

Dział "Projekty Czytelników" zawiera opisy projektów nadesłanych do redakcji EP przez Czytelników. Redakcja nie ponosi odpowiedzialności za poprawność tych projektów, gdyż nie testujemy ich laboratoryjnie.

Prosimy o nadsyłanie własnych projektów z modelami (do zwrotu). Do artykułu należy dołączyć podpisane **oświadczenie, że artykuł jest własnym opracowaniem autora i nie był dotychczas nigdzie publikowany**. Honorarium za publikację w tym dziale wynosi 200,- zł (brutto) za 1 stronę w EP. Przesyłanych tekstów nie zwracamy. Redakcja zastrzega sobie prawo do dokonywania skrótów.

Emulator pamięci EPROM

Obecnie na rynku znajduje się dużo różnych emulatorów pamięci EPROM.

Najczęściej są to skomplikowane urządzenia mogące emulować wiele rodzajów pamięci, posiadające bogate oprogramowanie oraz wiele udogodnień. Dla wielu (szczególnie młodych) projektantów urządzeń wykorzystujących pamięci EPROM, urządzenia takie są za drogie.

Proponowane przeze mnie rozwiązanie jest wynikiem doświadczeń z budową urządzeń wykorzystujących mikroprocesory. Jego największą zaletą jest brak konieczności stosowania specjalnego oprogramowania oraz prostota budowy, co znacznie zmniejsza koszt wykonania. Do obsługi wystarczą systemowe programy dostarczane wraz z systemem DOS tj.: MODE oraz COPY i port szeregowy. Gwarantuje to dużą niezawodność przesyłania danych oraz nie obciąża, często jedyne, portu równoległego, a wykorzystuje najczęściej wolny port szeregowy. Ograniczenie pojemności pamięci do 32KB wynika z wieloletnich doświadczeń, zebranych podczas konstruowania układów mikroprocesorowych. Ponadto przyczyniło się do znacznego uproszczenia układu, a szczególnie płytki drukowanej.

Założenia ogólne

„Sercem“ emulatora jest mikrokomputer jednokładowy firmy ATMEL 89C2051. Pełni on rolę jednostki pośredniczącej w transmisji danych z komputera oraz sterownika wszystkich procesów zachodzących w emulatorze. Dane do emulatora są przesyłane poprzez złącze szeregowe RS232 w ramce informacyjnej, zgodnej ze standardem IntelHex, który jest

standardem zbiorów wyjściowych większości crosassemblerów. Zastosowanie takiej postaci danych zapewnia prostą obsługę emulatora oraz dużą poprawność wpisaných danych. Podstawową prędkością przesyłania danych jest prędkość 1200 bit/s, lecz można ją programowo zmieniać. Do przesyłania jest wymagany pełny (8-żyłowy) przewód modemu zerowego lub przewód trójżyłowy z wykonaną specjalną zworą na wtyczce.

Budowa i działanie układu

Schemat układu jest przedstawiony na rys.1. Dane z komputera docierają do sterownika AT89C2051 (US7) za pośrednictwem popularnego układu MAX232 (US9), z którego doprowadzone są tylko sygnały danych: TxD i RxD. Do pamiętania danych zastosowano układ pamięci statycznej SRAM - 62256 (US1) o pojemności 32KB. Dane wyjściowe zostały wyprowadzone z układu mikroprocesora przez siedem bitów portu P1 oraz jeden bit portu P3 i zostały połączone w odwrotnej kolejności niż wynika to z numeracji wejść układu pamięci. Zostało to podyktowane chęcią uproszczenia połączeń elementów na płycie drukowanej, a tylko nieznacznie skomplikowało program sterujący. Adres

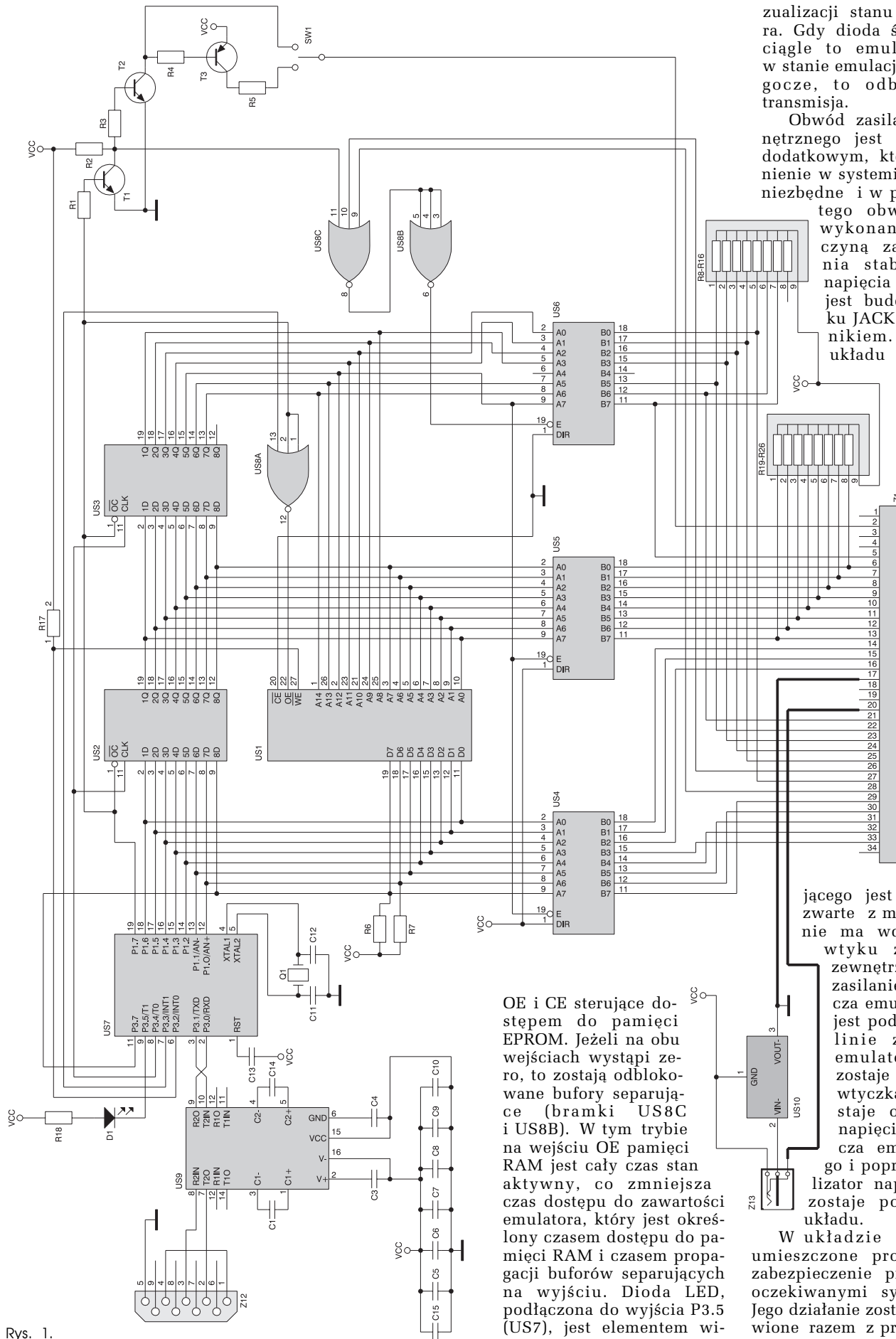
Projekt
038

wykorzystywany podczas zapisu i odczytu danych z pamięci jest wytwarzany w dwóch ośmiobitowych rejestrach 74HC574 (US2 i US3). Zapis do nich odbywa się poprzez przepisywanie bardziej znaczącego bajtu adresu z wyjścia procesora do pierwszego rejestru po wystąpieniu pierwszego ujemnego impulsu na wyjściu P3.4 (US7), a następnie przepisanie tej części adresu do drugiego rejestru z wyjść pierwszego i zapamiętanie mniej znaczącego bajtu adresu w pierwszym rejestrze po podaniu drugiego ujemnego impulsu na P3.4 (US7). Dopiero wtedy na wyjściu procesora pojawiają się dane do zapisu i zostanie wysłany impuls na wyjście P3.2 (US7), który jest sygnałem zapisu do pamięci RAM.

Wyjście P1.7 (US7) steruje trybem pracy emulatora. W stanie niskim (tryb transmisji) zostają odcięte bufory separujące (układy US4, US5, US6), a wyjścia rejestrów adresowych (US2 i US3) zostają otwarte. Dodatkowo sygnał ten steruje wyjściem AUTORESET emulatora, które na czas transmisji przyjmuje stan aktywny, a po zanegowaniu na tranzystorze T1, blokuje (US8C - 74LS27) sygnały odczytu z pamięci RAM docierające z zewnątrz emulatora.

Tryb emulacji (stan wysoki na wyjściu US7 - P1.7) polega na odcięciu wyjść rejestrów adresowych, wprowadzeniu wyjść procesora w stan wysoki, oraz odblokowaniu bramki US8C. Na odblokowaną bramkę dochodzą sygnały





Rys. 1.

zualizacji stanu emulatora. Gdy dioda świeci się ciągle to emulator jest w stanie emulacji, gdy migocze, to odbywa się transmisja.

Obwód zasilania zewnętrznego jest obwodem dodatkowym, którego istnienie w systemie nie jest niezbędne i w prototypie tego obwodu nie wykonano. Przyczyną zastosowania stabilizatora napięcia ujemnego jest budowa wtyku JACK z odłącznikiem. Wyjście układu stabilizującego

OE i CE sterujące dostępem do pamięci EPROM. Jeżeli na obu wejściach wystąpi zero, to zostają odblokowane bufor separujące (bramki US8C i US8B). W tym trybie na wejściu OE pamięci RAM jest cały czas stan aktywny, co zmniejsza czas dostępu do zawartości emulatora, który jest określony czasem dostępu do pamięci RAM i czasem propagacji buforów separujących na wyjściu. Dioda LED, podłączona do wyjścia P3.5 (US7), jest elementem wi-

jącego jest na stałe zwarte z masą. Jeśli nie ma wciśniętego wtyku zasilania zewnętrznego, to zasilanie ze złącza emulacyjnego jest podawane na linię zasilania emulatora. Gdy zostaje wciśnięta wtyczka, to zostaje odłączone napięcie ze złącza emulacyjnego i poprzez stabilizator napięcia 5V zostaje podane do układu.

W układzie zostało umieszczone programowe zabezpieczenie przed nieoczekiwanymi sytuacjami. Jego działanie zostanie omówione razem z programem.

Listing 1.

```

program reset_emulatora;
uses crt;
var
i,j:byte;
port:word;
s:string;
begin
if paramcount>0 then
begin
s:=paramstr(1);
if (s[1]+s[2]+s[3]='com')
or (s[1]+s[2]+s[3]='COM') then
begin
case s[4] of
'1':port:=$3f8;
'2':port:=$2f8;
'3':port:=$3e8;
'4':port:=$2e8;
end;
asm
mov dx,port
add dx,3
in al,dx
or al,40h
out dx,al
end;
delay(5500);
asm
mov dx,port
add dx,3
in al,dx
and al,0bfh
out dx,al
end;
end;
end.

```

Działanie i opis programu sterownika

Dane do programu muszą być zapisane w formacie IntelHex. Format ten można schematycznie przedstawić w postaci:

:*[liczba danych][adresH][adresL][polecenie][dana 1]...[dana ostatnia][CRC]*

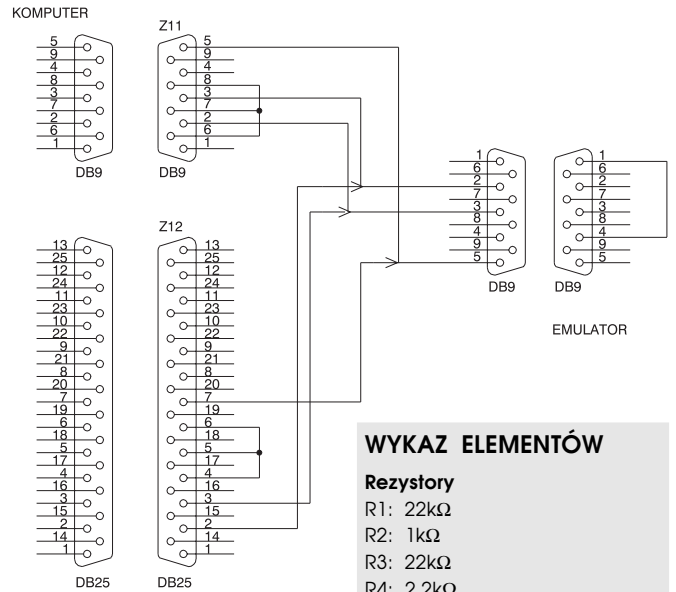
Ramka informacji musi być poprzedzona dwukropkiem. Każde pole ramki zbudowane jest z dwóch znaków ASCII z przedziału (0..9, A..F) określających liczbę w postaci heksadecymalnej, np.: 0= 0 i 0, 10h= 1 i 0, A0h =A i 0.

[Liczba danych] określa ile bajtów zostanie przesłanych w części **[dane J]**. Maksymalna liczba danych wynosi 64 bajty, a wynika to z ograniczonej pojemności pamięci wewnętrznej procesora. **[AdresH]** i **[adresL]** są to bajty odpowiadające starszej i młodszej części adresu, pod który zostaną wpisane (lub odczytane) dane.

Pole **[polecenie]** zostało omówione w **tab.1**. Pole **[CRC]** jest sumą kontrolną. Jego wartość musi równać się dopełnieniu sumy wszystkich bajtów do liczby będącej wielokrotnością 256 (100h). Emulator posiada kilka poleceń, które są wykorzystywane do prowadzenia transmisji danych (tabela 1).

Po ustawieniu rejestrów roboczych program czeka na transmisję z komputera. Jeżeli pojawi się znak i jest nim dwukropek, to układ przechodzi do trybu transmisji. Odbiera po dwa znaki i przetwarza je z postaci znaków ASCII na liczbę, której te znaki odpowiadają, po czym zapamiętuje je w obszarze wewnętrznej pamięci. Po odebraniu i sprawdzeniu poprawności składniowej otrzymanych danych w liczbie określonej polem **[liczba danych]** sprawdzona zostaje suma kontrolna. W przypadku jakiegokolwiek błędu zostaje nadany komunikat o błędzie i układ oczekuje na nową transmisję. Jeżeli wszystko jest w porządku, to następuje zinterpretowanie polecenia.

Zapis do pamięci RAM odbywa się w kilku etapach. Najpierw inicjalizuje się P1.7, który przestawia emulator w tryb transmisji. Na linii P1.0-P1.6 i P3.7 zostaje wysłana starsza część adresu, jest generowany impuls ujemny na linii P3.4, na szynie danych pojawia się młodsza część adresu i jeszcze raz generowany jest impuls na linii P3.4. Dopiero po takiej sekwencji na szynie danych pojawia się właściwa dana do zapisu i generowany jest impuls na linii P3.2 - zapis do pamięci RAM.



Rys. 2.

Zapisane dane są weryfikowane. Odczyt wygląda podobnie jak zapis lecz w ostatniej fazie nie wprowadza się danych ale pobiera się je po podaniu stanu wysokiego na wyjście P3.3. Maksymalną liczbę prób zapisu jednego bajtu określono na 100, lecz gdy pamięć RAM jest sprawna to najczęściej wystarczy jeden wpis, choć zdarzają się przypadkowe przekłamania i stąd tak duża liczba możliwych powtórek.

Podczas zapisu bez kontroli dane są weryfikowane tak samo jak dla wpisu z kontrolą, ale nie jest generowana odpowiedź do komputera. Przy odczycie danych z emulatora, aby odczytać jakąś liczbę danych, to należy wpisać taką samą liczbę danych, jaka ma być odczytana, lub wpisać w pole **[liczba danych]** zero, to sterownik odeśle 64 bajty danych spod adresu określonego w polach **[adresH]** i **[adresL]**. Wbudowane w program zabezpieczenie wykorzystuje przerwanie od Timera T0. Zabezpieczenie to ma dwa poziomy:

1. Zabezpieczenie czasu trwania transmisji - ogranicza czas, w którym emulator odczytuje ramkę informacji po odczytaniu dwukropka do 2 sekund. Po tym czasie program automatycznie przechodzi w stan oczekiwania na następną ramkę informacyjną.
2. Zabezpieczenie niekontrolowanego zachowania - jeżeli przez 5 sekund na

WYKAZ ELEMENTÓW

- Rezystory**
R1: 22kΩ
R2: 1kΩ
R3: 22kΩ
R4: 2.2kΩ
R5: 100Ω
R6: 22kΩ
R7: 22kΩ
R8..16: 8*10kΩ
R17: 1kΩ
R19..26: 8*10kΩ
R18: 510Ω
- Kondensatory**
C1, C3, C4, C14, C5: 1µF/16V
C2, C5, C6, C7, C8: 100nF
C10: 100µF/10V
C11, C12: 33pF
C13: 10µF
- Półprzewodniki**
LED: dowolna dioda
T1: BSXP63
T2: BC107
T3: BC178
US1: M62256
US2: 74HC574
US3: 74HC574
US4: 74HC245
US5: 74HC245
US6: 74HC245
US7: AT89C2051 - zaprogramowany
US8: 74LS27
US9: ICL232
(US10): UL79L05
- Różne**
Q1: kwarc 11,0952 MHz
Z11: złącze IDC34 kątowe do druku
Z12: gniazdo DB9 męskie do druku
(Z13): mono jack z przełącznikiem
Wtyk emulacyjny 28 pin
Krokodyłek w izolacji
Przewód w taśmie 28 żył - 25 cm
Przełącznik dwupozycyjny
Wtyk IDC 34 zaciskany na przewodach
Podstawka 28 pin pod pamięć
Podstawka 20 pin pod procesor i układy scalone

Tabela 2.	
Prędkość	Ramka wpisywana do emulatora
2400	:02000010F400FA
4800	:02000010FA00F4
9600	:02000010FD00F1
19200	:02000010FD8071

wejściu RxD emulatora będzie występował niski stan logiczny, to program wyzeruje sterownik. Prędkość transmisji i stan wyjść będą takie, jak po włączeniu zasilania.

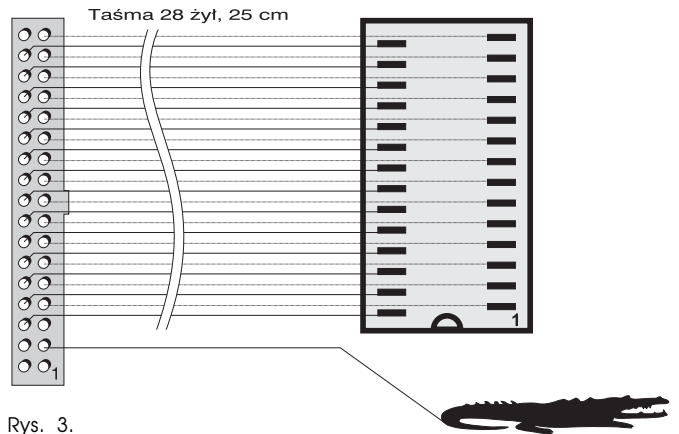
Najczęściej wykorzystywane polecenia emulatora to polecenie 00h i 01h. Pozostałe polecenia są wykorzystywane podczas serwisu emulatora lub w specjalnych programach. Polecenia 00 i 10h mogą być nadane tylko osobno. Resztę poleceń można łączyć przez sumowanie i tak np. polecenie 02 i 04 daje w efekcie zapis i jednoczesną weryfikację zapisanych danych przez ich odczyt. Polecenie 10h jest przeznaczone do przyspieszenia transmisji danych. Prędkość 1200 bit/s jest prędkością podstawową. Przy większych prędkościach mogą występować przekłamania. Zależy to od jakości wykonania przewo-

dów łączących, długości doprowadzeń oraz obciążalności wyjść RS232.

W tabeli 2 widać, że pierwsza wartość dla prędkości 2400 wynosi F4h, a druga 00. Pierwsza jest wartością wpisywaną do rejestru TH1 licznika 1 procesora, a druga wartością wpisywaną do rejestru PCON. Liczba F4h odpowiada podzielnikowi dla częstotliwości 11,0952 MHz.

Użytkowanie układu

Przy wyłączonym zasilaniu układ należy podłączyć do podstawki, w której normalnie pracuje EPROM oraz do złącza portu szeregowego. Końcówkę zaopatrzoną w krokodyłek można (lecz nie jest to konieczne) dołączyć do nóżki kondensatora lub bezpośrednio do nóżki RST procesora. Wyjście AUTORESET jest typu otwarty kolektor zarówno dla dodatniego jak i ujemnego impulsu zerującego. W pliku AUTOEXEC.BAT należy umieścić lub wpisać w trybie bezpośrednim komendę MODE COMn 1200 n 8 1, która ustawi port szeregowy w określony tryb pracy. Wyśłać do portu szeregowego



Rys. 3.

zbiór w formacie IntelHex poprzez standardowe DOS-owe komendy np.: COPY xx.hex COMn /b, przy czym XX.hex jest nazwą zbioru, a n jest numerem portu szeregowego, do którego wysyłamy dane. Opcja /b powoduje, że dane nie są przetwarzane przez komendę COPY i są bez zmiany wysyłane na wyjście portu bajt po bajcie. Aby wykorzystać w trybie bezpośrednim inne prędkości transmisji należy (przykład dla prędkości 2400 bit/s i portu COM2) utworzyć plik tekstowy o nazwie 2400.hex:
:02000010F400FA

:00000001FF
oraz plik wsadowy: 2400.bat
mode com2 1200 n|8|1
copy 2400.hex com2 /b
mode com2 2400 n|8|1
copy %1 com2 /b
Po wykonaniu pliku wsadowego 2400.bat z nazwą naszego zbioru do wpisania jako parametrem, do emulatora są wpisywane dane z prędkością 2400 bit/s. Aby wyzerować emulator po jakimś błędzie wpisu do emulatora lub innym zdarzeniu, po którym straciliśmy łączność z emulatorem, należy uruchomić program w Pascalu z list.1.
Wojciech Werwiński