

I NFORMATYKA K OMPUTERY S YSTEMY



CENA 80 zł

DODATEK „ŻOŁNIERZA WOLNOŚCI” NR 3/1988 ISSN 0860-2794

W szponach Atari
Interfejs magnetofonu
Compact Disc
Grafika na ekranie



W NUMERZE:

Doczekaliśmy się zdrowej konkurencji — str. 4
ODRA — Amstrad — str. 5

W szponach Atari (13) — str. 6—10

Tabelaryzacja — str. 10—11
Interfejs magnetofonu — str. 12—13

Lilavati (cz. 3) — str. 14—15

Laureaci „Ligi Myślących” — str. 15

Compact Disc — str. 16—18
Standard MSX, a mikrokomputer SVI-738 X'PRESS — str. 19

Grafika na ekranie (cz. 4) — str. 20—21

Animacja kwadratu — str. 22
Budżet domowy — str. 23—25

Krzywe Sierpińskiego — str. 26—28

Informatyczny słownik angielsko-polski — str. 29—30

Giełda pomysłów — str. 31

Komputerowy biznes Centralnej Składnicy Harcerskiej

Od roku 1986 Centralna Składnica Harcerska handluje również sprzętem komputerowym. Osiem wyspecjalizowanych placówek w całym kraju sprzedało w ubiegłym roku sprzęt wartości 6.5 miliarda złotych. CSH związała się z jednym dostawcą — jest nią austriacka firma PROSYSTEM, a warunki, jakie zaproponowała są na tyle atrakcyjne, że polski kontrahent nie myśli o zmianie partnera. Szczegóły umowy są oczywiście tajemnicą handlową. Muszą być jednak atrakcyjne dla obu stron, skoro na polski rynek trafiają za sprawą Austriaków kolejne generacje sprzętu kompute-

rowego. Rozmowy partnerów muszą być jednak długie, efektem są wypełnione półki w sklepach CSH i... coraz lepsza znajomość języka polskiego przez właściciela Prosystemu Hansa Hofera.

W najbliższej przyszłości na Mokotowskiej będzie można kupić komputery zgodne z IBM PC/XT i IBM PC/AT, zbudowane według koncepcji wspólnej magistrali, szybkie, 32-bitowe komputery z zegarem 24 MHz (!), a także wielokolorowe projektory grafiki kolorowej, sieci komputerowe, systemy edycji składu i komputerowego wspomaganie projektowania inżynierskiego.

Sklepy CSH prowadzące sprzedaż sprzętu mikrokomputerowego

BYDGOSZCZ
KATOWICE
KRAKÓW
LUBLIN
ŁÓDŹ
SZCZECIN
WARSZAWA
WROCŁAW

ul. Chodkiewicza 68,
ul. Rozdzieńskiego 88a,
ul. Rynek Główny 5
ul. Kowalska 14,
ul. Lutomierska 12,
ul. Mariana Buczka 24c,
ul. Mokotowska 26,
plac Grunwaldzki 6a

telefon 41 44 30
telefon 58 28 33
telefon 22 73 12
telefon 29 472
telefon 57 15 27
telefon 35 596
telefon 21 76 55
telefon 21 92 61

Pascal w drodze do Czytelników

Już wkrótce drukarnię opuści kolejny zeszyt programów „IKS-a”. Pierwszy w tym roku poświęcony będzie w całości pascalowi. Ten język należy do jednych z najefektywniejszych narzędzi programowania strukturalnego. Programy napisane w pascalu działają o wiele szybciej niż te, które do tej pory zwykliśmy pisać w basicu.

I jeszcze jedno: pascal jest jeden — program napisany w tym języku na przykład na Spectrum można przepisać bez żadnych przeróbek w postaci źródłowej i uruchomić na IBM-ie. W naukę pascala warto zainwestować.

POCZTOWA GIEŁDA

Rozwiązanie krzyżówki z nr. 1/88 „IKS-a”.

Nagrody książkowe otrzymują: Sławomir Łysz — Włocławek, Andrzej Erd — Radom, Marcin Kobiela — Świerklaniec, Małgorzata Nowak — Kraków, Andrzej Niedbała — Chorzów, Dorota Baranowska — Oleśnica, Adam Dubicki — Gdynia, Jarosław Rosiński — Radom, Maria Rabczak — Gdańsk, Adam Kuryłowski — Szczecin.

Rozwiązanie krzyżówki z nr. 2/88 „IKS-a”.

Nagrody książkowe otrzymują: Wacław Skrzypczak — Warszawa, Małgorzata Czaplą — Tarnowskie Góry, Eugeniusz Chaszczewski — Kłodzko, Hanna Kornatowska — Warszawa, Przemysław Niekrasz — Wrocław, Krzysztof Małek — Zwoleń, Michał Urbaniak — Turek, Tomasz Skalski — Mrągowo, Jacek Wolski — Łódź, Łukasz Gumkowski — Ostrołęka.

LAUREACI DRUGIEJ EDYCJI „LIGI MYŚLĄCYCH”

Zwycięzcami drugiej edycji „Ligi Myślących” zostali, uzyskując maksymalną liczbę 64 punktów, za prawidłowe rozwiązanie wszystkich zadań, następujący czytelnicy:

Zdzisław BEDYNEK
03-982 Warszawa
ul. Zawadzkiego 18/68

Wiesław RYCHLICKI
38-604 Hoczew
Średnia Wieś (szkoła)

O jeden punkt mniej, tj. 63 uzyskali następujący czytelnicy:

Magdalena ROJEK
01-006 Warszawa
ul. Nowolipki 19/15

Andrzej BARCEWICZ
18-400 Łomża
ul. Broniewskiego 2/39

Adam DOMIŃSKI
80-395 Gdańsk
ul. Rokossowskiego 11/49

O teminie wręczenia nagród laureaci zostaną powiadomieni listownie, natomiast od następnego numeru rozpoczynamy kolejną turę zmagania Myślących.

NAUKA — TECHNIKA

Firma MARTIN MARIETTA przystępuje do modyfikacji międzykontynentalnych rakiet balistycznych typu TITAN II w celu przystosowania ich do roli nosiciela satelitów. Pierwsza z czterech zamówionych, zmodyfikowana rakiet tego typu została już przetransportowana do bazy lotniczej Vanderberg w Kalifornii, gdzie w kwietniu tego roku ma nastąpić jej start. Umieści ona na orbicie okołozemskiej kolejnego amerykańskiego satelitę przeznaczony do celów wojskowych.

X X X

Nowym produktem zachodniemieckiej firmy ILC-DDC Elektronik GmbH jest, oznaczona symbolem MEM-8400, statyczna pamięć RAM o pojemności 64Kx16 wykonana technologią CMOS. Zbudowana jest z czterech bloków 32Kx8. Może ona pracować w zakresie temperatur od -55°C do +125°C, co w pełni odpowiada nowej wojskowej normie MIL-STD-883. Mały pobór prądu

Tegoroczny model japońskiego samochodu MAZDA 626 ma zostać wyposażony w system przeciwdziałający poślizgowi kół podczas hamowania. System ten jest efektem pięciu lat wspólnych prac badawczych firm MAZDA, SUMITOMO, LUCAS GIRLING i został wypróbowany zarówno w Japonii, jak i Europie. Jest to układ czterokanałowy nadzorujący indywidualnie każde z czterech kół pojazdu. Stałe ciśnienie w układzie zapewnia pompa napędzana silnikiem elektrycznym. Każdy z członów sterujących wyposażony jest w dwa mikroprocesory sterujące pracą układu i kontrolujące wzajemnie swoje działanie.

■ ■ ■

Dużym zaskoczeniem dla uczonych, była analiza danych o komecie HALLEYA dostarczonych przez zachodnioeuropejską sondę kosmiczną OIOTTO, która przeleciała w odległości 600 km od jej jądra. Okazało się, że w atmosferze komety występuje trzy razy więcej izotopu wodoru - deuteru, niż w atmosferze Jowisza i Saturna. Zawartość deuteru jest w komecie taka sama jak w wodrze spotykanym

umożliwia zapamiętywanie danych przy zewnętrznym zasilaniu baterią 2V. Czas dostępu wynosi 100 nanosekund. Pamięć ta, w połączeniu z 16- lub 32-bitowym mikroprocesorem znajduje zastosowanie między innymi w układach grafiki komputerowej.

X X X

Chemikom z amerykańskiego uniwersytetu RHODE ISLAND udało się wynaleźć polimer przewodzący prąd elektryczny i zmieniający swą barwę pod wpływem zmian jego natężenia. W stanie normalnym polimer jest bezbarwny, dopiero pod działaniem prądu staje się najpierw zielony, a później niebieski i fioletowy. Materiałem tym zainteresowali się już specjaliści planując skonstruowanie polimerowego ekranu telewizyjnego. Nowy materiał zostałby umieszczony pomiędzy dwiema szklanymi taflami pokrytymi drobną siatką przewodów elektrycznych. Dostarczany za jej pośrednictwem prąd zmieniałby w określonych miejscach barwę takiego ekranu.

na Ziemi. w skałach księżycowych oraz meteoroidach. Jest to ważne spostrzeżenie z punktu widzenia kosmochemii. Wysłunięto hipotezę, że planety grupy ziemskiej mogły formować się w tej samej części protosłonecznego obłoku, co komety. Odkrycie to rzuca nowe światło na problem powstania i ewolucji naszego Układu Słonecznego.

■ ■ ■

Na uniwersytecie w Birmingham (Wielka Brytania) opracowano nową technologię jednoczesnego wytapiania kilku metali z rud wieloskładnikowych. Wytapianie odbywa się w specjalnym wielopoziomym piecu, w którym podgrzewa się rudę do temperatury około 1300°C. Podstawą nowej technologii jest recykulacja stopionej rudy pomiędzy kolejnymi poziomami pieca, która, dając ciągłą reakcję chemiczną, pozwala na wyodrębnienie poszczególnych metali z rudy. Przewiduje się, że wdrożenie tej technologii spowoduje ponowny rozkwit hutnictwa metali nieżelaznych w XXI wieku. Umożliwi bowiem wykorzystanie olbrzymich pokładów rud wieloskładnikowych o strukturze drobnoziarnistej, które nie mogły być dotychczas eksploatowane.

Doczekaliśmy się zdrowej konkurencji

Rozmowa z dyrektorem Biura Organizacji Importu i Eksportu — Waldemarem ZNAMIEROWSKIM.



● Rynek informatyczny zapełniony już jest podobnymi, jak wy firmami, czy uważa Pan, że zdołacie się przebić.

— To chyba zdrowy objaw prawdziwej konkurencji, nawet tu na Chłodnej działają obok siebie trzy spółki zajmujące się informatyką. Nie przeszkadzamy sobie. Nasi kontrahenci stają się wybredniejsi, zmuszeni zatem jesteśmy nie tylko do rozszerzania oferty, ale i do doskonalenia jakości naszych usług i to decyduje o powodzeniu.

● Czy konkurencja wpływa także na ceny?

— Sprzętu już tak, choć nie zawsze ceny idą w dół. W zasadzie utrzymują się na stałym poziomie, co przy zmiennym przecież kursie złotówki jest dowodem względnego spadku cen. Ale klienci na ogół poszukują nowości.

● To znaczy?

— Coraz częściej odnotowujemy zainteresowanie sieciami. To zjawisko dobrze świadczy o rozwoju naszej informatyki. Myślę, że moda na „komputer” przemija już bezpowrotnie. Częściej informatyka ma być tą dziedziną działalności, która ułatwia funkcjonowanie przedsiębiorstw, a tego nie załatwi przecież spektakularnie zainstalowany mikrokomputer.

● Co zatem proponujecie?

— Przede wszystkim oczywiście sprzęt zgodny ze standardem IBM. Oferujemy sieci Ethernet, Delink, Novell, Arcnet w zależności od przeznaczenia i możliwości, a także wielodostęp. Pod tym względem nie różnimy się od innych tego rodzaju firm.

● I to ma skłonić potencjalnego klienta do wyboru właśnie was?

— W tej branży decyduje przede wszystkim jakość obsługi, sprawność serwisu, niezawodność proponowanego sprzętu. Staramy się sprzedawać komputery jednego producenta — to znacznie ułatwia sprawy serwisowe. Nie bez znaczenia jest też fakt, że za rok, dwa nasz pierwszy klient wróci do nas po kolejne zakupy.

● Jaką dajecie gwarancję, że będziecie istnieć.

— My chcemy istnieć, ale gwarancji dać nie możemy, bo za dwa miesiące mogą zmienić się przepisy finansowe i zbankrutujemy.

● Krucha ta perspektywa.

— Ale prawdziwa.

● Co zatem w najbliższej przyszłości?

— Instalujemy kilka sieci komputerowych wraz z oprogramowaniem, ale są to instalacje lokalne. Trzeba pamiętać, że im sieć bardziej rozbudowana, a poszczególne jej elementy od siebie bardziej odległe, jej cena szybko rośnie. W tym przypadku decyduje o tym cena... kabla — jeden metr 15 dolarów.

● Przecież można wykorzystać istniejącą już sieć telekomunikacyjną.

— Tak, ale nie tej jakości...

● Rozumiem, że ubogim klientom nie dajecie żadnych szans.

— Jesteśmy na etapie, w którym efekty wprowadzania informatyzacji już widać. Nie będzie w tej dziedzinie rewolucji, ale i od informatyki nie uciekniemy. Dlatego mało kto boi się informatyki i inwestycji w tej dziedzinie — to konieczność — a poza tym

zmieni się wkrótce jakość świadczonych usług.

● To znaczy?

— Myślę, że już niedługo większą wagę przywiązywać się będzie do oprogramowania. Fala inwestycji sprzętowych już przechodzi punkt kulminacyjny. Zresztą na całym świecie koszty sprzętu są jedynie mniej więcej jedną trzecią wydatków na informatykę — reszta to oprogramowanie.

● U nas oprogramowanie jest za darmo.

— Albo kosztuje krocie. Nie ma tu reguły, choć obowiązują dwa krańcowo różne podejścia. Prawo autorskie nie chroni oprogramowania, a zatem kopiuje się je bezkarnie i za darmo, albo dokonuje się drobnych przeróbek i przyznaje do jego autorstwa.

● Tak jest w przypadku oprogramowania narzędziowego i standardów.

— Prawo autorskie nie chroni również programów naszych polskich informatyków. Sądzę jednak, że wkrótce sprawy te zostaną uregulowane.

● Macie jakieś asy w rękawie.

— Stacje teleksowe sprzężone z komputerem personalnym klasy IBM. Handlujemy też sprzętem wideo, ostatnio sprowadziliśmy ekrany wizyjne (Cine Vision)

● Przyznaję, że jest to duży rozrzut zainteresowań.

— Tak, ale my proponujemy zawsze najwyższą jakość usług.

● Czego Wam i klientom życzę.

Wiesław Cetera



(Foto: J. Zelman)

ODRA — Amstrad

Wyposażenie uczelni w mikrokomputery ośmiobitowe, nawet takie jak AMSTRAD CPC-6128, w niewielkim stopniu rozwiązuje informatyzację (z punktu widzenia zaopatrzenia w sprzęt) zarówno dydaktyki, jak i procesów zarządzania — ze względu na małą pojemność pamięci (operacyjnej i dyskowej) oraz niezależną pracę każdego zestawu mikrokomputerowego.

Także eksploatowany dotychczas (rozbudowany) sprzęt komputerowy ODRA — 1305 pracujący pod kontrolą Systemu Operacyjnego (SO) GEORGE — 3 nie spełniał całkowicie związanych z nim oczekiwań. Powodem było skromne (jest to chyba problem wszystkich ośrodków obliczeniowych) wyposażenie w sprzęt teletransmisji — Urządzenia Przetwarzania Danych (UPD), modemy, a przede wszystkim w terminale — drukarki (DZM) lub monitory z klawiaturą.

W tej sytuacji zdecydowano się połączyć mikrokomputery z ODRA, nie tylko w celu uzyskania dodatkowych terminali do pracy w wielodostępie pod kontrolą SO GEORGE-3, ale także z zachowaniem niezależnej pracy AMSTRADA — z możliwością przepisywania danych między dyskietką a pamięciami masowymi ODRY.

Sprzężenia takiego dokonano przez szeregowy interface oraz odpowiednie oprogramowanie — emulator „LIDIA”. AMSTRAD CPC-6128 stał się standardową asynchroniczną dialogową końcówką ODRY — taką jak DZM. Wymuszenie (w czasie pracy pod kontrolą SO GEORGE-3) przerwania ([CONTROL] ^ K) umożliwiło wybranie kierunku przesyłania danych, z dyskietki AMSTRADA do Pamięci Zbiorów Systemu (PZS) GEORGE-3 — komendą INPUT lub w drugą stronę — z PZS na dyskietkę — komendą LISTFILE.

Takie rozwiązanie umożliwia przygotowanie danych do przetwarzania na dyskietkach z wykorzystaniem mikrokomputerów rozproszonych (w pomieszczeniach wykonawców, użytkowników lub w domu), a następnie przepisywanie zawartości dyskietki — z wydzielonych stanowisk — do pamięci masowych ODRY. Także wyniki przetwarzania (na ODRZE) mogą być przekazywane użytkownikom — przez transmisję w drugą stronę — z PZS na dyskietkę AMSTRADA, a nie jak dotychczas — w postaci wydruku.

Podstawową formą wykorzystania AMSTRADA będzie jednak traktowanie tego urządzenia jako terminalu do typowej pracy pod kontrolą SO GEORGE-3:

- realizacja zadań programowych (edycja zbiorów, kompilacje, testowania programów)
- inicjowanie i kontrola wykonania prac użytkowych.

Uruchomienie pracy AMSTRADA pod kontrolą SO GEORGE-3 następuje przez:

- uruchomienie systemu CPM+
- ustawienie parametrów transmisji (SETSIO, BITS, STOP, PARITY)
- uruchomienie emulatora.

Praktycznie ładowany jest tylko CPM+. Pozostałe operacje przebiegają zgodnie z komendami założonego na dyskietce zbioru PROFILE.SUB.

Informacje praktyczne

- zbiory do zapisu (odczytu dyskietki nie mogą przekraczać jej pojemności 180 kbajtów)

- długość rekordu zbiorów do zapisu (odczytu z dyskietki nie może przekraczać 255 bajtów)

— czas przepisywania danych zależy od obciążenia SO GEORGE-3, jest jednak długi. Przepisywanie pełnej dyskietki (180 kbajtów) trwa od 20 do 30 min (przy prędkości 4800 bodów)

— wykorzystując mikrokomputer jako typowy terminal SO GEORGE-3, nie odczuwa się różnicy w szybkości pracy — w porównaniu z pracującymi jednocześnie innymi urządzeniami (DZM, monitory)

— dostępne prędkości przesyłania informacji — od 150 do 9600 bodów, zależą w praktyce od parametrów technicznych stosowanego sprzętu. Sprzężenie zrealizowano na prędkościach 1200 i 2400 bodów.

- formaty zbiorów:
 - na dyskietce — zbiory tekstowe
 - w PZS — zbiory formatu graficznego (CR, LP, TP, TR)
- sprzęt i oprogramowanie:
 - komputer ODRA-1305 pracujący pod kontrolą SO GEORGE-3

— urządzenia teleprzetwarzania dla komputera ODRA (multiplexer, UPD i lub modemy)

— mikrokomputer AMSTRAD CPC-6128 z minimum jedną (wbudowaną) stacją dysków pracujący pod kontrolą systemu CPM+

— interface RS 232-AC produkowany przez AGROKOMPATER w Warszawie

— emulator „LIDIA” wykonany przez mgr inż. Lidę Stasiewicz i mgr inż. Pawła Stasiewicza (WPIPB ETOB w Warszawie)

— wykonawca sprzężenia — realizacja „pod klucz”, interface + emulator — spółka z o.o. SYKOMAT w Warszawie

— koszty (grudzień 1987 r.):

— interface — 140 tys. zł dla każdego mikrokomputera

— emulator — 150 tys. zł gdy praca na jednym stanowisku (mikrokomputerze AMSTRAD) lub 300 tys. zł przy dowolnej liczbie terminali.

W ramach tych kosztów realizowane jest wykonanie sprzężenia oraz zawarta jest roczna gwarancja.

— sprzężenie praktycznie zrealizowano z wykorzystaniem multiplexera i SO GEORGE-3. Można je wykonać przez inne urządzenia (np. skaner), pod kontrolą innego oprogramowania ODRY (MINIMOP) lub dowolnego, własnego oprogramowania np. multiplexera.

Wnioski

Sprzężenie komputera ODRA-1305 z mikrokomputerem AMSTRAD CPC-6128 zrealizowano w końcu 1987 roku w Ośrodku Obliczeniowym Wojskowej Akademii Politycznej. AMSTRAD nie jest droższym terminalem ODRY od dotychczas wykorzystywanych DZM lub monitorów MERA 7952 i jest dostępny na rynku.

AMSTRAD wykorzystywany jest w pracy pod kontrolą SO GEORGE-3 w każdej sesji — nawet chętniej niż inne terminale — przede wszystkim ze względu na klawiaturę. Edycja wprowadzanych informacji nie pozostawia na ekranie aktualizowanych (zmienianych, kasowanych) danych.

Należy nadmienić, że sprzężenie to rozbuduje również możliwości mikrokomputera dzięki dostępowi do:

- pamięci masowych ODRY
- szybkiej (1000 wierszy/min) drukarki wierszowej.

oprac.

Krzysztof ŚWIĘTOŃ



Pamięć taśmowa

Tomasz MROWIEC, Ludwik Pielą

Po wyłączeniu zasilania zawartość pamięci wewnętrznej komputera, zwykle półprzewodnikowej, ginie. Trudno sobie wyobrazić, aby za każdym razem, po włączeniu zasilania, użytkownik wprowadzał program za pomocą klawiatury. Podobna sytuacja jest ze zbiorami danych, o znacznej nieraz objętości. Dlatego niezbędna jest dodatkowa pamięć, której zawartość może być zachowana nawet po wyłączeniu zasilania i szybko wprowadzona do pamięci operacyjnej, po ponownym włączeniu komputera.

Najprostszą i najtańszą możliwość przechowywania programów i danych stwarza standardowa kaseeta magnetofonowa. Składa się ona z małego, plastikowego pojemnika o wymiarach 57x100 mm, zawierającego dwie szpule z nawiniętą taśmą o długości około 90 m i szerokości 4 mm. Otwór w bocznej krawędzi kasety umożliwia przesuwanie taśmy. Otwory w szpulach pozwalają na prowadzenie osi napędowych. Powszechnym rozwiązaniem mechanizmu przesuwu taśmy jest zastosowanie wałka napędowego i rolki dociskowej, które zapewniają stałą prędkość ruchu taśmy.

Zapis i odczyt informacji odbywa się podobnie jak w magnetofonie. Część producentów umożliwia współpracę komputera domowego ze zwykłym magnetofonem kasetowym. Dzięki temu użytkownik uzyskuje pamięć zewnętrzną minimalnym kosztem. Magnetofon może być najczęściej bezpośrednio podłączony do komputera, bez konieczności stosowania dodatkowych urządzeń i umożliwia zarówno nagrywanie programów na kasecie, jak i wprowadzanie ich do pamięci wewnętrznej komputera.

Niektóre komputery domowe, wśród nich również Atari, mogą współpracować tylko ze specjalnie do tego przeznaczonymi magnetofonami. Zapewnia to znacznie większy komfort obsługi (programowe włączanie i zatrzymywanie silnika) i większą niezawodność przesyłania programów i danych między komputerem i magnetofonem.

Podstawowymi wadami tej pamięci są: szeregowy rejestracja informacji oraz bardzo mała szybkość przesyłania. Dlatego niezwykle długi, jak na wymagania komputera, jest czas wyszukiwania potrzebnej informacji na taśmie. Komputer potrafi znaleźć program lub dane utrwalone na nośniku magnetycznym, ale szukanie polega na sprawdzaniu zawartości w takiej samej kolejności i z taką samą szybkością, w jakiej realizowany był zapis. Dlatego, między innymi, specjalne kasetki przeznaczone do rejestracji programów zawierają krótszą taśmę magnetyczną niż kasetki do nagrań muzycznych. Znacznie mniej czasu zajmuje wymiana kaset niż wyszukanie potrzebnego programu na długiej taśmie. Mniejsza jest także szansa nierówności przesuwu na skutek oporów kaset.

Z tych względów nie polecamy nikomu używania taśmowej pamięci magnetycznej do poważniejszych prac, zarówno amatorskich, jak i profesjonalnych. Jednak rzeczywistość jest okrutna, cena stacji dysków (pamięci zewnętrznej o dostępie bezpośrednim) znacznie przewyższa cenę popularnego komputera ośmiobitowego. Dlatego większość krajowych posiadaczy komputerów domowych wykorzystuje pamięć taśmową, której cena nie przekracza połowy ceny komputera.

Jak już wspomnieliśmy, ośmiobitowe komputery Atari w zasadzie współpracują ze specjalnie skonstruowanym magnetofonem firmowym. Piszemy w zasadzie, ponieważ naszym rodakom, chyba jako jedynym na świecie, udało się dostosować do współpracy zwykłe magnetofony, zarówno stereo — (zalecane), jak i monofoniczne. Za ułamek ceny oryginalnego urządzenia, o nazwie Program Recorder, można nabyć specjalny układ pośredniczący, który umożliwia wykorzystanie zwykłego magnetofonu jako taniej pamięci zewnętrznej.

Podobnie do różnorodnych sposobów organizacji wewnętrznej komputerów, zbudowanych często na podstawie tego samego mikroprocesora, istnieją różnorodne techniki zapisywania informacji na taśmie magnetycznej. Jest to rezultat fantazji projektantów w zakresie przyjętych rozwiązań układowych i programowych oraz braku jakichkolwiek standardów.

Atari współpracuje z pamięcią taśmową za pośrednictwem układu **POKEY**. Zapis i odczyt informacji realizowany jest przez odpowiednie procedury systemu operacyjnego. W odróżnieniu od innych rodzajów pamięci, obsługa magnetofonu, pomimo programowego sterowania włączaniem i wyłączaniem silnika, wymaga znacznego udziału użytkownika. Niezbędne jest odpowiednie ustawienie taśmy oraz włączenie jednego lub dwóch klawiszy, w zależności od rodzaju wykonywanej operacji.

System operacyjny zapisuje informacje blokami o stałej długości z szybkością 600 bodów (fizycznych bitów/s). Do przesyłania danych między komputerem i magnetofonem używana jest szeregowy transmisja asynchroniczna. Układ **POKEY** rozpoznaje każdy przesyłany bajt w następującej kolejności: 1 bit startu („0” logiczny), 8 bitów danych oraz bit stopu („1” logiczny). Bajt wysyłany jest i odbierany od najmniej znaczącego bitu. Do reprezentowania „1” logicznej używana jest częstotliwość 5327 Hz, do reprezentowania „0” logicznego — 3995 Hz. Postać bajtu danych jest następująca:

s 0 1 0 1 1 0 0 1 s

Każdy blok składa się ze 132 bajtów. Wy różnia się w nim: 2 znaki wskaźników, bajt

kontrolny, 128 bajtów danych i bajt sumy kontrolnej.

Znaki wskaźników mają kod szesnastkowy 55. Wraz z bitami startu i stopu każdy wskaźnik ma długość 10 bitów. Służą do pomiaru szybkości transmisji. Zakłada się, że transmisja wejściowa odbywa się z nominalną szybkością 600 bodów. Do takiej szybkości dostosowane są układy odbiorcze. Jednak szybkość faktyczna nie jest stała, zależy od odchyłek w szybkości silnika, napięciu taśmy, itp. Jest ona zatem mierzona przez odpowiednią procedurę SIO. Po określeniu rzeczywistej szybkości transmisji wejściowej odpowiednio dostrajane są układy odbiorcze. W ten sposób może być obsługiwana transmisja wejściowa o szybkości zmieniającej się od 318 do 1407 bodów. Pomiar szybkości wykonywany jest dla każdego bloku odczytywanego z taśmy.

*Bajt kontrolowany przyjmuje jedną z trzech wartości:

—\$FC wskazuje, że blok zawiera dokładnie 128 bajtów danych.

—\$FA wskazuje, że blok jest częściowo wypełniony; zawiera mniej niż 128 bajtów danych. Taki przypadek może wystąpić tylko w bloku przed końcem zbioru. Aktualna liczba bajtów danych, od 1 do 127, zapamiętana jest w ostatnim bajcie przed sumą kontrolną, tj. w 128 bajcie danych.

—\$FE wskazuje, że jest to ostatni blok w zbiorze i występuje w nim 128 bajtów zerowych.

Bajt sumy kontrolnej zawiera sumę wszystkich innych bajtów w bloku, włączając oba znaczniki. Suma obliczana jest z krążącym przeniesieniem, to znaczy, w miarę dodawania wartości nowych bajtów, dodawany jest także bit przeniesienia.

Jest to przyczyną potencjalnych problemów. Oprogramowanie, które zapisuje na taśmę, musi zostawić wystarczająco długie przerwy, aby początki bloków nie były губione podczas wprowadzania.

W celu odróżnienia jednego bloku od drugiego procedura obsługi magnetofonu dodaje przerwy międzyblokowe (IRG). Długość IRG zależy od trybu otwarcia zbioru do zapisu. Istnieją dwa typy IRG: normalna i skrócona. Normalna przerwa międzyblokowa używana jest do odczytu przemieszane go z przetwarzaniem, tzn. taśma zawsze zatrzymuje się po odczytaniu każdego bloku. Jeśli komputer zatrzyma taśmę i wykona swoje obliczenia wystarczająco szybko, może się zdarzyć, że następny rozkaz odczytu pojawi się przed zatrzymaniem taśmy. W trybie skróconego IRG taśma nie jest zatrzymywana między blokami, ani podczas zapisu, ani podczas odczytu. Powszechnym zastosowaniem tego trybu jest przechowywanie programów w języku BASIC za pomocą instrukcji **CSAVE**.

Pamięć zewnętrzna może być używana do przechowywania programów i zbiorów danych. Zbiory danych dzielą się na rekordy. Liczba rekordów ograniczona jest długością taśmy. Rekord może mieć dowolną liczbę pół, chociaż wszystkie rekordy w tym samym zbiorze powinny mieć taką samą liczbę pół. Jeśli pole nie jest używane, może mieć wartość zerową lub być puste. Dla oznaczenia końca zbioru komputer zapisuje specjalny rekord.

Zbiory danych zapisane na taśmie magnetycznej mają jedno poważne ograniczenie: dane w nich zawarte mogą być osiąga-

ne tylko sekwencyjnie. Oznacza to, że musimy zaczynać czytanie od początku zbioru i czytać do końca. Nie możemy również dawać, ani kasować rekordów.

Komputer nie wysyła danych do i z magnetofonu znak po znaku, lecz wykonuje to w 128 bajtowych blokach. Rekord może zajmować więcej niż jeden blok, dokładnie jeden blok lub mniej niż jeden blok, w zależności od jego długości. Ogólnie mówiąc, program nie martwi się o to jak blokowane są rekordy danych. Komputer wykonuje to automatycznie.

Atari komunikuje się z wszystkimi urządzeniami pośrednio, za pomocą kanałów wejścia/wyjścia. Istnieje osiem kanałów, numerowanych od 0 do 7. W celu komunikowania się z określonym urządzeniem, program najpierw podłącza urządzenie do jednego z kanałów. Jeśli program żąda wprowadzenia danych z tego kanału, nadszły one z żądanego urządzenia. Podobnie, program kieruje wyprowadzane dane do kanału, który przesyła je do wybranego wcześniej urządzenia wyjściowego.

BASIC rezerwuje kanały o numerach 0, 6 i 7 do określonych działań. Kanał nr 0 jest stale zarezerwowany dla edytora. Instrukcje wprowadzania z klawiatury, w trybie bezpośrednim, używają kanału nr 0. Wszystkie proste instrukcje INPUT i PRINT używają kanału 0. Kanał 7 używany jest do pewnych operacji drukarki oraz wprowadzania i przechowywania programów. Specjalne instrukcje graficzne używają kanału nr 6. Całkowicie dostępne dla programów są kanały o numerach od 1 do 5; kanały 6 i 7 — w ograniczonym zakresie.

Instrukcja OPEN podłącza kanał do wybranego urządzenia. Kolejne instrukcje wprowadzania lub wyprowadzania mogą mieć wtedy dostęp do tego urządzenia poprzez numer kanału. Postać instrukcji OPEN jest następująca:

OPEN #k, a, t, nazwa

Pierwsza liczba po słowie OPEN jest numerem kanału ($0 < k < 7$).

Ostatni parametr jest nazwą urządzenia (np. C:, E:, S:, P:).

Druga liczba określa rodzaj działania (akcji), które będzie dozwolone w kanale. Może to być wprowadzenie (4), wyprowadzenie (8) lub oba jednocześnie (12). Jednak działanie musi mieć sens. Nie można, na przykład, otworzyć klawiatury dla wyprowadzania.

Jest jeszcze jeden parametr, tuż przed nazwą urządzenia. Jest on używany w zależności od urządzenia. Dla klawiatury (K:) oraz edytora (E:) jest ignorowany. Dla ekranu (S:) może wybierać tryb wyświetlania, tekstowy lub graficzny. W przypadku pamięci taśmowej parametr ten określa wielkość przerwy międzyblokowej (IRG). Zerowa wartość parametru wybiera normalną przerwę międzyblokową, a 128 — skróconą.

Kanał pozostaje otwarty do zakończenia działania programu lub wyraźnego zamknięcia go. Gdy program kończy działanie instrukcją END, zamykane są wszystkie kanały. Lecz jeśli zatrzyma się w wyniku realizacji instrukcji STOP, naciśnięcia klawisza BREAK lub błędu, wszystkie otwarte kanały

pozostają otwarte. Instrukcja CLOSE zamyka tylko jeden kanał.

Informacja w zbiorze taśmowym staje się dostępna po otwarciu zbioru i pozostaje dostępna do czasu jego zamknięcia. Zbiór taśmowy nie może być jednocześnie otwarty dla wprowadzania i wyprowadzania.

Po napotkaniu instrukcji OPEN komputer sygnalizuje ten fakt dźwiękiem; jeden sygnał dla wprowadzania, dwa — dla wyprowadzania. Następnie czeka na ustawienie taśmy w odpowiedniej pozycji i naciśnięcie dowolnego klawisza. Dopiero wtedy rozpoczyna się ruch taśmy i komputer odczytuje lub zapisuje 20-sekundowy sygnał pilotujący. W tym czasie nie wykonuje on innych instrukcji.

Ważne jest zamknięcie nieaktywnego zbioru, ponieważ uwalnia kanał. Szczególnie ważne jest zamknięcie zbioru, który został otwarty do zapisu. Ominięcie tego może spowodować utratę danych. Bufor magnetofonu może być częściowo wypełniony danymi. Zamykanie zbioru powoduje wyprowadzenie tych danych z buforu do ostatniego bloku na taśmie. Jeśli zbiór nie zostanie zamknięty, zawartość częściowo wypełnionego buforu nigdy nie zostanie wyprowadzona.

Dane mogą być wyprowadzane do zbioru za pomocą instrukcji PRINT# lub PUT. Obie instrukcje wysyłają dane do otwartego kanału. Nie jest ważne do jakiego urządzenia podłączony jest kanał.

Instrukcja PRINT# wyprowadza wartości numeryczne i tekstowe w kodzie ATASCII. Dla rozdzielenia pozycji należy używać raczej średników niż przecinków. Przecinki są dozwolone, lecz powodują wyprowadzenie dodatkowych spacji.

Każda oddzielna wartość wysyłana do magnetofonu musi kończyć się znakiem EOL (end of line). Podczas wprowadzania znak EOL określa, gdzie ona się kończy. Jednym ze sposobów zapewnienia, że wystąpi znak EOL, jest wyprowadzenie każdej wartości oddzielną instrukcją PRINT#. Nigdy nie należy jej kończyć średnikiem lub przecinkiem. Innym sposobem wyprowadzenia znaku EOL jest funkcja CHR\$ (program nr 1).

```
FK 1 REM *****
FJ 2 REM * *
XA 3 REM * Program nr 1 *
FL 4 REM * *
FO 5 REM *****
NL 6 REM
II 10 PRINT "Ustaw tasme i nacisn
ij <RETURN>";
YU 20 OPEN #3,8,0,"C:"
QY 30 PRINT #3;A;CHR$(155);B;CHR$(155);C
MG 40 CLOSE #3
```

Każda instrukcja PUT wyprowadza pojedynczą wartość numeryczną od 0 do 255. Zajmuje ona tyle samo miejsca co jeden

```
FK 1 REM *****
FJ 2 REM * *
XU 3 REM * Program nr 2 *
FL 4 REM * *
FO 5 REM *****
NL 6 REM
II 10 PRINT "Ustaw tasme i nacisn
ij <RETURN>";
YU 20 OPEN #3,8,0,"C:"
PV 29 REM Wyprowadzenie pustego r
ekordu.
```

```
SH 30 FOR J=1 TO 128
BS 40 PUT #3,0
JC 50 NEXT J
KA 60 PRINT #3;"Poczatek danych"
MJ 70 CLOSE #3
YW 80 END
```

znak ATASCII. Program nr 2 wyprowadza pusty blok po otwarciu kanału wyjściowego.

Instrukcja INPUT# wprowadza wartości przechowane na taśmie instrukcją PRINT#. Określony przez nią kanał musi być otwarty dla wprowadzania z magnetofonu. Aby zobaczyć jak to działa, użyjmy najpierw programu nr 3 do utworzenia zbioru danych.

```
FK 1 REM *****
FJ 2 REM * *
YO 3 REM * Program nr 3 *
FL 4 REM * *
FO 5 REM *****
NL 6 REM
PX 10 DIM A$(120)
WY 20 PRINT "Jaki komunikat chces
z zapisac?"
YC 30 INPUT A$
XE 40 PRINT
IM 50 PRINT "Ustaw tasme i nacisn
ij <RETURN>";
JK 60 OPEN #3,8,0,"C:"
SN 70 PRINT #3;A$
MK 80 CLOSE #3
YX 90 END
```

Następny program, nr 4, używa instrukcji INPUT# do odczytania komunikatu zapisa-

```
FK 1 REM *****
FJ 2 REM * *
ZI 3 REM * Program nr 4 *
FL 4 REM * *
FO 5 REM *****
NL 6 REM
PX 10 DIM A$(120)
IJ 20 PRINT "Ustaw tasme i nacisn
ij <RETURN>";
WZ 30 OPEN #3,4,0,"C:"
UB 40 INPUT #3;A$
WN 50 PRINT A$
MI 60 CLOSE #3
YV 70 END
```

nego przez poprzedni program. Nie zapomnijmy o przewinięciu taśmy!

Instrukcja INPUT# interpretuje dane ze zbioru taśmowego jako kody ATASCII. Za każdym razem, gdy napotka znak EOL (kod ATASCII 155), podstawia znaki przeczytane od ostatniego EOL do następnej zmiennej w swoim wykazie. Jeśli zmienna jest numeryczna, INPUT# tłumaczy wczytane znaki na wartość numeryczną.

Każda instrukcja GET wprowadza jedną wartość numeryczną. Program musi zdecydować, jak tę wartość interpretować. Na przykład, może on interpretować wartość jako kod ATASCII i zamienić na znak za

```
FK 1 REM *****
FJ 2 REM * *
AC 3 REM * Program nr 5 *
FL 4 REM * *
FO 5 REM *****
NL 6 REM
II 10 PRINT "Ustaw tasme i nacisn
ij <RETURN>";
WY 20 OPEN #3,4,0,"C:"
EK 30 GET #3,A
JM 40 IF A<155 THEN PRINT CHR$(A
);GOTO 30
XF 50 PRINT
MI 60 CLOSE #3
YV 70 END
```

pomocą funkcji CHR\$. Program nr 5 używa funkcji GET do odczytania tego samego zbioru co program nr 4.

Program nr 6 jest odpowiedzią na częste pytania czytelników, dotyczące przechowywania na kasecie zawartości tablic numerycznych. W liniach 20—50 tego programu poszczególnym elementom tablicy A nadawane są wartości losowe z przedziału (0,100). W linii 60 następuje otwarcie zbioru

```
FK 1 REM *****
FJ 2 REM *
AW 3 REM * Program nr 6 *
FL 4 REM *
FO 5 REM *****
NL 6 REM
JL 10 DIM A(10,10),B(10,10)
GC 20 FOR I=1 TO 10
GL 30 FOR J=1 TO 10
MT 40 A(I,J)=RND(0)*100
OW 50 NEXT J:NEXT I
YE 60 OPEN #1,8,0,"C:"
GH 70 FOR I=1 TO 10
GQ 80 FOR J=1 TO 10
JW 90 PRINT #1,A(I,J)
EJ 100 NEXT J:NEXT I
KZ 110 CLOSE #1
CM 120 PRINT "Przewin taśmie i nacisnij <RETURN>."
KD 130 OPEN #2,4,0,"C:"
EU 140 FOR I=1 TO 10:FOR J=1 TO 10
AW 150 INPUT #2;C:B(I,J)=C
EV 160 NEXT J:NEXT I
LL 170 CLOSE #1
FC 180 FOR I=1 TO 10:FOR J=1 TO 10
GL 190 PRINT A(I,J),B(I,J)
EK 200 NEXT J:NEXT I
NT 210 END
```

do zapisu, następnie zapisanie tablicy na taśmie (linie 70—110). Linie 130—170 wprowadzają zapisane wartości do tablicy B. Proszę zwrócić uwagę na kolejność wyprowadzania, a następnie wprowadzania danych. Tablica dwuwymiarowa może być zapisywana (i odczytywana) kolumnami lub wierszami. Jeśli nie będziemy wystarczająco ostrożni, może okazać się, że przypadkowo uzyskaliśmy macierz transponowaną.

Rozważmy praktyczne wykorzystanie zbiorów taśmowych: wykaz adresów bliskich i znajomych. Potrzebne będą dwa programy. Pierwszy będzie służył do wprowadzania nazwisk i adresów z klawiatury i przechowywania ich na taśmie. Drugi będzie wprowadzał nazwiska i adresy z taśmy i wyświetlał je na ekranie.

Każdy z rekordów zbioru adresowego ma pięć pól: imię, nazwisko, ulica, miasto i kod pocztowy. Oba programy używają tych samych nazw zmiennych dla pól: IMS\$, NAZ\$, UL\$, MI\$, KOD\$.

Program nr 7, po zadeklarowaniu zmiennych, otwiera kanał wyjściowy dla magnetofonu i wyprowadza pusty blok. Podczas

```
FK 1 REM *****
FJ 2 REM *
BQ 3 REM * Program nr 7 *
FL 4 REM *
FO 5 REM *****
NL 6 REM
EY 10 DIM PRMT$(40),MSK$(40),BL$(40),ERM$(13)
OP 20 BL$=" ":BL$(40)=BL$:BL$(2)=BL$
WZ 30 MSK$="_:MSK$(40)=MSK$:MSK$(2)=MSK$
GM 40 ERRHDL=8000
SF 50 DIM IMS$(20),NAZ$(20),UL$(20),MI$(20),KOD$(6),R$(20),EOL$(1)
HW 60 EOL$=CHR$(155)
ND 70 GOSUB 3000
IC 90 GOTO 1000
```

```
XA 97 REM *****
XR 98 REM * Kasowanie linii *
XE 99 REM *****
EI 100 FOR J=L1 TO L2
VX 110 POSITION 0,J
ZP 120 PRINT BL$(1,39);
GH 130 NEXT J
ZF 140 RETURN
VX 165 REM *****
FY 166 REM * Komunikat błędu *
WD 167 REM *****
NE 170 FOR J=1 TO 3
MT 180 POSITION 20,0
JV 190 ? ERM$," ERROR";CHR$(253);
SM 200 FOR J1=1 TO 100:NEXT J1
MQ 210 POSITION 20,0
QQ 220 ? BL$(1,19)
GI 230 NEXT J
ZG 240 RETURN
EG 250 L1=20:L2=23:GOSUB 100
OA 260 POSITION 2,20
ZM 270 RETURN
MI 596 REM *****
BK 597 REM * Podprogram *
FK 598 REM * wprowadzania *
MR 599 REM *****
IB 600 TRAP ERRHDL
TO 620 GOSUB 250
AV 630 POSITION IC+1,IR
GB 640 ? MSK$(1,IL)
FE 650 POSITION IC,IR
HD 660 INPUT R$
ZE 665 FOR J1=1 TO IL:IF R$(J1,J1)<>MSK$(1,1) THEN NEXT J1
MI 666 R$=R$(1,J1-1)
TK 670 IF R$="?" THEN GOTO 630
MN 680 IF LO>HI THEN 750
GF 700 IF R$="" THEN R$="0"
UW 710 R=VAL(R$)
BS 720 IF R>=LO AND R<=HI THEN RETURN
```

```
VY 730 ERM$="Zakres liczb"
GP 740 GOSUB 170:GOTO 620
PG 750 IF LEN(R$)<=IL THEN RETURN
US 760 ERM$="Długość ciągu"
GV 770 GOSUB 170:GOTO 620
IK 796 REM *****
TL 797 REM * Wprowadzanie z *
ZF 798 REM * propozycja *
IT 799 REM *****
CU 800 L1=1:L2=2:GOSUB 100
HW 810 POSITION 2,1
AY 820 ? PRMT$
XI 830 IC=2:IR=2:GOSUB 600
ZM 840 RETURN
ZV 1000 ? CHR$(125);"Podaj następujące dane"
XN 1010 ? :? :?
MA 1020 ? "1) Imię:"
PE 1030 ? "2) Nazwisko:"
DJ 1040 ? "3) Ulica:"
JJ 1050 ? "4) Miasto:"
VV 1060 ? "5) Kod:"
WZ 1100 FOR F=1 TO 5
XA 1110 GOSUB 2000
DQ 1120 NEXT F
FI 1130 PRMT$="Czy wprowadzasz zmiany?"
IO 1140 LO=1:HI=0
VD 1150 IL=1:GOSUB 800
MO 1159 REM Analiza odpowiedzi
HW 1160 IF R$="N" OR R$="n" THEN 1300
PB 1170 IF R$="T" OR R$="t" THEN 1200
UL 1180 ERM$="PODAJ T LUB N":GOSUB 170
PQ 1190 GOTO 1130
HV 1200 PRMT$="Ktore pole"
KZ 1210 LO=1:HI=5
UW 1220 IL=1:GOSUB 800
WF 1230 F=R:GOSUB 2000:GOTO 1130
HI 1300 POSITION 8,21
YG 1310 ? " * * * PROSZE CZEKAĆ * * *";
MX 1319 REM Wyprowadzenie rekordu
SI 1320 ? #1;IMS$;EOL$;NAZ$;EOL$;UL$;EOL$;MI$;EOL$;KOD$
IP 1330 ? CHR$(156)
VT 1340 POSITION 2,1
SO 1350 PRMT$="Wprowadzasz dalej?"
```

```
IY 1360 LO=1:HI=0
VN 1370 IL=1:GOSUB 800
MW 1380 IF R$="T" OR R$="t" THEN 1000
SJ 1390 IF R$<>"N" AND R$<>"n" THEN 1350
YT 1400 FOR J=1 TO 5
XO 1410 ? #1;CHR$(253);EOL$;
FO 1420 NEXT J
NA 1430 CLOSE #1
FF 1900 END
IB 2000 LO=1:HI=0
AZ 2010 ON F GOTO 2100,2200,2300,2400,2500
WX 2100 IC=13:IR=4:IL=19:GOSUB 600
MT 2110 IMS=R$:RETURN
XO 2200 IC=13:IR=5:IL=19:GOSUB 600
PK 2210 NAZ=R$:RETURN
YF 2300 IC=13:IR=6:IL=19:GOSUB 600
PK 2310 UL=R$:RETURN
YW 2400 IC=13:IR=7:IL=19:GOSUB 600
MV 2410 MI=R$:RETURN
LI 2500 IC=13:IR=8:IL=6:GOSUB 600
LY 2510 KOD=R$:RETURN
CX 2999 REM Otwarcie kanału wyjściowego
MX 3000 ? CHR$(125);"Przygotuj kasetę i naciśnij RETURN"
BA 3010 OPEN #1,8,0,"C:"
DV 3020 FOR J1=1 TO 120:PUT #1,0:NEXT J1
AK 3030 RETURN
HN 8000 E=PEEK(195)
BT 8010 EL=PEEK(187)*256+PEEK(186)
BC 8020 IF E=3 OR E=8 THEN 8100
VP 8030 ? "ZNALEZIONO BŁĄD";E
YF 8040 ? "W LINII NR. ";EL
FS 8080 END
IF 8100 REM
LW 8110 ERM$="INPUT"
YT 8120 GOSUB 170
VQ 8130 TRAP ERRHDL
KX 8140 GOTO EL
```

wprowadzania rekordów użytkownik może go sprawdzić i poprawić w przypadku błędu.

Po zakończeniu wprowadzania danych program wyprowadza jeden dodatkowy rekord (zawierający specjalne wartości we wszystkich polach), oznaczający koniec zbioru, a następnie zamyka zbiór.

Program nr 8 otwiera kanał wejściowy dla magnetofonu, wprowadza z taśmy po-

```
FK 1 REM *****
FJ 2 REM *
CK 3 REM * Program nr 8 *
FL 4 REM *
FO 5 REM *****
NL 6 REM
JB 50 DIM IMS$(20),NAZ$(20),UL$(20),MI$(20),KOD$(6),R$(20)
NO 70 GOSUB 3100
FV 1000 ? CHR$(125)
TW 1010 ? "KSIĄZKA ADRESOWA"
OV 1020 POSITION 2,21:?"Naciśnij CTRL-1 dla zatrzymania,"
RI 1030 POSITION 0,22:?"Ponowne naciśnięcie wznowi przesłanie."
HM 1050 POSITION 2,2:?"Naciśnij RETURN gdy bedziesz gotowy"
LB 1070 INPUT R$
OV 1080 OPEN #4,8,0,"S:"
LX 1090 INPUT #1,IMS$,NAZ$,UL$,MI$,KOD$
SS 1099 REM Sprawdzenie, czy koniec zbioru
IS 1100 IF IMS=CHR$(253) THEN 1200
GZ 1110 ? IMS$
OM 1120 ? NAZ$
```

```

KO 1130 ? UL$
HE 1140 ? MI$
KH 1150 ? KOD$
KG 1160 ? :?
RY 1170 GOTO 1090
VP 1200 CLOSE #1:CLOSE #4
FF 1900 END
LO 3099 REM Otwarcie kanału dla w
prowadzenia
YV 3100 OPEN #1,4,0,"C:"
UL 3119 REM Wprowadzenie pustego
rekordu
UB 3120 FOR J1=1 TO 128:GET #1,R:
NEXT J1
AM 3130 RETURN

```

szczególne rekordy i wyświetla je na ekranie.

Istnieją trzy instrukcje służące do przechowywania programów w języku BASIC na kasecie: CSAVE, LIST i SAVE. Każda z nich ma swój odpowiednik: CLOAD, ENTER i LOAD, które wprowadzają programy z powrotem do pamięci.

CSAVE jest specjalną instrukcją do przechowywania programów tylko na kasecie. LIST i SAVE wyprowadzają program na dowolne urządzenie, w szczególności na magnetofon. Dla instrukcji LIST i SAVE musimy określić urządzenie, np. SAVE „C:”.

Zarówno CSAVE, jak i SAVE „C:” zawsze przechowują cały program z pamięci. Instrukcją LIST „C:” można przechować całość lub część programu. Podobnie jak dla innych postaci instrukcji LIST, możemy określić pierwszą i ostatnią linię programu, które mają być wyprowadzone (np. LIST „C:”, 1000, 3000).

Instrukcja CLOAD wprowadza programy przechowane na kasecie instrukcją CSAVE. ENTER i LOAD wprowadzają programy z dowolnego urządzenia. ENTER „C:” może wprowadzić tylko programy przechowane za pomocą instrukcji LIST „C:”. LOAD „C:” może wprowadzić tylko programy przechowane za pomocą SAVE „C:”.

Jak widać z powyższego przeglądu, żadna z instrukcji przechowywania programów na kasecie nie wymaga podania nazwy programu. Każdy z programów tworzy na taśmie odrębny zbiór, który nie jest rozróżniany przez komputer. Wprowadzany jest program, na początku którego ustawiona jest taśma magnetyczna. Identyfikacja programu może być dokonana tylko przez użytkownika: albo za pomocą licznika taśmy, albo dopiero po wprowadzeniu programu do pamięci komputera.

Instrukcje CLOAD i LOAD kasują program rezydujący w pamięci przed wprowadzeniem nowego. Instrukcja ENTER łączy wprowadzany program z istniejącym w pamięci. Jeśli wprowadzane wiersze programu mają takie same numery jak istniejące, to te ostatnie są podmieniane przez nowe.

Zapisywanie programu jest zewnętrznie takie samo, bez względu na użytą instrukcję: CSAVE, LIST „C:” lub SAVE „C:”. Lecz każda z nich zapisuje programy w innym formacie.

Instrukcja LIST wyprowadza programy w takim samym formacie bez względu na urządzenie. Wysyła ona kod ATASCII każdego znaku programu.

Instrukcje CSAVE i SAVE podmieniają słowa kluczowe jednoznakowymi symbolami. Różnica między nimi polega na długości przerw między blokami na taśmie. Instrukcja CSAVE zapisuje programy gęściej niż SAVE „C:”, z krótszymi przerwami międzyblokowymi. Zatem przechowywanie i wprowadzanie programów odbywa się trochę szy-

niej za pomocą CSAVE i CLOAD niż za pomocą SAVE „C:” i LOAD „C:”.

Program może być wprowadzony z taśmy magnetycznej do pamięci również za pomocą instrukcji RUN „C:”, która jest połączeniem instrukcji LOAD „C:” i RUN. Zatem programy muszą być przechowane na taśmie instrukcją SAVE „C:”. Instrukcja RUN „C:” działa jednakowo w trybie programowym i bezpośrednim. Program, który zawiera instrukcję RUN „C:”, może wprowadzić i uruchomić inny program. Poważną wadą tego sposobu jest kasowanie wartości wszystkich zmiennych przed wprowadzeniem następnego programu. Dlatego nie można używać wartości, które były wprowadzone lub obliczone przez wcześniejszy program. Dodatkowo zapis na taśmie ma długie przerwy międzyblokowe.

Istnieje sposób automatycznego wprowadzania i uruchamiania programów w języku BASIC zapisanych z krótkimi przerwami międzyblokowymi. Dodatkowo można to zrobić bez podawania jakiegokolwiek instrukcji. Niezbędne jest jednak umieszczenie na taśmie specjalnego programu wprowadzającego, który będzie wprowadzany do pamięci w chwili włączania zasilania, jako część procesu inicjowania systemu. Podczas włączania komputera musi być naciśnięty klawisz START. Po wprowadzeniu programu wprowadzającego przejmuje on sterowanie i wprowadza, a następnie uruchamia program przechowany bezpośrednio za nim.

Program, o którym mowa, można spotkać przed programami w języku BASIC wprowadzonymi poprzez włączenie komputera z naciśniętym klawiszem START (np. Konwerter 80k). Jeśli dysponujemy takim programem, wystarczy skopiować program wprowadzający. Dla czytelników, którzy nie mają skąd wziąć tego programu, przygotowaliśmy **program nr 9**, który po wpro-

```

FK 1 REM *****
FJ 2 REM *
DE 3 REM * Program nr 9 *
FL 4 REM *
FO 5 REM *****
NL 6 REM
KP 8 GRAPHICS 0:POSITION 10,10:7
"Proszę czekać..."
OD 10 DIM A$(270):I=0
GQ 20 READ A:IF A=-1 THEN 100
SJ 30 I=I+1:A$(I,I)=CHR$(A)
RU 40 GOTO 20
TR 100 OPEN #1,8,128,"C:"
PO 110 POKE 766,1:?: #1,A$:POKE 7
66,0
PF 120 CLOSE #1:GRAPHICS 0
EB 130 ? "KONIEC"
TP 140 STOP
ND 2060 DATA 0,2,0,6,16,6,169,60,
141,2,211,169,0,133,65,96,169,
148,141,197
EY 2070 DATA 2,173,48,2,133,128,1
73,49,2,133,129,160,4,177,128,
133,130,200,177,128
ZO 2080 DATA 133,131,230,131,230,
131,160,169,162,0,189,91,6,145
,130,232,136,208,247,169
WR 2090 DATA 13,141,74,3,169,0,13
3,04,169,254,141,179,2,169,127
,141,180,2,76,0
IT 2100 DATA 160,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0
HB 2110 DATA 0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0
AE 2120 DATA 0,0,0,0,0,0,0,0,9,9,
4,0,0,56,0,50,36,33,0,50
QP 2130 DATA 51,53,29,56,26,16,14
,50,39,26,18,17,12,20,22,23,0,
37,43,47

```

```

NS 2140 DATA 48,26,18,17,12,18,20
,24,0,37,43,0,0,47,48,26,2,155
,68,44
IA 2150 DATA 255,137,155,150,0,68
,137,40,52,137,40,151,137,218,
253,130,2,29,4,56
QQ 2160 DATA 26,9,16,20,8,4,56,0,
45,41,36,0,0,0,0,0,0,0,0
HT 2170 DATA 0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0
HW 2180 DATA 0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0
EC 2190 DATA 0,0,-1

```

zeniu i uruchomieniu, tworzy na taśmie magnetycznej właśnie taki program wprowadzający.

Oczywiście program ten może być bardziej rozbudowany i dostosowany do programu, który będzie przez niego wprowadzany. Chcielibyśmy jednak zaproponować czytelnikom program uniwersalny, który umożliwi wprowadzenie do pamięci dowolnego programu w języku BASIC. Nie przeszkadza to w zindywidualizowaniu procesu wprowadzania programu. Zamiast dopasowywania programu wprowadzającego, możemy dopasować proces wprowadzania.

Opisywany problem nie został przez nas wydumany, lecz jest odpowiedzią na liczne listy czytelników, których interesowało automatyczne wprowadzenie i uruchomienie programu oraz wprowadzanie programu po uprzednim wyświetleniu charakterystycznej dla niego czołówki. Pierwsza z tych spraw została już, mamy nadzieję, wyjaśniona.

Druga sprawa jest bardziej skomplikowana. Chcemy uzyskać następujący efekt: po odpowiednim ustawieniu taśmy magnetycznej włączamy komputer z wciśniętym klawiszem START, automatycznie pojawia się czołówka, charakterystyczna dla danego programu, która pozostaje na ekranie do chwili załadowania i automatycznego uruchomienia właściwego programu.

Rozwiązanie powyższego zadania wymaga podjęcia decyzji o sposobie tworzenia czołówki programu. Czy przygotowana będzie przez jeden z ogólnie dostępnych programów graficznych (np. Kolała Micro, Magic Painter, Design Master, itp.), czy utworzona przez własny program?

Rozważmy pierwszy wariant. Proponujemy tworzenie czołówki za pomocą programu Magic Painter. Wybraliśmy go dlatego, że prawdopodobnie jest najbardziej rozpowszechniony. Dwukrotnie emitowany był w audycji „Radiokomputer nocą”, dostępny jest na giełdach i w wypożyczalniach. Program ten umożliwia przygotowanie dowolnej czołówki, a następnie przechowanie jej na taśmie.

Aby mieć wszystkie niezbędne elementy, musimy jeszcze wprowadzić i przechować na taśmie magnetycznej (za pomocą CSAVE) **program nr 10**. Jego zadaniem jest wprowadzenie i wyświetlenie czołówki przy-

```

QQ 1 REM *****
MP 2 REM * Program nr 10 *
QQ 3 REM *****
BT 4 DIM ML$(28),X$(40):GRAPHICS
7+16
QF 5 FOR I=1 TO 28:READ A:ML$(I,I)
)=CHR$(A):NEXT I
ZL 6 DATA 104,104,104,170,104,104
,157,66,3,104,157,69,3,104,157
,68,3,104,157,73,3,104,157,72,
3,76,86,228

```

```

ID 7 DL=PEEK(560)+256*PEEK(561)
MI 8 ADDR=PEEK(DL+4)+256*PEEK(DL+5)
EW 9 POKE 764,12
AF 10 OPEN #1,4,128,"C:"
DY 30 FOR I=0 TO 4
LJ 40 GET #1,A:POKE 708+I,A
IT 50 NEXT I
EG 60 GET #1,B
TQ 70 X=USR(ADR(ML$),16,7,ADDR,4300)
LO 80 CLOSE #1
EX 90 REM LADOWACZ PROGRAMU
LF 100 DATA 162,253,154,169,183,72,169,84,72,169,4,32,182,187,169,255,76,4,187
JX 110 FOR I=1 TO 19:READ A
QI 120 X$(I,I)=CHR$(A)
FX 130 NEXT I
FN 140 POKE 764,12:X=USR(ADR(X$))

```

gotowanej programem Magic Painter (realizuje to procedura przechowywana w zmiennej ML\$), a następnie wprowadzenie i uruchomienie właściwego programu (wykonuje to procedura przechowywana w zmiennej X\$).

Możemy już przystąpić do przygotowania taśmy wynikowej:

1. Zapisujemy program wprowadzający.

2. Zapisujemy, instrukcją CSAVE, program nr 10.

3. Zapisujemy na taśmie obraz czołówki, utworzony programem Magic Painter.

4. Przechowujemy, za pomocą CSAVE, właściwy program.

Powyższe możemy wykonać dowolnym sposobem, ważne jest otrzymanie takiej kolejności poszczególnych zbiorów. Po przevinieciu taśmy możemy sprawdzić uzyskany efekt. Wprowadzanie programu trwa nieco dłużej (sama czołówka wprowadzana jest około 100 sekund), lecz jest bardziej efektowne.

W drugim wariacie czołówka programu tworzona jest przez własną procedurę. Proponujemy włączyć ją do programu nr 10 w miejsce linii wprowadzających dotychczas obraz czołówki. Jej postać zależy od naszych umiejętności oraz wyobraźni. Może to być sam tytuł programu napisany dużymi literami (tryb graficzny 1 lub 2), rysunek wykonany w trybie 0, rysunek w dowolnym trybie graficznym z napisem w oknie tekstowym. Może to być również bardziej rozbudowany obraz, z tytułem napisanym dużymi

literami i stosownym rysunkiem, uzyskany poprzez utworzenie własnego programu wyświetlania.

W tym wariacie na taśmie magnetycznej musimy zapisać następujące zbiory:

1. Program wprowadzający.

2. Zmodyfikowany program nr 10, który tworzy obraz czołówki oraz wprowadza właściwy program.

3. Właściwy program.

Programy wymienione w punktach 2 i 3 powinny być zapisane instrukcją CSAVE.

Zauważmy, że procedura przechowywana w zmiennej X\$ programu nr 10, może być wykorzystana do łączenia programów zapisanych na taśmie za pomocą CSAVE. Podobnie jak w przypadku używania instrukcji RUN „C:”, program nie może używać wartości, które były wprowadzone lub obliczone przez wcześniejszy program.

Podczas przygotowania niniejszego artykułu korzystaliśmy z następujących publikacji: De RE Atari oraz Your Atari Computer.

**Tomasz MROWIEC
Ludwik PIELA**

PROGRAM PROGRAM 42

ZX SPECTRUM

TABELARYZCJA

```

100 REM program TABELA
110 BORDER 4: DIM v(255): DIM v$(255): DIM x(255): DIM c$(32)
120 INPUT PAPER 7:"funkcja y":
PAPER 4:"=":"LINE f$:" IF f$=""
THEN GO SUB 580: GO TO 120
130 INPUT "x min = ";xmin,"x max = ";xmax
140 IF xmax<xmin OR xmax=xmin THEN GO SUB 580: GO TO 130
150 CLS: LET st=(xmax-xmin)/254
160 REM oblicz. tab. x() i v()
170 LET m=1: LET ymin=10+38: LET ymax=-ymin: GO SUB 520: PRINT AT 0,tab;t$
180 FOR z=1 TO 255: LET x=xmin+st*(z-1): LET x(z)=x: LET v(z)=VAL(f$)
190 IF ymin>v(z) THEN LET ymin=v(z)
200 IF ymax<v(z) THEN LET ymax=v(z)
210 PLOT z-1,157: NEXT z
220 BEEP .1,2: GO SUB 590
230 REM proporcje wydruku
240 LET z0=0: LET dy=ymax-ymin
250 PRINT AT 0,tab;t$
260 PRINT #0;AT 0,tab1; BRIGHT
1: PAPER 7:ymin;"<y<";ymax
270 FOR n=1 TO 255: LET x=x(n): LET v$(n)=CHR$(INT((v(n)-ymin)*166/dy)): PLOT n-1,155: NEXT n
280 CLS: PRINT AT 0,tab;t$: PLOT 0,0,CODE v$(1)
290 PLOT 128,81: DRAW 0,4: PLOT 128,83: DRAW 4,0
300 IF xmin<0 AND xmax>0 THEN LET z0=INT(255*(xmin/(xmin-xmax)))
310 PLOT z0,0: DRAW 0,155
320 IF ymax>0 AND ymin<0 THEN PLOT 0,ymin*166/dy: DRAW 255,0
330 PRINT #0;AT 0,tab1; BRIGHT
1: PAPER 7:ymin;"<y<";ymax
340 REM pocz. petli PLOT/DRAW
350 FOR n=2 TO 254
360 IF v(n-1)<0 AND v(n)>0 OR v(n-1)>0 AND v(n)<0 THEN GO SUB 480
370 IF v(n)=0 THEN GO SUB 530
380 IF (v(n-1)<v(n)) AND (v(n)>v(n+1)) THEN GO SUB 510
390 IF (v(n-1)>v(n)) AND (v(n)<v(n+1)) THEN GO SUB 540
400 IF n=z0 THEN GO SUB 550
410 PLOT n-1,0,CODE (v$(n-1)): DRAW 1,0,CODE (v$(n))-CODE (v$(n-1))
420 NEXT n
430 REM nowa funkcja ?
440 GO SUB 580: PRINT #0;AT 1,0: PAPER 7:"nowe klawisze,nowa funkcja, COPY?": PAUSE 0: IF INKEY$="f" THEN GO TO 120
450 IF INKEY$="c" THEN PLOT 0,0: DRAW 255,0: DRAW 0,157: DRAW 255,0: DRAW 0,-157: LPRINT: COPY
460 GO TO 130
470 REM podprogramy
480 BEEP .1,14: FOR q=166 TO 1 STEP -3: PLOT n-1,q: DRAW 0,-1: NEXT q:
490 LET yp=v(n-1): LET yk=v(n): LET xp=x(n-1): LET xk=x(n): LET xz=xp+(xk-xp)*(ABS yp/(ABS yk+ABS yp)): LET xz=.001*INT(1000*xz+.0005): PRINT "x0 =";: IF xz>0 THEN PRINT "+";

```

ZAGADKA

```

1 REM PROGRAM - ZAGADKA
10 LET a$="": FOR n=1 TO 10:
READ data: LET a$=a$+CHR$ data:
NEXT n
15 SAVE a$ LINE 10
20 DATA 22,11,15,19,1,137,8,21
,1.79

NAGRAJ PROGRAM NA TAŚMĘ, WYKACZ
KOMPUTER I WPROWADZ PROGRAM
Z TAŚMY. SKAD WZIAK SIE ZNAK
PO ŚRODKU EKRANU ?
A CZEKAMY NA LISTY CZYTELNIKÓW !
    
```

ZNAKI x 24

```

10 BORDER 4: PAPER 4: PAUSE 0:
LET a=CODE INKEY$: CLS: PRINT
AT 21,0: INK 4: CHR$ a: LET a$=CH
R$ a+CHR$ a+CHR$ a
20 BEEP .1,2: FOR n=7 TO 0 STE
P -1: FOR m=0 TO 7
30 IF POINT (m,n)=1 THEN PRINT
BRIGHT 1: PAPER 7: AT 18-3*n,4+3
*m;a$: AT 19-3*n,4+3*m;a$: AT 20-3
*n,4+3*m;a$
40 NEXT m: NEXT n: GO TO 10
50 SAVE "24 x" LINE 10

Program wyświetla znaki
w 24-krotnym powiększeniu
    
```

J. RUBINOWICZ

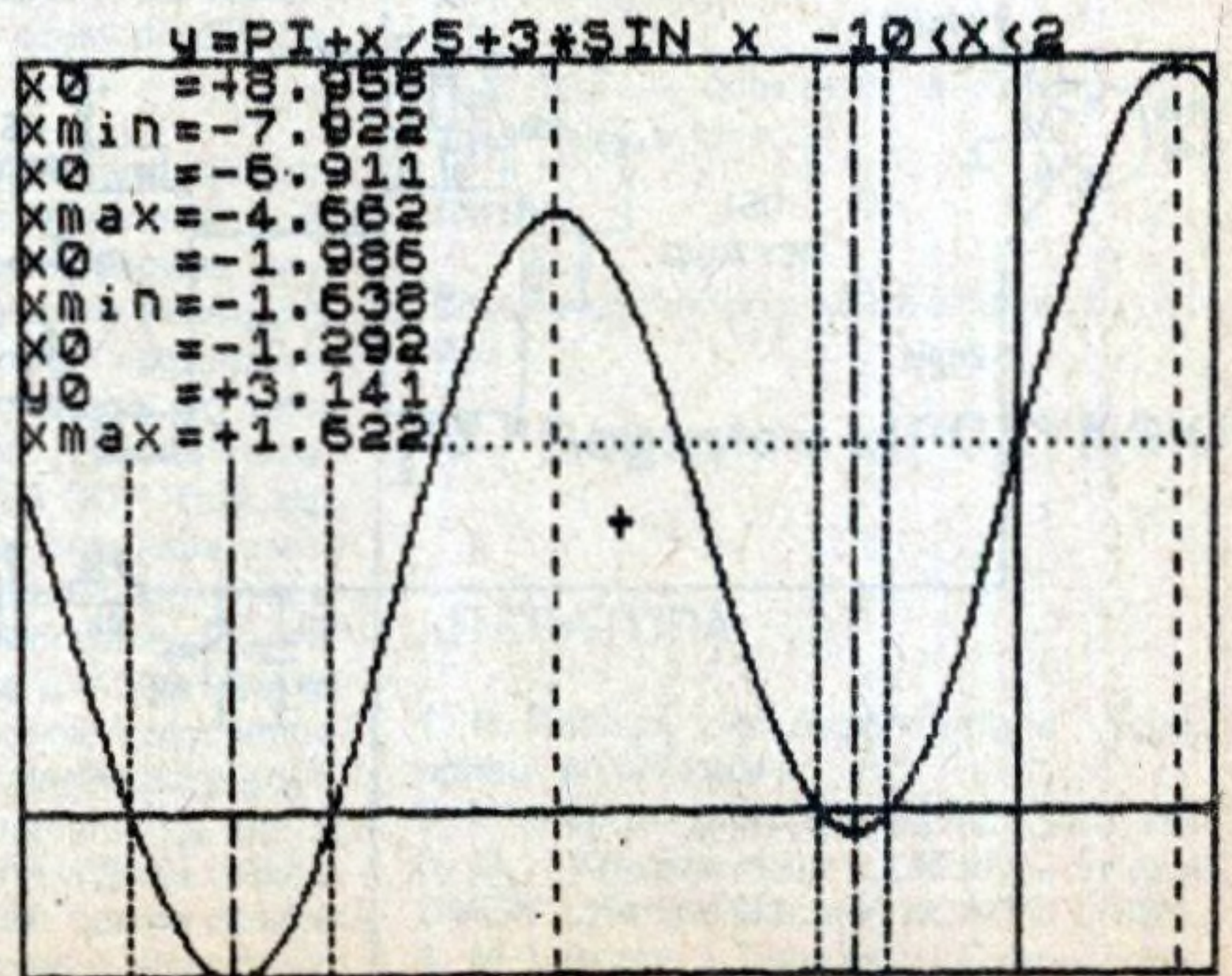
```

495 IF xz>0 AND xz<.1 THEN PRIN
T "0";
500 PRINT xz: RETURN
510 BEEP .1,25: FOR q=166 TO 1
STEP -6: PLOT n,q: DRAW 0,-2: NE
XT q
520 PRINT "xmax=";: GO SUB 0560
: RETURN
530 BEEP .1,38: FOR q=166 TO 1
STEP -2: PLOT n,q: NEXT q: PRINT
"x0=";: GO SUB 0560: RETURN
540 BEEP .1,2: FOR q=166 TO 1 S
TEP -8: PLOT n,q: DRAW 0,-5: NEX
T q
550 PRINT "xmin=";: GO SUB 560:
RETURN
560 LET yp=v (n-1): LET yk=v (n+1
): LET xp=x (n-1): LET xk=x (n+1):
LET xz=xp+(xk-xp)*(ABS yp/(ABS
yk+ABS yp)): LET xz=.001*INT (10
00*xz+.0005): IF xz>0 THEN PRINT
"+";
565 IF xz>0 AND xz<.1 THEN PRIN
T "0";
567 IF xz=0 THEN PRINT " ";
570 PRINT xz: RETURN
580 FOR e=0 TO 7: BORDER e: BEE
P .1,e: NEXT e: BORDER 4: RETURN

590 PRINT AT 21,0: c8: AT 21,0: I
NK 7: u$="": v$="": w$="": x$="":
600 FOR n=0 TO 31: IF SCREENS (
21,31-n)<>" " THEN PRINT AT 21,0
: c8: LET tab=1: RETURN
610 NEXT n
620 LET t$="": LET s$="": LET j$="S
T
630 FOR n=0 TO 31: IF SCREENS (
0,31-n)<>" " THEN LET tab=n/2: P
RINT AT 0,0: c8: RETURN
640 NEXT n: RETURN
650 PRINT "u=": BEEP .1,-10
LET x=0: LET u0=.001*INT (1000
*VAL (f$)): IF u0>0 THEN PRINT "
+";
660 IF u0=0 THEN PRINT " ";
670 PRINT u0: FOR e=0 TO 255 ST
EP 4: PLOT e, CODE v$ (n): NEXT e:
RETURN
    
```

```

700 REM s t a r t
710 BORDER 4: DIM v (256): DIM x
(256): DIM v$ (256): DIM c$ (32):
LET f$="PI+x/5+3*SIN x": LET xmi
n=-10: LET xmax=2:
720 CLS: PRINT " PAPER 6; c8: "
Program rysuje wykresy funkcji
y=f(x)
dla dowolnych wartosci x z wy-
jatkiem funkcji typu y=1/x
gdzie y dazy do nieskonczono-
sci. Program wyznacza maxima,
minima i punkty zerowe dla x i
y. Skale pozioma i pionowa
wyznaczane sa przy pierwszym
przebiegu funkcji, drugi przebie-
g oblicza metoda interpola-
cji wartosci punktow charakte-
rystycznych funkcji i kresli
jej wykres.
730 c8: PAUSE 0: GO TO 150
99999 REM napis na taśmie
99999 CLEAR: SAVE CHR$ 10+CHR$ 1
+CHR$ 22+CHR$ 8+CHR$ 13+CHR$ 173
+CHR$ 8+CHR$ 60+CHR$ 76+CHR$ 65
LINE 700: BEEP .5,5: BEEP .1,17:
VERIFY " ": GO TO 110
    
```



Interfejs magnetofonu

W artykule opisano konstrukcję interfejsu magnetofonu przeznaczonego dla komputerów Atari 800 XL, 65 XE i 130 XE. Jest to układ wzorowany na rozwiązaniu zaczerpniętym z czasopisma „Funkschau” [1]. Dzięki wykorzystaniu cyfrowych układów scalonych interfejs charakteryzuje się dużą pewnością działania oraz umożliwia współpracę komputera z dowolnym magnetofonem kasetowym. Układ wymaga jednego napięcia +5V i może być zasilany z komputera ze względu na mały pobór prądu. W urządzeniu zastosowano w większości elementy produkcji krajowej poza jednym wyjątkiem — komparatorem napięcia LM 339.

Zasada działania

Zasada działania interfejsu oparta jest na pomiarze okresu sygnału wejściowego z magnetofonu i przyporządkowaniu wynikowi pomiaru odpowiedniego stanu logicznego. Jak pamiętamy, [2] częstotliwości 5,1 kHz odpowiada stan wysoki, a częstotliwości 3,9 kHz stan niski. Częstotliwościom tym odpowiadają okresy odpowiednio równe 196 i 25. Średni okres wynosi $(196 \div 25) \cdot 2$, co odpowiada częstotliwości 4,42 kHz. Układ porównuje okres sygnału wejściowego z okresem średnim. Jeżeli okres sygnału

wejściowego jest mniejszy od $226 \mu s$ ($f > 4,42 \text{ kHz}$), to przyporządkowuje mu się stan wysoki. W przypadku okresu równego lub większego od $226 \mu s$ ($f \leq 4,42 \text{ kHz}$) przyporządkowuje się jemu stan niski. Możemy w skrócie zapisać to tak:

- $f > 4,42 \text{ kHz}$ to stan wysoki.
- $f \leq 4,42 \text{ kHz}$ to stan niski.

Konkretne rozwiązanie układowe interfejsu pracującego wg opisanej zasady przedstawiono na rys. 1.

Opis układu

W konstrukcji możemy wyróżnić kilka bloków funkcjonalnych. Układ z tranzystorem T1 jest przeciwzmacniaczem sterującym pracą wzmacniacza, którego funkcję pełni jeden z komparatorów układu scalonego US4. Pozostałe trzy komparatory tego układu sterują właściwą pracą pozostałej „cyfrowej” części przystawki. Układ US1 jest generatorem częstotliwości wzorcowej 186 kHz ($T = 5,376 \mu s$). Wraz z licznikiem US2 pełnią one funkcje układu porównującego okresy: sygnału wejściowego ze średnim. Układem, który wynikiem porównania przy-

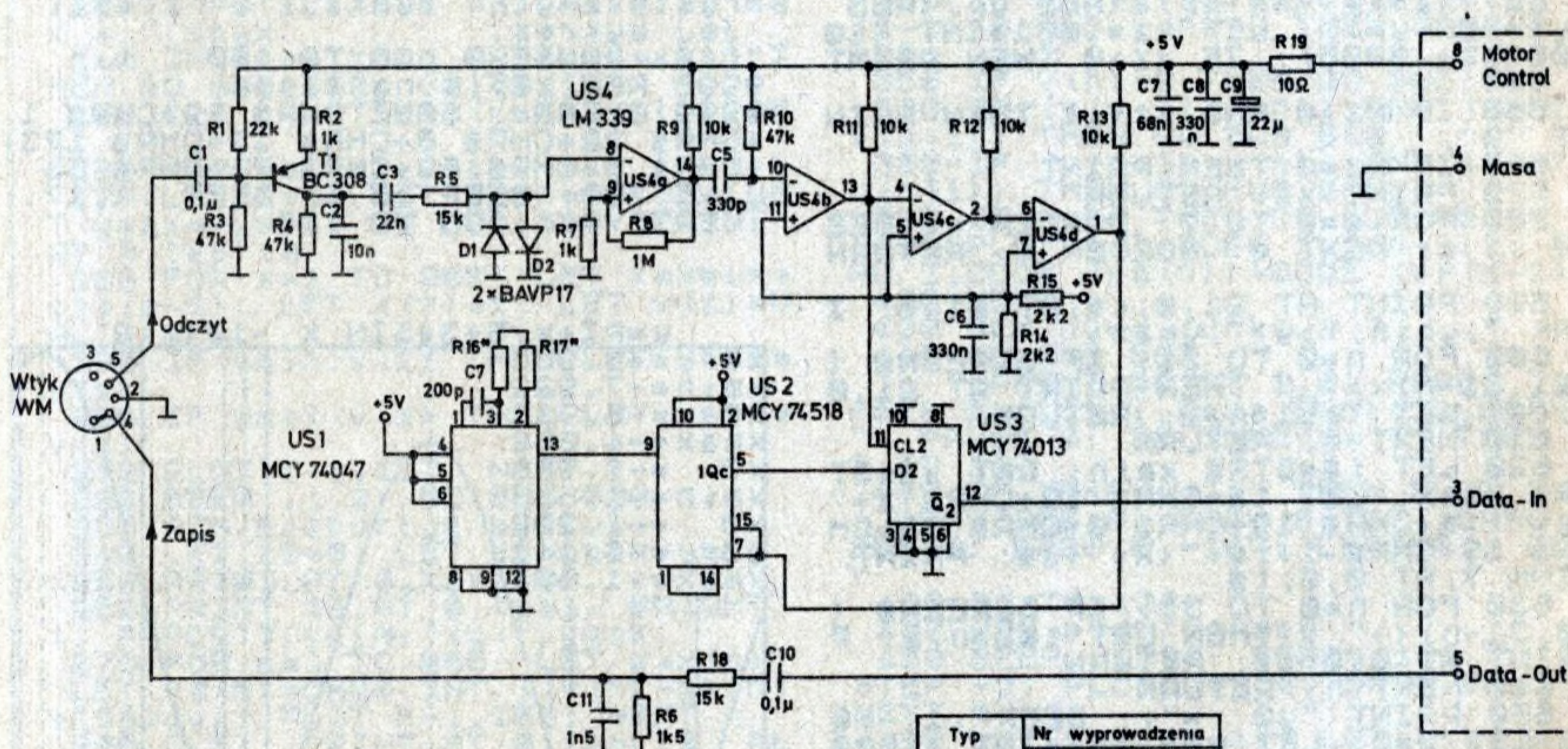
porządkowuje odpowiednie stany logiczne jest przerzutnik D US3. Bezpośrednio z jego wyjścia Q sterowane jest wejście Data-In złącza szeregowego Atari.

Przed dalszym opisem pracy interfejsu wprowadzimy pojęcie cyklu pracy. Przez pojęcie to rozumiemy bieżący pomiar jednego okresu sygnału wejściowego i przyporządkowanie wynikowi pomiaru, odpowiedniego stanu logicznego.

Cykl pracy

Sygnał z magnetofonu wzmacniany jest przez tranzystor T1 i układ US4a. Po wzmocnieniu jest różniczkowany (C5, R10) i formowany w układzie US4b.

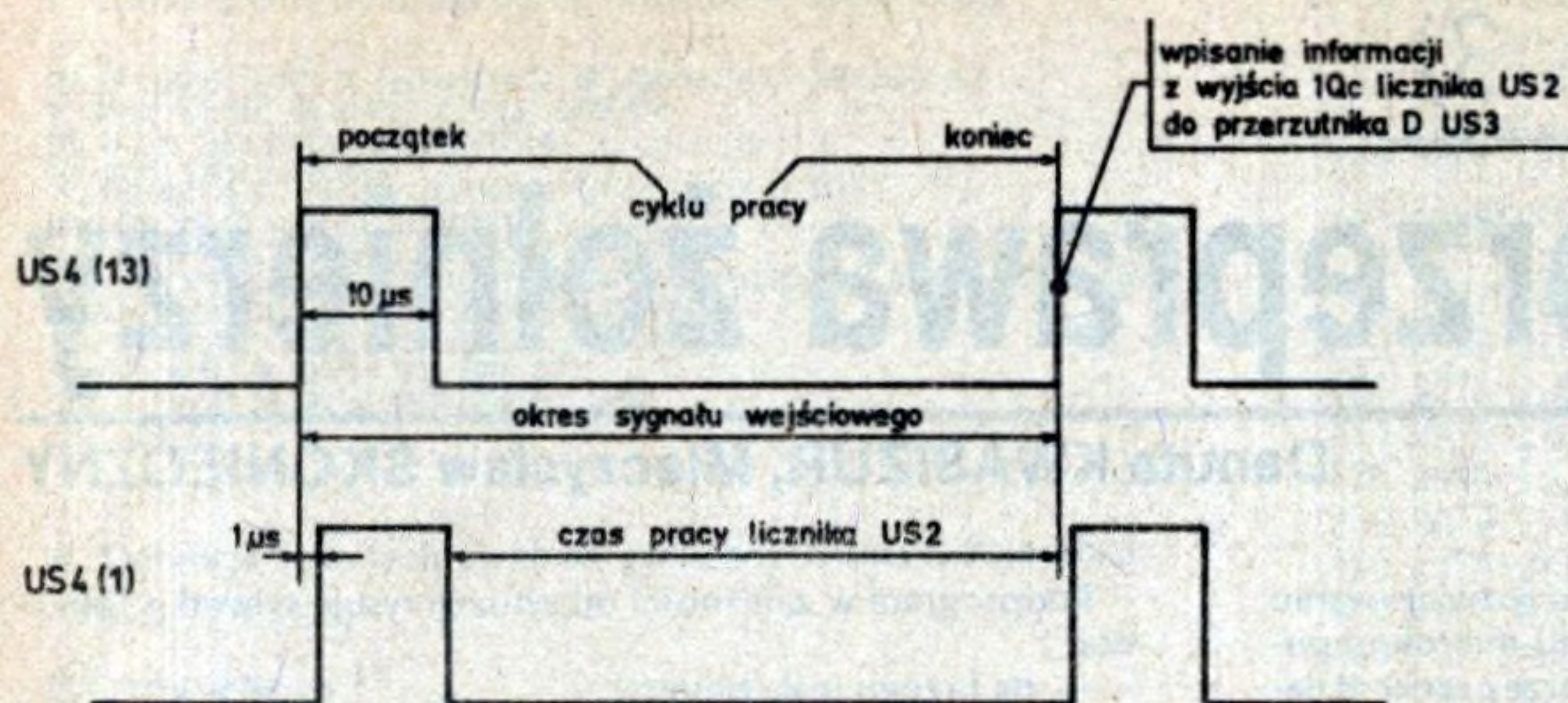
Komparator US4b wraz z elementami C5, R10 tworzy przerzutnik monostabilny, generujący dodatni impuls w odpowiedzi na opadające zbocze na wejściu (punkt C5, wypr. 14 US4). Szerokość impulsu, ustalona przez stałą czasową C5, R10, wynosi ok. $10 \mu s$. Zauważmy, że przerzutnik generuje te impulsy z częstotliwością sygnału wejściowego. Impuls steruje wejściem zegarowym przerzutnika D US3, a po przejściu przez komparatory US4, cd (z opóźnieniem ok. $1 \mu s$) licznik US2. Układ scalony US2 zawiera dwa niezależne liczniki BCD. W układzie połączono je w jeden licznik liczący do 40. Wyjście licznika 1Qc (wypr. 5 US2) ustawia się w stan wysoki, gdy licznik zliczy 40 i więcej impulsów generatora wzorcowego US1. W przeliczeniu na czas wyniesie to 40. Okres wzorcowy, czyli $40 \cdot 5,376 \mu s = 215 \mu s$. Po uwzględnieniu czasu trwania impulsu zegarowego oraz jego opóźnienia (rys. 2) łączny czas tego cyklu pracy wynosi $215 \mu s + 10 \mu s + 1 \mu s = 226 \mu s$. Jest to okres sygnałów o częstotliwości 4,42 kHz, który przyjęto jako wzorzec do porównań. W czasie normalnej pracy układu czas pracy licznika



Typ układu	Nr wyprowadzenia	
	masa	+5V
MCY74047	7	14
MCY74013	7	14
MCY74518	8	16
LM339	12	3

Złącze interfejsu szeregowego Atari

Rys. 1 Schemat ideowy interfejsu.



Rys. 2 Przebiegi czasowe cyklu.

jest różny i ograniczony okresem sygnału wejściowego. Zawartość licznika możemy obliczyć z zależności:

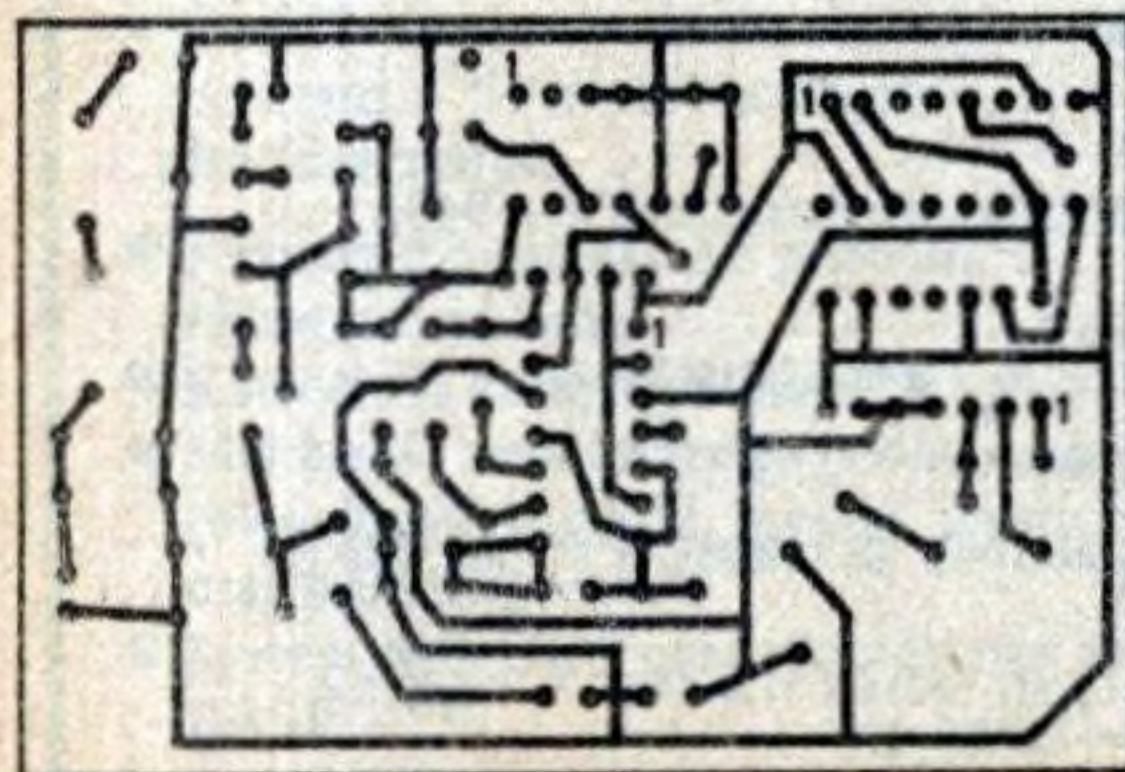
zawartość licznika = (okres sygn. wej. $11\mu\text{s}$) : $5,376\mu\text{s}$.

Dla sygnału o częstotliwości 3,9 kHz licznik zdąży policzyć do 45, a dla częstotliwości 5,1 kHz tylko do 34. W pierwszym przypadku wyjście 1Qc (US2) będzie w stanie wysokim, a w drugim w stanie niskim. Informacja z tego wyjścia wpisywana jest do przerzutnika D US3. Po jej wpisaniu w takim samym stanie będzie wyjście Q przerzutnika D. Natomiast na wyjście Q będzie przeciwny stan logiczny. Dla częstotliwości 3,9 kHz na wyjściu Q przerzutnika D będzie niski stan, a dla 5,1 kHz będzie stan wysoki. Informacja o każdym bicie przekazywana jest do komputera przez 1,666 ms (szybkość transmisji 600 bd). W związku z tym na informację o każdym bicie składa się kilka cykli pracy. Dla informacji o „1” logicznej stan wyjścia Q przerzutnika D nie będzie zmieniał się przez 8 cykli pracy, a dla informacji o „0” logicznym przez 6 cykli. Z wyjścia Q przerzutnika D informacja przekazywana jest bezpośrednio do komputera.

Podczas nagrywania programów sygnał z komputera (wyjście Data-Out), poprzez dzielnik napięcia R18/R6, podawany jest na wejście magnetofonu i zapisywany na kasecie magnetofonowej. Zadaniem dzielnika jest obniżenie amplitudy sygnału do poziomu wymaganego przez magnetofon.

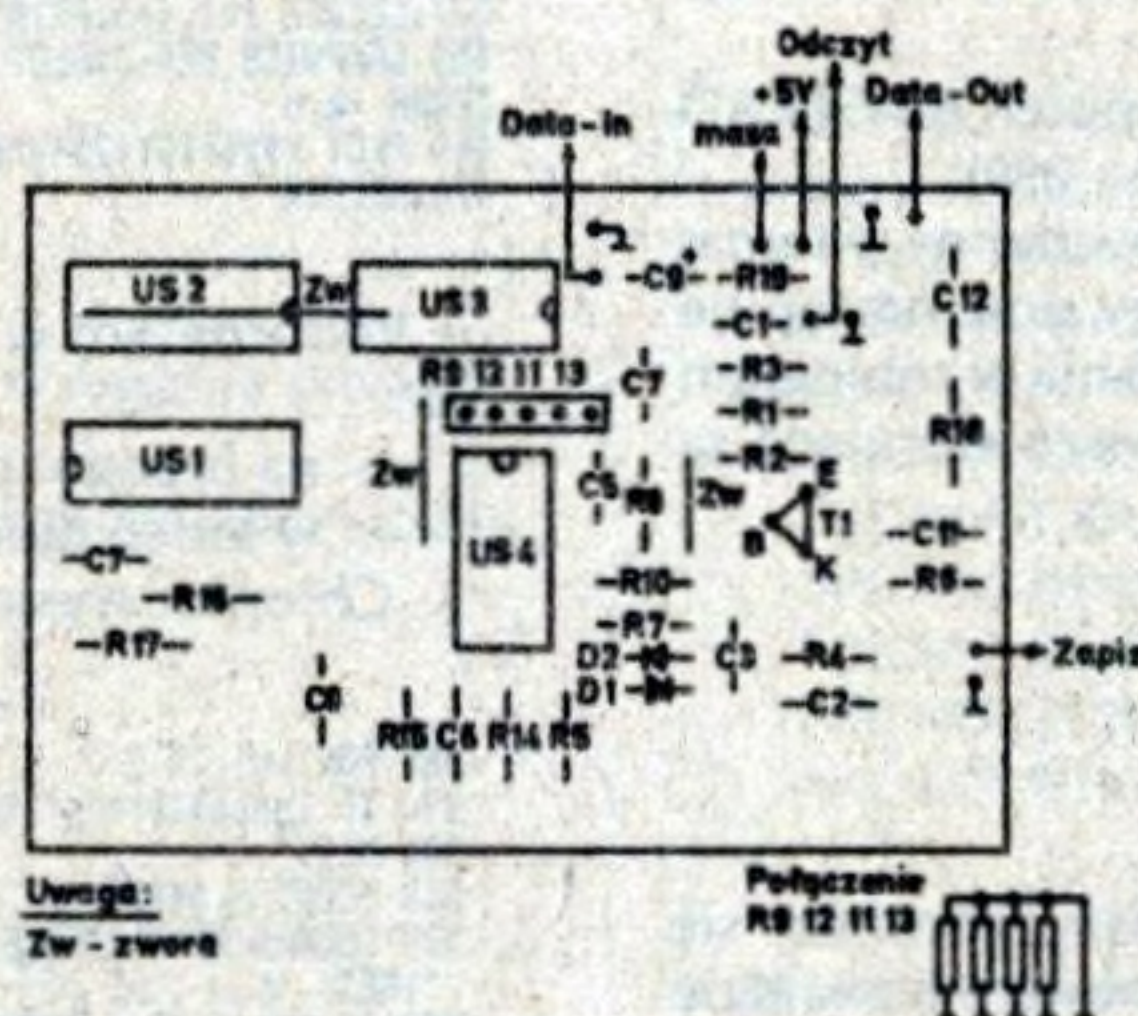
Konstrukcja i uruchomienie

Układ zmontowano na jednostronnie drukowanej płytce wykonanej wg rys. 3. Rozmieszczenie elementów zamieszczono na rys. 4. W pierwszej kolejności lutujemy elementy dyskretnie, a następnie układy scalone. Szczególną ostrożność należy zachować



Rys. 3

przy montażu cyfrowych układów scalonych, ponieważ są wykonane technologią CMOS (US1, US2, US3).

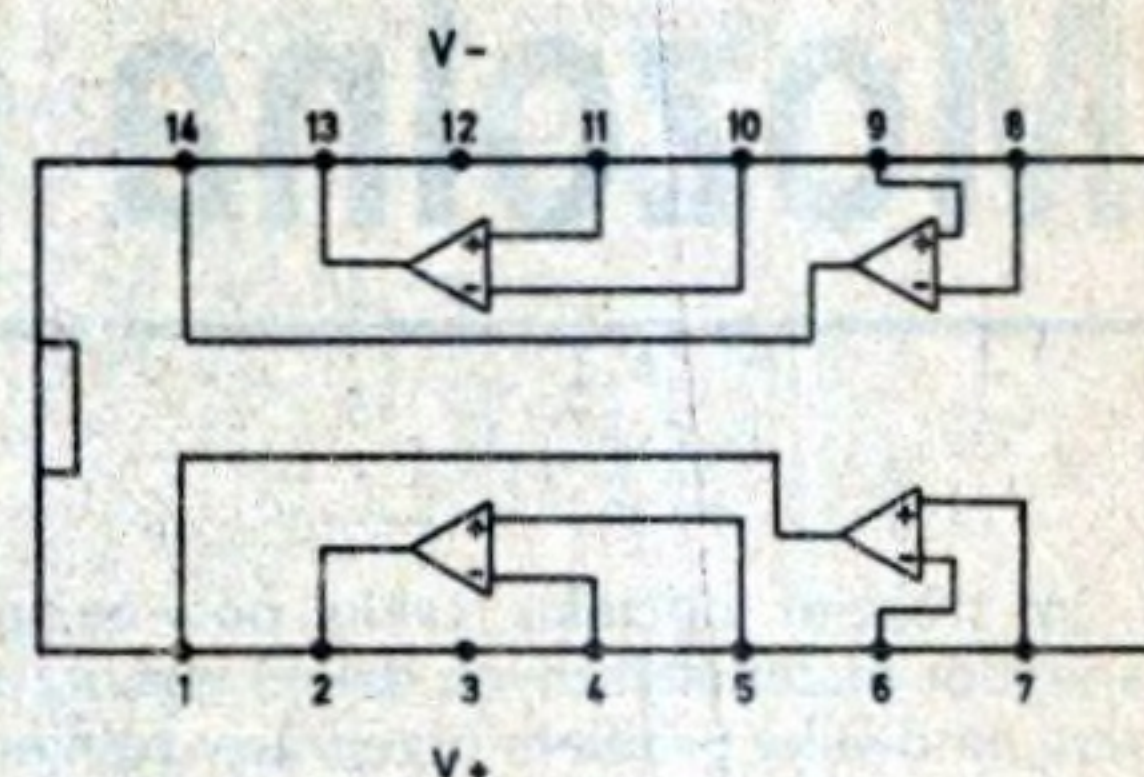


Rys. 4 Schemat montażowy

Po kontroli wzrokowej montażu przystępujemy do strojenia interfejsu, wykorzystując miernik częstotliwości lub oscyloskop. Interfejs zasilany napięciem +5V i do wyprowadzenia 13 układu US1 dołączamy miernik częstotliwości. Dobierając wartości rezystorów, R16 zgrubnie i R17 dokładnie staramy się uzyskać przebieg o częstotliwości 186 kHz. Po uzyskaniu tej częstotliwości strojenie interfejsu jest zakończone. W rozwiązaniu modelowym wartość rezystora R16 ustalono na 9,1 kΩ. Dobrane w ten sposób rezystory dolutowujemy do układu i jeszcze raz sprawdzamy częstotliwość generatora wzorcowego.

Przystawkę z magnetofonem łączymy dwużyłowym przewodem ekranowanym zakończonym typowym pięciostykowym wtykiem typu WM. Interfejs łączymy z komputerem trzyżyłowym przewodem ekranowanym zakończonym wtyczką wykonaną wg opisu zamieszczonego w „IKS-ie” [2]. Interfejs zasilany jest ze styku MOTOR CONTROL złącza Atari. Ma to znaczenie przy jednoczesnej współpracy komputera ze stacją dysków i magnetofonem. Na zakończenie podamy kilka informacji o układzie LM 339. Jest to poczwórny komparator monolityczny. Jedną z głównych jego zalet jest możliwość pracy z jednym napięciem zasilającym (już od +2V). Komparator zawiera na wyjściu układ z otwartym kolektorem, dzięki czemu istnieje możliwość doboru poziomów logicznych napięcia wyjściowego przez dołączenie

obwodów zewnętrznych. Rozmieszczenie wyprowadzeń układu przedstawiono na rys. 5.



Rys. 5 Rozmieszczenie wyprowadzeń układu scalonego LM339

Opisy pozostałych układów scalonych możemy znaleźć w [3] i [4].

Wykaz elementów

MCY 74013N — US3
MCY 74047N — US1
MCY 74518N — US2
LM 339 — US4

Diody i tranzystory

BC 308A — T1
BAVP17 — D1, D2

Kondensatory

200 pF — C17 (typu KSF)
330 pF — C5
1,5 nF — C11
10 nF — C2
22 nF — C3
68 nF — C7
100 nF — C1, C10
330 nF — C6, C8
22 F — 6,3V — C9

Rezystory 0,125 W

10 R — R19
1 k — R2, R7
1,5 k — R6
2,2 k — R14, R15
4,7 k — R4
10 k — R9, R11, R12, R13
15 k — R5, R18
22 k — R1
47 k — R3, R10
1 M — R8
R16, R17 — dobieramy w czasie strojenia — opis w tekście.

Inne

Wtyk magnetofonowy pięciostykowy typu WM

Bogusław LIPCZYŃSKI

LITERATURA

- H. Reelsen: „Recoederinterface”, „Funkschau” nr 17/1984
- Interfejs magnetofonu dla Atari, „IKS” 1987
- M. Kręciejewski: „Układy cyfrowe CMOS”, „Radioelektronik” nr 10/11/1985
- M. Łakomy, J. Zabrodzki: „Cyfrowe układy scalone” PWN 1980

Mozolna przeprawa żołnierzy

Danuta KWASIŹUR, Mieczysław SKONIECZNY

W trzecim odcinku cyklu poświęconego rozwiązywaniu anegdot matematycznych dzięki wykorzystaniu mikrokomputera omówimy problem przeprawy żołnierzy przez rzekę (Lilavati — anegdota nr 50). Treść historyjki jest następująca:

Oddział żołnierzy doszedł do rzeki, przez którą musi się przepłynąć. Most po niedawnej powodzi jest jeszcze w ruinie, rzeka zaś zbyt głęboka, by próbować przebrnąć ją w brud. W małej łódce przy brzegu bawią się dwaj chłopcy. Łódka jest tak mała, że zaledwie jeden żołnierz mógłby się w niej zmieścić. Mimo to, ta właśnie łódka przy czynnym udziale chłopców przewiozła na drugą stronę cały oddział żołnierzy. Jak to się stało?!

Segment sterujący programem na mikrokomputer AMSTRAD CPC 6128 ma następującą postać (instrukcje 10—290):

W tym fragmencie użytkownik zostaje wprowadzony w temat zagadnienia. Następnie pojawia się plansza gry oraz następuje zapoznanie grającego z możliwościami sterowania przebiegiem przeprawy. Możliwości i odpowiadające klawisze sterujące są następujące:

1. Chłopiec wysiada z łódki.
2. Chłopiec wsiada do łódki.
3. Żołnierz wysiada z łódki.
4. Żołnierz wsiada do łódki.
5. Łódka płynie.

Cała gra rozgrywana jest na tej samej planszy, a kończy się sukcesem w momencie, gdy cały oddział (9 żołnierzy) znajdzie się na drugim brzegu. Aby uzyskać większą przejrzystość wersji źródłowej programu, do segmentu zasadniczego dołączono następujące podprogramy:

1. Rysowanie żołnierza (instrukcje 300—400)

```

10 DIM zol(9,2):DIM chl(2,2):DIM lodka$(2):GOSUB 1070
20 MODE 1:INK 0,26:INK 1,0:INK 2,9:INK 3,14:PAPER 2:BORDER 9:CLS:PEN 1
30 LODKA$(1)="C":LODKA$(2)="C"
40 FOR k=1 TO 9:zol(k,1)=1:NEXT k:s=1
50 FOR k=1 TO 9:zol(k,2)=0:NEXT k:chl(1,2)=0:chl(1,1)=0:chl(2,1)=0
60 PRINT "9 żołnierzy doszło do rzeki, przez którą konieczne jest przepłynąć. Most po niedawnej powodzi jest jeszcze w ruinie, rzeka zaś zbyt głęboka, by próbować przebrnąć ją w brud. W małej łódce u brzegu bawią się dwaj chłopcy."
70 PRINT "Łódka jest tak mała, że zaledwie jeden żołnierz mógłby się w niej zmieścić. Mimo to, ta właśnie łódka przy czynnym udziale chłopców przewiozła na drugą stronę rzeki wszystkich żołnierzy."
80 PRINT "Jak to się stało?"
90 PEN 0:LOCATE 1,20:PRINT "NACISNIJ DOWOLNY KLAWISZ"
100 IF INKEY="" THEN GOTO 100
110 MODE 1:PAPER 0:BORDER 9:CLS:WINDOW 1,1,10,20,25:PAPER 1,2:WINDOW 2,31,40,20,25:PAPER 2,2:WINDOW 3,11,30,21,25:PAPER 3,3:CLS
120 WINDOW 4,1,40,1,1:PAPER 4,2:PEN 4,1:CLS 4
130 GRAPHICS PEN 1:FOR A=160 TO 440 STEP 56:PLOT A,16:DRAW 16,0:PLOT A+22,40:DRAW 16,0:PLOT A,64:DRAW 16,0:NEXT
140 FOR y=3 TO 15 STEP 6:FOR x=3 TO 11 STEP 3:GOSUB 300:GOSUB 370:NEXT:NEXT
150 PEN 1:LOCATE 11,20:PRINT CHR$(213);STRING$(5,CHR$(143));CHR$(212)
160 PAPER 0:y=18:FOR x=11 TO 14 STEP 3:GOSUB 410:NEXT
170 WINDOW 5,11,30,2,13:PAPER 5,2:CLS 5:PEN 0:PAPER 2
180 LOCATE 12,3:PRINT "MOZLIWOSCI:"
190 LOCATE 12,4:PRINT "1.Chłopiec wysiada":LOCATE 14,5:PRINT "z łódki"
200 LOCATE 12,6:PRINT "2.Chłopiec wsiada":LOCATE 14,7:PRINT "do łódki"
210 LOCATE 12,8:PRINT "3.Żołnierz wysiada":LOCATE 14,9:PRINT "z łódki"
220 LOCATE 12,10:PRINT "4.Żołnierz wsiada":LOCATE 14,11:PRINT "do łódki"
230 LOCATE 12,12:PRINT "5.Łódka płynie"
240 a$=INKEY$
250 IF a$="" THEN GOTO 240
260 P=VAL(A$)
270 IF P<1 OR P>5 THEN GOTO 240
280 DN P GOSUB 480,560,680,810,950
290 LOCATE 1,1:GOTO 240
300 PAPER 0:PEN 2:LOCATE X,Y:PRINT CHR$(190)
310 LOCATE X-1,Y+1:PRINT CHR$(191);CHR$(143);CHR$(192)
320 PLOT (X-1)*16+3,(Y-1)*16-10:DRAW 0,-4:DRAW 2,-2:DRAW 6,0:DRAW 2,2:DRAW 0,4
330 LOCATE X-1,Y+2:PRINT CHR$(193);CHR$(143);CHR$(194)
340 GRAPHICS PEN 1:PLOT (X-1)*16+3,(Y-1)*16-10:DRAW 0,-4:DRAW 2,-2:DRAW 6,0:DRAW 2,2:DRAW 0,4
350 PLOT (X-1)*16+1,(Y-1)*16-40:DRAW 14,0:DRAW 0,-2:DRAW -14,0
360 RETURN
370 LOCATE X-1,Y+3:PRINT CHR$(195);CHR$(196);CHR$(197)
380 LOCATE X-1,Y+4:PRINT CHR$(198);CHR$(199);CHR$(200)
390 PLOT (X-2)*16+1,(Y-1)*16-34:DRAW 0,-36:DRAW 2,0:DRAW 0,10:DRAW 0,-20:DRAW 2,0:DRAW 0,12:PLOT 2,-6:DRAW 0,-6
400 RETURN

```

Podprogram w zależności od sytuacji rysuje sylwetkę żołnierza:

- na brzegu (cała postać);
 - w łódce (górna część sylwetki).
2. Rysowanie chłopca (instrukcje 410—470).

```

410 PEN 3:LOCATE X,Y:PRINT CHR$(201);CHR$(210);CHR$(202)
420 GRAPHICS PEN 1:PLOT X*16+4,(25-Y)*16+6:DRAW 6,0:DRAW 2,2:DRAW 0,6:DRAW -2,2:DRAW -6,0:DRAW -2,-2:DRAW 0,-6:PLOT X*16+4,(25-Y)*16+4:PLOT X*16+10,(25-Y)*16+4:PLOT X*16+4,(25-Y)*16+14:DRAW 6,0:MOVE X*16+8,(25-Y)*16+10:FILL 0
430 LOCATE X,Y+1:PRINT CHR$(203);CHR$(204);CHR$(205):RETURN
440 LOCATE X,Y+2:PRINT CHR$(206);CHR$(208):PEN 1:LOCATE X+1,Y+2:PRINT CHR$(207):LOCATE X+1,Y+3:PRINT CHR$(209)
450 GRAPHICS PEN 3:PLOT X*16+4,(25-Y)*16-46:DRAW -6,0:PLOT X*16+4,(25-Y)*16-48:DRAW -6,0
460 PLOT X*16+10,(25-Y)*16-46:DRAW 6,0:PLOT X*16+10,(25-Y)*16-48:DRAW 6,0
470 RETURN

```

Podprogram w zależności od sytuacji rysuje sylwetkę chłopca:

- na brzegu (cała postać);
 - w łódce (górna część sylwetki).
3. Chłopiec wysiada z łódki (instrukcje 480—550).

```

480 FOR k=1 TO 2
490 IF LODKA$(k)="C" OR LODKA$(k)="C" THEN GOTO 520
500 NEXT
510 SOUND 1,458:PRINT 4,"NIE MA CHLOPCA W LODZI":GOTO 670
520 LODKA$(K)=" ":PAPER 0:FOR A=1 TO 3:LOCATE 11+(K-1)*3+13*(S-1),16+A:PRINT " ":NEXT
530 IF CHL(1,S)=1 THEN X=2+30*(S-1):Y=21:PAPER 2:GOSUB 410:GOSUB 440:CHL(2,S)=1:GOTO 550
540 CHL(1,S)=1:PAPER 2:X=6+30*(S-1):Y=21:GOSUB 410:GOSUB 440
550 RETURN

```

4. Chłopiec wsiada do łódki (instrukcje 560—670).

```

560 FOR k=1 TO 2
570 IF chl(k,s)=1 THEN GOTO 600
580 NEXT k
590 PRINT 4,"NIE MA CHLOPCA NA TYM BRZEGU":GOTO 670
600 IF NOT(LODKA$(1)="C") THEN GOTO 620
610 PRINT 4,"NIE MA NIEJSCA W LODCE":GOTO 670
620 FOR A=1 TO 2:IF LODKA$(A)="C" THEN GOTO 640
630 NEXT A
640 CHL(K,S)=0:LODKA$(A)="C":X=11+(A-1)*3+13*(S-1):Y=18:PAPER 0:GOSUB 410
650 PAPER 2:FOR t=1 TO 5:LOCATE 6-(K-1)*4+30*(S-1),19+t:PRINT " ":NEXT t
660 RETURN
670 SOUND 1,458:SOUND 1,348:FOR T=1 TO 3000:NEXT:CLS 4:RETURN

```

5. Chłopiec wysiada z łódki (instrukcje 680—800).

```

680 IF LODKA$(1)="Z" THEN GOTO 700
690 PRINT 4,"NIE MA ZOLNIERZA W LODCE":GOTO 670
700 LODKA$(1)=" "
710 FOR K=1 TO 9
720 IF ZOL(K,S)=0 THEN GOTO 740
730 NEXT K
740 ZOL(K,S)=1
750 FOR T=1 TO 3:LOCATE 12+(S-1)*13,16+T:PRINT " ":NEXT T
760 X=9+INT((K-1)/3)*9-(K-1)*3+30*(S-1):Y=15-INT((K-1)/3)*6:GOSUB 300:GOSUB 370
770 IF S=1 THEN RETURN
780 FOR K=1 TO 9
790 IF ZOL(K,S)=0 THEN RETURN
800 NEXT K:GOTO 1020

```

6. Żołnierz wsiada do łódki (instrukcje 810—940).

```

810 IF ZOL(1,S)=1 THEN GOTO 830
820 PRINT 4,"NIE MA ZOLNIERZY NA TYM BRZEGU":GOTO 670
830 FOR A=9 TO 1 STEP -1
840 IF ZOL(A,S)=1 THEN GOTO 860
850 NEXT A
860 IF LODKA$(1)="C" OR LODKA$(1)="Z" OR LODKA$(2)="C" THEN GOTO 610
870 ZOL(A,S)=0:LODKA$(1)="Z"
880 IF S=2 THEN GOTO 990
890 PAPER 0:FOR K=15 TO 19:LOCATE 8+30*(S-1),K:PRINT " ":NEXT K
900 PAPER 0:X=13+13*(S-1):Y=17:GOSUB 300
910 IF A=1 THEN RETURN
920 X=8+INT((A-1)/3)*9-(A-1)*3+30*(S-1):Y=15-INT((A-1)/3)*6
930 FOR T=0 TO 4:LOCATE X,Y+T:PRINT " ":NEXT T
940 X=9:Y=15:GOSUB 300:GOSUB 370:RETURN

```

7. Łódka płynie na drugi brzeg (instrukcje 950—1060).

```

950 IF lodka$(1)="C" OR lodka$(1)="2" OR lodka$(2)="C" THEN GOTO 970
960 PRINT £4,"NIE MA NIKOGO W LODCE":GOTO 670
970 IF S=1 THEN GOSUB 1250:S=2:RETURN
980 GOSUB 1380:S=1:RETURN
990 X=8+INT((A-1)/3)*9-(A-1)*3+30*(S-1):Y=15-INT((A-1)/3)*6
1000 FOR T=0 TO 4:LOCATE X,Y:T:PRINT " ":NEXT T
1010 PAPER 0:X=13+13*(S-1):Y=17:GOSUB 300:RETURN
1020 CLSE5:PEN 0:PAPER 2:LOCATE 12,3:PRINT"BRAMQ!":LOCATE 12,5:PRINT"KONIEC PRZE
PRAWY":FOR A=1 TO 5:SOUND 1,358:SOUND 1,458:NEXT
1030 PEN 1:LOCATE 12,8:INPUT"JESZCZE RAZ(T/N)";K$
1040 IF K$="T" OR K$="t" THEN GOTO 20
1050 IF K$="N" OR K$="n" THEN LOCATE 1,1:END
1060 GOTO 120
    
```

8. Definicja własnych symboli graficznych (instrukcje 1070—1240).

```

1070 SYMBOL AFTER 190
1080 SYMBOL 190,60,126,126,255
1090 SYMBOL 191,0,3,7,15,15,14,14,14
1100 SYMBOL 192,0,192,224,240,240,112,112,112
1110 SYMBOL 193,14,14,14,30,62,60,120,112
1120 SYMBOL 194,112,112,112,112,112,112,112,112
1130 SYMBOL 195,112,96:SYMBOL 196,255,255,231,231,231,231,231
1140 SYMBOL 197,112,112,112:SYMBOL 198,0,0,0,0,0,1,7,7
1150 SYMBOL 199,231,231,231,231,231,231,231:SYMBOL 200,0,0,0,0,0,128,224,224
1160 SYMBOL 201,0,0,0,0,0,0,0,1:SYMBOL 202,0,0,0,0,0,0,0,128
1170 SYMBOL 203,3,3,3,3,3,3,3,3
1180 SYMBOL 204,255,255,126,126,126,126,126,126
1190 SYMBOL 205,192,192,192,192,192,192,192,192
1200 SYMBOL 206,3,3,3
1210 SYMBOL 207,126,126,126,126,102,102,102,102
1220 SYMBOL 208,192,192,192
1230 SYMBOL 209,102,102,102,102,102,102,102
1240 RETURN
    
```

Symbole 190—209 są wykorzystywane do rysowania sylwetek żołnierzy i chłopców.

9. Przesuwanie łódki w prawo i w lewo (instrukcje 1250—1530).

```

1250 REM ' przesuwanie w prawo'
1260 RESTORE 1280:FOR i=&7F76 TO &7FA1
1270 READ v:POKE i,v:NEXT
1280 DATA 1,8,4,33,39,192,84
1290 DATA 93,19,197,1,14,0,237,184,193,35,54,0
1300 DATA 13,40,6,17,13,8,25,24,234,5,40,12,124
    
```

```

1310 DATA 214,56,103,17,93,0,25,14,8,24,219,201
1320 x=17:y=17:GOSUB 1490
1330 POKE &7F7B,w1
1340 FOR z=w2+1 TO w2+26
1350 POKE &7F7A,z:CALL &7F76
1360 FRAME:FOR t=1 TO 60:NEXT:NEXT:RETURN
1370 REM 'przesuwanie w lewo'
1380 RESTORE 1400:FOR i=&7F76 TO &7FA1
1390 READ v:POKE i,v:NEXT
1400 DATA 1,8,4,33,36,192,84,93
1410 DATA 27,197,1,14,0,237,176,193,43,54,0
1420 DATA 13,40,6,17,243,7,25,24,234,5,40,12,124
1430 DATA 214,56,103,17,67,0,25,14,8,24,219,201
1440 x=24:y=17:GOSUB 1490
1450 POKE &7F7B,w1
1460 FOR z=w2 TO w2-26+1 STEP -1
1470 POKE &7F7A,z:CALL &7F76
1480 FRAME:FOR t=1 TO 60:NEXT:NEXT:RETURN
1490 A=(Y-1)*80+(x-1)*2
1500 W1=INT(a/256)
1510 W2=a-W1*256
1520 w1=192+w1
1530 RETURN
    
```

W trakcie sterowania przeprowadzamy (klawisze numeryczne 1—5) program przeprowadza kontrolę logiczną zaistniałej sytuacji. W razie błędów logicznych są wyświetlane odpowiednie komunikaty i program czeka na poprawną dyrektywę. Całkowity zestaw komunikatów o błędach jest następujący:

- NIE MA CHŁOPCA W ŁODZI.
- NIE MA CHŁOPCA NA TYM BRZEGU.
- NIE MA MIEJSCA W ŁÓDCE.
- NIE MA ŻOŁNIERZA W ŁÓDCE.
- NIE MA ŻOŁNIERZA NA TYM BRZEGU.
- NIE MA NIKOGO W ŁÓDCE.

Życzymy wytrwałym Czytelnikom sukcesów w sterowaniu przeprowadzaniu żołnierzy przy użyciu mikrokomputera.

X'press 16

Komputer domowy na każdą okazję

- PC kompatybilny
- 256 kB o dostępie bezpośrednim
- wbudowana stacja dysków 5,25" z 360 kB
- PC standard (CGA) i wzbogacona kolorowa grafika
- 256x212 punktów rozdzielczości z 256 kolorami i 32 wielobarwnymi płaszczyznami
- 512x212 punktów rozdzielczości z 16 kolorami i 32 wielobarwnymi płaszczyznami
- 3 kanały dźwiękowe z 8 pełnymi oktavami na kanał
- Klawiatura kompatybilna z PC
- Możliwość zewnętrznego podłączenia: równoległej drukarki, myszki, joystick'a oraz świetlnego póra
- System MS-DOS oraz wzbogacony GW BASIC



Płyta kompaktowa pojawiła się w 1982 r. w Japonii. W marcu 1983 była już sprzedawana w Europie Zachodniej. Od początku sprzedaż płyt, zwanych najczęściej CD (od ang. Compact Disc) a także odtwarzaczy laserowych osiągnęła ogromne rozmiary. Do dziś nie ma zgody, jak po polsku te urządzenia nazwać — jedni lansują nazwę laserofon, inni — dyskofon.

Niezwykle powodzenie płyta laserowa zawdzięcza przede wszystkim doskonałej jakości odtwarzanego dźwięku, wygodzie korzystania z niej i jednakowemu w całym świecie standardowi.

Doskonałą jakość osiąga się dzięki nowej technologii nagrywania i odtwarzania dźwięku. Najistotniejszym elementem tej technologii jest laser, który pełni tu rolę igły gramofonowej. Drugim, nie mniej ważnym, jest mikrokomputer. Jest on czymś w rodzaju fabryki, w której cyfrowy „surowiec” jest wszechstronnie obrabiany.

W sygnale zakodowane są nie tylko dźwięki, lecz także informacje kontrolno-sterujące, bez których ta fabryka nie byłaby w stanie funkcjonować. Dzięki nim komputer wykrywa i natychmiast poprawia błędy wynikające np. z zadrapania płyty. Nawet plamka o średnicy ok. 2 mm nie jest odczuwalna podczas odtwarzania płyty.

Współczesne gramofony CD potrafią samoczynnie korygować nawet 60 najczęściej spotykanych błędów i przekłamań odczytu. Jest więc on skomplikowanym urządzeniem laserowo-komputerowym. Natomiast płyta CD odtwarzana jest przeciwnie, niż longplay tradycyjny — od środka.

Ponieważ strumień odtwarzanych informacji powinien być stały, płyta na początku

wiruje 500, a pod koniec odtwarzania 215 razy na minutę. W ciągu każdej sekundy mikroprocesor przetwarza 4 321 800 bitów informacji zawartych w sygnale cyfrowym.

Śpiewający laser, dziewczyny i złoto

Z czym się nam najczęściej kojarzy obecnie postęp? Z nowoczesnością przede wszystkim. A dziś nowoczesność, to w głównej mierze — elektronika. Ona wyznacza swoisty próg rozwojowy, dzieli na „wielki świat” i okolice, wyznacza miejsce w czo-

lanych na całym świecie takich jak: ITT Canon, Thomson, Brandt, Nordmende, Continental Edison Sabaj — nie wzbudzają tutaj żadnych emocji.

Radioodbiorniki oraz radiomagnetofony produkowane w „Eltrze” kupują klienci na całym świecie. I chwalą ich jakość oraz estetykę wykonania, parametry. Co warte podkreślenia, ten eksport nie jest wcale mały. Dlaczegoż więc mieliby mieć jakieś kompleksy?

— W bieżącym roku — mówi główny specjalista zakładów, inż. Piotr Terlecki wyprodukujemy 900 tysięcy różnorodnego sprzętu radiowo-elektronicznego. Z czego

Kostka w tekście

Compact Disc

Włodzimierz KALETA

łówce państw najbardziej technologicznie rozwiniętych lub spycha je do cywilizacyjnego ogona.

Ten „wielki świat” zawiązał przed kilku laty do... Bydgoszczy. Stało się to głównie za sprawą Zakładów Radiowych „Unitra — Eltra”. Japonia, Francja, RFN. Nazwy firm

40—50 procent radiomagnetofonów i ponad połowę odbiorników przenośnych wyślemy na eksport. Głównie do krajów Europy Zachodniej — Francji, Republiki Federalnej Niemiec, Szwecji, Jugosławii oraz Japonii. Produkcja eksportowa obejmuje około 24 proc. naszej globalnej produkcji.

Co my z tego mamy?

Pięknie prezentują się wyroby „Eltry”. Większość radioodbiorników nosi żeńskie imiona: „Ania”, „Zosia”, „Klaudia”, „Jowita”, „Hania”, „Liza”... Lubią tutaj dziewczyny?

— To może dlatego — tłumaczy kierownik sekcji technologii, inż. Eugeniusz Bazak — że w naszym zakładzie pracuje wiele kobiet. Produkcja tu wytwarzana wymaga ogromnej cierpliwości i dokładności. Wiadomo, że kobiety do tego rodzaju pracy nadają się najlepiej. Są cierpliwe z natury. No, może z wyjątkiem kilku naszych żon.

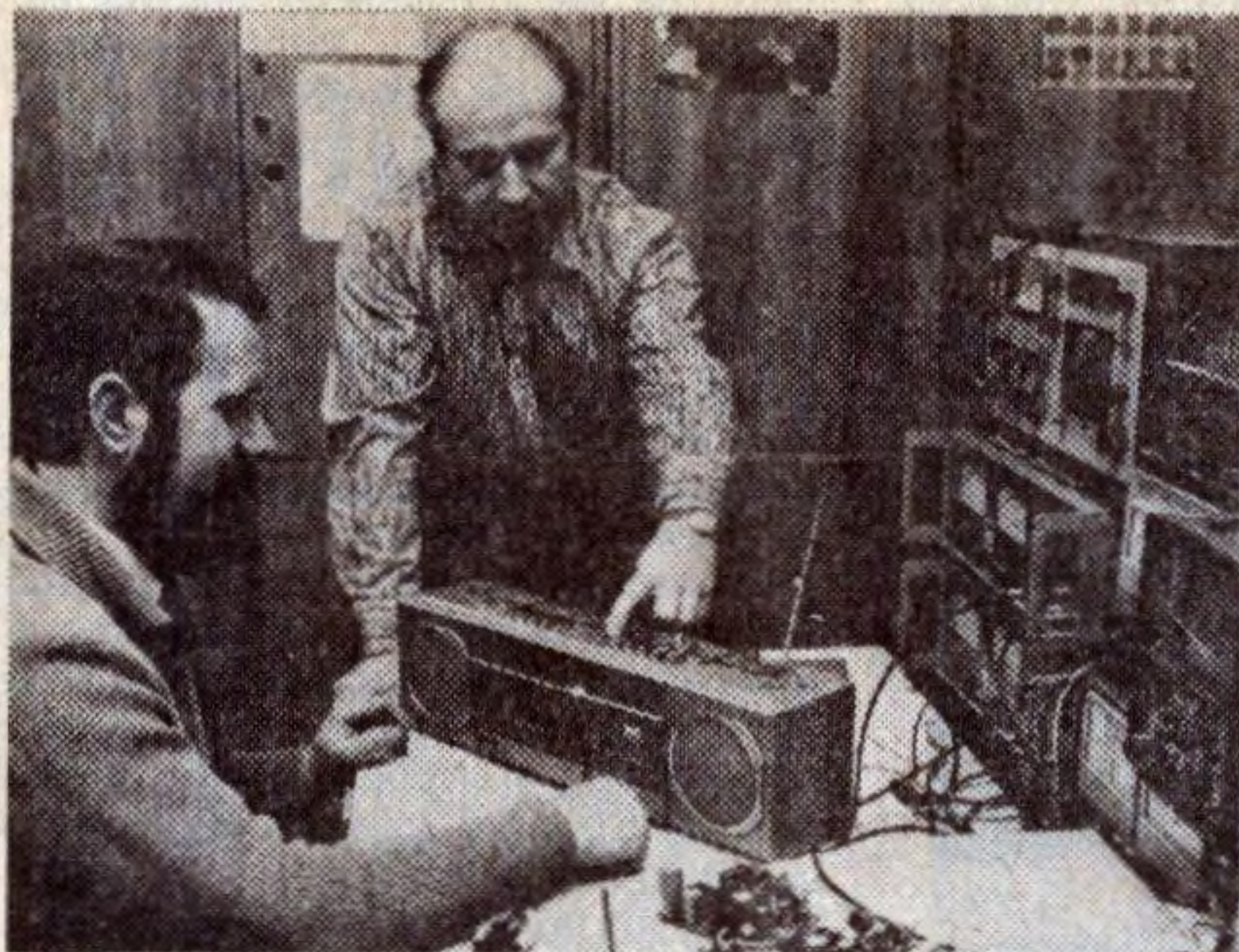
Patrząc na wytwarzane tutaj cacuszka, aż łza się kręci w oku, że radioodbiorniki o wdzięcznych dziewczęcych imionach opuszczają kraj i cieszyć będą wzrok i serca innych. Cóż z tego eksportu mamy my, polscy melomani i zwykli zjadacze chleba, którzy lubią posłuchać miłej piosenki przy goleniu?

— Nasza oferta na rynek krajowy jest bardzo bogata — wyjaśnia kierownik montażu odbiorników, inż. Andrzej Kazubowski. — Trafia nań przecież 46 proc. naszej produkcji. Nie rozróżniamy wśród niej tej na eksport i na rynek krajowy. Radioodbiorniki i radiomagnetofony do nas i za granicę wytwarzają ci sami ludzie, przestrzegając tych samych — zawsze wysokich — norm jakości. Gdybyśmy postępowali inaczej, demoralizowalibyśmy ludzi.

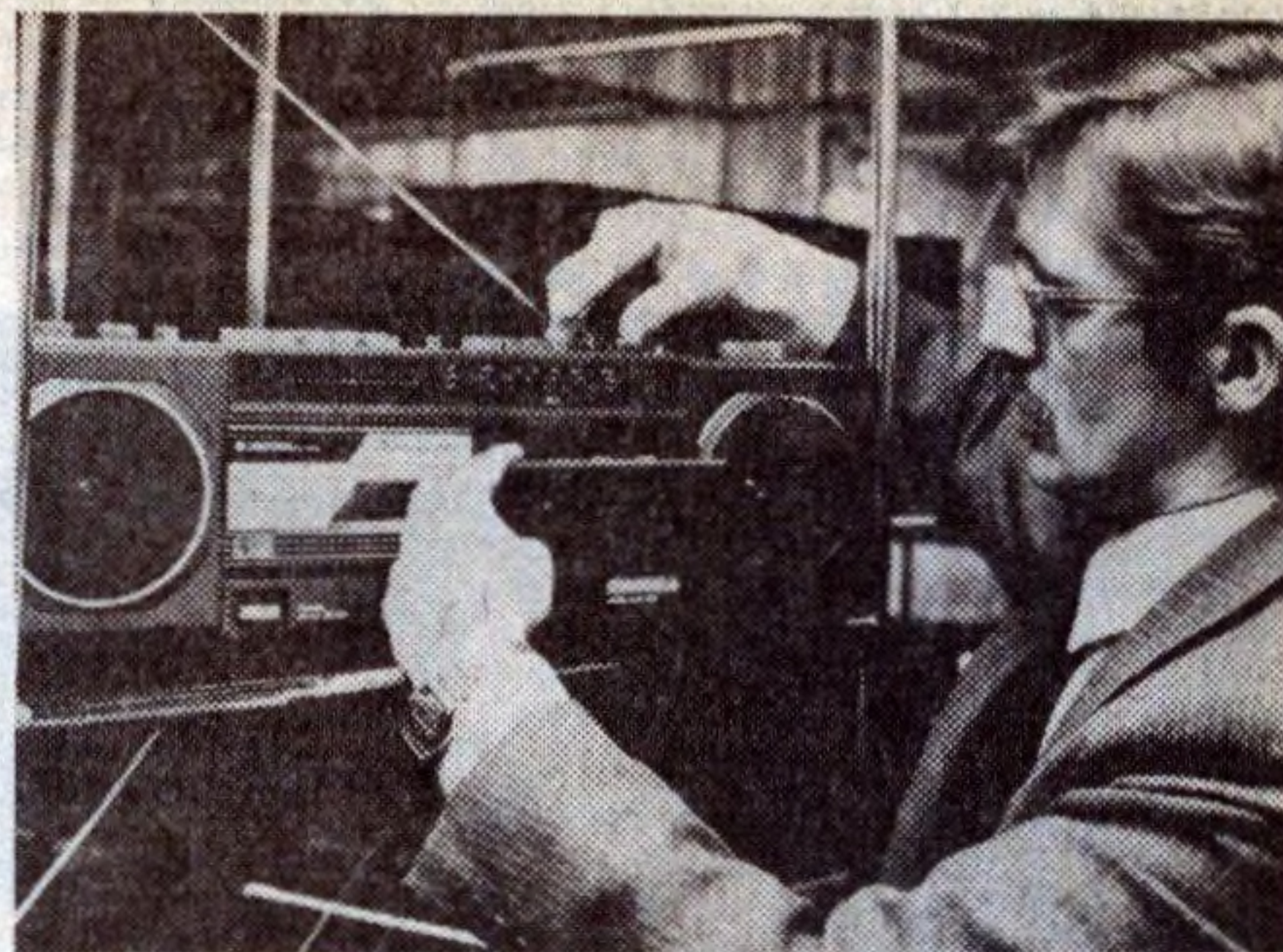
Przykład pierwszy z brzegu: „Ania”, „Liza” i „Emilia”... Te same odbiorniki — pod względem koloru, jakości wykonania — można kupić w sklepach Francji, RFN i naszych. Wyjątek stanowią radioodbiorniki, sprzedawane szwedzkiej firmie „Luxor”. Nie ma ich na naszym rynku. Ale tylko dlatego,



Z elektroniką mają w zakładzie do czynienia na co dzień. Na zdjęciu: monter układów elektrycznych Jarosław Sierant przeprowadza korektę zakresów radioodbiornika „Ania”.
(Foto: Marek Chelminiak)



Jan Meler (z prawej) jest jednym z głównych konstruktorów najnowszej generacji radiomagnetofonów produkowanych przez „ELTRĘ”



„Manuela” prezentowana przez inż. Piotra Terleckiego, to najnowsze dziecko zakładów. Ambicją firmy jest, by wszystkie modele eksportowe znalazły się na krajowym rynku nie później niż po roku.

iz nie ma na nie zbytu. To radio tylko z zakresem fal UKF i możliwością ich programowania, za którym przepadają Szwedzi, nam nie przypadło do gustu.

W dyrekcji zapewniają, że najnowsze dziecko zakładu, „Manuela”, również trafi do naszych sklepów i to jeszcze w drugim kwartale. Jest to radiomagnetofon dwukasetowy, stereofoniczny, o mocy 2x4 W, z dwiema prędkościami przesuwu taśmy, czterozakresowy, możliwością współpracy z każdym innym sprzętem.

Radiomagnetofon „palce lizać”. Mogliby na eksport wystać każdą ilość. Ale przecież nie o to chodzi. W bieżącym roku sprzedaż na rynek krajowy wyniesie 117 proc. ubiegłorocznej produkcji i wzrośnie o około 1 mld zł.

— Eksportujemy przede wszystkim dlatego — zapewnia inż. Terlecki — aby móc produkować dla kraju sprzęt najnowszej generacji. Bez tego eksportu nie mielibyśmy dewiz niezbędnych na unowocześnianie naszej produkcji, nie znalazłybyśmy nowinek technicznych w tym zakresie.

Okazjonalne słowa? W „Eltrze” mają wiele przekonujących argumentów, że nie. Tu potrafią nie tylko korzystać z obcych licencji, ale także udoskonalać je. I to tak, że już obecnie firmy zachodnie mają w nich groźnego konkurenta.

Przepchać wąskie gardło

— Co jest piętą achillesową krajowej elektroniki? — Inż. Terlecki wie, że to pytanie retoryczne, dlatego nie czekając na odpowiedź stwierdza: — Brak podzespołów. Aby je wytworzyć potrzebne są odpowiednie narzędzia, precyzja ich wykonania. Mozolne linie do ich wytwarzania, wymagają żmudnych badań oraz wyjątkowego wręcz przestrzegania wymogów jakościowych.

Do niedawna nie było problemów — kupowało się potrzebne podzespoły na Zachodzie. A dziś?

W latach osiemdziesiątych zakupiono w „Eltrze” holenderską licencję linii do selektywnego pokrywania styków. W latach 1982 i 1983, kiedy rozpoczynali pracę „pełną parą” kontakt z krajami zachodnimi był — nie z naszej zresztą winy — ograniczony. Dlatego w zakładzie wzięto się poważnie za bary z

najnowszymi technologiami dotyczącymi wytwarzania podzespołów stykowych.

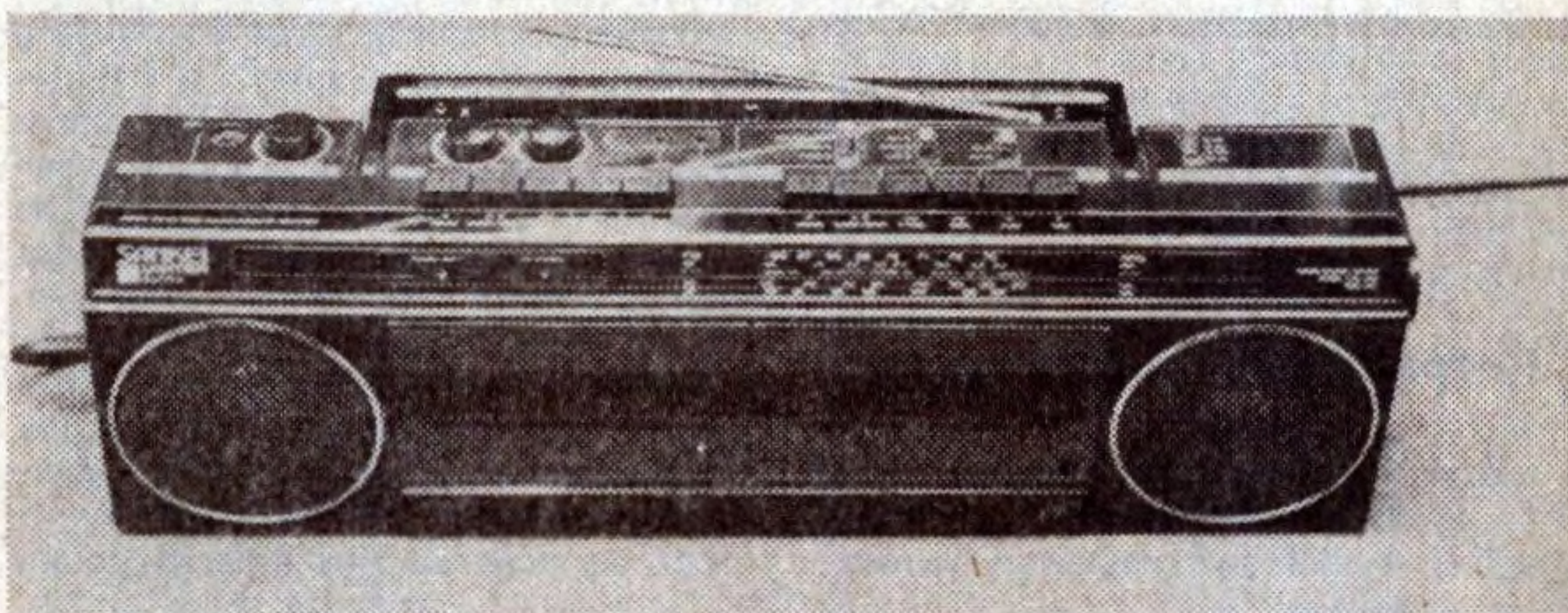
Efekty tej pracy nie dały długo na siebie czekać. Wkrótce mieli swoją własną linię. Równie dobrą, a może nawet lepszą. W każdym razie na pewno oszczędniejszą. Styki pokrywało się bowiem dotychczas... złotem. Rocznie zużywano go około 300 kg. Metoda selektywnej galwanizacji styków, którą opracowali, pozwala zaoszczędzić w roku blisko 100 kg tego cennego kruszcu. Bez obniżenia jakości elementów.

Już obecnie część tych podzespołów wysyła się na eksport. I nikt z kupujących — a konkurencja duża i wymagania jakościowe nie mniejsze — nie narzeka. Przeciwnie, chcieliby kupować więcej.

komputerowy, górnictwo, przemysł maszynowy. Wszystkich, którzy przy produkcji potrzebują łączyć obwody drukowane, scalone, panele.

Stale rozwija się także zaplecze naukowo-badawcze. Dziś w służbie głównego konstruktora, w sześciu działach, pracuje kilkaset ludzi. Właściwie byłoby powiedzieć — myśli. I efekty tego myślenia przynoszą zakładowi coraz większe zyski i... oszczędności. Kierownik działu konstrukcyjnego, inż. Janusz Pazderski, nie chce rozmawiać o tym, nad czym obecnie pracują. Co prawda konkurencja pomiędzy naszymi zakładami nie taka, jak w innych krajach na zachód od Łaby, ale... ostrożność nie zawadzi.

Ogólnie można powiedzieć tak: myślą w



Radiomagnetofon produkowany wspólnie z Japończykami SANKEI. Najnowocześniejszy z odtwarzaczem laserowym.

— Naszym głównym zadaniem, jakie stawiamy sobie na najbliższe lata — mówi inż. Terlecki — jest wzrost produkcji podzespołów. Przy tym samym np. zużyciu złota i nie zwiększając zatrudnienia chcemy, w ciągu najbliższych pięciu lat, zwiększyć ich produkcję do 6 mln sztuk, czyli dwukrotnie.

Obecnie produkcja podzespołów stanowi 35 proc. produkcji globalnej. Warto wiedzieć, że jesteśmy największym w Polsce producentem podzespołów dla wszystkich przedsiębiorstw, które mają coś wspólnego z elektroniką. Zaopatrujemy w nie przemysł

zakładzie o stykach przyszłości. Jeszcze lepszych, nowocześniejszych, bardziej zminiaturyzowanych. Ale nie tylko. O nowych modelach odbiorników z zastosowaniem ostatnich osiągnięć w zakresie przetworników i kondensatorów. Opracowano już wiele nowych rozwiązań. M.in. przetworniki suwakowo-obrotowe, uchylne, włączniki przyciskowe i impulsowe. Zamiast kondensatorów powietrznych stosować się będzie obrotowe.

Słowem, dalsza faza elektronicznej rozbudowy i wprężenia jej do sterowania układami cyfrowymi oraz do innych techno-

logii specjalnych. Chcą bowiem opracować własne linie produkcyjne włączników, myślą o możliwości produkowania własnych mechanizmów napędowych, gdyż te, które dostarczał kooperujący z „Eltrą” zakład były zbyt niskiej jakości.

zek. Niestety — wrócił z niczym. Gdyby to było dobre, usłyszał od urzędnika zza biurka, to Japończycy już dawno by to wymyślili.

Tymczasem Japończycy do „Eltry” przybyli sami. Uczyć się? No, bez przesady. Ale powierzyli zakładowi produkcję najnowszej

„CD”. Tak, tak, to najnowszy laserowy odtwarzacz komputerowy. Przebój ostatnich lat produkowany w „Eltrze”? A dlaczego go nie? Chociaż, co tu kryć, bez pomocy Japończyków nie mogłoby być mowy o podobnej produkcji. Jeszcze nie.

„Sankej” CD-99 — wyjaśnia inż. Terlecki — to model, w którym główne podzespoły otrzymujemy od Japończyków. To skomplikowane urządzenie komputerowe, do produkcji którego trzeba specjalistycznych i wyjątkowo nowoczesnych narzędzi.

Aby uzmysłowić sobie doskonałość jego funkcjonowania, a co za tym idzie, precyzję wykonania, najlepiej wyobrazić sobie płytę CD w tysiąckrotnym powiększeniu. Miałaby ona wtedy 120 m średnicy, a plamki tworzące spiralną ścieżkę zapisu, byłyby około 0,5 mm szerokości, 1 mm długości, i 0,1 głębokości. Mimo tak ogromnego powiększenia.

Odległość natomiast między dwiema sąsiednimi ścieżkami wynosiłaby 1,6 mm, a czytnik laserowy musiałby przemieszczać się nad taką spiralną ścieżką z dokładnością do 0,1 mm. Nie jesteśmy jeszcze po prostu technologicznie przygotowani do samodzielnego wyprodukowania takiego odtwarzacza.

Oczywiście nie oznacza to, że w najbliższej przyszłości nie będziemy umieli tego robić. Na razie podpatrujemy Japończyków, uczymy się najnowszych technologii, zdobywamy doświadczenie.

Inżynier twierdzi, że to, co robią obecnie, jest przymiarką do rodzimych rozwiązań. Przekroczyli już wiele technologicznych progów i nie potknęli się. Dlaczego w tym przypadku miałoby być inaczej? Dziś cały przemysł elektroniczny w Polsce przygląda się japońskim osiągnięciom. A jutro?

W „Eltrze” są przekonani, że za kilka lat odtwarzacze kompaktowe naszej produkcji znajdą się w sklepach. Przymierza się do ich produkcji łódzka „Fonika”. Być może i oni włączą się do tej produkcji, kto wie?

W zakładzie najwyraźniej widać, jak w elektronicznym pędzie mkną w przyszłość najbardziej rozwinięte kraje świata. Tu też najlepiej rozumieją, że nie zabrac się do tego elektronicznego pociągu, oznacza ograniczyć tempo rozwoju, trafić w cywilizacyjną dziurę. Stąd ta zapobiegliwość. Chęć trzymania się za wszelką cenę, jeśli nie w czółtówce, to przynajmniej w peletonie.

I może dlatego to, co robią w „Eltrze”, napawa optymizmem. Mimo wszystko.

Włodzimierz KALETA



Przy taśmie montażowej wiele dziewcząt. One najbardziej nadają się do prac, które wymagają cierpliwości i precyzji.

A przecież i tutaj nie wszystko przebiega bez zgrzytów i trudności. W przykładowej szkole zawodowej, kształcą się radiomechaników i ślusarzy. Właśnie brak ślusarzy narzędziowych hamuje tempo wzrostu produkcji. Nie zawsze też wywiązują się ze swych zobowiązań kooperanci. Nikt tu jednak nie szuka usprawiedliwień, co zrobić, żeby nic nie robić, tylko szuka sposobów rozwiązywania kłopotów własnymi siłami.

Podpatrywanie Japończyków

Krążyła kiedyś anegdota o tym, jak to pewien wynalazca pobiegł zgłosić do odpowiedniej komórki swój najnowszy wynalaz-

generacji radiomagnetofonów. Prawdziwych miniaturowych „kombajnów” muzycznych.

Właśnie tu, produkuje się od kilku lat serię radiomagnetofonów „Sankej”, z najnowszym jej modelem „Sankej” CDR-99. Cóż to takiego? Przenośna, tranzystorowa miniwieża zasilana bateriami lub z sieci, złożona z magnetofonu dwukasetowego, ekalizera, tunera z syntezą częstotliwości, programowaniem elektronicznym i automatycznym wyborem stacji oraz odtwarzaczem CD — również programowanego elektronicznie, umożliwiającego automatyczny wybór określonego utworu z płyty kompaktowej.

Cóż się kryje za tymi dwiema literami

CPC 6128

EKSPLOZJA

```
10 * Eksplozja
20 INPUT "ILE RAZY POWTORZYĆ GŁOS EKSPLOZJI":I
30 FOR L=1 TO I
40 ENT -1,15,1,5
50 SOUND 1,30,750,5,0,1
60 ENV 1,1,15,1,15,-1,100
70 SOUND 1,140,0,0,1,0,12
80 NEXT
```

KARABIN MASZYNOWY

```
10 * Karabin maszynowy
20 INPUT "ILE RAZY POWTORZYĆ GŁOS KARABINU":I
30 FOR L=1 TO I
40 FOR K=1 TO 3
50 ENV 1,5,1,1,1,-5,1,5,3,1,1,-15,1
60 ENT -1,1,5,1,1,-5,1
70 SOUND 1,162,-7,8,1,1,7
80 SOUND 1,0,10
90 NEXT
100 NEXT
```

SYRENA

```
10 * Syrena
20 INPUT "ILE RAZY POWTORZYĆ GŁOS SYRENY":I
30 FOR L=1 TO I
40 ENV 2,10,1,1,2,1,1
50 ENT -2,12,2,0,5
60 p=40:g=1
70 FOR K=1 TO 50
80 IF K>35 AND p<45 THEN p=p+0,5
90 IF K>35 AND g>1 THEN g=g-1
100 IF K<20 AND g<12 THEN g=g+1
110 SOUND 2,p,-2,g,2,2
120 NEXT
130 NEXT
```

Standard MSX, a mikrokomputer SVI-738 X'PRESS

Józef RAJZER

Pod koniec 1986 roku w Centralnej Składnicy Harcerskiej oprócz mikrokomputerów TC-2048 pojawiły się również mikrokomputery SVI-738 X'PRESS firmy SPECTRAVIDEO z Hongkongu wraz z pewną liczbą urządzeń zewnętrznych i oprogramowaniem podstawowym.

SVI-738 należy do tzw. standard MSX, który jest próbą ujednoczenia produkcji mikrokomputerów domowych na wzór standardu IBM, jaki wytworzył się dla profesjonalnych mikrokomputerów personalnych. Wraz z SVI-738 omówiony zatem zostanie jednocześnie standard MSX.

Prekursorem tego standardu jest tokijska filia amerykańskiej firmy MICROSOFT, a głównie jej projektant, Mr. Mishi, który w 1983 r. zaprojektował model mikrokomputera MSX dla firmy SPECTRAVIDEO. W standardzie MSX przyjęto szereg ujednoczeń. W mikrokomputerach zastosowano mikroprocesor Z80 z taktowaniem 3.58 MHz. Pamięć mikrokomputera składa się z kilku bloków: bloku pamięci ROM o pojemności 32 kB, w której umieszczony jest interpreter rozszerzonego języka BASIC, bloku ROM 16 kB dla potrzeb mikrodysku, 8kB ROM dla potrzeb łącza szeregowego RS-232C, bloku 16 kB RAM przeznaczony do pamięci obrazu i bloku 64 kB jako pamięci operacyjnej. Dodatkowo można dołączyć zewnętrzną kartę pamięci RAM o pojemności 64 kB. Mikrokomputer ma wbudowany generator akustyczny o trzech niezależnych kanałach akustycznych i kanale szumowy. Generator wytwarza dźwięki w zakresie 8 oktaw. Amplituda dźwięku i kształt obwiedni mogą być tworzone programowo.

Klawiatura mikrokomputerów MSX nie jest ściśle znowelizowana, występują jednak te same grupy klawiszy w ten sam sposób rozmieszczone. Mamy więc:

- cztery duże klawisze do przesuwania kursora w 4 kierunkach,
- klawiaturę alfanumeryczną, która może być dostosowana do alfabetu danego kraju. Pierwsza partia sprowadzonych SVI ma klawiaturę angielską, następne mają mieć już klawiaturę polską. Klawisze alfanumeryczne po użyciu klawiszy sterujących CODE, SHIFT — CODE, GRAPM i SHIFT—GRAPM dają różne znaki graficzne, symbole arytmetyczne, litery charakterystyczne dla alfabetów: greckiego, niemieckiego, szwedzkiego itp. Łącznie dzięki klawiaturze alfanumerycznej można uzyskać 255 znaków,
- klawisze sterujące np. ENTER, TAB, CTRL, STOP, CLS/HM itp.
- klawisze funkcyjne F1 — F10, które mogą być zaprogramowane zgodnie z potrzebami użytkownika.

We wspólnej obudowie z mikrokomputerem znajduje się stacja mikrodysku o małej znanej u nas wersji dyskietek 3,5 cala. Dyskietka zapisywana jest jednostronnie z podwójną gęstością. Powierzchnia robocza dyskietki podzielona jest na 80 ścieżek po 9 sektorów. Pojemność dyskietki wynosi 360 kB, a szybkość transmisji 250 kb/s.

Teraz o tym, co można dołączyć do mikrokomputera, a można dość dużo. Mikrokomputer wyposażony jest w szereg gniazd, przez które można do mikrokomputera dołączyć:

- monitor kolorowy lub odbiornik telewizyjny. Obraz na ekranie tworzony jest za pomocą kontrolera graficznego TMS-9918A. Rastrowy wymiar ekranu w trybie grafiki wykreślnej wynosi 256x192 punkty. W trybie grafiki tekstowej możemy uzyskiwać wydruk w 24 wierszach po 40 znaków lub w 24 wierszach po 80 znaków. Obraz może być wyświetlany w 16 kolorach,
- dla potrzeb gier można podłączyć dwa manipulatory (joysticki) typu QUICSHOT MSX,

- magnetofon kasetowy SVI-767 jako pamięć zewnętrzną, w której znajduje się oprogramowanie firmowe,
- dodatkową stację mikrodysków 3,5 cala lub 5.25 cala, co znacznie zwiększa możliwości przetwarzania dość dużych zbiorów danych,
- drukarkę — przez równoległe łącze typu Centronics. Drukarka może drukować różnymi wzorami druku. Do SVI-738 można dołączać drukarki wyposażone w trakcję łącza.

Mikrokomputer wyposażony jest ponadto w gniazdo łącza RS-232-C, które umożliwia przekazywanie informacji między dwoma mikrokomputerami. Można łączyć mikrokomputery liniami telefonicznymi. W tym wypadku mikrokomputery są dołączone do linii telefonicznych przez modemy.

Ciekawostką jest ponadto możliwość zastosowania karty sprzęgu SVI-709, która umożliwia podłączenie mikrokomputera do lokalnej sieci mikrokomputerowej SVI-609 jako inteligentnej końcówki i korzystanie z zasobów informacji, znajdujących się na dysku stałym komputera głównego tej sieci.



SVI
SPECTRAVIDEO

Andrzej CETERA

Poprzednio rozpatrywano obrót zobrażenia na ekranie wokół początku układu współrzędnych.

Znajduje on również zastosowanie przy obrocie względem dowolnego punktu, nie będącego początkiem układu współrzędnych. W ogólnym wypadku następuje to przez przesunięcie początku układu współrzędnych do punktu obrotu, w obrocie wokół początku układu i przesunięciu układu współrzędnych do pierwotnego położenia. Obrót wektora położenia $[xy1]$ wokół punktu (m, n) o dowolny kąt θ może być wykonany za pomocą przekształcenia.

$$[xy1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -m & -n & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ m & n & 1 \end{bmatrix} = [XYH]$$

Po przekształceniu otrzymamy:

$$[XYH] = [xy1] \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ -m(\cos\theta-1) + n\sin\theta & -m(\sin\theta) - n(\cos\theta-1) & 1 \end{bmatrix}$$

Rozpatrzmy następujący przykład:

Przyjmijmy, że chcemy wykonać obrót zobrażenia o kąt 90° , w kierunku przeciwnym do obrotu wskazówek zegara, wokół osi prostopadłej do płaszczyzn XY i przechodzącej przez punkt $(4, 3)$.

Obrót względem początku układu współrzędnych możemy wykonać za pomocą macierzy transformacji:

$$\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Przesunięcie układu współrzędnych realizujemy za pomocą macierzy transformacji:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -4 & 3 & 1 \end{bmatrix}$$

Po wykonaniu przesunięcia i obrotu otrzymany rezultat należy sprowadzić do pierwotnego układu współrzędnych, co wykonamy wykorzystując wsteczną macierz transformacji:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 4 & 3 & 1 \end{bmatrix}$$

Całą zatem operację zapiszemy:

$$[XYH] = [xy1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -4 & 3 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 4 & 3 & 1 \end{bmatrix}$$

Po wymnożeniu macierzy transformacji otrzymamy:

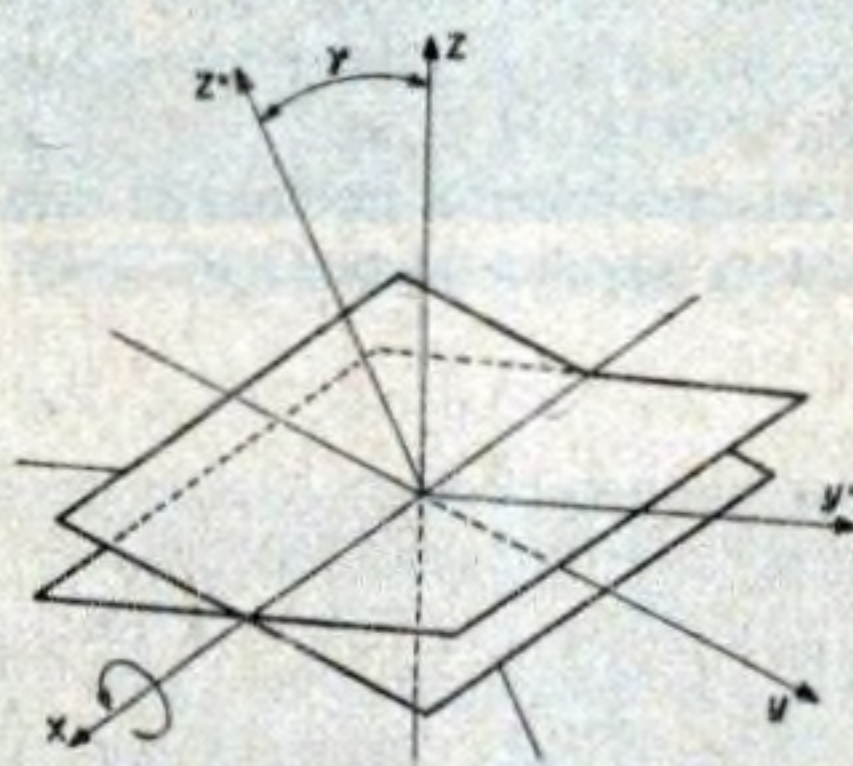
$$[XYH] = [xy1] \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 7 & -1 & 1 \end{bmatrix}$$

Rozważmy teraz przykłady, w których osią obrotu jest jedna z osi współrzędnych. Mamy więc trzy warianty do rozpatrzenia, w których jedna z osi będzie prostopadła do ekranu (oś półdotadnia):

- a) oś x biegnie do ekranu
- b) oś y biegnie do ekranu
- c) oś z biegnie do ekranu

Zadanie zatem jest zredukowane do obrotu w przestrzeni dwuwymiarowej i punkt (x, y, z) w pierwotnym układzie współrzędnych staje się w nowym układzie punktem (x', y', z') , gdzie:

dla obrotu wokół osi x



Względem osi X

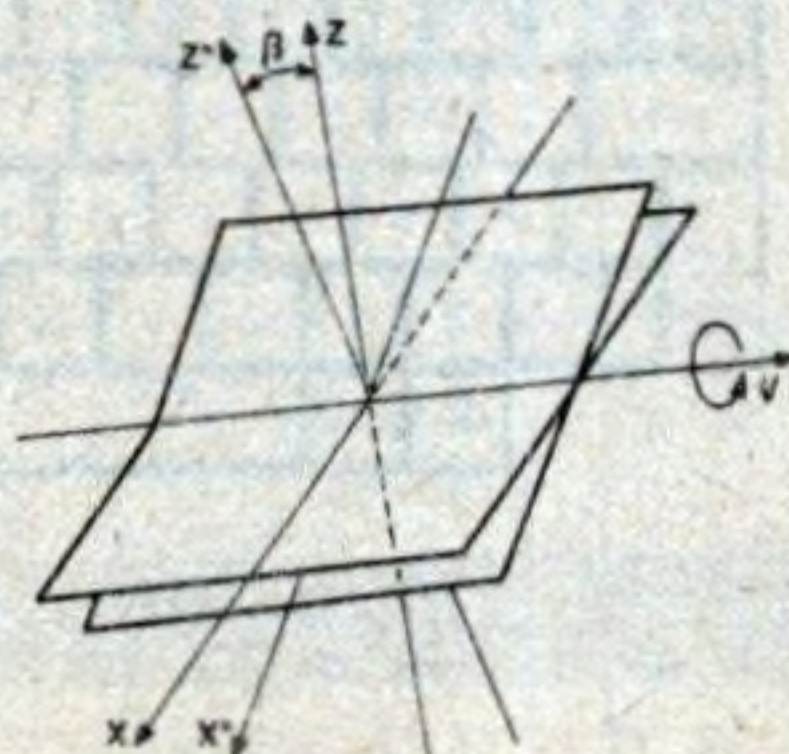
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{bmatrix}$$

$$\begin{aligned} x' &= x \\ y' &= y \cos\alpha + z \sin\alpha \\ z' &= -y \sin\alpha + z \cos\alpha \end{aligned}$$

a macierz transformacji ma postać:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha & 0 \\ 0 & -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

dla obrotu wokół osi y



Względem osi Y

$$\begin{bmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{bmatrix}$$

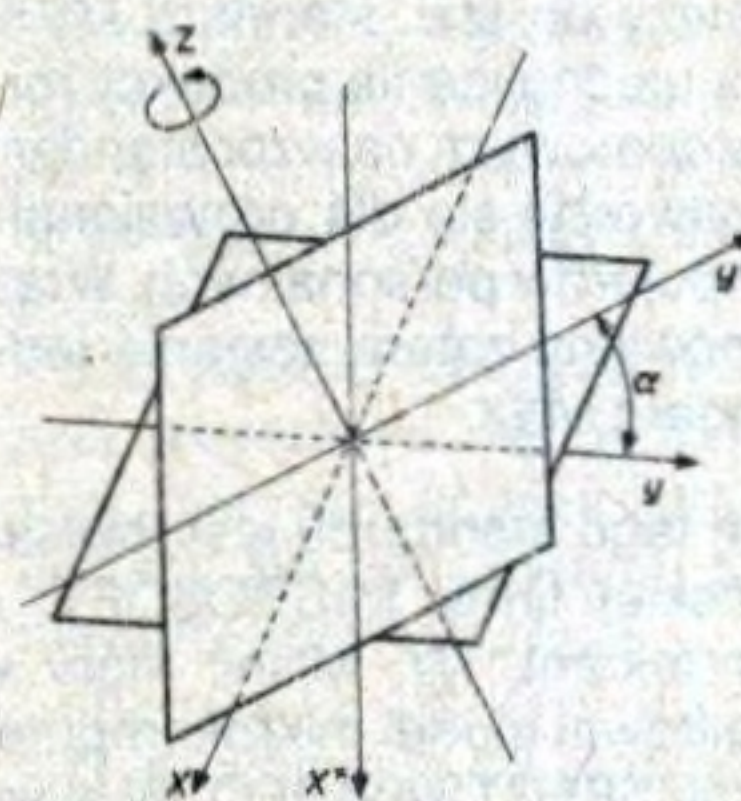
$$\begin{aligned} x' &= -z \sin\beta + x \cos\beta \\ y' &= y \\ z' &= z \cos\beta + x \sin\beta \end{aligned}$$

a macierz transformacji ma postać:

$$\begin{bmatrix} \cos\beta & 0 & -\sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

dla obrotu wokół osi z

$$\begin{aligned} x' &= x \cos\alpha + y \sin\alpha \\ y' &= -x \sin\alpha + y \cos\alpha \\ z' &= z \end{aligned}$$



Względem osi Z

$$\begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

a macierz transformacji ma postać:

$$\begin{bmatrix} \cos\alpha & \sin\alpha & 0 & 0 \\ -\sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

W sytuacji, gdy mamy do czynienia z figurami płaskimi zdefiniowanymi na płaszczyźnie, macierze transformacji przyjmują następującą postać:

R1, R2, R3

Omówiona problematyka została praktycznie wykorzystana w prezentowanym poniżej programie.

Program realizuje obroty zdefiniowanych figur względem dowolnie wybranego punktu lub osi układu współrzędnych. Możliwe jest także modyfikowanie figur przez dorysowywanie kolejnych elementów.

Współrzędne punktów definiujących figurę wprowadza się w jednostkach umownych, które są odmierzone na osiach układu współrzędnych. Mogą nimi być liczby całkowite i zmiennie-przecinkowe. Gdy cała zdefiniowana figura ma być widoczna na ekranie monitora, współrzędne muszą się zawierać w przedziałach

$$x: (-16, 16) \text{ i } y: (-9.5, 9.5)$$

W celu łatwiejszej orientacji figur na ekranie, należy wyświetlić siatkę punktów z dyskretnością jednostek umownych.

Wartości kątów obrotu wprowadzamy w stopniach. Wprowadzenie punktu o współrzędnych $x \ 998$ i dowolnej wartości współrzędnej y powoduje zakończenie definiowania figury.

Poniżej przedstawiamy przebieg i obsługę operatorską programu.

KOMUNIKAT OPERACJA

1. Całkowita widoczność rysowanej figury, gdy współrzędne zawierają się w przedziale:

x:(-16, 16),

y:(-9.5, 9.5)

Siatka współrzędnych (t/n)

Wprowadzamy literę „t”, gdy chcemy mieć na ekranie wyświetloną siatkę punktów, a gdy nie chcemy — literę „n”.

2. Podaj wsp. punktu x, y

Wprowadzamy wsp. (x, y) punktu definiującego figurę lub punkt kończący definiowanie.

3. Podaj kąt obrotu

Wprowadzamy w stopniach kąt obrotu.

4. Obrót względem punktu (t/n)

Wprowadzenie litery „t” spowoduje wybranie funkcji, obrót względem punktu i przejście do komunikatu 10. Wprowadzenie litery „n”, to wybranie funkcji obrót względem osi.

5. Obrót względem osi (x/y/z)

Wprowadzamy literę odpowiadającą jednej z osi x; y; z. W wyniku otrzymujemy zobrazowanie figury po obrocie.

```

10 REM *****
20 REM
30 REM Obrotu figury względem wybranego
40 REM punktu lub osi
50 REM
60 REM *****
70 DIM wsp(101,3),mat(3,3),nwsp(101,3)
80 MODE 1:INK 0,13:INK 1,3:INK 2,12:INK 3,0
90 BORDER 0:PEN 1:PAPER 2
100 CLS
110 PRINT "Całkowita widoczność rysowanej"
120 PRINT "figury gdy współrzędne zawierają"
130 PRINT "się w przedziale x:(-16,16),y:(-9.5,9.5)"
140 PRINT
150 PRINT "Siatka współrzędnych (t/n)"
160 z%=INKEY$
170 IF z%="" THEN 160
180 IF z%="t" THEN f1=1:GOTO 210
190 IF z%="n" THEN f1=0:GOTO 210
200 GOTO 160
210 GRAPHICS PAPER 1:REM Tlo grafiki
220 ORIGIN 320,208,0,640,16,400
230 CLEAR INPUT
240 WINDOW 1,40,25,25
250 GOSUB 1570:REM Rys. układu wspol.
260 i=0:REM Licznik punktow
270 INPUT "Podaj wsp. punktu x,y";x,y
280 IF x>998 THEN 350
290 i=i+1
300 wsp(i,1)=x#20
310 wsp(i,2)=y#20
320 wsp(i,3)=1
330 GOSUB 1050:REM Wykresl. punkt lub odcinek
340 GOTO 270
350 INPUT "Podaj kat obrotu";ak
360 ak=ak#PI/180
370 ac=COS(ak):as=SIN(ak)
380 PRINT "Obrot względem punktu (t/n)"
390 nk%=INKEY$
400 IF nk%="" THEN 390
410 IF nk%="t" THEN 440
420 IF nk%="n" THEN 480
430 GOTO 390
440 INPUT "Podaj wsp.puktu obrotu";p,q

```

```

450 p=-p#20;q=-q#20
460 GOSUB 1120:REM Ustaw macierz przesuniecie
470 GOTO 620
480 PRINT "Obrot względem osi (x/y/z)"
490 nk%=INKEY$
500 IF nk%="" THEN 490
510 IF nk%="x" THEN 540
520 GOSUB 1960:REM Ustawienie macierzy
530 GOTO 590
540 IF nk%="y" THEN 570
550 GOSUB 2040:REM Ustawienie macierzy
560 GOTO 590
570 IF nk%="z" THEN 490
580 GOSUB 1410:REM Ustawienie macierzy
590 GOSUB 1240:REM Mnozenie macierzy
600 GOSUB 1480:REM Zobrazowanie
610 GOTO 690
620 GOSUB 1240:REM Mnozenie
630 GOSUB 1410:REM Ustaw macierz obrotu
640 GOSUB 1240:REM Mnozenie macierzy
650 p=-p;q=-q
660 GOSUB 1120:REM Ustaw macierz przesuniecie
670 GOSUB 1240:REM Mnozenie macierzy
680 GOSUB 1480:REM Zobrazowanie
690 PRINT "Czy koniec (t/n)"
700 nk%=INKEY$
710 IF nk%="" THEN 700
720 IF nk%="t" THEN 2120
730 IF nk%="n" THEN 750
740 GOTO 700
750 PRINT "Czy od początku (t/n)"
760 nk%=INKEY$
770 IF nk%="" THEN 760

780 IF nk%="t" THEN 80
790 IF nk%="n" THEN 810
800 GOTO 760
810 PRINT "Czy obrot (t/n)"
820 nk%=INKEY$
830 IF nk%="" THEN 820
840 IF nk%="t" THEN 350
850 IF nk%="n" THEN 870
860 GOTO 820
870 PRINT "Czy dorysowywanie (t/n)"
880 nk%=INKEY$
890 IF nk%="" THEN 880
900 IF nk%="t" THEN 930
910 IF nk%="n" THEN 690
920 GOTO 880
930 CLS
940 GRAPHICS PAPER 1:REM Tlo grafiki
950 ORIGIN 320,208,0,640,16,400
960 CLEAR INPUT
970 WINDOW 1,40,25,25
980 GOSUB 1570:REM Rys.ukladu wspolrzednych
990 GOSUB 1480:REM Rysowanie figury
1000 GOTO 270
1010 REM *****
1020 REM PODPROGRAMY
1030 REM *****
1040 REM
1050 REM Rysowanie punktu lub odcinka
1060 REM
1070 GRAPHICS PEN 3:REM stare wspolrzedne
1080 IF i=1 THEN PLOT wsp(i,1),wsp(i,2)
1090 IF i>1 THEN DRAW wsp(i,1),wsp(i,2)
1100 RETURN
1110 REM
1120 REM Ustawianie macierzy przesuniecie
1130 REM
1140 mat(1,1)=1:mat(1,2)=0:mat(1,3)=0
1150 mat(2,1)=0:mat(2,2)=1:mat(2,3)=0
1160 mat(3,1)=p:mat(3,2)=q:mat(3,3)=1
1170 FOR k=1 TO 3:REM Zerowanie nwsp
1180 FOR l=1 TO 3
1190 nwsp(k,l)=0
1200 NEXT l
1210 NEXT k
1220 RETURN
1230 REM
1240 REM Mnozenie macierzy
1250 REM
1260 FOR n=1 TO 3
1270 FOR k=1 TO 3
1280 FOR m=1 TO 3
1290 nwsp(n,k)=nwsp(n,k)+wsp(n,m)*mat(m,k)
1300 NEXT m
1310 NEXT k
1320 NEXT n
1330 FOR n=1 TO 3
1340 FOR m=1 TO 3
1350 wsp(n,m)=nwsp(n,m)
1360 nwsp(n,m)=0
1370 NEXT m
1380 NEXT n
1390 RETURN
1400 REM
1410 REM Ustawianie macierzy obrotu
1420 REM
1430 mat(1,1)=ac:mat(1,2)=as:mat(1,3)=0
1440 mat(2,1)=-as:mat(2,2)=ac:mat(2,3)=0
1450 mat(3,1)=0:mat(3,2)=0:mat(3,3)=1
1460 RETURN
1470 REM
1480 REM Rysowanie nowej figury
1490 REM
1500 GRAPHICS PEN 0:REM PIORO nowych wspol.
1510 PLOT wsp(1,1),wsp(1,2)
1520 FOR n=2 TO i
1530 DRAW wsp(n,1),wsp(n,2)
1540 NEXT

```

```

1550 RETURN
1560 REM
1570 REM Rysowanie ukkladu wspolrzednych
1580 REM
1590 CLG 1
1600 GRAPHICS PEN 2:REM Piorko ukkladu wspol.
1610 FOR a=-200 TO 200 STEP 20
1620 PLOT -2,a
1630 DRAWR 4,0
1640 NEXT a
1650 PLOT 0,-208
1660 DRAW 0,208
1670 TAG
1680 MOVER -16,-20
1690 PRINT "Y";
1700 MOVER -310,0
1710 PRINT "20";
1720 TAGOFF
1730 PLOT -307,170:REM Kreslenie podzialki
1740 DRAWR 0,-4
1750 DRAWR 0,2
1760 DRAWR 20,0
1770 DRAWR 0,2
1780 DRAWR 0,-4
1790 IF f1=1 THEN 1800 ELSE 1850
1800 FOR b=-200 TO 200 STEP 20
1810 FOR a=-320 TO 320 STEP 20
1820 PLOT a,b
1830 NEXT a
1840 NEXT b
1850 FOR a=-320 TO 320 STEP 20
1860 PLOT a,-2
1870 DRAWR 0,4
1880 NEXT a
1890 PLOT -320,0
1900 DRAW 320,0
1910 TAG
1920 MOVER -16,-4
1930 PRINT "X";
1940 TAGOFF
1950 RETURN
1960 REM
1970 REM ustawianie macierzy obrotu
1980 REM względem osi X
1990 REM
2000 mat(1,1)=1:mat(1,2)=0:mat(1,3)=0
2010 mat(2,1)=0:mat(2,2)=ac:mat(2,3)=as
2020 mat(3,1)=0:mat(3,2)=-as:mat(3,3)=ac
2030 RETURN
2040 REM
2050 REM ustawianie macierzy obrotu
2060 REM względem osi Y
2070 REM
2080 mat(1,1)=ac:mat(1,2)=0:mat(1,3)=-as
2090 mat(2,1)=0:mat(2,2)=1:mat(2,3)=0
2100 mat(3,1)=as:mat(3,2)=0:mat(3,3)=ac
2110 RETURN
2120 REM
2130 REM Koniec
2140 REM
2150 WINDOW 1,40,1,25
2160 CLS
2170 CLG 1
2180 END

```

6. Czy koniec (t/n)

Wybranie litery „t” kończy działanie programu. Kontynuacja programu, gdy wybierzemy literę „n”.

7. Czy od początku (t/n)

Wybranie litery „t” powoduje przejście do komunikatu 1. Wybranie „n” kontynuacja programu. Wybranie „t” przejście do komunikatu 3, litera „n” dalszy ciąg programu.

8. Czy obrót (t/n)

Wybranie litery „t” to wybranie funkcji dorysowywania i przejście do komunikatu 2. Wybranie „n” powoduje przejście do komunikatu 6.

9. Czy dorysowywanie (t/n)

10. Podaj wsp. punktu obrotu

Wprowadzamy współrzędne punktu obrotu. Otrzymujemy nowe zobrazowanie i przejście do komunikatu 6.

Animacja kwadratu

```

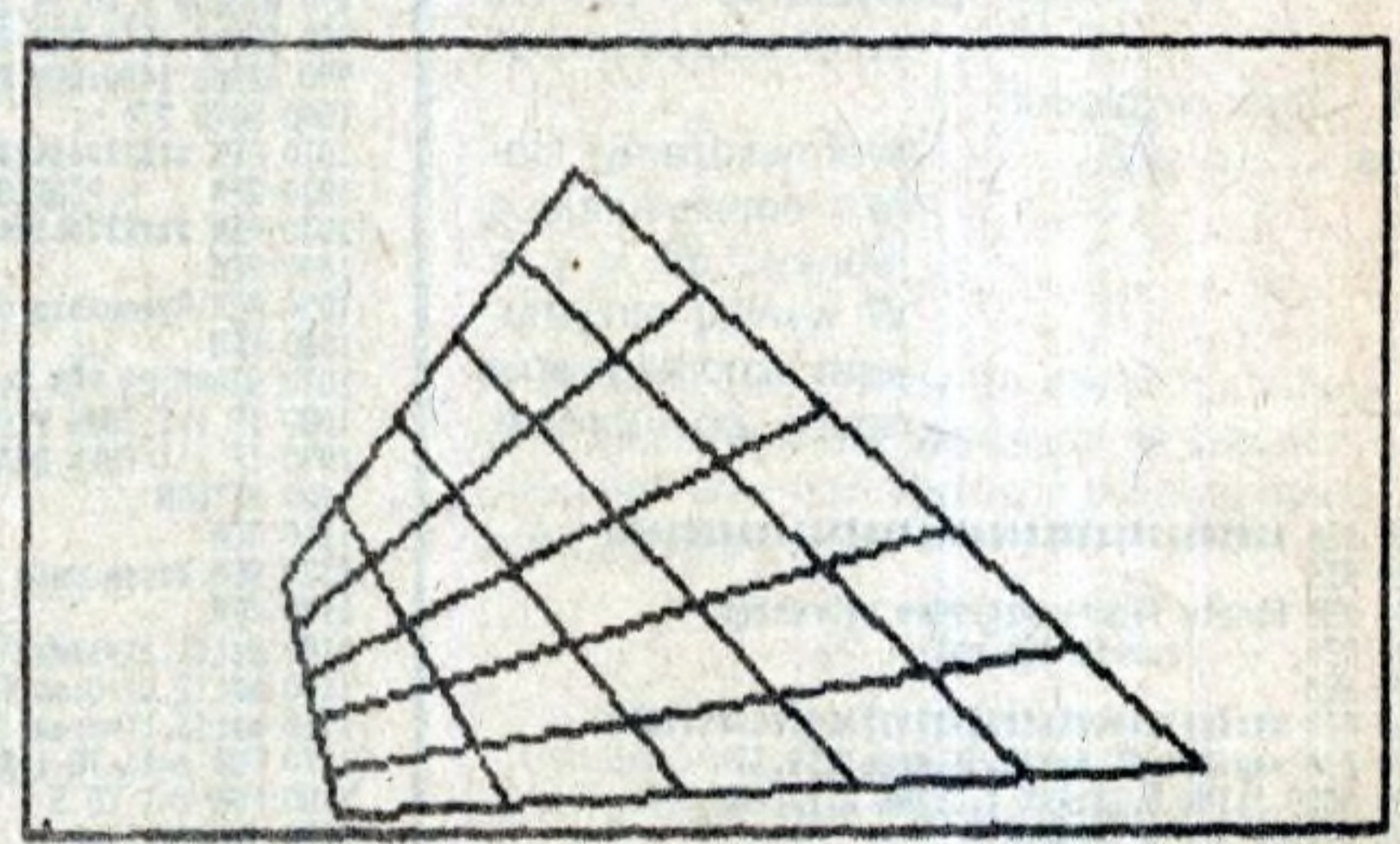
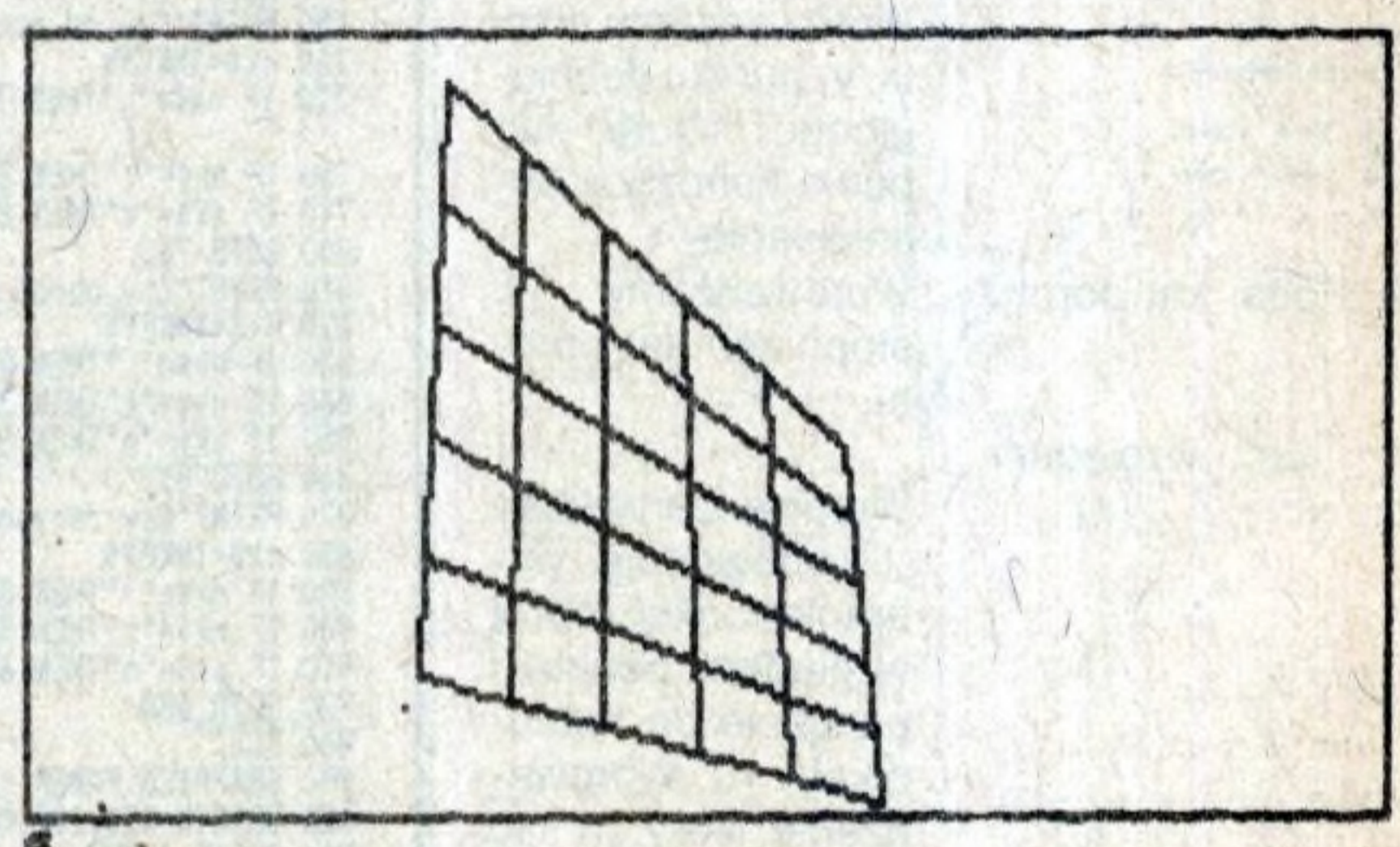
100 REM ANIMACJA KWADRATU
110 CLEAR 32755: FOR n=32755 TO
32757: READ b: POKE n,b: NEXT n
REM przemieszczenie blokow
1200 DATA 33,0,120,17,0,72,1,0,1
1300 7,176,204
CLS: BEEP .02,26: BEEP .01
140 PLOT 1,40: DRAW 253,0: DRAW
0,120: DRAW -253,0: DRAW 0,-120
140 DIM a$(512): DIM x(8,4): DI
M C(8,4)
150 REM oblicz. wspolrzednych
160 FOR m=1 TO 4: FOR n=1 TO 8:
X(m,n)=INT((251*PI)/360):
X(m,n)=INT((251*PI)/360):
X(m,n)=INT((251*PI)/360):
X(m,n)=INT((251*PI)/360)
NEXT n: NEXT m
170 FOR m=1 TO 4: FOR n=1 TO 8:
X(m,n)=X(m,n)-X(1,n)/8:
X(m,n)=X(m,n)-X(1,n)/8:
X(m,n)=X(m,n)+X(1,n)/8:
X(m,n)=X(m,n)+X(1,n)/8:
NEXT n: NEXT m
180 FOR m=1 TO 4: FOR n=1 TO 8:
X(m,n)=X(m,n)-X(1,n)/8:
X(m,n)=X(m,n)-X(1,n)/8:
X(m,n)=X(m,n)+X(1,n)/8:
X(m,n)=X(m,n)+X(1,n)/8:
NEXT n: NEXT m
190 FOR m=1 TO 4: FOR n=1 TO 8:
X(m,n)=X(m,n)-X(1,n)/8:
X(m,n)=X(m,n)-X(1,n)/8:
X(m,n)=X(m,n)+X(1,n)/8:
X(m,n)=X(m,n)+X(1,n)/8:
NEXT n: NEXT m
200 FOR m=1 TO 4: FOR n=1 TO 8:
X(m,n)=X(m,n)-X(1,n)/8:
X(m,n)=X(m,n)-X(1,n)/8:
X(m,n)=X(m,n)+X(1,n)/8:
X(m,n)=X(m,n)+X(1,n)/8:
NEXT n: NEXT m
210 FOR m=1 TO 4: FOR n=1 TO 8:
X(m,n)=X(m,n)-X(1,n)/8:
X(m,n)=X(m,n)-X(1,n)/8:
X(m,n)=X(m,n)+X(1,n)/8:
X(m,n)=X(m,n)+X(1,n)/8:
NEXT n: NEXT m
220 FOR m=1 TO 4: FOR n=1 TO 8:
X(m,n)=X(m,n)-X(1,n)/8:
X(m,n)=X(m,n)-X(1,n)/8:
X(m,n)=X(m,n)+X(1,n)/8:
X(m,n)=X(m,n)+X(1,n)/8:
NEXT n: NEXT m
230 IF INKEY$("<>") THEN GO TO 12
240 NEXT Z
250 FOR Z=0 TO 5: PLOT X(m,1)+Z
*X3,C(m,1)+Z*Y3: DRAW X(m,2)-Z*X
4-Y3,C(m,2)-Z*Y4-PEEK 23
260 IF INKEY$("<>") THEN GO TO 12
270 NEXT Z
280 POKE 32758,64: POKE 32761,1
28+16*(n-1): RANDOMIZE USR 32755
290 BEEP .02,n+26: CLS: NEXT n
300 PRINT BRIGHT 1;a$
310 REM wyswietlanie "tam"
320 FOR r=1 TO 12: PRINT AT 16,
r: FOR n=1 TO 8: BEEP .01,n: P
RUSE r/2
330 POKE 32761,64: POKE 32758,1
28+16*(n-1): RANDOMIZE USR 32755
330 IF INKEY$("<>") THEN GO TO 12
340 NEXT n
350 REM wyswietl "z powrotem"
360 FOR n=7 TO 2 STEP -1: BEEP
.01,n: PAUSE r/2

```

```

360 POKE 32761,64: POKE 32758,1
28+16*(n-1): RANDOMIZE USR 32755
370 IF INKEY$("<>") THEN GO TO 12
380 NEXT n: NEXT r: GO TO 120
9999 REM zapis na taśmie
9999 CLEAR: RESTORE: SAVE "ANI
MACJA" LINE 100: VERIFY "": GO T
0 100

```



ZX SPECTRUM

Przeliczanie

```

1 CLS: PRINT BRIGHT 1;
AT 0,12: "ENTER=CLS";
AT 14,12: "0 LUB 1 "
OR LET a$=""
2 PAUSE 0: LET b$=INKEY$:
IF b$="CH" THEN GO TO 1
4 IF (b$="0" OR b$="1") AND L
EN a$+b$: THEN BEEP .01,14: LET
IN a$+b$: PRINT AT 11,2;a$: " = B
5 GO TO 3
6 SAVE "BIN" LINE 1

```

Budżet domowy

S. WASILEWSKI

Każdy z nas prowadzi działalność finansową na mniejszą lub większą skalę. Jeśli chcesz ocenić i usprawnić gospodarowanie własnymi zasobami pieniężnymi, skorzystaj z programu „BUDŻET DOMOWY”, opracowanego na mikrokomputer Commodore 64 (lub C-128 w trybie pracy C-64).

„BUDŻET DOMOWY” ma na tyle uniwersalną konstrukcję, że może być wykorzystywany zarówno przez pojedyncze osoby, jak i grupy do rejestrowania dowolnej formy działalności finansowej. Nie wymaga tworzenia i korzystania z dodatkowych zbiorów danych. Wszystkie niezbędne informacje gromadzone są na bieżąco w nim samym. Z tego względu zamieszczona postać programu jest wyjściową dla wszystkich jego zastosowań.

Obszar działań finansowych rejestrowanych przez program określa użytkownik. Następuje to podczas wprowadzania elementów struktury tego obszaru. Dla przykładu weźmy pod uwagę obszar działań finansowych rodziny. Można go podzielić na grupy artykułów: spożywcze, mięso, wędliny i inne. W obrębie grupy można określić pojedyncze artykuły (mleko, sery, pieczywo i inne w grupie artykułów spożywczych). Tworząc taką sieć powiązań elementów struktury za pomocą programu, użytkownik sam ustala obszar pracy programu i sposób odwoływania się do poszczególnych artykułów w trakcie prowadzenia ewidencji finansowej. Dla poszczególnych artykułów, grup artykułów, a także dla całej istniejącej struktury (globalnie) prowadzona jest ewidencja finansowa w zakresie:

- wartości środków finansowych planowanych na zakupy (pozycja plan);
- wartość wydatków (pozycja wydatki).

Wyświetlając aktualny stan poszczególnych pozycji „BUDŻET DOMOWY” wyprowadza dodatkowo pozycję bilans, która jest różnicą między planem i wydatkami.

Z punktu widzenia użytkownika w programie można wyróżnić trzy poziomy pracy. Na każdym poziomie istnieje zestaw funkcji możliwych do wykonania. Są one każdorazowo prezentowane na ekranie za pomocą jednoznakowego symbolu wywołującego funkcję oraz wyrazu opisującego jej znaczenie. Przed rozpoczęciem eksploatacji programu „BUDŻET DOMOWY” należy zapoznać się z jego obsługą.

Poziom pierwszy — występuje po uruchomieniu programu komendą RUN, a także po zakończeniu pracy na poziomie drugim. Umożliwia globalne operacje na istniejącej strukturze finansowej. Zawiera funkcje:

KONIEC — kończy pracę programu. Po pojawieniu się pytania DATA? należy podać datę (MM.DD) lub inny dowolny łańcuch pięciodziesiętny identyfikujący aktualnie utworzoną wersję „BUDŻETU DOMOWEGO”. Program kończy się po wprowadzeniu wiersza postaci: SAVE BUDŻET xxxxx gdzie xxxxx jest podanym identyfikatorem wersji. Użytkownik może teraz skopiować aktualną wersję programu na taśmę (jeśli ją akceptuje).

BUDŻET — zobrazowuje globalny stan gospodarki finansowej, jaką obejmuje program. Zawiera pozycje: plan, wydatki i bilans.

PLAN — pozwala uaktualnić globalną pozycję plan. Wartość początkowa zero, a kolejne po-

dawane wartości są sumowane. powoduje przejście do poziomu drugiego. Jeżeli była utworzona struktura finansowa, to jest wyprowadzana na ekranie jej pierwsza część.

ZEROWANIE — automatyczne wyzerowanie wszystkich pozycji w utworzonej strukturze finansowej bez naruszenia tej struktury.

Poziom drugi — występuje po zainicjowaniu funkcji LISTA z poziomu pierwszego lub funkcji KONIEC z poziomu trzeciego. Umożliwia operacje w obrębie grup artykułów lub artykułów nie tworzących grupy. Zawiera funkcje:

KONIEC — powrót na poziom pierwszy.

DOPISANIE — umożliwia dopisanie na listę nowego elementu struktury. Wymagane jest podanie nazwy. Dostęp do dopisanych na tym poziomie elementów uzyskasz po ponownym wejściu z poziomu pierwszego (wykonanie kolejno funkcji: KONIEC, LISTA).

GRUPA — powoduje przejście na poziom drugi. Jeśli wskazany element ma tworzyć grupę, to na poziomie trzecim udostępniane są operacje pozwalające utworzyć listę artykułów wchodzących w skład tej grupy. Jeśli wskazany element tworzy już grupę, to na ekranie jest wyprowadzana pierwsza część listy artykułów wchodzących w jej skład.

USUNIĘCIE — umożliwia usunięcie z istniejącej struktury wskazanego elementu. Jeśli jest to grupa, to usuwana jest wraz ze wszystkimi wchodzącymi w jej skład artykułami. Globalna suma wydatków zarejestrowanych w programie nie jest pomniejszana o wartość wydatków usuniętych struktur.

LISTA — wyprowadza (jeśli istnieje) kolejną część listy z tego poziomu.

PLAN — pozwala zmienić wartość planu wskazanego elementu struktury (grupy artykułów lub pojedynczego artykułu). Wartość początkowa zero, a kolejne podawane wartości są sumowane.

WYDATKI — uaktualnia wartość wydatków związanych ze wskazanym elementem. Wartość początkowa zero, a kolejne podawane wartości są sumowane.

ZEROWANIE — wykonuje automatyczne wyzerowanie wszystkich pozycji we wskazanym elemencie struktury bez naruszania tej struktury.

Poziom trzeci — występuje po zainicjowaniu funkcji GRUPA z poziomu drugiego. Umożliwia operacje w obrębie grupy artykułów wskazanej na poziomie drugim, a udostępnianych na poziomie trzecim. Zawiera funkcje podobne jak na poziomie drugim. Niedostępna jest tylko funkcja GRUPA.

Na poziomach drugim i trzecim można poruszać się w obrębie wyświetlonej na ekranie listy za pomocą klawiszy przesuwających kursor

w dół i w górę. Takie funkcje jak: **PLAN, WYDATKI, ZEROWANIE** i **GRUPA** dotyczą zawsze tych elementów struktury, na które aktualnie wskazuje kursor.

```
10 REM ***** BUDZET DOMOWY *****
11 GOSUB30:POKE53280,0:POKE53281,0:POKET
,192:PRINTC$:POKET,4
12 FORI=HTO739:POKEI,0:NEXT:POKEL,2:POKE
D,9:POKEG,0
13 RESTORE:READL$,A:B=718:GOSUB40:B=700:
A=2001:GOSUB40:GOSUB20:GOTO100
20 RESTORE:READL$:PRINTC$:POKEE,19:PRINT
:PRINTA$D$L$"      BUDZET DOMOWY      C S.
W.";
21 PRINT:PRINT"K"B$" - KONIEC      "A$L"B$"
" - LISTA      ";
22 IFPEEK(L)=0THENPRINTA$"G" - GRUPA"
23 IFPEEK(L)=1THENPRINT
24 IFPEEK(L)=2THENPRINTA$"B"B$" - BUDZET
":GOTO26
25 PRINTA$"W"B$" - WYDATKI      "A"D"B$" -
DOPISANIE "A"U"B$" - USUNIECIE";
26 PRINTA$"P"B$" - PLAN";
27 PRINTSPC(19)A$"Z"B$" - ZEROWANIE";:RE
TURN
30 A$=CHR$(18):B$=CHR$(146):C$=CHR$(147)
:H$=CHR$(19):R$=A$:FORI=1TO40:R$=R$+" "
31 S$=S$+"      ":NEXT:R$=R$+B$:E=214:D=68
2:G=684:H=680:L=679:T=648:D$="DATA"
32 DEFFNB(X)=700+PEEK(L)*20+X*2:DEFFNP(Y
)=PEEK(Y+PEEK(L)):S=58732:X$="RETURN:"
33 DEFFNA(Q)=PEEK(Q)+PEEK(Q+1)*256:Y$=CH
R$(17):Z$=CHR$(145):RETURN
40 POKEB,A-INT(A/256)*256:POKEB+1,INT(A/
256):RETURN
50 PRINT"*      "L$
55 PRINTLEFT$(S$,40)Z$TAB(8)P,W,P-W:RETU
RN
60 POKEE,FNP(H)*2:SYSS:PRINT"*":IFK=OTHE
N66
65 POKEE,K*2:SYSS:PRINTA$"*"B$"
66 POKEH+PEEK(L),K:RETURN
70 POKET,192:PRINTC$:POKEE,2:PRINT:PRINT
"GOSUB"A":POKE65,PEEK(61):POKE66,";
71 PRINT"PEEK(62):GOTO"Y":PRINTC$;H$:PO
KE631,13:POKE198,1:END
75 W$=STR$(A)+D$+STR$(W+WR)
80 POKET,192:IFPEEK(688)=0THENPRINTC$
81 POKEE,2+PEEK(688):PRINT:PRINTW$:POKET
,4:POKE688,PEEK(688)+1:RETURN
85 GOSUB80
90 POKET,192:POKEE,2+PEEK(688):PRINT:PRI
NT"GOSUB30:GOTO"Y;H$:FORI=OTOPEEK(688)+1
91 POKE631+I,13:NEXT:POKE198,I:POKE688,0
:END
95 POKEE,16:PRINT:PRINTA$Y$WR$" WARTOSC"
:INPUTWR$:PRINTB$:WR=VAL(WR$):RETURN
96 FORI=1TO2:W$=STR$(A+I)+D$+"0":GOSUB80
:NEXT:GOTO90
100 POKET,4:POKEE,16:PRINT:PRINTY$:LEFT$
(S$,80);
101 GETL$:IFL$=""THEN101
102 ONPEEK(L)+1GOTO106,107,103
103 IFL$="B"THEN800
104 IFL$="L"THENPOKEL,0:GOSUB20:GOTO250
105 GOTO113
106 IFL$="G"THEN200
107 IFL$=Z$THENZ=-1:GOTO300
108 IFL$=Y$THENZ=1:GOTO300
109 IFL$="L"THEN250
110 IFL$="W"THEN550
```

```
111 IFL$="D"THEN500
112 IFL$="U"THEN350
113 IFL$="K"THEN150
114 IFL$="Z"THEN650
115 IFL$="P"THEN600
116 GOTO101
149 REM*****
149 REM      *** KONIEC ***
150 ONPEEK(L)+1GOTO12,170,155
155 PRINTC$:INPUT"DATA";L$:I=5-LEN(L$):I
FI<0THENI=0
160 L$=LEFT$(L$,5)+LEFT$(S$,I):W$="2000"
+D$+CHR$(34)+L$+CHR$(34):Y=165:GOTO85
165 POKET,4:READL$:PRINTC$"SAVE"CHR$(34)
"BUDZET "L$:END
170 POKEL,0:A=FNA(FNB(FNP(D))):POKEE,20:
PRINT:PRINTLEFT$(S$,40);:GOSUB20:GOTO260
198 REM*****
199 REM      *** GRUPA ***
200 IFPEEK(H)=0THEN100
205 A=FNA(FNB(PEEK(H))):B=720:A=A+4:GOSU
B40:A=A-4:Y=210:GOTO70
210 POKET,4:READL$,L$,L$,L$,A
215 POKEL,1:FORI=722TO737:POKEI,0:NEXT
220 B=738:GOSUB40:POKE681,0:POKE683,9:PO
KE685,0:GOSUB20
248 REM*****
249 REM      *** LISTA ***
250 A=FNA(FNB(9)):IFA=0THEN100
255 FORI=OTO1:POKEFNB(0)+I,PEEK(FNB(FNP(
G)))
260 PRINTH$A$"*      NAZWA ARTYKULU / GRUPY
ARTYKULOW      ";
261 PRINT"      PLAN      WYDATKI
BILANS      "B$;
265 PRINTS$S$S$S$:POKED+PEEK(L),1:K=0:Y=
270:GOTO70
270 POKET,4:B=FNB(K+1):GOSUB40:IFK=8THEN
285
275 READA,W,P,L$:POKEE,2+K*2:SYSS:GOSUB5
0:K=K+1:IFATHEN70
280 B=FNB(9):GOSUB40
285 POKEG+PEEK(L),K:K=1:GOSUB65:GOTO100
298 REM*****
299 REM      *** POZYCJONOWANIE KURSORA ***
300 IFFNP(D)>FNP(G)THEN100
305 K=FNP(H):IFK=9THENPOKEH+PEEK(L),0
310 IFK=0THEN335
315 IFK+Z<FNP(D)THEN340
320 IFK+Z>FNP(G)THEN100
325 K=K+Z:IFFNA(FNB(K))=0THEN325
330 GOSUB60:GOTO100
335 IFZ<>1THEN100
340 K=FNP(D):GOSUB65:GOTO100
348 REM*****
349 REM      *** USUNIECIE ***
350 K=FNP(H):IFK=0ORK=9THEN100
355 POKEE,17:PRINT:PRINTA$Y$"USUNAC T/N"
:INPUTL$:IFL$="T"THEN370
360 IFL$="N"THEN100
365 POKEE,17:PRINT:PRINTY$:LEFT$(S$,40);
:GOTO355
370 IFPEEK(L)THEN405
375 A=FNA(FNB(PEEK(H))):Y=380:GOTO70
380 READL$,L$,L$,L$,P
385 IFP=0THEN405
```

```

390 A=P:Y=395:GOTO70
395 READA:B=686:GOSUB40:FORI=0TO3:W$=STR
$(P+I):GOSUB80:NEXT:Y=400:GOTO90
400 P=FNA(686):GOTO385
405 A=FNA(FNB(FNP(H))):FORI=0TO4-PEEK(L)
:W$=STR$(A+I):GOSUB80:NEXT:Y=410:GOTO90
410 K=FNP(H):B=FNB(K):POKEB,0:POKEB+1,0:
I=K+1
415 P=FNA(FNB(I)):IFP=0ANDI<>9THENI=I+1:
GOTO415
420 I=K-1
425 A=FNA(FNB(I)):IFA=0ANDI<>0THENI=I-1:
GOTO425
430 W$=STR$(A)+X$+D$+STR$(P):Y=435:GOTO8
5
435 K=FNP(H):IFK<>FNP(G)THEN450
440 POKEG+PEEK(L),FNP(G)-1:IFFNP(G)<FNP(
D)THENPOKEG+PEEK(L),0:GOTO450
445 IFFNA(FNB(FNP(G)))=0THEN440
450 IFK<>FNP(D)THEN465
455 POKED+PEEK(L),FNP(D)+1:IFFNP(G)<FNP(
D)THENPOKED+PEEK(L),9:GOTO465
460 IFFNA(FNB(FNP(D)))=0THEN455
465 POKET,4:POKEE,FNP(H)*2-1:PRINT:PRINT
LEFT$(S$,80):POKEH+PEEK(L),0:GOTO100

498 REM*****
499 REM      *** DOPISANIE ***
500 POKEE,18:SYSS:PRINTA$"NAZWA":INPUTW
R$:PRINTB$
505 RESTORE:READL$,P,L$,L$,W:IFPEEK(L)TH
EN515
510 W$="2001"+X$+D$+STR$(W):GOSUB80:W$=S
TR$(W+4)+X$+D$+"0":GOSUB80:GOTO525
515 A=FNA(FNB(FNP(H))):Y=520:GOTO70
520 READP:W$=STR$(A)+X$+D$+STR$(W):GOSUB
80
525 W$="2004"+D$+STR$(W+5-PEEK(L)):GOSUB
80:W$=STR$(W)+X$+D$+STR$(P):GOSUB80
530 FORI=1TO2:W$=STR$(W+I)+D$+"0":GOSUB8
0:NEXT:W$=STR$(W+3)+D$+WR$:GOSUB80
535 Y=100:GOTO90

548 REM*****
549 REM      *** WYDATKI ***
550 K=FNP(H):IFK=0ORR=9THEN100
555 WR$="WYDATKI":GOSUB95:IFPEEK(L)=0THE
N570
560 A=FNA(700+PEEK(H)*2):Y=565:GOTO70
565 READL$,W1:A1=A+1
570 A=FNA(FNB(K)):Y=575:GOTO70
575 POKET,4:READL$,W,P:POKEE,K*2:PRINT:P
RINT$:W=W+WR:GOSUB55:A=A+1:W=W-WR
580 GOSUB75:IFPEEK(L)THENW=W1:A=A1:GOSUB
75
585 RESTORE:READL$,L$,W:A=2002:GOSUB75:Y
=100:GOTO90

499 REM      *** DOPISANIE ***
500 POKEE,18:SYSS:PRINTA$"NAZWA":INPUTW
R$:PRINTB$
505 RESTORE:READL$,P,L$,L$,W:IFPEEK(L)TH
EN515
510 W$="2001"+X$+D$+STR$(W):GOSUB80:W$=S
TR$(W+4)+X$+D$+"0":GOSUB80:GOTO525
515 A=FNA(FNB(FNP(H))):Y=520:GOTO70
520 READP:W$=STR$(A)+X$+D$+STR$(W):GOSUB
80
525 W$="2004"+D$+STR$(W+5-PEEK(L)):GOSUB
80:W$=STR$(W)+X$+D$+STR$(P):GOSUB80
530 FORI=1TO2:W$=STR$(W+I)+D$+"0":GOSUB8
0:NEXT:W$=STR$(W+3)+D$+WR$:GOSUB80
535 Y=100:GOTO90

```

```

548 REM*****
549 REM      *** WYDATKI ***
550 K=FNP(H):IFK=0ORR=9THEN100
555 WR$="WYDATKI":GOSUB95:IFPEEK(L)=0THE
N570
560 A=FNA(700+PEEK(H)*2):Y=565:GOTO70
565 READL$,W1:A1=A+1
570 A=FNA(FNB(K)):Y=575:GOTO70
575 POKET,4:READL$,W,P:POKEE,K*2:PRINT:P
RINT$:W=W+WR:GOSUB55:A=A+1:W=W-WR
580 GOSUB75:IFPEEK(L)THENW=W1:A=A1:GOSUB
75
585 RESTORE:READL$,L$,W:A=2002:GOSUB75:Y
=100:GOTO90

598 REM*****
599 REM      *** PLAN ***
600 K=FNP(H):IF(K=0ORR=9)ANDPEEK(L)<>2TH
EN100
605 WR$="PLAN":GOSUB95:IFPEEK(L)<>2THEN6
15
610 RESTORE:READL$,L$,L$,W:A=2003:GOSUB7
5:Y=100:GOTO90
615 A=FNA(FNB(K)):Y=620:GOTO70
620 READL$,I,W:A=A+2:GOSUB75:P=W+WR:W=I:
POKEE,K*2:PRINT:PRINT$:GOSUB55
625 Y=100:GOTO90

648 REM*****
649 REM      *** ZEROWANIE ***
650 K=FNP(H):IF(K=0ORR=9)ANDPEEK(L)<>2TH
EN100
655 POKEE,17:PRINT:PRINTA$Y$"ZEROWAC T/N
":INPUTL$:IFL$="T"THEN670
660 IFL$="N"THEN100
665 POKEE,17:PRINT:PRINTY$:LEFT$(S$,40):
:GOTO655
670 ONPEEK(L)+1GOTO735,740,675
675 A=2001:Y=680:GOTO96
680 RESTORE:READL$,A
685 IFA=0ORZTHEN745
690 B=740:GOSUB40:Y=695:GOTO96
695 A=FNA(740):Y=700:GOTO70
700 READA,L$,L$,L$,P:IFP=0THEN730
705 A=P:B=742:GOSUB40:Y=710:GOTO96
710 A=FNA(742):Y=715:GOTO70
715 READP:IFPTHEN705
720 A=FNA(740):Y=725:GOTO70
725 READA
730 IFPEEK(L)=0THENZ=1:GOTO685
731 GOTO685
735 Z=0:A=FNA(FNB(PEEK(H))):GOTO685
740 A=FNA(FNB(PEEK(681))):Y=750:GOTO96
745 IFPEEK(L)=2THEN100
750 POKET,4:POKEE,FNP(H)*2:PRINT:PRINTLE
FT$(S$,40)Z$Z$:W=0:P=0:GOSUB55:GOTO100

798 REM*****
799 REM      *** BUDZET ***
800 RESTORE:READL$,L$,W,P:POKEE,4:PRINT:
PRINTS$S$
805 POKEE,4:PRINT:PRINT:PRINTTAB(10)"PLA
N....."A$P
810 PRINTTAB(10)"WYDATKI..."A$W:PRINTTAB
(10)"BILANS..."A$P-W:GOTO100

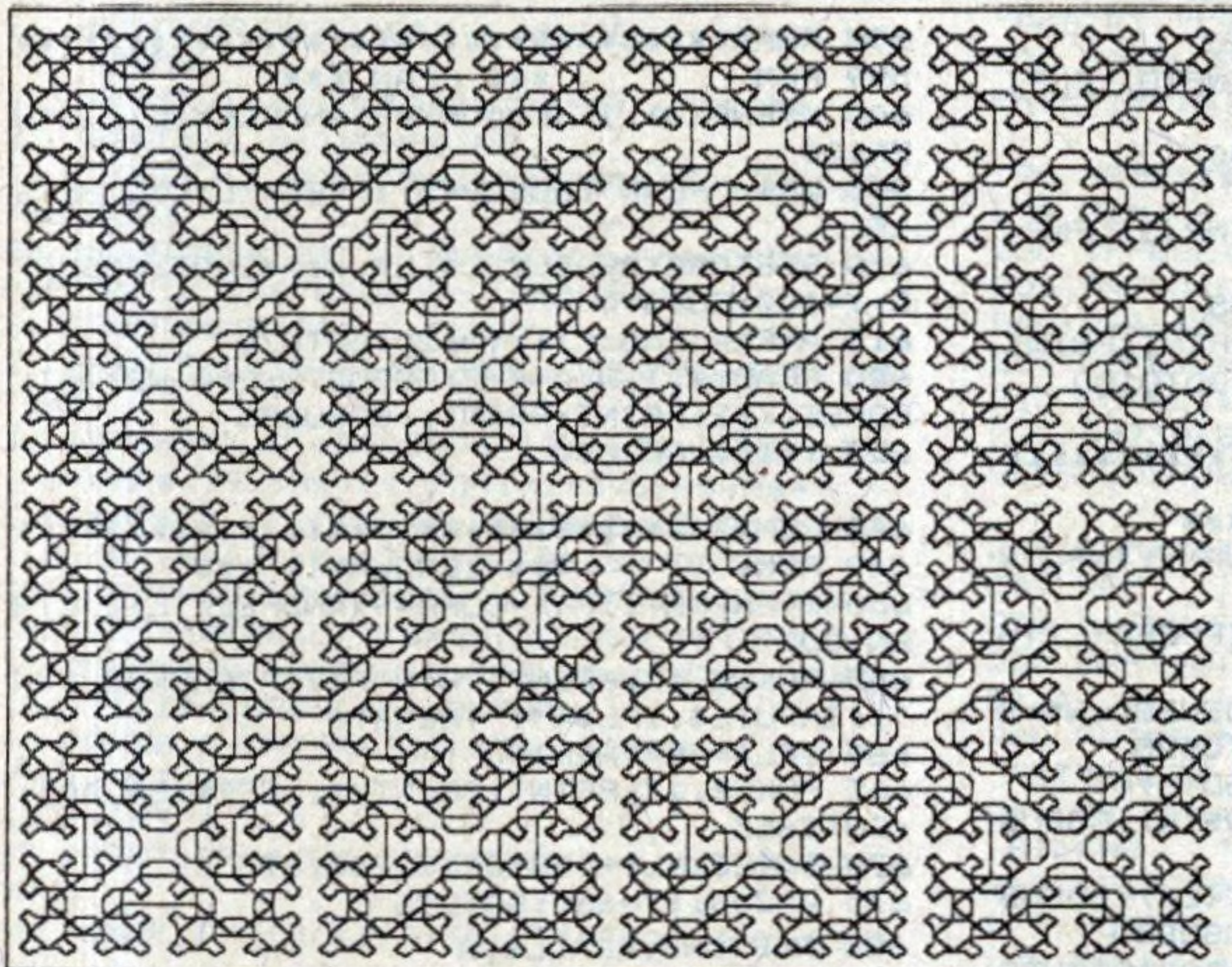
1998 REM*****
1999 REM      *** BAZA DANYCH ***
2000 DATA"MM.DD"
2001 RETURN:DATA0
2002 DATA0
2003 DATA0
2004 DATA2005

```

S. WASIELEWSKI

Krzywe Sierpińskiego

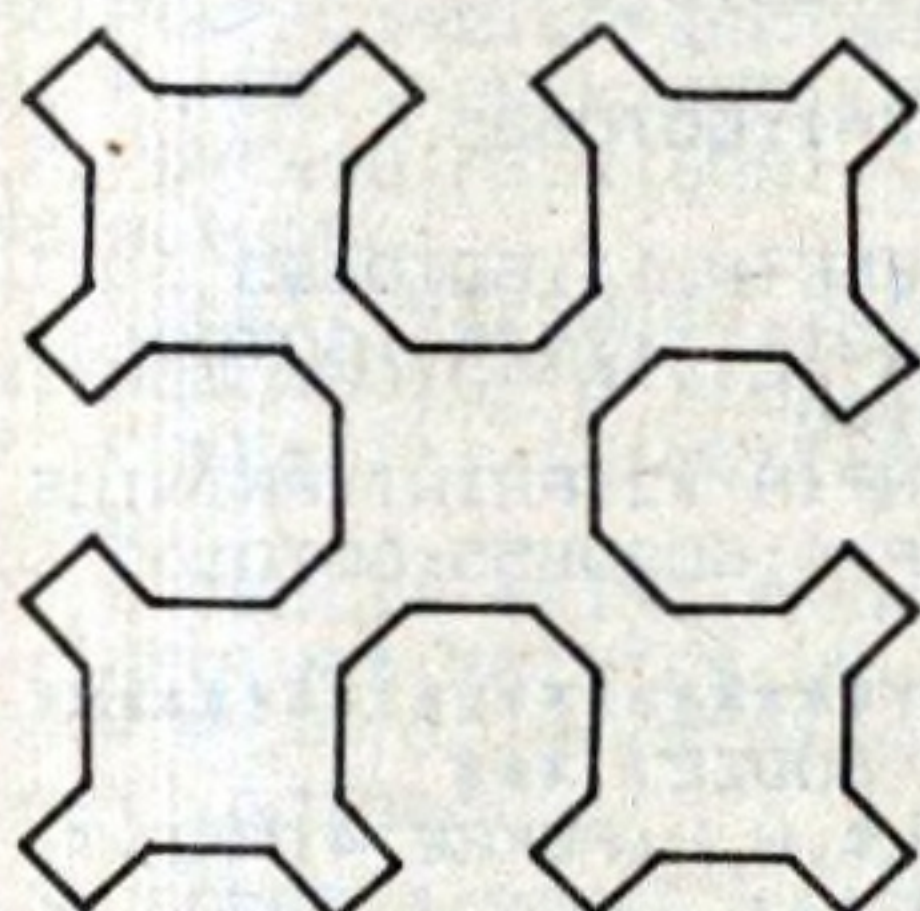
Efektowne układy graficzne można uzyskać, nakładając na siebie krzywe o regularnych kształtach.



Interesująca jest sytuacja, gdy kształt określonej krzywej K_i (krzywa rzędu i) można otrzymać ze złożenia odpowiedniej ilości krzywych K_{i-1} (o rząd niższych). Na przykład krzywą S_2

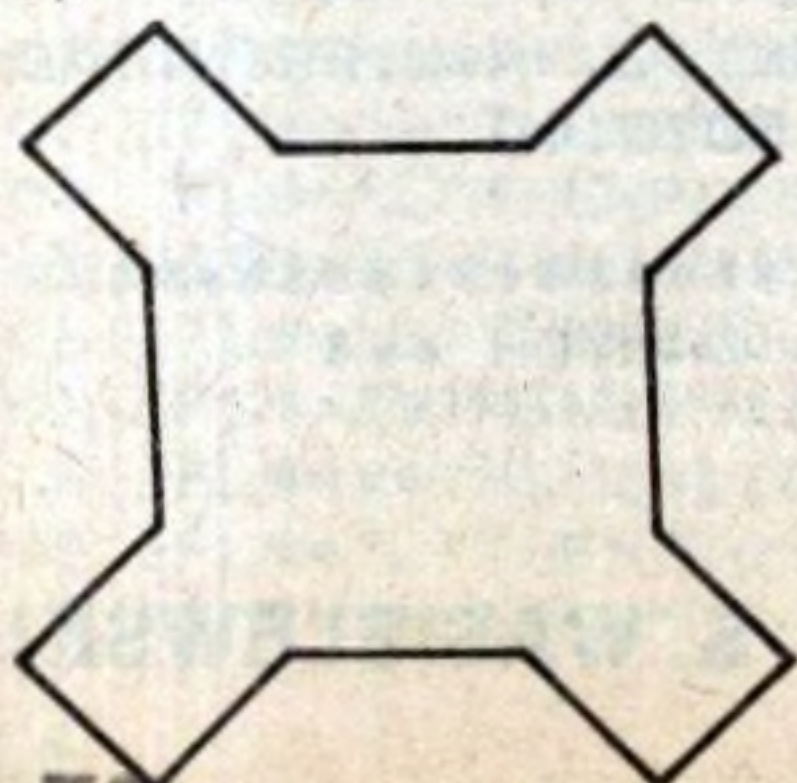
Krzywe S_2 i S_1 nazywane są krzywymi Sierpińskiego rzędu S_i . Obraz krzywej, której

częścią jest ta sama krzywa, nazywa się obrazem rekurencyjnym. Wykorzystując algorytm rekurencji, zamieszczony w książce Niklaus Wirth „ALGORYTMY + STRUKTURY DANYCH = PROGRAMY”, zapisany został w języku PASCAL program KRZYWE SIERPIŃSKIEGO.



S_2

można uzyskać z nałożenia na siebie krzywych S_1 .



S_1

```

(Program kresli krzywe Sierpińskiego rzędu 1 - 5)
{$a-}
{*****}
{ STALE I ZMIENNE PROGRAMU }
{*****}
CONST getnode:byte =2;
      tagflag:boolean =false;
      tagadr:integer =0;
      h0=256;
VAR
  i,h,x,y,x0,y0,n:integer;
  z,rz:char;
  o:boolean;
  warray [1..5] of
    record
      px,py,dl:integer;
    end;
{*****}
{ PROCEDURY GRAFICZNE }
{*****}
PROCEDURE initgrafik;
BEGIN
  write($1b,'0',$1b,'y');
END;

```

```

PROCEDURE leavegrafik;
BEGIN
  write($1b,'1',$1b,'x');
END;

PROCEDURE gotoxy(x,y:integer);
BEGIN
  write($1b,'Y',chr(y+31),chr(x+31));
END;

PROCEDURE sound(g:byte;h,i:integer;j,
                k,l,m:byte);
VAR sd:array[0..8] of byte;
BEGIN
  sd[0] := g;
  move(h,sd[3],2);
  move(i,sd[7],2);
  sd[1] := k;
  sd[2] := l;
  sd[5] := m;
  sd[6] := j;
  inline($21/sd/$cd/$5a/$fc/$aa/$bc);
END;

PROCEDURE draw(x,y:integer);
BEGIN
  inline($2A/x/$EB/$2A/y/$cd/$5a/$fc/$f9/$BB);
END;

PROCEDURE draw(x,y:integer);
BEGIN
  inline($2A/x/$EB/$2A/y/$cd/$5a/$fc/$f6/$BB);
END;

PROCEDURE plot(x,y:integer);
BEGIN
  inline($2A/x/$EB/$2A/y/$cd/$5a/$fc/$EA/$BB);
END;

PROCEDURE grafwindow(x1,y1,x2,y2:integer);
BEGIN
  inline($2A/x2/$EB/$2A/x1/$cd/$5a/$fc/$CF/$BB/$2A/y2/$EB/$2A/y1/$cd/$5a/$fc/$D2/$BB);
END;

PROCEDURE grafpen(fs:byte);
BEGIN
  inline($3A/fs/$cd/$5a/$fc/$DE/$BB);
END;

PROCEDURE grafpaper(fs:byte);
BEGIN
  inline($3A/fs/$cd/$5a/$fc/$E4/$BB);
END;

PROCEDURE clg(fs:byte);
BEGIN
  inline($cd/$5a/$fc/$DB/$BB);
  grafpaper(fs);
END;

PROCEDURE ink(fs,f1,f2:byte);
BEGIN
  inline($3a/f1/$47/$3a/f2/$4f/$3a/fs/$cd/$5a/$fc/$32/$bc);
END;

PROCEDURE paper(fs:byte);
BEGIN
  inline($3A/fs/$cd/$5a/$fc/$96/$BB);
END;

```

```

PROCEDURE pen(fs:byte);
BEGIN
  inline($3A/fs/$cd/$5a/$fc/$90/$BB);
END;

```

```

PROCEDURE mode(md:byte);
BEGIN
  if md in [0..2] then
  BEGIN
    getmode := md;
    inline($3a/md/$cd/$5a/$fc/$0e/$bc);
  END;
END;

```

```

PROCEDURE border(v1,v2:byte);
BEGIN
  inline($3a/v1/$47/$3a/v2/$4f/$cd/$5a/$fc/$38/$bc);
END;

```

```

PROCEDURE grafout(ch:char);
BEGIN
  inline($3A/ch/$cd/$5a/$fc/$FC/$BB);
END;

```

```

PROCEDURE tag;
BEGIN
  if not tagflag then
  BEGIN
    tagaddr := conoutptr;
    tagflag := true;
    conoutptr := addr(grafout);
  END;
END;

```

```

PROCEDURE tagoff;
BEGIN
  if tagflag then
  BEGIN
    conoutptr := tagaddr;
    tagflag := false;
  END;
END;

```

```

(*****
( PROCEDURE WYPRZEDZAJACE
)
(*****
PROCEDURE b(i:integer); FORWARD;
PROCEDURE c(i:integer); FORWARD;
PROCEDURE d(i:integer); FORWARD;

```

```

(*****
( PROCEDURE POMOCNICZE
)
(*****
PROCEDURE napis;
VAR nap:string[20];
j:integer;
BEGIN
  nap:='KRZYWE SIERPINSKIEBU';
  grafpen(1);
  tag; plot(10,200);
  FOR j:=1 TO 20 DO
    grafout(nap[j]);
  tagoff;
END;

```

```

PROCEDURE ramka;
BEGIN
  plot(w[n].px,w[n].py);
  drawr(w[n].dl,0); drawr(0,w[n].dl);
  drawr(-w[n].dl,0); drawr(0,-w[n].dl);
END;

```

```

PROCEDURE rzad;
BEGIN
  clg(6); napis;
  REPEAT

```

```

  clg(6); napis;
  n:=0;
  gotoxy(1,25);
  WRITE('RZAD(1-5)/ESC KONIEC');
  REPEAT
    sound(1,random(250),10,10,5,0,0);
    sound(2,random(200),10,10,10,0,0);
  UNTIL keypressed;
  READ(kbd,rz);
  n:=ORD(rz)-48;
  UNTIL ((n>0) and (n<6)) or (rz=#252);
END;

```

```

(*****
( PROCEDURE WYZNACZANIA ELEMENTOW
)
( KRZYWEJ RZEDU i
)
(*****
PROCEDURE a(i:integer);
BEGIN
  IF i>0 THEN
  BEGIN
    a(i-1);
    x:=x+h; y:=y-h; draw(x,y);
    sound(1,x#5,13,10,0,0,0);
    sound(2,y#10,7,10,10,0,0);
    b(i-1);
    x:=x+2#h; draw(x,y);
    d(i-1);
    x:=x+h; y:=y+h; draw(x,y);
    a(i-1);
  END;
END;

```

```

PROCEDURE b;
BEGIN
  IF i>0 THEN
  BEGIN
    b(i-1);
    x:=x-h; y:=y-h; draw(x,y);
    sound(1,x+20,13,10,0,0,0);
    sound(2,y+35,7,10,0,0,0);
    c(i-1);
    y:=y-2#h; draw(x,y);
    a(i-1);
    x:=x+h; y:=y-h; draw(x,y);
    b(i-1);
  END;
END;

```

```

PROCEDURE c;
BEGIN
  IF i>0 THEN
  BEGIN
    c(i-1);
    x:=x-h; y:=y+h; draw(x,y);
    sound(1,x+50,13,10,0,0,0);
    sound(2,y+75,7,10,0,0,0);
    d(i-1);
    x:=x-2#h; draw(x,y);
    b(i-1);
    x:=x-h; y:=y-h; draw(x,y);
    c(i-1);
  END;
END;

```

```

PROCEDURE d;
BEGIN
  IF i>0 THEN
  BEGIN
    d(i-1);
    x:=x+h; y:=y+h; draw(x,y);
    sound(1,x+60,13,10,0,0,0);
    sound(2,y+90,7,10,0,0,0);
    a(i-1);
    y:=y+2#h; draw(x,y);
    c(i-1);
    x:=x-h; y:=y+h; draw(x,y);

```

```

  d(i-1);
  END;
END;
(*****
( PROCEDURE INICJACJI
)
(*****
PROCEDURE inic1;
BEGIN
  mode(0);
  ink(0,1,1);
  ink(1,26,26);
  ink(2,18,18);
  ink(3,6,6);
  ink(4,14,14);
  ink(5,10,10);
  ink(6,0,0);
  grafwindow(0,16,640,400);
  paper(1); pen(3);
  clg(6);
END;

```

```

PROCEDURE inicdanych;
BEGIN
  inic1;
  w[1].px:=225;
  w[1].py:=125;
  w[1].dl:=210;
  w[2].px:=210;
  w[2].py:=110;
  w[2].dl:=240;
  w[3].px:=202;
  w[3].py:=102;
  w[3].dl:=254;
  w[4].px:=199;
  w[4].py:=98;
  w[4].dl:=262;
  w[5].px:=196;
  w[5].py:=96;
  w[5].dl:=266;
END;

```

```

PROCEDURE pocz;
VAR op:char;
BEGIN
  mode(1);
  ink(1,26,26);
  ink(2,6,6);
  paper(1); pen(2);
  border(9,9);
  REPEAT
    gotoxy(1,10);
    WRITE(' PROGRAM KRZYWE ');
    WRITELN(' SIERPINSKIEGO ');
    WRITELN;
    WRITE(' CZY CHCESZ AKCEPTOWAC ');
    WRITELN(' -WCISKAJAC DOWOLNY ');
    WRITE(' KLAWISZ- POSTAC KRZYWEJ ');
    WRITELN(' KOLEJNEGO RZEDU ');
    WRITE(' (t/n)? ');
    REPEAT
      sound(1,20+random(150),15,15,10,0,0);
      sound(2,30+random(200),12,12,20,0,0);
    UNTIL keypressed;
    READ(kbd,op);
  UNTIL (op='t') or (op='T') or
    (op='n') or (op='N');
  IF (op='t') or (op='T')
  THEN o:=true
  ELSE o:=false;
END;

```

```

(*****
( MODUL STERUJACY
)
(*****
BEGIN
  initgrafik;
  pocz;
  inicdanych;

```

```

REPEAT
  rzad; clg(6);
  IF rz<>#252 THEN
  BEGIN
    IF n=5 THEN
    BEGIN
      mode(2);
      grafwindow(0,16,640,400);
      grafpen(1); pen(0);
    END
  ELSE
    grafpen(3);
  ramka;
  i:=0; h:=h0 div 4;
  x0:=28h+200;
  y0:=38h+100;
  REPEAT
    i:=i+1; x0:=x0-h; h:=h DIV 2;
    y0:=y0+h;
    x:=x0; y:=y0;
    plot(x,y);
    IF n>4 THEN
      grafpen(1)

```

```

ELSE
  grafpen(i);
  a(i); x:=x+h;
  y:=y-h; draw(x,y);
  sound(1,x#i,13,20,0,0,0);
  sound(2,y#i,7,20,0,0,0);
  b(i); x:=x-h;
  y:=y-h; draw(x,y);
  sound(1,x+5#i,13,20,0,0,0);
  sound(2,y+5#i,7,20,0,0,0);
  c(i); x:=x-h;
  y:=y+h; draw(x,y);
  sound(1,x+10#i,13,20,0,0,0);
  sound(2,y+10#i,7,20,0,0,0);
  d(i); x:=x+h;
  y:=y+h; draw(x,y);
  sound(1,x+15#i,13,20,0,0,0);
  sound(2,y+15#i,7,20,0,0,0);
  IF o THEN
  BEGIN
    gotoxy(1,25);
    WRITE('RZAD',i:2,' WCISNIJ KL.');
```

```

  sound(2,15+i#20,20,15,10,0,0);
  sound(2,100+i#20,15,12,20,0,0);
  UNTIL keypressed;
  READ(kbd,z);

  END;
  UNTIL i=n;
  gotoxy(1,25);
  WRITE('WCISNIJ KLAWISZ ');
  REPEAT
    sound(1,55,5,30,20,0,0);
    sound(1,200,5,30,20,0,0);

  UNTIL keypressed;
  READ(kbd,z);
  END;
  IF n=5 THEN inic1;
  UNTIL rz=#252;
  mode(2); grafwindow(1,1,1,1);
  paper(0); pen(1);
  gotoxy(1,1);
  leavegrafik
  END.
```

AMSTRAD

Mieczysław SKONIECZNY
Robert ZUGEHOR

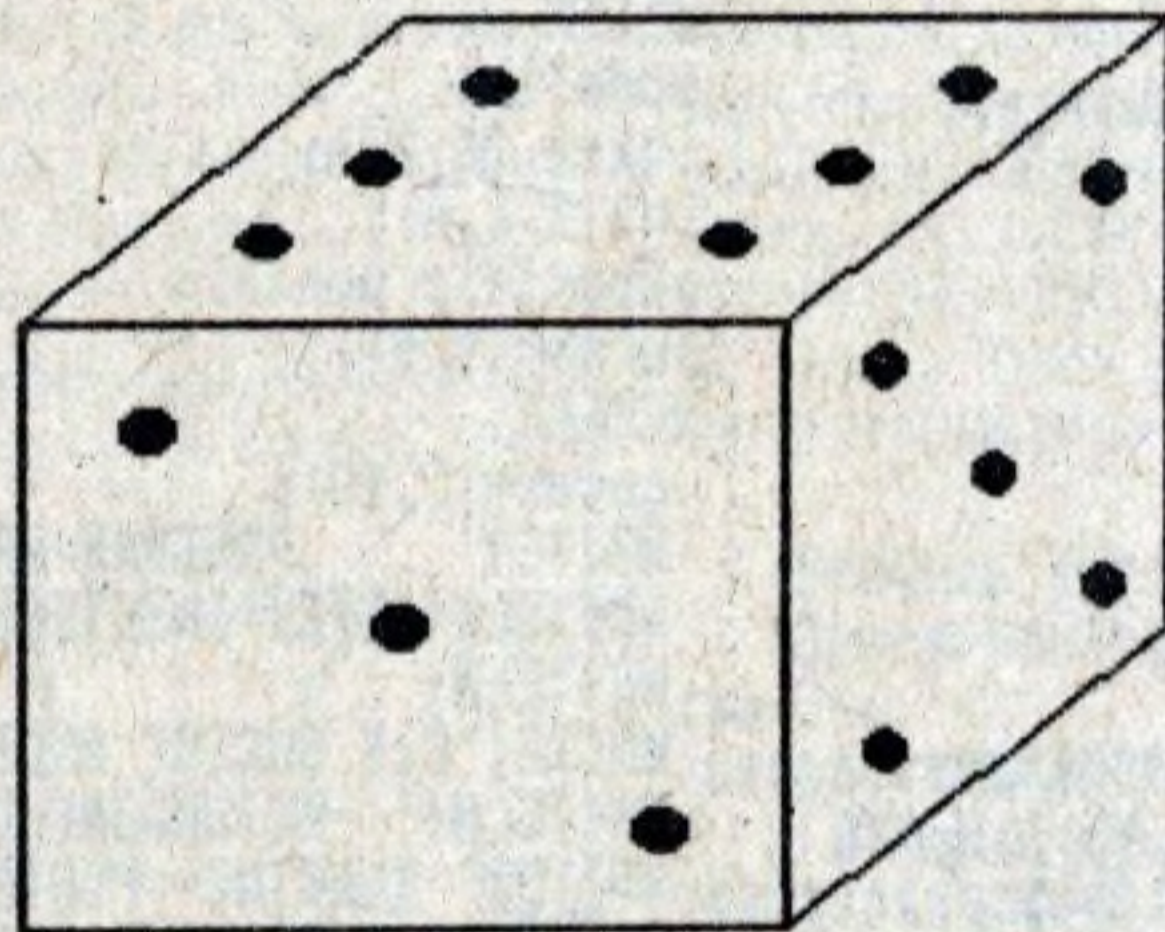
Kostka do gry

Ten program jest przykładem generowania liczb pseudolosowych przy pomocy funkcji RND. Ta instrukcja daje jako wartość wywoływanej funkcji liczbę rzeczywistą z przedziału <0,1>. Aby otrzymać liczby całkowite z ustalonego przedziału należy dodatkowo zastosować inną funkcję: INT. Wtedy zmienna całkowita y o wartości:

$$y = a + \text{INT}(\text{RND} * b)$$

należy do przedziału <a, b+a-1>.

W prezentowanym programie wykorzystano w/w własności generując liczbę całkowitą z przedziału <1,6>. Odpowiada ona liczbie wyrzuczonych oczek przy jednokrotnym rzucie kostką do gry. Obraz kostki widoczny na ekranie mikrokomputera pokazuje poniższy rysunek:



WYLOSOWANO 3

NACISNIJ DOWOLNY KLAWISZ

Po zainicjowaniu pracy programu naciśnięcie dowolnego klawisza powoduje szybką zmianę zawartości 3 widocznych ścianek kostki (odpowiednik toczenia się kostki po rzucie) i ustalenie wylosowanej ilości oczek na ścianie czołowej. Użytkownik ma nieograniczoną ilość rzutów.

Warto zauważyć, że funkcję RND wykorzystano także do ustalenia ilości prezentowanych sekwencji w czasie „toczenia” — ich liczba wynosi:

$$x = 10 + \text{INT}(\text{RND} * 20)$$

a więc jest liczbą całkowitą z przedziału <10,29>. Losowo ustalona wartość zmiennej OCZKO wpływa także na losowe generowanie dźwięku w instrukcji SOUND.

Szczególnie polecamy stosowanie tego programu do wszelkich gier i zabaw, w których niezbędne jest używanie kostki, jak również do wspomaganiania zajęć z matematyki dotyczących przeprowadzenia doświadczeń losowych z kostką.

Niezbędne do poprawnego działania tego programu na mikrokomputerze AMSTRAD CPC 6128 jest wcześniejsze wprowadzenie programu BANKMAN z dyskietki systemowej.

```

10 MODE 1:INK 0,2:INK 1,6:INK 2,26:INK 3,24: BORDER 26:PAPER 2:CLS
20 GOSUB 10000
30 CLS:a=150:b=64
35 GRAPHICS PEN 3
40 PLOT a,b
50 DRAWR 224,0:DRAWR 0,214:DRAWR -224,0:DRAWR 0,-214
60 PLOT a,b+214:DRAWR 112,107:DRAWR 224,0:DRAWR 0,-214:DRAWR -112,-107:PLOT a+224,b+214:DRAWR 112,107
65 MOVE a+10,b+10:FILL 0:MOVE a+10,b+220:FILL 0:MOVE a+230,b+100:FILL 0
70 !SCREENCOPY,2,1
80 LOCATE 1,25:PEN 0:PRINT"WCISNIJ DOWOLNY KLAWISZ";
90 A$=INKEY$
100 IF A$="" GOTO 90
110 x=10+INT(RND#20)
120 FOR C=1 TO x
130 oczko=1+INT(RND#6)
140 !SCREENCOPY,1,2
155 GRAPHICS PEN 3:PAPER 2
156 TAG
160 ON oczko GOSUB 1000,2000,3000,4000,5000,6000
170 SOUND 1,OCZKO#20,10,15
180 TAGOFF
190 NEXT
199 PEN 1:PAPER 2
200 LOCATE 13,23:PRINT"MYLOSOWANO ";OCZKO;
210 GOTO 80
1000 GOSUB 9000
1010 GOSUB 9150:GOSUB 9170:GOSUB 9190
1020 GOSUB 9320:GOSUB 9340
1999 RETURN
2000 GOSUB 9020
2010 GOSUB 9170:GOSUB 9190
2020 GOSUB 9320:GOSUB 9340:GOSUB 9360
2999 RETURN
3000 GOSUB 9000:GOSUB 9020
3010 GOSUB 9170:GOSUB 9190:GOSUB 9210
3020 GOSUB 9300:GOSUB 9320:GOSUB 9340
3999 RETURN
4000 GOSUB 9020:GOSUB 9060
4010 GOSUB 9170
4020 GOSUB 9300
4999 RETURN
5000 GOSUB 9000:GOSUB 9020:GOSUB 9060
5010 GOSUB 9150
5020 GOSUB 9300:GOSUB 9320
5999 RETURN
6000 GOSUB 9020:GOSUB 9060:GOSUB 9120
6010 GOSUB 9150:GOSUB 9170
6020 GOSUB 9320
6999 RETURN
9000 MOVE a+104,b+115:PRINT CHR$(241);
9010 RETURN
9020 MOVE a+30,b+185:PRINT CHR$(241);:MOVE a+180,b+43:PRINT CHR$(241);
9030 RETURN
9060 MOVE a+30,b+43:PRINT CHR$(241);:MOVE a+180,b+185:PRINT CHR$(241);
9070 RETURN
9120 MOVE a+30,b+115:PRINT CHR$(241);:MOVE a+180,b+115:PRINT CHR$(241);
9140 RETURN
9150 MOVE a+160,b+278:PRINT CHR$(243);
9160 RETURN
9170 MOVE a+130,b+305:PRINT CHR$(243);:MOVE a+200,b+250:PRINT CHR$(243);
9180 RETURN
9190 MOVE a+64,b+250:PRINT CHR$(243);:MOVE a+270,b+305:PRINT CHR$(243);
9200 RETURN
9210 MOVE a+97,b+276:PRINT CHR$(243);:MOVE a+234,b+276:PRINT CHR$(243);
9220 RETURN
9300 MOVE a+278,b+168:PRINT CHR$(242);
9310 RETURN
9320 MOVE a+246,b+206:PRINT CHR$(242);:MOVE a+310,b+128:PRINT CHR$(242);
9330 RETURN
9340 MOVE a+246,b+70:PRINT CHR$(242);:MOVE a+310,b+270:PRINT CHR$(242);
9350 RETURN
9360 MOVE a+246,b+140:PRINT CHR$(242);:MOVE a+310,b+200:PRINT CHR$(242);
9370 RETURN
10000 SYMBOL AFTER 240
10010 SYMBOL 241,60,126,255,255,255,255,126,60
10020 SYMBOL 242,24,60,126,126,126,126,60,24
10030 SYMBOL 243,0,60,126,255,255,126,60
10040 RETURN
```

INFORMATYCZNY SŁOWNIK ANGIELSKO-POLSKI

DATA BANK - bank danych, baza danych,
 DATA BASE - baza danych,
 DATABASE - patrz: DATA BASE,
 DATA BASE ADMINISTRATOR - administrator bazy danych,
 DATA BASE MANAGEMENT SYSTEM - system zarządzania bazą danych,
 DATA BIT - bit informacyjny,
 DATA BLOCK - blok danych,
 DATA BUS - szyna danych,
 DATA BUS ENABLE - blokada szyny danych,
 DATA CAPTURE - porcja danych,
 DATA CARRIER - nośnik danych,
 DATA CHAINING - wiązania łańcuchowe danych,
 DATA CHANNEL - kanał transmisji danych, kanał informacyjny,
 DATA-COLLECTING UNIT - urządzenie gromadzenia danych,
 DATA COLLECTION - gromadzenie danych, zbieranie danych,
 DATA COMPATIBILITY - wymiennosc danych,
 DATA COMPRESSION - zagęszczanie danych,
 DATA CONSISTENCY - zgodność (jednolitość) danych,
 DATA DENSITY - gęstość zapisu danych,
 DATA DESCRIPTION LANGUAGE (DDL) - język opisu danych,
 DATA DISPLAY FORMAT - postać danych na ekranie monitora,
 DATA DIVISION - część danych,
 DATA DRIVEN - także: wnioskowanie uprzedzające (w systemach ekspertowych),
 DATA ENCRYPTION STANDARD - system utajniania danych,
 DATA EXCHANGE CONTROL UNIT - jednostka sterująca wymianą danych, jednostka wymiany danych,
 DATA EXCHANGE CONTROLLER - jednostka sterująca wymianą danych, jednostka wymiany danych,
 DATA FLOW - przepływ danych,
 DATA FLOW CHART - schemat blokowy, schemat działania systemu,
 DATA FLOW DIAGRAM - patrz: DATA FLOW CHART,
 DATA FORMAT - układ danych, struktura danych,
 DATA GATHERING - rejestracja danych,
 DATA-GATHERING UNIT - urządzenie gromadzenia danych,
 DATA GROUP - grupa danych,
 DATA HANDLING - obróbka danych,
 DATA HOLE - dziurka informacyjna (taśmy, karty),
 DATA HOLE TRACK - ścieżka informacyjna (taśmy dziurkowane),
 DATA-IN - dane wejściowe,
 DATA INPUT - wprowadzanie danych,
 DATA INTERFACE UNIT - złącze transmisji danych, sprzęt transmisji danych, interfejs transmisji danych,
 DATA INTERLOCK - blokada dostępu do danych,
 DATA ITEMS - elementy danych,
 DATA KEYS - klawisze danych,
 DATA LINE - przewód danych,
 DATA LINK - kanał przesyłający dane, kanał informacyjny,
 DATA LINK CONTROL - sterowanie łączem na poziomie danych,
 DATA LOGGER - rejestrator danych,
 DATA MANAGEMENT - kierowanie przepływem danych, zarządzanie danymi,
 DATA MANIPULATION - operowanie danymi,
 DATA MANIPULATION LANGUAGE (DML) - język manipulowania danymi, DML,
 DATA MEDIUM - nośnik danych,
 DATA NAME - nazwa danych,
 DATA ORDERING - porządkowanie danych,
 DATA ORGANIZATION - organizacja danych,
 DATA-OUT - dane wyjściowe,
 DATA OUTPUT - wyprowadzanie danych,
 DATA PATH - ścieżka danych,
 DATA PLATE - tabliczka znamionowa,
 DATA POINTER (DP) - wskaźnik danych,
 DATA PREPARATION - przygotowywanie danych,
 DATA PROCESSING - przetwarzanie danych,
 DATA PROCESSING CENTRE - ośrodek przetwarzania danych,
 DATA PROCESSING COMPUTER - komputer do przetwarzania danych,
 DATA PROCESSING ENVIRONMENT - środowisko przetwarzania danych,
 DATA PROCESSING SYSTEM - system do przetwarzania danych,
 DATA PROCESSING TERMINAL - urządzenie końcowe, terminal, końcówka,
 DATA PROCESSOR - procesor, jednostka centralna,
 DATA PROTECTION - zabezpieczanie danych,
 DATA PURIFICATION - usuwanie błędnych elementów ze zbioru danych,
 DATA RATE - szybkość transmisji (przesyłania) danych,
 DATA RECORD - zapis (rekord) danych,
 DATA REDUCTION - kumulowanie danych,
 DATA RETRIEVAL - wyszukiwanie danych,

DATA SEARCH - przeszukiwanie danych,
 DATA SEARCH DEVICE - urządzenie wyszukiwania danych,
 DATA SEGMENT (DS) - segment danych,
 DATA SET - zbiór danych, także: urządzenie do transmisji danych,
 DATA SINK - ujście danych,
 DATA STATION - stacja abonencka, stacja teledacyjna,
 DATA STORAGE - pamięć danych,
 DATA SYSTEMS ENGINEERING - środki techniczne przetwarzania danych,
 DATA TERMINAL EQUIPMENT (DTE) - urządzenie końcowe transmisji danych,
 DATA TRANSFER - przenoszenie danych,
 DATA TRANSMISSION - transmisja danych,
 DATA TRANSMISSION DEVICE - urządzenie transmisji danych (UTD),
 DATA TRANSMISSION FACILITIES - urządzenia transmisji danych,
 DATA TRANSMISSION LINE - łącze transmisji danych, łącze przesyłowe,
 DATA TRANSMISSION PROGRAM - program transmisji danych,
 DATA TRANSMISSION RATE - szybkość transmisji, szybkość przesyłania (danych),
 DATA TRANSMISSION RELIABILITY - niezawodność transmisji, niezawodność przesyłania,
 DATA TRANSMISSION SPEED - patrz: DATA TRANSMISSION RATE,
 DATA TRANSMITTING CHANNEL - kanał transmisji danych,
 DATA UPDATING - aktualizacja danych,
 DATA WORD - słowo danych,
 DATE - data,
 DATEL SERVICE - usługi transmisji danych, usługi teledacyjne,
 DATER - datownik,
 DATUM - punkt wyjściowy (początkowy), także: dana wielkość, wielkość założona, element danych,
 DATUM LEVEL - poziom odniesienia,
 DATUM POINT - punkt odniesienia,
 DBMS - patrz: DATA BASE MANAGEMENT SYSTEM,
 DDL - patrz: DATA DESCRIPTION LANGUAGE,
 DE-ALLOCATION - wymazywanie (zbioru, obszaru zbioru),
 DEAD END - "ślepa uliczka" (na schemacie blokowym),
 DEADLINE DATE - termin prekluzyjny, nieprzekraczalny,
 DEADLOCK - blokada systemu,
 DEALLOCATION MEMORY - zwolnienie miejsca pamięci,
 DEBLOCKING - odblokowanie,
 DEBUG - usuwanie błędów z programu, uruchamianie,
 DEBUG DECK - pakiet testowy,
 DEBUGGING - "odpluskwanie", proces usuwania błędów z programu,
 DEBUG TIME - czas związany z wyszukiwaniem i usuwaniem błędów w programie,
 DECAY - rozpad, zanik,
 DECELERATE - zwalniać, zmniejszać np. prędkość ruchu taśmy magnetycznej,
 DECIDE - decydować,
 DECIMAL - cyfra dziesiętna, także: ułamek dziesiętny,
 DECIMAL CARRY - przeniesienie dziesiętne,
 DECIMAL CLASSIFICATION - klasyfikacja dziesiętna,
 DECIMAL-CODED DIGIT - cyfra dziesiętna,
 DECIMAL COMPUTER - komputer dziesiętny, maszyna dziesiętna,
 DECIMAL COUNTER - licznik dziesiętny,
 DECIMAL DIGIT - dit (dziesiątkowa jednostka informacji), cyfra dziesiętna,
 DECIMAL FRACTION - ułamek dziesiętny,
 DECIMAL NOTATION - zapis dziesiętny,
 DECIMAL NUMBER - liczba dziesiętna,
 DECIMAL NUMERATION SYSTEM - dziesiętny system liczbowy,
 DECIMAL PLACE - miejsce dziesiętne, przecinek dziesiętny,
 DECIMAL POINT - patrz: DECIMAL PLACE,
 DECIMAL SYSTEM - system dziesiętny, układ dziesiętny,
 DECIMAL-TO-BINARY CONVERSION - konwersja dziesiętno-dwójkowa,
 DECIMALIZATION - przejście na system dziesiętny,
 DECIPHER - rozszyfrowywać,
 DECISION - decyzja,
 DECISION BOX - figura na schemacie blokowym symbolizująca warunek,
 DECISION ELEMENT - element decyzyjny,
 DECISION FUNCTION - funkcja decyzyjna,
 DECISION INSTRUCTION - rozkaz decyzyjny,
 DECISION-MAKING - decydowanie,
 DECISION TABLE - tablica decyzyjna, tablica proceduralna,
 DECISION THEORY - teoria podejmowania decyzji, teoria decydowania,
 DECISION TREE - dendryt decyzyjny,
 DECIT - dit (dziesiątkowa jednostka informacji),
 DECK OF CARDS - plik kart dziurkowanych,
 DECLARATION - zgłoszenie, deklaracja,
 DECLARATIVE STATEMENT - deklaracja procedury,

DECODE - dekodować,
DECODER - dekodery,
DECODER MATRIX - tablica dekodowania,
DECODER NETWORK - patrz: DECODER MATRIX,
DECODIFICATION - patrz: DECODING,
DECODING - dekodowanie,
DECODING CIRCUIT - układ dekodujący,
DECOLLATOR - urządzenie do rozkładania (rozdzielania) druków wielowarstwowych,
DECOMPOSE - rozkładać (np. rozkaz na jego elementy składowe), dekomponować,
DECREASING FUNCTION - funkcja malejąca,
DECREMENT - zmniejszać, także: wielkość o jaką maleje zmienna,
DEDICATED SYSTEM - system (informatyczny) dedykowany,
DEDUCE - dedukować, wnioskować, wyprowadzać,
DEEPEN - pogłębiać,
DEFAULT - wartość standardowa, wartość zastępcza, wartość domyślna,
DEFAULT DEVICE - urządzenie standardowe, urządzenie domyślne,
DEFAULT VALUE - wartość domyślna,
DEFAULT - zaniedbanie,
DEFECT - błąd (np. kodowania), defekt, wada,
DEFERRED ENTRY - wejście asynchroniczne,
DEFINE - definiować, określać,
DEFINED NOTION - pojęcie zdefiniowane,
DEFINITION - definicja,
DEFLECTION - odchylenie, wychylenie (np. strzałki przyrządu),
DEFORNITY - zniekształcenie (np. karty),
DEGREE - stopień,
DEGREE OF ACCURACY - stopień dokładności,
DEGREE OF CONVERGENCE - stopień zbieżności,
DEGREE OF CORRELATION - stopień korelacji,
DEL - patrz: DELETE CHARACTER,
DELAY - opóźnienie, zwłoka,
DELAY LINE - element opóźniający,
DELETE - anulować, kasować, usuwać, wymazywać,
DELETE CHARACTER - znak wymazujący, kasujący,
DELETING A RECORD - wymazywanie (kasowanie) zapisu (rekordu),
DELIBERATELY - świadomie,
DELIMIT - oddzielać, separować, rozgraniczać,
DELIMITER - separator, ogranicznik,
DELIMITER CARD - karta rozgraniczająca,
DELINEATE - narysować, naszkicować,
DELVE - pogłębiać, zgłębiać (np. problem, zagadnienie),
DEMAND - żądanie, wymaganie, także: popyt,
DEMAND (DATA) PROCESSING - przetwarzanie danych swobodne,
DEMAND FETCHING - sprawozdanie na żądanie (sposób przemiennego wykorzystania pamięci),
DEMODULATION - demodulacja,
DEMODULATOR - demodulator,
DEMOUNTABLE - wymienny, możliwy do demontażu,
DENOMINATE NUMBER - liczba mianowana,
DENOMINATOR - mianownik,
DENOTATION - oznaczenie,
DENOTE - określać, oznaczać (np. wymiary urządzenia),
DENSITY - gęstość, zagęszczenie, upakowanie,
DENUMERABLE SET - zbiór przeliczalny,
DEPARTMENT - dział, wydział, departament,
DEPENDABILITY - niezawodność, pewność (działania),
DEPENDABLE - niezawodny,
DEPENDENCE - zależność,
DEPENDING ON ... - w zależności od ...,
DEPICT - opisywać coś, przedstawiać na rysunku,
DEPLOY - rozlewać (np. dane w pamięci maszyny),
DEPOSIT - odkładać, wkładać, także: przenosić (np. informacje z pamięci operacyjnej do zewnętrznej),
DEPRESS - naciskać, uruchamiać (np. klawisz-przycisk-maszyny),
DEPTH QUENING - obrazowanie głębi (tworzenie perspektywy),
DERIVE - wyprowadzać (np. wynik, wzór, zależność),
DES - patrz: DATA ENCRYPTION STANDARD,
DESCENDING - malejący (porządek), malejąco,
DESCENDING ORDER - porządek malejący,
DESCRAMBLER - deszyfrator,
DESCRIBE - opisywać,
DESCRIPTION - opis, deskrypt,
DESCRIPTIVE - opisowy,
DESCRIPTOR - deskryptor,
DESCRIPTOR PRIVILEGE LEVEL - deskrypcyjny poziom uprzywilejowania,
DESIGN - koncepcja, zamiar, projekt, schemat, także: konstruować, projektować, zamierzać,
DESIGN AUTOMATION - automatyzacja projektowania,
DESIGNATE - oznaczać, przedstawiać, wyznaczać,
DESIGNATOR - oznacznik,
DESK COMPUTER - komputer biurkowy, komputer przenośny,
DESKTOP COMPUTER - komputer stołowy,
DESTINATION INDEX REGISTER - rejestr indeksowy odbiorcy,

DESTINATION VARIABLE - zmienna przeznaczenia,
DESTROY - anulować, zniszczyć,
DESTRUCTIVE READ - odczyt niszczący, odczyt wymazujący,
DESTRUCTIVE READ-OUT STORE - pamięć o odczycie niszczącym, pamięć destrukcyjna,
DETACH - odejmować, odłączać,
DETACHABLE - odejmowany, odłączany,
DETAIL - drobiazgi, szczegóły, także: szczegółowy, wyszczególnić,
DETAIL BIT - bit szczegółowy,
DETAILED - szczegółowy,
DETECT - wykrywać,
DETECTABILITY - wybieralność (cecha elementu obrazu), wykrywalność,
DETECTABLE - wykrywalny,
DETECTABLE ELEMENT - element wykrywalny (np. przez pióro świetlne),
DETECTION - wykrywanie, wynajdywanie,
DETERIORATION - uszkodzenie,
DETERMINANT - wyznacznik,

DETERMINE - określać, ustalać, wyznaczać,
DETERMINE A RELATIONSHIP - określić zależność,
DETERMINE CONFORMITY - ustalać zgodność,
DEVELOP - budować, konstruować, także: opracowywać, przekształcać,
DEVELOP A METHOD - opracować metodę,
DEVELOPMENT - rozwój, rozbudowa, rozwinięcie, także: przekształcenie, adaptacja,
DEVELOPMENT SYSTEM - system wspomaganie, także: system uruchomieniowy,
DEVELOPMENT TIME - czas usunięcia błędów (np. w programie lub sprzęcie),
DEVIATE FROM - odbiegać od, odchylić się,
DEVIATION - odchylenie, także: błąd,
DEVICE - przyrząd, urządzenie,
DEVICE ALLOCATION - przydzielanie urządzeń, przyporządkowywanie urządzeń,
DEVICE CONTROL CHARACTER - znak sterujący urządzeniem,
DEVICE COORDINATES - układ współrzędnych urządzenia,
DEVICE DEPENDENCE - uzależnienie od jednostki peryferyjnej,
DEVICE DRIVER - program obsługi urządzenia,
DEVICE HANDLER - (standardowy) program obsługi urządzeń,
DEVICE SPACE - przestrzeń urządzenia graficznego, (obszar ograniczony pojemnością rejestrów pozycji X, Y),
DEVICE SPECIFIER - specyfikator urządzenia,
DEVICE STATUS - stan urządzenia,
DFA - patrz: DIGITAL FAULT ANALYSIS,
DI - patrz: DIGITAL INPUT,
DI (DESTINATION INDEX) - rejestr indeksowy odbiorcy,
DIAGNOSTIC - diagnostyczny, także: samoczynnie wysyłany przez komputer komunikat w celu zasygnalizowania i identyfikacji błędów,
DIAGNOSTIC CHECK - sprawdzenie sprzętu lub programu,
DIAGNOSTIC PROGRAM - program diagnostyczny,
DIAGNOSTICS ON-CHIP - układy diagnostyczne zawarte w układach scalonych,
DIAGONALLY - ukośnie,
DIAGRAM - rysunek, schemat, wykres,
DIAGRAM CHECK - kontrola schematu,
DIAGRAMMATIC - wykreslny, schematyczny,
DIAGRAMMATICALLY - schematycznie,
DIAGRAM PAPER - papier milimetry, papier milimetry,
DIAGRAM SYMBOL - symbol schematu,
DIAL COMMUNICATION LINE - łącze komutowane, łącze abonenckie,
DIC - patrz: DIGITAL INTEGRATED CIRCUIT,
DICHOTOMIZING SEARCH - wyszukiwanie dwudzielne, wyszukiwanie dychotomiczne,
DICTIONARY - słownik,
DID - patrz: DIGITAL INFORMATION DISPLAY,
DIE SORTING - sortowanie struktur półprzewodnikowych,
DIFFERENCE - różnica (np. przy odejmowaniu),
DIFFERENT LEVELS OF SOPHISTICATION - różne stopnie utrudnienia,
DIFFICULTY - trudność,
DIGIT - cyfra,
DIGITAL - cyfrowy,
DIGITAL ADDER - sumator,
DIGITAL CAPACITY - długość liczby,
DIGITAL-CIRCUIT ENGINEERING - patrz: DIGITAL ELECTRONICS,
DIGITAL COMPARATOR - komparator cyfrowy,
DIGITAL COMPUTER - komputer cyfrowy,
DIGITAL CONTROL - sterowanie cyfrowe,
DIGITAL CONVERTER - konwerter cyfrowy,
DIGITAL DATA - dane cyfrowe,
DIGITAL DESIGN - projektowanie układów logicznych,
DIGITAL DISPLAY - przedstawienie cyfrowe,
DIGITAL ELECTRONICS - technika cyfrowa, cyfronika,
DIGITAL FAULT ANALYSIS - cyfrowa analiza uszkodzeń,
DIGITAL INDICATOR - wskaźnik cyfrowy,

GIEŁDA POMYSŁÓW

Kredka

Jest to program graficzny, który może służyć do robienia rysunków, wykresów w zależności od umiejętności i cierpliwości użytkownika. Cierpliwości, bo poza jego powolnością, ma on i tę wadę, że jest czarno-biały. Poza tym sądzę, że ma resztę zalet, do których należy prostota w obsłudze. Rysowanie instrukcją PLOT odbywa się za pomocą kursorów (5,6,7,8) przy opuszczonej kredce lub odpowiednio R, T, U i Y przy kredce podniesionej (która jest w tym przypadku gumką). Ciągły odczyt współrzędnych x i y. Litera „S” można dokonać nagrania na taśmie, po zakończeniu którego obraz jest wczytywany w inny obszar pamięci komputera w celu

zabezpieczenia go przed uszkodzeniem w przypadku wadliwego nagrania. W celu ponownego nagrania można go wywołać przez RANDOMIZEUSR 30000. Program ma możliwość rysowania okręgów w wariacie „0 — normal”, lub usuwania w wariacie „1 — inverse”, albo rysowania białego okręgu na czarnym tle.

Promień podany jest w punktach tzw. pikselach. Po uruchomieniu programu mamy możliwość wgrania niedokończonego rysunku lub rozpoczęcia rysowania od początku. W obu przypadkach przed rysowaniem należy tylko podać współrzędne x i y, w których punkt ma się ukazać.

Zygmunt WIŚNIEWSKI

```
10000 CLEAR 30000: GO TO 10
13000 FOR PEEK 10000 TO 23296: POKE
FOR PEEK 10000 TO 23296: POKE
FOR PEEK 10000 TO 23296: POKE
NEXT n: NEXT f
READ d: POKE n,d
DATA 17,0,64,33,60,117,1,0,
27,17,176,201
CLS
PRINT PAPER 6; AT 0,0; "RYSUN
OSTAL NAGRANIE"; PRINT AT 2,0
WYKONAJ NAGRAWIE"; AT 3,0; "2-wcz
NAGRAWIE PRZEZ "; AT 4,2; "LO
PRINT AT 5,0; " W przypadku
dokonanego "; AT 9,0; "nagra
nie nalezy wyolac rysunek"; AT 1
0,0; "przez wyolac rysunek"; AT 1
1,0; "RANDOMIZEUSR 30000"
; AT 12,0; "wczyna na taśmie prze
z "; AT 13,0; "SAVE nazwa SCREEN$
; AT 13,0; "i dalej jak w punkci
e 10
PRINT AT 16,0; " Wywoływac
rysunek można wie-"; AT 17,0; "LO
KROTKIE";
STOP
PRINT AT 8,14; "L-Load"; AT 1
14; "D-Draw"; BRIGHT 1; AT 6,14;
AT 12,14; "D";
PRINT PAPER 6; AT 21,2; "Naci
odpowiedni klawisz"
IF INKEY$="L" THEN CLS : LO
AD "SCREEN$ : GO TO 35
IF INKEY$="d" THEN CLS : GO
TO 20
GO TO 15
PRINT BRIGHT 1; INVERSE 1; A
T 0,5; "Klawisze do sterowania"
PRINT AT 2,3; "kredka"; AT 2,
23; "kredka"
PRINT AT 3,1; "opuszczona"; A
T 3,20; "podniesiona"
PRINT BRIGHT 1; AT 6,5; "5"; A
T 6,6; "6"; AT 10,5; "7"; AT 12,5; "8"
; AT 16,5; "T"; AT 10,
25; "U"; AT 12,25; "I";
PRINT PAPER 6; AT 21,3; "Naci
dowolny klawisz"
PAUSE 0: CLS
INPUT "x="; x
IF x<0 OR x>255 THEN GO TO
35
INPUT "y="; y
IF y<0 OR y>175 THEN GO TO
45
PRINT #0; AT 1,0; "x="; x; AT 1
7; "y="; y; BRIGHT 1; AT 1,15; "S";
AT 1,23; "C"; BRIGHT 0; AT 1,16; "-
save"; AT 1,24; "-Circle"
PLOT x,y
PAUSE 0
PRINT #0; AT 1,2; " "; AT
1,9; "
```

```
80 IF INKEY$="5" THEN LET x=x-
1
90 IF INKEY$="8" THEN LET x=x+
1
100 IF INKEY$="6" THEN LET y=y-
1
110 IF INKEY$="7" THEN LET y=y+
1
120 IF x<0 THEN LET x=0
130 IF x>255 THEN LET x=255
140 IF y<0 THEN LET y=0
150 IF y>175 THEN LET y=175
160 IF INKEY$="r" THEN GO SUB 2
50
170 IF INKEY$="t" THEN GO SUB 3
50
180 IF INKEY$="u" THEN GO SUB 4
50
190 IF INKEY$="i" THEN GO SUB 5
50
200 IF INKEY$="s" THEN GO TO 80
0
210 IF INKEY$="c" THEN GO SUB 7
00
220 GO TO 55
250 LET x=x-1
255 IF x<0 THEN LET x=0
260 PLOT x,y: PLOT INVERSE 1;x+
1;y
265 RETURN
350 LET y=y-1
355 IF y<0 THEN LET y=0
360 PLOT x,y: PLOT INVERSE 1;x,
y+1
365 RETURN
450 LET y=y+1
455 IF y>175 THEN LET y=175
460 PLOT x,y: PLOT INVERSE 1;x,
y-1
465 RETURN
550 LET x=x+1
555 IF x>255 THEN LET x=255
560 PLOT x,y: PLOT INVERSE 1;x-
1;y
565 RETURN
600 INPUT "nazwa nagrania "; " "
; a$
610 SAVE a$SCREEN$
620 PRINT #0; FLASH 1; AT 1,0; "W
PISYWANIE RYSUNKU ZA RAMTOP"
630 GO TO 2
700 INPUT "circle : 1-inverse
0-normal "; c
701 IF c>1 THEN GO TO 700
702 INPUT "promien = "; p
710 IF p>175-y OR y-p<0 THEN GO
TO 702
720 IF p>255-x OR x-p<0 THEN GO
TO 702
740 CIRCLE INVERSE c;x,y,p
750 RETURN
```

KRZYŻÓWKA NR 3

DRZEWIE NIEMI WYSTĘP. PRZY WYSOKIEJ GORĄCZCE				AUTOR POWIĘŚCI „BARNY WALKI”	UBYTEK TOWARU SPOWOD. WYSCHN.	REP. ASER. Z GINCYWKAREN	PLAN REALIZ. FILMU	LITERA GRECKA					
ROZWIĘDŁA GAŁĄZ	D ²¹	TRARDY WĘDZONY SER OHICY	NIEJEMEN W JAKABIS	M KOBIECE NAKRYCIE GŁOWY	OPAL	K O K S	MAKTYCZOL MODALWOLCI	E					
RYBA Z RODZINY KARPIONOWATYCH	R	O	S	O	C	H	A ¹¹	PRZYPR. DO MARINAD	O C E T				
MIEJSKO- WOŚĆ NAD PILICĄ	L	E	S	Z	C	Z	Ł	ÓZDOLA CHOLIMKI D/BL. SPK	L A M E ² T A				
GRUBA BELKA STROP.	S	Z	C	Z	E	K	O	C	I ⁷ N Y				
POLANO	S	Z	C	Z	A ⁸	P	A	HAZAR- DOWE 21	O K ¹³ O				
ROŚNIE WA BAGNACH	O ¹⁴	C	Z	E	R	E	T	UŁAD OPTYCZNY SOCZEKAR	SIEDZI- BA PAPIEŻA	P ZUPA Z KRWI GESI			
ODWIERT	S	Z	Y	B	K	A ¹⁷	T	NAZWA MITY WJĘTY SOKOŁA	O	W	I ¹⁹	C	E
PUNKT SIECI MAGLACYJNEJ	E	P	E ⁵	R	GAT. GRUSZY	L	M	NA NIM FLAGA EBEN	M ¹⁰	A	S	Z	T
GEOP. SIECI MAGLACYJNEJ	E	L	A	B ¹	O	R	A	T	ŚWIĘTY BYK W STANOWYNYM EGIPCE	E	R	48	
DRZEWO LIŚCIA- STE	B	U	K	R	E	G	A ²³	T	Y	PLAZDA BAROKU DO PRZECIEGA- NIA KRÓB			
KISZONK ODZIĄGU KOPALNI	C	Z	M ²⁴	O	R	J	K	A	N	T			
AUGUST III NOCNY	C	Z	M ²⁴	O	R	J	K	A	N	T			
KISZONK ODZIĄGU KOPALNI	S	Z ²²	T	Y	G	A	R ¹²	T	A	P	I	R ¹⁶	
KARDA WŁÓCZA WANE KORON	A ²⁰	T	U	T	O	A	N ⁶	I	N	A			
LINCZ	S	A	M ⁴	O	S	A	D ¹⁵	N	S ³	A	K		

Litery z ponumerowanych kratek od 1 do 24 utworzą nazwy 6 komputerów, które wystarczy podać jako rozwiązanie zadania, przesyłając pod adresem redakcji na kartach pocztowych do końca kwietnia, naklejając kupon „IKS-a”. Wśród autorów prawidłowych odpowiedzi rozlosujemy bony pieniężne i nagrody książkowe.

11 | 15 | 10 | 15 | 120 | 124
B | E | S | M | E | N | T | A | C | M | A | R | K | O | D | R | I | A | R | I | A | D | Z | I | A | M

KUPON ►



MIKROCIEKAWOSTKI
IBM Personal System 2



„IKS” — dodatek „Żołnierza Wolności”. Redagują: Wiesław Cetera (kierownik zespołu), Ryszard Rogoń. Stali współpracownicy: Włodzimierz Gogolek, Krzysztof Mamcarz, Ireneusz Miernik, Janusz Miller, Jacek Szaniawski. Adres redakcji: 00-950 Warszawa ul. Grzybowska 77, telefon centrali 20-12-61 w. 486. Telex 313664. Rękopisów nie zamówionych redakcja nie zwraca i zastrzega sobie prawo do skrótów. Nakładem: Wydawnictwa „Czasopisma Wojskowe”, Warszawa ul. Grzybowska 77, Fotoskład i druk offsetowy — Wojskowe Zakłady Graficzne im. gen. dyw. A. Zawadzkiego. Nr zam. 1144. Nr ind. 361682.