Cyber Physical Systems

Wintersemester 2020/21

Hochschule München University of Applied Sciences

Lektion 06: MQTT Broker einrichten und in Rallye Anwendung integrieren

von: Max Holzmann

In der Rallye Anwendung wird MQTT zur Kommunikation im Netzwerk/WLAN verwendet, um Daten zwischen dem RallyMaster und den RallyESPs auszutauschen.

Ziele

- Inbetriebnahme eines MQTT Brokers auf dem Raspberry Pi
- Publishen und subscriben von Topics mit dem ESP32
- Publishen von Topics mit Node Red

Voraussetzungen und erforderliches Equipment

- Die Tutorial Lektionen ...
  01: Grundlegende Einstellungen und Installationen link
  04: Node Red installieren link
  ... müssen erledigt sein
- Hardware
  - 1x Raspberry Pi 3 B
  - 1x ESP DevKitC
- Verwendete Software und verwendete Tools in diesem Tutorial
  - MQTTFX oder MQTTX zum Testen und Debugen

Lösungsschritte

1. Installation des kostenlosen MQTT Brokers (mosquitto)

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

2. Starten des Brokers als Service

```
sudo systemctl enable mosquitto.service
```

3. Funktionstest des Brokers unter Verwendung der Programme MQTTFX oder MQTTX.
   - verbinden zum Server {IP_OF_PI}:1883

    - einen Topic subscriben z.b. Test/Top
    - und dann auf diesen Topic etwas publishen

4. Tests sowohl über die Ethernetverbindung als auch über den Wifi Acesspoint testen.

Topic structure in der Rallye Anwendung

Struktur:

-rally/

- cars/
    - car_X/
        - wifi_signal_quality (last will "not connected")
        - measurement_number
        - difference
        - drive_though_timestamp
        - beep
    - ...

Details:

Topic	**Publisher**	**Subscriber**	**Wann**	**Bemerkung**
wifi_signal_quality:	RallyESP	RallyMaster	jede Sekunde	mit last Will die Signalqualität am Brocker auf -1000 dbm zu setzen
measurement_number:	RallyMaster	RallyESP	nach Durchfahrt	Anzahl Messungen dieses RallyESPs
difference:	RallyMaster	RallyESP	nach Durchfahrt	Abweichung zur vollen Minute
drive_though_timestamp:	RallyMaster	RallyESP	nach Durchfahrt	Zeitstempel der Durchfahrt
beep:	RallyMaster	RallyESP	nach Durchfahrt	Auslösen des Piep-Tons bei der Durchfahrt

Anwendung im ESP Programm

Für die MQTT Verbindung wird die Arduino Wifi Bibliothek verwendet und die PubSubClient Bibliothek, welche über den Bibliotheksmanager der Arduino IDE installiert werden kann.

Die wichtigsten Code Zeilen des Programms werden hier kurz erklärt. Das gesammte Programm kann dem Ordner ../0100_Rally_ESP_Code entnommen werden.

Verbindung zum Broker:

```
void setup(void)
...
  client.setServer(mqtt_server, 1883);
  client.setCallback(mqttTopicChangedCallback);
...
}
```

Funktion, um bei Verbindungsunterbrechungen eine erneute Verbindung zum Broker aufzubauen:

```
boolean reconnect() {

  if (client.connect(topicChar, NULL, NULL, wifiTopic, 2,true,
willWifiMessage,true)) { //todo last will
    // list of cubscribed toppics
    client.subscribe(measurementNrTopic);
    client.subscribe(differenceTopic);
    client.subscribe(driveThroughtopic);
    client.subscribe(beepTopic);
  }
  return client.connected();
}
```

Callbackfunktion bei Änderung eines Subscribten Topics:

```
void mqttTopicChangedCallback(char* arrtopic, byte* msg, unsigned int length) {

  char message[32];
  memset(message, 0, sizeof(message));
  for (int i=0;i<length;i++) {
    message[i]=(char)msg[i];
  }
  Serial.print(" msg: ");
  Serial.println(message);

  //switch case for handling the different Toppics
}
```

ein einfacher Publish:

```
client.publish(wifiTopic, rssi);
```

```
***
# Anwendung in Node Red

Anwenden der MQTT Nodes aus der lib node-red-contrib-mqtt-dynamic Bibliothek. Um
Topics zu Subscriben und um Topics zu Publishen. Diese Bibliothek bietet Vorteile,
wenn dynamische Topics in der Anwendung erforderlich sind. Mehr Details hierzu in
der Lektion 04: Node Red.


## Further Inputs

Der verwendete MQTT Broker wird im Tutorial komplett unverschlüsselt und ohne
Zertifikate verwendet. Eine Weiterentwicklung kann sein, den Broker mit
Serverzertifikaten auszustatten und ggf. auch noch die Clients nur per Zertifikat
zu authentifizieren.
```