

KIERUNEK: EiT

PRACA DYPLOMOWA INŻYNIERSKA

**Aplikacja mobilna do śledzenia obiektów z
wykorzystaniem Bluetooth 4.0**

**Mobile application to track objects using
Bluetooth 4.0**

AUTOR:
Michał Hałucha

PROMOTOR:
Dr inż. Krzysztof Urbański

OCENA PRACY I PODPIS PROMOTORA:

Spis treści

1	Wstęp	3
2	Cel i zakres pracy	5
3	Internet of Things	6
4	Bluetooth Low Energy	8
5	Śledzenie obiektów	9
6	Beacon.....	10
6.1	Zastosowania	11
6.2	Jaka alternatywa?.....	11
6.3	Porównanie	11
7	Środowisko programowe	12
7.1	Android Studio	12
8	Środowisko sprzętowe	13
8.1	NRF51822	13
9	Schemat działania systemu	14
9.1	Aplikacja użytkownika	15
9.2	Komunikacja z serwerem	21
9.3	Use Case	22
10	Testy.....	23
11	Podsumowanie	26
12	Literatura.....	27

1 Wstęp

Rozwój technologii Bluetooth oraz jej popularność wśród użytkowników urządzeń elektronicznych wywiera nacisk na tworzenie urządzeń o jak najmniejszym poborze energii. Wynika to z faktu, że ludzie coraz częściej sięgają do swoich urządzeń elektronicznych, aby ułatwić sobie życie dzięki szybkości komunikacji i sterowania różnego rodzaju sprzętem elektronicznym. Do tego typu urządzeń zaliczamy m.in. AGD (artykuły gospodarstwa domowego), oświetlenie, sterowanie bramą i drzwiami. Użytkownicy chcący kontrolować tego typu sprzęty stawiają nacisk na urządzenia mobilne. Do wzrostu popularności przyczyniają się małe rozmiary oraz funkcjonalność. Za pomocą telefonu możemy komunikować się z całym światem korzystając z technologii Bluetooth oraz WiFi. Prostota użytkowania telefonów komórkowych zachęca użytkowników w różnym wieku. Mówiąc o komunikacji warto wspomnieć o koncepcji Internet of Things (IoT), czyli Internetu Rzeczy, która ma na celu podłączenie do sieci wszystkich możliwych urządzeń w sposób przewodowy jak również bezprzewodowy w celu zapewnienia wymiany informacji.

Technologia Bluetooth ma bogatą historię. W 1994 firma L.M.Ericsson opracowała wspólnie z czterema innymi firmami standaryzację SIG (Special Interest Group). Głównym celem standaryzacji bezprzewodowej technologii był zamiar wyeliminowania kabli połączeniowych. W lipcu 1999 roku opracowano pierwszą specyfikację wersji Bluetooth 1.0, która cechowała się małą szybkością transmisji oraz stosunkowo wysokim poborem energii. W ciągu kilkunastu kolejnych lat dynamicznie pojawiające się wersje Bluetooth doprowadziły do polepszenia szybkości transmisji danych jak i zwiększenia odległości transmisji oraz znacznego obniżenia poboru mocy. Aktualny najnowszy standard Bluetooth o numerze 4.0 zapewnia wysoki poziom bezpieczeństwa, łatwe nawiązywanie połączeń między urządzeniami, szybki transfer.

W niniejszej pracy podjęto się wykonania aplikacji mobilnej na systemy Android, umożliwiającej „śledzenie” urządzeń elektronicznych w sposób bezprzewodowy. Aplikacja wykorzystuje odbierane informacje do kontrolowania różnych obiektów. Informacje pochodzą od urządzeń o małym zapotrzebowaniu energii, głównie ze względów ekonomicznych. Pozwala to uniknąć częstych wymian źródeł zasilających, takich jak baterie i różnego rodzaju akumulatory. W pracy posłużono się komunikacją bezprzewodową. W tym wypadku przesyłanie informacji odbywa się w sposób bezpołączeniowy. Jedną z nich jest technologia Bluetooth wykorzystywana w

komunikacji „obiekt – użytkownik telefonu” oraz technologia WiFi, czyli komunikacja użytkownika z serwerem, za pomocą której możemy sterować różnego rodzaju automatyką zasilaną elektrycznie.

W rozdziałach zawarte zostały informacje na temat technologii Bluetooth używanej w urządzeniach mobilnych. W pierwszym rozdziale zostały opisane urządzenia prężnie rozwijające się na rynku. W kolejnych rozdziałach zaproponowano i omówiono zaprojektowaną aplikację do szerokich zastosowań. W pracy także umieszczono testy pokazujące jakość oferowanych usług przez aplikację.

2 Cel i zakres pracy

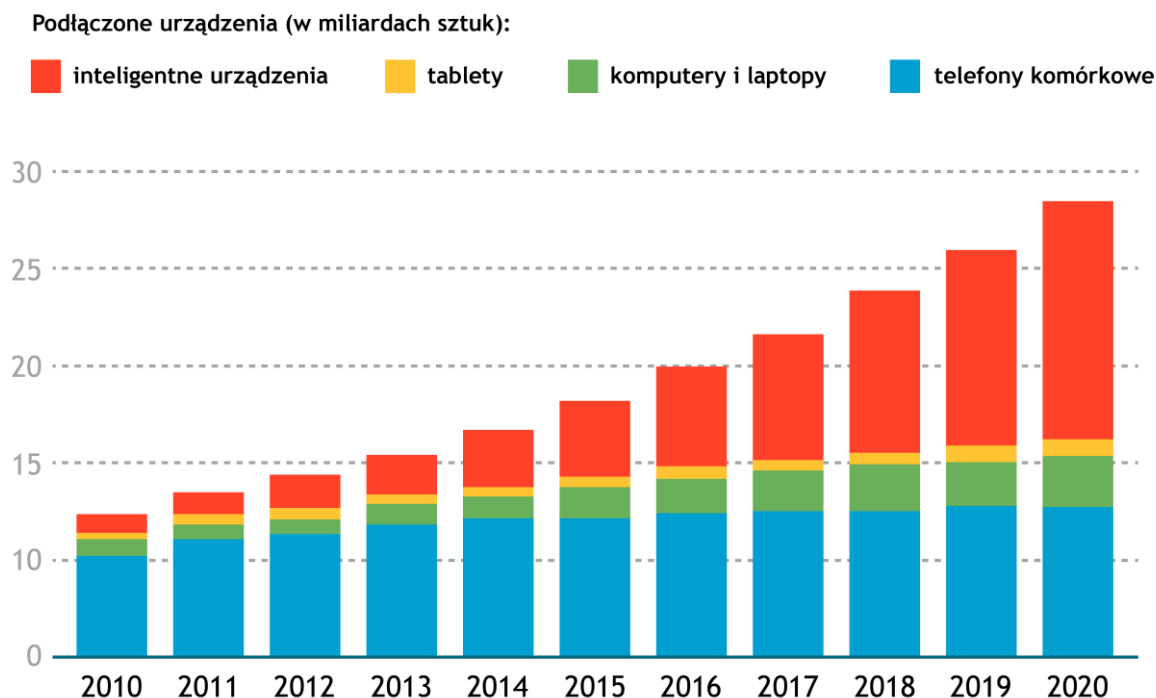
Celem pracy było wykonanie aplikacji mobilnej do śledzenia urządzeń Bluetooth dla systemu operacyjnego Android. Docelowa aplikacja znajduje zastosowanie w przemyśle, turystyce i użytku domowym. Praca obejmuje zagadnienia takie jak:

- analiza urządzeń wykorzystujących Bluetooth (ze szczególnym uwzględnieniem urządzeń typu *Beacon*)
- analiza informacji na temat tworzenia aplikacji mobilnych w oparciu o technologię Bluetooth Low Energy (BLE)
- wykorzystanie środowiska Android Studio do programowania mobilnego w oparciu o język programowania Java oraz język XML
- wykonanie aplikacji mobilnej umożliwiającej wykorzystanie nadajników posiadających Bluetooth
- zrealizowanie komunikacji pomiędzy serwerem a klientem
- umożliwienie użytkownikom zainstalowania aplikacji na swoich urządzeniach oraz zapoznanie się z ograniczeniami sprzętowymi
- testy aplikacji

3 Internet of Things

Internet Of Things jest koncepcją odnoszącą się do obiektów fizycznych połączonych do sieci, posiadających swój własny unikatowy adres IP. Przez komunikację bezprzewodową z Internetem jesteśmy w stanie identyfikować przedmioty mogące bezpośrednio wymieniać dane między sobą. Wykorzystujemy je w późniejszym czasie do zbierania informacji o aktualnym stanie urządzeń oraz reagowanie na odpowiednie dane, służące do sterowania różnymi obiektami elektronicznymi.

Internet daje możliwości komunikowania się pomiędzy urządzeniami innymi niż komputery. Pierwsza próba komunikacji z obiektami odbyła się około 1990 roku. Urządzeniem, które zostało podłączone i sterowane za pomocą Internetu był Toster. Przez kolejne lata liczba urządzeń podłączonych do sieci stale się zwiększała, aktualnie liczba ta wynosi około 22.9 miliardów. Na umieszczonym rysunku[1], widoczny jest wzrost popularności urządzeń stosowanych w IoT. W 2020 roku liczba urządzeń podłączonych do sieci powinna osiągnąć 50 miliardów, czyli odnotowany zostanie ponad dwukrotny wzrost, na przestrzeni zaledwie czterech lat.

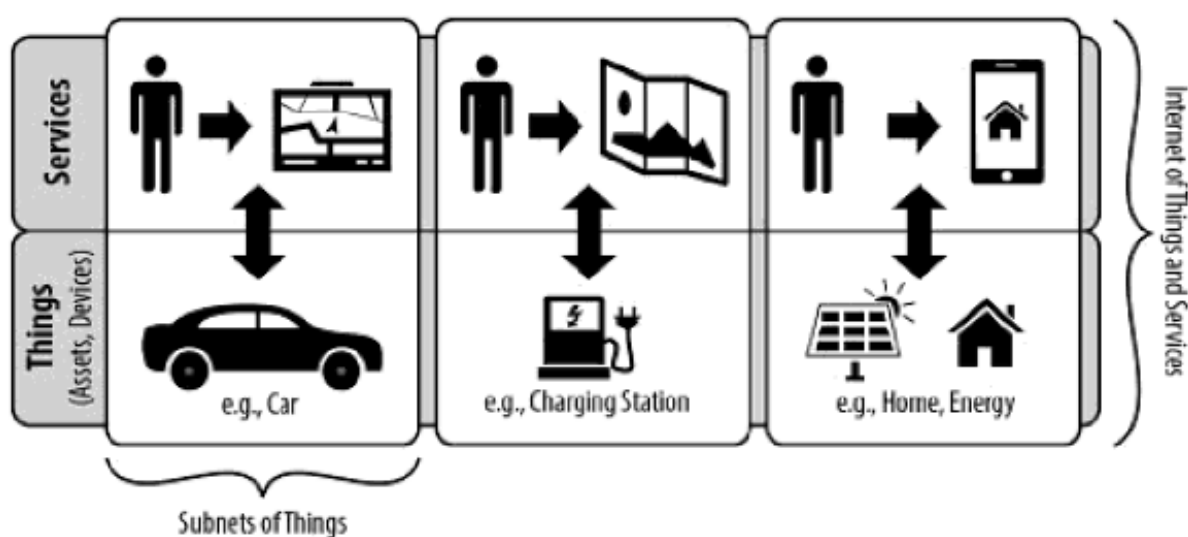


Rysunek 1. Liczba podłączonych urządzeń w latach 2012-2020.

Źródło: http://www.pi.gov.pl/parp/chapter_86197.asp?soid=32C123A05C7443289AAB886F3AE7A100

Ogromny wzrost zainteresowania tym aspektem nasuwa pytanie „Co uczyniło Internet rzeczy takim popularnym?”. Z perspektywy zmian technologicznych i rynkowych trendów ważne jest umożliwienie łączenia ze sobą coraz to większej ilości urządzeń w sposób tani i prosty. Prowadzi to do wszechobecnej łączności opartej na protokole IP, miniaturyzacji oraz postęp w usługach analitycznych.

Z punktu widzenia klienta główną zaletą oferowaną przez Internet rzeczy będą nowe usługi, pozwalające łączyć różnego rodzaju urządzenia. Rysunek 2 przedstawia przegląd urządzeń z usługami [2].



Rysunek 2. Internet rzeczy i usługi.

Źródło: Dirk Slama, Frank Puhlmann, Jim Morrish; *Enterprise IoT. Strategies and Best Practices for Connected Products and Services* [2]

4 Bluetooth Low Energy

Gwałtowny przyrost zapotrzebowania na tanie i energooszczędne moduły spowodował dynamiczny rozwój wersji Bluetooth. W 2010 roku opublikowano wersję standardu Bluetooth o numerze 4.0. Stanowi ona dopracowaną wersję klasycznego Bluetooth. Celem było zwiększenie prędkości transmisji. W porównaniu do wersji standardowej Bluetooth, pojawiło się dużo zalet dla urządzeń przenośnych, m.in. mały pobór mocy oraz szybki dostęp do danych.

Urządzenia Bluetooth Low Energy (BLE) mogą być nawet 100 – krotnie mniej energochłonne w porównaniu do klasycznego Bluetooth oraz potrafią wysyłać małe porcje danych w bardzo krótkim czasie. Standard ten umożliwia pracę na baterii przez długi czas, który trwa mniej więcej od 3 miesięcy do 2 lat. Nowością w tej technologii jest tryb rozgłaszania, który polega na okresowym rozsyłaniu danych przez urządzenie podrzędne. W określonych pomieszczeniach możemy zdefiniować odpowiednie reakcje na ich rozpoznanie. Wysyłany sygnał jest przetwarzany przez urządzenie skanujące oraz wysyłany w określonym interwale czasowym, który wynosi od 20 ms do 10,24 s. Im krótszy odstęp pomiędzy wysyłanymi sygnałami, tym wyższa częstotliwość pracy, co prowadzi do większego zużycia baterii. W praktyce przepustowość danych 5 - 10KB na sekundę, w zależności od ograniczeń obu stron, wyjaśnia zastosowania oraz ograniczenia technologii BLE. W takiej sytuacji BLE nie ma miejsca w sieci na swój debiut, który wyparłby technologie takie jak WIFI czy klasyczny Bluetooth. Natomiast doskonale odnajduje się w przekazywaniu krótkich informacji, odpowiednio interpretowanych, a następnie wykorzystywanych do sterowania.

Znacznie uproszczone zostało wykorzystywanie tej technologii, gdzie jedyną możliwością jest struktura gwiazdy. Pojedyncze urządzenie może działać jako urządzenie nadrzędne (Master) lub podrzędne (Slave), jednak nie może działać równocześnie. Urządzenie odpowiednio skonfigurowane może komunikować się z wieloma urządzeniami podrzędnymi. Zasięg jaki uzyskują urządzenia BLE został zwiększony aż do 100 m. Jedyną wadą BLE jest znacznie obniżony transfer danych, do 305 kbps/s, kosztem znacząco ograniczonego poboru energii. Rysunek 3 przedstawia porównanie parametrów Bluetooth i Bluetooth Low Energy [3].

	Standardowy Bluetooth	Bluetooth Low Energy
Częstotliwość	2,4 GHz	2,4 GHz
Zasięg	10-100 metrów	10-100 metrów
Przepustowość	0,7 – 2,1 Mbps	305 kbps
Liczba urządzeń w sieci	7	nieograniczona
Czas odpowiedzi	100+ ms	<6ms
Zużycie energii	1(odniesienie)	0,01-0,5

Tabela 1. Porównanie parametrów Bluetooth i Bluetooth Low Energy.

Kolejne wersje Bluetooth 4.0 (BLE) to 4.1 i 4.2 pojawiające się odpowiednio po 2012 roku. Następcy wersji 4.0 zostali wzbogaceni o szybsze transfery, wyższe poziomy bezpieczeństwa, łatwiejsze nawiązywanie łączności z przedmiotami oraz bezpośrednią łączność.

Urządzenia BLE mogą pracować w dwóch następujących trybach:

- Single Mode oznacza tryb, w którym komunikacja jest możliwa pomiędzy tylko i wyłącznie urządzeniami z Bluetooth Low Energy
- Dual Mode oznacza możliwość komunikacji z urządzeniami zarówno Bluetooth Low Energy jak i Bluetooth Classic.

Moduły BLE znajdują się w ofercie głównych dostawców takich jak Texas Instruments, Nordic Semiconductors czy Bluegiga. Kompletne urządzenia korzystające z tych innowacyjnych rozwiązań noszą nazwę *Beacons*. Zostały one wykorzystane do niniejszej pracy.

5 Śledzenie obiektów

Występuje kilka sposobów śledzenia obiektów drogą radiową. Pierwszym bardzo powszechnie stosowanym sposobem śledzenia urządzeń jest technologia GPS umożliwiająca lokalizację pod warunkiem, że sygnał GPS odbierany jest przez odbiornik z akceptowalną wydajnością. Drugim sposobem jest posłużenie się nadajnikami Bluetooth. W tym przypadku należy brać pod uwagę urządzenia

odbierające sygnał zmieniający się cyklicznie o określonej mocy bądź określające lokalizację danych nadajników.

Urządzeniami wchodzącymi w skład obiektów śledzonych są na przykład nadajniki *Beacon*. Nadajniki te pracują w sposób okresowy, czyli nadają sygnał w danych odstępach czasu. Przekłada się to na niski pobór energii oraz długą żywotność baterii przy stosunkowo długim czasie pracy. Przewidywane czasy użytkowania nadajnika przy jednej baterii, określone przez producentów, wahają się między 4 miesiącami do 2 lat. Ograniczenia ze strony urządzeń obsługujących aplikację oraz czytające sygnał z *Beacon* sprowadzają się do posiadania odpowiedniej minimalnej wersji Bluetooth. Minimalna wymagana wersja do obsługi BLE to wersja o numerze 4.0. Występuje ona przeważnie w wersjach oprogramowania Androida 4.3 lub nowszych.

6 Beacon

Obiekty *Beacon* są to małe urządzenia składające się z modułu Bluetooth oraz baterii. Moduł rozsyła określony sygnał, który kompatybilne urządzenia rozpoznają jako numer identyfikujący UUID. UUID jest identyfikatorem 126 bitowym (16 bajtowym), to numer unikatowy, czyli jedyny stosowany na całym świecie. *Beacon*, który jest tylko i wyłącznie transponderem oraz jest odpowiednio skonfigurowany, wykorzystuje maksymalną żywotność akumulatora. W przypadku *Beacon* mamy możliwość dowolnego ustawienia zasięgu nadawania (konfiguracja mocy nadajnika). Daje to nieograniczone możliwości, np. rozmieszczenia ich w dowolnych pomieszczeniach, poza którymi nie będą widoczne. Funkcjonalność jaką można przypisać do aplikacji, to reakcja na ich "Widzenie". Często mówi się o *Beacon* w kontekście wyświetlania powiadomień o spersonalizowanych ofertach i promocjach oraz o śledzeniu osób korzystających z usług lokalizacyjnych.

Smartfony z systemem operacyjnym Android oraz z uruchomioną aplikacją, na żądanie użytkownika reagują na rozmieszczone obiekty. Odbywa się to za pomocą skanowania dostępnych *Beacons*. Po skanowaniu następuje ewentualna identyfikacja danego nadajnika oraz wywołanie odpowiedniej pracy przypisanej do danego urządzenia.

6.1 Zastosowania

Beacons odnajdują się w użytkowaniu publicznym. Główne zastosowania sprowadzają się do:

- usług świadczonych przez sklepy
- zameldowania w określonym miejscu
- zbierania informacji o lokalizacji użytkownika
- reakcji na zbliżanie się użytkowników do danego nadajnika *Beacon*

6.2 Jaka alternatywa?

Alternatywą dla standardu Bluetooth Low Energy wykorzystywanego przez *Beacon*, są urządzenia wykorzystujące protokoły bezprzewodowe takie jak ZigBee, Ant+.

ZigBee jest protokołem o podobnym działaniu do protokołu BLE. Transmisja danych odbywa się w sieciach typu mesh, cluster tree. Protokół ten, tak jak w przypadku Bluetooth, charakteryzuje się niewielkim poborem energii, przepustowością danych do 250 kbps oraz zasięgiem między węzłami do 100 metrów.

Urządzenia ZigBee dzielimy na trzy typy:

- Koordynator, który służy jako węzeł początkowy oraz do którego zostają podłączone urządzenia. Pełni rolę urządzenia zbierającego dane
- Router, który przekazuje pakiety
- Urządzenie końcowe, które przesyła dane do routera do którego jest połączone

Ant+ został zaprojektowany do komunikacji urządzeń sportowych. Początkowo służył do komunikacji pulsometrów z opaską. Stopniowo zaczął obsługiwać coraz więcej urządzeń takie jak wagi, czujniki prędkości, pomiar mocy, sprzęt fitness. Zaletą tak jak w przypadku BLE, jest komunikacja między sprzętem wielu producentów. Możemy dobierać akcesoria według uznania, wygody czy oferowanych funkcji.

6.3 Porównanie

ZigBee w porównaniu do Bluetooth służy do automatyzacji, gdzie w przypadku Bluetooth jest to wymiana danych pomiędzy urządzeniami przenośnymi. Zwracając uwagę na prawie natychmiastowy czas łączenia z siecią, który wynosi 30 milisekund,

ZigBee jest bardziej odpowiedni dla aplikacji wymagających reakcji natychmiastowej. Protokół Ant+ w porównaniu do BLE, przez długi czas nie będzie stosowany w wielu urządzeniach. Głównym powodem jest wysoka cena modułu nadajnika.

7 Środowisko programowe

7.1 Android Studio

Środowisko programowe Android studio, firmy Google, pozwala tworzyć wysokopoziomowe oprogramowanie na urządzenia z systemem Android. Pierwsze wydanie Android Studio miało miejsce w maju 2013 roku. Nowy program wyparł, wcześniej stosowane środowisko programistyczne Eclipse. Android Studio umożliwia korzystanie z najnowszych wersji API, narzędzi i bibliotek, służących do tworzenia profesjonalnych aplikacji. Odpowiednie Android SDK, daje możliwość programowania i debugowania aplikacji na urządzenia mobilne (smartfony i tablety), ubieralne (zegarki i okulary) oraz telewizory i samochody.

Projekt budowania aplikacji na urządzenia mobilne, w środowisku Android Studio, sprowadza się do znajomości dwóch języków programowania - XML i Java. Język XML, czyli uniwersalny język znaczników. W środowisku Android Studio ma on szeroko rozwinięte zastosowanie:

- pozycjonowanie elementów znajdujących się na layoucie
- definiowanie parametrów odpowiedzialnych za wygląd poszczególnych elementów
- odwołania obiektowe do treści zawartych w aplikacji
- przechowywanie informacji na temat współgrających ze sobą layoutów

Język Java umożliwia w sposób obiektowy realizację wszystkich funkcjonalności aplikacji mobilnej. Dzięki współgraniu tych dwóch języków programowania jesteśmy w stanie zbudować wielofunkcyjną aplikację mobilną, dostosowaną do użytkowników indywidualnych lub zastosowań firmowych. Obiektość języka Java w połączeniu z językiem XML w tym przypadku daje nam duże możliwości na rozbudowywanie projektu w bardzo łatwy i intuicyjny sposób.

Android Studio umożliwia debugowanie, które jest ważnym etapem wykorzystywanym do sprawdzenia poprawności kodu używanego na docelowym urządzeniu. Debugger Android Studio umożliwia:

- wybranie urządzenia
- zbadania zmiennych i ocena ich w czasie wykonywania operacji logicznych
- sprawdzenia reakcji na zdarzenia wywołane przez użytkownika
- zrobienia zrzutu ekranu ze swojej aplikacji

8 Środowisko sprzętowe

8.1 NRF51822

Firma Nordic od wielu lat angażuje się w rozwiązania bezprzewodowe o małej mocy. Jest to produkt składający się z radia i procesora ARM w jednym, niedrogim pakiecie.

Charakterystyczne cechy to:

- rdzeń ARM Cortex-M0 pracujący przy częstotliwości 16 MHz
- pamięć flash w formacie 128 KB
- 16 KB pamięci SRAM



Rysunek 3. Beacon NRF51822.

Na załączonym obrazku przedstawiono Beacon nadający sygnał z technologią Bluetooth Low Energy. Zawiera on szeroko dostępny układ SoC (system on chip) BLE IC wyprodukowany przez Nordic Semiconductor. Cechy jakie posiadają te układy to:

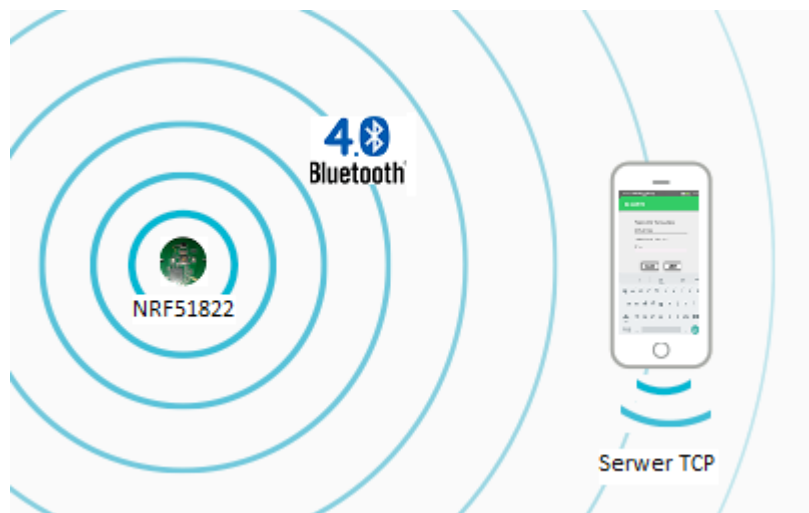
- przepustowość danych, która jest ograniczona maksymalnie do sześciu połączeń
- każdy wychodzący pakiet danych może zawierać do 20 bajtów danych użytkownika
- maksymalnie 133 zdarzeń połączeń (pojedyncza wymiana pakietów między dwoma rówieśnikami) na sekundę i 120 bajtów na każde połączenie (6*20 bajtów) sugeruje, że przesyłanie odbywa się z maksymalną szybkością transmisji NRF51822.

W rzeczywistości efektywna przepustowość wynosi:

133 połączenia na sekundę * 120 bajtów = 15960 bajtów/s albo 0,125 Mbit/s (~125 kbit/s)

9 Schemat działania systemu

Głównym założeniem niniejszej pracy było wykonanie aplikacji mobilnej do śledzenia obiektów z wykorzystaniem Bluetooth 4.0. Efekt działania wykonanej aplikacji został przedstawiony przy współpracy z aplikacją serwerową, Beaconami sprowadzonymi z Chin, wyprodukowanymi przez firmę Hongyi Yunjia, oraz telefonem HUAWEI Y560 z wersją Androida 5.1.1 i Bluetooth wersji 4.0.



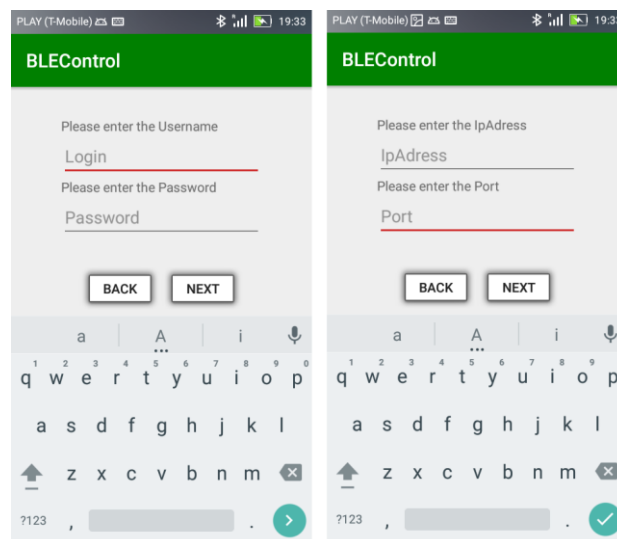
Rysunek 4. Schemat działania systemu

9.1 Aplikacja użytkownika

Aplikacja użytkownika oferuje szeroko pojęte zastosowanie „śledzenia” urządzeń Bluetooth o standardzie Bluetooth Low Energy. Model demonstracyjny pozwala kontrolować oświetleniem dzięki danym odbieranym od nadajników, przez urządzenie mobilne. Przy projektowaniu aplikacji posłużono się wzorcem projektowym MVP (Model-View-Presenter) gdzie:

- Model – moduł odpowiadający za dane
- View – moduł widoku, którym w praktyce staje się klasa Activity
- Presenter – moduł prezentera, odpowiada za całą logikę biznesową stojącą za Activity

Aplikacja została wykonana w środowisku programistycznym Android Studio. Po uruchomieniu aplikacji następuje proces logowania, czyli wywołania aktywności przez funkcję „onCreate” oraz przypisaniu źródła pliku, w którym opisany jest wygląd ekranu logowania. Po wprowadzeniu poprawnych danych, przechodzimy za pomocą przycisków (next, back) do następnego etapu. Dane serwera oraz portu zostają zapisane pod zmiennymi (port, adres), które następnie są przenoszone między aktywnościami dzięki paczką Bundle. Poniżej umieszczono graficzny wygląd formularzy oraz najważniejsze fragmenty kodu.



Rysunek 5. Aktywność „Login”

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_login);
}
```

R(resources) – oznacza zasoby.

Layout – nazwa folderu w którym zawarte są pliki XML dotyczące wyglądu oraz personalizacji ekranu.

activity_login – wskazanie na plik aktywności.

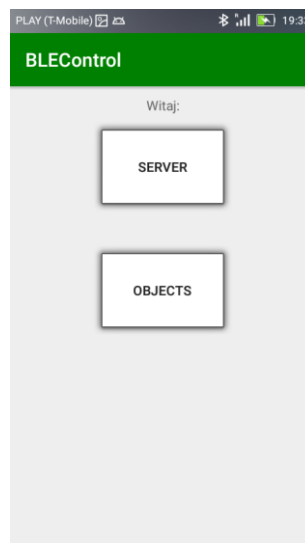
Poniżej umieszczono przykład kodu odpowiadającego za poprawność logowania oraz przejście z użytkownika do formularza związanego z portem i adresem.

```
Button buttonnext = (Button)findViewById(R.id.button6);
buttonnext.setOnClickListener((v) -> {
    if(status.equals("1")) {
        Login = etlogin.getText().toString();
        password = etpassword.getText().toString();
        Log.i("Login", Login);
        Log.i("Password", password);
        if (Login.equals("a") & password.equals("a")) {
            Log.i("czy prawidłowe", Login);
            etlogin.setHint("IpAdress");
            tvlogin.setText("Please enter the IpAdress");
            etpassword.setHint("Port");
            tvpassword.setText("Please enter the Port");
            status = "2";
        }
    }
}
```

Kolejnym etapem jest wywołanie aktywności dotyczącej klasy „MainPanel”, która ułatwia poruszanie się po aplikacji. Klasa „MainPanel” została wyposażona w metody przejścia takie jak:

```
Intent intent = new Intent(MainPanel.this, LightControl.class);
startActivity(intent);
```

Są to „Intencje” pozwalające poruszanie się między aktywnościami.



Rysunek 6. Aktywność „MainPanel”

Główną aktywnością, na której niniejsza praca została zrealizowana, jest skanowanie nadajników Bluetooth Low Energy. „LightControl”, czyli aktywność powstała z myślą o sterowaniu oświetleniem dotycząca śledzenia obiektów. Kluczowe pojęcia jakie są niezbędne do pracy z Bluetooth Low Energy oraz zrozumienia działania klasy LightControl sprowadzają się do znajomości zasad wymiany danych.

W przypadku BLE są to:

- General Access Profile (GAP) – określa jak urządzenia współpracują ze sobą. Dotyczy to wykrywania urządzeń, połączenia, ustanowienia zabezpieczeń i inne.
- GATT (Services and Characteristics) - Wprowadza podstawowe pojęcia usług i charakterystyk, a także procedury, które pozwalają na podłączenie urządzeń do wymiany danych ze sobą. Wszystkie profile aplikacji Low Energy są oparte na profilu GATT.

Role i obowiązki stosowane w przypadku używania nadajników BLE do komunikacji z telefonem:

- środkowe i peryferyjne które dotyczą tego samego połączenia BLE. Urządzenie w centralnej roli skanuje oraz poszukuje reklamy, a urządzenie na peryferiach robi reklamę
- serwer GATT vs. klient GATT. Określa komunikację dwóch urządzeń ze sobą, po nawiązaniu połączenia.

Profile GATT obejmują wszystkie procedury i formaty danych dzięki czemu:

- zezwalają urządzeniom na fizyczne zlokalizowanie innych urządzeń
- wykrywają obecność lub brak pobliskich urządzeń (na przykład sygnał dźwiękowy, jeśli obiekt opuszcza pomieszczenie)
- przenoszą dane HID na BLE (klawiatury, myszki, piloty)
- odczytują poziom glukozy
- odczytują temperaturę ciała
- umożliwiają odczyt prędkości jazdy na rowerze

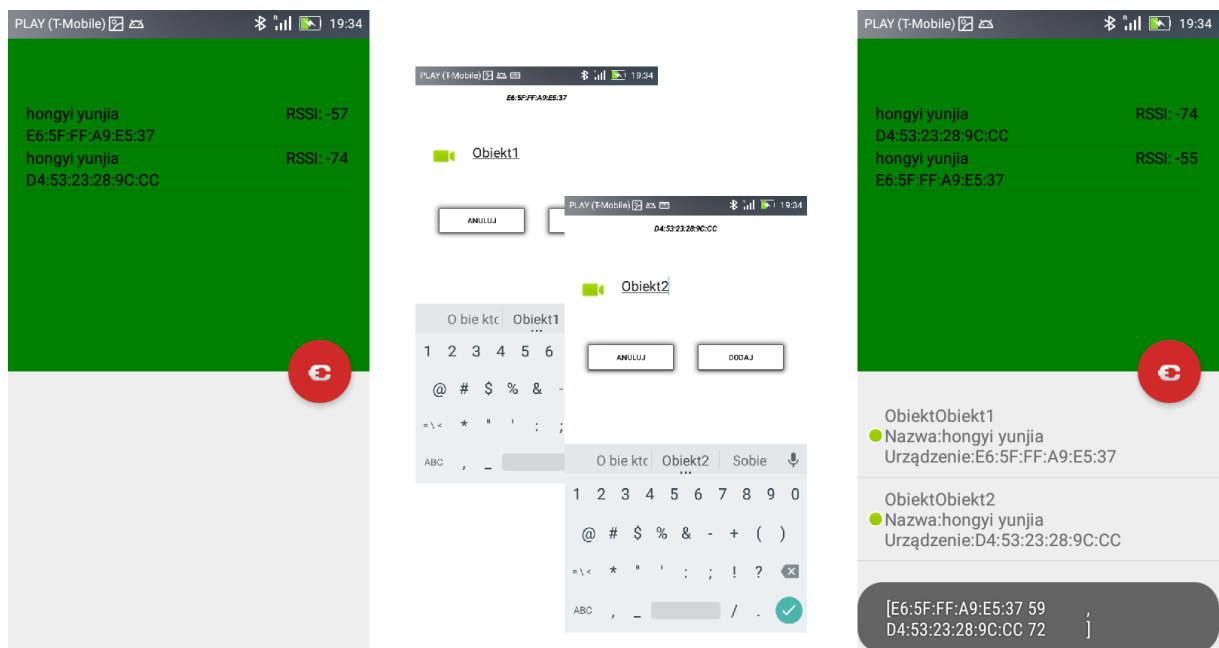
Pełna lista profili zatwierdzonych przez SIG jest dostępna w specyfikacji Bluetooth SIG.

Klasa `LightControl` wraz z towarzyszącym jej układem graficznym `activity_light_control.xml` została zaprojektowana w sposób przyjazny dla użytkownika. Składa się ona z dwóch list. Lista pierwsza, po wywołaniu funkcji skanującej, zostaje wypełniona urządzeniami BLE znajdującymi się w otoczeniu określonym przez zakres odczytu sygnału oraz konfigurację nadajników.

Gdy wywołamy zdarzenie skanowania, przez naciśnięcie przycisku zostaje uruchomiony cały proces oparty na:

1. sprawdzeniu włączonego Bluetooth na telefonie użytkownika
2. skanowaniu dostępnych obiektów Bluetooth Low Energy
3. komunikacji z obiektem oraz odczytanie jego adresu mac, RSSI oraz nazwy
4. dodaniu obiektu o odpowiedniej nazwie, adresie, oraz wartości RSSI
5. wysłaniu dolnej listy urządzeń na serwer w celu dodania wszystkich obiektów przypisanych na przykład do danego pomieszczenia
6. skanowaniu nadajników oraz wysyłaniu obiektów zawierających wartość RSSI z wcześniej zdefiniowanego przedziału (ograniczenie włączania i wyłączania)
7. wysłaniu informacji o nadajnikach zawartych w zdefiniowanym przedziale

RSSI jest to wartość, za pomocą której możemy określić siłę sygnału radiowego. Jest to względny pomiar określony przez każdego producenta tego typu układów scalonych. Nie ma znormalizowanego związku parametru z odczytywaniem wartości RSSI. W rozdziale dotyczącym testów aplikacji przeliczono zależność odległości od wartości RSSI, przy korzystaniu z układów scalonych NRF51822. Pozwala to na konfigurację zapalania światła w zależności od odległości danego pomieszczenia, w którym został umieszczony nadajnik.



Rysunek 7. Aktywność „LightControl”

Implementacja kodu rozpoczyna się od dodania uprawnień do obsługi Bluetooth LE. Ponadto zadeklarowano używanie funkcji „bluetooth_le”, aby aplikacja była dostępna tylko dla obsługujących ją urządzeń.

```
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
<uses-feature android:name="android.hardware.bluetooth_le" android:required="true"></uses-feature>
```

Zanim zostanie uruchomiony proces skanowania aplikacja sprawdza czy funkcja Bluetooth jest włączona. Poniższa metoda dotyczy wyświetlanego okna dialogowego z prośbą o włączenie Bluetooth.

```
public static void requestUserBluetooth(Activity activity) {
    Intent enableBtIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
    activity.startActivityForResult(enableBtIntent, LightControl.REQUEST_ENABLE_BT);
}
```

Następnie zostaje wywołana funkcja `mBTLEScanner.start()`. Zważywszy na fakt, że metoda dziedziczy po klasie `Scanner_BTLE`, zostaje uruchomiony proces skanowania opisany w tej właśnie klasie.

```

private BluetoothAdapter.LeScanCallback mLeScanCallback =
    new BluetoothAdapter.LeScanCallback() {

        @Override
        public void onLeScan(final BluetoothDevice device, int rssi, byte[] scanRecord) {

            final int new_rssi = rssi;
            if (rssi > signalStrength) {
                mHandler.post(new Runnable() {
                    @Override
                    public void run() {
                        ma.addDevice(device, new_rssi);
                    }
                });
            }
        }
    };

```

Gdy zostanie znalezione urządzenie, zostaje wywołana metoda zwrotna, w której można nawiązać połączenie. Poniżej umieszczono kod dotyczący połączenia:

```

public void onConnectionStateChange(BluetoothGatt gatt, int status, int newState) {
    String intentAction;
    if (newState == BluetoothProfile.STATE_CONNECTED) {
        intentAction = ACTION_GATT_CONNECTED;

        mConnectionState = STATE_CONNECTED;

        broadcastUpdate(intentAction);

        Log.i(TAG, "Connected to GATT server.");
        // Attempts to discover services after successful connection.
        Log.i(TAG, "Attempting to start service discovery:" + mBluetoothGatt.discoverServices());
    }
    else if (newState == BluetoothProfile.STATE_DISCONNECTED) {
        intentAction = ACTION_GATT_DISCONNECTED;

        mConnectionState = STATE_DISCONNECTED;

        Log.i(TAG, "Disconnected from GATT server.");

        broadcastUpdate(intentAction);
    }
}

```

W trakcie skanowania występuje cykliczny odczyt wartości RSSI od nadajnika. Długość pomiaru oraz zakres wartości odczytu RSSI zdefiniowany jest w następujący sposób:

```

mBTLEScanner = new Scanner_BTLE(this, 10000, -100);

```

|
|
 Czas skanowania Maksymalna wartość RSSI

Chcąc sterować światłami za pomocą niniejszej aplikacji napisano przykładową funkcję definiującą zakres wartości RSSI, w której oświetlenie może być włączone:

```

int szacowanie = rssa;
int zakres1 = 70;
int zakres2 = 30;
if (zakres1 > szacowanie && szacowanie < zakres2){
    //FRAGMENT ODPOWIEDAJĄCY WYSYŁANIU INFORMACJI Z DANEGO PRZEDZIAŁU NA SERWER
}

```

Wysłanie informacji końcowej, na przykład o dodanych pomieszczeniach, odbywa się za pomocą komunikacji TCP z serwerem. Po całym procesie zostają wysyłane adresy nadajników tylko tych, które mieszczą się w przedziale zdefiniowanym w aplikacji dla konkretnego użytkownika. Po stronie serwera powinny znaleźć się uprawnienia dla konkretnych adresów MAC oraz połączeń elektrycznych.

9.2 Komunikacja z serwerem

Fragment kodu odpowiadający za komunikację z serwerem umieszczono poniżej.

```
String textUserName = "Michał";
String textAddress = "192.168.0.101";

if (blokada == 0) {
    chatClientThread = new ChatClientThread(
        textUserName, textAddress, SocketServerPORT);
    chatClientThread.start();
    blokada = 1;
}

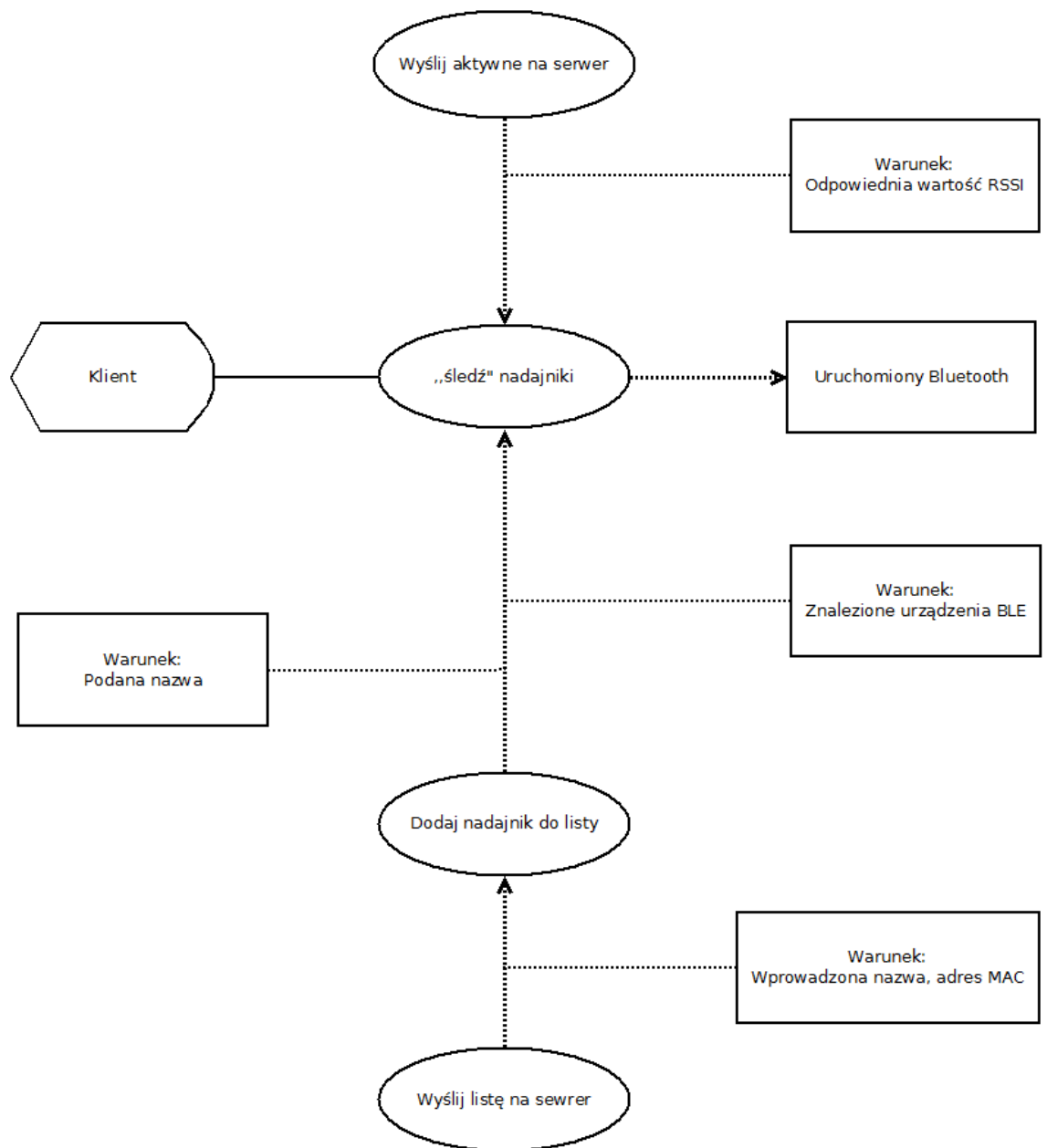
if(chatClientThread==null){
    return;
}

if (gornamac.isEmpty()){
    editSend = "--empty--";
} else {
    editSend = gornamac.toString();
}

// chatClientThread.sendMessage(editSend);
Log.i("Wiadomość",editSend);
// chatClientThread.sendMessage(editTextSay.getText().toString() + "\n");
chatClientThread.sendMessage(editSend);
```

W momencie obsługi wysyłania wiadomości na serwer, sprawdzona zostaje lista wcześniej zeskanowanych urządzeń BLE, zawartych w zdefiniowanym zakresie poniżej wartości 70 RSSI.

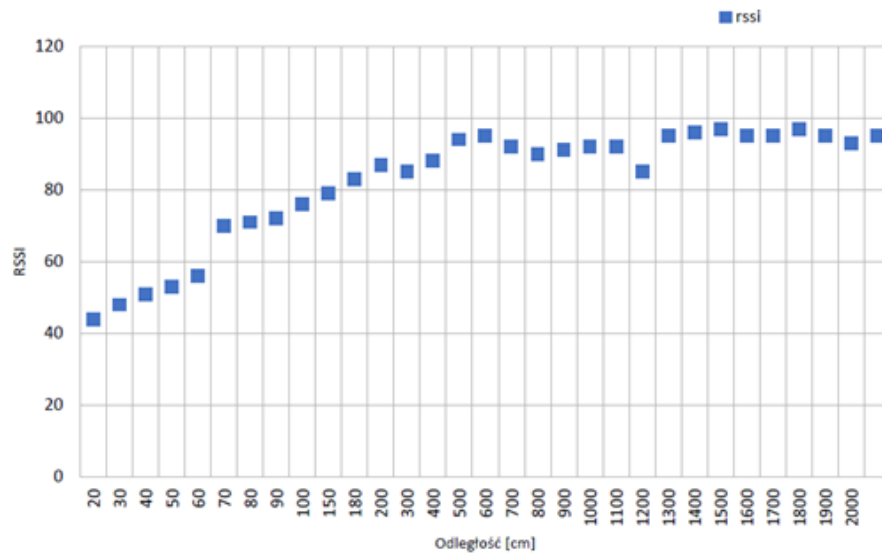
9.3 Use Case



Rysunek 10. Diagram przypadków użycia (Use Case).

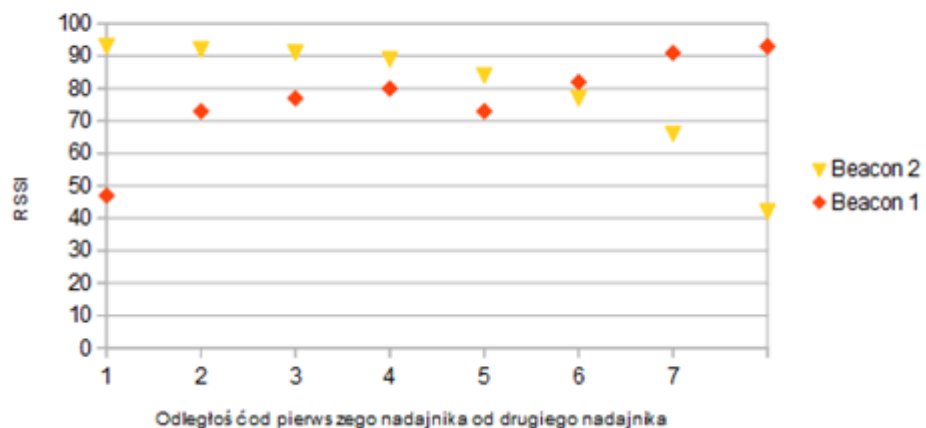
10 Testy

Aplikacja oraz Beacons NRF51822 zostały poddane testom. Zmierzono wartości RSSI od odległości w celu przeliczenia RSSI na rzeczywistą odległość. Na rysunku [11] pokazano liniowość parametru RSSI dla jednego Beaconu.



Rysunek 11. Zależność parametru RSSI od odległości.

Po otrzymanych wynikach możemy zauważyć, że liniowa zależność jest właściwa do odległości 7 metrów. Biorąc pod uwagę kontrole świateł w warunkach domowych, jest to wystarczający zakres odległości dla których możliwe jest określenie przedziałów. Na na rysunku [12] zbadano łatwość lokalizacji użytkownika pomiędzy dwoma Beaconami..



Rysunek 12. Wartość RSSI pomiędzy dwoma nadajnikami.

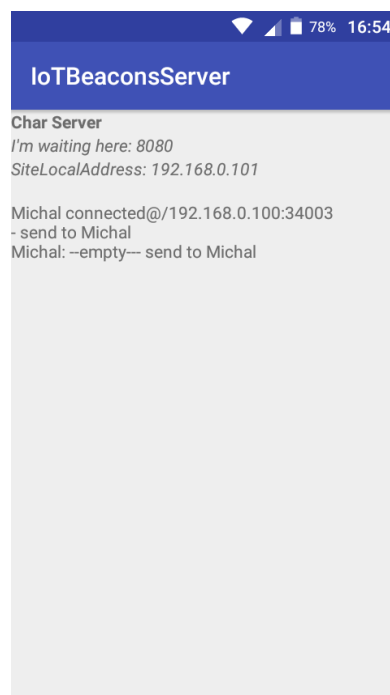
Na wykresie widoczne są liniowe zmiany wartości RSSI. Lokalizacja użytkownika jest możliwa zakładając, że sygnał odczytywany jest bez zakłóceń występujących pomiędzy użytkownikiem a Beaconem. W przypadku poruszania się pomiędzy elementami tłumiącymi sygnał konieczne byłoby zastosowanie algorytmu statystycznego, korzystającego z większej ilości danych.

W trakcie testów przetestowano komunikację z serwerem. Warunki określone przez użytkownika to:

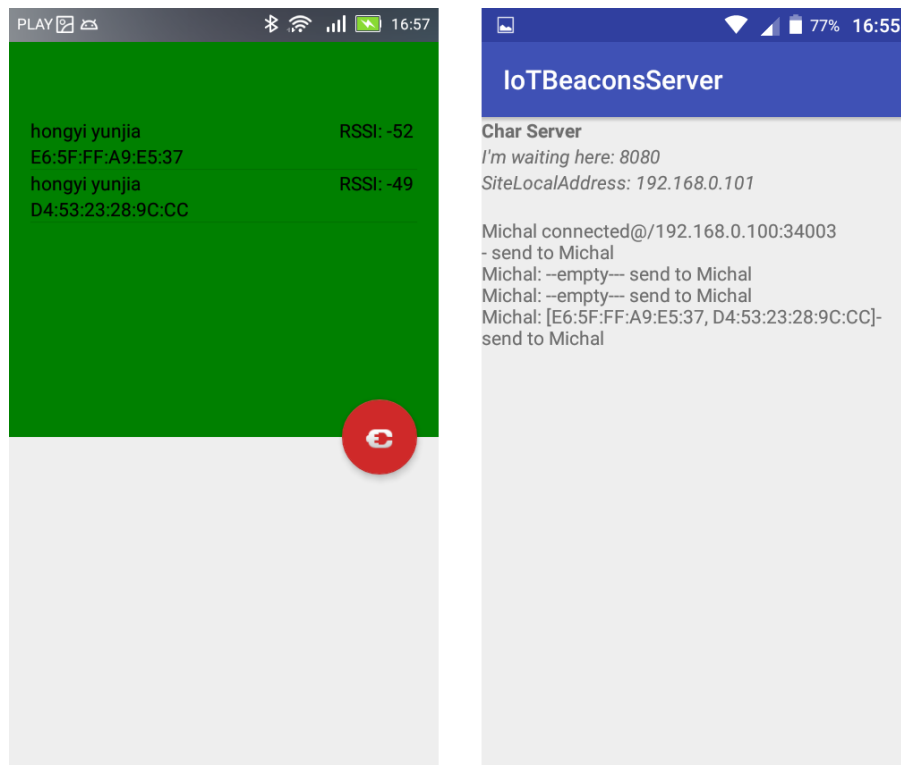
- wysyłanie na serwer informacji o nadajnikach umieszczonych w pomieszczeniach, których wartość RSSI nie przekraczają 70.

Wyniki przedstawiono w następujący sposób:

- rysunek [13] – akcja skanowania oraz wysłania informacji na serwer, w wyniku której, nie znaleziono urządzeń BLE
- rysunek [14] – akcja skanowania oraz wysłania informacji na serwer, w wyniku której, znaleziono urządzenia BLE spełniające wcześniej zdefiniowane warunki



Rysunek 13. Odpowiedź serwera.



Rysunek 14. Odpowiedź serwera.

Wiadomość otrzymana przez serwer zawiera adresy nadajników, dzięki którym możemyysterować włączenie odpowiedniego oświetlenia.

11 Podsumowanie

Celem niniejszej pracy inżynierskiej było zaprojektowanie aplikacji mobilnej, która zademonstruje działanie technologii Bluetooth Low Energy do śledzenia obiektów. W jej skład wchodzi główna aplikacja mobilna do śledzenia urządzeń, aplikacja serwerowa, nadajniki z modułami BLE.

Zaletą aplikacji mobilnych współpracujących z modułami BLE jest połączenie minimalizujące zużycie energii zasilania. Konsekwencją jest praca przy zasilaniu baterijnym przez długi czas. Wszystko to powoduje, że technologia świetnie nadaje się do zastosowania w różnych urządzeniach przenośnych. Obszary zastosowań aplikacji wykorzystujących BLE obejmują opiekę zdrowotną, sport/fitness, systemy bezpieczeństwa, systemy sterowania budynkami oraz kontrola różnych obiektów w przemyśle. Większość modułów BLE jest przeznaczona do zastosowań niewymagających dużych transmisji ilości danych. Sposób kontrolowania odbywa się za pomocą łatwych w użyciu komend tekstowych.

Zrealizowany projekt inżynierski kierowany jest do rosnącej w ostatnim czasie popularności systemów zarządzania elektrycznego budynkami bądź obiektami. Obecnie wydaje się, że jedynym ograniczeniem funkcjonalności aplikacji jest potrzeba klienta oraz inwencja twórcza programisty. Wydaje się, że tego typu aplikacje zrewolucjonizują większość branż.

12 Literatura

[1] Kevin TownSend, *Getting Started with Bluetooth Low Energy*, Wydawnictwo O'Reilly Media 2014.

[2] Erik Hellman, *Platforma Android. Nowe wyzwania*, Wydawnictwo Helion 2014.

[3] Bluetooth Low Energy.

<https://developer.android.com/guide/topics/connectivity/bluetooth-le.html>

[4] Kathy Sierra, Bert Bates, *Java Rusz głowę. Wydanie II*, Wydawnictwo Helion 2012.

[5] Dawn Griffiths, *David Griffiths Android. Programowanie aplikacji. Rusz głowę!*, Wydawnictwo Helion 2016.