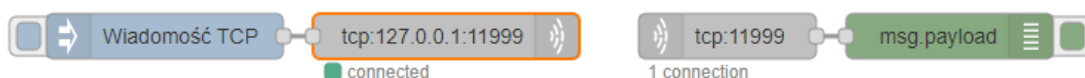


### 3. Protokoły TCP, UDP

#### Protokół TCP

TCP (ang. *Transmission Control Protocol*) to połączeniowy, niezawodny, strumieniowy protokół komunikacyjny stosowany do przesyłania danych między procesami uruchomionymi na różnych maszynach. Protokół TCP operuje w warstwie transportowej modelu OSI. Na rysunku 3.1 przedstawiono przykład lokalnego połączenia TCP zrealizowanego za pomocą programu Node-RED.



Rys. 3.1. Przykład lokalnego połączenia z wykorzystaniem protokołu TCP

W tabeli 3.1 przedstawiono bloczki Node-RED użyte w grafie z rysunku 3.1.

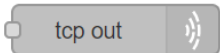
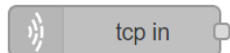
	Bloczek wysyła bufor danych do gniazda zdefiniowanego przez numer IP i numer portu, wykorzystuje protokół TCP.
	Bloczek odbiera bufor danych z wybranego portu na komputerze lokalnym, wykorzystuje protokół TCP.

Tabela 3.1. Bloczki Node-RED użyte w grafie z rys. 3.1

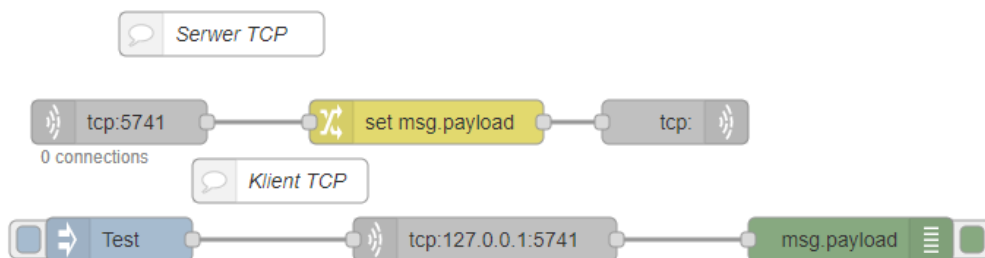
Przez połączenie zrealizowane w na rysunku 3.1 przesyłany jest tekst „Wiadomość TCP” wysyłany przez bloczek inject, odebrany tekst jest wyświetlany w okienku debugera. Protokół TCP nie określa formatu przesyłanej treści, przesyłany jest bufor danych. Program po stronie odbiorczej musi wiedzieć co zrobić z przesyłanymi danymi na podstawie tego co otrzyma. Na rysunku 3.2 przedstawiony jest przykład połączenia TCP przesyłającego plik zawierający obraz.



Rys. 3.2. Przesłanie pliku graficznego z wykorzystaniem protokołu TCP

Na rysunku 3.3 przedstawiono graf realizujący funkcję serwera TCP i klienta korzystający z tego serwera.

Projekt „SezAM wiedzy, kompetencji i umiejętności” jest współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego w ramach Programu Operacyjnego Wiedza Edukacja Rozwój



Rys. 3.3. Graf realizujący serwer TCP i klienta TCP

Na listingu 3.1 przedstawiono kod grafu z rysunku 3.3.

**Listing 3.1.** Kod grafu z rysunku 3.3

```
[{"id":"b7a1fe27.b67ea","type":"tab","label":"Broadcast","disabled":false,"info":"","z":"d00c33fa.975e5","type":"debug","z":"b7a1fe27.b67ea","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":false,"statusVal":"","statusType":"auto","x":570,"y":317,"wires":[]}, {"id":"6afff0bc.4673b","type":"udp in","z":"b7a1fe27.b67ea","name":"","iface":"","port":"13757","ipv":"udp4","multicast":"false","group":"192.168.0.255","datatype":"utf8","x":400,"y":317,"wires":[{"id":"d00c33fa.975e5"}]}, {"id":"a3ec806a.b0bda","type":"udp in","z":"b7a1fe27.b67ea","name":"","iface":"","port":"13757","ipv":"udp4","multicast":"false","group":"225.0.18.83","datatype":"utf8","x":400,"y":357,"wires":[{"id":"f5694416.d426f8"}]}, {"id":"f5694416.d426f8","type":"debug","z":"b7a1fe27.b67ea","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":false,"statusVal":"","statusType":"auto","x":570,"y":357,"wires":[]}, {"id":"bdb54661.cd2b18","type":"udp out","z":"b7a1fe27.b67ea","name":"","addr":"255.255.255.255","iface":"","port":"13757","ipv":"udp4","outport":"","base64":false,"multicast":"broad","x":340,"y":200,"wires":[]}, {"id":"cf7f404e.131da","type":"inject","z":"b7a1fe27.b67ea","name":"","props":[{"p":"payload"}, {"p":"topic","vt":"str"}],"repeat":"","crontab":"","once":false,"onceDelay":0.1,"topic":"","payload":"Broadcast","payloadType":"str","x":140,"y":200,"wires":[{"id":"bdb54661.cd2b18"}]}, {"id":"3f85a565.c52e3a","type":"udp out","z":"b7a1fe27.b67ea","name":"","addr":"192.168.1.255","iface":"","port":"13757","ipv":"udp4","outport":"","base64":false,"multicast":"broad","x":330,"y":240,"wires":[]}, {"id":"8f07337c.9c337","type":"inject","z":"b7a1fe27.b67ea","name":"","props":[{"p":"payload"}, {"p":"topic","vt":"str"}],"repeat":"","crontab":"","once":false,"onceDelay":0.1,"topic":"","payload":"Broadcast","payloadType":"str","x":140,"y":240,"wires":[{"id":"3f85a565.c52e3a"}]}, {"id":"7177844b.9449dc","type":"comment","z":"b7a1fe27.b67ea","name":"Nadawcy","info":"","x":350,"y":160,"wires":[]}, {"id":"926190df.01c11","type":"comment","z":"b7a1fe27.b67ea","name":"Odbiorcy","info":"","x":480,"y":280,"wires":[]}]
```

W tabeli 3.2 przedstawiono nowe bloczki Node-RED użyte w grafie z rysunku 3.3.

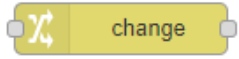

	Bloczek zmienia właściwości obiektu <code>msg</code> . Bloczek może ustawić nową wartość dowolnej właściwości obiektu <code>msg</code> , zmienić właściwość, usunąć właściwość lub zmienić nazwę właściwości.
	Bloczek klienta wysyła żądanie TCP o treści zawartej w <code>msg.payload</code> na wejściu, odbiera odpowiedź serwera i umieszcza ją w <code>msg.payload</code> na wyjściu.

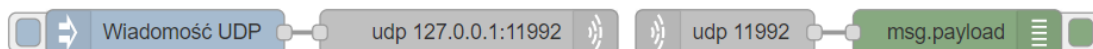
Tabela 3.2. Bloczki Node-RED użyte w grafie z rys. 3.3

Serwer pracuje na porcie 5741 po odebraniu żądania bloczek `change` zmienia `msg.payload` i zmienioną wartość odsyła jako odpowiedź.

## Protokół UDP

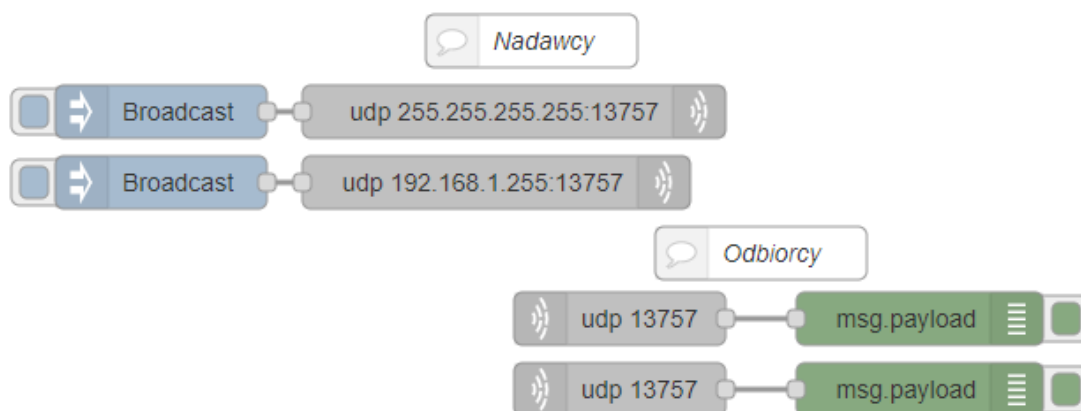
Protokół UDP (ang. *User Datagram Protocol* – protokół pakietów użytkownika) stosowany jest w warstwie transportowej modelu OSI. Nie gwarantuje dostarczenia datagramu. Jest to protokół bezpołączeniowy, nie ma narzutu na nawiązywanie połączenia i śledzenie sesji. Uproszczenie względem TCP zapewnia szybszą transmisję. Protokół UDP umożliwia *multicast* – transmisję danych do więcej niż jednego komputera.

Na rysunku 3.4 przedstawiono przykładowe lokalne połączenie UDP w trybie *unicast*, w którym istnieje tylko jeden nadawca i tylko jeden odbiorca.



Rys. 3.4. Przykład lokalnego połączenia z wykorzystaniem protokołu UDP

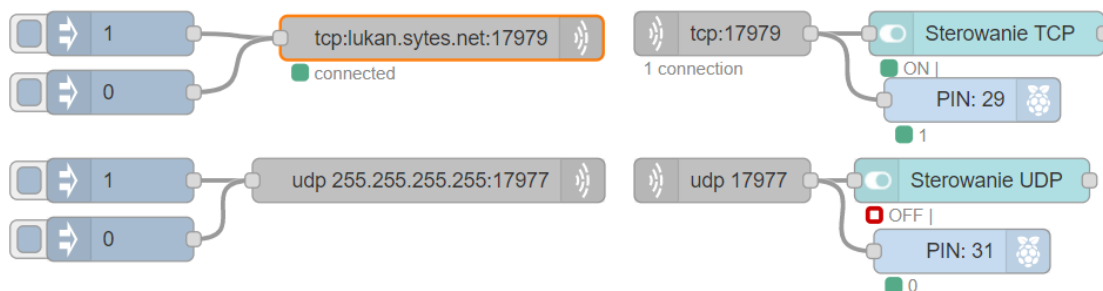
Na rysunku 3.5 przedstawiono przykładowe lokalne połączenie UDP w trybie *broadcast*, w którym jeden nadawca wysyła wiadomość, która powinna być odebrana wszystkie hosty przyłączone do danej domeny rozgłoszeniowej.



Rys. 3.5. Graf realizujący 2 nadajniki rozgłoszeniowe UDP i dwóch klientów UDP

Adres 255.255.255.255 używany przez pierwszego nadawcę jest adresem do wszystkich hostów w sieci lokalnej (LAN), można go używać nie znając adresu sieci lokalnej.

Na rysunku 3.6 przedstawiono przykład zastosowania połączenia TCP i UDP do zdalnego włączania i wyłączania urządzeń.



Rys. 3.6. Zastosowania połączenia TCP i UDP do zdalnego włączania i wyłączania urządzeń

Na listingu 3.2 przedstawiono kod grafu z rysunku 3.6.

**Listing 3.2.** Kod grafu z rysunku 3.6

```
[{"id":"85dd0f0b.096f8","type":"tab","label":"Sterowanie TCP i UDP","disabled":false,"info":""},{
  "id":"cbb28442.41fff8","type":"tcp out","z":"85dd0f0b.096f8","host":"127.0.0.1","port":"17979","beserver":"client","base64":false,"end":false,"name":"","x":320,"y":120,"wires":[],
  {"id":"4bf66e01.2881","type":"inject","z":"85dd0f0b.096f8","name":"","props":[{"p":"payload"}],
  {"p":"topic","vt":"str"},"repeat":"","crontab":"","once":false,"onceDelay":0.1,"topic":"","payload":"1","payloadType":"str","x":90,"y":120,"wires":[["cbb28442.41fff8"]]},
  {"id":"1feca662.20108a","type":"tcp in","z":"85dd0f0b.096f8","name":"","server":"server","host":"","port":"17979","datamode":"stream","datatype":"utf8","newline":"","topic":"","base64":false,"x":500,"y":120,"wires":[["586ce0a1.27293"]]},
  {"id":"7a056608.787d78","type":"inject","z":"85dd0f0b.096f8","name":"","props":[{"p":"payload"}],
  {"p":"topic","vt":"str"},"repeat":"","crontab":"","once":false,"onceDelay":0.1,"topic":"","payload":"0","payloadType":"str","x":90,"y":160,"wires":[["cbb28442.41fff8"]]},
  {"id":"586ce0a1.27293","type":"ui_switch","z":"85dd0f0b.096f8","name":"","label":"Sterowanie TCP","tooltip":"","group":"a4eee380.d5245","order":3,"width":0,"height":0,"passthru":false,"decouple":"true","topic":"","style":"","onvalue":"1","onvalueType":"str","onicon":"","oncolor":"","offvalue":"0","offvalueType":"str","officon":"","offcolor":"","x":680,"y":120,"wires":[[]]},
  {"id":"72b4243c.79292c","type":"inject","z":"85dd0f0b.096f8","name":"","props":[{"p":"payload"}],
  {"p":"topic","vt":"str"},"repeat":"","crontab":"","once":false,"onceDelay":0.1,"topic":"","payload":"1","payloadType":"str","x":90,"y":220,"wires":[["19f3c832.cc3fb8"]]},
  {"id":"c073287d.87e2c8","type":"inject","z":"85dd0f0b.096f8","name":"","props":[{"p":"payload"}],
  {"p":"topic","vt":"str"},"repeat":"","crontab":"","once":false,"onceDelay":0.1,"topic":"","payload":"0","payloadType":"str","x":90,"y":260,"wires":[["19f3c832.cc3fb8"]]}]
```

Projekt „SezAM wiedzy, kompetencji i umiejętności” jest współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego w ramach Programu Operacyjnego Wiedza Edukacja Rozwój

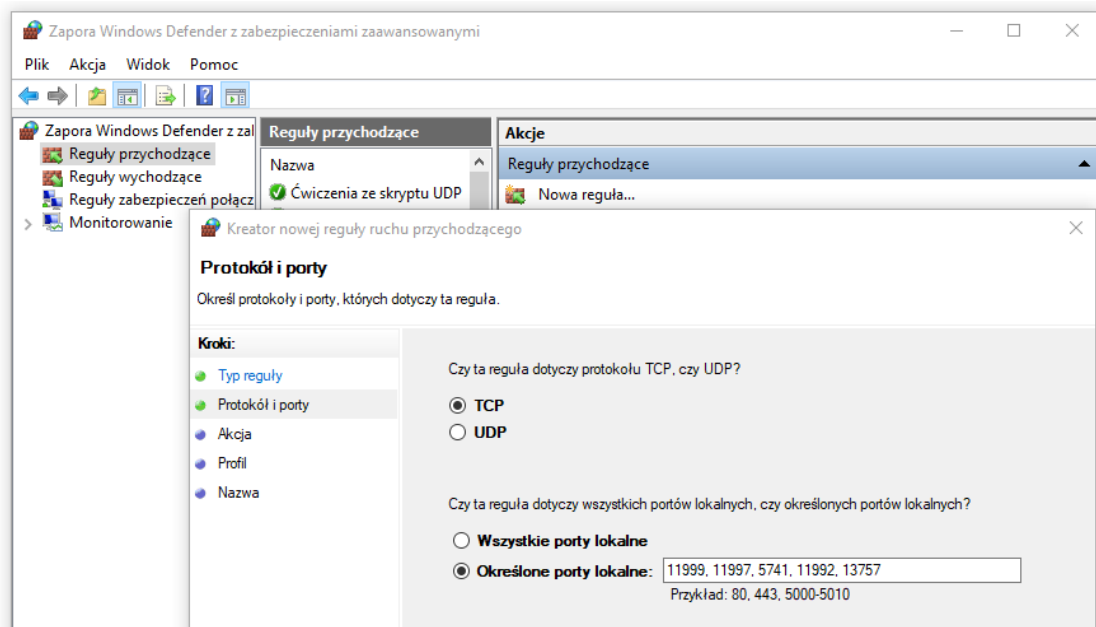
```
3c832.cc3fb8"]]],{"id":"d142c449.3e8cd8","type":"udp
in","z":"85dd0f0b.096f8","name":"","iface":"","port":"17977","ipv":"udp4","multi
cast":"false","group":"","datatype":"utf8","x":500,"y":220,"wires":[["a0254a64.5
80038"]]],{"id":"19f3c832.cc3fb8","type":"udp
out","z":"85dd0f0b.096f8","name":"","addr":"255.255.255.255","iface":"","port":"
17977","ipv":"udp4","outport":"","base64":false,"multicast":"broad","x":300,"y":
220,"wires":[]},{id:"a0254a64.580038","type":"ui_switch","z":"85dd0f0b.096f8",
"name":"","label":"Sterowanie
UDP","tooltip":"","group":"a4eee380.d5245","order":3,"width":0,"height":0,"passt
hru":false,"decouple":"true","topic":"","style":"","onvalue":"1","onvalueType":"
str","onicon":"","oncolor":"","offvalue":"0","offvalueType":"str","officon":"","
offcolor":"","x":670,"y":220,"wires":[[]]},{id:"a4eee380.d5245","type":"ui_gro
up","z":"","name":"Przełączniki","tab":"565a7e08.a7db3","order":4,"disp":true,"wi
dth":"6","collapse":false},{id:"565a7e08.a7db3","type":"ui_tab","z":"","name":
"Home","icon":"dashboard","disabled":false,"hidden":false}]}
```

Urządzenia na grafie z rysunku 3.6 są przedstawione jako przełączniki na pulpicie i piny Raspberry Pi. Jeśli część odbiorcza działa na Raspberry Pi to do pinów 29 i 31 można podłączyć przekaźniki, które będą włączały i wyłączały urządzenia. Górna część grafu włącza/wyłącza selektywnie urządzenie na hoście lukan.sytes.net, dolna część grafu włącza/wyłącza grupowo wszystkie urządzenia dołączone do hostów w sieci lokalnej. Odbiorcza część grafu powinna być umieszczona na hostach, które mają być sterowane.

Protokoły TCP i UDP są protokołami warstwy transportowej. Z wykorzystaniem tych protokołów przesyła się buforów danych, ale protokoły te nie określają co jest w tych danych. Obie strony łączności, nadawcza i odbiorcza, muszą wiedzieć co należy nadać i co zrobić z tym co zostało odebrane. W prostym przykładzie z rysunku 1.5 przyjęto, że przesyłane są stany logiczne „1” i „0”. Do prostego włączania i wyłączania urządzeń to wystarczy, ale przy bardziej złożonych systemach trzeba przyjąć bardziej złożony protokół. Taki protokół można wymyślić samemu lub zastosować znany, wypróbowany i posiadający wsparcie, na przykład protokół HTTP.

## Ćwiczenie 3.1. Realizacja połączeń TCP i UDP

1. W programie Node-RED zrealizować lokalne połączenia TCP i UDP przedstawione na rysunkach 3.1, 3.2, 3.3 i 3.4.
2. Zmienić adres lokalnego komputera 127.0.0.1 na adres sąsiedniego komputera w laboratorium, zrealizować i przetestować połączenia. Przy realizacji tego punktu może być wymagana zmiana ustawień Zapory Windows Defender i utworzenia nowej reguły przychodzącej, aby otworzyć wybrane porty dla połączeń przychodzących, rysunek 3.7. Przy pracy zdalnej (z domu) skonfigurowanie połączeń wymaga dodatkowo znajomości publicznych numerów IP i właściwego skonfigurowania routera.



Rys. 3.7. Tworzenie nowej reguły przychodzącej Zapory Windows Defender

3. Zrealizować graf z rysunku 3.5 przetestować połączenie na komputerze lokalnym następnie w sieci lokalnej.
4. Zrealizować i przetestować połączenie TCP na komputerze lokalnych tworząc program w języku C# w Visual Studio lub Visual Studio Code, wykorzystać kod z listingu 3.3.

**Listing 3.3.** Kod lokalnego połączenia TCP w języku C#

```
static void Main(string[] args)
{
    Socket serverSocket = new Socket(AddressFamily.InterNetwork,
SocketType.Stream, ProtocolType.Tcp);
    Socket clientSocket = new Socket(AddressFamily.InterNetwork,
SocketType.Stream, ProtocolType.Tcp);
    Socket internalSocket;
    byte[] recBuffer = new byte[256];
    serverSocket.Bind(new IPEndPoint(IPAddress.Parse("127.0.0.1"), 1024));
    serverSocket.Listen(1);
    clientSocket.Connect("127.0.0.1", 1024);
    internalSocket = serverSocket.Accept();
    clientSocket.Send(ASCIIEncoding.ASCII.GetBytes("Wiadomosc!"));
    internalSocket.Receive(recBuffer);
    Console.WriteLine(ASCIIEncoding.ASCII.GetString(recBuffer));
}
```

Projekt „SezAM wiedzy, kompetencji i umiejętności” jest współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego w ramach Programu Operacyjnego Wiedza Edukacja Rozwój

5. Zrealizować w języku C# zdalne połączenia do grafów z rysunków 3.2 3.4. Wykorzystać kod z listingu 3.4. Połączenia przetestować na hoście lukan.sytes.net, lokalne i w sieci.

**Listing 3.4.** Kod połączeń TCP i UDP do odbiorców z rysunku 3.2 i 3.4 umieszczonych na hoście lukan.sytes.net

```
static void Main(string[] args)
{
    System.Net.Sockets.UdpClient udpClient = new UdpClient(); // Klient UDP
    byte[] dane = ASCIIEncoding.ASCII.GetBytes("Wiadomosc UDP");
    udpClient.Send(dane, dane.Length, "lukan.sytes.net", 11992);
    udpClient.Close(); // Koniec klienta UDP
    TcpClient tcpClient = new TcpClient(); // Klient TCP
    tcpClient.Connect("lukan.sytes.net", 11999); // Próba połączenia
    BinaryWriter writer = new BinaryWriter(tcpClient.GetStream()); // Dane
    writer.Write("Wiadomość TCP");
    writer.Close();
    tcpClient.Close(); // koniec klienta TCP
}
```

6. Zrealizować w języku C# zdalne połączenia rozgłoszeniowe do grafów z rysunków 3.4 3.5. Wykorzystać kod z listingu 3.5. Przetestować połączenia sieci lokalnej.

**Listing 3.5.** Kod połączeń rozgłoszeniowych UDP do odbiorców z rysunku 3.4 i 3.5

```
System.Net.Sockets.UdpClient udpClient = new UdpClient(); // Klient UDP
byte[] dane = ASCIIEncoding.ASCII.GetBytes("Wiadomosc UDP");
udpClient.Send(dane, dane.Length, "255.255.255.255", 13757);
udpClient.Send(dane, dane.Length, "255.255.255.255", 11992);
udpClient.Close();
```

## Dyskusja

Protokół TCP jest protokołem warstwy transportowej i realizuje transport bufora danych pomiędzy hostami. Protokół nie określa formatu tego co jest przesyłane. W przykładach ze skryptu za pomocą tego protokołu przesyłano dane w różnych formatach, aby wykorzystać dane odbiorca musiał wiedzieć co jest przesyłane, tekst, plik graficzny czy plik audio. W prostych systemach, gdzie projektantem obu stron łączności jest ten sam podmiot takie rozwiązanie może być wystarczające. W złożonych systemach, gdzie są różni producenci programów uczestniczących w łączności, pojawia się potrzeba uzgodnień pomiędzy programami. Przykładem mogą być serwery WWW produkowane przez różnych producentów i łączące się z nimi przeglądarki różnych producentów.

Protokół HTTP, który będzie tematem kolejnego rozdziału, określa format tego co jest przesyłane za pomocą protokołu TCP. Protokół HTTP określa jak klient informuje serwer czego i w jakim formacie oczekuje. Informacja jest zawarta w adresie URL, nagłówkach i wybranej metodzie żądania. W odpowiedzi serwera nagłówki informują o statusie odpowiedzi i o formacie przesyłanych danych.

Projekt „SezAM wiedzy, kompetencji i umiejętności” jest współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego w ramach Programu Operacyjnego Wiedza Edukacja Rozwój