

# IoT als Enabler für ein interaktives Schaufenster

Am Standort Brugg/Windisch betreibt die FHNW ein Maker Studio mit einem Schaufenster zur Präsentation von Wechselausstellungen. Um Passanten in die Wechselausstellungen einbeziehen zu können, sind lichtempfindliche Schalter am Schaufenster angebracht, mit denen die interaktiven Elemente einer Ausstellung gesteuert werden können. Diese Schalter und anderen interaktiven Elemente sind mit Technologien des Internet of Things (IoT) realisiert worden. Unter IoT wird die virtuelle Repräsentation realer, physischer Objekte verstanden, die untereinander mithilfe der Internet-Technologien kommunizieren können.

Jürg Luthiger, Robin Schoch, Christoph Stamm | juerg.luthiger@fhnw.ch

Die FHNW betreibt am Standort Brugg/Windisch ein *Maker Studio* [MakStu]. Das Maker Studio stellt Studierenden, Mitarbeitenden der FHNW und der Öffentlichkeit eine Infrastruktur aus traditionellen und digitalen Werkzeugen zur Verfügung, um kreative Ideen selber umsetzen zu können. Es ist davon auszugehen, dass im Kontext des Maker Studios spannende Projekte entstehen werden. Für diese Projekte ist eine Plattform geschaffen worden, um die Projekte auszustellen und einer breiten Öffentlichkeit präsentieren zu können.

Die Ausstellungen sollen einerseits die Essenz der Maker Idee [Mak] wiedergeben und andererseits das Publikum zur Interaktion animieren. In einem Schaufenster des Maker Studios kann jeweils eine Ausstellung präsentiert werden.

Das in Abbildung 1 abgebildete Schaufenster soll mit maximal zwanzig interaktiven Elementen ausgerüstet werden, um vor allem ein jüngeres Publikum anzusprechen und ihnen auf spielerische Art und Weise einen Einblick in die Möglichkeiten der Maker Kultur geben. Da über das Schau-

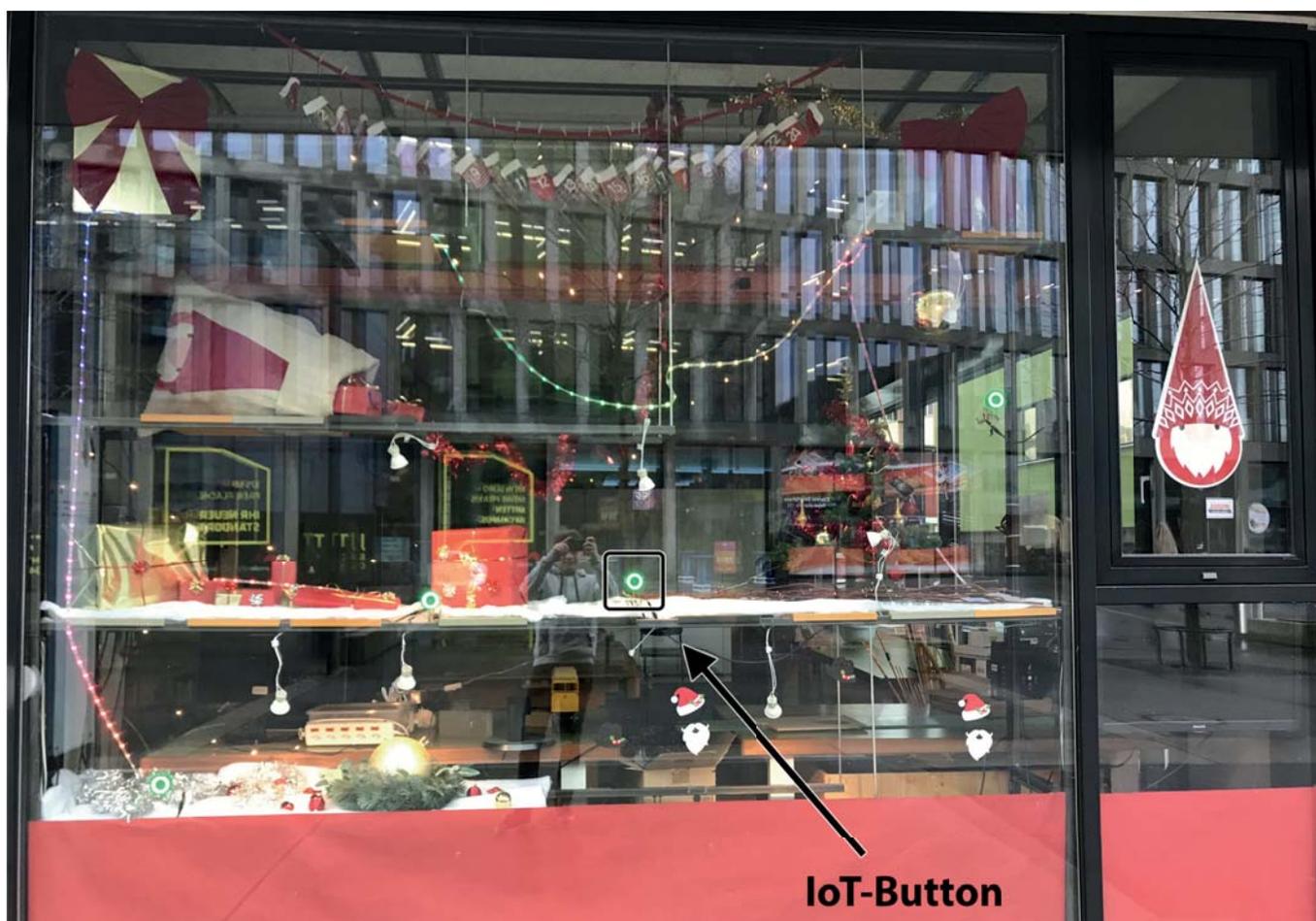


Abbildung 1: Schaufenster des Maker Studios in Brugg/Windisch

fenster wechselnde Ausstellungen gezeigt werden sollen, muss die Infrastruktur des Schaufensters dynamisch und flexibel gestaltet werden, um den Konfigurationsaufwand bei einem Ausstellungswechsel minimal halten zu können. Das gilt auch für die Verkabelung. In Anbetracht der Vielzahl der Geräte soll die Kommunikationsinfrastruktur drahtlos sein, um die Verkabelung auf ein Minimum reduzieren zu können.

Eine Wechselausstellung wird verschiedene Objekte umfassen, die mehrheitlich animiert werden können. Diese Animationen sollen aber erst durch eine Benutzerinteraktion ausgelöst werden, so dass die Benutzer in die Ausstellung involviert werden. Ebenfalls soll es möglich sein, zusätzliche Informationen zu den Objekten und zur aktuellen Ausstellung anzeigen zu können.

### Das Schaufenster

Das zur Verfügung stehende Schaufenster gehört zum Werkraum, der mit traditionellen Werkmaschinen ausgerüstet ist. Das Schaufenster öffnet sich auf einen grossen Platz, wo die Passanten zirkulieren. Es zeigt in Richtung Osten, so dass der Lichteinfall am Vormittag und am Nachmittag sehr unterschiedlich sein kann. Die Werkstatt wird regelmässig benutzt und deshalb ist der natürliche Lichteinfall zu gewährleisten. Ebenfalls sind Installationen ausserhalb des Werkraums nicht möglich.

Um über das Schaufenster mit der Ausstellung interagieren zu können, ist eine geeignete Sensorik notwendig, die kostengünstig, flexibel und robust ist. Verschiedene Sensor-Technologien wie Kamera oder Näherungssensor sind evaluiert worden, aber wegen Kostenfaktor/Privatsphäre (Kamera) oder wegen hoher Signaldämpfung der Fensterfläche (Näherungssensor) wieder verworfen worden. Der Lichtsensor ist die einzige valable Alternative geblieben.

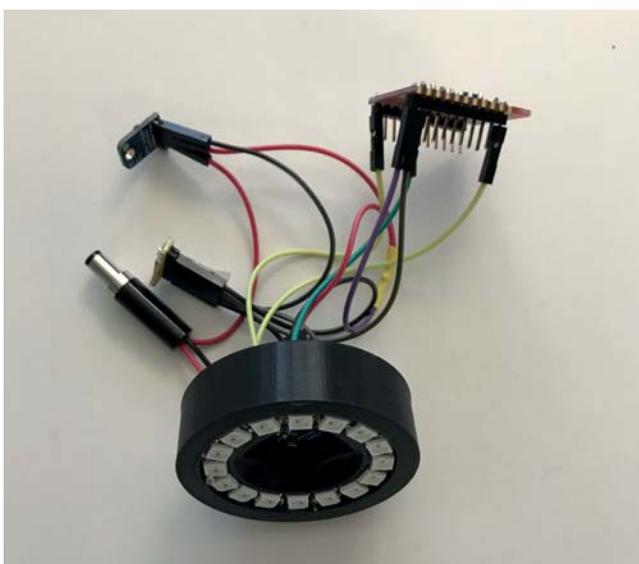


Abbildung 2: IoT-Button mit Lichtsensor, LED-Ring und ESP8266 mit WLAN-Modul

Der Lichtsensor kann auf einen schnellen Lichtwechsel reagieren, z.B. durch eine Wischbewegung über den Sensor, und ein Ereignis auslösen. Dieses Ereignis soll durch unterschiedliche Parteien verarbeitet werden können, um z.B. eine Lichtquelle einzuschalten, eine Präsentation zu starten, ein Objekt in Bewegung zu setzen oder dem Benutzer ein Feedback zu geben.

Als Hardware-Plattform verwenden wir im Schaufenster ein Raspberry Pi Model 3 [RPI3]. Mit dem darin integrierten WLAN-Modul kann diese Plattform auch die Funktion des Wireless Access Point übernehmen.

### Der Lichtsensor als IoT-Button

Ausgehend von einem Lichtsensor (TSL2561) [TSL], einem LED-Ring, einem Stromwandler und einem Mikrocontroller mit integriertem WLAN-Modul haben wir einen IoT-Button entwickelt, welcher als Interaktionselement im Schaufenster verbaut wird (siehe Abb. 2). Der Mikrocontroller ESP8266 ist ein günstiger, weit verbreiteter 32-bit Mikrocontroller mit integriertem WLAN-Modul und genügend Speicher, um auch die IoT-Button-Applikation aufnehmen zu können [ESP]. Alle diese Komponenten sind in ein selbstkonstruiertes und mit einem 3D-Drucker gefertigtes Gehäuse eingebaut worden.

Der Mikrocontroller liest ununterbrochen die gemessenen Beleuchtungsstärken (Lux-Werte) des Lichtsensors aus. Eine erste Testphase hat gezeigt, dass die Detektion eines einfachen Intensitätssprunges des Lux-Wertes nicht robust genug ist, da sich die Lichtverhältnisse vor dem Schaufenster auch ohne Benutzerinteraktion rasch ändern können, denn das Schaufenster öffnet sich zu einem Platz hin mit einer stark wechselnden Sonneneinstrahlung. Eine Wischbewegung über den Lichtsensor hingegen führt zu einem detektierbaren Signal, welches als kurzer Ausschlag (Spitze) sichtbar wird (siehe Abb. 3). Die Richtung des Ausschlages ist dabei irrelevant. Messungen mit einem ersten Sensor haben ergeben, dass die Dauer des Ausschlages im Bereich zwischen 0.1 und 1.5 Sekunden liegen kann. Für die Detektion werden die Lux-Werte ständig über die letzten hundert Messwerte gemittelt. Ausgehend vom Mittelwert  $m$  werden zwei Schwellwerte festgelegt:  $m + m_t$ . Sobald das Lichtsignal einen dieser beiden Schwellwerte über- bzw. unterschreitet wird eine Zeitmessung ausgelöst, welche wieder gestoppt wird, sobald der Lux-Wert den Mittelwert  $m$  wieder erreicht. Falls die Ausschlagdauer im gewünschten Bereich liegt, wird ein entsprechendes Ereignis ausgelöst und an eine zentrale Controller-Einheit übermittelt. Diese Controller-Einheit koordiniert alle Ereignisse im Schaufenster.

Sobald der IoT-Button ein Ereignis ausgelöst hat, gibt er über einen LED-Ring ein entsprechendes Feedback zurück. Dieser LED-Ring besitzt 24

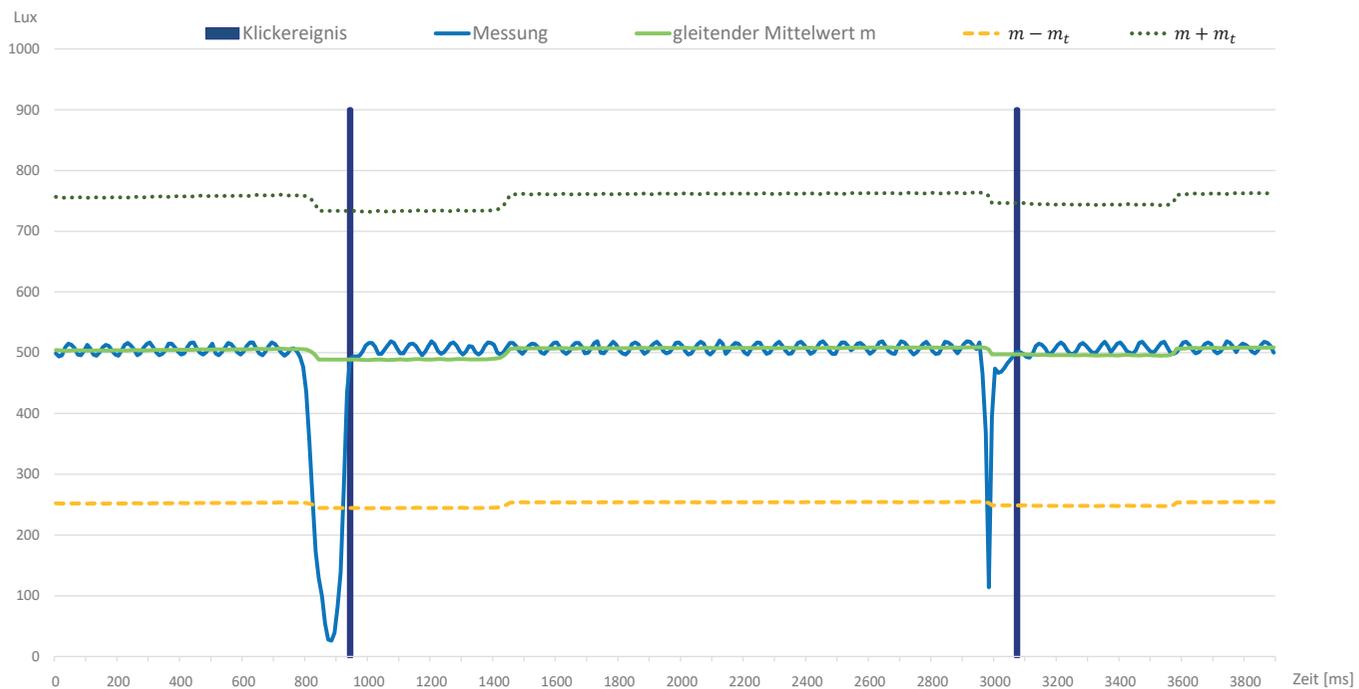


Abbildung 3: Lichtmessungen im Schaufenster. Die beiden senkrechten Balken zeigen ausgelöste Ereignisse an.

LEDs, die einzeln angesteuert werden können und mit denen unterschiedliche Feedbacks implementieren werden können. Der LED-Ring übernimmt aber noch eine weitere, nützliche Funktion: Er sorgt für eine konstante Lichtquelle, was vor allem bei schlechten Lichtverhältnissen am Abend und in der Nacht wichtig ist. Bei guten Lichtverhältnissen wird die Aktivierung des Sensors durch den Schattenwurf der Wischbewegung ausgelöst, bei schlechten Lichtverhältnissen hingegen durch die Reflektion des Lichtes des LED-Rings auf den Sensor.

### MQTT als Message Broker

Auf ein Ereignis eines Lichtsensors sollen unterschiedliche Reaktionen gleichzeitig ausgelöst werden können. Zum Beispiel soll eine Lampe eingeschaltet werden, um ein entsprechendes Ausstellungsobjekt zu beleuchten, oder es soll eine Präsentation gestartet werden, die zum Ausstellungsobjekte weitere Informationen liefert. Zwischen Ereignisquelle und Ereignisempfänger besteht eine 1-n Beziehung. Diese Beziehung ist in der IoT-Welt typisch. Deshalb gibt es auch geeignete Kommunikationsinfrastrukturen, die über das Publish-Subscribe-Pattern diese 1-n Beziehung zwischen Ereignisquelle und Ereignisempfänger abbilden. Ein bekannter Vertreter ist MQTT [MQTT], ein offenes Nachrichtenprotokoll für Machine-to-Machine (M2M) Kommunikation, das seit 2013 über OASIS [OASIS] standardisiert wird.

MQTT ist eine leichtgewichtige Publish-Subscribe-Lösung mit der Komponente „Topic“ als Schnittstelle zwischen Publisher und Subscriber.

Ein Topic lässt sich als schwarzes Brett mit eindeutiger Inventarnummer interpretieren, wobei sich alle anmelden müssen, die Informationen auf dem schwarzen Brett veröffentlichen wollen, ebenso diejenigen, die Nachrichten lesen wollen. Die Topics werden als hierarchische Baumstruktur aufgebaut, um eine eindeutige Identifikation zu ermöglichen.

Die zentrale Komponente von MQTT ist ein sogenannter MQTT-Broker, der die Topics inklusive der darin enthaltenen Nachrichten verwaltet, sowie den Zugriff auf die Topics regelt. Für Datensicherheit sind entsprechende Mechanismen vorhanden. Als MQTT-Broker wird Moquette eingesetzt [Moq]. Dies ist eine in Java implementierte Open Source Lösung.

### IoT-Button mit MQTT Technologie

Die hierarchische Struktur der Topics für das Schaufenster ist einfach gehalten. Für jeden IoT-Button existiert ein entsprechendes Topic „sensor/i“, wobei  $i$  die eindeutige Sensornummer ist. Jeder IoT-Button publiziert seine Button-Ereignisse auf seinem eigenen Topic in Form einer MQTT-Message. Aktoren, welche sich für IoT-Button-Ereignisse interessieren, schreiben sich bei den entsprechenden Topics als Empfänger ein. Damit ein IoT-Button selber ein Feedback für seinen LED-Ring erhalten kann, muss er sich auf seinem eigenen Topic auch als Empfänger einschreiben.

Andere Subscriber/Aktoren in unserem System sind die Lampen für die Beleuchtung der Ausstellungsgegenstände und der Informationsmonitor. Für die Beleuchtung verwenden wir Philips Hue-Lampen, da sie über das drahtlose Kommu-

nikationsprotokoll Zigbee kontrolliert werden können und viele Funktionalitäten zur Verfügung stellen, welche die Ausstellung bereichern und den heutigen Stand der IoT-Lampen darstellen [Hue, Zig]. Eine Philips Hue-Lampe verfügt über ein REST-API, welches über eine HTTP-Schnittstelle angesprochen werden kann. Diese Schnittstelle kann auch aus MQTT bedient werden, indem ein entsprechender Light-Service implementiert wird, der die Übersetzung zwischen MQTT und HTTP übernimmt. Der Light-Service meldet sich beim MQTT-Broker als Subscriber an und sobald er ein MQTT-Ereignis empfängt, sendet er einen entsprechenden HTTP-Request zur entsprechenden Philips Hue-Bridge, welche dann über Zigbee die dazugehörige Lampe schaltet. Die Einbindung von Hue-Lampen zeigt schön auf, wie bereits bestehende Systeme einfach in das Ökosystems des Schaufensters aufgenommen werden können.

Wie schon erwähnt, befindet sich im Schaufenster auch ein Informationsbildschirm mit zugehörigem Rechner (Mini-PC-System NC2000XA [XPC]). Darauf können Zusatzinformationen zur Ausstellung und den Objekten gezielt angezeigt werden. Dieser Informationsbildschirm ist ein weiterer Aktor. Er reagiert auf die Ereignisse der Lichtsensoren, um die entsprechende Diashow vorzubereiten und zu starten. Dazu wird die Software-Bibliothek Pygame verwendet [Pyg]. Pygame ist eine kleine Python-Bibliothek, welche den VLC-Player über den Simple DirectMedia Layer (SDL) steuern kann [SDL]. Damit ohne nennens-

werten Aufwand zwischen verschiedenen Diashows umgeschaltet werden kann, werden die Bilder und Videos in ein entsprechendes Directory hochgeladen und dort automatisch zu einem Film zusammengefügt und an das Format des Informationsbildschirmes angepasst. Die daraus entstandenen Videos werden dann mit einem VLC-Wrapper im VLC-Player abgespielt.

### Flexible Konfiguration einer Ausstellung

Für die Konfiguration der IoT-Buttons und Aktoren existieren drei weitere Topics: „discover“, „available“ und „update“. Die Aufteilung in drei Topics spart Ressourcen und vereinfacht die Logik der Nachrichtenbehandlung in den IoT-Buttons. Die Topics „discover“ und „update“ funktionieren ähnlich wie ein Broadcast: Alle Aktoren und Sensoren im System müssen sich zwingend bei diesen Topics einschreiben. Wenn das Schaufenster konfiguriert werden soll wird über das Topic „discover“ eine Nachricht zu jedem Gerät gesendet. Die Geräte melden sich daraufhin auf dem Topic „available“ unter Angabe ihrer ID.

Anschliessend an diese Initialisierungsphase kann das Schaufenster konfiguriert werden. Zur Konfiguration steht die in Abbildung 5 gezeigte Webapplikation zur Verfügung. Diese Applikation stellt alle im System bekannten Geräte grafisch dar. Aktoren und Sensoren lassen sich einfach zu dem entsprechenden Topic „sensor/i“ zuordnen. Sobald die manuelle Konfiguration abgeschlossen ist, generiert die Applikation eine Konfigurations-

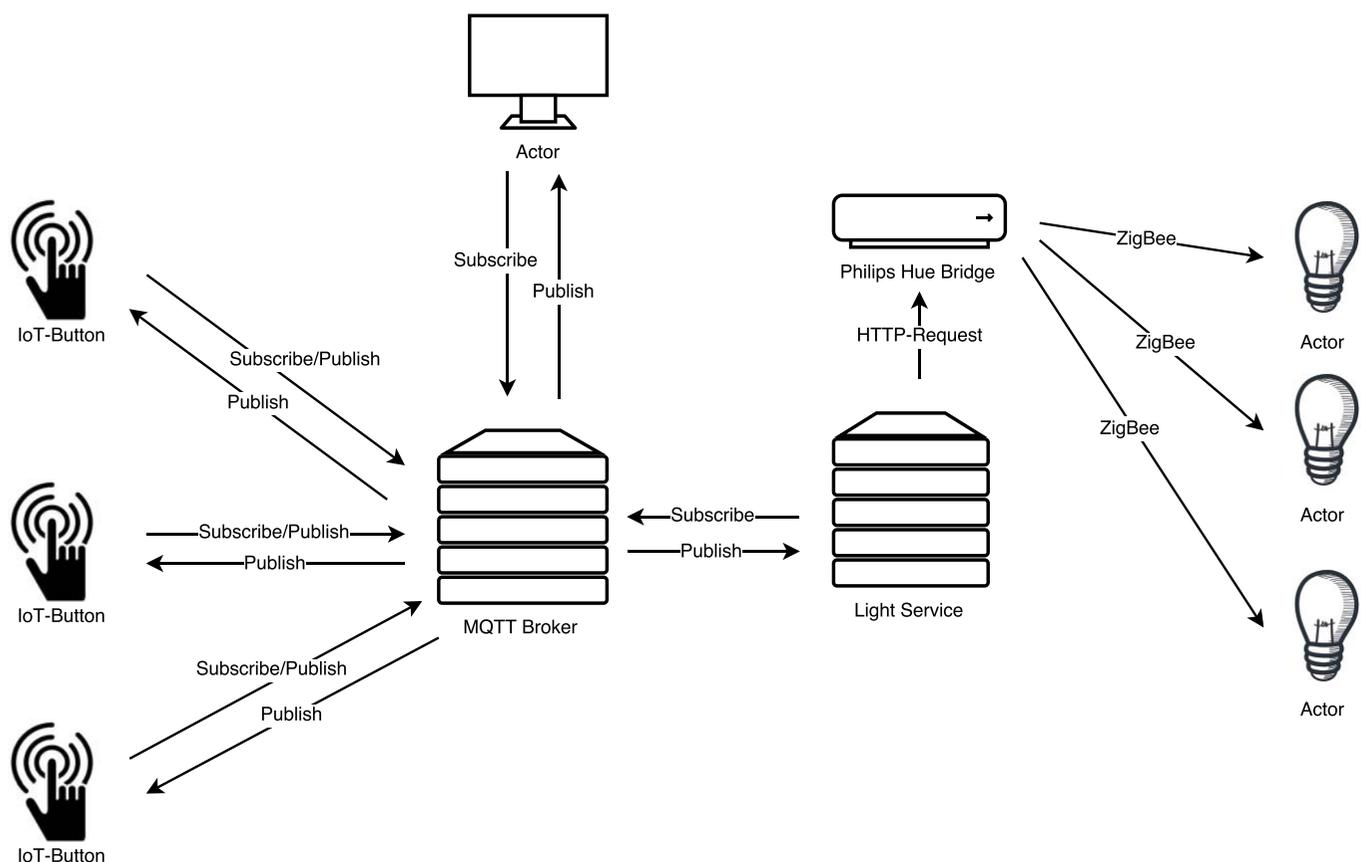


Abbildung 4: Konkreter Einsatz von MQTT mit IoT-Button als Publisher/Subscriber und Lampe/Bildschirm als Subscriber



Abbildung 5: Web-App zur Konfiguration: Lampen, Relais und Geräte können mit Schaltern (IoT-Button) verknüpft werden

datei im JSON-Format, welche dann unter dem Topic „update“ veröffentlicht wird, so dass jedes Gerät im System über das neue Konfigurationslayout informiert ist.

Die Webapplikation selber ist mit Node.js implementiert und unterstützt ein responsives Webdesign, so dass die Benutzerin die Konfiguration mit einem Tablet, Laptop oder sogar einem Smartphone direkt vor dem Schaufenster vornehmen kann [NJS].

### Zusammenfassung

Das Schaufenster bietet eine Plattform, um interaktive Ausstellungen präsentieren zu können. Die Interaktionselemente werden mit selbst entwickelten IoT-Buttons realisiert, die am Schaufenster angeklebt sind und die auf Änderungen im Lichtsignal reagieren. Die eigentlichen Ausstellungsgegenstände werden auf verstellbaren Tablaren angeordnet, durch IoT-Buttons gesteuert und mittels drahtlos angesteuerten Lampen beleuchtet. Über den Informationsbildschirm werden Bilder und Videos über die aktuelle Ausstellung präsentiert.

Eine selbst angefertigte Rahmenkonstruktion trägt die Tablare, die dynamisch in der Höhe und in der Länge angeordnet werden können. Die Kabelkanäle für die Stromversorgung verlaufen im Rahmen selbst und sind für den Betrachter unsichtbar.

Die Kommunikation baut auf MQTT auf. Detektierte Ereignisse der IoT-Buttons werden publiziert und an die entsprechenden Empfänger weitergeleitet, die anschliessend ihre Aktionen ausführen. Für Aktoren, welche nicht direkt ins System eingebunden werden können, weil ihnen die nötige MQTT-Schnittstelle fehlt, wie z.B. die

Philips Hue-Lampen, wird ein zusätzlicher Dienst angeboten, welcher die MQTT-Nachrichten empfängt, umwandelt und über eine entsprechende Schnittstelle an die Aktoren weiterleitet.

### Referenzen

- [ESP] Espressif ESP32 – A Different IoT Power & Performance: <https://espressif.com/en/products/hardware/esp32/overview>
- [Hue] Philips hue: <https://www2.meethue.com/de-de>
- [Mak] Maker: <https://de.wikipedia.org/wiki/Maker>
- [MakStu] Maker Studio: <https://web.fhnw.ch/plattformen/makerstudio>
- [Moq] Java MQTT lightweight broker: <https://github.com/andsel/moquette>
- [MQTT] MQTT: <http://mqtt.org>
- [NJS] Node.js: <https://nodejs.org/en>
- [OASIS] OASIS: [https://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=mqtt](https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=mqtt)
- [Pyg] Pygame: <https://www.pygame.org/news>
- [RPI3] Einplatinencomputer, Website Raspberry Pi: <https://www.raspberrypi.org>
- [SDL] Simple DirectMedia Layer: [https://de.wikipedia.org/wiki/Simple\\_DirectMedia\\_Layer](https://de.wikipedia.org/wiki/Simple_DirectMedia_Layer)
- [TSL] Adafruit TSL2561: <https://www.adafruit.com/product/439>
- [XPC] Shuttle XPC nano NC2000XA: <http://www.shuttle.eu/products/nano/nc2000xa>
- [Zig] Zigbee: <http://www.zigbee.org>