Dokumentacja Projektu grupowego

Dokumentacja techniczna projektu

Wydział Elektroniki, Telekomunikacji i Informatyki
Politechnika Gdańska

Nazwa i akronim projektu: *Lądowy dron komunikacyjny 5G*	**Zleceniodawca:** *dr inż. Krzysztof Gierłowski*	
Numer zlecenia: *13@KTIN'2023/23*	**Kierownik projektu:** *Mikołaj Storoniak*	**Opiekun projektu:** *dr inż. Krzysztof Gierłowski*

Nazwa / kod dokumentu: **Dokumentacja techniczna produktu – DTP**	**Nr wersji:** *2.0*
Odpowiedzialny za dokument: *Łukasz Czarzasty*	**Data pierwszego sporządzenia:** *20.01.2024*
	Data ostatniej aktualizacji: *20.01.2024*
	Semestr realizacji Projektu grupowego: *1*

Historia dokumentu

Wersja	Opis modyfikacji	Rozdział / strona	Autor modyfikacji	Data
1.00	*Wstępna wersja*	*całość*	*Karolina Rychert*	*20.02.2023*
2.00	*Dodano schemat modułu komunikacyjnego oraz opisy przypadków testowych*	*2.4 - 2.5*	*Mikołaj Storoniak*	*25.01.2024*

Spis treści

1 Wprowadzenie - o dokumencie

1.1 Cel dokumentu

Celem dokumentu jest udokumentowanie informacji dotyczących produktu, jego cech funkcjonalnych, parametrów technicznych, schematów blokowych, oprogramowania, wyników działania, zdjęć produktu, pomiarów, testów oraz innych elementów wymaganych przez opiekuna i klienta.

1.2 Zakres dokumentu

W chwili obecnej dokument ten zawiera instrukcję uruchomienia wszystkich modułów potrzebnych do realizacji projektu, wraz z całą potrzebną konfiguracją, a także schemat modułu komunikacyjnego.

1.3 Odbiorcy

Członkowie zespołu projektowego: Mikołaj Storoniak, Mateusz Sagan, Karolina Rychert, Łukasz Czarzasty

Katedra Teleinformatyki
Dr inż. Krzysztof Gierłowski

1.4 Terminologia

Moduł komunikacyjny – urządzenie odpowiedzialne za funkcje komunikacji bezprzewodowej i wykorzystujące WiFi, 5G, ZigBee i Bluetooth
Mobilna Platforma – Jeżdżąca platforma sterowana przy pomocy RC, na której zamontowany zostanie moduł komunikacyjny.
Dron / Dron komunikacyjny – Produkt końcowy projektu: moduł komunikacyjny osadzony na mobilnej platformie

2 Dokumentacja techniczna projektu

2.1 Komunikacja między raspberry pi i czujnikiem iNode CS#3 z pomocą Bluetooth LE

2.1.1 Wymagania

W celu odebrania sygnałów przez raspberry pi, należy pobrać oprogramowanie BlueZ. W tym celu należy użyć polecenia

```
sudo apt install bluez
```

2.1.2 Uruchomienie czujnika

W celu uruchomienia czujnika należy otworzyć obudowę i umieścić baterię w przeznaczonym na nie miejsce. Dioda czujnika powinna wtedy zacząć świecić, co oznacza że czujnik został uruchomiony.

2.1.3 Komunikacja między raspberry pi i czujnikiem

Po pobranie BlueZ oraz uruchomieniu czujnika należy przejść do folderu „tools” znajdujący się w folderze pakietu Bluez. Ścieżka nie jest zawsze taka sama więc należy wcześniej zlokalizować folder pakietu. Należy więc użyć polecenia

```
cd /ścieżka_do_tools
```

Następnie należy zresetować interfejs poleceniem

```
sudo hciconfig hci0 down
sudo hciconfig hci0 up
```

Po zresetowaniu interfejsu można rozpocząć komunikację z czujnikiem. W tym celu używa się polecenia

```
sudo hcidump
```

Polecenie to czyta i wyświetla ruch Bluetooth jako zbiór bajtów w postaci szesnastkowej. W celu znalezienia danych z czujnika iNode CS#3 należy zwrócić uwagę na pierwszy bajt w linijce który dla tego czujnika powinien mieć wartość 93. Reszta danych można zinterpretować na podstawie dokumentu ze strony producenta czujnika *https://docs.google.com/document/d/1hcBpZ1RSgHRL6wu4SITq2bvtKSL5_sFjXMu_HRyWZiQ/edit#heading=h.zfwfp5sey5ug.*

2.2 Uruchomienie Waveshare SIM8262E-M2 5G HAT

2.2.1 Wymagania

Do prawidłowego uruchomienia sprzętu i wykorzystania go w celu postawienia hotspota wi-fi wymagane są programy NetworkManager oraz Minicom. Wymagane jest również całkowite wyłączenie programu ModemManager poleceniem*:*

```
sudo killall ModemManager
```

Jego działanie uniemożliwia modemowi prawidłową współpracę z NetworkManagerem.

2.2.2 Hotspot Wi-Fi

W celu utworzenia hotspota bez hasła należy użyć polecenia

```
nmcli connection add \
type wifi \
con-name <nazwa> \
autoconnect no \
wifimode ap \
wifissid <nazwa> \
ipv4.method shared \
ipv6,method shared
```

Z kolei polecenie

```
Nmcli connection up <nazwa>
```

Pozwala go uruchomić.
Aby sprawdzić, czy hotspot został prawidłowo uruchomiony, należy posłużyć się komendą

```
nmcli device
```

Nazwa hotspota powinna znajdować się obok interfejsu wlan0 (domyślnego interfejsu wifi w Raspberry).

2.2.3 Konfiguracja modemu

Po podłączeniu i uruchomieniu, modem udostępni systemowi urządzenia od */dev/ttyUSB0* do */dev/ttyUSB3.* ttyUSB2 to urządzenie, którym możemy się posłużyć w celu wysłania komend do modemu (czasem zdarza się, że zamiast `ttyUSB0,1,2 i 3` pojawią się `ttyUSB0,2,3 oraz 4`; w takim przypadku należy podłączyć się do `ttyUSB3`).
Uczynić to można komendą:

```
sudo minicom –D /dev/ttyUSB2
```

Jeśli urządzenie jest zablokowane, należy upewnić się, że nie korzysta z niego żaden inny program. Po uzyskaniu połączenia wprowadzamy następujące komendy (w tym przypadku dla sieci Play):

```
AT+CRESET
ATZ
ATQ0 V1 E1 S0=0 &C1 &D2
```

```
AT+CFUN=1 //ustawiamy pełną funkcjonalność modemu
AT+CREG=1 +CGREG=1
AT+CGDCONT=1,”IP”,”internet” // internet to nazwa APN sieci PLAY
AT+CGACT=1,1
```

Po wprowadzeniu tych ustawień, modem powinien je zapamiętać, dzięki czemu po resecie nie będzie konieczne ich ponowne wprowadzanie.
Odpowiedni interfejs sieciowy powinien podłączyć się samoistnie dzięki NetworkManagerowi. Aby upewnić się, że tak się stało, możemy ponownie użyć polecenia

```
nmcli device
```

Nowe połączenie oznaczone jest jako „wired connection”.

2.2.4 Dostęp do internetu

Po wykonaniu powyższych czynności, powinniśmy być w stanie podłączyć się do internetu przez hotspot udostępniany przez raspberry pi.

2.3 ZigBee

2.3.1 Wymagania

Do prawidłowego działania zaprezentowanej poniżej konfiguracji na raspberry pi musi być zainstalowany Docker w wersji oraz Docker-Compose. Ten drugi w w wersji 1.25 lub wyższej, aby możliwe było obsłużenie pliku docker-compose w wersji 3.3. Domyślne repozytorium raspberry nie zawiera tych wersji, dlatego konieczne jest dodanie użycie repozytorium dockera.

Wymagane jest także podłączenie do raspberry urządzenia SONOFF Zigbee 3.0 USB dongle plus, a następnie zbadanie jak nazywa się odpowiadający mu plik w katalogu /dev. W naszym przypadku jest to /dev/ttyACM0. Nazwę przyłączonego urządzenia można zbadać przy pomocy komendy dmesg.

2.3.2 Zigbee2Mqtt

Do wysyłania dalej komunikatów odebranych przez drona będziemy używać Zigbee2Mqtt. W celu uruchomienia usługi skorzystamy z Dockera oraz Docker-Compose. Poniżej przedstawiono treść pliku docker-compose.yml:

```
version: ‘3.3’
services:
mqtt:
image: eclipse-mosquitto:2
restart: unless_stopped
volumes:
- „./mosquitto-data:/mosquitto”
ports:
- „1883:1883”
command: „mosquitto –c /mosquitto-no-auth.conf”
zigbee2mqtt:
container_name: zigbee2mqtt
restart: unless_stopped
image: koenkk/zigbee2mqtt:1.30.0
volumes:
- ./zigbee2mqtt-data:/app/data
- /run/udev:/run/udev:ro
ports:
- 8080:8080
 environment:
```

```
- TZ=Europe/Warsaw
devices:
- /dev/ttyACM0:/dev/ttyACM0
```

Kluczowe są sekcje environment/TZ oraz devices w usłudze zigbee2mqtt. Jako TZ należy podać strefę czasową w której będzie pracować dron. W devices należy podać nazwę urządzenia zigbee dongle, zbadaną przy pomocy komendy dmsg.

2.3.3 Konfiguracja

W tej samej lokacji, w której znajduje się plik docker-compose należy utworzyć folder o nazwie „zigbee2mqtt-data". W folderze tym należy utworzyć plik o nazwie „configuration.yaml", zawierający informacje o konfiguracji usługi zigbee2mqtt. Jego treść powinna wyglądać następująco:

```
permit_join: true
mqtt:
base_topic: zigbee2mqtt
server: mqtt://mqtt
serial:
port: /dev/ttyACM0
adapter: ezsp
frontend:
port: 8080
#frontend: false, jeśli nie będzie potrzeby odczytywania wyników
#pomiarów z przeglądarki
advanced:
network_key: GENERATE
```

Kluczowe są następujące sekcje:

- Permit_join, gdzie możemy pozwolić urządzenium na swobodne dołączenie do sieci.
- serial/port, gdzie musimy podać nazwę naszego urządzenia
- serial/adapter, gdzie musimy podać nazwę adaptera którego używa zigbee dongle i który
- zależy od użytego w nim SoC.
- frontend, gdzie można wyłączyć stronę prezentującą wyniki pomiarów.
- Base_topic pozwala określić do jakiego topicu mqtt usługa będzie publikować dane.

2.3.4 Uruchomienie

Aby uruchomić usługę, należy wykonać polecenie docker-compose up. Dzięki zastosowanym opcjom konfiguracji urządzenia będą mogły swobodnie dołączać do sieci i przesyłać wyniki pomiarów.

2.4 Fizyczny projekt urządzenia

Poniższa sekcja przedstawia projekt modułu komunikacyjnego. w jaki sposób elementy powinny zostać ze sobą połączone, w jaki sposób powinny zostać zamontowane na platformie oraz jak powinny zostać zabezpieczone.

2.4.1 Główne założenia przy projektowaniu

Przy projektowaniu kierowano się dwoma głównymi wymaganiami:

- Anteny nie mogą być zasłonięte.
- Pozostałe elementy powinny być zasłonięte.

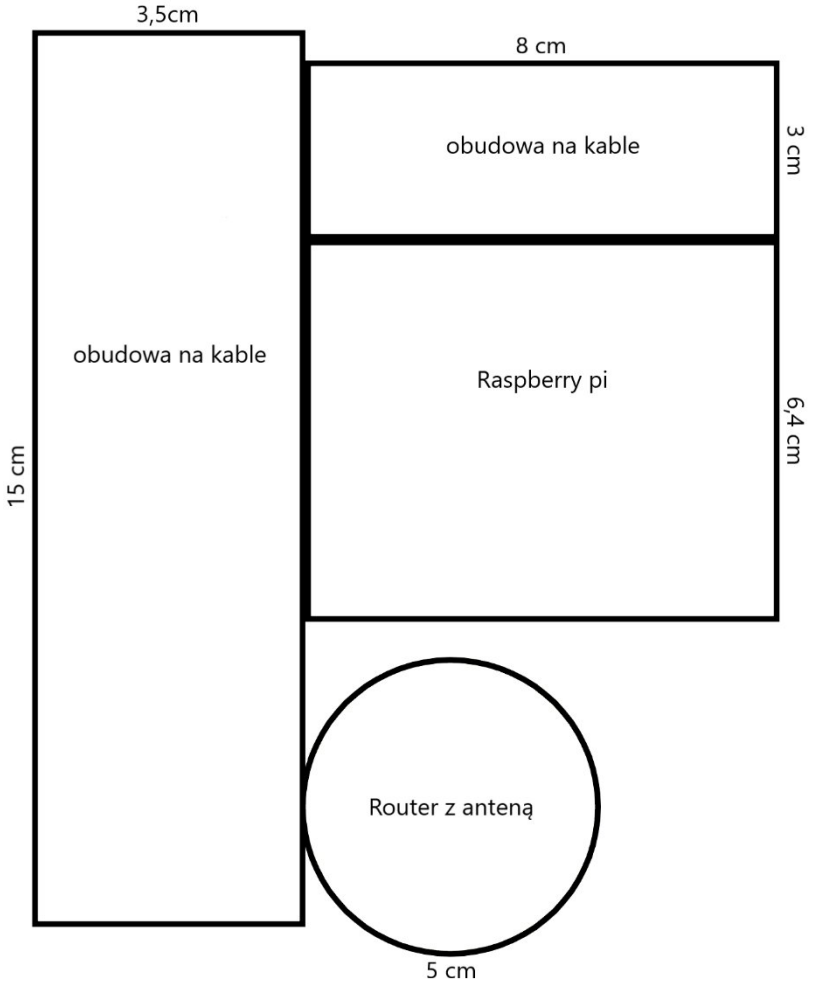
Do osłonięcia i zamocowania elementów zastosowane będą specjalnie zaprojektowane obudowy wydrukowane drukarką 3D przykręcone do platformy.

2.4.2 Elementy

- Raspberry pi
- Router zigbee
- Router wifi
- Antena
- Przedłużacz usb
- Kabel zasilający raspberry pi
- Kabel zasilający router wifi
- Przejściówka do routera wifi

2.4.3 Schemat

Na platformie o kształcie litery X umieszczona zostanie prostokątna platforma o wymiarach 15 cm szerokości,17 cm długości i 2 cm wysokości. Pod nią zostanie umieszczona bateria otoczona prostopadłościenną obudową z otworem na kable zasilające. Na platformie zostaną umieszczone pozostałe elementy Zgodnie z poniższym schematem:



Wszystkie elementy będą znajdować się pod obudowami przykręconymi do platformy.Router będzie osłonięty plastikową drukowaną obudową w kształcie cylindra z dziurą na górze z której będzie wystawać antena oraz spłaszczonym boku łączącym się z obudową na kable. obudowa Raspberry pi powinna być prostopadłościanem z otworami na kable i anteny. Anteny znajdują się na prawej i przedniej stronie raspberry pi. Mniejsza obudowa na kable jest niższa niż obudowa do raspberry pi tak żeby znajdowała się poniżej anten. Będzie się ona też łączyć z większą obudową na kable. Większa obudowa na kable będzie prostopadłościanem z

otworami na kable z raspberry pi, routera oraz mniejszej obudowy na kable. Będzie to miejsce przez które poprowadzone będą kable łączące urządzenia umieszczone na platformie. na obszarze zakrytym przez tą obudowę znajdować się będzie dziura w podstawie przez którą przechodzić będą kable zasilania. Na boku obudowy będzie się też znajdować otwór z którego wystawać będzie antena routera zigbee. Do raspberry pi będzie podłączony przedłużacz usb. Po drugiej stronie przedłużacza będzie podłączony router zigbee.

2.5. Testy

Urządzenie zostanie poddane szeregowi testów, mających sprawdzić jego zachowanie w wyjątkowych warunkach, których wystąpienie jest jednak dość prawdopodobne. Testy te zostały opracowane na stosunkowo wczesnym etapie projektu, aby już przy projektowaniu systemu możliwe było uwzględnienie opisanych w ich ramach przypadków.

2.5.1 Błędy crc

Test1

- Utworzenie bezpiecznego, możliwego do obserwowania, testowego środowiska
- Rozpocząć normalną komunikację między urządzeniami
- Wprowadzić sztuczne błędy do transmisji, modyfikując pojedyncze bity w pakietach danych. Można to zrobić za pomocą narzędzi do manipulacji danymi lub odpowiedniego oprogramowania.
- Monitorować ruch sieciowy i analizować, czy router jest w stanie wykrywać i obsługiwać błędy CRC. Należy skoncentrować się na logach routera, aby sprawdzić, czy rejestruje wykryte błędy.
  Kontynuacją tego testu byłoby stopniowe zwiększenie ilości błędów i obserwowanie jak będzie sobie radzić z coraz większą ich ilością. Następnie należy określić, na jakim poziomie intensywności błędów urządzenie przestaje skutecznie obsługiwać transmisję danych. To pomoże określić granicę odporności.

Test 2

- Utworzenie bezpiecznego, możliwego do obserwowania, testowego środowiska
- Rozpocząć normalną komunikację między urządzeniami
- Wprowadzić zakłócenia elektromagnetyczne, na przykład poprzez użycie urządzeń generujących pola elektromagnetyczne. Monitorować, jak router reaguje na te zakłócenia.
- Analizować logi i obserwować, czy urządzenie jest w stanie utrzymać komunikację w obecności zakłóceń elektromagnetycznych. Sprawdzić, czy występuje zwiększona liczba błędów CRC.

2.5.2 Cross protocol testing

- Skonfigurować kontrolowane środowisko testowe z routerem lub mostkiem obsługującym co najmniej dwa różne protokoły (np. Zigbee i Wi-Fi).
- Sparować urządzenia korzystające z różnych protokołów z routerem lub mostkiem, następnie upewnić się, że urządzenia są poprawnie skonfigurowane i zdolne do wymiany danych.
- Rozpocząć wymianę danych między urządzeniami korzystającymi z różnych protokołów. Potencjonalnymi możliwościami jest przesyłanie komend, danych pomiarowych lub innych informacji i monitorowanie, czy urządzenie poprawnie przekazuje je między protokołami.
- Sprawdzić czy router lub mostek skutecznie tłumaczy i przekazuje dane.
- Zweryfikować integralność danych po przejściu przez różne protokoły i Upewnić się, że dane są poprawnie odbierane i interpretowane przez urządzenia docelowe.

2.5.3 Działanie w warunkach słabego sygnału

Przed wprowadzeniem zakłócenia należy zmierzyć bazową siłę sygnału połączenia z Raspberry Pi do routera lub mostka. Można użyć narzędzia ‘iwconfig’ lub czegoś podobnego do tego celu. Można symulować słabe warunki sygnału poprzez wprowadzenie zakłóceń lub fizyczne przeniesienie Raspberry Pi w obszar, gdzie sygnał jest słaby. Konieczne jest ciągłe monitorowanie siły sygnału. Podczas monitorowania siły sygnału, należy przetestować łączność Raspberry Pi z siecią. Można to zrobić, próbując pingować inne urządzenia w sieci lub dostępu do zewnętrznych stron internetowych.

2.5.4 Przekroczenie długości ramki

- Skonfigurowanie Raspberry Pi jako router lub mostek, zgodnie z twoimi wymaganiami
- Skonfigurowanie komputer testowy jako nadawcę lub odbiorcę ruchu sieciowego w twoim teście. Upewnienie się, że jest on połączony z Raspberry Pi.
- Na komputerze testowym należy uruchomić narzędzie do generowania ruchu sieciowego, na przykład ‘iperf’.
- Skonfigurowanie narzędzie tak, aby generowało pakiety sieciowe o niestandardowej długości ramki, która przekracza standardowe ograniczenia.
- Przesłać ruch sieciowy z komputera testowego do Raspberry Pi, który będzie działał jako router lub mostek.
- Na Raspberry Pi użyć narzędzi do monitorowania ruchu sieciowego, takich jak ‘tcpdump’, aby zarejestrować przesyłane pakiety.
- Sprawdzić, czy Raspberry Pi przetwarza pakiety o niestandardowej długości ramki, czy też je odrzuca.

2.5.5 Ilość danych która może przejść jednocześnie

Test podobny jak w przypadku pierwszego, można do tego wykorzystać ‘iperf’

2.5.6 Efektywność energetyczna

- Należy wybrać możliwe scenariusze pracy, np.: bezczynność, działanie jako router z obciążeniem sieciowym, działanie jako mostek bez obciążenia, etc.
- Określić czas trwania każdego scenariusza testowego oraz wszelkie inne czynniki, które mogą mieć wpływ na zużycie energii, takie jak aktywność użytkownika, rodzaj wykorzystywanej aplikacji, itp.
- Podłączyć miernik energii do zasilania Raspberry Pi i zanotować początkowe zużycie energii w trybie bezczynności.
- Eksperymentować ze wszystkimi scenariuszami wraz z modyfikowaniem ustawień systemowych oraz aplikacji.