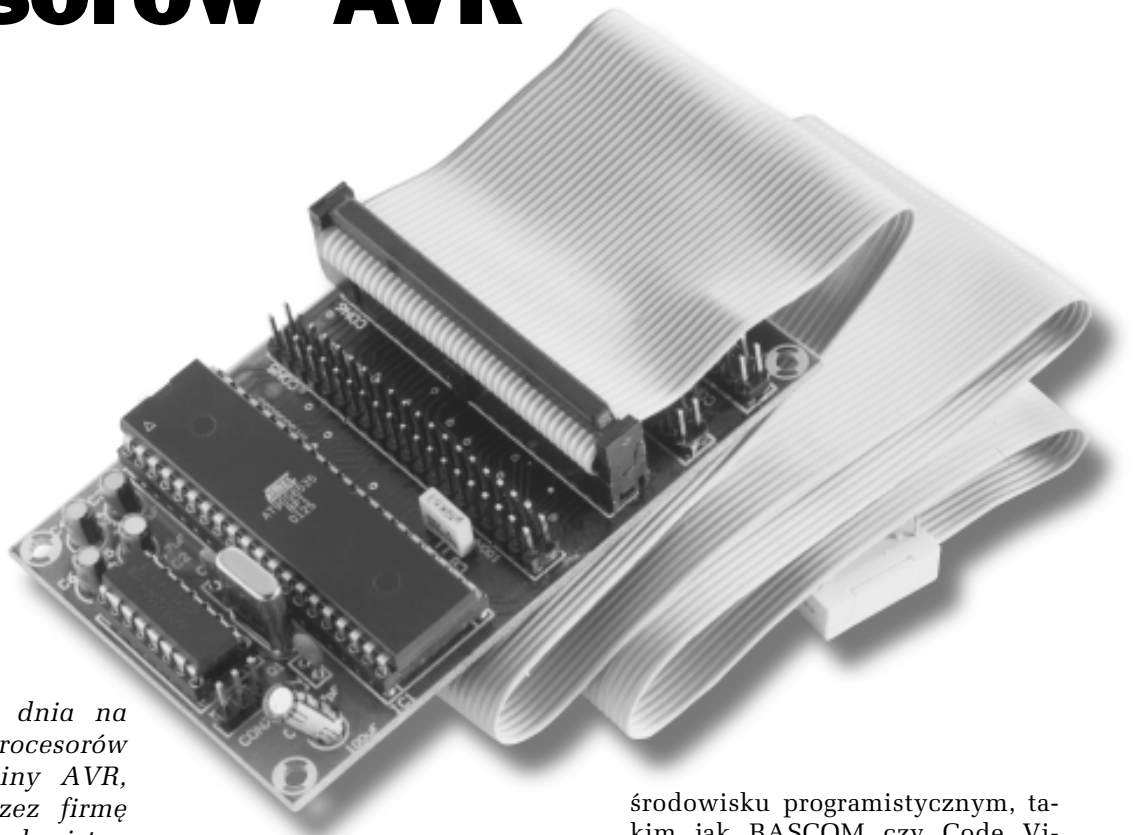


# Sprzętowy emulator procesorów AVR

## AVT-5039



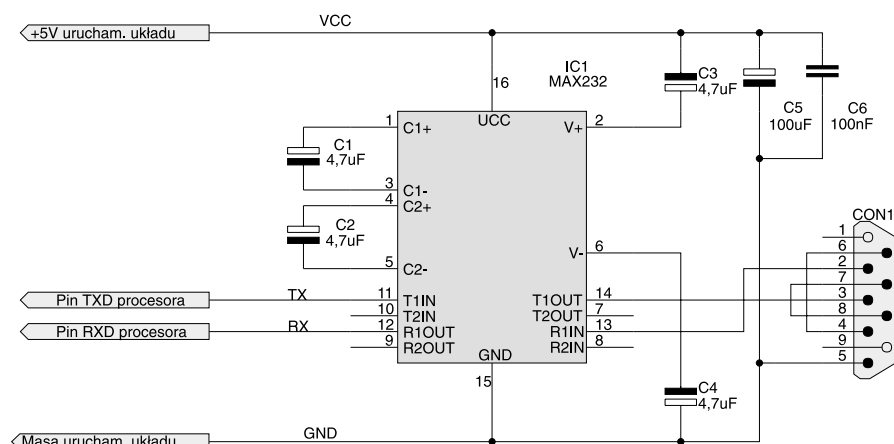
*Wzrastająca z dnia na dzień popularność procesorów RISC z rodziny AVR, produkowanych przez firmę ATMEL, spowodowała istną lawinę coraz to nowszych narzędzi programowych i sprzętowych wspomagających opracowywanie programów dla tych procesorów. Na łamach EP opisałem już kilka programatorów i zestawów uruchomieniowych, ale do pełni szczęścia brakowało - jak do tej pory - jedynie jakiegoś prostego emulatora sprzętowego, umożliwiającego testowanie napisanego oprogramowania w środowisku, dla którego zostało przeznaczone.*

Każdy elektronik wie, jak bardzo dobry emulator sprzętowy może ułatwić pracę i przyspieszyć tworzenie nowego oprogramowania. Jednak skonstruowanie naprawdę dobrego emulatora jest zadaniem trudnym, a układy tego typu są zazwyczaj dość skomplikowane i kosztowne. Dlatego też, pomimo że zdawałem sobie sprawę z konieczności zaprojektowania emulatora sprzętowego dla procesorów AVR, odwlekałem dość długo rozpoczęcie prac nad takim urządzeniem. Wszystkie procesory z rodziny AVR mogą być programowane w systemie ISP (*In System Programming*), co powoduje, że stosowanie emulatorów jest wprawdzie pożądane, lecz nie absolutnie konieczne. Każdy procesor AVR można zaprogramować bezpośrednio w jego docelowym środowisku bez konieczności wyjmowania z podstawki, umieszczania w programatorze i ponownego instalowania w systemie. Jeżeli przy tym pracujemy w dobrze opracowanym

środowisku programistycznym, takim jak BASCOM czy Code VisionC, to zaprogramowanie procesora i przygotowanie układu do przeprowadzenia stosownych testów sprowadza się najczęściej do jednego kliknięcia myszką lub naciśnięcia jednego klawisza. Wynika z tego, że emulator sprzętowy dla procesorów AVR, będący komfortowym, ale nie nieodzownym uzupełnieniem zestawu narzędzi pracy, nie może być układem nadmiernie skomplikowanym ani kosztownym.

Rozwiązanie problemu przyszło nagle i z pokorą muszę przyznać, że bynajmniej nie jest ono moją zasługą. Program monitora, umożliwiający emulację procesorów AVR za pomocą procesorów AVR, powstał w niezawodnej jak zwykle firmie MCS Electronics, został napisany przez mojego Przyjaciela, pana Marka Albertsa. Program ten umożliwił zbudowanie emulatora o dobrych parametrach, a do tego niezwykle prostego i łatwego do wykonania.

Mogę zaproponować dwie metody przeprowadzania emulacji sprzętowej procesorów AVR. Pier-



Rys. 1. Schemat elektryczny konwertera napięć RS232/TTL.

wsza z nich, nie wymagająca prawie żadnego dodatkowego sprzętu, polega na emulacji pracy procesora za pomocą procesora tego samego typu umieszczonego w przeznaczonym dla niego środowisku. Jest to rozwiązanie bardzo proste, a do jego realizacji będziemy potrzebować jedynie prostego układu, którego zadaniem będzie konwersja poziomów napięć występujących na łączy RS232 na poziomy napięcia TTL i odwrotnie. Taki układ konwersji najprościej wykonać z wykorzystaniem popularnego układu MAX232. Schemat konwertera został pokazany na rys. 1. Niestety, taka metoda realizacji emulacji sprzętowej niesie ze sobą jedno, poważne ograniczenie: wyprowadzenia TxD i RxD procesora nie mogą być wykorzystywane do niczego innego, za wyjątkiem obsługi transmisji RS232.

Znacznie doskonalsze rozwiązanie polega na zbudowaniu uniwersalnego sprzętowego emulatora procesorów AVR. Układ taki został przeze mnie skonstruowany w oparciu o jeden z „najwięk-

szych“ procesorów AVR - AT90S8535. Zastosowanie go pozwoli na testowanie układów zawierających taki właśnie procesor lub dowolny inny z rodziny AVR posiadający równą lub mniejszą liczbę wyprowadzeń.

Za pomocą tego układu emulatora możemy symulować pracę następujących typów procesorów z rodziny AVR: AT90S2313, AT90S2343, AT90S8515, AT90S8535, AT90S4433, AT90S4433, AT90S8535, AT90S4433, AT90S2323, AT MEGA161, AT MEGA16 i AT MEGA32. Nie jest natomiast możliwe testowanie układów zbudowanych na 64-pinowych procesorach z serii AT MEGA.

Proponowany układ emulatora sprzętowego jest urządzeniem bardzo prostym, zbudowanym z wykorzystaniem zaledwie dwóch układów scalonych. Jednak za tę prostotę i niewielki koszt wykonania urządzenia musimy zapłacić nie największą szybkością pracy emulatora. Ta właśnie wada uniemożliwia symulację niektórych czynności, wymagających bardzo precyzyjnego timingu (zestawienie możliwych do emulowania funkcji zamieszczono w tab. 1).

### Opis działania układu

Schemat elektryczny układu emulatora pokazano na rys. 2. Wygląda on pozornie na skomplikowany, ale jeżeli przyjrzymy mu się bliżej, to z pewnością zauważymy, że cała praca część schematu to tylko złącza, dla jasności przedstawione wraz z opisem wyprowadzeń obsługiwanych przez nie procesorów.

„Serdecem“ emulatora jest zaprogramowany procesor typu

AT90S8535, jeden z najbardziej rozbudowanych układów z rodziny AVR. Wybór procesora został jednak podyktowany nie przez chęć skorzystania z jego rewelacyjnych możliwości. Rzeczywistym powodem była liczba dostępnych wyprowadzeń, która umożliwia emulację praktycznie wszystkich procesorów AVR. Na list. 1 pokazano cały program sterujący pracą procesora, wraz z dyspozycjami konfiguracyjnymi i komunikatami wstępnymi, które mogą być wyświetlone po podłączeniu układu do monitora złącza RS232. Program, którego treści (nie będąc jego autorem) nie komentuję, jest niezwykle prosty i zajmuje w pamięci procesora zaledwie ok. 1kB. Może to sprawiać wrażenie, że ogromne możliwości '8535 zostały w tym układzie wykorzystane tylko w niewielkim stopniu. Tak jest w istocie, procesor spełnia tu bardzo proste funkcje i można nawet powiedzieć, że się „marnuje“. Myślę jednak, że wartość użytkowa emulatora jest tak duża, że budując go możemy pozwolić sobie na taką rozrzutność i zastosować w nim nawet drogi i nie w pełni wykorzystany element. Ponadto, chciałbym zaprezentować Czytelnikom swój sposób widzenia nowoczesnych i - co bardzo ważne - programowalnych w systemie procesorów. Przecież taki chip może pełnić bardzo użyteczną funkcję, o jakiej zapew-

```
List. 1.
$regfile = "8535def.dat"
$crystal = 11059200
$baud = 115200
Dim Krk As Byte
Dim Adr As Word
Dim Adr1 As Byte, Adr2 As Byte
Dim V1 As Byte

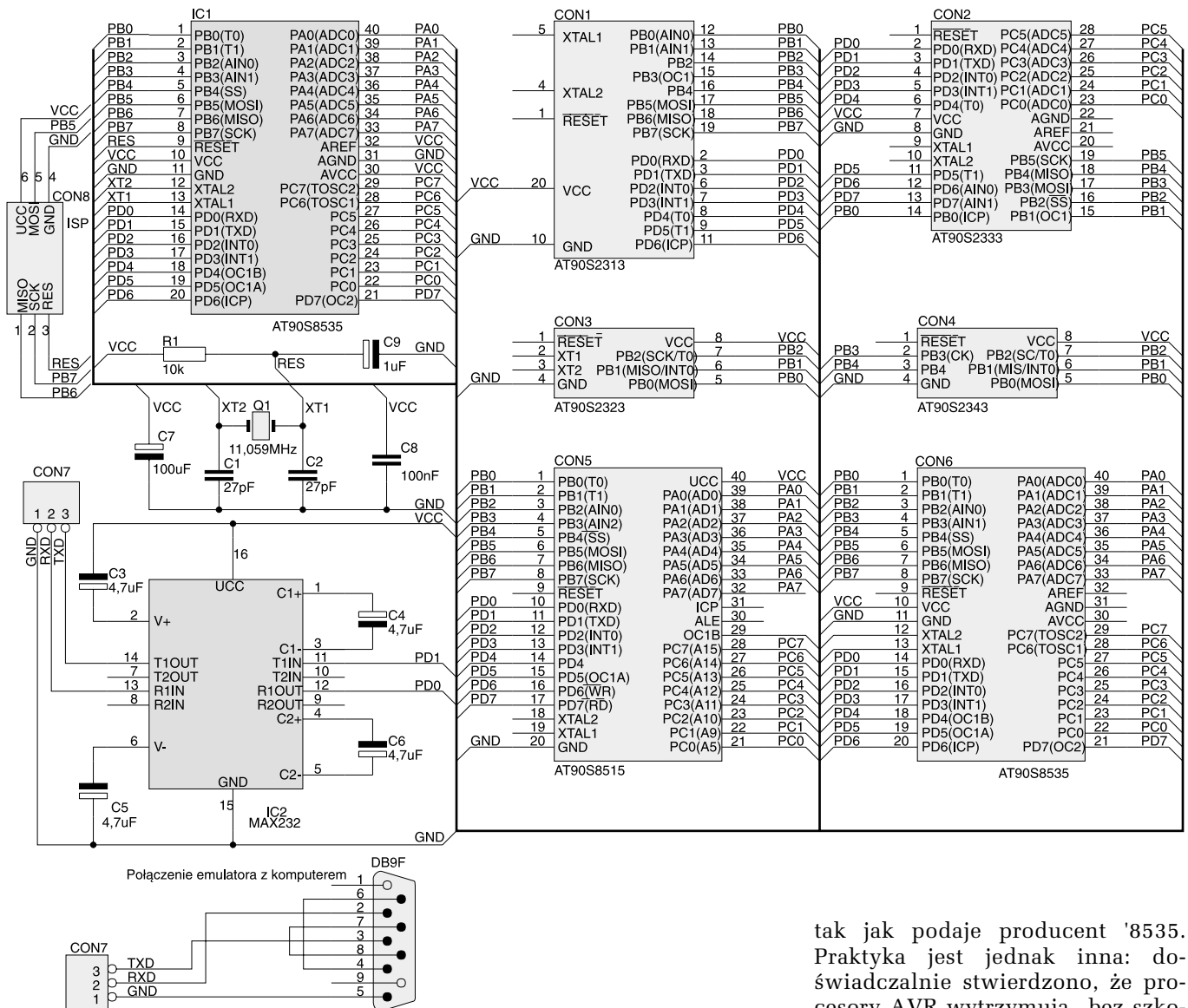
Print "Emulator sprzętowy procesorów AVR"
Print "*****AVT - 5039*****"
Print "Ustaw szybkość transmisji na 115200 Baud"
Print "Emulacja procesorów następujących typów:"
Print "AT90S2313, AT90S2323, AT90S8515"
Print "AT90S2343, AT90S8535, AT90S4433"
Print "i pozostałych z rodziny AVR, z wyjątkiem AT MEGA"

Do
  Krk = Inkey()
  If Krk = "T" Then
    Print Chr(13);
  ElseIf Krk = "W" Then
    Adr = Waitkey()
    V1 = Waitkey()
    Out Adr, V1
    Print Chr(13);
  ElseIf Krk = "R" Then
    Adr = Waitkey()
    V1 = Inp(adr)
    Print Chr(v1);
  ElseIf Krk = "O" Then
    Adr1 = Waitkey()
    Adr2 = Waitkey()
    V1 = Waitkey()
    Adr = Adr1 * 256
    Adr = Adr + Adr2
    Out Adr, V1
    Print Chr(13);
  End If
Loop
```

Tab. 1.

Funkcja	Możliwość realizacji
Transmisja danych poprzez magistralę I <sup>2</sup> C	Tak
Transmisja danych magistralą 1WIRE	Nie
Zegar czasu rzeczywistego	Nie
Obsługa klawiatury szesnastkowej (polecenie GETKBD)	Tak
Odbiór kodu RC5	Nie
Operacje na portach i pinach portów	Tak
Obsługa wyświetlacza LCD	Tak
Transmisja RS232	Tak*
Zapis i odczyt z pamięci danych EEPROM	Tak
Obsługa przetworników AD	Tak
Obsługa przerwań zewnętrznych	Tak
Multipleksowanie wyświetlaczy LED	Tak

\* transmisja danych poprzez RS232 jest dodatkowo symulowana programowo



Rys. 2. Schemat elektryczny emulatora.

ne nie myśleli jego twórcy z firmy ATMEL. Może on być czymś w rodzaju uniwersalnego, reprogramowalnego przyrządu laboratoryjnego. Programowanie procesorów AVR trwa dosłownie sekundy, a budowa potrzebnego do tego celu programatora jest banalnie prosta (np. programator AVT-871). Załóżmy, że w naszej pracy wykorzystujemy miernik częstotliwości AVT-5034 zbudowany także z wykorzystaniem procesora '8535. Jeżeli w jakimś momencie okaże się, że musimy skorzystać z emulatora sprzętowego AVR, to po prostu reprogramujemy procesor umieszczony w mierniku (wykorzystując wbudowany w ten układ interfejs ISP) i umieszczamy go w emulatorze. Po zakończeniu prac związanych z emula-

cją procesorów AVR umieszczamy procesor z powrotem w mierniku i ponownie programujemy go właściwym dla tego przyrządu programem. Taki cykl reprogramowania procesora możemy teoretycznie wykonać 1000 razy,

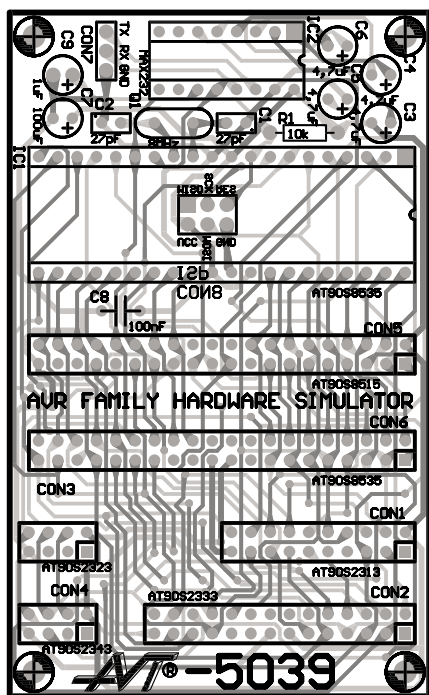
tak jak podaje producent '8535. Praktyka jest jednak inna: doświadczalnie stwierdzono, że procesory AVR wytrzymują „bez szkody dla zdrowia“ znacznie większą liczbę programowań.

Analizując schemat pokazany na rys. 2, wielu Czytelników zauważyło już z pewnością drastyczne odstępstwo od reguł projektowania układów z procesorem '8535. Już po raz drugi występuję w roli prowokatora lekkomyślnie

Tab. 2.

Kwarc	1000000Hz	4000000Hz	7372800Hz	8000000Hz	11059200Hz
<b>Baudrate</b>	<b>Zgodność uzyskanej szybkości transmisji z zadaną</b>				
2400	0,2%	0,2%	0,0%	0,2%	0,0%
4800	0,2%	0,2%	0,0%	0,2%	0,0%
9600	7,5%	0,2%	0,0%	0,2%	0,0%
14400	7,8%	2,1%	0,0%	0,8%	0,0%
19200	7,8%	0,2%	0,0%	0,2%	0,0%
28800	7,8%	3,7%	0,0%	2,1%	0,0%
38400	22,9%	7,5%	0,0%	0,2%	0,0%
57600	7,8%	7,8%	0,0%	3,7%	0,0%
76800	22,9%	7,8%	0,0%	7,5%	0,0%
115200	84,3%	7,8%	0,0%	7,8%	0,0%

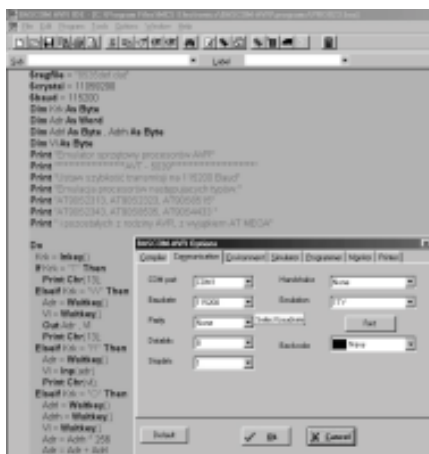




Rys. 3. Rozmieszczenie elementów na płytce drukowanej.

lekceważącego dane zawarte w karcie katalogowej opublikowanej przez Atmel. „Stoi tam jak byk“, że maksymalna częstotliwość taktowania tego procesora wynosi zaledwie 8MHz, a ja zastosowałem rezonator o częstotliwości podstawowej równej aż 11059200Hz. Co było powodem tej, być może ryzykownej decyzji? Otóż, była nią chęć uzyskanie jak największej prędkości transmisji danych poprzez interfejs RS232, od której w decydujący sposób zależy także szybkość działania emulatora.

Popatrzmy na tab. 2, w której zawarto dane zaczerpnięte z karty katalogowej procesora '8535.



Rys. 4. Konfiguracja łącza szeregowego w BASCOM-ie.

Z danych zawartych w tabeli wynika niezbicie, że maksymalną szybkość transmisji wynoszącą 115200bd można uzyskać jedynie z rezonatorami kwarcowymi o częstotliwości 7372800Hz i 11059200Hz. Częstotliwość 7372800Hz mieści się w granicach dopuszczalnej częstotliwości pracy procesora '8535, ale rezonatory kwarcowe o takiej częstotliwości podstawowej są dość trudno osiągalne. Kwarc 11059200Hz bez najmniejszego problemu kupimy w każdym sklepie z częściami elektronicznymi i dlatego zdecydowałem się na overclocking procesora. Zabieg ten nie pociągnął za sobą żadnych ujemnych skutków. Podczas długotrwałych testów emulator pracował zawsze poprawnie, a jego szybkość pracy można było uznać za zadowalającą.

Jeżeli jednak ktoś chciałby rygorystycznie przestrzegać zasad podanych w kartach katalogowych Atmela i posiadałby kwarc o częstotliwości podstawowej 7372800Hz, to może go bez problemu wykorzystać w układzie emulatora. Należy jednak pamiętać wtedy o (bezwzględnie koniecznej) zmianie polecenia konfiguracyjnego częstotliwość zegara systemowego. Zamiast:

```

$crystal = 11059200
    piszemy wtedy:
$crystal = 7372800
    
```

Oczywiście, można także zastosować kwarc o innej częstotliwości, np. 8MHz. Spowoduje to jednak znaczne zmniejszenie szybkości pracy emulatora oraz konieczność zmiany nie tylko polecenia konfiguracyjnego częstotliwość oscylatora, ale także określającego szybkość transmisji. Dane potrzebne do jej określenia zawarte są w tab. 2.

Drugim układem scalonym zastosowanym w emulatorze jest popularny konwerter RS233/TTL, czyli znany chyba każdemu elektronikowi MAX232. Za pomocą tego układu, wspieranego przez sprzętowy UART wbudowany w strukturę procesora AT90S8535, możemy nawiązać łączność z dowolnym komputerem wyposażonym w interfejs RS232.

Należy jeszcze wspomnieć o złączu CON8, którego przeznaczenie było jak dotąd zagadkowe. Złącze to nie pełni w układzie żadnej użytecznej funkcji i pier-

**WYKAZ ELEMENTÓW**

**Kondensatory**

- C1, C2: 27pF
- C3..C6: 4,7µF/10V
- C7: 100µF/10
- C8: 100nF
- C9: 1µF/16V

**Rezystory**

- R1: 10kΩ

**Półprzewodniki**

- IC1: AT90S8535
- IC2: MAX232

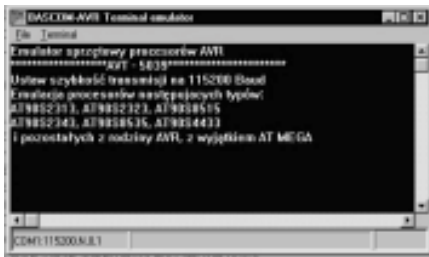
**Różne**

- CON1: 10x2 goldpin
- CON2: 14x2 goldpin
- CON3, CON4: 4x2 goldpin
- CON5, CON6: 20x2 goldpin
- Q1: rezonator kwarcowy 11,059MHz
- Wtyk emulacyjny 20 pin
- Wtyk emulacyjny 40 pin
- Wtyk zaciskany 20 pin
- Wtyk zaciskany 40 pin
- 0,5 m przewodu taśmowego 20 żył
- 0,5 m przewodu taśmowego 40 żył

wotnie miało być wykorzystywane jedynie podczas testowania prototypu. Umożliwia ono dołączenie do układu programatora ISP i przeprogramowywanie procesora bez konieczności wyjmowania go z podstawki. Ponieważ jednak to złącze (ani na schemacie, ani na płytce) w niczym nie przeszkadza, postanowiłem, podobnie jak w przypadku innych opracowanych przeze mnie układów z procesorami AVR, nie usuwać go. Może ono okazać się wielce użyteczne dla tych użytkowników, którzy zechcą napisać własną wersję programu.

**Montaż i uruchomienie**

Na rys. 3 pokazano rozmieszczenie elementów na płytce drukowanej emulatora. Ze względu na znaczną komplikację połączeń płytka została zaprojektowana na laminacie dwustronnym z metalizacją otworów. Montaż układu wykonujemy typowo, rozpoczynając od wlotowania w płytke dwóch podstawek pod układy scalone, a kończąc na kondensatorach elektrolitycznych i złączach do przewodów taśmowych. Pomimo podjętych starań, nie udało mi się jak dotąd „zlokalizować“ wtyków emulacyjnych przeznaczonych dla pro-



Rys. 5. Komunikat powitalny wyświetlany w oknie terminala.

cesorów w obudowach 8-pinowych. Nie jest to jednak większy problem, ponieważ do emulacji tych chipów można z powodzeniem wykorzystywać wtyki 20-pinowe, z których będziemy wykorzystywać tylko 8 końcówek. Z pewnością wtyk 20-pinowy o znacznych wymiarach nie zawsze zmieści się na płytce projektowanej dla procesora z ośmioma wyprowadzeniami, ale i z tym możemy sobie poradzić stosując jedną lub kilka połączonych ze sobą podstawek precyzyjnych, których zadaniem będzie odsunięcie wtyku emulacyjnego od powierzchni płytki. Poważny problem powstanie jedynie podczas przygotowywania wtyku emulacyjnego dla procesorów w obudowach DIL28. Wtyki takie nie są (chyba) produkowane, a zastąpienie ich elementami o innej liczbie pinów nie jest możliwe. Mam jednak wręcz nieograniczone zaufanie do pomysłowości Czytelników EP i mam nadzieję, że jakoś poradzą sobie z tym problemem.

Ostatnią czynnością będzie wykonanie kabla łączącego układ emulatora z komputerem. Czynność tą możemy sobie znacznie ułatwić, jeżeli dysponujemy przewodem od uszkodzonej (że nie powiem „zdechłej”) myszki. Odpadnie nam

wtedy konieczność lutowania złącza DB25, a potrzebne nam przewody będziemy mogli zidentyfikować za pomocą omomierza.

Emulator nie posiada własnego układu zasilającego, a potrzebne do jego działania napięcie pobierane jest z uruchamianego układu mikroprocesorowego.

### Posługiwanie się emulatorem

Wykonany układ emulatora łączymy z komputerem oraz z uruchamianym układem mikroprocesorowym. Jeżeli jeszcze takiego układu nie posiadamy i dopiero mamy zamiar go skonstruować, to do przetestowania poprawności działania emulatora wystarczy, w pierwszym etapie, doprowadzić do niego napięcie zasilania równe +5VDC. Napięcie to doprowadzamy lutując przewody np. do wyprowadzeń kondensatora C7.

Kolejną czynnością będzie odpowiednio skonfigurowanie środowiska BASCOM-a. Najważniejszą sprawą będzie ustawienie szybkości transmisji na 115200bd. Parametr ten ustawiamy w okienku `OPTIONS>COMUNICATION>BAUDRATE`, tak jak to zostało pokazane na rys. 4, a następnie z submenu `TOOLS` wybieramy opcję `TERMINAL EMULATOR`. Następnie musimy określić, który port szeregowy będzie wykorzystywany do obsługi emulatora. Większość współcześnie użytkowanych komputerów PC posiada fabrycznie zainstalowane dwa porty szeregowy: `COM1` i `COM2`. Do jednego z nich zazwyczaj jest na stałe dołączona myszka. Drugi port pozostaje najczęściej niewykorzystany i do niego właśnie dołączymy przewód transmitujący dane do i z emulatora. Jednak po uruchomieniu programu może się zdarzyć, że np. myszka umieszczona została w porcie `COM1` i na ten sam port został skonfigurowany monitor. Taka sytuacja prowadzi do natychmiastowego zawieszenia pracy myszy, a my mamy wtedy dwa wyjścia z sytuacji. Możemy przenieść myszkę do drugiego portu i ponownie uruchomić komputer lub wykorzystując tyl-

ko klawiaturę skonfigurować monitor do śledzenia wolnego aktualnie portu.

Mamy teraz możliwość sprawdzenia, czy poprawnie wykonaliśmy wszystkie połączenia. Łączymy port `RS232` komputera z emulatorem, a następnie otwieramy okno terminala interfejsu `RS232`. Musimy teraz wyzerować emulator (najlepiej poprzez zwarcie wyprowadzeń kondensatora C9). Jeżeli wykonaliśmy poprawnie wszystkie połączenia i skonfigurowaliśmy prawidłowo BASCOM-a, to na ekranie terminala powinien ukazać się komunikat powitalny (rys. 5). Od tego momentu emulator jest gotowy do pracy.

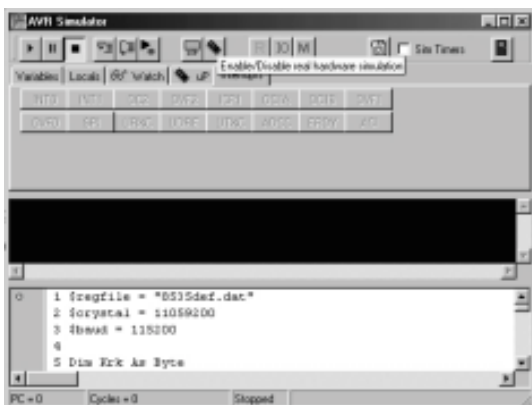
Posługiwanie się emulatorem jest dziecinnie łatwe. Do napisania programu musimy dodać tylko jedno polecenie konfiguracji: `$SIM` umieszczone na początku kodu źródłowego.

**UWAGA:** przed kompilacją programu przeznaczonego już do umieszczenia w procesorze należy polecenie `$SIM` bezwzględnie usunąć!

Następnie program kompilujemy i naciskamy klawisz `F2`, co zaowocuje pojawieniem się na ekranie okienka pokazanego na rys. 6. Należy teraz za pomocą przycisku oznaczonego symbolem układu scalonego uruchomić emulację sprzętową, a następnie rozpocząć wykonywanie programu (przycisk oznaczony strzałką).

**Zbigniew Raabe, AVT**  
**zbigniew.raabe@ep.com.pl**

*Wzory płytek drukowanych w formacie PDF są dostępne w Internecie pod adresem: <http://www.ep.com.pl/?pdf/grudzien01.htm>.*



Rys. 6. Widok okna sterującego pracą emulatora.