

Urządzenia peryferyjne laboratorium	
Temat:	Modemy
Data zajęć:	01.12.2021
Grupa zajęciowa:	TN środa 11:15
Członkowie zespołu:	Aleksandra Ciężka 252701 Mateusz Chalik 252735
Grupa	C
Prowadzący:	dr inż. Tomasz Walkowiak

1. Wstęp

Celem ćwiczenia było zapoznanie się z obsługą modemów oraz komendami używanymi do ich sterowania.

2. Realizacja zadania

2.1 Połączenie przy pomocy Putty

Realizację ćwiczenia rozpoczęto od próby połączenia z modemami przy wykorzystaniu Putty. Przy komunikacji wykorzystano podane poniżej komendy Hayes'a.

Wykorzystane komendy Hayes'a podczas próbie komunikacji przy pomocy Putty:

- AT – sprawdzenie łączności między komputerem i modemem,
- ATA – odebranie połączenia,
- ATDT – wybranie numeru podanego po komendzie,
- ATH0 – odłożenie słuchawki
- +++ - przełączenie w tryb komend.

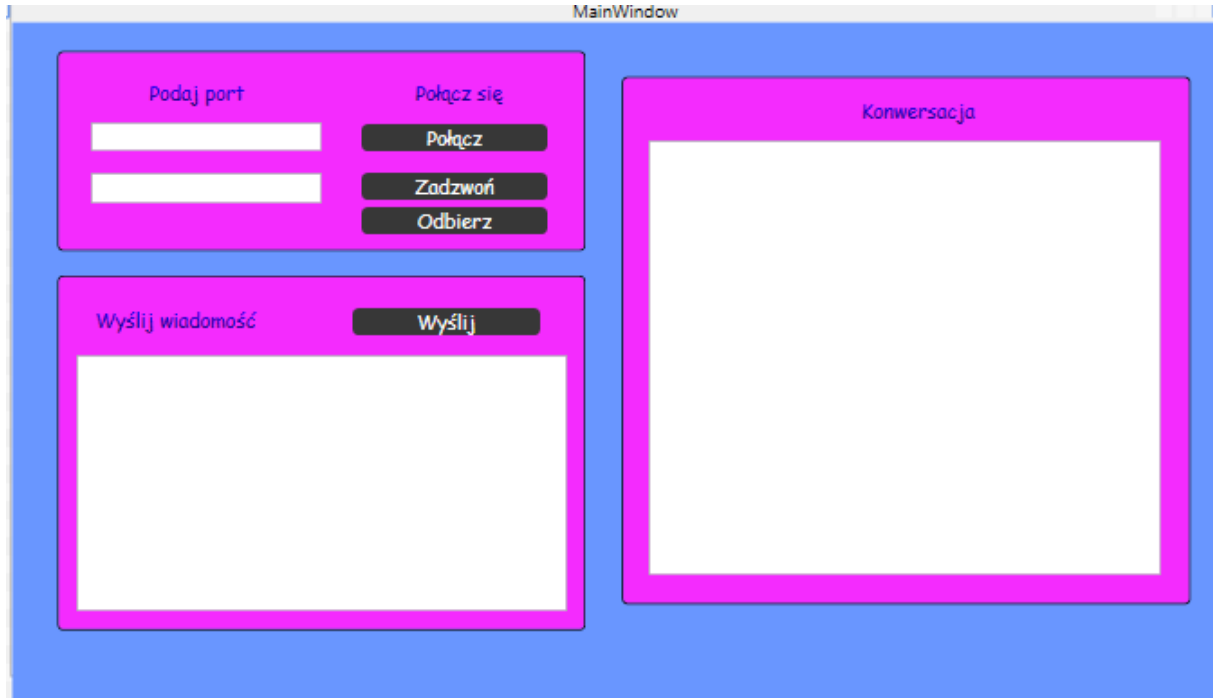
Zwrócono także uwagę na wysyłane przez modem do komputera komunikaty:

- NO CARRIER – brak sygnału nośnej,
- OK – prawidłowe wykonanie komendy,
- RING – wykrycie przychodzącego połączenia,
- CONNECT – wykryto sygnał nośnej.

Udało nawiązać się połączenie oraz przestać dane między dwoma komputerami przy pomocy Putty.

2.2 Własna aplikacja okienkowa

Kolejnym etapem zadnienia było napisanie aplikacji okienkowej służącej do komunikowania się komputerów przy pomocy modemów. Wygląd okna aplikacji został przedstawiony na rysunku nr 1.



Rysunek 1 Wgląd graficznego interfejsu użytkownika

W aplikacji zaimplementowano następujące funkcje:

1. Połączenie z portem

```
private void ConnectButton(object sender, RoutedEventArgs e)
{
    if (portTextBox.Text != null)
    {
        Connect(portTextBox.Text);
    }
}
```

2. Łączenie z modemem

```
public void Connect(string COM)
{
    Console.WriteLine(COM);

    if (serialPort != null)
        if (serialPort.IsOpen)
            serialPort.Close();

    serialPort = new SerialPort(COM);

    if (serialPort != null)
        serialPort.Open();

    if (serialPort.IsOpen)
    {
        serialPort.DtrEnable = true;
        serialPort.Handshake = Handshake.RequestToSend;

        Console.WriteLine(serialPort.PortName);
        Console.WriteLine(serialPort.BaudRate);
        Console.WriteLine(serialPort.Parity);
        Console.WriteLine(serialPort.DataBits);
        Console.WriteLine(serialPort.StopBits);
        Console.WriteLine(serialPort.Handshake);
        Console.WriteLine(serialPort.DtrEnable);

        reader = new Thread(Read);
        reader.Start();
    }
}
```

3. Wybranie numeru

```
private void CallButton(object sender, RoutedEventArgs e)
{
    string number = callTextBox.Text;

    serialPort.Write("ATDT " + number + Environment.NewLine);
}
```

4. Odebranie połączenia

```
private void PickupButton(object sender, RoutedEventArgs e)
{
    serialPort.Write("ATA" + Environment.NewLine);
}
```

5. Wysyłanie wiadomości – obsługa przycisku

```
private void SendButton(object sender, RoutedEventArgs e)
{
    string message;
    message = sendMessageTextBox.Text;

    Console.WriteLine("SENT MESSAGE: " + message);

    SendMessage(message);

    receiveMessageTextBox.Text += "Komputer2: ";
    receiveMessageTextBox.Text += message;
}
```

6. Wysyłanie wiadomości

```
public void SendMessage(string message)
{
    if (true)
        serialPort.Write(message + Environment.NewLine);

    else
        Console.WriteLine("CONNECTING TO SERIAL PORT FAILED");
}
```

7. Odbieranie wiadomości

```
private void Read()
{
    while (serialPort.IsOpen)
    {
        try
        {
            string message = serialPort.ReadLine();
            Console.WriteLine(message);
            //receiveMessageTextBox.Text += message;
        }
        catch (TimeoutException) { }
    }
}
```

3. Wnioski

Modemy to urządzenia, przy pomocy których można przesyłać sygnały między połączonymi przez nie komputerami. Do sterowania modemami wykorzystuje się komendy Hayes'a, z których każda rozpoczyna się od znaków AT, będących skrótem od słowa attention. Przydatna jest także wiedza na temat komunikatów informacyjnych wysyłanych do komputera przez modem.

Aplikacje służące do komunikacji z modemem napisano w języku C#. W aplikacji udało się zaimplementować funkcje służące połączenia z portem, połączenia z modemem, wybrania numeru, odebrania połączenia, wysłania wiadomości. Nie udało zaimplementować się prawidłowo działającej funkcji odbierającej wiadomości. Niepowodzenie w implementacji tej funkcji mogło wynikać z niezastosowania delegatów.

Bibliografia

- [1] Strona internetowa: <http://jaroslaw.mierzwa.staff.iiar.pwr.wroc.pl/uperyf/zdw-modemy.pdf>
- [2] Strona internetowa: <https://docs.microsoft.com/pl-pl/dotnet/api/system.io.ports.serialport?redirectedfrom=MSDN&view=dotnet-plat-ext-6.0>
- [3] Strona internetowa: <http://losno.lisa.univ-paris-diderot.fr/Electronik/Hayes.htm>
- [4] Strona internetowa: <http://www.intranetconcepts.com/sysadm/telecom/lights.htm>
- [5] Strona internetowa: <https://docs.rs-online.com/b9c6/0900766b80112e08.pdf>