

Kurs:  
Urządzenia Peryferyjne - Laboratorium

---

## TRANSMISJA WiFi

---

*Autor:*

ALEKSANDRA CHRUSTEK, 263900

JONASZ TROCHA, 263951

Prowadzący:  
dr inż. Dariusz Caban

24.01.2024

# 1 Wstęp

Celem ćwiczenia było zapoznanie się z zasadami bezprzewodowej transmisji WiFi oraz implementacja programu realizującego komunikację pomiędzy urządzeniami wyposażonymi w interfejs WiFi.

## 2 Informacje teoretyczne

### 2.1 Standard WiFi

Standard WiFi to zestaw specyfikacji określających protokoły bezprzewodowej komunikacji lokalnej (WLAN). Najbardziej powszechnymi standardami WiFi są 802.11a, 802.11b, 802.11g, 802.11n, 802.11ac i 802.11ax. Każdy z tych standardów określa zasady transmisji, częstotliwości, prędkości transferu danych itp.

### 2.2 Zasady realizacji transmisji WiFi

#### 2.2.1 Router

Urządzenie, które łączy różne sieci komputerowe, najczęściej łączące sieć lokalną z internetem.

#### 2.2.2 Serwer

Komputer lub system, który dostarcza usługi lub zasoby innym komputerom w sieci.

#### 2.2.3 Klient

Urządzenie, które korzysta z usług serwera.

#### 2.2.4 Access Point (AP)

Urządzenie umożliwiające bezprzewodowe połączenie urządzeń z siecią.

#### 2.2.5 Station

Urządzenie końcowe, takie jak komputer lub smartfon, które łączy się z access pointem.

#### 2.2.6 UDP (User Datagram Protocol)

Protokół warstwy transportowej, który nie gwarantuje dostarczenia pakietów i nie utrzymuje połączenia między nadawcą a odbiorcą.

### **2.2.7 TCP (Transmission Control Protocol)**

Protokół warstwy transportowej, który zapewnia niezawodne, uporządkowane i dostarczone dostarczanie danych.

## **2.3 Środowiska/układy wspierające transmisje WiFi**

### **2.3.1 Android**

System operacyjny dla urządzeń mobilnych, który obsługuje WiFi i zapewnia API dla programistów do korzystania z funkcji sieci bezprzewodowej.

### **2.3.2 ESP01 i ESP32**

Mikrokontrolery firmy Espressif Systems, popularne w projektach IoT, które obsługują WiFi.

### **2.3.3 Arduino IDE**

Środowisko programistyczne dla platformy Arduino, które może być używane do programowania mikrokontrolerów z obsługą WiFi.

## **2.4 Biblioteki wspierające oprogramowywanie transmisji WiFi**

1. Dla ESP01 i ESP32, można używać bibliotek takich jak WiFi.h w przypadku platformy Arduino, które ułatwiają programowanie funkcji sieci bezprzewodowej.
2. W przypadku programowania na platformie Android, istnieją różne klasy i metody dostępne w Android API do obsługi WiFi.
3. Dla ogólnego programowania w języku Python lub innych językach, istnieją także biblioteki, takie jak socket do obsługi transmisji danych poprzez protokoły TCP/UDP.

### 3 Realizacja zadania

Naszym zadaniem było skonfigurować router WiFi, a następnie napisać aplikację, która będzie umożliwiać:

1. aktywację i wyłączenie urządzenia (może to się odbywać automatycznie na stracie i przy wyjściu z programu),
2. uruchomienie oprogramowania realizujące transmisję WiFi,
3. realizację połączenia z wybranym urządzeniem WiFi,
4. przesłanie danych za pomocą WiFi.

Po pierwszym połączeniu z siecią, sposobem tradycyjnym należało skonfigurować router tak, aby nie blokował prób połączenia, przydzielał IP połączonym urządzeniom oraz przydzielił odpowiednie parametry jak nazwa, hasło do sieci. Naszemu routerowi nadaliśmy nazwę 'routerUP'. Aplikacja została napisana w języku Python. Program łącząc się z siecią WiFi za każdym razem wykonuje sekwencję: włączenie urządzenia WiFi i wyświetlenie dostępnych sieci, usunięcie wybranej sieci (jeśli istnieje), stworzenie nowego profilu WLAN dla wybranej sieci, połączenie z wybraną siecią. Do implementacji użyliśmy bibliotek:

1. socket - pozwala zrealizować połączenie za pomocą socketów
2. time - pozwala usypiać program na dany czas
3. os - pozwala używać funkcji i poleceń systemowych

Transfer plików miał zostać zrealizowany przez socket, jednak nie udało nam się go zrealizować. Poniżej przedstawiony został kod aplikacji do obsługi transmisji WiFi.

Klasa other:

```
import os
import time
import socket
import tqdm

SEPARATOR = "<SEPARATOR>"
BUFFER_SIZE = 4096

# os.system("wmic nic get name, index")
# os.system("wmic path win32_networkadapter where index=4 call disable")
# os.system("wmic path win32_networkadapter where index=4 call enable")
# os.system("netsh interface show interface")
```

```

def createNewConnection(name):
    config = """<?xml version=\\"1.0\\"?>
<WLANProfile xmlns=\\"http://www.microsoft.com/networking/WLAN/profile/v1">
<name>""" + name + """</name>
    <SSIDConfig>
        <SSID>
            <name>""" + name + """</name>
        </SSID>
    </SSIDConfig>
<connectionType>ESS</connectionType>
<connectionMode>auto</connectionMode>
<MSM>
    <security>
        <authEncryption>
            <authentication>open</authentication>
            <encryption>none</encryption>
            <useOneX>false</useOneX>
        </authEncryption>
    </security>
</MSM>
</WLANProfile>"""
    command = "netsh wlan add profile filename=\\" + name + ".xml\\" +
    " interface=Wi-Fi"
    with open(name + ".xml", 'w') as file:
        file.write(config)
    os.system(command)

def deleteConnection(name):
    command = "netsh wlan delete profile " + name

def connect(name):
    command = "netsh wlan connect name=\\" + name + "\\" ssid=\\" + name +
    "\\" interface=Wi-Fi"
    os.system(command)

# function to display available Wifi networks
def displayAvailableNetworks():
    os.system("netsh wlan show networks interface=Wi-Fi")

```

```

def enableWiFiCard():
    os.system("netsh interface set interface Wi-Fi enable")
    print("Wlaczono karte sieciowa Wi-Fi")
    time.sleep(5)
    displayAvailableNetworks()

def disableWiFiCard():
    os.system("netsh interface set interface Wi-Fi disable")
    print("Wylaczono karte sieciowa Wi-Fi")

def main():
    enableWiFiCard()
    name_of_router = input('\n\nEnter Name of Wi-Fi network you want to connect to: ')
    deleteConnection(name_of_router)
    time.sleep(3)
    createNewConnection(name_of_router)
    connect(name_of_router)
    while (True):
        choose = input("Choose menu option: ")
        if (choose == "1"):
            print("File transfer")
            s = socket.socket()
            host = socket.gethostname

            # host = input("Where do you want to put your files (ip)?")
            port = 55555
            s.bind((host, port))
            s.listen(5)
            while True:
                c, addr = s.accept()
                # s.connect(('192.168.1.100', port))
                # s.send("Hi")

        if (choose == "0"):
            disableWiFiCard()
            return

main()

```

Klasa main:

```
import os

tab = []
#response = os.system("nmap -sn 192.168.0.0/24")
response = os.system("telnet -o ip-address")
    #if response == 0:
#print ("192.168.0." + str(hostname), 'is up!')
    #tab.append("192.168.0." + str(hostname))
print(response)
```

## 4 Wnioski

Zadanie miało na celu wprowadzenie w obszar programowania urządzeń peryferyjnych, konkretnie WIFI. Realizacja tego zadania okazała się wymagająca, ponieważ wiązała się z koniecznością zaznajomienia się z zasadami działania transmisji oraz jak przebiega ona z perspektywy zaimplementowanej aplikacji. W trakcie laboratorium udało się zaimplementować wszystkie wymagane przez instrukcję metody, z wyjątkiem przesyłania danych. W rezultacie, zadanie dostarczyło nam praktycznej wiedzy na temat programowania urządzeń peryferyjnych, a zdobyte umiejętności mogą być użyteczne w projektach obejmujących obsługę różnorodnych urządzeń wejściowych.



Lab 6  
Transmisja WiFi

Naszym zadaniem było skonfigurowanie routera WiFi.  
Następnie mieliśmy napisać aplikację, która miała umożli-  
wiać:

- aktywację i wyłączenie urządzenia,
- uruchomienie oprogramowania realizującego transmisję WiFi,
- realizację połączenia z wybranym urządzeniem WiFi,
- przesyłanie danych za pomocą WiFi.

Skonfigurowaliśmy urządzenie poprzez zmieniienie jego nazwy na router i zresetujemy ustawienia.

W aplikacji udało nam się zaimplementować:

- aktywację i dezaktywację urządzenia,
- uruchomić kartę sieciową i w terminalu wyświetlić dostępne sieci,
- można połączyć się z siecią wybraną poprzez podanie jej nazwy,
- udało się połączyć z wybranym urządzeniem WiFi.
- przesył danych miał odbywać się przez socket, podaliśmy IP oraz port, program stale nasłuchuje opki przesyłania plików, jednak z jakiegoś powodu nie udaje się plików przysłać.

Cabe

24.01.2024

Bez przesyłania  
danych/plików.