**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 Rzeczywista płytka modułu HM-10 | Rysunek 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**

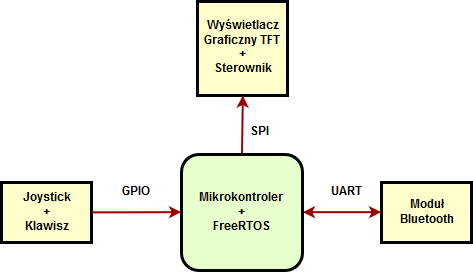
Centralnym punktem systemu jest sterownik. Steruje on pracą całego systemu. Zbiera dane z sensorów oraz wysyła rozkazy do modułów wykonawczych.

Założenia projektowe dla sterownika:

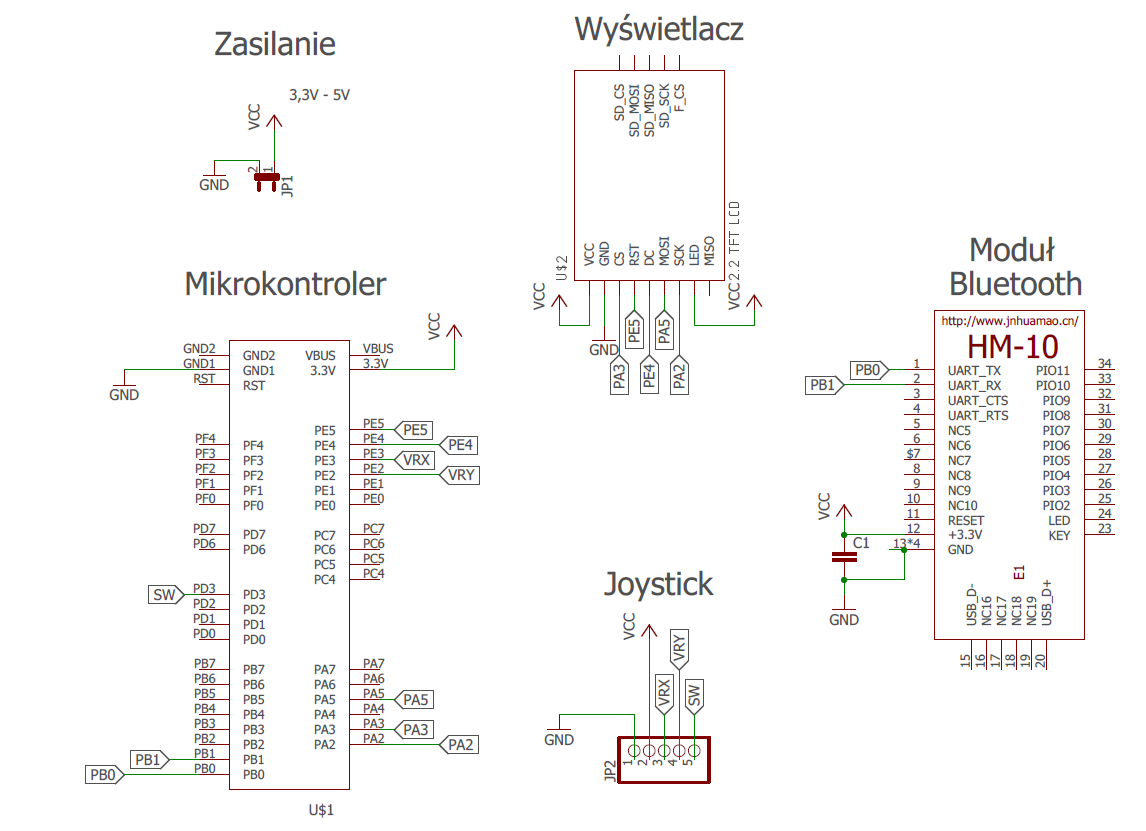
* Intuicyjny interfejs
* Możliwość obsługi wielu zadań współbieżnie
* Przejrzyste wyświetlanie danych

**ZDJĘCIA!**

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



Rysunek Schemat blokowy sterownika z wyszczególnionymi metodami komunikacji miedzy peryferiami



Rysunek 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**.

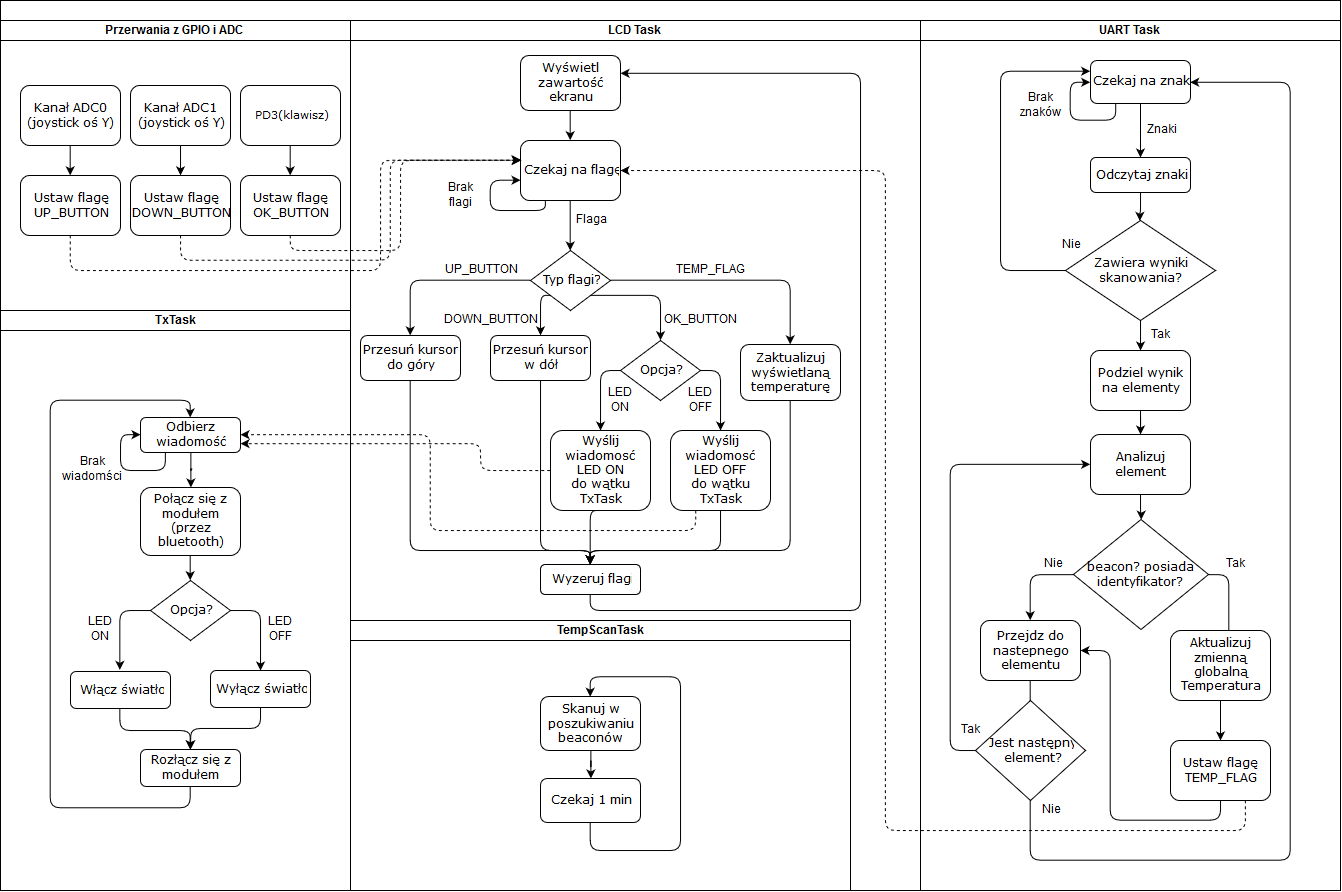
**Mikrokontroler**

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

**Wątki**

Poniżej przedstawiony został schemat ideowy zadań wykonywanych przez sterownik.



Rysunek Schemat ideowy zadań sterownika

**OPISAĆ wątki**

**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.

**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:

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.