

## Opracowanie architektury systemu lokalizacji przebywania pracowników w przestrzeniach wewnętrznych

**Streszczenie.** W artykule przedstawiono wyniki badań nad poprawą jakości transmisji danych oraz możliwościami zwiększenia precyzji pozycjonowania urządzeń Bluetooth Low Energy w zamkniętym środowisku. Przedmiotem badań było stworzone przez firmę urządzenie ISBLE, przeznaczone do sieci sensorowych, które powstało w wyniku realizacji projektu w ramach Regionalnego Programu Operacyjnego Województwa Śląskiego w latach 2014-2020 o tytule: „Opracowanie nowych typów terminali bezprzewodowych w oparciu o zastosowany innowacyjny mikrochipset wraz z algorytmem transmisji danych dla sieci Smart Mesh jako przykład uniwersalnej sieci sensorowej wykorzystującej różne standardy komunikacji bezprzewodowej”.

**Abstract.** The article presents the results of research on improving the quality of data transmission and the possibility of increasing the positioning precision of Bluetooth Low Energy devices in a closed environment. The subject of the research was the ISBLE device created by the company, intended for sensor networks, which was created as a result of the project under the Regional Operational Programme of the Silesian in 2014-2020 with the title: "Development of new types of wireless terminals based on applied innovative microchipset together with data transmission algorithm for Smart Mesh network as an example of universal sensor network using different wireless communication standards". (Development of architecture for employee presence location system in indoor spaces)

**Słowa kluczowe:** Bluetooth Low energy, sieci Mesh, określanie lokalizacji w środowisku zamkniętym.

**Keywords:** Bluetooth Low energy, Mesh networks, location determination in closed environments.

### Wstęp

Określanie pozycji osób w środowisku zamkniętym, w którym nie ma możliwości wykorzystania systemów nawigacji satelitarnej jest coraz częściej wykorzystywane w centrach handlowych oraz w zakładach produkcyjnych. W galeriach handlowych ułatwia nawigację klientom, dzięki której łatwiej mogą odnaleźć interesujący ich sklep lub towar. W zakładach produkcyjnych nawigacja znacznie ułatwia zarządzanie zasobami ludzkimi zmniejszając czas, który pracownik potrzebuje na przemieszczanie się między punktami produkcji.

Do określania pozycji można użyć różnorodnych urządzeń komunikacyjne takie jak smartfony lub beacony BLE (Bluetooth Low Energy). Rozmieszczenie stałych urządzeń lokalizacyjnych w różnych punktach zamkniętego środowiska umożliwia dostarczanie informacji o położeniu ruchomych urządzeń mobilnych w budynkach i halach z dokładnością poniżej jednego metra[1,2]. Usługi lokalizacji wewnątrzbudynkowych mogą również obejmować określanie lokalizacji określonych przedmiotów oraz śledzenie i obserwowanie aktywnych identyfikatorów lub tagów (na przykład w strefach, w które wymagają wysokiego poziomu bezpieczeństwa takich jak lotniska).. Systemy telefonii komórkowej (GSM, LTE itp.) nie zapewnia wymaganej dokładności lokalizacji a systemowi GPS brakuje możliwości pracy w pomieszczeniach. Standard BLE umożliwia stworzenie sieci lokalizacyjnej w oparciu o odpowiednio przygotowaną topologią opartą na aktywnych Beaconach[3,4,8]. Dotychczas znane systemy wewnętrzne wykorzystują właściwości fal radiowych, takie jak siła odbieranego sygnału lub bitowa stopa błędów, do oszacowania położenia urządzenia telekomunikacyjnego lub opierają się na systemach optycznych takich jak IrDA. Jak wykazały przeprowadzone badania oraz symulacje bardzo istotny wpływ na dokładność lokalizacji ma rozmieszczenie beaconów w zamkniętym środowisku z uwzględnieniem struktury pomieszczeń oraz znajdującego się tam wyposażenia. Również prawidłowe rozmieszczenie urządzeń komunikacyjnych wpływa na jakość i szybkość transmisji w budowanych sieciach mesh. Elementami aktywnymi umieszczanymi w określonych lokalizacjach są stworzone w ramach projektu stacje bazowe ISBLE (Infini

Systems Bluetooth Low Energy). Do określenia pozycji ruchomego terminala komunikacyjnego wykorzystano metodę triangulacji. Wykorzystuje ona pomiar siły sygnału z kilku urządzeń transmisyjnych. Po określeniu siły sygnału wysłanego przez stację bazową ISBLE, odległość pomiędzy ISBLE a terminalem pozycjonującym można obliczyć zgodnie z modelem tłumienia sygnału Bluetooth. Gdy liczba odbieranych przez terminal sygnałów jest większa niż trzy, a współrzędne stacji bazowej ISBLE są znane, można oszacować współrzędne terminala pozycjonowania[5,6]. W niniejszym artykule zastosowano metodę najmniejszego kwadratu do rozwiązania współrzędnych terminala pozycjonowania. Metoda ta jest prosta i łatwa do zastosowania. Wpływ rozmieszczenia nadajnika ISBLE przeanalizowano pod kątem trzech aspektów: algorytmu lokalizacji, środowiska lokalizacyjnego i rozmieszczenia użytkowników[9].

### Metoda określania pozycji wewnątrz pomieszczeń oparta na wskaźniku siły sygnału RSSI

Technika pozycjonowania oparta na triangulacji jest szeroko stosowana w pozycjonowaniu osób, ruchomych maszyn oraz przedmiotów[7]. W metodzie tej należy znać położenie stałych punktów referencyjnych, które można przedstawić w płaszczyźnie jako zbiór koordynat  $(x_1, y_1)$ ,  $(x_2, y_2)$ , ...,  $(x_n, y_n)$ , oraz ich odległości od ruchomego badanego punktu, które mogą być wyznaczone przy użyciu mapowania wskaźnika RSSI na odległość,  $l_1, l_2, \dots, l_n$ . Przy założeniu, że współrzędne węzła docelowego to  $(x, y)$ , można przyjąć następujące funkcje zależności(1)[2]:

$$(1) \quad \begin{cases} l_1^2 = (x_1 - x)^2 + (y_1 - y)^2 \\ l_2^2 = (x_2 - x)^2 + (y_2 - y)^2 \\ l_n^2 = (x_n - x)^2 + (y_n - y)^2 \end{cases}$$

Gdy istnieją więcej niż 4 punkty referencyjne to do obliczenia optymalnych współrzędnych węzła docelowego wykorzystuje się metodę najmniejszych kwadratów i największego prawdopodobieństwa. Trudność pozycjonowania w takim układzie jest związana z tym, że lokalizację punktów referencyjnych trzeba znać wcześniej, zwłaszcza gdy punkty te ulegają zmianie i wymagana jest

duża liczba aktualizacji[3]. Techniki określania pozycji oparte na wskaźniku RSSI zależą od dokładnego oszacowania odległości za pomocą odwzorowania RSSI na odległość. Pomimo tego, że RSSI jest funkcją odległości i teoretycznie zapewnia wystarczającą dokładność pomiaru to na jakość ta wpływa otoczenie, rozkład pomieszczeń, osoby znajdujące się w pomieszczeniach jak również materiały, z których wykonane są elementy wyposażenia. Zazwyczaj wystarczający jest uproszczony model propagacji do pomiaru RSSI (wzór 2) i w którym odwzorowanie wartości RSSI na odległość została przedstawiona w zależności (2).

$$(2) \quad R = \Phi - 10S \log(l)$$

$$l = 10^{(R-\Phi)/(10 \cdot S)}$$

gdzie:

l jest odległością od aktualnej pozycji do określonego nadajnika,

R jest wartością RSSI w aktualnej pozycji,

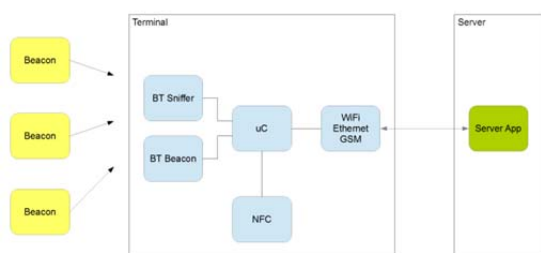
$\Phi$  jest wartością RSSI w pewnej założonej odległości referencyjnej (na przykład 1m),

S jest wykładnikiem strat w ścieżce.

Parametry  $\Phi$  oraz S określa się w celu dostosowania do różnych czujników i środowisk; wykładnik strat w ścieżce S ma zazwyczaj wartość w zakresie 1,6-1,8 w środowisku wewnętrznym. Co bardzo istotne, konieczne jest dopasowanie wykładnika straty w ścieżce w rzeczywistym środowisku. Parametry te powinny być ponownie wyznaczone, gdy węzeł docelowy przemieszcza się przez granicę dwóch różnych środowisk. Czasem istnieje również potrzeba ciągłej aktualizacji nawet wewnątrz tego samego środowiska.

### Podstawy specyfikacji systemu

Prace rozpoczęto od opracowania specyfikacji systemu. W tym przypadku wprowadzono kilka pojęć. Należą do nich układy Beacon (ISN) (nody pracujące w trybie lowpower) śledzone za pomocą terminali zwanych w dalszej części Terminalami (ISBLE) i analizowanych przez główny Serwer wyliczający dane (S). System jako całość ma za zadanie nasłuchiwać za pomocą Terminali ISBLE ramek rozgłoszeniowych emitowanych przez Beacons ISN. Ramki te zawierają identyfikator (UUID) na podstawie, którego identyfikowany jest dany beacon i określana jest jego odległość od Terminala ISBLE na podstawie parametru RSSI. Liczne Terminali o znanej pozycji w przestrzeni zbierają ramki rozgłoszeniowe od widocznych dla nich Beaconów i po dodaniu znacznika czasu odsyłają informację o Beaconie do Serwera. Na Serwerze następuje wstępne obliczenie pozycji. W kolejnym kroku następuje wyliczenie dokładnej pozycji po przetworzeniu przez algorytmy poprawiające dokładność lokalizacji. Wyniki w postaci nieprzetworzonej wędrują do aplikacji nadrzędnej gdzie następuje decyzja czy dane zostaną zapisane czy odrzucone.



Rys.1. Schemat systemu określania pozycji beaconów

Proces decyzyjny odbywa się na podstawie wymagań końcowego użytkownika. W celu przedstawienia założeń pracy system lokalizacji przedstawiono na rys. 1

### Beacony/Nody

Po wielu testach rozwiązanie zostało ostatecznie oparte na przetestowanych do tej pory układach BM70. Dzięki temu rozwiązaniu możliwa jest transmisja ramek rozgłoszeniowych z przy niskim poborze mocy. Ramki muszą zawierać unikalny UUID i opcjonalnie nazwę urządzenia.

### Zakres przeprowadzonych prac

Przeprowadzono badania w zakresie możliwości transmisji dodatkowych informacji zawierające dane w postaci temperatury, wilgotności, poziomu oświetlenia itp przez układ BM70. Podczas procesu określania lokalizacji przesyłanie dodatkowych informacji nie jest konieczne co znacznie redukuje ilość energii potrzebną do wysłania jednej ramki. W procesie ustalania lokalizacji wystarczy wysłać tylko ramki rozgłoszeniowe

Wyszukano inne rozwiązania niż zastosowane układy BM70. Szczególnie pożądanym jest konfigurowanie nazwy Noda i czasu rozgłaszania za pomocą bezprzewodowego interfejsu. Jeżeli podczas badań nie udało się zmienić parametrów związanych z nazwą i czasem rozgłoszeniowym to można założyć, że na etapie produkcyjnym te parametry są programowane na stałe.

Określono interwał czasowe, z jakim system musi emitować ramkę rozgłoszeniową by system jako całość pracował najbardziej efektywnie. Zbadano i wypracowano mechanizm i algorytm służący do definicji czasu dla konkretnych zastosowań

„Za często” powoduje zbyt duży ruch w eterze. Problem przy dużej ilości Nodów.

„Za często” generuje większe zużycie baterii

„Za rzadko” zmniejsza dokładność i czas odpowiedzi gdzie znajduje się Nod ISN

Zaprojektowano PCB oraz obudowy dla różnych wymagań systemowych takich jak:

- Największa dokładność pozycjonowania
- Najmniejszy rozmiar całego Node
- Wodoodporność jako opcja

Zaprojektowano programator konfigurujący urządzenie na etapie produkcyjnym. Stworzono Interfejs wewnętrzny szeregowy

Rozwiązano problem konfigurowania przez użytkownika. Najlepiej byłoby, aby proces konfiguracji mógł być wykonywany bezprzewodowo. Zbadano możliwość konfigurowania systemu poprzez interfejs Bluetooth. Rozważono i przetestowano transmisję NFC, wprawdzie podraża to całość, ale jak się okazało jest to dobre rozwiązanie. W przypadku problemów zaleca się by dostarczyć złącze zewnętrzne dostępne poprzez pogo pins lub podobne.

Przygotowano mechanizm filtracji RSSI dla danego Beacona (UUID). Ma to sens tylko wtedy, gdy ilość Beaconów jest niewielka. Rozwiązanie to generuje problemy związane z dynamicznym tworzeniem struktury filtrującej dla danego UUID i problem usuwania struktur już nieistniejących. Ostatecznie zdecydowano się na filtrację m.in filtrem Kalmana w wyższych warstwach systemu

### Terminal ISBLE

Zadaniem terminala jest nasłuch danych pochodzących od Beaconów, rozpoznanie w nich identyfikatora UUID, określenie mocy, dodanie znacznika czasowego TS (TimeStamp) i odesłanie Tych danych do Serwera. Zbiór w/w danych będzie dalej nazywane Ramką.

Zakłada się, że dane od Beaconów pojawiają się w czasie losowym. Konieczne jest, więc zastosowanie mechanizmu określającego, kiedy dana paczka danych pojawiła się w Terminalu. Dane te mają więc znacznik czasowy, a sam czas będzie ustawiany zdalnie tak, aby wszystkie Terminale miały ten sam czas z uwzględnieniem niedokładności zerowania. Terminal działa w oparciu o maksymalny czas opóźnienia, po jakim Terminal odsyła paczkę danych o Beaconach do serwera, zwany dalej Interwałem. Działa to w ten sposób, że w czasie 't' rozpoczyna się nasłuch danych. Dane te są wyposażane w TimeStamp a następnie kolejgowane w FIFO. Dane są gromadzone do momentu, gdy nie spełniony jest jeden z warunków:

- FIFO się przepełniło, czyli rozmiar FIFO przekroczył zakładany maksymalny rozmiar pakietu.
- Różnica znacznika czasowego dla najstarszego rekordu znajdującego się w FIFO jest większą lub równą interwałowi zdefiniowanemu przez użytkownika. Ten mechanizm ma za zadanie sprawić, że informacja o Beaconach nie będzie docierała z opóźnieniem do Serwera z większym niż zdefiniowany czas. Dzięki temu możliwe będzie śledzenie Beaconów w czasie rzeczywistym. Możliwe będzie ustalenie interwału równego 0, co skutkować będzie natychmiastowym wysłaniem Ramek z Terminala do Serwera.

Na rys. 2 przedstawiono postać przykładowej ramki.

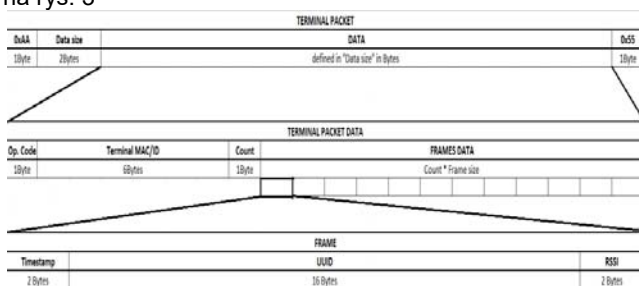
Timestamp	UUID	RSSI
2 Bytes	16 Bytes	2 Bytes

Rys. 2 Postać podstawowej Ramki (Frame)

Ramka zawiera następujące pola:

- 2 Byte - Timestamp
- 16 Byte – UUID
- 2 byte – RSSI

Sama Ramka (pojedyncza bądź zwielokrotniona w przypadku agregacji), uzupełniona o kod operacji (Op.Code) i adres MAC Terminala, będzie stanowić pole Frame Data pakietu wysyłanego przez Terminal do Serwera. Schemat agregacji ramek został przedstawiono na rys. 3



Rys. 3 Struktura agregacji ramek

Terminal Packet definiuje podstawowy pakiet danych przesyłany z terminala do serwera. Zawiera jednobajtowy znacznik początku pakietu (0xAA), rozmiar danych (Data Size - 2 Bajty), pole danych Data (o rozmiarze zdefiniowanym wcześniej, jako Data Size), oraz znacznik końca pakietu (0x55).

Podstawowy pakiet danych (Terminal Packet) w polu DATA zawiera kolejny rodzaj pakietu – na rys. 3 jest to pakiet nazwany „Terminal Packet Data”. Zawiera on kolejno: jednobajtowy kod operacji (Op.Code), dane specyficzne dla danego kodu operacji. W przykładzie na rys. 3 jest to pakiet zawierający zagregowane Ramki, poprzedzone adresem MAC Terminala oraz polem określającym ilość zagregowanych ramek. Jest to jedynie

opis specyficznego/przykładowego pakietu danych „Terminal Packet” zawierającego zagregowane Ramki. Terminal Packet jest używany jako uniwersalny pakiet do komunikacji pomiędzy Terminalem a serwerem. Szczegółowe jego implementacje są opisane poniżej.

Mechanizm transmisji ramek wprowadza pewną elastyczność systemu – przy małych wartościach interwału Serwer otrzymuje dane w bardzo krótkim czasie, natomiast większy interwał sprzyja agregacji ramek w większe pakiety. Widać, więc, że minimalny rozmiar pakietu zawierającego jedną ramkę to 32 bajty a maksymalny ogranicza rozmiar pola danych TCP/IP. Przeprowadzone badania wykazały, że bardziej korzystne jest użycie minimalnej wartości MTU (Maximum Transmission Unit) pakietu TCP/IP (zazwyczaj 1500 bajtów) pomniejszony o rozmiar nagłówek TCP i IP (łącznie 40 Bajtów). Używając najmniejszego rozmiaru MTU można uniknąć strat w przepustowości połączenia związanych z koniecznością fragmentacji pakietów w warstwach niższych. Zakładając maksymalny rozmiar payload pakietu TCP/IP jako 1460 Bajtów, jednocześnie można przesłać informację o 72 beaconach  $((1460 - 12) / 20 = 72)$ . Wartość 72 będzie stanowić maksymalną zalecaną wartość pola Count.

### Oszacowanie przepustowości systemu

Dane przesyłane są w postaci kolejnych rekordów. Co do przepustowości to można przyjąć, że w sytuacji maksymalnego nasilenia Beacon emituje rozgłoszenie, co 100ms. Można przyjąć, że jeden terminal odbiera dane z 100 Beaconów. Zakładając, że w systemie istnieje 100 Terminali a jedna ramka ma rozmiar 20 bajtów, do serwer trafia więc  $10 \times 100 \times 100 \times 20 = 2\text{MB}$  na sekundę, co nie stanowi większego wyzwania dla przepustowości całego systemu.

W wyniku prac ustalono, że konieczne jest:

Zaimplementowanie znacznika czasowego, dzięki któremu wszystkie Terminale będą miały ten sam TimeStamp. Mechanizm ten musi być na tyle dokładny by poprawnie parować dane z różnych terminali, ale nie aż tak by konieczne było wprowadzenie innych mechanizmów synchronizujących niż te dostępne przez interfejsy WiFi/Ethernet/GSM. Biorąc pod uwagę zakładaną wymaganą dokładność synchronizacji czasu, wystarczającym wydaje się być synchronizacja bazująca na protokole NTP wersji 4, zdefiniowana, jako RFC-5905. Przeprowadzone testy pokazały, że dokładność na poziomie 10ms jest wystarczająca. Ustawianie czasu musi odbywać się systematycznie w interwale określonym przez Serwer. Jednocześnie serwer w wersji testowej ma możliwość wymuszenia natychmiastowej synchronizacji czasu pojedynczego lub wszystkich Terminali. Terminal po nawiązaniu połączenia z serwerem będzie sygnalizował konieczność synchronizacji czasu Serwerowi i do czasu synchronizacji nie będzie wysyłał danych o Beaconach (Ramek), ponieważ błędne znaczniki czasu wpływałyby na błędne wyliczenie pozycji. Założono, że serwer będzie miał też możliwość konfiguracji zegara/czasu w Terminalu za pomocą odpowiedniego Rozkazu. Aby otrzymać zadowalającą dokładność synchronizacji czasowej pomiędzy Serwerem a Terminalami zaimplementowany został własny mechanizm synchronizacji będący częścią funkcjonalności Serwera oraz Terminali.

Określenie dokładności synchronizacji czasowej. Przy założeniu, że dokładność ustawienia czasu jest na poziomie 10ms wynika z tego, że podobny kwant czasu należy kodować w ramce przesyłanej z Terminala do Serwera. Zakładając, że znacznik czasu jest kodowany na 2 bajtach to przepełnienie licznika odbywa się 11 minutach. Jak pokazały testy rozmiar ten jest wystarczający.

Maksymalny czas „ważności” Ramki. Testy pokazały, że na potrzeby pozycjonowania szybko przemieszczających się elementów koniecznym może stać się wymaganie dokładniejszej synchronizacji czasu pomiędzy terminalami. W tej implementacji nie było to konieczne jednak w przyszłości należałoby rozważyć implementację protokołu PTP (Precise Time Protocol), zdefiniowanego, jako IEEE 1588-2002 pozwalającego na uzyskanie synchronizacji czasowej pomiędzy terminalami na poziomie 1 mikrosekundy.

Podczas badań ustalono kluczowe elementy systemu takie jak:

**Mikroprocesor.** Tutaj wybór padł na procesor ARM firmy STM, który może zostać zmieniony, gdy pozostałe interfejsy będą wymagały zastosowania systemu Linux. Obecne potencjalne problemy to brak dobrych układów gromadzących dane rozgłoszeniowe. Wówczas użyty zostanie nasłuch na bazie stosu Bluetooth Linuxa. Może również pojawić się konieczność szyfrowania i autoryzacji niewspieranej przez użyty moduł sieci Wifi/Ethernet/GSM

**BT Sniffer.** Założono, że należy wybrać długo wspierany moduł, który umożliwi nieprzerwane gromadzenie danych od Beacons. Jest to jednak problematyczne rozwiązanie, ponieważ niektóre moduły gromadzą jedynie n-pierwszych pakietów rozgłoszeniowych. Pewne z rozwiązań gromadzą dane w oparciu o okna czasowe definiowane przez zewnętrzny mikroprocesor. W badanym przypadku zdecydowano się aż na trzy własne moduły równoległe odbierające ramki rozgłoszeniowe z różnych kanałów

**BT Beacon.** Założono, że Terminal posiada na swoim pokładzie również Beacon. Ma on służyć do testowania sieci i jej kalibracji. Ze względu na stosunkowo statyczny charakter terminali czas emisji ramki rozgłoszeniowej Beacons może wynosić minimalnie jedną sekundę lub nawet rzadziej. Ostatecznie terminal będzie opcjonalnie posiadał Beacon ale w postaci standardowego modułu zasilanego z baterii i z zasilacza sieciowego

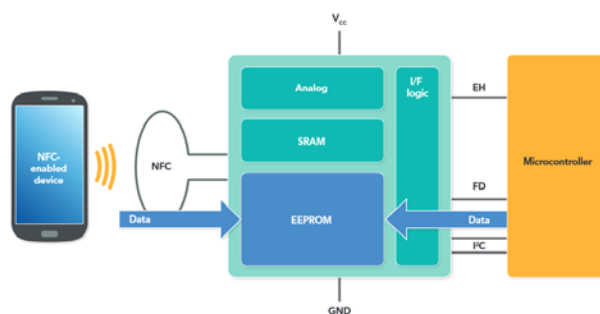
**Interfejs sieciowy.** Założono, że w zależności od wymagań będzie to albo Wifi, albo Ethernet albo GSM. Należy zaznaczyć, że dla rozwiązań krytycznych musi zostać użyty Ethernet. Ma on najmniejszy poślizg w transmisji danych oraz jest odporny na zakłócenie. W rozwiązaniu testowym ze względu na wygodę zaimplementowano transmisję Wifi.

**Interfejs NFC.** Po przeanalizowaniu różnych rozwiązań w końcowym rozwiązaniu zastosowany zostanie interfejs NFC. Dotyczy to seryjnej produkcji. Sam NFC jest modułem w postaci pamięci RAM lub pamięci nieulotnej. Założono, że opracowany zostanie mechanizm umożliwiający konfigurowanie Terminala oraz umożliwiający zmianę wewnętrznego oprogramowania Terminala. Na rys. 4 i rys. 5 przedstawiono przykładowe rozwiązania interfejsu NFC.

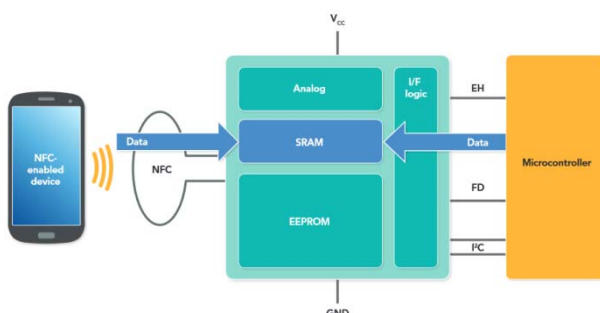
Zadaniem układu NFC w systemie jest:

- Konfigurowanie terminala w zakresie parametrów sieci. Reszta jest konfigurowana zdalnie.
- Zmiana oprogramowania wewnętrznego systemu. Przy założeniu, że proces może zostać przerwany zakłada się, że na stałe w uC znajduje się krótki kod bootloadera. Jego zadaniem na starcie lub w trakcie działania systemu (wątek sprawdzający pamięć NFC) jest testowanie zmian w pamięci NFC.

Jeśli zmiana zostanie wykryta, to układ wchodzi w tryb nadpisywania Firmware. Sprawdzana jest wersja Firmware. Następnie wybierana jest jedna z dwóch połówek pamięci uC która zawiera starszy lub nieaktualny Firmware, lub półka która obecnie nie pracuje. Następuje nadpisywanie są etapami kolejne bloki pamięci Flash.



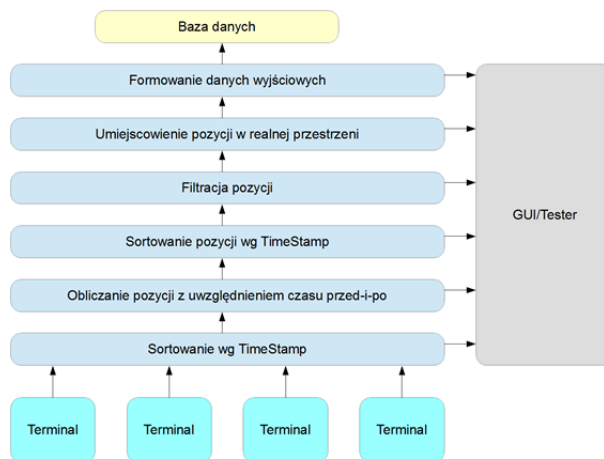
Rys. 4 Interfejs NFC wyposażony w pamięć nieulotną



Rys. 5 Interfejs NFC wyposażony w statyczną pamięć RAM

### Serwer

Serwer to aplikacja, do której spływają wszystkie ramki od wszystkich terminali. Założono, że dla każdego Terminala czas od emisji przez niego ramki a czas, kiedy dociera ona do Serwera jest określony i nie jest większy niż czas maksymalny. Jest to o tyle ważne, ponieważ przetwarzanie w czasie rzeczywistym wymaga założenia w danym czasie analizy wszystkich ramek z danego przedziału czasowego. Cały stos aplikacji Serwera ma postać którą upraszczając można przedstawić w postaci przedstawionej na rys. 6



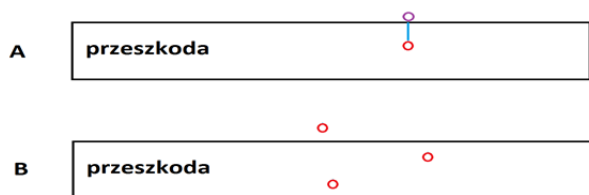
Rys. 6 Uproszczony schemat stosu aplikacji Serwera

Dane do Serwera przychodzą w postaci ramek od poszczególnych Terminali. W danej ramce dane są poszeregowane wg znaczników czasu. Ramki od różnych terminali mogą przyjść w nieco różnym czasie.

### Dodatkowe zagadnienia

Projektując system należało uwzględnić wiele sytuacji które muszą zostać rozpisane na procedury, schematy działania i algorytmy. Przykładowo poniżej przedstawiono kilka z nich. Bardzo istotne przy zaokrągleniu pozycji beaconów jest odniesienie pozycji do stref 'przeszkoda i

'poza'. W zależności przyjętej polityki (parametr dla odpowiedniej warstwy aplikacji na serwerze) konieczne jest zaimplementowanie modyfikacji pozycji w przypadku gdy wyszło, że obiekt znajduje się przykładowo w 'ścianie' czyli 'przeszkodzie'. Na rys. 7 przedstawiono przykładowe sytuacje, w których system określił pozycję nodów ISN



Rys. 7 Przykładowy układ nodów ISN w pomieszczeniu z uwzględnieniem sytuacji, w której nody zostaną wykryte wewnątrz przeszkody

W przypadku sytuacji A w zależności od parametru nowa pozycja będzie wyliczona do miejsca najbliższego obliczonej pozycji. Najbliżej w sensie poza obszarem przeszkody. Inne podejście, które można zastosować to uwzględnienie pozycji w obrębie czasu przed-i-po gdzie brane są pozycje poprzednie i przyszłe. Wynik uśredniany i ponownie rzutowany na obszar możliwy, gdy naruszamy obszar niemożliwy. W warstwie tej wykonuje się filtrację położenia przy założeniu, że położenie w warstwach wcześniejszych jest włączone lub wyłączone. Wychwycone podstawowe problemy systemu to nakładanie się ramek Beaconów. Terminal nic nie odczytuje, kiedy dwa beacony dokładnie w tym samym momencie emitują sygnał. Jako wniosek po przeprowadzeniu testów pojawiło się stwierdzenie, że należy określić czas co, ile Beacon może transmitować ramkę rozgłoszeniową by zminimalizować ten efekt. Im mniej ramek na sekundę tym mniejsze ryzyko kolizji. Po przeprowadzeniu badań wykazano, że pojawia się zmienna siła sygnału RSSI nawet przy założeniu, że geometria usytuowania Beacon-Stacja nie ulega zmianie. Konieczna jest filtracja w postaci przynajmniej uśredniania. Inne podejście to filtracja, która wyznacza trend sygnału. Brak wystarczającej ilości ramek dla danego Beacona. Przykładowo wykryto tylko na jednym terminalu jedno lub dwa identyfikatory UUID przez co nie da obliczyć się pozycji. Zbyt duże zagęszczenie Beaconów. Ten problem akurat nie generuje kolizji o ile Beacony nie emitują danych w tym samym momencie. Problem opóźnień w transmisji Wifi. Obawiano się, że Wifi będzie miało zbyt duży czas od otrzymania danych do pojawienia się ich na serwerze. Przeprowadzone testy pokazały jednak, że ten problem jest stosunkowo mało istotny. Nieodporność na zakłócenia BT i Wifi. W tym przypadku zdiagnozowano sytuacje, kiedy sprzęt komputerowy całkowicie zakłócał odbiór danych przez terminal.

Zabezpieczenie systemu przed „sprytnymi pracownikami”. Wyższa warstwa, która jak pokazały testy musi zostać rozwiązana na poziomie aplikacji zarządzającej danymi otrzymywanymi od serwera.

## Podsumowanie

System zaprojektowano jako rozwiązanie, które musi odbierać dane od Beaconów które emitują minimalną ilość danych w niewielkich przedziałach czasowych przez co pobierają minimalną ilość energii co ma kluczowe znaczenie w przypadku zasilania bateryjnego. Terminale, których pozycja jest znana w określonej przestrzeni, obierają ramki rozgłoszeniowe od Beaconów znając jednocześnie ich siłę sygnału. Mając te dane i dokładną pozycję terminali można wyznaczyć na serwerze pozycję Beaconów w środowisku zamkniętym. System zaprojektowano i przetestowano. Jak pokazały liczne testy w systemie istnieje jeszcze wiele miejsc, gdzie można poprawić dokładność pozycjonowania. Nie zmienia to jednak faktu, że takie rozwiązanie zwiększa przynajmniej o 30 % żywotność systemu zasilania poszczególnych Beaconów.

**Autorzy:** mgr. Jacek Bargiel, Infini Systems sp. z o.o, aleja Różdzieńskiego 188c, 40-203 Katowice, E-mail: biuro@infinisys.pl; mgr Krzysztof Fijak, E-mail: fijaklogic@gmail.com.

## LITERATURA

- [1] Ali W., Kareem, A., Jasim M., Survey on Wireless Indoor Positioning Systems. CUESJ 2019, No 3, 42.
- [2] Lee1 K., Nam1 Y., An indoor localization solution using Bluetooth RSSI and multiple sensors on a smartphone. Multimed. Tools Appl. 2018, No 77, 12635
- [3] Baala O., Zheng Y., The Impact of AP Placement in WLAN based Indoor Positioning System. Proceedings of the Eighth International Conference on Networks 2009, 12
- [4] Hara K., Shinsuke W., Fukumura T., Determination of the placement of anchor nodes satisfying a required localization accuracy. IEEE International Symposium on Wireless Communication Systems 2008, 128
- [5] Brena R., García-Vázquez J., Galván-Tejada C., Evolution of Indoor Positioning Technologies. Surv. J. Sens. 2017, 2630413
- [6] Huang H., Gartner G., Krisp J., Location based services: Ongoing evolution and research agenda. J. Locat. Based Serv. 2018, No 12, 63
- [7] Styła M., Adamkiewicz P.: Optimization of building control systems with use of alternative tracking methods using the RSSI index and artificial intelligence mechanisms, Przegląd Elektrotechniczny, 98(2), 138-202,2022
- [8] Ito K., Xiong K., Gaussian filters for nonlinear filtering problems. IEEE Trans. Autom. Control 2020, No 45, 910
- [9] Dulmage J., Characterization of Distance Error with Received Signal Strength Ranging. Wireless Communications and NETWORKING Conference, 2010, 1-6