



Höhere Technische Bundeslehranstalt Wien 3, Rennweg
IT & Mechatronik
HTL Rennweg: Rennweg 89b
A-1030 Wien, Tel +43 1 24215-10, Fax DW 18

Diplomarbeit

SGREENING School Greening

ausgeführt an der
Höheren Abteilung für
Informationstechnologie/Netzwerktechnik
der Höheren Technischen Lehranstalt Wien 3 Rennweg

im Schuljahr 2017/2018

durch

**Ruth Slavicky
Fabian Steiner
Dominik Turner**

unter der Anleitung von

Clemens Kussbach
Christian Schöndorfer
Herbert Buschbeck

Wien, Mai 2018

Kurzfassung

Diese Diplomarbeit befasste sich mit der Inventur der IT-Infrastruktur der Schule, dem Erstellen eines Berichtes über den ökologischen Fußabdruck der Schule sowie der Implementierung einer Serverraumüberwachung. Die IT-Geräte wurden in einer übersichtlichen Tabelle zusammengefasst. Diese wurde als Ausgangspunkt für die Planung der Strommessungen verwendet. Um die Strommessungen sinnvoll und effizient durchzuführen, wurden die Geräte in Gruppen, basierend auf deren Spezifikationen, eingeteilt und vermessen. Um auch Raummessungen zu bekommen, wurde, als Vergleich, auch einen Raum hinsichtlich des Stromverbrauchs komplett gemessen. Mit dem Bericht und den Messungen als Grundlage wurden des Weiteren auch Lösungsvorschläge sowie „Quick Wins“ ausgearbeitet. Die Serverraumüberwachung des Raumes 078 wurde durch eine selbstgebaute Lösung ersetzt und mit einem SMS-System versehen. Des Weiteren wurde auch in dem Raum 926 eben jenes Überwachungs- und Benachrichtigungssystem installiert. Außerdem ist eine Simulation eines Notfalls durchgeführt und ausgewertet worden. Eine Webseite, welche die aktuellen Messwerte der Sensoren der Serverräume darstellt, wurde programmiert und steht zur Verfügung.

Abstract

This diploma thesis deals with the inventory of the IT infrastructure of the school, the preparation of a report on the ecological footprint of the school as well as the implementation of a monitoring system for the server rooms. The IT equipment was summarized in a clearly arranged table, which was used as a basis for the planning of power measurements. In order to be able to carry out these measurements in a reasonable and efficient way, the equipment was classified and surveyed in groups based on their specifications. In order to be able to achieve room measurements a room was also completely measured with regard to power consumption for comparison. On the basis of the report and the measurements new approaches as well as “quick wins” were additionally elaborated. The server room monitoring system of room 078 was replaced with a self-made solution and provided with an SMS system. Furthermore, the same monitoring and notification system was installed in room 926. Moreover, an emergency case was simulated and assessed. A web site presenting the current measuring values of the sensors in the server rooms was programmed and is available.

Ehrenwörtliche Erklärung

Ich versichere,

- dass ich meinen Anteil an dieser Diplomarbeit selbstständig verfasst habe,
- dass ich keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe
- und mich auch sonst keiner unerlaubten Hilfe bzw. Hilfsmittel bedient habe.

Wien, am

Pl. Dominik Turner

Pl. Stv. Ruth Slavicky

Ma. Fabian Steiner

Präambel

Die Inhalte dieser Diplomarbeit entsprechen den Qualitätsnormen für „Ingenieurprojekte“ gemäß § 29 der Verordnung des Bundesministers für Unterricht und kulturelle Angelegenheiten über die Reife- und Diplomprüfung in den berufsbildenden höheren Schulen, BGBl. Nr. 847/1992, in der Fassung der Verordnungen BGBl. Nr. 269/1993, Nr. 467/1996 und BGBl. II Nr. 123/97.

Betreuende Lehrer:

Wien, am

*DI Clemens Kussbach
Hauptbetreuer*

*DI Herbert Buschbeck
Stv. Hauptbetreuer*

*DI Christian Schöndorfer
Betreuer*

Schulleitung:

Wien, am

*DI Gerhard Jüngling
Direktor*

*Dr. Mag. Gerhard Hager
Abteilungsleiter IT*

Liste der Kooperationspartner:

- BellEquip GmbH
- Stulz Austria GmbH

Inhaltsverzeichnis

1	ZIELE DES PROJEKTS	15
1.1	Hauptziele	15
1.1.1	Inventur der IT-Infrastruktur	15
1.1.2	Gerätegruppen	15
1.1.3	Stromverbrauch der einzelnen Geräte	15
1.1.4	Stromverbrauch von einzelnen Räumen	15
1.1.5	Ökologischer Fußabdruck der IT-Infrastruktur	16
1.1.6	Konzepte zur Verbesserung des ökologischen IT-Fußabdrucks	16
1.1.7	Umsetzen der „Quick Wins“	16
1.1.8	Feststellung von Temperaturverlauf und -verteilung im Serverraum 078	16
1.1.9	Aufbau eines repräsentativen Messsystems	16
1.1.10	Implementierung des Warn- bzw. Hellosystems	17
1.1.11	Implementierung einer Notabschaltung	17
1.2	Optionale Ziele	17
1.2.1	Lifecycle – Anschaffung	17
1.2.2	Lifecycle – Entsorgung	17
1.2.3	Webschnittstelle zur Steuerung und Überwachung	17
1.2.4	Zwei unabhängige Überwachungssysteme	18
1.2.5	Thin - Clients	18
1.2.6	Redundantes Sensorik-System	18
1.2.7	Kalt- / Warmgang Konzept entwickeln	18
1.2.8	Außenluftkühlkonzept entwickeln	18
1.3	NICHT Ziele	19
1.3.1	Kühlkonzept für die Steigschächte	19
1.3.2	Störung des Schulbetriebs	19
1.3.3	Stromverbrauch aller elektronischen Geräte	19
1.3.4	Komplette Umsetzung des Verbesserungskonzepts	19
1.3.5	Berücksichtigung der Werkstätten	19
1.3.6	Feuerlöschsystem für den Serverraum 078	19
2	KOOPERATION	21
2.1	BellEquip GmbH	21
2.2	Stulz Austria GmbH	21
3	SERVERRAUMÜBERWACHUNG	23
3.1	Kurzbeschreibung	23
3.2	Kontext	23
3.3	Notwendigkeit einer Sensorüberwachung	23
3.4	Situation vor dem Projekt	23

3.4.1	Temperaturkurve	25
3.5	Neues Überwachungssystem	28
3.5.1	Komponenten	28
3.5.2	Notabschaltung.....	43
3.5.3	Paketverwaltung.....	44
3.5.4	Visualisierung.....	45
3.5.5	Redundanz	47
3.5.6	Kühlsystem	47
3.6	Produkt.....	53
3.6.1	Struktur	53
3.6.2	Das Install Skript.....	55
3.6.3	Die Datenbanken.....	57
3.6.4	Services	59
3.6.5	Webinterface.....	83
3.7	Ausblick und Konfiguration.....	95
3.7.1	Wartung der SIM Karte	95
3.7.2	Konfiguration des „smsd“ Services	95
3.7.3	Erweiterung des Messsystems.....	95
3.7.4	Einrichten von Notabschaltungen.....	96
4	INVENTUR UND STROMMESSUNGEN.....	97
4.1	Kurzbeschreibung.....	97
4.2	Situation vor dem Projekt	97
4.3	Inventarlisten - Mengengerüst	97
4.4	Gruppierung der Geräte	97
4.4.1	PCs	98
4.4.2	Netzwerkkomponenten	99
4.4.3	Sonstiges	99
4.5	Messtechnik.....	100
4.5.1	Theoretischer Hintergrund	100
4.5.2	Messgeräte/Systeme	108
4.6	Messplanung / Messumsetzung	120
4.6.1	Messung von einzelnen Geräten	120
4.6.2	Messung von Räumen	121
4.6.3	Messung von Server-Rack.....	121
4.7	Messergebnisse	122
4.7.1	Einzelmessungen	122
4.7.2	Raummessung.....	130
4.7.3	Server-Rack-Messung	133
4.7.4	Hochrechnung Übungsrack.....	134
4.8	Ist Situation	134

5	BERICHT DES ÖKOLOGISCHEN FUßABDRUCKS.....	135
5.1	Kurzbeschreibung.....	135
5.2	Situation vor dem Projekt.....	135
5.3	Theorie.....	135
5.3.1	Grundlagen	135
5.3.2	Der Bericht über den ökologischen Fußabdruck.....	138
6	EMPFEHLUNGEN UND VERBESSERUNGEN	151
6.1	Situation vor dem Projekt.....	151
6.2	Beschreibung der „Quick Wins“, Verbesserungsmöglichkeiten + Vorschläge	151
6.2.1	Mögliche längerfristige Maßnahmen	151
7	ABBILDUNGSVERZEICHNIS.....	153
8	TABELLENVERZEICHNIS.....	155
9	LITERATURVERZEICHNIS.....	157
10	ANHANG	159
10.1	Ordner Struktur der beiliegenden CD	159
10.2	Dokumentation des Produktes	161
10.2.1	Schaltpläne.....	161
10.2.2	Stücklisten.....	162
10.3	Bericht des ökologischen Fußabdrucks	163

1 Ziele des Projekts

1.1 Hauptziele

1.1.1 Inventur der IT-Infrastruktur

Die Liste aller Geräte der IT-Infrastruktur ist aktualisiert.

Wir ziehen hierfür zunächst die Geräte-Liste der Schule zurate, überprüfen sie auf Aktualität und Vollständigkeit und ergänzen sie bei Bedarf. Dies wird in Zusammenarbeit mit dem Team „Atlas“ erfolgen.

1.1.2 Gerätegruppen

Geräte sind in Gruppen zusammengefasst.

Es werden Gerätegruppen nach Typähnlichkeit gebildet. Zum Beispiel werden Switches einer Generation in einer Gruppe zusammengefasst.

1.1.3 Stromverbrauch der einzelnen Geräte

Der konkrete Stromverbrauch der einzelnen Gerätegruppen steht für verschiedene Betriebszustände fest.

In jeder Gerätegruppe wird der Stromverbrauch einzelner Geräte stichprobenartig in verschiedenen Betriebszuständen gemessen, beispielsweise während der Unterrichtszeit und durch Benchmark-Tests.

1.1.4 Stromverbrauch von einzelnen Räumen

Der konkrete Stromverbrauch von exemplarischen Räumen steht für die Unterrichtszeit fest

Im Serverraum 078 planen wir eine Demoversion der Firma BellEquip für eine gesamte Messung des Raumes. Exemplarische Räume, wie ein EDV Saal, KU-Saal und Labor werden mittels Log-Scripts auf Aktivität untersucht. Mit diesen Werten und den Ergebnissen aus RE-M 3 erhaltenen Einzelmessungen kann mittels Hochrechnung ein Stromverbrauch pro Raum ermittelt werden.

1.1.5 Ökologischer Fußabdruck der IT-Infrastruktur

Der ökologische IT-Fußabdruck der Schule ist bestimmt.

Mithilfe des gemessenen Stromverbrauchs der einzelnen Gerätegruppen und durch Hochrechnungen mithilfe der Gerätegruppen und der Anzahl der Geräte laut Inventarliste wird der ökologische Fußabdruck der IT-Infrastruktur der Schule bestimmt. Des Weiteren wird der größte Teilfußabdruck der Schule mit dem höchsten Verbesserungspotential ermittelt.

1.1.6 Konzepte zur Verbesserung des ökologischen IT-Fußabdrucks

Die Konzepte zur Verbesserung des größten Teilfußabdrucks mit dem höchsten Verbesserungspotential, sowie für die erkannten „Quick Wins“ sind erstellt.

Einerseits wird ein Konzept zur Verbesserung des größten Teilfußabdrucks der IT-Infrastruktur der Schule erstellt, andererseits werden die Ergebnisse nach so genannten „Quick Wins“, das sind leicht umsetzbare Verbesserungen mit immanenter Wirkung, analysiert.

1.1.7 Umsetzen der „Quick Wins“

Die aus der Analyse des ökologischen IT-Fußabdrucks der Schule erkannten „Quick Wins“ sind umgesetzt.

Durch die Analyse der gewonnenen Daten erkennbare Sofortmaßnahmen, die den Energieverbrauch senken, werden umgesetzt.

1.1.8 Feststellung von Temperaturverlauf und -verteilung im Serverraum 078

Es ist festgestellt, wie die Temperaturverteilung im Zeitverlauf nach Ausfall einer / beider Klimaanlage aussieht.

Zur Feststellung des Temperaturverlaufs wird ein Mini Experiment aufgebaut, das als Grundlage für das neue Überwachungs- und Steuerungssystem dienen soll.

1.1.9 Aufbau eines repräsentativen Messsystems

Die Sensoren sind so im Serverraum verteilt, dass im Ernstfall innerhalb von 15 Minuten festgestellt werden kann, dass ein Problem vorliegt.

Die Analyse des Mini Experimentes wird verwendet, um die Sensoren zweckdienlich zu positionieren.

1.1.10 Implementierung des Warn- bzw. Hellosystems

Es ist ein Warn- bzw. Hello System am Raspberry Pi umgesetzt, das regelmäßig eine Benachrichtigung schickt und im Ernstfall eine Warnung ausschickt.

Es wird vermutlich wöchentlich eine SMS gesendet, die bestätigt, dass das Überwachungssystem online ist und bei Notfällen eine SMS, die die Administratoren alarmiert.

1.1.11 Implementierung einer Notabschaltung

Die Programmierung der Sensorik ist nach dem Installieren des Systems entsprechend angepasst und schützt die Komponenten des Serverraums vor beispielsweise Überhitzung.

Mit Hilfe von Schieberegistern wird eine Relaisansteuerung realisiert.

1.2 Optionale Ziele

1.2.1 Lifecycle – Anschaffung

Es wurden Empfehlungen für eine Green-IT bewusste Anschaffung von IT-Geräten erstellt.

Um den ökologischen IT-Fußabdruck der in der HTL Rennweg eingesetzten IT-Komponenten weiter zu senken, wird der Produktionsprozess der Komponenten in die Betrachtungen einbezogen und Richtlinien für die künftige Beschaffung erarbeitet.

1.2.2 Lifecycle – Entsorgung

Es wurden Empfehlungen für eine ressourcenschonende Weitergabe von IT-Geräten nach der Nutzung in der HTL Rennweg erstellt.

Dem Prinzip „Reduce – Reuse – Recycle“ entsprechend, werden Empfehlungen für die Weitergabe der IT-Altgeräte erarbeitet, so dass, soweit sinnvoll möglich, die Nutzungsdauer der Hardware weiter ausgedehnt wird, bzw. für eine ökologische Entsorgung gesorgt ist.

1.2.3 Webschnittstelle zur Steuerung und Überwachung

Es ist eine Webschnittstelle zur Überwachung der Werte, zum Erstellen von Notabschaltplänen und zum Einstellen von zeitgesteuerten Abschaltungen erstellt.

Die Website läuft am Raspberry PI und ist mit Hilfe von Bootstrap gestaltet.

1.2.4 Zwei unabhängige Überwachungssysteme

Zwei unabhängige Systeme laufen und erfüllen beide die idente Funktion.

Das funktionierende System wird dupliziert.

1.2.5 Thin - Clients

Es wurde ein Vergleich des Stromverbrauchs erstellt, wenn man die PCs in den Lehrzimmern durch Thin-Clients und Terminal Server ersetzt.

Der Stromverbrauch bei Einsatz einer Thin-Client-Lösung wurde errechnet und mit dem derzeitigen Stromverbrauch der betroffenen Geräte verglichen.

1.2.6 Redundantes Sensorik-System

Ein redundantes Sensorik-System ist entwickelt und unterstützt Self Recovery.

Zwei bis vier Raspberry Pis arbeiten als Master Slave Cluster, erkennen Ausfälle von eigenen Systemkomponenten und rebooten diese automatisch.

1.2.7 Kalt- / Warmgang Konzept entwickeln

Es ist ein Kalt - / Warmgang Konzept für 078 entwickelt.

Es wird ein Konzept für einen Luftstrom durch Abdichtungen und Repositionierungen der Klimaautomatik geben.

1.2.8 Außenluftkühlkonzept entwickeln

Es gibt ein Konzept mit dem der Serverraum 078 bei Bedarf mittels der Außenluft gekühlt werden soll.

Durch Ventilatoren, die an den oberen Fenstern im Serverraum 078 angebracht sind, wird eine Kühlung durch Außenluft ermöglicht.

1.3 NICHT Ziele

1.3.1 Kühlkonzept für die Steigschächte

Für die Etagen-Verteilerschrank und die Steigschächte wird ein Kühlkonzept umgesetzt.

1.3.2 Störung des Schulbetriebs

Der Schulbetrieb ist durch die Messungen gestört.

1.3.3 Stromverbrauch aller elektronischen Geräte

Der Stromverbrauch von allen elektronischen Geräten der Schule ist ermittelt.

1.3.4 Komplette Umsetzung des Verbesserungskonzepts.

Auch die zeitlich sehr aufwändigen Maßnahmen aus dem Verbesserungskonzept werden umgesetzt.

1.3.5 Berücksichtigung der Werkstätten

Im ökologischen IT-Fußabdruck werden die Maschinen der Mechatronik in den Werkstätten berücksichtigt.

1.3.6 Feuerlöschsystem für den Serverraum 078

Es wird ein Feuerlöschsystem im Serverraum 078 eingebaut.

2 Kooperation

2.1 BellEquip GmbH

Der Diplomarbeit wurde ein Messsystem des Herstellers „raritan“ von der Firma „BellEquip GmbH“ für einen Monat zu Demo-Zwecken zur Verfügung gestellt.

2.2 Stulz Austria GmbH

Drei M26422 Wassersensoren wurden von Werner Mayer der Stulz Austria GmbH zur Verfügung gestellt.

3 Serverraumüberwachung

3.1 Kurzbeschreibung

Bei der Serverraumtemperaturüberwachung geht es darum, die Serverräume 926 und 078 intelligent zu überwachen, den Stromverbrauch dieser zu senken, Kühlkonzepte zu entwickeln und eine Steuerungsautomatik, sowie auch einen Notfallabschaltplan vorzubereiten.

3.2 Kontext

Schon bei der Planung von Rechenzentren / Serverräumen ist es wichtig, auf Energieeffizienz zu achten. Die reicht von zeitgesteuertem Abschalten von nicht benötigten Ressourcen (vgl. 3.5.1.2) über die Bestimmung von Hotspots (siehe 3.4.1), bis zur Planung von effizienten Kühlkonzepten (siehe 3.5.6.3) und deren Abstimmung auf die Außentemperatur (siehe 3.5.6.4).

3.3 Notwendigkeit einer Sensorüberwachung

Da es in den Serverräumen der HTL Rennweg sehr teure und sensible Ressourcen gibt, kein Löschesystem existiert und auch der Ausfall der Klimaanlage nicht festgestellt werden kann, ist es nötig, mittels Umgebungssensoren die Luftfeuchtigkeit, Temperatur und einige weitere Werte zu messen. Außerdem besteht die Notwendigkeit einer Steuerung, damit sowohl Strom gespart, als auch im Ernstfall abgeschaltet werden kann.

3.4 Situation vor dem Projekt

Es gab vor dem Projekt schon den Versuch – in der Diplomarbeit „custodio“ – ein Überwachungssystem einzurichten. Dieses maß allerdings nur an drei nicht repräsentativen Stellen die Temperatur und Luftfeuchtigkeit mittels *DHT22* Sensoren im Serverraum 078. Auch jeweils ein nicht richtig angesteuerter Rauchsensor (er zeigte immer Rauch an) war genau vor den zwei Klimaanlage positioniert. Weiters gab es eine Benachrichtigung per SMS und E-Mail, wenn im Serverraum die vordefinierten Grenzwerte von 30 °C und 75 % Luftfeuchtigkeit – nach der Übernahme geändert auf 35 °C und 80 % Luftfeuchtigkeit – an einem Sensor überschritten wurden. Dann wurde bis zur Behebung des Problems im 5 Minutentakt eine SMS an den Systemadministrator versandt. Beim Ausfallen eines Sensors war dasselbe Vorgehen zu beobachten. Außerdem gab es ein in *WebtoPy* umgesetztes Webinterface, welches in der Lage war, die aktuellen Messwerte in voreingestellten Intervallen anzuzeigen. Diagramme waren auch möglich, jedoch bestanden diese lediglich aus dem Anfangs- und dem Endwert des gewünschten Zeitintervalls.

Im „Serverraum Alt“, wo sich die Produktivserver der Schule befinden, gab es kein Überwachungssystem; allerdings kam es auch erst im Oktober 2017 zur Verlegung der Produktivserver vom „Serverraum 078“ in den „Serverraum Alt“.

3.4.1 Temperaturkurve

Mit Hilfe des alten Sensorsystems „custodio“ (der unten abgebildeten Sensoren) konnte im „Serverraum 078“ folgende Temperatur- und Feuchtigkeitskurve festgestellt werden, an der erkannt werden kann, dass bei ausgefallener Klimaanlage die Temperatur immer langsamer ansteigt:

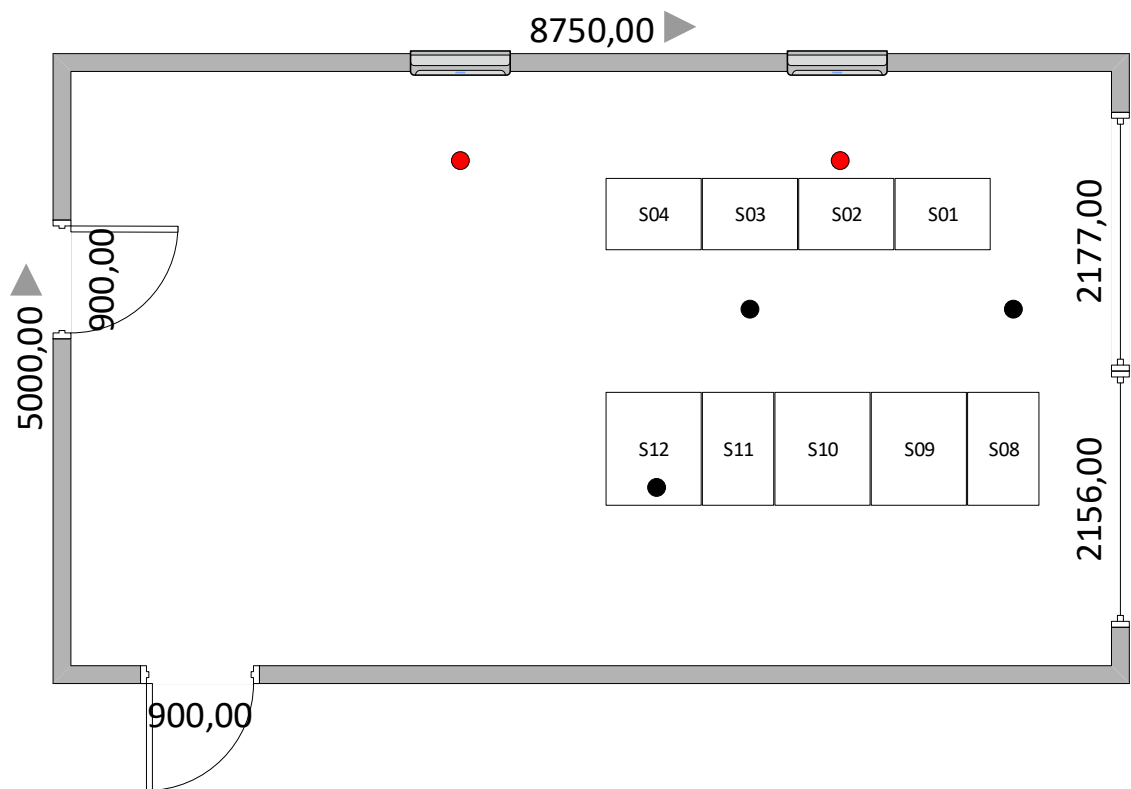


Abbildung 3-1
Schematische Darstellung der Position der DHT 22 und MQ-2 Sensoren

Tabelle 3-1
Legende

Farbe	Bedeutung
Rot	Rauchsensor
Schwarz	Temperatur- und Feuchtigkeitsensor

Tabelle 3-2
Name und Position der Sensoren

Position	Sensorname alt	Sensorname neu	Höhe
S12	Produktivrack	custodio2	1,9 Meter
Raum Mitte	Mitte	custodio1	2,1 Meter
Fenster	Fenster	custodio3	2,1 Meter
Klima Tür	n/a	custodioRauch1	2,1 Meter
Klima Fenster	n/a	custodioRauch2	2,1 Meter

Der Wiedereinschaltungspunkt der Klimaanlage ist jeweils an dem Hochpunkt der Temperaturfunktion, der Startpunkt jeweils um 17:25. Alle Zeitangaben sind in UTC. Wie

man an den Graphen unten sieht, regelt die Klimaanlage danach effizient die Luftfeuchtigkeit und Temperatur auf das gewünschte Niveau.

3.4.1.1 Produktivrack

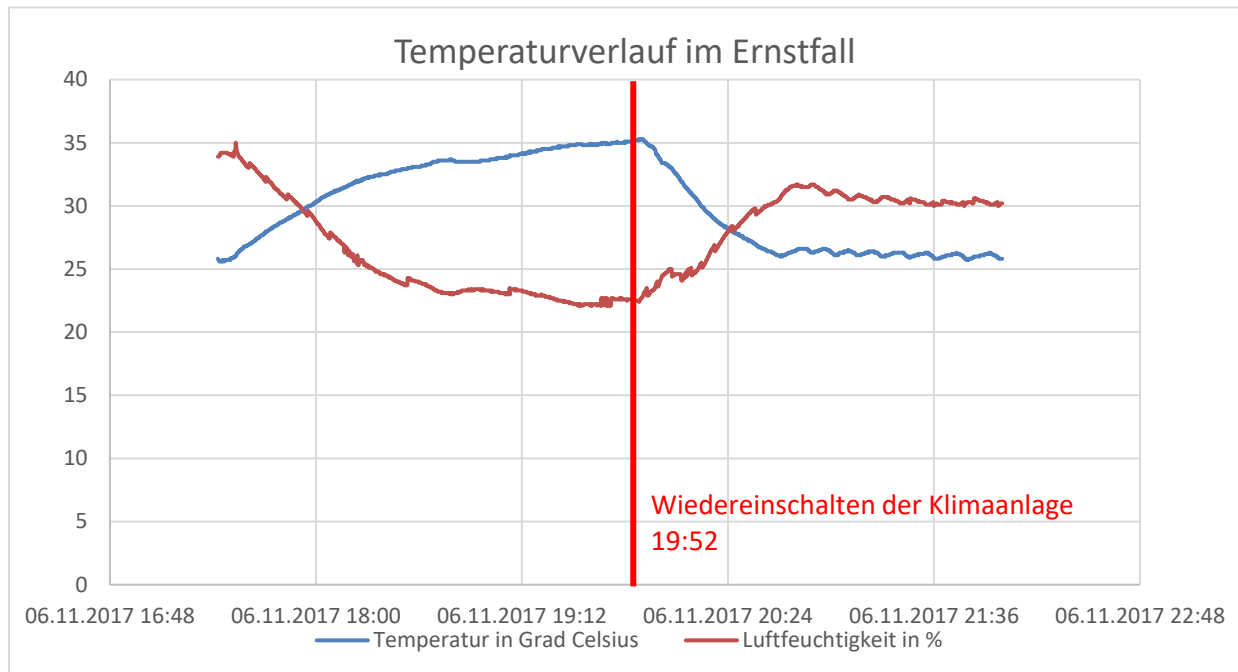


Abbildung 3-2
Temperaturverlauf Produktivrack

3.4.1.2 Raum Mitte

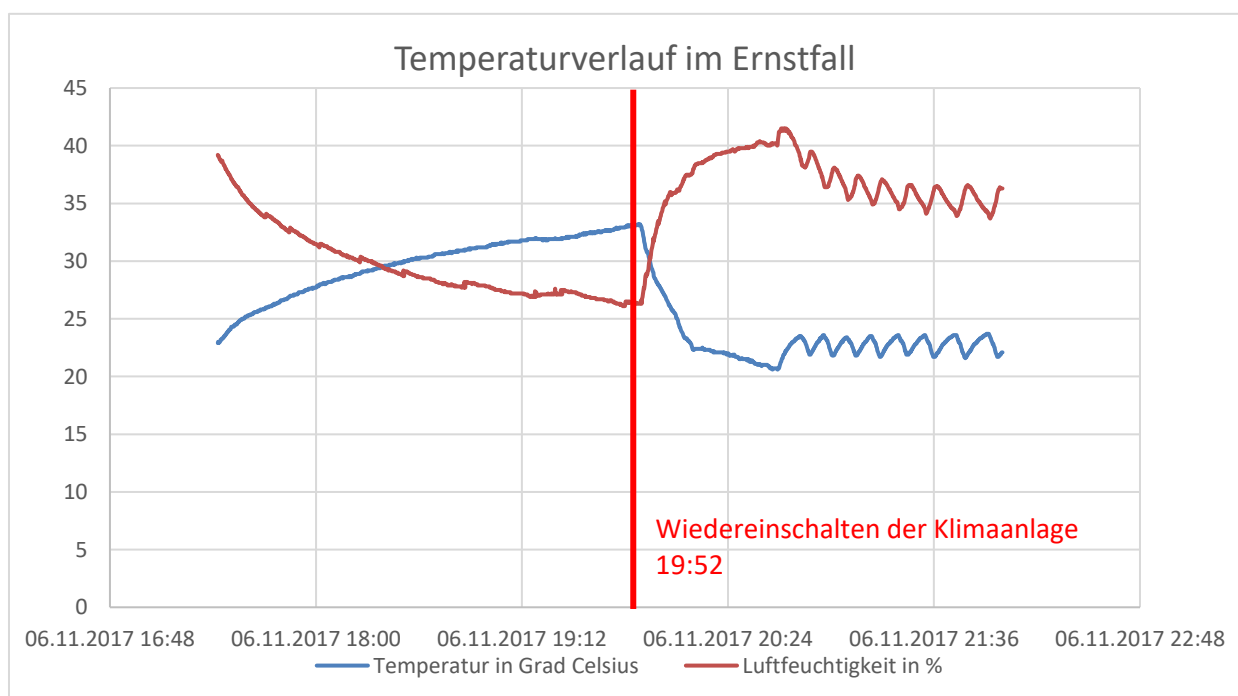


Abbildung 3-3
Temperaturverlauf Raum Mitte

3.4.1.3 Raum Fenster

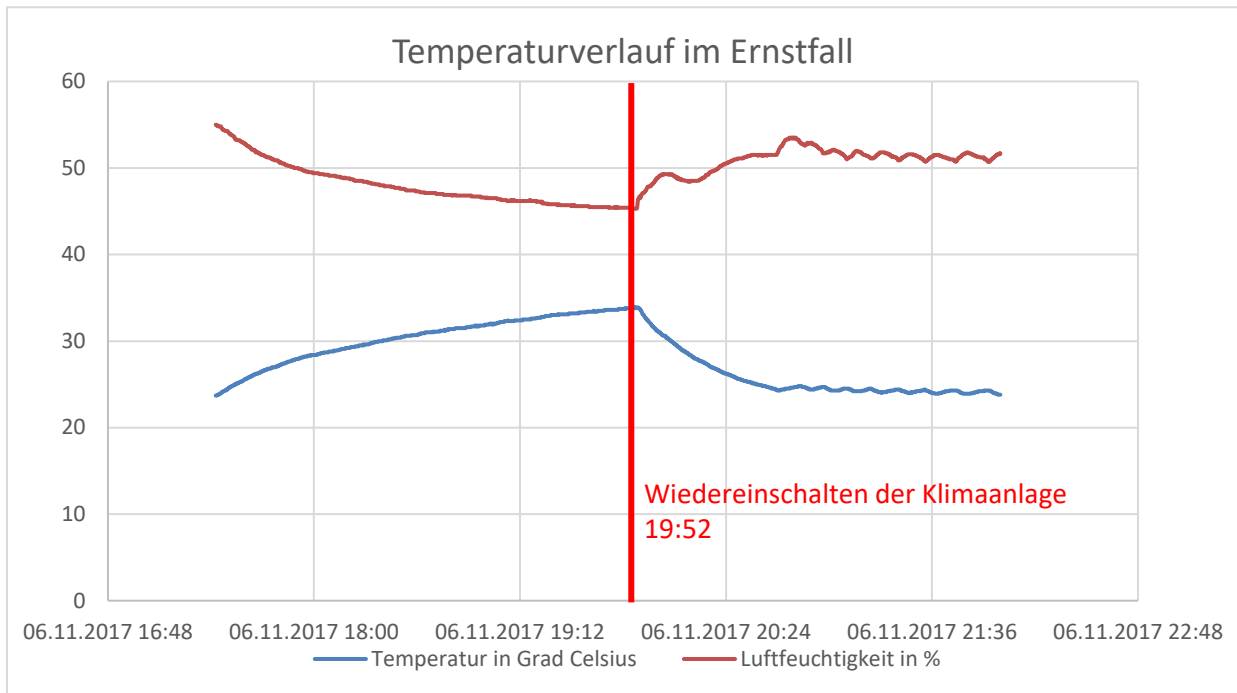


Abbildung 3-4
Temperaturverlauf Raum Fenster

Nach Durchführung des Experiments wurde durch Umbauarbeiten die Platine des Messsystems irreparabel beschädigt und das Sensorsystem „custodio“ musste somit abgebaut werden.

3.5 Neues Überwachungssystem

3.5.1 Komponenten

Das neue System besteht sowohl aus Sensoren, die Temperatur, Luftfeuchtigkeit, Wasserstand (jeweils unter den Klimaanlage) und Rauchentwicklung messen, als auch aus Aktoren zum Abschalten von einzelnen Servern oder sogar ganzen Racks, sowie Modulen zur Kommunikation mit dem Systemadministrator via SMS und Webinterface.

3.5.1.1 Sensoren

3.5.1.1.1 1-Wire Bus

Der 1-Wire Bus ist ein asynchroner, serieller und bidirektionaler Bus, der über eine einzige Leitung im Halbduplexverfahren Nachrichten überträgt. Es gibt einen Bus Master, der die Sensoren und Aktoren ansteuert, die auch als „Slaves“ bezeichnet werden. Hierbei kann immer nur ein Bus Master auf einmal verwendet werden, da es sonst zu Kollisionen kommt. Hingegen können bei einem System mit passivem *Pullup* Widerstand 100 Slaves mit einer maximalen Leitungslänge von 100 Metern und bei Systemen mit aktivem *Pullup* Widerstand bis zu 500 Slaves mit einer maximalen Leitungslänge von 300 Metern angebunden werden. Jeder Device im Bus hat eine eindeutige 64 Bit ROM-ID, die sich aus einem 8 Bit Family-Code, einer 48 Bit langen Seriennummer und einer 8 Bit CRC Prüfsumme zusammensetzt.

Da es bei diesem Bus keine eigene Taktungsleitung gibt, existiert ein vorgegebenes Abfrageschema.

„ (1 Wire Wikipedia, 2018) *Die Synchronisation erfolgt bei jedem Bit mit der vom Master erzeugten fallenden Flanke. Um eine logische 1 zu schreiben, wird der Bus vom Master für 1 bis 15 μ s auf Low-Pegel gezogen, bei einer logischen 0 für 60 bis 120 μ s. Zum Lesen zieht der Master wie beim „Write 1“ Signal den Bus für 1 bis 15 μ s auf Low-Pegel und der Slave hält für die Übertragung einer logischen 0 den Bus darüber hinaus auf Low. Für einen Reset sendet der Master ein Low-Pegel mit einer Dauer von 480 μ s. Ein Slave zeigt seine Anwesenheit an, indem er innerhalb von 60 μ s danach den Bus für mindestens 60 μ s auf Low zieht.“*

Außerdem können die 1-Wire Geräte im Overdrive Modus betrieben werden, wo für eine logische Eins nur für 1-2 μ s ein Low-Pegel anliegen muss und für eine logische Null 6 μ s. Ein Reset benötigt 48 μ s, dauert er länger als 80 μ s, so kehren die Geräte in den regulären Betriebsmodus zurück.

Ein typischer Kommunikationsablauf könnte so aussehen:

1 Wire reset, write and read example with DS2432

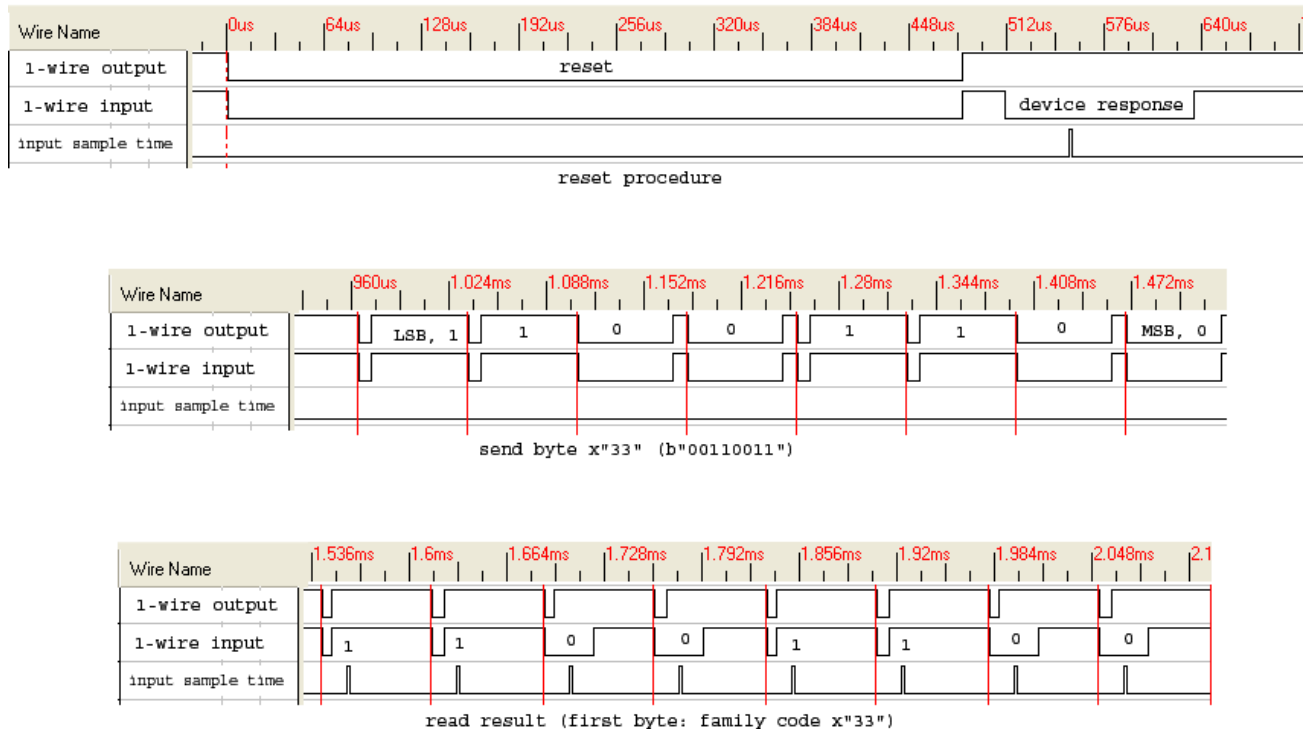


Abbildung 3-5

Beispiel einer Kommunikation zwischen einem FPGA als Master und einem DS2432 Slave

(1 Wire Wikipedia, 2018)

<https://de.wikipedia.org/wiki/Datei:1-Wire-Protocol.png>

Im regulären Modus kann man eine Datenrate von bis zu 16,3 Kbit/s erreichen, im Overdrive Modus sogar bis zu 142 Kbit/s.

Außerdem besitzt der 1-Wire Bus zwei Modi: den „*active mode*“, wo alle Sensoren parallel mit Spannungsversorgung angeschlossen werden und den „*parasite mode*“, in dem alle Sensoren nur eine Datenleitung und eine Masse benötigen. Hierbei ist es erforderlich, dass der 1-Wire Sensor einen zusätzlichen Kondensator besitzt, der über die Datenleitung geladen wird.

Es gibt drei unterschiedliche Modi:

Aktiver Modus ohne zusätzliche Spannungsversorgung:

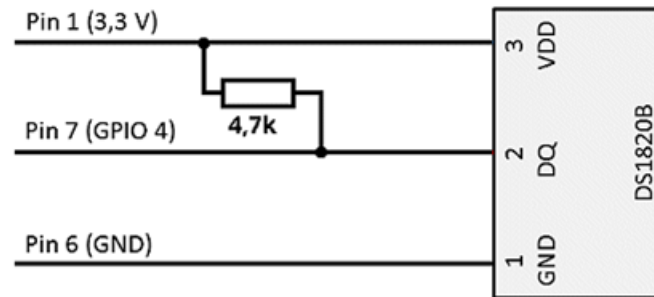


Abbildung 3-6
Aktiver Modus ohne zusätzliche Spannungsversorgung

<http://www.netzmafia.de/skripten/hardware/RasPi/Projekt-Onewire/DS1820-1.gif>

Die Anschlüsse links sollen die GPIO Pins des Raspberrys darstellen, wie auch hier ver-
bildlicht:

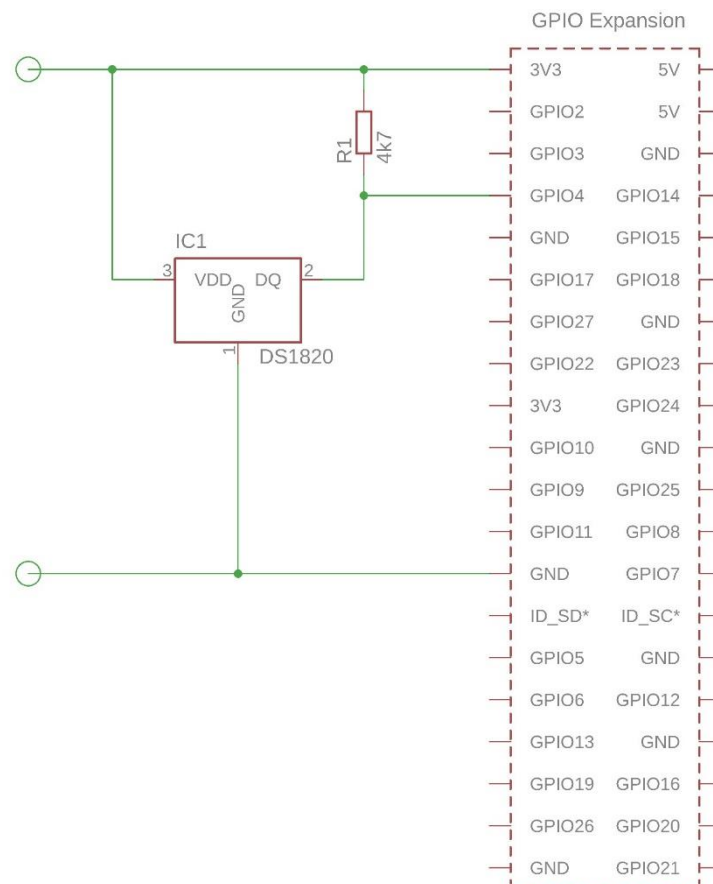


Abbildung 3-7
Schaltplan aktiver Modus ohne zusätzliche Spannungsversorgung

Dieser Modus ist vor allem für geringe Sensoranzahl und kurze Leitungen geeignet.

Aktiver Modus mit zusätzlicher Spannungsversorgung:

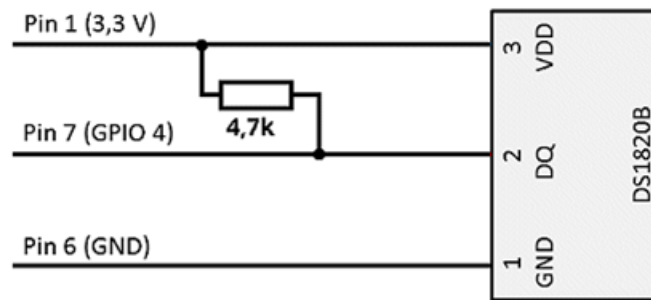


Abbildung 3-8

Blockschaltbild aktiver Modus mit zusätzlicher Spannungsversorgung

<http://www.netzmafia.de/skripten/hardware/RasPi/Projekt-Onewire/DS1820-1.gif>

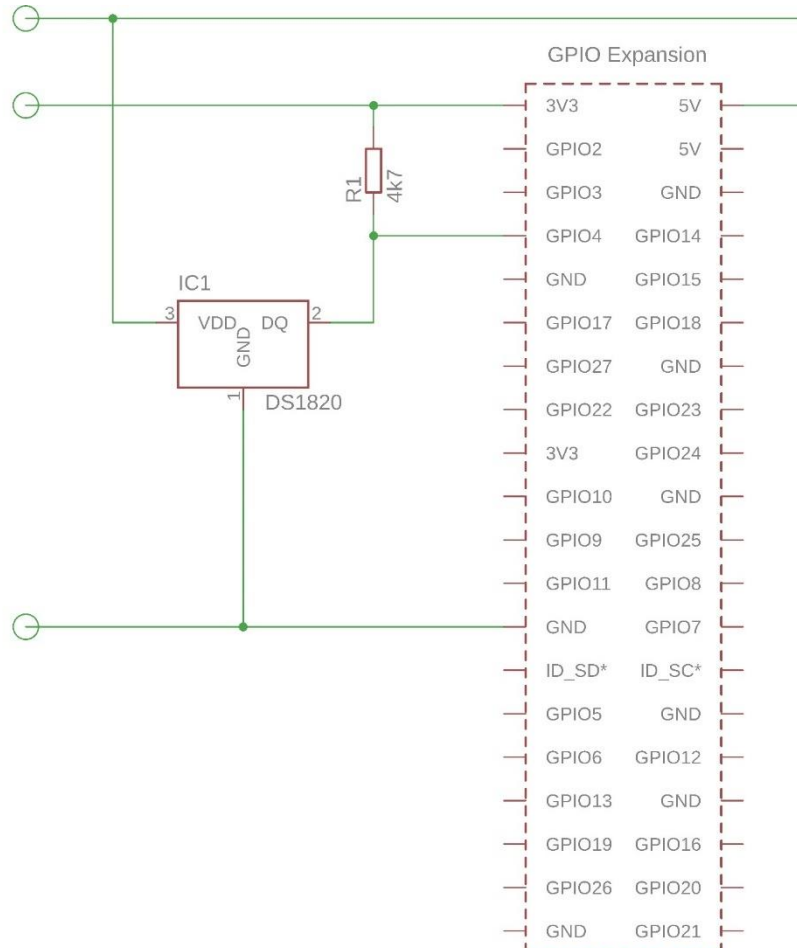


Abbildung 3-9

Schaltplan aktiver Modus mit zusätzlicher Spannungsversorgung

Dieser Modus ist für hohe Sensoranzahlen und große Leitungslängen geeignet (siehe aktiver Pullup Widerstand 3.5.1.1 erster Absatz).

„Parasite“ Modus:

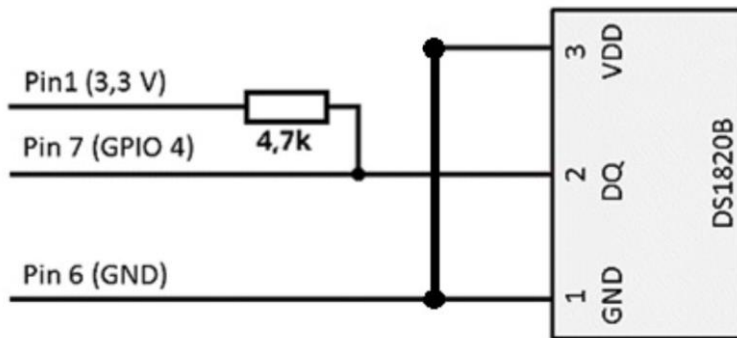


Abbildung 3-10

Blockschaltbild „parasite“ Modus

[http://www.netzmafia.de/skripten/hardware/RasPi/Project-Onewire/DS1820-1.gif](http://www.netzmafia.de/skripten/hardware/RasPi/Projekt-Onewire/DS1820-1.gif)

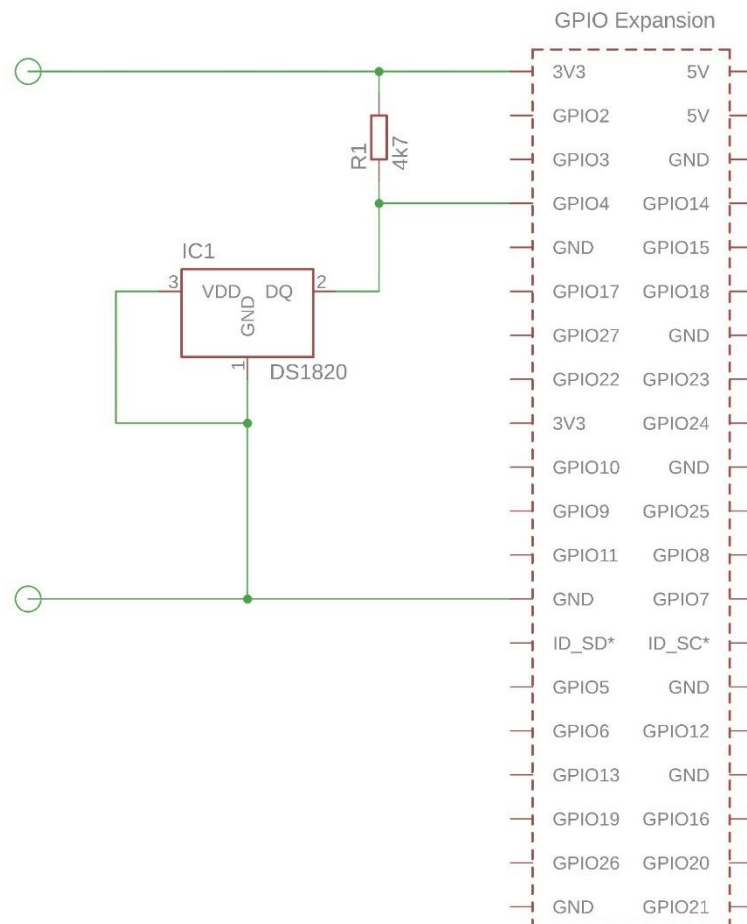


Abbildung 3-11

Schaltplan „parasite“ Modus

Dieser Modus ist geeignet, wenn nur zwei Leitungen (Masse und Daten) zur Verfügung stehen, und ein dedizierter Busmaster verwendet wird, der für den Betrieb von „parasite“

Slaves ausgelegt ist. Er kann allerdings auch mit dem Raspberry Pi verwendet werden, allerdings ist die maximale Slaveanzahl dabei mit vier begrenzt.

Die Eigenschaften des 1-Wire Bus wurden in folgendem Versuchsaufbau getestet:

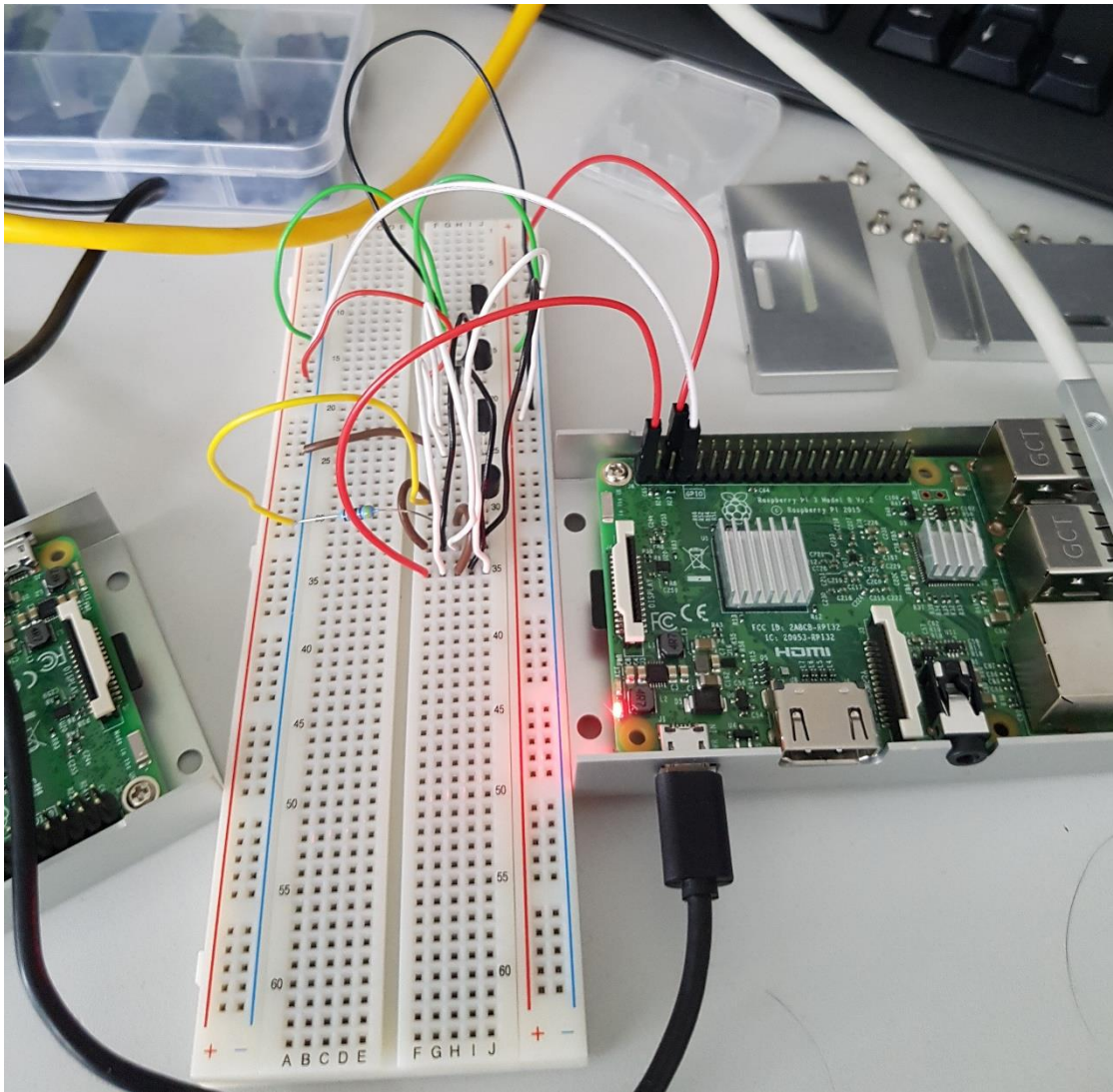


Abbildung 3-12
Versuchsaufbau 1-Wire Bus (Schaltplan siehe Anhang 10.2.1.1)

Auf Grund der besseren Leistung bei hoher Sensoranzahl wurde der „*active mode*“ ohne zusätzliche Spannungsversorgung gewählt, da die Leitungslängen so gering sind und die Sensoranzahl weit unter 100 liegt, so dass dies einen Overkill darstellen würde.

Die 1-Wire Sensoren wurden im Serverraum 926 wie folgt angebracht (Schaltplan siehe: 10.2.1.1)

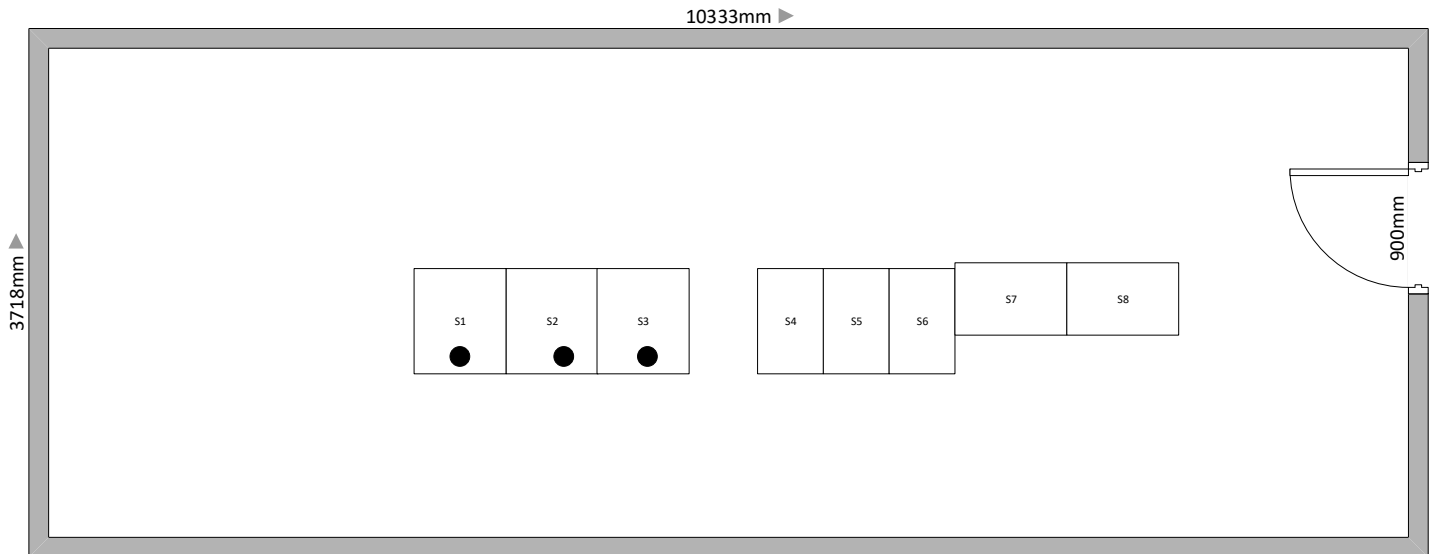


Abbildung 3-13
Positionierung der DS1820 1-Wire Sensoren im Serverraum 926

Tabelle 3-3
Höhenangaben vom Boden aus / Legende zum Raumplan

Schrank	Sensorname	Höhe
1	R01 oben	1,96 Meter
2	R02 oben	1,92 Meter
3	R03 oben	1,82 Meter
3	R03 unten	0,5 Meter

Die Sensoren wurden von dem System automatisch erkannt und von dem *read1wire* Programm (siehe 3.6.4.1) automatisch in der Sensortabelle vermerkt (Bezeichnungen, Positionen und das zugehörige Messsystem mussten manuell nachgetragen werden), so dass sofort mit den Messungen begonnen werden konnte:

Tabelle 3-4
Auszug aus der Tabelle „sensor“ der Datenbank „serverraum_temperaturueberwachung“

sensorID	sensorName	sensorKennung	sensorPosition
9	R01	10-0008034144a7	Rack1
10	R03o	10-00080341ee83	Rack3 oben
11	R03u	10-0008034187c2	Rack3 unten
12	R02	10-000803431045	Rack2

Im Serverraum 079 wurden die 1-Wire Sensoren wie folgt geplant:

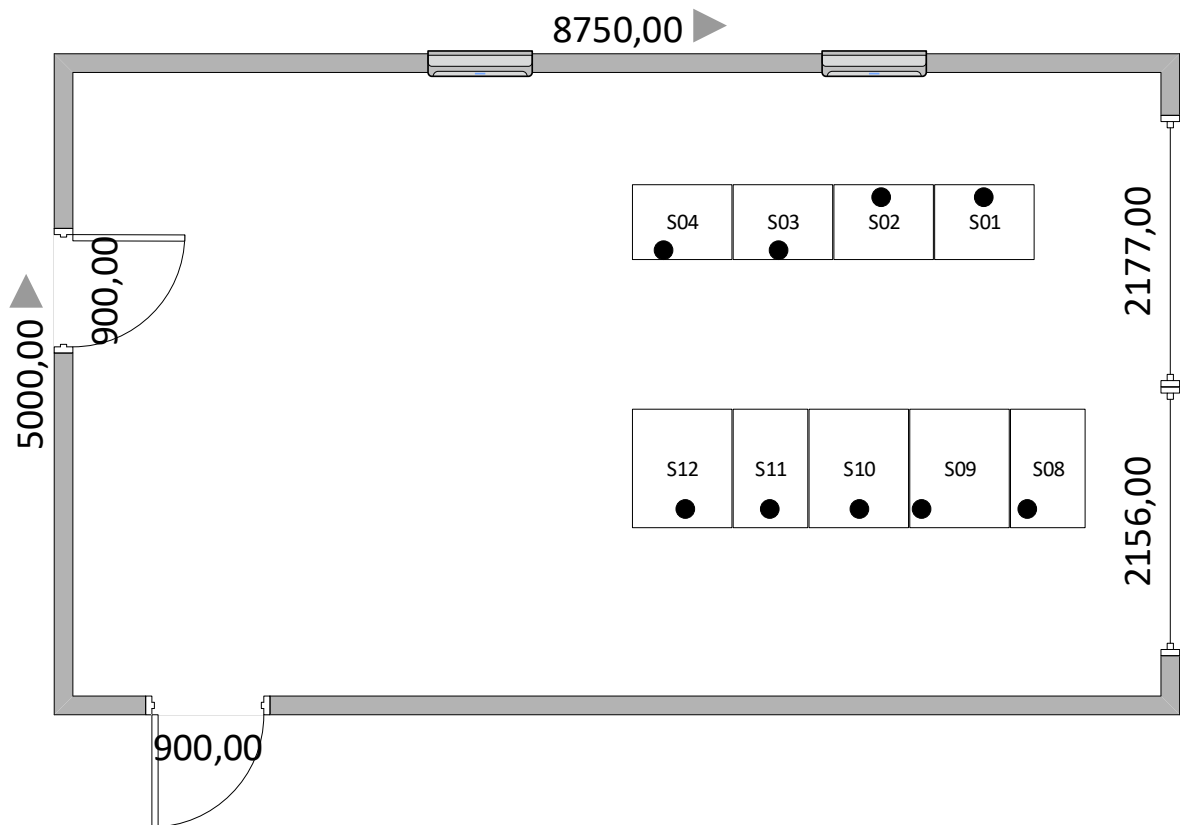


Abbildung 3-14
Positionierung der DS1820 1-Wire Sensoren im Serverraum 079

Tabelle 3-5
Höhenangaben vom Boden aus / Legende zum Raumplan

Schrank	Sensorname	Höhe
1	R01 oben	1,7 Meter
2	R02 oben	1,7 Meter
3	R03 oben	1,87 Meter
4	R04 oben	1,8 Meter
8	R08 mitte	1,2 Meter
8	R08 unten	0,5 Meter
9	R09 oben	1,5 Meter
9	R09 unten	0,95 Meter
10	R10 mitte	1,35 Meter
11	R11 oben	1,85 Meter
12	R12 oben	1,4 Meter
12	R12 mitte	1,2 Meter

Jedoch konnten aufgrund von Ressourcenmängeln nicht alle geplanten Sensoren angeschlossen werden:

Tabelle 3-6

Auszug aus der Tabelle „sensor“ der Datenbank „serverraum_temperaturueberwachung“

sensorID	sensorName	sensorKennung	sensorPosition
1	R09o	10-000803429824	Rack 9 oben
2	R09u	10-0008034271ce	Rack 9 unten
3	R12m	10-000803422999	Rack 12 mitte
4	R10o	10-000803421d63	Rack 10 oben
5	R04o	10-00080341e2d3	Rack 4 oben
6	R11o	10-00080341db65	Rack 11 oben
7	R03o	10-00080341fd74	Rack 3 oben
8	R08u	10-00080341b1a8	Rack 8 unten

3.5.1.1.2 Luftfeuchtigkeitssensoren

Als Luftfeuchtigkeitssensoren sind DHT11 und DHT22 Sensoren angedacht.

Im Serverraum 078 sind bereits DHT22 Sensoren verlegt (siehe Abbildung 3-1) doch nicht mit dem System verbunden. Für die Stromversorgung wurde immer die orange, für die Datenpins die grüne und für die Masse jeweils die grün-weiße Faser des Kabels verwendet.

Im Serverraum 926 wurde ein DHT11 Sensor wie folgt angebracht:

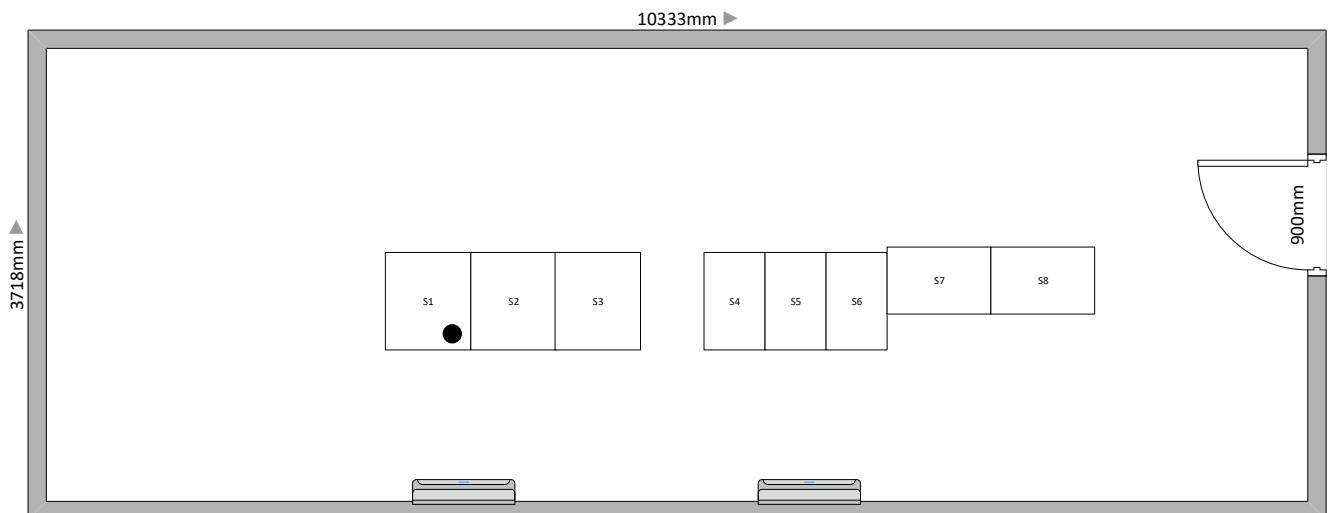


Abbildung 3-15

Positionierung der DHT 11 Sensoren im Serverraum 926

Tabelle 3-7

Höhenangaben vom Boden aus / Legende zum Raumplan

Schrank	Sensorname	Höhe
1	Rack 01 tf	2,10 Meter

Tabelle 3-8

Auszug aus der Tabelle „sensor“ der Datenbank „serverraum_temperaturueberwachung“

sensorID	sensorName	sensorKennung	sensorPosition
16	R01tf	dht926110	Rack 1 tf

3.5.1.1.3 Rauchsensoren

Die MQ-2 Gassensoren (siehe Abbildung 3-1) wurden zur neuen Position des Sensorsystems umverlegt, jedoch mangels Zeit nicht angeschlossen. Jedoch sind Mechanismen in der Software integriert, die ein unkompliziertes Einbinden ermöglichen. Die PIN-Belegung ist ident mit den schon im Serverraum 078 angeschlossenen DHT22 Sensoren.

3.5.1.1.4 Wassersensoren

Mit den Wassersensoren wird ermittelt, ob Wasser im Serverraum steht. Dazu wird eine Spannung an eine Metallplatte angelegt, welche dann in der gewünschten Höhe in ca. 5 cm Abstand von der zweiten Platte, die mit einem GPIO PIN verbunden ist, fixiert ist. Um die Wasserverteilung beziehungsweise die Höhe des Wasserstands messen zu können, sind mehrere Sensoren nötig.

Als Alternative wurden M26422 (zur Verfügung gestellt von Stulz Austria GmbH (siehe 2.2)) Sensoren verwendet.

Der M26422 Sensor wurde im Serverraum 926 wie folgt angebracht:

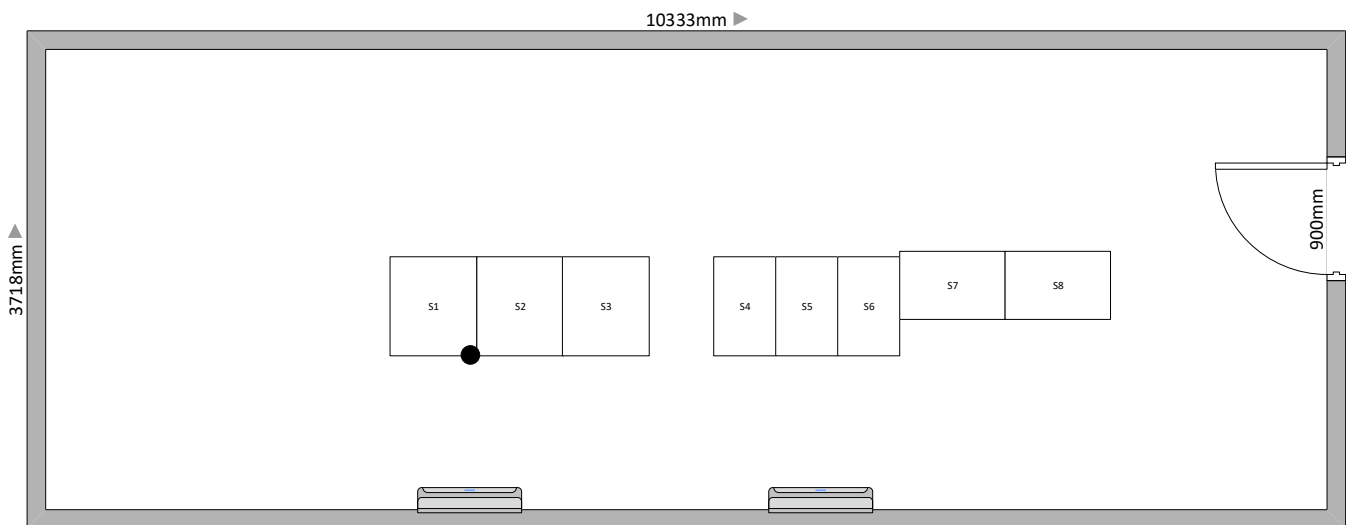


Abbildung 3-16
Positionierung des M26422 Sensors im Serverraum 926

Tabelle 3-9
Höhenangaben vom Boden aus / Legende zum Raumplan

Schrank	Sensorname	Höhe
1	Wasser Klima Fenster	0 Meter

Tabelle 3-10

Auszug aus der Tabelle „sensor“ der Datenbank „serverraum_temperaturueberwachung“

sensorID	sensorName	sensorKennung	sensorPosition
15	wasserfenster	wasser92611	Wasser Klima Fenster

Die M26422 Sensoren wurden im Serverraum 078 wie folgt angebracht:

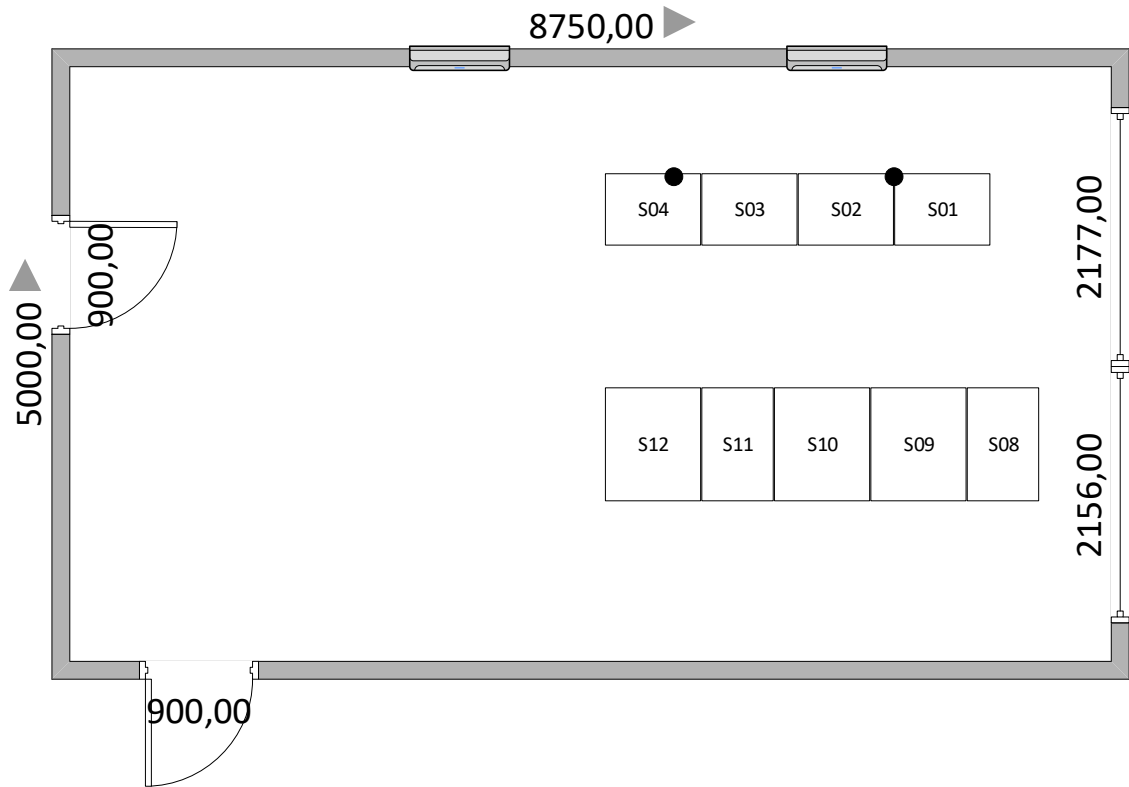


Abbildung 3-17

Positionierung der M26422 Sensoren im Serverraum 078

Tabelle 3-11

Höhenangaben vom Boden aus / Legende zum Raumplan

Schrank	Sensorname	Höhe
1	Wasser Klima Fenster	0 Meter
4	Wasser Klima Mitte	0 Meter

Tabelle 3-12

Auszug aus der Tabelle „sensor“ der Datenbank „serverraum_temperaturueberwachung“

sensorID	sensorName	sensorKennung	sensorPosition
13	wassermitte	wasser07810	Wasser Klima Mitte
14	wasserfenster	wasser07811	Wasser Klima Fenster

3.5.1.2 Relaisansteuerung

Es war ein Skript vorgesehen, das mit zwei Argumenten (der Relaisnummer und dem Befehl) aufgerufen wird. Es steuert den entsprechenden Output an, um das Relais zu schalten, dies sollte bei Notabschaltungen zum Einsatz kommen.

Eine weitere Möglichkeit stellt eine Schnittstelle zum Bussystem der Schule dar. Diese muss allerdings noch mittels einer Machbarkeitsstudie überprüft werden.

3.5.1.3 Warnsystem

Das Warnsystem informiert den Systemadministrator über das Verlassen des Normbereichs im Takt von 30 Minuten, der mit Hilfe eines Kommandos angepasst werden kann. Weiters gibt es eine tägliche Benachrichtigung, die um 18 Uhr gesendet wird und die die Durchschnittstemperaturen des letzten Tages enthält. Zusätzlich können auch Notabschaltungen durchgeführt werden.

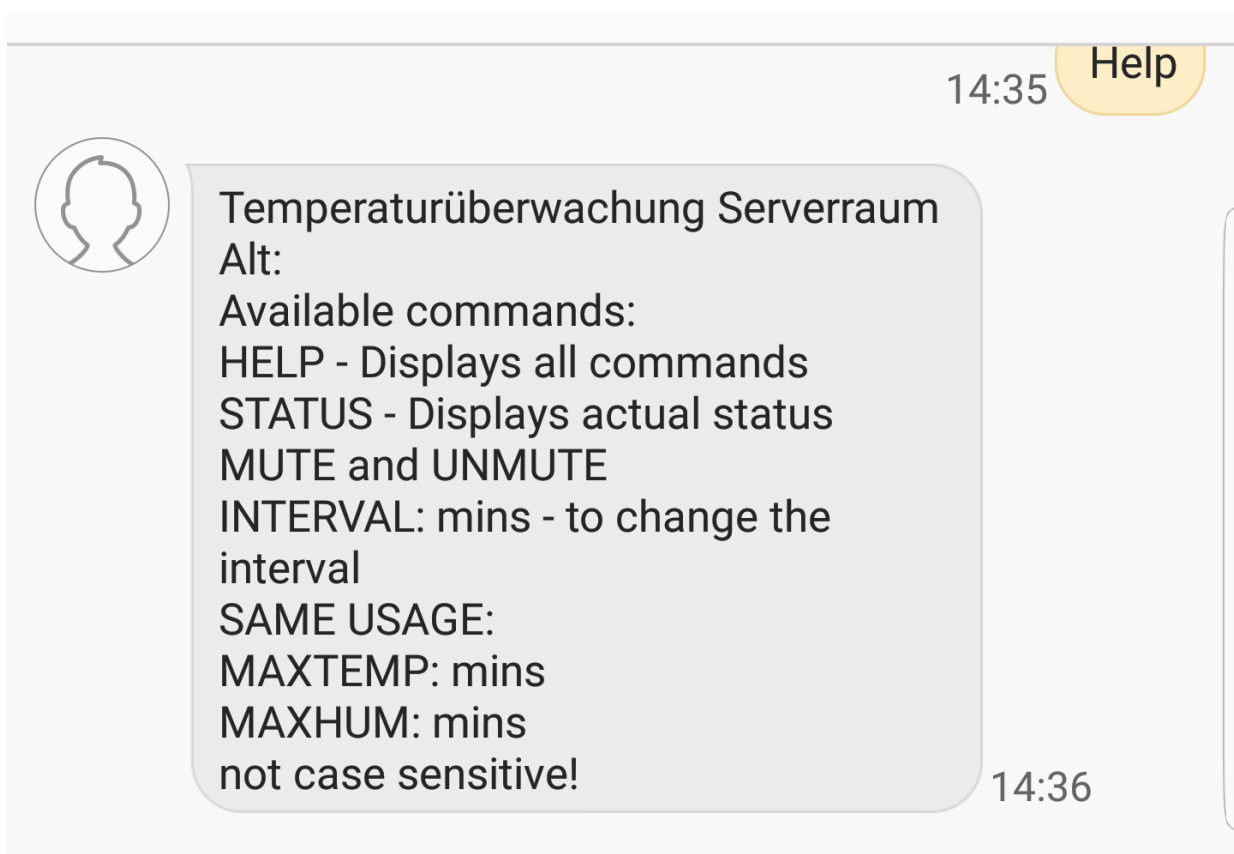


Abbildung 3-18
Verfügbare Kommandos

3.5.1.3.1 Informationssystem

Beim Überschreiten der vordefinierten Werte über einen Zeitraum von 5 Minuten wird eine SMS gesendet, in der der betroffene Raum und Sensor benannt wird, als auch der Durchschnittswert über die letzten 5 Minuten enthalten ist. Werden mehrere Werte gleichzeitig überschritten, wird dies in einer Nachricht zusammengefasst.

Kann eine SMS innerhalb von 225 Sekunden (3:45 Minuten) nicht versendet werden, so wird sie per SCP an ein anderes SMS Gateway übertragen, das diese dann zu senden versucht. Versendete SMS erscheinen auch im Webinterface unter Benachrichtigungen.

3.5.1.3.2 Interaktive Steuerung

Wird eine SMS gesendet und vom System die Notwendigkeit einer Abschaltung erkannt, so kann der Administrator mit einem „JA“ oder „NEIN“ darüber verfügen, ob der Notabschaltplan initialisiert werden soll. Erfolgt nach 3 Nachfragen, die im Abstand von Intervall / 2¹ gesendet werden, keine Antwort, so beginnt das System von sich aus mit den Abschaltungen.

Außerdem kann das Intervall von Warnungen bzw. von Abfragen per SMS geändert werden, weiters kann der Administrator mit dem Kommando „STATUS“ den aktuellen Status aller Serverräume abfragen.

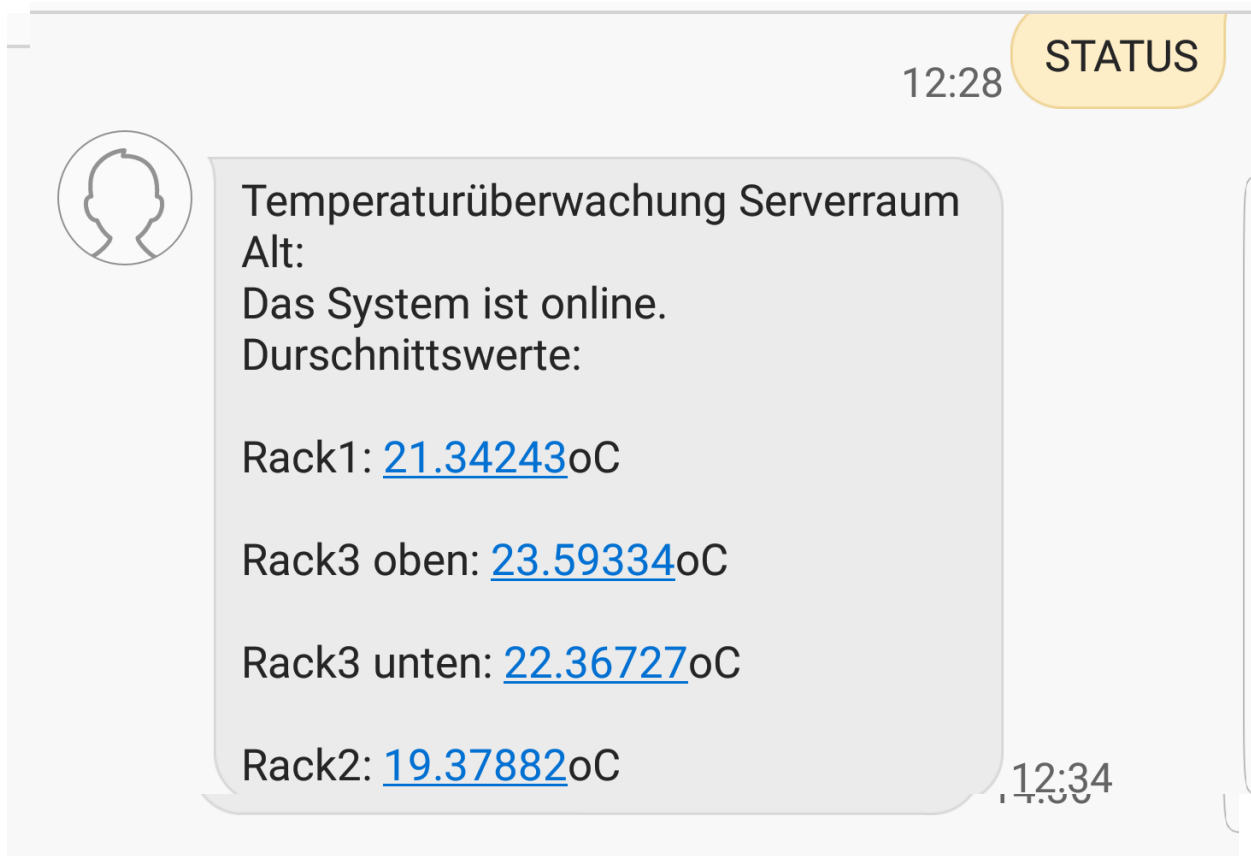


Abbildung 3-19
Screenshot des Status Kommandos

¹ Die Hälfte vom normalen Benachrichtigungsintervall

3.5.2 Notabschaltung

Um die IT Ressourcen der Schule im Notfall bestmöglich zu schützen, ist es notwendig, ein Modul zu entwickeln, welches automatisch bei Überschreiten der Werte Abschaltungen initiiert oder im Ernstfall den Strom der betroffenen Ressourcen komplett ausschaltet.

3.5.2.1 Abschaltplan

Die Reihenfolge setzt sich zusammen aus: Priorität (min. 0, max. 10) und Hitzeerzeugung (min. 1, max. 3), wobei die Gesamtpriorität wie folgt berechnet wird:

$$\text{Gesamtpriorität} = \text{Prioritätsindex} / \text{Hitzeerzeugungsindex}$$

Die Gesamtpriorität stellt also ein direktes Verhältnis aus Priorität und Hitzeerzeugung dar.

Tabelle 3-13
Erklärung des Hitzeerzeugungsindexes

Intervall	Beschreibung
[0 ; 1]	Keine / kaum messbare Hitzeerzeugung
]1 ; 2]	Akzeptable /normale Hitzeerzeugung
]2 ; 3]	Hohe Hitzeerzeugung

3.5.2.1.1 Serverraum 078

Tabelle 3-14
Reihung der Ressourcen nach Ausschaltzeitpunkt mit Priorität

Bezeichnung	Ort	Prioritätsindex	Hitzeerzeugungsindex	Gesamtpriorität
VLAB	Rack 8 und 9	0	3	0
VIRL	Rack 12	2	2	1
Schüler Umgebung	Rack 1-3	3	1	3
Produktivswitches	Rack 4	4	1	4
Diplomanden-rack	Rack 10 und 11	9	2	4,5

3.5.2.1.2 Serverraum Alt

Ein Notfallabschaltplan erfordert eine Simulation des Ausfalls der Klimaanlage und eine Priorisierung der Ressourcen. Da allerdings im Serverraum 926 keine der beiden Anforderungen erfüllt wird, kann kein Notfallabschaltplan errechnet werden.

3.5.2.2 Abschaltart

Die Abschaltart wird je nach Dringlichkeit gewählt, wenn möglich werden die Server und die darauf befindlichen VMs sanft heruntergefahren, bei Netzwerkgeräten, wie Switches, gibt es nur einen Hard Shutdown.

3.5.2.2.1 Soft Shutdown

Mittels einiger Befehle werden zuerst die VMs (falls vorhanden) heruntergefahren und dann erst der Host. Abschließend wird das Relais geschaltet, um minimalen Stromverbrauch zu gewährleisten.

3.5.2.2.2 Hard Shutdown

Hierbei wird das Relais, an dem das auszuschaltende Gerät hängt, geschaltet. Dazu wird die in 3.5.1.2 beschriebene Technologie verwendet.

3.5.3 Paketverwaltung

Um die Komplexität beim Neuaufsetzen des Systems zu verringern, gibt es ein *install* Skript, das interaktiv einige erforderliche Informationen einholt (SIM Karten PIN, die zu konfigurierenden IP-Adress-Informationen, den gewünschten Hostnamen), alle erforderlichen Pakete installiert, die Datenbank erstellt, das Webinterface lädt und alle Dienste startet. Der Nutzer muss danach nichts mehr konfigurieren, das System ist nach dem Ausführen des Skripts voll funktionsbereit und gehärtet.

3.5.4 Visualisierung

Eine auf dem kostenfreien *Bootstrap Framework* basierende Website läuft jeweils lokal auf den Überwachungssystemen und dient der Anzeige der aktuellen Werte, eines Durchschnittsverlaufs sowie der versendeten Nachrichten im *Notification* Bereich.

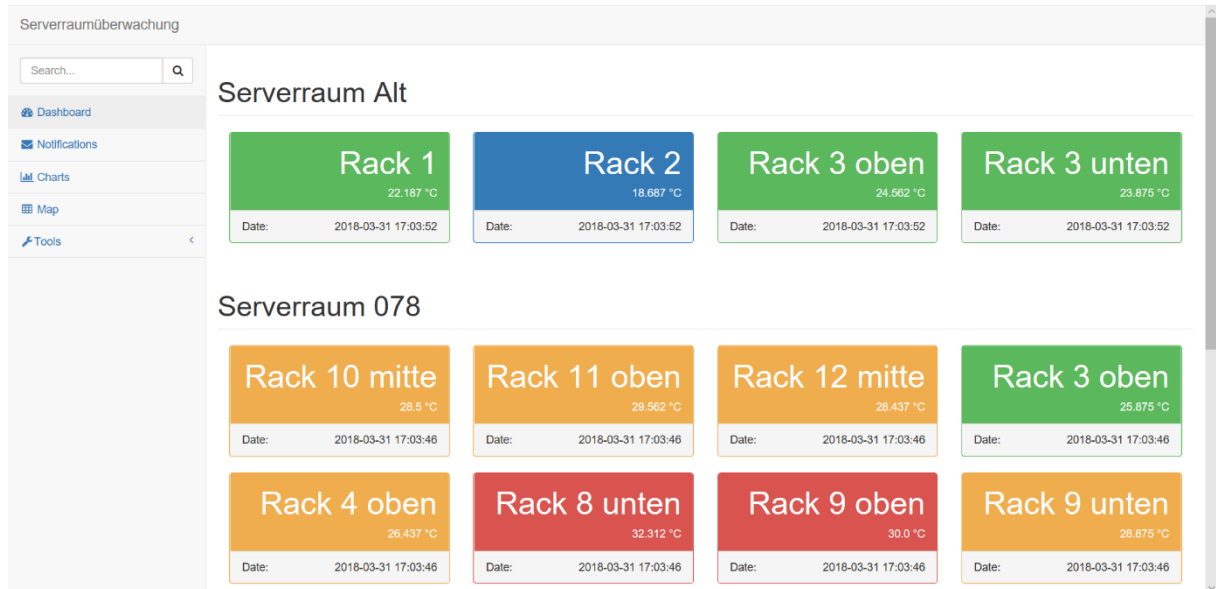


Abbildung 3-20
Screenshot der Index Seite

3.5.4.1 Allgemeine Beschreibung

Das read1wire Skript liest die aktuellen Werte ein und legt diese in einem MySQL Datenbankschema mit dem Namen `serverraum_temperaturueberwachung` ab. Außerdem werden die Daten in einer XML Datei abgespeichert. Diese Datei wird dann im Webinterface geladen und baut mittels XSL den HTML Code für die aktuellen Werte auf. In den *huge* Boxen wird außerdem ein Attribut nach dem Schema „data-temp“ gespeichert, welches zum Erstellen des Durchschnittstemperaturverlaufgraphen dient. Die Nachrichten werden mittels PHP aus einer Datenbank ausgelesen, in welche sie beim Senden gespeichert werden. Wenn man den Cursor über sie bewegt, öffnet sich ein *Popover* und man kann die Nachricht lesen. Außerdem ist es möglich, alle Nachrichten in einem speziell dafür eingerichteten Tab einzusehen.

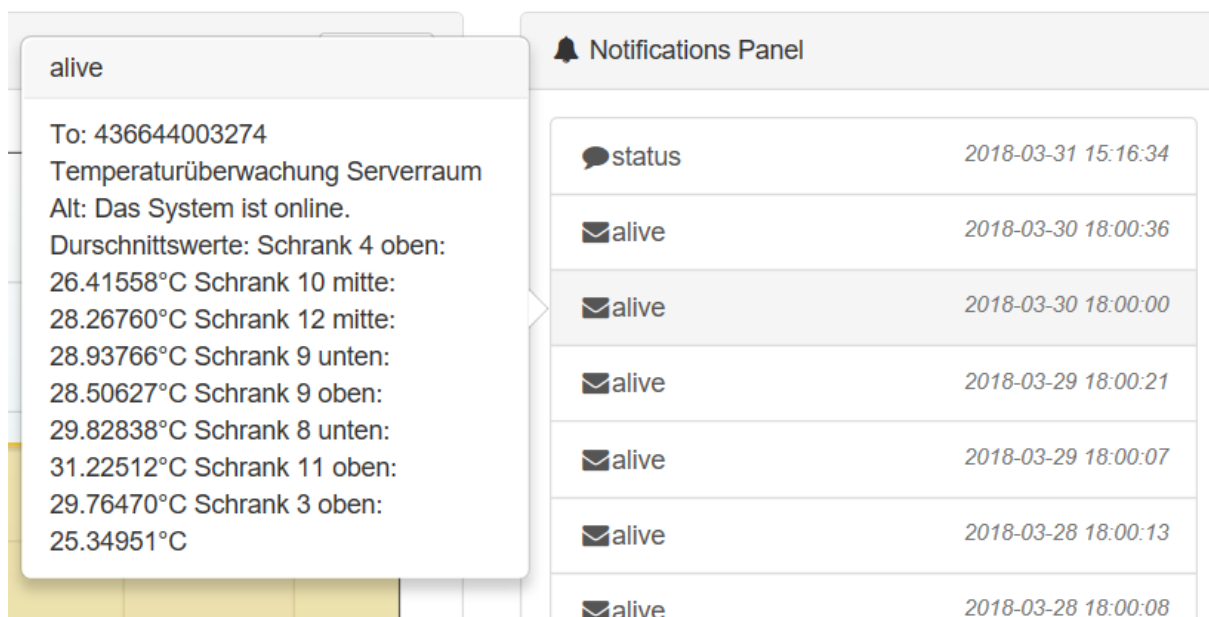


Abbildung 3-21
Notifications Panel (spezieller Tab siehe 3.6.5.2.4)

Um mehrere Serverräume in den Webinterfaces der einzelnen Systeme darzustellen, wird anhand der Node-List² der Inhalt dynamisch generiert. Allerdings werden immer oben alle aktuellen Messwerte dargestellt und darunter der Verlauf.

3.5.4.2 Synchronisierung der Daten

Um eine Synchronisierung der Daten zu ermöglichen, war es notwendig, die Datenbank flacher zu gestalten und den *primary key* der Messungstabelle als kombinierten *primary key* aus Zeit und Sensor zu entwerfen. Zum Übernehmen der Messwerte von der Datenbank „serverraum_temperaturueberwachung“ in die „serverraum_temperaturueberwachung“ wurde ein Python Skript entwickelt und die anderen stationären Werte wie Sensorbeschreibungen und Raumdaten wurden per SQL Skript überspielt.

² in der Überwachungsdatenbank enthalten

Damit die Synchronizität der Daten gewährleistet ist und in allen Webinterfaces dieselben Daten angezeigt werden können, gibt es ein *Code Snippet*, welches Datensätze an alle Server in der Node-List¹ überträgt. Um alle aktuellen Daten anzuzeigen, verweisen die Webseiten der einzelnen Systeme aufeinander, um keine zusätzliche Erstellung / Übertragung der erforderlichen XML Dateien zu benötigen.

3.5.5 Redundanz

Da jedes System das Risiko eines Ausfalls mit sich bringt und der Ausfall der Überwachung ein besonders kritisches Problem wäre, ist es notwendig, mehrere Systeme mit identer Funktion parallel zu betreiben, ohne dass diese voneinander abhängig sind.

3.5.5.1 Mehrere unabhängige Systeme

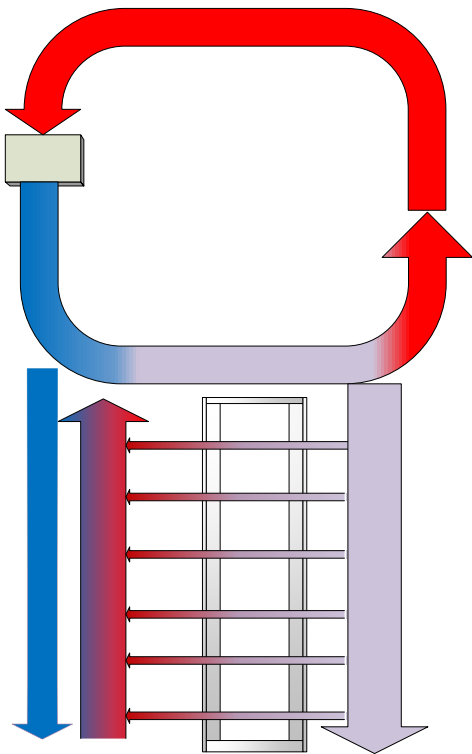
Mehrere idente Systeme laufen gleichzeitig, sind jedoch nicht in der Lage, miteinander zu kommunizieren. Zusätzlich können andere Komponenten nicht neu gestartet werden, sollten sie eine Fehlfunktion erleiden.

3.5.5.2 Clustered System mit Self Recovery

Die unabhängigen Systeme kommunizieren miteinander, tauschen ihre Messwerte aus und können bei Bedarf die Funktion eines Teils vollautomatisch übernehmen. Zusätzlich wird versucht, ausgefallene Komponenten automatisch durch Neustarten des gesamten Teilsystems zu reparieren. Ist dies nicht möglich, so wird eine Benachrichtigung an den Systemadministrator versandt. Diese wird im 12 Stunden Takt wiederholt.

3.5.6 Kühlsystem

Natürlich sind auch Konzepte zur effizienten Kühlung für den reibungslosen Betrieb eines oder mehrerer Serverräume erforderlich. Um ein solches zu entwickeln, wurde das Unterprojekt „Cooldown“ eingesetzt. Durch das Einsetzen des von dem Projekt entwickelten Kühlsystems könnte eine bessere Leistung der Hardware, eine längere Haltbarkeit der Hardware und der Klimaanlage sowie auch eine Verringerung der Brandgefahr erzielt werden.



3.5.6.1 Derzeitige Situation

Bei Nachforschungen – im Rahmen dieses Unterprojektes – wurde ermittelt, dass der Luftfluss im Serverraum Alt / Raum 926 suboptimal ist:

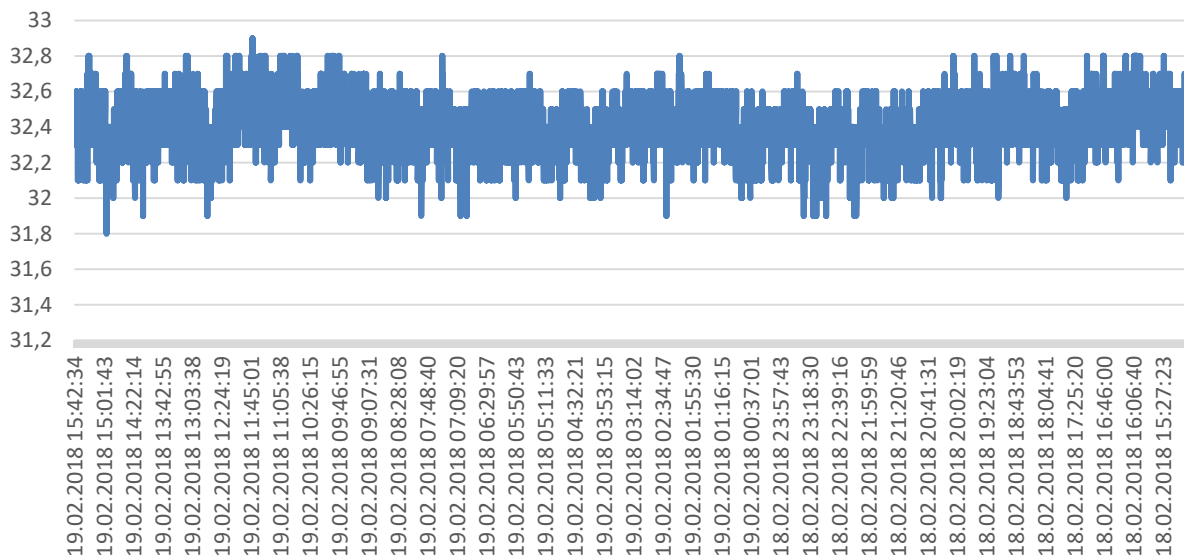
Wie man erkennen kann, zirkuliert die kühle Luft im Bereich der Wand und wird von der warmen Abluft der Server wieder nach oben gedrückt.

Von dort aus strömt die nun leicht erhitzte Luft über die Serverschränke und wird von den Servern selbst auf der anderen Seite teilweise wieder hinunter gesaugt. Der verbleibende Restanteil geht über in den oberen Hitzekreislauf.

Der Kühlvorgang ist also mehr indirekt und dadurch sehr ineffizient. Dies kann auch an den Messwerten unten erkannt werden.

Abbildung 3-22 Aktueller Kühlkreislauf Serverraum 926

Abbildung 3-23



Sensor Rack3 oben (bei den Produktivservern)

3.5.6.2 Drehen der Server zum Korrigieren des Kühlkreislaufes

Durch das einfache Drehen der Server um 180 Grad könnte man den Luftstrom auf den folgenden korrigieren:

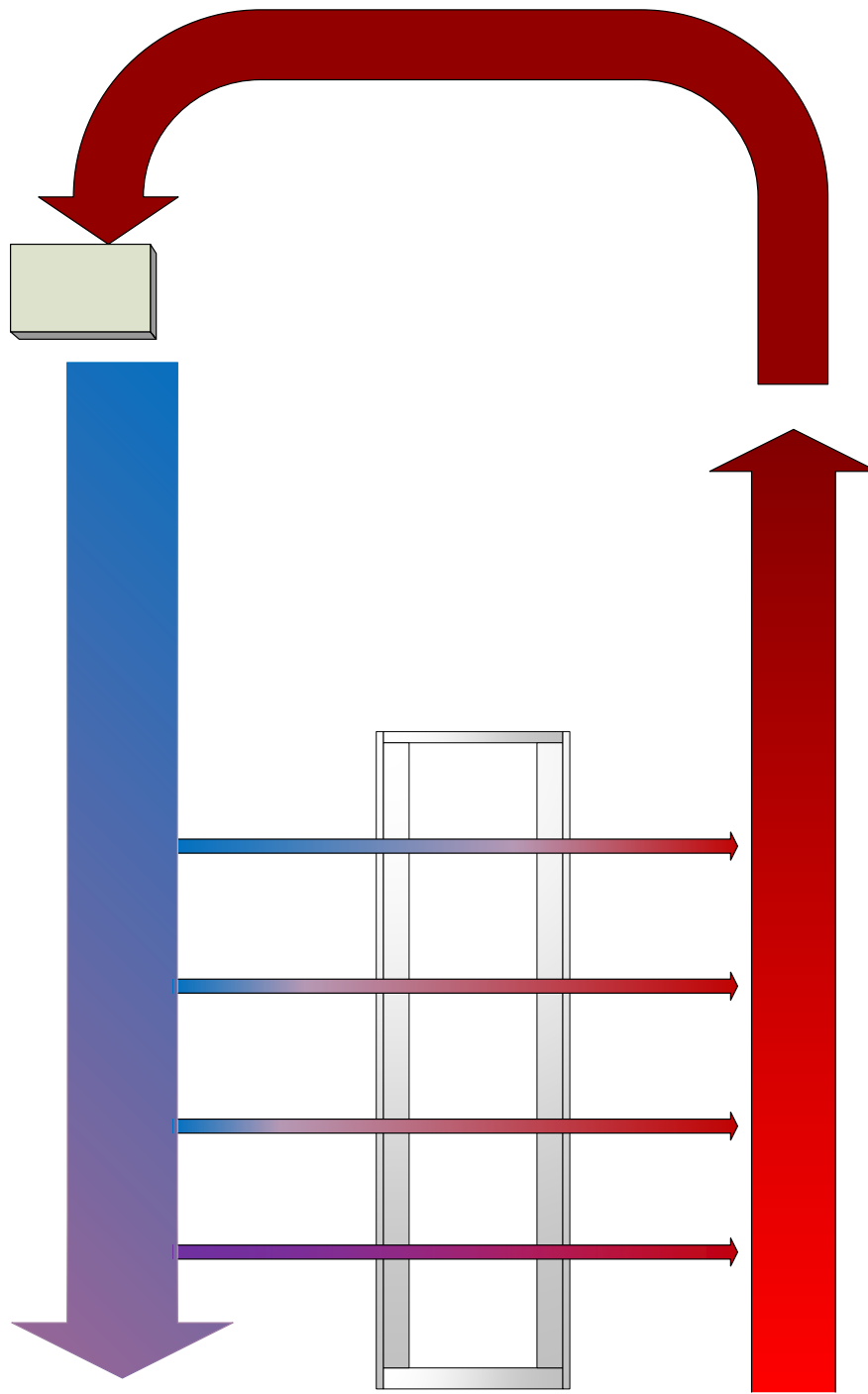


Abbildung 3-24
Neuer Luftströmungskreislauf

3.5.6.3 Kalt- / Warmgang Konzept

Das alleinige Drehen der Server bringt – wie man sehen kann – nur geringe Verbesserungen, da die oberen Server nämlich mehr kühle Luft bekommen als die unteren. Deswegen ist es notwendig, die Luftströmung so weit zu manipulieren, dass es zu einer gleichmäßigen Verteilung der kühlen Luft kommt.

Zuerst könnte man in einer Art Feldstudie durch direkte Zuleitungen zu den Servern, wie unten verbildlicht, eine bessere Verteilung erzielen. Jedoch würde das zu massiven Einschränkungen der Bewegungsfreiheit im Serverraum führen.

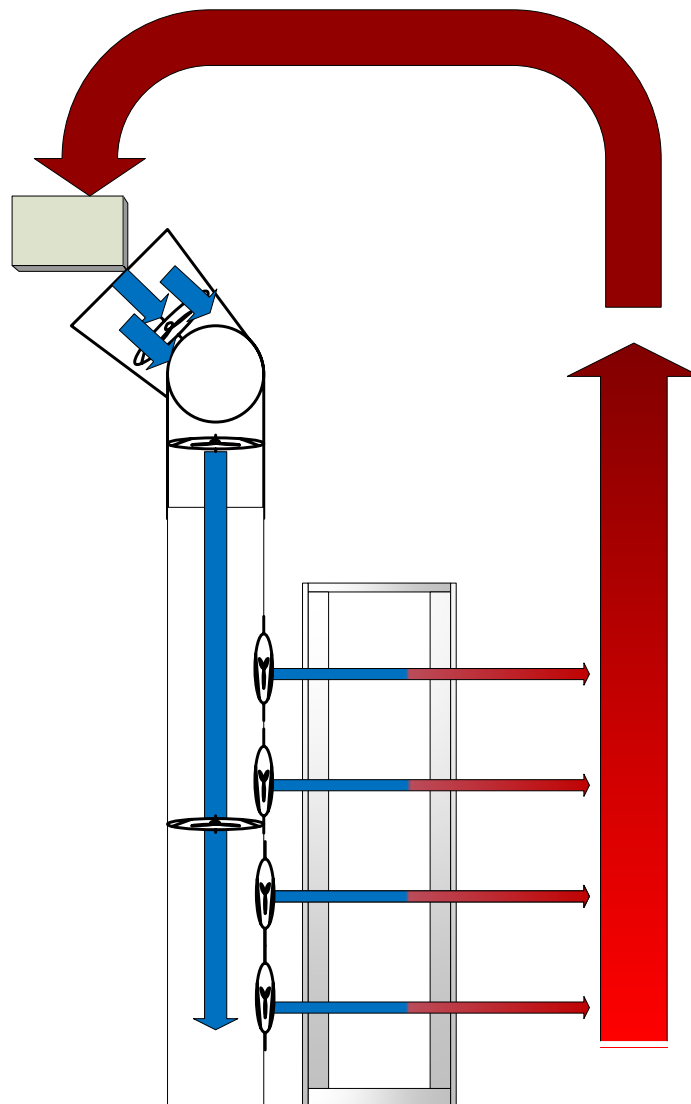


Abbildung 3-25
Luftfluss nach der Installation der direkten Zuleitungen

Um einen optimalen Luftfluss bei minimaler Einschränkung zu gewährleisten, muss man durch genaue Ausrichtung der Lüfter, den Luftstrom bei Einsatz der Rohre simulieren. Dies könnte ungefähr so aussehen:

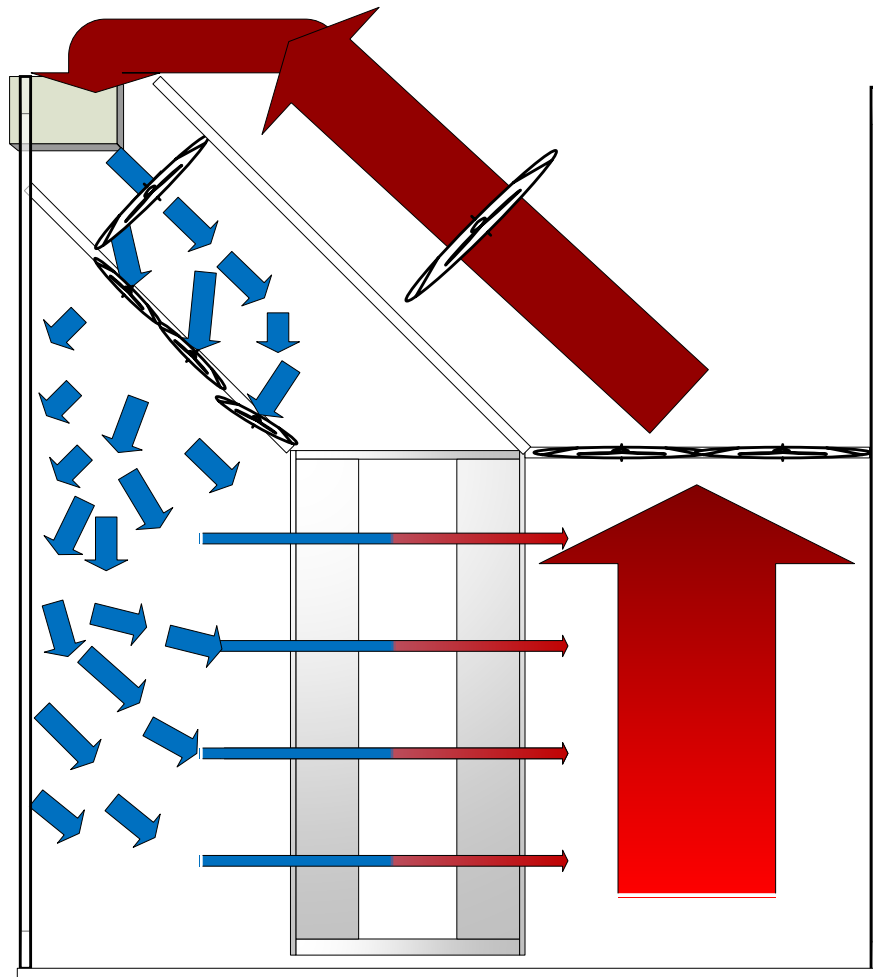


Abbildung 3-26
Luftstrom bei genau ausgerichteten Ventilatoren

Hierbei ist auch die Rückleitung der heißen Luft durch Ventilatoren gewährleistet. Durch die dadurch entstehende Sogwirkung kann ein zusätzlicher Kühleffekt erzielt werden. Der zu kühlende Raum wird durch eingezogene Wände verkleinert und die Klimaanlage werden somit optimal genutzt:

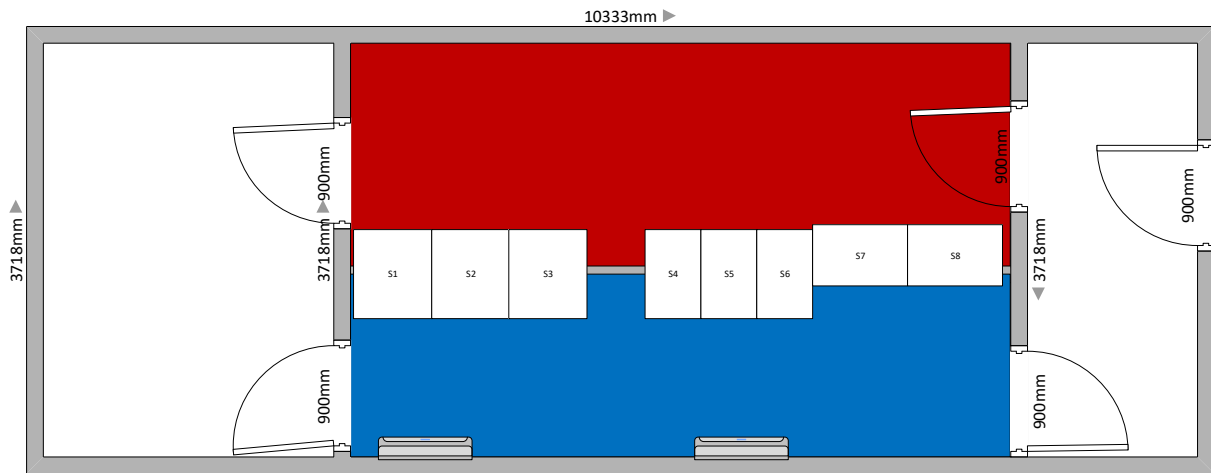


Abbildung 3-27
Kalt- / Warmgangkonzept Vogelperspektive

3.5.6.4 Erweiterung Außenluftkühlkonzept

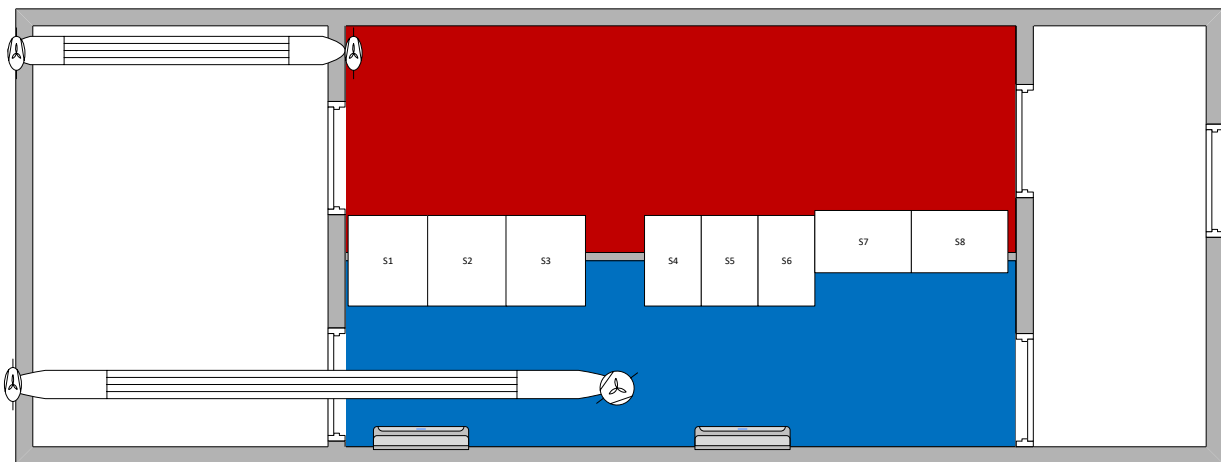


Abbildung 3-28
Vogelperspektive Erweiterung Außenluftkühlkonzept

Um die Temperaturdifferenz zwischen Innen- und Außenraum optimal zu nutzen, ist es notwendig, die Außenluft direkt in den Serverraum zu blasen und die erhitzte Luft wieder nach außen abzuführen. Natürlich ist dies alleine bei einer geringeren Temperaturdifferenz nicht ausreichend und deswegen ist eine automatisierte Regelung von Klimaanlage und Außenluftkühlkonzept erforderlich. Dadurch kann eine zusätzliche Stromersparnis erreicht werden, weil die Klimaanlage, die einen sehr hohen Verbrauch hat, nicht die ganze Zeit aktiv sein muss. Im Sommer muss eine ausreichende Abdichtung der Öffnungen nach außen eingesetzt werden, damit keine Kühlleistung für das Klimatisieren des Schulhofes verloren geht.

3.6 Produkt

Nachfolgend ist eine Dokumentation unseres Produktes dargestellt:

3.6.1 Struktur

Die Hardwarearchitektur und Beziehungen unserer Systeme lassen sich vereinfacht so darstellen:

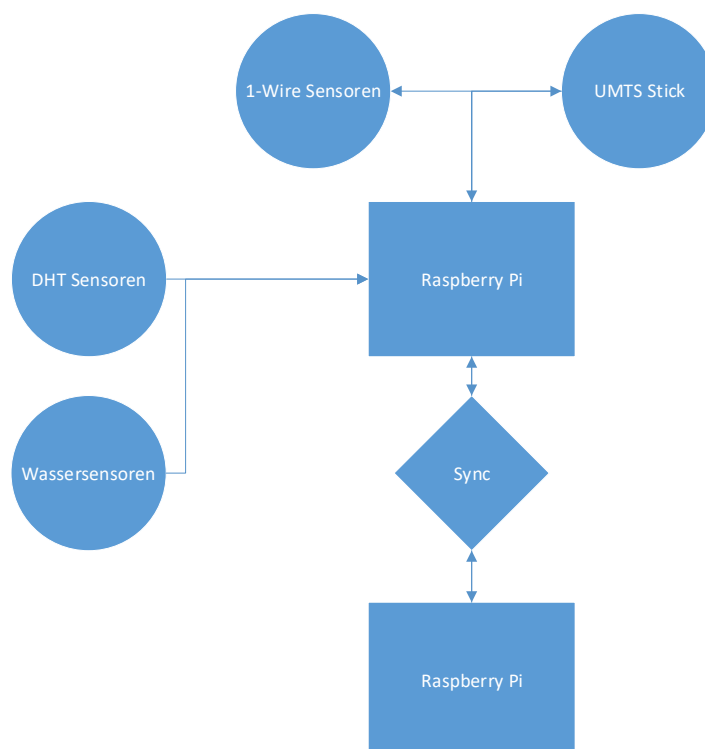


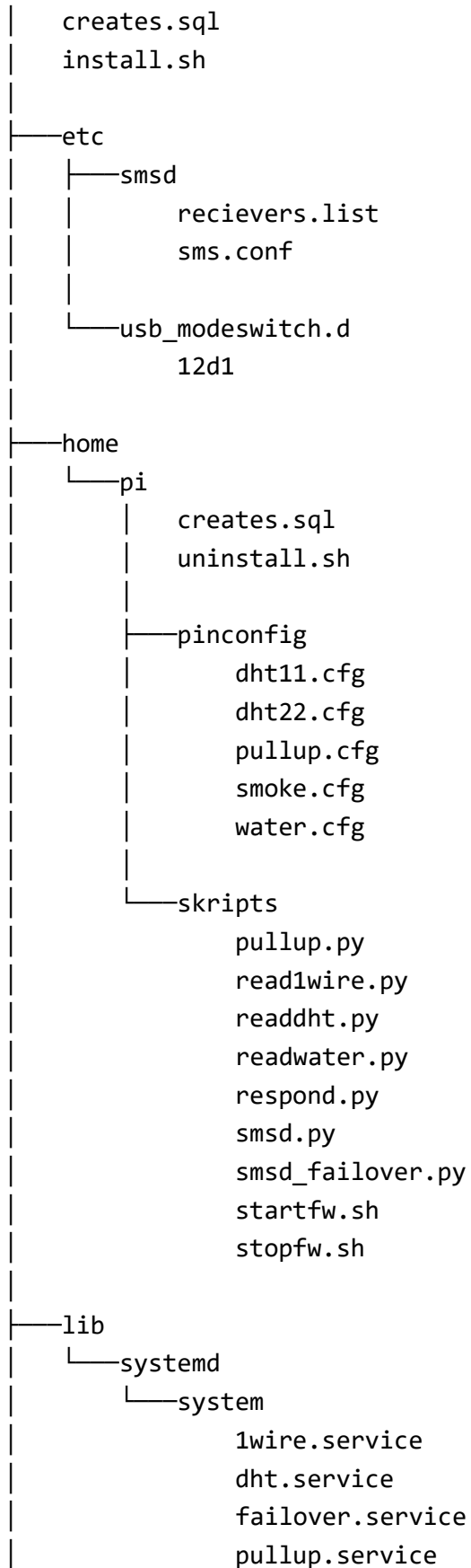
Abbildung 3-29
Schematische Darstellung der Hardware

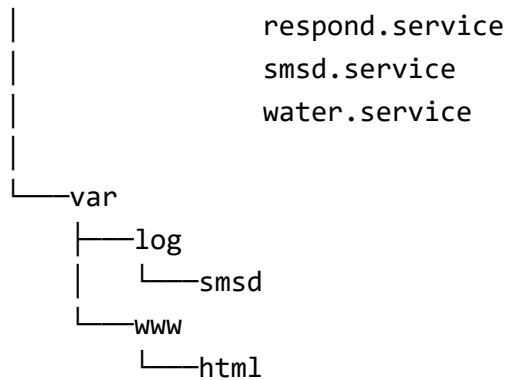
Hierbei ist zu beachten, dass das zweite System, um Platz zu sparen, minimiert dargestellt ist. Die an das zweite System angeschlossenen Objekte sind jedoch ident, sie können also einfach dazu gedacht werden. Die Richtung der Pfeile gibt die Art der Verbindung an:

Pfeil weg vom Objekt= Output Pfeil hin zum Objekt = Input Pfeil in beide Richtungen = Input & Output

Grundsätzlich besteht unser Produkt aus einem Webinterface, zwei Datenbanken und 4 Services.

Die Ordnerstruktur des Produktes:





Der html Ordner ist zur Übersichtlichkeit nur minimiert dargestellt, auf das Webinterface wird später noch genauer Bezug genommen.

3.6.2 Das Install Skript

Zum Starten aller Services, dem Erstellen der Datenbanken und Installieren aller erforderlichen Pakete wurde ein Installationsskript erstellt, um die Neuinstallation des Systems einfach und unkompliziert zu gestalten:

```

apt update && apt upgrade -y && apt install -y apache2 mysql-server php
php-mysql php-xml smstools python3-pip python3-pymysql git firewallld
python3-rpi.gpio build-essential libusb-dev
systemctl enable firewallld
systemctl start firewallld
cd /
git init
git remote add origin git@fabian.stei-ner.net:root_ueberwachung
git pull origin master
mv /etc/usb_modeswitch.d/12d1 /etc/usb_modeswitch.d/12d1:1f01
mkdir /var/log/smsd/
mkdir /var/log/lwire
mkdir /var/log/water
mkdir /var/log/pullup
mkdir /var/log/dht
mkdir /home/pi/incoming
mkdir /home/pi/lib
cd /home/pi/lib
apt install python-dev python-openssl python-pymysql
git clone https://github.com/adafruit/Adafruit_Python_DHT.git
cd Adafruit_Python_DHT
sudo python setup.py install
cd /
touch /home/pi/alive.txt
chown pi:pi -R /home/pi/incoming
sed -i "/bind-address.*=/c\bind-address          = 0.0.0.0"
/etc/mysql/mariadb.conf.d/50-server.cnf
  
```

```

systemctl restart mysql
mysql -e "source creates.sql"
systemctl enable 1wire
systemctl enable water
systemctl enable dht
systemctl enable smsd
systemctl enable failover
systemctl enable respond
systemctl start 1wire
systemctl start water
systemctl start dht
systemctl start smsd
systemctl start failover
systemctl start respond
echo -n "Provide PIN: "
read answer
sed -i "/device = .*/c\device = /dev/ttyUSB0" /etc/smsd.conf
sed -i "/pin = .*/c\pin = $answer" /etc/smsd.conf
systemctl restart smstools
rm creates.sql
rm install.sh
rm -rf .git
echo "# activating 1-wire\n" >> /boot/config.txt
echo "dtoverlay=w1-gpio" >> /boot/config.txt
firewall-cmd --permanent --add-service=ssh
firewall-cmd --permanent --add-service=http
firewall-cmd --permanent --add-service=https
firewall-cmd --permanent --add-service=mysql
firewall-cmd --reload
echo -n "A reboot is required to apply the configuration. Reboot? (y/n)?
"
read reb
if echo "$reb" | grep -iq "^y" ;then
    echo "Goodbye"
    systemctl reboot
else
    echo "Please reboot the system when convenient."
fi
echo "Goodbye."

```


3.6.3 Die Datenbanken

Unser Produkt verwendet zwei auf MySQL basierende Datenbanken: die „serverraum_temperaturueberwachung“ und die „messages“ Datenbank.

3.6.3.1 Datenbank „serverraum_temperaturueberwachung“

Diese Datenbank dient zum Speichern von Messwerten, Sensoren, Messsystemen und Raumbezeichnungen. Der Aufbau gestaltet sich wie folgt:

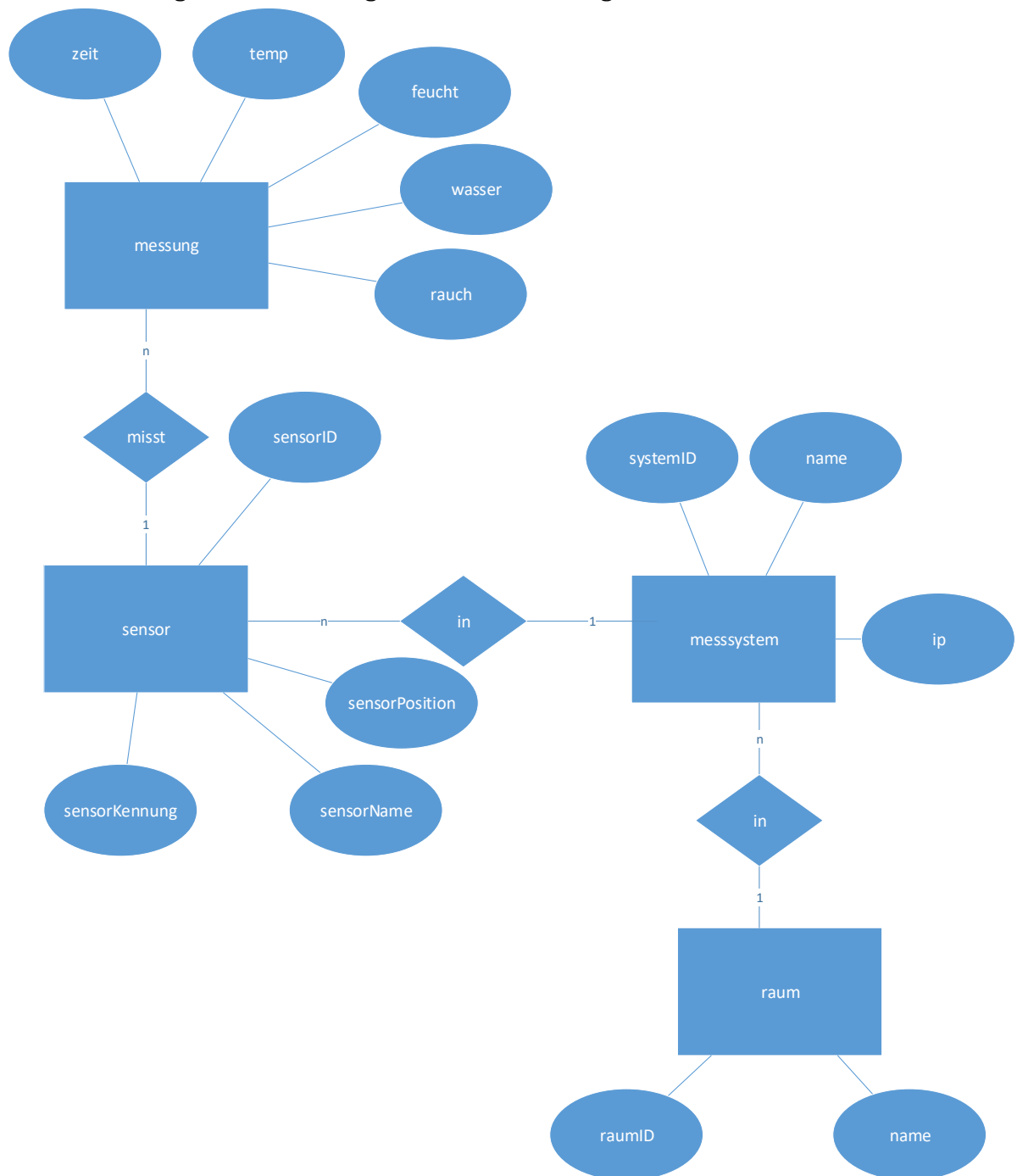


Abbildung 3-30
ER Modell der Datenbank „serverraum_temperaturueberwachung“

Nach Transformation ins Relationenmodell:

```
raum (raumID, name);
```

```
messsystem (systemID, name, ip, fk_raumID)
```

```
sensor (sensorID, sensorName, sensorKennung, sensorPosition, fk_systemID)
```

```
messung(zeit,fk_sensorID, temp, feucht, wasser, rauch)
```

3.6.3.2 Datenbank „messages“

Diese Datenbank dient zum Speichern aller vom System versandten Nachrichten:

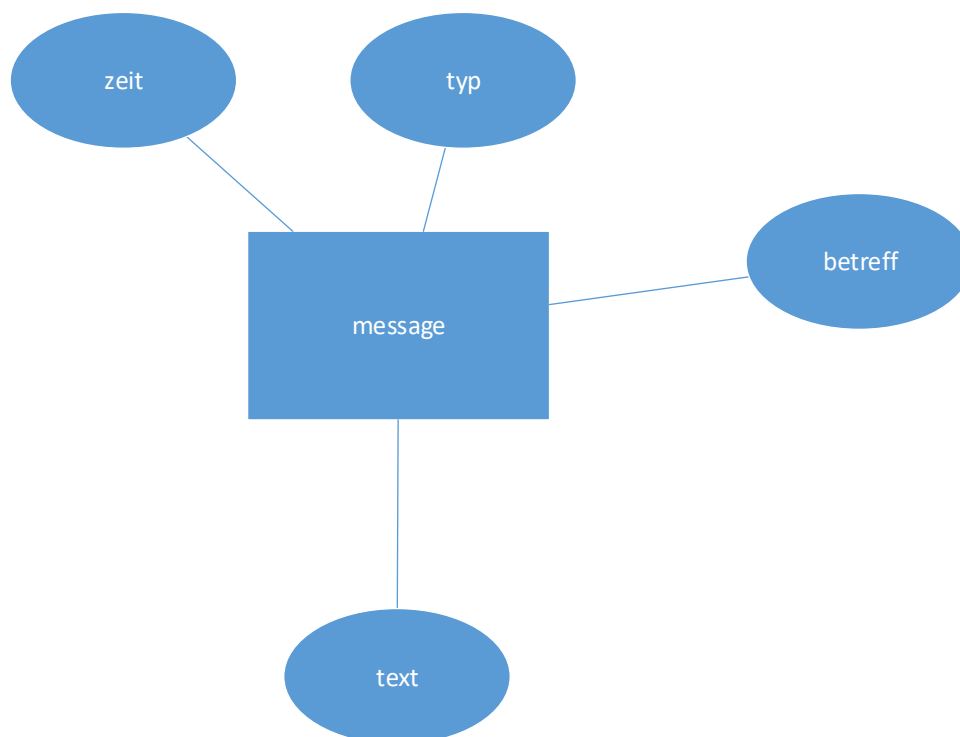


Abbildung 3-31
ER Modell der Datenbank „messages“

Nach der Transformation ins Relationenmodell:

```
message (zeit, typ, betreff, text)
```

Wenn eines der Systeme eine neue Nachricht sendet, wird automatisch in die „messages“ Datenbank aller Systeme eine Nachricht eingetragen, dazu werden die in der Tabelle „messsystem“ gespeicherten IP Adressen verwendet. Es müssen also zur vollen Funktion des Systems alle anderen Systeme in der Datenbank eingetragen sein.

3.6.4 Services

Es gibt ein Service zum Auslesen der 1-Wire Sensoren und 3 Services, die sich um das SMS Interface kümmern.

3.6.4.1 1wire

Das 1-Wire Service liest die aktuellen Messwerte aus den vom Busmaster erstellten Dateien aus und speichert sie als xml und in der Datenbank ab:

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
import pymysql
import sys
import glob
import datetime
import time
import logging
import logging.handlers
import socket

#destination for the XML File
path = '/var/www/html/xml/temp.xml'
#set up logging
logFormatter = logging.Formatter('%(asctime)s - %(levelname)s:
%(message)s')
rootLogger = logging.getLogger()
fileHandler = logging.handlers.RotatingFileHandler('/var/log/1wire/1wire.log',maxBytes=1
000000,backupCount=5)
fileHandler.setFormatter(logFormatter)
console = logging.StreamHandler()
rootLogger.addHandler(fileHandler)
rootLogger.addHandler(console)
rootLogger.setLevel(logging.INFO)

#find out local ip address (we want to tell the recievers the name of the
system)
local = ([ip for ip in socket.gethostbyname_ex(socket.gethostname())[2]
if not ip.startswith("127.")] or [{"s.connect(("8.8.8.8", 53)),
s.getsockname()[0], s.close()} for s in [socket.socket(socket.AF_INET,
socket.SOCK_DGRAM)]]][0][1]]) + ["no IP found"]][0]

while True:
    #connection stays open until an error
    try:
```

```

        db = pymysql.connect(host='localhost', user='webuser',
password='password',db='serverraum_temperaturueberwachung',autocommit=True
)

        cursor = db.cursor()
        logging.info("Connected to database")
    except Exception as e:
        logging.error(e)
        continue
    while True:
        #connect to database
        try:
            cursor.execute("select      sensorKennung,      sensorID,
sensorPosition from sensor where fk_systemID = (select systemID from
messsystem where ip = '"+local+"' ) order by sensorName;")
            sensoren = cursor.fetchall()
            logging.debug(sensoren)
            # read 1-wire slaves list
            paths = glob.glob(r'/sys/devices/w1_bus_master1/10-*')
            sql = "insert into messung (zeit , fk_sensorID, temp)
values"

            out = '<?xml version="1.0" encoding="ISO-8859-1" ?><?xml-
stylesheet type="text/xsl" href="./actualtemp.xsl"?><monitoring>'
            now = datetime.datetime.now().strftime("%Y-%m-%d
%H:%M:%S")

            logging.debug("beginning")
            #save measurements to the database and produce xml, missing
sensors will be created automatically
            for filename in paths:
                logging.debug("reading file: " + filename)
                try:
                    with open(filename + '/w1_slave', "r") as file:
                        filecontent = file.readlines()
                        stringvalue = filecontent[1].split("
")
                        temperature = str(float(stringvalue[2:]
/ 1000.0))

                        sql = sql + " ('%s',func_getSID('%s'),
%s)," % (now,str(filename).split('/')[4], temperature)
                        for sensor in sensoren:
                            if str(sensor[0]) ==
str(filename).split('/')[4]:
                                out= out + '<sensor name =
'+str(sensor[2])+'"><id>'+str(sensor[1])+</id><temp>'+str(temperature)+'
</temp><zeit>'+now+'</zeit></sensor>'

```

```

        except Exception as e:
            logging.error(e, "damn")
        out = out + "</monitoring>"
        logging.debug(out)
        with open(path, "w") as xml:
            print(out, file=xml)
        logging.info("XML File sucessfully generated")
        logging.debug(sql[: -1] + ";")
        cursor.execute(sql[: -1] + ";")
        logging.info("Data sucessfully inserted into database")
    except Exception as g:
        logging.error(g)
        logging.info("Database Connection will be reopened")
        db.close()
        break

```

Das Service kann mit den üblichen *systemctl* Befehlen gestartet, gestoppt bzw. neu gestartet werden. Der Name des Service ist „1wire“:

```

systemctl start 1wire #starten des Services
systemctl stop 1wrie #stoppen des Services
systemctl restart 1wire #neustarten des Services

```

3.6.4.2 DHT

Das DHT Service liest alle DHT 11 und DHT 22 Sensoren aus, die in den zugehörigen“ pin-configs“ angegeben wurden:

```

#!/usr/bin/python
# -*- coding: utf-8 -*-

#this skript was designed to work with python2
import pymysql
import sys
import Adafruit_DHT
import datetime
import time
import logging
import logging.handlers
import socket

#destination for the XML File
path = '/var/www/html/xml/feucht.xml'
#set up logging
logFormatter = logging.Formatter('%(asctime)s    -    %(levelname)s:
%(message)s')

```

```

rootLogger = logging.getLogger()
fileHandler = logging.handlers.RotatingFileHandler('/var/log/dht/dht.log',maxBytes=1000000,backupCount=5)
fileHandler.setFormatter(logFormatter)
console = logging.StreamHandler()
rootLogger.addHandler(fileHandler)
rootLogger.addHandler(console)
rootLogger.setLevel(logging.INFO)

```

```

#DHT11 Inputs setzen
dht11 = list()
with open('/home/pi/pinconfig/dht11.cfg', mode= 'r') as file:
    for line in file:
        dht11 = line.split(',')
        logging.info("DHT11 GPIOs are: " + str(dht11))

```

```

#DHT22 Inputs setzen
dht22 = list()
with open('/home/pi/pinconfig/dht22.cfg', mode= 'r') as file:
    for line in file:
        dht22 = line.split(',')
        logging.info("DHT22 GPIOs are: " + str(dht22))

```

```

#find out local ip address (we want to tell the receivers the name of the
system)
local = (([ip for ip in socket.gethostbyname_ex(socket.gethostname())[2]
if not ip.startswith("127.")] or [[(s.connect(("8.8.8.8", 53)),
s.getsockname()[0], s.close()) for s in [socket.socket(socket.AF_INET,
socket.SOCK_DGRAM)]]][0][1]]) + ["no IP found"])[0]

```

```

#header - figure out system name
sql = "select name from messsystem where ip = '%s';" % (local)
try:
    db = pymysql.connect(host='localhost', user='webuser',
password='password',db='serverraum_temperaturueberwachung',autocommit=True)
    cursor = db.cursor()
    logging.info("Connected to database")
    logging.debug(sql)
    cursor.execute(sql)
    systemname = cursor.fetchone()[0]
    logging.info("SystemName is: " +systemname)

```

```

        db.close()
except Exception as e:
    logging.error(e)
    sys.exit()

dhtprefix = systemname.split(' ')[1] + systemname.split(' ')[2]

while True:
    #connection stays open until an error
    try:
        db = pymysql.connect(host='localhost', user='webuser',
password='password',db='serverraum_temperaturueberwachung',autocommit=True)

        cursor = db.cursor()
        logging.info("Connected to database")
    except Exception as e:
        logging.error(e)
        continue
    while True:
        #connect to database
        try:
            cursor.execute("select      sensorKennung,      sensorID,
sensorPosition from sensor where fk_systemID = (select systemID from
messsystem where ip = '"+local+"' ) order by sensorName;")
            sensoren = cursor.fetchall()
            logging.debug(sensoren)
            # read 1-wire slaves list
            sql = "insert into messung (zeit , fk_sensorID, temp,
feucht) values"
            out = '<?xml version="1.0" encoding="ISO-8859-1" ?><?xml-
stylesheet type="text/xsl" href="./actualfeucht.xsl"?><monitoring>'
            now = datetime.datetime.now().strftime("%Y-%m-%d
%H:%M:%S")

            logging.debug("beginning")
            #save measurements to the database and produce xml, missing
sensors will be created automatically
            for pin in dht11:
                pin = int(pin)
                logging.debug("reading GPIO: " + str(pin))
                sk = 'dht'+ str(dhtprefix) + str(pin)
                try:
                    feucht, temp =
Adafruit_DHT.read_retry(Adafruit_DHT.DHT11, pin)

```

```

        sql = sql + " ('%s',func_getSID('%s'), %d,
%d)," % (now, sk, temp, feucht)
        for sensor in sensoren:
            if str(sensor[0]) == sk:
                logging.debug("here")
                out= out + '<sensor name =
"' +str(sensor[2])+' "><id>' +str(sensor[1])+'</id><temp>' +str(temp)+'</tem
p><feucht>' +str(feucht)+'</feucht><zeit>' +now+'</zeit></sensor>'
            except Exception as e:
                logging.error(e)
        for pin in dht22:
            pin = int(pin)
            logging.debug("reading GPIO: " + str(pin))
            sk = 'dht'+ str(dhtprefix) + str(pin)
            try:
                feucht, temp =
Adafruit_DHT.read_retry(Adafruit_DHT.DHT22, pin)
                sql = sql + " ('%s',func_getSID('%s'), %d,
%d)," % (now, sk, temp, feucht)
                for sensor in sensoren:
                    if str(sensor[0]) == sk:
                        logging.debug("here")
                        out= out + '<sensor name =
"' +str(sensor[2])+' "><id>' +str(sensor[1])+'</id><temp>' +str(temp)+'</tem
p><feucht>' +str(feucht)+'</feucht><zeit>' +now+'</zeit></sensor>'
                    except Exception as e:
                        logging.error(e)
                out = out + "</monitoring>"
                logging.debug(out)
                with open(path, "w") as xml:
                    xml.write(out)
                logging.info("XML File sucessfully generated")
                logging.debug(sql[:-1] + ";")
                cursor.execute(sql[:-1] + ";")
                logging.info("Data sucessfully inserted into database")
                #wait for 60 seconds
                time.sleep(5)
            except Exception as g:
                logging.error(g)
                logging.info("Database Connection will be reopened")
                db.close()
                break

```


3.6.4.3 Water

Das „water“ Service liest jede Minute alle in der „water.cfg“ angegebenen „Sensoren“ aus:

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
import pymysql
import sys
import glob
import datetime
import time
import logging
import logging.handlers
import socket
import RPi.GPIO as GPIO

#destination for the XML File
path = '/var/www/html/xml/water.xml'
#set up logging
logFormatter = logging.Formatter('%(asctime)s    -    %(levelname)s:
%(message)s')
rootLogger = logging.getLogger()
fileHandler = logging.handlers.RotatingFileHandler('/var/log/water/water.log',maxBytes
=1000000,backupCount=5)
fileHandler.setFormatter(logFormatter)
console = logging.StreamHandler()
rootLogger.addHandler(fileHandler)
rootLogger.addHandler(console)
rootLogger.setLevel(logging.INFO)

#Pin Modus setzen
GPIO.setmode(GPIO.BCM)

#Inputs setzen
pins = list()
with open('/home/pi/pinconfig/water.cfg', mode= 'r') as file:
    for line in file:
        pins = line.split(',')
        logging.info("GPIOs are: " + str(pins))
        for pin in pins:
            GPIO.setup(int(pin), GPIO.IN, pull_up_down =
GPIO.PUD_DOWN)

#find out local ip address (we want to tell the recievers the name of the
system)
```

```

local = (([ip for ip in socket.gethostbyname_ex(socket.gethostname())[2]
if not ip.startswith("127.")] or [[(s.connect(("8.8.8.8", 53)),
s.getsockname()[0], s.close()) for s in [socket.socket(socket.AF_INET,
socket.SOCK_DGRAM)]]][0][1])) + ["no IP found"])[0]

```

```

#header - figure out system name

```

```

sql = "select name from messsystem where ip = '%s';" % (local)

```

```

try:

```

```

    db = pymysql.connect(host='localhost', user='webuser',
password='password',db='serverraum_temperaturueberwachung',autocommit=True)

```

```

    cursor = db.cursor()

```

```

    logging.info("Connected to database")

```

```

    logging.debug(sql)

```

```

    cursor.execute(sql)

```

```

    systemname = cursor.fetchone()[0]

```

```

    logging.info("SystemName is: " +systemname)

```

```

    db.close()

```

```

except Exception as e:

```

```

    logging.error(e)

```

```

    sys.exit()

```

```

waterprefix = systemname.split(' ')[1]

```

```

while True:

```

```

    #connection stays open until an error

```

```

    try:

```

```

        db = pymysql.connect(host='localhost', user='webuser',
password='password',db='serverraum_temperaturueberwachung',autocommit=True)

```

```

        cursor = db.cursor()

```

```

        logging.info("Connected to database")

```

```

    except Exception as e:

```

```

        logging.error(e)

```

```

        continue

```

```

    while True:

```

```

        #connect to database

```

```

        try:

```

```

            cursor.execute("select sensorKennung, sensorID,
sensorPosition from sensor where fk_systemID = (select systemID from
messsystem where ip = '"+local+"') order by sensorName;")

```

```

            sensoren = cursor.fetchall()

```

```

            logging.debug(sensoren)

```

```

            # read 1-wire slaves list

```

```

        sql = "insert into messung (zeit , fk_sensorID, wasser)
values"

        out = '<?xml version="1.0" encoding="ISO-8859-1" ?><?xml-
stylesheet type="text/xsl" href="./actualwater.xsl"?><monitoring>'
        now = datetime.datetime.now().strftime("%Y-%m-%d
%H:%M:%S")

        logging.debug("beginning")
        #save measurements to the database and produce xml, missing
sensors will be created automatically
        for pin in pins:
            pin = int(pin)
            logging.debug("reading GPIO: " + str(pin))
            sk = 'wasser'+ str(waterprefix) + str(pin)
            try:
                logging.debug(str(GPIO.input(pin)))
                if str(GPIO.input(pin)) == "1":
                    water = 1
                    logging.warning("water signal on GPIO "
+ str(pin))

                else:
                    water = 0
                    logging.info("it's dry at GPIO " +
str(pin))

                sql = sql + " ('%s',func_getSID('%s'), %d)," %
(now, sk, water)

            for sensor in sensoren:
                if str(sensor[0]) == sk:
                    logging.debug("here")
                    out= out + '<sensor name =
'+str(sensor[2])+'><id>'+str(sensor[1])+'</id><level>'+str(water)+'</l
evel><zeit>'+now+'</zeit></sensor>'

                except Exception as e:
                    logging.error(e)
            out = out + "</monitoring>"
            logging.debug(out)
            with open(path, "w") as xml:
                print(out, file=xml)
            logging.info("XML File sucessfully generated")
            logging.debug(sql[:-1] + ";")
            cursor.execute(sql[:-1] + ";")
            logging.info("Data sucessfully inserted into database")
            #wait for 60 seconds
            time.sleep(60)
        except Exception as g:

```

```

logging.error(g)
logging.info("Database Connection will be reopened")
db.close()
break

```

3.6.4.4 Pullup

Sollten als Spannungsversorgung für Wassersensoren nicht genügend 3,3 V Pins zur Verfügung stehen, dann leistet das „pullup“ Service Abhilfe. Es schaltet die in der „pullup.cfg“ angegebenen GPIOs auf Output und erzeugt somit zusätzliche Pseudospannungsversorgungen:

```

#!/usr/bin/python
import time, subprocess
import logging
import logging.handlers
import RPi.GPIO as GPIO

#set up logging
logFormatter = logging.Formatter('%(asctime)s    -    %(levelname)s:
%(message)s')
rootLogger = logging.getLogger()
fileHandler = logging.handlers.RotatingFileHandler('/var/log/pullup/pullup.log',maxBytes=1000000,backupCount=5)
fileHandler.setFormatter(logFormatter)
console = logging.StreamHandler()
rootLogger.addHandler(fileHandler)
rootLogger.addHandler(console)
rootLogger.setLevel(logging.INFO)

#Pin Modus setzen
GPIO.setmode(GPIO.BCM)

#Inputs setzen
pins = list()
with open('/home/pi/pinconfig/pullup.cfg', mode= 'r') as file:
    for line in file:
        pins = line.split(',')
        logging.info("GPIOs are: " + str(pins))
        for pin in pins:
            GPIO.setup(int(pin), GPIO.OUT)
            GPIO.output(int(pin), GPIO.HIGH)
            logging.info("GPIO pulled up: " + pin )

```

3.6.4.5 smsd

Das „smsd“ Service ist zuständig für das Versenden von Warnungen bei Verlassen des Normbereichs und für das Senden der täglichen Statusmeldung:

```
# -*- coding: utf-8 -*-
import pymysql
import logging
import logging.handlers
import datetime
import os
import time
import sys
import socket

def save_to_messages_db(cur, nowf, type, title, sms):
    """
    Saves outgoing messages to the messages db
    :param cur: the cursor on serverraum_temperaturueberwachung
    :param nowf: the actual time formatted
    :param type: the type of the sms
    :param title: the title of the sms
    :param sms: the sms text
    """
    try:
        cur.execute('select ip from messsystem;')
        systeme = cur.fetchall()
        for system in systeme:
            logging.info("Saving sms in: " +system[0])
            mdb = pymysql.connect(host=system[0], user='webuser',
password='password',db='messages',autocommit=True)
            mcursor = mdb.cursor()
            sql = "insert into message values ('%s', %d, '%s', '%s')"
% (str(nowf),type,title, sms)
            logging.debug(sql)
            mcursor.execute(sql)
            mdb.close()
        except Exception as e:
            logging.error(e)

max_temp = 35.0;
max_hum = 70.0;
interval = 30
max_rauch = 10
silent = False
```

```

#set up logging
logFormatter = logging.Formatter('%(asctime)s    -    %(levelname)s:
%(message)s')
rootLogger = logging.getLogger()
fileHandler =
logging.handlers.RotatingFileHandler('/var/log/smsd/smsd.log',maxBytes=1
000000,backupCount=5)
fileHandler.setFormatter(logFormatter)
console = logging.StreamHandler()
rootLogger.addHandler(fileHandler)
rootLogger.addHandler(console)
rootLogger.setLevel(logging.INFO)

#reviewer add recievers in /etc/smsd/recievers.list
To = ""
try:
    with open("/etc/smsd/recievers.list", mode='r') as recievers:
        for reciever in recievers:
            logging.info("Adding reciever: " + reciever)
            To = To + "To: " + reciever + "\n"
except Exception as tel:
    logging.error(tel)
    sys.exit()
#general path
path = "/var/spool/sms"
#find out local ip address (we want to tell the recievers the name of the
system)
local = (([ip for ip in socket.gethostbyname_ex(socket.gethostname())[2]
if not ip.startswith("127.")] or [[(s.connect(("8.8.8.8", 53)),
s.getsockname()[0], s.close()) for s in [socket.socket(socket.AF_INET,
socket.SOCK_DGRAM)]]][0][1]]) + ["no IP found"])[0]
#header - figure out system name
sql = "select r.name from messsystem inner join raum r on fk_raumID =
r.raumID where ip = '%s';" % (local)
try:
    db = pymysql.connect(host='localhost', user='webuser',
password='password',db='serverraum_temperaturueberwachung',autocommit=Tr
ue)
    cursor = db.cursor()
    logging.info("Connected to database")
    logging.debug(sql)
    cursor.execute(sql)
    systemname = cursor.fetchone()[0]
    logging.info("SystemName is: " +systemname)

```

```

        db.close()
    except Exception as e:
        logging.error(e)
        sys.exit()
    up = To + "\nTemperaturüberwachung "+systemname+":\n"

while True:
    with open("/etc/smsd/sms.conf") as f:
        logging.info("Loaded configuration:")
        for line in f:
            key = line.split(':')[0]
            value = line.split(':')[1].strip()
            if key == "interval":
                interval = int(value)
                logging.info("interval: " + str(interval))
            if key == "silent":
                silent = bool(value)
                logging.info("muted: " + str(silent))
            if key == "max_temp":
                max_temp = float(value)
                logging.info("max_temp: " + str(max_temp))
            if key == "max_hum":
                max_hum = float(value)
                logging.info("max_hum: " + str(max_hum))
            if key == "max_smoke":
                max_rauch = value
                logging.info("max_smoke: " + str(max_rauch))

    try:
        while True:
            now = datetime.datetime.now()
            nowf = now.strftime("%Y-%m-%d %H:%M:%S")
            past = (now - datetime.timedelta(minutes=5)).strftime("%Y-%
            %m-%d %H:%M:%S")
            past30 = now - datetime.timedelta(minutes=interval)
            past18 = now - datetime.timedelta(hours=18)
            tm = now + datetime.timedelta(days=1)
            yes = (now - datetime.timedelta(days=1)).strftime("%Y-%m-
            %d %H:%M:%S")
            if silent and not
            datetime.datetime.fromtimestamp(os.stat("/home/pi/alive.txt").st_mtime)
            < past18:
                time.sleep(10)
                logging.info("muted")
                continue

```

```

        sms = up
        title = ""
        logging.info("try connecting to db")
        try:
            connection = pymysql.connect(host='localhost',
user='webuser',
password='passsword',db='serverraum_temperaturueberwachung',autocommit=True)

            except Exception as e:
                logging.error(e)
                continue
            cur = connection.cursor()
            if
datetime.datetime.fromtimestamp(os.stat("/home/pi/alive.txt").st_mtime)
< past18:
                cur.execute("select avg(temp) as temp , avg(feucht)
as feucht, avg(wasser) as wasser , avg(rauch) as rauch, sensorPosition
from view_24 where zeit > '"+yes+"' group by sensorPosition;")
                results = cur.fetchall()
                da = up + "Das System ist
online.\nDurschnittswerte:\n"
                with open("/var/spool/sms/outgoing/alive.txt",
mode='w') as f:
                    print(up + "Das System ist
online.\nDurschnittswerte:\n", file=f)
                    for result in results:
                        if result[0] == None and result[1] ==
None and result[2] == None:
                            print(result[4] + ": " +
str(result[3]) + "\n", file=f)
                            da = da + result[4] + ": " +
str(result[3]) + "\n"
                        elif result[0] == None and result[1] ==
None:
                            print(result[4] + ": " +
str(result[2]) + "\n", file=f)
                            da = da + result[4] + ": " +
str(result[2]) + "\n"
                        elif result[1] == None:
                            print(result[4] + ": " +
str(result[0]) + "°C\n", file=f)
                            da = da + result[4] + ": " +
str(result[0]) + "°C\n"
                        else:

```



```

        print(result[4] + ": " +
str(result[0]) + "°C / " + str(result[1]) + " %\n", file=f)
        da = da + result[4] + ": " +
str(result[0]) + "°C / " + str(result[1]) + " %\n"
        os.system("touch -d "+tm.strftime("%Y%m%d")+
/home/pi/alive.txt")
        save_to_messages_db(cur, nowf, 2, "alive", da)
        logging.info("daily SMS sent")
        cur.execute("delete from messung where zeit <
'"+yes+"'");

        connection.close()
        continue

        cur.execute("select avg(temp) as temp , avg(feucht) as
feucht, avg(wasser) as wasser , avg(rauch) as rauch, sensorName from web
where zeit > '"+past+"' group by sensorName;")
        results = cur.fetchall()
        msgs = dict()
        for result in results:
            try:
                if result[0] == None and result[1] == None and
result[2] == None:

                    if float(result[3]) > max_rauch:
                        msgs[result[4] + "r"] = "Die Werte
an Sensor " + result[4] + " sind außerhalb des Normalbereichs: " +
str(result[3]) + "\n"

                        continue
                    elif result[0] == None and result[1] == None:
                        if int(result[2]) > 0:
                            msgs[result[4] + "w"] = "Die Werte
an Sensor " + result[4] + " sind außerhalb des Normalbereichs: " +
str(result[2]) + "\n"

                            continue
                        elif result[1] == None:
                            if float(result[0]) > max_temp:
                                msgs[result[4] + "t"] = "Die Werte
an Sensor " + result[4] + " sind außerhalb des Normalbereichs: " +
str(result[0]) + "°C\n"

                                continue
                            else:
                                if float(result[0]) > max_temp and
float(result[1]) > max_hum:

                                    msgs[result[4] + "tf"] = "Die Werte
an Sensor " + result[4] + " sind außerhalb des Normalbereichs: " +
str(result[0]) + "°C/" + str(result[1]) + "%\n"

```

```

        continue
    except Exception as e:
        logging.error(e)
    files = os.listdir(path+"/sent/")
    for name, message in msgs.items():
        for file in files:
            for n in str(file)[-4].split("-"):
                if name == n and
(datetime.datetime.fromtimestamp(os.stat(path+"/sent/"+str(file)).st_mti
me) > past30):
                    message = ""
                    name = ""

                sms += message
                if message != "":
                    title += name + "-"
                logging.info("added " + message + "to SMS")
                if not title == "":
                    logging.info("title of sms is: "+ title)
                    save_to_messages_db(cur, nowf, 1, title[:-1], sms)
                    with open("/var/spool/sms/outgoing/"+title[:-
1]+".txt", mode='w') as f:
                        print(sms, file=f)
                    connection.close()
                    time.sleep(30)
except Exception as outer:
    logging.error(outer)
    connection.close()

```

Das gelb markierte *if* überprüft, ob die tägliche Benachrichtigung gesendet werden soll. Falls dem so ist, wird die SMS generiert, gesendet und in der „messages“ Datenbank abgespeichert.

Der richtige Zeitpunkt wird über ein File gefunden, welches den Namen „alive.txt“ trägt. Dieses wird mittels des folgenden Befehls täglich erstellt:

```
os.system("touch -d "+tm.strftime("%Y%m%d")+ " /home/pi/alive.txt")
```

Die Datei wird für 0 Uhr des folgenden Tages erstellt, wird es nun 18 Uhr, so beträgt die Zeitdifferenz zum Erstellen der Datei genau 18 Stunden und das *if* wird beim nächsten Schleifendurchgang betreten. Die grün markierte Code Sequenz bestimmt also den Zeitpunkt, zu dem die tägliche SMS gesendet wird.

3.6.4.6 failover

Das „failover“ Service sorgt dafür, dass Nachrichten, die zu lange zum Senden benötigen oder bei denen das Senden fehlgeschlagen ist, auf einem anderen verfügbaren System gesendet werden. Die Übertragung erfolgt per SCP, es ist also erforderlich, dass der *public key* des Root Users in der „authorized_keys“ Datei des pi Users am anderen System steht:

```
# -*- coding: utf-8 -*-
import logging
import logging.handlers
import datetime
import os
import time
import socket
import pymysql

#find out local ip address (we don't want to forward failed sms packets
to our own system)
local = ([ip for ip in socket.gethostbyname_ex(socket.gethostname())[2]
if not ip.startswith("127.")] or [[(s.connect(("8.8.8.8", 53)),
s.getsockname()[0], s.close()) for s in [socket.socket(socket.AF_INET,
socket.SOCK_DGRAM)]]][0][1]]) + ["no IP found"]][0]
#set up logging
logFormatter = logging.Formatter('%(asctime)s    -    %(levelname)s:
%(message)s')
rootLogger = logging.getLogger()
fileHandler = logging.handlers.RotatingFileHandler('/var/log/smsd/failover.log',maxByt
es=1000000,backupCount=5)
fileHandler.setFormatter(logFormatter)
console = logging.StreamHandler()
rootLogger.addHandler(fileHandler)
rootLogger.addHandler(console)
rootLogger.setLevel(logging.INFO)
#specify paths for failed sms
path = "/var/spool/sms/checked/"
failed = "/var/spool/sms/failed/"
def send(path, file, systeme):
    """
    Forwards the sms to other systems. Tries every available System until
it works
    :param path: dictionary
    :param file: filename
    :param systeme: the systems to try
    """
    for sys in systeme:
```

```

        try:
            if os.system("scp "+ path + str(file) + "
pi@"+sys[0]+":incoming") == 0:
                os.system("rm "+ path + str(file))
                logging.info("SMS forwarded to " + sys[1])
                break
        except Exception as s:
            logging.error(s)

while True:
    try:
        db = pymysql.connect(host='localhost', user='webuser',
password='password',db='serverraum_temperaturueberwachung',autocommit=True)

        cursor = db.cursor()
        logging.info("Connected to database")
        sql = "select ip, r.name from messsystem inner join raum r on
fk_raumID = r.raumID where ip != '%s';" % (local)
        cursor.execute(sql)
        systeme = cursor.fetchall()
        db.close()
    except Exception as e:
        logging.error(e)
        continue

    try:
        while True:
            now = datetime.datetime.now()
            past = (now - datetime.timedelta(seconds=225))
            files = os.listdir(path)
            for file in files:
                if
datetime.datetime.fromtimestamp(os.stat(path+str(file)).st_mtime) <
past:

                    send(path, file, systeme)
            files = os.listdir(failed)
            for file in files:
                os.system("sed -i \'2i\'" + failed + str(file))
                os.system("sed -i \'3~5d\' " + failed + str(file))
                send(failed, file, systeme)
            if len(os.listdir("/home/pi/incoming")) > 0:
                os.system("mv /home/pi/incoming/*
/var/spool/sms/outgoing")
                logging.info("moved files to Outgoing")
                time.sleep(10)

```

```
except Exception as outer:
    logging.error(outer)
```

3.6.4.7 respond

Das „respond“ Service stellt den interaktiven Teil des SMS Interfaces dar. Hier werden die in 3.5.1.3.2 beschriebenen Kommandos verarbeitet:

```
# -*- coding: utf-8 -*-
import pymysql
import logging
import logging.handlers
import datetime
import os
import time
import re
import sys
import socket

max_temp = 35.0;
max_hum = 70.0;
interval = 30
silent = False
max_rauch = 10

#set up logging
logFormatter = logging.Formatter('%(asctime)s - %(levelname)s:
%(message)s')
rootLogger = logging.getLogger()
fileHandler = logging.handlers.RotatingFileHandler('/var/log/smsd/smsd_respond.log',ma
xBytes=1000000,backupCount=5)
fileHandler.setFormatter(logFormatter)
console = logging.StreamHandler()
rootLogger.addHandler(fileHandler)
rootLogger.addHandler(console)
rootLogger.setLevel(logging.INFO)

#reviewer add recievers in /etc/smsd/recievers.list
To = ""
try:
    with open("/etc/smsd/recievers.list", mode='r') as recievers:
        for reciever in recievers:
            logging.info("Adding reciever: " + reciever)
            To = To + "To: " + reciever + "\n"
```

```

except Exception as tel:
    logging.error(tel)
    sys.exit()
#general path
path = "/var/spool/sms"
#find out local ip address (we want to tell the recievers the name of the
system)
local = (([ip for ip in socket.gethostbyname_ex(socket.gethostname())[2]
if not ip.startswith("127.")] or [[(s.connect(("8.8.8.8", 53)),
s.getsockname()[0], s.close()) for s in [socket.socket(socket.AF_INET,
socket.SOCK_DGRAM)]]][0][1]]) + ["no IP found"])[0]
#header - figure out system name
sql = "select r.name from messsystem inner join raum r on fk_raumID =
r.raumID where ip = '%s';" % (local)
try:
    db = pymysql.connect(host='localhost', user='webuser',
password='password',db='serverraum_temperaturueberwachung',autocommit=Tr
ue)
    cursor = db.cursor()
    logging.info("Connected to database")
    logging.debug(sql)
    cursor.execute(sql)
    systemname = cursor.fetchone()[0]
    logging.info("SystemName is: " +systemname)
    db.close()
except Exception as e:
    logging.error(e)
    sys.exit()
up = To + "\nTemperaturüberwachung "+systemname+":\n"
#text of the help message
help = up + "Available commands: \nHELP - Displays all commands\nSTATUS
- Displays actual status\nMUTE and UNMUTE\nINTERVAL: mins - to change the
interval\n\
Same Usage:\nMAXTEMP: degrees\nMAXHUM: percent\nMAXSMOKE: int [0;
100]\nnot case sensitive!"

while True:
    with open("/etc/smsd/sms.conf") as f:
        logging.info("reading configuration from sms.conf")
        for line in f:
            key = line.split(':')[0]
            value = line.split(':')[1].strip()
            if key == "interval":
                interval = int(value)

```

```

        if key == "silent":
            silent = bool(value)
        if key == "max_temp":
            max_temp = value
        if key == "max_hum":
            max_hum = value
        if key == "max_smoke":
            max_rauch = value

    try:
        while True:
            now = datetime.datetime.now()
            nowf = now.strftime("%Y-%m-%d %H:%M:%S")
            rec = (now - datetime.timedelta(minutes=15)).strftime("%Y-%m-%d %H:%M:%S")
            files = os.listdir(path+"/incoming/")
            changed= False
            try:
                connection = pymysql.connect(host='localhost',
user='webuser',
password='password',db='serverraum_temperaturueberwachung',autocommit=True)

                cur = connection.cursor()
                cur.execute('select ip from messsystem;')
                systeme = cur.fetchall()
            except Exception as e:
                logging.error(e)
            for file in files:
                logging.info("new sms recieved " + file)
                with open(path + "/incoming/" +file) as f:
                    for line in f:
                        line = str(line)
                        if re.match(r'status', line,
re.IGNORECASE):
                            logging.info("try connecting to
db")

                            try:
                                #get average temp over the
last 15 minutes

                                cur.execute("select
avg(temp) as temp , avg(feucht) as feucht, avg(wasser) as wasser ,
avg(rauch) as rauch, sensorPosition from view_24 where zeit > '"+rec+"'
group by sensorPosition;")

                                results = cur.fetchall()

```

```

sms = up + "Das System ist
online.\nDurschnittswerte:\n"

        with
open("/var/spool/sms/outgoing/status.txt", mode='w') as f:
        print(up + "Das System
ist online.\nDurschnittswerte:\n", file=f)

        for result in results:
            if result[0] ==
None and result[1] == None and result[2] == None:

                print(result[4] + ": " + str(result[3]) + "\n", file=f)
sms = sms +
result[4] + ": " + str(result[3]) + "\n"
                elif result[0]
== None and result[1] == None:

                print(result[4] + ": " + str(result[2]) + "\n", file=f)
sms = sms +
result[4] + ": " + str(result[2]) + "\n"
                elif result[1]
== None:

                print(result[4] + ": " + str(result[0]) + "°C\n", file=f)
sms = sms +
result[4] + ": " + str(result[0]) + "°C\n"
                else:

                print(result[4] + ": " + str(result[0]) + "°C / " + str(result[1]) + "
%\n", file=f)
sms = sms +
result[4] + ": " + str(result[0]) + "°C / " + str(result[1]) + " %\n"
                #writing outgoing message to
messages db

                for system in systeme:
                    logging.info("Saving
sms in: " +system[0])

                    mdb
                    =
pymysql.connect(host=system[0],
                    user='webuser',
password='password',db='messages',autocommit=True)
                    mcursor = mdb.cursor()
                    #sms is type 3 ==
requested info

                    sql = "insert into
message values ('%s', %d, '%s', '%s') " % \

```



```

                                                    (str(nowf),3,"status",
sms)
                                                    mcursor.execute(sql)
                                                    mdb.close()
                                                    logging.info("status      SMS
sent as requested")
                                                    connection.close()
except Exception as e:
    logging.error(e)
    continue
elif    re.match(    r'mute',    line,
re.IGNORECASE):
    logging.info("mute was requested")
    silent = True
    changed = True
    continue
elif    re.match(    r'unmute',    line,
re.IGNORECASE):
    logging.info("unmute      was
requested")
    silent = False
    changed = True
    continue
elif    re.match(    r'interval:',    line,
re.IGNORECASE):
    interval
    =
int(line.split(':')[1].strip())
    logging.info("change of interval
to "+str(interval)+" was requested")
    changed = True
    continue
elif    re.match(    r'maxtemp:',    line,
re.IGNORECASE):
    max_temp
    =
float(line.split(':')[1].strip())
    logging.info("changing max temp to
"+str(max_temp)+" was requested")
    changed = True
    continue
elif    re.match(    r'maxhum:',    line,
re.IGNORECASE):
    max_hum
    =
float(line.split(':')[1].strip())

```

```

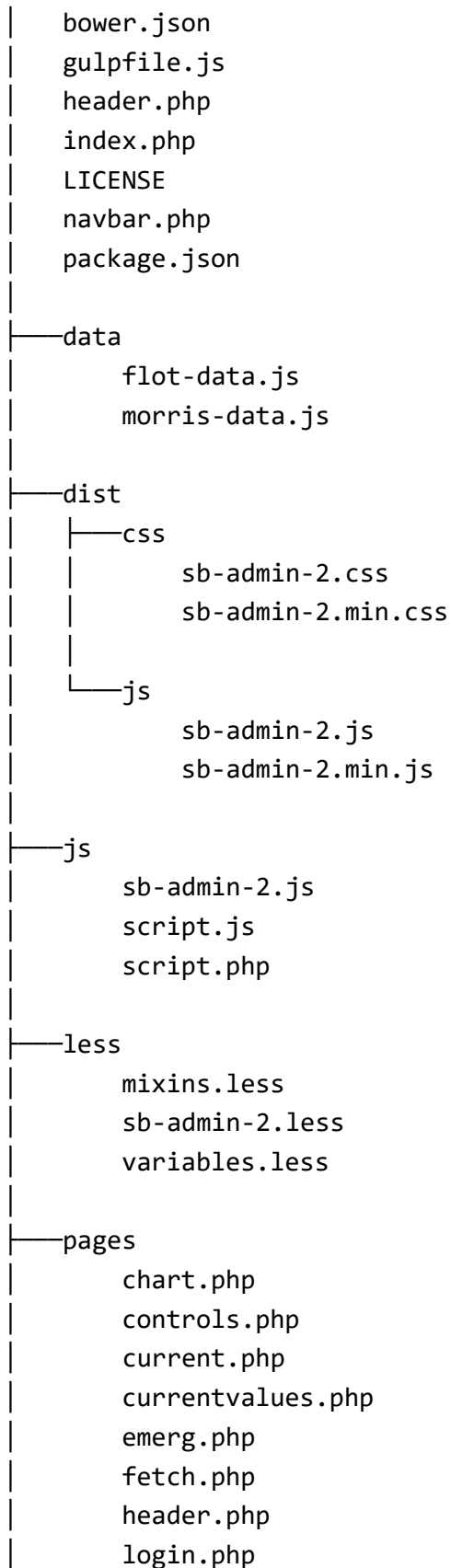
                                logging.info("changing max hum to
"+str(max_hum)+" was requested")
                                changed = True
                                continue
                                elif re.match( r'maxsmoke:', line,
re.IGNORECASE):
                                max_rauch =
float(line.split(':')[1].strip())
                                logging.info("changing max smoke
to "+str(max_rauch)+" was requested")
                                changed = True
                                continue
                                elif re.match( r'help', line,
re.IGNORECASE):
                                logging.info("help was requested")
                                logging.debug(help)
                                with
open("/var/spool/sms/outgoing/help.txt", mode="w") as f:
                                print(help, file=f)
                                continue
                                os.system("rm " + path + "/incoming/" + file)
                                if changed:
                                    if not silent:
                                        silent = ""
                                    with open("/etc/smsd/sms.conf", mode="w") as f:
                                        print("interval: " + str(interval), file=f)
                                        print("silent: " + str(silent), file=f)
                                        print("max_temp: " + str(max_temp) , file=f)
                                        print("max_hum: " + str(max_hum) , file=f)
                                        print("max_smoke: " + str(max_rauch) , file=f)
                                    logging.info("New Configuration was saved")
                                    os.system("systemctl restart smsd")
                                    logging.info("smsd has successfully loaded the new
configuration")
                                    #wait for 15 seconds
                                    time.sleep(15)
                                except Exception as outer:
                                    logging.error(outer)

```

Bei einer Konfigurationsänderung wird das „smsd“ Service neu gestartet. Dies wird durch den gelb markierten Befehl erreicht.

3.6.5 Webinterface

Das Webinterface befindet sich im Verzeichnis `/var/www/html/` und hat folgende Struktur:



```

|
|   map.php
|   messages.php
|   navbar.php
|   notifications.php
|   schedule.php
|
|---vendor
|   |---bootstrap
|   |   |---css
|   |   |   bootstrap.css
|   |   |   bootstrap.min.css
|   |   |---fonts
|   |   |   glyphsicons-halflings-regular.eot
|   |   |   glyphsicons-halflings-regular.svg
|   |   |   glyphsicons-halflings-regular.ttf
|   |   |   glyphsicons-halflings-regular.woff
|   |   |   glyphsicons-halflings-regular.woff2
|   |   |---js
|   |   |   bootstrap.js
|   |   |   bootstrap.min.js
|   |---bootstrap-social
|   |   bootstrap-social.css
|   |   bootstrap-social.less
|   |   bootstrap-social.scss
|   |---datatables
|   |   |---css
|   |   |   dataTables.bootstrap.css
|   |   |   dataTables.bootstrap.min.css
|   |   |   dataTables.bootstrap4.css
|   |   |   dataTables.bootstrap4.min.css
|   |   |   dataTables.foundation.css
|   |   |   dataTables.foundation.min.css
|   |   |   dataTables.jqueryui.css
|   |   |   dataTables.jqueryui.min.css
|   |   |   dataTables.material.css
|   |   |   dataTables.material.min.css
|   |   |   dataTables.semanticui.css
|   |   |   dataTables.semanticui.min.css
|   |   |   dataTables.uikit.css
|   |   |   dataTables.uikit.min.css

```

```
|
|
|
|      jquery.dataTables.css
|      jquery.dataTables.min.css
|      jquery.dataTables_themeroller.css
|
|      └──images
|          favicon.ico
|          Sorting icons.psd
|          sort_asc.png
|          sort_asc_disabled.png
|          sort_both.png
|          sort_desc.png
|          sort_desc_disabled.png
|
|      └──js
|          DataTables.bootstrap.js
|          DataTables.bootstrap.min.js
|          DataTables.bootstrap4.js
|          DataTables.bootstrap4.min.js
|          DataTables.foundation.js
|          DataTables.foundation.min.js
|          DataTables.jqueryui.js
|          DataTables.jqueryui.min.js
|          DataTables.material.js
|          DataTables.material.min.js
|          DataTables.semanticui.js
|          DataTables.semanticui.min.js
|          DataTables.uikit.js
|          DataTables.uikit.min.js
|          jquery.dataTables.js
|          jquery.dataTables.min.js
|          jquery.js
|
|      └──datatables-plugins
|          DataTables.bootstrap.css
|          DataTables.bootstrap.js
|          DataTables.bootstrap.min.js
|          index.html
|
|      └──datatables-responsive
|          DataTables.responsive.css
|          DataTables.responsive.js
|          DataTables.responsive.scss
|
|      └──flot
```

```

|
|
|   excanvas.js
|   excanvas.min.js
|   jquery.colorhelpers.js
|   jquery.flot.canvas.js
|   jquery.flot.categories.js
|   jquery.flot.crosshair.js
|   jquery.flot.errorbars.js
|   jquery.flot.fillbetween.js
|   jquery.flot.image.js
|   jquery.flot.js
|   jquery.flot.navigate.js
|   jquery.flot.pie.js
|   jquery.flot.resize.js
|   jquery.flot.selection.js
|   jquery.flot.stack.js
|   jquery.flot.symbol.js
|   jquery.flot.threshold.js
|   jquery.flot.time.js
|   jquery.js
|
|   └── flot-tooltip
|       |   jquery.flot.tooltip.js
|       |   jquery.flot.tooltip.min.js
|       |   jquery.flot.tooltip.source.js
|
|   └── font-awesome
|       |   HELP-US-OUT.txt
|       |
|       |   └── css
|       |       |   font-awesome.css
|       |       |   font-awesome.css.map
|       |       |   font-awesome.min.css
|       |
|       |   └── fonts
|       |       |   fontawesome-webfont.eot
|       |       |   fontawesome-webfont.svg
|       |       |   fontawesome-webfont.ttf
|       |       |   fontawesome-webfont.woff
|       |       |   fontawesome-webfont.woff2
|       |       |   FontAwesome.otf
|       |
|       |   └── less
|       |       |   animated.less
|       |       |   bordered-pulled.less

```

```
|
|
|
|      core.less
|      extras.less
|      fixed-width.less
|      font-awesome.less
|      icons.less
|      larger.less
|      list.less
|      mixins.less
|      path.less
|      rotated-flipped.less
|      screen-reader.less
|      spinning.less
|      stacked.less
|      variables.less
|
|      └── scss
|          font-awesome.scss
|          _animated.scss
|          _bordered-pulled.scss
|          _core.scss
|          _extras.scss
|          _fixed-width.scss
|          _icons.scss
|          _larger.scss
|          _list.scss
|          _mixins.scss
|          _path.scss
|          _rotated-flipped.scss
|          _screen-reader.scss
|          _spinning.scss
|          _stacked.scss
|          _variables.scss
|
|      └── jquery
|          jquery.js
|          jquery.min.js
|
|      └── metisMenu
|          metisMenu.css
|          metisMenu.js
|          metisMenu.min.css
|          metisMenu.min.js
|
|      └── morrisjs
```

```

|
|
|      morris.css
|      morris.js
|      morris.min.js
|
|      └─ raphael
|          raphael.js
|          raphael.min.js
|
|      └─ xml
|          actualfeucht.xml
|          actualsmoke.xml
|          actualtemp.xml
|          actualwater.xml
|          feucht.xml
|          smoke.xml
|          temp.xml
|          water.xml

```

Die Hauptseite befindet sich im Überverzeichnis, die Unterseiten in „pages“. Die Unterseiten und Hauptseiten haben unterschiedliche Dateien zur Erstellung der *navbar* und des *headers*.

3.6.5.1 Das Verzeichnis „/var/www/html“

3.6.5.1.1 header.php

- Setzen der head & header Attribute
- Laden der Bibliotheken
- Aufruf von navbar.php

3.6.5.1.2 navbar.php

- Erzeugen der Navigation für die „index“ Seite

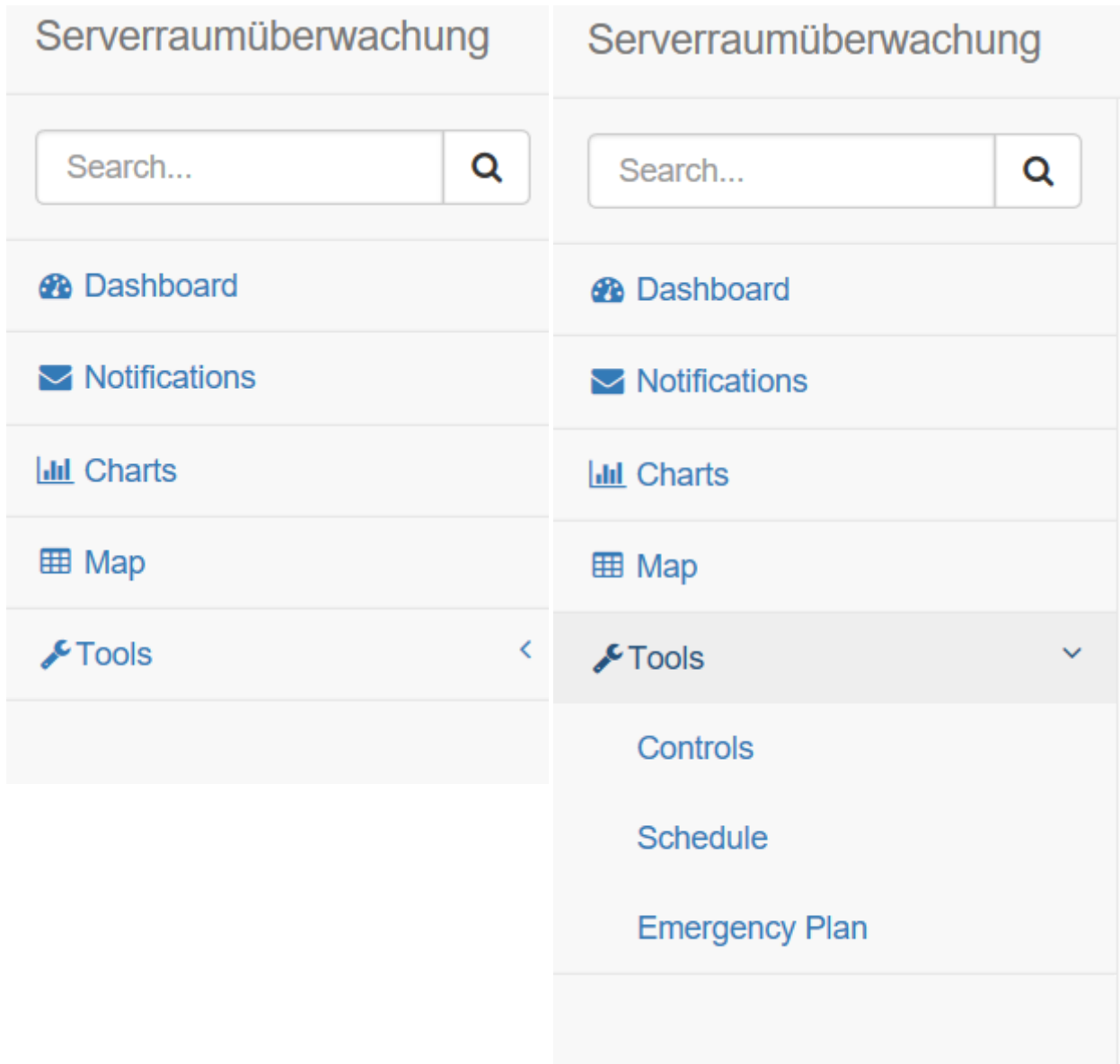


Abbildung 3-32
„Navbar“ eingeklappt und ausgeklappt

3.6.5.1.3 index.php

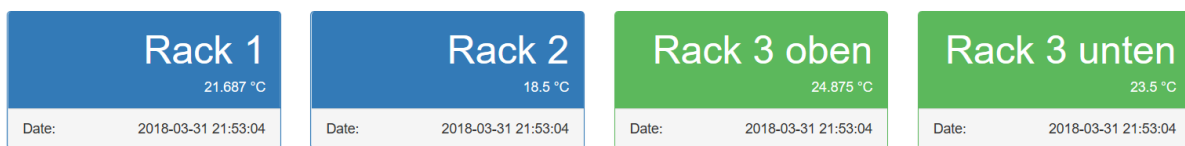
- Aufrufen von header.php
- Aufrufen von pages/current.php
- Aufrufen von pages/messages.php
- Laden der js/script.php
- Initialisieren der Tooltips

3.6.5.2 Das Verzeichnis „/var/www/html/pages“

3.6.5.2.1 current.php

- Aufrufen von „currentvalues.php“ aller in der Datenbank eingetragener Messsysteme und Darstellen dieser in n *div* Boxen
- Sammeln der Sensoranzeigen aller Systeme (Iterieren über alle Systeme und Generieren eines dynamischen Javascripts, welches die entsprechenden „currentvalues.php“ Seiten in die dafür vorgesehenen Container lädt.

Serverraum Alt



Serverraum 078



Abbildung 3-33
Screenshot Beispielausgabe der „current.php“

3.6.5.2.2 currentvalues.php

- XSL Transformation aller XML Files mit den zugehörigen XSL Stylesheets.
- Generieren der Sensoranzeigen des eines Systems

Serverraum Alt

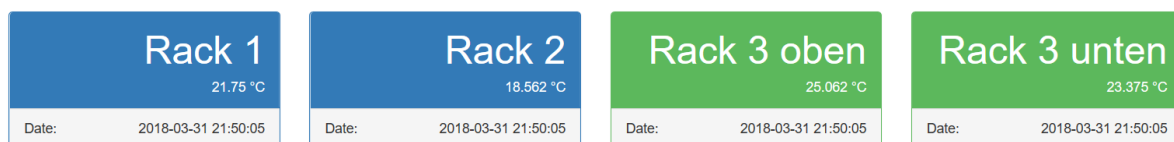


Abbildung 3-34
Screenshot Beispielausgabe „currentvalues.php“

3.6.5.2.3 messages.php

- Generieren des *Notification* Bereichs aus der „messages“ Datenbank (maximal 10 Nachrichten werden angezeigt) (siehe 3.5.4.1).

3.6.5.2.4 notifications.php

- Anzeigen von allen Nachrichten geordnet nach nicht kritisch und kritisch

Notifications

Critical	Non Critical																		
<div><div>⚡ R1L-R2t-R3ut-R3o</div><div>2018-04-02 01:55:24</div></div>	<div><div>✉ aliv</div><div>2018-04-02 01:59:49</div></div> <tr><td><div><div>✉ aliv</div><div>2018-04-02 01:59:37</div></div></td><td></td></tr> <tr><td></td><td><div><div>💬 status</div><div>2018-04-02 01:43:09</div></div></td></tr> <tr><td></td><td><div><div>💬 status</div><div>2018-04-02 01:37:46</div></div></td></tr> <tr><td></td><td><div><div>💬 status</div><div>2018-04-02 00:56:50</div></div></td></tr> <tr><td></td><td><div><div>💬 status</div><div>2018-04-02 00:43:15</div></div></td></tr> <tr><td></td><td><div><div>💬 status</div><div>2018-04-02 00:37:54</div></div></td></tr> <tr><td></td><td><div><div>💬 status</div><div>2018-04-02 00:34:25</div></div></td></tr> <tr><td></td><td><div><div>💬 status</div><div>2018-04-02 00:30:57</div></div></td></tr> <tr><td></td><td><div><div>💬 status</div><div>2018-04-02 00:23:29</div></div></td></tr>	<div><div>✉ aliv</div><div>2018-04-02 01:59:37</div></div>			<div><div>💬 status</div><div>2018-04-02 01:43:09</div></div>		<div><div>💬 status</div><div>2018-04-02 01:37:46</div></div>		<div><div>💬 status</div><div>2018-04-02 00:56:50</div></div>		<div><div>💬 status</div><div>2018-04-02 00:43:15</div></div>		<div><div>💬 status</div><div>2018-04-02 00:37:54</div></div>		<div><div>💬 status</div><div>2018-04-02 00:34:25</div></div>		<div><div>💬 status</div><div>2018-04-02 00:30:57</div></div>		<div><div>💬 status</div><div>2018-04-02 00:23:29</div></div>
<div><div>✉ aliv</div><div>2018-04-02 01:59:37</div></div>																			
	<div><div>💬 status</div><div>2018-04-02 01:43:09</div></div>																		
	<div><div>💬 status</div><div>2018-04-02 01:37:46</div></div>																		
	<div><div>💬 status</div><div>2018-04-02 00:56:50</div></div>																		
	<div><div>💬 status</div><div>2018-04-02 00:43:15</div></div>																		
	<div><div>💬 status</div><div>2018-04-02 00:37:54</div></div>																		
	<div><div>💬 status</div><div>2018-04-02 00:34:25</div></div>																		
	<div><div>💬 status</div><div>2018-04-02 00:30:57</div></div>																		
	<div><div>💬 status</div><div>2018-04-02 00:23:29</div></div>																		

Abbildung 3-35
Screenshot des Notifications Tab

3.6.5.3 /var/www/html/js/skript.php

- Automatisches Nachladen von Werten alle 5 Sekunden, durch Aufruf von „current.php“ aller Systeme.
- Generieren des „Flot Charts“ aus den in der Seite eingebetteten Sensorwerten. Dafür wird das „data-*“ Attribut verwendet.

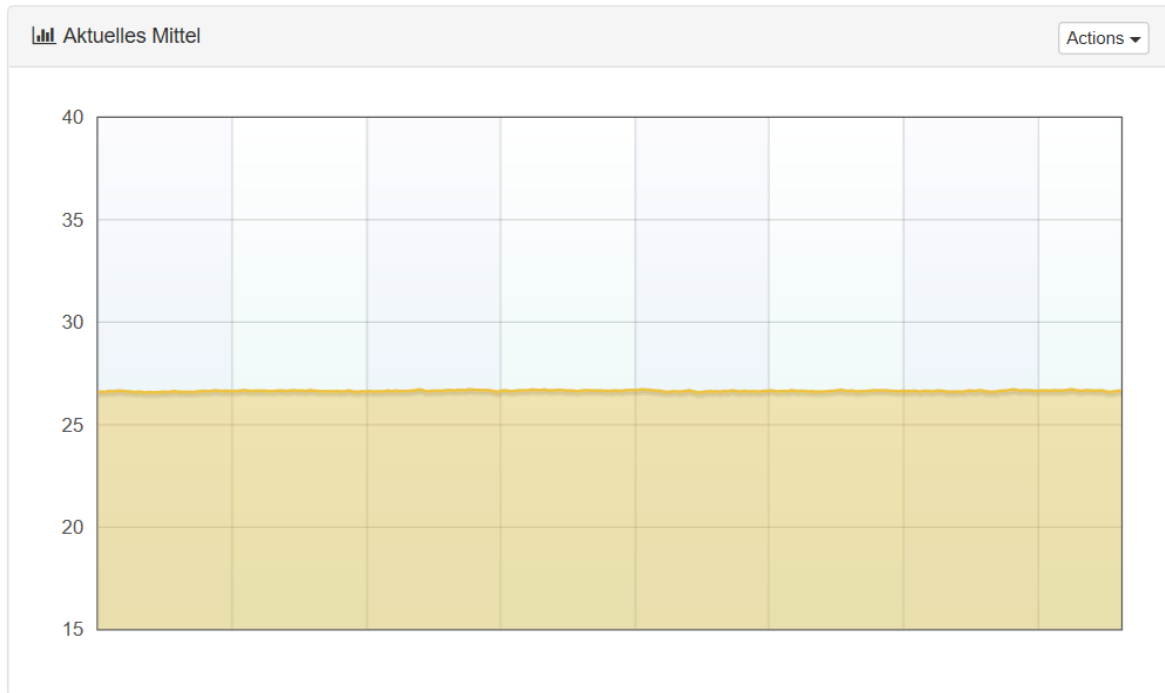


Abbildung 3-36
Screenshot des „Flot Charts“

3.6.5.4 Das Verzeichnis „/var/www/xml/“

3.6.5.4.1 actual*.xml

- Stylesheets für die XML Dateien, die zur Umwandlung in HTML Code dienen.

Rack 10 mitte
28.5 °C
Date: 2018-03-31 22:04:14
Rack 11 oben
29.437 °C
Date: 2018-03-31 22:04:14
Rack 12 mitte
28.25 °C
Date: 2018-03-31 22:04:14
Rack 3 oben
25.625 °C
Date: 2018-03-31 22:04:14
Rack 4 oben
26.312 °C
Date: 2018-03-31 22:04:14
Rack 8 unten
32.187 °C
Date: 2018-03-31 22:04:14
Rack 9 oben
29.875 °C
Date: 2018-03-31 22:04:14
Rack 9 unten
28.75 °C
Date: 2018-03-31 22:04:14

Abbildung 3-37
Ausgabe der Seite ohne Bootstrap

3.6.5.4.2 temp.xml

- Hier sind die aktuellen Sensorwerte (generiert von „1wire“ (siehe 3.6.4.1)) gespeichert.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<?xml-stylesheet type="text/xsl" href="./actualtemp.xsl"?>
<monitoring>
  <sensor name = "Rack 9 oben">
    <id>1</id>
    <temp>29.875</temp>
    <zeit>2018-03-31 22:07:14</zeit>
  </sensor>
  <sensor name = "Rack 9 unten">
    <id>2</id>
    <temp>28.75</temp>
    <zeit>2018-03-31 22:07:14</zeit>
  </sensor>
  <sensor name = "Rack 12 mitte">
    <id>3</id>
    <temp>28.25</temp>
    <zeit>2018-03-31 22:07:14</zeit>
  </sensor>
  <sensor name = "Rack 10 mitte">
    <id>4</id>
    <temp>28.5</temp>
    <zeit>2018-03-31 22:07:14</zeit>
  </sensor>
  .
  .
  .
  .
</monitoring>
```

Beispiel für das Aussehen einer „temp.xml“. Die Punkte symbolisieren, dass der Output abgekürzt wurde. Die anderen XML Dateien sind nach selbigen Schema aufgebaut, bei den Wassersensoren heißt der Sensorwert jedoch „level“ und bei den DHT Sensoren gibt es die Werte „temp“ und „feucht“.

3.7 Ausblick und Konfiguration

3.7.1 Wartung der SIM Karte

Die PIN der SIM Karte und auch die Modemschnittstelle, kann in der „/etc/smd.conf“ geändert werden. Außerdem können hier weitere USB Modems hinzugefügt werden (nicht ratsam wegen möglicher Schnittstellenkonflikte).

Die SIM Karte kann auf <https://www.yesss.at/kontomanager.at/aufladen/bezahl-art.php> aufgeladen werden. Die Login Daten sind:

Benutzername: **068120654859**
Passwort: **password**
PIN: **1234**

Benutzername: **068110616440**
Passwort: **password**
PIN: **1234**

3.7.2 Konfiguration des „smsd“ Services

Intervall, Schwellwerte und der lautlose Modus können in der „/etc/smsd/sms.conf“, sowie per SMS an das System, geändert / eingeschalten werden.

```
interval: 30
silent:
max_temp: 35.0
max_hum: 80.0
max_smoke: 10
```

Da der Schwellwert des Wasserstandes immer null sein muss, da nur null oder eins möglich ist, ist er nicht im Konfigurationsfile veränderbar; er ist direkt bei der Überprüfung des entsprechenden Wertes im „smsd“ Service eingetragen.

Weitere Telefonnummern können in der „/etc/smsd/recievers.list“ eingetragen werden. Einzelne Telefonnummern müssen durch ein *newline* getrennt werden.

```
43664xxxxxxx
43660xxxxxxx
```

3.7.3 Erweiterung des Messsystems

Zum Hinzufügen von neuen 1-Wire Sensoren muss der Sensor nur richtig – im aktiv Modus ohne zusätzliche Spannungsversorgung – montiert werden, dann wird ein neuer

Eintrag für den Sensor hinzugefügt. Die Aufzeichnung der Messwerte startet sofort, allerdings muss der Sensor für die Anzeige im Webinterface einem Messsystem zugeordnet werden. Benennung und Position sind optional, jedoch sehr empfehlenswert.

Beim Anschluss von neuen DHT11 / DHT22 / MQ-2 Sensoren muss in der entsprechenden Konfigurationsdatei in „/home/pi/pinconfig/“ der GPIO Pin in die Liste aufgenommen werden. Die Pins sind mit Beistrichen getrennt und ohne Abstände anzugeben.

Wird ein neuer Sensortyp angeschlossen, wird dafür ein neues Service zum Auslesen, ein neues XSL Stylesheet und eine neue Spalte in der Tabelle „messung“ benötigt. Außerdem müssen die Schwellwerte in der „/etc/smsd/sms.conf“ eingetragen werden und im „respond“ Service eine neue Option zum Ändern per SMS, sowie eine Überprüfung derselbigen Werte im „smsd“ Service geschaffen werden. Weiters muss das Einlesen der „/etc/smsd/sms.conf“ im „smsd“ und „respond“ Service erweitert werden.

3.7.4 Einrichten von Notabschaltungen

Zum Einrichten einer Notabschaltung können die in 3.5.1.2 vorgestellten Konzepte verwendet werden.

Wird die Option mit dem Skript und herkömmlichen Relais gewählt, ist es notwendig, einen richtig dimensionierten MOSFET auf der Masseseite zur Steuerung zwischenschalten, oder ein Relais zu wählen, welches mit den geringen Schaltströmen des Raspberry Pis arbeiten kann. Dann kann durch das einfache Schalten eines GPIO Pins das Relais geschaltet werden. Dazu ist zu beachten, dass der Zustand eines GPIOs nicht gespeichert werden kann, also muss dieser separat in einer Datei abgelegt werden.

GPIOs können am Raspberry Pi entweder in Python direkt geschaltet werden oder durch die Modifikation von Systemdateien, von der Linux Shell aus. Als Beispiel für das Schalten eines GPIOs mit Python kann das Skript „pullup.py“ betrachtet werden.

Beispiel Einschalten eines GPIOs (GPIO Pin 17 / physischer PIN 11) per Shell als root:³

```
echo "17" > /sys/class/gpio/export
echo "out" > /sys/class/gpio/gpio17/direction
echo "1" > /sys/class/gpio/gpio17/value
#Rückgängig machen
echo "17" > /sys/class/gpio/unexport
```

Ausschalten:

```
echo "0" > /sys/class/gpio/gpio17/value
```

³ <http://raspberrypiguide.de/howtos/raspberry-pi-gpio-how-to/>

4 Inventur und Strommessungen

4.1 Kurzbeschreibung

Um einen möglichst genauen ökologischen Fußabdruck ermitteln zu können, ist erforderlich zu wissen, welche Faktoren diesen beeinflussen. Die Basis für die Untersuchungen bildet eine Inventarliste, welche eine Übersicht der Geräte der IT-Infrastruktur der Schule darstellt. Um Aussagen bezüglich des Stromverbrauches der IT-Infrastruktur treffen zu können, werden unterschiedliche Messmethoden entwickelt.

4.2 Situation vor dem Projekt

Zu Projektbeginn waren die Inventarlisten der Schule vorhanden, die lediglich Kurzbezeichnungen der Geräte, Aufstellort und Inventarnummer enthalten. Da in diesen Aufzeichnungen keine genauen Modell- oder Typen-Bezeichnungen vorhanden sind, sind diese Angaben für die Entwicklung des Messplans nicht ausreichend. Der Anteil der hohen Energiekosten, der durch die IT-Infrastruktur verursacht wird, ist nicht bekannt.

4.3 Inventarlisten - Mengengerüst

Die ermittelte Inventarliste umfasst sowohl all jene Geräte, die bereits in Listen erfasst waren, als auch all jene Geräte, die durch Überprüfung des Inventars all jener Räume, zu denen wir Zugang hatten, zusätzlich ermittelt werden konnten.

In der Inventarliste wurden alle Geräte der IT-Infrastruktur erfasst, zu welchen das Team Zugang bekommen hatte oder eine Liste verfügbar war.

Die Informationen Gerätetyp (PC, Monitor, Switch, ...) und Modell (Nummer oder Bezeichnung) wurden in die Inventarliste aufgenommen. Diese Inventarliste bildete die Grundlage für ein Mengengerüst, in welchem die Information über Aufstellungsort und Anzahl der Geräte erfasst wurde. Das Mengengerüst wurde in Form einer Exceltabelle umgesetzt, in welcher nach Gerätetyp, Hersteller und Modell gefiltert werden kann. Das Mengengerüst ist auf der beiliegenden CD zu finden.

4.4 Gruppierung der Geräte

Die Geräte wurden entsprechend ihres Modelltyps gruppiert.

Ursprünglich war geplant eine Gruppierung nach in den Geräten verbauten Komponenten (Angabe aus Datenblätter / Herstellerangaben) vorzunehmen, jedoch stellte sich heraus, dass bei einigen PCs die Komponenten zwischenzeitlich von der Schule ausgetauscht wurden. Daher stimmten diese nicht mehr mit den Datenblättern überein.

Um eine exakte Information über die tatsächlich verbauten Komponenten erhalten zu können, musste jeder PC geöffnet und die darin enthaltenen Komponenten überprüft beziehungsweise in der Inventarliste ergänzt werden.

Aus zeitökonomischen Gründen wurden daher nur Geräte mit der gleichen Modellbezeichnung und Aufstellungsort zusammengefasst und es erfolgte eine Einschränkung auf aktive Komponenten und die Übungskomponenten aus Raum 078, weil bei diesen ein größeres Verbesserungspotenzial vermutet wurde.

4.4.1 PCs

Tabelle 4-1
Gruppen der PCs

Hersteller	Modell	Gruppe	Anzahl	Notiz
Acer	VERITON X680	PC-A1	39	
Acer	VERITON M680G	PC-A2	36	
dimotion	Mini N5	PC-D1	49	
Fujitsu Siemens	ESPRIMO P5730 E-Star4	PC-F1	24	
HP	compaq d530 cmt	PC-H1	2	
HP	EliteDesk 800 G1 TWR	PC-H2	20	
HP	EliteDesk 800 G2 TWR	PC-H3	6	
HP	ProDesk 400 G1 MT	PC-H4	38	
HP	ProDesk 400 G2 MT	PC-H5	68	
HP	ProDesk 400 G3 MT B	PC-H6	2	
HP	ProDesk 400 G4 MT Business PC	PC-H7	36	

4.4.2 Netzwerkkomponenten

Tabelle 4-2
Gruppen der Netzwerkkomponenten

Gerätetyp	Hersteller	Modell	Gruppe	Anzahl	Notiz
AP	Fortinet	AP832i	AP1	8	
AP	Fortinet	WLC 200D	AP2	1	
AP	Cisco	AIR-CT5508-K9	AP3	1	
AP	Cisco	AIR-LAP1242AG-E-K9	AP4	32	
AP	Cisco	AIR-LAP1262N-E-K9	AP5	13	
CCC	Cisco	WS-C6504	CCC1	2	
Konsolen-Server	perle	IOLAN STS	CON1	4	
Firewall	ciena	cn 3920	FW1	1	
Firewall	Fortinet	FortiGate300D	FW2	1	
Firewall	Cisco	ASA 5525-X	FW3	1	
Firewall	Cisco	ASA 5510	FW4	6	Übungsrack
Router	Cisco	2621 xm	R1	4	Übungsrack
Router	Cisco	2800	R2	7	Übungsrack
Router	Cisco	2811	R3	17	Übungsrack
Router	Cisco	2901	R4	9	Übungsrack
Router	Cisco	2911	R5	6	Übungsrack
Switch	Cisco	WS-C2948	SW1	2	
Switch	Cisco	WS-C2950	SW2	3	
Switch	Cisco	WS-C2960	SW3	27	teils Übungsrack
Switch	Cisco	WS-C3550	SW4	3	teils Übungsrack
Switch	Cisco	WS-C3560	SW5	10	teils Übungsrack
Switch	Cisco	WS-C3750	SW6	6	
Switch	Cisco	WS-C4506	SW7	3	

4.4.3 Sonstiges

Tabelle 4-3
Gruppen andere Gerätetypen

Gerätetyp	Hersteller	Modell	Gruppe	Anzahl	Notiz
Server	HP	ProLiant DL385G7	SE1	8	Übungs-Serverrack 1
Server	HP	ProLiant DL360p Gen8	SE2	8	
Storage	HP	StorageWorks P2000	ST1	8	teils Serverrack 1
Telefon	Mitel	6865i	T1	45	

4.5 Messtechnik

4.5.1 Theoretischer Hintergrund

4.5.1.1 Wechselstrom

„Bei Wechselstrom und Wechselspannung spricht man von elektrischen Größen, die in den Einheiten Ampere (A) und Volt (V) angegeben werden, deren Werte sich im Verlauf der Zeit (t) regelmäßig wiederholen. Der Wechselstrom ist ein elektrischer Strom, der periodisch seine Polarität (Richtung) und seinen Wert (Stromstärke) ändert. Dasselbe gilt für die Wechselspannung.“ (Schnabel, kein Datum)

Bei Wechselstrom sind Momentaufnahmen nicht aussagekräftig, da die Spannung im nächsten Moment schon einen ganz anderen Wert annehmen kann. Hierfür gibt es Globale-Kenngrößen, wobei meistens der Effektivwert angegeben wird.

4.5.1.1.1 Allgemeine Parameter von Wechselgrößen

$$f = \frac{1}{T}$$

$$\hat{U} = u_{\max}(t)$$

$$\bar{U} = \frac{1}{T} * \sum_{t=0}^{t=T} u(t) * \Delta t$$

$$|\bar{U}| = \frac{1}{T} * \sum_{t=0}^{t=T} |u(t)| * \Delta t$$

$$U = \sqrt{\frac{1}{T} * \sum_{t=0}^{t=T} u(t)^2 * \Delta t}$$

T ... Periodendauer (s)

f ... Frequenz (Hz)

\hat{U} ... Maximalwert (Scheitelwert, Spitzenwert) (V)

\bar{U} ... arithmetischer Mittelwert (V)

$|\bar{U}|$... Gleichrichtwert (V)

U ... Effektivwert (quadratischer zeitlicher Mittelwert) (V)

(Deimel & Hasnzagl, 2013, S. 10-11)

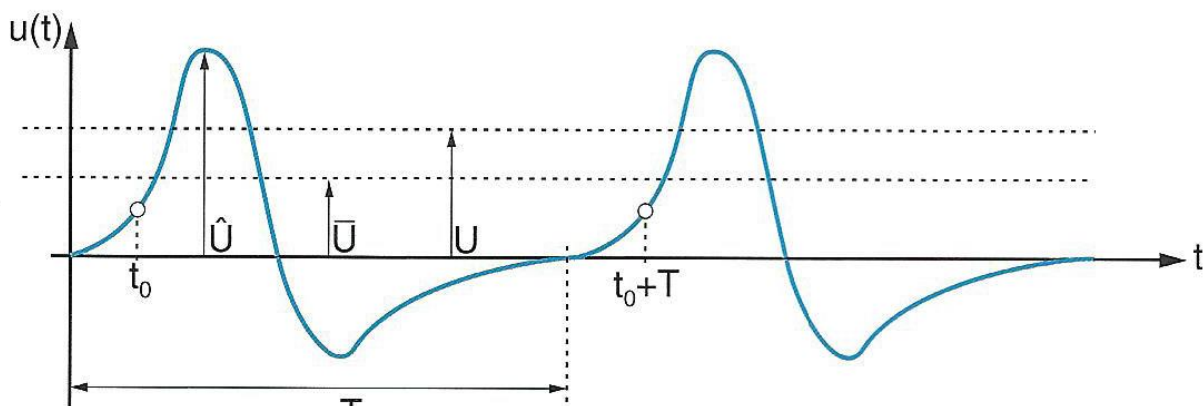


Abbildung 4-1

Beispiel einer Wechselstromkurve mit eingezeichneten Parametern

Quelle: Grundlagen der Elektrotechnik 2 von Franz Deimel und Andreas Hasnzagl Verlag: VERITAS
12.Auflage Seite 10

Es gibt verschiedene Arten von Wechselstrom. In der Elektrotechnik werden hauptsächlich Wechselstrom mit sinusförmigem Verlauf verwendet, da hier die geringsten Verluste und Verzerrungen auftreten. Daher bezieht sich die folgende Beschreibung auf sinusförmigen Wechselstrom. (Schnabel, kein Datum)

4.5.1.1.2 Parameter von sinusförmigen Größen

$$\omega = \frac{\alpha}{t}$$

$$u(t) = \hat{U} * \sin(\omega * \varphi_u)$$

$$\bar{U} = 0 \text{ V}$$

$$|\bar{U}| = \frac{2}{\pi} * \hat{U}$$

$$U = \frac{\hat{U}}{\sqrt{2}}$$

\hat{U} ... Maximalwert (Scheitelwert, Spitzenwert) (V)

\bar{U} ... arithmetischer Mittelwert (V)

$|\bar{U}|$... Gleichrichtwert (V)

U ... Effektivwert (quadratischer zeitlicher Mittelwert) (V)

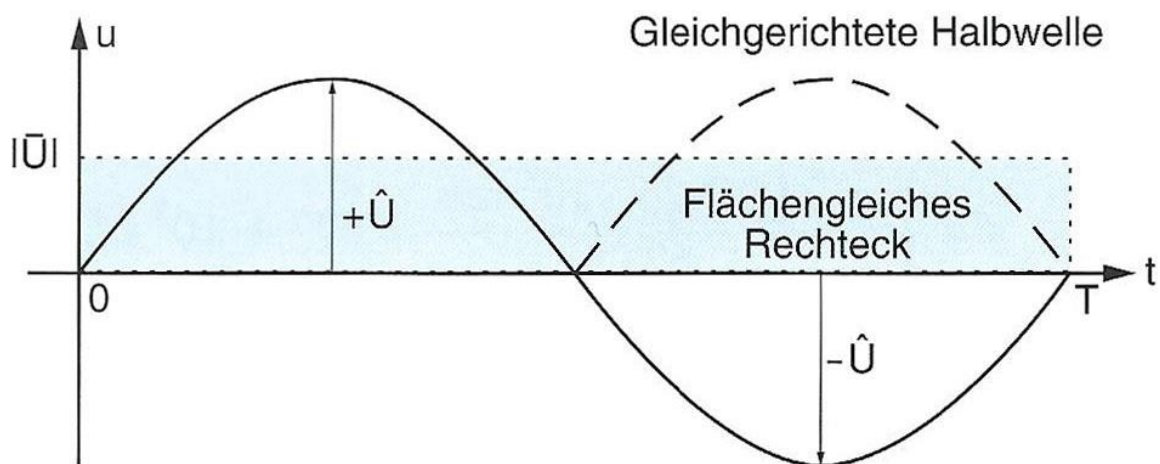


Abbildung 4-2

Beispiel einer Sinusförmigen Wechselstromkurve mit eingezeichneten Parametern

Quelle: Grundlagen der Elektrotechnik 2 von Franz Deimel und Andreas Hasnagel Verlag: VERITAS 12.Auflage Seite 16

4.5.1.2 Leistung bei Wechselstrom

Der Stromverbrauch ist die elektrische Gesamtleistung, welche über einen bestimmten Zeitraum verbraucht wurde. Auf Grund der spezifischen Gegebenheiten der Schulinfrastruktur musste die Leistung bei Wechselstrom betrachtet werden.

Die Leistung kann sowohl als Momentanwerte, als auch als Mittelwerte, welche für den gesamten Stromverlauf Aussagekraft besitzen, angegeben werden.

$u = \hat{U} * \sin(\omega * t)$	<i>u ... Momentanwert der Spannung (V)</i>
$i = \hat{I} * \sin(\omega * t - \varphi)$	<i>i ... Momentanwert der Stromstärke (A)</i>
$p = u * i$	<i>p ... Momentanwert der Leistung (W)</i>
oder	<i>U ... Effektivwert der Spannung (V)</i>
	<i>I ... Effektivwert der Stromstärke (A)</i>
	<i>P ... Mittelwert der Leistung und</i>
	<i>Wirkleistung (W)</i>
	<i>cos(φ) ... Leistungsfaktor</i>
$p = U * I * \cos(\varphi) - U * I * \cos(2\omega * t - \varphi)$	
$P = U * I * \cos(\varphi)$	

(Deimel & Hasnzagl, 2013, S. 98-99)

Bei ohmscher Last entsteht keine Phasenverschiebung (φ) zwischen Spannung und Strom. Bei induktiver oder kapazitiver Last entsteht hingegen eine Phasenverschiebung. Bei Mischformen von allen drei Lasten entsteht eine Phasenverschiebung. Bei Gleichheit von kapazitiver und induktiver Last, ergibt sich eine Phasenverschiebung von Null.

Bei Vorliegen einer Phasenverschiebung entsteht fallweise negative Leistung, da an manchen Punkten Strom und Spannung ein unterschiedliches Vorzeichen haben. Die Wirkleistung wird ermittelt, indem die negative von der Positiven Leistung abgezogen wird. Die Blindleistung ergibt sich aus jenen Teilen, welche sich gegenseitig aufheben, diese wird nicht verbraucht, sondern fließt in den Verbraucher und wieder ins Stromnetz zurück. Die Blindleistung wird beispielsweise zum Aufbau und Abbau eines Magnetfeldes benötigt. Die Scheinleistung ergibt sich aus Wirk- und Blindleistung.

Statt der Phasenverschiebung wird in den meisten Fällen der Leistungsfaktor (cos(φ)) angegeben. Die Wirkleistung wird im Verhältnis zur Scheinleistung größer, je näher der Leistungsfaktor am Wert 1 liegt.

$P = U * I * \cos(\varphi)$	<i>U ... Effektivwert der Spannung (V)</i>
$Q = U * I * \sin(\varphi)$	<i>I ... Effektivwert der Stromstärke (A)</i>
	<i>cos(φ) ... Leistungsfaktor</i>
$S = \sqrt{P^2 + Q^2} = U * I$	<i>P ... Mittelwert der Leistung und</i>
	<i>Wirkleistung (W)</i>
	<i>Q ... Blindleistung (var) (Volt Ampere reaktive)</i>
	<i>S ... Scheinleistung (VA) (Volt Ampere)</i>

(Deimel & Hasnzagl, 2013, S. 99; 101-102)

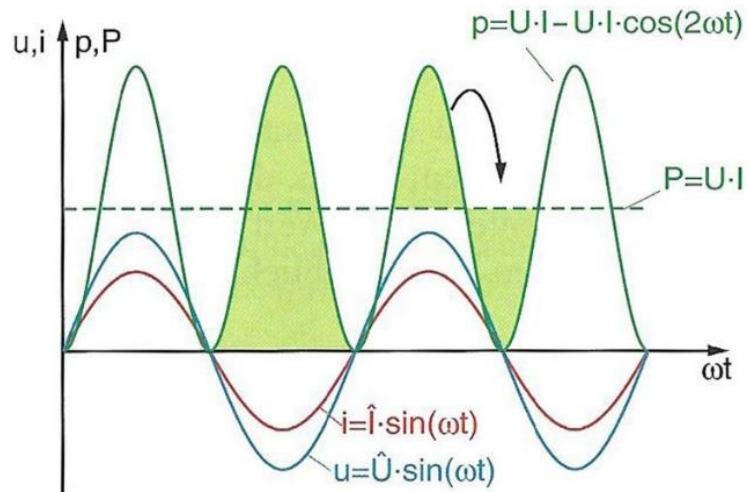


Abbildung 4-3
Leistung bei ohmscher Last

Quelle: Grundlagen der Elektrotechnik 2 von Franz Deimel und Andreas Hasznagl Verlag: VERITAS 12.Auflage Seite 99

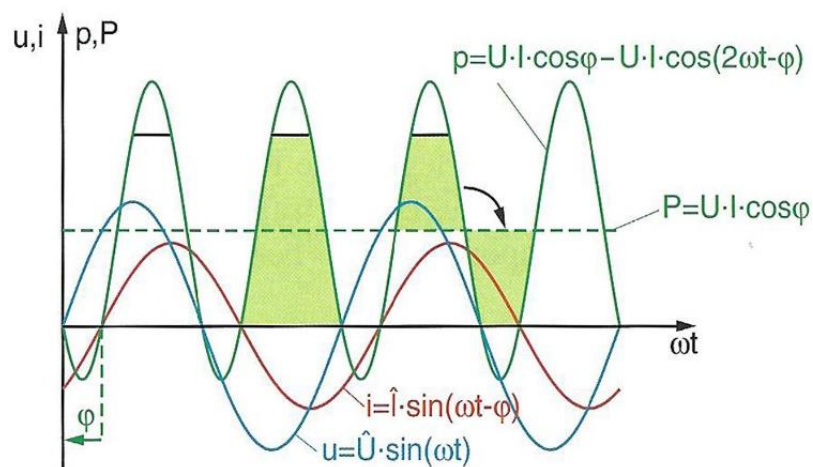


Abbildung 4-4
Leistung bei ohmscher-induktiver Last

Quelle: Grundlagen der Elektrotechnik 2 von Franz Deimel und Andreas Hasznagl Verlag: VERITAS 12.Auflage Seite 99

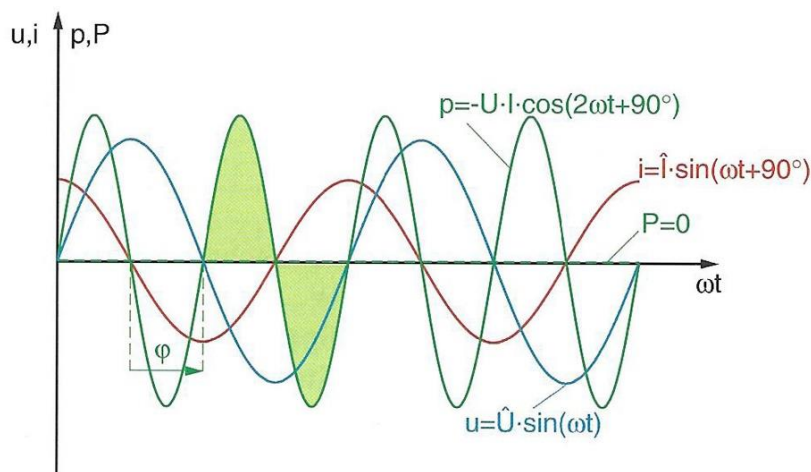


Abbildung 4-5
Leistung bei kapazitiver Last

Quelle: Grundlagen der Elektrotechnik 2 von Franz Deimel und Andreas Hasznagl Verlag: VERITAS 12.Auflage Seite 100

In Österreich verrechnen die Energieanbieter derzeit noch die verbrauchte Wirkleistung. Daher ist es bei der Bestimmung des Stromverbrauches wichtig die Wirkleistung zu messen oder zu berechnen.

Die Energieanbieter planen zukünftig zusätzlich Blindleistung zu verrechnen, da die Zahl der elektronischen Geräte zunimmt, welche in ihren Netzteilen Transformatoren zur Spannungsumwandlung verbaut haben. Diese erzeugen mit ihren Spulen induktive Last und daher Blindleistung. Blindleistung wird nicht verbraucht, belastet aber trotzdem das Stromnetz. Da die Schule besonders viele Geräte besitzt mit Transformator bestückten Netzteilen einsetzt, sind im Schulgebäude mehrere Kompensationskondensatoren als Ausgleichsmechanismus installiert.

4.5.1.3 Stromverbrauch (Energie)

Um den Stromverbrauch bestimmen zu können, wird nicht nur die Leistung, welche zum momentanen Zeitpunkt verbraucht wird, oder der Mittelwert der Leistung ermittelt, sondern auch die Dauer, über welche der Strom verbraucht wurde gemessen. Für eine ideale Berechnung wird das Integral über die Leistung gebildet. Diese Methode kann nur für theoretische Berechnungen verwendet werden, da hierfür für jeden Zeitpunkt der genaue Messwert bekannt sein muss. Mit Messgeräten, welche einen möglichst genauen Leistungsverlauf aufzeichnen können, ist es möglich über den Leistungsverlauf das Integral zu bilden und somit den Energieverbrauch zu ermitteln.

Eine weitere Möglichkeit ist, die mittlere Leistung mit der (Mess-)Dauer zu multiplizieren.

$$E = \int p(t) dt$$

$$E = P * t$$

E ... Stromverbrauch / Energie (Wh)/(J)

p(t) ... Leistungskurve

P ... Mittelwert der Leistung (W)

t ... Dauer (des Verbrauches) (h)

Tatsächlich kann nicht in jedem beliebig kleinen Zeitintervall gemessen werden, daher wird die Leistungskurve zwischen den einzelnen Messpunkten interpoliert. Dieses Verfahren ist nur zulässig, wenn die Abtastfrequenz mindestens doppelt so hoch wie die Netzfrequenz ist.

4.5.1.4 Elektromagnetismus

Ändert sich der Stromfluss in einem Leiter entsteht um den Leiter ein elektromagnetisches Feld.

4.5.1.5 Ruheinduktion

In einem Leiter, der sich in einem sich änderndem Magnetfeld befindet, wird Spannung induziert. Wenn sich der Leiter in einer ruhenden Position befindet, handelt es sich um Ruheinduktion. Das sich ändernde Magnetfeld kann durch einen mit Wechselstrom durchflossenen Leiter entstehen. (Deimel & Hasnzagl, 2013, S. 29-30)

4.5.1.6 Transformator

Transformatoren werden zur Übertragung elektrischer Energie zwischen zwei Stromkreisen, um Spannungen oder Ströme in andere Größenordnungen zu transformieren und um galvanische Trennungen zweier Stromkreise, eingesetzt.

Ein Transformator besteht aus einem magnetischen Kern, welcher von mindestens zwei Spulen umwickelt ist. Die Übertragung der Energie basiert auf Ruheinduktion (siehe 4.5.1.5). Ein Transformator besteht aus einer Primärspule, an der eine Wechselspannungsquelle angeschlossen ist und mindestens einer Sekundärspule, an der Verbraucher angeschlossen sind. An der Primärspule wird, durch einen sich ändernden Strom, ein elektromagnetisches Feld erzeugt, welches durch einen Eisenkern verstärkt werden kann. Das sich ändernde elektromagnetische Feld induziert in den Sekundärspulen eine Spannung. (Deimel & Hasnzagl, 2013, S. 49-50)

4.5.1.6.1 Idealer Transformator bei sinusförmigen Signalen

„Der ideale Transformator weist keine magnetischen oder elektrischen Verluste auf. Sowohl die Primär- als auch die Sekundärspule stellen ideale, widerstandslose Induktivitäten dar. Es treten auch keinerlei Streuflüsse entlang des magnetischen Kreises auf (Kopplungsgrad $k=1$). Die Verluste im Kernmaterial werden vernachlässigt.“ (Deimel & Hasnzagl, 2013, S. 50)

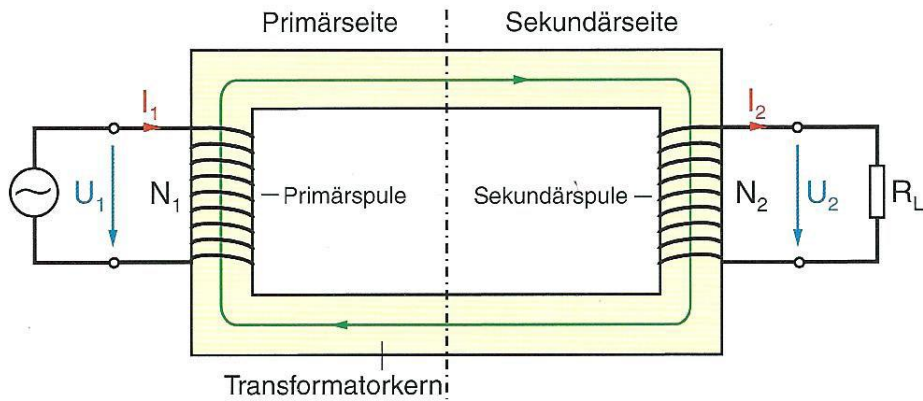


Abbildung 4-6
Schematische Darstellung eines Transformators

Quelle: Grundlagen der Elektrotechnik 2 von Franz Deimel und Andreas Hasnagl Verlag: VERITAS
12.Auflage Seite 49

Es gilt:

$$U_1 \cdot I_1 = U_2 \cdot I_2$$

$$\alpha = \frac{U_1}{U_2} = \frac{I_1}{I_2} = \frac{N_1}{N_2}$$

α ... Übersetzungsverhältnis

U_1 ... Primärspannung (V)

U_2 ... Sekundärspannung (V)

I_1 ... Primärstrom (A)

I_2 ... Sekundärstrom (A)

N_1 ... Primärwindungen

N_2 ... Sekundärwindungen

(Deimel & Hasnagl, 2013, S. 50-51)

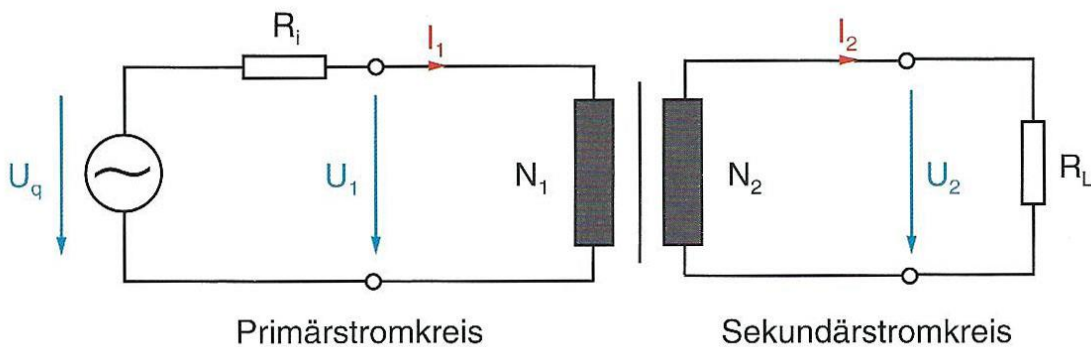


Abbildung 4-7
Schaltbild eines idealen Transformators

Quelle: Grundlagen der Elektrotechnik 2 von Franz Deimel und Andreas Hasnagl Verlag: VERITAS
12.Auflage Seite 50

4.5.1.6.2 Transformator in Bezug auf eine Stromzange

Eine Stromzange wendet das Prinzip eines Transformators an. Hierbei stellt der zu messende, stromdurchflossene Leiter die Primärspule mit einer Wicklung ($N_1=1$) dar. In der Stromzange befindet sich die Sekundärspule, welche den umgesetzten Strom an das Messgerät weitergibt. Die eigentliche Zange bildet den Eisenkern.

Diese Abbildung beschreibt schematisch die Funktionsweise und den Aufbau einer Stromzange.

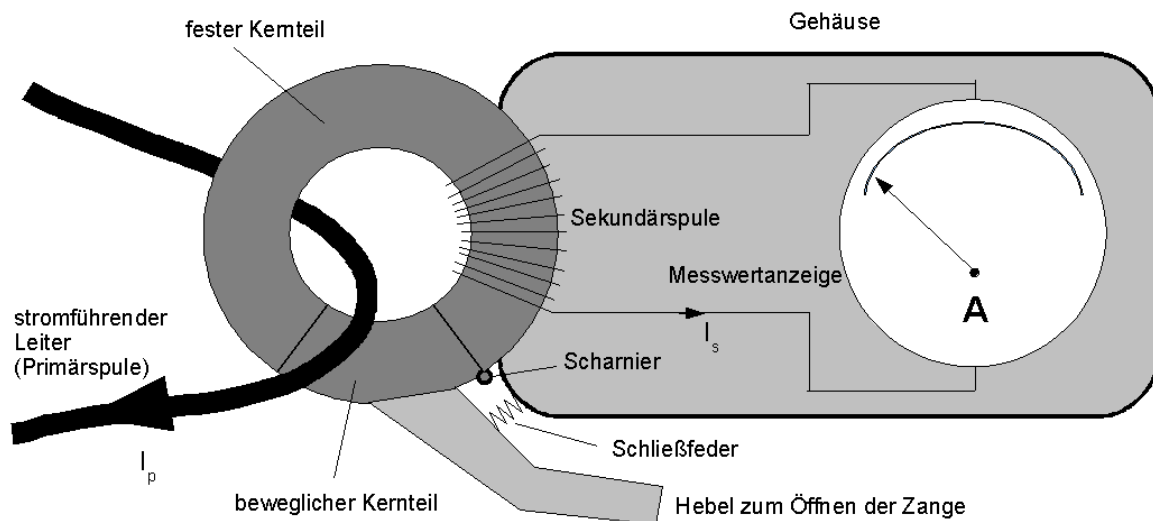


Abbildung 4-8
Schematischer Aufbau einer Stromzange

Quelle: <https://de.wikipedia.org/wiki/Zangenstrommesser#/media/File:Wechselstrom-Zangenamperemeter.png>

4.5.2 Messgeräte/Systeme

4.5.2.1 Haushaltsstrommessgerät

Haushaltsmessgeräte werden verwendet, um den Verbrauch einzelner Geräte oder mehrere Geräte, die an einem Stromverteiler angeschlossen sind, zu messen. Die Haushaltsmessgeräte sind so konfiguriert, dass auf ihnen der Preis einer Verbrauchseinheit (kWh) eingestellt werden kann und so der Gesamtpreis der verbrauchten Energie abgelesen werden kann.

Diese Geräte sind ohne Fachwissen einsetzbar und einfach in der Handhabung. Das Gerät wird durch das Einstecken in eine Schutz-Kontakt-Steckdose aktiviert und beginnt sofort zu messen. In den Schutz-Kontakt-Stecker des Messgeräts wird der zumessende Verbraucher angeschlossen.

Es sind Messgeräte mit flüchtigen oder nichtflüchtigen Speicher verfügbar.

Im Allgemeinen sind diese Messgeräte im unteren Preissegment angesiedelt.

Das Gerät misst Spannung, Strom und den Leistungsfaktor und errechnet damit die aktuelle Wirkleistung (W). Durch Messung der Laufzeit wird der Stromverbrauch (kWh) errechnet und angezeigt. Welche Messgrößen und Formeln verwendet werden sind im Kapitel Theoretischer Hintergrund (4.5.1).

Die Anzeige der Anzahl der Nachkommastellen im LC-Display ist nicht relevant, da die Varianz des Messergebnisses (kWh) laut Datenblatt nur für die erste Nachkommastelle signifikant ist.

Alle Messwerte können auf einem LC-Display abgelesen werden.

4.5.2.1.1 Spezialfall (mit Netzwerkanschluss):

Das Gerät „Expert Power Control 1100“ kam zum Einsatz.

Dieses Messgerät verfügt über viele verschiedene Funktionen, wie SNMP, Webinterface über http und eine Watchdog-Funktion. Es ist auch möglich eine ACL (Access Control List) zu konfigurieren um den Zugriff auf gewünschte Hosts einzuschränken. Zur Kommunikation über den Netzwerkanschluss benötigt das Gerät eine IP-Adresse, welche mittels Verbindungssoftware statisch konfiguriert werden kann oder mittels DHCP-Server eine Adresse beziehen kann. Bei der Zuweisung DHCP besteht der Nachteil, dass ausschließlich über den DHCP-Server die Adresse, welche das Messgerät bezogen hat, ermittelt werden kann. Damit sich diese Adresse nicht mehr ändert, muss die Leas-time am Server für dieses Gerät auf unendlich gesetzt werden.

Sobald eine Netzwerkverbindung aufgebaut worden ist, können komplexere Einstellungen über das Webinterface vorgenommen werden. Mittels Webinterface gibt es auch die Möglichkeit die Messwerte abzulesen.

Für genauere Konfiguration, hier ein Verweis auf die Bedienungsanleitung:
https://www.gude.info/fileadmin/user_upload/products/pdu/1100-1101/anleitung-epc1100-1101.pdf

Link zur Verbindungssoftware:

https://www.gude.info/fileadmin/user_upload/Download/gblconf_v2.7.11.exe

Link zu allen Unterlagen:

<https://www.gude.info/power-distribution/switched-metered-pdu/expert-power-control-1100-1101.html>

4.5.2.1.2 Vergleich verwendeter Haushaltsstrommessgeräte

Die Grafik zeigt das Ergebnis einer Testung von verschiedenen Gerätetypen durch die Stiftung Warentest.

Anbieter und Produkt	Mittlerer Preis in Euro ca.	 test -QUALITÄTSURTEIL	MESS-GENAUIGKEIT 50 % Details 	HANDHABUNG 35 % Details 	SICHERHEIT 10 %	STANDBY-VERBRAUCH 5 %
➤ Aldi (Nord) / Stromkosten-messgerät A	8	BEFRIEDIGEND (3,1)	befriedigend (3,2)	befriedigend (3,4)	gut (2,4)	gut (2,0)
➤ Conrad / Basetech Strom-verbrauchs-überwachung	10	MANGELHAFT (5,0)	gut (2,1)	befriedigend (2,6)	mangelhaft (5,0) 	gut (1,7)
➤ No-Energy NZR SEM 16	50	MANGELHAFT (5,0)	sehr gut (1,2)	gut (2,1)	mangelhaft (5,0) 	befriedigend (2,8)

Abbildung 4-9
Kleine Ansicht der Testergebnisse

Quelle: <https://www.test.de/Strommessgeraete-Nur-eins-ist-gut-1781202-1781218/?row=4&row=7&row=8>

Bei dem Eintrag Aldi (Nord) angeführten Gerät handelt es sich um das Messgerät der Firma Globaltronics (dritte Generation).

Im Projekt kamen die Geräte von Conrad (Base Tech) und No-Energy zum Einsatz. Beim Gerät Aldi Nord wurde hingegen eine weiter entwickelte Version der vierten Generation verwendet.

Für den Projektanwendungsfall kann gesagt werden, dass mit den Messgeräten Base-Tech und No-Energy am besten Ergebnisse erzielt werden. Die Bewertung mangelhaft für diese Geräte ergibt sich aus dem Umstand, dass sich die Geräte bei einer Belastung von 20 A thermisch verformen.

Die maximale Stromstärke die bei den Messungen keinesfalls in die Nähe des kritischen Wertes von 20 A erreicht hatte, konnte mit den verwendeten Messgeräten das Auslangen gefunden werden.

(Stiftung Warentest, 2009)

BaseTech: COST CONTROL 3000

Vorteile:

- Sehr einfach in der Handhabung
- Stiftung Warentest bewertet es mit „gut“ bezüglich Messgenauigkeit

Nachteile:

- Nur flüchtiger Speicher vorhanden
- Hat nur eine Zweistellen Stundenanzeige, dadurch entsteht schnell ein Overflow der Anzeige.
- Messwerte nur über Display ablesbar
- Stiftung Warentest stellte einen groben Mangel fest. Bei hohen Leistungen entwickelt sich im Messgerät Hitze und das Gehäuse kann sich verformen.

Preis: ca. 10,00 €⁴

Globaltronics: GT-PM-04

Vorteile:

- Einfach in der Handhabung
- Nichtflüchtiger Speicher
- Anzeige eines groben Verlaufs

Nachteile:

- Messwerte nur über Display ablesbar
- Stiftung Warentest beschreibt das Messgerät als „schlecht“, da es bei Messungen meist weit über dem tatsächlichen Wert liegt.

Preis: ca. 8,00€⁵

⁴ <https://www.test.de/Strommessgeraete-Nur-eins-ist-gut-1781202-1781218/?row=4&row=7&row=8#>
26.3.2018

⁵ <https://www.test.de/Strommessgeraete-Nur-eins-ist-gut-1781202-1781218/?row=4&row=7&row=8#>
26.3.2018

NZR: No-Energy

Vorteile:

- Einfach in der Handhabung
- Nichtflüchtiger Speicher
- Strompreis lässt sich einprogrammieren
- Zeitanzeige kann auch über mehrere Tage anzeigen
- Laut Stiftung Warentest bezüglich Messgenauigkeit und Bedienung am besten.

Nachteile:

- Messwerte nur über Display ablesbar
- Es ist keine Schwankungsbreite gegeben
- Stiftung Warentest stellte einen groben Mangel fest. Bei hohen Leistungen entwickelt sich im Messgerät Hitze und das Gehäuse kann sich verformen.

Preis: 59,95 €⁶

Gude: Expert Power Control 1100

Vorteile:

- Netzwerkanschluss
- Steuerung ohne Hardware-Zugriff
- viele Zusatzfunktionen
- enthält Steuerungseinheit

Nachteile:

- Kein Display für schnelles Ablesen vorhanden
- Mehr Aufwand nötig, da Software und eine gewisse Basis netzwerktechnischer Kenntnisse vonnöten sind.

Preis: 183,00 €⁷

⁶ <http://www.no-e.de/html/produkte.html>

26.3.2018

⁷ https://shop.gude.info/epages/GudeShop.sf/en_GB/?ViewObjectPath=%2FShops%2FGude-Shop%2FProducts%2F00001100

26.3.2018

4.5.2.2 Multimeter mit Strommesszange

4.5.2.2.1 Multimeter Fluke 289

Zum Einsatz kam Fluke 289 Multimeter, da es eine Möglichkeit zum Aufzeichnen über mehrere Tage bietet. Es wird mit 6 Stück AAA Batterien betrieben.

Das Multimeter wurde von der Schule für die Diplomarbeit zur Verfügung gestellt, welches folgende zusätzliche Komponenten umfasst:

- Infrarot-Adapter
- Magnetaufhängung
- Diverse Messspitzen/klemmen
- Handbuch
- DVD-ROM mit Software (FlukeView Forms)

Preis: 749 Euro ⁸

4.5.2.2.2 Stromzange Fluke i410

Die Stromzange wird benötigt, um eine Phase im Sicherungskasten zu messen. Sie wird mit einer 9V-Block Batterie betrieben.

Die Stromzange mit Bedienungsanleitung wurde von der Schule für die Diplomarbeit zur Verfügung gestellt.

Preis: ab 319 Euro (Preis ohne Kalibrierung, je nach Kalibrierung erhöht sich der Preis)⁹

4.5.2.2.3 Funktionsprinzip:

Die Stromzange ist mit einer Spule mit Eisenkern ausgestattet, welche aufklappbar ist und so um eine Phase befestigt werden kann. Mit Hilfe dieser Spule kann der Strom, welcher in der Phase fließt, gemessen werden. Das genaue Funktionsprinzip wird im Kapitel 4.5.1 beschrieben.

4.5.2.2.4 Messaufbau / Messeinrichtung

Zu beachten ist hier, dass die Batterien von Multimeter und Stromzange nur ca. eine Woche ausreichen!

Für die Messung wird als zentrale Einheit das Multimeter herangezogen, an welches die Stromzange angeschlossen wird.

Folgende Einstellungen müssen getroffen werden:

Das Auswählrad des Multimeters ist auf mV AC zu stellen und ist die Stromzange mit den dafür vorgesehenen Buchsen zu verbinden. Der Offset an der Stromzange ist auf Null einzustellen. Die Stromzange ist an die gewünschte Phase zu hängen. Mit den Funktionstasten ist am Multimeter die Funktion Save->Record auszuwählen und dann mit den

⁸https://www.conrad.at/de/hand-multimeter-digital-fluke-289eur-kalibriert-nach-werksstandard-ohne-zertifikat-grafik-display-datenlogger-cat-i-124366.html?gclid=Cj0KCQiAnuDTBRDUARIsAL41eDr1riifABOUFearlm16cuQH64lu685AwPxjB7FixdmYzDhSQOMaAjZzEALw_wcB&insert=U3

5.2.2018

⁹<https://www.conrad.at/de/fluke-i410-kit-stromzangen-adapter-0-400-bis-3khz-35-05-a-32-mm-121369.html?sc.ref=Service>

5.2.2018

Funktions- und Pfeiltasten die gewünschte Messdauer und das Intervall der Messungen einzustellen. Danach muss die Messung gestartet werden. Mit der Magnethalterung, welche hinten am Multimeter befestigt werden kann, wird eine leichte Befestigung im Sicherungskasten möglich.

4.5.2.2.5 Speicherung der Messergebnisse

Das Messergebnis wird sowohl nach erfolgreicher als auch abgebrochener Messung im Gerät gespeichert, wobei die Benennung des Ergebnisses mit dem Default-Wert „Save“ und einer fortlaufenden Nummer oder einer freien Bezeichnung erfolgen kann.

4.5.2.2.6 Daten auslesen mit Multimeter

Am Multimeter gibt es die Möglichkeit, die gespeicherten Daten direkt am Gerät mittels Funktionstasten über den Punkt „Memory“ auszulesen. Im Menü sind nun alle Arten von Speicherungen zu sehen. Mittels Auswahl des Punktes „Record“ kann mit den Pfeiltasten zwischen den einzelnen Aufzeichnungen gewählt werden. Um sich einen ersten Überblick über die Aufzeichnung verschaffen zu können, lässt sich mit einer Funktionstaste der Punkt „Trend“ auswählen. Daraufhin zeichnet das Multimeter ein grobes Diagramm der Aufzeichnung.

4.5.2.2.7 Daten auslesen mit Computer

Zum Auslesen der Daten vom Multimeter, wird ein Computer mit USB-Anschluss, der Infrarot-Adapter des Multimeters und die Software „FlukeViewForms“ benötigt. Die Software kann mit der beiliegenden DVD-ROM installiert werden. Nach dem Start der Software ist das richtige Art des Messgeräts auszuwählen. Sobald Multimeter und Computer korrekt verbunden sind, ist es möglich das Multimeter auszuwählen und den Speicherinhalt anzeigen zu lassen und die Daten von Multimeter auf den Computer übertragen.

4.5.2.3 System (Raritan)

Im Zuge der Diplomarbeit wurde uns von der Firma BellEquip GmbH für einen Monat ein professionelles Messsystem vom Hersteller Raritan zur Verfügung gestellt. Es handelt es sich um eine BCM2 Einheit, welche für das Messen von Spannung und Strömen an zentralen Punkten (Sicherungsschränke, Verteilerschränke, ...) in großen Betrieben eingesetzt wird.

4.5.2.3.1 Allgemeiner Aufbau und Funktion

Bei der Diplomarbeit wurde eine BCM2 Einheit getestet, welche aus drei grundlegenden Komponenten besteht.

Power Meter Controller (PMC)

Der PMC wird für das Auslesen der Werte benötigt und ist die Schnittstelle für den Anwender. Über diese Schnittstelle können verschiedene Funktionen implementiert/konfiguriert werden.

Mains Meter (PMM)

Der PMM ist das zentrale Element der Messeinheit. An dem PMM wird die Stromversorgung angeschlossen. Der PMM versorgt alle weiteren Komponenten mit Strom und steuert den PMB. Der PMM misst nur die Hauptstromversorgung.

Branch Meter (PMB)

Der PMB verfügt über 96 Anschlüsse für die Branche Circuits. Der PMB misst, mit Hilfe der Branche Circuits, den Strom, welcher von der zu messenden Phase verbraucht wird. Die Messwerte werden an den PMM übermittelt.

Zusätzlich werden auch die dazugehörigen Kabel, Verbindungsplatine (zwischen PMM und PMB), RJ45-Kabel (zwischen PMM und PMC) und die Branche Circuits (Funktion siehe Prinzip Strommesszange 0) benötigt.

Die gesamte Setup-Anleitung des Herstellers ist auf der beiliegenden CD zu finden.

4.5.2.3.2 Aufbau laut Hersteller

Der PMC kann in einem beliebigen Rack mit 19“ Einschüben platziert werden. Der PMC benötigt ein RJ45-Kabel als Kommunikationsschnittstelle und Stromversorgung.

PMM und PMB müssen mit der Verbindungsplatine an der Rückseite verbunden werden. Diese beiden Komponenten werden auf einer Hut-Schiene platziert werden. Dies kann in einem Stromverteilerschrank geschehen (siehe Abb. 10-4).

Auf einer der beiden RJ45-LAN-Anschlüsse am PMM wird das Kabel, welches mit dem PMC verbunden ist, angeschlossen. Der andere Anschluss wird bei einer Verkettung von PMM und PMB Einheiten benötigt.

Die Hauptaufgabe vom PMM ist es die Hauptzuleitung zu messen, welche bis zu drei Phasen besitzen exklusive Neutralleiter und Erdung besitzen kann. Wie in der Abbildung 10-4 ersichtlich wird der PMM für die Spannungsmessung parallel angeschlossen. Die Stromstärke wird mittels größerer Eisenkerne durchgeführt, welche mittels eines speziellen Kabels an dem PMM angeschlossen werden. Dieses Kabel unterscheidet sich von den anderen Verbindungskabeln dadurch, dass es einen schmäleren Stecker mit weniger Pins besitzt. Die Beschriftung der einzelnen Drähte ist nicht chronologisch nummeriert, sondern mit den Bezeichnungen „A,B,C (Phasen), N (Neutralleiter) und E (Erdung)“ versehen. Die Stromversorgung kann über eine normale einphasige 230 V Leitung erfolgen. Hierfür ist der Anschluss mit drei größeren Pins vorgesehen.

Am PMB werden die durchnummerierten Verbindungskabel verwendet, wobei auf die korrekte Anordnung am PMB zu achten ist. An diesen können dann die einzelnen Branche Circuits (kleine Eisenkerne) angebracht werden, welche um die zu messenden Phasen befestigt und so können insgesamt mit einem PMB 96 Endstromkreise gemessen werden.

Da im Elektrotechnikgesetz und den dazu gehörigen Verordnungen (ETV) eine strikte Trennung von spannungsführenden Teilen und nicht spannungsführenden Teilen (LAN-Kabel) vorgesehen ist, muss bei dem Einbau dieses System darauf geachtet werden, dass die LAN-Kabel des PMM gesetzeskonform abgeschirmt und von den spannungsführenden Teilen getrennt verlegt werden.

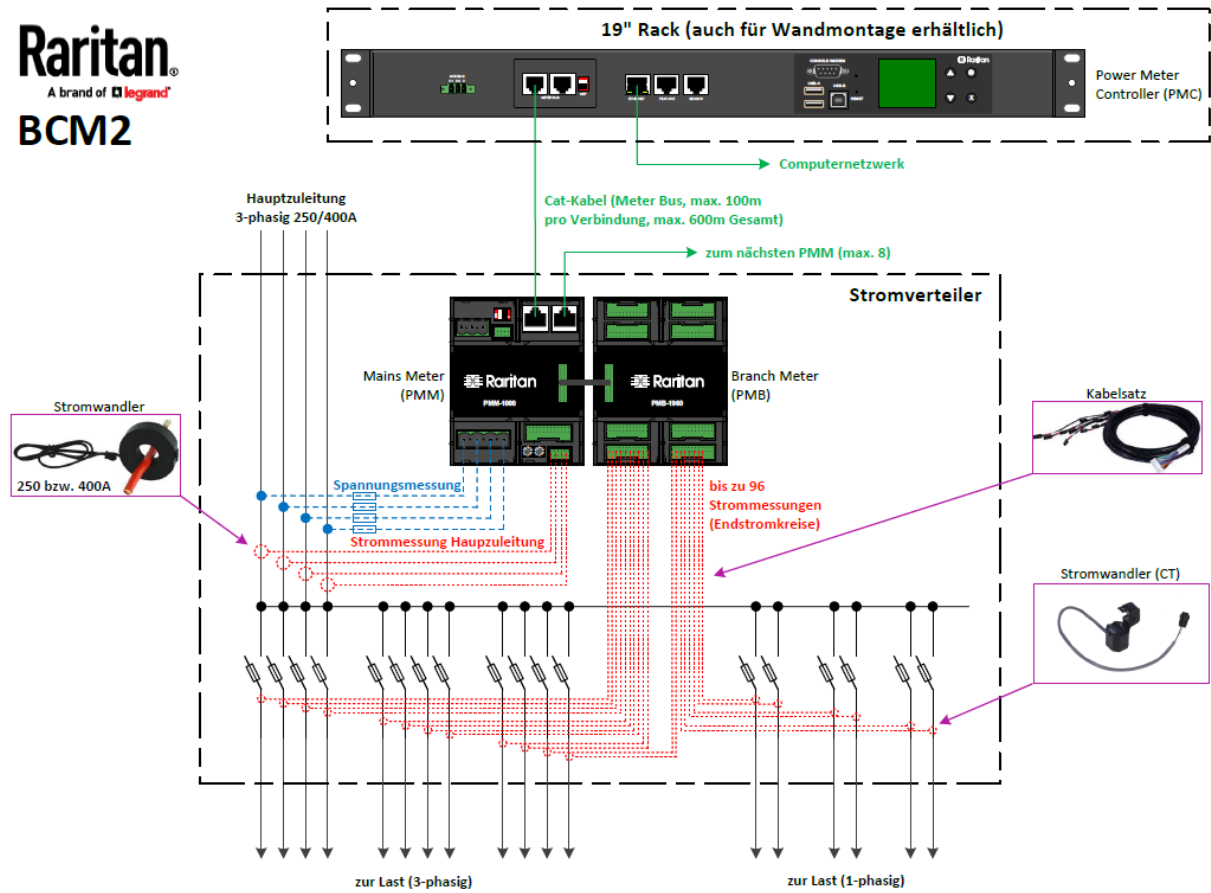


Abbildung 4-10
Aufbau und Anschluss einer BCM2 Einheit

Quelle: genau Quelle unbekannt, wurde von der Firma BelleEquip zugesandt

4.5.2.3.3 Umgesetzter Aufbau

Um dieses Messsystem zu testen, wurde ein Rack mit Servern, welches an vier Stromzuleitungen angeschlossen ist, gemessen. Da die Zuleitung aus einem 3 poligen Kabel (Phase, Neutralleiter und Erdung) besteht, musste zuerst an kurzen Passagen, die äußerste Isolationsschicht abisoliert werden, damit die Branche Circuits um die einzelnen Phasen befestigt werden konnten. Diese vier Branche Circuits (es wurde nur der jeweilige L-Leiter gemessen) wurden über das erste Verbindungskabel am PMB angeschlossen. PMM und PMB wurden in einem kleinen Plastikkästchen mit Hutschiene befestigt. Der PMC wurde in der Nähe eines Servers befestigt, da dieser Server dann via LAN-Kabel mit der externen Zugriffsschnittstelle verbunden ist.

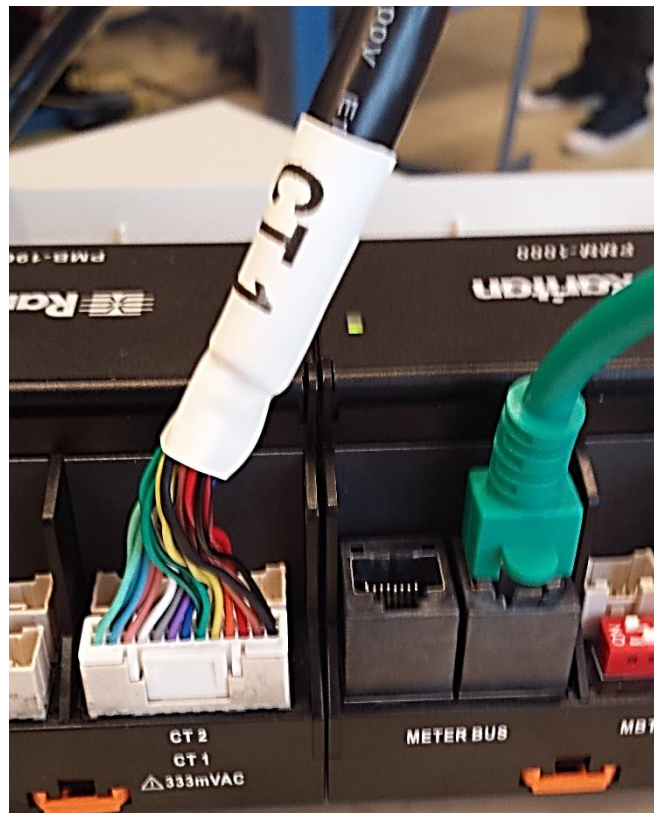


Abbildung 4-11

Foto von Anschluss von Verbindungskabel 1 und Cat5-Kabel Richtung PMC

Für den Anschluss des PMM wurde ein Kabel mit Schutz-Kontaktstecker, welches sich auf die Spannungsversorgung und auf die Spannungsmessung aufteilt, zur Verfügung gestellt. Durch diesen Kabeltyp ergab sich, dass die Hauptzuleitung nur einphasig ausgeführt wurde. Diese Art des Anschlusses unterscheidet sich von der Abbildung 4-11.

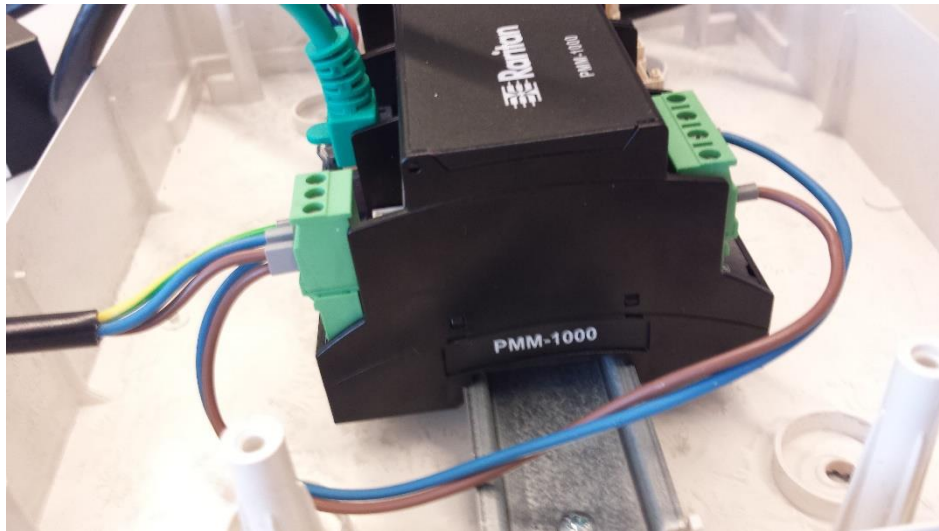


Abbildung 4-12
Foto von Spannungsversorgung und Spannungsmessung

4.5.2.3.4 PMC – Konfiguration

Verbindung

Für die Konfiguration wird ein beliebiger Computer oder Server benötigt, welcher über einen Web-Browser verfügt. Dieser wird mit dem PMC via Cat5-Kable am Port Ethernet verbunden. Danach muss eine Netzwerkverbindung hergestellt werden. Die IP-Adresse des PMC ist am LC-Display unter Menu> Device Information ablesbar. Im Web-Browser ist dann die URL <https://pdu.local> oder die IP-Adresse des PMC einzugeben. Mittels Standard-Administrator-User wird auf die Konfigurationsoberfläche zugegriffen.

Daten des Standard-Administrator:

Username: admin

Password: raritan

Einstellungen

Bei der ersten Konfiguration gilt es zu prüfen, ob der PMC auf Werkseinstellungen gesetzt ist, andernfalls kann am PMC der reset-Knopf gedrückt werden und der PMC wird auf Werkseinstellungen zurückgesetzt.

Am Panel müssen ein Name, der Typ (Anzahl der Phasen der Hauptzuleitung), ob der Modbus aktiviert werden soll und die Maximale Stromstärke auf den Phasen konfiguriert werden.

Wenn auf den Panel-Namen geklickt wird, erscheint das Branche-Circuits Menü. Hier können nun die gemessenen Branche-Circuits angelegt werden, wobei auf die rot markierte Nummer am Ende des Branche-Circuits vom Verbindungskabel zu achten ist.

Wenn alle Branche-Circuits korrekt angelegt wurden, beginnt der PMC diese zu überwachen und die Messdaten aufzuzeichnen. Über das LC-Display des PMC können auch die Momentanwerte abgelesen werden.

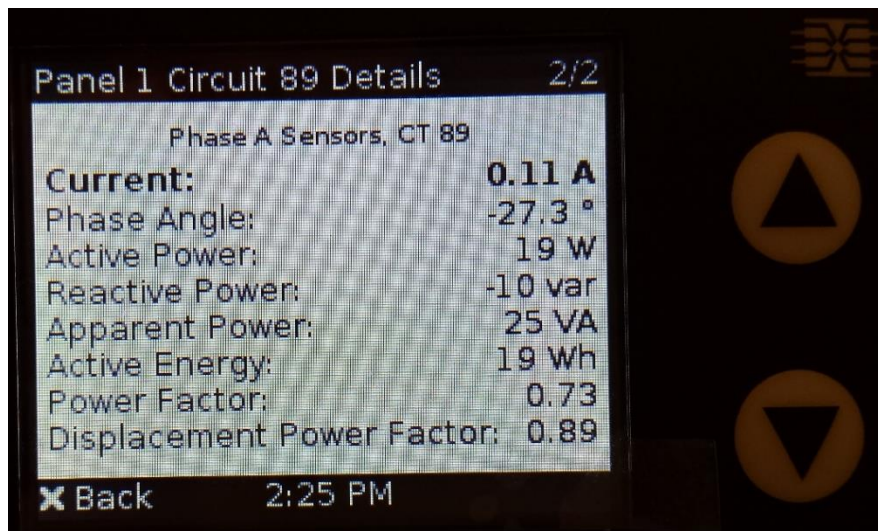


Abbildung 4-13
Anzeige der Circuit-Details am LC-Display des PMCs

4.5.2.3.5 PMC – Messdaten übermitteln

Der PMC bietet viele verschiedene Ausgangsschnittstellen (z.B.: MODBUS-RTU- und die Ethernet-Schnittstelle), um Messdaten an externe Geräte weiter zu leiten.

Für jede Schnittstelle gibt es mehrere Möglichkeiten Nachrichten mit Messdaten zu versenden. Bei der Ethernet-Schnittstelle ist es möglich SMTP, Syslog oder E-Mail zu nutzen.

Im Webinterface kann festgelegt werden wann Nachrichten mit welchem Inhalt versendet werden. Es besteht die Möglichkeit ein Zeitintervall, eine Grenzwertüberschreitung eines Wertes oder fixe Uhrzeiten als Auslöser einer Nachricht einzustellen. Eine zusätzlich schematische Nachricht muss generiert werden. In dieser wird festgelegt welche Messwerte welcher Branche Circuits und ob ein Zusatztext mitgeschickt werden soll.

4.5.2.3.6 PMC – verwendete Einstellungen

Bei der Testung am Server-Rack, wurde als Benachrichtigung Syslog verwendet. Diese Nachrichten wurden in einem 60 Sekunden Takt an den angeschlossenen Server geschickt, welcher diese Nachrichten in csv Dateien abspeichert.

4.5.2.4 Vor- und Nachteile

Vorteile:

- Viele Funktion bezüglich weiterleiten/auslesen der Messdaten
- Ein System welches zentral mehrere Messeinheiten überwachen kann
- PMC zeichnet Gesamtverbrauch für eine Woche auf
- Speicherung eines längeren Verlaufes durch Weiterleitung an Verarbeitungseinheit möglich
- Messeinheiten messen automatisch
- Unterbrechungsfreie Stromversorgung möglich

Nachteile:

- Reiner Anschaffungspreis von mindestens 5000 Euro
- Kompliziert und teuer einzurichten
- Ist nur eine Business-Lösung

4.5.2.5 Alle Messgeräte im Vergleich

Zusammengefasst haben alle Messmethoden und die dazu gehörigen Messgeräte ihr eigenes Anwendungsgebiet.

Messungen mit Haushaltsstrommessgeräten können nur durchgeführt werden, wenn diese Geräte kurzzeitig vom Strom genommen werden, das Haushaltsstrommessgerät eingesteckt wird und danach der Verbraucher angeschlossen wird. Daher ist diese Messmethode für alle aktiven Geräte, welche durchgehend laufen müssen, nicht geeignet. Für Test-Messungen im kleinen Umfang sind Haushaltsstrommessgeräte wiederum gut geeignet, da diese meist leicht zu bedienen sind. Diese Methode ermöglicht eine schnelle Installation und Konfiguration des Gerätes.

Messungen mit Spule und Eisenkern (Strommesszange und die Branche Circuits vom System von Raritan) können durchgeführt werden, ohne den laufenden Betrieb zu stören. Diese Systeme/Messgeräte sind nur für Messungen an zentralen Leitungen/Phasen ausgelegt, sodass die Genauigkeit für Messungen von einzelnen Geräten zu gering wäre. Daher wird diese Methodik meist in Sicherungs- oder Verteilerkästen angewandt. Umso eine Messung den einschlägigen Rechtsvorschriften konform und sicher durchführen zu können, muss ein Elektriker die Messeinheit einbauen. Dies kann hohe zusätzliche Kosten verursachen.

4.6 Messplanung / Messumsetzung

Wie in der Inventarliste erhoben, gibt es viele verschiedene Komponenten im Schulnetz, welche hinsichtlich der Messarten unterschieden werden. Manche Geräte können nicht gemessen werden, da diese aktive Netzwerkkomponenten sind, welche 24 Stunden unterbrechungsfrei in Betrieb sein müssen. Als Ersatz wurden Vergleichsmessungen durchgeführt oder Werte aus den Datenblättern übernommen.

4.6.1 Messung von einzelnen Geräten

Die Einzelmessungen wurden mit Haushaltsstrommessgeräten durchgeführt. Wenn es möglich war, wurden mindestens zwei Geräte pro Gerätegruppe gemessen.

4.6.1.1 Laufender Betrieb

Im laufenden Betrieb wurden die Gerätegruppen des Typs PC und Access-Point gemessen.

Bei den PCs wurden nur die Gerätegruppen gemessen, zu welchen uns der Zugang ermöglicht wurde und welche über 10 % der Gesamtanzahl der Computer ausmachte.

Bei den Access-Points wurden von jeder Gerätegruppe ein Gerät gemessen. Diese Messung wurde in einer Woche vor Ferienbeginn gestartet und in den Ferien beendet. Zu Ferienbeginn wurde ein Zwischenergebnis abgelesen, um so eine Veränderung zwischen Schulbetrieb und Ferien zu erkennen.

4.6.1.2 Benchmark-Tests

Es war geplant PCs, auch während Benchmark-Tests ausgeführt werden, zu messen. Der Hintergedanke dafür war, den Schulbetrieb zu simulieren.

Anforderungen an die Tests waren daher:

- Für viele PCs verfügbar (ein Test welcher mehrmals ausgeführt werden kann)
- Office-Programme simulieren
- Adobe-Programme oder grafikintensive Programme simulieren
- Zeit der Testung frei wählbar

Diese Anforderungen erfüllten einige Benchmark-Tests. Da die Lizenz-Kosten für diese Benchmark-Tests in unserem Budget nicht eingeplant waren, konnten diese nicht eingesetzt werden.

Es wurde gratis Benchmark-Tests gefunden, welche aber eine fixe und zu kurze Laufzeit (für ein repräsentatives Messergebnis) hatten.

Es konnten daher keine Messungen während Benchmark-Tests durchgeführt werden.

4.6.1.3 Leerlaufmessungen

Leerlaufmessungen wurden bei Geräten des Typs Router, Switch und Firewall durchgeführt. Im Besonderen war es Ziel dieser Messungen eine Aussage treffen zu können wie viel Strom die Übungsracks im Raum 078 verbrauchen, wenn sie nicht genutzt werden.

4.6.2 Messung von Räumen

Um den Verbrauch von Räume exemplarisch zu ermitteln, war geplant die Hauptstromzuleitung über einen längeren Zeitraum (mindestens eine Woche) zu messen. Ursprünglich bestand die Planung das System von Raritan dafür einzusetzen, da es damit möglich ist mehrere Phasen und Räume gleichzeitig zu Messen. Dieses Messsystem hätte für Ein- und Ausbau einen Elektriker benötigt, was das Projektbudget überstiegen hätte. Daher hat die Schulleitung dieses Vorhaben sehr kurzfristig abgelehnt.

Kurz vor den Weihnachtsferien wurde uns eine Methode mitgeteilt wie das Team einzelne Räume / Phasen vermessen kann. Hierfür wurde das Multimeter mit Strommesszange (Kapitel 0) verwendet, dieses kann wie bereits beschrieben immer nur eine Phase auf einmal messen.

Daher konnte erst nach den Weihnachtsferien mit den Messungen gestartet werden. Bei einer Besichtigung des Verteilerkastens der EDV-Saal Ebene wurde festgestellt, dass jeder Raum an jeweils 3 Phasen angeschlossen ist. Zusätzlich wurde ein bestimmter Lehrer mit elektrotechnischer Ausbildung benötigt. Daher stellte sich heraus, dass nur mehr ein Raum mit allen 3 Phasen über einen Zeitraum von jeweils einer Woche gemessen werden kann. Dabei wurde eine Fehlmessung eingeplant (welche auch benötigt wurde).

4.6.3 Messung von Server-Rack

Nachdem das System von Raritan nicht für die Raummessung eingesetzt werden konnte, wurde entschieden ein Server-Rack im Raum 078 zu messen. Das System wurde zur Messung, wie in Kapitel 0 beschrieben, angeschlossen.

4.7 Messergebnisse

Von allen durchgeführten Messungen werden die Messergebnisse, nach ihrer Messmethode unterschieden, angeführt.

4.7.1 Einzelmessungen

Bei den Einzelmessungen wird weiter nach Gerätetyp unterschieden (wie in 4.4 Gruppierung der Geräte). Die unterschiedlichen Gerätetypen wurden in unterschiedlichen Betriebszuständen gemessen (wie in 4.6.1 Messung von einzelnen Geräten).

4.7.1.1 PCs

Alle PCs wurden während dem laufenden Schulbetriebs ca. eine Woche lang gemessen.

Gruppe PC-A1

Modell: Acer X680; gemessen im Raum: 079

Tabelle 4-4
Messdaten von Gruppe PC-A1

Gerät	Messgerät	Start der Messung	Ende der Messung	Gesamtverbrauch (über gesamte Messdauer) (kWh)
PC18	Base Tech	05.03.2018 17:56	12.03.2018 16:07	1,47
PC19	Base Tech	05.03.2018 17:57	12.03.2018 16:07	1,34

Gruppe PC-A2

Modell: Acer M680G; gemessen im Raum: EDV7

Tabelle 4-5
Messdaten von PC-A2

Gerät	Messgerät	Start der Messung	Ende der Messung	Gesamtverbrauch (über gesamte Messdauer) (kWh)
PC49 / EDV07 07	Base Tech	12.03.2018 16:33	19.3.2018 16:07	0,52
PC33 / EDV08 114	Base Tech	12.03.2018 16:23	19.3.2018 16:17	0,62

Gruppe PC-D1

Modell: Dimotion Mini N5; gemessen im Raum: MEDT-Labor

Tabelle 4-6
Messdaten von Gruppe PC-D1

Gerät	Messgerät	Start der Messung	Ende der Messung	Gesamtverbrauch (über gesamte Messdauer) (kWh)
PC207	NoE	05.03.2018 17:56	12.03.2018 16:07	0,726
PC209	Base Tech	05.03.2018 18:31	12.03.2018 16:07	xxx

Gruppe PC-H2

Modell: HP EliteDesk G1 TWR; gemessen im Raum: 076

Tabelle 4-7

Messdaten von Gruppen PC-H2

Gerät	Messgerät	Start der Messung	Ende der Messung	Gesamtverbrauch (über gesamte Messdauer) (kWh)
PC8	Base Tech	28.02.2018 10:36	05.03.2018 14:25	0,16
PC9	Base Tech	28.02.2018 10:36	05.03.2018 14:25	0,45

Gruppe PC-H3

Modell: HP EliteDesk G2 TWR; gemessen im Raum: ME-Labor

Tabelle 4-8

Messdaten von Gruppe PC-H3

Gerät	Messgerät	Start der Messung	Ende der Messung	Gesamtverbrauch (über gesamte Messdauer) (kWh)
Lehrer-PC	Base Tech	12.03.2018 15:28	19.03.2018 17:55	0,44
PC01	Base Tech	12.03.2018 15:28	19.03.2018 17:55	xxx

Gruppe PC-H4

Modell: HP ProDesk G1 MT; gemessen im Raum: KU1

Tabelle 4-9

Messdaten von Gruppe PC-H4

Gerät	Messgerät	Start der Messung	Ende der Messung	Gesamtverbrauch (über gesamte Messdauer) (kWh)
PC18	Base Tech	05.03.2018 18:10	12.03.2018 14:25	0,69
PC11	Base Tech	05.03.2018 18:07	12.03.2018 14:25	3,71

Gruppe PC-H5

Modell: HP ProDesk G2 MT; gemessen im Raum: EDV4

Tabelle 4-10

Messdaten von Gruppe PC-H5

Gerät	Messgerät	Start der Messung	Ende der Messung	Gesamtverbrauch (über gesamte Messdauer) (kWh)
PC3	Base Tech	22.01.2018 17:55	28.02.2018 9:50	3,31
PC4	Base Tech	22.01.2018 17:54	28.02.2018 9:50	4,39
PC8	Base Tech	22.01.2018 17:59	28.02.2018 9:50	3,16

Gruppe PC-H7

Modell: HP ProDesk G4 MT Business PC; gemessen im Raum: EDV7

Tabelle 4-11
Messdaten von Gruppe PC-H7

Gerät	Messgerät	Start der Messung	Ende der Messung	Gesamtverbrauch (über gesamte Messdauer) (kWh)
PCx / EDV07 99	Base Tech	12.03.2018 16:24	19.03.2018 16:07	1,05
PC47 / EDV07 10	Base Tech	12.03.2018 16:22	19.03.2018 16:07	1,61

4.7.1.1.1 Nicht gemessene PCs

Zu den nicht gemessenen PCs gehören alle nicht identifizierbaren (im Mengengerüst unbekannt oder diverse) und mengenmäßig vernachlässigbare PCs, für welche später ein Durchschnittsverbrauch angenommen wird.

4.7.1.2 Netzwerkkomponenten + Übungsracks

Die Access Points wurden für insgesamt eine Woche gemessen, dabei wurden sie für drei Tage (bzw. 5 bei Gruppe AP1) während des laufenden Schulbetriebs und anschließend für 4 Tage während der Ferien, um so den Leerlaufstrom zu ermitteln, gemessen.

Alle Messungen des Übungsracks sich Leerlaufmessungen. Dabei wurden diese über einen Zeitraum von ca. 5 Tagen gemessen, wobei die Messdauer nicht ausschlaggebend ist, da die Geräte im Leerlauf durchgehend den annähernd gleichen Stromverbrauch aufweisen.

Gruppe AP1

Modell: Fortinet AP832i; gemessen in Raum: 079

Tabelle 4-12
Messdaten von Gruppe AP1

Gerät	Messgerät	Start der Messung	Zwischenergebnis (Ferienbeginn)	Zischen-Verbrauch (kWh)	Ende der Messung (in den Ferien)	Gesamtverbrauch (über gesamte Messdauer) (kWh)
AP 079 / Fortinet AP832i	Base Tech	19.03.2018 18:01	23.03.2018 16:42	0,85	28.03.2018 11:04	1,87

Gruppe AP4

Modell: Cisco AIR-LAP1242AG-E-K9; gemessen in Raum: 209

Tabelle 4-13
Messdaten von Gruppe AP4

Gerät	Messgerät	Start der Messung	Zwischenergebnis (Ferienbeginn)	Zischen-Verbrauch (kWh)	Ende der Messung (in den Ferien)	Gesamtverbrauch (über gesamte Messdauer) (kWh)
AP Klasse (209) / AIR-LAP1242AG-E-K9	Base Tech	21.03.2018 13:47	--	--	28.03.2018 10:35	0,32

Gruppe AP5

Modell: Cisco AIR-LAP1262N-E-K9; gemessen in Raum: EDV7

Tabelle 4-14
Messdaten von Gruppe AP5

Gerät	Messgerät	Start der Messung	Zwischenergebnis (Ferienbeginn)	Zischen-Verbrauch (kWh)	Ende der Messung (in den Ferien)	Gesamtverbrauch (über gesamte Messdauer) (kWh)
AP EDV7 / AIR-LAP1262N-E-K9	Base Tech	21.03.2018 13:45	23.03.2018 16:46	0,75	28.03.2018 10:37	2,43

Tabelle 4-15
Vergleich der Messergebnisse von APs

Gruppe	Verbrauch über gesamte Messdauer (kWh)			Messdauer (h)			Verbrauch (Energie) pro Stunde (kW)		
	bis Ferien	innerhalb der Ferien	Gesamt	bis Ferien	innerhalb der Ferien	Gesamt	bis Ferien	innerhalb der Ferien	Gesamt (Mittelwert)
AP1	0,85	1,02	1,87	94,7	138,4	233,1	0,009	0,007	0,008
AP4	-	-	0,32			164,8			0,002
AP5	0,75	1,7	2,45	51,0	113,8	164,8	0,015	0,015	0,015

Es wurden die Messergebnisse vor den Ferien, mit denen während der Ferien verglichen. Dabei stellte sich heraus, dass die einzelnen Geräte im Leerlauf und unter Last nahezu denselben Strom verbrauchen. Bei der Gruppe AP4 konnte das Messgerät kurz vor Ferienbeginn nicht abgelesen werden. Der Endmesswert ist stark unterschiedlich zu den anderen Geräten. Es wird ein Messfehler vermutet.

Gruppe FW3

Modell: Cisco ASA 5510; gemessen in Raum: 078

Tabelle 4-16
Messdaten von Gruppe FW3

Gerät	Messgerät	Start der Messung	Ende der Messung	Gesamtverbrauch (über gesamte Messdauer) (kWh)
3.12	Base Tech	23.03.2018 16:41	28.03.2018 09:31	6,61
3.13	Base Tech	23.03.2018 16:41	28.03.2018 09:32	6,58

Gruppe R1

Modell: Cisco 2621 xm; gemessen in Raum: 078

Tabelle 4-17
Messdaten von Gruppe R1

Gerät	Messgerät	Start der Messung	Ende der Messung	Gesamtverbrauch (über gesamte Messdauer) (kWh)
2.1	Base Tech	01.03.2018 17:04	05.03.2018 14:28	1,23
2.2	Base Tech	01.03.2018 17:04	05.03.2018 14:28	1,32

Gruppe R2

Modell: Cisco 2800; gemessen in Raum: 078

Tabelle 4-18
Messdaten von Gruppe R2

Gerät	Messgerät	Start der Messung	Ende der Messung	Gesamtverbrauch (über gesamte Messdauer) (kWh)
2.5	NoE	02.03.2018 14:19	05.03.2018 14:29	1,650
2.6	Base Tech	02.03.2018 14:19	Fehlgeschlagen	xxx

Gruppe R3

Modell: Cisco 2811; gemessen in Raum: 078

Tabelle 4-19
Messdaten von Gruppe R3

Gerät	Messgerät	Start der Messung	Ende der Messung	Gesamtverbrauch (über gesamte Messdauer) (kWh)
2.12	Base Tech	28.03.2018 15:01	04.04.2018 11:49	4,65
2.13	Base Tech	28.03.2018 15:01	04.04.2018 11:49	4,10

Gruppe R4

Modell: Cisco 2901; gemessen in Raum: 078

Tabelle 4-20
Messdaten von Gruppe R4

Gerät	Messgerät	Start der Messung	Ende der Messung	Gesamtverbrauch (über gesamte Messdauer) (kWh)
3.3	NoE	16.03.2018 14:46	19.03.2018 17:10	2,238
3.4	Base Tech	16.03.2018 14:46	fehlgeschlagen	xxx

Gruppe R5

Modell: Cisco 2911; gemessen in Raum: 078

Tabelle 4-21
Messdaten von Gruppe R5

Gerät	Messgerät	Start der Messung	Ende der Messung	Gesamtverbrauch (über gesamte Messdauer) (kWh)
3.19	Base Tech	28.03.2018 11:09	04.04.2018 11:50	7,17
3.20	Base Tech	28.03.2018 11:09	04.04.2018 11:50	7,12

Gruppe SW3

Modell: Switch CA 2960; gemessen in Raum: 078/076

Tabelle 4-22
Messdaten von Gruppe SW3

Gerät	Messgerät	Start der Messung	Ende der Messung	Gesamtverbrauch (über gesamte Messdauer) (kWh)
1.6	Base Tech	11.01.2018 17:00	15.01.2018 17:31	1,75
Switch aus 076	Base Tech	11.01.2018 17:14	15.01.2018 17:31	1,45

Gruppe SW5

Modell: Switch CA 3560; gemessen in Raum: 078

Tabelle 4-23
Messdaten von Gruppe SW5

Gerät	Messgerät	Start der Messung	Ende der Messung	Gesamtverbrauch (über gesamte Messdauer) (kWh)
1.18	Base Tech	19.01.2018 12:49	22.01.2018 17:48	2,37
1.19	Base Tech	19.01.2018 12:49	22.01.2018 17:48	2,27

4.7.1.3 Sonstiges

Gruppe SE2

Modell: HP ProLiant DL360p Gen8; freies Gerät in Raum 076

Dieser Server wurde mit einer Last von 7 virtuellen Maschinen getestet.

Tabelle 4-24
Messdaten von Gruppe SE2

Gerät	Messgerät	Start der Messung	Ende der Messung	Gesamtverbrauch (über gesamte Messdauer) (kWh)
Server	Base Tech	28.03.2018 10:49	04.04.2018 11:51	24,44

xxx ... fehlgeschlagene Messung

4.7.2 Raummessung

Wie bereits in Kapitel 4.6.2 beschrieben wurde, konnte nur ein Raum (EDV-Saal 4) gemessen werden. Da es bei diesen Messungen über 30.000 Einzelmessungen gab, wird hier das Gesamtergebnis in Diagrammform dargestellt. Sämtliche Ergebnisse sind auf der beiliegenden CD zu finden.

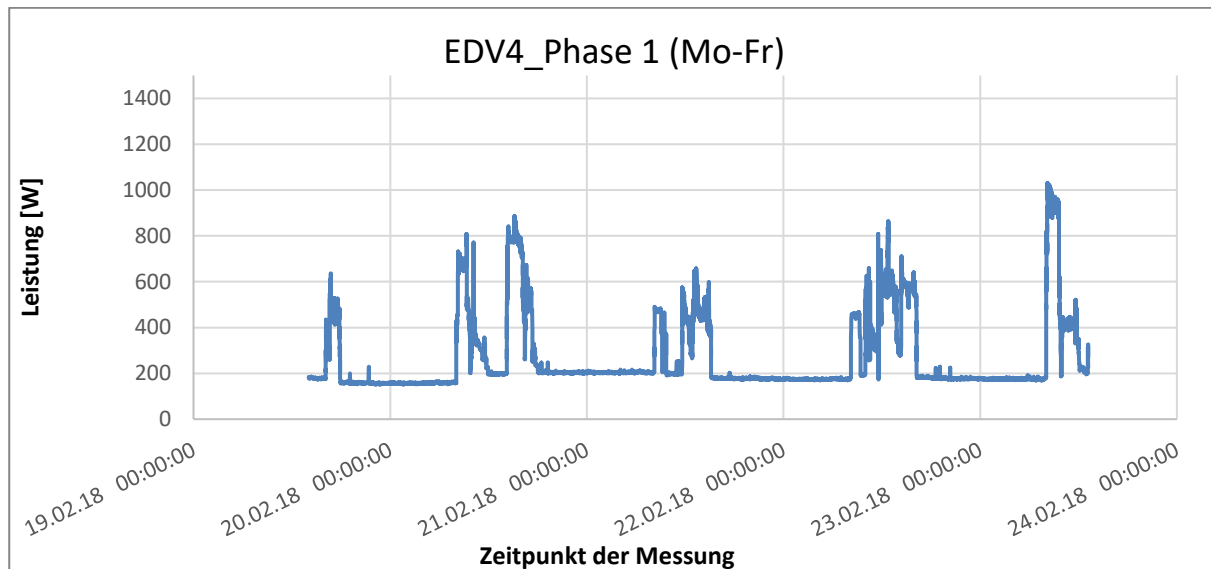


Abbildung 4-14
Stromverbrauch von EDV4 an Phase 1 (Mo-Fr)

Anhand des oberen Graphen ist erkennbar, zu welchem Zeitpunkt der Saal genutzt wurde beziehungsweise wann Unterricht stattgefunden hat. Die Linie des Graphen sinkt nie weit unter 200 W. Daher wird an dieser Phase unabhängig von der Raumnutzung durchgehend 200 W verbraucht.

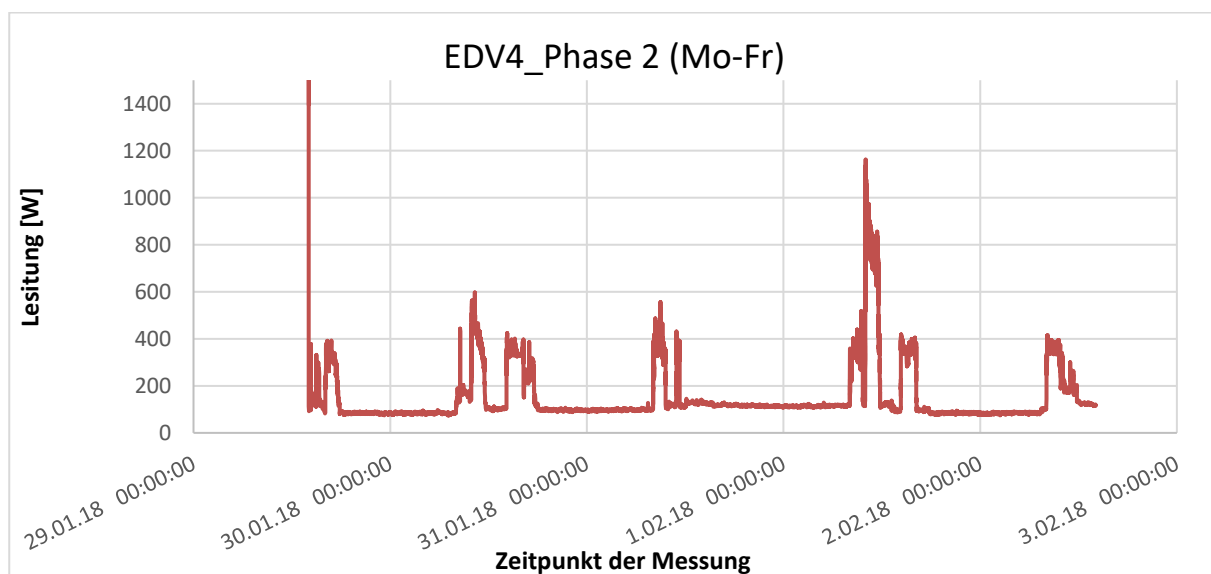


Abbildung 4-15
Stromverbrauch von EDV4 an Phase 2 (Mo-Fr)

Phase 2 weist einen ähnlichen Verbrauch wie Phase 1 auf, jedoch sind der durchschnittliche Verbrauch und der Standby-Verbrauch geringer. Der Standby-Verbrauch liegt hier bei ca. 100 W.

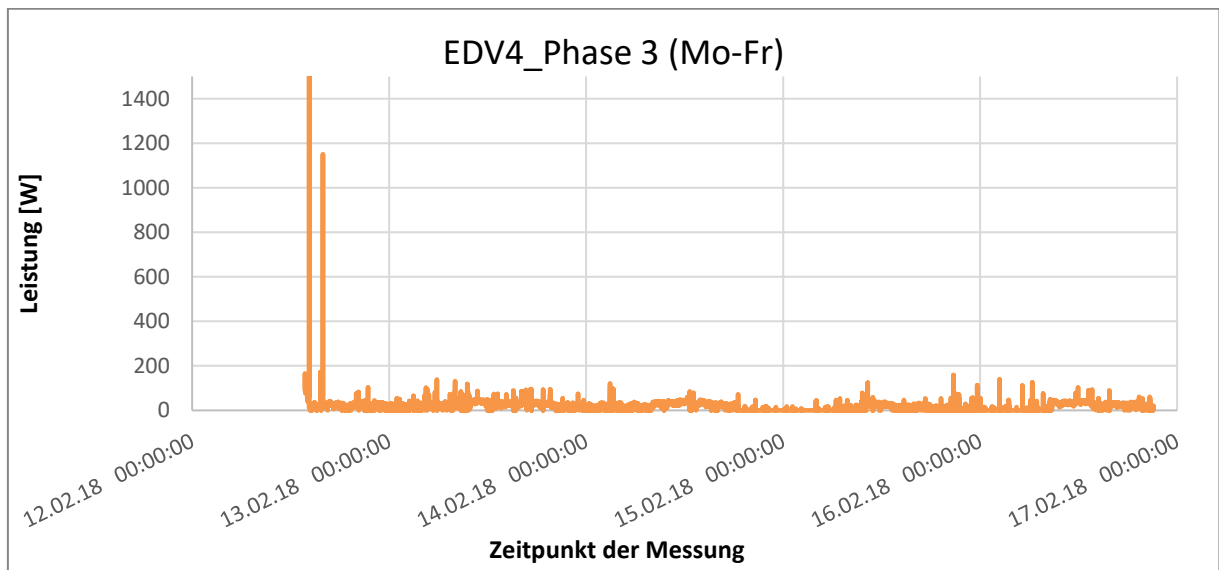


Abbildung 4-16
Stromverbrauch von EDV4 an Phase 3 (Mo-Fr)

An Phase 3 wird nur sehr wenig Strom verbraucht. Es ist nicht bekannt, dass Verbraucher an dieser Phase angeschlossen sind. Diese Messergebnisse könnten durch parasitäre Ströme oder Störungen der Messeinheit (äußere Faktoren wie andere Leiter, die zu nahe an dem Eisenkern entlang verlegt sind) entstehen.

Über alle drei Phasen ergibt sich so ein Gesamtstromverbrauch (Montag-Freitag) von **ca. 48 kWh**.

4.7.2.1 Vergleich mit Einzelmessungen aus EDV4

Tabelle 4-25
Mittelwerte der PCs EDV4

	Verbrauch (kWh)	Messdauer (Tagen)	Verbrauch über 5Tage (kWh)
PC3	3,31		
PC4	4,39		
PC8	3,16		
Durchschnitt	3,62	36,66	0,49

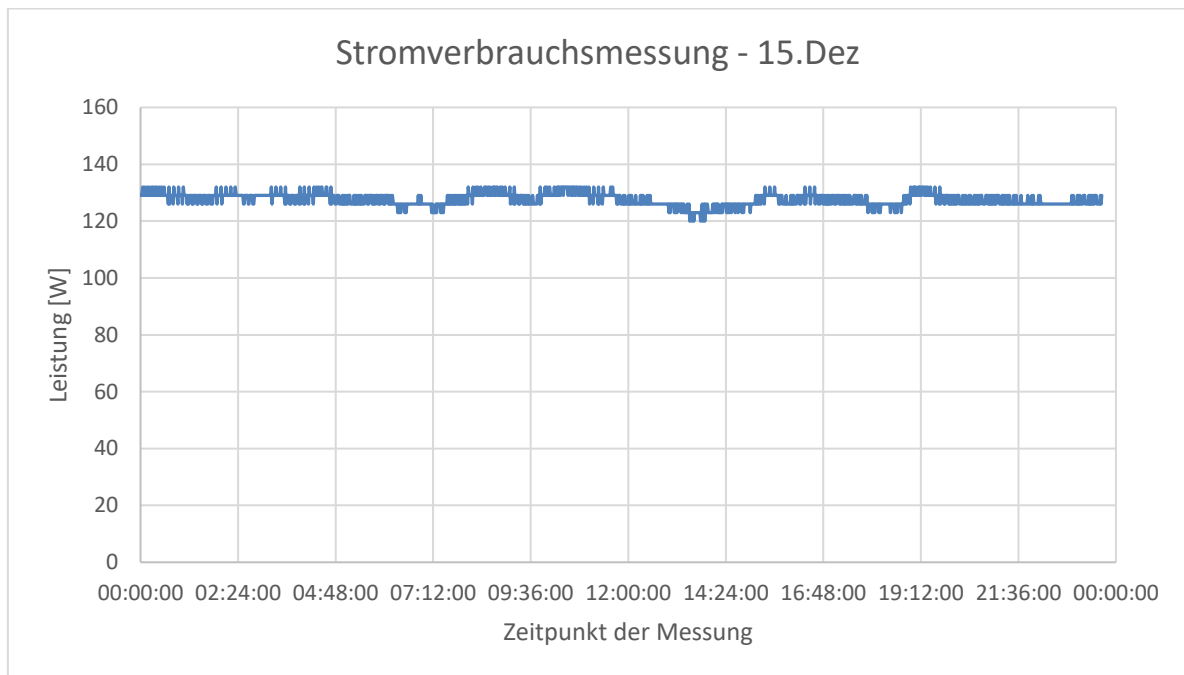
Tabelle 4-26
Hochrechnung der Einzelmessungen EDV4

Geräte in EDV4	Verbrauch
19 PCs	9,37
1 AP	1,80
1 Switch	3,96
Gesamt	15,14

Rein von den Einzelmessungen lässt sich zur Raummessung keine vergleichbare Hochrechnung erstellen. Das lässt sich dadurch erklären, dass bei den gemessenen PCs in dieser Hochrechnung die Laufzeit nicht berücksichtigt wurde.

4.7.3 Server-Rack-Messung

Wie bereits beschrieben wurde die Server-Rack-Messung mit dem System von Raritan durchgeführt. Hierbei wurde über einen Zeitraum von einer Woche der Stromverbrauch aller vier Zuleitungen gemessen. Leider schlugen zwei der Messungen fehl, da die Eisenkerneinheiten von den Leitungen abgegangen sind. Die Messungen der beiden anderen Leitungen wurden ausgewertet. Alle Rohdaten befinden sich auf der beiliegenden CD.



Hier ist der zeitliche Verlauf (gesamter 15. Dezember) der verbrauchten Leistung an einer der Leitungen zu sehen. Dieser pendelt nahezu konstant bei 130 W.

Über den gemessenen Zeitraum von einer Woche sind an der ersten Leitung 20,4 kWh und an der zweiten Leitung 33,8 kWh verbraucht worden. Hochgerechnet auf das gesamte Rack ergibt das einen wöchentlichen Stromverbrauch von **107,2 kWh**.

4.7.4 Hochrechnung Übungsrack

Nachdem alle in den Übungsracks verbauten Komponenten vermessen wurden, konnte eine Hochrechnung des Stromverbrauches erstellt werden. Diese bezieht sich auf das Szenario, in welchem die Übungsracks über das Wochenende nicht abgeschaltet werden und durchlaufen. Hierfür wurden die Komponenten im Leerlauf gemessen.

Die Messergebnisse und deren Messdauer wurden verwendet, um einen Stromverbrauch pro Stunde zu bestimmen. Dieser wurde mit 48 multipliziert, um den Verbrauch aller Geräte pro 48 Stunden zu errechnen. Bei dem grau hinterlegten Wert handelt es sich um einen Durchschnittswert von SW3 und SW5.

Tabelle 4-27
Hochrechnung des Stromverbrauches der Übungsracks über ein Wochenende (48 Stunden)

Gruppe	Anzahl	Verbrauch über Messung 1 (kWh)	Verbrauch über Messung 2 (kWh)	Mittelwert der Messungen (kWh)	Messdauer (h)	Verbrauch pro Stunde pro Gerät (kWh)	Verbrauch pro 48 Stunden (Wochenende) alle Geräte (kWh)
FW4	6	6,61	6,58	6,595	112,8	0,058	16,833
R1	4	1,23	1,32	1,275	93,4	0,014	2,621
R2	7	1,65	xxx	1,65	93,2	0,018	5,951
R3	17	4,65	4,1	4,375	164,8	0,027	21,663
R4	9	2,238	xxx	2,238	74,4	0,030	12,995
R5	6	7,17	7,12	7,145	168,7	0,042	12,198
SW3	9	1,75	1,45	1,6	48,4	0,033	14,288
SW4	2	xxx	xxx	xxx	xxx	0,032	3,034
SW5	7	2,37	2,27	2,32	77,0	0,030	10,124
Summe	67						99,706

4.8 Ist Situation

Es existiert ein Mengengerüst über die IT-Infrastruktur.

Es wurden alle PCs und Netzwerkkomponenten, die für das Team zugänglichen waren, mittels Einzelmessungen vermessen.

Es wurde der EDV-Saal 4 mittels einer Raummessung vermessen.

Das Raritan-Messsystem wurde anhand eines Server-Racks getestet und analysiert.

5 Bericht des ökologischen Fußabdrucks

5.1 Kurzbeschreibung

Um der Schule eine Zusammenfassung der Ergebnisse der Strommessungen liefern zu können, wurde ein Bericht des ökologischen Fußabdrucks verfasst. Diesem kann die Schulleitung die wichtigsten Erkenntnisse sowie Schlussfolgerungen entnehmen. Außerdem wurde eine Liste mit möglichen langfristigen Lösungen samt deren grober Beschreibung zusammengestellt.

5.2 Situation vor dem Projekt

Es existiert keine Übersicht über den aktuellen CO₂-Verbrauch der IT-Infrastruktur.

5.3 Theorie

5.3.1 Grundlagen

5.3.1.1 CO₂

CO₂ oder Kohlenstoffdioxid ist ein nicht brennbares Gas, welches in kleinen Mengen farb- und geruchslos ist. In höherer Konzentration ist es potentiell giftig und riecht scharf oder sauer. Des Weiteren ist es sehr wasserlöslich. Dadurch taucht es auch in sämtlichen Gewässern oder Wasserströmen auf. Beispiele dafür wären das Grundwasser sowie Gletscher. Mit Metalloxiden oder Hydroxiden vermischt, bildet es Salze. Der ursprüngliche Wert, also jener Wert welcher die letzten Millionen Jahre bis zu der zweiten Hälfte des 18. Jahrhunderts vorhanden war, betrug 280 Teile pro Million in der Atmosphäre. Durch die Industrialisierung stieg dieser Wert bis heute rapide an. 2015 betrug er bereits 400 Teile pro Millionen. CO₂ entsteht bei der Verbrennung von Substanzen, welche Kohlenstoffe beinhalten, oder als Produkt von biologischen Vorgängen wie zum Beispiel der menschlichen Verdauung oder der Arbeit der Zellen. Aber CO₂ wird auch durch natürliche Vorgänge wie zum Beispiel Vulkane und Geysire freigesetzt. Diese lösen das Kohlenstoffdioxid aus dem Boden und dem Wasser. Aber auch Erdbeben, sowie Muren und Lawinen können das Gas freisetzen, da dabei Sedimentsbecken welche CO₂ binden, gelöst werden. Kohlenstoffdioxid kommt in der Atmosphäre, Hydrosphäre und Lithosphäre vor, sowie in jedem Wald und Moor. Diese speichern aufgrund ihrer Beschaffenheit große Mengen CO₂ und binden es dauerhaft. Die größte Menge CO₂ befindet sich jedoch in den Ozeanen. Diese speichern circa 50-mal mehr CO₂ als die Atmosphäre. Photosynthese erlaubt das Binden eines gewissen Anteils des, in der Atmosphäre vorhandenen, CO₂s. Dies funktioniert jedoch nur in den oberen Schichten, da nur dort genügend Sonnenlicht dafür vorhanden ist.

Das Festland, insbesondere die Permafrostgebiete sowie borealen Nadelwälder, sind für die Bindung des CO₂ verantwortlich.

5.3.1.2 Die Auswirkungen von CO₂ auf die Umwelt

Pflanzen nehmen viel von dem freigesetzten CO₂ auf und des Weiteren verändern sie noch die Verdunstung auf der Erdoberfläche was ebenfalls verhindert, dass jenes CO₂ welches in dem Wasser gebunden ist, nicht erneut gelöst wird. Pflanzen geben außerdem Wasser ab, welches kein CO₂ enthält, und kühlen damit die Luft ab. Dieser Vorgang wird Evapotranspiration genannt und dient als Klimaanlage für die Umgebung. Durch die Poren, die sogenannten Stomata, welche dafür verantwortlich sind, wird jedoch auch das CO₂ aufgenommen. Ein zu hoher CO₂-Gehalt kann jedoch die Größe dieser Poren verringern wodurch diese nicht mehr genügend Wasser abgeben können und die Umgebungstemperatur somit wieder steigt. Dies ist der Grund für 16% der Erderwärmung. Dieser Wert wurde durch die Hochrechnung der Ergebnisse eines Experimentes ermittelt. Diese Experimente wurden von den Forschern Carnegie Chris Field und Joe Berry durchgeführt. Die restlichen 84% sind die Folge des Treibhauseffektes. In Gebieten, in denen dichtere Wälder vorkommen ist die Auswirkung jedoch bei über 25%. Gebiete, die darauf zutreffen wären, zum Beispiel Nordamerika und Ostasien. Aber auch Teile von Österreich zählen dazu. Zur Verdeutlichung dieser Verhältnisse bietet sich dieses Szenario an. Falls sich die Menge an CO₂ in der Atmosphäre verdoppeln würde, wäre das Ergebnis eine Steigerung der Temperatur um vier Grad. Dabei würden drei Viertel von dem Treibhauseffekt stammen und ein Viertel von der Verkleinerung der Pflanzenporen und der daraus resultierenden verringerten Wasserabgabe an die Atmosphäre und somit der Temperaturregulierung. Außerdem steigt dadurch die Wasserabgabe in den Boden. Eine weitere Auswirkung der Steigerung der Lösungsrate des CO₂s ist, dass mehr Kohlensäure in den Gewässern gebildet wird. Die Meere haben circa 28% des CO₂s aufgenommen welches von den Menschen freigesetzt wurde. Also Folge davon steigt der Säureanteil des Wassers. Außerdem steigt die Temperatur des Wassers als Folge der Aufnahme ebenfalls an. Dies resultiert jedoch in einem geringeren Sauerstoffgehalt was zu einem Problem der Meeresflora und Meeresfauna werden könnte. Eine andere offensichtlichere Folge ist das Abschmelzen des Eises. Dies resultiert in einem konstanten Ansteigen des Meeresspiegels und dem Schrumpfen der Polkappen. Außerdem hat dies Auswirkungen auf die meeresnahen Gebiete, welche dadurch ebenfalls erwärmt werden und im Falle eines Dauerfrostbodens weiteres CO₂ freisetzen. Jedoch hat die Steigerung des CO₂-Wertes nicht nur negative Auswirkungen. Viele Pflanzen können dieses CO₂ gut verwerten und wachsen als Folge dessen schneller. Bei Weizen, Kartoffeln und Soja, denn sogenannte C3-Pflanzen, kann dies, zum Beispiel bei einer Verdoppelung des CO₂-Anteils eine Steigerung von einem Fünftel zur Folge haben. C3-Pflanzen sind mit knapp 90% Anteil an allen Pflanzenarten die mit Abstand am häufigsten vorkommende Art. Bei C4-Pflanzen hat die Steigerung des CO₂-Anteils in der Atmosphäre auch positive Auswirkungen jedoch sind diese zu gering, um wirklich merkbar zu sein. Die Steigerung des Ertrages bewegt sich dabei im einstelligen Prozentbereich. Am Meeresboden gibt es ähnliche Auswirkungen. Durch das Lösen des

CO₂s im Meeresboden werden auch die Nährstoffe gelöst, wodurch die Pflanzen besser wachsen können. Auf die Tierwelt hat dies jedoch katastrophale Auswirkungen. Durch den erhöhten Anteil von CO₂ im Wasser fiel die Anzahl der Tiere auf 20% des ursprünglichen Wertes. Dies wurde mittels eines Experimentes, bei welchem Sand von einem Standort an dem Kohlenstoffdioxid austritt an einen anderen Standort gebracht und dem dies nicht der Fall ist. Diese Studie wurde von dem Max-Planck-Institut für Marine Mikrobiologie in Bremen und Katja Guilini von der Universität Gent in Belgien durchgeführt. Eine natürliche Selektion ist die Folge des Anstiegs. Nur jene Arten welche sich schnell anpassen oder schon vor dem Experiment die neuen Umgebungsbedingungen aushalten konnten, verblieben in den Gebieten. Eine andere Weise wie sich der steigende Anteil auf die Temperatur auswirkt, liegt in der Eigenschaft Strahlung zu reflektieren, begründet. Rund 30% der Strahlung welche die Erde trifft, wird von Eis, Schnee sowie anderen Oberflächen reflektiert und strahlt in den Weltraum ab. Der restliche Anteil wird von Gewässern und dem Land absorbiert und strahlt diese als Wärme wieder ab. Dadurch steigt die Temperatur auf der Erde ebenfalls stark an. Während der globale Durchschnitt der Heizwirkung pro Quadratmeter bei 0,27 Watt in den letzten zehn Jahren lag, so ist der steigende CO₂-Anteil für 0,2 Watt pro Quadratmeter verantwortlich.

5.3.1.3 Der Einfluss der IT-Infrastruktur auf den CO₂-Haushalt

Aufgrund des zunehmenden Einsatzes von IT-Ressourcen, sowohl in der Industrie als auch in den privaten Haushalten, und der steigenden Nachfrage nach der Leistung, dem Speicherplatz als auch der Anbindung an das Internet, ist der Einfluss der IT auf die Umwelt immer weiter gestiegen. Alleine durch die Herstellung werden bereits viele Schadstoffe freigesetzt. Da die Ressourcen zumeist toxisch sind, ist die Auswirkung davon auch entsprechend hoch. Hinzukommt die kurze Lebensspanne der IT-Ressourcen. Die durchschnittliche Verwendungsdauer liegt bei 3,5 Jahren. Daraus resultiert auch die hohe Freisetzung an Gasen in die Atmosphäre. Heutzutage werden mehr als 60 verschiedene Elemente verwendet um einen Mikroprozessor zu bauen. Vor 30 Jahren hingegeben waren es gerade einmal 12. Beispiele für die heute verwendeten Elemente sind Palladium, welches in den Medien in den letzten Jahren für Aufsehen gesorgt hat, da es ein seltenes Element ist und in vielen modernen Geräten verwendet wird, Niob, oder früher auch Columbium genannt, sowie Kobalt, welches hauptsächlich in Batterien verwendet wird. Obwohl die IT-Geräte keine nennenswerte Absonderung von Emissionen während der Laufzeit haben, so verursachen sie diese durch den Gebrauch von Strom welcher als Energiequelle dient. Die IT-Ressourcen der Erde sind für 2 Prozent der CO₂-Emissionen verantwortlich. Auch wenn dies gering erscheint, so ist dies das Feld welches am schnellsten wächst. Laut Experten wird der Anteil bis zum Jahre 2020 auf bis zu 2,3 Prozent ansteigen. Dabei sei jedoch noch angemerkt, dass der Anstieg exponentiell erfolgt. Den größten Teil des Stromverbrauchs machen dabei Rechenzentren und Serverfarmen aus. In den Jahren 2000 bis 2005 hat sich in diesem Teil der IT der Stromverbrauch mehr als verdoppelt.

Durch Green-IT, bei der es sich in diesem Fall um die Virtualisierung von Servern, modernere stromsparendere Geräte sowie effizientere und zumeist passive Kühlkonzepte handelt, betrug der Anstieg in den Jahren von 2005 bis 2010 nur noch 56 Prozent. Die Anzahl der im Einsatz befindlichen Server, betrug, soweit dies festgestellt werden konnte, rund 32 Millionen. Diese verbrauchten insgesamt circa 237 Milliarden kWh pro Jahr. Ein weiterer Punkt, welcher große Auswirkungen auf die Umwelt haben kann, ist die Entsorgung der IT-Komponenten. Da diese viele schädliche Substanzen beinhalten, müssen sie gesondert entsorgt werden. Dies ist jedoch meist nicht der Fall. Geräte die bei uns ausrangiert wurden, da sie bereits veraltet sind, werden oft in Entwicklungsländer verkauft, wo die gesetzlichen Vorgaben zur Entsorgung jedoch nicht so streng sind. Dadurch landen Elektroschrotteile häufig auf deren Mülldeponien was der Umwelt Schaden zufügt. Der anfallende Elektroschrott, welcher pro Jahr um beinahe um 10 Prozent steigt, lag 2012 bei 46 Millionen Tonnen.

Auf der anderen Seite wird die IT-Infrastruktur auch dadurch positiv beleuchtet, da sie es ermöglicht effizienter zu arbeiten. Dadurch sinken in einem Unternehmen die Emissionen in der anderen Abteilung was insgesamt zu einer Abnahme führen kann. Die nennt man auch Enabler-Effekt. Laut einer Studie wird der Faktor welcher das Verhältnis zwischen dem zusätzlichen Energieverbrauch durch die IT-Komponenten sowie der Einsparung durch eben jene, 2020 rund bei 7 liegen. Diese Studie wurde von der Global e-Sustainability Initiative in Auftrag gegeben. Auch wenn die IT einem Ansteigen der Emissionen unterliegt, so trägt sie doch auch dazu bei, dass der Verbrauch gesenkt wird. Je nach Umfang der von der IT beeinflussten Infrastruktur ist dieser Wert höher oder niedriger. Für die Unternehmen ist dabei weniger die Sorge um die Umwelt der Beweggrund, sondern der geringere Stromverbrauch oder die positive Publicity, welche sie dadurch erreichen können.

5.3.2 Der Bericht über den ökologischen Fußabdruck

5.3.2.1 Voraussetzungen für einen Bericht über ökologischen Fußabdrucks

Die Voraussetzungen, um den Bericht zu erstellen, setzen sich aus den Informationen die man dazu braucht, das Wissen zu deren Verarbeitung und zuletzt der Interpretation der Ergebnisse zusammen. Bei den Informationen ist insbesondere die richtige Auswahl entscheidend. Sollten es zu viele Daten oder die falschen Daten sein, kann es passieren, dass das Ergebnis irreführend ist oder nicht aussagekräftig. Bei dem ökologischen Fußabdruck wäre das zum Beispiel der Fall, falls die Zeiträume in denen die verschiedenen Daten ermittelt werden unterschiedlich sein sollte. Das würde dazu führen, dass das Ergebnis verfälscht ist. Falsche Daten wären zum Beispiel die Einbeziehung des Wasserverbrauchs. Diese ist eine so genannte Umlaufgröße, was so viel heißt wie, dass es weder erzeugt noch verbraucht wird und somit nicht relevant ist. Der größte Anteil der Daten, und auch so gut wie der Einzige, welche in Betracht gezogen werden, ist das Kohlendioxid welches bei der Erzeugung, dem Transport, der Verwendung sowie der Entsorgung gelöst wird. Dabei wird auch oft auf den Energieträger Strom zurückgegriffen, da dieser allgegenwärtig ist und somit einen großen Anteil daran trägt. Dies führt dazu, dass viele Berichte des ökolo-

gischen Fußabdrucks heutzutage IT-Komponenten als eigenen Punkt führen. Dieser Bereich ist in vielen Firmen bereits derart relevant und anders zu messen, dass es nötig ist, es eigens zu behandeln. Die Menge an CO₂ wird dabei mithilfe der Fläche an Wald, welche zum Binden des Gases benötigt wird, angegeben. Andere Daten, welche jedoch nicht mehr Teil des Fußabdrucks sind, wären jegliche Form von Abfällen abgesehen von CO₂. Diese werden eingeteilt in die 3 Gruppen „biologisch abbaubar“, „deponierbar“ und „künstliche Produkte“. Letzteres ist auch das größte Problem für die Umwelt, da diese nicht natürlich abgebaut werden, beziehungsweise nicht in einer relevanten Zeit. Was ebenfalls nicht einbezogen wird, sind nicht erneuerbare Ressourcen. Dazu zählen alle Metalle, Gesteine und fossile Brennstoffe. Die Verarbeitung der Daten, beziehungsweise das Wissen dazu, setzt sich aus 2 Punkten zusammen. Erstens ist die Einteilung der Ergebnisse dabei sehr wichtig. Dabei sollten auf keinen Fall, mehrere verschiedene Gruppen, wie die Transport- oder Herstellungsnebenprodukte, welches in diesem Fall CO₂ ist welches betrachtet wird, betrachtet werden. Diese werden komplett verschieden berechnet, wodurch dies zu ungenauen oder falschen Ergebnissen führen würde. Um diese Ergebnisse schlussendlich vergleichen zu können, werden sie alle in eine einheitliche Einheit umgerechnet. Erst daraus ergibt sich dann auch eine Möglichkeit diese global zu vergleichen. Der zweite wichtige Punkt ist die Bewertung und Gewichtung der Ergebnisse, oder anders ausgedrückt der Abweichungen eben jener. Dies ist auch bei den Messungen ein wichtiger Punkt. Falls wir ein Serverzentrum als Beispiel nehmen, so wäre der Stromverbrauch der Punkt bei dem diese Schwankungen auftreten können. Falls diese gemessen werden, während zum Beispiel ein Umbau im Gange ist so sind diese Daten nicht so aussagekräftig wie, wenn der Alltagsbetrieb gemessen wurde. Viele große Firmen messen den Stromverbrauch laufend, wodurch diese Bewertung sehr wichtig wird. Sollte dies nur einzeln, in kleinem Maße, Vorkommen ist es in der Endrechnung nicht sehr relevant, da dann der Fehlerbereich nur sehr gering ist. Sollte dies jedoch durch einige größere Ereignisse entstanden sein, die nicht regelmäßig wiederkehren, so sollten diese, wenn möglich, aus der Rechnung genommen werden und durch einen geeigneten Wert ersetzt werden. Eine Vorgehensweise ist dabei die Messwerte aus einem anderen Bereich welcher unter Normallast, das heißt in einer Zeit in der Alltagsbelastung vorlag, zu ersetzen, um so die Lücken zu füllen. Eine andere Variante ist den Durchschnittsverbrauch dafür zu nehmen. Dieser würde dann über eine längere Zeitspanne bei Normallast berechnet werden und zu der Berechnung herangezogen werden. Der dritte Punkt, welcher auch für das weitere Vorgehen eine große Rolle spielt, ist die Interpretation der Ergebnisse. Dabei wird vor allem auf auffällige Muster und mögliche Verbesserungsmöglichkeiten geachtet. Auffällige Muster wären zum Beispiel, dass um eine bestimmte Uhrzeit mehr Nutzer als gewöhnlich auf einen Server zu greifen. Aufgrund dieser Daten wäre eine mögliche Verbesserung das Hochfahren eines zusätzlichen Servers, welcher nur in dieser Zeit aktiv ist. Dadurch wird die benötigte zusätzliche Leistung bereitgestellt, jedoch wird nicht mehr Strom verbraucht, als unbedingt nötig ist. Eine weitere Auffälligkeit würde die Ballung von benötigten Ressourcen darstellen. Im Beispiel einer Schule wäre dies die Verwendung von besonders ressourcenintensiven Arbeiten in einem bestimmten Raum. Die Lösung dazu wird noch in dem Abschnitt [4.7.2] gesondert behandelt.

5.3.2.2 Der Aufbau eines Berichts über den ökologischen Fußabdruck

Der Bericht ist in 5 große Kapitel eingeteilt. Er beginnt mit der sogenannten Executive Summary. Diese fasst so kurz wie möglich die wichtigsten Dinge aus dem Bericht zusammen und vermittelt so einen Eindruck über den gesamten Bericht. Außerdem sind darin schon die wichtigsten Ergebnisse abzulesen, was es ermöglicht auch ohne den gesamten Bericht zu lesen, damit zu arbeiten. Er ist die Zusammenfassung des Berichts. Dem folgt, nach dem Inhaltsverzeichnis, die Einleitung. Darin wird beschrieben um was es in diesem Bericht geht und um welche Firma oder Organisation es darin geht. Außerdem wird beschrieben was der Anlass des Berichts ist. Sofern diese vorhanden sind, wird auch auf früherer Berichte als Vergleich verwiesen. Gleich darauf folgen die so genannten Beziehungen. Darin wird beschrieben, woher die Werte stammen und welche Teile des Firmeneigentums sowie der Geschäfte eine Rolle spielen. Danach werden die Werte präsentiert. Dies passiert übersichtshalber in Tabellen und Grafiken, welche erklärt werden. Darin werden auch endgültige Resultate und Ungewöhnlichkeiten angemerkt, welche für den Leser nützlich sein könnten. Eine Zusammenfassung liefert, als letzter Abschnitt, eine Schlussfolgerung, Bewertung sowie Verbesserungsvorschläge.

5.3.2.3 Beschreibung eines ökologischen Fußabdrucks

Der CO₂-Fußabdruck gibt die Größe der Fläche auf der Erde an, welche nötig ist die Menge an CO₂, für die der Fußabdruck berechnet wurde, zu binden. Um diese zu errechnen wird der gesamte Lifecycle eines Objektes betrachtet. Die Einheit dieser Fläche ist der globale Hektar. Dieser ist der durchschnittliche Hektar in Bezug auf die biologische Produktivität weltweit. Diese Methode der Darstellung eines Verbrauchs ist daher gut geeignet, da sie sehr leicht visualisierbar und umsetzbar ist. Die Verwendung einer Einheit erlaubt es außerdem verschiedenen Ergebnisse zu vergleichen. Dies ist jedoch zugleich auch eine große Schwäche da dies viele, teilweise auch nützliche Informationen, verliert. Daher ist es laut Experten notwendig neben der Fläche weitere Faktoren zu nennen, um ein relevantes Ergebnis zu erhalten. Ein weiterer Nachteil ist der Fakt, dass nicht alle Einflüsse auf die Natur dabei beachtet werden. Viele, wie zum Beispiel toxische sowie radioaktive Substanzen, werden bei dieser Methodik nicht beachtet. Dies führt dazu, dass sie verzerrte oder falsche Werte liefern kann.

5.3.2.4 Auswertung der Daten für den ökologischen Fußabdruck:

5.3.2.4.1 Ermitteln der Nutzzeiten der PCs

Die Zeiten wurden mithilfe einer GPO (Group Policy Object) festgestellt. Diese wurde auf sämtliche PCs in dem AD (Active Directory) der Schule angewendet. Die GPO gibt den PCs vor, dass sie das folgende Powershell-Script beim Start auszuführen haben.

```

#weist der Variablen netzlaufwerk den Pfad des Ordners auf dem Fileserver
zu

$netzlaufwerk = "\\10.0.0.3\sgreening\files\zeiten\"

#erstellt ein neues lokales Laufwerk welches auf den Pfad in der Variablen
netzlaufwerk verweist

New-SmbMapping -LocalPath x: -RemotePath $netzlaufwerk

#weist der Variablen local den Pfad des lokalen Speicherortes zu

$local = "C:\test\"

#überprüft ob der Pfad in der Variablen local bereits existiert
if (-Not (Test-Path $local)){
#falls dieser nicht existiert wird er erstellt
    mkdir $local
}

#weist der Variablen pfad den Pfad für die lokale Speicherdatei zu
$pfad = $local + $env:computername + ".txt"

#versucht den Code innerhalb der Klammern auszuführen
try{
#überprüft ob der Pfad in der Variablen netzlaufwerk bereits existiert
    if (-Not (Test-Path $netzlaufwerk)){
#falls dieser nicht existiert wird er erstellt
        mkdir $netzlaufwerk
    }
}

# der Code innerhalb der Klammer wird ausgeführt falls das Netzlaufwerk nicht
erreichbar ist oder nicht existiert

catch{
#schreibt eine Fehlermeldung in die lokale Speicherdatei
    "Netzlaufwerk nicht gefunden" | out-file $pfad -append
}

```

```

#weist der Variablen netzpfad den Pfad für die Speicherdatei auf dem
Fileserver zu

$netzpfad = $netzlaufwerk + $env:computername + ".txt"

#wiederholt den Code innerhalb der Klammern unbegrenzt oft
While ($true){

#schreibt das aktuelle Datum sowie die Uhrzeit in die lokale Speicherdatei

    Get-Date | out-file $pfad -append

#versucht den Code innerhalb der Klammern auszuführen

    try{

#schreibt das aktuelle Datum sowie die Uhrzeit in die Speicherdatei auf dem
Fileserver

        Get-Date | out-file $netzpfad -append

    }

# der Code innerhalb der Klammer wird ausgeführt falls das Netzlaufwerk nicht
erreichbar ist oder nicht existiert

    catch{

        #schreibt eine Fehlermeldung in die lokale Speicherdatei

        "Netzlaufwerk nicht gefunden" | out-file $pfad -append

    }

#wartet 30 Sekunden bis der nächste Durchgang der Schleife beginnt

    Start-Sleep -s 30

}

```

Dieses Script schreibt alle 30 Sekunden einen Eintrag in eine lokale Datei. Dieser Eintrag besteht aus dem aktuellen Datum sowie der Uhrzeit. Um diese Dateien zentral auswerten zu können, wird auch ein Fileserver verwendet. Falls dieser erreichbar ist, werden auch dort die Dateien erstellt und beschrieben.

Eine Beispielzeile aus der Datei welche davon geliefert wird hat diese Form:

Mittwoch, 21. März 2018 20:10:30

5.3.2.4.2 Weiterverarbeitung der Daten

Die Daten, welche verwendet wurden um die Ergebnisse zu berechnen, bestehen aus 2 Quellen. Die erste Quelle ist die Strommessung der PCs der Stichproben aus den Gerätegruppen (siehe 4.7.1). Weiters noch die Dateien, welche von der GPO auf dem Fileserver

gespeichert wurden (siehe 5.3.2.4.1). Um diese Dateien auszuwerten wurde ein Pythonscript geschrieben. Diese gibt die Anzahl an Minuten zurück die der PC, von welchem die Datei ausgewählt wurde, genutzt wurde. Der Code dieses Scriptes wird hier gezeigt:

```
def    einlesen(pfad,    anfangtag=0,    anfangmonat=0,    anfangjahr=0,
anfangsstunde=0, anfangsminuten=0, endetag=0, endmonat=0,
        endjahr=0,
        endstunde=0, endminuten=0):
    """
    :param pfad: der Pfad zu der Datei mit den Zeiten der GPO
    :param anfangtag: der Tag ab dem die Zeit gerechnet werden soll
    :param anfangmonat: das Monat ab dem die Zeit gerechnet werden soll
    :param anfangjahr: das Jahr ab dem die Zeit gerechnet werden soll
    :param anfangsstunde: die Stunde ab der die Zeit gerechnet werden soll
    :param anfangsminuten: die Minute ab der die Zeit gerechnet werden soll
    :param endetag: der Tag bis zu dem die Zeit berechnet werden soll
    :param endmonat: das Monat bis zu dem die Zeit berechnet werden soll
    :param endjahr: das Jahr bis zu dem die Zeit berechnet werden soll
    :param endstunde: die Stunde bis zu der die Zeit berechnet werden soll
    :param endminuten: die Minute bis zu der die Zeit berechnet werden soll
    :return:
    """

    # in lastMiinuten wird die derzeitige anzahl an Minuten mitgezählt
    lastMinuten = 0

    # öffnet das File welches durch den Pfad in der Variable pfad beschrieben
    wird und weist diesen Verweis der

    # Variable file zu welche sodann als Fileobject angesprochen werden kann,
    with open(pfad, 'r', encoding='utf-16') as file:

        # erstellt die Variable add und weist ihr den Wert True zu
        add = True
```

```
# erstellt die Variable monate welche eine Liste ist und fügt ihr
die Monate als String hinzu
```

```
monate = list()
monate.append("Jänner")
monate.append("Februar")
monate.append("März")
monate.append("April")
monate.append("Mai")
monate.append("Juni")
monate.append("Juli")
monate.append("August")
monate.append("September")
monate.append("Oktober")
monate.append("November")
monate.append("Dezember")
```

```
# überprüft ob der Inhalt der Variablen anfangtag dem Wert 0
entspricht.
```

```
if anfangtag != 0:
    # setzt den Wert der Variablen add auf False
    add = False
# initialisiert die Variable lastm mit dem Wert 0
lastm = 0
# initialisiert die Variable lastd mit dem Wert 0
lastd = 0
```

```
# initialisiert die Variable lines und weist ihr die Zeilen der Datei
zu welches durch die Variable file
```

```
# repräsentiert wird.
lines = file.readlines()
```

```
# führt die Befehle innerhalb der Schleife für jede Zeile welche in
lines gespeichert ist aus.
```



```

for line in lines:

    # überprüft ob die Zeile länger als 20 Zeichen ist.

    if len(line) > 20:

        # initialisiert die Variable tag mit dem Tag welcher aus der
        Zeile ausgelesen wird.

        tag = int(line.split(' ')[1][: -1])

        # initialisiert die Variable monat mit dem Monat welches aus
        der Zeile ausgelesen wird und dann

        # mithilfe des Index der Variable monate in eine Zahl
        umgewandelt wird.

        monat = monate.index(line.split(' ')[2]) + 1

        # initialisiert die Variable Jahr mit dem jahr welches aus
        der Zeile ausgelesen wird.

        jahr = int(line.split(' ')[3])

        # initialisiert die Variable stunde mit der Stunde welche
        aus der Zeile ausgelesen wird.

        stunde = int(line.split(' ')[4].split(':')[0])

        # initialisiert die Variable minuten mit dem minuten welcher
        aus der Zeile ausgelesen wird.

        minuten = int(line.split(' ')[4].split(':')[1])

        # überprüft ob das der Wert in der Variable jahr größer ist
        als der Wert der Variable anfangsjahr

        if jahr > anfangsjahr:

            # setzt den Wert der Variablen add auf True

            add = True

        # überprüft ob das der Wert in der Variable jahr gleich groß
        wie der Wert der Variable Anfangsjahr

        if jahr == anfangsjahr:

            # überprüft ob das der Wert in der Variable monat größer
            ist als der Wert der Variable anfangmonat

            if monat > anfangmonat:

                # setzt den Wert der Variablen add auf True

```

```

        add = True

        # überprüft ob das der Wert in der Variable monat gleich
        groß wie der Wert der Variable anfangmonat

        if monat == anfangmonat:

            # überprüft ob das der Wert in der Variable tag
            größer ist als der Wert der Variable anfangtag

            if tag > anfangtag:

                # setzt den Wert der Variablen add auf True

                add = True

            # überprüft ob das der Wert in der Variable tag
            gleich groß wie der Wert der Variable anfangtag

            if tag == anfangtag:

                # überprüft ob das der Wert in der Variable
                stunde größer ist als der Wert der Variable

                # anfangsstunde

                if stunde > anfangsstunde:

                    # setzt den Wert der Variablen add auf True

                    add = True

                # überprüft ob das der Wert in der Variable
                stunde gleich groß wie der Wert der Variable

                # anfangsstunde

                if stunde == anfangsstunde:

                    # überprüft ob das der Wert in der Variable
                    minuten größer oder gleich groß ist wie

                    # der Wert der Variable anfangsminuten

                    if minuten >= anfangsminuten:

                        # setzt den Wert der Variablen add auf
                        True

                        add = True

            # überprüft ob der Inhalt der Variablen endjahr nicht dem
            Wert 0 entspricht.

            if endjahr != 0:

```

```

        # überprüft ob das der Wert in der Variable jahr kleiner
ist als der Wert der Variable endjahr

        if jahr > endjahr:

            # setzt den Wert der Variablen add auf False

            add = False

        # überprüft ob das der Wert in der Variable jahr gleich
groß wie der Wert der Variable endjahr

        if jahr == endjahr:

            # überprüft ob das der Wert in der Variable monat
kleiner ist als der Wert der Variable endmonat

            if monat > endmonat:

                # setzt den Wert der Variablen add auf False

                add = False

            # überprüft ob das der Wert in der Variable monat
gleich groß wie der Wert der Variable endmonat

            if monat == endmonat:

                # überprüft ob das der Wert in der Variable tag
kleiner ist als der Wert der Variable

                # endetag

                if tag > endetag:

                    # setzt den Wert der Variablen add auf False

                    add = False

                # überprüft ob das der Wert in der Variable tag
gleich groß wie der Wert der Variable

                # endetag

                if tag == endetag:

                    # überprüft ob das der Wert in der Variable
stunde kleiner ist als der Wert der

                    # Variable endstunde

                    if stunde > endstunde:

                        # setzt den Wert der Variablen add auf
False

```

```

        add = False

        # überprüft ob das der Wert in der Variable
stunde gleich groß wie der Wert der

        # Variable endstunde

        if stunde == endstunde:

            # überprüft ob das der Wert in der
Variable minuten kleiner gleich ist wie der

            # Wert der Variable endminuten

            if minuten >= endminuten:

                # setzt den Wert der Variablen add
auf False

                add = False

            # überprüft ob der Wert der Variablen add True entspricht
            if add:

                # überprüft ob der Wert der Variablen tag dem Wert der
Variablen lastd plus Eins entspricht. Dies

                # entspricht der Überprüfung ob ein Tageswechsel
stattgefunden hat.

                if tag == lastd + 1:

                    # überprüft ob die Differenz zwischen dem Zeitpunkt
dieser Zeile und der vorherigen nicht

                    # größer als 7,5 Minuten ist

                    if (int(stunde) * 60 + int(minuten) + 1440) - lastm
< 450:

                        # addiert zu der Variablen lastMinuten 5 hinzu

                        lastMinuten += 5

                    # wird ausgeführt falls die Überprüfung, ob der Wert der
Variablen tag dem Wert der Variablen

                    # lastd plus Eins entspricht fehlschlägt
                else:

                    # überprüft ob die Differenz zwischen dem Zeitpunkt
dieser Zeile und der vorherigen nicht

```

```

        # größer als 7,5 Minuten ist

        if (int(stunde) * 60 + int(minuten)) - lastm < 450:

            # addiert zu der Variablen lastMinuten 5 hinzu

            lastMinuten += 5

        # setzt den Wert der Variablen lastm auf den neuen
        derzeitigen Zeitpunkt

        lastm = stunde * 60 + minuten

        # setzt den Wert der Variablen lastd auf den derzeitigen Tag

        lastd = tag

        # gibt das Ergebnis formatiert aus und gibt das File an aus welchem
        dies ausgelesen wurde.

        print("{0:<20s}:                                     {1:7d}
        Minuten".format(pfad.split("\\")[len(pfad.split("\\")) - 1].split('.')[0],
        lastMinuten))

```

```

einlesen(r'C:\Users\Dominik\Desktop\gerätemessungen\Lehrer_PC-ME.txt', 12,
3, 2018, 15, 28, 19, 3, 2018, 17, 55)

```

Eine Beispielausgabe diese Scriptes sieht wie folgt aus.

```

Lehrer_PC-ME           :      2535 Minuten

```

Mithilfe dieser Informationen wurde nun die Tabelle erstellt welche sämtliche benötigten Werte enthält. Zu diesen Werten zählen

- der Gerätetyp
- der Verbrauch eines einzelnen Gerätes in kW in der gemessenen Zeit
- die Anzahl der Minuten die das Gerät in dieser Zeit aktiv war
- der Durchschnittsverbrauch pro Stunde in kWh
- die Kosten pro Monat in Euro
- das bei der Herstellung des Stromes freigesetzte CO₂ pro Monat in Kilogramm
- der Verbrauch der Gerätegruppe pro Monat in kWh
- die Kosten der Gerätegruppe pro Monat
- sowie das bei der Herstellung des Stromes freigesetzte CO₂ der Gerätegruppe pro Monat in Kilogramm

Die vollständige Tabelle mit den Ergebnissen ist auf der beigelegten CD zu finden. Die Datei liegt in dem Ordner ökologischer_Fußabdruck.

6 Empfehlungen und Verbesserungen

6.1 Situation vor dem Projekt

Es gab keine direkten Bestrebungen gezielt den Fußabdruck der Schule zu verbessern sowie den Stromverbrauch durch Umstrukturierungen und Maßnahmen zu senken.

6.2 Beschreibung der „Quick Wins“, Verbesserungsmöglichkeiten + Vorschläge

6.2.1 Mögliche längerfristige Maßnahmen

6.2.1.1 Automatisiertes Herunterfahren der PCs

Der erste Lösungsvorschlag wäre ein automatisches Abschalten der nicht benötigten IT-Ressourcen zu implementieren. Dies würde zum Beispiel implizieren, dass die PCs in den EDV-Räumen um 18:30 heruntergefahren werden, falls dies nicht von einem Schüler abgebrochen wird. Ein zweiter Versuch würde sodann ein paar Stunden später erfolgen. Dadurch könnte es nicht passieren, dass Strom durch ein versäumtes Herunterfahren verbraucht wird.

Ein weiterer Einsatzpunkt sind die Access-Points. In den Ferien und auch am Abend ist es nicht nötig, dass diese laufen. An Tagen und Uhrzeiten, zu denen weniger als 50 Prozent der Schüler anwesend sind, wäre es sinnvoll nur 1 WLAN in Betrieb zu haben. Dies würde weniger Leistung erfordern und würde dennoch die benötigte Bandbreite zur Verfügung stellen.

6.2.1.2 Implementierung von Thin-Clients

Der zweite Lösungsvorschlag beschäftigt sich mit der Implementierung von Thin-Clients. Dies sind PCs mit verringerter Leistung und Hardware. Sie dienen dazu per Fernzugriff die Ressourcen eines Servers zu verwenden und als Interface für den Benutzer zu dienen. Da weniger Rohstoffe aufgewendet werden diese zu bauen und sie auch weniger Energie verbrauchen, sind sie wesentlich umweltschonender als Fat-Clients, welche herkömmliche PCs darstellen. Ein sinnvoller Einsatz wäre in den Lehrerzimmern sowie den Werkstätten. Die PCs in diesen Räumen werden nicht stark belastet und sind somit überdimensioniert.

Ein einzelner virtualisierter Server würde die Arbeitsaufgaben ebenfalls bearbeiten können. Dieser Server würde in der Produktivumgebung auf einem der Server virtualisiert werden und im Schulnetz mittels einer GPO bekanntgegeben werden. Dies ermöglicht es Änderungen schnell zu verteilen und dennoch sicherzustellen, dass nicht zu viel Datenverkehr durch die Synchronisation erzeugt wird.

6.2.1.3 Umstrukturierung der Ressourcen

Die dritte Methode um den Fußabdruck zu verbessern, ist die sinnvollere Aufteilung der Ressourcen. Dazu sollten die PCs je nach Leistung den Räumen effizient zugeordnet werden. Ein Beispiel dafür wäre es jene Geräte die leistungsstarken Grafikkarten besitzen dem Medientechniklabor sowie den EDV-Saal der Medientechnikabteilung zuzuordnen. Währenddessen würden in den anderen Räumen auch Thin-Clients oder schwächere PCs, welche auf Stromersparnis und Umweltfreundlichkeit ausgelegt sind, ausreichen. Dadurch würden der Stromverbrauch sowie die Anschaffungskosten deutlich sinken.

7 Abbildungsverzeichnis

Abbildung 3-1 Schematische Darstellung der Position der DHT 22 und MQ-2 Sensoren.....	25
Abbildung 3-2 Temperaturverlauf Produktivrack.....	26
Abbildung 3-3 Temperaturverlauf Raum Mitte	26
Abbildung 3-4 Temperaturverlauf Raum Fenster	27
Abbildung 3-5 Beispiel einer Kommunikation zwischen einem FPGA als Master und einem DS2432 Slave	29
Abbildung 3-6 Aktiver Modus ohne zusätzliche Spannungsversorgung.....	30
Abbildung 3-7 Schaltplan aktiver Modus ohne zusätzliche Spannungsversorgung	30
Abbildung 3-8 Blockschaltbild aktiver Modus mit zusätzlicher Spannungsversorgung	31
Abbildung 3-9 Schaltplan aktiver Modus mit zusätzlicher Spannungsversorgung	31
Abbildung 3-10 Blockschaltbild „parasite“ Modus	32
Abbildung 3-11 Schaltplan „parasite“ Modus	32
Abbildung 3-12 Versuchsaufbau 1-Wire Bus (Schaltplan siehe Anhang 10.2.1.1)	34
Abbildung 3-13 Positionierung der DS1820 1-Wire Sensoren im Serverraum 926.....	35
Abbildung 3-14 Positionierung der DS1820 1-Wire Sensoren im Serverraum 079.....	36
Abbildung 3-15 Positionierung der DHT 11 Sensoren im Serverraum 926	37
Abbildung 3-16 Positionierung des M26422 Sensors im Serverraum 926.....	38
Abbildung 3-17 Positionierung der M26422 Sensoren im Serverraum 078.....	39
Abbildung 3-18 Verfügbare Kommandos.....	40
Abbildung 3-19 Screenshot des Status Kommandos	42
Abbildung 3-20 Screenshot der Index Seite	45
Abbildung 3-21 Notifications Panel (spezieller Tab siehe 3.6.4.2.4)	46
Abbildung 3-22 Aktueller Kühlkreislauf Serverraum 926	48
Abbildung 3-23 Sensor Rack3 oben (bei den Produktivservern)	48
Abbildung 3-24 Neuer Luftströmungskreislauf.....	49
Abbildung 3-25 Luftfluss nach der Installation der direkten Zuleitungen	50
Abbildung 3-26 Luftstrom bei genau ausgerichteten Ventilatoren.....	51
Abbildung 3-27 Kalt- / Warmgangkonzept Vogelperspektive	52
Abbildung 3-28 Vogelperspektive Erweiterung Außenluftkühlkonzept.....	52
Abbildung 3-29 Schematische Darstellung der Hardware	53
Abbildung 3-30 ER Modell der Datenbank „serverraum_temperaturueberwachung“	57
Abbildung 3-31 ER Modell der Datenbank „messages“	58
Abbildung 3-32 „Navbar“ eingeklappt und ausgeklappt.....	89
Abbildung 3-33 Screenshot Beispielausgabe der „current.php“	90
Abbildung 3-34 Screenshot Beispielausgabe „currentvalues.php“	90
Abbildung 3-35 Screenshot des Notifications Tab.....	91
Abbildung 3-36 Screenshot des „Flot Charts“	92
Abbildung 3-37 Ausgabe der Seite ohne Bootstrap.....	93
Abbildung 4-1 Beispiel einer Wechselstromkurve mit eingezeichneten Parametern.....	100
Abbildung 4-2 Beispiel einer Sinusförmigen Wechselstromkurve mit eingezeichneten Parametern.....	101
Abbildung 4-3 Leistung bei ohmscher Last	103
Abbildung 4-4 Leistung bei ohmscher-induktiver Last	103
Abbildung 4-5 Leistung bei kapazitiver Last.....	104

Abbildung 4-6 Schematische Darstellung eines Transformators.....	106
Abbildung 4-7 Schaltbild eines idealen Transformators	106
Abbildung 4-8 Schematischer Aufbau einer Stromzange.....	107
Abbildung 4-9 Kleine Ansicht der Testergebnisse	109
Abbildung 4-10 Aufbau und Anschluss einer BCM2 Einheit.....	115
Abbildung 4-11 Foto von Anschluss von Verbindungskabel 1 und Cat5-Kabel Richtung PMC..	116
Abbildung 4-12 Foto von Spannungsversorgung und Spannungsmessung	117
Abbildung 4-13 Anzeige der Circuit-Details am LC-Display des PMCs.....	118
Abbildung 4-14 Stromverbrauch von EDV4 an Phase 1 (Mo-Fr).....	130
Abbildung 4-15 Stromverbrauch von EDV4 an Phase 2 (Mo-Fr)	130
Abbildung 4-16 Stromverbrauch von EDV4 an Phase 3 (Mo-Fr).....	131
Abbildung 10-1 Schaltplan Versuchsaufbau	161

8 Tabellenverzeichnis

Tabelle 3-1 Legende	25
Tabelle 3-2 Name und Position der Sensoren.....	25
Tabelle 3-3 Höhenangaben vom Boden aus / Legende zum Raumplan	35
Tabelle 3-4 Auszug aus der Tabelle „sensor“ der Datenbank „serverraum_temperaturueberwachung“	35
Tabelle 3-5 Höhenangaben vom Boden aus / Legende zum Raumplan	36
Tabelle 3-6 Auszug aus der Tabelle „sensor“ der Datenbank „serverraum_temperaturueberwachung“	37
Tabelle 3-7 Höhenangaben vom Boden aus / Legende zum Raumplan	37
Tabelle 3-8 Auszug aus der Tabelle „sensor“ der Datenbank „serverraum_temperaturueberwachung“	38
Tabelle 3-9 Höhenangaben vom Boden aus / Legende zum Raumplan	38
Tabelle 3-10 Auszug aus der Tabelle „sensor“ der Datenbank „serverraum_temperaturueberwachung“	39
Tabelle 3-11 Höhenangaben vom Boden aus / Legende zum Raumplan.....	39
Tabelle 3-12 Auszug aus der Tabelle „sensor“ der Datenbank „serverraum_temperaturueberwachung“	39
Tabelle 3-13 Erklärung des Hitzeerzeugungsindexes.....	43
Tabelle 3-14 Reihung der Ressourcen nach Ausschaltzeitpunkt mit Priorität.....	43
Tabelle 4-1 Gruppen der PCs	98
Tabelle 4-2 Gruppen der Netzwerkkomponenten.....	99
Tabelle 4-3 Gruppen andere Gerätetypen.....	99
Tabelle 4-4 Messdaten von Gruppe PC-A1.....	122
Tabelle 4-5 Messdaten von PC-A2	122
Tabelle 4-6 Messdaten von Gruppe PC-D1.....	122
Tabelle 4-7 Messdaten von Gruppen PC-H2.....	123
Tabelle 4-8 Messdaten von Gruppe PC-H3	123
Tabelle 4-9 Messdaten von Gruppe PC-H4	123
Tabelle 4-10 Messdaten von Gruppe PC-H5.....	123
Tabelle 4-11 Messdaten von Gruppe PC-H7.....	124
Tabelle 4-12 Messdaten von Gruppe AP1	125
Tabelle 4-13 Messdaten von Gruppe AP4	125
Tabelle 4-14 Messdaten von Gruppe AP5	126
Tabelle 4-15 Vergleich der Messergebnisse von APs	126
Tabelle 4-16 Messdaten von Gruppe FW3	126
Tabelle 4-17 Messdaten von Gruppe R1	127
Tabelle 4-18 Messdaten von Gruppe R2	127
Tabelle 4-19 Messdaten von Gruppe R3	127
Tabelle 4-20 Messdaten von Gruppe R4	127
Tabelle 4-21 Messdaten von Gruppe R5	128
Tabelle 4-22 Messdaten von Gruppe SW3	128
Tabelle 4-23 Messdaten von Gruppe SW5	128
Tabelle 4-24 Messdaten von Gruppe SE2.....	129
Tabelle 4-25 Mittelwerte der PCs EDV4.....	132

Tabelle 4-26 Hochrechnung der Einzelmessungen EDV4.....	132
Tabelle 4-27 Hochrechnung des Stromverbrauches der Übungsracks über ein Wochenende (48 Stunden).....	134

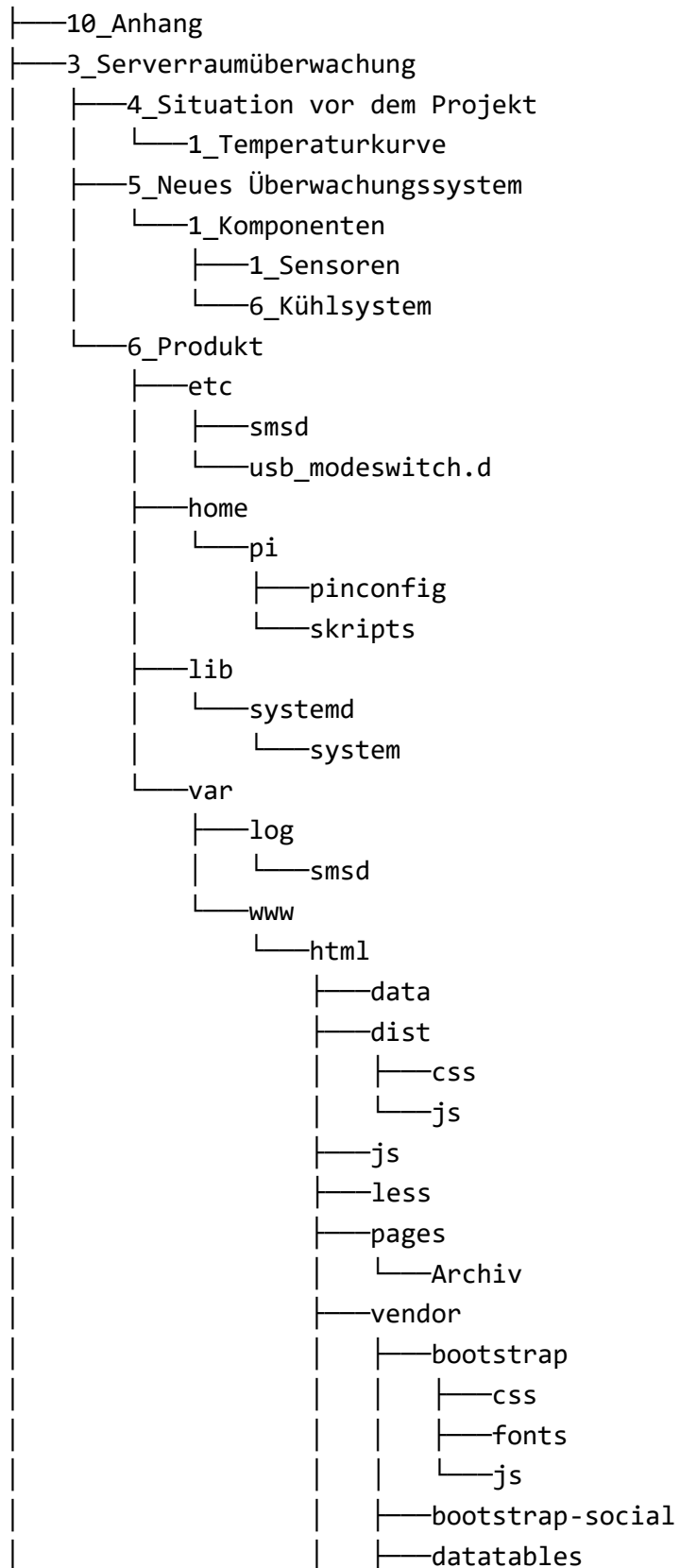
9 Literaturverzeichnis

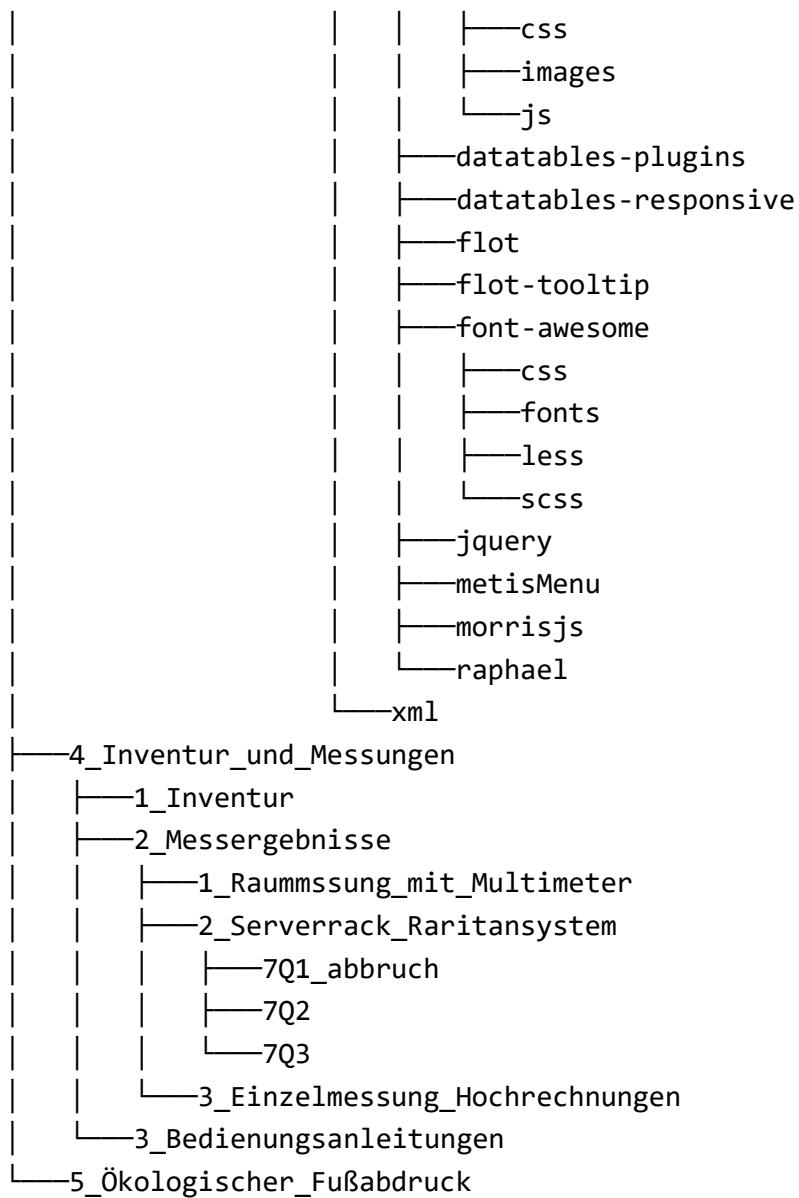
- (Hrsg.), J. M. (2010). *GREEN Computing & Sustainability*. unbekannt: dpunkt.verlag.
- 1 Wire Wikipedia. (11. 02 2018). Von <https://de.wikipedia.org/wiki/1-Wire> abgerufen
- Bertold, P. (2011). *CODE of CONDUCT on Energy Efficiency of AC Uninterruptible Power Systems*. Ispra: European Comission.
- CDW-G. (unbekannt). *Implementing Green I.T.* unbekannt: CDW-G.
- co2logic. (2011). *Carbon Footprint Report*. unbekannt: European Investment Bank.
- Deimel, F., & Hasnzagl, A. (2013). *Grundlagen der Elektrotechnik* (Bd. 2). Linz: VERITAS-VERLAG.
- DEME. (2014). *Carbon footprint report 2013*. Belgien, Niederlande: DEME.
- Erek, K., Kolbe, L. M., Schmidt, N.-H., & Zarnekow, R. (2009). *Studie: Nachhaltigkeit und Green IT in IT-Organisationen*. Göttingen/Berlin: unbekannt.
- European Investment Bank. (unbekannt. unbekannt 2016). Von European Investment Bank 2016 <http://www.eib.org>: <http://www.eib.org/infocentre/publications/all/carbon-footprint-report-2016.htm> abgerufen
- geSCI. (2010). *GREEN ICT GUIDE*. Global e-Schools and Communities Initiative: geSCI.
- global actxon plan. (2007). *An Inefficient Truth*. global actxon plan.
- global actxon plan. (2007). *An Inefficient Truth*. global actxon plan.
- Gold Fields. (unbekannt. 5 2010). Von Gold Fields May 2010 www.goldfields.com: <https://www.goldfields.com/pdf/sustainability/sustainability-reporting/carbon-submissions/cdp-submission/cdp-submission-2009.pdf> abgerufen
- greenpeace. (3 2010). 2010 www.greenpeace.com. Von <https://www.greenpeace.org/archive-international/Global/international/planet-2/report/2010/3/make-it-green-cloud-computing.pdf> abgerufen
- <http://www.chemie.de>. (kein Datum). Abgerufen am 2. 3 2018 von <http://www.chemie.de/lexikon/Kohlenstoffdioxid.html>
- Loser, F., & Zarnekow, R. (2015). *Nachhaltiges IT-Management*. dpunkt.verlag GmbH.
- Mark Acton, P. B. (2017). *2017 Best Practice Guidelines for the EU Code of Conduct on Data Centre Energy Efficiency*. Joint Research Center: European Comission.
- McSweeney, R. (25. 2 2015). www.carbonbrief.org. Abgerufen am 1. 4 2018 von <https://www.carbonbrief.org/new-study-directly-measures-greenhouse-effect-at-earths-surface>
- Mobbs, P. (2012). *A practical guide to sustainable IT*. South Africa: Karen Banks and Flavia Fascend.

- Paolo, B. (31. 3 2017). *EU Code of Conduct on Energy Consumption of Broadband Equipment*. Von ec.europa.eu: <https://ec.europa.eu/jrc/en/publication/eu-code-conduct-energy-consumption-broadband-equipment-version-6> abgerufen
- Raritan Inc. (22. Februar 2018). *Raritan Product Support BCM2 Power Meter Quick Setup Guide*. Abgerufen am 31. März 2018 von http://cdn.raritan.com/download/bcm2/version-3.4.0/PowerMeter_QSG_1C_3.4.0.pdf
- Schnabel, P. (kein Datum). *Elektronik Kompendium*. (P. Schnabel, Herausgeber) Abgerufen am 11. 02 2018 von <https://www.elektronik-kompendium.de/sites/grd/0208071.htm>
- Stiftung Warentest. (12. Juni 2009). *Strommessgeräte: Nur eins ist gut (Artikel)*. Abgerufen am 26. März 2018 von <https://www.test.de/Strommessgeraete-Nur-eins-ist-gut-1781202-0/>
- Tetz, M. (2014). *GREEN-IT*. Wien: Kirchliche Pädagogische Hochschule Wien/Krems.
- Wake, P. (13. 6 2017). Abgerufen am 21. 2 2018 von Pamela Wake Jun,13 2017 www.livestrong.com: <https://www.livestrong.com/article/160720-carbon-footprint-examples/>

10 Anhang

10.1 Ordner Struktur der beiliegenden CD





10.2 Dokumentation des Produktes

10.2.1 Schaltpläne

10.2.1.1 Versuchsaufbau

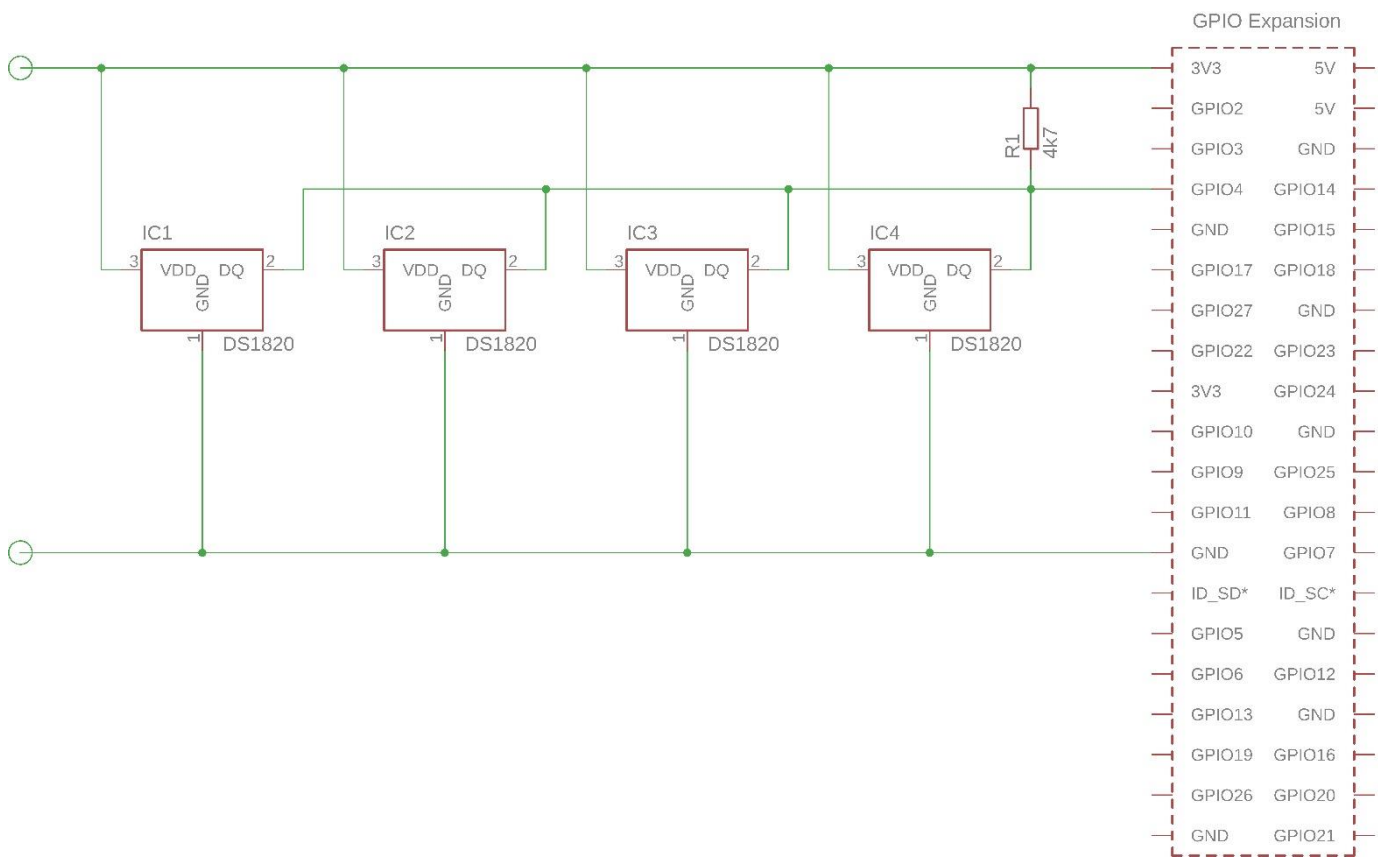


Abbildung 10-1
Schaltplan Versuchsaufbau

10.2.2 Stücklisten

Bezeichnung	Link	Anzahl	Ver- baut
DS18S20	https://www.conrad.at/de/temperatur-sensor-maxim-integrated-ds18s20-55-bis-125-c-to-92-radial-bedrahtet-176168.html	12	12
DHT11	https://www.conrad.at/de/ntc-temperatursensor-dht11-0-bis-50-c-radial-bedrahtet-1405544.html?sc.ref=Product%20Details	10	0
DHT22		10	3
MQ-2 Gassensor		8	2
Wassersensor M26422		3	0
Raspi 3B	https://www.conrad.at/de/raspberry-pi-3-model-b-1-gb-ohne-betriebssystem-1419716.html	2	1
Raspi 3		2	2
Netzteil	https://www.conrad.at/de/usb-ladegeraet-steckdose-voltcraft-sps-2100m-ausgangsstrom-max-2100-ma-1-x-usb-micro-usb-raspberry-pi-2-geeignet-1219951.html	2	1
Dual Netzteil 5V / 2A		1	1
Raspi Gehäuse	https://www.conrad.at/de/raspberry-pi-gehaeuse-aluminium-gehaeuse-b-2-b-3-b-1325789.html	2	
Kühlkörper Set	https://www.conrad.at/de/kuehlungs-kit-raspberry-pi-a-b-b-cubieboard-banana-pi-1267831.html	4	2
SD Karte	https://www.conrad.at/de/microsdhc-karte-32-gb-sandisk-ultra-class-10-uhs-i-inkl-sd-adapter-1381016.html	2	2
SD Karte 8 GB		1	1
SD Karte 16GB		1	1
Steckplatinen	https://www.conrad.at/de/steckplatine-verschiebbar-polzahl-gesamt-840-l-x-b-x-h-1727-x-645-x-85-mm-conrad-components-0165-40-4-27020-1-st-1268682.html	5	0

Widerstandset	https://www.conrad.at/de/metall-schicht-widerstand-sortiment-axial-be-drahtet-025-w-conrad-components-390-st-418714.html	1	1
---------------	---	---	---

10.3 Bericht des ökologischen Fußabdrucks