Raspberry Pi: bezpośrednie podłączenie do komputera

W poprzednich odcinkach serii zajmowaliśmy się konfiguracjami, w których Raspberry Pi (RPi) pracował w sieci domowej podłączony do routera i dalej – do Internetu. To router był odpowiedzialny za dostarczenie adresu IP i pośredniczenie w zdalnym logowaniu. W tym odcinku stworzymy uproszczoną konfigurację, gdzie RPi podłączymy bezpośrednio do komputera za pomocą kabla Ethernetowego lub karty bezprzewodowej WiFi.

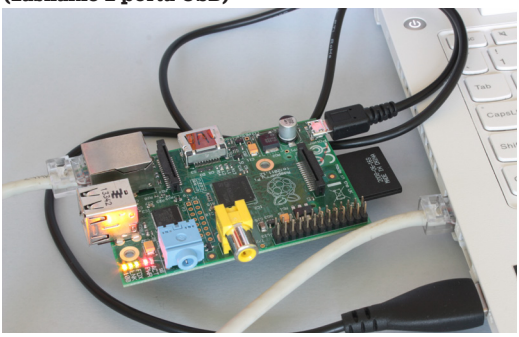
Jeżeli pracujecie z Raspberry Pi (RPi) poza siecią domową, istnieje możliwość podłączenia go bezpośrednio do komputera za pomocą kabla Ethernetowego. Prosta konfiguracja (nazywam ją „wyjazdową") umożliwi Wam dostanie się do „maliny" poprzez klienta SSH (np. Putty). Sposoby konfiguracji sieci domowej na potrzeby RPi opisałem w „Młodym Techniku" 8 i 9/2014. Tutaj zajmiemy się bardzo uproszczonym środowiskiem, kiedy nie ma routera. Kabel Ethernetowy podłączamy z jednej strony do RPi, a z drugiej bezpośrednio do karty sieciowej komputera. Następnie zajmiemy się stworzeniem podobnej konfiguracji, ale z użyciem łączności bezprzewodowej po WiFi.

Ethernet – na skróty

W największym możliwym skrócie:

- ustawiamy na RPi stały adres IP: kartę SD z RPi przekładamy do komputera, edytujemy znajdujący się na niej plik „cmdline.txt", dodając wpis „ip=169.254.1.1" (gdzie „169.254.1.1" jest nowym adresem IP);
- łączymy RPi i komputer kablem Ethernetowym;
- startujemy RPi;
- gdy uzyskamy połączenie (świecąca się dioda LNK na płytce RPi), na komputerze uruchamiamy Putty i otwieramy kanał SSH, podając zapamiętany adres „169.254.1.1".

1. Bezpośrednie podłączenie RPi do komputera (zasilanie z portu USB)



Powyższe instrukcje w zupełności wystarczą, żeby uzyskać bezpośrednie kablowe połączenie z RPi. Wiedząc jednak , jak dociekliwymi jesteście Czytelnikami, rozwinę temat w szerszym kontekście. Pomoże to Wam zrozumieć kilka zagadnień z dziedziny sieci komputerowych, które pojawiły się w poprzednich artykułach z serii „Młodego Technika" o Raspberry Pi – i znajdą się w następnych.

Ethernet? Internet?

Terminy „Ethernet" i „Internet" odnoszą się do technologii łączenia komputerów. Definiują właściwości fizyczne oraz logiczne sieci komputerowych, np. organizację okablowania, sposoby wymiany informacji. Główna różnica między nimi polega m.in. na zakresie. Określenia „Ethernet" używamy najczęściej w stosunku do sieci „lokalnych" LAN (ang. *Local Area Network*). Sieć lokalna to zazwyczaj zespół komputerów (urządzeń, np. drukarek sieciowych, kamer ochrony, czujników wejścia) znajdujących się w ograniczonym obszarze – np. tylko w Waszym domu. Z kolei Internet to sieć typu WAN (ang. *Wide Area Network*) łącząca urządzenia lub sieci LAN położone w pewnej odległości – np. znajdujące się w budynku za rogiem albo na drugim kontynencie. Można powiedzieć, że Internet to „sieć sieci". Istnieje niezliczona ilość sieci Ethernetowych, ale tylko jeden Internet.

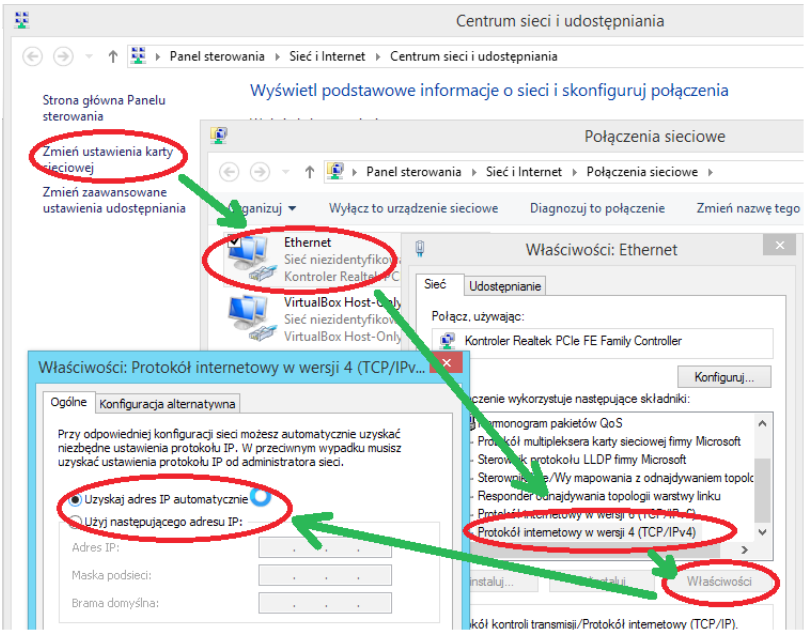
Adres IP

Czym właściwie jest adres IP (ang. *IP address*)? Jest to po prostu ciąg cyfr przyporządkowany danemu urządzeniu sieciowemu (np. karcie sieciowej). Ciąg ten umożliwia identyfikację konkretnego urządzenia (czy raczej jego „interfejsu sieciowego") oraz przesyłanych przez nie danych. Adres IP jest konieczny, żeby urządzenie mogło funkcjonować w sieci komputerowej. Dodatkowo, w obrębie jednej sieci wszystkie adresy IP muszą być unikalne. Są niepowtarzalne, żeby odróżnić, kto wysłał pakiety i do kogo są adresowane. Dokładniej, IP (ang. *Internet Protocol*) oznacza określony rodzaj protokołu komunikacyjnego, sposobu w jaki urządzenia „rozmawiają"

ze sobą. Musicie wiedzieć, że istnieją sieci IP w wersji 4 i 6. Tutaj ograniczymy się do IPv4. Pojedyncze adresy IPv4 zapisywane są najczęściej jako cztery liczby dziesiętne oddzielone kropkami, np. „169.254.1.1”. Każda z tych liczb jest 8-bitowa (oktet – cały adres IP jest 8x4 = 32-bitowy). To daje wartości z zakresu 0-255.

Kolejnym ważnym pojęciem jest „maska podsieci” (ang. *netmask*). Dzięki niej możemy określić przedział adresów IP. Dla przykładu: „169.254.0.0 z maską 255.255.0.0” oznacza zakres adresów od 169.254.0.0 do 169.254.255.255, czyli razem 256x256 adresów. Maska to także rodzaj filtra, który oddziela adres sieci (binarne „1” w masce) od adresów urządzeń w tej sieci (binarne „0” w masce). W podanym przykładzie część „169.254” jest adresem sieci, a dwie ostatnie liczby – adresami IP urządzeń znajdujących się w tej sieci. Można to w skrócie zapisać jako: 169.254.0.0/16, gdzie liczba „16” oznacza ilość „1” w masce (dziesiątkowe 255 w systemie binarnym zapisujemy jako 11111111).

Co najważniejsze dla tego opracowania, dwa urządzenia znajdujące się w tej samej podsieci mogą się komunikować bez pośrednictwa dodatkowych urządzeń sieciowych, np. routerów.



2. Właściwość „Uzyskaj adres IP automatycznie”

Dlaczego akurat 169.254.1.1?

Niektóre z puli adresów IP są traktowane specjalnie. Tak jest z zakresem „169.254.0.0/16” (czyli 169.254.0.0 z maską 255.255.0.0). Został on zarezerwowany przez IANA (ang. *Internet Assigned Numbers Authority* – organizacja zarządzająca przyznawaniem adresów IP) do komunikacji między urządzeniami połączonymi bezpośrednio. Maska 255.255.0.0 odnosi się do zakresu 169.254.[1-254].[0-255]. 169.254.255.255 zarezerwowano dla trybu rozgłoszeniowego (ang. *broadcast*), 169.254.0.x i 169.254.255.x zastrzeżono na przyszłe potrzeby. Adresy z tej puli są również przyznawane w ramach APIPA (ang. *Automatic Private IP Addressing*) – usługi auto-konfiguracji. Procedurę taką wywołuje się, gdy dany interfejs sieciowy nie może samodzielnie uzyskać adresu IP. Dzieje się tak, gdy adres statyczny nie jest skonfigurowany lub zewnętrzny serwer dostarczający adresy IP (DHCP) nie jest dostępny. Spójrzmy na domyślną konfigurację karty sieciowej pod Windows 8 (**ilustracja 2**):

- otwórz: Centrum sieci i udostępniania;
- wybierz: Zmień ustawienia karty sieciowej;
- wybierz interfejs sieciowy i z menu podręcznego: Właściwości;
- na liście usług znajdź: Protokół Internetowy w wersji 4 (TCP/IPv4); kliknij przycisk: Właściwości;
- w otwartym oknie Właściwości sprawdź, czy jest zaznaczone: Uzyskaj adres IP automatycznie.

W systemie Windows to właśnie ustawienie „Uzyskaj adres IP automatycznie” odpowiada za to, czy karta sieciowa oczekuje adresu IP z zewnątrz (jeżeli zaznaczony).

Jeżeli go nie uzyska, usługa APIPA automatycznie przydzieli jej adres z puli 169.254/16. Zdarzenie to można sprawdzić za pomocą polecenia „ipconfig/all” (część linii usunąłem, dla lepszej ilustracji):

```
C:\WINDOWS\system32>ipconfig /all
   Ethernet adapter Ethernet:
      Description . . . . . : Kontroler Realtek PCIe FE Family Controller
      DHCP Enabled. . . . . : Yes
      Autoconfiguration IPv4 Address. . : 169.254.198.218(Preferred)
      Subnet Mask . . . . . : 255.255.0.0
```

Widać stąd, że karta przyjęła adres „169.254.198.218”. Z usługi auto-konfiguracji urządzeń sieciowych pod Windows skorzystamy przy łączeniu RPi z komputerem. Wystarczy bowiem, że nasz Raspberry Pi także skonfigurujemy na adres z zakresu 169.254/16, a wtedy oba urządzenia znajdą się w jednej podsieci i będą mogły się porozumieć.

Statyczny adres IP

W przypadku Raspbiana (najpopularniejszej dystrybucji Linuksa dla RPi) auto-konfiguracja adresu IP nie włącza się domyślnie (trzeba doinstalować pakiet, np. avahi-autoipd). Łatwiej jest więc ustawić RPi stały adres IP. Można to osiągnąć na kilka sposobów (zob. [1]):

- (czasowa) zmiana adresu IP interfejsu sieciowego;

- zmiana pliku konfiguracyjnego /etc/network/interfaces;
- zmiana parametrów startowych kernela w pliku, /boot/cmdline.txt'.

Pierwsze dwa sposoby wymagają dostępu do samego RPi (np. przed wyjazdem lub z użyciem interfejsu UART). Adres IP RPi można zmienić czasowo, poleceniem:

```
sudo ifconfig eth0 169.254.1.1
```

Polecenie to musicie wydać **za każdym podłączeniem kabla sieciowego**. Jego przydatność do naszych celów jest więc ograniczona.

Jeżeli chodzi o plik konfiguracji sieciowej ‚/etc/network/interfaces':

- otwórzcie go za pomocą edytora tekstowego: sudo nano /etc/network/interfaces;
- odnajdźcie w pliku linię: „iface eth0 inet dhcp";
- wyłączcie ją, wstawiając znak ‚#' na jej początku, tzn.: „#iface eth0 inet dhcp (znak ‚#' sprawia, że linijka uznawana jest jako komentarz i będzie ignorowana);
- dopiszcie (w dowolnym miejscu) następujące trzy linijki:
  iface eth0 inet static
  address 169.254.1.1
  netmask 255.255.0.0
- zapiszcie plik, wciskając kolejno: CTRL-X, Y, ENTER;
- przeładujcie moduł sieciowy: *sudo service networking reload*;
- sprawdźcie, czy nowy adres IP został nadany: ifconfig eth0 | grep inet
  Powinniście zobaczyć:
  inet addr:169.254.1.1 Bcast:169.254.255.255 Mask:255.255.0.0;
- jeżeli interfejs dalej nie ma odpowiedniego adresu, zrestartujcie RPi komendą „sudo reboot".

Po dokonaniu powyższych zmian RPi będzie ustawiał wpisany przez Was adres za każdym włączeniem.

Ostatni ze sposobów jest chyba najprostszy. Nie wymaga nawet włączania RPi. Wyjmijcie kartę SD z RPi i włóżcie ją do komputera. Zignorujcie propozycję Windowsa w zakresie naprawiania jakoby uszkodzonej karty (karta wcale nie jest uszkodzona; Windows po prostu nie rozumie, co jest na niej zapisane). Znajdźcie na karcie plik cmdline.txt. Zawiera on parametry startu kernela linuksowego. Otwórzcie go w dowolnym edytorze (polecam darmowy Notepad++) i dopiszcie dodatkowy parametr:

ip=169.254.1.1

Zapamiętajcie: 169.254.1.1 będzie nowym, statycznym adresem IP Waszego RPi. Zmieniona zawartość pliku cmdline.txt może wyglądać tak:

dwc_otg.lpm_enable=0 ***ip=169.254.1.1*** *console=ttyAMA0,115200 kgdboc=ttyAMA0,115200 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline rootwait*

Zapiszcie plik, nie dodając żadnych „enterów" na końcu lub w środku (windowsowe znaki końca linii są inne niż linuksowe). Po dokonaniu zmian włóżcie kartę z powrotem do RPi. Przy następnym starcie nowy adres IP zostanie podany jako parametr startu kernela linuksowego.

Uwaga: przedstawione działania mogą sprawić, że Wasz RPi przestanie być widoczny dla sieci, w której się aktualnie znajduje. Najczęściej sieci domowe adresowane są „192.168.x.y". RPi skonfigurowane na „169.254.a.b" stanowić będzie całkiem odrębną sieć i router (bez dodatkowej konfiguracji) nie da rady zestawić do niego połączenia. Pamiętajcie o cofnięciu zmian, jeżeli chcecie RPi używać jak poprzednio.

eth0? ifconfig?

Występująca w powyższych listingach „eth0" to po prostu nazwa, jaką Linuks przyznaje pierwszej dostępnej w systemie karcie Ethernetowej. Jeżeli mielibyście więcej kart, Linuks oznaczyłby je jako „eth1", „eth2" itp. Podobnie numerowane są karty bezprzewodowe. „wlan0" oznacza pierwszą kartę bezprzewodową. Karty i ich właściwości można wyświetlać (a także zmieniać ich parametry) za pomocą używanego już wcześniej polecenia „ifconfig". Wydajcie takie polecenie RPi, a zobaczycie np.:

pi@raspberrypi:~$ ifconfig -a
eth0 *Link encap:Ethernet HWaddr b8:27:eb:55:8d:82*
inet ***addr:169.254.1.1*** *Bcast:169.254.255.255 Mask:255.255.0.0*
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:*160 errors:0 dropped:2 overruns:0 frame:0*
TX packets:*106 errors:0 dropped:0 overruns:0 carrier:0*
collisions:0 txqueuelen:1000
RX bytes:26258 (25.6 KiB) TX bytes:19988 (19.5 KiB)
lo *Link encap:Local Loopback*
inet ***addr:127.0.0.1*** *Mask:255.0.0.0*
UP LOOPBACK RUNNING MTU:65536 Metric:1
RX packets:*21 errors:0 dropped:0 overruns:0 frame:0*
TX packets:*21 errors:0 dropped:0 overruns:0 carrier:0*
collisions:0 txqueuelen:0
RX bytes:1220 (1.1 KiB) TX bytes:1220 (1.1 KiB)

Dzięki parametrowi ***-a*** sprawdzone zostaną również interfejsy nieaktywne. Widać, że w naszym systemie jest jedna karta Ethernetowa eth0. Jej obecny adres to „169.254.1.1". Z wydruku możecie odczytać również statystyki pakietów, czyli to, ile zostało ich wysłanych (ang. *TX packets*), odebranych (ang. *RX packets*) i inne. Jeszcze więcej o ruchu sieciowym możecie dowiedzieć się, używając narzędzia netstat.

Na pewno zauważyliście już, że oprócz eth0 pojawiło się lo z adresem „127.0.0.1". Jest to interfejs sieciowy, który oznacza... samą kartę. Adres „127.0.0.1"

to po prostu „ja sama”. Można się o tym łatwo przekonać, wydając polecenie śledzące przebieg pakietów traceroute:

pi@raspberrypi:~$ traceroute 127.0.0.1
traceroute to 127.0.0.1 (127.0.0.1), 30 hops max, 60 byte packets
1 **localhost (127.0.0.1)** 0.154 ms 0.088 ms 0.083 ms

Widzimy, że pakiety wysłane na adres „127.0.0.1” doszły do localhost – czyli samego RPi.

Ifconfig nie tylko dostarcza nam informacji o stanie interfejsów sieciowych. Pomaga także je kontrolować. W sekcji powyżej śledziliście, jak z jego pomocą ustawia się adres IP dla karty sieciowej. Interfejsy można włączać/wyłączać (ifconfig eth0 up lub ifconfig eth0 down). Polecam przeczytanie instrukcji obsługi (komenda „man ifconfig”).

Kabel połączeniowy

Dzisiejsze karty sieciowe są na tyle sprytne, że najczęściej nie wymagają używania specjalnych kabli „krosowanych” do podłączeń komputera z komputerem. Możecie użyć dowolnego kabla Ethernetowego zakończonego wtyczkami RJ45. Kabel taki można kupić w każdym markecie, lub wykonać go samemu. W drugim przypadku musicie zaopatrzyć się we wtyczki (ok. 20 gr/szt. – kupcie ich zawsze trochę więcej), w skrętkę UTP (ok. 1 zł/metr) oraz odpowiednią zaciskarkę do wtyczek RJ45 (jest konieczna, kosztuje ok. 20 zł, zob. **ilustracja 3**). Dzięki temu zestawowi będziecie mogli robić sobie kable o długości dokładnie dopasowanej do potrzeb. W sieci znajdziecie wiele samouczków, jak wykonać taki kabel. Dla porządku podaję ustawienie przewodów we wtyczce typu ‚A’ i ‚B’ (płaska strona do góry, styki od siebie; zob. [2]).

Wtyczka A:
- biało-zielony,
- zielony,
- biało-pomarańczowy,
- niebieski,
- biało-niebieski,
- pomarańczowy,
- biało-brązowy,
- brązowy.

Wtyczka B:
- biało-pomarańczowy,
- pomarańczowy,
- biało-zielony,
- niebieski,
- biało-niebieski,
- zielony,
- biało-brązowy,
- brązowy.

Kabel „prosty” (ang. *pathcord*; do połączenia np. komputer-router/switch) to wtyczki B i B lub A i A. Kabel krosowy to wtyczki A i B. Nie wszystkie żyły są wykorzystywane, ale lepiej zrobić pełną wtyczkę.

A co na komputerze?

Po „drugiej” stronie RPi znajduje się komputer stacjonarny lub (ze względu na przenośny charakter środowiska) laptop. Uruchomicie na nim klienta SSH. Ja najczęściej w tym celu używam darmowego programu Putty. Jest to uniwersalne narzędzie, które może działać zarówno jako klient SSH, jak i szeregowy (do obsługi podłączeń po UART, opisanych w jednym z poprzednich artykułów). Jego instalacja i konfiguracja nie sprawią Wam żadnych problemów. Putty można pobrać ze strony http://goo.gl/7DPObY.

Od strony konfiguracji sieciowej komputera wystarczy upewnić się, że aktywna jest opcja „Uzyskaj adres IP automatycznie” (zob. **ilustracja 2**).

Jeżeli Wasz komputer nie ma już wolnych portów sieciowych, możecie użyć zewnętrznej karty sieciowej podłączanej do portu USB (zob. **ilustracja 4**). Wtedy na liście „Wyświetlanie aktywnych sieci” pojawi się dodatkowa karta sieciowa.

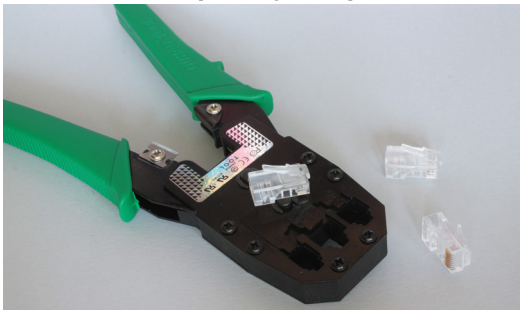
Zestawiamy przewodowe połączenie lokalne

Łączymy kablem Ethernetowym komputer oraz RPi, zasilamy RPi (można z portu USB komputera) i czekamy, aż załaduje się system. Obserwujemy diody stanu RPi. Dla modelu B (umieszczone przy krawędzi płytki, obok USB) będą to:
- LNK (zielona): świeci się, gdy Ethernet jest podłączony – miganie oznacza przesyłanie danych;
- FDX (zielona): tryb ang. *full duplex*, obustronna – jednoczesna wymiana danych;
- 100 (żółta): szybkość 100 Mb/s.

Dla modeli A+/B+ diody stanu połączenia sieciowego umieszczono na samym złączu Ethernetowym:
- żółta: jak 100 dla B (świeci się dla 100 Mb/s, zgaszona dla 10 Mb/s);
- zielona: jak LNK dla B.

Nawiązane połączenie i wymianę danych sygnalizuje mrugająca dioda LNK. Zapalona oznacza, że nasz kabel jest w porządku, a karty sieciowe RPi i komputera „dogadały się”. FDX oznacza tryb pełnego dupleksu, gdzie obie strony nadają i odbierają równocześnie. Gdy FDX

3. Zaciskarka do wtyczek RJ45 i wtyczki



4. Karta sieciowa na USB

się nie świeci, urządzenia pracują w trybie pół-dupleksu (ang. *half duplex*). Oznacza to, że najpierw nadaje jedna strona, a potem druga (nie nadają równocześnie). Świecąca się dioda 100 oznacza, że strony używają standardu 100BaseT o szybkości wymiany danych do 100 Mb/s (mega-bitów/s, ang. *fast ethernet*). Inaczej wymiana danych odbywa się w znacznie wolniejszym tempie 10 Mb/s. U mnie, korzystając z karty sieciowej na USB, uparcie świeciła się tylko dioda LNK. Oznaczało to, że karta pracowała w trybie pół-dupleksu i 10 Mb/s. Stan ten potwierdziły komunikaty systemu:

pi@raspberrypi:~$ dmesg | grep eth

[3.105307] smsc95xx 1-1.1:1.0 eth0: register 'smsc95xx' at usb-bcm2708_usb-1.1, smsc95xx USB 2.0 Ethernet, b8:27:eb:55:8:82

[25.136761] smsc95xx 1-1.1:1.0 eth0: hardware isn't capable of remote wakeup

[27.510532] ***smsc95xx 1-1.1:1.0 eth0: link up, 10Mbps, half-duplex****, lpa 0x0020*

smsc95 to układ kontrolera karty sieciowej. Niestety, dłuższa walka ze sterownikami do tej karty nie dała żadnych rezultatów. Za to po podłączeniu bezpośrednio do wbudowanego portu laptopa od razu zadziałał pełny duplex i 100BaseT (stan diod widać na **rysunku 1**). Sprawdziłem log:

dmesgpi@raspberrypi ~ $ dmesg|grep eth

[3.114510] smsc95xx 1-1.1:1.0 eth0: register 'smsc95xx' at usb-bcm2708_usb-1.1, smsc95xx USB 2.0 Ethernet, b8:27:eb:55:8d:82

[23.154783] smsc95xx 1-1.1:1.0 eth0: hardware isn't capable of remote wakeup

[25.297326] smsc95xx 1-1.1:1.0 ***eth0: link up, 100Mbps, full-duplex****, lpa 0xCDE1*

Jako ciekawostkę mogę Wam jeszcze zdradzić, że w RPi karta sieciowa jest podłączona do wewnętrznego huba USB. Możecie to sprawdzić, wydając komendę „lsusb –t".

Pozostaje połączenie się do terminala RPi po SSH:

- na komputerze uruchomcie Putty;
- wpiszcie adres ustawiony dla RPi – tu „169.254.1.1" i jako „connection type" zaznaczcie SSH;
- kliknijcie „Open", żeby nawiązać połączenie.

Jeżeli nie popełniliście żadnego błędu, powinniście po chwili dostać na terminalu znak zachęty:

login as:

Domyślny użytkownik to „pi", a hasło „raspberry".

Uwaga: najpierw połączcie urządzenia kablem Ethernetowym, dopiero później włączcie Raspberry. Inaczej start Linuksa może zostać opóźniony nawet o 2 minuty, dając systemowi szansę na poprawne skonfigurowanie interfejsu.

Adresy dynamiczne i DHCP

Adresy dynamiczne są dostarczane do urządzeń sieciowych przez serwer DHCP (ang. *Dynamic Host Configuration Protocol*). Zamiast polegać na procedurach automatycznych lub narzuconych adresach, moglibyśmy zainstalować DHCP na jednym z urządzeń. Musicie jednak pamiętać, że DHCP działa w trybie rozgłoszeniowym (ang. *broadcast*). Oznacza to, że pakiety z prośbą o zlokalizowanie serwera (DHCP DISCOVERY) rozsyłane są do wszystkich komputerów w danej sieci. Jeżeli więc zainstalujemy DHCP na komputerze, który później podłączymy np. do hotelowej sieci WiFi, istnieje ryzyko pojawienia się w jednej podsieci dwóch serwerów dystrybuujących adresy. Wasz komputer będzie próbował przydzielać je na wyścigi z serwerem hotelowym.

Zestawiamy bezprzewodowe połączenie lokalne (ad-hoc)

Umiecie już nawiązać bezpośrednie połączenie przewodowe z komputera do naszego RPi. Od razu nasuwa się pytanie o możliwość ustawienia **połączenia bezprzewodowego**. Taka konfiguracja może być przydatna dla np. RPi sterującego robotem mobilnym lub automatyką domową. Niestety, nasz RPi nie jest wyposażony we wbudowany interfejs WiFi. Można za to użyć zewnętrznej karty wpinanej do portu USB. W sprzedaży jest bardzo dużo takich modułów. Musicie jednak zwrócić uwagę na ich rodzaje, gdyż nie wszystkie działają bezproblemowo. Kupując firmowe adaptery, sprawdźcie najpierw stronę http://elinux.org/Rpi_VerifiedPeripherals. Jeżeli znajdziecie wybrany odbiornik na liście i ktoś potwierdził jego sprawność – macie dużą szansę, że i u Was zadziała bezproblemowo. Brak urządzenia na liście oznacza tylko tyle, że nikt go jeszcze do tej pory nie dopisał. Ponosicie wtedy jednak pewne ryzyko.

Problem tkwi najczęściej nie w firmie, której znaczek widnieje na karcie WiFi, ale w układzie, który kartę kontroluje. Podobnie jak Windows, Linuks również używa sterowników urządzeń. Żeby dana karta działała poprawnie, muszą być one dostępne. Spójrzmy na dystrybucję Raspbian (wrzesień 2014):

pi@raspberrypi:~$ ls /lib/modules/3.12.28+/kernel/drivers/net/wireless

W tym katalogu znajdziecie zestaw sterowników dostępnych dla Linuksa. Na liście zobaczycie np. bardzo popularne u nas układy firmy Realtek: rtl818x i rtl8192cu. Czy dany adapter naprawdę działa, często dowiecie się dopiero po włożeniu go do portu USB (RPi serii A/B mogą się same zrestartować) i wydaniu polecenia:

pi@raspberrypi:~$ lsusb

Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp.

Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub

Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp.

Bus 001 Device 004: ID 0bda:8176 Realtek Semiconductor Corp. RTL8188CUS 802.11n WLAN Adapter

Jeżeli komenda „lsusb" wyświetli Wasz adapter WiFi, to znaczy, że został on rozpoznany przez kernel i odpowiedni moduł się załadował. Możemy to sprawdzić poleceniem „lsmod" (większość linijek pominąłem, dla jasności):

pi@raspberrypi:~$ lsmod
Module Size Used by
…
8192cu 550797 0

Widzimy tu, że sterownik rtl-8192cu obsługuje również adaptery oparte o układ rtl8188cus. Skoro wygląda na to, że kernel poradził sobie z zainstalowaniem sprzętu, utworzymy połączenie ad-hoc do naszego Raspberry. Teraz skonfigurujemy ustawienia sieciowe RPi (zob. [3]):

pi@raspberrypi:~$ sudo nano /etc/network/

Dodajcie:

auto wlan0
iface wlan0 inet static
address 169.254.1.1
netmask 255.255.0.0
gateway 169.254.1.1
wireless-channel 1
wireless-essid RPiAM
wireless-mode ad-hoc

I zakomentujcie:

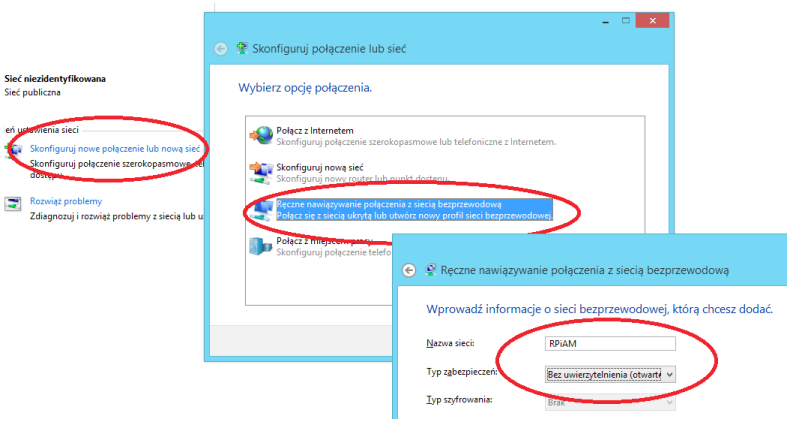
#wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
#iface default inet dhcp

Zwróćcie uwagę:

- wlan0 to interfejs karty sieciowej WiFi;
- nowa sieć ad-hoc nazywa się (SSID): RPiAM (możecie oczywiście podać własną nazwę);
- wlan0 skonfigurowano na adres „169.254.1.1”

Najwięcej problemów czeka Was jednak po stronie komputera z Windows 8.x. W tej wersji systemu usunięto bowiem z okienek możliwość zestawienia połączenia ad-hoc (dostępną np. w Windows 7). Żeby nawiązać połączenie ad-hoc pod Windows 8 (zob. [4]):

- otwórz: Centrum sieci i udostępniana;
- wybierz link: Skonfiguruj nowe połączenie lub nową sieć;
- w nowym okienku wybierz: Ręczne nawiązywanie połączenia z siecią bezprzewodową;
- w kolejnym okienku wpisz nazwę sieci (SSID) – u mnie RPiAM – i typ zabezpieczeń: Bez uwierzytelniania (otwarte).



5. Stworzenie profilu ad-hoc pod Windows 8

Powyższe działania utworzą profil ad-hoc. Żeby go jednak zobaczyć, trzeba uruchomić linię komend i wydać polecenie „netsh wlan show profiles” (niektóre linie pominąłem, dla jasności):

C:\WINDOWS\system32>netsh wlan show profiles
User profiles

All User Profile : RPiAM

W kolejnym kroku podłączymy profil:

C:\WINDOWS\system32>netsh wlan connect RPiAM

Connection request was completed successfully.

Ikona sieci bezprzewodowej powinna wskazać podłączenie, ale bez dostępu do Internetu (z czarnym wykrzyknikiem na tle żółtego trójkąta). Teraz wystarczy użyć Putty i połączyć się z RPi przez SSH do adresu „169.254.1.1”. Powinniście zobaczyć znany już znak zachęty.

Podsumowanie

Jak widzicie, bezpośrednie podłączenie Raspberry Pi do komputera wcale nie jest trudne. Jedynym wyzwaniem jest odpowiednie zsynchronizowanie adresów IP obydwu urządzeń. Z drugiej strony, za stosunkowo prostymi operacjami kryje się całkiem sporo „technikaliów”. Ich poznanie, nawet tak pobieżne jak w tym opracowaniu, pozwoli Wam na lepsze zrozumienie wielu procesów zachodzących w sieciach komputerowych i bardziej świadome rozwiązywanie problemów z nimi związanych. ■

Arkadiusz Merta

Tabela 1. Polecenia linuksowe związane z siecią dla RPi

Komenda	Opis	Przykładowe zastosowanie
ifconfig (Windows: ipconfig)	Sprawdź stan interfejsów sieciowych. Dodaj parametr -a, aby uwzględnić nieaktywne interfejsy	#Podręcznik użytkownika man ifconfig #Stan interfejsów ifconfig -a #Stan karty sieciowej ifconfig eth0 #konfiguracja adresu IP dla eth0 ifconfig eth0 169.254.1.1
dmesg | grep eth	Szczegóły inicjalizacji interfejsów eth	#Dla karty Wi-Fi dmesg | grep wlan
ethtool	Bardzo przydatne narzędzie do zarządzania siecią	#instalacja sudo apt-get install ethtool
lsusb	Lista urządzeń USB rozpoznanych przez Linuksa	
lsmod	Lista zainicjowanych modułów kernela	
netstat	Statystyki interfejsów sieciowych	
traceroute	Wyświetl drogę pakietów	traceroute mt.com.pl

Na podstawie: [1] http://goo.gl/t30PYR/, [2] http://goo.gl/B9e6rT, [3] http://goo.gl/LLIcOE, [4] http://goo.gl/UIEGek