

W poprzednim numerze "Elektroniki dla Wszystkich" zaznajomiliśmy Czytelników z pojęciem mikroprocesora i mikrokontrolera. Przedstawiliśmy ogólne założenia dotyczące budowy układów scalonych tego typu oraz przybliżony sposób współpracy z innymi układami peryferyjnymi. W tym odcinku przedstawimy ogólny opis wyprowadzeń mikroprocesora. Jest to drugi z odcinków wstępnych z cyklu obejmującego naukę programowania procesora 8051. Przypominamy, że wkrótce zamknijemy listę kandydatów na uczniów w "klasie mikroprocesorowej". Na zgłoszenia chętnych czekamy do końca maja. "Klasa mikroprocesorowa" to grupa 20-30 osób, które otrzymają bezpłatnie od firmy AVT zestaw edukacyjny (składający się z dwóch płytek z procesorem, klawiaturą, wyświetlaczami itp.). Osoby te zobowiązane będą do przeprowadzania wszystkich prostych ćwiczeń z zakresu nauki programowania oraz do zgłaszania autorowi cyklu wszelkich wynikłych niejasności czy problemów. Ma to na celu praktyczne sprawdzenie stopnia opanowania przedstawionego materiału, a także niewątpliwie zapewni skuteczność nauki. Do skorzystania z tej możliwości zapraszamy osoby w różnym wieku, od 12 do 80 lat.



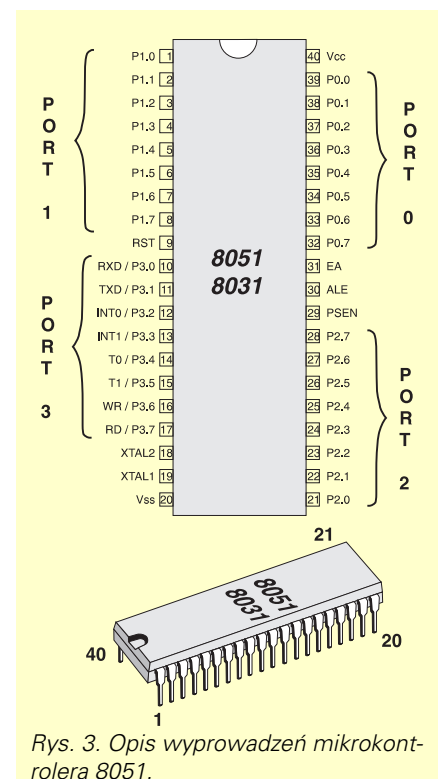
Dlaczego 8051?

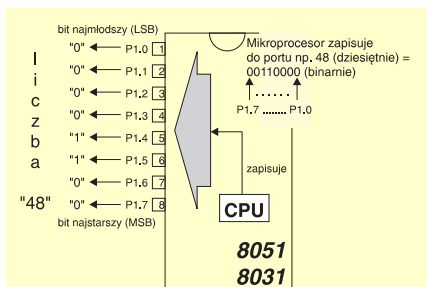
Ponieważ elektronika oparta na układach mikroprocesorowych wkracza pod strzechy coraz silniej, warto by przyrzeć się bliżej jednemu, bodaj najpopularniejszemu układowi tego typu, a mianowicie mikrokontrolerowi 8051.

Zapewne wielu Czytelników EdW spotyka się z oznaczeniem 8051. Niektórych z pewnością ogarnia zimny dreszcz, inni jak wynika z listów, są zaciekawieni tematem i możliwościami programowania mikroprocesorów. Postaram się w sposób przystępny, tak merytorycznie jak i finansowo, (niestety z nauką wiąże się część praktyczna, która wymaga minimum sprzętu do nauki programowania) pokazać i przekonać Was o tym że projektowanie układów przy wykorzystaniu mikrokontrolera 8051 nie jest trudne. Wymagane są jedynie podstawowe wiadomości z techniki cyfrowej, mówiąc konkretnie każdy, kto zna podstawowe bramki logiczne oraz najprostsze typy przerzutników, a jeżeli dodatkowo wykonał sam jakiś układ lub opisany w literaturze, z pewnością nie będzie miał problemów z opanowaniem sztuki korzystania z mikroprocesora 8051.

Przy opisie samego procesora, a w późniejszych numerach EdW, także podczas krótkich lekcji na temat programowania, będę za każdym razem odwoływał się do analogicznych układów wy-

konanych w standardowej technice cyfrowej (TTL czy CMOS). Dzięki temu każdy zainteresowany tematem czytelnik, zorientowany choć w podstawach cyfrowki, będzie w stanie strawić pewną porcję wiedzy, osławając się jednocześnie





Rys. 4. Zapis przykładowej liczby do portu P1.

nie z na pozór skomplikowanym układem cyfrowym, jakim jest 8051 ka.

Dla sceptyków, którzy sądzą, że programowanie procesora 8051, nawet na etapie "przedszkola", wymaga posiadania drogiego komputera klasy PC, mam miłą wiadomość. Otóż skonstruowałem podstawowy układ aplikacyjny na procesor 8051 (nie duża płytka drukowana + kilka podzespołów), dzięki któremu każda teoretyczna lekcja na łamach naszego pisma, będzie mogła być natychmiast powtórzona w praktyce na stole każdego z Was, drodzy Czytelnicy. Bynajmniej nie będzie potrzebny także żaden programator pamięci EPROM lub inne, często kosztowne wyposażenie. Wystarczą dobre chęci i trochę wolnego czasu, a z pewnością każdy z Was będzie zachwycony efektami swojej pracy, czyniąc pierwsze kroki w technice mikroprocesorowej.

Trochę o samym bohaterze

Na początek przyjrzyjmy się samemu mikroprocesorowi 8051. Warto w tym miejscu wyjąć z szuflady biurka taką kostkę, a jeżeli ktoś jej nie posiada, może ją nabyć prawie w każdym sklepie z podzespołami elektronicznymi, lub za pośred-

nictwem działu obsługi czytelników. Koszt zakupu 8051 w chwili obecnej waha się w granicach 2,00...4,00 nowych złotych, nie jest to więc dużo jak na kieszeń nawet nie zarabiającego amatora. W chwili obecnej na rynku znajduje się wersja mikroprocesora wykonana w technologii CMOS oznaczona jako 80C51.

Wszystkie parametry charakterystyczne (prądowe i napięciowe) podane w artykule będą odnosić się do tej wersji, aczkolwiek dla uproszczenia będziemy posługiwać się określeniem bez litery "C" mówiąc o typ układzie.

Mikrokontroler 8051 umieszczony jest w 40-nóżkowej obudowie (przeważnie plastikowej) typu DIL (skrót od "Dual In-Line Package", co po angielsku znaczy "obudowa dwurzędowa"). Sama ilość końcówek nie jest przerażająca, wszakże znamy inne układy np. ICL7106, które także umieszczone są w takiej obudowie.

W tym miejscu ktoś może powiedzieć: "No tak, ale opis typowej ICL ki (7106) można znaleźć prawie w każdym czasopiśmie lub podręczniku, znaczenie 40-tu wyprowadzeń też, a tu mam taki mikroprocesor, każdy mi mówi że to układ uniwersalny, a ja i tak nie wiem co mam z nim zrobić...".

Zapoznanie się z mikrokontrolerem rozpoczniemy od ogólnego poznania jego 40 wyprowadzeń, w końcu tylko to "wystaje" z obudowy i jest widoczne.

1. Końcówki o numerach 1...8 (port P1)

Są to wyprowadzenia 8-bitowego, uniwersalnego portu mikroprocesora oznaczanego w literaturze jako P1 (port nr 1 jak kto woli). Jeżeli słowo "port" nie

jest do końca jasne, posłużę się porównaniem (choć mało dokładnym) do układu typu rejestrowego o 8-miu wyprowadzeniach (8-bitach - stąd nazwa 8-bitowy). Czy pamiętamy układ serii TTL - 74198? Jest to coś w tym stylu, tylko że bardziej uniwersalne. Dla tych, którzy nie wiedzą, co to 74198, inne proste porównanie. Port, jak każdy port, pomaga w przyjmowaniu i wysyłaniu, tyle że nie towarów, lecz informacji.

Port może pełnić rolę wyjścia informacji binarnej (czyli że procesor może ustawiać stany logiczne na końcówkach tego portu). Tak więc, jeżeli zachodzi potrzeba, procesor może np. wpisać do portu P1 dowolną liczbę binarną z zakresu 0...255, np. 48. Binarne liczba 48 = 00110000₂. Oznaczenia poszczególnych końcówek portu P1 wskazują na kolejną pozycję bitu (cyfry liczby binarnej), co pokazuje rysunek 4.

Tak więc końcówka P1.7 (najstarsza) przyjmie poziom logiczny 0, końcówka P1.6 poziom 0, P1.5 poziom 1 itd.

Ale co daje "zapisanie jakiejś liczby do portu P1"? Otóż zastosowań może być wiele. Najprostsze z nich obrazuje **rysunek 5**. Do każdego wyprowadzenia portu P1 dołączono układ z przełącznikiem, którego styki

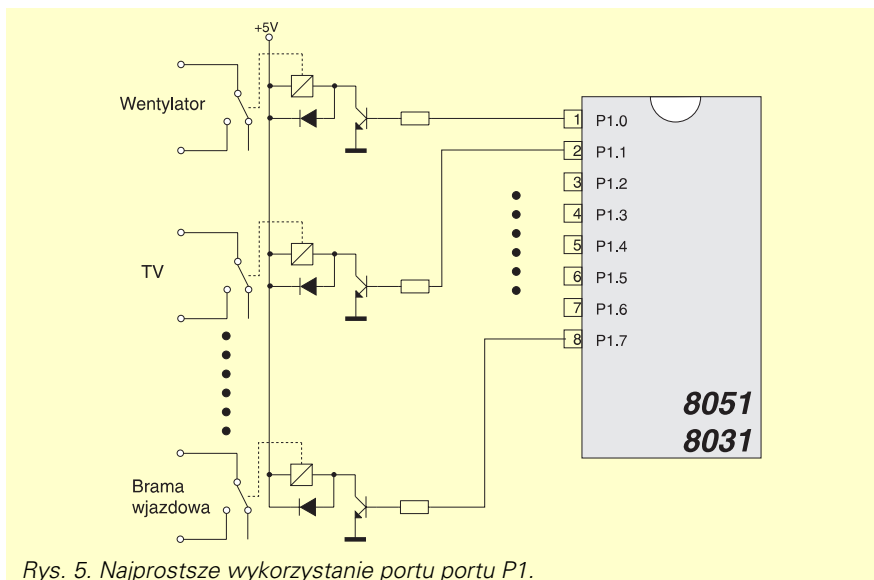
Końcówki dowolnego portu procesora mogą pełnić zarówno rolę wejść, jak wyjść.

złączają dowolne urządzenie elektryczne np. w mieszkaniu. W sumie na rysunku jest ich osiem,

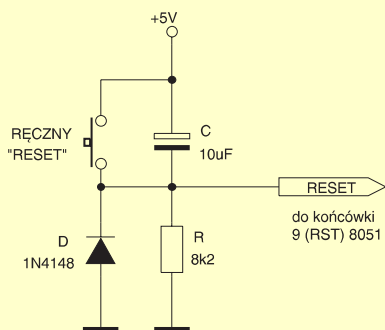
lecz w praktyce nie musimy korzystać ze wszystkich wyprowadzeń portu. Przy takim wykorzystaniu portu program zawarty w mikroprocesorze może na przykład włączać i wyłączać oświetlenie w mieszkaniu. Uzyskamy świetny symulator obecności domowników.

Projektując układ wykonawczy należy mieć na uwadze maksymalną obciążalność każdego z wyprowadzeń portu P1, z reguły wynosi ona 10mA (w obecnie oferowanych wersjach procesora) na każdy pin. Można zatemysterować za pomocą portu maksymalnie do czterech wejść TTL serii standard.

Istotną zaletą portów uniwersalnych procesora (w tym także P1) jest możliwość indywidualnego ustawiania poziomu logicznego na każdym wyprowadzeniu niezależnie. Nie trzeba zatem zapisywać całej liczby do portu aby np. zmienić stan tylko na jednym wyprowadzeniu, wystarczy ustawić (rozkazem zwanym SETB) lub wyzerować (rozkazem CLR) odpowiedni bit rejestru portu P1, toteż np. ustawienie pinu P1.5 na logiczne "0" nastąpi poprzez wydanie polecenia: CLR P1.5 ("clr" - clear, ang. zeruj, wyczyść). Niechcący zahaczyliśmy o programowanie, ale o tym będziemy mówić szczegółowo przy innej okazji.



Rys. 5. Najprostsze wykorzystanie portu P1.



Rys. 6. Podstawowy (użytkowy) układ resetowania 8051.

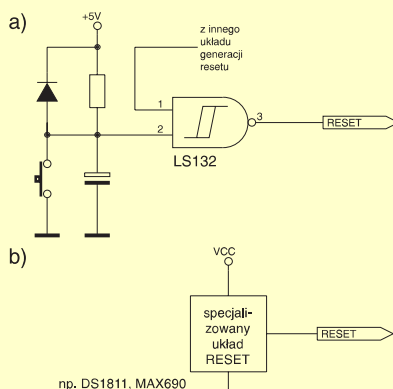
Port (cały lub niektóre z jego pinów), podobnie jak przy zapisie, można ustawić także jako wejście informacji logicznej. Każde z wyprowadzeń staje się wtedy wyjściem o wysokiej impedancji, dzięki temu dowolny poziom logiczny podany z wyjścia jakiegoś układu cyfrowego (np. z wyjścia bramki układu TTL lub CMOS) może być odczytany poprzez piny portu a informacja czy tym stanem była logiczna "1" czy "0", zostaje wykorzystana przez procesor dla dalszego jego działania w zależności od spełnianej akurat funkcji. Krótko mówiąc, procesor może odczytać stany logiczne, jakie z zewnątrz podano na końcówki portu.

Oczywiście poziomy logiczne napięć wejściowych portu P1 (oraz każdego innego) muszą zawierać się w przedziale napięć zasilania mikrokontrolera, czyli w zakresie 0...5V. Detekcja poziomów logicznych odbywa się jak dla bramek CMOS, stąd wartości progowe napięć tych stanów są zbliżone do połowy napięcia zasilającego. Istotną informacją jest fakt że w trybie "odczytu" z portu P1 końcówki są wewnętrznie podłączone (podciągane) do plusa zasilania poprzez wbudowane w 80C51 rezystory, co wymusza odczyt wysoki z portu w wypadku niepodłączenia końcówki portu.

2. Końcówka 9 (RST)

Z tematem mikroprocesorów czy mikrokontrolerów nierozłącznie wiąże się pojęcie "resetowania", czy jak kto woli "kasowania" układu. Czynność ta wykonywana poprzez podanie logicznej "1" na te wyprowadzenie na pewien okres czasu (jaki - o tym później) powoduje skasowanie układu, a więc natychmiastowe przerwanie wykonywanych czynności i rozpoczęcie działania procesora od samego początku (tak jakbyśmy włączyli zasilanie układu).

Czas trwania dodatniego impulsu kasującego zależy od częstotliwości z jaką



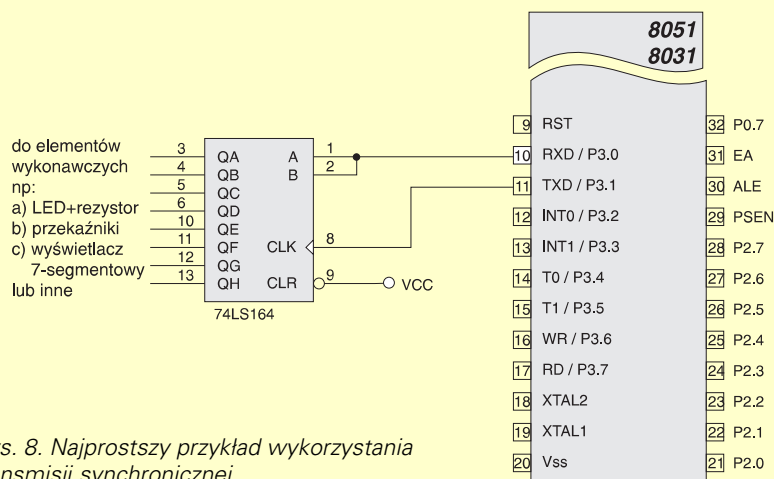
Rys. 7. Inne rozwiązania układów RESET

prace mikroprocesor. Wyjaśnię to dokładnie w dalszej części artykułu. Z reguły w typowych zastosowaniach czas 1ms w zupełności wystarcza.

W układach praktycznych do końcówki RST dołącza się mniej lub bardziej skomplikowany układ który generuje wymagany impuls zerujący najczęściej w trzech przypadkach:

- po włączeniu zasilania układu
- na nasze żądanie - poprzez np. przyciśnięcie klawisza (umieszczonego z reguły na płycie drukowanej tuż obok procesora).
- w sytuacjach awaryjnych, kiedy np. poprzez zakłócenie najczęściej na liniach zasilających nastąpi błędne działanie układu mikroprocesora, w żargonie często określa się to mianem "zawieszenia" lub "niekontrolowanej pracy" układu.

Trzeci przypadek dotyczy bardziej złożonych układów stosowanych szczególnie w automatyce i elektronice przemysłowej. My najczęściej spotkamy się z dwiema pierwszymi sytuacjami. Przykładowy układ zapewniający prawidłową skalowanie i ponowny start procesora 80C51 przedstawia **rysunek 6**. Głównym



Rys. 8. Najprostszy przykład wykorzystania transmisji synchronicznej.

elementem układu "resetu" jest kondensator elektrolityczny C. Z reguły jego wartość powinna wynosić 10...22μF. Jest on niezbędny do prawidłowej generacji impulsu resetu przez układy wewnętrzne mikrokontrolera.

W starszych wersjach 8051 wykonanych w technologii HMOS, niezbędny okazał się dodatkowy rezystor blokujący wejście RST do masy, co zapewniało wymuszenie stanu niskiego na tym pinie podczas normalnej pracy układu. W nowych katalogach opisujących układy w wersji CMOS, rezystor jest ten pomijany, aczkolwiek w praktycznych układach powinniśmy przewidzieć miejsce na płytce drukowanej, ze względu na różnorodność procesorów serii 8051.

Niech za przykład posłuży fakt, że producent najnowszych procesorów z rodziny 8051 w kartach katalogowych najnowszych wersji z wewnętrzna pamięcią EEPROM typu "Flash" zaleca stosowanie tego rezystora pomimo iż produkowane układy są wykonane w wersji CMOS. My możemy stosować rezystor o wartości 8,2...10k Ω .

Widoczny na rysunku 6 klawisz służy do resetowania procesora bez konieczności wyłączania napięcia zasilającego. Toteż w każdej chwili użytkownik może przerwać wykonywanie programu przez procesor.

Na **rysunku 7** pokazano inne, bardziej złożone wersje układów pełniących funkcje resetu, lecz w naszym przypadku w zupełności wystarczy wersja z rysunku 6.

3. Końcówki o numerach 10...17 (port P3)

Podobnie jak w przypadku portu P1, port P3 może pełnić wszystkie opisane wcześniej funkcje - może być wyjściem lub wejściem. Dodatkowe symbole na rysunku 3 tuż obok wyprowadzeń portu P3 sugerują że port może też spełniać inne dodatkowe funkcje. I tak też jest.

Piny P3.0 (RXD) i P3.1 (TXD) mogą pełnić rolę portu transmisji szeregowej. W praktyce poprzez te dwa wyprowadzenia można przesyłać informację (bajty i bity) z i do procesora z innych układów cyfrowych w sposób szeregowy, tzn. bit po bicie.

Ciekawostką niech będzie też fakt, że przesyłanie to może odbywać się na kilka sposobów:

- synchronicznie - wtedy pin P3.0 pełni rolę dwukierunkowej magistrali szeregowej, po której przesyłane są dane, zaś pin P3.1 generuje sygnał taktujący, pełniąc rolę zegara (podobnie jak w szeregowych rejestrach przesuwanych np. 74164, 74165). **Rysunek 8** obrazuje sposób transmisji synchronicznej do zewnętrznego 8-bitowego rejestru TTL typu 74164.

- asynchronicznie - kiedy z góry zadajemy prędkość transmisji pomiędzy naszym procesorem 8051 a innym, zewnętrznym układem np. łączem RS232c komputera PC. W takim przypadku końcówka P3.0 - RXD pełni rolę odbiornika przesyłanych szeregowo danych (pierwsza litera symbolu "R" oznacza receive - ang. odbiór), zaś końcówka P3.1 - TXD nadajnika ("T" - transmitt - ang. nadawanie).

Ponadto rozróżnia się kilka trybów pracy asynchronicznej.

Tych, którzy nie zrozumieli dokładnie dodatkowych funkcji wyprowadzeń RXD i TXD pocieszam, że temat ten wyjaśnię dokładnie w rozdziale na temat sposobów komunikacji szeregowej w jednym z kolejnych odcinków cyklu.

Alternatywną funkcją końcówek P3.2 (INT0) oraz P3.3 (INT1) jest funkcja detekcji przerwań zewnętrznych. Dla tych czytelników, którzy nie wiedzą, co to oznacza, wyjaśnię, że pojęcie przerwania w tym przypadku odnosi się do zmiany stanu logicznego (na omawianym wyprowadzeniu P3.2 lub P3.3) z "1" na "0". W efekcie "we wnętrzu" procesora 8051 zostaje ustawiona tak zwana flaga (nazywana także jako "znacznik zgłoszenia przerwania", co w odniesieniu do techniki cyfrowej można wyobrazić sobie jako przerzutnik). Konsekwencją tego jest automatyczne przerwanie wykonywania przez procesor programu i natychmiastowe przejście do wykonania czynności ściśle określonych przez programistę. Ciąg takich czynności nazywany jest w technice mikroprocesorowej: "procedurą obsługi przerwania". Najprostszą analogią do zasady działania dowolnego przerwania (także zewnętrznego typu INT0 lub INT1) jest np. sytuacja, kiedy sprzątam mieszkanie, czyli wykonujemy określone czynności, powiedzmy odkurzenie. Po tym mamy za zadanie sprzątnąć kurz z półek, a następnie umyć okna. W pewnej chwili rozlega się

gwizdek czajnika, więc oczywiście przerwamy wykonywanie - tu użyję sformułowania: "pętli głównej programu", którą jest sprzątanie pokoju - i szybko biegnie my wyłączyć gaz. Wykonaliśmy dwie dodatkowe czynności: biegniemy do kuchni i wyłączymy czajnik, czyli można powiedzieć, że wykonaliśmy "procedurę obsługi przerwania" (wyłączenia czajnika, jak kto woli). Wykrycie zmiany stanu logicznego na końcówkach przerwań zewnętrznych INT0 i INT1 wiąże się ze spełnieniem jednego warunku, a mianowicie, aby czas od wspomnianego ujemnego zbocza sygnału zgłoszenia przerwania do ponownego przejścia w stan wysoki był odpowiednio długi. Podobnie jak w przypadku warunku sygnału RST, czas ten zależy od częstotliwości zegara mikroprocesora.

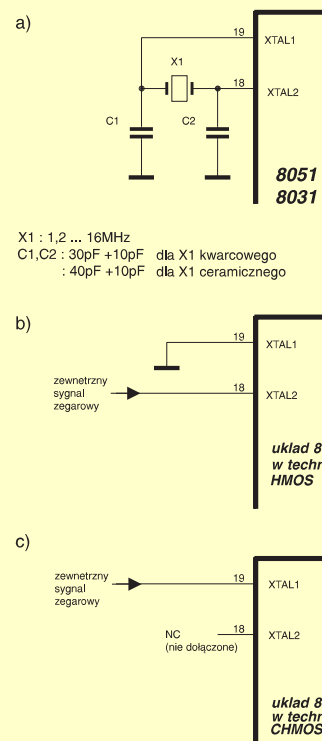
Dokładnie sposób działania systemu przerwań procesora 8051 przedstawię w kolejnych odcinkach.

Końcówki (P3.4 i P3.5) oznaczone na diagramie z rysunku 3 jako T0 i T1 pełnią dodatkową funkcję wejść uniwersalnych, programowalnych liczników, wbudowanych w strukturę 8051. Procesor 80C51 zawiera dwa bliźniacze liczniki T0 i T1 (oznaczenie takie same jak końcówek). Maksymalnie mogą one zliczać do $2^{16} = 65536$, po czym zostają wyzerowane. Liczniki te oprócz zliczania impulsów z wejść T0 i T1 mogą także zliczać impulsy wewnętrzne, pochodzące z generatora mikrokontrolera. W praktyce wykorzystywane jest to np. do odmierzania określonych odcinków czasu np. przy funkcji zegarka. Jak wspomniałem wcześniej, liczniki mogą być programowane przez użytkownika, a więc można np. zmniejszyć ich pojemność (do 2^8 lub 2^{13}), można także zapisać w nich wartość początkową, zatrzymać je w dowolnym momencie lub uruchomić. Szczegóły w rozdziale na temat układów licznikowych procesora 8051.

Pozostały do omówienia wyprowadzenia P3.6 i P3.7, oznaczone jako WR\ i RD\.

Jak pisaliśmy we wcześniejszych numerach EdW, prawie każdy mikrokontroler posiada możliwość współpracy z pamięcią zewnętrzną, którą przecież trzeba zaadresować. W pamięci tej można przechowywać istotne z punktu widzenia użytkownika dane, np. poziom temperatury z ostatnich dni półrocza (jeżeli procesor pracuje w układzie stacji meteorologicznej), lub inne w zależności od potrzeb.

Aby zapisać takie informacje w zewnętrznej pamięci danych potrzebne są oprócz podania adresu komórki pamięci do której ma nastąpić zapis, także sygnały sterujące zapisem lub w przypadku odczytywania - odczytem z pamięci.



Rys. 9. Typowe układy zewnętrznego oscylatora kwarcowego.

Właśnie pin WR\ jest sygnałem zapisu do zewnętrznej pamięci danych, a końcówka RD\ wysyła sygnał do odczytu. W praktycznych zastosowaniach jako elementy pamięci wykorzystuje się układy statycznych RAM - czyli w skrócie SRAM.

Procesor 8051 potrafi zaadresować maksymalnie 65536 (2^{16}) komórek pamięci (bajtów), ale o tym później.

I to tyle na temat alternatywnych funkcji portu P3, nie zapomnijmy jednak że port P3 (lub niektóre z jego pinów) może pełnić rolę zwykłego, uniwersalnego portu wejścia- wyjścia, podobnie jak P1.

4. Końcówki 18 i 19 (XTAL1 i XTAL2)

Końcówki te służą do dołączenia zewnętrznego rezonatora kwarcowego o częstotliwości zależnej od potrzeb użytkownika, ale także od wersji układu 8051.

W praktyce częstotliwość ta może wynosić od 1,2MHz do 12...16MHz, na rynku spotyka się także wersje procesorów pracujące przy wyższych częstotliwościach nawet do 40MHz, a także przy niskich - nawet do pojedynczych herców w wypadku procesorów 80C51 w wersji statycznej (np. 89C51 firmy Atmel).

Dołączony do tych pinów rezonator kwarcowy po uzupełnieniu o dodatkowe kondensatory o wartości z reguły z przedziału 22...40pF (w zależności od wartości rezonatora), umożliwiając pracę wbudo-

wanemu w 8051 generatorowi, który "napędza" cały mikroprocesor. Oczywiście od częstotliwości rezonatora ściśle zależy szybkość działania naszego mikrokontrolera. Typowy układ zewnętrznego oscylatora przedstawia rysunek 9a.

Częstotliwość, z jaką pracują wewnętrzne układy mikroprocesora, jest określona wzorem:

$$F = f_{\text{xtal}}/12,$$

gdzie f_{xtal} jest częstotliwością rezonatora kwarcowego.

Powodem takiego podziału częstotliwości rezonatora jest wewnętrzna architektura wszystkich procesorów serii 8051. Wiąże się z tym pojęcie "cykli maszynowych procesora" o których znaczeniu napiszę w rozdziale na temat oscylatora 8051.

W każdym razie z praktycznego punktu widzenia, przedstawiony na rysunku 9 układ, podobnie jak układ resetu z rys. 6 jest niejako obowiązkowym (przynajmniej na etapie nauki programowania).

Końcówka XTAL1 (pin 19) w układach w wersji CMOS może także pełnić rolę wejścia zewnętrznego sygnału zegarowego o częstotliwości w zakresie, jak opisano w przypadku stosowania rezonatora kwarcowego. Wtedy rezonator i dodatkowe kondensatory są zbędne.

W przypadku gdy mamy do czynienia z wersją w technologii HMOS wejściem takiego sygnału jest XTAL2 (pin 18). W obu przypadkach pozostały pin powinien być nie podłączony. Dokładnie sytuację tę wyjaśnia rys. 9b i c.

5. Końcówka 20 (Vss)

Podobnie jak w większości układów cyfrowych ostatnie wyprowadzenie w "dolnym rzędzie" obudowy jest końcówką ujemnego napięcia zasilającego - masy (GND). W przypadku układów CMOS podaje się oznaczenie Vss co

oznacza biegun ujemny napięcia zasilającego. W naszych zastosowaniach będziemy dołączać ten pin do masy przysługującego układu elektronicznego.

6. Końcówki o numerach 21...28 (port P2)

Są to wyprowadzenia drugiego 8-bitowego portu procesora. Port P2 spełnia wszystkie funkcje podobnie jak P1. Dodatkowo poprzez końcówki portu P2 podawana jest w razie potrzeby starsza część adresu (A8...A15) przy dostępie do zewnętrznej pamięci danych (SRAM) a także programu (np. EPROM). Sposób w jaki to się odbywa opiszemy przy okazji "dołączania pamięci zewnętrznej do mikrokontrolera 8051".

7. Końcówka 29 (PSEN)

W przypadku pracy procesora z zewnętrzną pamięcią programu (np. EPROM) końcówka ta wysyła sygnał odczytu z tej pamięci. W praktyce jest ona dołączona do wejścia OE\ współpracującej pamięci EPROM. Procesor chcąc odczytać kolejny rozkaz (polecenie do wykonania) z zewnętrznej pamięci programu podaje poziom niski na końcówkę "PSEN" a następnie dokonuje odczytu.

Dzieje się tak w ściśle określonych warunkach, synchronicznie z częstotliwością zegara procesora. Jeżeli posiadamy wersję procesora z wewnętrzną pamięcią (typu EPROM lub EEPROM), i wykorzystujemy pracę z tą wewnętrzną pamięcią, końcówka 29 powinna pozostać niepodłączona.

8. Końcówka 30 (ALE)

O końcówce 30 powiemy przy okazji omawiania portu P0.

9. Końcówka 31 (EA\)

Powinna być dołączona do masy, jeżeli mikroprocesor pobiera rozkazy z wewnętrznej pamięci programu (patrz

pkt.7), lub do plusa zasilania (+5V) jeżeli z wewnętrznej.

W pewnych układach procesor pomimo że posiada wewnętrzną pamięć programu, ze względu na zbyt małą jej pojemność, musi sięgać do zewnętrznej pamięci. W takim przypadku pin EA\ powinien być dołączony do plusa zasilania, tak aby procesor po jego "resecie" mógł rozpocząć pracę pobierając rozkazy z wbudowanej pamięci programu. Należy także pamiętać że dołączenie EA\ do masy blokuje wewnętrzną pamięć programu jeżeli ona istnieje.

W praktyce jest to często stosowany chwyt, kiedy kupujemy w sklepie na ogół kilkakrotnie tańszą wersję procesora z pamięcią wewnętrzną typu ROM. W pamięci takiej najczęściej zapisany jest pewien program lecz, z naszego punktu widzenia jest on zupełnie bezużyteczny. Toteż aby w pełni wykorzystać walory mikroprocesora (oczywiście przy pracy z zewnętrzną pamięcią programu) bez uruchamiania nieznanego nam programu, blokujemy pamięć ROM poprzez zwarcie EA\ do masy.

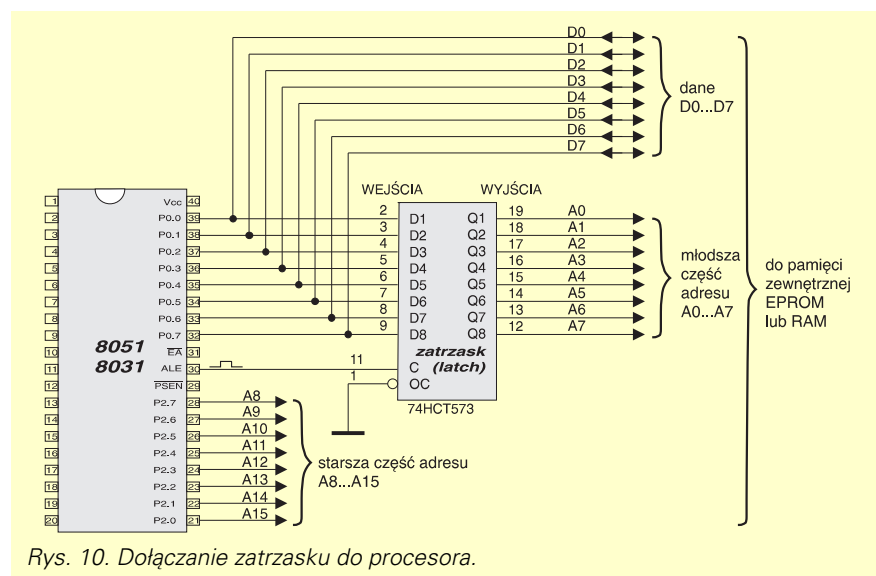
10. Końcówki o numerach 32...39 (port P0)

Trzecim i ostatnim uniwersalnym portem procesora 8051 jest P0. Podstawowe funkcje portu jako dwukierunkowej bramy do wymiany danych są takie same jak w przypadku portów P1 i P2. Zasadniczą różnicą jest jednak zwiększona obciążalność (do 8 wejść TTL) tego portu oraz fakt nie posiadania wbudowanych rezystorów podciągających końcówki portu do plusa zasilania w wypadku odczytu.

Dlatego przy projektowaniu dowolnych układów wyjściowych dołączanych do tego portu należy uwzględnić wspomniane właściwości tak, aby np. odpowiednio spolaryzować bazy tranzystorów z rys. 5.

Drugą bardzo ważną rolę, jaką pełni P0, jest funkcja multiplexowanej magistrali danych (8-bitów: D7...D0) i młodszej części adresu (A7...A0). Multiplexowanej w praktyce znaczy "przełączalnej", czyli raz na końcówkach portu P0 procesor może wystawić bajt danych (np. do zapisu do zewnętrznej pamięci danych), w innym przypadku adres, w celu wybrania potrzebnej komórki z pamięci SRAM, do której ma być zapisana.

Bardziej wnikliwy czytelnik zauważy, że przecież do zapisu danej w zewnętrznej pamięci SRAM potrzeba w sumie 16 sygnałów adresu (A0...A15) oraz 8-bitów (sygnałów) danej. Potrzebne są zatem 3 ośmiobitowe porty (2 na adres i jeden na daną), a my mamy do dyspozycji tylko dwa P2 i P0.



Rys. 10. Dołączanie zatrasku do procesora.

Tabela 2.

Parametr	Symbol	Wartość dopuszczalna	Wartość zalecana	Uwagi
napięcie zasilania	Ucc-Uss	6,6V	5V ± 20%	
napięcie wzgl. masy na dolnej końcówce układu	-1,0...7,0V			w zakresie napięcia zasilania
moc rozpraszana	Ptot	1W	nie dotyczy	
temperatura pracy		0°C...70°C	0°C...70°C	dla wersji specjalnych układu zakres pracy może być większy
temp. przechowywania		-65°C...+150°C	nie dotyczy	

Tabela 3.

Oznaczenie	Wbudowana pamięć programu	Uwagi
80C31	bez pamięci	układ nadaje się do pracy po dołączeniu zewnętrznej pamięci EPROM wraz z niezbędnym zatraskiem (np. 74373/573)
80C51	4kB ROM	wersja, która wymaga zablokowania pamięci, patrz opis pkt. 9 (reszta jak dla 80C31)
87C51	4kB EPROM lub EPROM OTP*	procesor z wbudowaną pamięcią typu EPROM i możliwością kasowania jej promieniami UV poprzez okienko kwarcowe
89C51	4kB EEPROM (Flash)	podobnie jak 87C51 z tym że pamięć programu można skasować drogą elektryczną przez podanie impulsu - dlatego mówi się o pamięci typu "Flash" (ang. błysk).

* OTP (One Time Programmable) - pamięć EPROM zapisywalna jednokrotnie (kostka nie ma okienka kwarcowego umożliwiającego skasowanie zawartości pamięci na pomocą promieniowania ultrafioletowego).

I tu właśnie leży zasada multipleksowania (naprzemiennego wystawiania adresu lub danej) procesora 8051. Otóż sygnał - informacja o tym, co aktualnie znajduje się na szynie portu P0, pojawia się na wyprowadzeniu 30 oznaczonym jako ALE. Sygnał ten można nazwać "sygnałem zapisu adresu" do dodatkowego zewnętrznego układu cyfrowego. Układ ten jest 8-krotnym zatraskiem aktywowanym wysokim poziomem logicznym. W serii TTL znajdują się dwie kostki spełniające rolę układu zatraskiwania młodszej części adresu przez 8051, są to 74373 lub 74573.

Różnica między nimi polega jedynie na innym wyprowadzeniu końcówek, reszta działa tak samo. **Rysunek 10** pokazuje sposób dołączenia zatrasku do procesora 8051. W momencie kiedy 8051 wystawi na port P0 młodszą część adresu (A7...A0), daje temu sygnał, zmieniając stan na końcówce ALE z niskiego na wysoki. W efekcie po nadejściu tym razem opadającego zbocza sygnału

ALE, dana (adres) z portu P0 zostaje zapisana w zatrasku 74373 (573). Teraz na ośmiu jego wyjściach adres będzie utrzymywany niezależnie od zmieniających się stanów w porcie P0 aż do nadejścia następnego sygnału z końcówki ALE. Skoro procesor posługując się dodatkowym układem "zapisal" na zewnątrz adres, może teraz śmiało wystawić na port P0 daną, która ma być zapisana w zewnętrznej pamięci danych. Oczywiście można też odczytać dane z pamięci.

Tak więc podsumowując, przeanalizowaliśmy sposób w jaki za pomocą jednego sygnału ALE procesor 8051 może niejako rozszerzyć liczbę linii adresowych z 8 do 16.

W przypadku niekorzystania z możliwości obsługi zewnętrznej pamięci tak programu (EPROM) lub danych (SRAM) końcówka ALE (30) jest nieodłączona.

W odcinku poświęconym rozbudowie systemu opartego na '51-ce powrócimy do tego tematu, na razie istotne są informacje ogólne.

11. Końcówka 40 (Vcc)

Oczywiście jest to końcówka zasilania mikroprocesora 8051. Napięcie względem końcówki Vss (czyli masy) z reguły nie może przekroczyć 6,5V. Dlatego układ mikrokontrolera należy zasilac napięciem $5V \pm 0,25V$ używając do tego celu dowolnego zasilacza stabilizowanego najlepiej przy pomocy znanego układu 7805.

Zasadą przy projektowaniu układów z 8051 jest blokowanie tego wyprowadzenia kondensatorem o wartości 100nF do masy układu cyfrowego. Praktycznie na płycie drukowanej należy zawsze przewidzieć miejsce na taki kondensator umieszczając go jak najbliżej samego układu procesora lub po prostu przylutowując go od strony wyprowadzeń na płycie drukowanej.

W tabeli 2 przedstawiono parametry dopuszczalne oraz zalecane przez producentów procesora 80C51 oraz pochodnych produkowanych w wersjach CMOS.

Na koniec pozostaje jeszcze krótkie wyjaśnienie oznaczenia samego kontrolera i kryjących się w nim dodatkowych istotnych dla nas informacji. Problem ten dokładnie przedstawia **tabela 3**.

Sławomir Surowiński