**ML03\_RPI-MQTTbroker-NodeRED**

**Téma:** Komunikace M2M s využitím protokolu MQTT. Konfigurace MQTT broker a klienta na RPI. V prostředí Node-Red generování a přijímání zprávy topic.

**Cíl hodiny:** Použít komunikaci M2M pomocí MQTT protokolu. Umět nainstalovat MQTT broker na Raspberry Pi. Naučit se pracovat ve vývojovém prostředí Node-Red, na RPI propojit MQTT klienta s MQTT brokerem, generovat a přijímat MQTT zprávy topic.

**Doba trvání:** 4x45min

**Pomůcky:** Rasberry Pi 3

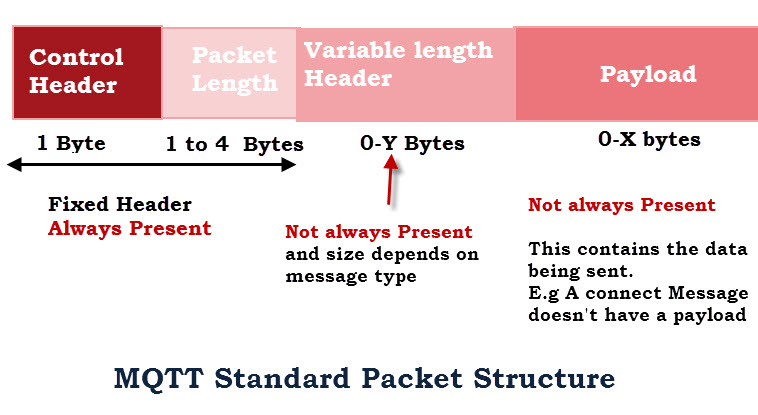
# Seznámení s protokolem MQTT

Jedná se o jednoduchý nelicencovaný protokol (Message Queuing Telemetry Transport) s malou režií standardizovaný technickou komisí OASIS, který mohou podporovat i nejmenší měřící a monitorovací zařízení a umožňuje přenášet data přes rozsáhlé nespojité sítě. MQTT je aplikační protokol publikování a doručování zpráv optimalizovaný pro připojení fyzických zařízení a přenos událostí do centrálního serveru nebo jinému uživateli. Je proto vhodný pro vznikající svět IoT a propojení zařízení Machine to Machine především v sítích, kde je vyžadována nízká spotřeba energie a ve kterých nevadí nízká šířka pásma, vysoká latence, omezení množství přenášených dat a nestabilní spojení.

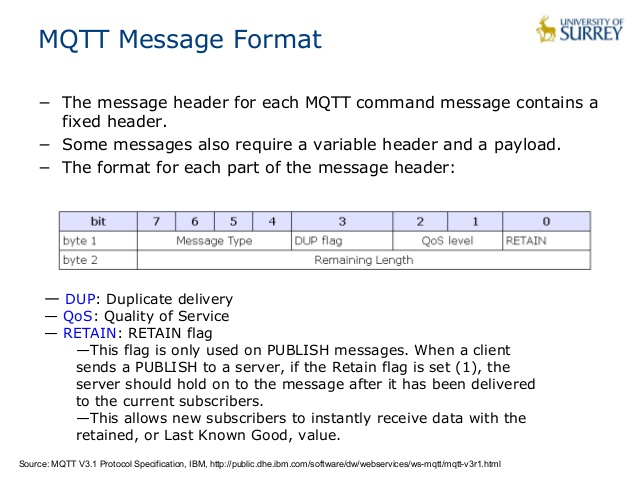
Protokol MQTT využívá na transportní vrstvě modelu TCP/IP protokol TCP s portem 1883, který je rezervován společností IANA. Šifrování v síti lze provádět pomocí protokolu SSL, pro který je rezervován port 8883 nezávisle na samotném protokolu MQTT, což ale přidává významnou síťovou režii.

**Komunikace MQTT**

Protokol MQTT je založen na výměně řídících paketů, podrobný přehled je uvedený v příloze B. Řídící pakety MQTT tvoří vždy pevné záhlaví o velikosti min. 2 B a ve většině případů variabilní záhlaví (obsahuje identifikátor paketu) a payload (max. 256 MB). Nejčastěji se využívají řídící pakety CONNECT, PUBLISH, SUBSCRIBE a DISCONNECT. Komunikace probíhá asynchronně prostřednictvím zpráv.



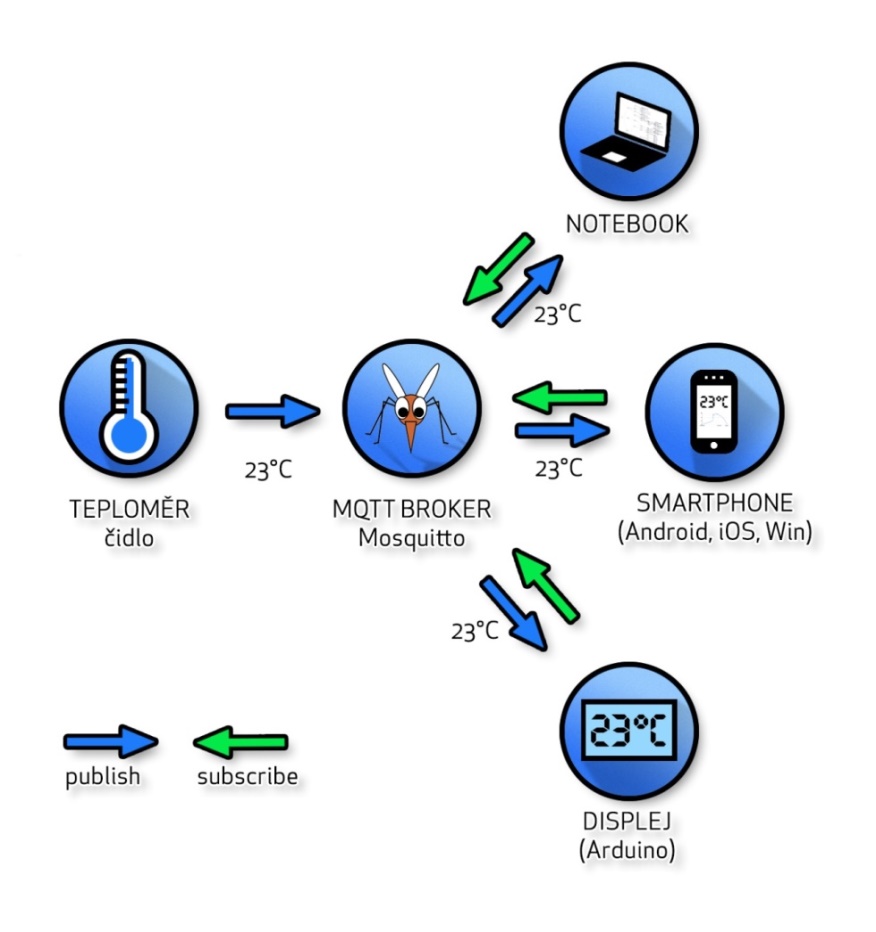
Struktura řídícího paketu MQTT



Formát pevného záhlaví MQTT zprávy

* DUP: Duplicate delivery (duplikované doručení)
* QoS: Quality of Service
* RETAIN: Retain flag (použit pouze při zprávě publish). Pokud je nastaven, měl by server po převzetí aktuálním účastníkům zprávu udržet. To umožňuje novým účastníkům okamžitě přijímat data se zachovanou nebo poslední známou platnou hodnotou.

Zprávy produkují MQTT klienti, např. čidla. Ta je publikují (publish) do jednoho centra nazývaném MQTT Broker. Každá zpráva je publikována s určitým názvem tématu (topic). Toto téma po jeho publikaci poté broker rozešle všem klientům, kteří si je objednali k odběru (subscribe). Klient je tedy koncové zařízení, které publikuje nebo přijímá zprávy, může plnit i obě funkce najednou. Může jím být nějaká webová služba, služba operačního systému, jednoduchý program nebo hardwarové řešení. Senzor publikuje (např. teplotu, pohyb), a aktor provádí určitou akci na základě přijaté zprávy topic (např. display, relé, motor, …). Hardwarové řešení neovlivňuje přenášenou zprávu (topic), zpracování topicu pro daný hardware řeší program daného koncového bodu.



Příklad topologie s komunikací pomocí MQTT protokolu

# Instalace MQTT brokeru a klienta

MQTT klient nemusí být hardwarové zařízení, ale pouze služba puštěná pod operačním systémem v terminálu. Na jednom Raspberry Pi může tedy běžet zároveň MQTT broker a MQTT klient.

1. Instalace MQTT broker

sudo apt-get install mosquitto //instalace brokeru

sudo /etc/init.d/mosquitto start //nastartování mosquitto

sudo ss -a -p | grep mosq //kontrola zda běží, MQTT používá port 1883

systemctl status mosquitto –l //zobrazení stavu mosquitto

1. Instalace MQTT klient

sudo apt-get install mosquitto-clients //instalace klienta

mosquitto\_pub -t RPI -m "text zpravy" //publikování zprávy „text zpravy“

mosquitto\_sub -t RPI //přijímání zprávy

# Seznámení s Node-Red

Node-Red je vývojové grafické prostředí, které je předinstalované v OS Raspbian. Je možné ho spustit z hlavního menu nebo z terminálu příkazem node-red-start. Konfigurace se provádí v grafickém prostředí v prohlížeči na IP adrese, kde Node-Red běží (přidělená IP adresa RPI z AP – 192.168.100.100 nebo localhost) a na portu 1880. Umožňuje sestavení diagramu toků dat z vývojových funkčních bloků propojených mezi sebou:

* Input – inject, mqtt klient, …
* Output – debug, mqtt broker, …
* Function – delay, trigger, switch, …
* Social – email, twitter, …
* Storage – file, …
* Analysis
* Advanced – watch, exec, …
* Raspberry Pi – rpi gpio, …

V hodině budou žáci pracovat hlavně s funkčními bloky – Input, Output. Jednotlivé funkční bloky se dají po dvojitém kliknutí konfigurovat. U vstupů je potřeba zadat data, která se mají posílat a jakého datového typu budou a název zprávy topic, který je stejný jako u klienta na druhé straně komunikace. U výstupu je nejjednodušší použít funkční blok msg. payload, pomocí kterého se budou data vypisovat vpravo na liště v záložce debug.

Obsah obrázku snímek obrazovky, počítač

Popis byl vytvořen automaticky

Konfigurace vstupního bloku ESP-LED1

Při použití MQTT vstupu a výstupu je potřeba zadat IP adresu MQTT serveru neboli MQTT brokeru, který si žáci nainstalovali na RPI a port 1883 používaný při komunikaci protokolem MQTT. Opět musí být zadán stejný název zprávy topic jako je na druhé straně komunikace.

Obsah obrázku snímek obrazovky, počítač, monitor, přenosný počítač

Popis byl vytvořen automaticky

Posílání teploty z terminálu a konfigurace vstupního MQTT bloku ESP1-Teplota

# Vypracování PL03\_RPI-MQTTbroker-NodeRED

Při úspěšném zpracování pracovního listu by žáci měli zvládnout:

* Na RPI nainstalovat MQTT broker a ověřit jeho aktivní stav.
* Na RPI nainstalovat MQTT klienta a vyzkoušet zasílání zprávy topic.
* Vytvořit ve vývojovém prostředí RPI Node-Red diagram pro publikování zprávy topic ESP1-LED na MQTT broker, která bude z MQTT brokeru vypisována na terminál RPI.
* Vytvořit ve vývojovém prostředí RPI Node-Red diagram pro přijímání zprávy topic ESP1-Teplota z MQTT brokeru. Tato zpráva topic bude publikována z terminálu RPI a vypisována na debug konzolu v prostředí Node-Red.
* Ve vývojovém prostředí RPI Node-Red vytvořit dva diagramy (public, subscribe) a posílat mezi nimi zprávu topic.