



Das PiLogger One Handbuch



Inhaltsverzeichnis

1 Einleitung.....	4
2 Sicherheitshinweise.....	5
3 Vorbereitung Raspberry Pi.....	6
3.1 Auswahl Betriebssystem.....	6
3.2 Download Raspbian.....	6
3.2.1 Raspbian oder Raspbian Lite.....	6
3.2.2 Archiv-Datei entpacken.....	8
3.3 Vorbereitung (micro) SD Karte.....	9
3.3.1 Welche SD Karte ?.....	9
3.3.2 Image auf SD Karte schreiben.....	9
3.4 Raspberry Pi einrichten.....	11
3.4.1 Micro SD Card in Raspberry Pi einsetzen.....	11
3.4.2 Netzteil, Monitor, Tastatur und Weiteres anschließen.....	11
3.4.3 Erstes Booten, Konfiguration.....	13
3.4.3.1 System.....	15
3.4.3.2 Schnittstellen.....	18
3.4.3.3 Leistung.....	19
3.4.3.4 Länderanpassung.....	20
3.4.4 Internet-Verbindung herstellen.....	26
3.4.5 Allgemeines Update.....	28
4 PiLogger Basis-Software einrichten.....	32
4.1 Alternativen I ² C Treiber aktivieren.....	32
4.2 Einrichten der PiLogger Basis-Software.....	35
4.2.1 Neuen Ordner erstellen.....	35
4.2.2 Archiv herunterladen.....	37
4.2.3 Archiv entpacken.....	39
4.2.4 Desktop-Starter einrichten.....	42
4.2.5 Probelauf (Ohne Hardware).....	43
4.2.6 Schnellstarter für Python.....	44
5 Montage und Inbetriebnahme.....	47
5.1 Vorbereitung.....	47
5.2 Modul aufsetzen.....	50
5.3 Erstes Einschalten.....	53
6 Bedienungsanleitung Basis-Software.....	54
6.1 Monitor-Fenster.....	54
6.2 Konfigurationsfenster.....	55
6.2.1 PiLogger Einstellungen.....	55
6.2.2 RasPi Einstellungen.....	58
7 Anschlussklemmen.....	61
7.1 Gleichspannungs- und Strommessung.....	61
7.2 Impulszähleingang.....	62
7.3 Temperaturfühlereingang.....	63
8 Anschlussbelegung GPIO-Leiste.....	64

9 Anwendungs-Programmierschnittstelle.....	65
9.1 Konfigurationsregister.....	65
9.2 Meßwertregister.....	66
9.3 Wertebereiche und Skalierung.....	68
10 Temperaturfühler-Tabellen.....	69
11 Technische Daten.....	72
12 Kalibrierung.....	72
13 Anhang.....	73
13.1 Die I ² C Tools.....	73
13.2 SD Karte schneller machen.....	74

Copyright © 2017,2018 , G. Weiß-Engel

PiLogger ® and the PiLogger Logo are registered trademarks of G. Weiß-Engel.

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

Linux ® 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.

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. Genauerer 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 Pi. Das ermöglicht auch einen geringen Stromverbrauch, weil der Pi gerade bei längeren Messintervallen zwischen den Speicherpunkten quasi nichts tun muss.

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 WLAN zu vernetzen, eröffnen sich tolle Möglichkeiten – zum Beispiel ein kleiner Server mit grafischen Auswertungen.

Die Basis-Software zum PiLogger soll hier nur einen Einstieg geben. Sie dient zur Inbetriebnahme, zur Konfiguration und als Test. Sie ist freie Beispiel-Software und eignet sich somit zum Lernen und als Startpunkt für eigene Projekte.

Dieses Handbuch versucht eine Schritt für Schritt Anleitung auch für Einsteiger zu bieten. Fortgeschrittene Raspberry Pi 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.

Es gibt inzwischen viele Varianten des Raspberry Pi, aber dadurch, dass der PiLogger One eine verkürzte Form des 'Hat Board' verwendet, also *kein* 'Hat Board' ist, kann er auch auf der ersten Generation des Raspberry Pi mit 26-poligen Stiftleisten verwendet werden. Eine gute Chance für einen vielleicht schon ausgemusterten Raspberry Pi 1 !

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 Vorbereitung Raspberry Pi

3.1 Auswahl Betriebssystem

Die kostenlos bereitgestellte Beispiel-Software für den PiLogger ist in 'Python 3' mit der grafischen Erweiterung 'tkinter' geschrieben.

Für die unkomplizierte Verwendung von Python, für die der Raspberry Pi ja ursprünglich konzipiert wurde, bietet es sich an das offiziell empfohlene Betriebssystem Raspbian zu wählen. Es ist ein Debian Linux mit wirklich exzellenter Anpassung auf die Hardware und breiter Unterstützung, und – abgesehen von einigen Spezialkomponenten (Broadcom) und Java (Oracle) ist alles quelloffen und frei verfügbar !

Wer bereits Erfahrung mit der Einrichtung von Raspian auf dem Raspberry Pi hat, kann hier weiterspringen zum Kapitel 3 'PiLogger Basis-Software installieren'.

3.2 Download Raspbian

3.2.1 Raspbian oder Raspbian Lite

Auf der Download-Seite von www.raspberrypi.org sind unter dem Link zu Raspbian zwei verschiedene Versionen aufgeführt: 'Full Desktop Image' und 'Lite : Minimal Image'.

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

Da hier die grafische Oberfläche benutzt werden soll, ist die Wahl eindeutig Raspbian in der Vollversion zu verwenden !

RASPBIAN

Raspbian is the Foundation's official supported operating system. You can install it with [NOOBS](#) or download the image below and follow our [installation guide](#).

Raspbian comes pre-installed with plenty of software for education, programming and general use. It has Python, Scratch, Sonic Pi, Java, Mathematica and more.

The Raspbian with Desktop image contained in the ZIP archive is over 4GB in size, which means that these archives use features which are not supported by older unzip tools on some platforms. If you find that the download appears to be corrupt or the file is not unzipping correctly, please try using [7Zip](#) (Windows) or [The Unarchiver](#) (Macintosh). Both are free of charge and have been tested to unzip the image correctly.

RASPBIAN STRETCH WITH DESKTOP

Image with desktop based on Debian Stretch

Version: August 2017
Release date: 2017-08-16
Kernel version: 4.9
Release notes: [Link](#)

[Download Torrent](#) [Download ZIP](#)

SHA-256:
309f355ad5ca3e15d4866dfa16f17e4a5412632fec00976fe270d59516668849

RASPBIAN STRETCH LITE

Minimal image based on Debian Stretch

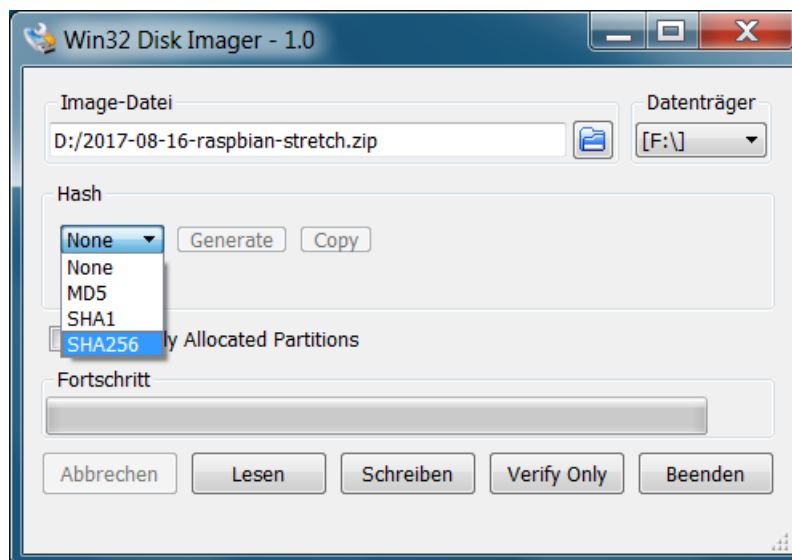
Version: August 2017
Release date: 2017-08-16
Kernel version: 4.9
Release notes: [Link](#)

[Download Torrent](#) [Download ZIP](#)

SHA-256:
52e68130c152895905abe66279dd9feaa68091ba55619f5b900f2ebcd381427b

Selbstverständlich lässt sich der PiLogger auch in einem sogenannten 'headless' System verwenden - also ohne die grafische Oberfläche - dies ist aber nicht das Ziel dieser Anleitung.

Wenn der Download mit einem Windows-PC erfolgt, lässt sich optional die SHA-256 Prüfsumme, die auf der Download-Seite angegeben ist, beispielsweise auch mit dem Tool 'Win32 Disk Imager' berechnen (<https://sourceforge.net/projects/win32diskimager/>). Dieses Tool werden wir später auch für das Schreiben auf die SD Karte verwenden. Nach dem Starten des Programms muss zunächst die Image-Datei in dem entsprechenden Eingabefeld angegeben werden. Dies kann auch durch Browsen erfolgen, indem man das kleine Ordner-Icon rechts neben dem Eingabefeld anklickt. Da die angegebene Prüfsumme für das .zip-Archiv gilt, muss man nun zunächst mit *.* alle Dateien zulassen, um die heruntergeladene Datei zu finden und auszuwählen. Anschließend wird in dem Bereich 'Hash' darunter mit der Ausklappliste SHA256 ausgewählt. Jetzt wird mit einem Klick auf den Knopf 'Generate' die Prüfsummenberechnung gestartet (dauert etwas bei 1,6 GB):



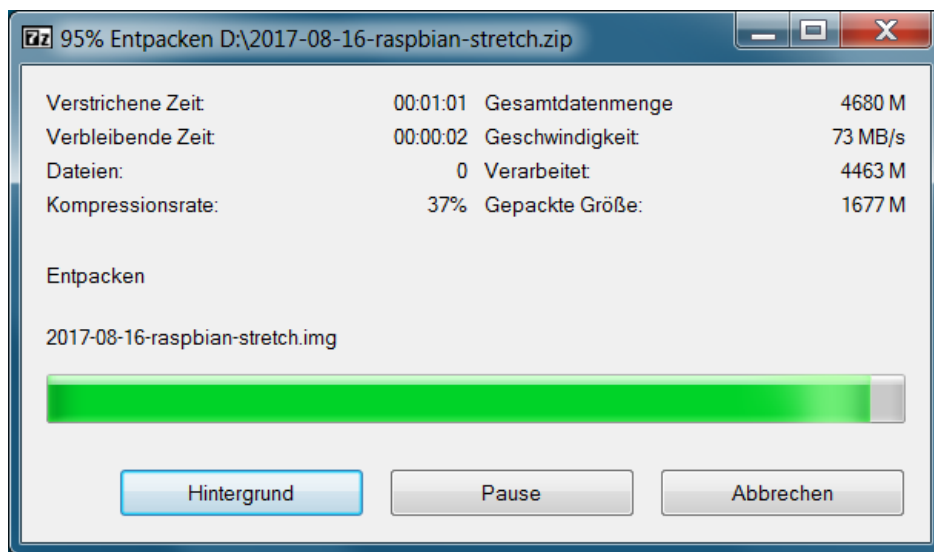
Damit lässt sich überprüfen, ob die Datei unbeschädigt und nicht manipuliert vorliegt – zum Beispiel, wenn man sie gar nicht selbst heruntergeladen hat, sondern von einem Freund auf USB-Stick erhalten hat.

3.2.2 Archiv-Datei entpacken

Die heruntergeladene Datei ist ein .zip-Archiv, das die eigentliche Image-Datei in komprimierter Form enthält. In diesem Zustand kann sie nicht auf die SD Karte geschrieben werden. Sie muss zunächst dekomprimiert werden – entpackt werden.

Bei den jüngsten Raspbian-Versionen gibt es teilweise Probleme mit den Bordmitteln von Windows. Wie auch auf der Download-Seite bei raspberrypi.org angegeben, sollte hier die Software 7zip zum Entpacken verwendet werden. Sie lässt sich unter <https://sourceforge.net/projects/sevenzip/> herunterladen.

Das Programm verhält sich wie ein normaler Datei-Manager. Nach dem Auswählen der im Archiv enthaltenen Image-Datei kann diese über das Kontext-Menü (Rechts-Klick) oder per Drag and Drop entpackt werden:



Dabei ist es wichtig sich zu merken, wo die Image-Datei gespeichert ist !

3.3 Vorbereitung (micro) SD Karte

3.3.1 Welche SD Karte ?

Für die SD oder MicroSD Karte (nur Generation 1 des Raspberry Pi hat einen großen SD-Kartenhalter) empfiehlt es sich immer eine möglichst schnelle Version zu verwenden, da der Raspberry Pi diese wie eine Festplatte benutzt. Die Größe sollte zwischen 8 GByte und 32 GByte liegen, was bedeutet, dass die Karte vom Typ SDHC (für high capacity) ist. Eine Karte mit 16 GByte ist für eine schnellere Version mit 40 MByte/s oder mehr wohl inzwischen schon die kleinste marktübliche Größe und damit die preiswerteste Empfehlung.

Mit den Standard-Einstellungen erreicht der Raspberry Pi am SD-Karten-Slot bis zu ca. 21 MByte/s. Dies lässt sich bei Verwendung von Karten mit UHS-1 / class 10 Geschwindigkeit mit einer Overclock-Einstellung (siehe Anhang) auf bis ca. 42 Mbyte/s steigern – aber natürlich immer begrenzt durch die tatsächliche Performance der SD Karte.



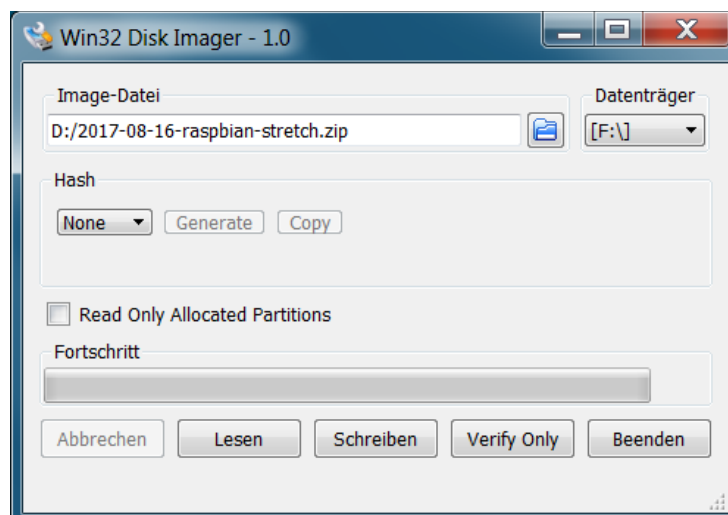
Beim Raspberry Pi 3 verhindert diese Overclock-Einstellung das Erkennen des integrierten WLAN/Bluetooth-Moduls.

Die bei der Vermarktung groß geschriebenen Geschwindigkeitsangaben gelten immer nur für den dauernden Lesebetrieb (streaming). Für das allgemeine Verhalten im Desktop-Betrieb sollte die Karte für das Lesen und Schreiben von 4K-Blöcken optimiert sein (wird manchmal beworben) – hier sind Geschwindigkeiten von 3 Mbyte/s schon Spitzenwerte.

3.3.2 Image auf SD Karte schreiben

Falls ein Windows-PC zu Verfügung steht, ist das Programm 'Win32DiskImager' zu empfehlen, da es das Schreiben des Image im korrekten Format auf die SD Karte sehr einfach und komfortabel ermöglicht.

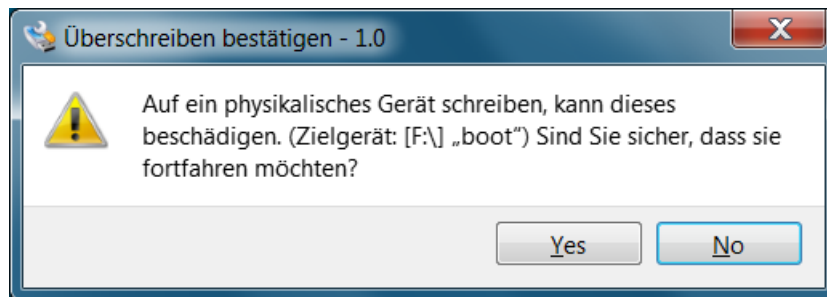
Nach dem Starten des Programms ist die Image-Datei im Eingabefeld 'Image File' anzugeben. Die Datei kann auch mit einem Klick auf der Ordner-Symbol rechts neben dem Eingabefeld durch Browsen ausgewählt werden.



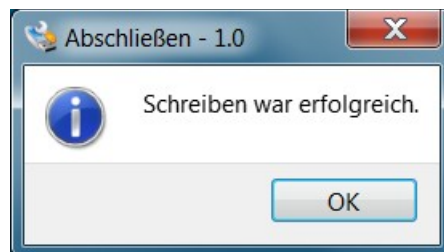
Rechts daneben muss im Bereich 'Device' in der Aufklappliste der Laufwerksbuchstabe für den Kartenleser gewählt werden – *Achtung : nicht mit anderen entfernbaren Speichern verwechseln !*

Der Kopiervorgang wird jetzt mit einem Klick auf den Knopf 'Schreiben' gestartet.

Daraufhin öffnet sich ein Bestätigungsfenster mit der Warnung, dass durch diesen Vorgang das Gerät beschädigt werden kann – das klingt dramatischer als es vermutlich gemeint war. Der Titel des Fensters gibt den Hinweis darauf, dass damit eventuell vorhandene Daten unwiederbringlich gelöscht werden. Außerdem wird tatsächlich in unserem Fall ein Dateisystem angelegt, welches Windows nicht lesen kann – keine Panik, dadurch wird die Karte nicht wirklich unbrauchbar.



Drücken Sie also jetzt auf 'Yes' und der Kopiervorgang startet wirklich. Er braucht eine Weile. Am Ende erscheint ein Benachrichtigungsfenster mit hoffentlich dieser Meldung:



Drücken Sie nun 'OK'. Das Meldungsfenster verschwindet und das Hauptfenster ist wieder sichtbar, in dem Sie nun durch Drücken auf 'Beenden' das Programm beenden.

Vergessen Sie nun nicht das Laufwerk sicher zu entfernen, bevor Sie die Micro-SD-Karte entnehmen.



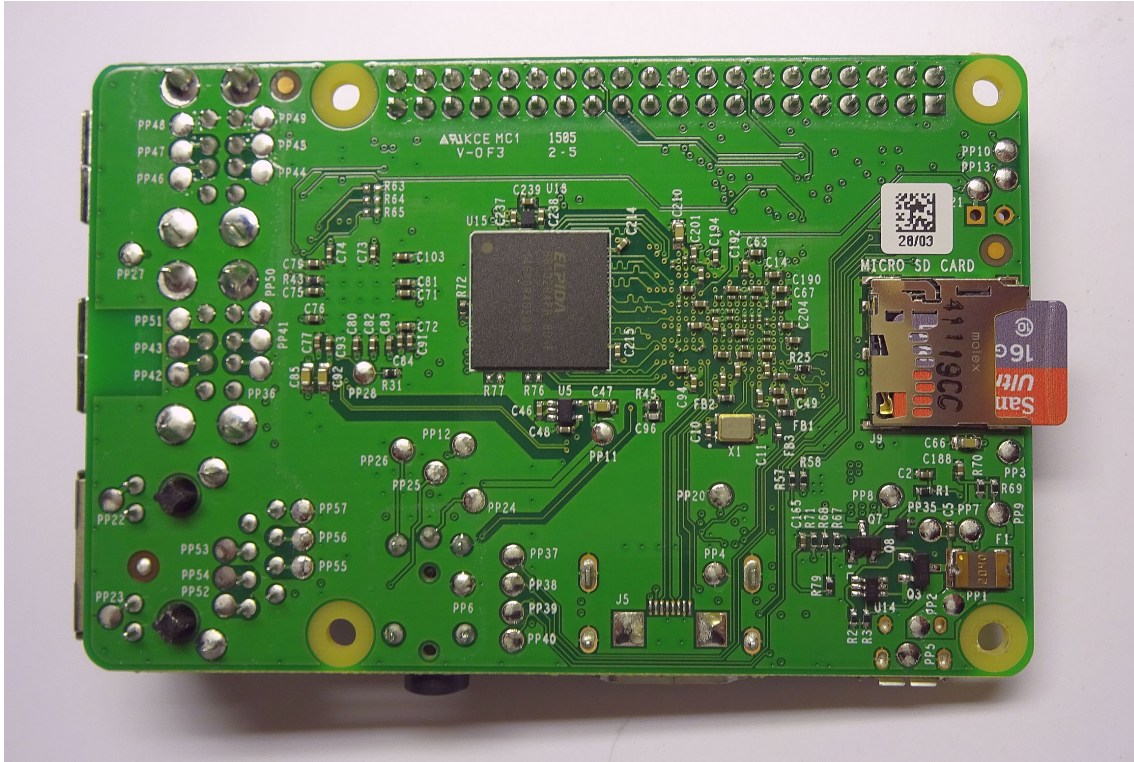
Entfernbarer Flash-Speicher haben die unangenehme Eigenschaft eigenständig interne Operationen auszuführen, von denen der Benutzer nichts mitbekommt. Wenn sie dabei durch plötzliches Abschalten der Versorgungsspannung – also simples Herausziehen – gestört werden, kann es auch heute noch vorkommen, dass der Speicher unbrauchbar wird.

3.4 Raspberry Pi einrichten

3.4.1 Micro SD Card in Raspberry Pi einsetzen

Die neu vorbereitete SD Card kann nun in den Slot (Halter) eingesetzt werden – dabei darf der Raspberry Pi nicht eingeschaltet sein!

Am besten sollte noch gar nichts angeschlossen sein.



3.4.2 Netzteil, Monitor, Tastatur und Weiteres anschließen

Nun braucht der Pi eine Basisausstattung für ein erstes Booten (Aufstarten):

- **Netzteil : ohne Strom läuft nichts**

Es wird ein USB-Netzteil mit einem Micro-USB-Anschlusskabel benötigt.

Alle USB-Netzteile haben 5 V Ausgangsspannung, aber ihre Fähigkeit Strom liefern zu können unterscheidet sich. Die schwächsten, und meist billigsten, müssen mindestens 500 mA liefern können (USB Spezifikation). Das wird aber je nach Raspberry Pi Modell und dem anzuschließenden USB-Zubehör nicht reichen. 1000 mA oder 1 A sollten es schon sein, für den neuen Raspberry Pi 3 werden gar 2,5 A empfohlen. Das Netzteil bitte während dieser Vorbereitung noch nicht einschalten (in die Steckdose stecken).

- **Ausgabe : eine grafische Anzeige braucht der Mensch**

Die einfachste und komfortabelste Art einen Bildschirm anzuschließen ist den HDMI-Stecker am Raspberry Pi zu benutzen. Falls kein moderner PC-Monitor mit HDMI-Eingang verfügbar ist und der Fernseher im Wohnzimmer gerade anderweitig benötigt wird, tut es auch ein ausgemusterter Monitor mit DVI-Eingang. Dafür gibt es günstige

Adapterkabel von HDMI auf DVI. Auch die Auflösung und das Bildformat sind in weiten Bereichen nicht ausschlaggebend – werden unterstützt und funktionieren deshalb.

Grundsätzlich ließe sich der Raspberry Pi auch fernbedient (remote controlled) über eine Netzwerkverbindung betreiben und auch einrichten – das ist aber nicht ganz so einfach und intuitiv, weshalb wir hier nicht näher darauf eingehen.

Hat der Monitor eingebaute Lautsprecher und einen HDMI-Eingang, kann der Pi darüber auch Audio ausgeben. Alternativ kann an der 3,5 mm Klinkenbuchse ein analoges Audio-Gerät angeschlossen werden, also zum Beispiel ein Kopfhörer oder ein aktives Stereo-Lautsprecher-Set.

- **Eingabe : der Pi braucht Input**

Als Eingabe reicht eine simple USB-Tastatur. Da auch eine Maus sehr zu empfehlen ist, die dann auch über USB angeschlossen werden muss, wäre eine drahtlose Tastatur- und Maus-Kombination die beste Lösung, die nur einen USB-Anschluss belegt, dabei etwas weniger Strom benötigt als zwei getrennte Geräte und ein komfortables Arbeiten ermöglicht.

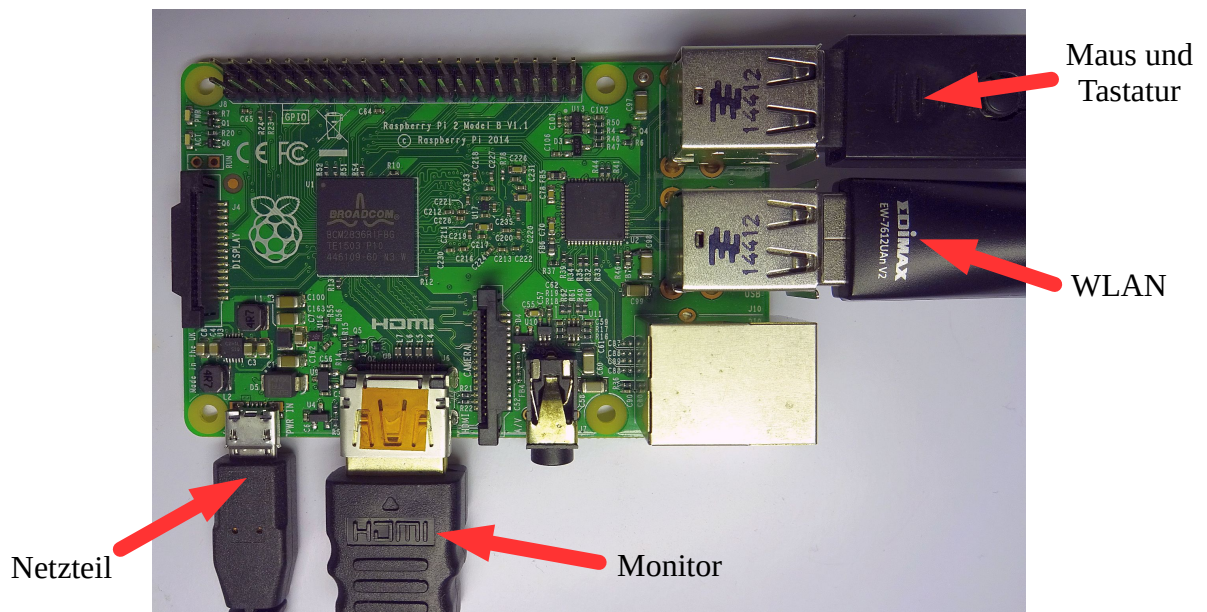
- **Netzwerk : Verbindungen sind hilfreich**

Mit einer direkten Verbindung zum Internet lassen sich nicht nur die am Anfang notwendigen Updates schnell und einfach erledigen. Raspbian stellt alle notwendigen Tools (*Werkzeuge, Hilfsmittel*) zur Verfügung, um zu Surfen und zu Googlen. Das ist in erster Linie ein Browser, in älteren Versionen war das Epiphany, ein schlanker Webbrowser für den Gnome Desktop – in den neusten Versionen wird Chromium, die quelloffene Variante des Google Chrome Browsers, installiert.

Je nach den gegebenen Möglichkeiten kann ein Netzkabel (zum Router) direkt angeschlossen werden (*nicht beim Modell A+*) oder ein USB-WLAN-Stick verwendet werden. Beim zu erwartenden Einsatzbereich des PiLogger ist vermutlich eine drahtlose Verbindung praktischer.

Der jüngste Spross der Pi-Familie, der Raspberry Pi 3 Model B, hat bereits ein WLAN-Modul an Bord.

Der fertig angeschlossene Raspberry Pi sieht dann etwa beim Modell 2 B so aus:



3.4.3 Erstes Booten, Konfiguration

Nun ist es Zeit den Strom einzuschalten und dem Raspberry Pi beim ersten Hochfahren (Booten) zuzusehen.

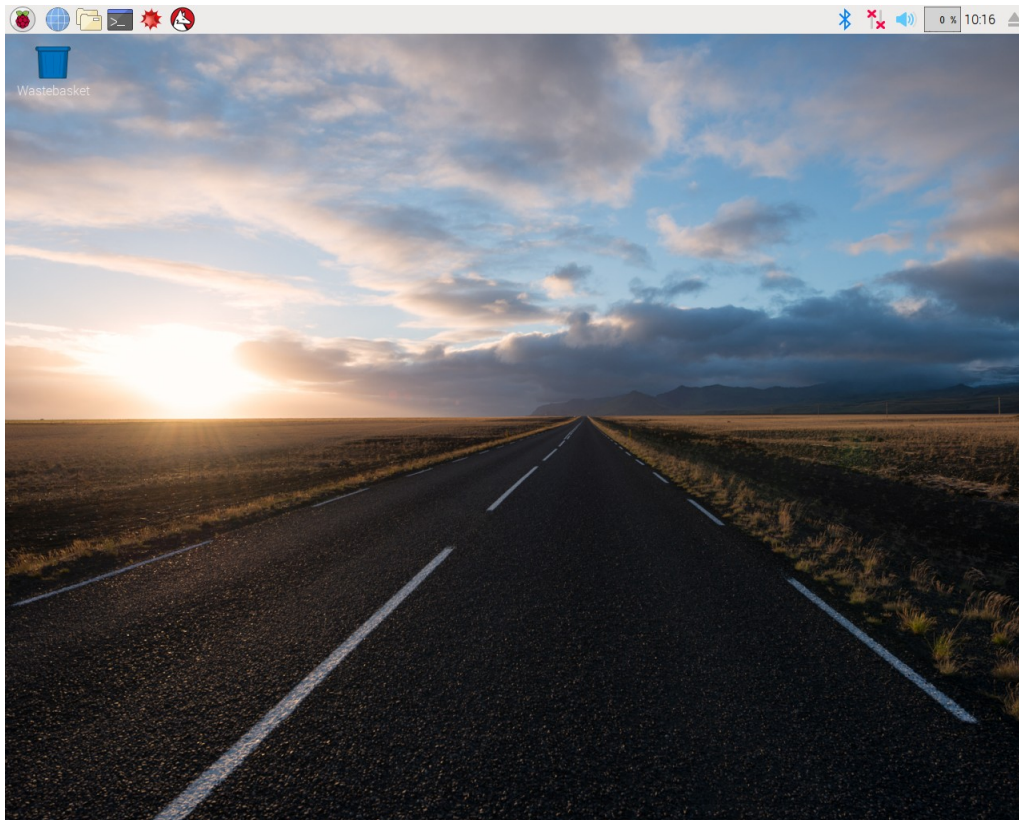
Auf dem Monitor sollten nun Himbeeren auftauchen – jede Himbeere steht für einen Prozessor-Kern. Beim Raspberry Pi 3 also 4 Stück.

Danach kommt eine kurze Meldung, dass das Dateisystem (der SD Karte) erweitert wurde und ein Neustart erfolgt – alles prima.

Beim nächsten Start taucht dann das erste Mal dieser Begrüßungsschirm auf:



Dann dauert es noch eine kleine Weile und der Desktop (Arbeitsplatz) erscheint:



Die Startleiste ist standardmäßig oben orientiert und das System ist auf Englisch gesetzt. Nun müssen zunächst ein paar grundlegende Dinge eingerichtet werden, wie zum Beispiel die Systemsprache. Dazu gibt es ein Hilfsprogramm namens 'raspi-config'. Seit Raspbian-Jessie gibt es eine grafische Version davon, die wie ein normales Anwenderprogramm zu starten ist. Gehen wir dazu auf 'Menu' > 'Preferences' > 'Raspberry Pi Configuration' :



Damit lassen sich alle notwendigen Einstellungsänderungen sehr komfortabel durchführen. Das Programm startet mit dieser Anzeige:



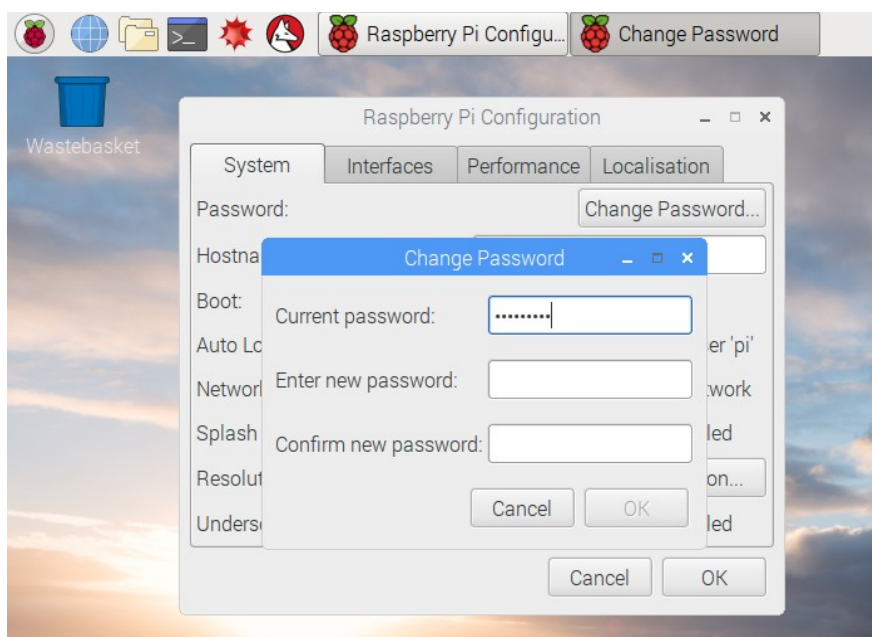
Hier nun Schritt für Schritt alle sinnvollen Einstellungen für den Einsatz des PiLoggers:

3.4.3.1 System

In der ersten Abteilung (Reiter, *Tab*) 'System' gibt es diese Möglichkeiten:

- **Change Password**

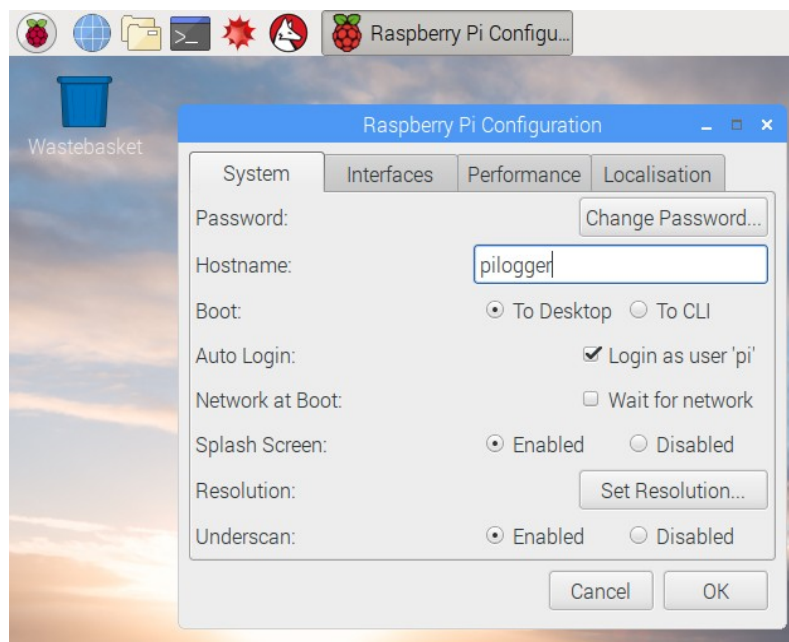
Hier gibt es die Möglichkeit das Passwort für die Systemanmeldung für den standardmäßig eingerichteten Benutzer 'pi' zu ändern. Dieses ist ebenso standardmäßig auf 'raspberry' gesetzt. Da alle Pi's dieses Passwort als Startwert benutzen, ist dies eine offene Tür. Es ist also dringend notwendig hier ein neues, sicheres Passwort zu vergeben (*merken und/oder aufschreiben !*) :



Dies ist nur erster kleiner Schritt den Raspberry Pi abzusichern, insbesondere wenn er mit Internet-Verbindung betrieben werden soll. Weitere Maßnahmen zur Absicherung findet man unter : <https://www.raspberrypi.org/documentation/configuration/security.md>

● **Hostname**

Der Hostname ist der Netzwerkname des Raspberry Pi. Gerade wenn mehrere Pi's in einem Netzwerk aktiv sind, möchte man sie einfach unterscheiden können, also ist hier die Vergabe eines eindeutigen Namens sinnvoll – als Vorschlag hier natürlich 'pillogger' :



● **Boot 'To Desktop' oder 'To CLI'**

Hier kann gewählt werden, ob das System mit der grafischen Oberfläche starten soll oder in der Kommandozeile bleibt (Command Line Interface). Da wir für die PiLogger Basis Software die grafische Oberfläche benötigen, kann die Standardeinstellung 'To Desktop' unverändert aktiviert bleiben.

● **Auto Login**

Diese Option ist standardmäßig so eingestellt, dass beim Systemstart der Benutzer 'pi' automatisch angemeldet wird. Damit wird der Start des Raspberry Pi schnell und unkompliziert, aber auch für jedermann ohne Passwort-Abfrage möglich. Für eine Entwicklungs- und Testumgebung sollte das kein Problem sein – wer sein System sicherer machen möchte, sollte das hier ändern.

● **Network at Boot**

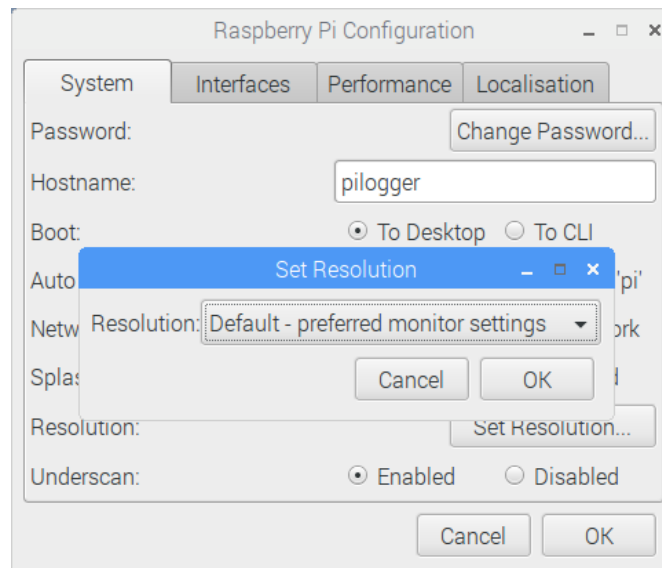
Hier kann eingestellt werden, dass während des Boot-Vorgangs auf eine erfolgreiche Netzwerkverbindung gewartet werden soll. Für ein System, dass direkt nach dem Booten zwingend eine Verbindung braucht (z.B. ein Server) kann das sinnvoll sein. Im Allgemeinen reicht es, wenn diese Option deaktiviert bleibt. Der Systemstart wird dann also nicht verzögert.

- **Splash Screen**

Diese Option ist standardmäßig so eingestellt, dass beim Booten eine grafische Willkommensmeldung angezeigt wird statt der Linux-üblichen Textmeldungen. Geschmackssache.

- **Resolution**

Hier kann aus einer kleinen Auswahl spezieller Bildschirmformate, besonders für LCDs mit geringerer Auflösung, ausgewählt werden. Sinnvoll beispielsweise bei Verwendung des Raspberry Pi Monitors.



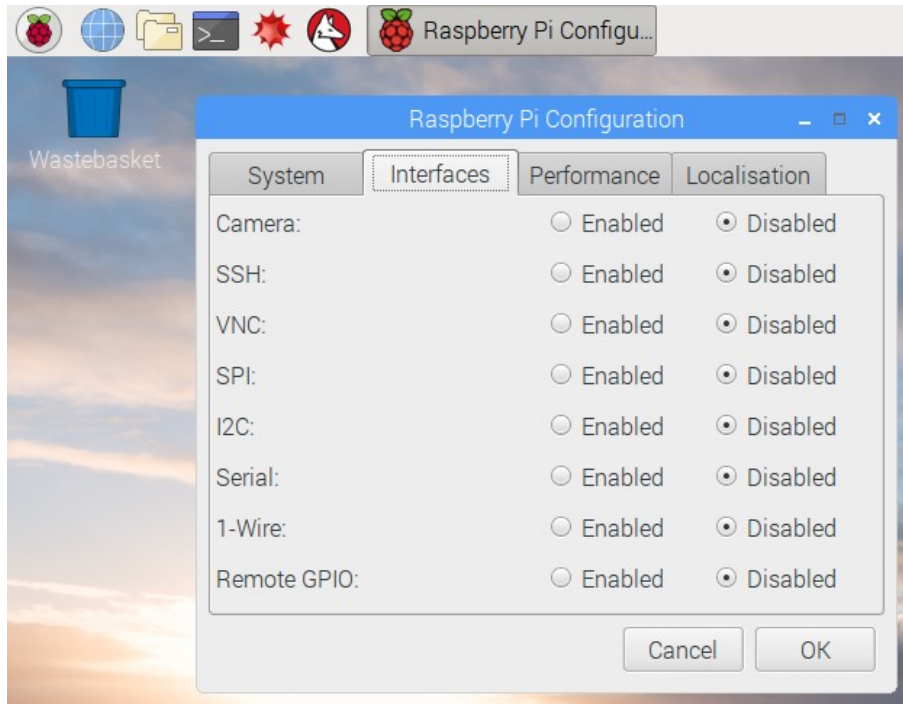
Wenn bisher die Anzeige gut funktioniert hat, gibt es keinen Grund hier etwas zu ändern.

- **Underscan**

Diese Einstellung ist eine Anpassung der Bildschirmdarstellung. Einige Monitore – speziell Fernseher - zeigen einige Bildpunkte (Pixel) am Rand nicht an. Dadurch können Teile des Inhalts weggeschnitten werden. Die aktivierte Einstellung 'Underscan' kann dies verhindern, wobei dann aber auf einigen Monitoren ein schwarzer Rand um den Inhalt erscheint. Einfach mit dem verwendeten Monitor ausprobieren !

3.4.3.2 Schnittstellen

In der nächsten Abteilung 'Interfaces' gibt es folgende Einstellmöglichkeiten:



- **Camera**

Wenn die Kamera-Schnittstelle (CSI) verwendet werden soll, muss dieser Punkt auf 'enabled' gesetzt werden. Falls keine Kamera verwendet wird, spart 'disabled' System-Ressourcen (Boot-Zeit, Speicher).

- **SSH**

SSH steht für 'Secure Shell'. Dies ist eine Möglichkeit den Raspberry Pi mit einem anderen Rechner fernzusteuern. Dabei wird auf dem entfernten (remote) Rechner ein Terminal-Programm benutzt, um die Kommandozeile (CLI) des Pi anzuzeigen. Das 'secure' bedeutet, dass die Verbindung gesichert wird, also eine Authentifizierung und Verschlüsselung benutzt wird. SSH kann über eine direkte serielle Verbindung (UART; an der Erweiterungsleiste anschließbar) oder über Netzwerk benutzt werden.

Wenn der Pi mit dem öffentlichen Internet verbunden wird, sollten sichere Passwörter für alle Benutzer eingerichtet werden, bevor SSH aktiviert wird.

Empfehlung : deaktiviert lassen, wenn es nicht wirklich benötigt wird.

- **VNC**

VNC (Virtual Network Computing) ermöglicht eine entfernte Bedienung des Raspberry Pi über eine Netzwerkverbindung. Dabei überträgt ein Server-Programm auf dem Pi den aktuellen grafischen Bildschirminhalt an den entfernten Rechner und nimmt im Gegenzug von diesem Eingaben von Tastatur und Maus oder Touchpad entgegen. Dazu wird auf dem entfernten (steuernden) Rechner das passende Client-Programm benötigt.

Raspian bringt hierfür das Programm RealVNC mit, das für private, nicht kommerzielle Verwendung kostenlos benutzt werden kann. Für eine gesicherte und verschlüsselte

Verbindung über das Internet ist eine Registrierung und Einrichtung eines Zugangs erforderlich. Die Verbindung wird dann über einen Cloud-Server hergestellt.

Mehr Information dazu gibt es hier :

<https://www.realvnc.com/de/connect/docs/raspberry-pi.html>

Achtung: Das Aktivieren dieser Option startet beim Booten immer automatisch den VNC-Server !

- **SPI**

Diese Option aktiviert die Verwendung der SPI (serial peripheral interface) auf der Erweiterungsleiste des Raspberry Pi. Diese Schnittstelle wird vom PiLogger One nicht benutzt, kann also deaktiviert bleiben.

- **I2C**

Diese Schnittstelle wird vom PiLogger One benötigt.

Das Aktivieren dieser Option ist also unbedingt notwendig!

Weitere Details zu dieser Schnittstelle folgen in Kapitel 4.1 und weiteren.

- **Serial**

Bei einer Standard Installation von Raspbian ist die primäre UART Schnittstelle (serial0) der klassischen Linux Console zugewiesen. Der Raspberry Pi verfügt über 2 solcher Schnittstellen, die je nach Modell unterschiedlich verwendet werden.

Diese Schnittstelle ist die oben bei 'SSH' bereits erwähnte Schnittstelle für eine direkte Verbindung mit der Raspberry Pi Kommandozeile. Physikalisch liegt sie auch auf der Erweiterungsleiste.

In den verschiedenen Versionen von Raspbian ist diese Option mal gesetzt und mal nicht. In einigen Fällen ist die Linux-Console die letzte Rettungsoption, deshalb hier die Empfehlung: Auf aktiviert setzen.

3.4.3.3 Leistung

Die nächste Abteilung heißt 'Performance'. Hier können zwei elementare Einstellungen zur Systemleistung des Raspberry Pi geändert werden : Der Prozessor-Takt und der Grafikspeicher. Die Einstellmöglichkeiten variieren je nach Raspberry Pi Modell. Grundsätzlich ist es nur in wenigen Fällen sinnvoll diese Einstellungen zu ändern, zum Beispiel wenn der Pi als Media-Center dienen soll. Für den Einsatz als Daten-Logger geht es aber eher um das Stromsparen, ist eine weitere Leistungssteigerung also eher kontraproduktiv.

Hier nur beispielhaft eine Liste der Einstellmöglichkeiten:

- **Overclock**

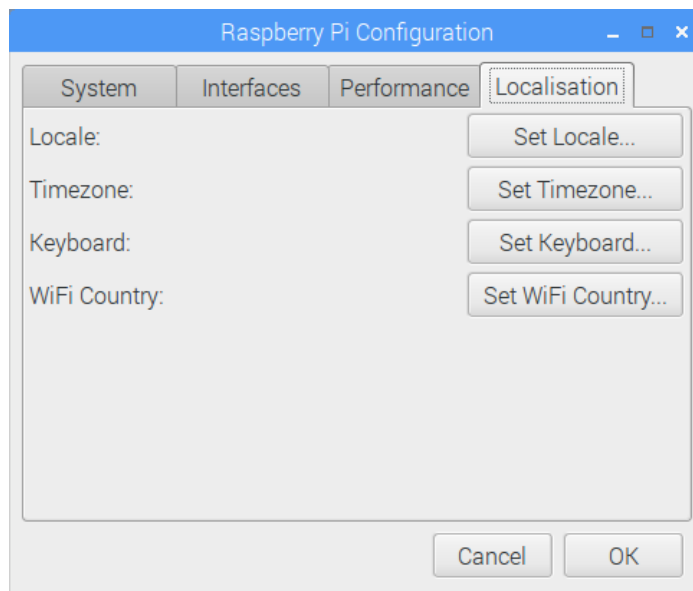
None (700 MHz) ; Modest (800 MHz) ; Medium (900 MHz) ; High (950 MHz) ; Turbo (1000 MHz)

- **GPU Memory**

16 ; 32 ; 48 ; **64** ; 80 ; 96 ; 112 ; 128...880

3.4.3.4 Länderanpassung

Diese Abteilung heisst 'Localisation'. Wörtlich übersetzt: Ortsbestimmung. Es geht hier aber um die Systemeinstellungen zu Land, Sprache und Zeitzone.

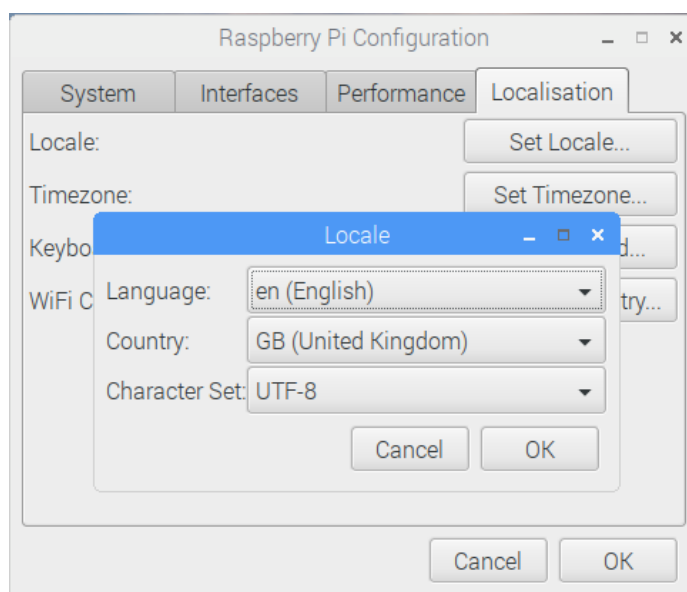


Standardmäßig ist hier 'English UK' voreingestellt – außer der Zeitzone, die auf 'UTC' gesetzt ist. Wer da bleiben möchte kann also vorrücken zu 'Timezone'. Ansonsten müssen alle Punkte einmalig angepasst werden.

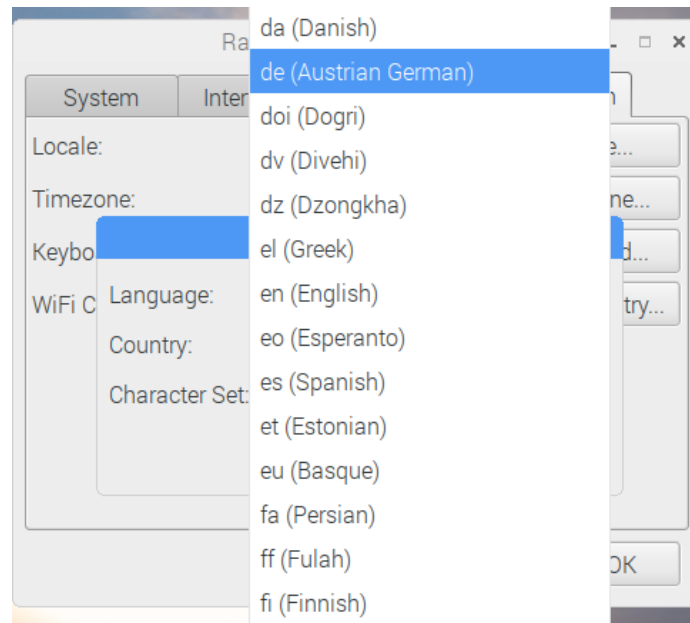
Folgende Menu-Punkte:

- **Locale**

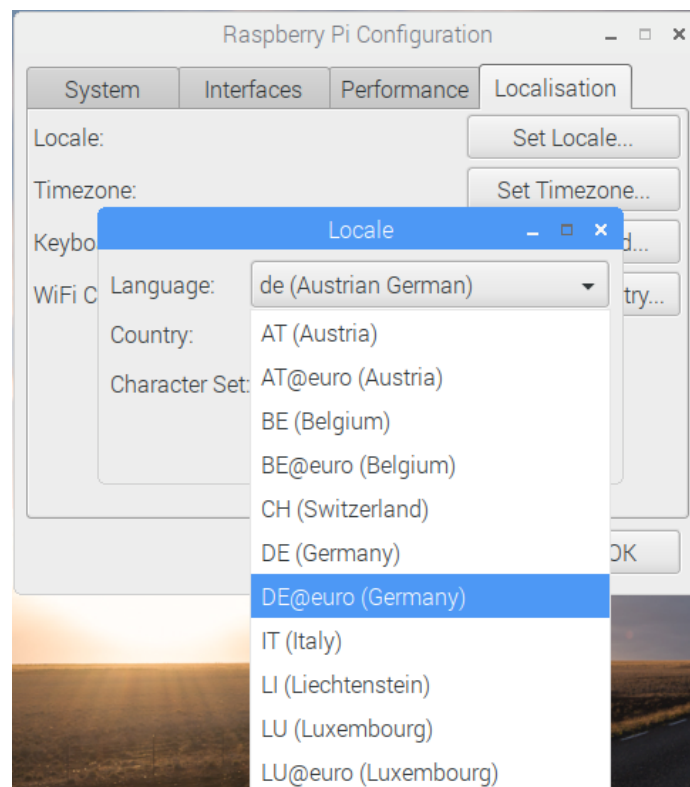
Dies ist die Hauptauswahl für Land, Sprache und Schrift. Klicken auf 'Set Locale' öffnet dieses Untermenü:



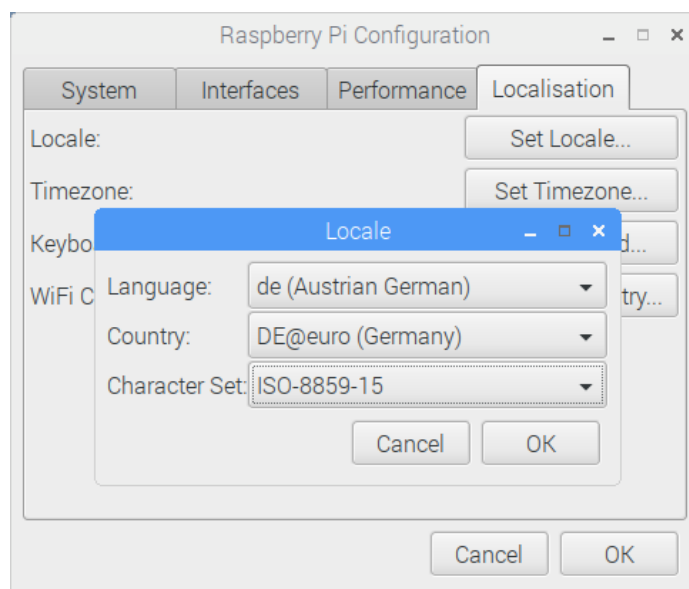
Klicken in die Auswahlbox 'Language' (Sprache) ergibt diese Anzeige:



In diesem Beispiel wird für deutsch also 'de (Austrian German)' gewählt.
Damit wird die Auswahl der nächsten Box 'Country' (Land) automatisch reduziert:



Um das Euro-Zeichen wie gewohnt verwenden zu können empfiehlt es sich die Variante 'DE@euro (Germany)' zu wählen. Damit ist dann auch die Auswahl des nächsten Punktes 'Character Set' automatisch auf 'ISO-8859-15' als einzig möglich reduziert:

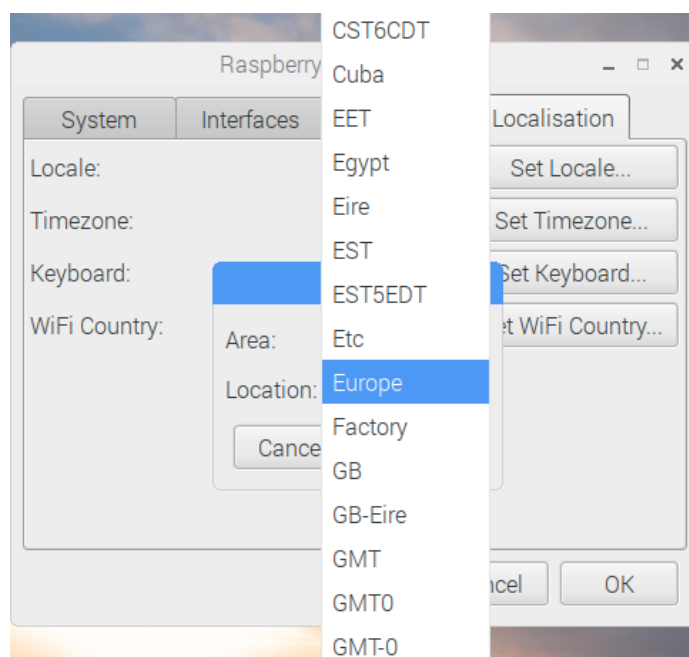


Mit Klicken auf 'OK' ist dieses Untermenü erledigt.

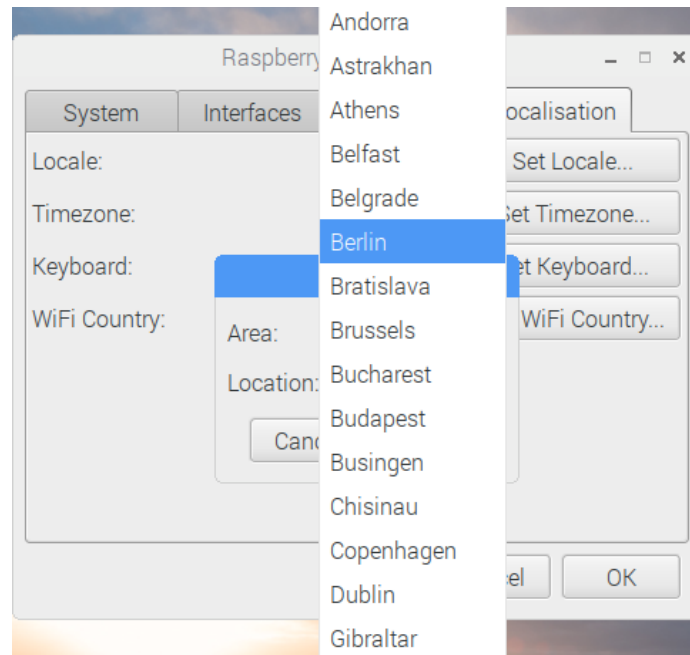
● Timezone

Hier wird die Zeitzone eingestellt in der der Raspberry Pi betrieben wird, damit das Betriebssystem die Datum- und Zeitangaben korrekt behandeln kann.

Ein Klick auf 'Set Timezone' öffnet dieses Untermenü:



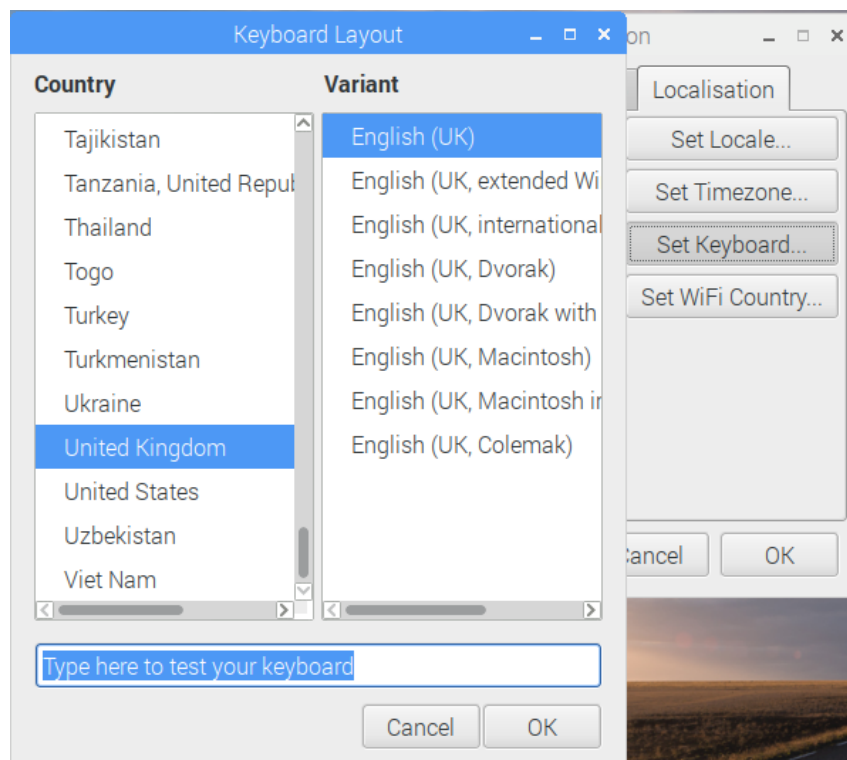
Die Hauptauswahl wird in der ersten Liste 'Area' (Bereich) getroffen. Es gibt einige generelle Optionen wie 'GMT' (Greenwich Mean Time) mit denen die Zeitzone direkt nach dem entsprechenden Standardschema gesetzt wird. Eine mehr alltagsorientierte Auswahl bekommt man, wenn man mit einer Länderregion, also zum Beispiel 'Europe' beginnt. Die zweite Auswahlliste bietet dann dazu passende große Städte an:



Nachdem die passende Stadt, im Beispiel 'Berlin', gewählt wurde, schließt ein Klick auf 'OK' dieses Untermenü ab.

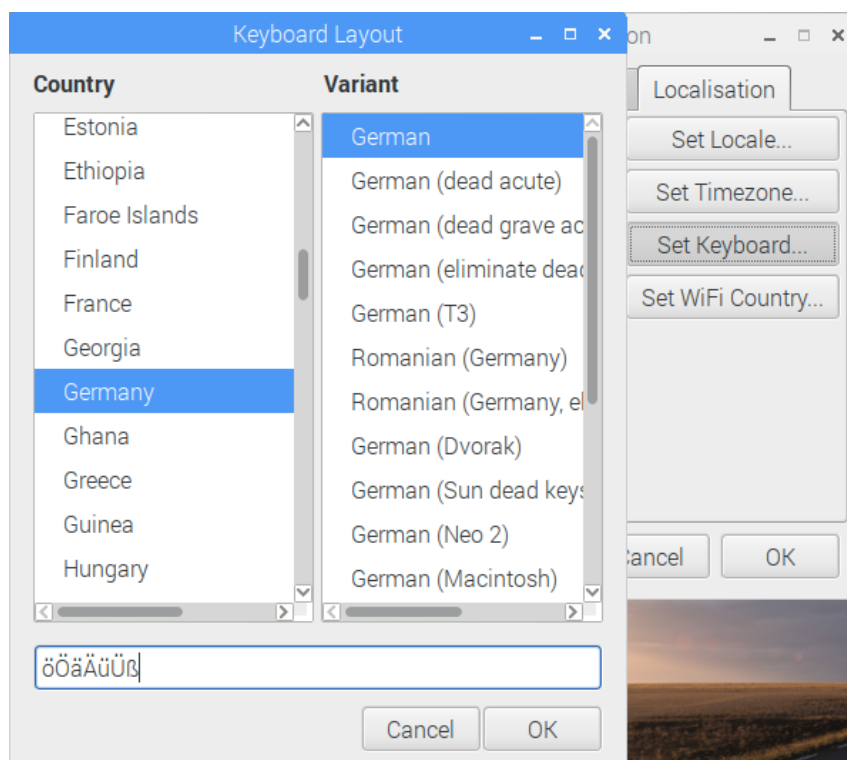
● Keyboard

Nun geht es um die korrekte Anpassung des Tastatur-Layouts. Ein Klick auf 'Set Keyboard' öffnet dieses Untermenü:



Wir beginnen hier in der linken Spalte 'Country' mit der Auswahl des Landes. Danach erscheint in der rechten Spalte 'Variant' die dafür verfügbare Variantenliste.

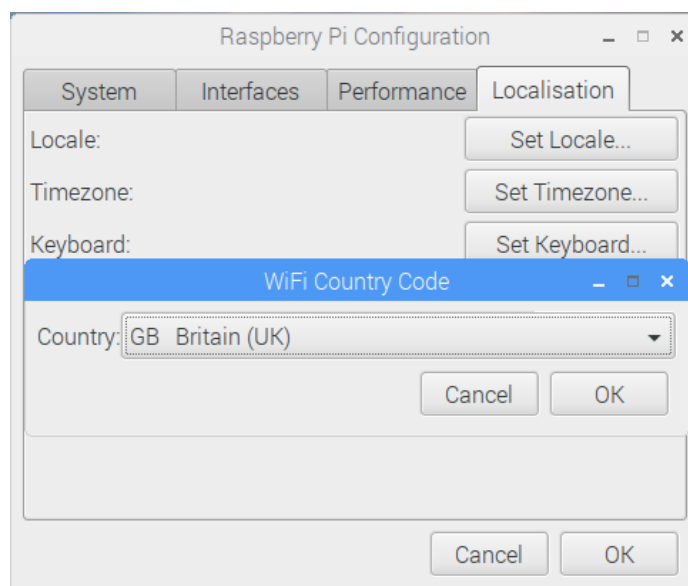
Bei unserem Beispiel für Deutschland heißt dies für 'Country' dann 'Germany' und die Auswahl von Varianten ist überraschend groß:



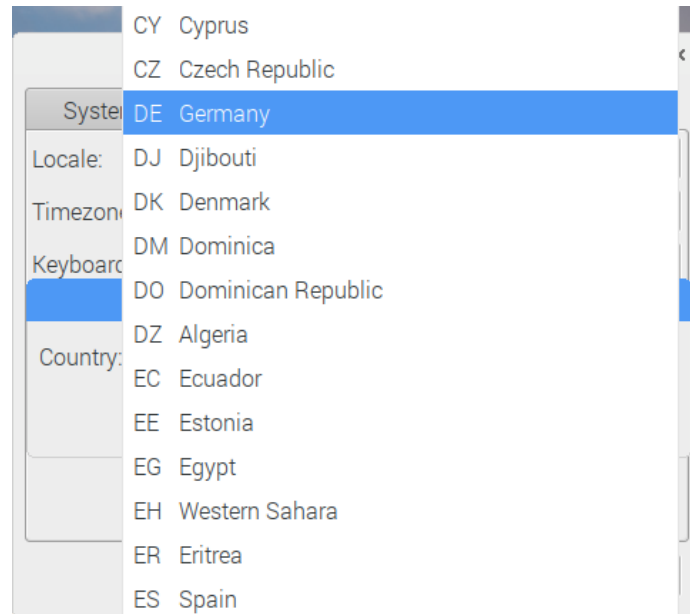
Das hat damit zu tun, dass es sehr viele verschiedene Ausführungen von Tastaturen gibt, die unterschiedlich viele Tasten bieten. In den meisten Fällen sollte die erste Position die gebräuchlichste sein. Für Zweifelsfälle gibt es unter den beiden Auswahlspalten eine Testzeile, in der der Hinweis 'Type here to test your keyboard' steht, also: 'hier schreiben um Ihre Tastatur zu testen'. Ein Test mit den typischen Sondertasten für 'äÄöÖüÜß' zeigt schnell, ob die Auswahl erwartungsgemäß funktioniert. Abgeschlossen wird wieder mit einem Klick auf 'OK'.

● WiFi Country

Dieser Menüpunkt setzt die Landeseinstellung für die korrekte Konfiguration des WLAN-Moduls.

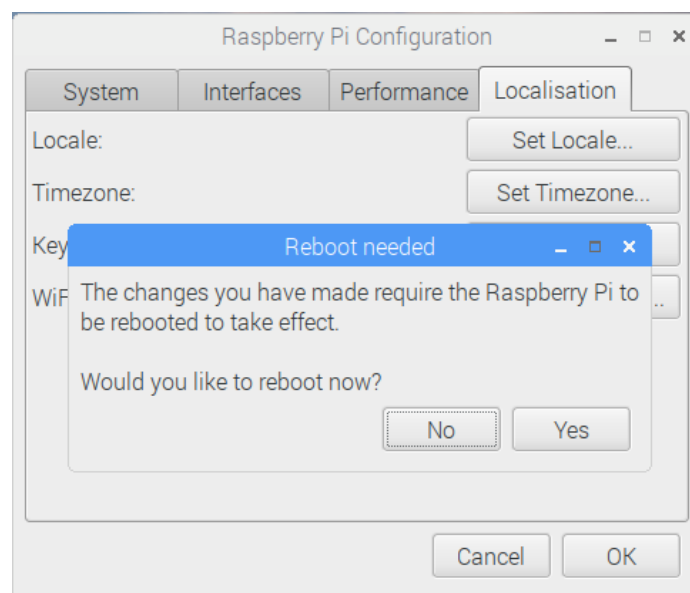


Für Deutschland wählen wir einfach 'DE Germany' aus:



Ein Klick auf 'OK' beendet diese Untermenü.

Nun sind alle notwendigen Einstellungen erfolgt und wir können auch im Hauptfenster auf 'OK' klicken.

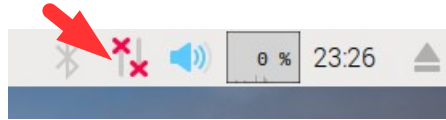


Es erscheint ein Rückfrage-Fenster (Confirmation Window), dass uns erstens informiert, dass für die Wirksamkeit der Änderungen ein Neustart notwendig ist und zweitens fragt, ob wir jetzt neu starten wollen – da wir mit den Einstellungen fertig sind, macht es Sinn das Ganze jetzt auch umzusetzen : Drücken wir also auf 'Yes'.

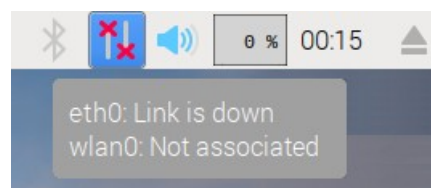
3.4.4 Internet-Verbindung herstellen

Nachdem die Erstkonfiguration abgeschlossen ist und der Raspberry Pi neu gestartet ist, und jetzt auch die richtige Sprache mit uns spricht, ist es jetzt an der Zeit eine Netzwerkverbindung zu testen.

Auf der rechten Seite der Startleiste ist ein Symbol für Netzwerkverbindungen und deren Zustand zu sehen:



In diesem Fall wird mit den beiden 'x' angezeigt, dass keine Verbindung besteht. Fahren wir mit dem Mauszeiger auf dieses Symbol, werden weitere Informationen hierzu angezeigt:

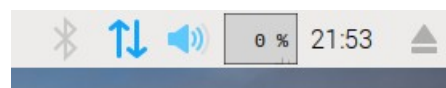


Die Zeile 'eth0: Link is down' besagt, dass keine LAN-Kabelverbindung besteht – kein Kabel eingesteckt ist (eth0 = Ethernet Schnittstelle Nummer 0).

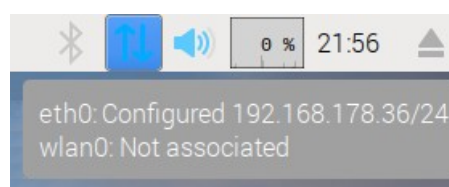
Die Zeile darunter 'wlan0: Not associated' taucht nur auf, wenn wie in diesem Fall ein WLAN-Stick wie in 3.4.2 bereits eingesteckt wurde, der Stick normal erkannt wurde und auch (wie viele) problemlos unterstützt wird. Deshalb wird er hier auch bereits als mögliche Netzwerkverbindung verwaltet, konnte aber noch nicht erfolgreich mit einem WLAN-Netz verbinden.

● Kabelverbindung

Für eine Kabelverbindung ist einfach ein Kabel zum Router notwendig – Einstecken und die Verbindung wird automatisch aufgebaut. Dabei blinken die Kontrollleuchten an der LAN-Buchse des Raspberry Pi und auch das Startleistensymbol wird animiert. Die erfolgreiche Verbindung wird so angezeigt:

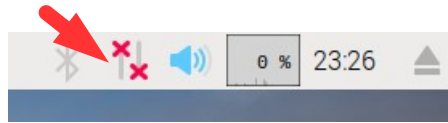


Der Mauszeiger zeigt dann diese Zusatzinformationen:

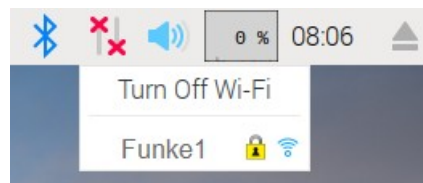


● WLAN-Verbindung

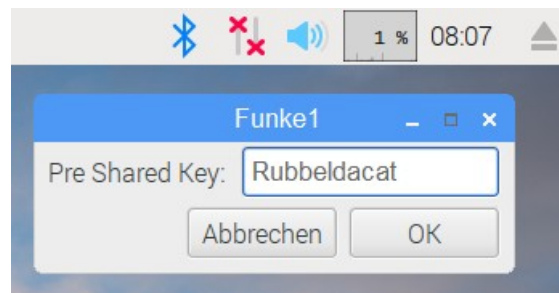
Für den Fall einer WLAN-Verbindung ist jetzt die erste Anmeldung am Router notwendig. Dazu klicken wir mit der linken Maus-Taste auf das Netzwerk-Symbol:



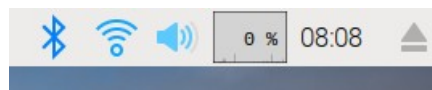
Daraufhin werden die in der Umgebung gefundenen WLAN-Netze gezeigt:



Mit einem Links-Klick wird das Heimatnetz ausgewählt, hier 'Funke1'. Damit wird ein Abfragefenster für den Netzwerkschlüssel zur Anmeldung geöffnet:



Nach dem Eintippen des Schlüssels wird die Eingabe mit einem Klick auf 'OK' abgeschlossen. Das Fenster wird geschlossen und das Netzwerk-Symbol ist bei erfolgreicher Verbindungsaufnahme nun ein Funk-Symbol.



3.4.5 Allgemeines Update

Die neue Betriebssystem-Installation wird erst richtig rund und sicher, wenn wir ein erstes allgemeines Update machen, also eventuell verfügbare, neuere Betriebs-Software suchen und einspielen. Insbesondere sicherheitsrelevante Updates werden so schnell wie möglich verteilt, also auch außer der Reihe von kompletten Installations-Images.

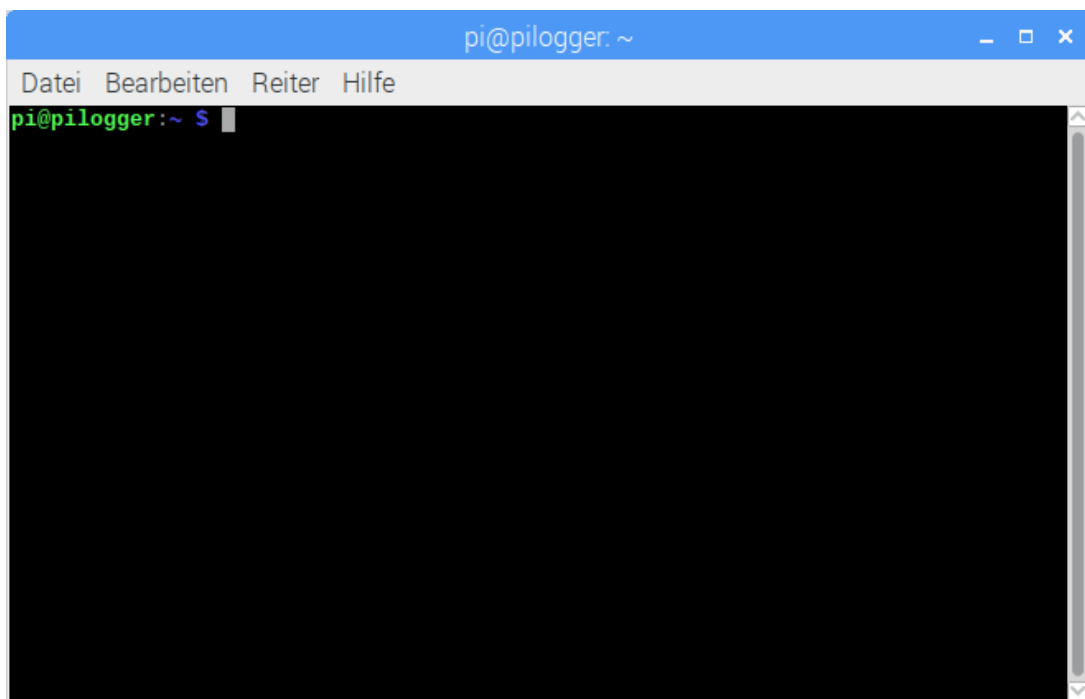
Die folgenden Aktionen setzen eine funktionierende Internet-Verbindung voraus.

Der erste Teil ist ein Update der Firmware. Das ist der Teil der die Raspberry Pi spezifischen Hardware-Anpassungen für das Betriebssystem zur Verfügung stellt (der angepasste Linux Kernel). Dazu starten wir ein Konsolenfenster – das ist eine grafische Version der Kommandozeile, zeigt also in einem Fenster einen reinen Textbildschirm an. Das machen wir mit einem Klick auf das 'Konsole'-Symbol in der Startleiste:



Der Mauszeiger informiert uns beim Überfahren des Symbols, dass das Programm 'LXTerminal' heißt. Das ist korrekt – das ist das aufzurufende Programm. Es ahmt, wie gesagt, einen einfachen Textbildschirm nach (emuliert), so wie historisch ein 'Terminal' oder eine 'Console' die ersten Ausgabebildschirme für einen 'Host'-Computer waren. *Genau diese Ausgabe kann man auch entfernt (remote) über SSH erhalten.*

Das Fenster sieht dann so aus:



Die Fenster-Titelzeile durch das 'pi@pilogger', dass die aktuelle Sitzung (session) dem

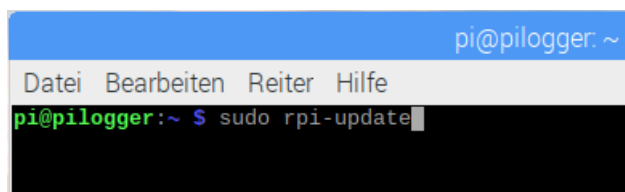
Nutzer 'pi' auf dem Computer 'pillogger' zugeordnet ist. Der Doppelpunkt und die Tilde bedeuten, dass aktuell das Verzeichnis 'home' des aufrufenden Benutzers, also 'pi' aktiv ist und die Ortsangaben-Basis ist.

Den gleichen Text findet man auch im interaktiven Bereich des Fensters, gefolgt vom Dollar-Zeichen, dem eigentlichen sogenannten 'prompt'. Der Prompt ist historisch ein einzelnes Zeichen gewesen, das zu einer Eingabe auffordert. Der Computer zeigt damit, dass alle Aufgaben abgearbeitet sind und er bereit für eine neue Eingabe, einen neuen Befehl ist. Mittlerweile ist der Prompt der ganze Text einschließlich '\$' - der erweiterte Prompt. Dahinter blinkt mit einem Zeichen Abstand ein graues Rechteck – der Text-Cursor. An dieser Stelle wird nun der Text angezeigt, der über die Tastatur vom Benutzer eingegeben wird – eine Kommandozeile.

Für unsere Aufgabe, die Firmware des Raspberry Pi zu aktualisieren, geben wir folgende Text exakt ein:

```
sudo rpi-update
```

Das sieht dann wie folgt aus:



'sudo' ist ein Systembefehl der dem nachfolgenden Kommando die Rechte eines Super-Users, dem Benutzer 'root', zuweist. Damit sind fast alle Sicherheitsbeschränkungen aufgehoben und normalerweise wird diese Rechte-Erhöhung nur mit der Eingabe des zugehörigen Passwortes autorisiert. Da der Raspberry Pi aber vorher auf 'automatische Anmeldung' in einem alleinigen Verwalter-Konto konfiguriert war, entfällt diese Passwort-Abfrage hier.

Wie bereits gesagt, kann dies für ein Entwicklungssystem toleriert werden – für ein eigenständig arbeitendes 'Produktiv-System' sollte dies aber als Sicherheitsrisiko betrachtet werden und rechtzeitig geändert werden!

Der Teil 'rpi-update' ruft ein spezielles Hilfs-Programm (tool) auf, das über eine bestehende Internet-Verbindung den Update-Server nach einer neueren Version abfragt und, falls eine verfügbar ist, auch installiert.

Die Eingabe wird mit der Taste 'Return': ↵ abgeschlossen. Damit ist die Kommandozeile komplett und der Befehl abgeschickt – die Ausführung beginnt.

Während der Ausführung werden etliche Meldungen ausgegeben und wenn eine neuere Version vorhanden ist, dauert der gesamte Vorgang einige Minuten.

Die komplette Ausgabe eines erfolgreichen Durchlaufs sieht dann etwa so aus:

```

pi@pillogger:~ $ sudo rpi-update
*** Raspberry Pi firmware updater by Hexxeh, enhanced by AndrewS and Dom
*** Performing self-update
  % Total      % Received % Xferd  Average Speed   Time    Time     Time  Current
 100 13403 100 13403    0     0  45454      0 --:--:-- --:--:-- --:--:-- 45433
*** Relaunching after update
*** Raspberry Pi firmware updater by Hexxeh, enhanced by AndrewS and Dom
*** We're running for the first time
*** Backing up files (this will take a few minutes)
*** Backing up firmware
*** Backing up modules 4.9.41-v7+
#####
This update bumps to rpi-4.9.y linux tree
Be aware there could be compatibility issues with some drivers
Discussion here:
https://www.raspberrypi.org/forums/viewtopic.php?f=29&t=167934
#####
*** Downloading specific firmware revision (this will take a few minutes)
  % Total      % Received % Xferd  Average Speed   Time    Time     Time  Current
 100   168      0   168    0     0    279      0 --:--:-- --:--:-- --:--:--   279
 100 54.1M 100 54.1M    0     0 2107k      0 0:00:26 0:00:26 --:--:-- 4706k
*** Updating firmware
*** Updating kernel modules
*** depmod 4.9.43-v7+
*** depmod 4.9.43+
*** Updating VideoCore libraries
*** Using HardFP libraries
*** Updating SDK
*** Running ldconfig
*** Storing current firmware revision
*** Deleting downloaded files
*** Syncing changes to disk
*** If no errors appeared, your firmware was successfully updated to bf19fe7aa723ef8dda7a7205e35749ddc
*** A reboot is needed to activate the new firmware
pi@pillogger:~ $

```

Am Ende erscheint wieder, wie oben erklärt, der Prompt – die Eingabeaufforderung. In der letzten Meldung des Firmware-Updates steht, dass ein Neustart notwendig ist, um die neue Firmware zu aktivieren. Für den noch geplanten zweiten Teil ist dies in der Regel nicht notwendig. Wer auf Nummer 'Sicher' gehen will, startet an dieser Stelle mit diesem Kommando den Pi neu:

```
reboot
```

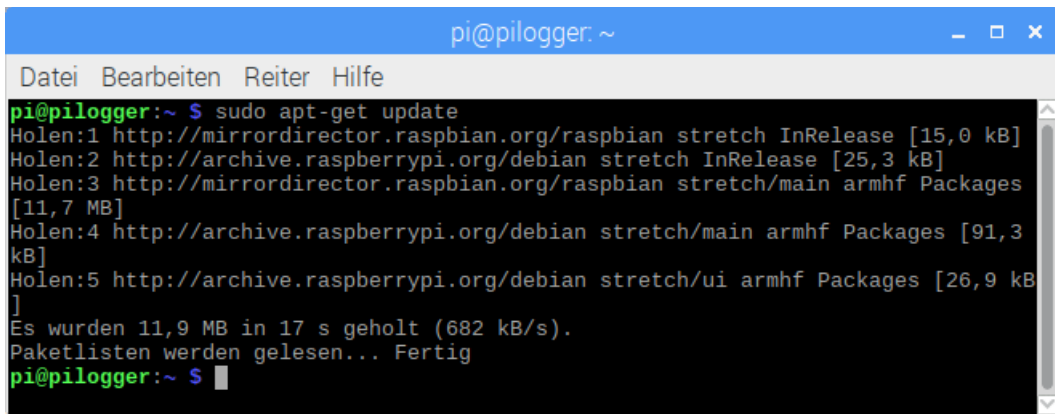
Das geht schnell und unkompliziert.

Nun kann ein allgemeines Upgrade der Raspbian-Distribution vorbereitet werden. Dazu muss die lokale Datenbank der installierten und verfügbaren Software-Pakete aufgefrischt werden.

Nach einem Neustart muss wieder ein Terminal-Fenster geöffnet werden, dann geben wir 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 – je frischer die verwendete Raspbian-Version ist, umso kürzer. Die Ausgabe sieht etwa so aus:



```

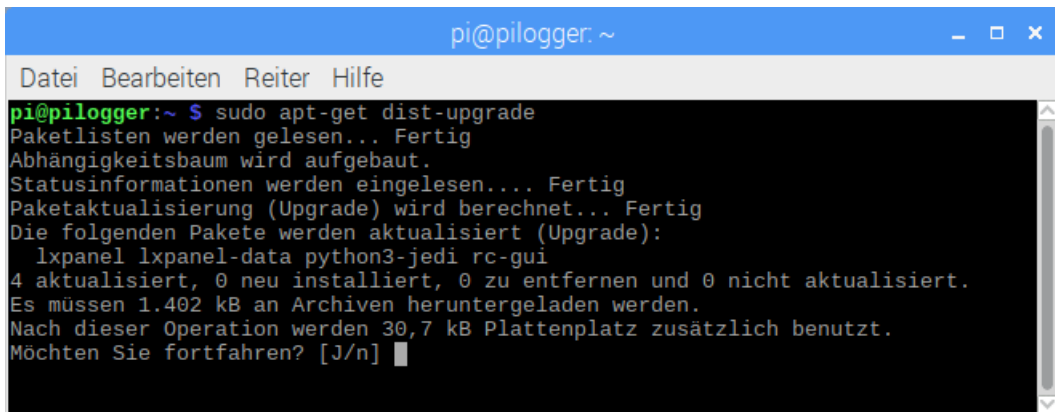
pi@pillogger: ~
Datei Bearbeiten Reiter Hilfe
pi@pillogger:~ $ sudo apt-get update
Holen:1 http://mirrordirector.raspbian.org/raspbian stretch InRelease [15,0 kB]
Holen:2 http://archive.raspberrypi.org/debian stretch InRelease [25,3 kB]
Holen:3 http://mirrordirector.raspbian.org/raspbian stretch/main armhf Packages
[11,7 MB]
Holen:4 http://archive.raspberrypi.org/debian stretch/main armhf Packages [91,3
kB]
Holen:5 http://archive.raspberrypi.org/debian stretch/ui armhf Packages [26,9 kB
]
Es wurden 11,9 MB in 17 s geholt (682 kB/s).
Paketlisten werden gelesen... Fertig
pi@pillogger:~ $

```

Nun geben wir die folgende Textzeile ein:

```
sudo apt-get dist-upgrade
```

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



```

pi@pillogger: ~
Datei Bearbeiten Reiter Hilfe
pi@pillogger:~ $ sudo apt-get dist-upgrade
Paketlisten werden gelesen... Fertig
Abhängigkeitsbaum wird aufgebaut.
Statusinformationen werden eingelesen.... Fertig
Paketaktualisierung (Upgrade) wird berechnet... Fertig
Die folgenden Pakete werden aktualisiert (Upgrade):
  lxpanel lxpanel-data python3-jedi rc-gui
4 aktualisiert, 0 neu installiert, 0 zu entfernen und 0 nicht aktualisiert.
Es müssen 1.402 kB an Archiven heruntergeladen werden.
Nach dieser Operation werden 30,7 kB Plattenplatz zusätzlich benutzt.
Möchten Sie fortfahren? [J/n]

```

Nach der Eingabe von 'J' mit Abschluss durch 'return' beginnt der Hauptteil des Updates. Je nachdem wie frisch das verwendete Raspbian Image ist, müssen eventuell etliche Pakete upgedatet werden. Das kann dann schon mal bis zu einer Stunde dauern. Das gezeigte Beispiel ist allerdings mit einer wirklich frische Installation gemacht, also ziemlich kurz.

Nach Abschluss dieses Updates und einem eventuellen Neustart, ist der Raspberry Pi grundsätzlich eingerichtet.

4 PiLogger Basis-Software einrichten

Für die Verwendung der PiLogger Basis Software müssen einige Voraussetzungen erfüllt sein, die erfreulicherweise in der Version 'Stretch' von Raspbian alle schon vorinstalliert sind:

- Der I²C-Kernel-Treiber für die Hardware-Schnittstelle muss geladen werden
→ haben wir in 3.4.3.2 per 'raspi-config' aktiviert
- Das Kommandozeilen Werkzeug 'i2c tools' ist installiert (mehr im Anhang 13.1)
- Die Unterstützung für I²C unter Python 'python3-smbus' ist installiert
- Python ist in der Version 3.x installiert
- Die grafische Erweiterung für Python 'tkinter' ist installiert

Damit kann es eigentlich sofort los gehen - leider geht bei Updates nicht immer alles gut, weshalb wir als Nächstes einen alternativen I²C Treiber aktivieren müssen.

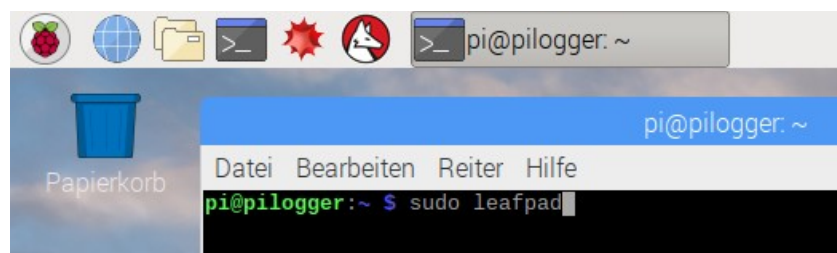
4.1 Alternativen I²C Treiber aktivieren

Seit der Umstellung von Raspbian auf die Kernel-Version 4.9.x ist standardmäßig ein neuer Treiber für die I²C Schnittstelle eingestellt, der derzeit noch ein Problem hat. Solange dies der Fall ist, müssen wir als Hilfslösung (workaround) den alten Treiber wieder aktivieren (fallback).

Dazu öffnen wir ein Terminal-Fenster und geben ein:

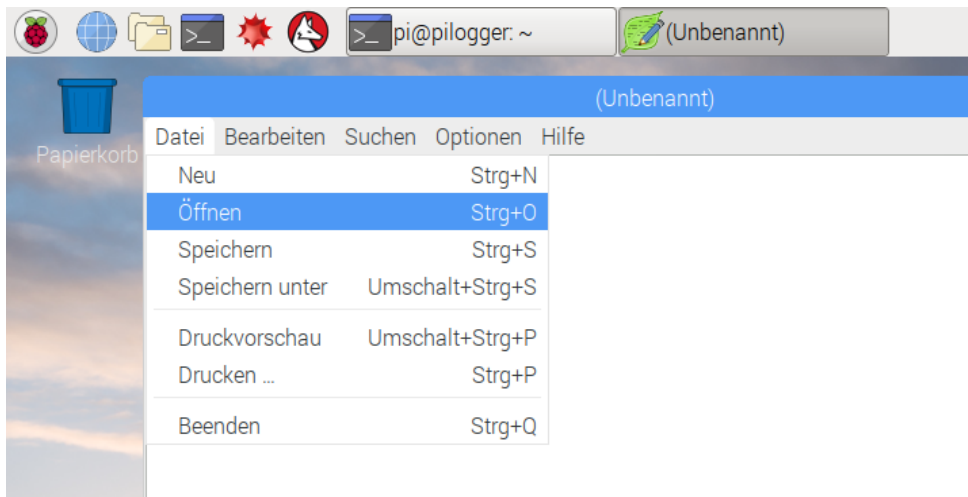
```
sudo leafpad
```

Das sieht dann so aus:

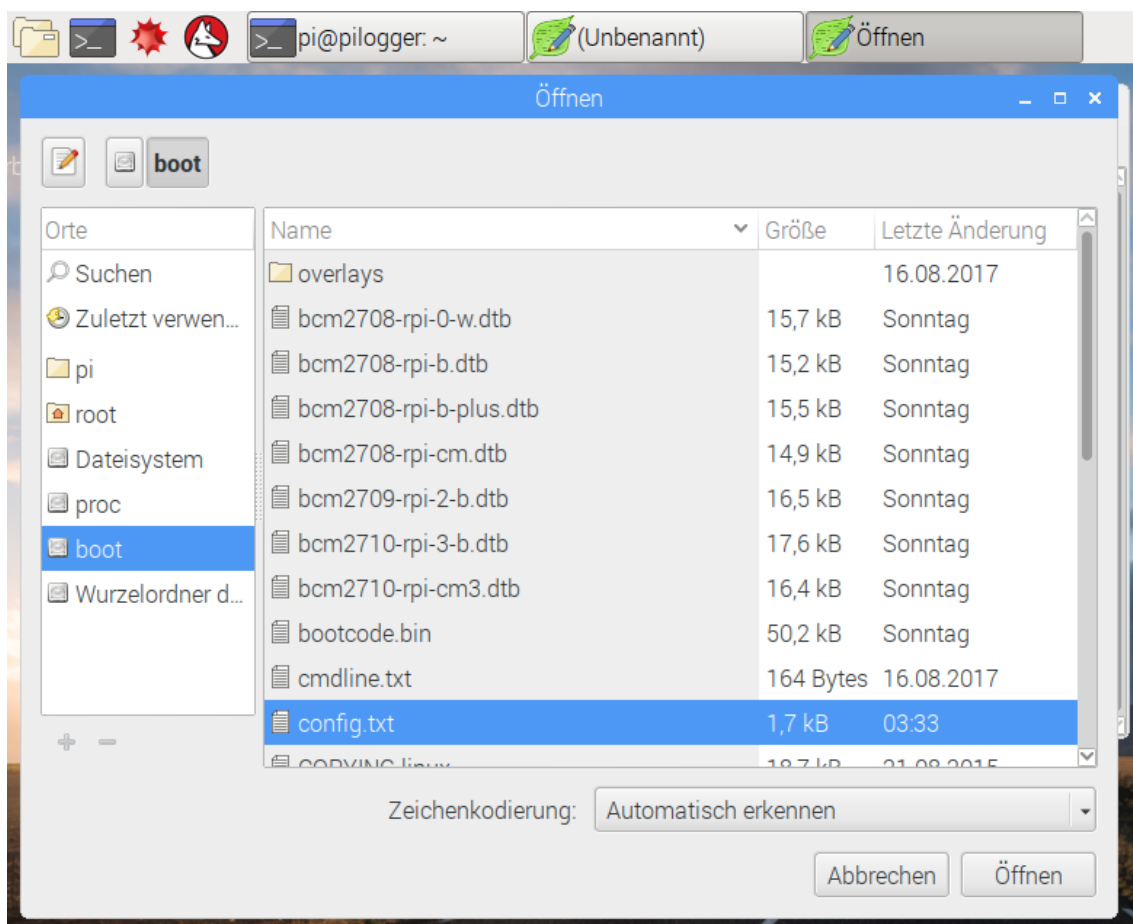


Damit wird der Text-Editor 'Leafpad' mit Administratorrechten geöffnet, um damit eine System-Datei zu ändern.

Im Editor-Fenster gehen wir auf 'Datei' → 'Öffnen' :

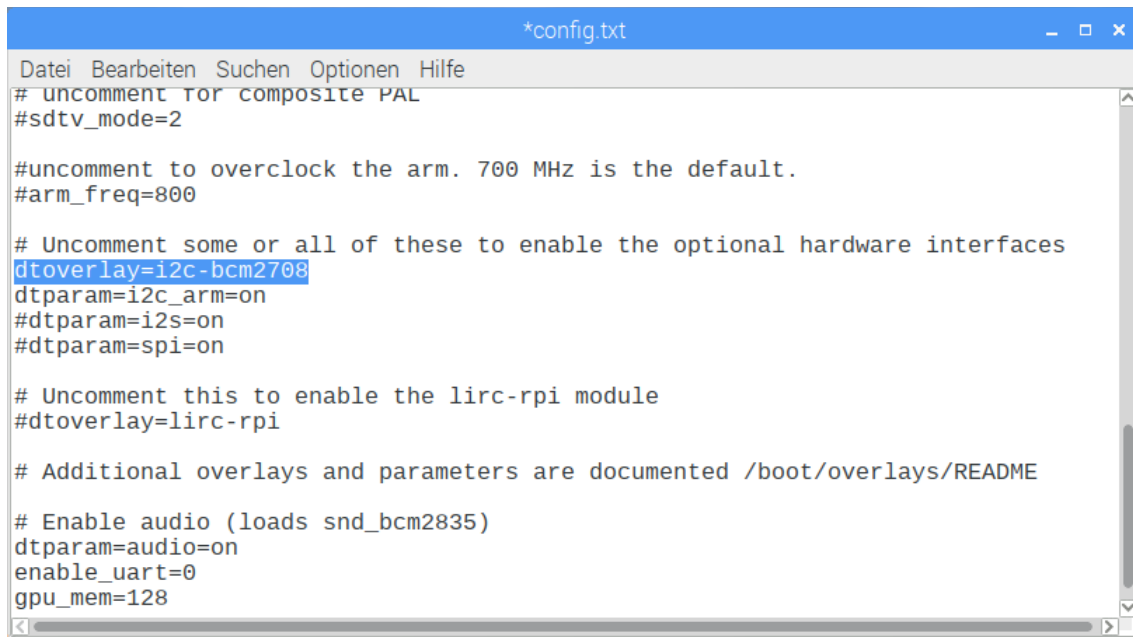


Das öffnet ein Dialog-Fenster in dem wir die Text-Datei durch Browsen auswählen:



In der linken Spalte 'Orte' klicken wir zunächst auf 'boot', woraufhin im rechten Fenster der Inhalt dieses Ordners (in diesem Fall sogar eine Partition) angezeigt wird. Dort wählen wir die Datei 'config.txt' und klicken anschließend rechts unten auf den Knopf 'Öffnen'.

Nachdem sich die Datei geöffnet hat, fahren wir im Text zu der Stelle, wo das I²C Interface bereits durch das Tool 'raspi-config' aktiviert worden ist :
 'dtparam=i2c_arm=on'
 und fügen davor eine neue Zeile ein:



```
*config.txt
Datei Bearbeiten Suchen Optionen Hilfe
# uncomment for composite PAL
#sdtv_mode=2

#uncomment to overclock the arm. 700 MHz is the default.
#arm_freq=800

# Uncomment some or all of these to enable the optional hardware interfaces
dtoverlay=i2c-bcm2708
dtparam=i2c_arm=on
#dtparam=i2s=on
#dtparam=spi=on

# Uncomment this to enable the lirc-rpi module
#dtoverlay=lirc-rpi

# Additional overlays and parameters are documented /boot/overlays/README

# Enable audio (loads snd_bcm2835)
dtparam=audio=on
enable_uart=0
gpu_mem=128
```

'dtoverlay=i2c-bcm2708'

Anschließend speichern wir die Datei ('Datei' → 'Speichern') und schließen den Editor ('Datei' → 'Beenden').

Jetzt schließen wir noch das Terminal-Fenster und starten den Raspberry Pi neu.

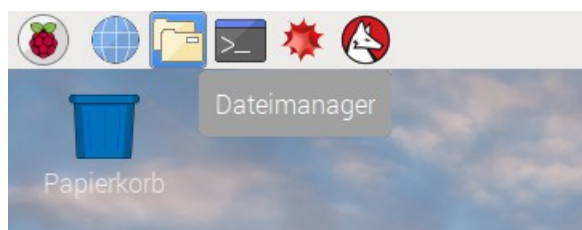
Damit ist jetzt der alte I²C Treiber aktiviert.

4.2 Einrichten der PiLogger Basis-Software

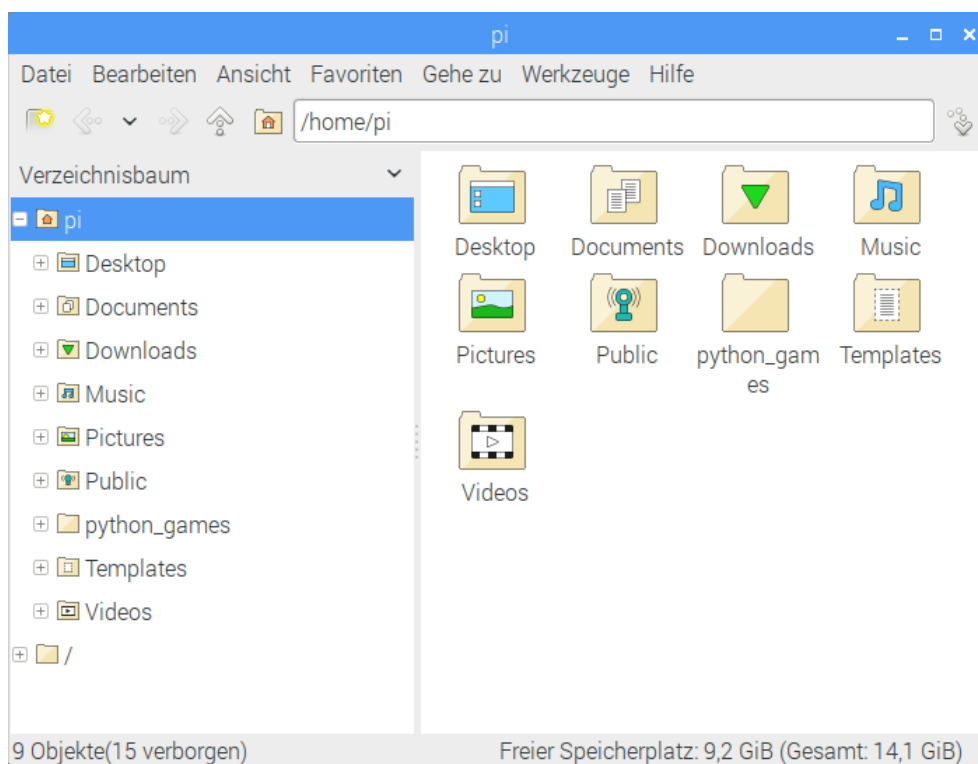
Die PiLogger Basis-Software benötigt keine Installation im eigentlichen Sinne. Es muss lediglich ein Archiv auf den Raspberry Pi geholt werden und in ein sinnvolles Verzeichnis entpackt werden. Der empfohlene Ort sollte im Sinne dieses Handbuchs zunächst eingehalten werden – eine Änderung ist später leicht möglich.

4.2.1 Neuen Ordner erstellen

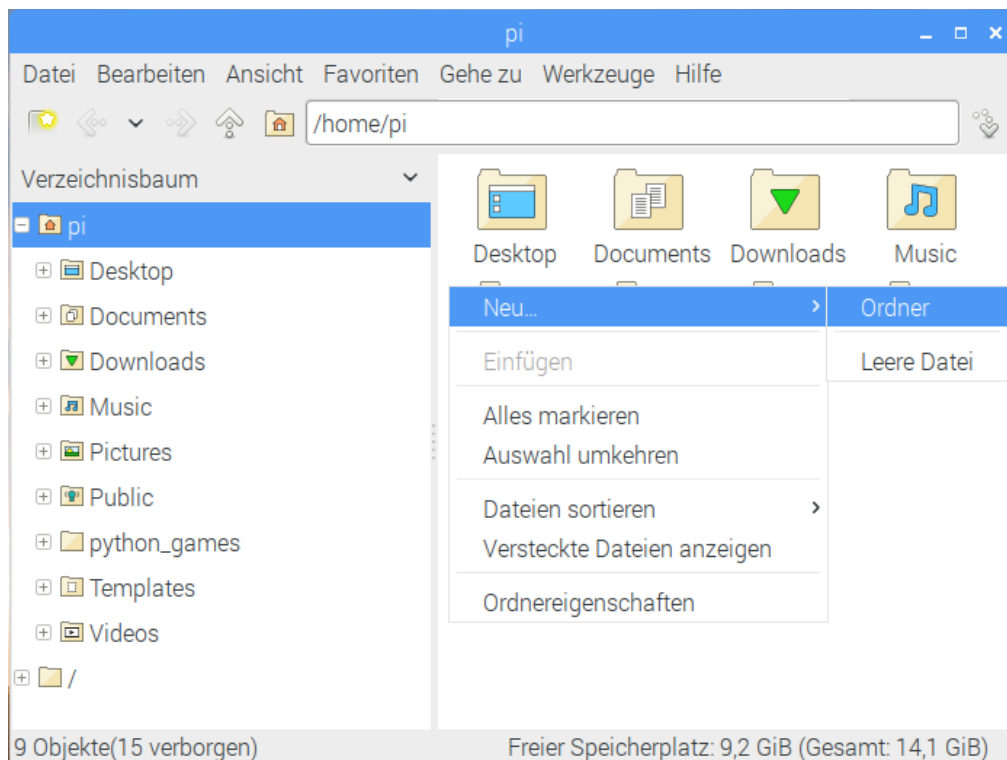
Dazu bereiten wir zunächst dieses Verzeichnis vor. Wir öffnen den Dateimanager mit einem Maus-Klick auf das Ordner-Symbol in der Task-Leiste:



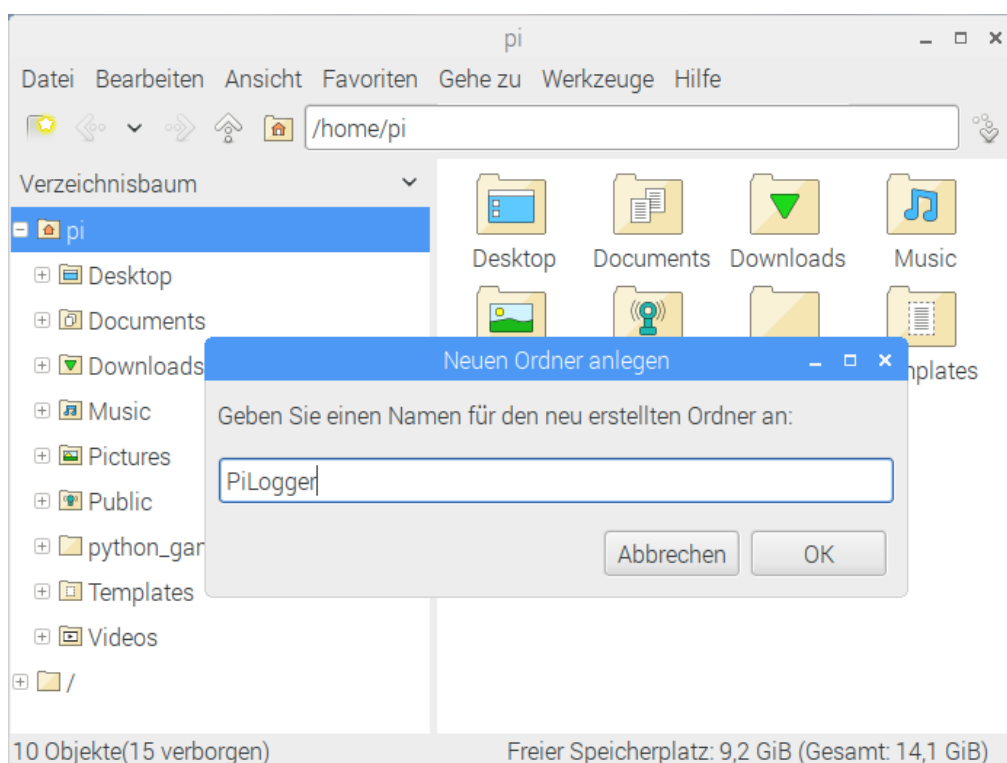
Der Dateimanager öffnet sein Fenster standardmäßig mit dem Verzeichnis /home/pi – dem Heimatomdner für den Benutzer 'Pi':



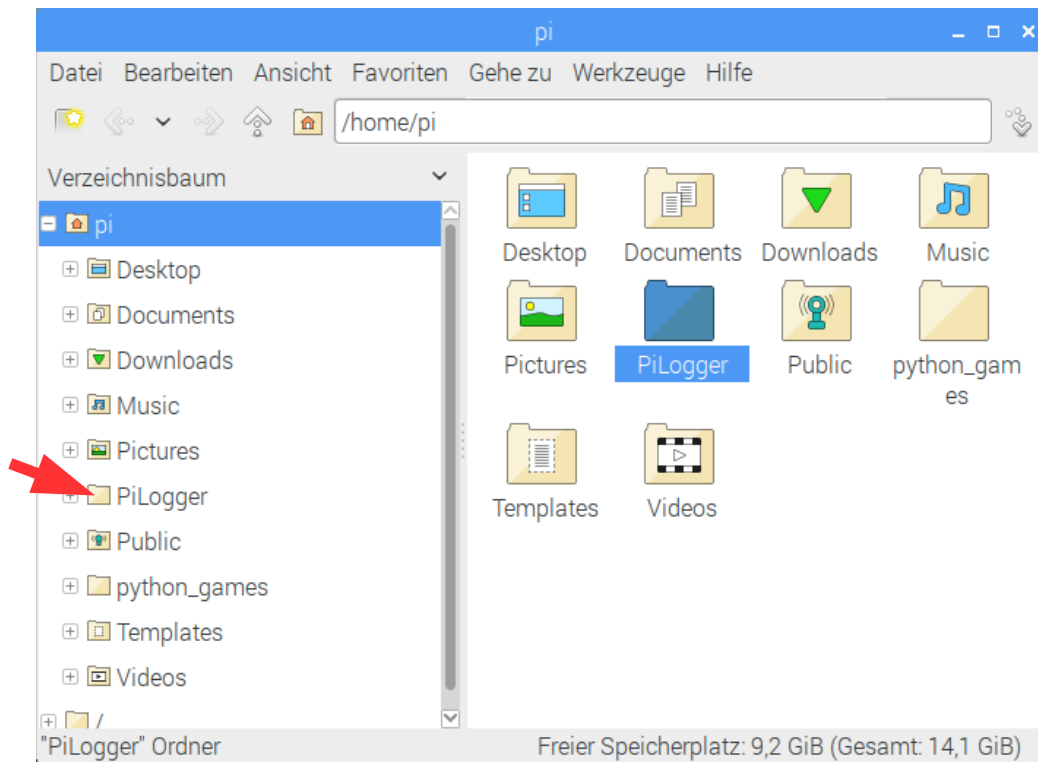
Hier ist ein guter Ort um einen neuen Ordner für den PiLogger anzulegen. Dazu öffnen wir mit einem Rechts-Klick im Hauptfenster das Kontext-Menü und gehen über 'Neu' rechts nach 'Ordner':



Nach einem Links-Klick öffnet sich eine Eingabemaske für den Namen des neuen Ordners:



Nach der Eingabe von 'PiLogger' und dem Klicken auf 'OK' reiht sich der gewünschte Ordner so ein:



Das ist nun der Heimat-Ordner für die Basis-Software. Der Dateimanager kann zunächst wieder geschlossen werden (*X ganz rechts oben anklicken*).

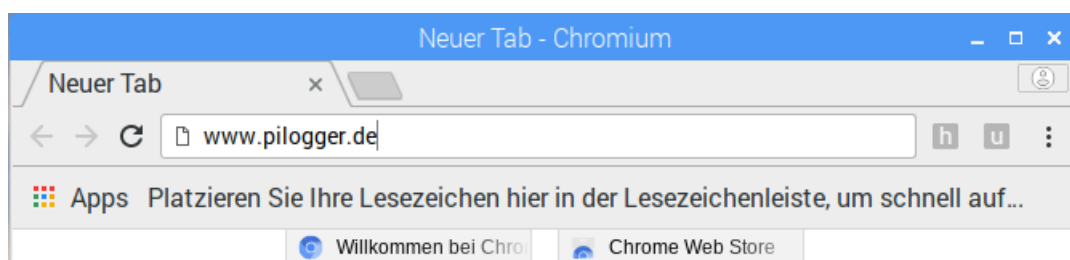
4.2.2 Archiv herunterladen

Das Archiv mit den benötigten Dateien heißt 'PiLogger_1v0_DE.zip' (oder eine entsprechend aktuellere Version) und wird an der gleichen Stelle bereitgestellt an der auch dieses Handbuch zu finden war: www.pillogger.de → 'Download' → 'Software'.

Da Raspbian einen Webbrowser an Bord hat, können wir den Download bei bestehender Internet-Verbindung auch direkt vornehmen. Dazu rufen wir den 'Web Browser' mit einem Maus-Klick auf das Weltkugel-Symbol auf:



Es öffnet sich ein Fenster mit dem Programm 'Chromium' - dem Open-Source Ableger des Google-Browsers 'Chrome'. In der Adressleiste geben wir 'www.pillogger.de' ein:

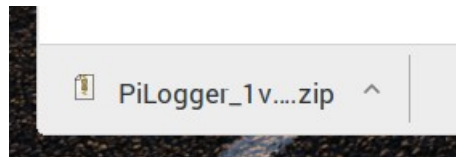


Auf der PiLogger Homepage gehen wir auf 'Download', dort in die Kategorie 'Software' und finden dort das Archiv:



The screenshot shows a software download interface. At the top, there is a header bar with a ZIP icon, the text 'Basis Software PiLogger DE 1v0', a 'Neu' (New) badge, a star rating of 5 stars with '(2 Stimmen)' (2 votes), and a large orange star icon. Below this, the text reads 'Basis-Software für PiLogger One (deutsch). Beispiel- und Test-Software in Python 3.' The bottom section contains metadata: 'Apache License, Version 2.0' with a document icon, a calendar icon showing '31' and the date '02.10.2017', a flag icon for 'Deutsch', a box icon for '16.88 KB', a download icon for '0', and an orange 'Download' button.

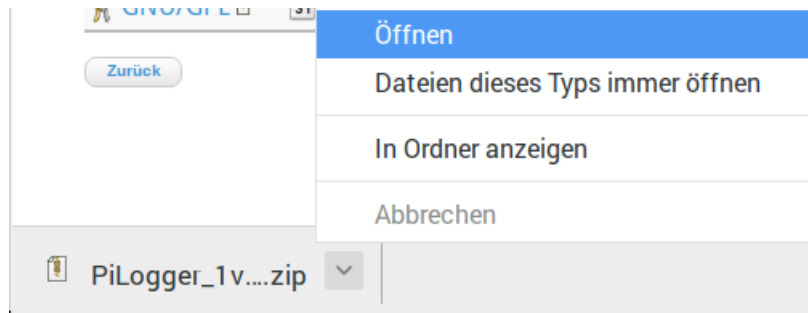
Nach dem Klick auf den Button 'Download' startet unauffällig der Download im Hintergrund und meldet sich am Ende in der linken unteren Ecke:



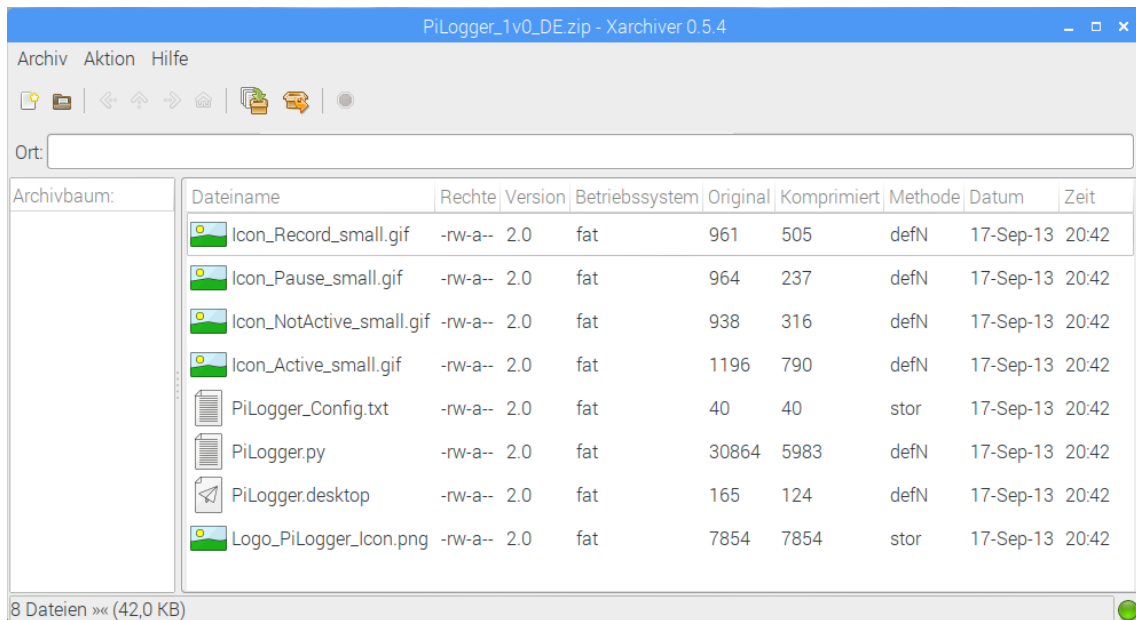
Das Archiv ist also erfolgreich heruntergeladen worden. Als Nächstes folgt das Entpacken.

4.2.3 Archiv entpacken

Direkt in dieser Melde-Box des Web-Browsers klicken wir nun auf den kleinen Pfeil, der nach oben zeigt, um eine Liste mit Aktionen anzeigen zu lassen:

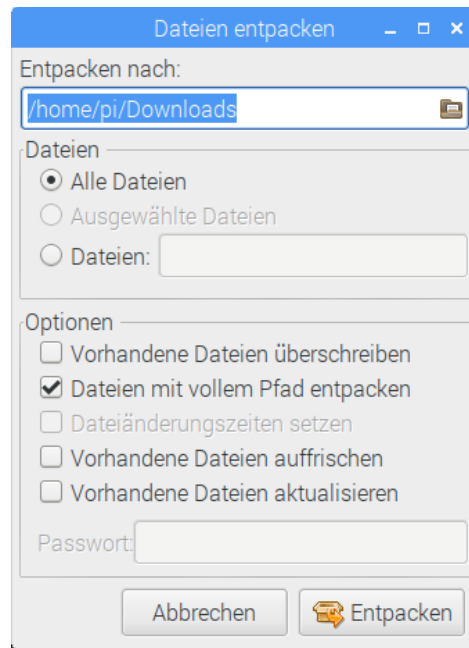


Wir wählen die Option 'Öffnen' durch klicken aus. Dadurch wird diese Datei direkt in dem Archiv-Tool 'Xarchiver' angezeigt, wodurch wir sofort den Inhalt dieses Archivs sehen:

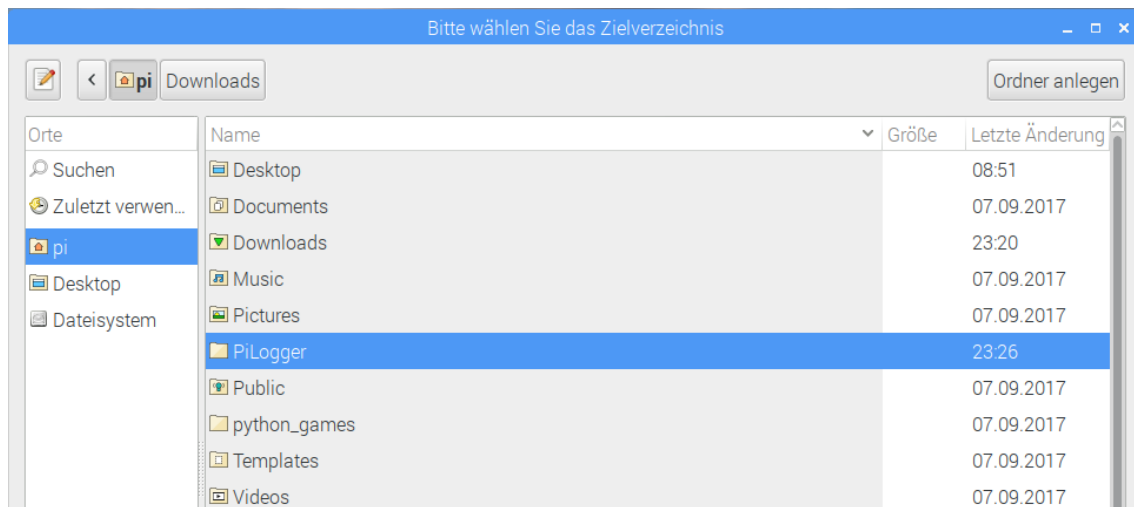


Die Datei 'PiLogger.py' ist dabei das eigentliche Python-Programm. Die anderen Dateien sind zusätzliche Hilfsdateien (Ressourcen), zu denen wir später noch kommen.

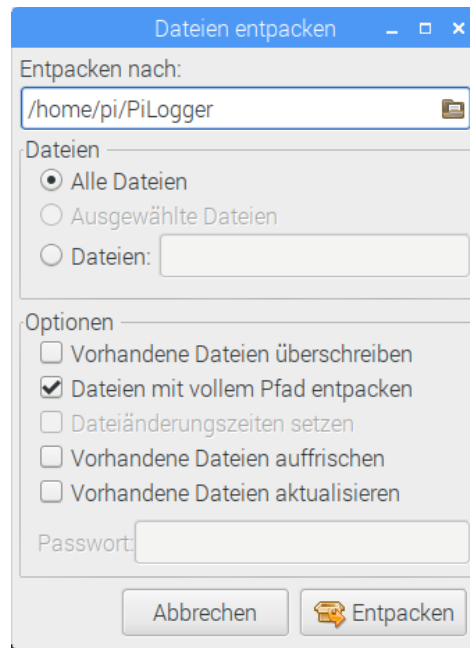
Um alle Dateien in das vorbereitete Verzeichnis zu entpacken drücken wir in der Icon-Leiste auf das letzte aktive Symbol. Es öffnet sich das folgende Dialog-Fenster:



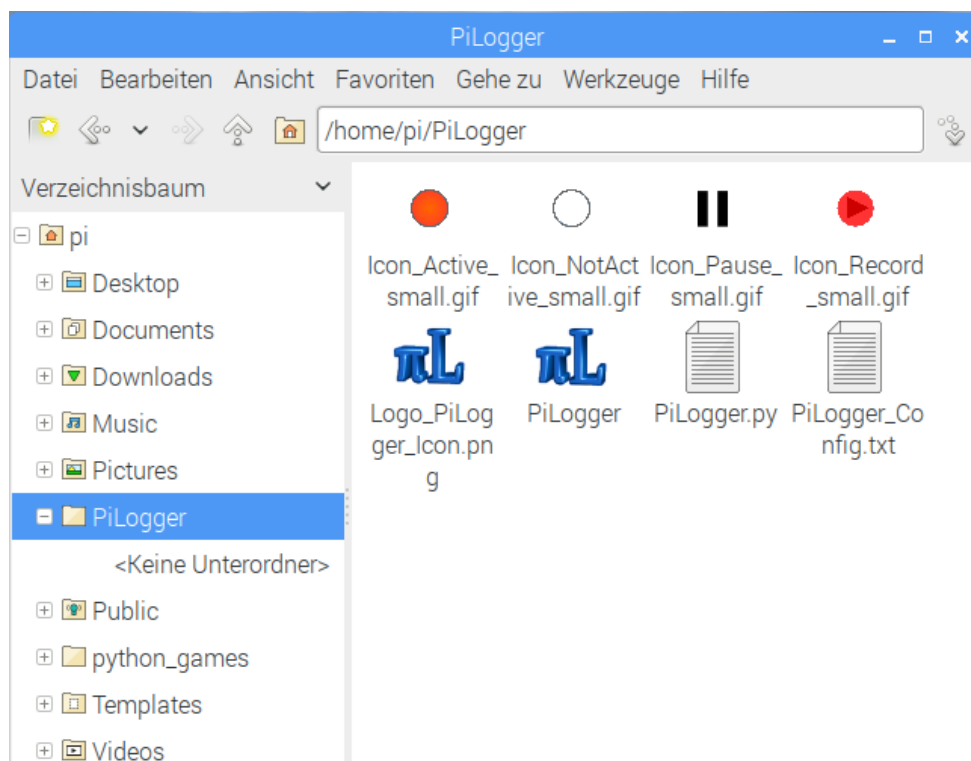
Die Option 'Alle Dateien' im Abschnitt 'Dateien' muss dabei aktiviert sein, insbesondere wenn vorher eine der Dateien markiert war. Anschließend klicken wir auf das Ordner-Symbol rechts im Feld 'Entpacken nach:', da hier noch der Ursprungsordner eingetragen ist. Es öffnet sich dieser Dialog:



Hier navigieren wir zum Ort 'pi' und darin zu unserem Zielverzeichnis 'PiLogger'. Nachdem wir ihn mit einem Links-Klick markiert haben, können wir auf 'Öffnen' klicken. Damit wird dieses Fenster geschlossen und der Pfad im vorherigen Dialog übernommen.



Nun klicken wir unten rechts auf den Knopf 'Entpacken'. Das Dialogfenster wird nun geschlossen und das Entpacken ausgeführt. Das geht sehr schnell und zeigt deshalb keine weitere Meldung. Wir schließen den 'Xarchiver' und öffnen den Datei-Manager. Hier navigieren wir wieder zu unserem Ordner 'PiLogger':

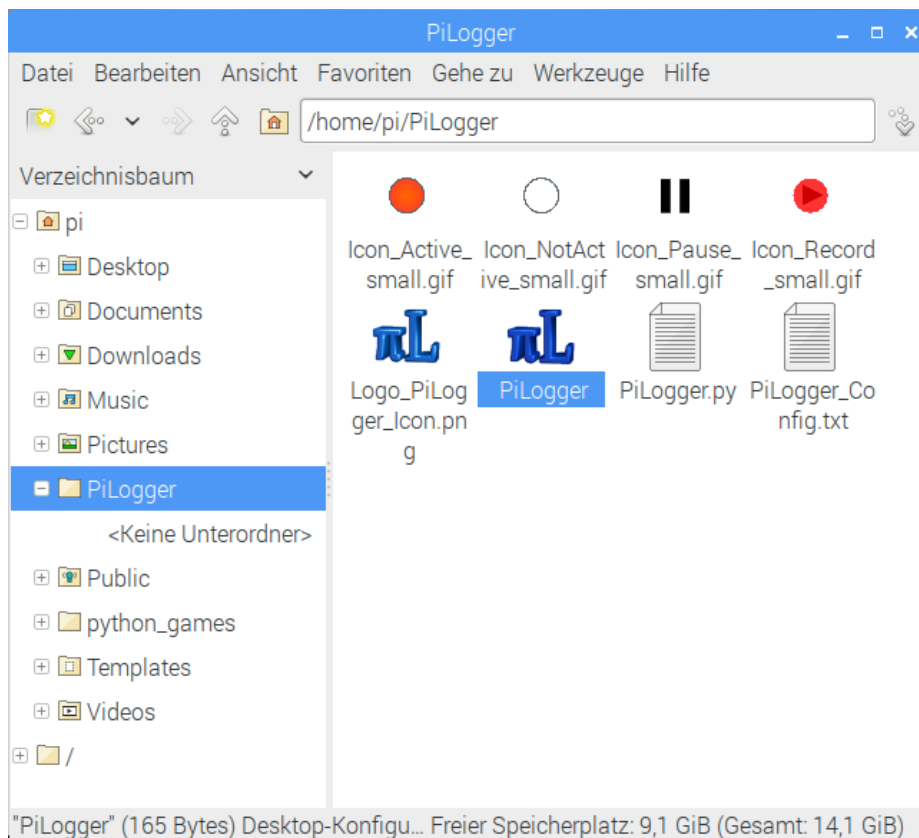


Die ausgepackten Dateien sind hier nun mit der Ansichts-Option 'Symbolansicht' zu sehen.

4.2.4 Desktop-Starter einrichten

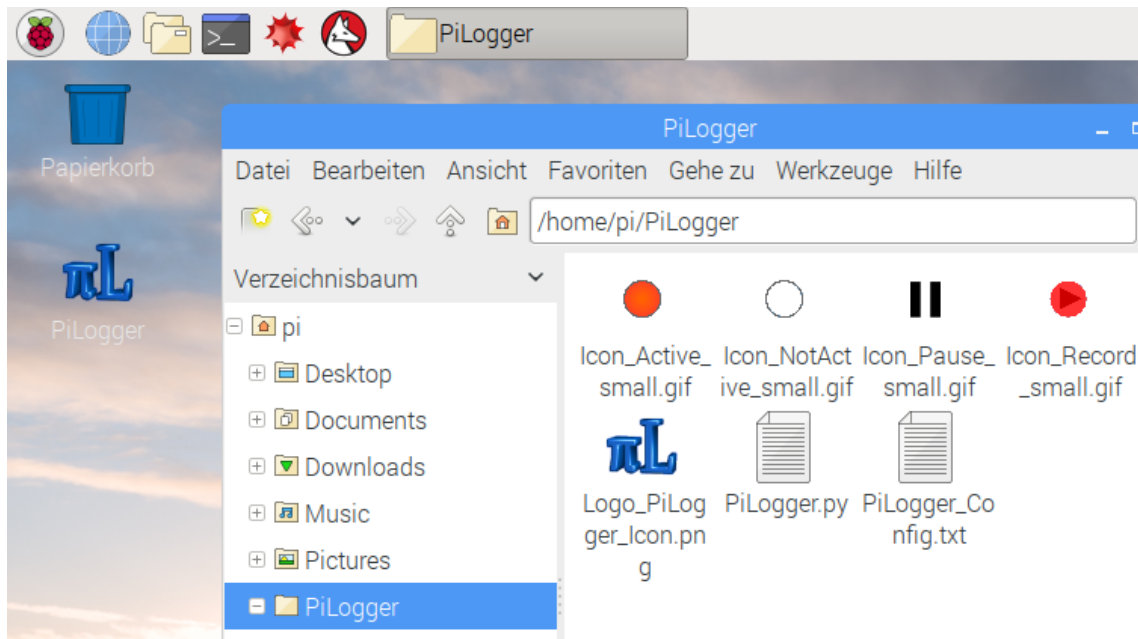
Eine der zusätzlichen Dateien hat die Endung '.desktop'. Diese Endung wird gar nicht angezeigt, weil es eine Datei ist, die einem festgelegten Zweck im System dient – eine Systemdatei. In der Xarchiver-Ansicht war die Endung noch zu sehen, im Dateimanager wird sie unterdrückt und es wird das Systemdatei-Icon benutzt oder wie in unserem Fall, das in ihr angegebene Icon – hier das PiLogger-Icon.

Diese Datei dient dazu auf dem Desktop ein Icon zu platzieren, mit dem man direkt ein bestimmtes Programm starten kann. Wenn wir die Datei im Ordner PiLogger mit einem Links-Klick markieren, wird in der Statuszeile des Dateimanagers angezeigt, dass es sich um eine Desktop-Konfigurations-Datei handelt:



Wir klicken die selektierte Datei nun nochmal mit der linken Maustaste an und halten die Taste dabei gedrückt – nun können wir das Icon am Mauszeiger hängend bewegen und einfach in den freien Bereich des Desktops unterhalb des Papierkorb-Symbols ziehen. Dort lassen wir dann die Maus-Taste los (Drag and Drop). Damit wird die Datei aus dem Ordner 'PiLogger' in den Ordner 'Desktop' verschoben.

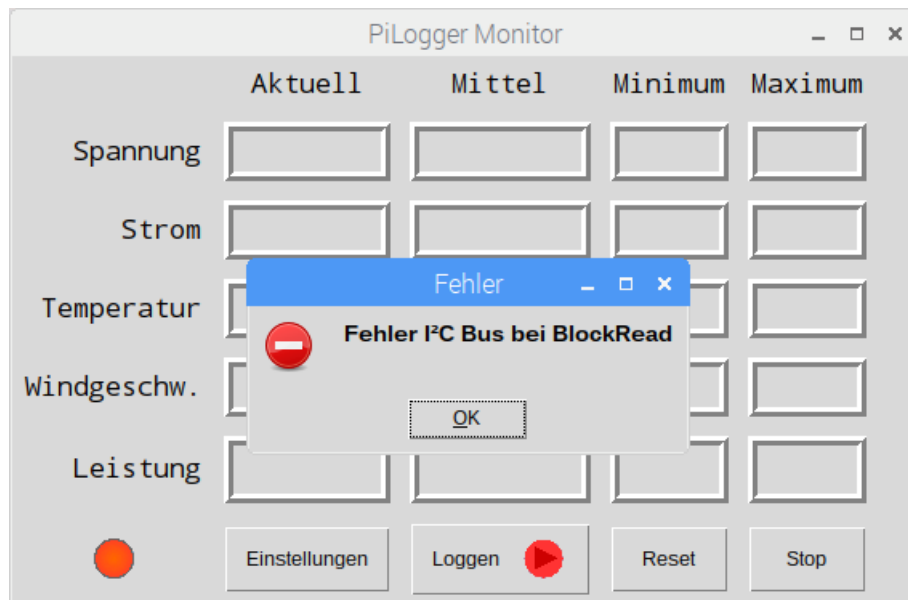
Danach sieht das dann so aus:



Jetzt können wir den Dateimanager wieder schließen.
Die PiLogger Basis-Software ist eingerichtet.

4.2.5 Probelauf (Ohne Hardware)

Um zu überprüfen, ob bisher alles gut geklappt hat und um einen ersten Eindruck zu bekommen, probieren wir den Desktop-Starter nun mit einem Doppelklick aus. Zunächst sollte der Mauszeiger durch eine Sanduhr Aktivität anzeigen und anschließend sollten in der Mitte des Bildschirmes diese zwei Fenster erscheinen:

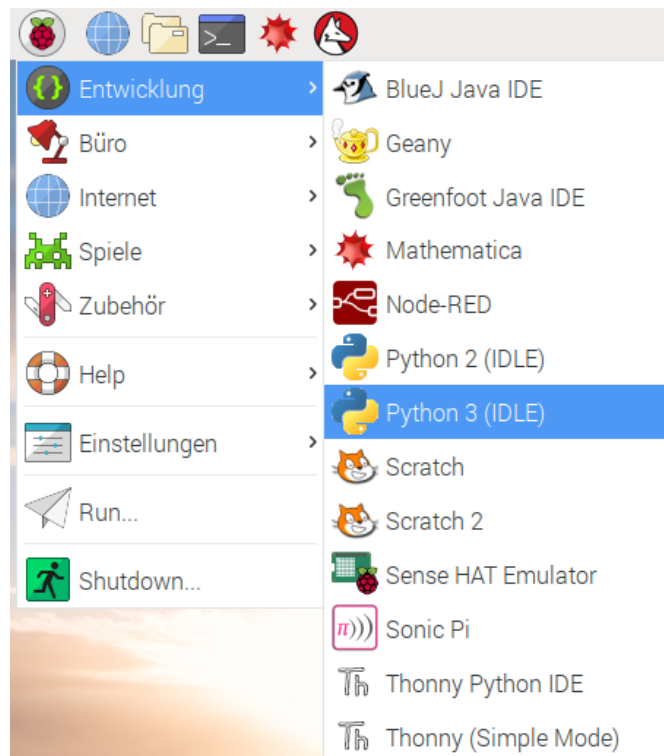


Das erste und größere Fenster heißt 'PiLogger Monitor' und ist das Hauptfenster der Basis-Software. Das zweite darüberliegende Fenster ist eine vom Programm erzeugte Fehlermeldung, die mitteilt, dass es auf dem I²C Bus einen Fehler gegeben hat. Das ist nicht verwunderlich, da wir bisher den PiLogger noch gar nicht angeschlossen haben.

Das ist also ok und so drücken wird dann auch auf 'OK' (quittieren).
Damit werden beide Fenster geschlossen, weil das Programm beendet wird.
Es hat also bis hier alles prima geklappt.

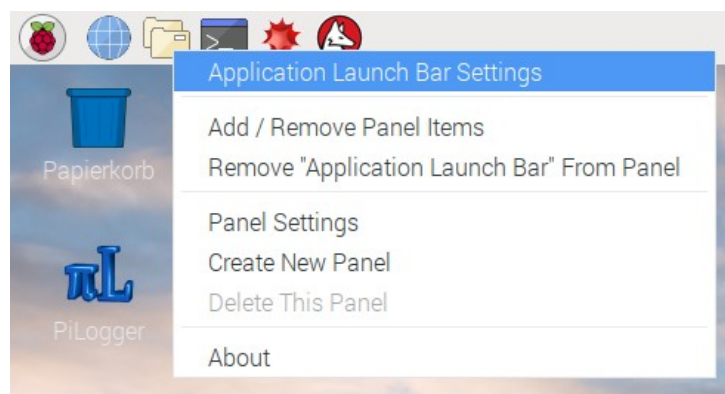
4.2.6 Schnellstarter für Python

Um die Basis-Software anzusehen und ändern zu können, benötigen wir eine sogenannte integrierte Entwicklungsumgebung für Python. Dies kann entweder die klassische 'IDLE für Python 3' sein, oder seit Raspbian Stretch auch 'Thonny'. Diese werden normalerweise über das Startmenü aufgerufen:

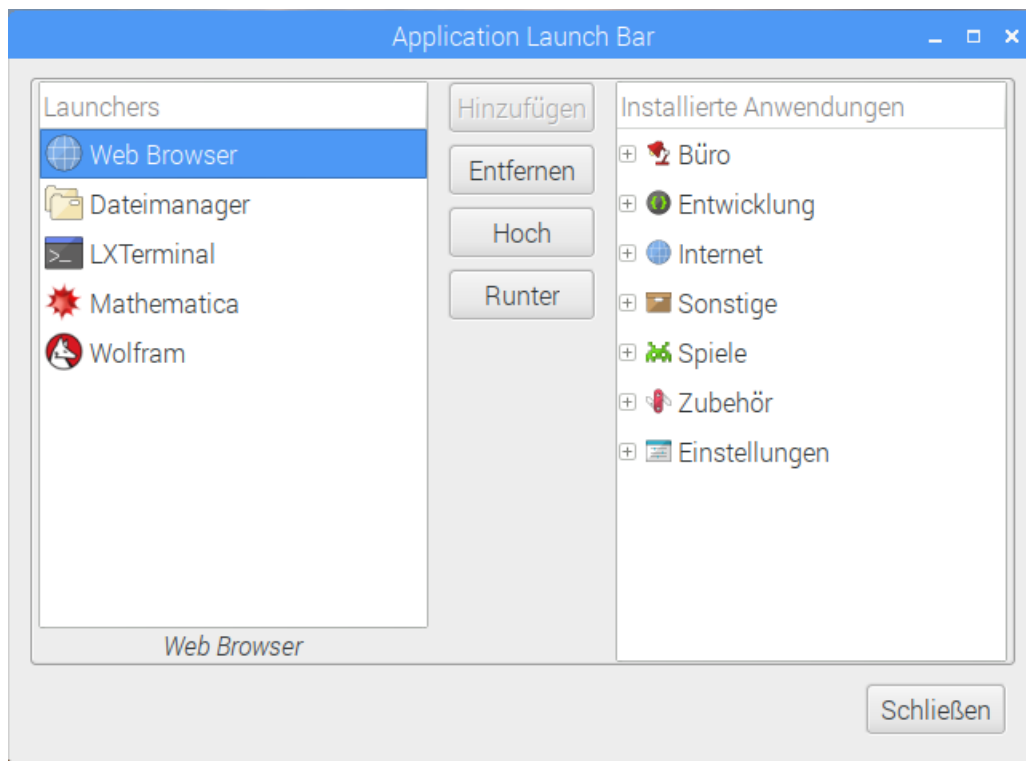


Also mit: 'Menu', 'Entwicklung' → 'Python 3 (IDLE)'.

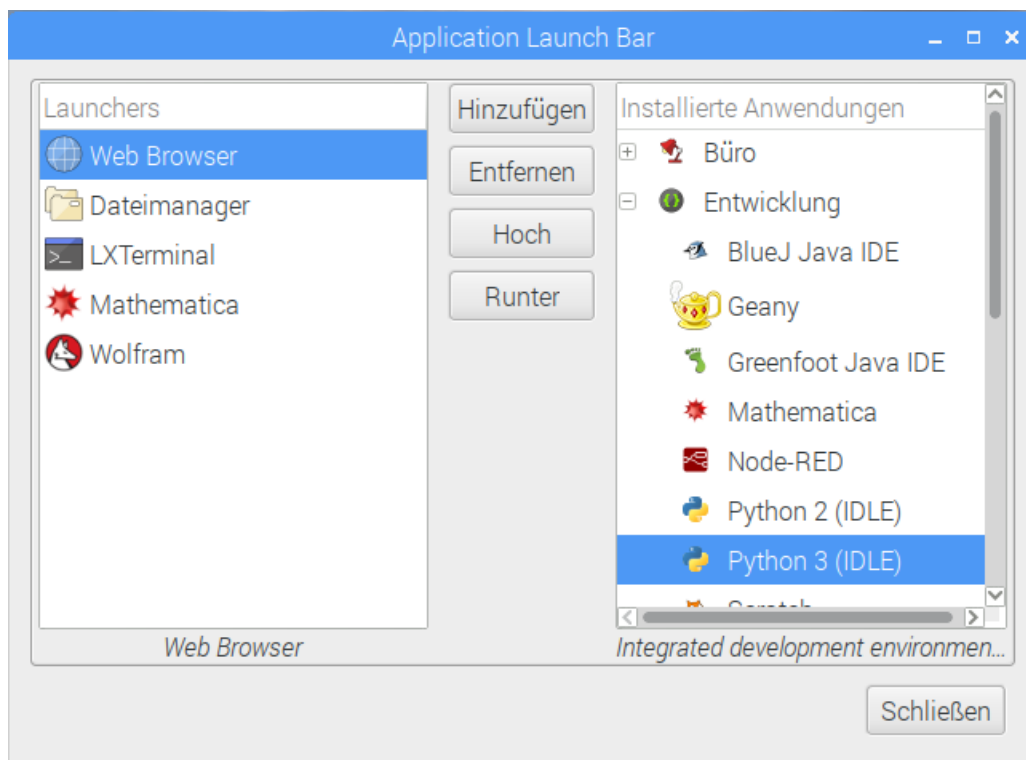
Das lässt sich komfortabler machen, indem wir in der Task-Leiste einen Schnellstarter dafür hinzufügen, so wie die für den Browser, den Dateimanager oder das Terminal. Dazu machen wir auf eines der schon vorhandenen Schnellstart-Symbole in der Task-Leiste einen Rechts-Klick (z.B. Terminal) und öffnen das Kontext-Menü:



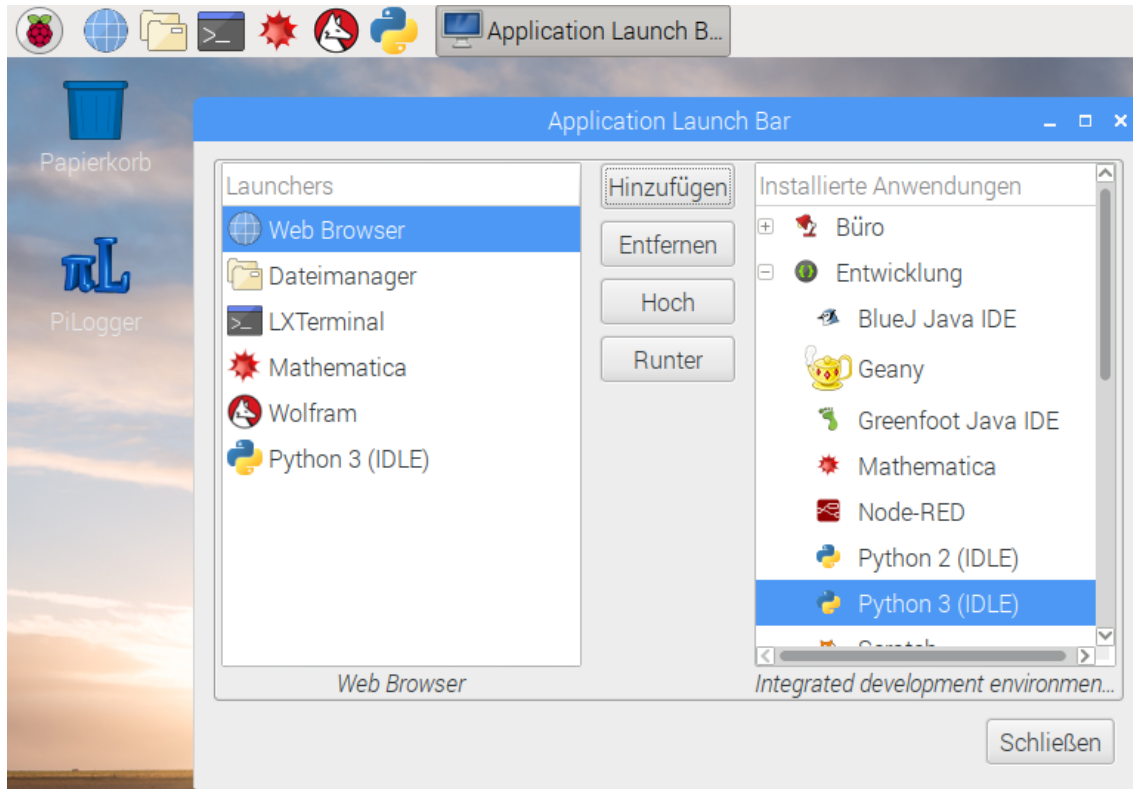
Hier navigieren wir auf den ersten Eintrag 'Application Launch Bar Settings' und klicken darauf. Damit öffnet sich dieses Fenster:



Hier sehen wir nun die bereits definierten Schnellstart-Symbole in der linken Spalte und die verfügbaren weiteren in einer Baumstruktur in der rechten Spalte. In der rechten Spalte klicken wir bei 'Entwicklung' auf das kleine '+' davor:



In der Unterauswahl markieren wir den Eintrag 'Python 3 (IDLE)' und klicken anschließend in der Mitte auf 'Hinzufügen':



Damit taucht dann dieses neue Schnellstart-Symbol einerseits in der linken Spalte auf, als auch in der Task-Leiste. Wir schließen dieses Fenster jetzt mit einem Klick auf 'Schließen'. Jetzt lässt sich die Python Entwicklungsumgebung mit einem einfachen Maus-Klick starten. Einfach mal ausprobieren !

5 Montage und Inbetriebnahme

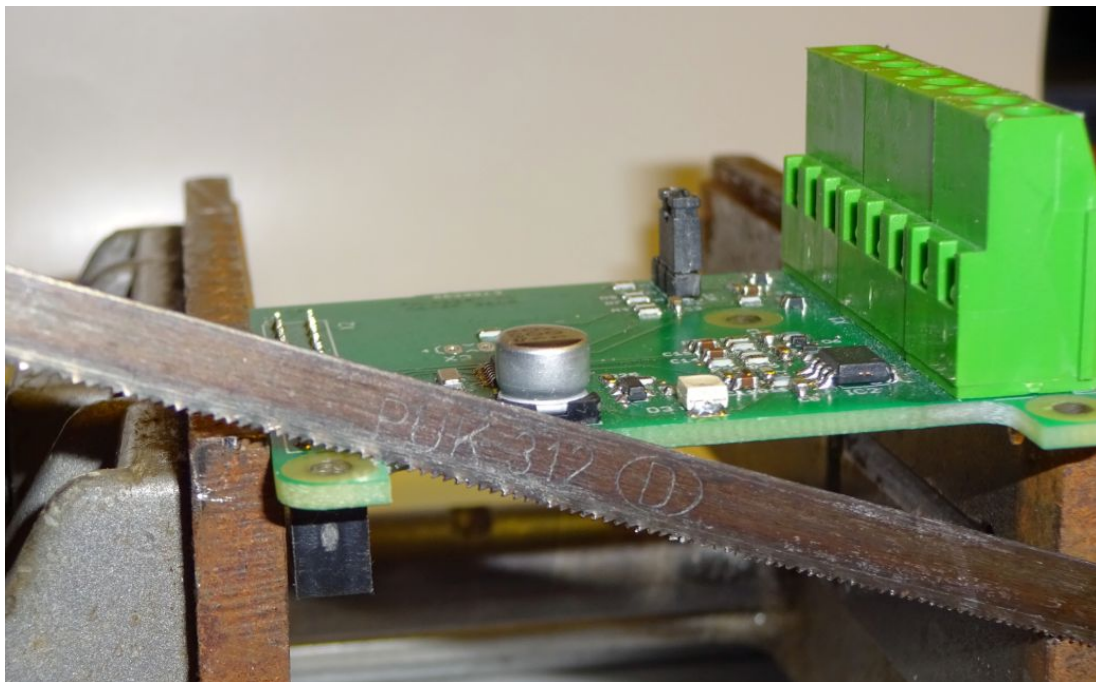
5.1 Vorbereitung

Das PiLogger One 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 Anschraubblöchern sind hier ohne Verwendung.

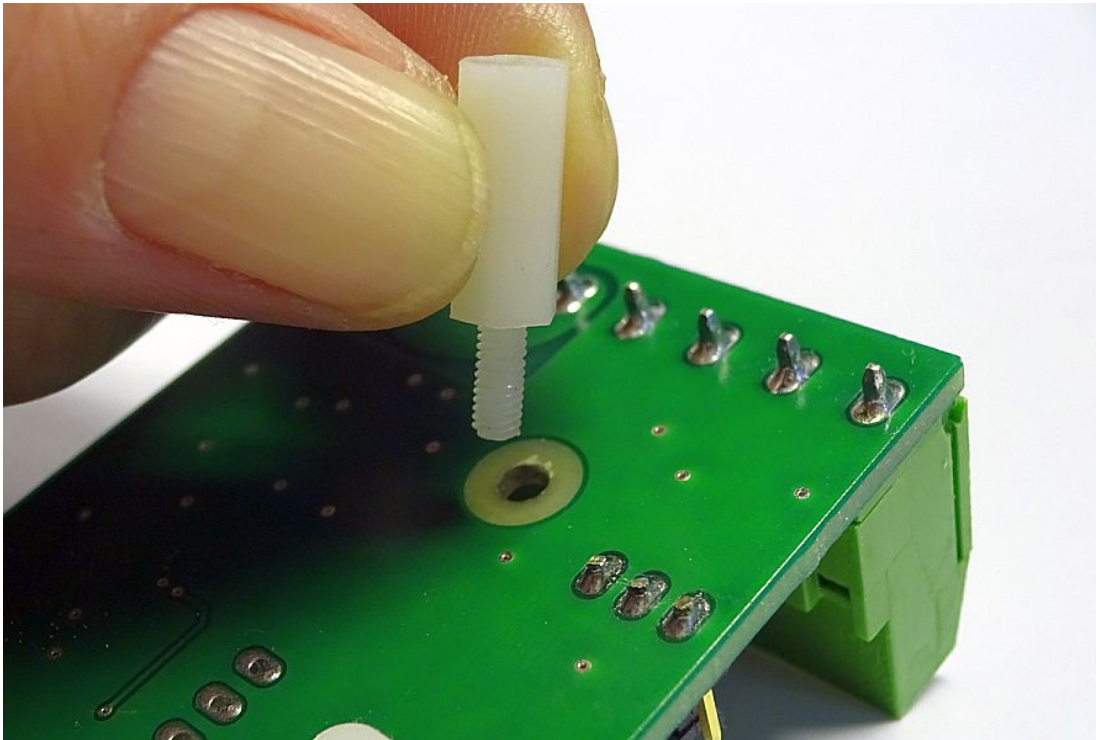
Bei Bedarf, zum Beispiel beim Einbau in ein enges Gehäuse, können diese Ecken vorsichtig entfernt werden. Das geht etwa mit einer Modellbau-Bügelsäge, wenn das Modul sicher und ohne großen Druck – schon gar nicht Verbiegung ! - eingespannt ist. Dabei dürfen weder die Steckleiste noch irgendwelche anderen Bauteile beschädigt werden !

Das sieht etwa so aus:



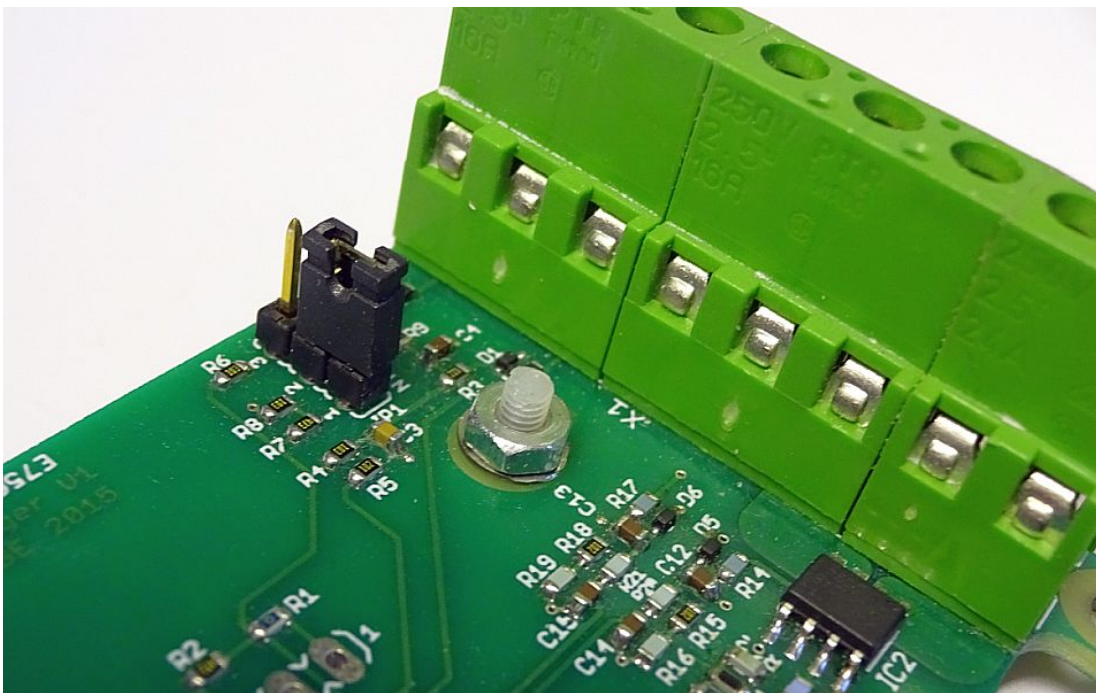
Für Raspberrys mit 40 poliger Stiftleiste sollten alle 3 Befestigungslöcher verwendet werden. Dabei gibt es für den mittleren Abstandshalter auf dem Raspberry Pi zwar kein passendes Loch, jedoch ist an dieser Stelle kein Bauteil auf der Oberseite, dass beschädigt werden könnte. So kann sich dort der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

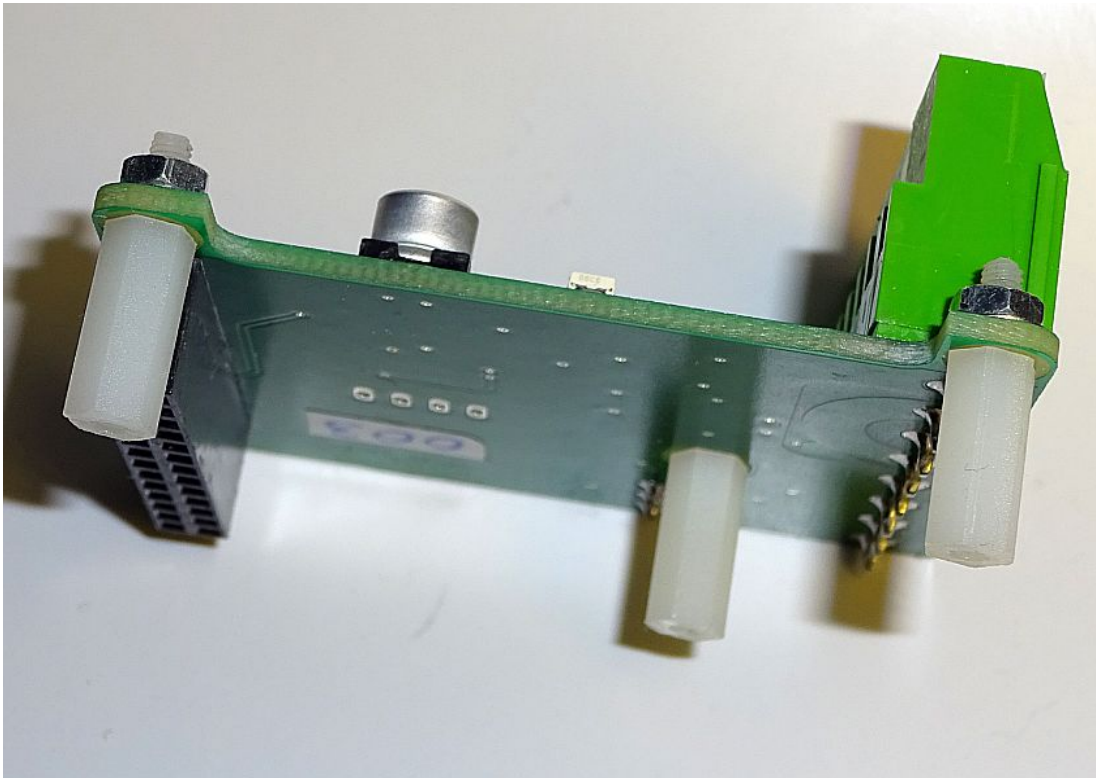
In diese Löcher des PiLogger One werden nun die beigelegten Abstandsbolzen eingesetzt. Dabei wird jeweils das Gewindeende direkt von unten in das Loch im PiLogger-Modul gesteckt:



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:





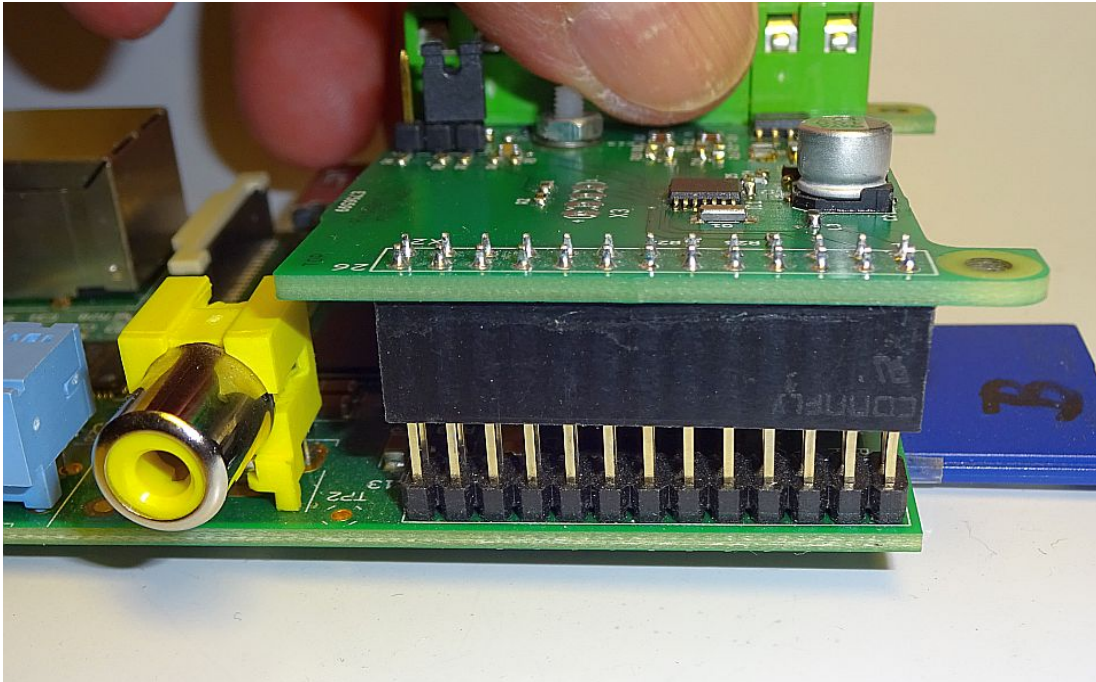
Nun kann das Modul aufgesteckt werden.

5.2 Modul aufsetzen

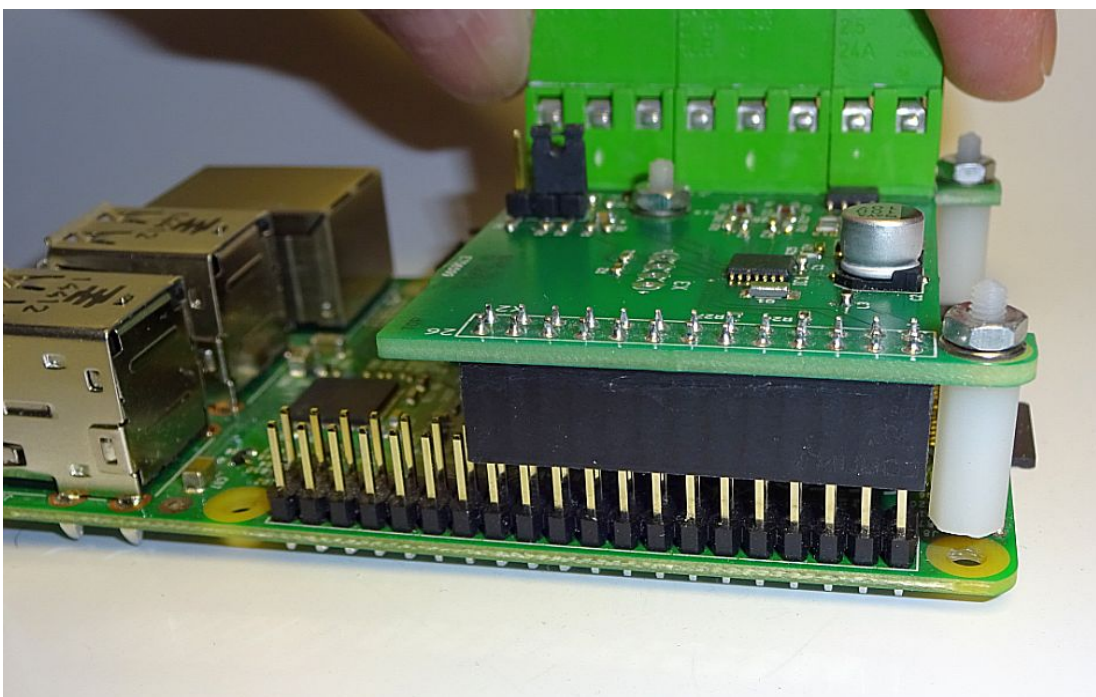
Das PiLogger Modul wird nun auf den Raspberry Pi aufgesetzt. Dazu muss mindestens die Versorgung ausgesteckt sein – besser ist es alle Kabel und angesteckte Peripherie hierfür zu entfernen.

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:

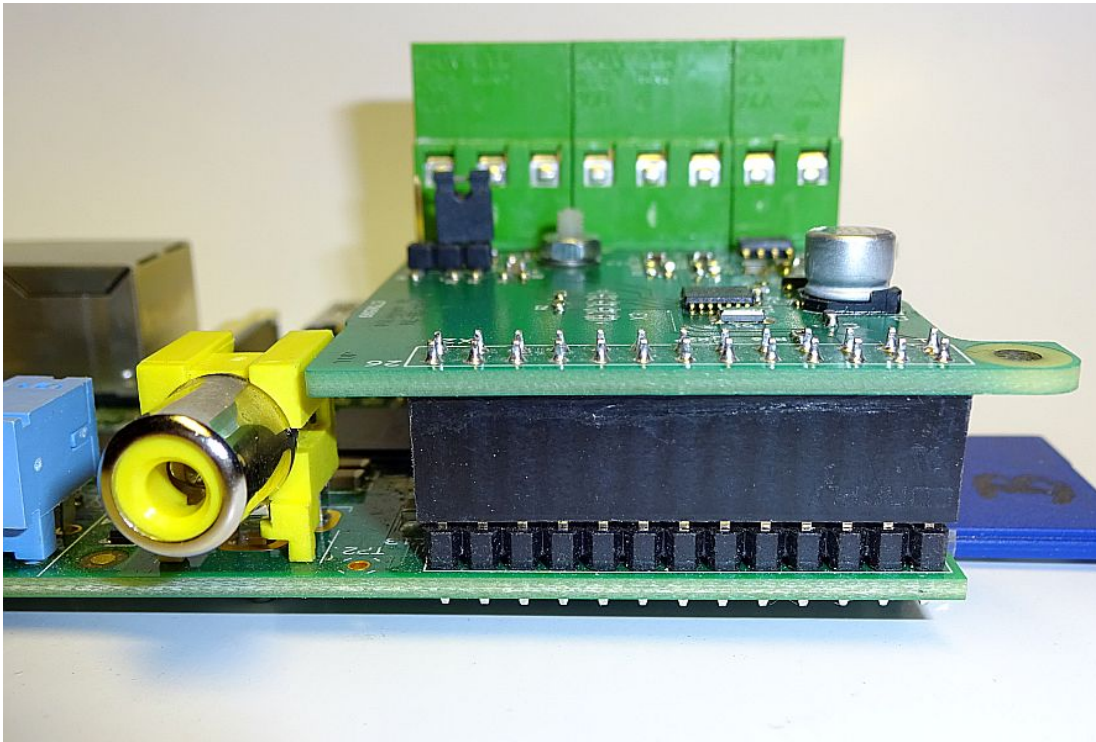


Und hier für einen Raspberry Pi 2 B:

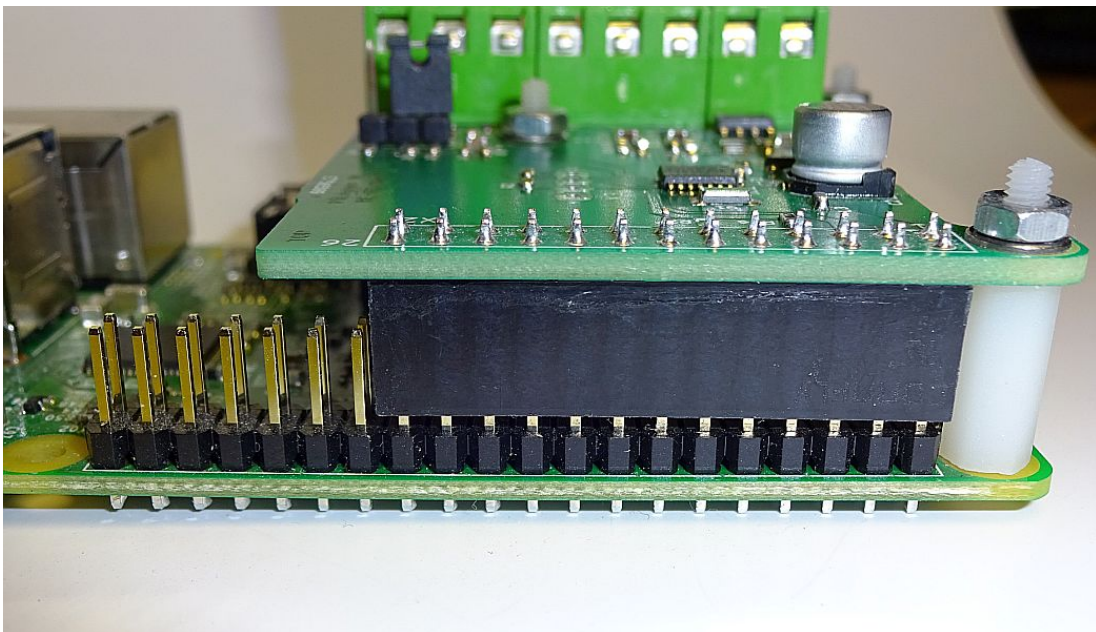


Jetzt muss die PiLogger Leiterplatte vorsichtig gleichmäßig heruntergedrückt werden, bis die Abstandsbolzen auf dem Raspberry Pi aufsetzen.

Das sieht dann beim Raspberry Pi 1 so aus:

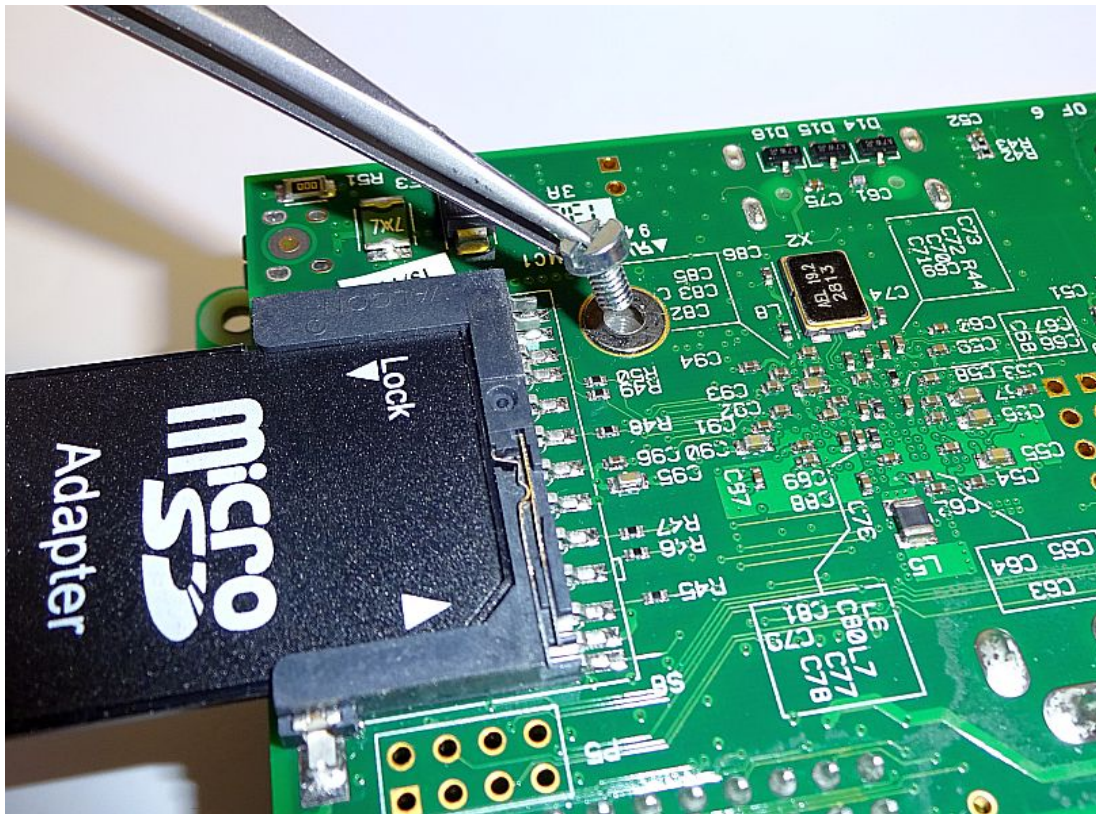


Und beim Raspberry Pi 2 B sieht es so aus:

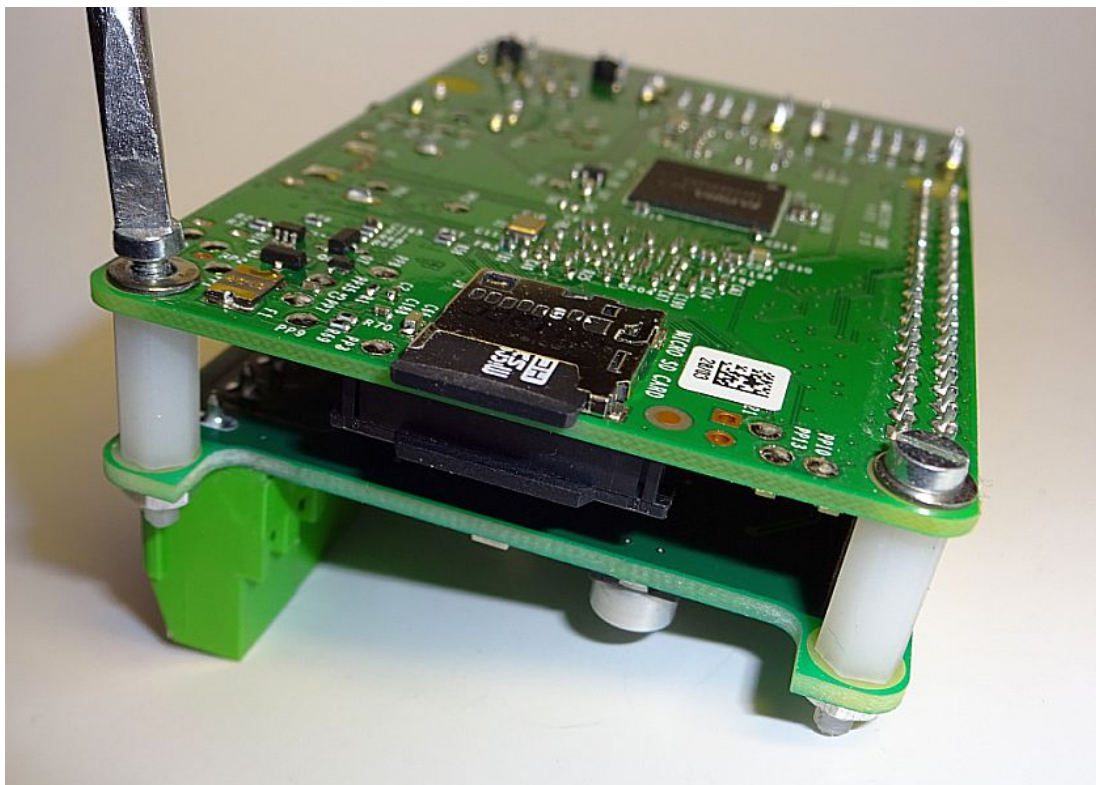


Das heißt: Die Buchsenleiste sitzt nicht ganz auf dem Sockel der Stiftleiste auf. Die Kontakte sind jedoch gut im Eingriff.

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.

5.3 Erstes Einschalten

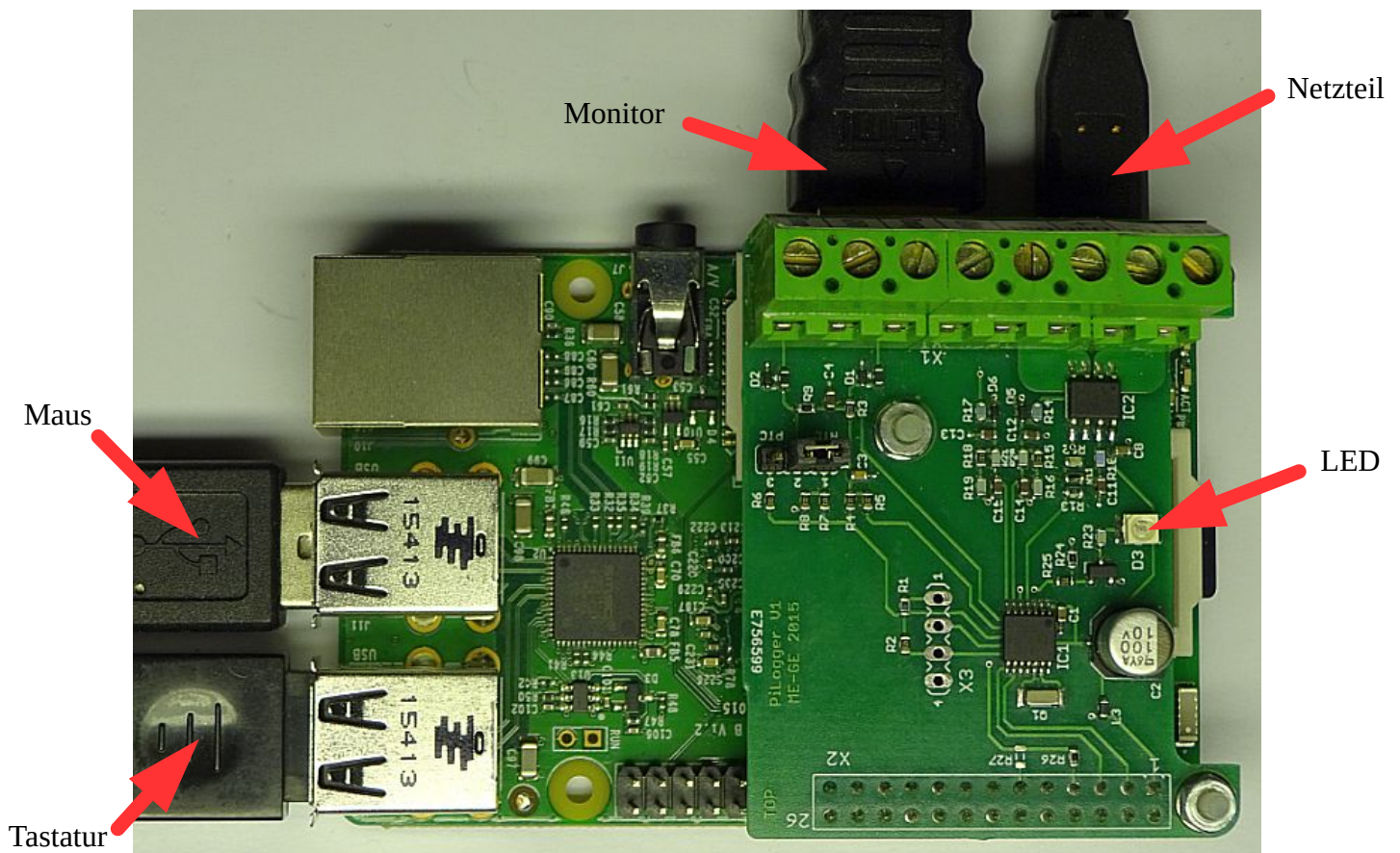
Vor dem ersten Einschalten des Raspberry Pi mit PiLogger müssen natürlich alle Kabel und Peripherie-Teile wie in 3.4.2 wieder eingesteckt werden.

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 Blitzten 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.

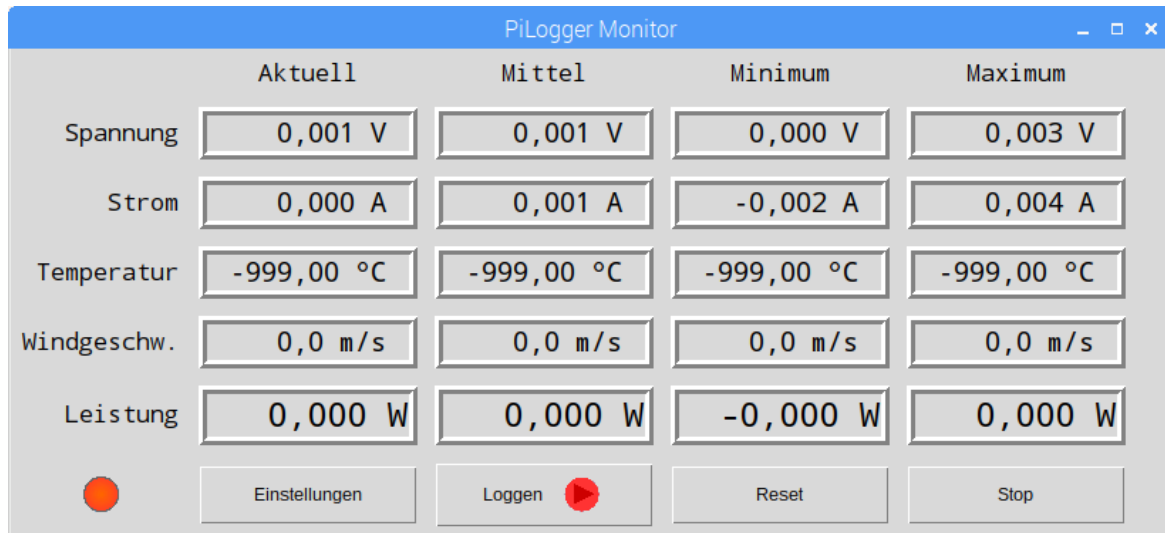
Hier als Beispiel der PiLogger auf einem Raspberry Pi 3 :



6 Bedienungsanleitung Basis-Software

6.1 Monitor-Fenster

Nachdem der Raspberry Pi vollständig gebootet hat, starten wir die PiLogger Basis-Software durch einen Doppelklick auf das PiLogger Desktop-Symbol. Es öffnet sich das Fenster 'PiLogger Monitor':



In diesem Fenster werden live alle Messwerte angezeigt.

Da zur Zeit an den PiLogger nichts weiter angeschlossen ist, sehen wir bei der Spannung fast eine Null, beim Strom fast eine Null und also bei der Leistung auch eine Null. Ohne entsprechenden Geber (Sensor) bleibt auch der Messwert für Windgeschwindigkeit bei Null stehen. Die Temperatur wird mit einem klar ungültigen Wert angezeigt, weil der Meßwert ohne den entsprechenden Temperaturfühler in einem Bereich kleben bleibt, der keinen Sinn macht.

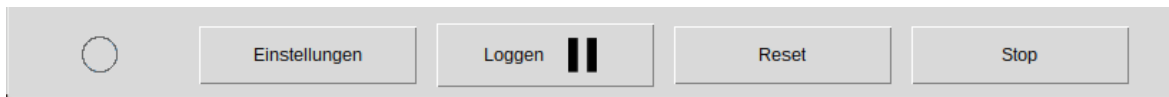
In der linken unteren Ecke ist ein rundes Indikator-Feld, das zum Zeitpunkt des Bildschirmfotos (screenshot) aktiv war und anzeigt, das die momentan angezeigten Werte gerade in die Log-Datei gespeichert werden. Mit den Vorgabeeinstellungen erfolgt die Aktualisierung der Messwertanzeige einmal pro Sekunde und das Abspeichern in die Messwertdatei alle 10 Sekunden – das lässt sich selbstverständlich alles konfigurieren.

■ Knopf 'Einstellungen'

Rechts neben dem Log-Indikator folgt ein Knopf (Button) 'Einstellungen', mit dem eben diese Konfigurationsmöglichkeiten aufgerufen werden. Dazu im nächsten Abschnitt 6.2 mehr.

■ Knopf 'Loggen'

Weiter rechts folgt der Knopf 'Loggen', der eine integrierte Zustandsanzeige besitzt. Das heißt, das Symbol im Knopf ändert sich mit dem aktuellen Zustand. Ein Druck auf diesen Knopf pausiert die Abspeicher-Aktivitäten, bei einem weiteren Klick werden sie wieder aufgenommen. Der pausierte Zustand sieht so aus:



■ Knopf 'Reset'

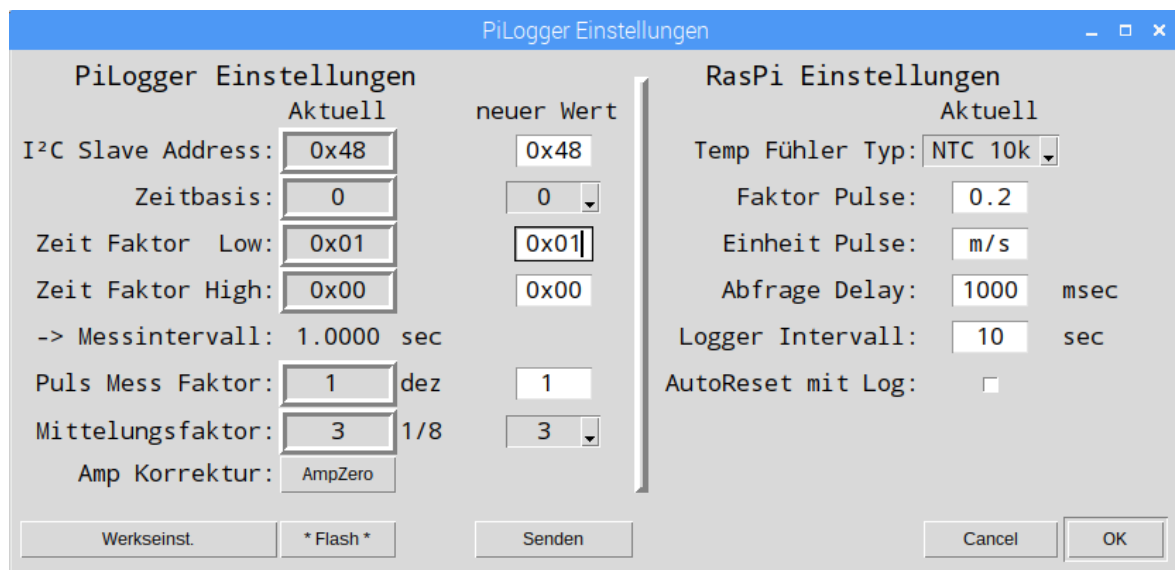
Der nächste Knopf 'Reset' setzt die Statistik-Werte 'Mittel', 'Minimum' und 'Maximum' zurück auf die gerade aktuellen Werte. Dieses Zurücksetzen findet dabei im PiLogger selber statt, der diese Werte intern ermittelt. Deshalb erfolgt die Reaktion in der Anzeige etwas verzögert nach dem nächsten Auslesen der Werte vom PiLogger.

■ Knopf 'Stop'

Der letzte Knopf 'Stop' ist der eigentliche Programm-Beenden-Knopf, damit wird das Programm beendet und das Fenster geschlossen – der PiLogger arbeitet autark weiter.

6.2 Konfigurationsfenster

Mit dem Knopf 'Einstellungen' wird ein weiteres Fenster 'PiLogger Einstellungen' aufgerufen, welches sich über das Monitor-Fenster legt:



Das Fenster ist grundsätzlich in zwei Bereiche geteilt. Die linke Seite bezieht sich auf Einstellungen die am PiLogger vorgenommen werden. Die rechte Seite bezieht sich auf Einstellungen die am Programm selber vorgenommen werden, also auf der Raspberry Pi Seite.

6.2.1 PiLogger Einstellungen

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

In der rechten Spalte 'neuer Wert' können für die einzelnen Parameter neue Werte eingegeben werden. Diese sind im Einzelnen:

● I²C Slave Address

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

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



Achtung :

Wird diese Adresse geändert, kann sofort die PiLogger Basis-Software den PiLogger nicht mehr ansprechen !

Es muss dann manuell die Konfigurationsdatei 'PiLogger_Config.txt' auf die neue Adresse angepasst werden !

Die neue Adresse wird sofort temporär übernommen, wenn der Knopf 'Senden' betätigt wird. Die permanente Übernahme kann nur erfolgen, wenn der Knopf 'Flashen' nach einem Neustart der Basis-Software mit geänderter Konfigurationsdatei betätigt wird.

● 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.

Als kleinstes Messintervall sollten 99 msec nicht unterschritten werden, da eine Messung etwa 84 msec dauert, also $51 \times 1,953125 \text{ msec} = 99,609375 \text{ msec}$. Als größtes Zeitintervall können 65535 sec eingestellt werden, was 18 Stunden, 12 Minuten und 15 Sekunden entspricht.

Die Basis-Software berechnet aus den aktuell eingestellten Werten jedesmal 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. 84 msec an).

● Zeit Faktor Low

Niederwertiges Byte des Faktors für die Messintervall-Einstellung.

● Zeit Faktor High

Höherwertiges Byte des Faktors für die Messintervall-Einstellung.

Der Gesamtfaktor berechnet sich zu $H \times 256 + L$. Mindestwert ist 1.

● Puls Mess Faktor

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 $\frac{1}{4}$ 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.

● Mittelungsfaktor

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 $\frac{1}{2}$ und $1/128$ einstellen. Der aktuelle Messwert wird mit diesem Faktor gewichtet und zum bisherigen Mittelwert addiert.

■ Knopf 'AmpZero'

Der Knopf in der Zeile 'Amp Korrektur' mit der Aufschrift 'AmpZero' 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 herauskalibriert. Vor Anwendung sollte der PiLogger etwa 10 Minuten aktiv gewesen sein, damit er thermisch eingeschwungen ist.

Diese Kalibration ist einmal bei Erstinbetriebnahme sinnvoll und gegebenenfalls wenn nach einiger Zeit im Betrieb eine deutliche Missweisung festgestellt wird.

■ Knopf 'Werkseinst.'

Für den Fall, dass, zum Beispiel durch extreme Messintervalle, der Zustand unklar ist, kann mit diesem Knopf ein Befehl an den PiLogger gesendet werden, der die Standard-Einstellungen wiederherstellt. Dieser Befehl kann natürlich nur gesendet werden, wenn die I²C Verbindung noch besteht.

■ Knopf 'Senden'

Die neu eingegebenen Werte werden durch einen Klick auf den Knopf 'Senden' an den PiLogger geschickt und sind sofort wirksam. Das kann man schön an der Aktivität im dahinter liegenden Monitor-Fenster beobachten. Und natürlich am Takt der PiLogger LED.

■ Knopf 'Flashen'

Sollen geänderte Einstellungswerte dauerhaft übernommen werden, so dass nach einem Strom-Ausschalten und wieder Einschalten (power cycle) der PiLogger auch wieder mit dieser Einstellung arbeitet, müssen diese Werte im internen Flash-Speicher des PiLoggers

abgelegt werden. Dazu wird einmal der Knopf 'Flashen' angeklickt.



Achtung: Für das Flashen gibt es keine direkte Rückmeldung !
Den Erfolg kontrolliert man durch einen Neustart nach 'Strom aus'.

6.2.2 RasPi Einstellungen

Die rechte Seite des Fensters enthält Einstellungen für das Python-Programm selber, ist also die Raspberry Pi Seite der Einstellungen.

Diese Einstellungen werden in einer separaten Datei 'PiLogger_Config.txt' 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.

Die RasPi Einstellungen im Einzelnen:

- **Auswahlliste 'Temp Fühler Typ'**

Dies ist eine Aufklappliste mit nur 2 Einträgen: 'NTC 10k' und 'PT1000'.

Je nach angeschlossenem Temperaturfühler wird hier die notwendige Umrechnung der Messwerte in Temperaturwerte festgelegt. Diese beiden Umrechnungsalgorithmen stehen in der Basis-Software zur Verfügung.



Achtung:

Die Eingangsbeschaltung muss dabei mit dem Jumper auf dem PiLogger auf den passenden Grundtyp NTC oder PTC eingestellt werden.

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

- **Faktor Pulse**

Hier wird der Umrechnungsfaktor für den Impulszähleingang eingestellt.

Dies ist die Fühlerkonstante. Also beispielsweise für einen Windgeber die Wegstrecke pro Impuls. Das Messzeitintervall wird zusätzlich von der PiLogger-Basis-Software mit berücksichtigt.

In dem gezeigten Beispiel also 0,2 m / Impuls – dabei geht natürlich noch die Einheit aus dem Feld danach mit ein.

- **Einheit Pulse**

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

Hier kann zum Beispiel statt der 'm/s' auch 'km/h' eingetragen werden, wobei dann aber natürlich die Fühlerkonstante entsprechend umgerechnet werden muss – im Beispiel also 0,72 für 'km/h' statt 0,2 bei 'm/s'.

- **Abfrage Delay**

Hier wird festgelegt wie oft die Basis-Software den PiLogger abfragen soll.

Dies betrifft hauptsächlich die Aktualisierung der Live-Anzeige, hängt aber zusammen mit dem nächsten Wert 'Logger Intervall'. Wenn also die Live-Anzeige des Monitor-Fensters nicht der Hauptzweck der Anwendung sein soll, kann hier auch ein ziemlich langer Zeitraum eingestellt werden, der dem gewünschten Logger-Intervall aus dem nächsten Feld entspricht.

- **Logger Intervall**

Hier wird der Zeitabstand zwischen 2 Eintragungen in die Log-Datei eingestellt.

Es ist sinnvoll hier ein ganzzahliges Vielfaches des Abfrage-Intervalls einzustellen – weil das Programm das so erwartet ;-)

Dieses Intervall ist also der zeitliche Messpunktabstand der mitgeschriebenen Messwerte. Die Messwerte werden mit Zeitstempel in der Datei 'logdata.csv' im Verzeichnis '/home/pi/PiLogger' gespeichert.

- **AutoReset mit Log**

Dieses Feld ist ein Ankreuzkästchen (checkbox). Ist es leer, so ist diese Funktion nicht aktiv. Ist in dem Kästchen ein Haken zu sehen, ist die Funktion aktiv.

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.

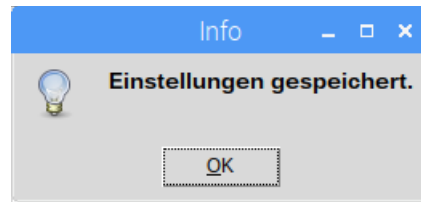
Damit werden diese Werte zu den Statistik-Werten die für genau ein Messintervall gelten. Das ist beispielsweise sinnvoll, wenn zusätzlich noch die Energiemenge gezählt werden soll, also Leistung mal Zeit (-Intervall).

- **Knopf 'Cancel'**

Der Knopf 'Cancel' schließt das Fenster 'PiLogger Einstellungen' ohne die aktuelle Konfiguration abzuspeichern.

■ Knopf 'OK'

Der Knopf 'OK' schließt das Fenster 'PiLogger Einstellungen' und speichert die aktuellen Einstellungen in der Datei 'PiLogger_Config.txt' im Verzeichnis '/home/pi/PiLogger'. Es erscheint diese Benachrichtigung:

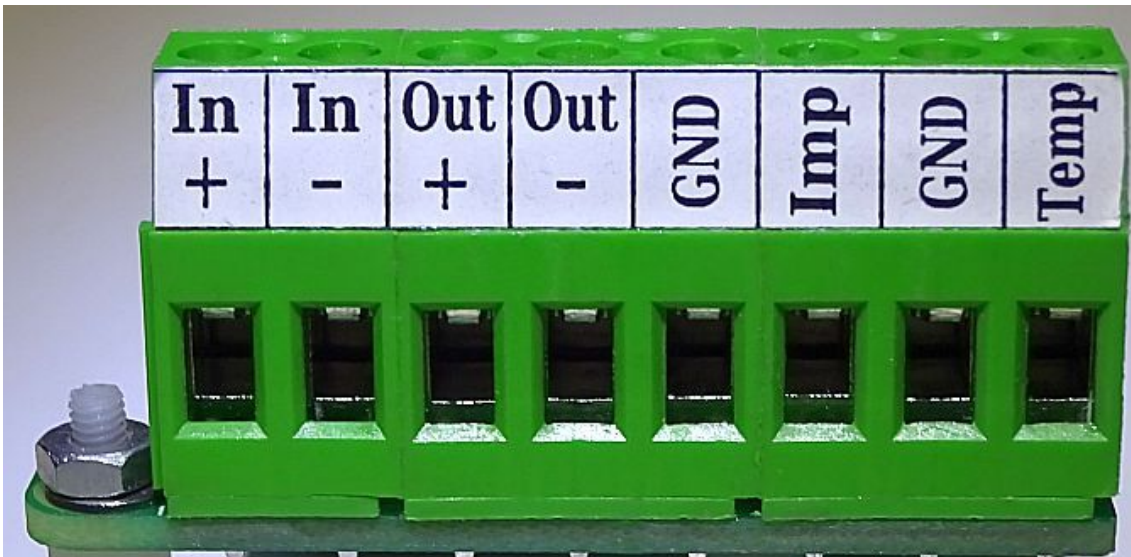


Diese Nachricht verschwindet nach einem Klick auf 'Ok'.

Damit ist alles Wesentliche zur Bedienung der Basis-Software gesagt. Jetzt ist es an der Zeit echte Messwerte zu erzeugen. Dazu ist es notwendig die Anschlussklemmen des PiLoggers mit Sensoren und Anderem zu verbinden.

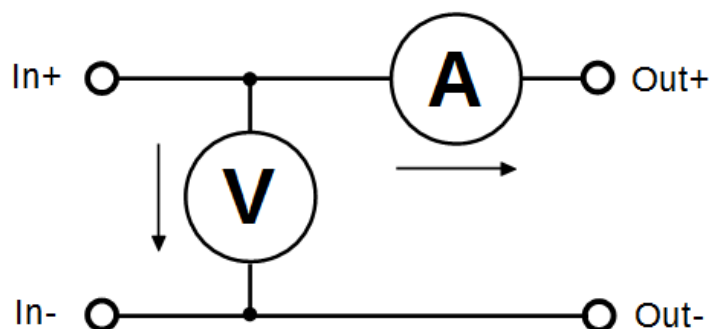
7 Anschlussklemmen

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. $2\text{ m}\Omega$ 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 mΩ 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 !

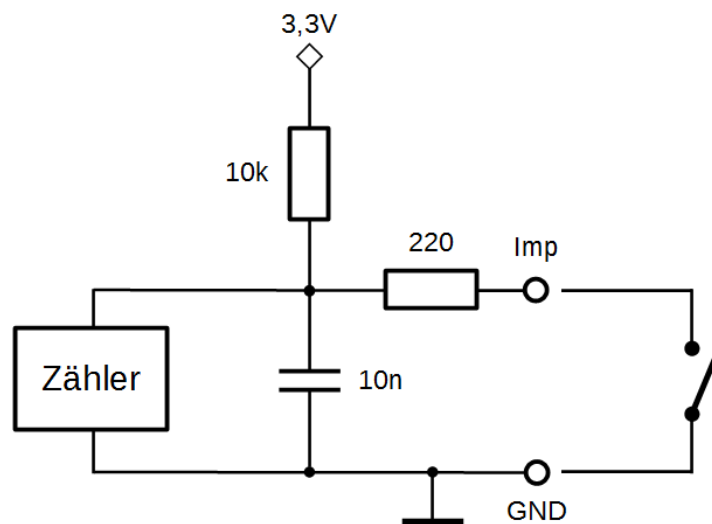
Also bitte auch die Kabellänge immer möglichst kurz halten.

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.

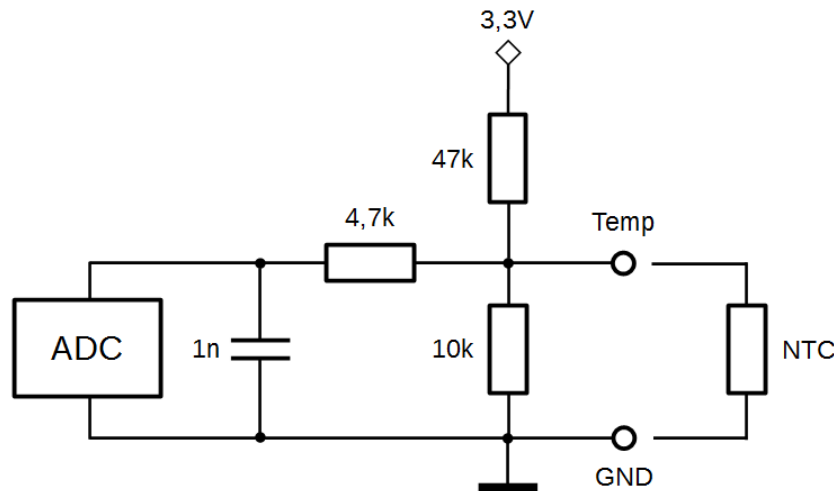
Bei 50% Puls-Pause-Verhältnis liegt die maximale Frequenz etwa bei 2 kHz. Der Widerstand des geschlossenen Schalters (bzw. durchgesteuerten Open Collectors) muss unter 3 kΩ liegen, um eine sichere Low-Erkennung noch zu ermöglichen.

7.3 Temperaturfühlereingang

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

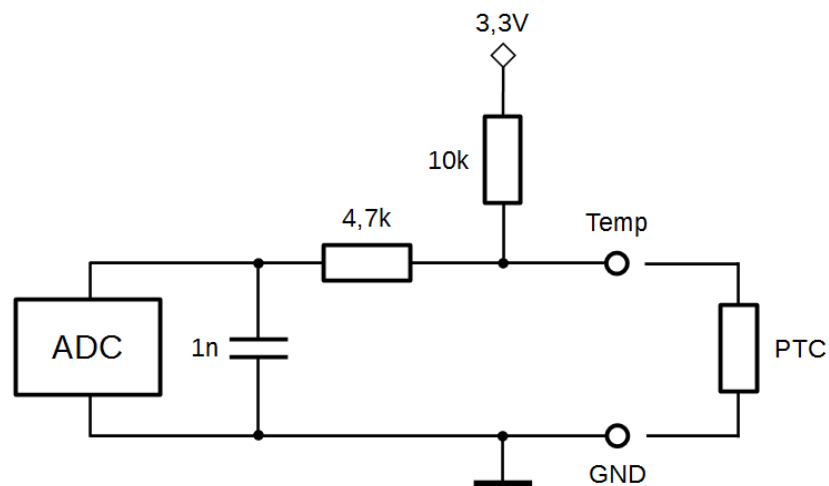
Der Eingang lässt sich mit Hilfe der Kurzschlussbrücke JP1 auf 2 Typen von Temperaturfühlern einstellen:

1. JP1 in Stellung 1-2 : **NTC**



In dieser Stellung ist die Beschaltung für den Anschluss eines NTC (Widerstand mit Negative Temperature Coefficient) mit 10 k Ω Nennwiderstand ausgelegt.

2. JP1 in Stellung 2-3 : **PTC**

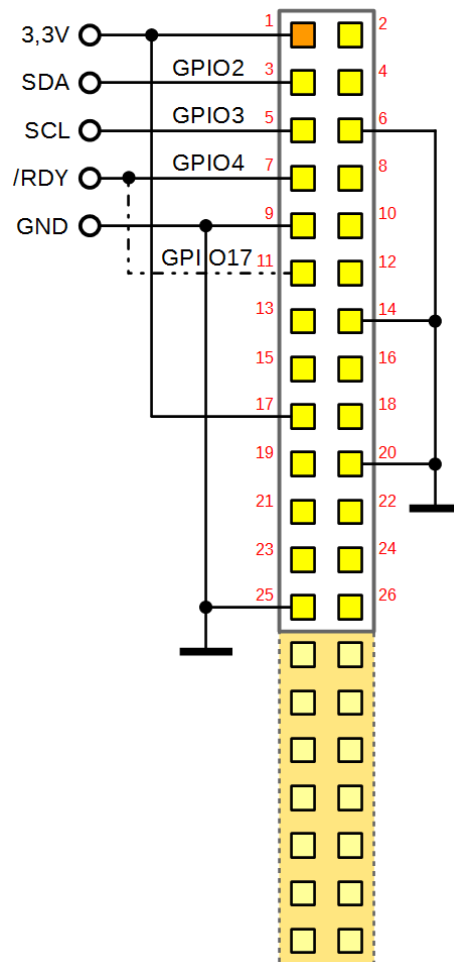


In dieser Stellung ist die Beschaltung für den Anschluss eines PTC (Widerstand mit Positive Temperature Coefficient) mit 1 k Ω Nennwiderstand ausgelegt.

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) kann am GPIO4 (Pin 7) des Raspberry Pi als Interrupt konfiguriert werden, 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 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 6.2.1 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 – das Passwort – mitgesendet wird. Dies soll die Wahrscheinlichkeit für ein versehentliches Ausführen, z.B. durch eine gestörte Kommunikation, senken.

Lesen/Schreiben	Register	1 Byte		
Konfiguration	Write	Read	Wert	
I ² C Slave Adresse	0x01	0x11	default 0x48	0x04...0x7C
Messzyklus Basis	0x02	0x12	0 = 32768 = 1 sec 1 = 8192 = 250 ms 2 = 512 = 15,625 ms 3 = 64 = 1,953125 ms	
Messzyklus Faktor 1	0x03	0x13	1 – 255 low	1...65535
Messzyklus Faktor 2	0x04	0x14	0 – 255 high	
Mittelungsfaktor	0x05	0x15	1 – 7	2 ^x : 2...128
Zeitexpansion 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

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 'pillogger' den Wert des zu benutzenden Busses zu.

```
import smbus
```

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

Lesen eines 1-Byte-Registers:

```
wert = pillogger.read_byte_data(address, register)
```

Beispiel:

```
timebase = pillogger.read_byte_data(0x48, 0x12)
```

Schreiben eines 1-Byte-Registers:

```
pillogger.write_byte_data(address, register, databyte)
```

Beispiel:

```
pillogger.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	0x20...0x43			0x78
Block2 Lesen 16x Word	0x80	0x30...0x63			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 'pillogger' den Wert des zu benutzenden Busses zu.

```
import smbus
```

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

Lesen eines 2-Byte-Registers:

```
wert = pillogger.read_word_data(address, reg)
```

Beispiel:

```
tempmittel = pillogger.read_word_data(0x48, 0x31)
```

Lesen eines 4-Byte-Wertes:

```
wertL = pillogger.read_word_data(address, reg)
```

```
wertH = pillogger.read_word_data(address, reg+1)
```

```
wert = wertH * 65536 + wertL
```

Beispiel:

```
LeistungMaxL = pillogger.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	0...600mV	1/109,2267
Impulseingang	unsigned 16 bit	0...+65535	0,2	m/s		
Spannung	unsigned 16 bit	0...+65535	0,000915541	V	0...60V	1/1092,267
Strom	signed 16 bit	-32768... +32767	0,000466933	A	-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 6.2.2 RasPi Einstellungen – Faktor Pulse). Der Standard Maßfaktor ist hier nur die Vorgabe aus der Basis-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 Basis-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 :

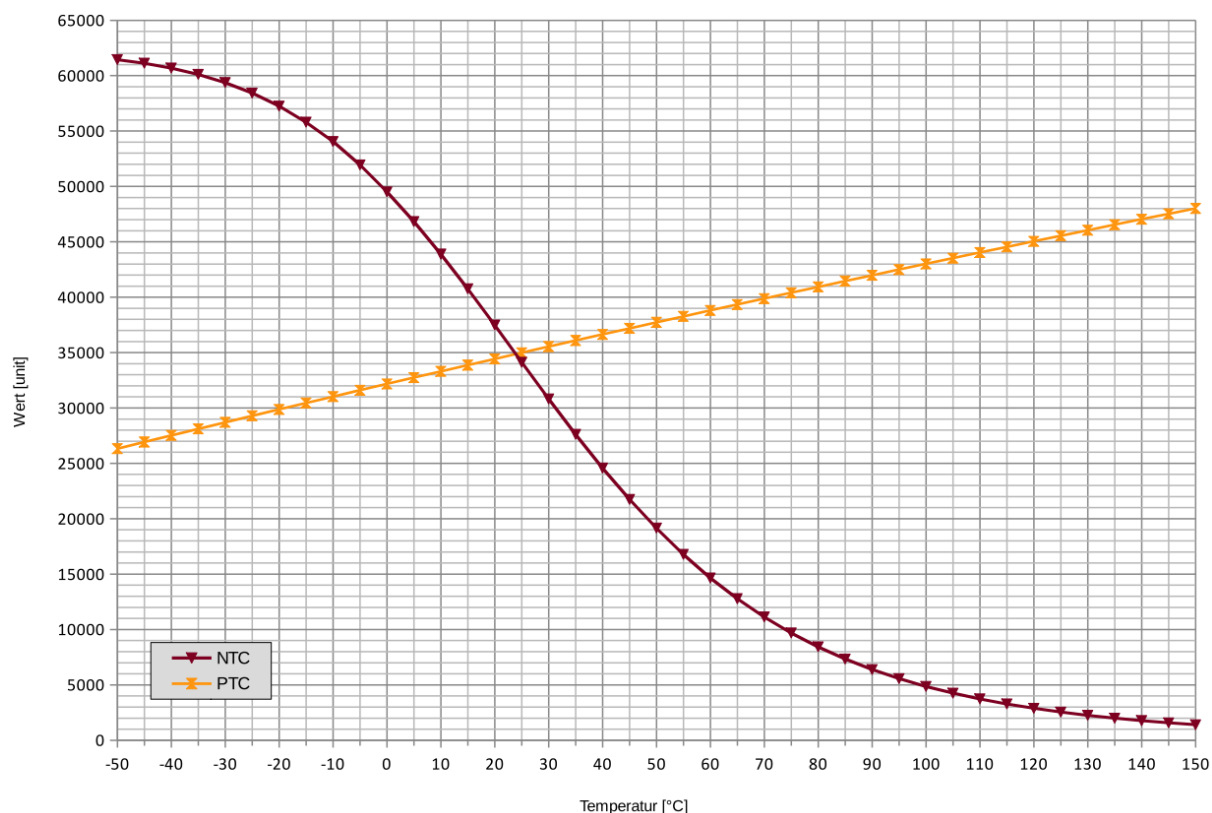
Pt1000				NTC 10k			
T [°C]	R [Ω]		T [°C]	R [Ω]		T [°C]	R _{nom} [Ω]
-55	783,30		55	1213,24		55	3039
-50	803,14		60	1232,46		60	2536
-45	822,96		65	1251,64		65	2128
-40	842,74		70	1270,79		70	1794
-35	862,50		75	1289,92		75	1518
-30	882,23		80	1309,01		80	1290
-25	901,93		85	1328,07		85	1100
-20	921,60		90	1347,10		90	942
-15	941,25		95	1366,09		95	809
-10	960,86		100	1385,06		100	697
-5	980,44		105	1403,98		105	604
0	1000,00		110	1422,87		110	525
5	1019,53		115	1441,72		115	457
10	1039,03		120	1460,54		120	400
15	1058,50		125	1479,31		125	351
20	1077,94		130	1498,04		130	308
25	1097,35		135	1516,74		135	272
30	1116,74		140	1535,38		140	240
35	1136,09		145	1553,99		145	213
40	1155,42		150	1572,55		150	189
45	1174,73		155	1591,06		155	168
50	1194,00						

T [°C]	R _{nom} [Ω]		T [°C]	R _{nom} [Ω]
-55	878900		55	3039
-50	617590		60	2536
-45	439340		65	2128
-40	316180		70	1794
-35	230060		75	1518
-30	169150		80	1290
-25	125550		85	1100
-20	94143		90	942
-15	71172		95	809
-10	54308		100	697
-5	41505		105	604
0	32014		110	525
5	25011		115	457
10	19691		120	400
15	15618		125	351
20	12474		130	308
25	10000		135	272
30	8080		140	240
35	6569		145	213
40	5372		150	189
45	4424		155	168
50	3661			

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

Pt1000				NTC 10k			
T[°C]	Wert	T[°C]	Wert	T[°C]	Wert	T[°C]	Wert
-55	25718	55	38262	-55	61682	55	16767
-50	26319	60	38800	-50	61441	60	14645
-45	26918	65	39334	-45	61114	65	12772
-40	27513	70	39866	-40	60679	70	11126
-35	28106	75	40396	-35	60107	75	9680
-30	28695	80	40923	-30	59367	80	8423
-25	29281	85	41447	-25	58424	85	7328
-20	29864	90	41968	-20	57247	90	6382
-15	30444	95	42487	-15	55797	95	5563
-10	31021	100	43002	-10	54054	100	4854
-5	31595	105	43516	-5	51942	105	4249
0	32167	110	44026	0	49509	110	3726
5	32735	115	44534	5	46824	115	3272
10	33300	120	45039	10	43884	120	2878
15	33863	125	45541	15	40748	125	2539
20	34422	130	46040	20	37484	130	2245
25	34979	135	46537	25	34124	135	1988
30	35533	140	47031	30	30815	140	1763
35	36085	145	47522	35	27607	145	1566
40	36633	150	48010	40	24561	150	1395
45	37179	155	48496	45	21740	155	1244
50	37722			50	19144		

Als Diagramm sieht das dann so aus:



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 Basis-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:

$$T = M^2 / 23006845,6 + M / 167,473355 - 237,0911$$

(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) :

$$T = m^3 * -50245,929 + m^2 * 16224,33 + m * -2068,25 + 184,196$$

Messwert M in 8424...54054 (+80...-10 °C) :

$$T = m^3 * -247,0087 + m^2 * 394,7776 + m * -305,093 + 111,6473$$

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

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.

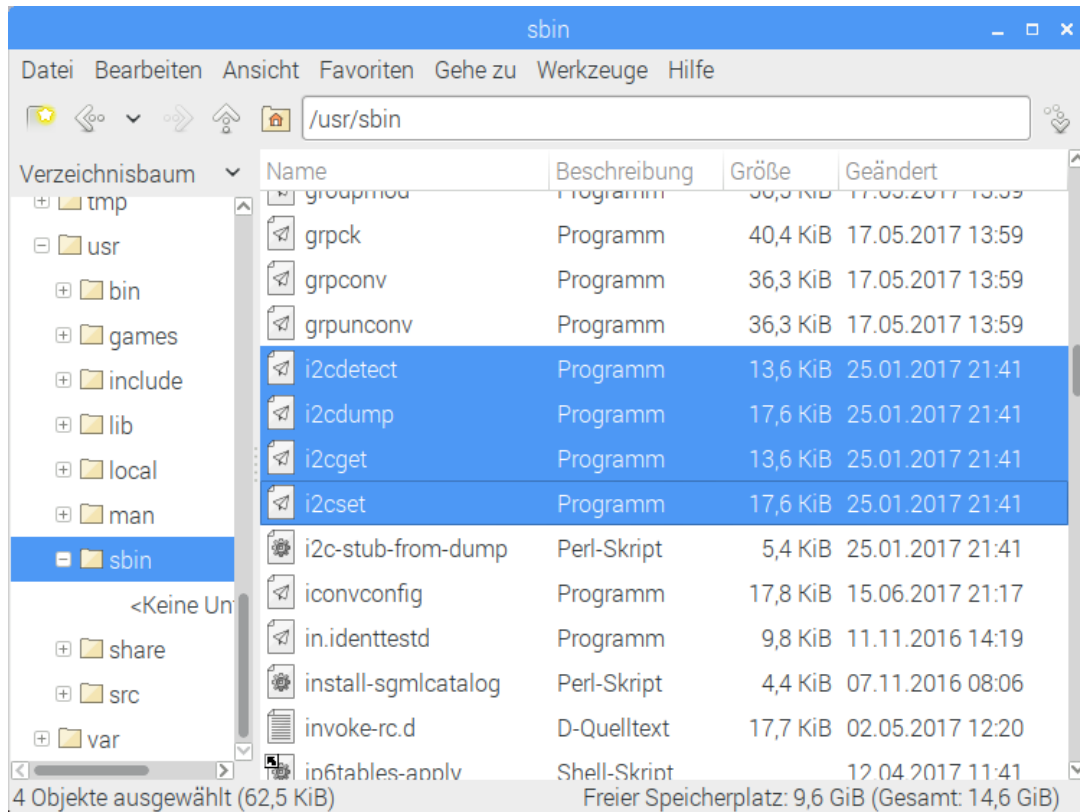
Die daraus resultierenden Korrekturwerte können dann einfach in der Software, die auf dem Raspberry Pi läuft, eingebaut werden (also zum Beispiel im Python-Programm der Basis-Software), um so zu erreichen, dass die angezeigten, bzw. in der Log-Datei abgelegten Werte so genau wie möglich sind.

13 Anhang

13.1 Die I²C Tools

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:

```
pi@PiLogger:~$ i2cdetect -y 1
 0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  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>

13.2 SD Karte schneller machen

Durch eine Overclock Einstellung kann beim Booten dem SD Card Interface die doppelte Geschwindigkeit zugewiesen werden. Standardmäßig wird dieses Modul im Prozessor mit 50 MHz getaktet, durch hinzufügen einer Zeile in der Datei 'config.txt' im Verzeichnis '/boot' wird stattdessen ein Takt (Clock) von 100 MHz eingestellt.



Beim Raspberry Pi 3 verhindert diese Overclock-Einstellung das Erkennen des integrierten WLAN/Bluetooth-Moduls.

Zum Schreiben in die Datei 'config.txt' sind Administratorrechte notwendig. Am besten ruft man den Editor 'Leafpad' direkt mit diesen Rechten und der Datei auf:

```
sudo leafpad /boot/config.txt
```

Danach fügt man am Ende der Datei diese Zeile ein:

```
dtoverlay=sdhost,overclock_50=100
```

Anschließend speichern und Leafpad beenden und den Raspberry Pi neu starten.

Weitere Informationen zum Thema SD Card und Interface Geschwindigkeit gibt es hier:

https://www.sdcard.org/consumers/speed/bus_speed/