Mateusz Krawczak 241318

Karol Jaskółka 241306

Grupa: Pon P 17:00

Data wykonania ćwiczenia: 25.11.2019

**Urządzenia Peryferyjne**

**Ćwiczenie 14 – Bluetooth - komunikacja z telefonem komórkowym**

# **Wstęp**

Na trzecich zajęciach laboratoryjnych otrzymaliśmy zadanie polegające na napisaniu programu pozwalającego na połączenie się przez bluetooth z dowolnym urządzeniem. Zadania do wykonania:

* Wykryć adaptery BT podłączone do PC.
* Użyć wybranego adaptera do zdalnego wyszukiwania urządzeń BT.
* Pobrać adres MAC wybranego (wyszukanego w pkt. 2) urządzenia.
* Dokonać autoryzacji obu urządzeń:   
     - po stronie urządzenia BT autoryzować PC  
     - po stronie PC autoryzować urządzenie BT.
* Uruchomić urządzenie BT w tryb pracy transferu plików.
* Przesłać plik tekstowy do urządzenia BT.
* Przesłać plik graficzny do urządzenia BT.

1. **Zagadnienia**

**Bluetooth**

Standard komunikacji bezprzewodowej opisan normami IEEE 802.15.1. Określane są trzy klasy mocy nadawczej:

* Klasa 1 – 100mW do 100m
* Klasa 2 – 2,5 MW do 10m
* Klasa 3 – 1mW do 1m

Standard korzysta z [fal radiowych](https://pl.wikipedia.org/wiki/Fale_radiowe) w paśmie [ISM](https://pl.wikipedia.org/wiki/Pasmo_ISM) 2,4 [GHz](https://pl.wikipedia.org/wiki/Herc). Na terenie Polski akceptowalne przedział częstotliwości wynosi 2,4 - 24,25 GHz

Bluetooth jest standardem cały czas rozwijamym:

* Bluetooth 1.0 – 21 [kb/s](https://pl.wikipedia.org/wiki/Bit_na_sekund%C4%99" \o "Bit na sekundę)
* Bluetooth 1.1 – 124 [kb/s](https://pl.wikipedia.org/wiki/Bit_na_sekund%C4%99" \o "Bit na sekundę)
* Bluetooth 1.2 – 328 [kb/s](https://pl.wikipedia.org/wiki/Bit_na_sekund%C4%99" \o "Bit na sekundę)
* Bluetooth 2.0 + EDR – wprowadzenie Enhanced Data Rate zwiększyło transfer teoretyczny do 2,1 [Mb/s](https://pl.wikipedia.org/wiki/Bit_na_sekund%C4%99" \o "Bit na sekundę) (około 3 [Mb/s](https://pl.wikipedia.org/wiki/Bit_na_sekund%C4%99" \o "Bit na sekundę) wliczając narzut protokołu)
* Bluetooth 2.1 + EDR - uproszczenie i ujednolicenie procesu parowania urządzeń BT, wsparcie dla przyszłych implementacji [NFC](https://pl.wikipedia.org/wiki/Near_Field_Communication), zmniejszenie zużycia energii[[4]](https://pl.wikipedia.org/wiki/Bluetooth#cite_note-4)
* Bluetooth 3.0 + HS (High Speed) – 24 [Mb/s](https://pl.wikipedia.org/wiki/Bit_na_sekund%C4%99) (3 MB/s)
* Bluetooth 3.1 + HS (High Speed) – 40 [Mb/s](https://pl.wikipedia.org/wiki/Bit_na_sekund%C4%99) (5 MB/s)
* Bluetooth 4.0 + LE (Low Energy) – 1 Mb/s znacząco ograniczono pobór energii (np. praca czujnika temperatury, przez wiele miesięcy na baterii pastylkowej), kosztem obniżonego transferu oraz zwiększono realny zasięg działania do 100 m
* Bluetooth 4.1 - standard opracowany do zastosowania w tzw. "[internecie rzeczy](https://pl.wikipedia.org/wiki/Internet_rzeczy" \o "Internet rzeczy)" (urządzenia typu "[wearables](https://pl.wikipedia.org/wiki/Wearables" \o "Wearables)"), umożliwiający bezpośrednią łączność przedmiotów z internetem
* Bluetooth 4.2 - w stosunku do poprzednich wersji: szybszy transfer, wyższy poziom bezpieczeństwa, nawiązanie łączności z przedmiotami - łatwiejsze
* Bluetooth 5.0 - ujednolicenie wersji, szybszy transfer – 2 Mb/s dla urządzeń typu ["wearables"](https://pl.wikipedia.org/wiki/Wearables) i 50 Mb/s do normalnych, realny zasięg działania do 140m

Standard Bluetooth określa wiele protokołów, pogrupowanych w warstwy. Struktura warstw nie odpowiada żadnemu znanemu modelowi ([OSI](https://pl.wikipedia.org/wiki/Model_OSI), [TCP/IP](https://pl.wikipedia.org/wiki/Model_TCP/IP), 802). IEEE prowadzi prace nad zmodyfikowaniem systemu Bluetooth, aby dopasować go do modelu określonego standardem 802.

Każde urządzenie ma 48 bitowy adres IEEE MAC (Bluetooth Device Address, BD\_ADDR) i jest on używany do inicjowania pewnych operacji oraz obliczania kodu dostępu.

Moduł Bluetooth wyposażony jest w 28-bitowy wewnętrzny zegar, który determinuje synchronizację i skakanie po częstotliwościach. Nigdy nie jest on dostrajany, ani wyłączany. Do synchronizacji z innym modułem Bluetooth wykorzystywana jest różnica (offset) pomiędzy zegarami jednostek chcących się komunikować. Częstotliwość zegara wynosi 3,2 kHz.

**OBEX**

Object Exchange - protokół komunikacyjny, określający procedury wymiany danych binarnych między urządzeniami. Rozwojem i utrzymaniem specyfikacji zajmuje się **[Infrared Data Association](https://pl.wikipedia.org/wiki/Infrared_Data_Association" \o "Infrared Data Association)**.

Specyfikacja protokołu OBEX opiera się architekturze klient - serwer. Klient wykorzystuje zaufane medium transportowe do połączenia z serwerem w celu zażądania transmisji obiektów. Przesyłane obiekty są zapisane w formacie binarnym

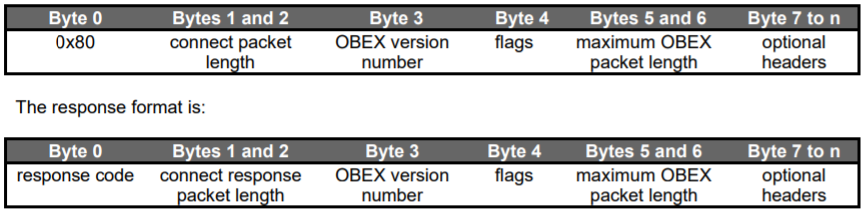
Protokół sesji OBEX opisuje podstawową strukturę sesji rozmowy między urządzeniami z wykorzystaniem określonych w standardzie zestawem kodów operacyjnych określających poszczególne działania.

Podczas trwania sesji OBEX realizuje połącznie zgodnie z paradygmatem request-respone dla klienta/serwera. Żądania są wystawianie przez klienta, który nastepnie oczekuje na odpowiedz serwera. Nie zostanie wydane kolejne żądanie bez otrzymanej odpowiedzi. Taki zestaw działań nazywamy pojedyńczą operacją.

Każdy pakiet Żądania składa się z kodu operacyjnego (np. GET), długości pakietu i co najmniej jednego nagłowka( zgodnie ze standardem nagłówki muszą być wysyłane w jedym pakiecie- nie można ich podzielić na wiele pakietów). Nagłowki mogą zawierać dane tj. : ops obiektu(nazawa długość, data). Nagłówki powinny być wysyłane jako pierwsze, przed właściwą zawartością.

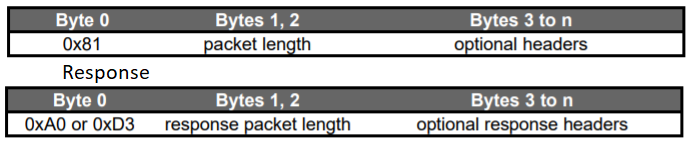
1. **Connect**

Ta operacja inicjuje połączenie i określa podstawowe oczekiwania każdej strony łącza.

**sass**

Żądanie i odpowiedź CONNECT muszą się mieścić w jednym pakiecie.

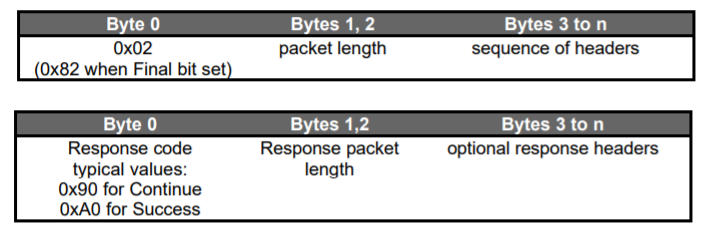
1. **DISCONNECT**

Ten kod operacyjny sygnalizuje koniec sesji OBEX. Może zawierać dodatkowy nagłówek Description- czyli informacje czytelne dla użytkownika. Żądanie i odpowiedź DISCONNECT muszą się mieścić w

-0xA0 Operacja udana

- 0xD3 Service Unavailable

**3. PUT**

Operacja ta wysyła obiekt z klienta do serwera. Request standardowo powinien zawierać nagłówki: nazwa i długość. Może występować także nagłowek Data/Time. Jednakże istnieje możliwość pominięcia wszystkich headerów- jeżeli urządzenie docelowe jest bardzo proste i może odpierać/obsługiwać tylko jeden typ obiektu a komunikacja z niezaufanych źródeł jest niemożliwa.

1. **Przebieg zajęć i kod programu**

Cały program znajduję się pod tym linkiem <https://github.com/matson19/UP/tree/master/Lab%202%20-%20analizator_sieci>

namespace Bluetooth

{

public partial class Form1 : Form

{

private BluetoothDeviceInfo[] devices;

private bool isPaired = false;

private BluetoothDeviceInfo deviceToPair = null;

int selectedID;

public Form1()

{

InitializeComponent();

buttonUnpair.Enabled = false;

buttonSendFile.Enabled = false;

}

private void buttonFind\_Click(object sender, EventArgs e)

{

Console.WriteLine("szukanie");

listBoxDevices.Items.Clear();

var bluetoothClient = new BluetoothClient();

devices = bluetoothClient.DiscoverDevices();

foreach(BluetoothDeviceInfo device in devices)

{

listBoxDevices.Items.Add(device.DeviceName);

Console.WriteLine(device.DeviceName + " " + device.DeviceAddress + " " + device.ClassOfDevice.Device);

}

}

private void buttonPair\_Click(object sender, EventArgs e)

{

foreach(BluetoothDeviceInfo device in devices)

{

if (device.DeviceName == (string)listBoxDevices.SelectedItem)

{

deviceToPair = device;

selectedID = listBoxDevices.SelectedIndex;

}

}

if(deviceToPair == null)

{

Console.WriteLine("Nie wybrano nic do sprowania");

}

else

{

Console.WriteLine("Parowanie z " + deviceToPair.DeviceName);

deviceToPair.Update();

deviceToPair.Refresh();

deviceToPair.SetServiceState(BluetoothService.ObexObjectPush, true);

string pin = "000000";

isPaired = BluetoothSecurity.PairRequest(deviceToPair.DeviceAddress, pin);

if (isPaired)

{

Console.WriteLine("Sparowano");

buttonSendFile.Enabled = true;

listBoxConnected.Items.Add(deviceToPair.DeviceName);

buttonPair.Enabled = false;

buttonUnpair.Enabled = true;

}

else

{

Console.WriteLine("Nie Sparowano");

}

}

}

private void buttonUnpair\_Click(object sender, EventArgs e)

{

if (isPaired)

{

buttonUnpair.Enabled = false;

BluetoothSecurity.RemoveDevice(deviceToPair.DeviceAddress);

Console.WriteLine("Odparowano");

}

}

private void buttonSendFile\_Click(object sender, EventArgs e)

{

openFileDialog.ShowDialog();

}

private void openFileDialog\_FileOk(object sender, CancelEventArgs e)

{

Task.Run(async () =>

{

await sendFileMethod(sender, this);

}

);

}

private async Task sendFileMethod(object sender, Form1 form)

{

OpenFileDialog dialog = (OpenFileDialog)sender;

BluetoothAddress address = devices[selectedID].DeviceAddress;

var path = String.Format("obex://{0}/{1}", address.ToString(), dialog.FileName);

ObexWebRequest request = new ObexWebRequest(new Uri(path));

Stream stream = request.GetRequestStream();

request.ReadFile(dialog.FileName);

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

response.Close();

}

}

}

1. **Podsumowanie**

Program działał poprawnie, najpierw odszukaliśmy wszystkie urządzenia w pobliżu, które mają włączone bluetooth. Następnie sparowaliśmy telefon z komputerem i udało nam się przesłać plik graficzny, a także plik tekstowy z komputera do telefonu.