

Besteht aus bis zu drei Teilen:

- **Fixed Header** - in allen MQTT Paketen vorhanden
 - Art des Paketes (SUBSCRIBE, PUBLISH, CONNECT, etc.)
 - ggf. QoS-Level
 - Verbleibende Länge im Paket (restlicher Variable-Header + Payload)
- **Variable Header**
 - Je nach Art des Paketes unterschiedlicher Inhalt
 - z.B. Topic, Verbindungs-Flags, Packet-Identifizier
- **Payload**
 - Ebenfalls verschiedener Inhalt je nach Paket-Typ
 - z.B. Daten oder Topics, die ein Client *abonnieren* möchte

Projektumsetzung

Implementierung von MQTT-CONNECT & MQTT-PUBLISH auf ESP8266

ESP8266 verbindet sich mit MQTT-CONNECT zum Mosquitto-Broker auf Raspi

- Protokol-Name und Version
- Art der Verbindung
- Keep-Alive
- Client ID

```
MQ Telemetry Transport Protocol, Connect Command
  Header Flags: 0x10, Message Type: Connect Command
    0001 .... = Message Type: Connect Command (1)
    .... 0000 = Reserved: 0
  Msg Len: 25
  Protocol Name Length: 4
  Protocol Name: MQTT
  Version: MQTT v3.1.1 (4)
  Connect Flags: 0x02, QoS Level: At most once delivery (Fire and Forget), Clean Session Flag
    0... .... = User Name Flag: Not set
    .0.. .... = Password Flag: Not set
    ..0. .... = Will Retain: Not set
    ...0 0... = QoS Level: At most once delivery (Fire and Forget) (0)
    .... .0.. = Will Flag: Not set
    .... ..1. = Clean Session Flag: Set
    .... ...0 = (Reserved): Not set
  Keep Alive: 240
  Client ID Length: 13
  Client ID: ESP8266NETZE2
```

Projektumsetzung

Implementierung von MQTT-CONNECT & MQTT-PUBLISH auf ESP8266

ESP8266 sendet Temperaturdaten mit MQTT-PUBLISH an Mosquitto-Broker auf Raspi

- QoS-Flags
- Topic
- Payload (Temperaturdaten)

```
▼ MQ Telemetry Transport Protocol, Publish Message
  ▼ Header Flags: 0x30, Message Type: Publish Message, QoS Level: At most once delivery (Fire and Forget)
    0011 .... = Message Type: Publish Message (3)
    .... 0... = DUP Flag: Not set
    .... .00. = QoS Level: At most once delivery (Fire and Forget) (0)
    .... ...0 = Retain: Not set
  Msg Len: 9
  Topic Length: 3
  Topic: a/b
  Message: 00000bda
```

Demo