

Message Queuing Telemetry Transport

Implementierung einer IoT-Anwendung auf Basis von MQTT

Maximilian Gaul, Lukas Dorner

01.07.2019

- Idee, IoT-Projekte in 5 Layer aufzuteilen
 - **Objects Layer:** Physikalischer Sensoren und Aktuatoren, die verschiedene Funktionen übernehmen ✓
 - **Object Abstraction Layer:** Transport der Daten zum nächsten Layer, z.B. über WiFi, Bluetooth ✓
 - **Service Management Layer:** Verarbeitet und abstrahiert empfangene Daten bzw. die Hardwareplattform ✓
 - **Application Layer:** Stellt den Anwendern die Daten zur Verfügung, die sie benötigen (z.B. Temperaturdaten)
 - **Business Layer:** Verwaltung der gesamten IoT-Applikation, Verwendung der Daten für z.B. *Big Data*

Paper

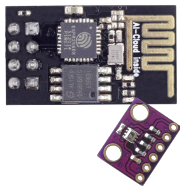
WiFi? Bluetooth? MQTT? Temperatursensor?

- Viele verschiedene Hardware- und Protokollkombinationen denkbar

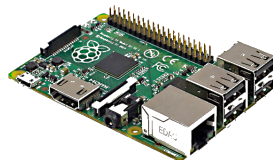
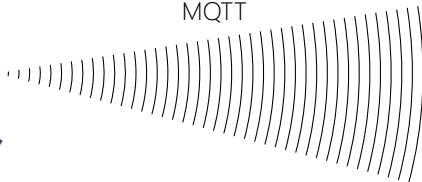
IoT Elements		Samples
Identification	Naming	EPC, uCode
	Addressing	IPv4, IPv6
Sensing		Smart Sensors, Wearable sensing devices, Embedded sensors, Actuators, RFID tag
Communication		RFID, NFC, UWB, Bluetooth, BLE, IEEE 802.15.4, Z-Wave, WiFi, WiFiDirect, , LTE-A
Computation	Hardware	SmartThings, Arduino, Phidgets, Intel Galileo, Raspberry Pi, Gadgeteer, BeagleBone, Cubieboard, Smart Phones
	Software	OS (Contiki, TinyOS, LiteOS, Riot OS, Android); Cloud (Nimbits, Hadoop, etc.)
Service		Identity-related (shipping), Information Aggregation (smart grid), Collaborative-Aware (smart home), Ubiquitous (smart city)
Semantic		RDF, OWL, EXI

Projektidee im IoT-Bereich

WiFi, MQTT, Temperatursensor



MQTT



ESP8266 + BMP280

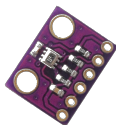
Raspberry Pi + Mosquitto

Projektidee im IoT-Bereich

WiFi, MQTT, Temperatursensor



- 32-Bit *RISC*-Controller
- 802.11 b/g/n mit bis zu 72,2Mbps
- 96 KByte RAM
- 4 MB Flash
- Soft-I2C Anbindung an BMP280



- Temperatur- und Drucksensor
- 20-Bit Auflösung
- I2C-Interface



- Raspbian OS
- Mosquitto MQTT Broker & Subscriber
- Python-Skripts zum Auswerten der empfangenen Daten