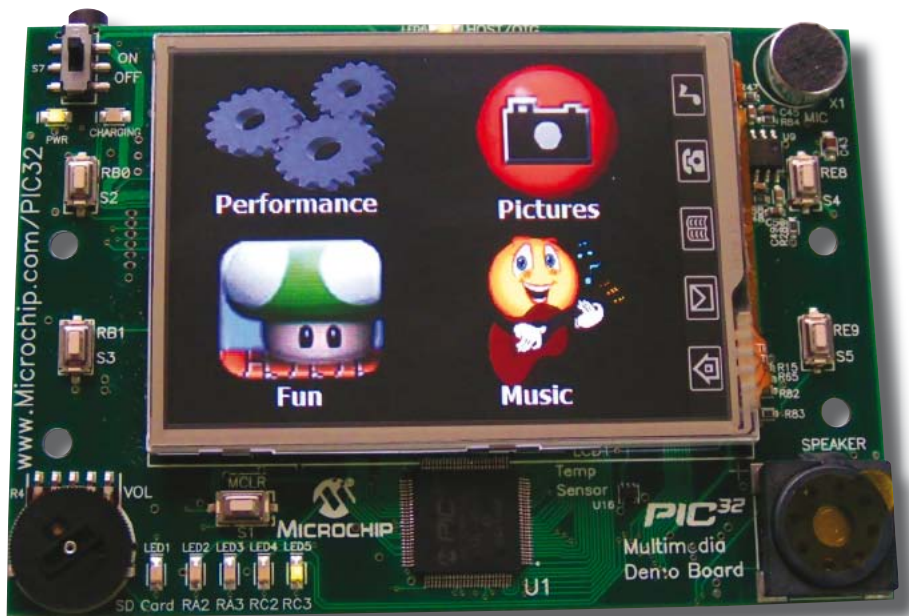


W mnogości siła?

Moduł starter kit PIC32

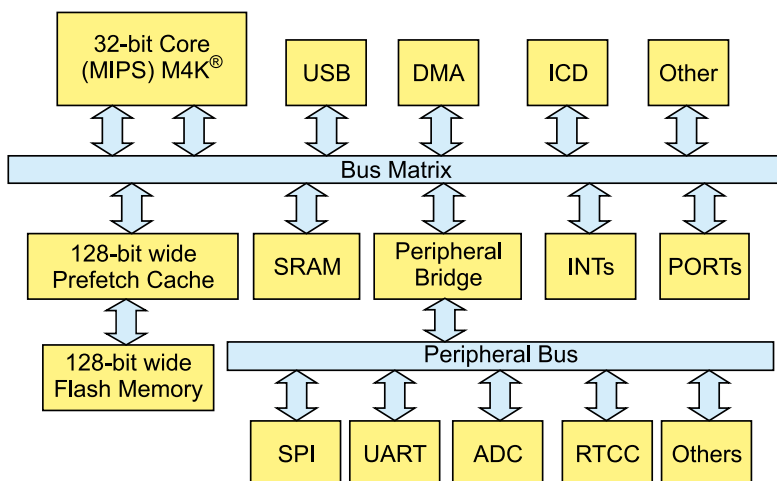
Przepis na sukces wydawał się prosty: kupujemy dobry rdzeń, obudowujemy swoimi peryferiami i dodajemy własną technologię produkcji. Pierwszy na dużą skalę postąpił tak NXP, wypuszczając niewiarygodnie tanie 32-bitowe mikrokontrolery serii LPC21xx z rdzeniem ARM7. Wydawało się, że instalowanie tego samego rdzenia w układach różnych producentów stanie się światowym standardem, jednak na tym monolicie pojawiły się rysy: Atmel zapowiedział, że będzie promował własny 32-bitowy rdzeń, a Microchip zdecydował się na zastosowanie rdzenia innego niż ARM.

Większość znanych dzisiaj firm produkujących mikroprocesory zaczynała od konstrukcji 8-bitowych. Wyjątkiem jest Intel, którego pierwszy mikroprocesor był 4-bitowy. Od początku rozwoju tych elementów każdy z nich był inny. Konstrukcje Intela, Motoroli, Ziloga, Microchipsa i innych różniły się między sobą wszystkim: typem architektury, rdzeniem, peryferiami. Różnice były tak duże, że kiedy konstruktor zdecydował się na jednego producenta, to zmiana na mikroprocesor lub mikrokontroler innego wiązała się nie tylko ze zmianami przyzwyczajęń, ale też ze sporymi wydatkami na nowe narzędzia projektowe.



Duże zapotrzebowanie na mikrokontrolery i silna konkurencja na rynku spowodowały duży spadek ich cen, ale też i cen narzędzi projektowych. Jeżeli dodamy do tego upowszechnienie się kompilatorów języka C, to migracja pomiędzy różnymi typami różnych producentów stała się łatwiejsza niż kiedykolwiek. Program napisany w języku C można było przenieść na inny mikrokontroler bez większych problemów, podczas gdy przeniesienie programu napisanego w assemblerze wiązało się w praktyce z napisaniem go od nowa. Jednak stare przyzwyczajenia konstruktorów dawały o sobie znać w postaci zażartych dyskusji o wyższości producenta A nad producentem I lub M.

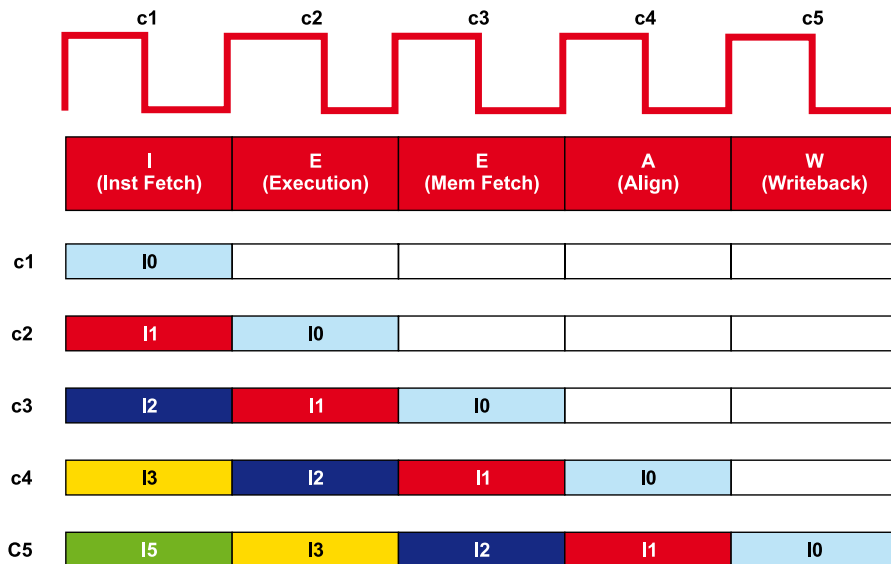
Kilka lat temu wszystkich pogodziły nowe mikrokontrolery z rdzeniem ARM. 32-bitowa jednostka zaprojektowana przez firmę specjalizującą się w projektowaniu i sprzedaży gotowych rdzeni okazała się najlepszym i chyba najtańszym sposobem na wyprodukowanie własnego szybkiego, niezawodnego mikrokontrolera. Na kupienie licencji od ARM zdecydowało się wiele firm, między innymi takie jak: NXP, STM i Atmel. Przepis na sukces wydawał się prosty: kupujemy dobry rdzeń, obudowujemy swoimi peryferiami i dodajemy własną technologię produkcji. Pierwszy na dużą skalę postąpił tak NXP, wypuszczając niewiarygodnie tanie 32-bitowe mikrokontrolery LPC21xx z rdzeniem ARM7. Krok w stronę unifikacji różnorodnego kiedyś rynku mikrokontrolerów został wykonany. Wydawało się, że instalowanie tego samego rdzenia w układach różnych producentów stanie się światowym standardem, jednak na tym monolicie pojawiły się rysy: Atmel zapowiedział, że będzie promował własny 32-bitowy rdzeń, a Microchip zdecydował się na zastosowanie rdzenia innego niż ARM.



Rys. 1. Schemat blokowy mikrokontrolera PIC32

PIC32

Microchip jest znany z własnych, cenionych na świecie konstrukcji mikrokontrolerów. Oferuje całą gamę rodzin mikrokontrolerów 8-bitowych, od najmniejszych PIC10F i PIC12F, poprzez najstarszą, ale produkowaną i nadal rozwijaną rodzinę PIC16F, do



Rys. 2. Potokowe wykonywanie rozkazów

bardzo udanej rodziny PIC18F Microchipowi udało się też „desant” na terytorium opalone przez silniejsze jednostki 16-bitowe. Uznanie zdobyły sobie mikrokontrolery z serii dsPIC30F i dsPIC33F z silnikiem DSP oraz szybkie mikrokontrolery PIC24F. Firma wyrobiła sobie dobrą markę dzięki rdzeniom własnej konstrukcji, bardzo dobrym peryferiom i przemysłowej linii rozwojowej swoich produktów. Znane są też i cenione jej narzędzia wspomagające proces projektowania i uruchamiania: sprzętowe debugery, programatory, moduły ewaluacyjne, kompilatory C i bezpłatne środowisko MPLAB. IDE i wersje studenckie kompilatorów C dostępne są bezpłatnie, a ceny i dostępność pozostałych narzędzi systematycznie spada. Jednak jak każda firma działająca na trudnym rynku, tak i Microchip, aby być konkurencyjnym, musi oferować nowe atrakcyjne rozwiązania.

Jak już wspominałem, Microchip zdecydował się wejść na rynek mikrokontrolerów 32-bitowych, ale wylamał się z ARM-owego monopolu i zastosował w swoim nowym 32-bitowym mikrokontrolerze rdzeń M4K, zaprojektowany przez firmę MIPS Technologies. Obecnie Microchip oferuje kilka typów

mikrokontrolerów rodziny PIC32, od PIC32-MX320F032 taktowanego z maksymalną częstotliwością 40 MHz, mającego tylko 32 kB pamięci programu Flash i 8 kB SRAM, do PIC32MX360F512 taktowanego z maksymalną częstotliwością 80 MHz i wyposażonego w 512 kB pamięci programu Flash oraz 32 kB pamięci SRAM. Co ciekawe, żadna z obecnie oferowanych wersji nie ma interfejsu USB, ale za to mają go wszystkie nowo zapowiadane wersje oznaczane PIC32MX460.

Schemat blokowy mikrokontrolera rodziny PIC32 pokazano na rys. 1. Rdzeń M4K bazuje na architekturze typu Harvard. Magistrale: danych (SDARAM) i rozkazów (Flash Memory) są rozdzielone.

MIPS Technologies podaje wydajność rdzenia na poziomie 1,5 Dhrystone MIPS/MHz. Dhrystone jest specjalnym testem opracowanym przez Reinholda Weickera w 1984 roku i jest używany do testowania wydajności jednostki centralnej. W czasie testu wykonywane są działania na liczbach całkowitych (integer) i operacje dostępu do plików. Test Dhrystone jest bardziej wiarygodny, niż proste podawanie wydajności w liczbie operacji na sekundę wynikające z częstotliwości

taktowania (bo testowi poddawana jest rzeczywista realizacja zadań).

Wynik testu wydajności samego rdzenia nie musi oznaczać takiej samej wydajności mikrokontrolera. Jeżeli wbudowana pamięć programu lub danych będzie zbyt wolna, to układ sterujący magistralą wprowadza opóźnienia (wait state) i wydajność się zmniejsza. Ponieważ dostęp do pamięci programu jest wolniejszy niż dostęp do pamięci RAM, to jak w wielu innych bardziej zaawansowanych mikroprocesorach, zastosowano pamięć SRAM Prefetch Cache. Do tej pamięci jest zapisywany blok szesnastu 128-bitowych słów z pamięci programu. W każdym 128-bitowym słowie zawarte są cztery 32-bitowe lub osiem 16-bitowych instrukcji. Do częstotliwości taktowania równej 30 MHz zastosowanie pamięci cache nie powoduje zwiększenia wydajności (dostęp do pamięci flash jest wystarczająco szybki). Powyżej 30 MHz stosowanie cache przynosi korzyści tym większe, im wyższa jest częstotliwość taktowania.

Zaawansowani programiści mogą mechanizm pobierania rozkazów z pamięci cache wyłączyć na przykład w czasie wykonywania czasowo krytycznych pętli programu, ponieważ wtedy (rozkazy skoków) pamięć cache może się każdorazowo na nowo zapelniać i zwalniać wykonywanie programu. W firmowych materiałach Microchipa można znaleźć procedury pozwalające sterować procesami wykorzystania pamięci cache.

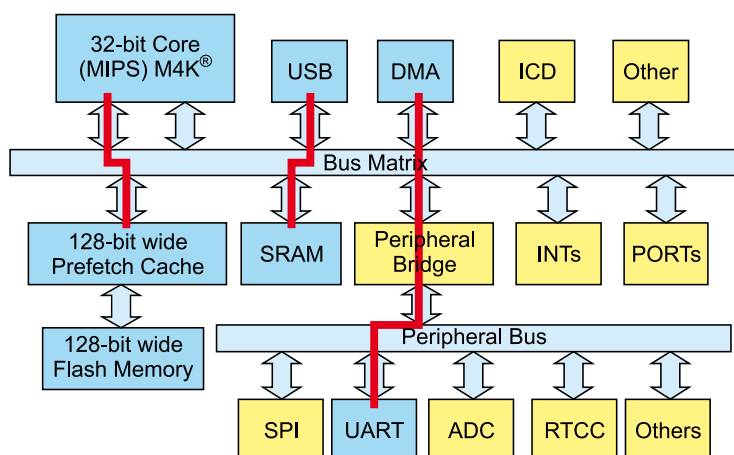
Rdzeń M4K wykonuje rozkazy w 5 cyklach rozkazowych (każdy z nich wykonywany w jednym cyklu zegara):

- I (Instruction fetch) pobierany jest kod rozkazu z pamięci programu Flash lub SRAM
- E (Execution) rozkaz jest dekodowany i wykonywany
- M (memory fetch) pobranie danych z pamięci danych SRAM
- A (align) cykl ustawiania w kolejce
- W (Writeback) dane przeznaczone do zapisania są zapisywane w miejscu przeznaczenia (np. w pamięci SRAM).

Żeby zwiększyć szybkość, zastosowano potokowe wykonywanie rozkazów (rys. 2). Potokowe wykonywanie rozkazów powoduje, że blok rozkazów, w którym nie ma rozkazu skoków, wykonuje się, tak jakby rozkaz wykonywał się w czasie jednego cyklu zegarowego.

Abstrahując od specjalizowanych mikrokontrolerów DSP, PIC32 ma unikalną właściwość niespotykaną w innych mikrokontrolerach z architekturą Harvard: rdzeń może wykonywać program zapisany w pamięci danych SRAM. Jest to możliwe dzięki bardzo szybkiej magistrali Bus Matrix.

Bus Matrix jest właściwie szybkim przełącznikiem potrafiącym połączyć moduły rdzenia, USB, DMA z modułami Flash memory, SRAM, GPIO, mostka magistrali pe-



Rys. 3. Przykład działania Bus Master

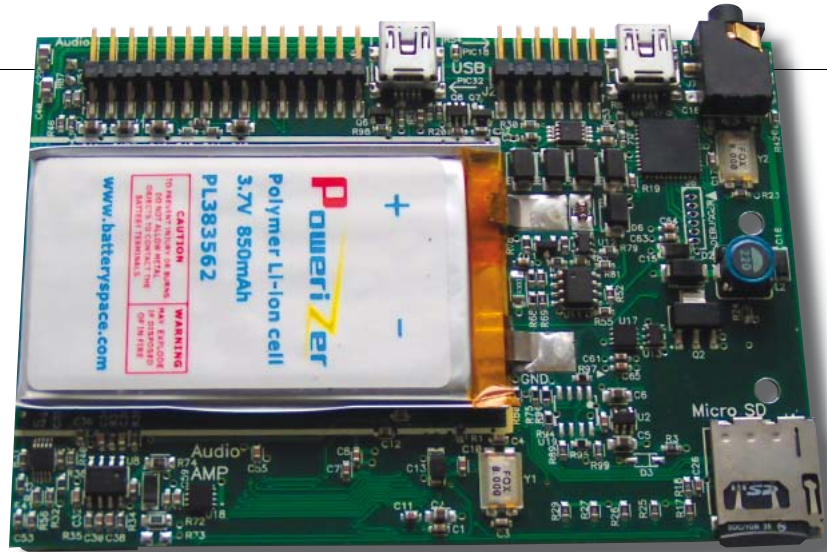
ryferii „punkt-punkt”. Bus Matrix pracuje z taką samą częstotliwością jak CPU, natomiast magistrala układów peryferyjnych Peripheral Bus może być taktowana inną częstotliwością.

Do zarządzania przepływem danych przez magistralę mikrokontroler wykorzystuje koncepcję nazywaną Bus Master. Definiuje ona sposób wymiany informacji modułu dołączonego do magistrali Bus Matrix przez określenie modułu docelowego, z którym moduł będzie wymieniał informacje. Na przykład można zdefiniować, że moduł DMA jest połączony z magistralą USB. Bus Master nie ma uprawnień do definiowania dostępu do CPU. Jest to mechanizm mogący wyraźnie zwiększyć prędkość wymiany informacji przez inicjowanie równoległych kanałów przepływu danych. Kiedy zostanie zainicjowany jeden kanał i rozpocznie się transfer danych „punkt-punkt”, to może być zainicjowany następny kanał wymiany informacji.

Na rys. 3 CPU inicjuje pobieranie kodów rozkazów z pamięci programu Flash Memory, a Bus Master inicjuje kolejno transfer danych pomiędzy magistralą USB i pamięcią danych SRAM oraz pomiędzy kanałem DMA i portem szeregowym UART.

Bus Master ma też wbudowane mechanizmy arbitrażu, w przypadku kiedy są przesyłane dane na przykład z modułu USB do pamięci SRAM i CPU w czasie wykonywania programu żąda dostępu do pamięci danych SRAM. Użytkownik może w takim przypadku programowo ustawić poziom priorytetów dostępu dla CPU, USB lub DMA.

Mikrokontroler ma prosty i wydajny system przerwania pracujący z jednym lub wieloma wektorami przerwania. W konfiguracji z jednym wektorem wszystkie przerwania są zgłaszane na jeden wektor. Kiedy układ jest skonfigurowany do pracy z wieloma wektorami, to każde źródło przerwania jest zgłaszane na swój wektor (dostępnych jest 64 wektorów). Każdy z wektorów ma programowo przypisany 8-poziomowy priorytet. Priorytet o wartości 7 jest najwyższy, a priorytet 0 oznacza zablokowanie wektora. System priorytetów jest uzupełniony o 4-poziomowe subpriorytety. Ustawiają one kolejność przerwania w przypadku, kiedy jednocześnie zo-



staną zgłoszone przerwania o takim samym priorytecie.

Przy obecnie dużej modzie na 32-bitowe mikrokontrolery z rdzeniami firmy ARM wprowadzanie na rynek mikrokontrolera z innym rdzeniem jest dość odważnym posunięciem. Rdzeń MIPS nie jest czymś nowym, ale nie jest zbyt popularny. Menedżerowie Microchipa zazwyczaj stawiali na dobre rozwiązania. Z wielu oferowanych produktów większość odniosła rynkowy sukces. Właściwie nie przyjęła się tylko rodzina mikrokontrolerów PIC17. Nowy dobry, czy nawet bardzo dobry produkt sam się nie sprzedaje i trzeba mu pomóc. Wiadomo czego, oczekuje potencjalny nabywca, żeby się zainteresować nowym mikrokontrolerem. Po pierwsze, pełnej i łatwo dostępnej informacji. Tu jak zwykle przychodzi z pomocą strona firmowa www.microchip.com. Znajdziemy tam wszystkie dane katalogowe produkowanych i planowanych do produkcji mikrokontrolerów. Oprócz danych technicznych są oczywiście noty katalogowe, przykładowe procedury, prezentacje szkoleniowe oraz gotowe rozwiązania: stos TCP/IP i USB, biblioteki audio i magistrali CAN.

Drugim elementem potrzebnym do poznania nowego mikrokontrolera są narzędzia: środowisko IDE, kompilatory języka C, programatory i emulatory. Nieodpłatnie można pobrać darmowy IDE MPALB8.0 lub wyższą oraz wersję studencką kompilatora MPLAB C32. Do programowania i emulacji można stosować znane i tanie narzędzia ICD2 (programator/emulator), oraz PICkit 2 i MPLAB PM3 (programatory). Ktoś, kto używa mikrokontrolerów Microchipa z narzędziami nie będzie miał problemu.

Starter Kit PIC32

Jeżeli już zdecydowaliśmy się na próby, to możemy zamówić u dystrybutora bezpłatne próbki. Można te próbki zamówić też przez Internet, ale dostępne są tylko dla sprawdzonych odbiorców. Jednak najlepszym sposobem łagodnego startu z PIC32 jest wykorzystanie gotowego modułu startowego. Tu Microchip również stanął na wysokości zadania, oferując tani zestaw PIC32 Starter Kit.

Zestaw jest umieszczony w plastikowym pudełku wielkości kasety VHS i zawiera: płytkę z mikrokontrolerem PIC32MX360F512, kabel USB i płytę CD z programami narzędziowymi i dokumentacją.

Przed pierwszym połączeniem modułu do komputera kablem USB trzeba włożyć do napędu płytę CD-ROM z zestawu. Uruchamia się automatycznie aplikacja setup.exe i pojawia się okno pokazane na rys. 4. Wszystkie niezbędne komponenty (środowiska MPLAB IDE V8.00 i kompilatora MPLAB C-32 (wersja studencka) itp.) są instalowane po kliknięciu na Install from CD.

Po ich zainstalowaniu można podłączyć moduł do portu USB. Sterowniki modułu zainstalują się automatycznie i będą widoczne w menedżerze urządzeń (rys. 5).

Nawigację po zawartości płyty ułatwia okno nawigatora widoczne po zamknięciu okna instalacji programów (rys. 6).

Wszelkie niezbędne informacje potrzebne do rozpoczęcia pracy z modułem zawarte są w dokumencie Starter kit user guide. Dla leniwych lub bardzo zapracowanych przygotowano multimedialny przewodnik Getting started. Prezentacja jest w języku angielskim. Oprócz wyjaśnień lektora można włączyć wyświetlanie czytanych informacji.

Z poziomu nawigatora można wygodnie przeglądać zawarte na płycie dane katalogowe obecnie produkowanych układów, zapoznać się z opisami kompilatora i bibliotek oraz z bardzo przydatnymi przykładami programów napisanych w assemblerze i w języku C. Dostęp do przykładów jest realizowany przez strony www.

Po instalacji wykonywanej na początku pracy z modułem wszystkie przykłady są też



Rys. 4. Instalacja oprogramowania dla PIC32 Starter Kit



Rys. 5. Manager urządzeń



kopiowane na twardy dysk w katalogu `C:\Program Files\Microchip\pic32_solutions`. Znajdziemy tam gotowe projekty dla MPLAB IDE i w osobnym katalogu tylko pliki źródłowe. Jak na początek pracy z nowymi mikrokontrolerami ilość przykładów jest imponująca. Są tam oprócz prostych procedur sterujących zapalaniem/gaszeniem diody LED również gotowe skomplikowane projekty.

Ponieważ PIC32 ma duże możliwości obliczeniowe i stosunkowo dużą pamięć, to naturalnymi aplikacjami dla niego będą systemy embedded wykorzystujące Internet jako medium do sterowania lub przesyłania danych. Microchip dawna umieszczał w swoich materiałach darmowy stos TCP/IP. Materiały dołączane do PIC32 są uzupełniane o klienta i serwer protokołu DHCP, serwer HTTP i agenta SNMP.

Procedury wspomagające zaawansowane wykorzystanie mikrokontrolera: kanału DMA, obsługi mechanizmu pamięci cache, obsługi przerwań od układów peryferyjnych, są umieszczone w katalogu `C:\Program Files\Microchip\pic32_solutions\labs`.

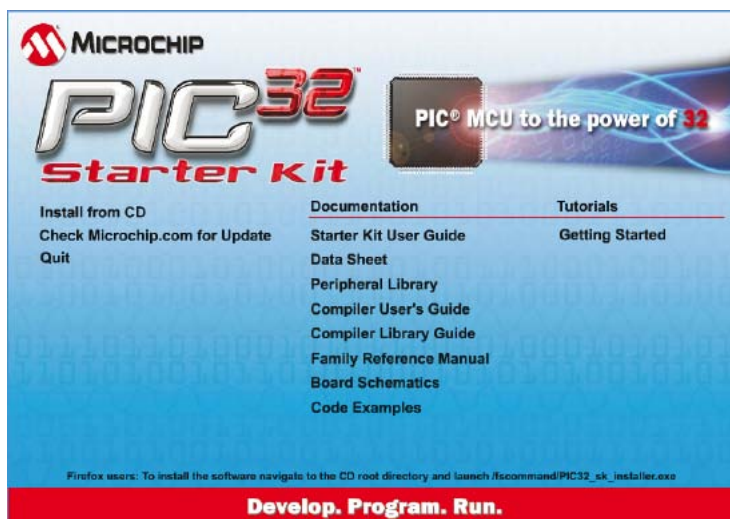
Po zainstalowaniu niezbędnego oprogramowania i przestudiowaniu dokumentacji można przejść do praktycznego poznawania nowego mikrokontrolera. Jeżeli moduł jest podłączony do portu USB, sterowniki zostały prawidłowo zainstalowane, to uruchamiamy MPLAB IDE V 8.00 lub wyższą.

Z katalogu `C:\Program Files\Microchip\pic32_solutions\simon_says_demo` otwieramy projekt prostej gry Simon says. W module PIC32 Starter kit dostarczanym przez producenta wgrana jest ta gra. W zakładce Debugger/Select tool wybieramy PIC32MX Starter Kit (rys. 7). MPLAB IDE nawiązuje połączenie USB z modulem i możliwe jest zaprogramowanie pamięci programu łącznie z bitami konfiguracji, odczytanie pamięci programu oraz debugowanie. Ta ostatnia opcja jest szczególnie przydatna przy uruchamianiu nowych programów.

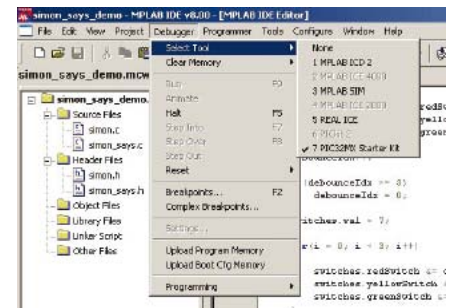
W trakcie debugowania na module zapala się żółta dioda „debug”.

Praca z MPLAB IDE jest doskonale znana użytkownikom mikrokontrolerów PIC i przejście na nowe, 32-bitowe układy jest wyjątkowo łagodne.

Sam moduł PIC32 starter kit jest niewielką płytką drukowaną o wymiarach 50×50mm. Oprócz głównego elementu: mikrokontrolera PIC32MX360F512, umieszczono na niej mikrokontroler PIC18LF4550, 2 generatory 8 MHz, układ zasilania +3,3 V, złącze mini USB, diody LED sygnalizujące wejście w tryb debatowania (DEBUG), włączone zasilanie (Pwr) i 3 diody połączone z liniami portów LED1 (RD0), LED2 (RD1) i LED3 (RD2). Do linii portów RD6, RD7 i RD13 podłączone zostały styki SW1, SW2 i SW3.



Rys. 6. Okno nawigacji płyty PIC32 Starter Kit



Rys. 7. Wybór debugera/programatora PIC32MX Starter Kit

Mikrokontroler PIC18LF4550 spełnia funkcję konwertera USB/ICSP. Jak już wspomniałem, PIC32MX360 nie mają interfejsu USB i zastosowanie dodatkowego układu było niezbędne. Na złącze J3 wyprowadzono sygnały MCLR, PGD, PGC oraz napięcia +3,3 V i masę. Można przez to złącze programować lub debatować inne mikrokontrolery.

Generatory 8 MHz taktują mikrokontrolery PIC32 i PIC18. Na płytce przewidziano też miejsce na wlutowanie oscylatora „zegarkowego” 32 kHz (punkty Y3) podłączonego do wyjść licznika T1.

Jako stabilizator napięcia 5 V→3,3 V zastosowano układ MCP1603.

Umieszczenie na płytce 3 przycisków i 3 diod LED wystarcza na zaledwie początkowe proste eksperymenty. Umieszczenie na niej silnego procesora z układem debugera predestynuje ją do wykorzystania w bardziej zaawansowanych przedsięwzięciach. Pewnie też o tym pomyśleli projektanci modułu umieszczając na nim wielopinowe złącze. Jednak według mnie zastosowanie tego złącza jest w zasadzie jedyną poważną wadą starter kitu. Złącze jest nietypowe i najprawdopodobniej będzie trudne do zdobycia, a bez niego mamy płytkę, którą nie bardzo da się wykorzystać po wstępnym przetestowaniu możliwości PIC32. Co prawda producent oferuje ekspander portów, ale są to dodatkowe koszty (według microchip direct ok. 52,60 Euro) i nie ma możliwości zastosowania modułu jako tak zwany dip moduł. Z mojego punktu widzenia lepszym rozwiązaniem byłoby zwiększenie rozmiaru płytki i umieszczenie na przykład pół lutowniczych na standardowe goldpiny o rastrze 2,54 mm.

Mimo tej niedogodności trzeba uznać, że starter kit jako całość jest bardzo udanym pomysłem. Jego niezaprzeczalnymi atutami jest bogate wsparcie programowe, możliwość podłączenia do komputera przez standardowy port USB bez konieczności stosowania dodatkowych programatorów/debugerów. Jeżeli dodamy do tego niską cenę (ok. 36,5 euro), to jest to najlepszy sposób na zaspokojenie pierwszej ciekawości wszystkich potencjalnych zainteresowanych.

Tomasz Jabłoński, EP
tomasz.jablonski@ep.com.pl