

|  |  |  |  |
| --- | --- | --- | --- |
| Projekt zespołowy | | | |
| **Kierunek** | *Informatyka* | **Termin** | *Czwartek 14:15* |
| **Temat** | *Projekt elastycznej aplikacji do zarządzania urządzeniami IoT w oparciu o bibliotekę QT* | **Zgłaszający** | *InterElcom* |
| **Skład grupy** | *Adam Krizar 241276*  *Katarzyna Czajkowska 242079*  *Mateusz Gurski 242089*  *Szymon Cichy 235093*  *Arkadiusz Cichy 236011* | **Nr grupy** | *-* |
| **Prowadzący** | *Dr inż. Jan Nikodem* | **data** | *9 kwiecień 2020* |

Spis treści

[1. Plan zadań 3](#_Toc36740130)

[ Grupa 1 3](#_Toc36740131)

[ Grupa 2 3](#_Toc36740132)

[ Grupa 3 3](#_Toc36740133)

[ Grupa 4 3](#_Toc36740134)

[2. Opis zadania 3](#_Toc36740135)

[3. Wymagania 4](#_Toc36740136)

[4. Założenia 4](#_Toc36740137)

[5. Środowisko 5](#_Toc36740138)

[5.1. Instalacyjne 5](#_Toc36740139)

[5.2. Programistyczne 6](#_Toc36740140)

[6. Sprzęt (TODO: Arkadiusz Cichy) 6](#_Toc36740141)

[7. Protokoły Sieciowe (TODO: Katarzyna Czajkowska) 6](#_Toc36740142)

[8. Biblioteka MQTT (TODO: Arkadiusz Cichy) 6](#_Toc36740143)

[9. Biblioteka HTTP (TODO: Szymon Gurski) 6](#_Toc36740144)

[10. Program na platformę Android (TODO: Adam Krizar/Szymon Cichy) 6](#_Toc36740145)

[11. Program na platformę Linux (TODO: Katarzyna Czajkowska) 6](#_Toc36740146)

[12. Oprogramowanie urządzenia IoT (TODO: Szymon Gurski) 7](#_Toc36740147)

[13. Kosztorys 7](#_Toc36740148)

[14. Plan realizacji 7](#_Toc36740149)

[14.1. Pierwszy punkt kontrolny [19.03] 7](#_Toc36740150)

[14.2. Drugi punkt kontrolny [02.04] 7](#_Toc36740151)

[14.3. Trzeci punkt kontrolny [23.04] 7](#_Toc36740152)

[14.4. Instalacja [07.05] 7](#_Toc36740153)

[14.5. Testy użytkownika [21.05] 7](#_Toc36740154)

[14.6. Oddanie projektu do użytku [04.06] 7](#_Toc36740155)

[14.7. Prezentacja naszych osiągnięć [10.06] 7](#_Toc36740156)

[15. Propozycja rozwoju systemu 7](#_Toc36740157)

[16. Źródła 9](#_Toc36740158)

# Plan zadań

Wykonawca: Adam Krizar

- Korekta dokumentacji i dodawanie nowych elementów  
- Zatwierdzanie prac.

### Grupa 1

Wykonawca: Mateusz Gurski  
Sprawdzenie: Arkadiusz Cichy

- Implementacja oprogramowania na wybrane urządzenie IoT do obsługi protokołu HTTP oraz wybranego czujnika. Program techniczny obsługi dla IoT - Instrukcja wgrywania utworzonego oprogramowania wraz z opisem użytych bibliotek oraz listingiem kodu wraz z komentarzami.

 - Implementacja obsługi protokołu HTTP w aplikacji desktopowej. Opis działania, użytych bibliotek oraz listing najważniejszych fragmentów kodu wraz z komentarzami

### Grupa 2

Wykonawca: Arkadiusz Cichy  
Sprawdzenie: Szymon Cichy

- Implementacja obsługi protokołu MQTT w aplikacji desktopowej. Opis działania, użytych bibliotek oraz listing najważniejszych fragmentów kodu wraz z komentarzami  
- Implementacja oprogramowania na wybrane urządzenie IoT do obsługi protokołu MQTT oraz wybranego czujnika. Program techniczny obsługi dla IoT - Instrukcja wgrywania utworzonego oprogramowania wraz z opisem użytych bibliotek oraz listingiem kodu wraz z komentarzami.

### Grupa 3

Wykonawca: Szymon Cichy/Adam Krizar  
Sprawdzenie: Katarzyna Czajkowska

- Implementacja interfejsu w aplikacji android oraz opis użytych funkcji do stworzenia projektu.  
- Implementacja komunikacji z urządzeniem IoT umożliwiających odbieranie prostych komunikatów wraz z opisem użytych bibliotek..

### Grupa 4

Wykonawca: Katarzyna Czajkowska  
Sprawdzenie: Mateusz Gurski

- Dopracowanie interfejsu graficznego ( obsługa przycisków, dodawanie nowego urządzenia, okno pomocy)  
- Opis interfejsu użytkownika (zrzuty ekranu, instrukcja obsługi, wykorzystane biblioteki)

# Opis zadania

Naszym zadaniem jest stworzenie aplikacji, która umożliwia komunikację z urządzeniami IoT zezwalając na zmianę protokołu komunikacji (elastyczność).

Stan początkowy określa jedynie platformy, które mamy wspierać oraz technologie które mają być wykorzystane do komunikacji z urządzenie IoT. Ze względu na bardzo mało precyzyjny opis wielu parametrów projektu jesteśmy zmuszeni samodzielnie doprecyzować wiele rzeczy takich jak na przykład wykorzystane protokoły sieciowe. Naszym zadaniem jest więc określenie następujących rzeczy:

- W jakiej wersji wykorzystać wymagane narzędzia  
- Określić w jakim środowisku oraz w jaki sposób urządzenia IoT będą komunikować się z naszą aplikacją. Określenie wymagań sieci, jak powinna być skonfigurowana i jakie wykorzystywać urządzenia.  
- Zdobycie informacji na temat wykorzystywanych protokołów, jak je obsłużyć oraz zaprogramować na różnych platformach.  
- Określenie czujnika oraz rodzaju mikrontrolera, który będzie służył do prezentacji możliwości naszej aplikacji.  
- Przygotowanie oprogramowanie dla testowanego urządzenia, które pozwoli mu współpracować z naszą aplikacją.

# Wymagania

Celem projektu jest utworzenie aplikacji działającej na kilku platformach w oparciu o bibliotekę QT i język C++. Jej elastyczność będzie polegała na możliwości zmiany protokołu komunikacji z urządzeniem IoT.

Wymagania, które powinna ona spełniać to:

 Użycie biblioteki QT oraz języka C++

 Stworzenie aplikacji działającej minimum na dwie platformy np. (Linux, Android).

 Stworzenie w aplikacji możliwości wyboru oraz sposobu dodawania nowych protokołów komunikacji z urządzeniem IoT.

 Obsługa w aplikacji minimum dwóch protokołów komunikacji z urządzeniem IoT np. (HTTP, MQTT).

# Założenia

Bazując na zgłoszonych wymaganiach opracowaliśmy następujące cele naszego projektu:

- Wymaganie wykorzystania biblioteki Qt: Wykorzystanie Qt w wersji 5.14 w wzwyż – Zapewnia wykorzystanie jak najdokładniejszych rozwiązań oraz gwarantuje dobre działanie na nowych systemach operacyjnych.

- Wymaganie obsługi dwóch platform: Wsparcie dla systemu Linux (ze względu na jego darmowość i łatwość instalacji na różnych urządzeniach) oraz dla systemu Android (obecnie najpopularniejsza platforma na urządzenia mobilne).

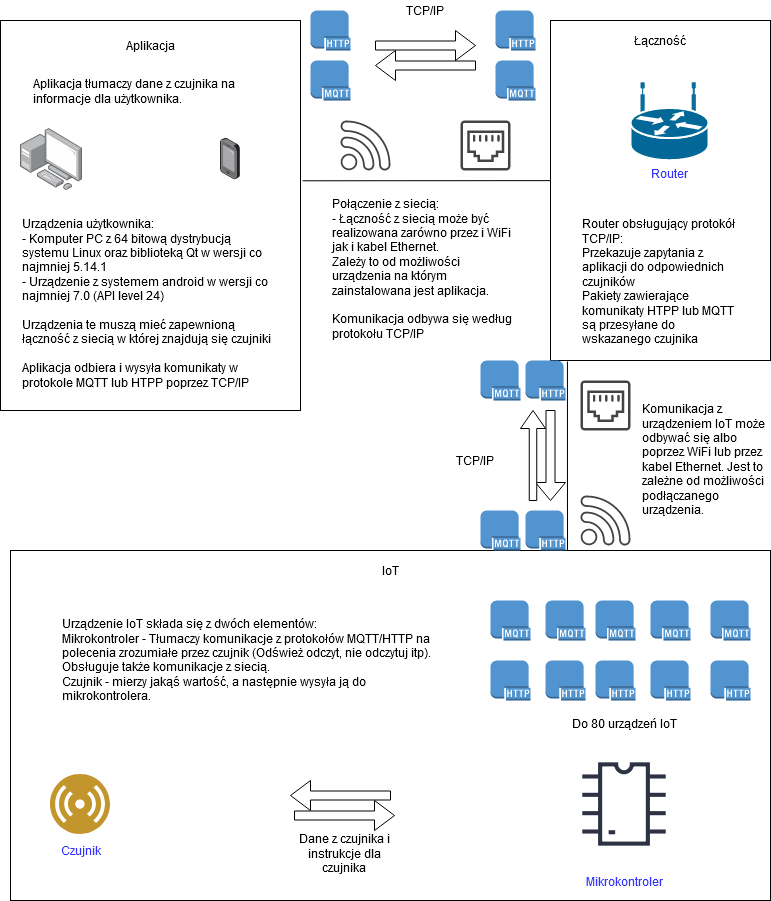
- Wymaganie implementacji minimum dwóch protokołów komunikacji z IoT: Implementacja protokołu HTTP oraz MQTT w naszej aplikacji oraz w testowym urządzeniu IoT. Te dwa protokoły zostały wyszczególnione jako przykładowe przez zgłaszającego oraz należą do najpopularniejszych rozwiązań na rynku co zapewni większą kompatybilność aplikacji.

- Wymaganie elastycznej aplikacji: Możliwość wyboru używanego protokołu komunikacji oraz przygotowanie możliwości dodania obsługi nowych protokołów)

- Komunikacja z IoT: Aplikacja będzie realizować komunikacje poprzez sieć lokalną, która może odbywać się po kablu lub bezprzewodowo z wykorzystaniem protokołu TCP/IP.

- Możliwość obsługi wielu IoT: Projekt aplikacji przewiduje obsługę do 80 urządzeń. Ta liczba zależy od możliwości wybranego routera obsługującego połączenia.

- Przygotowanie dwóch urządzeń IoT (Wykorzystanie gotowych rozwiązań takich jak mikrokontrolery Arduino i im podobne) w celu prezentacji możliwości aplikacji.



Rysunek . Ogólny diagram projektu.

# Środowisko

## Instalacyjne

Łączność między komputerami na których zainstalowana zostanie aplikacja a urządzeniami IoT będzie odbywać się przez sieć lokalną poprzez łącze przewodowe bądź z użyciem transmisji bezprzewodowej WiFi.

Wymagania sprzętowe dla naszej aplikacji są trudne do precyzyjnego określenia na etapie projektowym. Zakładamy jednak, że każdy sprzęt, na którym może działać nowoczesny system operacyjny (np. Android 8+, dystrybucje Linux tj. Ubuntu, Manjaro) będzie wystarczający.

## Programistyczne

Do budowy aplikacji wykorzystany zostanie język C++ i biblioteki Qt.

Framework Qt zostanie wykorzystany w najnowszej stabilnej wersji (na dzień 12.03.2020 jest to 5.14.1). Jest to zestaw narzędzi które pozwolą na stworzenie różnych interfejsów użytkownika na osobnych platformach, które to interfejsy będą spójne wizualnie oraz będą mogły przystosowywać się do różnic w konkretnych urządzeniach, jak np. dopasowanie elementów do rozmiarów ekranu. Ponadto zastosowanie bibliotek Qt pozwoli przyspieszyć tempo prac poprzez modularność kodu – to znaczy, nie będzie potrzeby przepisywania całego kodu przy przejściu na nową platformę.

Do tworzenia aplikacji desktopowej użyte zostaną narzędzia Qt Creator oraz QT Designer. Użycie ich usprawni utrzymanie aplikacji oraz wprowadzanie zmian w przyszłości. Wykorzystanie tych specjalnych środowisk poprawi jakość oraz obniży czas wykonania aplikacji, ponadto może skutkować niższymi kosztami obsługi w wypadku konieczności wprowadzenia zmian w interfejsie użytkownika.

Kończąc, użycie bibliotek Qt pozwoli na stworzenie kodu aplikacji który w spójny sposób obsługuje nie tylko interfejs użytkownika, lecz także obsługę protokołów komunikacji z urządzeniami.

Po stronie urządzeń IoT kod będzie napisany w języku C++ lub być może, w zależności od bieżących potrzeb, w innym języku jak np. skrypt Lua.

Wybór innych narzędzi programistycznych może nastąpić w trakcie wykonywania projektu i ich lista może zostać uzupełniona w późniejszej dacie.

W celu ułatwienia pracy w grupie wykorzystany zostanie system kontroli wersji. Repozytorium zostanie utworzone na platformie Github. Jest to sposób na centralizację zasobów w projekcie i ułatwi śledzenie zmian i postępu przez nie tylko programistów, lecz także zleceniodawców.

# Sprzęt (TODO: Arkadiusz Cichy)

# Protokoły Sieciowe (TODO: Katarzyna Czajkowska)

# Biblioteka MQTT (TODO: Arkadiusz Cichy)

# Biblioteka HTTP (TODO: Mateusz Gurski)

# Program na platformę Android (TODO: Adam Krizar/Szymon Cichy)

# Program na platformę Linux (TODO: Katarzyna Czajkowska)

# Oprogramowanie urządzenia IoT (TODO: Mateusz Gurski)

# Kosztorys

Autor Adam Krizar

Głównym kosztem w realizacji naszego projektu są urządzenia IoT koniecznie do testowania i prezentacji możliwości naszej aplikacji. Potrzebne są nam dwie platformy testowe:

Mikrokontroler ESP8266: <https://allegro.pl/oferta/esp8266-nodemcu-v3-wifi-2-4ghz-ch340-do-arduino-7241549772>, koszt 18,90 zł.

Czujnik DHT11: <https://allegro.pl/oferta/dht11-czujnik-temperatury-i-wilgotnosci-arduino-7487941486>, koszt 4,70 zł

Całkowity koszt w zależności od wybranej podstawki wynosi odpowiednio:

ESP8266: 47,20 zł

# Plan realizacji

## Pierwszy punkt kontrolny [19.03]

Implementacja prototypowej wersji aplikacji na system Linux. Zaimplementowanie protokołu http po stronie aplikacji.

## Drugi punkt kontrolny [02.04]

Rozwój aplikacji na system Linux. Przygotowanie pierwszego urządzenia IoT i przetestowanie działania protokołu HTTP. Implementacja protokołu MQTT (bez testów).

## Trzeci punkt kontrolny [23.04]

Przeniesie aplikacji na system android. Przygotowanie drugiego urządzania IoT oraz przetestowanie protokołu MQTT.

## Instalacja [07.05]

## Testy użytkownika [21.05]

Wprowadzenie ewentualnych korekt w projekcie interfejsu użytkownika zgodnie z uwagami użytkownika końcowego.

## Oddanie projektu do użytku [04.06]

## Prezentacja naszych osiągnięć [10.06]

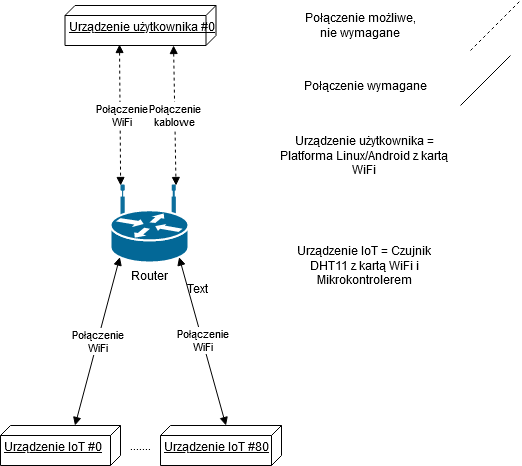
# Propozycja rozwoju systemu



Rysunek 2.. Obsługa protokołu HTTP



Rysunek 3.. Obsługa protokołu MQTT



Rysunek 4.. Ogólna propozycja użycia aplikacji

# Źródła

<https://store.arduino.cc/arduino-nano>

<https://www.qt.io/>

<https://store.arduino.cc/arduino-uno-rev3>

<https://learn.adafruit.com/dht>

<https://www.sparkfun.com/datasheets/Components/nRF24L01_prelim_prod_spec_1_2.pdf>

<https://en.wikipedia.org/wiki/ESP32>

<https://en.wikipedia.org/wiki/ESP8266>