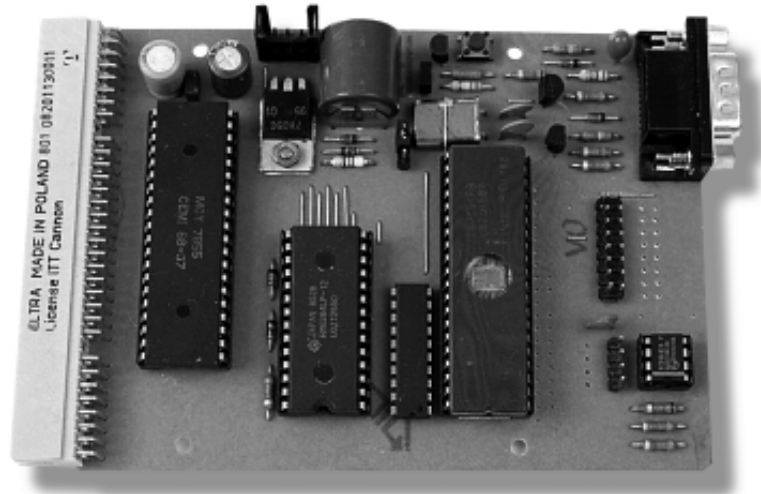


Mikroprocesorowy system edukacyjny, część 1

kit AVT-353

Prezentowane urządzenie powstało z myślą głównie o tych Czytelnikach, którzy stawiają pierwsze kroki w świecie mikrokontrolerów. Jest to bowiem kompletny mikrokomputer, z wbudowanymi wewnętrzną pamięć gotowymi procedurami, które użytkownik może w prosty sposób wykorzystać do tworzenia własnych, bardzo złożonych programów.

Także zaawansowani użytkownicy mikrokontrolerów znajdą tutaj wiele interesujących rozwiązań, które będą mogli wykorzystać w swoich opracowaniach. Dzięki bogatym możliwościom oprogramowania wbudowanego w urządzenie, prezentowany system można zastosować także jako bardzo przydatne narzędzie warsztatowe.



Spośród wielu dostępnych na rynku rozwiązań, wspomagających proces projektowy sterowników, opartych na mikrokontrolerach z rodziny MCS51, znaczna ich liczba charakteryzuje się wadami, częstokroć ograniczającymi ich zastosowanie przez amatorów techniki mikroprocesorowej. Do tych wad należą: wysoka cena modułu bazowego i często konieczność stosowania dodatkowego programatora pamięci lub emulatora mikroprocesora. Jednak najważniejszą niedogodnością jest konieczność własnoręcznego oprogramowania systemu od podstaw.

Projekt niniejszy stanowi odmienne podejście do problemu - stosunkowo prosty system mikroprocesorowy, oparty o mikrokontroler z rodziny MCS51, integruje w sobie funkcję systemu uruchomieniowego oraz emulatora pamięci typu EPROM, ułatwiającego ładowanie i testowanie kolejnych wersji programu. Ułatwia to zintegrowany system operacyjny, będący zbiorem procedur wspomagających proces tworzenia oprogramowania użytkownika. Udostępnia on operacje realizujące kontakt ze światem zewnętrznym przez: interfejs RS232, magistralę I²C, wyświetlacz ciekłokrystaliczny oraz programowaną klawiaturę. Dodatkowo, system operacyjny oferuje procedury operacji arytmetycznych, przetwarzania

danych, pomiarów i generacji wielkości nieelektrycznych, takich jak: czas, częstotliwość i okres.

Opis projektu został podzielony na trzy części. W pierwszej zostanie omówiony sposób działania oraz montaż systemu wraz z towarzyszącym mu oprogramowaniem funkcji emulatora EPROM. W drugiej części omawia się sposób połączenia i użytkowania wyświetlacza ciekłokrystalicznego, który ze względu na różnorodność typów i producentów może następczą kłopoty. W części trzeciej zostaną omówione procedury systemu operacyjnego wraz z przykładami ich użycia w programach użytkownika.

Budowa i działanie systemu uruchomieniowego

Schemat elektryczny układu przedstawiono na **rys.1**. Jednostką centralną systemu jest mikrokontroler U1, pracujący w konfiguracji z wewnętrzną pamięcią programu (końcówka EA=1), zawierającą procedury ładujące oprogramowanie użytkownika, jak i system operacyjny. Jednocześnie mikrokontroler wykorzystuje możliwość współpracy z zewnętrzną magistralą, dającą dostęp do zewnętrznego obszaru pamięci. Magistrala jest uzyskiwana przez porty P0 i P2 mikrokontrolera

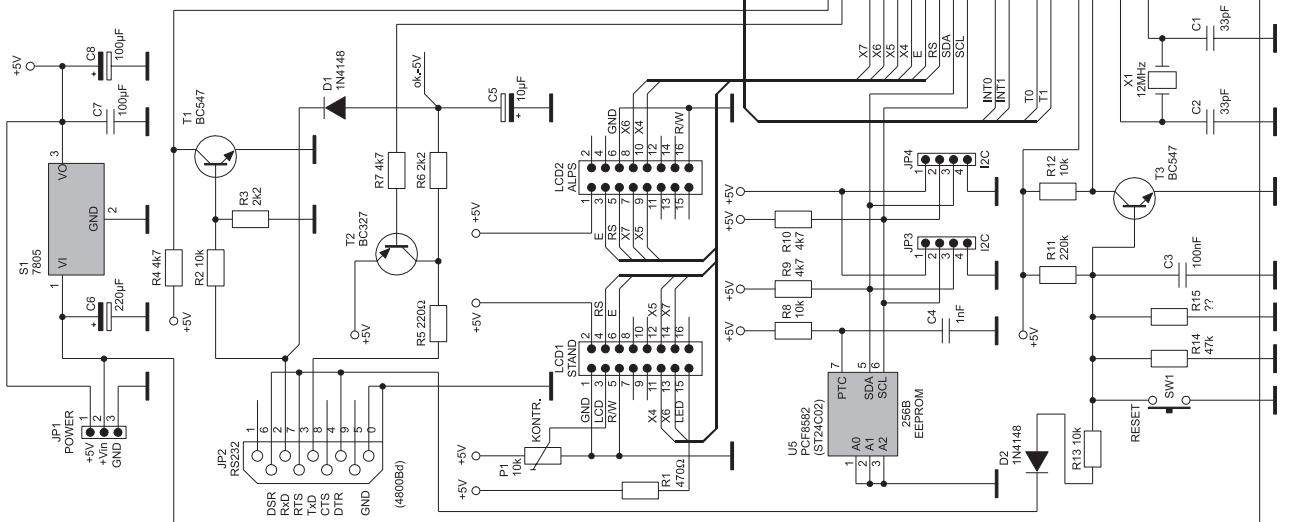
oraz rejestr zatraskowy U4, pracujący z aktywnymi wyjściami trójstanowymi (końcówka /OE = 0). Daje to dostęp do 16 linii adresowych - linie A0 do A7 zatraskiwane są sygnałem ALE na wyjściach rejestru U4, natomiast linie A8 do A15 uzyskiwane są bezpośrednio na końcówkach portu P2 (linie A13 i A14 pozostaną niewykorzystane). Dodatkowo jest wyodrębnionych osiem linii danych D0 do D7, uzyskiwanych na końcówkach portu P0 mikrokontrolera.

W układzie zastosowano koncepcję połączonej pamięci programu i danych, której realizacja polega na iloczynie logicznym sygnałów /PSEN oraz /RD mikrokontrolera (diody D3 i D4 wraz z rezystorem R16). Uzyskany w ten sposób sygnał steruje końcówką /OE pamięci RAM (U2), w efekcie czego mikrokontroler może zarówno pobierać rozkazy (sygnał /PSEN), jak i odczytywać (sygnał /RD) oraz (oczywiście) zapisywać (sygnał /WR) dane podczas odwołań do zewnętrznej pamięci RAM.

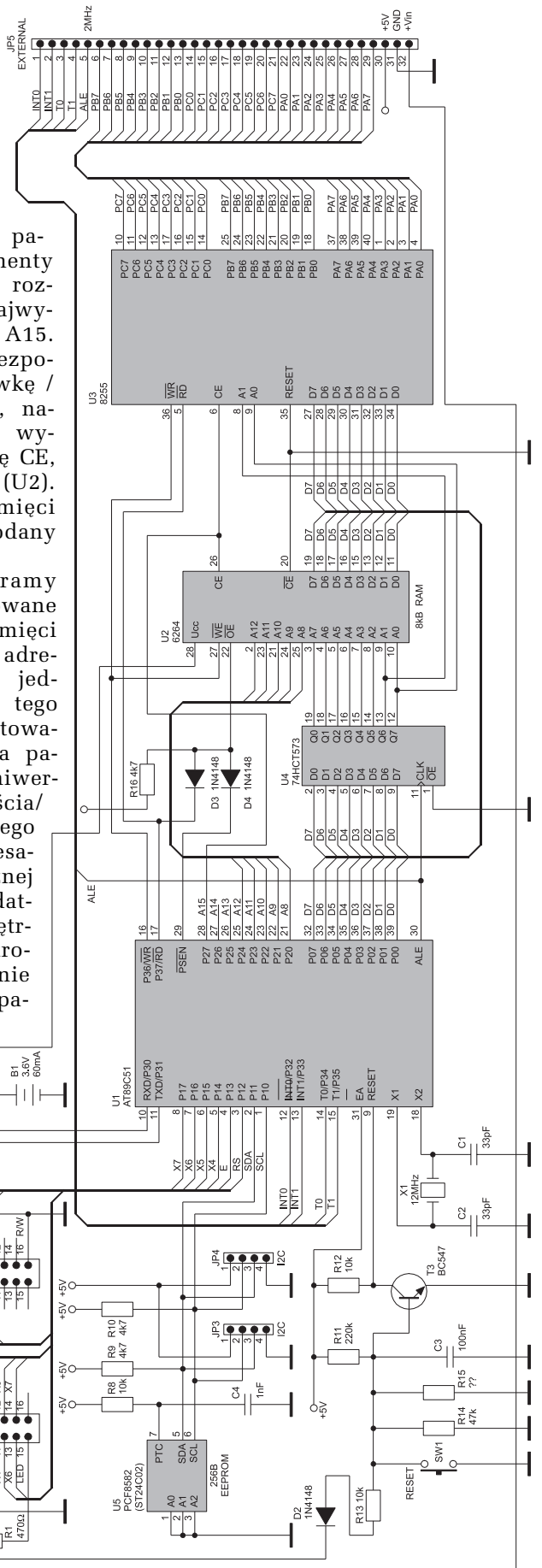
Dodatkowym układem scalonym dołączonym do magistrali jest uniwersalny układ wejścia/wyjścia PIO (U3), dający do dyspozycji użytkownika trzy osmiobitowe kanały wejścia/wyjścia. Został on umieszczony w przestrzeni zewnętrznej pamięci danych, zajmując w niej cztery adresy i bajty pamięci - trzy dla portów oznaczonych odpowiednio PA, PB i PC oraz dla rejestru sterującego oznaczonego CTRL.

Mapa pamięci systemu, określająca dostęp do wewnętrznej i zewnętrznej pamięci programu oraz danych, jak i układu wejścia/wyjścia PIO, przedstawiona jest na **rys. 2**. Wynika z niej podział obszaru pamięci na dwa segmenty o pojemności 32kB, różniane stanem najwyższej linii adresowej A15. Stan niski wybiera bezpośrednio, przez końcówkę /CE, układ PIO (U3), natomiast stan wysoki wybiera, przez końcówkę CE, układ pamięci RAM (U2). Końcówka /CE pamięci RAM ma na stałe podany stan niski.

Tak więc programy użytkownika są ładowane do zewnętrznej pamięci RAM, począwszy od adresu 8000h, uzyskując jednocześnie dostęp do tego samego obszaru traktowanego jako zewnętrzna pamięć danych, oraz uniwersalnego układu wejścia/wyjścia, umieszczonego pod najniższymi adresami obszaru zewnętrznej pamięci danych. Dodatkowo, dzięki wewnętrznej logice mikrokontrolera, każde odwołanie do najniższych 4kB pamięci programu jest



Rys. 1. Schemat układu.



zawsze przełączane na wewnętrzną pamięć programu mikrokontrolera, dzięki czemu program użytkownika może w prosty sposób korzystać z zapisanych tam procedur systemu operacyjnego. Skutkiem ubocznym tego mechanizmu oraz prostoty samego wybierania układów podłączonych do magistrali zewnętrznej mikrokontrolera jest fakt istnienia nieokreślonego obszaru w zewnętrznej pamięci programu. Rozciąga się on od adresu 1000h (4kB+1) aż do końca dolnego segmentu, wyznaczonego adresem 7FFFh. Jednak fakt ten, oprócz treści czysto informacyjnej, nie stanowi jakiegokolwiek przeszkody dla użytkownika systemu.

Bardziej dociekliwi Czytelnicy z pewnością zauważą zastosowanie w systemie niepełnego dekodowania pamięci RAM, jak i uniwersalnego układu wejścia/wyjścia - pamięć RAM jest powielona czterokrotnie w górnym segmencie zewnętrznej pamięci, natomiast układ PIO powielony jest aż 8192 razy w segmencie dolnym.

Na dalszym etapie, aby uniknąć niejednoznaczności, przyjęto początek zewnętrznej pamięci RAM od adresu 8000h, natomiast układu PIO od adresu 0h. Założenie powyższe jest uzasadnione wygodą użytkownika, gdyż generalnie dostęp do pamięci RAM jest realizowany przez wskaźnik DPTR, natomiast układ PIO będzie adresowany przez rejestry R0 lub R1 mikrokontrolera.

Budowę systemu uruchomieniowego uzupełniają obwody pomocnicze umożliwiające pracę, jak i podstawowy kontakt ze światem zewnętrznym. Należy do nich interfejs szeregowy RS232, zbudowany w oparciu o tranzystory T1 oraz T2 wraz ze współpracującymi rezystorami R2, R3, R4, R5, R6, R7, diodą D1 i kondensatorem C5. Tranzystor T1 stanowi prosty inwerter, przekształcający poziomy napięcia charakterystyczne dla standardu łącza szeregowego (typowo ±12V) na poziomy napięcia TTL, wymagane przez wejście szeregowe RxD mikrokontrolera. Ponieważ w systemie nie zastosowano przetwornicy napięcia ujemnego, jest ono pobierane w sposób sztuczny z końcówki RxD (dostępnej na złączu JP2). Stan spoczynkowy

nadajnika urządzenia sprzęgniętego z systemem, jakim jest w standardzie RS232 ujemny poziom napięcia, przenosi się przez diodę D1, ładując kondensator C5 napięciem ujemnym. Uzyskany na tym elemencie potencjał jest przenoszony z powrotem przez rezystory R6 i R5 do urządzenia sprzęgniętego z systemem, sygnalizując stan spoczynkowy nadajnika zintegrowanego w mikrokontrolerze.

Zastosowana na złączu JP2 zwora łącząca sygnały DSR, RTS i DTR wymusza przy podłączeniu urządzenia zewnętrznego pełnym kablem modemowym pracę w trybie *null modem*, przy jednoczesnym zachowaniu wszystkich funkcji dodatkowych. Kolejnym obwodem umożliwiającym kontakt systemu ze światem zewnętrznym jest sprzęg wyświetlacza ciekłokrystalicznego.

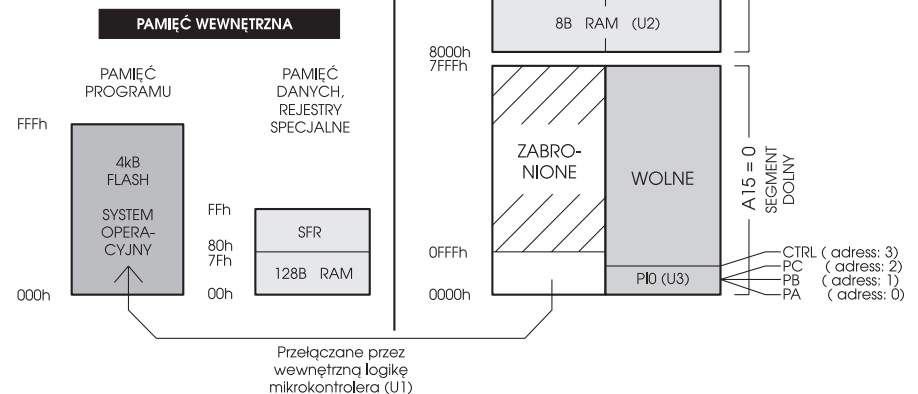
Trzecim obwodem służącym do komunikacji jest interfejs magistrali I²C. Składa się on z dwu złącz magistrali JP3 i JP4, na które wprowadzono zasilanie, sygnał danych (SDA) i zegara (SCL). Oporniki R9 i R10 zastosowano celem właściwej polaryzacji linii danych i zegara.

Na płytce systemu uruchomieniowego umieszczono opcjonalną pamięć EEPROM (U5) współpracującą z magistralą I2C. Jest to nieulotna, reprogramowana elektrycznie pamięć stała o pojemności 256B, w której użytkownik systemu może przechowywać dane konfiguracyjne swoich programów (z jednoczesną ich ochroną przed zanikiem zasilania systemu). Zastosowano układ firmy PHILIPS PCF8582 w dowolnym wykonaniu, wymagający do-

datkowo rezystora R8 oraz kondensatora C4. Użycie zastępczego układu firmy SGS-THOMSON typu ST24C02 wiąże się z koniecznością zastąpienia kondensatora C4 zworką. Oczywiście można zastosować inny, niż wymienione typ pamięci, stosownie do niego konfigurując elementy R8 i C4.

Ostatnim, widocznym na schemacie, ogniwem komunikacji jest złącze JP5, na które wyprowadzono końcówki trzech ośmiobitowych portów wejścia/wyjścia, linie przerwań mikrokontrolera /INT0 i /INT1 oraz wejścia liczników/czasomierzy T0 i T1. Dostępny jest także sygnał wzorcowy ALE o częstotliwości 2MHz oraz końcówki zasilania: +Vin, +5V, a także masa (GND).

Elementem bezpośrednio komunikacji, nie zaznaczonym na schemacie, a występującym tylko w projekcie obwodu drukowanego, jest złącze umieszczone po lewej stronie mikrokontrolera U1. Jak można zauważyć, integruje ono porty P1 oraz P3, które są wykorzystywane do komunikacji poprzez opisane wcześniej układy sprzęgające oraz dodatkowe punkty zasilania +5V i masę (GND). Dostępne w ten sposób pola lutownicze przeznaczyć można na rozszerzenie liczby dostępnych końcówek, realizujących standardowo określone funkcje - oznaczeniu punktu, przy pomocy



Rys. 2. Mapa pamięci systemu.

schematu ideowego, przyporządkować należy spełnianą w systemie rolę.

W skład obwodów umożliwiających pracę mikrokontrolera U1 wchodzi rezonator kwarcowy X1 wraz z pomocniczymi kondensatorami C1 i C2. Częstotliwość jego pracy powinna wynosić typowo 12MHz, co jest wartością wzorcową dla procedur systemu operacyjnego, i nie powinna być zmieniana.

Ostatnim układem wymagającym szczegółowego omówienia jest obwód zerowania mikrokontrolera, wykonany w oparciu o tranzystor T3 wraz z współpracującymi elementami (rezystory: R11, R12, R13, R14, R15 oraz D2, C3 i SW1), którego działanie polega na kontroli wartości napięcia zasilania systemu. Utrzymywany w stanie przewodzenia tranzystor T3, którego baza jest spolaryzowana przez kalibrowany dzielnik złożony z rezystorów R11, R14 i R15, wymusza w trakcie normalnej pracy stan niski na końcówce RESET mikrokontrolera.

Obwód zerowania posiada integralną funkcję restartu ręcznego, dokonywanego przez zwarcie styków mikroprzełącznika SW1 oraz zerowania z systemu nadrzędnego, podłączonego przez złącze JP2. Uzyskiwane jest to dzięki połączeniu rezystora R13 oraz diody D2 z końcówką DTR złącza JP2. Pojawienie się napięcia dodatniego lub pozostawienie tej końcówki nie podłączonej nie ma wpływu na pracę systemu uruchomieniowego.

Napięcie ujemne, wprowadzone na tę końcówkę, utrzymuje system w stanie ciągłego zerowania (stan wysoki na końcówce RESET mikrokontrolera). Możliwość powyższa jest wykorzystywana podczas ładowania programu użytkownika do pamięci RAM, gdyż po zerowaniu każdorazowo automatycznie jest uruchamiana procedura ładująca, umieszczona w wewnętrznej pamięci programu mikrokontrolera.

Na koniec opisu budowy systemu uruchomieniowego przedstawimy obwód zasilania, składający się ze stabilizatora głównego oraz rezerwowego zasilania pamięci RAM. Stabilizator główny jest wykonany w oparciu o monolityczny regulator S1, dostarczający niezbędnego napięcia o wartości +5V oraz kondensatorów

filtrująco-odsprzęgających C6, C7 i C8. Generalnie, system uruchomieniowy jest przystosowany do zasilania napięciem niestabilizowanym o wartości 8 do 15V (typowo 12V), ze źródła o wydajności prądowej wynoszącej około 200mA. Zastosować można zwykły zasilacz kalkulatorowy, spełniający wyżej wymienione warunki. Nie należy zasilać systemu napięciem o wartości +5V, doprowadzonym do oznaczonych tak końcówek złącz JP1 i JP5, ponieważ jest ono tam wyprowadzone celem zasilania zewnętrznych układów logicznych napięciem stabilizowanym.

W skład obwodu rezerwowego zasilania pamięci U2 wchodzi dioda D5 i rezystor R17, których zadaniem jest kluczkowanie źródła zasilania. Może być nim napięcie +5V dołączone przez diodę D5 (rezystor R17 ogranicza prąd ładowania akumulatora) lub akumulator B1 zasilający pamięć przez rezystor R17. Zastosowano tu prostą metodę ładowania akumulatora malejącym prądem, tj. w miarę wzrostu stopnia naładowania akumulatora wzrasta jego napięcie, a tym samym zmniejsza się spadek napięcia na rezystorze R17, maleje przepływający przez niego prąd, będący jednocześnie prądem ładowania akumulatora.

Funkcja emulatora EPROM

Funkcja emulatora EPROM jest uruchamiana każdorazowo po zerowaniu systemu, kiedy to zainstalowane wewnątrz mikrokontrolera oprogramowanie analizuje stan odbiornika łącza szeregowego. Nieodebranie jakiegokolwiek znaku przez czas około 0,9 sekundy lub odebranie znaku różnego od „?” powoduje natychmiastowe przekazanie sterowania do programu użytkownika (o ile ten został uprzednio załadowany do pamięci systemu). Odebranie natomiast znaku „?” powoduje nadanie winiety zintegrowanego programu ładującego oraz uruchomienie procedur realizujących funkcję emulatora EPROM, które oczekują na transmisję programu użytkownika. Jeżeli pierwszym, odebrany znakiem jest „:”, to cały przesyłany program użytkownika traktowany jest jako zbiór w formacie heksadecymalnym, uzyskiwanym z większości

kompilatorów i assemblerów stworzonych dla rodziny procesora 8051. Tak więc możliwe jest bezpośrednio przesyłanie do systemu uruchomieniowego plików wynikowych kompilacji, czy też asemblacji programów użytkownika, gdyż oprogramowanie obsługuje funkcję emulatora, automatycznie dokonując przekodowania z formatu heksadecymalnego na binarny. Nadmienić należy, iż obsługiwanym formatem zbiorów szesnastkowych jest format INTEL HEX (opisany w EP9/97).

Jeżeli pierwszym odebrany znakiem nie będzie „:”, cały przesyłany program użytkownika jest traktowany jako zbiór binarny - zostanie on bezpośrednio załadowany do pamięci systemu. Podczas ładowania zbiorów w tym formacie zrealizowano tzw. „echo”. Polega ono na kontroli poprawności zapisu bajtu programu do pamięci, a następnie na odesłaniu tak uzyskanego bajtu, przy pomocy łącza szeregowego, z powrotem do urządzenia przesyłającego kod programu użytkownika, którym będzie komputer typu PC. Umożliwia to użytkownikom z większym doświadczeniem zastosowanie w miejscu pamięci RAM typu 6264 układu pamięci EEPROM typu np. 28C64. Mechanizm ten należy wykorzystać następująco: bajt odsyłany jest dopiero po uzyskaniu poprawnego zapisu komórki pamięci EEPROM. Tak więc, program ładujący, działający na komputerze PC, otrzymuje w ten sposób informację o zakończeniu cyklu zapisu, co umożliwia przesłanie kolejnego bajtu programu użytkownika.

Po załadowaniu całości programu użytkownika, oprogramowanie emulatora EPROM przekazuje mu kontrolę nad systemem uruchomieniowym. Sytuacja ta pozostanie nie zmieniona, aż do sprzętowego wyzerowania procesora lub wykonania długiego skoku z wnętrza programu użytkownika pod adres 0 (tj. LJMP 0).

Programy użytkownika powinny być asemblowane od adresu 8000h (wektory przerwań odpowiednio 8003h, 800Bh, 8013h, itd.), gdyż pod takim adresem znajduje się połączona pamięć programu i danych U2 (patrz rys. 2 - mapa pamięci systemu).

Pamięć ta ma 8kB pojemności, co dla większości zastosowań jest

wartością w zupełności wystarczającą. Uważać jednak należy, aby nie przysyłać programów o objętości większej niż 8kB binarnie, ponieważ całość informacji znajdująca się poza granicą 8kB spowoduje zniszczenie danych znajdujących się już w pamięci. Spowodowane jest to niepełnym dekodowaniem układu pamięci. Te dwa opisane ograniczenia są jedynymi dotyczącymi programów użytkownika. Wraz z systemem uruchomieniowym są dostarczane na dyskietce dwa programy wspomagające jego funkcje: *emul51.exe* dla funkcji emulatora, oraz *recv51.exe* dla obsługi interfejsu szeregowego. Zastosowanie programów wymaga wykonania kabłka połączeniowego według rys. 3, który od strony komputera PC może być zakończony odpowiednio złączem CANON25 (jeżeli dysponujemy złączem dużym), lub CANON9 (jeżeli dostępne jest złącze małe). Wyboru wariantu należy dokonać indywidualnie, w zależności od wolnego portu szeregowego COMn. Jeżeli dysponujemy kilkużyłowym kablem w oplocie ekranującym, to oplót należy użyć jako połączenie masowe (GND).

Zastosowanie programu *emul51.exe* umożliwia w pełni automatyczne ładowanie i uruchomienie programu użytkownika wraz z jednoczesną, podstawową obsługą sytuacji nieprawidłowych. Program obsługuje system uruchomieniowy podpięty do komputera PC przez złącze interfejsu szeregowego COM1 lub COM2.

Składnia wywołania programu jest następująca:

`<emul51 nazwa_pliku [COMn]>`,
gdzie `<nazwa_pliku>` jest nazwą pliku wynikowego kompilacji lub asemblacji wraz z rozszerzeniem, który ma zostać załadowany do pamięci systemu. Natomiast opcjonalny parametr `<COMn>` informuje program o użytym łączy

szeregowym COM1 lub COM2 (jeżeli numer łączy zostanie pominięty, przyjęty będzie domyślnie COM1).

Program *emul51.exe* umożliwia ładowanie plików heksadecymalnych, jak i binarnych, jednak obsługa tych ostatnich nie uwzględnia tzw. „echa” realizowanego przez zainstalowane w mikrokontrolerze oprogramowanie emulatora EPROM.

Jak wspomniano, dodatkową opcją programu jest sygnalizacja nieprawidłowości, takich jak błędne podanie oznaczenia portu szeregowego, nazwy pliku lub brak podłączenia wtyków do komputera PC i systemu uruchomieniowego, nie włączone zasilanie systemu itd.

Drugi program - *recv51.exe* służy do obustronnej komunikacji użytkownika z systemem realizującym jego program. Zadanie to jest wykonywane przez odbiór i wyświetlanie na ekranie monitora napływających łączy szeregowych informacji, jak i przesyłanie danych wprowadzanych bezpośrednio z klawiatury komputera PC do systemu uruchomieniowego. Dodatkowo wprowadzono możliwość zdalnego zerowania (jednak bez możliwości ingerencji w załadowany do pamięci program użytkownika) oraz zapis przebiegu sesji łączności do pliku *recv51.ssa*. Należy wykażać jednak pewną ostrożność, gdyż w razie wykonania zdalnego zerowania, poprzednio zapisane w tym pliku informacje zostaną skasowane i zapis rozpocznie się od nowa.

Składnia wywołania programu jest następująca:

`<recv51.exe [COMn]>` - opcjonalny parametr `<COMn>` informuje o użytym łączy szeregowym COM1 lub COM2 (domyślnie przyjęty zostanie COM1). Podobnie jak poprzednio opisany program *emul51.exe*, program *recv51.exe* sygnalizuje nieprawidłowości w sposób niemal identyczny.

W działaniu obu opisanych programów przeszkadzać mogą rezydentnie zainstalowane w komputerze PC sterowniki urządzeń dodatkowych, używające do swoich celów łączy szeregowego, przeznaczonego do

komunikacji z systemem uruchomieniowym. Aby uniknąć niespodzianek należy je bądź zdeinstalować (przez modyfikację odpowiednich plików *config.sys*, i/lub *autoexec.bat*), bądź przenieść na inne dostępne łączy szeregowo.

Należy dodatkowo nadmienić, że ze względu na metodykę uruchomienia systemu uruchomieniowego przez wykorzystanie stanu linii DTR łączy szeregowego, podczas wyłączenia komputera PC oraz bezpośrednio po jego włączeniu system uruchomieniowy będzie pozostawać w stanie ciągłego zerowania, co w sposób oczywisty blokuje jego pracę. Aby umożliwić równoległy start systemu uruchomieniowego bezpośrednio po włączeniu zasilania komputera PC, na początku pliku *autoexec.bat* należy dopisać następujące linie:

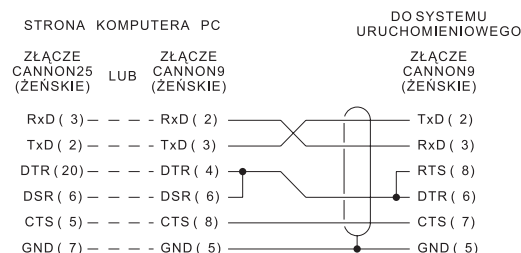
```
@mode COMn,4800,n,8,1 >nul
@echo ! >COMn,
```

gdzie `<COMn>` określa numer łączy przeznaczonego do komunikacji z systemem.

W pierwszej linii określamy parametry łączy wymagane przez zintegrowane oprogramowanie emulatora EPROM (szybkość 4800Bd, bez kontroli parzystości, osiem bitów danych i jeden bit stopu). Natomiast w drugiej linii przesyłamy znak różny od „?”, z jednoczesnym ustawieniem stanu 1 na linii DTR łączy szeregowego. Operacja ta spowoduje natychmiastowe uruchomienie programu użytkownika, o ile był on uprzednio załadowany do pamięci.

Montaż i uruchomienie

Widok płytki drukowanej znajduje się na wkładce wewnątrz numeru, a rozmieszczenie elementów pokazano na rys. 4. Montaż należy rozpocząć od ewentualnego sprawdzenia i rozwiercenia otworów pod następujące elementy: wiertłem 3-mm cztery otwory nośne umieszczone w pobliżu krawędzi płytki i wiertłem 2,5-mm otwór mocujący stabilizator S1. Otwory pod następujące elementy: złącze JP2, akumulator B1, stabilizator S1, złącze zasilania JP1 oraz ewentualnie punkty lutownicze złącza JP5 (o ile zamierzamy je w przyszłości lutować) należy rozwiercić wiertłem 1-mm. Dodatkowo, wier-



Rys. 3. Sposób wykonania połączeń w uproszczonym kablu RS232 (null modem).

tłem 3-mm należy rozwiercić dwa otwory nośne, umieszczone na dołączonej do zestawu płytce uniwersalnej.

Następnie, przy pomocy piłki włosowej należy wyciąć szczelinę pomiędzy dwoma otworami umieszczonymi w punktach lutowniczych elementu nośnego złącza JP2 - operację powyższą wykonać należy dwukrotnie, dla dwu punktów lutowniczych. Montaż układu rozpoczynamy od wlutowania osiemnastu mostków lutowniczych, zwracając szczególną uwagę na umieszczone pod układem scalonym U2.

UWAGA: nie montować na razie mostka przebiegającego nad rezonatorem kwarcowym X1.

Wlutowanie stabilizatora S1 po-

WYKAZ ELEMENTÓW

Rezystory

- R1: 470Ω
- R2, R8, R12, R13: 10kΩ
- R3, R6: 2,2kΩ
- R4, R7, R9, R10, R16: 4,7kΩ
- R5: 220Ω
- R11: 220kΩ
- R14: 47kΩ
- R15: dobierany
- R17: 270Ω
- P1: 10kΩ - potencjometr miniaturowy

Kondensatory

- C1, C2: 33pF
- C3, C7: 100nF
- C4: 1nF
- C5: 10μF/16V
- C6: 220μF/25V
- C8: 100μF/16V

Półprzewodniki

- U1: AT89C51
- U2: 6264
- U3: 8255 lub 82C55
- U4: 74HCT573
- U5: PCF8582 (ST24C02 - patrz tekst)
- S1: 7805
- D1, D2, D3, D4, D5: 1N4148
- T1, T3: BC547
- T2: BC327

Różne

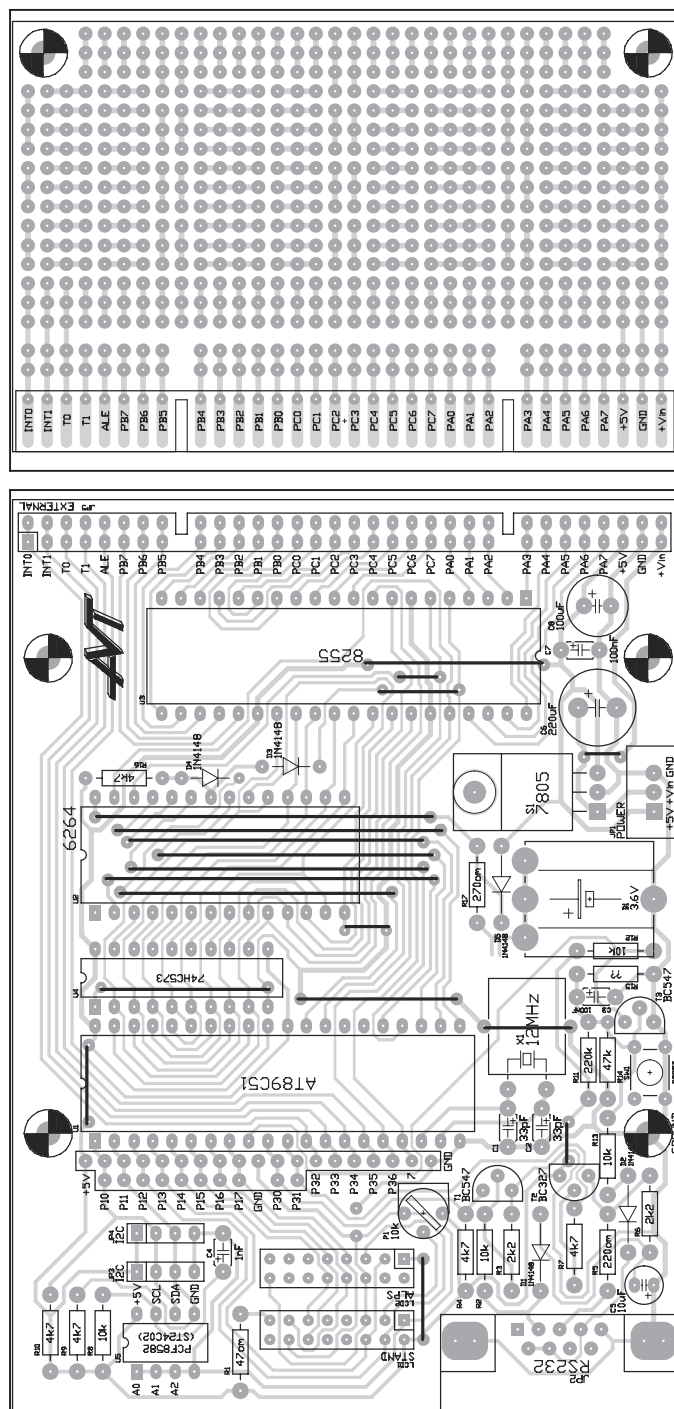
- X1: kwarc 12MHz
- JP1: gold-pin 1x3
- JP2: złącze DSUB-9
- JP3, JP4: gold-piny 1x4
- LCD1, LCD2: złącza męskie ZWS-10
- B1: 3,6V/60mAh
- Sw1: mikroprzełącznik

winno, w miarę możliwości, być poprzedzone delikatnym posmarowaniem powierzchni styku elementu pastą silikonową, po czym powinien on zostać, w pierwszej kolejności, dokręcony do powierzchni druku śrubką M2,5 i dopiero przylutowany. Ma to na celu zapewnienie dobrych warunków chłodzenia, osiąganych przez dobry kontakt elementu z podłożem.

Kolejno należy wlutować wszystkie rezystory i diody, pomijając potencjometr P1 oraz rezystor R15, który powinien być dobrany indywidualnie. Stosownie do typu zastosowanego układu U5, należy także dobrać elementy R8 i C4. Jeżeli zamierzamy korzystać z możliwości wymiennej instalacji płytki uniwersalnej, należy stosownie przyciąć złącze krawędziowe do odpowiedniego rozmiaru, usunąć kołki z pozycji, na których znajdują się szczeliny prowadzące płytki uniwersalnej, przygiąć końcówki złącza i wlutować je.

Następną czynnością jest wlutowanie podstawek pod układy U1, U2 i U3 oraz samych układów U4 i opcjonalnie U5 (jeżeli zamierzamy z niego korzystać).

UWAGA: podstawka pod U3 jest wlutowana odwrotnie w stosunku do U1 i U2.



Rys. 4. Rozmieszczenie elementów na płytce drukowanej.

W dalszej kolejności należy wlutować pozostałe złącza i elementy systemu, pozostawiając na koniec montaż rezonatora X1, którego obudowę należy przylutować kroplą cyny do opasowującego go mostka. Ostatnią czynnością pozostaje umieszczenie układów: U1, U2 i U3 w podstawkach - U3 odwrotnie w stosunku do U1 i U2 oraz dobór rezystora R15. Dokonujemy tego po uprzednim zasileniu systemu

napięciem niestabilizowanym o wartości 8 do 15V, podanym przez złącze JP1 oraz sprawdzeniu poprawności zasilania +5V.

Doboru wartości rezystora R15 dokonać należy podłączając w jego miejsce potencjometr o wartości typowo 220k Ω , ustawiony wstępnie na minimalną wartość rezystancji. Stopniowo zwiększać należy jego rezystancję, mierząc napięcie kolektor-emiter tranzystora T3 - po osiągnięciu przez nie wartości około 0,15V pomiar rezystancji potencjometru da poszukiwaną wartość rezystora R15. Oczywiście, należy zastosować rezystor o najbliższej, możliwej do pozyskania, wartości. Ostatnią kwestią pozostaje montaż, pominiętego wcześniej, potencjometru P1, stosowanego tylko w przypadku konieczności zewnętrznej regulacji kontrastu wyświetlacza ciekłokrystalicznego (w innym przypadku nie jest on wymagany). Tak przygotowana płytką jest gotowa do pierwszego uruchomienia.

Połączony system uruchomieniowy z komputerem typu PC przy użyciu opisanego wcześniej kabela, włączamy zasilanie systemu, w katalogu zawierającym dostarczone na dyskietce programy wydajemy komendę: `<emul51.exe start.hex [COMn]>`, gdzie COMn jest numerem portu szeregowego przeznaczonego do komunikacji z systemem. Pojawienie się winiety programu ładującego, a następnie wskaźnika procentowego zaawansowania procesu ładowania i komunikatu o zakończeniu ładowania jest sygnałem sukcesu - pierwszy program został załadowany i uruchomiony w naszym systemie. Aby zobaczyć efekt tego działania należy w tym samym katalogu uruchomić program `<recv51.exe [COMn]>`. Na ekranie monitora

powinna pojawić się kolumna utworzona z napisu: „System uruchomieniowy, program testowy.“. Dodatkowo na dyskietce znajduje się plik `start.a51`, zawierający uruchomiony program w postaci źródłowej wraz z odnośnymi komentarzami. Należy nadmienić, iż jest to pierwszy program wykorzystujący procedury systemu operacyjnego. Jeśli jednak nie uzyskaliśmy oczekiwanego sukcesu, konieczne jest wnikliwie sprawdzenie poprawności montażu płytki oraz ciągłości wszystkich ścieżek, jak i braku mikrozwarcień pomiędzy nimi, gdyż tylko takie mogą być przyczyny tego, że system nie działa.

Na dyskietce znajduje się jeszcze jeden plik testowy, służący do sprawdzenia poprawności komunikacji systemu ze złączem rozszerzenia JP5, w czym pośredniczy uniwersalny układ wejścia/wyjścia U3. Wykonać należy polecenie: `<emul51.exe piotest.hex [COMn]>` - załadowany i uruchomiony zostanie program wywołujący na wszystkich liniach portów PA, PB i PC naprzemienne stany niskie i wysokie, z tym, że częstotliwość uzyskiwana na linii oznaczonej przez 0 jest najwyższa, po czym kolejno zmniejsza dwukrotnie swą wartość dla każdej następnej linii, osiągając dla linii oznaczonej przez 7 wartość najniższą. Sprawdzenia poprawności generowanych przebiegów można dokonać najprościej przy pomocy diody świecącej, połączonej z odpowiednio dobranym rezystorem ograniczającym. Uzyskanie równomiernych przebiegów o kolejno malejącej częstotliwościach wskazuje na poprawną pracę układu U3.

W tym miejscu kończymy opis budowy i uruchomienia systemu. Dalej zajmować się będziemy jedynie aspektami tworzenia prog-

ramowania z użyciem procedur systemu operacyjnego, jednakże Czytelnicy dysponujący większym doświadczeniem już mogą rozpocząć użytkowanie proponowanego systemu uruchomieniowego.

Krzysztof Kuryłowicz

Oprogramowanie do obróbki plików binarnych i szesnastkowych oraz do kompilacji programów dla mikrokontrolerów 8051 znajduje się na płycie CD-EP1.