

Smarthome

Raport z kamienia milowego na dzień 28 października 2022

Marcin Retajczyk Kamila Skorupka Piotr Sokołowski

Contents

| | |
|--|-----------|
| Opis projektu | 2 |
| Podział na moduły | 3 |
| Literatura | 4 |
| Artykuły opisujące podobne do planowanego systemu rozwiązania . . . | 4 |
| Artykuły przedstawiające problematykę, rozwiązania, implementacje i temu podobne zagadnienia użyteczne z punku widzenia projek- towanego systemu | 5 |
| Stos technologiczny | 6 |
| Mikrokontroler | 6 |
| MQTT listener | 6 |
| API | 6 |
| Aplikacja | 6 |
| Diagramy przypadków użycia | 8 |
| Diagram interfejsów | 10 |

Opis projektu

Realizowany przez nas projekt zakłada powstanie systemu inteligentnego domu, obsługującego odczyt danych z czujników, jak i sterowanie podłączonymi urządzeniami. Planujemy umożliwienie automatyzacji niektórych czynności, wzorując się na aplikacji IFTTT:



Figure 1: IFTTT - dodawanie nowego apletu

Proponowane przez nas rozwiązanie pozwalałoby na niemal dowolną konfigurację automatyzacji domu, przez ustawienie w pierwszej sekcji (If This) progu dla wartości odczytanej z danego czujnika oraz akcji dla podpiętego urządzenia w sekcji drugiej (Then That).

Zakładamy, że cały system będzie utrzymywany na naszych serwerach. Chcemy jednak dopuścić do użytku urządzenia stworzone przez użytkowników. W związku z tym, przy konfiguracji czujników w aplikacji, użytkownik sam wybierze, jakie dane otrzymuje i na jakich tematach MQTT. Możliwe będzie więc dodanie wielu czujników obsługiwanych przez jedno urządzenie.

Najpopularniejszym rozwiązaniem tego typu jest Google Home, który pozwala na integrację urządzeń obsługiwanych przez inne systemy. Ma jednak ograniczoną ilość rodzajów urządzeń, przez co przykładowo czujnik temperatury i wilgotności musi być dodany jako termostat. Pojawia się wtedy informacja o ustawionym chłodzeniu, chociaż urządzenie w rzeczywistości takiej funkcjonalności nie posiada.

Trzeba również wspomnieć o Home Assistant, który obsługuje więcej typów

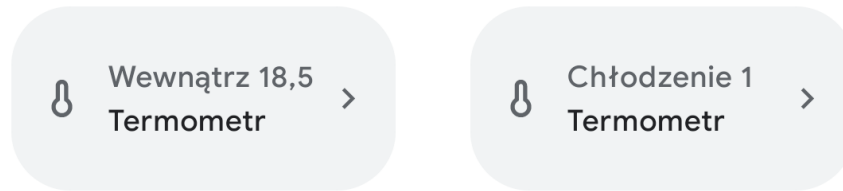


Figure 2: Czujnik DHT22 dodany do Google Home jako termostat

urządzeń, a także posiada możliwość ustawiania scen i automatyzacji. Ponieważ jest to projekt open-source, może się też pochwalić ogromnym wsparciem społeczności. Niestety, jest on trudny w konfiguracji.



Figure 3: Home Assistant (fot. Filip Tomczyk)

Podział na moduły

Podział na moduły:

- Urządzenia
 - Aplikacja bazowa
 - * Konfiguracja urządzenia przez Wi-Fi (tryb accesspoint)
 - * Usypianie/wybudzanie urządzenia
 - * Odczyt danych z czujnika przez bridge (generycznie!)
 - * Publikowanie danych do brokera MQTT

- Biblioteka czujnika (gotowa biblioteka + plik nagłówkowy z informacjami o dostępnych danych)
- Bridge (wspólny sposób czytania danych z różnych czujników)
- Webapp
 - GUI z pomiarami/wykresami
 - Ustawienia użytkownika (dane kontaktowe, hasło, username, avatar)
 - Ustawienia domu (dodawanie/usuwanie pomieszczeń, tworzenie/anulowanie zaproszeń)
- API
 - Endpointy z aktualnymi pomiarami (REST API - GET)
 - Endpointy z historycznymi danymi (REST API - GET)
 - Ustawienia użytkownika
 - Ustawienia domu
 - Zadania cron

Literatura

Artykuły opisujące podobne do planowanego systemu rozwiązanie

Data centre temperature monitoring with ESP8266 based Wireless Sensor Network and cloud based dashboard with real time alert system

Dzięki temu artykułowi dowiadujemy się jak monitorować **temperaturę i wilgotności** w różnych punktach lokalizacji i zrobienie alertu gdy temperatura przekroczy pewną wartość. Rozwiązanie na płytce ESP8266, Publikowanie danych z czujników w czasie rzeczywistym.

Wykorzystanie: W naszym projekcie chcielibyśmy zastosować funkcjonalność monitorowania temperatury i wilgotności. Którą będzie można w łatwy sposób sprawdzić w aplikacji. Artykułu użyliśmy dlatego, że problem został rozwiązany na płytce ESP8266, której używamy w projekcie oraz system alertów, który również chcemy zaimplementować.

Link: <https://ieeexplore.ieee.org/document/8073958>

Arduino-based smart irrigation using water flow sensor, soil moisture sensor, temperature sensor and ESP8266 WiFi module

Artykuł pokazuje monitorowanie **wilgotności, temperatury** potrzebnych różnym uprawom za pomocą czujników. Dane są zbierane i odbierane przez arduino, które mogą być połączone z interaktywną stroną internetową, która pokazuje wartości w czasie rzeczywistym wraz z wartościami standardowymi różnych czynników wymaganych przez uprawy.

Wykorzystanie: Kolejny przykład zastosowania monitorowania temperatury i wilgotności. Artykułu użyliśmy dlatego, że problem został rozwiązany na płytce

ESP8266 oraz Arduino którego używamy w projekcie.

Link: <https://ieeexplore.ieee.org/abstract/document/7906792>

Remote Access Weather Monitoring System Based on Soft Real-Time System (SRTS)

Artykuł wykorzystujący działanie **barometru** BMP280. Aplikacja informujące o warunkach pogodowych w czasie rzeczywistym. Dane zebrane są porównywane z danymi z kilku aplikacji prognozujących pogodę w niektórych mediach. Dane są przesyłane i przechowywane w bazie danych, a następnie wyświetlane w aplikacji internetowej i mobilnej. Na podstawie tych informacji, użytkownik może uzyskać informacje bezpośrednio w czasie rzeczywistym.

Wykorzystanie: W naszej aplikacji chcemy mieć również informację o warunkach w czasie rzeczywistym, w artykule został wykorzystany czujnik BMP280, który posiadamy i chcielibyśmy wykorzystać w projekcie. Oprócz tego, zainteresowało nas samo działanie aplikacji m.in przesyłanie, przechowywanie i przetwarzanie danych.

Link: <https://ieeexplore.ieee.org/document/8711997>

Development of a prototype smart home intelligent lighting control architecture using sensors onboard a mobile computing system

W artykule przedstawiono prototyp inteligentnego systemu oświetleniowego. Jest to aplikacja mobilna, która wykorzystywała wbudowany **czujnik światła** otoczenia do uruchomienia algorytmu sprzężenia zwrotnego w pętli zamkniętej w celu wdrożenia zbierania światła dziennego.

Wykorzystanie: W naszej aplikacji planujemy wykorzystanie czujnika światła. W artykule został wykorzystany właśnie taki czujnik w połączeniu z IoT, okaże się to pomocne podczas implementacji rozwiązania sterowania roletami w pomieszczeniu.

Link: <https://www.sciencedirect.com/science/article/pii/S0378778816319971>

Artykuły przedstawiające problematykę, rozwiązania, implementacje i temu podobne zagadnienia użyteczne z punktu widzenia projektowanego systemu

IOT Based on Garbage Monitoring System

Urządzenie IoT - Wykorzystanie PHP do stworzenia aplikacji webowej, Arduino, ESP8266, Wifi, Bazy danych

Link: <https://ijsrcseit.com/paper/CSEIT172289.pdf>

ESP8266 based implementation of wireless sensor network with Linux based web-server

Urządzenie IoT - ESP8266 + Apache+ PHP + SQLite3.

Link: <https://ieeexplore.ieee.org/abstract/document/7570919>

A Design of Greenhouse Monitoring System Based on Low-Cost Mesh Wi-Fi Wireless Sensor Network

Urządzenie IoT - MongoDB + esp8266, PlatformIO i Arduino

Link: <https://ieeexplore.ieee.org/abstract/document/9422655>

IoT based smart home automation system using sensor node

Urządzenie IoT - ESP8266 + Arduino, PHP, temperature

Link: <https://ieeexplore.ieee.org/abstract/document/8389037>

Stos technologiczny

TODO

Mikrokontroler

Docelową implementacją powstanie w frameworku Arduino (C++). Alternatywą dla naszego sprzętu (ESP8266) jest MicroPython, jednak na korzyść Arduino przemawia spora kolekcja bibliotek do czujników.

Wstępnie zastąpimy fizyczne urządzenia skryptami napisanymi w języku JavaScript lub Python. Oba te języki pozwalają na szybką implementację algorytmu publikowania danych przez MQTT.

MQTT listener

Implementacja powstanie w języku JavaScript (NodeJS) lub Python. Oba pozwalają na wygodną obsługę MQTT oraz bazy danych.

API

Zostanie napisane z wykorzystaniem frameworku Symfony w języku PHP 8 wraz z dodatkami PHPStan do analizy statycznej kodu i PHPUnit do testowania kodu. Alternatywą był język Python (frameworki Django oraz Flask).

Aplikacja

Planujemy stworzyć Single-Page Application, do czego możemy wykorzystać rozmaite frameworki JavaScript, jak AngularJS, React czy Vue.js, jednak nie

wykluczamy klasycznej aplikacji z podstronami. W tym wypadku mogłaby to być wspólna aplikacja z API (monolit?), bądź pracująca niezależnie.

Diagramy przypadków użycia

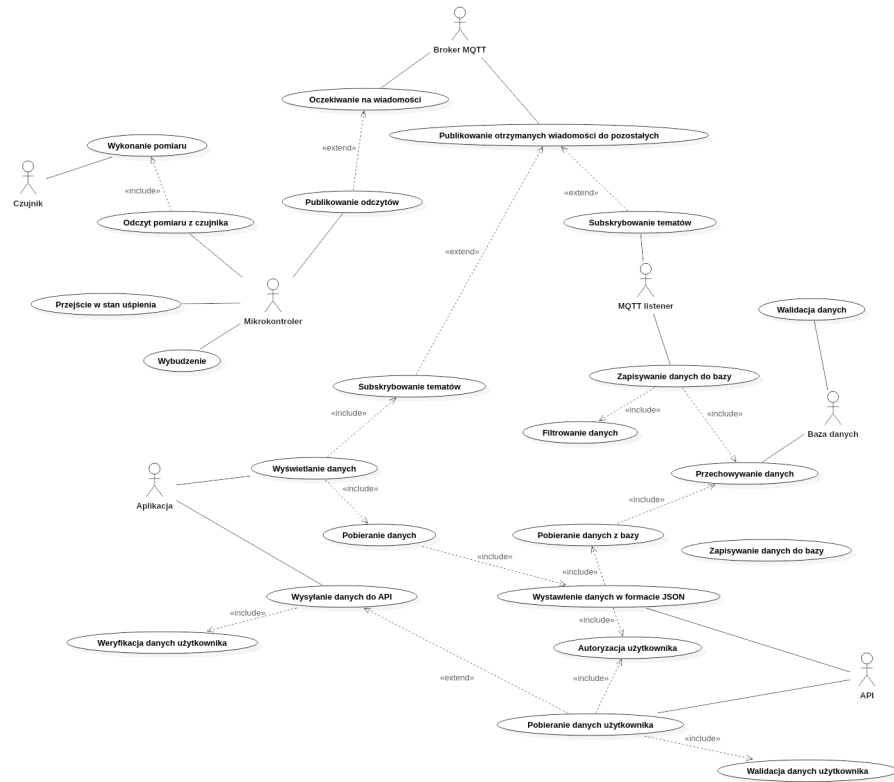


Figure 4: Diagram przypadków użycia - odczyty z czujnika

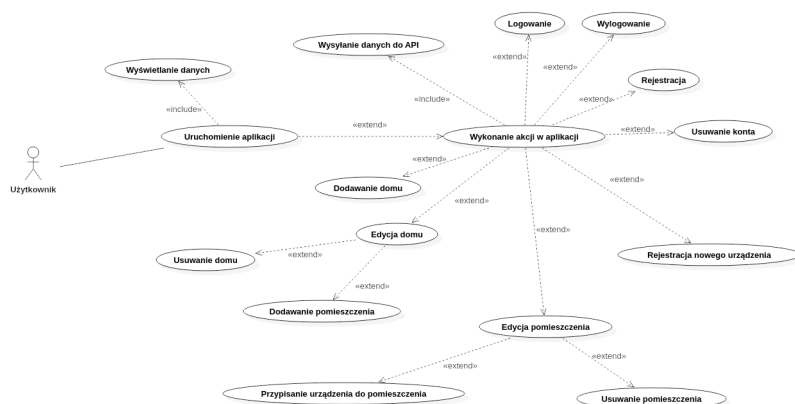


Figure 5: Diagram przypadków użycia - akcje użytkownika

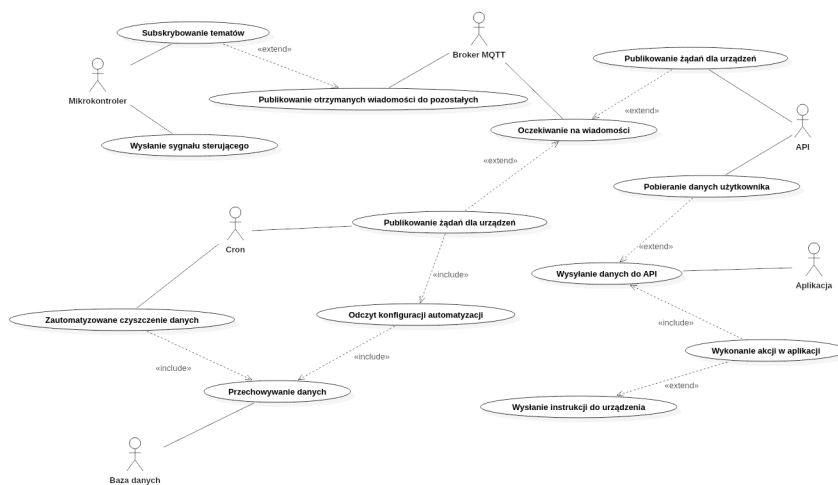


Figure 6: Diagram przypadków użycia - sterowanie urządzeniami

Diagram interfejsów

TODO