



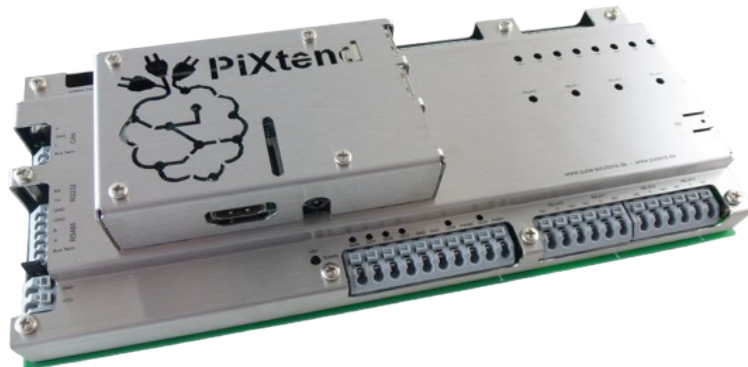
**PiXtend**

**Application-Note: MQTT mit PiXtend**

---

# Application-Note MQTT mit PiXtend

*Installation, Einrichtung, Programmierung*



---

**APP-PX-410**

---

---

**Stand 07.07.2017, V1.00**

---

Qube Solutions UG (haftungsbeschränkt)  
Arbachtalstr. 6, 72800 Eningen, Germany

<http://www.qube-solutions.de/>

<http://www.pixtend.de>



### Versionshistorie

Version	Beschreibung	Bearbeiter
1.00	Dokument erstellt	RT

### Inhaltsverzeichnis

1. Einleitung.....	3
1.1 Voraussetzungen.....	5
1.2 Haftungsausschluss.....	5
1.3 Sicherheitshinweise.....	5
2. MQTT Broker Einrichten.....	6
3. Installation MQTT für Python.....	7
4. Einrichtung des Python Beispiels.....	8
5. PiXtend per Smartphone und Tablet steuern.....	12
6. Frequently Asked Questions (FAQ).....	18



## 1. Einleitung

Wir freuen uns Ihnen eine weitere Anwendungsmöglichkeit der *PiXtend Python Library* präsentieren zu können und wie Sie Ihr *PiXtend* in ein IoT-Gerät (Internet of Things) verwandeln. Nutzen Sie das schlanke Nachrichtenprotokoll **MQTT**<sup>1</sup> um Daten mit dem Internet und anderen IoT-Geräten auszutauschen oder um Ihre Hausautomation von Unterwegs aus zu überwachen und zu steuern.



(Bildquelle: <https://www.python.org/>, The Python Brochure)



(Bildquelle: <http://mqtt.org/>)

Das **MQTT**-Protokoll wurde für die Maschine-zu-Maschine bzw. Internet of Things (IoT) Kommunikation entworfen. Bei der Entwicklung des Protokolls wurde großer Wert daraufgelegt, dass ein einfach zu implementierendes Protokoll entsteht und zudem ein möglichst geringes Datenvolumen beim Übertragen von Informationen benötigt. Das **MQTT**-Protokoll setzt dabei auf sog. *Publish*- und *Subscribe*-Mechanismen und ist somit bestens für kleine Geräte geeignet die über eine eingeschränkte Datenübertragungsbandbreite verfügen oder ihre Informationen nur gelegentlich per Modem übertragen können.

Das **MQTT**-Protokoll steht jedem frei zur Verfügung und wurde bereits auf vielen Plattformen umgesetzt und in vielen Programmiersprachen implementiert, wie z.B. *Python*.

Daher haben wir uns entschieden die bestehenden Beispiele der *PiXtend Python Library* (*PPL*) um ein **MQTT** Beispiel zu erweitern, da dieses Thema hervorragend zu *PiXtend* passt.

Das **MQTT**-Protokoll wurde von den Firmen *Eurotech* und *International Business Machines Corporation* (*IBM*) entwickelt und herausgebracht. Alle Rechte liegen bei diesen Autoren<sup>2</sup>.

<sup>1</sup> MQTT steht für MQ Telemetry Transport oder Message Queue Telemetry Transport

<sup>2</sup> <http://public.dhe.ibm.com/software/dw/webservices/ws-mqtt/mqtt-v3r1.html>



In dieser App-Note möchten wir Ihnen Schritt für Schritt aufzeigen wie schnell und einfach Sie mit der *PiXtend Python Library (PPL)*, dem *paho-mqtt* Python-Modul und Ihrem PiXtend-System online gehen und die IoT-Welt betreten bzw. Ihre Hausautomation über das Internet fernsteuern können.

Als Server (Broker) im Internet verwenden wir für das *Python MQTT* Beispiel den Dienst CloudMQTT. Der Anbieter hat die Internetadresse <https://www.cloudmqtt.com> und kann im Service-Level „Cute Cat“<sup>3</sup> kostenlos ausprobiert werden. Sie können gerne auch einen anderen *MQTT*-Dienst im Internet verwenden, unser Beispiel ist nicht auf diesen Dienst begrenzt.

Die Steuerung eines PiXtend-Boards vom Smartphone/Tablet oder Computer aus lässt sich mit einer der vielen kostenlosen *MQTT* Apps schnell und einfach realisieren.



Cute Cat

- 10 connections
- 10 Kbit/s

Try now for Free

(Bildquelle: <https://www.cloudmqtt.com>)

***Wir wünschen viel Spaß beim Testen, Programmieren und Experimentieren!***

Viele weitere Informationen, Tipps und Tricks finden Sie auch in unserem Support-Forum unter: <http://www.pixtend.de/forum/>

Die jeweils neusten Versionen aller Dokumente und Software-Komponenten finden Sie im Download-Bereich unserer Homepage: <http://www.pixtend.de/downloads/>

3 <https://www.cloudmqtt.com/plans.html>



### 1.1 Voraussetzungen

Das Beispiel wurde für Python 2.7.9 und später (nicht Python 3) entworfen und kann auf **PiXtend V1.2 und V1.3** ausgeführt werden. Sie können jedes PiXtend-System verwenden.

Auch gibt es keine spezielle Festlegung auf ein Raspberry Pi Modell. Wir empfehlen jedoch eines der folgenden Modelle: **B, B+, 2 B, 3 B**.

Laden Sie das **PiXtend Image - PiXtend Python Library** SD-Karten Image aus unserem [Download-Bereich](#) herunter und verwenden Sie dieses Image als Ausgangspunkt für das hier vorgestellte MQTT Python Beispiel. Alternativ können Sie auch ein original Raspbian Jessie Image verwenden, entsprechende Installationsschritte finden Sie in der App-Note zur [PiXtend Python Library \(PPL\)](#).

Auf den Raspberry Pi können Sie entweder mit direkt angeschlossener Tastatur und Monitor oder per SSH (TeraTerm / Putty) von einem PC aus zugreifen. Der Raspberry Pi benötigt zur Installation des *paho-mqtt* Python Moduls eine aktive Internetverbindung.

Wir empfehlen ferner die **App-Note** zur [PiXtend Python Library](#) zu konsultieren, insbesondere das Kapitel zur *Programmierung* wo gezeigt wird wie Sie die Beispiele der *PPL* ausführen können.

### 1.2 Haftungsausschluss

Qube Solutions UG kann nicht für etwaige Schäden verantwortlich gemacht werden die unter Umständen durch die Verwendung der zur Verfügung gestellten Software, Hardware, Treiber oder der hier beschriebenen Schritte oder Software von Dritttherstellern entstehen können.

### 1.3 Sicherheitshinweise



**PiXtend darf nicht in sicherheitskritischen Systemen eingesetzt werden.**

Prüfen Sie vor der Verwendung die Eignung von Raspberry Pi und PiXtend für Ihre Anwendung.



### 2. MQTT Broker Einrichten

Nach der Anmeldung bei [CloudMQTT](#) mit dem Service-Level „Cute Cat“, können Sie eine neue CloudMQTT Instanz (CloudMQTT Instance) anlegen und im Anschluss die Instanz-Infos (Instance info) aufschreiben bzw. ausdrucken.

Für die *MQTT* Python Demo sind folgende Punkte wichtig:

- Server
- User
- Password
- Port

Den SSL-Port bzw. den Websockets Port werden im Beispiel nicht verwendet.

#### CloudMQTT Console

[Overview](#)[Websocket UI](#)[Server log](#)[Statistics](#)[Restart](#)

#### Instance info

Server	mqtt.cloudmqtt.com
User	cloudmqtt
Password	cloudmqtt
Port	1883
SSL Port	8883
Websockets Port (TLS only)	8080
Connection limit	10



### 3. Installation MQTT für Python

Wurde eine SD-Karte entsprechend der [PiXtend Python Library](#) App-Note erstellt, kann jetzt das *paho-mqtt* Python Modul installiert werden.

Am schnellsten geht das mit *pip*, in dem Sie sich per SSH am Raspberry Pi einloggen und folgenden Befehl ausführen:

```
sudo pip install paho-mqtt
```

Bevorzugen Sie eine manuelle Installation oder konnte das Modul mittels *pip* nicht installiert werden, so können folgende Schritte durchgeführt werden um das *paho-mqtt* Python Modul manuell zu installieren:

```
cd ~  
git clone https://github.com/eclipse/paho.mqtt.python  
cd paho.mqtt.python  
sudo python setup.py install
```

Meldet *pip* oder das *setup.py* eine erfolgreiche Installation, kann von nun an das *paho-mqtt* Python Modul in eigenen Python Programmen genutzt werden.

Sind Sie nicht ganz sicher ob die Installation tatsächlich geklappt hat, verwenden Sie folgenden Befehl um per *pip* zu prüfen welche Module installiert sind und ob sich das *paho-mqtt* Modul darunter befindet:

```
pip freeze
```



### 4. Einrichtung des Python Beispiels

Ist das Python Modul *paho-mqtt* installiert, dann fehlt nur noch das *PiXtend MQTT Python Beispiel* damit es losgehen kann.

Über folgende Schritte kann das Beispiel von der PiXtend Homepage geladen werden oder Sie gehen zum PiXtend [Download-Bereich](#) und übertragen das Beispiel im Anschluss manuell auf den Raspberry Pi.

Folgende Schritte der Reihe nach ausführen:

```
cd ~  
cd ppl/examples/  
wget http://www.pixtend.de/files/downloads/px_mqtt_demo_v0.1.0.zip  
unzip px_mqtt_demo_v0.1.0.zip
```

Öffnen Sie das Python Programm *px\_mqtt\_demo.py* und führen Sie folgende Änderungen durch:

```
nano px_mqtt_demo.py
```

- In der Zeile mit ***client.username\_pw\_set*** das Wort ***USERNAME\_HERE*** durch den Benutzernamen der CloudMQTT Instance austauschen. Der Benutzername muss in Anführungszeichen stehen, da die Parameterübergabe als *String* erfolgt.
- In der Zeile mit ***client.username\_pw\_set*** das Wort ***PASSWORD\_HERE*** durch das Passwort der CloudMQTT Instance ersetzen. Das Passwort muss dabei in Anführungszeichen stehen, da die Parameterübergabe als *String* erfolgt.
- In der Zeile ***client.connect*** das Wort ***SERVER\_HERE*** ersetzen durch den Servernamen der CloudMQTT Instance, die Parameterübergabe erfolgt als *String*, deshalb muss der Servername in Anführungszeichen stehen.
- In der Zeile ***client.connect*** das Wort ***PORT\_HERE*** durch die Port-Nummer der CloudMQTT Instance ersetzen, hier erfolgt die Parameterübergabe direkt als Zahl.

Die Änderungen am Beispielprogramm speichern und schon kann es losgehen.







Gerne können Sie auch eigene *topics* vergeben, suchen Sie einfach im Programm nach *client.publish* und ändern Sie die *topics* nach Ihren Wünschen ab.

```
client.publish("/pixtend/digin0", p.digital_input0)
client.publish("/pixtend/digin1", p.digital_input1)
client.publish("/pixtend/digin2", p.digital_input2)
client.publish("/pixtend/digin3", p.digital_input3)
client.publish("/pixtend/digin4", p.digital_input4)
client.publish("/pixtend/digin5", p.digital_input5)
client.publish("/pixtend/digin6", p.digital_input6)
client.publish("/pixtend/digin7", p.digital_input7)
```

Damit auch etwas gesteuert werden kann, werden beim Verbindungsaufbau vom PiXtend zum CloudMQTT Broker vier *topics*, für alle vier Relais, *subscribed (abonniert)*, d.h. die Relais können über das *Websocket UI* oder einer anderen *MQTT Client* Anwendung gesteuert werden.

Die Relais stehen unter folgenden *topics* zur Verfügung:

- Relais 0 → /pixtend/rel0
- Relais 1 → /pixtend/rel1
- Relais 2 → /pixtend/rel2
- Relais 3 → /pixtend/rel3

Schreiben Sie auf eines oder alle 4 *topics* eine 1, dann schalten sich die Relais an bzw. schreiben Sie auf eines oder alle *topics* eine 0, dann gehen die Relais aus.

### Websocket

#### Send message

Topic

Message

Send

#### Received messages

Topic	Message
/pixtend/rel0	1

Relais 0 per Websocket einschalten

Digital Input

Relay 0:	1
Relay 1:	0
Relay 2:	0
Relay 3:	0



### Websocket

#### Send message

Topic

Message

Send

#### Received messages

Topic	Message
/pixtend/rel0	1
/pixtend/rel0	0

Relais 0 per Websocket ausschalten

Digital Input 0

Relay 0: 0

Relay 1: 0

Relay 2: 0

Relay 3: 0

In dieser App-Note und dem *MQTT* Python Beispiel zeigen wir die Verwendung der digitalen Eingänge und der Relais auf, Sie können jedoch alle Eigenschaften der PPL und von PiXtend zusammen mit *MQTT* nutzen, es gibt hier keine Einschränkung.

Sie müssen lediglich darauf achten, dass die *publish* Funktion des *paho-mqtt* Moduls nur die Daten Typen *String*, *Bytearray*, *Int* oder *Float* unterstützt. Zahlen werden automatisch in eine Zeichenkette (*String*) umgewandelt.

Die Veröffentlichung eines Temperaturwertes eines DHT11 Sensors könnte z.B. so aussehen:

```
client.publish("/pixtend/temp0", p.t0_dht11)
```

Fügen Sie zu Ihrer PiXtend Konfiguration im Python Programm den Eintrag *p.dht0 = p.ON* hinzu, damit der GPIO 0 in den DHT11/22 Mode geschaltet wird.

Die Eigenschaft *t0\_dht11* liefert den Temperaturwert als *float* Datentyp zurück und kann somit direkt verwendet werden. Haben Sie einen Wert der nicht unterstützt wird, wandeln Sie diesen vor der Übertragung um, damit es zu keinem Fehler kommt.

Den Datentyp zu jeder Eigenschaft der PPL finden Sie im *Anhang A* der App-Note zur *PiXtend Python Library*.



### 5. PiXtend per Smartphone und Tablet steuern

Eine weitere Möglichkeit das PiXtend mit *MQTT* abzufragen bzw. zu steuern, ist die Verwendung einer *MQTT* App auf einem Smartphone und/oder Tablet.

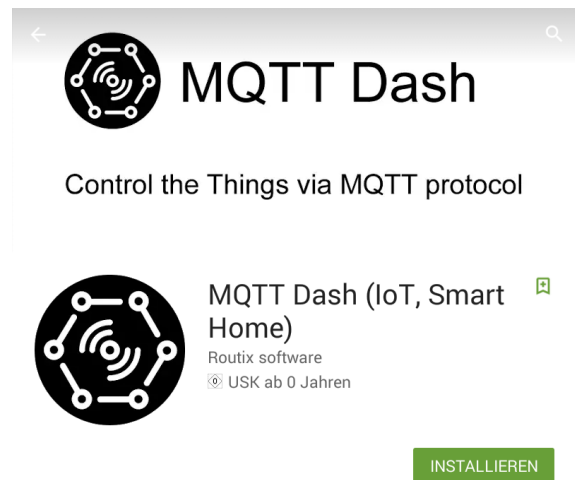
Wir haben uns als Beispiel die App "*MQTT Dash*" der Firma Routix software<sup>4</sup> im *Google Play Store*<sup>5</sup> herausgesucht, da diese App nach der Konfiguration eine graphische Oberfläche zur Bedienung bereitstellt. Viele andere *MQTT* Apps erfordern die Eingabe aller Informationen und Werte in Textform, dies kann recht umständlich werden, je nachdem welche Länge die verwendeten *topics* haben und welche Werte Sie verwenden möchten.

Die nachfolgenden Schritte zeigen die Konfiguration und Verwendung der *MQTT Dash* App auf einem Android Tablet.

#### 1. Installation der App

Suchen Sie im *Google Play Store* nach der App *MQTT Dash* und installieren Sie diese auf Ihrem Smartphone oder Tablet.

Zur Installation der App müssen Sie ggf. der Nutzung verschiedener Rechte und Dienste durch die App zustimmen. Ohne diese Erlaubnis kann die App nicht installiert werden oder funktioniert später nicht richtig.



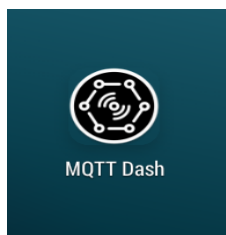
<sup>4</sup> <http://www.routix.net/>

<sup>5</sup> <https://play.google.com/store/apps/details?id=net.routix.mqttdash&hl=de>



### 2. Erster Start

Nach der Installation der App *MQTT Dash* sollte sich ein Symbol der App auf dem Desktop des Smartphone oder Tablets befinden. Öffnen Sie die App um mit der Konfiguration zu beginnen.

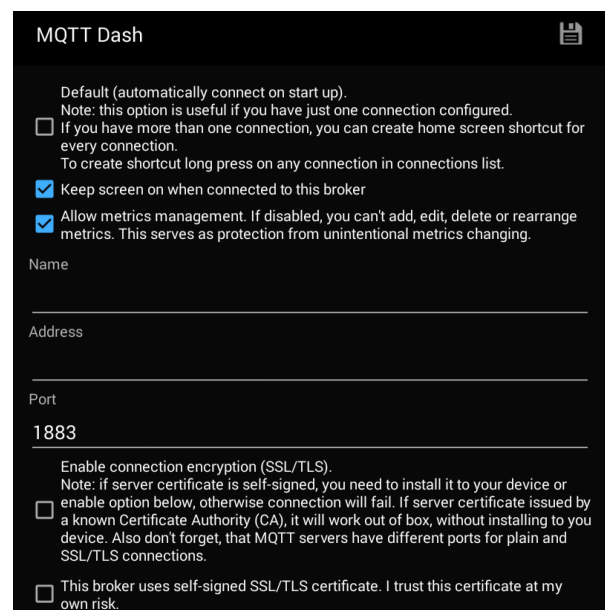


Nach dem ersten Start ist bis auf die Titel-Leiste der App, mit drei Schaltflächen, nichts weiter zu sehen. Mit einem Druck auf die *Plus-Schaltfläche* (+) wird eine neue Konfiguration bzw. Verbindung zu einem MQTT-Broker angelegt.

### 3. Konfiguration bzw. Verbindung anlegen

Im oberen Teil der Konfiguration müssen folgende Änderungen durchgeführt werden:

- Unter *Name* einen Namen (Bezeichnung) für diese Konfiguration angeben, z.B. *PiXtend MQTT*
- Bei *Address* muss die Adresse des CloudMQTT Servers eingetragen werden, diese finden Sie in der *CloudMQTT Instance info*
- Unter *Port* tragen Sie den *Port* aus der *Instance info* ein, nicht den *SSL Port* oder *Websockets Port (TLS only)* verwenden. Sie können diese Einstellungen später zu jeder Zeit ändern.





- Lassen Sie die SSL-Verbindung und die Zertifikatsbestätigung aus.
- Tragen Sie unter *User name* den Benutzernamen (*User*) der CloudMQTT Instance ein und unter *User password* das dazugehörige Passwort.
- Die *Client ID* können Sie für diesen Test so stehen lassen. Verwenden Sie mehrere Geräte, dann achten Sie darauf das es jede *Client ID* nur ein Mal gibt.
- Sind alle Informationen eingetragen, dann drücken Sie zum Speichern auf das *Disketten*-Symbol ganz rechts oben in der Ecke um die Konfiguration bzw. die Verbindungsdaten zum *MQTT*-Broker zu speichern.

Hat alles geklappt, so befinden Sie sich wieder auf der ersten Seite der App und der Name der angelegten Verbindung ist zu sehen.





### 4. Anzeige und Bedienung anlegen (Metrics)

Damit sich auch etwas tut bzw. damit Aktionen ausgelöst werden können, müssen in der *MQTT Dash* App sogenannte *Metrics (Kacheln)* angelegt werden. Nach einem Druck auf die *Plus*-Schaltfläche können Sie aus mehreren Typen auswählen was angezeigt werden soll bzw. ob eine Interaktion möglich ist.

Für das erste *Metric* wählen Sie aus der Typen-Liste den Punkt *Switch/button* aus. Dadurch können wir später das Relais 0 ein- und ausschalten.

- Unter *Name* einen Namen für dieses *Metric* vergeben, z.B. *Relay0*
- Unter *Topic (sub)* das gewählte *topic* eintragen. In unserem Beispiel ist das */pixtend/rel0*
- Alle anderen Punkte können unverändert bleiben. Wichtig ist der Zustand bzw. der Wert. Bei *On* muss *1* stehen und bei *Off* eine *0*.

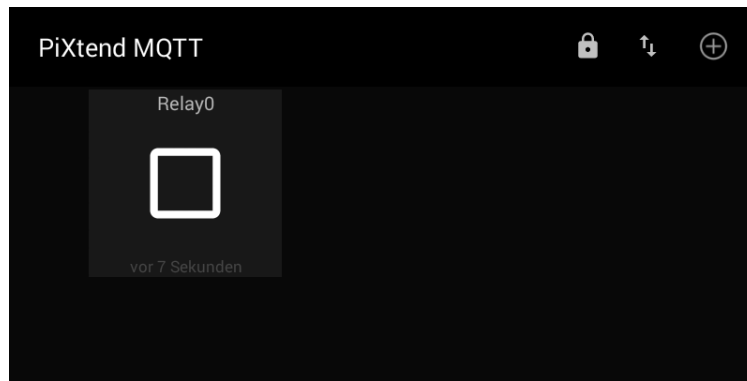
Über das *Disketten*-Symbol in der oberen rechten Ecke können die Einstellungen gespeichert werden.

The screenshot shows the 'MQTT Dash' app interface. At the top right is a diskette icon for saving. The main area has a dark background with white text. It starts with a description: 'This metric is intended for state displaying and switching (e.g. light on/off). Or it can behave as a simple static button. Payload is expected to be string.' Below this are input fields for 'Name' (containing 'Relay0') and 'Topic (sub)' (containing '/pixtend/rel0'). A link to JSON path documentation is provided. There are two checkboxes: 'Enable publishing' (checked) and 'Update metric on publish immediately (do not wait for incoming message to update visual state)' (checked). Below these are two columns for 'On' and 'Off' states. The 'On' state has a value of '1' and a checkmark icon. The 'Off' state has a value of '0' and an empty square icon. At the bottom, there are 'Other settings' with radio buttons for 'QoS(0)' (selected) and 'QoS(1)'.



### 5. Relais 0 ein- und ausschalten

Wurden die Einstellungen für das *Metric* gespeichert und die Verbindung zum *MQTT*-Broker steht, dann sollte sich folgendes Bild ergeben:

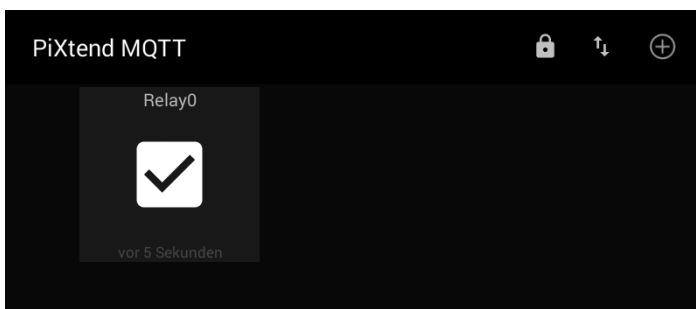


Das Element zeigt an, dass der Zustand für das *topic* /pixtend/rel0 (Relais 0) aus (0) ist. Unten im *Metric* wird die Zeit seit der letzten Aktualisierung angezeigt.



Sollte auf dem PiXtend das Python *MQTT* Beispiel noch nicht laufen, dann muss es jetzt gestartet werden, sonst kann keine Interaktion stattfinden. Sehen Sie im Kapitel 4 nach wie das Python *MQTT* Beispiel gestartet wird und die Funktion geprüft werden kann.

Mit einem Druck auf das *Metric* wird der Wert 1 auf das *topic* /pixtend/rel0 geschrieben und das Element zeigt einen Haken an. Das Relais 0 auf dem PiXtend sollte angehen und die Ausgabe des Python *MQTT* Beispiels auf der Console zeigt eine 1 an.



```
Digital Input 7: 0
Relay 0: 1
Relay 1: 0
Relay 2: 0
Relay 3: 0
```

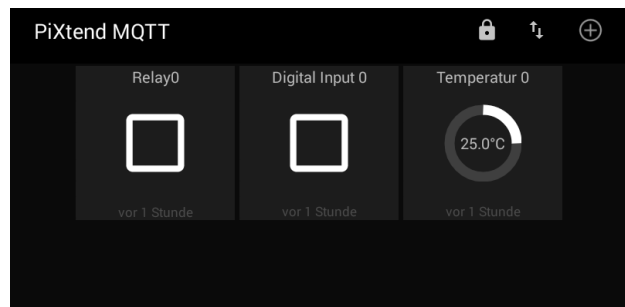
Ein erneuter Druck auf das *Metric* schaltet das Relais 0 wieder aus.





Sie können jetzt nach Belieben weitere Elemente hinzufügen, wenn Sie z.B. weitere Relais steuern wollen oder, wenn Sie einen DHT11 Sensor zur Hand haben, erweitern Sie das *MQTT* Python Beispiel so, dass es eine Temperatur überträgt und fügen Sie in der *MQTT Dash* App ein passendes *Metric* ein.

The screenshot shows the 'MQTT Dash' configuration screen. It includes a description of the metric, a name field set to 'Temperatur 0', a topic field set to '/pixtend/temp0', and a JSON path field. There are checkboxes for 'Enable publishing' and 'Display payload value instead of percentage' (which is checked). A progress bar is shown with a white color selection. At the bottom, there are 'Other settings' for QoS(0) and QoS(1).



*Glückwunsch!* Sie haben erfolgreich das Python *MQTT* Beispiel angewendet, ggf. sogar erweitert, und das PiXtend mit Ihrem Smartphone oder Tablet ferngesteuert.

Wir wünschen weiterhin viel Erfolg bei der Python-Programmierung und viel Spaß bei der Anwendung von *MQTT* zur Fernsteuerung Ihres PiXtend.



## 6. Frequently Asked Questions (FAQ)

**Muss ich unbedingt den Dienst *CloudMQTT* verwenden? Ich habe schon ein Konto bei einem anderen Broker.**

Sie können jeden anderen *MQTT* Broker im Internet verwenden, der Dienst *CloudMQTT* wird in diesem Dokument nur beispielhaft eingesetzt. Sie müssen jedoch darauf achten dass das Python Modul *paho-mqtt* auch die entsprechende *MQTT* Version Ihres Anbieters unterstützt. Zusätzlich müssen Sie die Informationen zu Server Adresse, Benutzername, Passwort und Port ermitteln.

**Ich möchte lieber eine verschlüsselte Verbindung nutzen und meine Daten sicher übertragen. Geht das auch?**

Grundsätzlich sollten Sie *MQTT* Kommunikation per *SSL* nutzen können. Wir haben dies selbst nicht getestet und können daher keine Aussage treffen ob dies sofort nach der Installation des *paho-mqtt* Moduls funktioniert oder ob Sie weitere Linux Packages installieren müssen um die *SSL*-Funktionalität zu erhalten.

Probieren Sie es am besten einfach aus. In jedem Fall müssen Sie das Python *MQTT* Beispiel von uns anpassen, so dass es eine verschlüsselte Verbindung benutzt.

**Muss ich unbedingt einen *MQTT* Dienst im Internet verwenden? Ich habe schon alle Geräte und Daten im WLAN.**

Natürlich können Sie sich auch einen eigenen *MQTT*-Server im eigenen Heimnetzwerk einrichten, so dass keine Daten ins Internet gelangen. Dieses Vorgehen würde jedoch den Umfang dieses Dokuments sprengen und ggf. sind noch viele weitere Schritte notwendig, damit man sich über das Internet ins eigene Heimnetzwerk einloggen kann.

Um diesen Teil so einfach wie möglich zu halten, haben wir uns entschieden einen Dienst im Internet zu verwenden.

Zum Einrichten eines eigenen Servers suchen Sie am besten im Internet nach: *MQTT Server*, *mosquitto* oder *Raspberry Pi MQTT Server*

Um nur ein paar Schlagwörter zu nennen.

Gerne möchten wir Sie für den Informationsaustausch in die Foren von Qube Solutions einladen:

<http://www.pixtend.de/forum/>