

Streszczenie Pracy nad Implementacją Modułów LoRa

Wprowadzenie

Praca nad projektem obejmowała szczegółową analizę i implementację modułów LoRa w celu stworzenia systemu komunikacji radiowej na platformie Raspberry Pi 4, zamontowanej na dronie. Celem projektu było umożliwienie przesyłania danych na duże odległości, przy zachowaniu kompaktowych wymiarów i niezawodności transmisji.

Pierwszym etapem było skonfigurowanie dostępnego modułu LoRa ****SX1278 (433 MHz)****, jednak po serii testów i napotkaniu problemów komunikacyjnych podjęto decyzję o analizie alternatywnych modułów LoRa, co doprowadziło do wyboru i implementacji modułu ****E220-900T30D (868 MHz)****. Poniżej opisano proces pracy, kamienie milowe, wyzwania i osiągnięte rezultaty.

Kamienie Milowe i Proces Pracy

1. Rozpoczęcie pracy z modułem LoRa SX1278 (433 MHz)

1.1 Podłączenie i konfiguracja sprzętowa

Pierwszym etapem było podłączenie modułu LoRa SX1278 do Raspberry Pi 4 oraz komputera przy użyciu konwertera USB-UART CP2102. Zgodnie z dokumentacją modułu wykonano:

- Podłączenie zasilania modułu do ****3.3V****.
- Połączenie pinów komunikacyjnych ****MOSI, MISO, SCK i NSS**** z odpowiednimi pinami Raspberry Pi.

- Włączenie monitora portu szeregowego na komputerze w celu wysyłania i odbierania komend AT.

1.2 Próby konfiguracji

Przeprowadzono konfigurację modułu, wykorzystując standardowe parametry UART:

- Prędkość transmisji: ****9600 baud****.
- Parzystość: brak (8N1).
- Zakończenie poleceń: ****CR+LF**** (znak powrotu karetki i nowej linii).

Wysyłano podstawowe komendy AT, takie jak:

- `AT` – w celu przetestowania połączenia.
- `AT+FREQ=433000000` – ustawienie częstotliwości pracy modułu.
- `AT+RX` i `AT+TX` – weryfikacja funkcji odbioru i transmisji danych.

1.3 Problemy i wyzwania

Mimo prawidłowego podłączenia i wysyłania komend moduł SX1278 nie reagował zgodnie z oczekiwaniami. Napotkane trudności obejmowały:

- Brak odpowiedzi na komendy AT.
- Niewłaściwa stabilność zasilania przy zmianie między ****3.3V a 5V****.
- Możliwa niezgodność oprogramowania modułu z dokumentacją.

W efekcie zdecydowano się na analizę alternatywnych rozwiązań.

2. Analiza alternatywnych modułów LoRa

W odpowiedzi na problemy z SX1278 rozpoczęto poszukiwania innego modułu LoRa, który spełniałby wymagania projektu. Kluczowe kryteria obejmowały:

- Prosta konfiguracja i niezawodność komunikacji.
- Zgodność z Raspberry Pi.
- Duży zasięg transmisji danych (do 10 km).
- Kompaktowe wymiary i wsparcie dla pasma ISM (868 MHz).

2.1 Moduł E220-900T30D (868 MHz)

Wybrano moduł **E220-900T30D**, który spełniał powyższe wymagania. Jego główne cechy to:

- **Interfejs UART**, ułatwiający konfigurację.
- Zasięg do 10 km dzięki wysokiej mocy wyjściowej (1W).
- Zasilanie 3.3V – 5V, kompatybilne z Raspberry Pi.
- Stabilność w pasmach 868 MHz, co jest zgodne z europejskimi normami.

3. Implementacja i testy modułu E220-900T30D

3.1 Podłączenie sprzętowe

Moduł E220-900T30D podłączono zgodnie z poniższą konfiguracją:

Pin LoRa	**Pin Raspberry Pi GPIO**	**Uwagi**
-----	-----	-----
VCC	5V (Pin 2)	Zasilanie modułu.
GND	GND (Pin 6)	Masa wspólna.
TX	RX (GPIO 15, Pin 10)	Odbiór danych z LoRa.
RX	TX (GPIO 14, Pin 8)	Wysyłanie danych do LoRa.
AUX	GPIO dowolny	Monitorowanie statusu.
M0/M1	GND	Ustawienie trybu pracy.
Pin LoRa	**Pin Raspberry Pi GPIO**	**Uwagi**
-----	-----	-----
VCC	5V (Pin 2)	Zasilanie modułu.
GND	GND (Pin 6)	Masa wspólna.
TX	RX (GPIO 15, Pin 10)	Odbiór danych z LoRa.
RX	TX (GPIO 14, Pin 8)	Wysyłanie danych do LoRa.
AUX	GPIO dowolny	Monitorowanie statusu.
M0/M1	GND	Ustawienie trybu pracy.
Pin LoRa	**Pin Raspberry Pi GPIO**	**Uwagi**
-----	-----	-----
VCC	5V (Pin 2)	Zasilanie modułu.
GND	GND (Pin 6)	Masa wspólna.
TX	RX (GPIO 15, Pin 10)	Odbiór danych z LoRa.
RX	TX (GPIO 14, Pin 8)	Wysyłanie danych do LoRa.
AUX	GPIO dowolny	Monitorowanie statusu.
M0/M1	GND	Ustawienie trybu pracy.

Pin LoRa	**Pin Raspberry Pi GPIO**	**Uwagi**
-----	-----	-----
VCC	5V (Pin 2)	Zasilanie modułu.
GND	GND (Pin 6)	Masa wspólna.
TX	RX (GPIO 15, Pin 10)	Odbiór danych z LoRa.
RX	TX (GPIO 14, Pin 8)	Wysyłanie danych do LoRa.
AUX	GPIO dowolny	Monitorowanie statusu.
M0/M1	GND	Ustawienie trybu pracy.
Pin LoRa	**Pin Raspberry Pi GPIO**	**Uwagi**
-----	-----	-----
VCC	5V (Pin 2)	Zasilanie modułu.
GND	GND (Pin 6)	Masa wspólna.
TX	RX (GPIO 15, Pin 10)	Odbiór danych z LoRa.
RX	TX (GPIO 14, Pin 8)	Wysyłanie danych do LoRa.
AUX	GPIO dowolny	Monitorowanie statusu.
M0/M1	GND	Ustawienie trybu pracy.
Pin LoRa	**Pin Raspberry Pi GPIO**	**Uwagi**
-----	-----	-----
VCC	5V (Pin 2)	Zasilanie modułu.
GND	GND (Pin 6)	Masa wspólna.
TX	RX (GPIO 15, Pin 10)	Odbiór danych z LoRa.
RX	TX (GPIO 14, Pin 8)	Wysyłanie danych do LoRa.
AUX	GPIO dowolny	Monitorowanie statusu.
M0/M1	GND	Ustawienie trybu pracy.
Pin LoRa	**Pin Raspberry Pi GPIO**	**Uwagi**
-----	-----	-----

VCC	5V (Pin 2)	Zasilanie modułu.
GND	GND (Pin 6)	Masa wspólna.
TX	RX (GPIO 15, Pin 10)	Odbiór danych z LoRa.
RX	TX (GPIO 14, Pin 8)	Wysyłanie danych do LoRa.
AUX	GPIO dowolny	Monitorowanie statusu.
M0/M1	GND	Ustawienie trybu pracy.
Pin LoRa	**Pin Raspberry Pi GPIO**	**Uwagi**
-----	-----	-----
VCC	5V (Pin 2)	Zasilanie modułu.
GND	GND (Pin 6)	Masa wspólna.
TX	RX (GPIO 15, Pin 10)	Odbiór danych z LoRa.
RX	TX (GPIO 14, Pin 8)	Wysyłanie danych do LoRa.
AUX	GPIO dowolny	Monitorowanie statusu.
M0/M1	GND	Ustawienie trybu pracy.

3.2 Testy komunikacji

Przy użyciu poniższego skryptu w Pythonie sprawdzono poprawność konfiguracji UART oraz reakcje modułu na komendy AT:

```

``python

import serial

import time

ser = serial.Serial("/dev/serial0", baudrate=9600, timeout=1)

def send_command(command):

```

```
ser.write(f"{command}"  
  
".encode()")  
  
response = ser.readline().decode().strip()  
  
print(f"Wysłano: {command}, Otrzymano: {response}")
```

try:

```
    send_command("AT")
```

finally:

```
    ser.close()
```

...

Wyniki testów potwierdziły prawidłowe działanie modułu. Moduł E220-900T30D reagował na komendy AT i umożliwiał konfigurację częstotliwości, mocy wyjściowej oraz innych parametrów.

3.3 Optymalizacja komunikacji

Dla zapewnienia maksymalnej stabilności transmisji w projekcie dokonano:

- Testów różnych anten.
- Dostosowania ustawień prędkości transmisji UART.
- Monitorowania statusu modułu za pomocą pinu AUX.

Rezultaty i dalsze kroki

Rezultaty:

1. Zidentyfikowano problemy z modułem SX1278, co umożliwiło podjęcie decyzji o zmianie technologii.
2. Implementacja modułu E220-900T30D zakończyła się sukcesem, zapewniając stabilną komunikację na dużą odległość.

Dalsze kroki:

1. Integracja modułu E220-900T30D z pełnym systemem drona.
2. Rozbudowa protokołów komunikacji o dodatkowe funkcje (np. przesyłanie danych telemetrycznych).
3. Optymalizacja układu zasilania i testy w rzeczywistych warunkach.

Podsumowanie

Przeprowadzone działania obejmowały analizę, testy i implementację dwóch modułów LoRa. W trakcie pracy napotkano liczne wyzwania związane z konfiguracją i testowaniem modułu SX1278, co doprowadziło do wyboru bardziej niezawodnego rozwiązania w postaci E220-900T30D. Dzięki przeprowadzonym testom i optymalizacjom osiągnięto stabilną i efektywną komunikację, spełniającą założenia projektu.