**Politechnika Wrocławska**

**Wydział Elektroniki**

**Kierunek: Elektronika**

**Specjalizacja: Aparatura Elektroniczna**

**PROJEKT INŻYNIERSKI**

**System bluetooth do sterowania urządzeniami w inteligentnym domu**  
*Bluetooth system for smart home control*

Autor: Grzegorz Korzeniewski

Prowadzący pracę:

Dr inż. Grzegorz Świrniak

Katedria Metrologii Elektronicznej i Fotonicznej

OCENA PRACY:

WROCŁAW, 2017

**Spis treści**

**Wprowadzenie**

1. **Główne założenia inteligentnego domu**
2. **Analiza bezprzewodowych interfejsów cyfrowych**
   1. Bluetooth
      1. Low Energy
      2. iBeacon
   2. Zigbee
   3. Wifi
3. **Współpraca systemu z urządzeniami mobilnymi z systemem operacyjnym Android**
4. **System bluetooth do sterowania urządzeniami w inteligentnym domu**
   1. Moduł bluetooth
   2. Sterownik
      1. Mikrokontroler
      2. Wątki
      3. Konfiguracja modułu bluetooth
      4. Wyświetlacz
      5. Joystick i klawisz
      6. Zasilanie
   3. Moduł pomiaru temperatury
      1. Czujnik temperatury
      2. Zasilanie
   4. Moduł sterowania oświetleniem

**Podsumowanie**  
**Bibliografia**

**Wprowadzenie**

Termin inteligentnego budynku pojawił się w latach 70 tych w Stanach Zjednoczonych i wywodzi się z sektora przemysłowego. Początki znajduje w optymalizacji środowiska rozwoju roślin oraz w systemach kontroli produkcji zautomatyzowanej. Wraz z postępem technologii koszt realizacji projektów inteligentnych budynków zmalał, co spowodowało wykorzystywanie ich w biurowcach by później przejść do prywatnych domostw.

Domy są automatyzowane w celu polepszenia komfortu zamieszkiwania, efektywnym wykorzystywaniu energii (np. cieplnej lub elektrycznej), zminimalizowanie emisji szkodliwych zanieczyszczeń, zwiększeniu bezpieczeństwa oraz maksymalizacji funkcjonalności. Możliwości są nieograniczone, z każdym dniem pojawiają się coraz to nowsze pomysły na rozbudowanie idei inteligentnego domu.

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 projekowania domu. Bezprzewodowa łączność również znacząco zwiększa wygodę użytkowania systemu.

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 przeanalizowaniu bezprzewodowych interfejsów jakimi są technologie WiFi, Zigbee oraz Bluetooth. W przypadku technologii Bluetooth szczegółowo opisano najnowszą wersje 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 zawarto również testy systemu.

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

**Opracowanie systemu opartego o komunikację bezprzewodową w technologii Bluetooth umożliwi sterowanie urządzeniami w inteligentnym domu**

**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.

Moc nadawcza podzielona jest na 3 klasy:

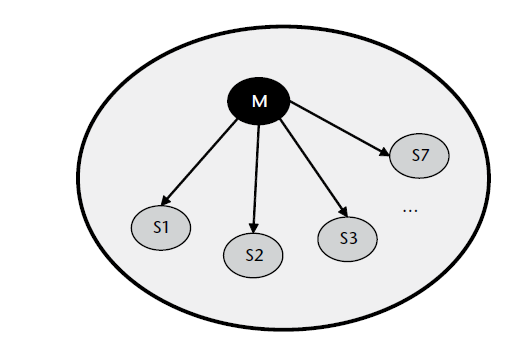
* klasa 1 (100 mW) zasięg teoretyczy do 100 m
* klasa 2 (2,5 mW) teoretyczny zasięg do 10 m, najczęściej stosowana
* klasa 3 (1 mW) teoretyczny zasięg do 1 m, używana rzadko

Przez lata opracowano wiele wersji technologii.

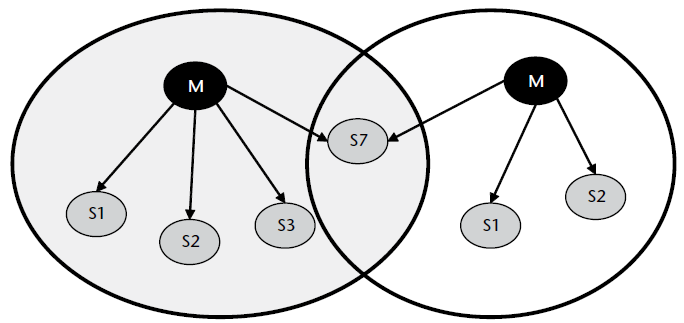
* Bluetooth 1.0 – 21 kb/s
* Bluetooth 1.1 – 124 kb/s
* Bluetooth 1.2 – 328 [kb/s](https://pl.wikipedia.org/wiki/Bit_na_sekund%C4%99)
* Bluetooth 2.0 + EDR – wprowadzenie Enhanced Data Rate zwiększyło transfer teoretyczny do 2,1 [Mb/s](https://pl.wikipedia.org/wiki/Bit_na_sekund%C4%99) (około 3 [Mb/s](https://pl.wikipedia.org/wiki/Bit_na_sekund%C4%99) wliczając narzut protokołu)
* Bluetooth 2.1 + EDR - uproszczenie i ujednolicenie procesu parowania urządzeń BT, wsparcie dla przyszłych implementacji [NFC](https://pl.wikipedia.org/wiki/Near_Field_Communication), zmniejszenie zużycia energii[[4]](https://pl.wikipedia.org/wiki/Bluetooth#cite_note-4)
* Bluetooth 3.0 + HS (High Speed) – 24 [Mb/s](https://pl.wikipedia.org/wiki/Bit_na_sekund%C4%99) (3 MB/s)
* Bluetooth 3.1 + HS (High Speed) – 40 [Mb/s](https://pl.wikipedia.org/wiki/Bit_na_sekund%C4%99) (5 MB/s)
* Bluetooth 4.0 + LE (Low Energy) – 1 Mb/s[[5]](https://pl.wikipedia.org/wiki/Bluetooth#cite_note-5) znacząco ograniczono pobór energii (np. praca czujnika temperatury, przez wiele miesięcy na baterii pastylkowej), kosztem obniżonego transferu oraz zwiększono realny zasięg działania do 100 m
* Bluetooth 4.1 - standard opracowany do zastosowania w tzw. "[internecie rzeczy](https://pl.wikipedia.org/wiki/Internet_rzeczy" \o "Internet rzeczy)" (urządzenia typu "[wearables](https://pl.wikipedia.org/wiki/Wearables" \o "Wearables)"), umożliwiający bezpośrednią łączność przedmiotów z internetem[[6]](https://pl.wikipedia.org/wiki/Bluetooth" \l "cite_note-6)
* Bluetooth 4.2 - w stosunku do poprzednich wersji: szybszy transfer, wyższy poziom bezpieczeństwa, nawiązanie łączności z przedmiotami - łatwiejsze[[7]](https://pl.wikipedia.org/wiki/Bluetooth#cite_note-7)[[8]](https://pl.wikipedia.org/wiki/Bluetooth#cite_note-8)
* Bluetooth 5.0 - ujednolicenie wersji, szybszy transfer – 2 Mb/s dla urządzeń typu ["wearables"](https://pl.wikipedia.org/wiki/Wearables) i 50 Mb/s do normalnych, realny zasięg działania do 140m[[9]](https://pl.wikipedia.org/wiki/Bluetooth#cite_note-9)

W architekturze Bluetooth główną jednostką jest pikosieć. 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.

Metodą na rozszerzenie pikosieci jest technika *Scatternet,*  czyli łączeniu kilku pikosieci między sobą urządzeniem typu *bridge.* Pozwala to na wzajemną komunikacje większej ilości urządzeń.



Rysunek 1 Pikosieć. M-Master, S- Slave



Rysunek 2 Technika scatternet (nieobslugiwane w wersji Low Energy).

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.

Pasmo 2,4GHz jest wykorzystywane przez wiele technologii, takich jak WiFI 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 miedzy 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.

**Bluetooth Low Energy**

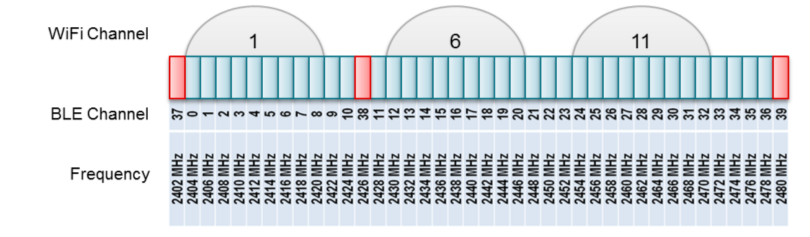
Podczas gdy pierwsze wersje bluetooth zmierzały w kierunku polepszenia przepustowości transmisji, w wersji Low Energy (znana również pod skrótem BLE) główny nacisk położono na drastyczne zmniejszenie zapotrzebowania prądowego do niezbędnego minimum. Podejście to wynikało z dynamicznie prężącego się sektora 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.

Główną ideą BLE jest przesyłanie małych paczek danych oraz spędzanie większości czasu pracy w trybie 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 na zredukowanie 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 tą kanały rozgłoszeniowe.

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

Aby zminimalizować interferencje kanałów rozgłoszeniowych ze standardem WiFi, rozmieszczenie kanałów wybrano w sposób przedstawiony na rys 3.



Rysunek 3 Rozmieszczenie kanałów rozgłoszeniowych BLE ( kolor czerwony) oraz kanałów WiFi

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 miedzy 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.

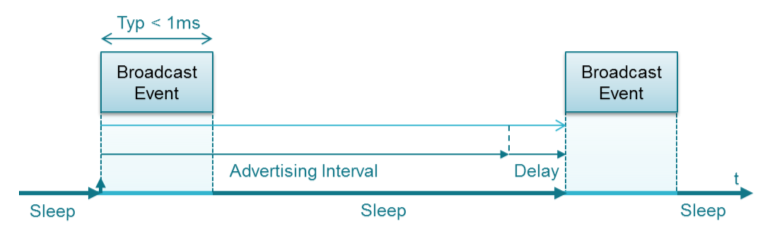
**Beacon**

Pojęcie **Beacon** (ang. nadajnik, latarnia) odnosi się do małych urządzeń zasilanych bateryjnie które periodycznie 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 1 s). W ideę takich nadajników idealnie wpasowuje się technologia Bluetooth Low Energy.



Rysunek 4 Urządzenie typu beacon

Urządzenia typu beacon korzystają z nadawania w trybie rozgłoszeniowym. Do odebrania danych inne urządzenia bluetooth nasłuchują kanałów rozgłoszeniowych. Tryb pracy beacona przedstawiony jest na rysunku nr 5.



Rysunek 5 Tryb pracy beacona

Istnieją już na rynku standardy określające format pakietów danych jakie nadają beacony. 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.

Dla przykładu pakiet w standardzie iBeacon wygląda następująco

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0x4C00 | 0x02 | 0x15 | 0x123e4567-e89b-12d3-a456-426655440000 | 0x1234 | 0x0001 | 0xc5 |
| Stałe | Stałe | Stałe | Modyfikowalne | Mod. | Mod. | Mod. |
| ID producenta | Typ danych | DLugość danych | UUID | Major | Minor | Zmierzona moc |

Standard 25 bajtami które można podzielić na różne sekcje. Pierwsza sekcja to unikalny kod producenta. Następnie UUID (Universally unique identifier) które może posłużyć do grupowania beaconów sekcje, nadawania im różnych typów (czy jest to czujnik, czy reklama itp.) Wartości Major Minor i Zmierzona moc używa się w zależności od zastosowania. Może to być np. zakodowana zmierzona wartość temperatury otoczenia.

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

**System bluetooth do sterowania urządzeniami w inteligentnym domu**

W tym rozdziale przedstawiona zostanie główna koncepcja systemu bluetooth do sterowania urządzeniami w inteligentnym domu. Zostały sformułowane następujące założenia projektowe:

* Bezprzewodowa komunikacja z modułami
* Periodyczne odczytywanie danych zmierzonych przez czujniki
* 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, czyli system składa się z głównej jednostki sterującej oraz z modułów wykonawczych podłączonych bezpośrednio do sterownika.

Zadaniem jednostki sterującej jest wysyłanie poleceń do modułów wykonawczych i zarządzaniem danymi przesyłanymi przez zdalne czujniki.

Podłączone do sterownika moduły można podzielić na dwie główne grupy:

* Moduły pomiarowe, czyli czujniki 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 przyjście rozkazu, np. otwarcie bramy bądź opuszczenie rolety. Charakter komunikacji również jednostronny: sterownik -> moduł wykonawczy.

SCHEMAT IDEOWY TUTAJ

Opisać zrealizowane moduły

**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.

|  |  |
| --- | --- |
| Rysunek 6 Rzeczywista płytka modułu HM-10 | Rysunek 7 Schemat ideowy modułu HM-10 |

Moduł oparty jest na **Bluetooth Low Energy 4.0** oraz implementuje technologię **iBeacon**, która stanowi kluczową role w projektowanym systemie. W module został wykorzystany chip od Texas Instruments **CC2540.**

Według dokumentacji, moduł zasilany powinien być zasilany napięciem z przedziału **+2.5V~3.3VDC**. Zużycie prądu podane przez producenta wynosi **8.5 mA podczas pracy** i **50~200 mikroA** w trybie uśpienia.

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.

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* (z ang. nadajnik), w tym przypadku moduł implementuje technologie iBeacon. Dopuszczalne są dwa sposoby działania nadajnika:

* Skanowanie i nadawanie
* Tylko nadawanie

**Sterownik**

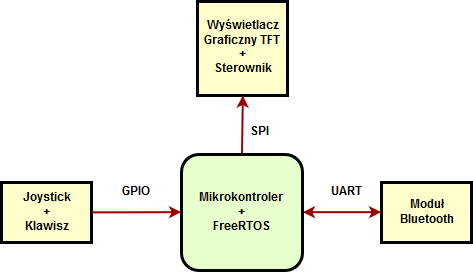
Centralnym punktem systemu jest sterownik. Steruje on pracą całego systemu. Zbiera dane z sensorów oraz wysyła rozkazy do modułów wykonawczych.

Założenia projektowe dla sterownika:

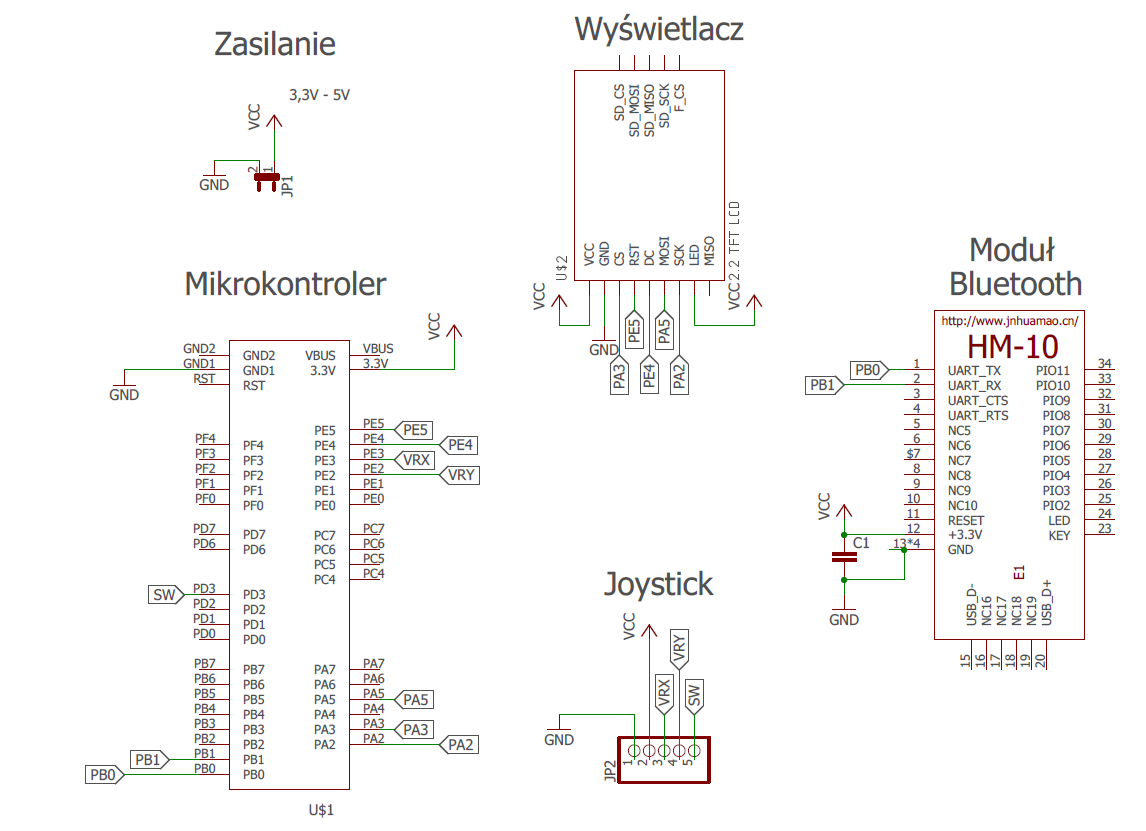
* Intuicyjny interfejs
* Możliwość obsługi wielu zadań współbieżnie
* Przejrzyste wyświetlanie danych

**ZDJĘCIA!**

Sterownik składa się z mikrokontrolera, wyświetlacza, modułu bluetooth i joysticka z klawiszem.



Rysunek 8 Schemat blokowy sterownika z wyszczególnionymi metodami komunikacji miedzy peryferiami



Rysunek 9 Schemat podłączenia urządzeń

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

**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 peryferiów takich jak 12 kanałow ADC 12 bit, 2 Komparatory ADC, 8 interfejsów UART, 4 interfejsy SPI, 43 piny GPIO i wiele więcej. Produkt firmy Texas Instruments spełnia wszystkie założenia projektowe i dlatego znakomicie się nadaje do zastosowania w opracowywanym systemie.

W projekcie wykorzystano płytkę rozwojową wspomnianego mikrokontrolera, co ułatwiło znacznie rozwój oprogramowania z powodu zamieszczonego na płytce interfejsu debuggera JTAG. Ponadto skorzystano z zamontowanych konektorów żeńskich aby skonstruować płytkę sterownika w postaci „nakładki” na płytkę rozwojową.

**Zasilanie**

Do zasilania urządzenia wykorzystano regulator napięcia w który wyposażona jest płytka rozwojowa. Aby zasilić wystarczy kabelek mini usb i jakikolwiek port USB bądź zasilacz sieciowy z wyjściem 5V. Regulator zmniejszy napięcie do wymaganego 3,3V z którego korzystają wszystkie elementy układu.

**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 pinów 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 ( naciśnięcie klawisza, odświeżenie temperatury po minucie). Zdecydowanie ważniejsza w projekcie jest mniejsza ilość użytych pinów GPIO mikrokontrolera.

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

**Joystick i klawisz**

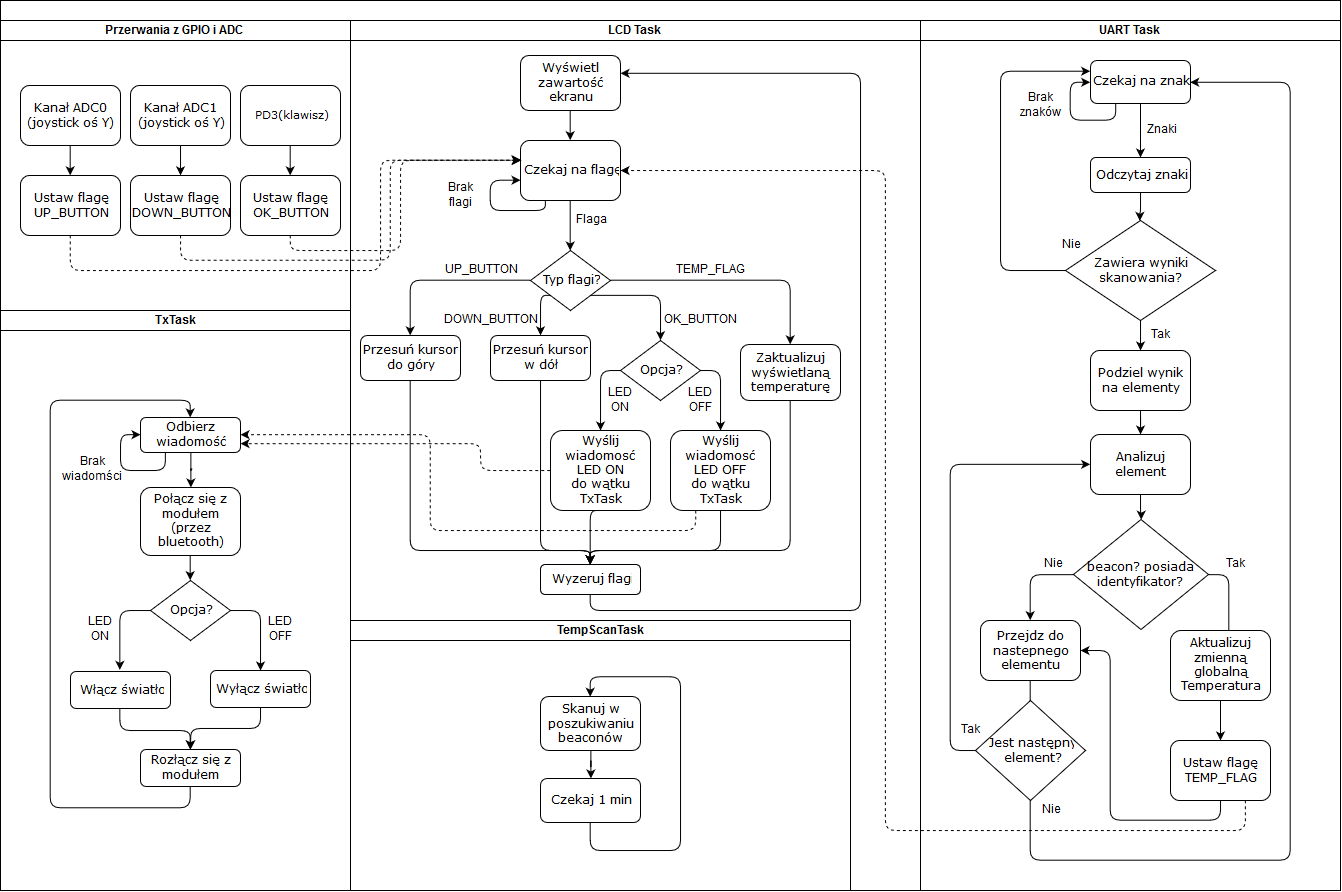
Mechanizm poruszania się po menu sterownika powinien:

* 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 rezystorów ( 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.

**Wątki**

Poniżej przedstawiony został schemat ideowy zadań wykonywanych przez sterownik.



Rysunek 10 Schemat ideowy zadań sterownika

**LCD Task**

Zadaniem tego procesu jest wyświetlanie danych na wyświetlaczu. Zajmuje się ponadto obsluga 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 pochodzącącym z pinu 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.

**TxTask**

Wątek ten zajmuje się wysyłaniem poleceń do modułów wykonawczych. Czeka na wiadomości które przychodzą z innych procesów, łączy się ze sparowanym modułem analizuje tą wiadomość i na podstawie tego przesyła rozkaz do zdalnego urządzenia. Nastepnie rozłącza się i wraca do oczekiwania na wiadomość. (W przyszłych implementacjach systemów możnaby 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**

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 do poinformowania procesu wyświetlania danych ustawiana jest flaga TEMP\_FLAG.

**Konfiguracja modułu bluetooth**

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

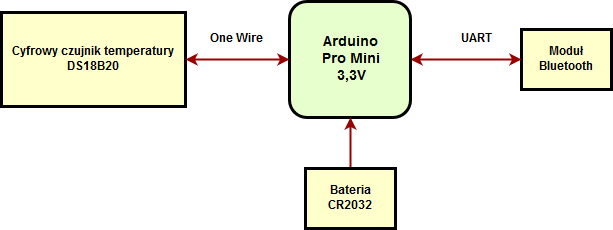
* **AT+ROLE1** , tryb *Master.*
* **AT+IMME1** , urządzenie czeka na wydanie polecenie nawiązania komunikacji, nie łączy się ze znanymi urządzeniami automatycznie. Taka konfiguracja wymagana jest ze względu na systematykę systemu, automatyczne połączenia mogłyby zaburzyć proces skanowania.
* **AT+MODE0**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

**Moduł czujnika temperatury**

Kolejnym elementem systemu jest moduł czujnika temperatury. Załozenia projektowe dla tego urządzenia sformułowano następująco

* Zasilanie bateryjne
* Żywotność na baterii do kilku miesięcy
* Pomiar oraz nadawanie temperatury co minutę
* Precyzja pomiaru do 0,1 stopnia Celsjusza
* Małe gabaryty
* Niska cena

Moduł składa się z mikrokontrolera, cyfrowego czujnika temperatury, modułu bluetooth i baterii.



Rysunek 11 Schemat ideowy modułu pomiaru temperatury

SCHEMAT ELEKTRONICZNY!!

**Mikrokontroler**

Do osiągnięcia żywotności układu na baterii do kilku miesięcy należało zastosować elementy o niskim poborze prądowym. Całym układem steruje Arduino Pro Mini pracujące w logice 3,3V. Małe rozmiary i bardzo niski pobór prądu spełniają wymagania projektowe.

**Czujnik temperatury**

Aby osiągnąć zamierzoną dokładność do 0,1 stopnia Celsjusza wybrano cyfrowy czujnik temperatury DS18B20. Posiada on 12 bitową rozdzielczość co przekłada się na ziarno 0,0625 st C. Komunikacja odbywa się z pomocą technologii One Wire.

Zmierzono zapotrzebowania prądowe mikrokontrolera:

* 1,12 mA podczas pracy układu
* 93 mikroA w trybie uśpienia

Zdecydowano się na wykorzystanie technologii beacon. Możliwe jest aby skonfigurować moduł bluetooth HM-10 tak aby przekształcić go w beacona. Implementowany jest standard iBeacon. Pola bitowe pakietu danych ustalono w następujący sposób:

|  |  |  |  |
| --- | --- | --- | --- |
| 0x12345678 | 0x0001 | 0xFFFF FFFF | 0x0000 0000 0001 |
| Kod producenta | Typ sensora:  1 – czujnik temperatury | NIeużywane | ID sensora |
| UUID | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 0x0001 | 0x16 | 0x32 | 0xC5 |
| Nr czujnika w systemie (Modyfikowalne) | Część całkowita pomiaru (hex) | Częśc ułamkowa pomiaru(hex) | Modyfikowalne |
| Major | Minor | | Zmierzona moc |

Wartości dla UUID ustawiane są tylko raz przy konfiguracji wstępnej. Wartosci aktualizowane na bieżąco to *Major*, które używa się do numerowania czujników w systemie, oraz część *Minor* która zawiera część całkowitą i ułamkową pomiaru temperatury. Dla przykładu pomiar 21,50 st C zostanie zakodowany tak : Minor 0x1632 (21 to w kodzie heksadecymalnym 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 cały układ przechodzi w stan uśpienia do kolejnego pomiaru. Używana jest instrukcja AT+MINO0x1632 do ustawienia nowej wartości Minor gdzie zakodowana jest temperatura.

**BIBLIOGRAFIA**

* <https://www.design-reuse.com/articles/5715/adaptive-frequency-hopping-for-reduced-interference-between-bluetooth-and-wireless-lan.html>
* Inside bluetooth low Energy version 1
* Inside bluetooth low Energy version 2
* hm-10 datasheet
* tiva datasheet
* freertos datasheet
* <http://www.ti.com/lit/ug/swru271g/swru271g.pdf>
* Ds18B20 datasheet

Dzięki technologii iBeacon, nie wymagana jest aktywna komunikacja obu stron aby odczytać dane czujników. Wystarczy że czujnik periodycznie zmierzy daną wielkość fizyczną sterownik oraz uaktualni identyfikator nadajnika. Przez reszte czasu moduł przechodzi w tryb uśpienia aby ograniczyć pobór prądu. Sterownik z kolei skanuje przestrzeń w poszukiwaniu czujników, odczytuje wartość zmierzoną zapisaną w identyfikatorze nadajnika iBeacon i również może przejść w tryb uśpienia bądź skomunikować się z innymi modułem wykonawczym.