
1 Komunikacja z systemem wbudowanym

1	Komunikacja z systemem wbudowanym	1
1.1	Kanały komunikacji, protokoły, programy	2
1.1.1	Zdalny interpreter poleceń shell	4
1.1.2	Przesyłanie plików	6
1.2	Wykorzystanie terminala szeregowego RS232 do komunikacji z systemem wbudowanym	7
1.3	Komunikacja z systemem wbudowanym z wykorzystaniem protokołu TCP/IP	10
1.3.1	Informacje ogólne	10
1.3.2	Adresowanie w sieci TCP/IP	11
1.3.3	Testowanie połączenia z systemem docelowym	14
1.3.4	Usługa i protokół telnet	15
1.3.5	Usługa i protokół SSH	18
1.3.6	Przesyłanie plików, protokoły FTP, SCP, SFTP	20
1.4	IDE - zintegrowane środowiska uruchomieniowe	27

1.1 Kanały komunikacji, protokoły, programy

System wbudowany nie posiada zwykle rozbudowanych interfejsów komunikacji z użytkownikiem. Bywa że wyposażony jest w pewną ilość diod sygnalizacyjnych, kilka przycisków czy przełączników, być może posiada prosty wyświetlacz LCD.

Należy zapewnić sobie możliwość komunikacji z systemem wbudowanym.

Występuje konieczność:

- Testowania prawidłowości działania sprzętu i instalacji systemu operacyjnego
- Konfigurowania systemu operacyjnego i sterowników urządzeń
- Testowanie i uruchamianie oprogramowania aplikacyjnego

Aby to przeprowadzić należy zapewnić sobie możliwość wykonywania w systemie wbudowanym następujących czynności:

- Przesyłanie plików z komputera macierzystego na docelowy (a także w drugą stronę)
- Poruszania się po systemie plików
- Edycji plików, w szczególności plików konfiguracyjnych systemu i aplikacji
- Uruchamiania programów i obserwacji skutków ich działania

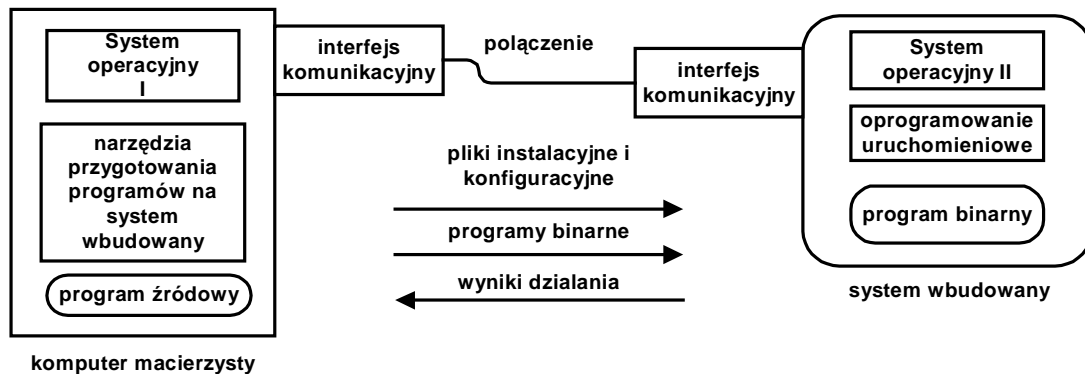
W celu zapewnienia takiej komunikacji należy określić:

- Medium komunikacyjne
- Protokół komunikacji
- Oprogramowanie działające po stronie komputera macierzystego
- Oprogramowanie działające po stronie komputera wbudowanego

Jako medium komunikacyjne najczęściej stosuje się:

- Interfejs RS232, RS484
- Sieć Ethernet
- Połączenie USB
- Połączenie JTAG

Protokół komunikacji związany jest zwykle z medium komunikacyjnym..



Rys. 1-1 Komunikacja komputera macierzystego z z systemem wbudowanym

Typową potrzebą jest możliwość uruchamiania w systemie wbudowanym programów i obserwacja rezultatów ich działania. Aby uruchomić program na systemie wbudowanym należy wykonać następujące kroki:

1. Utworzyć program przeznaczony na system wbudowany. Zazwyczaj program przygotowany jest w komputerze macierzystym .
2. Przesłać program do systemu wbudowanego.
3. Spowodować jego uruchomienie
4. Obserwować wyniki jego działania

Wymienione wyżej czynności realizowane są zazwyczaj przez oddzielne aplikacje działające według specyficznych protokołów. W niektórych przypadkach faza ta (edycja i/lub kompilacja) może odbywać się w środowisku systemu wbudowanego, (Raspberry Pi, BeagleBone Black).

Oprogramowanie działające po stronie komputera macierzystego to:

- Emulator terminala tekstowego,
- Klient programu uruchomieniowego np. `gdb GNU debugger`
- Zintegrowane środowisko uruchomieniowe jak np. Eclipse.

Oprogramowanie działające po stronie komputera wbudowanego to:

- Interpreter poleceń (ang. *shell*),
- Serwer popularnych usług przesyłania plików (FTP, SFTP, SCP),
- Program uruchomieniowy np. `gdbserver` lub innego rodzaju agent.

1.1.1 Zdalny interpreter poleceń shell

Standardowym narzędziem umożliwiającym wykonanie wymienionych wyżej poleceń jest interpreter poleceń czyli `shell`.

Interpreter poleceń przyjmuje polecenia w postaci łańcuchów ze standardowego wejścia, interpretuje te polecenia a wyniki przesyła na standardowe wyjście.

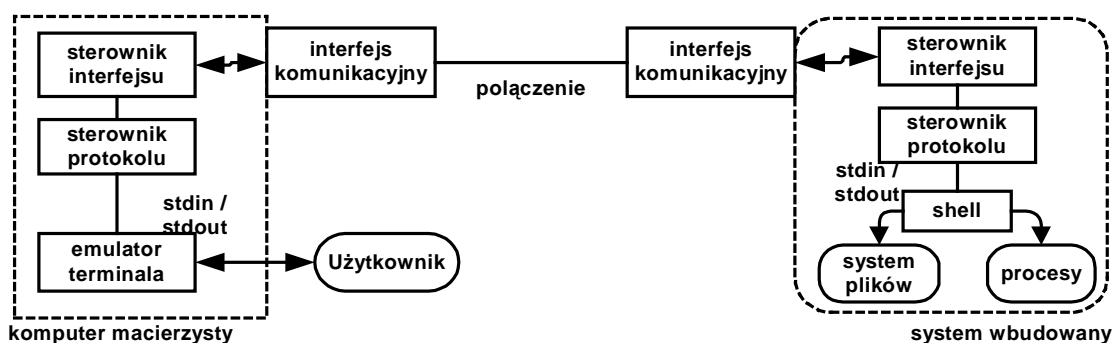
Polecenia polegają zazwyczaj na uruchamianiu programów (lub tak zwanych poleceń wewnętrznych) co wymaga możliwości poruszania się po systemie plików a także odczytu i zapisu tych plików.

Standardowe wejście / wyjście interpretera poleceń wysyła i odbiera znaki które muszą być dostarczone do komputera macierzystego. Odbywa się to za pomocą pewnego protokołu komunikacyjnego który z kolei komunikuje się z jakimś sterownikiem interfejsu a finalnie z samym interfejsem komunikacyjnym (urządzeniem).

Protokół komunikacyjny:

- zapewnia synchronizację nadawcy i odbiorcy,
- dokonuje detekcji i być może korekcji błędów
- pełni inne funkcje

Odbierane ze sterownika protokołu znaki muszą być prezentowane użytkownikowi co odbywa się za pośrednictwem programu nazywanego emulatorem terminala szeregowego. Program terminalowy wyświetla przychodzące ze standardowego wejścia znaki, odbiera znaki pisane przez operatora na klawiaturze, dokonuje ich translacji i wysyła na standardowe wyjście.



Rys. 1-2 Komunikacja z systemem wbudowanym za pośrednictwem emulatora terminala

Program Midnight Commander umożliwia zaawansowane zarządzanie plikami a także kopiowanie plików pomiędzy systemami z wykorzystaniem protokołów FTP i SCP.

```

Left      File      Command  Options  Right
-----
<--      .[^]>    <--      .[^]>
'n      Name      Size     Modify   time    'n      Name      Size     Modify   time
UP--DIR Jan 1 00:45
/.cache 4096 Jan 1 00:16
/.config 4096 Jan 1 00:16
/.gconf 4096 Jan 1 00:00
/.gnome2 4096 Jan 1 00:01
/.gvfs 4096 Jan 1 00:00
/.local 4096 Jan 1 00:00
/.nautilus 4096 Jan 1 00:01
/.npm 4096 Jan 1 02:18
/.pulse 4096 Jan 1 00:00
/.vnc 4096 Jan 1 2000
/Desktop 4096 Jan 1 00:01
/gpio 4096 Oct 30 2013
/vnc 4096 Jan 1 00:46
.ICEauthority 990 Jan 1 00:00

UP--DIR |----- 192M/1697M (11%) -----
UP--DIR|----- 192M/1697M (11%) -----

Hint: Key frequently visited ftp sites in the hotlist: type C-\.
#[^]
1Help 2Menu 3View 4Edit 5Copy 6RenMov 7Mkdir 8Delete 9PullDn 10Quit

```

Ekran 1-1 Program Midnight Commander wykonywany na komputerze BeagleBone Black

1.1.2 Przesyłanie plików

Zwykle aplikacje mające być wykonane w systemie wbudowanym, przygotowywane są na komputerze macierzystym. Występuje zatem potrzeba przesłania programów z komputera macierzystego do systemu wbudowanego.

Funkcję tę realizują różnego rodzaju aplikacje, ale opierają się zwykle na standardowych protokołach przesyłania plików. Rodzaje protokołów zależą od rodzaju kanału transmisyjnego.

Dla kanałów opartych o RS232, RS485 i pochodnych stosuje się takie protokoły:

- Xmodem,
- Zmodem
- Kermit .

Dla kanałów opartych o TCP/IP wykorzystuje się protokoły:

- FTP (ang. *File Transfer Protocol*),
- SFTP (ang. *Secure File Transfer Protocol*),
- SCP.

W każdym z tych przypadków po stronie systemu wbudowanego musi być uruchomiony serwer danej usługi a po stronie komputera macierzystego odpowiadający mu program kliencki.

1.2 Wykorzystanie terminala szeregowego RS232 do komunikacji z systemem wbudowanym

Najprostszym sposobem skomunikowania się z systemem wbudowanym jest połączenie poprzez interfejs RS232.

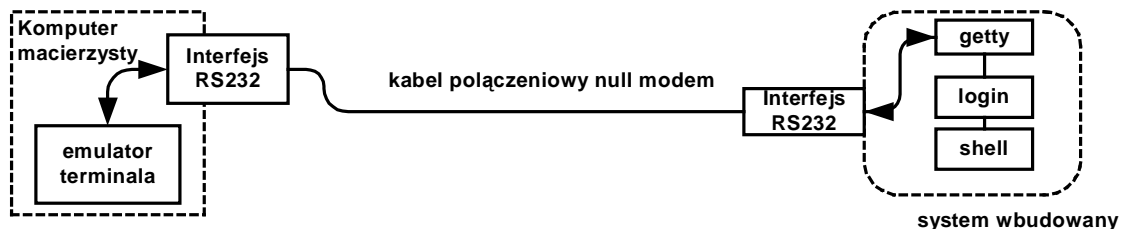
Jest ono od dawna stosowane do komunikacji z systemami wbudowanymi.

Zaletą takiego rozwiązania jest jego prostota. Układ UART (ang. *Universal Asynchronous Receiver Transmitter*) sterujący transmisją szeregową jest nieskomplikowany i dlatego spotkać go można nawet w najprostszycy mikrokontrolerach.

Nie wymaga on żadnej infrastruktury poza prostym kablem typu nullmodem.

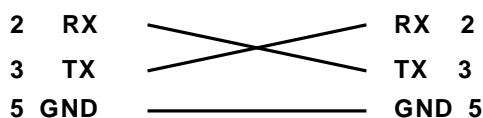
Także oprogramowanie sterujące jest nieskomplikowane. Maksymalna szybkość transmisji uzależniona jest od sprzętu a także od długości kabla połączeniowego.

Przy krótkim kablu szybkość transmisji może osiągać 115200 znaków na sekundę



Rys. 1-3 Połączenie komputera macierzystego z systemem wbudowanym poprzez interfejs RS232

Połączenie pomiędzy komputerem macierzystym a konwerterem powinno być wykonane kablem typu null modem

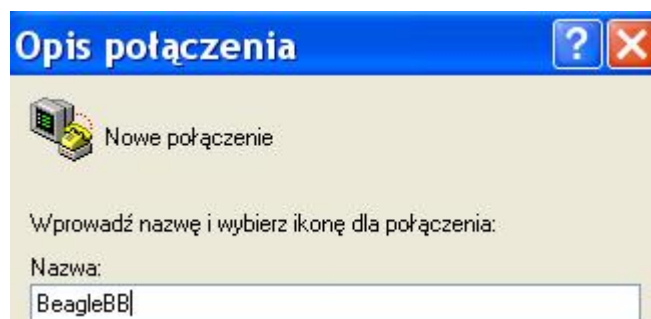


Rys. 1-4 Połączenie typu NULL MODEM dla złącza DB9

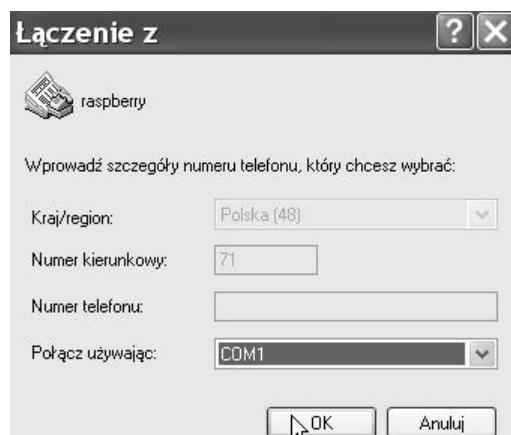
Po stronie komputera macierzystego należy użyć emulatora terminala szeregowego. W systemie Windows użyty może być program Hyper Terminal.

Sposób postępowania:

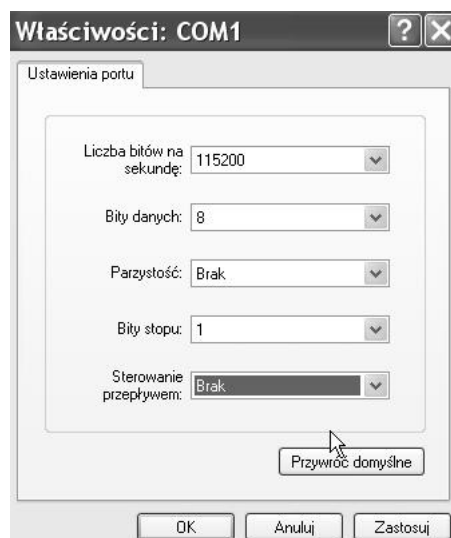
1. Uruchamiamy HyperTerminal
2. Tworzymy nowe połączenie wpisując jego nazwę jak w przykładzie poniżej i klikamy OK.



3. Wybieramy port komunikacyjny, na poniższym rysunku jest to port COM1 a następnie klikamy OK.



4. Ustalamy parametry transmisji: Szybkość transmisji 115200 bit/sek, bitów danych 8, parzystość brak, sterowanie przepływem – brak i klikamy w przycisk OK.



5. Włączamy zasilanie do komputera docelowego. Po chwili na ekranie zaczynają pojawiać się komunikaty. Gdy wpisujemy identyfikator `root` i hasło możemy zalogować się do systemu.

```
Angstrom v2012.12 - Kernel 3.8.13

beaglebone login: root
Password:
Last login: Sat Jan  1 01:21:13 UTC 2000 on tty00
root@beaglebone:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 90:59:AF:56:DA:2F
          inet addr:192.168.0.3  Bcast:192.168.0.255  Mask:255.255.255.0
          ...
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          ...
usb0      Link encap:Ethernet  HWaddr 6E:5E:6E:A6:9B:7D
          inet addr:192.168.7.2  Bcast:192.168.7.3  Mask:255.255.255.252
```

Przykład 1-1 Testowanie adresu IP systemu wbudowanego w komputerze BBB

W systemie Linux dostępnych jest wiele programów emulacji terminala szeregowego jak:

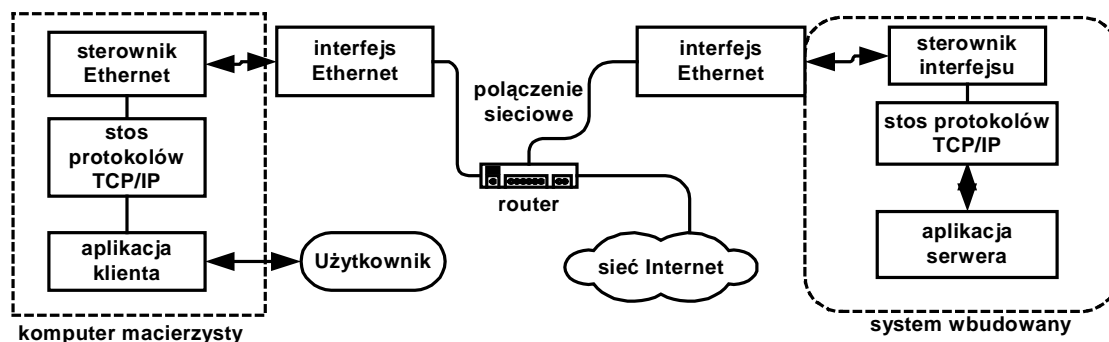
- PuTTY,
- Picocomm,
- Gkterm.

Do bardziej znanych należy program `minicom` który dołączony jest do większości dystrybucji Linuxa, w tym do Debiana.

1.3 Komunikacja z systemem wbudowanym z wykorzystaniem protokołu TCP/IP

1.3.1 Informacje ogólne

Do komunikacji z systemami wbudowanymi typowo wykorzystuje się sieć lokalną Ethernet i połączenie poprzez protokół TCP/IP. Aby zrealizować takie połączenie należy dodatkowo dysponować routerem i kablem połączeniowym Ethernet (ang. *patchcord*) co pokazuje Rys. 1-5.



Rys. 1-5 Komunikacja z systemem wbudowanym za pośrednictwem protokołu TCP/IP

Połączenie TCP/IP realizuje warstwy od 1 do 4 modelu ISO OSI czyli fizyczną, łącza, sieci i transportową. Nie określa jednak protokołu sesji, prezentacji i aplikacyjnego. Typ wykonywanych aplikacji uzależniony jest od pojawiających się w trakcie tworzenia systemów wbudowanych potrzeb.

W inżynierii systemów wbudowanych powszechnie stosowane są następujące aplikacje:

- Do realizacji funkcji interpretera poleceń `shell` protokoły telnet i SSH
- Do przesyłania plików protokoły FTP, SFTP i SCP

W oparciu o protokół TCP/IP działają też zintegrowane systemy tworzenia oprogramowania. W systemach takich przesyła się programy wynikowe na system docelowy a następnie tam uruchamia i obserwuje wyniki ich działania.

1.3.2 Adresowanie w sieci TCP/IP

Komunikacja w domenie Internetu oparta jest o protokół TCP/IP i wykorzystuje specyficzny mechanizm adresowania oparty o koncepcję adresu IP i portu. Adres w dziedzinie Internetu według najpopularniejszego obecnie specyfikacji IPv4 składa się z:

- Numeru Portu – liczba 16 bitowa z przedziału 0-65535
- Adresu IP maszyny – liczba 32 bitowa

Adres IP maszyny jest liczbą 32 bitową. W nowszej wersji protokołu oznaczonej jako IPv6 adresami są to liczby 128 bitowe. Adresy internetowe są zwykle zapisywane jako czwórki rozdzielone kropkami. Każda czwórka odpowiada jednemu bajtowi.


Na przykład:

Zapis z 156.17.24.42
kropką
Szesnastkowo 0x7D11182A
Dziesiętnie 2098272298

Istnieje wiele sposobów by zbadać jaki adres IP posiada nasz komputer. W systemie Linux można sprawdzić jaki adres IP ma nasz komputer wykonując polecenie `ifconfig`.

```
$/sbin/ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:3e:ce:3f
          inet addr:192.168.141.132  Bcast:192.168.141.255
          Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe3e:ce3f/64 Scope:Link
...
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
...

```




Przykład 1-2 Testowanie adresu IP komputera w systemie Linux

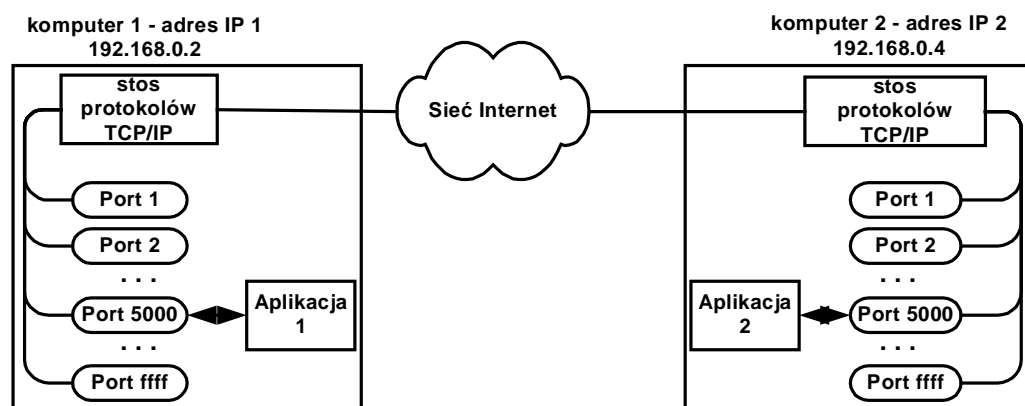
Testowanie adresu IP w systemie Windows - należy uruchomić wiersz poleceń a następnie wykonać polecenie `ipconfig`.

```
C:\Documents and Settings\juka>ipconfig
Konfiguracja IP systemu Windows
Sufiks DNS konkretnego połączenia : dom
Adres IP. . . . . : 192.168.0.3
Maska podsieci. . . . . : 255.255.255.0
Brama domyślna. . . . . : 192.168.0.1

```



Przykład 1-3 Testowanie adresu IP komputera w systemie Windows



Rys. 1-6 Adresowanie w sieci Internet

Działające w systemie Linux aplikacje sieciowe mają przypisane na stałe numery portów. Można je obejrzeć przeglądając plik `/etc/services`.

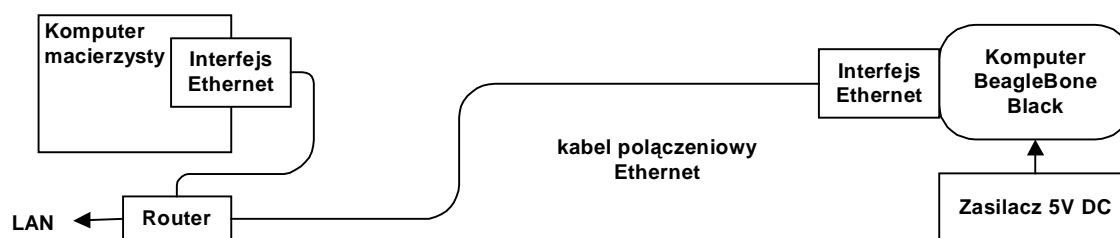
```

echo          7/tcp
echo          7/udp
daytime      13/tcp
daytime      13/udp
netstat      15/tcp
ftp-data     20/tcp          # protokół FTP - dane
ftp          21/tcp          # protokół FTP - sterowanie
ssh          22/tcp          # SSH Remote Login Protocol
ssh          22/udp
telnet       23/tcp          # Telnet
...
nameserver  42/tcp          name          # IEN 116
tftp        69/udp          # Trivial ftp
www         80/tcp          http          # WorldWideWeb HTTP
www         80/udp          # HyperText Transfer Protocol

```

Ekran 1-2 Fragment pliku `/etc/services` – system Linux

Komunikacja TCP/IP z komputerem BeagleBone Black



Rys. 1-7 Połączenie komputera macierzystego z komputerem BeagleBone Black poprzez sieć Ethernet

Powstaje pytanie w jaki sposób można uzyskać adres IP nowo uruchomionemu systemowi.

Można tu:

- Skorzystać z możliwości routera
- Wykorzystać terminal szeregowy.

The screenshot shows the 'Status Klienta' (Client Status) page of a router's administrative interface. It features a table titled 'Lista Klientów' (Client List) with the following data:

Typ	Nazwa	LAN IP	Priorytet	Zablokuj
	roma(Urządzenie...)	192.168.0.2	Normalny	
	beaglebone	192.168.0.3	Normalny	

Ekran 1-3 Uzyskanie adresu IP komputera beaglebone za pomocą ekranu administracyjnego routera

1.3.3 Testowanie połączenia z systemem docelowym

Gdy znamy już adres IP systemu docelowego należy sprawdzić prawidłowość połączenia sieciowego z tym systemem. Do tego celu służy program `ping` po którym wpisujemy adres IP systemu docelowego.

`ping adres_ip_komputera`

Program `ping` dostępny jest zarówno w systemie Windows jak i Linux

```
juka@maria:~$ ping 192.168.0.3
PING 192.168.0.3 (192.168.0.3) 56(84) bytes of data.
64 bytes from 192.168.0.3: icmp_req=1 ttl=128 time=1.58 ms
64 bytes from 192.168.0.3: icmp_req=2 ttl=128 time=1.11 ms
64 bytes from 192.168.0.3: icmp_req=3 ttl=128 time=0.748 ms
^C
--- 192.168.0.3 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 0.748/1.148/1.585/0.343 ms
```

Ekran 1-4 Testowanie połączenia z systemem docelowym za pomocą polecenia ping w systemie Linux

W przypadku gdy nie ma połączenia z systemem docelowym należy sprawdzić:

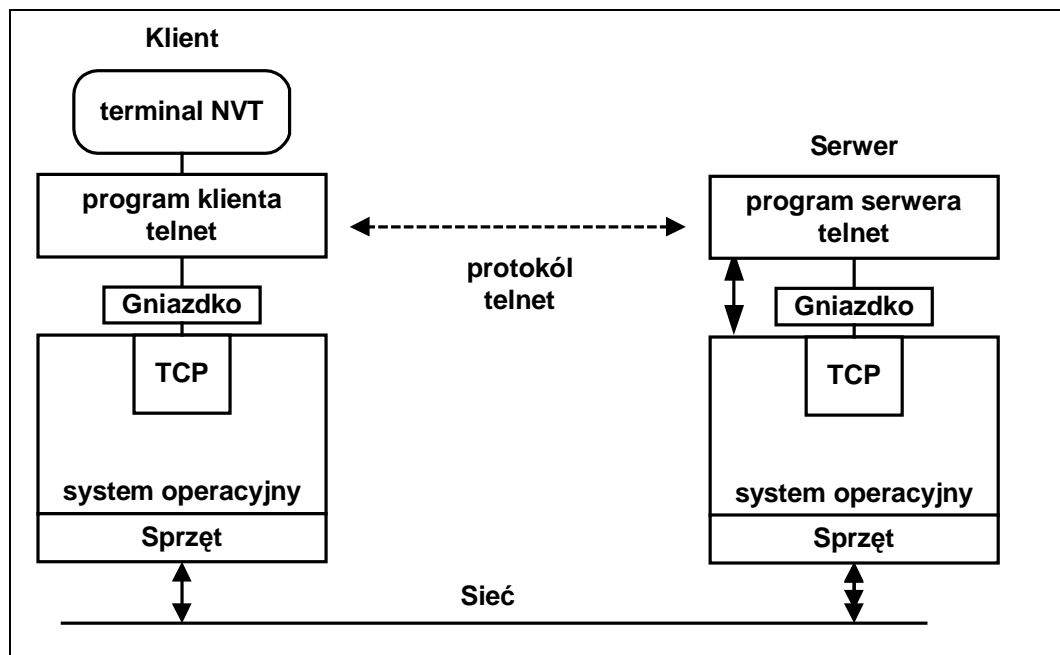
- Prawidłowość działania interfejsu sieciowego po stronie komputera macierzystego
- Prawidłowość połączenia kablowego pomiędzy systemem docelowym a routerem. Na routerze powinna palić się dioda sygnalizująca obecność kabla w odpowiednim gnieździe.
- Prawidłowość działania systemu docelowego, np. łącząc się z nim poprzez terminal szeregowy RS232

1.3.4 Usługa i protokół telnet

Telnet jest protokołem sieciowym używanym w Internecie i sieciach lokalnych. Protokół telnet jest jednym z najstarszych protokołów sieciowych, opracowany został w 1969 i opisany w RFC 15 i RFC 854. Zapewnia on komunikację ze zdalnym systemem poprzez wirtualny terminal implementowany przez program klienta.

W zamierzeniu telnet ma umożliwić komunikację pomiędzy różnymi terminalami i systemami operacyjnymi. Terminale posiadają różne właściwości i znaki sterujące. Podobnie z programami obsługującymi terminale po stronie komputera. Protokół telnet oparty jest na następujących zasadach:

- Koncepcji wirtualnego terminala sieciowego NVT (ang. Network Virtual Terminal)
- Procedurze negocjacji opcji pomiędzy terminalem NVT a obsługującym go procesem po stronie zdalnego komputera

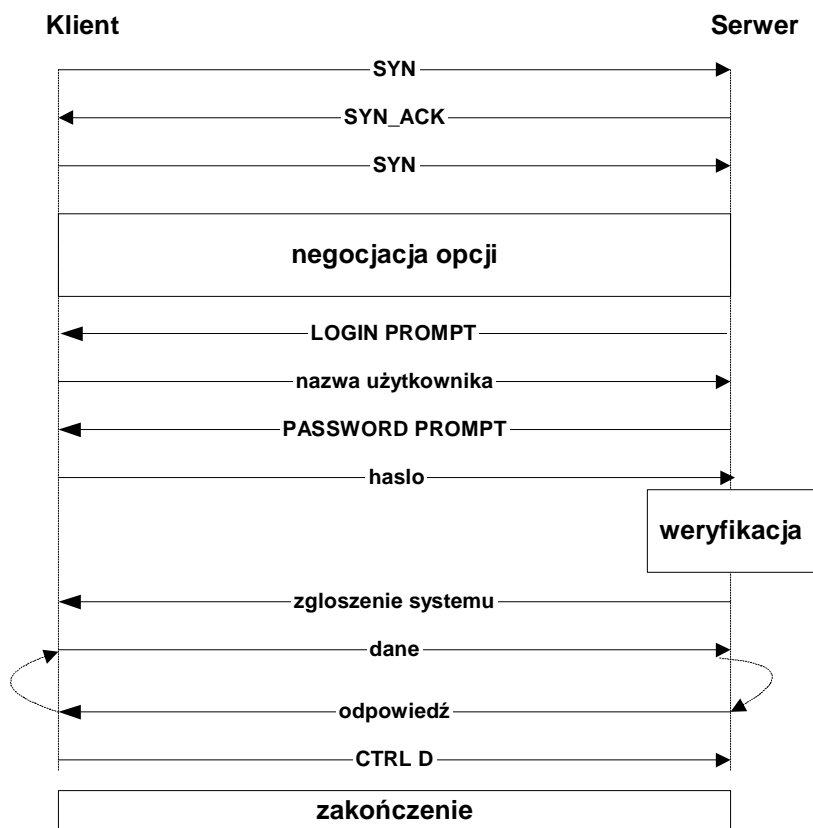


Rys. 1-8 Koncepcja dostępu do zdalnego systemu poprzez protokół telnet

Komunikacja jest 8 bitowa, dwukierunkowa realizowany w warstwie TCP. Przesyłane znaki są kodowane 7 bitowo, pozostałe znaki używane są jako sterujące.

Współpraca:

1. Program klienta, najczęściej o nazwie `telnet` sygnalizuje serwerowi chęć nawiązania połączenia. Odbywa się to poprzez wysłanie do serwera polecenia SYN na co ten odpowiada poprzez SYN_ACK co klient potwierdza znakiem ACK.
2. Po tym etapie następuje negocjacja opcji pomiędzy klientem a serwerem.
3. Serwer wysyła zapytanie o nazwę użytkownika i hasło. Gdy weryfikacja przebiegnie pomyślnie, w programie klienta wyświetlany jest znak zachęty i może on wprowadzać polecenia w postaci linii tekstu który przesyłany jest do serwera.
4. Serwer interpretuje polecenia i przesyła wyniki do programu klienta.



Rys. 1-9 Współpraca klienta i serwera protokołu telnet

Program `telnet` pracuje w trybie tekstowym - obsługiwane są specjalne sekwencje znaków, które umożliwiają pewien poziom formatowania tekstu i pozycjonowania kursora w oknie klienta telnet. Serwer i klient telnetu obsługują emulację różnych typów terminali. Najpopularniejsze to ANSI, VT-100, VT-52. Protokół telnet domyślnie używa portu 23.

Istnieje wiele programów będących klientem telnetu np. PuTTY oraz podstawowy program telnet zainstalowany w większości systemów.

Uruchomienie programu telnet:

```
$telnet [adres_IP [nr_portu]]
```

Aby skomunikować się z systemem wbudowanym za pomocą telnetu, po stronie tego systemu musi być zainstalowany i uruchomiony serwer telnetu (np. telnetd).

Zazwyczaj serwer telnetu uruchamiany jest przez superserwer sieciowy inetd lub xinetd.

Plik konfiguracyjny aplikacji inetd

Przy starcie `inetd` odczytuje swój plik konfiguracyjny (zwykle `/etc/inetd.conf`).

Każda linia pliku zawiera następujące informacje:

- Nazwa usługi
- Styl komunikacji gniazdka – `stream`, `dgram`, `raw`.
- Typ protokołu - `udp`, `tcp`.
- Opcje - `{wait|nowait}[/max-child[/max-connections-per-ip-per-minute[/max-child-per-ip]]]`
- Użytkownik, grupa która będzie przydzielona utworzonemu procesowi
- Nazwa programu wykonującego usługę
- Argumenty programu

```
#<service_name> <sock_type> <proto> <flags> <user> <server_path> <args>
#
#::INTERNAL: Internal services
#discard      stream      tcp      nowait      root      internal
#discard      dgram      udp      wait        root      internal
#daytime      stream      tcp      nowait      root      internal
#time         stream      tcp      nowait      root      internal

#::STANDARD: These are standard services.
ftp          stream tcp nowait root /usr/sbin/tcpd /usr/sbin/in.ftpd
telnet       stream tcp nowait root /usr/ucb/telnetd in.telnetd
tftp         dgram  udp  wait  bin /usr/ucb/tftpd  in.tftpd
```

Przykład 1-1 Plik konfiguracyjny `/etc/inetd.conf`

Przykład - Komunikacja z systemem QNX6 Neutrino

```
C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\juka>telnet 192.168.0.4_
```

Ekran 1-5 Uruchomienie programu telnet i próba połączenia z serwerem o adresie 192.168.0.4

```
C:\ Telnet 192.168.0.4
login: root
Fri Mar 21 12:11:29 2014 on /dev/tty0
Last login: Fri Mar 21 12:08:28 2014 on /dev/tty0
edit the file .profile if you want to change your environment.
To start the Photon windowing environment, type "ph".
# ps -ef | grep telnetd
  0      413719      172052  - Mar21 ?          00:00:00 in.telnetd
  0      462874      417816  - Mar21 ?          00:00:00 grep telnetd
# _
```

Ekran 1-6 Zgłoszenie się serwera telnetu - programu in.telnetd

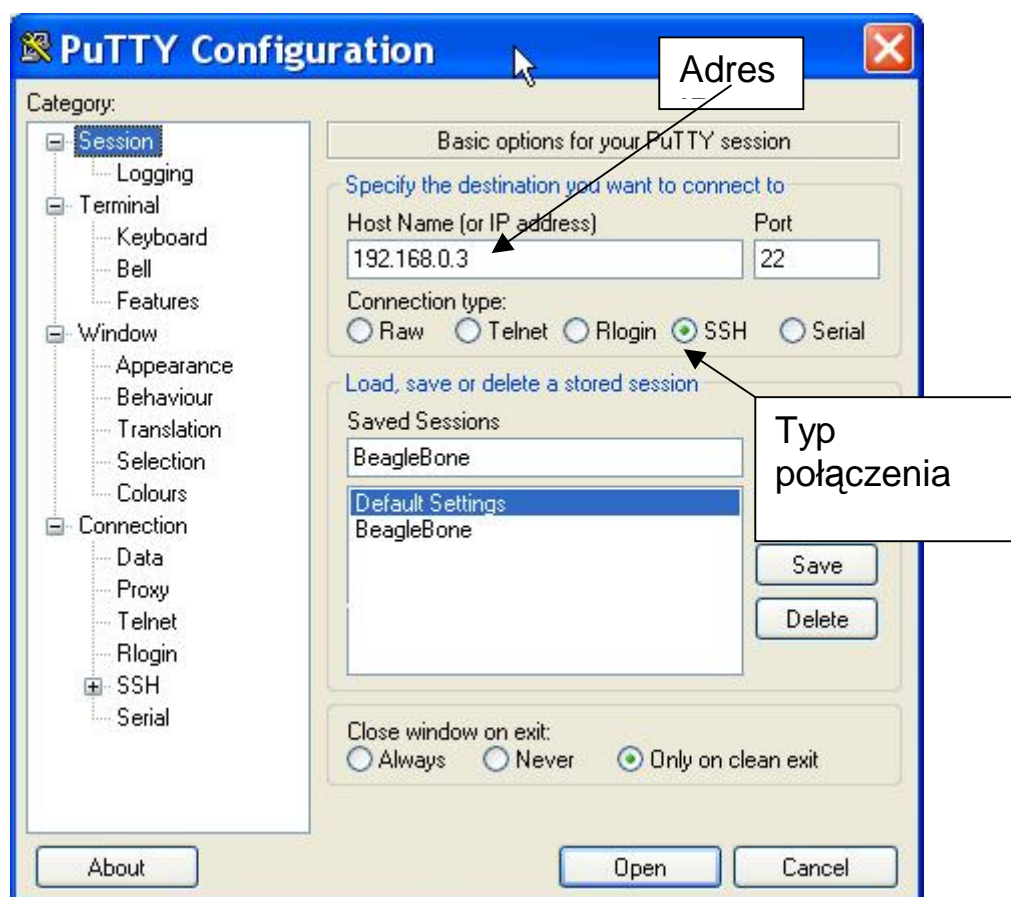
Usługa telnet jest szeroko stosowana do komunikacji z systemami wbudowanymi, jednak wykorzystuje ona połączenie nie zabezpieczone.

1.3.5 Usługa i protokół SSH

Po nazwą SSH (ang. *Secure Shell*) kryje się usługa będąca następcą telnetu a zapewniająca bezpieczne połączenie terminalowe ze zdalnym komputerem. W szerszym znaczeniu SSH to też nazwa rodziny szyfrowanych protokołów służących także do przesyłania plików, protokoły te to SCP (ang. *Secure Copy*) i SFTP (ang. *Secure File Transfer Protocol*).

Definicja protokołu SSH powstała w grupie roboczej IETF (ang. *Internet Engineering Task Force*) i opisano je w dokumentach RFC 4250, RFC 4251, RFC 4252, RFC 4253, RFC 4254. Protokoły SSH korzystają zazwyczaj z portu 22 połączenia TCP/IP. Transmisje SSH są szyfrowane według standardu AES choć stosowane są też standardy DES i Blowfish.

Najbardziej znanym klientem SSH jest program PuTTY.



Ekran 1-7 Konfiguracja połączenia SSH do komputera BeagleBone Black

```
login as: root
root@192.168.0.3's password:
root@beaglebone:~# uname -a
Linux beaglebone 3.8.13 #1 SMP Tue Jun 18 02:11:09 EDT 2013 armv7l GNU/Linux
root@beaglebone:~# cat /proc/version
Linux version 3.8.13 (koen@rrMBP) (gcc version 4.7.3 20130205 (prerelease) (Linaro GCC 4.7-2013.02-01) ) #1 SMP Tue Jun 18 02:11:09 EDT 2013
```

Ekran 1-8 Logowanie się do komputera BeagleBone Black za pomocą programu terminalowego

1.3.6 Przesyłanie plików, protokoły FTP, SCP, SFTP

Przy tworzeniu systemów wbudowanych często zachodzi potrzeba przesyłania plików pomiędzy komputerem macierzystym a systemem wbudowanym.

Najczęściej stosowane protokoły przesyłania plików to:

- protokół FTP (ang. *File Transfer Protocol*)
- protokół SCP
- protokół SFTP.

Protokół FTP

Protokół FTP jest standardowym protokołem przesyłania danych w sieci wykorzystującej TCP/IP a więc także w Internecie. Umożliwia on przesyłanie plików pomiędzy komputerami z których jeden jest serwerem (udostępnia on usługę) a pozostałe komputery, zwane klientami z tej usługi korzystają.

Pierwsza specyfikacja protokołu pojawiła się w RFC114 w roku 1971 a potem była rozwijana i dostosowana do protokołu TCP/IP co opisano w RFC765 i RFC959.

Standard FTP określa dokładnie w jaki sposób oprogramowanie na jednym komputerze współdziała z oprogramowaniem na drugim.

Popularny jest interfejs standardu BSD obejmujący ponad 50 poleceń, jednak w praktyce wykorzystuje się tylko kilka.

Standard FTP przewiduje obecność dwóch połączeń: połączenia sterującego (port 21) i połączenia przesyłania danych (port 20) co pokazuje Rys. 1-10. Połączenie sterujące utrzymywane jest przez cały czas sesji, połączenie dla danych może być stosownie do potrzeb tworzone i zamykane. Dzięki istnieniu dwóch oddzielnych kanałów polecenia i przesyłane dane mogą być łatwo separowane.

Początkiem pracy jest proces autoryzacji. Polega on na wysłaniu przez klienta do serwera polecenia **open**.

W odpowiedzi serwer przesyła żądanie podania nazwy użytkownika i hasła.

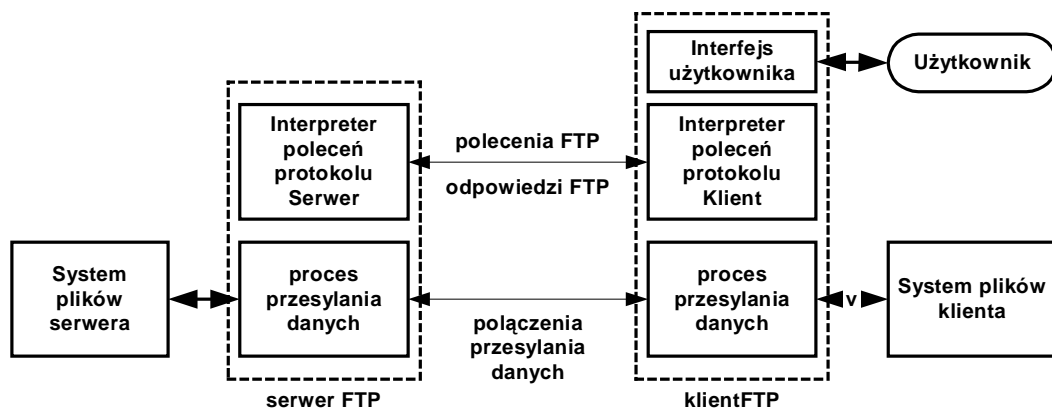
Klient przesyła nazwę użytkownika i hasło które podlega weryfikacji przez serwer.

Gdy przebiegnie ona pomyślnie, klient FTP ma takie prawa dostępu do plików serwera jakie miałyby lokalny użytkownik.

Po autoryzacji można pobierać pliki z serwera na komputer klienta wysyłając polecenie `get` lub `mget`.

Przesłanie pliku w drugą stronę następuje poprzez polecenia `put`, `mput` lub `send`.

FTP obsługuje hierarchiczny system plików, Można uzyskać zawartość katalogu serwera za pomocą poleceń `dir` a zmienić katalog bieżący za pomocą polecenia `pwd`. Połączenie zamykane jest poleceniem `close`.



Rys. 1-10 Schemat poglądowy aplikacji FTP – strona serwera i klienta

FTP umożliwia przesyłanie plików pomiędzy komputerami różnych typów o odmiennej reprezentacji danych.

Możliwe są dwa podstawowe tryby przesyłania: tekstowy i binarny. W trybie tekstowym przesyłane są teksty składające się ze znaków ASCII lub EBCDIC podzielone na wiersze.

Użytkownik może zlecić translację zestawu znaków z lokalnego na odległy lub odwrotnie. W trybie binarnym przesyłane są bajty bez możliwości translacji.

Istnieje wiele klientów protokołu ftp. Popularny jest program ftp który jest elementem większości systemów operacyjnych.

Polecenie	
help, ?	Uzyskanie pomocy
open	Uzyskanie połączenia z serwerem ftp
cd	Zmiana bieżącego katalogu na komputerze zdalnym
pwd	Wyświetlenie nazwy katalogu bieżącego na serwerze
lcd	Zmiana katalogu bieżącego komputera lokalnego
mget	Pobranie wielu plików z serwera do komputera lokalnego
put, send	Przesłanie pliku z komputera lokalnego do serwera
mput	Przesłanie wielu plików z komputera lokalnego do serwera
status	Podanie bieżącego stanu wszystkich opcji
by	Rozłączenie

Tabela 1-1 Podstawowe polecenia programu ftp

```

$ ftp 192.168.0.3
Connected to 192.168.0.3.
220 192.168.0.3 FTP server ready.
Name (192.168.0.3:root): juka
331 Password required for juka.
Password:
230-
    Welcome to QNX Neutrino!
230 User juka logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> dir
200 PORT command successful.
150 Opening ASCII mode data connection for '/bin/ls'.
total 0
226 Transfer complete.
ftp> _

```

Przykład 1-4 Komunikacja z systemem QNX za pomocą programu ftp

```

ftp> put pcm_cli
local: pcm_cli remote: pcm_cli
200 PORT command successful.
150 Opening BINARY mode data connection for 'pcm_cli'.
100% |*****| 5017 4.89 KB/s
226 Transfer complete.
5017 bytes sent in 00:00 (4.89 KB/s)
ftp> _

```

Przykład 1-5 Komunikacja z systemem QNX za pomocą programu ftp – wysłanie pliku

Protokół SFTP

Protokół SFTP (ang. *Secure File Transfer Protocol*) jest sieciowym protokołem służącym do zarządzania plikami i ich przesyłania.

Został opracowany jako rozszerzenie protokołu SSH w wersji 2.0. Zakłada on że transmisja przebiega przez bezpieczny kanał, dostarczany przez SSH.

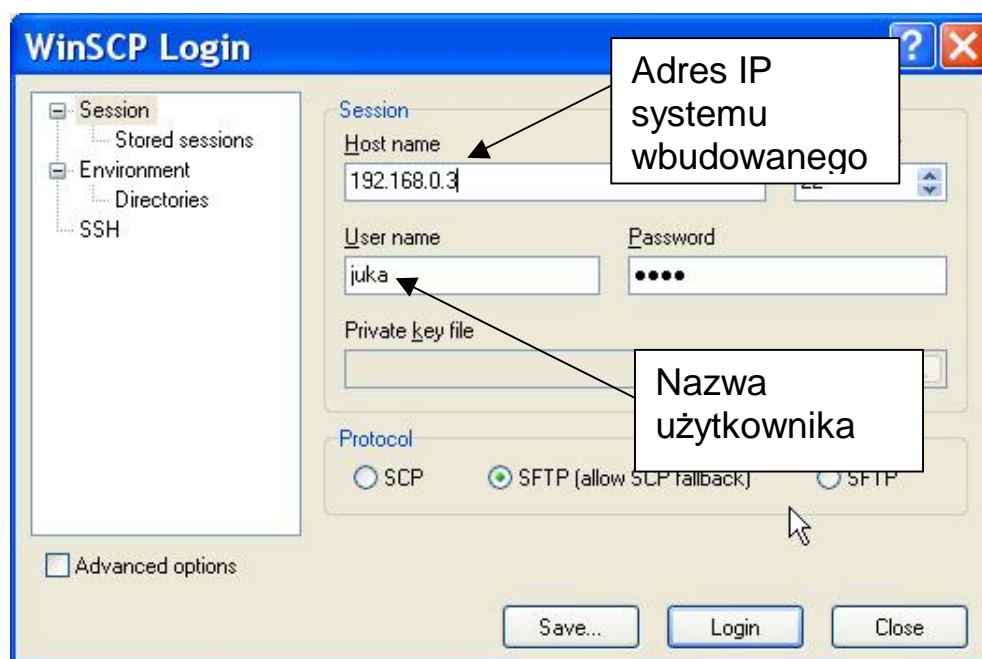
Oprócz przesyłania plików protokół umożliwia:

- wznowianie przerwanych transferów,
- listowanie zawartości zdalnych katalogów,
- usuwanie zdalnych plików
- inne.

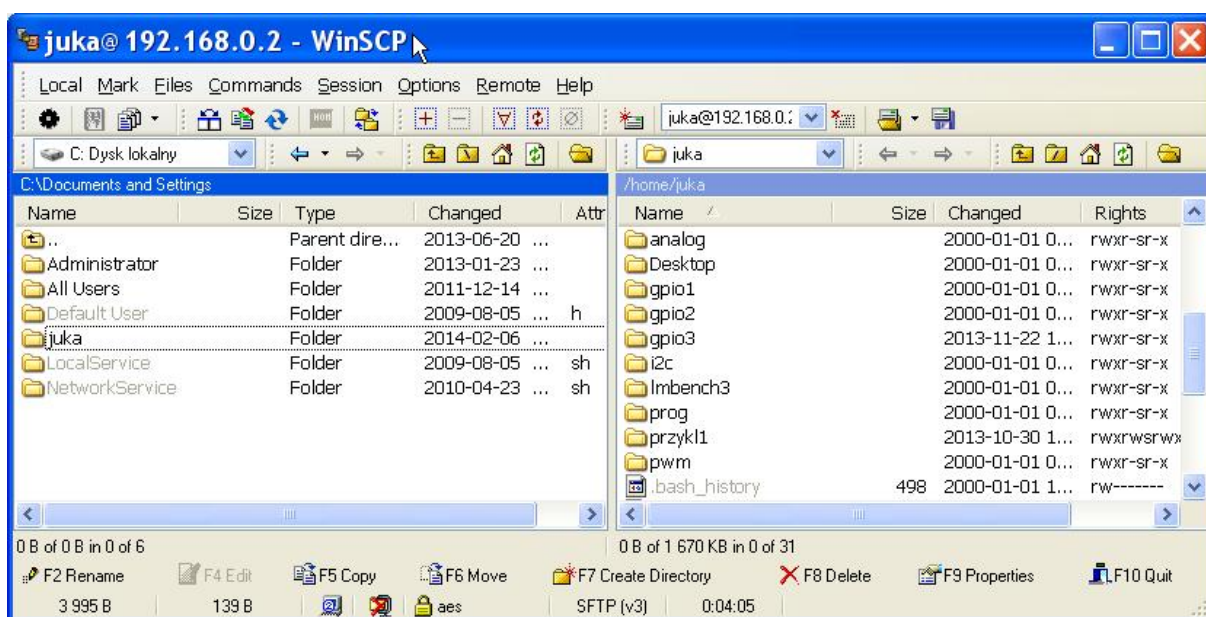
Kopiowanie plików zachowuje atrybuty plików jak np. czas ostatniego dostępu.

Aplikacja SFTP składa się z programu klienta i serwera. Protokół jako że jest częścią SSH używa portu 22.

Programy klienta mogą być w wersji konsolowej jak i korzystać z interfejsu graficznego. Bardziej znane implementacje klienta to WinSCP pracujący w systemie Windows i OpenSSH pracujący w Linuksie. Szeroko rozpowszechnioną wersją serwera SFTP jest OpenSSH pracujący w systemie Linux.



Ekran 1-9 Konfigurowanie połączenia z systemem wbudowanym BBB w programie WinSCP



Ekran 1-10 Widok programu WinSCP połączony z systemem wbudowanym

Protokół SCP

Podstawową wadą protokołu FTP jest brak możliwości szyfrowania nazwy użytkownika, hasła a także danych. Wady tej pozbawiony jest protokół SCP (ang. *Secure Copy Protocol*). Jest to sieciowy protokół, oparty na wcześniejszym rozwiązaniu BSD RCP (ang. *Remote Copy Protocol*) umożliwiającą przesyłanie plików między systemami.

Protokół SCP do przesyłania danych, szyfrowania i uwierzytelniania wykorzystuje protokół SSH. W oparciu o protokół SCP zrealizowano różnego rodzaju programy służące do kopiowania plików pomiędzy komputerami.

Podstawowa składnia polecenie `scp` występującego w systemie Linux w trybie konsolowym jest następująca:

```
scp [opcje][[użytkownik@]]host1:]plik1 [[użytkownik@]]host2:]plik2
```

Przykład:

```
$scp bhello.c juka@192.168.0.3:. 
```

```
juka@maria:~/prog/beagle$ scp bhello.c juka@192.168.0.3:.
The authenticity of host '192.168.0.3 (192.168.0.3)' can't be established.
RSA key fingerprint is 7f:25:1c:6e:fc:1d:93:d8:d6:38:a1:de:f9:7a:24:0a.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.0.3' (RSA) to the list of known hosts.
juka@192.168.0.3's password:
bhello.c                                     100% 135      0.1KB/s   00:00
```

Przykład 1-6 Kopiowanie pliku `bhello.c` z komputera macierzystego pracującego w systemie Linux do systemu wbudowanego BBB za pomocą programu `scp`

Porównanie protokołów przesyłania plików

	Protokół		
	SFTP	FTP	SCP
Bezpieczeństwo	Tak – zapewniane przez SSH	Nie, istnieje bezpieczna wersja: FTP nad TLS/SSL.	Tak
Szybkość	Najmniejsza ze względu na szyfrowanie	Duża	Średnia
Wznawianie przerwanych transferu	Tak	Tak	Nie
Przesyłanie dużych plików	Tak, plik może być większy niż 4 GB	Tak, plik może być większy niż 4 GB	Nie, plik musi być mniejszy niż 4 GB
Dopisywanie do końca pliku	Tak	Tak	Nie
Przerywanie transferu	Tak	Tak	Nie

Tabela 1-2 Porównanie protokołów SFTP, FTP, SCP

Z porównania wynika że protokół FTP może być stosowany w instalacjach lokalnych gdzie nie zachodzi niebezpieczeństwo przechwycenia hasła. Gdy pracujemy w systemie otwartym lepiej stosować protokoły SFTP i SCP.

1.4 IDE - zintegrowane środowiska uruchomieniowe

W używanych powszechnie systemach operacyjnych, do tworzenia oprogramowania stosuje się zintegrowane środowiska graficzne IDE (ang. *Integrated Development Enviroment*).

Zintegrowane środowisko uruchomieniowe jest aplikacją wykonywaną na komputerze macierzystym (najczęściej w trybie graficznym) ale komunikujące się z systemem wbudowanym.

Środowisko zintegrowane realizuje funkcje:

- edycji,
- kompilacji
- uruchamiania programów wykonywanych w systemie wbudowanym
- profilowanie
- analiza działania systemu operacyjnego (wywołania systemowe, procesy, obciążenie)

Jako interfejs komunikacyjny wykorzystuje się:

- RS232,
- Ethernet,
- USB
- JTAG.

Protokół komunikacji związany jest z interfejsem, najczęściej jest to TCP/IP.

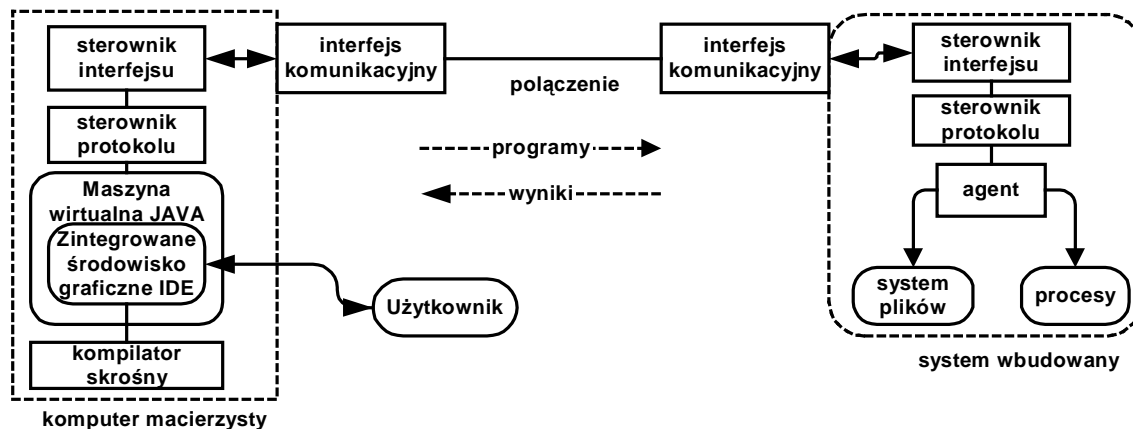
Zintegrowane środowiska uruchomieniowe IDE przeznaczone dla systemów wbudowanych często wykorzystują środowisko Eclipse. Środowisko jest w znacznym stopniu konfigurowalne przez zastosowanie tak zwanych wtyczek (ang. *plugins*). Istnieją wtyczki tak dla języka C jak dla systemów wbudowanych wykorzystujących różne procesory, a Eclipse może być wykonywane w systemach Windows i Linux.

Środowisko IDE zazwyczaj zawiera:

- edytor,
- system kompilacji skrośnej dostosowany do komputera docelowego,
- interfejs do programu uruchomieniowego,
- system przesyłania plików do komputera wbudowanego,
- rozbudowany system pomocy.

Po stronie systemu wbudowanego musi być wykonywany program agenta który pełni funkcje:

- komunikuje się z IDE i wykonuje jego polecenia
- odbiera przesyłane z komputera macierzystego programy,
- uruchamia je,
- przekazuje wyniki do IDE
- współpracuje z programem uruchomieniowym (np. gdb)



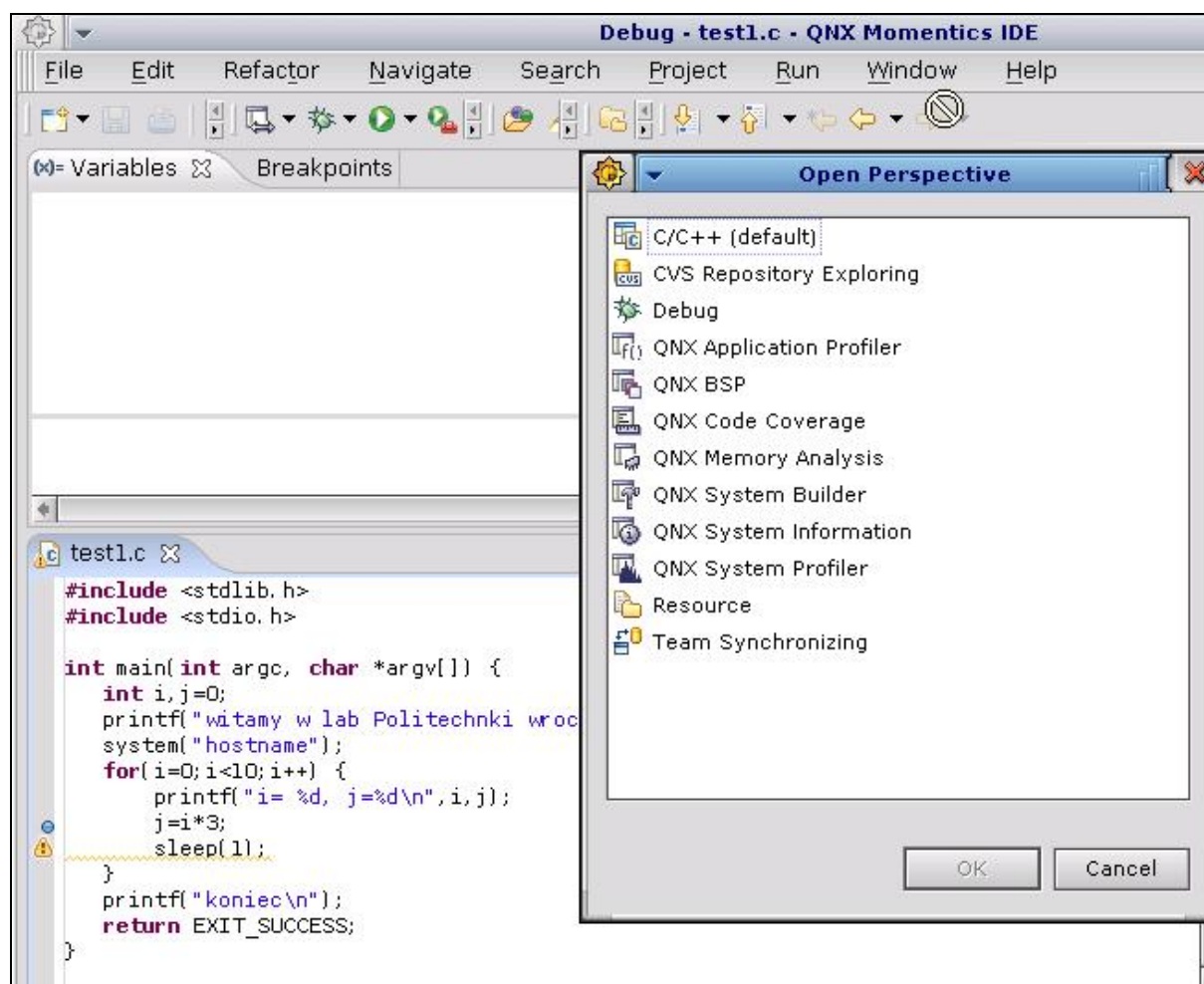
Rys. 1-11 Komunikacja z systemem wbudowanym za pomocą zintegrowanego środowiska graficznego

Zbudowane w oparciu o Eclipse IDE są nieraz bardzo rozbudowane i posiadają wiele możliwości.

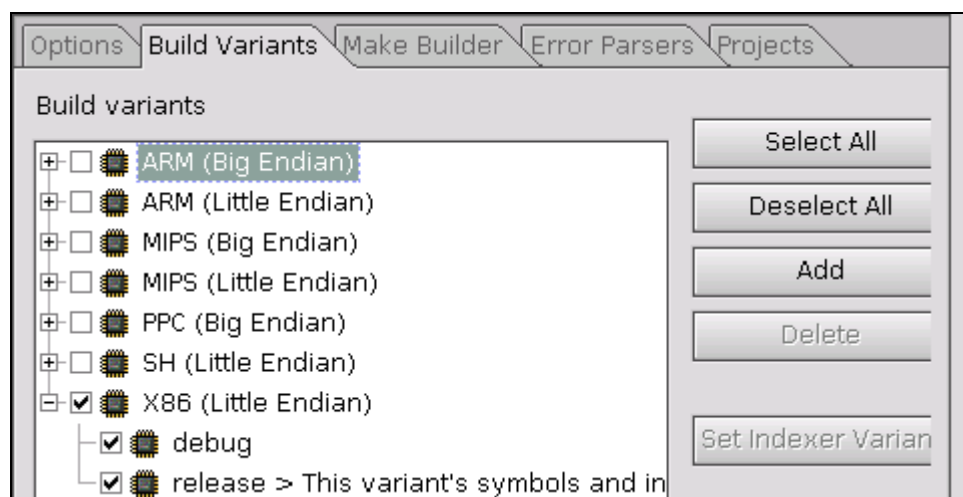
Przykłady:

- Momentics firmy QNX Software dla systemu QNX6
- Code Composer Studio firmy Texas Instruments QNX Momentics Tool Suite firmy QNX Software.

To ostatnie środowisko wykonywane może być w systemie Windows i Linux.



Ekran 1-11 QNX Momentics Development Suite wersja 4.1



Ekran 1-12 Wybór typu systemu docelowego