

Bernburg
Dessau
Köthen



Hochschule Anhalt
Anhalt University of Applied Sciences

emw

Fachbereich
Elektrotechnik, Maschinenbau
und Wirtschaftsingenieurwesen

Xumeng Zhang

Vorname Nachname

Elektro- & Informationstechnik, 2014, 4059994

Studiengang, Matrikel, Matrikelnummer

Bachelorarbeit

zur Erlangung des akademischen Grades

Bachelor of Engineering (B. Eng.)

Thema:

**Entwicklung eines verteilten Smartering
Managementsystem unter Verwendung der
Datenbank MySQL**

Prof. Dr. Ingo Chmielewski

Vorsitzender der Bachelorprüfungskommission/1. Prüfer

Prof. Dr. Michael Brutscheck

2. Prüfer

17.11.2017

Abgabe am

Selbstständigkeitserklärung

Hiermit erkläre ich, dass die Arbeit selbstständig verfasst, in gleicher oder ähnlicher Fassung noch nicht in einem anderen Studiengang als Prüfungsleistung vorgelegt wurde und keine anderen als die angegebenen Hilfsmittel und Quellen, einschließlich der angegebenen oder beschriebenen Software, verwendet wurden.

Köthen, den 20.11.2017

Ort, Datum

Unterschrift des Studierenden

Sperrvermerk

Sperrvermerk: ja nein

wenn ja: Der Inhalt der Arbeit darf Dritten ohne Genehmigung der/des (Bezeichnung des Unternehmens) nicht zugänglich gemacht werden. Dieser Sperrvermerk gilt für die Dauer von X Jahren.

Köthen, 20.11.2017

Ort, Datum

Unterschrift des Studierenden

Inhaltsverzeichnis

Selbstständigkeitserklärung	I
Sperrvermerk	I
1 Motivation und Zielsetzung	1
2 Struktur und Funktion des LENA Gerätes	2
2.1 Technische Daten	2
2.2 Messung	3
2.2.1 Spannungs- und Strommessung	3
2.2.2 3-Leiter + N - Netze und Leistungsberechnung	3
2.3 Datenspeichern	4
3 Embedded System-- Beaglebone Black	5
3.1 Geschichte und Entwicklung von BBB.....	5
3.2 Bestandteile der Hardware	6
3.2.1 Kernkomponenten	7
3.2.2 Stecker, LEDs und Schalter.....	7
3.3 Datenübertragen	8
4 Durchführung mit Linux SoC-Systemen	9
4.1 Vergleich der Windows und Linux	9
4.2 Linux-Distribution Ubuntu installieren	10
4.2.1 Vorbereitung der Installation.....	10
4.2.2 Auswahl der Partition bzw. Partitionierung und Bootloader	10
4.2.3 Zeitzone, Tastaturlayout und Benutzerdaten	12
4.2.4 Laufende Installation und Neustart.....	13
4.3 Interne Kommunikation im Beaglebone Black	13
4.3.1 Programm „socieer IPC“	15
4.3.2 Programm LENA „appurtenance.py“	16
4.3.3 Programm „test.py“	20
4.4 Auswerten über Frontend.....	22
5 Visualisierung durch Smart-Metering Volkszaehler	23
5.1 Grundlegendes zum System Volkszaehler	23
5.2 Einzelne Module des Volkszaehlers	23
5.3 Benutzung des Volkszaehlers	24
5.4 Installation des Volkszaehlers	24

5.4.1	Voraussetzungen.....	24
5.4.2	Installation der benötigten Module.....	25
5.4.3	Aktualisierung der Middleware einschließlich des Frontends	26
6	Datenbank des Smart-Metering Volkszaehler	27
6.1	Datenbankstruktur in MySQL	27
6.1.1	MySQL-Verbindung.....	27
6.1.2	MySQL-Datenbank erstellen.....	28
6.1.3	MySQL-Tabellen erstellen	28
6.2	Authentifizierung	30
6.3	Administration in der Datenbank	32
6.3.1	Konfiguration der phpmyadmin.....	32
6.3.2	Funktion der phpmyadmin	34
7	Messdaten von mehreren Knoten auf dem Volkszaehler visualisieren und aufteilen	36
7.1	Netzschaltung der mehreren Knoten aufzubauen.....	36
7.2	Aufteilung der Messwerte von mehreren Knoten	37
8	Zusammenfassung und Ausblick.....	40
9	Abkürzungsverzeichnis.....	41
10	Abbildungsverzeichnis.....	42
11	Tabellenverzeichnis.....	43
12	Quellen- und Literaturverzeichnis.....	44

1 Motivation und Zielsetzung

Um Energieverbrauch als Web-Oberfläche im Gebäudemanagement zu visualisieren, verwendet man ein Messcluster, das NAS-8 heißt. Grundsätzlich ist NAS-8 ein Messgerät, das als Stromzähler verwendet wird, um z.B. den Stromverbrauch in einem Kommunikationsnetz zu senden.

Zur Nutzung und Vermarktung lokal erzeugter Energie kommt zuerst ,

- Zentrales Monitoring und Steuerung mit Linux SoC-Systemen von Windkraftanlagen, PV-Feldern und Energiespeichern

Anbei die Gehäuse des NAS Gerätes, welches mittels eines kleinen Rechners sowohl Energiedaten (Strom, Spannung, Leistung etc.) als auch weitere Daten, die an einen Mod-Bus angeschlossen sind und aufnehmen kann. Danach werden die gespeicherte Messdaten durch Ethernet an PC übertragen, hier sollen diese Daten dann in einer Datenbank abgelegt werden und visualisiert werden . Endstufe ist eine Schinttstelle zwischen die Messwerte und Web-Oberfläche aufzubauen, damit die Messwerte von mehreren Geräte grafisch dargestellt werden kann, und das Energieverbrauch der mehreren Anlagen dadurch kontrolliert wird. Außerdem wird die Werke unter dem Linux Betriebssystem durchgeführt und mit Python Programm beschäftigt. Folgende Abschnitte werden über die Schritte und Prinzip konkret darstellen.

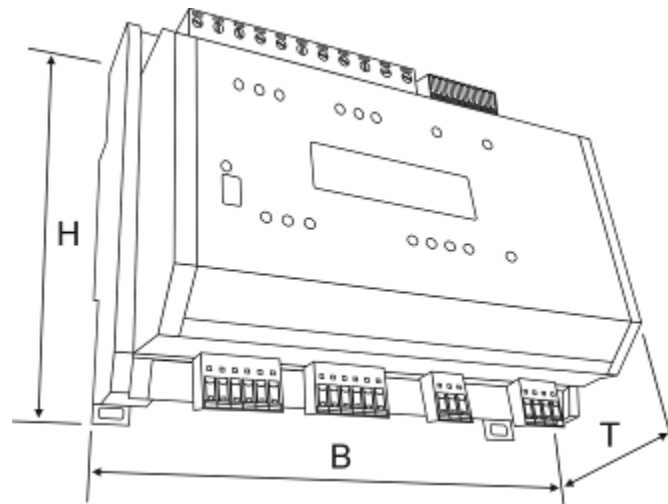
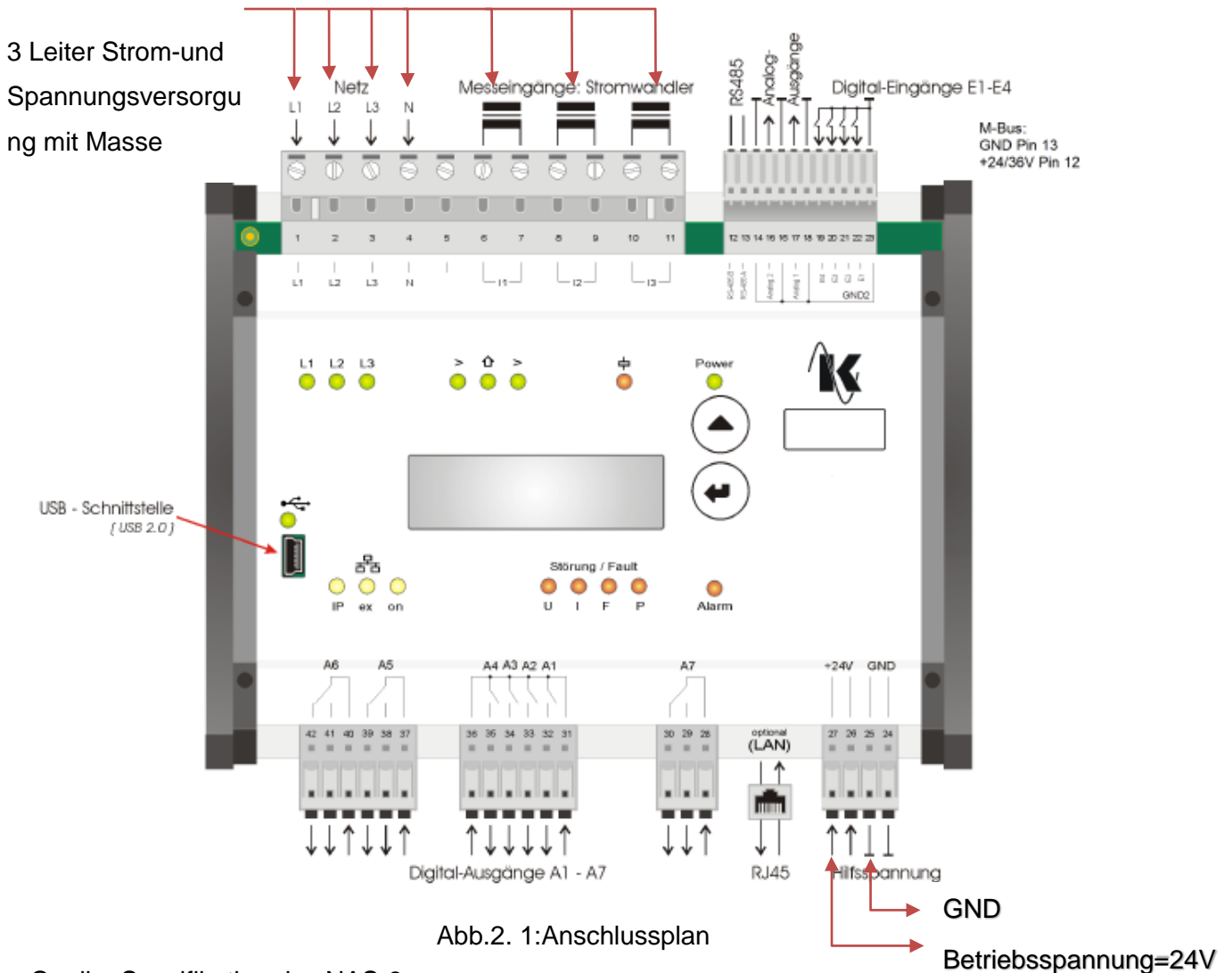


Abb.1. 1: äußere Form des LENA-Messgerätes (NAS-8)

Quelle: eigene Abbildung

2 Struktur und Funktion des LENA Gerätes



Quelle :Spezifikation des NAS-8

2.1 Technische Daten

Betriebsspannung:	24V DC (18...36V), galvanisch getrennt
Absicherung:	4A träge
Leistungsaufnahme:	ca. 8W bei 24V
Messbereich Spannung:	ca. 10 bis 280 V AC Sternpunktspannung ca. 16 bis 484 V AC Außenleiterspannung 35,0 bis 75,0 Hz besser 0,2 % vom Endwert / Klasse 0,2
Messbereich Frequenz:	Frequenzerkennung ab ca. 10 V Sternpunktspannung, 35,0 Hz bis 75,0 Hz besser 0,01 Hz absolut

Messbereich Strom:	Ca. 150 mA bis 6000 mA, 35,0 bis 75,0 Hz besser 0,5 % vom Endwert / Klasse 0,5
Digitale Eingänge:	4 Stück mit gemeinsamer Masse LowActive (Kontaktspannung 12 V 5mA DC, optoentkoppelt) Leitungen nicht länger als 3 m
Relaisausgänge:	230 V / 50 Hz / 2 A - 4 x Schließer mit gemeinsamer Wurzel - 3 x neutrale Wechsler

Tabelle 2. 1 Technische Daten

Quelle: Spezifikation des NAS-8

2.2 Messung

Es werden die Spannungen aller 3 Phasen Sternpunkt bezogen, die 3 Phasenströme, Frequenzen, und Phasenwinkel gemessen. Die Messfunktion ist eine echte Effektivwertmessung und erfolgt als simultane 3-phasige Abtastung mit 32 Werten je Periode bei 12 Bit Auflösung. Die Wirkleistung, Blindleistung, Scheinleistung und Außenleiterspannung werden aus diesen Werten berechnet.

2.2.1 Spannungs- und Strommessung

Für die Strommessung sind immer externe Stromwandler erforderlich

Die Strom und Spannung, die die Strom- und Spannungsversorgung liefert, beträgt sich 0.2A, 24V. Die Werte sind von uns eingestellt .

Die Spannungs- und Strommessung ist eine echte Effektivwertmessung. Es werden alle sechs Messpfade simultan mit 32 Abtastungen je Periode gemessen. Die Spannungsmessung arbeitet bis zu einer Spannungsuntergrenze von ca. 10V. Sobald eine Messspannung erkannt wurde, leuchtet die LED der jeweiligen Phase.

2.2.2 3-Leiter + N - Netze und Leistungsberechnung

Durch die Wahl des Messverfahrens kann mit oder ohne Sternpunkt gemessen werden. Bei der Messung ohne Sternpunkt braucht man nicht notwendig einen Nullleiter anschließt. Bei der 3-Leiter + N-Messung ist durch eine spezielle interne Beschaltung der Klemmen der Wegfall des Nullleiters erkennbar und wird in Form von Spannungsasymmetrie oder Unterspannung Lx angezeigt.

```
star_point_voltage_L1N = 10
star_point_voltage_L2N = 20
star_point_voltage_L3N = 30
conductor_current_L1 = 1066
conductor_current_L2 = 1532
conductor_current_L3 = 2043
```

Abb.2. 2: Messdaten

Aus den Strangspannungen und Leiterströmen wird die Leistung berechnet.

Die Berechnung der Wirkleistung erfolgt nach der Formel:

$$P = \int_0^t u(t) * i(t) dt$$

Die Wirkleistung wird also aus der Summe der Momentanleistungen über ein Zeitintervall bestimmt.

conductor_current_L1=1066

conductor_current_L2=1532

conductor_current_L3=2043

Die Stromwerte sind so ausgegeben. Die Einheit der Werte ist mA, aber bei der Leistung immer kw (wegen großes Verbrauchs von Haushalt), deswegen nur die Anzahlen zu vergleichen sind.

hier die Wirkleistung $P = U_{L1}I_{L1} + U_{L2}I_{L2} + U_{L3}I_{L3} = 14.8V * 1066mA + 21.3V * 1532mA + 33.5V * 2043mA = 116.9W$



Abb.2. 3 Displayanzeige

Quelle: eigene Abbildung

Die Messwerte der Wirkleistung werden auf die parametrisierte Hauptanzeige angezeigt, die und die oben berechnete Werte fast gleich sind.

2.3 Datenspeichern

Dieses Messgerät spielt noch eine wichtig Rolle, die Messwerte zu speichern.

Um die gemessene Werte zu speichern, kommt ein im Messgerät stehenden Miniatur-PC, das Beagle-Bone-PC genannt wird.

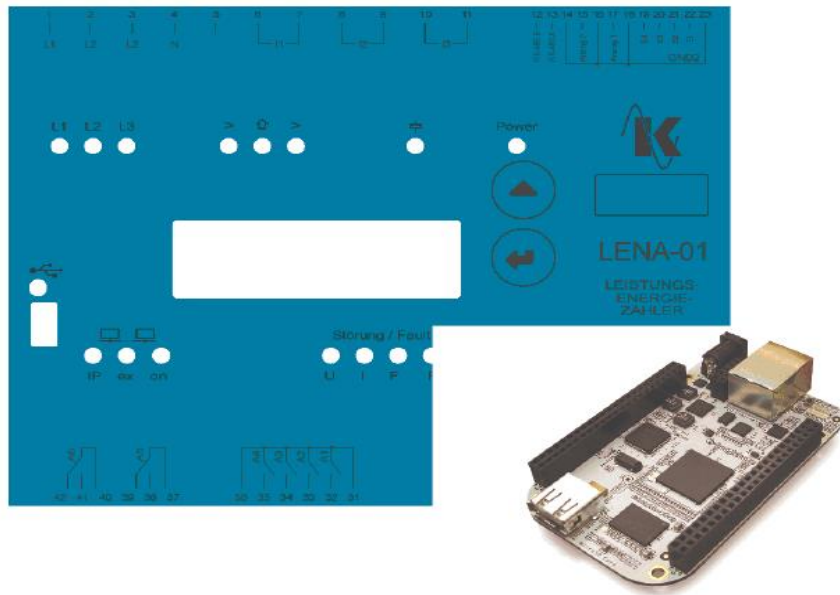


Abb.2. 4: LENA Gerät und Beaglebone-PC

Quelle: eigene Abbildung

3 Embedded System-- Beaglebone Black

Das BeagleBoard ist ein kostengünstiger Einplatinencomputer, Miniatur-PC.

3.1 Geschichte und Entwicklung von BBB

Modell	Prozessor	Speicher RAM	Video/Audio Out	USB	Onboard Speicher	Onboard Netzwerk
BeagleBoard 2008	OMAP3530,720MHz	DDR 128 oder 256MB	DVI-D, S- Video	1 x Standard USB Host 1x mini AB OTG	256MB NAND Flash, SD/MM C	-
BeagleBoard- xM 2010	DM3730,1GHz	DDR2 512MB	DVI-D, S- Video	4 x USB A Host 1x mini	microS D	Fast Ethernet

				AB OTG		
BeagleBone	AM3358/9,720MHz	DDR2 256MB	----	1x USB Standard Host 1x USB B Mini	microSD	Fast Ethernet
BeagleBone Black (die neueste)	AM3358BZCZ100, 1GHz	DDR3 512 MB	Micro- HDMI	1x USB Standard Host 1x USB B Mini	microSD	10/100Mbit/ s RJ45

Tabelle 3. 1 Entwicklung

Quelle: <https://de.wikipedia.org/wiki/BeagleBoard>

Im Vergleich zu den vorherigen BeagleBoard und BeagleBoard-xM, hat BeagleBone eine wesentliche Verbesserung, dass GPIO auf beiden Seiten gestellt sind. Solches Design bringt eine große Vorteile zur Entwicklung der Hardware plug in Board von BBB. Diese Hardware-Plug können direkt im auf beiden Seiten stehenden GPIO eingesteckt werden und sich überschneiden werden. Maximal bis zum 4 Plugs können sich kontinuierlich überlagern. Dies fördert die B-produktfamilie, weiter zu verbreiten, und die Gemeinschaft zu entwickeln und verstärken.

Bei dem Beaglebone Black handelt es sich um eine Entwicklungsplattform, die einen günstigen „Sitara“ Prozessor AM3358 nutzt, einen Singlecore ARM Cortes-A8 von Texas Instruments. BeagleBone Black, mit dem wir beschäftigt haben, ist die neueste. Folgende wichtige Komponente werden festgestellt.

3.2 Bestandteile der Hardware

Dieser Abschnitt beschreibt die wichtigsten Komponenten an Board ,es liefert Informationen über die Lage und die Funktion, um die verschiedenen Komponenten an Board kennenzulernen.

3.2.1 Kernkomponenten

Die Abbildung 3.2.1 zeigt die Lage der Kernkomponenten, die auf dem PCB-Layout des Boardes angelegt sind.

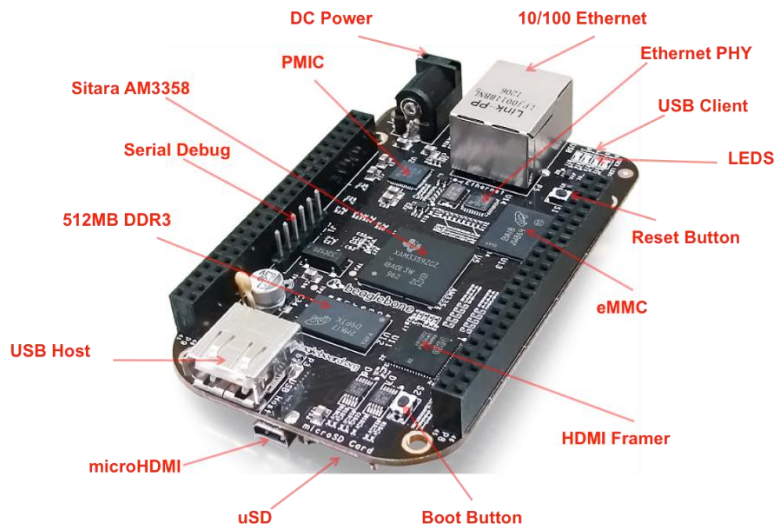


Abb.3. 1 :Kernkomponenten

Quelle: <http://beagleboard.org/support/bone101>

3.2.2 Stecker, LEDs und Schalter

Abbildung 3.2.2 zeigt die Lage der Stecker, LEDs und Schalter auf dem pcb - Layout .

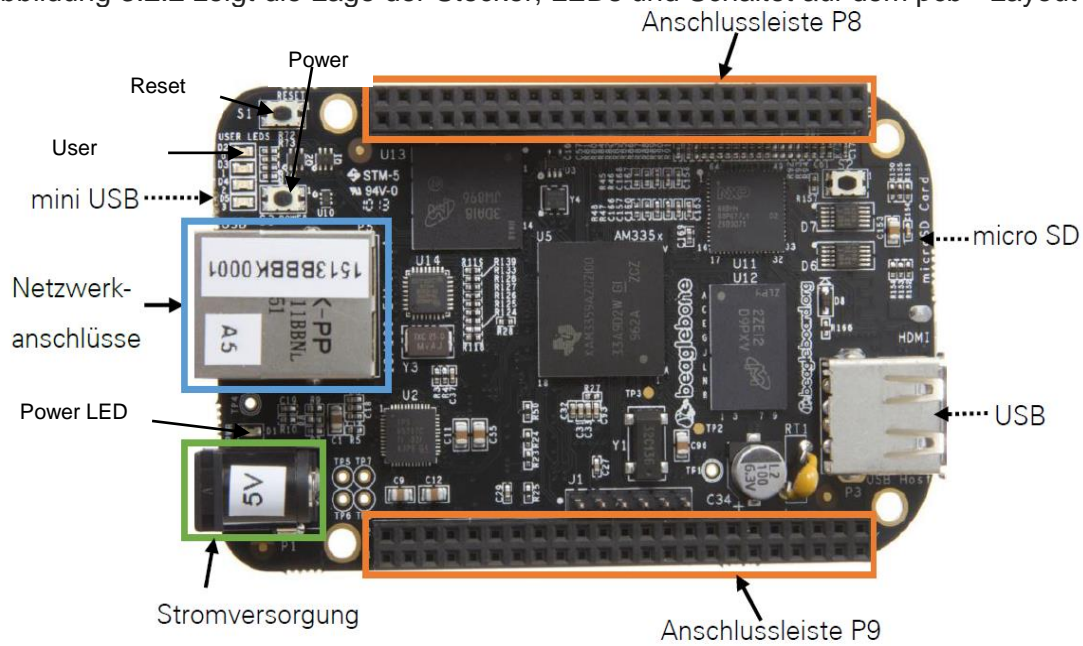


Abb.3. 2 : Anschlüsse auf dem Beaglebone Black

Quelle: <https://beagleboard.org/black>

•DC Power ist ein haupt DC input mit 5V Power.

- Power Button warnt das Prozessor dem fallenden Power.
- 10/100 Ethernet mit LAN verbinden.
- USB Client ist ein miniUSB, das den Board versorgt, mit dem PC verbinden
- Vier blaue LEDs , die durch den Anwender benutzt werden können.
- Reset Taste , durch die kann der Anwender den Prozessor zurücksetzen.
- microSD ist wo die Karte installiert werden kann.
- USB Host kann mit verschiedene USB Interfaces z.b. Wi-Fi, BT, Keyboard, usw.
- GPIO , Input/output Schnittstelle(maximal 69)

3.3 Datenübertragen

Das Messsystem verfügt über eine Ethernetschnittstelle, um die gemessenen Daten an einen zentralen Rechner zu übertragen.

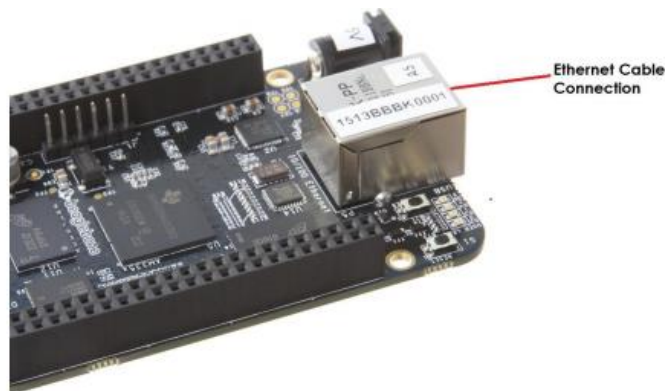


Abb.3. 3 Ethernet Kabel Verbindung

Quelle: cdn.sparkfun.com/datasheets/Dev/Beagle/BBB_SRM_C.pdf S24

Da die LENA-CPU und der BBB-PC bereits durch den Modbus angeschlossen worden sind, schliesst man das Beaglebone Black nicht direkt per USB-Kabel am PC an , sondern mithilfe des Ethernet unter LAN am PC anschliesst.

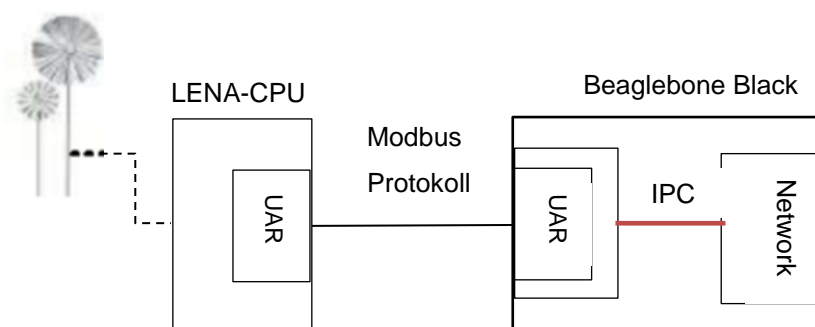


Abb.3. 4 zwischen LENA und BBB

Quelle : eigene Abbildung

4 Durchführung mit Linux SoC-Systemen

Um die Schnittstelle aufzubauen und Messdaten auszuwerten, braucht man unter Linux Betriebssystem bearbeiten. Linux ist ein neue Betriebssystem für mich, mit dem ich kaum beschäftigt habe.



Abb.4. 1 Linux

Quelle: <https://de.wikipedia.org/wiki/Linux>

Linux ist ein kostenloses Betriebssystem und man kann es sich selbst online herunterladen. Es gibt auch viele Varianten, berühmte Auflagen sind Ubuntu, Debian und CentOS. Die Oberfläche von Linux wirkt für uns Windows-Nutzer auf den ersten Blick verwirrend. Aber das ist wohl nur Gewohnheit.

4.1 Vergleich der Windows und Linux

- ✓ Windows ist mit fast jedem Programm dieser Welt kompatibel. Da es das am meisten verbreitete Betriebssystem dieser Welt ist, wird nahezu jedes Programm auch für Windows entwickelt.
- ✓ Spiele laufen heutzutage generell fast alle auf Windows. Daher werden nur wenige Spiele für Linux-Distributionen programmiert und unterstützt.
- ✓ Ubuntu ist eine Linux-Distribution, die sich auf Nutzerfreundlichkeit und Kompatibilität spezialisiert hat. Damit ist Ubuntu bei Privatanwendern das am meisten genutzte Linux weltweit.
- ✓ Ubuntu ist klar strukturiert und einfach gehalten. Sie finden mit wenigen Mausklicks jedes gewünschte Programm und jede Einstellung. Sie können Ubuntu ganz nach Ihren Wünschen gestalten und die Oberfläche ist nicht so restriktiv wie die von Windows.

- ✓ Sicherheit ist eine der größten Argumente für die Nutzung von Ubuntu.
- ✓ In Ubuntu ist alles vorinstalliert, was Sie sich wünschen könnten.

4.2 Linux-Distribution Ubuntu installieren

Ich habe versucht, das Ubuntu 16.04 auf dem PC neben Windows parallel zu installieren, damit das Experimente zu Hause durchgeführt werden kann.

4.2.1 Vorbereitung der Installation

- Laden Sie Ubuntu 16.04 LTS ISO-Datei herunter.
Laden Sie die ISO-Datei von Ubuntu 16.04 LTS von der offiziellen Website herunter.
- Wählen Sie "Ubuntu installieren", um die Installation zu starten.

4.2.2 Auswahl der Partition bzw. Partitionierung und Bootloader

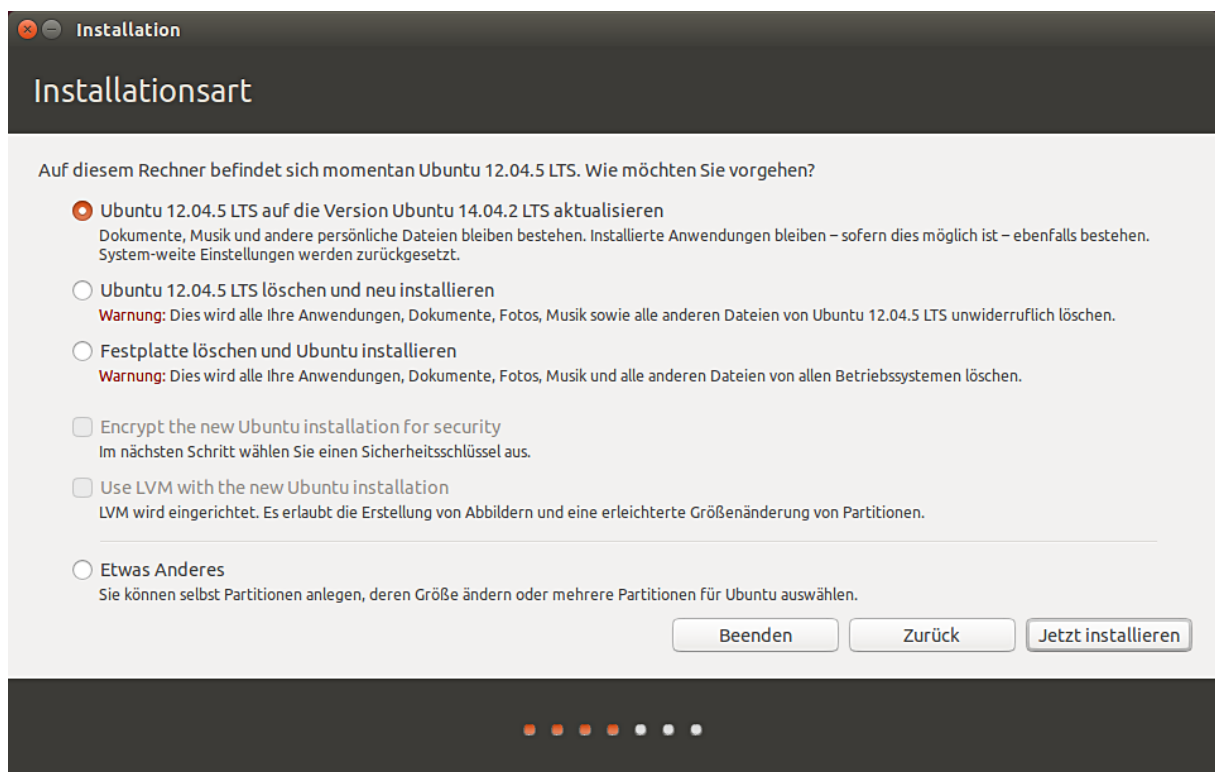


Abb.4. 2 Installationsart

Quelle: https://wiki.ubuntuusers.de/Ubuntu_Installation/

- Möchte man die Partitionierung der Festplatte selbst vornehmen, wählt man aber den Punkt "Etwas Anderes". Das kann v.a. dann angebracht sein, wenn Ubuntu neben einem anderen schon vorhandenen Betriebssystem zu installieren ist.
- Klicken Sie auf Neue Partitionstabelle.

- Klicken Sie im folgenden Bildschirm auf die Diskette. Klicken Sie auf die Option '+', um eine Partition zu erstellen.

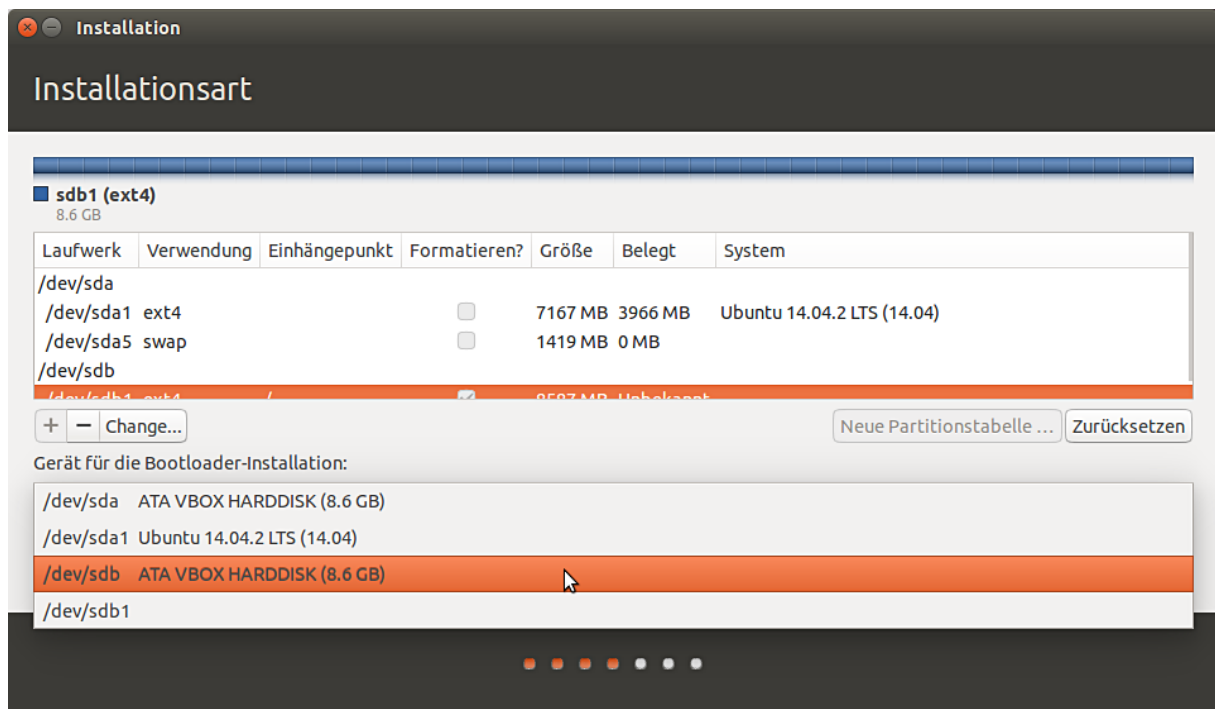


Abb.4. 3 Partitionierungs-Tool mit angelegten Partitionen

Quelle: https://wiki.ubuntuusers.de/Ubuntu_Installation/

Zum Bearbeiten der Partitionen wählt man durch Anklicken die betreffende Partition oder einen geeigneten freien Bereich angewählt, und es können folgende Operationen ausgeführt werden:

mit einem abschließenden "Ok" die nötigen Einstellungen vorzunehmen. Dieser Schritt ist so oft zu wiederholen, bis alle benötigten Partitionen angelegt sind.

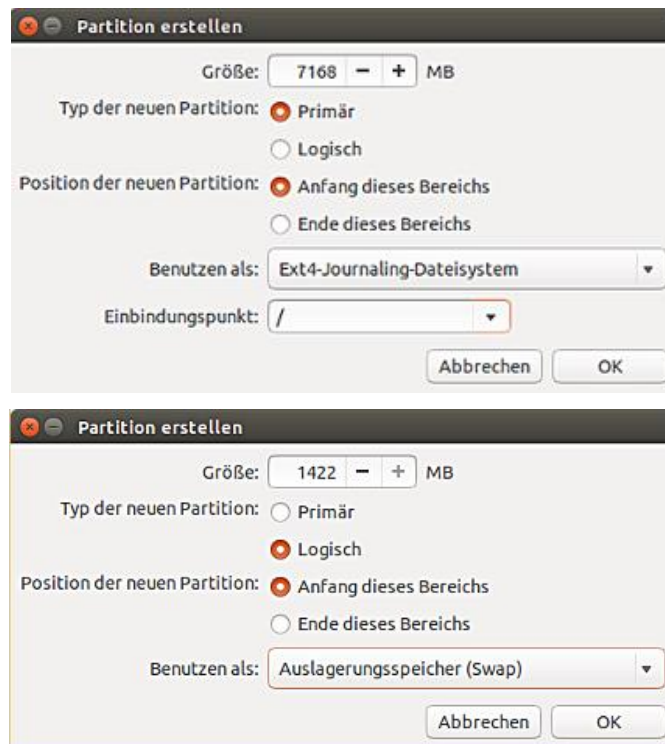


Abb.4. 4 Partition im Format *ext4* und *swap*

Quelle: https://wiki.ubuntuusers.de/Ubuntu_Installation/

Sobald Sie mit der Partitionstabelle fertig sind, klicken Sie auf ' Jetzt installieren ', klicken Sie auf Weiter, um fortzufahren.

4.2.3 Zeitzone, Tastaturlayout und Benutzerdaten

Grundlegende Einstellungen für das System und den Hauptbenutzer werden eingestellt.

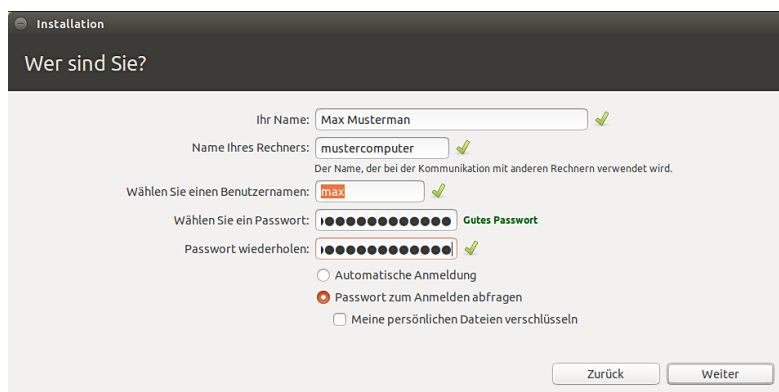


Abb.4. 5 Nutzerdaten, u.a. Name und Login-Passwort

Quelle: https://wiki.ubuntuusers.de/Ubuntu_Installation/

Nun müssen einige Daten zum Hauptbenutzer des Computers eingegeben werden. Dieser Nutzer kann nach Eingabe seines Passwortes wichtige Systemeinstellungen verändern.

4.2.4 Laufende Installation und Neustart

Im nächsten Installationsdialog wird die Installation vollständig durchgeführt und der Fortschritt derselben dabei angezeigt. Ist die Installation abgeschlossen, so muss man den Rechner einmal neu starten.



Abb.4. 6 Fenster-Menüzeile in Ubuntu

Quelle: <https://www.pcwelt.de/tipps/Fenster-Menuezeile-in-Ubuntu-immer-anzeigen-9790425.html>

4.3 Interne Kommunikation im Beaglebone Black

Die Funktion des Beaglebone Black ist die Speicherung und Übertragung der Messwerte aus dem LENA-Gerät. Das LENA-Gerät misst zuerst die Strom- oder Spannungswerte, danach sendet es diese Werte mithilfe des Modbus an das Beaglebone Black. Das interne verwendete Protokoll ist an das Modbus Protokoll angelehnt. Es ist ein Master-Slave Protokoll ohne weiteren Handshake.

Im Inneren des BBB gibt es einige Software, um die Speicherung und Übertragung der Messwerte zu realisieren. Diese Software läuft gegenseitig vom Modbus bis zur IPC. Mit solchen Linux Befehle kann man die Struktur der Software-Komponenten , die eigentlich innerhalb des BBBs aufgestellte Python-Programme sind, deutlich ersehen.

```

sstudent@student-15 ~ $ ssh root@192.168.10.223
root@192.168.10.223's password:
Linux bbb-koralewski 3.8.13-bone68 #1 SMP Sat Nov 22 02:12:03 UTC 20
14 armv7l
#####
#
#   Welcome on board LENA-KMA bei Koralewski   #
#
#####

Last login: Wed Oct  4 17:10:16 2017 from student-15
root@bbb-koralewski:~# ls
activeUART.sh  install          lwire           socieerMsg
etc            installed_packets  packages        socieerWSGI
git           labor.py         socieerIPC      uselesscode
hause.py      log             socieerLENA     zaehlerstand
root@bbb-koralewski:~# cd socieerLENA/
root@bbb-koralewski:~/socieerLENA# ls
analyse.py      configure.pyc      minimodbus.py    test_cl.bak
__init__.py    lenainstrument.py  minimodbus.pyc  test_cl.py
appurtenance.py  lenainstrument.py  socieerLENA.py  test.py
appurtenance.pyc  lenainstrument.pyc  socieerLENA.pyc  test.pyc
configure.py    lena_output        test.bak         test_sluh.py
root@bbb-koralewski:~/socieerLENA# █

```

Abb.4. 7 Interner BBB mit Linux Terminal-Befehl

Quelle: eigene Abbildung

Aus der Abbildung ist darzulegen, dass die Software aus den Dateien „socieerIPC“, „socieerMsg“, „socieerWSGI“ und „socieerLENA“ besteht, die Datei „socieerLENA“ besteht aus den angezeigten Python-Programme.

LENA--Software	socierIPC	socierIPC.py
	socieerMsg	socieerMsg.py
	socierWSGI	socierWSGI.py
	socieerLENA	analyse.py
		appurtenance.py
		configure.py
		lenainstrument.py
		minimodbus.py
		socieerLENA.py
		test.py
test_cl.py		
test_sluh.py		

Tabelle 4. 1 Software der LENA

Quelle: eigene Abbildung

Der Typ „socieerLENA“ ist ein Testeffekt, der Messdaten misst und sammelt. Der Typ „IPC“ ist ein Werkzeug, der als Pipe bezeichnet wird, mit dem Prozess und Netzwerk verbunden sind, Der Typ „WSGI“ ist eine Datenbank, um Messdaten zu speichern und anzuzeigen.

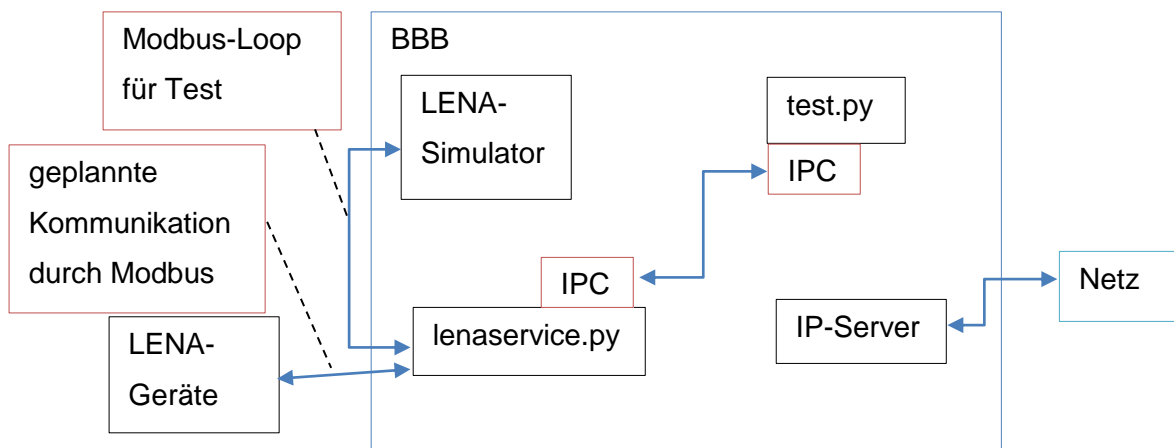


Abb.4. 8 Blockdiagramm der Kommunikation

Quelle: eigene Abbildung

4.3.1 Programm „socieer IPC“

Der Begriff Interprozesskommunikation meint in der Informatik verschiedene Verfahren des Informationsaustausches zwischen den Prozessen eines Systems.

Bei einer Message Queue dagegen werden Nachrichten von einem Prozess an eine Nachrichtenschlange geschickt, von wo diese von einem anderen Prozess abgeholt werden kann. Hier entsteht IPC zwischen die Python Programme und Netz .

```
def send(self, m): """Serialize message (eine Folge der Information) m
and sends it"""
```

```
    logging.debug(" * Sending message")
```

“ os” “error” logging” ist die schon definierte Bibliothek

.....

```
def recv(self):
```

"""Receives data deserialize, and return message object.

Reading is non-blocking. Return none if there were no message"""

```
    logging.debug(" * Receiveing message")
```

```
try:
```

```
    buf = os.read(self.__fr, self.max_message_length)
```

```
    message = self.cls.deserialize(buf)
```

```
    return message
```

```
except OSError, e:
```

```
    if e.errno == errno.EAGAIN:
```

```
logging.debug("# No message: EAGAIN")
return None
```

4.3.2 Programm LENA „appurtenance.py“

Diese Python Programm kann die Messwerte ausgeben

Funktionscode	Kurzname	Funktion
0	-----	Timestamp
65	FKT_CYCLIC_VALUE	Zustände zyklisch holen
66	FKT_ACTUAL_VALUE	Aktuelle Messwerte holen
67	FKT_CONFIG_GET	Konfiguration lesen
68	FKT_CONFIG_SET	Konfiguration schreiben
69	FKT_TIME_GET	Zeit lesen
70	FKT_TIME_SET	Zeit setzen
71	FKT_STATUS_SET	Status setzen

Tabelle 4. 2: Funktionsüberblick aus dem Programm „lenainstrument.py“

Quelle: LENA_01 BB Kommunikation_004 .doc

Die Funktion 65 holt den aktuellen Statuscode und einen Aktualisierungszähler von der LENA-CPU in den BB-PC.

```
if fktNr == 65:
    return "status_code = {0!r}\n" \
           "update_counter = {1!r}\n".format(payload[0], payload[1])
```

Längenbyte ist die Anzahl der gesamten Master oder Slave Byte.

Master:

0×01	Adressbyte
0×41	Funktionsbyte
0×05	Längenbyte(5 Byte)
0×??	2 Byte CRC
0×??	

Slave:

0×01	Adressbyte
0×41	Funktionsbyte
0×0B	Längenbyte (11 Byte)
0×??	Statuscode
0×??	4 Byte
0×??	Bit 0 bis 10 für Funktion
0×??(Low)	FKT_ERROR_GET110

	Bis FKT_ERROR_GET110
0x?? 0x??(Low)	Aktualisierungszähler 2 Byte
0x?? 0x??	2 Byte CRC

Tabelle 4. 3 FKT_CYCLIC_VALUE 65(0x41)

Quelle: LENA_01 BB Kommunikation_004 .doc

Die Funktion 66 holt die aktuellen Hauptmesswerte von der LENA_CPU in den BB-PC.

if fktNr == 66:

```

return "active_power = {0!r}\n" \
      "apparent_power = {1!r}\n" \
      "reactive_power = {2!r}\n" \
      "cosine_phi = {3!r}\n" \
      "star_point_voltage_L1N = {4!r}\n" \
      "star_point_voltage_L2N = {5!r}\n" \
      "star_point_voltage_L3N = {6!r}\n" \
      "conductor_current_L1 = {7!r}\n" \
      "conductor_current_L2 = {8!r}\n" \
      "conductor_current_L3 = {9!r}\n".format(payload[0], payload[1], payload[2], payload[3],
      payload[4], payload[5], payload[6], payload[7],
      payload[8], payload[9])

```

Master:

0x01	Adressbyte
0x41	Funktionsbyte
0x05	Längenbyte(5 Byte)
0x?? 0x??	2 Byte CRC

Slave:

0x01	Adressbyte	
0x42	Funktionsbyte	
0x0B	Längenbyte (43 Byte)	
0x?? 0x?? 0x?? 0x??(Low)	Wirkleistung P	Aktuelle Blindleistung in Watt
0x?? 0x?? 0x?? 0x??(Low)	Scheinleistung S	Aktuelle Blindleistung in VA

0x?? 0x?? 0x?? 0x??(Low)	Blindleistung Q	Aktuelle Blindleistung in Var
0x?? 0x??(Low)	CosPhi	Aktuelle CosPhi
0x?? 0x?? 0x?? 0x??(Low)	Sternpunktspannung L1-N	Aktueller Spannung L1-N in ganzen V
0x?? 0x?? 0x?? 0x??(Low)	Sternpunktspannung L2-N	Aktueller Spannung L2-N in ganzen V
0x?? 0x?? 0x?? 0x??(Low)	Sternpunktspannung L3-N	Aktueller Spannung L3-N in ganzen V
0x?? 0x?? 0x?? 0x??(Low)	Leiterstrom L1	Aktueller Leiterstrom L1 in mA
0x?? 0x?? 0x?? 0x??(Low)	Leiterstrom L2	Aktueller Leiterstrom L2 in mA
0x?? 0x?? 0x?? 0x??(Low)	Leiterstrom L3	Aktueller Leiterstrom L3 in mA
0x?? 0x??	2 Byte CRC	2 Byte CRC

Tabelle 4. 4 FKT_CYCLIC_VALUE 66(0x42)

Quelle: LENA_01 BB Kommunikation_004 .doc

Die Funktion 71 setzt den Status in der LENA-CPU. Es sind folgende Werte beschreibbar.

if fktNr == 71:

```

return "is_ip = {0!r}\n" \
       "is_network = {1!r}\n".format(int(payload[0]), int(payload[1]))

```

Master:

0x01	Adressbyte
0x47	Funktionsbyte
0x23	Längenbyte (35 Byte)
0x?? 0x??(Low)	Statuswort

0x?? 0x?? 0x?? 0x??(Low)	Leistungssollwertvorgabe
0x?? 0x?? 0x?? 0x??(Low)	CosPhi Sollwertvorgabe
0x?? 0x?? 0x?? 0x??(Low)	Reservewort 1
0x?? 0x?? 0x?? 0x??(Low)	Reservewort 2
0x?? 0x?? 0x?? 0x??(Low)	Reservewort 3
0x?? 0x?? 0x?? 0x??(Low)	Ein-Ausgangssteuerwort
0x?? 0x?? 0x?? 0x??(Low)	Duratation
0x?? 0x??	2 Byte CRC

Slave:

0x01	Adressbyte
0x47	Funktionsbyte
0x05	Längenbyte (5 Byte)
0x??	2 Byte CRC
0x??	

Tabelle 4. 5 FKT_CYCLIC_VALUE 71(0x47)

Quelle: LENA_01 BB Kommunikation_004 .doc

- Statuswort(2 Byte, 16Bit)
- Leistungssollwert in 0/oo (signed short, 2 Byte, 16 Bit)
- CosPhi Sollwert(signed short, 2 Byte, 16 Bit)
- Reservewort 1 (signed int, 4 Byte, 32 Bit)
- Reservewort 2 (signed int, 4 Byte, 32 Bit)
- Reservewort 3 (signed int, 4 Byte, 32 Bit)
- Ein-/Ausgangswort 3 (unsigned int, 4 Byte, 32 Bit)
- Duratation (signed int, 4 Byte, 32 Bit)-0=don't care

4.3.3 Programm „test.py“

Die Funktion des Programmes hängt vom Kommunikationsablauf ab. Die Kommunikation zwischen dem BBB-PC und der LENA-CPU läuft zyklisch und azyklisch ab. Es werden kontinuierlich, mindestens einmal pro Sekunde folgende Telegramme abgerufen:

```
FKT_CYCLIC_VALUE 65(0x41)
FKT_ACTUAL_VALUE 66(0x42)
FKT_STATUS_SET   71(0x47)
```

Weitere Telegramme werden azyklisch dazwischen abgearbeitet.

Die Telegrammfolge sieht dann beispielweise folgendermaßen aus:

```
65, 66, 71, xx, xx, xx, 65, 66, 71, xx, 65, .....
```

Durch die Programmierung wird die Schleife realisiert.

```
self.th.start()
logging.warning('<#> run the lena test module using the test methode()')
ct = 0
while True:
    try:
        self.mBackup.backup_timestamp(format_payload(0, time.time()), description="timestamp"), rewrite=True)
        response = self.dev.getCyclicvalue()
        self.m1.mtype = 65
        self.write(str(response))
        logging.warning('<#> Function 65 with response: %s\n', response)
        self.m1.payload = (';'.join(str(v) for v in response))
        self.msgr_c.send(self.m1)
        self.mBackup.backup_fkt_cycle_value(format_payload(65, response, description="fkt_get_cycle_values"))
        # Funktion 65 zyklisch ablaufen

        response = self.dev.getActualvalue()
```



```

self.m1.mtype = 66
self.write(str(response))
logging.warning('<#>Function 66 with response: %s\n', response)
self.m1.payload = (';'.join(str(v) for v in response))
self.msgr_c.send(self.m1)
self.mBackup.backup_fkt_actual_value(format_payload(66, response, descrip-
tion="fkt_get_actual_values"))
# Funktion 66 zyklisch ablaufen

self.lsl[0] = analyse.getNetworkstatusbits(self.ns.isIP, self.ns.isNetwork)
self.mBackup.backup_network_status(format_payload(71, [self.ns.isIP,
self.ns.isNetwork], description="network status"))
response = self.dev.setSystemstatus(self.lsl)
logging.warning('<#> Function 71 with response: %s\n', response)
# Funktion 71 zyklisch ablaufen

except (ValueError, TypeError, IOError) as ErrorString:
    logging.error('* Error info: {0!r}'.format(ErrorString))
else:
    pass
ct = ct + 1

```

Die aktuelle Messdaten sind durch den Befehl „tail_f lena_output“ auszugeben.

```
timestamp = '04-10-2017,17:19:26'  
status_code = 2048  
update_counter = 28552  
active_power = 117401  
apparent_power = 116  
reactive_power = 0  
cosine_phi = 0  
star_point_voltage_L1N = 10  
star_point_voltage_L2N = 20  
star_point_voltage_L3N = 30  
conductor_current_L1 = 1066  
conductor_current_L2 = 1532  
conductor_current_L3 = 2043  
is_ip = 1  
is_network = 1  
tail: lena_output: Datei abgeschnitten  
timestamp = '04-10-2017,17:19:27'  
status_code = 2048  
update_counter = 28555  
active_power = 117401  
apparent_power = 116  
reactive_power = 0  
cosine_phi = 0  
star_point_voltage_L1N = 10  
star_point_voltage_L2N = 20  
star_point_voltage_L3N = 30  
conductor_current_L1 = 1065  
conductor_current_L2 = 1532  
conductor_current_L3 = 2043  
is_ip = 1  
is_network = 1
```

Abb.4. 9 Ausgabe der Werte der „lena_output“

Quelle: eigene Abbildung

4.4 Auswerten über Frontend

Nachdem die Messdaten durch das Python Programm ausgegeben werden sind, werden die über Frontends grafisch dargestellt, damit man den Energieverbrauch deutlich ersehen kann. Hier sollen diese Daten dann in einer Datenbank abgelegt werden und visualisiert werden.

5 Visualisierung durch Smart-Metering Volkszaehler

5.1 Grundlegendes zum System Volkszaehler

Volkszaehler ist ein intelligenter Zähler (auch Smart Meter genannt), der dem Nutzer seinen Energieverbrauch transparent machen soll. Durch eine zeitnahe Rückmeldung soll ein sparsamer Umgang mit Energie gefördert werden. Detaillierte Lastprofile erlauben dem Verbraucher eine gezielte Optimierung seines Energieverbrauchs.

volkszaehler.org ist ein freier intelligenter Stromzähler im Selbstbau, bei dem die anfallenden Stromprofile unter der Kontrolle des Nutzers verbleiben. Die Daten des Volkszaehlers sind nicht durch den Versorger auslesbar. Mit einem Materialeinsatz von ca. EUR 100, etwas Geschick und Zeit lässt sich ein solcher Volkszaehler auf Basis eines Standard- μ C-Moduls aufbauen.

Der Volkszaehler kann nicht als fertiges Gerät gekauft werden. Es ist ein Selfmade-Projekt. Der Volkszaehler besteht nicht aus einem einzigen Element sondern aus insgesamt 4 Modulen (Messung, Verarbeitung, Speicherung & Auswertung). Das Modul zur Messung muss in der Regel von einem Elektriker im Schaltschrank der Hausinstallation eingebaut werden. Es gibt auch Varianten die keinen Eingriff in die Hausinstallation benötigen. Alle anderen Module lassen sich von interessierten Tüftlern selber nachbauen.

5.2 Einzelne Module des Volkszaehlers

Der Volkszaehler besteht aus 4 Modulen: Messen, Übertragen, Speichern und Auswerten. Für die meisten dieser Module gibt es verschiedene Möglichkeiten der Umsetzung. Die 4 Anteile sind auch gemäß der obengenannten Schritte der Projekt.

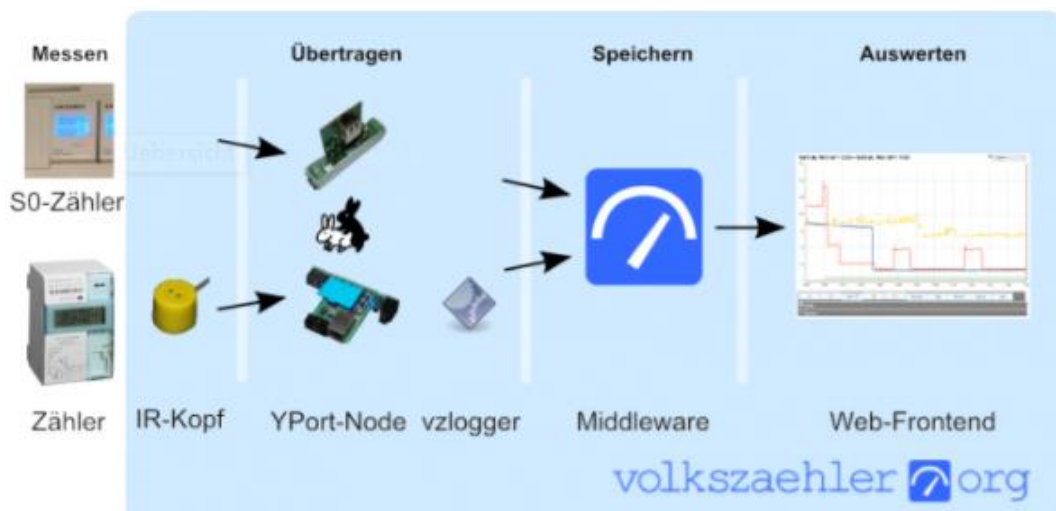


Abb.5. 1 Aufbau des Volkszaehler

Quelle: <https://wiki.volkszaehler.org/start>

5.3 Benutzung des Volkszaehlers

Wer seinen Energiebedarf analysieren möchte, braucht dazu genaue Messwerte. Bei heutigen Hausinstallationen mit konventionellen Drehstromzählern sind für den Stromverbrauch diese Messwerte nicht vorhanden. Es wird also ein intelligenter Zähler benötigt, der in der Lage ist den Energiebedarf über sehr kurze Zeiträume zu messen und zu speichern.

Diese Geräte werden heute schon von vielen Versorgern angeboten. Die einfachen Lösungen ermöglichen die Anzeige der aktuellen Leistung, der Verbrauchswerte des aktuellen Tages, der aktuellen Woche, des aktuellen Monats und der jeweils vorhergehenden Zeiträume. Bei den meisten Lösungen, bei denen darüber hinaus eine komfortable Auswertung möglich ist, werden die Verbrauchsdaten jedoch zum Versorger übertragen. Der Nutzer kann dann i.d.R. über eine Weboberfläche eine komfortable Analyse vornehmen.

Wie schon weiter oben geschildert besteht nun die Möglichkeit für den Versorger auf Grundlage der Daten Nutzerprofile anzulegen und diese auszuwerten.

5.4 Installation des Volkszaehlers

Die Middleware und Frontend sind benötigt zu installieren beim Befehl:
ssh root@192.168.10.113 mit Passwort: root anzumelden



```
sstudent@student-14 ~ $ ssh root@192.168.10.113
root@192.168.10.113's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Oct 6 18:06:45 2017 from student-15
root@volkszaehler:~# ls
ethersex  install.sh  vzlogger
```

Abb.5. 2 Volkszaehler in Server

Quelle: eigene Abbildung

192.168.10.113, die Adresse der Volkszaehler-Server

5.4.1 Voraussetzungen

Die Abhängigkeiten lassen sich schnell mit folgendem Befehl installieren:

```
sudo apt-get install git-core libapache2-mod-php7.0 php7.0-cli php7.0-mysql php-apcu  
mysql-server mysql-client php7.0-xml php-mbstring
```

Datenbank *mysql*, wo die dynamische Daten speichert, und dynamische Skriptspache *php* wurde installiert, die wichtige Datenpakete der Middleware sind.

Install-Skript

```
sudo wget https://raw.githubusercontent.com/volkszaehler/volkszaehler.org/master/misc/tools/install.sh  
  
sudo bash install.sh
```

Das soll dann ablaufen:

```
create volkszaehler.org database user? [y]  
  
creating db user vz with proper rights...  
  
allow channel deletion? [n] y  
  
granting db user vz delete rights...  
  
insert demo data in to database? [n]
```

„volkszaehler.org“ und „data“ sind schon vorhandene database durch den SQL-Befehl „create“ und „insert“.

```
Composer (version 1.5.2) successfully installed to: /var/www/volkszaehler.org/composer.phar
```

Das Script fragt, was es wohin installieren soll.

5.4.2 Installation der benötigten Module

- Zunächst PHP Paketmanager Composer von getcomposer.org installieren

```
cd /tmp  
  
curl -sS https://getcomposer.org/installer | php
```

```
sudo mv composer.phar /usr/local/bin/composer
```

```
sudo chmod +x /usr/local/bin/composer
```

- Konfiguration der Middleware
- Datenbank konfigurieren
- Nun noch ein Passwort für den mysql-Administrator „root“ setzen und die Passwortdatenbank aktivieren:

```
mysqladmin password -u root "<neues, gutes passwort>"
```

```
mysqladmin reload -p
```

5.4.3 Aktualisierung der Middleware einschließlich des Frontends

phpmyadmin installieren

```
sudo apt-get install phpmyadmin
```

phpMyAdmin ist eine freie Webanwendung zur Administration von MySQL-Datenbanken, in der die weitere Arbeit begründet.

```
Last login: Sat Oct 7 11:51:44 2017 from student-14
root@volkszaehler:~# cd /var/www
root@volkszaehler:/var/www# ls
html volkszaehler.org
```

Abb.5. 3 installierter Volkszaehler

Quelle: eigene Abbildung

Der Volkszaehler ist erfolgreich im Server 192.168.10.113 installiert.

6 Datenbank des Smart-Metering Volkszaehler

Die im Volkszaehler-Server installierte Datenbank heißt *MySQL*, die phpMyadmin hauptsächlich auf Webservern verwaltet, *mysql* und *mysqladmin* direkt auszuführen

6.1 Datenbankstruktur in MySQL

Um den Umgang mit Datenbanken, besser gesagt mit relationalen Datenbank wie MySQL zu verstehen, brauchen wir ein paar Grundbegriffe. Die wichtigsten Begriffe in diesem Zusammenhang sind dabei Datenbank, Tabellen, Spalten, Zeilen und Zellen. Eine Datenbank ist eine separate Anwendung, die eine Sammlung von Daten speichert. Jede Datenbank verfügt über eine oder mehrere unterschiedliche Schnittstelle zum Erstellen, Zugreifen, Verwalten, Suchen und Replizieren der Daten, die sie enthält.



Abb.6. 1 Datenbank MySQL

Quelle: <https://www.tutorialspoint.com/mysql/>

Auf dem Datenbankmanagementsystem, dem MySQL-Server, können mehrere Datenbanken erstellt werden. In einer Datenbank können mehrere Tabellen angelegt werden. Praktisch erstellt MySQL dabei für jede Datenbank einen Ordner auf der Festplatte, in dem Dateien für die Struktur und die Daten der einzelnen Tabellen abgelegt werden.

6.1.1 MySQL-Verbindung

Verwenden Sie das Dienstprogramm "mysqladmin", um den Serverstatus zu erhalten. Es wird das folgende Ergebnis unter Linux erzeugen. Es kann abhängig von Ihrer Installation variieren –

```
[root@host]# mysqladmin --version
mysqladmin Ver 8.42 Distrib 5.5.55, for Debian-linux-gnu on x86_64
```

Im Folgenden sehen Sie ein einfaches Beispiel für die Verbindung eines MySQL-Servers von der Befehlszeile aus:

```
root@volkszaehler:~# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4418310
Server version: 5.5.55-0+deb8u1 (Debian)
```

Abb.6. 2 Verbindung mit Mysql-Server

Quelle: eigene Abbildung

Im obigen Beispiel verwenden wir die root-Benutzeranmeldung am mysql-Server. Sie können natürlich auch eine andere mysql-Benutzeranmeldung verwenden. Nachdem das Login erfolgreich ist, können Sie eine beliebige SQL-Anweisung ausführen.

Beenden Sie das Fenster mysql> Eingabeaufforderung. Sie können den Befehl exit wie folgt verwenden:

```
mysql> exit
Bye
```

6.1.2 MySQL-Datenbank erstellen

Da wir als root-Benutzer am mysql-Server anmelden, kann man mit dem Befehl *mysql* *mysqladmin* benutzen, die Datenbank zu erstellen. Ich bezeichne die neue Datenbank als RUNOOB.

```
[root@host]# mysqladmin -u root -p create RUNOOB
Enter password:*****
```

6.1.3 MySQL-Tabellen erstellen

Mit dem SQL Befehl CREATE TABLE wird die Datentabelle erstellt.
SQL-Kommando zum Anlegen einer Tabelle

```
root@host# mysql -u root -p
```


Enter password:*****

```
mysql> use RUNOOB;
```

Database changed

```
mysql> CREATE TABLE runoob_tbl( → SQL-Kommando zum Anlegen einer Tabelle
```

```
-> runoob_id INT NOT NULL AUTO_INCREMENT,
```

```
-> runoob_title VARCHAR(100) NOT NULL,
```

```
-> runoob_author VARCHAR(40) NOT NULL,
```

```
-> submission_date DATE,
```

```
-> PRIMARY KEY ( runoob_id )
```

```
-> )ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Eingeba für unser Beispiel

Query OK, 0 rows affected (0.16 sec)

```
mysql>
```

- USE- Datenbankname :

Wählen Sie die zu betreibende Mysql-Datenbank aus, verwenden Sie den Befehl nach dem Befehl Mysql nur für die Datenbank.

```
mysql > use RUNOOB
```

Database changed

- DATENBANKEN ANZEIGEN:

Listen Sie die Datenbank des MySQL-Datenbankverwaltungssystems auf. Die erstellte Datenbank runoob steht schon darauf.

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| Gebäude 1 |
| mysql |
| performance_schema |
| phpmyadmin |
| runoob |
| ssn |
| volkszaehler |
+-----+
8 rows in set (0.00 sec)
```

Abb.6. 3 Liste aller vorhandenen Datenbanken

Quelle: eigene Abbildung

Zur Kontrolle:

- TABELLEN ZEIGEN

Zeigen Sie alle angegebenen Datenbanktabellen an, verwenden Sie die Reihenfolge, bevor Sie die Reihenfolge verwenden müssen, um die zu verwendende Datenbank auszuwählen. Solche Befehl zeigt die Tabelle und die Eigenschaften der Tabelle

```
mysql> use runoob
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_runoob |
+-----+
| runoob_tbl       |
+-----+
1 row in set (0.00 sec)

mysql> SHOW COLUMNS FROM runoob_tbl;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| runoob_id      | int(11)       | NO   | PRI | NULL    | auto_increment |
| runoob_title   | varchar(100)  | NO   |     | NULL    |                |
| runoob_author  | varchar(40)   | NO   |     | NULL    |                |
| submission_date | date          | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Abb.6. 4 Anzeige der Tabelle

Quelle: eigene Abbildung

Für die Spalten title und author haben wir den Typ varchar gewählt. varchar erlaubt Texteingaben bis maximal 65.535 Zeichen. In unserem Fall haben wir die Länge der Spalten bis 100 und 40 begrenzt.

6.2 Authentifizierung

Datenbanknutzer: virtueller Nutzer eines Datenbanksystems → MySQL hat eine eigene Nutzerverwaltung (Datenbank-Login + Passwort)

Um einen Mysql-Benutzer hinzuzufügen, braucht man einen neuen Benutzer in der User-Tabelle der Mysql Datenbank hinzuzufügen, mit host, user-Namen und Passwort einzugeben.

```
mysql> use mysql; → Zugriff auf eine bestimmte Datenbank
Database changed

mysql> INSERT INTO user
      (host, user, password,
```

```
select_priv, insert_priv, update_priv)
VALUES ('localhost', 'guest',
PASSWORD('guest123'), 'Y', 'Y', 'Y');
```

Query OK, 1 row affected (0.20 sec)

```
mysql> FLUSH PRIVILEGES;
```

Query OK, 1 row affected (0.01 sec)

Die Anweisung FLUSH PRIVILEGES muss ausgeführt werden . Nach der Ausführung dieses Befehls entsteht die Autorisierungstabelle, damit neuer Benutzer mit mysql-Server verbinden kann.

Zugriffsrechte: in MySQL kann spezifiziert werden, welcher (Datenbank-)Nutzer von welchem/er Rechner/Rechnergruppe aus mit welcher Datenbank welche Operationen ausführen darf.

Kontrolle - Ausgabe der vorhandenen Daten

```
mysql> SELECT host, user,password from user ;
+-----+-----+-----+
| host      | user      | password                                     |
+-----+-----+-----+
| localhost | root      | *81F5E21E35407D884A6CD4A731AEBFB6AF209E1B |
| volkszaehler | root      | *81F5E21E35407D884A6CD4A731AEBFB6AF209E1B |
| 127.0.0.1 | root      | *81F5E21E35407D884A6CD4A731AEBFB6AF209E1B |
| ::1       | root      | *81F5E21E35407D884A6CD4A731AEBFB6AF209E1B |
| localhost | debian-sys-maint | *2D35B185F84CEA0C128241A2F709DAB1906122DB |
| localhost | vz        | *C142FB215B6E05B7C134B1A653AD48455157FD79 |
| localhost | phpmyadmin | *81F5E21E35407D884A6CD4A731AEBFB6AF209E1B |
| volkszaehler | root2     | *81F5E21E35407D884A6CD4A731AEBFB6AF209E1B |
| localhost | root2     | *81F5E21E35407D884A6CD4A731AEBFB6AF209E1B |
| localhost | guest     | *F1573429579994EEA4459170FDAC55DF96C4BBE6 |
+-----+-----+-----+
10 rows in set (0.00 sec)

mysql> SELECT host, user,password from user where user='guest' ;
+-----+-----+-----+
| host      | user      | password                                     |
+-----+-----+-----+
| localhost | guest     | *F1573429579994EEA4459170FDAC55DF96C4BBE6 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Abb.6. 5 Anzeige der Benutzer

Quelle: eigene Abbildung

Die Prüfmethode der Passwort ist md5. Mit Benutzernamen guest und passwort guest123 sich anzumelden.

6.3 Administration in der Datenbank

phpMyAdmin ist ein kostenloses Software-Tool, das in PHP geschrieben wurde und die Verwaltung von MySQL über das Web abwickeln soll. phpMyAdmin unterstützt eine Vielzahl von Operationen auf MySQL und MariaDB. Häufig verwendete Operationen (Verwaltung von Datenbanken, Tabellen, Spalten, Beziehungen, Indizes, Benutzer, Berechtigungen usw.) können über die Benutzeroberfläche ausgeführt werden, während Sie immer noch die Möglichkeit haben, eine SQL-Anweisung direkt auszuführen.



The image shows the phpMyAdmin login interface. At the top, there is a logo for phpMyAdmin and the text 'Willkommen bei phpMyAdmin'. Below this, there is a language selection dropdown menu labeled 'Sprache - Language' with 'Deutsch - German' selected. Underneath is a login form with two input fields: 'Benutzername:' containing 'root' and 'Passwort:' containing 'root'. An 'OK' button is located at the bottom right of the login form.

Abb.6. 6 Anmeldefenster bei phpmyadmin

Quelle: 192.168.10.113/phpmyadmin/

Beim ersten Aufruf von phpMyAdmin sollten Sie einen Login-Bildschirm sehen, hier können Sie sich mit dem Benutzernamen root und einem Passwort root anmelden.

6.3.1 Konfiguration der phpmyadmin

- Datenbank-User anlegen:

phpmyadmin:

Home -> "Rechte" -> "Neuen Benutzer hinzufügen"

Benutzername: vz

Host: localhost

Passwort: <root>

keine globale Rechte vergeben!

-> ok

- Datenbank anlegen:

phpmyadmin

Home -> "Datenbank anlegen", Name: "volkszaehler" -> "Anlegen"

oder:

```
CREATE DATABASE `volkszaehler`;
```

- Rechte setzen:

Datenbank "mysql", Tabelle "tables_priv":

"Einfügen":

Host: "localhost"

Db: "volkszaehler"

User: "vz"

Table_name: "data" (anschließend das Gleiche mit "entities", "entities_in_aggregator", "properties", "tokens")

Table_priv: "select", "update" und "insert"

Die kann auch mit SQL-Anweisung im Mysql ausgeführt werden.

6.3.2 Funktion der phpmyadmin

- Intuitive Weboberfläche
- Unterstützung für die meisten MySQL-Funktionen:
 - durchsuchen und löschen von Datenbanken, Tabellen, Ansichten, Feldern und Indizes
 - Erstellen, Kopieren, Löschen, Umbenennen und Ändern von Datenbanken, Tabellen, Feldern und Indizes
 - Wartungsserver, Datenbanken und Tabellen mit Vorschlägen zur Serverkonfiguration
 - Ausführen, Bearbeiten und Merken von SQL- Anweisungen, sogar Batch-Abfragen
 - MySQL-Benutzerkonten und Privilegien verwalten
 - Verwalten von gespeicherten Prozeduren und Triggern
- Datenbank volkszaehler

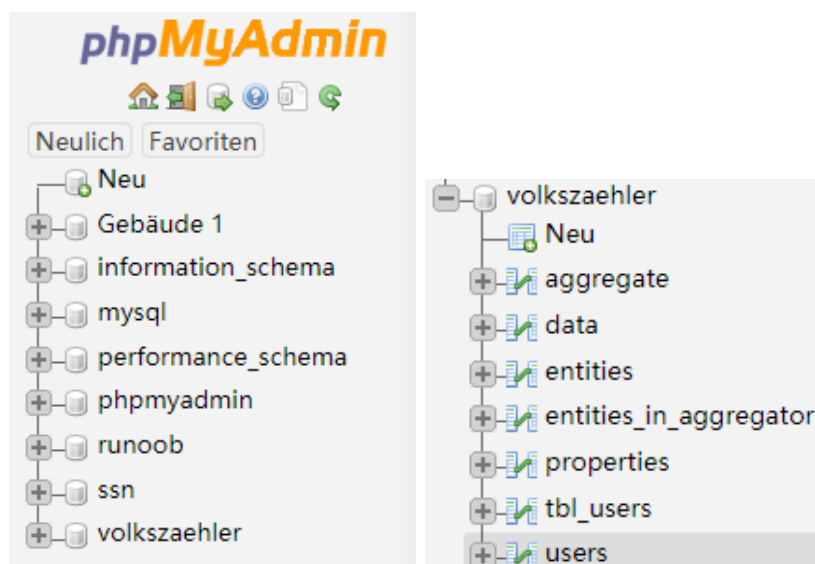


Abb.6. 7 Datenbank in der phpmyadmin

Quelle: <http://192.168.10.113/phpmyadmin/index.php>

Die Datenbanken in der phpmyadmin sind auch die im mysql angezeigte Databases. Tabelle users der Datenbank mysql zeigt alle Benutzer, die mit mysql-server verbunden sind. Hier kann man auch MySQL-Benutzerkonten und Privilegien verwalten. Die Messwerte sind in der Datenbank volkszaehler unter data gespeichert, und users der volkszaehler ist für die Authentifizierung der mehreren Knoten.

- `SELECT email, password, hash FROM `users``

email	password	hash
913805167@qq.com	\$2y\$10\$d9pc0W91m0.BmVLkvtZVoeCi4wnq8E4wxnV5JlLNCLd...	fae0b27c451c728867a567e8c1bb4e53
591456108@qq.com	\$2y\$10\$GfQ.ZyCO79DFc1XuU5vtqecj77vesCKCtbfjsjQKVu.y...	8fe0093bb30d6f8c31474bd0764e6ac0

Abb.6. 8 Daten zur Authentifizierung

Quelle: 192.168.10.113/phpmyadmin

```
mysql> select email,password,hash from users;
+-----+-----+-----+
| email          | password          | hash          |
+-----+-----+-----+
| 913805167@qq.com | $2y$10$d9pc0W91m0.BmVLkvtZVoeCi4wnq8E4wxnV5JlLNCLdNWfsJw0LT2 | fae0b27c451c728867a567e8c1bb4e53 |
| 591456108@qq.com | $2y$10$GfQ.ZyCO79DFc1XuU5vtqecj77vesCKCtbfjsjQKVu.ylFt3SUp.vK | 8fe0093bb30d6f8c31474bd0764e6ac0 |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

- Datenbank mysql

SQL ist eine sogenannte Datenbanksprache. Mit dieser senden wir Befehle an die MySQL-Datenbank und erhalten von dieser eine entsprechende Antwort zurück, beispielsweise die angefragten Daten einer Tabelle.

Diese SQL-Befehle könnt man zum einen mittels PHP ausführen, oder auch direkt in phpMyAdmin. Drückt man auf den Tab SQL, so könnt man direkt SQL-Befehle eingeben und diese ausführen lassen. Ein SQL-Befehl zum Anzeigen der user-Tabelle ist:

- SELECT * FROM user

The screenshot shows the phpMyAdmin interface with the following elements:

- SQL query input: `SELECT * FROM `user``
- Buttons: Bearbeiten, Kopieren, Löschen
- Filters: Anzahl der Datensätze: 25, Zeilen filtern: Diese Tabelle durchsuchen
- Sortierung: Nach Schlüssel sortieren: keine
- Table with columns: Host, User, Password, Select_priv

Host	User	Password	Select_priv
localhost	root	*81F5E21E35407D884A6CD4A731AEBFB6AF209E1B	Y
volkszaehler	root	*81F5E21E35407D884A6CD4A731AEBFB6AF209E1B	Y
127.0.0.1	root	*81F5E21E35407D884A6CD4A731AEBFB6AF209E1B	Y
:::1	root	*81F5E21E35407D884A6CD4A731AEBFB6AF209E1B	Y
localhost	debian-sys-maint	*2D35B185F84CEA0C128241A2F709DAB1906122DB	Y
localhost	vz	*C142FB215B6E05B7C134B1A653AD4B455157FD79	N
localhost	phpmyadmin	*81F5E21E35407D884A6CD4A731AEBFB6AF209E1B	N
volkszaehler	root2	*81F5E21E35407D884A6CD4A731AEBFB6AF209E1B	Y
localhost	root2	*81F5E21E35407D884A6CD4A731AEBFB6AF209E1B	Y
localhost	quest	*F1573429579994EEA4459170FDAC55DF96C4BBE6	Y

Abb.6. 9 Benutzerübersicht bei der SQL-Abfrage

Quelle: 192.168.10.113/phpmyadmin

- Die wichtigsten SQL-Operationen
 - Einfügen von Datensätzen insert ...
 - Anzeigen von Datensätzen select ...
 - Ändern von Datensätzen update ...
 - Löschen von Datensätzen delete ...

7 Messdaten von mehreren Knoten auf dem Volkszähler visualisieren und aufteilen

Bei der grafischen Darstellung und Synchronisation der aktuellen Messdaten an einen Intelligenzzähler (Volkszähler) sind nicht nur ein Messgerät, sondern mehrere Messgeräte zu realisieren.

7.1 Netzschaltung der mehreren Knoten aufzubauen



Abb.7. 1 Anschluss zwischen Messgerät und Stromnetz

Quelle:eigene Abbildung

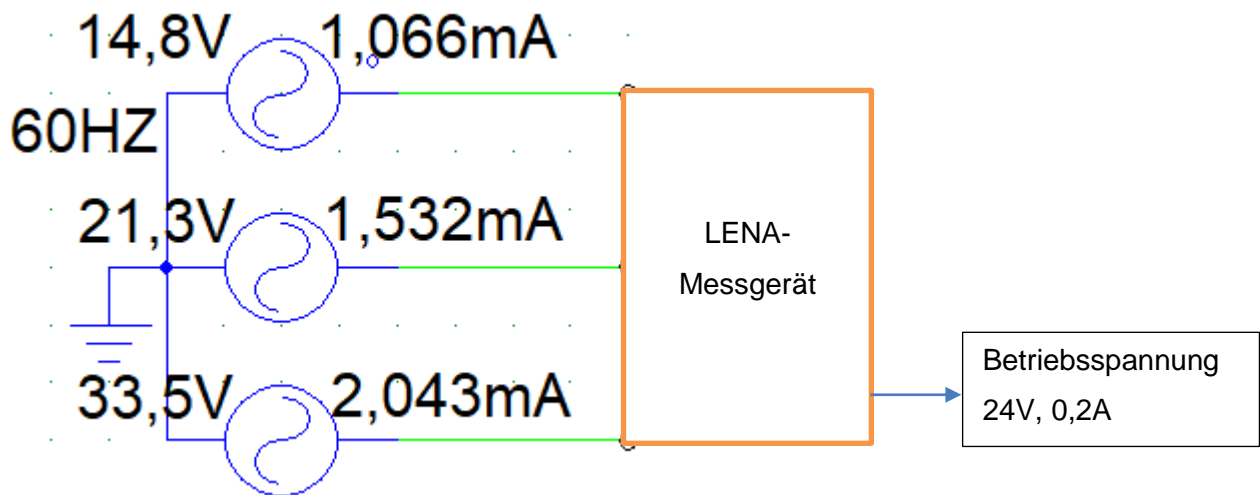


Abb.7. 2 Netzschaltung eines Knotens

Quelle: eigene Abbildung

Jeweiliger Messgerät wird an das Stromnetz angeschlossen, das die Betriebsspannung 24V liefert. Auf der anderen Seite des LENA-Messgerätes ist der auch mit 3-Phasen Strom- und Spannung angeschlossen, die 6 Werte als Messdaten bezeichnet werden.

7.2 Aufteilung der Messwerte von mehreren Knoten

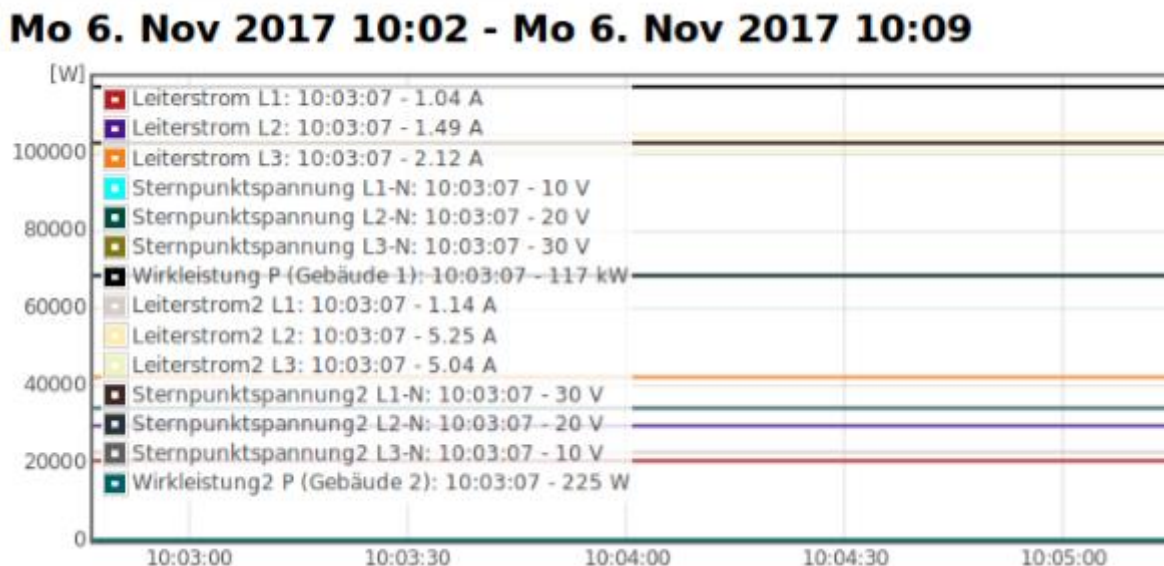


Abb.7. 3 synchronisierte Messwerte auf dem Volkszaehler

Quelle: 192.168.10.113

Nach der Synchronisation der Messwerte auf dem Volkszaehler, treten die aktuelle Werte kontinuierlich als Kurve auf, die zeitlich verändert werden können.

Jeder Messwert hat seinen eigenen Kanal, der beschreibt nicht nur aktuellen Wert, sondern auch Minimum, Maximum und Durchschnittswert. Bei Wirkleistung P steht noch Wh als Verbrauch, damit der Benutzer den Energieverbrauch eines Haushalts deutlich zu ersehen. Dann werden die Messdaten sich in Gruppen ordentlich aufgeteilt.

- Kanal hinzufügen
- Kanal erstellen
- Typ wählen
- Titel eingeben

Nachdem der neue Kanal Gebäude1 erstellt wird, besitzt der seinen eigenen UUID. Der UUID, den jeder Messwert hat, besteht aus einer 16-Byte-Zahl, die hexadezimal notiert und in fünf Gruppen unterteilt wird. Der dient der Identifikation der Daten.

The image shows a software dialog box titled "Kanal hinzufügen" (Add Channel). It has three tabs: "Öffentliche Kanäle", "Private Kanäle", and "Kanal erstellen" (highlighted in blue). Below the tabs is a table with two columns: "Eigenschaft" (Property) and "Wert" (Value).

Eigenschaft	Wert
Middleware:	Local (default) [v]
Typ:	Gebäude [v]
Titel	Gebäude1
Öffentlich	<input type="checkbox"/>
Farbe	[red color swatch]
Stil	lines [v]
Füllgrad	[empty input field]
Achse	auto [v]

Below the table is an "Erstellen" (Create) button and a "Cookie:" label with a small icon. At the bottom of the dialog is a separate box titled "Kanal UUID" containing the value "104dcbe0-c2dd-11e7-aa21-3d60a55e889f". An "Ok" button is located at the bottom right of the entire dialog area.

Abb.7. 4 Knoten erstellen

Quelle: 192.168.10.113

Dann tritt ein Kanal Gebäude1 darunter auf, alle Messwerte, die zu dem Messgerät1 gehört, sind auf der Gruppe Gebäude1 verschoben. Ebenfalls wird bei mehreren Knoten solche Schritte wiederholen.

Min	Max	Ø	Aktuell	Verbrauch
0.768 A	1.18 A	1.04 A	1.09 A	
1.48 A	1.51 A	1.49 A	1.49 A	
2.08 A	2.12 A	2.12 A	2.12 A	
10 V	10 V	10 V	10 V	
20 V	20 V	20 V	20 V	
30 V	30 V	30 V	30 V	
117 kW	117 kW	117 kW	117 kW	59 MWh
1.11 A	1.14 A	1.14 A	1.14 A	
5.25 A	5.25 A	5.25 A	5.25 A	
5.03 A	5.04 A	5.04 A	5.03 A	
30 V	30 V	30 V	30 V	
20 V	20 V	20 V	20 V	
10 V	10 V	10 V	10 V	
212 W	225 W	225 W	224 W	113 kWh

Abb.7. 5 Messwerte der Kanäle von mehreren Knoten

Quelle: 192.168.10.113

8 Zusammenfassung und Ausblick

Die nennen die Übung Praktikum, und da lernen sie, wie man das Linux Betriebssystem bedient und Experimente macht. Und auch die Python-Programmierung, die über das Linux Betriebssystem läuft, wird kennengelernt. Auch Beaglebone Black, der als Middleware bezeichnet, mit dem Web-Frontend mitgeliefert. Der verbindet die Messdaten und das Frontend. Seine Struktur und Funktion wurden umfangreich studiert.

In der Mitte wurden die Module und die Kommunikation des Messgerätes sowie die Grundlagen von Software und Datenanalyse einfach erläutert. Das war sehr wichtig und auch schwierig in dem ganzen Projekt. Die Nutzung von Linux und die Analyse der Software waren der Hauptteil in diesem Berufspraktikum. In der anschließenden Bachelorarbeit werden die Erkenntnisse aus der Projektarbeit einfließen und weitere Aufgaben des Projektes gelöst und das Thema abgeschlossen.

Die nachfolgende Hauptarbeit ist grafische Darstellung und Synchronisation der aktuellen Messdaten an einen Smartmetering (Volkszähler). Hier sollen diese Daten dann in einer Datenbank Mysql abgelegt werden. Nachdem eine Schnittstelle zwischen Mysql und Volkszaehler aufgebaut wird, sind die Messdaten auf dem Volkszaehler kontinuierlich visualisiert. Schließlich werden die von mehreren Knoten aufgeteilt, weil nicht nur ein Messgerät, sondern mehrere Messgeräte Authentifizierung zu realisieren sind.

9 Abkürzungsverzeichnis

Abkürzung	Bezeichnung
BDEW	Bundesverband der Energie- und Wasserwirtschaft
LCD	Liquid Crystal Display
VDE	Verband der Elektrotechnik
LENA	Leistungsmesser/Energiezähler NA-Schutz
USB	Universal Serial Bus
RTU	Remote Terminal Unit
TCP	Transmission Control Protocol
LED	Light Emitting Diode
IPC	Inter-Process Communication
IP	Internet Protocol
VZ	Volkszaehler

10 **Abbildungsverzeichnis**

Abb.1. 1: äußere Form des LENA-Messgerätes (NAS-8)	1
Abb.2. 1: Anschlussplan	2
Abb.2. 2: Messdaten.....	3
Abb.2. 3 Displayanzeige.....	4
Abb.2. 4: LENA Gerät und Beaglebone-PC	5
Abb.3. 1 :Kernkomponenten.....	7
Abb.3. 2 : Anschlüsse auf dem Beaglebone Black.....	7
Abb.3. 3 Ethernet Kabel Verbindung	8
Abb.3. 4 zwischen LENA und BBB	8
Abb.4. 1 Linux	9
Abb.4. 2 Installationsart	10
Abb.4. 3 Partitionierungs-Tool mit angelegten Partitionen	11
Abb.4. 4 Partition im Format ext4 und swap	12
Abb.4. 5 Nutzerdaten, u.a. Name und Login-Passwort	12
Abb.4. 6 Fenster-Menüzeile in Ubuntu	13
Abb.4. 7 Interner BBB mit Linux Terminal-Befehl.....	14
Abb.4. 8 Blockdiagramm der Kommunikation	15
Abb.4. 9 Ausgabe der Werte der „lena_output“	22
Abb.5. 1 Aufbau des Volkszaeler.....	23
Abb.5. 2 Volkszaehler in Server	24
Abb.5. 3 installierter Volkszaehler	26
Abb.6. 1 Datenbank MySQL.....	27
Abb.6. 2 Verbindung mit Mysql-Server	28
Abb.6. 3 Liste aller vorhandenen Datenbanken	29
Abb.6. 4 Anzeige der Tabelle	30
Abb.6. 5 Anzeige der Benutzer.....	31
Abb.6. 6 Anmeldefenster bei phpmyadmin	32
Abb.6. 7 Datenbank in der phpmyadmin.....	34
Abb.6. 8 Daten zur Authentifizierung	35
Abb.6. 9 Benutzerübersicht bei der SQL-Abfrage.....	35

Abb.7. 1 Anschluss zwischen Messgerät und Stromnetz.....	36
Abb.7. 2 Netzschaltung eines Knotens	37
Abb.7. 3 synchronisierte Messwerte auf dem Volkszaehler	37
Abb.7. 4 Knoten erstellen	38
Abb.7. 5 Messwerte der Kanäle von mehreren Knoten.....	39

11 Tabellenverzeichnis

Tabelle 2. 1 Technische Daten	3
Tabelle 3. 1 Entwicklung.....	6
Tabelle 4. 1 Software der LENA	14
Tabelle 4. 2: Funktionsüberblick aus dem Programm „lenainstrument.py“	16
Tabelle 4. 3 FKT_CYCLIC_VALUE 65(0x41)	17
Tabelle 4. 4 FKT_CYCLIC_VALUE 66(0x42)	18
Tabelle 4. 5 FKT_CYCLIC_VALUE 71(0x47)	19

12 Quellen- und Literaturverzeichnis

[1]

NAS-8: Spezifikation des NAS-8

[2]

Beaglebone Black: <https://de.wikipedia.org/wiki/BeagleBoard>

[3]

Beaglebone Black: cdn.sparkfun.com/datasheets/Dev/Beagle/BBB_SRM_C.pdf S24

[4]

Beaglebone Black: <https://beagleboard.org/black>

[5]

Linux: https://wiki.ubuntuusers.de/Ubuntu_Installation/

[6]

Modbus: <https://de.wikipedia.org/wiki/Modbus>

[7]

Funktionsüberblick: LENA_01 BB Kommunikation_004 .doc

[8]

volkszaehler :<https://wiki.volkszaehler.org/start>

[9]

mysql :<https://www.tutorialspoint.com/mysql/>

[10]

phpmyadmin :192.168.10.113/phpmyadmin/

[11]

phpmyadmin :<http://192.168.10.113/phpmyadmin/index.php>