste verzichtet und stattdessen eine 5 polige Stiftleiste zum Anschluss besitzt. Die Verbindung zum Raspberry Pi erfolgt dann über ein kurzes 5 poliges Dupont-Kabel (Buchse-Buchse). Diese Variante heißt 'PiLogger One Typ B'. Wir nennen also nachfolgend die ursprüngliche Variante 'PiLogger One Typ A' – zur besseren Unterscheidung.

4.2.1 Montage PiLogger One Typ A

4.2.1.1 Vorbereitung

Das PiLogger One Typ A Modul ist sowohl für den Raspberry Pi 1 mit einer 26 poligen Stiftleiste verwendbar, als auch für die neueren Raspberry Pi's mit der 40 poligen Stiftleiste.

Bei der Verwendung mit einem Raspberry Pi mit 26 poliger Stiftleiste kann nur der Anschraubpunkt nahe der Schraubklemmenleiste verwendet werden, die beiden herausstehenden Ecken mit Anschraublöchern sind hier ohne Verwendung.

Für Raspberrys mit 40 poliger Stiftleiste sollten alle 3 Befestigungslöcher verwendet werden. Dabei gibt es für den mittleren Abstandshalter auf dem Raspberry Pi zwar kein passendes Loch, jedoch ist an dieser Stelle kein Bauteil auf der Oberseite, dass beschädigt werden könnte. So kann sich dort der dritte Nylon-Bolzen gut abstützen. (*Bitte zur Sicherheit bei dem tatsächlich vorliegenden Pi-Exemplar noch einmal überprüfen !*)

In diese Löcher des PiLogger One werden nun die beigelegten Abstandsbolzen eingesetzt. Dabei wird jeweils das Gewindeende direkt von unten in das Loch im PiLogger-Modul gesteckt:

Handbuch PiLogger One



Danach wird der Bolzen im Loch festgehalten und mit dem Modul umgedreht. Jetzt kommt zuerst eine Unterlegscheibe auf den Gewindezapfen, danach eine Sechskantmutter, die zunächst lose aufgeschraubt wird, bis der Bolzen nicht mehr wackelt, aber noch verschiebbar ist.

Für das mittlere Loch sieht das dann von oben so aus:



Und für alle 3 von der Seite so:



Nun kann das Modul aufgesteckt werden.

4.2.1.2 Modul aufsetzen

Das PiLogger Modul wird nun auf den Raspberry Pi aufgesetzt. Dazu muss mindestens die Versorgung aus gesteckt sein – besser ist es alle Kabel und angesteckte Peripherie hierfür zu entfernen.

Die 26 polige Buchsenleiste muss dabei genau auf die Stifte des Raspberry Pi so aufgesetzt werden, dass die ersten Stifte an der Außenecke des Raspberry Pi mit den ersten Buchsen an der Ecke mit dem Befestigungsloch des PiLoggers fluchten (1 auf 1).



Hier zuerst für einen Raspberry Pi 1:

Und hier für einen Raspberry Pi 2 B:



Jetzt muss die PiLogger Leiterplatte vorsichtig gleichmäßig heruntergedrückt werden, bis die Abstandsbolzen auf dem Raspberry Pi aufsetzen.

Das sieht dann beim Raspberry Pi 1 so aus:



Und beim Raspberry Pi 2 B sieht es so aus:



Das heißt: Die Buchsenleiste sitzt nicht ganz auf dem Sockel der Stiftleiste auf. Die Kontakte sind jedoch gut im Eingriff.

Nun drehen wir das Doppelpack auf den Kopf. Auf der Unterseite des Raspberry Pi legen wir auf jedes Loch, bei dem ein Abstandsbolzen dahinter sitzt, eine Unterlegscheibe.


Anschließend drehen wir jeweils eine Zylinderkopfschraube lose ein, so dass bei 2 Abstandsbolzen auch beide sauber fluchten – ohne Biegen. Danach drehen wir sie nacheinander mit einem Schraubendreher handwarm fest:



Das PiLogger Modul ist nun montiert.

Das Doppelpack kann jetzt wieder normal auf den Tisch gelegt werden.

4.2.2 Anschluss PiLogger One Typ B

Der PiLogger One Typ B sieht so aus:



Er wird ohne Abstandsbolzen, aber mit einem kurzen Anschlusskabel geliefert. Die 5 polige Stiftleiste ist wie folgt belegt:



Es empfiehlt sich das Kabel mit der roten Ader für die Versorgungsspannung 3,3 V und mit der schwarzen Ader für GND (Masse) zu verwenden. Eine Buchsenleiste des Kabels stecken wir also auf die Stiftleiste des PiLogger und die andere auf die Pins 1, 3, 5, 7 und 9 der Raspberry Pi Stiftleiste. Mit einem Raspberry Pi 4 sieht das Ganze dann so aus:





4.2.3 Erstes Einschalten

Vor dem ersten Einschalten des Raspberry Pi mit PiLogger müssen natürlich alle benötigten Kabel und Peripherie-Teile wie Monitor und Tastatur angesteckt werden.

Wir stecken nun als Letztes das Netzteil in die Steckdose – und neben dem normalen Booten des Raspberry Pi startet der PiLogger sofort mit einem kurzen Blitzen seiner LED einmal pro Sekunde.

Das ist die Voreinstellung ab Werk und zeigt, dass er korrekt arbeitet! Immer wenn der Raspberry Pi mit Strom versorgt ist, arbeitet der PiLogger autark mit seinen gespeicherten Parametern.

Hier als Beispiel der PiLogger One Typ A auf einem Raspberry Pi 3 :



Tastatur

4.3 PiLogger WebMonitor installieren

Die Installation des PiLogger WebMonitor wird durch ein interaktives Installations-Programm (Shell-Skript) – dem 'Installer' durchgeführt.

Der Installer lässt sich einfach, wie unten beschrieben, mit einer Zeile in der SSH-Konsole starten. Da dies eine ausführbare Datei ist, die aus dem Internet geladen wird, kann man den Installer auch separat von unserem Server herunterladen, begutachten und manuell starten:

https://www.pilogger.de/index.php/de/download-de/category/2-software

→ Eintrag : Installer PiLo-WebMon

Der Installer führt alle notwendigen Schritte aus und bietet optionale Einstellungen.

Dazu gehört auch die Möglichkeit die Uhrzeit-Synchronisation mit einem Internet-Zeitserver umzulenken auf den eigenen Heimnetzrouter. Diese

Zeitsynchronisation ist notwendig, weil der Raspberry Pi keine eigene Echtzeituhr (RTC) besitzt.

Allerdings ist dies die einzige externe Internet-Verbindung die benötigt wird und so verhindert, dass dem Raspberry Pi der Zugriff auf das öffentliche Internet im Router gesperrt werden kann. Wenn der eigene Router die Möglichkeit bietet im Heimnetz einen NTP-Server bereitzustellen, kann auch diese Verbindung intern gehalten werden und der Internet-Zugang für den Raspberry kann gesperrt werden. Deshalb wird als nächstes diese optionale Vorbereitung beschrieben.

4.3.1 Router als NTP-Server einrichten

Internet-Router synchronisieren ihre eigene Zeit mit NTP-Servern im Internet. Bei Telefonie per VOIP stellt in der Regel der Anbieter seine eigenen Zeitserver im Router ein. Bei den meisten WLAN-Routern lässt sich zusätzlich einstellen, dass der Router selber als Zeitserver im eigenen Netzwerk zur Verfügung steht. Als Beispiel zeigen wir hier die Einstellungsseite einer Fritz-Box. Hier meldet man sich auf der Konfigurationsseite des Routers an und navigiert über den Menüpunkt 'Heimnetz' in das Untermenü 'Netzwerk'. Dort wechselt man in das zweite 'Tab' (Karteikarte) 'Netzwerkeinstellungen'.

Hier gibt es, ziemlich weit unten, die Möglichkeit '**weitere Einstellungen**' auszuklappen. Dort versteckt sich die Rubrik 'Zeitsynchronisation' und dort die Einstellmöglichkeit "Fritz!Box als Zeitserver im Heimnetz bereitstellen". Diese Option aktivieren wir und drücken anschließend auf den Knopf unten 'Übernehmen':

	FRITZ!	FRITZ!Box 7490 MyFritzi Fritzinas :
① ③ く 子	Übersicht Internet Telefonie Heimnetz	Heimnetz > Netzwerk Netzwerkverbindungen Netzwerkeinstellungen Der umersam rug wing (or m), windermet einfluss auf Sicherheitseinstellungen der FRITZIBox.
	Mesh Netzwerk	Zeitsynchronisation Die FRITZ!Box synchronisiert ihre Uhrzeit mit dem angegebenen Zeitserver im Internet.
•1)]	USB / Speicher Mediaserver FRITZ!Box-Name WLAN	Zeitserver ntp0.voip.telefonica.de; ntp1.voip.telefo Die FRITZ!Box kann als Zeitserver im Heimnetz verwendet werden. FRITZ!Box als Zeitserver im Heimnetz bereitstellen
() () ()	Smart Home Diagnose System	DNS-Rebind-Schutz Ihre FRITZIBox unterdrückt DNS-Antworten, die auf IP-Adressen im eigenen Heimnetz verweisen (DNS-Rebind-Schutz). Hier können Sie Ausnahmen angeben, für die der DNS-Rebind-Schutz nicht gelten soll. Tragen Sie da Übernehmen ^{Ste} Verwerfen Hostnamen (Domainname inklusive Subdomain) in die Liste ein.

Das war's.

4.3.2 Installer starten

Um die Installation zu starten, nehmen wir mit dem SSH Terminal-Programm (hier 'Putty') Verbindung zum Raspberry auf.

Für die Installation braucht der Raspberry eine funktionierende Internet-Verbindung.

In der Kommando-Zeile geben wir dann exakt diese Zeile ein (am besten hier kopieren) :

```
curl -sSL https://www.pilogger.de/get/installer-webmon | bash
```

Die Eingabe schließen wir mit 'Enter' ab.

```
login as: karl
karl@192.168.178.60's password:
Linux pilogger-test 6.6.28+rpt-rpi-v7 #1 SMP Raspbian 1:6.6.28-1+rpt1 (2024-04-22) armv71
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Apr 29 03:16:28 2024 from 192.168.178.29
karl@pilogger-test:~ $ curl -sSL https://www.pilogger.de/get/installer-webmon | bash
Dieses Script installiert die PiLogger WebMonitor Software
im Verzeichnis /home/karl/pilogger
```

Das Hilfsprogramm 'curl' lädt das Script von der PiLogger.de Website und startet seine Ausführung. Um einen Abbruch bei versehentlichem Aufruf zu ermöglichen, muss der Installationsstart bestätigt werden – mit der Taste 'j'.

Danach werden wir gefragt, ob wir den Heimnetz-Router als Zeitserver verwenden wollen:



Wenn wir den Router wie in 4.3.1 entsprechend eingerichtet haben, können wir hier 'j' antworten.

Als Nächstes wird ein alternativer I²C Treiber aktiviert.



Die I²C Hardware-Module der Broadcom Prozessoren haben Fehler, weshalb mit den Standardtreibern das I²C Bus Verhalten 'Clock Stretching', (also der Busmaster Raspi wartet auf das Peripheriegerät, das um kurze Pause bittet) nicht funktioniert. Der PiLogger und andere I²C Module brauchen aber dieses Feature.

Deshalb wird hier ein I²C-Modul installiert, das den I²C Port in Software auf den richtigen Pins nachbildet. Das funktioniert für alle Raspberry Pi's von Zero bis Modell 5 sehr gut.

Danach werden automatisch diese Aktionen durchgeführt:

- Installieren von 'python3-smbus' (Python Erweiterung für die I²C Schnittstelle)
- Installieren von 'python3-rpi.gpio'
 (Python Erweiterung f
 ür den direkten GPIO Zugriff)
- Installieren von python3-gpiozero (neue Python Erweiterung f
 ür den direkten GPIO Zugriff)
- Installieren von 'python3-bottle' (WSGI Micro Web Framework f
 ür Python)

- Herunterladen und Entpacken der PiLogger WebMonitor Software-Komponenten
- Einrichten des Autostarts der WebMonitor Software

Im Anschluss wird noch eine Option angeboten, mit der die täglichen Kontakte zu den Raspberry Pi OS-Update-Servern abgeschaltet werden können:



Das Abschalten ist nicht unbedingt notwendig, wenn dem Raspberry der Internetzugang gesperrt wird, weil die Anfragen dann einfach ins Leere gehen. Wenn der Internetzugang nicht gesperrt wird, erzeugen diese Anfragen eine zunächst harmlose Nutzerstatistik, machen andererseits aber auch nur wirklich Sinn auf einem Raspberry der noch andere Internet-Aufgaben hat – zum Beispiel auf einem Desktop-System (auf dem dann bei den neueren Raspberry Pi OS automatische Updates ausgeführt werden).

Ansonsten kann der Raspberry völlig privat im eigenen Heimnetz als Logger arbeiten und bietet ohne diese Brotkrumen auch keine besondere Angriffsfläche und somit keinen dringenden Bedarf an Updates – einen guten und sicheren Router vorausgesetzt.

Wenn diese letzte Wahl mit 'j' beantwortet wird, sieht das Ergebnis etwa so aus:



Damit ist die Installation komplett und mit einem Neustart des Raspberry kann der PiLogger WebMonitor gestartet werden.

Wir bestätigen also mit einem 'j' – und der Raspberry beendet die Verbindung und startet neu. Nach diesem Neustart können wir dann mit einem Web-Browser die erste Verbindung zum WebMonitor aufnehmen.

Die Bedienung der WebMonitor Software wird in Kapitel 6 beschrieben.

5 Installation WebMonitor auf ESP32 Modul

Der PiLogger One kann jetzt auch mit einem ESP32 Modul von Espressif Systems betrieben werden.

Dazu haben wir den PiLogger Web-Monitor entsprechend angepasst (portiert). Das Archiv mit allen benötigten Dateien findet ihr hier : <u>PiLogger Web-Monitor</u> <u>ESP32</u>



Voraussetzung für den erfolgreichen Logger-Betrieb ist eine schnelle SD Card als Datenspeicher. Da der ESP32 üblicherweise ohne SD-Card-Halter kommt und verfügbare SD card shields dann die SPI Schnittstelle mit nur 1 Datenleitung benutzen, haben wir für den PiLogger One eine spezielle Adapterplatte entworfen, die die SDIO Schnittstelle des ESP32 mit 4 Datenleitungen benutzt. Der Adapter ist in unserem Shop erhältlich : <u>Adapterplatte kaufen</u>

Der PiLogger Web-Monitor ESP32 basiert auf <u>MicroPython</u> von Damien George. Das bedeutet, dass zunächst die Firmware des ESP32 Moduls mit dem ESP32 port von MicroPython überschrieben werden muss. Die notwendige .bin Datei ist im Download-Archiv enthalten.

Danach kann der ESP32 direkt auf der Kommandozeile des Python-Interpreters angesprochen werden (REPL).

Deshalb empfehlen wir für die weiteren Schritte Thonny als komfortable

Entwicklungsumgebung. Thonny kann auch als GUI für das Espressif Flash-Tool esptool verwendet werden.

In den folgenden Kapiteln wird die komplette Installation Schritt für Schritt beschrieben.

5.1 Adapter und Modul verlöten

Die Adapterplatte hat auf der Seite, auf die das ESP32-DevKit-C Modul gesteckt werden muss, weiße Markierungen für die Micro-USB-Buchse und das ESP32-Kernmodul. Damit wird angezeigt, wie herum das Modul gesteckt werden muss:



Zusammengesteckt sieht das dann so aus:

4	ØC		000						0	•	
100	ee cuo-	🤹 😨 DS D3	😵 🔮 E: ONS		23 9	SP S	🛞 🛞 85 33	🖑 🖑	NO dO		
								FOCIDI2ACTZ-ESP32WR00		WITET ESP32-WROOM	
CL	K CO	BOOT 15	2 0	4 1e	5 17 e	5 18	19 GND	MR 20000 21 RX	TX 22	23 GNQ	
ູ ເ			0		d (11) (1	e w	6	egi egi e	9 69 6 10 69 6		

Jetzt muss diese Kombination umgedreht werden, so dass die durchgesteckten Stifte, die auf der anderen Seite freiliegen, verlötet werden können.



Fertig verlötet sieht das dann so aus :



5.1.1 Adapter mit NodeMCU ESP32 Modul

Der PiLogger Adapter für ESP32 Module lässt sich auch mit einem sogenannten NodeMCU Modul verwenden. Dieses Modul hat ebenfalls 38 Pins, die die gleiche Belegung haben. Das Modul sieht so aus:



Allerdings gibt es ein Detail, welches eine kleine Anpassung notwendig macht: Bei diesem Modul gibt es an GPIO G2 einen Pull-Down (5,1 k Ω Widerstand nach GND), der den Betrieb der SD-Card blockiert.

Dieser Widerstand muss entfernt oder wie hier im Bild 'hoch' gelötet werden:



Ausgabe V2.2 , 29.06.2024

5.1.2 Belegung optionale Stiftfelder

Die nicht verwendeten Pins des ESP32 Moduls sind auf 2 nachbestückbare Stiftleisten herausgeführt. J4 ist eine 12 polige, einreihige Stiftleiste, J5 ist eine 8 polige, zweireihige Stiftleiste.

In die Lötaugen der unbestückten, optionalen Stiftleisten können natürlich auch einzelne Kabel eingelötet werden.

Die Belegung dieser Stiftleisten zeigt dieses Bild :



5.2 Installation Thonny

Auf der Homepage von <u>Thonny</u> findet ihr die für euren PC passende Installationsvariante. Thonny kann bei Bedarf Python gleich mitbringen. Nach erfolgreicher Installation - einmal Starten bitte - muss Thonny nun auf den ESP32 eingestellt werden. Dazu bitte unter Extras' auf 'Optionen...' gehen. Hier auf den Karteireiter 'Interpreter' wechseln. Jetzt in der Ausklappliste nach 'MicroPython (ESP32)' suchen und auswählen.

Die USB Einstellung 'automatisch' unter 'Port oder WebREPL' wählen und den Rest erst mal so lassen – 'OK'.

Nun könnt ihr den ESP32 mit einem MicroUSB-Kabel an einen USB Port eures PC anschließen.

Die rote LED zeigt an, dass das Modul Strom hat - der Strombedarf ist niedrig genug (< 250 mA) um vom PC versorgt zu werden. Jetzt in Thonny einmal auf den roten Stop/Restart-Knopf drücken und anschließend am Modul auf den Reset-Knopf. Ein frisches Modul sollte sich dann so melden:

ets Jul 29 2019 12:21:46									
<pre>rst:0x1 (POWERON_RESET), boot:0x13 (SPI_FAST_FLASH_BOOT) configsip: 0, SPIWP:0xee clk_drv:0x00, q_drv:0x00, d_drv:0x00, cs0_drv:0x00, hd_drv:0x00, wp_drv:0x00 mode:DIO, clock div:2 load:0x3fff0018, len:4 load:0x3fff001c, len:5564 load:0x40078000, len:0 load:0x40078000, len:13756</pre>									
entry 0x40078fb4									
I (29) boot: ESP-IDF v3.0.3 2nd stage bootLoader									
I (29) boot: compile time 08:53:32									
1 (29) boot: Enabling RNG early entropy source									
I (34) boot: SPI Speed : 40MHz									
I (38) boot: SPI Mode : DIO									
I (42) boot: SPI Flash Size : 4MB									
I (46) boot: Partition Table:									
I (49) boot: ## Label Usage Type ST Offset Length									
I (57) boot: 0 phy_init RF data 01 01 0000f000 00001000.									
I (64) boot: 1 otadata									
I (72) boot: 2 nvs WiFi data 01 02 00012000 0000e000									
I (79) boot: 3 at_customize unknown 40 00 00020000 000e0000									
I (87) boot: 4 ota_0 OTA app 00 10 00100000 00180000									
I (94) boot: 5 ota 1 OTA app 00 11 00280000 00180000									
I (102) boot: End of partition table									
I (106) boot: No factory image, trying OTA 0									

Das heißt: es ist keine Application im Flash. Alles gut bis hierher.

5.3 Überprüfung/Aktualisierung 'esptool'

Thonny bringt nicht nur Python mit, sondern auch das Flash-Tool 'esptool', welches selber unter Python (auf dem PC) läuft. Hier sollte darauf geachtet werden die neueste Version (> 4.62) zu haben. Das kontrollieren wir in Thonny, indem wir unter 'Werkzeuge' auf 'Verwalte Plug-ins...' gehen.

Jetzt in der links angezeigten Liste von Paketen nach 'esptool' suchen und mit einem Mausklick auswählen. Das Hauptfenster zeigt jetzt die installierte Version und das Installationsverzeichnis. Bei Bedarf dann bitte eine neuere Version von <u>PyPi</u> nachinstallieren.

In diesem Fenster werden Th Falls Sie Pakete für Ihre eiger	ionny Plug-ins und Ihre Korrespondierenden Pakete verwaltet. ien Programme installieren möchten, wählen Sie "Extras → Pakete verwalten".
Hinweis: Sie müssen Thonny	nach der Installation / Aktualisierung / Deinstallation eines Plugins neu starten.
	Suche im PyPI
<installieren> astroid asttokens</installieren>	esptool
bcrypt	Installierte Version: 4.6.2
bitarray	Installiert in: /home/normal/apps/thonny/lib/python3.10/site-packages
ortifi	Neueste stabile Version: 4.7.0
cffi	Zusammenfassung: A serial utility to communicate & flash code to Espressif chips.
cryptography	Autor: Fredrik Ahlberg (themadinventor) & Angus Gratton (projectgus) & Espressif Systems
dbus-next	Lizenz: GPLv2+
dill	Homepage: https://github.com/espressif/esptool/
distro	PyPI-Seite: <u>https://pypi.org/project/esptool/</u>
docutils	
ecdsa	
esptool	
install 🗸	Schließen

5.4 Vorbereiten Modul mit Adapter

Wenn das ESP32 Modul bereits auf die Adapter-Platte gelötet ist, verhindert der Pull-Up für das Signal SD_D0 an GPIO2, dass das Flash erkannt und programmiert werden kann. Also muss vor dem Flashen dieser Pin mit seinem Nachbarn GPIO0 kurzgeschlossen werden. Diese beiden Pins sind auf der Adapterplatte durch den *nicht* bestückten Widerstand R7 gekennzeichnet. Die einfachste Methode ist hier, sich die Steckbrücke (Jumper) vom PiLogger One auszuleihen und auf die beiden Modulstifte zu stecken (nicht vergessen, ihn nachher wieder auf den PiLogger zu stecken) :



5.5 Flashen MicroPython

Thonny sucht die zu flashende Datei standardmäßig im Root-Verzeichnis. Da die Entwicklung und die Tests auf der MicroPython Version 'ESP32_GENERIC-20240222-v1.22.2.bin' stattgefunden haben, ist diese Version der .bin Datei im Download-Archiv enthalten.

Diese Datei entpacken wir also in das Root-Verzeichnis auf dem PC. In Thonny rufen wir unter 'Ausführen', 'Konfiguriere den Interpreter...' die Seite 'Interpreter' auf:

_	_	_		ioniny Extras	_	_	v /
Allgemeines	Interpreter	Editor	Theme & Schriftart	Ausführen & Debuggen	Terminal	Kommandozeile	Assisten
Welche Art	on Interpret	er soll Th	ionny zur Ausführun	g Deines Codes benutzen?	,		
MicroPytho	n (ESP32)						-
Details							
Connectin	g via USB cab	le:					
Verbinden	Sie Ihr Gerät	mit den	Computer und wäh	len Sie den relevanten Por	t unten aus	;	
(Suchen Si Falls Sie es	e nach Ihrer (nicht finden	eratesc müssen	hnittstelle "USB Seria Sie möglicherweise (l" oder "UARI"). erst den richtigen USB-Tre	iber installi	ieren.	
	,						
Connectin	g via WebREP	L:					
(import w	vice supports ebrepl setup)	. connec	L, first connect via se t your computer and	rial, make sure webREPL I device to same network a	s enabled nd select		
< WebREP	L > below		, i				
Port oder	WebREPL						
							•
× Beim Ve	rhinden laufe	ondes Pro	ogramm unterbreche	n			
× Synchro	nisiere die Ec	htzeituh	r des Geräts				
× Benutze	lokale Zeit in	Echtzeit	uhr				
× Vor Star	t des Skript Ir	nterprete	er neu starten				
				MicroPython in	stallieren o	der aktualisieren (<u>esptool)</u>
							<u>(UF2)</u>
						OK .	Abbrechei

Jetzt klicken wir auf 'MicroPython installieren'.

Th	Install MicroPython (esptool) V 🔨 🔨	×						
Click the ≡ button to 'info' page for details	o see all features and options. If you're stuck then check the variant's s or ask in MicroPython forum.							
NB! Some boards need to be put into a special mode before they can be managed here (e.g. by holding the BOOT button while plugging in). Some require hard reset after installing.								
You may need to twe your device precisely	eak the install options (\equiv) if the selected MicroPython variant doesn't match . For example, you may need to set flash-mode to 'dio' or flash-size to 'detect	ť.						
Target port	USB Single Serial @ /dev/ttyACM0	•						
	🗷 Erase all flash before installing (not just the write areas)							
MicroPython family	ESP32	•						
variant	Espressif • ESP32 / WROOM	•						
version	1.22.2	•						
info	https://micropython.org/download/ESP32_GENERIC							
	■ Installieren Abbrecher	1						

Dieser Screenshot ist beispielhaft von einem Linux-Rechner, weshalb hier der 'Target port' 'ttyACM0' heißt. Durch das automatische Suchen ist der echte USB Port in der Ausklappliste auswählbar.

Durch das Auswählen von 'ESP32' bei 'MicroPython family' sowie 'Espressif • ESP32 / WROOM' bei 'variant' wir durch eine Online-Abfrage die aktuelle Version als Download ausgewählt und angezeigt. Wenn hier eine neuere Version als '1.22.2' erscheint und es deshalb Kompatibilitätsprobleme gibt, kann stattdessen auch die im Archiv enthaltene lokale Datei verwendet werden. Dazu drücken wir auf den Knopf mit dem 'Hamburger'-Symbol und bekommen dieses Kontext-Menü:



Danach können wir mit einem Datei-Wähler zur *ausgepackten* Datei 'ESP32_GENERIC-20240222-v1.22.2.bin' navigieren und diese mit 'Öffnen' auswählen.

Mit einem Klick auf 'Installieren' starten wir jetzt den Flash-Prozess.

	Install MicroPython (esptool) \sim \sim \times							
Click the ≡ button to 'info' page for details	Click the ≡ button to see all features and options. If you're stuck then check the variant's 'info' page for details or ask in MicroPython forum.							
NB! Some boards need to be put into a special mode before they can be managed here (e.g. by holding the BOOT button while plugging in). Some require hard reset after installing.								
You may need to tweak the install options (≡) if the selected MicroPython variant doesn't match your device precisely. For example, you may need to set flash-mode to 'dio' or flash-size to 'detect'.								
Target port	USB Single Serial @ /dev/ttyACM0							
	Erase all flash before installing (not just the write areas)							
MicroPython family	ESP32							
variant	Espressif • ESP32 / WROOM							
version	1.22.2							
info	https://micropython.org/download/ESP32_GENERIC							
Writing at 0x00132be7 (72 %)								

Nach dem Schreiben erscheint unten links die Meldung 'Done !'.

Wie schließen die beiden Fenster und resetten den ESP32 mit dem Reset-Knopf am Modul. Nach einem Klick auf das Stop-Symbol in der Menüleiste von Thonny sollte sich der ESP32 mit dem MicroPython Prompt melden.

Danach können wir den Jumper wieder entfernen und beim PiLogger in der richtigen Position wieder aufstecken.

5.6 Kopieren der Archiv-Dateien

Jetzt entpacken wir das heruntergeladene Archiv in einen passend benannten Arbeitsordner für Thonny.

Wenn wir jetzt Thonny öffnen, das ESP32 Modul anstecken, in Thonny

'Stop/Restart' drücken und das Modul reseten 😎

dann sollte es auf dem Bildschirm so ähnlich aussehen:



MicroPython (ESP32) • USB Single Serial @ /dev/ttyACM0 ≡

Das heißt: Auf dem 'Target' - dem 'MicroPython device' gibt es jetzt genau eine Datei: 'boot.py'. Die tut nichts - die spielt noch nicht mal...

Mit einem Rechtsklick auf den Ordner 'static' im oberen Datei-Baum rufen wir das Kontextmenü auf, in dem wir den ganzen Ordner auf das Target mit 'Upload nach /' zum Target kopieren können.

Dann wechseln wir in beiden Datei-Bäumen in die Ordner 'static' und kontrollieren kurz, dass sie den gleichen Inhalt haben:



Jetzt können wir in beiden Bäumen wieder jeweils eine Ebene zurückgehen. In 'Dieser Computer' wählen wir mit 'Strg'+Linksklick die folgende 8 Dateien aus :



Mit einem Rechtsklick auf eine der gewählten Dateien können wir dann mit 'Upload nach /' alle 8 gleichzeitig hochladen.

Für die Datei 'boot.py' wird dabei rückgefragt, ob sie überschrieben werden soll, was wir bestätigen.

Damit sind die benötigten Dateien auf dem ESP32.

5.7 Anpassen der WiFi Zugangsdaten

In der Datei 'wifi_config.py' auf dem ESP32 müssen nun die tatsächlich benötigten Daten eingetragen werden, mit denen sich der ESP32 im Heimnetzwerk anmelden kann.

Dazu doppelklicken wir auf diese Datei, wodurch sie vom ESP32 heruntergeladen wird und im oberen Fenster von Thonny - dem Editor - angezeigt wird:



Nach dem Eintragen der echten individuellen Werte für die Platzhalter zwischen den Hochkommas, speichern wir mit Klick auf das Icon 'Speichern (Ctrl+S)' oder über 'Datei', 'Speichern'.

<u>Wichtiger Hinweis</u> : Die Software geht normal davon aus, dass der Heimnetz-Router als interner NTP Server agiert (bei vielen Routern einstellbar).

Wenn dies nicht möglich ist, muss die Datei 'boot.py' auf dem ESP32 geändert werden. Dazu muss der Wert für 'NTP_SERVER' in Zeile 16 auf zum Beispiel 'pool.ntp.org' geändert werden :



Bitte beachten !

Dadurch ist für die Zeitsynchronisation ein Internet-Zugriff notwendig und der ESP32 kann nicht mehr für den Internet-Zugriff gesperrt werden ! Die lokale Zeitsynchronisation (Router als interner NTP Server) ist aus Sicherheitsgründen die bessere Lösung.

5.8 Netzwerknamen ändern

Standardmäßig erhält der ESP32 den Netzwerknamen 'espressif'. Das lässt sich ändern, wenn in der Datei 'boot.py' auf dem ESP32 eine zusätzliche Zeile eingefügt wird.

Die Zeile lautet:

```
network.hostname('pilogger-esp32')
```

Damit wird das MicroPython Modul 'network' konfiguriert, deshalb muss die Zeile nach dem Import dieses Moduls auftauchen. Und diese Konfiguration sollte erfolgen, bevor die WLAN Schnittstelle initialisiert wird.

Der gewählte Name ist hier natürlich nur ein Beispiel und kann beliebig individualisiert werden. Dabei empfiehlt es sich gemäß RFC Vorgaben nur Kleinbuchstaben, Zahlen und das Minuszeichen zu verwenden – für maximale Kompatibilität.

Das Ganze sieht dann etwa so aus :



5.9 PiLogger One montieren

Für den Zusammenbau von ESP32 mit Adapterplatte und dem PiLogger One setzen wir zunächst 2 Abstandsbolzen in die Montagelöcher der Adapterplatte und verschrauben sie *ohne* Unterlegscheibe jeweils mit einer Mutter :



Ausgabe V2.2 , 29.06.2024

Anschließend setzen wir die Stiftleiste der Adapterplatte genau in die Buchsenleiste des PiLoggers ein und drücken sie so zusammen, dass ein gleichmäßiger Spalt nach Maßgabe des Abstandsbolzen bleibt :



Von der anderen Seite sieht das Ganze so aus :



Wenn das 3D-gedruckte ESP32 Gehäuse aus unserem Shop verwendet werden soll, kann das Doppelpack jetzt kopfüber in das Gehäuse eingesetzt werden. Danach können direkt die 2 Schrauben eingesetzt werden, die in Abstandsbolzen geschraubt werden. Anschließend wird die dritte Schraube mit Unterlegscheibe und Mutter eingesetzt :



Wenn kein Gehäuse verwendet wird, werden die beiden Abstandsbolzen jeweils mit Unterlegscheibe und Schraube am PiLogger verschraubt.

5.10 Erstes Einschalten

Nun muss nur das Modul mit dem USB Kabel wieder angeschlossen werden und schon bootet der ESP32 als PiLogger ⁽²⁾ mit dem Netznamen 'espressif' oder dem speziell gewählten Netznamen.

Der ESP32 meldet beim Starten mit Thonny im Debug-Fenster die IP-Adresse unter der er zu erreichen ist.

An einem beliebigen Gerät im Heimnetz kann nun in einem Browser als Adresse die IP mit dem Zusatz ':8080' eingeben.

Also zum Beispiel '192.168.178.53:8080'.

6 Bedienungsanleitung WebMonitor Software

6.1 Aufruf der Web-Seite mit dem Browser

Wenn alle Vorbereitungen aus den vorherigen Kapiteln erfolgreich abgeschlossen sind, kann die Web-Seite des PiLogger WebMonitor auf einem beliebigen Gerät im Heimnetzwerk mit einem Web-Browser aufgerufen werden.

Dazu geben wir den Netzwerknamen des Raspberry gefolgt von '.local:8080' in die Adressleiste des Browsers ein:



Bei einigen Browsern ist '.local' nicht unbedingt notwendig – es zeigt dem Browser nur an, dass die Webseite im lokalen Netz liegt. Bei besonders hartnäckigen Browsern / Routern muss ersatzweise die IP-Adresse mit dem Zusatz ':8080' dem Port - verwendet werden. Also beispielsweise '192.168.178.20:8080'. Die Eingabe müssen wir mit 'Enter' abschließen.



Das ist die Startseite unseres frisch aufgesetzten Servers, wenn alles wie geplant gelaufen ist. Wir können nun für den Browser einen Favoriten anlegen – in diesem Beispiel bei 'Firefox' mit einem Klick auf den Stern ganz rechts in der Adressleiste. Damit sind die weiteren Aufrufe dieser Seite mit einem Klick möglich.

6.2 Die Startseite - Live Werte

Die Startseite zeigt die aktuellen Mess- und Zählerwerte des PiLogger One an und ruft diese automatisch alle 2 Sekunden frisch ab.

Auf der linken Seite gibt es ein Navigationsmenü, das es ermöglicht die weiteren Seiten aufzurufen. Die jeweils aktuelle Seite ist dabei blau unterlegt. Dieses Menü verschwindet bei Geräten mit zu kleinem Bildschirm und muss dort mit einem Tippen (Tap) auf das stattdessen auftauchenden Menü-Icon extra aufgerufen werden.





Auf kleineren Bildschirmen, wie etwa Smartphones, erscheinen die Messwertfelder als Liste. Wird in dieser Liste gescrollt (hoch gewischt), erscheint ein rotes Quadrat, das den schnellen Rücksprung zum Anfang der Liste ermöglicht.

Jedes Anzeigefeld für einen Messwert oder Zählerstand dient außerdem als Schaltfläche für einen Link auf die Zeitdiagrammdarstellung des jeweiligen Wertes.

Ein Klick oder Tap auf die Mitte des

Temperaturfeldes ruft also die Diagramme-Seite auf und hat dabei die Anzeige der Mittelwerte der Temperaturmessungen aktiviert.

Gleiches gilt sinngemäß für die anderen Messwertfelder.

Ein Klick (oder Tap) auf das kleine Zahnradsymbol rechts unterhalb des Messwertes öffnet ein Pop-Up-Fenster zur Auswahl der Hintergrundfarbe für die betreffende Messwert-Kachel.

PiLogger				πĹ
Menu	Donnerstag, 02.Mai 2024 01:44:	08 0.0 m/s	4	R
 Live Wert Live Kurv Diagramm Einstellur Einstellur 	(6) 50,0 1	Farbauswahl für Farbe01 - hier klicken : Alles zurück Fertig		0,005 A ♪ 1h 39m 0s .00 Wh
Download	Leistung C C Entladen 0,0 Ω Widerstand	Tages-Ertrag Image: Stars and Stars	Tages-Verbrauch Image: State of the st	Tages-Bilanz 36d 5h 35m 1,54 kWh Dauer-Bilanz

Mit einem weiteren Klick auf das Farbmusterfeld wird der eigentliche Farbwähler des verwendeten Browsers geöffnet. Die Farbwähler der verschiedenen Browser sind sehr unterschiedlich – hier beispielsweise der der Desktop-Variante von Firefox:

PiLogger				rL.
Menu	Donnerstag, 02.Mai 2024 01:45	21 Wählen Sie eine Earbe		
Live Wert Live Kurv),006 A
Diagramr				0
Einstellur				1h 39m 0s
Einstellur	Benutz	erdefiniert		,00 Wh
CALIBRATION	Leistung	Abbrechen	Auswählen Prbrauch	Tages-Bilanz
_	- Entladen	36d 5h 35m	36d 5h 35m	36d 5h 35m
	0,0 Ω	1,54 kWh	0,25 Wh	1,54 kWh
	Widerstand	Dauer-Ertrag	Dauer-Verbrauch	Dauer-Bilanz

Hier gibt es schon eine beachtliche Vorauswahl – man kann aber mit einem Klick auf das '+' im Bereich 'Benutzerdefiniert' jede beliebige Farbe zusammenmischen. Das ist aber in der Android-Variante des Firefox total anders – da gibt es genau 10 Farben zur Auswahl ...

Weil das auch zu ungewollten Ergebnissen führen kann, gibt es im Pop-Up-Fenster die Schaltfläche 'Alles zurück', mit der die vordefinierten Farben wiederhergestellt werden können. Die sind genau deshalb etwas blasser und unauffälliger, weil die knallbunten Farben überall zur Verfügung stehen 🙂

Mit einem Klick auf die Schaltfläche 'Fertig' wird die zuvor gewählte Farbe für dieses eine Feld übernommen. Alle Feldfarben werden in diesem Browser lokal abgespeichert – das heißt : auf einem anderen Gerät können ganz andere Farben individuell ausgewählt werden.

6.3 Die Seite Live Kurve

Die Seite 'Live Kurve' lässt sich über das Navigationsmenü aufgerufen und startet beim ersten Aufruf mit der Darstellung der aktuellen Temperatur-Messwerte über der Zeit:



Der Diagrammbereich passt sich automatisch der verfügbaren Bildschirmfläche an. Die Achsen werden automatisch den Datenbereichen angepasst. Die Daten werden dabei mit dem gewählten Intervall jeweils frisch vom Raspberry Pi abgefragt und so als aktueller Wert vom PiLogger bezogen. Dazu sollte das Messintervall entsprechend kurz eingestellt sein (Seite 'Einstellungen 1'), damit

die angezeigten Messpunkte sich auch wirklich aktuell ändern können. Oben rechts im Diagrammbereich wird eine Legende der aktiven Kurven angezeigt und bei aktivem Marker auch die jeweiligen Einzelwerte zu einem gewählten Zeitpunkt (siehe 6.4.4).

Unter dem Diagrammbereich kommt zunächst eine Zeile mit Einstellmöglichkeiten für das Füllen der dargestellten Kurve, dem Anzeigemittelwertfaktor und dem Aktualisierungsintervall.

V Füllen	Mittelwert: 1 Werte	Intervall:	kurz	🔘 normal	O Benutzer	5 sec
----------	---------------------	------------	------	----------	------------	-------

Anzeige linke Achse:				Anzeige rechte Achs	Anzeige rechte Achse:				
Temp	🗸 Akt			Temp	Akt				
Wind	Akt			Wind	Akt				
Volt	Akt			Volt	Akt				
Amps	Akt			Amps	Akt				
Watt	Akt			Watt	Akt				
Ohm	Akt			Ohm	Akt				
Wh Dauer	🗌 Bila	Ertr	Verb	Wh Dauer	🗌 Bila	Ertr	Verb		
Wh Tag	BilTg	ErtTg	VerTg	Wh Tag	BilTg	ErtTg	VerTg		

Darunter folgt eine Tabelle der verfügbaren Messwerte :

Beim ersten Aufruf dieser Seite ist hier nur die Anzeige 'Temp' für die linke Achse aktiv. Durch Klicken oder Antippen können die einzelnen Auswahlfelder (Checkboxes) aktiviert oder deaktiviert werden. Wenn mindestens ein Messwert für die rechte Achse ausgewählt ist, wird im Diagrammbereich eine zweite Werteachse auf der rechten Seite angezeigt.

Wird ein bereits aktivierter Messwert auf der anderen Achse aktiv geschaltet, wird er automatisch auf der ersten Achse deaktiviert.

Alle diese Einstellungen werden lokal im Browser für diese Seite abgespeichert. Dadurch wird diese Seite beim nächsten Aufruf mit den zuletzt gewählten Einstellungen gestartet.

Die gespeicherten Einstellungen gelten also für genau dieses Endgerät und den benutzten Browser. Wird die gleiche Seite auf einem anderen Gerät (oder auf demselben Gerät mit einem anderen Browser) aufgerufen, so sind dort andere individuelle Einstellungen gespeichert.

Das geschieht mittels der in HTML 5 verfügbaren Technologie 'Web Storage', genauer hier mittels 'Local Storage', die nichts mit der 'Cloud' zu tun hat, tatsächlich eher eine Art 'Cookie' ist.



Deshalb muss zur Nutzung beim Browser entweder das 'Löschen der Website-Daten beim Beenden' deaktiviert sein oder besser für die Adresse des PiLogger WebMonitor eine Ausnahme hinzugefügt werden.

Die Bedienung dieser Seite ist grundsätzlich die selbe wie für die Seite 'Diagramme' – bis auf die Einstellung des Aktualisierungsintervalls. Deshalb wird hier zunächst diese Einstellung beschrieben und für die übrigen Punkte auf das nächste Kapitel 6.4 'Die Seite Diagramme' verwiesen.

6.3.1 Aktualisierungsintervall einstellen

Das Aktualisierungsintervall ist standardmäßig auf 'normal' gestellt:



Das entspricht den rechts angezeigten 5 Sekunden. Mit einem Klick oder Tap auf den Knopf 'kurz' lässt sich schnell auf 2 Sekunden umschalten.

Klicken wir in das Zahlenfeld, so können wir direkt eine Zeit in Sekunden angeben, die im Bereich 1...120 liegen muss. Mit einem 'Enter' beenden wir die Eingabe und der Wert wird als Benutzer-Wert übernommen und aktiviert. Sinnvollerweise sollte das Aktualisierungsintervall nicht kürzer sein als das Messintervall vom PiLogger, das auf der Seite 'Einstellungen 1' eingestellt wird und sollte möglichst ein ganzzahliges Vielfaches davon sein.

6.4 Die Seite Diagramme

Wird die Seite 'Diagramme' über das Navigationsmenü aufgerufen, startet diese Seite beim ersten Aufruf mit der Darstellung der Temperatur-Mittelwerte über der Zeit:



Der Diagrammbereich passt sich automatisch der verfügbaren Bildschirmfläche an. Die Achsen werden automatisch den Datenbereichen angepasst.

Die Daten werden dabei aus der aktuellen Logdatei plus dem vorherigen Logzeitraum bezogen. Dazu wird die Datei 'showdata.csv' jeweils beim Aufruf der Seite frisch vom Raspberry heruntergeladen.

Oben rechts im Diagrammbereich wird eine Legende der aktiven Kurven angezeigt und bei aktivem Marker auch die jeweiligen Einzelwerte zu einem gewählten Zeitpunkt (siehe 6.4.4).

Unterhalb des Diagrammbereiches folgen eine Reihe von Einstellungen. Diesen Bereich erreichen wir durch Scrollen, also durch Hochschieben des Inhalts der Seite.

Nachfolgend werden diese Einstellmöglichkeiten beschrieben.

6.4.1 Kurve füllen

Zunächst erscheint eine Zeile mit einem Schaltfeld (Button) 'Füllen' und einem Eingabefeld 'Mittelwert':

	Füllen	Mittelwert:	1	Werte	
--	--------	-------------	---	-------	--

Aktivieren wir die Option 'Füllen', ändert sich die Darstellung der Messwertkurve so, dass der Bereich unter der Kurve als gefüllte Fläche erscheint:



Die Füllfarbe ist dabei eine hellere transparente Variante der Kurvenfarbe. Diese Einstellung wird im 'local storage' gespeichert.

6.4.2 Gleitenden Mittelwert verwenden

Die Diagramm-Kurven lassen sich temporär mit einem gleitendem Mittelwert geglättet darstellen (die Original-Daten bleiben unverändert). Dazu klicken wir in das kleine Eingabefeld 'Mittelwert':



Danach geben wir einen größeren Faktor als '1' (keine Mittelung) ein, zum

Beispiel '10', und schließen die Eingabe mit 'Enter' ab. Das ergibt dann diese Darstellung:



Durch Eingabe von '1' wird die Mittelung wieder zurückgesetzt. Es können ganzzahlige Werte von 1 bis 1000 eingegeben werden. Diese Einstellung wird im 'local storage' gespeichert.

6.4.3 Anzuzeigende Messwertreihen auswählen

Wenn wir weiter nach unten scrollen, also zum unteren, verdeckten Seitenbereich, wird diese Tabelle mit Schaltfeldern (Buttons) sichtbar:

Anzeige linke Achse:				Anzeige rechte Achse:			
Temp	🗸 Avg	Min 🗌	Max 🗌	Temp	Avg	🗌 Min	Max 🗌
Wind	Avg	Min 🗌	Max	Wind	Avg	Min 🗌	Max
Volt	Avg	Min 🗌	Max 🗌	Volt	Avg	🗌 Min	Max 🗌
Amps	Avg	Min 🗌	Max	Amps	Avg	Min 🗌	Max
Watt	Avg	Min 🗌	Max 🗌	Watt	Avg	Min 🗌	Max 🗌
Ohm	Ohm			Ohm	Ohm		
Wh Dauer	🗌 Bila	Ertr	Verb	Wh Dauer	🗌 Bila	Ertr	Verb
Wh Tag	BilTg	ErtTg	VerTg	Wh Tag	BilTg	ErtTg	VerTg

Auf kleineren Bildschirmen (z.B. Smartphone senkrecht) erscheinen die beiden Hälften (links und rechts) untereinander.

Beide Hälften zeigen die gleichen Auswahlmöglichkeiten. Das heißt, jede Messwertreihe kann für die linke oder die rechte senkrechte Achse aktiviert werden.

Ist mindestens eine Messwertreihe für die rechte Achse aktiviert, wird im Diagrammbereich zusätzlich an der rechten Seite die zweite Werteachse angezeigt.

Durch Anklicken oder Tippen wird die Anzeige der jeweilige Messwertreihe

aktiviert oder deaktiviert. Wird ein bereits aktivierter Messwert auf der anderen Achse aktiv geschaltet, wird er automatisch auf der ersten Achse deaktiviert.

Auch diese Einstellung wird im 'local storage' abgespeichert.

6.4.4 Einzelmesswert anzeigen

Auf Bildschirmen mit Mausbedienung taucht sofort, wenn der Mauszeiger im Diagrammbereich ist, ein Marker in Form eines gelb gefüllten Kreises auf, der der horizontalen Mauszeigerposition folgt, aber auf der Messwertkurve bleibt. Werden mehrere Messwertkurven angezeigt, hat jede Kurve ihren eigenen Marker. Die jeweiligen Einzelmesswerte am Markerpunkt werden oben rechts in der Legende angezeigt zusammen mit Datum und Zeit.

Auf Touch-Bildschirmen wird dieser Marker mit einem kurzen, leichten Tipp an eine gewünschte Stelle im Diagramm aktiviert. Der Tipp darf nicht zu lang sein und keine Bewegung enthalten, da er sonst als Geste interpretiert wird. Auch das kräftige Aufdrücken kann als Bewegung missverstanden werden. Hier ein Beispiel für eine Mauszeiger-gesteuerte Messwertanzeige:



6.4.5 Verschieben und Zoomen

Der angezeigte Kurvenbereich innerhalb des Diagramms lässt sich verschieben und zoomen. Das ist sowohl mit Maus als auch mit Touch möglich, aber sehr unterschiedlich.

6.4.5.1 Maus-Bedienung

<u>Zoomen</u>

Um mit der Maus einen Bereich zu vergrößern, fahren wir an eine Stelle für den gewünschten Start des zu vergrößernden Bereiches – drücken und halten die linke Maus-Taste und ziehen den Mauszeiger zum gewünschten Ende des Zoom-Bereiches. Dabei wird der Bereich farblich markiert. Dann lassen wir die linke Maus-Taste los und das Diagramm wir auf den gewünschten Bereich vergrößert.



Der Screenshot zeigt das Markieren eines Zeitbereiches. Nach dem Zoomen sieht das so aus:



Analog lässt sich auch ein vertikaler Bereich, also ein Wertebereich, zoomen. Hier das vertikale Markieren des Bereiches:
Handbuch PiLogger One



Mit diesem Ergebnis:



Zoom zurücksetzen

Mit einem Doppel-Klick im Diagrammbereich wird der Zoom wieder zurückgesetzt. Es wird wieder die automatische Vollsicht über alle Daten angezeigt.

<u>Verschieben</u>

Um den Diagrammbereich zu verschieben (Pan), setzen wir den Mauszeiger irgendwo in den Diagrammbereich und drücke und halten die ,'Shift'-Taste. Wenn wir nun den Mauszeiger bewegen, verschieben wir damit den Diagrammbereich. Nach dem Loslassen der 'Shift'-Taste, haben wir wieder die normale Bewegung des Markers mit dem Mauszeiger.

Ist das Diagramm im nicht gezoomten Zustand, können wir nur waagerecht verschieben. Im gezoomten Zustand ist die Bewegung auch senkrecht möglich oder in Kombination.

 Verschieben zurücksetzen Auch hier führt ein Doppel-Klick irgendwo im Diagrammbereich zurück zur Standard-Anzeige.

6.4.5.2 Touch-Bedienung

Bei der Touch-Bedienung ist die Flüssigkeit der Steuerung stark vom verwendeten Browser abhängig. So reagiert zum Beispiel der Firefox-Browser für Android eigentlich immer auch selber auf die Zwei-Finger-Geste mit einer Zoom-Änderung – die hier aber nicht gemeint ist – das führt zu heftigen Verzögerungen oder auch zum ungewollten Zoom des gesamten Seiten-Inhalts. Das lässt sich am besten im Navigationsmenü-Bereich wieder rückgängig machen.

<u>Zoomen</u>

Auf einem Touch-Bildschirm wird mit einer Zwei-Finger-Geste gezoomt. Einfach zwei Finger gleichzeitig im Diagrammbereich aufsetzen und anschließend die Finger auf dem Bildschirm voneinander entfernen, um den Bereich unter den Fingern zu vergrößern – oder aufeinander zu bewegen um den Bereich zu verkleinern (Pinch). Das funktioniert gleichzeitig in waagerechter, wie in senkrechter Richtung.

<u>Verschieben</u>

Verschieben funktioniert mittels dauerhaft aufgesetztem Finger und anschließendem Bewegen des aufgesetztem Fingers in beliebiger Richtung – im Diagrammbereich. Es können auch zwei Finger dafür benutzt werden. Es lässt sich also auch kombinieren mit dem Zoomen.

<u>Zurücksetzen</u>

Auch bei Touch-Bedienung werden Zoom und Verschiebung durch einen Doppel-Tipp im Diagrammbereich zurückgesetzt.

6.5 Die Seite Einstellungen 1

PiLogger				πĹ				
Menu	Einstellungen	Einstellungen 1						
D Live Werte	PiLogger Einstellungen	Ist	Soll	Eingabe				
Live Kurve	I ² C Adresse	72	72					
Miagramme	Faktor Mittelung	8	8	~				
👛 Einstellungen 1	Pulszähl Zeitfaktor	1	1					
	Zeitbasis	1 sec	1 sec	~				
Einstellungen 2	Intervall Lo Byte	5	5					
ک آ ک Kalibration	Intervall Hi Byte	0	0					
🛓 Download	ergibt Messintervall	5s	5s	<				
	Senden	ichern	Am	p-Zero Werkseinst.				

Auf dieser Seite werden alle Einstellungen verwaltet, die direkt mit dem PiLogger selber zusammenhängen.

Sobald das Fenster aufgerufen wird, werden die Werte der linken Spalte 'Ist' automatisch einmal vom PiLogger gelesen.

In der rechten Spalte 'Eingabe' können für die einzelnen Parameter neue Werte eingegeben werden. Die Spalte 'Soll' listet die Werte auf, die beim nächsten 'Senden' oder 'Speichern' verwendet werden.

Diese sind im Einzelnen:

• I²C Adresse

Dieser Parameter wird als Dezimalwert angegeben und ist die Bus-Adresse auf die der PiLogger reagiert. Diese Adresse darf im Bereich 4 bis 124 (0x04 bis 0x7C) liegen.

Die Standard-Adresse ist die 72 (0x48). Nur in Ausnahmefällen sollte es einen Grund geben diese Adresse zu ändern – wenn zum Beispiel zwei PiLogger an einem Raspberry Pi betrieben werden sollen.

Damit eine geänderte I²C Adresse dauerhaft im PiLogger gespeichert wird, muss anschließend einmal auf 'Speichern' gedrückt werden – siehe unten.

Faktor Mittelung

Der PiLogger berechnet für jeden der abrufbaren 5 Messwerte fortlaufend einen gleitenden Mittelwert. Der dafür benutzte Gewichtungsfaktor kann hier gewählt werden.

Gültige Werte für diesen Parameter M liegen zwischen 1 und 7, wobei der Mittelungsfaktor dann 1/(2^M) beträgt. Somit lässt sich der Mittelungsfaktor zwischen ½ und 1/128 einstellen. Der aktuelle Messwert wird mit diesem Faktor gewichtet und zum bisherigen Mittelwert addiert.

• Pulszähl Zeitfaktor

Hier kann ein Vervielfachungsfaktor für die Messzeit am Impulszähleingang gesetzt werden. Gerade bei kleinen Messintervallen würden sonst viele Messungen mit Null Impulsen erzeugt werden, die keinen sinnvollen Wert für z.B. die Windgeschwindigkeit ergeben. Wenn beispielsweise das Messintervall für Spannung, Strom und Leistung auf ¼ Sekunde (250 msec) eingestellt wird, kann hier durch einen Faktor 40 dafür gesorgt werden, dass die Werte für die Windgeschwindigkeit alle 10 Sekunden ermittelt werden.

• Zeitbasis

Der PiLogger arbeitet mit einer internen Zeitbasis die für das Messintervall verwendet wird. Hier kann zwischen 4 Basis-Zeiteinheiten gewählt werden:

- 0:1 sec
- 1 : 250 msec
- 2 : 15,625 msec
- 3 : 1,953125 msec

Damit wird der insgesamt einstellbare Zeitbereich für das Messintervall und die Auflösung darin bestimmt. Mit den nachfolgenden 2 Werten für den Faktor kann ein Vielfaches dieser Zeitbasis von 1x ... 65535x als Messintervall eingestellt werden.

<u>Wichtig</u> : Hier wird die interne Messrate des PiLoggers eingestellt – nicht die Datenabfragerate. Die Messwerte werden im PiLogger kontinuierlich gemittelt. Die Messrate kann und sollte höher sein als die Abfragerate (Zeitintervall kleiner). Dazu mehr unter 6.6 'Einstellungen 2 - Log-Faktor'

Als kleinstes Messintervall dürfen 7,8 msec nicht unterschritten werden, also 4 x 1,953125 msec = 7,8125 msec. Das entspricht 128 Messungen pro Sekunde. Als größtes Zeitintervall können 65535 sec eingestellt werden, was 18 Stunden, 12 Minuten und 15 Sekunden entspricht.

Die WebMonitor Software berechnet aus den aktuell eingestellten Werten jedes Mal das resultierende Messintervall in Sekunden. Man kann also durch Ändern der Werte und senden an den PiLogger das neue Messintervall live testen und den Wert in Sekunden sehen. Die LED am PiLogger blitzt für jede Messung einmal auf (ca. 5 msec an).

• Intervall Low Byte

Niederwertiges Byte des Faktors für die Messintervall-Einstellung.

• Intervall High Byte

Höherwertiges Byte des Faktors für die Messintervall-Einstellung. Der Gesamtfaktor berechnet sich zu H x 256 + L. Mindestwert ist 1.

Knopf 'Senden'

Neu eingegebene Werte werden durch einen Klick auf den Knopf 'Senden' an den PiLogger geschickt und sind sofort wirksam. Bei einer Änderung des Messintervalls sieht man das sehr schön an der Aktivität der PiLogger-LED.

• Knopf 'Speichern'

Sollen die geänderten Einstellungswerte dauerhaft übernommen werden, so dass nach einem Strom-Ausschalten und wieder Einschalten (power cycle) der PiLogger auch wieder mit diesen Einstellungen arbeitet, müssen diese Werte im internen Flash-Speicher des PiLoggers abgelegt werden. Dazu wird einmal der Knopf 'Speichern' angeklickt.

• Knopf 'Amp-Zero'

Der Knopf in der unteren Zeile mit der Aufschrift 'Amp-Zero' sendet einen Befehl an den PiLogger um den aktuellen Mittelwert für den Strom als Nullwert zu setzen.



Achtung : Dieser Befehl darf nur angewandt werden, wenn zur Zeit wirklich kein Strom durch den Sensor fliesst, also zum Beispiel an den Schraubklemmen nichts angeschlossen ist.

Dadurch werden Exemplarstreuungen und andere Toleranzen im Strommesszweig heraus kalibriert. Vor Anwendung sollte der PiLogger etwa 20 Minuten aktiv gewesen sein, damit er thermisch eingeschwungen ist. Das Drücken des Knopfes 'AmpZero' kann bei Bedarf beliebig oft wiederholt werden.

Ist das Ergebnis nach einigen Minuten stabil nahe Null, muss diese Einstellung durch das Drücken des Knopfes 'Speichern' permanent im PiLogger gespeichert werden, damit auch nach einem Neustart diese Korrektur aktiv ist.

Diese Kalibration ist einmal bei Erstinbetriebnahme sinnvoll und gegebenenfalls wenn nach einiger Zeit im Betrieb eine deutliche Missweisung festgestellt wird, oder der Aufbau sich verändert hat.

Knopf 'Werkseinst.'

Für den Fall, dass der Zustand des PiLogger unklar ist, zum Beispiel durch extreme Messintervalle, kann mit diesem Knopf ein Befehl an den PiLogger gesendet werden, der die Standard-Einstellungen wiederherstellt. Diese Standard-Einstellungen lädt der PiLogger aber erst bei einem Neustart – das heißt, der PiLogger muss einmal für einige Sekunden stromlos geschaltet werden und dann wieder eingeschaltet werden. Dafür kann unter 'Einstellungen 2' der Knopf 'Herunterfahren' genutzt werden, um den Raspberry mitsamt dem PiLogger kurz von der Versorgung zu trennen.

6.6 Die Seite Einstellungen 2

PiLogger				n	L		
Menu	Einstellungen 2						
D Live Werte	Raspberry Einstellungen	lst	Soll	Eingabe			
Live Kurve	Messintervall (PiLogger Interrupt)	5s					
Magramme	Log Faktor	36	36				
🔅 Einstellungen 1	ergibt Log-Intervall	3m 0s	3m 0s				
🔎 Einstellungen 2	Statistik Reset zum Log-Zeitpunkt	ja	ja	~			
لم	Faktor Pulse 1 Sensorkonstante	0.36	0.36				
🛃 Download	Einheit Pulse 1 Haupteinheit	km/h	km/h				
	Faktor Pulse 2 (Neben)	0.1	0.1				
	Einheit Pulse 2 Nebeneinheit	m/s	m/s				
	Typ Temp Sensor	PTC Pt1000	PTC Pt1000	v			
	Split Log-Datei	ja	ja	~			
	Zeilenzahl für Split Log-Datei	3360	3360				
	Neuer Wert ergibt:	7d 0h 0m 0s Z	eit 880	kB			
	Speichern	Reset Energiezähler	Split Lo	g jetzt			
	Neustart	Herunterfahren					

Diese Seite enthält Einstellungen für das Python-Programm selber, ist also die Raspberry Pi (ESP32) Seite der Einstellungen.

Diese Einstellungen werden in einer separaten Datei 'PiLogger_Config.json' im Programmverzeichnis gespeichert.

Einige der PiLogger-Einstellungen werden dort ebenfalls gespeichert, damit zum Beispiel bei geänderter I²C Adresse, das Programm den PiLogger auch ansprechen kann.

Auch hier werden die aktuellen Werte in der linken Spalte 'Ist' mit dem Aufruf der Seite aktuell vom Raspberry abgerufen. In der rechten Spalte 'Soll' werden die neu abzuspeichernden Werte aufgelistet und in der Spalte 'Eingabe' werden entsprechend die Werte geändert.

Die erste Zeile gibt die aktuell eingestellte Zeit für das Messintervall des PiLoggers an. Das ist die Basis für die Logger-Einstellungen. Die Einstellungen im Einzelnen:

• Log Faktor / Log-Intervall

Mit dem hier eingestelltem 'Log Faktor' wird das Messintervall multipliziert um das Log-Intervall einzustellen. Das resultierende Log-Intervall wird in der Zeile darunter angezeigt.

Mit diesem Zeitabstand holt der Raspberry Pi (oder ESP32) die Daten per I²C vom PiLogger One. Die I²C Abfrage aller Daten mit 2 Blockread Operationen dauert 14,7 msec bei 100 kBaud. Da die Programmschleife (je nach

Rechnergeschwindigkeit) bis zu ca. 75 msec selber verbraucht und auch der SD-Karten-Schreibvorgang Zeit benötigt, sollte das Log-Intervall nicht kürzer als ca. 100 ms gewählt werden.

Bei kurzen Intervallen bitte mit dem tatsächlich verwendeten Rechner die Zuverlässigkeit vor dem Dauereinsatz testen. Gegebenenfalls kann auch das zugrunde liegende Messintervall verlängert werden.

Das Abfrage-Intervall ist auch der zeitliche Messpunktabstand der mitgeschriebenen Messwerte. Die Messwerte werden mit Zeitstempel in der Datei 'logdata.csv' im Verzeichnis '/home/user/pilogger' gespeichert.

• Statistik-Reset zum Log-Zeitpunkt

Dieses Feld ist eine Ja/Nein-Entscheidung mittels einer Aufklappliste. Diese Funktion bewirkt, dass jedes Mal, wenn ein Eintrag in die Log-Datei gemacht wird, auch ein Befehl an den PiLogger gesendet wird um die Statistik-Werte 'Mittel', 'Minimum' und 'Maximum' auf die aktuellen Messwerte zurückzusetzen.

Damit werden diese Werte zu den Statistik-Werten die für genau ein Log-Intervall gelten. Das ist insbesondere wichtig für die Energiezähler, die das Produkt von Leistung mal Zeit (-Intervall) pro Log-Intervall aufaddieren.

Aus diesem Grund ist die Standard-Einstellung für diesen Punkt 'aktiv' - also 'ja'.

• Faktor Pulse 1

Hier wird der Umrechnungsfaktor für den Impulszähleingang eingestellt. Die Grundeinheit ist Impulse pro Sekunde. Das Messzeitintervall wird zusätzlich von der PiLogger WebMonitor Software mit berücksichtigt. Hier muss die Fühlerkonstante unter Einbeziehung der Einheit, die im nächsten Feld eingegeben wird, angegeben werden.

Also beispielsweise für einen Windgeber die Wegstrecke pro Impuls multipliziert mit dem Zeitfaktor der angestrebten Anzeige-Einheit.

In dem gezeigten Beispiel also 0,36 km/h - das entspricht 0,1 m pro Impuls pro 3600 Sekunden.

Dieses Feld gilt für die Hauptanzeige (groß) auf der 'Live Werte'-Seite.

• Einheit Pulse 1

Dies ist die zugehörige Maßeinheit für die Umrechnung der Impulszähler-Messwerte.

Dieses Feld gilt für die Hauptanzeige (groß) auf der 'Live Werte'-Seite.

• Faktor Pulse 2

Wie unter 'Faktor Pulse 1', hier aber für die Nebenanzeige (klein) auf der 'Live Werte'-Seite. In der Log-Datei wird nur der Wert für die Haupteinheit gespeichert.

• Einheit Pulse 2

Dieses Feld gilt für die Nebenanzeige (klein) auf der 'Live Werte'-Seite.

• Typ Temp Sensor

Dies ist eine Aufklappliste mit zur Zeit 6 Einträgen: NTC 10k B3928 NTC 10k B3477 NTC Tabelle PTC Pt1000 PTC KTY81-110 PTC Tabelle Je nach angeschlossenem Temperaturfühler wird hier die notwendige Umrechnung der Messwerte in Temperaturwerte festgelegt. Diese Umrechnungsalgorithmen stehen in der WebMonitor Software zur Verfügung.



Achtung : Der Jumper (Steckbrücke) auf dem PiLogger One muss entsprechend der Klasse des angeschlossenen Fühlers gesteckt werden. Werkseitig ist dieser in der Position ,NTC' gesteckt.

Als NTC eignet sich ein Fühler mit 10 kOhm bei 25°C, als PTC ist ein Platinfühler mit 1000 Ohm bei 0°C vorgesehen.

Siehe hierzu auch Kapitel 7.3 'Temperaturfühlereingang'.

• Split Log-Datei

Dieses Feld ist eine Ja/Nein-Entscheidung mittels einer Aufklappliste. Ist diese Funktion aktiviert, wird automatisch nach der Anzahl der geloggten Zeilen, die im nächsten Feld angegeben werden, eine neue Log-Datei angelegt. Ist diese Funktion nicht aktiviert, wird die Log-Datei unendlich fortgeführt. Die Log-Datei kann trotzdem manuell geteilt werden mit dem Knopf 'Split Log jetzt' - siehe unten.

• Zeilenzahl für Split Log-Datei

Hier wird die Anzahl der Zeilen angegeben, bei der eine neue Log-Datei begonnen wird. Jede neu angelegte Log-Datei bekommt im Namen den Zeitstempel der Neuanlage. Je nach Log-Intervall können schnell sehr große Dateien entstehen, deren Übertragung für die Diagramm-Anzeige dann entsprechend lange dauern. Dabei wird für die Diagramm-Anzeige tatsächlich eine noch größere Datei übertragen, die zusätzlich noch die Daten der vorherigen Log-Datei enthält. Damit ist direkt nach der automatischen Anlage einer neuen Log-Datei zumindest der Zeitraum davor als Kurve anzeigbar.

Die Zeile darunter zeigt an welcher Zeitraum und welche Dateigröße je Log-Datei mit den Eingabewerten entsteht.

• Knopf 'Speichern'

Mit Betätigung diese Knopfes werden die Werte aus der Spalte 'Soll' in der Konfigurationsdatei 'PiLogger_Config.json' gespeichert. Zusätzlich wird das Programm mit diesen Werten neu initialisiert.

• Knopf 'Reset Energiezähler'

Dieser Knopf setzt sofort alle Energiezähler auf Null, also sowohl die Dauerzähler als auch die Tageszähler. Inklusive der zugehörigen Zeitzähler.

• Knopf 'Split Log jetzt'

Wird dieser Knopf betätigt, wird sofort eine neue Log-Datei angelegt. Die bisherige Datendatei ist dabei weiterhin herunterladbar und auch anzeigbar – siehe 6.8 'Die Seite Download'.

Die Daten der bisherigen Log-Datei sind nun der Anzeigen-Vorlauf für die Zeitdiagramme.

Soll auch dieser Teil nicht mehr angezeigt werden, zum Beispiel, weil die Einstellungen stark geändert wurden (andere Impuls-Einheiten, andere Kalibrierwerte etc.), muss der Knopf 'Split Log jetzt' noch einmal gedrückt werden. Dabei entsteht eine unnütze Log-Datei ohne Messwerte oder nur sehr wenigen Messwertzeilen. Sie lässt sich gegebenenfalls auf der Seite 'Download' löschen.

• Knopf 'Neustart'

Dieser Knopf bietet die Möglichkeit den Raspberry kontrolliert neu zu starten. Im Gegensatz zum Trennen der Stromversorgung wird mit dieser Methode ein Datenverlust oder gar Unbrauchbar machen der SD-Karte vermieden.

• Knopf 'Herunterfahren'

Dieser Knopf bietet die Möglichkeit den Raspberry geordnet herunterzufahren. Im Gegensatz zum Trennen der Stromversorgung wird mit dieser Methode ein Datenverlust oder gar Unbrauchbar machen der SD-Karte vermieden. Nach den üblichen letzten Blink-Zeichen (4 mal gleichmäßig) der Raspberry Pi LED kann die Stromversorgung gefahrlos getrennt werden.

6.7 Die Seite Kalibration

PiLogger						
Menu	Kalibration					
D Live Werte	Temperatur Korrekturwerte	lst		Soll E	ngabe	
Live Kurve	Offset NTC	0.00	°C	0.00		
M Diagramme	Faktor NTC	1.0000		1.0000		
Einstellungen 1	Offset PTC	0.50	°C	0.50		
	Faktor PTC	1.0000		1.0000		
CALIBRATION	Warnung Bevor Sie die nach - Für die Strommes unter 'Einstellun - Abweichungen gr Zu schwach dim	nfolgende ssung fül gen 1' du rößer ±1	n Ko nren s urch. % sin	rrekturmö Sie unbedi Id ein Hinv	ylichkeiten verwenden, sollten Sie folgendes bedenken: ngt vorher den Nullpunktabgleich 'Amp-Zero' reis auf Aufbauprobleme: der ungewollte Masseströme können Ursache für Brand	gefahr sein
CAlibration	Warnung Bevor Sie die nach - Für die Stromme: unter 'Einstellun - Abweichungen gr Zu schwach dim - Die nachträgliche Spannung/Strom Korrekturwerte	nfolgende ssung fül gen 1' du rößer ±1 ensionier e Korrekt	n Ko nren S nrch. % sin te Ka ur de	rrekturmö Sie unbedi Id ein Hinv abel und/o r Leistung Soll	plichkeiten verwenden, sollten Sie folgendes bedenken: ngt vorher den Nullpunktabgleich 'Amp-Zero' reis auf Aufbauprobleme: der ungewollte Masseströme können Ursache für Brand swerte verringert deren Genauigkeit und die der Energi Eingabe	gefahr sein. ezähler.
Download	Warnung Bevor Sie die nach - Für die Strommes unter 'Einstellun - Abweichungen gr Zu schwach dim - Die nachträgliche Spannung/Strom Korrekturwerte Volt/Amp Korrektur Aktivieren	nfolgende ssung fül igen 1' du rößer ±1 ensionier e Korrekt Ist nein	n Ko nren S Irch. % sin te Ka ur de	rrekturmö Sie unbedi ad ein Hinv abel und/o r Leistung Soll nein	plichkeiten verwenden, sollten Sie folgendes bedenken: ngt vorher den Nullpunktabgleich 'Amp-Zero' reis auf Aufbauprobleme: der ungewollte Masseströme können Ursache für Brand swerte verringert deren Genauigkeit und die der Energi Eingabe	lgefahr sein. ezähler.
Download	Warnung Bevor Sie die nach - Für die Strommer unter 'Einstellun - Abweichungen gr Zu schwach dim - Die nachträgliche Spannung/Strom Korrekturwerte Volt/Amp Korrektur Aktivieren Offset Spannung	nfolgende ssung fül gen 1' du rößer ±1 ensionier e Korrekt Ist nein 0.0000	n Ko nren S urch. % sin te Ka ur de	rrekturmö Sie unbedi Id ein Hinv abel und/o r Leistung Soll nein 0.0000	plichkeiten verwenden, sollten Sie folgendes bedenken: ngt vorher den Nullpunktabgleich 'Amp-Zero' der ungewollte Masseströme können Ursache für Brand swerte verringert deren Genauigkeit und die der Energi Eingabe	lgefahr sein. ezähler.
 ✔ Kalibration ✔ Download 	Warnung Bevor Sie die nach - Für die Strommer unter 'Einstellun - Abweichungen gr Zu schwach dim - Die nachträglicher Spannung/Strom Korrekturwerte Volt/Amp Korrektur Aktivieren Offset Spannung Faktor Spannung	nfolgende ssung fül gen 1' du rößer ±1 ensionie e Korrekt Ist nein 0.0000 1.0000	n Ko nren S irch. % sin te Ka ur de V	rrekturmö Sie unbedi id ein Hinva abel und/o r Leistung Soll nein 0.0000 1.00000	plichkeiten verwenden, sollten Sie folgendes bedenken: ngt vorher den Nullpunktabgleich 'Amp-Zero' reis auf Aufbauprobleme: der ungewollte Masseströme können Ursache für Brand swerte verringert deren Genauigkeit und die der Energi Eingabe	lgefahr sein. ezähler.
 ▲ Kalibration ▲ Download 	Warnung Bevor Sie die nach - Für die Strommes unter 'Einstellun - Abweichungen gi Zu schwach dim - Die nachträgliche Spannung/Strom Korrekturwerte Volt/Amp Korrektur Aktivieren Offset Spannung Faktor Spannung Offset Strom	nfolgende ssung fül igen 1' du rößer ±1 ensionier e Korrekt Ist nein 0.0000 1.0000 0.0000	n Ko nren : urch. % sin te Ka ur de V	rrekturmö Sie unbedi d ein Hinv abel und/or r Leistung Soll nein 0.0000 1.00000 0.0000	plichkeiten verwenden, sollten Sie folgendes bedenken: ngt vorher den Nullpunktabgleich 'Amp-Zero' reis auf Aufbauprobleme: der ungewollte Masseströme können Ursache für Brand swerte verringert deren Genauigkeit und die der Energi	lgefahr sein. ezähler.

Die Seite 'Kalibration' bietet die Möglichkeit Korrekturwerte für die Temperaturmessung und auch für die Spannungs- und Strommessung einzustellen.

Dabei handelt es sich grundsätzlich um eine einfache lineare Umrechnung mit einer Verschiebung (Offset) und einer Steigung (Faktor).

Für die Temperaturmessung ist diese Korrekturrechnung immer sinnvoll und ermöglicht etwa eine klassische 2-Punkt-Kalibration entsprechend der Definition der Celsius-Skala:

Dafür setzt man zuerst den Temperaturfühler einem Eiswasserbad aus (eine gute, wasserdichte Isolation der Anschlüsse vorausgesetzt) und nimmt den erhaltenen Anzeigewert negativ (mal -1) als 'Offset'. Anschließend wird der Temperaturfühler siedendem Wasser ausgesetzt und der 'Faktor' wird auf '100/Anzeigewert' gesetzt.

Die Standardeinstellung ist für beide Temperaturfühlerklassen:

'0' für den Offset und '1' für den Faktor \rightarrow das bedeutet : keine Änderung.

Die Korrektur der Spannungs- und Stromwerte dagegen sollte nur aktiviert werden, wenn folgendes beachtet wird:

Bei Abweichungen der Stromwerte immer zuerst den Nullpunkt-Abgleich 'Amp-Zero' unter 'Einstellungen 1' durchführen – siehe Seite 77. Wenn die Abweichungen der Spannungsmessung größer ±1% des Messwertes sind, ist das ein Hinweis auf ein mögliches Aufbauproblem:

- zu schwach dimensionierte Anschlusskabel führen zu Spannungsabfällen und können durch Eigenerwärmung ein Sicherheitsproblem darstellen. Der PiLogger misst die Spannung zwischen seinen 'OUT+' und 'OUT-' Klemmen. Durch einen Spannungsabfall am Zuleitungskabel kann diese Spannung von der Spannung am Generator oder Akku abweichen. Stellen Sie sicher, dass die Kabel nicht überlastet sind und auch die Kontakte gut verbunden sind (keinen nennenswerten Spannungsabfall erzeugen). Ebenso sollten die Kabel immer so kurz wie möglich gehalten werden. Soll also, zum Beispiel, die Akkuspannung als Referenz genommen werden, ist es sinnvoll den PiLogger näher am Akku zu montieren als am Generator.

- zusätzliche Ströme auf dem Minus-Kabel des Akkus, die nicht auch auf der vom PiLogger gemessenen Plus-Leitung fließen, führen zu einer Spannungsanhebung an der Messstelle des PiLoggers – einem Masseversatz.

Wird die Spannungs- und Stromwert-Korrektur aktiviert, werden auch die Leistungswerte, die der PiLogger direkt berechnet hat, entsprechend nachträglich korrigiert. Das bedeutet zusätzliche Divisions- und Multiplikationsoperationen, die die Genauigkeit der Leistungswerte verringern. Und damit natürlich auch die Genauigkeit der Energiezähler.

Generell ist ein optimierter Aufbau einer nachträglichen Korrekturrechnung vorzuziehen.

Deshalb ist die Standardeinstellung hier 'nein' für 'Korrekturrechnung deaktiviert'.

• Knopf 'Speichern'

Mit Betätigung diese Knopfes werden die Werte aus der Spalte 'Soll' in der Konfigurationsdatei 'PiLogger_Config.json' gespeichert. Zusätzlich wird das Programm mit diesen Werten neu initialisiert.

• Knopf 'Zurücksetzen'

Mit diesem Knopf werden alle Werte dieser Seite auf ihre Standardwerte zurückgesetzt und auch *sofort* gespeichert.

6.8 Die Seite Download



Diese Seite ermöglicht es die Logdaten vom Raspberry (oder ESP32) herunterzuladen, anzusehen und auch zu löschen.

Alle Logdateien sind '.csv'-Dateien. Um sie in ein deutschsprachiges Tabellenkalkulationsprogramm laden zu können, muss jedoch mit einem Textprogramm eine Ersetzung von '.' zu ',' durchgeführt werden. Die '.csv'-Dateien verwenden für die Zahlendarstellung die amerikanisch-englische Darstellung, um mit der Diagramm-Bibliothek kompatibel zu sein. Das führt ohne diese Ersetzung zu einer Fehlinterpretation beim Import. Als Feld-Trennzeichen wird das Semikolon (;) verwendet, was in der Regel vom Importfilter erkannt wird, manchmal aber extra gesetzt werden muss.

Mit Aufruf der Seite wird das Verzeichnis aller verfügbaren 'logdata_xxx.csv' Dateien frisch vom Raspberry (ESP32) abgefragt.

Hinter jeder dieser Dateien gibt es 3 Schaltflächen:

Die erste blaue Fläche mit dem Download-Symbol startet einen normalen Datei-Download für diese Datei. Je nach Konfiguration des aufrufenden Browsers erscheint dabei ein Dialog-Fenster mit der Frage ob und wohin die Datei gespeichert werden soll.

Die zweite gelbe Fläche mit dem Augen-Symbol ermöglicht es, diese Datei mit der Diagramm-Funktion anzuschauen. Dabei wird, wie gewohnt, die Seite Diagramme aufgerufen und standardmäßig die Durchschnittstemperatur dargestellt. Damit können Daten aus zurückliegenden Zeiträumen durchgesehen werden. Die dritte rote Fläche mit dem X-Symbol löscht diese Datendatei von der SD-Karte des Raspberry Pi (ESP32). Da dieser Vorgang endgültig ist, wird zunächst eine Sicherheits-Rückfrage per Overlay-Fenster zwischengeschaltet, um einen Abbruch bei versehentlicher Betätigung zu ermöglichen:



Unter dem Datei-Verzeichnis gibt es noch die Schaltfläche 'aktuelle Daten-Datei herunterladen'. Damit lässt sich die aktuell verwendete Log-Datei 'logdata.csv' herunterladen. Da diese Datei zur Zeit zum Loggen benutzt wird, kann man sie natürlich nicht löschen, und auch das Anzeigen ist an dieser Stelle nicht sinnvoll, da dies über den normalen Aufruf der Diagramme-Seite erfolgt.

7 Anschlussklemmen

Die Messeingänge des PiLoggers liegen alle auf dem 8 poligen Klemmenblock:



7.1 Gleichspannungs- und Strommessung

Links die ersten 4 Klemmen sind für die Messung von Spannung und Strom so beschaltet, dass die beiden ersten Klemmen den Eingang und die beiden nächsten Klemmen den Ausgang für eine Durchschleifung bilden. Dabei sind die beiden mit '-' gekennzeichneten Klemmen direkt miteinander verbunden, während bei den mit '+' gekennzeichneten Klemmen der Stromsensor in Serie geschaltet ist:



Der Innenwiderstand des Stromsensors ist mit ca. 2 m Ω schon wirklich gering, dennoch gehen etwa bei 12 A also 24 mV an ihm verloren, was einer Leistung von 288 mW entspricht.

Das ist also ein systematischer Messfehler dieser Anordnung.

Bei kleinen Spannungen und hohen Ströme ist er am größten. Im obigen Beispiel bei 12 A und 3 V beträgt er aber mit 0,008 weniger als 1%.

Dass für Ströme bis zu 12 A Kabel mit ausreichendem Querschnitt verwendet werden müssen, erklärt sich auch aus dieser Überlegung des Verlustes an scheinbar kleinen Widerständen.

Auch wenn nach VDE eine 0,75 mm² Einzel-Leitung (90°C) mit bis zu 12 A belastbar ist, so reduziert sich diese Belastbarkeit bei 85°C Umgebungstemperatur auf 35% !

Wird also der Aufbau, z.B. durch Sonneneinstrahlung, erhitzt, braucht man deutlich dickere Kabel – in diesem Fall 4 mm² – um einen sicheren Aufbau zu haben.

Die Klemmen des PiLogger können diese 4 mm² (Litze mit Aderendhülse) aufnehmen.

Ein Kabel mit 4 mm² Kupfer-Querschnitt hat bei einer Länge von 4 m bereits einen Widerstand von ca. 17,2 m Ω und verursacht somit bei 12 A einen Verlust von fast 2,5 W !

Und das passiert im Rückleiter nochmal, macht also fast 5 W Verlust !

Also bitte auch die Kabellänge immer möglichst kurz halten.

7.2 Impulszähleingang

Die nächsten beiden Klemmen sind mit 'GND' und 'Imp' beschriftet. Sie bilden die Anschlüsse für den Impulszähleingang.

Der Eingang ist für die Verwendung mit mechanischen Schaltern, wie etwa Reed-Kontakten, sowie Open-Collector-Ausgängen (OC) ausgelegt. Im Ruhezustand zeigt er etwa 3,1V am Anschluss 'Imp' gegen die Bezugsmasse 'GND'. Wird der Anschluss 'Imp' gegen 'GND' kurzgeschlossen, wird diese Flanke als ein Impuls gezählt.

Seine Hauptelemente sehen so aus:



Der Pull-Up Widerstand bildet mit dem Kondensator eine Zeitkonstante für die

Erholung des Eingangs für die nächste negative Flanke. Der Serien-Widerstand im Eingang bildet mit dem Kondensator einen Tiefpass. Beides zusammen sorgt für eine Entprellung und eine Verringerung der Empfindlichkeit gegen Störimpulse. Bei 50% Puls-Pause-Verhältnis liegt die maximale Frequenz etwa bei 2 kHz. Der Widerstand des geschlossenen Schalters (bzw. durchgesteuerten Open Collectors) muss unter 3 k Ω liegen, um eine sichere Low-Erkennung noch zu ermöglichen.

7.3 Temperaturfühlereingang

Die beiden rechten Klemmen sind mit 'GND' und 'Temp' beschriftet. Sie sind für den Anschluss eines Temperaturfühlers vorgesehen.

Der Eingang lässt sich mit Hilfe der Kurzschlussbrücke JP1 auf 2 Typen von Temperaturfühlern einstellen:



In dieser Stellung ist die Beschaltung für den Anschluss eines NTC (Widerstand mit <u>N</u>egative <u>T</u>emperature <u>C</u>oefficient) mit 10 k Ω Nennwiderstand ausgelegt.

2. JP1 in Stellung 2-3 : PTC



In dieser Stellung ist die Beschaltung für den Anschluss eines PTC (Widerstand mit <u>P</u>ositive <u>T</u>emperature <u>C</u>oefficient) mit 1 k Ω Nennwiderstand ausgelegt.

Mehr Details zu Temperaturfühlern folgen in Kapitel 10.

8 Anschlussbelegung GPIO-Leiste

Der PiLogger verwendet lediglich 5 der ersten 8 Pins der Raspberry Pi GPIO-Leiste.

Es sind zwar noch weitere Pins angeschlossen – dies aber nur zur Erhöhung der Kontaktsicherheit (3,3V & GND) und als Alternative (/RDY).



Mit dem Signal '/RDY' (= busy) informiert der PiLogger One den Raspberry Pi über seine Messaktivität. Der Zustand entspricht dem der LED am PiLogger. Ein 'High' (+3,3V) bedeutet, dass gerade eine Messung läuft – die LED leuchtet. Wenn das Signal auf 'Low' wechselt, ist die Messung abgeschlossen – die LED erlischt.

Diese negativ gehende Flanke (falling edge) wird am GPIO4 (Pin 7) des Raspberry Pi als Interrupt konfiguriert, um die frischen Messwerte mit dem korrekten Zeitstempel abzuholen und zu loggen.

9 Anwendungs-Programmierschnittstelle

Der PiLogger One benutzt die I²C Schnittstelle um mit dem Raspberry Pi (oder ESP32) zu kommunizieren. Dabei ist der Raspberry Pi der Master und der PiLogger der Slave bei dieser Verbindung.

Der I²C Bus (bzw. SMBus) verwendet dabei das durch die Spezifikation vorgegebene Hardware-Protokoll

(http://www.nxp.com/documents/user_manual/UM10204.pdf).

Der SMBus ist dabei eine Weiterentwicklung mit einigen Ergänzungen, die aber von reinen I²C Geräten (ICs) nicht oder nur teilweise unterstützt werden.

Dieses Protokoll wird in Linux durch den Kernel-Treiber 'i2c-dev' behandelt und mit dem Modul 'i2c-dev' aus dem 'user land', also dem Bereich der Applikationen, als Gerät zugänglich gemacht.

Über den Wrapper 'python3-smbus' bekommt Python dann Zugriff auf den I²C Bus über den Kernel-Treiber. Alternativ können die 'I2C-Tools' von der Kommandozeile aus benutzt werden (siehe Anhang).

Die beiden Letzteren stellen einen Satz von Funktionen (Methoden) zur Verfügung, mit dem alle notwendigen Operationen auf dem I²C Bus bzw. SM Bus durchgeführt werden können.

Nachfolgend nun der Nachrichtenkatalog, der die Kommunikation mit dem PiLogger One definiert. Da der I²C Bus bzw. SM Bus byte-orientiert ist, werden alle Zugriffe über virtuelle 1-Byte-Adressen gesteuert – die Register.

Der PiLogger One hat zwei Klassen von Registern:

Konfigurationsregister und Messwertregister.

Zusätzlich gibt es noch direkte Kommandos, die durch Schreiben an das betreffende Register ausgelöst werden (mit und ohne Passwort).

9.1 Konfigurationsregister

Die nachfolgende Tabelle listet alle Konfigurationsregister des PiLogger One auf. Diese Register liegen im Bereich 0x01 bis 0x16.

Ihre jeweilige Bedeutung ist bereits in Kapitel 6.5 beschrieben worden.

Die Register von 0x01 bis 0x06 sind gültige Adressen für einen schreibenden Zugriff.

Die korrespondierenden Register von 0x11 bis 0x16 sind die jeweiligen Adressen für einen Lesezugriff.

Diese Register halten jeweils 1 Byte. Das heißt, es wird beim Schreiben genau 1 Datenbyte erwartet und beim Lesen genau 1 Datenbyte zurückgegeben.

Zusätzlich gibt es 3 Befehle (0x07 , 0x0D , 0x0F), die durch Schreiben an dieses Register ausgelöst werden, wenn als Wert (data byte) der erwartete fixe Wert –

Lesen/Schreiben Register 1 Byte Konfiguration Write Wert Read I²C Slave Adresse 0x04...0x7C 0x01 0x11 default 0x48 0 = 32768 = 1 sec 1 = 8192 = 250 ms Messzyklus Basis 0x02 0x12 2 = 512 = 15,625 ms3 = 64 = 1,953125 ms Messzyklus Faktor 1 0x03 0x13 1 – 255 low 1...65535 Messzyklus Faktor 2 0x04 0x14 0 – 255 high 2^x : 2...128 Mittelungsfaktor 0x05 0x15 1 - 7Zeitexpansion Pulszählung 0x06 0x16 1 – 255 Amp Avg zu Null setzen 0x07 0x44 Nur Schreiben -Auf Werkseinstellung 0x0D 0xAB Nur Schreiben Konfig Speichern (Flash) 0x0F 0x55 Nur Schreiben -

das Passwort – mitgesendet wird. Dies soll die Wahrscheinlichkeit für ein versehentliches Ausführen, z.B. durch eine gestörte Kommunikation, senken.

Die Verwendung in Python sieht dann wie folgt aus:

Die erste Anweisung steht ganz am Anfang des Programms und bindet das Modul 'SMBus' ein. Die zweite Anweisung weist der Variablen 'pilogger' den Wert des zu benutzenden Busses zu.

import smbus
pilogger = smbus.SMBus(1) ; für Raspberry Pi Typ A (256 Mbyte): smbus.SMBus(0)

Lesen eines 1-Byte-Registers:

wert = pilogger.read_byte_data(address, register)

Beispiel:

timebase = pilogger.read_byte_data(0x48, 0x12)

Schreiben eines 1-Byte-Registers:

pilogger.write_byte_data(address, register, databyte)

Beispiel:

```
pilogger.write_byte_data(0x48, 0x02, 0x01)
```

9.2 Messwertregister

Die nachfolgende Tabelle listet alle Messwertregister des PiLogger One auf. Diese Register liegen im Bereich 0x20 bis 0x88.

Die Register im Bereich von 0x20 bis 0x63 ohne [0x28,0x38,0x48,0x58] sind gültige Zugriffsadressen für einen lesenden Zugriff.

Diese Register halten jeweils 2 Byte. Das heißt es werden beim Lesen genau 2 Datenbytes zurückgegeben. Diese 2 Byte bedeuten jeweils ein Word – also eine 16 Bit Zahl.

Dabei sind die Messwerte im Registerbereich 0x20 bis 0x27 (Leistung) jeweils paarweise zu 4 Byte Werten, also 32 Bit Zahlen zusammenzufassen.

Die Register 0x70 und 0x80 leiten jeweils das Lesen eines ganzen 32 Byte Blockes ein.

Diese 2 Blöcke überlappen sich zwar, so dass beim Lesen beider Blöcke die Werte für Temperatur und Impulszähler doppelt gelesen werden – dennoch ist diese Abfrage-Methode die schnellste und effektivste, wenn alle Werte benötigt werden.

Zusätzlich gibt es 7 Befehle (0x28...0x88), die durch Schreiben an dieses Register ausgelöst werden. Schreiben an diese Register setzt jeweils die Statistik-Werte der zugehörigen Basis-Registers zurück (auf den aktuellen Wert).

Nur Lesen	Register	2 Byte			Avg, Min & Max
Messwerte	Momentan	Mittelwert	Minimum	Maximum	Reset
Leistung	0x20 0x21	0x22 0x23	0x24 0x25	0x26 0x27	0x28
Temperatur	0x30	0x31	0x32	0x33	0x38
Impulseingang	0x40	0x41	0x42	0x43	0x48
Spannung	0x50	0x51	0x52	0x53	0x58
Strom	0x60	0x61	0x62	0x63	0x68

Block Lesen	Befehl	Register		Reset
Block1 Lesen 16x Word	0x70	0x200x43		0x78
Block2 Lesen 16x Word	0x80	0x300x63		0x88

Die Verwendung in Python sieht dann wie folgt aus:

Die erste Anweisung steht ganz am Anfang des Programms und bindet das Modul 'SMBus' ein. Die zweite Anweisung weist der Variablen 'pilogger' den Wert des zu benutzenden Busses zu.

```
import smbus
pilogger = smbus.SMBus(1) ; für Raspberry Pi Typ A (256 Mbyte):
smbus.SMBus(0)
```

```
Lesen eines 2-Byte-Registers:
```

wert = pilogger.read_word_data(address, reg)

Beispiel:

tempmittel = pilogger.read_word_data(0x48, 0x31)

Lesen eines 4-Byte-Wertes:

```
wertL = pilogger.read_word_data(address, reg)
wertH = pilogger.read_word_data(address, reg+1)
wert = wertH * 65536 + wertL
```

Beispiel:

```
LeistungMaxL = pilogger.read_word_data(0x48, 0x26)
LeistungMaxH = pilogger.read_word_data(0x48, 0x27)
LeistungMax = LeistungMaxH * 65536 + LeistungMaxL
```

Lesen eines 32-Byte-Blocks:

```
werte = pilogger.read_i2c_block_data(address,blockreg) ; block 1 = 0x70, block 2 = 0x80
```

Beispiel:

```
werte = pilogger.read_i2c_block_data(0x48,0x70)
```

Zurücksetzen der Statistikwerte eines Blocks:

```
pilogger.write_byte(address,blockreg) ; block 1 = 0x78 , block 2 = 0x88
```

Beispiel:

```
pilogger.write_byte(0x48,0x78)
```

9.3 Wertebereiche und Skalierung

Die Messresultate werden vom PiLogger als Rohwerte mit unterschiedlichen Wertebereichen ausgegeben. Die nachfolgende Tabelle gibt für jeden Kanal den Wertebereich und die zugehörigen Skalierungsfaktoren an:

Messwert	Format	Wertebereich	Standard Maßfaktoren				
Leistung	signed 32 bit	-2147483648 +2147483647	918/2147483648	W	-918+918	1/2339236	
Temperatur	unsigned 16 bit	0+65535	0,00915541	mV	0600mV	1/109,2267	
Impulseingang	unsigned 16 bit	0+65535	0,2	m/s			
Spannung	unsigned 16 bit	0+65535	0,000915541	V	060V	1/1092,267	
Strom	signed 16 bit	-32768 +32767	0,000466933	А	-15,3+15,3A	1/2141,634	

Dies sind die jeweiligen Vollausschlag-Grenzen (full scale), die nutzbaren Messbereiche sind immer etwas kleiner (siehe 'Technische Daten').

Eine Ausnahme ist der Impulszählkanal. Hier ist der Rohwert immer Pulse pro

Messintervall – die Interpretation hängt vom verwendeten Sensor ab (siehe 6.6 'Einstellungen 2 – Faktor Pulse'). Der Standard Maßfaktor ist hier nur die Vorgabe aus der WebMonitor-Software, die natürlich angepasst werden kann.

Die Temperaturmessung ist für den PiLogger intern eine Spannungsmessung. Die Umrechnung in Temperaturwerte hängt vom verwendeten Temperaturfühler ab. Die Umrechnung erfolgt deshalb in der WebMonitor-Software.

10 Temperaturfühler-Tabellen

Der PiLogger One Temperaturfühlereingang ist für zwei gebräuchliche Temperaturfühler optimiert. Das Eine ist ein sogenannter Pt1000, ein Sensor der auf Platin (Pt) basiert, einen positiven Temperaturkoeffizienten (PTC) hat und in der Ausführung Pt1000 bei 0°C einen typischen Widerstand von 1000 Ω zeigt. Dieser Typ ist recht genau und standardisiert.

Das Andere ist ein sogenannter NTC, also ein temperaturabhängiger Widerstand mit negativem Temperaturkoeffizienten, von dem es leider sehr viele verschiedene Varianten gibt. In der Einstellung 'NTC' ist der PiLogger Eingang für einen Typ optimiert, der im Heizungsbau recht gängig ist und bei 25°C einen typischen Widerstand von 10 k Ω hat (z.B. Epcos B57891S0103H008). Ein NTC überstreicht typischerweise einen größeren Widerstandsbereich, weshalb die Messung etwas einfacher wird – dafür ist die Charakteristik deutlich nichtlinearer, was wiederum für mehr Approximierungsaufwand sorgt. Beide Typen haben weitere Vor- und Nachteile, die wir hier nicht weiter vertiefen, aber durch die hohe Messauflösung des PiLoggers (16 Bit) können mit beiden Typen sehr genaue Temperaturmessungen erzielt werden.

Die nachfolgende Tabelle zeigt für die beiden typischen Vertreter die Widerstände bei Temperaturen im Bereich -55°C bis 155°C :

	Pt	10	00		NTC 10k				
T [°C]	R [Ω]		T [°C]	R [Ω]	T [°C]	R _{nom} [Ω]		T [°C]	R _{nom}
-55	783,30		55	1213,24	-55	878900		55	303
-50	803,14		60	1232,46	-50	617590		60	253
-45	822,96		65	1251,64	-45	439340		65	212
-40	842,74		70	1270,79	-40	316180		70	179
-35	862,50		75	1289,92	-35	230060		75	151
-30	882,23		80	1309,01	-30	169150		80	129
-25	901,93		85	1328,07	-25	125550		85	110
-20	921,60		90	1347,10	-20	94143		90	94
-15	941,25		95	1366,09	-15	71172		95	80
-10	960,86		100	1385,06	-10	54308		100	69
-5	980,44		105	1403,98	-5	41505		105	60
0	1000,00		110	1422,87	0	32014		110	52
5	1019,53		115	1441,72	5	25011		115	45
10	1039,03		120	1460,54	10	19691		120	40
15	1058,50		125	1479,31	15	15618		125	35
20	1077,94		130	1498,04	20	12474		130	30
25	1097,35		135	1516,74	25	10000		135	27
30	1116,74		140	1535,38	30	8080		140	240
35	1136,09		145	1553,99	35	6569		145	21
40	1155,42		150	1572,55	40	5372		150	189
45	1174,73		155	1591,06	45	4424		155	16
50	1194,00				50	3661			

Handbuch PiLogger One

Mit der Beschaltung aus dem Kapitel 7.3 ergeben sich damit diese Werte-Tabellen:

Pt1000							
T[°C]	Wert		T[°C]	Wert			
-55	25718		55	38262			
-50	26319		60	38800			
-45	26918		65	39334			
-40	27513		70	39866			
-35	28106		75	40396			
-30	28695		80	40923			
-25	29281		85	41447			
-20	29864		90	41968			
-15	30444		95	42487			
-10	31021		100	43002			
-5	31595		105	43516			
0	32167		110	44026			
5	32735		115	44534			
10	33300		120	45039			
15	33863		125	45541			
20	34422		130	46040			
25	34979		135	46537			
30	35533		140	47031			
35	36085		145	47522			
40	36633		150	48010			
45	37179		155	48496			
50	37722						

NTC 10k								
T[°C]	Wert		T[°C]	Wert				
-55	61682		55	16767				
-50	61441		60	14645				
-45	61114		65	12772				
-40	60679		70	11126				
-35	60107		75	9680				
-30	59367		80	8423				
-25	58424		85	7328				
-20	57247		90	6382				
-15	55797		95	5563				
-10	54054		100	4854				
-5	51942		105	4249				
0	49509		110	3726				
5	46824		115	3272				
10	43884		120	2878				
15	40748		125	2539				
20	37484		130	2245				
25	34124		135	1988				
30	30815		140	1763				
35	27607		145	1566				
40	24561		150	1395				
45	21740		155	1244				
50	19144							

Als Diagramm sieht das dann so aus:



Ausgabe V2.2 , 29.06.2024

Diese Tabellen können benutzt werden, um in der Anwendungs-Software für einen PiLogger-Messwert die zugehörige Temperatur nachzusehen (look up table). Zwischenwerte müssen dabei durch lineare Interpolation berechnet werden.

Die PiLogger WebMonitor-Software geht einen anderen Weg. Sie berechnet für jeden Messwert den Temperaturwert mithilfe einer Approximierungsfunktion (Formel für einen Annäherungswert für die gegebene Kurve).

```
Für den Pt1000 benutzt sie:
Messwert M in 25000...50000:
T = M<sup>2</sup> / 23006845,6 + M / 167,473355 – 237,0911
(quadratische Approximation)
```

Für den NTC 10k reicht das nicht aus. Die Kurve ändert sich so stark über die Temperaturbereiche, dass mit einer einzigen Näherungsformel die Abweichung zur Originalkurve zu groß würde. Deshalb wird hier der Bereich von -40°C bis 140°C in 3 Teilbereiche aufgeteilt, für die jeweils eine andere Näherungsformel benutzt wird:

Mit m = M / 65536 ; Messwert M in 1763...8423 (+140...+80 °C) : T = m³ * -50245,929 + m² * 16224,33 + m * -2068,25 + 184,196 Messwert M in 8424...54054 (+80...-10 °C) : T = m³ * -247,0087 + m² * 394,7776 + m * -305,093 + 111,6473 Messwert M in 54055...60679 (-10...-40 °C) :

```
T = m^3 * -18593,11 + m^2 * 47031,885 + m * -39846,685 + 11292,757
(kubische Approximation)
```

Das ist zwar insbesondere für den NTC eine Menge Rechenaufwand, aber für den Raspberry Pi kein Problem.

Die Genauigkeit der Temperaturmessung ist damit zwar schon sehr gut, aber sie hängt noch immer entscheidend vom Einzelexemplar des verwendeten Fühlers ab. Die Abweichungen können schnell +/- 3° oder mehr betragen. Für genaue Absolutmesswerte sollte also eine Kalibration durchgeführt werden.

Zum Beispiel eine 2-Punkt-Kalibration gegen ein bekannt gutes (geeichtes) Flüssigkeitsthermometer (Alkoholsäule).

```
Oder die klassische 2-Punkt-Kalibration mit Eiswasserbad (0°C) und kochendem
Wasser (100°C) – die Definition der Celsius-Skala. Dafür wird bei 0°C der
Korrekturwert ermittelt, der zum Rohmesswert addiert werden muss, damit der
Anzeigewert korrekt 0°C ist (der Offset) und anschließend der lineare
Korrekturwert mit dem der Zwischenwert multipliziert werden muss, damit dann
zusätzlich die Anzeige bei 100°C passt. Damit sollte dann im Bereich -10°C bis
+120°C eine Genauigkeit von 0,1°C erreicht werden können.
```

11 Technische Daten

Abmessungen : 56 x 47 x 29 mm (L x B x H)

Gewicht : 29 g

Stromversorgung : 3,3 V vom Raspberry Pi über Pfostenverbinder

Stromverbrauch : 5 ... 6 mA

Messungsdauer ca. 84 ms (Python Schleife; abhängig von der Rechengeschwindigkeit des Gast-Rechners)

Abfrage aller Werte (2 Blöcke) : ca. 14,7 ms (@100 kbaud , BCM2708)

Intervallbereich : 100 msec ... 18h 12min 15sec

Messbereiche:

Spannung : 0 ... 50 V Gleichspannung (max. 60 V, Auflösung < 1 mV)

Strom : -12 ... +12 A Gleichstrom (max. +/- 15 A, Auflösung < 0,5 mA)

Leistung : -600 W ... +600 W (max. +/- 900 W)

Impulszähler : 0 ... 65535 Cnt/Intervall

Temperatur : abhängig vom verwendeten Fühler (s. Kap. 10), Auflösung < 0,01 K

12 Kalibrierung

Obwohl der PiLogger mit eng tolerierten Bauteilen aufgebaut ist und mit einem 16 Bit A/D-Wandler arbeitet, ist er *kein* kalibriertes Messinstrument. Der PiLogger ist ab Werk *nicht* nach irgendeinem Standard kalibriert. Für Präzisionsmessungen sollte jedes Exemplar gegen ein geeignetes Messnormal kalibriert werden.

Die daraus resultierenden Korrekturwerte können dann einfach in der Software, die auf dem Raspberry Pi (o. ESP32) läuft, eingebaut werden (also zum Beispiel im Python-Programm der WebMonitor Software), um so zu erreichen, dass die angezeigten, bzw. in der Log-Datei abgelegten Werte so genau wie möglich sind.

Wie in Kapitel 6.7 'Die Seite Kalibration' beschrieben, bietet die WebMonitor Software bereits in der Benutzeroberfläche eine Möglichkeit für eine einfache lineare Korrektur der Messwerte.

Das ist etwa für die Temperaturwerte in der Regel völlig ausreichend, da die Charakteristik der verwendeten Temperaturfühler mithilfe der typischen Kurven bereits gut angenähert ist und so nur die Exemplarstreuungen von Sensor und PiLogger korrigiert werden müssen.

Nachfolgend werden beispielgebende Rezepte beschrieben, wie die 4 Basis-Messungen kalibriert werden können.

12.1 Kalibrierung Temperaturmessung

Wenn der verwendete Temperaturfühler wasserdicht ist, bietet sich eine Kalibrierung nach der klassischen Celsius-Skala an. Die Abweichungen dieser historischen Skala sind gegenüber der seit 2019 eingeführten Definition der Temperatur über die Boltzmann-Konstante im Bereich bis 300 °C kleiner als 0,06 K. Da sind die Abweichungen, die über einen abweichenden Luftdruck bei der Messung entstehen deutlich größer.

Die historische Celsius-Skala verwendet als unteren Fixpunkt den Schmelzpunkt von Wasser (0 °C) und als oberen Fixpunkt den Siedepunkt von Wasser (100 °C). Und das bei Normaldruck. Das bedeutet bei einem offenen Gefäß, dass der umgebende Luftdruck bei 1,01325 bar bzw. 1013,25 Hektopascal liegen muss. Vorgehen:

- Den PiLogger mit angeschlossenem Temperatur-Sensor und Anzeigemöglichkeit etwa 20 Minuten betreiben, um einen eingeschwungenen thermischen Zustand zu erreichen.
- Währenddessen ein Wasserbad mit Eiswürfeln in einem offenem Gefäß vorbereiten. Der Sensor muss vollständig in das Wasser zwischen den Eiswürfeln

eintauchbar sein. Die Eiswürfel selber sind zu kalt und tiefer gelegenes Wasser ist zu warm. Wichtig ist auch hier ein thermisch eingeschwungener Zustand.

- Jetzt den Sensor vollständig in das Schmelzwasser einbringen und die Temperaturanzeige beobachten. Nach einiger Zeit sollte sich ein stabil bleibender Wert nahe 0 °C einstellen. Den tatsächlichen Wert notieren.
- Den notierten Wert negativ nehmen (mal -1) und als Offset auf der Seite Kalibration der WebMonitor Software eintragen (für den verwendeten Sensor-Typ). Direkt nach dem Speichern sollte nun die Anzeige unter 'Live Werte' tatsächlich 0 °C sein. Wenn das Eisbad noch ausreichend Eis enthält – also das Schmelzwasser noch als Referenz dienen kann, kann eine eventuell noch bestehende Abweichung noch weiter verringert werden.
- Jetzt den Sensor aus dem Eisbad nehmen und stattdessen ein Gefäß mit kochendem Wasser vorbereiten etwa einen *offenen* Wasserkocher.
- Den Sensor nun vollständig in das kochende Wasser eintauchen. Dabei darf der Sensor nicht in Kontakt mit dem Boden und dem Heizelement kommen !
 Achtung : Verbrühungsgefahr !

Den Sensor am besten mit einer *langen* Zange am Anschlusskabel halten ! Der Sensor sollte möglichst waagerecht, aber vollständig bedeckt in einer höheren Wasserschicht gehalten werden.

Auch hier jetzt die Temperaturanzeige beobachten und den nach einiger Zeit stabil bleibenden Wert nahe 100 °C notieren.

Mit dem notierten Anzeigewert den Korrekturfaktor berechnen:
 Faktor = 100 / Anzeigewert

Den Faktor für den verwendeten Sensor-Typ auf der Seite 'Kalibration' im WebMonitor eingeben und abspeichern. Nun sollte die Anzeige unter 'Live Werte' tatsächlich 100 °C zeigen. Falls die Abweichung noch nicht zufriedenstellend ist, noch einmal einen neuen Faktor ermitteln :

NeuerFaktor = 100 / Anzeigewert x AlterFaktor

Und den neuen Faktor eingeben und überprüfen.

Nach Durchführung dieser Prozedur sollte insbesondere mit einem Pt1000 Sensor eine Genauigkeit von +/- 0,1 °C erreicht werden.

Da sind die in der Praxis auftretenden Abweichungen beim Messen, beispielsweise durch Luftzug oder Sonneneinstrahlung, schnell 30 mal so groß.

12.2 Kalibrierung Impulszählung

Wie schon der Name sagt, handelt es sich hier um einen Zähler. Es geht um die Erfassung von Impulsen pro Zeiteinheit. Die Genauigkeit hängt also im Wesentlichen von der Genauigkeit des Zeitgebers ab. Das ist beim PiLogger ein quarzgesteuerter Taktgenerator. Der Quarz ist ein sogenannter Uhrenquarz mit 32,768 kHz und einer Genauigkeit von +/- 20 ppm (**p**arts **p**er **m**illion) – also +/- 0,002 % , entsprechend etwa +/- 0,7 Hz.

An dieser Stelle gibt es zwar keine Justiermöglichkeit, aber im Zusammenspiel mit dem verwendeten Sensor kann die Gesamtgenauigkeit korrigiert werden. In der WebMonitor Software gibt es auf der Seite 'Einstellungen 2' die Sensorkonstante (siehe 6.6) - den 'Faktor Pulse'.

Dies ist also ein linearer Faktor, der typisch für den verwendeten Sensor ist (bauartbedingt). Durch Anpassung dieses Faktors kann eine eventuelle Abweichung der Messwerte von den erwarteten Werten korrigiert werden. Voraussetzung dafür ist natürlich eine Versuchsanordnung mit bekannten, verlässlichen Erwartungswerten.

Beispielsweise für einen Durchflusssensor kann ein eingestellter Wasserdurchfluss durch Messen der Wassermenge in einer ebenfalls gemessenen Zeitspanne überprüft werden. Hier sind die zur Hilfe genommenen Messnormale ein Messgefäß bekannten Volumens (Eimer) und eine präzise Stoppuhr.

Oder für einen Windsensor eine bekannte Windgeschwindigkeit in einem Windkanal... Steht ein kalibriertes zweites Windmessgerät zur Verfügung und lassen sich beide Geräte der gleichen Windströmung ohne Verfälschung aussetzen, kann die Anzeige des PiLoggers mit dem 'Faktor Pulse' auf diese Referenz angepasst werden.

12.3 Kalibrierung Spannungsmessung

Die PiLogger WebMonitor Software bietet auch die Möglichkeit die Spannungsmesswerte mit Offset und Faktor zu korrigieren. Siehe hierzu Kapitel 6.7, insbesondere die Hinweise zu den möglichen Aufbauproblemen als Grund für eine Messwertabweichung. Der PiLogger misst die Spannung an den Klemmen 'Out+' und 'Out-'. Soll also die angezeigte Spannung mit einem Referenzmessgerät verglichen werden, zum Beispiel einem Multimeter mit hoher Genauigkeit (< 0,5 %), müssen die Messspitzen dieses Vergleichsgerätes an diesen PiLogger-Klemmen angesetzt werden. Die Messung an einem entfernteren Ort – etwa an den Batterieklemmen wird durch den Spannungsabfall über den Kabeln verfälscht. Je höher der aktuell durch die Kabel fließende Strom ist, desto größer ist der entstehende Spannungsunterschied zwischen den Messorten.

In manchen Fällen kann aber genau dieser systematische Messunterschied die Motivation für eine Messwertkorrektur sein:

Wenn das Ziel ist, den Akku-Zustand genau zu überwachen, ist die Akku-Spannung der wichtigere Wert für die Energiebilanz. Ist das Ziel die vom Generator (Wind-,Solar-) erzeugte Leistung zu überwachen, ist die Generatorspannung die interessantere. Wenn irgend möglich, sollte dies bereits bei der Wahl der Platzierung des PiLoggers beachtet werden und die Kabel sollten immer so kurz wie möglich und der Kabelquerschnitt so groß wie möglich sein.

Grundsätzlich sieht der Kalibriervorgang so aus:

• Nullpunktabgleich :

Die Kabel von den Klemmen 'In+', 'In-', 'Out+' und 'Out-' entfernen.

Die Klemmen 'Out+' und 'Out-' mit einer Kurzschlussbrücke miteinander verbinden.

Die Spannungsanzeige sollte nun stabil nahe Null sein, also etwa 0 … 10 mV. Ist dies nicht der Fall, gibt es möglicherweise ein Problem mit der Stromversorgung des Raspberry Pi (oder ESP32)

→ In der Regel sollte unter diesen Versuchsbedingungen kein Offset nötig sein.

• Linearabweichung:

An die Klemmen 'Out+' und 'Out-' eine stabile Spannung von möglichst 30 V (halbe Maximalspannung) aus einem Labornetzteil anschließen.

Die Spannung mit einem bekannt guten (kalibrierten) Messgerät an den PiLogger-Klemmen messen und wenn möglich fein nachregulieren.

Korrekturfaktor berechnen :

Faktor Spannung = U Referenz Anzeige / U PiLogger Anzeige

Falls kein Labornetzteil zur Verfügung steht, kann ersatzweise der Akku im Leerlauf (Sonst nichts angeschlossen) verwendet werden – damit liegt die Spannung zwar nicht beim halben Vollausschlag, dafür aber im interessantesten Bereich.

Bitte beachten : Das Ergebnis kann nur so gut sein wie das Referenzmessgerät. Das Referenzmessgerät sollte möglichst genau und möglichst kalibriert sein.

12.4 Kalibrierung Strommessung

Auch die Strommesswerte können in der PiLogger WebMonitor Software mittels Offset und Faktor korrigiert werden.

Der PiLogger arbeitet mit einem Hallsensor zur Strommessung. Das analoge Ausgangssignal muss beim Strom Null genau die halbe Ausgangsspannung haben und der Analog-Digital-Wandler muss bei dieser Spannung genau Null ausgeben. Beide Exemplarabweichungen müssen aufeinander abgeglichen werden. Dies kann durch Drift und externe Magnetfelder auch nach einiger Zeit wiederholt notwendig werden.

Dazu hat der PiLogger die Funktion 'Amp-Zero', die über die WebMonitor Software ausgeführt werden kann – siehe Kapitel 6.5 .

Bei der Ausführung darf kein Strom durch den Sensor fließen – also zum Beispiel nichts an die Klemme 'In+' angeschlossen sein. Durch Ausführen von 'Amp-Zero' wird der aktuelle Strommittelwert genommen und zu Null gesetzt.



Wenn der Abgleich erfolgreich ist, bewegt sich der Live-Wert für den Strom zwischen -10 mA und +10 mA. Das kann gut auf der Seite 'Live Kurve' begutachtet werden.

Dieser Nullpunkt-Abgleich ist immer vor einer eventuellen Kalibrierung mittels Offset und Faktor durchzuführen.

Der Ablauf der Strom-Kalibrierung sind dann so aus:

Offset Strom

Normalerweise sollte hier der Abgleich per 'Amp-Zero' ausreichen. In manchen Fällen soll aber ein (überwiegend) konstanter Strom – zum Beispiel für einen DC/DC-Wandler für die Versorgung des Raspi – heraus gerechnet werden. Dann muss hier dieser Strom negativ genommen (mal -1) eingetragen werden und abgespeichert werden. • Faktor Strom

Für die Kalibrierung der linearen Messwertabweichung muss zunächst ein konstanter Strom im mittleren Messbereich erzeugt werden.

- Dazu wird an die Klemmen 'In+' und 'In-' eine geeignete Konstantspannungsquelle (Labornetzteil oder Akku) angeschlossen – zum Beispiel mit 12 V.
- Für einen Referenzstrom von beispielsweise ungefähr 5 A wird nun ein Hochlastwiderstand mit 2,2 Ω benötigt.
 Achtung : Bei diesen Beispielwerten muss der Widerstand bereits 65,5 W aushalten können und wird dabei heiß ! Der Widerstand wird nun mit einem bekannt genauen Messgerät ausgemessen, das heißt der Widerstand in Ohm wird exakt ermittelt.
- 3. Der Widerstand wird nun als Last an die Klemmen 'Out+' und 'Out-' angeschlossen.
- 4. Jetzt wird mit einem bekannt genauen Messgerät bei eingeschalteter Spannungsquelle die tatsächlich anliegende Spannung direkt an den Anschlüssen des Lastwiderstandes gemessen.

Mit diesen 2 Messwerten können wir nun den tatsächlich fließenden Referenzstrom berechnen (Ohmsches Gesetz) :

I Referenz = U Widerstand / R gemessen

Anmerkung : Soll der Referenzstrom ein möglichst glatter Wert sein – also im obigen Beispiel genau 5 A, dann kann durch Änderung der Spannung (Labornetzteil) hier ein Abgleich erfolgen. Das ist aber nur zu Vergleichszwecken hilfreich.

5. Nun wird mit dem PiLogger WebMonitor der Live-Wert für den Strom beobachtet und abgelesen.

Den Korrekturfaktor Strom berechnen wir so :

Faktor Strom = I Referenz / I Anzeige

Diesen Faktor tragen wir auf der Seite 'Kalibration' ein, setzen den Wert für 'Volt/Amp Korrektur Aktivieren' auf 'Ja' und speichern alles ab.

6. Jetzt noch einmal den Live-Wert Strom überprüfen und gegebenenfalls die Prozedur noch einmal durchführen, um das Ergebnis zu verbessern.

13 Einbindung in Smart Home Systeme

Die Einbindung des PiLogger One in Smart Home Systeme ist grundsätzlich auf zwei Arten möglich :

Direkt – der PiLogger wird auf dem Raspberry Pi auf dem das Smart Home System läuft, als Erweiterung betrieben und ist somit ein I²C Multi-Sensor. Das setzt voraus, dass die PiLogger Verkabelung in der Nähe des Raspberry Pi möglich ist.

Über Netzwerk – der PiLogger wird auf einem kleineren, separaten Gastrechner mit dem WebMonitor betrieben und ist im lokalen Netzwerk erreichbar.

Ab der Version 0.16 des WebMonitor für Raspberry Pi oder 0.4 für ESP32 gibt es einen speziellen zusätzlichen Datenabfrage-Pfad /rawdata/, der die Daten in einem leicht übernehmbaren Format zurücksendet.

(Also gegebenenfalls die PiLogger WebMonitor Software upgraden !) Dies eine einfache Form einer REST Schnittstelle und wird von vielen Smart Home Systemen unterstützt.

Dieser Ansatz erfordert einen kleinen Rechner mehr, bietet dafür aber maximale Flexibilität.

Es gibt im Raspberry Pi Umfeld eine Reihe verschiedener Smart Home Lösungen. Hier einige Beispiele – ohne Anspruch auf Vollständigkeit:

- Home Assistant
- ioBroker
- FHEM
- openHAB
- Gladys Assistant
- nymea
- RaspberryMatic
- Homebridge

Nachfolgend zunächst eine Anleitung für eine Home Assistant Anbindung über Netzwerk. Weitere Anleitungen folgen sobald verfügbar ...

13.1 Home Assistant

Wenn der PiLogger One auf einem Raspberry Pi oder einem ESP32 mit dem WebMonitor betrieben wird, stellt der WebMonitor ein Web-Interface zur Verfügung, das mit einem beliebigen Gerät im eigenen Hausnetz mit dem Web-Browser aufgerufen werden kann.

Dabei ruft der Browser die Live-Daten ständig vom Raspi (oder ESP32) ab - das kann auch eine eventuell schon vorhandene 'Home Assistant' Instanz machen.



Damit können Automatisierungen in Home Assistant auch die Messdaten des PiLogger als Grundlage benutzen - zum Beispiel um bei zu starkem Wind die Markise einzufahren.

Um den PiLogger einzubinden muss in Home Assistant die Datei 'configuration.yaml' bearbeitet werden. Das kann im Web-Interface von HA mit dem 'File editor' gemacht werden. In der aktuellen Version 2024.3.3 ist dieses Add-on nicht standardmäßig an Bord und muss manuell nachinstalliert werden.

$\equiv \langle$	Home Assistant	\leftarrow	Informationen	Dokumentation	Konfiguration	Protokoll	
5	Übersicht	File editor					
4	Energie	Current version: 5.8.0 (Änderun	<u>gsprotokoll)</u>				
Ę	Karte	8 Bewertung 🚱 Kern 🗦	💲 Ingress 🖙 Signiert				
≣	Logbuch	Simple browser-based file edito Weitere Informationen findest d	r for Home Assistant. u auf der Seite <mark>File editor</mark>				
11.	Verlauf	J.C.					
2	File editor	Daim Deater states					
Ľ	Medien	Das Add-on beim Systemstart starter		-			
Ê	To-do-Listen	Watchdog Dadurch wird das Add-on gestartet, fa	ills es abstürzt				
		In der Seitenleiste anzeigen Add-on zur Seitenleiste hinzufügen		•			
		STARTEN					DEINSTALLIEREN
7	Entwicklerwerkzeuge						
۰	Einstellungen	Home Assistar	t Add-on: File	editor			
		Browser-based configuration f	ile editor for Home Assistan	t.			
.	Benachrichtigungen 🕕	aarch64 yes amd64 yes ar	mhf yes armv7 yes i386	yes			
MP	Max Pi	Home Assistant <	Configurato	r			

Nun muss im File 'configuration.yaml' am Ende dieser Textblock eingefügt werden:

```
rest:
   resource: http://192.168.178.20:8080/rawdata/
    scan_interval: 10
    sensor:
    - name: "PiLoO1 Temperatur"
      unique_id: "pilo_temp"
       value_template: '{{value_json.PiLoTemp1}}'
       device_class: "temperature"
      unit_of_measurement: "°C"
      name: "PiLo02 Wind"
       unique_id: "pilo_wind"
      value_template: '{{value_json.PiLoWind1}}'
device_class: "wind_speed"
      unit_of_measurement: "km/h"
     - name: "PiLo03 Spannung"
       unique_id: "pilo_volt"
      value_template: '{{value_json.PiLoVolt}}'
device_class: "voltage"
      unit_of_measurement: "V"
      name: "PiLo04 Strom"
       unique_id: "pilo_amps"
       value_template: '{{value_json.PiLoAmps}}'
       device_class: "current"
       unit_of_measurement: "A"
      name: "PiLo05 Leistung"
       unique_id: "pilo_watt"
       value_template: '{{value_json.PiLoWatt}}'
       device_class: "power"
       unit_of_measurement: "W"
      name: "PiLo06 TagesBilanz"
       unique_id: "pilo_dayenergy"
       value_template: '{{value_json.DayEnergy}}'
      device_class: "energy_storage"
unit_of_measurement: "Wh"
      name: "PiLo07 TagesErtrag"
       unique_id: "pilo_dayharvest"
       value_template: '{{value_json.DayHarvest}}'
       device_class: "energy"
       unit_of_measurement: "Wh"
      name: "PiLo08 TagesVerbrauch"
       unique_id: "pilo_dayconsumption"
      value_template: '{{value_json.DayConsumption}}'
device_class: "energy"
       unit_of_measurement: "Wh"
      name: "PiLo09 DauerBilanz"
       unique_id: "pilo_permenergy"
      value_template: '{{value_json.PermEnergy}}'
device_class: "energy_storage"
       unit_of_measurement: "Wh"
    - name: "PiLo10 DauerErtrag"
       unique_id: "pilo_permharvest"
      value_template: '{{value_json.PermHarvest}}'
device_class: "energy"_____
       unit_of_measurement: "Wh"
      name: "PiLo11 DauerVerbrauch"
      unique_id: "pilo_permconsumption"
value_template: '{{value_json.PermConsumption}}'
device_class: "energy"
       unit_of_measurement: "Wh"
     - name: "PiLo12 Dauer Zeit"
      unique_id: "pilo_permenertime"
value_template: '{{value_json.PermEnerTime}}'
       device_class: "duration"
       unit_of_measurement: "min"
```

Dieser Textblock kann hier heruntergeladen werden: <u>HA Konfiguration</u> Natürlich muss in der zweiten Zeile die tatsächliche IP-Adresse des eigenen PiLogger im Heimnetz eingetragen werden
<u>Hinweis</u> : Dazu gehört auch, dass im Router festgelegt wird, dass der PiLogger immer diese eine IP-Adresse fest zugewiesen bekommt.

Die Abfragerate wird mit 'scan_interval' in Sekunden angegeben. Sie sollte den eigenen Erfordernissen angepasst werden.

Ebenso lassen sich bereits hier sinnvolle eigene Namen für die verschiedenen Messwerte angeben - die lassen sich aber auch später noch in HA ändern.

Jetzt muss die Datei noch abgespeichert und geschlossen werden. Anschließend muss Home Assistant neu gestartet werden :

- Einstellungen
- rechts oben auf die 3 Punkte (Menü) klicken
- 'Home Assistant neu starten' wählen
- im Pop-Up-Fenster die Option 'Home Assistant neu starten' anklicken
- im Rückfrage-Fenster dann final bestätigen

... und warten : unten links wird der Fortschritt gemeldet, bis HA wieder vollständig geladen ist.

Jetzt auf die 'Übersicht' Seite gehen und - wenn alles geklappt hat - taucht ein neues Panel 'Sensor' mit den PiLogger Werten auf :

=	Home						Q	P
:: +		•	Max Pi	Unbekannt	6	Teilweise bewölkt		13 °C ۇ 59 %
₽		Sen	ISOF					
≣≣		8	PiLo01 Temperatur	13,3 °C				
		പ്പി	PiLo02 Wind	3,1 km/h				
٩		\sim	PiLo03 Spannung	5,852 V				
۵		~	PiLo04 Strom	0,925 A				
Ê		ş	PiLo05 Leistung	5,413 W				
		Ē	PiLo06 TagesBilanz	47,61 Wh				
		4	PiLo07 TagesErtrag	47,61 Wh				
		4	PiLo08 TagesVerbrauch	0,00 Wh				
		÷	PiLo09 DauerBilanz	296,92 Wh				
7		4	PiLo10 DauerErtrag	296,96 Wh				
\$		4	PiLo11 DauerVerbrauch	0,04 Wh				
		3	PiLo12 Dauer Zeit	169:20:00				
MP								

Damit steht dem Einbinden einzelner Werte in eigene Dashboards (Übersichtsseiten) nichts mehr im Weg.

Die jetzt dem System hinzugefügten Entitäten können nun auch für Automatisierungen verwendet werden.

14 Anhang

14.1 Die I²C Tools

Dieses Kapitel gilt für Raspberry Pi's auf denen Raspberry Pi OS installiert ist. Das ist ein vollständiges Linux Betriebssystem basierend auf der Debian Distribution.

Durch die Installation des SMbus Moduls werden automatisch auch die Module der 'i2c-tools' installiert worden, die über die Kommandozeile aufrufbar sind.

Die I²C Tools bestehen aus 4 Programmen: 'i2cdetect', 'i2cdump', 'i2cget' und 'i2cset'. Sie befinden sich im Ordner '/usr/sbin' :



Insbesondere das Tool 'i2cdetect' kann sehr nützlich sein um die Verbindung zu prüfen und die tatsächliche Adresse des PiLoggers zu verifizieren. Der Aufruf erfolgt im Konsolenfenster mit dieser Zeile:

i2cdetect -y 1

Die Option -y bewirkt, dass die normalerweise erscheinende Sicherheitsabfrage unterdrückt wird. Die '1' gibt den zu verwendenden Bus an (beim Raspberry Pi Typ A wäre das die '0'). Die Anzeige sieht dann etwa so aus:

<pre>pi@PiLogger:~ \$ i2cdetect -y 1</pre>																
	0	1	2	3	4	5	6	7	8	9	а	b	С	d	е	f
00:																
10:																
20:																
30:																
40:									48							
50:																
60:																
70:																
pi@PiLogger:~ \$																

Die offizielle Dokumentation findet sich hier: http://www.lm-sensors.org/wiki/man/i2cdetect

Das Tool 'i2cdump' ist grundsätzlich ein Block-Lese-Tool und speziell für das Auslesen älterer, kleiner EEPROMs gedacht – bitte nicht mit dem PiLogger verwenden.

Die Tools 'i2cget' und 'i2cset' sind zum Lesen und Schreiben einzelner Register eines I²C Bus Bausteins verwendbar. Dies sind die Grundbausteine für die I²C Bus Operationen, die so im Konsolenfenster verwendbar sind. Die jeweiligen Dokus finden sich hier:

http://www.lm-sensors.org/wiki/man/i2cdump http://www.lm-sensors.org/wiki/man/i2cget http://www.lm-sensors.org/wiki/man/i2cset

14.2 Konzept PiLogger WebMonitor

In diesem Kapitel beschreiben wir die Arbeitsweise und den Ansatz der WebMonitor Software für den *Raspberry Pi* – die Unterschiede für den ESP32 Port werden an anderer Stelle beschrieben.

Das soll dem Verständnis dienen und auch als Anregung für eigene Anpassungen oder Projekte – PiLogger WebMonitor ist freie offene Software (FOS).

Der PiLogger WebMonitor ist eine Netzwerk-Anwendung (WebApplication) die die Fähigkeiten eines Raspberry Pi sowohl eine Netzwerk-Verbindung als auch den Zugriff auf die GPIOs (General Purpose Input/Output; Universeller Eingangs- und Ausgangs-Anschluss) bereitzustellen nutzt.

Kernstück dieser Webanwendung ist dabei das Python Web Framework 'Bottle' von Marcel Hellkamp (<u>https://bottlepy.org/</u>).

"Bottle ist ein schnelles, einfaches und leichtgewichtiges WSGI (Web Server Gateway Interface) micro web-framework für Python. Es wird als ein Eine-Datei-Modul verteilt und hat keine weiteren Abhängigkeiten als die zur Python Standard Bibliothek."

Es wird unter MIT Lizenz angeboten und kann aus den Raspberry Pi OS-Software-Ablagen (Repositories) installiert werden. Der eingebaute Server für Entwicklungszwecke reicht völlig für eine private (nicht Internet-verbundene) WebSeite. Bottle lässt sich mit Erweiterungsmodulen (Plugins) ausbauen – was bezüglich Zugriffssteuerung mit beispielsweise 'Bottle-Cork' noch folgen soll. Statt des eingebauten Servers lassen sich auch alternativ viel andere Server nutzen – siehe Bottle-Dokumentation.

Ausgehend von diesem Grundbaustein liegt es nahe auch den Rest direkt in Python zu lösen. Das Hauptprogramm auf dem Raspberry ist demzufolge ein Python-Programm (üblicherweise Script genannt – obwohl Python eine ausgewachsene Hochsprache ist).

Die Programmdatei heißt 'PiLogger-bottle.py' und ist als Teil des Archivs 'PiLo-WebMon.zip' herunterladbar :

https://www.pilogger.de/index.php/de/download-de/send/2-software/8-pilowebmon

Voraussetzung ist ein aktiviertes I²C Interface mit aktiviertem alternativen I²C Treiber, sowie die Installation von 'python3-smbus', dem Modul für Python um auf den I²C Bus zuzugreifen. Außerdem das Installieren von 'python3-gpiozero', einer Python-Erweiterung für den direkten Zugriff auf die GPIOs. Damit kann die Interrupt-Leitung des PiLogger One auch als solche ausgewertet werden. Das ist die Grundausstattung für alle I²C Sensoren, die mit dem Raspberry verwendet werden sollen. Der PiLogger One ist sozusagen ein Multi-Sensor. Die Anwendungs-Programmier-Schnittstelle (API) ist im Kapitel 9 beschrieben.

Das Python-Script erledigt also nun die Hauptaufgabe – das Loggen, indem es die Messwerte vom PiLogger One ausliest, nachbearbeitet und auf der SD-Karte des Raspberry Pi speichert. Dabei wird keine große Datenbank benutzt, sondern die Daten werden in einer reinen Textdatei im Anhängen-Modus gespeichert. Das ist eine klassische CSV-Datei (comma separated values), die Zeile für Zeile die Messwerte aufzeichnet.

Das Python-Script bietet nun zusätzlich über das Modul 'bottle' die Möglichkeit Webseiten an einen Browser auszugeben. Was diese HTML Dateien ermöglichen ist im Kapitel 6 ausführlich beschrieben worden.

Damit wird eine plattformübergreifende Benutzeroberfläche mit allen modernen Darstellungsmöglichkeiten bereitgestellt.

Um den Datenverkehr zu minimieren und insbesondere keine externen Internet-Server ansprechen zu müssen, werden keine Dritt-Webschriften (Fonts) und keine riesigen Universal-Frameworks verwendet. Tatsächlich wird für alle Seiten eine gemeinsame CSS-Datei verwendet um den Darstellungsstil festzulegen. Darin enthalten ist auch die Verwendung einer einzigen Bilddatei, die alle benötigten Icons enthält, als sogenannte Sprites. Ausnahme ist hier die zusätzliche Datei für das sogenannte favicon – also das kleine Icon zur Seitenerkennung. So ist der Aufruf einer Seite (außer Diagramme) mit 4 kleinen Dateien beantwortet (1 document, 1 stylesheet, 1 img iconset & 1 img favicon). Diese Ressourcen sind statisch und werden in der Regel vom Browser zwischengespeichert (gecashed). Das heißt, der nächste Seitenaufruf geht nochmal schneller.

Die eigentlich anzuzeigenden Daten werden mit XHR dynamisch vom Raspberry geladen. Diese Technik nennt sich Ajax und XHR steht für 'XMLHttpRequest'. Kern ist hierbei, dass im Browser ein JavaScript-Script ausgeführt wird, das in unserem Fall direkt in die HTML-Seite eingebettet ist, um nicht noch eine weitere Datei anfordern zu müssen.

Dieses JavaScript-Script führt die erwähnten Requests (Anforderungen) für die anzuzeigenden Daten aus, die dann als kleine JSON-Datei vom Raspberry gesendet werden.

In der Gegenrichtung werden die Daten in Richtung Server mit der übertragenen Navigationsadresse als sogenannte Query-Parameter gesendet (url-data). Auch dies ist ein XHR – nur das hier die Antwort des Servers lediglich ok oder nok ist – also eine reine Erfolgsrückmeldung (Acknowledge).

Ein weiterer wichtiger Baustein der PiLogger WebMonitor Software ist die JavaScript-Bibliothek 'dygraphs' von Dan Vanderkam (<u>http://dygraphs.com/</u>). Diese sogenannte Charting-Library (Diagramm-Bibliothek) wird auf der Seite 'Diagramme' zum Erzeugen der Messwert-Zeitdiagramme verwendet. Dafür werden auf dieser Seite 2 zusätzliche Dateien vom Server angefordert : 'dygraph_p.min.js' - die Charting-Library in minimierter Form und 'dygraph_p.css' mit den Diagramm-spezifischen Stilvorgaben.

Die Datei 'dygraph_p.min.js' basiert dabei auf der Original-Version 2.1.0 und beinhaltet Patches (Reparaturen) für die Touch-Bedienung.

Dieses JavaScript-Modul ruft beim Aufruf der Seite 'Diagramme' die Logdatendatei 'showdata.csv' vom Raspberry ab, zerlegt und analysiert (parsed) die Daten und erzeugt ein Zeitdiagramm der Messwertreihen als Grafik auf einer zugewiesen Fläche der Seite.

Diese Charting-Library kommt dabei auch mit sehr umfangreichen Messreihen in erstaunlich kurzer Zeit zurecht. Die längste Zeit nimmt in der Regel die Übertragung der Datendatei in Anspruch. Das hängt in erster Linie von der Verbindungsqualität und der Größe der Datei ab. Dabei ist zu bedenken, dass diese Datendatei bis zu doppelt so groß werden kann, wie die eingestellte Datendateigröße, da sie aus der bisherigen Logdatei plus der aktuellen Datendatei gebildet wird – um immer einen sinnvollen Betrachtungszeitraum darstellen zu können.

Bisher sind alle Daten und die Kommunikation rein nur zwischen dem Raspberry Pi als Server und einem im lokalen Netz angemeldeten Benutzer-Rechner als Client über den eigenen Router gelaufen. Damit der Raspberry beim Loggen einen korrekten Zeitstempel zu den Messdaten hinzuzufügen kann, muss er seine interne Zeit mit einem Zeitserver synchronisieren. Das geschieht standardmäßig mit einem Internet-Zeitserver per NTP (Network Time Protocol). Mit einer einfachen Maßnahme lässt sich auch dieser Internet-Zugriff vermeiden:

Der Router synchronisiert selber seine interne Zeit mit einem Internet-Zeitserver, meistens mit einem Zeitserver des Netzanbieters. Die meisten Router bieten die Möglichkeit diese Zeit für das lokale interne Netz als Zeitserver zur Verfügung zu stellen – siehe Kapitel 4.3.1.

Wenn diese Voraussetzung erfüllt ist, kann der Raspberry sich nun lokal mit dem Router synchronisieren.

Nun bleiben nur noch die regelmäßigen (täglichen) Anfragen bei den Raspberry Pi OS-Update-Servern. Ist der Raspberry ausschließlich als Logger vorgesehen und strikt hinter der Firewall des Routers, so ist ein Update in der Regel nicht sicherheitsrelevant. Zumal diese Update-Anfragen ohne ein eingerichtetes Auto-Update sowieso nur die Datenbank aktualisieren – und andernfalls ein unbeaufsichtigtes Update den Logger-Betrieb unterbrechen kann. Diese Update-Anfragen lassen sich wahlweise abschalten – siehe Kapitel 4.3.2.

Zur Zeit ist der Zugang zu allen Funktionen des PiLogger WebMonitor allen Benutzern des heimischen Netzwerk offen. Dies lässt sich mit einem Erweiterungsmodul für 'Bottle' wie etwa 'Bottle-Cork' ändern – todo...

Wer den PiLogger WebMonitor auch weit entfernt über das Internet benutzen möchte, sollte dies über ein VPN (virtuell privates Netzwerk) tun – also über eine gesicherte Punkt-zu-Punkt-Verbindung.

PiLogger WebMonitor zeigt somit auch, dass 'Smart Home' und 'Internet der Dinge' auch ohne Cloud und Fremdserver möglich sind.