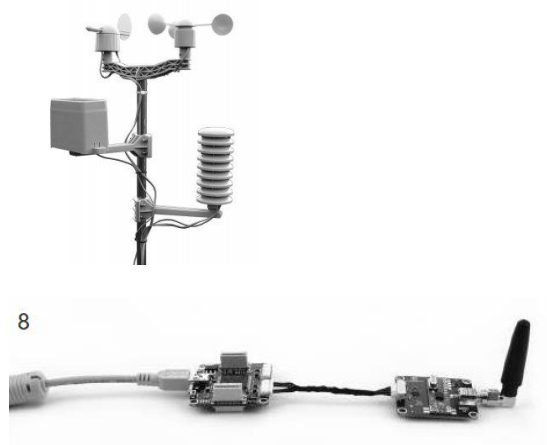


WS-6147

<https://www.rasppishop.de/Aussen-Wetterstation-WS-6147>



Outdoor Weather Station WS-6147

<https://www.rasppishop.de/Aussen-Wetterstation-WS-6147>

- Anemometer (Windgeschwindigkeitssensor),
- Windrichtungssensor,
- Pluviometer (Niederschlagssensor),
- Thermometer (Temperatursensor) und
- Hygrometer (Luftfeuchtigkeitssensor).
- Temperatur in °C mit 0,1°C Auflösung
- Luftfeuchtigkeit in %RH mit 1%RH Auflösung
- Windgeschwindigkeit in m/s mit 0,1m/s Auflösung
- Böengeschwindigkeit in m/s mit 0,1m/s Auflösung
- Niederschlag in mm mit 0,1mm Auflösung
- Windrichtung in 16 Schritten (N, NNE, NE, ENE, E, ESE, SE, SSE, S, SSW, SW, WSW, W, WNW, NW, NNW)
- Batteriestatus

Outdoor Weather Bricklet

<https://www.rasppishop.de/Outdoor-Weather-Bricklet>

Master Brick:

<https://www.tinkerforge.com/de/shop/bricks/master-brick.html>

ZeroBrick

<https://www.rasppishop.de/HAT-Zero-Brick>

Aufbau:

<https://www.manualslib.de/manual/449548/Tinkerforge-Ws-6147.html>

Brick und Bricklet an RPI

(1) https://www.tinkerforge.com/de/doc/Embedded/Raspberry_Pi.html

(2) https://www.tinkerforge.com/de/doc/Software/Brickd_Install_Linux.html#brickd-install-linux

Brick und Bricklet incl. Brick Viewer installieren:

(3) https://www.tinkerforge.com/de/doc/Embedded/Raspberry_Pi.html

TH-6148 Sensor

<https://www.tinkerforge.com/de/shop/temperature-humidity-sensor-th-6148.html>

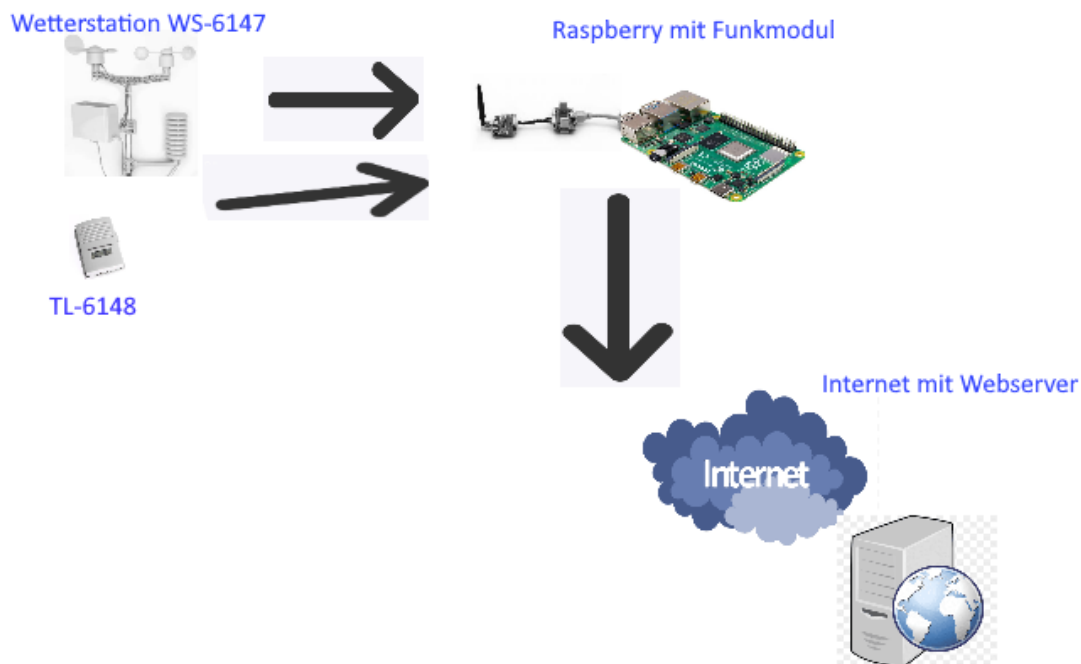
Brickviewer straten:

Brickv

Brick-Logger:

(5) https://www.tinkerforge.com/de/doc/Software/Brick_Logger.html#brick-logger-standalone

```
python3 brick-logger.py test.json
```



A) Wetterstation

Die Wetterstation hat eine ID, die aber mit dem Batteriewechsel wechselt. Diese und die Wetterdaten werden gesendet.

Der Sensor TS-6148 sendet ebenfalls und verfügt über eine eigene ID, die ebenfalls beim Batteriewechsel wechselt.

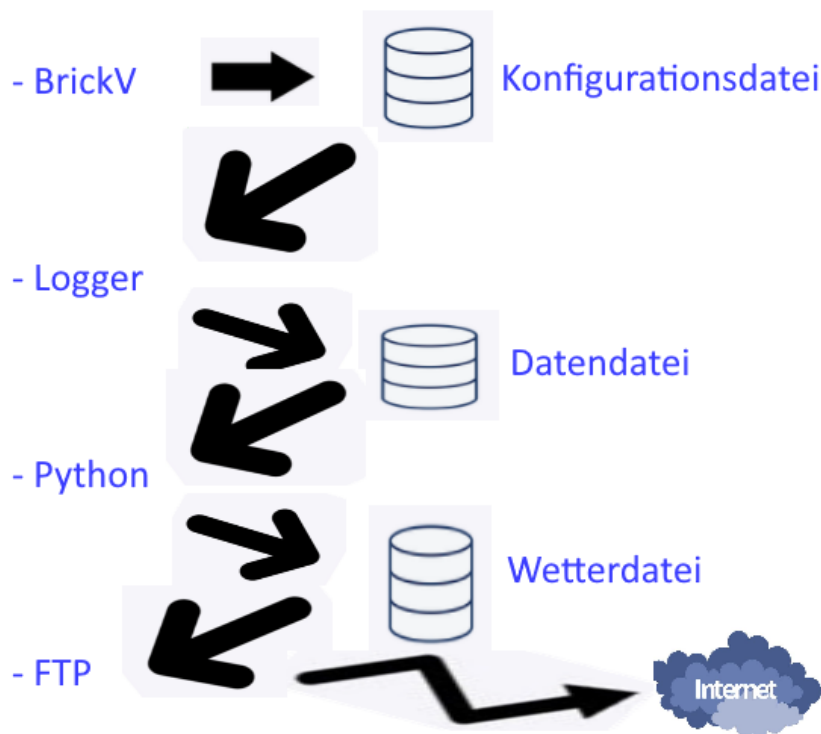
B) Raspberry (RPI)

Der Computer empfängt über eine Platine (Brick und Bricklet genannt) die Wetterdaten (per USB angebunden).

Brickv: Dieses Programm zeigt die Daten der Wetterstation und des Sensors auf dem RPI an. UND mittels dieser Software wird eine Konfigurationsdatei angelegt, die vom Logger benutzt wird. Es ist ein Python-Programm. Wie man das Programm installiert, steht oben unter Aufbau und Link (5): Brickviewer installieren.

Logger: der Logger wird z.B. über die RC.LOCAL-Datei mit dem Start des RPIs im Hintergrund gestartet. Es ist ein Python-Programm. Siehe auch Link (5). Dort findet man auch das Python-Script Brick-Logger.py.

Der Logger wird über die RC-Local-Datei gestartet und läuft permanent im Hintergrund.



Konfigurationsdatei:

Diese Datei (erstellt mit BrickV) wird vom Logger benutzt. Hier steht drin, wohin die Daten geschrieben werden sollen.

Datendatei:

In diese Datei schreibt der Logger alle Daten chronologisch fort. Diese Datei steht auf der RAM-Disk im RPI und enthält bis zum nächsten Runterfahren alle Daten des Tages.

```

Datei Bearbeiten Format Ansicht Hilfe
|TIME;NAME;UID;VAR;RAW;UNIT
21.11.2021 19:39:23;Outdoor Weather Bricklet;SD8;Chip Temperature;22;°C
21.11.2021 19:39:23;Outdoor Weather Bricklet;SD8;Sensor Data:208-Temperature;224;°C/1
21.11.2021 19:39:23;Outdoor Weather Bricklet;SD8;Sensor Data:208-Humidity;53;%RH
21.11.2021 19:39:23;Outdoor Weather Bricklet;SD8;Sensor Data:208-Last Change;36;s
21.11.2021 19:39:23;Outdoor Weather Bricklet;SD8;Station Data;ERROR[Outdoor Weather B
21.11.2021 19:44:23;Outdoor Weather Bricklet;SD8;Chip Temperature;22;°C
21.11.2021 19:44:23;Outdoor Weather Bricklet;SD8;Sensor Data:208-Temperature;224;°C/1
21.11.2021 19:44:23;Outdoor Weather Bricklet;SD8;Sensor Data:208-Humidity;53;%RH

```

Da beim Batteriewechsel die die doppelte Menge an Datensätzen für eine Messung entstehen können, kann es zeitweise zu mehreren gleichen Datensätzen je Einheit kommen.

Wetterdatei:

Hier stehen die Daten der letzten Messung:

23.02.2022 07:51:29;19;52;81;56;75;10;20;0;Nord-West;ok

Dieser Datensatz wird auf den Webserver übertragen. Zuvor wird noch der Regenwert als Tageswert ermittelt.

Station Data

Chip

Sensor Data

Die ID der jeweiligen Station (im Beispiel 2 oder 208) wechselt nach jedem Batteriewechsel. Und die einzelnen Messwerte werden meist als eine Zahl oder Werte dargestellt, die aber häufig noch umgerechnet werden müssen (Kommastellen, Batteriestatus,, ...).

Die Anzahl der Datensätze ist je Messung konstant, außer nach einem Batteriewechsel. Hier kann es zur doppelten Datensätzen (jeweils mit der alten und neuen ID) kommen, aber nur bis zum nächsten Booten des RPIs.

Die Wetterdatei wird auf den Webserver übertragen. Hierzu wird ein eigenes SH-Script verwendet: ums.sh.

In dem Script erfolgt dann auch das Übertragen der Daten auf den Webserver. Es wird über die Crontab aufgerufen. IN der Crontab steht auch das Herunterfahren des RPI. Die Crontab steht in ETC.

In der RC.Local (steht in etc) steht der Aufruf der Internetverbindung und der Aufruf des Loggers.

In der Datei Interfaces (steht in etc/network) steht die Daten für die Simcard und auch für die LAN-Verbindung.

Das Script ums.sh liest die Datendatei (daten.csv), die den ganzen Tag über fortgeschrieben wird, und erstellt die Übergabedatei wetter.csv, die nur die letzten Werte enthält. Ebenfalls ruft das Script den Job2.sh auf, der die FTP-Übertragung macht.

Damit es besser für die Speicherkarte ist, werden die Bewegungsdaten in der RAM-Disk abgelegt. Dieses Ramdisk ist der Ordner LOG, der nach dem Booten leer ist.

C) Internet

Die Wetterdaten (Wetterdatei) stehen in einer Datei, die alle 5 Minuten mit den neuesten – Werten vom RPI übertragen und überschrieben wird.

23.02.2022 07:51:29;19;52;81;56;75;10;20;0;Nord-West;ok

Diese Datei heißt werteneu.csv

Ablauf im Einzelnen auf dem RPI

In der RC.Local wird der Logger gestartet.

```
python3 root/brick-logger.py home/pi/test.json >>/home/pi/rclocallog.txt
2>>/home/pi/rclocallog.txt
```

UMS.PY

Auf dem RPI läuft das Programm UMS.PY. Dieses Programm holt und sammelt die Daten der Wetterstation und legt sie im Ordner LOG und der Datei daten.csv ab.

```
 Datei  Bearbeiten  Format  Ansicht  Hilfe
|TIME;NAME;UID;VAR;RAW;UNIT
21.11.2021 19:39:23;Outdoor Weather Bricklet;SD8;Chip Temperature;22;°C
21.11.2021 19:39:23;Outdoor Weather Bricklet;SD8;Sensor Data:208-Temperature;224;°C/1
21.11.2021 19:39:23;Outdoor Weather Bricklet;SD8;Sensor Data:208-Humidity;53;%RH
21.11.2021 19:39:23;Outdoor Weather Bricklet;SD8;Sensor Data:208-Last Change;36;s
21.11.2021 19:39:23;Outdoor Weather Bricklet;SD8;Station Data;ERROR[Outdoor Weather B
21.11.2021 19:44:23;Outdoor Weather Bricklet;SD8;Chip Temperature;22;°C
21.11.2021 19:44:23;Outdoor Weather Bricklet;SD8;Sensor Data:208-Temperature;224;°C/1
21.11.2021 19:44:23;Outdoor Weather Bricklet;SD8;Sensor Data:208-Humidity;53;%RH
```

UMS.PY wird in der Crontab gestartet.

JOB2.sh

Am Ende des Scripts ums.py wird Job2.sh gestartet. Dieser Job überträgt die Wetterdatendaei werteneu.csv auf den Server.

Ablauf im Einzelnen auf dem Webserver

Auf dem Webserver gibt es drei Dateien:

WerteNEU.csv die Wetterdaten der letzten Messung, wird alle 3 Minuten überschrieben
WerteNEUGES.csv die Gesamtwertterdatei mit den einzelnen Messung aus werteNEU.csv
werteLAST.txt eine Prüfdatei

Aktuellneu.php

Anzeigen der aktuellen Wetterdaten. Hier wird die Datei werteNEU.csv ausgelesen und angezeigt. Dieses Script wird auch auf der Homepage verwendet (Kopie).

aktuell-neu2.php

Dieses Programm liest die WerteNEU.csv und hängt die Daten an die Wettergesamtdatei werteNEUGES.csv an. Dabei wird geprüft, ob die Daten schon vorhanden sind (Prüfdatei wertelAST.txt). Gestartet wird dieses Programm über einen Cronjob auf dem Webserver.

aktuell-neu3.php

Dieses Programm liest die WerteNEUGES.csv und zeigt die Daten an.