

Bajtek

NUMER SPECJALNY

TYLKO O ATARI

CENA 200 ZŁ

NR INDEKSU 353965

złącza

rodzina ATARI

OKIENKA
BULDOGER
COPY

GONIMY ZA RYNKIEM

Action! (2)



DROGI CZYTELNIKU!

„Nie ma to jak ATARI!” — takie opinie słyszymy często nie tylko od posiadaczy komputerów z charakterystycznymi trzema zbiegającymi się u góry paskami, ale i od wielu tych, którzy dopiero o swoim krzemowym cacku marzą. Klient nasz pan! Wydajemy więc drugi już numer dodatkowy „Bajtki” poświęcony w całości komputerom „Atari”.

Po ukazaniu się „Atari-1” przeczyliśmy prawdziwe obłędzenie telefoniczne, gdyż w kioskach „Ruchu” nie zagrażał on miejsca. Przy okazji raz jeszcze prosimy o wyrozumiałość tych wszystkich, którzy odeszli z kwitkiem, ale liczba egzemplarzy redakcyjnych, jaką przeznaczył do naszej dyspozycji wydawca była naprawdę symboliczna.

Z informacji, które dotarły do nas z „Ruchu”, wynika, że również dodatkowy numer „Bajtki” poświęcony komputerom „Commodore” spotkał się z przychylnym przyjęciem Czytelników.

Fakty powyższe upoważniają do stwierdzenia, iż nasza inicjatywa wydawania monotematycznych zeszytów dodatkowych sprawdziła się w praktyce. Cieszy nas to. Zgodnie z zapotrzebowaniem Czytelników będziemy więc tę formę kontynuować i rozwijać.

Po wydaniu „Atari-2” ukaże się — jeszcze przed Gwiazdką — numer dodatkowy „Bajtki” adresowany specjalnie do tych wszystkich, którzy dopiero zaczynają swoją komputerową przygodę. W wielu listach do redakcji ciągle ponawiane są prośby o powrót do podstaw programowania, posługiwania się komputerem itp. Spełniamy je wydaniem specjalnym.

W planach mamy również kolejne wydania dla miłośników „Atari” i „Commodore”, a także — uwaga! — pierwszy zeszyt poświęcony wyłącznie komputerom „Amstrad-Schneider”.

Poza nimi, oczywiście, co miesiąc czeka na Was kolejny „zwykły” numer „Bajtki” — od października o zwiększonej objętości i bogatszej zawartości merytorycznej.

Przyjemnej lektury!

Waldemar Siwiński

„BAJTEK-ATARI”
(WYDANIE SPECJALNE „BAJTKA”
— POZA PRENUMERATĄ!)

PRZYGOTOWAŁ ZESPÓŁ W SKŁADZIE: Waldemar Siwiński, Grzegorz Onichimowski, Roman Poznański, Wojciech Zientara, Wanda Roszkowska (opr. graficzne), Sławomir Gajda (red. techniczny), Leopold Dzikowski (zdjęcia).
Fotokład — Tadeusz Olczak
Montaż offset. — Grażyna Ostaszewska

ADRES: 00-687 Warszawa, ul. Wspólna 61. Tel. 21-12-05.

WYDAWCA: RSW „Prasa-Książka-Ruch” Młodzieżowa Agencja Wydawnicza, al. Stanów Zjednoczonych 53, 04-028 Warszawa. Telefony: Centrala 13-20-40 do 49. Redakcja Reklamy wewn. 403, 414.

Skład techniką CRT-200.

przygotowalnia offsetowa i druk:
PRASOWE ZAKŁADY GRAFICZNE
RSW „PRASA-KSIĄŻKA-RUCH”
w Ciechanowie, ul. Sienkiewicza 51.
Nr zlecenia 078518
nakład 100 000 + 300 egz., U-113
Cena 200 zł,—

CO W numerze



OPROGRAMOWANIE

Action! /2/ 15
W drugim odcinku naszego kursu Action! poznajemy podstawowe wrażenia i instrukcje.

Okienka 6
Okienka na ekranie to nie tylko domena ST. Możesz je mieć również w programach na swoim Atari XL/XE.

GRY

Nowe i stare 24
Opisy „Mercenary”, „Wargames Construction Set”, „Spy vs. Spy”, „Spy xs. Spy II”, „Green Bret” i „Blue Team Bridge”.

PROGRAMY UŻYTKOWE

Bulldozer Copy 8
Ten program usunie zmore z zanikania zapisu na taśmie i kłopoty z wczytywaniem dawno nagranych programów.

Optymalizacja zapisu na dyskietkach 12

Dobierając odpowiednio programy można zapisać dyskietkę tak, aby zająć wszystkie sektory. Jest to bardzo żmudne zajęcie, lecz od tego jest komputer.

Suplement do „Print Shop Converter” 8
Niewielka poprawka umożliwiająca współpracę z programem „Print Shop Companion”.

SPRZĘT

Rodzina Atari /2/ 4
W drugim odcinku opisujemy magnetofony i stacje dysków przeznaczone do współpracy z komputerami Atari.

Cudowna skrzynka 10
Przystawka XEPBO umożliwiająca uzyskanie 80-kolumnowego obrazu zarówno na monitorze monochromatycznym, jak i kolorowym.

MicroPrint 29
Interfejs Centronics dla komputerów Atari XL/XE renomowanej firmy Supra Corp. jest już dostępny w Polsce.

Złącza 10
Coś dla majsterkowiczów — schematy wprowadzeń wszystkich złączy wszystkich modeli Atari.

SZKÓŁKA

Podstawy obsługi 5
Przed rozpoczęciem programowania należy poznać działanie edytora i klawiatury naszego Atari.

INSTRUKCJE

Fun with Art 13
Popularny program graficzny do wykonywania za pomocą joysticka wielokolorowych rysunków w trybie GRAPHICS 15.

EDUKACJA

Latarnia na horyzoncie 19
Komputeryzacja szkół przebiega z oporami, a tymczasem Atari stosuje się już od półtora roku w szkoleniu żeglarskim.

DZIAŁ ST

ST szachy 32
Przegląd najpopularniejszych programów szachowych przeznaczonych dla komputerów serii ST.

Sharp i ST 32
Kalkulator kieszonkowy i 16-bitowy komputer? Oczywiście! Taka współpraca jest możliwa.

INNE

Wywiad 3
Rozmowa z Bogdanem Wiśniewskim — dyrektorem P.Z. „Karen”, które prowadzi serwis komputerów Atari w Polsce.

TACE 20
Klub użytkowników Atari w Teksasie.

Kluby Atari w USA 20

Ocena 23

Recenzje 21

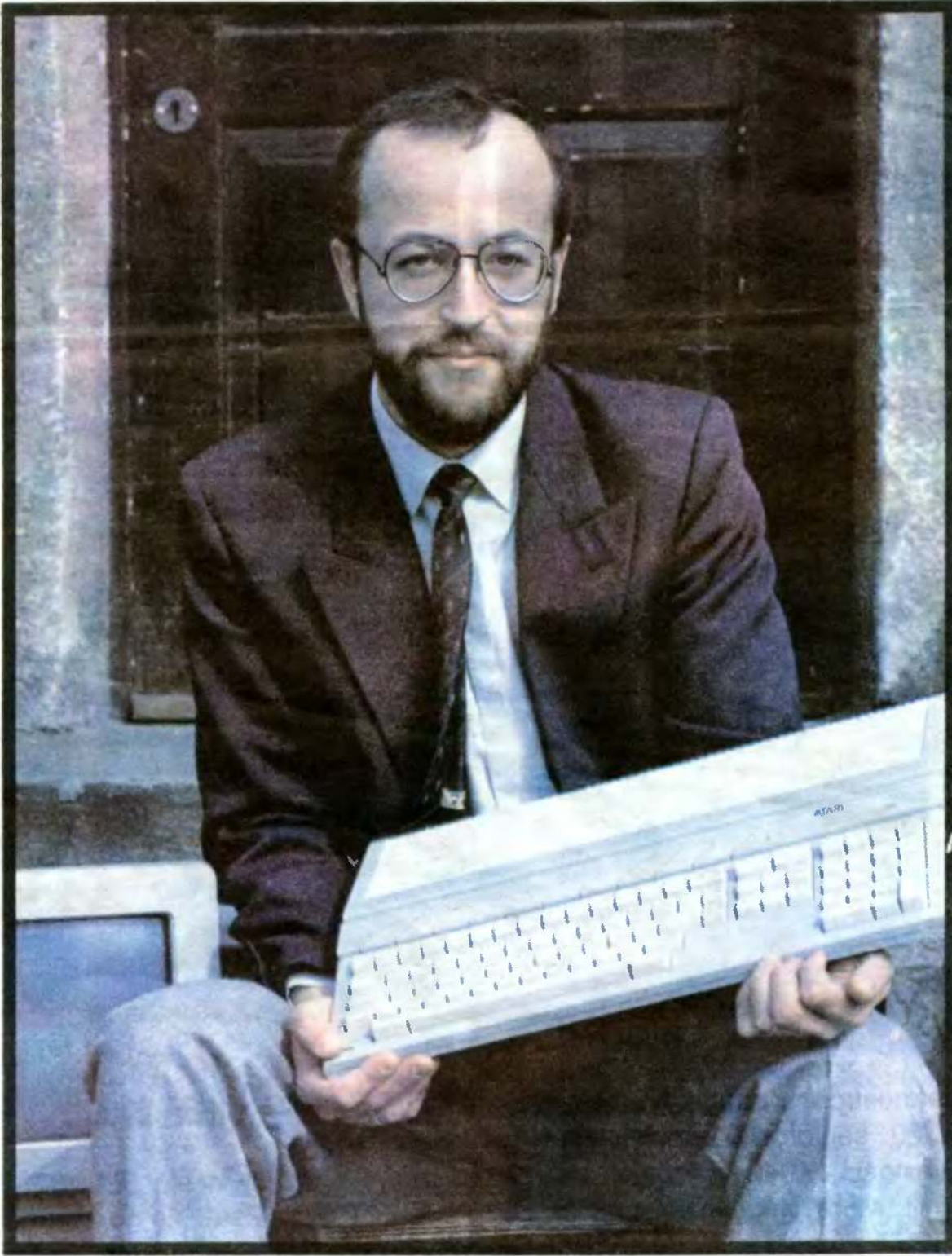
Kruczki i sztuczki 29

Więści 14

Listy 22

Ankieta 7





to ciągła pogoń za potrzebami rynku. Nie zawsze potrafimy im dotrzymać kroku.

— „Pewex” też powinien się o to starać.

— Kupowanie w „Pewexie” ma swoje zalety. Firma ta dysponuje bardzo rozbudowaną siecią handlową, wspólnie z nami udało się jej także chyba zapewnić dobre warunki serwisu. Z drugiej strony jednak rzeczą naturalną jest, że taki potentat zainteresowany jest przede wszystkim tymi produktami, których może sprzedać wiele. To zubaża ofertę.

— A mnie także nie bardzo odpowiada kupowanie komputera jak bułki — bez fachowej porady, obsługi, itp.

— No cóż, to zależy od sklepu. Podstawowym wymogiem „Pewexu” wobec kontrahenta było dostarczanie instrukcji obsługi w języku polskim. I tę klient dostaje. W sklepach największych miast, sprzedających głównie sprzęt elektroniczny, pracują sprzedawcy, którzy na ogół, potrafią sporo powiedzieć o oferowanych wyrobach. Inaczej jest w małych miejscowościach, gdzie jest tylko jeden sklep „Peweksu” sprzedający wszystko. Ten sam problem istnieje jednak i na Zachodzie. Kupując komputer np. w domu towarowym nie uzyskasz od sprzedających żadnych fachowych porad.

— Jeśli chcesz je jednak uzyskać możesz zamiast supermarketu wybrać specjalistyczny magazyn.

— Tak. Myślisz, że i mnie się nie marzy stworzenie, choćby i rzadkiej, sieci sklepów nastawionych niemal wyłącznie na sprzęt komputerowy? W takim sklepie obsługa nie nosiłaby wyłącznie paczek, a klient mógłby usiąść przy kawie ze

— W odwodzie jest jeszcze LDW. To właśnie na jego zamówienie produkowane będą na polski rynek nowe stacje dyskiety pozwalające na przeniesienie programów z IBM. LDW rozpoczęła także wysyłkową sprzedaż do Polski komputerów serii ST.

— Robiąc konkurencję „Pewexowi”?

— Nie. Gdy „Pewex” na poważnie zainteresuje się ST, LDW wycofa z oferty to, co sprzedaje „Pewex”.

— Szkoda, bo ceny tam są bardziej do zniesienia. Zmierzymy jednak nieco temat. Czy Atari dostrzega Waszą firmę i Wasz sukces?

— Atari dostrzega głównie sukces swojego dystrybutora. Kierownictwo firmy jest zadowolone, że wciąż sprzedaje komputery 8-bitowe i, o ile zainteresowanie nimi w naszym kraju nie spadnie, będzie kontynuowało ich produkcję specjalnie dla Polski. Poza tym fakt, że nasz rynek przestał być nieistotny z punktu widzenia poziomu sprzedaży dodatnio wpłynął np. na terminowość dostaw.

— A dla Was nic z tego sukcesu nie wynika? Czyżbyście nie brali np. udziału w organizowanych przez koncern wystawach, pokazach i innych imprezach „Atari”?

— Jesteśmy na nie także zapraszani. Jednak istnieje podział ról między nas a dystrybutora. Naszym głównym zadaniem, powtórzę raz jeszcze, jest serwis.

— A oprogramowanie? Przecież je wykonujecie.

— To prawda. Historia firmy zaczęła się w ogóle od produkcji oprogramowa-

GONIMY

rozmowa z Bogdanem Wiśniewskim,
dyrektorem PZ „Karen”.

ZA RYNNKIEM

— Czy czujecie się odpowiedzialni za samopoczucie polskiego użytkownika „Atari”?

— Zależy, co masz na myśli. W założeniu jesteśmy firmą, która ma prowadzić serwis „Atari” zakupionych w „Pewexie”. Uzupełnianie oferty handlowej, marketing produktów, rozwój softwaru nie należą do naszych „statutowych” zadań.

— Tym niemniej zajmujecie się tym. Organizujecie „Atarowisko” i inne spotkania użytkowników, promujecie wyroby „Atari” choćby poprzez konkursy organizowane także wspólnie z „Bajtkiem”, staracie się o zapewnienie firmie „Atari” w miarę profesjonalnego marketingu.

— No cóż, jeśli nikt inny tego nie robi. Nie jest przecież tajemnicą, że to właśnie właściciel naszej firmy przed ponad dwoma laty doszedł do wniosku, iż rynek polski może być atrakcyjny dla 8-bitowych komputerów. Miał swoją firmę w USA, dobre stosunki z „Atari”, był „Pewex” zainteresowany wzbogaceniem swej oferty, przy dobrej woli obu stron szybko doszło do pierwszych kontaktów. Zaczęło się od 300 sztuk 800XL. Potem zamówienia narastały. Kilkunastoosobowa firma musiała częściowo przejść na siebie także np. marketing „Atari”.

— Nie tylko wy go promujecie.

— Zgadza się. Robi to także np. pośrednio „Bajtek”. Istnieje również sieć klubów, gdzie działa wielu wspaniałych fanatyków. Tak naprawdę to oni, ten ruch ludzi wypromował u nas „Atari”. Co by bowiem nie powiedzieć o roli naszej „Bajtki”, czy też małych firm zajmujących się np. rozprowadzaniem programów, to użytkownicy sami wypromowali ten komputer.

— Dziwny system promocji.

— Ale przecież działał. Początki „Atari” w Polsce określam zawsze jako „klubowe”. „Karen” zaczynał swoją działalność też poniekąd klubowo — w piwnicy. Pierwsze programy dostaliśmy od działającego w Krakowie klubu użytkowników Atari...

— Czy jednak dalej też można liczyć na metodę klubową?

— Chyba już nie. 800XL, 65 XE i 130 XE zrobiły taką karierę w Polsce, bo były względnie tanie, a przy tym miały również półprofesjonalne możliwości pracy. Na ogół jednak kupowało się komputer i magnetofon by pograć w gry. Teraz ludzie chcą uzupełniać sprzęt, kupować rzeczy coraz bardziej specjalistyczne i coraz droższe. Potrzeba im rady, pomocy, potrzeba marketingu i serwisu. Naszym zadaniem jest głównie ten ostatni. Ale staramy się nie zaniedbywać również promocji, chociaż wiemy, że nasza praca

sprzedawcą i porozmawiać o tym, jakie go interesują zastosowania i jaki sprzęt oraz programy powinien wobec tego nabyć. Przy kupnie bardziej skomplikowanego sprzętu taki sposób postępowania jest wręcz niezbędny.

— A czy „Pewex” będzie sprzedawał sprzęt wyższej generacji?

— Tak, chociaż oferta firm wysyłkowych stanowi dla niego konkurencję, z którą niełatwo mu będzie wygrać. W każdym razie polityka dystrybutora „Atari” na Polskę, firmy Logical Design Works (LDW) jest taka, że wszystko, co tylko wchodzi na rynek jest natychmiast oferowane „Pewexowi”. Ten, z kolei, przy wszystkich uwarunkowaniach o jakich mówiłem, stara się, by jego oferta była kompletna. Stąd wziął się w jego sklepie model 520 ST, czy też MicroPrint.

— Czy pojawi się coś jeszcze?

— Zapewne tak. Na razie jednak myślimy o tym, by w pierwszym okresie wprowadzania wyrobu na rynek zająć się samodzielnie jego produkcją i dystrybucją „Pewexowi” oferując to, co się sprawdzi. Bardzo liczę m.in. na telekomunikację. Dostaliśmy homologację na modem XM 301. Pierwsze 500 sztuk sprzedamy sami. Jeśli przyjmie się na rynku może zainteresuje się nim „Pewex”.

— „Karen” jednak sam nie jest w stanie wyprodukować wielu rzeczy.

nia. Tutaj, jednak, też nie mamy mocarstwowych ambicji. W „Karenie” pracuje grupa programistów — sam z niej się wywodzę — która na zlecenie gotowa jest opracować dla „Atari” i LDW niemal dowolny program. I to się dzieje. Nie mieliśmy jednak dotychczas ambicji promowania polskiego oprogramowania, tego np. które powstało w klubach, na rynki zachodnie.

— Jesteście zatem głównie serwisantem, a promocją, produkcją, oprogramowaniem zajmujecie się jakby przy okazji. Czy takie postawienie sprawy nie świadczy o braku ambicji firmy?

— Jest nas kilkudziesięciu, a mamy pod opieką 200 tys. urządzeń — komputerów, stacji dysków, drukarek itp. Prowadzimy odpłatny serwis wszystkich „Atari”, jakie są w Polsce — XL, XE, ST i to niezależnie od tego, gdzie zostały one kupione. Pracy zatem starczy. Tym bardziej, że to, co nazywam serwisem obejmuje także szeroko pojęte poradnictwo. Użytkownicy „Atari” zwracają się do nas z każdą sprawą, traktują nas jak polską część koncernu. Nic na to nie poradzi mi.

— Zatem nie możesz się wyprzeć miana „człowieka Atari”?

— Nie mam takiej szansy.

rozmawiał
Grzegorz Onichimowski

W drugiej części opisu 8-bitowej rodziny Atari prezentujemy pamięci masowe przeznaczone do współpracy z komputerami serii XL/XE.

MAGNETOFONY

Wszystkie firmowe magnetofony Atari mają takie same parametry. Różnice występują jedynie w konstrukcji mechanicznej, wystroju zewnętrznym oraz w szczegółach wykonania części elektronicznej.

A oto ich dane techniczne: 2 kanały (zapis stereofoniczny), kanał prawy służy do zapisu sygnału komputerowego, a lewy do zapisu dźwięku towarzyszącego. Należy tu dodać, że kanał lewy jest wykorzystywany jedynie w programach nagrywanych firmowo (dźwięk nie daje się skopiować poprzez komputer). System ten został zastosowany na wydanej przez „Bajtka” kasecie „Polskie Logo”, co wywołało zdziwienie dużej części użytkowników, mających dotychczas do czynienia jedynie z pirackimi kopiami programów. Zapis odbywa się z szybkością 600 bodów (bitów na sekundę) na czterech ścieżkach (po dwie dla każdej strony kasety). Poza tym magnetofony posiadają sterowanie silnika przez komputer i automatyczną regulację poziomu zapisu.

ATARI 410

Pierwszy model przeznaczony do komputerów 400/800 i zgodny z nimi wzorniczo. Obecnie spotykany w Polsce w pojedynczych egzemplarzach.



Atari 410

ATARI 1010

Model przeznaczony do współpracy z komputerami serii XL. Wy-

gląd zewnętrzny dostosowany do wyglądu komputerów XL. Sprowadzany do Polski przez Pewex wraz z pierwszymi partiami Atari 800XL. Poważną wadą tego magnetofonu są nietrwałe klawisze, natomiast mechanizm napędowy jest dobrej jakości.

ATARI XC 11

Przeznaczony do współpracy z XE i dostosowany do nich wyglądem. Najlepszy z magnetofonów firmowych Atari. Sprowadzony przez Pewex w niewielkiej ilości.

ATARI XC 12

Najpopularniejszy w Polsce i jednocześnie najgorszy z magnetofonów Atari. Posiada kiepskiej jakości układ mechaniczny, który wymaga dosyć częstej regulacji. Kolorem dopasowany do serii XE, lecz odbiegający od niej wzorniczo.

STACJE DYSKÓW

ATARI 810

Pierwszy model stacji dysków dla 8-bitowych komputerów Atari. Wzorniczo dostosowana do serii 400/800. Umożliwia jednostronny zapis dyskietek z pojedynczą gęstością (720 sektorów po 128 bajtów), co pozwala na zapisanie 90 KB (część tej pojemności zabiera DOS). Dostarczana wraz z systemem DOS 2.0 (pierwsze egzemplarze z DOS 1.0). Szybkość transmisji 19200 bodów. Pracą stacji steruje procesor 6507.

ATARI 815

Również przeznaczona dla serii 400/800. Zapis jednostronny z podwójną gęstością (720 sektorów po 256 bajtów), a więc razem 180 KB. Najczęściej używana do pracy pod kontrolą DOS XL. Szybkość transmisji 19200 bodów. Wbudowany procesor 6507.

ATARI 1050

Podstawowy model stacji dysków dla komputerów Atari. Pozwala na



Atari 1010

LDW SUPER 2000

Sprowadzana przez Pewex od grudnia 1987. Zależnie od zastosowanego DOS umożliwia jednostronny zapis w pojedynczej, rozszerzonej lub podwójnej gęstości (90, 130 lub 180 KB). W Polsce sprzedawana wraz z systemem DOS XL 2.35L (pojedyncza lub podwójna gęstość zapisu). Szybkość transmisji 19200 bodów z możliwością dwukrotnego jej przyspieszenia przy użyciu programu SYNCHROMESH. Wbudowany procesor Z80. W Stanach Zjednoczonych znana pod nazwą Indus GT. Redakcyjny test tej stacji zamieszczony był w pierwszym „Bajtku-Atari”.

ASTRA 1001

Stacja dysków produkcji Astra Systems. Umożliwia dwustronny zapis z pojedynczą lub podwójną

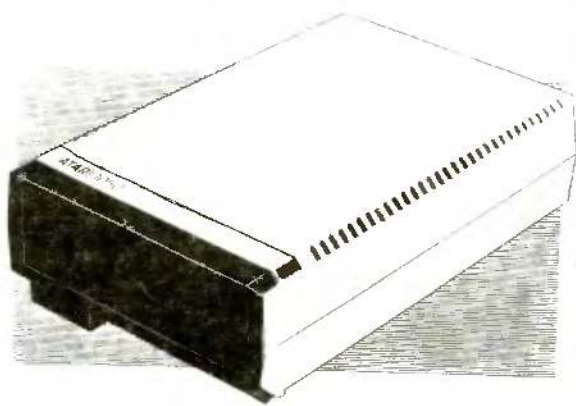
Astra 2001

ATARI XF551

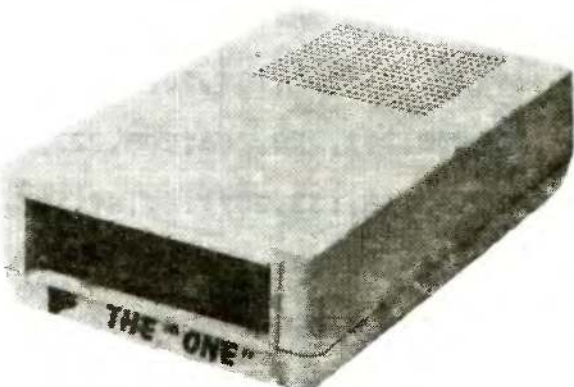
Najnowszy model stacji dysków dla komputerów serii XE. Umożliwia jednostronny zapis z pojedynczą lub podwójną gęstością. Innych danych na razie brak (zdjęcia także).



PODSTAWY OBSŁUGI



Atari 1050



Astra The One



Supra 20 MB

gęstością — maksymalnie 360 KB na jednej dyskietce. System operacyjny Smart DOS.

ASTRA 2001

Podwójna stacja Astra 1001 (dwie stacje w jednej obudowie). Całkowita pojemność stacji wynosi 720 KB (dwie dwustronne dyskietki z podwójną gęstością — 4 x 180KB). System operacyjny Smart DOS.

ASTRA BIG D

Ulepszona wersja Astra 2001. Podwójna, dwustronna stacja o pojedynczej lub podwójnej gęstości zapisu — całkowita pojemność 720 KB. System operacyjny Smart DOS.

ASTRA THE ONE

Dwustronna stacja o pojedynczej lub podwójnej gęstości zapisu — maksymalna pojemność — 360 KB. System operacyjny Smart DOS z możliwością zainstalowania RAM dysku 78 KB w komputerze 130XE. Wbudowany interfejs równoległy Centronics (do drukarki).

SUPRA 20 MB

Twardy dysk o pojemności 20 MB. Dołączany do szyny równoległej. Prędkość transmisji danych wynosi około 1 megabita na sekundę.

Wojciech Zientara

Zanim zajmiemy się praktycznym wykorzystaniem komputera, musimy się z nim trochę zapoznać.

EDYTOR

Po uruchomieniu komputera bez żadnego programu na ekranie pojawia się napis „READY”. Jest to zgłoszenie edytora i oznacza jego gotowość do przyjmowania poleceń. Edytor służy do wprowadzania przez użytkownika poleceń i programów. Komputery Atari posiadają edytor ekranowy, to znaczy, że przyjmuje on informacje z dowolnego miejsca ekranu w odróżnieniu od edytora liniowego (np. w ZX-Spectrum), który przyjmuje tylko polecenia wpisane w dolnej linii ekranu.

Pod napisem „READY” widoczny jest jasny kwadracik zwany kursorem. Wskazuje on miejsce ekranu, na którym zostanie umieszczona wprowadzona przez nas informacja (znak).

Należy jeszcze wspomnieć o organizacji ekranu, która choć niewidoczna, to jednak jest istotna dla pracy edytora. Ekran składa się z 24 wierszy (linii) fizycznych, czyli można na nim umieścić 24 wiersze znaków. Dla potrzeb edytora jest on jeszcze podzielony na wiersze (linie) logiczne. Początkowo jest ich także 24. Jeżeli wprowadzana informacja nie mieści się w jednym wierszu, to następuje przejście do następnego i jest on dołączany do poprzedniego tworząc jeden wiersz logiczny. Wiersz logiczny może składać się z maksymalnie trzech wierszy fizycznych. Zbliżanie się do końca wiersza logicznego komputer sygnalizuje ostrzegawczym brzęczykiem.

KLAWIATURA

Edytor jest obsługiwany przez klawiaturę, jak normalna maszyna do pisania. Oprócz tego niektóre klawisze i ich kombinacje mają znaczenie specjalne. Po uruchomieniu wszystkie klawisze literowe powodują pisanie dużych liter. Aby spowodować pisanie małymi literami należy nacisnąć klawisz „CAPS”. Teraz można pisać małymi literami, a duże uzyskuje się przez naciśnięcie klawisza literowego przy wciśnięciu klawiszu „SHIFT”. Ponowne naciśnięcie „CAPS” przywraca stan poprzedni. Na niektórych klawiszach namalowane są dwa symbole. W takim przypadku górny symbol w obu trybach uzyskuje się przez naciśnięcie „SHIFT” (jak w maszynie do pisania).

Komputer ma jeszcze trzeci tryb pracy klawiatury — pseudograficzny. Znaki graficzne uzyskuje się przez naciśnięcie klawisza literowego przy wciśnięciu klawisza „CONTROL”. Znaki te są w modelach XE namalowane z przodu klawiszy. Jeżeli wprowadzamy większą liczbę znaków graficznych, to możemy utrwać taki tryb pracy przez jednoczesne naciśnięcie klawiszy „CONTROL” i „CAPS”. Powrót do normalnych liter następuje przez „CAPS” lub „SHIFT” i „CAPS”.

Do zmiany sposobu interpretacji wprowadzanych znaków służy jeszcze jeden klawisz. Zwany on jest „INVERSE” i powoduje przełączanie na negatywy znaków i odwrotnie (jak „CAPS”). Klawisz ten jest oznaczony prostokątem przekreślonym i w połowie zamalowanym.

Poza tym pewne klawisze ułatwiają wprowadzanie tekstu. Klawisze ze strzałkami naciśnięte razem z klawiszem „CONTROL” przesuwają po ekranie kursor w odpowiednim kierunku. Po osiągnięciu dowolnej krawędzi ekranu kursor przenoszony jest na krawędź przeciwną. Drobna różnica występuje z lewej strony ekranu. Ponieważ niektóre telewizory dają dość duży obraz, to zdarza się, że lewa krawędź ekranu jest niewidoczna. Aby temu zapobiec lewy margines ekranu jest ustawiony na 2, więc

krawędź ekranu znajduje się w drugiej kolumnie.

Ruch kursora jest wywołany także klawiszem „TAB”. Po jego naciśnięciu kursor wykonuje skok do następnej pozycji tabulacji. Działa to dokładnie tak, jak w maszynach do pisania. Pozycję tabulacji można skasować przez ustawienie na niej kursora i naciśnięcie klawiszy „CONTROL” i „TAB”. Przez naciśnięcie „SHIFT” i „TAB” ustawiamy nową pozycję tabulacji w tym miejscu, w którym aktualnie znajduje się kursor.

Cały szereg klawiszy służy do korekty wprowadzanych informacji. Klawisz „DELETE BACK SPACE” (w skrócie „DELETE”) przesuwają kursor o jedno miejsce w lewo i usuwa znajdujący się tam znak. Gdy zostanie naciśnięty razem z „CONTROL”, to usuwa znak znajdujący się pod kursorem i przesuwają pozostałą zawartość linii logicznej znajdującą się w prawo od kursora o jedno miejsce w lewo. Jednocześnie naciśnięcie „SHIFT” i „DELETE” powoduje usunięcie całej linii logicznej i przeniesienie kursora na początek następnej.

Klawisz „INSERT” naciśnięty wraz z „CONTROL” przesuwają całą zawartość linii logicznej znajdującą się w prawo od kursora o jedno miejsce w prawo, robiąc miejsce na jeden znak (pod kursorem). Naciśnięcie „INSERT” razem z „SHIFT” powoduje wstawienie jednego wiersza fizycznego i przeniesienie kursora na jego początek.

Klawisz „CLEAR” naciśnięty razem z „SHIFT” lub „CONTROL” powoduje wymazanie całej zawartości ekranu. Jeżeli znajdującą się na ekranie informacja nie została przedtem wprowadzona do pamięci komputera, to zostaje bezpowrotnie utracona.

Poza klawiszami, na których ich funkcje są opisane, także trzy pierwsze klawisze cyfrowe mają specjalne znaczenie przy naciśnięciu razem z „CONTROL”. Klawisz „1” zatrzymuje i wznowia przesuw zawartości ekranu przy wyświetlaniu dużej ilości informacji (np. programu). Klawisz „2” wywołuje dźwięk brzęczyka. Klawisz „3” służy jako znacznik końca zbioru i jest przeznaczony dla zaawansowanych użytkowników.

Pozostały jeszcze trzy ważne klawisze: „ESC”, „RETURN” i „BREAK”. Naciśnięcie klawisza „ESC” powoduje potraktowanie następnego klawisza jako znaku do wyświetlenia. Na przykład naciśnięcie kolejno „ESC” oraz „CONTROL” i „→” nie spowoduje przesunięcia kursora, lecz wyświetlenie strzałki. Klawisz „ESC” nie działa tylko na klawisze „BREAK”, „RETURN”, „CAPS” i „INVERSE”. Klawisz „BREAK” przerywa działanie programu, oczywiście jeśli nie jest on zabezpieczony przed takim przerwaniem.

Najważniejszym chyba klawiszem jest „RETURN”. Służy on do zatwierdzenia wprowadzonej informacji. Przed jego naciśnięciem można dokonywać dowolnych zmian — informacja znajduje się wyłącznie na ekranie. „RETURN” działa jak indiańskie „Howgh!”. Gdy zostanie naciśnięty, to informacja z linii logicznej, w której znajduje się kursor, jest przekazy-

wana komputerowi do zapamiętania lub wykonania. Po skasowaniu zawartości ekranu przez „CONTROL” — „CLEAR”, w komputerze pozostają tylko te informacje, których zapisanie do pamięci zostało potwierdzone przez „RETURN”.

KONSOLA

Poza klawiaturą znajduje się jeszcze pięć klawiszy zwanych klawiszami konsoli lub klawiszami funkcyjnymi. Umieszczone są one w modelach XL (oprócz 1200XL) z prawej strony klawiatury, a w XE i 1200XL nad klawiaturą. Wykorzystanie czterech z nich — „OPTION”, „SELECT”, „START” i „HELP” — jest zależne wyłącznie od programu i normalnie nie mają one żadnego działania (test komputera jest również programem, tyle że zainstalowanym na stałe).

Klawisz „RESET” jest przyciskiem zerującym układy komputera i przywracającym stan początkowy — jak po włączeniu. Program znajdujący się w pamięci nie jest kasowany, o ile nie dokonano zmian zawartości kilku komórek pamięci sprawdzanych przez komputer po naciśnięciu „RESET”.

INFORMACJA

Wspomniałem wcześniej o przekazywaniu komputerowi informacji do wykonania lub zapamiętania. Czym to się różni? Ze względu na sposób działania interpretera Basicu można wyróżnić dwa tryby wprowadzania informacji (poleceń): tryb bezpośredni i tryb programowy.

Każde polecenie wpisane i potwierdzone przez „RETURN” jest sprawdzane pod względem składni i jeśli nie zawiera błędów, jest wykonywane. Jest to tryb bezpośredni. Aby spowodować zapamiętanie polecenia bez wykonywania należy poprzedzić go „numerem linii”, to znaczy dowolną liczbą całkowitą z zakresu od 0 do 32767. Wpisanie samej liczby usuwa z pamięci linię o tym numerze. Nazywamy to trybem programowym. Wykonanie tak zapisanych poleceń następuje dopiero po wprowadzeniu w trybie bezpośrednim polecenia „RUN”. Właśnie na tym polega pisanie programów w Basicu. Nie jest to jeszcze programowanie, lecz tylko kodowanie, czyli wprowadzenie programu w formie zrozumiałej dla komputera.

Napisałem „... jeśli nie zawiera błędów...”. A jeśli zawiera? Wtedy komputer wyświetla wprowadzone polecenie poprzedzając je słowem „ERROR” — błąd (w trybie programowym po numerze linii — taka linia też jest zapamiętywana). Miejsce, w którym komputer wykrył błąd, jest zaznaczone wyświetlaniem w negatywie. Nie zawsze jest to miejsce wystąpienia błędu. Komputer analizuje wprowadzoną informację znak po znaku i próbuje ją do czegoś dopasować. Błąd wskazywany jest dopiero wtedy, gdy mu się to nie uda. Na przykład, jeśli zamiast „RUN” wpiszemy „RYN”, to znak błędu wystąpi na końcu, a nie w literze „Y”. „RYN” potraktowane zostanie jako nazwa, z którą nie wiadomo co zrobić, a nie jako błędnie wpisane „RUN”.

Wojciech Zientara

ESC	1	2	3	4	5	6	7	8	9	0	<	>	DEL	BRK
TAB	Q	W	E	R	T	Y	U	I	O	P	-	=	RETURN	
CONTROL	A	S	D	F	G	H	J	K	L	:	+	*	CAPS	
SHIFT	Z	X	C	V	B	N	M	,	.	/	SHIFT			

OKIENKA

Na pewno nie raz z zazdrością patrzyłeś, drogi Czytelniku, na nowe komputery firm Apple czy Atari serii ST, pracujące pod systemem GEM lub podobnymi. Szalenie atrakcyjna forma przedstawiania informacji za pomocą okien skłoniła mnie do podjęcia próby przeniesienia tej metody na wystużony Atari 800XL.

Na szczęście projektanci stosunkowo niezłego niemalże wolnego od błędów systemu operacyjnego tego komputera, pozostawili użytkownikowi szerokie pola do manewru. Doskonałym tego przykładem jest obsługa urządzeń peryferyjnych (którymi w Atari są jak wiadomo: drukarka, magnetofon, edytor, ekran, klawiatura oraz — po załadowaniu DOS — stacja dysków elastycznych). Każde urządzenie ma swój wpis w tablicy HATABS (adresy 794÷832, a więc obszar RAM), przy czym nie jest ona wypełniona do końca (jest w niej miejsce na trzynaście wpisów). Pojedynczy wpis składa się z trzech bajtów: jeden bajt nazwy urządzenia w kodzie ASCII oraz dwa bajty wskazujące początek tablicy wektorów procedury obsługi. W tablicy tej są zapisane kolejno wektory skoków dla rozkazów OPEN, CLOSE, GET, PUT, STATUS, SPECIAL oraz sekwencja skoku do procedury inicjacji (JUMPTO).

Uwaga jednak: sześć pierwszych wpisów ma wartość pomniejszoną o jeden w stosunku do

faktycznej. Tak więc np. rozkaz BASIC-a OPEN # 1,4,0,"C:" powoduje najpierw znalezienie odpowiedniego wpisu w HATABS, następnie dwa bajty znajdujące się za wartością 67 ("C" w kodzie ASCII) są traktowane jako adres początku tablicy wektorów procedury obsługi magnetofonu. Pierwsze dwa bajty tej tablicy zwiększone o jeden stanowią wektor, który wskazuje miejsce w pamięci, gdzie rozpoczyna się procedura rozkazu OPEN dla magnetofonu.

Dzięki takiemu rozwiązaniu obsługi urządzeń peryferyjnych możliwa jest łatwa ingerencja w istniejące procedury a także tworzenie własnych urządzeń. Takim właśnie urządzeniem są wymienione w tytule okienka. Przyporządkowano im nazwę "O:". Instrukcja OPEN # nr kanału, WX, WY, "O:" powoduje pojawienie się na ekranie okienka, którego lewy górny róg ma współrzędne WX i WY. Ponieważ konieczne jest podanie wymiarów okna, a za pomocą instrukcji OPEN można przekazać tylko dwa parametry, trzeba posłużyć się rozkazem POKE. Do komórki 205 należy załadować szerokość, a do 206 wysokość okienka, przy czym należy pamiętać, aby zrobić to przed otwarciem kanału. Instrukcja OPEN sprawdza, czy okienko jest otwierane w trybie GRAPHICS 0 (w przeciwnym razie sygnalizowany jest błąd nr 255) oraz czy nie wychodzi poza ekran lub nie jest zbyt małe (błąd 200).

Wyświetlanie tekstu w okienku można uzyskać za pomocą instrukcji PUT # nr kanału, A, gdzie A jest kodem ASCII drukowanego znaku lub PRINT # nr kanału, "zmienna łańcuchowa". Rozkazy PUT # nr kanału, 125 lub PRINT # nr kanału, CHR\$(125) powodują wyczyszczenie tylko fragmentu ekranu wewnątrz okienka.

Zarówno po tych rozkazach, jak i bezpośrednio po otwarciu okna kursor dla danego okna jest ustawiany w jego lewym górnym rogu. Na jego położenie nie można wpłynąć instrukcją POSITION, ale zmieniają ją rozkazy przesuwające kursor (kody 28, 29, 30, 31). W przypadku, gdy drukowany tekst przewyższa pojemność okna część „wystająca” pojawia się znowu poczynając od jego lewego górnego rogu. Należy jeszcze pamiętać, że w przypadku otwarcia więcej niż jednego kanału dla obsługi okien, instrukcje PRINT i PUT będą działały tylko na ostatnim okienku. (tzn. ostatnio otwartym) niezależnie od tego, na który kanał będziemy usiłovali je wysyłać.

Program nie zmieścił się niestety na szóstej stronie pamięci, umieściłem go więc na stronie 158 (po przesunięciu RAMTOP). Gdyby użytkownik zechciał wybrać inne miejsce, należy zmienić wszystkie liczby 158 z linii 100 i 101 oraz pierwszą z linii 129, a także wpisać inną wartość do komórki 814. Aby sprawdzić działanie programu można wykorzystać drugi program, który demonstruje okna (pierwszy program musi być przedtem uruchomiony).

Andrzej Biazik

LISTING 1

```

WO 1 REM *****
PC 2 REM * Okna dla Atari 800XL/130XE *
HT 3 REM * (c) 1987 A. Biazik *
WR 4 REM *****
NK 5 REM
OV 10 AD=40448:RESTORE 100:POKE 106,158:G
    RAPHICS 0
BK 20 FOR I=0 TO 37:SUMA=0:FOR J=0 TO 9
QN 30 READ A:POKE AD+10*I+J,A:SUMA=SUMA+A
    :NEXT J:READ S
VZ 40 IF S<>SUMA THEN ? " BŁĄD DANYCH W L
    INII ";100+I:LIST 100+I:END
YB 50 NEXT I:POKE 812,79:POKE 813,0:POKE
    814,158
UE 60 ? CHR$(125):POKE 205,14:POKE 206,4:
    OPEN #1,13,9,"O":? #1:CHR$(29);" GOT
    OWE"
NG 90 POSITION 2,20:END
BF 100 DATA 14,158,248,158,248,158,251,15
    8,248,158,1799
CE 101 DATA 248,158,76,249,158,189,74,3,1
    33,203,1491
JI 102 DATA 189,75,3,133,204,165,87,240,3
    ,160,1259
    
```

```

LU 103 DATA 255,96,56,165,205,201,3,176,3
    ,160,1320
IQ 104 DATA 200,96,165,206,201,3,176,3,16
    0,200,1410
ST 105 DATA 96,24,165,203,101,205,133,205
    ,56,201,1389
QL 106 DATA 40,144,3,160,200,96,165,204,1
    01,206,1319
VP 107 DATA 133,206,56,201,24,144,3,160,2
    00,96,1223
EK 108 DATA 165,88,133,207,165,89,133,208
    ,162,0,1350
WY 109 DATA 232,24,169,40,101,207,133,207
    ,169,0,1282
MW 110 DATA 101,208,133,208,228,204,208,2
    38,169,0,1697
JW 111 DATA 164,203,145,207,200,196,205,2
    08,249,24,1801
FR 112 DATA 169,40,101,207,133,207,169,0,
    101,208,1335
DO 113 DATA 133,208,232,228,206,208,227,1
    98,6,198,1844
OL 114 DATA 205,165,88,133,207,165,89,133
    ,208,162,1555
EW 115 DATA 0,232,24,169,40,101,207,133,2
    07,169,1282
WY 116 DATA 0,101,208,133,208,228,204,208
    ,238,169,1697
PH 117 DATA 81,164,203,145,207,200,169,82
    ,145,207,1603
TC 118 DATA 200,196,205,208,249,169,69,14
    5,207,24,1672
GM 119 DATA 169,40,101,207,133,207,169,0,
    101,208,1335
GB 120 DATA 133,208,169,124,164,203,145,2
    07,164,205,1722
HQ 121 DATA 145,207,232,228,206,208,228,1
    64,203,169,1990
TY 122 DATA 90,145,207,200,169,82,145,207
    ,200,196,1641
XP 123 DATA 205,208,249,169,67,145,207,16
    5,203,133,1751
AR 124 DATA 4,165,204,133,5,230,4,230,5,1
    60,1140
BD 125 DATA 1,96,170,165,203,133,82,165,2
    05,133,1353
CL 126 DATA 83,165,84,72,165,85,72,165,4,
    133,1028
GY 127 DATA 85,165,5,133,84,138,201,125,2
    08,11,1155
AX 128 DATA 104,133,85,104,133,84,230,205
    ,76,80,1234
FC 129 DATA 158,32,176,242,165,85,133,4,1
    65,84,1244
MK 130 DATA 133,5,165,4,197,203,208,8,165
    ,205,1293
SV 131 DATA 133,4,198,4,198,5,197,205,208
    ,8,1160
NH 132 DATA 165,203,133,4,230,4,230,5,165
    ,5,1144
OD 133 DATA 197,204,208,6,165,206,133,5,1
    98,5,1327
CG 134 DATA 197,206,208,6,165,204,133,5,2
    30,5,1359
SR 135 DATA 104,133,85,104,133,84,169,2,1
    33,82,1029
UM 136 DATA 169,39,133,83,169,30,32,176,2
    42,169,1242
ZD 137 DATA 31,32,176,242,160,1,96,0,0,0,
    738
    
```

LISTING 2

```

XL 10 ? CHR$(125)
WC 20 OPEN #1,4,0,"K":DIM A$(120):POKE 75
    2,1
GZ 30 FOR I=1 TO 7:READ A1,A2,A3,A4:POKE
    205,A3:POKE 206,A4:CLOSE #2:OPEN #2,A1
    ,A2,"O"
PK 40 READ A$:? #2:CHR$(29);" ";A$:GET #1
    ,A
IT 50 NEXT I
KF 60 FOR I=1 TO 2:READ A$:? #2;" ";A$:GE
    T #1,A:NEXT I:END
FT 100 DATA 3,4,24,7,Nie tylko dla dużych
    komputerow mozna ko- rzystac z syst
    emu okien.
JS 110 DATA 5,8,33,6,Rowniez na twoim osm
    iobitowym ATARI mozna uzyskac podobne
    e- fekty.
QI 120 DATA 2,15,30,6,Oczywiscie nie sa
    one tej jakosci ale coz - nie moz- n
    a miec wszystkiego.
WE 130 DATA 18,3,21,8,Nie narzekaj jed-
    nak - ten program zaspokoi twoje
    szesnastobitowe ambicje.
WF 140 DATA 10,13,21,7,Spojrz sam - czyz
    to co masz na ek- ranie nie jest
    piekne ?
YM 150 DATA 8,9,27,7,Nawet jezeli uwazasz
    ze nie jest - to sadze ze bedziesz
    z tego progra- mu korzystal.
QW 160 DATA 4,1,23,18,Obsluga okien jest
    stosunkowo prosta - korzysta ona z
    wbu- dowanych kanalow IOCB.
NI 170 DATA Wspolrzedne lewego gornego
    rogu okna podaje sie jako pa- rame
    try dla instru- kcji OPEN.
GY 180 DATA Wymiary okna nalezy wpisac w
    czesniej do komorek 205 (wx) o- raz
    206 (wy). Powodzenia.
    
```



```

0100 RX = 203
0110 RY = 204
0120 DX = 205
0130 DY = 206
0140 W1 = 207
0150 W2 = 208
0160 KX = 4
0170 KY = 5
0180 PUTCH = 6212B
0190 ** = 4044B
0200 .WORD OPE-1,SUK-1,SUK-1
0210 .WORD PUT-1,SUK-1,SUK-1
0220 JMP SUK
0230 OPE LDA B42,X ;Przepisywanie
0240 STA RX ;zawartosci AUX1
0250 LDA B43,X ;i AUX2
0260 STA RY ;do RX i RY
0270 LDA B7 ;sprawdz, czy
0280 BEQ DL1 ;GRAPHICS 0
0290 LDY #255 ;
0300 RTS ;Blad - 255
0310 DL1 SEC
0320 LDA DX
0330 CMP #3
0340 BCS DL2 ;Sprawdzanie
0350 LDY #200 ;czy okienko
0360 RTS ;nie jest
0370 DL2 LDA DY ;zbyt male
0380 CMP #3
0390 BCS DL3
0400 LDY #200
0410 RTS
0420 DL3 CLC ;lub czy
0430 LDA RX ;nie przekracza
0440 ADC DX ;zakresu
0450 STA DX ;39 w poziomie
0460 SEC ;23 w pionie
0470 CMP #40
0480 BCC DL4 ;w przeciwnym
0490 LDY #200 ;razie -
0500 RTS ;blad - 200
0510 DL4 LDA RY
0520 ADC DY
0530 STA DY
0540 SEC
0550 CMP #24
0560 BCC DTW
0570 LDY #200
0580 RTS
0590 DTW LDA BB ;poczatek pod-
0600 STA W1 ;programu rysow-
0610 LDA B9 ;wania okna
0620 STA W2
0630 LDX #0
0640 SK1 INX
0650 CLC
0660 LDA #40
0670 ADC W1
0680 STA W1
0690 LDA #0
0700 ADC W2
0710 STA W2
0720 CPX RY
0730 BNE SK1
0740 SK2 LDA #0 ;kasowanie
0750 LDY RX ;fragmentu
0760 SK3 STA (W1),Y ;pamieci ekranu,
0770 INY ;gdzie ma sie
0780 CPY DX ;znalezc okno
0790 BNE SK3
0800 CLC
0810 LDA #40
0820 ADC W1
0830 STA W1
0840 LDA #0
0850 ADC W2
0860 STA W2
0870 INX
0880 CPX DY
0890 BNE SK2
0900 DEC 6
0910 DEC DX
0920 LDA BB
0930 STA W1
0940 LDA B9
0950 STA W2
0960 LDX #0
0970 SK4 INX
0980 CLC
0990 LDA #40
1000 ADC W1
1010 STA W1
1020 LDA #0
1030 ADC W2
1040 STA W2
1050 CPX RX
1060 BNE SK4
1070 LDA #B1 ;rysowanie
1080 LDY RX ;ramki okna
1090 STA (W1),Y
1100 INY
1110 SK5 LDA #B2
1120 STA (W1),Y
1130 INY
1140 CPY DX
1150 BNE SK5
1160 LDA #69
1170 STA (W1),Y
1180 SK6 CLC
1190 LDA #40
1200 ADC W1
1210 STA W1
1220 LDA #0
1230 ADC W2
1240 STA W2
1250 LDA #124
1260 LDY RX
1270 STA (W1),Y
1280 LDY DX
1290 STA (W1),Y
1300 INX
1310 CPX DY
1320 BNE SK6
1330 LDY RX
1340 LDA #90
1350 STA (W1),Y
1360 INY
1370 LDA #B2
1380 SK7 STA (W1),Y
1390 INY
1400 CPY DX
1410 BNE SK7
1420 LDA #67
1430 STA (W1),Y
1440 LDA RX ;przepisanie
1450 STA KX ;wspolrzednych
1460 LDA RY ;lewego rogu do
1470 STA KY ;wspolrzednych
1480 INC KX ;kursora
1490 INC KY
1500 SUK LDY #1 ;bezblednie
1510 RTS ;koniec OPEN
1520 ;
1530 PUT TAX ;poczatek PUT
1540 LDA RX ;ustawianie
1550 STA B2 ;marginesow
1560 LDA DX
1570 STA B3
1580 LDA B4 ;odkladanie na
1590 PHA ;stos biezacej
1600 LDA B5 ;pozycji kursora
1610 PHA
1620 LDA KX ;ustawienie
1630 STA B5 ;kursora dla
1640 LDA KY ;okienka
1650 STA B4
1660 TXA
1670 CMP #125 ;sprawdzenie czy
1680 BNE NIE ;czyszcic okno
1690 PLA
1700 STA B5 ;jesli tak, to
1710 PLA ;rysuj okienko
1720 STA B4 ;od nowa
1730 INC DX
1740 JMP DTW
1750 NIE JSR PUTCH ;wyswietlanie
1760 LDA B5 ;znaku
1770 STA KX
1780 LDA B4
1790 STA KY
1800 LDA KX
1810 CMP RX
1820 BNE PB1
1830 LDA DX
1840 STA KX
1850 DEC KX
1860 DEC KY
1870 PB1 CMP DX
1880 BNE PB2
1890 LDA RX
1900 STA KX
1910 INC KX
1920 INC KY
1930 PB2 LDA KY
1940 CMP RY
1950 BNE PB3
1960 LDA DY
1970 STA KY
1980 DEC KY
1990 PB3 CMP DY
2000 BNE PB4
2010 LDA RY
2020 STA KY
2030 INC KY
2040 PB4 PLA ;podnawianie
2050 STA B5 ;polozenia
2060 PLA ;kursora
2070 STA B4
2080 LDA #2
2090 STA B2
2100 LDA #39 ;ustawianie
2110 STA B3 ;marginesow
2120 LDA #30
2130 JSR PUTCH
2140 LDA #31
2150 JSR PUTCH
2160 LDY #1 ;sukces
2170 RTS ;i powrot

```

ATARI

Drodzy Czytelnicy!

Do naszej redakcji nadchodzi wiele listów, w których zgłaszacie swoje uwagi i prośby dotyczące redagowania „Bajtka” jako całości oraz poszczególnych klanów i wydań specjalnych. W miarę możliwości zawsze staraliśmy się reagować na te sygnały. Aby móc czynić to jeszcze lepiej, zamieściliśmy w 9 numerze „Bajtka” ankietę dotyczącą redagowania naszego pisma. Zachęcamy do jej wypełnienia i nadesłania do redakcji. Bardzo nam to ułatwi pracę!

Natomiast w niniejszym wydaniu dodatkowym zamieszczamy ankietę „pomocniczą” dotyczącą wyłącznie redagowania „Bajtek”. Prosimy o uważne przeczytanie i podkreślenie lub wpisanie odpowiedzi. Wypełnioną ankietę należy włożyć w kopertę i wysłać na adres redakcji: „Bajtek”, 00-687 Warszawa, ul. Wspólna 61, z dopiskiem na kopercie „Ankieta-Atari”.

Ankieta jest anonimowa, lecz wśród Czytelników, którzy zechcą podać swoje nazwisko i adres, zostanie rozlosowanych kilka programów na Atari.

1. Posiadany komputer:

- Atari 600XL (16KB RAM)
- Atari 800XL lub 600XL z 64 KB RAM
- Atari 65XE
- Atari 130XE lub 256 XT
- Atari 520ST
- Atari 1040ST
- inny (jaki?).....
- żaden

2. Posiadana pamięć masowa:

- stacja dysków i magnetofon firmowy
- stacja dysków
- magnetofon firmowy
- inny magnetofon
- żadna

3. Posiadany monitor

- telewizor czarno-biały
- telewizor kolorowy SECAM
- telewizor kolorowy PAL
- monitor monochromatyczny
- monitor kolorowy
- żaden.

4. Posiadana drukarka.

- Atari
- Star
- inna (jaka?)
- żadna

5. Jakie urządzenia peryferyjne planujesz nabyć w najbliższym czasie?

6. Źródło nabycia komputera:

- Pewex
- w kraju za złotówki (giełda, komis itp.)
- import prywatny.

7. Jak długo posiadasz komputer:

- pół roku lub mniej
- rok lub mniej
- półtora roku lub mniej
- dwa lata lub mniej
- dłużej

8. Czy to Twój pierwszy komputer:

- tak
- nie — poprzednio

9. Czy sam programujesz?

- nie
- czasami
- często
- jeżeli tak, to w jakich językach najczęściej?

ANKIETA

10. Czy należysz do klubu komputerowego, szkolnego koła lub innej grupy zainteresowań?
- tak
 - nie
11. Czy z Twojego komputera korzysta jeszcze ktoś oprócz Ciebie?
- rodzice
 - dzieci
 - koledzy
 - nikt
12. Najczęściej używane programy (wymień dwie lub trzy grupy):
- gry zręcznościowe
 - gry strategiczne
 - gry przygodowe (adventure)
 - gry symulacyjne
 - edytory tekstu
 - programy graficzne
 - programy muzyczne
 - programy obliczeniowe
 - programy edukacyjne
 - bazy danych
 - języki programowania
 - inne programy użytkowe.
13. Jakich materiałów jest w „Bajtku-Atari”:
- | za mało | odpowiednio | za dużo | |
|---------|-------------|---------|--|
| X | X | X | opisy gier |
| X | X | X | wydruki gier (do wpisania) |
| X | X | X | opisy programów użytkowych |
| X | X | X | instrukcje programów użytkowych |
| X | X | X | programy użytkowe (do wpisania) |
| X | X | X | opisy i listy sprzętu |
| X | X | X | modyfikacje sprzętu, interfejsy itp. |
| X | X | X | recenzje książek |
| X | X | X | recenzje programów |
| X | X | X | opisy języków programowania |
| X | X | X | kursy programowania |
| X | X | X | opisy i adresy klubów |
| X | X | X | nowości rynku komputerowego |
| X | X | X | wywiady i reportaże |
| X | X | X | podstawy działania komputerów i urządzeń peryferyjnych |
14. Wiek: lat.
15. Płeć
- mężczyzna
 - kobieta
16. Wykształcenie:
- uczeń szkoły podstawowej
 - uczeń szkoły zawodowej
 - uczeń technikum lub liceum
 - student — kierunek techniczny
 - student — kierunek nietechniczny
 - wykształcenie podstawowe
 - średnie techniczne
 - średnie nietechniczne
 - wyższe techniczne
 - wyższe humanistyczne
17. Miejsce zamieszkania:
- Wieś
 - Miasto do 5 tys. mieszkańców
 - miasto 5–25 tys. mieszkańców
 - miasto 25–50 tys. mieszkańców
 - miasto 50–100 tys. mieszkańców
 - miasto 100–250 tys. mieszkańców
 - miasto 250–500 tys. mieszkańców
 - miasto ponad 500 tys. mieszkańców
- Podaj województwo

OPTYMALIZACJA ZAPISU NA DYSKIETCE

Przełóżając zawartość naszych dyskietek zauważamy, że na wielu z nich jest jeszcze sporo wolnego miejsca, zbyt mało jednak do zapisania nowych programów. Gdyby jednak połączyć wszystkie wolne sektory na różnych dyskietkach w jedną całość, okazałoby się, że można w nich zapisać sporo programów.

Istnieje więc problem optymalnego zagospodarowania dyskietek, szczególnie przy ograniczonej możliwości ich zakupu. Próba rozwiązania tego problemu jest przedstawiony niżej program zrealizowany w dwóch wersjach: Atari BASIC i BASIC XE.

Dla zilustrowania korzyści zastosowania opisywanego programu podaję wyniki optymalizacji 6 dyskietek:

przed optymalizacją	po optymalizacji
MRDO 324	GHSTBUS 370
SPINDIZ 276	BOULDER 184
GHSTBUS 370	POLEPOS2 164
040 FREE SECTORS	MONOPOL 164
	BOULDER2 128
	000 FREE SECTORS
	DYSK 2
OSTFRONT 111	MRDO 324
ZENJI 066	SEADRAG 127
ARCHON 222	JUMBOJET 123
QIX 132	OSTFRONT 111
REVERSI 034	RATSREV 087
TEATREUR 260	QBERT 070
MONOPOL 164	PLAMANIA 067
021 FREE SECTORS	KEYSTONE 067
	REVERSI 034
	000 FREE SECTORS
	DYSK 3
FORTAPOC 135	DROPZONE 308
SEADRAG 127	DONSBALL 242
BOULDER2 128	COLCHESS 234
TAPPER 214	ARCHON 222
BLUEMAX 192	004 FREE SECTORS
BOULDER 184	
030 FREE SECTORS	
	DYSK 4
DROPZONE 308	SPINDIZ 276
PLANET 029	TEATREUR 260
BLUEMAX2 255	BLUEMAX2 255
NINJAMAS 232	TAPPER 214
PHARAOH 151	005 FREE SECTORS
035 FREE SECTORS	
	DYSK 5
POLEPOS 134	BEHIND 263
QUASIMOD 188	NINJAMAS 232
JUMBOJET 123	BLUEMAX 192
BEHIND 263	QUASIMOD 188
POLEPOS2 164	FORTAPOC 135
PITSTOP 133	000 FREE SECTORS
005 FREE SECTORS	
	DYSK 6
TRIAD 235	TRIAD 235
COLCHESS 234	PHARAOH 151
RATSREV 087	POLEPOS 134
QBERT 070	PITSTOP 133
PLAMANIA 067	QIX 132
DONSBALL 242	ZENJI 066
KEYSTONE 067	PLANET 029
008 FREE SECTORS	130 FREE SECTORS

Z przedstawionych wyników widać, że ze 139 wolnych sektorów na 6 dyskietkach, po optymalizacji uzyskano na jednej dyskietce 130 wolnych sektorów. Pozwala to na zapisanie na tej dyskietce dodatkowych programów.

OPIS METODY

Do rozwiązania omawianego problemu niezmiernie trudno jest zastosować znane z literatury metody optymalizacji (bardzo złożone algorytmy, duża objętość programu, długi czas realizacji). Postanowiłem więc zastosować metodę polegającą na wyborze z różnych

układów programów takiego, który najlepiej wypełnia dyskietkę. Wymaga to tworzenia zbioru ciągów złożonych z różnych programów poddawanych optymalizacji. Każdy z takich ciągów zawiera maksymalną liczbę programów jaka może się zmieścić na jednej stronie dyskietki. Z tak utworzonego zbioru wybierany jest ciąg dający maksymalne wypełnienie dyskietki.

Tworzenie zbioru ciągów jest przerywane gdy:

- pojawia się ciąg programów całkowicie wypełniających dyskietkę,
- moc zbioru ciągów będzie równa 100 (utworzonych zostanie 100 różnych ciągów programów).

Po wybraniu optymalnego ciągu liczba programów do optymalizacji zostaje pomniejszona o programy zawarte w tym ciągu. Proces ten powtarzany jest do momentu „rozlokowania” wszystkich programów.

W metodzie tej zastosowałem również odpowiednio zorganizowany proces tworzenia opisanych wyżej ciągów. Do jej realizacji wymagane jest posortowanie optymalizowanych programów według malejących długości.

OPIS PROGRAMU

Do opracowania programu realizującego omówioną metodę optymalizacji przyjęto następujące ograniczenia:

- maksymalna pojemność dyskietki — 20 programów (na jednej stronie),
- w jednym cyklu optymalizacyjnym można objąć maksymalnie 100 programów,
- programy będą rozmieszczone na dyskietkach zaformatowanych DOS-em 2.5 w rozszerzonej gęstości (tzn. każda dyskietka zawiera 1010 wolnych sektorów). Jeżeli posiadamy dyskietki zaformatowane w innej gęstości, wtedy w linii 20 pod zmienną LSEC należy podstawić właściwą liczbę wolnych sektorów.

Obydwie wersje programu po uruchomieniu żądają podania liczby dyskietek (stron dyskietek), z których mają być pobrane programy do optymalizacji. Następnie wczytywane są spisy zawartości (directory) z kolejnych dyskietek.

Po wczytaniu spisów zawartości, posortowaniu oraz przeprowadzeniu optymalizacji pojawiają się kolejno na ekranie wyniki w postaci:

- numer dyskietki (strony)
- nazwy programów, które należy umieścić na tej dyskietce
- liczba wolnych sektorów.

Porównując obie prezentowane wersje programu łatwo zauważyć, że program napisany w BASIC XE posiada wiele zalet i ułatwień, a przede wszystkim:

- prosta realizacja operacji na tekstach,
- sortowanie dyrektywą SORT-DOWN (w Atari BASIC zastosowano do tego celu metodę Shella),
- możliwość unikania wielu skoków w programie dzięki zastosowaniu instrukcji strukturalnych,
- program w BASIC XE jest krótszy i działa znacznie szybciej niż w Atari BASIC (dla podanego przykładu program ten działał ponad 4-krotnie szybciej),
- wydruk programu w BASIC XE jest bardziej czytelny ze względu na małe litery oraz wcięcia.

Należy zaznaczyć, że program w BASIC XE będzie działał tylko z rozszerzeniem wczytanym z dysku systemowego (chodzi o dyrektywę SORTDOWN). Nie należy również poddawać optymalizacji dyskietek, na których zapisany jest DOS, gdyż będzie on również poddany optymalizacji.

Zaprezentowany program optymalizacji zapisów na dyskietkach był wielokrotnie stosowany przeze mnie i moich znajomych. Uzyskane wyniki okazały się bardzo korzystne. Na przykład moje dyskietki w większości przypadków mają całkowicie wykorzystane pojemności. Wyniki stosowania programu zachęciły mnie do napisania niniejszego artykułu. Podkreślić trzeba, że program prowadzi tylko do suboptymalizacji. Do celów praktycznych jest to jednak — jak się wydaje — w zupełności wystarczające. Zachęcam do podjęcia próby zastosowania programu do optymalizacji swoich zbiorów.

Przemysław Strzelecki

LISTING 1

```

HO 10 REM :PROGRAM "OPTYMALIZACJA"
GC 20 CLR :LSEC=1010
MV 30 DIM FO$(17),F1$(1500),P(100),C(100,
30),T$(15),M(100),Z(10,30)
EQ 40 REM :WCZYTANIE DIRECTORY Z DYSKU
WT 50 ? CHR$(125):SETCOLOR 2,0,0:? :? :?
"Podaj liczbe dyskow " :INPUT DK:? CHR
$(125)
RC 60 SETCOLOR 2,1,0:TRAP 160
SI 70 N=0
ZI 80 POSITION 6,10:? "Wloz dysk i nacisn
ij START "
LE 90 IF PEEK(53279)<>6 THEN 90
AS 100 ? CHR$(125)
QE 110 OPEN #1,6,0,"D:*. *"
YH 120 INPUT #1,FO$
YK 130 IF FO$(5,16)="FREE SECTORS" THEN 1
60
SD 140 N=N+1:F1$(15*N-14,15*N)=FO$(3,17)
MM 150 GOTO 120
LJ 160 CLOSE #1
PJ 170 IF DK=1 THEN 200
DB 180 DK=DK-1:GOTO 80
GY 190 REM :SORTOWANIE PROGRAMOW
LI 200 SI=N
WV 210 IF SI<1 THEN 350
GJ 220 SI=INT(SI/2)
EU 230 F=0
NM 240 FOR I=1 TO N-SI
FN 250 WEC1=VAL(F1$(15*I-2,15*I)):WEC2=VA
L(F1$(15*(I+SI)-2,15*(I+SI)))
WK 260 IF WEC1>WEC2 THEN 300
FJ 270 F=1
GW 280 T$=F1$(15*(I+SI)-14,15*(I+SI)):F1$
(15*(I+SI)-14,15*(I+SI))=F1$(15*I-14,1
5*I)
IE 290 F1$(15*I-14,15*I)=T$
FT 300 NEXT I
RZ 310 IF F=0 THEN 210
ND 320 GOTO 230
UL 330 REM :PRZEPISANIE DLUGOSCI PROGRAMO
W DO TABLICY P(I)
HK 340 REM :ORAZ UZUPELNIENIE TABLICY IND
EKSOV
QM 350 FOR I=1 TO N:P(I)=VAL(F1$(15*I-2,1
5*I)):M(I)=I:NEXT I
DQ 360 LP=0:D=1:J=1:E=1:A=2
YY 370 REM :PO ZALOZENIU STALYCH DO ROBOT
Y !!!
XV 380 REM :GLOWNA CZESC PROGRAMU
YB 390 FOR I=1 TO N:LP=LP+P(I):NEXT I
ZV 400 IF LP=0 THEN 750
VG 410 IF LP>LSEC THEN 430
RA 420 C(J,0)=LSEC-LP:FOR I=1 TO N:C(J,I)
=M(I):NEXT I:GOTO 580
LF 430 L=LSEC-P(1):C(J,1)=M(1)
FY 440 Y=1:FOR I=A TO N STEP D
YQ 450 LP=0:FOR X=I TO N:LP=LP+P(X):NEXT
X
GK 460 IF L>=LP THEN 510
OS 470 IF L<P(I) THEN 490
KG 480 Y=Y+1:L=L-P(I):C(J,Y)=M(I):IF L=0
THEN C(J,0)=L:GOTO 580
XQ 490 NEXT I:C(J,0)=L:J=J+1:IF J>100 THE
N 540
QW 500 A=A+1:GOTO 430
TN 510 L=L-LP:C(J,0)=L:FOR I=X-1 TO N:Y=Y
+1:C(J,Y)=M(I):NEXT I
AU 520 IF L=0 THEN 580
LD 530 D=D+1:A=2:IF D<=3 THEN 430
CV 540 C=1100
UQ 550 FOR I=1 TO J:IF C<=C(I,0) THEN 570
ZX 560 C=C(I,0):X=1
ZT 570 NEXT I:J=X:GOTO 590
PT 580 X=J
QC 590 Z(E,0)=C(X,0):FOR I=1 TO 20
XG 600 Z(E,I)=C(X,I):NEXT I
PZ 610 X=1:FOR I=1 TO 20
KM 620 IF C(J,I)<>M(X) THEN X=X+1:GOTO 62
0
CZ 630 P(X)=0:X=X+1
GE 640 NEXT I
BA 650 FOR I=1 TO J
WM 660 FOR X=0 TO 20
NU 670 C(I,X)=0:NEXT X:NEXT I
ZA 680 X=0:FOR I=1 TO N
AU 690 IF P(I)=0 THEN 710
CV 700 X=X+1:P(X)=P(I):M(X)=M(I)
FZ 710 NEXT I
BT 720 LP=0:D=1:J=1:A=2:E=E+1:N=X:GOTO 39
0
LN 730 REM :PO SKONCZONEJ PRACY WYDRUK OP
TYMALNEGO
NA 740 REM :ROZMIESZCZENIA PROGAMOW NA DY
SKACH
TU 750 FOR Q=1 TO E-1:? CHR$(125);"DYSK N
EE-1R.":Q:? :? "Programy : "
OD 760 FOR I=1 TO 20:IF Z(Q,I)=0 THEN 790
ER 770 NR=Z(Q,I)
IR 780 ? " ";I;" = ";F1$(15*NR-
14,15*NR-3)
GP 790 NEXT I
ZS 800 ? :? "WOLNE SEKTORY ";Z(Q,0)
LA 810 ? :? :? :? :? " *NACISNI
J START*"
AW 820 IF PEEK(53279)<>6 THEN 820
JG 830 NEXT Q
IC 840 ? CHR$(125):? :? :? :? "
KONIEC OBLICZEN"

```

LISTING 2

```

VC 10 Rem :PROGRAM "OPTYMALIZACJA"
EH 20 CLR :Fast :Lsec=1010
GI 30 Dim FO$(17),F1$(100,15),P(100),C(10
0,20),T$(15),M(100),Z(10,20)
SE 40 Rem :WCZYTANIE DIRECTORY Z DYSKU
CK 50 ? CHR$(125):Setcolor 2,0,0:? :? :In
put "Podaj liczbe dyskow ",Dk
IN 60 Setcolor 2,1,0:Trap 170
SI 70 N=0
BV 80 While Dk
ED 90 ? CHR$(125):Position 6,11:? "Wloz d
ysk i nacisnij START "
XH 100 If Peek(53279)<>6 Then 100
HX 110 ? CHR$(125):Open #1,6,0,"D:*. *"
RT 120 Q=1:While Q
JH 130 Input #1,FO$
FW 140 If FO$(5,16)="FREE SECTORS" Then Q
=0:Goto 160
OO 150 N=N+1:F1$(N;)=FO$(3,17)
BK 160 Endwhile
WJ 170 Close #1
MA 180 If Dk=1 Then 210
NR 190 Dk=Dk-1:Endwhile
WH 200 Rem :SORTOWANIE PROGRAMOW
VS 210 Sortdown F1$ Using ;13,15
KI 220 Rem :PRZEPISANIE DLUGOSCI PROGRAMO
W DO TABLICY P(I)
XH 230 Rem :ORAZ UZUPELNIENIE TABLICY IND
EKSOV
EO 240 For I=1 To N:P(I)=Val(F1$(I;13)):M
(I)=I:Next I
KX 250 Lp=0:D=1:J=1:E=1:A=2
OV 260 Rem :PO ZALOZENIU STALYCH DO ROBOT
Y !!!
NS 270 Rem :GLOWNA CZESC PROGRAMU
OC 280 For I=1 To N:Lp=Lp+P(I):Next I
Wl 290 If Lp=0 Then 680
IW 300 If Lp<=Lsec:C(J,0)=Lsec-Lp:For I=1
To N:C(J,I)=M(I):Next I:Goto 480
TK 310 Endif
SI 320 L=Lsec-P(1):C(J,1)=M(1)
ML 330 Y=1:For I=A To N Step D
TH 340 Lp=0:For X=1 To N:Lp=Lp+P(X):Next
X
WB 350 If L<Lp:If L>=P(I):Y=Y+1:L=L-P(I):
C(J,Y)=M(I)
OD 360 If L=0 Then C(J,0)=L:Goto 480
TW 370 Endif
OO 380 Next I:C(J,0)=L:J=J+1:If J>100 The
n 440
MF 390 A=A+1:Goto 320
MQ 400 Else :L=L-Lp:C(J,0)=L:For I=X-1 To
N:Y=Y+1:C(J,Y)=M(I):Next I
TL 410 Endif
HE 420 If L=0 Then 480
IK 430 D=D+1:If D<=3 Then A=2:Goto 320
CU 440 C=1100
VY 450 For I=1 To J:If C>C(I,0):C=C(I,0):
X=I
TV 460 Endif
NF 470 Next I:J=X:Goto 490
PS 480 X=J
WZ 490 Z(E,0)=C(X,0):For I=1 To 20
WT 500 Z(E,I)=C(X,I):Next I
LY 510 X=1:For I=1 To 20
GM 520 While C(J,I)<>M(X):X=X+1:Endwhile
CY 530 P(X)=0:X=X+1
FZ 540 Next I
IF 550 For I=1 To J
DR 560 For X=0 To 20
BT 570 C(I,X)=0:Next X:Next I
UZ 580 X=0:For I=1 To N
XN 590 If P(I)<>0
CU 600 X=X+1:P(X)=P(I):M(X)=M(I)
TN 610 Endif
FW 620 Next I
BV 630 Lp=0:D=1:J=1:A=2:E=E+1:N=X:Goto 28
0
BO 640 Rem :PO SKONCZONEJ PRACY WYDRUK OP
TYMALNEGO
DB 650 Rem :ROZMIESZCZENIA PROGAMOW NA DY
SKACH
TT 660 For Q=1 To E-1:? CHR$(125);"DYSK N
R.":Q:? :? "Programy : "
ZB 670 For I=1 To 20:If Z(Q,I)=0 Then 700
MC 680 Nr=Z(Q,I)
IT 690 ? " ";I;" = ";F1$(Nr;1,1
1)
FT 700 Next I
ZT 710 ? :? "WOLNE SEKTORY ";Z(Q,0)
LB 720 ? :? :? :? " *NACISNI
J START*"
HO 730 If Peek(53279)<>6 Then 730
JD 740 Next Q
DE 750 ? Chr$(125):? :? :? :? "
KONIEC OBLICZEN"

```


BULDOGER COPY

Chyba wszyscy użytkownicy komputerów Atari, wykorzystujący magnetofon jako pamięć masową, wiedzą ile kłopotów może sprawić starzejąca się taśma. Mała szybkość transmisji, a co za tym idzie bardzo długie programy, przy niskiej jakości taśm dostępnych w Polsce za złotówki powodują, że często podczas wpisywania do komputera programu zapisanego na takiej taśmie występują błędy transmisji.

Zwłaszcza częste są sytuacje, w których są kłopoty z wpisaniem programu, który jeszcze poprzedniego dnia wpisywał się bez przeszkód. Powodem jest starzenie się nośnika, a także zapisu na taśmie. Taki zapis należy odświeżyć przez ponowne nagranie, jednakże skopiowanie go, gdy nie posiada się go na innej taśmie może zniechęcić nawet najbardziej cierpliwych użytkowników. Z reguły taki program uważa się za stracony. Próba ułatwienia sobie życia w takich i podobnych sytuacjach jest program kopiujący "BULDOGER COPY".

Jego podstawową zaletą wyróżniającą go spośród popularnych programów kopiujących na Atari jest fakt, iż przy wystąpieniu błędu transmisji podczas wprowadzania nie trzeba cofać taśmy do początku programu, lecz wystarczy tylko cofnąć o 2 bloki. Program próbuje przejść ponownie krytyczny punkt. Jeśli znów wystąpi błąd transmisji to sytuacja się powtarza, a jeśli nie, to wprowadzanie jest kontynuowane. W przypadku, gdy prawidłowe przejście krytycznego punktu jest niemożliwe, należy przerwać wprowadzanie klawiszem BREAK lub RESET. W takiej sytuacji pliku nie da się już odtworzyć. Opisana własność programu kopiującego pozwala wpisać i skopiować program, w którym zdarzają się błędy transmisji w wielu miejscach, jednak każdy z prawdopodobieństwem mniejszym od 1. Również możliwe jest złożenie programu z kilku osobnych fragmentów (np. znajdujących się na różnych taśmach), które mogą być niekompletne, jednakże pod warunkiem, że wszystkie w sumie zawierają kompletny plik. BULDOGER COPY sam znajdzie punkty kontynuacji i złoży wszystkie fragmenty w całość.

Do innych zalet programu BULDOGER COPY należą:
 — liczniki bloków liczące na bieżąco, osobne dla wprowadzania i wyprowadzania,
 — sygnalizacja momentów zakończenia otwarcia kanałów przez pojawienie się stanu 000 odpowiedniego licznika,
 — sygnalizacja na bieżąco bajtu statusu (STB) operacji we/wy,
 — ignorowanie błędów transmisji w bloku EOF,
 — zabezpieczenie przed kasowaniem po wciśnięciu RESET,
 — możliwość wyboru długości przerw między blokami,
 — aktywny klawisz BREAK, pomocny w niepożądanych sytuacjach,
 — sygnalizacja ewentualnego przepelnienia pamięci,
 — prosta obsługa za pomocą klawiszy konsoli.

Wadą tego programu jest tolerowanie tylko standardowych formatów oraz współpraca tylko z magnetofonami. Jednakże w zdecydowanej większości zastosowań nie ma to żadnego znaczenia.

Co trzeba zrobić, aby posiadać BULDOGER COPY? Należy wpisać poniższy program w Basicu i uruchomić go instrukcją RUN. Jeżeli nie został popełniony błąd przy wpisywaniu, to komputer po kilkunastu sekundach oznajmi gotowość do wprowadzenia na taśmę właściwego BULDOGER COPY. Ten właśnie program wpisuje się później do komputera włączając go z wciśniętymi klawiszami START i OPTION.

Dla interesujących się programowaniem w języku maszynowym zamieszczamy dodatkowo program źródłowy w assemblerze 6502. Oczywiście nie trzeba go wpisywać dla uzyskania właściwego programu BULDOGER COPY.

BULDOGER COPY kopiuje pliki o długości do 47 KB.

Grzegorz Mikrut

```
VQ 10 ? CHR$(125);"PROSZE CZEKAC":? :?
AO 15 RESTORE
WW 20 DIM A$(918)
KQ 30 FOR I=0 TO 33:SK=0
LU 40 FOR J=1 TO 27:READ P:SK=SK+P:A$(27*
I+J)=CHR$(P):NEXT J
OE 50 READ SK1:IF SK<>SK1 THEN 120
IU 60 NEXT I
VL 70 ? "USTAW MAGNETOFON DO NAGRYWANIA,N
ASTE- PNIE WCISNIJ <RETURN>"
CI 80 OPEN #1,8,128,"C:"
XF 90 ? #1;A$
KX 100 CLOSE #1
NS 110 END
NP 120 POP :? "BLAD DANYCH W LINII ":500+
10*1:LIST 500+10*1
HG 130 ? :? "ZLICZONA SUMA KONTROLNA WYNO
SI ":SK:?:? "PO POPRAWNIENIU BLEDU NAP
ISZ <RUN>":? "I WCISNIJ <RETURN>"
NY 140 END
TM 500 DATA 0,8,248,5,25,7,24,96,112,112,
112,71,41,6,112,6,6,112,112,7,7,7,112,
112,6,112,6,1584
EL 510 DATA 112,112,1,25,6,7,65,0,6,7,6,6
,65,0,6,7,71,5,7,65,0,6,0,0,0,34,53,67
2
RW 520 DATA 44,36,47,39,37,50,0,35,47,48,
57,0,0,0,0,34,57,0,39,50,58,37,39,47
,50,58,909
LU 530 DATA 0,45,41,43,50,53,52,0,0,0,52,
33,50,46,47,34,50,58,37,39,0,17,25,24,
23,0,0,819
SM 540 DATA 0,0,51,52,33,50,52,0,13,52,47
,0,44,47,33,36,41,46,39,0,0,0,51,37,44
,37,35,840
VG 550 DATA 52,13,52,47,0,55,50,41,52,41,
46,39,0,0,0,47,48,52,41,47,46,13,48,33
,53,51,37,1004
AD 560 DATA 51,26,51,40,47,50,52,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,317
TZ 570 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,10,10,10,0,30
TE 580 DATA 37,50,50,47,50,0,10,10,10,0,0
,0,0,50,37,55,41,46,36,0,52,33,48,37,0
,51,37,787
NH 590 DATA 54,37,50,33,44,0,50,37,35,47,
50,36,51,0,6,0,48,50,37,51,51,0,40,37,
44,48,0,936
KP 600 DATA 0,0,47,53,52,0,47,38,0,45,37,
45,47,50,57,0,0,0,0,162,0,160,6,140,49
,2,142,1179
QF 610 DATA 48,2,142,68,2,162,25,160,7,13
4,10,132,11,32,214,253,216,169,254,141
,143,9,169,122,141,196,2,2964
FQ 620 DATA 160,0,32,172,7,165,62,240,1,2
32,173,160,6,240,1,232,224,14,240,23,1
73,31,208,201,7,240,249,3493
PK 630 DATA 162,255,142,252,2,201,6,208,3
,76,196,8,201,5,240,105,169,128,69,62,
133,62,240,5,162,4,76,3172
DI 640 DATA 118,7,162,9,160,5,189,141,7,1
53,155,6,202,136,208,246,169,10,101,20
,197,20,208,252,76,80,7,3044
HH 650 DATA 51,40,47,50,52,44,47,46,39,0,
185,164,7,141,22,6,185,165,7,141,23,6,
96,25,6,24,6,1625
CE 660 DATA 28,6,34,6,162,6,185,187,7,157
,161,6,200,232,224,13,208,244,96,0,50,
37,33,36,57,1,55,2431
OG 670 DATA 50,41,52,41,46,39,44,47,33,36
,41,46,39,160,7,32,172,7,32,17,8,32,52
,253,32,178,8,1545
AK 680 DATA 160,0,177,6,201,254,240,31,14
1,255,3,160,128,177,6,153,255,3,136,20
8,248,32,127,254,32,110,8,3505
EK 690 DATA 160,1,204,3,3,240,11,160,2,32
,151,7,32,234,253,76,60,7,32,143,8,76,
219,7,160,0,32,2313
VC 700 DATA 151,7,162,12,169,0,157,187,6,
202,208,250,141,140,9,141,141,9,162,14
3,160,9,134,6,132,7,96,2941
TD 710 DATA 32,60,8,140,182,6,142,183,6,1
41,184,6,96,173,140,9,170,32,90,8,72,1
38,32,82,8,170,173,2483
SF 720 DATA 141,9,32,90,8,168,104,96,74,7
4,74,74,32,96,8,96,41,15,32,96,8,96,9,
16,24,168,233,1914
GK 730 DATA 25,48,4,152,105,6,96,152,96,1
62,3,189,139,8,157,194,6,202,16,247,17
3,3,3,170,32,82,8,2478
UG 740 DATA 141,198,6,138,32,90,8,141,199
,6,96,51,52,34,26,248,24,169,1,109,140
,9,141,140,9,144,4,2356
DZ 750 DATA 24,238,141,9,216,165,6,164,7,
133,4,132,5,105,129,133,6,144,3,230,7,
24,96,32,60,8,140,2361
GK 760 DATA 188,6,142,189,6,141,190,6,169
,26,141,191,6,96,160,14,32,172,7,32,17
,8,32,125,228,32,47,2403
IB 770 DATA 8,169,82,32,63,254,32,110,8,1
60,0,173,255,3,145,6,201,254,240,56,17
4,3,3,224,1,240,55,2951
GG 780 DATA 224,143,240,4,224,140,208,31,
173,140,9,208,5,173,141,9,240,21,160,4
,32,151,7,32,214,253,32,3218
ZW 790 DATA 60,9,48,8,160,0,32,151,7,76,2
07,8,160,2,32,151,7,169,254,141,143,9,
32,207,253,76,60,2462
TT 800 DATA 7,209,6,240,5,160,6,76,21,9,1
60,128,185,255,3,145,6,136,208,248,32,
143,8,76,207,8,169,2856
XX 810 DATA 0,141,220,2,205,220,2,240,251
,169,52,141,2,211,32,112,9,169,82,32,6
3,254,173,3,3,201,1,2990
QU 820 DATA 208,15,160,128,177,4,217,255,
3,208,235,136,16,246,169,0,96,201,128,
208,222,169,128,96,173,15,210,3823
HY 830 DATA 41,16,141,142,9,169,3,101,20,
170,173,15,210,41,16,205,142,9,208,238
,228,20,208,242,96,0,0,2863
```

```
0100 *= $05FE
0110 CLC
0120 RTS
0130 ANTIC = *
0140 LAST = 4
0150 NAST = 6
0160 .BYTE "pppG"
0170 .BYTE PA0&#FF,PA0/256
0180 .BYTE 'p,6,6,"pp",7,7,7
0190 .BYTE "pp",6,'p,6,"pp",1
0200 ADR = *
0210 .BYTE STAND&255,STAND/256
0220 BRK = *
0230 STAND = BRK+1
0240 .BYTE 7,65,0,6
0250 LOER = *
0260 .BYTE 7,6,6,65,0,6
0270 AUT = *
0280 .BYTE 7,71,PA0A0M&255
0290 .BYTE PA0A0M/256,65,0,6
0300 PA0 = *
0310 .BYTE 0,0,0,34,"5,$/'%2"
0320 .BYTE 0,"#/09",0,0,0,0,0
0330 .BYTE 34,57,0,"'2:%'/2:"
0340 .BYTE 0,"-)+254",0,0,0
0350 .BYTE "4!2./",34,"2:%'",0
0360 .BYTE 17,25,24,23,0,0,0
0370 PA01 = *
0380 .BYTE 0,"34!24" 0,13,"4/"
0390 .BYTE 0,"/!$).'",0,0
0400 .BYTE 0,"3%,%#4",13,"4/"
0410 .BYTE 0,"72)4).'",0,0
0420 PA02 = *
0430 .BYTE 0,"/04)/.",13,"0!5"
0440 .BYTE "3%3",26,"3(/24"
0450 PA03 = *
0460 .WORD 0,0,0,0,0,0,0,0,0,0,0
0470 PA0NR = *
0480 .WORD 0,0,0,0,0,0,0,0,0,0,0
0490 PA0ER = *
0500 .BYTE 0,0,0,10,10,10,0
0510 .BYTE "%22/2",0,10,10,10
0520 .BYTE 0,0,0,0,"2%7).$",0
0530 .BYTE "4!0%",0,"3%6%2!",0
0540 .BYTE 0,"2%#/2%3",0,6,0
0550 .BYTE "02%33",0,"(%,0"
0560 PA0A0M = *
0570 .BYTE 0,0,0,"/54",0,"/&"
0580 .BYTE 0,"-%-/29",0,0,0,0
0590 START LDX #ANTIC&#FF ;START
0600 LDY #ANTIC/256
0610 STY 561
0620 STX 560
0630 STX 580
0640 LDX #START&#FF
0650 LDY #START/256
0660 STX 10
0670 STY 11
0680 JSR $FDD6
0690 CLD
0700 LDA #FE
0710 STA TEXT
0720 LDA #7A
0730 STA 708
0740 POCZ LDY #0
0750 JSR RLW
0760 LDA #3E
0770 BEQ 20
0780 INX
0790 ZO LDA PA02+19
0800 BEQ 21
0810 INX
0820 Z1 CPX #14
0830 BEQ GTYP
0840 PATRZ LDA #D01F
0850 CMP #7
0860 BEQ PATRZ
0870 LDX #FF
0880 STX 764
0890 CMP #6
0900 BNE TXX
0910 JMP LOADING
0920 TXX CMP #5
0930 BEQ WRITING
0940 GTYP LDA #80 ;OBSLUGA OPTION
0950 EOR #3E
0960 STA #3E
0970 BEQ LONG
0980 LDX #4
0990 JMP LONG3
1000 LONG LDX #9
1010 LONG3 LDY #5
1020 OPT2 LDA OPT1,X
1030 STA PA02+14,Y
1040 DEX
1050 DEY
1060 BNE OPT2
1070 LDA #10
```


1080	ADC #14	1870	TAX	2660	LDX #0303
1090	AX CMP #14	1880	LDA NRREC+1	2670	CPX #1
1100	BNE AX	1890	JSR PRAWY	2680	BEQ DOBRZE
1110	JMP PATRZ	1900	TAY	2690	CPX ##8F
1120	OPT1 = *	1910	PLA	2700	BEQ INN1
1130	.BYTE "3(/24,/.'"	1920	RTS	2710	CPX ##8C
1140	ERROR LDA ER, Y ;ERROR	1930	LEWY LSR A ;LEWY	2720	BNE INNY
1150	STA ADR	1940	LSR A	2730	INN1 LDA NRREC
1160	LDA ER+1, Y	1950	LSR A	2740	BNE NIEZERO
1170	STA ADR+1	1960	LSR A	2750	LDA NRREC+1
1180	RTS	1970	JSR KOD	2760	BEQ INNY
1190	ER = *	1980	RTS	2770	NIEZERO LDY #4
1200	.BYTE STAND##FF, STAND/256	1990	PRAWY AND ##0F ;PRAWY	2780	JSR ERROR
1210	.BYTE BRK##FF, BRK/256	2000	JSR KOD	2790	JSR \$FDD6
1220	.BYTE LOER##FF, LOER/256	2010	RTS	2800	JSR REPLAY
1230	.BYTE AUT##FF, AUT/256	2020	KOD ORA #16 ;KOD	2810	BMI INNY
1240	RLW LDX #6 ;RLW	2030	CLC	2820	LDY #0
1250	RLWP LDA RLWT, Y	2040	TAY	2830	JSR ERROR
1260	STA PA03, X	2050	SBC #25	2840	JMP WPISZ
1270	INY	2060	BMI CYFRA	2850	INNY LDY #2
1280	INX	2070	TYA	2860	TYLE JSR ERROR
1290	CPX #13	2080	ADC #6	2870	LDA ##FE
1300	BNE RLWP	2090	RTS	2880	STA TEXT
1310	RTS	2100	CYFRA TYA	2890	TYLE1 JSR \$FDCf
1320	RLWT = *	2110	RTS	2900	JMP POCZ
1330	.BYTE 0, "2%!*9", 1	2120	STB LDX #3 ;STB	2910	DOBRZE CMP (NAST), Y
1340	.BYTE "72)4).',/!\$).'"	2130	XX LDA STAT, X	2920	BEQ TAK
1350	WRITING LDY #7 ;WRITING	2140	STA PAONR+13, X	2930	LDY #6
1360	JSR RLW	2150	DEX	2940	JMP TYLE
1370	JSR INIT	2160	BPL XX	2950	TAK LDY #128
1380	JSR \$FD34	2170	LDA #0303	2960	PRZEP LDA \$03FF, Y
1390	DALEJW JSR NRECWR	2180	TAX	2970	STA (NAST), Y
1400	LDY #0	2190	JSR LEWY	2980	DEY
1410	LDA (NAST), Y	2200	STA PAONR+17	2990	BNE PRZEP
1420	CMP ##FE	2210	TXA	3000	JSR INCR
1430	BEQ KONW	2220	JSR PRAWY	3010	JMP WPISZ
1440	STA \$03FF	2230	STA PAONR+18	3020	REPLAY LDA #0 ;REPLAY
1450	LDY #128	2240	RTS	3030	STA \$02DC
1460	PISZ LDA (NAST), Y	2250	STAT = *	3040	CZEKAJ CMP \$02DC
1470	STA \$03FF, Y	2260	.BYTE 51, 52, 34, 26	3050	BEQ CZEKAJ
1480	DEY	2270	INCR SED ;INCR	3060	LDA ##34
1490	BNE PISZ	2280	CLC	3070	STA \$D302
1500	JSR \$FE7F	2290	LDA #1	3080	NOWA JSR PRZERWA
1510	JSR STB	2300	ADC NRREC	3090	NIETEN LDA ##52
1520	LDY #1	2310	STA NRREC	3100	JSR \$FE3F
1530	CPY \$0303	2320	BCC RET	3110	LDA \$0303
1540	BEQ DOBW	2330	CLC	3120	CMP #1
1550	LDY #2	2340	INC NRREC+1	3130	BNE ZLE
1560	JSR ERROR	2350	RET CLD	3140	LDY #128
1570	KONW JSR \$FDEA	2360	LDA NAST	3150	KOL LDA (LAST), Y
1580	JMP POCZ	2370	LDY NAST+1	3160	CMP \$03FF, Y
1590	DOBW JSR INCR	2380	STA LAST	3170	BNE NIETEN
1600	JMP DALEJW	2390	STY LAST+1	3180	DEY
1610	INIT LDY #0 ;INIT	2400	ADC #129	3190	BPL KOL
1620	JSR ERROR	2410	STA NAST	3200	LDA #0
1630	LDX #12	2420	BCC JUZINC	3210	RTS
1640	LDA #0	2430	INC NAST+1	3220	ZLE CMP ##80
1650	CLR STA WRSTB, X	2440	CLC	3230	BNE NOWA
1660	WRSTB = PAONR+6	2450	JUZINC RTS	3240	LDA ##80
1670	DEX	2460	NRECWR JSR KODY ;NRECWR	3250	RTS
1680	BNE CLR	2470	STY PAONR+7	3260	PRZERWA LDA \$D20F ;PRZERWA
1690	STA NRREC	2480	STX PAONR+8	3270	AND #16
1700	STA NRREC+1	2490	STA PAONR+9	3280	NST STA STAN
1710	LDX #TEXT##FF	2500	LDA #26	3290	LDA #3
1720	LDY #TEXT/256	2510	STA PAONR+10	3300	ADC \$14
1730	STX NAST	2520	RTS	3310	TAX
1740	STY NAST+1	2530	LOADING LDY #14 ;LOADING	3320	NST1 LDA \$D20F
1750	RTS	2540	JSR RLW	3330	AND #16
1760	NRECRD JSR KODY ;NRECRD	2550	JSR INIT	3340	CMP STAN
1770	STY PAONR+1	2560	JSR \$E47D	3350	BNE NST
1780	STX PAONR+2	2570	WPISZ JSR NRECRD	3360	CPX \$14
1790	STA PAONR+3	2580	LDA ##52	3370	BNE NST1
1800	RTS	2590	JSR \$FE3F	3380	RTS
1810	KODY LDA NRREC ;KODY	2600	JSR STB	3390	NRREC = *
1820	TAX	2610	LDY #0	3400	STAN = **2
1830	JSR PRAWY	2620	LDA \$03FF	3410	TEXT = **3
1840	PHA	2630	STA (NAST), Y	3420	.END
1850	TXA	2640	CMP ##FE		
1860	JSR LEWY	2650	BEQ TYLE1		

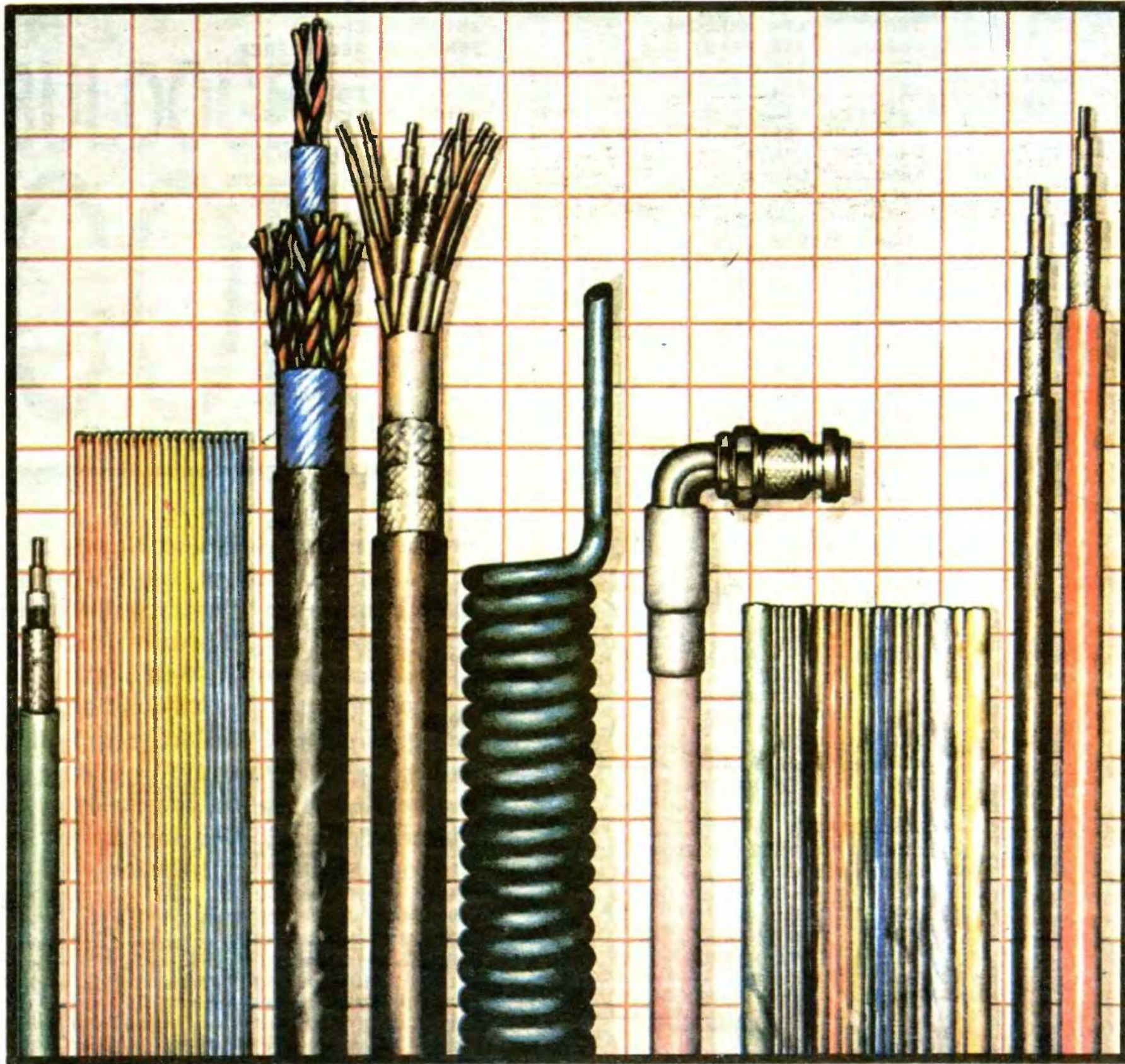
SUPLEMENT DO „PRINT SHOP CONVERTER”

Ostatnio pojawił się nowy program — „Print Shop Companion”, który jest rozwinięciem „Print Shop”. Oprócz dodatkowych funkcji umożliwia on także wprowadzenie zmian w „Print-Shopie”, przez co również ten program uzyskuje nowe możliwości. Zamieszczony poprzednio program „Print Shop Converter” jest niestety niezgodny z nową wersją.

Korzystanie z niego będzie możliwe po wprowadzeniu kilku drobnych zmian. Należy wprowadzić nową linię o numerze 2320 oraz dodać linię o numerach od 2632 do 2636. Tak poprawiony program będzie odczytywał ikony wykonane przez obie wersje „Print Shop” oraz przez „Print Shop Companion”.

Wojciech Zientara

DT	2320 ON D\$(N*32-12, N*32-11)<>"X@" AND D\$(N*32-12, N*32-11)<>"x@" GO# BAD
CJ	2632 IF ASC(D\$(N*32-12))=95 THEN ? "To jest zestaw znakow"
FD	2634 IF ASC(D\$(N*32-12))=144 THEN ? "T o jest kalendarz"
MH	2636 IF ASC(D\$(N*32-11))=148 THEN ? "T o jest ramka"



ZŁĄCZA

Znaczna część listów nadchodzących do redakcji dotyczy złączy stosowanych w komputerach Atari. Mam nadzieję, że poniższy opis zadowoli wszystkich użytkowników modeli XL i XE.

Poszczególne modele komputerów Atari różnią się między sobą jedynie złączem szyny równoległej. Natomiast pozostałe złącza są prawie identyczne, lecz mają różne nazwy. Komputery serii XL posiadają następujące złącza: POWER IN, MONITOR, PARALLEL BUS, CARTRIDGE SLOT, PERIPHERAL i CONTROLLERS. W komputerach serii XE znajdują się: POWER ADAPTER PLUG, MONITOR JACK, CARTRIDGE SLOT, ENHANCED CARTRIDGE INTERFACE (ECI), SERIAL I/O PORT i CONTROLLER PORTS. Celowo pomijam w tym wykazie gniazdo do połączenia z wej-

ciem antenowym telewizora, gdyż nie ma w nim nic interesującego, a zastosowanie jest oczywiste.

POWER IN / POWER ADAPTER PLUG: złącze zasilane, przez które jest doprowadzane do komputera napięcie 5 V z zewnętrznego zasilacza. Jest to siedmiostykowe gniazdo DIN.

MONITOR: wyjście sygnałów fonii i wizji małej częstotliwości. Standardowe pięciostykowe gniazdo DIN. Sygnał **Audio in** może być doprowadzony bezpośrednio do wejścia dowolnego wzmacniacza akustycznego. Sygnał **Composite luminance** jest wykorzystywany przez monitory monochromatyczne, a **Composite video** przez kolorowe.

Ostatni sygnał — **Composite chroma** — występuje tylko w starszych egzemplarzach 800XL.

CARTRIDGE SLOT (w XL nie opisany, na górnej pokrywie obudowy): złącze krawędziowe służące do dołączenia modułów ROM. W modelach

XE pełni jednocześnie funkcję części szyny równoległej.

ENHANCED CARTRIDGE INTERFACE (tylko XE): złącze krawędziowe stanowiące uzupełnienie **CARTRIDGE SLOT**. Oba te złącza razem zastępują w modelach XE szynę równoległą, która nie została wyprowadzona na zewnątrz obudowy.

PARALLEL BUS: złącze krawędziowe (krawędź płytki drukowanej ze stykami) umożliwiające bezpośredni dostęp do magistrali systemu. System operacyjny Atari umożliwia obsługę 10 urządzeń zewnętrznych przyłączonych do szyny równoległej. Aktualnie dostępne są (za Zachodzie) jedynie dyski sztywne 20 MB i specjalne odmiany stacji dysków elastycznych. W Atari 600XL z tego złącza korzysta zewnętrzny moduł pamięci 64 kB RAM.

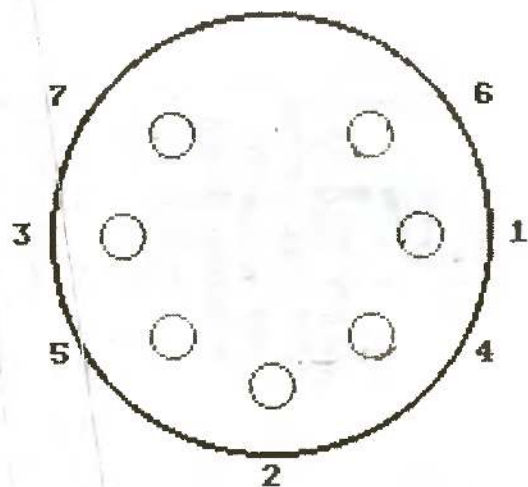
PERIPHERAL / SERIAL I/O PORT: szeregowo złącze wejścia/wyjścia według standardu Atari. Tu podłączana jest większość urządzeń peryferyjnych. W starszych modelach na styk 10 było wyprowadzone napięcie 5 V, a styk 12 nie był wykorzystany. Nowsze modele mają napięcie 12 V na styku 12 i sygnał **Ready** na styku 10.

CONTROLLERS / CONTROLLER PORTS: standardowe gniazda joysticków, paddle i pióra świetlnego. W Atari — dzięki możliwości programowania PIA — są również wykorzystywane do innych urządzeń zewnętrznych.

Teraz trochę informacji dla Czytelników nieco bardziej zainteresowanych „wnętrznymi” Atari. Sygnały A/0 A-15 i D0-D7 to oczywiście linie adresowe i danych magistrali systemu. Duża część pozostałych sygnałów również ma swoje odpowiedniki w innych komputerach.

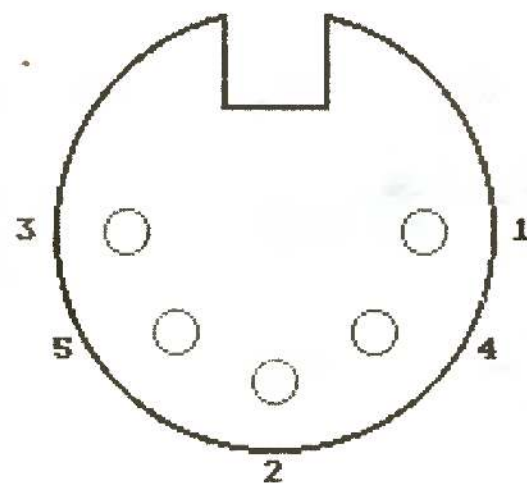
Szyna równoległa i cartridge

Reset — wejście asynchroniczne do zerowania układów wewnętrznych. **IRQ** — wejście żądania przerwania (zwarne wewnętrznie przez rezystancję 3 kOhm). **Ready** — wejście sygnału gotowości (także zwarte przez 3 kOhm). **R/W** — buforowane wyjście sygnału Read/Write CPU. **Ref** — sygnał odświeżania pamięci dynamicznych RAM. Sygnały **CAS**, **RAS**, **Casinh** i **D1xx** służą do sterowania pamięcią i wymagałyby szerszego omówienia. **ExtSel** — wejście sygnału odłączającego wewnętrzną pamięć RAM. **MPDis** — wejście sygnału odłączającego obszar ROM z procedurami zmiennoprzecinkowymi (Math-Pack: \$d800-\$dfff). **Audio in** — wejście sygnału akustycznego małej częstotliwości. **En4** i **En5** — wejścia sygnałów odłączających pamięć RAM w obszarach \$8000-\$9fff i \$A000-\$Bfff; o stanie tych obszarów informują sygnały **Sel4** i **Sel15**. Sygnał I/O5 — steruje obszarem pamięci \$D500-\$D5ff, który jest używany jako obszar adresowany I/O cartridge'a.



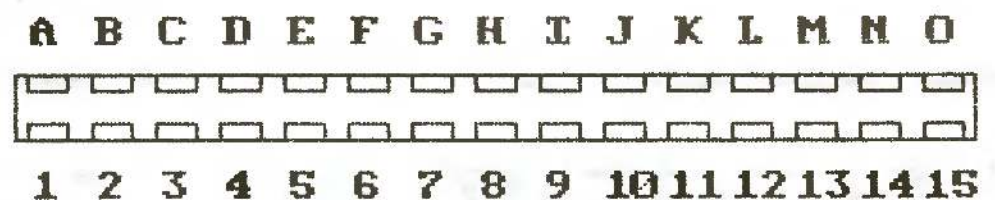
POWER IN

- 1. +5V
- 2. ekran
- 3. masa
- 4. +5V
- 5. masa
- 6. +5V
- 7. masa



MONITOR

- 1. Composite luminance (sygnał luminancji)
- 2. masa
- 3. Audio out (sygnał fonii — m.cz.)
- 4. Composite video (złożony sygnał wizji)
- 5. Composite chroma (sygnał koloru)

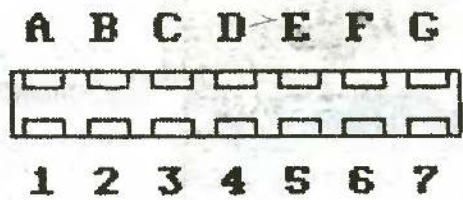


CARTRIDGE SLOT

- 1. Sel4
- 2. A3
- 3. A2
- 4. A1
- 5. A0
- 6. D4
- 7. D5
- 8. D2
- 9. D1;
- 10. D0
- 11. D6
- 12. Sel5
- 13. +5V
- 14. En5
- 15. I/O5
- A. En4
- B. masa
- C. A4
- D. A5
- E. A6
- F. A7
- G. A8
- H. A9
- I. A12
- J. D3
- K. D7L. A11
- M. A10
- N. R/W
- O. zegar (faza 2)

Złącze szeregowe

Data input i output — linie szeregowego przesyłania danych do i z komputera. **Clock input i output** — linie przesyłania sygnałów zegarowych przy transmisji synchronicznej. **Command** — wyjście poleceń komputera dla urządzeń zewnętrznych. **Motor control** — sygnał sterujący silnikiem magnetofonu. **Proceed** — wyjście sygnałów z urządzeń zewnętrznych. **Interrupt** — wejście żądania przerwania z urządzeń zewnętrznych.



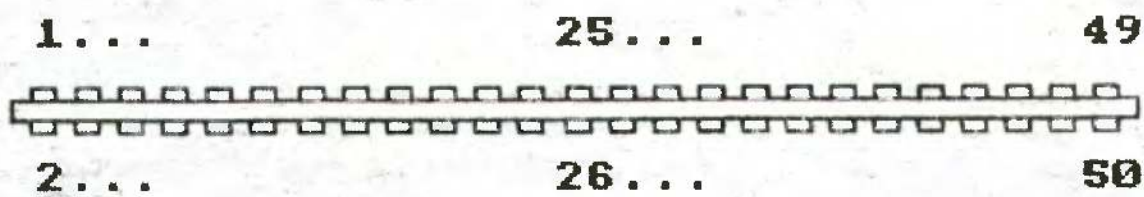
ENHANCED CART-RIDGE INTERFACE

Schematy

Zamieszczone obok schematy pokazują rozkład wyprowadzeń wszystkich opisanych złączy. Wyprowadzenia są pokazane od zewnątrz w normalnym położeniu komputera, tzn. góra rysunku odpowiada górze złącza w komputerze stojącym na stole.

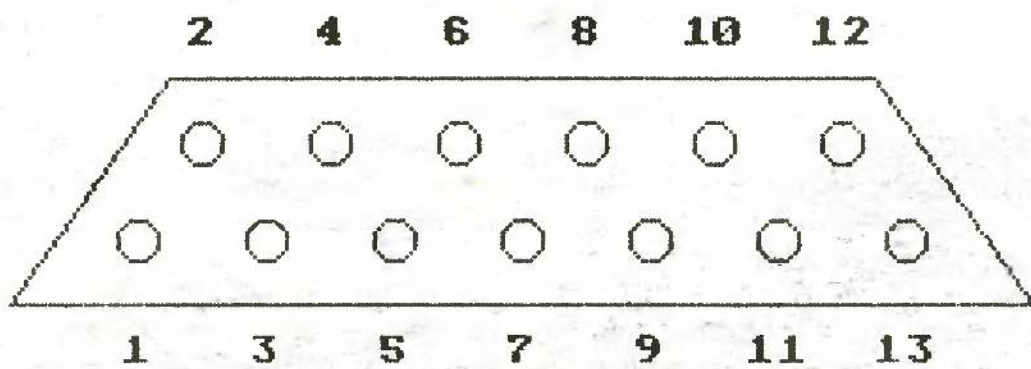
- | | |
|-----------|----------------|
| 1. ExtSel | A. nie używane |
| 2. Reset | B. IRq |
| 3. D1xx | C. Halt |
| 4. MPDiS | D. A13 |
| 5. Audio | E. A14 |
| 6. Ref | F. A15 |
| 7. +5V | G. masa |

Wojciech Zientara



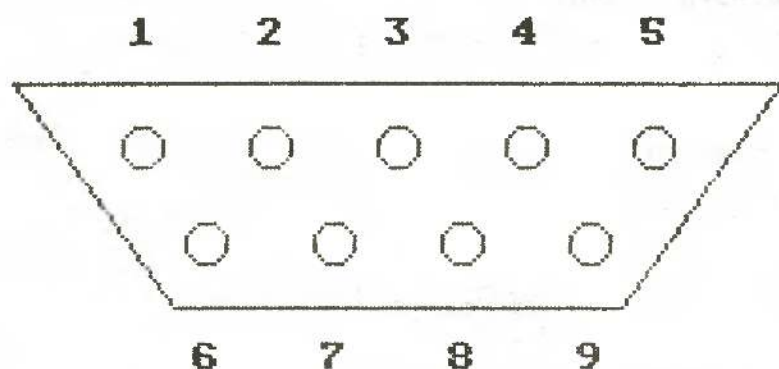
PARALLEL BUS

- | | | |
|-----------|--------------------|-----------------|
| 1. masa | 18. A14 | 35. IRQ |
| 2. ExtSel | 19. masa | 36. Ready |
| 3. A0 | 20. A15 | 37. nie używane |
| 4. A1 | 21. D0 | 38. CasInh |
| 5. A2 | 22. D1 | 39. nie używane |
| 6. A3 | 23. D2 | 40. Ref |
| 7. A4 | 24. D3 | 41. CAS |
| 8. A5 | 25. D4 | 42. masa |
| 9. A6 | 26. D5 | 43. MPDis |
| 10. masa | 27. D6 | 44. RAS |
| 11. A7 | 28. D7 | 45. masa |
| 12. A8 | 29. masa | 46. R/W |
| 13. A9 | 30. masa | 47. +5V |
| 14. A10 | 31. zegar (faza 2) | 48. +5V |
| 15. A11 | 32. masa | 49. Audio in |
| 16. A12 | 33. nie używane | 50. masa |
| 17. A13 | 34. Reset | |



SERIAL I/O PORT

- | | | |
|-----------------|------------------|----------------------|
| 1. Clock input | 6. masa | 11. Audio input |
| 2. Clock output | 7. Command | 12. +12V/nie używane |
| 3. Data input | 8. Motor control | 13. Interrupt |
| 4. masa | 9. Proceed | |
| 5. Data output | 10. Ready/+5V | |



CONTROLLER PORT

- | | | |
|---------------------|------------------------------|--------------------------|
| 1. joystick naprzód | 4. joystick w prawo | 7. +5V |
| 2. joystick wstecz | 5. suwak potencjometru 2 | 8. masa |
| 3. joystick w lewo | 6. przycisk / pióro świetlne | 9. suwak potencjometru 1 |



MICROPRINT

Nareszcie! Wiele mam do zarzucenia przedsiębiorstwu „Pewex”, lecz tym razem należy je pochwalić. Po dwóch latach sprzedawania komputerów Atari zaczęły się pojawiać różne dodatki do nich.

Wszyscy użytkownicy drukarek Atari 1029 zdają sobie sprawę, że jest to konstrukcja z początków komputeryzacji i jako taka znacznie odbiega swoimi możliwościami od innych drukarek, przewyższając je za to ceną.

Ze względu na niestandardowe złącze posiadane przez Atari zakup innej drukarki wymaga dokupienia interfejsu Centronics. Dotychczas było to możliwe na Zachodzie (za około 100 marek RFN) lub na giełdzie („tylko” 55 tys. zł.). Obecnie pojawiły się w sklepach „Pewexu” interfejsy „MicroPrint” po 39 dolarów (tyle samo co w USA!), produkowane przez znaną i solidną firmę Supra Corp.

„MicroPrint” jest interfejsem przyłączanym do złącza szeregowego Atari. Dokonuje on konwersji sygnału szeregowego na sygnał równoległy w standardzie Centronics. Interfejs jest wyposażony we wtyk Centronics typu

Amphenol, który jest stosowany w większości drukarek.

Oprócz zwykłej zamiany sygnału szeregowego na równoległy „MicroPrint” wymienia także kod znaku RETURN (155) na standardowy kod ASCII — CR (13). Oczywiście tylko wtedy, gdy oznacza on koniec linii. Podczas druku grafiki wymiana ta jest blokowana, aby nie wprowadzać błędów do rysunku.

Bezбłędne wykonywanie tych funkcji wymaga dokładnego śledzenia przesyłanej informacji. Zostało to zrealizowane za pomocą jednocukładowego mikroprocesora INTEL 8048 wyposażonego we własną pamięć ROM o pojemności 2 KB.

Dzięki pełnej zgodności ze standardem Centronics zapewniona jest możliwość wykorzystania bogatego oprogramowania użytkowego przeznaczonego dla drukarek mozaikowych, w tym m.in. „Print Shop”, „Design Master”, „MagniPrint” oraz serii programów firmy XLEnt.

Mam nadzieję, że pojawienie się interfejsów „MicroPrint” jest pierwszą jaskółką zwiastującą napływ innego sprzętu tego rodzaju. Przede wszystkim chciałbym zobaczyć w sklepach „Pewexu” modemy telefoniczne oraz interfejsy RS-232. Te ostatnie będą się na pewno cieszyły znacznym popytem, głównie wśród osób, które w domu używają Atari, a w pracy IBM PC.

Marek Zachar

Jest już nowy emulator IBM PC działający na każdym Atari ST. Firma IB Computers produkuje stacje dyskietek 5,25-calowych w dwu wersjach: 40 i 80 ścieżek. Obie mają identyczne rozmiary: 30 x 15 x 6 i wyposażone są w przewód łączący je z komputerem. Stacja dysków A staje się wtedy stacją B. Dołączony do nich program zmienia prędkość obrotową stacji B tak, aby była ona identyczna z prędkością IBM PC. Inny program narzędziowy inicjalizuje dyskietki w formacie MS-DOS. Dzięki niemu można wymieniać zbiory między Atari ST i klonami IBM PC. Ale IB Drive wykazuje w pełni swoje możliwości dopiero w połączeniu z programem PC-Ditto, produkowanym przez firmę Avant-Garde Systems (90 dol.). Przy jego użyciu Atari ST wreszcie wiernie imituje IBM PC, włączając w to kolorową grafikę i dźwięk, sterując wyjściami seryjnymi i równoległymi, stacjami 3,5-cala, 80 ścieżek i 5,25-cala 40 ścieżek jak również stacjami dysku sztywnego.

Jak większość emulatorów PC-Ditto jest wolniejszy o około 20—30% niż IBM PC, ale mimo to jego prędkość dla większości zastosowań jest wystarczająca. PC-Ditto musi być ładowany do RAM co ogranicza wielkość wolnej pamięci ST dostępnej dla użytkownika. I tak, w wypadku 1040 ST, zostaje 703 kB, tj. o 63 kB więcej niż w przypadku IBM PC normalnie wyposażonego. Modelowi 520 ST zostaje tylko 364 kB, co jest, niestety, niewystarczające, aby móc korzystać z programów na IBM PC. 1040 ST natomiast z powodzeniem odtwarza 1-2-3, Symphony, Enable, Wordperfect, Framework, dBase III, Turbo Pascal, Multimate, jak również większość pozostałych programów.

ce, ale — niestety — bardzo niebezpieczne. Nie tylko można stracić gwarancję producenta, ale i uszkodzić poważnie komputer, a dokładniej — jego obwód MMU i zasilacz poprzez dodawanie modułów RAM po 256 kB każdy, co sprawia, iż zasilacz się szybko nagrzewa.

Inne możliwe rozwiązanie polega na zastąpieniu modułów 256 kB przez moduły 1 MB, te same, które tworzą pamięć Mega ST. Zużywają one tyle samo energii przy mocy 4 razy większej. Niestety, moduły te posiadają zamiast 16, o dwie nóżki więcej. Poczujemy jednak posiadaczy 1040, który zdolny jest pilotować drukarkę laserową w trybie „tekst”, ponieważ jedna strona A4 — emulowana w Diablo — wymaga ok. 300 K. Ta sama strona grafiki wydrukowana przy użyciu drukarki laserowej wymaga jednak już zastosowania pamięci co najmniej 2 Mega.

Odnotujmy na koniec, iż model 2 Mega ST ma dokładnie 1,7 MB pamięci RAM, a model 4 Mega ST — 3,7 MB.

Mega ST. Jak do tej pory test kompatybilności między 1st Word, Super Base, Master Plan i większością programów graficznych wypadł pozytywnie.

Jak do tej pory budowa sieci lokalnej w oparciu o Atari ST była raczej nierealna. Przeszkodę tę usunął specjalny pakiet, A-Net, w którym zastosowano światłowody łączące Atari ST 1040 jako komputer główny i ST 520 jako terminale. W opisywanej sieci poszczególne stanowiska łączą się ze sobą za pośrednictwem portu Midi (szybkość transmisji 31250 bodów), przy czym pozostałe wyjścia są wolne. Kosztowne urządzenia peryferyjne, takie jak stacja dysku twardego czy drukarka laserowa są w pełni wykorzystane.

Sterowanie całością odbywa się za pośrednictwem rozkazów ze związanego menu komputera-matki. Maksymalna odległość między stanowiskami wynosi 40 m, a całkowita, teoretyczna liczba tych stanowisk wynosi 255. W praktyce jednak sieć lokalna tego typu jest stosowana w przypadku małej liczby stanowisk w celu uniknięcia zbyt dużego „tłoku” panującego na linii komputer-hard disk-drukarka.

A-Net jest sprzedawany w wersji 1 komputer-matka i 2 komputery końcowe wraz z 15-metrowej długości światłowodem za około 850 dolarów. Każde dodatkowe stanowisko kosztuje 280 dolarów.

Wzorując

się na znanym programie „Windows” dla IBM PC, firma UpGrade, wychodząc naprzeciw nowej serii Atari Mega St wypuściła na rynek wielozadaniowy program narzędziowy o bardzo ciekawym zastosowaniu. Zaproponowany switcher „ST Twist” (tu znowu rozgorzeje walka wśród purystów językowych, ale póki co, w języku polskim nie istnieje słowo opisujące ten rodzaj programu) pozwala na jednoczesne załadowanie aż... 6 różnych programów (maks. ilość okienek na jaką pozwala GEM), przy czym jedynym ograniczeniem jest pamięć komputera. Prezentacja graficzna programu jest bardzo udana: płynne przechodzenie od programu do programu, związanie ekranu w każde okienko i dowolne ich mieszanie. Po załadowaniu programu użytkownik określa wielkość pamięci potrzebnej do ich działania. Podprogramy mogą być uruchamiane myszką, albo na cały ekran albo rzutowane w okienko. Program może być zastosowany już na 1040 ST, ale „ST Twist” adresowany jest przede wszystkim do posiadaczy

D-TACK GROUNDED jest twórcą ultraszybkiego Basicu dla Atari ST, D-Basic. Kosztuje 40 dolarów, oferuje on wiele nowych funkcji nie zawartych w Basicu ST, jak: TIMER, ASSIGN, RENUM, CATALOG, CLS, INKEY. Jego szybkość jest zdumiewająca: 20 razy szybszy niż Basic ST i 3 razy szybszy niż Basic GFA.

Atari ostrzega wszystkich użytkowników-majsterkowiczów komputerów serii 520 i 1040 ST przed samowolnym powiększaniem pamięci RAM wspomnianych komputerów do — odpowiednio — 1 i 2 MB nawet jeśli jest to teoretycznie możliwe. Z chwilą kiedy na rynku pojawiła się tania drukarka laserowa Atari SM-804, takie rozbudowanie pamięci staje się kuszą-



bez większego trudu, oczy mimo nawet wielogodzinnej pracy specjalnie nie boją, ale w sumie nie jest to ideał. Coś ma ten kolor w sobie, że nie wzbudza sympatii.

Z hardwarem — jak wy to nazywacie — mam zresztą poważniejszy problem. Otóż macierzysta redakcja („Przegląd Tygodniowy”) zakupiła IBM-y wraz z programem składania i łamania gazety. Powoli zaczynamy już uruchamiać cały interes, a ja ciągle zamiast dostarczyć dyskietkę przywożę wydrukowany tekst, który dziewczyny mierzalnie przepisują do redakcyjnych komputerów.

Wnioski z tego są dwa. Po pierwsze, jeśli wykorzystuje się komputer profesjonalnie, prędzej czy później musi nadejść chwila, kiedy niezbędna okaże się jego kompatybilność ze standardem. Lepiej nawet poczekać pół roku albo rok, jeszcze trochę pooszczędzać i od razu kupić PC. A jeśli nie lepiej, to przynajmniej warto się zastanowić, co się bardziej opłaca.

Wniosek drugi jest natury praktycznej. Otóż uczeni w piśmie poinformowali mnie, że można napisać program i zmontować interfejs, który „przetłumaczy” moją dyskietkę na IBM. No to ja ogłaszam konkurs, kto mi tę pomoc naukową wykona. Rzecz jasna odpłatnie. Cena będzie umowna, czyli do uzgodnienia w toku owocnych negocjacji. Porozmawiamy sobie jak laik z fachmanem, czekam na zgłoszenia.

Do ogłoszenia konkursu jestem tym bardziej domotywowany, że nadal nie sposób kupić kasety z taśmą do drukarki 1029. Pewex w osobie dyr. Zacharskiego odesłał mnie do firmy „Karen”, co jest o tyle zabawne, że firma „Karen” jako serwisanci

„Atari” wielokrotnie namawiała Pewex do zakupu kaset, niestety bezskutecznie. Sama zaś handlować nimi nie może.

W miękim towarzyszu (wy to nazywacie software) odniosłem natomiast wiekopomny sukces. Podszedłem mianowicie po raz drugi do takiego oto zadania adaptacyjnego: Za pomocą programu „Synalc” obliczyć produkcję i zarobki około 50 pracowników. Każdy z nich wytwarza na akord 20 różnych wyrobów, z których z kolei każdy jest w innej cenie jednostkowej. No i po dwóch zarwanych nocach zrobiłem co trzeba korzystając wyłącznie z liczącej około 200 stron instrukcji obsługi programu. Nadzwyczaj wciągająca zabawa i rozumiem, że skoro mnie odłączyło od życia zaledwie praktyczne zastosowanie gotowego programu, to choroba komputerowa programistów ma tym cięższy przebieg.

Oczywiście nie jest to jeszcze elegancko zrobione, nie zmieściła mi się całość na jednym arkuszu, a czas obliczeń wydaje się zbyt długi, ale nic nie jest w stanie pozbawić mnie uczucia dumy. Praktyczna uwaga jest natomiast taka: nie należy się załamywać pierwszymi niepowodzeniami, natomiast uważnie czytać instrukcję i, po trzykroć, myśleć.

Zachęcony tym sukcesem zabrałem się za „Synfile”, który ma mi posłużyć do opracowania archiwum. Na początek zacząłem katalogować książki. Mam ich około 4 tys., z czego co najmniej połowa przydatna bywa w pracy. W rezultacie coraz częściej kupuję drugi raz tę samą pozycję, albo pisząc tekst zapominam, jakie źródła mam pod ręką.

Po wstępnych przymiarkach okazało się, że najważniejsze jest nawet nie wpisanie

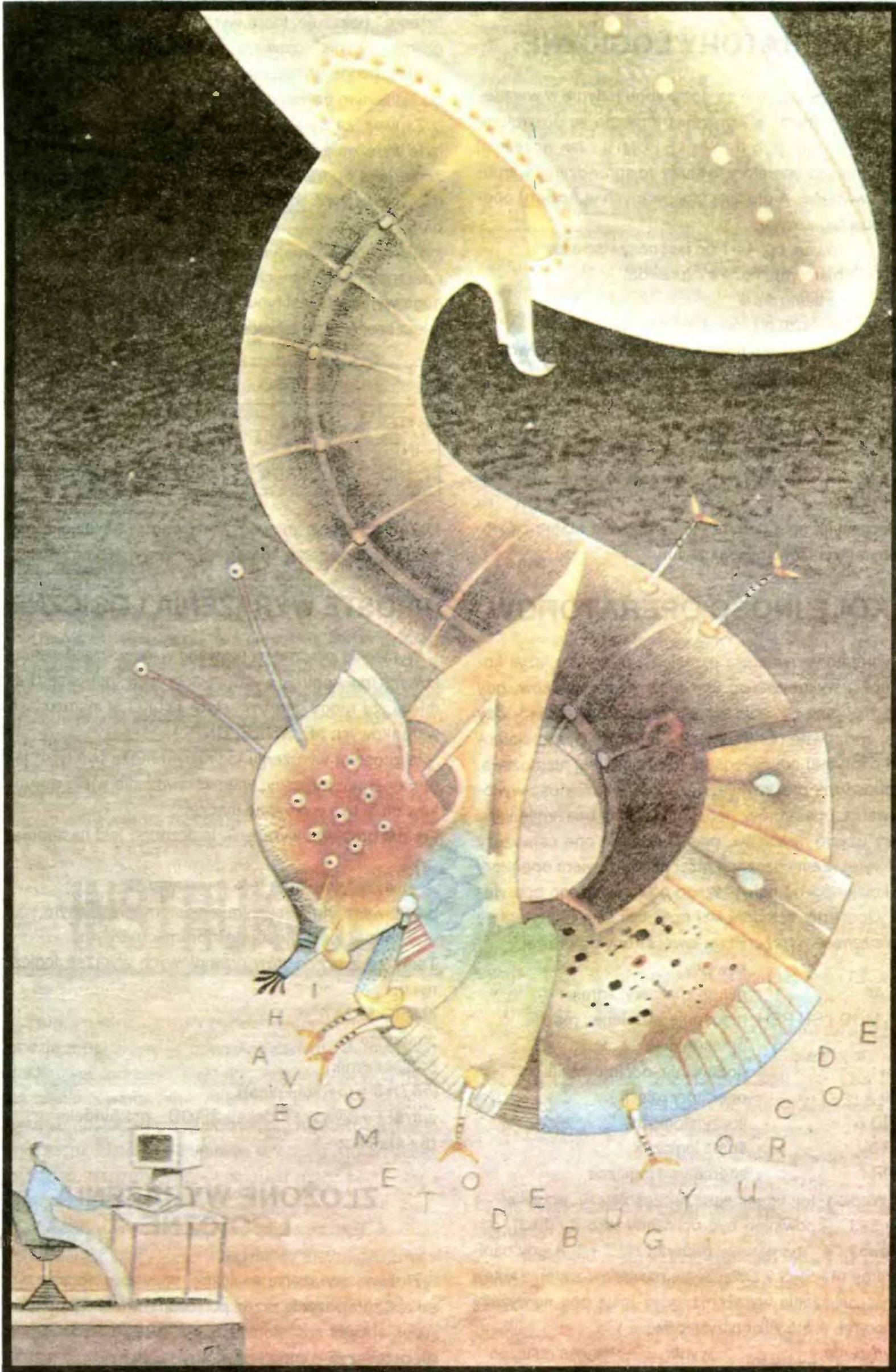
tego wszystkiego w postaci rekordów (po waszem), ale odpowiednie opracowanie słów kluczowych, żeby zawrzeć maksimum użytecznych informacji w minimalnej liczbie słów. Powoli posuwam się do przodu i pewnie za parę miesięcy odmelduję się z wynikiem pozytywnym. Po czym przyjdzie zabrać się do miesięczników i artykułów prasowych.

Oczywiście zamiast całej tej katorzniczej roboty powinienem mieć dostęp do odpowiednich bibliotek i banków danych, a nade wszystko sprawny telefon. Coś mi się jednak wydaje, że mój termin będzie konkurencyjny i szybciej sam sobie zrobię archiwum niż powołane do tego instytucje zorganizują ogólnodostępną sieć informacji komputerowej.

Trzecim wydarzeniem ostatnich miesięcy była wspomniana już wystawa „Komputer'88”. Przed rokiem szukałem najkorzystniejszej dla siebie oferty, tym razem spacerowałem się jako laik-posiadacz czyli na pełnym luzie. Może dlatego największe zaciekawienie wzbudziły sympatyczne dziewczyny w mini-spódniczkach, które pojawiły się chyba już w większości stoisk. Oczywiście jest to widomy znak nie tylko zwycięstwa frapującej mody, lecz nasilającej się konkurencji na rynku sprzętu i programów. Ponadto z prawdziwą satysfakcją obserwowałem młodych ludzi, którzy pracują w kilkuset już firmach komputerowych, i — jak się zdaje — nie muszą wyjeżdżać za granicę, żeby znaleźć satysfakcję w pracy i w miarę przyzwoite zarobki. Oto najlepsza odtrutka na nieciekawe czasy.

P.A.

Action!



Zamieszczony w poprzednim wydaniu specjalnym o Atari wstępny opis języka Action! spotkał się z ogromnym zainteresowaniem Czytelników. Będziemy więc kontynuować kurs programowania w tym języku, a równolegle w normalnych „Bajtkach” będą publikowane krótkie programy przykładowe.

(2)

WYRAŻENIA

Wyrażenia są konstrukcjami, które uzyskują wartość ze zmiennych, stałych i warunków za pomocą specjalnego zestawu operatorów. Na przykład „4+3” jest wyrażeniem, które jest równe „7” dopóki znakiem „+” oznaczamy dodawanie. Jeżeli jako operatora użyjemy znaku „*” (oznaczającego mnożenie), to wyrażenie będzie równe „12” ($4*3=12$). W Action! występują dwa typy wyrażen: arytmetyczne i logiczne. W powyższych przykładach zostały użyte wyrażenia arytmetyczne. Wyrażeniem logicznym jest takie, które w wyniku daje odpowiedź „prawda” lub „fałsz”. „5>=7” jest fałszywe, jeśli „>=” oznacza „większe lub równe”. Takie wyrażenia są używane w instrukcjach warunkowych. Instrukcją warunkową stosowaną codziennie jest na przykład zdanie: „Jeżeli jest czwarta lub później, to czas kończyć pracę”.

Przed szczegółowym omówieniem wyrażen musimy zdefiniować operatory, które mogą wystąpić w tych wyrażeniach.

OPERATORY

W Action! stosuje się trzy rodzaje operatorów:

1. operatory arytmetyczne
2. operatory bitowe
3. operatory logiczne

Jak sugerują nazwy, pierwsze i ostatnie operatory odnoszą się do odpowiednich typów wyrażen. Operatory bitowe służą również do operacji arytmetycznych, lecz wykonywanych na poszczególnych bitach liczb.

OPERATORY ARYTMETYCZNE

Operatorami arytmetycznymi są te, których używamy w matematyce, lecz nieco zmodyfikowane, aby umożliwić wprowadzenie ich z klawiatury komputera. Oto lista operatorów dostępnych w Action! wraz z opisem:

- minus znakowy, oznacza liczbę ujemną, np. -5
- * mnożenie, np. 4*3
- / dzielenie całkowite (bez reszty), np. 13/5 (jest to równe 2, gdyż reszta jest pomijana)
- MOD reszta z dzielenia całkowitego, np. 13 MOD 5 (jest to równe 3, gdyż $13/5 = 2$ z resztą 3)
- + dodawanie, np. 4+3
- odejmowanie, np. 4-3

Zwróć uwagę, że znak „=” nie jest operatorem arytmetycznym. Jest on używany tylko w wyrażeniach logicznych, instrukcjach przypisania i niektórych deklaracjach.

PROGRAMOWANIE

OPERATORY BITOWE

Operatory bitowe działają na liczbach w postaci dwójkowej (binarnej). Oznacza to, że możesz wykonywać operacje podobne do wykonywanych przez komputer (ponieważ on zawsze pracuje na liczbach dwójkowych). Poniższa lista zawiera wszystkie dostępne operatory bitowe:

- & bitowy iloczyn logiczny — AND
- % bitowa suma logiczna — OR
- ! bitowa alternatywa logiczna — EXCLUSIVE-OR
- XOR to samo co „!”
- LSH przesunięcie w lewo
- RSH przesunięcie w prawo
- @ adres

Pierwsze trzy operatory porównują liczby bit po bicie i zwracają wynik zależny od rodzaju operatora, jak pokazano niżej.

& porównuje dwa bity zwracając wartość według tabelki:

bit A	bit B	wynik
1	1	1
0	1	0
1	0	0
0	0	0

Przykład: 5 & 39 — 00000101 (równe dziesiętnie 5)
00100111 (równe dziesiętnie 39)
& -----
00000101 (wynikiem & jest 5)

% porównuje dwa bity zwracając wartość według tabelki:

bit A	bit B	wynik
1	1	1
0	1	1
1	0	1
0	0	0

Przykład: 5 % 39 -- 00000101 (5)
00100111 (39)
% -----
00100111 (wynikiem % jest 39)

! porównuje dwa bity zwracając wartość według tabelki:

bit A	bit B	wynik
1	1	0
0	1	1
1	0	1
0	0	0

Przykład: 5 ! 39 -- 00000101 (5)
00100111 (39)
! -----
00100010 (wynikiem ! jest 34)

Zarówno LSH jak i RSH przesuwają bity. Jeżeli operują na liczbach dwubajtowych (CARD i INT), to działają na oba bajty jednocześnie. W przypadku liczby typu INT może przy tym nastąpić zmiana znaku. Format użycia jest następujący:

<operand> <operator> <liczba przesunięć>
gdzie <operand> jest stałą lub zmienną liczbową,
<operator> LSH lub RSH
<liczba przesunięć> stała lub zmienna liczbowa określająca liczbę bitów do przesunięcia

Najlepiej działanie tych operatorów zilustrują przykłady:

```
(5)          00000101 (39) 00100111
(5 LSH 1 = 10) 000001010 (39 LSH 1 = 78) 01001110
(5 RSH 1 = 2) 00000010 (39 RSH 1 = 19) 00010011
operacja  MSB  LSB
-----  01010110 11001010 ($56CA)
LSH 1    10101101 10010100 ($56CA LSH 1 =
          $AD94)
RSH 1    00101011 01100101 ($56CA RSH 1 =
          $2B65)
LSH 2    01011011 00101000 ($56CA LSH 2 =
          $5B28)
RSH 2    00010101 10110010 ($56CA RSH 2 =
          $15B2)
```

Łatwo zauważyć, że LSH 1 jest równe mnożeniu przez dwa, RSH 1 — dzieleniu przez 2 (dla liczb dodatnich). W rzeczywistości ten sposób mnożenia i dzielenia przez 2 jest szybszy niż użycie „*2” i „/2”, ponieważ komputer nie musi już tłumaczyć wyrażenia na formę dwójkową.

Operator „@” daje w wyniku adres zmiennej stojącej po prawej stronie operatora. Nie może on być używany ze stałymi. „@ctr” zwraca adres, pod którym znajduje się w pamięci zmienna „ctr”. Operator ten jest bardzo użyteczny przy działaniach na wskaźnikach.

OPERATORY LOGICZNE

Operatory logiczne są dozwolone jedynie w wyrażeniach logicznych, a wyrażenia logiczne są dozwolone tylko w instrukcjach IF, WHILE i UNTIL. Jak wcześniej wspomniano operatory te służą do sprawdzania warunków równości. A oto lista stosowanych w **Action!** operatorów logicznych:

- = równe, np. 4=7 (to jest oczywiście fałsz)
- # nierówne, np. 4#7 (prawda)
- <> to samo co #
- > większe niż, np. 9>2 (prawda)
- >= większe lub równe, np. 5>=5 (to jest prawda)
- < mniejsze niż, np. 2<9 (prawda)
- <= mniejsze lub równe, np. 5<=5 (to jest prawda)
- AND iloczyn logiczny (I)
- OR suma logiczna (LUB)

Zarówno „#” jak i „<>” oznaczają to samo, więc możesz używać tego, który wolisz. „AND” i „OR” są specjalnymi operatorami i będą opisane razem ze złożonymi wyrażeniami logicznymi.

KOLEJNOŚĆ OPERATORÓW

Operatory wymagają określenia priorytetu, czyli kolejności wykonywania, aby uniknąć przypadków, gdy nie wiadomo jak obliczyć wyrażenie, np. 4+5*3. Czy jest ono równe (4+5)*3, czy 4+(5*3)? Bez znajomości priorytetu odpowiedź na to pytanie jest niemożliwa.

Action! ma precyzyjnie wyznaczoną kolejność wykonywania operatorów, lecz może być ona zmieniana przy użyciu nawiasów, ponieważ mają one najwyższy priorytet. Zamieszczona niżej tabela zawiera operatory w kolejności od najwyższego do najniższego priorytetu. Operatory w jednej linii mają taki sam priorytet i są obliczane w wyrażeniu od lewej strony do prawej.

- () nawiasy
- @ minus znakowy, adres
- * /MOD LSH RSH mnożenie, dzielenie, reszta, przesunięcie
- + - dodawanie, odejmowanie
- =# <> >= <= operatory relacji
- AND & iloczyn logiczny
- OR % suma logiczna
- XOR ! alternatywa logiczna

Według tej tabeli nasz wcześniejszy przykład — 4+5*3 — powinien być obliczany jako 4+(5*3), ponieważ „*” ma wyższy priorytet niż „+”. A gdy napiszemy (4+5)*3? Dołączenie nawiasów zmienia kolejność obliczania wyrażenia, gdyż mają one najwyższy priorytet. A oto kilka przykładów:

wyrażenie	wynik	kolejność obliczeń
4/2*3	6	/,*
43 MOD 7*2 + 19	21	MOD,*,+
—((4+2)/3*	-2	+,/,—

WYRAŻENIA ARYTMETYCZNE

Dowolne wyrażenie arytmetyczne może składać się ze stałych i zmiennych liczbowych oraz operatorów porządkujących sposób uzyskania wyniku. Format wyrażenia jest następujący:

<operand> <operator> <operand>
gdzie operand jest stałą liczbową, zmienną liczbową,

wywołaniem funkcji (FUNC) lub innym wyrażeniem arytmetycznym. Pierwsze trzy możliwości są jasne, lecz czwarta może sprawić kłopot. Prosty przykład wyjaśni o co chodzi:

krok	wyrażenie	wynik	uproszczone wyrażenie
0	3*(4+(22/7)*2)	--	---
1	22/7	3	3*(4+3*2)
2	(22/7)*2	6	3*(4+6)
3	4+(22/7)*2	10	3*10
4	3*(4+(22/7)*2)	30	30

„krok” jest numerem wykonywanego obliczenia; „wyrażenie” pokazuje, które wyrażenie jest aktualnie obliczane; „wynik” zawiera wynik tego obliczenia; a „uproszczone wyrażenie” pokazuje wygląd wyrażenia po aktualnym obliczeniu.

Zauważ, że wyrażenia w krokach od 2 do 4 zawierają inne wyrażenie jako jeden z operandów, lecz to „wyrażenie jako operand” jest już obliczone i można w to miejsce wstawić liczbę jak w „uproszczonym wyrażeniu”.

Wyrażenia arytmetyczne w **Action!** mogą mieć jako operandy dane różnych typów. Wynik takiego mieszanego wyrażenia jest typu nadrzędnego, jak widać w zamieszczonej niżej tabelce:

	BYTE	INT	CARD
BYTE	BYTE	INT	CARD
INT	INT	INT	CARD
CARD	CARD	CARD	CARD

UWAGA: użycie minusa znakowego (—) daje wynik typu INT, a operator adresu @ daje wynik typu CARD.

PROSTE WYRAŻENIA LOGICZNE

Wyrażenia logiczne są używane w instrukcjach warunkowych do kontroli wykonywania tych instrukcji. Pamiętaj, że mogą one być użyte TYLKO w instrukcjach warunkowych (IF, WHILE, UNTIL).

W prostym wyrażeniu logicznym może być tylko jeden operator logiczny, więc sprawdzanie wielu warunków musi być wykonane inaczej.

Format prostego wyrażenia logicznego jest następujący:

<wyr arytm> <oper log> <wyr arytm>
gdzie <wyr arytm> jest wyrażeniem arytmetycznym
<oper log> jest operatorem logicznym
Teraz kilka przykładów prawidłowych wyrażeń logicznych:
@kot*=\$22A7
zmienna<wyr arytm>
5932#licznik
(5&7)*8=(3*(kot+pies))
adr/\$FF+(@wsk+indeks)*\$FU3D—wsk&indeks
(5+4)*9*licz-1

ZŁOŻONE WYRAŻENIA LOGICZNE

Złożone wyrażenia logiczne pozwalają rozszerzyć zakres sprawdzania przez połączenie wielu warunków. Jeżeli chcesz coś zrobić tylko w lipcową niedzielę, to jak przekazać komputerowi, aby sprawdził, czy jest lipiec, a potem, czy jest niedziela. W **Action!** służą do tego operatory logiczne AND i OR. Kompilator traktuje je jako specjalne operatory do sprawdzania warunków złożonych z prostych wyrażeń logicznych. Ich format jest następujący:

<wyr log> <oper sp> <wyr log> 1: <oper sp> <wyr log>:1
gdzie <wyr log> jest prostym wyrażeniem logicznym
<oper sp> jest specjalnym operatorem logicznym (AND lub OR)

UWAGA: ten format nie ma wyjątków. Jeżeli spróbujesz napisać inaczej, to uzyskasz błąd kompilacji „Bad Expression” (złe wyrażenie).

Poniższa tablica prawdy pokazuje, jak każdy z tych operatorów działa w różnych sytuacjach. „wyr 1” i

wyrażenia		wyniki	
wyr 1	wyr 2	AND	OR
prawda	prawda	prawda	prawda
prawda	fałsz	fałsz	prawda
fałsz	prawda	fałsz	prawda
fałsz	fałsz	fałsz	fałsz

„wyr 2” są prostymi wyrażeniami logicznymi z obu stron operatora specjalnego; „prawda” i „fałsz” to możliwe wyniki wyrażenia logicznego.

UWAGA: można użyć nawiasów do zmiany kolejności obliczania wyrażenia logicznego. Jeśli tego nie zrobisz, to wyrażenie będzie obliczane według priorytetów operatorów.

I znów kilka przykładów prawidłowych złożonych wyrażen logicznych:

```
kot<=5 AND pies<13
(@wsk+7) * 3 # $60FF AND @wsk <= $1FFF
x!$F0<0 OR pies>=100
(8&kot)<10 OR @wsk<$OD
kot<0 AND (pies<400 OR pies<-400)
wsk=$D456 OR wsk=$E000 OR wsk=$600
```

Teraz trochę dziwna sytuacja:

```
$F0 AND $OF
```

jest to fałszywe, ponieważ „AND” zostanie potraktowane jako operator bitowy w wyrażeniu arytmetycznym, podczas gdy

```
$F0<0 AND $OF<0
```

jest prawdziwe, ponieważ „AND” rozdziela dwa proste wyrażenia logiczne, a więc jest traktowany jako specjalny operator w wyrażeniu złożonym.

UWAGA: operatory w parach „AND” – „&” i „OR” – „%” są sobie równoważne i kompilator zależnie od kontekstu traktuje je jako bitowe lub logiczne. Pożądane jest jednak przyzwyczajenie się do używania tu konwencji stosowania „&” i „%” jako operatorów bitowych, a „AND” i „OR” jako logicznych, ze względu na zgodność z ewentualnymi wersjami rozwojowymi systemu **Action!**.

INSTRUKCJE

Program komputerowy będzie nieużyteczny, jeśli nie będzie mógł aktywnie operować danymi. Potrafisz już deklarować zmienne, stałe itd., lecz nie jest to sposób na manipulowanie nimi. Aktywną częścią każdego języka komputerowego są instrukcje i **Action!** nie jest tu wyjątkiem. Instrukcje tłumaczą działania, które chcesz wykonać, na formę, którą komputer może zrozumieć i prawidłowo wykonać.

W **Action!** występują dwa rodzaje instrukcji: instrukcje proste i instrukcje strukturalne. Instrukcje proste nie zawierają w sobie innych instrukcji, zaś instrukcje strukturalne są złożone z innych instrukcji (zarówno prostych, jak i strukturalnych) umieszczonych w pewnej określonej kolejności. Instrukcje strukturalne można jeszcze podzielić na instrukcje warunkowe i instrukcje pętli. Każda z tych kategorii będzie omówiona osobno.

INSTRUKCJE PROSTE

Instrukcje proste są to te, które wykonują tylko jedną czynność. Są one podstawowymi składnikami programu, ponieważ każde działanie komputera jest przez nie wykonywane. W **Action!** są dwa rodzaje instrukcji prostych: instrukcje przypisania (w tym również wywołania funkcji) oraz wywołania procedur. Także dwa słowa

kluczowe (EXIT i RETURN) są instrukcjami prostymi, lecz ze względu na specyficzne zastosowanie będą omówione później.

INSTRUKCJA PRZYPISANIA

Instrukcja przypisania służy do nadania wartości zmiennej. Jej najczęściej używanym formatem jest:

<zmienna> = <wyrażenie arytmetyczne>
<zmienna> może być podstawowego typu danych albo też może być tablicą, wskaźnikiem lub określeniem rekordu.

<wyrażenie> MUSI być arytmetyczne. Jeżeli spróbujesz użyć wyrażenia logicznego, to wystąpi błąd, ponieważ kompilator **Action!** nie podstawia wartości liczbowej po obliczeniu wyrażenia logicznego.

Operatorem przypisania jest „=”. Wskazuje on komputerowi, że chcesz przypisać nową wartość podanej zmiennej. Nie należy mylić go z logicznym „=”. Pomimo że jest to ten sam znak, kompilator odczytuje go różnie, zależnie od kontekstu.

Poniższe przykłady ilustrują instrukcję przypisania. Zwróć uwagę na deklaracje zmiennych poprzedzające przykłady. Są one potrzebne, gdyż niektóre przykłady pokazują mieszanie typów, tzn. zmienna i przypisywana jej wartość są różnego typu.

```
BYTE b1,b2,b3,b4
```

```
INT i1
```

```
CARD c1
```

```
b3='D
```

umieszcza kod ATASCII znaku „D” w bajcie zarezerwowanym dla zmiennej b3.

```
b4=$44
```

umieszcza liczbę szesnastkową \$44 w bajcie zarezerwowanym dla zmiennej b4 (\$44 jest kodem ATASCII znaku „D”, więc zmienne b3 i b4 zawierają teraz to samo).

```
b1=b4+16
```

dodaje 16 do wartości liczbowej zmiennej b4 i wynik umieszcza w bajcie zarezerwowanym dla zmiennej b1.

```
c1=23439-$07D8
```

umieszcza wartość 21431 (\$53B7) w dwóch bajtach zarezerwowanych dla c1

```
i1=c1*(-1)
```

umieszcza wartość — 21431 (\$AC49) w dwóch bajtach zarezerwowanych dla i1.

```
b2=i1
```

umieszcza wartość \$49 (73) w bajcie zarezerwowanym dla b2. Zauważ, że komputer uwzględnia tylko LSB (MSB z i1 jest \$AC, LSB — \$49).

```
b2=b2+1
```

dodaje 1 do aktualnej wartości b2 i umieszcza sumę ponownie w b2. Zmienna b2 zawiera teraz \$4A (74).

Zwróć uwagę na ostatni przykład. Jego format jest następujący:

```
<zmienna> = <zmienna> <operator> <operand>
```

Ponieważ ten format instrukcji przypisania jest bardzo często używany przez programistów, to w **Action!** zastosowano do tego celu format skrócony:

```
<zmienna> == <operator> <operand>
```

Operator musi być arytmetyczny lub bitowy, a operand musi być wyrażeniem arytmetycznym. Oto kilka przykładów skróconego formatu:

```
b2==+1      jest równoważne   b2=b2+1
b2==-b1     jest równoważne   b2=b2-b1
b2==&$OF    jest równoważne   b2=b2 & $OF
b2==LSH
(5+3)       jest równoważne   b2=b2 LSH
(5+3)
```

Skrócony format zmniejsza ilość błędów popełnianych przy pisaniu programu, a także daje po skompilowaniu prostszy i krótszy kod wynikowy.

Jeżeli dostępne byłyby tylko instrukcje proste, to czynności wykonywane przez komputer byłyby bardzo ograniczone:

Jedynym sposobem powtórzenia jakiejś grupy instrukcji byłoby powtórzenie ich w programie odpowiednią ilość razy w prawidłowej kolejności. Jeśli chciałbyś powtórzyć grupę dziesięciu instrukcji 10 razy, to musiałbyś wpisać 100 instrukcji.

Nie można by wykonywać grupy instrukcji warunkowo, to znaczy wykonywać je tylko wtedy, gdy spełniony jest jakiś warunek.

Celem instrukcji strukturalnych jest usunięcie tych i innych problemów. Instrukcje strukturalne dzielą się na dwie kategorie: instrukcje warunkowe i instrukcje pętli.

WYKONANIE WARUNKOWE

Wykonanie warunkowe pozwala na sprawdzenie wartości wyrażenia i wykonanie różnych instrukcji zależnie od rezultatu tego sprawdzenia. Ponieważ wyrażenie kontroluje warunkowe wykonanie, to nazywane jest ono wyrażeniem warunkowym.

Wykonanie warunkowe umożliwiają trzy instrukcje **Action!:** IF, WHILE, UNTIL. Dwie ostatnie są równocześnie instrukcjami pętli.

WYRAŻENIA WARUNKOWE

Ponieważ wyrażenie warunkowe jest wykorzystywane jako test, to może mieć tylko dwie wartości — prawda lub fałsz. Nie oznacza to jednak, że wyrażenie warunkowe jest jakimś nowym typem wyrażenia. W rzeczywistości wyrażenie warunkowe jest wyrażeniem logicznym lub arytmetycznym. Tylko interpretacja jest nieco inna. Poniższa tabelka pokazuje interpretację warunkową w zależności od typu wyrażenia:

typ wyrażenia	wynik normalny	wynik warunkowy
arytmetyczne	zero (0) nie zero	fałsz prawda
logiczne	fałsz prawda	fałsz prawda

INSTRUKCJA IF

Instrukcja IF (jeżeli) w **Action!** jest bardzo podobna do słowa „jeżeli” w języku polskim. Na przykład:

„Jeżeli mam 90 złotych lub więcej, to kupię hamburgera.”

W **Action!** może to być wyrażone następująco:

```
BYTE pieniadze,
   hamburger=[90],
   paluszki=[80],
   hotdog=[60],
   ciastko=[20]
```

```
IF pieniadze>=90 THEN
   zakup(hamburger,pieniadze)
FI
```

UWAGA: zakup (hamburger, pieniądze) jest wywołaniem procedury i będzie dokładnie opisane później.

Z podanego przykładu widać, że podstawowym formatem instrukcji IF jest:

```
IF <wyrażenie warunkowe> THEN
   <instrukcja(e)>
FI
```

FI jest odwrotnie napisanym IF i jest słowem kluczowym, które wskazuje kompilatorowi koniec instrukcji IF. Ponieważ IF może dotyczyć większej liczby instrukcji, to konieczne jest zakończenie tych instrukcji przez FI. Bez tego słowa kompilator nie będzie wiedział, ile następujących po THEN instrukcji należy do struktury IF.

Oprócz podanego formatu podstawowego instrukcja IF ma jeszcze dwa warianty: ELSE i ELSEIF. Po polsku można to wyrazić następująco:

„Jeżeli mam 90 złotych lub więcej, to kupię hamburgera, w przeciwnym razie kupię paluszki.”

Odpowiednikiem tego zdania w **Action!** jest:

```
IF pieniadze>=90 THEN
  zakup (hamburger,pieniadze)
ELSE
  zakup (paluszki,pieniadze)
FI
```

ELSEIF jest nieco odmienne:

„Jeżeli mam 90 złotych lub więcej, to kupię hamburgera, w przeciwnym razie jeżeli mam mniej niż 90 a i więcej niż 80, to kupię paluszki, w przeciwnym razie jeżeli mam mniej niż 80 i więcej niż 60, to kupię hotdoga, w przeciwnym razie kupię ciastko.”

I zapis w **Action!**:

```
IF pieniadze>=90 THEN
  zakup (hamburger,pieniadze)
ELSEIF pieniadze>=80 THEN
  zakup (paluszki,pieniadze)
ELSEIF pieniadze>=60 THEN
  zakup (hotdog,pieniadze)
ELSE
  zakup (ciastko,pieniadze)
FI
```

Zauważ, że nie trzeba sprawdzać „pieniadze=80 AND pieniadze<90”. Jest tak, ponieważ komputer wykonuje instrukcje kolejno z góry na dół. Jeżeli jeden przypadek jest prawdziwy, to odpowiednia grupa instrukcji jest wykonywana, a cała reszta instrukcji IF (razem ze wszystkimi następnymi ELSE i ELSEIF) jest pomijana. Tak więc gdy komputer doszedł do „pieniadze=80”, to wiadomo, że masz mniej niż 90 złotych, gdyż poprzednio musiał być sprawdzony warunek „pieniadze=90” i wynik był fałszem.

Wariant ELSEIF jest bardzo użyteczny, gdy chcesz kilkakrotnie sprawdzić zmienną, a dla różnych jej wartości są przewidziane różne instrukcje do wykonania.

INSTRUKCJA PUSTA

Instrukcja pusta nie robi nic. Po pokazaniu instrukcji wykonujących różne czynności, po wskazaniu konieczności, aby instrukcje coś wykonywały, teraz instrukcja, która nic nie robi? Są jednak pewne zastosowania dla takich instrukcji: pętle czasowe i przypadki ELSEIF.

Ponieważ nie opisywaliśmy jeszcze pętli, to powiedzmy po prostu, że pętle czasowe służą do opóźnienia wykonywania programu (np. gdy potrzebna jest przerwa między wyświetlaniem kolejnych linii na ekranie).

Użycie instrukcji pustej w przypadku ELSEIF zilustruje natomiast następujący przykład:

Scenariusz: piszesz program, który pozwala magazynerowi odszukać informacje o towarach przy użyciu ustalonych poleceń. Dostępne są polecenia: PRZYJĘCIE, WYDANIE, SZUKANIE i KONIEC, lecz polecenie SZUKANIE nie jest jeszcze zaimplementowane. Polecenie jest rozpoznawane po pierwszej literze, a więc należy porównać wprowadzony znak z P, W, S i K. Jednak SZUKANIE nie jest zrobione, co więc czynić, gdy naciśnięty został klawisz „S”? Po prostu nic, aż do czasu ukończenia operacji SZUKANIE. Odpowiedni fragment programu może wyglądać tak:

```
IF znak='P THEN
  zrobprzyjecie()
ELSEIF znak='W THEN
  zrobwydanie()
ELSEIF znak='S THEN
  ;**** tu jest instrukcja pusta
ELSEIF znak='K THEN
  zrobkoniec()
ELSE
  pomyłka() ;**** zle polecenie
FI
```

Wszystkie „zrób...” są procedurami, które wykonują podane polecenia. Jeżeli spojrzysz na przypadek „znak= 'S'”, to zobaczysz, że nic tam nie jest robione. To jest właśnie instrukcja pusta. Gdy SZUKANIE jest

gotowe, wystarczy tylko zamiast instrukcji pustej wstawić wywołanie procedury „zróbszukanie ()”.

PĘTLE

Pętle są stosowane do powtarzania pewnych instrukcji. Jeżeli w jakimś celu chcesz wypełnić ekran gwiazdkami (*), to możesz wyświetlić każdą gwiazdkę oddzielną instrukcją lub możesz do tego użyć pętli. Wystarczy tylko wskazać, ile razy pętla ma wyświetlić pojedynczą gwiazdkę i to wszystko (oczywiście jeśli użyjesz prawidłowego formatu instrukcji).

Są dwa sposoby wskazania, ile razy pętla ma coś wykonać. Można podać konkretną liczbę lub użyć wyrażenia warunkowego i wykonywać pętlę zależnie od wartości tego wyrażenia. Pierwszy sposób jest stosowany w instrukcji FOR, a drugi w instrukcjach WHILE i UNTIL.

Co będzie, gdy nie wskażemy, ile razy ma być wykonana pętla? Co zrobić, gdy wyrażenie warunkowe nigdy nie osiągnie wartości kończącej pętlę? Taki przypadek jest zwany „pętlą bez końca”. Jest tylko jeden sposób przerwania pętli bez końca — naciśnięcie klawisza «RESET».

Action! udostępnia pętle w następujący sposób: Podstawowa pętla, gdy jest użyta samodzielnie, to nie ma końca. Oprócz tego są instrukcje kontrolujące pętlę (FOR, WHILE, UNTIL), które ograniczają liczbę wykonań pętli. Zrobimy w ten sam sposób: najpierw poznamy strukturę pętli podstawowej, a następnie instrukcje kontrolujące pętlę.

INSTRUKCJE DO i OD

„DO” i „OD” są używane do oznaczania odpowiednio początku i końca pętli podstawowej. Wszystko między nimi jest częścią tej pętli. Jak wcześniej powiedziano, sama pętla (tzn. bez żadnej instrukcji kontroli pętli) jest pętlą bez końca. Poniższy przykład ilustruje pętlę DO—OD. Nie przejmuj się instrukcjami „PROC” i „RETURN”. Są one tu dołączone, aby program mógł być prawidłowo skompilowany i uruchomiony, a opisane będą później.

PROC razydwa()

```
CARD i=[0],j

DO          ;początek petli DO-OD
i==+1      ;dodanie 1 do 'i'
j=i*2      ;j rowne i*2
PrintC(i)  ;zob. nizej
Print(" razy 2 jest rowne ")
PrintCE(j)
OD          ;koniec petli DO-OD
RETURN
```

UWAGA: Słowa pisane dużymi i małymi literami (PrintC, Print, PrintCE), które znajdują się w przykładzie są procedurami bibliotecznymi **Action!**. Muszą one być użyte dla lepszego pokazania przykładu, a opis biblioteki **Action!** znajdzie się na końcu naszego kursu.

Po skompilowaniu i uruchomieniu programu zobaczysz coś takiego:

```
1 razy 2 jest rowne 2
2 razy 2 jest rowne 4
3 razy 2 jest rowne 6
4 razy 2 jest rowne 8
5 razy 2 jest rowne 10
6 razy 2 jest rowne 12
7 razy 2 jest rowne 14
8 razy 2 jest rowne 16
...
...
```

Punkty na końcu wskazują, że wyświetlanie będzie trwało dalej, aż do naciśnięcia «RESET». Sama w sobie pętla DO—OD może być mniej lub bardziej użyteczna, lecz dopiero w połączeniu z instrukcjami kontrolującymi

mi FOR, WHILE i UNTIL staje się jedną z najbardziej pożytecznych instrukcji.

UWAGA: naciśnięcie klawisza «BREAK» również przerywa ten program, ponieważ pętla używa operacji wejścia/wyjścia, a «BREAK» działa tylko wtedy (więcej informacji znajdziesz w opisie kompilatora).

Zawsze gdy w opisie formatu instrukcji kontroli pętli zobaczysz „pętla DO—OD”, pamiętaj, że oznacza to całą pętlę złożoną z różnych instrukcji ograniczonych instrukcjami DO i OD.

INSTRUKCJA EXIT

Instrukcja EXIT służy do opuszczenia dowolnej pętli. Powoduje ona przejście wykonywania programu do instrukcji następującej po „OD”. Oto przykład

PROC razydwa()

```
CARD i=[0],j

DO          ;początek petli DO-OD
i==+1      ;dodanie 1 do 'i'
j=i*2      ;j rowne i*2
PrintC(i)
Print(" razy 2 jest rowne ")
EXIT       ;tu jest wyjscie
PrintCE(j)
OD          ;koniec petli DO-OD
;*** stad jest kontynuowane
;*** wykonywanie po 'EXIT'
PrintE("Koniec tablicy")
RETURN
```

i jego rezultat:

```
1 razy 2 jest rowne Koniec tablicy
```

Jak widać, instrukcja „PrintCE (j)” nigdy nie będzie wykonana. Instrukcja EXIT wymusza skok do instrukcji „PrintE („Koniec tablicy)”. EXIT jest rzadko używana samodzielnie (jak wyżej), lecz staje się bardzo wygodna w połączeniu z instrukcją IF (daje np. warunkowe wyjście z pętli). Taki sposób jej wykorzystania pokazuje następny przykład.

PROC razydwa()

```
CARD i=[0],j

DO          ;początek petli DO-OD
IF i=15 THEN
  EXIT     ;EXIT w IF
FI
i==+1
j=i*2
PrintC(i)
Print(" razy 2 jest rowne ")
PrintCE(j)
OD          ;koniec petli DO-OD
;*** stad jest kontynuowane
;*** wykonywanie, gdy i=15
PrintE("Koniec tablicy")
RETURN
```

Wynik:

```
1 razy 2 jest rowne 2
2 razy 2 jest rowne 4
3 razy 2 jest rowne 6
4 razy 2 jest rowne 8
5 razy 2 jest rowne 10
6 razy 2 jest rowne 12
7 razy 2 jest rowne 14
8 razy 2 jest rowne 16
9 razy 2 jest rowne 18
10 razy 2 jest rowne 20
11 razy 2 jest rowne 22
12 razy 2 jest rowne 24
13 razy 2 jest rowne 26
14 razy 2 jest rowne 28
15 razy 2 jest rowne 30
Koniec tablicy
```

Zamienia to pętlę bez końca w pętlę z wyjściem. EXIT może kontrolować wykonywanie pętli, lecz nie oznacza to strukturalnej instrukcji kontroli pętli, gdyż działa ona tak tylko w połączeniu z IF.

Wojciech Zientara

WARGAME CONSTRUCTION SET XL/XE



Po wielu różnych grach strategicznych obejmujących historię wojen (głównie na zachodzie Europy) od średniowiecza aż po czasy współczesne sympatycy tego gatunku rozrywki doczekali się gry uniwersalnej.

Wydany przez SSI „Zestaw konstrukcyjny gier wojennych” zawiera na dwóch dyskiecie programy, które powinny zadowolić każdego, nawet najwybredniejszego użytkownika. W skład zestawu wchodzi „Edytor”, „Gra” i 8 gotowych scenariuszy.

Edytor pozwala na zaprojektowanie dowolnej gry strategicznej. Dzięki czterem poziomom szczegółowości można zaplanować grę praktycznie w dowolnej skali — od drobnej potyczki po całą kampanię wojenną. Szeroki wybór symboli jednostek i definiowanie ich parametrów umożliwi ułożenie scenariusza dla dowol-

nego okresu historycznego — od starożytności do dalekiej przyszłości, a także scenariusza fantastycznego. Opis projektowania scenariusza w instrukcji programu pokazuje właśnie sposób ułożenia gry typu „fantasy”.

Posługiwanie się edytorem jest stosunkowo proste. Wszystkie funkcje są wybierane joystickiem z menu. Również większa część prac projektowych jest wykonywana przy użyciu joysticka. Znacznym ułatwieniem są funkcje zwielokrotniania operacji, np. powielanie zaprojektowanych jednostek, czy wypełnianie symbolami dużych obszarów leśnych. Pomimo odróżniania gier dla jednego lub dwóch graczy, edytor umożliwia takie zaprojektowanie gry, aby nadawała się do obu wariantów (choć nie jest to polecane).

Ułożoną przez siebie grę można zapisać na dyskiet-

ce do późniejszego wykorzystania lub wprowadzenia zmian. Istnieje także możliwość przeredagowania gotowych scenariuszy dołączonych do programu.

Po skończonej pracy czas na rozrywkę. Do tego właśnie służy gra. Zawiera ona dwa warianty różniące się ilością faz w każdej rundzie i ich kolejnością. Pierwszy wariant — uruchamiany klawiszem START — jest przeznaczony dla jednego gracza. Drugi — uruchamiany przez OPTION — pozwala na rozegranie starcia z partnerem przy wykorzystaniu komputera jako inteligentnej planszy. Oczywiście nie wszystkie gry można rozgrywać w obu wariantach, przeważnie rodzaj gry jest podawany przy wyborze scenariusza.

Rozgrywaną grę można zapisać na końcu każdej rundy. Pozwala to na wypróbowanie różnych wariantów akcji, a także na jej przeredagowanie za pomocą edytora.

Jeśli nie mamy ochoty na układanie własnych scenariuszy, program oferuje nam osiem gotowych gier przedstawiających szeroki wachlarz odmian. Są to następujące gry:

1. Rommel's 88 — fragment walk we Francji podczas II wojny światowej, skala taktyczna, dla jednego gracza.
2. To cross a river — forsowanie rzeki łodziami i śmigłowcami, skala taktyczna, dla dwóch graczy.
3. Abuse on the Meuse — kontrofensywa w Ardenach, skala strategiczna, dla dwóch graczy.
4. Return to Beta 4 — walka o asteroidę w XXX w., skala taktyczna, dla jednego gracza.
5. Delta Force Rescue — walka z terrorystami, skala taktyczna, dla jednego gracza.
6. Fulda Gap — bitwa o przełęcz Fulda, skala operacyjna, dla jednego gracza.
7. First Bull Run — bitwa podczas wojny secesyjnej, skala operacyjna, dla dwóch graczy.
8. Castle Siege — oblężenie zamku w XII wieku, skala taktyczna, dla jednego gracza.

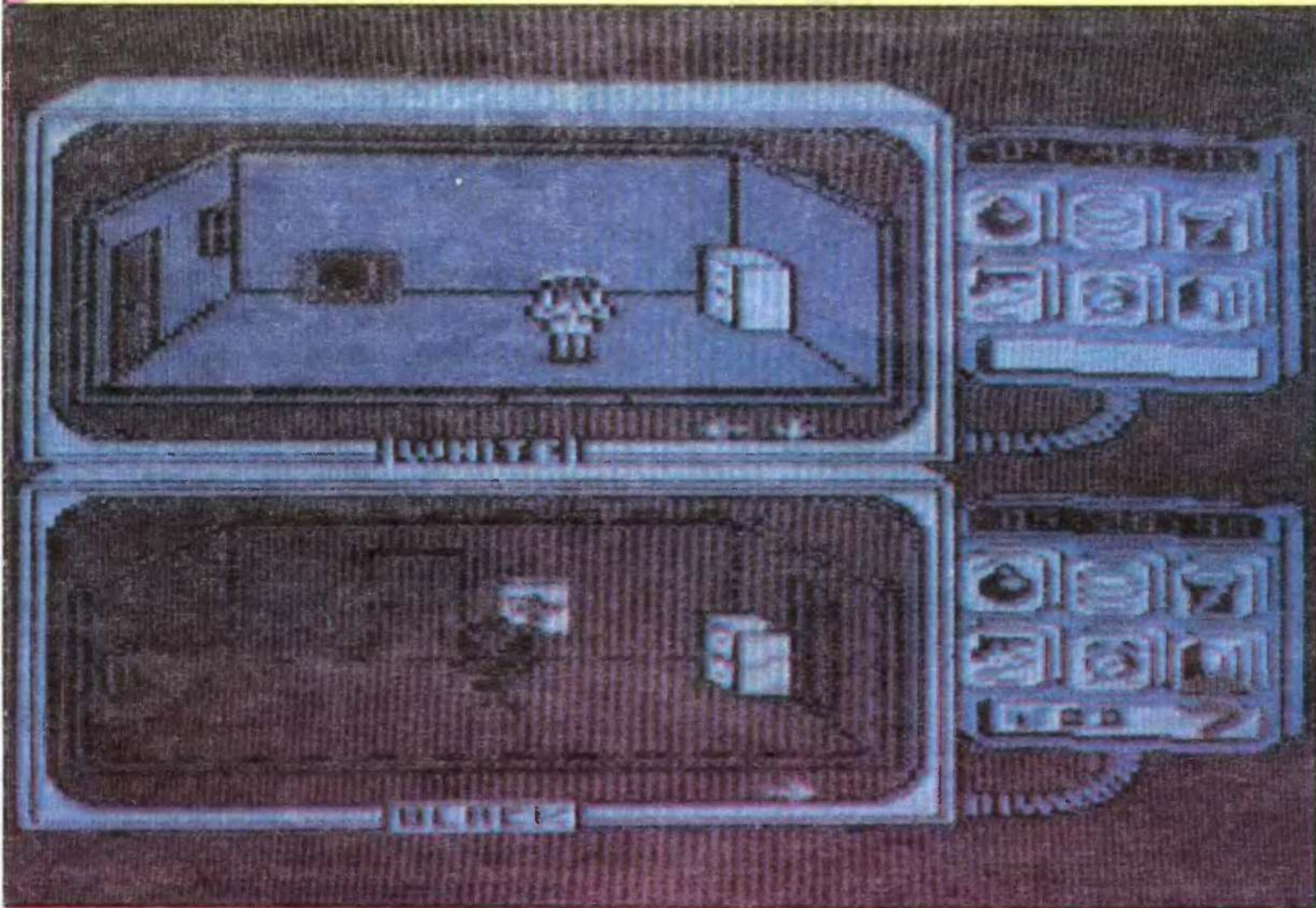
Ocena (w skali 1—10):

GRAFIKA	7
WIERNOŚĆ	7
SENS GRY	8

Producent: Strategic Simulations Inc.

(ziew)

SPY VS SPY XL/XE



Można powiedzieć, że firma „First Star” weszła na rynek oprogramowania przebojem. Od samego początku produkty tej firmy znalazły rzeszę fanów, bo ktoś nie zna „BOULDER DASH 'a” czy właśnie „Spy vs Spy”. Pod względem graficznym gra jest poprawnie wykonana, animacja całkiem płynna, a dyskretna muzyka i starannie dobrane dźwięki sprawiają, że można w nią grać godzinami.

Fabula gry nie jest skomplikowana, ale dość popularny wątek zmagania obcych wywiadów na terenie neutralnego państwa przyciąga graczy. Bohaterami są szpiegowie („Czarny” i „Biały”), którzy mają do wykonania takie samo zadanie z tym, że każdy pracuje dla siebie i jednocześnie przeszkadza drugiemu.

Polem zmagania szpiegów wywiadu jest budynek ambasady, opuszczony przez personel po godzinach pracy, ale pełen zabezpieczeń anty szpiegowskich.

Misja (powierzona Tobie, jako jednemu ze szpiegów) polega na odnalezieniu tajnych dokumentów, klucza (aby opuścić teren ambasady), paszportu (potrzebny do ucieczki za granicę) i pewnej ilości twardej gotówki (bo trzeba przecieć za coś żyć). Wszystkie wymienione przedmioty należy schować do teczki i samolotem opuścić „gorący” teren.

Ambasada składa się z pewnej liczby pomieszczeń, które trzeba dokładnie przeszukać (wszystkie sprzęty w każdym pokoju także). Potrzebne Ci przedmioty mogą być wszędzie: w skrytce za obrazem, pod tele-

wizorem, w szafce, biurku, czy w innych sprzętach.

Dla uatrakcyjnienia gry możesz (ale Twój przeciwnik także) zastawiać pułapki (niebezpieczne i dla Ciebie, nie zapominaj gdzie je zastawiłeś, aby samemu nie wpaść):

1. Bomba (w meblach) — można ją zlikwidować za pomocą wiadra (skrzynka na lewej ścianie).
2. Sprężyna (w meblach) — likwidacja kleszczami (skrzynka na prawej ścianie).
3. Wiadro z kwasem (na drzwiach) — wystarczy zabrać parasol (z wieszaka).
4. Pistolet na lince (w drzwiach) — likwidacja nożyczkami (z apteczki).
5. Bomba zegarowa (miejsce dowolne) — wybuch po 10 sekundach.

Ekran podzielony jest na dwie części pokazujące miejsca, w których aktualnie znajdują się gracze (strzałki pod ekranem pokazują drogę do teczki z przedmiotami). Możesz też korzystać z terminala funkcyjnego (tzw. Trapulator), gdzie oprócz wymienionych przedmiotów (bomba itp.) masz do dyspozycji mapę (rozkład pokoi i pięter) ambasady, zegar (odliczający czas misji), światło ostrzegawcze i wykaz zdobytych przedmiotów (jeżeli dany przedmiot „miga” oznacza to, że należy go schować do teczki, by nie zgubić w starciu z przeciwnikiem).

Po skompletowaniu przedmiotów i umieszczeniu ich w teczce należy odnaleźć drzwi (z rysunkiem dwóch samolotów) prowadzące na lotnisko, co jest równoznaczne z zakończeniem misji. Zwycięski szpieg odleci samolotem.

Na koniec trzeba dodać, że w czasie gry mogą nastąpić starcia (na pałki) między szpiegami (najlepiej podczas „walki” być atakującym stojąc po prawej stronie), w wyniku których są tracone przedmioty „migające”, które masz przy sobie. Przedmioty te można odnaleźć w tym samym pomieszczeniu, gdzie toczyła się walka.

Czy Twoja misja będzie udana, zależy tylko od Ciebie. Powodzenia!

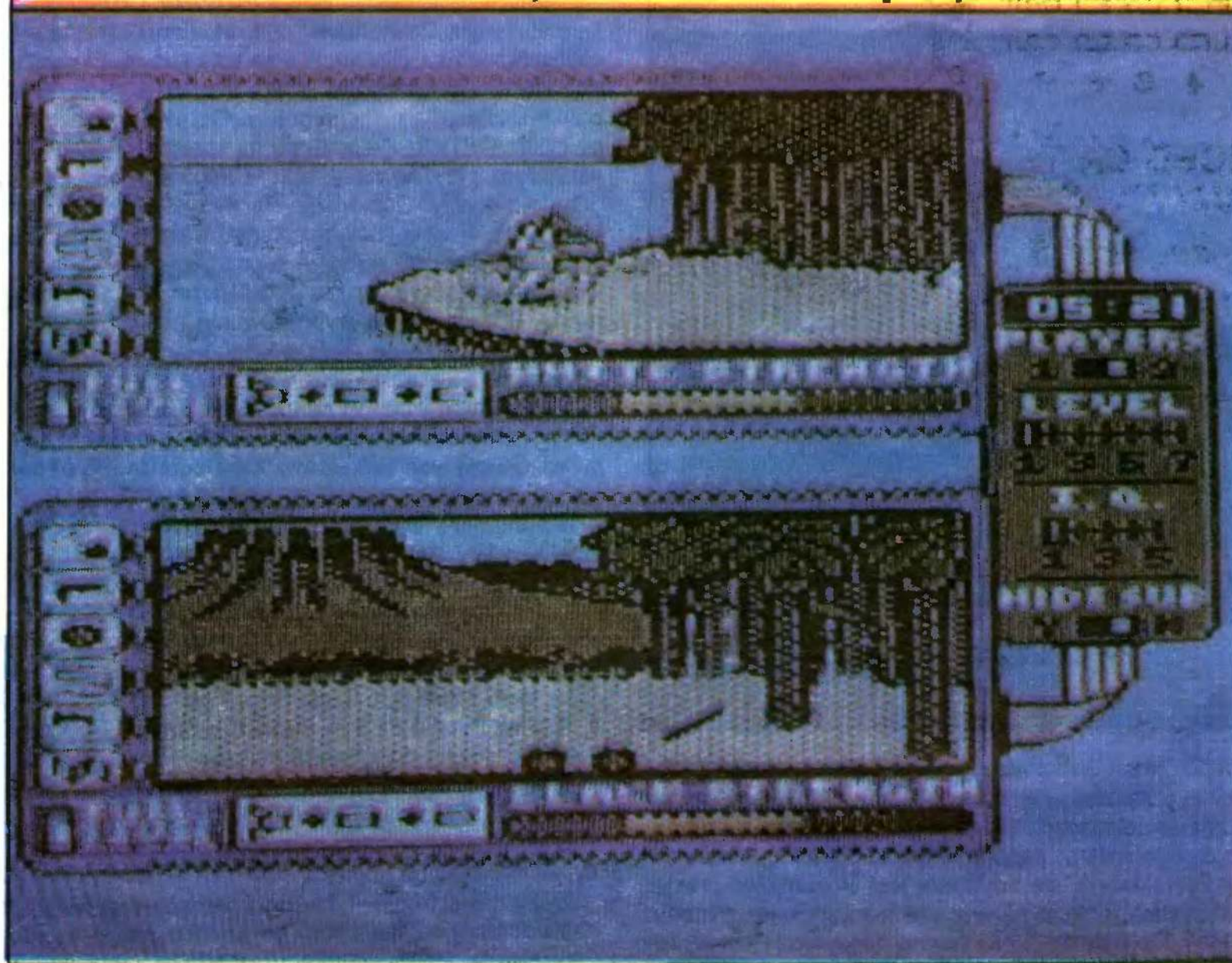
OCENA (w skali od 1 do 10):

GRAFIKA	8
DŹWIĘK	7
SENS GRY	8

Producent: First Star Software Inc.

Autor: Mike Riedel, wersja Atari — Jim Nangano

SPY VS SPY II (The Island Caper) XL/XE



Każdy, kto z powodzeniem ukończył misję w ambasadzie („Spy vs Spy”) wie, że bohater odleciał samolotem, aby zameldować swym pracodawcom o sukcesie.

Już po przylocie dowiadujemy się, że w dowództwie czeka na niego nowy rozkaz. Brzmi on: „Dostać się na

wyspę Pionier i wykraść prototyp nowej rakiety z bazy pod wulkanem”.

Rakieta składa się z trzech części, a każda z części znajduje się w innej części wyspy. Należy dodać, że nie będziesz sam na wyspie, gdyż Twój — znany już — wróg też ma to samo zadanie i na pewno będzie Ci

przeszkadzał.

Na wyposażeniu masz ten sam Trapulator, co podczas poprzedniej misji. Masz na nim zaznaczony Twój stan posiadania przedmiotów, które znajdziesz na wyspie.

Ponadto na wyspie może czyhać na Ciebie wiele pułapek, najczęściej zastawianych przez wroga lub całym naturalnych (ruchome piaski, rekin blisko brzegu itp.). Ale i Ty możesz przeszkadzać wrogowi:

1. Bomba — chowana w piasku.
2. Potrząsk — umieszczany na drzewie.
3. Napalm — umieszczany w wodzie.
4. Łopata — którą można wykopać dołek i przykryć go patykami.

Dla swojego bezpieczeństwa masz na wyposażeniu pistolet (nie zawsze sprawny), którym możesz pozbawić życia przeciwnika. Masz także mapę, z której możesz odczytać swoje położenie i miejsce ukrycia rakiety.

UWAGA! Gdy jesteś na jednym ekranie ze swoim wrogiem, to niesione przedmioty znikają, a Ty jesteś zmuszony do walki.

W grze istnieją możliwości wyboru kilku opcji: liczby graczy, poziomu trudności, inteligencji przeciwnika, ujawnienia lub nie okrętu podwodnego (przed końcem gry) — wszystkie te opcje należy wybrać przed rozpoczęciem gry.

Powodzenie tej misji (podobnie jak poprzedniej) zależy tylko od Ciebie, więc bądź czujny. Powodzenia.

Ocena (w skali od 1 do 10):

DŹWIĘK	8
GRAFIKA	8
SENS GRY	8

Producent: First Star Software Inc.
Autor: Mike Riedel

GREEN BERET XL/XE



Wielki „renesans”, jaki przeżywają gry zręcznościowe zawdzięczamy takim popularnym tytułom jak „Who Dares Wins II”, „Gunlaw” czy właśnie „Green Beret” (znane już z innych komputerów). Gry te można śmiało nazwać lekimi i przyjemnymi, ale na pewno nie prostymi, gdyż wymagają refleksu, zręczności i precyzji. Sama gra jest dość ciekawa, ale niestety także monotonna i może szybko się znudzić (choć są tacy, którzy mogą grać godzinami).

Rzecz dzieje się na terenie „obcej” bazy wojskowej, a bohater jest komandosem ze słynnej formacji Green Beret (Zielone Berety).

TOP SECRET

Raport nr. 1673/3a/87, dn. 3.6.87, ze zwiadu powietrznego.

Na terenie bazy wojskowej wroga w sektorze 17/A wykryto mały obóz jeniecki. Z raportów wywiadu można przypuszczać, że na jego terenie znajdują się jeszcze nasi żołnierze. Baza jest bardzo pilnie strzeżona, a sprzęt zgromadzony na jej terenie nadaje się do desantu na większe miasto.

Lot rozpoznawczy trwał 3h 48 min., na średniej wysokości 460 m, przy zmiennej pogodzie i niskim pułapie chmur.

Za zgodność:
f/o. Clive Townsend
stg. Gres Follis
TOP SECRET

Rozkaz dzienny nr 79/4b/87, dn. 10.6.87.

Do cpt. Fergusona „Delta 4” McNeill, jednostka specjalna „Green Beret”.

Na podstawie rozpoznania powietrznego z dn. 3.6.87

W sektorze 17/A, w bazie wroga znajduje się grupa naszych jeńców wojennych. Baza jest strzeżona przez 90 żołnierzy i psy. Broń znajdująca się w bazie wystarczy do regularnego desantu. Cała baza jest podzielona na trzy podstawowe strefy, z których każda składa się z trzech mniejszych (ze względu na rozśrodkowanie broni ciężkiej).

Stref bronią uzbrojeni żołnierze (średnio co piąty zdąży użyć broni), komandosi (walczący bez broni w stylu karate), psy wielkie owczarki tresowane do walki z ludźmi) oraz śmigłowce. Ponadto baza jest podminowana i nieostrożny krok może być ostatnim w życiu. Broń można zdobyć jedynie odbierając wrogowi (dopiero w części środkowej pierwszej strefy).

ZADANIE: za wszelką cenę uwolnić jeńców, a każdego napotkanego wroga zabić.

Podpisano: mjr Steve Crow.

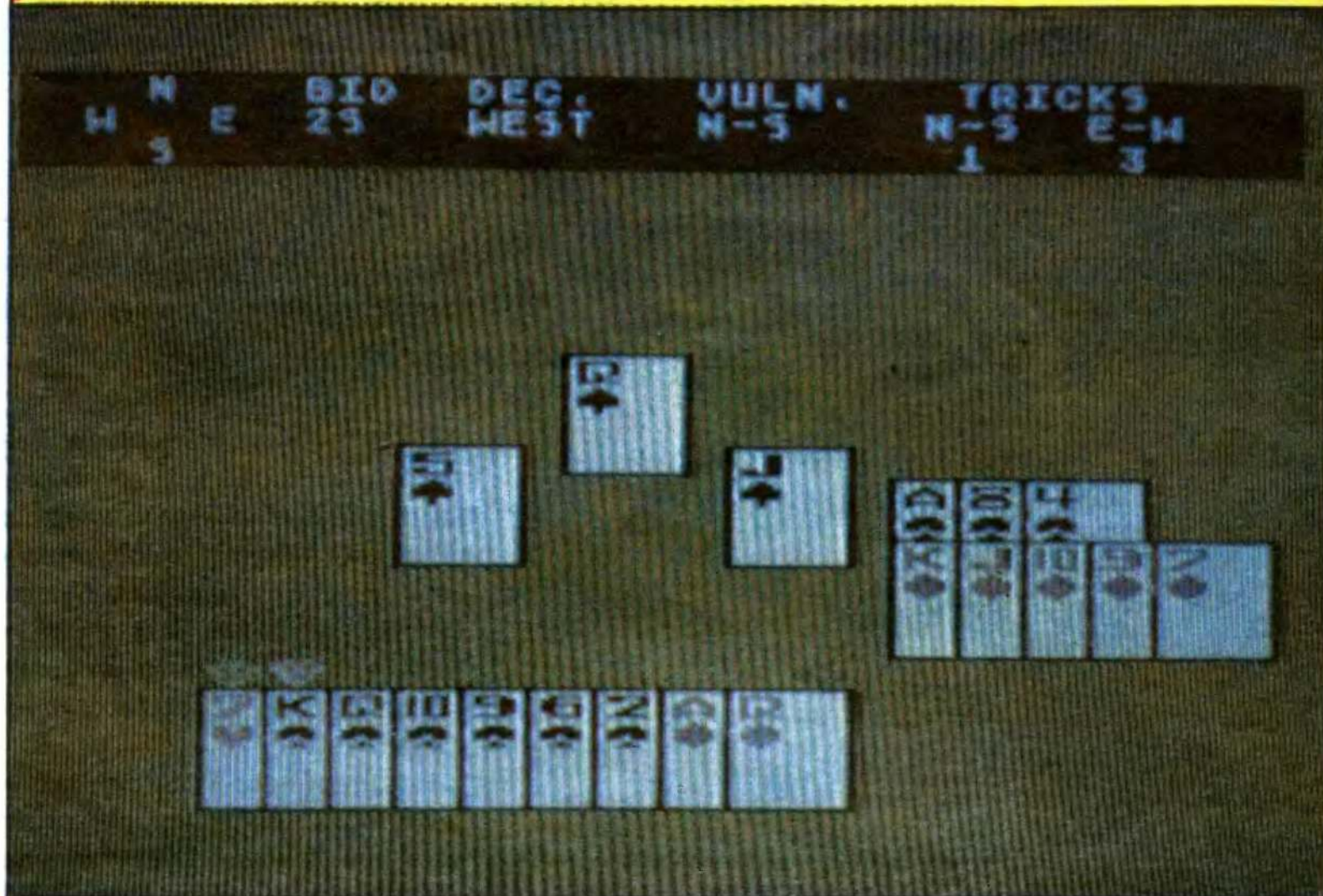
Ocena (w skali od 1 do 10):

GRAFIKA	5
DŹWIĘK	4
SENS GRY	4

Producent: Imagine/Konami, wersja Atari: De Re Software

BLUE TEAM BRIDGE

XL/XE



Użytkownicy ośmiobitowych komputerów Atari doczekali się wreszcie dobrego programu brydżowego. Zajmuje on wprawdzie razem ze wszystkimi dodatkami aż dwie pełne dyskietki, ale ich zawartość usatysfakcjonuje każdego gracza.

Program składa się z dwóch podstawowych części: licytacji i rozgrywki. Są one wywoływane jedna z drugiej i w zasadzie w dowolnej chwili. Można więc dokonać kilku licytacji (maksymalnie 10) zapisując układ kart po każdej z nich. Można również grać normalnie, to znaczy rozgrywka natychmiast po licytacji.

Oczywiście nie trzeba koniecznie grać. Jeżeli uczysz się dopiero, to po każdej licytacji możesz sobie obejrzeć wszystkie karty i ewentualnie potem zagrać, oraz obejrzeć karty po rozgrywce. Podczas nauki można też dowolnie ułożyć karty i wtedy przeprowadzić licytację i rozgrywkę.

Licytację można przeprowadzać kilkakrotnie w różnych wariantach. Oczywiście to samo dotyczy rozgrywki. Dodatkową możliwością jest ćwiczenie niektórych konwencji brydżowych.

Bardziej zaawansowanych graczy zainteresuje z pewnością system licytacji wykorzystywany przez program. Podstawowa licytacja jest prowadzona we Wspólnym Języku. Uzupełniony jest on przez kilka użytecznych konwencji. Najbardziej znana jest konwencja Blackwooda — pytanie o asy przez zgłoszenie 4 BA. Niestety nie można już zapytać o króle.

Pozostałe konwencje nie są tak rozpowszechnione. Jest to konwencja Staymana — pytanie o czwórki w starszych kolorach — 2 trefl po 1 BA. Bezpośrednio z nią jest związana druga konwencja, tzw. Jacoby Transfer Bid. Jest to wersja konwencji Texas zgłaszana już

na wysokości dwóch, a więc w odpowiedzi na pytanie Staymana.

Program w całości sterowany jest joystickiem, a kolejne warianty wybierane są z trzech menu: licytacji, rozgrywki i tworzenia rąk. Ujawnia się tu drobna wada, gdyż przejście z jednego do drugiego menu wymaga wczytania odrębnej części programu, a więc trwa kilkanaście sekund.

Oprócz dziesięciu różnych układów kart można zapisać na dyskietce również osiągnięte wyniki. Program prowadzi niezależną punktację dla czterech graczy, wystarczy więc miejsca dla całej rodziny lub dla kilku kolegów.

Prezentowany obraz jest wyraźny dzięki zastosowaniu trybu GRAPHICS 1 do wyświetlania kart. Poza tym zamieniono miejscami trefle i kara, aby ułatwić orientację w układzie ręki. Inne informacje są wyświetlane w standardowym trybie tekstowym. Podczas licytacji są to kolejne odzywki graczy, zaś przy rozgrywce ilość lew wziętych przez obie pary, rozgrywany kontrakt i inne istotne dane.

Wspomniałem wcześniej o uzupełnieniach. Właściwy program zajmuje jedynie jedną stronę dyskietki. Na jej drugiej stronie znajduje się krótka instrukcja korzystania z programu wraz z wykazem stosowanych w programie otwarć, pierwszych odpowiedzi i konwencji specjalnych.

Prawdziwą niespodzianką jest natomiast druga dyskietka. Z tytułowana ona jest „Fundamentals of Bridge” (Podstawy brydża) i tytuł ten wiernie odzwierciedla jej zawartość. Jest to skrócony podręcznik brydża dla początkujących. Zapisanie go w plikach tekstowych pozwala na wykonanie wydruku (prawie 60 stron) i odrębne jego wykorzystanie.

Podręcznik zwięźle (przeważnie w postaci zbiorów reguł) przedstawia zasady licytacji i rozgrywki oraz bezpieczne i agresywne ich warianty. Wszystko to ilustrowane jest licznymi przykładami i poparte przejrzystymi, choć suchymi objaśnieniami.

Mogę z czystym sumieniem polecić ten program wszystkim brydżystom, nawet mającym dość dużą praktykę w grach „w kółko”. Nie ośmielam się polecać go „turniejowcom”, ale chętnie usłyszałbym opinie któregoś z liczących zawodników tej konkurencji.

Ocena (w skali 1—10):

GRAFIKA 8

MOŻLIWOŚCI 7

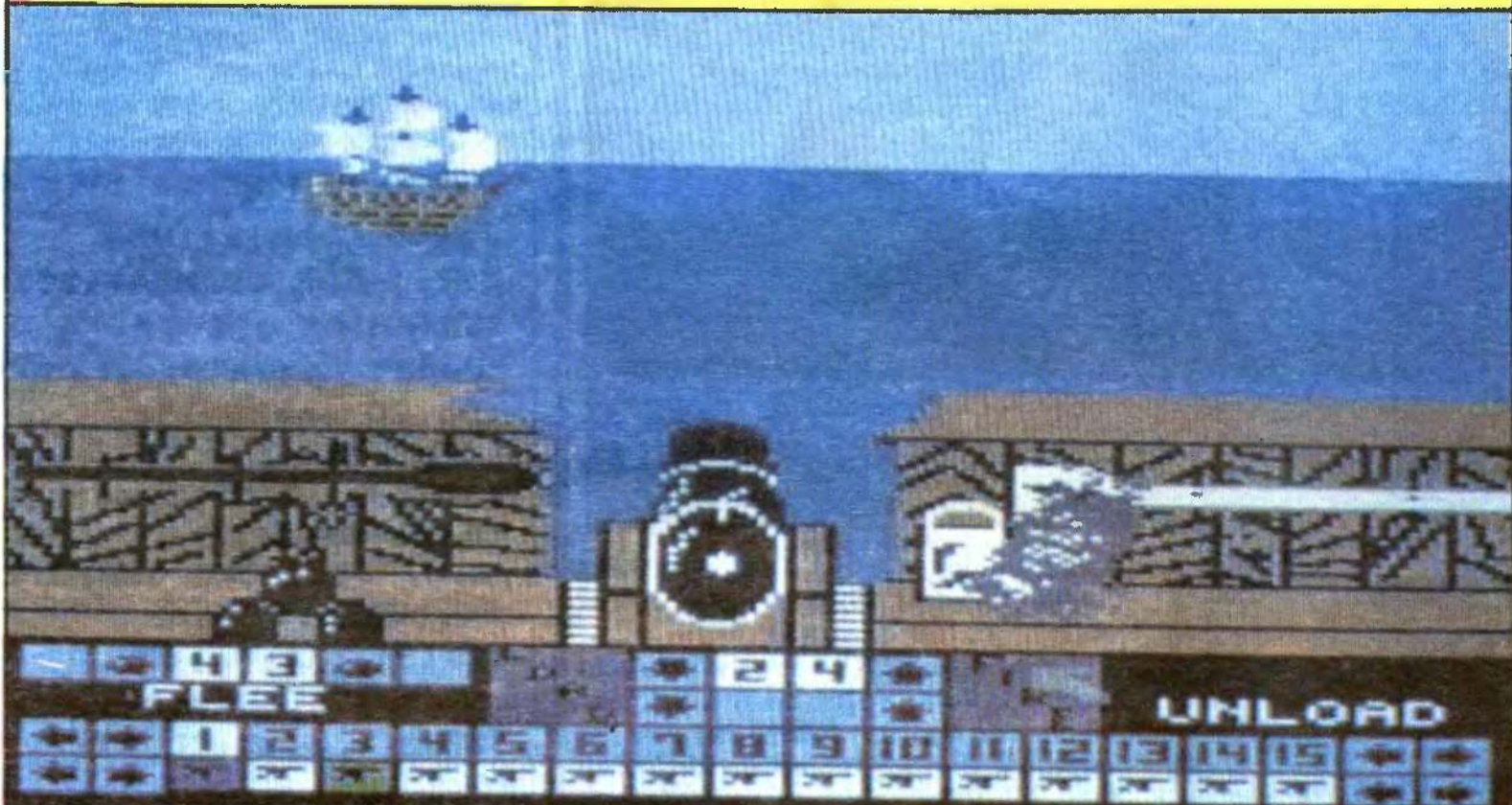
OBSŁUGA 9

Producent: Antic Publishing, Inc.

Autor: Walt Huber

(ziew)

PIRATES OF THE BARBARY COAST XE/XL/ST



Piraci z Wybrzeża Berberyjskiego (Afryka Zachodnia) porwali córkę kapitana fregaty „American Star”. Trzeba zapłacić okup lub odbić ją siłą. Oba zadania są jednakowo trudne. Pieniądze na okup można zdobyć jedynie handlując z arabskimi kupcami w różnych portach. Targowanie się z nimi nie jest łatwe, poza tym na morzu stale grasują piraci. Uwolnienie porwanej wymaga przede wszystkim odnalezienia kryjówek piratów i oczywiście stoczenia z nimi walki. Wszystko to trzeba wykonać w ciągu 30 dni.

Scenariusz nie jest oryginalny — wątek ten przewija się w światowej literaturze przynajmniej od 200 lat. Wystarczy wymienić tu Stevenson, Marrayata czy Forestera, nie mówiąc już o licznych powieściach wyko-

rzystujących ten motyw. Gra „Pirates of the Barbary Coast” prezentuje nam to samo w wersji komputerowej.

Do dyspozycji masz fregatę „American Star” z załogą, wyposażeniem i ładunkiem. Pływając między śródziemnomorskimi portami Afryki musisz przewozić różne towary, handlując nimi w poszczególnych portach, aż zarobisz 50 tysięcy dukatów na okup. Niestety część zarobionych pieniędzy trzeba wydać na żywność i wynagrodzenie dla załogi oraz na naprawy.

Podczas żeglugi często będziesz spotykał statki pirackie. Można, a nawet trzeba, z nimi walczyć, lecz wymaga to sporej zręczności i dokładnego zapamiętania sposobu obsługi dział. Jest on odwzorowan bardzo

dokładnie — tak dokładnie, że nieco nuży i denerwuje. Starcia można oczywiście uniknąć, ale spowoduje to przedłużenie podróży, a nawet powrót do portu wyjścia. Walka natomiast zawsze powoduje większe lub mniejsze straty i zmusza do naprawy statku oraz werbowania nowych marynarzy (przy okazji agent może udzielić interesujących informacji, ale to też kosztuje).

Handel również nie jest najprostszy. Trzeba uważać, aby nie przepłacić za nabywany towar lub nie sprzedać go za bezcen. Z drugiej strony kupiec może się obrazić, gdy targujesz się zbyt długo. Wybór towaru do sprzedania lub zakupu jest przy tym wspaniałym testem intuicji i zmysłu handlowego.

Powszechnie wiadomo, że zrabowane skarby piraci zakopywali na samotnych, bezludnych wyspach. W Twoim posiadaniu znajduje się mapa takiej wyspy z zaznaczonym miejscem ukrycia pirackiego skarbu. Jego odnalezienie znacznie ułatwi zebranie odpowiedniej kwoty. Ale najpierw trzeba znaleźć wyspę! Może się wtedy okazać, że na tej wyspie zamiast skarbów jest kryjówka piratów.

Niestety nie ma róży bez kolców. Gra jest bardzo pasjonująca, lecz wymaga ciągłości — w trakcie gry nie można jej przerwać i zapisać na dyskietce. Przerwę można zrobić tylko w porcie, gdyż wtedy zatrzymywany jest upływ czasu (ale nie można wyłączyć komputera). Początkowo denerwujące jest również ładowanie dział i strzelanie z nich — wyraźnie autorzy przesadzili w tym miejscu z dbałością o realia. Mimo tych niedociągnięć mogę ją jednak z czystym sumieniem polecić wszystkim, którym znudziło się już bezmyślne łamanie joysticka.

Ocena (w skali 1—10):

GRAFIKA 8

DŹWIĘK 5

REALNOŚĆ 9

SENS GRY 7

Producent: Starsoft Development Labs, 1986

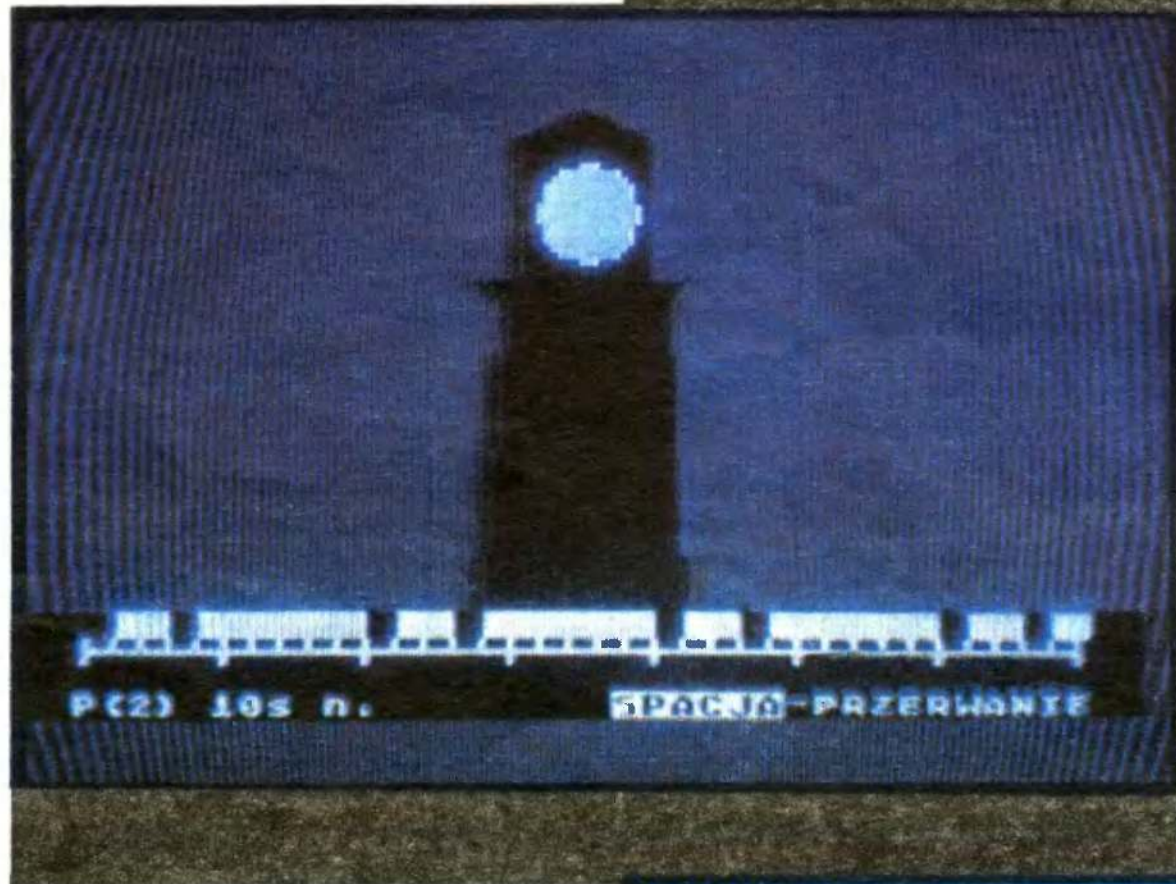
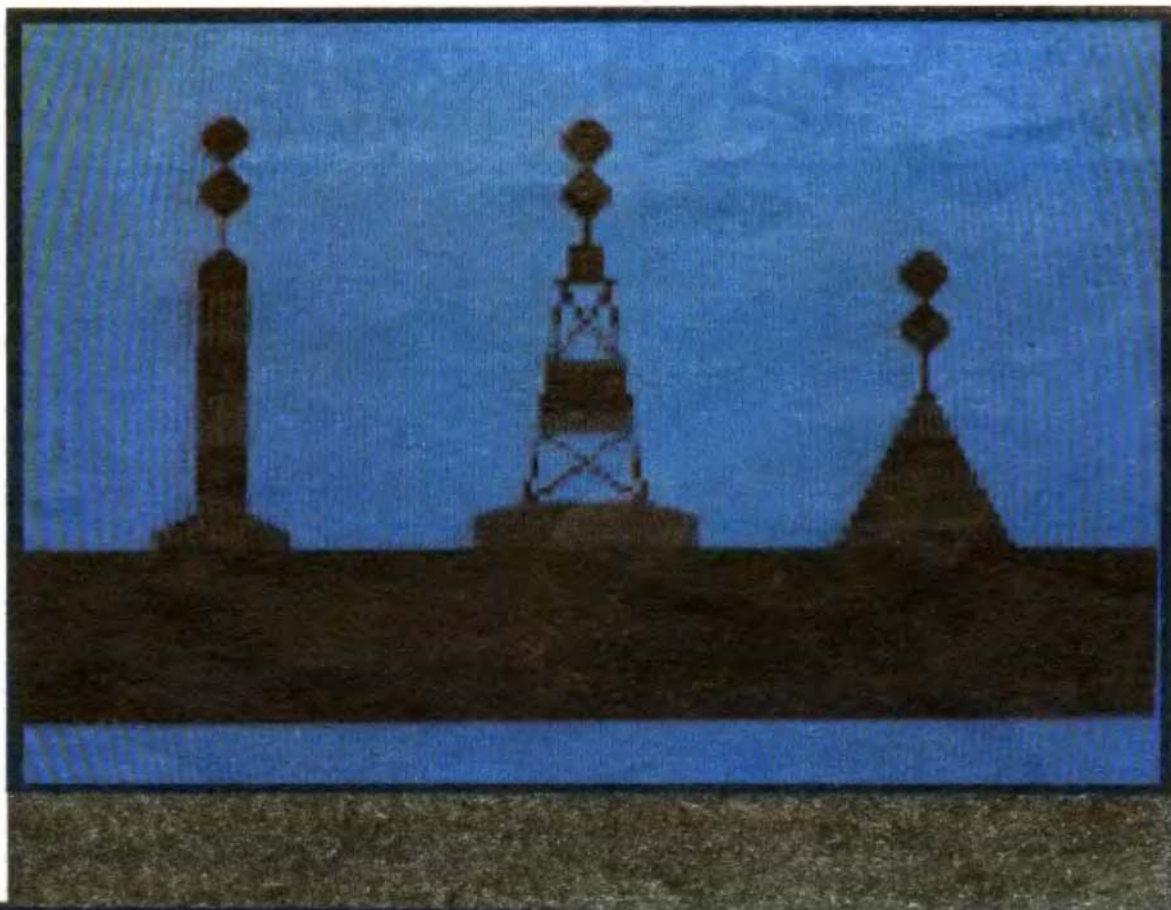
Autorzy: Hal McCrery i Craig Morehouse

Wersja Atari: James McCormick

(ziew)

EDUKACJA LATARNIA NA HORYZONCIE

Nasz jacht wpływa właśnie nocą na wody duńskich cieśnin. Wokół groźne mieliżny i rafy. Przepłyniemy między nimi szczęśliwie jeśli tylko prawidłowo odczytamy wszystkie sygnały świetlne przekazywane nam przez latarnie morskie i pławy.



Znajomość locji, czyli podstawowych sygnałów obowiązujących na morzu, możemy zdobyć nie wstając od monitora naszego „Atari 800XL”. Wystarczy jedynie sięgnąć do domowej biblioteczki programów i wyciągnąć stamtąd dzieło naszego redakcyjnego kolegi Wojtka Zientary — program „Locja”.

„Locja” składa się z dwóch części: pokazu i egzaminu. Pokaz rozpoczyna się od widoku latarni morskiej mrugającej do nas z ekranu światłami nawigacyjnymi. W dole pod latarnią widać opis sygnału. Różnią się one między sobą ilością i długością poszczególnych rozbłysków a także kolorem światła.

Znika latarnia morska. Na jej miejsce zjawiają się pławy i światła systemu IALA — morskie oznakowania sygnalizacyjne. Dla każdej pławy najważniejszy jest znak szczytowy i świecące na niej światło. Po kolei pojawiają się pławy oznaczające prawą i



lewą stronę toru wodnego, jego rozgałęzienia, znak bezpiecznej wody itd.

Gdy już wydaje nam się, że zjedliśmy wszystkie rozumy i możemy startować do egzaminu na stopień sternika jachtowego rozpoczynamy zmagania z drugą częścią „Locji”. Znow na ekranie pojawia się światło. Tym razem jednak sygnał powtarza się zaledwie 5 razy, później nasz „Atari” podsuwa nam kilka odpowiedzi, a my mamy, bagatela, kilkanaście sekund na wybranie prawidłowej odpowiedzi. Po 10 losowo wybranych światłach latarni zjawia się 10 pław — część z nich widoczna jest „w nocy”, część „w ciągu dnia”. Staramy się jak możemy, za każdym razem wybieramy rozważnie odpowiedź, po czym na zakończenie próby komputer bezlitośnie oblicza nasz wynik i wystawia ocenę. Jeśli myślisz, że ewentualną „dwóję” łatwo będzie ci wymazać z ekranu, zawiedziesz się. Jedynym sposobem na to, by egzaminator czy wykla-

dowca nie dojrzał twojej oceny, jest wyłączenie komputera. Tylko że wtedy na ogół wiadomo, co było na ekranie.

„Locję” stosuje na co dzień w szkoleniu Yacht Klub Polski w Warszawie. Może mieć ten program również twój klub żeglarski czy drużyna wodniacka. Warunkiem jest jednak posiadanie „Atari” z kolorowym monitorem i stacją dysków.

Omijać rafy można uczyć się siedząc w fotelu. Jednak nie namawiamy tylko do takiej formy uprawiania żeglarstwa — za mało w niej wiatru. Ahój żeglarze!

Grzegorz Onichimowski

Aktualnie zestaw programów dla żeglarstwa zawiera następujące pozycje:

1. Locja morska
2. MPZZM
3. Znaki żeglugowe
4. Morskie sygnały wzywania pomocy. Powyższe programy należą do tzw. Public Domain Software i można je otrzymać bezpłatnie u autora.

PCHĘŁKA

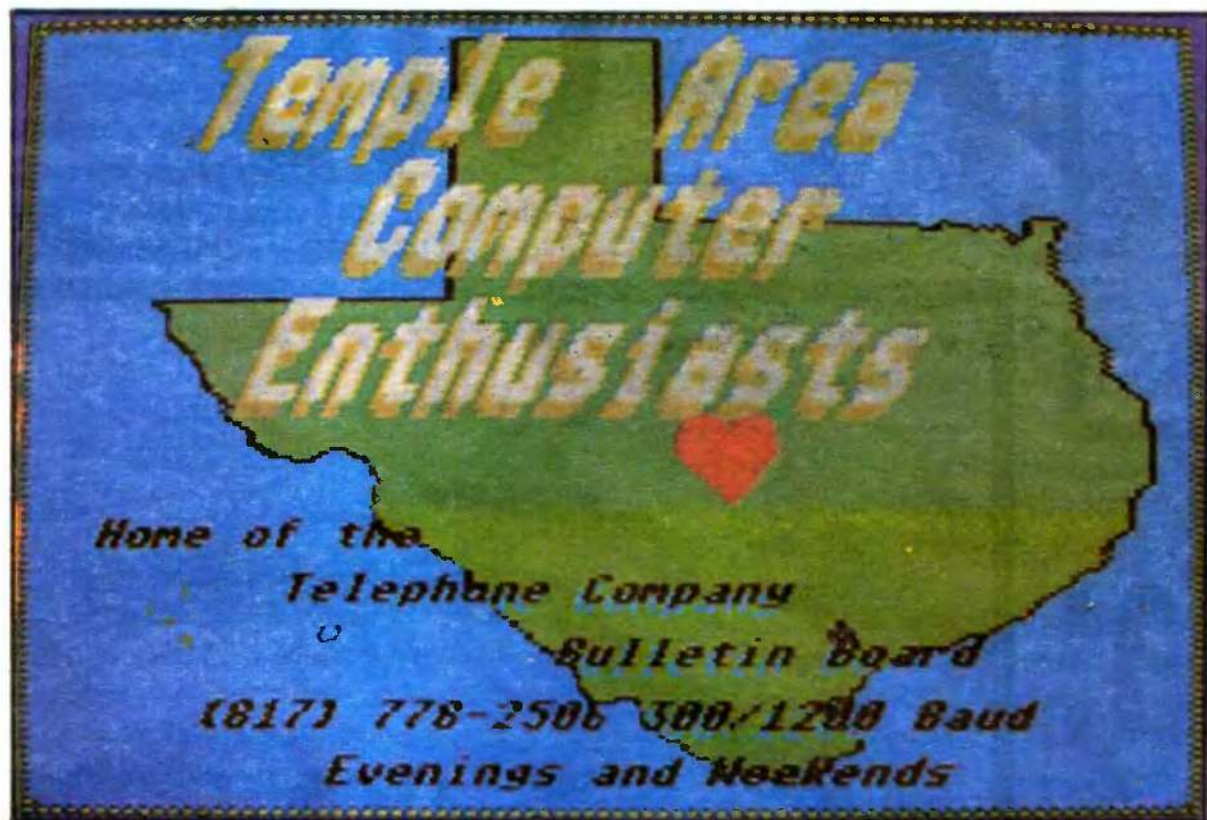
Jeżeli jesteś użytkownikiem magnetofonu, wiesz dobrze, jaką bolączką jest zapis na kasetę programów bez nazwy.

Rozwiązań tego problemu jest wiele, ale zazwyczaj cechują je dwie wady: pierwsza to wielkość programu, który należałoby wpisać do pamięci (np. Kasetowy System Operacyjny na 208 linii), druga to konieczność wgrzywania programu zarządzającego przed programem właściwym.

Prezentowany przez mnie program, choć dość ubogi, jest skuteczny i omija obie wady. Po wpisaniu zapisujemy go na kasecie komendą LIST „C:”. Jeżeli mamy program o numerach linii mniejszych od 32750 i chcemy, by podczas ładowania ukazał się napis z nazwą programu, musimy wykonać kilka czynności: doładowujemy pchełkę z kasyty komendą ENTER „C:” i uruchamiamy rozkazem GOTO 32750. Podajemy nazwę programu (max 20 liter), przygotowujemy taśmę do zapisu i naciskamy dowolny klawisz. Program zapisany w tej wersji jest dłuższy jedynie o jeden rekord dodanego pilota. Ładuje się go komendą ENTER „C:”, uzyskując mimo tego efekt identyczny z komendą CLOAD (gdyż pilot używa tej komendy).

```
32750 REM PCHLA by
Jakub Cebula 1988
32751 CLR :DIM A$(20)
:GRAPHICS 0:?:?
"Podaj nazwe programu
(Do 20 znakow)":
INPUT A$
32752 ? :? "Przygotuj
tasme !":OPEN #1
,8,0,"C":POKE 559,0:?:
#1;"GR.0:?:?";C
HR$(34);A$;" 1s
loading ";CHR$(34);
32753 ? #1;":CL.#7:
POKE 764,28:CLOAD":
CLOSE #1:?:CHR$(125)
:POSITION 2,5:?"3
2750":?"32751":?
"32752":?"32753"
32754 ? "32754":?:?
"POKE 764,5:CS.:P
OKE 842,12:GR.0":POSIT
TION 0,0:POKE 842
,13:END
```


TACE



TACE — Temple Area Computer Enthusiasts (Entuzjaści Komputerów z Okolic Temple) znajduje się dokładnie w środku Teksasu. Obejmuje swoim działaniem obszar około 500 mil kwadratowych. Grupa ma również członków w innych stanach oraz w RFN.

Okolica Temple jest silnie zmilitaryzowana — znajduje się tu ponad 280 tysięcy żołnierzy. Z drugiej strony samo Temple ma 42 mieszkańców i 10 kilometrów od jednego końca do drugiego. Do najbliższego dostawcy sprzętu Atari jest stąd ponad 100 km, choć ostatnio pojawił się nowy w odległości „tylko” 65 kilometrów.

Bibliotekarzem grupy i operatorem biuletynu (sysopem) jest Jim Wesolowski — inwalida po wojnie w Wietnamie. W jego domu odbywają się trzy razy w miesiącu oficjalne spotkania. Niektórzy członkowie dojeżdżają na nie z odległości ponad 150 kilometrów.

Wielu członków TACE jest żołnierzami, więc najliczniej reprezentowani są ludzie w wieku 18—34 lat. Jednakże członkostwo TACE obejmuje całe rodziny, a nie pojedyncze osoby. Łącznie z dziećmi całkowita liczba członków grupy przekracza 400 — 103 rodziny są użytkownikami Atari.

Mamy w grupie nie tylko użytkowników Atari — mówi Wesolowski — Moim komputerem jest obecnie St. Na przykład z właścicielami IBM współpracują przez język c. Początkowo A w nazwie grupy oznaczało Atari, lecz z braku innych grup zrzeszani są użytkownicy wszystkich komputerów.

Biuletyn

Telefoniczny biuletyn TACE pracuje na jednomegabajtowym 520ST Wesolowskiego z twardym dyskiem Atari 20 MB i programem MichTron BBS. Jest on czynny codziennie od 5 po południu do 10 rano, a w dni wolne od pracy całą dobę.

Dysponujemy około 8 MB zbiorów muzycznych i graficznych — mówi Wesolowski — zawiera się w tym 1 MB — 108 tytułów — utworów do Music Studio i 3,5 MB piosenek do Hybrid Arts MIDI. Mamy również 2 MB rysunków DEGAs i NEOchrome.

Zgromadzone utwory reprezentują szeroką gamę — od „Uwertury do Wilhelma Tella” do „Błękitnej Rapsodii”. Wszystkie zbiory muzyczne i graficzne są maksymalnie skondensowane przy użyciu programu ARCHiver. Na przykład zbiory Hybrid Arts są zmniejszane z 330.000 bajtów do 64.000, a rysunki NEOchrome z 32.000 do 5000 bajtów.

45% wywołań biuletynu to połączenia długodystansowe — bywają wywołania z Ohio i z Kalifornii. Biuletyn TACE jest dostępny dla wszystkich. Wywołujący może zapisywać i odczytywać zbiory. Czas połączenia jest ograniczony do 35 minut. Stosuje się prędkości transmisji 300 i 1200 bodów.

Wiadomości dostępne z biuletynu są aktualizowane przez Wesolowskiego raz na miesiąc. Zajmuje się on głównie przeglądem oprogramowania, lecz twierdzi, że ma stale zbyt mało czasu, gdyż posiada tylko jeden komputer.

Działalność

Prezes TACE Paul Fische, jest szefem rekreacji i rehabilitacji w Centrum Medycznym Weteranów (VA) koło Temple, a TACE jest jedną z kilku grup użytkowników Atari w Stanach Zjednoczonych współdziałających z VA.

Grupa wyposażyła szpital VA w komputery 8-bitowe, które służą w domu opieki do rekreacji i rehabilitacji, głównie dla treningu koordynacji wzrokowo-ruchowej ofiar wypadku.

TACE współpracuje także ze szkołą powszechną w Temple, gdzie prowadzi szkolenie w obsłudze różnych komputerów i urządzeń peryferyjnych.

Statystyka

Ze 103 rodzin członkowskich posiadających Atari 9,8% używa ST, a pozostałe modeli 8-bitowych w kolejności: 800XL, 800, 130 XE, 65XE i 1200XL. Najpopularniejsze są stacje dysków Atari 1050 i Indus GT.

Najczęściej stosowanymi modemami są Avatex, Volksmodem Atari Supra i Hayes. Dwadzieścia procent członków grupy korzysta z płatnych biuletynów telefonicznych, przede wszystkim Genie, CompuServe i Dow Jones.

Około 6% właścicieli Atari w TACE jest aktywnymi programistami, dalsze 12% programuje sporadycznie. Kolejność stosowanych języków programowania jest następująca Atari Basic, Microsoft Basic i Action! Właściciele ST preferują C, ST Basic i assembler (w takiej kolejności).

Rodzaje oprogramowania najczęściej używane przez członków TACE uporządkowane według ważności tworzą następującą listę:

1. Interesy i finanse
2. Edytory tekstu
3. Programowanie
4. Edukacja
5. Komunikacja
6. Gry
7. Sterowanie
8. Sztuczna inteligencja

**TACE — Temple Area Computer Enthusiasts
3202 Las Cruces Drive, Temple, TX 76502
Biuletyn: (817) 778-2506**

opracowanie wg.
„Antic” 3/87

KLUBY ATARI W USA

Bay Area Atari Users 'Group (BAAUG)
P.O.Box 58459, Palo Alto, CA 94303, USA

Atari Users of Greater Hartford Area (GHA)
503-B East Center Street, Manchester, CT 06040, USA

Atari Lovers of Illiana Equal to None (ALIEN)
c/o Jeff Coe, 706 Center Street, Crown Point, In 46307, USA

Jackintosh Boston Users 'Group (J-BUG)
c/o The Computer Society
One Center Piazza, Boston, MA 02108, USA

Coastal Area Atari Users 'Group (CAAUG)
P.O.Box 5098, Biloxi, MI 39534-0098, USA

High Sierra Atari Users 'Group (HISUG)
P.O.Box 2152, Sparks, NV 89432, USA

Western New York Atari Users 'Group (WNYAUG)
P.O.Box 59, Buffalo, NY 14216, USA

Tinker Atari Computer Enthusiasts (TACE)
P.O.Box 19956, Oklahoma City, OK 73144, USA

Spectrum Atari Group of Erie (SAGE)
P.O.Box 10562, Erie, PA 16514-0562, USA

Peninsula Atari Computer Enthusiasts (PACE)
1212 N. King Street, Lot 37, Hampton, VA 23669, USA

Fort Leavenworth Atari Group (FLAG)
P.O.Box 3233, Ft. Leavenworth, KS 66027-0233, USA

Bluegrass Region Atari Computer Enthusiasts (BRACE)
P.O.Box 13063, Lexington, KY 40583, USA

Atarians of Maine User Group (A-MUG)
P.O.Box 1088, Westbrook, ME 04092, USA

Kansas City Atari Computer Enthusiasts (KCACE)
P.O.Box 5286, Kansas City, MO 64112, USA

Southern Nevada Atari Computer Club (SNACC)
P.O.Box 27617, Las Vegas, NV 89126, USA

National Atari Association (NAA)
35 Pleasant Avenue, Begenfield, NJ 07621, USA

Westmoreland Atari Computer Organisation (WACO)
230 Clairmont Street, North Huntingdon, PA 15642, USA

Austin Atari Computer Enthusiasts (AACE)
8207 Briarwood Lane, Austin, TX 78758, USA

Southside Tidewater Atari Technical Users' Society (StATUS)
4836 Honeygrove Road, Virginia Beach, Va 23455, USA

Huntsville Atari Users' Group
3911 W. Crestview, S.W., Huntsville, AL 35816, USA

Seattle-Puget Sound Atari Computer Enthusiasts (SPACE)
P.O. Box 110576, Tacoma, WA 98411-0576, USA

Madison Area Atari Users' Group (MAAUG)
P.O.Box 56191, Madison, WI 53705, USA

Toronto Atari Federation (TAF)
5647 Yonge Street, Box 1527, Willowdale, Ontario, Kanada M2M 4E9

Panama Canal Atari Computer Users' Group (PCACUG)
Apartado No. 5265, Balboa-Ancon, Republika Panamy

Temple Area Computer Enthusiasts (TACE)
3202 Las Cruces Drive, Temple, TX 76502, USA

Boise Users' Group (BUG)
1030 El Pelar Drive, Boise, ID 83702, USA

Atari Users' Group
1196 Borregas Avenue, Sunnyvale, CA 94086, USA



A oto



nowa gwiazda: star LC-10

Najwyższy poziom technologii japońskiej:

Szeroki wybór zestawów znaków:

Łatwość użytkowania:

Szybkość druku:

Druk kolorowy:

Rewelacyjne ceny:

Pełna oferta:

Funkcja „**PAPER PARK**”: możliwość stosowania pojedynczych stron oraz papieru z perforacją.

8 różnych krojów wbudowanych w drukarkę i znaki ASCII/IBM; wersja Commodore C-64/128; znaki dowolnie programowane.

Kilkanaście funkcji wybieranych za pomocą przycisków na obudowie.

120 lub 144 zn/sek w trybie standard; 30 lub 36 zn/sek w trybie korespondencyjnym.

Wersja LC-10 colour, drukuje w 7 kolorach!

LC-10 lub LC-10C (do C-64/128) — DM 450
LC-10 colour lub LC-10C colour — DM 590
plus transport: DM 40, kabel: DM 20

Oczywiście oferujemy Państwu pełną gamę drukarek Star łącznie z najnowszą **drukarką laserową LS-08** (8 str/ min), 1MB, kompatybilna z HP Laser Jet II) za DM 4500.

Wylączny autoryzowany przedstawiciel na Polskę:

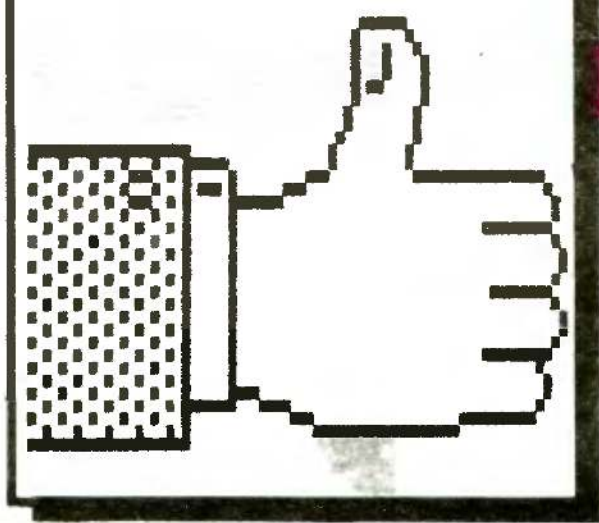
ABC Data
peripherals & computer systems

star
Twoja drukarka

ABC Data Im- und Export GmbH
AugustasträÙe 40. 5300 Bonn 2, RFN
tel. 0228/35.44.80,-90. telex 88.55.66

ABC Computersystems
Alt Moabit 80
1000 Berlin 21
tel. 391.50.99
Telex 181.365

ABC Data GmbH
Ditmar-Koel-Str. 13
2000 Hamburg 11
tel. 31.40.03
Telex 21.66.002



OCCENNA

Mam 13 lat. Napisałem edukacyjny program z matematyki. Program zadaje pytania z mnożenia, dodawania i odejmowania. Próbowałem zrobić również dzielenie, ale komputer żądał dokładnego wyniku, więc wyeliminowałem to. Mój program działa, ale gdy bez wypisania wyniku naciśnie się RETURN, pojawia się komunikat błędu. Proszę Was o ocenę mojego programu i podanie sposobu ominięcia napotkanych trudności.

**Rafał Janiak
al. Gwardii Ludowej 17
64-500 Szamotuły**

Ocenę zaczniemy od struktury programu (patrz listing 4). Wyraźnie widoczne są trzy prawie identyczne bloki — linie 320-430, 440-550 i 560-670. Bloki te znacznie lepiej połączyć w jeden, a różne elementy wyodrębnić za pomocą instrukcji ON/GOSUB. W tym programie można zastosować zwykłą instrukcję IF/THEN, lecz przy większych różnicach między poszczególnymi blokami konstrukcja ON/GOSUB jest znacznie wygodniejsza (a także bardziej elegancka).

Również pętle opóźniające dobrze jest wyodrębnić w oddzielne procedury. Przy większej liczbie różnych wartości wystarczy jedna procedura opóźniająca, a różny czas jej trwania uzyskamy stosując zmienną jako górną wartość licznika pętli.

Wiele instrukcji jest zupełnie zbędnych. Zalicza się do nich większość instrukcji POSITION. Skoro po wykonaniu PRINT komputer przechodzi do następnej linii ekranu, to niepotrzebne jest ustawianie go tam oddzielną instrukcją. Gdy chcemy odsunąć wydruk od lewego marginesu, to również lepiej zastosować kilka spacji niż instrukcję POSITION. Dopiero gdy trzeba wstawić ponad 12 spacji, instrukcja POSITION zajmuje mniej miejsca w pamięci. Nie dotyczy to trybu GRAPHICS 0, w którym dysponujemy jeszcze znakami kontroli kursora.

LISTING 1

```
10 GRAPHICS 2+16
20 FOR A=1 TO 200 STEP 1
30 POSITION 3,4: ? #6;"RAFAL & JANIAK"
40 POSITION 5,6: ? #6;"PREZENTUJE"
50 NEXT A
60 GRAPHICS 2+16
70 FOR A=1 TO 10 STEP 1
80 FOR B=1 TO 30 STEP 1
90 POSITION 0,5: ? #6;"M A T E M A T Y
K A"
100 NEXT B
110 FOR C=1 TO 30 STEP 1
120 GRAPHICS 2+16
130 NEXT C
140 NEXT A
150 GRAPHICS 1+16
160 FOR A=1 TO 150 STEP 1
170 POSITION 5,2: ? #6;"INSTRUKCJA"
180 POSITION 0,4: ? #6;"PO ZADANIU"
190 POSITION 0,5: ? #6;"PRZYKŁADU PRZEZ"
"
200 POSITION 0,6: ? #6;"KOMPUTER"
210 POSITION 0,7: ? #6;"NALEZY"
220 POSITION 0,8: ? #6;"WYPISAC WYNIK"
230 POSITION 0,9: ? #6;"I NACISNAC RET
URN"
240 POSITION 0,10: ? #6;"A KOMPUTER"
250 POSITION 0,11: ? #6;"POINFORMUJE"
260 POSITION 0,12: ? #6;"O JEGO"
270 POSITION 0,13: ? #6;"POPRAWNOSCI"
280 POSITION 0,14: ? #6;"ZYCZYMY"
290 POSITION 3,15: ? #6;"PRZYJEMNEJ"
300 POSITION 6,16: ? #6;"ZABAWY"
310 NEXT A
320 GRAPHICS 2+16
330 FOR A=1 TO 100 STEP 1
340 POSITION 6,5: ? #6;"ETAP 1"
350 NEXT A
360 FOR A=1 TO 10 STEP 1
370 X=INT(1+(100-1)*RND(0))
380 Y=INT(1+(100-1)*RND(0))
390 ? X;"+";Y;"="";:INPUT Z
400 W=X+Y
410 IF Z<>W THEN ? "ZACZYNAJ OD NOWA.
ZLE":GOTO 360
420 IF Z=W THEN ? "DOBRZE"
430 NEXT A
440 GRAPHICS 2+16
450 FOR A=1 TO 100 STEP 1
460 POSITION 6,5: ? #6;"ETAP 2"
470 NEXT A
480 FOR A=1 TO 10 STEP 1
490 X=INT(1+(100-1)*RND(0))
500 Y=INT(1+(100-1)*RND(0))
510 ? X;"-";Y;"="";:INPUT Z
520 W=X-Y
530 IF Z<>W THEN ? "ZACZYNAJ OD NOWA.
ZLE":GOTO 480
540 IF Z=W THEN ? "DOBRZE"
550 NEXT A
560 GRAPHICS 2+16
570 FOR A=1 TO 100 STEP 1
580 POSITION 6,5: ? #6;"ETAP 3"
590 NEXT A
600 FOR A=1 TO 10 STEP 1
610 X=INT(1+(100-1)*RND(0))
620 Y=INT(1+(100-1)*RND(0))
630 ? X;"*";Y;"="";:INPUT Z
640 W=X*Y
650 IF Z<>W THEN ? "ZACZYNAJ OD NOWA.
ZLE":GOTO 600
660 IF Z=W THEN ? "DOBRZE"
670 NEXT A
680 GRAPHICS 2+16
690 FOR A=1 TO 100 STEP 1
700 POSITION 1,3: ? #6;"BRAWO"
710 POSITION 2,4: ? #6;"KONCZYSZ"
720 POSITION 3,5: ? #6;"Z WYNIKIEM"
730 POSITION 4,6: ? #6;"POZYTYWNYM"
740 NEXT A
750 GRAPHICS 2+16
760 FOR A=1 TO 100 STEP 1
770 POSITION 5,5: ? #6;"KONIEC"
780 NEXT A
```

Zbędne są także instrukcje IF Z=W THEN ..., skoro po sprawdzeniu warunków Z<W są one omijane. W tej ostatniej instrukcji jest jeszcze jeden błąd trudny do zauważenia. Przez instrukcję skoku działanie pętli jest przerywane przed końcem. Powoduje to pozostawienie parametrów pętli na stosie. Przy większej liczbie takich przypadków może zabraknąć pamięci na stos. Aby tego uniknąć należy w przypadku opuszczania pętli przed jej zakończeniem użyć instrukcji POP. Dotyczy to także wyjścia z procedur przez GOTO zamiast przez RETURN.

Autor skarży się na „nadmierną” dokładność komputera, która uniemożliwia wprowadzenie dzielenia. Można to ominąć w bardzo prosty sposób wykorzystując mnożenie i zamianę wartości zmiennych (linia 330 w drugim wydruku).

Drugi kłopot to występowanie błędu w instrukcji INPUT. Jeżeli wprowadzamy ciąg znaków (zmienną tekstową), to samo naciśnięcie RETURN daje ciąg pusty — mający zero elementów. W przypadku wprowadzania liczby nie istnieje liczba mająca zero cyfr — nawet 0 ma jedną cyfrę. Atari Basic ma jednak bardzo pożyteczną instrukcję TRAP, która służy do wyłapywania błędów. Po wystąpieniu błędu następuje skok do linii podanej po instrukcji TRAP. Po każdym błędzie musi ona być odnawiana, gdyż działa tylko jednorazowo. Użycie liczby większej od dopuszczalnego numeru linii (np. 40000) wyłącza TRAP.

Pozostałe poprawki dotyczą już tylko drobiazgow. W instrukcji GRAPHICS lepiej używać obliczonych wartości niż wyrażań. Oszczędzamy w ten sposób 7 bajtów pamięci. To samo dotyczy instrukcji STEP 1, którą można całkowicie pominąć.

Oszczędność pamięci uzyskuje się także przez umieszczenie kilku instrukcji w jednej linii. Wpływa to jednak negatywnie na przejrzystość programu, ale niestety nie ma róży bez kolców.

Oto program po dokonaniu wszystkich poprawek (patrz listing 2).

Przy analizie programu widać wyraźnie zmiany, które nie były omawiane. Dotyczą one treści merytorycznej. Przede wszystkim program musi dostosowywać się do użytkownika, a nie odwrotnie. Dlatego nie należy zakładać określonego czasu na planszę tytułową i instrukcję. Czas ich wyświetlania powinien być regulowany przez użytkownika. Aby nie musiał się on domyślać, jak przejść do dalszej części, trzeba informację umieścić na ekranie. Samo rozmieszczenie informacji powinno też ułatwiać jej odbiór.

Następna uwaga dotyczy wykorzystywania możliwości graficznych komputera. Jak widać uzyskanie różnokolorowych napisów jest bardzo proste. Równie łatwe jest spowodowanie migania jednego tekstu, gdy inne wyświetlane są stale. Tymi sposobami można zwrócić uwagę użytkownika na istotne informacje.

Nie chcę się tu wypowiadać na temat wartości praktycznej zastosowanego sposobu nauki matematyki, lecz jedna rzecz narzuca się każdemu. O ile dodawanie i odejmowanie w zakresie do 100 jest stosunkowo proste, to mnożenie i dzielenie w tym zakresie wymaga bardzo dobrego opanowania tabliczki mnożenia lub... kalkulatora. Dlatego ograniczyłem zakres mnożenia i dzielenia do liczb od 1 do 20 (linie 180 i 190). Jeżeli komuś to nie odpowiada, wystarczy zmienić liczbę 8 na inną.

Ostatnia uwaga dotyczy używania w programach znaków i napisów obcojęzycznych. Osobiście stosuję to bardzo często, bo znika wtedy problem „ogonków” w polskich literach. Trzeba jednak choć trochę znać ten język. A co czytamy (po przetłumaczeniu na polski) na początku nadesłanego programu? „Rafał i Janiak prezentuje”. Czy to znaczy, że autorów jest dwóch?

Wojciech Zientara

LISTING 2

```
10 GRAPHICS 18
20 POSITION 4,4: ? #6;"rafał janiak":PO
SITION 5,6: ? #6;"PREZENTUJE"
30 GOSUB 280:GRAPHICS 18
40 POSITION 1,5: ? #6;"M A T E M A T Y
K A":POSITION 3,11: ? #6;"NACISNIJ
START"
50 GOSUB 290:SETCOLOR 0,0,0
60 GOSUB 290:SETCOLOR 0,2,8
70 IF PEEK(53279)<>6 THEN 140
80 GRAPHICS 17:POSITION 5,2
90 ? #6;"instrukcja":? #6: ? #6;"PO ZAD
ANIU PYTANIA":? #6;"PRZEZ KOMPUTER":?
#6;"NALEZY NAPISAC WYNIK";
100 ? #6;"I NACISNAC RETURN,":? #6: ? #
6;"A KOMPUTER":? #6;"POINFORMUJE O JEG
O":? #6;"POPRAWNOSCI."
110 POSITION 3,14: ? #6;"zyczymy":POSIT
ION 6,15: ? #6;"przyjemnej":POSITION 9,
16: ? #6;"zabawy"
120 POSITION 3,21: ? #6;"NACISNIJ STAR
T"
130 GOSUB 290:DIM E$(4):E$="+-*:"
140 IF PEEK(53279)<>6 THEN 140
150 FOR E=1 TO 4:GRAPHICS 18:POSITION
6,5: ? #6;"ETAP ";E
160 GOSUB 280:GRAPHICS 0: ?
170 FOR A=1 TO 10
180 X=INT(10*(10-8*(E>2))*RND(0))+1
190 Y=INT(10*(10-8*(E>2))*RND(0))+1
200 ON E GOSUB 300,310,320,330: ?
210 TRAP 210: ? X;E$(E);Y;"="";:INPUT
Z:TRAP 40000: ?
220 IF Z<>W THEN ? "ZACZYNAJ OD NOWA.
ZLE":POP :GOTO 170
230 ? "DOBRZE":NEXT A:GOSUB 290
240 NEXT E:GRAPHICS 18
250 POSITION 1,3: ? #6;"BRAWO !":? #6: "
KONCZYSZ":? #6: " Z WYNIKIEM":? #6:
" POZYTYWNYM"
260 GOSUB 280:GRAPHICS 18:POSITION 7,5
: ? #6;"KONIEC":GOSUB 280:END
270 REM *** PROCEDURY ***
280 FOR A=1 TO 1000:NEXT A:RETURN
290 FOR A=1 TO 300:NEXT A:RETURN
300 W=X+Y:RETURN
310 W=X-Y:RETURN
320 W=X*Y:RETURN
330 W=X:X*X*Y:RETURN
```




Jednym z najpopularniejszych programów graficznych jest „Fun With Art”. Pozwala on na wykonywanie rysunków w trybie GRAPHICS 15 (160 na 192 punkty), a dzięki wykorzystaniu przerwań umożliwia uzyskanie na ekranie praktycznie wszystkich kolorów.

MENU

Po uruchomieniu programu na ekranie pojawia się rysunkowe menu. Opcje wybierane są z niego joystickiem przez nrowadzenie krzyżyka na właściwy rysunek i wciśnięcie przycisku. Potwierdzeniem wyboru jest ukazujący się w dolnej części menu wykaz aktualnie wybranych opcji. W górnym wierszu tego wykazu wymieniona jest opcja wskazywana przez kursor.

Dodatkową możliwością — dla wprawnych użytkowników — jest bezpośrednie wybieranie opcji za pomocą klawiszy literowych. Nie we wszystkich wypadkach jest to jednoznaczne i radzę przed narysowaniem czegokolwiek sprawdzić dokonany wybór.

Wykaz wybranych opcji zawiera trzy pozycje:

Using (użyte) — określa wybrany sposób rysowania lub inne funkcje;

Fill (wypełnianie) — określa sposób wypełniania, przy czym „Off” oznacza brak wypełniania;

Width (szerokość) — podaje aktualną szerokość pędzla.

Przejdzie z menu do rysunku lub odwrotnie następuje po każdorazowym naciśnięciu klawisza START.

WARIANTY RYSOWANIA

Dostępne warianty rysowania zostaną opisane w kolejności, w jakiej znajdują się w menu: kolumnami od lewej do pra-

wej i z góry na dół w poszczególnych kolumnach. Na rysunku operuje się kurorem mającym postać niewielkiego krzyża. Niestety w przeciwieństwie do wielu innych programów, przy rysowaniu figur nie pojawia się na ekranie ich przewidywany zarys, co osobom o małej wprawie utrudni wykonanie zaplanowanego obrazka.

Draw mode (klawisz D) — rysowanie: poruszenia joysticka powodują przesuwanie się po ekranie kursora. Przy wciśnięciu przycisku kursor pozostawia ślad „odręcznej” linii. Jeśli kursor jest nieruchomy, jego śladem jest po prostu punkt.

Outline mode (O) — rysowanie odcinków od ostatniego punktu: pierwsze wciśnięcie przycisku powoduje wyznaczenie początku odcinka i pozostawienie w nim kopii kursora. Każde kolejne naciśnięcie przycisku rysuje odcinek od poprzedniego do aktualnego położenia kursora.

Line mode (L) — rysowanie odcinków: pierwsze wciśnięcie przycisku wyznacza początek odcinka, drugie jego koniec i powoduje narysowanie całego odcinka.

Box mode (B) — rysowanie prostokątów: kolejne naciśnięcia przycisku wyznaczają położenie przeciwległych wierzchołków prostokąta (czyli przekątną). Prostokąt jest rysowany po drugim wciśnięciu przycisku, przy czym jego boki są zawsze równoległe do krawędzi ekranu.

Circle mode (C) — rysowanie okręgów: pierwsze naciśnięcie przycisku wyznacza położenie środka rysowanego

FUN WITH ART

okręgu; drugie określa jego promień i powoduje rysowanie.

Large Small Text (T) — pisanie tekstu dużymi lub małymi literami. Mają one rozmiary liter uzyskiwanych odpowiednio w trybach GRAPHICS 2 i 1. Miejsce wpisywania liter określa się joystickiem, zaś samo ich wpisywanie odbywa się z klawiatury.

OPEROWANIE BLOKAMI

Kolejna grupa opcji dotyczy operacji na większych fragmentach rysunku zwanych blokami. Blok wyznacza się trzema naciśnięciami przycisku. Pierwsze dwa działają jak przy rysunku prostokąta, lecz powodują pojawienie się ramki wyznaczonej czterema narożnikami, którą można dowolnie przesuwać. Dopiero trzecie naciśnięcie ustala położenie ramki i wykonanie wybranej operacji (czasem

potrzebne jest jeszcze czwarte naciśnięcie → zobacz niżej).

Block transfer (X) — przeniesienie bloku. Po wybraniu bloku do przeniesienia przesuwamy ramkę na wybrane miejsce i wciskamy przycisk. Wskazany blok usuwany jest z poprzedniego miejsca i umieszczany na nowym, wyznaczonym przez nas.

Block copy (Y) — kopiowanie bloku: jest to funkcja podobna do poprzedniej, z tą różnicą, że przemieszczany blok nie jest usuwany z poprzedniego miejsca. Pozwala to na łatwe kopiowanie powtarzających się elementów rysunku.

Block upend (U) — kopiowanie bloku z odwróceniem: w tym wypadku skopiowany blok jest umieszczany w nowym miejscu „do góry nogami”. W ten sposób łatwo można osiągnąć efekt odbicia w wodzie.

Block mirror (M) — kopiowanie z lustrzanym odbiciem: skopiowany blok jest odwracany jak w poprzednim wariantcie, ale tym razem wokół osi pionowej. Zamienia to miejscami lewą i prawą stronę bloku.

Block zoom (Z) — powiększenie bloku: naciśnięcie przycisku powoduje powiększenie wybranego bloku na cały ekran, co umożliwia dokonanie dokładnych poprawek. Na powiększonym bloku możliwe jest rysowanie wyłącznie w trybie **Draw Mode**. Powrót do całości rysunku następuje po naciśnięciu klawisza ESC.

Block erase (E) — kasowanie bloku: powoduje usunięcie wybranego fragmentu rysunku. Należy uważać przy wybieraniu tego wariantu, gdyż komputer nie sprawdza, czy nie nastąpiła pomyłka i możemy nieodwracalnie stracić część pracy.

WYPEŁNIANIE

Ważną możliwością każdego programu graficznego jest wypełnianie. Pozwala ono na szybkie zamalowanie wybranego obszaru jednym kolorem. W odróżnieniu od innych programów graficznych „Fun With Art” nie ma funkcji wypełniania zamkniętego obszaru. Dysponuje natomiast czterema innymi funkcjami: Fill up (wypełnianie w górę), Fill down (w dół), Fill right (w prawo) i Fill left (w lewo).

Działanie ich jest następujące: po narysowaniu każdego punktu aktualnie tworzonej linii (draw, outline lub line) we wskazanym kierunku wypełniania prowadzona jest linia, aż do osiągnięcia punktu o wyższym kolorze. Dopiero wtedy rysowany jest następny punkt linii i cała procedura się powtarza. Znaczenie wysokości koloru będzie wyjaśnione później.

Niezależnie od wybranego sposobu wypełniania (oprócz „Off”) podczas rysowania prostokąta lub okręgu zawsze zostaje wypełnione tylko jego wnętrze, a wypełnianie jest wykonywane w dół.

Wyłączenie wypełniania następuje po wybraniu opcji Fill off (koniec wypełniania).

ZAPIS I ODCZYT

Każdy rysunek po wykonaniu można zapisać do późniejszego wykorzystania

lub wydrukowania. Można także wczytać wykonany rysunek celem dokonania poprawek. „Fun With Art” jako jeden z nielicznych programów graficznych pozwala na zapisanie i odczytanie fragmentu rysunku. Wszystkie te funkcje są wykonywane przez zestaw operacji wejścia / wyjścia wybieranych z menu.

Disk Directory — odczyt spisu zawartości dyskietki: po wskazaniu tej funkcji i naciśnięciu przycisku automatycznie odczytywany jest spis zawartości dyskietki umieszczonej w stacji dysków numer 1. Pozycje spisu są wyświetlane po trzy w dolnej części menu. Wyświetlenie następnej grupy następuje po naciśnięciu dowolnego klawisza.

Save Picture — zapis rysunku: po wybraniu tej opcji pojawiają się na dole napisy „Current Name is:...” i „Use this Name (Y/N)?” („Aktualną nazwą jest:...” i „Użyć tej nazwy (tak/nie)”). Naciśnięcie klawisza „N” powoduje zapisanie rysunku pod wyświetloną nazwą. Umożliwia to proste zapisywanie rysunku podczas tworzenia go. Gdy chcemy podać inną nazwę, naciskamy klawisz „Y” i pojawia się pytanie „Enter New Name:” („Podaj nową nazwę:”). Wpisujemy teraz pełną nazwę rysunku do zapisania — dla stacji dysków „D: nazwa”, a dla magnetofonu „C:”. Zapis następuje po naciśnięciu „RETURN”.

Load Picture — odczyt rysunku: najpierw pojawiają się pytania jak przy zapisie. Po naciśnięciu „N” wczytywany jest rysunek o poprzedniej nazwie, a po „Y” należy podać nową nazwę rysunku.

Block Write — zapis bloku: pierwszą czynnością jest ustalenie rozmiaru bloku do zapisania. Wykonuje się to w taki sposób jak przy innych operacjach na fragmentach rysunku. Po ustaleniu bloku wskazany fragment jest zapisywany — dalsze czynności są analogiczne jak przy zapisie całego rysunku.

Block Read — odczyt bloku: po wskazaniu nazwy bloku do odczytu komputer odczytuje z zapisanego pliku rozmiar bloku. Następnie na ekranie pojawia się ramka, którą należy umieścić w miejscu, gdzie chcemy wczytać blok. Po wciśnięciu przycisku blok jest wczytywany bezpośrednio do pamięci obrazu.

Load Colorform — odczyt formy koloru: opcja ta umożliwia odczyt z normalnego pliku rysunkowego tylko informacji o kolorach. Postępowanie jest analogiczne jak przy odczycie całego rysunku.

WYBÓR KOLORU

„Fun With Art” należy do programów o największych możliwościach kolorystycznych. Przy umiejętnym operowaniu kolorami można uzyskać jednocześnie na ekranie dowolną ich liczbę — nawet wszystkie 256 kolorów naraz.

Pick Color (spacja) — wybór koloru: aktualnie wybrany kolor (jeden z czterech) jest wskazywany przez pędzel umieszczony nad jego polem. Używany kolor można zmieniać przez naciskanie spacji lub przez wskazanie koloru kursorem i wciśnięcie przycisku. Zmianę barwy używanego koloru uzyskuje się przez naciskanie klawisza „SELECT”, a zmianę jego jasności przez „OPTION”.

Zmiany te dotyczą wszystkich miejsc narysowanych danym kolorem.

Color Height — priorytet koloru: obok paska każdego koloru znajduje się niewielki słupek wskazujący priorytet (wysokość) tego koloru. Czym słupek jest wyższy, tym większy jest priorytet koloru. Przez ustawienie kursora nad słupkiem i naciskanie przycisku można ten priorytet zmieniać. Wykorzystywane jest to głównie przy wypełnianiu. Jeżeli podczas wypełniania napotkany zostanie kolor o wyższym priorytecie, to wypełnianie jest w tym miejscu przerywane. Gdy napotkany kolor ma niższy priorytet, to wypełnianie jest kontynuowane tak, jakby nie było tam żadnego koloru.

ZMIANY KOLORÓW

Oprócz wybierania i zmiany kolorów na całym rysunku możliwa jest także zmiana jedynie fragmentów obrazu. Mamy tu dwa rodzaje zmian: zastąpienie koloru innym i zmianę barwy.

Alter Current Color (A) — zmiana aktualnego koloru: opcja ta nie jest dostępna z menu, lecz wyłącznie z klawiatury podczas rysowania. Po ustawieniu kursora w miejscu, od którego chcemy zmienić barwę, naciskamy klawisz „A”. W miejscu kursora pojawia się strzałka skierowana w dół. Teraz klawiszami „SELECT” i „OPTION” ustalamy barwę i jasność, jaką będzie miał wybrany kolor od tego miejsca do dolnej krawędzi rysunku. Naciśnięcie „START” zatwierdza dokonaną zmianę. Stosując tę opcję kilka razy otrzymujemy na ekranie poziome pasy, w których wskazany kolor ma różne barwy. Na przykład można ustalić barwę tła (kolor 0) błękitną, od połowy rysunku zielną, następnie żółtą i szmaragdową. Dzięki temu przy użyciu tylko jednego koloru otrzymujemy niebo, trawę, piasek i wodę.

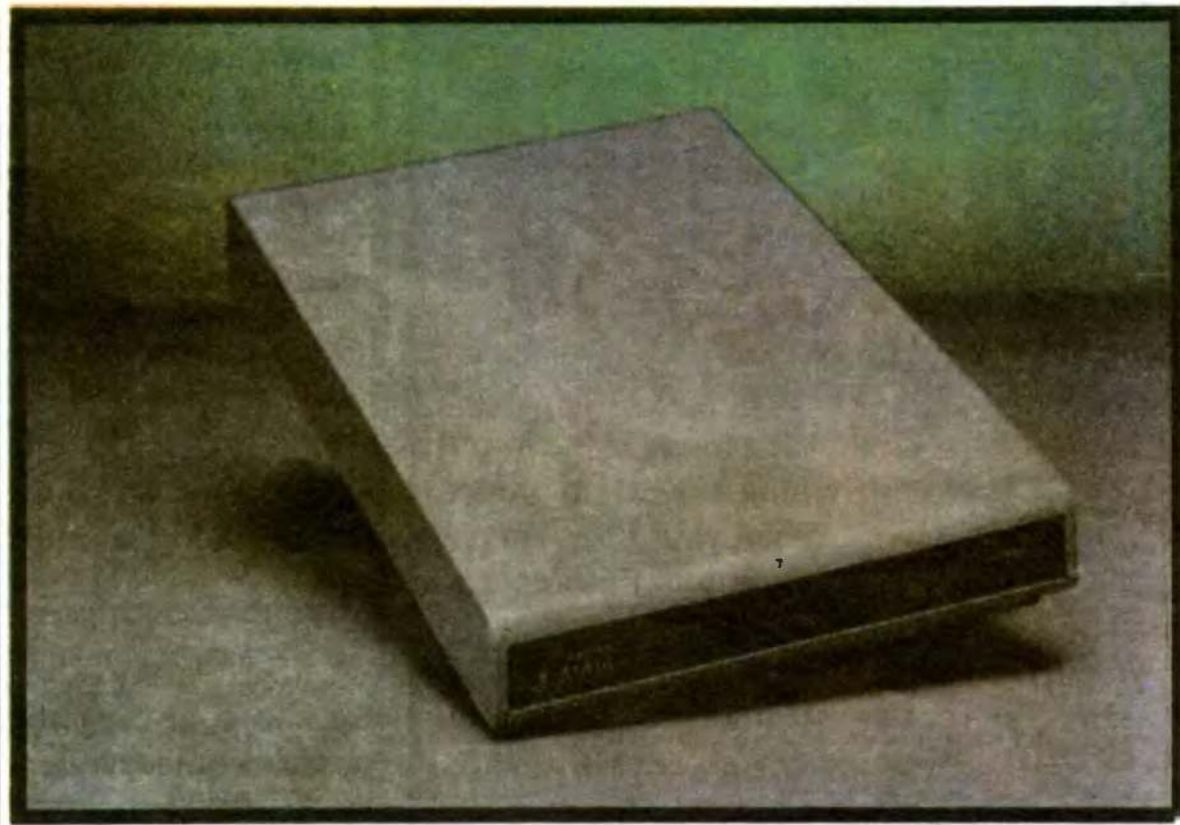
Block Color Swap — wymiana koloru w bloku: po wybraniu tej opcji pojawia się polecenie „Please Point and Fire at the Color to be Changed” („Proszę wskazać kolor, który ma być zmieniony, i nacisnąć przycisk”). Po wykonaniu polecenia pojawia się drugie — „Please Point and Fire at the Color to Swap to!” („Proszę wskazać kolor do wymiany i nacisnąć przycisk”). Teraz trzeba w opisany wcześniej sposób wyznaczyć na rysunku fragment do wymiany. We wskazanym bloku wszystkie punkty pierwszego koloru zostaną zamienione na drugi kolor.

INNE FUNKCJE

Erase Picture — kasowanie rysunku: całkowicie usuwa rysunek z pamięci komputera. Aby uniknąć przypadkowego zniszczenia wykonanej pracy komputer pyta: „Erase Pic — Are You Sure (Y/N)?” („Kasowanie rysunku — jesteś pewny (tak/nie)?”). Naciśnięcie klawisza „N” powoduje powrót do menu, a naciśnięcie „Y” kasuje rysunek.

Stop F.W.A. — koniec „Fun With Art”: skasowanie całego programu. Tu także komputer upewnia się „Quit FWA — Are You Sure (Y/N)?”. Naciśnięcie „Y” usuwa program z pamięci komputera, a „N” powoduje powrót do menu.

Marek Zachar



CUDOWNA SKRZYŃKA

Zakłady Atari na Tajwanie produkują 80-kolumnową przystawkę do Atari XL/XE o nazwie XEP80. Głównym projektantem tego systemu jest Lane Winner.

XEP80 wyświetla na ekranie 80 kolumn i 24 wiersze tekstu. Na monitorze monochromatycznym tekst ten jest bardzo wyraźny. Jest on także czytelny na monitorze kolorowym, choć oczywiście litery są mniejsze niż w standardowym 40-kolumnowym trybie Atari. W każdym razie XEP80 jest znacznie lepszym produktem dającym 80-kolumnowy obraz niż rozwiązania czysto programowe.

Oprogramowanie

Jak twierdzi Dyrektor Atari ds. Oprogramowania **John Skruch** w końcowej fazie opracowania jest Atari Writer 80 — nowa wersja programu AtariWriter+ i powinien się on pokazać na rynku w czerwcu 1987 roku.

AtariWriter80 i nowa wersja Silent Butler (program kalkulacyjny) są pierwszymi programami działającymi z XEP80. Jednakże prototypowe wersje tego urządzenia były rozesłane do najważniejszych producentów oprogramowania — takich jak Batteries Included, Broderbund, XLEnt i OSS — w celu przerobienia istniejących programów na 80 kolumn.

Użycie XEP80

XEP80 ma wymiary zbliżone do modemu 1030 (136x235x35 mm) i wagę około 1 kg. Dobrze pasuje do umieszczenia na stacji dysków.

Urządzenie jest dołączane do komputera przez jeden z portów joysticków (dowolny). Przewód przenoszący sygnał do monitora jest dołączany z tyłu XEP80 (wg Atari obraz w odbiorniku telewizyjnym jest niezadowolający).

System jest zasilany z niewielkiego 9-woltowego zasilacza. Niestety pierwsze egzemplarze zasilaczy miały tendencję do nadmiernego grzania się podczas pracy.

Dodatkowo zainstalowany jest również port równoległy drukarki, który używa takiego samego 25-stykowego złącza jak ST. Jeżeli przy włączaniu komputera trzyma się wciśnięty klawisz SELECT, to XEP80 obsługuje tylko port drukarki bez włączania 80-kolumnowego obrazu.

Programy użytkowe i demonstracyjne

Dyskietka dołączana do XEP80 zawiera plik AUTORUN SYS, który instaluje handler mający tylko około 200 bajtów. Dodatkowo jest zapisany plik źródłowy handlera w formacie MAC65.

Na dyskietce znajduje się także kilka imponujących programów demonstracyjnych napisanych w BASIC i języku maszynowym, jak również szczegółowa dokumentacja i programy użytkowe do dołączenia 80-kolumnowego handlera do własnych programów.

Handler XEP80 dodaje do Atari BASIC kilka nowych rozkazów. Mają one formę instrukcji XIO i umożliwiają:

- * zmianę kolorów tekstu i tła (standardowo jest biały tekst i czarne tło);
- * włączenie podkreślenia;
- * włączenie błyskania kursora;
- * mieszanie standardowego tekstu z tekstem o podwójnej szerokości lub wysokości;
- * mieszanie standardowego tekstu z błyskającym (o dowolnej szerokości i wysokości);
- * poziomy przesuw znak po znaku za pomocą instrukcji POSITION i XIO.

XEP80 jest bezpośrednio kompatybilny z całym oprogramowaniem wykorzystującym urządzenie „E:” — jak Atari BASIC (wersja A, B i C) i DOS 2.5. Podczas testów XEP80 nie działał z DOS-em 2.0.

Grafika

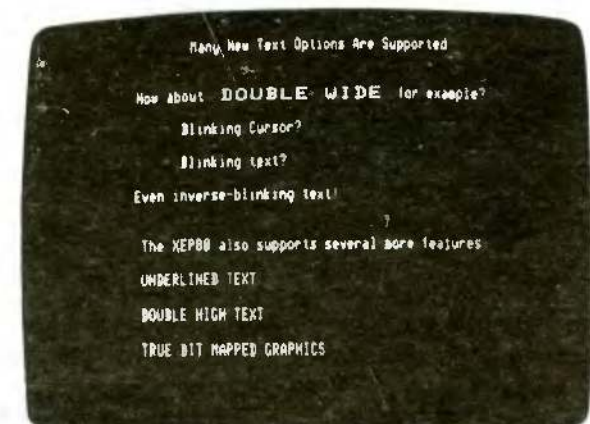
Do urządzenia jest wbudowana pamięć statyczna RAM o pojemności 8 KB, która jest używana jako pamięć obrazu w celu przyspieszenia pracy. XEP80 ma zainstalowane dwa zestawy znaków: standardowy zestaw XL/XE i zestaw międzynarodowy Atari.

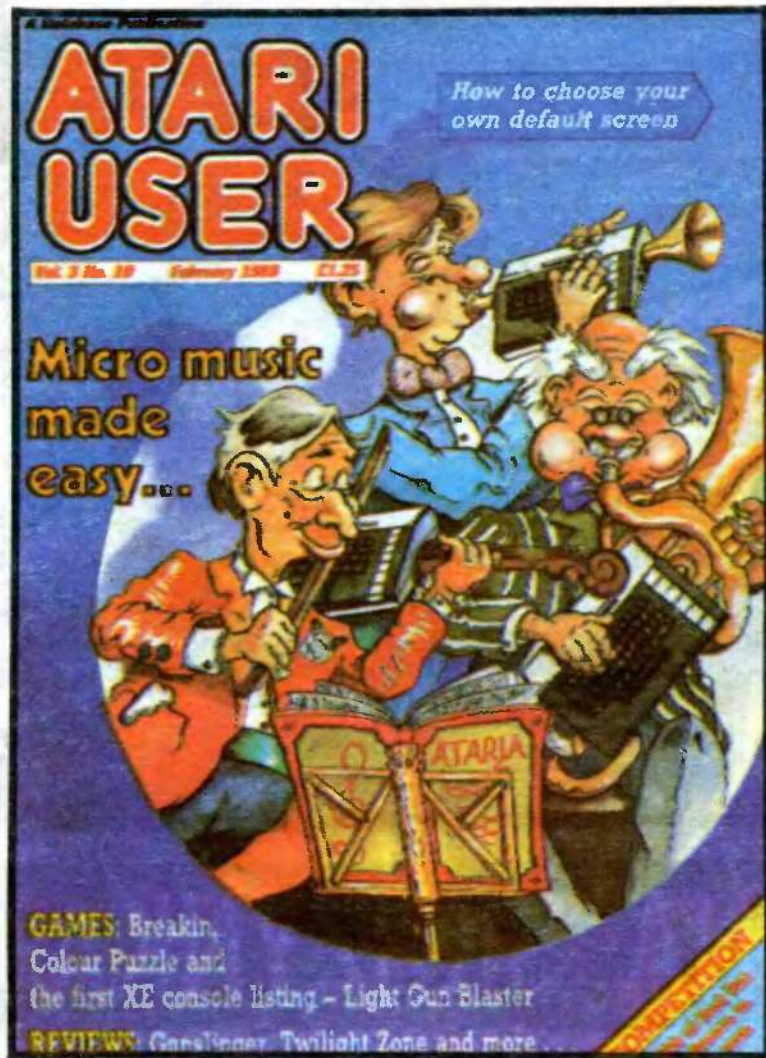
XEP80 może określić grafikę wysokiej rozdzielczości pokrywającą połowę ekranu. Jednakże procedury kreślenia 80-kolumnowego są wolniejsze niż standardowe 40-kolumnowe. Kreślenie i wypełnianie okręgu o średnicy 6 cm trwa około 5 minut.

Samo rysowanie nie jest łatwe. Instrukcje PLOT i DRAWTO nie działają i niedozwolone jest okno tekstowe. Jeżeli program „padnie” w środku jednej z tych długich i skomplikowanych operacji, to obraz powraca do trybu bitowego. Należy ponownie uruchomić handler i rozpocząć od nowa.

Jeżeli poważnie myślisz o 80-kolumnowym obrazie, to XEP80 nie zawiedzie Cię. Tekst jest znakomity na monitorze monochromatycznym i zadowalający na kolorowym. Początkujący programiści chętnie wykorzystują nowe sposoby użycia instrukcji XIO. Programiści zaawansowani z pewnością zaadaptują handler XEP80 do swojego ulubionego oprogramowania, głównie edytorów tekstu, programów kalkulacyjnych i telekomunikacyjnych.

opracowane wg „Antic” 7/87





Czytelnicy „Bajtka Atari” z pewnością pozazdroszczą trochę swoim brytyjskim kolegom, którzy już od dawna mają własne pismo. Miesięcznik „Atari User”, czyli „Użytkownik Atari”, informuje o wszystkich ważniejszych wydarzeniach oraz nowościach oprogramowania, które mogą zainteresować posiadacza Atari.

Czasopismo najwięcej uwagi poświęca odbiorcom traktującym komputer, jako narzędzie twórczej rozrywki intelektualnej. Na 64 stronach zdecydowanie dominują wydruki gier, programów muzycznych, graficznych i użytkowych oraz procedur ułatwiających rozwiązywanie różnych problemów. Do wspólnej rywalizacji w programowaniu zachęca stały konkurs na 5-liniowy program. Najciekawsze propozycje są publikowane, a autorzy otrzymują nagrodę 25 funtów. Niektóre z nadsyłanych rozwiązań są bardzo pomysłowe i zaskakujące. Dla nowicjuszy przeznaczona jest seria artykułów edukacyjnych, które uczą efektywnego korzystania z Atari Basica i innych języków.

Sporą grupę czytelników „Atari User” stanowią miłośnicy zabawy z joystickiem. Świadczą o tym zamieszczone w piśmie liczne opisy gier oraz listy do redakcji. Brytyjscy gracze mają podobne problemy, jak nasi

czytelnicy. Publikowane opisy zawierają informacje ułatwiające przebrnięcie kolejnego etapu gry oraz porady, jak unikać czyhających w nich pułapek. Odpowiednikiem naszej listy przebojów jest publikowana w piśmie karta sondażowa Instytutu Gallupa. Trzy najpopularniejsze w lutym gry to: „Decathlon”, „Feud” oraz „Henry’s House”, która ma szansę w kolejnych miesiącach wspiąć się na najwyższą pozycję.

Najbardziej zacięła mnie informacja o możliwości podłączenia Atari za pomocą modemu do sieci komputerowej MicroLink. Wydatek rzędu 140 funtów daje ogromne możliwości: korzystanie z poczty elektronicznej, zamianę komputera na telex, dokonywanie rezerwacji biletów lotniczych, kolejowych lub teatralnych, dostęp do ogromnej biblioteki oprogramowania, baz danych w Europie i USA, informację o 100000 ważniejszych brytyjskich przedsiębiorstwach, a także wielojęzyczny słownik zawierający 380000 różnych słów i terminów. Za pomocą Micro Link są również rozpowszechniane niektóre publikowane w piśmie gry i programy oraz ukazujące się w nim ogłoszenia czytelników. „Atari User” pełni w ten sposób rolę skrzynki kontaktowej dla osób, które pragną tanio sprzedać lub zakupić sprzęt i oprogramowanie. Zestaw Atari 130 XE, stacja dysków 1050, drukarka 1027, joystick i magnetofon były osiągalne zaledwie za 200 £. Jedyńm warunkiem stawianym przez redakcję uczestnikowi tej swistej giełdy jest wymiana tylko oryginalnych programów. Jest to oczywiście przejawem troski o twórców oprogramowania i wrogiemu nastawieniu do piractwa.

Czytelnicy lutowego numeru „Atari User” mieli także okazję zamówić bilety na odbywającą się w Londynie w dniach 22 — 24 kwietnia br. ogromną wystawę sprzętu i oprogramowania Atari. Wśród eksponatów znalazły się komputery 8-bitowe oraz najnowsze, rewelacyjne Mega ST.

Do ciekawszych rubryk zaliczyłbym także systematycznie pojawiający się w kolejnych numerach tzw. „Gadget”. Jest to przeznaczony dla majsterkowiczów opis prostego urządzenia, które czasami może być bardzo użyteczne, np. interfejs do drukarki, blokada zasilania uniemożliwiająca korzystanie z komputera osobom nie znającym cyfrowego hasła, joysticki dodatkowo reagujący na dźwięk.

„Atari User” jest w sumie bardzo interesującym, estetycznie wydanym pismem, które na pewno dobrze spełnia oczekiwania użytkownika 8-bitowego komputera Atari.

(j.j.)

Miesięcznik „Atari User”, Wydawca: Database Publications Ltd, Londyn, 64 str. cena 1,25 £.



Pascal był kilkakrotnie prezentowany w „Bajtku” w rubryce „Programować może każdy”. O jego zaletach przekonali się już na pewno ci czytelnicy, którzy z uwagą przeczytali artykuły „Bez wyboru” (nr 9—11/87), „Pascal” (nr 10/86—1/87) czy „Turbo PASCAL” (nr 6—7/87). Znakomitą pomocą ułatwiającą poznanie wszystkich tajemnic Pascala jest „Programowanie w języku Pascal na mikrokomputerze Atari” Bogdana Celer. Książka ta opisuje implementację Pascala opracowaną przez firmę Kyan Sof-

tware, przeznaczoną dla użytkowników Atari 800 XL i Atari 130 XE.

Podręcznik ten jest wręcz wymarzoną lekturą dla hobbyisty zamierzającego od podstaw poznać arkaną sztukę programowania w Pascalu. W pierwszej części zaznajamia nas z zasadami użytkownika Kyan Pascala. Metoda praktycznej obserwacji reakcji komputera na rozkazy wydawane zgodnie z zawartymi w książce wskazówkami sprawdziła się już nie raz. Dzięki niej w krótkim czasie możemy opanować reguły posługiwania się edytorem programów źródłowych. Umiejętne przeplatanie praktycznych ćwiczeń z teorią sprawia, że z wielkim zainteresowaniem czytamy krótki wykład o metodzie strukturalnego programowania. W miarę potrzeb wprowadzane są kolejno nowe pojęcia i objaśniane podstawowe konstrukcje języka. Nie zawsze jasna dla laika nomenklatura informatyczna jest zawsze komentowana, co umożliwia szybkie zrozumienie omawianych zagadnień.

Łatwo przy tym dostrzec różnicę pomiędzy realizacją tego samego algorytmu w Basicu i Pascalu. Duża liczba umieszczonych w książce programów będzie dobrą pomocą w samodzielnym programowaniu, a także pozwoli czytelnikowi skontrolować, czy w dostatecznym stopniu zrozumiał wykładane wiadomości. Końcowe rozdziały mają na celu wyeksponowanie dodatkowych możliwości Kyan Pascala: instrukcji skoku, operacji graficznych i na łańcuchach znakowych, stosowania asemblera, wykorzystania pamięci i posługiwania się drukarką w programie.

Po przeczytaniu książki doszedłem do wniosku, że może ona być przydatna nie tylko dla użytkowników Atari i nie tylko dla programujących w Pascalu. Decyduje o tym przede wszystkim duża przystępność. Czytelnik innych książek (np. „Pascal” M. Iglewskiego, J. Madeya i S. Matwina) na pewno chętnie do niej sięgnie, by znaleźć dodatkowe wyjaśnienia i informacje. Poza tym z konstrukcji charakterystycznych dla Pascala często korzysta się w innych językach programowania. Twórcy ulepszonych wersji już znanych języków (np. Basic) oraz zupełnie nowych często wzorują się na koncepcji Pascala, którą stworzył Niklaus Wirth.

(j.j.)

Bogdan Celer, „Programowanie w języku Pascal na mikrokomputerze Atari”, Naczelna Organizacja Techniczna, Ośrodek Doskonalenia Kadr Technicznych Rady Stołecznej-Forum Atari, Warszawa 1987, Wyd. I, cena 900,-



Nasza redakcja otrzymuje wiele listów z pytaniami o zgodność komputerów Atari 800XL, 65XE i 130 XE. Wyjaśniam więc, że komputery 800XL i 65XE różnią się TYLKO obudową — wewnętrzna konstrukcja pozostała bez zmian. Wszystkie programy napisane dla 800XL działają poprawnie na 65XE i odwrotnie.

Nieco inaczej przedstawia się sprawa modelu 130XE. Posiada on dodatkową pamięć RAM o pojemności 64KB i jest to jedyna wprowadzona zmiana. Większość programów przeznaczonych dla 800XL i 65XE powinna działać bez żadnych problemów. Jednak niektóre programy, w niewłaściwy sposób gospodarujące pamięcią, powodują zawieszanie się 130 XE. Są to m.in. „Bruce Lee” i „Pole Position II”. Istnieją także programy specjalnie napisane dla 130XE — przeważnie są to programy użytkowe. Nie działają one na 800XL i 65 XE z powodu zbyt małego obszaru pamięci dostępnego w tych modelach.

Proszę o wyjaśnienie dwóch spraw:

1. Dlaczego Romek redagując rubrykę „Tylko dla przedszkolaków” zapomina, że Atari nie reaguje zbyt przychylnie na komendę INKEY\$. Poprawienie tego nie jest może zbyt trudne, ale błąd jest ewidentny.

2. Chciałem wypróbować działanie programu „BFCh 2.0” z numeru 7/87 i przekracza to moje zdolności. Przy zapisie skorygowanego programu komputer podaje „ERROR 9 AT LINE 260”. Poza tym nie zawsze dokonywana zmiana tekstu jest sygnalizowana dźwiękiem i wyświetleniem numeru bajtu.

**Piotr Oktenbreit
ul. Gawędy 66/15
46-020 Czarnowąsy**

Projektanci Atari wiedząc, że komputer będzie często używany do gier, nie wprowadzili do Atari Basic instrukcji INKEY\$. Chodziło o uniknięcie niszczenia klawiatury przy grach, jak to właśnie ma miejsce w Spectrum i Amstradzie. Zamiast tego

ST

B

Z

A

Komputer, który bierze udział w regularnych mistrzostwach świata w szachach, to jeszcze marzenie. Tak się jednak składa, że marzenia z reguły zmieniają się w zamiary i budują życzenia. Od czasu pojawienia się pierwszych bardzo dużych i niezmiernie skomplikowanych komputerów wiele tego typu życzeń zostało zrealizowanych, ale również przybyło wiele nowych. Ludzie nauczyli się dziejów postępu. Można wyciągnąć z tego tylko jeden wniosek, że i my być może doczekamy się „elektronicznych ludzi” współzawodniczących

z umysłem człowieka w walce o mata.

Do takich spotkań zresztą już dochodziło. Nie były to co prawda szachy. Oto w 1979 roku w Monte Carlo Luisi Villa rozegrał partię w BACKGAMMONA przeciwko programowi o nazwie BKG 9,8. Luisi Villa był wtedy koronowanym mistrzem świata. Przegrał 7:1, a twórca programu Hans Berliner otrzymał premię dla zwycięzcy w wysokości 5000 dol.

Szachy w przeciwieństwie do backgammona, halmy czy skata należą do nielicznych gier strategicznych, które nie wpadły jeszcze do końca w „elektroniczne łapy komputera”. Bądź co bądź są one jednym z najbardziej miarodajnych mierników ludzkiego intelektu. Uczą prawidłowej koncentracji, łączenia ze sobą w logiczną całość skutków i przyczyn, a także prowokują do działania intuicję drzemającą w człowieku.

Komputerowe szachy po raz pierwszy pojawiły się w 1978 roku. Od tej daty przybywa coraz więcej nowych odmian tej gry, przeznaczonych dla

komputerów domowych i osobistych. Programy te są z każdym mijającym rokiem lepsze i wykazują swoją przewagę nad elektronicznymi przodkami. Wiąże się to z dążeniem szachowych firm softwarowych do wypuszczania na rynek coraz bardziej doskonałych produktów, a także z coraz lepszymi komputerami, pozwalającymi zbliżyć się do granic ludzkiego poznania.

Jednym z takich komputerów jest Atari serii ST. Od chwili jego pojawienia się w sprzedaży systematycznie powiększa się biblioteka programów szachowych dla tego komputera. Porównanie dostępnego na rynku software'u obejmującego tę dziedzinę powinno przydać się każdemu użytkownikowi ST. Może być także pomocą w wyborze najlepszego w stosunku do potrzeb programu szachowego.

1. KRABAT — program nie jest tak trudny, jak mogłoby się wydawać. Jego obsługa na ekranie pomimo braku DROP-DOWN-MENU jest możliwa przez naciskanie klawiatury. Strategia programu to kompilacja:

a) strategii A — technika „Brute Force” wypracowana przez maszynę nie przewiduje ruchów przypadkowych, komputer w sposób ograniczony wykonuje posunięcie wynikające z wytworzonej na szachownicy sytuacji i

nie sięga dalej, czyli nie przewiduje rozwoju akcji, b) strategii B — technika selektywna, komputer ma do wyboru kilka możliwych wariantów i eliminuje zbędne wg własnej oceny, „głębokość” liczenia, maszyny wynika z jej możliwości hardware'owych.

Istnieje jeszcze nie użyta tu strategia C. Jest to technika „ludzka” — program formułuje myśli i życzenia, a następnie szuka takich rozwiązań, aby pomysłem spowodować wykonanie najlepszego w danej sytuacji ruchu. Chodzi tu o umożliwienie wywołania symulacji ludzkiego sposobu myślenia. Opisaną terminologię stworzył w latach 1949/50 wybitny strateg szachów Claude E. Shannon.

Przebieg gry w KRABACIE można obserwować na ekranie, a poszczególne posunięcia pokazywane są oddzielnie. Program jest dostosowany do gustów różnych użytkowników. Dla początkujących szachistów nie jest za trudny, a dla trochę zaawansowanych kryje w sobie tajemnice, których na pewno nie znają (na przykład całą gamę błędnych otwarć). Wadą KRABATA jest długie oczekiwanie na od-

Studio Kijowianka
Amstrad, Atari XL, XE, ST
Commodore 64, 128
Poleca literaturę i programy na kasetach i dyskach. Warszawa, ul. Targowa 26. Rachunki oraz wysyłka pocztą. Informacje za załączeniem koperty i znaczka. D-64

COMPUTER SERVICE

IBM® PC·XT/AT
KOMPATYBILNE

ZX-Spectrum

Amstrad TIMEX

Schneider Sharp

Reg. Trade Marks of IBM Corporation.

PMS ELEKTRONIK

37-76-65

WARSZAWA

ul. LEGIONOWA 23, ☐01-343



system

**Przedsiębiorstwo Wielobranżowe „System”
działa jako wyłączny przedstawiciel serwisowy firmy:**

MEGA

KURFÜRSTENDAMM 202 1000 BERLIN 15

Tel. 8825641 tlx 182888 MEGA

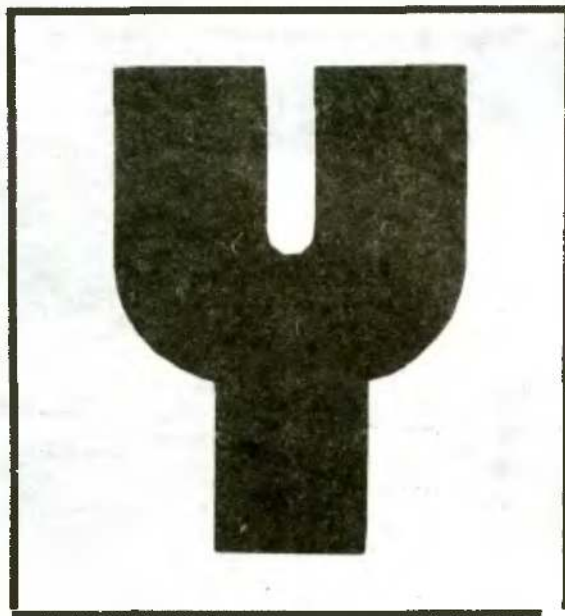
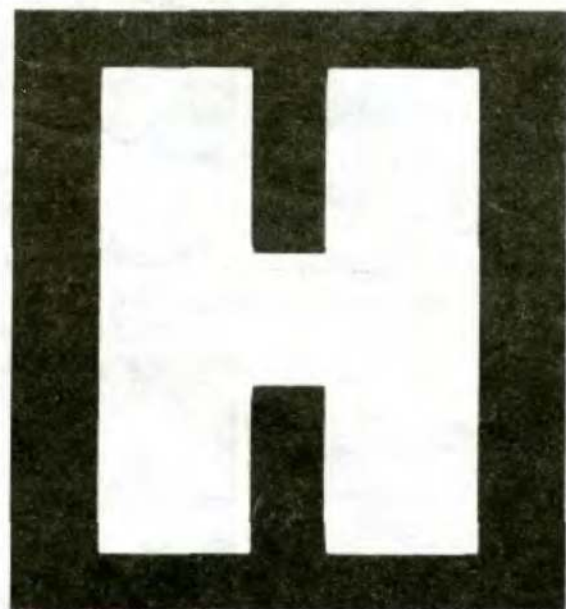
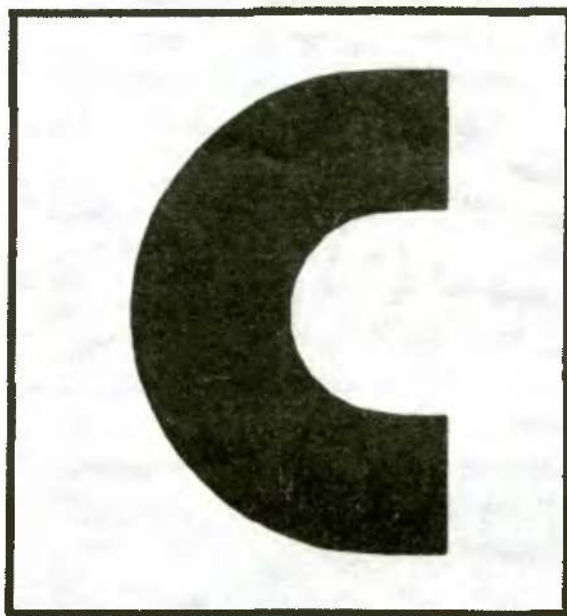
KTÓRA DOSTARCZA SPRZĘT ELEKTRONICZNY WYSOKIEJ KLASY W TYM: PC/XT/AT/RT; 16 I 32 BITOWE, DRUKARKI, PLOTERY, DIGITIZERY, MAGNETOWIDY, DYSKIETKI, TAŚMY BARWIĄCE ITP. INFORMACJĘ TECHNICZNO-HANDLOWĄ (KATALOGI I CENNIKI) MOŻNA UZYSKAĆ W SIEDZIBIE FIRMY:

„SYSTEM”

WARSZAWA, UL. WOLSKA RÓG MŁYNARSKIEJ (PRZEJŚCIE PODZIEMNE OBOK PDT WOLA)

TEL. 32-80-93 tlc 817819 pws pl

K-48



powieź po wykonanym ruchu, nawet wtedy, kiedy można ją znaleźć w każdym podręczniku dla początkujących szachistów. Program nie pozwala na zapis rozegranych partii, jak również interesujących sytuacji na dyskietce oraz czasami nieprawidłowo wykonuje posunięcia.

2. HAMLET (producent Knupe) — stanowi udoskonaloną wersję KRA-BATA. Dysponuje biblioteką różnych otwarć. Każdy gracz może dzięki temu dostosować otwarcie do własnych umiejętności. Także rozegrane partie mogą być zapisywane na dyskietce, co pozwala na prowadzenie archiwum czynionych postępów. Znacznie został skrócony czas odpowiedzi komputera na wykonany ruch. Krótki pisk oznacza gotowość maszyny do przyjęcia kolejnego posunięcia.

3. CHECKMATE (producent Alpha-tron) — to program, który posiada bibliotekę otwarć i budowę obrazu przypomina HAMLETA. Dysponuje jednak możliwością zmiany strategii gry programu z AB na samą B. Daje to użytkownikowi prawo decyzji, czy grać przeciwko programowi taktycznemu czy też pozycyjnemu. Wadą, która „rzuci się w uszy” jest brak sygnału dźwiękowego po zakończonym przez komputer ruchu.

4. DEEP THOUGHT (producent Galactic) — to prawdziwa niespodzianka. Uczestniczenie w grze przeciwko temu programowi jest przyjemnością. Wybór wariantu otwarcia dokonywany jest przed przystąpieniem do samej gry. Program łatwo rozpoznaje dokonywane na początku zmiany i dzięki prostej obsłudze pozwala także dowolnie manipulować biblioteką otwarć. Każdą rozgrywkę można oczywiście zapisać na dyskietce. Dla wszystkich dostępna jest także Chess Base, czyli kilkanaście najlepszych partii szachowych uznanych mistrzów. Wada programu to brak kompletnej i pełnej informacji o możliwych wariantach gry.

5. CHESSMASTER 2000 (producent Atari) — program stworzony został przez jednego z najlepszych znanych programistów szachowych Dave Kittingera. Produkt ten w pełni odpowiada wymaganiom komputerów osobistych. Poziom trudności tych szachów nie jest może tak wysoki jak w PSION CHESS, ale za to użytkownik mile rozczarowuje się już podczas ładowania programu. Najpierw radość tę sprawia piękna plansza tytułowa, a potem głos z komputera, który zapra-

sza do gry. Menu programu oferuje bogaty wybór opcji jego ustawienia. Początkujący szachiści mogą poćwiczyć z wbudowanym trenerem szachów, który pokazuje, jakie ruchy należy wykonać, aby osiągnąć w sposób prawidłowy wytyczony cel. Po każdym posunięciu jest możliwy powrót do:

- a) ruchu wykonanego wcześniej
- b) ruchu wykonanego w danym momencie
- c) ostatnia ewentualność to odpowiedź komputera, jaki ruch jest najlepszy.

Na sposób gry na pewno mają wpływ trzy udostępnione tryby szachowe:

- a) normalny — charakteryzuje się solidną grą i pewnymi otwarciem, w tym przypadku wyłączony jest generator posunięć przypadkowych
- b) solidny — bez udziału otwarć eksperymentalnych, mocna, ostra i wyważona gra
- c) „kawiarniany” — gra bogata w ryzykowne sytuacje, niesolidne i zaskakujące idee otwarć, włączony generator posunięć przypadkowych.

CHESSMASTER oferuje także analizę rozegranych partii. Jeżeli gracz ma taką ochotę, może wybraną przez siebie rozgrywkę pozostawić do zbadania przez komputer nawet na całą noc. Atari ST realizując ten program

wykonuje posunięcia w przyjęty przez siebie sposób postępowania i przy okazji komentuje własne ruchy, wskazując tym samym najlepsze rozwiązania. Program pozwala na rozgrywanie tzw. szybkich partii, a także na zapis dowolnie wybranej sytuacji na szachownicy. Użytkownik, który zetknie się z CHESSMASTER 2000, nie zostanie być może rewelacyjnym graczem w szachy, ale w zamian za to znajdzie prawdziwą frajdę podczas zabawy tym programem.

6. PSION CHESS (producent Atari)

— to bardzo inteligentny, selektywny program dla każdego. Posiada wszystkie zalety opisanych wcześniej elektronicznych szachów i jest dobry pod każdym względem. Pomimo że posiada wiele poziomów trudności i bardzo dużą bibliotekę otwarć z podaniem przewidywanego rozwoju sytuacji, jest również niedrogi. Dodatkowa zaleta to piękna trójwymiarowa grafika z figurami na czarnym tle. Nie jest to oczywiście program dla graczy klasy światowej, bo takie osiągają na software'owym rynku ceny zawrotne.

Na podstawie opisanego oprogramowania o Atari serii ST można bez wątplenia powiedzieć, że jest to doskonała maszyna do gry w szachy. Tania, zapewniająca wszechstronne wykształcenie szachowe, łatwa w obsłudze, wygodna, wytykająca popelnione błędy i pokazująca prawidłowy kierunek rozwoju.

Następnym krokiem na drodze oprogramowania będzie być może stworzenie szachowej bazy wszystkich pięknych partii i wspaniałych rozgrywek najlepszych szachistów świata. ST doskonale się do tego nadaje.

Z pomocą „ST Computer”

Sergiusz Piotrowski

ATARI ZX SPECTRUM
programy, opisy
NISKIE CENY
wysyłka na cały kraj
skrytka pocztowa 25
07-200 WYSZKÓW
D-97

WOJEWÓDZKIE PRZEDSIĘBIORSTWO HANDLU WEWNĘTRZNEGO ODDZIAŁ W TYCHACH

VIDEOBIT

43-100 Tychy, aleja ZMP 77
tel. 27-69-75

- minikomputery 8-bitowe (Atari, Commodore, Schneider-Amstrad)
- minikomputery 16-bitowe kompatybilne z IBM PC
- drukarki 10" i 15" firm STAR, EPSON, AMSTRAD
- magnetowidy
- kamery video
- anteny satelitarne
- aparaturę badawczo-naukową

Zapewniamy o atrakcyjnych cenach.
G-7

AGENCJA INFORMATYCZNA "Beta 33"

SKRYTKA P-254
41-200 SOSNOWIEC

oferuje, również wysyłkowo-pocztą:
PROGRAMY,
INSTRUKCJE,
OPISY I SCHEMATY TECHNICZNYCH UDOSKONALEN KOMPUTERÓW

ACORN AMSTRAD ATARI
COMMODORE IBM SHARP

KATALOGI INFORMACYJNE-BEZPŁATNE

Telef. 69-03-85

K-106

UDOSKONALENIA PROGRAMOWE I SPRZĘTOWE

DLA WSZYSTKICH MODELI ATARI ORAZ KOMPUTERÓW PRACUJĄCYCH POD SYSTEMEM MS-DOS

WYSYŁKA POCZTĄ

agencja mikro-komputerowa *Amiga*

41-200 Sosnowiec P-157
K-91

Szanowni Czytelnicy!

Ku Waszej wygodzie postanowiliśmy zmienić system przyjmowania ogłoszeń do naszego miesięcznika. Obok biura ogłoszeń MAW przyjmujemy je także bezpośrednio w naszej redakcji. Oferujemy także opracowanie graficzne reklam a także przyjmujemy zlecenia na przeprowadzenie kompleksowych akcji reklamowych prowadzonych nie tylko na naszych łamach.

Ceny naszych ogłoszeń ulegają zmianie i wynoszą:

- 1 słowo — 300 zł + 100% za kolor
- 1 cm² — 550 zł + 100% za kolor
- 1 strona — 368 tys. + 100% za kolor

Pamiętaj: pieniądze przeznaczone na reklamę w naszym piśmie zwracają się zawsze dziesięciokrotnie.



Kalkulatory na dobre zdomowały się już w kieszeniach marynarów i spodni ludzi na całym świecie. Są oczywiście lepsze i gorsze. Takie, na których można wykonać najbardziej nawet skomplikowane obliczenia matematyczne i takie, które nadają się wyłącznie do sprawdzenia wydanej w sklepie reszty.

Grupą ludzi, która używa kalkulatorów na co dzień są bez wątpienia pracownicy i studenci wyższych uczelni. Tam właśnie liczenie odgrywa podstawową rolę w systemie nauczania. Wiele spośród kalkulatorów wychodzi poza ramy podstawowych działań matematycznych. Okazuje się, że na przykład dzięki możliwościom programowania mogą być doskonałymi narzędziami uzupełniającymi dla dużych komputerów personalnych, takich jak Atari ST lub IBM. Niestety, kalkulatory podręczne mają też swoje wady. Zdarza się, że podczas liczenia użytkownik gubi liczby lub myli się przy wykonywaniu działań. Wynika to zarówno z gabarytów sa-

megu urządzenia, jak i z faktu, że kalkulator może wyświetlić tylko jedną liczbę. Druga sprawa to fakt, że często sam posiadacz „liczydła” nie bardzo orientuje się, ile i jakie programy znajdują się w kalkulatorze oraz jak z nich korzystać. Nie zawsze jest też znany obszar pamięci dla zapisu informacji z Sharpa dostępny przy użyciu magnetofono-

wego interfejsu. Zwłaszcza, że sama taśma magnetyczna nie jest najpewniejszym nośnikiem danych z uwagi na jej szybkie zużycie przy częstym odtwarzaniu i brak odporności na pole magnetyczne.

Zapewne z wymienionych pobudek zrodził się pomysł polegający na wykorzystywaniu, do zapisu i redagowania programów dostępnych na kalkulatorze, obszaru pamięci komputerów personalnych. Jedną z takich maszyn jest Atari serii ST. Ma zamkniętą stację dyskietek i ekran monitora, który mieści kilkadziesiąt liczb naraz. Te zalety skłoniły firmę YELLOW COMPUTING do wyprodukowania urządzenia elektronicznego, które za pomocą programu pod nazwą TRANSFILE ST firmy ECPS umożliwia połączenie kalkulatora kieszonkowego Sharp PC 14xx, PC 126x i PC 13xx z komputerem Atari serii ST.

Interfejs ten przenosi programy i dane z Sharpa do pamięci Atari ST. Dzięki temu użytkownik może łatwo opracować niezbędną informację nie męcząc oczu i organizmu, a następnie zapisać ją na dyskietce. Jednocześnie można korzystać z programów pomocniczych dla Atari, które nie są dostępne na zwykłym kalkulatorze. Po zakończeniu „obróbki danych” można wykonać czynność odwrotną i przesłać uzyskane w ten sposób informacje z powrotem do kalkulatora. Przekazywanie danych jest bezproblemowe i odbywa się przy użyciu rozkazów CLOAD, CSAVE lub PRINT przez wejście Centronics, gdzie podłączony jest interfejs. Niezbędne do tej czynności oprogramowanie wykorzystuje GEM i jest bardzo łatwe w obsłudze. Podczas transmisji informacji z kalkulatora do pamięci ST można kontrolować na ekranie monitora prawidłowość wprowadzanych danych. Zalety opisanego urządzenia są oczywiste. Omija się w ten sposób ryzykowną pamięć taśmową i uzyskuje się autentyczny przegląd zawartości wnętrza Sharpa.

Można chyba pokusić się o stwierdzenie, że każdy posiadacz komputera Atari serii ST i kalkulatora Sharp na pewno skorzysta kupując interfejs z programem TRANSFILE ST.

Sergiusz Piotrowski



KALKULATOR SHARP I ATARI ST