

Mikrokontrolery z rdzeniem ARM, część 7

Środowisko programistyczne Eclipse

Mikrokontrolery z rdzeniem ARM zazwyczaj wykorzystywane będą do budowy bardziej zaawansowanych projektów, dlatego do ich programowania będziemy używać najczęściej języków C lub C++. Na rynku istnieje wiele komercyjnych kompilatorów ze zintegrowanym środowiskiem IDE np. Keil ARM czy Rowley Cross Works Studio for ARM. Zawierają one rozbudowane edytory projektów, symulatory oraz wiele innych narzędzi ułatwiających pracę, jednak ich ceny sprawiają, że są niejednokrotnie poza zasięgiem małych firm, nie mówiąc już o warsztacie przeciętnego elektronika. Możemy wykorzystać wersję ewaluacyjną komercyjnych narzędzi, ale wówczas musimy liczyć się z ograniczeniami generowanego kodu wynikowego (Keil ARM – do 16 KB), lub ograniczeniami czasowymi (Rowley Cross Works – 30 dni). Dużo lepszym rozwiązaniem jest wykorzystanie oprogramowania open-source, które jest pozbawione wszelkich ograniczeń, a swoją funkcjonalnością niejednokrotnie nie będzie odbiegać od rozwiązań komercyjnych.

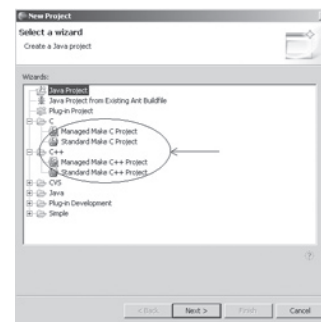
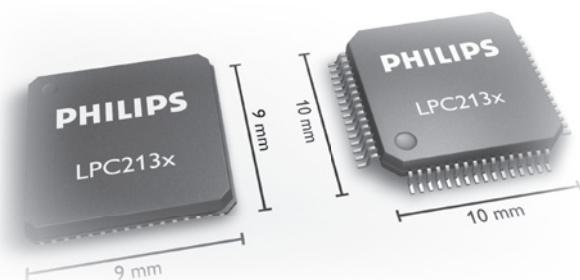
W bieżącym odcinku zajmemy się przygotowaniem wygodnego środowiska służącego do programowania mikrokontrolerów ARM, zapoznamy się z plikami startowymi występującymi w każdym projekcie, a za miesiąc uruchomimy pierwszy przykładowy projekt.

Do projektowania aplikacji dla ARM-ów będziemy wykorzystywać zintegrowane środowisko programistyczne (IDE) Eclipse. Eclipse pierwotnie zostało zaprojektowane do pisania aplikacji w języku Java, jednak po zainstalowaniu dodatkowego pluginu CDT umożliwia również pisanie oprogramowania w języku C/C++. Do kompilacji programów posłużymy się doskonałym kompilatorem języka C/C++ *gnuarm* (gcc), natomiast do programowania mikrokontrolerów LPC213x/214x posłużymy nam program *LPC2000 Flash Utility*. Kompilator oraz programator dają się doskonale zintegrować ze środowiskiem Eclipse, tak więc po poświęceniu odrobiny czasu na zainstalowanie i skonfigurowanie wszystkich narzędzi staniemy się

posiadaczami doskonałego środowiska dla mikrokontrolerów ARM, które nie posiada żadnych ograniczeń.

Instalacja oprogramowania

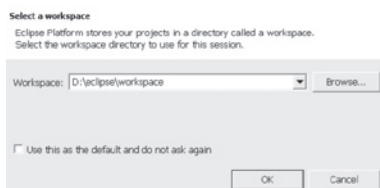
Przed rozpoczęciem instalacji środowiska Eclipse musimy upewnić się, że na komputerze została zainstalowana maszyna wirtualna Javy (*JRE – Java Runtime Environment*), która jest niezbędna do jego działania. Możemy to sprawdzić poprzez panel sterowania w zakładce *Dodaj/Usuń Programy*. W przypadku braku maszyny Java musimy pobrać jej najnowszą wersję ze strony: <http://www.java.com/en/download/>, a następnie zainstalować ją w systemie. Kolejną czynnością jest pobranie środowiska Eclipse ze strony: <http://www.eclipse.org/downloads/>. Należy wybrać najnowszą wersję (co najmniej 3.1.1) przeznaczoną dla środowiska Windows. Z uwagi na dużą objętość pliku (ponad 100 MB) ściąganie może zająć trochę czasu. Niezbędna nam również będzie nakładka CDT, która umożliwia tworzenie projektów w języku C/C++. Nakładkę CDT w wersji 3.0.1 pobieramy ze strony: <http://www.eclipse.org/cdt/>. Instalację rozpoczynamy od rozpakowania pliku Eclipse, (który występuje w postaci archiwum zip) w dowolnie wybrane miejsce np. *d:\programy*, a następnie rozpakowanie nakładki CDT w to samo miejsce (np. *d:\programy*). Po



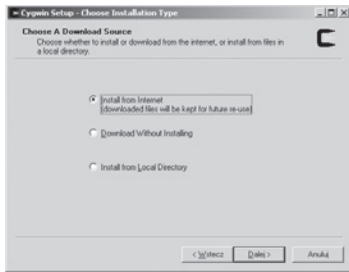
Rys. 15.

rozpakowaniu obu plików tworzymy skrót na pulpicie lub/i w menu start do pliku *eclipse.exe*, który znajduje się w podkatalogu *eclipse*. Następnie sprawdzamy poprawność działania Eclipse uruchamiając go. Przy pierwszym uruchomieniu pojawi się okno dialogowe przedstawione na rys. 14. W oknie tym wpisujemy domyślny katalog, gdzie przechowywane będą pliki projektów. Jeżeli chcemy aby pytanie to nie było ponawiane przy każdym uruchomieniu, zaznaczamy opcję *Use this as the default and do not ask again*, a następnie zatwierdzamy przyciskiem OK. Po pojawieniu się okna głównego aplikacji z menu wybieramy opcję *File->New->Project*. Na ekranie pojawi się okno dialogowe przedstawione na rys. 15.

W oknie powinniśmy zobaczyć opcje umożliwiające tworzenie projektów w języku C oraz C++. Gdy są



Rys. 14.

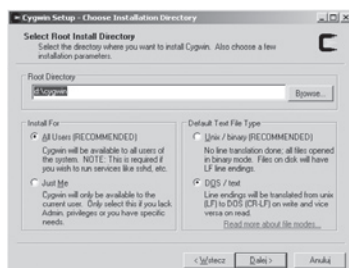


Rys. 16.

one niedostępne świadczy to o złej instalacji *pluginu* CDT, musimy wówczas upewnić się czy ściągnęliśmy dobrą wersję, oraz czy *plugin* został rozpakowany do katalogu, gdzie znajduje się *Eclipse*. Kolejną czynność, jakiej musimy dokonać to zmiana perspektywy projektu z Java na C++. Można to zrobić wybierając z menu polecenie *Window->Open Perspective->Other* i z dostępnych opcji wybierać perspektywę C/C++. Po stwierdzeniu poprawności działania programu, możemy zamknąć *Eclipse* i przystąpić do dalszych czynności instalacyjnych. Do prawidłowej pracy kompilatora *gnu-arm* oraz nakładki CDT niezbędna jest instalacja środowiska *Cygwin*. Jest to zestaw bibliotek DLL emulujących funkcje API środowiska LINUX w systemie Windows oraz zestaw progra-

Uwaga!
Komplet programów wymienionych
w artykule publikujemy na CD-EP6/2006B.

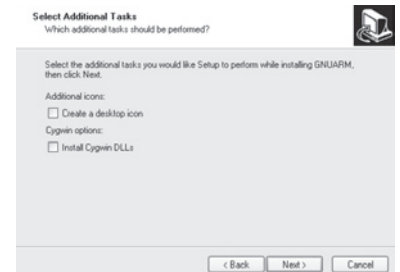
mów narzędziowych, które są zawarte w Linuxie. Z tych narzędzi interesować nas będzie *make* służące do automatyzacji procesu kompilacji projektów oraz kompilator *gcc* dla systemu Windows. Kompilator dla Windows będzie wykorzystywany przez nakładkę CDT do wstępnego kompilowania plików źródłowych na potrzeby systemu inteligentnych podpowiedzi, oraz eksploratora projektu. Instalacje *Cygwina* rozpoczynamy od ściągnięcia i uruchomienia pliku *setup.exe* ze strony: www.cygwin.com (link *install or update now*) Po uruchomieniu zobaczymy dialog powitalny instalatora, klikamy przycisk *Dalej>*, wówczas



Rys. 17.

pokaże się dialog wyboru źródła instalacji przedstawiony na rys. 16. Po zatwierdzeniu tej opcji wyświetli się dialog ustawień konfiguracyjnych środowiska przedstawiony na rys. 17.

W polu tekstowym *Root Directory* wpisujemy katalog, w którym będzie zainstalowane środowisko *Cygwin*, natomiast pozostałe opcje ustawiamy tak jak na rys. 18. Po zatwierdzeniu ustawień pojawi się ekran z oknem tekstowym służącym do wpisania ścieżki gdzie będą zapisane pliki pobrane przez instalator. Kolejną czynnością konfiguracyjną będzie wybór sposobu połączenia z Internetem. W większości przypadków wystarczy wybrać opcję *Direct connection*. Po wybraniu tej opcji pojawi się pytanie o wybór serwera, z którego należy pobrać dane. Tutaj z listy najlepiej jest wybrać serwer <ftp://sunsite.icm.edu.pl>. Kolejnym ekranem będzie wybór pakietów do zainstalowania. W tym wypadku musimy zmienić opcję instalacji pakietu *Develop* z *Default* na *Install* poprzez kliknięcie na małe kółko znajdujące się koło napisu *Develop*. Pozostałe pakiety zostawiamy na domyślnym poziomie instalacji. Po zatwierdzeniu rozpocznie się procedura pobierania plików i instalacja, co przy modemie DSL może zająć kilkadziesiąt minut. Gdy instalacja dobiegnie końca zostaniemy zapytani czy chcemy stworzyć skrót do konsoli *Cygwina* możemy tutaj odpowiedzieć *No*. Po zakończeniu instalacji, musimy jeszcze upewnić się, że katalog *cygwin\bin* został dodany do zmiennej systemowej *PATH*. Można to sprawdzić na przykład poprzez uruchomienie wiersza polecenia i wpisanie *gcc*. Jeżeli nie pojawi się odpowiedź *gcc: no input files* wówczas do zmiennej systemowej *PATH* musimy wpisać ścieżkę do katalogu *cygwin\bin*. Po upewnieniu się, że środowisko *Cygwin* zostało prawidłowo zainstalowane przystępujemy do instalacji kompilatora *gnuarm*. Można go pobrać ze strony www.gnuarm.org, na której w zakładce *files* znajduje się binarna wersja kompilatora w wersji 4.0 dla środowiska *cygwin* (link: http://www.gnuarm.org/bu-2.16.1_gcc-4.0.1-c-c++_nl-1.13.0_gi-6.1.exe) Po uruchomieniu pliku instalatora pojawia się ekran powitalny, z którego przechodzimy klawiszem *Dalej>* do okna z warunkami licencji, które akceptujemy wybierając *I accept the agreement*. Następnie przechodzimy do ekranu z wyborem katalogu, w którym zostanie zainstalowany kompila-



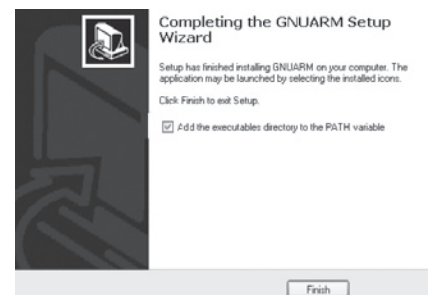
Rys. 18.

tor. Wybieramy dowolne miejsce na dysku, a następnie zatwierdzamy nasz wybór klawiszem *Dalej>*. Po tej czynności pojawi się kolejne okno z opcjami wyboru instalacji wybieramy *Full Installation*. Następnie przechodzimy przez ono tworzenia skrótów w menu Start. Po zatwierdzeniu pojawi się dodatkowe okno wyboru opcji przedstawione na rys. 18.

Musimy tutaj koniecznie odznaczyć opcję *Install Cygwin DLLs*, ponieważ w przypadku wybrania tej opcji może nastąpić konflikt pomiędzy bibliotekami DLL zainstalowanymi przez środowisko *Cygwin*, a bibliotekami dostarczonymi wraz z kompilatorem. Następnie przechodzimy do kolejnego dialogu, na którym wciskamy przycisk *install* rozpoczynając proces instalacji. Na zakończenie instalacji pokaże się okno przedstawione na rys. 19.

Musimy pamiętać o zaznaczeniu opcji dodania ścieżki kompilatora do zmiennej systemowej *PATH*. Ostatnią czynnością, jaką musimy zrobić to zainstalowanie programu „*LPC2000 Flash Utility*” służącego do programowania pamięci mikrokontrolera *LPC213x/214x* poprzez port szeregowy. Oprogramowanie to pobieramy ze strony producenta: <http://www.semiconductors.philips.com/pip/LPC2132FB-D64.htm> oraz instalujemy w sposób standardowy w dowolnym miejscu na dysku. Po tej czynności mamy już w pełni skompletowane środowisko programistyczne, możemy, więc zająć się tworzeniem pierwszego projektu

Lucjan Bryndza, EP
lucjan.bryndza@ep.com.pl



Rys. 19.