

MOJE

Atari

CENA 8000 ZŁ

MAGAZYN UŻYTKOWNIKÓW KOMPUTERÓW ATARI • MAJ/CZERWIEC '91 (3/91)

WIELKI
KONKURS!

5

JAK KODOWAĆ?

TOMS 720

RAMBRANDT

ACTION! BEZ ACTION!

TT — WIZYTÓWKA ATARI

CZY KUPIĆ UŻYWANE ST

192 KB PAMIĘCI W 65XE

CO? W N U M E R Z E:

Drogi Czytelniku!

Otrzymujemy wiele listów i telefonów z prośbami o pomoc w korzystaniu z różnych programów — gier i użytków. Pomimo iż staramy się udzielać porad zarówno w sprawach sprzętowych, jak i programowych, to większość zwracających się do nas osób jest niezadowolona z uzyskanych odpowiedzi. Skąd to się bierze?

W takich sprawach zawsze rozpoczynamy od zadania pytania: „Gdzie został nabyty program?”. Jeżeli w odpowiedzi dowiadujemy się, że na giełdzie, od kolegi, w „wypożyczalni” lub w podobnym miejscu, to jedyną informacją jakiej udzielamy jest sugestia, aby po pomoc zwrócić się do wskazanego złodzieja. Choć nie ma jeszcze w naszym kraju odpowiedniego ustawodawstwa, to jednak rozprowadzanie programów bez zgody ich autorów (lub właścicieli praw autorskich) jest niczym innym jak zwykłą kradzieżą.

Podobna sytuacja występuje w dziedzinie ulepszeń sprzętowych. Rozwiązania opracowane w jednej firmie są następnie bez żenady instalowane w innych. Co gorsza, tacy „fachowcy” niejednokrotnie szczytą się swoimi osiągnięciami, a nawet je reklamują. To samo dotyczy także instrukcji i książek powielanych na kserokopiarkach, a często na dodatek nieudolnie tłumaczonych. Należy jednak uczciwie powiedzieć, że brak odpowiedniej ustawy nie zmienia moralnej kwalifikacji takich praktyk.

Z pewnością zauważyłeś, że z naszych łamów zniknęła większość reklam firm rozprowadzających oprogramowanie. Żądamy bowiem okazania zgody autora lub licencji firmy. Działanie takie nie jest powodowane chęcią szykanowania „przedsiębiorczych” złodziei, lecz dbałością o użytkowników komputerów. Nabywca legalnego programu może bowiem liczyć na pomoc firmy w przypadku kłopotów, czy to z obsługą, czy z zainstalowaniem programu lub modyfikacji sprzętowej. Bliżej temat ten był zaprezentowany w artykule „Jak wygląda prawdziwy program?” w poprzednim numerze „Mojego Atari”.

Z zadowoleniem muszę stwierdzić, że nasze działania przynoszą widoczne efekty — coraz więcej otrzymujemy pytań o to, gdzie można kupić program LEGALNIE. Coraz więcej naszych Czytelników rozumie, że taki program jest uprawdzie droższy, lecz sprawia mniej kłopotów i daje więcej pożytku. Niestety takich miejsc jest jeszcze bardzo mało.

W tym miejscu chciałbym coś zasugerować. Otóż na Zachodzie komputery 8-bitowe należą już do przeszłości. Myślę, że wiele firm za niewygórowaną opłatą lub za niewielki procent odstąpi prawa autorskie do programów, których i tak już nie sprzedają. Wystarczy więc trochę gotówki i można rozkręcać interes. Ze swej strony zapewniam, że takie inicjatywy będziemy popierać.

Wojciech ZIENTARA

MOJE Atari

Redaktor naczelny:

Wojciech Zientara

Sekretarz redakcji:

Tomasz Wiśniewski

Opracowanie graficzne:

Wanda Roszkowska

Redaktor techniczny:

Maria Radzimińska

Zdjęcia:

Leopold Dzikowski

Redakcja:

02-776 Warszawa, ul. Wasilkowskiego 7

tel. 643-18-40

Wydawca:

Spółdzielnia „Bajtek”

00-687 Warszawa, ul. Wspólna 61,

tel. 21-12-05

Skład i druk:

Przedsiębiorstwo Wydawniczo-

Poligraficzne „GRYF”

Ciechanów, ul. Sienkiewicza 51

Fotoskład:

Grażyna Kurzątkowska

Montaż offsetowy:

Grażyna Ostaszewska

Korekta:

Maria Krajewska

Teresa Rutkowska

Nr zlecenia: 86430

Nakład: 105 tys. egz.

Materiałów nie zamówionych nie

zwracamy.

Za treść ogłoszeń redakcja nie odpo-

wiada.

PROGRAMOWANIE

Action! bez Action! 15

Uruchomienie programu w Action! jest możliwe również bez Action!. Z tego artykułu dowiesz się, jakie trzeba w tym celu spełnić warunki.

Tylko dla zaawansowanych (2) 13

Drugi odcinek porad dla programistów — wbrew tytułowi nie tylko dla zaawansowanych. Tym razem ponownie o polskich literach.

Projektowanie programu ANTIC-a 12

Chcesz, aby Twój program miał ładną, wielokolorową planszę? Skorzystaj z programu, który umożliwia łatwe tworzenie programu ANTIC-a wykorzystującego wszystkie możliwości tego procesora.

GRY

Robot R-29 6

Prosta, lecz ciekawa gra — wędrówka robota przez labirynt. Możliwość samodzielnej rozbudowy.

TOP 12 20

Rozpoczynamy prezentację najlepszych i najnowszych gier przeznaczonych dla komputerów Atari, zarówno XL/XE, jak i ST.

Nowe i stare 24

Opisy gier „Rings of Medusa”, „Fred”, i „Indiana Jones — The Last Crusade”.

PROGRAMY UŻYTKOWE

Analiza stałoprądowa 8

Program wykonujący obliczenia elektrycznych obwodów prądu stałego.

SPRZĘT

Stacja dysków TOMS 720 10

Test nowej stacji dysków dla 8-bitowych Atari. Posiada ona dwie głowice i na jednej dyskietce mieści 720 KB informacji.

Systemy instalowane w stacjach dysków 14

Pierwszy z cyklu artykułów prezentujących modyfikacje rozszerzające możliwości stacji dysków do Atari XL/XE.

192 KB RAM w 65XE 22

Każdy komputer ma za mało pamięci. Posiadaczy 65XE z pewnością zainteresuje jego przeróbka na 130XE, a nawet więcej.

SZKÓŁKA

Dyskietka? To proste! 7

Podstawowym nośnikiem, na którym zapisuje się programy i dane dla większości komputerów domowych, jest dyskietka. Przeczytaj, jak jest zbudowana i jak działa.

Kodowanie (2) 19

Nasza gra uzyskuje już tytuł, instrukcję i podstawowy szkielet konstrukcji oraz kilka innych elementów.

INSTRUKCJE

RAMbrandt 9

Kolejny odcinek „Vademecum Atari” — instrukcja doskonałego programu graficznego na „małe” Atari.

DZIAŁ ST

Kupujemy używane ST 28

Krótki poradnik dla wszystkich, którzy planują zakup komputera Atari ST z „drugiej ręki”.

Touch-Up 29

Krótki opis rewelacyjnego programu graficznego na „duże” Atari rozpoczyna „Vademecum ST”.

PCDitto 29

W poprzednim numerze opisywaliśmy sprzętowe emulatory IBM na ST. Teraz kolej na redakcyjny test emulatora programowego.

Atari TT 32

Nowa jakość w rodzinie Atari — komputer w pełni 32-bitowy o ogromnych możliwościach.

INNE

Wielki konkurs 25

Wieści 3

Listy 4

Edytor Basica 5

Wpisywanie znaków specjalnych 5

UWAGA: W naszej redakcji (Warszawa, ul. Wasilkowskiego 7, wtorek — czwartek, godz. 10–14) można nabyć dyskietkę zawierającą wszystkie programy, których wydruki są opublikowane w tym numerze. Cena dyskietki wynosi 15 000 zł. Niestety z przyczyn technicznych nie rozpowszechniamy tych programów na kasetach magnetofonowych.



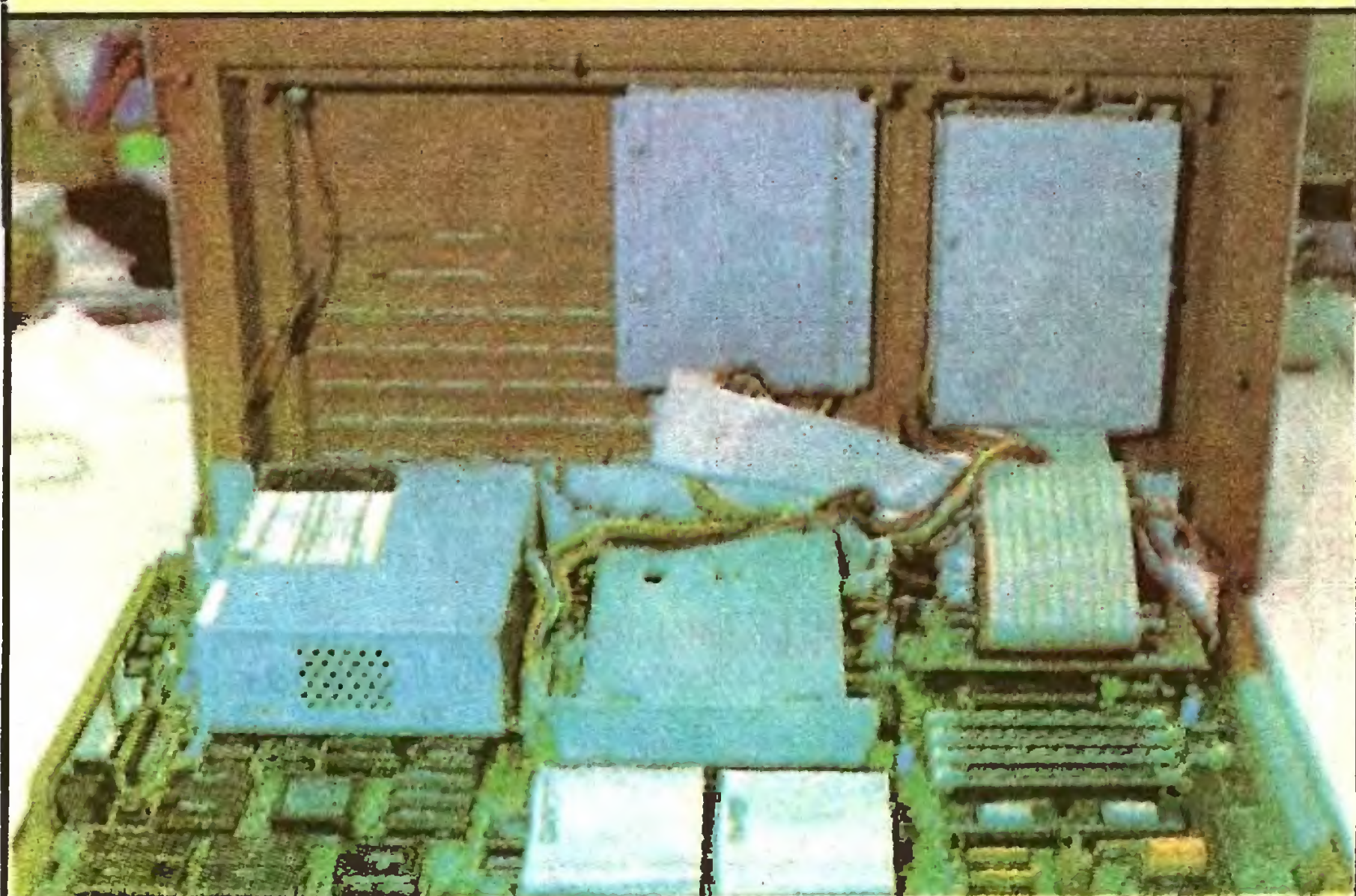
Mega STE z zewnątrz...

Premiera Mega STE

W grudniu 1990 roku firma Atari Deutschland GmbH zaprosiła dziennikarzy na konferencję prasową do Monachium. Uczestnicy imprezy mieli okazję obejrzeć nowy komputer z rodziny Atari, który wyglądem do złudzenia przypomina TT. Głównym bohaterem konferencji było Mega STE, następcą 1040 STE.

Obudowa komputera została zapożyczona z 32-bitowego Atari TT. W pierwszej chwili dziennikarze sądzili, że będzie prezentowana kolejna wersja TT. Ale podobieństwo do TT okazało się bardzo powierzchowne.

Mega STE jest zbudowane na bazie 16-bitowego procesora 68000, taktowanego z częstotliwością 16 MHz i wspomaganego 16 KB pamięci Cache. Komputer może mieć od 1 do 4 MB pamięci RAM, jednakże zdecydowano się rozpocząć dystrybucję od wersji STE/4. Zainstalowano w nim twardy dysk 48 MB i stację dysków elastycznych 720 KB. W standardowej konfiguracji znajduje się także interfejs umożliwiający przyłączenie komputera do sieci. (J.J)

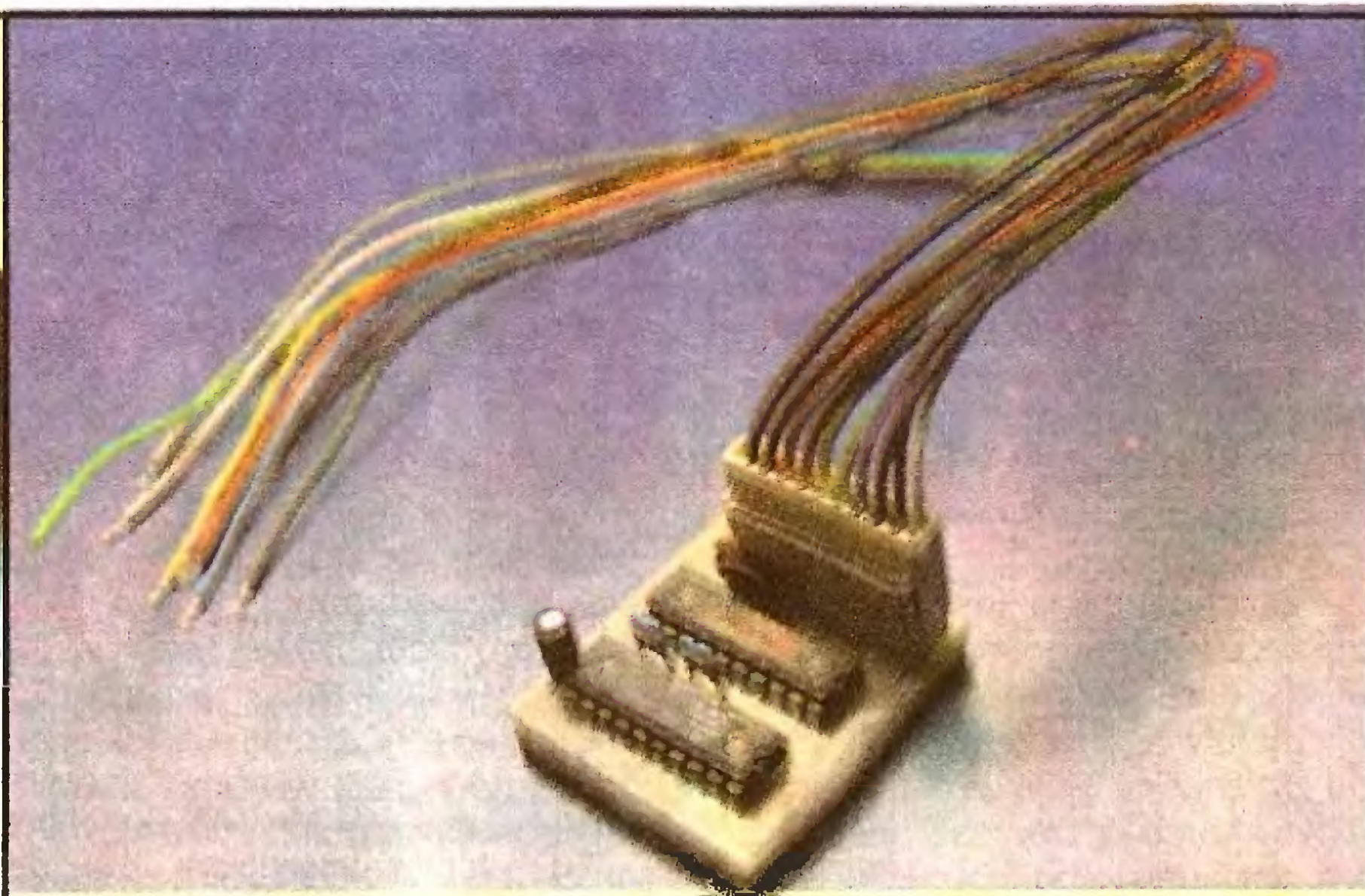


SuperCharger plus

Tempo, w jakim pojawiają się nowe rozwiązania emulatorów PC, jest oszałamiające. Producenci sprzętu prześcigają się w coraz wymyślniejszych pomysłach przeróbek Atari ST na PC.

W poprzednim numerze „Mojego Atari” pisaliśmy o emulatorze SuperCharger. Od pewnego czasu producent reklamuje jego kolejną, o wiele doskonalszą wersję SCplus.

Nowy SuperCharger plus/286, w odróżnieniu od poprzednika z procesorem NEC-V30, ma procesor Intel 80286, który pracuje z częstotliwością 12 MHz. Do najpotężniejszych atutów SCplus zalicza się własną pamięć RAM o pojemności od 1 do 4 MB. W jego wnętrzu można zainstalować dodatkowo dowolną kartę rozszerzającą PC AT, np. kartę graficzną VGA, interfejsy, kartę umożli-



Overscan to tylko dwie kości GAL i parę kabli.

OVERSCAN

Wielu użytkowników Atari marzy o poprawieniu graficznych możliwości swojego komputera. Nie każdy z nich wie, że rozdzielczość monitora współpracującego z ST można poprawić stosunkowo niewielkim kosztem. Efekt ten uzyskujemy dzięki układowi Overscan, którego podstawowymi elementami są dwie kości GAL (programowalne układy logiczne) oraz dyskietka z oprogramowaniem.

Każdy bardziej doświadczony majsterkowicz może samodzielnie zamontować taką płytkę w swoim ST.

Rozdzielczość SM 124 wzrasta z 640 x 400 punktów do 672 x 480 punktów, a w przypadku najnowszych modeli tego monitora nawet do 704 x 480 punktów. Oznacza to możliwość wyświetlenia 5 dodatkowych wierszy i 4 kolumn tekstu. Największy przyrost powierzchni użytkowej, bo aż 78%, uzyskujemy w trybie średniej rozdzielczości dla kolorowych monitorów NEC MultiSync GS i 3D. (J.J)



Magazyn TOS

Historia trzydziestodwubitowego Atari TT rozpoczęła się bardzo niedawno. Nie znaczy to, że nie pisze się na temat tego komputera. Użytkownicy TT zdążyli się już doczekać własnego, specjalistycznego czasopisma, które nosi nazwę „TOS”.

W magazynie są publikowane artykuły poświęcone najnowszemu oprogramowaniu na ST i TT oraz sprzętowi. Redakcyjne testy pozwalają czytelnikowi wybrać naj-

lepszą i najciekawszą grę lub program użytkowy.

Dużą atrakcją są korzystne oferty firm software'owych, które zachęcają czytelników TOS-u do kupna pakietów oprogramowania po specjalnie dla nich obniżonych cenach. (J.J)

wiającą wykorzystanie komputera jako telefaksu. Jeśli ktoś uważa mikroprocesor 80286 za przestarzały i zbyt wolny, to nic nie stoi na przeszkodzie, by zamówić emulator z trzydziestodwubitowym procesorem 386SX. Premierę tej wersji zapowiadano na pierwszy kwartał 1991 r.

Komputer, którego nie można podłączyć do sieci, jest oczywiście dla wielu użytkowników tylko kupą złomu. Każdy PC można bez problemu połączyć z innymi komputerami tego typu, oczywiście po zainstalowaniu odpowiedniej karty. Sieci komputerów PC XT lub AT pracują już od lat w wielu in-

stytucjach i nie wzbudzają najmniejszej sensacji. Ale o Atari ST pracującym w sieci Novell warto już wspomnieć. Jest to możliwe dzięki wersji emulatora, która nosi nazwę SCplus/NET.

Na zakończenie warto się chwilę zastanowić, czy tak potężny emulator określić mianem dodatkowego wyposażenia do 520 lub 1040 ST? Dowiecinni powiedzą raczej, że Atari jest w tym przypadku tylko inteligentną klawiaturą współpracującą z SuperChargerem.

(J.J)

W
I
E
Ś
C
I

W „Tylko o Atari” i w pierwszych numerach „Mojego Atari” były publikowane pełne instrukcje programów. Natomiast instrukcje do „Wargame Construction Set” i „Broadsides” są nadmiernie skrócone i nie można na ich podstawie korzystać z tych programów.

**Krzysztof Wiśniewski
Warszawa**

Pełna instrukcja jest dołączana do każdego programu przez jego producenta lub licencjonowanego dystrybutora. Programy kupowane od złodziei na giełdzie lub w „studiach komputerowych” instrukcji nie posiadają albo instrukcje te są tłumaczone łamaną polszczyzną i odbite na kserografie. Zadaniem cyklu „Vademecum Atari” nie jest zastępowanie tych instrukcji. Zamieszczane w nim opisy mają pełnić rolę podręcznej pomocy (tzw. *quick reference*) i celowo są drukowane w taki sposób, aby umożliwić ich wycięcie bez niszczenia pozostałej części pisma.

(WZ)

Czy można poprawić program „Bulldogger Copy” tak, aby odczyt był normalny (z magnetofonu), ale zapis na dyskietkę?

**Piotr Mierzejewski
Poznań**

Oczywiście, że można, ale równa się to napisaniu od początku prawie połowy tego programu. Tylko, kto ma to zrobić?

(WZ)

Atari Turbo 2000 F

Nowy system transmisji danych z magnetofonem przyspieszony do 6700 bodów.

Komplet:

- cartridge
- oprogramowanie
- przeróbka magnetofonu
- instrukcja obsługi
- 12 miesięcy gwarancji

Instalacje wykonujemy na oczekaniu.

Interfejs do zwykłego magnetofonu.

Duży wybór oprogramowania w standardzie TURBO-2000.

Informacja:
Tel.33-40-91

B82

Korespondencja:

MUEL
Ul.Cząstkowska 30
01-678 Warszawa

W swojej grze wykorzystałem program „Okna dla Atari” z „Baj-

tki — Tylko o Atari” 2. Mam zamiar swoją grę sprzedawać. Czy nie będzie to kolidowało z prawami autorskimi tego programu? Jednocześnie informuję, że w grze jest podane, jaki to program i gdzie był zamieszczony.

**Marek Michalik
Toruń**

W świetle obowiązujących przepisów podanie autora i miejsca publikacji procedur użytych we własnym programie jest wystarczające, aby nie naruszyć praw autorskich. Dotyczy to jednak tylko procedur, które zostały opublikowane lub w których autor wyraził zgodę na dalsze stosowanie w innych programach. Wszystkie zamieszczane w „Bajtku” i „Moim Atari” programy mogą być w ten sposób wykorzystywane.

(TW)

Jestem radioamatorem, posiadam świadectwo uzdolnienia i licencję operatora klasy A, a interesuje mnie technika RTTY. Byłbym bardzo wdzięczny za opublikowanie w waszym magazynie programu RTTY dla Atari 800XL oraz schematu modemu.

**Jerzy Zdanowski
SP9WFW
Kraków**

Niestety, nikt w naszej redakcji nie zajmuje się krótkofalarstwem (z wyjątkiem kilku użytkowników CB i jednego operatora służby morskiej). Chętnie opublikowalibyśmy programy umożliwiające wykorzystanie komputera Atari w krótkofalarstwie. W tej dziedzinie zdani jesteśmy jednak wyłącznie na czytelników, którzy przesyłają do redakcji swoje programy. Być może po opublikowaniu tego listu ktoś taki program nam nadeśle.

(WZ)

Posiadam drukarkę Atari 1029. Niestety, nie potrafię uzyskać poprawnego wydruku z programu „Design Master”. Próbowałem zmieniać kody sterujące, lecz nie daje to oczekiwanych efektów. Podobne kłopoty mam z programem „Print Shop”.

**Jarosław Czereśniowski
Poznań**

Program „Print Shop” działa tylko z drukarkami wymienionymi w jego menu lub z takimi, które im odpowiadają. Atari 1029 jest niezgodna z żadnym innym typem drukarki i nie może współpracować z tym programem. Natomiast „Design Master” ma kody ustawiane przez użytkownika, więc można go przystosować do niemal każdej drukarki. W wypadku 1029 należy zastosować następujące kody:

Line spacing: 27, 57
Bit-Map Code: 27, 65, 1, 64
Unidirectional: 0, 0

Uzyskany wydruk nie będzie jednak idealny, gdyż program jest przystosowany do druku grafiki przy użyciu ośmiu igieł drukarki, a 1029 posiada tylko siedem igieł.

(WZ)

Bardzo zdziwiłem się, gdy przeczytałem test interfejsu Turbo 2600. Około 1,5 roku temu system ten był intensywnie reklamowany w audycji „Ra-

diokomputer nocą”. Zachęcony reklamą, nabyłem taki interfejs, lecz w październiku ubiegłego roku przestał on działać. Niestety, w listopadzie firmę zlikwidowano. Do dnia dzisiejszego nie otrzymałem z powrotem interfejsu, a większość programów mam w tym systemie. Piszę o tym, aby ostrzec innych użytkowników i jednocześnie chciałbym prosić o pomoc, może ktoś odsprzedałby mi lub pożyczył interfejs Turbo 2600, abym mógł odzyskać swoje programy.

**Stanisław Przybysz
Pułtusk**

Test interfejsu Turbo 2600 z różnych przyczyn, częściowo również z winy redakcji, został opublikowany z dużym opóźnieniem. Niestety nie wiedzieliśmy, że firma „Szok” przestała istnieć. Przepraszamy wszystkich czytelników i prosimy o pomoc dla kolegi.

(WZ)

Po opublikowaniu w „Moim Atari 3” ogłoszenia o możliwości zakupu twardego dysku do Atari XL/XE otrzymaliśmy wiele listów z pytaniami o szczegóły. Do komputerów Atari (również ST) można przyłączyć tylko twarde dyski z kontrolerem SCSI. Konieczne jest przy tym

zastosowanie odpowiedniego interfejsu, który dołącza się do szyny równoległej komputera. W starszych egzemplarzach 65XE, które nie mają wyprowadzenia tej szyny, niezbędna będzie więc ingerencja do wnętrza komputera. Pełne wykorzystanie twardego dysku wymaga także posiadania zwykłej stacji dysków. Jest ona potrzebna do zainicjowania pracy dysku oraz przeniesienia nań programów i danych. Wprawdzie wystarczy to wykonać jednorazowo, lecz później stacja dysków elastycznych będzie również przydatna do uzupełniania biblioteki programów, wykonywania kopii bezpieczeństwa oraz przeniesienia danych na inne komputery. Ponieważ napęd twardego dysku można w Polsce nabyć (jest on jednak droższy od stosowanych powszechnie w IBM), to problem polega tylko na wykonaniu odpowiedniego interfejsu. Można więc zastosować napęd o większej pojemności, np. 30, 40 lub nawet 80 MB. Publikując ogłoszenie nie mogliśmy przewidzieć jak duży będzie na nie odzew. W związku z tym prace nad interfejsem zostały rozpoczęte dopiero po nadejściu 10 zgłoszeń. Powinny one zostać zakończone we wrześniu, gdyż realizujący to kolega robi to tylko w czasie wolnym.

(WZ)

SPECTRA

oficjalny dystrybutor polskich autoryzowanych gier i programów oferuje:

Atari:

INSIDE — czyli Chipwar, komputerowa wojna w...
GABI — gra niczym z filmów Walta Disneya
KVADRYK — smok w labiryncie
PWW — zestaw gier edukacyjnych dla dzieci
FUTURE COMPOSER — najnowszy, rewelacyjny program muzyczny
IRON DEBUGGER — monitor o niespotykanych możliwościach
SPECTRA COPY — „inteligentny” kopier kasetka-kaseta-dysk
IRON HACKER — sam wykonuje wersje „file”

ZX Spectrum:

THE OMNI CORPUS — rewelacyjna gra wojenna. Absolutny przebój!!!
MOB — zestaw gier gwarantujących dobrą zabawę
COMPOSER 128 — najnowszy program muzyczny na generator AY

Timex:

TIMEX STUDIO — jedyny program graficzny w pełni wykorzystujący możliwości tego komputera
TIMEX BASIC — wykorzystuje ukryte możliwości Timexa

Commodore:

VOICE TRACKER — pierwszy polski w pełni profesjonalny edytor muzyczny. Rewelacja!
W przygotowaniu nowe propozycje.
Zapraszamy do współpracy sklepy, hurtownie, studia komputerowe.
Realizujemy zamówienia hurtowe, jednostkowe i wysyłkowe.

Adres: SPECTRA
21-422 Stanin
tel. 11-70, woj. siedleckie

B 113

Na listy
Czytelników
odpowiadają
Janusz
Jarmoch,
Tomasz
Wiśniewski
i Wojciech
Zientara



EDYTOR BASIC-a

Przy przepisywaniu programów w Basicu nie sposób ustrzec się błędów.

Aby uniknąć żmudnego wyszukiwania popełnianych omyłek wszystkie programy w Basicu są publikowane wraz z kodami kontrolnymi. Zamieszczony obok program służy do kontroli tych kodów podczas przepisywania programu.

Wydruk „Edytora Basica” należy dokładnie przepisać i zapisać na kasecie lub dyskietce (najlepiej instrukcją LIST „C:” lub LIST „D:EDYTOR.LST”). Poprawność przepisania można sprawdzić samym „Edytorem” w opisany niżej sposób.

Przystępując do wpisywania dowolnego programu z naszego pisma trzeba najpierw wczytać i uruchomić „Edytor Basica”. Następnie należy przepisywać kolejne wiersze programu. Po wpisaniu każdego wiersza i naciśnięciu klawisza RETURN wiersz ten pojawia się w dowolnej części ekranu wraz z obliczonym kodem kontrolnym. Jeżeli wyświetlony kod jest taki sam, jak wydrukowany przed numerem wiersza, można przystąpić do wpisywania następnego wiersza. Jeśli kody są różne, to ponowne naciśnięcie RETURN wyświetla wpisany wiersz w górnej części ekranu i umożliwia dokonanie w nim niezbędnych poprawek.

Samo naciśnięcie RETURN wywołuje zawsze ostatnio wpisany wiersz. W celu wywołania innego, wcześniej napisanego wiersza trzeba podać jego numer poprzedzony gwiazdką (np. 1000) i nacis-

nąć RETURN. Wpisanie samej liczby powoduje wymazanie z pamięci komputera wiersza programu o takim numerze.

Po poprawnym przepisaniu całego programu trzeba przerwać pracę „Edytora Basica” przez naciśnięcie klawisza BREAK lub RESET. Następnie w celu usunięcia „Edytora” i zbędnych zmiennych (z tablic nazw i wartości zmiennych) zapisujemy program na kasecie instrukcją LIST „C:”, 0,31999 lub na dyskietce instrukcją LIST „D:nazwa”.0,31999. Teraz kasujemy zawartość pamięci komputera instrukcją NEW i odczytujemy program przy pomocy ENTER „C:” (z kasy) lub ENTER „D:nazwa” z dyskietki. Po wykonaniu tych czynności w pamięci komputera znajduje się tylko gotowy program bez „Edytora Basica” i można go już ostatecznie zapisać na odpowiedni nośnik.

Procedura ta jest może nieco kłopotliwa, lecz zabezpiecza w stu procentach przed popełnieniem omyłki przy przepisywaniu programu. PAMIĘTAJ: ZAWSZE UŻYWAJ „EDYTORA BASICA”.

```

NC 32000 REM EDYTOR BASICA
SE 32010 REM wersja 1.1 dla "Bajtka"
JB 32020 CLR :DIM LINIA$(120):CLOSE #2:CLOSE #3
MN 32030 OPEN #2,4,0,"E:":OPEN #3,5,0,"E:
"
ZR 32040 ? CHR$(125):POSITION 11,1:? "ED
YTOR BASICA"
YB 32050 TRAP 32040:POSITION 2,3:? "Wpisz
linie programu"
KH 32060 POSITION 1,4:? " ":INPUT #2;LINI
4$:IF LINIA$="" THEN POSITION 2,4:LIST
B:GOTO 32060
XL 32070 IF LINIA$(1,1)="*" THEN B=VAL(LI
NIA$(2,LEN(LINIA$))):POSITION 2,4:LIST
B:GOTO 32060
TH 32080 POSITION 2,10:? "CONT"
BI 32090 B=VAL(LINIA$):POSITION 1,3:? " "
;
NY 32100 POKE 842,13:STOP
CN 32110 POKE 842,12
KI 32120 ? CHR$(125):POSITION 11,1:? "ED
YTOR BASICA":POSITION 2,15:LIST B
BZ 32130 C=0:ODP=C
LL 32140 POSITION 2,16:INPUT #3;LINIA$:IF
LINIA$="" THEN ? "LINIA ";B;" USUNIET
A":GOTO 32050
GU 32150 FOR D=1 TO LEN(LINIA$):C=C+1:ODP
=ODP+(C*ASC(LINIA$(D,D))):NEXT D
SR 32160 KOD=INT(ODP/676)
BX 32170 KOD=ODP-KOD*676
BM 32180 KODS=INT(KOD/26)
VF 32190 KODM=KOD-(KODS*26)+193
NF 32200 KODS=KODS+193
UW 32210 POSITION 0,16:? CHR$(KODS);CHR$(
KODM)
HN 32220 POSITION 2,13:? "Jezeli kod sie
nie zgadza, naciśnij RETURN i popr
aw linie.":GOTO 32050
    
```

WPISYWANIE ZNAKÓW SPECJALNYCH

Część programów publikowana w naszym piśmie jest drukowana w trybie graficznym.

Jest to spowodowane występowaniem w tych programach znaków, które są stosowane tylko w komputerach Atari i nie mogą być przedstawione przez drukarkę w trybie znakowym.

W trybie graficznym znaki znajdujące się obok siebie są

```

ABCDEFGHIJKLMNPOQRSTUVWXYZ
ABCDEFGHIJKLMNPOQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
abcdefghijklmnopqrstuvwxyz
01234567890123456789
!"#$%&'()*+,-,./:;?
!"#$%&'()*+,-,./:;?
    
```

Zestaw znaków Atari zawiera ponadto znaki specjalne, które wymagają specjalnego wpisywania. Wygląd tych znaków oraz klawisze, które służą do ich uzyskania, są pokazane w tabelkach umieszczonych obok. Zawsze, gdy są używane klawisze <CONTROL> (w tabelach CTRL) i <SHIFT>, trzeba trzymać je wciśnięte podczas naciskania następnego klawisza. Gdy konieczne jest użycie klawisza <ESC>, należy go nacisnąć tylko jeden raz i zwolnić przed naciśnięciem następnego klawisza. Znaki pokazane w tabeli „INVERSE” są negatywami znaków z tabeli „NORMALNIE”. Przełączenie na znaki w negatywie i odwrotnie następuje po naciśnięciu klawisza z symbolem przekreślonego prostokąta (tzw. klawisz Ata-

ri). Przełączanie na pisanie w negatywie (lub odwrotnie) nie jest konieczne dla uzyskania znaków, które są uzyskiwane przy użyciu <ESC>. Są one zawsze wyświetlane w sposób pokazany w tabelach.

Błędy najczęściej występujące przy przepisywaniu programów są powodowane przez wpisanie dużej litery zamiast małej — dotyczy to szczególnie liter P, X i O. Ponadto łatwo pomylić literę O z cyfrą 0, kropkę (.) z przecinkiem (,) oraz średnik (;) z dwukropkiem (:). Podobnie, niektóre znaki specjalne mają wygląd bardzo zbliżony do zwykłych znaków alfanumerycznych. Na wydruku różnić je można tylko według grubości linii. Dla porównania i zapamiętania znaki są pokazane poniżej.

SPECJALNE		NORMALNE	
	CTRL F		znak /
	CTRL G		SHIFT +
	CTRL N		SHIFT -
	CTRL R		znak -
	CTRL S		znak +

NORMALNIE

	CTRL ,		CTRL S
	CTRL A		CTRL T
	CTRL B		CTRL U
	CTRL C		CTRL V
	CTRL D		CTRL W
	CTRL E		CTRL X
	CTRL F		CTRL Y
	CTRL G		CTRL Z
	CTRL H		ESC ESC
	CTRL I		ESC CTRL -
	CTRL J		ESC CTRL =
	CTRL K		ESC CTRL +
	CTRL L		ESC CTRL *
	CTRL M		CTRL .
	CTRL N		CTRL ;
	CTRL O		SHIFT =
	CTRL P		ESC CTRL CLEAR
	CTRL Q		ESC DELETE
	CTRL R		ESC TAB

INVERSE

	CTRL ,		CTRL S
	CTRL A		CTRL T
	CTRL B		CTRL U
	CTRL C		CTRL V
	CTRL D		CTRL W
	CTRL E		CTRL X
	CTRL F		CTRL Y
	CTRL G		CTRL Z
	CTRL H		ESC SHIFT DELETE
	CTRL I		ESC SHIFT INSERT
	CTRL J		ESC CTRL TAB
	CTRL K		ESC SHIFT TAB
	CTRL L		CTRL .
	CTRL M		CTRL ;
	CTRL N		SHIFT =
	CTRL O		ESC CTRL 2
	CTRL P		ESC CTRL DELETE
	CTRL Q		ESC CTRL INSERT
	CTRL R		

G
R
AROBOT
R-29

Ta gra nie jest
zwykłą
zręcznościówką!
Potrzebne jest
logiczne
myślenie,
spostre-
gawczość
i umiejętność
kojarzenia.

Kierujesz robotem R-29,
który porusza się po planszy
z różnymi przedmiotami.
Niektóre pomagają, lecz są
i takie, które niszczą robota.

Oto wykaz przedmiotów
wrogich: miny, pola
magnetyczne, pola
elektryczne.

Pomocne są:
— bomby: każda może
zniszczyć jedną minę,
— magnes: dezintegruje
jedno pole magnetyczne,
— klucz francuski: pozwala
na jedno bezpieczne
przejście przez pole
magnetyczne.

Zadanie robota polega na
dotarciu do wyjścia,
oznaczonego literą „E”,
prowadzącego na wyższy
poziom.

Mimo iż gra jest dość
prosta, ma duże możliwości
rozbudowy (teleportacje,
nowe kombinacje
przedmiotów, drzwi-klucz,
groźne stwory, pistolet itd.)
służą do tego wolne wiersze
od 210 do 298.

Gra ma pięć plansz —
pierwsze trzy są niewielkie,
o małym stopniu trudności.

Czwarta to plansza
z pułapką, piąta —
najbardziej rozbudowana
— przykład dużej
i skomplikowanej
planszy o dużym
stopniu trudności.

Plansze można
dowolnie zmieniać,
a także dorabiać
nowe (od wiersza

600 do 619,
następna
od 620 do 639 itd.).

Każda plansza powinna
mieścić się w 20 wierszach
i zaczynać się od wiersza,
którego numer jest
wielokrotnością 20.

Grę można dodatkowo
utrudnić przez wprowadzenie
limitu czasu — oto wiersze
uzupełniające:

```
51 CZAS=150: POSITION
2,19: ? „CZAS -”
109 CZAS=CZAS-1: POSITION
9,19: ? CZAS: IF CZAS=0
THEN 300
```

Ilość czasu można
zmienić w pierwszej
instrukcji 51 wiersza.
joysticka
Życzę połamania
i nowych udoskonaleń.

Tomasz
ROGACEWICZ

```
NR 0 REM ROBOT R-29
EU 1 REM Tomasz Rogacewicz
DU 2 REM (c) 1991, Sp. Bajtek
NI 3 REM
FW 10 GRAPHICS 0:POKE 752,1: ? "CZEKAJ 30
S":POKE 82,0
KD 20 FOR A=0 TO 1023:POKE 32768+A,PEEK(5
7344+A):NEXT A
RN 30 READ A:TRAP 40:FOR B=0 TO 7:READ C:
POKE 32768+A*8+B,C:NEXT B:GOTO 30
ED 40 PL=0:ZY=3:POKE 756,128
RZ 50 X=1:Y=1:B=0:C=0:D=0:GOSUB 500+PL*20
TC 60 POSITION 2,20: ? "BOMBY-": ? " MAGN.
-": ? " KLUCZE-":
NH 99 REM GLOWNA PETLA
EA 100 X1=X:Y1=Y:IF STICK(0)=7 THEN X=X+1
:W=64:GOTO 110
FN 101 IF STICK(0)=11 THEN X=X-1:W=65:GOT
0 110
ZB 102 IF STICK(0)=13 THEN Y=Y+1:W=66
DI 103 IF STICK(0)=14 THEN Y=Y-1:W=67
MW 110 IF PEEK(40000+40*Y+X)<0 THEN GOSUB
B 200
CJ 120 POKE 40000+40*Y1+X1,0:POKE 40000+4
0*Y+X,W:IF PEEK(764)=28 THEN POKE 764,
255:GOTO 300
LM 130 GOTO 100
TT 200 L=PEEK(40000+40*Y+X)
AW 201 IF L=64 OR L=65 OR L=66 OR L=67 TH
EN RETURN
FV 202 IF L=68 THEN B=B+1:GOSUB 400:POSIT
ION 9+B,20: ? "J":RETURN
NL 203 IF L=69 THEN C=C+1:GOSUB 400:POSIT
ION 9+C,21: ? "I":RETURN
RF 204 IF L=70 THEN D=D+1:GOSUB 400:POSIT
ION 9+D,22: ? "I":RETURN
RM 205 IF L=71 THEN POSITION 9+B,20: ? " "
:B=B-1:GOSUB 410:X1=X:Y1=Y:IF B<0 THEN
300
CF 206 IF L=72 THEN POSITION 9+C,21: ? " "
:C=C-1:GOSUB 420:X1=X:Y1=Y:IF C<0 THEN
300
LY 207 IF L=73 THEN POSITION 9+D,22: ? " "
:D=D-1:GOSUB 430:GOSUB 450:IF D<0 THEN
300
QL 208 IF L=37 THEN 350
PZ 299 X=X1:Y=Y1:RETURN
QS 300 X=X1:Y=Y1:POP :GOSUB 410:GOSUB 410
:ZY=ZY-1:IF ZY<0 THEN 310
RH 305 GOTO 50
CR 310 FOR A=0 TO 255:SOUND 0,A,14,8: ? "T
HE END.":NEXT A:SOUND 0,0,0
PD 320 ? "K++
VT 321 ? : ? " ROBOT R-29"
TOMASZ ROGACEWICZ
LJ 322 ? : ? " ELBLAG 20.11.90"
: ? : ? : ? " Fire"
YN 330 IF STRIG(0)=0 THEN 40
NR 340 GOTO 330
WS 350 GOSUB 410:PL=PL+1:GOTO 50
RP 400 SOUND 0,20,14,10:FOR A=0 TO 15:NEX
T A:SOUND 0,0,0,0:RETURN
LJ 410 FOR A=0 TO 100 STEP 4:SOUND 0,A+50
,14,14:POSITION X,Y: ? "X":NEXT A
RR 415 SOUND 0,0,0,0:RETURN
MS 420 FOR A=15 TO 0 STEP 0.4:SOUND 0,100
,8,A:NEXT A:RETURN
NR 430 FOR A=0 TO 15:SOUND 0,100,A,15:NEX
T A:RETURN
BR 450 IF W=64 THEN X=X+1
DV 451 IF W=65 THEN X=X-1
EF 452 IF W=66 THEN Y=Y+1
GJ 453 IF W=67 THEN Y=Y-1
TG 455 X1=X:Y1=Y:RETURN
MN 480 DATA 64,0,48,96,108,124,108,32,120
,65,0,12,6,110,62,110,2,30
SQ 481 DATA 66,0,24,60,102,102,60,24,60,6
7,0,24,60,126,126,60,24,60
HC 482 DATA 68,6,8,28,62,62,62,28,0,70,0,
0,102,60,60,102,0,0
LM 483 DATA 69,0,24,62,96,96,62,24,0,72,2
04,102,51,153,204,102,51,153
GU 484 DATA 73,8,64,20,41,148,40,2,16,71,
0,60,90,102,102,90,60,0
SU 500 ? "K"
LU 501 ? " "
XH 502 ? " "
KH 503 ? " "
TI 504 ? " "
AI 505 ? " "
KB 506 ? " "
SP 507 ? "":RETURN
RT 519 REM
LS 520 ? "K"
JX 521 ? " "
QC 522 ? " "
CX 523 ? " "
QF 524 ? " "
AJ 525 ? " "
HV 526 ? " "
AD 527 ? "":RETURN
RX 539 REM
TC 540 ? "K"
FT 541 ? " "
OO 542 ? " "
LX 543 ? " "
AU 544 ? " "
TX 545 ? " "
SU 546 ? "":RETURN
SB 559 REM
VI 560 ? "K"
NF 561 ? " "
HQ 562 ? " "
BY 563 ? " "
SO 564 ? " "
AF 565 ? "":RETURN
SF 579 REM
EO 580 ? "K"
WM 581 ? " "
RM 582 ? " "
FD 583 ? " "
LA 584 ? " "
QC 585 ? " "
XM 586 ? " "
MK 587 ? " "
JU 588 ? " "
ET 589 ? " "
YZ 590 ? " "
KQ 591 ? " "
FQ 592 ? " "
XT 593 ? " "
BW 594 ? "":RETURN
YF 595 REM ***
```

DYSKIETKA?

TO

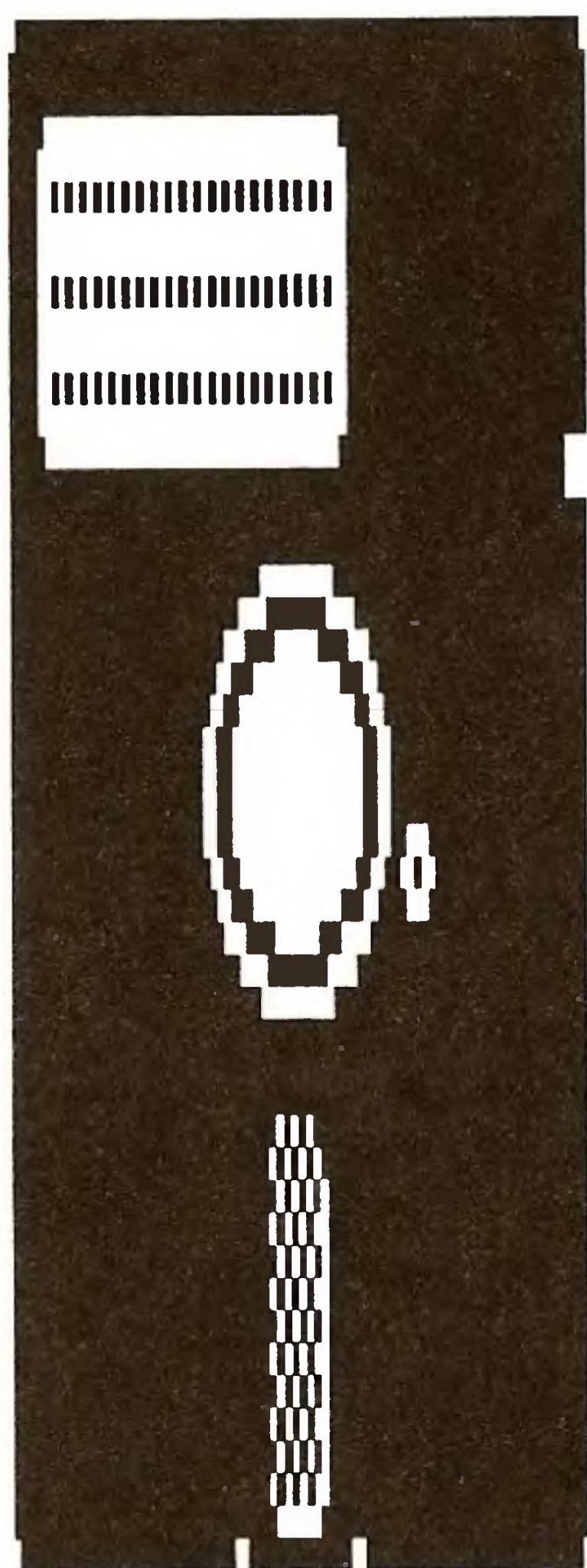
PROSTE!

Pamięć masowa służy do przechowywania dużych ilości programów i danych. Podstawowym rodzajem takiej pamięci jest dla większości komputerów osobistych stacja dysków elastycznych, a jej nośnikiem — dyskietka.

W miarę rozwoju komputerów konstruowano coraz lepsze i sprawniejsze systemy zapisu na nośnikach magnetycznych. Początkowo była to pamięć taśmowa. W znacznie udoskonalonej formie jest ona nadal stosowana w dużych komputerach, a w formie zdegenerowanej w prostych komputerach domowych (z wykorzystaniem magnetofonu). Aby wyeliminować konieczność przewijania taśmy i tym samym zwiększyć szybkość dostępu do zapisanych informacji, umieszczono warstwę magnetyczną na krążkach, tworząc w ten sposób pamięć dyskową. Miała ona jednak niewielką pojemność. Jej rozwój poszedł w dwóch kierunkach: zwiększania liczby dysków w jednym napędzie i zwiększania prędkości obrotowej, a tym samym gęstości upakowania danych i szybkości dostępu do nich.

Nie wnikając w szczegóły techniczne i historyczne powiedźmy tylko, że druga z wymienionych dróg rozwoju doprowadziła do powstania twardych dysków. Natomiast liczba dysków w jednym napędzie szybko osiągnęła wartość graniczną (ze względów konstrukcyjnych i ekonomicznych). Wyjściem było umieszczanie w napędzie jednej wymiennej dyskietki.

Początkowo dyskietki takie miały średnicę 8 cali (203 mm), są one jeszcze używane w niektórych starszych komputerach profesjonalnych. Dążąc do ograniczenia rozmiarów sprzętu i — dzięki postępowi technicznemu — mając takie możliwości, konstruktorzy wprowadzili nowy standard: dyskietkę o średnicy 5,25 cala (133 mm). Wielkość ta przyjęła się powszechnie w komputerach osobistych. Dopiero stosunkowo niedawno — kilka lat temu — opracowano nowe dyskietki



o średnicy 3,5 cala (89 mm). Kilka firm próbowało wprowadzić dyskietki 3-calowe (76 mm), lecz standard ten został przyjęty tylko w komputerach Amstrad i Spectrum.

W komputerach Atari XL i XE stosowane są dyskietki o średnicy 5,25 cala, w ST zaś 3,5 cala.

Budowa dyskietki

Dyskietka jest to krążek z miękkiego tworzywa pokryty warstwą magnetyczną i umieszczony w ochronnej kopercie (również z tworzywa). Wygląd krążka i koperty jest pokazany na rysunku. Widać tam, że oba elementy mają różne otwory o określonym przeznaczeniu. Zajmiemy się nimi kolejno.

Duży, okrągły otwór w środku dyskietki i koperty służy do zamocowania dyskietki na osi silnika nadającego jej prędkość obrotową. Dyskietki niektórych firm mają ten otwór wzmocniony przez nałożenie dodatkowego pierścienia, który zwiększa sztywność zamocowania. Ponadto dyskietka posiada mały otworek indeksowy umożliwiający rozpoznanie przez stację położenia dyskietki. W stacji znajdują się fototranzystor i dioda świecąca, umieszczone po obu stronach dyskietki. Gdy otworek indeksowy obracającej się dyskietki znajdzie się pomiędzy nimi, w tranzystorze powstaje impuls elektryczny sygnalizujący położenie dyskietki. Aby otworek ten był widoczny, w kopercie również znajduje się otwór indeksowy, którego położenie odpowiada położeniu

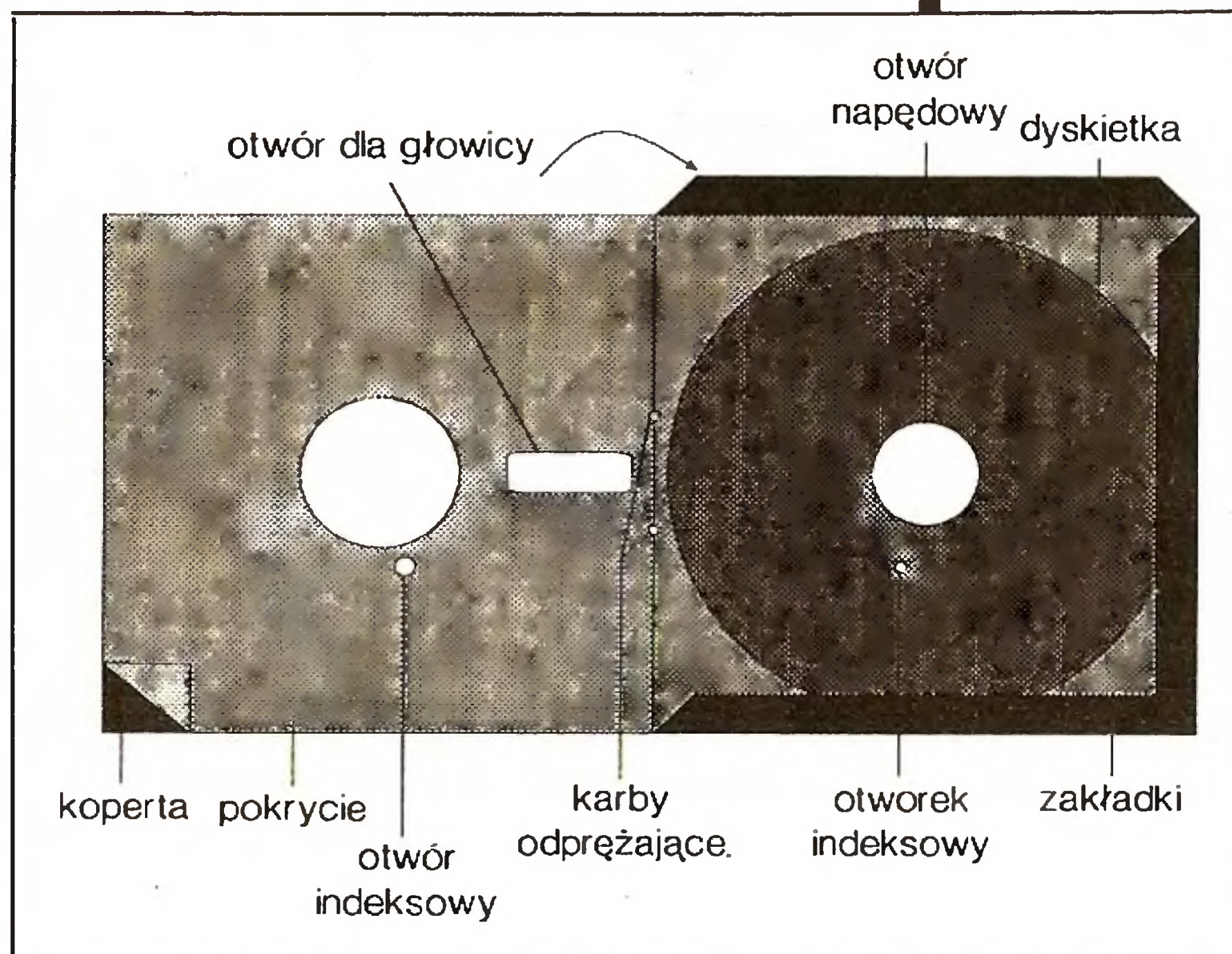
diody i tranzystora w stacji.

Podłużny otwór w kopercie umożliwia dostęp głowicy odczytująco-zapisującej do powierzchni dyskietki. Ponieważ przez otwór ten widoczna jest bezpośrednio ta część dyskietki, na której zapisywane są informacje, to pod żadnym pozorem nie można go dotykać, aby nie uszkodzić delikatnej warstwy magnetycznej. Dodatkowe dwa małe otworki na krawędzi koperty zmniejszają naprężenia powstałe na skutek zagięcia koperty i nie pozwalają na odkształcanie się jej w rejonie otworu dla głowicy.

Na jednej z połówek koperty znajdują się ponadto trzy podłużne zakładki, które pozwalają na zaklejenie koperty po jej złożeniu. Wewnętrzna powierzchnia koperty jest wyklejona specjalnym śliskim pokryciem zmniejszającym tarcie dyskietki o kopertę, a więc ułatwiającym jej obracanie i zmniejszającym zużycie.

Koperta dyskietki ma jeszcze jedno wycięcie nie uwidocznione na rysunku. Jest ono umieszczone na krawędzi koperty i służy do zablokowania zapisu na dyskietce. Zasada działania tego wycięcia jest podobna jak otworu indeksowego — gdy jest ono zaklezione, zapis na dyskietce jest niemożliwy. Fabrycznie każda dyskietka ma takie wycięcie, a w pudełku dołączone są specjalne zakładki. Większość stacji dysków do Atari ma tylko jedną głowicę i zapisanie drugiej strony dyskietki wymaga jej odwrócenia. Ponieważ po drugiej stronie brak takiego fabrycznego wycięcia, to użytkownik musi wykonać je sam.

Widok po rozklejeniu koperty dyskietki



Działanie dyskietki

W największym uproszczeniu można powiedzieć, że dyskietka działa tak jak taśma magnetofonowa. Aby porównanie to przybliżyć do rzeczywistości, trzeba sobie wyobrazić krótką taśmę sklejoną w pętlę i ruchomą głowicę magnetofonu, która może zapisać na taśmie 40 ścieżek. Teraz wystarczy tylko przenieść tę pętlę taśmy na płaską powierzchnię krążka i już mamy dyskietkę. W rzeczywistości jest prawie tak samo.

Dyskietka obraca się w stacji z prędkością 286 obrotów/minutę (w IBM — 300 obr/min). Po jej powierzchni przesuwają się głowice, która jest zwykłym, choć miniaturowym, elektromagnesem. Do tego elektromagnesu przesyłane są szeregowo (kolejno, bit po bicie) informacje z komputera. Pole magnetyczne powstające w szczelinie głowicy powoduje uporządkowanie dipoli (można je sobie wyobrazić jako mikroskopijne igiełki magnetyczne) w warstwie magnetycznej dyskietki. Kierunek uporządkowania dipoli odpowiada wartości zapisywanego bitu. Utworzony w ten sposób zapis można sobie wyobrazić jako szereg malutkich magnesików ułożonych obok siebie na dyskietce i skierowanych w dwie różne strony.

Podczas odczytu te magnesiki (dipole), przesuwając się przed szczeliną głowicy, powodują powstanie w niej zmiennego pola magnetycznego, które z kolei wywołuje w uzwojeniu elektromagnesu (głowicy) przepływ prądu. Kierunek tego prądu zależy od kierunku ustawienia dipoli w warstwie magnetycznej. Prąd uzyskany z głowicy jest następnie wzmacniany, zamieniany na ciąg bitów i przesyłany do komputera.

Naturalnie w rzeczywistości jest to znacznie bardziej skomplikowane. Stacja dysków musi bowiem rozpoznawać właściwe miejsce na dyskietce, gdzie mają być zapisane albo skąd mają być odczytane dane. Wymaga to umieszczenia na dyskietce znaczników wskazujących początki poszczególnych porcji (sektorów) danych oraz ich grup (ścieżek). To jednak będzie tematem następnego artykułu.

Marek ZACHAR

Prezentowany program służy do analizowania nieskomplikowanych układów elektronicznych — tylko planarnych.

Program ten użytkowałem z powodzeniem przy projektowaniu zasilaczy stabilizowanych i prostych przedwzmacniaczy. Może on również pomóc przy naprawie układów poprzez podawanie napięć w poszczególnych punktach. Nie sprawdza on jednak poprawności układu, stąd np. przy błędnej polaryzacji tranzystora wyliczone napięcia mogą być bardzo zaskakujące. Wszystkie elementy z opisem końcówek przedstawia rysunek w programie.

```

QY      1 REM ANALYZA STALOPRADOWA'
KZ      2 REM Jacek Tarasjuk
IA      3 REM (C) 1990, Bajtek
NJ      4 REM
KE      10 REM W INT JALIZACJA SYSTEMU N
GH      15 POKE 82,0
RC      20 DIM T(31,30),K$(30),R(30),ZPR(3),ZM
          P(3),IXU(3),IXI(3),XU(3),UXI(3),TR(10)
          PRM(3,2),PRM(3,2),ZMPC(3,3)
UK      25 EXEC INFO
SW      26 DIM IXUW(3,4),IXIW(3,4),UXUW(3,5),U
          XIW(3,5),TRW(10,5),X(4),Y(4)
DR      27 X(1)=1:X(2)=26:X(3)=13:X(4)=26:Y(1)
          =5:Y(2)=5:Y(3)=13:Y(4)=13
YD      30 EXEC MENU
ZU      33 POSITION 2,19:?"
          "
RM      34 TRAP 34:POSITION 6,19:?"Podaj ilos
          c wezlomow :";INPUT K$:ILWE=VAL(K$):MAX
          W=ILWE:EXEC CZYSC
QU      35 EXEC WYBOR
TF      40 GRAPHICS 0:END
XO      100 PROC GAUSS
FJ      105 POSITION 6,19:?"PROSTE zmiennosci
          i zmiennosci"
TW      110 N=MAXK
OB      130 FOR I=1 TO N-1:IF T(I,J)=0 THEN EK
          EC ZAMIANA
PB      120 FOR I=J+1 TO N:IF T(I,I)=0 THEN GO
          TO 130
OD      125 M=T(I,J)/T(I,J):FOR K=J TO N:T(K,I)
          =T(K,I)-M*T(K,J):NEXT K:T(I,I)=T(I,I)
          -M*T(I,J)
EW      130 NEXT I:NEXT J
KD      135 IF T(N,N)=0 THEN STOP
ZS      140 T(31,N)=T(31,N)/T(N,N):FOR I=N-1 T
          O 1 STEP -1:0=D:FOR J=I+1 TO N:0=D+T(I
          ,J)*T(I,J):NEXT J
          145 T(I,I)=T(I,I)-D)/T(I,I):NEXT I
KN      147 EXEC CZYSC
VT      150 ENDPROC
UL      155 PROC ZAMIANA

```

```

0Y      1710 POSITION 3,20:? "wspolczynnik xs"
IE      IXI;";INPUT KS:IXI(IXI)=VAL(K$)
FG      1715 EXEC CZYSO
SD      1717 TRAP 1717:POSITION 3,19:? "Podaj
        opornosc wejscowa w omach "
SD      1719 POSITION 3,20:? "Opornosc we.Rwe"
HN      ";INPUT KS:RWE=VAL(K$)
        1720 EXEC CZYSO
        1722 TRAP 1722:POSITION 3,19:? "Podaj
        numery wezlow dla zrodla U/I"
0Y      1723 POSITION 3,20:INPUT "Wezel +Iwy="
IM      ;K$:W1=VAL(K$);IXIW(IXI,1)=W1
        1724 POSITION 23,20:INPUT "Wezel -Iwy="
ZL      ;K$:POSITION 0,0;W2=VAL(K$);IXIW(IXI,
        2)=W2
        1725 POSITION 3,20:? "
SH      1726 TRAP 1726:POSITION 3,20:INPUT "We
        zeI +Iwe=";KS:W3=VAL(K$);IXIW(IXI,3)=W
        3
RG      1728 POSITION 23,20:INPUT "Wezel -Iwe="
        ";KS:POSITION 0,0;W4=VAL(K$);IXIW(IXI,
        4)=W4
HQ      1730 EXEC CZYSO
HD      1731 Y=IXI(IXI)/RWE:T(W3,W1)=T(W3,W1)-
        Y:I(W4,W2)=T(W4,W2)-Y:T(W3,W2)=T(W3,W2
        )+Y:T(W4,W1)=T(W4,W1)+Y
HI      1735 Y=1/RWE:T(W3,W3)=T(W3,W3)+Y:T(W3,
        W4)=T(W3,W4)+Y
ZO      1736 T(W4,W4)=T(W4,W4)+Y:T(W4,W3)=T(W4,
        W3)+Y:T(W2,W2)=T(W2,W2)+Y
KD      1738 POSITION 6,6:? "Zrodlo pradowe st
        er.pradem"
ZC      1740 ENDPROC
LF      1750 PROC UKU
AB      1752 POSITION 6,7:? "PROLOG"
        1753 Y=XUK(UKI);UKU=UKU+1;MAXW=MAXW+1
        1755 TRAP 1755:POSITION 3,19:? "Podaj
        bazowymsarowy wspolczynnik vs "
UF      1760 POSITION 3,20:? "wspolczynnik vs"
IT      ;UKU;";INPUT KS:UKU(UKU)=VAL(K$)
        1765 EXEC CZYSO
DN      1770 TRAP 1770:POSITION 3,19:? "Podaj
        numery wezlow dla zrodla U/I"
WF      1771 POSITION 3,20:INPUT "Wezel +Uwy="
BF      ;K$:W1=VAL(K$);UXIU(UKU,1)=W1
        1774 POSITION 23,20:INPUT "Wezel -Uwy="
AA      ;K$:POSITION 0,0;W2=VAL(K$);UXIU(UKU,
        2)=W2
        1775 POSITION 3,20:? "
KH      1776 TRAP 1776:POSITION 3,20:INPUT "We
        zel +Uwe=";KS:W3=VAL(K$);UXIU(UKU,3)=W
        3
PZ      1778 POSITION 23,20:INPUT "Wezel -Uwe="
        ";KS:POSITION 0,0;W4=VAL(K$);UXIU(UKU,
        4)=W4
IFWK    1780 EXEC CZYSO
        1781 Y=XUK(UKI):T(MAXW,W1)=T(MAXW,W1)-
        1:T(MAXW,W2)=T(MAXW,W2)+1:T(W3,MAXW)=T
        (W3,MAXW)+Y
HC      1786 T(W2,MAXW)=T(W2,MAXW)-1:T(W1,MAXW
        )=T(W1,MAXW)+1:T(W4,MAXW)=T(W4,MAXW)+Y
JU      1788 POSITION 6,7:? "Zrodlo napiecieowe
        ster.napieciem"
ZR      1790 ENDPROC
ES      1800 PROC UXI
MT      1802 POSITION 6,8:? "Zrodlo napiecieowe"
        1803 Y=XUK(UXI);UXI=UXI+1;MAXH=MAXH+1;UX
        I(UXI,S)=MAXH
AI      1805 TRAP 1805:POSITION 3,19:? "Podaj
        Wspolczynnik zs w U/AJ"
AQ      1806 POSITION 3,20:? "wspolczynnik zs"
IG      ;UXI;";INPUT KS:UXI(UXI)=VAL(K$)
EU      1815 EXEC CZYSO
SD      1817 TRAP 1817:POSITION 3,19:? "Podaj
        opornosc wejscowa w omach "
SF      1819 POSITION 3,20:? "Opornosc we.Rwe"
HP      ";INPUT KS:RWE=VAL(K$)
        1820 EXEC CZYSO
KK      1822 TRAP 1822:POSITION 3,19:? "Podaj
        numery wezlow dla zrodla U/I"
TW      1823 POSITION 3,20:INPUT "Wezel +Uwy="
LU      ;K$:W1=VAL(K$);UXIW(UXI,1)=W1
        1824 POSITION 23,20:INPUT "Wezel -Uwy="
ZM      ;K$:POSITION 0,0;W2=VAL(K$);UXIW(UXI,
        2)=W2
        1825 POSITION 3,20:? "
CF      1826 TRAP 1826:POSITION 3,20:INPUT "We
        zel +Iwe=";KS:W3=VAL(K$);UXIW(UXI,3)=W
        3
KK      1828 POSITION 23,20:INPUT "Wezel -Iwe="
        ";KS:POSITION 0,0;W4=VAL(K$);UXIW(UXI,
        4)=W4
H5      1830 EXEC CZYSO
OH      1831 Y=UXI(UXI)/RWE:T(W3,MAXW)=Y:T(W4,
        MAXW)=Y:T(W2,MAXW)=1:T(W1,MAXW)=1
IP      1835 T(MAXW,W2)=1:T(MAXW,W1)=1
HE      1836 Y=1/RWE:T(W3,W3)=T(W3,W3)+Y:T(W4,
        W4)=T(W4,W4)+Y:T(W3,W4)=T(W3,W4)+Y:T(W4,
        W3)=T(W4,W3)+Y
QI      1838 POSITION 6,8:? "Zrodlo napiecieowe
        ster.pradem"
ZE      1840 ENDPROC
UV      1850 PROC NPM:T=T+1;MAHW=MAXH+1
XT      1853 IF P=-1 THEN POSITION 6,10:? "Tra
        nzytor p-n-p"
KA      1854 IF P=-1 THEN POSITION 6,11:? "Tra
        nzytor p-n-p"
KL      1855 TRAP 1855:POSITION 3,19:? "Podaj
        wspolczynnik beta "
OK      ;TRW(T,4)=-P:TRW(T,5)=MAXH
        1860 POSITION 3,20:? "wsp. beta tranzu
        "
IV      ;T=PRA:"";INPUT KS:TRCT=VAL(K$)
LH      1865 EXEC CZYSO
PK      1870 TRAP 1870:POSITION 1,19:? "Podaj
        numery wezlow dla tranzystora I"/I"
OC      1875 POSITION 3,20:INPUT "Wezel B=";K$
        :W1=VAL(K$);TRWT(T,1)=W1
        1876 POSITION 16,20:INPUT "Wezel C=";K
        $:W2=VAL(K$);TRWT(T,2)=W2
        1877 POSITION 27,20:INPUT "Wezel E=";K
        $:W3=VAL(K$);TRWT(T,3)=W3
        1880 EXEC CZYSO
PC      1885 T(MAXH,W1)=P:T(MAXH,W3)=P-P*TRCT
        );T(MAXH,W2)=P*TRCT);T(W3,MAXH)=P:T(C31
        ,MAXH)=T(C31,W3)+1:T(M1,MAXH)=P
EB      1890 IF P=-1 THEN POSITION 6,10:? "Tra
        nzytor n-p-n"
PO      1891 IF P=-1 THEN POSITION 6,11:? "Tra
        nzytor p-n-p"
AN      1895 ENDPROC
NI      3000 PROC NYDRUK
RO      3003 GRAPHICS 0:Poke 752,3:SETCOLOR 1,
        0,0:SETCOLOR 2,0,10:? " "
        3005 ? "|";??"E HWYD ANALYZ M"
        3006 FOR Y=3 TO 21:? "|"?:NEXT Y
        3007 ? " "
        3008 POSITION 12,20:? " "
        3010 POSITION 1,3:? "Napiecia w":POSIT
        ION 1,4:? "wezliach":POSITION 1,5:? "uk
        ladu "
UJ      3011 POSITION 13,3:? "Parametry ukladu "
JG      3015 POSITION 1,7:? "0-0.0 U":POW=0:F0
        R Y=1 TO ILWE:POSITION 1,7+Y-(14*POW):
        ? Y:"-";
KE      3020 Q=T(C31,Y):Q=Q*1000:IF Q=INT(Q):Q=
        0
AS      3022 Q=INT(Q)/1000
BI      3025 ? Q;" U";
YO      3028 IF Y=14 THEN POW=1:EXEC Klaw
GP      3030 NEXT Y:K=0
II      3034 IF Y=0 THEN 3055
FR      3035 FOR I=1 TO T:K=K+1:IF K=5 THEN K=
        1:EXEC Klaw
FW      3036 POSITION X(K),Y(K):? "Tr.";I;" "
LW      3037 IF TRW(I,4)=1 THEN ? "p-n-p"
GB      3038 IF TRW(I,4)=1 THEN ? "p-n-p"
NY      3039 POSITION X(K),Y(K)+1:? "B=";TRW(T,
        1);? "C=";TRW(T,2);? "E=";TRW(T,3)
TF      3040 POSITION X(K),Y(K)+2:? "wsp.beta="
        :TRCT
FC      3042 Q=T(C31,TRW(I,2))-T(C31,TRW(I,3)):E
        XEC ZAO:POSITION X(K),Y(K)+3:? "Uce=";
        Q;" U":U=Q
CW      3043 Q=T(C31,TRW(I,5)):EXEC ZAO:POSITIO
        N X(K),Y(K)+4:? "Ib=";Q;" A"
OW      3044 Q=T(C31,TRW(I,5))*TRCT):EXEC ZAO:P
        OSITION X(K),Y(K)+5:? "Ic=";Q;" A"
0W      3045 Q=Q*U:EXEC ZAO:POSITION X(K),Y(k)
        +6:? "Pe=";Q;" W"
EG      3050 NEXT I
EN      3055 FOR I=1 TO R:K=K+1:IF K=5 THEN K=
        1:EXEC Klaw
AZ      3056 POSITION X(K),Y(K):? "Rr.I;"
IR      3057 ? "P1=";RW(I,1);? "P2=";RW(I,2)
BU      3060 POSITION X(K),Y(K)+1:? "Rr.I;"
RC      3062 POSITION X(K),Y(K)+2:? "Ur=";Q=A

```


RAMbrandt

pasy poziome — **Hrz** lub <H> i dwa kolory
pasy pionowe — **Vrt** lub <V> i dwa kolory
malowanie EX-OR (ze złożeniem kolorów) —
<SHIFT><X>
ustawienie przzerwania — <I>, dalej jak przy
zmianie kolorów skasowanie przzerwania —
<W>

OKNO

okno — **Wnd** lub <SHIFT><W> i określenie roz-
miaru menu okien:
opuszczenie okna — <E>
lustrzane odbicie w pionie — <V>
lustrzane odbicie w poziomie — <H>
obróć — <R>
zmniejszenie kąta — <+>
zwiększenie kąta — <*>
wykonanie obrotu — <START>
zatrzymanie obrotu — <*>
animacja — <A>
kolejność rysunków — <CONTROL><D>
uruchomienie — <CONTROL><A>
zwalnianie — <+>
przyspieszanie — <*>
zatrzymanie/wznowienie — <spacja>
przerwianie — <ESC>

DEFINIOWANIE

definiowanie znaczka — <R> i numer — <1>-<4>
definiowanie kafa — <SHIFT><C> i wielkość —
<1>-<3>
koniec — <SHIFT><L> i numer kafa — <1>-
<5>
definiowanie pikowania — <SHIFT><D> i wiel-
kość — <1>-<3>
koniec — <SHIFT><R> i numer pikowania
— <1>-<5>
definiowanie pędzla — <SHIFT><M>
koniec — <SHIFT><E>
definiowanie losowego zestawu kolorów — <Y>

ZAPIS I ODCZYT

odczyt rysunku z dyskietki — **Loa** <L>
odczyt spisu rysunków — <D>

wydruk spisu rysunków — <P>
numer rysunku — <0>-<9>
przerwianie odczytu — <START>

zapis rysunku na dyskietkę — **Sav** lub <S>
inicjowanie nowej (sformatowanej) dys-
kietki — <I>
numer rysunku — <0>-<9>
nazwa rysunku — do 32 znaków
przerwianie zapisu — <START>
zapis rysunku do ramdysku — <SHIFT><=>
numer rysunku — <0>-<7>
odczyt rysunku z ramdysku —
<SHIFT><=>
numer rysunku — <0>-<7>
odczyt zestawu znaków — <SHIFT><A>
odczyt wzorów (pędzle, kafe i pikowa-
nia) — <CONTROL><L>
zapis wzorów (pędzle, kafe i pikowania) —
<CONTROL><S>

odczyt dodatkowego modułu — <G>
spis modułów — <D>
wybór modułu — <1>-<3>
1 — drukowanie
2 — zapis i odczyt w formacie DOS
pierwsze menu:
1 — zapis
2 — odczyt
3 — katalog
drugie menu:
1 — rysunek MicroPainter
2 — rysunek Koala
3 — rysunek RAMbrandt
4 — zestaw znaków
5 — zestaw wzorów
6 — koniec
powrót do RAMbrandta — <START>
3 — druk trybów 1, 2 i 3

Marek ZACHAR

— STACJA

**Firma TOMS, której rozszerze-
nia stacji dysków były kilkakrot-
nie prezentowane na łamach
„Bajtka”, wprowadziła na rynek
nowy produkt — stację dysków
TOMS 720. Jedną z pierwszych
(jeszcze ciepłą) stacji została
przekazana do przetestowania
naszej redakcji.**

TOMS 720 jest dwustronną (dwugłowicową) sta-
cją dysków elastycznych 5,25 cala, przeznaczoną
do współpracy z 8-bitowymi komputerami Atari
(800XL, 800XE, 65XE i 130XE). Umożliwia ona jed-
no- lub dwustronny zapis i odczyt dyskietek w sied-
miu różnych formatach. Przyłączenie do komputera
jest zrealizowane poprzez standardowe złącze sze-
regowe Atari.

BUDOWA

Obudowa stacji jest wykonana z pomalowanej na
szaro blachy i ma na górnej powierzchni otwory uła-
twiające chłodzenie jej układów i mechanizmów.
Otwory te są ukośne, aby zbliżyć wygląd stacji do
pozostałych elementów systemu. Część przedniej
ścianki obudowy stanowi płytę czołową napędu. W
porównaniu z dotychczasowymi stacjami **TOMS
720** jest nieco węższa, a za to wyższa i dłuższa. Jej
wymiary wynoszą: szerokość 155 mm, wysokość
85 mm i długość 320 mm (dla porównania: LDW
2000 ma wymiary 183 x 280 x 72 mm). Aby umożli-
wić posiadaczom dwóch stacji ustawienie ich jedna
na drugiej **TOMS 720** ma tylne nóżki przesunięte
nieco do przodu.

Na tylnej ścianie znajduje się wyłącznik zasilania
i gniazdo złącza Centronics. Ponadto wyprowadzo-
ne są tam dwa przewody: zasilający i łączący z
komputerem. Z przodu ponad kieszenią dyskietki
umieszczone zostały cztery przyciski sterujące,
dwucyfrowy wyświetlacz LED oraz dioda sygnalizu-
jąca możliwość zapisu.

Układ mechaniczny stacji jest typowym napędem
dyskowym o pojemności 720 KB, stosowanym w
komputerach IBM. Pracą napędu steruje układ
elektroniczny zbudowany na bazie procesora 8085
i kontrolera dysku WD2797. Program pracy proces-
ora jest zapisany w pamięci ROM 27256 o pojem-
ności 32 KB, znaczną część tej pojemności pochłania
jednak DOS i programy użytkowe. Ponadto sta-
cja ma własną pamięć RAM o pojemności 8 KB.
Komunikację z komputerem poprzez złącze szere-
gowe realizuje CPU, z drukarką zaś poprzez Cen-
tronics zapewnia typowy układ PIA 8255.

Zasilacz jest umieszczony wewnątrz stacji, co
znakomicie zmniejsza liczbę przewodów i pudełek
znajdujących się pod biurkiem. Należy się tylko dzi-
wić, iż producenci innych stacji nie wpadli na ten
pomysł. Odpowiednie miejsce można przecież łat-
wo wygospodarować, minimalnie zwiększając wy-
sokość i długość obudowy.

PRACA

Do współpracy z komputerem niezbędny jest
DOS. **TOMS 720** ma zapisany w pamięci ROM MY-
DOS 4.50. Jest to jeden z dwóch DOS-ów Atari,
które mogą rozpoznać dyskietki o pojemności
większej niż 360 KB (drugim jest SpartaDOS). Wy-
bór ten został podyktowany niemal stuprocentową
zgodnością tego DOS-u z DOS 2.5 i innymi stan-
dardowymi DOS-ami Atari.

Jak już wspominałem, stacja **TOMS 720** rozpo-
znaje siedem formatów dyskietek. Są one następu-
jące:

SS/SD — jednostronny, pojedyncza gęstość (40
ścieżek po 18 sektorów po 128 bajtów =
90 KB).
SS/ED — jednostronny, rozszerzona gęstość (40
ścieżek po 26 sektorów po 128 bajtów =
130 KB).
SS/DD — jednostronny, podwójna gęstość (40
ścieżek po 18 sektorów po 256 bajtów =
180 KB).
SS/iD — jednostronny format IBM S-9 (40 ścieżek
po 9 sektorów po 512 bajtów = 180 KB).



DYSKÓW

TOMS 720

DS/DD — dwustronny, podwójna gęstość (40 ścieżek po 18 sektorów po 256 bajtów = 360 KB),

DS/QD — dwustronny, poczwórna gęstość (80 ścieżek po 18 sektorów po 256 bajtów = 720 KB),

DS/ID — dwustronny format IBM D-9 (40 ścieżek po 18 sektorów po 512 bajtów = 360 KB).

Przesyłanie danych do i z komputera odbywa się w systemie Turbo z prędkością 70000 bodów. W wypadku programów, które nie mogą pracować w Turbo, przewidziana jest standardowa prędkość 19200 bodów lub amerykański standard Ultra Speed — 70000 bodów (tego systemu używa między innymi SpartaDOS). Przyspieszenie transmisji uzyskuje się również dzięki buforowaniu ścieżek, więc przy odczycie nie jest ważny przepływ sektorów na dyskietce. Czasów odczytu dyskietek nie podaję, gdyż dla standardowych formatów są analogiczne jak dla **TOMS MULTI DRIVE**, dla formatów dwustronnych zaś nie ma ich z czym porównać.

DODATKOWE MOŻLIWOŚCI

Informacje o pracy stacji są wyświetlane na dwucyfrowym wyświetlaczu LED. Rodzaj informacji jest wybierany przyciskami, może to być numer stacji i format dyskietki, numer ścieżki lub numer błędu, który wystąpił podczas pracy. W odróżnieniu od LDW 2000 i CA 2001 funkcje wyświetlacza nie zmieniają się samoczynnie po wystąpieniu błędu. Wyświetlacz przekazuje ponadto dwie dodatkowe informacje. Migająca kreska w pierwszej cyfrze oznacza włączenie buforowania ścieżek, natomiast świecący punkt sygnalizuje korzystanie z drugiej strony dyskietki.

Zadania przycisków nie ograniczają się jedynie do sterowania wyświetlaczem — za ich pomocą można bowiem określać również inne parametry pracy stacji (łącznie przyciski mają dziesięć funkcji). Ponieważ trudno je wszystkie spamiętać, a ciągłe zaglądnienie do instrukcji jest kłopotliwe, to na górnej powierzchni stacji została umieszczona naklejka z opisem funkcji przycisków. Należy to uznać za bardzo dobry pomysł i duży plus dla projektantów urządzenia.

Stacja rekompensuje również brak w komputerze złącza Centronics dla drukarki. Stanowi ono integralną część układu elektronicznego stacji, a użytkownik ma możliwość wybrania sposobu translacji (*Normal/Graphic*). Ze względu na odbiegający od standardu ASCII kod końca wiersza, zastosowany w Atari, podczas druku tekstu konieczna jest jego zamiana. Powoduje to jednak błędny wydruk grafiki. Przez przełączenie trybu translacji można więc wyłączyć tę zamianę i dzięki temu uzyskać poprawne odwzorowanie grafiki.

Jedną z najważniejszych zalet stacji **TOMS 720** jest możliwość zapisania na jednej dyskietce aż 720 KB danych. Co jednak zrobić z programami całodyskowymi? Konstruktorzy znaleźli rozwiązanie także tego problemu, i to dwoma sposobami. Pierwszym sposobem jest niezależne formatowanie ścieżek parzystych i nieparzystych, dzięki czemu na jednej dyskietce znajdują się dwie niezależne od siebie — jedną z nich można nawet odczytać na standardowej stacji Atari. Tą metodą można również zapisywać dyskietki zabezpieczone. Drugi sposób polega na umieszczeniu na jednej dyskietce kilku ścieżek o numerze 0, z których każda oznacza początek nowej dyskietki logicznej. Uzyskuje się przez to efekt zapisania na jednej dyskietce kilku dyskietek umieszczonych jedna za drugą, podobnie jak to ma miejsce w przypadku zwykłych plików.

OPROGRAMOWANIE

DOS jest odczytywany ze stacji przy podniesionej dźwigni, więc zbędne jest umieszczanie go na dyskietkach. Rozwiązanie to stało się już zasadą w urządzeniach firmy **TOMS**. Poza DOS-em w stacji zapisane są podstawowe programy użytkowe uruchamiające szybką transmisję danych w różnych standardach, program do kopiowania plików z i na dyskietki w formacie IBM S9 (dwustronne w pojedynczej gęstości — 360 KB) oraz popularny program TRACK COPIER.

Wszystkie programy zapisane w ROM stacji są również dostępne po zamknięciu dźwigni. W takiej sytuacji ROM stacji zgłasza się jako stacja numer 3. Ponieważ na specjalne życzenie nabywcy możliwe jest zapisanie w ROM wybranego przezeń programu, to znacznie skraca się czas uruchamiania systemu do pracy. Dotyczy to oczywiście użytkowników, dla których komputer jest narzędziem, a nie zabawką. Na przykład, eksploatowany przeze mnie egzemplarz ma zapisane tam programy pomocnicze SpartaDOS (TREE, MDUMP, SCOPY i DUPDSK), które nie znalazły się w module SpartaDOS X.

Ubogo — niestety — wygląda pozostałe oprogramowanie stacji: brakuje odpowiednich programów narzędziowych i kopiujących. Programy takie są jednak aktualnie opracowywane i w chwili, gdy ten test dotrze do czytelników, najprawdopodobniej będą już dostępne.

EKSPLLOATACJA

Obsługa stacji **TOMS 720** jest prosta i bardzo wygodna, dzięki łatwemu dostępowi do kieszeni dyskietki i wszystkich elementów sterujących. Początkowo miałem trochę trudności w opanowaniu wszystkich jej funkcji, lecz korzystając z naklejonego opisu szybko je przezwyciężyłem. Ponieważ stacja jest dwustronna, to odpada uciążliwe przekładanie dyskietek na drugą stronę (w formatach dwustronnych).

Podstawową zaletą **TOMS 720** jest dwukrotne zwiększenie pojemności dyskietki. Oczywiście nie może to być byle jaka dyskietka (tzw. „bezfirmitowa”), lecz stosunkowo dobrej jakości — najlepiej 96 tpi, choć wystarcza także 48 tpi renomowanych firm (o dziwo „załapał” się tu również „Stilon” z Gorzowa!). Największe kłopoty sprawiały mi dyskietki „Data Magnetics”, w czym nie ma nic dziwnego, gdyż na kilka ich pudełek trafia się co najmniej jedna, której nie można sformatować nawet w stacji 1050.

Przy „upychaniu” programów na dyskietkę natrafiłem na pewną niedogodność. Po przeszukaniu całej dyskietki podczas wyszukiwania dysków logicznych stacja zatrzymuje się na ostatniej ścieżce i jedynym wyjściem jest jej wyłączenie i ponowne włączenie. Przy częstszym korzystaniu z tej funkcji marzyłoby się, aby wyłącznik zasilania znajdował się z przodu stacji. Innym wyjściem może być samoczynny powrót głowicy do ścieżki zerowej po osiągnięciu końca dyskietki. Trudno jednak określić położenie tego końca, gdyż niektóre napędy mogą tworzyć na dyskietce nawet 86 ścieżek (jest to cecha poszczególnych egzemplarzy, a nie typów).

Wadą jest niewątpliwie istnienie tylko jednego złącza do łączenia z innymi elementami systemu. Zmusza to do przyłączenia stacji jako ostatniego urządzenia w łańcuch i uniemożliwia równoczesne przyłączenie modemu lub magnetofonu. W takich wypadkach producent instaluje (na specjalne życzenie) dodatkowe złącze szeregowo.

Posiadacze kilku stacji dysków zauważą jeszcze jedną wadę **TOMS 720**, a mianowicie niemożność trwałej zmiany numeru stacji. Po włączeniu zasilania ma ona zawsze numer 1 i trzeba to zmieniać za

pomocą przycisków. Naturalnie można numer 2 lub 4 nadać innej stacji i wtedy problem znika.

Do cennych zalet trzeba natomiast zaliczyć umieszczenie podstawowego oprogramowania w pamięci ROM stacji. Dodatkową zaletą jest możliwość umieszczenia tam programu wskazanego przez użytkownika, choć wielkość tego programu jest ograniczona pojemnością pamięci.

PODSUMOWANIE

Stacja **TOMS 720** będzie z pewnością cennym nabytkiem dla każdego posiadacza „małego” Atari. Jej cena (w kwietniu 1991 około 3 mln złotych) przewyższa wprawdzie cenę innych stacji, lecz jest tylko minimalnie wyższa od ceny LDW lub CA 2001 i oddzielnego interfejsu Centronics. Zwiększony koszt zakupu zwróci się szybko dzięki oszczędności na dyskietkach **TOMS 720** potrzebuje ich dwa razy mniej do zapisania tej samej ilości informacji niż inne stacje. Polecam ją więc wszystkim użytkownikom zamiast nie najlepszych (i drogiej) stacji sprzedawanych w Pewexie.

Marek ZACHAR

Z A L E T Y

- * wbudowany zasilacz
- * duża pojemność dyskietki
- * możliwość zapisania kilku programów całodyskowych na jednej dyskietce
- * duża szybkość transmisji
- * wielofunkcyjność
- * wbudowany Centronics
- * oprogramowanie w pamięci ROM
- * możliwość wyboru oprogramowania

W A D Y

- * konieczność wyłączania przy poszukiwaniu dysków logicznych
- * brak ustalania numeru stacji
- * jedno złącze szeregowo Atari



PROJEKTOWANIE PROGRAMU

ANTIC-

Zadaniem zamieszczonego obok programu jest ułatwienie projektowania wyglądu obrazu przy wykorzystaniu zmienionego programu ANTIC-a oraz jego przerwań.

Program umożliwia tworzenie programu ANTIC-a ze wszystkimi udogodnieniami, które udostępnia ANTIC, tzn. w każdej linii można ładować licznik pamięci obrazu, zmieniać zawartość rejestrów koloru oraz włączać przewijanie poziome lub pionowe. Zmiana zawartości rejestrów dokonywana jest poprzez wywołaną w przerwaniu DL procedurę maszynową, która znajduje się na szóstej stronie pamięci. Można ją oczywiście zastąpić inną procedurą, np. zmieniającą adres zestawu znaków, zmieniającą zawartość tylko jednego rejestru, przepisującą adres początku pamięci.

Program generuje procedurę rozpoczynającą się od wiersza 30000 i realizującą zaprojektowany wygląd obrazu, a następnie nagrywa ją na kasety. W celu wykorzystania tej procedury we własnym programie dołączamy ją do tego programu instrukcją **ENTER** i uruchamiamy przez **GOSUB 30000**. Należy jedynie pamiętać o umieszczeniu we własnym programie odpowiednich instrukcji ograniczających dostępność pamięci (**POKE 106,...**).

Jacek TARASIUK

```

HO 1 REM PROJEKTOWANIE DL
KZ 2 REM Jacek Tarasiuk
JA 3 REM (c) 1990, Bajtek
NJ 4 REM
RM 5 GRAPHICS 0:POKE 559,0:GOSUB 3200:GOS
UB 3200:GOSUB 3215:POKE 752,5
6 DIM A$(1),B$(3),PR(300),PR$(300),PAM
(10,2),KOL(30,4):IL=0:IL5=0:PR=0:W5=0:
KONIEC=0:PAM=1:KOLOR=0
CF 7 FOR X=1 TO 30:FOR Y=1 TO 4:KOL(X,Y)=
0:NEXT Y:NEXT X
AR 10 GOSUB 100:GOTO 200
NI 100 POSITION 1,0:?" "
WL 101 POSITION 1,1:?" "1|2|3| 4| 5| 6|
FR 102 POSITION 1,2:?" "
RW 103 POSITION 1,3:?" "2|0|2| 8| 40|40|
CS 104 POSITION 1,10:?" "9|4|2| 4| 80|10|
105 POSITION 1,4:?" "3| -|2|10| 40|40|
GG 106 POSITION 1,11:?" "A|S|4| 4| 80|20|
107 POSITION 1,5:?" "4| -|4| 8| 40|40|
EW 108 POSITION 1,12:?" "B|6|2| 2|160|20|
109 POSITION 1,6:?" "5| -|4|16| 40|40|
DY 110 POSITION 1,13:?" "C| -|2| 1|160|20|
111 POSITION 1,7:?" "6|1|5| 8| 20|20|
BE 112 POSITION 1,14:?" "D|7|4| 2|160|40|
113 POSITION 1,8:?" "7|2|5|16| 20|20|
HG 114 POSITION 1,15:?" "E| -|4| 1|160|40|
115 POSITION 1,9:?" "8|3|4| 8| 40|10|
HZ 116 POSITION 1,16:?" "F|8|2| 1|320|40|
JE 117 POSITION 1,17:?" "
TS 118 POSITION 19,1:?"1-nr.trybu ANTIC-
ON 119 POSITION 19,2:?"2-nr.trybu BASIC-
TF 120 POSITION 19,3:?"3-liczba kolorow
UR 121 POSITION 19,4:?"4-wysokosc lini
CO 122 POSITION 19,5:?"5-ilosc punktow
GQ 123 POSITION 19,6:?"6-ilosc bajtow
EE 124 POSITION 18,7:?"
GP 125 POSITION 18,8:?"
JE 126 POSITION 18,9:?"
WT 127 POSITION 18,10:?"
TI 128 POSITION 18,11:?"
CM 129 POSITION 18,12:?"
AW 130 POSITION 18,13:?"
WN 131 POSITION 18,14:?"
NB 132 POSITION 18,15:?"
TB 133 POSITION 18,16:?"
CC 134 POSITION 18,17:?"
TP 135 POSITION 18,18:?"
AE 136 POSITION 18,19:?"
FM 137 POSITION 18,20:?"
NG 138 POSITION 18,21:?"
YL 139 POSITION 18,22:?"
UV 140 POSITION 18,23:?"
RL 141 POSITION 18,24:?"
WS 142 POSITION 18,25:?"
TV 143 POSITION 18,26:?"
CM 144 POSITION 18,27:?"
PZ 145 POSITION 18,28:?"
AL 146 POSITION 18,29:?"
NH 147 POSITION 18,30:?"
HT 148 POSITION 18,31:?"
QI 149 POSITION 18,32:?"
HO 150 POSITION 18,33:?"
NJ 151 POSITION 18,34:?"

```

```

EE 254 GOSUB 2000:POSITION 19,11:?"Podaj
starszy bajt":POSITION 19,12:?"adres
u pamieci":POSITION 19,14:INPUT B$
UY 255 TRAP 252:POCZ2=VAL(B$):IF POCZ2<0
OR POCZ2>255 THEN 254
TM 257 GOSUB 2030
JB 260 GOSUB 2000:POSITION 19,11:?"Czy t
o juz koniec?":POSITION 19,12:?"list
y? [t/n]":POSITION 19,14:INPUT A$
WD 261 IF A$="T" THEN KONIEC=1
YY 270 IF IL5<0 THEN FOR I=1 TO IL5:PR
R=PR+1:PR(PR)=ROZKAZ:IL=IL+1:IL5=IL5+I
5:IF IL5>224 THEN 290
6F 275 IF IL5<0 THEN NEXT I
GM 280 IF KONIEC=0 THEN 200
JR 290 GOSUB 2000:POSITION 19,11:?"Podaj
mlodszy bajt":POSITION 19,12:?"adres
u DLIST":POSITION 19,14:INPUT B$
SP 291 TRAP 290:POCZ1=VAL(B$):IF POCZ1<0
OR POCZ1>255 THEN 290
GG 292 GOSUB 2000:POSITION 19,11:?"Podaj
starszy bajt":POSITION 19,12:?"adres
u DLIST":POSITION 19,14:INPUT B$
CC 293 TRAP 292:POCZ2=VAL(B$):IF POCZ2<0
OR POCZ2>255 THEN 292
KA 295 PR=PR+1:PR(PR)=65:PR=PR+1:PR(PR)=P
OCZ1:PR=PR+1:PR(PR)=POCZ2
NZ 300 GOSUB 2000:POSITION 19,12:?"Help
- obraz":POSITION 19,13:?"Help - powr
ot"
EF 301 IF PEEK(732)<>17 THEN 301
WA 302 POKE 732,0:POKE 559,0:AD=30720:FOR
I=1 TO PR-2:PR$(I)=CHR$(PR(I)):K=PEEK
(ADR(PR$)+I-1):POKE AD+I-1,K:NEXT I
QR 303 POKE AD+PR-2,0:POKE AD+PR-1,120:PO
KE AD+WSK-1,64:POKE AD+WSK,156
JP 305 FOR I=1 TO KOLOR:POKE 1669+I,KOL(I
,1):POKE 1699+I,KOL(I,2):POKE 1729+I,K
OL(I,3):POKE 1759+I,KOL(I,4):NEXT I
NM 306 POKE 54286,0:POKE 1579,0
IV 310 POKE 1579,KOLOR:POKE 512,0:POKE 51
3,6:POKE 560,0:POKE 561,120:POKE 88,64
:POKE 89,156:POKE 54286,192
RA 311 POKE 559,34
GP 312 IF PEEK(732)<>17 THEN 312
JE 315 GRAPHICS 0:GOSUB 100:POSITION 19,1
1:?"Nagrywasz czy ":POSITION 19,12:?"
projektujesz ":POSITION 19,13
BS 316 ? "jeszcze raz?(N/P)":POSITION 19,
14:INPUT A$
MH 317 IF A$="P" THEN RUN
PT 318 IF A$<"N" THEN 316
AE 320 LINIA=30003
LR 321 GRAPHICS 0:POSITION 2,4:?"LINIA;"
IO 322 "DATA ";KOLOR:?"CONT"
YU 322 GOSUB 150:LINIA=30050
325 GRAPHICS 0:POSITION 2,4:?"30006 D
ATA ";POCZ1+256*POCZ2:?" ";KOLOR:?" ";P
M(1,1)+PAM(1,2)*256:?" ";PR:?"CONT"
TS 326 GOSUB 150
JI 330 FOR I=1 TO 4:LINIA=LINIA+1:GRAPHIC
S 0:POSITION 2,4:?"LINIA;" DATA " ";FOR
J=1 TO KOLOR
RP 332 ? KOL(J,I):IF J<KOLOR THEN ? " ",
J
TK 334 NEXT J:?" ":?"CONT"
TR 335 GOSUB 150
GR 336 NEXT I
ON 337 GRAPHICS 0:POSITION 2,4:LINIA=LINI
A+1:?"LINIA;" DATA " ";FOR I=1 TO PAM-1
:?" PAM(I,1)+256*PAM(I,2);
JI 338 IF I<PAM-1 THEN ? " ",I;
HA 339 NEXT I
ZU 340 ? "REM ADRESY POCZ.PAMIECI EKRANU
":?"CONT"
LA 341 W=-30:GOSUB 150:LINIA=30100
VZ 343 LINIA=LINIA+1:W=W+30:GRAPHICS 0:PO
SITION 2,4:?"LINIA;" DATA " ";
DG 345 IF PR>30 THEN FOR I=1 TO 30:?"PR(I
+WS):IF I<30 THEN ? " ",I;
GM 346 IF PR>30 THEN NEXT I:?" ":?"CONT
"
AN 347 IF PR>30 THEN GOSUB 150
UC 348 PR=PR-30:IF PR>30 THEN 343
OU 350 W=W+30:LINIA=LINIA+1:GRAPHICS 0:PO
SITION 2,4:?"LINIA;" DATA " ";
FI 351 IF PR>0 THEN FOR I=1 TO PR:?"PR(I+
WS):IF I<PR THEN ? " ",I;
AE 352 IF PR>0 THEN NEXT I:?" ":?"CONT"
SU 353 IF PR>0 THEN GOSUB 150
FM 360 LIST "C:",30000,30160
OT 365 END
OL 1000 STOP
DE 2000 FOR I=11 TO 16:POSITION 19,I:?"
":NEXT I:RETURN
PJ 2010 IF A$="0" THEN ROZKAZ=0:W5=1:IS=1

```

```

UD 2011 IF A$="2" THEN ROZKAZ=2:W5=1:IS=2
OK 2012 IF A$="3" THEN ROZKAZ=3:W5=1:IS=1
GJ 2013 IF A$="4" THEN ROZKAZ=4:W5=1:IS=8
AW 2014 IF A$="5" THEN ROZKAZ=5:W5=1:IS=1
6
JU 2015 IF A$="6" THEN ROZKAZ=6:W5=1:IS=8
EI 2016 IF A$="7" THEN ROZKAZ=7:W5=1:IS=1
6
ML 2018 IF A$="8" THEN ROZKAZ=8:W5=1:IS=8
JI 2019 IF A$="9" THEN ROZKAZ=9:W5=1:IS=4
EO 2020 IF A$="A" THEN ROZKAZ=10:W5=1:IS=
4
DI 2021 IF A$="B" THEN ROZKAZ=11:W5=1:IS=
2
DP 2022 IF A$="C" THEN ROZKAZ=12:W5=1:IS=
1
GW 2023 IF A$="D" THEN ROZKAZ=13:W5=1:IS=
2
HD 2024 IF A$="E" THEN ROZKAZ=14:W5=1:IS=
1
IX 2025 IF A$="F" THEN ROZKAZ=15:W5=1:IS=
1
BE 2026 RETURN
DU 2030 PR=PR+1:LAD=PR:PR(PR)=ROZKAZ+64:P
R=PR+1:PR(PR)=POCZ1:PR=PR+1:PR(PR)=POC
Z2:IL=IL+1:IL5=IL5+15
EB 2032 IL5C=IL5C-1:PAM(PAM,1)=POCZ1:PA
M(PAM,2)=POCZ2:PAM=PAM+1:IF PAM=2 THEN
WSK=PR-1
AZ 2034 RETURN
VA 2040 IF IL5C<0 THEN PR=PR+1:PR(PR)=11
2:IL5=IL5+8:IL5C=IL5C-8:GOTO 2040
KA 2042 ROZKAZ=ROZKAZ+16*(IL5C-1):IL5=0
5:IL5C=PR-PR+1:PR(PR)=ROZKAZ:IL5C=IL
5:RETURN
YG 2045 IF LAD=PR-2 THEN PR(PR-2)=PR(PR-2
)+128:GOTO 2050
HI 2048 PR(PR)=PR(PR)+128
LO 2050 GOSUB 2000:POSITION 19,11:?"Klaw
iszami < i > ":POSITION 19,12:?"Zmie
niasz kolor"
UC 2051 POSITION 19,14:?"RETURN - akcept
acja"
II 2052 KOLOR=KOLOR+1:FOR I=1 TO 4:K=0
FS 2053 IF I=4 THEN POSITION 19,13:?"COL
BAK":GOTO 2055
XE 2054 POSITION 19,13:?"COLPF";I=1
YR 2055 GOSUB 3000
XF 2056 IF A$="+" THEN K=K-1
VA 2057 IF A$="*" THEN K=K+1
FJ 2058 IF K=-1 THEN K=255
QU 2059 IF K=256 THEN K=0
GW 2060 W=I:IF I=4 THEN W=0
GB 2061 POSITION W*10+4,19:?"X;" ":POKE
1670+(I-1)*30,K
QO 2062 IF K<>155 THEN 2055
SI 2063 KOL(KOLOR,I)=K
FY 2064 NEXT I
BM 2065 RETURN
QO 3000 OPEN #1,4,0,"K":GET #1,KX:A$=CHR
$(KX):CLOSE #1:RETURN
ZG 3200 FOR I=1536 TO 1590:READ A:POKE I,
A:NEXT I:RETURN
WC 3201 DATA 72,138,72,238,255,6,174,255,
6,189,133,6,72,189,163,6,72,189,193,6,
72
ID 3202 DATA 189,223,6,141,10,212,141,26,
208,104,141,24,208,104,141,23,208,104,
141,22,208,224
KJ 3203 DATA 2
FG 3204 DATA 208,5,169,0,141,255,6,104,17
0,104,64
FI 3209 FOR I=39968 TO 39999:READ A:POKE
I,A:NEXT I:POKE 1670,4:POKE 1700,8:POK
E 1730,12:POSITION 12,20
FS 3210 POKE 1760,0:?"** PROJEKTOWANIE P
ROGRAMOW ANTIC-A **":POKE 1791,0:RETUR
N
NR 3211 DATA 112,112,112,112,66,64,156,2,
2,2,2,2,2,2,2,2,2,2,2,2,2,2,130,
136,2,112,65,32,156
PE 3215 FOR I=40800 TO 40889:READ A:POKE
I,A:NEXT I:POKE 1671,4:POKE 1701,0:POK
E 1731,15:POKE 1761,0:POKE 54286,192
BG 3216 RETURN
AQ 3217 DATA 0,0,5,85,85,170,170,175,255,
255
FM 30000 FOR I=1536 TO 1590:READ A:POKE I
,A:NEXT I
TL 30001 DATA 72,138,72,238,255,6,174,255
,6,189,133,6,72,189,163,6,72,189,193,6
,72
YY 30002 DATA 189,223,6,141,10,212,141,26
,208,104,141,24,208,104,141,23,208,104
,141,22,208,224

```

TYLKO DLA ZAAWANSOWANYCH (2)

Problem uzyskania polskich liter na Atari poruszany był na łamach niemal wszystkich czasopism komputerowych w naszym kraju.

Niestety prezentowane rozwiązania zawsze obarczone były pewnymi, dość kłopotliwymi wadami:

- generator znaków znajduje się w obszarze pamięci Basica;
- po każdej instrukcji GRAPHICS trzeba przywracać jego adres;
- RESET zwykle usuwa generator z pamięci, a procedura uniemożliwiająca jego zniszczenie zajmuje często wykorzystywaną szóstą stronę pamięci.

Spróbujmy pozbyć się tych wad. Aby tego dokonać, musimy zagłębić się w dziedzinę zarezerwowaną tylko dla fachowców — zmiany w systemie operacyjnym!

Powszechnie wiadomo, że Atari ma co najmniej 64 KB pamięci RAM, mało kto wie jednak, jak je wykorzystać. W „Bajtku” zamieszczono kilka programów wykorzystujących tę właściwość — koronnymi przykładami są procedury Speedtrans i program Multikopier.

Podział pamięci Atari jest następujący:

0—40959 (\$0000—\$9FFF) — tu zawsze znajduje się pamięć RAM, w szczególnym przypadku w obszarze \$5000—\$57FF bywa odwzorowywany program testujący z ROM;

40960—49151 (\$A000—\$BFFF) — tu jest RAM, jeśli przy uruchamianiu komputera wcisnęliśmy <OPTION>, a w przeciwnym razie ROM z interpreterem Basica;

49152—53247 (\$C000—\$CFFF) — normalnie znajduje się tu pamięć ROM z systemem operacyjnym, możliwe jest jednak zastąpienie jej pamięcią RAM;

53248—55295 (\$D000—\$D7FF) — RAM, której nie można użyć, gdyż jest to również obszar rejestrów sprzętowych, ponadto znajduje się tu program testujący odwzorowywany w RAM;

55296—65535 (\$D800—\$FFFF) normalnie obszar ROM (2 KB procedur matematycznych i 8 KB systemu operacyjnego), ewentualnie RAM.

Uważne prześledzenie tego zestawienia uwidacznia fakt, że każdemu obszarowi zajętemu przez ROM odpowiada identyczny wielkością obszar RAM. Powstaje więc pytanie: jak odłączyć ROM? Sprawa jest nadzwyczaj prosta. Sterowanie przyłączaniem i odłączaniem układów odbywa się przez rejestr PORTB układu PIA — 54017 (\$D301). Bit 7 odsłania program testujący, bity 2—6 są wykorzystywane do przełączania dodatkowych banków pamięci w 130XE, bit 1 — jeśli jest zero, to podłączony jest Basic, bit 0 — jeżeli 1 — podłączony jest ROM z systemem operacyjnym. Zaraz po włączeniu komputera i zgłoszeniu Basica wartość tego bajtu wynosi 253.

Do czego mogą się przydać te dodatkowe 24 KB pamięci RAM? Na przykład jako podręczna pamięć danych lub do umieszczenia w RAM zawartości ROM, co umożliwi nam dokonywanie zmian w systemie. Taką czynność wykonuje program 1, korzystający z procedury przesyłającej z Multikopiera. Można go wykorzystać we własnych eksperymentach z systemem (zajęcie pasjonujące, jeszcze o nim wspomnimy).

Skoro mamy już ROM w RAM, to nic nie przeszkadza w umieszczeniu tam polskich liter lub nawet nowego generatora znaków, a mamy przecież dostępne dwa miejsca: \$CC00—\$CFFF — znaki międzynarodowe i \$E000—\$E3FF — znaki standardowe. Miejsca aż nadto. Kto zechce tego dokonać z poziomu Basica, na pewno da sobie radę sam. Ja natomiast proponuję program 2 zapisujący na dyskietce lub kasecie plik, który realizuje przepisanie ROM do RAM i wprowadzenie polskich liter w języku maszynowym (program źródłowy jest pokazany w listingu 3). Polskie litery są umieszczone następująco (wszystkie z wcisniętym <CONTROL>):

a	—	A
Ą	—	Q
ć	—	C
Ć	—	D
ę	—	E
Ę	—	R
ł	—	L
Ł	—	;

ń	—	N
Ń	—	M
ó	—	O
Ó	—	P
ś	—	S
Ś	—	T
ż	—	W
Ż	—	X
ź	—	Y
Ź	—	Z

Wydawać by się więc mogło, że mamy oto złoty środek usuwający dolegliwości. Niestety muszę was nieco rozczarować. Skutkiem ubocznym mojego rozwiązania jest złe działanie instrukcji BYE (proszę spróbować) oraz nie najlepsza współpraca z DOS-ami. Najczęstszym objawem są kłopoty z powrotem do Basica. Czasem wystarcza nacisnąć <RESET> i wpisać POKE 54017,254. Ze względu na znaczną liczbę DOS-ów trudno jest podać jeden uniwersalny schemat postępowania w takim wypadku. Każdy użytkownik musi więc wypracować go sobie sam metodą prób i błędów. Sądzę, że te drobne mankamenty są mniejszym złem w porównaniu z pierwotnymi wadami. Życzę dużo pożytku z polskich liter, a za dwa miesiące: o magnetofonie od podszewki.

Andrzej ZALEWSKI

LISTING 1

```

UY 0 REM TYLKO DLA ZAAWANSOWANYCH (2)
KT 1 REM PROGRAM 1
LU 2 REM Andrzej Zalewski
DV 3 REM (c) 1991, Sp. Bajtek
KR 10 CLR :FOR N=1536 TO 1593:READ A:POKE
    N,A:S=S+A:NEXT N:AD=40960:? CHR$(125)
    ;CHR$(127);"Przepisuje ROM do RAM !"
JW 20 POKE 1556,253:X=USR(1537,AD,24576)
PV 30 POKE 1556,254:X=USR(1537,24576,AD)
OA 40 AD=AD+256:IF AD>=53248 AND AD<55000
    THEN AD=55296
IL 50 IF AD=65535 THEN 70
JF 60 POSITION 0,0:? AD:GOTO 20
HF 70 ? CHR$(29);CHR$(29);"Uwaga!":? "Sys
    tem operacyjny znajduje sie"
YL 80 ? "w pamieci RAM! Ostroznie z instr
    ukcja POKE!!!":POKE 54017,254
GS 90 DATA 64,104,104,141,44,6,104,141,43
    ,6,104,141,47,6,104,141,46,6,120,169,2
    54,141,1,211,169,0,141
ZB 100 DATA 250,255,141,254,255,169,6,141
    ,251,255,141,255,255,162,0,189,120,230
    ,157,64,156,232,208,247,88,169
FL 110 DATA 253,141,1,211,96

```

LISTING 2

```

UY 0 REM TYLKO DLA ZAAWANSOWANYCH (2)
LI 1 REM PROGRAM 2
LU 2 REM Andrzej Zalewski
DV 3 REM (c) 1991, Sp. Bajtek
RE 10 DIM A$(380),B$(1):GRAPHICS 0
XV 20 FOR N=1 TO 319:READ A:A$(N)=CHR$(A)
    :NEXT N
CS 30 ? "Kaseta /K/, czy dysk /D/":INPUT
    B$
QO 40 IF B$="K" THEN A$(1,6)="♦♦♦♦♦":OPE
    N #1,8,128,"C":GOTO 60
EE 50 OPEN #1,8,0,"D:POLZN.COM"
KX 60 ? #1:A$:CLOSE #1:END
JP 100 DATA 255,255,0,80,50,81,160,159,16
    9,0,141,68,2,141,124,80,141,23,80,200,
    140,125,80,140,24
IQ 110 DATA 80,162,0,189,0,160,157,0,96,2
    32,208,247,32,96,80,192,207,208,2,160,
    215,192,255,240,2
BH 120 DATA 208,223,169,254,141,1,211,162
    ,0,138,10,168,185,137,80,133,112,185,2
    1,81,133,116,200,185,137
KR 130 DATA 80,133,113,185,21,81,133,117,
    160,0,177,116,145,112,200,192,8,208,24
    7,232,224,14,208,216,76
GX 140 DATA 0,160,120,169,254,141,1,211,1
    69,136,141,250,255,141,254,255,169,80,
    141,251,255,141,255,255,162
JG 150 DATA 0,189,0,96,157,0,160,232,208,
    247,88,169,253,141,1,211,96,64,8,226,2
    4,226,32,226,40
VU 160 DATA 226,96,226,104,226,112,226,12
    0,226,152,226,160,226,184,226,192,226,
    200,226,208,226,0,0,60,6
II 170 DATA 62,102,62,7,12,24,60,96,96,96
    ,60,0,12,24,60,102,96,96,102,60,0,0,60
    ,102,126
KA 180 DATA 96,60,6,0,48,54,60,120,240,63
    ,0,0,56,30,60,120,24,60,0,12,24,124,10
    2,102,102
XB 190 DATA 102,0,12,24,60,102,102,102,60
    ,0,12,24,62,96,60,6,124,0,12,24,60,96,
    60,6,6
UG 250 DATA 60,12,24,126,12,24,48,126,0,2
    4,0,126,12,24,48,126,0,12,24,126,12,24
    ,48,96,126
BA 260 DATA 24,0,126,12,24,48,96,126,165,
    80,173,80,181,80,189,80,197,80,205,80,
    213,80,221,80,229
CQ 290 DATA 80,237,80,245,80,253,80,5,81,
    13,81,0,0,224,2,225,2,0,80

```



LISTING 3

```

0100 ; Polskie znaki w RAM pod ROM
0110 ; Andrzej Zalewski
0120 ; (c) 1991, Sp. Bajtek
0130 ;
0140 COLDST = $0244
0150 TRAM = $6000
0160 ROM = $A000
0170 PORTB = $D301
0180 NMIVEC = $FFFA
0190 IRQVEC = $FFFE
0200 ;
0210 * = $5000
0220 ;
0230 ;przygotowanie
0240 LDY #$9F
0250 LDA #0
0260 STA COLDST
0270 STA GDZIE+1
0280 STA PO+1
0290 ;
0300 ;przenosimy ROM do RAM - $6000
0310 ;RAM do RAM pod ROM
0320 TRS
0330 INY
0340 STY GDZIE+2
0350 STY PO+2
0360 LDX #0
0370 PO LDA ROM,X
0380 STA TRAM,X
0390 INX
0400 BNE PO
0410 JSR TRBYT
0420 CPY #$CF
0430 BNE CNT
0440 LDY #$D7
0450 CNT CPY #$FF
0460 BEQ EXIT
0470 BNE TRS
0480 EXIT LDA #$FE
0490 STA PORTB
0500 ;
0510 LDX #0
0520 WW TXA
0530 ASL A
0540 TAY
0550 LDA ADRESY,Y
0560 STA $70
0570 LDA TABL,Y
0580 STA $74
0590 INY
0600 LDA ADRESY,Y
0610 STA $71
0620 LDA TABL,Y
0630 STA $75
0640 LDY #0
0650 QQ LDA ($74),Y
0660 STA ($70),Y
0670 INY
0680 CPY #8
0690 BNE QQ
0700 INX
0710 CPX #14
0720 BNE WW
0730 RUNBAS JMP ROM
0740 ;
0750 TRBYT SEI
0760 LDA #$FE
0770 STA PORTB
0780 LDA # <RTI
0790 STA NMIVEC
0800 STA IRQVEC
0810 LDA # >RTI
0820 STA NMIVEC+1
0830 STA IRQVEC+1
0840 LDX #0
0850 SKAD LDA TRAM,X
0860 GDZIE STA ROM,X
0870 INX
0880 BNE SKAD
0890 CLI
0900 LDA #253
0910 STA PORTB
0920 RTS
0930 RTI RTI
0940 ;
0950 ADRESY .WORD 57864,57880,57888
0960 .WORD 57896,57952,57960,57968
0970 .WORD 57976,58008,58016,58040
0980 .WORD 58048,58056,58064
0990 ;
1000 L1 .BYTE 0,0,60,6
1010 .BYTE 62,102,62,7
1020 L2 .BYTE 12,24,60,96
1030 .BYTE 96,96,60,0
1040 L3 .BYTE 12,24,60,102
1050 .BYTE 96,96,102,60
1060 L4 .BYTE 0,0,60,102
1070 .BYTE 126,96,60,6
1080 L5 .BYTE 0,48,54,60
1090 .BYTE 120,240,63,0
1100 L6 .BYTE 0,56,30,60
1110 .BYTE 120,24,60,0
1120 L7 .BYTE 12,24,124,102
1130 .BYTE 102,102,102,0
1140 L8 .BYTE 12,24,60,102
1150 .BYTE 102,102,60,0
1160 L9 .BYTE 12,24,62,96
1170 .BYTE 60,6,124,0
1180 L10 .BYTE 12,24,60,96
1190 .BYTE 60,6,6,60
1200 L11 .BYTE 12,24,126,12
1210 .BYTE 24,48,126,0
1220 L12 .BYTE 24,0,126,12
1230 .BYTE 24,48,126,0
1240 L13 .BYTE 12,24,126,12
1250 .BYTE 24,48,96,126
1260 L14 .BYTE 24,0,126,12
1270 .BYTE 24,48,96,126
1280 TABL .WORD L1,L2,L3,L4,L5
1290 .WORD L6,L7,L8,L9,L10
1300 .WORD L11,L12,L13,L14

```

Wielu posiadaczy stacji dysków po okresie zachwyty (spowodowanego możliwością rezygnacji z magnetofonu) stwierdza, że ich urządzenie mogłoby być jeszcze lepsze. Dla nich właśnie liczne firmy opracowały modyfikacje (systemy) instalowane w stacjach i rozszerzające ich możliwości.



SYSTEMY

INSTALOWANE W STACJACH DYSKÓW

Najpopularniejsze z tych systemów opiszę w cyklu artykułów. Każdy z omawianych systemów (poza podstawowym instalowanym fabrycznie) zwiększa zakres eksploatacji zarówno stacji dysków, jak i dyskietek. Dodatkowo przy opisie każdego z systemów będziecie mogli przeczytać o najważniejszym oprogramowaniu przeznaczonym specjalnie dla niego. Zakładam, że po przeczytaniu całego cyklu będziecie mieli ułatwiony wybór właściwego dla swoich potrzeb i możliwości systemu lub zestawu systemów (osobiście posiadam stację Atari 1050 z zainstalowanymi pięcioma systemami, jednak przeciętnemu użytkownikowi taka liczba nie jest potrzebna).

Systemy będą opisane w sposób skondensowany tak, aby uwiarygodnić ich najważniejsze cechy. Pełny opis jest zawarty w dostarczanej przez producenta instrukcji obsługi każdego z tych systemów, na przykład instrukcja SPEEDY 1050 liczy ponad 100 stron formatu A4. Aby lepiej uwiarygodnić możliwości poszczególnych systemów i różnice między nimi, najpierw zostanie opisana podstawowa wersja stacji Atari 1050. Dla większej przejrzystości w opisach będę posługiwał się następującymi symbolami literowymi dla określenia gęstości zapisu:

S — gęstość pojedyncza — *SINGLE* (18 sektorów po 128 bajtów) — wszystkie DOS-y zgodne z DOS 2.0s;

E — gęstość rozszerzona — *ENHANCED* (26 sektorów po 128 bajtów) — wszystkie DOS-y zgodne z DOS 2.5 (znacznik VTOC — 02);

M — gęstość średnia — *MEDIUM* (26 sektorów po 128

bajtów) — DOS II/+, TOP-DOS, MYDOS (znacznik VTOC — 03);

D — gęstość podwójna — *DOUBLE* (18 sektorów po 256 bajtów) — DOS-y pracujące w gęstości tzw. *true double* (prawie wszystkie poza DOS 2.0s, DOS 2.5, DOS 2.6, DOS 3.0);

I — gęstość IBM S9 (9 sektorów po 512 bajtów) — IBM-DOS v.2.0, IBMATARI v.2.2, HACKER KING III;

A — gęstość APPLE (4 sektory po 1024 bajty) — HACKER KING III.

Przy każdym systemie będzie także podana maksymalna prędkość transmisji (wg instrukcji producenta systemu). Poza nazwą firmową podane będą również inne nazwy, pod jakimi może ten system występować (zwracam przy tym uwagę, że instalowanie systemu przez osoby nie posiadające licencji producenta oryginału jest zwykłym złodziejstwem).

W systemach instalowanych w stacjach dysków spotykamy dwa rodzaje „przyspieszczy” transmisji: tryb stosowany w rozszerzeniu US DUBLER, zwany trybem US lub amerykańskim, oraz tryb zastosowany w rozszerzeniu TURBO 1050 przez B. Engla (bazujący na przerwaniu \$C030), zwany trybem TURBO, TOP lub prędkością Engla. Tryb TURBO nie jest zgodny z trybem US. W wypadku uruchamiania programu na niewłaściwym dla niego systemie może wystąpić blokada komputera.

W opisach poszczególnych systemów Czytelnik spotka się z pojęciami stosowanymi przez producentów w oryginalnych opisach. I tak np. prędkość transmisji jest obliczana za pomocą specjalnych programów (*Benchmark*), które stosując ten sam algorytm dla obliczania prędkości dają miarodajny wynik bez względu na rodzaj systemu zainstalowanego w stacji. Prędkość ta podawana jest w bodach (bod — w transmisji danych jednostka prędkości modulacji równoważnej przekazaniu 1 bitu w ciągu 1 sekundy — nazwa pochodzi od nazwiska J.M.E. Baudota). Dlatego też spotykana czasem w ogłoszeniach krajowych rzemieślników prędkość podawana w kB lub kb (mniemam, iż jest to określenie kilobity lub kilobody) jest czasem zwykłym naciąganiem klientów na superprędką stację dysków. Przy określaniu możliwości kopiowania będą podawane nazwy sektorów w zależności od sposobu zapisu (**DUPLICATE, SKEW, PHAZED, PHANTOM, MISSING, BAD, CRC, RNF, WEAK** itd.). Dla początkującego użytkownika zapewne nie będzie to zrozumiałe, dlatego postaram się wyjaśnić to w oddzielnym artykule.

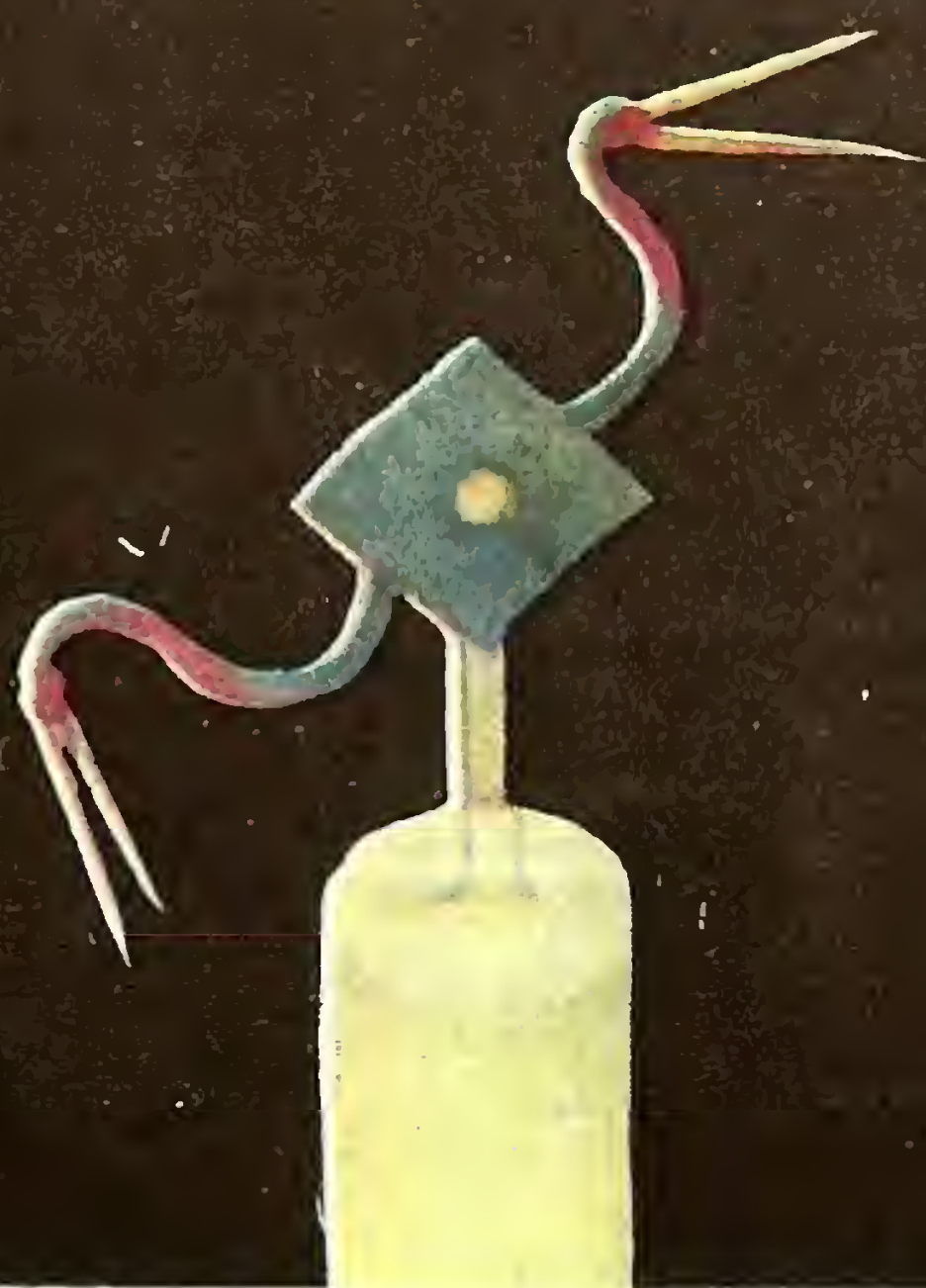
Zbigniew GRADOMSKI

ACTION! BEZ ACTION!



Wielokrotnie — zarówno w opisie **Action!**, jak i w różnych artykułach w „Bajtku” — pisałem, że program napisany w **Action!** i skompilowany nie działa, jeżeli sam **Action!** nie znajduje się w pamięci komputera. Wynika to z faktu korzystania przez skompilowany program z procedur, które znajdują się w module języka. Wspominałem jednak, iż możliwe jest uruchomienie programu bez języka, jeśli tylko spełnionych zostanie kilka warunków.

Język **Action!** powstał w firmie OSS (obecnie jest ona połączona z ICD). Dla programistów, którzy chcą sprzedawać swoje programy, został również napisany specjalny program o nazwie „OSS Runtime Library” (biblioteka uruchamiająca). Po jego dołączeniu (za pomocą dyrektywy **INCLUDE**) otrzymamy w wyniku kompilacji program, który działa nawet wtedy, gdy **Action!** nie został uprzednio umieszczony w pamięci. Posiadacz takiego programu ma kłopot „z głowy”.



Istnieje jednak możliwość uruchomienia programu w **Action!** bez biblioteki procedur. W tym celu trzeba podczas pisania programu przestrzegać czterech następujących zasad:

a) Nie wolno wykorzystywać ŻADNYCH procedur i funkcji bibliotecznych **Action!**.

b) Jako parametry do zdefiniowanych w programie procedur można przekazywać nie więcej niż 3 bajty. Trzeba przy tym pamiętać, że jedna zmienna typu CARD lub INT zawiera już dwa bajty (zmienna typu BYTE ma oczywiście jeden bajt). Jeżeli potrzeba więcej parametrów, to muszą być one zdefiniowane poza procedurą — jako zmienne globalne.

c) Nie wolno używać mnożenia ani dzielenia.

d) Operatory przesunięcia (RSH i LSH) mogą być stosowane tylko z argumentami typu BYTE. Nie wolno wykonywać operacji przesunięcia na zmiennych typu CARD i INT.

Program napisany zgodnie z powyższymi zasadami po skompilowaniu będzie działał niezależnie od obecności języka w pamięci komputera. Bardzo proste, prawda? Ale jak będzie wyglądał program bez GRAPHICS, PRINT itd.? Czy nie ma innego sposobu?

BIBLIOTEKA MODUŁU

Jak wiesz (lub też nie), obszar pamięci zajmowany przez moduł **Action!** rozciąga się od adresu \$A000 do \$C000. Znajdują się tam również procedury w języku maszynowym realizujące procedury i funkcje **Action!**, jak np. PRINT, INPUT. Najprościej byłoby zapisać zawartość tego obszaru pamięci na dyskietce (funkcja DOS-u "K": ACTION.LIB,A000,BFFF). Naturalnie można tak zrobić. Pojawia się jednak następujący problem: obszar pamięci A000—C000 nie jest rezerwowany przy uruchamianiu komputera z wciśniętym klawiszem <OPTION> (znajduje się tam między innymi pamięć obrazu). Dlatego też przed odczytem procedur **Action!** trzeba zmienić w komórce 106 (numer najwyższej będącej do dyspozycji strony pamięci) zawartość ze 192 na 160 i otworzyć ponownie kanał graficzny (np. kanał 0). Jest to wystarczająco skomplikowane, aby zniechęcić przeciętnego użytkownika.

WŁASNA BIBLIOTEKA

Czy nie ma z tej sytuacji wyjścia? Przypomnij sobie, że zdefiniowanie procedury, której nazwa była już wcześniej użyta, powoduje skasowanie poprzedniej definicji. Na przykład zdefiniowanie procedury **Open()** spowoduje, że wszystkie odwołania będą wywoływały naszą procedurę, zamiast procedury bibliotecznej znajdującej się w **Action!**. Wystarczy więc zdefiniować własną bibliotekę procedur **Action!** i zapisać ją pod nazwą "LIB.ACT". Teraz na początku tworzonego programu piszemy:

```
INCLUDE "LIB.ACT"
```

a następnie:

```
MODULE
```

po czym umieszczamy definicje zmiennych globalnych występujących w programie oraz jego procedury i funkcje.

Naturalnie można również plik LIB.ACT wczytać bezpośrednio do edytora **Action!** i dopisać do niego pozostałą część programu.

UWAGA: Plik "LIB.ACT" znajdujący się na dyskietce „Moje Atari 5” nie jest gotową biblioteką, lecz tylko zbiorem zamieszczonych poniżej procedur.

Proste procedury i funkcje

Najprostsze do zastąpienia są (jak zwykle) te procedury, które są rzadko używane lub nawet zbędne. Należą do nich przede wszystkim funkcje manipulatorów, których działanie polega na odczytywaniu zawartości określonych komórek pamięci. Można je zastąpić zestawem następujących funkcji:

```
; funkcje manipulatorow
```

```
BYTE FUNC Stick(BYTE port)
  BYTE ARRAY joy(2)=$278
  RETURN(joy(port))
```

```
BYTE FUNC Strig(BYTE port)
  BYTE ARRAY trig(2)=$284
  RETURN(trig(port))
```

```
BYTE FUNC Paddle(BYTE port)
  BYTE ARRAY pad(4)=$270
  RETURN(pad(port))
```

```
BYTE FUNC Ptrig(BYTE port)
  BYTE ARRAY ptrg(4)=$27C
  RETURN(ptrg(port))
```

Jest to jednak zupełnie zbędne, gdyż zamiast tego wystarczy na początku programu zdefiniować cztery tablice:

```
BYTE ARRAY joy(2)=$278, trig(2)=$284,
  pad(4)=$270, ptrg(4)=$27C
```

Równie łatwa jest do zdefiniowania procedura **Position()**, która podane wartości umieszcza w odpowiednich rejestrach. Niejako przy okazji można także zdefiniować dodatkową procedurę **TextPosition()**, która umożliwi sterowanie położeniem kursora w oknie tekstowym, a której oryginalny **Action!** nie posiada.

```
PROC Position(CARD col BYTE row)
  BYTE crsrow=$54
  CARD crscol=$55
  crsrow=row
  crscol=col
  RETURN
```

```
PROC TextPosition(CARD col BYTE row)
  BYTE txtrow=$290
  CARD txtcol=$291
  txtrow=row
  txtcol=col
  RETURN
```

Zamiast wywoływać takie procedury można wszakże bezpośrednio umieszczać żądane wartości w tych rejestrach. Dodatkową zaletą takiego rozwiązania jest możliwość zmieniania tylko jednej współrzędnej kursora. W tym celu trzeba na początku zdefiniować zmienne:

```
BYTE crsrow=$54,txtrow=$290
CARD crscol=$55,txtcol=$291
```

Zbliżona jest także konstrukcja służącej do zmiany kolorów procedury **SetColor()**, która zamiast odrębnych zmiennych zawiera pięcioelementową tablicę:

```
PROC SetColor(BYTE reg,hue,lum)
  BYTE ARRAY col(5)=$2C4
  col(reg)=hue LSH 4 + lum
  RETURN
```

Tablica ta może być zdefiniowana na początku programu jako zmienna globalna:

```
BYTE ARRAY col(5)=$2C4
```

Wybór koloru w programie następuje wtedy przez wykonanie instrukcji przypisania, w której zamiast zmiennych można umieścić również stałe wartości:

```
col(reg)=hue LSH 4 + lum
```

Ostatnią w tej grupie, choć nieco trudniejszą w realizacji, jest funkcja **Rand()**, która zwraca liczbę losową o wartości nie przekraczającej wartości parametrem procedury jest zero, to zwracana jest po prostu zawartość tego rejestru. Dla innych parametrów procedury jest zero, to zwracana jest po prostu zawartość tego rejestru. Dla innych parametrów zwracana wartość losowa musi być resztą z dzielenia zawartości rejestru przez parametr. Ponieważ dzielenie jest niedozwolone, stosujemy odejmowanie w pętli WHILE.

```
; procedura losowa
```

```
BYTE FUNC Rand(BYTE range)
  BYTE rnd=$D20A, i
  IF range=0 THEN
    RETURN(rnd)
  ELSE
    i=rnd
  FI
  WHILE i>=range
    DO
      i=i-range
    OD
  RETURN(i)
```

Procedury dźwiękowe

W **Action!** istnieją dwie procedury sterujące tworzeniem dźwięku: **Sound()** i **SndRst()**. Druga z nich jest bardzo łatwa do realizacji, natomiast znacznie gorzej jest z pierwszą. Wymaga ona bowiem czterech bajtów parametrów. Ponieważ zwykle generatorem dźwięku steruje się pojedynczo, to zamiast jednej procedury **Sound()** napiszmy cztery — osobno dla każdego generatora. Cały zestaw będzie następujący:

```
; procedury dźwiękowe
```

```
PROC SndRst()
  BYTE audctl=$D208,skctl=$D20F
  audctl=0
  skctl=3
  RETURN
```

```
PROC Sound0(BYTE pit,dis,vol)
  BYTE audf=$D200,audc=$D201
  audf=pit
  audc=dis LSH 4 + vol
  RETURN
```

```
PROC Sound1(BYTE pit,dis,vol)
  BYTE audf=$D202,audc=$D203
  audf=pit
  audc=dis LSH 4 + vol
  RETURN
```

```
PROC Sound2(BYTE pit,dis,vol)
  BYTE audf=$D204,audc=$D205
  audf=pit
  audc=dis LSH 4 + vol
  RETURN
```

```
PROC Sound3(BYTE pit,dis,vol)
  BYTE audf=$D206,audc=$D207
  audf=pit
  audc=dis LSH 4 + vol
  RETURN
```

Jeżeli w pisany program konieczne będzie uzależnienie generatorów od siebie, to można to uzyskać zmieniając parametry przekazywane do procedury lub deklarując dodatkową zmienną globalną, która określi numer generatora.

Procedury wejścia/wyjścia

Po tej prostej wprawce przejdźmy do jednej z trudniejszych (i ważniejszych) rzeczy: procedur obsługujących komunikację komputera z urządzeniami zewnętrznymi. Do ich definiowania wykorzystamy systemową procedurę **CIO**, która musi być następująco zadeklarowana na początku programu:

```
; główna procedura wejścia/wyjścia
```

```
PROC CIO=$E456(BYTE acc,xreg)
```

Najprostszą procedurą I/O (*Input/Output* — wejście/wyjście) jest **Close()**. Ponieważ dla zamknięcia kanału trzeba tylko wpisać odpowiedni rozkaz do bloku sterowania I/O, to narzuca się następujące rozwiązanie:

```
PROC Close(BYTE chn)
  BYTE iccmd=$342+(chn LSH 4)
  iccmd=$C
  CIO(0,chn LSH 4)
  RETURN
```

Niestety, deklaracja zmiennej musi być wartością stałą. Zastosujmy więc rozwiązanie podobne jak w procedurach dźwiękowych. Nie trzeba oczywiście definiować ośmiu procedur, lecz tylko tyle, ile kanałów będziemy wykorzystywać w programie, na przykład:

```
PROC Close1()
  BYTE iccmd=$352
  iccmd=$C
  CIO(0,$10)
  RETURN
```

```
PROC Close2()
  BYTE iccmd=$362
  iccmd=$C
  CIO(0,$20)
  RETURN
```

Podobnie niezbędne będzie rozdzielenie na poszczególne kanały procedury **Open()**. Dodatkową trudność sprawia tu konieczność przekazania nazwy urządzenia. Najwygodniejsze jest więc przypisanie poszczególnych kanałów różnym urządzeniom, na przykład:

kanal	urządzenie
0	edytor
1	odczyt
2	zapis
4	drukarka
5	klawiatura
6	ekran

Teraz możemy definiować procedury w miarę potrzeby. Dla odczytu z klawiatury **Open()** może wyglądać następująco:

```
PROC OpenK()
  BYTE iccmd=$392,icax1=$39A
  CARD icba=$394,icbl=$398
  BYTE ARRAY name=['K':$9B]
  iccmd=$3
  icba=name
  icbl=3
  icax1=4
  CIO(0,$50)
  RETURN
```

a dla zapisu na drukarkę:

```
PROC OpenP()
  BYTE iccmd=$382, icax1=$38A
  CARD icba=$384, icbl=$388
  BYTE ARRAY name=['P ': $9B]
  iccmd=$3
  icba=name
  icbl=3
  icax1=8
  CIO(0, $40)
  RETURN
```

W wypadku ekranu trzeba dodatkowo przekazać procedurze wartość trybu graficznego, w którym ma być utworzony obraz. Definicja procedury będzie więc nieco zmodyfikowana:

```
PROC OpenS(BYTE mode)
  BYTE iccmd=$3A2, icax1=$3AA, icax2=$3AB
  CARD icba=$3A4, icbl=$3A8
  BYTE ARRAY name=['S ': $9B]
  iccmd=$3
  icba=name
  icbl=3
  icax1=(mode&$F0)!$C
  icax2=mode
  CIO(0, $60)
  RETURN
```

W zamieszczonej wyżej tabeli znajdują się pozycje "odczyt" i "zapis". Nie są to oczywiście urządzenia, lecz operacje wykonywane na pamięci zewnętrznej. Na przykład otwarcie magnetofonu do odczytu zrealizujemy następująco:

```
PROC OpenCR(BYTE gap)
  BYTE iccmd=$352, icax1=$35A, icax2=$35B
  CARD icba=$354, icbl=$358
  BYTE ARRAY name=['C ': $9B]
  iccmd=$3
  icba=name
  icbl=3
  icax1=4
  icax2=gap
  CIO(0, $10)
  RETURN
```

Parametr "gap" oznacza w tej procedurze długość przerw między odczytywanymi rekordami. Wartość 0 oznacza długie przerwy, a 128 — krótkie. Analogicznie wykonamy odczyt do zapisu:

```
PROC OpenCW(BYTE gap)
  BYTE iccmd=$362, icax1=$36A, icax2=$36B
  CARD icba=$364, icbl=$368
  BYTE ARRAY name=['C ': $9B]
  iccmd=$3
  icba=name
  icbl=3
  icax1=8
  icax2=gap
  CIO(0, $20)
  RETURN
```

Tą samą metodę można zastosować do plików dyskowych. Ponieważ nazwa pliku jest stosunkowo długa, to trzeba zastosować tu nieco inny sposób jej przekazania. Otwarcie do odczytu będzie więc następujące:

```
PROC OpenDR()
  BYTE iccmd=$352, icax1=$35A, icax2=$35B
  CARD icba=$354, icbl=$358
  BYTE ARRAY name
  name="D1:NAZWA.EXT"
  iccmd=$3
  icba=name(1)
  icbl=name(0)
  icax1=4
  CIO(0, $10)
  RETURN
```

do zapisu zaś:

```
PROC OpenDW()
  BYTE iccmd=$362, icax1=$36A, icax2=$36B
  CARD icba=$364, icbl=$368
  BYTE ARRAY name
  name="D1:NAZWA.EXT"
  iccmd=$3
  icba=name(1)
  icbl=name(0)
  icax1=8
  CIO(0, $20)
  RETURN
```

Dwie ostatnie z prezentowanych procedur mają zasadniczą wadę: dotyczą tylko jednego konkretnego pliku. Rozwiązanie takie jest do przyjęcia tylko w wypadku pliku o ustalonej nazwie, na przykład z danymi dla programu. Zwykle jednak występuje konieczność otworzenia pliku, którego nazwa jest nieznana w czasie pisania programu. Pożądane byłoby więc przekazywanie nazwy pliku do procedury, na przykład poprzez podanie adresu tablicy zawierającej tę nazwę. Dla odczytu może to wyglądać tak:

```
PROC OpenDR(CARD addr)
  BYTE iccmd=$352, icax1=$35A, icax2=$35B
  CARD icba=$354, icbl=$358
  iccmd=$3
  icba=addr+1
  icbl=addr
  icax1=4
  CIO(0, $10)
  RETURN
```

a dla zapisu tak:

```
PROC OpenDW(CARD addr)
  BYTE iccmd=$362, icax1=$36A, icax2=$36B
  CARD icba=$364, icbl=$368
  iccmd=$3
  icba=addr+1
  icbl=addr
  icax1=8
  CIO(0, $20)
  RETURN
```

Zastosowanie tego sposobu wymaga zadeklarowania na początku programu globalnej tablicy znakowej, w której będzie przechowywana nazwa pliku (oczywiście tablic takich może być kilka). Na przykład:

```
BYTE ARRAY name
```

Przed otwarciem kanału odpowiednią nazwę pliku należy umieścić w tablicy, a następnie wywołać właściwą procedurę:

```
name="D1:NAZWA1.EXT"
OpenDR(name)
name="D1:NAZWA2.EXT"
OpenDW(name)
```

Procedury zapisu

Korzystając z opisanego wyżej schematu, można tworzyć kolejne procedury. Jako pierwszą zrealizujemy procedurę **Put()**, która zapisuje jeden bajt do edytora (kanał 0), w odróżnieniu od normalnej wersji zapisującej do urządzenia określonego przez zmienną "device".

```
PROC Put(BYTE chr)
  BYTE iccmd=$342
  CARD icbl=$348
  iccmd=$B
  icbl=0
  CIO(chr, $00)
  RETURN
```

Możemy teraz zastosować ją w następnej procedurze, która umieszcza w edytorze znak końca wiersza:

```
PROC PutE()
  Put($9B)
  RETURN
```

UWAGA: Kanał 0 jest automatycznie otwierany dla edytora przez system operacyjny i nie powinien być zamykany. Procedury **Open()** i **Close()** dla tego kanału są więc zbędne.

W identyczny sposób definiujemy procedury **Put()** dla pozostałych urządzeń wykorzystywanych przez program. Na przykład dla drukarki (kanał 4) będą one następujące:

```
PROC PutP(BYTE chr)
  BYTE iccmd=$382
  CARD icbl=$388
  iccmd=$B
  icbl=0
  CIO(chr, $40)
  RETURN
```

```
PROC PutPE()
  PutP($9B)
  RETURN
```

W celu realizacji procedury **Print()** trzeba przekazać do bloku sterowania I/O adres napisu i jego długość. Jednakże prostsze (choć nieco dłuższe) będzie zastosowanie tu procedury **Put()** umieszczonej w pętli. Dla określenia wartości granicznej pętli użyjemy przy tym pierwszego elementu tablicy znakowej, który określa jej długość.

```
PROC Print(BYTE ARRAY txt)
  BYTE i
  FOR i=1 TO txt(0)
  DO
    Put(txt(i))
  OD
  RETURN
```

Przez proste dodanie na końcu wywołania **PutE()** otrzymamy procedurę **PrintE()**. Aby jednak nie pisać jej od nowa, skorzystamy z uprzednio zdefiniowanej procedury **Print()**:

```
PROC PrintE(BYTE ARRAY txt)
  Print(txt)
  PutE()
  RETURN
```

Powyższych procedur nie można jednak wykorzystać do zapisu wartości liczbowych. Podanie liczby typu BYTE spowoduje bowiem zapisanie znaku o odpowiadającym jej kodzie, a dla liczb CARD i INT będą to dwa znaki. Konieczne jest zatem uprzednie przekształcenie liczby na znaki kodu ASCII (cyfry), które będą umieszczone w tablicy tekstowej. Ponieważ tablica nie może być wartością zwracaną przez funkcję, to zamiana musi być doko-

nywana przez procedurę. Adres tablicy może być jednak parametrem procedury tylko w wypadku zamiany liczby typu BYTE. Równoczesne przekazanie dwubajtowej liczby CARD lub INT i wskaźnika tablicy (również dwa bajty) jest niemożliwe, gdyż przekracza dopuszczalną liczbę parametrów. Jedynym wyjściem jest więc zadeklarowanie globalnej tablicy, w której będą umieszczane przekształcone liczby:

```
BYTE ARRAY numstr
```

Aby utrzymać jednolitą strukturę procedur i uniknąć pomyłek, zastosujemy identyczny schemat dla wszystkich typów zmiennych, pomimo iż dla typu BYTE nie jest to konieczne. Odpowiednia procedura przekształcania liczby BYTE będzie więc miała następującą postać:

```
PROC StrB(BYTE val)
  BYTE tmp
  IF val=0 THEN
    numstr="0"
    RETURN
  FI
  tmp=val
  numstr="000"
  WHILE tmp>99
  DO
    numstr(1)==+1
    tmp==--100
  OD
  WHILE tmp>9
  DO
    numstr(2)==+1
    tmp==--10
  OD
  WHILE tmp>0
  DO
    numstr(3)==+1
    tmp==--1
  OD
  WHILE numstr(1)='J
  DO
    numstr(1)=numstr(2)
    numstr(2)=numstr(3)
    numstr(3)=numstr(0)
  OD
  RETURN
```

Mając już liczbę zapisaną w postaci ciągu znaków, możemy ją wyświetlić za pomocą procedur:

```
PROC PrintB(BYTE num)
  StrB(num)
  Print(numstr)
  RETURN
```

```
PROC PrintBE(BYTE num)
  PrintB(num)
  PutE()
  RETURN
```

Przekształcenie liczby typu CARD jest realizowane w ten sam sposób jak BYTE, lecz — ze względu na większą długość liczby — w większej liczbie kroków. Wynik przekształcenia jest również umieszczany w tablicy "numstr":

```
PROC StrC(CARD val)
  CARD tmp
  IF val=0 THEN
    numstr="0"
    RETURN
  FI
  tmp=val
  numstr="00000"
  WHILE tmp>9999
  DO
    numstr(1)==+1
    tmp==--10000
  OD
  WHILE tmp>999
  DO
    numstr(2)==+1
    tmp==--1000
  OD
  WHILE tmp>99
  DO
    numstr(3)==+1
    tmp==--100
  OD
  WHILE tmp>9
  DO
    numstr(4)==+1
    tmp==--10
  OD
  WHILE tmp>0
  DO
    numstr(5)==+1
    tmp==--1
  OD
  WHILE numstr(1)='J
  DO
    numstr(1)=numstr(2)
    numstr(2)=numstr(3)
    numstr(3)=numstr(4)
    numstr(4)=numstr(5)
    numstr(5)=numstr(0)
  OD
  RETURN
```

Uzyskaną w rezultacie przekształcenia liczbę możemy już wyświetlić za pomocą procedur:

```
PROC PrintC(CARD num)
  StrC(num)
  Print(numstr)
RETURN

PROC PrintCE(CARD num)
  PrintC(num)
  PutE()
RETURN
```

Liczba typu INT różni się od CARD tylko sposobem interpretacji zapisanej wartości. Przy przekształcaniu jej na ciąg znaków można więc wykorzystać zdefiniowaną już procedurę **StrC()**. Dla dodatnich wartości liczby INT przekształcenie będzie identyczne jak CARD, dla wartości ujemnych zaś przed przekształceniem należy zmienić znak liczby, a potem dopisać do uzyskanego ciągu znak "-".

```
PROC StrI(INT val)
  CARD tmp
  BYTE i
  IF val=0 THEN
    numstr="0"
    RETURN
  FI
  IF val>0 THEN
    tmp=val
    StrC(tmp)
    RETURN
  FI
  tmp=-val
  StrC(tmp)
  numstr(0)=="-1"
  FOR i=numstr(0) TO 2
    DO
      numstr(i)=numstr(i-1)
    OD
  numstr(1)="-"
  RETURN
```

Procedury **PrintI()** i **PrintIE()** będą oczywiście analogiczne do pokazanych wyżej:

```
PROC PrintI(INT num)
  StrC(num)
  Print(numstr)
RETURN

PROC PrintIE(INT num)
  PrintI(num)
  PutE()
RETURN
```

Wszystkie pokazane wyżej procedury powodują zapisanie wskazanych wartości w edytorze, a więc na ekranie, dla innych urządzeń (drukarka, magnetofon i stacja dysków) należy zdefiniować podobne procedury. Oczywiście w pisany program wystarczy umieścić tylko te, które są niezbędne. Na przykład, nie trzeba pisać procedur dla drukarki, jeśli nie przewiduje się w programie wykonywania wydruków.

Procedury odczytu

Procedury służące do odczytu danych skonstruujemy w taki sam sposób jak procedury zapisu. Także i w tym wypadku zostaną zaprezentowane tylko procedury odczytu z jednego urządzenia, zdefiniowanie zaś pozostałych pozostanie dla czytelników. Tym wybranym urządzeniem jest klawiatura, która w odróżnieniu od edytora musi być uprzednio otwarta w programie (procedurą **OpenK()**).

Podstawową procedurą odczytu jest **Get()**. W jej definicji konieczne jest użycie krótkiego bloku kodu maszynowego. Ponieważ procedura **CIO** nie zwraca żadnej wartości, a po wykonaniu przez nią operacji GET odczytany bajt znajduje się w akumulatorze, to zapisujemy bezpośrednio po **CIO** rozkaz STA i adres zmiennej "val" (typu BYTE). Cała funkcja **Get()** wygląda następująco:

```
CHAR FUNC Get()
  BYTE val,icmd=$392
  CARD icbl=$39B
  icmd=$7
  icbl=0
  CIO(0,$50)
  [$8D val]
  RETURN(val)
```

Przez wywoływanie funkcji **Get()** w pętli, aż do odczytania znaku końca wiersza (\$9B), utworzymy procedurę **InputS()** odczytującą ciąg znaków:

```
PROC InputS(BYTE ARRAY txt)
  BYTE i=101,k
  k=Get()
  WHILE k#$9B
    DO
      i==+1
      txt(i)=k
      k=Get()
    OD
  txt(i)=k
  txt(0)=1
  RETURN
```

Przy odczycie liczb stajemy przed problemem odwrotnym niż przy zapisie. Ponieważ jednak liczba może być zwracana przez funkcję, to w takiej właśnie formie zdefiniujemy przekształcenia ciągów na liczby. Pierwszą taką funkcją będzie oczywiście **ValB()**:

```
BYTE FUNC ValB(BYTE ARRAY str)
  BYTE i,j,num=101
  FOR i=1 TO str(0)
    DO
      j=num
      j==LSH 2 + num
      num=j LSH 1
      num==+str(i)-$30
    OD
  RETURN(num)
```

Korzystając z tego przekształcenia oraz z procedury **InputS()** możemy teraz zbudować funkcję odczytującą wartości typu BYTE:

```
BYTE FUNC InputB()
  BYTE val
  BYTE ARRAY txt
  InputS(txt)
  val=ValB(txt)
  RETURN(val)
```

Niemal identyczna z **ValB()** jest funkcja **ValC()**. Jednakże musi ona być zdefiniowana oddzielnie, ze względu na inny typ występującej w niej wartości.

```
CARD FUNC ValC(BYTE ARRAY str)
  BYTE i
  CARD j,num=101
  FOR i=1 TO str(0)
    DO
      j=num
      j==LSH 2 + num
      num=j LSH 1
      num==+str(i)-$30
    OD
  RETURN(num)
```

W ten sam sposób **InputC()** różni się od **InputB()**:

```
CARD FUNC InputC()
  CARD val
  BYTE ARRAY txt
  InputS(txt)
  val=ValC(txt)
  RETURN(val)
```

Przy realizacji funkcji **ValI()** wykorzystamy (podobnie jak przy **StrI()**) funkcję **ValC()**. Po rozpoznaniu znaku minus na początku ciągu zostaje on usunięty i liczba jest przekształcana jak liczba CARD, a w wyniku zmieniany jest znak. Jeśli liczba jest dodatnia, to jest przekształcana jak CARD.

```
INT FUNC ValI(BYTE ARRAY str)
  BYTE i
  INT num
  IF str(1)="-" THEN
    FOR i=2 TO str(0)
      DO
        str(i-1)=str(i)
      OD
    str(0)=-1
    num=-ValC(str)
  ELSE
    num=ValC(str)
  FI
  RETURN(num)
```

Tak zdefiniowaną funkcję **ValI()** użyjemy do określenia funkcji **InputI()**:

```
INT FUNC InputI()
  INT val
  BYTE ARRAY txt
  InputS(txt)
  val=ValI(txt)
  RETURN(val)
```

UWAGA: W odróżnieniu od normalnych procedur **Input()** w **Action!** procedury zdefiniowane powyżej korzystają z kanału 5. Musi on więc zostać otwarty przed użyciem którejkolwiek z nich.

Procedury graficzne

W zasadzie procedury graficzne są także procedurami wejścia/wyjścia. Zostały one jednak wyodrębnione, ponieważ korzystają zawsze z kanału 6. Ponadto znakiem zapisywanym przez te procedury jest kod koloru określony przez zadeklarowaną na początku programu zmienną:

```
BYTE color=$2FD
```

W **Action!** zmienna ta ma zawsze początkową wartość równą 0. Nie można jednak w deklaracji zmiennej określić równocześnie jej adresu i wartości, więc należy to wykonać w początkowej części programu lub bezpośrednio przed użyciem jednej z procedur rysujących na ekranie.

Zanim zostanie wykonana jakakolwiek procedura rysująca na ekranie lub odczytująca z niego dane, musi być otwarty przeznaczony do tego kanał. Realizuje to procedura **Graphics()**, której parametrem jest tryb graficzny, w jakim będzie otwarty ekran. Procedura ta jest właściwie sekwencją procedur **Close()** i **Open()**. Ponieważ jednak nie stosuje się oddzielnie procedury **Close()** dla kanału 6, to została ona tu włączona. W rezultacie procedura **Graphics()** wygląda następująco:

```
PROC Graphics(BYTE mode)
  BYTE icmd=$3A2
  icmd=$C
  CIO(0,$60)
  OpenS(mode)
  RETURN
```

Jeżeli nie przewidujemy w programie samodzielnego stosowania procedury **OpenS()** (czyli niemal zawsze), to można jej definicję zapisać bezpośrednio w definicji **Graphics()**.

Narysowanie punktu jest równoważne zapisaniu pojedynczego bajtu w określonym miejscu ekranu. Można więc zrealizować to przez połączenie procedur **Position()** i **Put()**. Wymagałoby to jednak zdefiniowania procedury **Put()** dla kanału 6. Zamiast tego odpowiednie instrukcje zapiszemy od razu w procedurze **Plot()** (właśnie tak można wykonać również **Graphics()**):

```
PROC Plot(CARD col BYTE row)
  BYTE icmd=$3A2
  CARD icbl=$3A8
  Position(col,row)
  icmd=$B
  icbl=0
  CIO(color,$60)
  RETURN
```

Rysowanie linii jest natomiast realizowane przez specjalną procedurę systemu operacyjnego. Jest ona wywoływana przez procedurę **CIO**, jeśli kod rozkazu wynosi \$11. Przedtem trzeba jeszcze umieścić kursor na końcu linii i zapisać kolor w rejestrze ATACHR. W efekcie otrzymujemy następującą procedurę:

```
PROC DrawTo(CARD col BYTE row)
  BYTE atachr=$2FB,
  icmd=$3A2,icax1=$3AA
  Position(col,row)
  atachr=color
  icmd=$11
  icax1=$C
  CIO(0,$60)
  RETURN
```

Ta sama procedura systemu realizuje wypełnianie obszaru, jeśli kod rozkazu będzie wynosił \$12 (dziwne, że nie wykorzystali tego autorzy Atari Basic). Po zmianie tej wartości uzyskamy procedurę **Fill()**:

```
PROC Fill(CARD col BYTE row)
  BYTE atachr=$2FB,
  icmd=$3A2,icax1=$3AA
  Position(col,row)
  atachr=color
  icmd=$12
  icax1=$C
  CIO(0,$60)
  RETURN
```

Funkcja **Locate()** służy do odczytu informacji z ekranu. Różni się ona od **Get()** dokładnie tak samo, jak **Plot()** różni się od **Put()**. Bez dalszych komentarzy można więc przedstawić jej definicję:

```
BYTE FUNC Locate(CARD col BYTE row)
  BYTE val,icmd=$3A2
  CARD icbl=$3A8
  Position(col,row)
  icmd=$7
  icbl=0
  CIO(0,$60)
  [$8D val]
  RETURN(val)
```

Inne procedury

Pozostały jeszcze nie zdefiniowane procedury operujące na blokach pamięci (**MoveBlock()**, **SetBlock()** i **Zero()**) oraz tablicach znakowych (**SAssign()**, **SCopy()**, **SCopyS()** i **SCompare()**), także procedury **Poke()** i **Peek()**. Ostatnie z wymienionych są praktycznie zbędne i dlatego zostały pominięte.

Inaczej przedstawia się sprawa procedur operujących na blokach i tablicach. Wymagają one tak wielu parametrów, że trzeba je definiować indywidualnie do konkretnych zastosowań albo przekazywać niezbędne parametry przy użyciu zmiennych globalnych. Ponieważ poza tym są one bardzo proste, to uznałem, że każdy programista napisze je w takiej wersji, jaka będzie najbardziej odpowiednia dla tworzonego przezeń programu.

Wojciech ZIENTARA



Dwa miesiące temu rozpoczęliśmy kodowanie naszej gry według opracowanego algorytmu. Teraz w kilku następnych odcinkach będziemy rozwijać powstający program w taki sam sposób, w jaki uprzednio rozwijaliśmy algorytm.

Cofnijmy się zatem do „Mojego Atari” 2. Zrealizowaliśmy już zamieszczone tam fragmenty algorytmu. Kolejnym etapem będzie więc rozwinięcie właściwej gry (według schematu z „Mojego Atari” 3). Zapisanie tej części algorytmu jest zadaniem dziecinnie prostym, gdyż składa się on wyłącznie z wywołań innych procedur. Trzeba jednak pamiętać, że pominęliśmy w nim wyjścia z pętli. Zrealizujemy to przez sprawdzenie warunku zakończenia gry po każdej z procedur. W efekcie otrzymamy w **Action!** następującą procedurę:

```
PROC Gra()  
  Plansza()  
  DO  
    RuchStatku()  
    IF q#0 THEN  
      RETURN  
    FI  
    RuchNiszcz()  
    IF q#0 THEN  
      RETURN  
    FI  
    RuchOkretu()  
    IF q#0 THEN  
      RETURN  
    FI  
    Odpalenie()  
    IF q#0 THEN  
      RETURN  
    FI  
    Peryskop()  
    IF q#0 THEN  
      RETURN  
    FI  
    RuchOkretu()  
    IF q#0 THEN  
      RETURN  
    FI  
    Peryskop()  
    IF q#0 THEN  
      RETURN  
    FI  
  OD  
RETURN
```

Pamiętaj, że procedura ta musi znaleźć się w programie przed miejscem jej wywołania, czyli przed procedurą Konwoj (). Niemal identyczna jest realizacja tego fragmentu w **Basicu**. Ponieważ jednak **Atari Basic** nie ma instrukcji pętli bez końca (jak DO/OD w **Action!**), to trzeba ją zastąpić instrukcją skoku (wiersz 1580). Jest to jeden z wypadków, gdy instrukcja GOTO jest trudna do wyeliminowania. Gra właściwa przybierze więc postać:

```
KH 1497 REM
DN 1498 REM *** Gra ***
KP 1499 REM
WT 1500 GOSUB 1000
GV 1510 GOSUB 500:IF Q<>0 THEN RETURN
GM 1520 GOSUB 400:IF Q<>0 THEN RETURN
GD 1530 GOSUB 300:IF Q<>0 THEN RETURN
IC 1540 GOSUB 200:IF Q<>0 THEN RETURN
JR 1550 GOSUB 820:IF Q<>0 THEN RETURN
GM 1560 GOSUB 300:IF Q<>0 THEN RETURN
JX 1570 GOSUB 820:IF Q<>0 THEN RETURN
QR 1580 GOTO 1510
```

wyświetlenia tryb 2, który ma duże kolorowe litery. Na umieszczenie informacji o autorze i zastrzeżenia praw autorskich przeznaczymy natomiast okno tekstowe. Aby kolory tła na ekranie głównym i w oknie tekstowym były jednakowe, konieczne jest odpowiednie ustawienie koloru 0 (tło ekranu) i/lub 2 (tło okna). Ponieważ ekran jest czarny, to zmienimy tylko kolor 2. Ponadto poprzez zmianę wartości w programie ANTIC-a spowodujemy wyświetlanie w jednym wierszu zmniejszonych liter. Niezbędne w tym celu instrukcje można przeanalizować na wydruku, który wygląda następująco w **Action!**:

```
PROC Tytul()
CARD dlptr=560
Graphics(2)
SetColor(2,0,0)
Poke(dlptr+9,6)
Close(3)
Open(3,"K:",4,0)
Position(7,2)
PrintDE(6,"BAJTEK")
Position(5,4)
PrintDE(6,"Prezentacja")
Position(7,7)
PrintDE(6,"konwoj")
Position(11,6)
PutD(6,12) PutDE(6)
crsinh=1 PutE()
PrintE("      Copyright (c) 1990 by Bajtek")
PrintE("      Wojciech Zientara, Warszawa")
RETURN
```

w Basicu zaś:

```

K1 2497 REM
HS 2498 REM *** Tytuł ***
KQ 2499 REM
VV 2500 GRAPHICS 2:SETCOLOR 2,0,0:DL=PEEK
    (560)+256*PEEK(561):POKE DL+9,6:CLOSE
    #3:OPEN #3,4,0,"K1":
NH 2510 POSITION 7,2:? #6;"BAJTEK"
WH 2520 POSITION 5,4:? #6;"Przewidyw"
ZX 2530 POSITION 7,7:? #6;"konwoj"
YR 2540 POSITION 11,6:? #6;CHR$(12)
FG 2550 POKE 752,1:? :? " Copyright (c
    ) 1990 by Bajtek":? " Wojciech Zielen
    tara, Warszawa":RETURN

```

UWAGA: w obu procedurach litera "p" w napisie "prezentuje" jest mała, a wygląda na dużą, gdyż drukarka tak ją drukuje w negatywie.

Do wyjaśnienia pozostały w tych procedurach jeszcze trzy rzeczy. Instrukcja `Open` otwierająca kanał dla odczytu z klawiatury może być umieszczona w dowolnym miejscu przed główną pętlą programu. Może to być więc zarówno przed tytułem, jak i po nim, a ja wpisałem ją do procedury `Tytuł()`. Dwie pozostałe instrukcje służą do poprawienia wyglądu strony. Znak o kodzie 12 wyświetla w trybie 2 przecinek nad literą "o", co daje efekt "ô". Umieszczenie wartości 1 w komórce 752 czyni natomiast kursor niewidocznym. Oczywiście w **Action!** trzeba uprzednio odpowiednio zdefiniować zmienną "crsinh". Jest to zmienna globalna, ponieważ występuje w kilku procedurach.

Drugą prostą procedurą jest Instrukcja () — nie wymaga ona nawet objaśnień poza jednym szczegółem. Program musi czekać, aż użytkownik tą instrukcję przeczyta. Dodatkowe przerwy trzeba również wstawić pomiędzy "strony" instrukcji, jeśli cała nie mieści się jednorazowo na ekranie. Przerwę taką realizuje pomocnicza procedura Oczekiwanie(). Instrukcja w **Action!** będzie więc miała postać:

```

PROC InstrukcjaJ,
Graphics(0)
SetColor(2,0,0)
crsinh=1
Put(125)
PrintE("                "),
PrintE("                KONWOJ")
PutE()
PrintE(" Gra toczy sie na planszy 10 na 10.")
PrintE("Komputer dowodzi statkiem handlowym")
PrintE("i dwoma niszczycielami, a Ty okretem")
PrintE("podwodnym. Statek startuje z punktu")
PrintE("9,9 i posuwa sie w strone punktu 0,0")
PrintE("poruszajac sie losowo o 0-3 p. k.")
PrintE("Jeden z niszczycieli szuka Twojego")
PrintE("okrętu, a drugi eskortuje statek nie")
PrintE("oddalajac sie od niego. Twój okręt")
PrintE("startuje z punktu 0,1 i przesuwa sie")
PrintE("po jednym polu, dwukrotnie w kazdej")
PrintE("rundzie. Po wykonaniu ruchu prowadzi")
PrintE("sie przez peryskop obserwacje sasla")
PrintE("dujacych pol. Czasem dostajesz takze")
PrintE("raporty wywiadu. Jeden raz w kazdej")
PrintE("rundzie mozesz strzelac torpedy.")
Oczekiwanie()
Put(125) PutE()
PrintE("Gra toczy sie w nastepujacych fazach:")
PutE()
PrintE("      1. ruch konwoju")
PrintE("      2. Twoj ruch")
PrintE("      3. odpalenie torped")
PrintE("      4. obserwacja")
PrintE("      5. Twój ruch")
PrintE("      6. obserwacja")
PrintE("      itd.")
PutE()
PrintE(" Twoim zadaniem jest zatopienie przy")
PrintE("pomocy czterech posiadanych torped")
PrintE("statku, zanim dotrze on do portu.")
PrintE("Oczywiscie, wejscie na pole zajete")
PrintE("przez niszczyciel spowoduje utrate")
PrintE("okrętu podwodnego i przegranie gry.")
PutE() PutE()

```

```
PrintE("          ** POWODZENIA **")
PutE() PutE()
Oczekiwanie()
RETURN
```

a w Basicu:

```

KT 3997 REM
KK 3998 REM *** Instrukcja ***
LB 3999 REM
MO 4000 GRAPHICS 0:SETCOLOR 2,0,0:POKE 75
2,1: CHR$(125);
ZC 4010 ? " " ,":? "
KONWOJ":?
IH 4020 ? " Gra toczy sie na planszy 10
na 10.":? "Komputer dowodzi statkiem h
andlowym"
SC 4030 ? "i dwoma niszczycielami, a Ty o
kretem":? "podwodnym. Statek startuje
z punktu"
HP 4040 ? "9,9 i posuwa sie w strone punk
tu 0,0":? "poruszajac sie losowo o 0-3
pola."
UL 4050 ? "Jeden z niszczycieli szuka Two
jego":? "okretu, a drugi eskortuje sta
tek nie"
WI 4060 ? "oddalajac sie od niego. Twój o
kret":? "startuje z punktu 0,1 i przes
uwa sie"
SN 4070 ? "po jednym polu, dwukrotnie w k
azdej":? "rundzie. Po wykonaniu ruchu
prowadzi"
QQ 4080 ? "sie przez peryskop obserwacje
sasla-":? "dujacych pol. Czasem dostaj
esz takze"
ZZ 4090 ? "raporty wywiadu. Jeden raz w k
azdej":? "rundzie mozesz strzelac torp
edy." :GOSUB 120
WX 4100 ? CHR$(125):? "Gra toczy sie w na
stepujacych fazach":? :? " 1
. ruch konwoju"
RN 4110 ? " 2. Twój ruch":? "
3. odpalenie torped":? "
4. obserwacja"
OS 4120 ? " 5. Twój ruch":? "
6. obserwacja":? " itd
.:?
CC 4130 ? " Twoim zadaniem jest zatopien
ie przy"
EC 4140 ? "pomocy czterech posiadanych to
rpied":? "statku, zanim dotrze on do po
rtu."
BY 4150 ? " Oczywiście, wejscie na pole
zajete":? "przez niszczyciel spowoduje
utrata"
UG 4160 ? "okretu podwodnego i przegranie
gry.":? :? :? " ** POWODZENI
A ***:? :?
4170 GOSUB 120:RETURN
IB

```

Również i tu dla poprawienia wyglądu należy schować kursor (jak w procedurze wyświetlającej tytuł).

Zastanówmy się jeszcze nad sposobem zrealizowania przerwy między poszczególnymi częściami instrukcji i na jej końcu. Użycie w tym celu pętli opóźniającej wstrzymującej wykonywanie programu na określony czas nie jest dobrym rozwiązaniem. Różni użytkownicy będą bowiem czytać z różną szybkością, a ponadto trzeba umożliwić zastanowienie się nad przeczytaną treścią. Najlepszym wyjściem jest oczekiwanie na sygnał od użytkownika, co w **Action!** zrealizuje następująca procedura:

```
PROC Oczekiwanie()
  BYTE k
  PutE()
  Print("          NACISNIJ DOWOLNY KLAWISZ")
  k=GetD(3)
RETURN
```

Wykorzystywany przez tą procedurę kanał 3 został uprzednio otwarty podczas wyświetlania tytułu. Procedura ta nie zwraca żadnej wartości, gdyż jest zbędne. Natomiast w **Basicu** analogiczna procedura:

```
RM 118 REM
CV 119 REM *** Oczekiwanie ***
LY 120 ? : ? " NACISNIJ DOWOLNY KLAWI
SZ":GET #3,KY:RETURN
```

przypisuje odczytaną wartość zmiennej KY, która jednak nie znajduje później nigdzie zastosowania

Przed przystąpieniem do dalszej części kodowania programu należy ustalić liczbę i rodzaj zmian, które muszą być użyte w programie. Jedną z nich będzie "q" określająca wynik. Warto dodatkowo określić zmienne "a1" i "a2", które przechowywać będą zwycięstw gracza i komputera. Umożliwi to określenie końcowego wyniku serii kolejnych rozgrywek. Zmienne te muszą mieć początkową wartość równą zero. Następną zmienną ("l") będzie zawierała liczbę torped.

Podstawowymi informacjami, które muszą być dostępne w wielu miejscach programu, są wsob-

SZKOŁKA

JAK K O D O W A Ć ? (2)

rzędne poszczególnych obiektów: okrętu podwodnego (*marine* — "mx" i "my"), statku (*ship* — "sx" i "sy"), niszczycieli (*destroyers* — "dx" i "dy") oraz dodatkowo poprzednie współrzędne obiektu niezbędne do skasowania symbolu na ekranie (*old* — "ox" i "oy"). Przy ruchu obiektu trzeba także zapamiętywać kierunek zmiany współrzędnych (*change* — "cx" i "cy"). Ponieważ niszczyciele są dwa, to zmienne "dx" i "dy" zrealizujemy jako tablice, aby uprościć procedury. Ponadto wprowadzimy tablicę "ns", która będzie określała istnienie niszczyciela — jest to konieczne, gdyż zatopienie niszczyciela (a nawet obu) nie kończy jeszcze gry.

Potrzeba zadeklarowania jeszcze jednej pary zmiennych (*text* — "tx" i "ty") nie jest oczywista, lecz staje się natychmiast widoczna po uruchomieniu programu. Ponieważ napisy informacyjne są wyświetlane na przemian z symbolami obiektów, to konieczne jest zapamiętywanie końca ostatniego napisu, aby kolejny znalazł się we właściwym miejscu. Ostatnia zmienna jest za to oczywista — tablica tekstowa "p" do przechowywania odpowiedzi użytkownika na pytania zadawane przez komputer.

Deklaracja wszystkich tych zmiennych powinna w **Action!** wyglądać następująco:

```
BYTE cx,cy,ox,oy,tx,ty,mx,my,sx,sy,t,
      q,q1=101,q2=101,
      crsrow=64,crscol=65,crsinh=752
BYTE ARRAY p(3),dx(2),dy(2),ns(2)
```

Użyte w niej nazwy "crscol", "crsrow" i "crsinh" określają nazwy komórek pamięci, co pozwoli później na zastosowanie instrukcji przypisania zamiast procedury *Poke()*. W **Basicu** wystarczy zadeklarowanie tablic, gdyż pozostałe zmienne są automatycznie deklarowane przez interpreter. Znalazły się tu tylko zmienne "Q1" i "Q2", gdyż muszą mieć one określoną wartość początkową. Deklaracja będzie więc znacznie krótsza:

```
LK 10 DIM P$(3),DX(1),DY(1),NS(1):Q1=0:Q2=0:GOTO 5000
```

Wątpliwość mogą budzić różne wymiary tablic w **Action!** i **Basicu**. Deklaracja w **Action!** określa bowiem zawsze liczbę elementów. Natomiast w **Basicu** dla tablic tekstowych ("P\$") podaje się liczbę elementów, dla liczbowych zaś — liczbę elementów zmniejszoną o jeden, ponieważ pierwszy element ma numer zero. Informację tę trzeba zapamiętać — ta różnica między obydwoma językami jest przyczyną różnego indeksowania tablic w dalszej części programu. W **Action!** elementy tablic mają indeksy 1 i 2, a w **Basicu** — 0 i 1.

Mając określone zmienne możemy ułożyć procedurę inicjowania, która ustali ich początkowe parametry: położenie okrętu podwodnego, statku i niszczyciela eskortującego oraz liczbę torped. Znacznik końca gry ("q") otrzymuje wartość 0, znaczniki istnienia poszczególnych niszczycieli wartości 1. Niszczyciel poszukujący okrętu podwodnego powinien znajdować się w przypadkowym miejscu, lecz nie dokładnie między okrętem a statkiem. Dlatego też trzeba wprowadzić warunek sprawdzający równość współrzędnych. W **Action!** będzie to wyglądało tak:

```
PROC Inicjowanie()
mx=0 my=1 sx=8 sy=8 t=4 q=0
ns(1)=1 ns(2)=1
dx(1)=7 dy(1)=6
DO
  dx(2)=Rand(2)+8
  dy(2)=Rand(2)+8
UNTIL dx(2)<>dy(2) OD
RETURN
```

W **Basicu** zamiast pętli DO/OD zastosujemy instrukcję GOTO (jak w procedurze Gra). Ponadto musimy użyć funkcji INT, gdyż RND daje wynik niecałkowity. Efektem będzie procedura:

```
KS 2997 REM
WO 2998 REM *** Inicjowanie ***
LA 2999 REM
VD 3000 MX=0:MY=1: SX=8:SY=8:T=4:Q=0:NS(0)=1:
NS(1)=1:DX(0)=7:DY(0)=6
YG 3010 DX(1)=INT(RND(0)*2)+8:DY(1)=INT(RND(0)*2)+8:IF DX(1)=DY(1) THEN 3010
AH 3020 RETURN
```

W ten sposób zakodowaliśmy już całą początkową część programu.

Wojciech ZIENTARA

TOP 12

Na całym świecie modne są różne listy przebojów: filmów, piosenek, programów itd. To, co chcę wam zaprezentować, nie jest próbą rozdziału miejsc dla poszczególnych programów, ale raczej wyborem (zapewne subiektywnym) najlepszej dwunastki, jednak bez wyróżniania pierwszego czy dwunastego.

Celem tej rubryki jest poprowadzenie was przez dżunglę oprogramowania, nie narzucanie, lecz zaproponowanie pewnych programów — moim zdaniem najlepszych, a także sprostowanie wielu nieścisłości dotyczących oprogramowania. Ponadto będę dbał, aby znalazły się tu także programy polskie (jeżeli na to zasługują), wydawane oficjalnie i sprzedawane w „przyswoitej” formie przez dystrybutorów. Tym samym chciałbym zachęcać do popierania oficjalnej dystrybucji programów — kupując legalną kopię (na razie mowa tylko o polskim oprogramowaniu) ma się pewność, że program będzie działał poprawnie. Forma jego „podania” (przyswoita kaseta, wkładka, instrukcja) jest o wiele przyjemniejsza w odbiorze. A co najważniejsze: kupując legalne kopie (a nie pokątne „przegrywki”) zachęca się programistów i dystrybutorów do powiększania rynku oprogramowania, co leży oczy-

wiście w naszym wspólnym interesie.

Wracając do naszej rubryki — proponowana dwunastka zostanie wszechstronnie zaprezentowana, przy każdym z programów znajdą się następujące informacje: tytuł, producent, rodzaj programu, rodzaj nośnika, na jakim jest dostępny (C — kasecie, D — dyskietka) oraz krótki opis programu. Poza tą dwunastką chciałbym napisać parę słów na temat wybranych trzech programów, których w chwili opracowywania listy nie ma na naszym rynku, natomiast właśnie pojawiły się w katalogach zachodnich firm. Czekaających na same nowości muszę trochę zasmucić: w każdej dwunastce znajdą się także programy starsze, jednak — moim zdaniem — w pełni zasługujące na zainteresowanie. Na końcu znajdziecie również parę słów o programach, których niestety nie ma na nasz ulubiony komputer, oraz kilka ciekawostek. Podobnie jak dla komputerów 8-bitowych chciałbym zaprezentować dwanaście gier dla 16-bitowych komputerów Atari ST. W odróżnieniu od sąsiedniej rubryki znajdą się tu przede wszystkim programy nowe, wybrane z wielkiego potoku napływającego do nas z Zachodu — o naszych rodzimych programach nie warto wspominać, po prostu kompletny brak, a raczej nic godnego uwagi. Na zakończenie trochę ciekawostek i oczywiście trzy z wielu programowych zapowiedzi. Mam nadzieję, że wybaczycie mi, iż ze względu na cykl wydawniczy nie wszystko, co super NEW, będzie aż tak nowe ale mimo to zapewne nadal ciekawe. Naturalnie nie będzie tu już informacji na temat nośnika (zawsze dyskietka), w zamian za to podam, czy wykorzystuje on możliwości STE (dźwięk stereo, itp.).

Oto наша dwunastka:

XE/XL		
DRACONUS Zeppelin Games przygodowa	C/D	Jedna z najlepszych gier labiryntowo- przygodowych, znakomita grafika, bar- dzo dobra muzyka, wiele komnat, duże postacie. Aby ukończyć grę, należy ze- brać odpowiednie przedmioty. Dostęp- na instrukcja z mapą.
ROAD RACE Silverbird auto-moto	C/D	Najlepszy program „samochodowy” na „małe” Atari. Wspaniały wyścig do- okoła Stanów Zjednoczonych, wybór trasy, różne warunki pogodowe, tanko- wanie paliwa, policja, pomiar czasu, re- jestracja wyników itp.
WINTER OLYMPIAD Tynesoft sportowa	C/D	Zimowa olimpiada, pięć konkurencji (zjazd, skoki, slalom, biathlon i bobsleje) w znakomitej grafice, świetna dla kilku osób, rywalizacja, punkty, medale. Naj- lepszy w swoim rodzaju wobec braku Winter Games na Atari XL/XE.
UNIVERSAL HERO Mastertronic przygodowa	C/D	Jedna z najlepszych gier labiryntowych, bardzo dobra grafika. Znalazłeś się na obcej planecie, musisz zebrać przed- mioty i uruchomić komputer — oryginal- na gra zawieszona w tym miejscu, w Polsce poprawiono program i działa pra- widłowo.
THE GOONIES Activision przygodowa	C/D	Znakomita zabawa dla jednego lub dwóch graczy. Opowieść o dwóch chłó- pakach szukających skarbu, na podsta- wie słynnego filmu S. Spielberga. Znak- omita grafika, wpadająca w ucho muzyka i możliwość współpracy dwóch graczy.
ACE OF ACES US Gold symulacyjna	C/D	Najlepszy beletrystyczny symulator lotu, znakomita grafika, wiele wariantów, kilka misji do wykonania, walki w powietrzu i z celami naziemnymi itp. Wersja kasetowa stworzona w Polsce, różne rodzaje — niektóre mogą działać nieprawidłowo.
INFILTRATOR US Gold przygodowa	D	Znakomita gra, która przywędrowała z USA. Łączy w sobie kilka rodzajów gier: symulację (lot śmigłowcem), zręcznoś- ciową (walki w bazie), labiryntową (bu- dynek dowództwa) itp. Na kasecie sprzedawane często tylko fragmenty.
ZYBEX Zeppelin Games zręcznościowa	C/D	To prawdziwy hit — znakomita grafika i muzyka. Musisz pokonać różne niebez- pieczeństwa na planecie och i uciec z sy- stemu. Wersja ZYBEX II jest bliska origi- nału, natomiast krążący po Polsce ZYBEX — to zubożona przez złodziei wersja gry.
BLINKY’S SCARY S. Zeppelin Games przygodowa	C/D	Największy hit ostatnich dni, fenomena- lna grafika i świetna zabawa z duszkiem wędrującym po zamku. Problem klasy- czny — trzeba coś zebrać, coś prze- nieść, ale wykonanie znakomite, całości dopełnia bardzo dobry dźwięk.
CULTIVATION Ke-Soft logiczna	D	Mało znana, stosunkowo nowa gra. Plansza o różnych kształtach składa się z kamieni nieruchomych oraz rucho- mych, które należy dobrać w pary. Mo- żna je przesuwac tylko w poziomie, czas wymusza szybkie myślenie.
WARSAW TETRIS Mirage zręcznościowa	C/D	Powstała w Polsce wersja słynnej gry na Atari XL/XE. Był to pierwszy program firmy Mirage, co prawda jeszcze nie podpisany i rozprowadzany w obiegu „towarzystkim”, jednak wartość i po- ziom gry bardzo wysokie.
MISJA Avalon przygodowa	C/D	Firma Avalon jako pierwsza zaistniała poważnie na naszym rynku. MISJA to przygoda komandos w bazie wojsko- wej, bardzo dobra grafika, świetna fabu- ła, gra musi się podobać każdemu. Po- lecam zakup oryginalnej wersji.

W pierwszym odcinku poprzestańmy tylko na liście hitów. O nowościach z zagranicy następnym razem. Natomiast, po raz któryś, muszę zmartwić część Atarowców: takich programów jak ROBOCOP, PLATOON, COMMANDO, OPERATION WOLF, LAST NINJA, TEST DRIVE i GAME OVER nie ma w żadnym katalogu na komputery Atari XL/XE, ale czas płynie i może się coś zmienić.

ST	
CADAVER Image Works przygodowa	Program roku 1990 — znakomita grafika trójwymiarowa, świet- nie animowana postać, przygody rycerza w lochach pełnych niebezpieczeństw, pułapek, stworów. Niezbędna może się okazać także umiejętność szybkiego kojarzenia.
LAST NINJA System 3 przygodowa	Gra, której raczej nie trzeba reklamować. W głównej roli wale- czny mistrz wschodniej sztuki walki wędrujący przez niebez- pieczne miasto, aby pokonać swojego odwiecznego wroga. Grafika bez zastrzeżeń, dużo dobrze animowanych walk.
MIDNIGHT RESISTANCE Ocean zręcznościowa	Rodzaj gry szczególnie lubianej przez najmłodszych, czyli ko- mandosi w akcji. Zasada dobrze znana z osławionej OPERA- TION WOLF, dużo strzelania, dużo akcji, raczej niewiele my- ślenia, ale ogólne wykonanie gry wzorowe.
SPACE ACE Readysoft przygodowa	Don Blouth stworzył nową generację gier, grafika i animacja jak z filmu rysunkowego, dużo okrzyków i efektów specjalnych. Nasze zadanie polega tylko na odpowiednim krótkim ruchu we właściwym momencie. Potrzebne mocne nerwy, ale efekt koń- cowy znakomity — fascynuje dźwięk i to, co dzieje się na ekr- anie, choć przy dużej rutynie grę kończy się dość szybko.
BLOCKOUT Rainbow Arts zręcznościowa	Jednak jest polski akcent — gra została napisana w Polsce dla California Dreams. Wielki przebój wykorzystujący zasadę słyn- nego TETRISA, lecz „studnia” jest przestrzenna, klocki trójwy- miarowe i dodatkowo potrzebna jeszcze jedna składowa do wyobrażenia. Oczywiście niezbędna także zręczność. Gra go- dna polecenia dla każdego od lat 7.
COLORADO Palace przygodowa	Właściwie każdy w dzieciństwie marzył, aby znaleźć się na Dzi- kim Zachodzie. COLORADO to znakomita grafika, świetne efekty dźwiękowe i przede wszystkim dobra zabawa. Najpierw przeprawa przez rzekę, potem wioska Indian i... można hand- lować, można wojować i można się znakomicie bawić.
WINGS OF DEATH Thalion GmbH zręcznościowa	Na pozór zwykła strzelanina jakich wiele, coś leci, coś strzela, coś się broni, ale jeżeli chwilę pogramy, to okaże się, iż gra wyrasta ponad wiele podobnych. Przede wszystkim została ona napisana dla serii STE, mamy więc znakomity dźwięk ste- reo, świetną grafikę i mimo wszystko wciągającą fabułę.
JUMPING JACK SON Infogrames zręcznościowa	Gra trochę zręcznościowa, trochę edukacyjna, trochę logiczna, przeznaczona raczej dla małych dzieci, ale myślę, że i dorośli mogą się znakomicie bawić. Wspaniała „dowcipna” grafika, świetny dźwięk, który odgrywa także ważną rolę w programie. Sympatyczny bohater porusza się po planszy dopasowując od- powiednie płyty do właściwych gramofonów.
F-29 RETALIATOR Ocean symulacyjna	Jeden z wielu programów symulacji lotniczej, trudno wybrać najlepszy z nich, wszystko jest kwestią gustu, gdyż każdy ma doskonałą grafikę i wiernie odzwierciedla lot danym samolo- tem. Panuje jednak przekonanie, że właśnie ten program — obok F-19 STEALTH FIGHTER — należy do najlepszych.
KICK OFF II Anco sportowa	Program, którego nie trzeba rekomendować. Ci, którzy lubią piłkę nożną, od pierwszych sekund obcowania z programem nie potrafią się rozstać z joystickiem ci, którzy nie lubią, zaczy- niają darzyć ją sympatią. Olbrzymia liczba wariantów, niezliczo- ne możliwości i wierność symulacji pozwalają ocenić ten pro- gram jako jeden z najlepszych.
SIM CITY Infogrames ekonomiczna	Przedstawiciel — choć nie klasyczny — bardzo mocnych osta- tnio programów. SIM CITY to miasto, które sam budujesz, roz- wijasz i którym zarządzasz na przestrzeni lat. Może cię spotkać to samo, co zdarza się w rzeczywistości: klęski żywiołowe, wojny, olbrzymie wydatki i nagłe fortuny. Bardzo dobre opraco- wanie graficzne i świetnie rozwiązane warianty gry pozwalają na spędzenie długich godzin przed komputerem.
LIGHT CORRIDOR Infogrames zręcznościowa	Zasada gry jest bardzo prosta — sterujemy piłeczką w pozi- mym korytarzu. Naszym zadaniem jest jak najszybsze prze- mieszczanie się do przodu. Brzmi to banalnie, ale gra wciąga niczym TETRIS. Znakomite opracowanie dźwiękowe bardzo pomaga w grze. Aby zrozumieć, co to za nowa „choroba”, trzeba po prostu zagrać.

W ten sposób zapoznaliśmy się z całą dwunastką, na zakończenie więc trzy
obietane nowości, na które niecierpliwie będziemy czekali:

SIMULCRA (Microstyle) — znakomita graficznie (grafika trójwymiarowa) symu-
lacja promu kosmicznego. Podstawowym atutem programu jest prawie doskonały
sposób jego obsługi. ST ACTION ocenił wartość tej gry na bliską ideału: 91%.

SPEEDBALL II (Image Works) — jest to gra sportowa przyszłości, trochę
przypominająca współczesne rugby. Nowa wersja tego programu znacznie
przewyższa i graficznie, i dźwiękowo poprzednią. Oceny „ST ACTION” podob-
nie jak dla poprzedniej gry powyżej 90%.

TOYOTA RALLY (Gremlin) — oczywiście rajd samochodowy, jednak różny
od dotychczasowych. Autorzy programu główny nacisk położyli na wierność
symulacji jazdy w różnych warunkach atmosferycznych, a ponieważ towarzyszy
temu sugestywna grafika, więc czeka nas znakomita zabawa, znacznie bezpie-
czniejsza niż za kierownicą prawdziwego samochodu.

Tomasz MAZUR

192 KB PAMIĘCI W 65XE



Najczęściej spotykane modele komputerów Atari to 800 XL, 65XE i 130XE, jak również 800XE, który jest niczym innym jak komputerem 65XE zmontowanym na płycie przeznaczonej dla 130XE.

Zasadnicze różnice dzielą tylko jeden model od pozostałych, a jest nim 130XE, mający dodatkowe 64 KB pamięci RAM.

Jak to jest możliwe, że ośmiobitowy procesor mający szesnaście wyprowadzeń adresowych, co pozwala mu na zaadresowanie 65536 komórek, może korzystać z tak dużej pamięci? Otóż jest to możliwe dzięki podmienianiu części podstawowej pamięci na część pamięci dodatkowej. Wielkość tego wymiennego obszaru została ustalona na 16 KB, a jego lokacja od adresu \$4000 do \$7FFF. Konsekwencją takiego stanu rzeczy jest podział każdej dodatkowej pamięci na szesnastokilobajtowe bloki. Łatwo zatem obliczyć liczbę takich bloków: gdy komputer ma zainstalowane dodatkowo 64 KB, jest ich cztery; gdy dodane jest 128 KB, mamy osiem bloków itd. Przełączenie pamięci podstawowej z obszaru \$4000—\$7FFF na jeden z bloków dodatkowej wykonuje automatycznie specjalny układ scalony, który wykrywa, czy dostępu do pamięci dodatkowej żąda główny procesor, czy też procesor obrazu (ANTIC).

Dodatkowa pamięć w komputerze nie byłaby do niczego potrzebna, gdyby nie programy umiające ją spożytkować. Podstawowy sposób jej wykorzystania to użycie jej do stworzenia tzw. ramdysku, czyli symulacji dodatkowej stacji dysków niejako wewnątrz komputera. Poza tym istnieją programy bezpośrednio używające dodatkowej RAM.

Jedno z praw Murphy'ego mówi, że każdy komputer ma za mało pamięci i dlatego namawiamy użytkowników „małych” Atari do samodzielnego dobudowania 64 lub 128 KB pamięci. Do wykonania takiego rozszerzenia potrzebne są przede wszystkim pamięci dynamiczne takiego samego typu, jakie znajdują się już w komputerze, czyli jednobitowe 4164 lub czterobitowe 4464 oraz kilka układów scalonych TTL serii LS. Układ sterujący można wykonać na płycie uniwersalnej lub drukowanej, którą należy zrobić według rysunków 1 i 2. Potrzebne będzie jeszcze trochę cienkich przewodów w izolacji do podłączenia sygnałów i oczywiście odpowiednie narzędzia.

OPIS WYKONANIA

Przed przystąpieniem do pracy z lutownicą należy odpowiednio przygotować układy pamięci RAM przez wygięcie do góry nóżek, do których doprowadzony będzie potem sygnał CAS. Tak przygotowane kostki nakładamy na już istniejące tak, aby można było zlutować ich wyprowadzenia o tych samych numerach. Po przylutowaniu pamięci należy

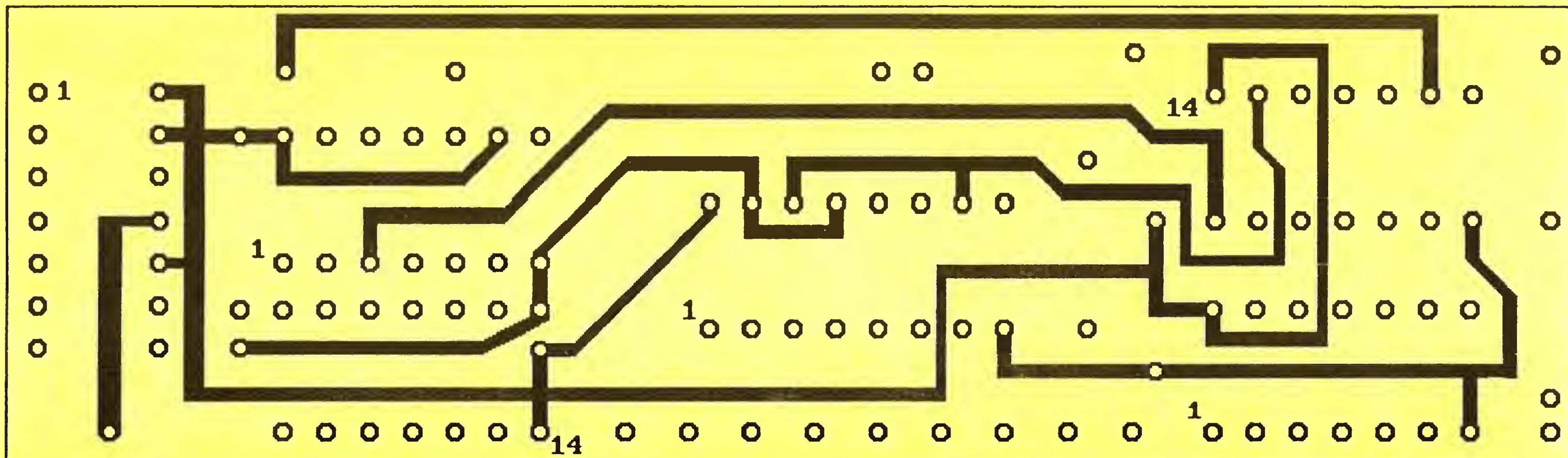
połączyć ze sobą ich wygięte do góry nóżki. Można powiedzieć, że jest to pierwsza faza wykonania rozszerzenia, a poprawność jej realizacji można sprawdzić przez dołączenie wygiętych wyprowadzeń do napięcia +5V i włączenie komputera, który powinien normalnie zadziałać. Jeżeli nie, to trzeba szukać zwarców pomiędzy lutowanymi nóżkami układów pamięci. Gdy dodajemy do komputera 128 KB RAM, to drugi blok pamięci umieszczamy w pierwszym dokładnie według tych samych zasad i tak samo sprawdzamy poprawność montażu.

Sprawa przedstawia się trochę inaczej, jeżeli mamy do czynienia z komputerem, na którego płycie zostały przewidziane miejsca na dodatkowy bank pamięci — chodzi tu o modele 800XE i ostatnio sprzedawane w sklepach Pe-wexu 65XE. W takim przypadku dodatkowy rząd układów pamięci (bez żadnego przygotowania) wlutowujemy w miejsca dla nich przeznaczone, a z otworów dla układu scalonego U34 wyciągamy trzy zwory oznaczone jednym czarnym pasmkiem każda.

Układ sterujący, zmontowany według rysunku 3 na przygotowanej uprzednio płycie drukowanej, można osadzić bezpośrednio na głównym procesorze CPU, lutując punkty oznaczone literami od „A” do „F” do odpowiednich jego wyprowadzeń.

W modelach serii XE i niektórych XL do multipleksowania adresów dla pamięci służy specjalizowany układ scalony wielkiej skali integracji o poto-

Rys. 1



WIDOK PŁYTKI DRUKOWANEJ OD STRONY ELEMENTÓW SKALA 2:1

cznej nazwie FREDIE, natomiast w pozostałych modelach XL funkcję tę spełniają dwa układy TTL 74LS158. Nóżki układu multipleksującego, do których doprowadzone są adresy A14 i A15, należy odłączyć od płyty komputera uważając, aby ich nie ułamać, a następnie podłączyć je do punktów układu sterującego, oznaczonych na schemacie

A14 i A15.

FREDIE jest też źródłem sygnału CAS, w komputerach bez tego układu CAS jest wytwarzany przez zespół TTL-i sterowanych między innymi przebiegiem 02 przepuszczonym przez linię opóźniającą. Do pamięci sygnał CAS doprowa-

pamięci dołączamy do punktu **CAS BANK 1**.

Doprowadzenia sygnałów do wejść układu sterującego należy wykonać według tabeli 1, w której uwzględnione są różne modele Atari. Wyjścia układu sterującego należy dołączyć do komputera według tabeli 2. Podłączenie pozostałych wyjść sygnałów CAS zostało opisane wcześniej.

OPIS DZIAŁANIA

Generalnie rzecz biorąc, do wytwarzania sygnałów dla układu sterującego dodatkową pamięcią zostały użyte nie wykorzystane wyjścia portu B układu scalonego PIA (6520).

UKŁAD STERUJĄCY	K O M P U T E R							
	800XL bez FREDDIE				800XL z FREDDIE, 800XE, 65XE			
	SYGNAŁ WEJŚCIOWY	PIN	NR. UKŁ.	NUMER KATALOG.	NAZWA	PIN	NR. UKŁ.	NUMER KATALOG.
HALT	35	U8	C014806	6502 CPU	35	U8	C014806	6502 CPU
Ø2	39	"	"	"	39	"	"	"
A14	24	"	"	"	24	"	"	"
A15	25	"	"	"	25	"	"	"
PB2	12	U23	C014795	6520 PIA	12	U23	C014795	6520 PIA
PB3	13	"	"	"	13	"	"	"
PB4	14	"	"	"	14	"	"	"
PB5	15	"	"	"	15	"	"	"
PB6	16	"	"	"	16	"	"	"
CAS	8	U30	----	74LS51	35	U6	C061991	FREDDIE
RAS	6	"	"	"	33	"	C061922	"

TABELA 1

UKŁAD STERUJĄCY		K O M P U T E R							
SYGNAŁ WYJŚCIOWY		800XL bez FREDDIE				800XL z FREDDIE, 800XE, 65XE			
		PIN	NR. UKŁ.	NUMER KATALOG.	NAZWA	PIN	NR. UKŁ.	NUMER KATALOG.	NAZWA
A14'		13	U27	----	74LS158	23	U6	C061991 C061922	FREDDIE
A15'		10	"	----	"	24	"	"	"

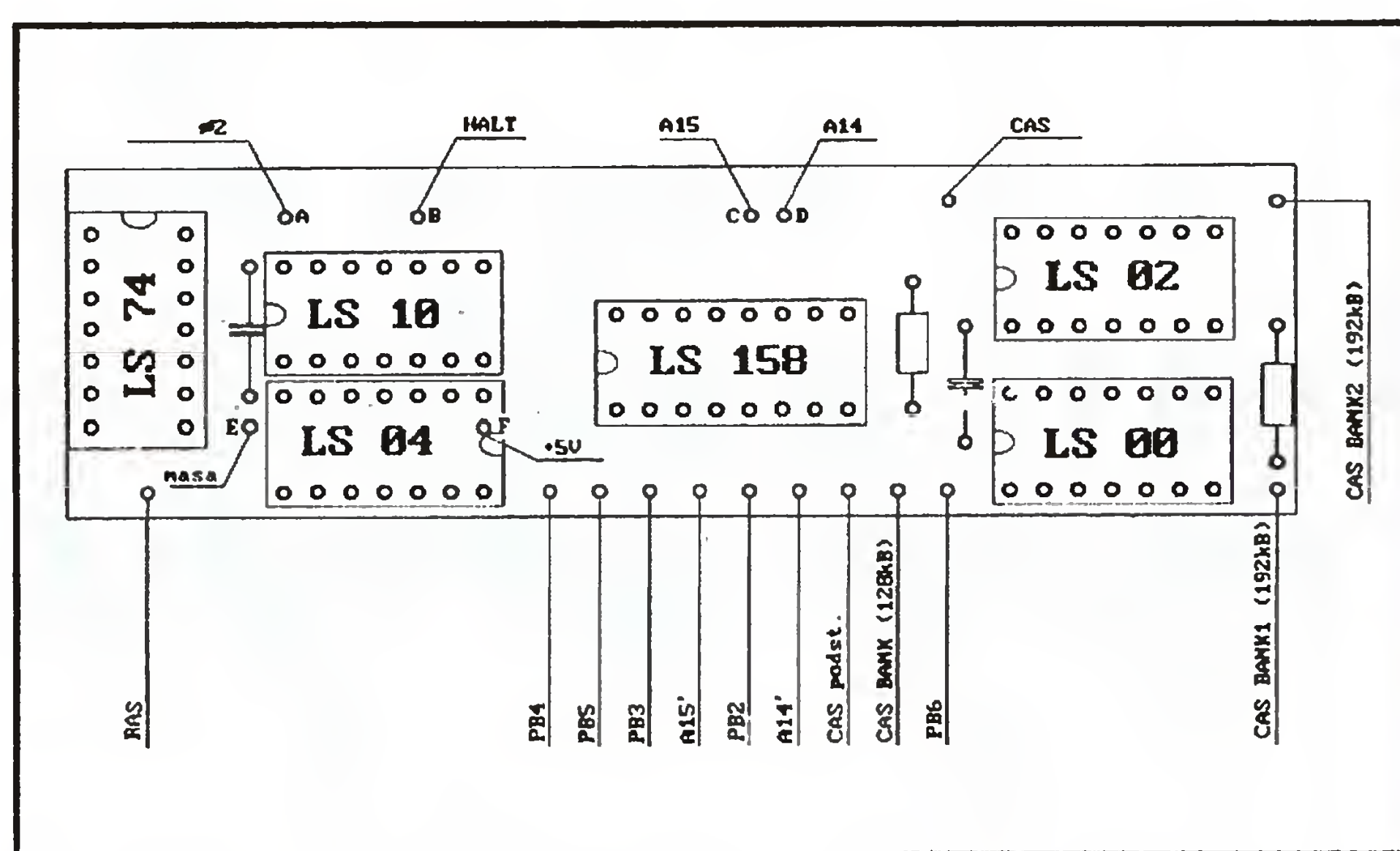
TABELA 2

dzony jest przez rezystor o wartości 33 Ohm. Końcówkę tego rezystora, połączoną ze źródłem CAS, trzeba wylutować z płyty komputera i podłączyć do punktu **CAS BANK** układu sterującego, a jeżeli dodajemy 128 KB RAM, to rezystor łączymy z **CAS BANK 2**. Połączone ze sobą wejścia CAS dodatkowego rzędu układów pamięci łączymy z punktem **CAS podstawowy** układu sterującego, a w przypadku rozszerzenia o 128 KB wejścia CAS drugiego rzędu układów

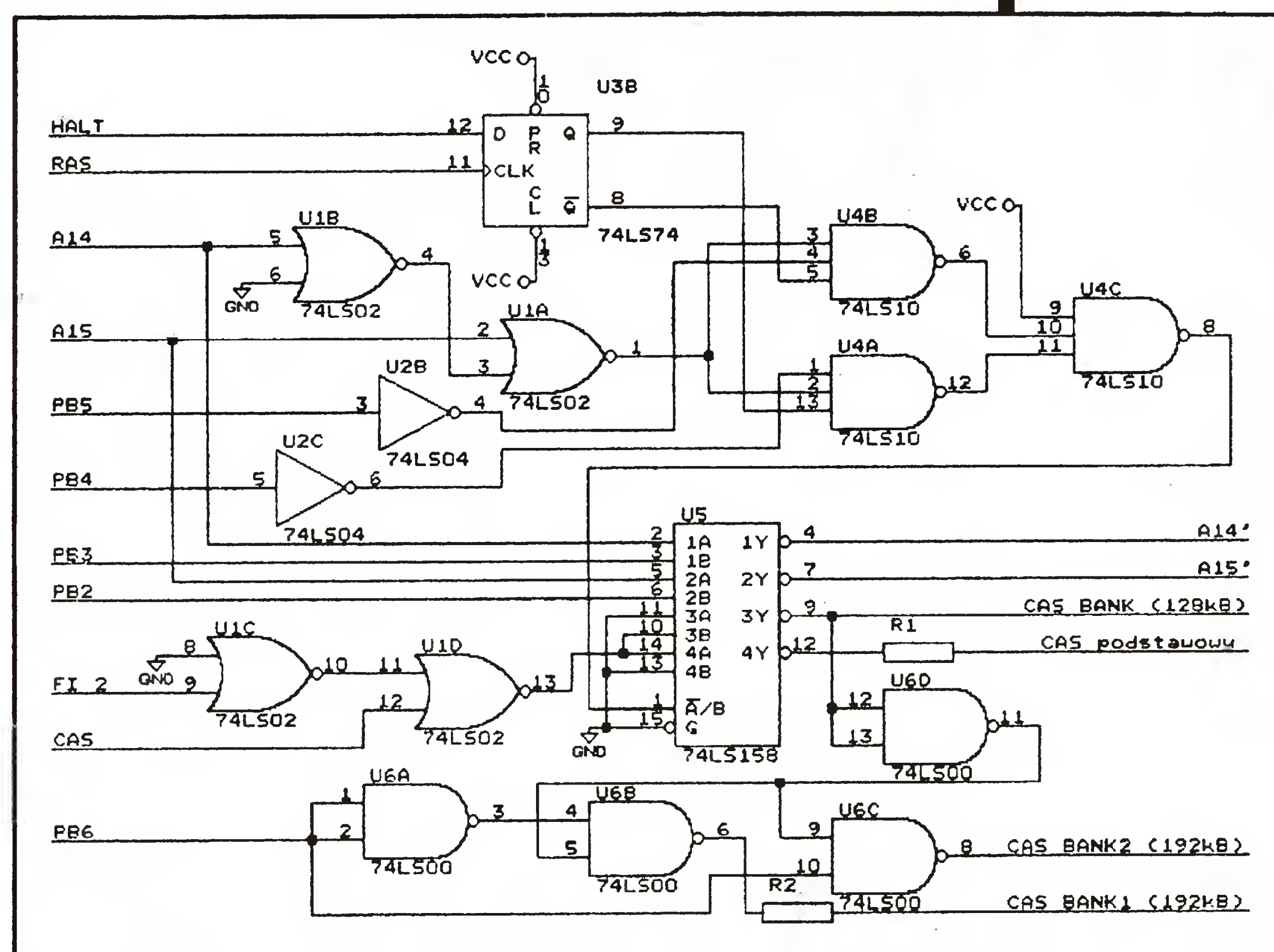
I tak skasowanie bitu PB4 lub PB5 zezwala CPU lub ANTIC-owi na dostęp do dodatkowego bloku pamięci. Bity PB2, PB3 i PB6 wyznaczają numer bloku pamięci, który ma być użyty. Dzięki doprowadzonym sygnałom HALT i RAS układ sterujący jest w stanie rozpoznać, czy adresy wysłane na szynę pochodzą od CPU czy też od ANTIC-a. Adresy A14 i A15 są dołączone do układu bramek wykrywającego kombinację stanów A14-wysoki — A15-niski, która występuje tyl-

ko wtedy, gdy na szynę adresową wysłany jest adres z zakresu \$4000—\$7FFF. Gdy zostaną spełnione wszystkie warunki dostępu do dodatkowej pamięci, na wejściu multipleksera U5 pojawia się sygnał przełączający CAS na bank dodatkowej pamięci. Szczegółowo można to prześledzić korzystając ze schematu ideowego pokazanego na rysunku 4.

Mariusz GEISLER



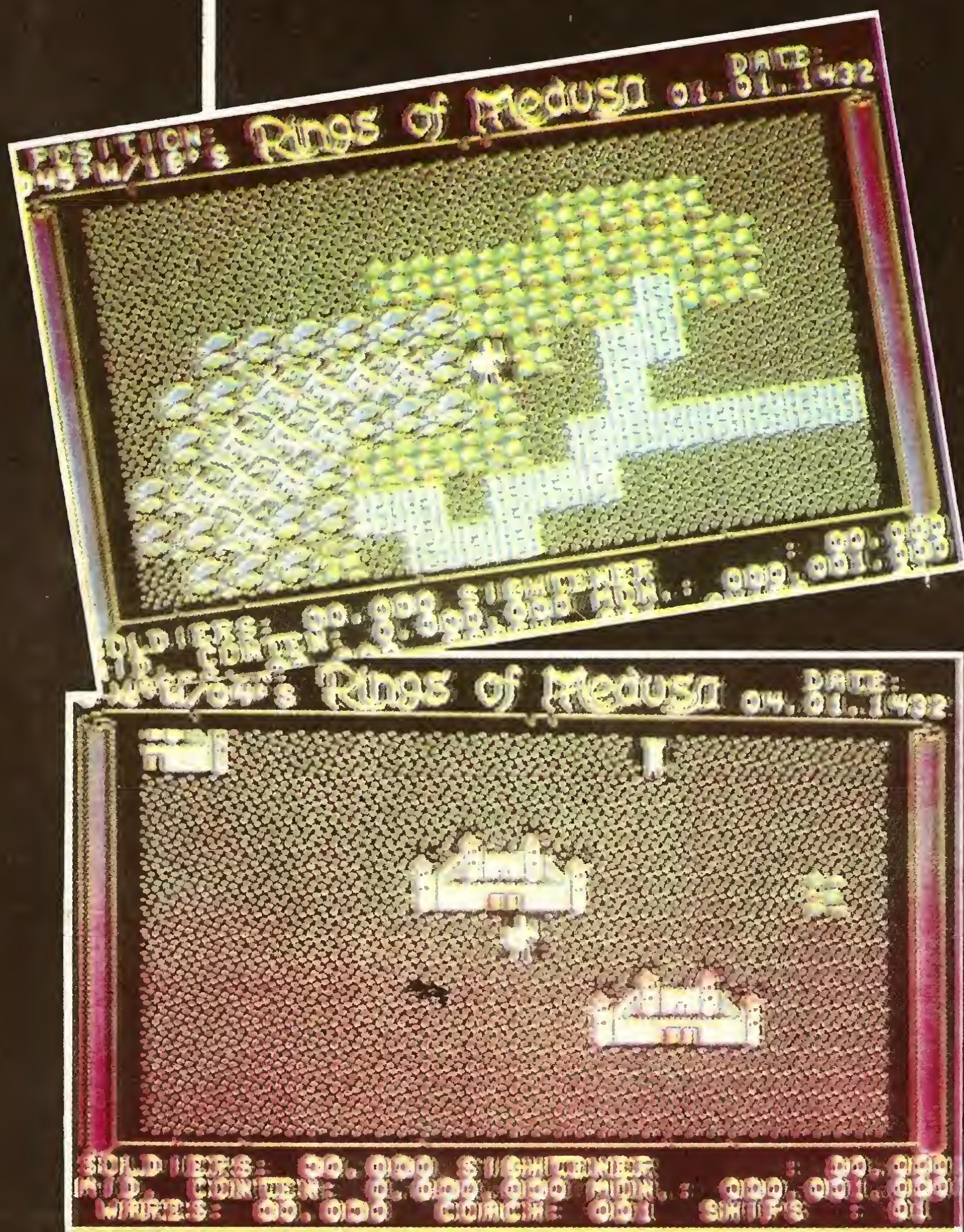
Rys. 3
Schemat
montażowy



GRUCZAKI

RINGSST

of MEDUSA



Gier komputerowych jest już bez liku, a ciągle powstają nowe. Coraz trudniej jest więc ich autorom wymyślać nowe scenariusze, choć przesigają się w pomysłowości. Na szczęście historia ludzkości, mity i legendy różnych narodów kryją jeszcze wiele pomysłów. W ostateczności zawsze znajdzie się jakaś obca, nieznana planeta oraz zły czarownik lub podobny, paskudny potwór.

W największym skrócie tak właśnie przedstawia się scenariusz „Pierścieni Meduzy”. Mamy tu nieznana planetę, wrędną boginię i oczywiście upadające królestwo, które koniecznie trzeba uratować. Co z tego wynika dla gracza? To co zwykle: walka, handel, hazard i praca. Naturalnie odpowiednie połączenie tych elementów może dać (i zwykle daje) pożądany efekt, czyli dużą liczbę sprzedanych gier, a więc zysk dla producenta.

Nie oznacza to jednak, iż chodzi tu o zwykłe naciąganie klientów. Większość takich gier jest

naprawdę interesująca i można przy nich spędzić niejedną wieczór. Osobiście lubię gry tego typu, lecz drażnią mnie już ci królewicze, czarownicy, smoki itd. Także „Rings of Medusa” jest grą, której rozgrywanie sprawiło mi przyjemność, choć starałem się nie myśleć o występujących w niej legendarnych postaciach.

Zadaniem gracza jest tu powstrzymanie wojsk złej bogini Meduzy niszczących królestwo i jego jedność. Cel ten zostanie osiągnięty po zwycięskiej walce z Meduzą. Aby zmusić Meduzę do walki, trzeba najpierw odszukać pięć magicznych pierścieni i zanieść je do świątyni.

Do odszukiwania pierścieni niezbędne jest posiadanie armii, lecz w królewskiej szkatule dno świeci pustką. Dzielnego gracza musi więc rozpoczynać od mało ambitnych zajęć: handlu lub pracy w kopalni. Można także zająć się czymś znacznie przyjemniejszym, na przykład napadaniem na wędrownych kupców, zdobywaniem i łupieniem miast lub — w ostateczności — zwykłym hazardem. Włóczęgę po malowniczej okolicy i bacznie rozglądając (za pomocą słono opłacanych zwiadowców), można wykryć skarby, które zasila kiesę gracza. Niestety czasem zamiast na skarby trafia się na armię, która wcale nie przejawia przyjacielskich zamiarów.

Powyższy opis nie ukazuje jednak wszystkich trudności. W rzeczywistości gra jest skomplikowana i wymaga trochę zastanowienia. Na przykład organizując armię, możemy utworzyć siedem różnych rodzajów oddziałów (piechota, łucznicy, kawaleria, artyleria itd.), a do każdego z nich zaciągnąć dziesięć różnych rodzajów żołnierzy (ludzi, elfy, karły, orki itp.). Dodatkowo trzeba uwzględniać stan swoich finansów, aby po dwóch-trzech miesiącach z trudem skompletowana armia nie rozpierzchnęła się z powodu... załegłego żołdu. A część pieniędzy pochłonie z pewnością wykształcenie wojsk.

Podobnie skomplikowaną czynnością jest handel. Zanim zrobisz jakieś zakupy, musisz się dokładnie zorientować w cenach, aby nie przepłacić za towar, który w sąsiednim mieście może być znacznie tańszy. Odwrotna sytuacja wystąpi przy sprzedawaniu — należy szukać miasta, gdzie posiadane dobra będzie można zbyć jak najdrożej. Niekiedy wsparcia udzieli bank, lecz musisz być dla niego wiarygodnym dłużnikiem. Byłe łachudra nie ma szans na najmniejszą pożyczkę.

W walce zamiast zmysłu handlowego trzeba wykazać się zdolnościami taktycznymi. Trzeba tu bowiem rozpoznać siłę

przeciwnika, właściwie rozstawić swoje jednostki i manewrować nimi odpowiednio do rozwoju sytuacji. Pamiętaj też, że walka nie jest nieunikniona — można przedtem próbować negocjacji. Podobna sytuacja ma miejsce przy zdobywaniu miast i podczas walk między okrętami. Tak, tak — obszar gry obejmuje również morza i przyda się choć niewielka flota.

Pierwszoplanowym celem jest oczywiście odszukanie pierścieni. Część z nich znajduje się w ukrytych skarbach (również na wyspach, więc flota jest niezbędna), a inne są w posiadaniu wrogich armii. Pierścienie są oczywiście magiczne i mają następujące właściwości:

- 1 — ujawnia siłę napotkanej wrogiej armii,
- 2 — ujawnia ceny na dobra dostępne w różnych miastach,
- 3 — daje zdolność rozpoznawania terenu bez zwiadowców,
- 4 — wzmacnia półtorakrotnie siłę własnej armii,
- 5 — wywołuje Meduzę i umożliwia stoczenie z nią ostatecznej walki.

Odnalezione lub zdobyte pierścienie należy zanieść do świątyni (w każdym mieście jest jedna) i umieścić w znajdującej się tam gwiazdzie, a dokładniej — w jej promieniach.

Obsługa gry jest bardzo łatwa — do sterowania postaciami i armiami wykorzystuje się mysz. Także za pomocą myszy wybiera się różne warianty z menu znajdującego się w dolnej części ekranu. Rozgrywce towarzyszy przyjemna muzyka i — naturalnie — odpowiednie efekty dźwiękowe. Jeśli ktoś woli ciszę, aby spokojnie się zastanawiać, nic nie stoi na przeszkodzie — menu zawiera również warianty wyłączania i włączania muzyki i innych efektów (oddzielnie).

„Rings of Medusa” może więc zadowolić każdego gracza, pod warunkiem wszakże, iż lubi on tego rodzaju programy. Niestety posiadacze monitorów monochromatycznych nie będą mieli z niej żadnego pożytku, korzystnie zaś z telewizora nie jest najszcześniejszym rozwiązaniem, gdyż trudno na nim zauważyć niektóre drobne szczegóły. W tym miejscu pragnę przeprosić Czytelników, gdyż z powodu awarii redakcyjnego monitora kolorowego prezentowane zdjęcia musiały być wykonane na komputerze Commodore.

(ziew)

Producent: Starbyte Software

Ocena (w skali 1—10):
GRAFIKA: 8
DŹWIEK: 6
SENS GRY: 8

FRED

XL/XE

Słońce stało wysoko na niebie, kiedy wymęczony wczorajszym polowaniem, wreszcie wstałeś. Nagły impuls rozbudził cię ostatecznie — przecież dziś umówiłeś się z uroczą Neandertalką na wernisażu Scrawlera — najbardziej cenionego przedstawiciela surrealistycznego malarstwa ściennego.

Jego Jaskinia Sztuki jest miejscem elitarnych spotkań miejscowej śmietanki towarzyskiej. Szybko wypomadowałeś włosy żółtkiem z jaja pterodaktyla, wdziałeś odświętną skórę, i wyruszyłeś na spotkanie z kulturą, nie przypuszczając nawet, jakie przeszkody spiętrzyły się na twojej drodze.

Autor gry musiał mieć fachowego konsultanta historycznego, gdyż realia epoki są w pełni zachowane. Twoim podstawowym uzbrojeniem są niewielkie porę-

czne, i, wbrew pozorom, szalenie efektywne kamienie. Wystarczy dobrze wycelować i agresywne wielkie kraby żałośnie rozplaszczą się na piasku. Równie prosto można strącić ptaszka, który zgoła odmiennie od ciebie zapatrzuje się na sprawę układania włosów na żółtko z jego jajka. Uważaj także na śmiercionośne kaktusy — przytulenie się do takiej roślinki może kosztować życie, choć istnieje sposób, by się ochronić.

Nie przeocz niewielkich dzbanów, rozrzuconych w różnych dziwnych miejscach. Mimo że są to zwykłe domowe naczynia, bynajmniej nie kryją w sobie kaszy. Jeśli względnie precyzyjnie rzucisz kamieniem, możesz znaleźć wśród skorup tak ciekawe rzeczy, jak spray przeciwko wszystkim, co żywe, czy też głośno wybuchający dynamit. Równie istotne są kupki kamieni, bez których nie masz żadnych szans przeżycia.

Ogólne wrażenie, jakie sprawia gra, jest bardzo dobre. Staranna grafika, dokładna animacja i całe mnóstwo dobrze przemyślanych,

ciekawych niespodzianek na kolejnych etapach gry sprawiają, iż Fred nudzi się nieprędko. Ja sam zdradziłem dla niego nawet ukochane Silent Service, zamieniając szum fal Pacyfiku na chrzęst piasku półpustyni. Efekty dźwiękowe są skąpe, lecz realistyczne, co jest i tak dużo bardziej strawne od lecących „w kółko” kilku taktów melodii. Akcja jest spokojna (lecz nie powolna!), dzięki czemu grający we Freda nie notują przypadków połamania joysticka.

Za ukończenie Freda do określonej daty producent przewidział nagrodę w postaci swoich nowych programów. Termin co prawda dawno już minął, redak-

cyjni gracze w kwestii wygrania konkursu zawiedli, lecz i tak nowych gier z Laboratorium Komputerowego „Avalon” nadal oczekujemy z niecierpliwością.

Master

Producent: Laboratorium Komputerowe „Avalon”
Autor: Mirosław Liminowicz
Rok produkcji: 1990

Ocena (w skali 1–10):
GRAFIKA: 8
DŹWIEK: 4
SENS GRY: 5



Wielki Konkurs!

Niniejszym ogłaszamy nieustający konkurs dla czytelników naszego pisma. W pierwszym jego wydaniu można wygrać następujące przedmioty ufundowane przez firmę „TAL” z Warszawy:

- 1 joystick SV 127
- 10 joysticków SV 120
- 5 joysticków SV 119
- 3 wycinarki do dyskietek 5.25"
- 2 pudełka po 10 dyskietek 5.25"
- 1 pudełko na dyskietki 5.25"

W konkursie może wziąć udział każdy, kto zechce

odpowiedzieć na kilka pytań i wypełni dokładnie zamieszczoną obok ankietę. Nie zależy nam, aby pytania konkursowe były zaskakująco trudne, ale raczej nie wykraczające poza wiadomości dostępne w polskiej literaturze komputerowej.

Pod każdym pytaniem znajdują się trzy odpowiedzi, lecz tylko jedna z nich jest prawidłowa. Wystarczy ją podkreślić.

Tym razem będzie kilka pytań z ogólnej wiedzy o komputerach z rodziny Atari. Polecamy, jako materiały źródłowe, poprzednie wydania „Moje Atari” oraz specjalne wydanie „Bajtki — tylko o Atari”.

1 Co oznacza znak firmowy Atari?

- słowo Atari w języku japońskim;
- stylizowany symbol góry Fuji;
- przenośnia pokazująca wyższość komputerów Atari.

2 Co znaczy słowo Atari?

- imię żony założyciela firmy;
- sytuację bez możliwości dalszego ruchu w grze „GO”;
- imię japońskiego bożka.

3 Kto założył firmę Atari?

- Lucjan Wencel;
- Jack Tramiel;
- Nolan Bushnell.

4 Ile klawiszy ma komputer Atari 400?

- 60 i 5 funkcyjnych;
- 57 i 4 funkcyjne;
- 64 i 4 funkcyjne.

5 Jaka firma jest wyłącznym przedstawicielem Atari w Polsce?

- Karen;
- ABC Data;
- LDW.

ANKIETA

Imię i nazwisko:

dokładny adres:

posiadany komputer: Atari 800XL, 65XE, 130XE, 520ST, 1040ST, inny:

pamięć RAM posiadanego komputera:

stosowana pamięć masowa: magnetofon, turbo, stacja dyskowa.

wiek: lat.

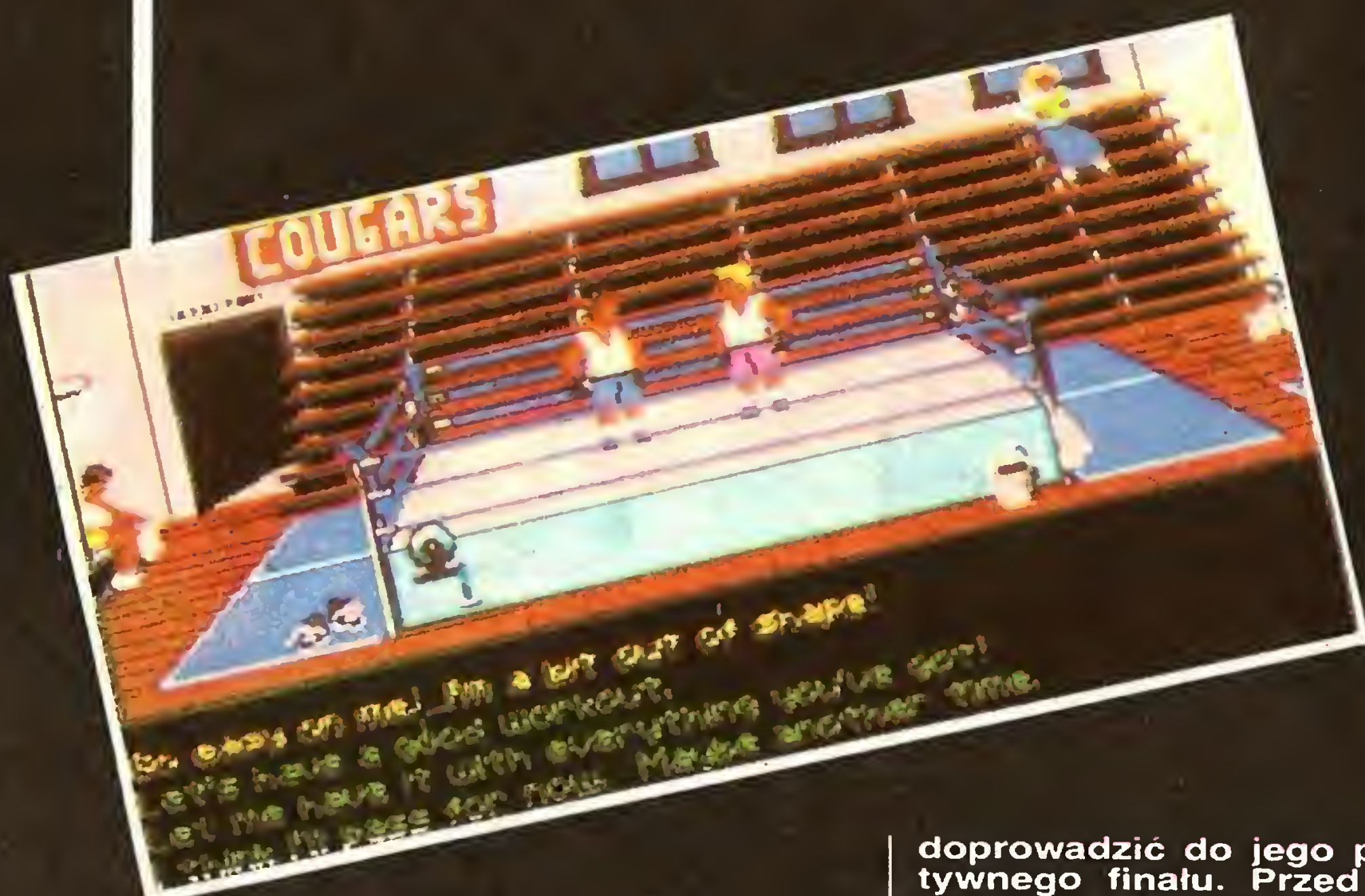
Uwaga!

Odpowiedzi przyjmujemy do dnia ukazania się kolejnego numeru „Moje ATARI”.

G
R
U

K
O
N
K
U
R
S

INDIANA JONES — The Last Crusade



Jeżeli byłeś w kinie na filmie „Indiana Jones — the last crusade” i chciałbyś przeżyć to, co tytułowy bohater, to masz teraz okazję.

Na rynku pojawiły się niedawno dwie gry komputerowe o tym tytule. Jedna z nich jest typowo zręcznościowa. Znacznie ciekawsza jest druga, czyli gra dla lubiących posiedzieć i pomyśleć. Od razu powiem, że postępując logicznie i opierając się na fabule filmu łatwo jest ją skończyć, tym bardziej że przed każdą niespodzianką można aktualny stan gry zapisać na dyskietce (przez naciśnięcie klawisza «F5») i powtarzając dany fragment

doprowadzić do jego pozytywnego finału. Przed rozpoczęciem gry dobrze mieć pod ręką słownik, ponieważ gra w znacznej części jest tekstowa. Na szczęście wszystkie teksty, którymi operuje Indiana, są wyświetlane na ekranie i po wybraniu odpowiedniego zdania wystarczy wcisnąć przycisk myszy.

Celem zabawy jest odnalezienie Świętego Graala. Gra rozpoczyna się na Uniwersytecie Barnetta w 1938 roku, kiedy to Indiana Jones powraca z wyprawy po krzyż Coronado. Kierowanie postacią bohatera jest proste i polega na wskazaniu kursorem miejsca, do którego ma się udać dr Jones, i naciśnięciu lewy przycisk myszy. Do dokładnego kontrolowania postaci służą wypisane u dołu ekranu polecenia, których selekcja następuje także przy użyciu myszy. Wszystkie polecenia, które mogą być wykonane, są wyświetlane w kolorze zielonym. Jeżeli napis ma kolor szary, to nie należy

brać go pod uwagę. Na przykład, gdy obok naszego bohatera stoi jakaś postać — możemy z nią porozmawiać naprowadzając kursor na polecenie „TALK”, a następnie na tę postać. Dzięki temu często możemy uzyskać ważne informacje, ponieważ gra miejscami odbiega treścią od filmu. Między innymi na początku Indiana ma okazję pocwiczyć boks — co należy wykorzystać. Sugeruję wybór najtrudniejszego stopnia walki. Wynikiem tego — po znokautowaniu przeciwnika — będzie większa ilość energii w czasie dalszej gry. Podczas tej walki wykorzystuje się klawisze «1», «3», «4», «6», «7» i «9» z bloku numerycznego. Wybór klawiszy powinien odpowiadać miejscu, gdzie spadnie cios, tak więc jeżeli przeciwnik będzie stał po naszej prawej stronie, do ataku używać należy klawiszy «3», «6» i «9», a do cofania się — klawisza «4». Analogicznie postępujemy, kiedy przeciwnik podejrze z lewej strony — używamy do ataku klawiszy «1», «4» i «7», do odwrotu zaś klawisza «6».

Niemal niezbędnym warunkiem ukończenia gry jest wzięcie przez Indianę ze swego pokoju na Uniwersytecie pamiętnika ojca, w którym zawarte są wszystkie informacje na temat Świętego Graala (GRAIL DIARY). Pamiętnik znajduje się na biurku pod sterą listów. Niekiedy gra toczy się sama. Czasem po wykonaniu w jakimś miejscu zadania nie wiesz, co dalej robić. Wtedy wybierz wariant „TRAVEL”. Mając w pamięci przebieg przygód w filmie, możesz dzięki otrzymanym w ten sposób informacjom określić dalszy cel wędrówki. Podczas gry trzeba zwracać uwagę na wszystkie szczegóły i kojarzyć je ze sobą, gdyż wszystkie one są istotne.

Rozpatrzmy teraz przykładowy tok rozumowania: jeśli napotkasz w podziemiach katedry w Wenecji salę, w której znajduje się zatknięta w ścianie pochodnia (TORCH), to wydaj polecenie „LOOK TORCH” (obejrzyj pochodnię). Dostaniesz informację, że pochodnia jest sucha (DRY). Musisz zatem znaleźć butelkę, aby nabrać naftę, którą zwiłżysz pochodnię, co umożliwi jej zapalenie. Jeżeli nie będziesz miał pomysłu, co zrobić będąc w jakiejś sali, to zajrzyj do pamiętnika ojca (LOOK TO GRAIL DIARY). Zawsze istnieje szansa, że znajdziesz tam wskazówkę.

Grafika nie jest mocną stroną gry — przeznaczony jest na nią tylko fragment ekranu, gdyż pozostałą część zajmują napisy. Również efekty dźwiękowe nie są bogate, lecz brakuje im zbyt wiele przeszkadzają, ponieważ istota gry wymaga spokojnego zastanowienia. Ponadto brakuje im w znacznym stopniu rekompensowane przez prostotę obsługi i łatwość opanowania zasad sterowania przebiegiem gry. Nie ma przy tym ograniczenia czasu, więc jest to zabawa na ładnych parę godzin. Oczywiście autor pomyślał i o tym — wcisnięcie klawisza spacji wstrzymuje grę i umożliwia rozprostowanie kości, ponowne jego naciśnięcie wznowia akcję. (jg)

Producent: Lucasfilm Games
Autor: Ron Gilbert

Ocena (w skali od 0 do 10):
GRAFIKA: 6
DŹWIĘK: 5
SENS GRY: 10

UWAGA! Firma TOMS

mającą na koncie opracowanie szeregu usprawnień komputerów i stacji dysków dla ATARI rozpoczyna wojnę z piratami komputerowymi, kopiującymi nielegalnie efekt naszej wielomiesięcznej pracy. W związku z tym prosimy wszystkich, którym gdziekolwiek poza naszą firmą zainstalowano usprawnienia stacji dysków mające cechy systemów TOMS TURBO i TOMS MULTI o przysłanie o tym informacji pod adres: TOMS, ul. Widok 14/1, 00-023 Warszawa.

Prosimy o informacje o zainstalowaniu usprawnień zapewniających w stacjach dysków trzykrotne przyspieszenie transmisji, możliwość kopiowania zabezpieczonych dysków, przenoszenie plików między ATARI i IBM.

Rozpocznijmy wspólnie walkę z piractwem, nie czekając na ustawę sejmową! Serdecznie dziękujemy, w zamian wyślemy informator o naszych usprawnieniach ATARI.

Równocześnie informujemy, że dysponujemy już lokalem w samym centrum miasta, koło Rotundy PKO, przy ulicy Widok 14/1, telefon (0-22) 27-16-01. Zapraszamy w godzinach 9—17. Polecamy wszystko do małego ATARI!

Sklepy firmowe

"ATARES"

polecają:

NAJTAŃSZE komputery COMMODORE, AMIGA, ATARI, instrumenty muzyczne (organy) firmy HOHNER, oprogramowanie, BLIZZARD TURBO, FLASH SYSTEM do LDW 2000, CA 2001 (format dyskietki 500 KB, transmisja 130 KB), cartridge systemowe i z gramami, monitory, drukarki, joysticki, dyskietki i inne akcesoria. Zapewniony serwis gwarancyjny i pogwarancyjny (CHORZÓW, TRUCHANA 35). Również realizacja zamówień hurtowych.

Zapraszamy:

"ATARES" Chorzów, Truchana 35, tel. 415-791 (hurt i detal)

"ATARES" Świętochłowice, A. Czerwonej 20 (detal)

"MIRAGE" Rybnik, Sobieskiego 7, tel. 212-42

"BIT" Racibórz, Szkolna 34
Racibórz, Browarna 2

"ABC ELECTRONICS" Gliwice, Wrocławska 7

"HOBBIT" Chorzów, Szczecińska 10

"KRAM" Bytom, PPR, tel. 81-65-29

"HERMES" Mysłowice, Wyspiańskiego 1

ATARI

- naprawy magnetofonów i zasilaczy do komputerów
- montaż systemu Turbo-Rom-Plus w magnetofonach Atari (około 100 gier na kasecie C60, Feud w 1 min. 21 sek., również praca w "Blizzardzie" - licencja programowa firmy "Atares" z Chorzowa, cartridge systemowe).

Honorujemy gwarancję firmową, na przerobione magnetofony.

- Montaż wejść monitorowych w OTV turystycznych

"PLUS"

Kraków, ul. Mochnackiego 67, tel. 33-23-12
godz. 10.00 - 18.00, w soboty 9.00 - 13.00

Punkty przyjęć:

Tarnów, ul. Traugutta 7/10, tel. 33-15-41

środy 16.00 - 18.00

Rzeszów, ul. Rejtana 43/6, tel. 548-82

środy 10.00 - 14.00

Informacje:

Sanok, Studio "Amicus", Pl. Św. Michała (MDK)

pon. - piątek 10.00 - 17.00, tel. 332-99 wieczorem

TURBO - ROM - PLUS

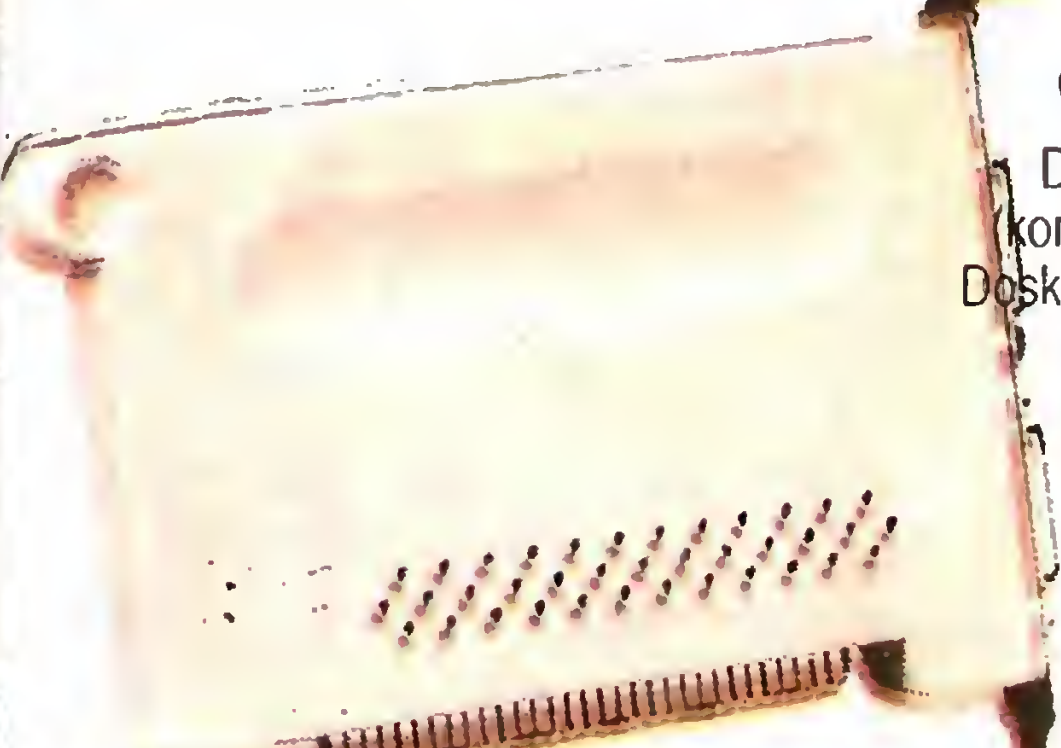
przeróbki magnetofonów Atari

"Atarom"

83-300 Pruszcz Gdański, ul. Dywizjonu 303 nr. 8/18
tel. Gdańsk 82-20-71 w. 447

Licencja Zakładu Elektr. "Plus"

**Wajtańsze
w Polsce!**

 SV 119 Junior 2 Fire 6 Blaszanych styków Prosty mechanizm	 SV 120 Junior-Stick 2 Fire 6 Blaszanych styków Uchwyt pistoletowy	 SV 122 Quickjoy II 2 Fire 6 Blaszanych styków AutoFire Drażek lotniczy
 SV 124 Turbo 6 Mikrostryków AutoFire Drażek lotniczy	 SV 123 Supercharger 2 Fire 6 Mikrostryków Ergonomiczna budowa Precyzyjny mechanizm	 SV 126 Jet Fighter 2 Fire 6 Mikrostryków AutoFire ACS-Regulator szybkości AUTO Obsługa pod kciuk" Drażek lotniczy
 SV 125 Superboard 6 Fire 10 Mikrostryków AutoFire Cyfrowy wyświetlacz czasu Sygnał dźwiękowy Przełącznik dla leworecznych Drażek lotniczy	 SV 130 IR Infrared 1 Fire 5 Mikrostryków Podczerwień Daleki zasięg Odbiornik	 SV 128 Megaboard 4 Fire 10 Mikrostryków AutoFire 6 cyfrowy stoper ATM — Anti Tilt Mechanism Fire Pad
 SV 140 Enterprice 2 Fire 6 Mikrostryków AutoFire ACS — Regulator szybkości AUTO Drażek lotniczy „kierownica” Kabel 4 m	 SV 201 Quickjoy M 5 Do IBM XT/AT (kompatybilnych) Współpracuje z Game-Card lub I/O Card 2 Fire 2 AutoFire 6 Mikrostryków Wybór AUTO PSC — Regulator XY Sygnalizacja Światlna Fire ASC — Regulator szybkości AUTO	 SV 202 M 6 analog Analogowy DO IBM XT/AT (kompatybilnych) Współpracuje z Game-Card lub I/O Card 2 Fire
 SV 210 Game Card Do IBM XT/AT (kompatybilnych) Doskonale pracuje z M 5 i M 6	 SV 127 Top Star 2 Fire 6 Mikrostryków AutoFire Przeźroczysta obudowa SAS — Shock Absorbing System Platinowane części	 SV 500 Van 3 Pudełko na dyskietki 80 sztuk 3 1/2" Zamknięcie na klucz SV 510 Van 5 Pudełko na dyskietki 80 sztuk 5 1/4" Zamknięcie na klucz

SV 119 — 64 900
SV 120 — 74 900
SV 122 — 89 900
SV 124 — 109 900

SV 123 — 119 900
SV 125 — 239 900
SV 126 — 169 900
SV 127 — 249 900

SV 128 — 319 000
SV 130 — 339 000
SV 201 — 249 000
SV 202 — 229 000

SV 210 — 219 000
SV 500 — 119 000
SV 510 — 119 000

**interwencje
tel. 23-92-21**

SV 201 + SV 210 — 439 000
SV 202 + SV 210 — 399 000

Należy przestać pieniądze przekazem zwykłym (czerwony blankiet) na adres:

TAL

Quickjoy

Uwaga! W miejscu na korespondencję prosimy podać po raz drugi: adres domowy, kod i symbol zamówionej pozycji. W przypadku braku adresu zwrotnego nie realizujemy zamówienia.

„TAL” — QWERTY
ul. Mikowa 45
02-411 Warszawa
fax: 659-12-35

Czas realizacji 14 dni

KUPUJEMY UŻYWANE ST

Kolorowe reklamy i pięknie wydane prospekty informują o kolejnych, coraz doskonalszych komputerach Atari. Co roku firma wypuszcza przynajmniej jeden

tym droższy. W zasięgu możliwości finansowych przeciętnej polskiej rodziny nie zawsze zmieści się nawet 520 ST, które w Pewexie kosztuje ponad 7.000.000 zł.

Kupujący musi usuwać wszystkie awarie na swój koszt. Często sprzedawane są znacznie taniej nowe komputery przywiezione z RFN, Wielkiej Brytanii lub USA. Do nich też dołączona jest karta gwarancyjna. Ale pamiętajmy, że **karta gwarancyjna wystawiona za granicą jest w Polsce nieważna**. Jeśli będziemy chcieli bezpłatnie zreperować komputer, to trzeba będzie się zgłosić na przykład do jakiejś firmy w RFN. A kogo stać na taką wycieczkę?

SPRAWDŹ, CO KUPUJESZ

Po tylu przestrobach na pewno wielu osobom przeszła ochota na kupowanie sprzętu na giełdach. Na szczęście ryzyko zakupu wadliwego komputera można znacznie zmniejszyć. Nie daj się zwieść lśniącym opakowaniem. Na-

cym ST (np. program „Memtest” lub „RAMtest”), programem do pomiaru prędkości obrotowej stacji dysków, edytorem tekstu oraz w czystą dyskię i śrubokręt.

Niektóre usterki ujawniają się już po włączeniu komputera do sieci. Należy więc poprosić właściciela, by zmontował wszystkie elementy zestawu i przystępujemy do pracy. Uszkodzony komputer może zresetować się samoistnie już po włączeniu lub w innym nieoczekiwanym momencie.

MONITOR

Na pierwszy ogień pójdzie monitor. Nieciągłość obrazu, poprzeczna smuga (jakby ktoś porysował ekran) może świadczyć o uszkodzeniu pamięci RAM komputera. Stan układów RAM sprawdzimy dokładnie za pomocą wcześniej wspomnianego programu testującego. Niektóre monitory mogą być podrasowane. Przez podkręcenie synchronizacji pionowej i poziomej można doprowadzić do zniknięcia nie wykorzystywanej ramki przy brzegach ekranu. Takie monitory zużywają się znacznie szybciej. Intensywniejsze wypalanie się kineskopu następuje także, jeżeli do komputera wbudowano układ podwyższający rozdzielczość „Autoswitch-Overscan”.

Stan monitora ocenimy na podstawie położenia pokręteł jasności i kontrastu. Jeżeli są one w krańcowych położeniach, a mimo to nie osiąga się zadowalającej poprawy jakości obrazu, to możemy być pewni, że kineskop nie pożyje już długo. Na zużytym kineskopie widać też wypalone ramki, teksty lub menu najczęściej używanego programu. Inne nieprawidłowe objawy to drgania, pływanie i zniekształcenia obrazu.

KLAWIATURA

Sprawdzenie klawiatury polega na ocenie jej zewnętrznego wyglądu i ocenie stopnia zużycia klawiszy. Najbardziej do tego nadaje się edytor tekstu. Próbowujemy po kolei działanie wszystkich klawiszy. Niektóre z nich mogą być szczególnie zmaltrretowane i nie reagują na delikatne naciśnięcie lub nawet zacinają się. Są to zwykle klawisze sterujące szczególnie lubianej gry. Więcej uwagi należy poświęcić klawiszom „RETURN” i „SPACE”. Należy przekonać się, czy prawidłowo reagują po naciśnięciu w środku i przy brzegach.

Podobnie sprawdza się klawisze myszy. Niektórzy energiczniejsi użytkownicy mogą doprowadzić je do opłakanego

dokończenie na str. 31



Atari 520 ST™

nowy model. I tak po — 520 ST pojawiły się: 1040 ST, Mega ST 1—4, Portfolio, laptop Stacy 1—4, 1040 STE, Mega STE, TT i cała rodzina komputerów osobistych Atari PC i ABC. W naszym magazynie systematycznie publikujemy opisy techniczne tych urządzeń. Nic dziwnego, że wielu czytelników zastanawia się nad kupnem jednego z tych cacek.

Ale jest jedna przeszkoda: pieniądze. Im lepszy komputer,

Nic dziwnego, że wiele osób poszukuje tańszych okazji. Takie możliwości stwarza giełda. Tu można kupić trochę taniej nowy lub używany komputer.

PROBLEMY Z GWARANCJĄ

Istnieje jednak pewne poważne niebezpieczeństwo, które sporą oszczędność może przerodzić w jeszcze większe wydatki. Na giełdzie całe ryzyko związane z zakupem uszkodzonego sprzętu ponosi nabywca. W sklepie otrzymujemy zawsze kartę gwarancyjną, w której podany jest adres punktu serwisowego. Jeśli w ciągu roku ujawnią się jakieś usterki, to serwis usunie je bezpłatnie. Na sprzęt kupowany na giełdzie nie dostaje się gwarancji.

wet fabrycznie nowy, jeszcze nie rozpakowany komputer (wprawdzie bardzo rzadko) może być zepsuty. Usterki fabryczne wychodzą zwykle w pierwszych dniach lub tygodniach użytkowania. Czasami zdarzają się przypadki, że nieuczciwy sprzedawca stara się pozbyć bubla, nie informując nabywcy o jego wadach.

KUPUJ BEZ POŚPIECHU

Jeśli chcesz uniknąć przykrości, to przed zawarciem transakcji włącz komputer do sieci. Nawet kilka minut pracy i uważnej obserwacji pozwala odkryć najpoważniejsze wady. Najlepiej jednak kupować sprzęt bez pośpiechu. Proponujemy umówić się ze sprzedającym i poświęcić trochę czasu na sprawdzenie ważniejszych elementów komputera. Przed spotkaniem należy zaopatrzyć się w dyskietkę z programem testują-

PCDitto

Każdy, kto potrafi myśleć obiektywnie i zetknął się z Atari ST, nie ma wątpliwości, że jest to komputer o bardzo ciekawych możliwościach, choć w rzadko którym programie do końca wykorzystywanych. Przy wszystkich swoich możliwościach ma jednak Atari jedną poważną wadę — nie jest standardem w żadnej dziedzinie. Osiągnięcie takiego statusu udało się na dobrą sprawę tylko dwóm komputerom osobistym — są to PC i McIntosh. Ten pierwszy dzierży palmę pierwszeństwa wśród sprzętu przeznaczonego do prac biurowych (między innymi ze względu na ilość dostępnego oprogramowania), ten drugi stanowi podstawę najpopularniejszych (i najlepszych) zestawów do Desk Top Publishing. Choć Atari ST nadaje się i do prac biurowych, i do DTP, nie stało się nigdy tak popularne w tych dziedzinach, by zaczęło się z nim poważnie liczyć.

Stąd biorą się częste rozterki posiadaczy ST, zwłaszcza tych, którzy kupili go jako komputer do domu. Jeżeli w pracy korzysta się z PC (a mało kto zatrudniony w biurze na Zachodzie tego nie robi), brak tego komputera w domu potrafi być poważnym utrudnieniem życia; z drugiej strony szkoda już wydanych pieniędzy — po co w domu dwa komputery?

W tej sytuacji jednym z najlepszych rozwiązań okazuje się być emulator PC-eta, zainstalowany w Atari. Korzysta on z tej samej pamięci, z tej samej samej stacji dyskiety, a procesor rodziny 80x86 — w zależności od rodzaju emulatora — jest albo bezpośrednio włączony do Atari, albo naśladowany przez odpowiedni program. W tym pierwszym przypadku mamy do czynienia z emulatorem sprzętowym — jest nim np. PCSpeed, w tym drugim — z emulatorem programowym.

Emulatory programowe są w pewnym sensie łatwe do zrobienia — a przynajmniej ich koncepcja jest bardzo prosta. Należy napisać program, który będzie wiernie naśladować wnętrze emulowanego komputera, następnie wprowadzić do niego oryginalne oprogramowanie i nie przejmować się niczym więcej — emulator zachowuje się jak oryginalny komputer z tego prostego powodu, że oprogramowanie służące do komunikowania się z komputerem jest oryginalne. W ten sposób zrealizowano między innymi (oprócz PCDitto) emulatory Spectrum na PC-eta i C64 na Amigę. Przyjrzyjmy się jednak PCDitto.

Pierwszą i podstawową wadą programu jest jego bardzo mała szybkość. Wprowadzić trudno się temu dziwić w przypadku emulatora programowego, ale też nie można tego zaakceptować na dłuższą metę. PCDitto na Atari ST ma szybkość hipotetycznego PC-eta z zegarem 1 MHz — czyli jest wolniejszy od najwolniejszego XT niemal 5 razy. Strasznie utrudnia to jakiegokolwiek próby skorzystania z nieco bardziej wyrafinowanych programów, gdyż po kilkudziesięciu sekundach nie wiadomo, czy jeszcze należy czekać na „przemienienie” przez emulator wstępnej części programu, czy też komputer już się zawiesił, co się niestety od czasu do czasu zdarza. Jedną z takich sytuacji zdarzyła mi się podczas testowania programu — po jego uruchomieniu zapomniałem zmienić dyskietki z PCDitto na dyskietkę systemową DOS-u — po kilku-

dokończenie na str. 30

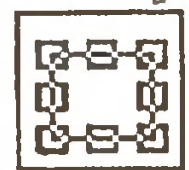
TOUCH-UP

Touch-Up jest programem graficznym przeznaczonym dla wszystkich wersji Atari ST z dowolnym monitorem. Umożliwia on wykonanie rysunku w niskiej, średniej i wysokiej rozdzielczości. Touch-Up rozpoznaje większość formatów stosowanych do zapisu rysunków w komputerach Atari ST, IBM i Apple.

MENU GŁÓWNE



— tryb rysowania — <F1>



— tryb wycinków — <F2>



— tryb tekstowy — <F3>



— tryb błyskawiczny — <F4>



— tryb plikowy — <F5>



— tryb skanowania — <F6>

MENU POMOCNICZE

Przełącznik koloru
Poziom powiększenia
pełna strona — F lub <Alt>F
skala 1:1 — 1 lub <Alt>1
skala 2:1 — 2 lub <Alt>2
skala 4:1 — 4 lub <Alt>4
Położenie/powiększenie — przełączanie <F9>

TWORZENIE OBIEKTÓW

Większość obiektów tworzy się w podobny sposób. W tym celu należy zwykle wykonać następujące czynności:
— wybrać parametry rysowania (styl linii, wzór itd.),
— wybrać odpowiednią ikonę,
— wcisnąć lewy przycisk i utworzyć obiekt,
— trzymając lewy przycisk ustalić kształt obiektu,
— określić położenie obiektu (oba przyciski zwolnione),
— ewentualnie zmienić kształt obiektu (wciskając lewy przycisk),
— utrwalić obiekt przez naciśnięcie prawego przycisku.

Aby przerwać tworzenie obiektu, należy wcisnąć lewy przycisk myszy, a następnie — nie zwalniając go — wcisnąć prawy przycisk.

TRYB RYSOWANIA



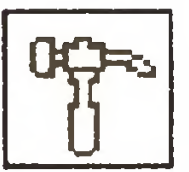
— atrament

Black — czarny
White — biały
Opposite — przeciwny



— sposób rysowania

Replace — zastępowanie
Transparent — nakładanie
XOR — mieszanie
R. Transparent — nakładanie z odwróceniem



— funkcje pomocnicze

Clear Page — kasowanie całej strony — <Alt>C
Invert Page — negatyw strony — <Alt>I
Clip To Page — powiększenie wycinka na całą stronę
Defaults — standardowe ścieżki do plików
Page/Clip — informacja o stronie i wycinku — <Alt>S
Wipe — skasowanie rysunku
Adjust — dopasowanie rysunku
Lock — blokada przesuwania tworzonego obiektu — <Alt>L



— styl linii



— cień

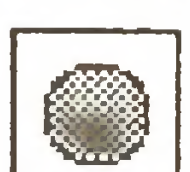


— wzory wypełniania



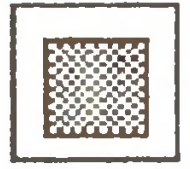
— punkt

wielkość punktu: <1>-<4>
linia pozioma: +<Shift>
linia pionowa: +<Alt>



— elipsa

okrąg: +<Alt>



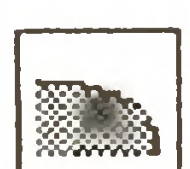
— prostokąt

kwadrat: +<Alt>
zaokrąglone naroża: +<Shift>



— krzywa łamana

odcinek pod kątem 45°, 135°, 225° lub 315°: +<Control>



— łuk/wycinek kołowy

zmiana dolnego końca o 5°: strzałka w dół/w górę
zmiana górnego końca o 5°: strzałka w lewo/w prawo
zmiana o 0,5°: +<Shift>
łuk kołowy: +<Alt>
wycinek elipsy: +<Shift>
wycinek koła: +<Shift>+<Alt>

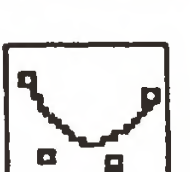


— wypełnianie

działa tylko z <Shift> lub <Alt>



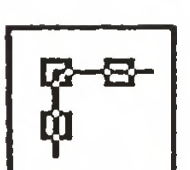
— rysowanie odręczne



— krzywe B-S i Beziera

TRYB WYCINKÓW

przesuwanie i zmiana rozmiaru wycinka — lewy przycisk myszy przesuwanie i zmiana wycinka razem z zawartością: +<Shift>
jw., lecz razem z kopią zawartości: +<Control>
przesuwanie kopii zawartości wycinka: +<Control>+<Alt>



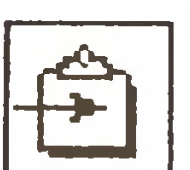
— ramka wycinka (wł/wył)



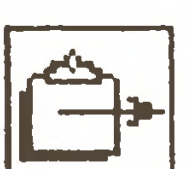
— sposób rysowania — jak w trybie rysowania



— funkcje pomocnicze — jak w trybie rysowania



— wycięcie wycinka do notesu — <F7>



— wklejenie wycinka z notesu



— wzory wypełniania



— odczyt rysunku do wycinka

.IMG — odczyt z pliku .IMG
Reload — ponowny odczyt ostatniego pliku .IMG
.GEM — odczyt z pliku .GEM
Aspect Ratio — proporcje rysunku
Preserve — zachowanie
Ignore — zignorowanie



— zapis wycinka w pliku

sub-menu — wybór formatu zapisu



— kolory wycinka

White — biały
Black — czarny
Invert — negatyw



— przerzucenie wycinka

Horizontally — poziomo
Vertically — pionowo
Both — w obu kierunkach



— odbicie wycinka

Left — w lewo
Right — w prawo
Up — w górę
Down — w dół
+<Shift> — układanie kafli



— obracanie

Clip Degrees — obrót wycinka o podany kat

Rotate — obrót wycinka
R Copy — obrót kopii wycinka
Cancel — anulowanie
zmiana kąta: strzałki o 1°
+<Shift> o 0,1°

środek obrotu = środek krzyża
<Home> — krzyż na środek wycinka
<F1> — krzyż do ostatnio użytego punktu
<F2> — krzyż na środek ekranu
Clip Left — obrót wycinka o 90° w lewo
Clip Right — obrót wycinka o 90° w prawo
Clip 180 — obrót wycinka o 180°
Page Left — obrót strony o 90° w lewo
Page Right — obrót strony o 90° w prawo



— przekrzywienie wycinka

Horizontal — poziomo
Vertical — pionowo



— funkcje dodatkowe

Bold — pogrubienie
Horizontal — poziomo
Vertical — pionowo
Both — w obu kierunkach
Cleanup — oczyszczenie
White — z białych punktów
Black — z czarnych punktów
Both — z białych i czarnych punktów
Outline — obramowanie
Mask — maskowanie
White — kolorem białym
Black — kolorem czarnym



— przegląd rysunków

View — wygląd rysunku
Info — rozmiar i rozdzielczość
Select — wybór rysunku
Set Clip — ustawienie rozmiaru wycinka
Invert — tło okna: białe/czarne
Delete — skasowanie

TRYB TEKSTOWY



— atrament — jak w trybie rysowania



— sposób rysowania — jak w trybie rysowania

TOUCH-UP



— funkcje pomocnicze —
jak w trybie rysowania



— wybór czcionki (z dys-
kietki programu)



— atrybuty tekstu

Normal — normalny
Bold — pogrubiony
Light — rozjaśniony
Italics — kursywa
Backslant — odwrotna kursywa
Outline — obramowany
Filled — wypełniony
Underline — podkreślony
Fat — poszerzony

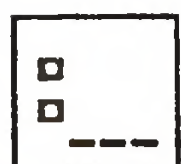


— wzory wypełniania



— wysokość tekstu

+<Shift> — zmiana o 10
+<Alt> — zmiana o 100



— treść napisu

TRYB BŁYSKAWICZNY

użycie obiektu jako pędzla: +<Shift>
anulowanie ostatniej operacji — <Undo>
anulowanie ostatniej sekwencji operacji — <Shift>+<Undo>
utrwalenie wykonanych operacji — <F10>



— atrament — jak w try-
bie rysowania



— sposób rysowania —
jak w trybie rysowania



— styl linii



— parametry aerografu

Slow — wypełnianie powolne
Fast — wypełnianie szybkie
Small — mały rozmiar
Medium — średni rozmiar
Large — duży rozmiar
Square — kwadrat
Round — okrąg
% — poziom wypełnienia
Use Fill — użycie wzoru

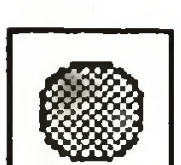


— wzory wypełniania



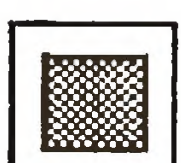
— aerograf

linia pozioma: +<Shift>
linia pionowa: +<Alt>



— elipsa/wycinek

Ellipse — elipsa
okrąg: +<Alt>
Arc — łuk
łuk kołowy: +<Alt>
Pie — wycinek elipsy
wycinek koła: +<Alt>
zmiana kształtu i kąta: +<Shift>



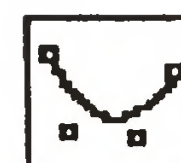
— prostokąt

kwadrat: +<Alt>
zaokrąglone naroża (wł/wył): <Alt>+<R>



— rysowanie linii

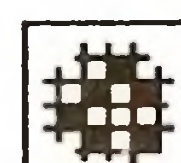
Sketch — linia odręczna
Polyline — linia łamana



— krzywe Beziera — jak
w trybie rysowania



— wypełnianie — jak w
trybie rysowania



— powiększenie



— lasso



— gumka

zmiana wymiarów: +<Shift>

TRYB PLIKOWY



— odczyt rysunku z pliku
.IMG na stronę



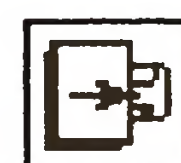
— zapis całej strony w
pliku IMG

Full Page — cała strona
Image Area — tylko obszar rysunku



— odczyt rysunku na
stronę

sub-menu — wybór formatu odczytu



— zapis całej strony w
pliku

sub-menu — wybór formatu zapisu



— odczyt rysunku do wy-
cinka — jak w trybie
wycinków



— zapis wycinka w pliku

sub-menu — wybór formatu zapisu
ustawienie rozmiaru wycinka

MacPaint — <Alt>+<A>
Degas — <Alt>+<D>
PrintMaster
1 ikona — <Alt>+<P>
3 ikony — <Alt>+<M>



— funkcje pomocnicze
— jak w trybie rysowa-
nia



— wzory wypełniania



— wydruk (nie działa,
gdy mniej niż 2 MB
RAM)

Device — urządzenie:

21 — FX80
23 — LQ800
24 — Toshiba
25 — HP Laserjet II 150 DPI
27 — HP Laserjet II 300 DPI
29 — Atari SLM804



— informacja o programie



— przegląd rysunków —
jak w trybie wycinków

TRYB SKANOWANIA



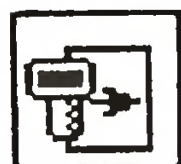
— ramka wycinka (wł/
wył)



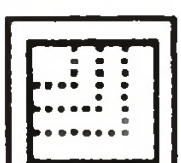
— sposób rysowania —
jak w trybie rysowania



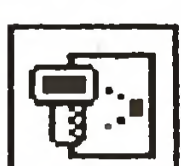
— funkcje pomocnicze —
jak w trybie rysowania



— skanowanie na całą
stronę



— parametry skanowania



— skanowanie do wycinka



— rozdzielczość 100 DPI



— rozdzielczość 200 DPI



— rozdzielczość 300 DPI



— rozdzielczość 400 DPI

MENU ROZWIJANE

DESK

Touch-Up Info — informacja o progra-
mie

FILE

Load .IMG — odczyt strony z pliku
.IMG
Save .IMG — zapis strony do pliku
.IMG
Load Other — odczyt strony z innego
pliku
Save Other — zapis strony do innego
pliku
Load... — odczyt z pliku do wycinka
Save... — zapis wycinka do pliku
.IMG Viewer — przegląd rysunków
Defaults — standardowe ścieżki do
plików
Page/Clip — informacje o stronie i wy-
cinku
To Outprint — przejście do drukowa-
nia
Quit — koniec programu

VIEW

Full — pełna strona
Normal — skala 1:1
2x — skala 2:1
4x — skala 4:1
Hide — schowanie menu
Tear-Away — menu ruchome —
<Alt>+<T>
Close Tools — menu nieruchome

EDIT

Color — wybór koloru
Mode — sposób rysowania
Pattern — wzór wypełniania
Line Style — styl linii
Shadow — cień
Lock — blokada ruchu obiektu
Clear Page — kasowanie strony
Invert Page — negatyw strony
Clip to Page — wycinek na stronę

CLIP

Set Color — kolor wycinka
Flip — przerzucenie wycinka
Mirror — odbicie wycinka
Rotate — obracanie
Slant — przekrzywienie wycinka
Process — funkcje dodatkowe
Cut To — wycięcie wycinka do notesu
Paste From — wklejenie wycinka z no-
tesu

TEXT

Normal — normalny
Bold — pogrubiony
Light — rozjaśniony
Italics — kursywa
Backslant — odwrotna kursywa
Outline — obramowany
Filled — wypełniony
Underline — podkreślony
Fat — poszerzony
Height — wysokość tekstu
Faces — wybór czcionki

Ponadto w systemach z ramdyskiem lub
twardym dyskiem działają klawisze:

<Shift>+<F1> — szybki zapis całej strony
<Shift>+<F3> — szybki odczyt całej strony

dziesięciu sekundach zreseto-
wałem komputer, nie mogąc
się doczekać z jego strony żad-
nej reakcji. Do dziś nie wiem,
czy było to zawieszenie się
komputera, czy też należało
jeszcze trochę poczekać na
komunikat „Non system disk,
replace and strike any key...”

Zeby sprawdzić, jak dokład-
nie jest emulowany procesor
PC-eta tudzież wszystkie do-
datkowe otaczające go układy,
posłużyłem się programem
checkit, stanowiącym znako-
mite narzędzie do testowania
prawdziwych PC-etów i spra-
wdzania, w jakim stanie jest
hardware — wszelkie kontrol-
ery przerwań, kanały DMA i
inne. PCditto w czasie testu
wypadł całkiem nieźle. Pod-
czas testowania konfiguracji
jako procesor został rozpozn-
any 80188 (jest to swoista
ciekawostka — nie widziałem
nigdy XT wyposażonego w tę
kość). Poprawnie przebiegało
również rozpoznawanie przy-
pisania przerwań do poszcze-
gólnych urządzeń zewnętrznych i test
pamięci (choć początkowo byłem
przekonany, że komputer znowu wisi
— tak wolno to przebiegało). Z innymi
testami bywało różnie — emulator
DMA nie dał sobie rady i został rozpo-
znany jako uszkodzony; z testu konfi-
guracji wynikało, że do komputera po-
dłączone są dwa joysticki (jako żywo
nie było ani jednego) i że komputer
dysponuje dwiema stacjami dyskietek,
a nie jedną. Były także kłopoty z łą-
czem szeregowym i równoległym —
być może dlatego, że samo Atari
przed uruchomieniem emulatora nie
wiedziało o ich istnieniu. Najwięk-
szych kłopotów miał jednak dostar-
czyć test pamięci ekranu i monitora.
Okazało się bowiem, że PCditto nie-
prawidłowo tłumaczy atrybuty znaków
i w trakcie pracy z nim nie zawsze to,
co widać na ekranie, odpowiada za-
wartości videoRAMu. Powoduje to
kłopoty zwłaszcza wtedy, gdy okazuje
się, że najistotniejsza część komuni-
katu wyświetlonego na ekranie jest
niewidoczna.

PCDitto był przez różne osoby uży-
wany w redakcji przez kilka tygodni —
po części dla zabawy, z rzadka wtedy,
gdy trzeba było wykonać jakieś działa-
nia pod kontrolą DOS-u, a nie było
wolnego IBM-a. Przez cały ten czas
spisywał się nie najgorzej — choć nie-
zwykle wolno, ale wykonywał wszyst-
ko, czego od niego zażądano. Najpo-
ważniejszy problem, jaki występował
dosyć często, związany jest ze stacją
osiemdziesięciościeżkową — MS-DOS
w XT nie zawsze potrafi sobie z taką
stacją poradzić (wie o tym każdy, kto
próbował zainstalować w XT stację
720 KB). Na szczęście istnieje bardzo
wygodny program rezydentny, o naz-
wie 800, który pomagał nam w takich
momentach. Od czasu do czasu oka-
zywało się również, że niektóre pro-
gramy z bliżej nie znanych przyczyn
nie chcą działać w PCDitto — dotyczy-
ło to na przykład ChiWritera 1.0 i Bloc-
kout. Nie wiem, dlaczego tak się dzia-
ło, zwłaszcza że inne wersje ChiWri-
tera zachowywały się poprawnie.

Podsumowując, emulator powoli re-
alizuje postawione przed nim zadania
— choć nie zawsze tak, jak można by
było sobie tego życzyć. Jego podsta-
wową wadą jest wyjątkowo mała szyb-
kość, podstawową zaletą — nie trzeba
dokonywać jakichkolwiek przeróbek
komputera. Czy jednak jest to rozwią-
zanie, które może satysfakcjonować
kogoś pracującego na co dzień przy
normalnym PC? Nie wiem.

Marcin BORKOWSKI

PCditto

dokończenie ze str. 28

stanu. Dobrze jest też otworzyć pokrywę od spodu myśły, wyjąć kulkę i skontrolować stan zabrudzenia rolek oraz zabezpieczenie śrub justujących fotokomórki.

ZŁĄCZA

Po pierwsze, trzeba przekonać się, czy wszystkie wtyczki po osadzeniu trzymają się dostatecznie mocno w gniazdach. To nie znaczy jeszcze, że złącze jest dobre. Mogło ono zostać uszkodzone przez wyciąganie i wkładanie wtyczki przy włączonym zasilaniu. Najwrażliwszym ze złączy jest port ROM. Popróbujmy poruszać delikatnie końcówkami kabli. Nie powinno to spowodować zresetowania komputera. O stanie złącza równoległego przekonamy się próbując wydrukować krótki tekst na drukarce. Uszkodzeniom portu równoległego towarzyszą często defekty układów generujących dźwięk. Najłatwiej to sprawdzić uruchamiając jakąś grę z muzyczką. Na koniec prosimy o zademonstrowanie funkcji złącza szeregowego za pomocą drukarki lub modemu

STACJA DYSKÓW

Bardzo ważne jest sprawdzenie napędu dysków. Rozregulowana stacja dysków będzie czytać tylko dyskietki, które zostały na niej zapisane. Przy wczytaniu dyskietki zapisanej na innym komputerze wystąpią błędy. Najlepiej sprawdzić to programem testującym prędkość obrotową stacji dysków. Jeśli nie mamy go pod ręką, to należy spróbować sformatować czystą dyskietkę i przeprowadzić próbę kopiowania i wczytywania plików zapisanych na innej stacji.

SPOJRZENIE DO ŚRODKA

Na koniec zaglądamy pod obudowę. Niektóre komputery mają rozszerzoną pamięć i wbudowane dodatkowe karty. Czasami zdarza się, że podczas przeróbek zdjęto blachę ekranującą. Taki komputer będzie bardzo wrażliwy na wszelkie zakłócenia: wyładowania atmosferyczne, pracującą u sąsiada wiertarkę, przeskok iskry z naelektryzowanego ubrania. W przebudowanym ST należy sprawdzić staranność wykonanych połączeń, czy podstawki układów scalonych nie są uszkodzone i powyginane, czy nie ma obluźwionych, wiszących kabli. Standardowy zasilacz może nie mieć wystarczającej mocy do zasilania wszystkich dodatkowych rozszerzeń ST. Objawem tego jest zawieszanie się komputera, żółtawa poświata monitora i brak sygnału na niektórych złączach.

Przestrzeganie powyższych uwag powinno w 99% zabezpieczyć przed kłopotliwym nabytkiem. Życzę więc pomyślnych zakupów.

Janusz JARMOCH

TT

- WIZYTÓWKA ATARI

dokończenie ze str. 32

960 oraz tryb „Duo-chrome” — 640 x 400 punktów w dwóch dowolnych kolorach. Decyzja o wyborze rodzaju monitora i najkorzystniejszego trybu pracy należy do użytkownika. Do zastosowań DTP szczególnie zalecany jest 19-calowy monitor TTM 194, który w pełni wykorzystuje zalety trybu wysokiej rozdzielczości.

Komputer ma również duże możliwości dźwiękowe. Wyposażono go w 8-bitowy stereofoniczny generator dźwięku PCM (Pulse Code Modulation). Automatycznie regulowane filtry, regulator tonów wysokich i niskich, regulacja balansu są zrealizowane sprzętowo. Dźwięk może być generowany przez wewnętrzny głośnik lub przesyłany na wyjście stereofoniczne.

ZŁĄCZA

Bogaty asortyment interfejsów pozwala przyłączyć do TT prawie każde typowe urządzenie peryferyjne. Wśród złączy Atari TT znajdujemy: interfejs równoległy Centronics, złącze szeregowo RS232, interfejs MIDI, złącze Atari ACSI-DMA, interfejs SCSI, złącze sieci lokalnej LAN (Local Area Network) oraz wewnętrzną szynę VME do przyłączenia kart rozszerzających. Tak uzbrojony komputer nie powinien sprawić żadnej nieprzyjemnej niespodzianki. Bez problemu przyłączymy do niego skaner, dysk optyczny, drukarkę laserową, modem, ploter itd.

PRZESIADKA Z ST NA TT

Twórcy TT przywiązywali ogromne znaczenie do zgodności z dotychczasowym standardem ST. Oczywiście podyktowane to było koniecznością wykorzystania bogatej biblioteki oprogramowania ST, bez której nowy komputer początkowo byłby niewiele wart dla użytkownika.

Redakcja niemieckiego miesięcznika „ST Magazin” przetestowała działanie wielu programów na TT. Z DTP nie było żadnych problemów. Calamus i Timeworks Publisher pracowały bez zarzutu. Podobnie Signum 2 i Script 2.0. Pozytywne rezultaty dał test programów graficznych. Arabesque, MegaPaint, STAD, Neochrome, Degas Elite, SciGraph, Easy Draw spełniały swoje zadanie bez zarzutu. Niektóre udało się uruchomić nawet w trybie 16-kolorowym TT. Dla wielu użytkowników naj-

istotniejsza jest możliwość korzystania z CAD. TechnoCad i DynaCad pracowały we wszystkich rozdzielczościach i w kolorze. Bardzo dobrze sprawdziła się większość programów kalkulacyjnych i baz danych.

Niestety nie wszystkie próby wypadły pozytywnie. Omikron-Basic zawieszał się prawie natychmiast. GFA-Basic 3.5 działał z pewnymi ograniczeniami. Fatalnie wyglądała sytuacja z programami Public Domain, które w większości zostały napisane w GFA- lub Omikron-Basic i oczywiście „odziedziczyły” ich wady²⁾.

Czy Atari TT ma szansę stać się popularnym komputerem domowym? W RFN wersja 4 MB z twardym dyskiem 46 MB kosztuje prawie 4.300 DM. Na taki wydatek na razie mogą pozwolić sobie tylko najbogatsi. Należy się spodziewać, że ceny będą w przyszłości spadać, ale na pewno sympatycy gier i hobbisci, parający się w wolnych chwilach przetwarzaniem tekstów, będą woleli kupować znacznie tańsze 1040 ST/STE lub którąś z wersji Mega ST. Badania prowadzone przez Commodore wykazały, że sprzedaje się prawie 10 razy więcej komputerów Amiga 500 niż znacznie droższych Amiga 2000/3000.

Inaczej sprawa wygląda w przypadku profesjonalistów, dla których komputer jest narzędziem pracy. Atari TT jest idealnym komputerem do zastosowań z dziedziny CAD i DTP. Za wyborem TT przemawia bardzo wysoka szybkość działania i bogate wyposażenie w różne interfejsy. Pozwala to użytkownikowi na jednoczesne przyłączenie wielu niezbędnych urządzeń i właściwe zorganizowanie stanowiska pracy.

Na razie istniejące oprogramowanie użytkowe nie zawsze efektywnie wykorzystuje zalety komputera, ale wkrótce sytuacja ta na pewno się zmieni. Nowe wersje programów muszą uwzględnić standardy TT i wszystkie innowacje wprowadzone do komputera, jeśli producent poważnie myśli o ich sprzedaży.

Janusz JARMOCH

1) Personal Computer World 10/89 s. 100

2) ST Magazin 11/90 s. 54

DANE TECHNICZNE ATARI TT

Architektura wewnętrzna

- * 32-bitowy procesor Motorola 68030 taktowany z częstotliwością 32 MHz

- * koprocesor arytmetyczny Motorola 68882 taktowany z częstotliwością 32 MHz

Pamięć wewnętrzna

- * 4, 6, lub 8 MB RAM
- * 512 KB ROM

Pamięć zewnętrzna

- * twardy dysk 46 MB
- * średni czas dostępu 28 ms
- * napęd dysków elastycznych 720 KB

Grafika

- * Tryby rozdzielczości kompatybilne z ST

- * 320 x 480 punktów w 256 kolorach z palety 4096 barw

- * 640 x 480 punktów w 16 kolorach z palety 4096 barw

- * 640 x 400 punktów w trybie „Duo-chrome” w dwóch kolorach z palety 4069 barw

- * 1280 x 960 punktów w trybie monochromatycznym

Interfejsy

- * interfejs równoległy Centronics

- * interfejs szeregowy RS232C

- * możliwość wbudowania 3 dodatkowych złączy szeregowych

- * interfejs MIDI

- * złącze Atari ACSI-DMA do przyłączenia twardego dysku, drukarki laserowej lub CD ROM Atari

- * interfejs SCSI do przyłączenia twardego dysku, CD ROM lub skanera standardu SCSI

- * trzy złącza szeregowo lub dwa złącza szeregowo i jedno złącze sieciowe

- * wewnętrzna szyna VMEbus (A24/D16, A16/D16) umożliwiająca przyłączenie karty graficznej, sieciowej, modemu, emulatora

Dźwięk

- * 8-bitowy, stereofoniczny generator dźwięku PCM (Puls Code Modulation)





TT

O Atarii TT mówi się już od ponad 3 lat.

Prototypowy model pokazywano po raz pierwszy na targach CeBIT'88. Komputer ten miał wkrótce stać się wizytówką firmy i umocnić jej dobrą pozycję na rynku. Jednakże wersja handlowa musiała jeszcze poczekać, zanim ujrzała światło dzienne. Problemy techniczne uniemożliwiły wejście na rynek z seryjnym modelem super ST. Klienci odwiedzający stoiska Atari na kolejnych targach musieli niestety odchodzić z kwitkiem. Niektórzy sceptycy zaczęli już powątpiewać, czy komputer w ogóle wyjdzie poza biura konstrukcyjne i laboratoria firmy.

W trakcie prac nad finalną wersją Atari TT kilkakrotnie zmieniano koncepcję wykonania pewnych elementów komputera.

Dane techniczne, publikowane jeszcze w 1989 roku, znacznie różnią się od parametrów komputera, który zaczęto sprzedawać w ubiegłym roku.

CHARAKTERYSTYCZNA SYLWETKA

Nowe Super-Atari ma bardzo

charakterystyczną sylwetkę, która odróżnia je nie tylko od wszystkich modeli ST, ale i od komputerów osobistych PC. Jednostka centralna stanowi osobny zespół. Estetyczną, ładnie wyglądającą obudowę TT wykorzystano także do Mega STE, które pojawiło się w handlu na początku tego roku.

Klawiatura różni się nieco od tradycyjnych rozwiązań wzorniczych ST. Zmieniono kształt klawiszy funkcyjnych. Rozmieszczenie pozostałych klawiszy jest zgodne ze standardem przyjętym w innych modelach ST. Do gniazda z prawej strony klawiatury można przyłączyć mysz lub joystick. Zamiast klawiatury TT można stosować także seryjną klawiaturę używaną do Mega ST.

32-BITOWY PROCESOR

Dwie litery „T” w nazwie pochodzą od angielskich liczebników *Thirty-two Thirty-two*, które po polsku oznaczają liczby 32/32 (ST wzięło się od *Sixteen Thirty-two*, tzn. 16/32)¹⁾. Skrót ten dotyczy oczywiście procesora, najważniejszego elementu komputera. W Atarii TT zastosowano 32-bitowy procesor Motorola 68030, który jest następcą bardzo udanego 16-bitowego 68000. Pierwotnie planowano taktowanie komputera z częstotliwością 16 MHz, lecz ostatecznie zdecydowano się na wersję 32 MHz. Na płycie głównej zainstalowano także koprocesor arytmetyczny Motorola 68882, również taktowany z częstotliwością 32 MHz. Wpływa to w decydujący sposób na sprawność komputera i zdolność przetwarzania danych, którą często określa się trudnym do przetłumaczenia angielskim terminem *performance*. Szybkość działania TT jest tak duża, że zbyteczny stał się blitter — układ wspomagający w ST tworzenie grafiki. Współczynnika przyrostu szybkości w porównaniu do ST nie da się jednoznacznie określić. Zależy on od rodzaju wykonywanych operacji. Zazwyczaj osiąga się przyspieszenie działania programu o 2–8 razy. W programach wykonujących dużo operacji arytmetycznych na liczbach zmiennoprzecinkowych, a więc intensywnie korzystających z koprocesora, wartość ta dochodzi nawet do stukilkudziesięciu. W rzadkich przypadkach może okazać się, że program uruchomiony na TT działa tylko nieznacznie szybciej.

26 MB RAM

Atari TT w podstawowej wersji jest wyposażone w pamięć roboczą o pojemności 4 MB. Cztery megabajty RAM gwarantują poprawną i komfortową pracę z podstawowymi pakietami oprogramowania użytkowego, ale bardziej wybredni użytkownicy mogą sobie zafundować 6 lub 8 MB. Licząc się z niewielkim zainteresowaniem klientów, zrezygnowano z zapowiadanej wcześniej wersji 2 MB.

pobierają dane z tego obszaru, to ich działanie jest wtedy o wiele efektywniejsze. Pamięć ROM ma aż 512 KB. To tyle co całe 520 ST. ROM jest zbudowany z czterech układów po 128 KB.

TOS i UNIX

System pracuje pod kontrolą zmodyfikowanej wersji TOS, którą przystosowano do 32-bitowego 68030. Wersja ta jest oznaczana symbolem TOS 030. System jest zapisany na stałe w pamięci ROM.

Atari zapowiadało także zaimplementowanie innych systemów operacyjnych. Wiele korzyści obiecuje zastosowanie systemu UNIX. Aplikacja tego systemu dla Atari TT nazywa się ATX (Atari UNIX). Łączy ona cechy dwóch wersji tego systemu: UNIX System V wersja 3.1 i Berkeley UNIX BSD 4.3. Środowisko graficzne X/Window System 11.4 gwarantuje prostą obsługę komputera pracującego pod potężnym, wielozadaniowym i wielodostępnym systemem operacyjnym. Przypomina on GEM stosowany w ST. Przewiduje się również aplikacje standardu OSF i X/Open. Czy te projekty zostaną zrealizowane w najbliższej przyszłości, na razie nie wiadomo.

TWARDY DYSK 46 MB

Obowiązkowym wyposażeniem komputera wyższej klasy są urządzenia pamięci dysponujące odpowiednio dużą pojemnością. Atari TT w wersji podstawowej jest dostarczane z twardym dyskiem 46 MB. Oczywiście możliwa jest instalacja znacznie pojemniejszego dysku.

TT ma również wbudowaną stację dysków elastycznych 3,5"/720 KB. W tym przypadku Atari dotrzymało wierności standardowi stosowanemu w innych komputerach ST i STACY, chociaż istnieją napędy 3,5" o pojemności 1,44 MB. Być może konstruktorzy zdecydują się w przyszłości na zwiększenie pojemności stacji. Kontroler dysków elastycznych zainstalowany w TT może sterować jednocześnie dwoma napędami, a więc w razie potrzeby użytkownik

- WIZYTÓWKA ATARI

W tej kategorii nadal dominować będzie Mega ST i jego zmodernizowany następca Mega STE.

Foldery reklamowe Atari zapowiadają także możliwość rozszerzenia pamięci do 26 MB. Konieczne jest wtedy zastosowanie 4-megabitowych układów RAM. W komputerach 6- i 8-megabajtowych podzielono RAM na dwa obszary. Programy odziedziczone po ST mogą pracować tylko w obszarze do 4 MB, który nosi nazwę ST-RAM. Powyżej 4 MB zaczyna się tak zwany Fast-RAM. Obszar ten jest niedostępny dla procesora 68000, na którym pracuje ST. Dostęp procesora 68030 do Fast-RAM jest znacznie szybszy. Jeśli procedury, które intensywnie korzystają z pamięci,

nik ma możliwość przyłączenia drugiej, zewnętrznej stacji dysków.

GRAFIKA I DŹWIEK

TT może współpracować z monitorem RGB lub monochromatycznym. W trosce o zgodność ze starszymi modelami Atari zachowano wszystkie tryby rozdzielczości ST. Oprócz tego komputer ma cztery zupełnie nowe tryby pracy. Paleta 4096 barw daje możliwość uzyskiwania niezwykle efektownych kolorystycznych. Nowe tryby graficzne TT to: 320 x 480 punktów i 256 barw z palety 4096 kolorów, 640 x 480 i 16 kolorów z 4096, monochromatyczny tryb wysokiej rozdzielczości 1280 x

dokończenie na str. 31