

<p style="text-align: center;">Urządzenia peryferyjne</p> <p style="text-align: center;">Laboratorium 2 – Modemy</p>	
<p style="text-align: center;">Michał Lewandowski, Dominik Kilijan</p> <p style="text-align: center;">Grupa F</p>	<p style="text-align: center;">Czwartek 17:30 – 19:30 TNP</p> <p style="text-align: center;">12.10.2023</p>

1. Wstęp teoretyczny

1.1 Podstawowe rodzaje modulacji:

- AM – kodowaniu sygnału informacyjnego w chwilowych zmianach amplitudy sygnału nośnego
- FM – kodowanie informacji w fali nośnej przez zmiany jej chwilowej częstotliwości w zależności od sygnału wejściowego
- PM – kodowanie informacji w fali nośnej przez zmianę jej chwilowej fazy

1.2 Standardy modulacji:

- ASK – amplitudowe
- QAM – amplitudowo-fazowe
- PSK – kluczowanie fazy
- FSK – kluczowanie częstotliwości (przy logicznym zerze mniejsza częstotliwość i na odwrót)

1.3 Pojęcia podstawowe

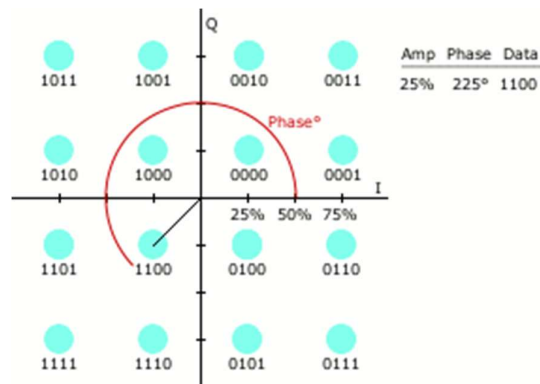
- Bit – jednostka danych (0 lub 1)
- Hz – herc, jednostka częstotliwości
- Bd - baud, prędkość transmisji w danym czasie, ile symboli można przesłać na sekundę

1.4 Standardy modulacji/kodowania Ethernet

- **4B5B** – system kodowania danych cyfrowych, w którym grupa czterech bitów jest zamieniana na 5 bitów, tak by w wyjściowym ciągu 5 bitów bit o wartości 1 występował przynajmniej dwa razy. Zastosowany w 100BASE-TX
- **MLT-3** (ang. Multi-Level Transmit) – system liniowego kodowania danych cyfrowych w medium transmisyjnym wykorzystujący do kodowania 3 poziomy sygnał. Stosowany jest między innymi w standardzie sieci Ethernet 100Base-T. Zastosowanie 3 poziomów sygnału w porównaniu z systemami opartymi na 2 poziomach sprawia, że tak kodowany sygnał ma mniejszą częstotliwość przez co,

emituje mniej zakłóceń elektromagnetycznych i wymaga mniejszej przepustowości łącza.

- **8B/6T** przebiega w ten sposób, że każdej sekwencji ośmiu bitów ze strumienia danych wejściowych przyporządkowany zostaje ciąg sześciu symboli trzystanowych (o trzech możliwych poziomach napięć: $-V$, 0 , $+V$). Możliwych jest więc $3^6 = 729$ ciągów, z czego wykorzystywanych jest $2^8 = 256$ ciągów. Ciągi kodowe zostały tak dobrane, aby zapewnić możliwość dobrej detekcji błędów, zmniejszyć efekty wysokoczęstotliwościowe oraz wyeliminować składową stałą. kategorii 3 do transmisji sygnału 100Mb/s
- **8b/10b** to kod liniowy, który odwzorowuje 8-bitowe słowa na 10-bitowe symbole w celu osiągnięcia równowagi DC i ograniczonej rozbieżności, a jednocześnie zapewnia wystarczające zmiany stanu, aby umożliwić rozsądne przywrócenie zegara. Oznacza to, że różnica między liczbą jedynek i zer w ciągu co najmniej 20 bitów nie jest większa niż dwa i że w rzędzie nie ma więcej niż pięć jedynek lub zer. Pomaga to zmniejszyć zapotrzebowanie na dolną granicę pasma kanału niezbędną do przesłania sygnału. Standard 802.3z (1000Base-LX, 1000Base-SX, 1000Base-CX), 1000Base-T (802.3ab)
- **WiFi** (Quadrature Amplitude Modulation) kwadraturowa modulacja amplitudowo-fazowa służąca do przesyłania danych cyfrowych przez kanał radiowy, stosowana m.in. w transmisjach DVB. Modulacja QAM jest połączeniem modulacji amplitudy i modulacji fazy. Dane formowane są w dwójki, trójki, czwórki itd.,



Rys 1. Diagram konstelacji 16-QAM z przykładowymi punktami konstelacji

- **EFM** (akronim ang. Eight-to-fourteen modulation - modulacja „8-w-14”) – modulacja danych stosowana na płytach kompaktowych (CD- DVD). Według zasad EFM proces konwersji danych rozpoczyna się na podzielenie strumienia danych na bloki 8-bitowe (bajty). Następnie każdy z 8-bitowych bloków jest tłumaczony na odpowiadający mu kod 14-bitowy. Przekształcenie to jest wyznaczone przez stabilizowanie.

1.5 Zasady realizacji transmisji:

Warstwa Fizyczna (Warstwa 1):

- Zasada Przesyłania Bitów: Warstwa fizyczna jest odpowiedzialna za przesyłanie bitów przez medium transmisyjne. Określa fizyczne cechy medium, takie jak typ kabla czy częstotliwość fal radiowych.
- Zasada Szybkości i Sposobu Transmisji: Określa szybkość transmisji danych (np. 100 Mbps, 1 Gbps) oraz rodzaj modulacji i kodowania sygnału.
- Zasada Topologii Sieci: Określa, jak urządzenia są połączone w sieci, czyli czy jest to topologia gwiazdy, pierścienia czy magistrali.

Warstwa Łączy Danych (Warstwa 2):

- Zasada Frame'ów (Ramki): Dane przesyłane są w formie ramek, które zawierają nagłówki z adresami MAC (Media Access Control) źródła i celu oraz dane.
- Zasada Kontroli Błędów: Wykorzystuje techniki takie jak CRC (Cyclic Redundancy Check) do wykrywania i ewentualnej korekcji błędów w przesyłanych ramach.
- Zasada Kontroli Dostępu do Medium: Protokoły takie jak CSMA/CD (Carrier Sense Multiple Access with Collision Detection) kontrolują dostęp do medium w przypadku, gdy wiele urządzeń próbuje przesłać dane jednocześnie.

Warstwa Sieci (Warstwa 3):

- Zasada Pakietów: Dane są podzielone na pakiety, które zawierają nagłówki z adresem źródłowym i docelowym IP oraz dane.
- Zasada Routingu: Określa, jak pakiety są przesyłane przez różne urządzenia (routery) w sieci w celu dotarcia do ich celu.

Warstwa Transportowa (Warstwa 4):

- Zasada Segmentacji i Reasemblacji: Długie strumienie danych są dzielone na mniejsze jednostki (segmenty), które są przesyłane przez sieć, a następnie ponownie łączone na końcowym urządzeniu docelowym.
- Zasada Kontroli Błędów i Flow Control: Zapewnia integralność danych przez stosowanie mechanizmów kontroli błędów oraz reguluje przepływ danych, aby uniknąć przeciążenia nadmierną ilością danych.

Warstwa Sesji (Warstwa 5), Warstwa Prezentacji (Warstwa 6) i Warstwa Aplikacji (Warstwa 7):

- W warstwach sesji, prezentacji i aplikacji zasady dotyczą bardziej wysokopoziomowych aspektów komunikacji, takich jak ustanawianie, utrzymanie i zamykanie sesji, konwersja formatów danych czy interakcja z aplikacjami użytkownika. W tych warstwach komunikacja odbywa się już w formie konkretnych komunikatów czy żądań aplikacyjnych.

1.6 Metody konfiguracji modemów

- Dla Połączenia DSL (linia telefoniczna):
Ustawienia typu połączenia: PPPoE (Point-to-Point Protocol over Ethernet).
Wprowadź nazwę użytkownika (login) i hasło dostępowe przekazane przez dostawcę usług.
- Dla Połączenia Kablowego:
Zazwyczaj modem kablowy automatycznie pobiera konfigurację od dostawcy. W większości przypadków nie wymaga dodatkowej konfiguracji.
- Dla Połączenia Bezprzewodowego (Wi-Fi):
Skonfiguruj unikalną nazwę sieci (SSID) oraz hasło dostępu (klucz WPA/WPA2) dla sieci bezprzewodowej

1.7 Najważniejsze komendy Hayes'a:

ATA: odbieranie połączenia przychodzącego (w konsolach obu uczestników rozmowy pojawia się CONNECT)

ATD<numer>: Wybiera numer telefonu, gdzie <numer> to numer telefonu, który chcesz wybrać

ATH: zakończenie połączenia (u obu użytkowników pokazuje się NO CARRIER)

ATSr=n przypisanie do rejestru wybranej wartości

ATO: powrót z trybu komend podczas połączenia

+++: wejście do trybu komend (polega to na możliwości pisania w swojej konsoli zamiast tak jak zwykle, przeciwnika)

ATZ: powraca do ustawień początkowych modemu,

AT&V: wypisuje konfigurację,

ATS2?: Wypisuje znak wyjścia za pomocą kodu ASCII, w naszym przypadku '043', czyli '+'. W przypadku podania ATS2=45 można było podczas aktywnego połączenia przejść w tryb komend podając znaki „---”, zamiast domyślnych „+++”

1.8 Tryby pracy modemu:

Command Mode (Komendy AT): W tym trybie modem oczekuje na komendy AT od komputera.

Data Mode (Tryb Danych): W tym trybie modem przesyła dane między komputerem a siecią, na przykład podczas ustanawiania połączenia internetowego.

Online Command Mode: Jest to mieszany tryb, który pozwala na zarówno wysyłanie komend AT, jak i przesyłanie danych

1.9 Diagram stanów modemu:

Start/Idle (Początek/Bezczynność): Modem oczekuje na komendy AT lub na próbę połączenia.

Command State (Stan Komend): Modem odbiera i interpretuje komendy AT od komputera.

Data State (Stan Danych): Modem przesyła dane między komputerem a siecią.

Online Command Mode (Tryb Komend Online): Modem pozwala na odbieranie komend AT, ale nadal utrzymuje połączenie danych.

Error State (Stan Błędu): Modem przechodzi do tego stanu w przypadku błędu podczas przetwarzania komend lub danych.

Sekwencja Escape (ESC): pozwala przejść z trybu danych do trybu komend AT bez rozłączania połączenia. Po wysłaniu ESC, modem wraca do trybu komend AT, pozwalając na wysłanie dodatkowych komend bez przerywania połączenia danych.

2. Przebieg zadania

W pierwszej kolejności do podłączonego modemu połączono się przy pomocy programu PuTTY. Z początku występował problem z pisaniem w programie, ale ostatecznie udało się połączyć z drugim modemem korzystając z komend Hayes'a. Połączenie działało prawidłowo, wiadomości pisane na jednym komputerze, wyświetlane były na drugim. Obie strony mogły być zarówno nadawcą jak i odbiorcą. Następnym etapem zajęć było napisanie programu zawierającego podstawowe metody związane z połączeniem modemów. Aplikacja konsolowa została napisana w języku Python. Najistotniejszą biblioteką wykorzystaną przy pisaniu programu było pySerial związane z obsługą portu szeregowego. Ponadto wykorzystana została także biblioteka threading do obsługi wątków. Napisany kod działał poprawnie i spełniał założone mu cele.

3. Kod programu

```
import serial
import threading

def read_from_port(port):
    while True:
        data = port.read_until('\r'.encode('utf-8'))
        try:
            data = data.decode('utf-8')
            if data == '':
                continue
            print(data)
        except:
            continue

modem_port = serial.Serial("COM1", 9600, parity='N', stopbits=1,
bytesize=8, xonxoff=True, timeout=5)
print('Port opened!\n')
t = threading.Thread(target=read_from_port, daemon=True,
args=(modem_port,))
print('Made a listening thread!')
t.start()
while True:
    chose = input("Wpisz liczbe od 0 do 4\n"
                  "0 : Odbierz polaczenie\n"
                  "1 : Polaczenie z wpisanym numerem\n"
                  "2 : Wysluj komende AT\n"
                  "3 : Zakoncz polaczenie\n"
                  "4 : Zamknij port\n")

    if chose == "0":
        modem_port.write("ata\r".encode())
    elif chose == "1":
        number = input("Wpisz numer z ktorym chcesz sie polaczyc\n")
        message = number + '\r'
        print(f'Sending atd{number} ...')
        modem_port.write(message.encode())
    elif chose == "2":
        message = input("Wpisz komende AT\n")
        msg = message + '\r'
        print(f'Sending: {msg}')
        modem_port.write(msg.encode())
    elif chose == "3":
        print('Declining connection')
        modem_port.write("+++ath\r".encode())
    elif chose == "4":
        modem_port.close()
```

4. Wnioski

Laboratorium pozwoliło na zapoznanie z mechanizmami umożliwiającymi działanie modemów w teorii jak i w praktyce. Lepiej zrozumiane zostało także działanie serial portów i parametrów z nimi związanymi. Poszerzona została wiedza z programowania w języku python, szczególnie poprzez wykorzystanie biblioteki pySerial, z którą nie mieliśmy wcześniej kontaktu.

5. Bibliografia

<https://pyserial.readthedocs.io/en/latest/index.html>

<https://pl.wikipedia.org/wiki/Modulacja>

https://en.wikipedia.org/wiki/Hayes_AT_command_set

<https://learn.microsoft.com/pl-pl/dotnet/api/system.io.ports.serialport?view=dotnet-plat-ext-7.0>