Politechnika Warszawska
Wydział Elektroniki i Technik Informacyjnych
Instytut Automatyki i Informatyki Stosowanej

Zaawansowane programowanie w C++

Sprawozdanie końcowe

Autorzy:
Marcin Dolicher
Konrad Winnicki

Prowadzący projekt:
mgr inż. Konrad Grochowski

Warszawa, 7 czerwca 2019

Spis treści

1. Dokumentacja użytkownika

Dokumentację kodu można wygenerować przy pomocy narzędzia doxygen, które na podstawie komentarzy w kodzie tworzy dokumentację.

2. Statystyki

2.1. Liczba linii kodu

- Biblioteka MQTT Client
  - *MQTT_Client.cpp* - 229 linijek
  - *Connection.h* - 36 linijek
  - *ConnectionUnscripted.h* - 49 linijek
  - *MQTT_Client.hpp* - 147 linijek
  - *SocketHandle.cpp* - 48 linijek
  - *Connection.cpp* - 25 linijek
  - *ConnectionUnscripted.cpp* - 105 linijek
  - Łącznie 639 linijek
- Biblioteka DataStore
  - *Data_Archive.cpp* - 25 linijek
  - *DataJSON.cpp* - 35 linijek
  - *DataStore.cpp* - 25 linijek
  - *DataArchive.hpp* - 30 linijek
  - *DataJSON.hpp* - 25 linijek
  - *DataStore.hpp* - 34 linijek
  - Łącznie 174 linijek
- Program Subscriber
  - *Subscriber.cpp* - 54 linijki
  - *SubscriberData.cpp* - 15 linijek
  - *subscriber_test_1.cpp* - 14 linijek
  - *subscriber_test_2.cpp* - 14 linijek
  - *Subscriber.hpp* - 15 linijek
  - *SubcriberData.hpp* - 8 linijek
  - Łącznie 120 linijek
- Program Publisher
  - *Publisher.cpp* - 41 linijek
  - *publiser_test_1.cpp* - 14 linijek
  - *publiser_test_2.cpp* - 14 linijek
  - Łącznie 69 linijek

Łącznie 1000 linijek kodu

2.2. Liczba testów

2.2.1. Biblioteka MQTT Client

Przygotowano dwa testy jednostkowe testujące próbę połączenia.
Pierwszy test sprawdza próbę połaczenia wspieranego - nieszyfrowanego.
Drugi test sprawdza próbę połączenia niespieranego - szyfrowanego.

2.2.2. Biblioteka DataStore

Przygotowano dwa testy jednostkowe pokazujące pomyślne skonfigurowanie środowiska.

2.2.3. Program Subscriber

Przygotowano dwa testy jednostkowe pokazujące pomyślne skonfigurowanie środowiska.

2.2.4. Program Publisher

Przygotowano dwa testy jednostkowe pokazujące pomyślne skonfigurowanie środowiska.

2.3. Procentowe pokrycie kodu testami

2.3.1. Biblioteka MQTT Client

Testy pokrywają jedną z ośmiu funkcji biblioteki, daje to 12,5 procent pokrycia.

2.3.2. Biblioteka DataStore

Testy pokrywają jedną z ośmiu funkcji biblioteki, daje to 12,5 procent pokrycia.

2.3.3. Program Subscriber

Testy pokrywają jedną z ośmiu funkcji biblioteki, daje to 12,5 procent pokrycia.

2.3.4. Program Publisher

Testy pokrywają jedną z ośmiu funkcji biblioteki, daje to 12,5 procent pokrycia.

2.4. Liczba godzin poświęcona na projekt

Zespół powięcił na projekt łącznie około 120 godzin.
Większą część poświęcono na konfigurację środowiska.

3. Opis napotkanych problemów i popełnionych błędów

Podczas projektu napotkaliśmy spore problemu z dołączaniem bibliotek za pomocą cmake do projektu. Spowodowało to spore utrudnienia w pracy, które wygenerowały duże opóźnienia. Początkowym problemem było zrozumienie i zaprojektowanie prawidłowej architektury dla sniffera MQTT. Nie przewidzieliśmy problemów z samą implementacją protokołu MQTT. Dotyczyły one zrozumienia dokumentacji i praktycznego zastosowania informacji w niej zawartych.