

Kurs:  
Urządzenia Peryferyjne - Laboratorium

---

# MODEMY- TRANSMISJA SYGNAŁÓW CYFROWYCH

---

*Autor:*

ALEKSANDRA CHRUSTEK, 263900

JONASZ TROCHA, 263951

Prowadzący:  
dr inż. Dariusz Caban

10.01.2024

# 1 Wstęp

Celem ćwiczenia było zapoznanie się z zasadami działania modemów oraz napisanie oprogramowania do obsługi komunikacji między modemami w sposób terminal-terminal.

## 2 Informacje teoretyczne

### 2.1 Zasady i typy modulacji

#### 2.1.1 AM (Amplituda Modulowana)

Sygnał nośny amplitudy jest modyfikowany w zależności od amplitudy sygnału źródłowego.

#### 2.1.2 FSK (Frequency Shift Keying)

Zmiana częstotliwości nośnej zgodnie z danymi wejściowymi.

#### 2.1.3 PSK (Phase Shift Keying)

Zmiana fazy nośnej w zależności od danych wejściowych.

#### 2.1.4 QAM (Quadrature Amplitude Modulation)

Kombinacja modulacji amplitudy i fazy.

### 2.2 Pojęcia podstawowe

#### 2.2.1 Bd (Baud)

Jednostka oznaczająca liczbę zmian stanu sygnału na sekundę, wyrażana w baudach.

#### 2.2.2 Bit

Najmniejsza jednostka informacji, reprezentująca wartość 0 lub 1.

#### 2.2.3 Hz (Hertz)

Jednostka częstotliwości, oznaczająca ilość cykli na sekundę.

## **2.3 Standardy modulacji/kodowania**

### **2.3.1 Ethernet**

- 4b5b: Kodowanie 4 bity danych w 5-bitowe kody zapewniające lepszą synchronizację.
- MLT3 (Multi-Level Transition 3): Wykorzystywane do kodowania sygnałów dla transmisji na skrętce.
- 8b6T: Kodowanie 8 bitów danych w 6-bitowe symbole, stosowane w 1000BASE-T.

### **2.3.2 WiFi**

QAM Modulacja amplitudy i fazy, często używana w sieciach bezprzewodowych do przesyłania danych.

### **2.3.3 CD-DVD**

EFM (Eight-to-Fourteen Modulation) Sposób kodowania danych na płytach CD i DVD dla lepszej detekcji i korekcji błędów.

## **2.4 Zasady realizacji transmisji**

### **2.4.1 Metody konfiguracji modemów**

### **2.4.2 Komendy Hayesa**

### **2.4.3 Tryby pracy/diagram stanów modemu**

- Command/Data Mode: Modem może działać w dwóch trybach, komendowym lub danych.
- Digit/Word Mode: Określa, czy dane są przesyłane jako pojedyncze cyfry czy całe słowa.
- Sekwencja escape (ESC): Pozwala na przesyłanie znaków specjalnych.

### 3 Realizacja zadania

Komendy Hayes to zestaw standardowych komend kontrolujących używanych do konfiguracji i zarządzania modemami. Nazwa pochodzi od firmy Hayes Microcomputer Products, która była jednym z pionierów rynku modemów w latach 80. i 90. Komendy Hayes stały się powszechnie stosowane i znormalizowane, co umożliwiło interoperacyjność między różnymi modemami i systemami.

Oto kilka kluczowych komend Hayes:

1. AT (Attention): Ta komenda inicjuje komunikację z modemem. Gdy terminal wysła "AT", modem powinien odpowiedzieć "OK".
2. ATD (Dial): Używana do nawiązywania połączeń telefonicznych. Po ATD wprowadza się numer, np. "ATD3965".
3. ATA (Answer): Modem przechodzi z trybu komendowego do trybu danych, aby odebrać połączenie.
4. ATH (Hang-up): Zamyka aktualne połączenie.
5. ATZ (Reset): Przywraca ustawienia fabryczne modemu.
6. +++: pełni specjalną rolę i jest znana jako "escape sequence" lub "attention sequence" w ramach komend Hayes. Gdy komunikacja między komputerem a modemem jest w trybie przesyłania danych (Data Mode), wysłanie sekwencji "+++" powoduje przejście modemu z trybu danych do trybu komendowego (Command Mode).

Poniżej przedstawiony został kod aplikacji do obsługi modemu:

```
import serial
import time
dataToWrite = serial.Serial("COM1", 9600, parity="N",
stopbits=1, bytesize=8, xonxoff=True, timeout=5)

def menu():
    print("""
Opcje:
    o - ODBIERZ
    z - ZAKONCZ
    p - ZADZWON POD NUMER 3965
    m - wyslij wiadomosc
    c - Odczytaj konsole
```

```

        k - zamknij port i zakoncz program
        s - MENU
    """)

menu()

while True:

    userSelect = input("Wybierz co chcesz zrobic: ")

    if userSelect == 'o':
        dataToWrite.write("ata\r".encode())

    if userSelect == 'z':
        dataToWrite.write("+++\r".encode())
        dataToWrite.write("ath\r".encode())

    if userSelect == 'k':
        dataToWrite.close()
        break

    if userSelect == 'p':
        dataToWrite.write("atd3965\r".encode())

    if userSelect == 'm':
        mg = input("Napisz wiadomosc: ")
        dataToWrite.write("{}\r".format(mg).encode())

    if userSelect == 's':
        menu()

    if userSelect == 'c':
        time.sleep(1)
        try:
            print(dataToWrite.read(dataToWrite.in_waiting).decode("utf-8"))
        except UnicodeDecodeError:
            print('###')
```

## 4 Wnioski

Zadanie miało na celu wprowadzenie w obszar programowania urządzeń peryferyjnych, konkretnie modemów. Realizacja tego zadania okazała się wymagająca, ponieważ wiązała się z koniecznością zaznajomienia się z zasadami działania modemu oraz jak obsługa modemu wygląda z perspektywy zaimplementowanej aplikacji. W trakcie laboratorium udało się zaimplementować wszystkie wymagane przez instrukcję metody, z wyjątkiem przesyłania plików. W rezultacie, zadanie dostarczyło nam praktycznej wiedzy na temat programowania urządzeń peryferyjnych, a zdobyte umiejętności mogą być użyteczne w projektach obejmujących obsługę różnorodnych urządzeń wejściowych.