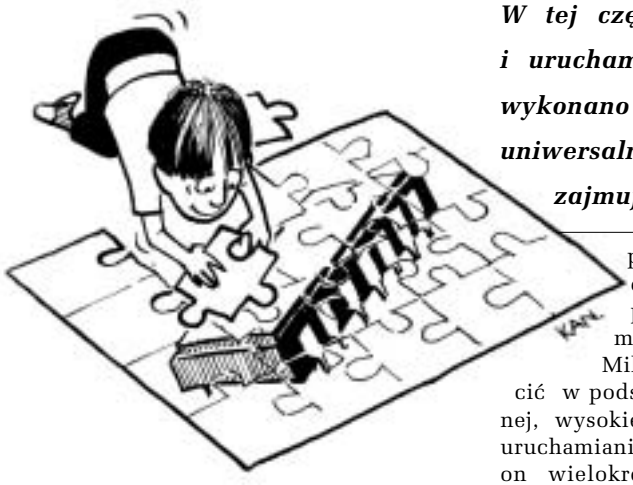
Podstawy projektowania systemów mikroprocesorowych, część 12



W tej części kursu skupiłem się na omówieniu zasad montażu i uruchamiania urządzeń elektronicznych, zwłaszcza tych, które wykonano z użyciem mikrokontrolerów. Zasady te mają walor uniwersalności, więc mogą przydać się także elektronikom zajmującym się techniką analogową.

Montaż i uruchamianie systemu mikroprocesorowego

Mając gotowy projekt urządzenia, możemy przystąpić do jego wykonania. W przypadku firm zajmujących się masową produkcją tego rodzaju urządzeń wiele etapów produkcji realizowanych jest jednocześnie, czego jednak nie można zastosować w przypadku układu amatorskiego, gdzie zazwyczaj całą pracę wykonuje jedna osoba w następujących etapach:

- projektowanie i wykonanie płytki drukowanej oraz montaż elementów,
- uruchamianie i testowanie części sprzętowej,
- pisanie programu,
- uruchamianie i testowanie programu.

Postaram się przedstawić zasady postępowania w poszczególnych etapach prac oraz zwrócić uwagę na pułapki czyhające na konstruktorów.

Etap 1

Jeżeli konstruktor wykonał kiedykolwiek jakieś urządzenie elektroniczne, to w zasadzie nie potrzebuje żadnych dodatkowych informacji do realizacji tego etapu projektu. Obowiązujące zasady postępowania są identyczne jak w przypadku innych urządzeń elektronicznych. Należy jedynie pamiętać o tym, aby zwracać uwagę na elektryczną poprawność i czystość montażu - jakiekolwiek zimne luty, zwarcia, pęknięcia ścieżek czy inne tego rodzaju uszkodzenia mogą być bardzo trudne do wykrycia, zwłaszcza jeżeli pojawią się na liniach, których stan może być trudny do określenia podczas pracy programu (np. magistrale danych czy adresowe, linie portu szeregowego). Nie będzie przesadą sprawdzenie każdej ścieżki omomierzem przed i po wykonaniu montażu.

Mikrokontroler należy umieścić w podstawce, najlepiej precyzyjnej, wysokiej jakości, gdyż na etapie uruchamiania oprogramowania będzie on wielokrotnie wyjmowany w celu zmiany zawartości pamięci Flash.

Podczas projektowania płytki drukowanej należy pamiętać o takim ułożeniu elementów, aby w pobliżu krótszej krawędzi mikrokontrolera (w obudowie DIP) nie było żadnych przeszkód (np. wysokich kondensatorów, przekaźników) uniemożliwiających łatwe wyjęcie go z podstawki (przez np. podważenie śrubokrętem). Brak dobrego „dojścia“ w najlepszym przypadku skończy się połamaniem nóżek mikrokontrolera.

Etap 2

Po zmontowaniu urządzenia należy przeprowadzić sprawdzenie poprawności jego działania. Na początku dołączamy do urządzenia zasilanie (najlepiej stosować zasilacze z ograniczeniem prądowym) i sprawdzamy napięcia w układzie, zwłaszcza na szynie zasilającej. Jeżeli wszystko jest w porządku, to możemy przystąpić do testów.

Najpierw wkładamy w podstawkę mikrokontroler (oczywiście po odłączeniu zasilania) i sprawdzamy działanie oscylatora kwarcowego generującego sygnał zegarowy. Oscyloskopem lub częstotliwościomierzem sprawdzamy przebiegi na wyprowadzeniu XTAL1 lub XTAL2 przy użyciu wysokoimpedancyjnej sondy. W przypadku „dużych“ '51 można również sprawdzić występowanie przebiegu na wyprowadzeniu ALE. Oprócz sprawdzenia przebiegów zegarowych należy również woltomierzem sprawdzić wyprowadzenie końcówki RESET - napięcie na tym wyprowadzeniu nie powinno przekraczać 1 V - jeżeli jest inaczej, to należy wymienić kondensator obwodu zerowania na inny o mniejszej upływności. Podczas tych prób nie należy obawiać się, że coś złego stanie się z układem wskutek braku programu w mikrokontrolerze - jeżeli testy przeprowadzimy z mikrokontrolerem fabrycznie nowym (lub wcześniej używanym, ale wyzerowanym), to podczas jego pracy na wyprowadzeniach będą utrzymywały się stany linii takie jak po wyzerowaniu, a współpracujący z mikrokontrolerem układ musi być na taki stan odporny. Dzieje się tak dlatego, że cała pamięć programu wypełniona jest jedynkami logicznymi, co w przekształceniu na instrukcje asemblera przekłada się na cykliczne wykonywanie instrukcji MOV R7,A, która nie ma wpływu na stan wyprowadzeń.

Po sprawdzeniu elementów wpływających na poprawną pracę mikrokontrolera należy wyjąć go z podstawki i przejść do testów układów peryferyjnych. W tym celu na określonych wyprowadzeniach mikrokontrolera wywołujemy stany logiczne odpowiadające sekwencjom sterującym poszczególne urządzenia - jedynkę logiczną uzyskujemy łącząc wyprowadzenie do Vcc mikrokontrolera poprzez rezystor 10...47 kΩ, natomiast zero logiczne uzyskujemy zwierając wyprowadzenie do masy. W ten sposób możemy sprawdzić działanie większości układów współpracujących z mikrokontrolerem.

Niestety nie jest możliwe sprawdzenie w ten sposób skomplikowanych układów synchronicznych, reagujących na sekwencje stanów linii - po prostu nie będziemy w stanie ich wygenerować, chociażby wskutek wielokrotnych impulsów powstających przy dołączaniu wyprowadzeń do masy czy Vcc. W tym przypadku pozostaje nam jedynie sprawdzenie poprawności połączenia elektrycznego konkretnej nóżki mikrokontrolera z wyprowadzeniem współpracującego elementu. Na tym etapie należy również sprawdzić napięcia poziomów logicznych na wyprowadzeniach poszcze-

gólnych układów, aby uniknąć kłopotów z ich interpretacją przez mikrokontroler na etapie uruchamiania programu.

Po sprawdzeniu wszystkich elementów układu i usunięciu ewentualnych nieprawidłowości możemy przystąpić do pisania programu.

Etap 3

Należy pamiętać, że jest potrzebny jakiś czas na wykonywanie poszczególnych instrukcji, a zwłaszcza czas wykonywania pętli, który może być bardzo długi mimo niezbyt skomplikowanej sekwencji rozkazów. Należy o tym pamiętać zwłaszcza podczas pisania procedur obsługi przerwań - zbytnio rozbudowana procedura obsługi przerwania może całkowicie zablokować pracę programu głównego (kolejne wywołanie przerwania zanim zakończy się działanie programu obsługi poprzedniego).

Z punktu widzenia czysto dokumentacyjnego warto zadbać o wprowadzanie komentarzy do kodu źródłowego opisujących krok po kroku działanie programu oraz przyjęcie nazewnictwa etykiet opisujących komórki pamięci zgodnych z przechowywaną zawartością (np. WYNIK, BAJT_WEJSCIOWY, BLOKADA, itp.). Znakomicie ułatwi to ewentualne modyfikowanie programu opracowanego np. pół roku temu - same suche instrukcje asemblera niestety nie mówią zbyt wiele, zwłaszcza jeżeli nie pamiętamy już działania poszczególnych bloków programu.

Oczywiście, program nie musi być pisany w asemblerze - obecnie do dyspozycji projektantów wykorzystujących w swoich projektach mikrokontrolery '51 jest wiele różnych kompilatorów języka C, Pascal czy Basic.

Etap 4

Etap tan jest w zasadzie najbardziej pracochłonny - rzadko bowiem się zdarza, że napisany program „rusza" za pierwszym razem i pracuje bez błędu.

Nie bez powodu wśród programistów krąży pogląd, że napisanie programu to 10% pracy, następne 90% to usuwanie błędów i wprowadzanie zabezpieczeń związanych z występowaniem sytuacji nietypowych. Dlatego już na etapie pisania programu warto zaplanować podział pracy na kilka części i uruchamianie każdego fragmentu programu oddzielnie, dopiero po stwierdzeniu poprawności działania części poprzedniej. Najlepiej zacząć od elementów programu przydatnych do testowania kolejnych części, a następnie przechodzić do funkcji bardziej szczegółowych, ale mniej istotnych z punktu widzenia użytkownika urządzenia. Przykładową sekwencję prac przedstawiono poniżej:

- Oprogramowanie timerów i ewentualnie obsługi ich przerwań wykorzystywanych do synchronizacji pracy poszczególnych elementów programu.
- Napisanie procedury obsługi wyświetlacza i próba wyświetlenia jakiejkolwiek informacji.
- Napisanie procedury obsługi klawiatury.
- Przygotowanie procedur obsługujących komunikację z użytkownikiem, np. wyświetlanie komunikatów, prezentacja wyników, reakcja na naciskanie klawiszy.
- Przygotowanie fragmentów programu związanych z elementami współpracującymi: obsługa RS-232, I^2C, dołączonych przekaźników, czujników itp.
- Przygotowanie funkcji realizujących pomiary czy w inny sposób pozyskujących dane.
- Przygotowanie procedur obliczeniowych wykonujących operacje na gromadzonych danych i wyznaczających wyniki obliczeń przeznaczone do prezentacji.

Zaprezentowana sekwencja nie musi być identyczna dla każdego systemu mikroprocesorowego, najlepiej jednak jest, gdy program przygotowujemy z grubsza zgodnie z przepływem danych, gdzie wyniki działania jednej części programu są danymi dla innej.

Podczas pisania oprogramowania obsługi kolejnych urządzeń czy realizujących obliczenia, zawsze należy uwzględniać wszystkie możliwe do wystąpienia sytuacje. Chodzi tutaj np. o zakłócenia związane z transmisją RS-232 (konieczność wprowadzenia maksymalnego czasu oczekiwania na dane), przepełnienie zakresu liczbowego podczas obliczeń (przewidywanie maksymalnej wartości wyników) i inne, a także o odporność np. na nieodpowiednie kombinacje przyciskanych klawiszy, kilka klawiszy wciśniętych jednocześnie, próba wprowadzenia parametru spoza dozwolonego zakresu, itp.

Wszystkie problemy tego rodzaju powinny być uwzględnione i odpowiednio rozwiązane. Należy także pamiętać o praktycznym sprawdzeniu tych zabezpieczeń (czyli spróbować wywołać takie zdarzenie).

W przypadku, gdy napisany przez nas program nie działa lub działa nieprawidłowo, należy oczywiście odnaleźć błąd i go usunąć. Poszukiwania błędu wbrew pozorom nie należy rozpoczynać od „grzebania" w kodzie źródłowym, ale od zastanowienia się nad algorytmem wykorzystywanym do wykonania danej operacji. Często na tym etapie można znaleźć jakiś błąd w rozumowaniu czy inny błąd algorytmu. Po upewnieniu się co do poprawności algorytmu możemy zacząć sprawdzać kod źródłowy. Najpierw należy sprawdzić, czy napisany przez nas program realizuje określony algorytm - błędy jak zwykle tkwią w szczegółach: np. licznik liczący od 1 zamiast od 0 (lub odwrotnie), źle określone skoki warunkowe (przy nieprawidłowym warunku), brak rozkazu powrotu do programu głównego (jeżeli procedura wywoływana przez LCALL), itp.

Niekiedy pojawiają się problemy ujawniające się dopiero przy wywoływaniu konkretnych fragmentów programu, na przykład zbytnie rozrastanie się stosu związane z dużą liczbą wywołań LCALL - powoduje to zamazywanie najniżej leżących komórek pamięci przechowujących istotne dane.

Uruchomienie oprogramowania i zakończenie jego testów w zasadzie kończy proces budowy układu mikroprocesorowego. Pozostaje jeszcze przygotowanie odpowiedniej obudowy i można przystąpić do używania zaprojektowanego, wykonanego i oprogramowanego urządzenia.

Zakończenie

Mam nadzieję, że prezentowany artykuł przybliżył Czytelnikom zagadnienia związane z projektowaniem systemów mikroprocesorowych, a także zwrócił im uwagę na fakt, że wbrew pozorom zbudowanie urządzenia z mikrokontrolerem nie jest ani trudne, ani drogie, a pełen sukces zależy wyłącznie od przestrzegania kilku prostych zasad oraz trzymania się wskazówek producenta dotyczących parametrów wykorzystywanych układów.

Korzystając z nowoczesnych pakietów uruchomieniowych, również oprogramowanie takiego systemu nie jest trudne, a nic tak przecież nie cieszy jak samodzielnie opracowane urządzenie wyposażone we wszystkie potrzebne funkcje, którego działanie znamy od podszewki i w razie konieczności będziemy potrafili dowolnie je zmodyfikować.

Życzę wszystkim Czytelnikom sukcesów w projektowaniu urządzeń z mikrokontrolerami i wydaje mi się, że jednym z efektów publikacji tego cyklu będzie duża liczba urządzeń mikroprocesorowych prezentowanych w EP, również w dziale „Projekty Czytelników".

Paweł Hadam, EP
pawel.hadam@ep.com.pl