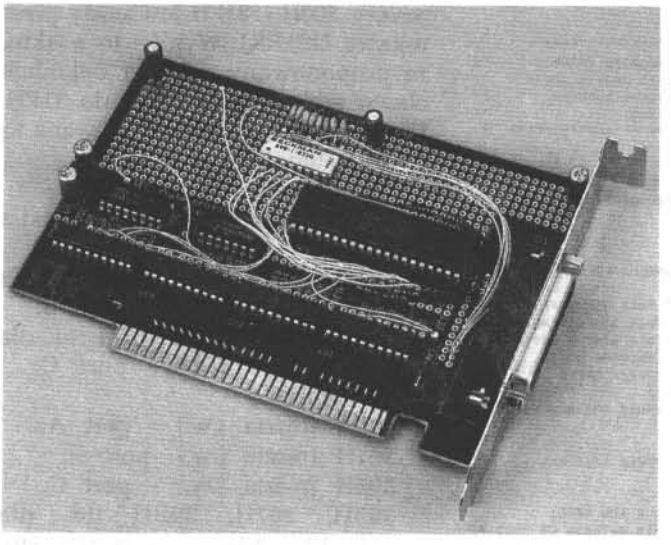
Uniwersalne, równoległe we/wy cyfrowe do PC

druga aplikacja kitu AVT-114

kit AVT-114/2

Proponowana karta uniwersalnego wejścia - wyjścia równoległego do komputera PC może znaleźć wiele zastosowań w układach sterowania i nadzoru różnego rodzaju procesów (analogowych lub cyfrowych) lub jako dwukierunkowe, uniwersalne 24-bitowe złącze równoległe do transmisji danych. Egzemplarz wzorcowy wykorzystywany był jako tester do układów cyfrowych TTL i CMOS.

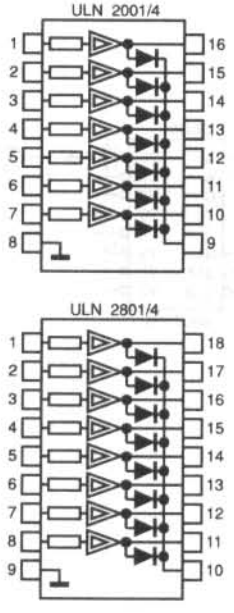


Ze względu na prostotę rozwiązania nie zaprojektowano specjalnej płytki drukowanej, montaż przewidziano na płytce karty prototypowej (kit AVT-114).

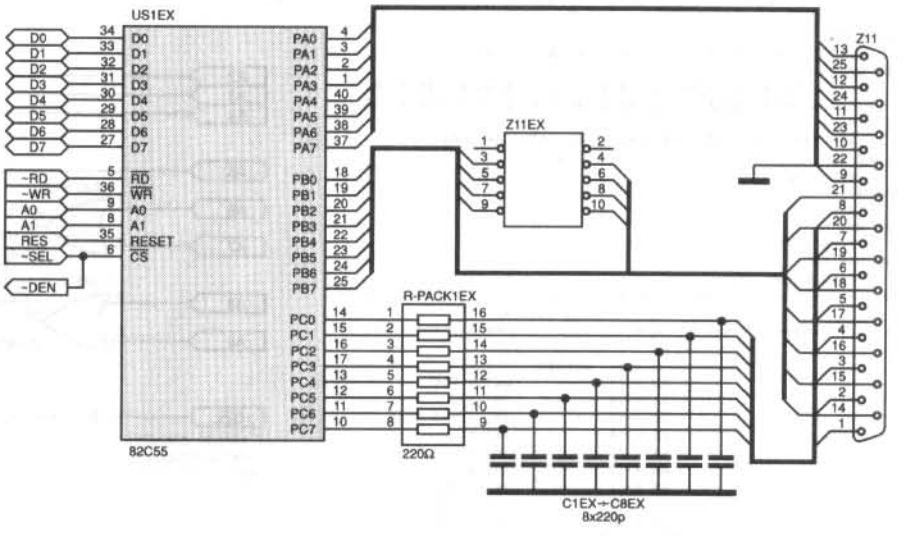
Wykonane zostały dwie wersje - uproszczona z wyjściami małej mocy (bezpośrednio z układu 82C55) i nieco bardziej skomplikowana z driverami firmy SGS ULN2003 (zawiera 7 driverów 500mA) oraz ULN2803 (zawiera 8 driverów 500mA). Na **rys. 1** są przedstawione wyprowadzenia tych układów.

Opis układu

Na schemacie (**rys. 2**) przedstawiono prostszą, przeznaczoną do zastosowań czysto „cyfrowych", wersję interfejsu - bez driverów. Układ US1



Rys. 1. Wyprowadzenia układów ULN2001/4 i ULN2801/4



Rys. 2. Schemat elektryczny uproszczonej wersji uniwersalnego interfejsu we/wy

```
PARTNO US2EX;
NAME Dekoder;
DEVICE G16V8;
REV 1.2;
DATE 4/4/93;
DESIGNER Piotr Zbysinski;
ASSEMBLY N/A;
LOCATION N/N;

/* ************************************** */
/* Dekoder adresowy do karty uniwersal. I/O */
/*    Dla adresow w zakresie 300H - 303H    */
/*  dla odczytu (!RD=0) i dla zapisu (!WR=0) */
/*     na wyjsciu CS_8255 jest stan "0".    */
/* ************************************** */

/*INPUTS*/
/*       Definicje wejsc dekodera          */

PIN [1..9,14] = [A0..A8,A9];
PIN 11 = !AEN;
PIN 12 = !RD;
PIN 13 = !WR;

/* ************************************** */
/*  Sygnal AEN zapisano z negacja (!AEN),  */
/*    ponieważ synchronizacja zapsiu portu */
/*  następuje wraz z opadajacym zboczem    */
/*              tego sygnalu!              */
/* ************************************** */


/*Outputs*/
/*     Definicja wyjscia dekodera          */

PIN 18 = !CS_8255;

/* Deklaracja zmiennych pomocniczych       */

field IOADR = [A0..A9]; /* Pole adresowe    */
field IOWR = [AEN,WR]; /* Pole steruj.zapisem */
field IORD = [AEN,RD]; /* Pole steruj.odczytem */
WR_NOW = IOWR:0;       /* Zapis do portu   */
RD_NOW = IORD:0;       /* Odczyt portu     */
ADR_GOOD = IOADR:[300..303];
                  /*Selekcja obszaru adr. */

/*Logic Equations*/
/* Rownania definiujace funkcje logiczne  */
/* wyjsc dekodera                         */

CS_8255 = (WR_NOW # RD_NOW) & ADR_GOOD;
                  /* Rownanie wyjscia CS_8255 */
                  /* wykorzystujace do opisu  */
                  /* zmienne pomocnicze       */


*********************************************
                  Dekoder
*********************************************
CUPL            4.0a Serial# MD-40A-8209
Device          g16v8s  Library DLIB-h-26-9
Created         Sun Oct 24 00:56:33 1993
Name            Dekoder
Partno          US2EX
Revision        1.2
Date            4/4/93
Designer        Piotr Zbysinski
Company         xxxxxx
Assembly        N/A
Location        N/N

================================================================
            Expanded Product Terms
================================================================

ADR_GOOD =>
    !A2 & !A3 & !A4 & !A5 & !A6 & !A7 & A8 & A9
CS_8255 =>
    !A2 & !A3 & !A4 & !A5 & !A6 & !A7 & A8 & A9 & !AEN & !RD
  # !A2 & !A3 & !A4 & !A5 & !A6 & !A7 & A8 & A9 & !AEN & !WR
IOADR =>
    A0 , A1 , A2 , A3 , A4 , A5 , A6 , A7 , A8 , A9
IORD =>
    AEN , RD
IOWR =>
    AEN , WR
RD_NOW =>
    !AEN & !RD
WR_NOW =>
    !AEN & !WR
================================================================
                  Chip Diagram
================================================================

                  _____________
                 |   Dekoder   |
            A0 x-|1          20|-x Vcc
            A1 x-|2          19|-x
            A2 x-|3          18|-x !CS_8255
            A3 x-|4          17|-x
            A4 x-|5          16|-x
            A5 x-|6          15|-x
            A6 x-|7          14|-x A9
            A7 x-|8          13|-x !WR
            A8 x-|9          12|-x !RD
           GND x-|10         11|-x !AEN
                 |_____________|
```

List. 1.

jest „sercem" całej konstrukcji. W zależności od trybu pracy może pracować jako brama wejściowa, wyjściowa lub dwukierunkowa. Możliwe jest mieszanie tych trybów pracy.

Znaczne uproszczenie konstrukcji umożliwiło wykorzystanie dekodera adresowego, znajdującego się na płytce AVT-114. Wadą tego rozwiązania jest dekodowanie pod wieloma adresami rejestrów układu US1EX (praktycznie cały obszar adresowy 300H - 31FH jest zajęty przez rejestry US1EX). Wynika to z faktu, że proponowany dekoder dekoduje obszar adresowy od 300H do 31FH. Wewnętrzne rejestry dekodowane są pod następującymi adresami:

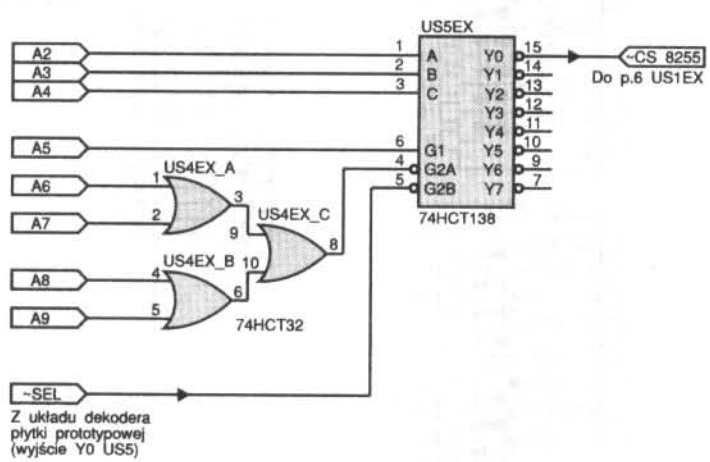
- dla odczytu:
  300H, 304H, 308H, 30CH itd. dla A0=„0" i A1=„0" - port A;
  301H, 305H, 309H, 30DH itd. dla A0=„1" i A1=„0" - port B;
  302H,306H, 30AH, 30EH itd. dla A0=„0" i A1=„1" - port C;
- dla zapisu:
  300H (reszta j.w.) - port A;
  301H (reszta j.w.) -port B;
  302H (reszta j.w.) - port C;
  303H, 307H, 30BH itd. dla A0=„1" i A1=„1" - rejestr sterujący;

Należy pamiętać, że wszystkie adresy mieszczą się w obszarze przeznaczonym dla kart prototypowych, więc takie bardzo uproszczone dekodowanie nie wpływa na jakość pracy komputera.

Może się oczywiście okazać, że niezbędne jest ograniczenie obszaru adresowego zajmowanego przez kartę. Dlatego na **rys. 3** jest przedstawiona propozycja takiej rozbudowy dekodera, aby rejestry karty zajmowały obszar od 300H do 303H, pozostawiając pozostałe adresy wolne.

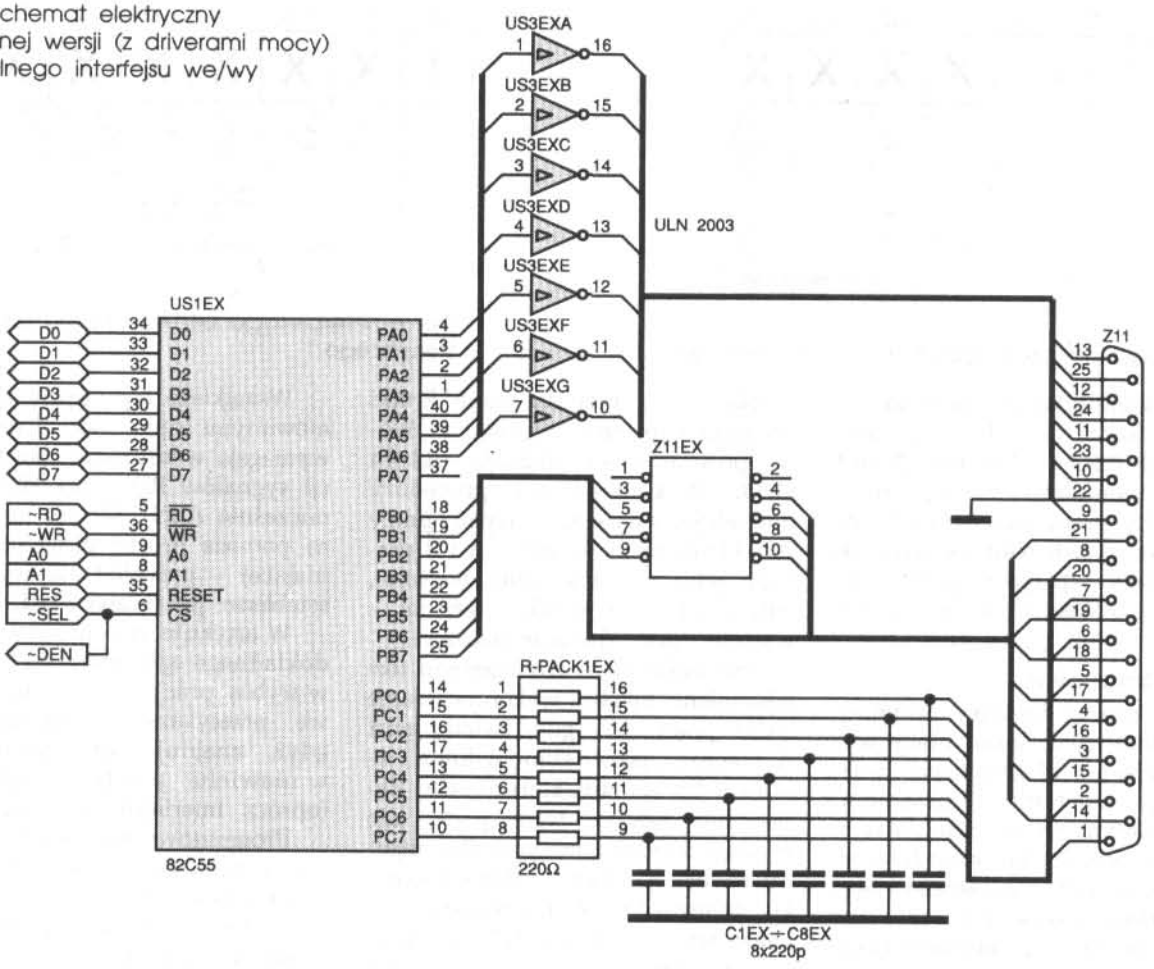
Oprócz takiego, w pewnym sensie „dyskretnego" (potrzebne jest kilka układów TTL) rozwiązania możliwe jest wykonanie dekodera z układem PLD - np. w układzie GAL16V8 (PALCE16V8) lub PAL16L8. Na **list. 1** przedstawiono projekt takiego dekodera (odpowiednika wersji dekodującej obszar 300H - 31FH). Program z list. 1 przygotowany został do kompilacji przez program CUPL - ze względu na prostotę ewentualne korekty (zmiana adresów bazowych) są możliwe do wykonania nawet przez początkujących elektroników. Do listingu dołączone są fragmenty pliku dokumentacyjnego tworzonego w czasie kompilacji przez CUPL. Zawarte są w nim wzory algebry Booleía opisujące funkcję wyjściową (ozn. CS_8255). Sygnał ten jest aktywny dla zapisu lub odczytu portu o adresie zawartym w zakresie 300H do 303H. Należy wykorzystać go do sterowania wyprowadzenia CS\ (pin 6) układu 8255. Pozostałe wejścia sterujące (WR\, RD\, RESET) dołączamy do buforowanej szyny na karcie prototypowej.

Rysunek 4 przedstawia schemat wersji rozszerzonej, wyposażonej w drivery mocy. Tak rozbudowana karta może sterować przekaźnikami (układy ULN wyposażone są



Rys. 3. Proponowany układ rozbudowy dekodera

Rys. 4. Schemat elektryczny rozszerzonej wersji (z driverami mocy) uniwersalnego interfejsu we/wy



w diody zabezpieczające tranzystory wyjściowe przed przepięciami powstającymi w cewce przełączanego przekaźnika), transoptorami lub optoprzekaźnikami.

Niezależnie od wersji (z driverami, czy bez) należy stosować na wejściach portów filtry dolnoprzepustowe RC, które eliminują zakłócenia indukowane w przewodach dołączonych do złącza wyjściowego. W celu uproszczenia montażu zastosowano zintegrowany pakiet rezystorów (firmy BECKMANN) w obudowie DIL16. Kondensatory należy tak dobrać, aby ich pojemność zapewniała poprawną filtrację sygnałów zakłócających (o dużych częstotliwościach), ale jednocześnie możliwie mało wpływała na szybkość transmisji.

Montaż i uruchomienie

Montaż należy przeprowadzić na płytce prototypowej AVT-114. Takie rozwiązanie obniża koszt interfejsu oraz ułatwia ewentualne modyfikacje - elastyczność projektu jest bardzo duża, ostateczną konfigurację można dostosować całkowicie do własnych potrzeb zmieniając połączenia wykonane przewodem w izolacji teflonowej (tzw. „kynarem").

Układy scalone dobrze jest umieszczać na podstawkach - ułatwia to ewentualne naprawy i modyfikacje urządzenia.

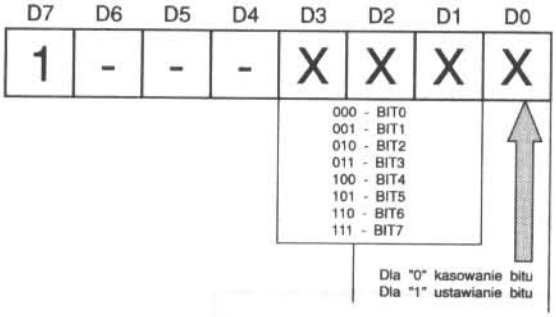
Do uruchomienia karty bardzo przydatny jest program PTEST, który jest dołączany do każdej dyskietki z oprogramowaniem przeznaczonym do obsługi urządzeń oferowanych przez AVT jako kity.

Procedura uruchomienia sprowadza się do sprawdzenia za pomocą miernika uniwersalnego lub testera poziomów logicznych, czy na kolejne bity D0-D7 portów A, B i C możliwe jest wpisanie „0" lub „1". Po włączeniu zasilania komputera należy pamiętać o konieczności wstępnej inicjalizacji układu US1EX. Można do tego celu wykorzystać procedurę z **list. 2** lub wykonać to za pomocą programu PTEST.

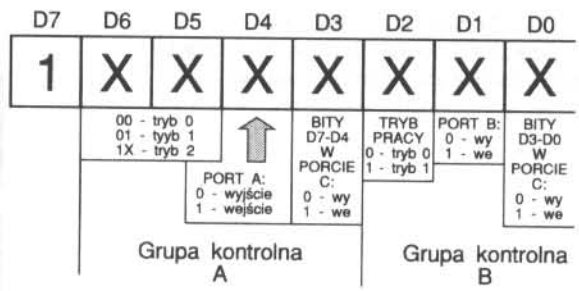
Procedura inicjalizacyjna polega na wpisaniu słowa sterującego (8-bitowego) do rejestru o adresie A0=„1" i A1=„1". W **tab. 1** są przedstawione stany sygnałów ste-

```
// Listing 2. Program inicjujacy uklad 8255.

#include <stdio.h>
#include <conio.h>  // dla prototypu getch()
#include <dos.h>

#define ADRES 0x303
#define TRYB1 0x89
#define TRYB2 0x80



/* ************************************** */
/*    Program inicjujacy prace ukladu 8255.   */
/*  Ustawiany jest tryb "0" - porty A i B sa  */
/*  definiowane jako wyjscia, port C moze byc */
/*           wejsciem lub wyjsciem.           */
/*  Mozliwe sa dowolne inne kombinacje - ten  */
/*      program jest ilustracja sposobu       */
/*             inicjalizacji.                 */
/* ************************************** */


int main(void)
{
  int c;

  printf("\nWybierz:\n");
  printf("       1. = Awy, Bwy, Cwe\n");
  printf("       2. = Awy, Bwy, Cwy\n");

  c = getch();

  if (c == '1')
    outportb( ADRES, TRYB1);
  else
  if (c == '2')
    outportb( ADRES, TRYB2);
  else
    printf("\nZly klawisz! Wybierz "1" lub "2".\n");

  return 0;

} //------------------- main
```

List. 2.



Rys. 5. Słowo sterujące trybami pracy dla rejestrów



Rys. 6. Interpretacja bitów w słowie rejestru sterującego

rujących komunikacją pomiędzy rejestrami a szyną danych (identyczne dla wersji 82C55). Na **rys. 5** jest przedstawione słowo sterujące ustalające tryby pracy dla rejestrów A, B i C. Na **rys. 6** pokazano znaczenie bitów słowa wpisywanego do rejestru sterującego w czasie programowania bitów portu C.

Uwagi końcowe

Mimo prostej budowy proponowanego interfejsu, sporo pracy wymaga napisanie efektywnego i pozbawionego „potknięć" programu, który pozwoli na wykorzystanie wszystkich możliwości oferowanych przez kartę. Ponieważ zastosowań tego tego interfejsu jest bardzo dużo, jego oprogramowanie pozostawiamy twórczej inwencji Czytelników - naszym zdaniem stworzenie uniwersalnego programu obsługi takiej karty mijałoby się z celem.

Na list. 2 przedstawiono krótki program napisany w języku C, który powoduje' inicjalizację układu 8255. Wykonanie tej procedury powoduje ustawienie trybu pracy „0". Oznacza to, że porty A i B pracują jako wyjścia (lub wejścia), natomiast port C jako wejście - wyjście (port dwukierunkowy).

Być może nieco zręczniejszym rozwiązaniem byłoby wykorzystywanie trybu pracy „2", gdyż wtedy port A pracuje jako standardowa 8-bitowa brama dwukierunkowa. Zapis lub odczyt tego portu nie wymaga każdorazowego zapisu rejestru sterującego (przy zmianie kierunku transmisji). Wykorzystano jednakże tryb „0" ze względu na możliwość programowej konfiguracji kierunku przepływu informacji przez port C. Zwiększa to elastyczność interfejsu.

Wszelkiego typu operacje programowe typu zapis - odczyt na portach wymagają wydłużenia czasu generacji sygnałów RD\ oraz WR\ (i jednocześnie AEN). Można to osiągnąć za pomocą prostej procedury opóźniającej - przykład napisany w asemblerze przedstawia **list. 3**.

W artykule nie przedstawiliśmy dokładnego opis pracy układu 8255 w trybie pracy „1" (dwukierunkowe przesyłanie z potwierdzeniem), gdyż znajduje on zastosowanie w niewielu prostych aplikacjach (oprócz interfejsu drukarki).

Programowe sterowanie interfejsem umożliwia oczywiście dowolne przekonfigurowanie wykorzystywanych wejść i wyjść, dlatego proponujemy potraktowanie tego artykułu jako zachęty do prób realizacji własnych rozwiązań.

Piotr Zbysiński, AVT

Tab. 1. Operacje wykonywane przez układ 8255 w zależności od stanu sygnałów sterujących

!CS	!RD	!WR	A1	A0	Operacja
0	0	1	0	0	Odczyt portu A
0	0	1	0	1	Odczyt portu B
0	0	1	1	0	Odczyt portu C
0	0	1	1	1	**OPERACJA ZABRONIONA**
0	1	0	0	0	Zapis portu A
0	1	0	0	1	Zapis portu B
0	1	0	1	0	Zapis portu C
0	1	0	1	1	Zapis rejestru sterującego
0	1	1	X	X	Żadna operacja nie jest
0	X	X	X	X	wykonywana

WYKAZ ELEMENTÓW

C1EX, C2EX, C3EX, C4EX, C5EX, C6EX, C7EX, C8EX: 220pF

R-PACK1EX: 220Ω
US1EX: 82C55
ZI1EX: złącze 5x2

```
/* ********************************************
                    AVT 114/2
   program do zapisu portu (adres podany
       z klawiatury) wartoscia HEX
         (podana z klawiatury)
                                        1994-02-15
   ******************************************** */

#include <stdio.h>
#include <dos.h>
#include <conio.h>
int main( void )
{
  int port;
  int val;
  printf( "\nPodaj adres portu (HEX): " );
  scanf( "%x", &port );
  printf( "\nPodaj wartosc wysylana do portu (HEX): " );
  scanf( "%x", &val );
  printf( "\n\n Wcisnij dowolny klawisz, zeby \
zakonczyc wysylanie %xH do portu %xH\n", val, port );
  while ( !kbhit() )
    outport( port, val );

  return 0;
} /* ------------------- main */
```

List. 3.