

Ethernet w jednym module, część 2

Taiko: Ethernet w Basicu

Ethernet do niedawna był kojarzony jako wysoce wyspecjalizowane medium transmisji danych przeznaczone wyłącznie do budowy sieci komputerowych typu LAN. Jego zalety dostrzeżone przez producentów urządzeń automatyki przemysłowej takie jak: tania warstwa fizyczna, długi odcinek kanału bez konieczności stosowania repetera (huba lub switcha) oraz duże prędkości transmisji spowodowały, że zaczęli go również stosować producenci przyrządów pomiarowych. Ethernet pojawia się więc coraz częściej w ich wyrobach obok sprawdzonego standardu GPIB.

Aby rozpocząć pracę z Taiko, potrzebujemy przede wszystkim platformy sprzętowej. Obecnie mamy do wyboru cztery moduły.

EM200 (fot. 2) to moduł typu OEM przeznaczony do montażu na płytce drukowanej. Nie posiada transformatora separującego, zatem musimy zastosować transformator zewnętrzny, bądź gniazdo RJ45 z wbudowanym transformatorem. Moduł posiada 9 linii I/O, z których cztery zostały użyte do obsługi kontroli przepływu portu szeregowego (w firmwarze rze pracują-

cym jako most). W TIDE mamy jednak do dyspozycji wszystkie 9 linii. Pracują one z poziomami CMOS i mają wydajność prądową 10 mA. Dodatkowo moduł posiada 4 linie wyjściowe służące do kontroli diod LED. Dwie z nich stanowią diody statusowe Ethernet (aktywność linku oraz prędkość połączenia), dwie pozostałe mogą być dowolnie wykorzystane przez użytkownika.

Kolejny moduł typu OEM to EM202 (fot. 3). Posiada on 4 linie I/O, wyprowadzenia UART oraz, po-



Fot. 2. Moduł EM200



Fot. 3. Moduł EM202

dobnie jak poprzednik, dwa wyprowadzenia wbudowanych diod LED. Jak widzimy na rys. 3, kompletny moduł mieści się w metalowej obudowie ze zintegrowanym gniazdem RJ45 oraz transformatorem.

Dwa pozostałe urządzenia proponowane przez Tibbo są oparte na modułach EM202 (EM202-EV oraz DS202 – fot. 4). Tak naprawdę są one identyczne, odróżnia je jedynie obudowa, której EM202-EV po prostu nie posiada. Moduły te wyposażono w przycisk, który został uwzględnio-

AMPER 2007

15-te międzynarodowe targi elektrotechniki i elektroniki

27. - 30. 3. 2007

Praga, Republika Czeska

maszyny, urządzenia, narzędzia i pomoce dla elektroniki i elektrotechniki - urządzenia do produkcji i rozprowadzania energii elektrycznej - technika zabezpieczająca i systemowa technika budowlana - technika automatyzacyjna, kierująca i regulująca - elektroniczne komponenty i moduły - napędy i elektronika mocy - technika pomiarowa i testująca - technika oświetlenia - technika elektrociepłna - technika elektroinstalacyjna - przewody i kable - usługi - telekomunikacja - usługi sieciowe - Technologie komunikacji radiowej - technologie komputerowe - technologie audio/video - maszyny, urządzenia, narzędzia i pomoce dla elektroniki i elektrotechniki

- jedyne targi elektrotechniki w RCz wspierane przez polski rząd

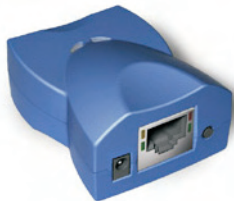
TERINVEST, tel.: +420 221 992 134, 126, e-mail: amper@terinvest.com, www.terinvest.com, www.amper.cz



a)



b)



Fot. 4 Moduły EM202-EV a) i DS202 b)

ny w Taiko jako obiekt, zatem nie jest on podłączony do jednej z kilku linii I/O.

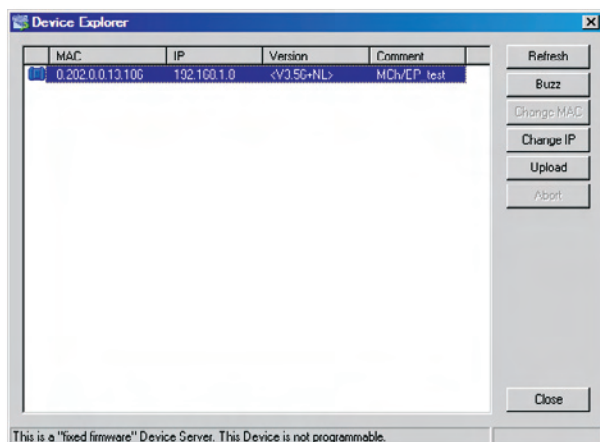
Wszystkie wymienione platformy sprzętowe zasilane są napięciem 5 V. Mają jednakowe zalecenia, co do obwodu zerującego (wszystkie posiadają linię RST). Dokumentacja mówi o za-

stosowaniu wyspecjalizowanego układu zerowania, np. MAX810. Rozmiar pamięci programu (Flash), jak i danych (RAM) jest identyczny dla wszystkich modułów i wynosi odpowiednio 65408 i 20480 bajtów. W tym miejscu zapewne wszyscy zastanawiamy się, jaki procesor został użyty jako baza modułu. Niestety taka informacja nie widnieje w dokumentacji Taiko. Jako ciekawostkę dodam, iż specjalnie w tym celu wysłałem zapytanie do supportu Tibbo. W odpowiedzi stwierdzono, iż typ zastosowanego kontrolera nie jest oficjalnie podany, ponieważ nie ma to żadnego praktycznego znaczenia. Można się oczywiście z tym zgodzić, jednak natura elektronika konstruktora rządzi się swoimi prawami, co pozostawia pewien niedosyt wiedzy na temat sprzętu, z jakim przychodzi nam pracować. W przypadku środowiska Taiko, jego twórcy zdecydowali się na zupełne abstrahowanie od sprzętu.

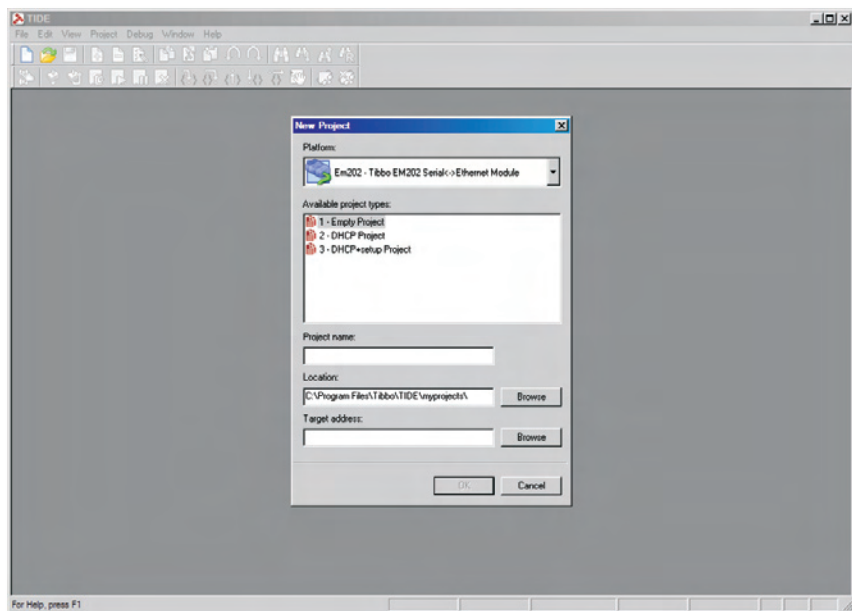
Wszelkie zasoby są zamodelowane obiektami, których zbiór ma się zmieniać wraz ze sprzętem (na chwilę obecną pozostaje on jednak niezmienny dla wszystkich platform).

Zaczynamy...

od pobrania ze strony http://www.tibbo.com/taiko_dl.php najnowszej wersji TIDE (w chwili pisania tego artykułu jest to: beta - 1.10.00) – plik *tide-1-10-00-be*



Rys. 5 Okno „Device Explorer”



Rys. 6 Okno nowego projektu

WG

Electronics

wg.com.pl

MAXIM

www.maxim-ic.com

DALLAS

www.maxim-ic.com

SILICON LABORATORIES

www.silabs.com

Lattice Semiconductor Corporation

www.latticesemi.com



cyan technology

www.cyantechology.com

POWER INTEGRATIONS

www.powerint.com

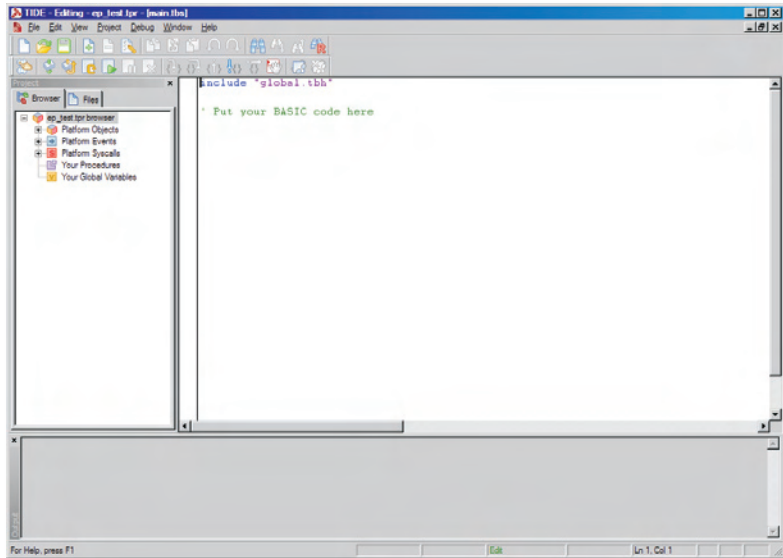
ELAN

www.elan-europe.com

KENT

www.kentdisplays.com

AUTORYZOWANY DYSTRYBUTOR



Rys. 7 Główne okno TIDE

ta.exe. Beta oznacza tu nic innego jak to, że Taiko jest wciąż rozwijane i nie mamy żadnej gwarancji, że będzie działało bezbłędnie. Jednak doświadczenie zdobyte przy pracy z wersją beta na pewno zaprocentuje w przyszłości. Następnie pobieramy TiOS – *em202-1_10_00.bin* – system operacyjny, który zainstalujemy w wybranym module. Po instalacji TIDE jest to pierwsza czynność, jaką musimy wykonać. W tym celu uruchamiamy „Device Explorer” (Znajdziemy go w zakładce *Tibbo* → *Tibbo IDE* menu *Start*). Na rys. 5 widzimy główny ekran aplikacji.

Tuż po starcie wyszukiwane są moduły podłączone do naszego segmentu sieci. Odbywa się to poprzez

rozsyłanie (*broadcast*) datagramów protokołu UDP, na które moduły odpowiadają podając takie informacje, jak np. MAC adres. Na rys. 5 przedstawiono zrzut ekranu aplikacji po wykryciu modułu EM202. Widzimy obecny adres MAC i IP modułu, jak również wersję zainstalowanego firmware'u. Jeśli jest to moduł prosto od producenta, będzie to np. 3.56+NL, czyli najnowsza wersja standardowego oprogramowania modułu (most RS232 – Ethernet). Po prawej stronie widzimy kilka przycisków: *Refresh* – odświeża listę znalezionych urządzeń, *Buzz* – użyteczna funkcja do identyfikacji modułu w przypadku, gdy do naszej sieci jest podłączonych kilka z nich. Po wybraniu z listy interesu-

jącego nas modułu klikamy *Buzz*, po czym wybrany moduł zasygnalizuje przy pomocy diod LED, (jeśli takowe zostały podłączone do jego wyprowadzeń) swoją obecność w sieci. Przyciski *Change MAC* i *IP* służą odpowiednio do nadawania nowych wartości adresów. Wreszcie opcja *Upload*, przy pomocy której załadujemy nowy firmware do naszego urządzenia (system TiOS). Po kliknięciu *Upload* mamy do wyboru metodę załadowania pliku. W chwili obecnej możemy to zrobić zarówno poprzez połączenie sieciowe, jak również szeregowo poprzez RS232. W poprzedniej wersji możliwe było ładowanie firmware'u jedynie przy pomocy portu szeregowego, co oczywiście wymagało podłączenia modułu do portu COM (w przypadku, gdy dysponujemy modulem OEM, nie jest to zawsze możliwe, nasze projektowane urządzenie nie musi wcale posiadać portu szeregowego). Zatem wybieramy „Load Firmware Through the Network”, a następnie nawigujemy do pliku *em202-1_10_00.bin*. Funkcja *Upload* zawiera jeszcze jedną opcję: „Load Application Through the Network”. Jest ona przeznaczona do umieszczania w pamięci modułu programu napisanego i skompilowanego w TIDE. W klasycznym przypadku nie będziemy jej używać, gdyż za każdym razem będziemy ładować nasz program z poziomu TIDE. Opcja może być jednak przydatna, kiedy planujemy wykonanie partii urządzeń, w której wszystkie muszą być zaprogramowane tym samym kodem. Nie

CONTRANS TI

oficjalny partner handlowy firmy

nowość!

oferujemy w szerokim asortymencie:

- karkasy w wersjach standardowych M, Ei, EE, UI oraz na zamówienie,
- karkasy dla rdzeni ferrytowych,
- karkasy z wtapieniami końcówkami,
- pudełka - obudowy transformatorów,
- akcesoria do kaskasów
- **nowość - obudowy na szynę DIN**

• **zapewniamy ciągłość dostaw**

• **atrakcyjne ceny**

Szczegółowych informacji technicznych i handlowych udzielamy telefonicznie i korespondencyjnie; na życzenie Klientów, przesyłamy dodatkowe materiały o oferowanych podzespołach i częściach elektronicznych.

CONTRANS TI Sp. z o.o.

ul. Polanowicka 66, 51-180 WROCLAW tel. 071/325-26-21...24, fax 071/ 325-44-39
e-mail: weisser@contrans.pl http:// www.contrans.pl

ma wówczas potrzeby uruchamiania środowiska TIDE. Jednak taka potrzeba pojawiła się właśnie teraz.

Na rys. 6 pokazano główne okno programu po wybraniu opcji stworzenia nowego projektu. Mamy do wyboru rodzaj platformy, jakiej chcemy użyć oraz 3 szablony projektu:

- **Empty Project** – zawiera dwa puste pliki: global.tbh (.tbh – tibbo basic header) oraz main.tbc (.tbc – tibbo basic code). Są to odpo-

wiednio pliki nagłówkowy i źródłowy programu.

- **DHCP Project** – zawiera dodatkowo plik dhcp.tbs dostarczający funkcji, przy pomocy których został zaimplementowany klient DHCP.
- **DHCP + setup Project** – od poprzednika odróżnia go to, że zawiera kod do inicjalizacji socket'ów i alokacji buforów nadawczych i odbiorczych dla TCP, jak

i dla portu szeregowego.

Wybieramy „DHCP + setup Project”. Nadajemy nazwę dla naszego projektu, katalog, w którym będzie przechowywany oraz fizyczną platformę docelową. Wyboru dokonujemy poprzez kliknięcie przycisku *Browse* po prawej stronie, co spowoduje ponowne uruchomienie „Device Explorer” – wybieramy moduł, którego chcemy użyć i zatwierdzamy przyciskiem OK.



MICROS sp.j.
Hurtownia podzespołów elektronicznych

Kraków, ul. Godlewskiego 38
tel. (012) 636 93 66
fax. (012) 636 93 99
e-mail: biuro@micros.com.pl
<http://www.micros.com.pl>

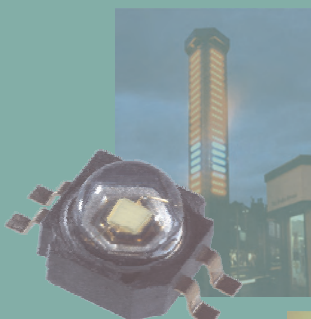
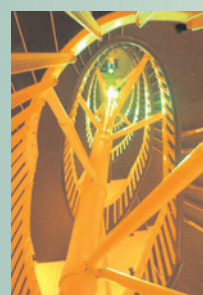
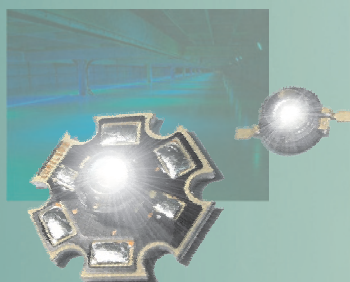
Szeroki wybór podzespołów elektronicznych. Prowadzimy obsługę sklepów, zakładów produkcyjnych oraz innych podmiotów gospodarczych.

Wyświetlacze LCD

- + szeroki wybór, konkurencyjne ceny!
- + wyświetlacze graficzne i tekstowe
- + różne kolory podświetlenia
- + dostępne wersje z cyrylicą i o rozszerzonym zakresie temperatur



LUMILEDS



Nowoczesne LED:
uniwersalne źródło światła

- Oferujemy diody LED:**
- do zastosowań specjalnych (medycyna)
 - emitujące błyski o dużej energii
 - świecące na biało
 - dużej mocy
 - ultrajazne (do 190 lm)
 - montowane w szerokiej gamie obudów (w tym na listwach)



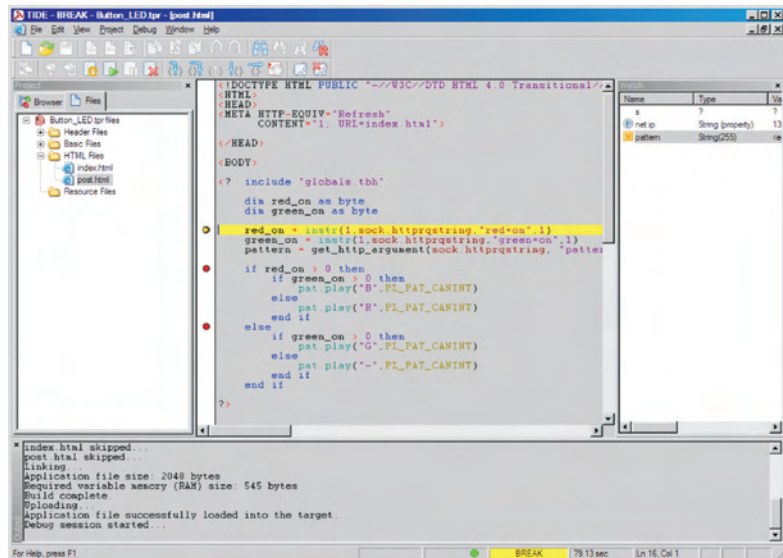
FUTURE ELECTRONICS POLSKA Sp. z o.o.
ul. Panieńska 9, 03-704 Warszawa,
tel.: (022) 6189202, fax: (022) 6188050

Making the Difference

Pełna informacja techniczna dostępna na naszych stronach:
www.futureelectronics.com
www.lumileds.com

Na rys. 7 widzimy gotowe do pracy okno TIDE. Oprócz standardowych pozycji w pasku menu, widnieją dwie najbardziej dla nas interesujące, czyli „Project” i „Debug”. „Project” zawiera takie pozycje jak: „Project Settings” – ustawienia projektu – te same, które wybieraliśmy przy tworzeniu nowego projektu. „Select/Manage Target” – opcja uruchamia „Device Explorer” i pozwala na wybór lub zmianę platformy sprzętowej. „Add, Remove, Rename File” – trzy pozycje pozwalające zarządzać plikami projektu. Kolejne dwie służą do budowy całego projektu do pliku wykonywalnego w wirtualnej maszynie działającej w module. „Build” – przebudowuje części (pliki źródłowe), które uległy zmianie od czasu ostatniej poprawnej kompilacji. „Rebuild All” przebudowuje bezwzględnie cały projekt. TIDE używa klasycznej metody budowy kodu wykonywalnego. Zawiera kompilator, który konwertuje kod źródłowy każdego z modułów (a więc plików) do kodu wykonywalnego *P-code* – zrozumiałego dla maszyny wirtualnej. Kompilator generuje tyle zbiorów wykonywalnych, ile nasz projekt zawierał modułów. Jest to tzw. kompilator pojedynczego przejścia (*single pass compiler*), co – jak sama nazwa wskazuje – oznacza, że każdy moduł jest analizowany przez program od początku do końca i ta analiza (i konwersja do *P-code*) odbywa się jednokrotnie dla każdego z modułów. To jednak nie wystarcza do uzyskania jednego wykonywalnego pliku, który możemy umieścić w pamięci modułu. Potrzebny jest Linker, który z elementów otrzymanych w wyniku kompilacji przeanalizuje zależności pomiędzy modułami i zadba o prawidłowe przyporządkowanie adresów pamięci tak, aby każdy z elementów modułu mógł odzyskać elementy programu potrzebne do jego działania (np. funkcja pracująca na zmiennej publicznej zdefiniowanej w innym pliku). Cała generacja kodu wykonywalnego odbywa się zatem w dwóch etapach. Na każdym z nich mogą wystąpić błędy oraz ostrzeżenia. Jeśli wystąpił błąd kompilatora, to linker nie zostanie uruchomiony. Kody błędów kompilatora rozpoczynają się literą „C”, natomiast linkera literą „L”

Pod powyższymi pozycjami widnieją ich odpowiedniki funkcjonalne powiększone o programowanie modułu („Build and Upload”, „Rebuild All and Upload”). Ostatnią opcją menu „Project” jest „Stop Build/Upload/De-



Rys. 8 Przykładowy projekt z ustawionymi breakpointami i rozpoczętą sesją debugera

bug” – przerywa odpowiednio kompilację, ładownie programu, sesję debugera.

Sesja debugera rozpoczyna się automatycznie po załadowaniu programu do pamięci modułu. Otrzymujemy wówczas kontrolę nad modułem z poziomu TIDE. Opcja automatycznego rozpoczynania sesji debugera może być zablokowana w ustawieniach projektu. Gdy to uczynimy, to po załadowaniu programu moduł po prostu wystartuje z nowym kodem. Czyli w przypadku zakończenia pracy nad projektem możemy tę opcję odznaczyć. Tak naprawdę nie decyduje ona tylko o tym, czy debugger wystartuje czy nie. Przede wszystkim informuje ona kompilator, czy kod wykonywalny ma zawierać funkcje do obsługi komunikacji z debugerem (co oczywiście dla użytkownika jest niewidoczne). W menu „Debug” widnieją standardowe dla tego typu narzędzi opcje. Możemy więc np. zatrzymać, startować i restartować program. Jednak prawdziwą wartość debugera niosą breakpointy – czyli punkty, w których program wykonywany zatrzyma się pod kontrolą TIDE, a także okno nazwane „Watch” – służące do podglądu zmiennych. Na rys. 8 widzimy prosty kod z ustawionymi breakpointami. W TIDE są one zaznaczone jako czerwone kropki po lewej stronie linii kodu. Jeśli debugger napotka na breakpoint, wówczas odpowiadająca mu linia zostanie podświetlona na żółto, natomiast czerwony punkt zostanie przykryty przez symbol tzw. wskaźnika programu (*program pointer*). W teorii (podczas pisania kodu

programu) breakpointy mogą być ustawione gdziekolwiek, jednak przy kompilacji i po starcie sesji debugera zostaną one przesunięte do najbliższej wykonywalnej części kodu. Oznacza to tyle, że breakpointy ustawione np. przy definicji zmiennej, (która jedynie alokuje pamięć na tę zmienną) zostaną przesunięte do pierwszej faktycznej operacji (np. przyporządkowanie tej zmiennej wartości innej zmiennej). Interesującą rzeczą jest to, że możemy dodawać breakpointy w trakcie wykonywania programu. W tym przypadku jednak debugger od razu umieści punkt na początku pierwszej wykonywalnej części kodu (musimy się do tego efektu przyzwyczaić, gdyż nie zawsze czerwona kropka pojawia się w miejscu klikania). Jednocześnie w projekcie możemy umieścić do 16 breakpointów w różnych jego miejscach (w różnych plikach – modułach), a ich rozmieszczenie zostanie zapamiętane w pliku projektu. W praktyce musimy pamiętać, że im więcej breakpointów umieścimy w programie, tym większe będzie spowolnienie działania maszyny wirtualnej. Wynika to z naturalnej przyczyny komunikacji TiOS z TIDE. Jest to po prostu dodatkowe zadanie dla maszyny wirtualnej. W kolejnej części zaprezentuję prosty przykładowy program oraz założenia naszego tajemniczego projektu.

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

Dodatkowe informacje

Autoryzowanym dystrybutorem produktów jest firma Soyter Sp. z o.o., www.soyter.pl, tel. 022 752 82 55