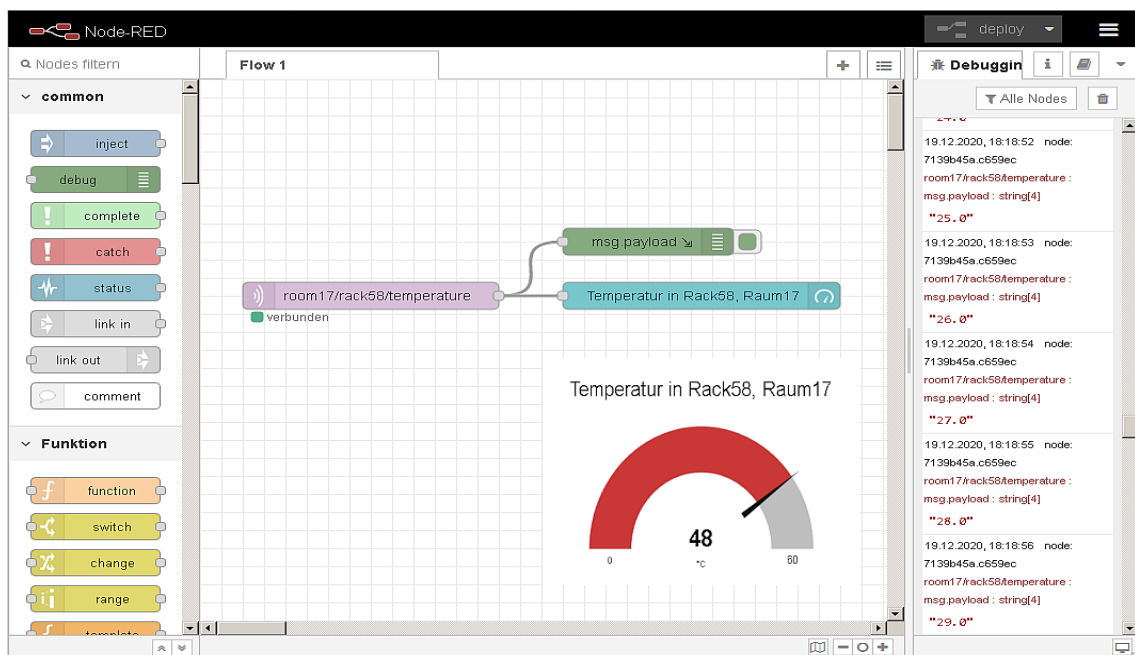


IT-Technik ITT11

Temperaturüberwachung im Serverraum



Aufgabe

Die Serverracks der FAI.DA.TE AG wurden von der SmartDevices GmbH mit Temperatursensoren ausgestattet und in das Cyber-physisches System der Firma eingebunden. Die von den Sensoren erfassten Werte werden über das Netzwerk mit MQTT bereitgestellt. Ihre Aufgabe ist es, mit der grafischen Entwicklungsumgebung Node-RED die Sensordaten zu visualisieren.

Inhalt

| | |
|--|----|
| 1. Ausgangslage und Sollzustand | 2 |
| 2. Was ist ein Cyber-physisches System (CPS)? | 3 |
| 3. Node-RED und MQTT auf dem Raspberry Pi installieren..... | 4 |
| 4. (Alternativ) Node-RED und MQTT über eine VM bereitstellen | 6 |
| 5. Sensordaten mit Node-RED darstellen..... | 7 |
| Anhang..... | 11 |
| Cyber-physische Systeme (CPS) | 11 |
| Node-RED | 13 |
| MQTT..... | 14 |
| Wichtige Linux-Befehle..... | 15 |
| Sicherheitshinweise..... | 16 |
| Netzwerkconfiguration in den Unterrichtsräumen..... | 17 |
| 6. Fragen..... | 18 |

Hinweise

- Lesen Sie die **Sicherheitshinweise** auf S. 16
- Beachten Sie die Verhaltensregeln bei Netzspannung und Elektrostatischer Entladung
- Hier finden Sie Informationen zur **Netzwerkconfiguration im Unterrichtsraum**
- **Diese** und alle weiteren Unterrichts-Unterlagen und auch die **Folien** finden Sie hier:
<https://webdav-ad-muenchen.musin.de/intk/austausch/lehrer/joachim.wolf/Unterricht/>

1. Ausgangslage und Sollzustand

Die FAI.DA.TE AG betreibt bundesweit 152 Baumärkte. Die Serverracks der FAI.DA.TE wurden von der SmartDevices GmbH mit Temperatursensoren ausgestattet und in das Cyber-physische System (CPS) der Firma eingebunden. Die Sensordaten werden bereits im Netzwerk über das MQTT-Protokoll bereitgestellt.

Im nächsten Schritt sollen die Sensordaten mit der grafischen Entwicklungsumgebung Node-RED visualisiert werden.

Sie als Mitarbeiter*in der SmartDevices GmbH werden zusammen mit Ihrem Team diesen Auftrag übernehmen.

Arbeitsschritte

- Lernen, was ein Cyber-physisches System (CPS) ist (S. 3)
- Installieren von Node-RED und MQTT auf dem Raspberry Pi (S. 4)
- Darstellen von Sensordaten mit Node-RED (S. 7)

2. Was ist ein Cyber-physisches System (CPS)?

Aufgabe: Ein Senior Analyst der SmartDevices GmbH (Lehrkraft) wird Sie mit einem Vortrag zum Thema Cyber-physische Systeme (CPS) informieren.

A

Beantworten Sie anschließend mit Hilfe des [Infotextes zu Cyber-physische Systeme \(CPS\)](#) auf S. 11 im Anhang und mit den [CPS-Vortragsfolien](#) die nachfolgenden Fragen.

Einigen Sie sich in Ihrem Team jeweils auf eine Antwort. Laden Sie Ihre Antworten zusammen mit den Fragen auf die von der Lehrkraft genannte Dateiablage hoch.

Der Senior Analyst bespricht anschließend die Antworten mit Ihnen.

- 1.) Erläutern Sie den Begriff CPS.
- 2.) Wo werden CPS hauptsächlich eingesetzt?
- 3.) Was sind Embedded Systems?
- 4.) Was ist eine SPS und wozu kann eine SPS eingesetzt werden?
- 5.) Nennen Sie 3 Beispiele für Single-Board Computer.
- 6.) Erklären Sie an Beispielen, was Sensoren und Aktoren sind und wie sie eingesetzt werden.
- 7.) Worin unterscheiden sich analoge von digitalen Signalen?
- 8.) Binäre Signale sind Digitalsignale. Was ist das Besondere bei binären Signalen?
- 9.) Worin unterscheiden sich die Anschlusstechniken Schnittstelle und Bus?
- 10.) Welche Programmiersprachen werden oft bei SBCs verwendet?
- 11.) Mit welchen Sicherheitsproblemen muss bei IoT-Systemen gerechnet werden?
- 12.) Bei CPS- oder IoT-Systemen werden leitergebundene und drahtlose Layer-2 Protokolle eingesetzt. Nennen Sie jeweils 2 dieser Protokolle.
- 13.) Welche Layer-2 Netzwerkkomponenten kommen bei CPS-Systemen zum Einsatz?
- 14.) Worin unterscheiden sich Router und Switch?
- 15.) Eine Bridge ist ein Medienwandler, der zwischen verschiedenen Layer-2 Protokollen vermittelt. Nennen Sie 2 Beispiele für den Einsatz von Bridges im Bereich IoT/CPS.
- 16.) Was ist NodeRED?

die nächsten Schritte:

Die gemessenen Temperaturdaten werden über das Netzwerkprotokoll MQTT übertragen, Um diese Daten grafisch darzustellen, benötigen wir ein Steuerungssystem (Raspi bzw. VM) auf dem ein MQTT-Server und eine grafische Entwicklungsumgebung laufen.

3. Node-RED und MQTT auf dem Raspberry Pi installieren

1.) Raspberry Pi installieren

- **Hinweis:** Da wir den RaspPi *headless* betreiben, d.h. ohne Tastatur und Bildschirm, müssen Sie sehr genau arbeiten!
- laden Sie von www.raspberrypi.org/software/operating-systems das *Raspberry Pi OS Lite* herunter und speichern Sie das ZIP-Archiv auf dem Windows-Desktop
- kopieren Sie aus dem ZIP-Archiv die .img-Datei heraus und legen Sie diese ebenfalls auf dem Desktop ab
- installieren Sie das Programm "[Raspberry Pi Imager](https://www.raspberrypi.org/software)" (<https://www.raspberrypi.org/software>)
- starten Sie den RasPi-Imager und wählen Sie unter dem Button *Betriebssystem* den Punkt "*Eigenes Image*" und wählen Sie die auf dem Desktop abgelegte .img-Datei aus
- stecken Sie einen USB-Stick (oder einen USB-Card-Reader mit MicroSD-Karte) an und wählen Sie diesen im RasPi-Imager unter "*SD Karte*" aus
- öffnen Sie im "Raspberry Pi Imager" mit der Tastenkombination **[Strg] [Shift] x** die *Erweiterten Optionen*, aktivieren Sie bei OS Modifizierungen *Immer verwenden*, geben Sie als Hostname **Pi-zzz** ein (**zzz** ist die Zahl im vierten Oktett der IP-Adresse Ihres PCs: z.B. 192.168.17.13 --> Hostname: Pi-13), aktivieren Sie SSH und geben Sie als Passwort *raspberry* ein, legen Sie in den Spracheinstellungen die Zeitzone *Europe/Berlin* und das Tastaturlayout *de* fest, und deaktivieren Sie *Telemetry*

Geben Sie hier den Hostnamen Ihres RasPi an:

RasPi-Hostname = Pi- _____

- schreiben Sie jetzt das Image auf den ausgewählten USB-Datenträger
- Windows fordert jetzt zur Datenträger-Formatierung auf. **Diese Meldung immer abbrechen!**
- **nur für Räume 2.3.X:** Auf dem Lehrer-Labor-PC muss ein DHCP-Server laufen!
- werfen Sie den USB-Datenträger aus und stecken Sie ihn am RasPi an
Achtung: MicroSD-Karten VORSICHTIG in den RasPi einlegen!
- verbinden Sie den RasPi mit einem Patchkabel mit dem Labor-Netz und mit der Stromversorgung
- warten Sie einige Zeit (ca. 2 min) bis Ihr RasPi vollständig gestartet ist
- der RasPi fordert über DHCP seine IP-Adresse an (und gibt Sie auch auf dem ggf. angeschlossenen Bildschirm aus)
- **nur für Räume 1.3.13/14 und 1.2.13/14:** In diesen Räumen können Sie die IP-Adresse Ihres RasPi ermitteln, indem Sie im Browser die Datei <http://192.168.1.242/files/dhcpd.leases> öffnen und dort nach dem DHCP-Lease für den von Ihnen vergebenen Hostnamen (Pi-xxx) suchen

Geben Sie hier die IP-Adresse Ihres RasPi an:

RasPi-IP = 192.168._____. _____

- versuchen Sie Ihren RasPi per Ping zu erreichen und prüfen Sie dann mit **nmap -p 22 RasPi-IP** ob auf dem RasPi der SSH-Port geöffnet ist
- bauen Sie mit **putty pi@RasPi-IP** eine SSH-Verbindung zum RasPi auf und akzeptieren Sie beim ersten Mal den neuen SSH-Host-Key (alternativ: Windows SSH-Client: **ssh pi@RasPi-IP**)
- melden Sie sich mit als Benutzer *user* und Passwort *raspberry* an und führen Sie die weiteren Konfigurationsschritte durch:

```
wget http://192.168.1.242/files/raspi.cfg
sudo bash raspi.cfg
```

2.) Node-RED installieren

- geben Sie folgenden Befehl GENAU SO ein (Achtung: **Das ist EINE Zeile!**):

```
bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)
```
- installieren Sie noch die Dashboard-Nodes:

```
cd ~/.node-red
```



```
npm install node-red-dashboard
```
- Node-RED soll bei jedem Systemstart geladen werden, dazu geben Sie folgendes ein:

```
sudo systemctl enable nodered.service
```
- zur Info: Node-RED stoppen:

```
sudo systemctl stop nodered
```


Node-RED starten:

```
sudo systemctl start nodered
```

3.) Node-RED direkt starten

- testen Sie, ob Node-RED korrekt funktioniert, dazu müssen Sie auf dem RasPi `node-red-start` eingeben, dann startet Node-RED direkt im Vordergrund
- war der Start erfolgreich, kann Node-RED mit einem Browser unter folgender [URL](http://RasPi-IP:1880) geöffnet werden:

```
http://RasPi-IP:1880
```
- das direkt gestartete Node-RED beenden Sie mit der Tastenkombination `Strg-c`

4.) Informieren Sie sich über Node-RED

- informieren Sie sich im Anhang auf S. 13 über Node-RED und beantworten Sie auch die Fragen!

5.) Den MQTT-Server mosquitto auf dem RasPi installieren Verwenden Sie dazu diese Befehle:

- ```
sudo apt-get update
```
- ```
sudo apt-get install -y mosquitto mosquitto-clients
```
- ```
sudo systemctl enable mosquitto.service
```
- zur Info: MQTT-Server stoppen: 

```
sudo systemctl stop mosquitto
```

  
MQTT-Server starten: 

```
sudo systemctl start mosquitto
```

**6.) Konfiguration des MQTT-Servers anpassen**

- editieren Sie mit `sudo nano /etc/mosquitto/mosquitto.conf` die Konfigurationsdatei des MQTT-Servers und ergänzen Sie am Ende der Datei die beiden folgenden Zeilen:  

```
listener 1883
```

```
allow_anonymous true
```

Hinweis zu nano: Datei mit [Strg] O schreiben und mit RETURN bestätigen  
nano verlassen mit [Strg] X
- starten Sie dann den MQTT-Server neu: 

```
sudo systemctl restart mosquitto
```

**7.) Informieren Sie sich über MQTT**

- informieren Sie sich im Anhang auf S. 14 über MQTT und beantworten Sie auch die Fragen!

**8.) Den MQTT-Server testen**

- öffnen Sie mit putty zwei getrennte SSH-Verbindungen zu Ihrem RaspPi
- starten Sie den MQTT-Server: 

```
sudo systemctl start mosquitto
```
- abonnieren (*subscribe*) Sie im aktuellen SSH-Fenster mit dem `mosquitto_sub` Client ein Topic. Wir verwenden das Topic `raum17/rack05/temperatur`:  

```
mosquitto_sub -h RasPi-IP -d -t "raum17/rack05/temperatur"
```
- senden Sie (*publish*) aus dem anderen SSH-Fenster mit dem Client `mosquitto_pub` zum angegebenen Topic einen Temperaturwert an den MQTT-Server. Dieser Wert sollte dann in dem ersten SSH-Fenster (dort wo wir abonniert haben) angezeigt werden  

```
mosquitto_pub -h RasPi-IP -d -t "raum17/rack05/temperatur" -m "21.0"
```
- wiederholen Sie diesen Befehl mit anderen Temperatur-Werten

## 4. (Alternativ) Node-RED und MQTT über eine VM bereitstellen

### 1.) Virtuelle Maschine mit Node-RED und MQTT-Server herunterladen

Zur weiteren Bearbeitung benötigen Sie einen MQTT-Server und das Node-RED-System. Diese Dienste haben Sie bereits auf einem RasPi bereitgestellt. Steht kein RasPi zur Verfügung, können Sie von <http://intranet/files/VMs/Windows10-NodeRED-MQTT.vm12.exe> eine selbstextrahierende VM mit MQTT-Server und NodeRED herunterladen und ausführen.

**Hinweis:** Die VM wird zunächst einmal neu starten! **Bitte warten Sie dies ab!**

Nach dem Neustart zeigt die VM ihre IP-Adresse (**VM-IP**) rechts oben am Desktop an.

Geben Sie hier die IP-Adresse Ihrer VM an:

**VM-IP** = 192.168. 1 . 42

**HINWEIS:** Ersetzen sie nachfolgend die Angabe **vm-IP** immer durch die IP-Adresse Ihrer VM (http://**vm-IP**: 1880 wird dann beispielsweise zu http://192.168.99.123:1880).

### 2.) Node-RED testen

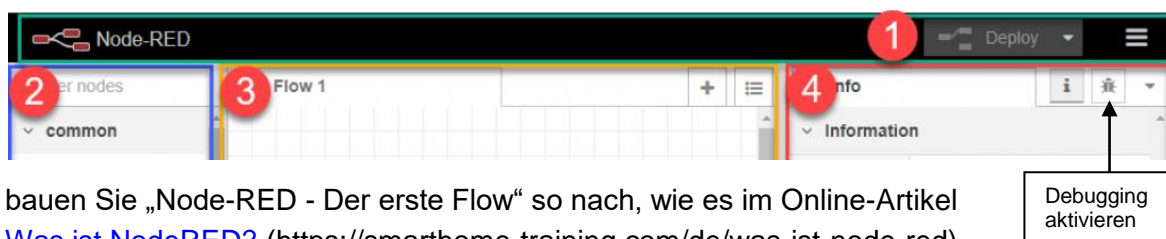
- ist der Start erfolgreich, kann Node-RED vom Host-PC aus mit einem Browser unter folgender [URL](#) geöffnet werden: `http://vm-IP:1880`

### 3.) Informieren Sie sich über Node-RED

- informieren Sie sich im Anhang auf S. 13 über Node-RED und beantworten Sie auch die Fragen!

### 4.) Erste Schritte mit Node-RED

- öffnen Sie vom Host-PC aus Node-RED über die entsprechende URL im Browser
- die Node-RED Oberfläche teilt sich in 4 Bereiche: Menüleiste (1, oben), Palette mit Nodes (2, links), Flow-Bereich (3, mitte), Seitenleiste (4, rechts) mit z.B. Infos, Debug-Meldungen



- bauen Sie „Node-RED - Der erste Flow“ so nach, wie es im Online-Artikel [Was ist NodeRED?](https://smarthome-training.com/de/was-ist-node-red) (<https://smarthome-training.com/de/was-ist-node-red>) gezeigt wird



### 5.) Informieren Sie sich über MQTT

Die Temperaturdaten werden über das Netzwerkprotokoll MQTT übertragen.

Um mehr über MQTT zu erfahren, sollten Sie sich jetzt im Anhang auf S. 14 über MQTT informieren und dort auch die Fragen beantworten!

## 6.) Den MQTT-Server mosquitto auf der VM testen

- öffnen Sie auf der VM die Windows-Dienstverwaltung, z.B. mit dem Befehl `services.msc`
- stellen Sie sicher, dass der Dienst "Mosquitto Broker" gestartet ist

- öffnen Sie jetzt ein Kommando-Fenster und abonnieren (*subscribe*) Sie mit dem Client `mosquitto_sub` ein Topic. Wir verwenden das Topic `raum17/rack05/temperatur`:

```
mosquitto_sub -h VM-IP -d -t "raum17/rack05/temperatur"
```

- öffnen Sie jetzt ein weiteres Kommando-Fenster und senden Sie (*publish*) mit dem Client `mosquitto_pub` an den MQTT-Server zum angegebenen Topic einen Temperaturwert.

Dieser Wert sollte dann im oben geöffneten Kommando-Fenster, in dem wir abonniert haben, angezeigt werden:

```
mosquitto_pub -h VM-IP -d -t "raum17/rack05/temperatur" -m "21.0"
```

- wiederholen Sie den Befehl auch mit anderen Temperatur-Werten

## 5. Sensordaten mit Node-RED darstellen

### 1.) Prüfen, ob die Sensordaten zur Verfügung stehen

- die Sensordaten werden bereits laufend über das Netz an Ihren MQTT-Server gesendet. Dies können Sie simulieren, indem Sie auf dem RasPi den Befehl **send-temp** eingeben bzw. auf dem Desktop der VM auf die Batch-Datei **send-temp.cmd** klicken.

Für die Temperaturdaten wird das folgende Topic verwendet:

```
raum17/rack05/temperatur
```

- öffnen Sie eine SSH-Verbindung zu Ihrem MQTT-Server (RasPi) oder falls Sie die VM verwenden, öffnen Sie dort ein Kommandofenster und abonnieren Sie mit dem MQTT-Client `mosquitto_sub` das Topic und beobachten Sie die ausgegebenen Werte.

Vervollständigen Sie dazu den benötigten Befehl:

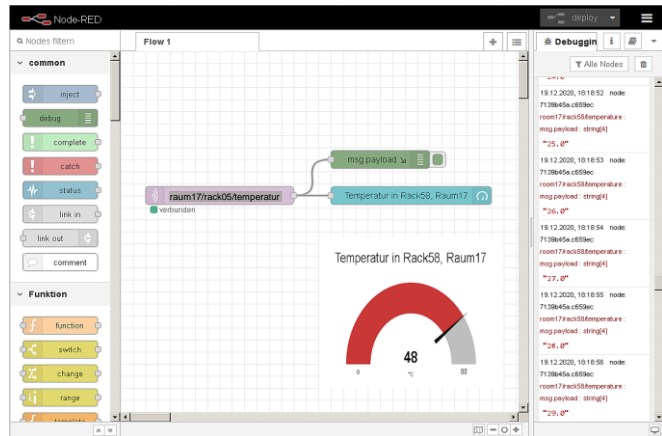
```
mosquitto_sub -h _____ -t _____
```

### 2.) Sensordaten in Node-RED ausgeben

- ziehen Sie einen "*mqtt in*"-Node in den Flow-Bereich (3)
- tragen Sie nach Doppelklick auf den "*mqtt in*"-Node dort unter *Server* die IP-Adresse Ihres MQTT-Servers (*RasPi-IP* bzw. *VM-IP*) und das oben unter **1.)** festgelegte Topic entsprechend ein
- ziehen Sie einen *Debug*-Node in den Flow-Bereich und verbinden Sie diesen mit dem Ausgang des MQTT-in-Node
- aktivieren Sie über den Button *Deploy* (1) den Flow
- beobachten Sie in der Seitenleiste (4) über die Debug-Meldungen die Sensordaten, evtl. müssen Sie über das Käfer-Symbol noch die Debugging-Nachrichten aktivieren
- geben Sie den im Rack gemessenen Temperaturbereich an: \_\_\_\_\_

### 3.) Sensordaten im Dashboard darstellen

- ziehen Sie zusätzlich einen *Gauge*-Node (Messgerät) in den Flow-Bereich und verbinden Sie ihn mit dem Ausgang des MQTT-Node
- Doppelklicken Sie auf den *Gauge*-Node und erstellen Sie unter *Group* den Namen *Temperaturüberwachung* erstellen Sie dort auch einen neuen *Tab* mit dem Namen "*Raum17 Rack05*", tragen Sie unter *Label* den Begriff *Temperatur* ein, geben Sie unter *Units* "*°C*" an und stellen Sie den *Range* min *10*, und max *40* ein
- aktivieren Sie über den Button *Deploy* den Flow
- öffnen Sie das Dashboard unter der URL (RasPi): <http://RasPi-IP:1880/ui> bzw. (VM) <http://VM-IP:1880/ui>
- da Sie das Dashboard jetzt häufiger aufrufen, könnten Sie diese URL auch als Verknüpfung in der Browser-Favoritenleiste ablegen oder als Link auf dem Desktop speichern

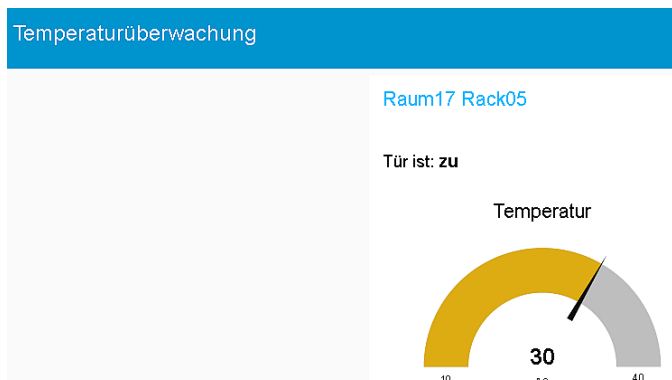


### 4.) Optional: Darstellung der Sensordaten erweitern

- von einem Tür-Sensor werden die Daten ebenfalls laufend über das Netz an Ihren MQTT-Server gesendet: offen/zu alle 10 s, mit **send-tuer** (RasPi) bzw. **send-tuer.cmd** (VM). Für den Tür-Sensor wird das folgende Topic verwendet: **raum17/rack05/tuer**
- ergänzen Sie Ihre Datendarstellung in Node-RED und konfigurieren Sie für dieses Topic einen zusätzlichen "*mqtt in*"-Node
- wählen Sie aus den *dashboard*-Nodes einen *Text*-Node aus, verbinden Sie ihn mit dem Ausgang des neuen "*mqtt in*"-Node, übernehmen Sie die angegebene *Group* "[*Temperaturüberwachung*] *Raum17 Rack05*", tragen Sie unter *Label* die Bezeichnung "*Tür ist:*" ein und wählen Sie eine linksbündige Darstellung aus

### 5.) Sensordaten anordnen

- die empfangenen Daten des Temperatur-Sensors und (optional) die des Tür-Sensors sollten nun auf dem Dashboard angezeigt werden
- wenn alles funktioniert hat, sichern Sie den erstellten Flow über das Hamburger-Menü in der Menüleiste unter *Exportieren* als **JSON**-Datei
- importieren können Sie einen Flow, indem Sie in der Menüleiste ganz rechts neben dem *Deploy*-Button über das Hamburger-Menü '≡' und dort unter *import* --> *select a file to import* die entsprechende JSON-Datei öffnen





**6.) (Optional) WLAN-Steckdose konfigurieren**

- die WLAN-Steckdose auf Werkseinstellungen setzen: Taste an der eingesteckten Steckdose für mehr als 20 Sekunden drücken
- WLAN-Client (PC, Smartphone) mit dem WLAN "delock-xxxx" verbinden (xxxx: das sind die 4 Ziffern, die auf Ihrer Steckdose vermerkt sind)
- im Browser delock-xxxx oder die IP-Adresse der WLAN-Steckdose (192.168.4.1) eingeben
- die Steckdose mit Name und Passwort des zu benutzenden WLANs konfigurieren und neu starten
- auf dem WLAN-Client mit **arp -a** den ARP-Cache auslesen und zu 192.168.4.1 die MAC-Adresse der WLAN-Steckdose ermitteln: MAC-Adresse: \_\_\_\_\_
- die WLAN-Steckdose sollte jetzt mit dem angegebenen WLAN verbunden sein
- die IP-Adresse der WLAN-Steckdose ermitteln: dazu in den Räumen 1.3.xx/1.2.xx in der Datei <http://192.168.1.242/files/dhcpd.leases> nach der MAC-Adresse oder dem Hostnamen der WLAN-Steckdose "delock-xxxx" suchen (xxxx: die 4 Ziffern auf der Steckdose)  
In allen anderen Räumen muss im Status-Fenster des DHCP-Servers über die MAC-Adresse die IP Ihrer WLAN-Steckdose ermittelt werden
- die IP Ihrer WLAN-Steckdose ist: **IP-Steckdose** : \_\_\_\_\_
- weitere Informationen in der Bedienungsanleitung der [Delock WLAN Steckdose 11827](https://www.delock.de/files/57391.download) (<https://www.delock.de/files/57391.download>)

**7.) (Optional) Schaltzustand der WLAN-Steckdose über MQTT protokollieren**

- verbinden Sie sich im Browser über `http://IP-Steckdose` mit der WLAN-Steckdose und testen Sie die Steuerung über den "An/Aus"-Button (Test um die Steckdose mit HTTP ein-/ausschalten)
- gehen Sie vom Hauptmenü der WLAN-Steckdose in das Menü "Konsole"
- schalten Sie die Steckdose über den Schalter an der physischen Steckdose ein und aus und beobachten Sie die Ausgabe der Konsole. Geben Sie das Topic an, das den Schaltzustand der WLAN-Steckdose direkt ausgibt: \_\_\_\_\_
- ergänzen Sie Ihre Sensordaten in Node-RED. Konfigurieren Sie hierzu einen neuen "mqtt-in"-Node und wählen Sie aus den dashboard-Nodes einen *text*-Node aus

### 8.) (Optional) WLAN-Steckdose über MQTT schalten

- die IP-Adresse Ihrer WLAN-Steckdose ist: *IP-Steckdose* : \_\_\_\_\_

- zunächst als erster Test die Steckdose mit HTTP ein-/ausschalten:

```
http://IP-Steckdose/cm?cmnd=Power%20on
```

```
http://IP-Steckdose/cm?cmnd=Power%20off
```

```
http://IP-Steckdose/cm?cmnd=Power%20Toggle
```

- die IP-Adresse Ihres MQTT-Servers (MQTT-Broker) ist:

*IP-MQTT-Server* : \_\_\_\_\_

- Steckdose für MQTT konfigurieren:

dazu unter *Einstellungen* --> *MQTT* --> *Host* die IP Ihres MQTT-Servers eintragen

- dann alle von der Steckdose an Ihren MQTT-Server übertragenen Topics abfragen

```
mosquitto_sub -h IP-MQTT-Server -t "#" -v
```

- Steckdose mit MQTT ein-/ausschalten

unter *Einstellungen* --> *MQTT* --> *topic* rackYY eintragen!

```
mosquitto_pub -h IP-MQTT-Server -t "cmnd/rackYY/POWER" -m ON
```

```
mosquitto_pub -h IP-MQTT-Server -t "cmnd/rackYY/POWER" -m OFF
```

```
mosquitto_pub -h IP-MQTT-Server -t "cmnd/rackYY/POWER" -m TOGGLE
```

YY repräsentiert die Platz-Nummer (zweistellig, ggf. mit führender Null)

- Schaltzustand der WLAN-Steckdose abfragen:

```
mosquitto_sub -h IP-MQTT-Server -t "stat/rack13/POWER"
```

- Zustand des Leistungsmessungs-Sensors abfragen:

a) zuerst das Reporting-Interval auf 10 sec einstellen

```
mosquitto_pub -h IP-MQTT-Server -t "cmnd/rackYY/TelePeriod" -m 10
```

oder im Browser: <http://192.168.1.150/cm?cmnd=Teleperiod%2010>

b) dann die Sensor-Daten abfragen

```
mosquitto_sub -h 192.168.1.148 -t "tele/rack13/SENSOR" -v
```

- erstellen Sie mit Node-RED einen Flow, so dass die Steckdose ab einer Temperatur von 28 °C eingeschaltet und unterhalb von 20 °C ausgeschaltet wird

### Weitere optionale Aufgaben

- [Linux-Befehle kennenlernen](#)

- in der [Node-Red Flow-Library](#) stöbern

## Anhang

### Cyber-physische Systeme (CPS)

#### CPS und IoT

Der Begriff Cyber-physische Systeme (CPS) kommt aus dem Industrie4.0-Umfeld. Bei CPS handelt es sich um einen Zusammenschluss intelligenter, vernetzter Steuerungssysteme, die aus Sensoren, Aktoren, Rechen- und Steuerungskomponenten bestehen.

CPS werden hauptsächlich in der industriellen Prozesssteuerung und Automation z.B. bei der Automobilherstellung und in der Gebäudeautomation ([Smart Home](#)) eingesetzt.

Weitere Beispiele für CPS sind: [Intelligente Stromnetze](#) (Smart Grid), [Smart City](#), [E-Health](#), [Smart Farming](#) (Landwirtschaft 4.0), [Smart-Factory](#) (Cyber-physisches Produktionssystem, CPPS).

Wenn CPS sich auf das Internet ausdehnen oder Internetdienste benutzen, werden sie auch [IoT](#) (Internet of Things) genannt. Häufig werden beide Begriffe gleichbedeutend verwendet.

#### CPS-Hardware

In CPS werden zur Daten- und Signalverarbeitung speziell dafür hergestellte Steuerungs-Rechner eingesetzt. Diese Steuerungs-Rechner besitzen viele Ein- und Ausgänge und werden auch Eingebettetes System (Embedded Systems) genannt. Da bei Steuerungsaufgaben keine hohe Rechenleistung benötigt wird, sind Steuerungsrechner meistens klein und haben einen geringen Energieverbrauch. Bei der Prozessautomatisierung im industriellen Umfeld werden häufig speicherprogrammierbare Steuerungen ([SPS](#), auch: PLC, Programmable Logic Control) eingesetzt, z.B. Siemens SIMATIC. Bei der Entwicklung von Steuerungssystemen oder bei IoT-Projekten werden meistens preiswerte [Einplatinencomputer](#) (Single-Board Computer, [SBC](#)) verwendet, z.B. [Arduino](#), [Raspberry Pi](#) oder [NodeMCU](#).



#### Sensoren und Aktoren

Ein Sensor ist ein Messfühler für physikalische oder chemische Eigenschaften, z.B. für Temperatur, Feuchtigkeit, Helligkeit, Beschleunigung, GPS-Koordinaten.

Ein Aktor (auch Steuerelement, engl. Aktuator) ist ein meist elektrisch steuerbares mechanisches Antriebselement, z.B. Relais (Schütz), Motor, Servo.

#### Analoge und digitale Signale

Sensoren erzeugen bzw. Aktoren benötigen analoge oder digitale Signale meist in Form einer elektrischen Spannung. Analoge Signale haben einen stufenlosen und beliebig feinen Verlauf. Digitale Signale können nur einzelne, fest vorgegebene Werte annehmen, z.B. haben binäre Signale nur zwei Zustände (z.B. die Werte 0 oder 1). Da Steuerungssysteme intern nur digitale Werte verarbeiten können, müssen die beiden Signalarten ineinander umgewandelt werden, dazu werden Analog-nach-Digital-Wandler (A/D-Wandler) bzw. Digital-nach-Analog-Wandler (D/A-Wandler) eingesetzt.

## Ein- und Ausgänge: Schnittstellen und Bus

Steuerungssysteme verfügen über eine große Zahl an Ein- und Ausgängen (Schnittstelle, Interface) um beispielsweise Sensoren oder Aktoren anzuschließen. Oft kann an eine Schnittstelle genau ein Gerät angeschlossen werden, z.B: serielle Schnittstelle ([RS-232](#)).

Schnittstellen, die den gleichzeitigen Anschluss mehrerer Geräte erlauben, heißen **Bus**, z.B: [Ethernet](#), [USB](#), [RS-485](#), [CAN](#), [KNX](#), [I<sup>2</sup>C](#), [SPI](#), [1-Wire](#).

## Netzwerk-Protokolle

sind Regeln und Formate darüber, wie Daten im Netz ausgetauscht werden. Da ein Protokoll meist genau eine Aufgabe übernimmt, ist ein Zusammenspiel von mehreren Protokollen nötig. Dazu werden die einzelnen Protokolle in Schichten (Layer) organisiert (z.B. [OSI-Modell](#), [DoD-Modell](#)).

Layer-2 Datenpakete (Frames) beinhalten mindestens die physikalischen Ziel- und Absender-Adressen (MAC), die zu übertragenden Daten und eine Prüfsumme zur Fehlererkennung.

Bei L3-Datenpaketen (IP) werden logische Adressen verwendet, die durch ihren Aufbau eine Gruppenbildung (Netzwerk) und eine gemeinsame Zustellung (Routing) ermöglichen.

In L4 (UDP, TCP) werden Ports zur Adressierung der Anwendungen in den Schichten 5-7 verwendet.

**Layer 5-7** HTTP, [MQTT](#), [OPC UA](#), [REST](#), [CoAP](#), [Modbus/TCP](#)

(Anwendung)

**Layer 4** UDP, TCP

**Layer 3** IP

**Layer 2** leitergebunden

[Ethernet](#), [CAN](#), [KNX](#),  
[Modbus/RTU](#) über RS-485

drahtlos

[WLAN](#), [Bluetooth/BLE](#), [DECT](#), [ZigBee](#), [Z-Wave](#),  
[LPWAN](#) (z.B. [LoRaWAN](#), [LTE](#), [NB-IoT](#), [Sigfox](#))

## Netzwerk-Komponenten

**Bridge** Umsetzung zwischen verschiedenen L2-Protokollen, z.B. mit einem AccessPoint von (Medienwandler) WLAN zu Ethernet, mit der Philips [Hue-Bridge](#) von ZigBee zu WLAN, usw.

**Switch** Zustellung auf L2 über die physikalischen Adressen (MAC-Adresse)

**Router** Zustellung auf L3 über die logischen Adressen (IP-Adresse)  
(IoT-Gateway)

## Netzwerksicherheit

Da Angriffe auf IoT-Systeme zunehmen und die Zahl IoT-basierte Malware stetig [ansteigt](#), müssen auch CPS- und IoT-Systeme durch entsprechende Maßnahmen vor Bedrohungen geschützt werden.

### organisatorische Schutzmaßnahmen

- [ISMS](#) einführen
- Informationssicherheits-Beauftragten für IoT bestimmen (IoT-ISB)
- Schutzbedarfsanalyse
- Risikobewertung

### technische Schutzmaßnahmen

- Netze über VLANs trennen
- Datenverkehr filtern und verschlüsseln
- Network Access Control ([NAC](#))
- VPN einsetzen
- Firewall (IoT-Secure-Gateway)

## Programmierung

Bei SPS kommen spezielle Programmier-Sprachen zum Einsatz: [AWL](#) (Anweisungsliste, Instruction List, IL), [KOP](#) (Kontaktplan) und FUP/FBS ([Funktionsbausteinsprache](#)).

Single-Board Computer werden meist mit den Sprachen [C](#), [C++](#), [JavaScript](#), [Python](#) programmiert.

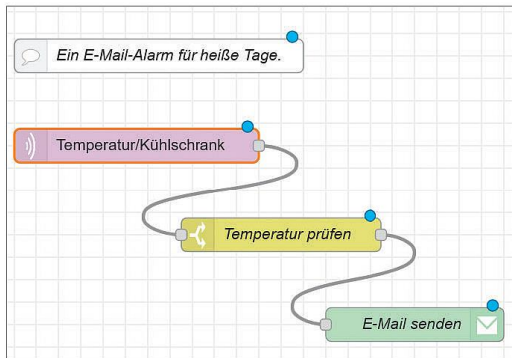
Häufig werden dazu Entwicklungsumgebungen (Integrated Development Environment, [IDE](#), z.B. [Arduino-IDE](#)) oder grafische Entwicklungswerkzeuge (z.B. [Node-RED](#)) eingesetzt.

## Node-RED

ist eine grafische Entwicklungsumgebung, die es Nutzern ermöglicht, typische CPS/IoT-Abläufe auch ohne nennenswerte Programmier-Kenntnisse umzusetzen. **Node-RED** arbeitet nach dem Baukastenprinzip, dabei werden einzelne Funktionsbausteine (Knoten, Nodes) durch Ziehen von Verbindungen zu *Flows* verbunden. Ein Flow (Programmablauf) verknüpft unterschiedliche Kommunikationspartner, wie zum Beispiel Geräte, Web-Dienste, Dateien oder Schnittstellen. Node-RED kann alles miteinander verbinden, was in irgendeiner Form Nachrichten senden oder empfangen kann. Um umfangreiche Abläufe umzusetzen, kann die Kommunikation an Bedingungen geknüpft werden.

### Beispiel

Eine konkrete Anwendung ist zum **Beispiel** ein **Kühlschrank-Alarm** in einer Apotheke. Dort muss die Temperatur der Medikamenten-Kühlschränke permanent überwacht werden. Sinkt sie unter einen festgelegten Wert, muss eine Aktion erfolgen, z.B. eine E-Mail an die Service-Mitarbeiter senden. Mit Node-RED kann dafür ein Flow definiert werden, in dem das entsprechende Topic vom MQTT-Broker abgefragt wird, an das der Sensor z.B. die aktuelle Temperatur in einem festen Intervall per (W-)LAN sendet. Sobald der Temperatur-Wert einen vorgegebenen Wert überschreitet wird dann eine E-Mail an die Service-Mitarbeiter gesendet. Das Alarm-System könnte auch die Temperatur automatisch herunterregeln lassen. Genauso ist es möglich, die Temperatur-Werte zur späteren Auswertung in eine Datei oder eine Datenbank zu schreiben.



Beispiel für einen Node-RED Flow

### Knoten

Die einzelnen Bausteine der Flows sind die Knoten (*Nodes*). Node-RED unterstützt drei Knotentypen, die Informationen empfangen, verarbeiten und senden können.

Eingabeknoten versorgen ein laufendes Programm mit eingehenden Daten aus unterschiedlichen Quellen, z.B. Anfragen von Web-Diensten, E-Mails, Twitter-Nachrichten oder Daten, die über einen Bus eines Steuerungsrechners oder über eine serielle Schnittstelle empfangen werden (z.B. mit MQTT).

Eingaben können auch durch Node-RED selbst erzeugt werden, z.B. durch zeitgesteuerte Prozesse. Die sind immer dann nützlich, wenn Vorgänge in festen Intervallen angestoßen werden müssen.

Verarbeitungsknoten sind dafür verantwortlich, Eingabe-Daten zu prüfen und auszuwerten.

Ausgabeknoten können auf verschiedene Arten mit der Außenwelt kommunizieren. Mit ihnen kann Node-RED unter anderem E-Mails verschicken, HTTP-Anfragen an Web-Dienste senden oder Schnittstellen und Busse ansteuern.

Nach dem Installieren der Dashboard-Nodes stehen auch einige Ausgabeknoten zur Visualisierung der Messwerte zur Verfügung.

Das Dashboard wird im Browser unter dieser URL geöffnet: <http://IPAdresse:1880/ui>

Node-RED verwandelt Flows nach dem Drücken des Buttons *Deploy* automatisch in JavaScript-Code der dann in der [Node.js](#)-Laufzeitumgebung ausgeführt wird.

### Fragen

Lesen Sie den obigen Infotext zu Node-RED durch und beantworten Sie folgende Fragen:

- 1.) Skizzieren Sie einen Programmablaufplan (PAP) zum genannten Beispiel „Kühlschrank-Alarm“
- 2.) Was ist der wesentliche Vorteil der Node-RED Entwicklungsumgebung?
- 3.) Erläutern Sie die drei Knotentypen von Node-RED.

A

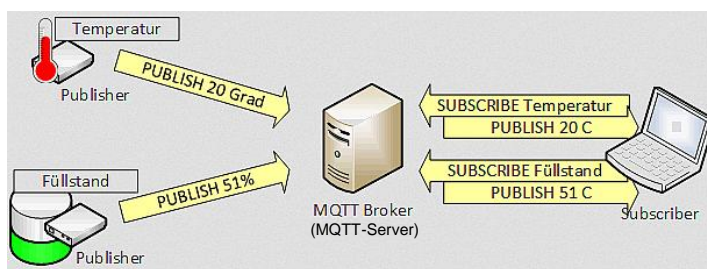


## MQTT

**MQTT** (*Message Queuing Telemetry Transport*) ermöglicht die robuste Übertragung von Nachrichten zwischen Geräten und wird auch zur Machine-to-Machine-Kommunikation (**M2M**) genutzt. Wegen seiner einfachen Verwendung ist MQTT weit verbreitet und eignet sich sehr gut für Automatisierungslösungen auch im Bereich CPS/IoT. MQTT ist ein Client-Server-Protokoll und arbeitet mit TCP auf Port 1883, bei verschlüsselter Übertragung auf Port 8883.

Zum MQTT-Server (Broker) können sich MQTT-Clients als Datenproduzent (Publisher) oder Datenutzer (Subscriber, Abonnent) verbinden. Das Senden (publish) und Empfangen (subscribe) von Nachrichten wird über sogenannte Topics (Thema) organisiert, die die Art der Nachricht darstellen. Mit Topics können die Nachrichten auch hierarchisch eingestuft werden, z.B. *Auto/Rad/3/Luftdruck* oder *raum17/rack02/temperatur*. Wenn Clients bestimmte Topics abonnieren, leitet der Server alle empfangenen Nachrichten dieses Topics an diese Abonnenten weiter.

Hinweis: Bei der Bezeichnung der Topics wird nach Groß- und Kleinschreibung unterschieden!



### Qualitätskontrolle

MQTT verwendet zur Datenübertragung das zuverlässige TCP. Bei Netzen mit vielen Übertragungsfehlern reicht dies jedoch nicht aus. Hier verwendet MQTT drei Servicequalitäten (QoS-Level), die das erfolgreiche Übertragen von Nachrichten garantieren sollen.

Die Zusicherung variiert von keiner Garantie (Level 0) über die, dass die Nachricht mindestens einmal ankommt (Level 1), bis hin zur Garantie, dass die Nachricht genau einmal ankommt (Level 2).

Der Unterschied zwischen Level 1 und 2 liegt darin, dass es bei Level 1 passieren kann, dass eine Nachricht öfter einen Client erreicht. Je nach Anwendungsfall sollte das passende Level gewählt werden, denn je höher das Level, desto höher ist die benötigte Bandbreite.

Mit *retained messages* können Nachrichten so gekennzeichnet werden, dass wann immer sich ein Client für ein Topic anmeldet, er zunächst diese Nachricht zugestellt bekommt. Das stellt sicher, dass ein Client sofort bei der Anmeldung einen definierten Wert bekommt und damit auch weiß, dass der Broker noch funktioniert. Die ist vor allem bei Topics mit Sensor-Daten interessant, denn dort könnte zum Beispiel die letzte gemessene Temperatur als *retained* gekennzeichnet werden.

Schließlich können MQTT-Clients beim Broker eine Nachricht hinterlegen, die dieser an ein ebenfalls hinterlegtes Topic sendet, wenn der Client unerwarteterweise nicht mehr erreichbar sein sollte (*last will, letzter Willen*).

### Fragen

Lesen Sie den obigen Infotext zu MQTT durch und schauen Sie sich folgendes Video an:

**MQTT in 7 Minuten einfach erklärt.** (<https://www.youtube.com/watch?v=ltcGWCvYa8o>)

A

Dann beantworten Sie folgende Fragen:

- 1.) Wozu wird das MQTT-Protokoll eingesetzt?
- 2.) Erläutern Sie folgende MQTT-Begriffe: Broker, Client, Publisher, Server, Subscriber, Topic.
- 3.) Erläutern Sie an einer Grafik den bei MQTT bestehenden Zusammenhang zwischen Broker, Topic, Publisher und Subscriber.
- 4.) Schlagen Sie eine Bezeichnung für ein Topic vor, die für die Helligkeit der Stehlampe in Ihrem Wohnzimmer verwendet werden könnte: \_\_\_\_\_

## Wichtige Linux-Befehle

### Dateiverarbeitung

|              |                                    |              |                                     |
|--------------|------------------------------------|--------------|-------------------------------------|
| <b>ls</b>    | Dateien und Verzeichnisse anzeigen | <b>cp</b>    | Dateien kopieren                    |
| <b>cd</b>    | Verzeichnis wechseln               | <b>rm</b>    | Dateien/Unterverzeichnisse löschen  |
| <b>cat</b>   | Datei-Inhalt ausgeben              | <b>mv</b>    | Dateien umbenennen/verschieben      |
| <b>less</b>  | Texte seitenweise betrachten       | <b>mkdir</b> | Verzeichnis erstellen               |
| <b>nano</b>  | einfacher Text-Editor              | <b>find</b>  | nach Dateien suchen                 |
| <b>df</b>    | freien Speicherplatz ausgeben      | <b>grep</b>  | nach Zeichenketten suchen           |
| <b>du</b>    | Platzverbrauch anzeigen            | <b>file</b>  | Analysieren des Inhalts von Dateien |
| <b>chmod</b> | Zugriffsrechte ändern              | <b>exit</b>  | abmelden                            |

### Administration

**Hinweis:** Die meisten der folgenden Befehle benötigen administrative (root) Rechte!

|                                     |                                                              |
|-------------------------------------|--------------------------------------------------------------|
| <b>sudo</b> <i>Befehl</i>           | den angegebenen Befehl mit administrativen Rechten ausführen |
| <b>ps ax</b>                        | Prozesse des Systems abfragen                                |
| <b>kill</b> <i>ProzessNr</i>        | einen Prozess terminieren                                    |
| <b>ifconfig</b>                     | eigene IP-Adresse abfragen (alt)                             |
| <b>ip addr</b>                      | eigene IP-Adresse abfragen (neu)                             |
| <b>nslookup</b> <i>Host</i>         | Namensauflösung testen                                       |
| <b>apt-get update</b>               | aktuelle Paket-Listen laden                                  |
| <b>apt-get install</b> <i>Paket</i> | ein Programm-Paket installieren                              |
| <b>apt-get purge</b> <i>Paket</i>   | ein Programm-Paket komplett löschen                          |
| <b>service</b>                      | einen Dienst starten oder stoppen (alt)                      |
| <b>systemctl</b>                    | einen Dienst starten oder stoppen (neu)                      |
| <b>systemctl</b>                    | das Start-Verhalten eines Dienstes ändern                    |
| <b>mount</b>                        | Einhängen von Dateisystemen                                  |
| <b>poweroff</b>                     | Rechner herunterfahren und ausschalten                       |
| <b>reboot</b>                       | Rechner neu starten                                          |

**Hinweis:** Linux-Befehle geben mit dem Argument **--help** eine Hilfe aus, z.B.: **ls --help**

## Sicherheitshinweise

### Verhaltensregeln bei Netzspannung

- arbeiten Sie **niemals** an elektrischen Geräten, solange diese unter Spannung stehen!  
Deshalb immer zuerst den Netzstecker ziehen!
- Arbeiten an spannungsführenden Teilen oder in deren Nähe sind grundsätzlich **verboten!**
- beschädigte elektrische Kabel dürfen nicht mehr verwendet werden
- **Sicherheitskleinspannung (Schutzkleinspannung)**: nur bei Wechselspannungen unter 25 V oder bei Gleichspannung unter 60 V kann auf einen Schutz gegen Berühren verzichtet werden. Diese Spannungen sind auch für Kinder und Tiere ungefährlich.

### Elektrostatische Entladung (ESD)

- sind Spannungsdurchschläge, die durch große Potentialunterschiede entstehen
- ESD kann elektrische Komponenten in Geräten schädigen
- fast alle elektronischen Bauelemente sind ESD-empfindlich!



Gefahrenzeichen  
für ESD-gefährdete Bauteile

### Schutz vor ESD

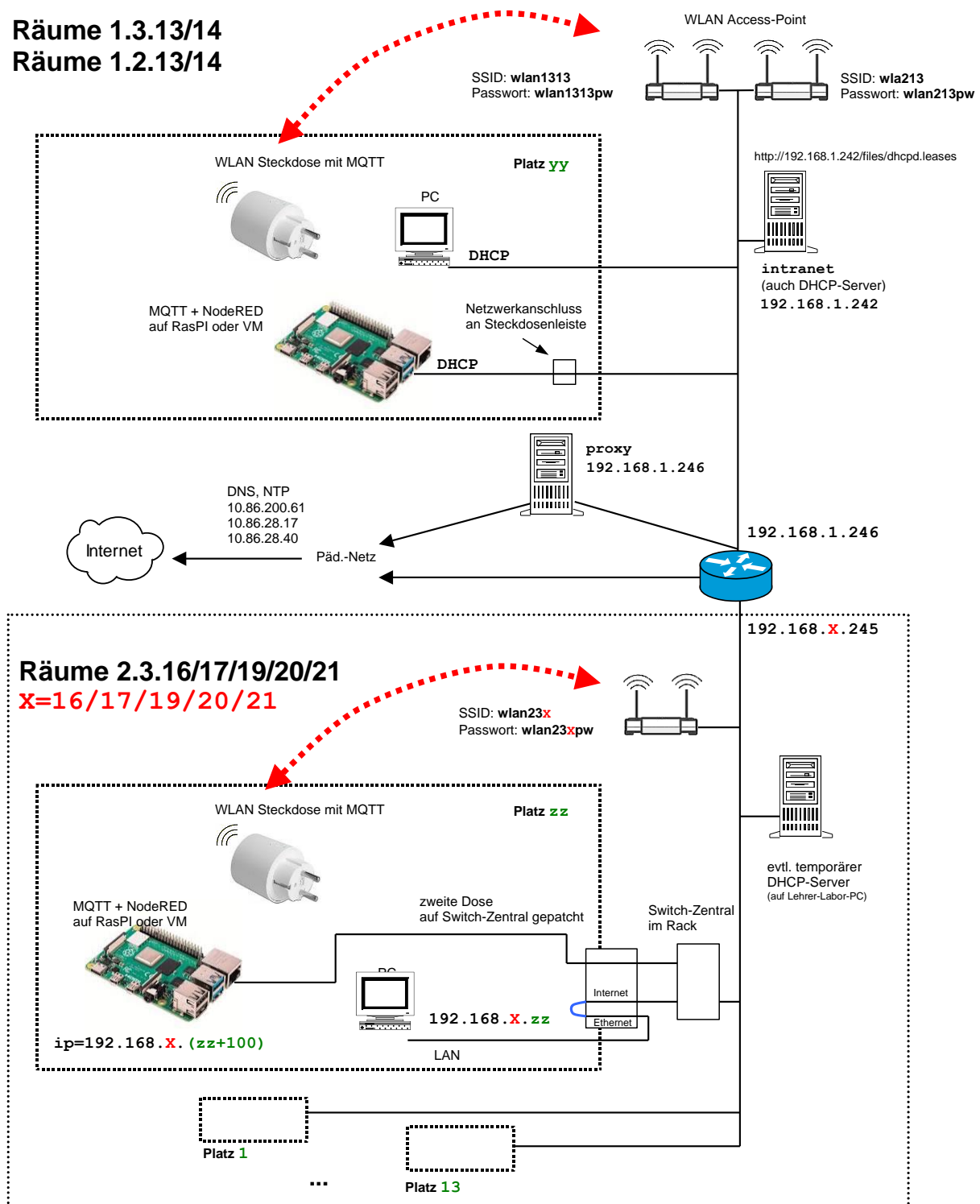
- Aufladung vermeiden, z.B. durch Ableiten und Erdung
- schnelles Entladen vermeiden, z.B. vorhandene elektrische Ladungen langsam über einen großen elektrischen Widerstand abfließen lassen



## Netzwerkconfiguration in den Unterrichtsräumen

Räume 1.3.13/14

Räume 1.2.13/14



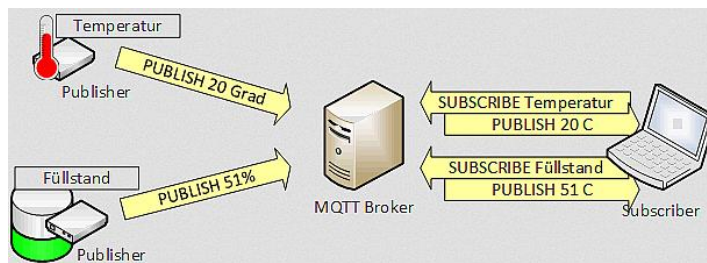
## Hinweise zur Konfiguration der Labornetz-Rechner

- die verwendeten Labornetz-Rechner müssen so konfiguriert sein, dass sie auf das Intranet zugreifen können: Test mit dem Befehl `ping intranet`
- die Labornetz-Rechner in den Räumen 2.3.xx müssen **korrekt gepatcht** sein und einen aktiven Link haben, bei den Patchfeldern auf dem Tisch muss *Ethernet* mit *Internet* verbunden sein
- in den Räumen 2.3.xx sollten Sie die **korrekte IP-Konfiguration** mit dem Befehl `ipx` setzen
- die Kommandozeilen-Tools sollten aktuell sein: Aktualisierung mit dem Befehl `ut`

## 6. Fragen

- die folgenden Fragen stammen aus dem [Fragenpool](#). Sie zeigen den Umfang und die Intensität der Unterrichtsinhalte und dienen zur Vorbereitung auf Leistungskontrollen und Abschlussprüfung
- *manchmal* haben schwierige Fragen ein Sternchen "\*", schwierigere zwei "\*\*\*"
- es werden KEINE Lösungen bereitgestellt, Ziel ist es, dass SIE die Lösungen selbst erstellen!
- Nachfragen, Anmerkungen, Lob und Kritik können Sie an Ihre Lehrkraft richten

- 1.) Erläutern Sie den Begriff CPS. Cyber-Physische Systeme
- 2.) Wo werden CPS hauptsächlich eingesetzt?
- 3.) Was sind Embedded Systems?
- 4.) Was ist eine SPS und wozu kann eine SPS eingesetzt werden?
- 5.) Nennen Sie 3 Beispiele für Single-Board Computer. Raspberry Pi, Arduino, NodeMCU
- 6.) Erklären Sie an Beispielen, was Sensoren und Aktoren sind und wie sie eingesetzt werden.  
sensor: erfasst physische gröÙe, aktor: empfangen signale und leiten diese weiter
- 7.) Worin unterscheiden sich analoge von digitalen Signalen? Digital: festgelegte zustände  
Analog: Kontinuierliche änderungen
- 8.) Binäre Signale sind Digitalsignale. Was ist das Besondere bei binären Signalen?  
Binäre signale haben nur 2 zustände (an/aus | 1/0)
- 9.) Anschlusstechnik bei Steuerungsrechnern: Was ist ein Bus? Anschluss mit nur einer schnittstelle (seriell)  
Bus: anschlüsse an mehrere schnittstellen
- 10.) Welche Programmiersprachen werden oft bei SBCs verwendet? C, C++, Python, JavaScript (z.b. NodeRed)
- 11.) Mit welchen Sicherheitsproblemen muss bei IoT-Systemen gerechnet werden?
- 12.) Bei CPS- oder IoT-Systemen werden leitergebundene und drahtlose Layer-2 Protokolle eingesetzt.  
 Nennen Sie jeweils 2 dieser Protokolle.
- 13.) Welche Layer-2 Netzwerkkomponenten kommen bei CPS-Systemen zum Einsatz?
- 14.) Worin unterscheiden sich Router und Switch? Router ist nur zwischen LAN und WAN (2 anschlüsse)  
Switch ist Netzwerk-intern
- 15.) Eine Bridge ist ein Medienwandler, der zwischen verschiedenen Layer-2 Protokollen vermittelt.  
 Nennen Sie 2 Beispiele für den Einsatz von Bridges im Bereich IoT/CPS.
- 16.) Wozu wird das MQTT-Protokoll eingesetzt?
- 17.) Wofür steht der Begriff [M2M](#)?
- 18.) Erläutern Sie folgende MQTT-Begriffe: Broker, Publisher, Topic, Subscriber.
- 19.) Erläutern Sie an der folgenden Darstellung den bei MQTT bestehenden Zusammenhang zwischen Broker, Topics, Publisher und Subscriber.



- 20.) Node-RED ist eine grafische Entwicklungsumgebung, bei der Programmabläufe durch Ziehen von Verbindungen zwischen Funktionsbausteinen erstellt werden können.  
 Nennen Sie 3 Vorteile und 2 Nachteile dieser Art der Programmierung.
- 21.) Beim Umgang mit elektronischen Geräten taucht häufig der Begriff ESD auf. Erläutern Sie ESD und nennen Sie Gefährdungen, die bei elektronischen Geräten durch ESD entstehen können.