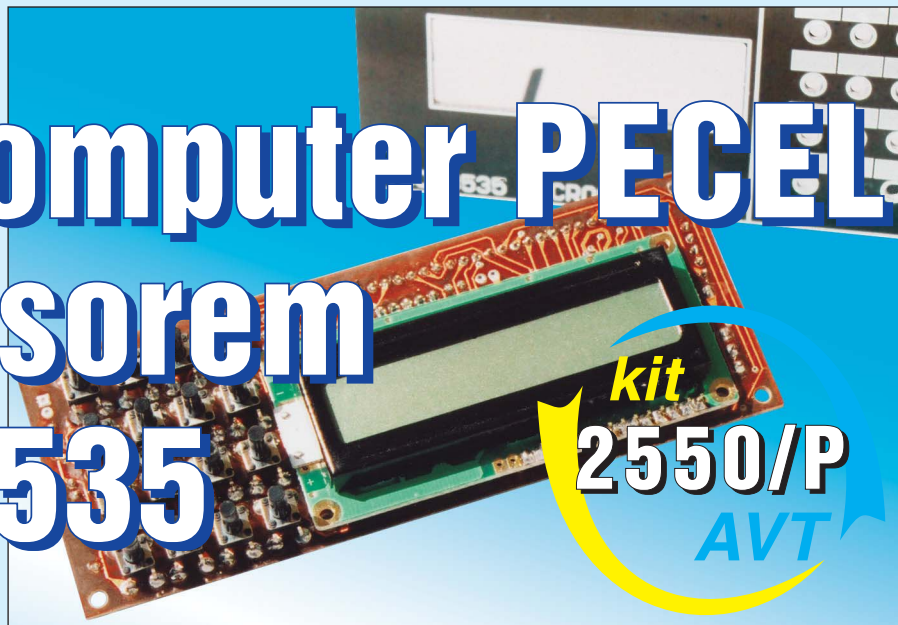




# Mikrokomputer PECEL z procesorem AT90S8535



## Część 2

Nasz minikomputer jest w zasadzie gotowy. Do wykonania pozostały już tylko drobiazgi: banalnie prosty programator, sterowany z pakietu BASCOM AVR, który będzie służył do wprowadzania napisanego programu do pamięci minikomputera oraz dwa kabelki: jeden do programatora, a drugi do łączenia PECEL-a z portem szeregowym komputera. Zacznijmy od montażu programatorka.

Schemat elektryczny układu programatora AVR został pokazany na **rysunku 4**. Jak łatwo zauważyć ponieważ cała inteligencja programatora skupiona została w jego części software'owej układ został maksymalnie uproszczony i zawiera tylko aktywny element: IC1 - 74HCT244.

Układ jest zmodyfikowaną wersją programatora STK200 firmy Kanda, bardzo popularnego wśród elektroników. W Internecie można znaleźć sporo oprogramowania obsługującego ten programator, dostępnego jako shareware. Nic więc dziwnego, że wiedząc o popularności tej tysiące razy sprawdzonej konstrukcji, Mark zapewnił jej wsparcie software'owe z poziomu pakietów BASCOM AVR i BASCOM 8051. Widoczny na schemacie układ programatora nie jest urządzeniem związanym na śmierć i życie z naszym minikomputerem: za jego pomocą możemy zaprogramować każdy procesor AVR i niektóre procesory z rodziny '51 (np. AT89S8252). Procesory można programować zarówno w podstawkach wyposażonych w niezbędny rezonator kwarcowy, jak i w systemie, za pomocą specjalnego złącza ISP. Ponieważ nie wszystkie układy bazujące na procesorach AVR posiadają takie złącza, pozwoliłem sobie zaprojektować specjalne

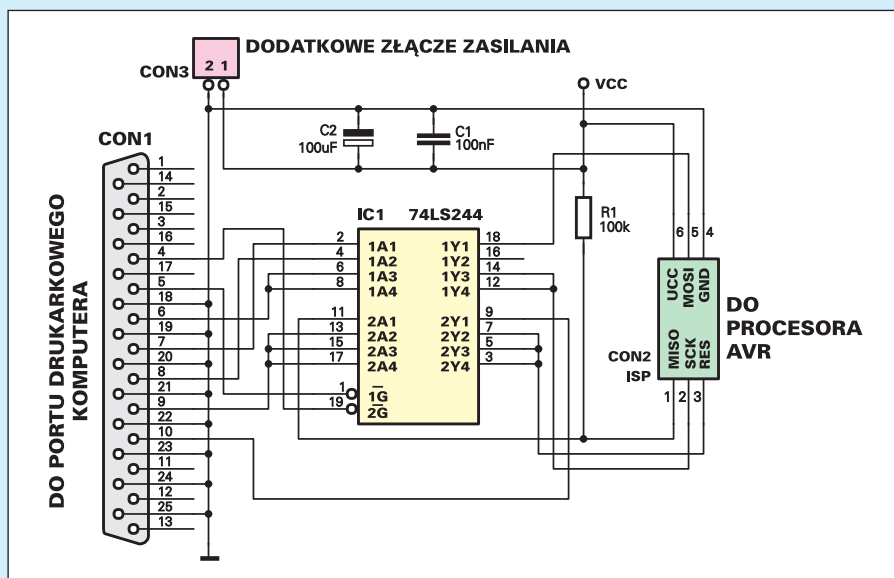
konektory umożliwiające programowanie w systemie bez konieczności dolutowywania przewodów, ani dołączania ich za pomocą chwytaków. Rozmieszczenie wyprowadzeń tych złączy (CON2 na schemacie programatora i CON8 na schemacie minikomputera) jest zgodne ze standardem zalecanym przez firmę ATMEL.

Dla szczególnie dociekliwych Czytelników podaję teraz uproszczony algorytm programowania procesorów AVR. Ci spośród Was, których zagłębianie się w teoretyczne podstawy działania procesorów zbytnio nie interesuje, mogą spokojnie pominąć ten fragment artykułu.

Aby zaprogramować pamięci procesora, programator musi wykonać następujące czynności:

1. Podczas włączania zasilania wymusić stan niski na wejściach RESET! i SCK procesora. Nie wszystkie programatory (w tym opisywany) są w stanie wykonać tę czynność i w taki przypadku konieczne jest, po wymuszeniu stanu niskiego na wejściu SCK, podanie na wejście RESET! dodatniego impulsu o czasie trwania dwóch cykli zegarowych.
2. Po upływie co najmniej 20 ms programator musi wysłać do procesora instrukcję zezwolenia na programowanie. Składnia tej i innych instrukcji podana jest w tabeli poniżej.

Rys. 4



Komentarz:

- a - wyższe bity adresu
- b - niższe bity adresu
- H=0 - niższy bajt, H=1 - wyższy bajt
- o - odczyt danych
- i - zapis danych
- x - bez znaczenia
- A - bit zabezpieczający 1
- B - bit zabezpieczający 2

Instrukcja	Format instrukcji				Działanie
	Bajt 1	Bajt 2	Bajt 3	Bajt 4	
Programming Enable	1010 1100	0101 0011	xxxx xxxx	xxxx xxxx	Zezwolenie na programowanie
Chip Erase	1010 1100	100x xxxx	xxxx xxxx	xxxx xxxx	Kasowanie obu pamięci
Read Program Memory	0010 H000	xxxx xaa	bbbb bbbb	oooo oooo	Odczyt górnej lub dolnej (H) części danych spod adresu a:b
Write Program Memory	0100 H000	xxxx xaa	bbbb bbbb	iiii iii	Zapis górnej lub dolnej (H) części danych spod adresu a:b
Read EEPROM Memory	1010 0000	xxxx xxxx	xbbb bbbb	oooo oooo	Odczyt z pamięci danych spod adresu b
Write EEPROM Memory	1100 0000	xxxx xxxx	xbbb bbbb	iiii iii	Zapis do pamięci danych pod adres b
Write Lock Bits	1010 1100	111x xABx	xxxx xxxx	xxxx xxxx	Zapis bitów zabezpieczających A i B
Read Signature Bits	0011 0000	xxxx xxxx	xxxx xabb	oooo oooo	Odczyt typu układu o spod adresu b

3. Kolejną czynnością będzie sprawdzenie poprawności transmisji. Po wysłaniu przez programator drugiego bajtu instrukcji Programming Enable, procesor powinien odpowiedzieć „odesłaniem“ do programatora wartości tego bajtu. Jeżeli tak się stanie, to należy uznać, że transmisja jest prawidłowa i przystąpić do wykonywania kolejnych instrukcji. Jeżeli jednak programator nie otrzymał „echa“ od procesora, to należy powtórzyć próby nawiązania transmisji. Brak „echa“ po 32 próbie świadczy o niemożności zsynchronizowania układów.

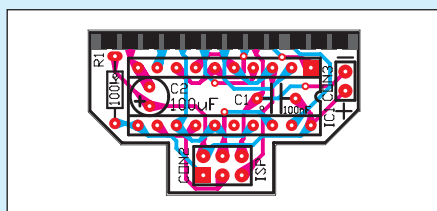
4. Po nawiązaniu transmisji programator powinien wysłać do procesora kolejne instrukcje, przewidziane dla aktualnie wykonywanego zadania. Możliwe jest zaprogramowanie zarówno pamięci danych, jak i programu, odczyt ich wartości oraz zabezpieczenie pamięci programu za pomocą dwóch bitów zabezpieczających.

Po zakończeniu programowania ustawienie stanu wysokiego na wejściu RESET procesora umożliwi jego poprawną pracę.

Jak widać, algorytm programowania poprzez złącze SPI jest dość skomplikowany. Na szczęście, nie musimy go znać na pamięć, ponieważ jest on automatycznie i bez naszego udziału realizowany przez „mądrego“ BASCOM-a.

Na **rysunku 5** została pokazana mozaika ścieżek płytki obwodu drukowanego, wykonanego na laminacie dwustronnym z metalizacją. Płytkę programatorka została tak zwymiarowana, że po zmontowaniu i przylutowaniu do złącza DB25M mieści się „lekkko na wcisk“ w typowej obudowie DB25. Taki spo-

**Rys. 5 Płytkę drukowaną programatora**



sób obudowania programatora daje nam dużą wygodę w posługiwaniu się urządzeniem, które z pewnością będzie dość często podłączane i odłączane do komputera, na zmianę z kablem drukarkowym.

Płytkę montujemy dość nietypowo, ponieważ ze względu na drastyczne ograniczenie jej wymiarów musimy montować elementy po obydwu jej stronach. Montaż rozpoczyna-

my od połączenia płytki ze złączem DB-25.

Na dłuższej krawędzi płytki programatora, po obydwu jej stronach został umieszczony szereg punktów lutowniczych, rozmieszczonych identycznie do wyprowadzeń złącza DB-25. Płytkę wsuwamy „na wcisk“ pomiędzy końcówki lutownicze złącza tak, aby wyprowadzenia konektora DB25 pokryły się dokładnie z punktami lutowniczymi. Podczas dopasowywania płytki do złącza właściwie nie można się pomylić, ponieważ na jednej stronie płytki mamy 12, a na drugiej 13 punktów lutowniczych. Po dokładnym dopasowaniu obu elementów do siebie lutujemy wyprowadzenia złącza, używając lutownicy o cienkim grocie.

Kolejną czynnością będzie wlutowanie w płytkę kondensatorów. Oba te elementy lutujemy od strony druku, oczywiście w przypadku płytki dwustronnej jest to strona umowna. Dla ułatwienia: w płytkach AVT maska lutownicza na stronie ścieżek jest zawsze zabarwiona na czerwono. Kondensatory montujemy na płask, równoległe do powierzchni płytki, a po przylutowaniu obcinamy jak najkrócej ich końcówki.

Układ IC1 montujemy „po bożemu“ na stronie elementów płytki drukowanej. Odstępstwem od regułu jest rezygnacja ze stosowania podstawki, której użycie uniemożliwiłoby umieszczenie płytki w maleńkiej obudowie.

Montaż elektryczny kończymy na przylutowaniu do płytki jedynego rezystora i złącza CON2. Podwójny szereg kątowych goldpinów lutujemy tak, aby jego wolne końcówki znalazły się jak najbliżej płaszczyzny powierzchni płytki.

A więc, programator mamy już w zasadzie gotowy! Pozostaje tylko wyposażyć go w kabel łączący go z programowanym procesorem. Kabel ten wykonujemy z odcinka dziesięciożyłowego przewodu taśmowego o długości ok. 50 cm, zaciskając na jego końcach dwa wtyki 10-pinowe. Niestety, nie są produkowane takie wtyki o sześciu końców-

kach i w naszym kablu cztery przewody pozostaną niewykorzystane.

Na zakończenie umieszczamy płytkę programatora wraz z dołączonym do niej kablem w przeznaczony dla niej obudowie od wtyku DB-25 i skręcamy całość śrubkami.

Uwieńczeniem naszej pracy będzie teraz połączenie programatora z minikomputerem i komputerem PC, na którym został zainstalowany pakiet BASCOM AVR. Pamiętajcie, że połączenia te musimy zawsze wykonywać przy wyłączonym zasilaniu obu urządzeń. Natomiast **przypadkowe, odwrotne połączenie przewodu prowadzącego od programatora do złącza ISP na płycie minikomputera nie grozi żadnymi przykrymi konsekwencjami! Złącze ISP zostało przez ATMEL-a tak sprytnie zaprojektowane, że po zmianie kierunku jego włączenia nie może dojść do uszkodzenia ani procesora, ani programatora i jedynym objawem będzie nieprawidłowe działanie całości.** W praktyce, odwrotne połączenie tego kabla będzie sygnalizowane komunikatem o niemożności zidentyfikowania dołączonego do programatora procesora.

Chciałbym jeszcze wyjaśnić sprawę widocznego na schemacie złącza CON3, o którego roli jak dotąd nie wspominaliśmy. Jest to złącze nie używane podczas pracy programatora z naszym minikomputerem, ponieważ programator jest tu zasilany z płyty minikomputera za pośrednictwem złącza ISP. Mam jednak nadzieję, że wykorzystacie zbudowany programator nie tylko do programowania PECEL-a, ale także podczas budowy innych układów. Może wtedy okazać się korzystne, aby testowany układ zasilany był z programatora (w każdym razie ja często stosuję tę metodę, wygodną podczas pracy nad kilkoma prototypami naraz). Do złącza CON3 należy wtedy doprowadzić napięcie o wartości +5VDC, którego idealnym źródłem może być np. game port komputera, a w ostateczności dowolny inny zasilacz o podanym napięciu i maksymalnym prądzie dostosowanym do wymagań uruchamianego układu.

Czy wiecie, moi Drodzy, do jakiego etapu pracy doszliśmy w tym momencie? Prawdę mówiąc, zakończyliśmy już budowę minikomputera PECEL i potrzebnego do jego programowania hardware! Pozostał nam jeszcze wprowadzić jeden kabelek do wykonania, ale możemy odłożyć tę pracę na później, do czasu kiedy zajmijemy się komunikacją nawiązywaną przez nasz minikomputer z „dużym“ PC-tem za pośrednictwem portu RS232. Co zatem teraz zrobimy? Powinniśmy zająć się teraz opisem metod programistycznych służących ożywieniu PECEL-a, ale wiem, na co macie bardziej ochotę! Zapewne chcielibyście wypróbować programator i minikomputer i byłoby z mojej strony okrucieństwem, gdybym kazał Wam na to czekać. A zatem, do dzieła!

Wiecie co? Strasznie mi ten artykuł zaczyna się „rozlać“ i mam nadzieję, że

połapiecie się w tych licznych dygresjach! Przecież zanim wykonamy pierwsze próby programowania procesora naszego minikomputera musimy coś zrobić z programem, który jest już umieszczony w jego pamięci. W kicie AVT-2550 dostarczany jest procesor z umieszczonym w jego pamięci EEPROM programem, który dla Was napisałem. Działanie tego programu zostanie opisane w dalszej części artykułu i nieskromnie mam nadzieję, że zyska on Wasze uznanie. Pamięć procesora nie została w jakikolwiek sposób zabezpieczona przed kopiowaniem, a listing programu został opublikowany na internetowej stronie Elektroniki dla Wszystkich (www.edw.com). Program stanowi zatem Waszą niepodzielną własność, ale co zrobić, jeżeli posiadamy tylko jeden, dostarczony w kicie procesor AT90S8535? Jakkolwiek próba programowania procesora spowoduje nieodwołalne zniszczenie zapisanego w jego pamięci „fabrycznego” programu. Arcydzieło sztuki programowania to chyba nie jest, ale może warto go zachować na przyszłość?

Na szczęście mamy już gotowy programator, który bynajmniej nie służy tylko do programowania procesora. Ile użytecznych funkcji może on jeszcze spełniać, dowiecie się w najbliższej przyszłości, a na razie, trochę wbrew logice, zajmijmy się nie programowaniem, ale odczytywaniem programu już zapisanego w pamięci EEPROM procesora.

O instalacji i ogólnym konfigurowaniu pakietu BASCOM AVR nie będę pisał, ponieważ praktycznie nie różnią się one od obsługi znanego już Wam pakietu BASCOM 8051. Wspomnijmy tylko o konfigurowaniu programatora, ponieważ nie mieliśmy z tym jeszcze do czynienia.

Po uruchomieniu BASCOM-a AVR klikamy na pasek OPTIONS i z rozwiniętego menu wybieramy opcję PROGRAMMER. Ukaze się nam wtedy panel pokazany na rysunku 6. W okienku PROGRAMMER wybieramy teraz typ programatora, którym musi być STK200/STK300 Programmer. Następnie zamykamy okienko i naciskamy klawisz F4, co owocuje pojawieniem się okienka programatora pokazanego na rysunku 7. Na wszelki wypadek naciskamy jeszcze na przycisk CHIP, a następnie IDENTIFY. Po tym zabiegu w małym okienku obok napisu CHIP powinien pokazać się napis informujący o typie zidentyfikowanego procesora, czyli w naszym przypadku AT90S8535.

Może się jednak zdarzyć, że programator nie będzie w stanie zidentyfikować typu procesora i na ekranie ukaże się mało sympatyczny napis widoczny na rysunku 8. Jeżeli jesteśmy całkowicie pewni, że montaż minikomputera i programatora przeprowadziliśmy poprawnie, to zapewne przyczyną jest nieprawidłowe podłączenie kabla łączącego programator z minikomputerem. Po sprawdzeniu tego połączenia i ewentualnym od-

wróceniu wtyku o 180 stopni wszystko powinno zacząć działać normalnie.

Zajmijmy się teraz zachowaniem dla potomności programu zapisanego w EEPROM-ie dostarczonego w kicie procesora. Po identyfikacji typu procesora klikamy na pasek CHIP, a następnie wybieramy opcję READ CHIPCODE INTO BUFFER (załaduj kod zawarty w pamięci procesora do bufora). W tym momencie rozpocznie się proces odczytywania zawartości pamięci EEPROM, który niestety potrwa chwilę, no powiedzmy dłuższą chwilę. Tak to już jest: zawsze „coś za coś” i za liczne udogodnienia związane z programowaniem ISP płacimy zwiększonym czasem trwania szeregowej transmisji danych. Ręczę jednak, że to się Wam opłaci!

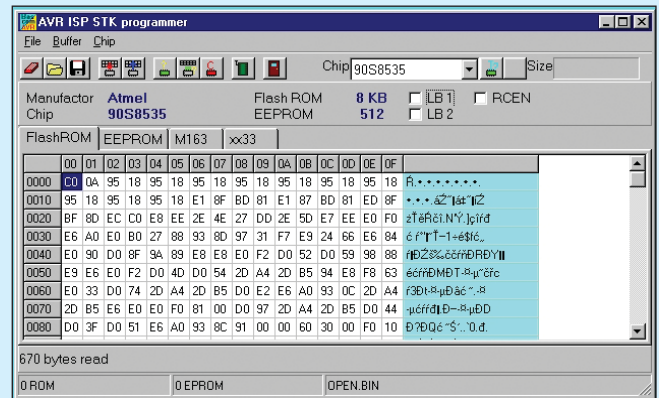
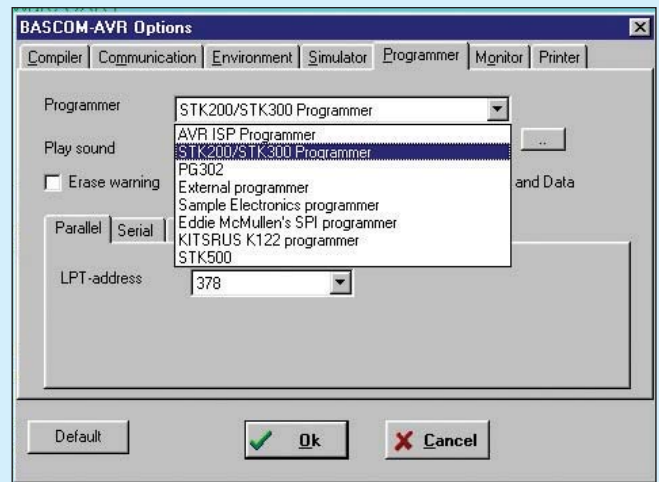
Mam nadzieję, że programator uporał się już z mozolnym odczytywaniem programu z pamięci procesora. A zatem, następną czynnością będzie zapisanie tego programu na dysku, pod dowolną nazwą i w dowolnym, wskazanym katalogu. W tym celu klikamy na pasek BUFFER, wybieramy opcję SAVE TO FILE i podajemy nazwę pliku, w którym ma być zapisany program w formacie binarnym (rysunek 9). Proponuję wykonać przynajmniej jedną kopię zapasową tego pliku i zapisać ją w innym katalogu, niż oryginał.

Nadeszła wreszcie pora, aby sprawdzić działanie zbudowanego układu w jego podstawowej, ale nie jedynej funkcji, jaką jest wprowadzanie programu do pamięci EEPROM minikomputera PECEL. Ponieważ niewiele jeszcze wiemy o programowaniu procesora AT90S8535, wykonamy tylko proste testy, wykorzystując wiedzę nabytą podczas kursu BASCOM College. No, może pokażę Wam coś więcej...

Zanim napiszemy nasz pierwszy program na minikomputer PECEL, musimy jeszcze dokończyć konfigurowanie programatora. W jego okienku konfiguracyjnym zaznaczamy dodatkowo opcje AUTO FLASH i AUTO VERIFY, tak jak pokazano na rysunku 10.

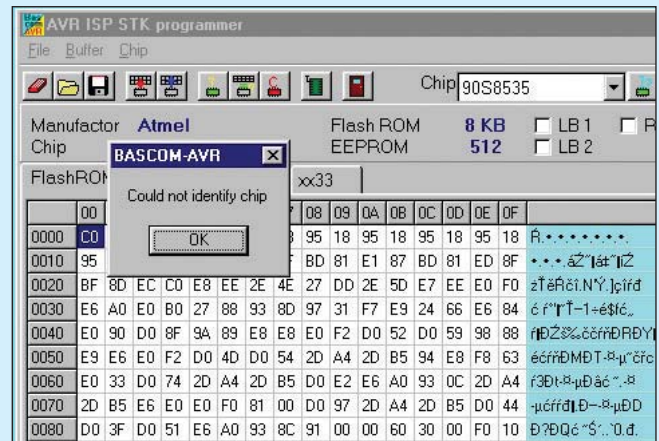
Zacznijmy od czegoś bardzo prostego, pamiętając że mamy tylko przetestować programator, a na naukę programowania PECEL-a przyjdzie czas trochę później. Napiśmy zatem:

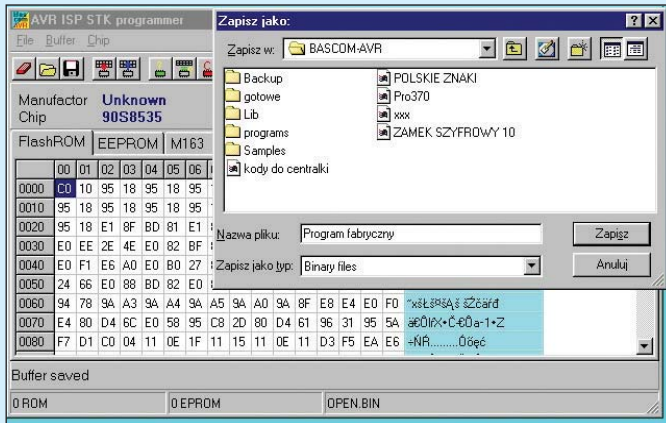
Rys. 6



Rys. 7

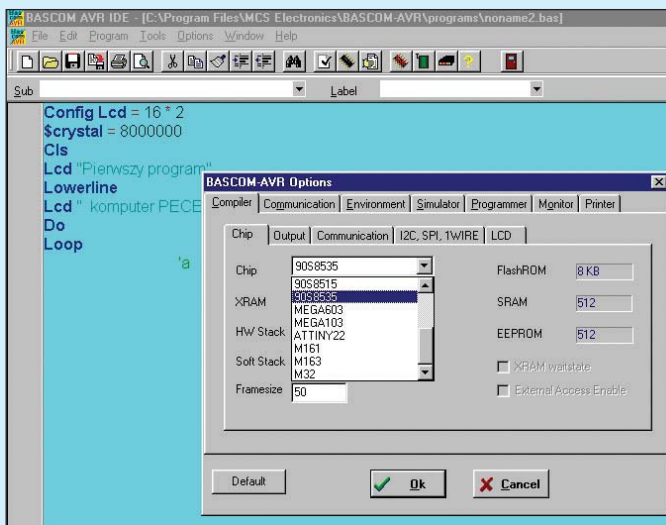
Rys. 8





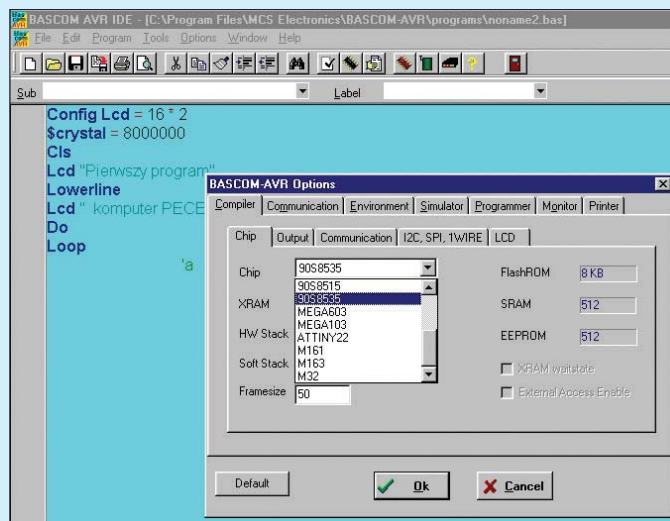
Rys. 9

Rys. 10



programu na PECEL-a możemy zobaczyć na jego wyświetlaczu alfanumerycznym. Nie musieliśmy wyjmować kosztownego i łatwego do uszkodzenia procesora z podstawki, wkładać go do programatora, a następnie ponownie umieszczać w minikom-

Rys. 11



**Listing 1**  
 Config Lcd = 16 \* 2  
 \$crystal = 8000000  
 CIs  
 Lcd "Pierwszy program"  
 Lowerline  
 Lcd "komputer PECEL"  
 Do  
 Loop

Przed kompilacją tego programiku musimy jeszcze poinstruować kompilator, jakiego typu procesora będziemy używać. Otwieramy zatem okienko OPTIONS, a następnie COMPILER i CHIP (rysunek 11) i zaznaczamy procesor typu AT90S8535.

Naciskamy teraz „magiczny” klawisz F7. Dlaczego nadałem mu taki przydomek? Ano dlatego, że po jego naciśnięciu napisany przez nas program został nie tylko skompilowany, ale natychmiast umieszczony w pamięci procesora. Procesor został następnie zresetowany, a efekt działania

puterze. W tym, między innymi, tkwi siła programowania ISP połączona z fantastycznymi możliwościami BASCOM-a! Fajne, prawda?

Najwyższy czas, aby rozpocząć systematyczną naukę programowania minikomputera PECEL. Ale nawet mnie samego korci, aby awansem pokazać Wam jakiś „fajerwerk”, jakiś spektakularny przykład możliwości naszych nowych urządzeń. Wspomniałem uprzednio, że zbudowany przed chwilą programator może służyć do wielu celów, często nie bardzo związanych z samym procesem programowania. Może pamiętacie z kursu BASCOM College lub z własnych doświadczeń, w jaki sposób zapisywaliśmy dane w zewnętrznych pamięciach danych EEPROM? Było z tym trochę problemów, trzeba było napisać kilkanaście linijek programu, nie mówiąc o konieczności dodawania do systemu dodatkowego układu – zewnętrznej pamięci danych EEPROM. No to popatrzcie, jak to wygląda w naszym minikomputerze wspartym siłą BASCOM-a!

Pisząc ostatnie zdanie zauważyłem, że narobiło się trochę bałaganu w stosowanym w artykule nazewnictwie i że początkujący Koledzy mogą mieć z tym trochę kłopotu. Procesor AT90S8535 posiada aż trzy rodzaje pamięci i musimy dokładnie uprzytomnić sobie, do czego każda z nich służy.

1. Pamięć programu EEPROM służy do zapisywania treści programu sterującego pracą procesora. Jej pojemność wynosi 8kB i może być programowana wyłącznie za pomocą zewnętrznego programatora. Jakiegokolwiek zmiany w jej zawartości bez stosowania programatora są niemożliwe. Pamięć programu może być przeprogramowywana do 1000 razy.
2. Pamięć danych EEPROM służy do zapisywania tych informacji, które nie mogą być utracone po wyłączeniu zasilania. Pamięć ta programowana jest przez odpowiednie polecenia obsługującego procesor programu. Istnieje także możliwość zaprogramowania pamięci danych EEPROM za pomocą zbudowanego przed chwilą programatora, a także odczytania jej zawartości. Pamięć danych EEPROM może być przeprogramowywana do 100 000 razy.

3. Pamięć danych RAM służy do chwilowego przechowywania danych, a jej zawartość jest bezpowrotnie tracona po wyłączeniu zasilania.

## Wykaz elementów

### Programator ISP

#### Kondensatory

- C1 .....100nF
- C2 .....100µF/10V

#### Rezystory

- R1 .....100kΩ

#### Półprzewodniki

- IC1 .....74LS244

#### Pozostałe

- Wtyk DB15M + obudowa
- Odcinek ok. 50 cm dziesięciożyłowego przewodu taśmowego
- 2 wtyki zaciskane 10 pin goldpin kątowny 3x2

Komplet podzespołów z płytką jest dostępny w sieci handlowej AVT jako kit szkolny AVT-2550/P

Do napisanego uprzednio programu dopiszmy trzy linijki, tak aby całość wyglądała tak, jak na listingu 2 (dodatkowe linie zaznaczono pogrubionym drukiem). Nie będziemy na razie tłumaczyć sobie znaczenia nowych poleceń i wspomnę tylko, że dodatkowym zadaniem programu jest teraz zapisanie w wewnętrznej pamięci danych EEPROM pod adresem 1 wartości zmiennej X, czyli 214.

**'Listing 2**

```
Config Lcd = 16 * 2
Crystal = 8000000
Dim X As Byte
Cls
Lcd "Pierwszy program"
Lowerline
Lcd "komputer PECEL"
X = 214
Writeeprom X, 1
Do
Loop
```

Ponownie naciskamy magiczny klawisz i... właściwie nic nowego się nie stało. Napis został wyświetlony, ale czy wartość zmiennej rzeczywiście znalazła się w pamięci? No, to popatrzcie teraz, jak wygodne narzędzia dostaliście do ręki i jak w przyszłości ułatwią nam one testowanie napisanych programów. Rzeczywiście, kontrolowanie, czy dane są zapisywane i czy lokowane są pod takim adresem, pod jakim byśmy chcieli, nie jest sprawą prostą. Podczas posługiwania się „normalnym” oprogramowaniem znalezienie ewentualnych błędów może „trochę” potrwać i kosztować „trochę” nerwów. Przecież pamięci danych EEPROM nie można zobaczyć! Nie można? W BASCOM-ie wszystko można!

Zmieńmy teraz trochę konfigurację BASCOM-a, usuwając zaznaczenie opcji AUTO FLASH w okienku konfiguracyjnym programatora. Naciśnijmy następnie klawisz F4, co zaowocuje pojawieniem się na ekranie okienka programatora. No i co w tym nowego? Popatrzcie na **rysunek 12**: mniej więcej w jednej trzeciej wysokości okienka programatora znajdują się dodatkowe przyciski, a wszystko

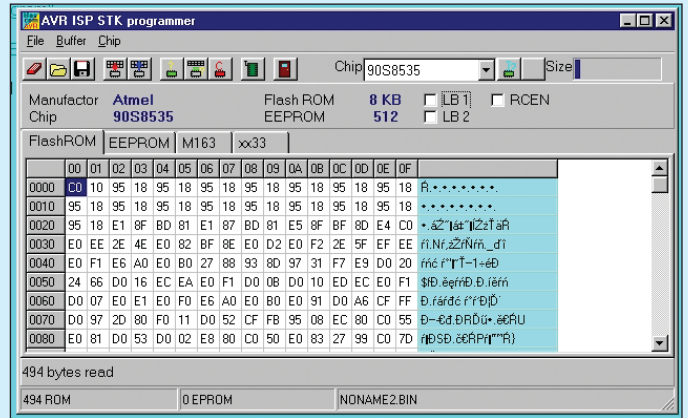
wskazuje, że w obecnej chwili aktywny jest pierwszy z nich, oznaczony jako FLASHROM. Naciśnijmy zatem drugi z przycisków, ten, na którym widnieje napis EEPROM, słusznie przypuszczając, że może on mieć coś wspólnego z wewnętrzną nieulotną pamięcią danych EEPROM.

Rzeczywiście, wydarzyło się coś nowego: na ekranie pojawiła się nowa tabelka, tylko że w niej zapisane są same wartości FF(HEX), czyli dziesiętnie 255. Do czasu! Powtórzmy teraz operację, którą wykonywaliśmy podczas kopiowania „fabrycznego” programu PECEL-a, czyli klikamy na pasek CHIP, a następnie wybieramy opcję READ CHIPCODE INTO BUFFER.

Tylko że tym razem do bufora ładowana będzie nie zawartość pamięci programu, ale interesująca nas pamięć danych EEPROM!

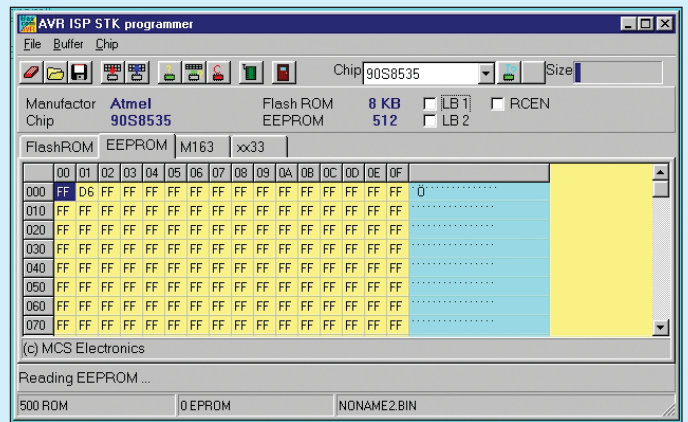
Efekt naszych poczynań jest widoczny na **rysunku 13**. No i co? Nie można zobaczyć pamięci danych EEPROM? Sprawdźmy jeszcze: pod adresem 1 widoczna jest tam wartość D6h, zapisana w formacie heksadecymalnym. Po przeliczeniu na format dziesiętny mamy: D6(HEX) = 214(DEC).

Chciałbym jeszcze na chwilę powrócić do charakterystyk różnych pamięci, jakimi dysponuje procesor AT90S8535, serce komputera PECEL. Zgodnie z danymi zawartymi w karcie katalogowej tego układu podałem, że pamięć programu może być zapisywana do 1000 razy, a pamięć danych EEPROM do 100000 razy. Są to liczby ogromne i trudno chyba obawiać się ich przekroczenia. Jednak w przypadku, kiedy PECEL byłby uży-



Rys. 12

Rys. 13



REKLAMA . REKLAMA . REKLAMA

**ZAKŁAD PRZEZWOŁÓW INDUKCYJNYCH** tel./fax (0-48) 874 31 37  
**INDEL** Sp.z o.o. (0-48) 874 21 28 (0-48) 874 32 27  
 98-068 BRZEZINY ul. Piłsudskiego 29 centr. (0-48) 874 31 48  
 http://www.indel.pl; e-mail:handelowy@indel.pl kartowizna tel./fax (0-22) 669-93-37

**TRANSFORMATORY 0,5 ÷ 5000VA**

DO ZASILANIA UKŁADÓW ELEKTRONICZNYCH,  
 DO OŚWIETLENIA HALOGENOWEGO,  
 BEZPIECZYSTWA,  
 SEPARACYJNE,  
 GŁOSNIKOWE,  
 DŁUCZOWE,  
 CENKI I DRABKI,  
 AUTOTRANSFORMATORY.

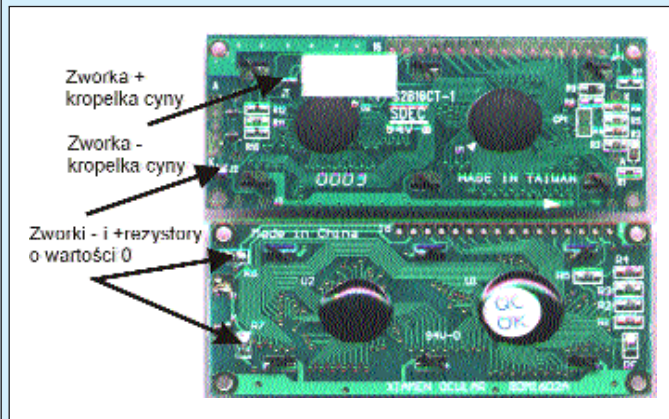
wany przez grupę użytkowników, np. na zajęciach w Technikum Elektronicznym, można obawiać się przekroczenia liczby dozwolonych programowań pamięci programu. Z doświadczenia jednak wiem, że dane podane przez producenta zostały obliczone mocno „na wyrost”, najprawdopodobniej z uwzględnieniem najbardziej krytycznych warunków pracy procesora. Nie testowałem nigdy, jaką maksymalną liczbę cykli zapisu może wytrzymać pamięć programu, ale dokonałem barbarzyńskiego eksperymentu z pamięcią danych EEPROM. Napisałem program, którego jedynym zadaniem było nieustanne zapisywanie całego obszaru tej pamięci coraz to nowymi danymi. Każda operacja zapisu była zliczana a wynik przekazywany do komputera. I wicie, co się zdarzyło? Po 324 567 cyklach zapisu dałem sobie spokój z dalszym prowadzeniem eksperymentu, uznając procesor ATMEL-a za produkt najwyższej klasy, a dane podawane przez tę firmę za więcej niż wiarygodne.

Musimy jednak skończyć z tym chaotycznym działaniem i licznymi dygresjami. Rozpoczynamy systematyczną naukę programowania minikomputera PECEL, a tym samym wszystkich procesorów AVR, poszerzając przy okazji nasze wiadomości o chipach z rodziny '51.

**Zbigniew Raabe**  
 zbigniew.raabe@edw.com.pl

Chciałbym przy okazji rozwiać pewne wątpliwości, które jak wiem z listów e-mailowych nurtują od dawna Czytelników EdW i EP. Ponieważ na podobne trudności mogą natrafić także użytkownicy minikomputera PECEL, chciałbym do nich wyrazić nadzieję, że będą także w nich pewnych, zresztą drobnych, zlokalizować tę zworkę, która odcina zasilanie do podświetlenia wyświetlacza od strony tylnej płytki.

W ofercie handlowej AVT znajdują się najpopularniejsze wyświetlacze LCD 16\*2 i 16\*1 z podświetleniem. Takie wyświetlacze których wlotowanie przesądziło o permanentnym włączeniu podświetlenia! Elementami tymi mogą być "niby rezystory SMD", czyli przewidywane części do budowy naszego minikomputera. Problem, na jaki napotkali Czytelnicy po prostu zworki lub...



W tym bardziej atrakcyjna, że będziemy mogli dokonywać tych operacji programowo, a nawet płynnie regulować jasność świecenia wyświetlacza. A zatem, musimy najpierw zlokalizować tę zworkę, która odcina zasilanie do podświetlenia wyświetlacza od strony tylnej płytki. Czynność tę najprościej wykonać za pomocą omiarmy, jeszcze przed wylutowywaniem jakichkolwiek elementów z płytki wyświetlacza. Następnie wylutować z płytki zworkę i podłączyć ją do zasilania (+5VDC - nóżka 2, masa - nóżka 1). Po włączeniu zasilania wyświetlacz nie powinien dawać żadnych "znaków życia", co jest zjawiskiem prawidłowym. Nie sądzę, aby komukolwiek udało się wyprowadzić z wyświetlacza z masą, co powinno spowodować trwałe włączenie podświetlenia. Jeżeli tak się stanie, to może to oznaczać, że wylutowaliśmy niewłaściwą zworkę. Jeżeli jednak podświetlenie wyłączyło się, to możemy przeprowadzić do dalszego etapu montażu komputerka. Jest to możliwe włączenia i wyłączenia podświetlenia w zależności od aktualnych potrzeb.

**Zbigniew Raabe**

zbignew.raabe@edw.com.pl