

# Urządzenia Peryferyjne

Ćwiczenie 12 – Transmisja Bluetooth (do wersji 3.1)

## **Autorzy**

Dawid Waligórski (264015)

Adrian Kotula (263989)

## **Data wykonania ćwiczenia**

16.11.2023

## **Prowadzący**

Dr. Inż. Dariusz Caban

## Spis treści

1	Cele ćwiczenia.....	2
2	Opis technologii .....	2
2.1	Standard Bluetooth oraz zasady realizacji transmisji .....	2
2.2	Zasady nawiązywania połączeń Bluetooth.....	2
2.3	Biblioteki wspierające oprogramowywanie transmisji Bluetooth.....	3
3	Realizacja zadania laboratoryjnego.....	3
3.1	Testy sprawności adaptera USB .....	3
3.2	Implementacja aplikacji.....	3
3.2.1	Opis działania aplikacji .....	3
3.2.2	Wyszukiwanie urządzeń .....	4
3.2.3	Parowanie .....	5
3.2.4	Transfer plików.....	6
4	Protokół z laboratorium.....	7
5	Podsumowanie.....	8

# 1 Cele ćwiczenia

Celem ćwiczenia było zapoznanie studentów z ideą oraz praktycznym działaniem technologii Bluetooth (do wersji 3.1). Dodatkowo mieli oni zaimplementować aplikację na komputer PC, która pozwoliłaby na komunikację za pośrednictwem wspomnianej technologii między adapterem USB (wpiętym do komputera) a telefonem komórkowym.

## 2 Opis technologii

### 2.1 Standard Bluetooth oraz zasady realizacji transmisji

Bluetooth to technologia krótkodystansowego połączenia bezprzewodowego, umożliwiająca komunikację między urządzeniami w zasięgu do 10 metrów. Działa ona w paśmie częstotliwości ISM (2,402-2,480 GHz). Standard działania tej technologii został precyzyjnie określony w standardzie *IEEE 802.15.1*.

Bluetooth korzysta z technologii radiowej FHSS (Frequency Hopping Spread Spectrum). Polega ona na sekwencyjnych przeskokach urządzeń pomiędzy 79 kanałami częstotliwościowymi w celu ustanowienia połączenia. Dane przesyłane poprzez wspomniane kanały dzielone są dodatkowo na pakiety.

Początkowo technologia Bluetooth używała metody kluczowania częstotliwości (GFSK), ale wraz z nadejściem wersji 2.0+EDR pojawiła się możliwość wykorzystania kluczowania fazy (DQPSK). Urządzenia korzystające z kluczowania częstotliwości działają z przepustowością podstawową (BR) wynoszącą 1 Mb/s. Urządzenia korzystające z kluczowania fazy działają w trybie Enhanced Data Rate (EDR), czyli z przepustowością 2 lub 3 Mb/s. Możliwe jest również działanie w trybie BR/EDR, o ile urządzenie obsługuje oba typy modulacji. Wprowadzenie na rynek wersji 3.0+HS przyniosło ze sobą zwiększenie przepustowości łącza transmitującego dane do 24 Mb/s. Została ona osiągnięta poprzez użycie równoległego połączenia zgodnego ze standardem *IEEE 802.11* do transmisji danych (łącze Bluetooth pełni wyłącznie funkcję sterowania połączeniem między urządzeniami).

Rozważana technologia działa w architekturze master-slave, gdzie urządzenie nadrzędne kontroluje komunikację w pikosieci, składającej się z maksymalnie 7 urządzeń. Komunikacja między urządzeniami w takiej pikosieci jest synchronizowana za pomocą zegara urządzenia nadrzędnego. Dzięki niemu tworzone są „sloty” przesyłowe o długości 625  $\mu$ s. Urządzenie master może nadawać w slotach zaczynających się w momentach parzystych (podzielnych przez 1250  $\mu$ s). Urządzenia slave mogą nadawać w slotach zaczynających się w momentach nieparzystych (niepodzielnych przez 1250  $\mu$ s). Rola urządzenia nadrzędnego może być przekazywana między urządzeniami w pikosieci. Dodatkowo pikosieci mogą zostać połączone w większą sieć nazywaną scatternet.

### 2.2 Zasady nawiązywania połączeń Bluetooth

Pierwszym krokiem niezbędnym do nawiązania połączenia Bluetooth jest umieszczenie urządzenia w trybie *discoverable*, co oznacza, że jest gotowe do nawiązania połączeń z innymi urządzeniami w swoim otoczeniu. W tym trybie urządzenie przesyła informacje identyfikujące takie jak: nazwa, adres (MAC, dzięki któremu urządzenia mogą się połączyć), klasa, lista oferowanych usług Bluetooth oraz dane techniczne (np. na temat producenta).

Nawiązanie połączenia wymaga zidentyfikowania innego urządzenia jako potencjalnego partnera. Służy do tego proces skanowania, który pozwala urządzeniu odkryć inne urządzenia Bluetooth w jego zasięgu. Po znalezieniu interesującego nas urządzenia proces parowania może zostać zainicjowany. Parowanie jest niezwykle istotną częścią nawiązywania połączenia dla tych usług Bluetooth, które wymagają szczególnego bezpieczeństwa, takich jak dostęp do danych urządzenia czy sterowanie nim.

Proces parowania zazwyczaj obejmuje przesłanie specjalnego żądania, które może być inicjowane ręcznie przez użytkownika lub automatycznie w ramach danej usługi. Zwykle wymagana

jest wówczas interakcja z użytkownikiem, który potwierdza chęć sparowania i zezwala na nawiązanie połączenia poprzez podanie klucza bezpieczeństwa (PIN-u).

## 2.3 Biblioteki wspierające oprogramowywanie transmisji Bluetooth

W ramach przygotowania do laboratorium zapoznano się z funkcjonalnościami oferowanymi przez API *32feet.NET* dla języka C#. Jest to otwarto-źródłowe rozwiązanie, które pozwala na relatywnie proste oprogramowywanie transmisji Bluetooth. Co ważne rozważane API pozwala również na wykorzystanie protokołu OBEX (Object Exchange), który umożliwia transmisję plików dowolnego typu (w formie binarnej) poprzez Bluetooth. Kod źródłowy API *32feet.NET* znajduje się pod adresem [1], dokumentację znaleźć można pod adresem [2].

[1] <https://github.com/inthehand/32feet>

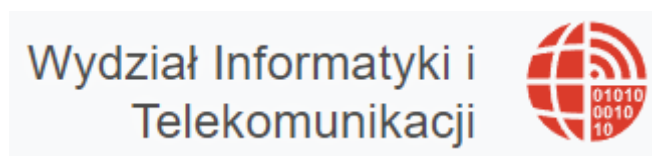
[2] [https://inthehand.github.io/html/R\\_Project\\_32feet\\_NET.htm](https://inthehand.github.io/html/R_Project_32feet_NET.htm)

## 3 Realizacja zadania laboratoryjnego

### 3.1 Testy sprawności adaptera USB

Pierwszym krokiem pracy na zajęciach było zweryfikowanie sprawności adapterów USB, pozwalających na użycie technologii Bluetooth na komputerach laboratoryjnych. Wykorzystano przy tym program *Ustawienia Bluetooth* dostarczany wraz z systemem operacyjnym *Windows 10*. Za jego pomocą dokonano:

- Skanowania otoczenia komputera w poszukiwaniu aktywnych urządzeń Bluetooth
- Sparowania komputera z jednym z telefonów komórkowych posiadanych przez studentów
- Przesłania pliku *.png* (rysunek 1) z komputera na telefon komórkowy



Rysunek 1: Obraz zawarty w pliku *.png*, który użyto do testów adaptera USB a później zaimplementowanej aplikacji

### 3.2 Implementacja aplikacji

#### 3.2.1 Opis działania aplikacji

W drugiej części laboratorium studenci mieli zaimplementować aplikację na komputer PC, która umożliwiać miała:

- Wyszukiwanie aktywnych urządzeń Bluetooth w pobliżu
- Realizację połączenia z wybranym urządzeniem (parowanie) Bluetooth
- Przesłanie piku z komputera na telefon komórkowy za pomocą Bluetooth

W ramach zajęć udało się zaimplementować wszystkie wymienione wyżej funkcjonalności (szczegóły w odpowiednich podsekcjach). Stworzona aplikacja została wykonana z pomocą technologii *.NET (Windows Forms)*. Umożliwiło to łatwe wyposażenie jej w graficzny interfejs użytkownika (rys. 2). Działanie poszczególnych elementów zawartych w interfejsie użytkownika przedstawiono w tabeli 1.

Rysunek 2: Wygląd graficznego interfejsu użytkownika zaimplementowanej w ramach laboratorium aplikacji

Tabela 1: Opis działania poszczególnych elementów interfejsu

Element interfejsu	Funkcja
Przycisk „Wyszukaj urządzenia”	Wykonanie skanowania otoczenia w poszukiwaniu urządzeń Bluetooth.
Lista „Wykryte urządzenia”	Wyświetla listę nazw wszystkich wykrytych podczas ostatniego skanowania urządzeń wraz ewentualną informacją na temat sparowania.
Przycisk „Sparuj”	Sparowanie z urządzeniem wybranym z listy „Wykryte urządzenia”.
Przycisk „Rozłącz”	Usunięcie sparowania z urządzeniem wybranym z listy „Wykryte urządzenia”.
Dolna lista	Wyświetla listę ścieżek do wszystkich plików, które mogą zostać przesłane za pomocą Bluetooth do urządzenia wybranego z listy „Wykryte urządzenia”.
Przycisk „Wyślij plik”	Wysyła plik wybrany z listy plików do urządzenia wybranego z listy „Wykryte urządzenia”.
Przycisk „Wczytaj plik”	Pozwala wczytać do aplikacji plik z komputera wskazany przez użytkownika.

### 3.2.2 Wyszukiwanie urządzeń

Wyszukiwanie urządzeń zostało zaimplementowane w ramach metody *lookForDevices* (listing 2), która wywoływana jest po użyciu przycisku „Wyszukaj urządzenia” w interfejsie użytkownika. Wewnątrz niej skorzystano z obiektu klasy *BluetoothClient* z API *32feet.NET*, który reprezentował komputer laboratoryjny jako urządzenie Bluetooth. Wywołanie metody *DiscoverDevices* tej klasy inicjowało proces skanowania otoczenia w poszukiwaniu innych urządzeń Bluetooth. Owa metoda po zakończeniu procesu skanowania wypełniała zapamiętywaną przez aplikację tablicę *discovered\_devices* (listing 1) danymi otrzymanymi od wszystkich znalezionych urządzeń (w sekcji 2.2 dokładniej opisano spis tychże danych).

Listing 1: Deklaracja tablicy przechowującej informacje nt. wykrytych urządzeń Bluetooth

```
BluetoothDeviceInfo[] discovered_devices;
```

Listing 2: Metoda odpowiedzialna za skanowanie otoczenia urządzenia w poszukiwaniu urządzeń Bluetooth

```
private void lookForDevices()
{
    BluetoothClient this_machine = new BluetoothClient();
    discovered_devices = this_machine.DiscoverDevices();
}
```

### 3.2.3 Parowanie

Parowanie urządzeń zostało zaimplementowane w ramach metody *btn\_pair\_Click* (listing 3), która wywoływana jest po użyciu przycisku „Sparuj” w interfejsie użytkownika. W ramach jej działania w pierwszej kolejności ustalano urządzenie, z którym nastąpić sparowanie. W tym celu z tablicy wykrytych urządzeń wybierano te, które użytkownik wskazał na odpowiedniej liście w interfejsie graficznym. Następnie sprawdzano, czy owo urządzenie nie zostało już wcześniej sparowane (posiada aktywną flagę *Authenticated*). Jeżeli okazało się, że nie zostało ono jeszcze sparowane, to za pomocą metody statycznej *BluetoothSecurity.PairRequest* z API *32feet.NET* inicjowano proces parowania (szerzej opisany w sekcji 2.2). W celu uproszczenia testowania aplikacji PIN niezbędny do sparowania urządzeń przyjął statyczną postać i wynosił „1”. Po sparowaniu urządzeń powiadamiano użytkownika o sukcesie procesu i wykonywano ponowne skanowanie otoczenia oraz odświeżenie danych na interfejsie użytkownika.

Listing 3: Metoda odpowiedzialna za parowanie z innym urządzeniem

```
private void btn_pair_Click(object sender, EventArgs e)
{
    if (found_devices_display.SelectedItem == null)
    {
        MessageBox.Show("Nie wybrano urządzenia!");
        return;
    }

    BluetoothDeviceInfo end_device =
        discovered_devices[found_devices_display.SelectedIndex];

    if(!end_device.Authenticated)
    {
        if(!BluetoothSecurity.PairRequest(end_device.DeviceAddress, "1"))
        {
            MessageBox.Show("Parowanie nieudane!");
            return;
        }
    }
    MessageBox.Show("Sparowano!");
    lookForDevices();
    updateFoundDevicesDisplay();
}
```

Aplikacja pozwalała także na usunięcie relacji sparowania między urządzeniami. Implementowała to metoda *unpairDevice* (listing 4), która mogła być wywołana ręcznie przez użytkownika z pomocą przycisku „Rozłącz” bądź automatycznie przy zamykaniu aplikacji (wówczas następuje usunięcie sparowania z każdym ze sparowanych urządzeń). Wewnątrz niej wykorzystuje się metodę *BluetoothSecurity.RemoveDevice* z API *32feet.NET*. Usuwa ona relację parowania między aktualnym urządzeniem a urządzeniem o wskazanym adresie Bluetooth MAC.

Listing 4: Metoda odpowiedzialna za rozparowanie z innym urządzeniem

```
bool unpairDevice(BluetoothDeviceInfo device)
{
    if (device.Authenticated)
    {
        if (!BluetoothSecurity.RemoveDevice(device.DeviceAddress))
        {
            MessageBox.Show("Nie rozparowano!");
            return false;
        }
        MessageBox.Show("Rozparowano!");
        return true;
    }
    return false;
}
```

### 3.2.4 Transfer plików

Transfer plików do innego urządzenia Bluetooth został zaimplementowany w ramach metody *sendFileToDevice* (listing 5). Jest ona wywoływana za pomocą przycisku „Wyślij plik” na interfejsie użytkownika. Wykorzystywała ona protokół OBEX. W pierwszej kolejności generowała ona żądanie przesłania pliku poprzez OBEX, który wymagał podania identyfikatora pliku (URI, Uniform Resource Identifier). W owym identyfikatorze precyzowano adres urządzenia, które miało odebrać plik, jak i ścieżkę do wysyłanego pliku. Po wygenerowaniu żądania metoda wykonuje odczyt danych z pliku. Następnie wysyła żądanie do odbiorcy i odbiera nadaną przez niego odpowiedź. Po odczytaniu flag zawartych w odpowiedzi OBEX aplikacja informuje użytkownika o powodzeniu lub niepowodzeniu transferu pliku.

Listing 5: Metoda pozwalająca na przesłanie pliku do innego urządzenia za pomocą Bluetooth w protokole OBEX

```
void sendFileToDevice(string file, BluetoothDeviceInfo device)
{
    try
    {
        ObexWebRequest request = new ObexWebRequest
        (
            new Uri(String.Format("obex://{0}/{1}",
                device.DeviceAddress,
                file))
        );

        request.ReadFile(file);

        ObexWebResponse response = (ObexWebResponse)request.GetResponse();

        if (response.StatusCode ==
            (ObexStatusCode.OK | ObexStatusCode.Final))
        {
            MessageBox.Show
            ("Plik wysłany!");
        }
        else if (response.StatusCode ==
            (ObexStatusCode.Forbidden | ObexStatusCode.Final))
        {
            MessageBox.Show
            ("Plik niewysłany! Odmowa dostępu do urządzenia.");
        }
    }
    catch (Exception e)
    {
        MessageBox.Show($"{e.Message}");
    }
}
```



#### 4 Protokół z laboratorium

RAPORT Z LABORATORIUM - 16.11.2023

Temat: Blue tooth (ćw nr. 12)

##### Zad 1

- uruchomienie aplikacji ustawień bluetooth systemu Windows i spowodowanie połączenia oraz przesłanie pliku na telefon komórkowy.

##### Zad 2

Implementacja aplikacji umożliwiającej

- otwarcie i zainicjowanie urządzenia - wykonano
- wyszukanie ur. Bluetooth - wykonano
- porównanie z urządzeniem - wykonano
- przesłanie zbioru (tekstowego fotografii) przez Bluetooth - wykonano

Wojciech

Kotula

Albo  
16.11.23  
(kom. z tel.)

Rysunek 3: Protokół z laboratorium



## **5 Podsumowanie**

Udało się z powodzeniem zaimplementować aplikację pozwalającą na komunikację między komputerem a telefonem komórkowym za pomocą technologii Bluetooth. Ponadto wykonane zadania oraz przygotowanie przed zajęciami istotnie pozwoliły studentom na lepsze poznanie technologii Bluetooth i jej praktycznych zastosowań.