

Bestandteile und Aufbau eines IoT (2)

Herr der Dinge

von Dr. Christian Knermann und Andreas Schröder

Im ersten Teil dieser Workshopserie haben wir Übertragungsprotokolle und Sensoren für das Internet der Dinge vorgestellt. Als praktische Übungen haben wir einen ersten Sensor per WLAN angebunden und Daten an einen öffentlichen Broker geschickt. Im Hinblick auf die Informationssicherheit ist dies in der Praxis natürlich nicht zu empfehlen. Daher bauen wir nun im zweiten Teil unsere eigene IoT-Plattform samt Datenbank, Broker, Raspberry Pi und Visualisierung auf.



Quelle: Pop Nukoomat – 123RF

Teil 1 unseres Workshops verschaffte uns einen Überblick über relevante Funktechnologien und Protokolle auf Anwendungsebene, die Sensoren und Aktoren über kurze wie auch größere Distanzen energieeffizient den Weg ins Internet of Things (IoT) ebnen. Weiterhin haben wir uns der Technik entsprechender Sensor-/Aktor-Knoten gewidmet und einen Mikrocontroller vom Typ ESP8266 dazu gebracht, per MQTT Daten an den öffentlichen MQTT-Broker der Eclipse Foundation zu schicken. Dabei wurde schnell deutlich, dass dies im produktiven Betrieb keine gute Idee ist, da alle Nutzer eines öffentlichen Brokers MQTT-Topics mitlesen und beliebige Kommandos an die verbundenen Sensoren und Aktoren schicken könnten.

Im zweiten und abschließenden Teil des Workshops werden wir daher unsere Testumgebung (Bild 1) weiter ausbauen und die IoT-Plattform in eigener Regie installieren. Der erste Baustein, den wir dazu benötigen, ist ein eigener Broker, bei dem unser Sensor seine Daten abliefern kann. Ein kommerzieller Vertreter dieser Art ist beispielsweise HiveMQ, im Bereich der kostenfrei einsetzbaren Lösungen stehen etwa RabbitMQ oder auch Eclipse Mosquitto [1] bereit. Letzteren werden wir im Folgenden verwenden, um die Daten unserer Sensoren ins Backend zu befördern.

Für den Anfang eine kleine IoT-Plattform

Mit dem weiten Feld möglicher IoT-Plattformen hat sich IT-Administrator bereits in der Vergangenheit beschäftigt [2, 3]. Neben den Platzhirschen mit ihren Clouddiensten wie Azure IoT, AWS IoT oder auch der KI von IBM Watson bringen sich hier unter anderem etablierte Hersteller, darunter Siemens und SAP, in Stellung.

Doch für erste Gehversuche müssen Sie sich nicht gleich an einen großen Anbieter binden. Eine Workflow-Engine zur weiteren Verarbeitung der Sensordaten können Sie auch selbst betreiben. In unserem Beispiel kommt hierfür Node-RED [4] zum Einsatz. Die Open-Source-Software kümmert sich um die Validierung eingehender Daten, kann unplausible Werte und Ausreißer erkennen und aussortieren sowie auch bekannte Abweichungen korrigieren. Node-RED kann selbst unmittelbar Ereignisse auslösen, etwa einen Aktor ansteuern, eine Lampe einschalten oder ein Ventil öffnen. Ebenso reicht die Software die Daten zur weiteren Verarbeitung an nachgelagerte Dienste weiter. Die universell einsetzbare Plattform IFTTT ("if this then that") verknüpft nahezu beliebige Systeme miteinander. Alternativ landen die Daten zur späteren Auswertung erst einmal in einer Datenbank.

Abkehr von SQL in der Datenhaltung

Sollen die Daten der Sensoren dauerhaft gespeichert werden, liegt es nahe, hierfür eine Datenbank zu installieren. Im reichhaltigen Angebot der Open-Source-Produkte fällt die Wahl schnell auf MySQL oder dessen Fork MariaDB. Doch als klassische SQL-Datenbanken eignen sich diese nur bedingt für unsere Zwecke. Besser geeignet sind hier Lösungen aus der Familie der NoSQL-Datenbanken. MongoDB, CouchDB und Artverwandte speichern Daten, ohne dass vorab ein fixes Datenbankschema definiert werden müsste. Das Schema entsteht implizit, sobald Datensätze in eine Datenbank eingefügt werden.

Im Fokus unseres Interesses steht ein spezialisierter Vertreter dieser Familie: Bei InfluxDB [5] handelt es sich um eine sogenannte Time-Series-Database (TSDB). Gegenüber den zuvor genannten SQL-Datenbanken besteht der Vorteil einer TSDB darin, dass sie genau für den Einsatzzweck entwickelt wurde, große Mengen von Datensätzen einer chronologischen Reihenfolge aufzunehmen. Im einfachsten Fall handelt es sich dabei jeweils um ein Tupel bestehend aus einem Zeitstempel, einer namentlich bezeichneten Variablen und deren konkreten Wert zu diesem Zeitpunkt. Eine TSDB schreibt diese Datensätze nicht nur weg, sondern

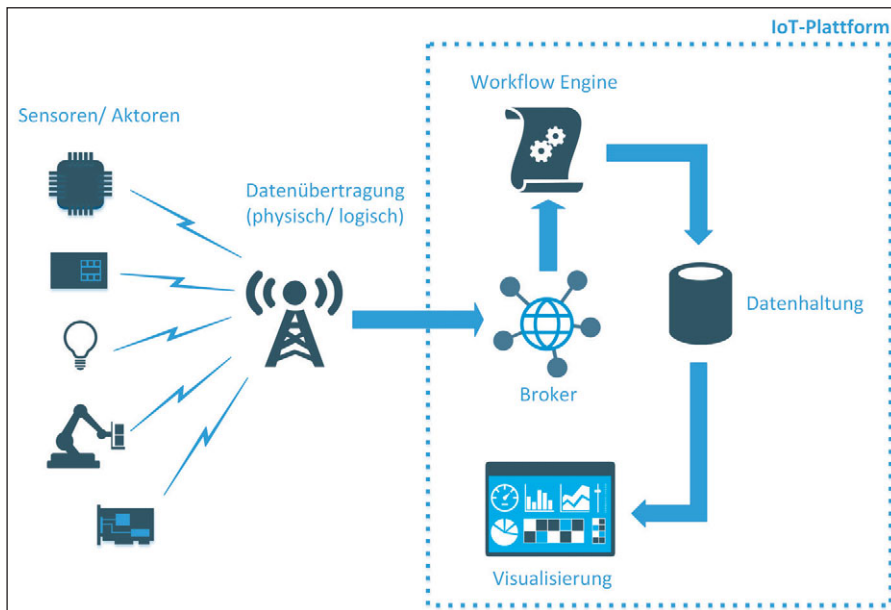


Bild 1: Unser kompletter Testaufbau in der Übersicht. In diesem Teil des Workshops bauen wir eine IoT-Plattform on-premises auf.

bringt auch sehr effiziente Routinen mit, um ältere Daten zu konsolidieren. Für länger zurückliegende Perioden kann die Datenbank etwa Einzelwerte zu Mittelwerten aggregieren und so Speicherplatz sparen.

Um die Visualisierung der Daten in Dashboards kümmert sich dann wiederum eine der vielen IoT-Plattformen, Node-RED oder aber das für diesen Zweck verbreitete Grafana [6], mit der wir auch unseren Anwendungsfall kompletieren wollen. Entscheidend ist dabei heutzutage, dass wir zur Visualisierung der Daten Systeme einsetzen, die den Zugriff per Webbrowser ermöglichen, um allen Beteiligten zu jeder Zeit und von jedem Ort aus einen komfortablen, sicheren Zugriff auf die aktuellen Daten zu erlauben.

Raspberry Pi als Server vorbereiten

Zunächst benötigen Sie ein System, auf dem Sie Ihre eigene IoT-Plattform installieren. Hierfür soll uns im Folgenden ein Raspberry Pi dienen. Mit Abstrichen bei der Performance reicht im Zweifelsfall der kleinste Spross der Familie, der Raspberry Pi Zero. Mehr Freude kommt auf mit einem Raspberry Pi 2 oder 3 Model B. Letzteren schließen wir per Kabel ans Netzwerk an und sparen uns so das WLAN-Setup.

Laden Sie das aktuelle Image des Betriebssystemes "Raspbian Lite" [7] herunter und

entpacken Sie es. Nun benötigen Sie ein Tool, um das Image auf eine SD-Karte zu schreiben. Hierfür eignet sich beispielsweise der kostenfreie "Win32 Disk Imager" [8]. Installieren und starten Sie das Tool. Nun wählen Sie die Image-Datei sowie den gewünschten Datenträger und schreiben das Image. Die SD-Karte enthält anschließend zwei Partitionen. Die erste davon mit dem Label "boot" ist auch unter Windows lesbar. Legen Sie darin eine leere Textdatei mit dem Namen "ssh" ohne Dateiendung an. Dies erlaubt den Remote-Zugriff per SSH, sobald der Raspberry Pi gebootet ist. Legen Sie nun die SD-Karte ein und starten Sie das Gerät. Überprüfen Sie auf Ihrem DHCP-Server, welche IP-Adresse der Raspberry Pi erhalten hat, verbinden Sie sich per SSH-Client mit dem System und melden Sie sich mit dem Benutzer "pi" und dem Passwort "raspberrypi" an.

Als erste Amtshandlung starten Sie die Systemkonfiguration mittels

```
sudo raspi-config
```

Ändern Sie das Admin-Passwort, passen Sie unter "Localisation Options / Change Timezone" die Zeitzone an und vergrößern Sie mittels "Advanced Options / Expand Filesystem" das Dateisystem, sodass es den kompletten auf der SD-Karte verfügbaren Platz belegt. Verlassen Sie nun die Systemkonfiguration und lehnen Sie den Wunsch

nach einem Neustart ab. Vor dem Reboot aktualisieren Sie noch das System und lösen erst dann den Neustart aus:

```
sudo apt-get update
sudo apt-get dist-upgrade
sudo reboot
```

MQTT-Broker und Workflow-Engine in Betrieb nehmen

Anschließend installieren Sie den MQTT-Broker Mosquitto, starten den Broker und richten ihn so ein, dass er zukünftig beim Boot automatisch startet:

```
sudo apt-get install -y mosquitto
mosquitto-clients

sudo systemctl start mosquitto.
service

sudo systemctl enable mosquitto.
service
```

Nun können Sie Ihren Sensor so konfigurieren, dass er seine Daten nicht mehr in die Cloud, sondern an den lokalen Broker schickt. Rufen Sie im Browser das Webfrontend des Sensors auf und editieren Sie unter dem Reiter "Controllers" den dort vorhandenen Eintrag. Die richtige Wahl für die Option "Locate Controller:" lautet nun "Use IP Address". Im Feld darunter tragen Sie die IP-Adresse Ihres Raspberry Pi ein und speichern das Ganze per "Submit".

Weiter geht es im Client "MQTT.fx". Auch dort editieren Sie über das Zahnrad-Symbol die Einstellungen. Im Feld "Broker Address" tragen Sie statt "m2m.eclipse.org" ebenfalls die IP-Adresse des Geräts ein. Verbinden Sie sich nun mit dem Broker, können Sie im Bereich "Subscribe" die Topics Ihres Sensors abonnieren. Wir haben damit den ersten Baustein unserer IoT-Plattform in Betrieb genommen.

Im nächsten Schritt folgt die Installation der Workflow-Engine Node-RED mit dem Befehl

```
bash <(curl -sL https://raw.githubusercontent.com/node-red/
```

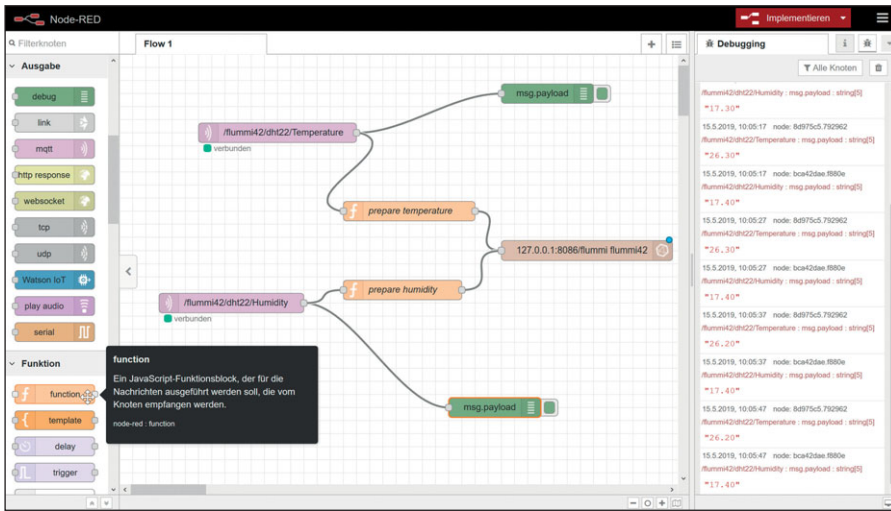


Bild 2: Mittels grafisch konfigurierter Workflows befördert Node-RED per MQTT empfangene Messwerte in die InfluxDB.

```
raspbian-deb-package/master/
resources/update-nodejs-and-nodered)
```

Die Installation kann eine Weile dauern und ist nicht sehr mitteilzaam. Um den Fortschritt im Detail zu verfolgen, öffnen Sie eine zweite SSH-Sitzung und verschaffen sich Gewissheit mit

```
sudo tail -f /var/log/nodered-
install.log
```

Um im Folgenden die Influx-Datenbank an Node-RED anzufanschen, benötigen Sie noch ein passendes Modul. Das ergänzen Sie mittels

```
cd /home/pi/.node-red/node_modules
npm install node-red-contrib-
influxdb
```

Ist die Installation beendet, starten Sie auch Node-RED und sorgen ebenfalls dafür, dass der Dienst nach einem Reboot automatisch startet:

```
sudo systemctl start nodered.service
sudo systemctl enable nodered.
service
```

Nun öffnen Sie im Browser über die URL "http://IP-des-Raspberry-Pi:1880" das Webfrontend von Node-RED. Auf der linken Seite des Fensters finden Sie eine Auswahl der in Node-RED zur Verfügung stehenden Knoten. Diese können Sie einfach per Drag-

and-Drop auf die Arbeitsfläche ziehen, per Doppelklick konfigurieren und zu Workflows verbinden. Die Workflows können nahezu beliebig komplex werden und liefern genügend Stoff für einen weiteren Workshop. Wir beschränken uns daher an dieser Stelle auf einige grundlegende Funktionen.

Messwerte per MQTT empfangen und ablegen

Im ersten Schritt geht es an die Integration unseres Sensors per MQTT. Ziehen Sie einen Knoten vom Typ "Eingabe / MQTT" auf die Arbeitsfläche. Per Doppelklick darauf erreichen Sie dessen Eigenschaften. Das Feld "Server" ist zunächst noch leer. Über das "Stift"-Symbol daneben fügen Sie einen MQTT-Broker hinzu, dem Sie einen beliebigen Namen geben und als Zieladresse "127.0.0.1" verwenden.

Um die Vertraulichkeit und Integrität der Daten sicherzustellen, empfiehlt es sich spätestens im Echtbetrieb, auf verschlüsselte Varianten der eingesetzten Protokolle zu achten. So sollten nicht nur HTTP und REST mittels Zertifikaten TLS-gesichert werden. Weiterhin sollte auch die Variante MQTTs zum Einsatz kommen. Auf Grund der damit verbundenen Komplexität verzichten wir jedoch zunächst auf die Absicherung, um den Rahmen des Workshops nicht zu sprengen.

Weitere Angaben für den Server sind somit nicht erforderlich. In das Feld "Thema" tragen Sie das gewünschte MQTT-Topic, in diesem Fall "/flummi42/dht22/

Temperature", ein und bestätigen das Ganze per Klick auf "Fertig".

Ziehen Sie nun einen Knoten vom Typ "Ausgabe / Debug" auf die Arbeitsfläche und verbinden Sie diesen mit dem ersten Knoten. Über die Schaltfläche "Implementieren" oben rechts in der Kopfzeile des Fensters speichern Sie Ihr Werk. Direkt darunter finden Sie in der vertikalen Leiste auf der rechten Seite des Fensters ein "Käfer"-Icon. Klicken Sie darauf, erscheinen im Bereich darunter die Debug-Nachrichten. Hier sollte Node-RED nun die per MQTT vom Sensor empfangenen Temperaturwerte anzeigen.

Wiederholen Sie die Prozedur mit einem weiteren MQTT-Knoten für das Topic "/flummi42/dht22/Humidity". Sobald Sie sich anhand der Debug-Nachrichten davon überzeugt haben, dass die Daten Ihres Sensors ankommen, können Sie über die Buttons am rechten Rand der Debug-Knoten das Debugging deaktivieren. Vergessen Sie nicht, jede Änderung innerhalb der Arbeitsfläche per Klick auf "Implementieren" zu speichern.

Im nächsten Schritt wollen wir die per MQTT empfangenen Werte nun weiterverarbeiten und zur späteren Visualisierung per Grafana in einer Influx-Datenbank speichern. Dazu fügen Sie das entsprechende Repository hinzu und installieren den Datenbank-Server

```
sudo curl -sL https://repos.
influxdata.com/influxdb.key |
sudo apt-key add -
```

```
sudo echo "deb https://repos.
influxdata.com/debian stretch
stable" | sudo tee /etc/apt/
sources.list.d/influxdb.list
```

```
sudo apt-get update
sudo apt-get install influxdb
```

Auch Influx soll künftig zusammen mit dem System automatisch starten:

```
sudo systemctl enable influxdb
sudo systemctl start influxdb
```

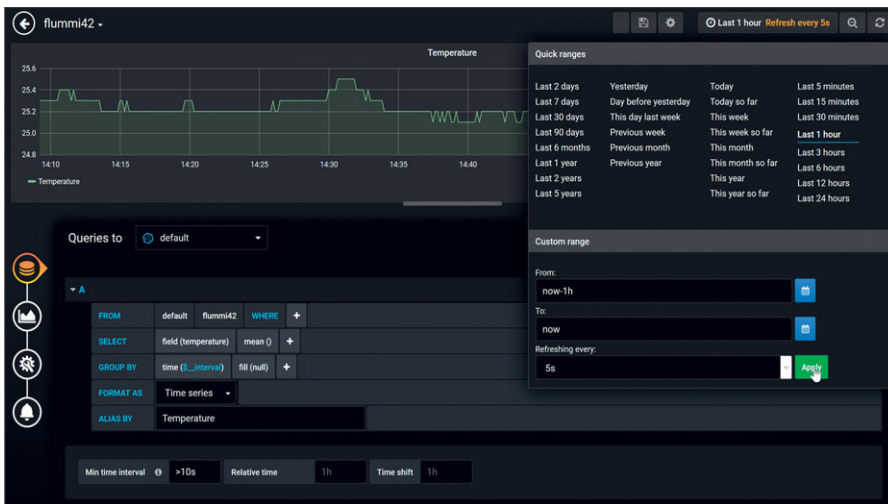


Bild 3: Grafana unterstützt mit einem Editor dabei, Datenbankabfragen zu formulieren.

Nun legen Sie eine Datenbank zur Aufnahme der Messwerte an:

```
sudo influx -execute 'CREATE
  DATABASE flummi'
```

Mittels *sudo influx* starten Sie den interaktiven Prompt des Datenbankservers und können sich so davon überzeugen, dass dieser läuft. Das Kommando *show databases* sollte die Systemdatenbank

"_internal" und unsere soeben angelegte Datenbank "flummi" zurückliefern. Mit *quit* oder *exit* verlassen Sie den Prompt dann wieder.

Damit sind wir nun gerüstet, um Messwerte in die Datenbank zu befördern. Das erledigen Sie im Frontend von Node-RED, indem Sie einen "InfluxDB Output Node" auf die Arbeitsfläche ziehen. Das ist der zweite Knoten von unten im Be-

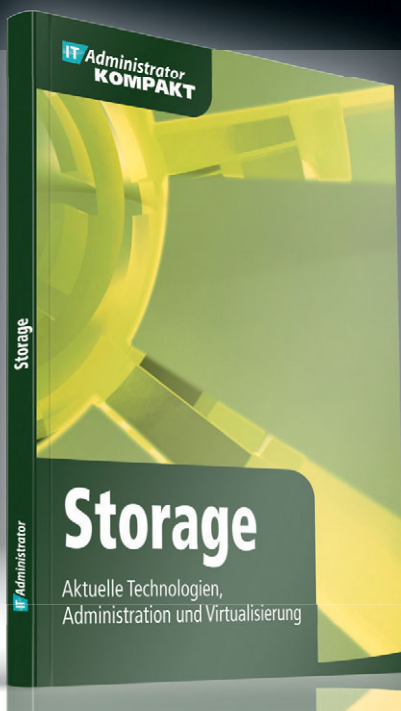
reich "Speicher". Auch diesen konfigurieren Sie per Doppelklick und fügen wiederum über das "Stift"-Icon einen neuen Server hinzu. Ins Feld "Host" tragen Sie die lokale Adresse 127.0.0.1 des Raspberry Pi ein und ins Feld "Database" den Namen der zuvor erzeugten Datenbank "flummi".

Zurück in den Eigenschaften des Knotens ist noch das Feld "Measurement" zu füllen. Ein Measurement innerhalb einer InfluxDB ist entfernt vergleichbar mit einer Tabelle in einer klassischen relationalen Datenbank und nimmt zusammengehörige Werte auf. Entsprechend tragen Sie dort den Namen des Sensor-Knotens "flummi42" ein. Speichern Sie die Eigenschaften und verbinden Sie die Ausgänge beider MQTT-Knoten mit dem Eingang des Influx-Knotens.

Input-Daten per JavaScript aufbereiten

Zu beachten ist noch, dass die Datenbank die rohen Daten des Sensors ohne Weiteres nicht verarbeiten kann. Sie müssen die Daten hierfür erst noch vorbereiten. Das erreichen Sie, indem Sie jeweils einen

IT Administrator KOMPAKT



Hier werden Sie zum Storage-Experten!

Verlieren Sie in Zeiten rasant wachsender Datenmengen nicht den Überblick – mit dem neuen Buch "Storage" widmet sich IT-Administrator Trendthemen wie Flash-Speicher, Storage-Virtualisierung, SAN-Management und softwaredefiniertem Speicher. Es bündelt die wichtigsten im Heft zum Thema Storage erschienenen Artikel der letzten zwei Jahre und bietet einen Überblick zu aktuellen Speichertechnologien und der Verwaltung im laufenden Betrieb.

Sichern Sie sich jetzt praxiserprobtes Experten-Know-how zum Thema Storage.

Für Abonnenten zum Vorzugspreis* von nur 12,90 €!

Abo- und Leserservice
IT-Administrator

vertriebsunion meynen
Herr Stephan Orgel
D-65341 Eltville

Tel: 06123/9238-251
Fax: 06123/9238-252
leserservice@it-administrator.de

* Nichtabonnenten erhalten das Buch für 14,90 €. Die Preise verstehen sich inklusive 7% Mehrwertsteuer und Versandkosten im Inland.

shop.heinemann-verlag.de

Knoten vom Typ "Funktion / Function" auf die Verbindung von MQTT- und InfluxDB-Knoten ziehen. Sobald das Frontend die Verbindung gestrichelt anzeigt, lassen Sie den Funktionsknoten los. Dieser wird dann automatisch in den Workflow eingefügt und ebenfalls per Doppelklick konfiguriert. Der Knoten bekommt einen beliebigen Namen und nimmt im Funktionsblock JavaScript-Code entgegen, der den per MQTT empfangenen Input für die Datenbank aufbereitet. Tragen Sie dazu für die Temperatur den folgenden Code ein:

```
value=parseFloat(msg.payload);
msg.payload=[{
  temperature: value
}];
return msg;
```

Für die Luftfeuchtigkeit ändert sich lediglich die dritte Zeile der Funktion ("humidity: value"). Die Funktionen lesen jeweils den im MQTT-Topic enthaltenen Fließkommawert aus, ordnen diesen dem gewünschten Feld in der Datenbank zu und reichen diese modifizierte Nachricht dann an die Datenbank weiter. Speichern Sie das Ganze wiederum per Klick auf "Implementieren". Die Datenverarbeitung in Node-RED ist damit komplett (Bild 2).

Überzeugen Sie sich in der Shell des Raspberry Pi davon, dass die Messwerte tatsächlich in der InfluxDB ankommen:

```
sudo influx -precision rfc3339
use flummi
select * from flummi42
```

Sie sollten als Output eine Liste mit drei Spalten für den Zeitstempel sowie Messwerte für Temperatur und Luftfeuchte zum jeweiligen Zeitpunkt erhalten.

Gesammelte Daten visualisieren

Mit den optional zu installierenden Dashboards bietet Node-RED selbst grundlegende grafische Darstellungen. Wir wollen stattdessen die wesentlich mächtigeren Visualisierungen von Grafana verwenden. Sie laden und installieren die Software per

```
sudo curl -sL https://packages.
grafana.com/gpg.key | sudo apt-key
add -
```

```
sudo echo "deb https://packages.
grafana.com/oss/deb stable main" |
sudo tee /etc/apt/sources.list.
d/grafana.list
```

```
sudo apt-get update
```

```
sudo apt-get install grafana
```

Zu guter Letzt richten Sie auch Grafana für den automatischen Start ein und führen den Server aus mittels:

```
sudo systemctl daemon-reload
```

```
sudo systemctl enable grafana-server
```

```
sudo systemctl start grafana-server
```

Anschließend öffnen Sie das Webfrontend von Grafana im Browser über die URL "http://IP-Adresse-des-Raspis:3000" mit dem Benutzernamen "admin" und dem gleichlautenden Passwort. Grafana fordert Sie daraufhin auf, das Passwort zu ändern, und führt Sie dann per Assistent durch die weitere Einrichtung. Der erste Schritt "Add data source" bietet diverse SQL- und NoSQL-Datenbanken zur Auswahl. Wir entscheiden uns natürlich für InfluxDB. Auf der folgenden Seite bestätigen Sie die vorgeschlagene URL "http://localhost:8086", tragen im Feld "Database" den Namen der Datenbank ("flummi") ein und finalisieren das Ganze über den Button "Save & Test". Grafana sollte dies mit "Data source is working" quittieren.

Datenbank abfragen

Zurück auf der Startseite ist der nächste Schritt "New Dashboard", gefolgt von "Add Query". Im oberen Bereich der nun folgenden Seite sehen Sie einen leeren Graphen, im Kontext von Grafana als "Panel" bezeichnet. Darunter finden Sie einen Editor, der Ihnen dabei hilft, ein Statement für Datenbank-Abfragen zusammenzubauen. Sie werden feststellen, dass die Syntax herkömmlichen SQL-Abfragen ganz ähnlich ist. Wenn Sie die Bestandteile der Abfrage anklicken, schlägt das Frontend Ihnen die möglichen Optionen vor.

Im ersten Panel wollen wir die Temperaturwerte unseres Sensors anzeigen. Klicken Sie dazu in der "From"-Zeile auf "select measurement" und wählen Sie die einzige Option "flummi42". Die "Where"-Funktion darf leer bleiben. In der "Select"-Zeile darunter klicken Sie auf "field(value)" und wählen die "temperature". Die Aggregat-Funktion "mean()" ist als Standard ausgewählt und wir belassen es dabei. Alternativ dazu sind viele weitere mathematische Funktionen verfügbar, die Sie über das "+"-Symbol am Ende der Zeile erreichen.

Die nächste Zeile "Group by" legt den Zeitraum für die Aggregat-Funktion "mean()" fest. Die "time"-Funktion ist hier mit dem variablen Zeitraum "\$_interval" vorbelegt. Grafana wählt damit das Zeitintervall dynamisch passend zum Anzeigefenster, damit es nicht mehr Werte aus der Datenbank abfragen muss, als angezeigt werden können. Letzteres würde die Anzeige des Graphen unnötig verlangsamen. Die Stärke von Grafana ist es aber gerade, stets Diagramme in sehr kurzer Zeit zu zeichnen.

Die "fill"-Funktion regelt den Umgang mit Messunterbrechungen. Wählen Sie hier "fill(null)", erkennen Sie am Graphen sofort, wenn Messwerte fehlen. Dafür

Link-Codes

- [1] Eclipse Mosquitto
j9p61
- [2] Von Sensordaten zu nutzbaren Informationen
j9p62
- [3] IoT-Anbindung an die Amazon-Cloud
j9p63
- [4] Node-RED
j9p64
- [5] InfluxDB
h1z61
- [6] Grafana
j6zaa
- [7] Raspbian Lite
j9p65
- [8] Win32 Disk Imager
j9p66
- [9] Grafana SMTP
j9p67

müssen Sie unterhalb des Abfrage-Editors im Feld "Min time interval" einen Wert eintragen, der größer ist als das Intervall, innerhalb dessen Sie Messwerte erwarten, im vorliegenden Fall also ">10s".

Während Sie die Optionen gewählt haben, hat das Frontend den Graphen darüber bereits dynamisch angepasst und Sie sehen Ihre Messwerte. Geben Sie dem Graphen noch im "Alias by"-Feld einen Namen, wählen Sie oben rechts auf der Seite den gewünschten Zeitraum zur Darstellung und stellen Sie die automatische Aktualisierung ein (Bild 3). Mit dem "Stift"-Symbol oberhalb des Editors können Sie übrigens vom grafisch geführten Aufbau der Statements auch in einen reinen Text-Modus umschalten. Dort sollte nun stehen:

```
SELECT mean("temperature") FROM
  "flummi42" WHERE $timeFilter GROUP
  BY time($__interval) fill(null)
```

Damit ist das Grundgerüst für Ihr erstes Diagramm bereits fertig. Über die Icons am linken Bildrand kümmern Sie sich nun noch um den Feinschliff für das Dia-

gramm. Wechseln Sie zum Bereich "Visualization". Dort können Sie verschiedene Optionen zur Darstellung des Graphen wählen oder im Bereich "Axes" die Einheit für die Y-Achse festlegen.

Das Icon "General" gibt dem Panel einen Namen und eine Beschreibung. Zu guter Letzt legen Sie mit den Optionen hinter dem Icon "Alert" Schwellenwerte für eine Alarmierung fest. Werden diese überschritten, hebt Grafana dies im Diagramm hervor und kann auch Benachrichtigungen per E-Mail verschicken – eine passende SMTP-Konfiguration vorausgesetzt [9].

Über das "Disketten"-Symbol zuoberst auf der Seite speichern Sie Ihr Werk und können dem Dashboard nun weitere Panels hinzufügen, indem Sie das Ganze für die Luftfeuchte des Sensors wiederholen. Grafana ist jedoch längst nicht auf Graphen beschränkt, sondern kann auch Anzeigen für den jeweils aktuellen Messwert erzeugen, die sich abhängig von Schwellwerten einfärben, um optisch auf Fehlerzustände hinzuweisen. Hier bietet die Software viel Freiraum zum Experimentieren.

Fazit

Mit unserem Workshop haben wir Ihnen einen Überblick über die Architektur von Plattformen für das Internet der Dinge und eine mögliche Implementierung verschafft. Dabei haben wir mehrere Tools kennengelernt, die auf Grund ihrer Komplexität jeweils einen eigenen Workshop verdient hätten. Hier konnten wir nur an der Oberfläche kratzen. Für weitere Ausflüge ins IoT sei darauf hingewiesen, dass jedes dieser Werkzeuge austauschbar ist und unsere Plattform natürlich nur eine kleine Lernumgebung darstellt.

Für dauerhaft produktiven Betrieb ist der verwendete Raspberry Pi schon wegen der SD-Karte als Speicher eher nicht geeignet. Auch die Informationssicherheit bietet ein – notwendiges – Feld zur Weiterentwicklung. So sollten Sie in der Praxis auf Verfügbarkeit bei der Hardware, sichere Protokolle sowie Authentifizierung und Autorisierung an den Servern setzen. Der Grundstein hierfür ist nun gelegt, sodass Sie die im IoT relevanten Technologien einschätzen und auf Kundenwünsche reagieren können. (jp) 



„Wie mache ich
meine Mitarbeiter
fit für IT-Sicherheit?“

> Christian Honner, 34,
Abteilungsleiter IT

it sa 2019
Die IT-Security Messe und Kongress

HOME OF IT SECURITY

Lösungen haben eine Plattform

Auf der international führenden Fachmesse für IT-Security erfahren Sie alles über die aktuellsten Sicherheitsstandards. **Sichern Sie sich Ihr Gratis-Ticket zur it-sa 2019!**



Nürnberg, Germany | 8.-10. Oktober 2019

it-sa.de/it-sicherheit4U

NÜRNBERG MESSE