MQTT

Der Kleber, der das Internet of Things zusammenhält

Pi and More 6

Nico Maas

Zur Person

Nico Maas

IT Systemelektroniker (RZ UNI SB)

Bachelor of Science (HTW SB)

mail@nico-maas.de

www.nico-maas.de

@nmaas87

Agenda

- **I. Einführung**
  - MQTT
  - Modell
  - Topic
  - QoS
  - Security
- **II. Mosquitto**
  - Info
  - Installieren
  - Demo 1
  - Demo 2
- **III. Ausblick**
  - Ausblick
  - Installation Mosquitto 1.4
- **IV. Abschluss**
  - Fragen
  - Quellenangaben

I. Einführung

Redis

zeroMQ

P0

RabbitMQ

MQTT

Einführung
- MQTT
- Modell
- Topic
- QoS
- Security

Mosquitto

Abschluss

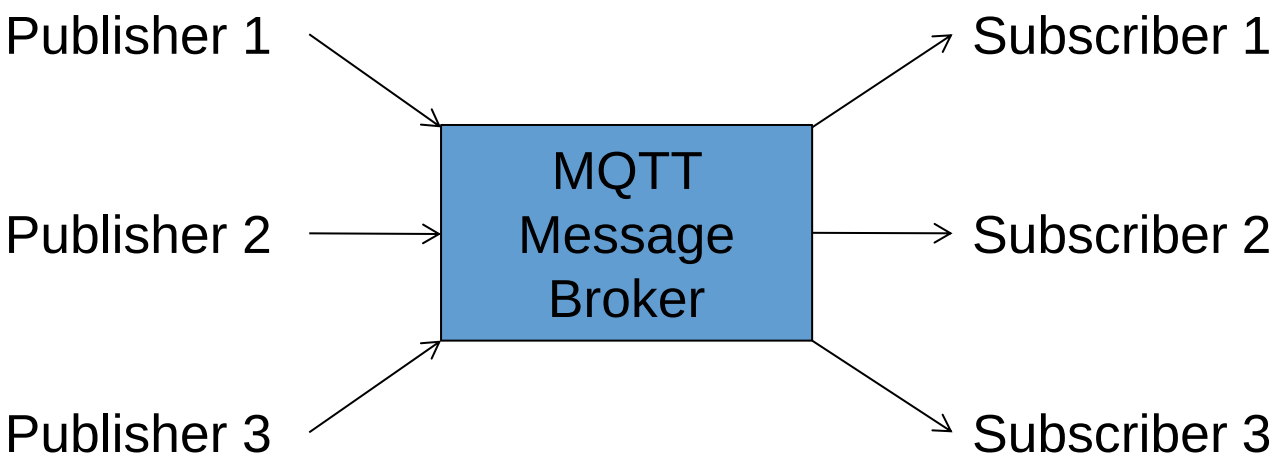
- Message Queue Telemetry Transport
- 1999 von Dr. Andy Standford-Clark (IBM) und Arlen Nipper (Arcom/Eurotech) entwickelt
- M2M (Machine To Machine) Protokoll
- OASIS Standard (07.11.2014 / Version 3.1.1)
- MQTT – TCP/IP
- MQTT-SN – non-TCP/IP Networks (ZigBee)
- „light weight“ (kleinstes MQTT Paket in 2 Bytes möglich)
- „small code foot print“
- Binär Format
- Payload als Byte Array
- Publisher / Subscriber Modell via Message Broker
- „Last will and testament“ – falls Client Offline geht
- Actionscript, Arduino, C, C++, Clojure, Dart, Delphi, Erlang, Go, Haskell, Java, Javascript / Node.js, LotusScript, Lua, mbed, .Net, Objective-C, Ocaml, Perl, PHP, Python, REXX, Ruby, Tcl, …

Modell

Einführung
- MQTT
- **Modell**
- Topic
- QoS
- Security

Mosquitto

Abschluss



Topic

Einführung
- MQTT
- Modell
- **Topic**
- QoS
- Security

Mosquitto

Abschluss

- Thema in welches ein Publisher Informationen ablegen bzw ein Subscriber von diesem beziehen kann
  - sensor/1/temperatur
  - sensor/1/luftfeuchtigkeit
  - sensor/1/strahlung/alpha
  - sensor/1/strahlung/beta
  - sensor/1/strahlung/gamma

- + Wildcard: Alle Werte auf dem gleichen Level
  - sensor/+/temperatur
    - sensor/1/temperatur
    - sensor/2/temperatur
    - sensor/3/temperatur

- # Wildcard: Alle Werte in allen nachfolgenden Leveln
  - sensor/1/strahlung/#
    - Werte von alpha, beta und gamma

QoS

Einführung
- MQTT
- Modell
- Topic
- **QoS**
- Security

Mosquitto

Abschluss

QoS	Bedeutung
0	„At most once“ – So sicher wie TCP/IP, Paket wird einmal losgeschickt und über TCP/IP übertragen. Ist der Subscriber nicht mehr online oder gibt es andere Probleme wird das Paket nicht neu übertragen. => Gut für häufig übermittelte Sensordaten
1	„At least once“ – Die Übermittelung der Nachrichten wird sichergestellt, allerdings könnten doppelte Zustellungen passieren.
2	„Exactly once“ – Stellt sicher dass die Nachricht zugestellt wird und zwar exakt einmal. => Gut für Abrechnungssysteme

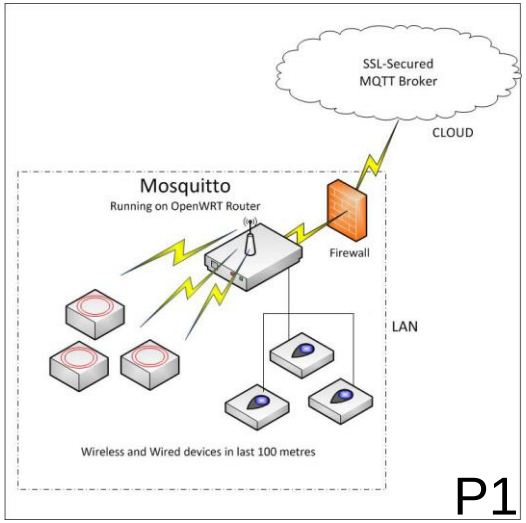
Security

Einführung
- MQTT
- Modell
- Topic
- QoS
- **Security**

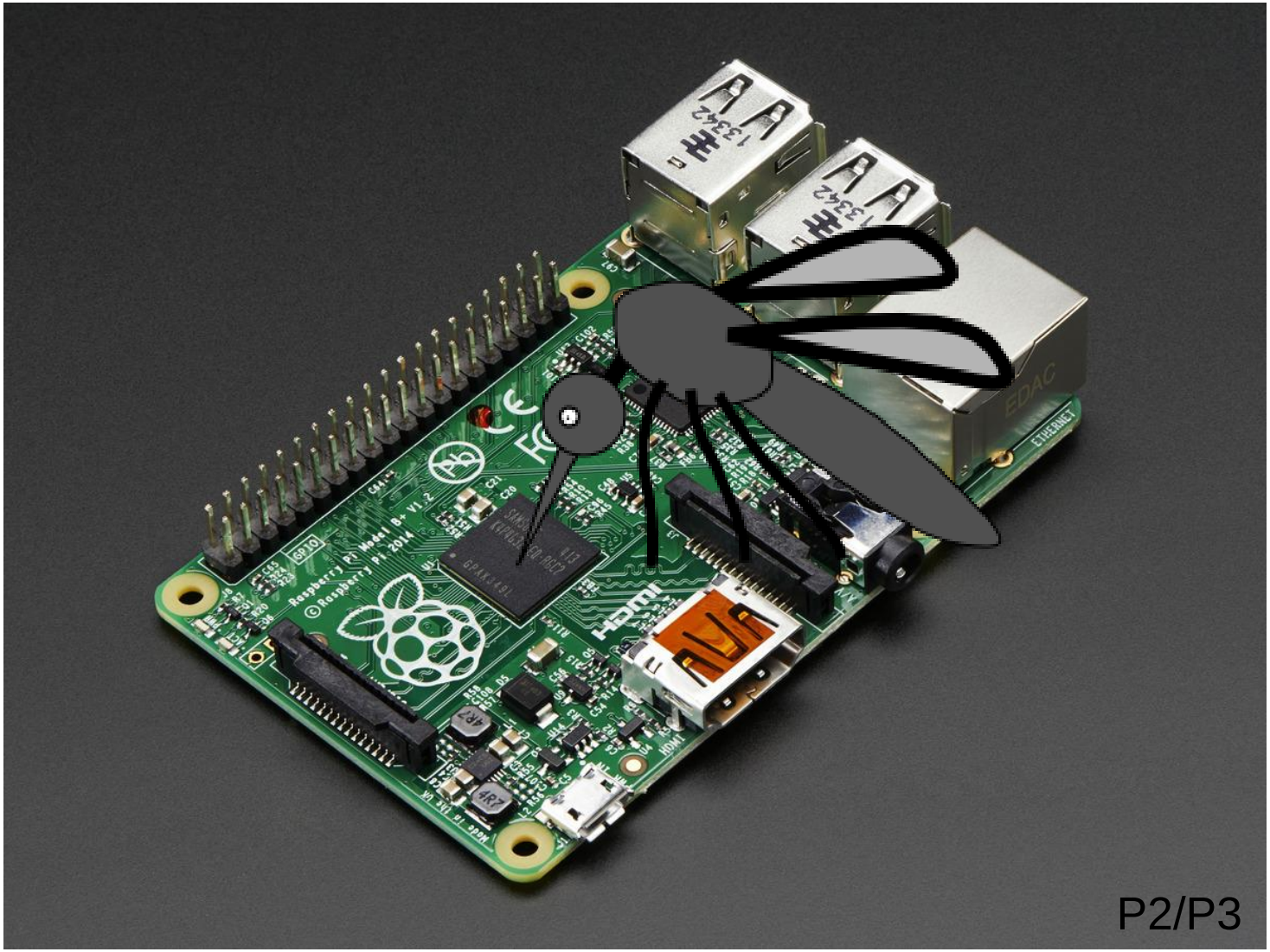
Mosquitto

Abschluss

- „light weight“
- Benutzername / Password
- ACL
  - Begrenzt Zugriff bestimmter Nutzer auf bestimmte Topics
  - Kann „nur lesen“ oder auch „nur schreiben“ erlauben
- SSL möglich
- Nutzung von VPN oder anderen, externen Möglichkeiten
- „Payload agnostic“ – Daten können unabhängig von MQTT durch die Anwendung verschlüsselt werden
- Broker2Broker Verbindung möglich per SSL / „MQTT Trunk“



II. Mosquitto



Info

Einführung

Mosquitto
- Info
- Installieren
- D1: Hello World
- D2: Temp Ctrl

Abschluss

- http://mosquitto.org/
- OpenSource
- MQTT v.3.1.1 kompatible
- C Code
- Windows, Mac, Arch Linux, CentOS, Debian, Fedora, FreeBSD, Gentoo, openSUSE, OpenWrt, Raspberry Pi, Redhat Enterprise Linux, Slackware, Ubuntu, QNX, iPhone, …
- Verfügt leider (noch) über keine Cluster Funktion…
- Dokumentation:
  - http://mosquitto.org/documentation/
- Allgemeine Infos:
  - www.mqtt.org
  - https://github.com/mqtt/mqtt.github.io/wiki

Installieren

Einführung

Mosquitto
- Info
- **Installieren**
- D1: Hello World
- D2: Temp Ctrl

Abschluss

- Externes Repo for Mosquitto hinzufügen (0.15-2 -> 1.3.5):
  curl -O http://repo.mosquitto.org/debian/mosquitto-repo.gpg.key
  sudo apt-key add mosquitto-repo.gpg.key
  rm mosquitto-repo.gpg.key
  cd /etc/apt/sources.list.d/
  sudo curl -O http://repo.mosquitto.org/debian/mosquitto-repo.list
- Paketlisten aktualisieren:
  sudo apt-get update
- Auf allen Geräten: Mosquitto Client / Python Library installieren:
  sudo apt-get install -y mosquitto-clients python-pip expect-dev
  sudo pip install paho-mqtt
- Zusätzlich nur auf Broker: Mosquitto Broker installieren:
  sudo apt-get install -y mosquitto
- Konfigurationsdateien liegen in /etc/mosquitto

Demo 1: Hello World

Einführung

Mosquitto
- Info
- Installieren
- **D1: Hello World**
- D2: Temp Ctrl

Abschluss

rpi-broker, SSH Session 1

```
# Starte Subscriber im Topic piandmore/test
mosquitto_sub -h 127.0.0.1 -t piandmore/test
```

rpi-broker, SSH Session 2

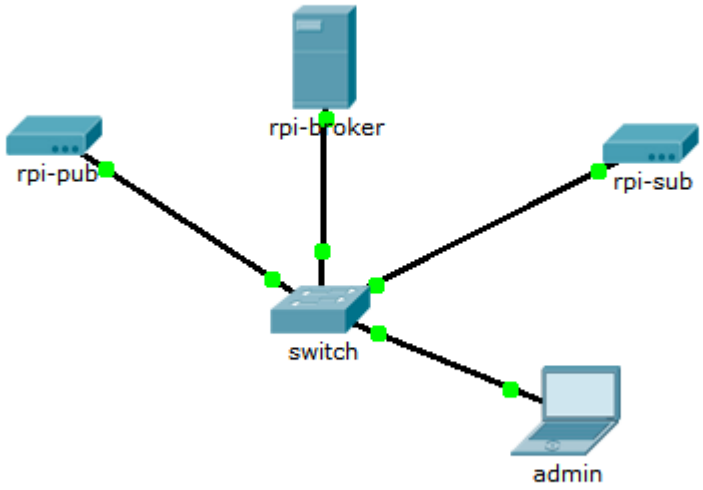
```
# Veröffentliche "Hello World" in den Topic piandmore/test auf dem
# lokal laufenden Broker
mosquitto_pub -h 127.0.0.1 -t piandmore/test -m "Hello World"
```

Demo 2: Temperature Control

Einführung

Mosquitto
- Info
- Installieren
- D1: Hello World
- **D2: Temp Ctrl**

Abschluss



Demo 2: Subscriber / Publisher

Einführung

Mosquitto
- Info
- Installieren
- D1: Hello World
- **D2: Temp Ctrl**

Abschluss

rpi-pub (Temperatursensor Träger)

```
# Starte Publisher in Topic piandmore/rpi00/tmp auf Broker,
# lese Eingaben von stdio
unbuffer ./lm75b.py \
 | mosquitto_pub -h 192.168.1.3 -t piandmore/rpi-pub/temp -l
```

rpi-broker

```
mosquitto_sub -h 127.0.0.1 -t piandmore/rpi-pub/temp
```

rpi-sub (LED Träger)

```
# Lese Werte aus dem Topic piandmore/rpi-sub/led auf Broker,
# schreibe auf LED
unbuffer mosquitto_sub -h 192.168.1.3 -t piandmore/rpi-sub/led \
 | sudo ./led
```

rpi-broker

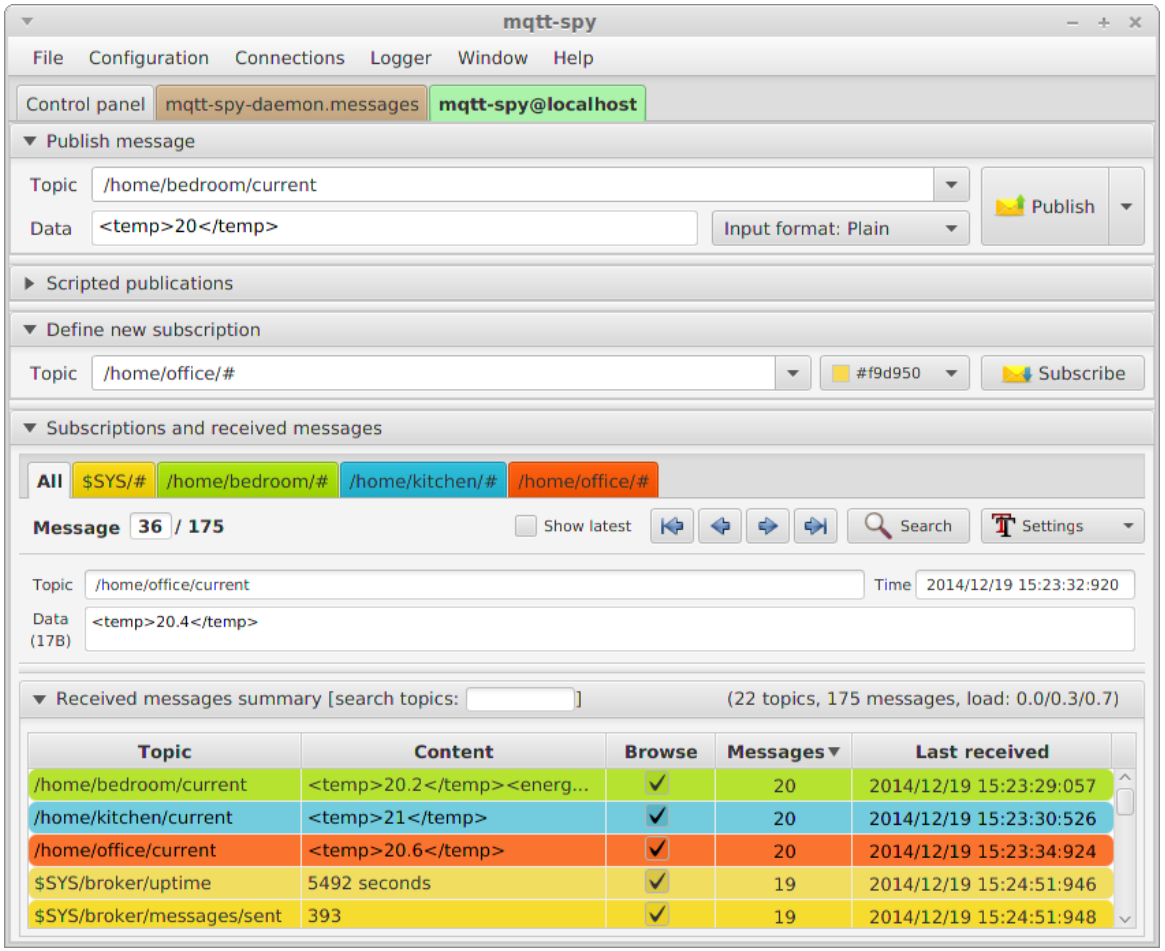
```
mosquitto_pub -h 127.0.0.1 -t piandmore/rpi-sub/led -m 4
mosquitto_pub -h 127.0.0.1 -t piandmore/rpi-sub/led -m 3
mosquitto_pub -h 127.0.0.1 -t piandmore/rpi-sub/led -m 2
mosquitto_pub -h 127.0.0.1 -t piandmore/rpi-sub/led -m 1
mosquitto_pub -h 127.0.0.1 -t piandmore/rpi-sub/led -m 0
```

Demo 2: externe Tools (mqtt-spy)

Einführung

Mosquitto
- Info
- Installieren
- D1: Hello World
- **D2: Temp Ctrl**

Abschluss



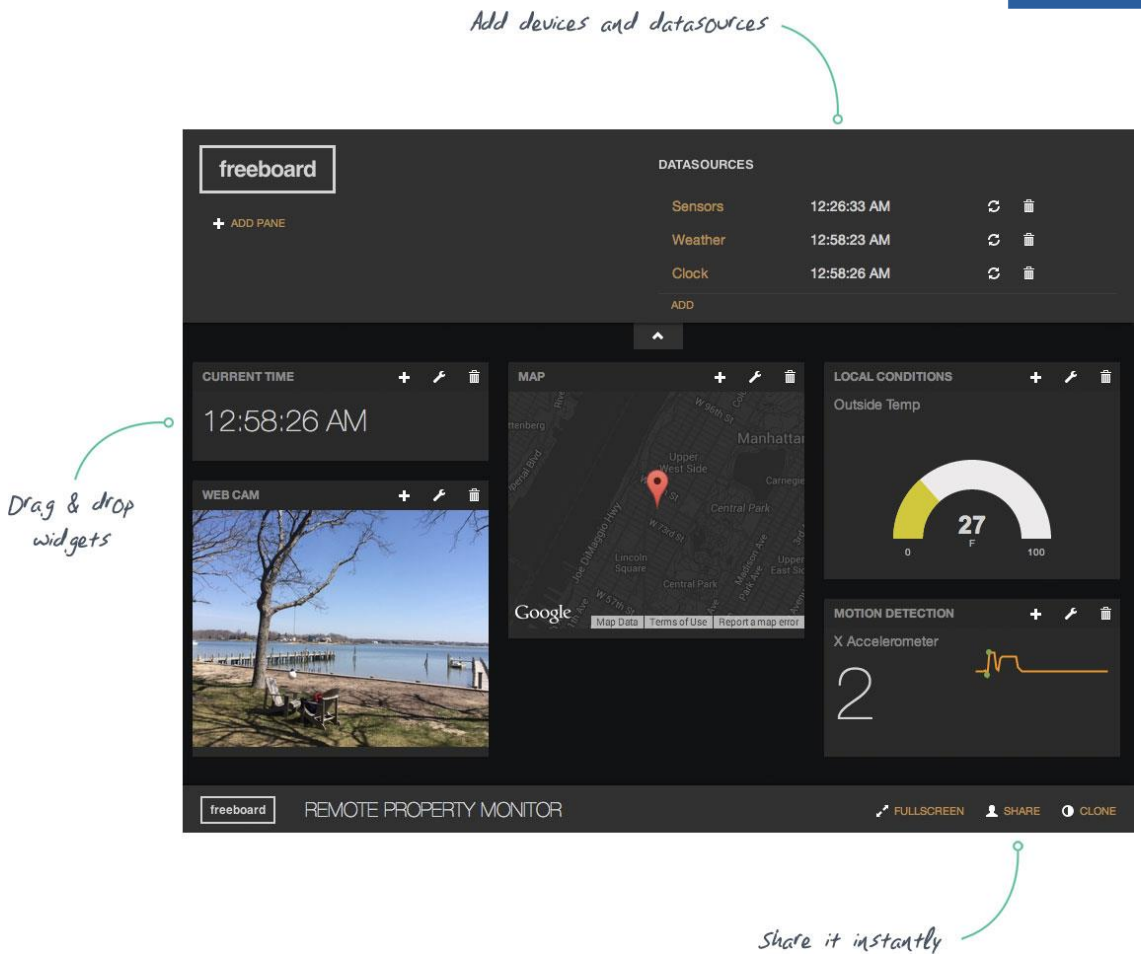
https://code.google.com/p/mqtt-spy/

Demo 2: externe Tools (freeboard)

Einführung

Mosquitto
- Info
- **-** Installieren
- D1: Hello World
- **D2: Temp Ctrl**

Abschluss



https://github.com/Freeboard/freeboard
http://jpmens.net/2014/11/12/freeboard-a-versatile-dashboard/

Demo 2: Controller (1/2)

Einführung

Mosquitto
- Info
- Installieren
- D1: Hello World
- **D2: Temp Ctrl**

Abschluss

rpi-broker

```
#!/usr/bin/python
import sys
import paho.mqtt.client as mqtt

def on_connect(mqttc, obj, flags, rc):
    print("Connected with result code "+str(rc))
    # Subscribing in on_connect(): if we lose connection and
    # reconnect subscriptions will be renewed. QoS level 0
    mqttc.subscribe("piandmore/rpi-pub/temp", 0)

def on_message(mqttc, obj, msg):
    print(msg.topic+" "+str(msg.qos)+" "+str(msg.payload))
    temp = float(msg.payload)
    if (temp < 33.00):
      mqttc.publish("piandmore/rpi-sub/led", "1")
    elif (temp > 33.00):
      mqttc.publish("piandmore/rpi-sub/led", "2")

def on_publish(mqttc, obj, mid):
    print("mid: "+str(mid))
```

Demo 2: Controller 2/2

Einführung

Mosquitto
- Info
- Installieren
- D1: Hello World
- **D2: Temp Ctrl**

Abschluss

rpi-broker

```
def on_subscribe(mqttc, obj, mid, granted_qos):
    print("Subscribed: "+str(mid)+" "+str(granted_qos))

def on_log(mqttc, obj, level, string):
    print(string)

mqttc = mqtt.Client()
mqttc.on_message = on_message
mqttc.on_connect = on_connect
mqttc.on_publish = on_publish
mqttc.on_subscribe = on_subscribe
# Uncomment to enable debug messages
#mqttc.on_log = on_log
# Last will and testament
mqttc.will_set("piandmore/rpi-sub/led", "4")

mqttc.connect("127.0.0.1", 1883, 60)

# loop to keep connection and handling alive
mqttc.loop_forever()
```

III. Ausblick

P4

Ausblick

Einführung

Mosquitto

Abschluss
- Ausblick
- mosquitto 1.4

- ACL Dateien
- Retain Flag
  - Daten auf dem Broker speichern
- QoS 0, 1, 2
- Broker Bridge
- Arduino Beispiel
- http://test.mosquitto.org/
- http://lelylan.github.io/lab-projects/raspberry-pi-light/

Build mosquitto 1.4 with websockets 1/2

Einführung

Mosquitto

Abschluss
- Ausblick
- **mosquitto 1.4**

- # This tutorial assumes you got the mosquitto paket already installed

- # Stop mosquitto service
  sudo /etc/init.d/mosquitto stop

- # Install libwebsockets
  sudo apt-get install -y cmake libssl-dev
  mkdir ~/mqtt_dev
  cd ~/mqtt_dev
  wget http://git.warmcat.com/cgi-bin/cgit/libwebsockets/snapshot/libwebsockets-1.3-chrome37-firefox30.tar.gz
  tar -xzvf libwebsockets-1.3-chrome37-firefox30.tar.gz
  cd libwebsockets-1.3-chrome37-firefox30/
  mkdir build
  cd build
- # cmake .. -DOPENSSL_ROOT_DIR=/usr/bin/openssl
- # Disable IPv6, creates error in mosquitto make test
  cmake .. -DLWS_IPV6=OFF -DOPENSSL_ROOT_DIR=/usr/bin/openssl
  make
  sudo make install
  sudo ldconfig

- # Install git tools
  sudo apt-get install -y git

- # Clone the Mosquitto repo and switch to the 1.4 branch
  cd ~/mqtt_dev
  git clone https://git.eclipse.org/r/mosquitto/org.eclipse.mosquitto
  cd org.eclipse.mosquitto/
  git checkout origin/1.4

- # Change files to enable WEBSOCKETS
- # Edit config.mk and ensure that the websockets option is set to “yes”.
  WITH_WEBSOCKETS:=yes

- # Install pre-requirements:
  sudo apt-get install -y uuid-dev xsltproc docbook-xsl

- # Make Mosquitto
  make
  make test
  # even if test gives "address familiy not supported", just keep on installing
  sudo make install

Build mosquitto 1.4 with websockets 2/2

Einführung

Mosquitto

Abschluss
- Ausblick
- **mosquitto 1.4**

- # Change mosquitto daemon to new path
  sudo vi /etc/init.d/mosquitto
  Change DAEMON=/usr/sbin/mosquitto to DAEMON=/usr/local/sbin/mosquitto

- # Create additional mosquitto configuration file
  sudo vi /etc/mosquitto/conf.d/mqtt14.conf
  autosave_interval 1800
  #persistence true
  persistence_file m2.db
  #persistence_location /var/mosquitto/
  connection_messages true
  log_timestamp true

  #acl_file /etc/mosquitto/user.acl
  #password_file /etc/mosquitto/user.pw

  listener 1883

  listener 9001 127.0.0.1
  protocol websockets

- # Restart mosquitto service
  sudo /etc/init.d/mosquitto start

IV. Fragen?

Vielen Dank für Ihre Aufmerksamkeit!

Quellenangaben

- Foto der Frontfolie: http://www.theconnectivist.com/2013/05/declassified-the-governments-quantum-internet/
- P0: http://blogs.vmware.com/vfabric/2013/02/choosing-your-messaging-protocol-amqp-mqtt-or-stomp.html
- www.mqtt.org
- http://en.wikipedia.org/wiki/MQTT
- http://2lemetry.com/2013/08/22/mqtt-in-a-nutshell/
- http://www.xappsoftware.com/wordpress/2014/10/20/is-mqtt-the-panacea-for-all-the-problems-of-iot/
- http://de.slideshare.net/andypiper/mqtt-27769811
- P1: http://rijware.com/mosquitto-bridge-to-secure-mqtt-broker/
- https://www.justinribeiro.com/chronicle/2012/11/08/securing-mqtt-communication-between-ardruino-and-mosquitto/
- P2: http://www.xappsoftware.com/wordpress/2014/10/27/installing-mosquitto-on-raspberry-pi/
- P3: http://www.adafruit.com/product/1914
- http://jpmens.net/2013/09/01/installing-mosquitto-on-a-raspberry-pi/

Quellenangaben

- http://jpmens.net/2014/07/03/the-mosquitto-mqtt-broker-gets-websockets-support/
- https://goochgooch.wordpress.com/2014/08/01/building-mosquitto-1-4/
- https://answers.launchpad.net/mosquitto/+question/252173
- http://jpmens.net/2014/11/12/freeboard-a-versatile-dashboard/
- http://2lemetry.com/2014/10/24/connecting-correctly-mqtt/
- http://2lemetry.com/2014/10/24/avoiding-mqtt-pitfalls/
- P4: http://www.electronicsweekly.com/news/iot-2/iot-killed-many-standards-2014-07/

- http://jpmens.net/2013/02/25/lots-of-messages-mqtt-pub-sub-and-the-mosquitto-broker/
- http://www.willowdesign.info/blog/mqtt-on-arduino/
- http://blog.atx.name/building-avr-board-with-mqtt-support-for-iot/
- http://davidegironi.blogspot.de/2014/12/a-web-configurable-xively-logger-build.html#.VMAOztzIYjK