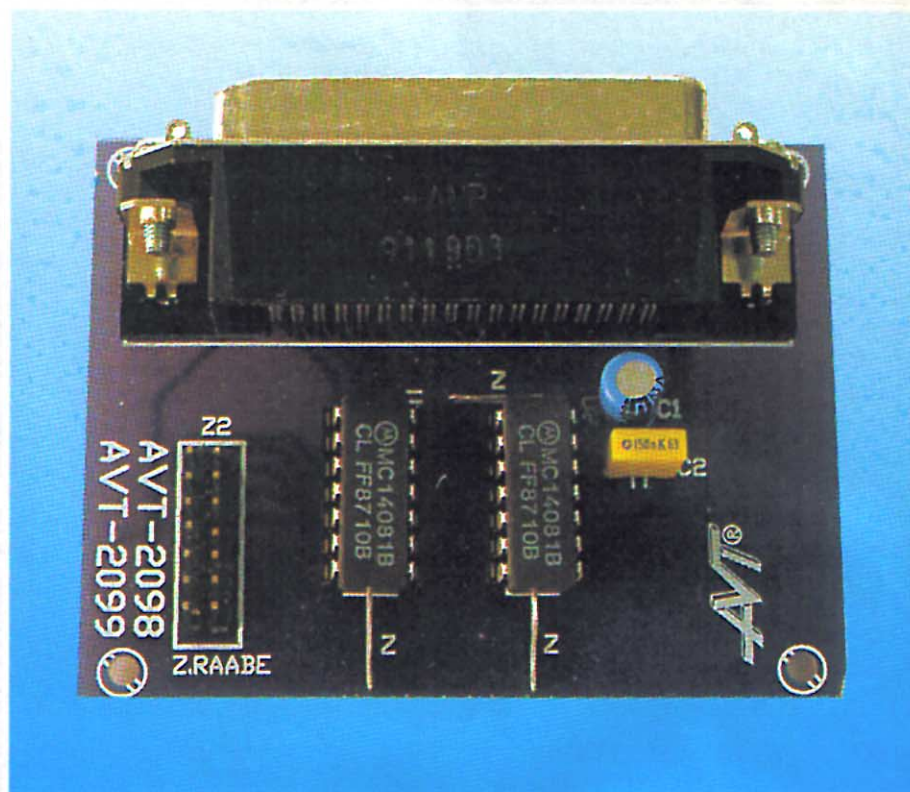


Nazwa proponowanego układu nie jest zbyt ścisła: nie jest on bowiem interfejsem z prawdziwego zdarzenia. Umożliwia on komunikację komputera z urządzeniem peryferyjnym tylko w jedną stronę, możemy wysłać rozkazy do tego urządzenia, natomiast komputer pozostaje ślepy i głuchy na wszelkie sygnały dochodzące z zewnątrz. Natomiast angielskie słowo "interface" sugeruje raczej możliwość porozumiewania się w obie strony. Proponowane urządzenie umożliwia sterowanie przez komputer zgodny ze standardem IBM PC (lub jakiegokolwiek inny wyposażony w port CENTRONICS) najróżniejszych urządzeń peryferyjnych. Sam układ nic nie może, jest tylko "prześciówką" pomiędzy komputerem, a jednym z już opracowanych i opublikowanych w EdW układów wykonawczych.



Najprostszy ośmiokanałowy interfejs

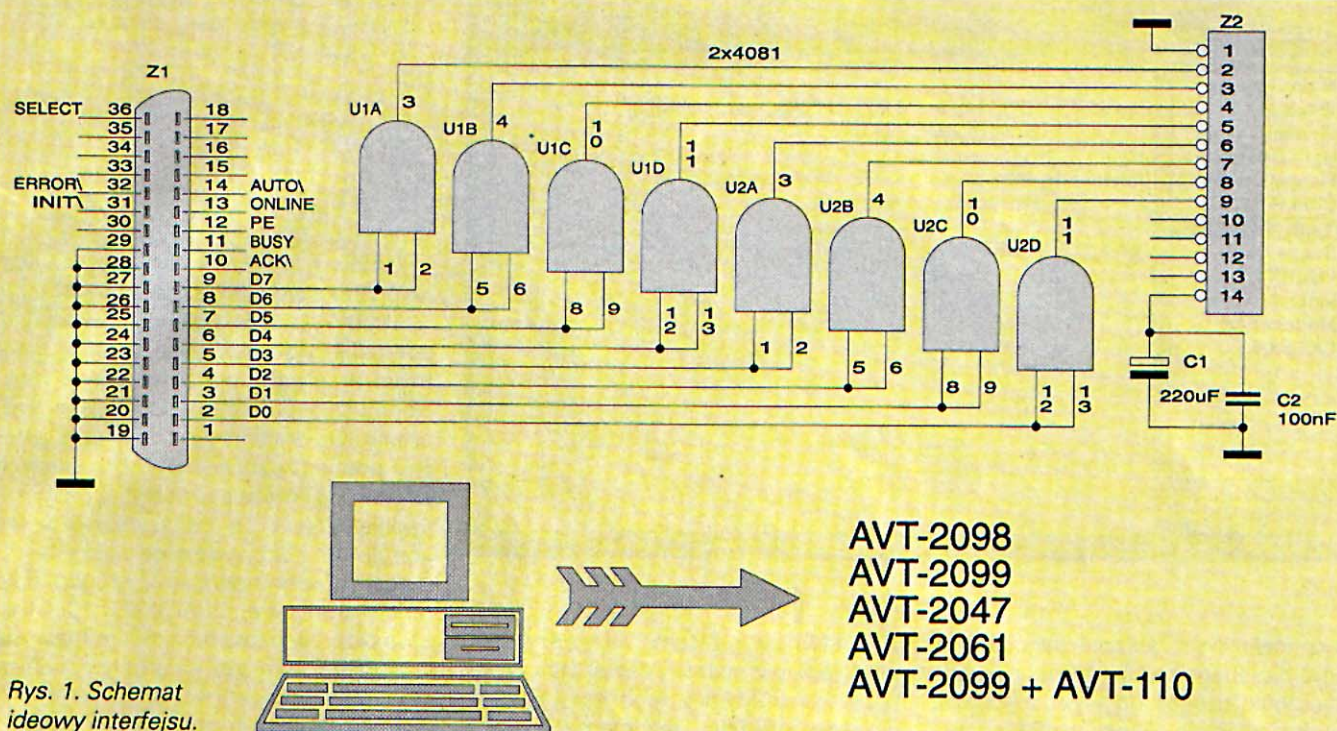


Układy te to: AVT-2047 (sterownik do modeli zabawek), AVT-2061 (sterownik silników krokowych), AVT-2098 (sterownik dowolnych urządzeń zasilanych prądem elektrycznym, nawet tych o wielkiej mocy) i wreszcie AVT-2099 (sterownik układów średniej mocy). W tych dwóch ostatnich urządzeniach zastosowanie systemu zależy od Waszej pomysłowości. Czy nie będzie wielką satysfakcją dla młodego elektronika i programisty zaprezentowanie rodzinie całkowicie skomputeryzowanej choinki? Za pomocą komputera i sterownika AVT-2099 lub AVT-2098 możecie zrealizować absolutnie dowolne efekty świetlne, ograniczeniem może być tylko inwencja programisty! Możecie konstruować girlandy świetlne, gwiazdy, dowolne figury geometryczne, pełzające "węże" świetlne i wszystko, co sobie wyobrazicie.

Do tego wszystkiego będzie potrzebny komputer, najlepiej zgodny ze standardem IBM PC. Nie musi to być jednak najnowocześniejsza maszyna. Wręcz przeciwnie, do naszych celów wystarczy nawet muzealny XT ze stacją dysków

360kB. Wykorzystanie komputera jako inteligentnego sterownika jest zresztą doskonałym zastosowaniem dla przestarzałych maszyn, które nie nadają się już do jakiegokolwiek pracy, ale które żal wyrzucić na śmietnik. W praktyce nie było żadnej różnicy pomiędzy tą konfiguracją, a zastosowaniem maszyny z procesorem INTEL PENTIUM i kartą SVGA. Myślimy też o wykorzystaniu jako sterowników komputerów ośmiobitowych. W swoim czasie, dawno temu autor zbudował dla swojej córki kilka zabawek sterowanych przez COMMODORE C64. Zabawki te działały znakomicie, lecz niestety dawno uległy kompletnej dewastacji. Podczas przeprowadzki autorowi zaginęła cała dokumentacja do C64 - dwa grube tomy z wyczerpującym opisem tej maszyny. I tu autor zwraca się z ogromną prośbą do swoich Czytelników: może ktoś z Was posiada taką dokumentację i mógłby ją udostępnić redakcji na kilka dni, potrzebnych do wykonania kserokopii?

Po raz pierwszy znajdziemy się w zupełnie nowej sytuacji. Naszym podsta-



Rys. 1. Schemat ideowy interfejsu.

wowym narzędziem pracy nie będzie już lutownica. Moduł AVT-2027 można zmontować w ciągu dosłownie dziesięciu minut, ale pisanie programów może nam zająć całe noce. Jest to zresztą całkowicie zgodne z trendami panującymi obecnie w elektronice. Układy mikroprocesorowe stają się coraz prostsze i tańsze, możliwe do wykonania nawet przez początkującego amatora. Natomiast odpowiednie ich oprogramowanie wymaga wielkiej pracy, sporego doświadczenia i umiejętności.

Drugim elementem niezbędnym do wykorzystania komputera jako sterownika jest oczywiście język programowania. Z tym nie będziemy mieli najmniejszego kłopotu, w każdym bez wyjątku komputerze klasy PC zainstalowany jest system operacyjny MS-DOS, który zawiera w sobie interpreter popularnego języka BA-

SIC. W starszych wersjach DOSu był to GWBASIC, w nowszych zastąpiono go nowocześniejszym QBASICem.

Opis działania

Schemat ideowy interfejsu pokazany został na rysunku 1 i trzeba przyznać, że schemat ten jest mało interesujący. Tak naprawdę jedynym zadaniem układu jest buforowanie wyjść portu CENTRONICS komputera i zapewnienie przejścia tego złącza do typowego gniazda stosowanego w naszych układach automatyki.

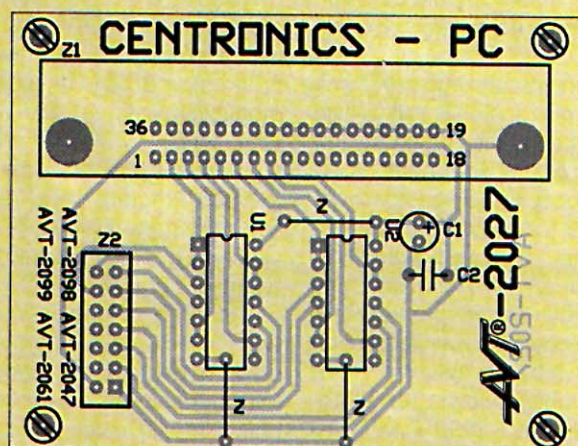
Rolę buforów spełnia (całkowicie zgodnie ze swoim przeznaczeniem) osiem bramek AND typu 4081. Ponieważ bramki te nie negują stanów logicznych dostarczanych na ich wejścia, na wyjściu interfejsu mamy dokładny odpowiednik słowa ośmiobitowego wysłanego na wyjście CENTRONICS komputera.

Układ zasilany jest napięciem +5VDC dostarczonym ze współpracującego z nim modułu wykonawczego.

Montaż i uruchomienie

Mozaika ścieżek płytki drukowanej jest pokazana na wkladce, a rozmieszczenie elementów pokazane są na rysunku 2. Montaż tego układu nie nastręczy nikomu najmniejszych trudności, nawet tym, którzy są bardziej programistami niż elektronikami. Wykonujemy go tradycyjną metodą, rozpoczynając od podstawek pod układy scalone, a kończąc na złączu CENTRONICS. Włutowanie tego złącza wymaga ze względu na dużą (36) ilość końcówek pewnej uwagi. Autor nie musi chyba zaznaczać, że układ nie wymaga regulacji ani uruchamiania.

Po zmontowaniu układu musimy go oczywiście wypróbować. Nie musimy



Rys. 2. Płytkę drukowaną.

WYKAZ ELEMENTÓW

Kondensatory

C1: 220µF/16V
C2: 100nF

Półprzewodniki

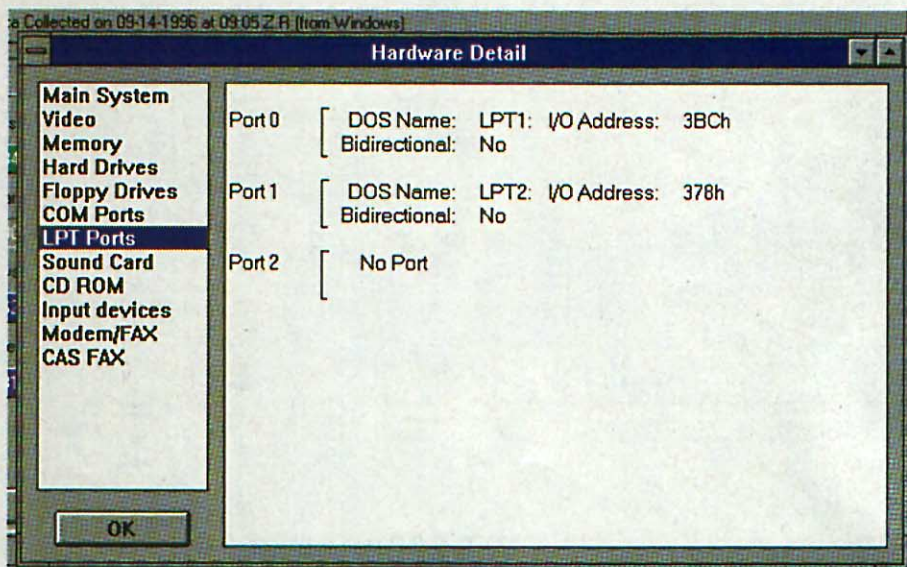
U1, U2: 4081

Różne

Z1: złącze CENTRONICS do lutowania w płytce
Z2: goldpin 2x7

Dwa wtyki do przewodów taśmowych 14

Przewód taśmowy 14 ok. 0,5m.



Rys. 3.

w tym celu pisać jeszcze jakiegokolwiek programu. Pierwszą czynnością będzie podłączenie czegoś do wyjścia modułu współpracującego. W przypadku modułu AVT-2099 najlepiej dołączyć do jego wyjść osiem diod LED z rezystorami szeregowymi ograniczającymi płynący przez nie prąd. Do modułu AVT-2098 nie musimy na razie niczego dołączać, ponieważ posiada on już osiem diod sygnalizujących pracę przekaźników wykonawczych.

Następna czynnością musi być ustalenie adresu portu CENTRONICS naszego komputera. Do tego celu wykorzystamy jeden z popularnych programów diagnostycznych, np. MSD lub CHECKIT. Także popularny NORTON COMMANDER posiada możliwość sprawdzenia konfiguracji systemu za pomocą opcji **System Information**. Jeżeli w naszym komputerze posiadamy tylko jeden port: LPT1 to jego adresem będzie najczęściej 378H. W każdym wypadku adresy portów zostaną z pewnością ustalone przez program diagnostyczny (patrz rys. 3). Po wybraniu portu i dołączeniu do niego naszego układu przystępujemy do prób. Uruchamiamy posiadany przez nas interpreter BASICu i w trybie natychmiastowym wydajemy polecenie:

```
OUT &H[adres portu], 255
```

Przy jednym porcie wyjściowym LPT1 polecenie to będzie miało najprawdopodobniej postać:

```
OUT &H378, 255
```

Natychmiast po wydaniu tego polecenia powinny zapalić się wszystkie diody lub włączyć się wszystkie urządzenia połączone z modułem wykonawczym. Dlaczego tak się stanie? Wysłanie do portu wyjściowego jakiegokolwiek liczby z przedziału 0 - 255 powoduje pojawienie się jej reprezentacji binarnej na ośmiu wyjściach danych portu. 255(DEC) to

11111111(BIN), a zatem na wszystkich wyjściach wymuszony został stan wysoki i w konsekwencji tego na wyjściach naszego interfejsu też mamy wysokie poziomy logiczne. Stany wysokie włączyły tranzystory w układach wykonawczych, dając opisany efekt.

Poeksperymentujmy jeszcze chwilę, podając na wyjście portu LPT komputera inne liczby z dozwolonego zakresu, a następnie przystąpimy do pisania prostych programów.

Przykłady programów

Autor tego artykułu jest kiepskim (czytaj: żadnym) programistą. Zbudować młotek wprawdzie potrafi, ale z wbijaniem nim gwoździ ma problemy. Dlatego też prosi swoich kolegów o potraktowanie zamieszczonych poniżej listingów programów jedynie jako wskazówek i o nie czerpanie wzorów z tak marnych przykładów. Jednocześnie ogłaszamy konkurs na najciekawsze oprogramowanie wykorzystujące jeden z proponowanych systemów. Autor podejmuje się zmontować pięć układów AVT-2027 i przeznaczyć je na nagrody dla Czytelników, którzy nadeślą najciekawsze opracowania.

Zacznijmy od najprostszego programiku. Pozwala on na zapalenie całkowicie przypadkowych kombinacji diod LED lub żarówek podłączonych do modułu AVT-2099. Można oczywiście zastosować moduł AVT-2098 i sterować nawet reflektorami o dużej mocy, ale sterownik musi być w takim wypadku umieszczony w jakimś dzwiękochłonnym pomieszczeniu (trzaski włączanych i wyłączanych przekaźników byłyby na dłuższą metę trudne do zniesienia). Można też zastosować kombinacje AVT-2099 + AVT-110 (patrz opis modułu AVT-2099). Listing programu wygląda następująco:

```
* PROGRAM GENERUJE LICZBY LOSOWE
* I ZAPALA PRZYPADKOWE KOMBINACJE
* DIOD LED LUB ZAROWEK.
1000 CLS : Z = 10000 : Z DECYDUJE
* O SZYBKOSCI PRACY
1100 X = INT(RND(1) * 255)
1200 PRINT X,
1300 OUT &H3BC, X
1400 FOR R = 1 TO Z: NEXT R
1500 IF INKEY$ = CHR$(27) THEN END
GOTO 1100
```

Omówmy kolejno funkcje wykonywane przez kolejne linie programu:

1000 Pierwsze polecenie w tej linii powoduje wyczyszczenie ekranu, natomiast drugie ma dla poprawnego funkcjonowania programu decydujące znaczenie. Od zmiennej Z zależy czas przerwy pomiędzy zmianami wyświetlanych kombinacji i może być ustalony doświadczalnie. Zależy on od szybkości pracy komputera i oczywiście od naszych wymagań. Zadeklarowanie Z=10000 dało na komputerze autora (DX468 100MHz) czas przerwy około 0,5 sek. Im maszyna wolniejsza, tym ten parametr powinien mieć mniejszą wartość.

1100 W linii tej zawarta jest najważniejsza funkcja programu: generowanie liczby losowej z przedziału 0 - 255. Wydanie polecenia RND(1) zwraca nam losowo wybraną liczbę z zakresu 0 - 1, np. 0,2356781. Są to liczby dla naszych celów jakby trochę za małe i dlatego mnożymy je przez 255. Wynik takiego mnożenia przedstawiałby się następująco: 255 x 0,2356781 = 60,0979155. Mamy już wybraną liczbę z wskazanego przedziału, ale z ułamkowym "ogonkiem". Ponieważ port wyjściowy komputera ułamków raczej nie lubi, zastosujemy jeszcze jedną funkcję: INT (ang. INTEGER - liczba całkowita). Funkcja ta zwraca nam liczbę zaokrągloną do postaci całkowitej, czyli w naszym przypadku 60.

1200 Polecenie zawarte w tej linii to typowy "ozdobniczek". Wyświetla ono na ekranie wartości kolejnych liczb losowych.

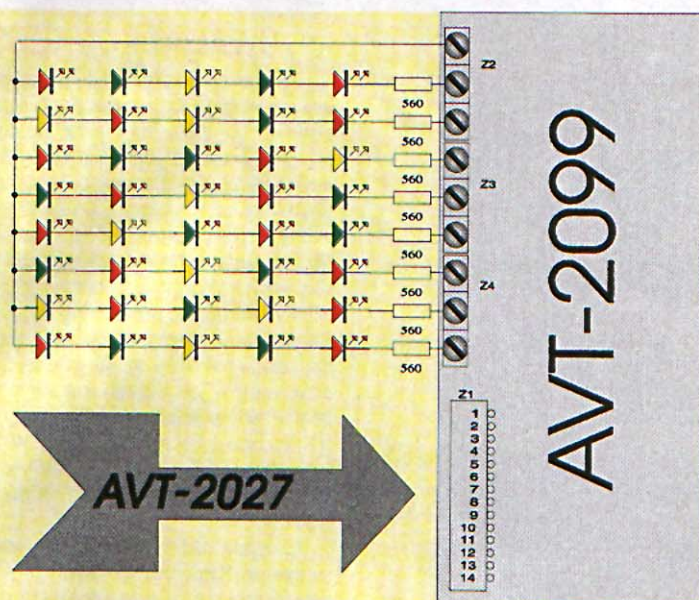
1300 Polecenie OUT wysyła do portu wyjściowego wybraną liczbę.

1400 W linii tej realizowane jest za pomocą instrukcji FOR ... TO ... NEXT opóźnienie potrzebne do prawidłowej pracy programu. Gdyby tej linii nie było nie moglibyśmy nawet zauważyć jakichkolwiek zmian w układzie wykonawczym, tak szybko by się one odbywały.

1500 Program sprawdza, czy nie został naciśnięty klawisz [ESC]. Jeżeli tak się stało to praca programu zostanie natychmiast przerwana.

1600 Nie lubiane polecenie GOTO powoduje powrót programu do linii 1100, wygenerowanie następnej liczby losowej, podanie jej do portu wyjściowego i tak dalej, tak dalej....

Pomimo że powyższy program jest jedynie przykładem, to może on mieć cie-



Rys. 4. Podłączenia diod LED.

List. 1. Program sterowania silnikiem krokowym.

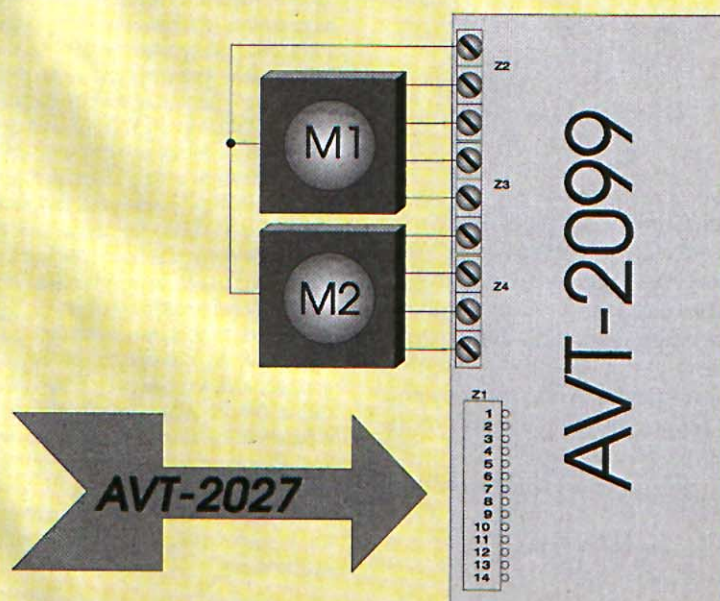
```

10 CLS
100 INPUT "JAKA MA BYC SZYKOSC OBROTOWA"; X
200 INPUT "ILE KROKOW x 4"; Z
300 INPUT "W KTORA STRONE [Left/Right]"; A$
400 IF A$ = "L" THEN 2000 ELSE 2200

1000 CLS
2000 FOR R = 1 TO Z: GOSUB 3000: PRINT "WYKONUJE...": NEXT R: OUT &H3BC, 0:
GOTO 1
2200 FOR R = 1 TO Z: GOSUB 4000: PRINT "WYKONUJE...": NEXT R: OUT &H3BC, 0:
GOTO 1

3000 OUT &H3BC, 16: FOR K = 1 TO X: NEXT K
3100 OUT &H3BC, 32: FOR K = 1 TO X: NEXT K
3200 OUT &H3BC, 64: FOR K = 1 TO X: NEXT K
3300 OUT &H3BC, 128: FOR K = 1 TO X: NEXT K
3400 RETURN

4000 OUT &H3BC, 128: FOR K = 1 TO X: NEXT K
4100 OUT &H3BC, 64: FOR K = 1 TO X: NEXT K
4200 OUT &H3BC, 32: FOR K = 1 TO X: NEXT K
4300 OUT &H3BC, 16: FOR K = 1 TO X: NEXT K
4400 RETURN
    
```



Rys. 5. Dołączenie silników krokowych.

kawę zastosowanie praktyczne. Podłączając do wyjść modułu AVT-2099 girlandy diod świecących możemy uzyskać interesujące efekty świetlne, np. na choince. Schemat podłączenia diod LED do modułu AVT-2099 pokazany został na rysunku 4.

Urządzeniem, które aż się prosi o sterowanie go za pośrednictwem naszego interfejsu jest silnik krokowy, tym bardziej że sterować możemy pracą dwóch silników na raz. I w tym przypadku możliwości zastosowania sterownika silników są wprost nieograniczone i także zależą wyłącznie od inwencji programisty i stawianych mu zadań. Precyzja z jaką będą działać sterowane komputerem silniki jest ogromna. Wystarczy zresztą popatrzeć jak pracuje ploter lub drukarka, aby zorientować się w otwierających się przed nami możliwościach. Wiele urządzeń mechanicznych, których realizacja tradycyjnymi metodami byłaby niemożliwa, da się teraz skonstruować minimalnym wysiłkiem.

Na list. 1 zamieszczamy króciutki programik do sterowania pracą silnika krokowego. Jest to tylko przykład, program nie pełni w zasadzie jakichkolwiek funkcji użytkowych. Umożliwia jedynie obracanie się silnika w dowolną stronę z zadaną szybkością i o podaną ilość kroków. Nie będziemy już opisywać szczegółowo jego działania, chociażby dlatego, że nie jest to program zbyt wysokiego lotu. Silniki krokowe (czterofazowe) można podłączyć do modułu AVT-2099 w sposób pokazany na rysunku 5.

Zamieszczony na listingu 2 podprogram dobitnie świadczy, jak wielkie możliwości daje zastosowanie komputera do sterowania silnikiem krokowym. W niektórych przypadkach doskonała precyzja tego silnika może przestać nam wystarczać i możemy zechcieć aby pracował on dwukrotnie precyzyjniej: co pół kroku. Nic prostszego, prezentowany program jest najprostszym przykładem rozwiązania takiego problemu. Właściwie nie jest to kompletny program, ale jedynie kilka linijek, które możemy włączyć do większego programu.

Kolejną grupą urządzeń jakimi możemy sterować za pośrednictwem interfejsu

List. 2. Program do sterowania silnikiem ze skokiem co pół kroku.

```

1000 OUT &H3BC, 128: GOSUB 2000
1050 OUT &H3BC, 128 + 64: GOSUB 2000
1100 OUT &H3BC, 64: GOSUB 2000
1150 OUT &H3BC, 64 + 32: GOSUB 2000
1200 OUT &H3BC, 32: GOSUB 2000
1250 OUT &H3BC, 32 + 16: GOSUB 2000
1300 OUT &H3BC, 16: GOSUB 2000
1350 OUT &H3BC, 16 + 8: GOSUB 2000
1400 GOTO 1000
2000 FOR R = 1 TO 30: NEXT R
2100 RETURN
    
```


List. 3. Polecenia sterowania pojazdem

```

REM OUT &H3BC,64+32 DO PRZODU
REM OUT &H3BC,128+16 DO TYLU
REM OUT &H3BC,64 SKRET W PRAWO
REM OUT &H3BC,32 SKRET W LEWO
REM OUT &H3BC,128 SKRET DO TYLU W PRAWO
REM OUT &H3BC,16 SKRET DO TYLU W LEWO
REM OUT &H3BC,128+32 OBROT W LEWO
REM OUT &H3BC,64+16 OBROT W PRAWO
5 DIM A$(200): DIM A(200): L = 0
10 CLS : PRINT "PODAJ RODZAJ FUNKCJI PROGRAMU"
20 PRINT "ZAPIS -1"
30 PRINT "ODCZYT -2"
35 PRINT "KONIEC -3"
40 INPUT C
50 IF C = 1 THEN 100
60 IF C = 2 THEN 2000
65 IF C = 3 OR C <> 1 OR C <> 2 THEN END
70 GOTO 10
100 CLS : OUT &H3BC, 0
120 PRINT "PODAJ RODZAJ PORUSZENIA:"
140 PRINT "FF - NAPRZOD"
160 PRINT "BB - DO TYLU"
180 PRINT "FR - SKRET W PRAWO"
200 PRINT "FL - SKRET W LEWO"
220 PRINT "BR - SKRET DO TYLU W PRAWO"
240 PRINT "BL - SKRET DO TYLU W LEWO"
260 PRINT "RL - OBROT W LEWO"
280 PRINT "RR - OBROT W PRAWO"
282 PRINT "END- KONIEC"
285 INPUT X$: A$(L) = X$
286 IF X$ = "END" THEN 10
290 INPUT "PODAJ CZAS DZIAŁANIA POLECENIA": X: X = X * 1000
295 A(L) = X: L = L + 1
300 IF X$ = "FF" THEN GOSUB 1000: GOTO 100
320 IF X$ = "BB" THEN GOSUB 1100: GOTO 100
340 IF X$ = "FR" THEN GOSUB 1200: GOTO 100
360 IF X$ = "FL" THEN GOSUB 1300: GOTO 100
380 IF X$ = "BR" THEN GOSUB 1400: GOTO 100
400 IF X$ = "BL" THEN GOSUB 1500: GOTO 100
420 IF X$ = "RL" THEN GOSUB 1600: GOTO 100
440 IF X$ = "RR" THEN GOSUB 1700: GOTO 100
500 PRINT "BLAD W POLECENIU": FOR R = 1 TO 10000: NEXT R: GOTO 100
1000 OUT &H3BC, 64 + 32: FOR R = 1 TO X: NEXT R: RETURN
1100 OUT &H3BC, 128 + 16: FOR R = 1 TO X: NEXT R: RETURN
1200 OUT &H3BC, 64: FOR R = 1 TO X: NEXT R: RETURN
1300 OUT &H3BC, 32: FOR R = 1 TO X: NEXT R: RETURN
1400 OUT &H3BC, 128: FOR R = 1 TO X: NEXT R: RETURN
1500 OUT &H3BC, 16: FOR R = 1 TO X: NEXT R: RETURN
1600 OUT &H3BC, 128 + 32: FOR R = 1 TO X: NEXT R: RETURN
1700 OUT &H3BC, 64 + 16: FOR R = 1 TO X: NEXT R: RETURN
2000 L = -1
2100 L = L + 1: PRINT A$(L), A(L)
2200 X$ = A$(L): X = A(L)
3000 IF A$(L) = "FF" THEN GOSUB 1000: GOTO 2100
3200 IF A$(L) = "BB" THEN GOSUB 1100: GOTO 2100
3400 IF A$(L) = "FR" THEN GOSUB 1200: GOTO 2100
3600 IF A$(L) = "FL" THEN GOSUB 1300: GOTO 2100
3800 IF A$(L) = "BR" THEN GOSUB 1400: GOTO 2100
4000 IF A$(L) = "BL" THEN GOSUB 1500: GOTO 2100
4200 IF A$(L) = "RL" THEN GOSUB 1600: GOTO 2100
4400 IF A$(L) = "RR" THEN GOSUB 1700: GOTO 2100
4500 IF A$(L) = "END" THEN L = 0: OUT &H3BC, 0: GOTO 10

```

su są "raabowozy" (tak właśnie dowcipni koledzy redakcyjni autora nazywają skonstruowane przez niego pojazdy z silnikami krokowymi). Zamieszczony na list. 3 przykład programu jest jedynie prostą "przymiarką" do komputerowego sterowania robotami. Pamiętajmy bowiem, że za pomocą opisanego wyżej interfejsu możemy jedynie wysyłać polecenia do robota, ale nie możemy odebrać od niego żadnego sygnału. Do czasu, prototyp pełnego interfejsu do sterowania robotami jest już prawie gotowy!

Program ten nie tylko umożliwia wysłanie z komputera ośmiu różnych pole-

ceń, ale zapisanie w pamięci do dwustu komend i odtworzenie w dowolnym momencie drogi przebytej przez pojazd.

Wszystkie pokazane wyżej programy są jedynie przykładami i nie będą rozpowszechniane na dyskietkach. Jeszcze raz prosimy Czytelników o nadsyłanie własnych propozycji i przykładów programów.

Zbigniew Raabe