

W poprzednim odcinku zapoznałeś się, drogi Czytelniku ze znaczeniem poszczególnych wyprowadzeń mikrokontrolera 8051. Ze względu na chęć czysto praktycznej nauki „użytkowania” tego układu, nie opisywałem dokładnie wszystkich funkcji każdej z „nózek”, a jedynie krótko zaznajomiłem Cię z przedstawionym w EdW 5/97 tematem. Być może nie wszystkie pojęcia są dla Ciebie od razu oczywiste, lecz nie powinieneś się tym przejmować, na tym etapie poznawania mikroprocesora wszystko co Ci potrzeba to „osłuchanie się” z typowymi hasłami na temat naszego bohatera. Na późniejszym etapie – praktycznej nauki z wykorzystaniem układu elektronicznego, wiedza ta przyda Ci się z pewnością, szczególnie że wtedy zaczniemy wspólnie wchodzić w temat 8051 coraz głębiej.

W tym odcinku kolejna porcja podstawowych informacji które pozwolą Ci na oswojenie się z naszym 8051!

Pamięć mikroprocesowa

Jak zapewne pamiętasz, w pierwszym odcinku naszego cyklu mówiąc o budowie „rasowego” mikrokontrolera jednoukładowego, wspominaliśmy o drugim ważnym elemencie jego architektury, a mianowicie – pamięci. Ten odcinek naszego cyklu zostanie poświęcony właśnie jej.

Prawdopodobnie spotkałeś się wcześniej z pojęciami pamięci RAM, ROM, EPROM, EEPROM itd. Wszystkie one odnoszą się do cyfrowych układów scalonych w strukturze których możliwe jest zapisanie i przechowanie informacji. Od wielu lat na rynku elektronicznym znajduje się wiele takich układów, różniących się typem, pojemnością pamięci, technologią wykonania, wszystko to bardzo często narzuca sposób ich wykorzystywania w konkretnych rozwiązaniach układowych.

Zaczniemy od krótkiej powtórki dotyczącej samych pamięci i sposobu przechowywania w nich informacji, i tak:

- podstawową jednostką przechowania informacji w cyfrowych pamięciach jest bit;
- bit może przyjmować jedną z dwóch wartości: logiczne 0 lub 1;
- podobnie jak np. w układzie metrycznym, gdzie w celu uproszczenia pomiarów wprowadzono jednostki pochodne odległości (metr = 100cm = 1000mm), tak w przypadku jednostek informacji wprowadzono bajt, który jest równy 8 bitom.
- w odróżnieniu do typowych dziesiętnych systemów liczenia, przy omawianiu rozmiarów jak i odwoływania się do pamięci – stosuje się szesnastkowy (heksadecymalny) zapis liczb;

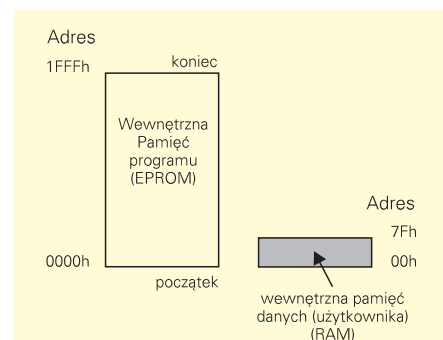


- wszystkie interesujące nas pamięci będą miały architekturę 8-bitową – czyli bajtową bowiem nasz bohater – 8051 jest mikrokomputerem 8-bitowym.
- podczas omawiania mikroprocesora, często przy okazji pamięci zamiast słowa „komórka pamięci” będziemy używać zamiennie słowa „rejestr”. Praktycznie każdy rejestr w 8051 możemy traktować jako oddzielną, posiadającą swoje miejsce (adres) komórkę w przestrzeni jego pamięci danych.
- każdy rejestr w 8051 jest 8-bitowy, niektóre z nich tworzą pary, dlatego czasami będziemy mówić o 16-bitowych rejestrach ($2 \times 8 \text{ bitów} = 16 \text{ bitów} = 2 \text{ bajty}$);
- powinieneś wiedzieć że w prawie każdym wskazanym (zaadresowanym) rejestrze (jak w komórce pamięci) możesz zapisać dowolną liczbę 8-bitową, lub odczytać wskazany (zaadresowany) rejestr;
- traktuj więc rejestr jako miejsce zapisu lub odczytu 8 bitów (bajtu) informacji, tak jak to ma miejsce w komórce 8-bitowej pamięci (dla maniaków cyfrówki z serii TTL rada – możesz sobie wyobrazić rejestr fizycznie jako trochę zmodyfikowany np. 74198 lub 74373). Wracajmy jednak do tematu.

Z pierwszego odcinka wiesz już że 8051 posiada 2 rodzaje pamięci. Pierwsza służy do przechowywania instrukcji programu, który ma być wykonany po

włączeniu zasilania układu. W drugiej pamięci znajdują się zmienne (tak jak w równaniach matematycznych) przechowujące określone dane i wyniki obliczeń. W 8051 dodatkowo w wydzielonej części tej drugiej pamięci znajdują się także specjalne komórki zwane rejestrami. W słownictwie związanym z 8051 używa się pojęcia SFR – z angielskiego „Special Function Registers” – rejestry specjalnego przeznaczenia („funkcji specjalnych”, jak kto woli). Tego zwrotu będziemy w przyszłości używać bardzo często, warto więc abyś sobie go zapamiętał.

Rysunek 1 przedstawia poglądową mapę pamięci zawartą w mikroprocesorze 87C51 (8751). Już wiesz że ten typ '51-ki charakteryzuje się 4 kB (kilobajtami) wewnętrznej pamięci stałej do przechowywania programu typu EPROM. Do-



Rys. 1. Organizacja pamięci wewnętrznej w mikrokontrolerze 8751.

Też to potrafisz

datkowo układ ten (podobnie jak wszystkie inne '51-ki) zawiera w swojej strukturze 128 B (bajtów) pamięci danych RAM.

Tak więc masz do dyspozycji 4kB = 4096 bajtów pamięci stałej EPROM – wszystkie komórki są zawsze numerowane (adresowane) jak wspomniano wcześniej w kodzie heksadecymalnym – tworzącej przestrzeń adresową o adresach: 0 – 4095 (dziesiętnie) lub 0000h – 1FFFh (heksadecymalnie). Dalej będziemy posługiwać się tylko tym drugim sposobem zapisu.

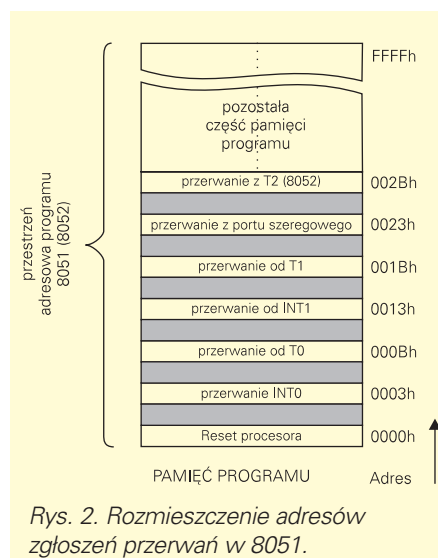
Wewnętrzna pamięć danych zajmuje adresy: 00h – 7Fh (0 – 127 dziesiętnie). Musisz w tym miejscu wiedzieć że pomimo, że adresy komórek pamięci RAM pokrywają się z częścią adresów pamięci programu, fizycznie w układzie nie występuje żaden konflikt, bowiem dostęp do omawianych obydwu rodzaj pamięci jest zupełnie inny. Mikroprocesor korzysta z innych poleceń w przypadku czytania lub zapisu do wewnętrznych 128 bajtów pamięci RAM, inne rozkazy służą do obsługi pamięci programu.

Poniżej pokrótce opiszę oba rodzaje pamięci i ich znaczenie w pracy mikrokontrolera.

Wewnętrzna pamięć programu

Program napisany przez użytkownika, dedykowany konkretnemu zastosowaniu 8051-ki powinien zostać umieszczony wewnątrz mikrokontrolera – czyli w wewnętrznej pamięci programu. Jak powiedziano wcześniej pamięć ta służy mikrokontrolerowi wyłącznie do odczytu rozkazów programu. W pamięci tej mogą być umieszczone także argumenty bezpośrednie rozkazów oraz tablice ze stałymi potrzebnymi do pewnych działań programu, np. tablica sinusów, tablica czasów zachodu słońca, lub cokolwiek innego. Mikroprocesor 8051 ma możliwość późniejszego pobrania ze swojej pamięci programu takiej stałej i wykorzystania jej np. w obliczeniach. Stała i tablice wprowadzane są przez programistę na etapie tworzenia programu, ale o tym innym razem.

Jeżeli program został przez nas utworzony a następnie zapisany w pamięci programu (o tym jak to się robi będzie mowa dalej), mikrokontroler jest gotowy do działania. Otóż po włączeniu zasilania dzięki obwodowi „Reset” (cz. I artykułu), wyzerowane zostają prawie wszystkie wewnętrzne układy mikroprocesora w tym także uwaga: „licznik rozkazów”. Ten ostatni służy mikroprocesorowi do kolejnego pobierania rozkazów z pamięci programu, a dokładnie do adresowania (czyli wskazywania) gdzie w przestrzeni adresowej pamięci programu znajduje się kolejna komenda. Jak się możesz domyś-



Rys. 2. Rozmieszczenie adresów zgłoszeń przerwań w 8051.

łać jego początkowa wartość wynosi 0 (zero), toteż pierwszym rozkazem pobranym z tej pamięci będzie ten umieszczony pod adresem 0000h.

Licznik rozkazów oznaczany jest w skrócie jako PC z angielskiego „Program Counter” – licznik programu (rozkazów) – warto o tym pamiętać. Licznik PC ma długość 16 bitów, czyli maksymalnie może liczyć do 65535 włącznie, po czym zostaje wyzerowany. Stąd wynika m.in. maksymalna wielkość pamięci programu z jakiej procesor może korzystać a mianowicie 64kB (65536 bajtów). Tak dużą pamięć posiadają niektóre mutacje '51-ki, ale prawie każdy z mikroprocesorów może współpracować z tak dużą pamięcią zewnętrzną.

W trakcie pobierania i wykonywania przez mikrokontroler kolejnych instrukcji licznik PC zmienia swoją wartość zawsze wskazując na aktualny adres kolejnego rozkazu w pamięci programu. Nasuwa się prosty wniosek, że maksymalną wartość jaką może osiągnąć licznik w naszym przypadku będzie 4095 – bowiem w naszym przykładzie z kostką 87C51 mamy do dyspozycji 4kB pamięci programu. O tym co się stanie po przekroczeniu tej wartości powiem później.

Na początek warto też wiedzieć, że oprócz wspomnianego miejsca „startowego” programu – czyli adresu 0000h (zero), w przestrzeni adresowej pamięci programu istnieje kilka innych istotnych dla programisty miejsc. Czy pamiętasz potoczne objaśnienie pojęcia „przerwanie”, pisaliśmy o tym w EdW 4/97?, jeśli nie to radzę sobie to przypomnieć. Otóż wyobraź sobie, że nasz mikroprocesor wykonuje określony program pobierając kolejne instrukcje z pamięci programu, która to jest adresowana poprzez licznik rozkazów PC. Wtem nadchodzi „przerwanie” – mikroprocesor w zależności co było jego źródłem powinien wykonać od-

powiednią dla niego procedurę obsługi (przyjęcia) przerwania.

W celu ujednolicenia systemu przerwań procesora w pamięci programu określono odpowiednie miejsca – adresy od których rozpoczyna się wykonywanie określonych procedur obsługi przerwań. W podstawowej rodzinie '51 są to adresy: 3, 11, 19, 27, 35 i 43 (03h, 0Bh, 13h, 1Bh, 23h, 2Bh szesnastkowo). Każdy z tych adresów określa początek wykonania innej procedury obsługi przerwania, dla 8051 są one następujące:

- 0003h** – przerwanie zewnętrzne z wejścia (końcówki) INT0 (pin 12)
- 000Bh** – przerw. wynikłe z przepełnienia pierwszego wewnętrznego licznika T0 procesora
- 0013h** – przerwanie zewnętrzne z wejścia (końcówki) INT1 (pin 13)
- 001Bh** – przerw. wynikłe z przepełnienia drugiego wewnętrznego licznika T1 procesora
- 0023h** – przerwanie wynikłe z odebrania lub zakończenia wysyłania danej poprzez wewnętrzny port szeregowy mikroprocesora

Dodatkowo w układach 8052, 8032 (8752) występuje:

- 002Bh** – przerw. wynikłe z przepełnienia trzeciego wewnętrznego licznika T2.

Na rysunku 2 zilustrowano rozmieszczenie w/w adresów zgłoszenia przerwań.

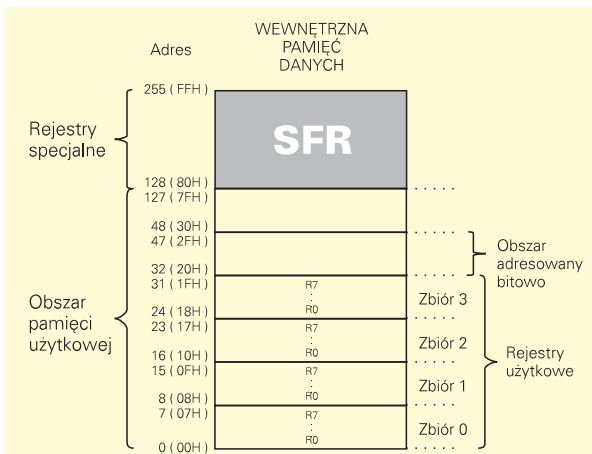
Praktycznie wygląda to tak, że w momencie zgłoszenia któregoś z wymienionych przerwań, automatycznie zachowana zostaje aktualna wartość licznika PC, a następnie zostaje wpisana do niego wartość odpowiednia do rodzaju przerwania jak opisano wyżej. Czyli np. jeżeli wewnętrzny licznik procesora T1 został przepełniony, do PC zostaje wpisana wartość 001Bh, po czym mikroprocesor rozpoczyna wykonywanie programu od tego adresu w pamięci programu. Po zakończeniu wykonywania czynności związanych z przepełnieniem T1, licznik rozkazów PC przyjmie ponownie wartość jak z przed nadejścia przerwania i program „potoczy się” dalej.

Dokładne objaśnienie działania systemu przerwań omówię przy innej okazji, na razie istotne jest abyś wiedział o istnieniu adresów specjalnych w pamięci programu procesora 8051.

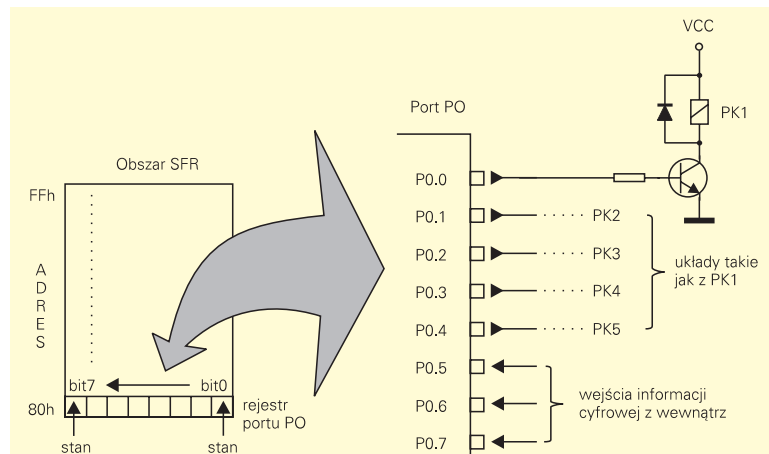
Wewnętrzna pamięć danych

W mikrokontrolerze pamięć ta przeznaczona jest dla użytkownika do przechowywania argumentów wartości zmiennych oraz wyników obliczeń arytmetyczno – logicznych.

W zależności od typu mikrokontrolera pamięć ta ma pojemność 128 lub 256 bajtów.



Rys. 3. Wewnętrzna pamięć danych w mikrokontrolerze 8051.



Rys. 4.

Dla 8051 wynosi ona 128 B (8052 – 256 B). Na rysunku 3 przedstawiono organizację wewnętrznej pamięci danych.

W przestrzeni tej pamięci można wyróżnić kilka obszarów. Dwa główne, wspomniane wcześniej to obszar pamięci użytkowej, oraz obszar rejestrów specjalnych SFR. Pamięć użytkowa zajmuje 128 komórek, adresy: 0 – 127 (00h – 7Fh), natomiast obszar SFR obejmuje adresy 128 – 255 (80h – FFh), z tym że nie wszystkie są wykorzystane przez rejestry specjalne.

I chociaż pamięć użytkownika podzielona jest na obszary, do których dostęp może odbywać się przez tzw. indeksowa-

Tabela 1. Rejestry specjalne mikrokontrolera 8051.

| Adres | Symbol | Nazwa |
|-------|--------|---|
| E0h | ACC | Akumulator |
| F0h | B | Rejestr B |
| D0h | PSW | Słowo stanu programu |
| 81h | SP | 8-bitowy wskaźnik stosu |
| 83h | DPH | 8-bitowy wskaźnik danych |
| 82h | DPL | 8-bitowy wskaźnik danych |
| 80h | P0 | Port 0 |
| 90h | P1 | Port 1 |
| A0h | P2 | Port 2 |
| B0h | P3 | Port 3 |
| B8h | IP | Rejestr sterujący priorytetem przerwań |
| A8h | IE | Rejestr kontrolny sterujący pracą systemu przerwań |
| 88h | TCON | Rejestr kontrolny pracy liczników T0 i T1 oraz przerwań INT0 i INT1 |
| 89h | TMOD | Rejestr sterujący trybem pracy liczników T0 i T1 |
| 8Ch | TH0 | 16-bitowy licznik T0 |
| 8Ah | TL0 | 8-bitowy licznik T0 |
| 8Dh | TH1 | 16-bitowy licznik T1 |
| 8Bh | TL1 | 8-bitowy licznik T1 |
| C8h | T2CON | Rejestr sterujący licznikiem T2 (w 8052) |
| CDh | TH2 | 16-bitowy licznik T2 |
| CCh | TL2 | 8-bitowy licznik T2 |
| CBh | RLDH | Słowo ładowane do licznika T2 |
| CAh | RLDL | Słowo ładowane do licznika T2 |
| 98h | SCON | Rejestr sterujący portem szeregowym |
| 99h | SBUF | Bufor portu szeregowego |
| 87h | PCON | Rejestr sterujący zasilania |

nie obszaru, to użytkownik może adresować ją poprzez proste adresowanie.

W pamięci użytkowej komórki o adresach 0...7, 8...15, 16...23 i 24...31 tworzą cztery zbiory uniwersalnych rejestrów roboczych. Każdy z rejestrów oznacza się symbolami R0...R7. W danej chwili użytkownik ma możliwość dostępu (poprzez nazwy R0...R7) tylko do jednego „banku” (zbioru) rejestrów roboczych. Przełączanie zbiorów odbywa się poprzez odpowiednie ustawienie dwubitowego wskaźnika zwanego jako RS – z angielskiego „Register bank Switch”. O szczegółach powiemy przy okazji omawiania znaczenia poszczególnych rejestrów specjalnych SFR.

Rejestry R0 i R1 z aktywnego banku pełnią rolę wskaźników danych do pośredniego adresowania wewnętrznej pamięci danych jak i zewnętrznej. W przypadku adresowania pamięci wewnętrznej można adresować cały obszar 8051 czyli adresy 0...7Fh. Sposoby adresowania pamięci przedstawimy przy okazji „pierwszych kroków w asemblerze”.

Na rys. 3 górna część przestrzeni adresowej: 80h...FFh zajmują SFR. W tabeli 1 opisano symbole oraz nazwę każdego z nich. W tym miejscu warto zapamiętać iż rejestry specjalne stanowią niejako sprzętowy „pomost” komunikacyjny pomiędzy programistą a wszystkimi blokami funkcjonalnymi mikrokontrolera. Dla przykładu, aby „dobrać się” i odpowiednio ustawić wewnętrzny licznik T1, należy odpowiednio zmodyfikować zawartość rejestru TMOD – (rejestr trybu liczników T0 i T1) oraz TCON (rejestr sterujący licznikami oraz zgłaszaniem przerwań zewnętrznych INT0 i 1).

W przestrzeni adresowej SFR znajdują się także rejestry będące jednocześnie portami wejścia-wyjścia, tymi o których mówiliśmy w poprzednim odcinku artykułu. Dzięki temu możliwy jest łatwy i szybki dostęp do dowolnych bitów por-

tu czyli fizycznie do jego wyprowadzeń. Zapis do odpowiedniego rejestru portu spowoduje pojawienie się kombinacji na końcówkach mikrokontrolera, odczyt rejestru pozwoli użytkownikowi na zbadanie poziomu logicznego na wybranej linii portu.

Jak widać z tabeli nie wszystkie 128 adresów z przestrzeni SFR jest wykorzystanych. „Puste” adresy nie nadają się do wykorzystania przez użytkownika. Nie jest to bynajmniej marnotrawienie cennych bajtów pamięci, lecz czysta przezorność projektantów rodziny 8051, którzy konstruując rozszerzone wersje poczwójnej ’51-ki wyposażają je w nowe dodatkowo bloki funkcjonalne, a w wolnych miejscach przestrzeni SFR umieszczane są dodatkowe rejestry sterujące ich pracą (wspomniane „pomosty”).

I tak np. w mikrokontrolerze 8052 umieszczono dodatkowy licznik T2, do sterowania którego niezbędne stało się zaimplementowanie w strukturze SFR rejestrów T2CON, TH2, TL2, RLDH i RLDL – patrz tabela 1.

W tym miejscu widoczny jest geniusz architektury jednoukładowców z rodziny ’51. Otóż producenci wytwarzając nowe mutacje tych procesorów, nie muszą się martwić o kompatybilność programową, czy architekturę dostępu do poszczególnych bloków układu. W każdym przypadku dodatkowe rejestry specjalne sterujące ich pracą umieszczane są w tej samej przestrzeni SFR, w taki sam sposób dostępnej dla użytkownika. Czyli jeżeli np. któryś z producentów zechce umieścić w strukturze 8051 8-bitowy przetwornik analogowo – cyfrowy, to prawdopodobnie w wolnych miejscach obszaru SFR umieści dodatkowe rejestry: a) rejestr sterujący pracą przetwornika, oraz b) rejestr danych z przetwornika, prawda że proste, no przynajmniej z naszego punktu widzenia.

Dla Ciebie drogi Czytelniku stąd wychodzi ogromna korzyść. Jeżeli poznasz podstawowy układ – 8051, to w przyszłości biorąc do ręki jedną z kilkudziesięciu mutacji tej rodziny, nie będziesz musiał niczego uczyć się od nowa, wystarczy że przeczytasz kartę katalogową dotyczącą: po pierwsze SFR w danym egzemplarzu procesora, oraz sposobu obsługi dodatkowego bloku (np. przetwornika A/C, pamięci EEPROM, lub modułu PWM) , a aplikacja układu nie zajmie Ci więcej niż zwykłej 8051–ki.

Na koniec tej części warto powiedzieć o dodatkowej możliwości tzw. adresowania bitowego wewnętrznej pamięci danych (w tym także SFR).

Otóż zwykle odwołania do konkretnego rejestru (komórki) pamięci odbywa się poprzez zapisanie lub odczytanie całego bajtu –

czyli 8 bitów. Czasem jednak bardziej praktyczne jest zbadanie tylko wybranych bitów z danego rejestru. Najlepszym przykładem niech będzie sytuacja kiedy do końcówek portu P0 mamy dołączonych 5 wyjść sterujących przekaźnikami, pozostałe 3 linie wykorzystywane są jako wejścia. Sytuację tę ilustruje **rysunek 4** (patrz str. 43).

Jeżeli chcemy np. zmienić stan przekaźnika nr 2, nie trzeba wpisywać całego 8-bitowego słowa do rejestru P0, wystarczy zmienić tylko pojedynczy bit – prawda że prostsze.

Nie wszystkie rejestry specjalne można adresować w sposób bitowy, możesz być jednak pewien Czytelniku, że te z nich które nie posiadają tej cechy, po prostu nie wymagają takiego sposobu obsługi ze strony programisty.

Adresowanie poszczególnych bitów może mieć także miejsce w obszarze

pamięci użytkownika : 00h...7Fh. Tutaj jednak dozwolone jest adresowanie tylko rejestrów z zakresu 20h...2Fh – czyli po przemnożeniu: $(2Fh - 20h) * 8 = 128$ bitów. Podczas pisania programu użytkownik ma dostęp do nich wszystkich podobnie jak w trybie adresowania pamięci, z tym że w tym przypadku odczyt i zapis poszczególnych bitów może odbywać się w sposób bezpośredni, czyli poprzez podanie adresu bitu (0...127).

Uff !, jeżeli czegoś nie rozumiesz, nie przejmuj się przy okazji nauki programowania, wszystkie wątpliwości natychmiast znikną.

W kolejnym odcinku dokończenie opisu wewnętrznej pamięci użytkownika oraz omówienie możliwości adresowania przez 8051 zewnętrznej pamięci programu oraz danych.

Sławomir Surowiński