

Każdy chciałby być pięknym, młodym i bogatym. Elektronicy marzą o szybkich, niepobierających energii i tanich mikrokontrolerach. Niestety, w obu przypadkach są to warunki trudne do jednoczesnego spełnienia. Czasami jednak...

Mikrokontrolery 8-bitowe HCS08

Oprócz dużej szybkości pracy, niewielkim zapotrzebowaniu na energię i niskiej cenie, coraz częściej oczekiwanym parametrem mikrokontrolerów jest również małe napięcie zasilające, pozwalające na stosowanie tych układów w urządzeniach zasilanych bateryjnie. Spełnienie wszystkich wymagań jest trudne, ale przy zastosowaniu pewnych wybiegów można uzyskać całkiem zadawalające rezultaty. Przykładem może być rodzina mikrokontrolerów HCS08, produkowanych przez Motorolę. Zastosowany w nich system zarządzania mocą pozwala na zredukowanie poboru prądu przez mikrokontroler

ze źródła zasilającego do wartości zaledwie 20 nA w trybie power-down. No nieźle, nieźle. Robi się interesująco, ale szczegóły zostaną podane w dalszej części artykułu, gdyż układy te warte są dokładniejszego, systematycznego przyjrzenia się im.

Rodzina mikrokontrolerów HCS08 od środka

Mikrokontrolery serii HCS08 zostały opracowane z myślą o zastosowaniach w sprzęcie powszechnego użytku z zasilaniem bateryjnym. Mimo tylko 8-bitowej budowy rdzenia i niskiej ceny, układy te charakteryzują się całkiem niezłymi parametrami. Wszystkie mikrokontrolery tej rodziny posiadają identyczną jednostkę centralną, a jedyne różnice między nimi to wielość i typ pamięci oraz rodzaj obudowy. CPU może być takto wyzwalany sygnałem zegarowym o częstotliwości maks. 40 MHz. W wielu przypadkach wystarczy natomiast wewnętrzny oscylator niewymagający zewnętrznego rezonatora kwarcowego. Przebieg zegarowy generowany w takim trybie pracy ma częstotliwość 8 MHz. Zgodnie z panującymi trendami lista rozkazów procesora została zoptymalizowana pod kątem języka

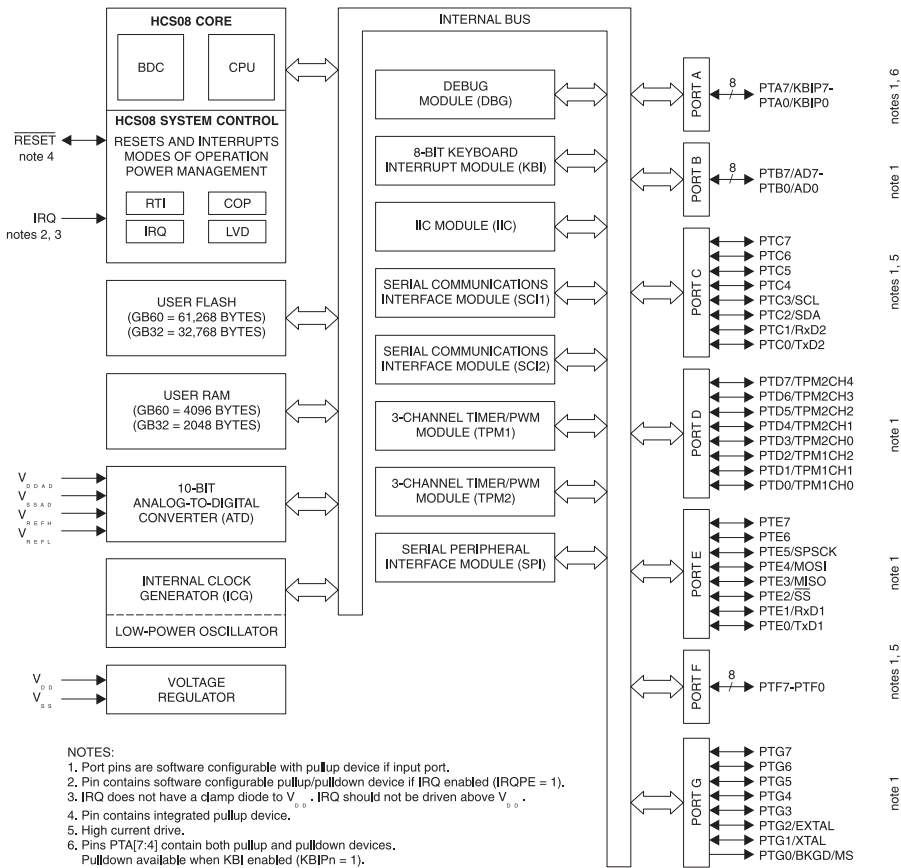


MOTOROLA

C. Na uwagę zasługuje rozkaz BGND przeznaczony do celów uruchomieniowych. W ostatecznych wersjach programów mikrokontrolera rozkaz ten powinien być usuwany, ale w wersjach testowych jest bardzo przydatny (wreć niezbędny) do zakładania pułapek i śledzenia pracy CPU. Rozkaz BGND wymaga, by program debugera był przystosowany do jego wykorzystania. Oferowane przez Motorolę środowisko uruchomieniowe - tradycyjnie już CodeWarrior - spełnia oczywiście ten warunek. Z listą rozkazów procesora związane są w pewnym stopniu jeszcze dwie inne cechy, jakimi charakteryzują się układy rodziny HCS08. Są to: detekcja nielegalnego kodu operacji oraz nieprawidłowego adresu. W obu przypadkach po wykryciu błędu następuje zerowanie mikrokontrolera. Nad prawidłowym wykonywaniem programu czuwa ponadto tzw. COP (Computer Operating Properly) Watchdog. Zasada jego działania, mimo dość dziwnej nazwy, jest tradycyjna - polega na wpisywaniu określonej wartości do rejestru specjalnego timera (COP) w różnych miejscach programu tak, aby jego stan nigdy nie osiągnął wartości zerowej. Rejestr ten jest samoistnie dekrementowany z ustaloną częstotliwością. Niespełnienie powyższego warunku powoduje wyzerowanie mikrokontrolera.

Na rys. 1 przedstawiono schemat blokowy jednostki centralnej mikrokontrolerów rodziny HCS08 w wersji GB. Już na pierwszy rzut oka widać dość pokaźną liczbę wyprowadzeń uniwersalnych portów we/wy. Jest ich aż 56. Mimo że w większości przypadków pełnią one funkcje alternatywne, to i tak do zastosowań ogólnych pozostaje ich sporo. Liczba i rodzaj zastosowanych bloków funkcjonalnych powinna zadowolić nawet najwybredniejszych konstruktorów. Mamy więc: podwójny interfejs SCI (Serial Communications In-





Rys. 1

terface); interfejs SPI (*Serial Peripheral Interface*); interfejs I²C (choć z wiadomych powodów występuje pod oznaczeniem IIC); 8-kanałowy, 10-bitowy przetwornik analogowo-cyfrowy (ATD); jeden 3-kanałowy i jeden 5-kanałowy timer z możliwością pracy jako modulator PWM; 8-bitowy moduł klawiatury (KBI) obsługiwany poprzez system przerwania (sic!). Ostatni komponent nie jest często umieszczany w mikrokontrolerach, a bywa bardzo przydatny w aplikacjach wykorzystujących klawiatury matrycowe. Dołączenie do mikrokontrolera elementów stanowiących duże obciążenia prądowe (np. bezpośrednio sterowane LED-y) jest możliwe dla 16 wyjść układu. Wyprowadzenia portów posiadają programowo włączane układy podciągania typu pull-up (indywidualnie dla każdego pinu). Jednostka centralna obsługuje 32 wektory przerwania. Pewną niedogodnością dla konstruktora jest możliwość korzystania tylko z jednego wejścia przerwania zewnętrznego, ale mimo tego, wymienione wcześniej cechy mikrokontrolerów rodziny HCS08 powodują, że konstruktorzy na pewno chętnie będą po nie sięgać. Układy te staną się idealne w urządze-

niach, w których korzysta się z wielu różnorodnych bloków funkcjonalnych. Opracowanie takich aplikacji od strony programowej wymaga sporego wysiłku od konstruktora. Dziś się właściwie nikt już nie robi tego inaczej niż w języku C. I tu często okazuje się, że zaczyna brakować pamięci programu. W przypadku mikrokontrolerów HCS08 jest to raczej mało prawdopodobne, jako że w zależności od typu mają one od 32 do 60 kB Flasha oraz od 2 do 4 kB pamięci RAM. Pamięć Flash może być oczywiście programowana w układzie, bez konieczności stosowania zewnętrznych programatorów. Mikrokontrolery HCS08 są produkowane w obudowach LQPF-64, QFP-64 i SDIP-42.

Prawie un-plug

Jak już było powiedziane wcześniej, mikrokontrolery HCS08 zadawalają się niskim napięciem zasilającym, jak również pobierają niewielki prąd ze źródła. Wykorzystując tzw. auto-wakeup timer możliwe jest ograniczenie tego prądu do wartości średniej około 300 nA. Timer auto-wakeup powoduje automatyczne wybudzenie mikrokontrolera ze stanów uśpienia. Pracuje z częstotli-

wością 2kHz. Mikrokontroler może przy tym pracować z napięciem zasilającym o wartości zaledwie 1,8 V. Dla podkreślenia tego faktu firma Motorola opracowała bardzo wymowny zestaw demonstracyjny. Płytkę M68DEMO908GB60 wraz z oprogramowaniem narzędziowym na CD-ROM-ie umieszczono w okrągłej obudowie przypominającej baterię typu „paluszek”. Podkreśleniem faktu, że mikrokontrolery mogą pracować przy zasilaniu bateryjnym, jest też umieszczenie pojemnika z dwoma alkalicznymi bateriami AA na tej płytce. Programy demonstracyjne działają przy takim zasilaniu. Pewną atrakcją jest również umieszczony na CD-ROM-ie, specjalny program-kalkulator, służący do szacowania czasu działania danego urządzenia dla określonych warunków zasilania (rys. 2). Jako argumenty podaje się tu m.in. rodzaj zastosowanego ogniwa, procentowy czas przebywania CPU w określonym stanie, częstotliwość rezonatora oraz wykorzystanie przetwornika ADC. Program oblicza przewidywany czas działania urządzenia na jednym komplecie baterii. Spróbujmy teraz wyjaśnić, jak uzyskano zadziwiająco małe zapotrzebowanie na energię przez mikrokontrolery HCS08. Pierwszym czynnikiem jest zastosowanie odpowiedniej technologii wytwarzania struktury półprzewodnikowej. Okazało się to jednak niewystarczające, aby uzyskać zadawalające rezultaty. Opracowano więc trzy tryby pracy mikrokontrolera: tryb aktywny (*active background mode*), tryb oczekiwania (*wait mode*) i tryb zatrzymania pracy (*stop mode*). W pierwszym przypadku CPU pracuje bez żadnych ograniczeń, wykonując kolejne rozkazy umieszczone w pamięci programu. W trybie aktywnym możliwe jest śledzenie programu przez układ BDC (*Background Debug Controller*). Jak wiadomo, jest to możliwe przy



Rys. 2

wsparciu przez odpowiednio napisany program debuggera. Wejście w tryb *wait* następuje po wykonaniu rozkazu WAIT. CPU jest wówczas przełączane w stan obniżonego poboru energii (*low-power*), w którym zostaje wyłączony oscylator. Pozostają natomiast aktywne przerwania, które mogą przywrócić normalny tryb pracy. W stanie *wait*, z oczywistych powodów powstają ograniczenia pracy układu BDC. Dostępny jest jedynie specjalny rozkaz BACKGROUND, wyprowadzający jednostkę centralną ze stanu *wait* oraz rozkazy dostępu do pamięci tzw. *memory-access-with-status*. W stanie zatrzymania pracy (*stop mode*) wyróżnia się trzy tryby ustawiane odpowiednimi bitami w rejestrze konfiguracyjnym. Pierwszy z nich - *Stop1 mode*, dzięki wprowadzeniu niemal wszystkich bloków mikrokontrolera w stan *power-down*, zapewnia największą oszczędność energii. Wyłączony jest nawet wewnętrzny regulator napięcia. Wyprowadzenie układu z tego stanu jest możliwe tylko poprzez sygnał podany na linię zewnętrznego przerwania lub zerowania mikrokontrolera. MCU budzi się z tego stanu tak, jak po włączeniu zasilania (*power-on reset*). Nieznacznie większe zużycie energii obserwujemy w stanie *Stop2 mode*. W tym przypadku zachowywana jest zawartość wewnętrznej pamięci RAM mikrokontrolera, utrzymywane są również aktywne stany na wyprowadzeniach *we/wy*. Rejestry portów *we/wy* (oraz inne istotne rejestry procesora) powinny być jednak zapisane w pamięci RAM przed uśpieniem mikrokontrolera. Powrót do normalnej pracy może nastąpić po odpowiednim sygnale na wejściu /RESET, IRQ lub RTI. Jak w poprzednim przypadku, tak i tym razem procesor budzi się w sposób analogiczny z sygnałem *power-on reset*. Trzeba pamiętać o tym, że stany wszystkich portów są ustawiane na wartości domyślne. Konieczne może więc być ewentualne przywrócenie stanów sprzed uśpienia, jeśli tylko

były zapisane w wewnętrznej pamięci RAM. Ostatnim trybem oszczędzania energii jest *Stop3 mode*. Powoduje on zatrzymanie pracy wszystkich oscylatorów oprócz timera auto-wakeup. W tym trybie nie jest konieczne zachowywanie zawartości rejestrów. Pozostają one niezmiennie po obudzeniu procesora, podobnie jak wewnętrzna pamięć RAM. Wyjście ze stanu *Stop3* następuje po sygnale na wejściu /RESET lub na liniach przerwań asynchronicznych: IRQ lub KBI. Wybudzenie CPU może wywołać również tzw. przerwanie real-time. Praca jednostki centralnej jest kontynuowana począwszy od wektora odpowiedniego przerwania.

Płytki demonstracyjna M68DEMO908GB60

Najłatwiejszą metodą zapoznania się z mikrokontrolerami rodziny HCS08 jest wykonanie kilku eksperymentów przy użyciu płytki demonstracyjnej M68DEMO908GB60. Zamontowano na niej układ M9S08GB60 oraz kilka zewnętrznych bloków funkcjonalnych wraz z odpowiednimi gniazdami. Są to: stabilizator napięcia zasilającego 3,3 V z wyłącznikiem, dwa interfejsy szeregowy ze złączami DB-9, 4 przełączniki, 5 diod LED, gniazdo portów *we/wy*, źródło referencyjne 1,8 V, wejścia sygnałów analogowych dla przetwornika ADC. Jest też niewielkie pole do montażu prototypowego oraz pojemnik na dwie baterie AA. Firma Motorola przygotowała kilka programów przykładowych, które są fabrycznie zapisane w pamięci układu. Wyboru dokonuje się poprzez naciśnięcie odpowiedniego przycisku w chwili zerowania mikrokontrolera. W zestawie zawarto CD-ROM z zamieszczonymi na nim wersjami źródłowymi programów. Użytkownik może więc dokładnie prześledzić, jak wygląda programowa obsługa demonstrowanych trybów pracy i obsługi wybranych komponentów mikrokontrolera. Mając na względzie reklamowane cechy rodziny HCS08, najbardziej interesującym

będzie zapewne program demonstrujący pracę układu w trybie *Stop3*. Po jego wybraniu jeden z LED-ów błyska na krótko co ok. 1 s. Między błyskami CPU jest wprowadzane w stan *Stop3*, a wybudzanie zapewnia przerwanie real time regularnie generowane przez specjalny oscylator. Zmierzony, średni prąd zasilania (zastosowano zasilanie baterijne) wynosił około 0,85 mA. Mikrokontroler był taktowany zegarem o częstotliwości 32768 Hz. Kolejne programy demonstrują obsługę przetwornika analogowo-cyfrowego, generację przebiegu PWM oraz mechanizm okresowo wywoływanych pętli programu. Oprócz tego można oczywiście stworzyć i uruchamiać własne eksperymenty. Próby mogą być prowadzone w trybie normalnej egzekucji lub krokowo, z wykorzystaniem debugera. W tym przypadku oprogramowanie zainstalowane na komputerze PC musi komunikować się z mikrokontrolerem i dlatego „zabierany” jest jeden z interfejsów szeregowych. Drugi natomiast może służyć jako interfejs wykorzystywany przez uruchamiany program.

Jak wynika z przeprowadzonych prób, hasło reklamowe Motoroli „Low Power. High Performance” wymyślone dla rodziny mikrokontrolerów HCS08 nie jest tylko sloganem. Układy te rzeczywiście posiadają powyższe cechy. Jeśli dodamy do tego jedno z lepszych środowisk uruchomieniowych IDE dla małych mikrokontrolerów - CodeWarrior, to okaże się, że możemy tworzyć bardzo wydajne, tanie i wygodne w opracowywaniu systemy, a to przecież tak, jak być pięknym, młodym i bogatym.

Jarosław Doliński
jaroslaw.dolinski@ep.com.pl

Dodatkowe informacje

Więcej informacji o mikrokontrolerach rodziny HCS08 można znaleźć na stronach:
- <http://e-www.motorola.com/webapp/sps/site/taxonomy.jsp?nodeId=03t3ZGpnLn84498634>
- http://e-www.motorola.com/files/microcontrollers/doc/data_sheet/MC9S08GB60.pdf