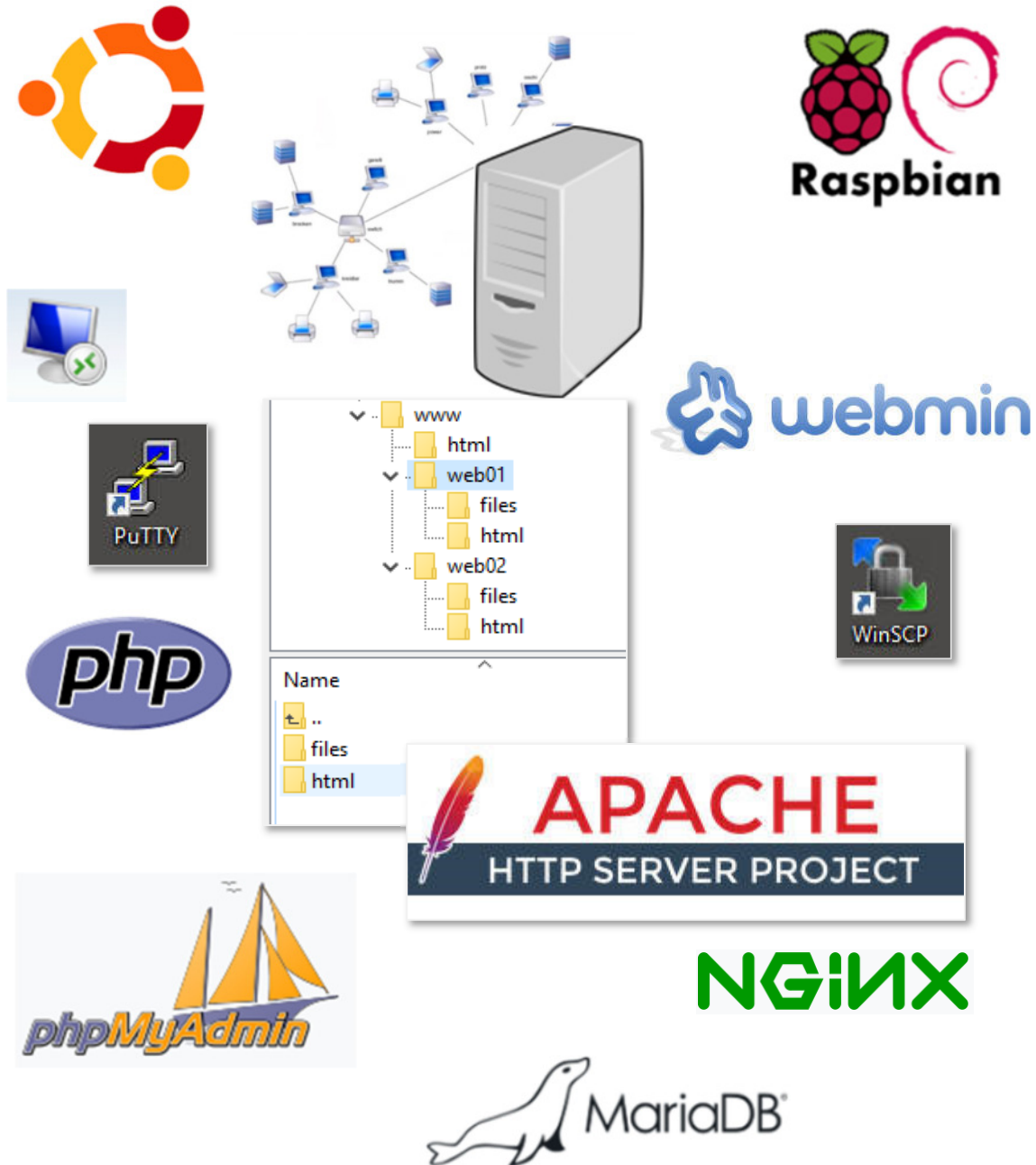


Raspberry Pi - Wetterstation

ACHTUNG:
Beta-Version

Ausbaustufe 2 - Datenbankanbindung

darin enthalten sind Beispiele zur Linux-Installation und zu bash-Scripting, Remotezugriff, Linux-Serverkonfiguration, Datenbankmodellierung, Pythonscripting u.v.m.



Vorwort

Die vorliegende Dokumentation soll die Planung, Entwicklung und den Aufbau einer kleinen Wetterstation mit dem Kleincomputer Raspberyy Pi Zero ausführlich darstellen. Sie erhebt jedoch keinen Anspruch auf Vollständigkeit. Eventuelle technische Änderungen können nicht laufend in der Dokumentation berücksichtigt werden.

Bei der Umsetzung des Projekts wurden sowohl auf der Hardware- wie auch auf der Softwareseite einige Versuchsinstallationenen durchgeführt, die aus mehreren Gründen wieder verworfen wurden. Die Beschreibung stützt sich auf Versionsstände ca. Sommer 2019.

Als im Juli 2019 das neue Raspberry-Image „Buster“ veröffentlicht wurde, konnte nochmals eine komplette Installation in kleinen Schritten nachvollzogen und hier endgültig dokumentiert werden.

Inhalt

Abbildungsverzeichnis	III
Verzeichnis der Programmcode-Ausschnitte	III
Internetquellen	III
Quellangaben der verwendeten Grafiken	IV
1 Das Projekt – aktueller Stand	1
1.1 Ist-Zustand	1
1.2 geplante Erweiterung und Verbesserung	1
2 Planung der Erweiterungen	2
3 Testdaten in der Datenbank ablegen	3
3.1 MariaDB als MySQL-Datenbankmaschine	3
3.2 Vorbereitung des Datenbankzugriffs	3
3.3 Test der Datenbankverbindung	5
3.4 Test-Messdaten abspeichern	6
3.5 Erkenntnisse aus den Testscripts	8
4 Datenban.....	9
Anhang	11

Abbildungsverzeichnis

Abbildung 1: Sensoreinbau, Raspberry Pi Zero Einbau	1
Abbildung 2: Endmontage	1
Abbildung 3: Datenbankmodell	3
Abbildung 4: Python-Version anzeigen	3
Abbildung 5: mysql-Connector für Python installieren	4
Abbildung 6: Datenbankanbindung an ein Python-Script	5
Abbildung 7: Datenbankanbindung an ein Python-Script	6

Verzeichnis der Programmcode-Ausschnitte










Programmcode 1: Test-Messwerte in Datenbank ablegen	7
---	---



Internetquellen

Grafik GPIOs

<https://indibit.de/raspberry-pi-die-gpio-schnittstelle-grundlagenbelegung/>

Quellangaben der verwendeten Grafiken

	<p>Das Logo des Linux-Tux wurde freigegeben von Larry Ewing unter folgenden Bedingungen: „Es ist erlaubt, diese Grafik zu verwenden und/oder zu verändern. Bedingung ist jedoch: Falls jemand fragt, muss man mich – lewing@isc.tamu.edu – als Urheber nennen und auf GIMP hinweisen.“ (Ewing, Budig, Gerwinsky, & Hagel, 2008)</p>
<p>Linux-UBUNTU</p> 	<p>Grafik gemeinfrei Datei: https://de.wikipedia.org/wiki/Ubuntu#/media/File:Ubuntu_logo.svg</p>
<p>Webserver Apache</p> 	<p>Datei: https://de.wikipedia.org/wiki/Apache_Software_Foundation#/media/File:Apache_Software_Foundation_Logo_(2016).svg</p> <p>Licensed under the Apache License, Version 2.0 (the "License");you may not use this file except in compliance with the License.You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.</p>
<p>PHP Skriptsprache</p>  <p>Datenbankserver</p>  <p>DB-Admin</p> 	<p>Datei php: https://de.wikipedia.org/wiki/PHP#/media/File:PHP-logo.svg Datei DB: https://de.wikipedia.org/wiki/MariaDB#/media/File:MariaDB_Logo.png Datei : https://de.wikipedia.org/wiki/PhpMyAdmin#/media/File:PhpMyAdmin-Logo.svg</p> <p>Attribution-ShareAlike 4.0 International (CC BY-SA 4.0) (Creative Commons PoBox 1866, Mountain View, CA 94042)</p> <p>You are free to: Share — copy and redistribute the material in any medium or format Adapt — remix, transform, and build upon the material for any purpose, even commercially. This license is acceptable for Free Cultural Works.</p> <p>The licensor cannot revoke these freedoms as long as you follow the license terms. Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.</p> <p>ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.</p> <p>No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.</p> <p>https://creativecommons.org/licenses/by-sa/4.0/legalcode</p>
<p>WinSCP</p> 	<p>Datei: https://de.wikipedia.org/wiki/WinSCP#/media/File:WinSCP_Logo.png Lizenz: https://www.gnu.org/licenses/gpl-3.0.html</p>
<p>PuTTY</p> 	<p>Datei: https://de.wikipedia.org/wiki/PuTTY#/media/File:PuTTY_icon_128px.png Lizenz: https://opensource.org/licenses/mit-license.php</p>
<p>RaspBerryPi</p> 	<p>Lizenzen: https://www.raspberrypi.org/trademark-rules/ https://www.raspberrypi.org/</p>

<div>WebAdmin</div> 	Datei: http://www.webmin.com/graphics/webmin-full.jpg Lizenz: http://www.webmin.com/graphics.html
	Grafik gemeinfrei Datei: https://upload.wikimedia.org/wikipedia/commons/c/c5/Nginx_logo.svg Autor: https://commons.wikimedia.org/w/index.php?curid=24774395

Weitere Abbildungen (Bilder, Grafiken) wurden selbst angefertigt und sind mit einer Fußnote gekennzeichnet.

1 Das Projekt – aktueller Stand

Ziel des ersten Teils des Projekts soll sein, eine praktische Anwendung zur intensiven Einarbeitung in die Möglichkeiten des Raspberry Pi zu erhalten. Hierzu wurde das Modell Raspberry Pi Zero mit mehreren Sensoren und der internen Kamera ausgestattet als kleine Wetterstation an einer Hauswand montiert.

Die Daten wurden in einzelnen html-Dateien gesammelt und auf der Raspi-Webseite mit Hilfe des Apache2-Webservers angezeigt.

1.1 Ist-Zustand

Die folgenden Abbildungen zeigen den Ist-Zustand des Hardware-Aufbaus.

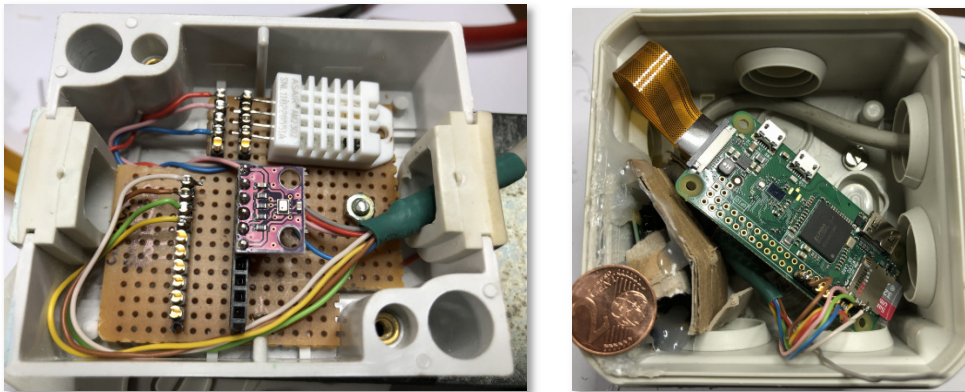


Abbildung 1: Sensoreinbau, Raspberry Pi Zero Einbau



Abbildung 2: Endmontage

1.2 geplante Erweiterung und Verbesserung

Zur komfortableren Datenspeicherung und -auswertung wird nun eine SQL-Datenbank unter dem Open-Source-System MariaDB verwendet. Die Daten sollen gesammelt und sowohl in Tabellen- als auch grafischer Form aufbereitet und angezeigt werden.

2 Planung der Erweiterungen

Damit die Messdaten längere Zeit aufbewahrt werden können soll eine Datenbank konstruiert werden. Mit unterschiedlichen Methoden werden die Daten dann aufbereitet und auf der Webseite angezeigt. Durch die Anwendung unterschiedlicher Techniken Darstellung der Messwerte sollen Scripte entstehen, die für andere Anwendungen als Beispiele dienen können.

- Datenbank erstellen
- Messdaten in die Datenbank speichern
- Daten tabellarisch darstellen
- Diagramme zur Aufbereitung der Messwerten verwenden

3 Testdaten in der Datenbank ablegen

Bisher wurden die Messwerte der Sensoren in einzelnen html-Dateien gespeichert und dann wieder auf der Webseite dargestellt. Diese Vorgehensweise ist relativ unflexibel und umständlich. Damit die Wetterdaten längere Zeit aufbewahrt und für statistische Zwecke ausgewertet werden können, sollen sie nun in einer Datenbank abgespeichert werden.

Die bisherigen Messvorgänge werden zum Vergleich weiterhin aktiv geschaltet. Die nun beschriebene Erweiterung wird in einem separaten Bereich der Website angezeigt.

3.1 MariaDB als mySQL-Datenbankmaschine

Die Einrichtung der Datenbank MariaDB wurde in Abschnitt **Fehler! Verweisquelle konnte nicht gefunden werden.** (siehe Seite **Fehler! Textmarke nicht definiert.**) bereits ausführlich beschrieben. In Abbildung 3 wird das Datenbankmodell Messwerte-Datenbank dargestellt.

dbMesswerte	tblMessung	dbMesswerte	tblBesuch
🔑	m_id : int(11)	🔑	b_id : int(11)
📅	m_messzeitpunkt : datetime	📄	b_ip : varchar(50)
#	m_temp : decimal(10,0)	📅	b_zeitpunkt : datetime
#	m_druckrel : decimal(10,0)	📄	b_sonstiges : varchar(255)
#	m_druckabs : decimal(10,0)		
#	m_feucht : decimal(10,0)		

Abbildung 3: Datenbankmodell

Zum Ansteuern von MariaDB soll die bereits zum Erfassen der Messwerte genutzten Programmiersprache Python eingesetzt werden.

- Messdaten erfassen
- Temperatur, Luftfeuchtigkeit und später auch Luftdruck in der Tabelle abspeichern
- Datenbankabfragen auf einer einzigen Webseite ausführen und tabellenartig anzeigen
- Grafiken aufbereiten und anzeigen

3.2 Vorbereitung des Datenbankzugriffs

Zunächst ist es wichtig, die korrekte Version des installierten Python-Moduls zu kennen, um den passenden Datenbanktreiber auswählen zu können.

```
$ python -V
```

```
Python 2.7.16
```

Abbildung 4: Python-Version anzeigen

Wenn (wie hier) mit Python 2.x gearbeitet wird, muss der **mysql-connector-python** manuell nachinstalliert werden.

Das Paket wird heruntergeladen und das Setup wird ausgeführt:

```
$ sudo pip install mysql-connector-python
```

```
pi@h-raspi2:~ $ sudo pip install mysql-connector-python
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting mysql-connector-python
  Downloading https://files.pythonhosted.org/packages/99/74/f41182e6b7aad62b0
90dc9ab/mysql_connector_python-8.0.17-py2.py3-none-any.whl (345kB)
    100% |████████████████████████████████████████| 348kB 204kB/s
Collecting protobuf>=3.0.0 (from mysql-connector-python)
  Downloading https://files.pythonhosted.org/packages/0d/2e/d4b1b67c264ce6722c
247bfa6/protobuf-3.9.1-py2.py3-none-any.whl (432kB)
    100% |████████████████████████████████████████| 440kB 163kB/s
Requirement already satisfied: setuptools in /usr/lib/python2.7/dist-packages
(40.8.0)
Requirement already satisfied: six>=1.9 in /usr/lib/python2.7/dist-packages (f
or python) (1.12.0)
Installing collected packages: protobuf, mysql-connector-python
[ ]
```

Abbildung 5: mysql-Connector für Python installieren

3.3 Test der Datenbankverbindung

Der Zugriff auf MariaDB über das Mess-Pythonscript muss zunächst in kleinen Schritten programmiert werden, um die einzelnen Schritte auch nachhaltig zu verstehen. Hierzu dienen kleine Testprogramme, z.B. `db_test_connect.py`. Diese Programme sollen später als Scriptquelle für das eigentliche Abfrageprogramm der Sensoren verwendet werden.

Abbildung 6 zeigt schematisch, wie aus einem Python-Script auf die Datenbank zugegriffen werden kann.

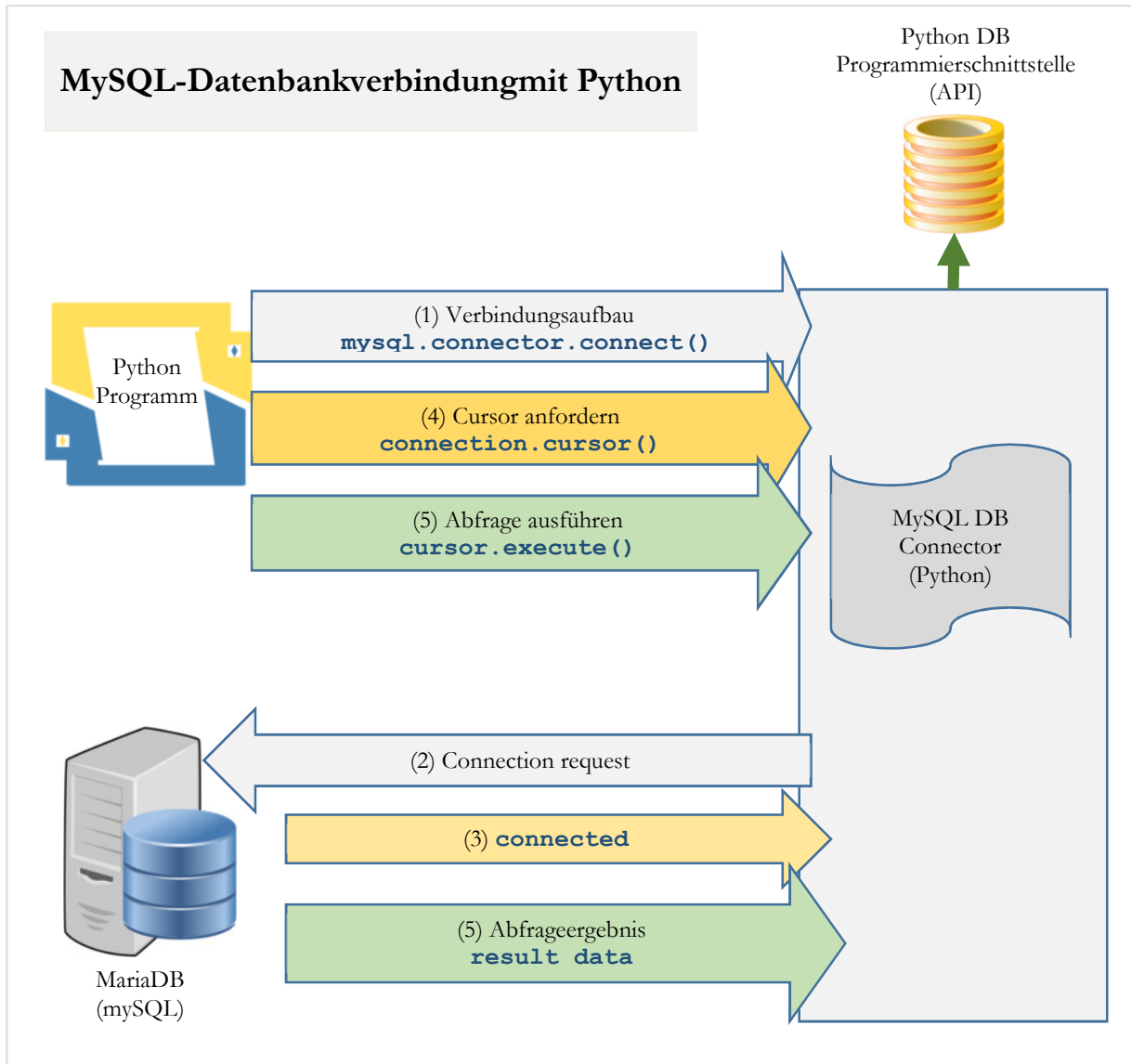


Abbildung 6: Datenbankverbindung an ein Python-Script¹

Das Verbindungstestprogramm wird aufgerufen und zeigt die entsprechenden Meldungen an.

```
$ sudo python dbtest_connect.py
```

```
pi@h-raspi2:/var/www/cgi-bin/wetterstation2019 $ sudo python dbtest_connect.py
('Verbunden mit MySQL-Server Version: ', u'5.5.5-10.3.15-MariaDB-1')
('...mit folgender Datenbank steht die Verbindung: ', (u'dbMesswerte',))
MySQL-Verbindung geschlossen
pi@h-raspi2:/var/www/cgi-bin/wetterstation2019 $
```

¹ Grafik bearbeitet nach <https://pynative.com/python-mysql-database-connection/>

3.4 Test-Messdaten abspeichern

Damit die Sensordaten in der Tabelle gespeichert werden, muss der SQL-Befehl INSERT INTO mit den Messwerten an die Datenbank gesendet werden.

Prinzip der Datensatzspeicherung in MariaDB

Die folgende Grafik zeigt den Vorgang der Datensatzspeicherung. Um den Datenbankzugriff zu realisieren, wird das vorhandene Script der Sensorabfrage erweitert.

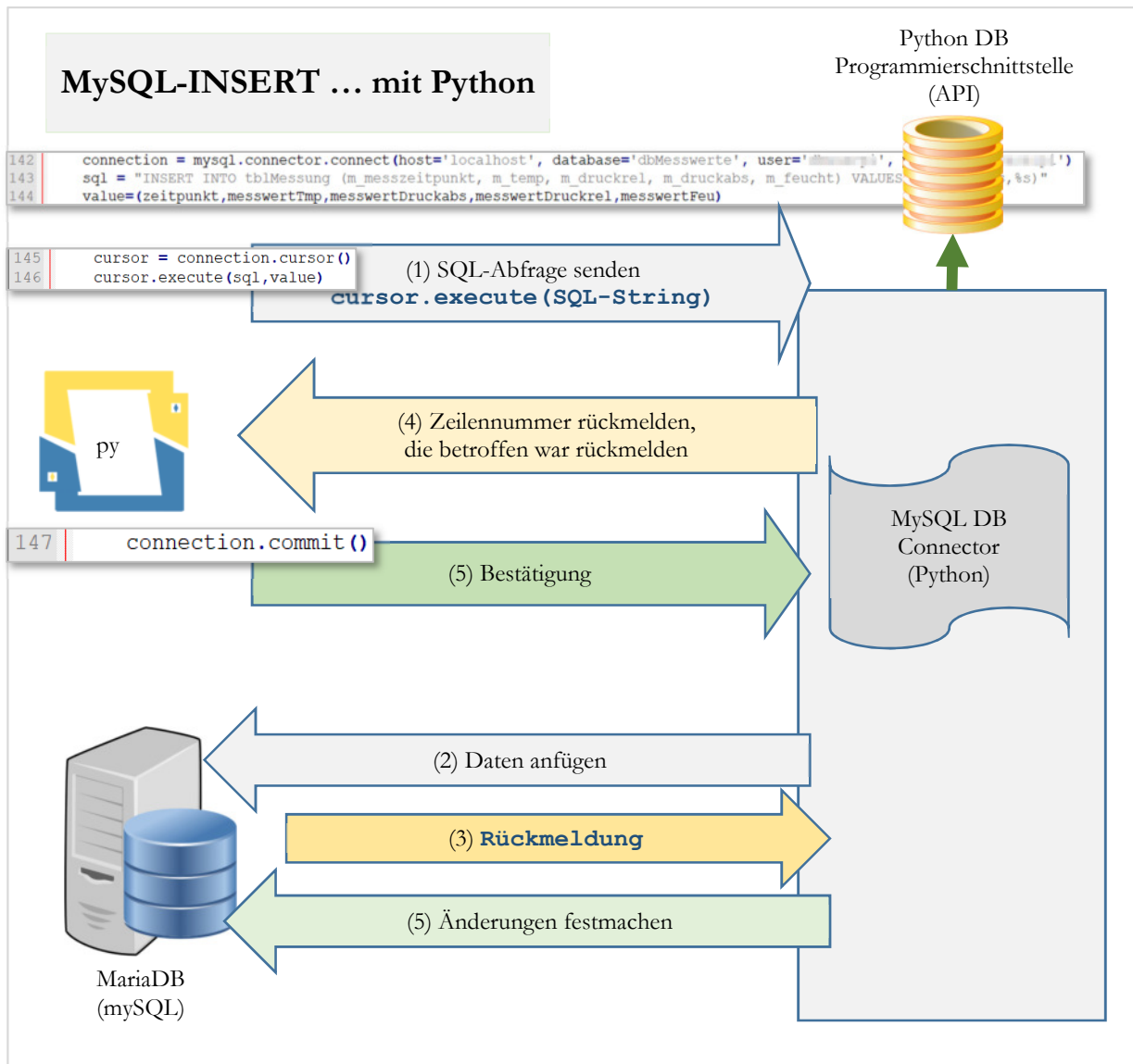


Abbildung 7: Datenbankbindung an ein Python-Skript²

Den kompletten Datenzugriff zeigt gesamte Script (siehe Folgeseite).

² Grafik bearbeitet nach <https://pynative.com/python-mysql-insert-data-into-database-table/>

Python-Script INSERT INTO

Nun gilt es, die Erkenntnisse aus den Tests zur konkreten Abspeicherung der Messwerte der Sensoren zu nutzen. Zunächst werden die Messwerte zusammengestellt und formatiert. Der **try**-Bereich baut die Verbindung zur Datenbank auf, stellt das SQL-Statement zusammen und übergibt es MariaDB. Fehlermeldungen werden bei Bedarf auf der Console ausgegeben. Abschließend wird die Verbindung geschlossen.

```

131 #ab hier Datenbankzugriff
132 # Messwerte in die Datenbank schreiben-----
133 zeitpunkt = time.strftime('%Y-%m-%d %H:%M:00')
134 messwertTmp = '{0:0.1f}'.format(lngTemp)
135 messwertFeu = '{0:0.1f}'.format(lngFeuchte)
136 messwertDruckabs = '{0:0.1f}'.format(pressure)
137 messwertDruckrel = '{0:0.1f}'.format(pressure_nn)
138
139
140 try:
141     #Verbindung zur Datenbank aufbauen - bei Fehler-->expection
142     connection = mysql.connector.connect(host='localhost', database='dbMesswerte', user='dbuserpi', password='dbpass')
143     sql = "INSERT INTO tblMessung (m_messzeitpunkt, m_temp, m_druckrel, m_druckabs, m_feucht) VALUES(%s,%s,%s,%s,%s)"
144     value=(zeitpunkt,messwertTmp,messwertDruckabs,messwertDruckrel,messwertFeu)
145     cursor = connection.cursor()
146     cursor.execute(sql,value)
147     connection.commit()
148     cursor.close()
149     print("Datensatz erfolgreich gespeichert in der Messdatentabelle")
150     cursor.close()
151
152 except mysql.connector.Error as error:
153     print("Fehler beim Abspeichern in die Messdatentabelle {}".format(error))
154
155 finally:
156     if (connection.is_connected()):
157         connection.close()
158         print("MySQL connection geschlossen")
159

```

Programmcode 1: Test-Messwerte in Datenbank ablegen

Der erste Zugriff auf MariaDB zeigt eine Fehlermeldung, die durch Übermitteln eines fehlerhaften Passworts entstand.

```

pi@h-raspi2:/var/www/cgi-bin/wetterstation2019 $ sudo python sensorabfrage2019.py
Temperatur=18.0*C  Feuchtigkeit=81.9%
12.10.2019 - 20:34
Temperatur      : 18.11 C
abs. Luftdruck  : 986.85 hPa
rel. Luftdruck NN : 1017.83 hPa
/var/www/html/wetterstation2019/daten/20.html
Fehler beim Abspeichern in die Messdatentabelle 1045 (28000): Access denied for use
r 'dbuserpi'@'localhost' (using password: YES)
Traceback (most recent call last):
  File "sensorabfrage2019.py", line 156, in <module>
    if (connection.is_connected()):
NameError: name 'connection' is not defined

```

Beim zweiten Versuch mit korrektem Passwort zeigt das Script die erfolgreiche Speicherung der Messwerte an.

```

pi@h-raspi2:/var/www/cgi-bin/wetterstation2019 $ sudo python sensorabfrage2019.py
Temperatur=18.0*C  Feuchtigkeit=81.8%
12.10.2019 - 20:35
Temperatur      : 18.11 C
abs. Luftdruck  : 986.88 hPa
rel. Luftdruck NN : 1017.86 hPa
/var/www/html/wetterstation2019/daten/20.html
Datensatz erfolgreich gespeichert in der Messdatentabelle
MySQL connection geschlossen
pi@h-raspi2:/var/www/cgi-bin/wetterstation2019 $

```

Eine Kontrolle mit phpMyAdmin zeigt die gespeicherten Testwerte.

m_id	1	m_messzeitpunkt	m_temp	m_druckrel	m_druckabs	m_feucht
2		2019-10-12 18:00:00	21	986	1017	62
4		2019-10-12 18:30:00	21	987	1018	62
5		2019-10-12 19:00:00	20	987	1018	69
6		2019-10-12 19:30:00	19	986	1017	72
7		2019-10-12 20:00:00	19	987	1018	74
8		2019-10-12 20:12:00	18	987	1018	79
9		2019-10-12 20:13:00	18	987	1018	80
10		2019-10-12 20:22:00	18	987	1018	81
11		2019-10-12 20:30:00	18	987	1018	81
12		2019-10-12 20:35:00	18	987	1018	82

3.5 Erkenntnisse aus den Testscripts

Das Zusammenfügen der einzelnen Scripts zu einer komfortablen Seite mit Wetterdaten-Auswertung ist erst geplant, wenn auch der zusätzlich zum Einbau geplante Sensor zur Messung des aktuellen Luftdrucks ausführlich getestet und in Raspberry-Gehäuse montiert ist.

4 Datenban

Anhang

Scripting-/Programmcode
