



ZÁPADOČESKÁ UNIVERZITA V PLZNI

KIV/ASWI

POKROČILÉ SOFTWAROVÉ INŽENÝRSTVÍ

**Konfigurovatelný dashboard zobrazování
senzorových dat**

Tým:

Jan Rach
Milan Kohoutek
Pavel Průcha
Martin Forejt

Číslo studenta

A20N0103P
A20N0090P
A20N0102P
A20N0079P

6. června 2021

Obsah

1	Zadání	2
2	Implementace	4
3	Uživatelská dokumentace	5
3.1	Simulace odesílání dat	5
4	Testovací scénáře	7
4.1	Vytvoření nového topicu	8
4.2	Odeslání dat pomocí MQTTLens	9
4.3	Zobrazení více grafů (topiců)	11

1 Zadání

Úkolem týmu bude vytvořit aplikaci pro zobrazení dat z IoT senzorů s využitím MQTT protokolu. Cílem projektu je vytvoření MQTT klienta, který sleduje příchozí zprávy (témata) na MQTT serveru a pro každé téma vytvoří okno, ve kterém bude zobrazovat časový průběh přijímaných dat.

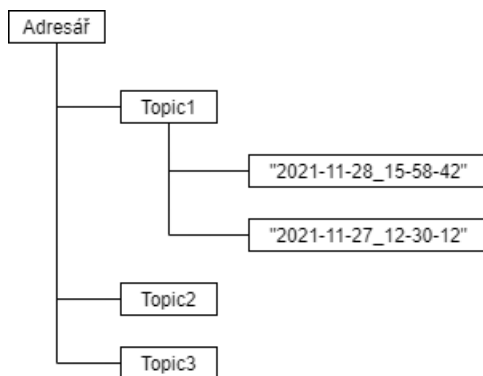
Minimální akceptační kritéria

- P1:** Client Subscriber sleduje MQTT Broker,
- P2:** jakmile přijde zpráva s daným tématem otevře se okno a začnou se zobrazovat data.
- P3:** Spolu s přijatým tématem se založí soubor pod daným tématem a začnou se ukládat data.
- P4:** Pokud přestane Publisher posílat data, tak se po nastavitelné době ukončí odběr, zavře se okno pro téma a uzavře se soubor s uloženými daty.

Dodatečné funkce

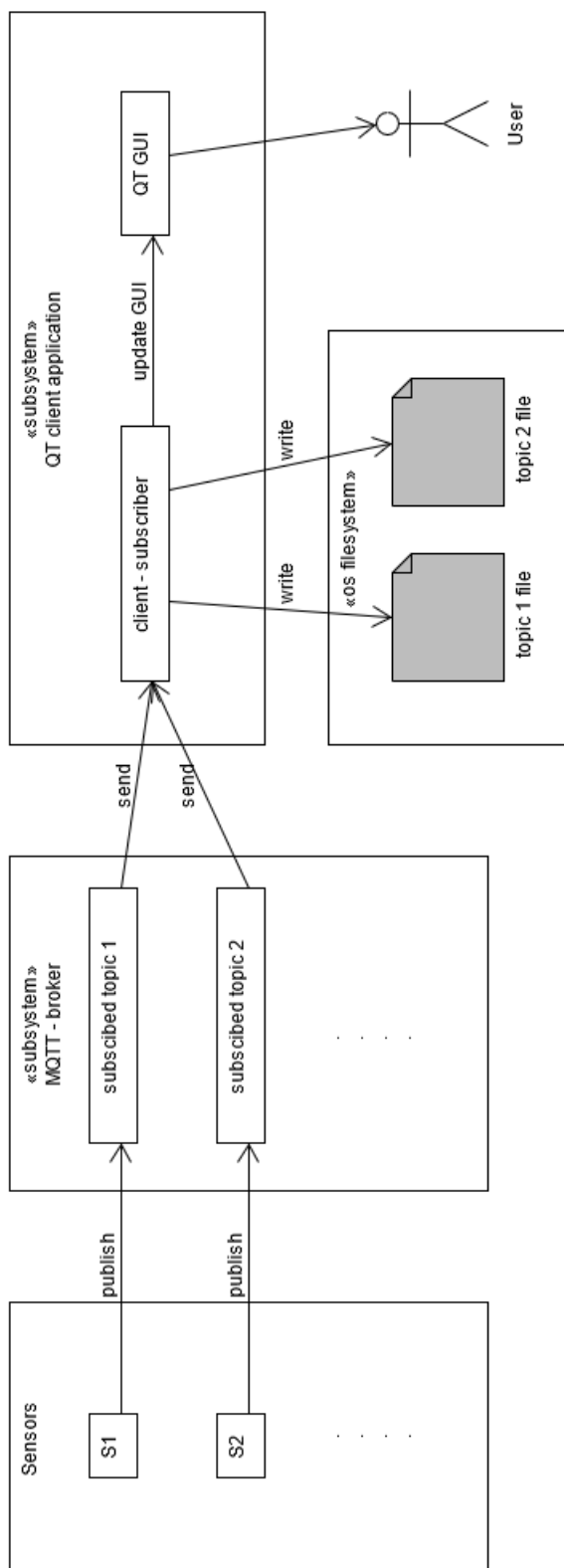
- DF1:** Nad soubory s daty bude možné provádět následující operace - export, porovnání dat se staršími daty a otevření sady dat

Publisher odesílá zprávy ve formátu **datum ; čas ; index ; hodnota senzoru**. Přijaté zprávy se budou ukládat do CSV souborů, jejichž jméno se skládá z datumu a času. Jednotlivé topicy (senzory) budou mít vlastní adresář s CSV soubory přijatých dat viz obrázek 1.



Obrázek 1: Ukládání dat do souborů

Component diagram projektu lze vidět na obrázku 2.



Obrázek 2: Component Diagram

2 Implementace

Aplikace je vytvořena v *Python v2.7.18*, klient (subscriber) je vytvořen pomocí *Paho-mqtt v1.5.1*, GUI je řešeno pomocí *PyQt5 v5.15.4* a zobrazování grafů přes knihovnu *Matplotlib v3.4.1*.

Vzhledem k omezenému počtu dostupných mikropočítačů Raspberry Pi, byla aplikace vyvíjena a komunikace simulována na počítači s operačním systémem Windows 10.

Pro spuštění projektu je zapotřebí správce balíčků *Poetry*, aplikaci *MQTTLens* pro simulování klient publisher (ESP32 M5stack) a *Mosquitto Broker*.

Zdrojové soubory jsou hierarchicky uspořádány podle MVC¹ architektury.

Kořenový adresář obsahuje podadresáře Model, View a Service.

- **Model** - adresář obsahuje strukturu přijímaných zpráv.
- **View** - adresář obsahuje zdrojové soubory týkající se grafického uživatelského rozhraní.
- **Service** - adresář obsahuje vše ohledně ukládání dat do souborů, získávání dat a parsování zpráv.

Spuštění projektu ze zdrojových souborů

Příkazem `poetry install` se nainstalují všechny knihovny které byly použity v projektu. Příkazem `poetry run start` se projekt spustí.

Vytvoření buildu a spuštění aplikace

Build projektu se vytvoří následujícími příkazy:

- `poetry install` – nainstaluje všechny použité knihovny
- `poetry build` – vytvoří nasaditelný soubor

Vytvoří se složka `dist`, která obsahuje mimo jiné i soubor

```
aswi2021vochomurka-1.0.0-py3-none-any.whl
```

Instalace projektu se pak provádí pomocí příkazu

```
pip install aswi2021vochomurka-1.0.0-py3-none-any.whl
```

Projekt spustíme příkazem `python -m aswi2021vochomurka.main`

¹Model View Controller

3 Uživatelská dokumentace

Spuštění

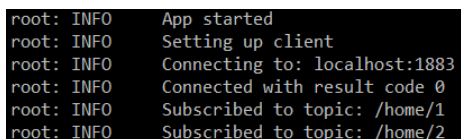
Pro spuštění projektu je zapotřebí správce balíčků *Poetry*, aplikaci *MQTTLens* pro simulování klient publishera (ESP32 M5stack) a *Mosquitto Broker*.

Sestavený projekt je uložen v souboru s koncovkou `.whl`. Vykonáním následujícího příkazu, se nainstalují použité knihovny:

```
pip install aswi2021vochomurka-1.0.0-py3-none-any.whl
```

Projekt spustíme příkazem:

```
python -m aswi2021vochomurka.main
```



```
root: INFO      App started
root: INFO      Setting up client
root: INFO      Connecting to: localhost:1883
root: INFO      Connected with result code 0
root: INFO      Subscribed to topic: /home/1
root: INFO      Subscribed to topic: /home/2
```

Obrázek 3: Příkazová řádka - spuštění aplikace

Ovládání

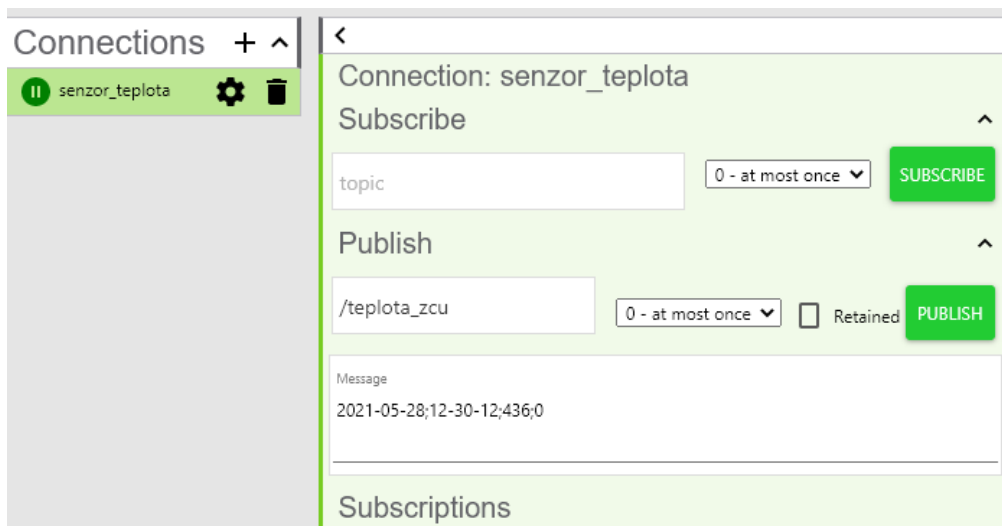
V nastavení můžeme nastavit host adresu, port, v jakém intervalu kontrolovat stav připojení (keepalive) a zdali připojení požaduje jméno a heslo či nikoliv. Dále se v nastavení vytváří a mažou topicy a nastavuje hodnota časového limitu v sekundách, kdy se zobrazený graf odstraní.

Na pravé straně grafu se vyskytují 3 tlačítka. Načtení sady dat k porovnání (*Load*), smazání načtené sady dat (*Delete*) a tlačítko *Focus*, které přepíná možnost sledování posledních 10 příchozích hodnot nebo všech příchozích hodnot v grafu.

3.1 Simulace odesílání dat

Aplikace *MQTTLens* slouží pro simulaci zasílání dat. V aplikaci lze jednoduše vytvořit spojení zadáním host adresy a portu. Samotné odesílání dat funguje tak, že do kolonky topic v sekci Publish vyplníme název našeho vytvořeného topicu (z nastavení hlavní aplikace) a zadáme "naměřenou" hodnotu senzoru ve formátu **datum ; čas ; index ; hodnota senzoru** viz obrázek 4.

Pro platformu Windows i Raspberri Pi (Raspbian) je instalace stejná, tedy *Mosquitto broker* a přes *Poetry* spustit aplikaci.



Obrázek 4: MQTTLens - vzor vyplnění

4 Testovací scénáře

Výsledná aplikace byla podrobena testování pomocí scénářů a také byla úspěšně otestována na mikropočítači Raspberri Pi. Scénáře byly testovány na počítači s operačním systémem *Windows 10*.

Scénáře na sebe navazují:

- **Vytvoření nového topicu** - spuštění hlavní aplikace, nastavení připojení, konfigurace topiců.
- **Odeslání dat pomocí MQTTLens** - spuštění MQTTLens, konfigurace připojení, odeslání zpráv na vytvořený topic z předchozího scénáře.
- **Zobrazení více grafů (topiců)**

Požadované technologie

Pro spuštění projektu je zapotřebí správce balíčků *Poetry*, aplikaci *MQTTLens* pro simulování klient publishera (ESP32 M5stack) a nainstalovaný *Mosquitto Broker*.

4.1 Vytvoření nového topicu

Spuštění příkazu `poetry run start` přes příkazový řádek v kořenovém adresáři projektu, se objeví okno hlavní aplikace MQTT klienta. Zvolením tlačítka *Settings* v levém horním rohu aplikace (viz obrázek 5), se zobrazí okno s konfigurací.

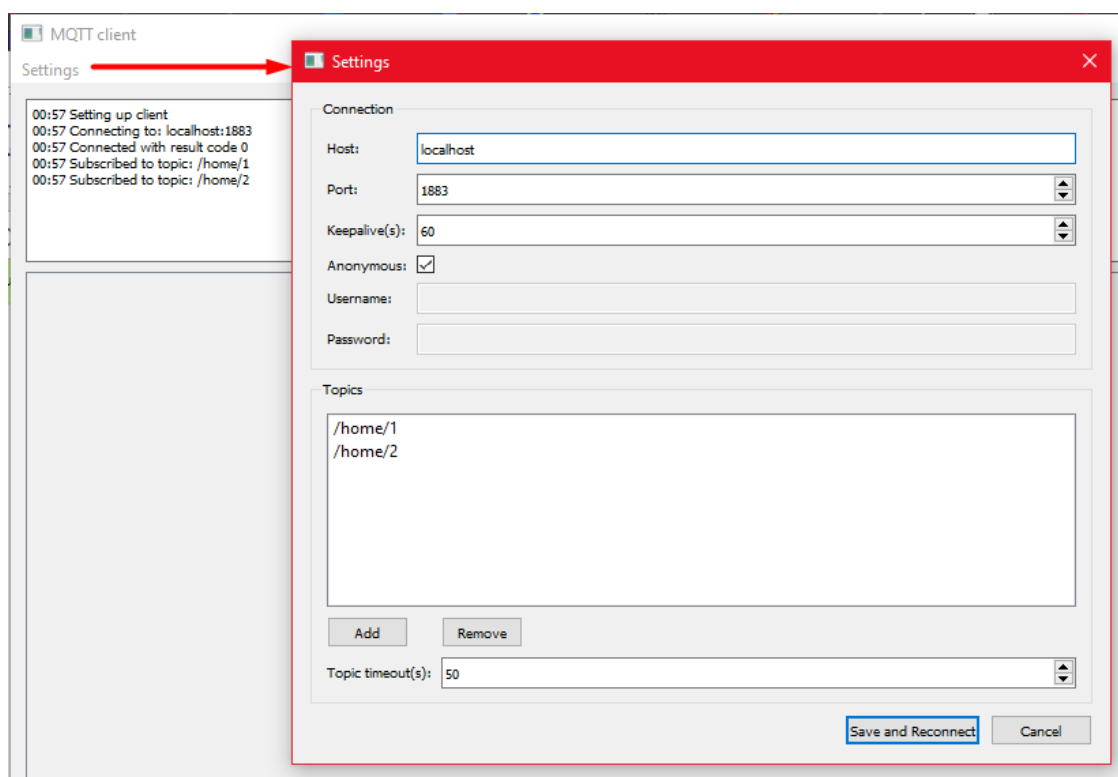
Zvolením tlačítka *Add* se zobrazí v listu topiců nový záznam. Dvojným kliknutím na nově přidáný záznam přejmenujeme název topicu na nový název `/home/3`.

Označením defaultního topicu `/home/1` v seznamu a zvolením tlačítka *Remove*, se námi zvolený topic smaže.

Nově zvolené hodnoty v konfiguraci (nastavení), uložíme tlačítkem *Save and Reconnect*.

Výsledek

Aplikace reagovala přesně podle popsaného scénáře. Jako výchozí hodnoty v seznamu topiců se vyskytoval topic `/home/1` a `/home/2`. První topic `/home/1` byl úspěšně smazán přes jeho označení a tlačítko *Remove*.



Obrázek 5: Správa topiců

4.2 Odeslání dat pomocí MQTTLens

Ve spuštěné aplikaci ověříme hodnoty v okně *Settings* ohledně konektivity. Host je nastaven na *localhost*, port je nastaven na *1883* a anonymní připojení je zaškrtnuté.

Spustíme aplikaci MQTTLens. Vytvoříme nový záznam připojení, kde vyplníme základní hodnoty *Hostname localhost* a port *1883*.

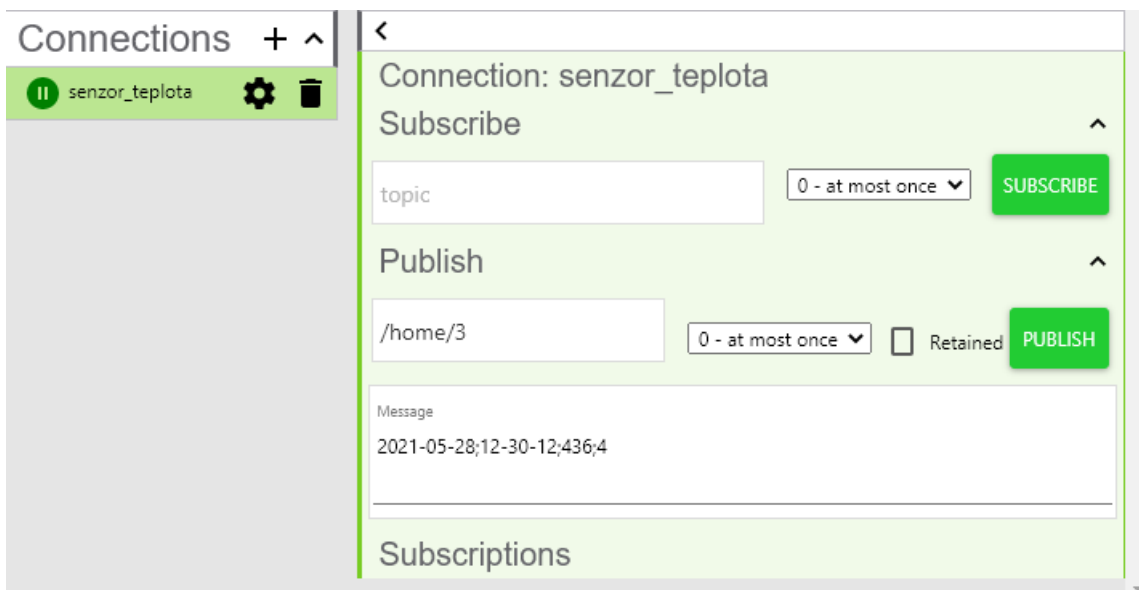
Zobrazí se nám 3 textová pole viz obrázek 6. První (Subscribe) necháme prázdný, do druhého textového pole zapíšeme hodnotu našeho vytvořeného topicu z minulého testu – */home/3*. Následně odešleme (pomocí tlačítka *Publish*) celkem 6 zpráv v následujícím formátu:

- 2021-05-28;12-31-12;436;4
- 2021-05-28;12-31-16;436;10
- 2021-05-28;12-31-17;436;4
- 2021-05-28;12-31-20;436;6
- 2021-05-28;12-31-22;436;8
- 2021-05-28;12-31-25;436;14

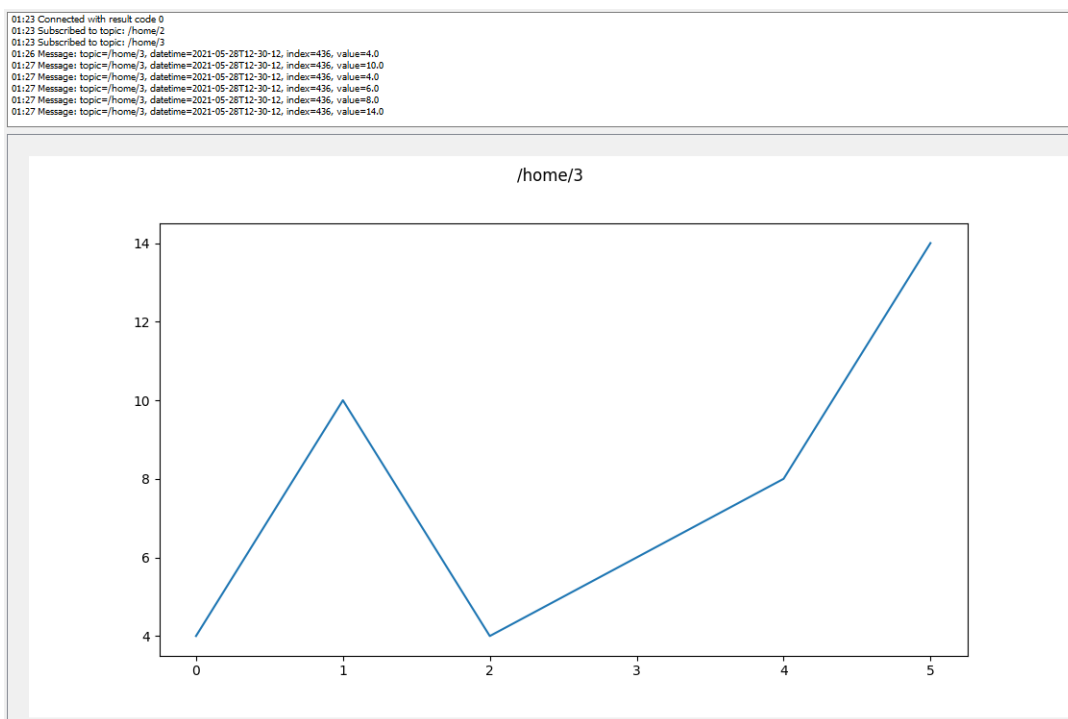
V hlavní aplikaci se zobrazí graf, na kterém budou zaznamenány přeposlané hodnoty viz obrázek 7.

Výsledek

Aplikace reagovala přesně dle popsaného scénáře. V adresáři *data/home/3* byl vytvořen soubor *2021-05-28_12-30-12.csv*, dle očekávání. Hlavní aplikace mimo jiné obsahuje záznamy aktivit získaných zpráv, které souhlasily s odeslanými daty.



Obrázek 6: Odeslání dat přes aplikaci MQTTLens



Obrázek 7: Graf z přijatých dat

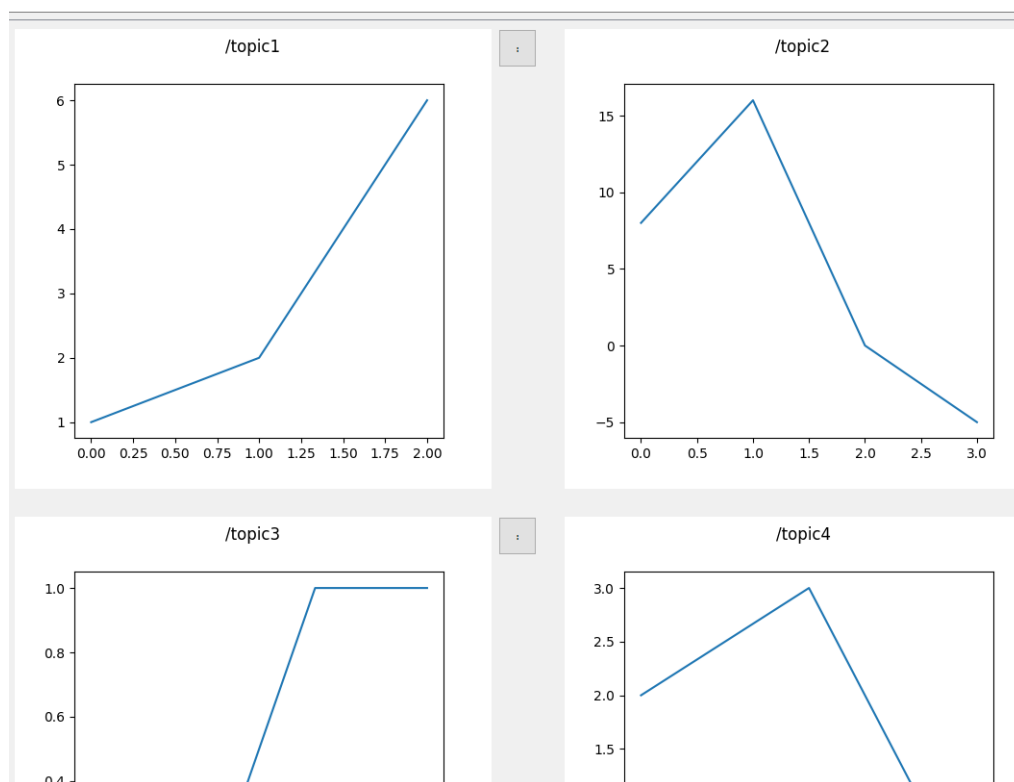
4.3 Zobrazení více grafů (topiců)

V nastavení aplikace vytvoříme 4 topicy s různými názvy a nastavíme hodnotu *Topic timeout* na 120 sekund.

Z MQTTLens odešleme různá data pod 4 různé topicy. Počkáme 120 sekund, než se grafy smažou a zkontrolujeme uložené hodnoty v CSV vytvořených souborech.

Výsledek

Přeposlané hodnoty odpovídaly hodnotám v grafu a také hodnotám v CSV souborech. Po odeslané poslední zprávě a vyčkání 120 sekund, grafy zmizely. Aplikace reaguje přesně dle popsaného scénáře.



Obrázek 8: Zobrazení více grafů