

Arbeitspaket AP 2.1.12 Messwertstreckendokumentation

Unterarbeitspaket: Messboxensystem

1 Einleitung

Auf den folgenden Seiten wird der Aufbau des Messboxsystems beschrieben und erläutert.

Hierbei handelt es sich um ein einfaches, eigenständiges, System welches einen kostengünstigen praxisnahen und offenen Weg der Messwernerfassung ermöglicht.

Das Messboxsystem nimmt, ähnlich zur Gebäudeleittechnik, Daten auf und visualisiert diese lokal für den Anwender. Die dazu notwendige Software ist bereits als „Open-Source“ frei zugänglich oder wurde im Rahmen von Veröffentlichungen frei zugänglich gemacht.

2 Installierte Programme

Auf dem Raspberry Pi (Wheezy) sind folgende Open Source Pakete (Programme) installiert:

- LAMP mit Apache und MySQL
- php5
- phpMySQL
- bootstrap

3 Ressourcen

Des Weiteren werden folgende Ressourcen genutzt:

- Ø Node.JS (Javascript Interpreter)
 - Informationen unter www.nodejs.org
- Ø Bootstrap (Javascript-Framework zur adaptiven Webentwicklung)
 - Informationen unter www.getbootstrap.com
- Ø Grunt (BuildTool um Prozesse bei der Entwicklung mittels Bootstrap zu automatisieren)
 - Informationen unter www.gruntjs.com

4 FrontEnd

Das FrontEnd ist als Web-Anwendung entwickelt. Die Dateien werden von einem Webserver an einen Browser übertragen.

Alle nötigen Ressourcen werden in der index.html Datei verlinkt. Diese liegen auf dem Raspberry unter dem Pfad „\var\www“

Ein weiteres wichtiges Element im Code des FrontEnds bildet das API. Dieses bildet die Schnittstelle zwischen Backend-Verarbeitung der Daten und dem FrontEnd. In dem Ordner „\var\www\bootstrap\API“ liegen.

Hierbei werden folgende Übertragungsmittel genutzt:

- Transmit_Data.php → Empfang der digitalen Messwert-Daten auf dem lokalen Server (Raspberry Pi)
- Transmit_Data_analog.php → Empfang der analogen Messwert-Daten auf dem lokalen Server (Raspberry Pi)

5 Python Skripte

Die notwendigen Python Skripte für den autarken Betrieb befinden sich in der Ordnerstruktur „\home\pi\pythonMySQL“.

- script.py → Skript zur Verarbeitung der Messwerte in die Datenbank auf dem lokalen Server (Raspberry Pi)
- startUMTS.sh → Skript zum Aufbau der UMTS-Verbindung
- stopUMTS.sh → Skript zum Schließen der UMTS-Verbindung

6 JavaScript Bibliotheken

Die notwendigen JavaScript Bibliotheken sind bei Unix-Betriebssystemen wie dem hier verwendeten Raspbian Wheezy mit Querverweisen in der Installationsanweisung hinterlegt.

Somit werden bei der Installation der einzelnen Komponenten aus Kapitel 2 alle notwendigen Bibliotheken nachgeladen.

7 Diagramme

Die Konfigurationen die zur Darstellung des Diagramms im Frontend benötigt werden, sind in JavaScript-Variablen gespeichert.

Diese sind im Pfad „\var\www\bootstrap\dist\js“ hinterlegt. Von den hier liegenden Skripten sind vorwiegend das options.js und das plot-functions.js wichtig.

Alle Optionen, welche zur Darstellung der Visualisierung nötig sind, befinden sich in der „options.js“-Datei. Die Funktionen die zur Zeichnung des Diagramms nötig sind, befinden sich in der „plot-functions.js“-Datei.

- plot: Beinhaltet das flot-Objekt. Das flot-Objekt ist für die Darstellung und Berechnung der eigentlichen Diagramme zuständig.
- plotData: Ein mehrdimensionales Javascript-Objekt, das alle Daten beinhaltet die benötigt werden um mittels flot.js ein Diagramm zu zeichnen. Diese Variable wird während der Laufzeit durch Funktionen modifiziert und ergänzt.
- plotDataRaw: Javascript-Objekt, das alle Rohdaten der Datenreihen beinhaltet. Bei der Initialisierung des Skripts sind keine Daten vorhanden, sodass sie mittels der Funktion getRawData() vom Backend mittels AJAX-Request geholt werden.

8 Sensor-API

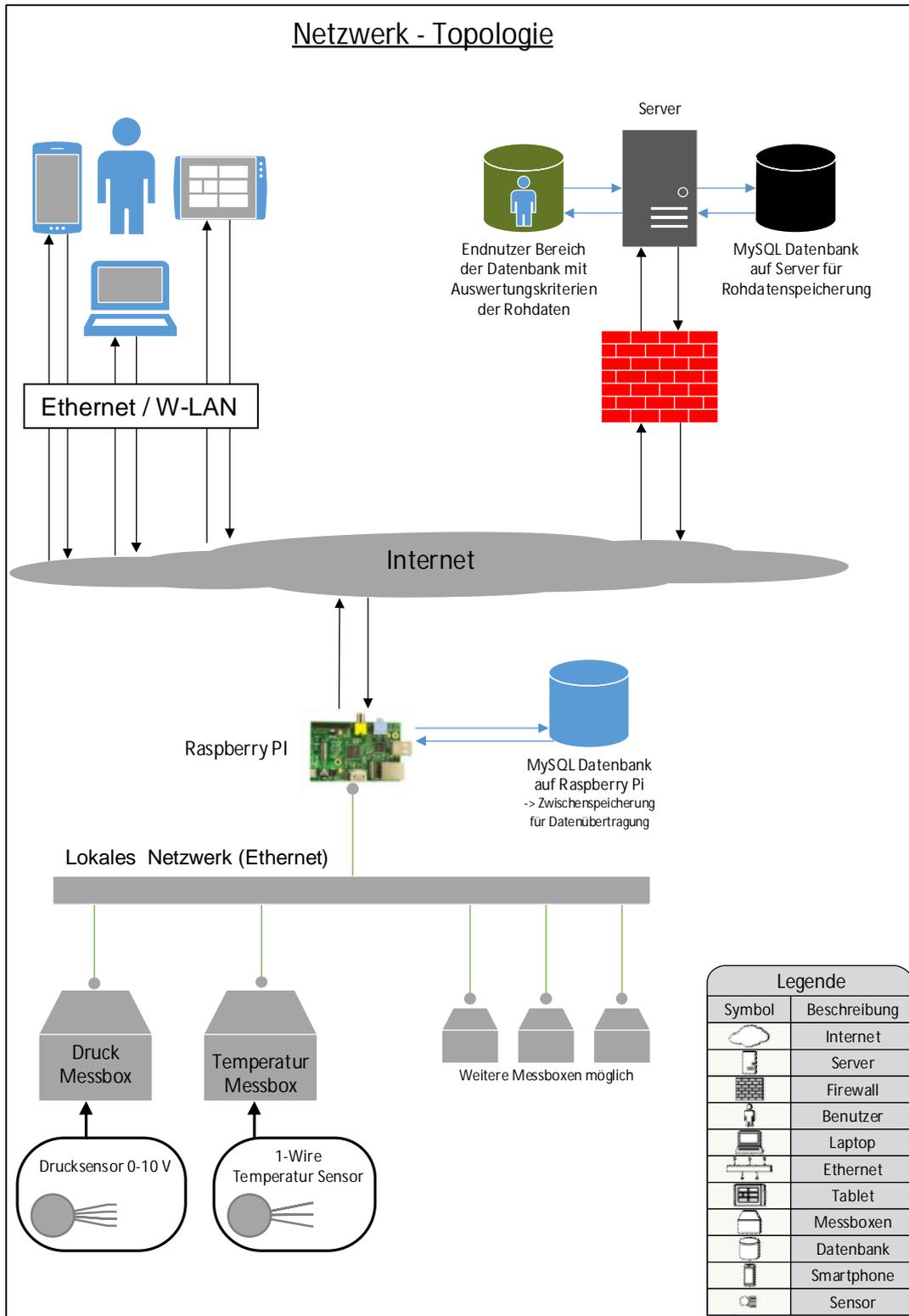
Die API liefert die zeitlich sortierten Datenreihen jedes Messpunktes zurück. Im Verzeichnis „\var\www\bootstrap\API\sensors\“ liegt für jede mögliche Datenreihe eine eigenständige Datei, die für die Aufarbeitung der einzelnen Datenreihe nötig ist.

9 Weiteres

Alle FrontEnd Funktionen und Funktionen die zur Datenverarbeitung gehören, befinden sich in app.js Datei im Pfad der „\var\www\bootstrap\dist\js“.

Zum Verbindungsaufbau mit dem UMTS-Netz wird das Package ppp0 genutzt.

10 Grafischer Aufbau



11 Veröffentlichungen:

Im Rahmen des Forschungsvorhabens wurden bereits detaillierte Vorträge zu der umgesetzten Messwertstrecke öffentlich gehalten. Ebenso bei einer renommierten Fachtagung im Bereich Regelungstechnik (Bälz-Seminar 2015, 12.09.2015 Hochschule Esslingen) unter dem Titel „PreComBo - Anlagenanalyse ohne Gewährleistungseingriff“.

Die Vortragspräsentation zur Messwertstreckendokumentation befindet sich in Kapitel 12.1.

12 Anlagen:

In der digitalen Anlage befinden sich folgende Dateien:

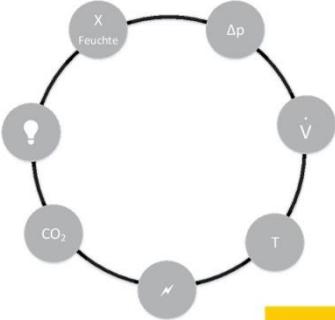
- Die erstellten Skripte für die Mikrokontroller in den Sensorboxen „Arduino UNO R.3“ in kompletter Form. Eine „arduino_sensor_analog.ino“ für analoge Sensoren und eine „arduino_sensor_digital.ino“ für digitale Sensoren. Dabei werden als analoge Sensoren die in der Regelungstechnik üblichen 0-10 V Signale verarbeitet. Als zu verarbeitende digitale Sensordaten werden OneWire DS18B20 verwendet.
- Partielles Datenträgerabbild für den Raspberry Pi v2 Typ B zum Erstellen des lokalen Datenspeichers für die Messboxen auf „Arduino“-Basis.

12.1 Vortrag Bälz-Seminar 2015 – 12.09.2015:




PreComBo

**Anlagenanalyse
ohne Gewährleistungseingriff**




B.Eng. Raphael Bär (TGA-S)

TEAMS WORK.

PRECOMBO – ANLAGENANALYSE OHNE GEWÄHRLEISTUNGSEINGRIFF

ZIEL DES SYSTEMS:

- Einen Möglichen Weg von der Messstelle zum Nutzer:


→


- Anforderungen an das System:
 - Open Source
 - Low Budget Lösung
 - Robuster, simpler Aufbau
 - Modularer Aufbau
 - Veröffentlichung

© Ed. Züblin AG, 11.09.2015
Seite 2

B.Eng. Raphael Bär – TGA-S



TEAMS WORK.



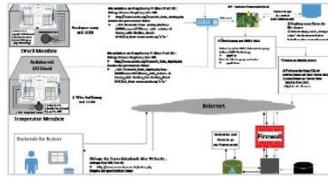
1 NICHTINVASIVE
MESSORTE



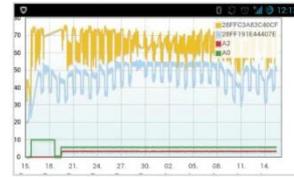
2 SENSORARTEN



3 AUFBAU MESSBOXEN



4 NETZWERKTOPOLOGIE



5 MOBILE NUTZUNG

© Ed. Züblin AG, 11.09.2015
Seite 3

B.Eng. Raphael Bär – TGA-S



1 NICHTINVASIVE
MESSORTE



NICHTINVASIVE MESSORTE

NICHTINVASIV

- Bedeutung Invasiv:
 - Eindringen
 - Hineingehen
 - Einbrechen

 - Folge:
 - Eine Messung ohne Eingriff in das System
 - Kein Eingriff in die Gewährleistung des Nachunternehmers

 - Ziel:
 - Messstellen ohne Systemkontakt
- und:
- Messstellen die „sowieso“ offen sind

NICHTINVASIVE MESSORTE

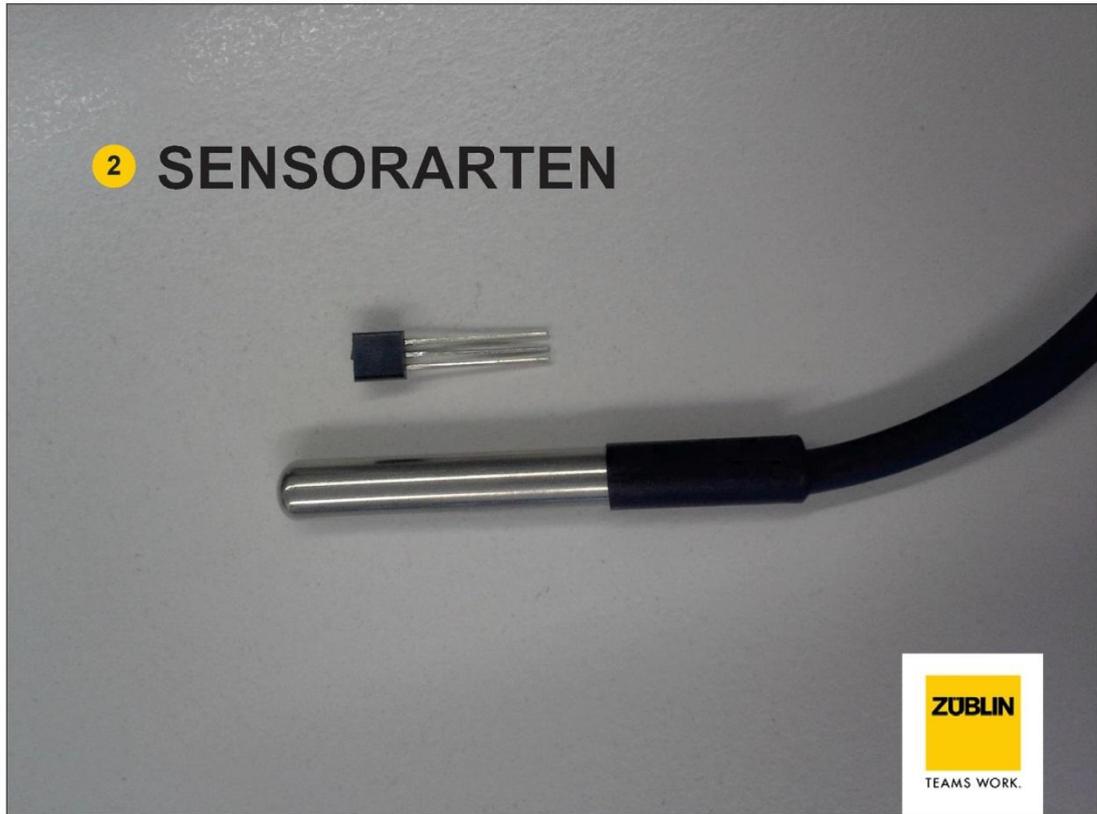


Verteileranschluss mit Entleerung und Absperrung



Stellantrieb Kieback&Peter MD200Y

2 SENSORARTEN



SENSORARTEN



**1-WIRE
TEMPERATURSENSOR**

- Klein (4 bzw. 6 mm)
- Edelstahl, vergossen
- Kostengünstig (~2 €)

Nachteil:

- Genauigkeit ± 0.5 K



DRUCKMESSZELLEN

- 1/8 "
- Edelstahl
- Kostengünstig (~ 35 €)

Nachteil:

- Signalausgabe im mV Bereich
- Keine Kompensation



PT100 / PT1000

- Klein (3 bis 6 mm)
- Langlebig
- Hohe Genauigkeit

Nachteil:

- Teuer (~15 €)



DRUCKMESSUMFORMER

- 1/4 " , 1/8 " , 1/2 "
- Edelstahl
- Fertig Kompensiert
- Signal 0-10 V , 4-20 mA

Nachteil:

- Teuer (~90 €)

Quelle PT1000: www.pollin.de Artikel 180 030

SENSORARTEN



Drucksensor und Kappenventil



Drucksensor mit Kappenventil

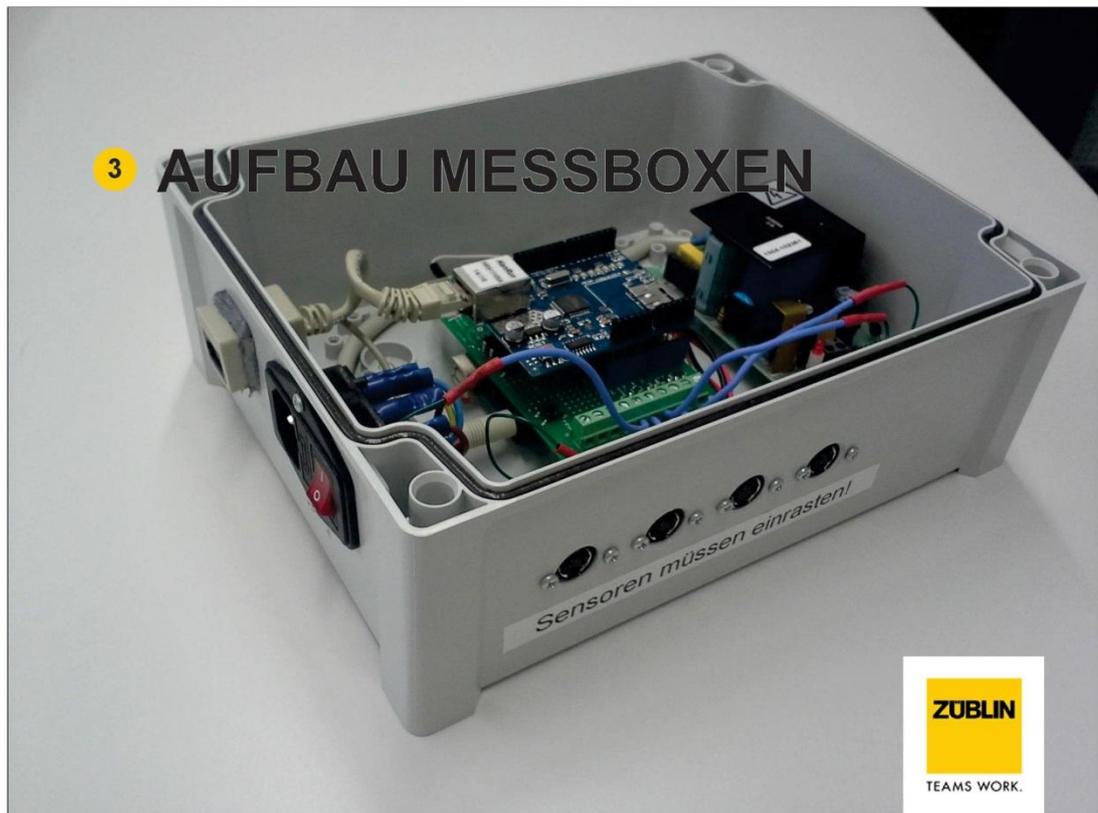
2

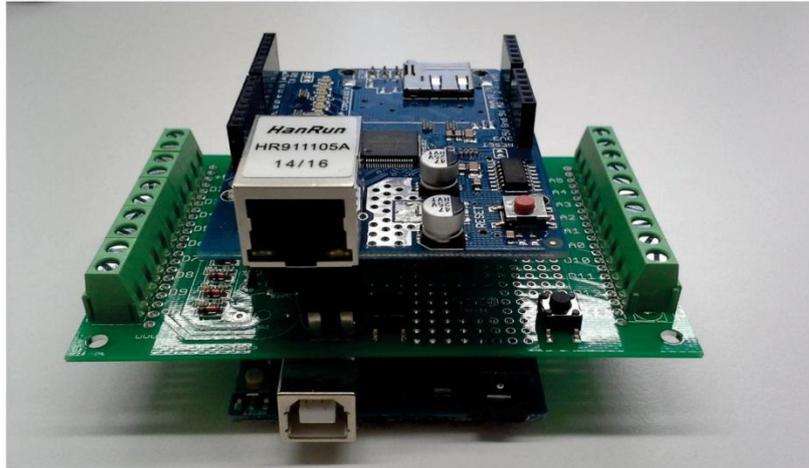
© Ed. Züblin AG, 11.09.2015
Seite 9

B.Eng. Raphael Bär – TGA-S

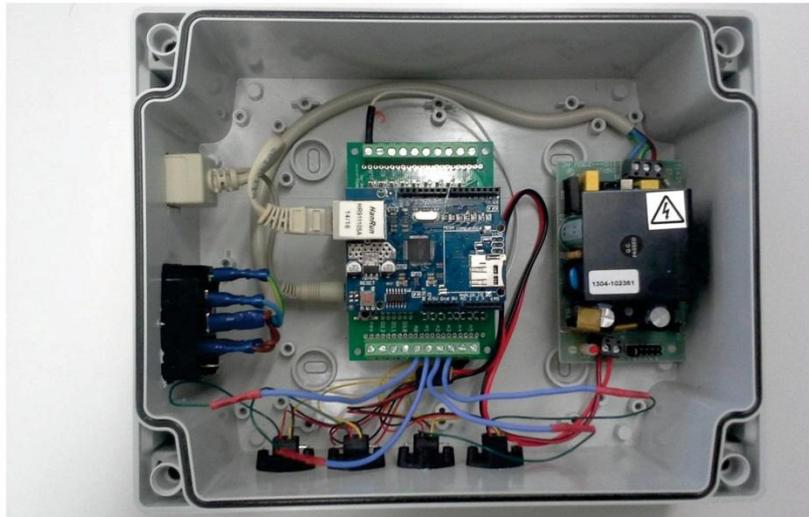


3 AUFBAU MESSBOXEN

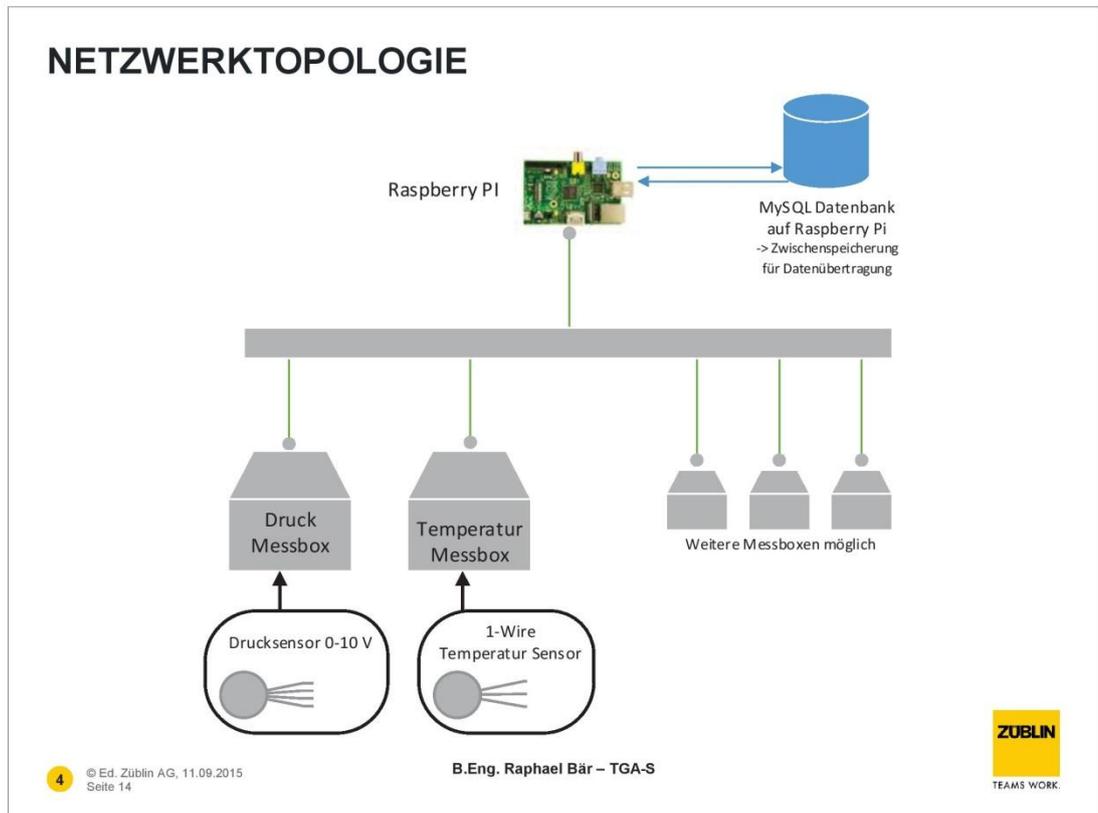
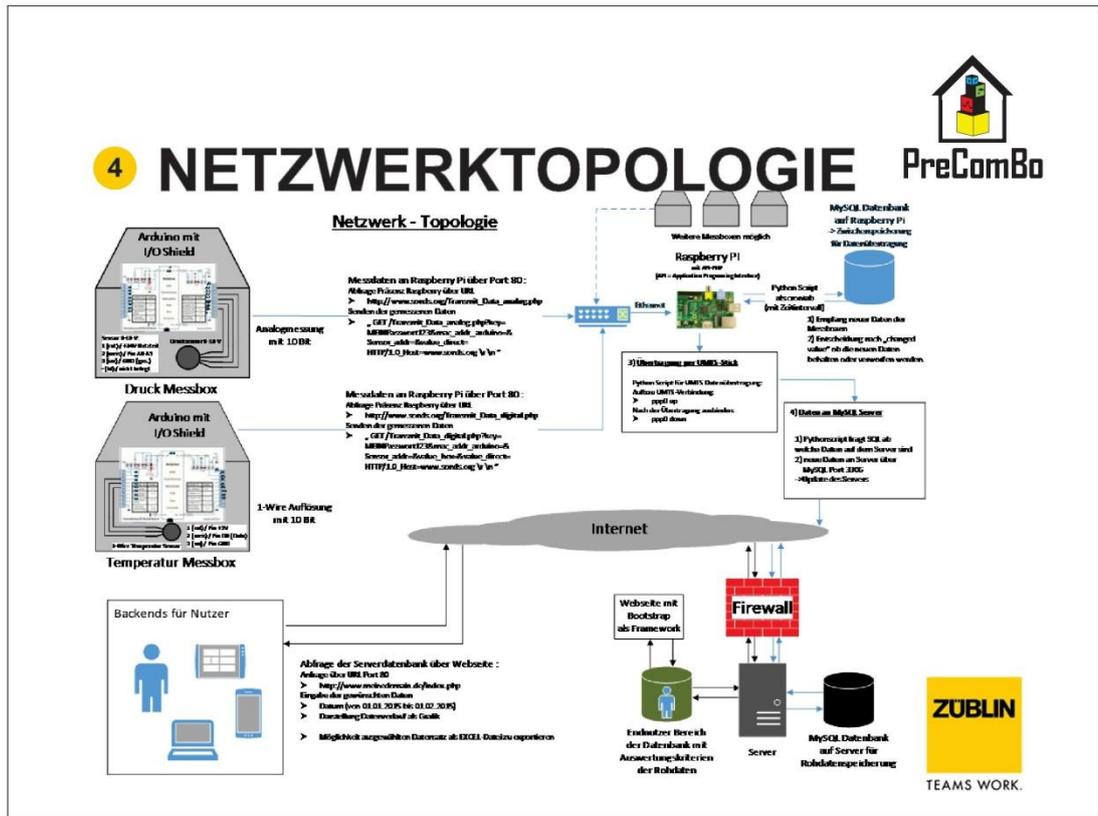


AUFBAU MESSBOXEN

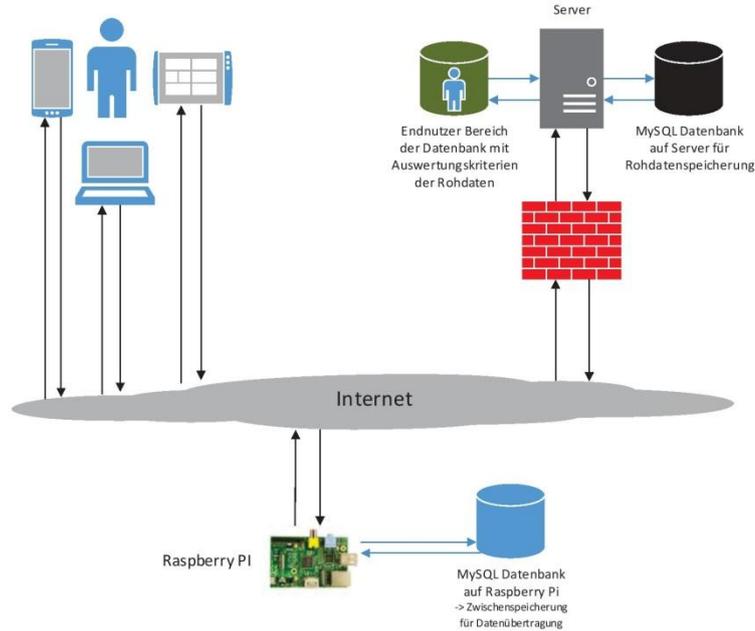
Arduino UNO + Universales I/O Shield Dr. Stall + Ethernet Shield

AUFBAU MESSBOXEN

Druckmessbox



NETZWERKTOPOLOGIE

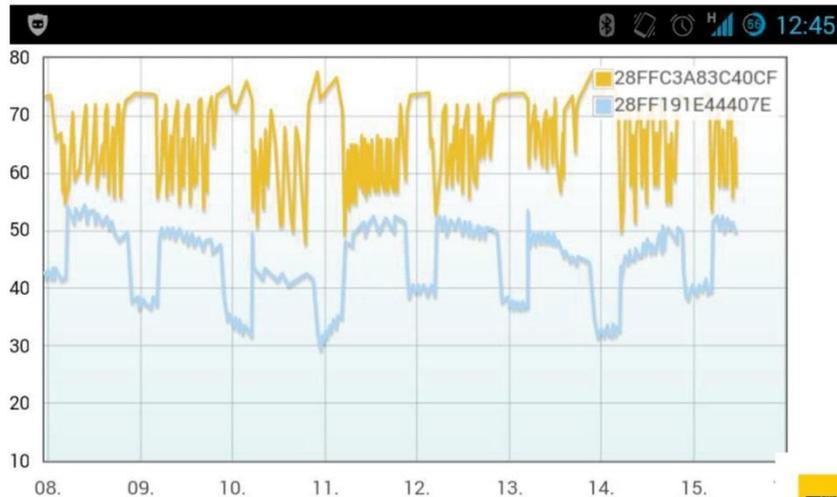


4 © Ed. Züblin AG, 11.09.2015
Seite 15

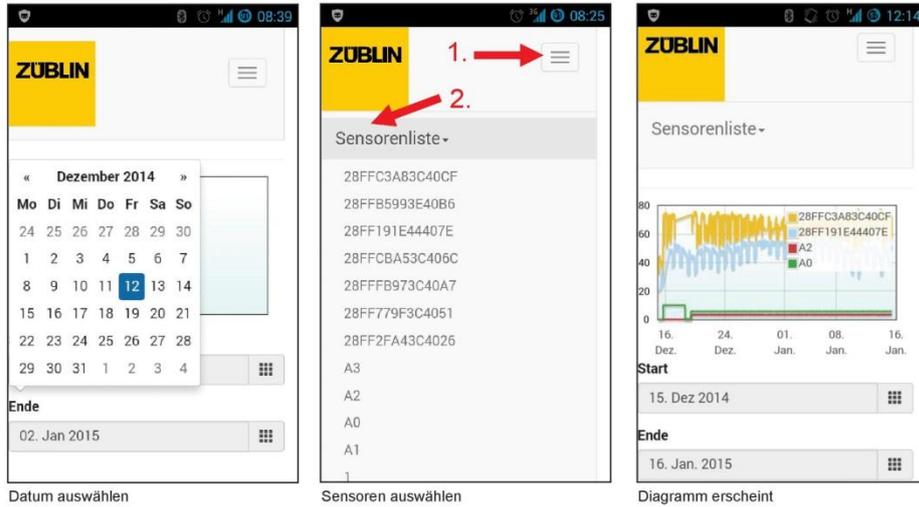
B.Eng. Raphael Bär – TGA-S



5 MOBILE NUTZUNG



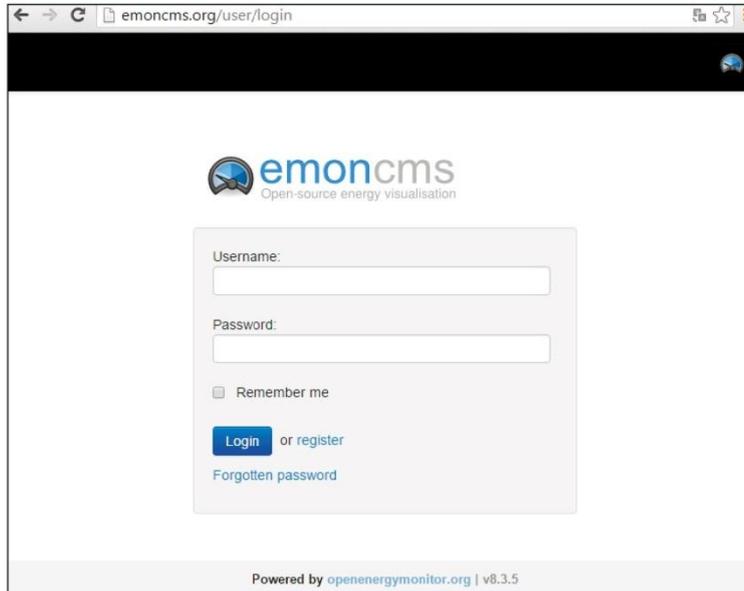
MOBILE NUTZUNG



MOBILE NUTZUNG



MOBILE NUTZUNG



Login-Terminal emoncms.org

MOBILE NUTZUNG



User-Bereich emoncms.org

MOBILE NUTZUNG

Node	Key	Name	Process list	last updated	value
0		power		inactive	0.00
3	28FFA5983C40A28FF			inactive	0.00
3	28FFFB973C40A728FF			inactive	0.00
3	28FF779F3C405100			inactive	0.00
3	transferred			inactive	0.00
3	28FF3A9B3C405A28FF			inactive	0.00
24	28FF109B3C401F00		log	inactive	72.4

Bereich „Inputs“ der lokalen emoncms Installation

5

© Ed. Züblin AG, 11.09.2015
Seite 21

B.Eng. Raphael Bär – TGA-S



MOBILE NUTZUNG

Id	Name	Tag	Datatype	Engine	Public	Size	Updated	Value
8	node:24.28FFB5993E40B600	Node:24	REALTIME	PHPFINA	🔒	0.0kb	3s ago	50.2
1	node:24.28FF3A9B3C405A28FF	Node:24	REALTIME	PHPFINA	🔒	0.0kb	inactive	23.6
3	node:24.28FFFB973C40A728FF	Node:24	REALTIME	PHPFINA	🔒	0.0kb	inactive	23.7
4	node:24.28FF779F3C405100	Node:24	REALTIME	PHPFINA	🔒	0.0kb	inactive	23.6
5	node:24.transferred	Node:24	REALTIME	PHPFINA	🔒	0.0kb	3s ago	1.00
6	node:24.28FF19783B405728FF	Node:24	REALTIME	PHPFINA	🔒	0.0kb	25s ago	-0.60
7	node:24.28FF109B3C401F00	Node:24	REALTIME	PHPFINA	🔒	0.0kb	inactive	0.00
2	node:24.28FFA5983C40A28FF	Node:24	REALTIME	PHPFINA	🔒	0.0kb	inactive	23.4
9	node:24.28FF109B3C401F28FF	Node:24	REALTIME	PHPFINA	🔒	0.0kb	99s ago	73.3
10	node:24.28FFEC5E3D408B28FF	Node:24	REALTIME	PHPFINA	🔒	0.0kb	77s ago	40.2
11	node:24.28FFC3A83C40CF00	Node:24	REALTIME	PHPFINA	🔒	0.0kb	inactive	0.00
12	node:24.28FFC3A83C40CF28FF	Node:24	REALTIME	PHPFINA	🔒	0.0kb	54s ago	72.0
13	node:24.28FF2FA43C402600	Node:24	REALTIME	PHPFINA	🔒	0.0kb	32s ago	59.7
14	node:24.28FF101E14407E00	Node:24	REALTIME	PHPFINA	🔒	0.0kb	38s ago	28.6

Bereich „Feeds“ der lokalen emoncms Installation

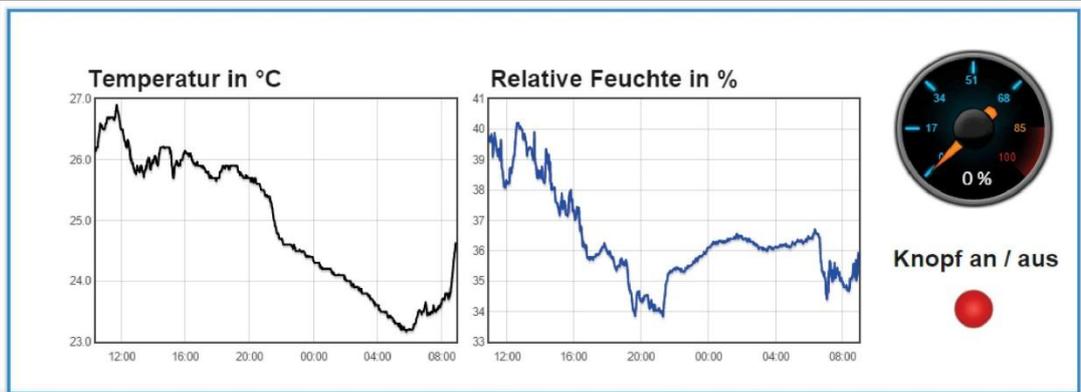
5

© Ed. Züblin AG, 11.09.2015
Seite 22

B.Eng. Raphael Bär – TGA-S



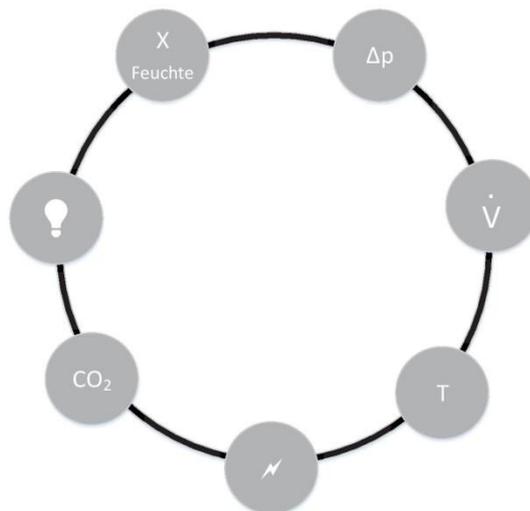
MOBILE NUTZUNG



Powered by openenergymonitor.org | v8.3.5

Beispiel eines einfachen Dashboards in emoncms

MOBILE NUTZUNG



**Anwendungsbereiche:
(Beispiele)**

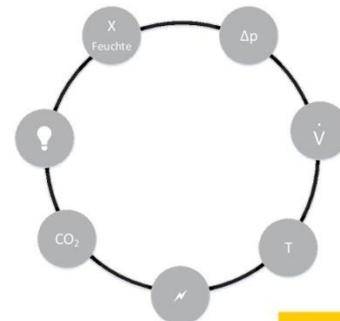
- Betonarbeiten
- Taupunktüberwachung
- Trocknungsprozesse
- Ventilauslegungen
- Differenzdruck -Kontrolle
- Funktionsnachweise Strahlpumpen
- Elektrische Leistungsverteilung
-

➤ **Datenschutz !!!**



PreComBo

Vielen Dank für
die Aufmerksamkeit!



12.2 Skript Sensorabfrage Arduino UNO R.3 für analoge Sensoren:

12.3 Skript Sensorabfrage Arduino UNO R.3 für digitale Sensoren: