



# Politechnika Wrocławska

---

## Sprawozdanie 5

Ćwiczenie 5. Transmisja WiFi

Krzysztof Zalewa, Wiktor Wojnar

14.1.2025

# Spis treści

<b>1</b>	<b>Wstęp teoretyczny</b>	<b>2</b>
1.1	Standard WiFi . . . . .	2
1.2	Warstwa fizyczna . . . . .	4
1.3	Kanały . . . . .	4
<b>2</b>	<b>Zadanie laboratoryjne</b>	<b>5</b>
2.1	Treść zadania . . . . .	5
2.2	Opis działania programu . . . . .	5
2.3	Schemat sieci . . . . .	5
2.4	Kod programu . . . . .	6
<b>3</b>	<b>Wnioski</b>	<b>9</b>
<b>4</b>	<b>Źródła</b>	<b>9</b>

## 1 Wstęp teoretyczny

### 1.1 Standard WiFi

Standard IEEE 802.11(WiFi - wireless fidelity)to fragment zbioru standardów sieci lokalnych (LAN). Zasięg sieci WiFi to od kilku metrów do kilku kilometrów. Rozwojem tego standardu zajmuje się grupa WiFi Alliance.

Rok publikacji	Generacja	Standard IEE	Częstotliwość[GHz]	Maksymalny przesył[Mbps]
1999	WiFi 1*	802.11b	2.4	1 - 11
1999	WiFi 2*	802.11a	5.0	6 - 54
2003	WiFi 3*	802.11g	2.4	6 - 54
2008	WiFi 4	802.11n	2.4 & 5.0	72 - 600
2013	WiFi 5	802.11ac	5.0	433 - 6 933
2019	WiFi 6	802.11ax	2.4 & 5.0	600 - 9 608
2020	WiFi 6E	802.11ax	2.4 & 5.0 & 6.0	600 - 9 608
2024	WiFi 7**	802.11be	2.4 & 5.0 & 6.0	400 - 23 059

\*Oznaczenie nie są oficjalne (Nie występują w dokumentacji WiFi Alliance)

\*\*Istnieją już urządzenia obsługujące ten standard mimo tego że nadal nie jest oficjalnie zatwierdzony

Tabela 1: Standardy WiFi

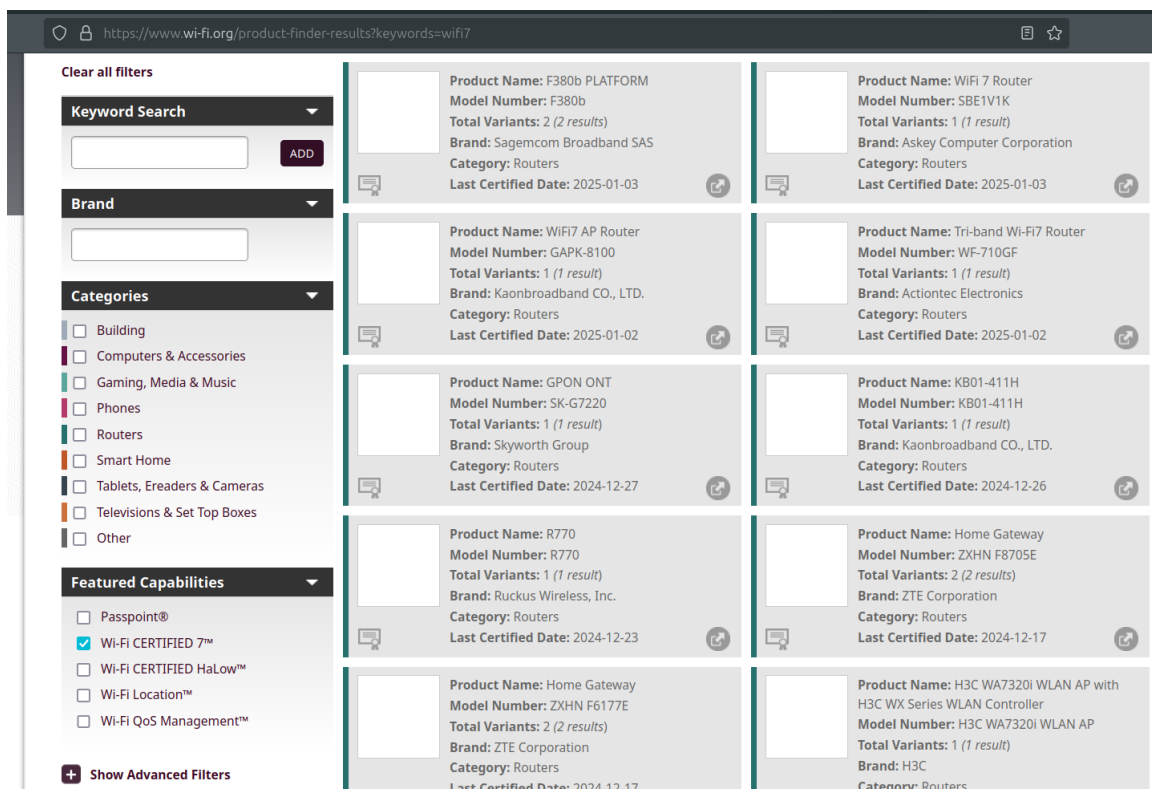
IEEE 802.11be-2024

## IEEE Approved Draft Standard for Information technology-- Telecommunications and information exchange between systems Local and metropolitan area networks--Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment: Enhancements for Extremely High Throughput (EHT)

Last modified: 23 October 2024

This amendment defines standardized modifications to both the IEEE Std 802.11 physical layers (PHY) and the Medium Access Control Layer (MAC) that enable at least one mode of operation capable of supporting a maximum throughput of at least 30 Gbps,... [Expand Description >](#)

Rysunek 1: Zrzut ekranu ze strony Ieee



Rysunek 2: Zrzut ekranu ze strony WiFi Alliance

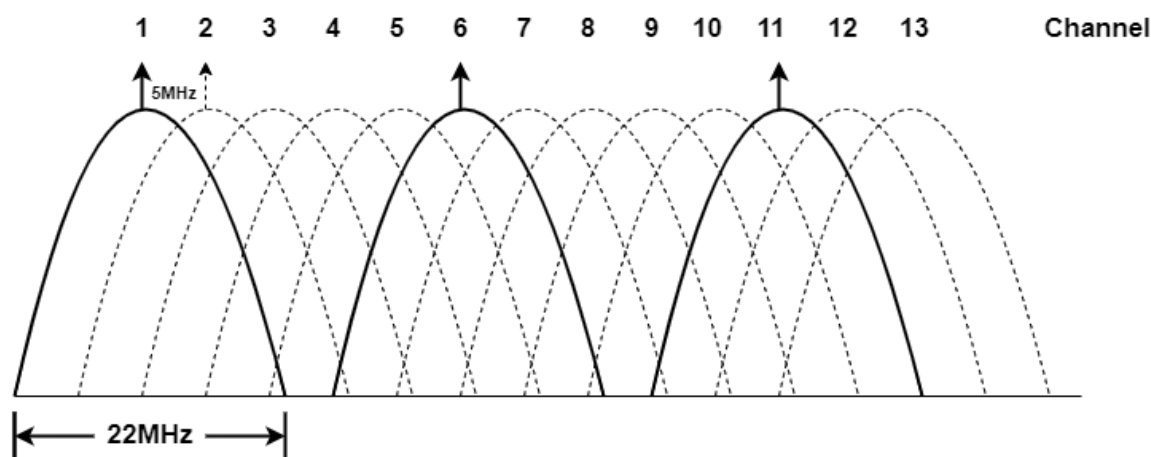
## 1.2 Warstwa fizyczna

Standard IEEE 802.11 specyfikuje kilka protokołów warstwy fizycznej.

1. DSSS (ang. Direct Sequence Spread Spectrum) Jest techniką modulacji widmem rozproszonym której szerokość pasma nadawanego jest znacznie większa niż wymagana. Sygnał o małej gęstości mocy w mniejszym stopniu zakłóca działanie innych systemów na tym samym paśmie. Jako że sygnał jest wysyłany równocześnie na wielu częstotliwościach jest on redundantny i może być odtworzony mimo zakłuceń.
2. FHSS (ang. Frequency Hopping Spread Spectrum) Jest techniką w której częstotliwość nadawania zmienia się w regularnych odstępach czasu. Czas kolejnych skoków jest na tyle długi by można było przesłać kilka bitów wiadomości. Widmo tego sygnału zajmuje szerokie pasmo i nie musi być ciągle co pozwala na omijanie częstotliwości. Przed transmisją nadawca i odbiorca muszą ustalić wzór skoków.
3. OFDM (ang. Orthogonal Frequency-Division Multiplexing) Zamiast transmitowania jednego strumienia o dużej szybkości można przesłać wiele o niższej prędkości.

## 1.3 Kanały

Według standardu IEEE 802.11 zapewnione jest kilka pasm używanych w komunikacji WiFi. Pasma dzielą się na kanały z których korzystają poszczególne sieci. Najczęściej używane pasmo to 2.4GHz (Ma rozpiętość od 2.4GHz do 2.5GHz). Pasma to jest podzielone na 14 kanałów. Jednakże nie wszystkie są używane np w Polsce korzystamy z kanałów 1 - 13 a w USA korzysta się z kanałów 1 - 11.



Rysunek 3: Kanały pasma 2.4GHz

Kanały są rozmieszczone co 5MHz i mają szerokość 20 MHz. Nadawanie na sąsiednich kanałach może powodować zakłucenia. Dlatego zaleca się używanie kanałów 1,6,11 (Ich zakres nigdy się nie pokrywa). W przypadku gdy w okolicy jest więcej niż trzy sieci WiFi ma kilka możliwości rozwiązania problemu.

1. Wykożystanie pasma 5GHz które ma znacznie więcej kanałów które się nie nakładają (od 19 do 23). Jednakże problemem może być krótszy zasięg.
2. Dynamiczny wybór pasma pozwala routerowi na zmianę pasma na którym nadaje.
3. Wykożystanie protokołu CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) router sprawdza czy pasmo z którego chce skorzystać jest zajęte i w razie potrzeby czeka na "swoją kolej".

## 2 Zadanie laboratoryjne

### 2.1 Treść zadania

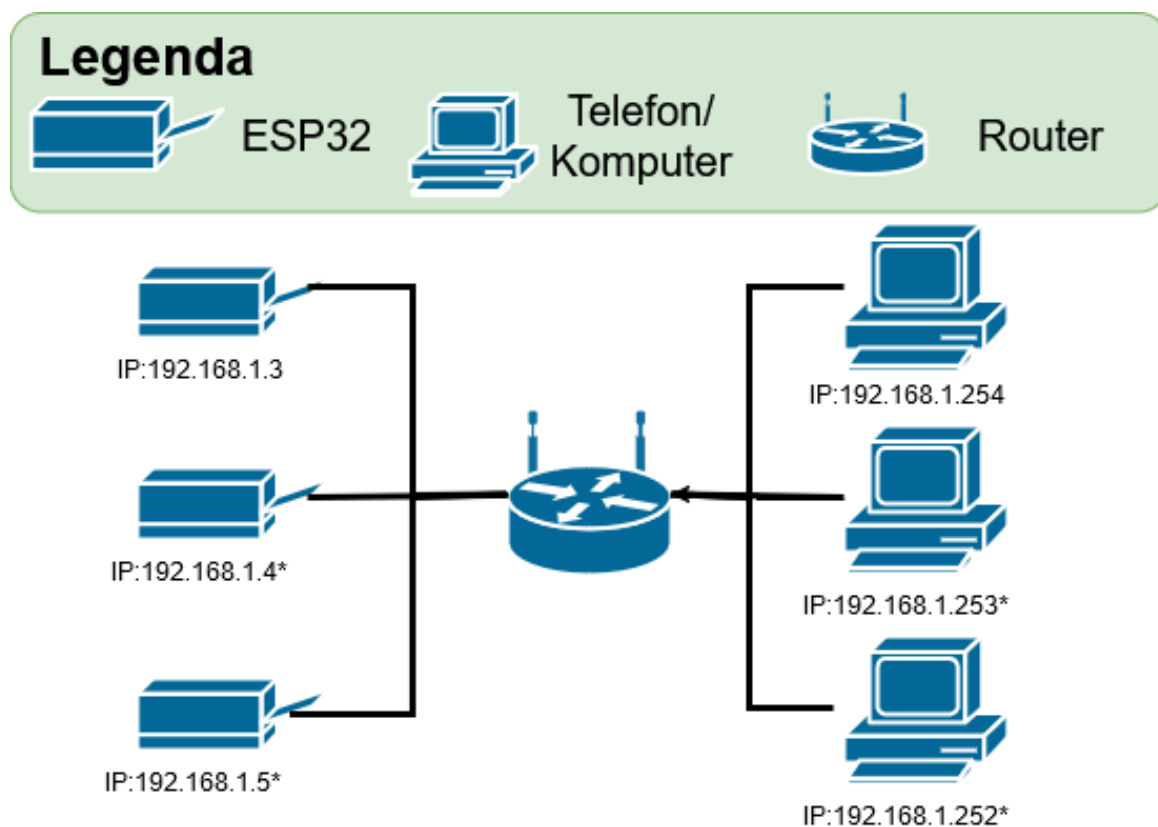
W ramach zadania laboratoryjnego należało skonfigurować router WiFi, uruchomić układ SoC. Następnie należało napisać program dzięki któremu można będzie:

1. Włączyć i wyłączyć urządzenie
2. Uruchomić transmisję WiFi
3. Połączyć się z wybranym urządzeniem WiFi
4. Przesłać dane za pomocą WiFi

### 2.2 Opis działania programu

Program wyszukuje wszystkie dostępne sieci WiFi i wypisuje ich ssid. Następnie łączy się z siecią o ssid maszt\_sygnalowy. Po poprawnym połączeniu włączany jest serwer na porcie 21. Następnie na serwer wysyłana jest prosta strona https na której rysuje się trójkąt.

### 2.3 Schemat sieci



*\* Na zajęciach podłączony był jeden układ ESP32 i jeden telefon ale powinna być możliwość dodania wielu urządzeń.*

Rysunek 4: Schemat sieci

**Dane sieci:**

1. **Nazwa:** maszt\_sygnalowy

2. **Kanał:** 6
3. **IP:** od 192.168.1.1 do 192.168.1.255
4. **Adresy:** 253 (-1 adres sieci,-1 adres bramy domyślnej,-1 adres rozgłoszeniowy)
5. **Maska:** 255.255.255.0
6. **Brama domyślna:** 192.168.1.2
7. **Urządzenia ESP32:** na adresach 192.168.1.3 do 192.168.128(rosnąco)
8. **Urządzenia klienckie:** na adresach 192.168.1.254 do 192.168.129(malejąco)

## 2.4 Kod programu

```
1  #include <WiFi.h>
2  #include <SPI.h>
3
4  const int ledPin = 5;
5  const char ssid[] = "maszt_sygnalowy";
6  int status;
7
8  WiFiServer server(21);
9
10 void setup() {
11     Serial.begin(115200);
12
13     uint8_t networkList = WiFi.scanNetworks();
14
15     for(uint8_t i =0;i<networkList;i++)
16         Serial.printf("%d -> %s\n", i+1, WiFi.SSID(i));
17
18     // check for the presence of the shield:
19     if (WiFi.status() == WL_NO_SHIELD) {
20         Serial.println("WiFi shield not present");
21         // don't continue:
22         exit(0);
23     }
24
25     IPAddress ip(192, 168, 1, 2);
26     WiFi.config(ip);
27
28     WiFi.begin(ssid);
29     Serial.printf("\nConnecting\n");
30     while(WiFi.status() != WL_CONNECTED)
31     {
32         Serial.printf(".");
33         delay(100);
34     }
35     Serial.printf("Connection succesfull\n");
36
37
38     server.begin();
39     Serial.print("Connected to wifi. My address:");
40     IPAddress myAddress = WiFi.localIP();
```

```

42   Serial.println(myAddress);
43 }
44
45 uint8_t x =0;
46 void loop() {
47
48     // listen for incoming clients
49
50     WiFiClient client = server.available();
51
52     if (client) {
53
54         Serial.println("new client");
55
56         // an http request ends with a blank line
57
58         bool currentLineIsBlank = true;
59
60         while (client.connected()) {
61
62             if (client.available()) {
63
64                 char c = client.read();
65
66                 Serial.write(c);
67
68                 // if you've gotten to the end of the line (received a newline
69                 // character) and the line is blank, the http request has ended,
70                 // so you can send a reply
71
72                 if (c == '\n' && currentLineIsBlank) {
73
74                     // send a standard http response header
75
76                     client.println("HTTP/1.1 200 OK");
77
78                     client.println("Content-Type: text/html");
79
80                     client.println("Connection: close"); // the connection will be closed
81                     ↪ after completion of the response
82
83                     client.println("Refresh: 5"); // refresh the page automatically every 5
84                     ↪ sec
85
86                     client.println();
87
88                     client.print("<!DOCTYPE html>\n"
89                                "<html lang=\"en\">\n"
90                                "<head>\n"
91                                "<meta charset=\"UTF-8\">\n"
92                                "<meta name=\"viewport\" content=\"width=device-width,"
93                                ↪ initial-scale=1.0\">\n"
94                                "<title>Increasing Numbers with Delay</title>\n"
95                                "</head>\n"

```

```

95     "<body>\n"
96     "<h1>Increasing Numbers</h1>\n"
97     "<div id=\"numberDisplay\"></div>\n"
98     "<script>\n"
99     "const displayElement = document.getElementById('numberDisplay');\n"
100    "let cspace = \"&nbsp;\";\n"
101    "let cleft = \"/\";\n"
102    "let cright = \"\\\\\\\\\";\n"
103    "let ctop = \"^\";\n"
104    "let cbottom = \"-\";\n"
105    "let count = 0;\n" // Start count at 0 for the first iteration
106    "const intervalId = setInterval(() => {\n"
107    "    var str = \" \";\n"
108    "    if (count === 0) {\n"
109    "        str = cspace.repeat(20) + ctop + cspace.repeat(20);\n"
110    "    } else if (count === 20) {\n"
111    "        str = cbottom.repeat(41);\n"
112    "    } else {\n"
113    "        str = cspace.repeat(20 - count) + cleft + cspace.repeat(count
    ↪ * 2) + cright + cspace.repeat(20 - count);\n"
114    "    }\n"
115    "    displayElement.innerHTML += str + '<br>'; // Append the current
    ↪ count\n"
116    "    count++;\n"
117    "    if (count > 20) { // Stop the interval after reaching 41\n"
118    "        clearInterval(intervalId);\n"
119    "    }\n"
120    "}, 200); // 200 milliseconds\n"
121    "</script>\n"
122    "</body>\n"
123    "</html>\n");
124
125    break;
126
127    }
128
129    if (c == '\n') {
130
131        // you're starting a new line
132
133        currentLineIsBlank = true;
134
135    } else if (c != '\r') {
136
137        // you've gotten a character on the current line
138
139        currentLineIsBlank = false;
140
141    }
142
143    }
144
145    }
146
147    // give the web browser time to receive the data
148

```



```
149     delay(15);
150
151     // close the connection:
152
153     client.stop();
154
155     Serial.println("client disconnected");
156
157 }
158 }
159
```

Fragment kodu 1: Fragment kodu z programu

### 3 Wnioski

### 4 Źródła

1. <https://www.wi-fi.org/discover-wi-fi>
2. <https://standards.ieee.org/ieee/802.11be/7516/>
3. <https://www.youtube.com/watch?v=QSq1TYTiESA>
4. <https://www.tp-link.com/pl/support/faq/4309/>
5. <https://www.audyt-wifi.pl/index.php/2024/04/05/kanaly-wifi-w-europie-sa-inne-niz-w-usa/>
6. [https://www.dipol.com.pl/warstwy\\_fizyczne\\_w\\_standardzie\\_ieee\\_802\\_11\\_bib511.htm](https://www.dipol.com.pl/warstwy_fizyczne_w_standardzie_ieee_802_11_bib511.htm)