

POLITECHNIKA WROCŁAWSKA  
WYDZIAŁ ELEKTRONIKI

---

KIERUNEK: Elektronika

SPECJALNOŚĆ: Aparatura Elektroniczna

PRACA DYPLOMOWA  
INŻYNIERSKA

System bluetooth do sterowania urządzeniami  
w inteligentnym domu.

Bluetooth system for smart home control.

AUTOR:

Grzegorz Korzeniewski

PROWADZĄCY PRACĘ:

Dr inż. Grzegorz Świrniak

Katedra Metrologii Elektronicznej i Fotonicznej

OCENA PRACY:

# Spis treści

<b>WPROWADZENIE .....</b>	<b>3</b>
<b>1. INTELIGENTNY DOM – PODSTAWY .....</b>	<b>5</b>
<b>2. ANALIZA BEZPRZEWODOWYCH INTERFEJSÓW CYFROWYCH .....</b>	<b>6</b>
2.1. Wi-Fi .....	8
2.2. ZIGBEE .....	10
2.3. BLUETOOTH.....	11
2.3.1. <i>Bluetooth Low Energy</i> .....	14
2.3.2. <i>Beacon</i> .....	16
<b>3. SYSTEM BLUETOOTH DO STEROWANIA URZĄDZENIAMI W INTELIGENTNYM DOMU.....</b>	<b>18</b>
3.1. MODUŁ BLUETOOTH .....	19
3.2. STEROWNIK.....	21
3.2.1. <i>Mikrokontroler</i> .....	22
3.2.2. <i>Wyświetlacz</i> .....	23
3.2.3. <i>Joystick i klawisz</i> .....	24
3.2.4. <i>Interfejs</i> .....	24
3.2.5. <i>Konfiguracja modułu bluetooth</i> .....	25
3.2.6. <i>Oprogramowanie sterujące</i> .....	25
3.2.7. <i>Zasilanie</i> .....	28
3.3. MODUŁ POMIARU TEMPERATURY .....	28
3.3.1. <i>Mikrokontroler</i> .....	30
3.3.2. <i>Czujnik temperatury</i> .....	31
3.3.3. <i>Konfiguracja beacons</i> .....	31
3.3.4. <i>Zasilanie i zapotrzebowanie prądowe</i> .....	33
3.4. MODUŁ STEROWANIA OŚWIETLENIEM.....	34
3.4.1. <i>Mikrokontroler</i> .....	36
3.4.2. <i>Przełącznik</i> .....	36
3.4.3. <i>Konfiguracja modułu HM-10</i> .....	37
3.4.4. <i>Zasilanie</i> .....	38
3.5. ANALIZA WSPÓŁPRACY SYSTEMU Z URZĄDZENIAMI DZIAŁAJĄCYMI POD KONTROLĄ SYSTEMU OPERACYJNEGO ANDROID .	38
<b>PODSUMOWANIE.....</b>	<b>40</b>
<b>BIBLIOGRAFIA .....</b>	<b>42</b>

# Wprowadzenie

Termin inteligentnego budynku pojawił się w latach 60 tych. W tamtym okresie zaczęto wprowadzać do budynków mieszkalnych nowe, na ówczesne czasy, technologie budynkowe, takie jak ogrzewanie i klimatyzacja. Dekadę później wdrażano już rozwiązania z działu automatyki i telekomunikacji. Rozwój wspomnianych technologii spowodował, iż podczas projektowania mieszkań zaczęto zwracać uwagę na zadowolenie użytkownika. W związku z tym od lat 80 tych pojęcie budynku inteligentnego rozpatruje się jako środowisko życia człowieka, które rozpoznaje jego potrzeby i na nie odpowiada inteligentnie [1].

Domy są automatyzowane w celu polepszenia komfortu zamieszkiwania, efektywnego wykorzystywania energii (np. ciepłej lub elektrycznej), zminimalizowania emisji szkodliwych zanieczyszczeń, zwiększenia bezpieczeństwa oraz maksymalizacji funkcjonalności. Możliwości są nieograniczone. Z każdym dniem pojawiają się coraz to nowsze pomysły na rozbudowanie idei inteligentnego domu [1].

Szczególne zastosowanie w systemach inteligentnego domu znajdują technologie komunikacji bezprzewodowej. Bezprzewodowy system jest o wiele łatwiej zaadaptować do istniejącego już budynku. W przypadku przewodowego odpowiednika instalacja systemu wymaga ingerencji w infrastrukturę ścian lub okablowanie musi być wzięte pod uwagę już w fazie projektowania domu. Bezprzewodowa łączność również znacząco zwiększa wygodę użytkowania systemu [2].

**Celem pracy** jest opracowanie koncepcji i projektu oraz realizacja podstawowej funkcjonalności systemu służącego do sterowania wybranymi urządzeniami w inteligentnym domu za pośrednictwem interfejsu bluetooth.

**Zadania** jakie wyznaczono, aby osiągnąć cel pracy przedstawiają się następująco:

- Analiza i systematyka bezprzewodowych interfejsów cyfrowych służących do wymiany danych w inteligentnym domu.
- Analiza możliwości współpracy systemu z urządzeniami mobilnymi pracującymi pod kontrolą systemu Android.
- Opracowanie koncepcji, projektu i realizacja podstawowej funkcjonalności systemu.
- Przetestowanie systemu.

Część pierwsza pracy poświęcona została podstawowym ideom inteligentnego domu oraz analizie bezprzewodowych interfejsów jakimi są technologie Wi-Fi, Zigbee oraz Bluetooth. W przypadku technologii Bluetooth szczegółowo opisano najnowszą wersję 4.0 Low Energy oraz technologię beacon. Zbadana również została możliwość współpracy opracowanego systemu z urządzeniami wspierającymi system Android.

W części drugiej skupiono się na opracowaniu koncepcji projektu systemu. Zawiera ona schematy ideowe oraz schematy blokowe przedstawiające systematykę działa systemu. Opisano wykorzystany do komunikacji moduł bluetooth oraz przedstawiono poszczególne części systemu (jednostka sterująca oraz moduły wykonawcze i sensory). W części tej umieszczono również testy systemu.

W pracy zawarto metodykę czynności odpowiadającą regułom postępowania inżynierskiego, opierając się na sformułowanej **tezie**:

**Opracowanie systemu opartego na komunikacji bezprzewodowej wykorzystując technologię Bluetooth umożliwi sterowanie urządzeniami w inteligentnym domu.**

# 1. Inteligentny dom – podstawy

Inteligentny dom jest wyposażony w zestaw czujników, detektorów oraz jeden zintegrowany system zarządzania znajdującymi się w budynku instalacjami. Dzięki cyklicznym pobieraniu danych pochodzącym z różnych elementów systemu, budynek może zaadaptować się do zmian w środowisku wewnątrz i na zewnątrz, co prowadzi do maksymalizacji funkcjonalności, komfortu i bezpieczeństwa, minimalizacji kosztów eksploatacji i modernizacji oraz ograniczenia emisji szkodliwych zanieczyszczeń [1].

Przykładem jest utrzymywanie zadanej temperatury dla wybranego pomieszczenia która może się różnić od pożądanej temperatury innego pomieszczenia. Ponadto, system może wykryć, czy domownik zasnął (np. na podstawie czujników ruchu) i odpowiednio na tą sytuację zareagować, gasząc światło, wyłączając telewizję i blokując drzwi domu.

Głównymi zadaniami inteligentnego domu i krytycznymi aspektami systemu są [2]:

- Łączność – system musi sprawnie komunikować się z podłączonymi urządzeniami, borykając się z takimi problemami jak m.in. opóźnienia transmisji, fizyczne przeszkody (ściany, schody, meble itp.), przepustowość łączności.
- Unifikacja protokołu – urządzenia w domu są zróżnicowane i mają różny stopień skomplikowania. Obsługa prostej lampy znacznie się różni od skomplikowanej łączności np. z serwerem. Przed projektantem systemu stoi zadanie skonstruowania protokołu komunikacji optymalnego dla urządzeń każdego rodzaju.
- Dynamiczność – System musi adaptować się do zmieniających się parametrów infrastruktury. Użytkownik systemu może w dowolnej chwili zażądać włączenia danego urządzenia lub mierzone wielkości mogą ulec zmianie.
- Kompatybilność urządzeń różnych producentów - decyduje to o tym czy dane urządzenie będzie obsługiwane przez system. W przypadku, gdy producenci modułów dochodzą do porozumienia w sprawie standaryzacji swoich produktów, możliwa jest współpraca podzespołów różnych producentów w jednym systemie. W przeciwnym przypadku praca systemu jest niemożliwa
- Skalowalność – system musi być tak zaprojektowany, aby mógł działać zarówno z małą ilością połączonych urządzeń (np. 3) jak i znacznie większą ilością modułów. Projektant systemu powinien określić maksymalną ilość obsługiwanych w systemie urządzeń

- Bezpieczeństwo – krytyczny aspekt w przypadku danych osobistych bądź kontroli dostępu do domu.
- Auto-Adaptacja – adaptacja do zmian w otoczeniu systemu, takim jak np. różne pory roku czy zmienne nastroje domowników. System powinien być zdolny do wprowadzania zmian w sterowaniu i do wprowadzania nowych scenariuszy reakcji urządzeń.
- Prostota użytku – decyduje o zadowoleniu domowników z użytkowania systemu. Obsługa powinna być prosta, intuicyjna, niewymagająca zbyt dużo uwagi od użytkownika.

Komercyjnym przykładem systemu inteligentnego domu jest np. system Fibaro, KNX, x10.

Technologie bezprzewodowe są powszechnie wykorzystywane w systemach inteligentnych domów. Usprawniają działanie systemu oraz polepszają komfort użytkowania. Ponadto instalacja bezprzewodowego systemu w istniejącym już budynku jest prostsza i mniej kosztowna niż adaptacja budynku do przewodowej wersji systemu [2].

## **2. Analiza bezprzewodowych interfejsów cyfrowych**

Bezprzewodowa łączność to każda forma wymiany informacji, która odbywa się bez połączeń za pomocą kabli. Dane mogą być wymieniane między dwoma lub kilkoma punktami, które nie są fizycznie połączone. Odległość między końcami transmisji może być zupełnie mała (kilka centymetrów) lub nadawanie może się odbywać na przestrzeni kilkuset kilometrów. W obecnych czasach łączność bezprzewodowa jest obecna w każdej dziedzinie życia (radio, telewizja, systemy GPS, telefonia komórkowa, Internet itp.) [3].

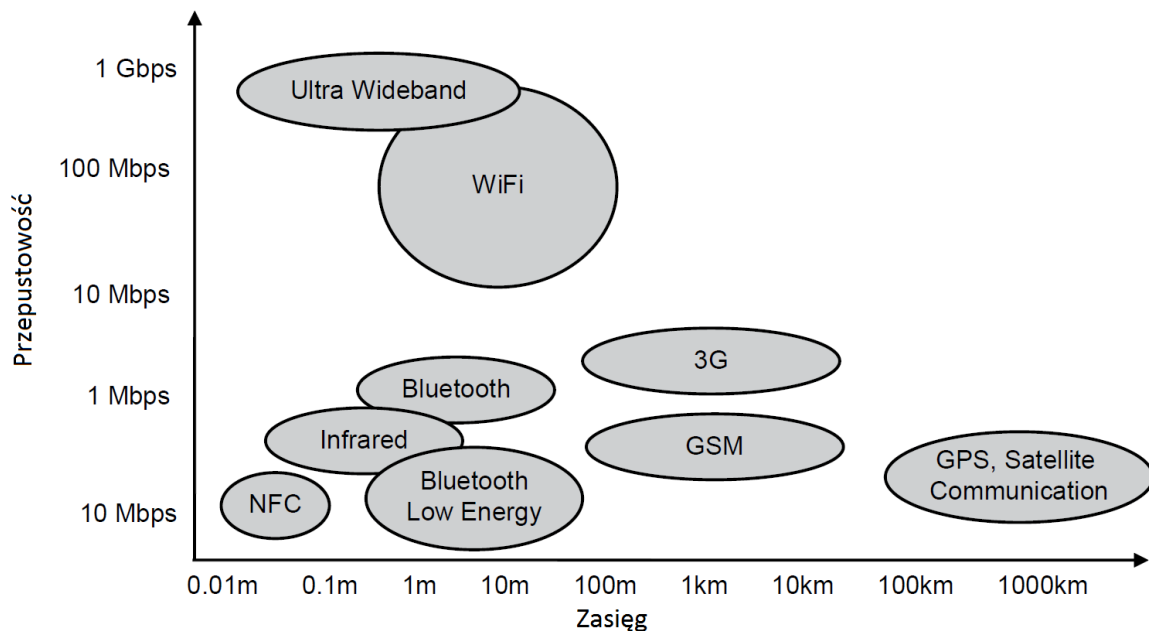
Głównymi zaletami łączności bezprzewodowej są

- Mobilność, czyli urządzenia można dowolnie przenosić nie zważając na długość kabli czy niewygodne „uwiązanie” połączeniami kablowymi,
- Zmniejszony koszt, czyli brak kosztownej infrastruktury kabli,
- Zwiększona produktywność, czyli możliwość pracy będąc w ruchu. Przykładem jest zdalna praca przemieszczając się np. pociągiem.

Typy sieci można podzielić na kilka rodzajów ze względu na obszar który pokrywają [3]:

- WAN (ang. *Wide Area Network*), sieć ta ma zasięg od kilku do kilku tysięcy kilometrów. Przykładem są technologie GSM (*Global System for Mobile Communications*), GPRS (*General Packet Radio Service*).
- MAN (*Metropolitan Area Network*), pokrywa obszar mniejszy niż WAN. Działa w obrębie miasta bądź metropolii. Przykładem technologii jest WiMAX (*Worldwide Interoperability for Microwave Access*) która operuje w zasięgu kilku kilometrów.
- LAN (*Local Area Network*), zasięg do kilkuset metrów. Zazwyczaj jest to sieć w budynkach, kampusach, osiedlach. Jest to np. technologia Wi-Fi.
- PAN (*Personal Area Network*) sieć łącząca urządzenia użytkowe typu klawiatura, smartfon, tablet, drukarka, znajdujące się w zasięgu kilku metrów. Przykładowe technologie: IrDa, Bluetooth, USB, NFC, Zigbee.
- BAN (*Body Area Network*) komunikacja między urządzeniami który użytkownik nosi ze sobą, czyli np. smartfon, zegarek, czujnik pracy serca. Technologie: Bluetooth, NFC, RFID, ANT.

Na rys. 1 przedstawiono popularne technologie komunikacji bezprzewodowej w zależności od ich przepustowości i ich teoretycznego zasięgu.



Rysunek 1 Zasięg i przepustowość popularnych technologii bezprzewodowych [3].

## 2.1.Wi-Fi

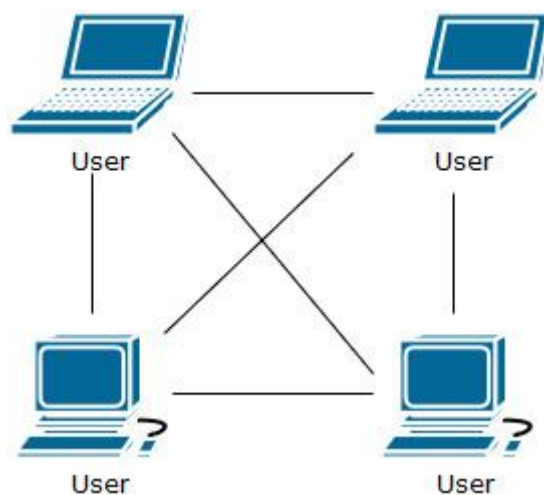
Wi-Fi, czyli „Wireless Fidelity” to technologia komunikacji bezprzewodowej służąca do budowania sieci lokalnych WLAN (ang. *Wireles Local Area Network*). Wi-Fi to znak towarowy którego właścicielem jest stowarzyszenie Wi-Fi Alliance, które również dba o promowanie technologii i przestrzegania wzajemnej zgodności urządzeń standardu [5].

Wi-Fi oparte jest na standardzie 802.11x. Sieć Wifi operuje w paśmie częstotliwości od 2400 do 2485 MHz (technologia 2,4 GHz) lub 4915 do 5825 MHz (5 GHz). Główne wersje standardu 802.11 to [4]:

- 802.11a – do 54 Mb/s, częstotliwość 5 GHz.
- 802.11b – 11 Mb/s częstotliwość 2,4 GHz ma zasięg ok. 30 m w pomieszczeniu i 120 m w otwartej przestrzeni; w praktyce można osiągnąć transfery rzędu 5,5 Mb/s. Standard 802.11b podzielony jest na 14 kanałów o szerokości 22 MHz które częściowo się pokrywają.
- 802.11g – 54 Mb/s częstotliwość 2,4 GHz,
- 802.11n – 300 Mb/s częstotliwość 5 GHz oraz 150Mb/s w częstotliwości 2,4 GHz,
- 802.11ac – do 1 Gb/s.

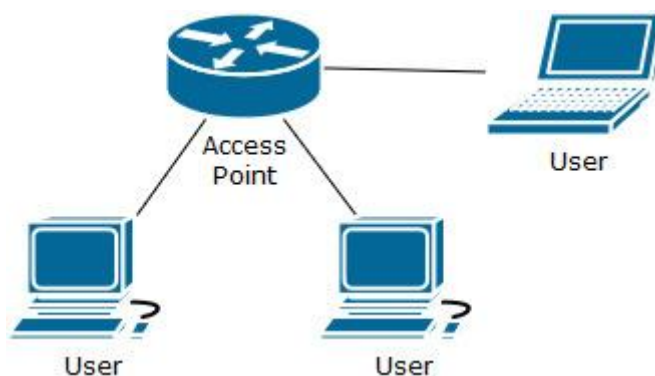
Sieć w standardzie może być zorganizowana na różne sposoby. Jednym z nich jest sieć typu IBSS (ang. *Independence Basic Service Set*) czyli sieć niezależna umożliwiająca połączenie bezpośrednio kilku urządzeń [4]. Sieci IBBS tworzone są spontanicznie, dlatego też nazywa się je *ad-hoc*. W sieci typu ad-hoc każde z urządzeń może nawiązać bezpośrednie połączenie z dowolnym aktywnym terminalem znajdującym się w zasięgu lub poprzez terminal innego klienta, o ile udostępnione są jego zasoby. Standard nie przewiduje żadnych ograniczeń liczby urządzeń w sieci. Topologię sieci przedstawiono na rys. 2.





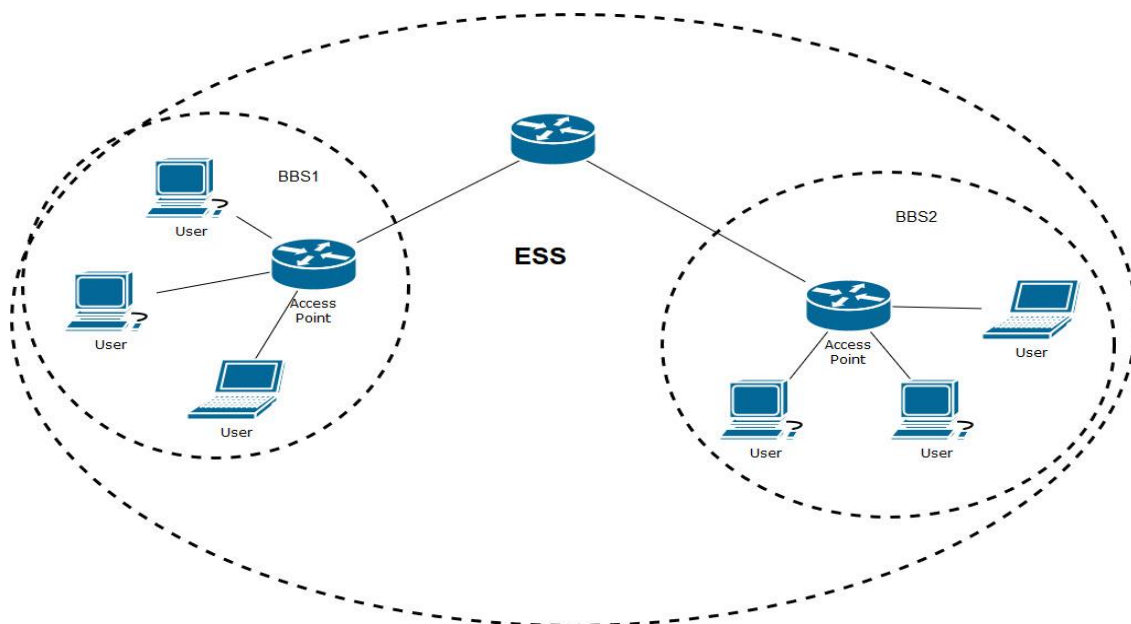
Rysunek 2 Sieć typu IBSS.

Inny rodzaj sieci standardu 802.11x to BSS (ang. *Basic Service Set*). Jest to sieć połączonych ze sobą urządzeń za pomocą punktu dostępowego Access Point. Jest on centralnym punktem komunikacyjnym dla wszystkich urządzeń należących do sieci BSS. Inna nazwa sieci to sieć punkt-wielopunkt [4]. Przykładową sieć typu BSS pokazano na rys. 3.



Rysunek 3 Sieć typu BSS.

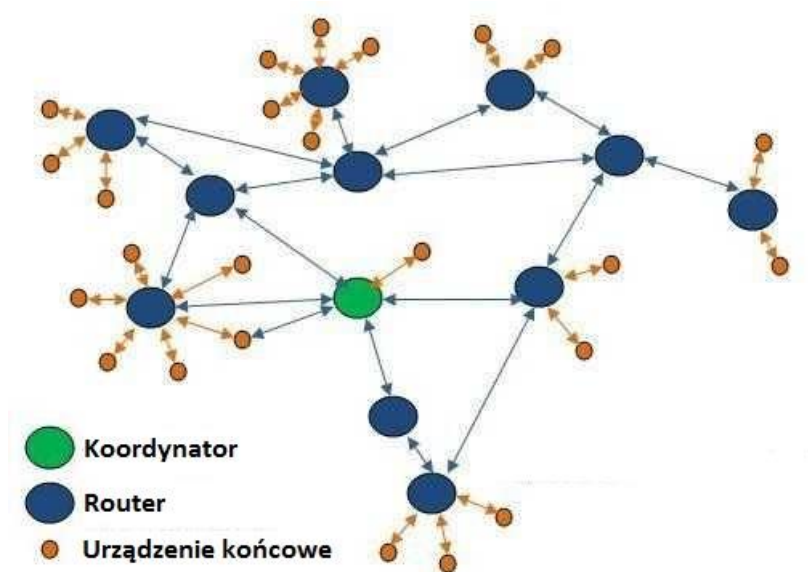
Sieci typu ESS (ang. *Extended Service Set*) wynika z połączenia ze sobą kilku sieci BSS za pomocą sieci szkieletowej, Punkty dostępowe w pojedynczym ESS pełnią rolę mostów radiowych. Sieć ESS zapewnia większy zasięg poprzez występowanie wielu punktów dostępowych [4]. Idea sieci ESS przybliżona została na rys. 4.



Rysunek 4 Sieć typu ESS.

## 2.2. Zigbee

Zigbee to technologia łączności bezprzewodowej opisana w standardzie IEEE 802.15.4. Protokół pozwala na tworzenie sieci personalizowanych WPAN (ang. *Wireless Personal Area Network*), sieci sensorowych, z zastosowaniem w automatyce domowej, systemach monitoringu czy systemach alarmowych. Sieci w technologii ZigBee cechują się zasięgiem do 100 m, małym poborem prądu i niską przepustowością (do 250 kb/s) [6]. Topologię sieci zobrazowano na rys 5.



Rysunek 5 Przykład sieci Zigbee [7].

Technologia operuje w pasmach częstotliwościowych 2.4 GHz, 900 MHz i 868 MHz. Protokół wspiera sieci typu *Mesh*, czyli każdy węzeł sieci może być połączony z każdym innym węzłem danej sieci. Łączność jest zabezpieczona 128-bitowym symetrycznym kluczem szyfrującym.

Protokół pakietów danych w standardzie Zigbee jest względnie prosty i odporny na błędy komunikacji. Prostotę protokołu potwierdzają niskie wymagania stawiane mikrokontrolerom które omawiany standard implementują. Implementacja stosu Zigbee zajmuje jedynie ćwierć pamięci Flash przeznaczonej na potrzeby stosu protokołu bluetooth [5].

Są trzy rodzaje urządzeń partycypujących w systemie ZigBee:

- Koordynator - zachowuje się jako punkt centralny w sieci. Zbiera informacje od podłączonych do niego urządzeń. Tylko jedno takie urządzenie może się znaleźć w sieci.
- Router - funkcjonalnością przypomina router sieciowy znany ze standardu 802.11. Przekazuje pakiety danych oraz zwiększa zasięg sieci, tworząc ścieżki z tzw. „przeskokami” (ang. *multihop routing*).
- Urządzenie końcowe - przesyła dane do koordynatora (również za pomocą routera). W celu minimalizacji poboru energii urządzenia końcowe większość czasu pracują w trybie uśpienia.

Liczba urządzeń końcowych w sieciach ZigBee teoretycznie może wynosić 65536 (16 bitów adresu). Zazwyczaj liczba ta wynosi maksymalnie 256 [5].

### **2.3. Bluetooth**

Bluetooth to standard komunikacji bezprzewodowej krótkiego zasięgu. Jest to technologia ciesząca się wielką popularnością, wykorzystywana do wymiany plików między komputerami i smartfonami, słuchania muzyki bezprzewodowo, drukowania dokumentów, bezprzewodowych zestawów głośnomówiących i wielu więcej aplikacji. Aktualnie zdecydowana większość urządzeń mobilnych jest wyposażona w moduły bluetooth.

Początki standardu datuje się na rok 1994, kiedy to uformowano grupę SIG (ang. *Special Interest Group*) składającą się z firm L.M. Ericsson, Nokia, Intel, Toshiba i IBM w celu standaryzacji bezprzewodowego standardu niewielkiego zasięgu, o małym poborze prądu, z niskim poziomem mocy promieniowanej i o niskiej cenie. Pierwotnym zamierzeniem technologii było zastąpienie szeregowych interfejsów przewodowych łączących różne

urządzenia. Koncepcję poszerzono o obszar bezprzewodowych sieci LAN, dzięki czemu standard stał się bardziej praktyczny i stał się konkurencją dla standardu 802.11 [3].

Moc nadawcza podzielona jest na 3 klasy: [3]

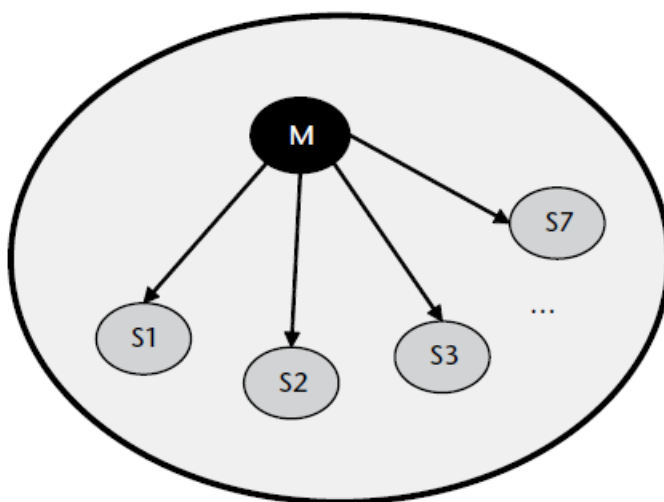
- klasa 1 - 100 mW - zasięg teoretyczny do 100 m
- klasa 2 - 2,5 mW - teoretyczny zasięg do 10 m, najczęściej stosowana
- klasa 3 - 1 mW - teoretyczny zasięg do 1 m, używana rzadko

Przez lata opracowano wiele wersji technologii: [3]

- Bluetooth 1.0 – przepustowość 21 kb/s
- Bluetooth 1.1 – przepustowość 124 kb/s
- Bluetooth 1.2 – przepustowość 328 kb/s
- Bluetooth 2.0 EDR (*Enhanced Data Rate*) – zwiększono transfer teoretyczny do 2,1 Mb/s
- Bluetooth 2.1 EDR - uproszczenie i ujednolicenie procesu parowania urządzeń BT, wsparcie dla przyszłych implementacji technologii NFC, zmniejszenie zużycia energii
- Bluetooth 3.0 HS (*High Speed*) – transfer 24 Mb/s
- Bluetooth 3.1 HS – transfer 40 Mb/s
- Bluetooth 4.0 LE (*Low Energy*) – 1 Mb/s znacząco ograniczono pobór energii (np. praca czujnika temperatury przez wiele miesięcy na baterii pastylkowej) kosztem obniżonego transferu, zwiększono realny zasięg działania do 100 m (szerzej opisane w rozdziale 2.3.1.)
- Bluetooth 4.1 - standard opracowany do zastosowania w tzw. "Internecie rzeczy" umożliwiający bezpośrednią łączność urządzeń z Internetem
- Bluetooth 4.2 - szybszy transfer, ułatwione nawiązanie łączności, bezpieczniejsze,
- Bluetooth 5.0 - ujednolicenie wersji, szybszy transfer – 2 Mb/s dla urządzeń typu w Internecie rzeczy i 50 Mb/s do zwykłych, realny zasięg działania do 140m

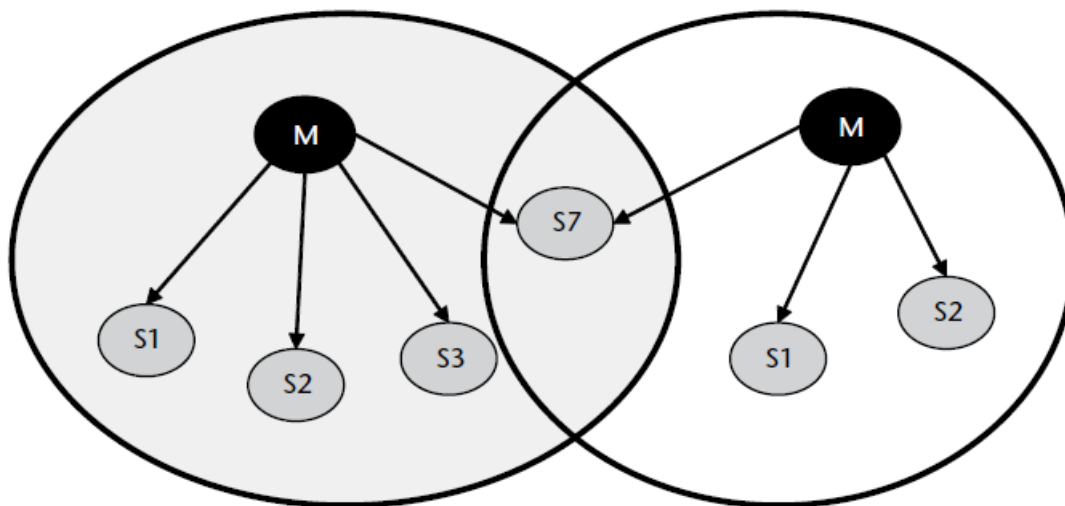
W architekturze Bluetooth główną jednostką jest pikosieć. Jest to rodzaj sieci tworzonej spontanicznie, nazywane również *ad-hoc*. Do utworzenia pikosieci wystarczy połączenie dwóch urządzeń, z których jedno pełni rolę *Master*, a drugie jest typu *Slave*. Rola urządzenia Master polega na inicjowaniu połączeń, podawaniu zegara taktującego i wybraniu kanału transmisji. Urządzenie typu slave przyporządkowywane jest masterowi w pikosieci.

W pikosieci może istnieć tylko jeden Master oraz do siedmiu urządzeń slave. [3] Przykładowa sieć ukazana została na rys 6.



Rysunek 6 Pikosieć. M-Master, S- Slave [3].

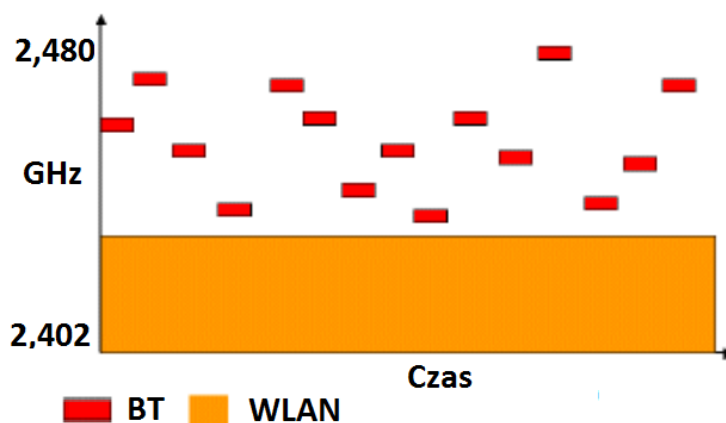
Metodą na rozszerzenie pikosieci jest technika *Scatternet* (zobrazowana na rys. 7), czyli łączenie kilku pikosieci między sobą urządzeniem typu *bridge*. Pozwala to na wzajemną komunikację większej ilości urządzeń.



Rysunek 7 Technika scatternet (nieobsługiwane w wersji Low Energy) [3].

Bluetooth działa w paśmie radiowym 2.400 - 2.4835 GHz. W początkowych wersjach bluetooth pasmo to było podzielone na 79 kanałów o szerokości 1 MHz zaczynając od częstotliwości 2402 MHz. [3]

Pasmo 2,4GHz jest wykorzystywane przez wiele technologii takich jak Wi-Fi lub NFC, czy nawet przez sterowane radiowo samochody-zabawki. W celu przeciwdziałaniu zaszumiania transmisji Bluetooth korzysta się z techniki *Adaptive Frequency Hopping*, czyli adaptacyjne przełączanie częstotliwości. Technika ta polega na przełączaniu się między kanałami z prędkością 1600 skoków na sekundę. Podczas nawiązywania łączności urządzenie master i slave „szukają” się nawzajem losowo zmieniając kanały. Gdy natrafią na ten sam kanał, master dyktuje na jaki kanał należy się przełączyć by kontynuować transmisję. Pozwala to na unikanie częstotliwości już zajmowanych przez inne urządzenia. Na rys. 8 zobrazowano działanie techniki w praktyce. Kanały zajęte przez sieć WLAN (kolor żółty) unikane są przez kanały Bluetooth (kolor czerwony) za pomocą szybkich zmian częstotliwości [8].



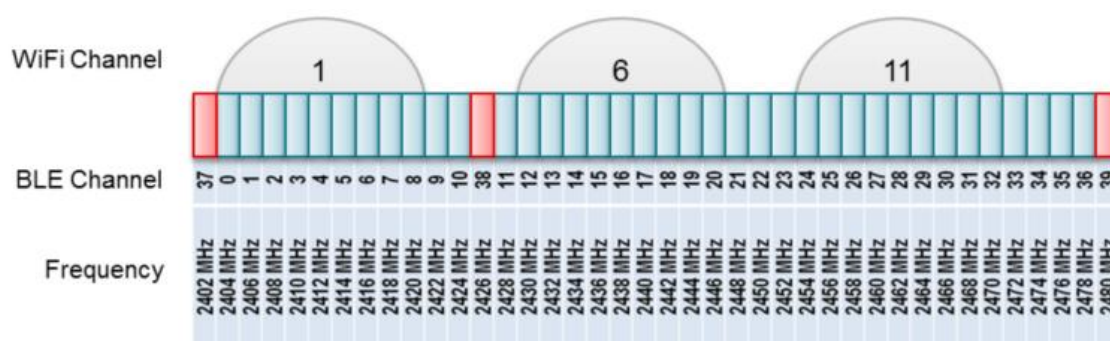
Rysunek 8 Unikanie kolizji pomiędzy kanałami Bluetooth i siecią WLAN za pomocą techniki adaptive frequency hopping [8].

### 2.3.1. Bluetooth Low Energy

Podczas gdy pierwsze wersje bluetooth zmierzały w kierunku polepszenia przepustowości transmisji, w wersji Low Energy (znana również pod skrótem BLE) główny nacisk położono na zmniejszenie zapotrzebowania prądowego do niezbędnego minimum. Podejście to wynikało z dynamicznie prężącego się sektora małych urządzeń mobilnych zasilanych bateryjnie. Należało tak opracować koncepcje technologii by urządzenia mogły działać na bateriach pastylkowych np. CR2032 nawet do kilku lat. Wymagało to gruntownego przebudowania architektury standardu. Wprowadzono szereg zmian i uproszczeń architektury [3].

Główną ideą BLE jest przesyłanie małych paczek danych oraz korzystanie przez większość czasu pracy z trybu uśpienia. Dlatego też transmisja danych sprowadza się do jak najszybszego połączenia się, nadania danych, rozłączenia i powrotu urządzenia w tryb uśpienia.

W wersji Low Energy zdecydowano się zredukować ilości kanałów do 40 o szerokości 2 MHz. Numeracja kanałów zawiera się w przedziale 0-39. Kanały od 0 do 36 przeznaczone są dla transmisji danych. Z kolei kanały 37, 38 i 39 są kanały rozgłoszeniowe. Aby zminimalizować interferencje kanałów rozgłoszeniowych ze standardem Wi-Fi, rozmieszczenie tych kanałów wybrano tak aby ich częstotliwości nie pokrywały się z kanałami sieci Wi-Fi [9]. Na rys. 9 przedstawiono częstotliwości kanałów BLE, ich numerację oraz usytuowanie kanałów rozgłoszeniowych (wyróżnione kolorem czerwonym) względem kanałów Wi-Fi.



Rysunek 9 Rozmieszczenie kanałów BLE wraz z zajmowanymi częstotliwościami, numeracja oraz usytuowaniem względem kanałów Wi-Fi [9].

Stosowana w BLE metoda modulacji to GFSK (*Gaussian Frequency Shift Keying*), czyli kluczowanie częstotliwości z gaussowskim kształtowaniem sygnału.

Pakiet przesyłanych danych znacząco ograniczono. W poprzednich wersjach połączenia między urządzeniami trwały długo i przesyłana była duża ilość danych. Pakiety zawierały do 1021 bajtów danych. Duża paczka danych wiązała się z wydłużonym czasem pracy nadajnika radiowego i wysokim poborem prądu przez niego. W BLE maksymalna ilość danych to 27 bajtów. Powoduje to krótki czas nadawania, a w konsekwencji zmniejszenie zapotrzebowania prądowego [3].

Zrezygnowano z techniki *scatternet* (opisanej w rozdziale 2.3). Nie wspierane są również kanały zawierające mowę, gdyż urządzenia takie jak zestawy głośnomówiące nie wpisują się w koncepcje Low Energy ze względu na nieustanne przesyłanie dużych ilości danych.

### 2.3.2. Beacon

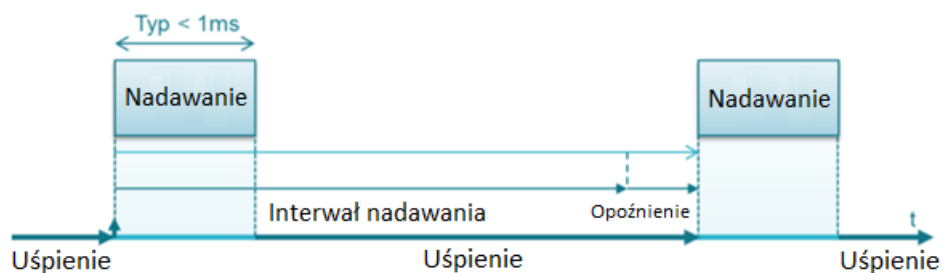
Pojęcie *Beacon* (ang. nadajnik, latarnia) odnosi się do małych urządzeń zasilanych bateryjnie które okresowo przesyłają pewne statyczne (np. jakiś rodzaj ogłoszenia bądź reklamy) lub dynamiczne (np. zmierzona temperatura) dane. Wysyłane informacje są niewielkie, nadawane z małą częstotliwością (zazwyczaj co około 1 s). W ideę takich nadajników idealnie wpisuje się technologia Bluetooth Low Energy [10]. Przykładowe urządzenie korzystające z technologii beacon wraz z obudową i baterią przedstawiono na rys. 10. Rozmiar takiego urządzenia nie przekracza kilku centymetrów. Obudowa jest niewiele większa od wymiarów baterii pastylkowej.



Rysunek 10 Urządzenie typu beacon [10].

Urządzenia typu beacon korzystają z nadawania w trybie rozgłoszeniowym. Do odebrania danych inne urządzenia bluetooth nasłuchują kanałów rozgłoszeniowych. Tryb pracy beaconsa przedstawiony jest na rys. 11. Widać na nim, iż urządzenie przez większość czasu jest bezczynne, oszczędzając energię w trybie uśpienia. Wybudzane jest jedynie w momencie nadawania, które zazwyczaj trwa mniej niż 1 ms. Interwał nadawania wynosi zazwyczaj około sekundy, może jednak być modyfikowany na potrzeby projektu. Do wspomnianego interwału dodawane jest opóźnienie, którego czas trwania jest losowany z przedziału od 1ms do 10 ms, po to, aby zmniejszyć prawdopodobieństwo wystąpienia sytuacji, w której urządzenia nadają w tym samym momencie [9].





Rysunek 11 Tryb pracy beaкона [9].

Istnieją już na rynku standardy określające format pakietów danych jakie nadają beacon'y. Firma Apple posiada swój standard iBeacon który współpracuje z systemem operacyjnym iOS. Otwartym standardem jest AltBeacon firmy Radius Networks. Z kolei firma Google opracowała swój standard o nazwie EddyStone [9].

Dla przykładu pakiet w standardzie iBeacon wygląda następująco (tabela 1) [9].

Tabela 1 Struktura pakietu w standardzie iBeacon

Nazwa sekcji	ID producenta	Typ danych	Długość danych	UUID	Major	Minor	Zmierzona moc
Rodzaj	Stałe	Stałe	Stałe	Modyfikowalne	Modyfikowalne	Modyfikowalne	Modyfikowalne
Rozmiar [bajty]	2	2	2	16	2	2	1

Pakiet danych w standardzie iBeacon posiada 27 bajtów które można podzielić na różne sekcje. Pierwsza sekcja to unikalny kod producenta. Następnie występuje UUID (*Universally Unique Identifier*) które może posłużyć do grupowania beaconów w sekcje, nadawania im różnych typów (czy jest to czujnik, czy reklama itp.). Wartości Major, Minor i Zmierzona moc używają się w zależności od zastosowania. Może to być np. zakodowana zmierzona wartość temperatury otoczenia [9].

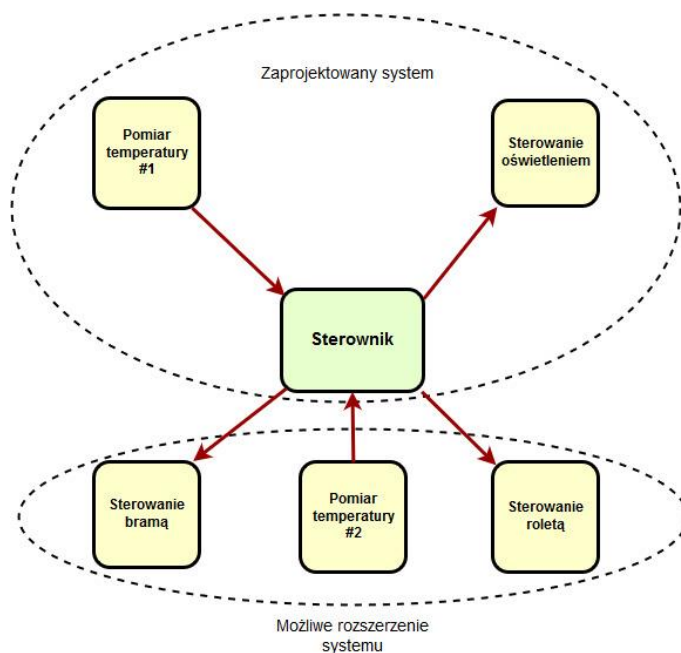
Technologia ta znajduje wiele zastosowań. Może być wykorzystana np. w interaktywnym muzeum. Gdy podejdziesz się do eksponatu, aplikacja w smartfonie automatycznie wyświetli nam opis dzieła sztuki. Innym zastosowaniem może być mobilna płatność bądź reklama w sklepie która aktywuje się po zbliżeniu się do regału sklepowego. Urządzenia typu beacon mogą służyć jako uliczne przewodniki dla osób niewidomych.

### 3. System bluetooth do sterowania urządzeniami w inteligentnym domu

W tym rozdziale przedstawiony zostanie system bluetooth do sterowania urządzeniami w inteligentnym domu. Zostały sformułowane następujące założenia projektowe:

- Bezprzewodowa komunikacja z modułami
- Periodyczne odczytywanie danych zmierzonych przez moduły pomiarowe
- Wydawanie poleceń dla modułów wykonawczych na żądanie użytkownika
- Możliwość obsługi wielu zadań współbieżnie
- Możliwość łatwego rozwoju oprogramowania w przyszłości
- Możliwość dołączania nowych modułów

Ze względu na bezprzewodową komunikację oraz na konieczność scentralizowania rozkazów i danych zdecydowano się na topologię gwiazdy. System składa się z jednostki sterującej oraz z modułów wykonawczych podłączonych bezpośrednio do sterownika. Na rys. 12 przedstawiono topologię projektowanego systemu. Dla pokazania idei systemu w pracy opisano i zrealizowano trzy moduły: sterownik, moduł pomiaru temperatury oraz moduł sterowania oświetleniem. W praktyce system mógłby składać się z większej ilości modułów o różnych zadaniach. Wiązałoby się to jedynie z rozbudową oprogramowania sterownika.



Rysunek 12 Topologia zaprojektowanego systemu wraz z jego możliwym rozszerzeniem.

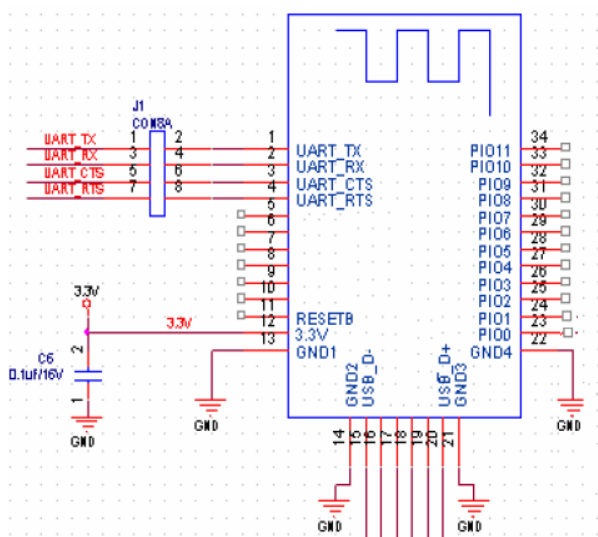
Zadaniem jednostki sterującej jest wysyłanie poleceń do modułów wykonawczych i zarządzaniem danymi przesyłanymi przez moduły pomiarowe.

Podłączone do sterownika moduły można podzielić na dwie główne grupy:

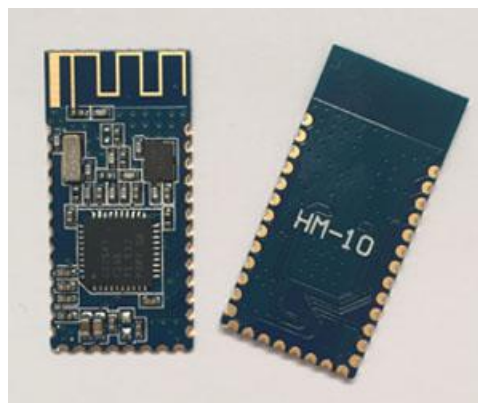
- Moduły pomiarowe, czyli urządzenia które okresowo mierzą wybraną wielkość fizyczną. Komunikacja ze sterownikiem jest jednostronna, tzn. moduł wysyła pomiar w postaci cyfrowej do jednostki centralnej.
- Moduły wykonawcze, czyli regulatory bądź przełączniki które oczekują na przyjęcie rozkazu, np. otwarcie bramy bądź opuszczenie rolety. Charakter komunikacji jest również jednostronny.

### 3.1.Moduł bluetooth

Aby spełnić założenie projektowe odnośnie bezprzewodowej komunikacji, zdecydowano się wykorzystać moduł bluetooth HM-10 firmy JN Huamao. Za wyborem właśnie tego modułu przemawia obsługiwana wersja Bluetooth Low Energy 4.0, niska cena, duża dostępność oraz bogata dokumentacja. Moduł ten będzie wykorzystywany przez wszystkie elementy systemu, gdyż stanowi niezbędną składową urządzenia wymaganą do komunikacji w systemie.



Rysunek 13 Opis wyprowadzeń modułu HM-10 [11].



Rysunek 14 Widok płytki modułu HM-10 [11].

Moduł oparty jest na Bluetooth Low Energy 4.0 oraz implementuje technologię iBeacon, która stanowi kluczową rolę w projektowanym systemie. W module został wykorzystany chip od Texas Instruments CC2540 [14]. Na rys. 13 opisane zostały wyprowadzenia modułu. W przypadku tego projektu wykorzystuje się tylko wyprowadzenia

nr 1 i 2 (UART\_TX oraz UART\_RX), zasilanie z filtrującym kondensatorem 100 nF podłączone do wyprowadzenia 12 oraz masa układu (wyprowadzenie nr 13). Widok płytki modułu HM-10 przedstawiono na rys. 14.

Według dokumentacji moduł powinien być zasilany napięciem z przedziału +2.5V~3.3VDC. Zużycie prądu podane przez producenta wynosi 8.5 mA podczas pracy i 50~200  $\mu$ A w trybie uśpienia [11].

Do skomunikowania się z modułem wykorzystuje się interfejs szeregowy UART. Transmisje na potrzeby systemu skonfigurowano następująco: 9600 baud, 8 bitów danych, brak bitu parzystości, 1 bit stopu.

Komunikacja opiera się na komendach AT. Służą one do wstępnej konfiguracji oraz do sterowania pracą modułu. Komenda może być zapytaniem (*Query*) bądź rozkazem (*Set*).

- Format komend *Query*: **AT+[polecenie]?**

Przykład: **AT+ADDR?** Zapytanie o adres modułu.

Odpowiedź w formacie: **OK+[Get/specjalny deskryptor]:[parametr]**

Przykład: **OK+ADDR:[adres MAC]**

- Format komend *Set*: **AT+[polecenie][parametr]**

Przykład: **AT+BAUD1** ustawienie prędkości 11900 baud/s

Odpowiedź w formacie: **OK+[Set/specjalny deskryptor]:[parametr]**

Przykład: **OK+Set:1**

Szczegółowa lista komend wraz z opisem znajduje się w dokumentacji technicznej modułu [11].

HM-10 może spełniać dwie role:

- *Master*, czyli możliwe jest inicjowanie połączenia
- *Slave*, czyli oczekiwanie na połączenie

Oprócz tradycyjnych trybów moduł można skonfigurować jako urządzenie beacon, w tym przypadku moduł implementuje technologie iBeacon. Dopuszczalne są dwa sposoby działania nadajnika [11]:

- Skanowanie i nadawanie
- Tylko nadawanie

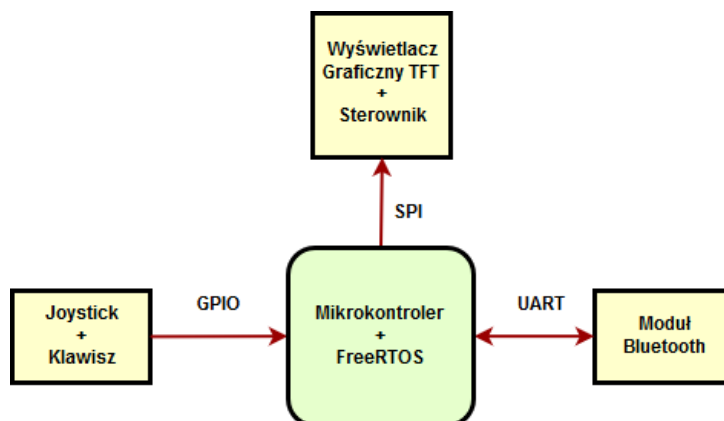
### 3.2. Sterownik

Centralnym punktem systemu jest sterownik. Steruje on pracą całego systemu. Zbiera dane z modułów pomiarowych oraz wysyła rozkazy do modułów wykonawczych.

Założenia projektowe dla sterownika:

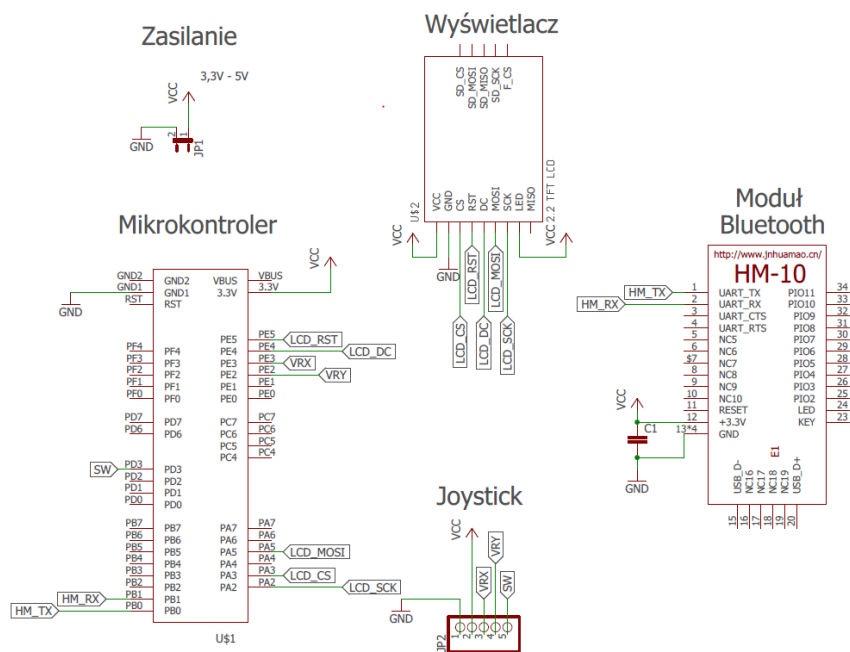
- Intuicyjny interfejs
- Możliwość obsługi wielu zadań współbieżnie
- Przejrzyste wyświetlanie danych

Sterownik składa się z mikrokontrolera, wyświetlacza, modułu bluetooth i joysticka z klawiszem. Na rys 15. zobrazowano schemat blokowy sterownika. Komunikacja między mikrokontrolerem a sterownikiem wyświetlacza graficznego zrealizowana jest za pomocą interfejsu szeregowego SPI, przy czym jest ona jednostronna, tj. jedynie mikrokontroler wysyła dane do wyświetlacza. Łączność z modułem bluetooth odbywa się za pomocą interfejsu UART (szerzej opisane w rozdziale 3.1). Z kolei Joystick jest podłączony do wyprowadzenia portu GPIO mikrokontrolera.



Rysunek 15 Schemat blokowy sterownika z wyszczególnionymi metodami komunikacji między peryferiami.

Na rys 16. zilustrowano schemat ideowy sterownika systemu. Opisano w nim połączenia między elementami składowymi urządzenia. Szczegółowy opis wyprowadzeń zawarto w rozdziałach o danym elemencie modułu.



Rysunek 16 Schemat ideowy sterownika.

Aby możliwe było obsługiwanie wielu zadań naraz, dogodnie jest zastosowanie systemu operacyjnego który znacząco ułatwia przełączanie się między wątkami oraz wymianę danych między zadaniami. Ze względu na dostępność dokumentacji i przykładów, prostoty obsługi i darmowy charakter, zdecydowano się na FreeRTOS [13].

### 3.2.1. Mikrokontroler

Użyty w projekcie mikrokontroler powinien posiadać:

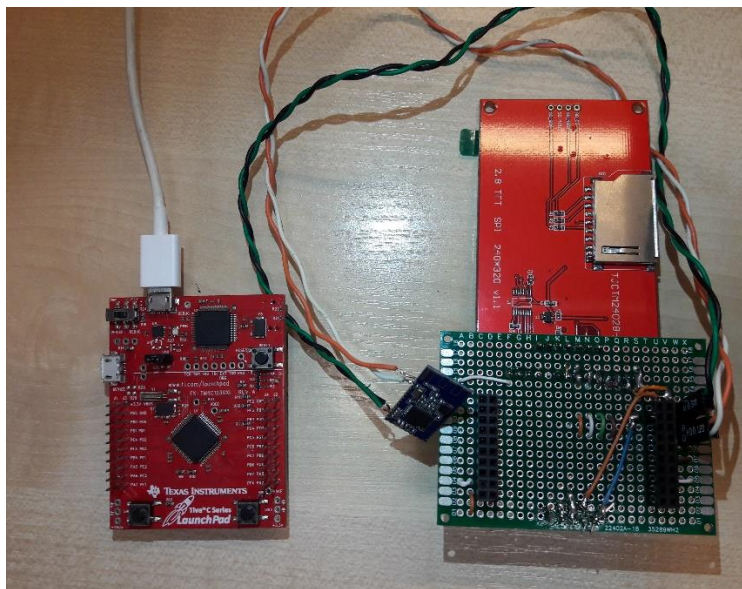
- Interfejs UART do komunikacji z modułem bluetooth
- Interfejs szeregowy SPI do komunikacji ze sterownikiem wyświetlacza
- Architekturę 32 bitową
- Kanały ADC dla joysticka
- Parametry wystarczające do zastosowania systemu operacyjnego

Zdecydowano się na mikrokontroler Tiva C Series TM4C123G firmy Texas Instruments. Mikrokontroler ten cechuje się 32-bitową architekturą ARM Cortex-M4. Ponadto posiada bogaty zestaw peryferii takich jak 12 kanałów ADC 12 bit, 2 Komparatory ADC, 8 interfejsów UART, 4 interfejsy SPI, 43 piny GPIO i wiele więcej [12]. Produkt firmy Texas Instruments spełnia wszystkie założenia projektowe i dlatego znakomicie się nadaje do zastosowania w opracowywanym systemie.

W projekcie wykorzystano płytke rozwojową wspomnianego mikrokontrolera, co ułatwiło znacznie rozwój oprogramowania z powodu zamieszczonego na płycie interfejsu debuggera JTAG. Ponadto skorzystano z zamontowanych konektorów żeńskich, aby skonstruować płytkę sterownika w postaci „nakładki” na płytke rozwojową. Na rys. 17 widoczny jest sterownik systemu. Z kolei na rys. 18 przedstawiono jak zrealizowano wspomnianą wcześniej „nakładkę”.



Rysunek 17 Widok sterownika.



Rysunek 18 Widok "nakładki" na płytke rozwojową mikrokontrolera.

### 3.2.2. Wyświetlacz

Podczas wyboru wyświetlacza brano pod uwagę następujące aspekty:

- Wyświetlacz graficzny
- Duża przekątna ekranu
- Niska cena
- Sterownik obsługujący interfejs szeregowy

Zdecydowano się na wyświetlacz 2,8 cali TFT z sterownikiem ILI9341. Zasilany jest on napięciem 3,3V co odpowiada zasilaniu sterownika. Komunikacja z mikrokontrolerem odbywa się za pomocą interfejsu szeregowego SPI. Zastosowanie tego interfejsu pozwala na zminimalizowanie ilości wyprowadzeń potrzebnych do sterowania wyświetlaczem do pięciu wyprowadzeń. Rozwiązanie szeregowe jest wolniejsze od równoległego sterowania, jednak w przypadku tego projektu prędkość wyświetlania danych nie jest priorytetowa, gdyż dane



na ekranie zmieniane są rzadko (tylko w przypadku naciśnięcia klawisza, odświeżenia temperatury). Zdecydowanie ważniejsza w projekcie jest mniejsza ilość użytych wyprowadzeń GPIO mikrokontrolera.

Spora przekątna ekranu pozwala na czytelne wyświetlanie danych oraz ułatwia zaprojektowanie intuicyjnego interfejsu. Ponadto sprzyja rozbudowaniu funkcjonalności systemu w przyszłości.

### **3.2.3. Joystick i klawisz**

Mechanizm poruszania się po menu sterownika powinien:

- Umożliwiać zmianę opcji w sposób intuicyjny
- Posiadać klawisz do zatwierdzania wyborów

Do tego zadania wykorzystano joystick z wbudowanym klawiszem. Pole manewru joysticka pozwala na sterowanie w dwóch wymiarach co znacząco wzbogaca możliwości interfejsu. Joystick składa się z dwóch potencjometrów (po jednym na oś). Mierząc napięcie na wyprowadzeniu danej osi wiadomo w jakim położeniu znajduje się gałka. Podając to napięcie na wejście komparatora można odczytać informację, w którą stronę wychylono joystick.

### **3.2.4. Interfejs**

Na rys. 19 zobrazowano widok interfejsu sterownika. W górnej części ekranu wyświetlana jest informacja o zmierzonej temperaturze. Poniżej umieszczono stan sterowanego oświetlenia. W środkowej części wyświetlacza znajduje się sekcja wyboru interakcji z oświetleniem. Dostępne są dwie opcje: włącz, wyłącz. Za pomocą znaku „>” wskazywana jest pożądana alternatywa. Znak przesuwany jest za pomocą ruchu gałki joysticka w górę lub w dół. Wybór jest zatwierdzany wciśnięciem gałki joysticka. W dolnej części zawarta jest nazwa systemu oraz imię i nazwisko autora oraz rok produkcji.





Rysunek 19 Widok interfejsu sterownika.

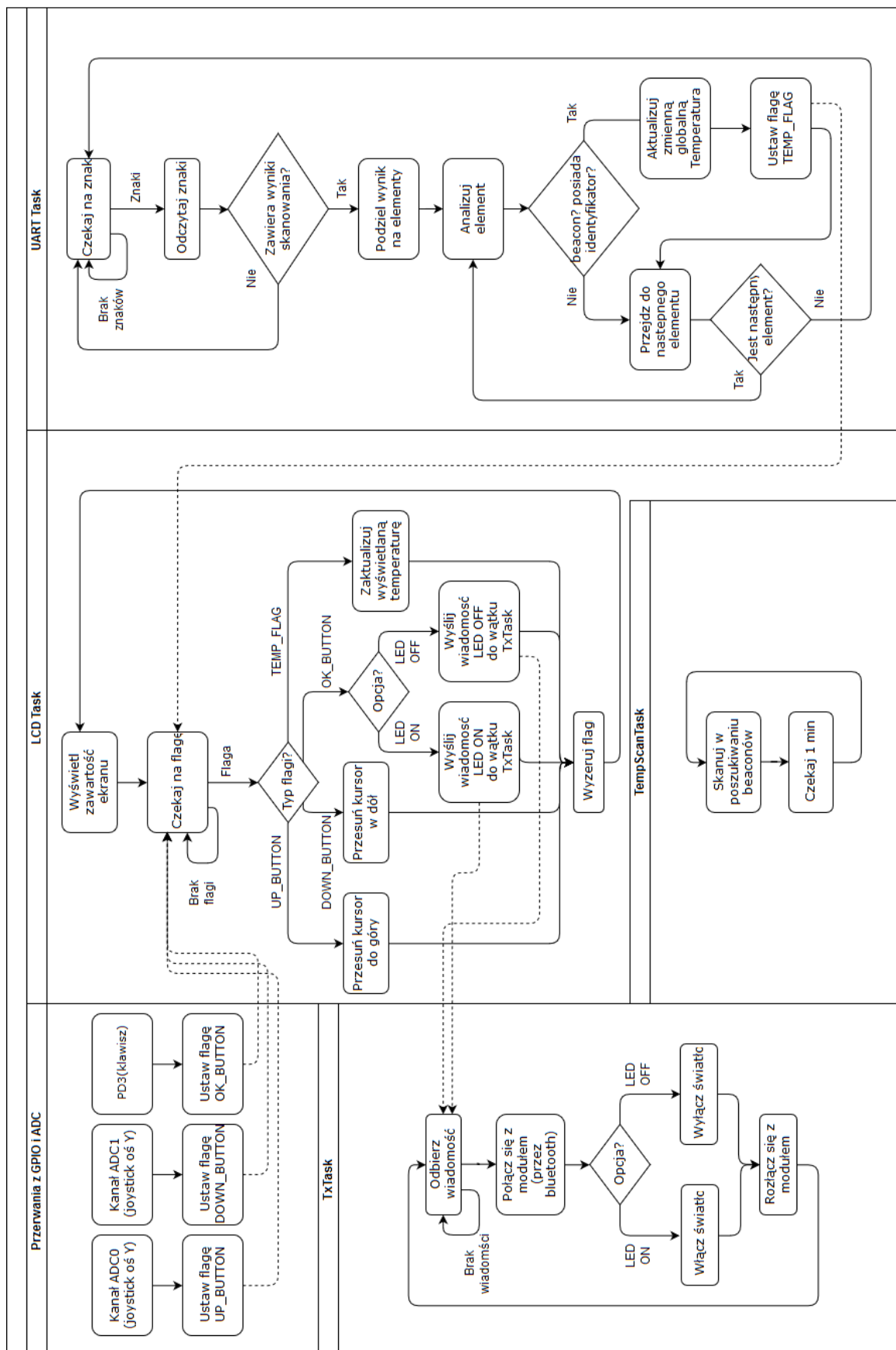
### 3.2.5. Konfiguracja modułu bluetooth

Aby moduł HM-10 mógł spełnić wyznaczone zadania, należało go skonfigurować w trybie *Master*. Użyto do tego następujących komend AT:

- **AT+ROLE1**, tryb *Master*.
- **AT+IMME1**, urządzenie czeka na wydanie polecenia nawiązania komunikacji, nie łączy się ze znanymi urządzeniami automatycznie. Taka konfiguracja wymagana jest ze względu na charakter systemu, automatyczne połączenia mogłyby zaburzyć proces skanowania.
- **AT+MODE0**, tryb transmisji, pozwala tylko na przesyłanie danych bez możliwości zdalnego konfigurowania modułu komendami AT.

### 3.2.6. Oprogramowanie sterujące

W projektowanym urządzeniu oprogramowanie ma strukturę wielowątkową i pracuje pod kontrolą systemu operacyjnego FreeRTOS. Na rys.20 przedstawione zostały schematy blokowe wątków programowych wykonywanych przez sterownik.



Rysunek 20 Schemat blokowy zadań sterownika.

Role każdego z wątków przedstawiają się następująco:

➤ UART Task

Rola zadania polega na analizowaniu ciągu znaków odbieranych przez interfejs szeregowy UART służący do komunikacji z modułem bluetooth. Jeśli odebrany został wynik skanowania, analizowana jest jego zawartość. W łańcuchu znaków szukane są konkretne wzorce takie jakie początek skanowania (OK+DISIS:[...] ), identyfikatory sensorów i koniec skanowania. Wyłuskana dana jest zapisywana w zmiennej globalnej a dla poinformowania procesu wyświetlania danych ustawiana jest flaga TEMP\_FLAG.

➤ LCD Task

Zadaniem tego procesu jest wyświetlanie danych na wyświetlaczu. Zajmuje się ponadto obsługa menu i przechodzeniem do jego poszczególnych podsekcji. Cały mechanizm został tak skonstruowany by zawartość ekranu była odświeżana tylko w momencie wystąpienia zmiany wyświetlanych informacji. Podejście to wynika z dość wolnej komunikacji szeregowej z ekranem. Nie sprawdziłoby się rozwiązanie z odświeżaniem o dużej częstotliwości.

Przez większość czasu proces jest zamrożony do czasu nadejścia flagi. Zastosowano następujące flagi:

- UP\_BUTTON – pochodzi z obsługi przerwań komparatora kanału ADC. Sygnalizuje ruch joysticka w górę
- DOWN\_BUTTON – jak wyżej, z tą zmianą, że ruch joysticka jest w dół
- OK\_BUTTON – flaga ustawiana w przerwaniu pochodzącym od wyprowadzenia GPIO. Wskazuje na naciśnięty przycisk zatwierdzający
- TEMP\_FLAG – Temperatura została zaktualizowana

Po odczytaniu flagi proces przechodzi do zasygnalizowanego polecenia. Następnie zeruje flagę, aktualizuje informacje na ekranie i ponownie czeka na kolejną flagę.

➤ Tx Task

Wątek ten zajmuje się wysyłaniem poleceń do modułów wykonawczych. Czeką na wiadomości które przychodzą z innych procesów, łączy się ze sparowanym modułem, następnie analizuje tą wiadomość i na podstawie tego przesyła rozkaz do zdalnego urządzenia. Na koniec rozłącza się i wraca do oczekiwania na wiadomość. W przyszłych

implementacjach systemów można by dołączyć możliwość wyboru modułu, z którym należy się połączyć.

#### ➤ Temp Task

Zadanie tego procesu jest bardzo proste. Cyklicznie (co 1 minutę) skanuje przestrzeń w poszukiwaniu sensorów. Wynik skanowania zostanie przeanalizowany w współbieżnym wątku (UART Task).

### **3.2.7. Zasilanie**

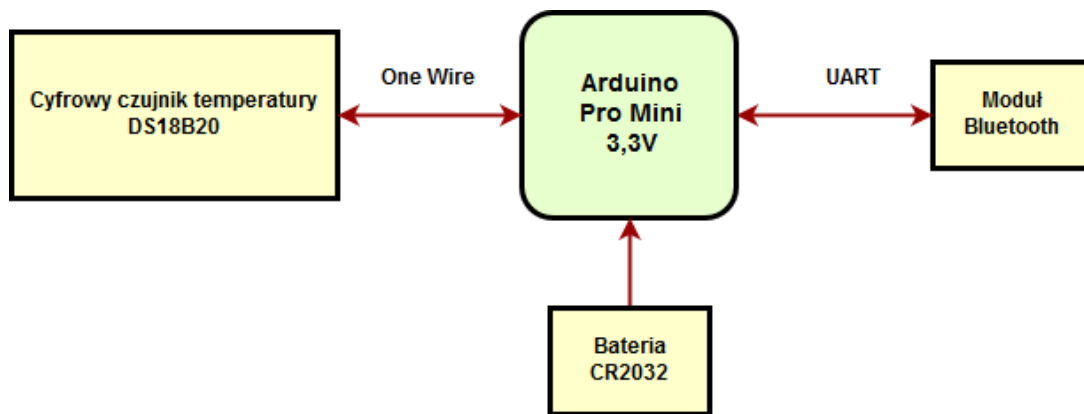
Płytką rozwojową wyposażoną jest w stabilizator napięcia o wartości 3,3 V. W związku z tym do zasilenia układu wystarczy kabel mini usb i jakikolwiek port USB bądź zasilacz sieciowy z wyjściem 5V. Stabilizator obniży napięcie do wymaganego 3,3V z którego korzystają wszystkie elementy układu.

## **3.3. Moduł pomiaru temperatury**

Kolejnym elementem systemu jest moduł pomiaru temperatury. Założenia projektowe dla tego urządzenia sformułowano następująco

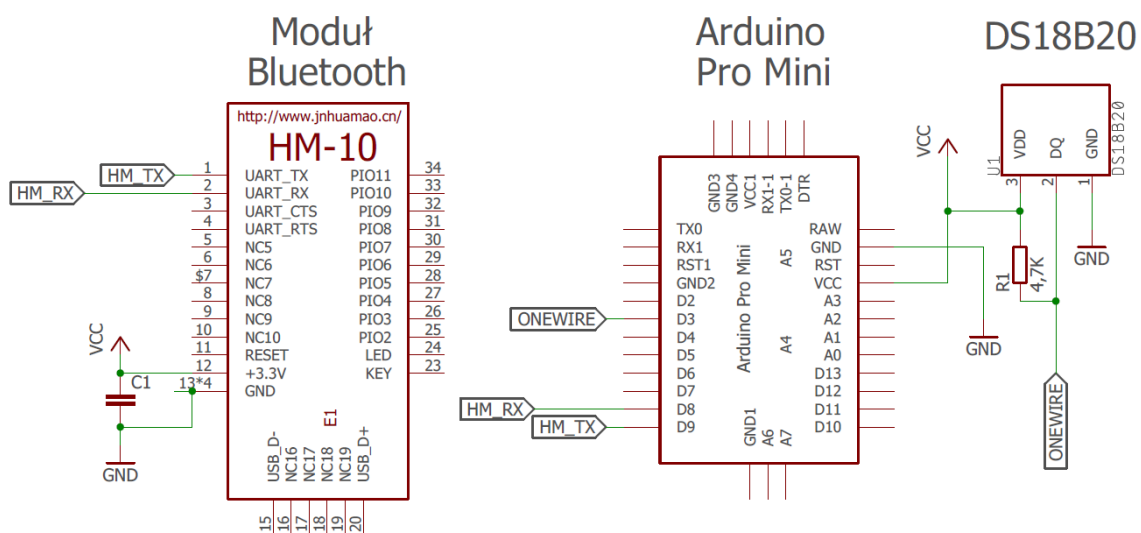
- Zasilanie bateryjne
- Żywotność na baterii do kilkudziesięciu dni
- Pomiar oraz nadawanie temperatury co minutę
- Dokładność pomiaru do 0,5°C
- Małe gabaryty
- Niska cena

Moduł składa się z mikrokontrolera, cyfrowego czujnika temperatury, modułu bluetooth i baterii. Schemat blokowy modułu oraz sposób komunikacji między elementami urządzenia zilustrowany został na rys. 21. Komunikacja z Arduino realizowana jest za pomocą interfejsu szeregowego 1-Wire. Informację do i z modułu bluetooth wysyłane są przy użyciu interfejsu UART (podobnie jak w każdym elemencie systemu). Całość zasilana jest baterią pastylkową CR2032.



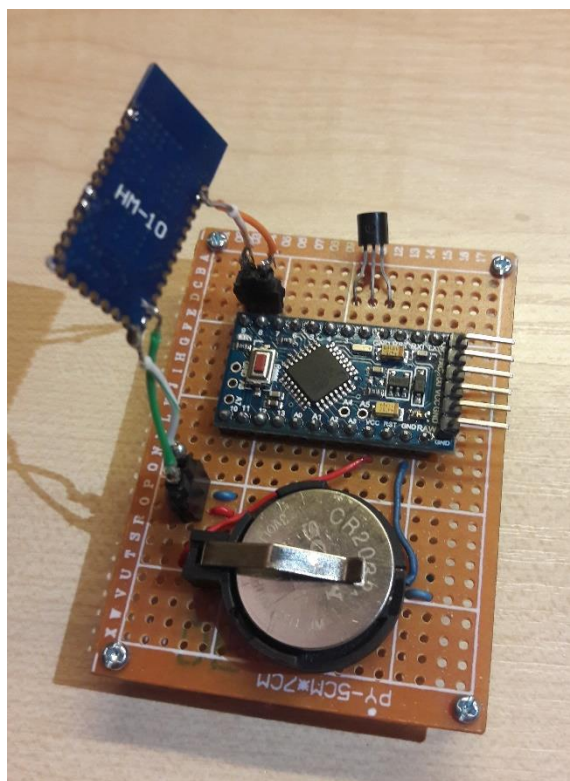
Rysunek 21 Schemat blokowy modułu pomiaru temperatury.

Na rys. 22 przedstawiono schemat ideowy urządzenia. Wykorzystane wyprowadzenia Arduino to D8 i D9 dla interfejsu UART oraz D3 dla komunikacji z czujnikiem temperatury.



Rysunek 22 Schemat ideowy układu pomiaru temperatury.

Układ został zmontowany na płytce uniwersalnej. Uzyskano małe rozmiary urządzenia ze względu na wykorzystane wersje mini elementów i niewielkie rozmiary baterii pastylkowej. Rys. 23 obrazuje widok zmontowanego modułu. Wymiary płytki to 5 cm na 7 cm.



*Rysunek 23 Widok płytki modułu pomiaru temperatury.*

### **3.3.1. Mikrokontroler**

Aby żywotność układu zasilanego bateryjnie wynosiła do kilku miesięcy należało zastosować elementy o niskim poborze prądowym. Całym układem steruje Arduino Pro Mini pracujące w logice 3,3V. W module wykorzystany został mikrokontroler AtMega 328 pracujący z częstotliwością 8 MHz. Peryferia w które wyposażone jest Arduino Pro Mini w zupełności wystarczą do zastosowania w projekcie. Używany jest tylko interfejs UART do komunikacji z modułem bluetooth oraz jeden pin GPIO do łączności z czujnikiem temperatury za pomocą standardu 1-Wire. Małe rozmiary i bardzo niski pobór prądu spełniają wymagania projektowe. Ponadto wykonywane przez układ zadanie jest mało złożone, dlatego nie wymagany jest skomplikowany mikrokontroler. Dla ułatwienia rozwoju oprogramowania skorzystać można było z prostych w obsłudze bibliotek Arduino [19].

### 3.3.2. Czujnik temperatury

Aby osiągnąć zamierzoną precyzję pomiaru do  $0,5^{\circ}\text{C}$  wybrano cyfrowy czujnik temperatury DS18B20. Według noty katalogowej czujnik ten posiada dokładność  $\pm 0,5^{\circ}\text{C}$  przy mierzonych temperaturach w zakresie od  $-10^{\circ}\text{C}$  do  $+85^{\circ}\text{C}$ . Posiada on 12 bitową rozdzielczość co przekłada się na ziarno  $0,0625^{\circ}\text{C}$ . Komunikacja odbywa się z pomocą technologii 1-Wire. Wyprowadzenie sygnałowe DQ wymaga podpięcia do szyny zasilania za pomocą rezystora  $4,7\text{ k}\Omega$  [15] (jest to tzw. *pull-up*, przedstawione na rys.22).

### 3.3.3. Konfiguracja beacons

Ze względu na wykonywane przez moduł zadanie, oparte na okresowym pomiarze temperatury oraz zważając na założenie o długiej żywotności pracy układu przy zasilaniu baterijnym, zdecydowano się na wykorzystanie technologii beacon. Technologia ta została opracowana dla tego typu zadań i spełnia wszystkie postawione przed modułem założenia. Nadawanie wykonywane przez urządzenie typu beacon ma bardzo małe zapotrzebowanie prądowe (rzędu kilkuset  $\mu\text{A}$ ) co pozwoli na znaczne wydłużenie czasu pracy układu na baterii.

Możliwe jest, aby skonfigurować moduł bluetooth HM-10 tak aby przekształcić go w beacon [11]. Implementowany jest standard iBeacon. W tabeli 1 (rozdział 2.3.2) przedstawiono strukturę pakietu danych standardu. Modyfikowalne sekcje pakietu danych można konfigurować za pomocą dostępnych komend AT. Autor projektu ustalił zastosowanie modyfikowalnych pól, za pomocą których identyfikowane są kompatybilne z systemem urządzenia typu beacon oraz dekodowana jest zawarta w nadajniku informacja. Zamyśl autora został przedstawiony w tabeli 2a i 2b.

Tabela 2a Pakiet nadawanych danych przez projektowany beacon, część 1.

Nazwa sekcji	UUID			
Przeznaczenie	Kod producenta (4 bajty)	Typ sensora: 1 – czujnik temperatury	Nie używane	ID sensora
Rozmiar [bajty]	4	2	4	6
Zawartość	0x12345678	0x0001	0xFFFF FFFF	0x0000 0000 0001

Tabela 2b Pakiet nadawanych danych przez projektowany beacon, część 2.

Nazwa sekcji	Major	Minor		Zmierzona moc
Przeznaczenie	Nr modułu pomiarowego w systemie	Część całkowita pomiaru (hex)	Część ułamkowa pomiaru(hex)	Nieużywane
Rozmiar [bajty]	2	1	1	1
Zawartość	0x0001	0x16	0x32	0xC5

Wartości dla UUID ustawiane są tylko raz przy konfiguracji wstępnej. Wartości aktualizowane na bieżąco to *Major*, które używa się do numerowania modułów pomiarowych w systemie, oraz część *Minor* która zawiera część całkowitą i ułamkową pomiaru temperatury. Dla przykładu pomiar 21,50°C zostanie zakodowany tak: Minor 0x1632 (21 to w kodzie szesnastkowym 16; 50 z kolei to 32).

Wstępna konfiguracja modułu wygląda następująco:

- **AT+IBEA1** włączanie trybu iBeacon
- **AT+DELO2** dozwolone tylko nadawanie, bez skanowania
- **AT+ADVI9** nadawanie danych co 1285 ms
- **AT+IBE012345678** pierwsze 4 bajty UUID
- **AT+IBE10001FFFF** kolejne 4 bajty UUID
- **AT+IBE2FFFF0000** następne 4 bajty UUID
- **AT+IBE300000001** ostatnie 4 bajty UUID
- **AT+MARJ0x0001** numer sensora ustawiony na 1

Mikrokontroler dokonuje pomiaru temperatury co minutę i wysyłając komendy AT do modułu bluetooth aktualizuje pakiet danych, po czym, za pomocą komendy AT+SLEEP, cały układ przechodzi w stan uśpienia do kolejnego pomiaru. Używana jest instrukcja AT+MINO0x1632 do ustawienia nowej wartości Minor, gdzie zakodowana jest temperatura [11].

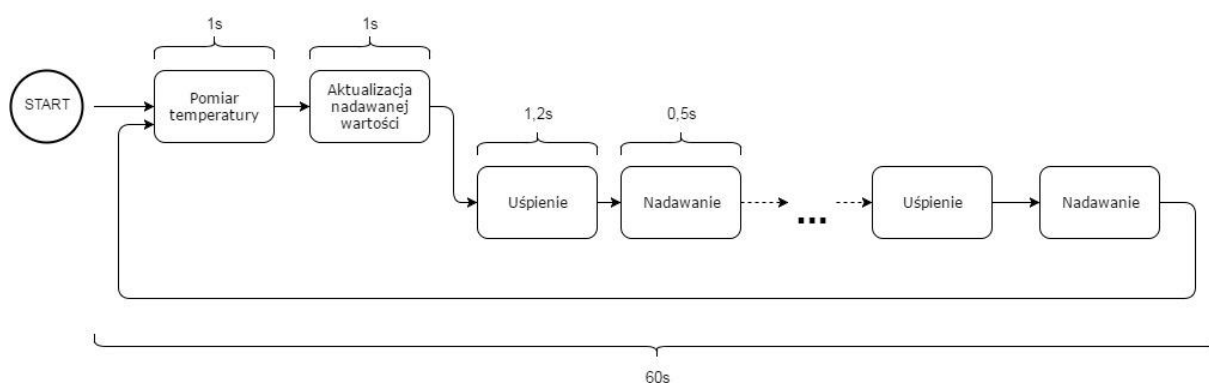


### 3.3.4. Zasilanie i zapotrzebowanie prądowe

Aby obliczyć czas pracy układu przy zasilaniu bateryjnym należało zmierzyć prąd pobierany przez moduł. Cykl (który trwa 60 s) pracy układu pomiaru temperatury można podzielić na 4 fazy:

- pomiar temperatury
- zaktualizowanie wartości nadawanej przez nadajnik
- uśpienie układu
- nadawanie wiadomości rozgłoszeniowej przez nadajnik

Każda z faz ma różny czas trwania oraz pobierana jest inna wartość prądu. Rys. 24. pokazuje przebieg cyklu układu wraz z czasem trwania poszczególnych faz.



Rysunek 24 Cykl pracy i jego poszczególne fazy.

W tabeli 3 umieszczono zmierzony czas trwania poszczególnych faz oraz pobierany przez nie prąd.

Tabela 3 Poboru prądu dla każdej z faz pracy.

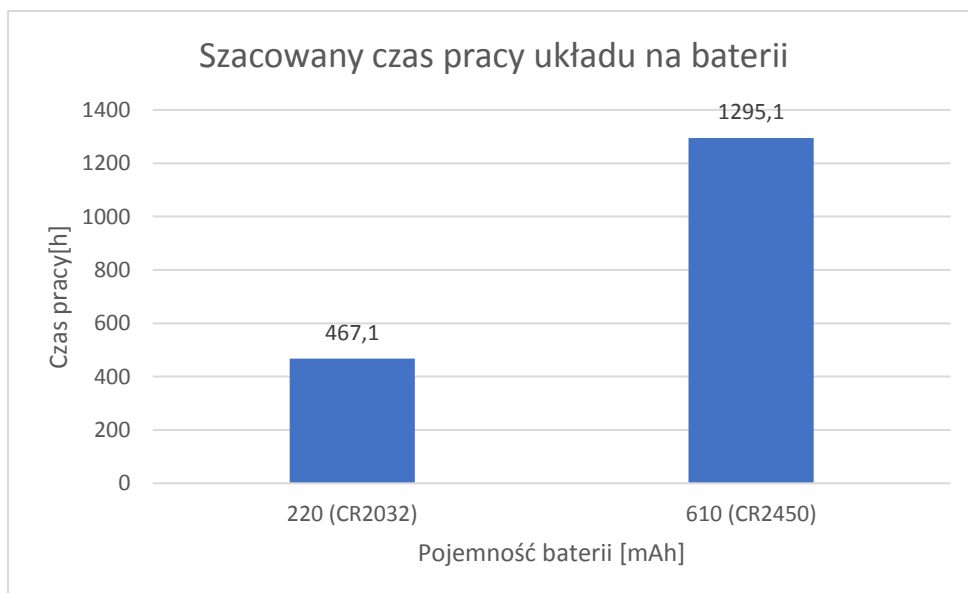
Faza	Pobór prądu [mA]	Czas trwania fazy [s]	Czas wykonywania podczas cyklu [s]
Pomiar temp.	3,07	1	1
Aktualizacja	10,97	1	1
Uśpienie	0,1	1,2	40,94
Nadawanie	0,6	0,5	17,06

Na podstawie danych z tabeli 3 wyliczono uśredniony pobór prądu, który wynosi 0,471 mA.

Wykorzystując wyliczony uśredniony pobór prądu można oszacować czas pracy układu przy użyciu baterii. Ze względu na założenia projektowe (niewielkie rozmiary), zdecydowano się wykorzystać baterie tzw. pastylkową. Przeanalizowano dwa modele baterii:

- CR2032 3V 220mAh [16]
- CR2450 3V 610mAh [17]

Czas pracy układu w przypadku wykorzystania baterii CR2032 wynosi 467,1 h (19 dni 11,1h). W przypadku baterii CR2450 o większej pojemności czas ten wydłuża się do 1295,1h (53 dni 23,1h). Na rys. 25 przedstawiono szacowane czasy pracy układu przy wykorzystaniu wymienionych modeli baterii.



Rysunek 25 Wykres szacowanego czasu pracy układu przy zastosowaniu zasilania baterijnego o różnej pojemności.

### 3.4.Moduł sterowania oświetleniem

Kolejnym modułem zaprojektowanym w systemie jest moduł sterowania oświetleniem. Zalicza się on do grupy urządzeń, które wykonują tylko polecenia przesyłane przez sterownik.

Podczas projektowania układu priorytetowe były następujące aspekty:

- Włączanie i wyłączanie oświetlenia (lampy) na życzenie użytkownika.
- Oświetlenie zasilane napięciem sieciowym.

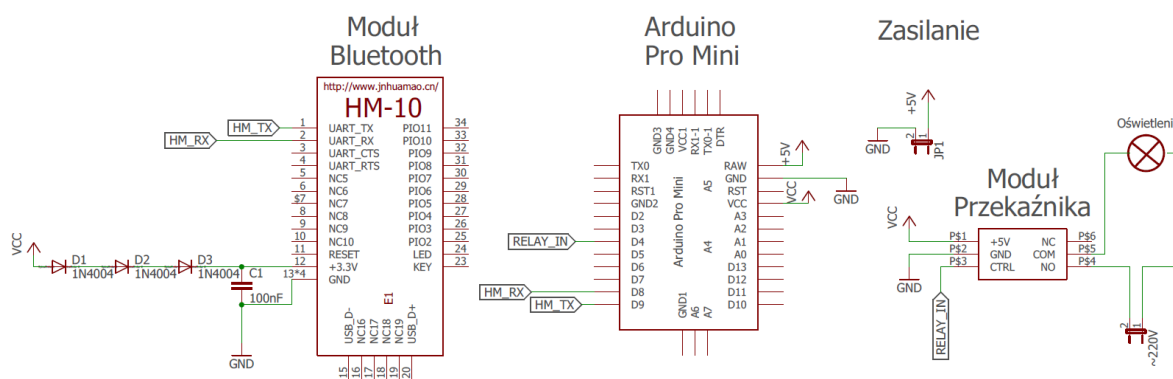
Skonstruowany układ składa się z mikrokontrolera, modułu bluetooth i przełącznika. Schemat blokowy został przedstawiony na rys. 26. Podobnie jak w przypadku poprzednio

opisanych elementów systemu, komunikacja między Arduino a modulem bluetooth przeprowadzana jest za pomocą interfejsu UART. Moduł przekaźnika sterowany jest wyprowadzeniem portu GPIO mikrokontrolera.



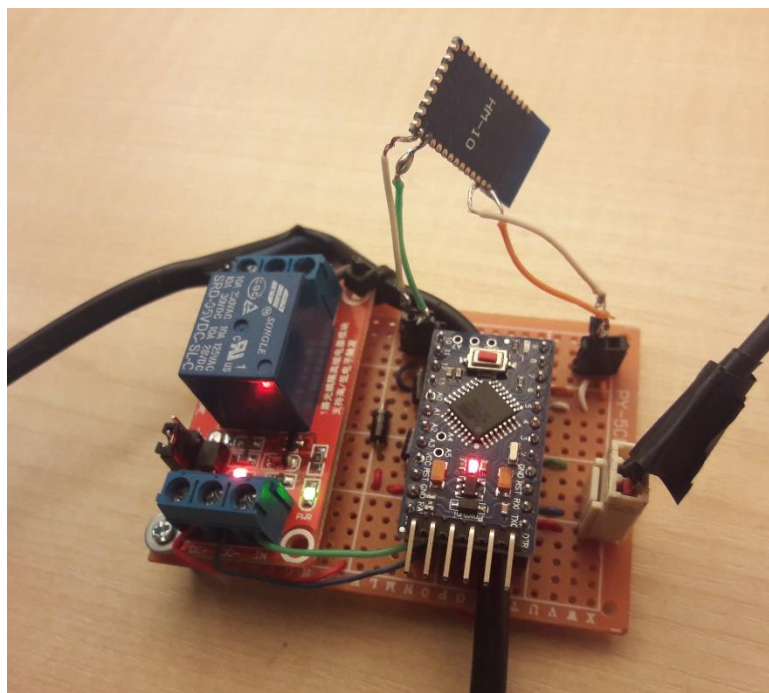
Rysunek 26 Schemat blokowy modułu sterowania oświetleniem.

Rys. 27 obrazuje schemat ideowy modułu sterowania oświetleniem. Podobnie jak w przypadku modułu pomiaru temperatury, wykorzystywane są wyprowadzenia D8 i D9 Arduino dla interfejsu UART. Zasilanie układu zostało opisane w rozdziale 3.4.4. Szerszy opis przekaźnika zawarto w rozdziale 3.4.2.



Rysunek 27 Schemat ideowy modułu sterowania oświetleniem.

Układ zmontowano na płytce uniwersalnej co pozwoliło, podobnie jak w przypadku modułu pomiaru temperatury, na uzyskanie niewielkich gabarytów urządzenia. Widok zmontowanego urządzenia przedstawiono na rys. 28. Wymiary płytki wynoszą 5 cm na 7 cm.



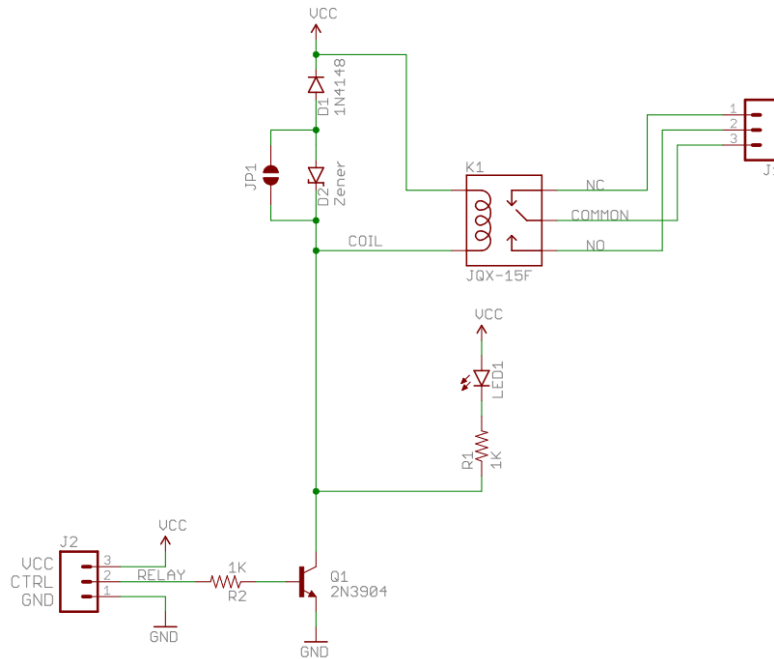
Rysunek 28 Widok modułu sterowania oświetleniem.

### 3.4.1. Mikrokontroler

Podobnie jak w przypadku modułu pomiaru temperatury wykorzystano Arduino Pro Mini. Zdecydowano się jednak na wersję 5V. Wybór ten wynika z konieczności podłączenia przekaźnika, który musi być zasilony, wyższym niż jak w przypadku modułu pomiaru temperatury, napięciem 5 V. Zastosowany mikrokontroler szerzej opisano w rozdziale 3.3.1.

### 3.4.2. Przekaźnik

Do sterowania oświetleniem wykorzystano moduł przekaźnika firmy SparkFun, którego specyfikacja pozwala na podłączenie do jego wyprowadzeń napięcia sieciowego 220 V. Schemat ideowy modułu przedstawiono na rys. 29. Wyprowadzenie sterujące CTRL połączone jest z wyprowadzeniem D4 portu GPIO Arduino. Sterowanie odbywa się za pomocą ustawienia stanu logicznego wysokiego, aby włączyć zasilanie oświetlenia lub stanu niskiego, aby stworzyć w obwodzie lampy przerwę i tym samym wyłączyć oświetlenie. Kabel oświetlenia został wpięty w wyprowadzenia przekaźnika COM (ang. *common*, wspólny) oraz NO (ang. *Normally Open* rozarty).



Rysunek 29 Schemat ideowy modułu przekaźnika [18].

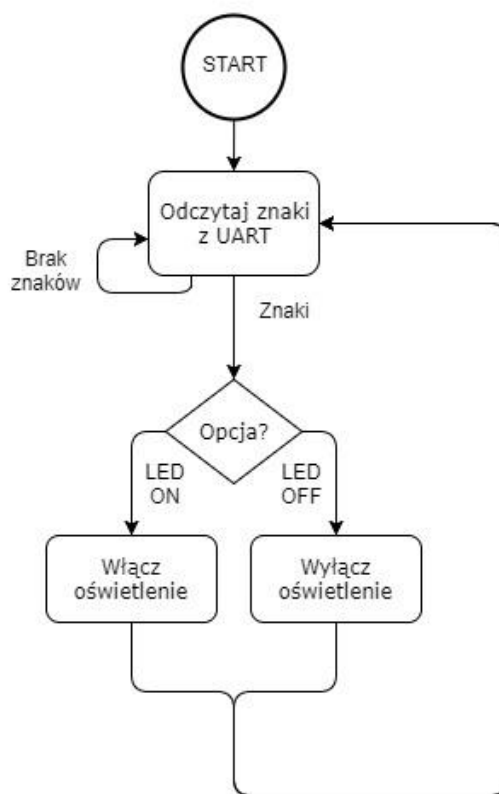
### 3.4.3. Konfiguracja modułu HM-10

Moduł bluetooth został skonfigurowany jako urządzenie *Slave*, gdyż układ jedynie czeka na polecenia sterownika. W tym celu wykorzystano następujące komendy AT:

- **AT+ROLE0**, tryb *Slave*.
- **AT+MODE0**, tryb transmisji (wy tłumaczony w rozdziale 3.2.5.).

Praca układu jest przedstawiona na rys. 30. Gdy nawiązana zostanie łączność ze sterownikiem, moduł bluetooth otrzyma wiadomość wysłaną poprzez układ sterujący, wówczas prześle łańcuch znakowy do Arduino za pomocą UART. Wiadomość może zawierać jeden z dwóch komunikatów:

- **LEDON**, oznacza polecenie włączenia oświetlenia. Wówczas wartość logiczna wyprowadzenia sterującego przekaźnikiem zostanie ustawiona na poziom wysoki.
- **LEDOFF**, polecenie wyłączenia oświetlenia. Niski poziom logiczny wyprowadzenia sterującego.



Rysunek 30 Schemat blokowy zadań modułu sterowania oświetleniem.

#### 3.4.4. Zasilanie

Układ jest zasilany zasilaczem sieciowym o napięciu 5,5 V. Napięcie z zasilacza podawane jest na wyprowadzenie RAW Arduino, przez co wykorzystywany również jest stabilizator 5 V zamontowany na płycie mikrokontrolera [19]. Na potrzeby modułu bluetooth napięcie obniżono do 2,9V za pomocą trzech diod prostowniczych połączonych szeregowo.

### 3.5. Analiza współpracy systemu z urządzeniami działającymi pod kontrolą systemu operacyjnego Android

Obecnie bardzo popularne są urządzenia pracujące pod kontrolą systemu Android, takie jak smartfony czy tablety. Urządzenia te również korzystają z komunikacji w technologii Bluetooth.

Istnieje wiele zastosowań wymienionych urządzeń w opracowywanym projekcie. Podstawowym zastosowaniem jakie od razu nasuwa się na myśl jest rola sterownika systemu.

Możliwości teoretycznej aplikacji w telefonie są na tyle duże, iż pozwalają spełnić przyjęte zamierzenia projektowe dotyczące sterownika systemu. W takim przypadku zastąpiono by całkowicie opracowany w projekcie moduł sterownika (skonstruowany sterownik nie uczestniczyłby w systemie). Rozwiązanie to jednak ma jedną podstawową wadę: Które urządzenie zarządzałoby pracą systemu w momencie, gdy domownik przebywałby poza domem, a wraz z nim osobisty telefon z androidem, który zachowywał się jako jednostka sterująca? Z tego powodu mało praktyczna jest, w ocenie autora, przedstawiona idea.

Bardziej funkcjonalnym zastosowaniem byłaby możliwość ingerencji smartfona w zarządzanie systemem. Istniałyby dwa urządzenia w systemie nadzorujące pracę. Zaletą tego rozwiązania byłaby większa wygoda w sterowaniu urządzeniami, gdyż użytkownik korzystałby z systemu za pomocą telefonu, który zazwyczaj nosi przy sobie bez konieczności podchodzenia do sterownika, który ze względu na zasilanie zasilaczem umieszczony jest w jednym stałym miejscu. Wadą tego pomysłu jest znaczące zaburzenie i komplikowanie topologii systemu ze względu na dwa urządzenia typu Master, a co za tym idzie możliwe byłyby konflikty między tymi urządzeniami.

Korzystne wydaje się być zastosowanie urządzenia z Androidem jako modułu w roli Slave. Przykładowa aplikacja mogłaby odpytywać sterownik odnośnie zebranych danych pomiarowych. Innym przykładem jest zastosowanie urządzenia jako bramki do Internetu w celu pobrania potrzebnych systemowi danych z sieci lub umieszczenie danych o statusie systemu na serwerze.

Niestety smartfon nie mógłby być wykorzystywany jako zestaw głośnomówiący dla telefonu stacjonarnego, ponieważ wersja BLE nie obsługuje kanałów przeznaczonych dla mowy (ograniczenie to opisano w rozdziale 2.3.2).

## Podsumowanie

Celem pracy było zaprojektowanie systemu bluetooth do sterowania urządzeniami w inteligentnym domu. W skład skonstruowanego systemu wchodził układ sterownika systemu, moduł pomiaru temperatury i moduł sterowania oświetleniem. Zaprojektowane elementy systemu pozwoliły na zaprezentowanie i przetestowanie omawianej koncepcji. Założenia projektowe opisane w części praktycznej projektu zostały spełnione.

Zastosowany mikrokontroler przy pomocy systemu operacyjnego FreeRTOS pozwolił na sprawne zarządzanie urządzeniami w przedstawionym systemie. Wyświetlacz TFT sterownika o przekątnej 2,8 cala umożliwił przejrzyste wyświetlanie danych. Dołączony do sterownika joystick wraz z klawiszem zapewnił wygodny interfejs urządzenia. Używany w systemie moduł bluetooth pozwolił na sprawną komunikację między urządzeniami.

Wykorzystanie technologii Bluetooth Low Energy, a w szczególności technologii beacon, pozwoliło na uzyskanie czasu pracy modułu pomiaru temperatury przy zasilaniu baterią CR2032 wynoszącym 19 dni i 11,1 godzin. Sposobem na wydłużenie czasu pracy do 53 dni 23,1 godzin byłoby zastosowanie baterii o zwiększonej pojemności typu CR2450.

Moduł sterowania oświetleniem włączał bądź wyłączał lampę w momencie, gdy odebrany został rozkaz użytkownika. Sterowanie lampą występowało z opóźnieniem około 1s ze względu na konieczność nawiązania łączności ze sterownikiem za każdym razem, gdy przesyłane było polecenie.

Przetestowano realny zasięg łączności modułów. Przeprowadzone próby wykazały, iż komunikacja w otwartej przestrzeni jest możliwa z maksymalną odległością około 50 m. W teście, w którym na torze przesyłowym występowały trzy ściany o grubości 15 cm uzyskano maksymalny zasięg około 10 m. Uzyskane wyniki pozwalają na zastosowanie opracowanego systemu w budynku.

Z analizy możliwości współpracy systemu z urządzeniami opartymi o system operacyjny Android wynikało, iż współpraca jest możliwa jednak wnosi do koncepcji zarówno nowe zagrożenia jak i ulepszenia. Wadą takiej współpracy byłaby np. zaburzanie i komplikowanie topologii gwiazdy ze względu na obecność kilku urządzeń pełniących nadzorującą rolę. Zaletą mogłoby być wykorzystanie np. smartfona w roli łącznika systemu z Internetem.



Koncepcja systemu ma, w opinii autora, duży potencjał rozwojowy. Sterownik został tak zaprojektowany, aby możliwa była rozbudowa funkcjonalności urządzenia. W przyszłości można by opracować interfejs bardziej obszerny wykorzystując dwa wymiary joysticka (wybieralne opcje rozmieszczone w osiach X i Y). Ponadto korzystając z zasobów systemu operacyjnego zadania sterownika można wzbogacić o samodzielne wykrywanie urządzeń w pobliżu. Kolejnym przydatnym elementem rozwoju oprogramowania może być indeksowanie połączonych urządzeń i nadawanie im nazw wybranych przez użytkownika, np. temperatura-sypialnia czy oświetlenie-garaż. Oczywistym rozszerzeniem systemu jest opracowanie kolejnych modułów pomiarowych bądź wykonawczych.

Podsumowując, zaprojektowany system spełnia postawione przed nim wymagania. Skonstruowanie tego systemu pozwoliło na potwierdzenie prawdziwości tezy, iż opracowanie systemu opartego na komunikacji bezprzewodowej wykorzystując technologię Bluetooth umożliwi sterowanie urządzeniami w inteligentnym domu.

## Bibliografia

- [1] Mikulik J., red. Niezabitowska E., „Budynek inteligentny” t. I – „Potrzeby użytkownika a standard budynku inteligentnego”, Wydawnictwo Politechniki Śląskiej, Gliwice, 2005
- [2] P. Lalanda, J. Bourcier, J. Bardin and S. Chollet, “Smart Home Systems”, Uniwersytet Grenoble, 2010, strony 6-7
- [3] N. Gupta, “Inside Bluetooth Low Energy”, Artech House, 2013
- [4] Piotr Gajewski, Stanisław Wszelak, „Technologie bezprzewodowe sieci teleinformatycznych”, Warszawa Wydawnictwa Komunikacji i Łączności, 2008
- [5] <https://www.wi-fi.org/who-we-are> [dostęp 29.11.2017]
- [6] <http://www.zigbee.org/zigbee-for-developers/applicationstandards/zigbee-building-automation/> [dostęp 2.11.2017]
- [7] <http://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-zigbee.html> [dostęp 10.11.2017]
- [8] <https://www.design-reuse.com/articles/5715/adaptive-frequency-hopping-for-reduced-interference-between-bluetooth-and-wireless-lan.html> [dostęp 5.11.2017]
- [9] <http://www.ti.com/lit/an/swra475a/swra475a.pdf> [dostęp 15.11.2017]
- [10] <https://www.appfutura.com/blog/beacons-all-you-need-to-know-about-them/> [dostęp 29.11.2017]
- [11] JN Huamao Technology Company, “Bluetooth 4.0 BLE module datasheet”
- [12] <http://www.ti.com/lit/ug/spmu296/spmu296.pdf> [dostęp 22.11.2017]
- [13] [http://www.freertos.org/Documentation/FreeRTOS\\_Reference\\_Manual\\_V9.0.0.pdf](http://www.freertos.org/Documentation/FreeRTOS_Reference_Manual_V9.0.0.pdf) [dostęp 19.11.2017]
- [14] <http://www.ti.com/lit/ug/swru271g/swru271g.pdf> [dostęp 28.11.2017]
- [15] <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf> [dostęp 18.11.2017]
- [16] <http://data.energizer.com/pdfs/cr2032.pdf> [dostęp 20.11.2017]
- [17] <http://data.energizer.com/pdfs/cr2450.pdf> [dostęp 20.11.2017]

- [18] [https://cdn.sparkfun.com/datasheets/Components/General/SparkFun\\_Beefcake\\_Relay\\_Control\\_Kit\\_v20a.pdf](https://cdn.sparkfun.com/datasheets/Components/General/SparkFun_Beefcake_Relay_Control_Kit_v20a.pdf) [dostęp 29.11.2017]
- [19] <https://store.arduino.cc/arduino-pro-mini> [dostęp 29.11.2017]