

Ethernet coraz łatwiejszy

Czytelnicy EP przyzwyczaili się do szerokiej oferty mostów Ethernet <-> RS232 (lub ogólnie Ethernet <-> UART). Można by odnieść wrażenie, że niewiele już jest w stanie nas zaskoczyć. Jeśli myślimy o architekturze czy o sposobie działania tych komponentów, to rzeczywiście tak jest. Jedynym elementem, który pozostaje zmienny w czasie, jest forma oraz, jak się okazuje, dodatkowa funkcjonalność. W artykule opiszemy najnowszy produkt potentata rozwiązań interfejsowych: firmy MOXA.

Serwer portu szeregowego MiiNePort E1 jest miniaturowym mostem Ethernet <-> UART. Moduł jest zbudowany w oparciu o mikrokontroler z rdzeniem ARM. MCU ma 4 MB wewnętrznej pamięci SDRAM, 2 MB pamięci flash oraz, co nie jest często spotykane, wbudowany kontroler warstwy fizycznej (PHY). Architektura urządzenia przedstawiono na **rys. 1**.

Fizycznie moduł jest niewiele większy od standardowego gniazda RJ45 (**foto. 2**). Zewnętrzne wymiary to 33,9×16,2×13,5 mm, a rozkład wyprowadzeń odpowiada typowemu złączu RJ45 do druku. Poza wielkością należy zwrócić uwagę na niskie zapotrzebowanie mocy. Podobne urządzenia innych producentów potrzebują do prawidłowego działania prądu rzędu ~250 mA (przy 3,3 V), natomiast MiiNePort E1 pobiera 160 mA przy 3,3 V. Zatem różnica przekracza 56%. To sporo, jednak wciąż niewystarczająco, by

zastosować moduł w urządzeniu zasilanym z baterii.

Dla użytkownika dostępne są dwa porty komunikacyjne: UART oraz Ethernet. UART ma wszystkie sygnały kontroli przepływu wymagane np. przez standard RS232 (RTS/CTS, XON/XOFF). Ramka danych może składać się z 5, 6, 7 lub 8 bitów danych, może zawierać 1, 1,5 lub 2 bity stopu oraz bit parzystości. Prędkość transmisji może być ustawiana w zakresie 50...230 kbps, ale istnieje możliwość zamówienia wersji z prędkościami dochodzącymi do 921,6 kbps. Fizycznie dysponujemy ośmioma wyprowadzeniami (**rys. 3**), których funkcje zebrane są w **tab. 1**.

Moduł poza podstawową funkcjonalnością konwertera interfejsów oferuje również trzy proste w obsłudze linie cyfrowe. Jedną z nich to konfigurowalna linia wyjścia/wejścia, dwie pozostałe mają na stałe przypisane

funkcje wejścia lub wyjścia. Ponadto, interfejs szeregowy może współpracować nie tylko z transceiverem RS232, ale również RS485 (jedną z linii I/O może być skonfigurowana jako przełącznik RX/TX w RS485).

Moduł wyposażony jest w zintegrowany przycisk zerowania konfiguracji IP. Przycisk jest dostępny we wnętrzu gniazda RJ45. Za jego pomocą możemy wymusić restart urządzenia i przywrócić fabryczne ustawień. Funkcjonalność ta dostępna jest też poprzez odpowiednią konfigurację pinu 7.



Rys. 2. Moduł MiiNePort E1 (www.moxa.com)

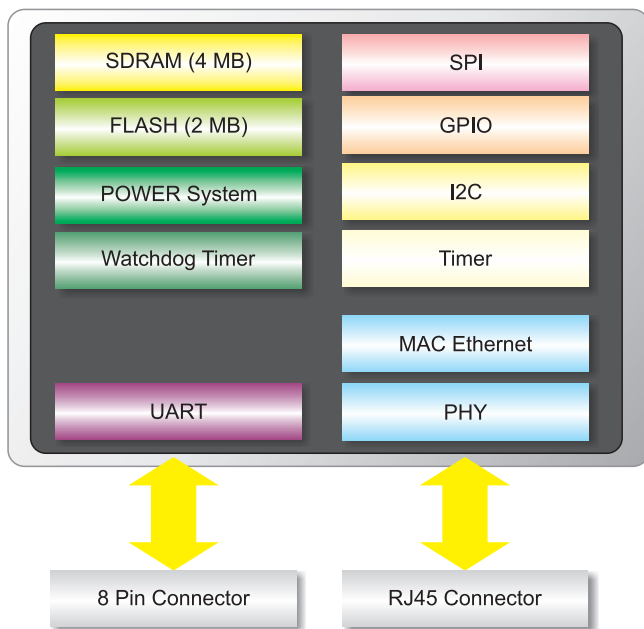
Interfejs Ethernet urządzenia może pracować z prędkościami 10 lub 100 Mbps oraz jest wyposażony w funkcję auto MDI/MDIX (rozpoznawania rodzaju kabla, którym moduł jest połączony z innym urządzeniem sieciowym). Moduł ma dwie diody LED sygnalizujące stan połączenia Ethernet.

Oprogramowanie

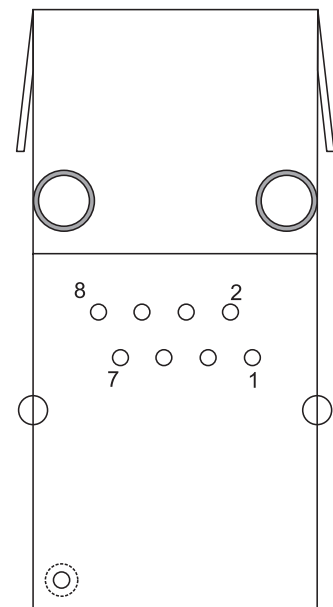
Aby rozpocząć pracę z urządzeniem, musimy najpierw dokonać jego konfiguracji. Obejmuje ona warstwę IP, proste zabezpieczenia dostępu, port szeregowy oraz linie wejścia-wyjścia. Ponadto, wybierany jest jeden spośród kilku trybów pracy modułu.

Przewidziane zostały cztery kanały, przez które możemy komunikować się z samym modułem oraz monitorować jego pracę.

Podstawowym narzędziem umożliwiającym obsługę urządzenia jest oprogramowa-



Rys. 1. Architektura modułu MiiNePort E1 (za www.moxa.com)



Rys. 3. Wyprowadzenia modułu (widok od spodu)

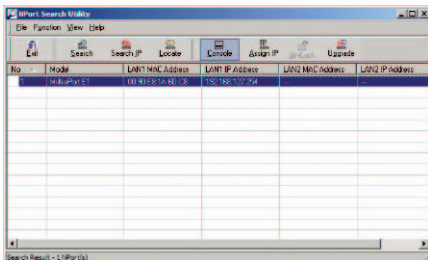
Tab. 1. Opis wyprowadzeń modułu MiiNePort E1

| Lp. | Symbol wyprowadzenia | Funkcja |
|-----|---------------------------------|--|
| 1. | GND | Zasilanie GND |
| 2. | VCC | Zasilanie +3,3 V |
| 3. | RST | Sygnal zerowania modułu |
| 4. | Tx | UART Tx |
| 5. | Rx | UART Rx |
| 6. | RTS/DO/485_en | Funkcje alternatywne: linia RTS, wyjście cyfrowe, wyjście przełączające nadawanie i odbiór dla transmisji RS485 |
| 7. | DTR/DIO/485_en/Reset to default | Funkcje alternatywne: linia DTR, wejście/wyjście cyfrowe, wyjście przełączające nadawanie i odbiór dla transmisji RS485/ Przywracanie fabrycznej konfiguracji (IP) |
| 8. | CTS/DI/DSR | Funkcje alternatywne: linia CTS, wejście cyfrowe, linia DSR |

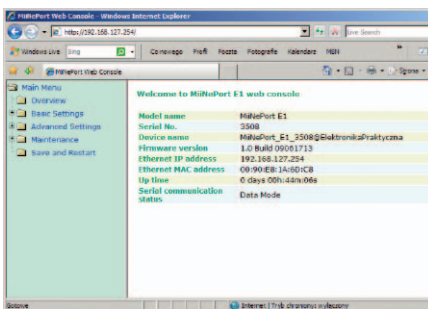
nie NPort Search Utility. Umożliwia ono wyszukiwanie modułów podłączonych do sieci w przypadku, gdy nie znamy ich adresów. Narzędzie to umożliwia również dokonywanie pełnej konfiguracji IP oraz, jeśli jest taka potrzeba, pozwala na przeprowadzenie aktualizacji oprogramowania urządzenia. Widok aplikacji przedstawiono na rys. 4.

Znając adres modułu, z którym chcemy się połączyć, możemy przeprowadzić konfigurację urządzenia z poziomu wbudowanego interfejsu WWW (rys. 5).

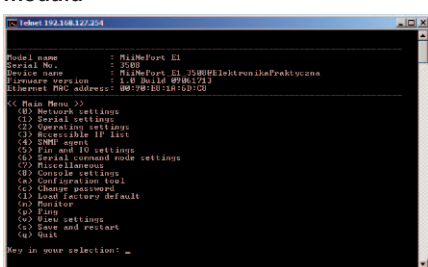
Dla bardziej konserwatywnych użytkowników (bądź systemów bez graficznego interfejsu użytkownika) przygotowano możliwość obsługi modułu z poziomu konsoli telnet (rys. 6).



Rys. 4. Aplikacja NPort Search Utility



Rys. 5. Główna strona interfejsu WWW modułu



Rys. 6 Interfejs telnet modułu

Powyższe kanały umożliwiają komunikację poprzez jego sieciowy. Możliwa jest też obsługa urządzenia przez interfejs szeregowy. Jest to przydatne wówczas, gdy np. istnieje potrzeba zmiany ustawień sieciowych z poziomu urządzenia, z którym współpracuje moduł (np. za pomocą lokalnej klawiatury). Producent przygotował specjalny protokół szeregowy, za pomocą którego możemy wymieniać ustawienia konfiguracyjne z urządzeniem. Przewidziano też kilka metod pozwalających na odróżnienie danych, które mogą napływać do urządzenia oraz komend konfiguracyjnych.

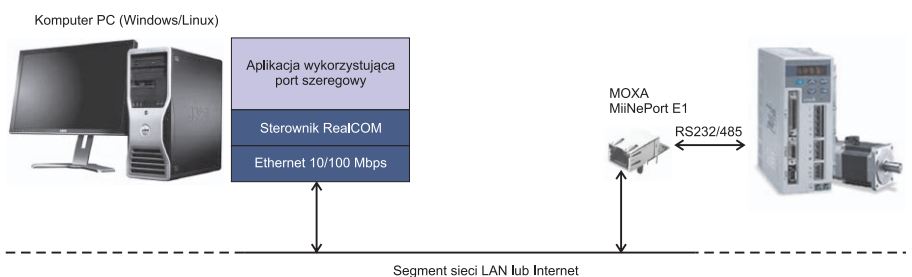
Tryby pracy

Jednym z podstawowych parametrów, które należy określić przed rozpoczęciem użytkowania modułu, jest tryb pracy. Określa on sposób wymiany i enkapsulacji danych wysyłanych i odbieranych przez interfejs szeregowy w interfejsie sieciowym. Użytkownik ma do wyboru 5 trybów pracy: Real COM, TCP, UDP, MCSC, RFC2217.

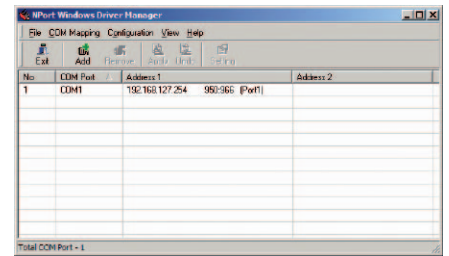
Real COM

Aby maksymalnie uprościć integrację modułu z urządzeniami i oprogramowaniem, które dotychczas używały łącza szeregowo do wymiany danych, moduł dostarczany jest ze sterownikami wirtualnego portu szeregowego. Sposób jego użycia przedstawiono na rys. 7. Sterowniki instalowane i zarządzane są (w przypadku systemu Windows) przez narzędzie NPort Windows Driver Manager (rys. 8)

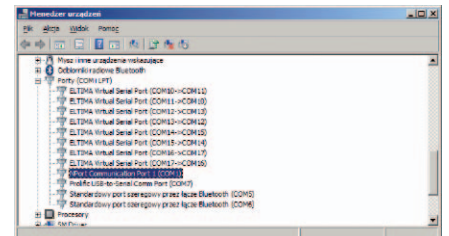
Oprogramowanie to wyszukuje dostępne moduły (podobnie jak NPort Search Utility), a następnie umożliwia zainstalowanie sterownika portu szeregowego, który skojarzony będzie z konkretnym modulem (rys. 9). Po za-



Rys. 7. Idea stosowania sterownika wirtualnego portu szeregowego RealCOM



Rys. 8. Aplikacja NPort Windows Driver Manager



Rys. 9. Wirtualny port szeregowy zainstalowany przez oprogramowanie NPort Windows Driver Utility

instalowaniu portu musimy pamiętać o przełączeniu modułu w tryb pracy RealCOM (np. z poziomu interfejsu WWW). Zaletą takiego rozwiązania jest oczywiście to, że nie musimy ponosić żadnych nakładów w postaci modyfikacji oprogramowania. Wadą natomiast jest fakt, że w całym łańcuchu przepływu danych dokładamy jeszcze jedno ogniwo w postaci sterownika, które może zawieść. W rezultacie obniża to niezawodność całego systemu. W przypadku wystąpienia problemu nie mamy też tak naprawdę możliwości jego diagnozy, nie mówiąc o naprawie. Rozwiązanie to, jakkolwiek wygodne i logiczne, nie powinno być, moim zdaniem, stosowane w nowo projektowanych systemach.

TCP/UDP

Oba tryby wymagają, by aplikacja sterująca urządzeniem szeregowym wykorzystywała bezpośrednio protokoły warstwy transportowej (TCP lub UDP) (rys. 10). W przypadku nowo tworzonej aplikacji jest to rozwiązanie bardzo wygodne. Nie musimy się martwić o całą otoczkę związaną z konfiguracją transmisji szeregowy. Wszelkie ustawienia transmisji muszą być dokonane bezpośrednio w module za pomocą np. interfejsu WWW. Bezpośrednie wykorzystanie protokołów wymaga jednak precyzyjniejszej definicji modelu komunikacyjnego.

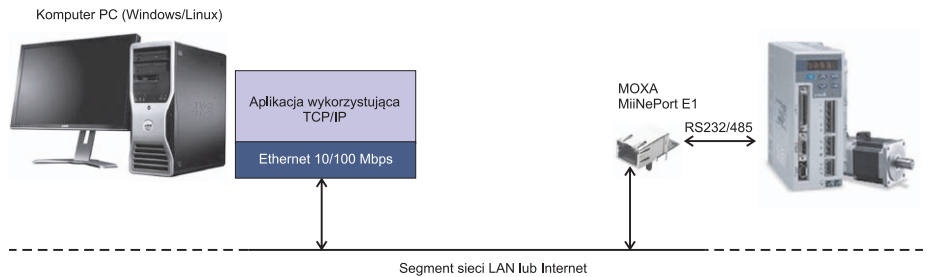
W przypadku UDP (protokół bezpołączeniowy) sytuacja jest uproszczona: Aplikacja chcąc nadać dane do urządzenia końcowego, po prostu przesyła je do socketu UDP skojarzonego z konkretnym adresem, pod którym nasłuchuje moduł. Ten z kolei przekazuje wszystkie odebrane pakiety UDP bezpośrednio do układu UART. W drugą stronę: jeśli urządzenie końcowe chce nadać wiadomość do komputera PC, wysyła swoje dane łączem szeregowym, a moduł przesyła je dalej, pod uprzednio zdefiniowany adres IP oraz port UDP. Oczywiście UDP nie gwarantuje nam poprawności transmisji, zatem efektywnie obniżamy wiarygodność kanału transmisyjnego.

Stosując pewny protokół, jakim jest TCP, musimy zadbać o zdefiniowanie węzła, który będzie nawiązywał połączenie przed faktyczną transmisją danych. Moduł MiiNePort E1 oferuje trzy opcje przy wyborze TCP jako protokołu komunikacyjnego: TCP Server, TCP Client, Mixed.

W trybie TCP Server moduł oczekuje na nadchodzące połączenia na uprzednio zdefiniowanym porcie TCP. Jeśli host (np. komputer PC) nawiąże połączenie, to transmisja danych może odbywać się w sposób nieograniczony w obie strony. Połączenie może zostać przerwane przez zdalnego hosta lub przez moduł po upływie określonego czasu bezczynności łącza (kiedy dane nie są wymieniane). Oznacza to, iż urządzenie końcowe nigdy nie może samodzielnie wysłać danych do nadzorującego komputera PC. Tryb ten jest idealny do zastosowań, w których urządzenie końcowe jest podrzędnym w stosunku do komputera PC. Innymi słowy: kiedy urządzenie ma prawo wysłać dane do PC będące odpowiedzią na otrzymane zapytanie lub potwierdzeniem otrzymania komendy/zapytania.

Tryb TCP Client odwraca rolę węzłów w systemie. Moduł oczekuje na dane nadchodzące do jego układu UART z urządzenia końcowego. Po otrzymaniu pierwszego bajtu nawiązuje on próbę ustanowienia połączenia TCP ze zdalnym hostem pod uprzednio zdefiniowanym adresem IP oraz portem. Znów, po ustanowieniu połączenia, oba węzły mogą dowolnie wymieniać dane między sobą. Tryb ten jest przydatny w pracy z wszelkiego rodzaju urządzeniami obsługiwany przez operatora: np. skaner kodów paskowych, gdzie w żaden sposób nie jesteśmy w stanie przewidzieć momentu, w którym urządzenie zacznie wysyłać dane do komputera.

W ostatnim trybie – Mixed – jak można się domyślić, do komunikacji stosowane są oba opisane powyżej modele. Oba węzły mogą nawiązywać połączenie i je zamykać. Model ten wykorzystywany jest przez omówiony wcześniej sterownik wirtualnego portu szeregowego RealCOM. Po konfiguracji modułu do pracy w tym trybie musimy podać zestaw parametrów serwerowych i klienckich, czyli port, na którym ma odbywać się nasłuch oraz



Rys. 10. Komunikacja przy bezpośrednim użyciu protokołów TCP/UDP

IP i port komputera, do którego mają napływać dane odebrane z łącza szeregowego.

MCSC

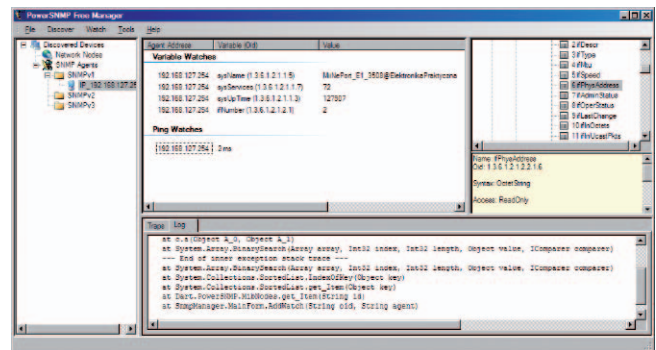
Odmianą trybu Mixed jest tryb MCSC (Multiple Channel Serial Communication), w którym jeden moduł MiiNePort E1 działa jako serwer i klient, ale jego klient i serwer niekoniecznie muszą pracować na tym samym fizycznym hoście. Dane nadsyłane przez urządzenie końcowe bez zapytania mogą trafiać do hosta opisanego parą IP/Port, natomiast dane, o które zapytał klient z (dowolnym IP), trafiają wyłącznie do niego. W bardziej wyszukanim scenariuszu, tryb ten pozwala na definiowanie dwóch kanałów komunikacyjnych oraz na niezależne definiowanie protokołu transportowego dla każdego z nich.

RFC2217

Tryb ten pozwala na pracę modułu kompatybilną ze standardem RFC2217. Został on zdefiniowany jako rozszerzenie standardowego protokołu *Telnet* (czyli połączenia TCP) o możliwość przesyłania informacji o parametrach transmisji szeregowej (prędkości bitowej, rodzaju ramki bitów stopu i kontroli przepływu). Pozwala to na nawiązywanie połączeń TCP z modulem przy jednoczesnej kontroli jego układu UART. Sterowniki RealCOM, dostarczone przez producenta, używają podobnego protokołu do konfiguracji portu szeregowego. W praktyce oznacza to, że moduł jest kompatybilny ze wszystkimi sterownikami portów szeregowych operujących w tym standardzie (a nie tylko z autorskimi sterownikami MOXA).

Ponadto...

MiiNePort E1 został wyposażony w funkcje predestynujące go do zastosowań bardziej profesjonalnych lub po prostu dla wymagających użytkowników. Pierwszą z nich jest *Auto IP Report Protocol*, który pozwala na skonfigurowanie modułu tak, aby co określony interwał czasu wysyłał on (pod wcześniej zdefiniowany adres i port) informację o swoim aktualnym adresie IP. Jest to funkcjonalność banalna, ale z drugiej strony, rzadko dostępna w urządze-



Rys. 11. Zrzut programu PowerSNMP Manager zarządzającego urządzeniami sieciowymi poprzez SNMP

niach tej klasy. Pozwala ona na jednoznaczna identyfikację urządzeń pracujących w sieci, w których konfiguracja IP jest przeprowadzana w sposób dynamiczny przez serwer DHCP. Drugą ciekawą funkcjonalnością jest wbudowany agent SNMP (*Simple Network Management Protocol*). Jest on zgodny ze standardem RFC1317 oraz RFC1213. Na rys. 11 przedstawiono podgląd niektórych zmiennych oferowanych przez agenta SNMP modułu w programie PowerSNMP Free Manager.

Podsumowanie

Na pierwszy rzut oka Moduł MiiNePort E1 można było zaszufłakować z innymi tego typu urządzeniami oferowanymi przez konkurencję. Jednak po głębszym zapoznaniu się z jego możliwościami, należy stwierdzić, że konstruktorzy podeszli do sprawy bardzo poważnie. Implementacja agenta SNMP, *Auto IP Report Protocol* oraz *RFC2217* to wyraźny znak dla użytkowników, mówiący o przeznaczeniu urządzenia do zastosowań w rozbudowanych infrastrukturach sieciowych. Moduł będzie też dobrym wyborem dla rozwiązań miniaturowych i energooszczędnych (jest to najmniejszy gabarytowo produkt tego typu). Jak zwykle są i minusy, a w zasadzie jeden, ale za to poważny: producent nie udostępnia SDK dla tego modułu, co jest powodem, dla którego jesteśmy skazani na funkcjonalność, którą sam przewidział. Ta na szczęście nie jest uboga i w parze z niewygórowaną ceną urządzenia (oczywiście w stosunku do konkurencji – około 177 zł netto) powoduje, że jest to ciekawa pozycja w tej klasie produktów.

Marcin Chruściel, EP
marcin.chrusciel@ep.com.pl