

Baraniecki Karol Byczko Maciej	Prowadzący: Dr inż. Dominik Żelazny	Numer ćwiczenia laboratoria 14
PT 16:30 TP	Temat ćwiczenia: Bluetooth - komunikacja z telefonem komórkowym	Ocena:
Grupa: D	Data wykonania: 26 listopada 2021	

1 Zagadnienia do opracowania

1.1 MS Platform SDK

zestaw do tworzenia oprogramowania (SDK) firmy Microsoft, który zawiera dokumentację, pliki nagłówkowe, biblioteki, próbki i narzędzia wymagane do tworzenia aplikacji dla systemu Microsoft Windows i .NET Framework. Platform SDK specjalizuje się w tworzeniu aplikacji dla systemów Windows 2000, XP i Windows Server 2003. Platform SDK jest następcą oryginalnego Microsoft Windows SDK dla Windows 3.1x i Microsoft Win32 SDK dla Windows 9x. Został wydany w 1999 roku i jest najstarszym SDK. Platform SDK zawiera kompilatory, narzędzia, dokumentację, pliki nagłówkowe, biblioteki i próbki potrzebne do tworzenia oprogramowania na architekturach procesorów IA-32, x64 i IA-64.

Zestawy SDK systemu Windows są dostępne za darmo; kiedyś były one dostępne w Centrum pobierania Microsoft, ale w 2012 roku zostały przeniesione do MSDN.

Programista może chcieć użyć starszego zestawu SDK z konkretnego powodu. Na przykład pakiet Windows Server 2003 Platform SDK wydany w lutym 2003 roku był ostatnim pakietem SDK zapewniającym pełną obsługę Visual Studio 6.0. Niektóre starsze wersje PSDK można nadal pobrać z Centrum pobierania firmy Microsoft; inne można zamówić na płycie CD/DVD.

1.2 Znajomość funkcji

Znajomość najważniejszych funkcji zdefiniowanych w:

1. winsock2.h - najważniejsze funkcje:

- accept - zezwolenie na próbę połączenia przychodzącego na gniazdo.
- bind - skojarzenie adresu lokalnego z gniazdem.
- closesocket - zamknięcie istniejącego gniazda.

2. Ws2bth.h - najważniejsze interfejsy:

- BTH_QUERY_DEVICE - Wykorzystywana przy zapytaniu o obecność urządzenia Bluetooth.
- BTH_QUERY_SERVICE - Używana do odpytywania usługi Bluetooth.
- BTH_SET_SERVICE - Udostępnia informacje o usłudze dla określonej usługi Bluetooth.
- SOCKADDR_BTH - Używana w połączeniu z operacjami na gniazdach Bluetooth, zdefiniowanymi przez rodzinę adresów AF_BTH.

3. BluetoothAPIs.h - najważniejsze interfejsy:

- BluetoothGATTAbortReliableWrite - Określa koniec procedur niezawodnego zapisu, a zapisy powinny zostać przerwane.

- BluetoothGATTBeginReliableWrite - Informuje o rozpoczęciu niezawodnego zapisu.
- BluetoothGATTEndReliableWrite - Określa koniec niezawodnych zapisów, które powinny zostać wykonane.
- BluetoothGATTGetCharacteristics - Uzyskuje wszystkie charakterystyki dostępne dla określonej usługi.
- BluetoothGATTGetCharacteristicValue - Uzyskuje wartość określonej charakterystyki.
- BluetoothGATTGetDescriptors - Uzyskuje wszystkie deskryptory dostępne dla określonej charakterystyki.
- BluetoothGATTGetDescriptorValue - Uzyskuje wartość określonego deskryptora.
- BluetoothGATTGetIncludedServices - Uzyskuje wszystkie dołączone usługi dostępne dla danej usługi.
- BluetoothGATTGetServices - Pobiera wszystkie usługi podstawowe dostępne dla danego serwera.
- BluetoothGATTRegisterEvent - Rejestruje procedurę, która ma być wywoływana podczas zdarzenia zmiany wartości cechy na danej charakterystyce identyfikowanej przez jej uchwyt.
- BluetoothGATTSetCharacteristicValue - Zapisuje określoną wartość charakterystyki do urządzenia Bluetooth.
- BluetoothGATTSetDescriptorValue - Zapisuje określoną wartość deskryptora do urządzenia Bluetooth.
- BluetoothGATTUnregisterEvent - Wyrejestrowuje podane zdarzenie zmiany wartości charakterystyki.

1.3 Ogólnie pojęcie o mechanizmach rejestracji funkcji callbackowych

Wywołanie zwrotne (callback) - technika programowania będąca odwrotnością wywołania funkcji. Zwykle korzystanie z właściwości konkretnej biblioteki polega na wywołaniu funkcji (podprogramów) dostarczanych przez tę bibliotekę. W tym przypadku jest odwrotnie: użytkownik jedynie rejestruje funkcję do późniejszego wywołania, natomiast funkcje biblioteki wywołają ją w stosownym dla siebie czasie.

1.4 Zapoznanie się ze specyfikacją komunikacji poprzez BT

Bluetooth jest otwartą specyfikacją standardu sieci, która umożliwia bezprzewodową, szybką komunikację na małe odległości między różnymi urządzeniami elektronicznymi (np. telefon komórkowy, PDA, piloty telewizyjne, laptopy, urządzenia peryferyjne) . Specyfikacja ta powstała, aby wyeliminować połączenia kablowe pomiędzy urządzeniami

1.5 Protokół transferu plików OBEX

OBEX (OBject EXchange [wymiana obiektów]) - protokół komunikacyjny, który ułatwia wymianę obiektów binarnych pomiędzy urządzeniami. Jest on utrzymywany przez Infrared Data Association, ale został również przyjęty przez Bluetooth Special Interest Group i skrzydło SyncML organizacji Open Mobile Alliance (OMA). Jednym z najwcześniejszych popularnych zastosowań OBEX był Palm III. Ten PDA i jego liczni następcy używają OBEX do wymiany wizytówek, danych, a nawet aplikacji. Chociaż OBEX został początkowo zaprojektowany dla podczerwieni,

obecnie został zaadoptowany przez Bluetooth, a także jest używany przez RS-232, USB, WAP oraz w urządzeniach takich jak smart-peny Livescribe.

Znajomość poleceń:

- CONNECT - Połącz
- PUT - Umieść
- DISCONNECT - Rozłącz

2 Zadania do wykonania

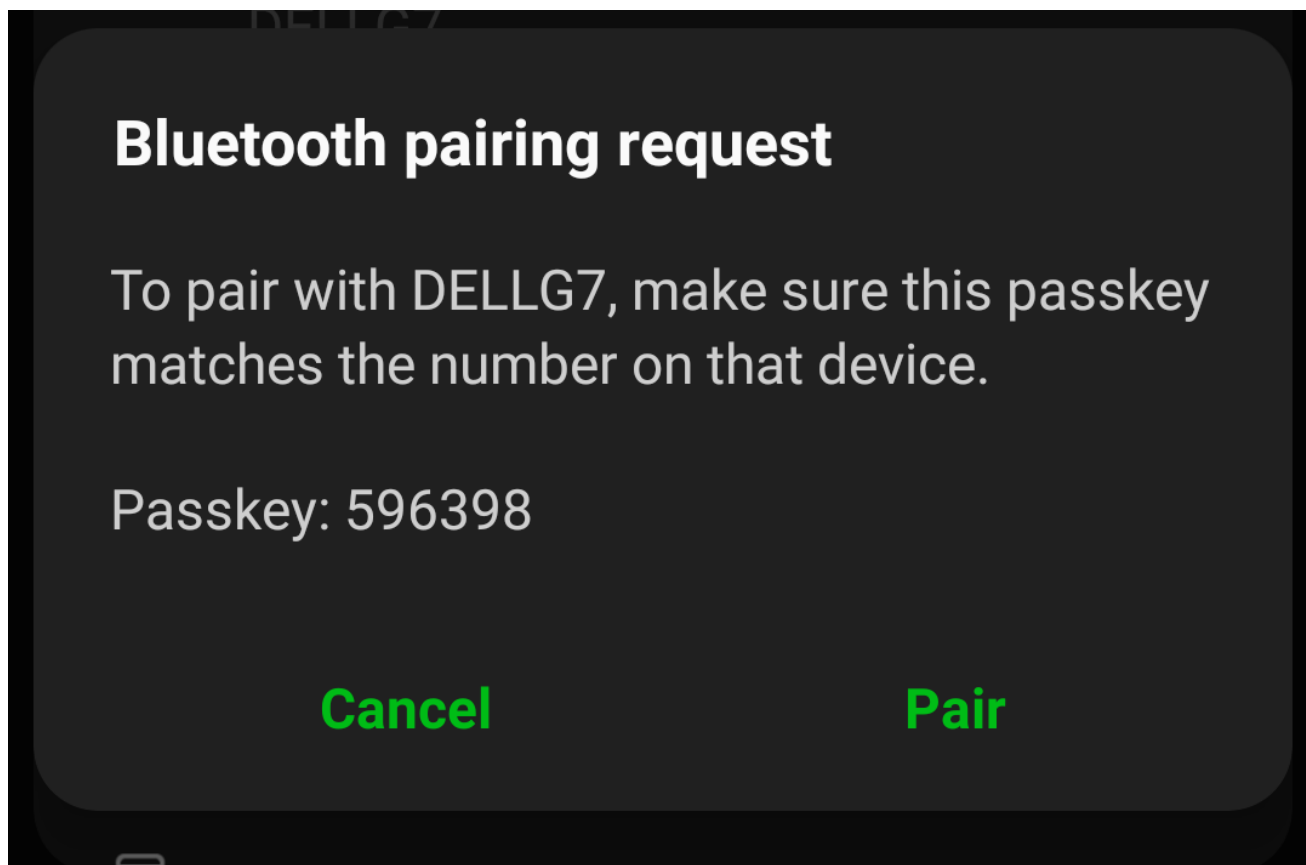
napisać aplikację graficzną, która:

1. Wykryć adaptory BT podłączone do PC.
2. Użyć wybranego adaptera do zdalnego wyszukiwania urządzeń BT.
3. Pobrać adres MAC wybranego (wyszukanego w pkt. 2) urządzenia.

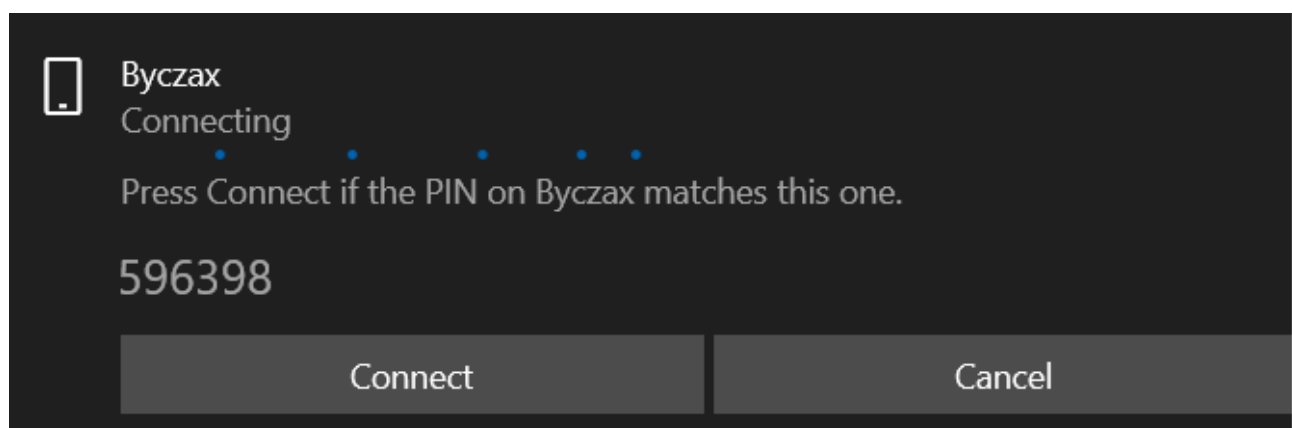
```
PS D:\Documents\Programming\V-Semester\UP-Lab\lab03> python.exe .\zaj3.py
Found 5 devices.
FC:4D:D4:54:F7:95 - PAROG
5C:87:9C:FC:F0:66 - DESKTOP-8I3A000
00:1C:88:10:F8:C1 - PENTA-GPS
7C:F3:1B:77:43:73 - Byczax
7C:7A:91:49:4A:07 - ktp
```

Nazwa naszego urządzenia to "Byczax"

4. Dokonać autoryzacji obu urządzeń:
 - po stronie urządzenia BT autoryzować PC



- po stronie PC autoryzować urządzenie BT.



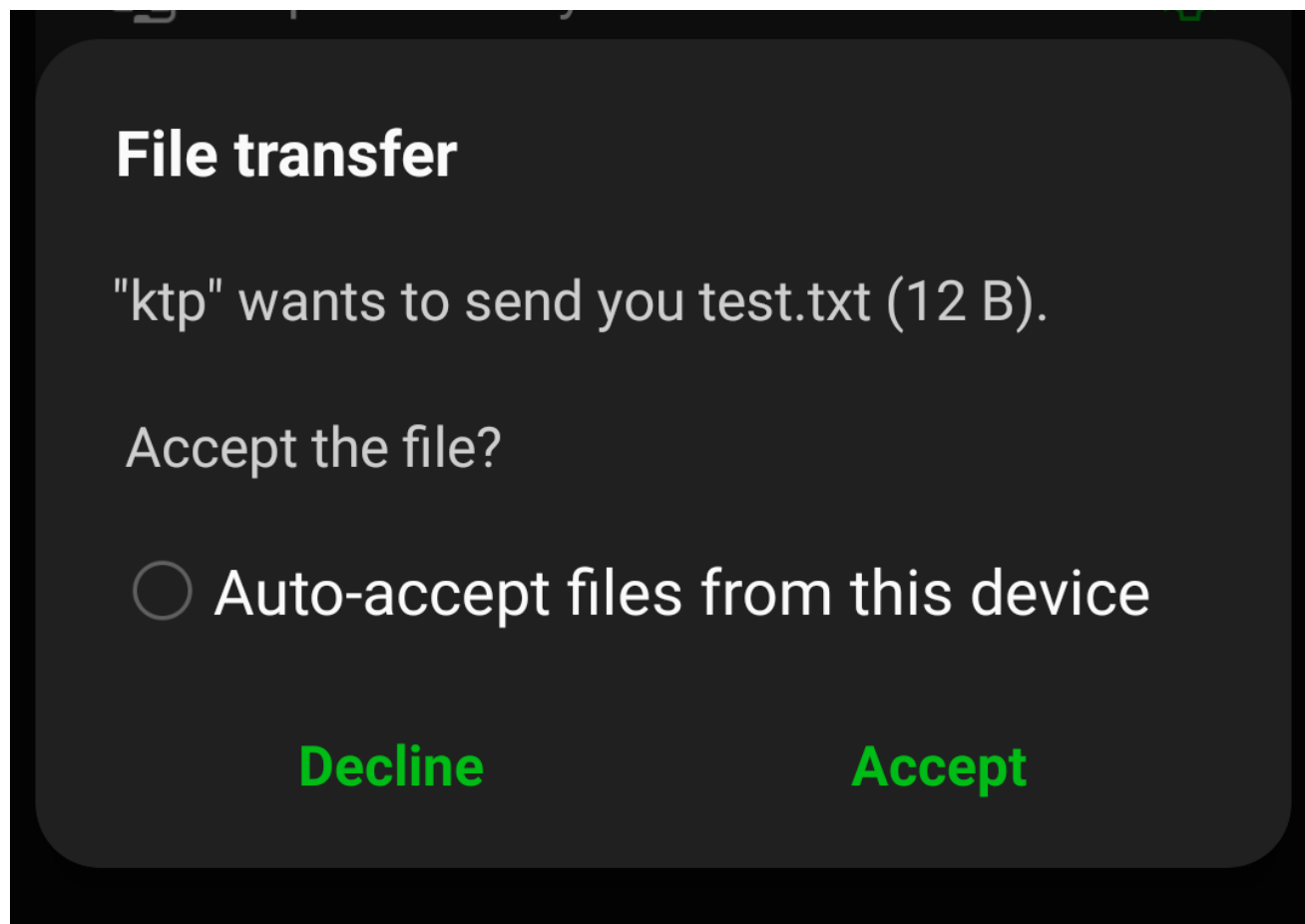
5. Uruchomić urządzenie BT w tryb pracy transferu plików.

```
1 karol@ktp ~/s5/UP/zaj3 % python3 all.py 7C:F3:1B:77:43:73
2 Found 12 services on 7C:F3:1B:77:43:73.
3
4 Service Name: None
5   Host:          7C:F3:1B:77:43:73
6   Description:  None
7   Provided By:  None
8   Protocol:     L2CAP
9   channel/PSM:  31
10  svc classes:  ['1801']
11  profiles:     []
12  service id:   None
13
14 Service Name: None
15   Host:          7C:F3:1B:77:43:73
```

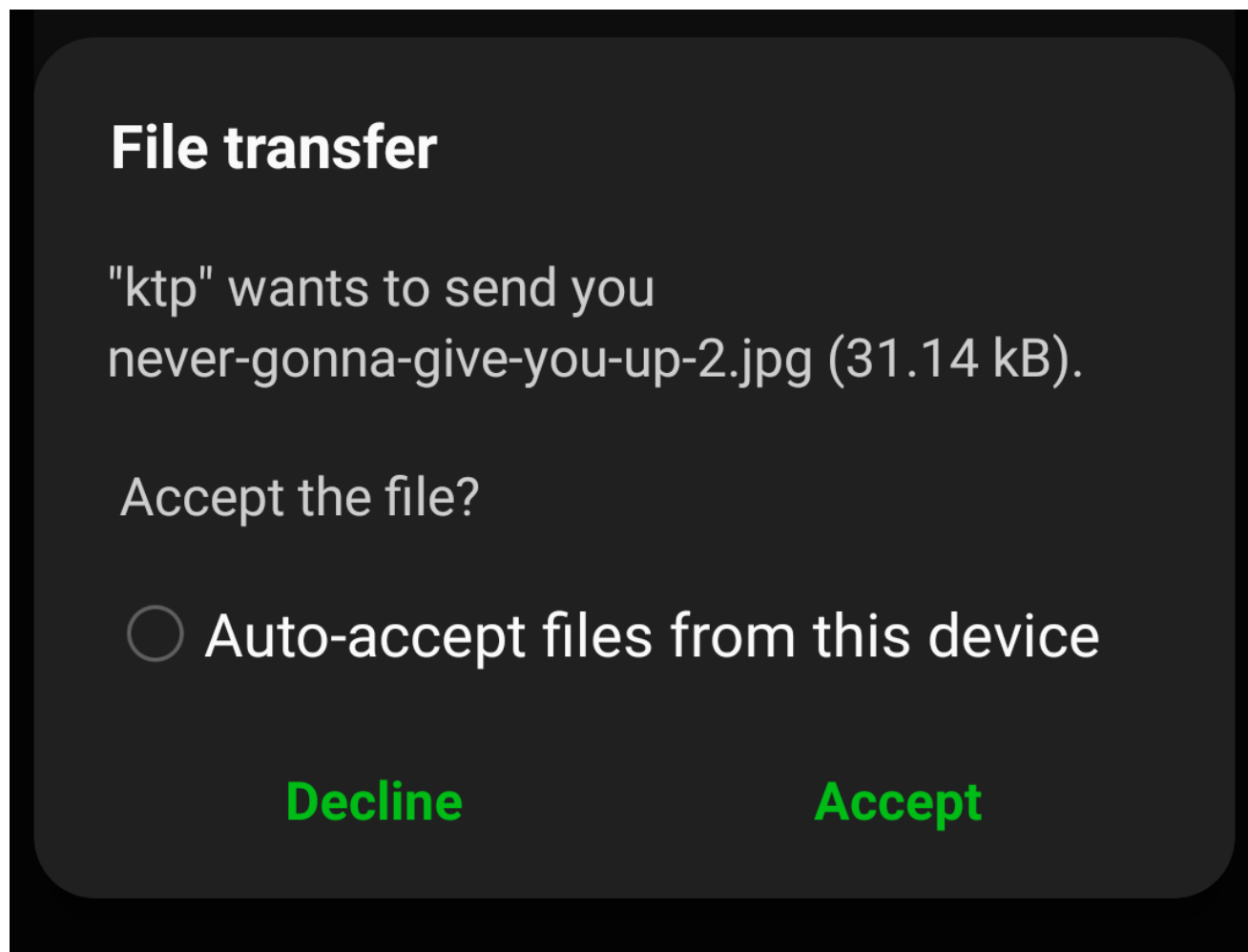
```
16     Description: None
17     Provided By: None
18     Protocol:    L2CAP
19     channel/PSM: 31
20     svc classes: [ '1800' ]
21     profiles:    []
22     service id:  None
23
24 Service Name: Headset Gateway
25     Host:        7C:F3:1B:77:43:73
26     Description: None
27     Provided By: None
28     Protocol:    RFCOMM
29     channel/PSM: 2
30     svc classes: [ '1112', '1203' ]
31     profiles:    [( '1108', 258)]
32     service id:  None
33
34 Service Name: Handsfree Gateway
35     Host:        7C:F3:1B:77:43:73
36     Description: None
37     Provided By: None
38     Protocol:    RFCOMM
39     channel/PSM: 3
40     svc classes: [ '111F', '1203' ]
41     profiles:    [( '111E', 262)]
42     service id:  None
43
44 Service Name: AV Remote Control Target
45     Host:        7C:F3:1B:77:43:73
46     Description: None
47     Provided By: None
48     Protocol:    L2CAP
49     channel/PSM: 23
50     svc classes: [ '110C' ]
51     profiles:    [( '110E', 259)]
52     service id:  None
53
54 Service Name: Advanced Audio
55     Host:        7C:F3:1B:77:43:73
56     Description: None
57     Provided By: None
58     Protocol:    L2CAP
59     channel/PSM: 25
60     svc classes: [ '110A' ]
61     profiles:    [( '110D', 259)]
62     service id:  None
63
64 Service Name: None
65     Host:        7C:F3:1B:77:43:73
66     Description: None
67     Provided By: None
68     Protocol:    L2CAP
69     channel/PSM: 23
70     svc classes: [ '110E', '110F' ]
71     profiles:    [( '110E', 261)]
72     service id:  None
73
74 Service Name: Android Network Access Point
75     Host:        7C:F3:1B:77:43:73
76     Description: NAP
```

```
77     Provided By: None
78     Protocol:     L2CAP
79     channel/PSM: 15
80     svc classes:  ['1116']
81     profiles:     [('1116', 256)]
82     service id:   None
83
84 Service Name: MAP SMS/MMS
85     Host:         7C:F3:1B:77:43:73
86     Description:  None
87     Provided By:  None
88     Protocol:     RFCOMM
89     channel/PSM: 26
90     svc classes:  ['1132']
91     profiles:     [('1134', 258)]
92     service id:   None
93
94 Service Name: OBEX Phonebook Access Server
95     Host:         7C:F3:1B:77:43:73
96     Description:  None
97     Provided By:  None
98     Protocol:     RFCOMM
99     channel/PSM: 19
100    svc classes:  ['112F']
101    profiles:     [('1130', 258)]
102    service id:   None
103
104 Service Name: SIM Access
105     Host:         7C:F3:1B:77:43:73
106     Description:  None
107     Provided By:  None
108     Protocol:     RFCOMM
109     channel/PSM: 16
110     svc classes:  ['112D', '1204']
111     profiles:     [('112D', 258)]
112     service id:   None
113
114 Service Name: OBEX Object Push
115     Host:         7C:F3:1B:77:43:73
116     Description:  None
117     Provided By:  None
118     Protocol:     RFCOMM
119     channel/PSM: 12
120     svc classes:  ['1105']
121     profiles:     [('1105', 258)]
122     service id:   None
123
124
125
```

6. Przesłać plik tekstowy do urządzenia BT.



7. Przesłać plik graficzny do urządzenia BT.



3 Wnioski

Bluetooth jest prosty w użyciu, wystarczyło zmienić rozszerzenie. Cały kod został napisany w pythonie.

Problemy na które się natknęliśmy to:

- Urządzenie wariowało gdy próbowało się podłączyć do kilku urządzeń,
- iPhone-y nie posiadają protokołu OBEX do przesyłania plików.