

Paweł Twardawa 235072

Data: 4.12.2018 r.

Aleksandra Wieczorkiewicz 234980

# Urządzenia peryferyjne

## Ćwiczenie 3

Zasady obsługi modemów i nawiązywania połączeń

Prowadzący:

Dr inż. Jacek Mazurkiewicz

## 1. Wstęp

Celem zadania było zapoznanie się z obsługą modemów oraz nawiązanie połączenia za pomocą portu szeregowego. Następnie należało nawiązać połączenie pomiędzy dwoma komputerami wykorzystując modem korzystający z wewnętrznej sieci telefonicznej oraz za pomocą komend AT przesyłać dane tekstowe a także plik.

## 2. Zagadnienia

Do wykonania zadania zostały użyte dwa modemy Shadow 56k firmy Pentagram. Modem do komunikacji wykorzystuje port szeregowy, złącze RJ-11 oraz posiada następujące diody sygnalizacyjne:

- MR (modem ready) – modem gotowy do współpracy z komputerem
- HS (high speed) – transmisja danych odbywa się z maksymalną prędkością
- AA (auto answer) – praca w trybie automatycznego odbierania połączeń
- OH (on hook) – sygnalizacja podniesionej słuchawki
- RD (received) – sygnalizuje odbieranie danych
- TD (transmit) – sygnalizuje wysyłanie danych
- CD (carrier detected) – obecność nośnej, gdy modem jest podłączony, a na linii znajduje się nośna

Sterowanie modemem odbywa się za pomocą komend Hayes'a. Podczas realizacji zadania wykorzystaliśmy następujące polecenia:

- AT – sprawdzenie łączności między komputerem i modemem – modem odpowiada OK.
- ATA – odebranie połączenia przychodzącego (podniesienie słuchawki)
- ATH – zakończenie połączenia (odłożenie słuchawki)
- ATDT numer – wybranie tonowe podanego numeru telefonu
- ATE n – włączenie (dla n=1) lub wyłączenie (dla n=0) funkcji echa
- ATSO=n – włączenie (n=1) lub wyłączenie (n=0) funkcji automatycznego odbierania połączeń
- ATO – powrót z trybu komend (wywołanego sekwencją +++ ) do trybu danych.

Modem może także informować komputer za pomocą następujących komend:

- OK – potwierdzenie prawidłowego wykonania komendy
- RING – informuje o połączeniu przychodzącym
- CONNECT – informuje o pomyślnym nawiązaniu łączności między modemami i przejściu do trybu danych
- NO CARRIER – informuje o braku sygnału nośnej (np. linia telefoniczna nie jest podłączona)
- BUSY – informuje, że wybierany numer jest obecnie zajęty

### 3. Komunikacja z modemem

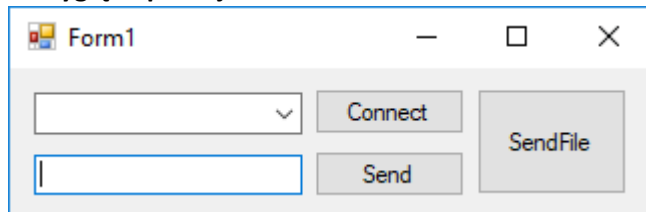
#### 3.1. PuTTY

Do nawiązania połączenia, przetestowania oraz zrozumienia zasad komunikacji z modemem wykorzystaliśmy program PuTTY. Po sprawdzeniu który port COM został przypisany modemowi, nawiązaliśmy połączenie o następujących parametrach: port COM1, szybkość 9600, bit danych 8, brak parzystości, bit stopu 1. Następnie na jednym z komputerów wydaliśmy polecenie ATD numer. Na drugim komputerze pojawił się komunikat RING. Po zaakceptowaniu połączenia poleceniem ATA zaczął się proces nawiązywania połączenia. Po chwili mogliśmy poprawnie przesyłać znaki tekstowe pomiędzy urządzeniami.

#### 3.2. Aplikacja

Kolejnym etapem zadania było stworzenie aplikacji nawiązującej połączenie z modemem oraz wysyłanie komunikatów. Do stworzenia programu użyliśmy języka C#. Program został zrealizowany w wersji okienkowej.

##### 3.2.1. Wygląd aplikacji



##### 3.2.2. Wyświetlanie dostępnych portów COM

Po uruchomieniu aplikacji wywołana zostaje metoda przedstawiona na Listingu 1 która pobiera dostępne porty COM. Następnie zwróconymi wartościami uzupełniany jest comboBox.

Listing 1.

```
public List<String> GetCOM()
{
    List<String>com = new List<String>();
    foreach(string s in SerialPort.GetPortNames())
    {
        com.Add(s);
    }
    com.Sort();
    return com;
}
```

### 3.2.3. Otwieranie połączenia

Po naciśnięciu przycisku „Connect” zostaje wywołana metoda `Connect(string com)`. Metoda rozpoczyna próbę otwarcia portu. Po nawiązaniu połączenia do konsoli wypisywane są podstawowe parametry połączenia. Natomiast w nowy wątku uruchamiana jest metoda odpowiedzialna za odbieranie komunikatów.

Listing 2.

```
public void Connect(string 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();
    }
}
```

### 3.2.4. Odczytywanie danych

Metoda uruchamiana w funkcji `Connect` odpowiedzialna jest za odbieranie komunikatów od modemu oraz wypisywanie ich w konsoli.

Listing 3.

```
private void Read()
{
    while(_serialPort.IsOpen)
    {
        try
        {
            string message = _serialPort.ReadExisting();
            Console.Write(message);
        }
        catch(TimeoutException)
        {
            Console.WriteLine("ERROR");
        }
    }
}
```

### 3.2.5. Wysyłanie danych

Wysyłanie danych odbywa się przez wpisanie komend do pola tekstowego a następnie naciśnięcie przycisku Send. Zdarzenie przycisku wywołuje metodę SendMessage(string message) która przekazuje tekst z parametru do otwartego portu szeregowego.

Listing 4.

```
public void SendMessage(string message)
{
    if (_serialPort != null)
        _serialPort.Write(message);
}
```

### 3.2.6. Wysyłanie pliku

Program umożliwia także przesłanie pliku. Odbywa się to przez otwarcie okna dialogowego po naciśnięciu przycisku SendFile. Po wybraniu odpowiedniego pliku i zatwierdzeniu przyciskiem Open, wywoływana jest metoda przedstawiona na Listingu 5 która rozpoczyna transmisję pliku.

Listing 5.

```
public void SendFile(string file)
{
    if (_serialPort.IsOpen)
    {
        using (FileStream stream = File.OpenRead(file))
        {
            _serialPort.Write((new BinaryReader(stream)).ReadBytes,
                ((int)stream.Length), 0, (int)stream.Length);
        }
    }
}
```

## 4. Wnioski

Dzięki tym zajęciom mogliśmy zapoznać się ze sposobem działania modemów, zrozumieć jak wygląda nawiązywanie połączenia oraz komunikacja między urządzeniami. Poznaliśmy także budowę oraz zastosowanie komend Hayes'a.