

**Politechnika Warszawska**

W Y D Z I A Ł E L E K T R O N I K I  
I T E C H N I K I N F O R M A C Y J N Y C H



Instytut Mikroelektroniki i Optoelektroniki

# Praca dyplomowa inżynierska

na kierunku Elektronika  
w specjalności Elektronika i Inżynieria Komputerowa

Konstrukcja dedykowanej stacji usługowej Internetu Rzeczy z  
zastosowaniem komputera jednopłytkowego.

**Marcin Michał Piotrowski**

Numer albumu 300346

promotor  
dr hab. inż. Michał Borecki, prof. uczelni

WARSZAWA 2022



## **Streszczenie**

**Tytuł:** Konstrukcja dedykowanej stacji usługowej Internetu Rzeczy z zastosowaniem komputera jednopłytkowego.

Niniejsza praca ma na celu konstrukcje urządzenia służącego do integracji w sposób bezprzewodowy węzłów sieci Internetu Rzeczy, które jest określone mianem stacji usługowej oraz przeprowadzenie jego weryfikacji eksperymentalnej w zmieniających się warunkach.

W ramach projektu skonfigurowano komputer jednopłytkowy, aby pełnił funkcje serwera, stworzono dedykowane oprogramowanie oraz dokonano projektu węzła wraz z jego konstrukcją. Następnie porównano wpływ zakłóceń innymi sygnałami oraz fizycznych przeszkód obecnych w bloku mieszkalnym i jednorodzinnym domu w celu ustalenia warunków niezawodnej komunikacji bezprzewodowej.

**Słowa kluczowe:** Internet Rzeczy, komputer jednopłytkowy, łączność bezprzewodowa

## **Abstract**

**Title:** Construction of a dedicated Internet of Things service station using a single board computer.

The aim of this work is to construct a device for wireless integration of the Internet of Things nodes, which is referred to as a service station, and to conduct its experimental verification under changing conditions.

As part of the project, a single-board computer was configured to act as a server, dedicated software was created and the node was designed together with its construction. They then compared the interference effects of other signals and the physical obstacles present in the apartment block and the single-family house to establish conditions for reliable wireless communication.

**Keywords:** Internet of Things, single board computer, wireless connectivity



„załącznik nr 3 do zarządzenia nr 24/2016 Rektora PW

.....  
miejscowość i data

.....  
imię i nazwisko studenta

.....  
numer albumu

.....  
kierunek studiów

## OŚWIADCZENIE

Świadomy/-a odpowiedzialności karnej za składanie fałszywych zeznań oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie, pod opieką kierującego pracą dyplomową.

Jednocześnie oświadczam, że:

- niniejsza praca dyplomowa nie narusza praw autorskich w rozumieniu ustawy z dnia 4 lutego 1994 roku o prawie autorskim i prawach pokrewnych (Dz.U. z 2006 r. Nr 90, poz. 631 z późn. zm.) oraz dóbr osobistych chronionych prawem cywilnym,
- niniejsza praca dyplomowa nie zawiera danych i informacji, które uzyskałem/-am w sposób niedozwolony,
- niniejsza praca dyplomowa nie była wcześniej podstawą żadnej innej urzędowej procedury związanej z nadawaniem dyplomów lub tytułów zawodowych,
- wszystkie informacje umieszczone w niniejszej pracy, uzyskane ze źródeł pisanych i elektronicznych, zostały udokumentowane w wykazie literatury odpowiednimi odnośnikami,
- znam regulacje prawne Politechniki Warszawskiej w sprawie zarządzania prawami autorskimi i prawami pokrewnymi, prawami własności przemysłowej oraz zasadami komercjalizacji.

Oświadczam, że treść pracy dyplomowej w wersji drukowanej, treść pracy dyplomowej zawartej na nośniku elektronicznym (płycie kompaktowej) oraz treść pracy dyplomowej w module APD systemu USOS są identyczne.

.....  
czytelny podpis studenta”

# Spis treści

<b>1. Wstęp</b>	6
1.1. Cele pracy	6
<b>2. Internet rzeczy od strony użytkownika</b>	7
2.1. Zastosowania Internetu Rzeczy	7
2.2. Przegląd technologii	8
2.3. Koncepcje komunikacji bezprzewodowej	9
2.3.1. Rodzaje sieci	9
2.3.2. Topologie sieci	10
2.4. Standardy komunikacji bezprzewodowej	11
2.4.1. Wi-Fi	12
2.4.2. Bluetooth	14
2.5. Prywatność i bezpieczeństwo	14
<b>3. Cel projektu</b>	16
<b>4. Dobór technologii</b>	17
4.1. Stacja usługowa	17
4.1.1. Sprzęt	17
4.1.2. Oprogramowanie	19
4.2. Węzeł	21
4.3. Protokół komunikacyjny	21
<b>5. Realizacja stacji usługowej</b>	23
5.1. Aplikacja internetowa	23
5.2. Baza danych	24
5.3. Łączność bezprzewodowa	26
5.4. Węzeł	28
<b>6. Testowanie</b>	30
6.1. Testy niezawodności łączności bezprzewodowej	30
6.1.1. Układ pomiarowy	30
6.1.2. Środowisko testowe	31
6.1.3. Pomiary	35
6.1.4. Wnioski	36
6.2. Testy wersji demonstracyjnej stacji usługowej	37
<b>7. Podsumowanie</b>	40
<b>Bibliografia</b>	41
<b>Spis rysunków</b>	43

<b>Spis tabel</b> . . . . .	44
<b>Listingi kodu</b> . . . . .	45

# 1. Wstęp

Internet Rzeczy jest to koncepcja wedle której rozmaite przedmioty mają pozyskiwać i przetwarzać dane oraz komunikować się ze sobą w celu współpracy. Wprowadzenie tego typu rozwiązań ma na celu usprawnienie wszystkich dziedzin naszego życia poprzez optymalizację kosztów i zużycia energii. W literaturze i biznesie idea ta najczęściej określana jest anglojęzycznym skrótwcem IoT (Internet of Things).

Analizując dane rynkowe można zaobserwować gwałtowny wzrost popularności Internetu Rzeczy. W 2017 wartość światowego rynku IoT przekroczyła 100 miliardów dolarów, z czego prognozowane jest, że w 2024 będzie wynosić ponad bilion dolarów [9]. Jest to tym bardziej imponujące, jeśli weźmiemy pod uwagę jak młoda jest ta koncepcja. Według raportu Cisco Systems [1] Internet rzeczy powstał na przełomie 2008 i 2009 roku, kiedy liczba połączonych ze sobą „rzeczy” przewyższyła liczbę ludzi na ziemi. Obecną sytuację zawdzięczamy przede wszystkim dynamicznemu rozwojowi takich dziedzin nauki jak elektronika, telekomunikacja i informatyka. To właśnie dzięki nim możliwe jest skonstruowanie urządzeń dostarczających kluczowe dane, komunikacja przy użyciu zaawansowanych protokołów i przetwarzanie ogromnej ilości informacji w chmurze. W efekcie na rynku pojawiło się mnóstwo urządzeń określanych mianem inteligentnych (ang. smart) które potrafią porozumiewać się ze sobą i światem zewnętrznym. Prognozowane jest, że do 2025 roku na świecie będzie około 25 miliardów urządzeń określanych mianem IoT. Tak prężny rozwój może oznaczać, że Internet Rzeczy zdominuje praktycznie każdą dziedzinę naszego życia tak jak zrobił to dotychczas Internet ludzi.

Obecna sytuacja stanowi ogromne wyzwanie dla konstruktorów, aby dostarczyć użytkownikom gotowe rozwiązania, które umożliwią im czerpanie korzyści jakie niesie za sobą IoT. Jednym z problemów, który wymaga rozwiązania jest integracja ogromnej liczby urządzeń i tym samym umożliwienie użytkownikom tworzenie własnych inteligentnych systemów. Rozwiązaniem tego problemu jest stacja usługowa Internetu Rzeczy, której konstrukcja jest tematem niniejszej pracy. Stacja taka stanowi interfejs dla człowieka tak aby mógł on łatwo stworzyć inteligentną sieć i pozyskiwać z niej potrzebne informacje.

## 1.1. Cele pracy

- Zaprojektowanie stacji usługowej Internetu Rzeczy z zastosowaniem komputera jednopłytkowego,
- Konstrukcja wersji demonstracyjnej,
- Przeprowadzenie testów w celu sprawdzenia czy zostały spełnione założenia projektu.

## **2. Internet rzeczy od strony użytkownika**

### **2.1. Zastosowania Internetu Rzeczy**

Główne założenie Internetu Rzeczy w postaci sieci komunikujących się ze sobą urządzeń sprawia, że może znaleźć on zastosowanie praktycznie wszędzie. W kolejnych latach rozwoju tej technologii będą wynajdowane kolejne jej zastosowania niemniej jednak już można wyróżnić kilka najpopularniejszych.

Pierwszym najbardziej popularnym zastosowaniem IoT jest użycie go w gospodarstwie domowym. Większość naszego życia spędzimy właśnie w domu, dlatego nawet niewielkie usprawnienia w zakresie oszczędzania energii bądź automatyzacji mogą przynieść znaczne korzyści. W obecnych czasach w prawie każdym domu obecna jest bez przewodowa sieć wifi która stanowi świetny kanał komunikacji dla wszelkiego rodzaju urządzeń. Inteligentny dom charakteryzuje się pełną automatyką w zakresie sterowania oświetleniem, roletami bądź termostatem. Wszystko odbywa się w oparciu o zdalnie sterowane przekaźniki, dane z czujników bądź zewnętrznych źródeł. Możliwe również jest bieżące monitorowanie z dowolnego miejsca na ziemi informacji takich jak które okna i drzwi są otwarte, zużycie mediów czy obraz z monitoringu [2]. Wszystkie urządzenia takiego systemu są integrowane i wchodzą w interakcje z użytkownikiem poprzez dedykowaną stację usługową Internetu Rzeczy.

Kolejnym obszarem, w którym Internet Rzeczy może przynieść wiele korzyści jest Rolnictwo. Dla rolnika kluczowe jest zapewnienie roślinom dobrych warunków do wzrostu poprzez odpowiednie nawożenie i nawodnienie. Internet Rzeczy oferuje rozwiązanie tego problemu w postaci sitaki czujników umieszczonych głęboko w glebie monitorujących jej parametry [3]. Na podstawie informacji z takiego systemu oraz zewnętrznych danych meteorologicznych możliwe jest optymalne zaplanowanie nawadniania oraz ograniczenie ilości środków ochrony roślin do niezbędnego minimum.

Monitorowanie parametrów gleby, powietrza i wody może nieść również za sobą szersze korzyści. Innym zastosowaniem Internetu Rzeczy jest przewidywanie nadchodzących klęsk żywiołowych takich jak powódzie, trzęsienia ziemi czy huragany [3]. Architektura służących w tym celu systemów ma postać sieci, w której węzłami są komunikujące się ze sobą czujniki. W ten sposób możliwe jest pokrycie dużych obszarów tak aby wykrywać zbliżające się niebezpieczeństwa z odpowiednim wyprzedzeniem i zapewnić okolicznej ludności czas na ewakuacje. Innym obszarem naszego życia, w którym Internet Rzeczy może okazać się pomocny jest medycyna. Wiele ludzi dopiero dostrzega, że ma problemy ze zdrowiem, gdy ich stan znacząco się pogorszy, dlatego tak ważnym zadaniem jest ciągłe monitorowanie parametrów życiowych takich jak ciśnienie krwi, tętno czy poziom glukozy. Według koncepcji IoT może się to odbywać za pomocą urządzeń noszonych bezpośrednio na ciele pacjenta. Urządzenia takie łączą się z In-

## **2.2. Przegląd technologii**

---

ternetem poprzez standard LTE lub w przypadku chęci uzyskania niskiego poboru prądu można wykorzystać identyfikację radiową (RFID). Rozwiązania takie są w stanie częściowo wyręczyć lekarza i zmniejszyć ilość okresowych wizyt oraz dostarczyć informacje o nagłym pogorszeniu stanu zdrowia pacjenta [4].

Ostatnim obszarem, w którym w którym Internet Rzeczy powoli zaczyna zmieniać rzeczywistość jest przemysł. Dzieje się to za sprawą trwającej czwartej rewolucji przemysłowej (ang. Industry 4.0) która stanowi koncepcje według której nastąpi automatyzacja większości procesów przemysłowych między innymi za sprawą IoT. Urzeczywistnieniem tej idei jest inteligentna fabryka, w której maszyny tworzące linię produkcyjną są sterowane przez oprogramowanie, które podejmuje decyzje na podstawie danych z czujników bez ludzkiej interwencji. Przykładem tego, że ta rewolucja już się zaczęła jest chociażby złożony proces palenia kawy, który udało się zautomatyzować dzięki rozwiązaniom, które oferuje Internet Rzeczy [5].

## **2.2. Przegląd technologii**

Szerokie zastosowanie Internetu Rzeczy stawia różnie wymagania sprzętowe w kwestii interakcji z maszynami. Z tego powodu w tej dziedzinie stosuje się wszelkiego rodzaju systemy wbudowane ze względu na ich uniwersalność i różnorodność. Urządzenia stosowane w IoT można podzielić na trzy grupy, gdzie w skład pierwszej grupy wchodzą systemy o najmniejszej mocy obliczeniowej i ilości pamięci operacyjnej, z kolei urządzenia w trzeciej grupie dorównują mocą obliczeniową komputerom osobistym [6].

Pierwsza grupa składa się z urządzeń, które charakteryzują się zazwyczaj 8 bądź 16 bitową architekturą procesora i pamięcią RAM liczoną w dziesiątkach kilobajtów. Oznacza to, że niemożliwe jest na nich uruchomienie systemu operacyjnego takiego jak Windows bądź Linux, więc do programowania takich urządzeń wykorzystuje się niskopoziomowe języki jak na przykład c. Tak mocno ograniczone zasoby sprzętowe sprawiają, że urządzenia z tej grupy posiadają niską cenę oraz zużycie energii i wymiary. Stosuje się je głównie w czujnikach, gdzie jedynym zadaniem jest dokonanie pomiaru i przesłanie jego wyniku dalej poprzez określony protokół.

Urządzenia w drugiej grupie to również systemy mikroprocesorowe z tą różnicą, że posiadają większą moc obliczeniową względem pierwszej grupy. W przypadku tych urządzeń parametry takie jak taktowanie procesora i ilość pamięci RAM przyjmują wartości rzędu setek mega herców i kilobajtów. Jest to wystarczająca ilość zasobów, aby możliwe było przetwarzanie zebraanych danych w celu na przykład rozpoznawania obrazu lub dźwięku. Ceny urządzeń z tej grupy są równie niskie jak w przypadku pierwszej grupy, lecz wyższe taktowanie procesora powoduje większe zużycie energii co może być problemem, jeśli zależy nam na zasilaniu baterijnym.

Trzecia grupa składa się z jednopłytkowych komputerów, które pomimo wymiarów karty płatniczej posiadają podobną ilość zasobów do komputerów osobistych. Oznacza to, że można na nich uruchomić system operacyjny taki jak Windows bądź Linux przez co do ich programowania da się wykorzystać wysokopoziomowe języki jak na przykład Python. Tak duża moc obliczeniowa umożliwia wykonywanie skomplikowanych zadań takich jak algorytmy szyfrowania i kryptografii bądź pełnienie funkcji serwera. Urządzenia z tej grupy często są stosowane jako stacje usługowe Internetu Rzeczy, gdyż dobrze sobie radzą z obsługą dużej ilości zadań

na raz takich jak komunikacja z innymi urządzeniami, przetwarzanie danych i obsługa żądań. Zużycie energii w tym przypadku jest największe, na przykład najpopularniejszy jednopłytkowy komputer dostępny na rynku, czyli Raspberry Pi potrzebuje średnio 4 watów mocy, dlatego musi być zasilany poprzez sieciowy adapter.

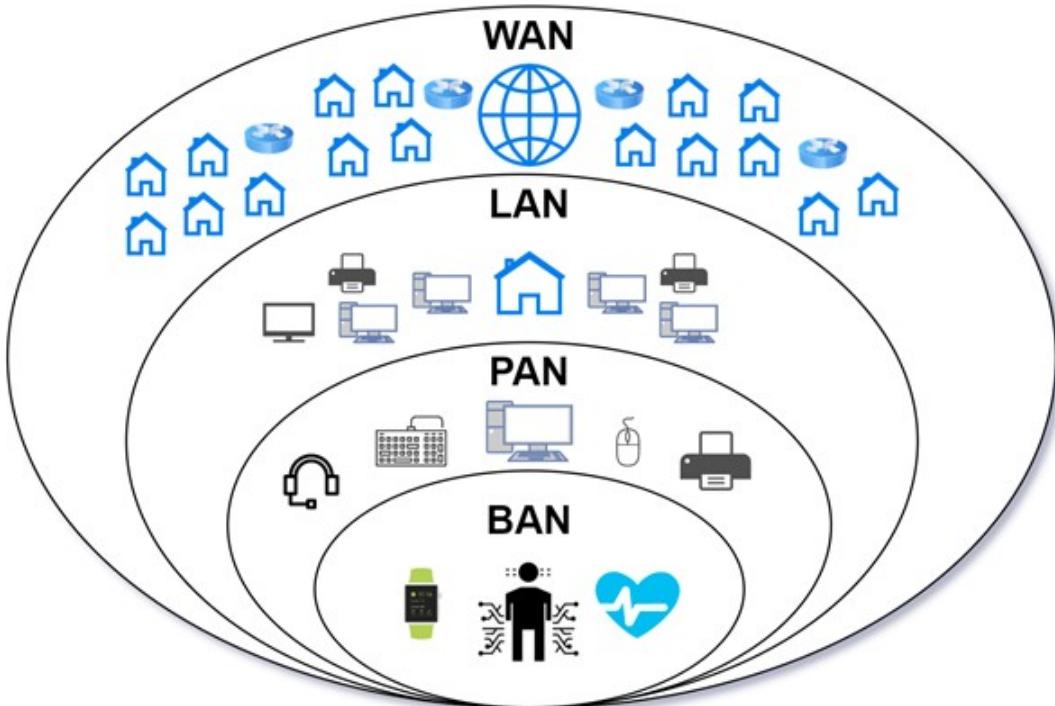
## 2.3. Koncepcje komunikacji bezprzewodowej

Nieodłącznym elementem Internetu Rzeczy jest łączność bezprzewodowa, dlatego warto przybliżyć związane z nią koncepcje. Wszystkie z nich cechują się innym zastosowaniem z uwagi na osiągalne parametry takie jak zasięg, zużycie energii i koszt przez co znajdują zastosowanie w różnych dziedzinach życia.

### 2.3.1. Rodzaje sieci

Sieci bezprzewodowe wykorzystywane między innymi w Internecie Rzeczy można klasyfikować ze względu na zasięg i rodzaju urządzeń, które ze sobą łączą. Poniżej zestawiono kilka ich rodzajów począwszy od tych o najmniejszym zasięgu i skończywszy na największym.

- Sieć typu BAN (ang. Body Area Network) jest to sieć tworzona z urządzeń noszonych przez człowieka [12]. Mogą to być wszczepiane w ciało implanty, urządzenia przytwierdzane do powierzchni ciała lub noszone w kieszeni [13]. Interakcja takiej sieci ze światem zewnętrznym odbywa za pośrednictwem urządzenia, które ma bezpośredni dostęp do Internetu, czyli tzw. bramki, dzisiaj najczęściej jest to smartfon.
- Sieć prywatna PAN (ang. Personal Area Network) łączy urządzenia w obrębie miejsca pracy pojedynczej osoby [14]. Przykładem takiej sieci może być komputer wraz z bezprzewodowymi urządzeniami peryferyjnymi takimi jak mysz i klawiatura. Jeśli komputer ma dostęp do Internetu pełni wtedy również pełni funkcje bramki.
- Lokalna sieć LAN (ang. Local Area Network) łączy ze sobą co najmniej dwa komputery na ograniczonym obszarze takim jak gospodarstwo domowe, uczelnia bądź biurowiec [15]. Tym co wyróżnia ją spośród wcześniejszych wymienionych sieci jest to, że wykorzystuje ona pakiet protokołów internetowych TCP/IP. Dzięki temu lokalna sieć daje namiastkę „zwykłego” Internetu, gdyż korzysta się z niej przy użyciu takich samych narzędzi. Istnieją również odmiany sieci LAN jak na przykład MAN (ang. Metropolitan Area Network) czyli sieci o zasięgu całych miast, lecz zasada ich działania pozostaje taka sama.
- Globalna Sieć WAN (ang. Wide Area Network) jest to sieć, która obejmuje swoim zasięgiem całe państwa czy kontynenty. Największym i najbardziej znanym przykładem sieci WAN o zasięgu prawie całego globu jest Internet [16]. Jak łatwo się domyślić wykorzystuje on pakiet protokołów internetowych TCP/IP przez co można również podłączać do niego mniejsze, lokalne sieci LAN o podobnej architekturze.

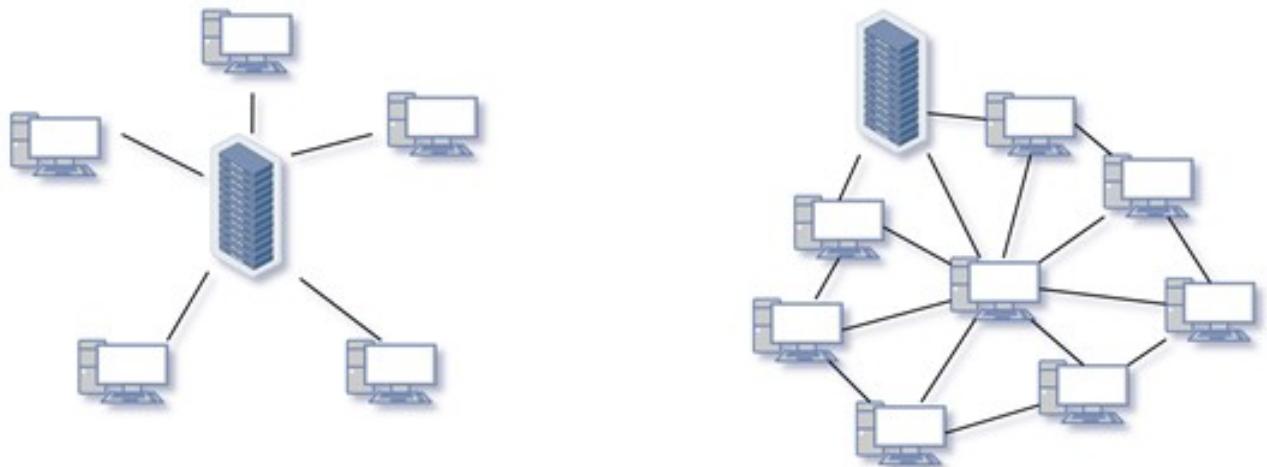


Rysunek 2.1. Diagram różnych rodzajów sieci

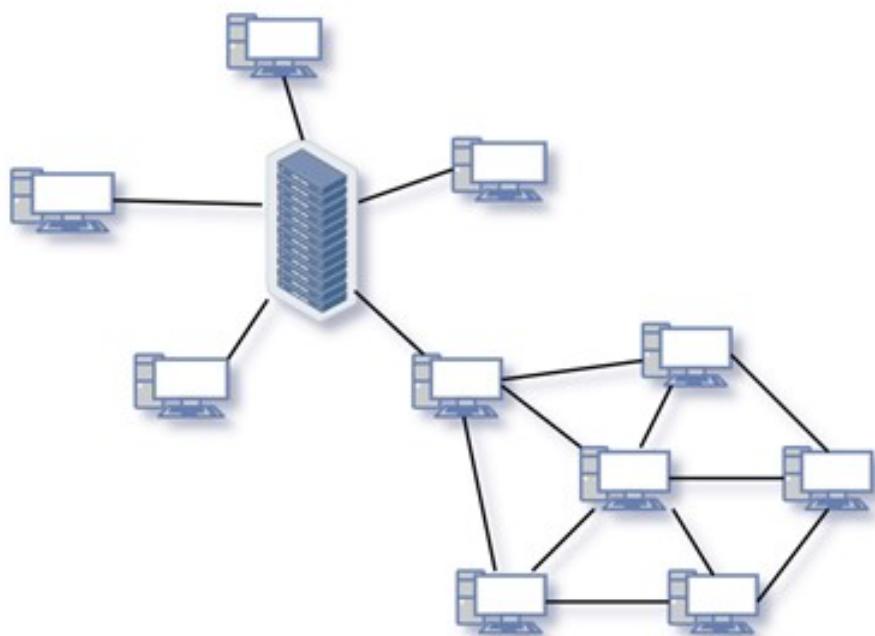
### 2.3.2. Topologie sieci

Innym kryterium, na które można podzielić sieci Internetu Rzeczy i nie tylko jest sposób w jaki węzły komunikują się ze sobą. Poszczególne konfiguracje są nazywane topologiami wraz ze kształtem, który najlepiej oddaje ich wygląd. Poniżej opisano najczęściej spotykane rodzaje.

- **Gwiazda** – w tej topologii wszystkie węzły są podłączone do jednego punktu, gdzie punkt centralny pełni również funkcje bramki do świata zewnętrznego. Jej główną zaletą jest prostota, oraz fakt, że awaria peryferyjnego węzła nie ma wpływu na działanie całej sieci. Z drugiej strony prostota może okazać się wadą, ponieważ w przypadku awarii centralnego punktu cała sieć przestaje działać.
- **Siatka** – w tym przypadku każdy z węzłów ma połączenie z paroma sąsiednimi węzłami. Zaletą tej topologii jest to, że punkt centralny pełniący funkcje bramki nie musi mieć bezpośredniego połączenia z każdym węzłem, więc można uzyskać pokrycie dużego obszaru niższym kosztem niż w przypadku gwiazdy. Inną korzyścią jest większa niezawodność komunikacji, gdyż zawsze istnieje droga alternatywna w przypadku awarii jednego węzłów. Potencjalną wadą może być skomplikowany proces trasowania pakietów, który wymaga większej mocy obliczeniowej co przekłada się na większy pobór prądu.
- **Mieszana** – jest to połączenie dwóch lub więcej topologii w jedną sieć. Stosuję się ją w celu wykorzystania zalet bądź ominięcia ograniczeń danej topologii w newralgicznych punktach.



Rysunek 2.2. Przykłady topologii sieci, po lewej stronie topologia gwiazdy, po prawej stronie topologia siatki



Rysunek 2.3. Topologia mieszana

## 2.4. Standardy komunikacji bezprzewodowej

W obecnych czasach stosowane są rozmaite standardy komunikacji bezprzewodowej z czego wszystkie oparte są o fale elektromagnetyczne w zakresie częstotliwości od 3 kHz do 3000GHz

## 2.4. Standardy komunikacji bezprzewodowej

---

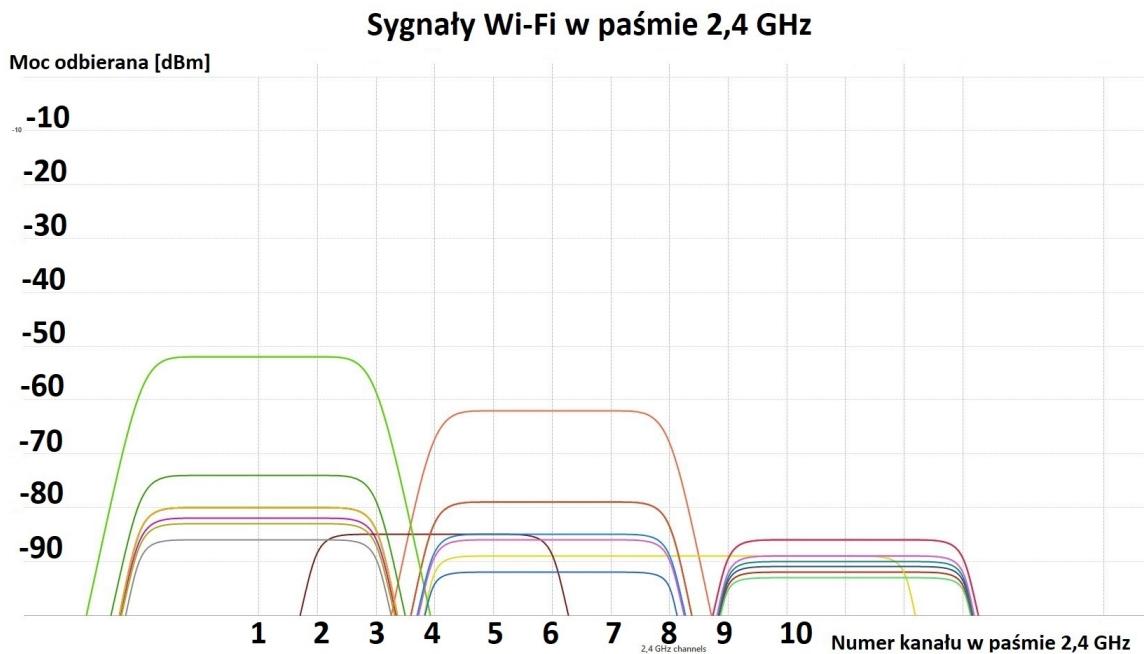
[17] zwane również falami radiowymi. Z tego powodu trzeba być świadomym wykorzystywanej częstotliwości, ponieważ tylko wąski zakres spośród wszystkich użytecznych pasm został wydzielony przez ministra administracji i cyfryzacji do bezpłatnego i swobodnego korzystania [11]. Takim zakresem jest pasmo ISM (ang. Industrial, Scientific and Medical) służące do zastosowań przemysłowych, naukowych i medycznych w skład, którego wchodzą częstotliwości 2.4 GHz i 5 GHz. Większość z opisywanych w tej pracy technologii będzie korzystać z pasma 2.4 GHz, ponieważ zapewnia ono lepszą propagację sygnału w porównaniu do 5 GHz i zadowalającą szybkość transmisji.

### 2.4.1. Wi-Fi

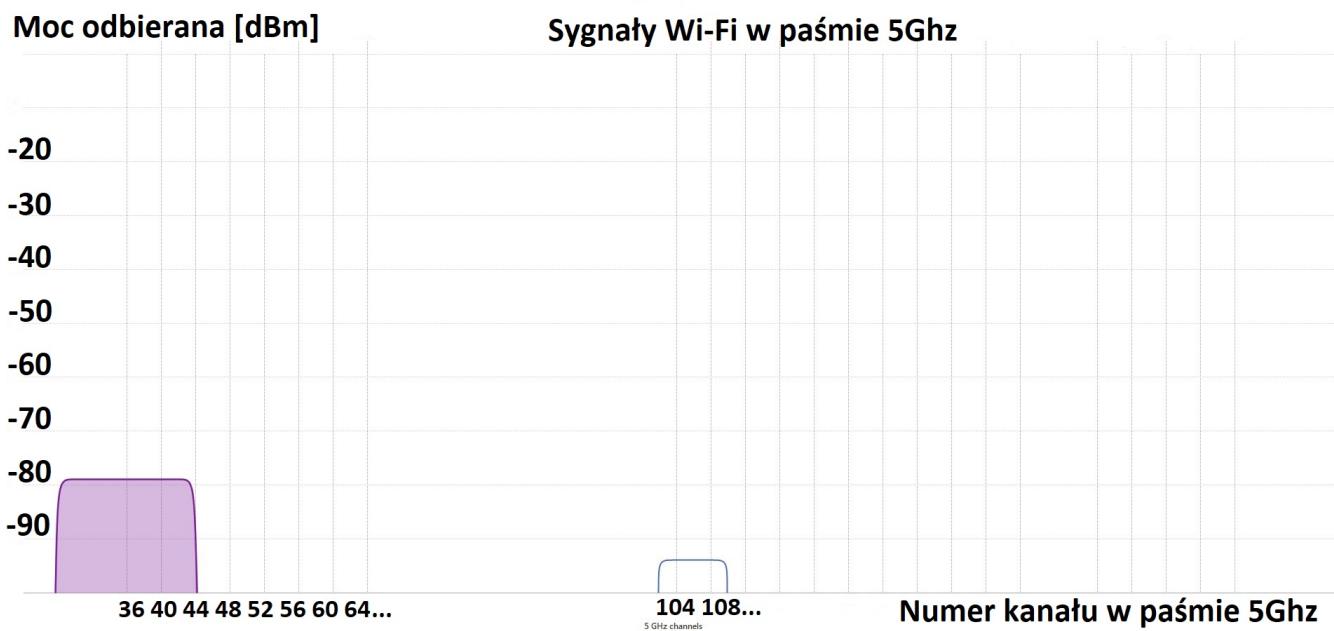
Wi-Fi jest technologią opartą na standardzie IEEE 802.11 i od początku było projektowane jako bezprzewodowy zamiennik dla standardu IEEE 802.3, czyli Ethernet. Najbardziej powiększnym jej zastosowaniem jest tworzenie lokalnych sieci LAN, gdzie router Wi-Fi pełni funkcje punktu centralnego w topologii gwiazdy.

Wi-Fi od samego początku, czyli od 1997 roku w ramach standardu IEEE 802.11 korzysta z pasma 2,4 GHz, które jest podzielone na 13 kanałów. Takie rozwiązanie ma wadę w postaci problemów z siłą oraz stabilnością sygnału, ponieważ częstotliwość 2,4 GHz jest wykorzystywana również przez kamery, smartfony, głośniki Bluetooth czy kuchenki mikrofalowe. W konsekwencji pasmo to jest przeciążone co szczególnie daje się we znaki mieszkańcom bloków mieszkalnych w postaci kiepskiego zasięgu. Rozwiązańem tego problemu ma być wprowadzone wraz z kolejnymi wersjami standardu 802.11 w 2003 roku pasmo 5 GHz. Jest to częstotliwość nie wykorzystywana przez domowe sprzęty i do tego została podzielona na większą liczbę, szerszych kanałów wynoszącą 19, co w teorii powinno zapewnić lepszą stabilność sygnału i szybkość transmisji [18].

Pomimo możliwości interferencji z innymi urządzeniami to właśnie częstotliwość 2,4 GHz zapewnia większy zasięg. Jest to spowodowane tym, że sama fala jest dłuższa niż w przypadku 5 GHz przez co jest mniej podatna na zakłócenia powodowane przez przeszkody terenowe takie jak drzwi, ściany czy meble. Efektywnie przekłada się to na od kilkunastu metrów zasięgu do kilkudziesięciu, popularne modele routerów Wi-Fi dostępnych na rynku pracujące na częstotliwości 2,4 GHz są w stanie bez problemu zapewnić dostęp do Internetu w całym, średniej wielkości domu podczas gdy w przypadku 5 GHz zasięg często kończy się kilka metrów dalej od nadajnika. Najlepiej to ilustrują poniższe rysunki 2.4 i 2.5 na których przedstawiono moc poszczególnych sygnałów Wi-Fi o różnych częstotliwościach zmierzone w jednym z wieżowców w Warszawie. Wyraźnie na nich widać, że pasmo 5 GHz, poza tym, że ma więcej kanałów, nie jest tak przeciążone jak pasmo 2,4 GHz.



Rysunek 2.4. Moce poszczególnych sygnałów Wi-Fi w paśmie 2,4 GHz



Rysunek 2.5. Moce poszczególnych sygnałów Wi-Fi w paśmie 5 GHz

Z racji, iż w obecnych czasach sieć Wi-Fi jest dosłownie wszędzie oczywiste jest, że znajduje zastosowanie również w Internecie Rzeczy. Na rynku już istnieje wiele rozwiązań w postaci gotowych modułów wraz z zaimplementowanym stosem protokołów TCP/IP i wbudowaną anteną co pozwala bardzo łatwo stworzyć urządzenie łączące się ze światem zewnętrznym poprzez Wi-Fi. Niestety poważną wadą takich systemów jest wysoki pobór prądu, który sprawia, że nie

## 2.5. Prywatność i bezpieczeństwo

---

nadają się do zasilania przy użyciu baterii. Z tego powodu w sieciach Internetu Rzeczy czujniki zasilane baterijnie komunikują się wykorzystując bardziej energooszczędne standardy.

### 2.4.2. Bluetooth

Bluetooth określany też skrótwcem BT wykorzystuje fale radiowe o częstotliwości 2,4 GHz do przesyłania danych między urządzeniami na niewielkie odległości. Od Wi-Fi różni się pod tym względem, że komunikacja odbywa się bezpośrednio pomiędzy dwoma urządzeniami. Oznacza to, że dwa urządzenia wyposażone w Bluetooth mogą wymieniać informacje między sobą, podczas gdy w przypadku Wi-Fi zawsze będzie wymagany router, który pośredniczy w tym procesie. Zasięg Bluetooth wynosi teoretycznie około 10 metrów w najpopularniejszej klasie mocy 2.5mW [21], a szybkość transmisji to 2.1 MB/s w trybie EDR (ang. Enhanced Data Rate) o zwiększonej szybkości transmisji. Jest to znacznie mniej niż w przypadku Wi-Fi, gdzie przy odpowiednim umieszczeniu routera skuteczny zasięg może być kilkukrotnie większy, a szybkość transmisji dla nowych standardów zaczyna się od 600 MB/s [20]. Ma to bezpośredni związek z zużyciem energii, gdzie to właśnie Bluetooth jest postrzegany jako standard bardziej energooszczędny. Przykładowo Wi-Fi pobiera około  $500 \mu\text{W}$  energii elektrycznej przy wysłaniu 10 wiadomości dziennie podczas gdy Bluetooth w wersji Low Energy blisko 10 razy mniej [19].

W zastosowaniach IoT Bluetooth sprawdza się wszędzie tam, gdzie wymagane jest zasilanie baterijne. Niewątpliwą zaletą jest to, że dzisiaj każdy smartfon czy komputer posiadają wbudowany interfejs BT, więc mogą pełnić one funkcję wcześniej omawianej bramki Wi-Fi. Dodatkowym atutem może być również wprowadzona w 2017 możliwość konstruowania sieci o topologii siatki nazywana Bluetooth Mesh. Pozwala ona na ominięcie ograniczeń BT w postaci niskiego zasięgu przy założeniu, że dysponujemy gęsto upakowaną siecią węzłów.

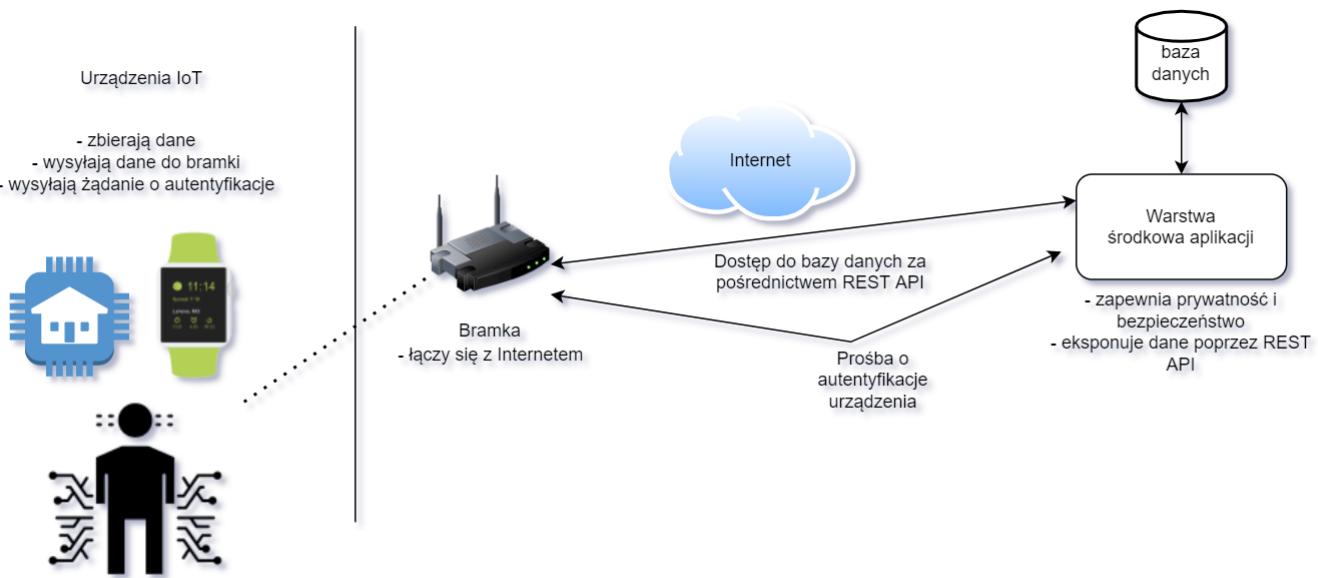
## 2.5. Prywatność i bezpieczeństwo

Prywatność i bezpieczeństwo użytkowników Internetu Rzeczy stanowi istotny problem, który dalej nie został do końca rozwiązany. Faktem jest, że rozmaite urządzenia często w niejasny sposób gromadzą informacje w celu zapewnienia spersonalizowanych doświadczeń. Budzi to uzasadniony niepokój dotyczący tego, gdzie gromadzone są nasze dane, kto ma do nich dostęp i czy są bezpieczne. Inną istotną kwestią jest często brak wystarczających zabezpieczeń w produktach dostępnych na rynku, co czyni je podatnymi na ataki hakerskie. Nawet na pozór nie przydatne dane takie jak temperatura i wilgotność powietrza w pomieszczeniu, mogą stanowić wrażliwą informację o tym czy dom jest pusty [7].

W 2019 roku przeprowadzono badanie na grupie 25 osób, aby ustalić czego oczekują w temacie bezpieczeństwa od inteligentnego domu [7]. Najczęściej pojawiającymi się odpowiedziami była przejrzystość gromadzonych danych i kontrola nad nimi. Użytkownicy w momencie, kiedy nie są informowani jakie dane są gromadzone zaczynają podejrzewać, że w każdej chwili są inwigilowani. Z tego też powodu uczestnicy domagali się opcji tymczasowego wyłączenia zbierania danych w widoczny, sprzętowy sposób jak na przykład klucz w postaci wtyku USB. Inną sugestią wysuniętą w stronę konstruktorów było stwierdzenie, że nie wszystkie urządzenia

muszą mieć dostęp do Internetu, a dane można przechowywać lokalnie zamiast w chmurze. Jednocześnie uczestnicy twierdzili, że są producenci, których darzą wystarczającym zaufaniem, aby bez obaw powierzyć im swoje dane. Powyższe badanie pokazuje, że poza samą ochroną danych poprzez stosowanie odpowiednich algorytmów i architektury równie ważna jest świadomość użytkownika i zaangażowanie go w ten proces.

Poza zapewnieniem subiektywnego odczucia bezpieczeństwa użytkownikowi oczywiście trzeba o nie zadbać w fizyczny sposób. Głównym czynnikiem, który utrudnia te zadanie są ograniczone zasoby sprzętowe urządzeń tworzących inteligentne systemy. Urządzenia z pierwszej i drugiej grupy nie są przystosowane do szyfrowania danych w konwencjonalny sposób przy pomocy klucza publicznego. Wymusza to na konstruktorach stosowanie architektury z środkową warstwą aplikacji (ang. middleware), w której jest ona odpowiedzialna za przydzielanie każdemu urządzeniu unikalnej tożsamości w celu późniejszej identyfikacji [8]. W takiej konfiguracji urządzenia tworzące dany system nie komunikują się bezpośrednio ze światem zewnętrznym, lecz robią to poprzez bramkę, czyli na przykład smartphone bądź komputer jednoplatformowy który podłączony jest do Internetu. Bramka w czasie rzeczywistym wysyła zebrane dane oraz żądania o identyfikację za pośrednictwem protokołu http do Internetu. Żądania oraz nadchodzące dane są odbierane przez środkową warstwę aplikacji, która komunikuje się z bazą danych oraz przedstawia zebrane informacje użytkownikowi w przystępnej formie.



Rysunek 2.6. Schemat blokowy opisywanej architektury

### 3. Cel projektu

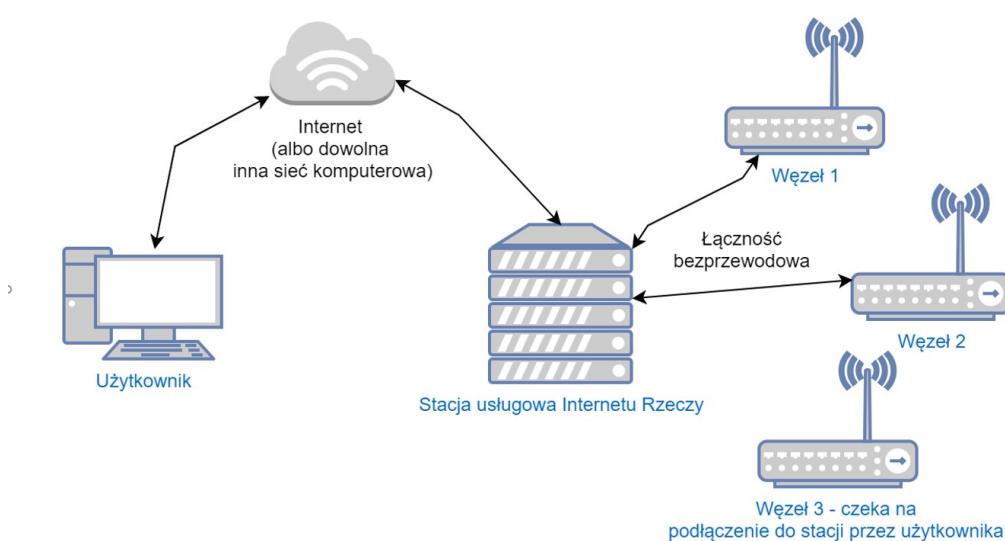
Celem projektu jest konstrukcja stacji usługowej Internetu Rzeczy z zastosowaniem komputera jednopłytkowego. Do najważniejszych funkcjonalności stacji należą:

- Dostarczenie użytkownikowi interfejsu do tworzenia sieci urządzeń.
- Gromadzenie danych pochodzących z węzłów sieci.
- Umożliwienie użytkownikowi komunikacji z węzłami z dowolnego miejsca.
- Zapewnienie prywatności i bezpieczeństwa użytkownikowi.

Aby osiągnąć powyższy cel przyjęto następujące założenia:

- Oprogramowanie uruchomione na komputerze jednopłytkowym pełni funkcje serwera, który obsługuje żądania dodania nowego urządzenia do sieci, komunikacji z węzłami i podglądu gromadzonych danych.
- Stacja posiada graficzny interfejs użytkownika będący aplikacją Internetową dostępną za pośrednictwem protokołu http.
- Węzeł to urządzenie, które może pełnić dowolną funkcję w sieci.
- Węzły komunikują się ze stacją w sposób bezprzewodowy.
- Każdy węzeł jest wykonany według określonego standardu i posiada unikatowy identyfikator.
- Dane są gromadzone lokalnie w pamięci komputera jednopłytkowego i tylko użytkownik ma do nich dostęp.

Wizualizacją powyższych celów i założeń jest następujący schemat blokowy:



Rysunek 3.1. Schemat blokowy systemu IoT z dedykowaną stacją usługową

## 4. Dobór technologii

Na podstawie założeń z poprzedniego rozdziału wyróżniono trzy główne zadania:

- konstrukcja stacji,
- konstrukcja węzła sieci,
- opracowanie protokołu komunikacyjnego.

Pracę nad każdym z powyższych punktów rozpoczęto od odpracowania stosu technologicznego w postaci sprzętu, języka programowania i pozostałych niezbędnych narzędzi, które posłużą do realizacji.

### 4.1. Stacja usługowa

Projekt stacji usługowej która ma za zadanie pełnić funkcje serwera, oraz gromadzić i przetwarzać dane, zakłada stworzenie oprogramowania i uruchomienie go na komputerze. Z tego powodu do jej konstrukcji trzeba wykorzystać zarówno odpowiednie oprogramowanie jak i sprzęt które będą optymalne z punktu widzenia potrzeb projektu.

#### 4.1.1. Sprzęt

Na wstępie warto wytlumaczyć rzecz zawartą bezpośrednio w temacie pracy, czyli wykorzystanie komputera jednoplatformowego do konstrukcji stacji. Teoretycznie w tym celu sprawdziłby się każdy komputer osobisty, lecz byłoby to marnotrawienie jego faktycznych możliwości oraz zużycie energii i koszt samego urządzenia byłoby nieproporcjonalnie duże do złożoności zadania. O wiele lepszym rozwiązaniem jest zastosowanie komputera jednoplatformowego, który charakteryzuje się wielokrotnie mniejszą ceną i poborem prądu oraz wymiarami w porównaniu do zwykłych komputerów.

W przypadku stacji należy wybrać sprzęt o odpowiedniej ilości zasobów tak aby możliwe było uruchomienie na nim wysokopoziomowego oprogramowania, które będzie pełnić rolę serwera. Samo urządzenie powinno wspierać najpopularniejsze protokoły komunikacyjne oraz mieć możliwość rozbudowy pamięci ze względu na potencjalnie dużą ilość gromadzonych danych. Poniżej przedstawiono kilka spośród najpopularniejszych modeli komputerów jednoplatformowych dostępnych na rynku:

- Raspberry Pi 4 B - komputer z centralną jednostką obliczeniową Broadcom BCM2711 wyposażoną w 4 rdzenie procesora o 64 bitowej architekturze ARM Cortex-A53 pracujący z częstotliwością taktowania 1,5 GHz. W połączeniu z nawet 8 GB wbudowanej pamięci ram jest to wystarczająca ilość zasobów, aby uruchomić na komputerze dedykowany system operacyjny z graficznym interfejsem użytkownika oparty na jądrze systemu Linux. Z urządzenia można korzystać tak z komputera osobistego przy użyciu bezpośrednio podłączonych

#### 4.1. Stacja usługowa

---

urządzeń peryferyjnych, lecz w celu oszczędzania zasobów rekommendowane jest stosowanie systemu operacyjnego bez graficznej powłoki i obsługa poprzez zdalną sesję wiersza poleceń. Raspberry Pi 4b posiada między innymi 40 pinów GPIO, porty USB, gniazdo Ethernet, wejście HDMI, Bluetooth i Wi-Fi. Moc urządzenia wynosi około 4W, ale przy większym obciążeniu procesora może się okazać, że potrzebne jest dodatkowe aktywne chłodzenie, które zwiększy zapotrzebowanie na energię.

- Raspberry Pi Zero WH – uboższa wersja pod względem ilości zasobów sprzętowych niż wyższego komputera. Charakteryzuje się częstotliwością taktowania procesora wynoszącą 1 GHz, zaledwie 512 MB wbudowanej pamięci RAM i 1 rdzeniowym procesorem. Relatywnie gorsze parametry przekładają się na niższą cenę, zużycie energii oraz wymiary, co sprawia, że Raspberry Pi Zero jest przeznaczone do mniej wymagających zastosowań.
- Intel NUC – seria wydajnych minikomputerów wyposażona w funkcje do obsługi multimediów gier oraz pracy. Urządzenia z tej gamy posiadają procesory Intel w architekturze x86 wraz z wbudowaną kartą graficzną i obsługują do 32 GB pamięci RAM. Powyższa specyfikacja niczym nie odbiega od współczesnych komputerów dlatego można powiedzieć, że jest to sprzęt przeznaczony do bezpośredniego korzystania przez użytkownika, a nie do zastosowania w Internecie Rzeczy.

SBC	Raspberry Pi model 4 b	Raspberry Pi Zero WH	Intel NUC
Moc	6,25W	0,9W	90W
Procesor	ARM, 4 rdzenie	ARM, 1 rdzeń	x86, 4 rdzenie
Taktowanie	1,5GHz	1GHz	od 2,7 do 4,5GHz
Pamięć RAM	od 2 do 8GB	512mB	do 32GB
GPIO	40 pinów	40 pinów	Brak
I2C/SPI/UART	Tak	Tak	Nie
Ethernet	Tak	Nie	Tak
Cena	≈300zł	≈60zł	≈1500zł

Tabela 4.1. Porównanie parametrów opisywanych komputerów jednopłytkowych

Spośród powyższych modeli wybrano Raspberry Pi 4 model b z dwoma giga bajtami pamięci RAM. Wszystkie jego właściwości wraz z przystępnią sprawiają, że świetnie nadaje się do wykorzystania w projekcie. Raspberry Pi Zero nie posiada istotnych zalet, które by rekompensowały gorsze parametry, z kolei Seria Intel NUC jak już wcześniej wykazano nie nadaje się do IoT.



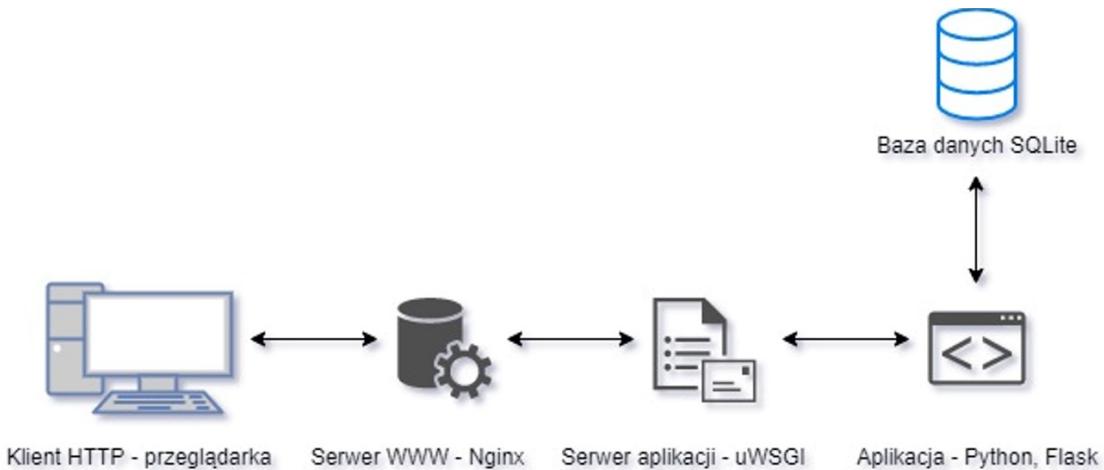
Rysunek 4.1. Komputer jednopłytkowy Raspberry Pi model 4b użyty do konstrukcji stacji usługowej

#### 4.1.2. Oprogramowanie

Po wyborze sprzętu należało wybrać oprogramowanie. Zgodnie ze schematem blokowym z rysunku 3.1 komputer jednopłytkowy ma komunikować się z komputerem użytkownika poprzez Internet. Z tego stacja usługowa między innymi musi pełnić funkcję serwera WWW oraz potrzebuje oprogramowania, które przetworzy dane i pokaże je użytkownikowi, dlatego wybrano następujące technologie:

- Python 3.6 – język programowania
- Flask – biblioteka do stworzenia aplikacji internetowej
- SQLite – baza danych
- Nginx – serwer WWW
- uWSGI – serwer aplikacji

#### 4.1. Stacja usługowa



Rysunek 4.2. Diagram stosu technologicznego stacji usługowej

Wybór Python'a 3 jest dość oczywisty w przypadku chęci tworzenia aplikacji na Raspberry Pi. Jest on preinstalowany razem z systemem operacyjnym, posiada gotowe biblioteki do interakcji z pinami GPIO i powszechnie się go stosuje w niemal każdej dziedzinie programowania począwszy od tworzenia stron internetowych skończywszy na uczeniu maszynowym.

Istotnym elementem, o który trzeba zadbać jest przechowywanie danych pochodzących z czujników. Można by gromadzić je w surowych plikach tekstowych, lecz biorąc pod uwagę przewidywaną ilość danych nie jest to optymalne podejście. Z tego powodu należy zastosować relacyjną bazę danych opartą na standardzie SQL. Do zarządzania bazą zostanie wykorzystane otwarte oprogramowanie SQLite, które jest kompatybilne z Pythonem.

Z racji, że projekt zakłada stworzenie aplikacji Internetowej warto skorzystać z gotowej biblioteki, aby nie implementować od zera logiki odpowiedzialnej za kierowanie ruchem sieciowym i obsługę podstawowych żądań. W tym celu wybrano jedną z najpopularniejszych bibliotek do tworzenia aplikacji Internetowych, czyli Flask. Jest ona relatywnie prosta i „lekka”, przykładowa aplikacja typu „hello world”, która w tym przypadku ma za zadanie obsłużyć żądanie http typu GET i zwrócić krótki tekst zawierający dosłownie kilka linii kodu. Dodatkową jej zaletą jest elastyczność, skalowalność i możliwość dostosowania części składowych projektu do własnych potrzeb.

Aby aplikacja była dostępna z poziomu przeglądarki potrzebny jest program, który obsługuje żądania protokołu http, czyli serwer WWW. Na potrzeby relatywnie prostego projektu każdy z popularnych programów dobrze sprawdzi się w tej roli dlatego ze względu na osobiste preferencje autora pracy wybrano Nginx.

Ostatnim elementem jest komunikacja serwera WWW samą aplikacją. Internetowe frameworki języka Python w tym celu korzystają z protokołu wsgi (ang. Web Server Gateway Interface). Niestety Nginx nie obsługuje tego protokołu, dlatego wymagany jest program uWSGI pełniący funkcję serwera aplikacji. Przechwytuje on żądania http z serwera WWW i przekazuje je w akceptowalnej formie do aplikacji stworzonej w języku Python.

## 4.2. Węzeł

Konstrukcja węzła sieci nie jest bezpośrednio tematem pracy, ale jeśli zależy nam na bezpieczeństwie musi podlegać on pewnym standardom. Informacje wysypane w przestrzeń przez węzeł powinny być możliwe do odczytania tylko przez stację która została z nim sparowana.

Do konstrukcji mógłby posłużyć drugi komputer jednoplatformowy, lecz wymagania sprzętowe nie są tak wygórowane jak w przypadku stacji. Z tego powodu zdecydowano się na wykorzystanie mikrokontrolera, który jest znaczco tańszy i charakteryzuje się mniejszym poborem prądu. Wybór padł na prawdopodobnie najpopularniejszą platformę programistyczną dla systemów wbudowanych, czyli Arduino UNO. Wspomniany moduł posiada mikrokontroler ATmega238P oparty o 8 bitową architekturę AVR taktowany zegarem o częstotliwości 16 MHz. Urządzenie posiada 2 KB pamięci SRAM oraz 32 KB flash. Układ wyposażono w 14 cyfrowych wyjść obsługujących najpopularniejsze interfejsy komunikacyjne oraz przetwornik analogowo-cyfrowy, a całość można programować w języku C++.

## 4.3. Protokół komunikacyjny

Ostatnim elementem całości jest protokół komunikacyjny. Na wstępnie założono, że komunikacja jest bezprzewodowa dlatego oczywiste jest, że będzie odbywać się poprzez fale radiowe. Na rynku obecnych jest przynajmniej kilkanaście technologii bezprzewodowej transmisji z czego każda posiada inne parametry takie jak zasięg, szybkość transmisji czy pobór prądu.

Z racji, iż stacja nie ma konkretnego zastosowania oznacza to, że nie jest powiedziane w jakich warunkach będzie pracować. Z tego powodu przyjęto najprostszy przypadek, czyli węzły sieci znajdują się w odległości maksymalnie kilku metrów od stacji, są zasilane z sieci a ilość przesyłanych danych jest niewielka. Oznacza to, że na wstępnie można odrzucić wszystkie technologie charakteryzujące się niskim poborem prądu bądź dużą szybkością transmisji.

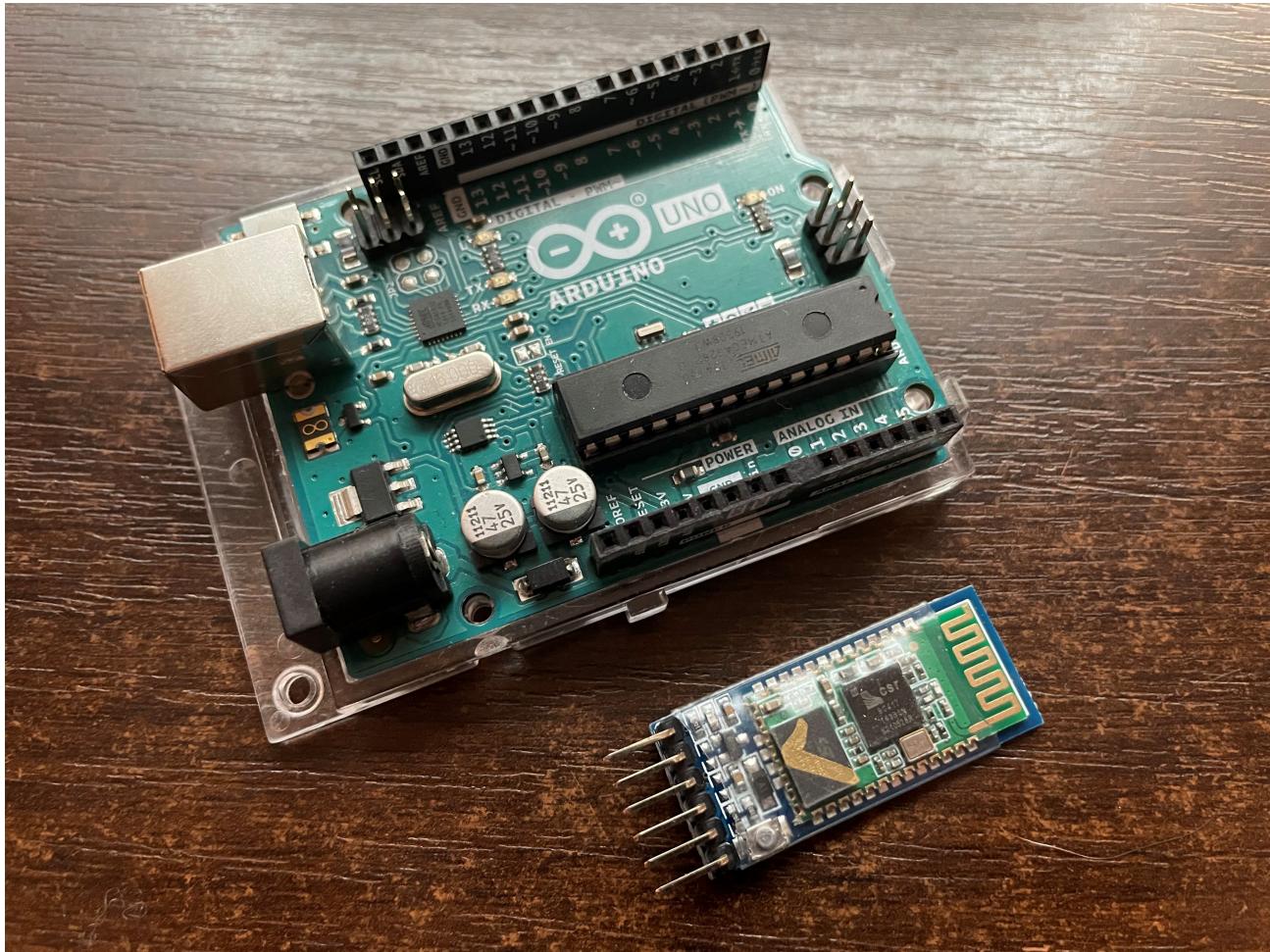
Po zawężeniu obszaru poszukiwań oczywistym wyborem w kwestii komunikacji staje się Bluetooth. Działa on w nielicencjonowanym paśmie ISM o częstotliwości 2.4 GHz [10] i szybkość transmisji wynosi 2.1 MB/s w trybie EDR (ang. Enhanced Data Rate). Standard działa w architekturze „master-slave”, gdzie jeden węzeł typu „master” może na raz komunikować się z 7 węzłami typu „slave”. W najpowszechniejszej klasie mocy 2.5mW teoretyczny zasięg wynosi 10 metrów co w połączeniu z pozostałymi parametrami idealnie odpowiada na potrzeby projektu.

Standard Bluetooth również posiada liczne profile, które służą do zapewnienia kompatybilności pomiędzy aplikacjami oraz urządzeniami. Istnieje kilkanaście profili stosowanych między innymi w bezprzewodowych zestawach słuchawkowych bądź do komunikacji z drukarką. W projekcie ograniczono się do jednego profilu określonego mianem wirtualnego portu szeregowego SPP (ang. Serial Port Profile). Po wybraniu protokołu należy się jeszcze upewnić, że zarówno węzeł jak i stacja będą z nim współpracować. O ile komputer jednoplatformowy Raspberry Pi posiada wbudowany moduł Bluetooth tak nie znajdziemy go w platformie programistycznej użytej do konstrukcji węzła. Z tego powodu potrzebny jest zewnętrzny moduł, w tym przypadku wybrano model powszechnie stosowany w amatorskich konstrukcjach, czyli HC-05. Moduł ko-

#### 4.3. Protokół komunikacyjny

---

munikuje się poprzez interfejs szeregowy UART, moc jego nadajnika wynosi 2.5mW i posiada wcześniej wymieniony profil SPP.



Rysunek 4.3. Elementy, które posłużyły do konstrukcji węzła, Arduino UNO i moduł Bluetooth HC-05

## 5. Realizacja stacji usługowej

### 5.1. Aplikacja internetowa

Realizacje rozpoczęto od konfiguracji komputera jedno płytowego poprzez zainstalowanie systemu operacyjnego Raspberry Pi OS oraz pozostałego oprogramowania i bibliotek wymienionych w stosie technologicznym stacji usługowej. Dodatkowo podłączono komputer do sieci LAN poprzez Wi-Fi, aby umożliwić wygodną pracę zdalną z komputera osobistego za pośrednictwem protokołu SSH. Kolejnym krokiem było stworzenie szkieletu samej aplikacji, co dzięki bibliotece Flask okazało się być niezwykle proste. Cała logika odpowiedzialna za obsługę żądań przychodzących z serwera aplikacji zawarta jest w jednym pliku i została przedstawiona we fragmencie kodu 5.1.

---

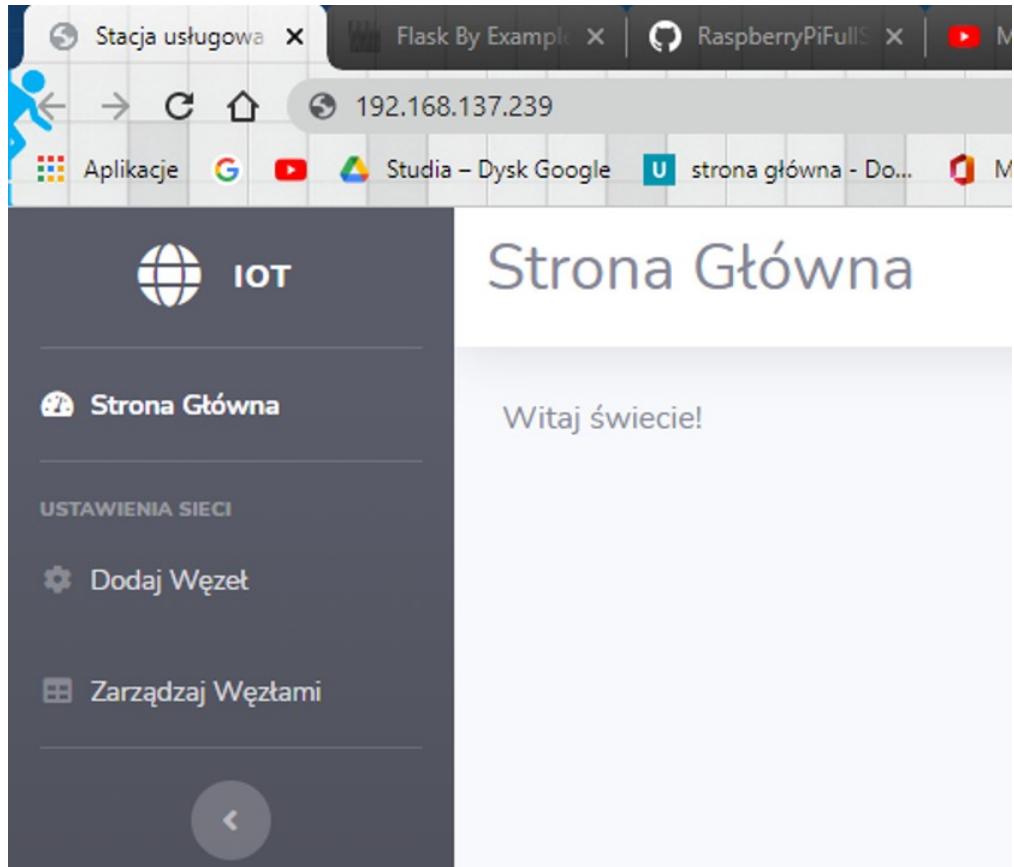
**Kod 5.1** Fragment odpowiedzialny za obsługę żądań przychodzących z serwera

```
from flask import Flask
from flask.templating import render_template
app = Flask(__name__)
@app.route("/")
def dashboard():
    return render_template("index.jinja", message="Witaj świecie!")
```

---

Powyższy kod sprawia, że aplikacja po otrzymaniu żądania typu GET od serwera aplikacji zwróci kod strony zawarty w pliku „index.jinja” wraz z dynamicznie dodaną wiadomością „Witaj świecie!”. Sama treści strony została stworzona na podstawie darmowego szablonu HTML dostępnego w Internecie. Aby przetestować, czy całość działa wystarczy wpisać w przeglądarkę adres IP stacji który został przydzielony przez router. Poniżej przedstawiono rezultat:

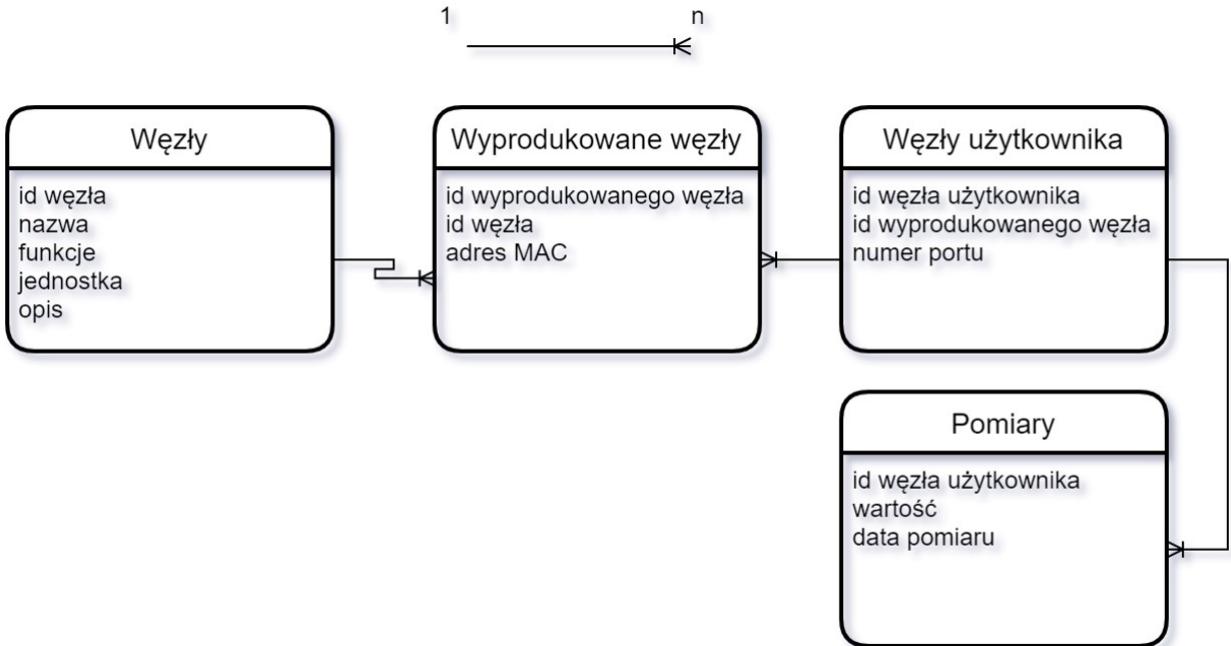
## 5.2. Baza danych



Rysunek 5.1. Zrzut ekranu działającego szkieletu aplikacji

## 5.2. Baza danych

Kolejnym krokiem było zaimplementowanie funkcjonalności wymienionych w założeniach. Z racji, iż aplikacja przede wszystkim ma zbierać i przetwarzać informacje prace rozpoczęto od projektu bazy danych.



Rysunek 5.2. Szkic bazy danych

Na rysunku 5.2. przedstawiono logiczny model bazy danych który posłużył później do implementacji. Wszystkie relacje należy rozumieć jako „jeden do wielu” gdzie linia zakończona rozwidleniem oznacza, że w danej tabeli występuje wielokrotnie rekord z innej tabeli będącej z nią w relacji.

W zaproponowanym modelu tabela „Węzły” zawiera w sobie listę wszystkich rodzajów węzłów. Może być to dowolny czujnik temperatury, albo zdalnie sterowany przekaźnik. Przykładowo we fragmencie kodu 5.2. wstawiono do tej tabeli węzeł o nazwie „Węzeł testowy” zadaną listą funkcji.

Tabela „Wyprodukowane węzły” reprezentuje fizyczne, wyprodukowane węzły sieci. Oznacza to, że każdy taki węzeł posiada unikalną kombinację swojego własnego identyfikatora, identyfikatora rodzaju oraz adresu MAC. Dzięki temu użytkownik podając numer konkretnego węzła będzie mógł dodać go do sieci, z kolei aplikacja będzie wiedziała jak z nim „rozmawiać”. W tym momencie możliwe jest wyprodukowanie dowolnej ilości urządzeń, które stacja będzie postrzegać jako „Węzeł testowy” lecz każde z nich będzie miało swoją unikalną tożsamość. Proponowana architektura również ma na celu skalowalność, dzięki czemu dodanie nowego rodzaju węzła o nowych, dotychczas nieznanych funkcjach nie będzie stanowić problemu i wymaga jedynie utworzenia odpowiedniego rekordu w tabeli „Węzły”. Ponad to zapewnia ona prywatność, ponieważ każdy wyprodukowany węzeł musi zostać wcześniej otrzymać unikalny identyfikator od producenta, więc nie będzie możliwe korzystanie z urządzeń wyprodukowanych przez osoby trzecie, które mogłyby nie spełniać standardów bezpieczeństwa. Co więcej cały plik bazy danych jest przechowywany w pamięci komputera jednopłytowego więc tylko użytkownik ma dostęp do gromadzonych danych. Kod 5.2. przedstawia fragment skryptu odpowiedzialnego za

### 5.3. Łączność bezprzewodowa

---

stworzenie całej bazy danych. W pierwszej kolejności nawiązywane jest połączenie z pustą bazą danych, po czym wykonywane są polecenia napisane zgodnie ze standardem SQL. Polecenie na poniższym listingu tworzy tabele o nazwie „nodes” odpowiadającej tabeli „węzły” z rysunku 5.2. i umieszcza w niej jeden testowy rekord o nazwie „Węzeł testowy”.

---

#### Kod 5.2 Fragment skryptu odpowiedzialnego za tworzenie bazy danych

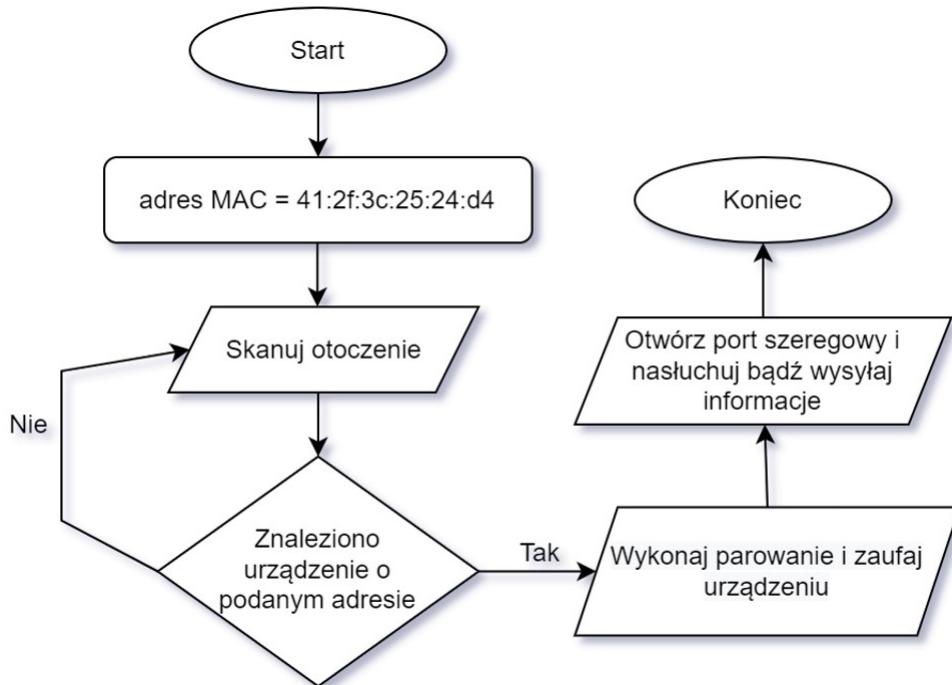
---

```
# Stwórz obiekt con aby
# ustanowić połączenie z bazą "data_base.db"
con = sqlite3.connect("data_base.db")
# stwórz obiekt kursora
cur = con.cursor()
# wykonaj podany skrypt
cur.executescript("""
    create table nodes(
        node_id INTEGER PRIMARY KEY,
        name TEXT NOT NULL,
        direction TEXT NOT NULL,
        functions TEXT NOT NULL,
        unit TEXT,
        description TEXT
    );
    insert into nodes (name, direction,
functions, unit, description)
        values ( 'Węzeł testowy', 'Wyjścia',
'pomiar,przełącznik', 'jednostka', 'wartość');
""")
""")
```

---

### 5.3. Łączność bezprzewodowa

Ostatnim elementem oprogramowania stacji usługowej było zapewnienie łączności bezprzewodowej z węzłami przy użyciu Bluetooth. Schemat blokowy algorytmu służącego do nawiązania takiej łączności przedstawiono poniżej:



Rysunek 5.3. Schemat blokowy algorytmu do nawiązywania łączności bezprzewodowej po- przez Bluetooth

Na wstępnie musimy znać adres MAC modułu Bluetooth z którym chcemy się porozumieć. Kolejnym krokiem jest skanujemy otoczenie aż do skutku po czym należy wykonać procedurę parowania i zaufać urządzeniu. Po wykonaniu tych czynności pozostaje otworzyć port szeregowy i można wysyłać bądź odbierać informacje od modułu Bluetooth. Wszystkie powyższe operacje są implementowane przez system operacyjny i dostępne za pośrednictwem komend terminala systemu Linux więc autor pracy stworzył skrypt w języku Python który cały ten proces automatyzuje. Przykładowy fragment kodu 5.3. przedstawia funkcję odpowiedzialną za wysłanie do wiersza poleceń komputera polecenia „scan on”, które uruchamia procedurę skanowania w poszukiwaniu wykrywalnych urządzeń Bluetooth.

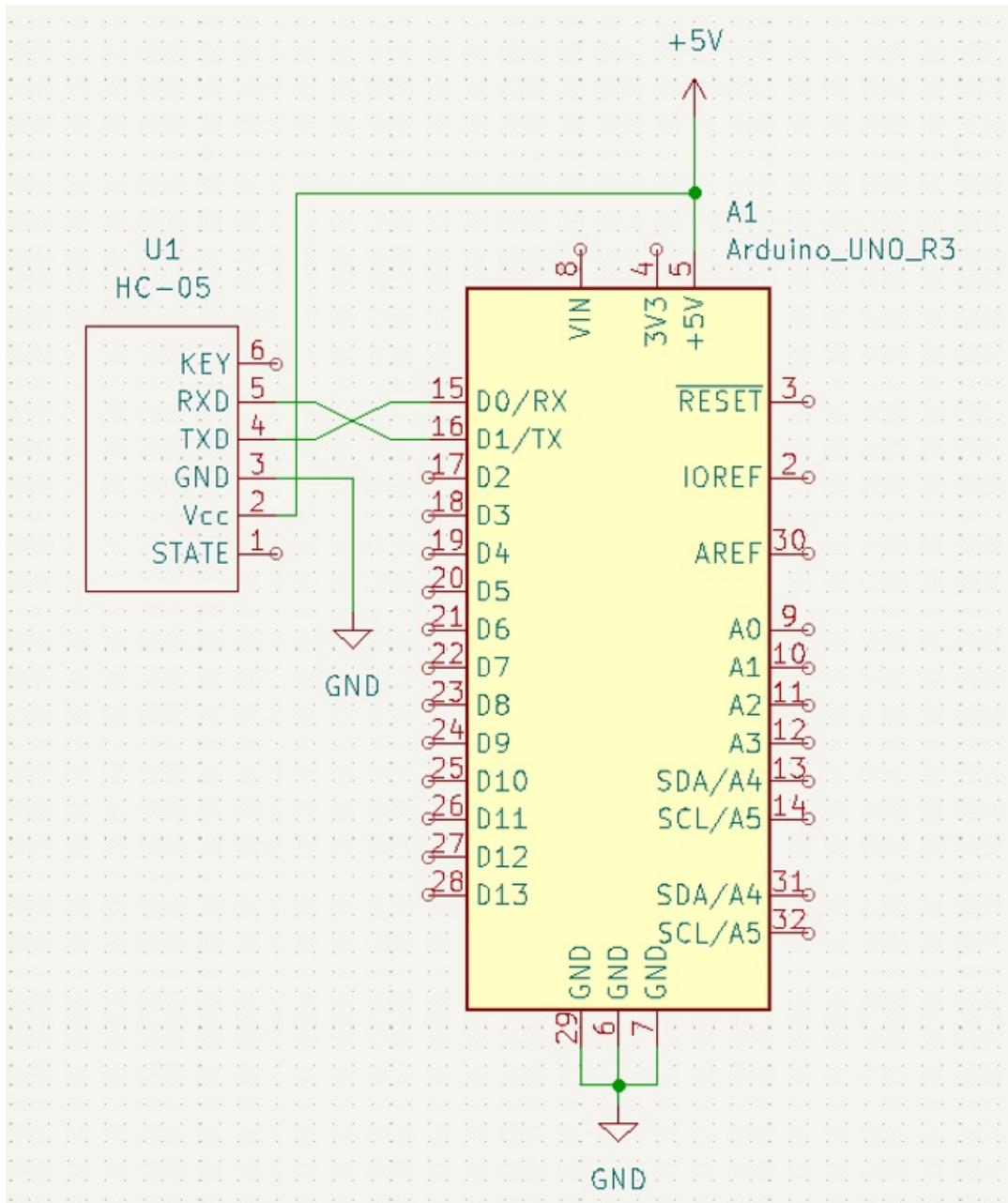
**Kod 5.3** Fragment skryptu automatyzującego nawiązywanie łączności bezprzewodowej

```

def start_scan(self):
    """Start bluetooth scanning process."""
    try:
        self.send("scan on")
    except Exception as e:
        logger.error(e)
  
```

## 5.4. Węzeł

Ostatnim elementem projektu był węzeł sieci z którym stacja będzie się komunikować. Do konstrukcji węzła posłużyła platforma programistyczna dla systemów wbudowanych Arduino Uno zintegrowana z modułem Bluetooth HC-05. Schemat połączeń został przedstawiony na rysunku 5.5.



Rysunek 5.4. Schemat połączeń elektrycznych węzła sieci

Moduł HC-05 komunikuje się poprzez interfejs szeregowy UART, więc wystarczy podłączyć jedynie wyprowadzenia służące do odbierania i wysyłania danych oraz masę i zasilanie. Mikrokontroler został zaprogramowany, aby wysyłać losowo generowaną liczbę co ma na celu symuluwać dokonywanie pomiaru za pośrednictwem zewnętrznego czujnika. Dodatkowo mikrokon-

troler w przypadku otrzymania odpowiedniej komendy zmieni stan jednego z wyjść cyfrowych które podłączone jest do wbudowanej w płytę diody LED. Fragment kodu 5.4. przedstawia główną pętlę programu mikrokontrolera. Docelowo zamiast diody do takiego wyjścia można podłączyć na przykład przekaźnik, który steruje urządzeniem zasilanym wysokim napięciem.

---

**Kod 5.4** Główna pętla programu mikrokontrolera

---

```
void loop() {
    delay(100);
    if (Serial.available() > 0) {
        //Przeczytaj nadchodzący bajt
        incomingByte = Serial.read();
        if(incomingByte == 48) {
            // Wyłącz LED
            digitalWrite(LED_BUILTIN, LOW);
        }
        if(incomingByte == 49) {
            // Włącz LED
            digitalWrite(LED_BUILTIN, HIGH);
        }
        if(incomingByte == 50) {
            // Wyślij losowo wygenerowaną liczbę
            randomNumber = random(1,100);
            Serial.print(randomNumber);
            Serial.print("\n");
        }
    }
    // Wyczyść bufor
    Serial.flush();
}
```

---

## 6. Testowanie

### 6.1. Testy niezawodności łączności bezprzewodowej

W ramach testów sprawdzono wpływ zakłócających sygnałów oraz przeszkód terenowych obecnych w mieszkaniu na niezawodność łączności bezprzewodowej przy użyciu standardu Bluetooth. W tym celu nadano 10000 ramek danych z w odstępach czasowych wynoszącym 1 ms i zbadano następujące parametry:

- Czas transmisji – różnica czasu pomiędzy ostatnią a pierwszą odebraną ramką

$$t = t_{koniec} - t_{start} \quad (6.1.)$$

- Sprawność transmisji – stosunek liczby poprawnie odebranych do ilości wszystkich wysłanych ramek

$$S = \frac{N_{odebranych}}{N_{wyslanych}} \quad (6.2.)$$

- Średnia arytmetyczna mocy odbieranego sygnału wyrażona w dBm wyliczona z pięciu pomiarów

#### 6.1.1. Układ pomiarowy

Do pomiaru mocy odbieranego sygnału standardu Bluetooth użyto aplikacji „bluetooth scanner” dostępnej na telefonach z systemem operacyjnym android. Urządzeniem wykorzystanym do pomiarów był smartfon Samsung Galaxy A3 2016 wykorzystujący standard Bluetooth 4.0 w drugiej klasie mocy wynoszącej 2.5mW.

W celu wyznaczenia niepewności pomiaru wykorzystano metodę statystyczną, polegającą na dokonaniu obliczeń na podstawie wielokrotnych pomiarów. Obliczona w ten sposób niepewność odpowiada odchyleniu standardowemu średniej z dokonanych pomiarów i jest określona wzorem 6.3.

$$u(x) = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x}_{srednia})^2}{n(n-1)}} \quad (6.3.)$$

Gdzie:

$x_i$  - wynik pomiaru nr i

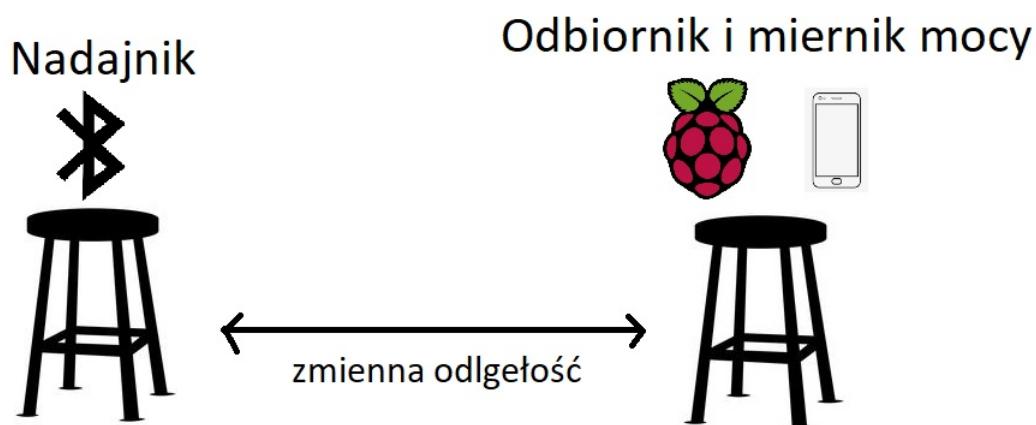
$\bar{x}_{srednia}$  - średnia arytmetyczna ze wszystkich pomiarów

n - liczba pomiarów

Za nadajnik sygnału posłużył węzeł sieci wykonany w ramach niniejszego projektu. Jest on wyposażony w moduł Bluetooth HC-05 działający w standardzie Bluetooth 2.0 i w drugiej klasie mocy 2,5mW [22].

Odbiornikiem była stacja również uprzednio skonstruowana stacja usługowa Internetu Rzeczy z zastosowaniem komputera jednopłytkowego Raspberry Pi model 4 b. Podobnie jak nadajnik pracuje ona w klasie mocy 2,5mW lecz posiada standard Bluetooth w wersji 5.0 [23]

Wszystkie pomiary zostały przeprowadzone gdy smartfon był trzymany pionowo, tyłem do nadajnika, na wysokości około 50cm. Podczas mierzenia sprawności transmisji nadajnik pozostawał nieruchomy, a odbiornik zmieniał swoje położenie. Oba urządzenia znajdowały się na tej samej wysokości 50 cm nad ziemią. Cały układ pomiarowy został przedstawiony na rysunku 6.1.

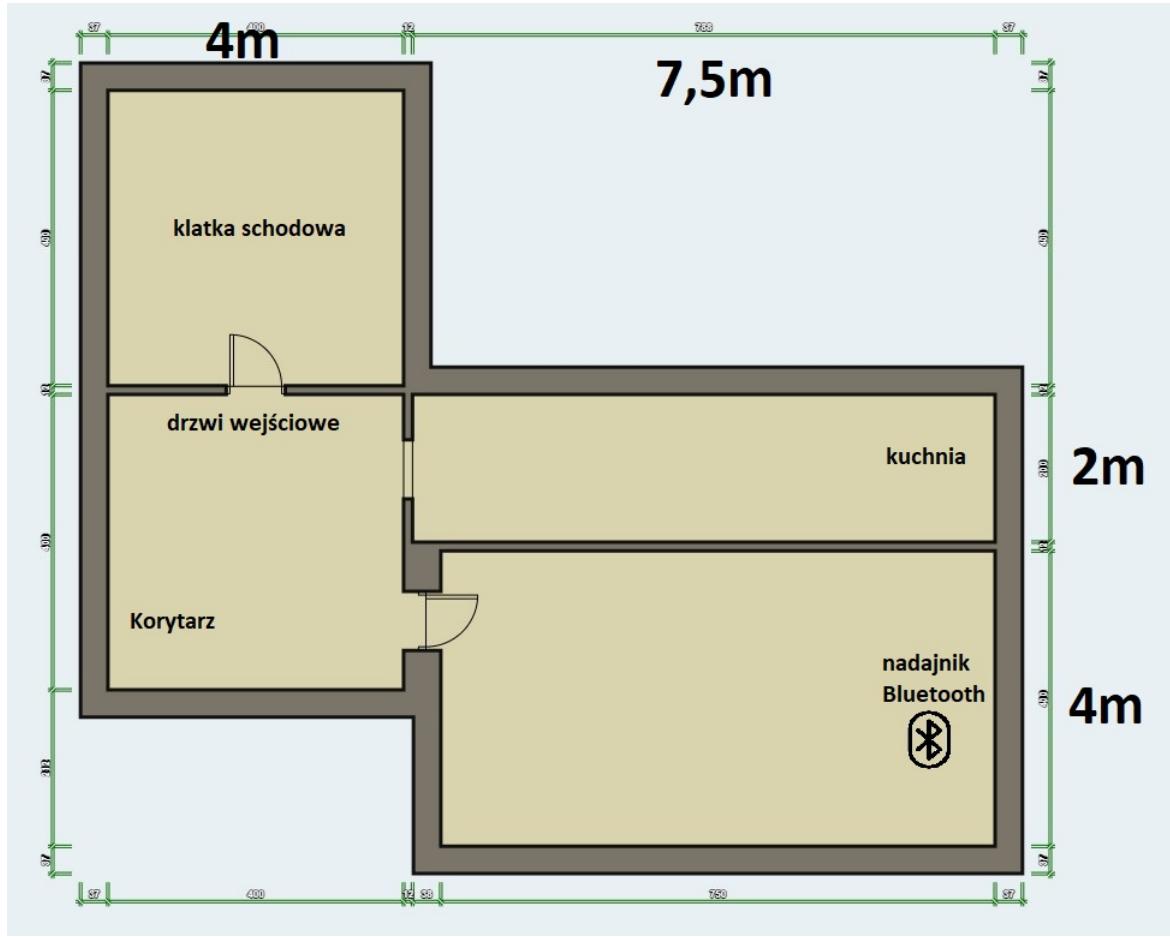


Rysunek 6.1. Schemat układu pomiarowego

### 6.1.2. Środowisko testowe

Testy przeprowadzono w dwóch różnych lokalizacjach. Pierwszą z nich jest mieszkanie w jednym z bloków mieszkalnych w Warszawie, schemat środowiska testowego przedstawiono na rysunku 6.2.

## 6.1. Testy niezawodności łączności bezprzewodowej

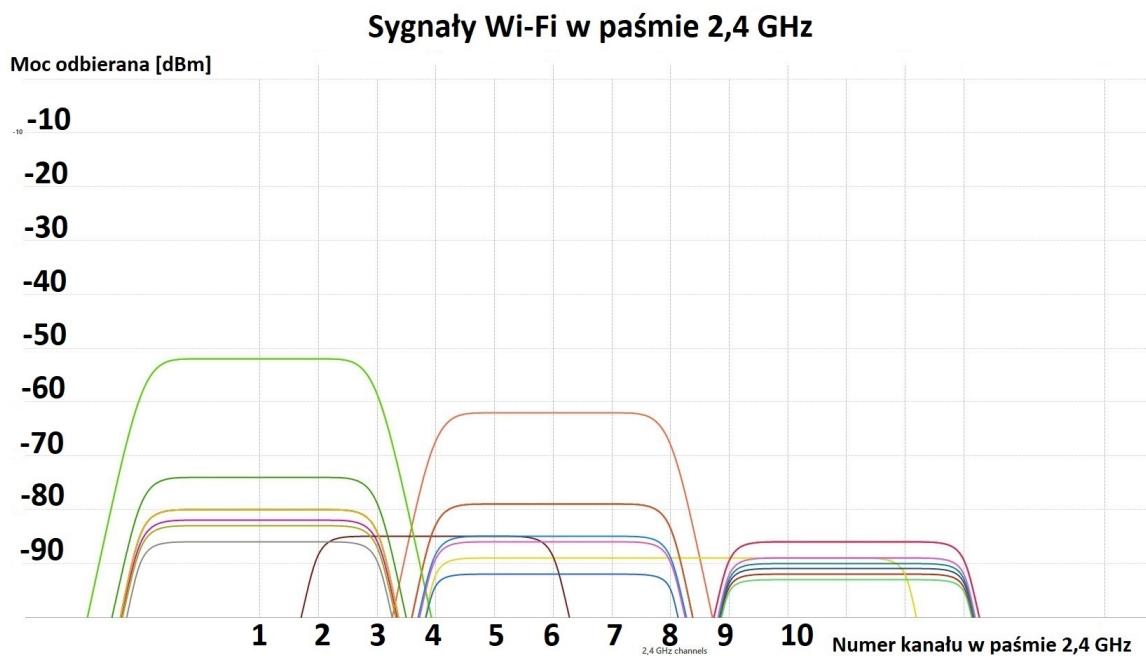


Rysunek 6.2. Schemat środowiska testowego nr 1.

W otoczeniu obecnych jest wiele sieci Wi-Fi oraz Bluetooth, które potencjalnie utrudniają transmisje. W tabeli 6.1. i na rysunku 6.3. przedstawiono wyniki pomiarów mocy poszczególnych sygnałów w paśmie 2,4GHz przeprowadzonych w najbliższym otoczeniu nadajnika, zdecydowana większość z nich pochodzi od sąsiadów.

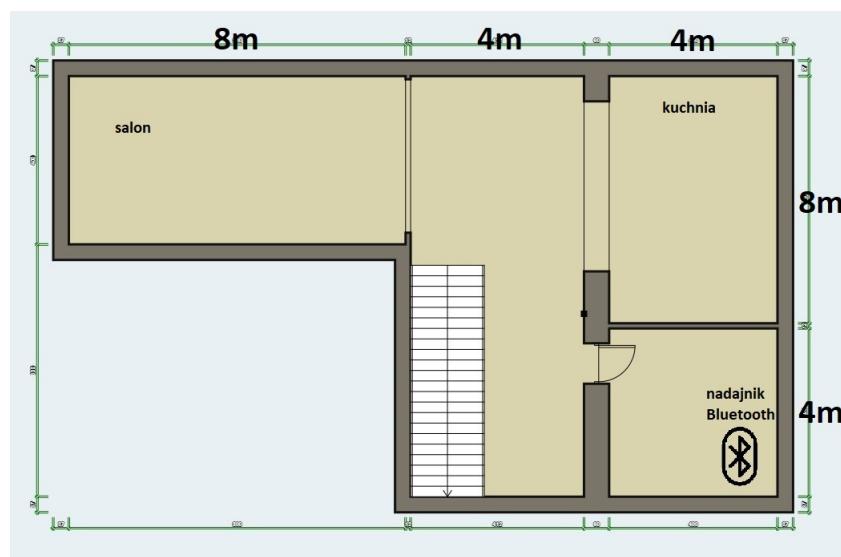
6F-F5-15-9A-C6-8D	Samsung Electronics Ltd	-83 dBm		F
9C-8C-6E-59-5D-C5		-87 dBm		F
2B-7D-A6-FF-6B-2E	Nintendo Ltd	-73 dBm		F
ACTON II	54-B7-E5-51-A4-0A	-83 dBm		F
	A4-C1-38-85-AF-95	-59 dBm		F
	70-28-45-5E-2A-17	-83 dBm		F
	58-2D-34-54-73-BF	-63 dBm		F
	14-BB-6E-86-09-A0	-86 dBm		F
	E9-40-2C-01-B8-C9	-68 dBm		F
	66-B1-6A-3E-1C-E7	-90 dBm		F
	73-47-03-DD-E0-4A	-67 dBm		F
	E5-32-97-60-04-22	-50 dBm		F

Tabela 6.1. Moce poszczególnych sygnałów Bluetooth w otoczeniu nadajnika w środowisku testowym nr 1.



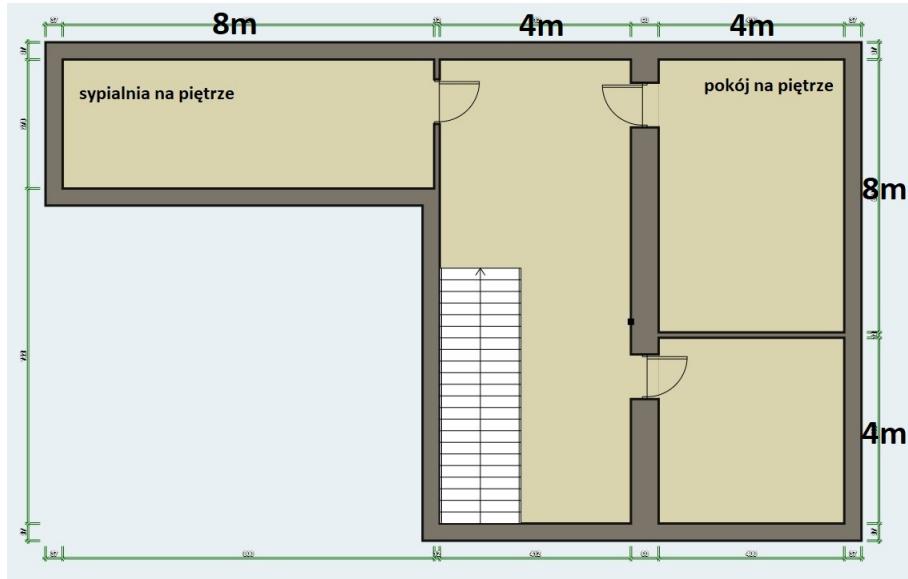
Rysunek 6.3. Moce poszczególnych sygnałów Wi-Fi w otoczeniu nadajnika w środowisku testowym nr 1.

Drugą lokalizacją jest jednorodzinny dom na prowincji. Na rysunku 6.4. został przedstawiony schemat środowiska testowego numer 2, budynek posiada dwie kondygnacje na których przeprowadzono pomiary – parter i pierwsze piętro. Czynnikami które uległy zmianie w porównaniu do środowiska numer 1 jest znaczco mniejsza ilość sygnałów Wi-Fi oraz Bluetooth. Na rysunku 6.6. widoczne są tylko dwie sieci z czego jedna z nich to prywatna sieć domowników, z kolei w tabeli 6.2. znajdują się tylko dwa sygnały pochodzące z bezprzewodowych urządzeń periferyjnych.

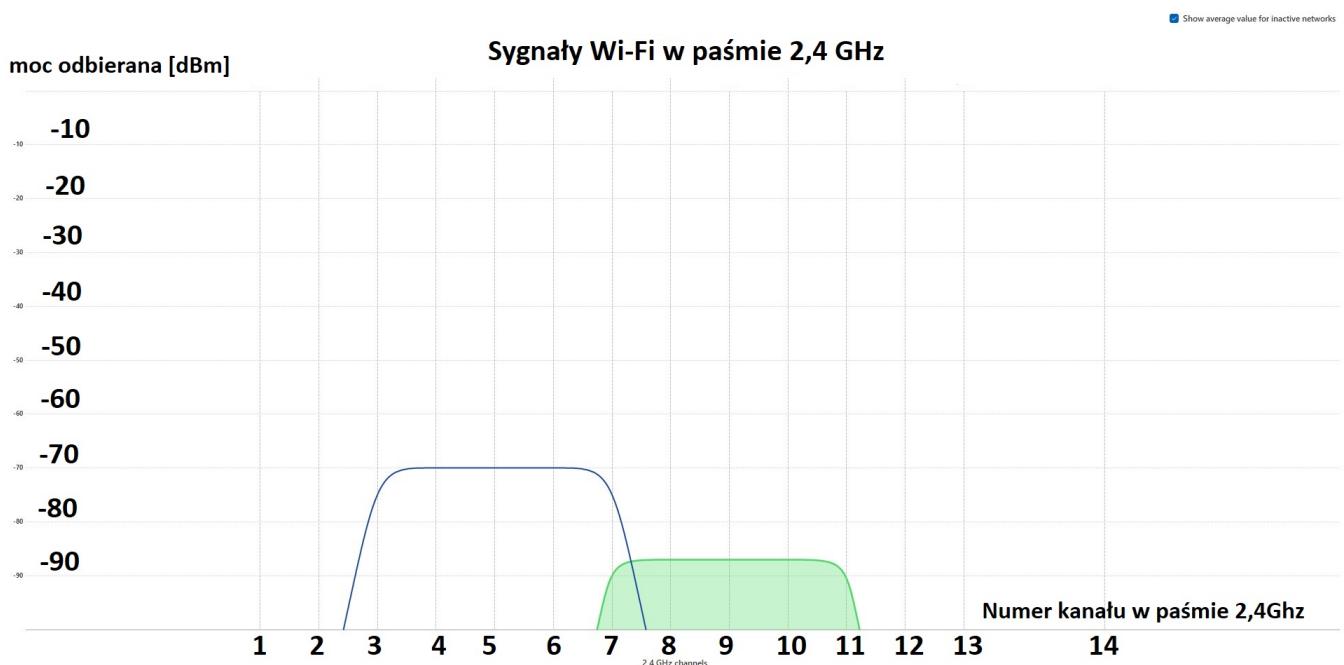


Rysunek 6.4. Schemat środowiska testowego nr 2 - parter.

## 6.1. Testy niezawodności łączności bezprzewodowej



Rysunek 6.5. Schemat środowiska testowego nr 2 - pierwsze piętro.



Rysunek 6.6. Moce poszczególnych sygnałów Wi-Fi w otoczeniu nadajnika w środowisku testowym nr 2.

Device	MAC Address	Manufacturer	Signal Strength	Signal Strength History
LE_WH-1000XM3	D5-DB-7F-2C-CE-5C 94-DB-56-54-BF-28		-35 dBm -46 dBm	

Tabela 6.2. Moce poszczególnych sygnałów Bluetooth w otoczeniu nadajnika w środowisku testowym nr 2.

### 6.1.3. Pomiary

W pierwszej kolejności w ramach próby kontrolnej przeprowadzono pomiary dla łączności przewodowej, a następnie powtórzono ten sam proces dla każdego wyszczególnionego na schemacie punktów. Wyniki dla środowiska testowego numer 1 przedstawiono w tabelach 6.3. i 6.4.

Miejsce	moc [dBm]	niepewność [dBm]	odległość od nadajnika [m]
obok nadajnika (przewód)	n. d.	n. d.	0
obok nadajnika	-31,6	1,326649916	0
korytarz	-71,6	1,720465053	10
kuchnia	-62,6	1,630950643	3
drzwi wejściowe	-83,2	1,881488772	11,5
klatka schodowa, otwarte drzwi	-86,8	1,356465997	12
klatka schodowa, zamknięte drzwi	?	?	12

Tabela 6.3. Wyniki pomiarów mocy przeprowadzonych w środowisku testowym nr. 1

Miejsce	czas transmisji [s]	sprawność transmisji [%]
obok nadajnika (przewód)	50,805	100
obok nadajnika	50,781	100
korytarz	50,732	100
kuchnia	50,512	100
drzwi wejściowe	53,355	100
klatka schodowa, otwarte drzwi	58,839	90,8
klatka schodowa, zamknięte drzwi	-	38,5

Tabela 6.4. Wyniki pomiarów sprawności transmisji przeprowadzonych w środowisku testowym nr. 1

Pierwszą rzeczą, która można wywnioskować z powyższych pomiarów jest minimalna moc sygnału która zapewnia bezbłędną transmisję. Wynosi ona około -83 dBm i jest to wynik zgodny ze specyfikacją standardu Bluetooth według której minimalna moc mieści się w przedziale od -70 do -82 dBm. Warto również zwrócić uwagę na to, że dla dostatecznie dobrego zasięgu czas transmisji pozostaje praktycznie taki sam jak w przypadku transmisji przewodowej.

Kolejną kwestią jest wpływ przeszkód i odległości na propagację sygnału. Wyraźnie widać, że odległość 10 metrów w przypadku korytarza oraz ściana w przypadku kuchni nie wpływały na odbiór sygnału, zarówno czas jak i sprawność transmisji pozostają bez zmian. Dopiero przy drzwiach wejściowych, w odległości około 12 metrów od nadajnika obserwowne stają się utrudnienia w postaci wydłużonego czasu nadawania. Poza obszarem mieszkania, na klatce schodowej można zaobserwować obniżenie sprawności transmisji do około 90%. Zamknięcie drzwi wejściowych powoduje prawie całkowity zanik sygnału do tego stopnia, że transmisja zostaje w pewnym momencie bezpowrotnie przerwana, a moc odbieranego sygnału nie daje się zmierzyć przy użyciu wybranej metody, można jedynie szacować, że wynosi ona poniżej -90 dBm.

## 6.1. Testy niezawodności łączności bezprzewodowej

---

Całą powyższą procedurę powtórzono dla środowiska testowego numer 2. Wyniki przedstawiono w tabelach 6.5. i 6.6.

Miejsce	moc [dBm]	niepewność [dBm]	odległość od nadajnika [m]
Obok nadajnika (przewód)	n. d.	n. d.	0
Obok nadajnika	-30,8	0,8	0
Salon	-82	1,414213562	15
kuchnia	-75,8	1,280624847	10
Pokój na piętrze	-75,6	1,029563014	12
Sypialnia na piętrze	-84,6	1,326649916	18

Tabela 6.5. Wyniki pomiarów mocy przeprowadzonych w środowisku testowym nr. 2

Miejsce	czas transmisji [s]	sprawność transmisji [%]
Obok nadajnika (przewód)	50.805	100
Obok nadajnika	50.755	100
Salon	50.707	99,54
kuchnia	50.681	100
Pokój na piętrze	50.787	100
Sypialnia na piętrze	52.234	99,09

Tabela 6.6. Wyniki pomiarów sprawności transmisji przeprowadzonych w środowisku testowym nr. 2

W tym przypadku w porównaniu do pierwszej serii pomiarów nie udało się znaleźć w domu punktu, dla którego transmisja została bezpowrotnie utracona. Nawet w skrajnym punkcie takim jak sypialnia na piętrze sprawność transmisji uległa zmniejszeniu o tylko mniej niż jeden procent a czas transmisji uległ wydłużeniu o mniej niż 2 sekundy.

Podobnie jak w przypadku środowiska testowego numer 1 można wywnioskować, że minimalna moc sygnału która zapewnia bezbłędną transmisję wynosi więcej niż -80 dBm, lecz tym razem skuteczny zasięg uległ zwiększeniu o około 20%.

### 6.1.4. Wnioski

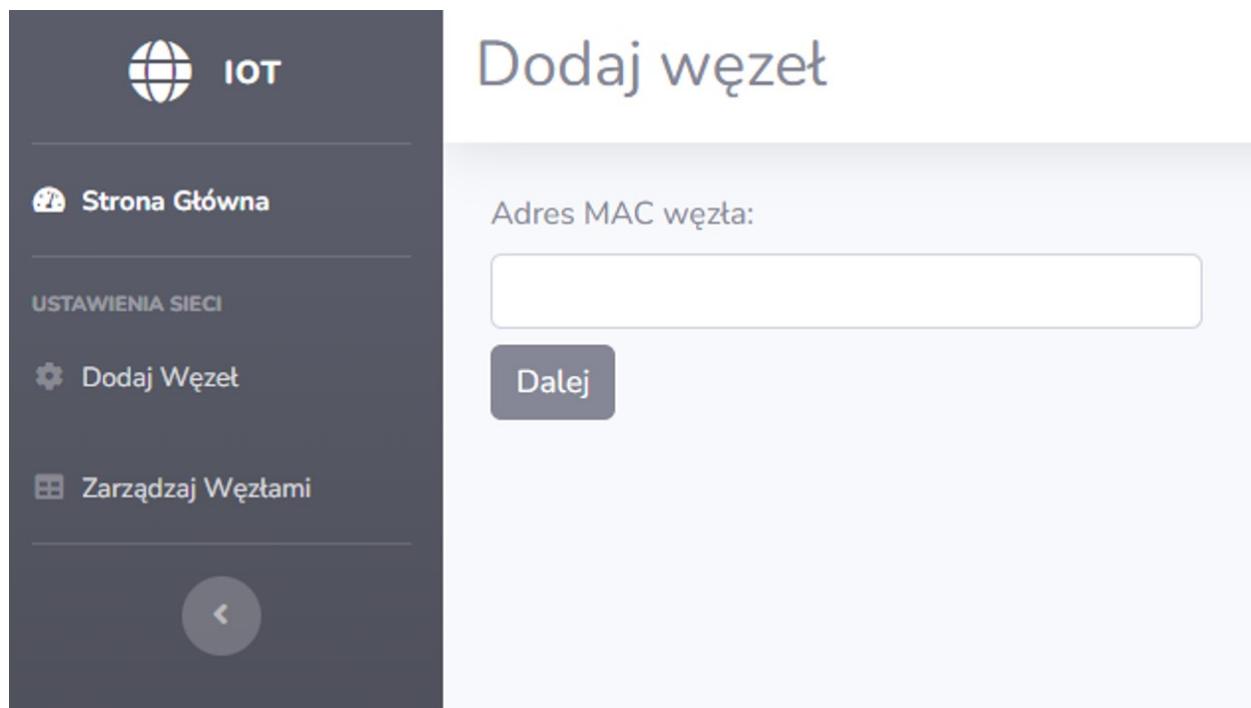
Na podstawie wyników powyższych pomiarów można stwierdzić, że fizyczne przeszkody w postaci pojedynczych ścian bądź sufitu nie stanowią bariery dla sygnału standardu Bluetooth. Wskazują na to między innymi pomiary wykonane w kuchni dla środowiska testowego numer 1 gdzie nie zaobserwowano w żadnych zmian w stosunku do próby kontrolnej.

Z drugiej strony czynnikiem znacząco wpływającym na zasięg co za tym idzie jakość sygnału są zakłócenia w postaci niepożądanych sieci Wi-Fi oraz innych urządzeń korzystających z Bluetooth. Efekt ten jest widoczny w mieszkaniu, gdzie skuteczny zasięg wynosi około 10 metrów podczas gdy w domu jednorodzinnym prawie bezbłędną transmisję uzyskano na dystansie 15 metrów.

Podsumowując można przyjąć, że minimalna odległość, na której transmisja odbywa się w sposób niezawodny wynosi do 10 metrów z czego w dobrych warunkach, czyli przy braku zakłóceń dystans ten może ulec zwiększeniu o około 2 metry. Ponad to przeszkody w postaci pojedynczej ściany bądź sufitu nie mają znaczącego wpływu na propagację sygnału.

## 6.2. Testy wersji demonstracyjnej stacji usługowej

Graficzny interfejs użytkownika gotowej stacji wygląda następująco:

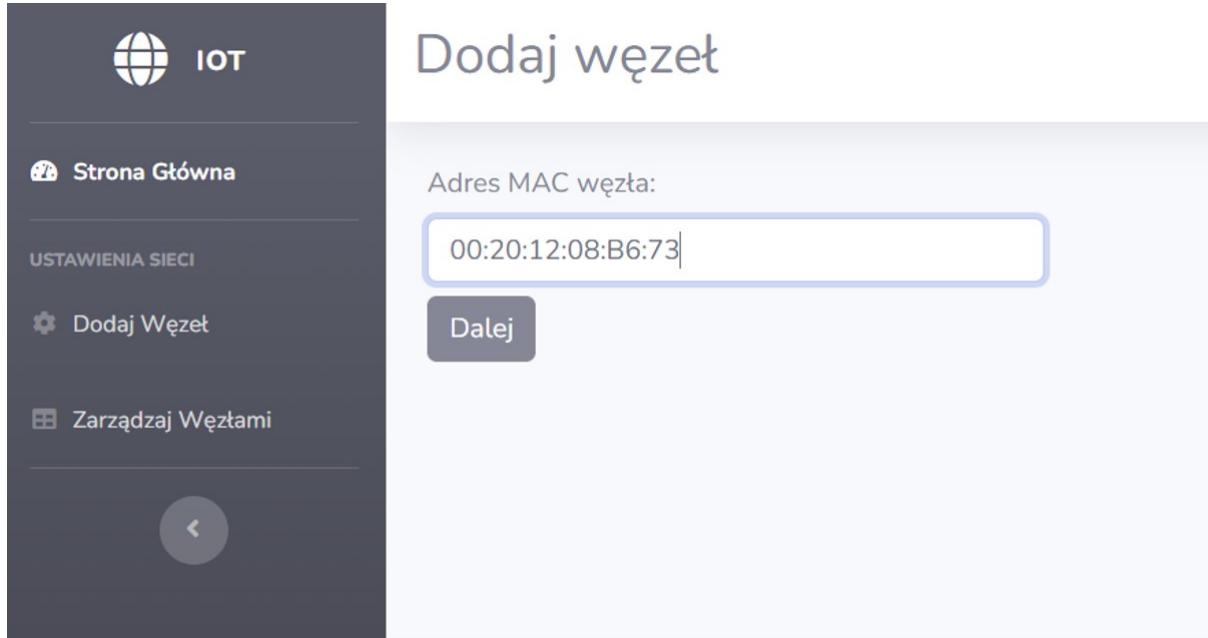


Rysunek 6.7. Widok dodawania nowego węzła.

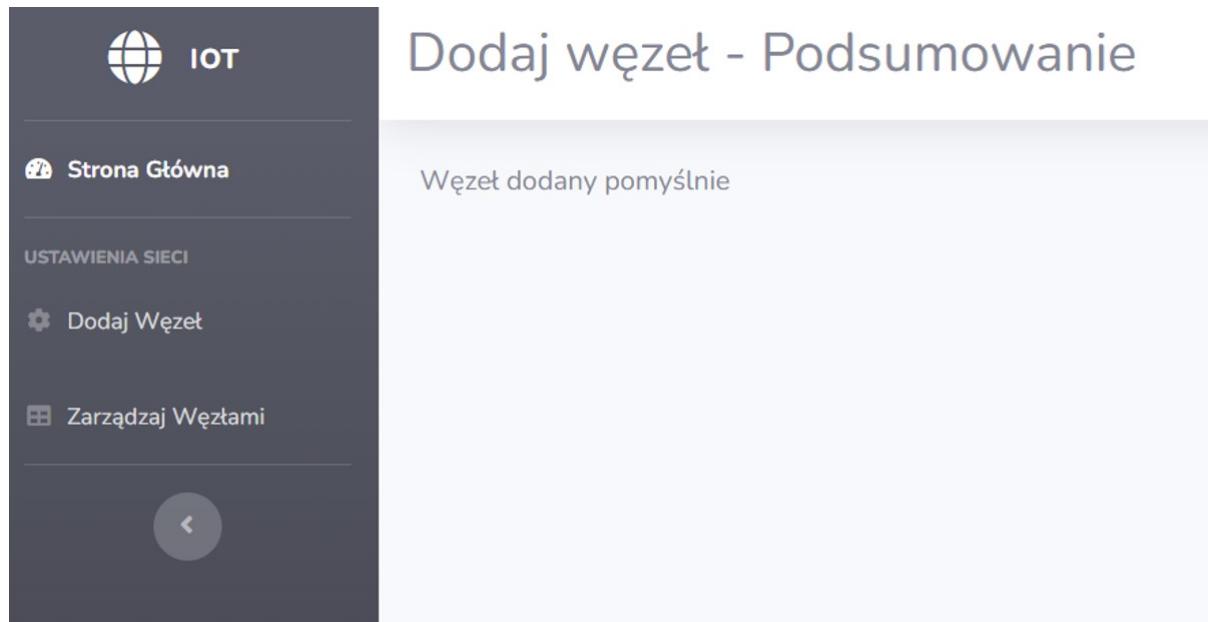
Wprowadzenie adresu MAC modułu Bluetooth użytego do budowy węzła i kliknięcie „Dalej” sprawia, że aplikacja sprawdzi czy podany adres znajduje się w bazie danych a następnie zostanie wykonana procedura nawiązania łączności bezprzewodowej.

Po kilku chwilach, jeśli wszystko przebiegło pomyślnie aplikacja zwraca stosowny komunikat.

## 6.2. Testy wersji demonstracyjnej stacji usługowej

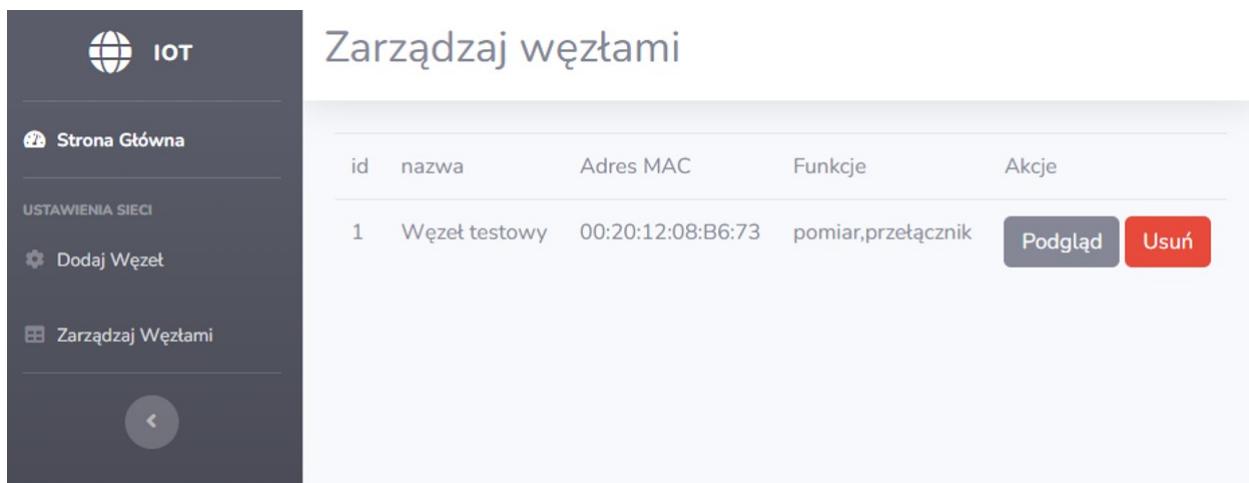


Rysunek 6.8. Widok dodawania nowego węzła po wpisaniu adresu MAC.



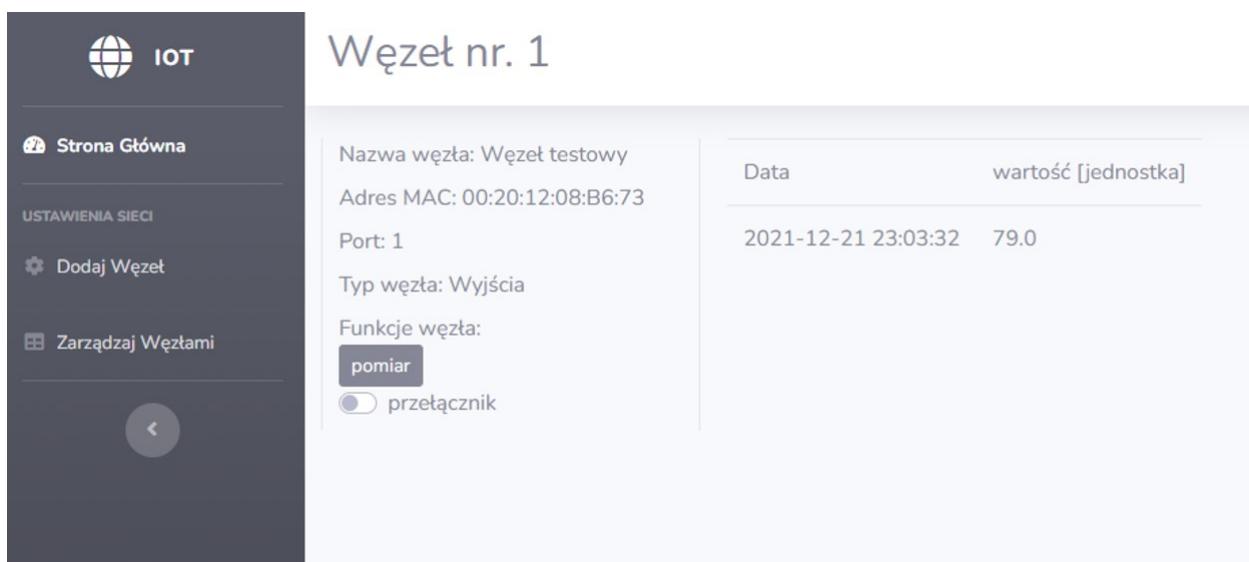
Rysunek 6.9. Komunikat informujący o pomyślnym dodaniu węzła.

Od tej pory dodany węzeł jest powiązany ze stacją i można uzyskać do niego dostęp poprzez zakładkę „Zarządzaj węzłami”



Rysunek 6.10. Widok zarządzania węzłami

Na rysunku 6.6 znajduje się tabela ze wszystkimi węzłami powiązanymi ze stacją. Znajduje się w niej id węzła, czyli unikatowy identyfikator przydzielony przez stację oraz informacje pochodzące z bazy danych, czyli nazwa węzła i funkcje jakie pełni. W kolumnie „Akcje” znajduje się przycisk „Usuń” służący do usuwania węzła z tabeli oraz przycisk „Podgląd” który prowadzi do widoku służącego do komunikacji z węzłem.



Rysunek 6.11. Widok szczegółów dotyczących danego węzła

Na rysunku 6.7 przedstawiono widok po wcisnięciu przycisku podgląd. Znajdują się w nim podstawowe informacje o węźle oraz elementy interfejsu służące do wydawania poleceń. Wywołanie metody, która odczyta wartość z bufora portu szeregowego i zapisze ją do bazy danych jest możliwe poprzez aktywowanie przycisku „pomiar”. Podobnie działa pole wyboru „przełącznik”. Po kliknięciu go przez użytkownika wysyłane jest żądanie do stacji usługowej, aby wydała ona polecenie węzłowi zgaszenia bądź zapalenia diody LED.

## 7. Podsumowanie

W ramach projektu niniejszej pracy udało się osiągnąć cel projektu i spełnić wszystkie jego założenia. Realizacja projektu obejmowała następujące zagadnienia:

- Konstrukcja dedykowanej stacji usługowej Internetu Rzeczy – skonfigurowano komputer jednopłytkowy poprzez zainstalowanie odpowiedniego oprogramowania tak aby przystosować urządzenie do pełnienia funkcji serwera www. Stworzono dedykowane oprogramowanie w tym:
  - Skrypt w języku Python odpowiedzialny za procedurę nawiązywania łączności bezprzewodowej z wykorzystaniem standardu Bluetooth.
  - Aplikacja Internetowa języka Python wraz z biblioteką Flask służąca do obsługi żądań protokołu HTTP pochodzących z przeglądarki.
  - Warstwa wizualna aplikacji internetowej wyświetlana w przeglądarce opracowana przy pomocy takich technologii jak HTML i CSS oraz JavaScript.
  - Relacyjna baza danych oparta na standardzie SQL.
- Przeprowadzenie weryfikacji eksperymentalnej – wykazano negatywny wpływ zakłóceń takich jak obce sygnały w paśmie 2,4GHz na niezawodność komunikacji przy użyciu standaru Bluetooth. Zmierzono największą odległość, która gwarantuje bezbłędną transmisję w środowisku testowym którym był blok mieszkalny i wynosi ona 10 metrów. Pomiary powtórzone dla domu jednorodzinnego, czyli środowiska o znaczco mniejszej liczbie zbędnych sygnałów co poskutkowało zwiększeniem maksymalnego zasięgu o 20%. Ponad to dokonano testów manualnych wersji demonstracyjnej stacji usługowej. Udało się w pomyślnie ustanowić połączenie węzła ze stacją przy użyciu graficznego interfejsu użytkownika w postaci aplikacji Internetu. Urządzenie działa i posiada wszystkie założone funkcjonalności.

# Bibliografia

- [1] Dave Evans, "The Internet of Things: How the Next Evolution of the Internet Is Changing Everything", CISCO White Paper, kwiecień 2011, s. 3.
- [2] Smart Energy Efficient Home Automation System Using IoT, IEEE, 29 July 2019, DOI: 10.1109/IoT-SIU.2019.8777607
- [3] CRAIoT: Concept, Review and Application(s) of IoT, IEEE, 29 July 2019, DOI: 10.1109/IoT-SIU.2019.8777467
- [4] Sudhir K. Routray, Sharath Anand, NARROWBAND IOT FOR HEALTHCARE, IEEE, 19 October 2017, DOI: 10.1109/ICICES.2017.8070747
- [5] S. Rajendran, T. Hari Prasath, S. Revathi, K. Rajesh, Basic Food Safety Monitoring And Enhancement in Coffee Industry Using IOT, IEEE, 20 April 2021, DOI: 10.1109/ICACI-TE51222.2021.9404609
- [6] Mike O. Ojo, Stefano Giordano, Gregorio Procissi, Ilias N. Seitanidis, A Review of Low-End, Middle-End, and High-End Iot Devices, December 18, IEEE, 09 November 2018, DOI: 10.1109/ACCESS.2018.2879615
- [7] Yaxing Yao, Justin Reed Basdeo, Smirity Kaushik, Yang Wang, Defending My Castle: A Co-Design Study of Privacy Mechanisms for Smart Homes, IEEE, May 4–9, 2019, DOI: 10.1145/3290605.3300428
- [8] Hittu G., Mayank D., Securing IoT Devices and Securely Connecting the Dots Using REST API and Middleware, IEEE, 18-19 April 2019, DOI: 10.1109/IoT-SIU.2019.8777334
- [9] URL: <https://www.statista.com/statistics/976313/global-iot-market-size/>
- [10] bluetooth.org, Bluetooth Core Specification v.5.2, s.196, 2019-12-31
- [11] Rozporządzenie Ministra Administracji i Cyfryzacji z dnia 12 grudnia 2014 r. w sprawie urządzeń radiowych nadawczych lub nadawczo-odbiorczych, które mogą być używane bez pozwolenia radiowego. URL: <http://isap.sejm.gov.pl/isap.nsf/DocDetails.xsp?id=WDU20170000096>
- [12] 802.15 WPAN Task Group 6 Body Area Networks, IEEE, 13 January 2022 URL: <https://www.ieee802.org/15/pub/TG6.html>
- [13] Poslad Stefan, Ubiquitous Computing Smart Devices, Smart Environments and Smart Interaction, s. 15, 2009
- [14] Gratton, Dean A, The Handbook of Personal Area Networking Technologies and Protocols, Cambridge University Press. Strony 15–18, 2013
- [15] Gary A. Donahue, Network Warrior, O'Reilly. s. 5, June 2007
- [16] Groth, David and Skandler, Toby, Network+ Study Guide, Fourth Edition, s. 121 June 2007
- [17] Dyrektywa Parlamentu Europejskiego i Rady 2014/53/UE z dnia 16 kwietnia 2014 r. w sprawie harmonizacji ustawodawstw państw członkowskich dotyczących udostępniania na rynku urządzeń radiowych i uchylająca dyrektywę
- [18] Nyein Aye Maung Maung , Win Zaw, Comparative Study of RSS-based Indoor Positioning Techniques on Two Different Wi-Fi Frequency Bands, IEEE, 24-27 June 2020, DOI: 10.1109/ECTI-CON49241.2020.9158211

## Bibliografia

---

- [19] Dilip Geevarghese, BLE vs Wi-Fi: Which is Better for IoT Product Development?, February 01, 2018, URL: <https://www.cabotsolutions.com/ble-vs-wi-fi-which-is-better-for-iot-product-development>
- [20] Bradley Mitchell, How Fast Is a Wi-Fi Network?, June 16 2021
- [21] bluetooth.com, Understanding Bluetooth Range,  
URL: <https://www.bluetooth.com/learn-about-bluetooth/key-attributes/range/>
- [22] HC-05 Bluetooth Module User's Manual V1.0 URL: <https://www.gme.cz/data/attachments/dsh.772-148.1.pdf>
- [23] raspberrypi.org, Raspberry Pi 4 Computer Model B – product brief

# Spis rysunków

2.1.	Diagram różnych rodzajów sieci . . . . .	10
2.2.	Przykłady topologii sieci, po lewej stronie topologia gwiazdy, po prawej stronie topologia siatki . . . . .	11
2.3.	Topologia mieszana . . . . .	11
2.4.	Moce poszczególnych sygnałów Wi-Fi w paśmie 2,4 GHz . . . . .	13
2.5.	Moce poszczególnych sygnałów Wi-Fi w paśmie 5 GHz . . . . .	13
2.6.	Schemat blokowy opisywanej architektury . . . . .	15
3.1.	Schemat blokowy systemu IoT z dedykowaną stacją usługową . . . . .	16
4.1.	Komputer jednopłytowy Raspberry Pi model 4b użyty do konstrukcji stacji usługowej . . . . .	19
4.2.	Diagram strosu technologicznego stacji usługowej . . . . .	20
4.3.	Elementy, które posłużyły do konstrukcji węzła, Arduino UNO i moduł Bluetooth HC-05 . . . . .	22
5.1.	Zrzut ekranu działającego szkieletu aplikacji . . . . .	24
5.2.	Szkic bazy danych . . . . .	25
5.3.	Schemat blokowy algorytmu do nawiązywania łączności bezprzewodowej poprzez Bluetooth . . . . .	27
5.4.	Schemat połączeń elektrycznych węzła sieci . . . . .	28
6.1.	Schemat układu pomiarowego . . . . .	31
6.2.	Schemat środowiska testowego nr 1 . . . . .	32
6.3.	Moce poszczególnych sygnałów Wi-Fi w otoczeniu nadajnika w środowisku testowym nr 1 . . . . .	33
6.4.	Schemat środowiska testowego nr 2 - parter . . . . .	33
6.5.	Schemat środowiska testowego nr 2 - pierwsze piętro . . . . .	34
6.6.	Moce poszczególnych sygnałów Wi-Fi w otoczeniu nadajnika w środowisku testowym nr 2 . . . . .	34
6.7.	Widok dodawania nowego węzła . . . . .	37
6.8.	Widok dodawania nowego węzła po wpisaniu adresu MAC . . . . .	38
6.9.	Komunikat informujący o pomyślnym dodaniu węzła . . . . .	38
6.10.	Widok zarządzania węzłami . . . . .	39
6.11.	Widok szczegółów dotyczących danego węzła . . . . .	39

# **Spis tabel**

4.1.	Porównanie parametrów opisywanych komputerów jednoplątkowych . . . . .	18
6.1.	Moce poszczególnych sygnałów Bluetooth w otoczeniu nadajnika w środowisku testowym nr 1 . . . . .	32
6.2.	Moce poszczególnych sygnałów Bluetooth w otoczeniu nadajnika w środowisku testowym nr 2 . . . . .	34
6.3.	Wyniki pomiarów mocy przeprowadzonych w środowisku testowym nr. 1 . . . . .	35
6.4.	Wyniki pomiarów sprawności transmisji przeprowadzonych w środowisku testowym nr. 1 . . . . .	35
6.5.	Wyniki pomiarów mocy przeprowadzonych w środowisku testowym nr. 2 . . . . .	36
6.6.	Wyniki pomiarów sprawności transmisji przeprowadzonych w środowisku testowym nr. 2 . . . . .	36

## **Listingi kodu**

5.1	Fragment odpowiedzialny za obsługę żądań przychodzących z serwera . . . . .	23
5.2	Fragment skryptu odpowiedzialnego za tworzenie bazy danych . . . . .	26
5.3	Fragment skryptu automatyzującego nawiązywanie łączności bezprzewodowej	27
5.4	Główna pętla programu mikrokontrolera . . . . .	29