

11
Z MIKROKOMPUTEREM NA TY

Bajtek

MIESIĘCZNY DODATEK DO SZTANDARU MŁODYCH

NR 11(35)

LISTOPAD 1988

CENA 150 ZŁ

KONKURS
„O ZŁOTĄ
DYSKIETKĘ
BAJTKA”



„Bo nic, co ludzkie,
nie jest nam obce
BAJTKU dopomóż —
jesteśmy w kropce!”

UWAGA:
Virus!
str. 3-5



śpiewają bohaterowie nowego filmu Krzysztofa Gradowskiego pt. „Pan Kleks w Kosmosie”
To będzie przebiegi rokul (str. 22)

SPIKY HAROLD

TEMAT MIESIĄCA

Tematem miesiąca, a piszę te słowa w październiku, jest wirus komputerowy — niewidoczny zabójca atakujący i niszczący szare komórki mózgu elektronicznych. Poświęcamy temu tematowi główną pozycję w numerze, ale — sądząc po kolejnych doniesieniach o rozprzestrzenianiu się epidemii — na tym temacie z pewnością nie wyčerpiemy. Wszystko wskazuje więc na to, że problematyka wirusów komputerowych zagości u nas na stałe.

Kolejnym komputerowym tematem miesiąca jest fiasko misji „Fobos 1”. Napisałem że jest to temat komputerowy, a nie kosmiczny, gdyż to właśnie nieuwaga programisty zadecydowała o utracie łączności z unikalnym statkiem kosmicznym. O misji „Fobos” pisaliśmy już na łamach „Bajtki” dwukrotnie i to piórem najlepszym z możliwych, bo Jurija Zajcewa z Instytutu Badań Kosmicznych w Moskwie. I dla nas, i zapewne dla Jurija, utrata łączności z „Fobosem 1” jest wielką stratą. Pouczające jest wyjaśnienie, jak do tego doszło.

Otóż, jak wyjaśnił akademik Ralald Sagdiejew, dyrektor Instytutu Badań Kosmicznych w Moskwie, podczas każdego seansu łączności, z ośrodka kierowania lotem wysyłano do „Fobosa 1” zestaw komend, odpowiadający wielu stronom tekstu. Komendy te były przyjmowane przez pokładowy komputer, sterujący wszystkimi urządzeniami statku.

1 września br. „Fobos 1” nie odpowiedział na wezwanie z Ziemi. Nie udało się też uzyskać łączności podczas wielokrotnie ponawianych prób. Przeanalizowano więc dokładnie treść oraz kolejność komend i instrukcji wysyłanych do „Fobosa 1” przed 1 września. Okazało się że w jednej z instrukcji „zgubiono” jedną literę, w rezultacie czego komputer sformułował błędną komendę wyłączenia systemu zapewniającego orientację przestrzenną stacji. O dalszym przebiegu wydarzeń można sądzić na podstawie badań symulacyjnych: „Fobos 1” powoli zmienił swe położenie, aż w pewnym momencie jego baterie słoneczne odwróciły się całkowicie od Słońca i — pozbawiona energii elektrycznej — stacja zamarła...

Przekonujemy stale na łamach „Bajtki” młodych adeptów infor-

matyki o konieczności przestrzegania podstawowych rygorów przy pisaniu programów. Oto macie jeszcze jeden przekonujący dowód, do czego może doprowadzić niechlujstwo i brak uwagi programisty! Całe szczęście, że radzieccy badacze kosmosu mają zwyczaj wysyłać w każdą misję dwa bliźniacze statki. „Fobos 2” czuje się bowiem dobrze, utrzymuje stałą łączność z Ziemią i — jeśli nie zajdą nieprzewidziane okoliczności — w połowie 1989 roku rozpocznie wykonywanie finałowej części swej unikalnej misji!

Natomiast w sierpniu 1990 roku — to już wiadomość z innej beczki — rozpocznie się w Londynie pierwsza Olimpiada Gier Komputerowych. Olimpiada rozgrywana będzie w 20 konkurencjach, przy czym główną z nich będą szachy. Nie rozstrzygnięto jeszcze na czyich szczytach zawisną złote medale zwycięzców, gdyż w olimpijskie szranki staną minikomputery. Ludzie będą tylko przesuwac pionki — ściśle według wskazań komputerów — na planszach dla publiczności. Najlepsi specje już przystąpili do pisania olimpijskich programów... Czyj software okaże się najlepszy? Kto wygra? To dopiero będą emocje!

Emocje towarzyszyć będą również uczestnikom pierwszej Ogólnopolskiej Olimpiady Komputerowej, którą organizuje „Teleekspress” oraz redakcja „Bajtki”. Mogą mieć swoją olimpiadę poloniści, chemicy czy matematycy — to dla czego mieliby jej nie mieć komputerowcy? Ponieważ wydawanie „Bajtki” trwa stosunkowo długo, więc bliższych informacji o naszej olimpiadzie szukajcie w „Sztandarze Młodych”. Włączajcie też telewizor o 17.15... I szukajcie swoje szare komórki!

A co mają robić Ci, którzy są jeszcze za słabi do olimpijskich pojedynków? Proszę bardzo, dla nich też mamy propozycję nie do odrzucenia: już wkrótce ukaże się w kioskach „Ruchu” specjalne wydanie „Bajtki” przeznaczone tylko dla początkujących. Zachęcamy do lektury nawet tych, którzy jeszcze o komputerach i informatyce nie mają zielonego pojęcia. Każdy musiał przecież kiedyś zacząć!

Waldemar Siwiński

BAJTEK

„BAJTEK” — MIESIĘCZNY DODATEK DO „SZTANDARU MŁODYCH”

ADRES: 00-687 Warszawa, ul. Wspólna 6f. Tel. 21-12-05 Przewodniczący Rady Redakcyjnej: Jerzy Domański — redaktor naczelny „Sztandaru Młodych”.

ZESPÓŁ REDAKCYJNY: Waldemar Siwiński (z-ca redaktora naczelnego „SM” — kierownik zespołu „Bajtki”), Grzegorz Onichimowski (sekretarz redakcji „Bajtki”), Roman Poznański (kierownik działu klanów), Krzysztof Czernek, Sławomir Gajda (red. techniczny), Andrzej Pilaszek, Sławomir Polak, Wanda Roszkowska (opr. graficzne), Kazimierz Treger, Marcin Waliński, Roman Wojciechowski. Zdjęcia w numerze: Leopold Dzikowski

Klany redagują:
Commodore — Klaudiusz Dybowski, Dominik Falkowski
Amstrad-Schneider — Jonasz Mayer
Spectrum — Marcin Przasnyski
Atari — Wojciech Zientara, Sergiusz Piotrowski

Fotokład — Tadeusz Olczak,
Montaż offsetowy — Grażyna Ostaszewska,
Korekta — Maria Krajewska, Zofia Wóltańska

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

Cena 150 zł.
Skład technika CRT-200, przygotowalnia offsetowa i druk: PRASOWE ZAKŁADY GRAFICZNE RSW „PRASA-KSIAZKA-RUCH” w Ciechanowie, ul. Sienkiewicza 51.
Nr zleczenia 055528 n. 150.000 egz. U-113

ZA MIESIĄC:

W grudniowym „Bajtku” znajdziecie:

- **Wielki Świąteczny „Konkurs z Ciuchcią” — do wygrania atrakcyjne nagrody**
- **„Bity z Lodówki” czyli o szansach jakie stwarza nadprzewodnictwo dla informatyki**

● **Przekonamy Was, że kalkulator to też komputer**

● **Zdradzimy ostatecznie tajemnice pakietu graficznego na Amstrada**

● **Przedstawimy listę rozkazów mikroprocesora Z80**

● **Zapraszamy do gry „HACKER”!**



UWAGA!

WIRUS!

Jeszcze do niedawna pojęcie „wirus” kojarzyło się nam głównie z epidemią grypy, która prawie każdej zimy daje się we znaki setkom tysięcy ludzi. Niestety, słowo to weszło również do języka komputerowego. Określa się nim groźną plagę, która — szybko się rozprzestrzenia i bardzo dokucza użytkownikom komputerów.

Czym jest wirus komputerowy? Nie jest to nowy typ mikroba, który z niewiadomych powodów zasmakowałby nagle w tworzywach sztucznych. Tak daleko bioinżynieria się jeszcze nie posunęła... Póki co, możemy więc spać spokojnie, nie obawiając się, że w tym czasie jakieś tajemnicze żyjątko pożera klawiaturę naszego komputera! Czyha za to na nas, a właściwie na nasze dyskietki, niebezpieczeństwo o wiele poważniejsze. Nie wierzycie? No to posłuchajcie co przytrafiło się w pewnej redakcji.

Forma Joselow, dziennikarz z „Journal-Bulletin”, zamierzał właśnie wydrukować kolejny artykuł. Z nieukrywanym zadowoleniem sięgnął po dyskietkę zawierającą gromadzone od sześciu miesięcy notatki i wywiady.

Machinalnie wsunął ją do stacji dysków. Jakże wielkie było jednak jego zdziwienie, gdy usłyszał przenikliwy, ostrzegawczy ton, a na ekranie zamiast tekstu pojawił się komunikat błędu. Po kilku daremnych próbach odczytania zapisków zrozpaczony autor udał się do specjalisty z redakcyjnego ośrodka komputerowego. Tam wzięto pod lupę dziwnie zachowującą się dyskietkę i posługując się specjalnym programem przeanalizowano bajt po bajcie całą jej zawartość. Ciąg ułożonych bezsensownie znaków w niczym nie przypominał zdań napisanych przez reportera. **Dopiero w pobliżu sektora „0” udało się wykryć trop prowadzący do sprawców tego niespotykanego dotychczas aktu sabotażu.** Był to numer telefonu

oraz, komunikat zaczynający się od słów WELCOME TO THE DUNGEON.

ZEMSTA BRACI ALVI

W jaki więc sposób dyskietka zawierająca tak groźnego, niewidocznego intruza dotarła do redakcji Journal-Bulletin? Ślad prowadził do historycznego, pakistańskiego miasta Lahore. Pod podanym numerem telefonu odnaleziono niewielki wytapetowany dywanami sklep komputerowy. Dźwięczna nazwa „Brain Computer Services” i wyjątkowo niskie ceny przyciągały tłumy klientów. Już za 1,50 \$ można było tam nabyć programy kosztujące wówczas w USA setki dolarów. Nic dziwnego, że od 1986 do 1987 roku tysiące młodych Amerykanów zaopatrzyło się tu w sławny pakiet „Lotus 1-2-3” czy „Word-Star”. Oprócz kupionego niezwykle tanio oprogramowania wywozili oni do ojczystego kraju ukryty na prawie każdej dyskietce dodatkowy blok kodu maszynowego, którego dostarczania nie przewidywał żaden z producentów. Był to jeden z najbardziej wyrafinowanych wirusów komputerowych. Niczego nie spodziewający się użyt-

kownicy, którzy chętnie pożyczali sobie nawzajem programy, przyczynili się do rozprzestrzenienia epidemii na ponad 100.000 dyskietek*).

Twórcami wirusa okazali się dwaj bracia: 26 letni Amjad Farooq Alvi i 19 letni Basit Farooq Alvi. Właścicielem sklepu był starszy z nich. On to właśnie po ukończeniu fizyki na uniwersytecie w Pendżabie zainteresował się programowaniem i naprawami komputerów. Przez kilka lat zarabiał na życie składając komputery osobiste. W 1985 r. przerzucił się wyłącznie na software i wydawał programy, które ku wielkiemu zaskoczeniu zostały natychmiast bez zezwolenia piracko skopiowane i rozprowadzone w okolicach Lahore. Rozgoryczony Amjad wpadł na pomysł ukarania nieuczciwych użytkowników i napisania programu, który rozprzestrzeniałby się podczas sporządzania kopii, a następnie zaktówał pracę komputera.

Wkrótce bracia zaczęli sami powielać amerykańskie programy i sprzedawać je po wyjątkowo niskiej cenie. Niektóre kopie zarażali wirusem. Zainfekowane dyskietki przeznaczone były tylko dla cudzoziemców, których obaj uznali za głównych winowajców powstania zjawiska kradzieży oprogramowania. W 1987 roku bracia Alvi doszli do wniosku, że dali już wszystkim piratom wystarczającą nauczkę i zaprzestali działalności.





WIRUS POD LUPĄ

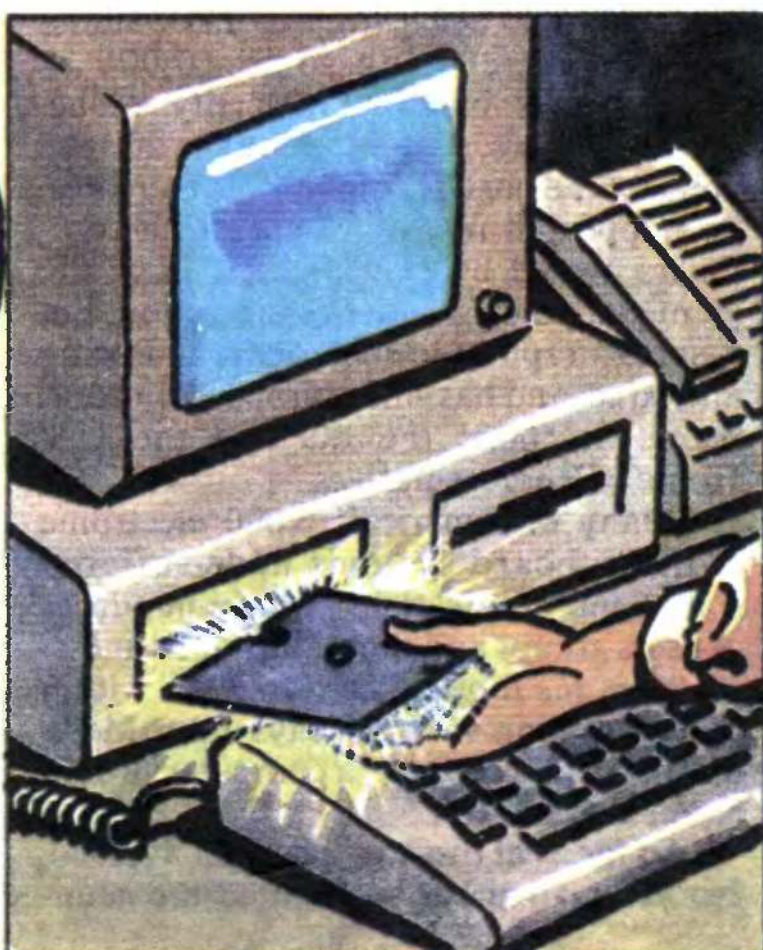
Przyjrzyjmy się bliżej naturze tego unikalnego wirusa. **Atakuje on przede wszystkim komputery osobiste klasy IBM PC/XT, Atari ST, Apple Macintosh.** Okazuje się, że przyjęto w nich takie rozwiązania, systemów operacyjnych, które nie zapewniają wystarczającej ochrony pamięci i zbiorów.

Wirus może być przenoszony wszędzie tam, gdzie następuje przepływ informacji. Znalazienie nowej ofiary nie jest dla niego trudne. Szczególnie narażone na zarażenie są dyskietki nie zabezpieczone przed zapisem. Są to na ogół te, na których umieszczono programy wymagające stałej aktualizacji znajdujących się na nich danych. Przykładem są wspomniane wcześniej „Lotus 1-2-3” lub „Wordstar”, bazy danych, programy kalkulacyjne i graficzne. Dopisanie czegokolwiek na takiej dyskietce może przez długi czas pozostać nie-

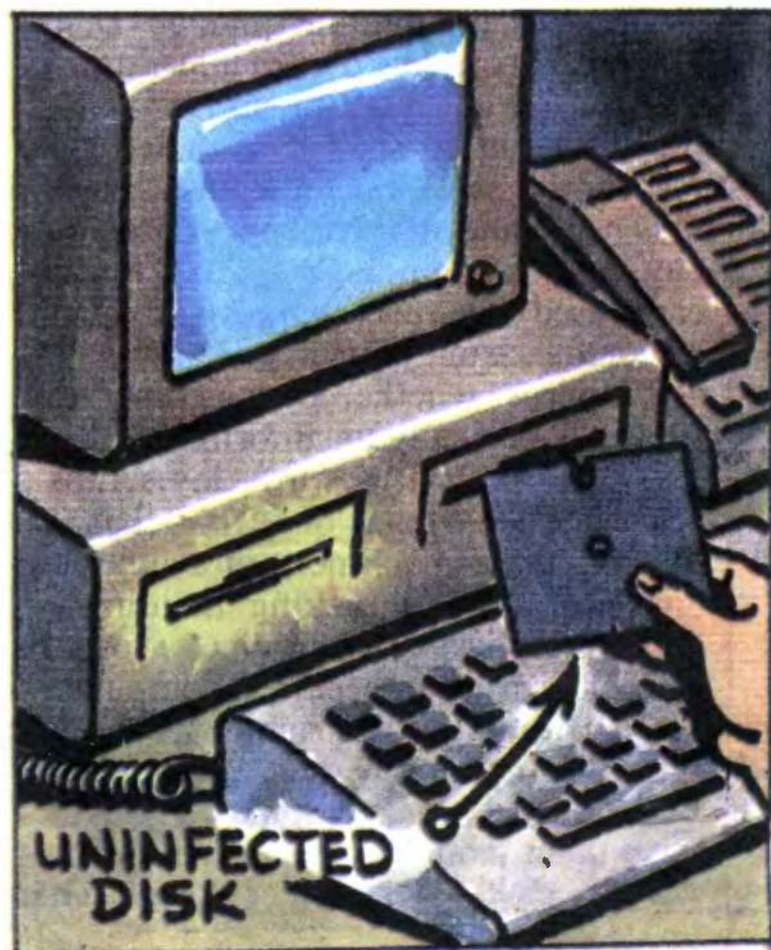
zauważone. Czas ten wirus wykorzystuje po to by się dobrze zadomowić, dokonać spustoszeń i wypatrzeć kolejne ofiary.

Nosicielem choroby może okazać się nawet niewinnie wyglądająca gra, którą skopiowaliśmy od kolegi. Intruz może wtargnąć już przy pierwszym uruchomieniu zarażonych programów, a nawet jeszcze wcześniej. Włożenie dyskietki do stacji dysków jest momentem najbardziej nęralgicznym, gdyż wtedy następuje tak zwany „booting”. Wyjaśnimy ten termin na przykładzie komputera Atari 520 ST. Otóż każda dyskietka ma określoną strukturę, która umożliwia bezbłędne odnalezienie zapisywanych na niej informacji. Podzielona jest ona na koncentryczne ścieżki, a te z kolei na sektory. 3,5-calowa dyskietka do Atari 520ST mieści z jednej strony aż 80 ścieżek zawierających po dziewięć 512-bajtowych sektorów.

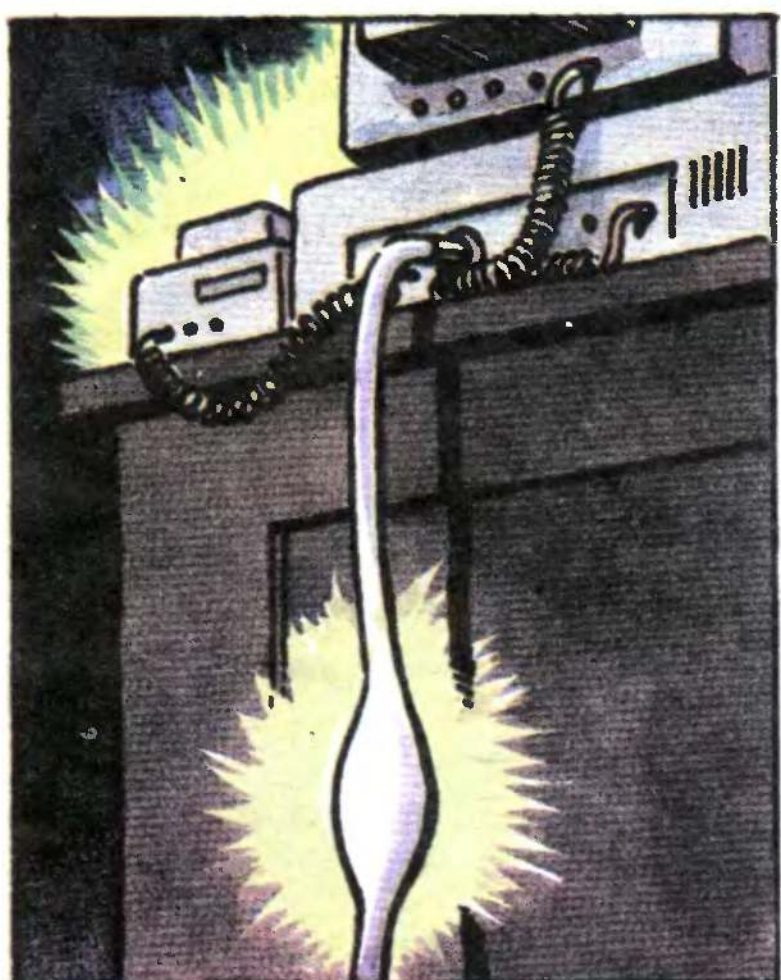
Odszukanie czegokolwiek wśród tak wielu bajtów i sektorów byłoby niemożliwe, gdyby nie wydzielono miejsca na pewne dodatkowe informacje



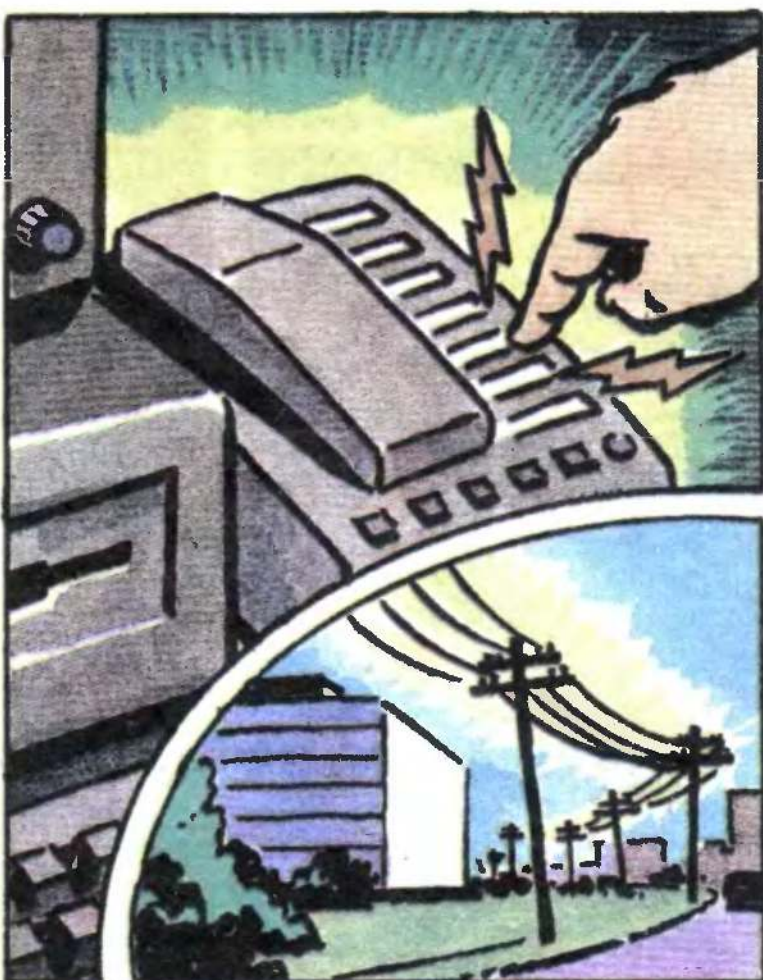
WYGLĄDA JAK ZWYKŁY PROGRAM, DLE JEST ZAINFEKOWANY!



KAŻDA INNA DYSKIETKA, WŁOŻONA DO STACJI DYSKÓW, ZARAZI SIĘ!



... PODOBNIIE PAMIĘĆ KOMPUTERA. NASTĘPNIIE WIRUS MOŻE PRZEDOSTAĆ SIĘ...



... PRZEZ DOWOLNE POŁĄCZENIE TELEFONICZNE DO ŚCIECI KOMPUTEROWEJ.



WIRUS JEST ZWYKLE SPRZĘŻONY Z WEWNĘTRZNYM ZEGAREM KOMPUTERA I PRZYCZAJONY, CZEKA NA WŁAŚCIWY MOMENT, BY UDERZYĆ

pozwalające zaadresować każdy zapisywany zbiór. Zaformatowana, czyli przygotowana do zapisu dyskietka, ma zawsze zarezerwowane na pierwszej ścieżce 7 sektorów na katalog (directory) nazw zbiorów. Rozmieszczenie poszczególnych sektorów przypisanych danemu zbiorowi jest zakodowane w tablicy alokacji czyli książce adresowej zbiorów, która znajduje się zwykle pomiędzy 2 a 6 sektorem ścieżki 0. Najważniejsze dane o dyskietce, można odczytać z pierwszego sektora ścieżki „0” na stronie „0”. Nadano mu nazwę „boot sector” — sektor automatycznego startu. Zapisane są w nim parametry systemu oraz pewien króciutki program. Zawartość tego sektora odczytywana jest zaraz po włożeniu dyskietki do stacji dysków (o czym nietrudno się przekonać słysząc ciche buczenie napędu). Proces ten to właśnie „booting”. Po załadowaniu „boot sektora” może zostać wykonywany znajdujący się w nim program. To właśnie stwarza niepowtarzalną możliwość wczytania i doklejenia do innego zbioru kilkudziesięciobajtowej procedury powodującej zakłócenia albo nawet zniszczenie zawartości całej dyskietki.

WODA W TWARDYM DYSKU?

Po raz pierwszy informacja o wirusach porażających Atari ST opublikowana została przez czasopismo o nazwie „CT”. W jednym z jego numerów wydrukowano listing napisanego w assemblerze prostego wirusa na Atari ST^{**}. Wirus ten zagnieździł się właśnie w „boot sektorze” podczas wymiany dyskietek nie chronionych przed zapisem. Zakłócał on naturalny podział pamięci w dyskietce, co powodowało nieoczekiwane efekty w niektórych programach. W wyniku wymiany programów pomiędzy użytkownikami wirus wkrótce znacznie się rozprzestrzenił — znalazł się nawet na dyskietkach sprzedawanych przez oficjalne wydawnictwa.

Z biegiem czasu programiści pracujący w assemblerze modyfikowali możliwości zakłócania pracy komputera przez ten wirus. Gwarancje uchronienia całego zbioru programów przed

zarażeniem dawało tylko wczesne rozpoznanie zainfekowanej dyskietki i odizolowanie jej od innych.

Sytuacja pogorszyła się znacznie, gdy na targach CeBIT w Hanowerze w 1986 roku jedna z firm zaprezentowała program „Virus Construction Set” dla Atari ST. Kilka tysięcy kopii dostarczono wcześniej dziennikarzom i redakcjom czasopism zaznaczając przy tym stanowczo, że nie należy ich dalej rozpowszechniać. Pojawienie się tego programu wywołało sensację, a równocześnie gwałtowne oburzenie użytkowników i programistów. Firmę już wkrótce zmuszono do zaprzestania prezentacji wzbudzającego kontrowersję wyrobu i zdjęcia go ze stoiska. Nie udało się jednakże zapobiec przedostaniu się kilkunastu pirackich kopii do grona niezbyt uczciwych użytkowników. Już wkrótce VCS wykorzystano do wygenerowania nowych, groźnych szczepów wirusa.

„Virus Construction Set” zniósł naturalną barierę, która uprzednio skutecznie odstraszała większe grono hackerów od robienia niemiłych psikusów. Była nią konieczność doskonałego poznania dyskowego systemu operacyjnego i nabycia umiejętności posługiwania się assemblerem. VCS jest narzędziem pozwalającym konstruować nowe wirusy w podobny sposób jak pracuje się z edytorem tekstów czy bazą danych. Wystarczy wybierać tylko odpowiednie opcje menu oznaczające przeróżne efekty.

Wśród „niespodzianek”, które można w ten sposób spreparować znajduje się zaformatowanie twardego dysku, skasowanie kilkunastu sektorów dyskietki czy nieoczekiwane zapytanie użytkownika o hasło (w tej sytuacji nie pozostaje nic innego, jak wyłączyć zasilanie komputera). W pewnych przypadkach wirus powoduje niekontrolowane rozrastanie się niektórych typów zbiorów i szybkie zaśmieszenie całego dysku. Jeśli zakres proponowanych przez VCS „dowcipów” wydaje się komuś niewystarczający, to może on dołączyć własną procedurę w języku maszynowym. Oczywiście, nie wszyscy mają tak wiesielcze poczucie humoru, by sięgać wyłączenie po formatowanie dyskietek. Pewien żartowniś wymyślił taki oto program na

IBM: na ekranie pojawia się komunikat „W napędzie C wykryto wodę”, po czym przy dźwiękach naśladowujących bulgotanie wody następuje „odwirowanie” dysku i sytuacja powraca do normy.

Wirusy zwykle dokleją się do innego programu i mogą atakować natychmiast lub z opóźnieniem. Taka właśnie „bomba zegarowa” miała zniszczyć informacje zapisane na wielu dyskietkach w Izraelu. Wirus sprzężony z zegarem czasu rzeczywistego czekał na drugi piątek w maju 1988 roku. Gdyby nie szczęśliwy zbieg okoliczności, to obchody 40 rocznicy Izraela przypadającej na 13.05.1988 (bo taką datę wybrał sobie dowcipniś) zakończyłyby się dramatycznie dla wielu posiadaczy elektronicznych mózgow. Na szczęście wskutek znacznego spowolnienia pracy komputera udało się wcześniej zauważyć wirusa i unieszkodliwić go.

SZANSA DLA TERRORYSTY

Specjaliści obawiają się poważnych następstw jakie mogą wywołać wirusy zaszczerpane komputerom w kluczowych instytucjach kierujących życiem państwa. Nie trzeba wielkiej wyobraźni, by uzmysłowić sobie konsekwencje ich wtargnięcia do ośrodków obliczeniowych towarzystw ubezpieczeniowych, banków, systemów militarnych lub ośrodków kontroli ruchu lotniczego. Nie wykryto jeszcze takich przypadków, ale groźba jest całkiem realna.

Na razie wirusy nękają głównie użytkowników komputerów osobistych. Rozprzestrzenianiu się wirusa sprzyja upowszechnienie sieci. Tysiące użytkowników korzystają z szybkich połączeń telefonicznych i banków informacji. Daje to poważne korzyści, ale oznacza również, że wirus wprowadzony do jednego komputera może się błyskawicznie przenieść nawet w bardzo odległe miejsce.

W grudniu 1987 roku zaczęły, na przykład, dziać się dziwne rzeczy z terminalami dołączonymi do jednej z sieci amerykańskich. Na ekranach monitorów pojawiały się życzenia

święteczne. Użytkownicy, którzy postępując zgodnie z pojawiającymi się instrukcjami napisali słowo Christmas, uwalniali mechanizm powodujący przesłanie kopii programu do wszystkich odbiorców figurujących w elektronicznej książce adresowej.

ZŁOTY INTERES

Docierające zewsząd sygnały o istniejącym realnie niebezpieczeństwie są skrętnie rejestrowane przez ludzi, którzy zwiertzyli znakomity interes w podsycaniu psychozy wirusa. Należą do nich przede wszystkim specjaliści i konsultanci zajmujący się ochroną użytkowników przed katastrofalnymi skutkami infekcji. Nieźle zarobiły również firmy sotwarowe, które zdążyły już wydać liczne programy — mniej lub bardziej skutecznie zabezpieczające przed wtargnięciem wirusa: Flu Shot+, Vaccinate, Data Phisician, Disk Defender, Antidote, Vires RX, Vi-Alarm.

Walka z wirusami, którym już udało się zapanować nad naszymi programami jest niezwykle trudna. Wymaga ona żmudnego zbadania każdej dyskietki. W szczególnie ciężkich przypadkach nie możemy właściwie nic zrobić. Należy wtedy skasować zainfektowane dyskietki i ponownie odtworzyć zbiory postępując się tylko oryginalnymi programami z gwarancją producenta. A najprostszym sposobem ochrony przed wirusem jest po prostu nie używanie dyskietek pochodzących z niepewnych źródeł!

Janusz Jarmoch

* TIME 26.09.88

** ATARI MAGAZIN 6/88



CENTRONICS DLA

Jak wiadomo posiadacze komputerów Atari XL/XE są skazani na fabryczną drukarkę 1029.

Powodem jest niestandardowe wyjście szerego-
we. Będące w sprzedaży interfejsy, umożliwiające
współpracę tego komputera z innymi drukarkami
(Star, Citizen, Epson itp.) są dość drogie (w przeli-
czeniu na złotówki).

Większości użytkowników Atari nie stać po pro-
stu na zakup drukarki. Gdy nadarza się okazja sko-
rzystania z dobrej drukarki, na przeszkodzie stoi
owo fatalne wyjście szerego-
we. Wszystkie drukarki
przystosowane do współpracy z komputerami klasy
PC są wyposażone w wejście równoległe standar-
du Centronics.

Przedstawiony obok prosty układ wraz z progra-
mikiem zmienia wejścia joysticków w wyjście zgod-
ne ze standardem Centronics na poziomie podsta-
wowej komunikacji komputer-drukarka.

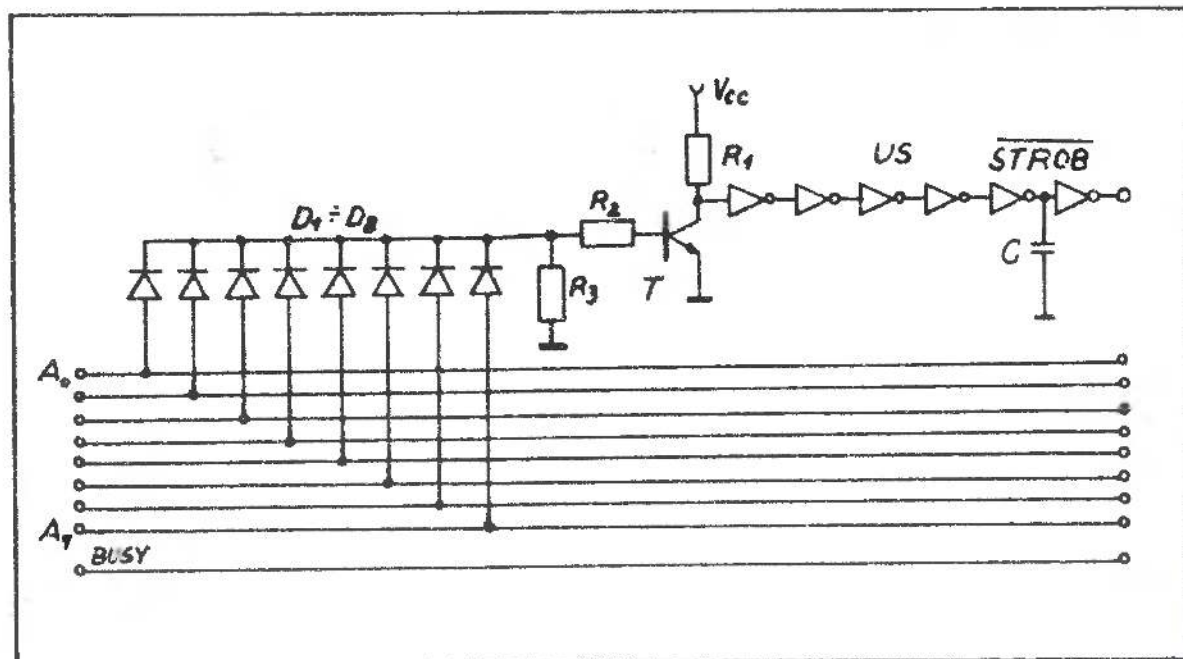
Ze względu na dostępność wtyczek najłatwiej
wykonać interfejs do drukarki wyposażonej w kabel
zakończony 25-kołkową wtyczką (np. współpracu-
jącej z IBM PC). Do płytki obwodu drukowanego
należy więc przylutować 25-stykowe gniazdo pro-
dukowane przez zakłady ELTRA oraz dwa 9-styko-
we do joysticków.

ATARI

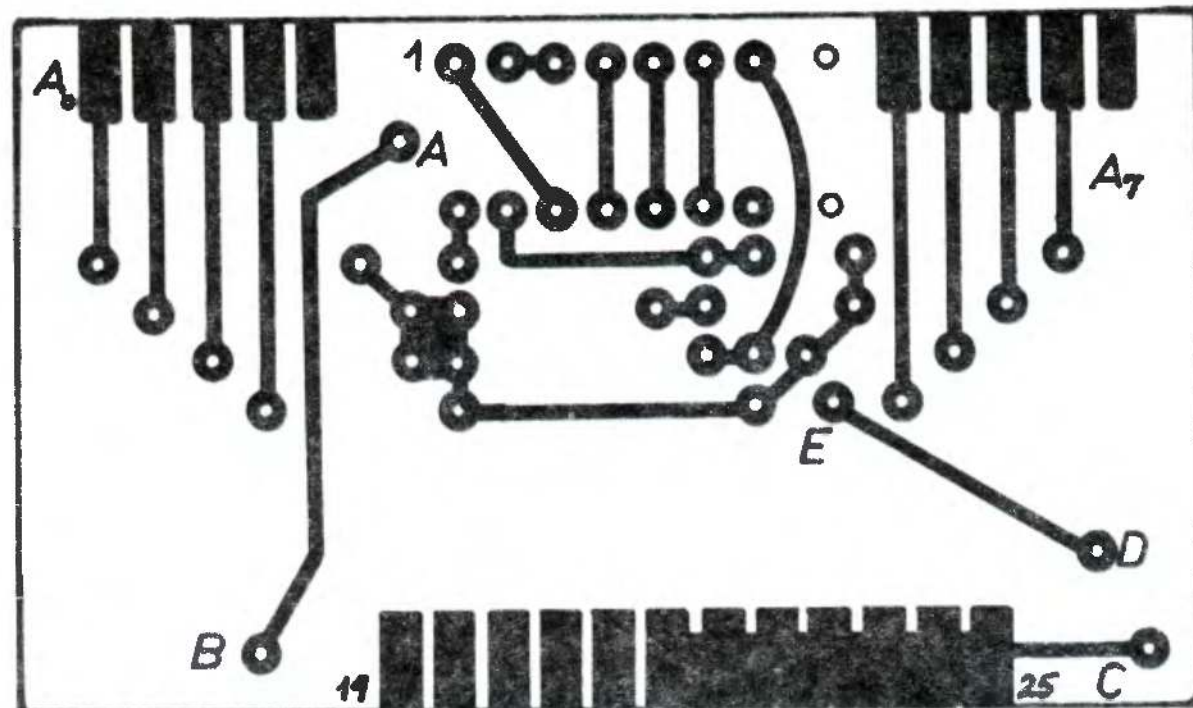
LISTING 1

```

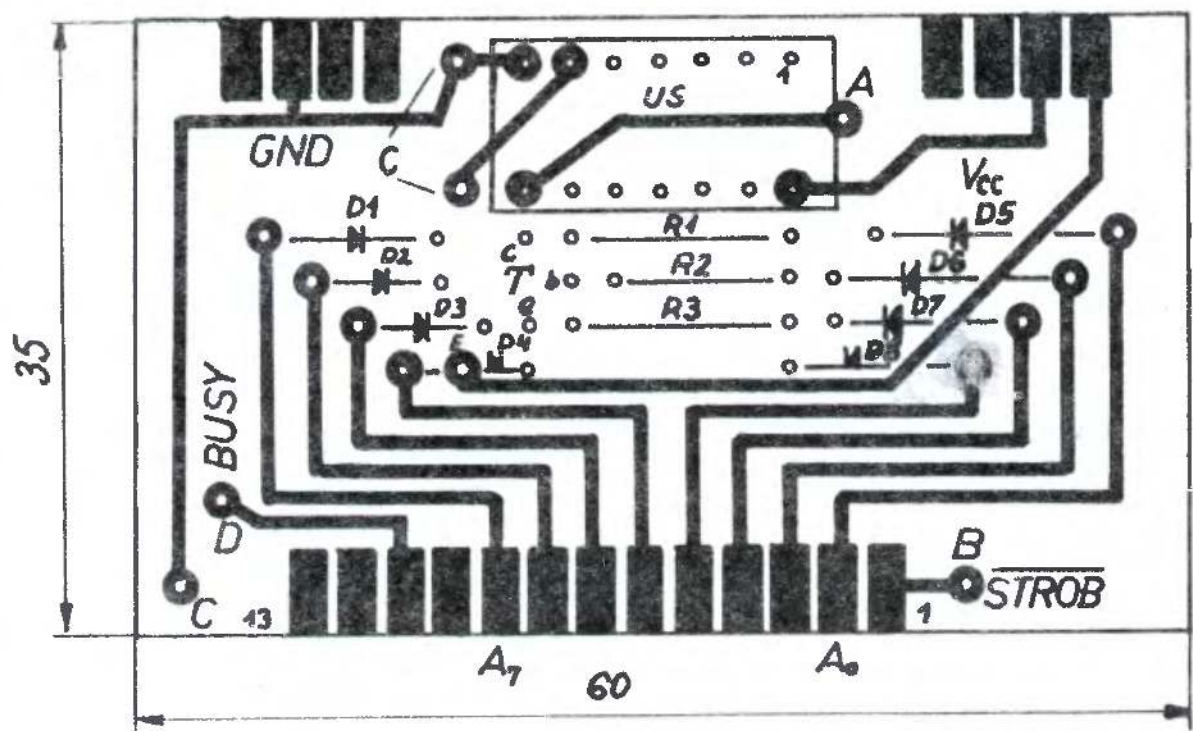
HY 10 REM *****
GJ 20 REM * OBSLUGA WYJSCIA CENTRONICS *
IA 30 REM *****
XL 40 CLR: DIM A$(6): RESTORE 160
LE 50 FOR I=5000 TO 5112
AM 60 READ K
AR 70 POKE I,K
XL 80 NEXT I: READ K
TK 90 IF K<>-10 THEN ? "BLAD W DATA": END
RR 100 A$="hxxLVx": A$(2,2)=CHR$(162)
HF 110 A$(3,3)=CHR$(16):A$(6,6)=CHR$(228)
JI 120 OPEN#1,8,128,"C:": POKE 850,11
KB 130 POKE 856,113: POKE 857,0
MN 140 POKE 853,19: POKE 852,136
QY 150 I=USR(ADR(A$)): CLOSE#1
DQ 160 DATA 0,1,250,6,6,7,169,60,141,2
UU 170 DATA 211,96,169,0,133,14,141,231
PF 180 DATA 2,169,8,133,15,141,232,2,173
PH 190 DATA 105,7,141,27,3,173,106,7,141
UD 200 DATA 28,3,96,160,255,208,2,160,0
VL 210 DATA 173,2,211,170,41,251,141,2
CL 220 DATA 211,140,0,211,142,2,211,160
GJ 230 DATA 1,96,160,131,96,201,155,208
FU 240 DATA 2,169,10,166,20,202,172,16
PH 250 DATA 208,208,16,160,0,140,0,211
EL 260 DATA 162,10,202,16,253,141,0,211
YH 270 DATA 160,1,96,228,20,208,231,160
HK 280 DATA 138,96,32,7,36,7,56,7,59,7
QT 290 DATA 97,7,-10
EV 300 FOR I=5000 TO 5112: POKE I,0
RS 310 NEXT I: NEW
    
```



Rys. 1. Schemat ideowy



Rys. 2. Widok płytki od strony druku



Rys. 3. Widok płytki od strony elementów

Przed wlutowaniem elementów na płytkę trzeba
w punktach A, B, C, D i E obustronnie wlutować ka-
wałki drutu miedzianego lub np. obcięte wyprowa-
dzenia rezystorów, aby uzyskać połączenia ścieżek
po przeciwnych stronach płytki. W następnej kolej-
ności należy wlutować układ scalony, rezystory,
diody, tranzystor i na końcu wtyczki.

Układ nie wymaga żadnych regulacji ani selekcji
elementów. Ze względu na wymiary płytki rezysto-
ry powinny mieć obciążalność 0,25 lub 0,125 W.

Tutaj drobna uwaga: od strony elektrycznej i
montażowej układ jest bardzo prosty i zaawanso-
wanemu hobbyście nie powinien sprawić żadnych
kłopotów. Pamiętaj jednak, że każdy błąd może Cię
drogo kosztować — ingerujesz przecież we wnę-
trze komputera. Również bardzo zgubne są wszęd-
obylskie ładunki elektrostatyczne.

Zamieszczony program należy przepisać i uru-
chomić (RUN). Podwójny sygnał powiadamia o ko-
ńieczności nagrania na taśmę wersji boot progra-
mu. Odtąd program należy wczytywać jak inne w
kodzie maszynowym: wciśnięty klawisz START
przy włączaniu komputera.

Aby pozostawić wolną, często wykorzystywaną
do własnych celów, szóstą stronę pamięci, pro-
gram umieszczony jest od adresu 1792 do 2000.
Zamieszczona obok wersja źródłowa ułatwi zmiany
i pozwoli na wykonanie własnej wersji użytkowni-
kom stacji dysków — wersja kasetowa zajmuje bo-
wiem miejsce przeznaczone dla DOS-u.

Transmisja danych przez opisane wyjście równo-
ległe przebiega z dużą prędkością: np. przestanie
bloku 8.5 KB trwa zaledwie 2.5 sekundy. Interfejs
można wykorzystywać praktycznie tylko z poziomu
Basica dlatego, że uruchamiający się program w
kodzie maszynowym ustawia swoje wskaźniki, czyli
skasuje nasz. Można też obsługę złącza dopisać
do interesującego nas programu. Po takim zabiegu
edytor tekstu SpeedScript 3.0 umożliwia np. uzys-

kanie polskich liter na drukarce Centronics GLP II.
UWAGA!!! WYKONYWANIE KOPII PROGRAMU I
OPISANEGO UKŁADU W CELACH HANDLO-
WYCH JEST ZABRONIONE!!!

Marek Renner
40-532 Katowice
ul. Dzieciołów 26


```

1000      *= $06FA
1010  ET   .BYTE $00,$01
1020      .WORD ET
1030      .WORD ET1
1040      LDA  #$3C
1050      STA  $D302
1060      RTS
1070  ET1  LDA  #$00
1080      STA  $000E
1090      STA  $02E7
1100      LDA  #$0B
1110      STA  $0F
1120      STA  $02EB
1130      LDA  ET12
1140      STA  $031B
1150      LDA  ET12+1
1160      STA  $031C
1170      RTS
1180  ET2  LDY  #$FF
1190      BNE  ET4
1200  ET3  LDY  #$00
1210  ET4  LDA  $D302
1220      TAX
1230      AND  #$FB
1240      STA  $D302
1250      STY  $D300
1260      STX  $D302
1270      LDY  #$01
1280      RTS
1290  ET5  LDY  #$B3
1300      RTS
1310  ET6  CMP  #$9B
1320      BNE  ET7
1330      LDA  #$0A
1340  ET7  LDX  $0014
1350      DEX
1360  ET8  LDY  $D010
1370      BNE  ET10
1380      LDY  #$00
1390      STY  $D300
1400      LDX  #$0A
1410  ET9  DEX
1420      BPL  ET9
1430      STA  $D300
1440      LDY  #$01
1450      RTS
1460  ET10 CPX  $0014
1470      BNE  ET8
1480      LDY  #$BA
1490      RTS
1500  ET11 .WORD  ET2-1,ET3-1,
1510      .WORD  ET5-1,ET6-1
1520  ET12 .WORD  ET11
    
```

STEROWANIE MAGNETOFONEM MK 232 PRZEZ KOMPUTER ATARI

Ze względu na koszty duża część posiadaczy komputerów Atari korzysta z interfejsów do zwykłych magnetofonów zamiast z firmowych pamięci kasetowych.

Jednym z najpopularniejszych magnetofonów używanych do tego celu jest MK 232 i jego pokrewne wersje. Firmowy magnetofon Atari posiada jednak pewną przewagę — jego praca jest sterowana przez komputer. Takiej możliwości nie daje interfejs. Aby ją uzyskać należy wykonać kilka drobnych przeróbek w fabrycznym MK 232, które zdoła z pewnością zrobić każdy, kto umie posługiwać się lutownicą:

1) Wykonać płytkę drukowaną wg rysunku i wmontować w nią elementy wyszczególnione na schemacie.

2) Na płycie czołowej magnetofonu, z lewej strony, w miejscu przeznaczonym na wskaźnik zużycia baterii w MK 235 wykonać odpowiedni otwór i wmontować mały przełącznik stały (mogą być pewne kłopoty ze zdobyciem odpowiednio małego, ja użyłem przełącznika z japońskiego mikrofonu).

3) Zamontować płytkę drukowaną w magnetofonie (najlepiej obok gniazda wejścia — wyjścia).

4) Wlutować przewód w otwór oznaczony literą „H” i połączyć poprzez wyłącznik ze środkowym stykiem znajdującym się tuż przy silniku (są tam 3 styki).

6) Do otworu „L” wlutować przewód i podłączyć go tymczasowo do dodatniego bieguna płaskiej baterii (4.5V), biegun ujemny dołączyć do masy magnetofonu poprzez mikroampero-

mierz. W czasie, gdy włącznik jest rozarty lub gdy jest zwarty i podłączona jest bateria, magnetofon powinien pracować normalnie. Natomiast, gdy włącznik jest zwarty i bateria odłączona, magnetofon nie powinien pracować. Podczas, gdy dołączona jest bateria, należy odczytać wskazanie mikroamperomierza, które nie może przekroczyć 0,005 A.

7) Znaleźć na płytce drukowanej magnetofonu styk połączony z piątym kontaktem w gnieździe wejścia — wyjścia, przylutować do niego przewód połączony uprzednio z dodatnim biegunem baterii i odciąć jego połączenie na płytce z kontaktem trzecim.

8) Do trójżyłowego ekranowanego kabla należy przylutować dwie wtyczki stereofoniczne, przy czym ekran połączyć z drugą igłą, a czwartą igłą pozostawić wolną.

9) Piąty kontakt gniazda stereofonicznego w interfejsie połączyć ze ścieżką na płytce drukowanej interfejsu zasilającego układ napięciem 5V.

I na koniec jeszcze kilka uwag. Jeżeli przerobiony magnetofon będzie współpracował z innymi urządzeniami, to należy go łączyć tylko wtykiem monofonicznym. Po odłączeniu napięcia przez komputer silnika zatrzymuje się po około 2 s. Jest to efekt, którego nie udało mi się wyeliminować. Przedstawione ulepszenie zwiększa poprawność wczytywania programów z długimi przerwami (szczególnie gdy są one dołączane do długich programów). Dzieje się tak, ponieważ komputer zatrzymuje magnetofon pomiędzy rekordami. Inną zaletą jest wyłączanie magnetofonu po zakończeniu wczytywania lub nagrywania programu oraz w przypadku błędu wczytywania.

Dariusz Tabaczyński

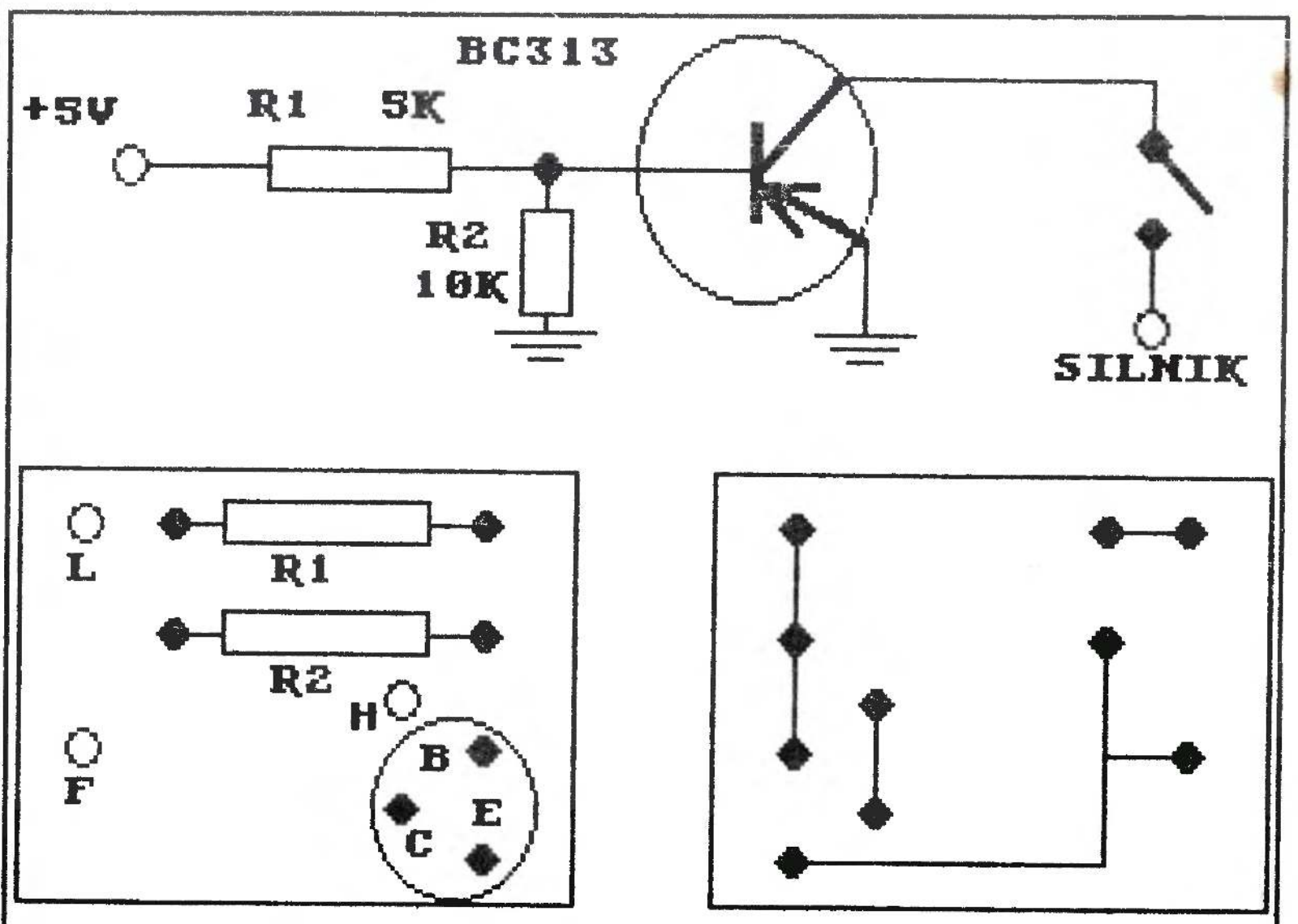
Chętnie pomogę mającym kłopoty z wykonaniem interfejsu. Po podaniu, jaki wtyk (po stronie komputera) ma kabel drukarki, do której masz dostęp wysłę gotową płytkę albo (po dostarczeniu wtyczek) kompletny układ interfejsu.

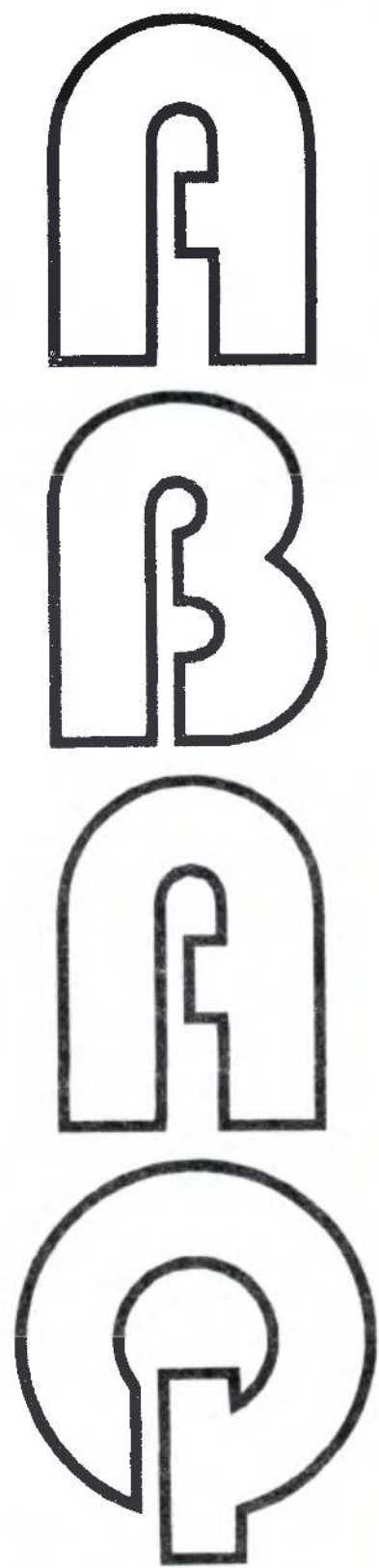
Pomogę również w rozszerzeniu programów o program obsługi drukarki.

Zestawienie części:

- D1-D8 diody BAVP 20
- T tranzystor BC 107
- R1 rezystor 5,1 kΩ
- R2 rezystor 1,0 kΩ
- R3 rezystor 3,0 kΩ
- US układ scalony UCY 7404
- C kondensator 3,3 nF

OD REDAKCJI: Powyższy układ został przetestowany w redakcji. Wykazuje on dwie wady, o których autor nie wspomina w opisie. Interfejs nie przenosi kodu 0 czyli CHR\$ (0). Ponadto program obsługujący zawsze zamienia kod RETURN (155) na kod CR (13). Prowadźcie to oczywiście błędy podczas wydruku grafiki.





**najmłodsze
dziecko
Atari**



Na międzynarodowych targach komputerowych CeBTT'88 w Hanowerze firma Atari Corp. pokazała swój najnowszy produkt zbudowany w oparciu o najnowocześniejszą technologię wykorzystującą układy zwane transputerami.

Urządzenie tego typu, które można traktować jako tzw. stanowisko robocze musi mieć niezwykle możliwości, a mianowicie:

- wykonywać przynajmniej milion operacji w ciągu jednej sekundy
- mieć co najmniej 1 MB pamięci podstawowej
- uzyskiwać grafikę o rozdzielczości powyżej jednego miliona punktów
- posiadać kanał do szybkiej transmisji danych (powyżej 1 MB na sekundę)
- pracować pod wielozadaniowym systemem operacyjnym opartym na technice okien.

Typowa stacja pracy (workstation), na przykład Apollo Domain DN 4000 oparta jest na procesorze 68020 z typowym koprocesorem arytmetycznym (16—25 MHz). Urządzenie pracuje pod systemem operacyjnym UNIX lub podobnym i może przesyłać przez sieć Ethernet do 1,5 MB na sekundę. Cena, produkowanych do tej pory tego typu maszyn, spełniających po-

dane warunki, dochodziła do 40000 DM.

Trafiamy w tym miejscu do sedna sprawy. Oto ponownie okazało się, że slogan reklamowy firmy Atari „Power without the price” znajduje swoje uzasadnienie w praktyce. System transputerowy Atari spełnia wszystkie wymagane warunki stawiane profesjonalnym stanowiskom roboczym, a jego cena waha się w granicach 10000 DM. Dzieje się tak dzięki zastosowaniu w tym rozwiązaniu specjalnego układu scalonego firmy INMOS o nazwie T800-Transputer. Jest to szczególnie szybki, 32-bitowy procesor o niezwykle zaletach.

Transputery są oparte o procesory typu RISC. Skrót ten pochodzi od angielskiej nazwy Reduced Instruction Set Computer, co oznacza komputer o ograniczonej liczbie rozkazów. Stosowanie takiej właśnie techniki zyskało na znaczeniu od momentu, kiedy okazało się, że maszyny z takim kodem buduje się szybciej. Wynika to z faktu, że ograniczoną liczbę rozkazów w procesorach typu RISC można zaimplementować w sprzęcie (hardware). Natomiast rozkazy w konwencjonalnych mikroprocesorach realizowane są przez mikroprogram.

Sumując, często może tak być, że procesor jest szybszy wtedy, kiedy potrzebuje dla wykonania określonego zadania kilka bardzo szybkich rozkazów, niż jeżeli musi wykonać jeden wolny. Drugim zastosowanym wybiegiem jest dołączenie dodatkowej pamięci notatnikowej. Procesor traci bar-

dzo wiele cennego czasu, kiedy musi ściągać informacje i rozkazy z dużej, lecz relatywnie wolnej pamięci podstawowej. Z tego powodu uzupełnia się nowoczesne procesory o tzw. pamięć notatnikową, która jest szybsza od pamięci głównej. W przypadku omawianego systemu transputerowego pojemność takiej stacji pamięci wynosi 4 KB dla każdego układu T800. Dzięki temu rozwiązaniu można realizować duże pętle programu bez ingerencji w pamięć podstawową. Transputer T800 ma także układ spełniający rolę arytmometru oraz jest zintegrowany z koprocesorem numerycznym. Arytmometr T800 jest wyjątkowo szybki. Może sterować i wykonywać 1.5 miliona operacji zmiennoprzecinkowych na sekundę przy 20 MHz. Wersja 30 MHz jest w przygotowaniu. T800 jest szybszy o kombinacji układów 68030/68882.

Szczególnością cechą transputerów są cztery wejścia szeregowo, które mogą wysyłać i przyjmować do 20 megabodów. Porty te umożliwiają łączenie wielu transputerów w sieć, co z kolei zwiększa możliwości procesora właściwego i pozwala na liczenie równoległe. Mimo osiąganego postępu w tej dziedzinie, ludzie zajmujący się tymi sprawami są dopiero na początku drogi. Przyczynkiem do rozwoju stała się w tym wypadku koncepcja połączenia urządzenia z odpowiednim oprogramowaniem. Firma INMOS, która prowadzi prace nad transputerami, rozwinięta także specjalny współbieżny język programowania. Nosi on nazwę

OCCAM, często też mówi się o nim „współbieżne C”. OCCAM pozwala na programowanie procesów szeregowych i równoległych, które mogą krążyć i komunikować się ze sobą poprzez kanały informacji.

ABAQ system transputerowy Atari zbudowany jest w oparciu o cztery procesory typu RISC — T800, a jego płyta główna ma pojemność 4 MB pamięci RAM. System powinien zawierać dysk twardy (przynajmniej 4 MB). Jeden z czterech transputerów połączony jest interfacem z Megą ST, która obsługuje klawiaturę, mysz i komunikację ze stacjami dysków elastycznych. Dzięki Mega ST mogą być dołączone do ABAQ wszystkie urządzenia peryferyjne do ST. Transputer przejmuje przepływ obrazu na płycie głównej. W tym celu ma do dyspozycji 1 MB pamięci obrazu składającego się z tzw. Dual-Port-Ram, co oznacza pamięć, z której korzystają równocześnie dwa urządzenia. Są to procesor obrazu i blitter, który pomaga systemowi wrócić do normalnej dwuwymiarowej grafiki. Blitter przenosi 128 milionów pikseli na sekundę, a w przypadku pisma może przedstawić do 64 milionów pikseli/sek.

System ABAQ ma cztery tryby graficzne. W najwyższym użytkownik dysponuje rozdzielczością 1280 960 punktów z 16 kolorami lub odcieniami jednego koloru. Drugi ma rozdzielczość 1024 768 punktów przy 256 kolorach wyprowadzonych jednocześnie na ekran z palety 16,7 milionów kolorów. W trzecim trybie są dwa obrazy po 640 480 punkty każdy i 256 kolorów z palety 16,7 milionów. Ostatni to mała rozdzielczość grafiki 512 480 punktów, która umożliwia równoczesne przedstawienie całej palety 16,7 milionów kolorów na ekranie monitora.

Na płycie głównej znajdują się trzy wejścia, przez które można dołączyć dodatkowe transputery, karty rozszerzenia lub pamięci. Do jednej karty rozszerzenia można wykorzystać maksymalnie 4 transputery. Aktualna konstrukcja systemu ogranicza ilość użytych transputerów do 13, które osiągają przepustowość 20 megaflopów. Istnieje również możliwość dołączenia dodatkowych układów T800 poprzez sieć zewnętrzną. W tej chwili jednak standard takiej sieci nie jest zdefiniowany.

Sam idea ABAQ (sprzęt i oprogramowanie) została rozwinięta przez firmę PeriHelion, która wypracowała ten model wspólnie z jednym z angielskich uniwersytetów. Marzeniem twórców było stworzenie odpowiedniego oprogramowania zanim produkt trafi na rynek. Doprowadziło to do opracowania systemu operacyjnego Helios. Ma on szansę stać się standardem na transputerowym rynku. Helios jest systemem operacyjnym typu Multitasking kompatybilnym z UNIX-em. Jest to system napisany specjalnie dla transputerów. Obecnie trwa intensywna praca nad możliwością realizacji obliczeń współbieżnych bez używania specjalnego języka typu OCCAM. Zamiast dotychczasowych procedur należy znaleźć takie, które spowodują, że równoległe wykonywane części programu będą od siebie niezależne. Helios jest na dobrej drodze do osiągnięcia sukcesu. Oprogramowniem systemowym mają być najpierw C i Fortran, potem OCCAM, a w końcu Pascal i Modula II.

Jedno jest pewne, że razem z rozwojem systemu transputerowego Atari nadejdą czasy, w których cena wyboru zastosowania będzie zależna w dużym stopniu od użytego sprzętu.

*Na podstawie „ST Computer”
Sergiusz Piotrowski*

ZAMIANA NAPISÓW RAZ JESZCZE

Posiadam Atari 800XL z magnetofonem. Komputera używam głównie pisząc własne programy. Ponieważ lubię jednak czasem pograć, postanowiłem ułatwić sobie życie w grach Drop Zone i Starquake. Korzystając z wiadomości zamieszczonych w „Bajtku” nr 5/88 (art. „Zostań nieśmiertelnym”) użyłem program „Zamiana napisów” opublikowany w „Bajtku” nr 7/87.

Jednakże, tak jak przypuszczałem, zabrakło pamięci, aby wczytać grę. Ponieważ nie dysponowałem innym programem o podobnych możliwościach, postanowiłem napisać swoją wersję. Mimo, iż użyłem do tego celu Basica, program dysponuje przeszło dziesięcioma kilobajtami pamięci więcej od programów „Zmiana napisów” i „Nie tylko dla graczy” („Bajtek” nr 3/88). Uzyskałem to przez przesunięcie programu ANTIC-a i pamięci ekranu w obszar tuż nad głównym programem. Dodatkowo podczas operacji czytania i zapisu odłączany jest ROM Basica, co pozwoliło zaoszczędzić osiem kilobajtów pamięci. Ze względu na zastosowanie tego triku, procedura wyszukująca i wymieniająca dostarczony ciąg bajtów musiała zostać napisana w języku wewnętrznym. Jedną z zalet tego rozwiązania jest niewspółmierne skrócenie czasu przeszukiwania pamięci. Przeszukanie czterdziestu kilobajtów w celu odnalezienia określonego ciągu pięciu bajtów trwa około jednej sekundy. W Basicu trwałoby to przez 7 minut...

Program składa się z dwóch części. Część pierwsza umieszcza potrzebne procedury na szóstej stronie i wczytuje część drugą, która jest głównym programem. Rozwiązałem to w ten sposób, gdyż zależało mi jak na największym obszarze wolnej pamięci RAM. Pierwsza część jest zapisana przez SAVE”C”, część druga bezpośrednio po pierwszej przez CSAVE. Całość należy wczytywać poprzez RUN”C”. Po wczytaniu całości program należy uruchomić instrukcją RUN.

Po ustyszeniu pojedynczego sygnału należy wczytać program, w którym chcemy wprowadzić zmiany. Po wczytaniu wyświetlana jest ilość bajtów zajmowanych przez wczytany blok. Następnie program pyta się czy pokazać wczytany zbiór w kodach ATASCII. Potem program oczekuje wprowadzenia kodów ATASCII, które chcemy zmienić. Naciśnięcie RETURN nagrywa zmieniony program.

Po znalezieniu dostarczonego ciągu znaków program wyświetla adresy, ich zawartość i odpowiadające im kody, a następnie oczekuje wprowadzenia ciągu kodów do zmiany (tej samej długości co odnaleziony ciąg). Zmiana sygnalizowana jest ponownym wyświetleniem adresów.

UWAGA:

Wyświetlanie wczytanego zbioru w kodach ATASCII możemy przerwać klawiszem BREAK. Podczas wyświetlania klawisz BREAK zatrzymuje tylko listowanie zbioru, nie zatrzymuje głównego programu. Ze względu na przełączanie banków pamięci, procedura wyświetlająca — zbiór jest napisana w języku wewnętrznym, wykorzystuje ona procedurę obsługi urządzeń I/O CIOMAIN.

Mirosław Honkowicz

```

XP 0 REM ZAMIANA NAPISOW - cz. I
JD 1 REM Mirosław Honkowicz
XM 2 REM Copyright 1988 Bajtek
XN 5 GRAPHICS 0:POSITION 11,10:? "ZACZEKA
   J CHWILE...":POSITION 7,8:? "NIE WYLAC
   ZAJ MAGNETOFONU."
VC 10 FOR X=1536 TO 1734:READ A:POKE X,A:
   S=S+A:NEXT X
AC 20 IF S<>22509 THEN ? "BLAD"
BA 30 POKE 764,12:CLOAD
MF 100 DATA 104,104,104,170,104,104,157,6
   6,3,104
WL 110 DATA 157,69,3,104,157,68,3,104,157
   ,73
RL 120 DATA 3,104,157,72,3,169,131,141,1,
   211
VF 130 DATA 32,86,228,169,253,141,1,211,9
   6,234
IM 140 DATA 104,104,104,133,22,104,133,25
   ,104,133
TL 150 DATA 24,104,133,27,104,133,26,169,
   131,141
DF 160 DATA 1,211,160,0,165,25,197,27,240
   ,22
CJ 170 DATA 177,24,205,128,5,240,33,165,2
   5,197
KT 180 DATA 27,240,9,230,24,208,233,230,2
   5,76
FP 190 DATA 70,6,165,24,197,26,240,5,230,
   24
MR 200 DATA 76,70,6,169,253,141,1,211,96,
   234
AY 210 DATA 160,0,177,24,217,128,5,208,6,
   200
ZN 220 DATA 196,22,208,244,169,253,141,1,
   211,96
KH 230 DATA 160,0,76,77,6,234,104,169,131
   ,141
TJ 240 DATA 1,211,160,0,185,128,5,145,24,
   200
BN 250 DATA 196,22,208,246,169,253,141,1,
   211,96
PG 260 DATA 234,104,169,131,141,1,211,104
   ,141,69
SX 270 DATA 3,104,141,68,3,104,141,73,3,1
   04
YY 280 DATA 141,72,3,169,11,141,66,3,162,
   0
SU 290 DATA 32,86,228,169,253,141,1,211,9
   6
DV 0 REM ZAMIANA NAPISOW - cz. II
JD 1 REM Mirosław Honkowicz
XM 2 REM Copyright 1988 Bajtek
QR 10 DIM A$(40)
PH 20 RAM=PEEK(14)+256*PEEK(15)+992+864
RS 30 RAM=INT(RAM/256)+1
WU 40 POKE 106,RAM:GRAPHICS 0:RAMP=RAM*25
   6:BYTE=49152-RAMP
AR 50 ? CHR$(125):POKE 710,0:CLOSE #1:OPE
   N #1,4,128,"C":? "CZYTAM C:"
NW 60 I=USR(1536,16,7,RAMP,BYTE):B=PEEK(8
   56)+256*PEEK(857)
PJ 70 ST=PEEK(851):IF ST=1 OR ST=136 THEN
   90
NA 80 ? CHR$(253);"BLAD ";ST:GOSUB 270:RU
   N
EI 90 ? "ZBIOR ZALADOWANY W CALOSCI ";B;"
   BAJTOW."
UZ 100 ? "CHCESZ ZOBACZYC ZBIOR ";:INPUT
   A$:IF A$<>"T" THEN 140
XH 110 POKE 766,255
TS 120 I=USR(1696,RAMP,B)
BC 130 POKE 766,0:POKE 764,255:GOSUB 270
TJ 140 ? CHR$(125):? "PODAJ KODY ATASCII
   ";:INPUT A$:X=LEN(A$):IF A$="" THEN 22
   0
UF 150 I=USR(1576,X,RAMP,RAMP+B)
NZ 160 AD=PEEK(24)+256*PEEK(25):IF AD=RAM
   P+B THEN ? "NIE MA TAKICH KODOW !":GOS
   UB 270:GOTO 140
VD 170 GOSUB 280
KQ 180 ? "PODAJ NOWE KODY TEJ SAMEJ DLUGO
   SCI ";:INPUT A$:IF X<>LEN(A$) THEN 180
AU 190 I=USR(1672)
UQ 200 GOSUB 280
BD 210 GOSUB 270:GOTO 100
LI 220 CLOSE #1:OPEN #1,8,128,"C:"
DU 230 ? "ZAPISUJE C:"
QS 240 U=USR(1536,16,11,RAMP,B)
ZO 250 IF PEEK(851)<>1 THEN 80
OD 260 END
OW 270 FOR OP=1 TO 500:NEXT OP:RETURN
DY 280 ? CHR$(125):POKE 766,2:FOR L=0 TO
   X-1:? AD+L,PEEK(AD+L),CHR$(PEEK(AD+L))
   :NEXT L:POKE 766,0:RETURN
    
```



Stoleczny Ośrodek
Elektronicznej
Techniki Obliczeniowej

INFORMATYKA mikrokomputerowa

STEFAN NAWROCKI



PROGRAMOWANIE ST. 11
W. M. 1988

ATARI

Wydanie I, 1988

Atari 520ST od dawna nie jest już najnowszym szlagierem techniki mikrokomputerowej. Na rynkach krajów Europy Zachodniej ma się wkrótce pojawić komputer Atari nowej generacji, który na razie nazwano roboczo „Abaq”. Dla nas jednak zwykła 520 ciągle jest nowością, o której tak naprawdę niewiele można się dowiedzieć.

Nielicznych jeszcze użytkowników tego komputera bardzo ucieszy fakt wydania książki, która być może okaże się pierwszą jaskółką zapowiadającą ukazanie się całej serii poświęconej ST. Książka Stolecznego Ośrodka Techniki Obliczeniowej pod tytułem „GFA BASIC” jest pewnego rodzaju vademecum programowania w BASIC-u na Atari 520ST. Zawiera ona związany opis funkcji i komend GFA BASIC-u, który jest najpopularniejszą wersją tego języka dla ST. Rozkazy omówione są według z góry określonego schematu: opis instrukcji, składnia, opis parametrów z krótkim objaśnieniem i przykład. Wszystkie hasła umieszczone są w porządku alfabetycznym, co ma bardzo duże znaczenie przy systematycznym korzystaniu z książki podczas samodzielnego programowania. Dodatkowym udogodnieniem jest wykaz błędów występujących na różnych etapach pracy nad programem (podczas edycji, wykonania, kompilacji oraz błędy sygnalizowane przez interpreter. Skróto-wo opisano także kompilator GFA BASIC-a oraz najważniejsze terminy obcojęzyczne występujące w książce. Dla zilustrowania możliwości języka autor podał kilka przykładowych procedur, które można wykorzystać we własnych programach.

„GFA BASIC” nie jest podręcznikiem uczącym programować w BASIC-u. Jego czytelnik powinien znać przynajmniej podstawy tego języka. Przyda się on najbardziej tym, którzy zdobyli już pewne doświadczenia w pisaniu programów potrzebują poradnika pozwalającego szybko rozstrzygnąć różne niejasności.

SOETO decydując się na wydanie skryptu „GFA BASIC” zdecydowanie wyprzedził inne wydawnictwa, które jeszcze nie dostrzegły zachodzących w coraz większym tempie zmian w dziedzinie sprzętu i oprogramowania. O ile książki o komputerach 8 bitowych można dostać prawie w każdej księgarni, to w poszukiwaniu literatury o 16 bitowych moglibyśmy bezskutecznie wędrować nawet kilka dni. Sytuacja ta musi się zmienić, jeśli chcemy spokojnie myśleć o przyszłości naszej informatyki. Atari 520ST jest bardzo atrakcyjnym komputerem i niewielkie na razie grono jego użytkowników będzie się systematycznie powiększać. Oprogramowanie i sam sprzęt wchodzący w skład rodziny ST jest ciekawym tematem, który warto bliżej przedstawić czytelnikom.

(J.J.)

Stefan Nawrocki, „GFA BASIC”, Stoleczny Ośrodek Techniki Obliczeniowej, Warszawa 1988, Wydanie I, Nakład 5000 egz., Cena 900 zł.

LICZBY

RZECZYWISTE W

Action!

Podobno jedną z wad języka Action! jest brak możliwości wykonywania obliczeń na liczbach rzeczywistych. Jest to wada pozorną, gdyż większość problemów można rozwiązać korzystając wyłącznie z liczb całkowitych. Najlepszym przykładem jest tu język wewnętrzny komputera, który operuje tylko na liczbach całkowitych.

Jeżeli jednak liczby rzeczywiste są niezbędne, to także nie stanowi to żadnego problemu. Przecież procedury do obliczeń na liczbach rzeczywistych (zwanymi w informatyce zmiennoprzecinkowymi) są zawarte w pamięci ROM komputera w tzw. pakiecie matematycznym. Wystarczy więc tylko z nich skorzystać. Trzeba wszakże od razu powiedzieć, że zmniejsza to znacznie szybkość działania programu, gdyż procedury zmiennoprzecinkowe są stosunkowo powolne (jest to główna przyczyna niewielkiej szybkości Atari Basic). Zainteresowanych odsyłam do książki „Mapa pamięci Atari XL/XE. Podstawowe procedury systemu operacyjnego”, w której znajduje się szczegółowy opis procedur zmiennoprzecinkowych.

Po tym wstępie czas na właściwe procedury. Przed obliczeniami zmiennoprzecinkowymi trzeba najpierw zadeklarować odpowiednie zmienne. Ponieważ w Atari każda liczba rzeczywista zajmuje sześć bajtów, to typ REAL zadeklarujemy jako tablicę złożoną z trzech liczb typu CARD. Konieczność zadeklarowania pozostałych zmiennych wynika z wymagań pakietu zmiennoprzecinkowego. Liczby rzeczywiste, na których wykonywane są operacje, przechowywane są w rejestrach FR0 i FR1. W pierwszym z nich przechowywane są także liczby całkowite uzyskane z przekształcenia liczb rzeczywistych. Bufor wejściowy pakietu jest wskazywany

**;Liczby rzeczywiste w Action!
;Wojciech Zientara
;Copyright (c) 1988 Bajtek**

MODULE

```
TYPE REAL=[CARD r1,r2,r3]
REAL fr0=$D4, fr1=$E0
INT fr0int=$D4
CARD inbuff=$F3
BYTE ARRAY lbuff=$580
```

```
PROC AFP=$D800()
PROC FASC=$D8E6()
PROC IFP=$D9AA()
PROC FPI=$D9D2()
PROC FSUB=$DA60()
PROC FADD=$DA66()
PROC FMULT=$DADB()
PROC FDIV=$DB28()
PROC FEXP=$DDC0()
PROC FEXP10=$DDCC()
PROC FLOG=$DECD()
PROC FLOG10=$DED1()
```

przez zawartość rejestru INBUFF, zaś bufor wyjściowy znajduje się w rejestrze LBUFF o długości 255 bajtów. Po dodaniu adresów procedur otrzymujemy pierwszy fragment programu:

Aby umożliwić wprowadzanie i wyprowadzanie liczb rzeczywistych oraz wykonywanie na nich obliczeń, trzeba przede wszystkim napisać procedury przypisania wartości oraz przekształcenia ich do i z postaci zrozumiałej dla kompilatora. Ponieważ liczby rzeczywiste normalnie nie są używane w Action!, to dostęp do nich uzyskamy pośrednio przez zastosowanie wskaźnika. Wskaźnik jest to zmienna typu CARD, która zawiera adres interesującej nas zmiennej dowolnego typu. Zamianę liczby rzeczywistej na całkowitą można wykonać jako funkcję, gdyż zwracana przez nią wartość ma typ normalnie

```
;procedura przypisania: LET a=b
```

```
PROC RAssign(REAL POINTER a,b)
```

```
    b.r1=a.r1
    b.r2=a.r2
    b.r3=a.r3
```

```
RETURN
```

```
;zamiana REAL na INT
```

```
INT FUNC RtI(REAL POINTER r)
```

```
    RAssign(r,fr0)
    FPI()
    RETURN(fr0int)
```

```
;procedura pusta
```

```
PROC Dummy()
RETURN
```

```
;zamiana INT na REAL
```

```
PROC ItR(INT i REAL POINTER r)
```

```
    fr0int=i
    Dummy=IFP
    Dummy()
    RAssign(fr0,r)
    RETURN
```

```
;zamiana REAL na tekst ASCII
```

```
PROC StrR(REAL POINTER r BYTE ARRAY s)
    BYTE i,c
    BYTE POINTER pnt
```

```
    RAssign(r,fr0)
    FASC()
    pnt=inbuff
    WHILE pnt^='0
        DO
            pnt==+1
        OD
        i=0
        DO
            c=pnt(i)
            i==+1
            s(i)=c&#7F
        UNTIL c&#80
        OD
        s(0)=i
    RETURN
```

```
;zamiana tekstu na REAL
```

```
PROC ValR(BYTE ARRAY s REAL POINTER r)
    BYTE i,cix=$F2
```

```
    FOR i=1 TO s(0)
        DO
            lbuff(i-1)=s(i)
        OD
        lbuff(i-1)=0
```

```
inbuff=lbuff
cix=0
AFP()
RAssign(fr0,r)
RETURN
```

stosowany w Action!. Pozostałe zamiany muszą być zapisane jako procedury. Zawarta w ROM procedura zamiany liczby całkowitej na rzeczywistą działa jako funkcja i dla zapewnienia poprawnego jej działania konieczne jest zastosowanie procedury pustej. Warto także zwrócić uwagę na sposób

```
;dodawanie: c=a+b
```

```
PROC RAdd(REAL POINTER a,b,c)
```

```
    RAssign(a,fr0)
    RAssign(b,fr1)
    FADD()
    RAssign(fr0,c)
    RETURN
```

```
;odejmowanie: c=a-b
```

```
PROC RSub(REAL POINTER a,b,c)
```

```
    RAssign(a,fr0)
    RAssign(b,fr1)
    FSUB()
    RAssign(fr0,c)
    RETURN
```

```
;mnozenie: c=a*b
```

```
PROC RMul(REAL POINTER a,b,c)
```

```
    RAssign(a,fr0)
    RAssign(b,fr1)
    FMULT()
    RAssign(fr0,c)
    RETURN
```

```
;dzielenie: c=a/b
```

```
PROC RDiv(REAL POINTER a,b,c)
```

```
    RAssign(a,fr0)
    RAssign(b,fr1)
    FDIV()
    RAssign(fr0,c)
    RETURN
```

przekazywania wartości zmiennych zadeklarowanych jako typ REAL — dotyczy to również innych typów zmiennych deklarowanych przez programistę. Teraz łatwo już napisać procedury wykonujące

```
;podnoszenie e do potegi: b=e^a
```

```
PROC Exp(REAL POINTER a,b)
```

```
    RAssign(a,fr0)
    FEXP()
    RAssign(fr0,b)
    RETURN
```

```
;podnoszenie 10 do potegi: b=10^a
```



```
PROC Exp10(REAL POINTER a,b)
    RAssign(a,fr0)
    FEXP10()
    RAssign(fr0,b)
RETURN

;logarytm naturalny: b=ln a

PROC Log(REAL POINTER a,b)
    RAssign(a,fr0)
    FLOG()
    RAssign(fr0,b)
RETURN

;logarytm dziesiętny: b=log a

PROC CLog(REAL POINTER a,b)
    RAssign(a,fr0)
    FLOG10()
    RAssign(fr0,b)
RETURN

;potęgowanie: c=a^b

PROC Power(REAL POINTER a,b,c)
    Log(a,c)
    RMul(b,c,c)
    Exp(c,c)
RETURN

;zapis REAL na urządzenie

PROC PrintRD(BYTE d REAL POINTER a)
    BYTE ARRAY aux(20)
    StrR(a,aux)
    PrintD(d,aux)
RETURN

;zapis REAL na urządzenie standardowe

PROC PrintR(REAL POINTER a)
    PrintRD(device,a)
RETURN

;zapis REAL na urządzenie
;ze znakiem konca wiersza (RETURN)

PROC PrintRDE(BYTE d REAL POINTER a)
    PrintRD(d,a)
    PutDE(d)
RETURN

;zapis REAL na urządzenie standardowe
;ze znakiem konca wiersza (RETURN)

PROC PrintRE(REAL POINTER a)
    PrintRDE(device,a)
RETURN

;odczyt REAL z urządzenia

PROC InputRD(BYTE d REAL POINTER a)
    BYTE ARRAY aux(128)
    InputMD(d,aux,128)
    ValR(aux,a)
RETURN
```

```
;odczyt REAL z urządzenia
standardowego

PROC InputR(REAL POINTER a)
    InputRD(device,a)
RETURN

bibliotece Action! (takie uporządkowanie znacznie
ułatwia zapamiętanie nazw i pisanie programów).
Do tych operacji należy zadeklarować pomocniczą
tablicę, w której będą przechowywane liczby rze-
czywiste w postaci ciągu znaków ASCII. Znacznie

;przykład - kalkulator

MODULE
REAL POINTER k,l,m

PROC Setup()
    Graphics(0)
    PrintE(" REAL Calculator")
    PutE() PutE() PutE()
    PrintE(" + = dodawanie")
    PrintE(" - = odejmowanie")
    PrintE(" * = mnożenie")
    PrintE(" / = dzielenie")
    PrintE("HELP = koniec")
    PutE() PutE()
    PrintE("Wybierz działanie") PutE()
RETURN

PROC Operate(BYTE o)
    BYTE n,key=$2FC

    key=$FF
    FOR n=1 TO 6
        DO Put($9C) OD
    PrintE("pierwsza liczba ")
    InputR(k)
    PrintE(" druga liczba ")
    InputR(l) PutE()
    IF o=6 THEN
        RAdd(k,l,m)
    ELSEIF o=14 THEN
        RSub(k,l,m)
    ELSEIF o=7 THEN
        RMul(k,l,m)
    ELSE
        RDiv(k,l,m)
    FI
    Print("wynik = ")
    PrintRE(m)
    FOR n=1 TO 6
        DO Put($1C) OD
RETURN

PROC Work()
    BYTE key=$2FC,help=$2DC

    help=0 key=$FF
    DO
        IF help=$11 THEN EXIT FI
        IF key=6 OR key=7 OR
            key=14 OR key=38 THEN
            Operate(key)
        FI
    OD
RETURN

PROC Main()
    Setup()
    Work()
    Graphics(0)
RETURN

większy rozmiar tablicy w procedurze Input wynika
z długości bufora wejściowego używanego przez
system operacyjny komputera. Ostatnia uwaga do-
tyczy zmiennej „device”. Pominięcie jej deklaracji
nie jest pomyłką. Jest to bowiem zmienna bibliote-
czna, która określa numer ustalonego aktualnie
urządzenia wejścia/wyjścia. Normalnie urządze-
niem tym jest edytor (klawiatura + ekran).

Na zakończenie przykład praktycznego zastosowa-
nia liczb rzeczywistych. Jest to prosty kalkulator
czterodziałaniowy. Przykład ten nie wykorzystuje
wszystkich wcześniej opisanych procedur, lecz
służy jedynie do pokazania, w jaki sposób można je
spożytkować.
```

Wojciech Zientara

ZOSTAŃ NIEŚMIERTELNYM!

Kolejną porcję poprawek
zawdzięczamy w całości naszym
czytelnikom. Jako pierwszy w sprawie
POKE'ów napisał do nas 16-letni
Radosław Szybiński z Babimostu.
Proponuje on poprawienie
następujących gier:

SYNTRON

ROZKAZ: DEC \$9D (\$C6, \$9D —
„F” w negatywie, „ESC-CTRL-” w
negatywie), należy zamienić na: LDA
\$9D (\$AD, \$9D — „%” w negatywie,
„ESC-CTRL-” w neg.) i mamy
nieskończoną liczbę „życ”.

ATTACK OF THE MUTANT CAMEL

DEC \$13BE (\$CE,\$BE,\$13 — „N”
w neg., „>” w neg., „CTRL-S”) —
zmniejszanie licznika błędów. Aby
móc popełnić dowolną ich ilość należy
zmienić tę instrukcję na: LDA \$13BE
(\$AD,\$BE,\$13 — „-” w neg., „>” w
neg., „CTRL-S”).

CAVERNS OF MARS

DEC \$1AAC (\$CE,\$AC,\$1A — „N”
w neg., „,” w neg., „CTRL-Z”) —
zamieniamy na: LDA \$1AAC
(\$AD,\$AC,\$1A — „-” w neg., „,” w
neg., „CTRL-Z”).

Piotr Rudnow (16 lat) z Chorzowa
przysłał nam następujące poprawki:

JUMP

DEC \$801F (\$CE,\$1F,\$80 — „N”
w neg., „ESC-STR-L*” „CTRL-” w
neg.) zastępujemy trzema rozkazami:
NOP (\$EA — „i” w neg.).

Postępujemy analogicznie w trzech
kolejnych grach:

ARKANOID

DEC \$31F4 (\$CE,\$F4,\$31 — „N”
w neg., „t” w neg., „1”).

JET SET WILLY

DEC \$46F3 (\$CE,\$F3,\$46 — „N”
w neg., „s” w neg., „F”).

URIDIUM

Życia: DEC \$9E7F (\$CE,\$7F,\$9E
— „N” w neg., „ESC-TAB”, „ESC-
CTRL-+” w neg.)

Bomby: DEC \$9E93 (\$CE,\$93,\$9E
— „N” w neg., „CTRL-P” w neg.,
„ESC-CTRL-+” w neg.)

Sporą porcję POKE'ów dostaliśmy
od 14-letniego Jarosława Chłada z
Kłobucka. Oto niektóre z nich:

FIRE CHIEF

1. Liczba samochodów: DEC \$2F80
(\$CE,\$80,\$2F — „N” w neg.,
„CTRL-” w neg., „/”)

2. Kombinezony: DEC \$5047
(\$CE,\$47,\$50 — „N” w neg.,
„G”, „P”)

3. Paliwo: ciąg rozkazów:

JMP \$304A

DEC \$2F81

(\$4C,\$4A,\$30,\$CE,\$81,\$2F — „L”,
„J”, „O”, „N” w neg., „CTRL-A” w
neg., „/”)

4. Woda: ciąg rozkazów:

STA \$5F3F

DEC \$5DB9

(\$8D,\$3F,\$5F,\$CE,\$B9,\$5D —
„CTRL-M” w neg., „?” „SHIFT-”,
„N” w neg., „9” w neg., „SHIFT-”)

Wszystkie rozkazy DEC (\$CE- „N”
w neg.) zamieniamy na LDA (\$AD-
„-” w neg.) i mamy nieskończoną
liczbę danych atrybutów.

DIAMONDS

DEC \$0605 (\$CE,\$05,\$06 — „N”
w neg., „CTRL-E”, „CTRL-F”) —
występuje w programie dwa razy. W
obu przypadkach zamieniamy DEC
(\$CE — „N” w neg.) na LDA
(\$AD — „-” w neg.)

AIR STRIKE II

1. DEC \$0665 (\$CE,\$65,\$06 — „N”
w neg., „e”, „CTRL-F”) —

2. DEC \$0664 (\$CE,\$64,\$06 — „N”
w neg., „d”, „CTRL-F”) —

3. DEC \$0666 (\$CE,\$66,\$06 — „N”
w neg., „f”, „CTRL-F”) —

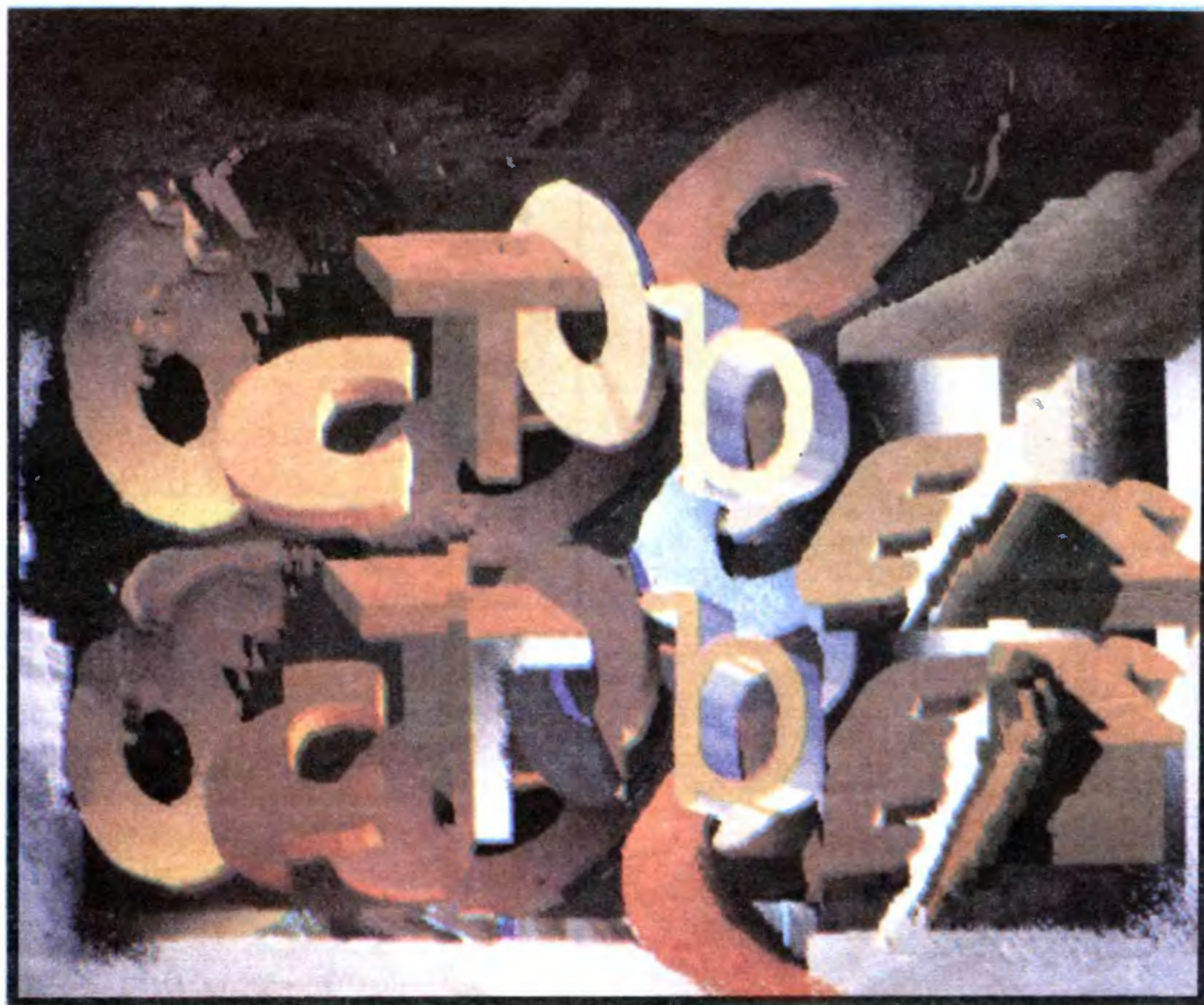
Zamieniamy wszystkie rozkazy: DEC
(\$CE — „N” w neg.) na LDA
(\$AD — „-” w neg.)

SUPER PAC-MAN

DEC \$05F6 (\$CE,\$F6,\$05 — „N”
w neg., „v” w neg., „CTRL-E”) —

zmniejszanie limitu „życ”. Zamieniamy DEC
(\$CE — „N” w neg.) na LDA
(\$AD — „-” w neg.)

Tomasz Wiśniewski



MATEMATYCZNE PODSTAWY SPOSOBU ZAPISU LICZB W POSTACI ZMIENNOPRZECINKOWEJ (CZ. 2)

Powróćmy obecnie do rozpoczętych przed miesiącem rozważań, wyjaśniając, jak zapisane są liczby w pamięci Spectrum.

Jeżeli w praktyce ilość wyrazów równania (2) jest skończona, to ile tych wyrazów należy uwzględnić, aby dana liczba zapisana została z określoną dokładnością? W komputerze ZX Spectrum każda liczba może być podawana z dokładnością ośmiu liczb po przecinku. Ile więc cyfr po przecinku należy uwzględnić w zapisie binarnym, aby tę dokładność zachować?

Jak wykazano w (1), dokładność liczby dziesiętnej zapisanej binarnie obliczonej przy pomocy równania (5) określona jest wyrażeniem:

$$\delta = 2^{-(n+1)} \quad (10)$$

czyli, aby otrzymać liczbę z dokładnością 9-ciu cyfr po przecinku, należy uwzględnić 32 znaki liczby binarnej, gdyż:

$$\delta = \frac{1}{2^{32}} = 1.16415 \cdot 10^{-10} \approx 10^{-10}$$

Wynik ten gwarantuje, że właśnie cyfra 9-ta po przecinku jest jeszcze liczbą pewną. Dlatego właśnie w komputerze ZX Spectrum mantysa „m” jako liczba ułamkowa ustalana jest w oparciu o 32 bity, aby, między innymi, zaokrąglanie cyfr bazowało także na cyfrach pewnych. Konsekwencją tego niejako jest to, że aby te 32 bity zapisać w komórkach 8-mio bitowych pamięci, należy użyć aż 4 komórki (po 8 bitów w każdej). Ale to nie wszystko. Z równania (5) wiemy, że aby liczba „a” była jednoznacznie określona, poza mantysą „m” trzeba jeszcze podać — a więc zapisać w pamięci — wykładnik potęgowy „n”. W tym właśnie celu, użyta jest 5-ta komórka pamięci, w której wpisywana jest cyfra „p” wg równania

$$p = 128 + n \quad (10')$$

Tak więc dla „p” przyjmującego wartości liczb całkowitych z przedziału $(0 \div 255)$ „n” przyjmuje odpowiednio wartości $(-128 \div +127)$.

Tym sposobem uzasadniliśmy, dlaczego do zapisu dowolnych liczb w postaci zmiennoprzecinkowej wystarczy użyć tylko pięć komórek pamięci. W taki sposób zapisywane są wszystkie liczby, na których system komputera ZX Spectrum może wykonywać obliczenia, a więc liczby całkowite i ułamkowe, dodatnie i ujemne należące do przedziału $\langle -1.70141183469 \text{ (9) E}+38, 1.70141183469 \text{ (9) E}+38 \rangle$ z wyłączeniem liczb całkowitych dodatnich i ujemnych z przedziału $\langle -65535, +65535 \rangle$, które zapisywane są inaczej oraz liczb mniejszych od $2.938735877 \text{ E}-39$ lecz większych od $-2.938735877 \text{ E}-39$, które rozumiane są przez system komputera jako liczby o wartości zero.

Wyjaśniliśmy zarówno matematyczne podstawy zapisu liczb w postaci zmiennoprzecinkowej ze szczególnym uwzględnieniem komputera ZX Spectrum, jak też potrzebę użycia 5-ciu bajtów pamięci, aby daną liczbę zapisać z określoną dokładnością.

Dla wyczerpania tematu pozostał jeszcze do omówienia problem obliczania liczby w oparciu o dane zapisane w omawianych 5-ciu bajtach. W tym celu powróćmy do równania 5, na bazie którego obliczana jest mantysa „m”, zapisana w czterech komórkach pamięci. Wiemy, że mantysa „m” jako liczba ułamkowa w ogólnym przypadku przedstawiona jest jako ciąg nieskończony. W naszym przypadku uwzględniane są tylko 32 bity, które zostają podzielone na 4 równe części i zapisane w kolejnych czterech komórkach pamięci (w 5-cio bajtowym formacie są to 2-g, 3-cia, 4-ta i 5-ta komórka), przy czym ósmy bit drugiej komórki jest zwolniony dla zapisu bitu znaku. Odczytując każdą z tych czterech komórek otrzymamy 4-ry oderwane i niezależne liczby całkowite będące niejako „kodami” poszczególnych ułamkowych części, ułamkowej mantysy „m”.

Aby ustalić rzeczywistą wartość mantysy, musimy tym liczbom przyporządkować zarówno pozycję w szeregu nieskończonym, jak też i ich wartość. Pozycja liczby w szeregu określona jest pozycją

komórki w 5-cio bajtowym formacie pamięci, w której dana liczba jest zapisana, wartość natomiast należy ustalić na podstawie równania 3 wyrażonego dla odróżnienia przy pomocy innych symboli, a mianowicie:

$$b = \frac{k}{2^r} \quad (11)$$

gdzie:

- b — wartość danej części mantysy „m”,
- k — liczba zapisana na danej pozycji,
- r — waga najwyższego bitu danej pozycji (dla liczby zapisanej w komórce 2-giej 5-cio bajtowego formatu $r = 8$, w 3-ciej $r = 16$, w 4-tej $r = 24$ i w 5-tej $r = 32$).

Jeżeli kolejne liczby zapisane w czterech komórkach pamięci oznaczymy przez: b_2, b_3, b_4, b_5 , wówczas równanie (3) można wyrazić w postaci:

$$\frac{a}{2^n} \text{ odpowiada } b_2 + b_3 + b_4 + b_5 \quad (12)$$

Uwzględniając (11) wyrażenie (12), możemy zapisać:

$$\frac{a}{2^n} = \frac{b_2}{2^8} + \frac{b_3}{2^{16}} + \frac{b_4}{2^{24}} + \frac{b_5}{2^{32}} = m \quad (13)$$

Gdybyśmy tę samą mantysę „m” chcieli wyrazić tylko jedną liczbą dziesiętną, odpowiadającą 32-u bitowej liczbie binarnej, wówczas oznaczając tę liczbę przez „B”, równanie (13) będzie:

$$\frac{a}{2^n} = \frac{B}{2^{32}} \quad (14)$$

przy czym „B” nie jest równe „a”, a „n” spełnia warunek (9).

I to właściwie wszystko, jeżeli chodzi o ustalenie „źródłowej” wartości mantysy „m” zapisanej liczby. Aby jednak ustalić „źródłową” wartość liczby, trzeba jeszcze obliczyć wykładnik potęgowy „n”. W tym celu wystarczy odczytać liczbę „n1” z pierwszej komórki pamięci i na podstawie znanego równania

$$n_1 = 128 + n$$

obliczyć wykładnik wyrażony równaniem:

$$n = n_1 - 128$$

Mając „m” i „n” z równania (5), w prosty sposób obliczymy już liczbę pierwotną „a”.

Przykład:

Niech zapis w 5-ciu komórkach pamięci będzie:
komórka 1-132 liczba informująca o wykładniku „n”
komórka 2-24
komórka 3-219 liczby reprezentujące mantysę „m”
komórka 4-244
komórka 5-136
Jaka to liczba?

1. Obliczamy wykładnik „n” z równania

$$132 = 128 + n$$

$$\text{stąd} \quad n = 4$$

2. Obliczamy wartości ułamkowe poszczególnych pozycji mantysy „m”

$$b_2 = \frac{24 + 128}{2^8} = 0.59375$$

$$b_3 = \frac{219}{2^{16}} = 0.00334167$$

$$b_4 = \frac{244}{2^{24}} = 0.0000145435$$

$$b_5 = \frac{136}{2^{32}} = 0.0000000316648$$

3. Ustalamy wartość mantysy „m”

$$m = b_2 + b_3 + b_4 + b_5 = 0.5971062$$

4. Na podstawie równania (5) obliczamy liczbę „a”

$$a = m \cdot 2^n = 0.5971062 \cdot 2^4 = 9.5537$$

Uwaga!

Dodatknie cyfry 128 w liczniku wyrażenia b_2 wynika ze sposobu zapisu przez system komputera pierwszego z czterech bajtów ułamkowej mantysy „m”, polegającego na tym, że z bajtu tego zostaje „odrzucony” składnik 0.1 binarnie, tzn. 0.5 dziesiętnie, celem „zarezerwowania” ósmego bitu tego bajtu na informacji o znaku liczby. Nieuwzględnienie tego bitu jest równoznaczne z nieuwzględnieniem składnika $2^7 = 128$. Zagadnienie to zostanie omówione szerzej w jednym z następnych numerów.

Poznaliśmy matematyczne podstawy zapisu liczb w postaci zmiennoprzecinkowej oraz omówiliśmy ogólnie sposób zapisu tych liczb w 5-cio bajtowym formacie pamięci. Wydaje się jednak, że problemem nie mniej interesującym jest zagadnienie techniki obliczania przez system komputera poszczególnych bajtów jako danych do zapisu liczby a pamięci.

Jest to jednak obszerne zagadnienie i dlatego zostanie omówione osobno.

Piotr Sumara

LITERATURA:

1. A.W. Mostowski — *Algebry Boole'a i ich zastosowanie*
2. SINCLAIR ZX Spectrum User Guide — instrukcja obsługi
3. K. Kuryłowicz, D. Madej, K. Marasek — *Przewodnik po ZX Spectrum*

SCROLL W GÓRĘ

Prezentowany program umożliwi płynne (a nie skokowe jak w przypadku „scroll”) przesuwanie zawartości ekranu do góry.

Wykorzystana została przy tym procedura z ROM-u zawarta pod adresem 8874 (#22AA), przeliczająca współrzędne punktu na jego adres w pamięci ekranu. Program zamienia

współrzędne początku danej linii (przechowywane w rejestrze BC) oraz linii o jedną pozycję wyżej na adresy (umieszczane w rejestrze HL) i przesuwają całą linię (32 bajty) pod nowy adres. Tę procedurę uruchamia **RANDOMIZE USR 32000**. Procedura, zaczynająca się od adresu 32040, przesuwa zawartość pamięci ekranu wraz z atrybutami pod adres 36384, skąd można ją z powrotem „wsunąć” płynnie na ekran przez **RANDOMIZE USR 32052**, a następnie pokolorować przez **RANDOMIZE USR 32080**. Działanie podprogramu, rozpoczynającego się od adresu 32052, oparte jest na podobnej zasadzie z tym, że „nową” linię stanowi zawsze najniższa linia ekranu. Kolorowanie obrazu (**RANDOMIZE USR 32040**) polega na przemieszczeniu zawartości pamięci ekranu rozkazem **ldir**. O ile pierwsza z procedur może pracować samodzielnie, to pozostałe wymagają obecności pierwszej. Program może obsługiwać obrazy o długości do 22 wierszy. Wszystkie procedury napisane zostały w kodzie maszynowym, lecz można je wczytać z poziomu BASIC-a korzystając z poniższego programu ładującego, który zawiera również przykład praktycznego zastosowania programu.

Grzegorz Krupa

```

30 REM CLEAR FAR 31999
30 REM FOR n=32000 TO 320
NEXT n: READ a: POKE n,a: LET s=s+a:
30 IF s<>3944 THEN PRINT "Bład
w linii: 40": STOP
40 DATA 193,5,197,205,170,34
50 DATA 176,5,197,205,170,34
60 DATA 176,5,197,205,170,34
70 DATA 176,5,197,205,170,34
80 DATA 176,5,197,205,170,34
90 DATA 176,5,197,205,170,34
100 REM Przykład
100 CIRCLE 128,87,80: CIRCLE 12
87,87: RANDOMIZE USR 32040: CL
110 RANDOMIZE USR 32052
    
```

SORTOWANIE PO POLSKU

Każdy użytkownik Spectrum, który chciał zrobić chociażby komputerowy spis telefonów znajomych i mozolnie wprowadził zaprojektowane przez siebie polskie litery natyka się na problem: jak zmusić komputer aby układał wyrazy według polskiego alfabetu?

Kłopot jest spowodowany tym, że Spectrum (a także inne komputery) SORTUJE WYRAZY WEDŁUG kodów ASCII POSZCZEGÓLNYCH liter, a polskie znaki nie są w nim umieszczone. Najprostszym rozwiązaniem mogłoby się wydawać przerobienie generatora znaków przez umieszczenie w nim kolejno liter polskiego alfabetu, jednak to utrudniłoby korzystanie z klawiatury. W takim przypadku jednym z lepszych rozwiązań jest dołączenie do programu dwóch poniższych procedur. Mają one też dodatkową zaletę, że można z ich pomocą sortować wyrazy np. po niemiecku czy rosyjsku (po niewielkich przerób-

kach dadzą się też przystosować do innych komputerów). Oto one:

Teraz o praktycznym wykorzystaniu procedur: Przed programem głównym należy umieścić w tablicy alfabet, wg którego wyrazy będą sortowane (ze spacją na początku!) a następnie przekodować wyrazy (najlepiej zrobić to zaraz po wczytaniu wyrazu, gdyż program ten działa dosyć wolno, a na taśmie przechowywać wyrazy zakodowane), np.:

```
LET z$=a$(x): GOSUB 9990: LET a$(x)=z$
```

gdzie tablica a\$ zawiera wyrazy, a x to zmienna sterująca pętlą. Teraz następuje sortowanie (aby je przyspieszyć, pożądane jest użycie programu maszynowego — takiego jak np. z nr 5/87 „Komputera”) i wysyłanie danych na ekran, np.:

```
LET z$=a$(x): GOSUB 9995: PRINT z$
```

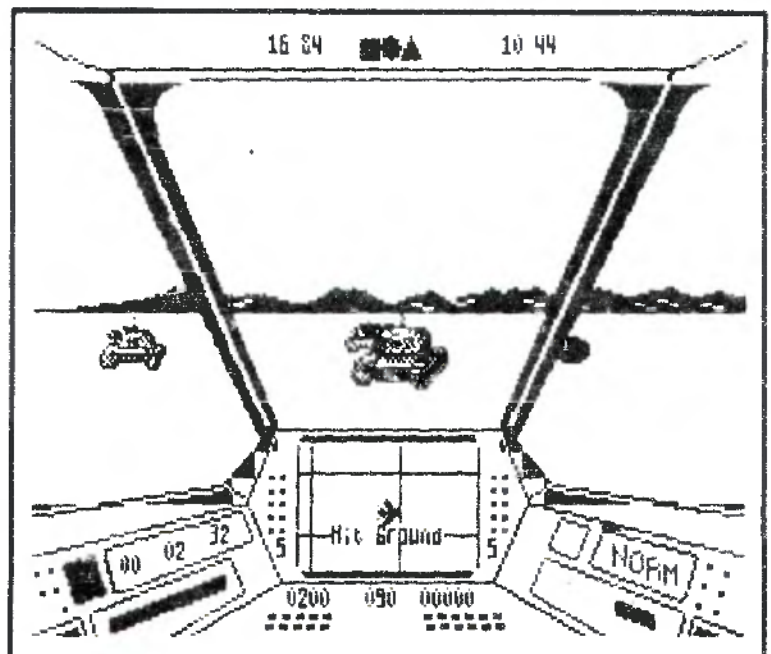
Myślę, że teraz nasza „książka telefoniczna” będzie lepiej spełniała swoje zadanie, wykazując przy okazji iż Polacy nie gęsi...

Andrzej Wawrzyńczyk

```

9990 FOR i=1 TO LEN z$: FOR j=1
TO LEN w$: IF z$(i)=w$(j) THEN L
ET z$(i)=CHR$(63+j): NEXT j: GO
TO 9992: REM 35
9991 NEXT j: REM 92
9992 NEXT i: RETURN: REM CC
9995 FOR i=1 TO LEN z$: LET zmie
nna=CODE z$(i)-63: LET z$(i)=w$(
zmienna): REM 3F
9995 NEXT i: RETURN: REM D0
9999 REM nazwy zmiennych można z
mienić na inne, aby przystosować
je do programu głównego
    
```

CAŁEGO EKRANU. DRUKOWANIE



Często widziałeś na pewno gry lub inne programy, w których podczas ładowania można podziwiać obrazek znajdujący się na ekranie.

Próbowałeś może także skopiować ten obrazek na drukarce, lecz pewnie nie udało ci się to w całości — instrukcja COPY „obcina” dwa dolne wiersze. Jeśli zamiast „COPY” użyjesz przedstawionego poniżej programu, uzyskasz wydruk całego ekranu — razem z zawartością tych dwóch wierszy. Pamiętaj tylko, żeby nie zatrzymywać programu, gdy na ekranie znajduje się obrazek, który chcesz uwiecznić, gdyż komunikat o zatrzymaniu programu skasuje zawartość tych linii i będziesz musiał jeszcze raz wczytywać obrazek z magneto fonu. Ten sam obrazek możesz kopiować wielokrotnie naciskając dowolny klawisz, za wyjątkiem „n”, które spowoduje załadowanie następnego obrazka.

(ts)

```

10 FOR n=USR "a" TO USR "a"+5:
REM C2
20 READ a: POKE n,a: NEXT n: R
EM 2B
30 DATA 6,192,243,195,175,14:
REM B5
40 LOAD "SCREEN$": REM 0E
50 RANDOMIZE USR USR "a": REM
5B
60 PAUSE 0: LET a$=INKEY$: REM
1E
70 IF a$="n" OR a$="N" THEN GO
TO 40: REM 3E
80 GO TO 50: REM EF
    
```


SPRITE'Y, RAMKA I BASIC

Ciekawe efekty graficzne, jakie pozwala nam uzyskać układ kontrolera wizji C-64, stały się ostatnio domeną grup młodych ludzi zajmujących się tworzeniem tzw. „demosów”, rozpowszechnianych poprzez duże sieci komputerowe.

Co ciekawsze efekty wizualne osiągnane w tych programkach biją niejednokrotnie na głowę duże i rozbudowane programy wytwórni-rekinów takich jak Electronic Arts czy Access. Zobaczmy więc jak wygląda taki program od kuchni...

Sprite'y wyświetlane na ramce są jednym z przykładów wykorzystania przerw graficznych. W normalnych warunkach pozycje sprite'a możemy określić dowolnie, jednakże, jeśli znajdzie się on na ramce, nie będzie on na ekranie widoczny. Dzięki przerwaniom graficznym ekran można rozszerzyć, co ilustruje program 1. W procedurze RAST2 ekran jest rozszerzany do 25 wierszy. W momencie wyświetlania 25 wiersza generowane jest przerwanie graficzne i wykonywana jest procedura RAST1 zwięzająca ekran do 24 wierszy. W ten sposób układ graficzny nie jest w stanie odnaleźć końca ekranu (którym jest normalnie właśnie 25 wiersz; jeśli tylko kolory samej ramki i tła nie są jednakowe, to z pewnością zauważysz wyraźne rozszerzenie się ekranu).

Dużo trudniejszą, ale nie niemożliwą sprawą jest rozszerzenie poziome ekranu. Wymaga to zmiany liczby kolumn z 40 na 39 w każdej linii, którą chcemy mieć rozszerzoną. W tak rozszerzonym obszarze mogą znajdować się jedynie sprite'y. Ekran możemy teraz wykorzystać w dowolny sposób — wyświetlając dowolny tekst czy np. czas.

W programie 2 chciałbym zaprezentować Czytelnikom nieco inny sposób wykorzystania sprite'ów na ramce — wyświetla on zamiast tekstu numer aktualnie wykonywanej linii Basic. Pomimo korzystania ze sprite'ów program przez cały czas „pamięta” ich właściwe pozycje; w naszym programie w BASIC możemy więc korzystać ze sprite'ów, zmieniać tryby graficzne, kolory, itp., i nie będzie to przeszkadzało głównej funkcji wyświetlania numeru linii na ramce. Numer aktualnie wykonywanej linii programu znajduje się w wektorze \$39/\$3A. Ponieważ BASIC jest na tyle szybki, że czasem numer linii jest nie do odczytania, rozszerzyłem w swoim programie działanie klawisza CTRL — jego wciśnięcie powoduje zatrzymanie wykonywania programu (napisanego w BASIC). Program 2 wykorzystuje pamięć od adresu \$C000 do \$C6FF oraz komórki SFD, SFF i SFF.

Jacek Czok

Od redakcji: Powyższy program (nr 2) stanowi najlepszy przykład, że można połączyć przyjemne z pożytecznym — program ten to po prostu dodatkowe i efektownie wykonane polecenie TRACE...

Przy okazji chcielibyśmy sprostować błąd jaki wkraść się z naszej winy do artykułu tego samego autora w numerze specjalnym BAJTKA TYLKO O COMMODORE — przerwania graficzne można zablokować poprzez polecenie LDA #\$FF, a nie jak podaliśmy przez LDA \$7F (artykuł „Przerwania graficzne”, str. 13).

Serdecznie przepraszamy.

Wydruk wersji maszynowej programu 1 został sporządzony za pomocą programu Commodore Macroassembler Development System.

```

00001 0000          ; PROGRAM NR 1
00002 0000          ;
00003 0000          *= $C000
00004 C000 78      SEI
00005 C001 A9 01   LDA #$01
00006 C003 8D 1A D0 STA $D01A
00007 C006 A9 17   LDA #<RAST1
00008 C008 8D 14 03 STA $0314
00009 C00B A9 C0   LDA #>RAST1
00010 C00D 8D 15 03 STA $0315
00011 C010 A9 7F   LDA #$7F
00012 C012 8D 0D DC STA $DC0D;BLOKADA IRQ
00013 C015 58      CLI
00014 C016 60      RTS
00015 C017 AD 19 D0 RAST1 LDA $D019;ODEBRANIE
00016 C01A 8D 19 D0 STA $D019; PRZERWANIA
00017 C01D A9 13   LDA #$13
00018 C01F 8D 11 D0 STA $D011;24 LINIE
00019 C022 A9 20   LDA #$20
00020 C024 8D 12 D0 STA $D012
00021 C027 A9 34   LDA #<RAST2
00022 C029 8D 14 03 STA $0314
00023 C02C A9 C0   LDA #>RAST2
00024 C02E 8D 15 03 STA $0315
00025 C031 4C B1 EA JMP $EAB1
00026 C034 AD 19 D0 RAST2 LDA $D019;ODEBRANIE
00027 C037 8D 19 D0 STA $D019; PRZERWANIA
00028 C03A A9 1B   LDA #$1B
00029 C03C 8D 11 D0 STA $D011;25 LINII
00030 C03F A9 F9   LDA #$F9
00031 C041 8D 12 D0 STA $D012
00032 C044 A9 17   LDA #<RAST1
00033 C046 8D 14 03 STA $0314
00034 C049 A9 C0   LDA #>RAST1
00035 C04B 8D 15 03 STA $0315
00036 C04E 4C 31 EA JMP $EA31;KEYBOARD
00037 C051          .END
    
```

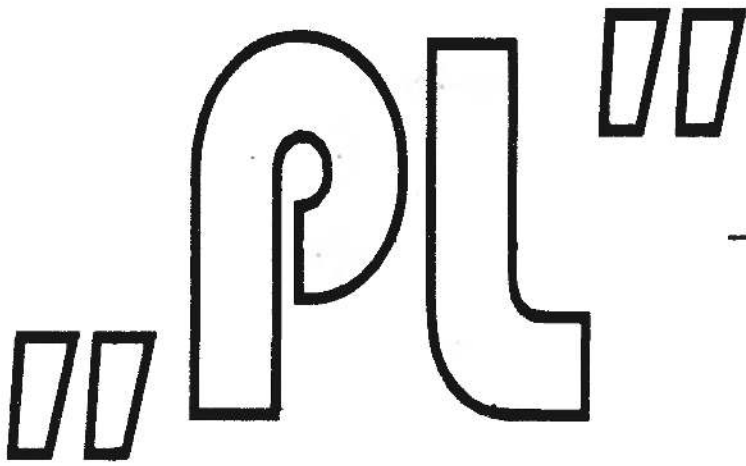
```

2E 100 REM *** PROGRAM NR 1 ***
38 105 :
07 110 REM *** JACEK CZOK ***
F8 115 :
7B 120 :
DF 125 FOR I=49152 TO 49232
BF 130 READ A:CK=CK+A:POKE I,A
2A 135 NEXT
69 140 IF CK<>9021 THEN PRINT "ZLE DANE !":
    STOP
6E 145 SYS 49152
BA 150 :
C6 155 DATA 120,169,001,141,026,208,169,023
    ,141,020,003,169,192,141,021,003
70 160 DATA 169,127,141,013,220,088,096,173
    ,025,208,141,025,208,169,019,141
BF 165 DATA 017,208,169,032,141,018,208,169
    ,052,141,020,003,169,192,141,021
5E 170 DATA 003,076,129,234,173,025,208,141
    ,025,208,169,027,141,017,208,169
10 175 DATA 249,141,018,208,169,023,141,020
    ,003,169,192,141,021,003,076,049,234
    
```

```

19 100 REM *** PROGRAM 2 ***
38 105 :
07 110 REM *** JACEK CZOK ***
F8 115 :
7B 120 :
4E 125 FOR X=49152 TO 49551:READ A:CK=CK+A:P
    OKE X,A:NEXT
6B 130 IF CK<>46913 THEN PRINT "ZLE DANE !":
    STOP
7A 135 :
C4 140 PRINTCHR$(147):PRINT:PRINT" ZAPIS PRO
    GRAMU: ":PRINT
14 145 PRINT"POKE 43,0:POKE 44,192:POKE 45,1
    44:POKE 46,193":PRINT
EB 150 PRINT"SAVE"+CHR$(34)+"NAZWA"+CHR$(34)
    +",8,1:REM DYSKIETKA":PRINT
96 155 PRINT"SAVE"+CHR$(34)+"NAZWA"+CHR$(34)
    +",1,1:REM KASETA":PRINT:PRINT
81 160 PRINT:PRINT" WCZYTYWANIE PROGRAMU: ":
    PRINT
6C 165 PRINT"LOAD"+CHR$(34)+"NAZWA"+CHR$(34)
    +",8,1:REM DYSKIETKA":PRINT
F3 170 PRINT"LOAD"+CHR$(34)+"NAZWA"+CHR$(34)
    +",1,1:REM KASETA":PRINT:PRINT
25 175 PRINT" URUCHAMIANIE PROGRAMU: SYS 491
    52":NEW
FC 180 :
01 185 DATA 120,169,001,141,026,208,169,019,
    141,020,003,169,193,141,021,003
ED 190 DATA 169,127,141,013,220,169,051,133,
    001,162,000,138,157,000,196,157
9D 195 DATA 000,197,157,000,198,232,208,244,
    160,128,169,209,132,254,133,255
43 200 DATA 160,196,134,252,132,253,169,010,
    141,080,193,160,000,177,254,129
5B 205 DATA 252,200,230,252,230,252,230,252,
    152,041,007,208,240,024,165,252
36 210 DATA 105,040,133,252,165,253,105,000,
    133,253,206,080,193,208,222,174
BE 215 DATA 010,003,172,011,003,142,091,193,
    140,092,193,162,123,160,192,142
B8 220 DATA 010,003,140,011,003,169,055,133,
    001,088,096,072,173,141,002,041
B1 225 DATA 004,208,249,104,108,091,193,173,
    025,208,141,025,208,160,046,185
AC 230 DATA 140,193,153,000,208,136,048,012,
    192,026,208,002,136,136,192,018
18 235 DATA 240,250,208,235,173,157,193,041,
    119,009,008,141,017,208,173,079
62 240 DATA 193,141,000,221,169,249,141,018,
    208,169,019,141,020,003,169,193
D7 245 DATA 141,021,003,166,057,164,058,134,
    254,132,255,169,001,133,253,160
50 250 DATA 008,162,015,056,165,254,249,081,
    193,133,254,165,255,249,082,193
D4 255 DATA 133,255,232,176,238,165,254,121,
    081,193,133,254,165,255,121,082
E4 260 DATA 193,133,255,224,016,208,008,165,
    253,240,004,162,026,208,004,169
ED 265 DATA 000,133,253,138,072,152,074,170,
    104,157,248,195,136,136,016,193
57 270 DATA 076,049,234,173,025,208,141,025,
    208,169,019,141,017,208,160,046
24 275 DATA 185,000,208,153,140,193,185,093,
    193,153,000,208,136,048,008,192
5A 280 DATA 026,208,237,136,136,208,233,173,
    000,221,141,079,193,169,148,141
C0 285 DATA 000,221,169,135,141,020,003,169,
    192,141,021,003,076,129,234,151
EB 290 DATA 000,001,000,010,000,100,000,232,
    003,016,039,000,000,176,016,160
94 295 DATA 016,144,016,128,016,112,016,000,
    000,000,000,000,000,000,147,048
EB 300 DATA 000,000,031,004,031,000,000,000,
    000,000,031,000,000,003,000,000
05 305 DATA 000,000,000,000,006,006,006,006,
    006,006,006,006,000,000,000,000
    
```


Z KOŃCÓWKĄ



Od pana Sławomira Sobańca z Łodzi otrzymaliśmy do testowania program o nazwie PRINTFOX PL — spolszczoną wersję programu PRINTFOX, dla Commodore 64.

PRINTFOX PL jest podobnie jak jego pierwowzór graficznym edytorem tekstu pozwalającym na dowolne w zasadzie formatowanie tekstu i jego łączenie z grafiką. Dzięki temu użytkownik może tworzyć dokumenty o dowolnym wystroju graficznym i przeznaczane na dowolne okazje — od prywatnych listów czy reklamówek do dyplomów władznie.

Podstawową cechą odróżniającą ten edytor od oryginału jest konwersja rysunków wykonanych za pomocą innych programów: PRINTSHOP, NEWSROOM, PRINTSHOP GRAPHIC LIBRARY, RAINBOW PAINTER, BLAZING PADDLE, DOODLE itp. Oprócz tego autor opracował specjalny edytor do czcionek dostosowując go do potrzeb polskiego użytkownika.

Sterowanie programem nie różni się w zasadzie od oryginału. Zamiast niemieckich napisów na ekranie ukazują się komunikaty w języku polskim. Podczas testu natknąłem się jednak na jeden komunikat niemiecki; był on prawdopodobnie spowodowany faktem, iż program zakłada stosowanie drukarki MPS-803 przez użytkownika

podczas gdy w omawianym przypadku stosowałem drukarkę NL-10.

Wraz z 5 dyskietkami na których zapisany jest program główny (1 strona dyskietki) oraz czcionki i rysunki otrzymaliśmy także odbitkę kserograficzną instrukcji obsługi sprzedawanej wraz z programem. Liczy sobie ona 91 stron i lwią jej część obejmuje wydruki wszystkich rysunków i czcionek. W chwili obecnej użytkownik może korzystać z 34 dostępnych w postaci zbiorów na dyskietce krojów pisma. Wszystkie opisywane czcionki oraz prezentowane rysunki są starannie opisane; instrukcja obsługi jest podzielona na rozdziały których numeracja odnosi się do konkretnej strony dyskietki. Dzięki temu odnalezienie samego rysunku czy czcionki jest stosunkowo proste i szybkie. Wszystkie prezentowane czcionki obejmują znaki polskie. Zgodnie z oświadczeniem autora trwają prace nad 20 nowymi krojami pisma.

Tajniki korzystania z programu są podzielone na trzy etapy: EDYTOR TEKSTU, EDYTOR GRAFICZNY i EDYTOR SPRITE'OW. Proste i jasne

opisy pozwalają na szybkie uruchomienie programu i korzystanie z jego możliwości. Nieco słabiej przedstawiono informacje odnoszące się do drukowania (pół strony), co może nastręczyć pewnych trudności podczas pracy. Program współpracuje z drukarkami Commodore VIC 1525, MPS-801 i MPS-803, urządzeniami firmy STAR i EPSON oraz niektórymi rzadziej spotykanymi modelami (np. SHI-NWA).

Zasada pracy tego edytora (tak jak i oryginału) jest nieco odmienna niż w normalnych programach tego typu. Po pierwsze użytkownik pracuje na kilku ekranach graficznych, po drugie wydruk jest podzielony na dwie fazy. W każdej fazie drukowana jest połowa strony. Dostępne są w zasadzie wszystkie niezbędne polecenia od ustawienia tabulatorów i zapisywania własnych formatów po wyszukiwanie fraz tekstu czy pojedynczych słów, znaków itp. Użytkownik może adjustować tekst, operować blokami, definiować wiele czcionek i różne parametry (np. odstęp międzywierszowy) na jednej stronie, pośrodkować tekst itp. Możliwe jest także wydawanie poleceń odnoszących się do stacji dysków. Wydruk tekstu i rysunków zależy w dużej mierze od stosowanej drukarki; dyskietka systemowa zawiera programy sterujące dla drukarek EPSON, STAR oraz rodziny MPS. Możliwe jest także dopasowanie innej drukarki dzięki programowi SETUP-PL.

Podczas testowania programu natknąłem się jednak na dość istotny i niewygodny moim zdaniem brak, zwłaszcza dla użytkowników, którzy chcieliby przenieść swoje własne, wcześniej wykonane rysunki do PRINTFOX PL. Na dyskietkach nie ma zapisanych wszystkich traslatorów umożliwiających zmianę grafiki wykonanej np. za pomocą programu NEWS ROOM na rysunek czytelny przez omawiany program. Nie wspomina także nic o tym instrukcja obsługi z wyjątkiem akapitu mówiącego, że PRINTFOX PL akceptuje kilka powszechnie znanych formatów m.in. HI EDDI i ART STUDIO. Znalazłem co prawda traslatory PRINT SHOP — PRINTFOX oraz PRINT MASTER-

PRINTFOX lecz wydaje mi się, że dokładna informacja (choćby w instrukcji obsługi) na ten temat oraz uzupełnienie programów brakujących jest niezbędne. W przeciwnym wypadku — musimy ograniczyć się do grafiki zapisanej na dyskietkach dodatkowych.

Drugą uwagą jaką miałbym do testowanego programu jest brak omówienia niektórych programów o dość niejasnym przeznaczeniu i tytule. Dla przykładu nazwa RAINBOW PAINTER może sugerować użytkownikowi, że jest to translator tymczasem jest to zgoła zupełnie co innego. Ponadto dwa wspomniane wyżej translatory wyświetlają napisy w języku angielskim co niekoniecznie musi być rozumiane przez polskiego użytkownika spodziewającego się po końcówce PL całkowicie spolszczonej wersji programu.

Sumując PRINTFOX PL wydaje się być narzędziem dość uniwersalnym i pożytecznym aczkolwiek ma on jeszcze pewne niedociągnięcia. Duża ilość rysunków i dostępnych czcionek, polskie opisy wraz z instrukcją (wymagającą może rozszerzenia niektórych części) mogą stanowić dla nabywcy pewien magnes, zwłaszcza dla zainteresowanych grafiką i jej łączeniem z tekstem. Koncepcję przetworzenia rysunków z innych programów tak samo jak opracowanie generatora czcionki polskiej należy uznać za ciekawą i udaną; dołączenie do programu pełnego zestawu translatorów, a do instrukcji podręcznika o ich obsłudze uczyni ten program na pewno bardziej uniwersalnym.

Klaudiusz Dybowski

WIEŚCI • WIEŚCI • WIEŚCI • WIEŚCI • WIEŚCI • WIEŚCI

AIRBORNE RANGER to gra, która ukazała się niedawno na rynkach zachodnich. Sam temat jest już dość ograny — działalność dywersyjna na tyłach nieprzyjaciela. Pikanterii dodaje tu fakt, że autorem jest znana szeroko z wielu doskonałych programów symulacyjnych amerykańska firma MICROPROSE. AIRBORNE RANGER daje do wyboru 12 misji i jest typową grą zręcznościową. Sądząc z reklamówek grafika jest jedną z mocniejszych stron tej gry.

CHUCK YEAGER'S ADVANCED FLIGHT SIMULATOR, to następny symulator lotu dla Commodore 64. Na uwagę zasługuje tu fakt, że jest to jeden z nielicznych programów gdzie więcej nacisku położono na parametry lotu (użytkownik może wybrać sobie jeden z 15 samolotów — od PIPER CHEROKEE ARCHER przez SPITFIRE i MUSTANG do wywiadowczego SR-71 i kilku samolotów doświadczalnych) niż na strzelanie. Program ten pojawił się ostatnio w kraju i gwarantuje niezłą zabawę amatorom i fanom lotnictwa.

Firma **THOSE DESIGNERS** opracowała kilka nowych dyskietek zawierających ilustracje i obrazki

do programu GEOS. Obrazki te mogą być wykorzystywane zarówno z wersją GEOS dla C-64 jak i programem GEOS 128. Format zapisu jest zgodny z samym GEOS'em dzięki czemu użytkownik może wczytać potrzebny mu obrazek bezpośrednio z programów GEOPAINT czy GEOWRITE. Produkt został opatrzony nazwą DISKART 7 oraz MUSI-KIT.

64-TRAN będzie takimym kąskiem dla wszystkich programujących w języku FORTRAN. Program ten umożliwi generowanie kodu źródłowego za pomocą dowolnego edytora tekstu zapisującego dane w zbiorach sekwencyjnych, oraz posługuje się standardowymi poleceniami i dyrektywami stosowanymi w FORTRANIE (z rozszerzeniami). Autorem opracowania jest firma TRIDENT SOFTWARE.

W chwili gdy numer ten dotrze do rąk Czytelników na rynku amerykańskim znajdzie się zapewne ciąg dalszy bardzo popularnej swego czasu gry — IMPOSSIBLE MISSION. IMPOSSIBLE MISSION II gwarantuje znacznie lepszą grafikę, a sam temat jest bardzo podobny do oryginału — tym razem na-

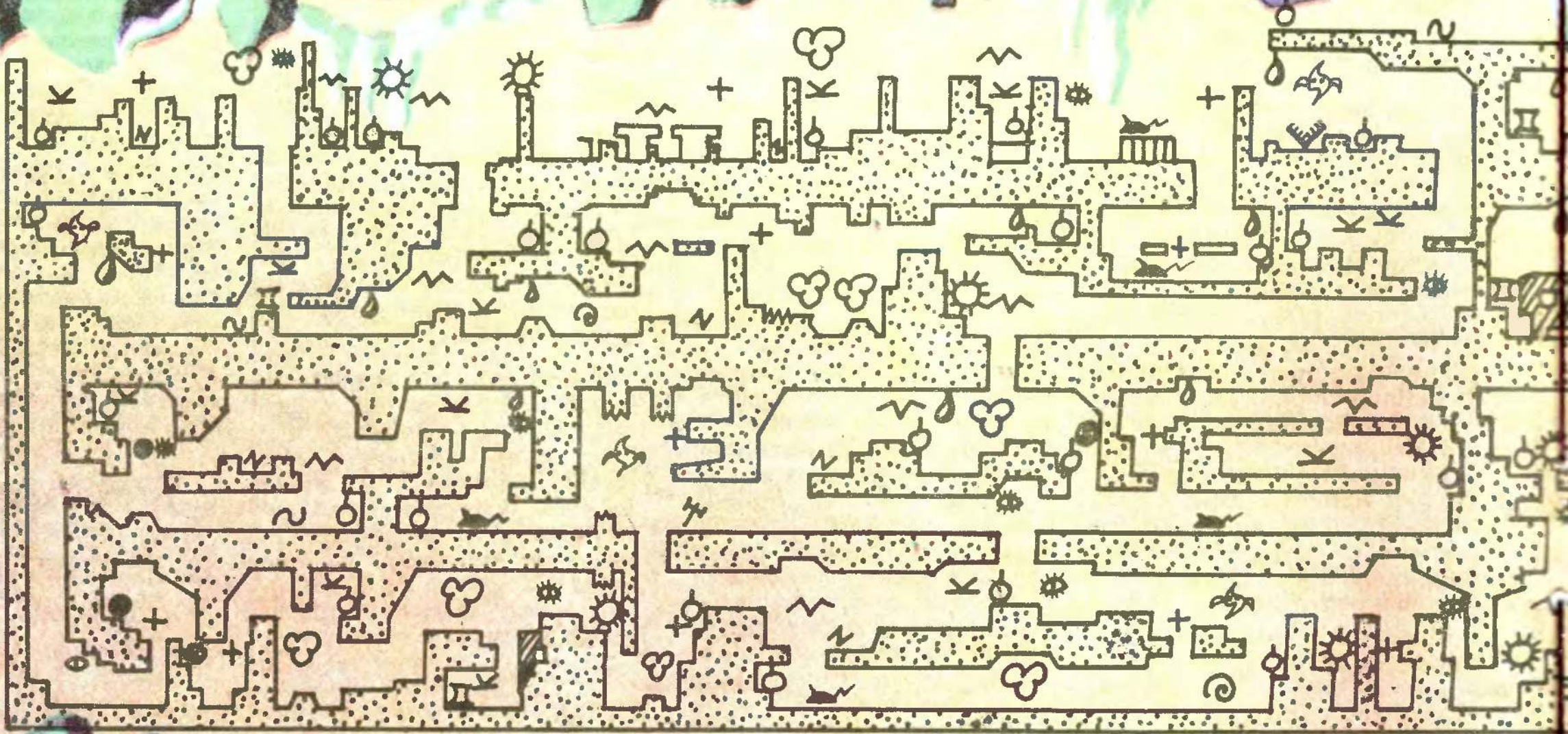
leży odnaleźć fragmenty kodu znajdujące się w wielu niedostępnych (z pozoru) miejscach.

Po raz pierwszy od wielu lat na tegorocznych Targach CES odbywających się co pół roku w USA Commodore 64 cieszył się znacznie mniejszym zainteresowaniem niż dotychczas. Najwięcej uwagi poświęcono tym razem Amidze, która według opinii ekspertów ma duże szanse na powtórzenie sukcesu rynkowego „staruszka” C-64. Pomimo tego na tych samych targach wiele firm przedstawiło sporo nowego oprogramowania dla niekorowanego króla komputerów 8-bitowych. W większości wypadków są to gry, choć nie brakuje także programów użytkowych zarówno dla C-64 jak i C-128.

Lista dostępnego oprogramowania do Amigi nadal rośnie i poszerza się o coraz to nowe programy. Ostatnio w Toronto zademonstrowano programy Perfect Page (program typu Desktop Publishing) oraz AIRWARE — program przeznaczony dla linii lotniczych.

Klaudiusz Dybowski

SPIKY HAROLD



SPIKY HAROLD to typowa przygodowo-zręcznościowa gra dla każdego. Cel jest prosty — należy zebrać wszystkie porozmieszczone w labiryncie przedmioty. Postać języka Harolda jest duża, co sprawia grającemu wiele kłopotów, gdyż

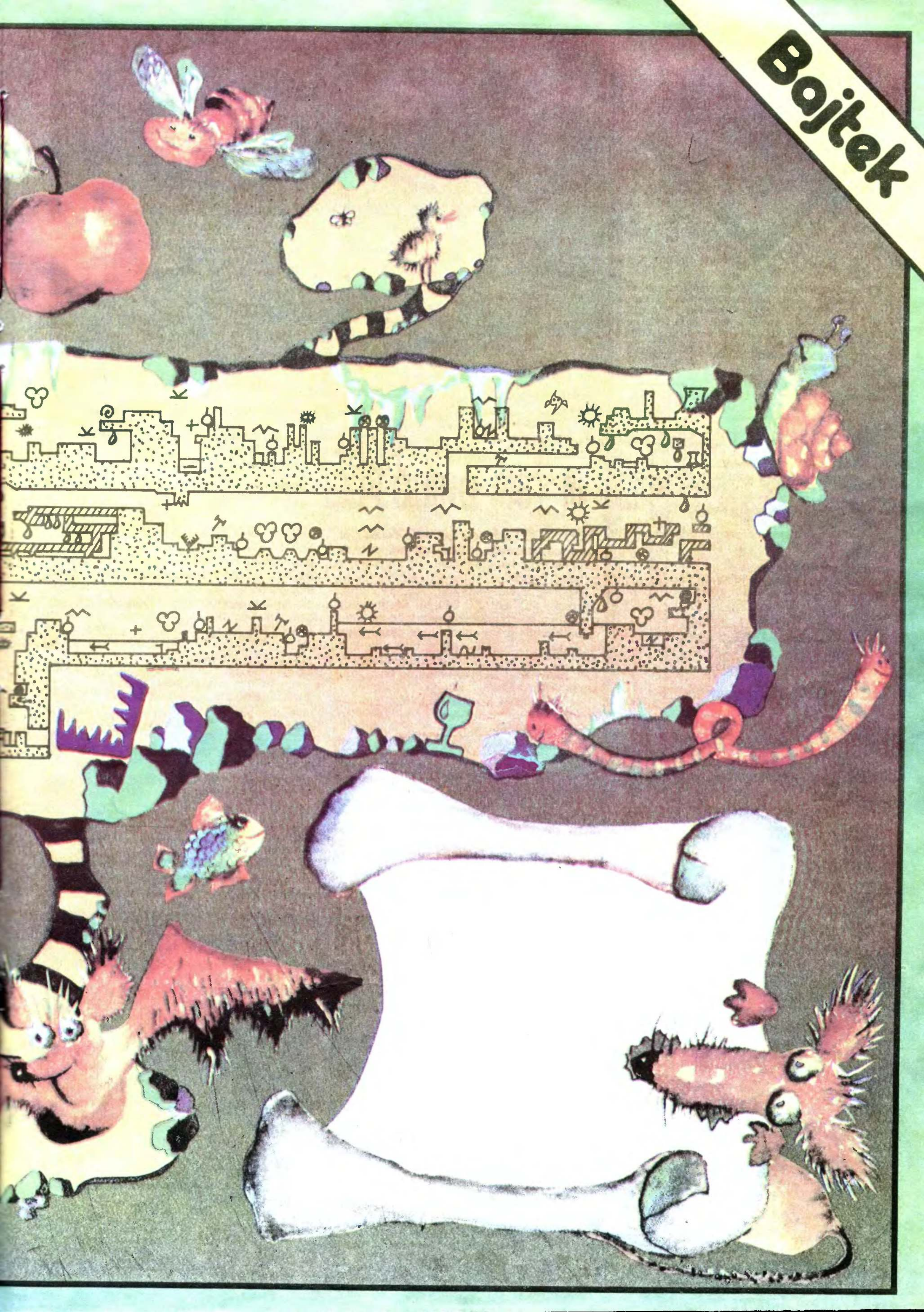
wszystko, co się rusza, jest niebezpieczne. Mimo, że na początku język dysponuje trzydziestoma „życiami”, to przejście całego labiryntu należy uznać za sukces.

Gra posiada pięćdziesiąt siedem obrazów, grafika i animac-

ja stoją to na najwyższym poziomie. Po zebraniu wszystkich przedmiotów należy udać się tam, gdzie wskazuje strzałka, w ostatniej komnacie czeka nagroda w formie...
sprawdź sam!

Tomasz Pietrzykowski





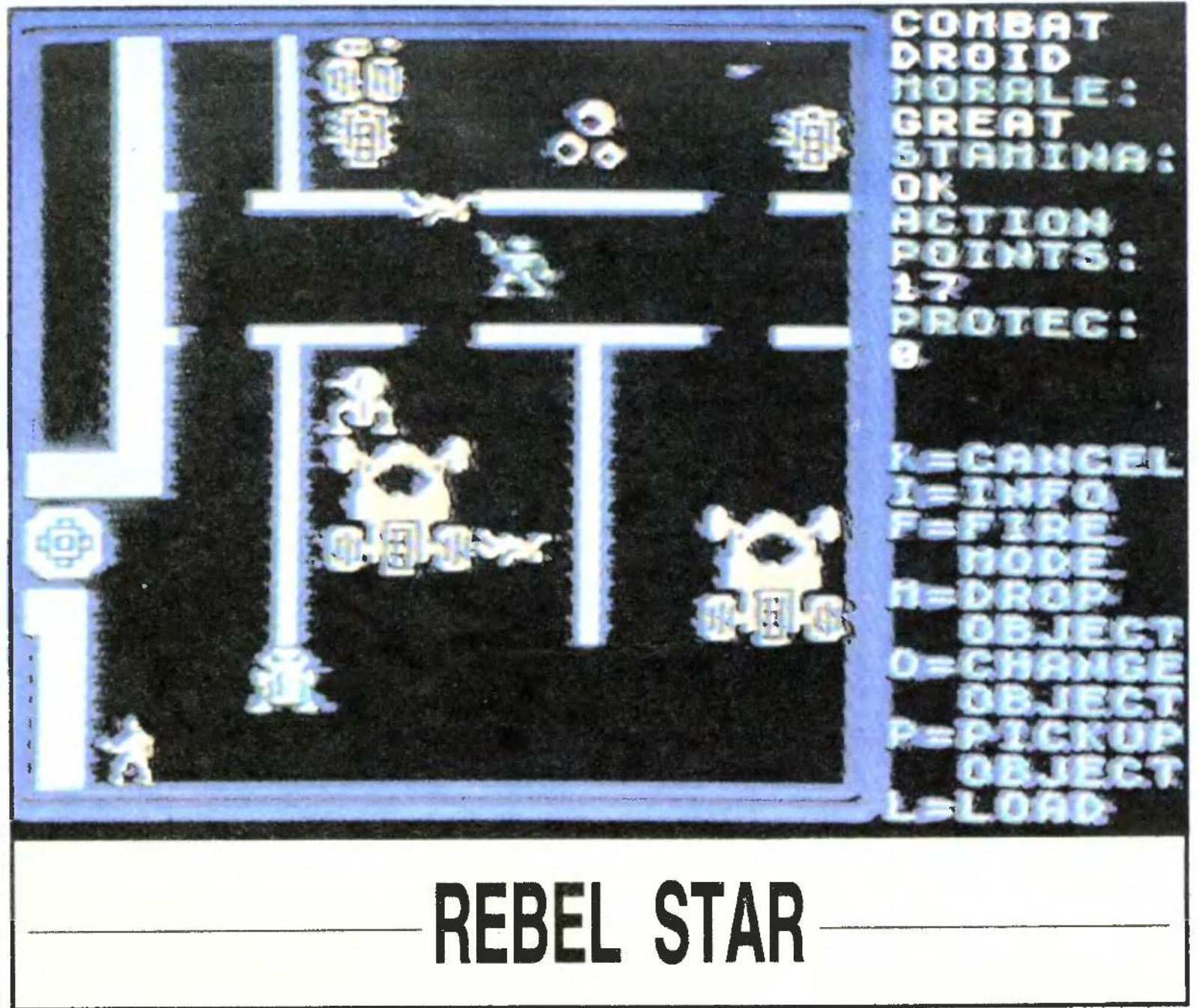
10

BAJTKOWA LISTA PRZEBOJÓW (11/88)

Symulacyjna strzelanina SKYFOX podoba się najbardziej. Zaraz po niej gra, która wymaga myślenia i skójarzeń — DETECTIVE. EXPLODING FIST III powoli spada, lecz wszystko wskazuje na to, że utrzyma się jeszcze długo. Z nowości zauważyć można arcytrudny CHIPWAR, którego mapę zamieścimy już wkrótce. Na listopadowe notowanie otrzymaliśmy 2451 propozycji na 157 tytułów gier.

	ATARI	AMSTRAD	COMMODORE	SPECTRUM
1 SKYFOX	↑	x	x	x
2 DETECTIVE	↑		x	
3 EXPLODING FIST III	↓	x	x	
4 SECRET DIARY	↓	x	x	x
5 URBAN UPSTART	↓			x
6 CHIPWAR	!		x	
7 WEST BANK	!	x	x	x
8 RENEGADE	↓	x	x	x
9 STRIKE FORCE COBRA	↓	x	x	x
10 NOSFERATU	!	x	x	

Nagrody otrzymują: Carmen Tarcha i Karol Szymanowski



REBEL STAR

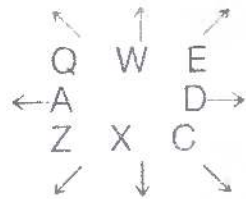
REBEL STAR jest jedną z niewielu gier, które nie posiadają scenariusza ani legendy. Założenie gry bardzo proste, ale i wydane w ciekawej formie.

Akcja rozgrywa się na Księżycu, w kosmicznej bazie MOONBASE DELTA. Stacjonuje w niej grupa robotów i cyborgów różnego typu, dowodzonych przez centralny komputer bazy — ISAAC COMPUTER. Wokół bazy zgromadzeni są rebelianci (w większości ludzie), których zadaniem jest wdarcie się do bazy i zniszczenie ISAAC'a.

Na oryginalnej kasecie, gra REBEL STAR nagrana jest w dwóch wersjach — jednoosobowej, gdzie przeciwnikiem jest komputer i dwuosobowej, gdzie gracze grają przeciwko sobie.

Każda z walczących stron ma na początku taką samą ilość postaci. Wszystkie one mają swoją indywidualną charakterystykę — składają się na nią punkty akcyjne, odporność, morale itp. Postaciami sterujemy z klawiatury — do wyboru są trzy opcje:

— CURSOR MODE



- I — informacja o postaci
- O — koniec tury
- S — selekcja postaci
- J — wypośrodkowanie ekranu
- N — zmiana postaci

Po wciśnięciu S przechodzimy do SELECT MODE:

- M — wyrzucenie przedmiotu
- P — podniesienie przedmiotu
- O — wybór przedmiotu
- L — ładowanie amunicji
- F — przejście do opcji strzelania

W opcji strzelanina (FIRE MODE) zmienia się nieco mapa terenu — obraz jest uproszczony, co umożliwia strzelanie i trafianie. Do

wykorzystania są dwa rodzaje strzału: AIMED SHOT i SNAP SHOT. Różnią się one ilością potrzebnych punktów akcyjnych i skutecznością. W grze jest wiele rodzajów broni, każda z nich ma określoną ilość amunicji, którą uzupełnić można przy pomocy odpowiedniego magazynka:

- LASER GUN — potrzebuje LASER PACK 1
- LASER PISTOL — potrzebuje LASER PACK 2
- PHOTON — potrzebuje LASER PACK 3
- PISTOL — potrzebuje PISTOL CLIP
- AUTO RIFLE — potrzebuje RIFLE CLIP.

Każdy trafny strzał powoduje obniżenie poziomu odporności trafionej postaci. Jeżeli poziom ten spadnie do zera, postać nie będzie zdolna do walki i zginie. W dalszej grze widoczna będzie jako szczątki lub ciało, oznaczone kolorem białym.

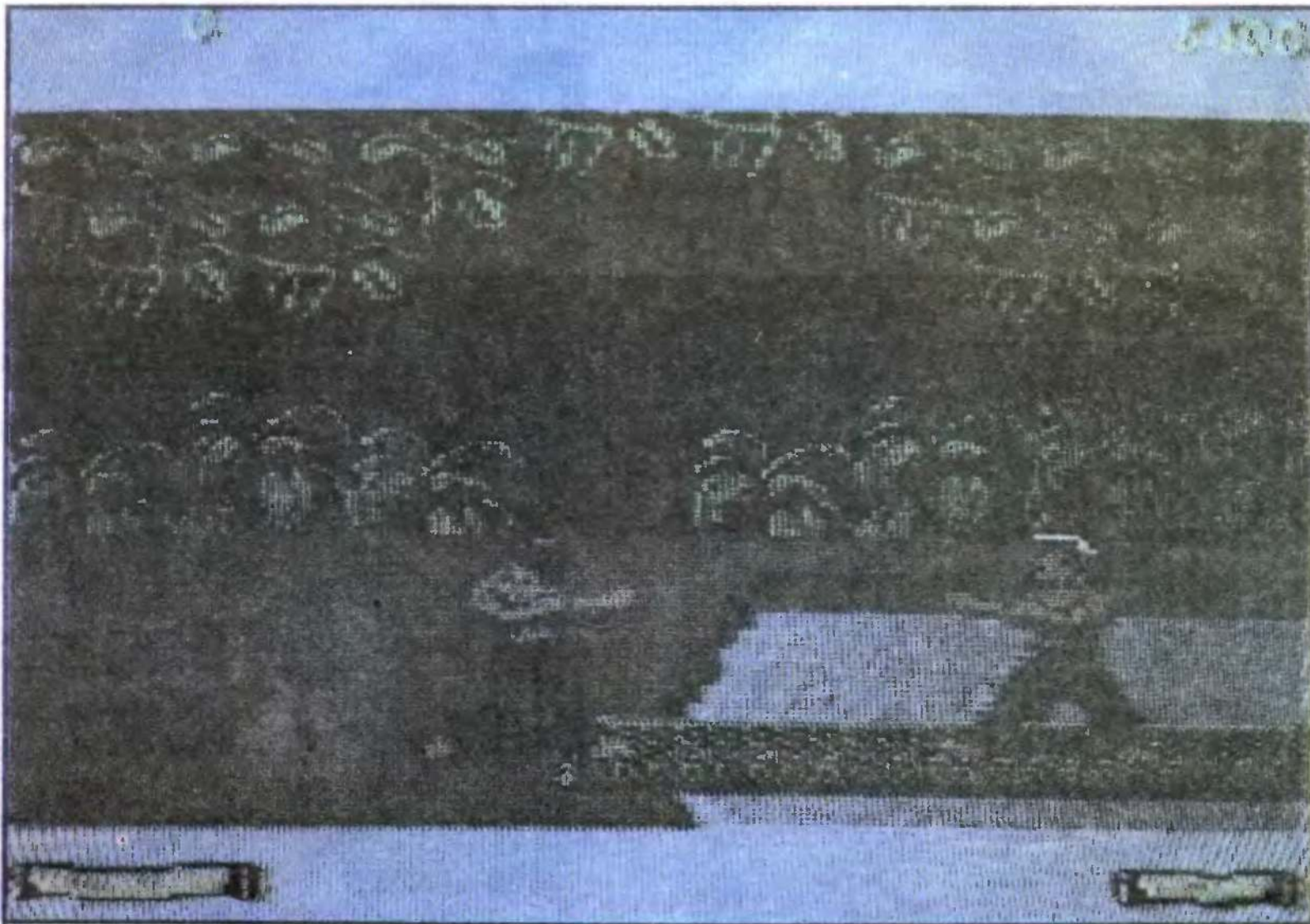
Duże znaczenie ma używanie odpowiednich przedmiotów, co jest jednocześnie bardzo proste. Wystarczy stanąć na wybranym przedmiocie, nacisnąć klawisz podnoszenia i już można go używać.

Oprócz karabinów, pistoletów, mieczy świetlnych występują przedmioty: MEDI PROBE — używany do leczenia ludzi, DRIOD PROBE — do reperacji robotów, DRIOD KEY — do uruchamiania i wyłączenia robotów, KEY — otwiera zamknięte drzwi.

Na terenie bazy znajduje się ogródek, skąd można zabrać kosiarkę do trawy i odpowiednio potraktować nią przeciwnika. W grze jednoosobowej do wyboru jest stopień inteligencji komputera od 1 do 8. Każda postać może jednocześnie trzymać 4 przedmioty. Możliwe jest ustawienie automatycznego strzału — klawisz U. Główne wejścia do bazy mogą być zniszczone jedynie przez roboty rebeliantów. Na zakończenie dodam, że przeciętna rozgrywka trwa 3-4 godziny.

Firma: Firebird
Komputer: ZX Spectrum 48/+

(jk)



THE WAY OF THE EXPLODING FIST III

Nie tak dawno fascynowaliśmy się wspaniałą grą firmy Melbourne Housa, zatytułowaną „The Way of the Exploding Fist” — Droga Eksplodującej Pięści. Po roku pojawiła się jej druga część. Różnica polegała w istocie na scenerii, w której odbywały się pojedynki.

Było jednak wprowadzone niewielkie usprawnienie — komputer „uczył” się walczyć od gracza, swego przeciwnika. Na początku bowiem umiał wykonywać jedynie podstawowe ruchy, lecz w miarę gry nabierał wprawy i starał się pokonać przeciwnika, analizując jego taktykę.

Minął rok już mamy część trzecią, dużo bardziej rozbudowaną. Przede wszystkim tytułowy karateka znajduje się w labiryncie! Jego zadaniem jest skompletowanie ośmiu części świętego papirusu mnichów Shao-Lin. W labiryncie znajdują się też agresywni osobnicy, którzy sami zaczynają ze spokojnym karateką. Trzeba im więc wytłumaczyć, że nie przystoi przeszkadzać w tak ważnej i trudnej misji. Podstawowym argumentem będzie oczywiście pięść, od której wiele w tej grze zależy.

Labirynt po którym porusza się bohater jest

wielopiętrową płataniną japońskich domów, ogrodów, dróg, jaskiń i wodospadów. Znajduje się tam także osiem świątyń, w których należy odczytać modlitwy ze znalezionych kawałków papirusu. I tu pewna przeszkoda.

Każda część jest inna i trzeba uważać, by czytać odpowiednią modlitwę w odpowiedniej świątyni. Pomyłka oznacza stratę wszelkich szans na powodzenie i konieczność rozpoczęcia gry od początku.

Siła życiowa, czy też energia karateki pokazana jest w lewym dolnym rogu ekranu w postaci częściowo rozwiniętego zwoju. Po spotkaniu ze strażnikiem, w prawym dolnym rogu ekranu pojawia się zwój symbolizujący jego energię. I teraz — kto pierwszy! Walka toczy się do momentu, gdy zwój jednego z walczących zwinie się do końca...

Straconą energię uzupełnić można, odwiedzając świątynie. Jednak w każdej z nich jest to możliwe tylko raz. Karatece wypada życzyć szczęścia, a my... czekamy na czwartą część!

Komputer: Commodore 64/128

(mp)

KRÓL I KRÓLOWA GIER



Carmen Tarcha

klasa 7b, Szkoła Podstawowa nr 2 w Gdyni
Posiadany mikrokomputer: Commodore 64
Ulubiona gra: Hyper Sport
Hobby: sport, w szczególności tenis, język angielski



Karol Szymanowski

klasa 5, Szkoła Podstawowa nr 272 w Warszawie
Posiadany mikrokomputer: ZX Spectrum
Ulubiona gra: Renegade
Hobby: ping-pong i język angielski.

S.O.S.

Poszukuje dokładnego opisu gry „Universal Hero” na ATARI 800 XL.

Florian Brudny
ul. Tuwima 3, 43-252 Golasowice

Jestem w kłopotcie, gdyż nie znam kodu uruchamiającego do gry „RAMBO” w wersji na ATARI XL. Przepisałem już prawie połowę słowniczka angielsko-polskiego, bez rezultatu. Zrewanżuję się opisami gier Archon, Ninja, Zorro, Ghostbusters i Drop Zone.

Marcin Słowiak
ul. Sportowa 6, 34-220 Maków Podh.

Bardzo proszę o pomoc w grach Alchemist i The Goo-nies na ZX Spectrum.

Andrzej Tarski
ul. Modzelewskiego 46/50 m 606, Warszawa

Mam ATARI 800 XL i nie wiem, co robić po zebraniu wszystkich przedmiotów w grze Pharaohs Curse oraz jak ukończyć Ninja Master. Proszę o pomoc.

Jakub Beczek
ul. 150-lecia 49 m 21, 96-300 Żyrardów

Już od wielu miesięcy poszukuję gry Barbarian na ATARI 800 XL. Może ktoś z Czytelników może mi pomóc.

Marek Stabyn
ul. Pstrowskiego 18/18, 37-700 Przemyśl

Poszukuję opisów do następujących gier na ZX Spectrum: Duet, Flight Simulation, Hunter Killer, Space Harrier, F-15 Strike Eagle, Miami Vice, Top Gun. W zamian oferuję m.in. opis do Silent Service.

Roman Tworek
ul. Kościelna 31/2, 47-220 Kędzierzyn

Poszukuję kodu do gry Kennedy Approach oraz sposobu rozpoczęcia gry Mouse Trap. Mam Atari 800 XL.

Mariusz Krzysztofik
os. Piastów 38/7, 31-624 Kraków-N.Huta

Mam komputer ZX Spectrum+. Bardzo proszę o opisy do gier Lunar Jetman, Booty i Tempest. Poszukuję też gier Yabba Babba Dooo, Trap Door, Pyjamarama i Jak the Nipper.

Tomasz Nitkiewicz
ul. Okólna 111 m 23, 42-200 Częstochowa

Mam pewien problem, gdyż od prawie roku próbuję sprzedać oryginalne kasety z programami Willow Pattern i Chimera na Amstrada oraz One on One na Commodore. Może ktoś z Was jest nimi zainteresowany?

Marcin Tańczak
ul. Kanałowa 15/17, 60-710 Poznań

Jesteśmy miłośnikami gier tekstowych. Proponujemy wymianę doświadczeń o grach Silent Service, Seven Cities of Gold, Hacker i Ghostbusters na ZX Spectrum i C-64. Prosimy też o pomoc w grach Hobbit i Lord of the Rings.

Przemysław i Bartłomiej Bukowscy
ul. 1 Maja 11, 83-110 Tczew

Pilnie szukam gier Saboteur, Turbo Esprit, Ace, Starquake, Chimera wraz z opisami. W zamian oferuję Exolon, Spirits, Zynaps, Army Moves, Scooby Doo i Basket Master. Mam komputer Spectrum 48.

Mariusz Czykier
ul. Kopernika 43, 34-330 Żywiec

Pomocy! Mam opis do gry Ghostbusters w wersji na ATARI. Czytałem, że można powiększyć kwotę do 40 000 dolarów, ale ja zawsze dostaję 10 000 i chciałbym chociaż 20 000 dolarów. Nie wiem też, jak wziąć klucz do świątyni. Będę bardzo wdzięczny za pomoc.

Michał Kowalik
os. XX-lecia PRL 4 m 63, 27-100 Iłża

Potrzebuję pomocy w grach Theatre Europe (kod do użycia bomby atomowej), Super Huey i Hobbit na ATARI.

Krzysztof Biernat
ul. Wiejska 3, 58-230 Niemcza

Mam wielkie kłopoty z kierowaniem samochodem w grze Pole Position. Może ktoś jest w stanie mi pomóc.

Bartek Danek
al. 1000-lecia 47/113, 33-380 Krynica

Liczę na pomoc w rozszyfrowaniu gier Nosferatu, Rasputin, Flash Gordon i Enigma Force na Commodore 64. W zamian służę opisami m.in. Colossus Chess.

Marek Wojakowski
ul. 1 Maja 52/79, 41-300 Dąbrowa Górnicza

P PRZEKAZYWANIE P PARAMETRÓW

Przedstawiony przez nas dotychczas program rozbudowy interpretera pozwala używać procedur bez parametrów. Wywołanie procedury bez parametrów polega na załadowaniu jej i wykonaniu instrukcji zawartych w treści tej procedury. Jeśli programowany przez nas algorytm wymaga wielokrotnego powtarzania pewnych operacji w różnych miejscach programu, to ich każdorazowe wypisywanie możemy zastąpić wywołaniem procedury.

Na przykład, algorytm zmiany koloru ramki na niebieski możemy zapisać za pomocą następujących instrukcji:

```
10 fh"ramka1"
20 poke 13*4096+2*16+1.6
30 fe
```

Przypominamy, że przed wykonaniem procedury "ramka1" należy zapisać ją w pamięci zewnętrznej w zbiorze o tej samej co procedura nazwie. Wywołać procedurę "ramka1" z dyskietki można za pomocą instrukcji fc:

```
fc"ramka1",8
```

Procedury bez parametrów mogą służyć jedynie do wykonywania stałe tych samych operacji. Na przykład, chcąc wykonać algorytm zmiany koloru ramki na czerwony musielibyśmy napisać nową procedurę "ramka2":

```
10 fh"ramka2"
20 poke 13*4096+2*16+1.2
30 fe
```

Aby otrzymać procedurę bardziej ogólną należy użyć procedury z parametrami. Napiszemy zatem taką wersję procedury "ramka", aby kolor ramki był jej parametrem.

```
10 fh"ramka",kolor
20 poke 13*4096+2*16+1,kolor
30 fe
```

Nazwa (zmienna) kolor służy tutaj do zapisania ogólnej postaci algorytmu. Jest to tzw. parametr formalny. Mówiąc intuicyjnie, wykonanie procedury z parametrami polega na wykonaniu treści tej procedury, w której parametry formalne zostały zastąpione odpowiednimi parametrami aktualnymi. Na przykład, wywołanie procedury "ramka" w postaci:

```
fc"ramka",8,x (z dyskietki),
fc"ramka",1,x (z taśmy).
```

spowoduje zmianę koloru ramki na kolor określony wartością wyrażenia x. Wyrażenie x jest to tzw. parametr aktualny i jego wartość podstawiana jest w procedurze zamiast parametru kolor.

W Warsaw BASIC'u wyróżnia się następujące sposoby przesyłania parametrów:

- (1) Po wyliczeniu wartości parametru aktualnego podstawia się ją za wartość odpowiedniego parametru formalnego traktowanego jako zmienna lokalna procedury. Jest to przekazywanie przez wartość i występuje najczęściej.
- (2) Parametr formalny jest zmienną prostą, nazwą tablicy lub nazwą funkcji. Tym razem w czasie wykonywania procedury nazwa parametru jest podstawiana zamiast odpowiedniego parametru formalnego. Jest to przesyłanie przez nazwę i bywa stosowane wtedy, gdy dany parametr reprezentuje wynik działania procedury. Zmiana wartości takiego parametru formalnego w procedurze powoduje modyfikację odpowiadającego mu parametru aktualnego w programie wywołującym.

Przekazywanie przez nazwę w przypadku tablic jest jedynym możliwym sposobem i oznacza dużą oszczędność pamięci, bowiem dane zawarte w tablicy nie są przepisywane w pole zmiennych procedury, lecz każdorazowo, gdy trzeba się do nich odwołać następuje tylko zamiana nazwy parametru formalnego i aktualnego.

Przekazywanie funkcji umożliwia projektowanie bardzo ogólnych procedur przetwarzających wartości obliczone według dowolnego wzoru zadanego procedurze poprzez funkcję, będącą parametrem aktualnym. Na przykład, mając te możliwości możemy stworzyć procedurę, która posortuje dwukolumnową tablicę nie tylko według wartości elementów każdej z kolumn, ale także według wartości dowolnej funkcji określonej na elementach wierszy, jeśli tylko odpowiedni wzór zdefiniujemy w programie wywołującym i prześlemy do procedury przez nazwę. Jeśli funkcja jest parametrem aktualnym wywołania procedury, to jako wartości zmiennych, które nie są argumentami formalnymi funkcji, a występują we wzorze tej funkcji, przy obliczaniu jej wartości w procedurze przyjmuje się wartości z programu, w którym wołano procedurę.

W Warsaw BASIC'u przez wartość można przesyłać dane numeryczne i tekstowe. Parametrami przekazywanymi przez nazwę mogą być zmienne proste typu rzeczywistego, całkowitego i łańcuchowego, tablice wszystkich typów i funkcje zdefiniowane przez DEF FN. W prezentowanych obok programach 1 i 2 ograniczamy się do przekazywania przez wartość danych numerycznych.

Ponadto program 1 zawiera uzupełnienie algorytmu wywołania o sprawdzenie tego, czy procedura jest już w pamięci komputera, przed załadowaniem jej z dyskietki lub taśmy. Ponowne załadowanie procedury będzie konieczne tylko wtedy, jeśli wywołaną procedurę zastąpiła w pamięci inna lub jeśli objętość poziomu wywołującego zmieni się od ostatnio wykonanego przełączenia. Zmniejszenie lub zwiększenie objętości poziomu następuje na skutek zmian w programie na poziomie wywołującym lub na skutek utworzenia nowych zmiennych na tym poziomie.

Parametrem aktualnym w interpreterze rozbudowanym dla Czytelników Bajtka może być dowolne wyrażenie arytmetyczne. Należy tylko pamiętać, aby przez użyciem jakiejś zmiennej w tym wyrażeniu zadeklarować ją przed wywołaniem procedury. Z braku miejsca opuszciliśmy tu odpowiednią procedurę wykrywającą ten błąd w Warsaw BASIC'u. Parametrem formalnym w interpreterze rozbudowanym dla Czytelników Bajtka musi być zmienna prosta typu rzeczywistego.

Ograniczenie się do przekazywania przez wartość sprawia, że komunikacja z procedurą jest możliwa tylko w jedną stronę. W Warsaw BASIC'u wyniki działania procedury można przesyłać do programu wywołującego korzystając z przekazywania przez nazwę. Jak sobie z tym poradzić inaczej opowiemy w następnym odcinku.

Krzysztof Gajewski
Bogusław Radziszewski

```

Program 1
.. C7CC 4C D8 C7 JMP $C7D8 .. C82C 90 01 BCC $C82F
.. C7CF 4C 17 C8 JMP $C817 .. C82E C8 INY
.. C7D2 4C B6 C5 JMP $C5B6 .. C82F 85 35 STA $35
.. C7D5 4C 55 C8 JMP $C855 .. C831 84 36 STY $36
.. C7D8 A0 00 LDY #$00 .. C833 18 CLC
.. C7DA B1 31 LDA ($31),Y .. C834 69 05 ADC #$05
.. C7DC D0 33 BNE $C811 .. C836 90 01 BCC $C839
.. C7DE A0 02 LDY #$02 .. C838 C8 INY
.. C7E0 B1 31 LDA ($31),Y .. C839 85 2F STA $2F
.. C7E2 F0 2F BEQ $C813 .. C83B 84 30 STY $30
.. C7E4 A0 05 LDY #$05 .. C83D 20 A5 C6 JSR $C6A5
.. C7E6 B1 31 LDA ($31),Y .. C840 A6 35 LDX $35
.. C7E8 C9 5C CMP #$5C .. C842 A4 36 LDY $36
.. C7EA D0 25 BNE $C811 .. C844 20 D7 BB JSR $BBD7
.. C7EC C8 INY .. C847 20 C2 C5 JSR $C5C2
.. C7ED B1 31 LDA ($31),Y .. C84A E6 93 INC $93
.. C7EF C9 48 CMP #$48 .. C84C A0 00 LDY #$00
.. C7F1 D0 1E BNE $C811 .. C84E B1 7A LDA ($7A),Y
.. C7F3 A2 00 LDX #$00 .. C850 C9 2C CMP #$2C
.. C7F5 C8 INY .. C852 F0 C8 BEQ $C81C
.. C7F6 C8 INY .. C854 60 RTS
.. C7F7 98 TYA .. C855 A5 93 LDA $93
.. C7F8 48 PHA .. C857 F0 41 BEQ $C89A
.. C7F9 B1 31 LDA ($31),Y .. C859 A5 2D LDA $2D
.. C7FB 48 PHA .. C85B A4 2E LDY $2E
.. C7FC 8A TXA .. C85D 85 35 STA $35
.. C7FD A8 TAY .. C85F 84 36 STY $36
.. C7FE 68 PLA .. C861 20 FD AE JSR $AEFD
.. C7FF D1 BB CMP ($BB),Y .. C864 A0 00 LDY #$00
.. C801 D0 0F BNE $C812 .. C866 A2 00 LDX #$00
.. C803 68 PLA .. C868 20 79 00 JSR $0079
.. C804 A8 TAY .. C86B 91 35 STA ($35),Y
.. C805 E8 INX .. C86D C8 INY
.. C806 E4 B7 CPX $B7 .. C86E 20 79 00 JSR $0079
.. C808 D0 EC BNE $C7F6 .. C871 20 13 B1 JSR $B113
.. C80A C8 INY .. C874 20 73 00 JSR $0073
.. C80B B1 31 LDA ($31),Y .. C877 90 05 BCC $C87E
.. C80D C9 22 CMP #$22 .. C879 20 13 B1 JSR $B113
.. C80F F0 00 BEQ $C811 .. C87C 90 07 BCC $C885
.. C811 60 RTS .. C87E AA TAX
.. C812 68 PLA .. C87F 20 73 00 JSR $0073
.. C813 A9 01 LDA #$01 .. C882 20 13 B1 JSR $B113
.. C815 D0 FA BNE $C811 .. C885 8A TXA
.. C817 A9 00 LDA #$00 .. C886 91 35 STA ($35),Y
.. C819 85 93 STA $93 .. C888 C6 93 DEC $93
.. C81B A8 TAY .. C88A F0 0D BEQ $C899
.. C81C 20 FD AE JSR $AEFD .. C88C A5 35 LDA $35
.. C81F 20 8A AD JSR $AD8A .. C88E A4 36 LDY $36
.. C822 20 80 C5 JSR $C580 .. C890 18 CLC
.. C825 A5 2F LDA $2F .. C891 69 07 ADC #$07
.. C827 A4 30 LDY $30 .. C893 90 01 BCC $C896
.. C829 18 CLC .. C895 C8 INY
.. C82A 69 02 ADC #$02 .. C896 18 CLC
.. C82C 90 01 BCC $C82F
.. C82E C8 INY
.. C82F 85 35 STA $35
.. C831 84 36 STY $36
.. C833 18 CLC
.. C834 69 05 ADC #$05
.. C836 90 01 BCC $C839
.. C838 C8 INY
.. C839 85 2F STA $2F
.. C83B 84 30 STY $30
.. C83D 20 A5 C6 JSR $C6A5
.. C840 A6 35 LDX $35
.. C842 A4 36 LDY $36
.. C844 20 D7 BB JSR $BBD7
.. C847 20 C2 C5 JSR $C5C2
.. C84A E6 93 INC $93
.. C84C A0 00 LDY #$00
.. C84E B1 7A LDA ($7A),Y
.. C850 C9 2C CMP #$2C
.. C852 F0 C8 BEQ $C81C
.. C854 60 RTS
.. C855 A5 93 LDA $93
.. C857 F0 41 BEQ $C89A
.. C859 A5 2D LDA $2D
.. C85B A4 2E LDY $2E
.. C85D 85 35 STA $35
.. C85F 84 36 STY $36
.. C861 20 FD AE JSR $AEFD
.. C864 A0 00 LDY #$00
.. C866 A2 00 LDX #$00
.. C868 20 79 00 JSR $0079
.. C86B 91 35 STA ($35),Y
.. C86D C8 INY
.. C86E 20 79 00 JSR $0079
.. C871 20 13 B1 JSR $B113
.. C874 20 73 00 JSR $0073
.. C877 90 05 BCC $C87E
.. C879 20 13 B1 JSR $B113
.. C87C 90 07 BCC $C885
.. C87E AA TAX
.. C87F 20 73 00 JSR $0073
.. C882 20 13 B1 JSR $B113
.. C885 8A TXA
.. C886 91 35 STA ($35),Y
.. C888 C6 93 DEC $93
.. C88A F0 0D BEQ $C899
.. C88C A5 35 LDA $35
.. C88E A4 36 LDY $36
.. C890 18 CLC
.. C891 69 07 ADC #$07
.. C893 90 01 BCC $C896
.. C895 C8 INY
.. C896 18 CLC
.. C897 90 C4 BCC $C85D
.. C899 60 RTS
.. C89A 4C 08 AF JMP $AF08
    
```

```

Program 2
700 PRINT"Cześć 7"
702 X=51148: N=208: C=0
704 FOR I=0 TO N: READA: POKEX+I,A: C=C+A: NEXT
706 IF C<>24098 THEN PRINT"Błąd w części 7": END
708 PRINT"Cześć 7 OK"
710 DATA 76,216,199,76,23,200,76,182,197,76,85,200
712 DATA 160,0,177,49,208,51,160,2,177,49,240,47
714 DATA 160,5,177,49,201,92,208,37,200,177,49,201
716 DATA 72,208,30,162,0,200,200,152,72,177,49,72
718 DATA 138,168,104,209,187,208,15,104,168,232,228,183
720 DATA 208,236,200,177,49,201,34,240,0,96,104,169
722 DATA 1,208,250,169,0,133,147,168,32,253,174,32
724 DATA 138,173,32,128,197,165,47,164,48,24,105,2
726 DATA 144,1,200,133,53,132,54,24,105,5,144,1
728 DATA 200,133,47,132,48,32,165,198,166,53,164,54
730 DATA 32,215,187,32,194,197,230,147,160,0,177,122
732 DATA 201,44,240,200,96,165,147,240,65,165,45,164
734 DATA 46,133,53,132,54,32,253,174,160,0,162,0
736 DATA 32,121,0,145,53,200,32,121,0,32,19,177
738 DATA 32,115,0,144,5,32,19,177,144,7,170,32
740 DATA 115,0,32,19,177,138,145,53,198,147,240,13
742 DATA 165,53,164,54,24,105,7,144,1,200,24,144
744 DATA 196,96,76,8,175
    
```


CZYTANIE DYSKIETEK Z PCW NA CPC

CP/M Plus jest systemem operacyjnym wspólnym dla wszystkich 8-bitowych komputerów firmy Amstrad. Niestety format dyskietek zastosowany w serii PCW nie daje się odczytać na starszych modelach serii CPC. Wygodnym rozwiązaniem tego problemu jest, przedstawiony niżej, krótki program w assemblerze.

W normalnej konfiguracji, Amstrad CPC 6128 wyposażony jest w jeden napęd 3-calowych dysków.

W systemie operacyjnym widziany jest on jako logiczna stacja A: lub B:. Każda z tych stacji ma własną tablicę parametrów dyskowych. Po załadowaniu systemu operacyjnego CP/M Plus obie tablice są te same. Jeśli zmienimy opis stacji B:, wstawiając parametry dysku z komputera PCW 8256, to będziemy mogli czytać te dyski na Amstradzie CPC.

Na rys. 1 znajduje się wydruk zbioru ONPCW.MAC. Wykonanie operacji: A>m80 =ONPCW <CR> (asemblacja) A>link ONPCW <CR> (konsolidacja) pozwala otrzymać program ONPCW.COM, który modyfikuje w opisany poprzednio sposób wirtualny napęd B:

Rys. 2 przedstawia zbiór OFFPCW.MAC pozwalający na uzyskanie, po asemblacji i konsolidacji, programu likwidującego skutki programu ONPCW.

Jeśli chcemy korzystać z dyskietek z PCW w napędzie B:, to piszemy: A>ONPCW<CR>, a po zakończeniu pracy z dyskietkami z PCW: A>OFFPCW<CR> i obie stacje mają znów ten sam format.

Nie bez znaczenia jest fakt, że podobną metodę zmiany tablicy parametrów dyskowych możemy zastosować do podłączenia dwustronnego napędu 5¼ 360 KB. Dotyczy to zarówno PCW jak i CPC.

Jonasz Mayer

```

*****
Nazwa programu: OffPCW
Cel: odtworzenie tablicy parametrów
dyskowych stacji B: komputera CPC 6128,
(Likwidacja skutków programu OnPCW).
*****
        z80
bdos    equ 0005
        cseg
; *****      wybór stacji A:
        ld  c,0EH
        ld  e,0
        call bdos
; *****      znajdź adres DPB
        ld  c,1FH
        call bdos
; *****      hl = adres DPB stacji A:
        ld  bc,53
        add hl,bc
        ld  (hl),0
; *****      włączenie autodetekcji formatu
;                dla stacji B:
        ld  c,9
        ld  de,msg
        call bdos
; *****      wypisz tekst na ekranie
        ret
msg:    db  'Program czytający dyski PCW '
        db  'usunięty z pamięci',13,10,'$'
        end
    
```

```

*****
Nazwa programu:      OnPCW
Cel:                umożliwienie odczytu i zapisu dysków
                    ze stacji pojedynczej gęstości komputera PCW
                    8256/8512 na komputerze CPC 6128.
*****
        z80
bdos    equ 0005
        cseg
; *****      wybór stacji A:
        ld  c,0EH
        ld  e,0
        call bdos
; *****      znajdź adres DPB
        ld  c,1FH
        call bdos
; *****      hl = adres DPB stacji A:
        ld  bc,27
        add hl,bc
; *****      hl = adres DPB stacji B:
        ld  de,ndpb
        ld  bc,27
        ex  de,hl
        ldir
; *****      nowe DPB dla stacji B:
                    wpisane w miejsce starego
        ld  c,9
        ld  de,msg
        call bdos
; *****      wypisz tekst na ekranie
        ret
*****
                    TABLICA PARAMETROW DYSKOWYCH DLA PCW 8256
*****
ndpb:   db  36,0      36 rekordów na ścieżkę (Records per track)
        db  3        3 (ang. Block Shift)
        db  7        7 (ang. Block Mask)
        db  0        0 maska rozszerzenia (Extend Mask)
        db  174,0    175 bloków danych (Data Blocks)
        db  63,0     64 miejsca w katalogu (Directory Entries)
        db  192,0    2 bloki na katalog (Directory Blocks)
        db  16,0     16 bitowy wektor sumy kontrolnej (Checksum Vector)
        db  1,0      1 ścieżka zarezerwowana na system (Reserved Track)
        db  2        2 (ang. Physical Sector Shift)
        db  3        3 (ang. Physical Sector Mask)
        db  0        jednostronny zapis (Single Sided)
        db  40       40 ścieżek na stronę (Tracks per side)
        db  9        9 sektorów na ścieżkę (Sectors per track)
        db  !        1 numer pierwszego sektora (First sector number)
        db  0,2      512 bajtów na sektor (Bytes per sector)
        db  42       długość przerwy (czyt/zapis) (Gap length - r/w)
        db  82       długość przerwy (format.) (Gap length - format)
        db  60H      rodzaj zapisu MFM
endpb:  db  255      wyłącz autodetekcję formatu (auto-select format)
*****
msg:    db  'Program czytający dyski PCW '
        db  'zainstalowany dla stacji B',13,10,'$'
        end
    
```

SYMBOL RSX DLA CPC 6128

W numerze 5/88 przedstawiono komendę SYMBOL dostępną w Turbo-Pascalu na komputerze Amstrad PCW 8256. Przedstawiony poniżej wydruk zbioru CSYMBOL.MAC pozwala na uzyskanie własnych znaków również na starszym modelu Amstrada — CPC 6128. W przypadku tego komputera należy zastąpić opisywany wcześniej zbiór SYMBOL.MAC zbiorem CSYMBOL.MAC. Recepta postępowania jest taka sama jak dla PCW.

Jonasz Mayer

```

; rex for symbol procedure
; in Pascal
; (C) JM December 1987
; version for CPC 6128
        z80
        cseg
; header
        ds  6
        jp  start
next:   db  0c3h,0,0,0ffh,0
        db  'symbol'
        db  0,0,0
start:
        ld  a,c
        cp  79
        jp  nz,next
begin:
        ex  de,hl
        ld  de,buffer
        ld  bc,9
        ldir
        ld  hl,(0001)
        ld  bc,87
        add hl,bc
        ld  (adr),hl
        ld  hl,buffer
        ld  a,(hl)
        inc hl
        call jphl
        db  0a8h,0bbh
        ret
jphl:   db  0c3h
        adr: db  0,0
        buffer:
            ds  9
        end
;
; Listing of file: CSYMBOL.MAC
    
```


PROGRAMOWAĆ MOŻE KAŻDY



KALKULATOR TO TEŻ KOMPJUTER

Już wkrótce Boże Narodzenie wraz z choinką, a pod nią... może nareszcie ten wymarzony komputer. Niektórzy z Was pewnie już dokładnie przemyśleli, do czego będą go używać. Marzą się Wam oprogramowania, biblioteki procedur matematycznych czy np. inżynierskich dołączanych na zawołanie do każdego pisanego akurat programu.

Ale rzeczywistość nie wygląda jednak tak różowo. Po pierwsze dużą część oprogramowania trzeba stworzyć samemu, a takie programy nie są często napisane optymalnie i uniwersalnie, działają długo i są nieraz nieznośnie niedokładne. Po drugie dołączanie programów wymieniających między sobą parametry nie jest takie proste, a często okazuje się całkiem niemożliwe, o czym autor przekonał się już na własnej skórze. Ponadto czasami okazuje się, że dokładność własna komputera (rzędu 10^{-8}) nie jest wystarczająca. Wtedy nadochodzi człowieka refleksja, że może jednak komputer pozostać zarządzaniu bazami danych, wykorzystując go jako edytor tekstu, arkusz kalkulacyjny, a do obliczeń użyć poczyty kulator z wbudowanym oprogramowaniem, programowalny, szybki i dokładny.

Właśnie na Gwiazdkę chciałem przedstawić jako alternatywę dla komputera najnowocześniejszy obecnie kalkulator: HP-28S firmy Hewlett Packard. Nazwa firmy mówi już sama za siebie. Wyroby ze znacznikiem HP cieszą się takim samym uznaniem wśród użytkowników sprzętu elektronicznego jakim cieszą się Rolls-Royce wśród znawców motoryzacji. O uznaniu dla firmy decyduje doskonała jakość produktów (na niektóre wyroby firma daje nawet 25 lat gwarancji), fantastyczne wręcz możliwości obliczeniowe, szybkość i duża dokładność prowadzonych obliczeń. Zdaniem wielu użytkowników kalkulatory tej firmy przewyższają swoimi walorami technicznymi i eksploatacyjnymi kalkulatory SHARP, Casio czy Texas Instruments, choć są jednocześnie od nich znacznie droższe. Kalkulatory serii HP-28 pojawiły się w 1986 roku. Najpierw był to HP-28C z 2 kilobajtami pamięci RAM. Po jego udoskonaleniu, dodaniu nowych funkcji, rozszerzeniu pamięci do 32 KB na rynku pojawił się model HP-28S. Jest on sprzedawany od jesieni zeszłego roku za umiarkowaną cenę w USA (170 dolarów) i za zawrotną sumę na rynku europejskim (np. w Finlandii 380 dolarów ze zniżką studencką, czy 315 dolarów w Austrii). Za tę cenę otrzymujemy kalkulator wraz z dokumentacją, którą stanowią 2 książki po około 350 stron każda.

Pomówmy jednak teraz o możliwościach tego kalkulatora. Po raz pierwszy wśród kalkulatorów zyskujemy możliwość prowadzenia nie tylko obliczeń numerycznych, ale i symbolicznych. Takie możliwości były do tej pory z trudem osiągalne nawet na profesjonalnych komputerach (przykładem jest program MUMATH na IBM). Na czym polegają obliczenia symboliczne? W skrócie chodzi o to, że odpowiedź na dane zagadnienie nie jest liczbą, ale wzór ogólny. I tak HP-28S może symbolicznie rozwiązywać równania, znajdować pierwiastki, może przekształcać wzory (skrótanie i rozszerzanie wyrażenia) np. podając $(x+y)^2$ otrzymamy po rozszerzeniu znany wzór $x^2+2xy+y^2$ i odwrotnie. Oprócz tego z danej postaci równania z kilkoma zmiennymi kalkulator potrafi znaleźć interesującą nas zmienną i przedstawić ją jako funkcję pozostałych. Wśród jeszcze wielu możliwości pozostaje punkt najważniejsz-

szy i chyba najciekawszy. Rachunek różniczkowy i całkowy. Kalkulator potrafi wyciągnąć pochodną z każdego poprawnego pod względem matematycznym wyrażenia. Odwrotnie może też policzyć całkę nieoznaczoną z danej funkcji np. $\int (x^2+0.5x)dx$ da w wyniku $1/3x^3 + .25x^2$ z dokładnością co do stałej. Jeżeli okaże się, że jakiejś całki kalkulator nie potrafi obliczyć, to daną funkcję podcałkową rozwija w szereg Taylora, a następnie całkuje tak otrzymany wielomian z niewielkim błędem wynikającym z zamiany funkcji na aproksymujący ją wielomian.

A teraz UWAGA nauczyciele i uczniowie. Oprócz podania prawidłowej odpowiedzi w błyskawicznym tempie, kalkulator może pokazać, w jaki sposób, krok po kroku, doszedł do otrzymanego wyniku, tzn., że gdy uczeń otrzyma na klasówce pytanie: „Proszę policzyć pochodną funkcji $\arctan(\ln/x^2)$ ”, to wystarczy przepisać odpowiedzi podawane przez kalkulator w kolejnych krokach i ocena bardzo dobra jest mierzona. Te wspaniałe własności można z powodzeniem wykorzystywać na wyższych uczelniach, gdyż i tu możliwości kalkulatora się jeszcze nie kończą. Daleki jednak jestem od zniechęcania do nauki matematyki. Uczyć się jej z pewnością warto, choćby dlatego, żeby się nie okazało, gdy jak wyłączą prąd, że nie potrafimy niczego policzyć na kartce papieru.

Powróćmy jednak na ziemię, do obliczeń numerycznych. Tu też czekają na nas nowości i rewelacje. Obliczenia prowadzone są z dokładnością do 12 cyfr. Niektóre funkcje wymagające podwyższonej dokładności prowadzą obliczenia z dokładnością do 16 cyfr. Dla liczb całkowitych mamy do dyspozycji 19 cyfr. Zakres numeryczny kalkulatora (dla fanatyków wielkich liczb i wykładników) to $+(-1.000000000000E-499$ do $+(-9.999999999999E499$. Numerycznie rozwiązywać można układy równań liniowych, prowadzić działania na liczbach zespolonych, na wektorach i macierzach. Można prowadzić obliczenia z zakresu rachunku prawdopodobieństwa i statystyki matematycznej, obliczenia w systemach: binarnym ósemkowym, dziesiętnym i szesnastkowym (operowanie na słowach maksymalnie 64-bitowych oraz zamiana liczb między tymi systemami). Kalkulator ma wbudowaną możliwość rysowania wykresów funkcji. Do tego celu służy ekran, który w trybie alfanumerycznym daje 4 linie po 32 znaki w linii, a przy grafice ma rozdzielczość 32×137 pixeli (prawie HIRE) co okazuje się być całkowicie wystarczające szczególnie, że wybrany fragment wykresu możemy powiększać dowolnie. Możliwe jest też graficzne przedstawienie danych statystycznych. W sumie HP-28S ma wbudowanych ponad 1500 różnych funkcji i rozkazów (głównie jeden tysiąc pięćset) HP-28S ma wbudowanych również 120 jednostek fizycznych, astronomicznych itp. Umożliwia on zamianę dowolnych jednostek np. stopni Fahrenheita na Celsjusza czy stop na cale.

Cechą użytkową nierozłącznie związaną z kalkulatorami Hewlett-Packard jest tzw. odwrotna notacja polska (RPN) dająca wyniki przy mniejszej liczbie wciśnięć klawiszy a przez to i szybciej. Cechą charakterystyczną RPN jest to, że operator wprowadzany jest dopiero po wprowadzeniu na stos argumentów. I tak dla obliczenia $(15+23) \sin 45$ należy wcisnąć kolejno 15 ENTER 23 + 45 sin. Dla osób, którym ten sposób liczenia nie przypadnie do gustu np. z powodu przyzwyczajenia wyniesionych z innych kalkulatorów, HP-28 daje możliwość prowadzenia obliczeń w trybie algebraicznym, czyli tak jak na kartce papieru przy wykorzystaniu nawiasów i priorytetu działań. W trybie algebraicznym obliczenie powyższego przykładu wyglądałoby następująco $(15+23) \times \sin(45)$. Już na tym prostym przykładzie widać, że faktycznie RPN wymaga mniejszej liczby wciśnięć klawiszy (tu 10 a dla systemu algebraicznego 14). Do tej pory kalkulatory Hewlett-Packard wyposażone były w cztero-poziomowy stos. W HP-28C i HP-28S stos jest zmienny i rośnie dynamicznie w miarę potrzeby co znakomicie ułatwia prowadzenie obliczeń.

Kalkulator, tak jak wszystkie inne wyroby HP, jest bardzo wygodny w użyciu. W stanie złożonym stanowi on książeczkę o wymiarach $158 \times 94 \times 18$ mm. Po rozłożeniu okładek na pierwszy rzut oka nie widać wszystkich wbudowanych funkcji. Przeważnie każdemu klawiszowi przyporządkowane są tylko dwie funkcje. Wszystko to dzięki temu, że okienka i menu trafiły już do kalkulatorów. Chcąc obliczyć logarytm musimy wcisnąć czerwony klawisz SHIFT a następnie klawisz oznaczony LOGS. Na ekranie pojawi się 6 okienek z funkcjami LOG, ALOG, LN, EXP, LNPI, EXPM. Każdą z funkcji otrzymujemy naciskając odpowiedni biały klawisz pod okienkiem menu. To jednak nie wszystko. Wciśnięcie klawisza NEXT udostępni nam kolejnych 6 okienek z następnymi funkcjami itd. W ten sposób, dzięki wielopoziomowemu menu mamy dostęp właśnie do tych wcześniej wspomnianych 1500 funkcji i rozkazów.

Kalkulator pozwala na definiowanie własnych funkcji, które następnie pojawiają się w menu użytkownika lub mogą być połączone w menu wraz z funkcjami standardowo wbudowanymi. Dzięki ciągłej pamięci po wyłączeniu kalkulatora raz wprowadzone informacje pozostaną w nim niezmiennie. Wspomniałem już, że kalkulator jest szybki i dokładny. Oto dowód. Przeprowadziłem na nim test, publikowany między innymi w Komputerze 7/87, polegający na obliczeniu 1000 razy funkcji SIN, COS, TAN, SQRT, LN i odwrotnych do nich ASIN, ACOS, ATAN, SQ, EXP dla zmieniających się argumentów. HP-28S zakończył test w 4 minuty z dokładnością własną $1.13E^{-11}$. Dla porównania C-64 (komputer) wykonał test w 8 minut 20 sekund z dokładnością $6E^{-8}$, a kalkulator Texas Instruments TI59 uporał się z problemem w 59 minut z dokładnością $8.57E^{-11}$. Wyniki mówią same za siebie. Różnica w szybkości widoczna jest również pomiędzy HP-28C i HP-28S np. obliczenie całki z wyrażenia $\exp(x^3+2x^2-x+4)$ w przedziale $(0,1)$ z dokładnością 0.000001 zajęło pierwszemu 20 sekund, a drugiemu tylko 9 sekund. Jako wynik kalkulator podaje całkę i błąd z jakim całka została policzona. HP-28 oblicza również wartości liczbowe i symboliczne wyrażań wprowadzonych jako wzory. Żądany wzór zostaje zapamiętany a obliczenie jego wartości dla różnych parametrów nie nasręca żadnych trudności.

Ponadto kalkulator jest w pełni programowalny i to wyłącznie strukturalnie, gdyż nie istnieje instrukcja GOTO. Posiada natomiast instrukcje ułatwiające strukturalne programowanie FOR... NEXT, START... STEP, WHILE... REPEAT, DO... UNTIL. Możliwe jest definiowanie zmiennych lokalnych które istnieją wyłącznie podczas działania programu w obrębie procedury czy podprogramu.

Całej pracy kalkulatora towarzyszą komunikaty o ewentualnych błędach. Komunikaty te to nic nie mówiące SYNTAX ERROR czy tylko ERROR, ale podają dokładnie rodzaj błędu i miejsce jego wystąpienia. Ponadto, gdy użytkownik zapomni jakich argumentów należało użyć do jakiejś funkcji, to zawsze może skorzystać z wbudowanego katalogu komend, który w sposób jasny i zwięzły wyjaśnia sposób użycia instrukcji i wymaganych argumentów. Kalkulator posiada możliwość sprawdzenia poprawności działania klawiatury oraz wszystkich wewnętrznych struktur- SELF-TEST. Prawidłowe funkcjonowanie objawia się komunikatem OK-28S. Możliwy jest też SYSTEM HALT przerywający zawieszony programy i RESET kalkulatora powodujący przywrócenie pierwotnej konfiguracji. Do kalkulatora można zakupić termiczną drukarkę mozaikową HP82240A. Sprzężenie z nią jest bezprzewodowe. Kalkulator emituje promieniowanie podczerwone, które odbierane jest przez czujnik drukarki. Drukarka posiada 2 szybkości pracy: wolne przy użyciu baterii i szybkie przy użyciu konwertera sieciowego.

Ponieważ kalkulatorów nie kupuje się wyłącznie na dziś, ale i z myślą o jutrze nie należy się więc przerażać olbrzymimi możliwościami HP-28S. Z czasem dojdzie się do ich pełnego wykorzystania, a jeżeli nawet będziemy wykorzystywać tylko kilka opcji np. grafika i statystyka, to przyjemnością będą szybkie i dokładne wyniki. Tak więc przed zbliżającą się Gwiazdką zastanówmy się czy naprawdę potrzebujemy superkomputer bez specjalistycznego oprogramowania, czy też kalkulator z dobrym i pełnym oprogramowaniem oraz z literaturą, zawierającą listingi programów, nie będzie wystarczającą alternatywą.

falcon



PRZENOŚNY TŁUMACZ

Wyobraź sobie, że jesteś obywatelem niemieckim, który idąc pewnego dnia ulicą spotyka dziwnie uśmiechniętego obcokrajowca. Ten bierze do ręki małe, elektroniczne urządzenie, przybliża je do ust i powoli doń przemawia: „Con you tell me where I can buy some sauerkraut?” (Czy mógłby mi pan powiedzieć, gdzie kupić kiszoną kapustę?).

Co powinieneś zrobić?

- a) Uciekać
- b) Wezwać policję
- c) Słuchać uważnie, co mówi to urządzenie po niemiecku: „Konnen Sie mir bitte sagen, wo ich Sauerkraut kaufen kann?” Najrozsądniejszą jest trzecia odpowiedź. Ten człowiek przed tobą to turysta, który próbuje poradzić sobie za pomocą techniki.

Wspomniane urządzenie jest pierwszym w świecie przenośnym tłumaczem — mieszczącym się w ręku komputerem tłumaczącym jeden mówiony język na inne. Ten zasilany bateriami, ważący niespełna 2 kilogramy produkt nazywa się „Voice” (Głos) i jest wytworem amerykańskiej firmy Advanced Products and Technologies, specjalizującej się w dodatkach dla podróżnych.

Kiedy Voice ukazał się w USA pod koniec kwietnia tego roku — cena 1500 \$ — był w stanie tłumaczyć mówiony angielski na włoski, niemiecki, francuski i hiszpański. Są dostępne osobne moduły pamięci (cartridges) dla każdego z tych języków, które można wymieniać w zależności od państwa, w którym znajduje się użytkownik.

Voice miał być wprowadzony na rynek europejski wkrótce po swym amerykańskim debiucie, wyposażony w moduły „konwertujące” włoski, niemiecki, francuski i hiszpański na angielski.

Urządzenie to pracuje w oparciu o prawnie zastrzeżony procesor i sztuczną inteligencję. „Ożywia” się na dźwięk słownego polecenia i produkuje głos przez wbudowany głośniczek. Na wypowiedziane przez właściciela zdanie odpowiada tym samym zdaniem w innym języku. „To cię słucha, a potem mówi” powiedział Steve Rondel, prezes Ad-

vanced Products. Twierdzi on, że choć Voice posługuje się „nieco chrypliwym dialektem komputerowym, to jednak osoba mówiąca danym językiem zrozumie go bez problemu”.

Ten produkt posiadając blisko 0.75 MB pamięci, jest w stanie rozpoznać i odpowiedzieć na ponad 2000 fraz — przede wszystkim prostych zdań, zwykle zawartych w tradycyjnych rozmówkach obcojęzycznych. Voice posiada również 16 liniowy ekran, na którym wyświetla przeliczenia walutowe i inne informacje — oczywiście po podaniu słownego polecenia.

Jego najbardziej dostrzegalnym ograniczeniem jest jednostronna translacja: po przetłumaczeniu pytania angielskiego na niemieckie nie może przetłumaczyć niemieckiej odpowiedzi na angielską. Stąd, jeśli nawet „tubylec” zrozumie komputer, to jego użytkownik może nie zrozumieć odpowiedzi. Steve Rondel twierdzi jednak, iż większość podstawowych kontaktów turystycznych polega na podawaniu kierunku, a tu wystarczy pokazywanie palcem, znacząca mina i prosty słownik — użytkownik z pewnością da sobie radę. Na pytanie kiedy będą osiągalne dwukierunkowe translatory, Steve odpowiedział, że mają już gotowe technologie, ale te wymagają bardzo szybkich procesorów, które są zbyt duże i kosztowne by je montować w urządzeniach przenośnych. W każdym razie jest przekonany, iż obcokrajowcy będą mogli swobodnie porozumiewać się ze sobą za pomocą „czarnych pudełek” już całkiem niedługo.

Na podstawie tygodnika „Newsweek” opracował (wist)

Żegnaj monitorze!?

U nas na razie bez niego nikt nie wyobraża sobie sprawnie działającego zestawu komputerowego. Szklany ekran, długa rura kineskopu i obowiązkowa obudowa. A wszystko to waży i, nawet zmniejszone, zajmuje sporo miejsca. Monitor, czy też przyłączony do komputera telewizor, jest nadal kulą u nogi prawdziwie przenośnego elektronicznego zestawu osobistego.

Ten standard się zmienia. Już do obowiązkowego wyposażenia podróżującego po świecie zachodniego reportera należy komputerowa maszyna do pisania wielkości arkusza papieru formatu A4. Złożona nie jest grubsza od typowej książki. Tekst pojawia się na ekranie zamontowanym w wieczku urządzenia. Sekret miniaturyzacji jest zastosowanie płynnego kwarcu. Litery wyczarowywane są więc na podobieństwo cyfr ukazujących się na naszych podręcznych zegarkach elektronicznych. Latem ubiegłego roku japońska firma Sharp wypuściła pobudzający marzenia posiadaczy komputerów lekki, superpłaski ekran przeznaczony do współpracy z profesjonalnymi i domowymi zestawami elektronicznymi. Ale wyzwanie rzucone telewizyjnej tradycji było jeszcze zbyt słabe. Wybredni stwierdzali wszak, zgodnie z prawdą, że z tej próby sił z kwarcową płytą wytwarzającą dzięki płynnym kryształom tylko czarno-białe obrazy, kolorowy kineskop wyszedł jeszcze obronną ręką. Nikły ciężar urządzenia nie mógł być jeszcze atutem w rozgrywce o przyszły kształt systemu obrazowania.

Dopiero we wrześniu amerykański magazyn „Newsweek” doniósł, iż Sharp nie zrezygnował z walki. W laboratoriach koncernu w Osace zaprezentowano bowiem coś, co z pewnością już w najbliższym czasie zrewolucjonizuje komputerową wizualizację i technikę telewizyjną. Tym czymś jest kolorowa wersja urządzenia sprzed roku czyli, Color Liquid Crystal Display Screen w skrócie LCD. Zdaniem rzeczownika prasowego japońskiego potentata (Sharp uważa się w obecnej chwili za wiodącego wytwórcę elektronicznych cudów) ekran wyświetlający kolorowe obrazy przy pomocy płynnych kryształów kwarcu już teraz dorównuje jakością wizualizacji najlepszym monitorom kineskopowym. Badania i doskonalenie systemu będą trwały, zanosi się więc na ogromny przewrót, którego zasięg trudno w tej chwili sobie wyobrazić.

Ekran Sharpa może wytworzyć aż 512 kolorów i w pierwszej kolejności kusić będzie miliony zasobnych w gotówkę posiadaczy komputerów. Z całą pewnością następnym krokiem będzie wbudowanie japońskiej nowości do elektronicznego układu współczesnego telewizora. W efekcie powstanie odbiornik, który będzie sobie można powiesić na ścianie niczym obraz. Cała ich gama od maleńkich po ogromne przeznaczone do dużych pomieszczeń, ma zostać zaprezentowana już w roku 1989.

Na razie Sharp oferuje dwa modele swoich kwarcowych ekranów. Oba mają przekątną 14 cali (355,6 mm). Pierwszy przeznaczony jest do odbioru cyfrowych impulsów telewizyjnych i może być stosowany na przykład jako wizualizator obrazów ruchomych z cyfrowych magnetowidów. Drugi zaś skonstruowany został do współpracy z klawiaturami komputerowymi. Na razie cena nowości jest dość wysoka. W Stanach Zjednoczonych kwarcowe ekrany 14-calowe powinny być już sprzedawane wiosną 1989 roku za 1500 dolarów.

Sami Amerykanie jednak też nie zasympują gruszek w popiele i za 18 miesięcy, jak donosi „Newsweek”, również włączą się do rywalizacji z Japończykami. Ich działania zmierzają również w kierunku uniwersalizacji komputerów, w stronę uczynienia z nich podręcznych przyrządów każdego szanującego się człowieka interesu, naukowca, czy dziennikarza. Do problemu eliminacji uciążliwych monitorów telewizyjnych podszedli oni jednak zupełnie inaczej niż Japończycy. Spółka Reflection Technology w Cambridge w stanie Massachusetts spojrziała na zagadnienie bardziej konwencjonalnie, po prostu maksymalnie miniaturyzując znany wszystkim ekran.

Urządzenie ochrzczone mianem handlowym Private Eye (Prywatne Oko), którego prototyp przedstawiono właśnie dziennikarzom za Atlantykiem, ma masę około 2 uncji — 62,2 g, może więc być śmiało noszone, jak reklamuje firma, w każdej kieszeni, niczym podręczny kalkulator. Z przedmiotu o wymiarach 1 na 1,2 na 3,2 cala (25,4 mm na 30,48 mm na 81,28 mm) wyłania się okienko mikroekranu. Przy pomocy kabla przyłączonego do jakiegokolwiek komputerowego banku danych, klawiatury, czy innego sprzętu elektronicznego, w szklanym okienku Prywatnego Oka zjawiają się wedle woli kolumny cyfr wykresy, rysunki czy tekst. Private Eye jest więc super zminiaturyzowanym monitorem wytwarzającym obrazy o wysokim standardzie, takie jak te, które otrzymujemy na ekranie standardowego odbiornika o przekątnej 12 cali (304,8 mm). Pod warunkiem oczywiście, iż do Prywatnego Oka zbliżymy maksymalnie nasze własne. W sumie na ekranie mieści się 25 wierszy tekstu (w każdym wierszu po 80 znaków).

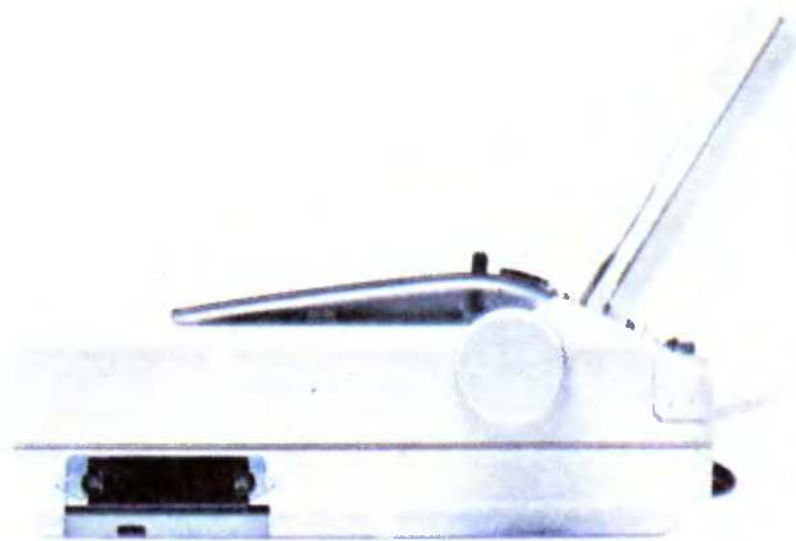
Za rok z okładem Reflection Technology wprowadza więc do akcji coś, co rozwiązało ma po części problem posługiwania się komputerami w każdej sytuacji. W przyszłości cienki przewód zastąpiony powinien być, jak zapewnia szef firmy, łącznością radiową, czy podczerwienią. Wyobraźmy sobie zatem podróżującego samolotem handlowca, któremu nagle potrzebne będą dane z centralnego komputera jego przedsiębiorstwa. W takim wypadku wystarczy połączyć się telefonicznie z pokładem maszyny z odpowiednim numerem na ziemi (tego rodzaju usługi zapewnia już kilka powiatowych przewoźników) wyjąć z kieszeni Prywatne Oko i za pomocą serii sygnałów dźwiękowych rzucić na miniekran potrzebne informacje.

Jest to jedno z wielu tysięcy możliwych zastosowań Private Eye. Reflection Technology spodziewa się zamówień przede wszystkim z kręgów ludzi interesu. Ich praca wymaga ciągłego przemieszczania, a oni nie są w stanie z oczywistych powodów wozic ze sobą całego komputerowego oprzyrządowania łącznie z pełnoskalowymi monitorami. Klientami firmy kierowanej przez Allena Beckera powinni być również dziennikarze, którym Oko zapewni w każdej chwili potrzebne dane z komputerowych zasobów.

Cena wahać się ma od 50 do 2000 dolarów w zależności od wyposażenia dodatkowego, takiego jak na przykład system integracji z telefonem. Także od wielkości zamówień. Przedsiębiorstwa skłonne przejść na system Private Eye w całości, mają ponoć zagwarantowane poważne obniżki.

Czy to wszystko oznacza całkowite i definitywne pożegnanie z klasycznym monitorem telewizyjnym dołączonym do komputerowej klawiatury? Na razie nikt tego sobie nie może wyobrazić. Z dużą dozą prawdopodobieństwa można jednak przewidzieć ich powolny zmierzch w najbliższych latach szczególnie w świecie profesjonalnej informatyki. Nam, amatorom służyć wszakże będzie jeszcze długo. Kineskopowa lampa ma w każdym razie w naszej części Europy zapewnioną przyszłość.

Wojciech Łuczak



Wszechstronna

i do tego kolorowa



star

LC-10 Colour

Druk kolorowy: LC-10 colour drukuje w 7 kolorach!
Po założeniu kasety z czarną taśmą, drukuje również na czarno.

Najwyższy poziom technologii japońskiej: funkcja „Paper Park”: możliwość stosowania pojedynczych stron oraz papieru z perforacją.

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

Łatwość użytkowania: Kilkanaście funkcji wybieranych za pomocą przycisków na obudowie.

Szybkość druku: 120 lub 144 zn/sek w trybie standart; 30 lub 36 zn/sek w trybie korespondencyjnym.

Pełna oferta: Oprócz LC-10 colour oferujemy Państwu również wszystkie pozostałe modele drukarki STAR.



LC-10 — najpopularniejsza drukarka świata!	DM 450
LC 24-10 — NOWOŚĆ! dostępna dla każdego 24-igłowa!	DM 770
NX-15 — sprawdzona, Nr 1 w Polsce!	DM 700
SG-15 — taśma na szpulkach, 16 KB pamięci!	DM 700
NR-15 — profesjonalna, 240 zn/sek!	DM 1200
NB 24-15 — szybka 24-igłowa, 216 zn/sek!	DM 1350
LS-03 — laserowa, 8 stron/min, 1 MB pamięci!	DM 4500
Koszty przesyłki: DM 40; kabel: DM 20	

Wylączny autoryzowany przedstawiciel na Polskę:

ABC Data

peripherals & computer systems

star

Tvoja 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

BO NIC CO LUDZKIE,
NIE JEST MU OBCE,
BAJTKU DOPOMÓŻ,
JESTEŚMY W KROPCE!



Fot. Roman Sumik

Rozmowa z Krzysztofem Gradowskim, reżyserem, autorem filmów o przygodach profesora Ambrożego Kleksa — „Akademia pana Kleksa” i „Podróże pana Kleksa” znane są chyba wszystkim dzieciom. Już za kilka dni na ekrany kin wejdzie „Pan Kleks w kosmosie”.

— Dlaczego profesor Ambroży Kleks, mając do wyboru tak wiele pięknych baśniowych krain, wybrał się w kosmos, w świat komputerów, elektroniki i... przedziwnych stworów „nie z tej Ziemi”?

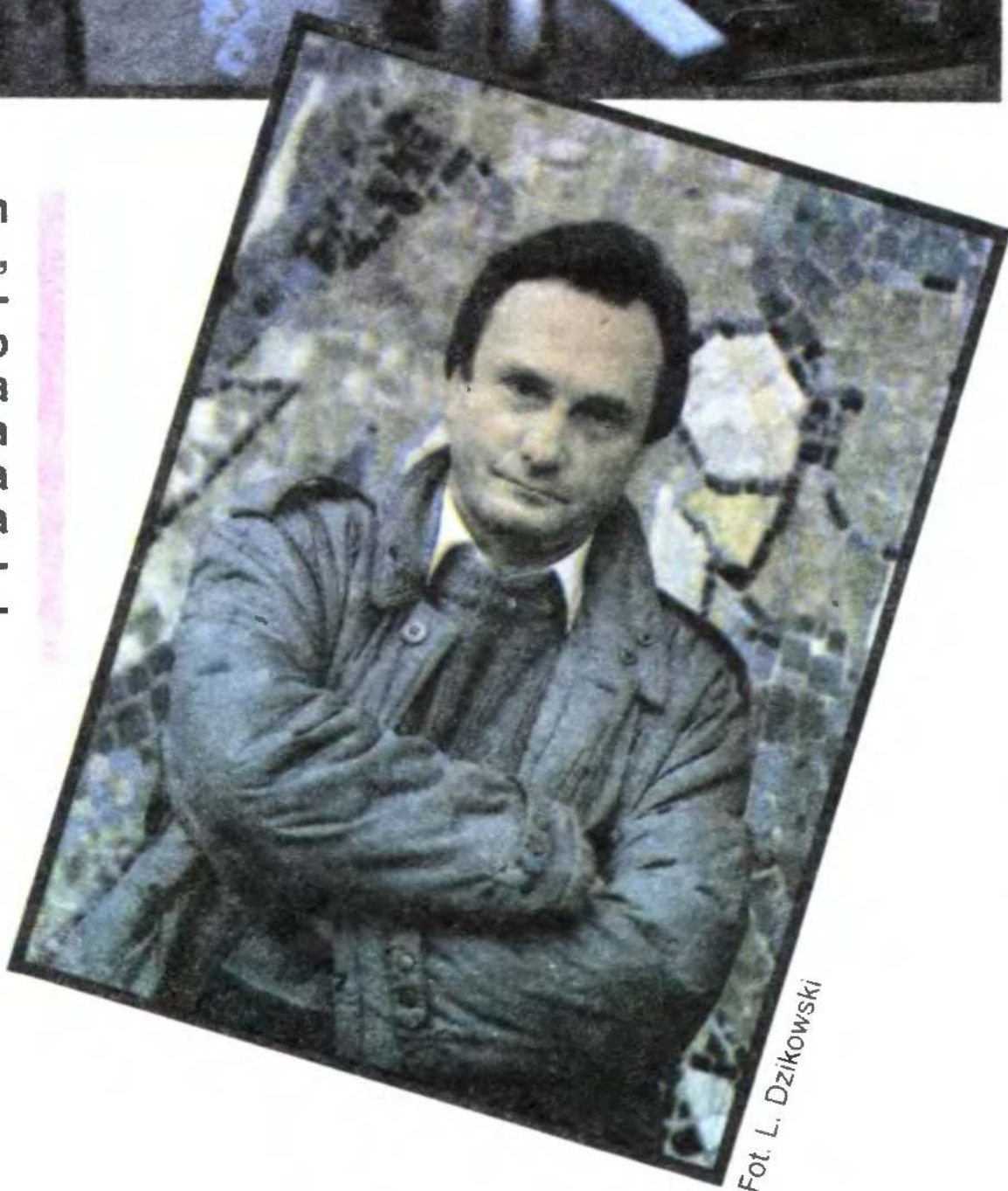
— Ten kierunek podróży pana Kleksa wymyśliły dzieci. Po filmie „Podróże pana Kleksa” zamierzałem rozstać się z tą filmową przygodą. Ze względu na stopień komplikacji technicznych nie wyobrażałem sobie ekranizacji „Triumfu pana Kleksa”, trzeciej części trylogii napisanej przez Jana Brzechwę.

I wtedy, nastąpiło coś, co było dla mnie zupełnie nową przygodą w kontakcie z widzami. Zostałem zasypany wielką liczbą listów. Przyszło ich kilkadziesiąt tysięcy. Sam trochę przyczyniłem się do wywołania tej burzy, bo w programie — ulotce do filmu „Podróże pana Kleksa” zwróciłem się do widzów z pytaniem, czy ta opowieść powinna być kontynuowana. Myślałem wówczas o serii telewizyjnych spotkań z Kleksem, w formie promocji dla dwóch poprzednich filmów. Tymczasem z listów wynikało, że dzieci chcą, aby pan Kleks poleciał w kosmos. I wówczas Zespół Filmowy „ZODIAK” zamówił u mnie taki scenariusz.

— Skoro przyjął Pan to zamówienie, to przypuszczam, że uznał Pan za słuszny ten właśnie kierunek wojaży pana Kleksa. Jako artysta ma Pan przecież prawo wyboru.

— Ja pojmuję swoją funkcję jako rodzaj działalności usługowej. Jestem oddany do dyspozycji widza. Kino sztuką bywa bardzo rzadko. Zdarzają się oczywiście arcydzieła, ale pojawiają się raz na parę lat.

Jeżeli natomiast chodzi o mój pogląd na to, czy kosmos jako kierunek podróży Kleksa był najwłaściwszy, to muszę przyznać, że specjalnie się nad tym nie zastanawiałem. Mnie kosmos fascynuje tak, jak moją córkę i miliony innych dzieci. W ogóle świat współczesnej techniki, elektroniki jest dla mnie nowym rodzajem magii. Nie jestem elektronikiem ani matematykiem, nie potrafię spojrzeć na ten świat jak inżynier, a więc widzę w



Fot. L. Dzikowski

BAJKA O KOSMOSIE

WSZYSTKO DLA WSZYSTKICH

nim działalność niepojętych sił, które powinny być poddane ocenom humanistycznym.

Wydaje mi się, że dzieci właśnie w ten sposób pojmują kosmos. Dla nich jest to nowy rodzaj baśni. Motywy takie jak tajemnicze zamki, wróżki, smoki, zostały w pewnym sensie wyeksplorowane, (choć pojawiają się echem na przykład w programach gier komputerowych). Na nikim nie zrobi dziś wrażenia tajemnica zamku w Karpatach. Teraz w Karpaty podróżuje się z wycieczką Orbisu. Kosmos natomiast, w wyobraźni dziecka jest miejscem, w którym bohaterów jego opowieści spotkać mogą nieprawdopodobne przygody.

— **Czy uważa Pan, że wejście najnowocześniejszej techniki do literatury dziecięcej jest czymś zupełnie nowym, czy jedynie kontynuacją dawnych form i treści?**

— Myślę, że mitologia kosmosu stwarza nowy sposób spojrzenia na sprawy ludzkie. Jeżeli mówimy o Ziemi w kategorii planety i widzimy, że jest ona umieszczona w jakimś ogólnym systemie, to troszeczkę inaczej traktuje się pojęcie wspólnoty międzyludzkiej. Nagle dostrzegamy — i sędzę dziecko odczuwa podobnie — że nasz świat, to jeden wspólny organizm. Że nie jest to szereg zupełnie odrębnych państw, które mogą toczyć ze sobą wojny. Wobec niewiadomej, jaka czeka nas we wszechświecie, ludzie powinni się zjednoczyć. I to przesłanie wydaje mi się bardzo wartościowe.

— **To Pańska wizja. Częściej jednak mamy do czynienia z zupełnie przeciwną filozofią pojmowania kosmosu. W „Gwiezdnym Wojnach” jest on jedną wielką areną walki.**

— Można dyskutować, czy treści zawarte w filmach Lucasa i Spielberga są właściwe. Czy transponowanie pewnych, dosyć już wytartych fabuł i schematów fabularnych na opowieści o kosmosie jest celowe. Mnie fascynuje w tych filmach technika i sposób organizowania widowiska dla publiczności.

W „Kleksie w kosmosie” widz nie zobaczy wojujących społeczeństw. Jest to film o odpowiedzialności za drugiego człowieka. Zostaje uprowadzona dziewczynka, i cała klasa próbuje ją uratować. Pan Kleks jest traktowany nieco instrumentalnie. To po prostu dobry przyjaciel, który pomaga dzieciom.

Bardzo ważne jest powiedzenie młodym ludziom — właśnie teraz — o czymś takim, jak partnerstwo międzyludzkie. Nastąpiła ogromna dezintegracja. Ludzie są rozproszeni, skłócenia, patrz na siebie wilkiem. Już w najmłodszych klasach zaczyna się rywalizacja, inspirowana oczywiście przez rodziców. Dzieci są ze sobą w pewnej wewnętrznej niezgodzie. Kleks robi krok w kierunku wzajemnego zrozumienia, odpowiedzialności za partnera.

— **Pan wykorzystuje najnowocześniejszą technikę do propagowania idei głęboko humanistycznych. Czy**

„Bajtek”, będący pomostem pomiędzy młodym człowiekiem, a tą techniką jest w stanie — Pańskim zdaniem — podolać przesłaniu tej idei?

— „Bajtek” zachwylił mnie sposobem podania tego, co było dla mnie „abrakadabram”. Można wyobrazić sobie, że pismo o komputerach to brulion, w którym znajdują się wytłaczane schematy i programy. W przypadku „Bajtki” jest przeciwnie. Waszą ofertę myślową i plastyczną, sposób rozmowy z czytelnikiem uważam za niesłychanie współczesny. Przede wszystkim, „Bajtek” jest pierwszym piśmie przeznaczonym dla młodego odbiorcy, które traktuje go po partnersku. Nie jest to „Piomyk”, który mówi: „no to małe dzieci, teraz siadajcie, a my, starsi opowiemy jak to powinno wszystko wyglądać”. Lansujecie młodych ludzi, którzy chcą rozmawiać z jeszcze młodszymi, także na zasadzie pewnej wspólnoty.

Poprzez „Bajtkę” zrozumiałem, na czym polega zabawa z komputerem. Że komputer to po prostu maszyna, taka jak wiele innych, która ma służyć człowiekowi. I że spotkanie z nim to wspaniała przygoda, ale jedynie przygoda, jak wiele innych przygód w życiu człowieka.

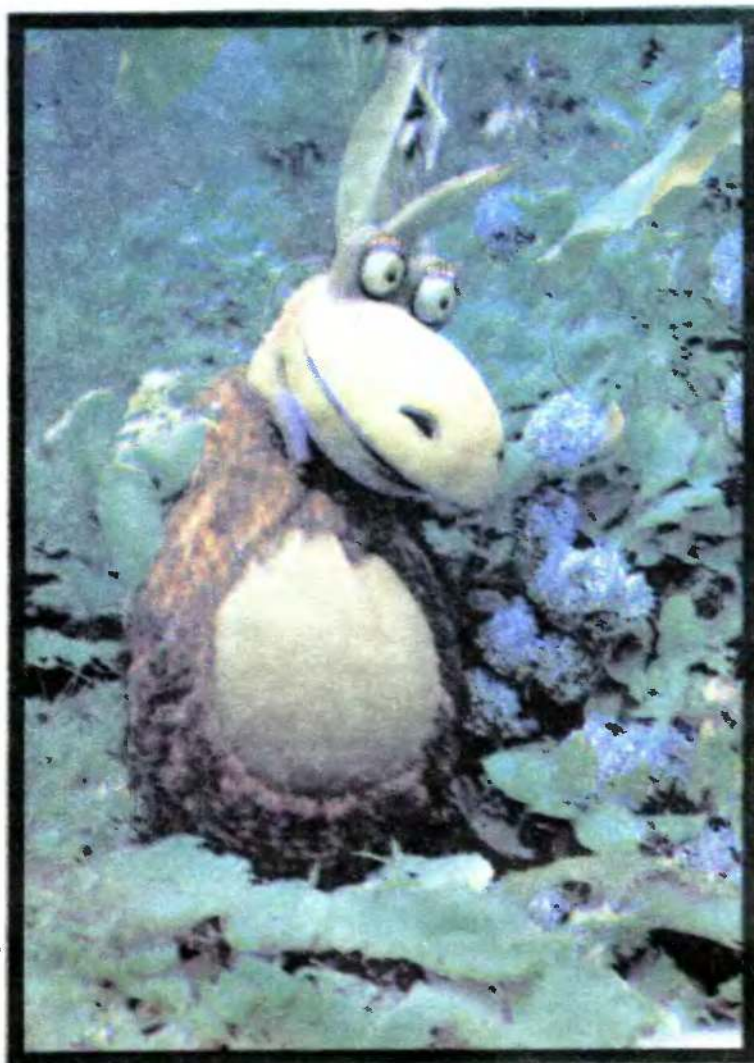
W wyniku konkursu na pomysł robota występującego w filmie, niespodziewanie okazało się, że większość dzieci chciała, aby nosił on imię „Bajtek”. Podjąłem tę propozycję bo wydaje mi się, że „Bajtek” znalazł dobry sposób kontaktowania się ze swoim czytelnikiem, a moim zadaniem jako twórcy filmowego, jest znalezienie pomostu łączącego ekran z widownią.

— **Czy możemy liczyć na następne spotkanie z panem Kleksem?**

— Uważam, że lepiej rozstawać się z widzem z pewnym uczuciem niedosytu, niż przeciwnie. Jan Brzechwa jest dla mnie nie tylko autorem, ale także inspiratorem pewnego stosunku do dziecka, patrzenia na świat. On — jako jeden z niewielu w polskiej twórczości literackiej dla dzieci — podjął trud edukacji poprzez zabawę. Nie transmisji treści pedagogicznych na siłę, nie kształtowania modelu wychowawczego na zasadzie oleju wlewane go do głowy. Przy okazji zabawy słownej, sytuacji fabularnej oferuje dzieciom pewien system wartości.

Trzecia część przygód pana Kleksa jest przygodą nie opisaną przez Jana Brzechwę. Wynika jednak z jego przesłania. Wydawało mi się, że pewne filmowe wątki zostały zawieszony w próżni. Trzecia część splota je i pozwala traktować te trzy filmy jako pewną zamkniętą całość. Myślę, że naruszenie tej całości byłoby eksploatacją ponad wytrzymałość materiału.

Rozmawiał
Roman Poznański



Fot. Krzysztof Gradowski

To Melo Śmiacz z ciemnego boru, to Melo Śmiacz pełen humoru. Wróg ponuraków, przyjaciel ptaków, postrach wampirów i nocnych mar!

ATARI ● SPECTRUM
● SHARP

nowości programowe
instalowanie TURBO w ATARI —
bezpłatnie!
horoskopy indywidualne.
NOWOŚĆ !!!
wysyła pocztą oraz oferuje na
miejscu
ASTRO-KOMPUTER STUDIO
54-515 WROCLAW, GDACJU-
SZA 39 (a-bus 139) G-133

— ATARI —

Szeroki wybór oprogramowania na kasetach i dyskietkach

- co piąty program bezpłatny
 - gwarancja jakości
 - rachunki
 - katalogi gratis
- ART-SOFTWARE
66-542 Zwierzyn, P-1 D-158

ATASERW

39-460 Nowa Dęba, skr. poczt.
Tel. Tarnobrzeg 46-22-58
oferuje świetne rozwiązania sprzętowe

do ATARI XL/XE:

1. Ataserw DOS-pierwszy DOS na kartidżu
 2. TOP Drive 1050 — wersja 3.0
 3. Pióro świetlne — wersja 4.0
 4. Basic XE-kartidż oraz kartidże z dowolnym programem
 5. Rozszerzenia pamięci do 128 i 256 kB
 6. Programator pamięci EPROM
 7. Interfejs Atari — Centronics
 8. Oprogramowanie użytkowe
- Informacje po otrzymaniu koperty zwrotnej, dla instytucji rachunki. K-212

UZYTKOWNICY ATARI XL/XE
ATAREX oferuje TANIO!
duży wybór programów do komputerów ATARI!
na taśmach kasetowych oraz dyskietkach.
Szczegółowych informacji po zwróceniu znaczka udziału:
ul. 22 Lipca 17 62-300 WRZESNIA
ATAREX
ul. 20 Października 42/27 63-000 SRODA WLKP. K-102

„BETA B”
AGENCJA
INFORMATYCZNA

Tel.
632-935
690-385
41-200 Sosnowiec, Skrytka 254
oferuje również wysyłkowo:
Programy, Instrukcje,
Literaturę dla komputerów

ACORN AMSTRAD
ATARI COMMODORE
SHARP IBM K-106

UDOSKONALENIA
PROGRAMOWE I SPRZĘTOWE
DLA WSZYSTKICH
MODELI ATARI
ORAZ KOMPUTERÓW PRACUJĄCYCH
POD SYSTEMEM MS-DOS
WYSYLA POCZTĄ
agencja mikro-
komputerowa *
41-200 Sosnowiec P-157

ATARI
ZX SPECTRUM
TIMEX

programy, instrukcje, nowości,
oferuje:

SPEKTRA

21-422 Stanin
Informacje kopertą zwrotną D-100

„Atares” spółka z o.o.
Chorzów Ba-
tory, Jesionowa 3, tel. 417-573 oferuje:
oprogramowanie „Atari”, „Commodo-
re”, „Spectrum”, system „Blizzard Tur-
bo”, magnetofon „Atari” po przeróbce
czyta 10 razy szybciej i pewniej, rocz-
na gwarancja, cena 33.200 zł. system
„Crystal Sound” do digitalizacji dźwię-
ku „Atari”, cena 15.500,- komis, skup,
sprzedaż komputerów, RTV, video. K-217

ATARI
COMMODORE C16 C64 C+4
● Największy wybór programów
● Ekspresowa obsługa
● Ceny — prawie za darmo
● Literatura
● Katalogi gratis
KOMBIT
ZWYCIĘSTWA 143/6
75-950 KOSZALIN G-124

SPECTRUM ● ATARI
AMSTRAD

— literatura w języku polskim
— gry i programy użytkowe
— polskie programy edukacyjne
wysyła na cały kraj

MEGABAJT

ul. Paryska 17/29
03-945 WARSZAWA
tel. 17-76-16 D-181

— ATARI — ZX SPECTRUM

**INSTRUKCJE
OPISY
LITERATURA**

Szkoły i Kluby — Zniżka
Katalogi — Gratis
Co piąty program — Gratis
Wysyłka na cały kraj
Wypożyczalnia programów
D.H. „Sezam” II p. g. 16.00—19.00
00-849 Warszawa UPT 66, skr. p. 14.

D-163

WOJEWÓDZKIE PRZEDSIĘBIORSTWO HANDLU WEWNĘTRZNEGO ODDZIAŁ W TYCHACH

VIDEOBIT

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

Poleca dla j.g.u.:

- 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

Gry, programy użytkowe, opisy na Atari XL/XE oferuje „MIKROFAN” 45-064 Opole 1, skr. poczt. 158.
(informacje za załączeniem znaczka)

G-110

**TWÓJ SPECTRUM
W NOWYM WYDANIU!**

**Nawet Zamożni
Oszczędzają**

UNIMAK oferuje przystawkę do ZX Spectrum (Timex) zawierającą

- stację dysków 5,25"
- sterownik drukarki
- sterownik joysticka
- sterownik monitora
- zestaw do ćwiczeń laboratoryjnych

System umożliwia przeniesienie zbiorów do (z PC) XT IBM; oraz tworzenie sieci, do 10 komputerów ZX Spectrum bez konieczności instalowania żadnych dodatkowych urządzeń.

UNIMAK
62-041 Puszczykowo/k. Poznań
ul. Kopernika 3, tel. 133 444; 40701

D-172

„MIKROELEKTRONIKA OD PODSTAW DLA KAZDEGO”
Błyskawicznie, tanio, rewelacyjną metodą — od prawa Ohma do poznania możliwości i wnętrza mikrokomputerów. Wysyłkowa sprzedaż wiedzy oraz płytek do samodzielnego montażu mikrokomputera CA80 skierowanego na sterowanie. Szczegółowa wielotomowa dokumentacja. Informacje — koperta zwrotna ze znaczkiem „MIK” Stanisław Gardynik, 05-090 Raszyn.

D-157

Studio komputerowe

ATARI-BAJT

ATARI ● AMSTRAD
COMMODORE ● SPECTRUM

oferuje najnowsze programy edukacyjne, użytkowe, gry, opisy interfejsy do wpisywania programów z każdego magnetofonu i TURBO — interface ATARI FINAL II — C 64
tel. 20-80-34 Warszawa
do zamówienia katalogi — gratis.

D-151

— ATARI — ZX SPECTRUM

— programy użytkowe i gry
— dokładne informacje po nadesłaniu koperty zwrotnej na adres:

05-220 ZIELONKA
skr. pocztowa 9/2

D-149

ZX Spectrum

Naprawy komputerów
Naprawy klawiatur
Interfejsy
KEMPSTON — 7.400
SINCLAIR — 7.900
KEMPSTON +
+ SINCLAIR — 12.800
RESET dopłata — 700

ATARI

Interfejs dla każdego magnetofonu — 8.000
Ceny roku 1988.
Natychmiastowa wysyłka pocztą za pobraniem WiNUE ul. Meissnera 14 m 1
03-982 Warszawa
tel. 15-93-38 wieczorem

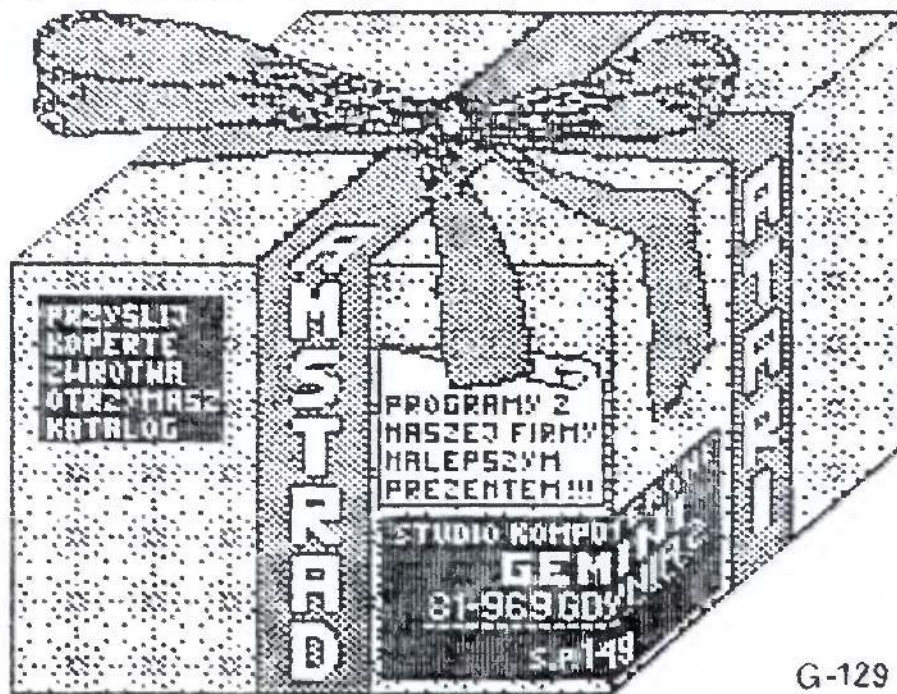
D-137

STUDIO KJOWIANKA
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-143

Programy do ATARI
Robert Sliwinski
Pobłocie 96/6
76-222 Pobłocie

D-155



G-129

INTERFACE DO SPECTRUM

system Kempston 9300,- zł
system Sinclair (dla dwóch joysticków) . . . 9800,- zł

JOYSTICK

(również do Atari 65 XE i 130 XE 6700,- zł
Gwarancja: interface 12-mcy joystick 6 m-cy
Wysyłka natychmiastowa za zaliczeniem pocztowym
Dla instytucji rachunki płatne przelewem
Elektromechanika, ul. Cegielniana 17,
32-410 Dobczyce

G-123

KONKURS „Mam Plan”

Rozpoczęcie konkursu 1988-10-16
Już od 1 października program „Mam plan” możesz kupić osobiście w sklepach RSW i CSH lub drogą pocztową u producenta:

M.C.I. POLSOFT
00-495 Warszawa, Al. Jerozolimskie 2
tel. 27-87-73 w. 7

cena 2000 zł.

K-223

COMPUTER SERVICE

IBM® PC·XT/AT
KOMPATYBILNE

K-57

ZX·Spectrum

Amstrad TIMEX

Schneider Sharp

Reg Trade Marks of IBM Corporation.

PMS ELEKTRONIK

37-76-65

WARSZAWA

ul. LEGIONOWA 23, ☐ 01-343

„AKCES-SYSTEM”

ul. K. Marksa 169
80-416 Gdańsk-Wrzeszcz
tel. 411-901

**poleca !!!
sprzęt i oprogramowanie**

Atari XL/XE/ST
Commodore 64/16/116/+4
IBM PC XT/AT/PS 2

rozszerzenia sprzętowe

Atari XL/XE/ST
Floppy 5 1/4" do ST

K-163

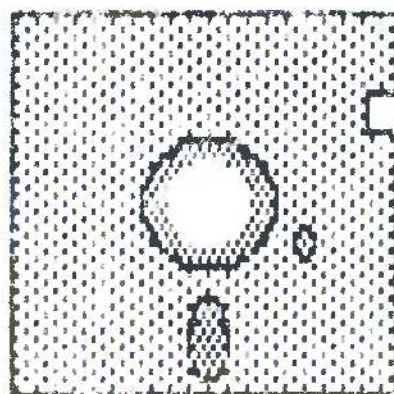
	GIEŁDA „BAJTKA” (tys.zł)	PEWEX BALTONA (USD)	RFN (śred.) (DM)
SINCLAIR			
ZX-Spectrum 48 k	125	115	90
ZX-Spectrum plus	160	—	99
ZX-Spectrum + 2	260	—	180
ZX-Spectrum + 3	325	—	280
TIMEX 2048	165	146	—

COMMODORE			
C-64C	300	219	275
C-128	400	299	399
C-128 D	900	—	777
Magnetofon 1530, 1531	70	48	40
Stacja dysk. 1541	310	—	315
Stacja dysk. 1570	345	—	369
Stacja dysk. 1571	360	299	495
Drukarka MPS 801	210	—	—
Drukarka MPS 803	270	—	—
AMIGA 500	1,4	—	888
C-16	120	—	90

ATARI			
65 XE	170	125	100
130 XE	200	199	199
Stacja dysk. 1050	260	185	299
LDW 2000 Super	360	199	—
XC-12	60	48	48

AMSTRAD			
464 mono monitor	650	—	320
464 kolor monit.	725	—	450
6128 mono monit.	895	—	689
6128 kol. monit.	1.0	—	777
PC 1512 SD MM	1.5	—	889
Dyskiety 3"	5	—	4-9
Dyskiety 3.5"	2.8	—	3-9
Dyskiety 5.25"	0.9-1.5	—	0.6
Joystick	6-15	4-9	6-25

Sklep BAJTKA w Bytomiu ul. Koniewa 6 tel. 81-57-01	
ZX-Spectrum 48 K	130.000
ZX-Spectrum plus	175.000
ZX-Spectrum +2	280.000
SEIKOSHA GP-50S	120.000
Commodore 64	240-270.000
Commodore 128	380.000
Commodore 128 D	—
AMIGA 500	—
Commodore 16	120.000
Commodore 116	100.000
Commodore +4	165.000
Magnetofon 1530	50-60.000
Stacja dysków 1541	270.000
ATARI 65 XE	190.000
ATARI 130 XE	260-290.000
XC-12	60-65.000
Stacja dysków 1050	240-270.000
LDW 2000 Super	270.000
464 mono	340.000



INDYWIDUALNY
BANK
DANYCH

Radosław Dubella, lat 15. Posiada mikrokomputer C-64, magnetofon, dwa joysticki, około 1500 programów i gier. Proponuje wymianę doświadczeń oraz gier. Adres: 75-347 Koszalin, ul. Władysława IV 58a/37.

Marcin Pośpiech, uczeń lat 11. Posiada ZX Spectrum 48 KB oraz magnetofon Panasonic. Zainteresowania: informatyka, sport i plastyka. Oprogramowanie: gry i programy użytkowe. Nawiąże kontakt w sprawie wymiany informacji o grach, proponuje wymianę gier. Adres: 53-404 Wrocław, ul. Stalwowska 4/7.

Krzysztof Pawłowski, uczeń lat 15. Posiada Atari 800 XL oraz około 200 programów. Proponuje wymianę doświadczeń i programów. Adres: 98-300 Wieluń, ul. Kopernika 13.

Wiesław Albrant, pragnie nawiązać kontakt listowy z posiadaczami Atari 800 XL lub 65 XE w celu wymiany programów a głównie doświadczeń w dziedzinie zastosowania małych mikrokomputerów do półprofesjonalnych zadań w projektowaniu obliczaniu konstrukcji. Adres: 01-926 Warszawa, ul. Kwitnąca 8 m 13.

Justyna Łaszewska, lat 10. Uczennica Szkoły Muzycznej I-go stopnia. Posiada Commodore+4, joystick, magnetofon 1531 i telewizor. Proponuje

wymianę gier i programów użytkowych. Adres: 81-704 Sopot, ul. Kościuszki 9 m 2.

Jarosław Żarczyński, uczeń lat 17. Posiada Commodore VC-20, magnetofon 1531, telewizor w systemie Pal-Secam. Programy: użytkowe — matematyka, fizyka, gramatyka, oraz gry. Adres: 11-200 Bartoszyce, ul. Nad Łyną 9/16.

Krzysztof Kraska, lat 14. Komputer: Commodore C-128, programy muzyczne, użytkowe oraz sporo gier. Zainteresowania informatyka i elektronika. Proponuje wymianę programów literatury oraz doświadczeń (szczególnie programy graficzne na drukarkę). Adres: 66-400 Gorzów Wlkp., ul. Walczaka 15a/1.

Irena Kiedroń, studentka lat 20. Posiada Atari 800 XL, magnetofon XC 12. Nawiąże korespondencję w celu wymiany literatury, doświadczeń i oprogramowania. Korespondencja w języku polskim. Adres: 735-64 Havirov-Sucha, CSSR, ul. Szazarmovska 13.

Tomasz Karaśkiewicz, uczeń LO, 17 lat. Posiada komputer Amstrad-Schneider 464. Oprogramowanie: programy użytkowe, edukacyjne oraz gry. Proponuje wymianę programów oraz doświadczeń. Adres: 99-300 Kutno, ul. Zamoyskiego 9/64.

Ziemowit Ogrodowski, uczeń 14 lat. Posiada mikrokomputer C 16 + 64 kb, magnetofon 1531. Oprogramowanie: około 100 programów. Proponuje wymianę programów z użytkownikami C-16 z powiększoną pamięcią do 64 kb. Adres: 63-500 Ostrzeszów, ul. Bolesława Śmiałego 5.

Marek Adamus, inżynier mechanik. Posiada mikrokomputer Atari 800 XL, magnetofon XC 12. Zainteresowania: radioelektronika. Oprogramowanie: około 100 gier oraz programów użytkowych. Proponuje wymianę oprogramowania, literatury i doświadczeń. Adres: 41-908 Bytom, ul. Nowa 19a/6.

REKLAMUJ SIĘ W BAJTKU!

NAJWYŻSZA
JAKOŚĆ

NAJNIŻSZA
CENA

Jednostki pamięci na dysku elastycznym 130 mm (5 1/4") typu SLIM-LINE o pojemnościach:

360 KByte w cenie od 139.000,-
do 125.000,-
720 KByte w cenie od 167.000,-
do 150.000,-
1,2 KByte w cenie od 183.000,-
do 165.000,-

Termin dostawy dla wszystkich typów w ilościach 1.000 szt. i więcej w ciągu 30 dni. Zapewniamy serwis gwarancyjny i pogwarancyjny oraz udzielamy 12-to miesięcznej gwarancji.

oferuje

Krakowska Fabryka Aparatów Pomiarowych

„MERA-KFAP”

Kraków ul. G. Zapolskiej 38
Telefon do Działu Handlowego 37-87-20

(SB-7)

Drogi Bajtku!

Na listy Czytelników
odpowiada
Marek Waligórski

-100 jest „problem kabelka”. Zarówno drukarka, jak i kabel od niej posiadają nietypowe gniazda i wtyki. Dodatkowym problemem jest oprogramowanie transmisji danych do drukarki, zwłaszcza w trybie graficznym. Należy je wykonać samemu. Jako ciekawostkę mogę dodać, że również autorzy oprogramowania Elwro 800 Junior nie uporali się z tym problemem (D-100 przyjęto jako standardową drukarkę dla tego komputera); w komputerze tym drukarka działa poprawnie tylko z poziomu języka BASIC.

Po niedawnej zmianie komputera na „klon” PC starałem się odnaleźć oprogramowanie pozwalające mi na kontynuację pracy rozpoczętej na Atari 800XL. Zaopatrzyłem się w Print Shop, Lattice C, Turbo Pascal — pierwszy z tych programów to praktycznie to samo, pozostałe dwa kompilatory różnią się co prawda znacznie w obsłudze od ich odpowiedników na Atari — ale też, jak się zdołałem zorientować, możliwości programisty są też dużo większe. Dobre gry dla PC jest znacznie trudniej znaleźć niż to jest w przypadku Atari. Ale moje pytanie dotyczy jeszcze jednego języka programowania. Próbowałem swoich sił w Polskim Logo na Atari. Obecnie znajomy udostępnił mi Dr Logo dla IBM PC. Mam opis tego języka zamieszczony w „Bajtku” (3/88). Opis ten zawiera jednak tylko różnice pomiędzy Dr Logo i Logo dla Amstrada. Jak się przedstawiają różnice Dr Logo i Logo dla Atari 800XL? Czy dostępna jest polska wersja Dr Logo?

Michał Rachoń
(adres do wiadomości redakcji)

Trudność w zdobyciu dobrych gier dla komputerów PC wynika z profesjonalnego charakteru samego komputera, w związku z czym gry rozpowszechniają się znacznie wolniej od programów użytkowych i narzędziowych. Obecnie sytuacja ta ulega zmianie w związku ze znacznym wzrostem zakupów PC przez indywidualnych nabywców.

Jak dotąd nic nie słychać o spolszczonej wersji Dr Logo dla IBM PC, jakkolwiek przeróbki tej dokonano w przypadku Dr Logo dla komputerów Amstrad.

Dr Logo jest w ogólności językiem znacznie bardziej rozbudowanym od Atari Logo. Lista procedur zamieszczonych w artykule „Doktor! nareszcie z dyplomem?” pozostaje w zasadzie także listą rozszerzeń Dr Logo w stosunku do Logo na Atari. Pełną listę procedur Dr Logo wraz z objaśnieniami można uzyskać wpisując HELP jako pierwszą komendę po uruchomieniu interpretera.

„Przesiadka” na Dr Logo nie wiąże się jednak z samymi korzyściami. Dr Logo dysponuje tylko jednym żółciem; nie można definiować jego kształtu ani nadawać mu stałej prędkości. Powoduje to, że programowanie animacji jest w Dr Logo znacznie bardziej utrudnione.

Czy system operacyjny GEOS może być uruchomiony również na starej wersji C-64? Jakie programy użytkowe oprócz edytora grafiki i graficznego edytora tekstów działają w tym systemie?

Krzysztof Kobzej
Sopot

GEOS działa bez zastrzeżeń dla wszystkich wersji C-64. Biblioteka programów użytkowych dla tego systemu stale wzrasta — co jest usprawiedliwione jego wartością dla użytkownika. M.in. opracowano GEOPUBLISH — program z serii DTP, oczywiście znacznie uproszczony w stosunku do standardu Ventura Publisher, niemniej dość bogaty jak na możliwości C64; GEOCALC — arkusz kalkulacyjny; GEOFILLE — program obsługi bazy danych. Szerszy opis i dłuższą listę programów znaleźć można w numerze specjalnym „Bajtek” — tylko o Commodore”. Zachęcam do lektury.

Posiadam mikrokomputer Atari 130XE. Przeglądając „Bajtki” i „Komputer” nie znalazłem odpowiedzi na następujące pytania:

1. Który z translatorów: C, Pascal, Action! jest najszybszy (domyślam się, że Action!, ale chodzi mi głównie o dwa pierwsze);
2. Czy po skompilowaniu szybkość wykonywania programu napisanego w tych językach jest taka sama (jeżeli nie, to w którym najszybciej się odbywa)?
3. Który z tych języków jest najbardziej przydatny w pisaniu gier?
4. Czy języki te dostępne są na cartridge’u?

Marcin Wojtasik
os. Sikorskiego 3/18
28-100 Busko Zdrój

Nie ma większego sensu porównywanie prędkości generowania kodu wynikowego w przypadku wymienionych kompilatorów dla Atari. Nie ma ona bowiem większego wpływu na tempo pracy z tymi programami, skoro np. Kyan Pascal wymaga przy kompilacji zamiany dyskietki, zaś Action! na cartridge’u — oczywiście nie. Action! jest jedynym z tych kompilatorów wydanym na cartridge’u, i w tej wersji praca z nim jest niewątpliwie najefektywniejsza.

Do pomiaru szybkości realizacji programów wynikowych opracowano szereg standardowych testów tzw. Benchmarks, pozwalających na orientacyjne porównanie prędkości realizacji wybranych instrukcji. Większość tych tekstów dotyczy jednak programów nie wykonujących operacji graficznych, które zajmują lwią część czasu w programach gier. Proponuję przeprowadzić proste testy we własnym zakresie, uwzględniające kluczowe obliczenia i operacje graficzne planowanej gry. Osobiście stawiam na Action!, który pod względem szybkości sprawia korzystne wrażenie, a poza tym został zaprojektowany specjalnie dla Atari i zapewni dobry dostęp do aparatu grafiki tego komputera.

Przy świadomym i rozważnym wykorzystaniu C daje również możliwość tworzenia szybkich programów, zwłaszcza w porównaniu z Pascal-em. Odbywa się to jednak moim zdaniem ze szkodą dla czytelności programów i dlatego nie polecam C, o ile nie jest Pan doświadczonym programistą.

Uważam się za programistę „lepszego z tych przeciętnych” — z reguły jestem w stanie samodzielnie rozwiązać wszystkie swoje potrzeby w dziedzinie oprogramowania „na miarę”. W codziennej praktyce (komputer klasy XT z dyskiem twardym 20MB i kartą Hercules)

używam Turbo Pascal-a 3.0. Wprawdzie ostatnio pojawiło się kilka podobno lepszych języków programowania, w tym nowy Turbo Pascal, ja jednak jestem urodzonym konserwatystą i nie chciałbym zmieniać języka programowania, jeżeli nie będę do tego zmuszony. Jako argument służy mi fakt istnienia, bogatych bibliotek podprogramów dla wymienionego Pascala, głównie chodzi mi o Turbo Graphix Toolbox, które dobrze znam i z przyjemnością używam. A’propos — czy nie warto by pakietów z serii Toolbox rozpropagować na łamach „Bajtki”? Byłoby to sensowną kontynuacją artykułów poświęconych Turbo-Pascalowi.

Mam jeszcze pytanie istotniejsze dla mnie od ciekawości o plany wydawnicze redakcji. Dotyczy ono właśnie Turbo Graphix Toolbox. Pakiet ten umożliwia mianowicie tworzenie grafiki zarówno przy użyciu karty Hercules, jak i CGA. Zmiana typu grafiki odbywa się jednak na poziomie programu źródłowego — poprzez wymianę części programu zawartej na pliku GRAPHIX.SYS. Uniemożliwia to stworzenie programu, który sam rozpoznawałby rodzaj karty graficznej i do niej się dostosowywał. Czy redakcji znana jest jakaś metoda zlikwidowania tej niedogodności? Przeszkadza ona w wykonaniu oprogramowania działające na różnych egzemplarzach „klonów”.

(nazwisko i adres do wiadomości redakcji)

Rozwiązanie jest możliwe nawet z poziomu języka źródłowego, wymaga jednak dokonania dużych zmian w pakiecie Turbo Graphix Toolbox. Najlepiej więc zrobić jego roboczą kopię, a następnie na niej dokonać niezbędnych zmian.

Jest oczywiste, że do programu należy włączyć obie wersje pliku GRAPHIX.SYS, tzn. GRAPHIX.IBM i GRAPHIX.HGO. W tym celu należy nadać procedurę zawartą w tych plikach różne nazwy — choćby przez dodanie do nich przedrostków lbn i Hgc. Teraz jednak zdezaktualizowały się wszystkie późniejsze odwołania do tych procedur. Każde z później występujących wywołań zastępujemy więc wywołaniem warunkowym, uzależnionym od wartości zmiennej logicznej Typ Karty. Zmienną tą zadeklarujemy w pliku TYPEDEF.SYS.

Pozostaje jeszcze problem rozpoznania, z jaką kartą mamy do czynienia i nadanie wartości zmiennej TypKarty. Czynimy to przez modyfikację procedury InItGraphic. Nowe funkcje boolowskie lbnHardwarePresent i HgcHardwarePresent pozwalają na jednoznaczne określenie, z którą kartą mamy do czynienia.

Jest również znacznie prostsze rozwiązanie problemu, które jednak ciągnie za sobą stratę przewagi karty Hercules nad CGA, jaką jest rozdzielczość. Należy mianowicie przed uruchomieniem programu przełączyć kartę Hercules w tryb emulacji CGA. Można tego dokonać poprzez utworzenie pliku BAT i użycie w nim programu COLOR.

Innym wyjściem z sytuacji jest użycie języka Turbo Pascal 4.0, który standardowo pozwala na użycie grafiki z rozróżnieniem karty graficznej. Firma Borland zapowiedziała wydanie pakietu Turbo Graphix Toolbox również w wersji dla Turbo 4.0 (w zwykłej postaci nie daje się on zastosować w tym języku).

Planujemy zamieszczenie materiałów poświęconych Turbo Graphix Toolbox już w niedalekiej przyszłości.

Marcin Waligórski

Jestem posiadaczem mikrokomputera Timex 2048 i chciałbym go wykorzystać do współpracy z drukarką D-100. Problem tkwi w tym jak to uczynić? Prosiłbym o poradę praktyczną, o rodzaj interfejsu, który można byłoby wykorzystać do współpracy oraz sposobu jego podłączenia. Czy można do tej współpracy wykorzystać wyjście manipulatora?

(nazwisko i adres do wiadomości redakcji)

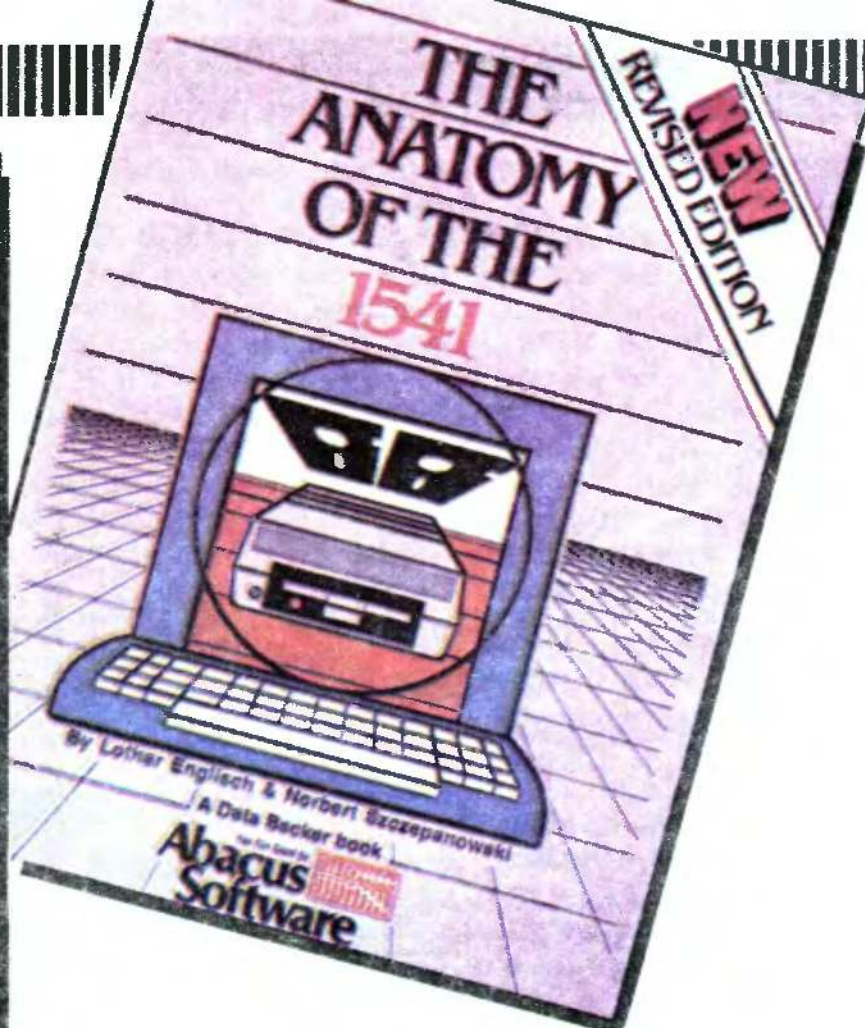
Drukarka D-100 powinna być obsługiwana przez protokół transmisji standardu Centronics i tego typu interfejs jest potrzebny. Na rynku znaleźć można przynajmniej kilka typów takich interfejsów. Wszystkie one są podłączane do szyny krawędziowej komputera — podłączenie przez gniazdo joysticka nie jest możliwe chociażby z tego względu, że w przypadku łącza Centronics zachodzi transmisja równoległa.

Kłopotem w przypadku drukarki D-



Redakcja BAJTKA otrzymała do recenzji od firmy DATA BECKER dwie książki, które zainteresują zapewne właścicieli stacji dysków 1541 i 1571. Są to „1571 INTERNALS” oraz „ANATOMY OF THE 1541 DISK DRIVE” (wersja poprawiona i uzupełniona). Obydwie publikacje są adresowane zarówno do początkujących jak też zaawansowanych użytkowników tych stacji.

„1571 INTERNALS” tak samo zresztą jak i druga książka to licząca prawie 500 stron mapa pamięci i kompendium wiedzy o stosunkowo nowym — jeszcze produkcie Commodore — stacji 1571. Autor, Rainer Ellinger, zaczyna swe omówienie od kompletnych podstaw — wyglądu urządzenia i przeznaczenia poszczególnych gniazd, przełączników, struktury dyskietki, definicji kanału transmisyjnego itp. Po „galkologii” następuje wprowadzenie do BASIC i poleceń odnoszących się do sterowania pracą stacji dysków — wymienione są tu wszystkie odnośne instrukcje wraz ze składnią i przykładami. Dużą pomocą jest tu formuła przyjęta przez autora iż wszystkie polecenia są omawiane w odniesieniu do wersji BASIC 3,0 (a więc C-64, VIC-20) oraz BASIC 3,0 (rodzina C-16/



C-116 i PLUS/4, C-128/128D). Duża ilość przykładów ułatwia zapoznanie się z poszczególnymi poleceniami.

Dwie odrębne części rozdziału i poświęcone są tworzeniu zbiorów sekwencyjnych i zbiorów o bezpośrednim dostępie (typu REL) i zawierają omówienie podstawowych zasad korzystania z tych plików danych. Również tu znajdują się przykłady, co prawda nieco bardziej skąpe aniżeli w części poświęconej BASIC. Z następnych rozdziałów dowiadujemy się o rozkazach bezpośrednich (np. BLOCK-READ czy BUFFER-POINTER), strukturze ścieżki systemowej, organizacji zbiorów; wszystko to jest opisane językiem przystępnym, dość łatwym do przyswojenia.

Od rozdziału 3 zaczynają się wiadomości dla zaawansowanych — wczytywanie i wykonywanie programów w pamięci RAM stacji, tworzenie programów samouruchamiających się (zawierających znak „&” w nazwie zbioru), systemu CP/M, BIOS, o formacie MFM i GCR i wiele innych. Dodatkowo autor wspomina wielokrotnie o stacji 1570 tak więc i dla właścicieli tego modelu książka ta będzie pożyteczna.

Trzystustronicowy załącznik A zawiera dokładny wydruk ROM stacji 1571 wraz z komentarzami. Aby jednak użytkownik mógł sam zbadać ewentualne różnice w nowszych wersjach podano odpowiedni program pozwalają

jący na przepisanie ROM stacji do RAM komputera (C-128). Dodatkowo w książce tej znalazły się także inne, prawdziwe perełki: lista błędów w DOS, różnice w ROM i DOS pomiędzy stacjami 1570 i 1571, omówienie organizacji strony zerowej.

Osobiście uważam tę książkę za bardzo przydatną i bardzo pożyteczną, zwłaszcza wtedy gdy użytkownik chciałby wykorzystać wszystkie walory tej stacji co nie zawsze jest możliwe w BASIC (np. tryb pracy „burst”). Jest to moim zdaniem przewodnik o rzeczywiście dużej klasie.

W bardzo podobny sposób jest zredagowana druga książka — ANATOMY OF 1541 DISK DRIVE, której autorami są Lothar Englisch oraz Norbert Szczepanowski. Dla początkujących przeznaczono tu 122 strony na których omówiono „galkologię”, programowanie z poziomu BASIC oraz tworzenie zbiorów. W rozdziale 3 przedstawiono działanie stacji z punktu widzenia elektronicznego, współdziałania poszczególnych układów itp. Lwią część książki zajmuje natomiast (tak jak i w poprzedniej) wydruk ROM stacji wraz z komentarzami.

Szczegółowej uwadze Czytelników polecałbym rozdział 4. Zawarto w nim sporo cennych wskazówek dotyczących programowania stacji 1541 oraz wiele programów i sztuczek (blokowanie wyświetlania katalogu, zabezpieczenie dyskietki przed zapisem, zmiana identyfikatora, łączenie programów, monitor stacji dysków itp.). Omówiono tu także dokładnie wszystkie programy znajdujące się na dyskietce demonstracyjnej.

Obydwie książki łączą się ze sobą w pewien sposób — są one zredagowane z myślą o przekazaniu jak największej ilości pożytecznej i cennej informacji. Informacja ta poparta jest wieloma przykładami co ułatwia przyswojenie sporej dawki wiedzy zawartej w tych publikacjach. Dokładnie i solidnie przedstawiony materiał o dużej użyteczności praktycznej oraz dobre ujęcie tematu sprawiają, że książki te będą cenną lekturą podczas sesji programowania.

Klaudiusz Dybowski

1571 INTERNALS
wydana w 1985 r.
przez DATA BECKER GmbH
Merowingerstr. 30
4000 Dusseldorf, WEST GERMANY
język niemiecki lub angielski
stron. ok. 500.

THE ANATOMY OF 1541 DISK DRIVE
wydana w 1985 r.
przez DATA BECKER GmbH
adres jak wyżej i języki
stron 500.

Przedsiębiorstwo Zagraniczne KAREN

ul. Obrońców 23,
03-933 Warszawa
tel. 17 84 10
tlx 813948 kren pl

Szanowny Panie Dyrektorze,

Dziękujemy za zainteresowanie naszą firmą.

Z przyjemnością informujemy, że możemy zaspokoić wszystkie potrzeby Pana Przedsiębiorstwa określone w skierowanym do nas zapytaniu.

- 1. Oferujemy niezawodne i jednolite systemy komputerowe typu PC/XT/AT/386.*
- 2. Instalujemy adaptery i oprogramowanie sieciowe ETHERNET.*
- 3. Do Zakładu Poligrafii polecamy zestaw ATARI ST DESKTOP PUBLISHING bogato oprogramowany i oczywiście z polskimi literami.*
- 4. Do Klubu i Szkoły proponujemy ośmiobitowe ATARI XE.*

Proszę nie niepokoić się o "wsad dewizowy" - to wszystko jest za złotówki. Sprzęt objęty jest roczną gwarancją a przy odbiorze będzie mógł Pan uzupełnić swoje zbiory oprogramowania i literatury.

Z poważaniem,

DZIAŁ HANDLOWY

ZEGAR

Cześć Maluchy!

Nie ma chyba nic bardziej nudnego, niż ciągle zajmowanie się rzeczmi mądrymi i użytecznymi. Dlatego czasem, po prostu muszę zrobić coś najzupełniej niepotrzebnego.

Od razu więc zastrzegam, że nie odpowiadam na pytania, czy komukolwiek może się do czegośkolwiek przydać zegar napędzany przez żółwia LOGO. Pewnego dnia przyglądając się temu sympatycznemu gadowi (żółwie to gady, pamiętam ze szkoły) stwierdziłem, że ma on kształt podobny do grota strzały lub wskazówki. Wskazówka skojarzyła mi się z zegarem i... tak powstał ten program.

Rozpocząłem od narysowania tarczy, która ma kształt okręgu. Jak rysuje się w LOGO okrąg, wiedzą już chyba wszyscy. Po prostu rysuje się wielobok o tak dużej liczbie boków, że na ekranie wygląda on zupełnie jak okrąg. Warto zwrócić uwagę na zależność pomiędzy liczbą boków a kątem o jaki za każdym razem obraca się żółw. Podczas rysowania dowolnego wieloboku żółw wykonuje w sumie pełen obrót, czyli 360 stopni. Na jeden bok przypada więc $360/n$ stopni, gdzie n jest liczbą boków. Ponieważ założyłem, że mój wielobok ma mieć 180 boków, to wynika stąd, że kąt obrotu żółwia wynosi $360/180=2$ stopnie.

Swoją drogą, z punktu widzenia żółwia wszystkie twierdzenia o sumie boków wielokąta wydają się oczywiście.

Rysując okrąg za pomocą cyrkla bardzo łatwo możesz ustalić położenie jego środka i długość promienia. Rysując okrąg przy pomocy żółwia LOGO nie znasz tych parametrów, więc musisz je obliczyć. Znasz przybliżoną wartość długości okręgu — jest to suma długości wszystkich boków.

Długość okręgu (albo obwód koła — co na jedno wychodzi) można obliczyć ze wzoru:

$$o = 2\pi r$$

gdzie: o — obwód koła, długość okręgu

r — promień

π — liczba pi, równa w przybliżeniu 3,14

Jeśli więc chcemy narysować okrąg o promieniu 100 żółwich kroków, to jego obwód wyniesie

$$2 * 3,14 * 100 = 628$$

Na jeden bok wieloboku przypadnie

$$628/180 = 3,48(8)$$

a więc w przybliżeniu 3,5.

Dzięki temu, że znamy promień naszego okręgu, możemy ustalić położenie jego środka. Najwygodniej będzie umieścić go w środkowym punkcie ekranu, do którego żółw powraca po komendzie home.

Oto więc pierwsza procedura (zakładamy, że żółw znajduje się w pozycji startowej):

to tarcza

```
pu
fd 100
pd
rt 90
repeat 180 [rt 2 fd 3.5]
pu
home
end
```

Żółw podnosi pióro (pu) i posuwa się o długość promienia (sto kroków) do przodu (fd 100). Następnie opuszcza pióro (pd), obraca się o kąt prosty w prawo (rt 90) i zaczyna rysować okrąg o parametrach, które obliczyliśmy wcześniej. Po zakończeniu okręgu, ponownie podnosi pióro i wraca do punktu startowego (home).

Na tarczy powinny być zaznaczone cyfry, albo przynajmniej kreski na ich miejscach. Poprośmy żółwia aby je wykonał:

to cyfry

```
repeat 12 [pu fd 85 pd fd 10
pu bk 95 rt 30]
end
```

czyli żółw na karuzeli

Żółw powtarza 12 razy (tyle właśnie godzin znajduje się na tarczy zegara) te same czynności: podniesienie pióra (pu), marsz 85 kroków do przodu (fd 85) opuszczenie pióra (pd) i 10 kroków do przodu — w ten sposób powstaje kreska. Następnie żółw znowu podnosi pióro i cofa się o 95 kroków (bk 95). Teraz tylko obrót o 30 stopni (rt 30) i to samo od nowa.

Wskazówki narysować bardzo łatwo. Kłopot w tym, żeby jeszcze wskazywały właściwą godzinę i minutę. Wprowadzimy sobie dwie zmienne dla określenia czasu, właśnie minuta i godzina. Nadajmy im jakieś wartości początkowe, np.:

```
make "minuta 20
```

```
make "godzina 3
```

Duża wskazówka w ciągu sześćdziesięciu minut obraca się o 360 stopni, a więc na jedną minutę przypada $360/60=6$ stopni. Mała wskazówka w ciągu godziny wykonuje obrót o kąt 30 stopni, a w ciągu minuty o $30/60=1/2$ stopnia. Znając te wartości możemy już nauczyć żółwia rysowania wskazówek:

to wskazówki

```
home
rt :minuta * 6
fd 80
home
rt :godzina * 30 + :minuta *
30 / 60
fd 60
home
end
```

Żółw ustawia się w pozycji startowej i obraca o kąt równy liczbie minut pomnożonej przez 6. Następnie wędruje 80 kroków do przodu rysując dużą wskazówkę i powraca do pozycji startowej. Teraz obraca się dwukrotnie. Najpierw o wartość liczby godzin pomnożonej przez 30 stopni a potem liczby minut (po godzinie) pomnożonej o 1/2 stopnia. Małą wskazówkę rysuje posuwając się 60 kroków do przodu i wraca do pozycji wyjściowej.

Kolejne zadanie, to nauczyć żółwia jak ma udawać sekundnik:

to sekundnik

```
pu
home
ht
repeat 59 [fd 80 st repeat
200 [] ht bk 80 rt 6]
rt 6
end
```

Najpierw podnosi on pióro, ustawia się jak zwykle w pozycji startowej i znika z ekranu (ht), 59 razy powtarza następujące czynności: maszeruje 80 kroków do przodu i pojawia się (st), teraz następuje pętla opóźniająca repeat 200 [], zniknięcie żółwia, cofnięcie się o 80 kroków i obrót o 6 stopni w prawo. Liczba pętli opóźniających musi być tak dobrana, aby jeden przeskok sekundnika trwał rzeczywiście sekundę. Na moim komputerze (Amstrad 6128) najdokładniejszy zegar otrzymałem przy 203 obiegach pętli.

Zdziwiło Was zapewne, że sekundnik odmierza 59 o nie 60 sekund. Otóż ta jedna sekunda zużyta zostanie

na rysowanie co minutę nowych pozycji wskazówek: godzinowej i minutowej.

Chodzący zegar wygląda więc tak:

to tiktak

```
pd
wskazówki
sekundnik
pe
wskazówki
make "minuta :minuta + 1
if :minuta = 60 [make "minuta
0 make "godzina :godzina + 1]
if :godzina = 12 [make
"godzina 0]
tiktak
end
```

Żółw opuszcza pióro, rysuje wskazówki i jako sekundnik odlicza 59 sekund. Następnie zmienia pióro na gumkę do wycierania (pe) i za jej pomocą ponownie „rysuje” wskazówki, w efekcie czego znikają one z ekranu. Teraz może już zwiększyć o 1 wartość zmiennej minuta. Jeśli zmienna ta osiągnie wartość 60, wówczas przyjmie na powrót wartość zero a zmienna godzina zostanie zwiększona o 1. Jeżeli ta ostatnia osiągnie wartość 12, też powróci do 0.

Ostatnim rozkazem tej procedury jest wywołanie jej samej. Oznacza to, że trzeba rozpocząć jej wykonywanie od nowa. Taką konstrukcję nazywamy rekurencją. Polega ona na wykorzystywaniu w treści definicji definiowanego pojęcia.

Teraz wystarczy poskładać wszystkie klocki w jedną całość i nasz zegar jest gotowy.

to zegar

```
cs
tarcza
cyfry
pr [godzina?]
make "godzina rq
pr [minuta?]
make "minuta rq
pd
tiktak
end
```

Aby ustawić właściwą godzinę, należy nadać początkowe wartości zmiennym: godzina i minuta. Robimy to przy pomocy instrukcji:

```
make "godzina rq
```

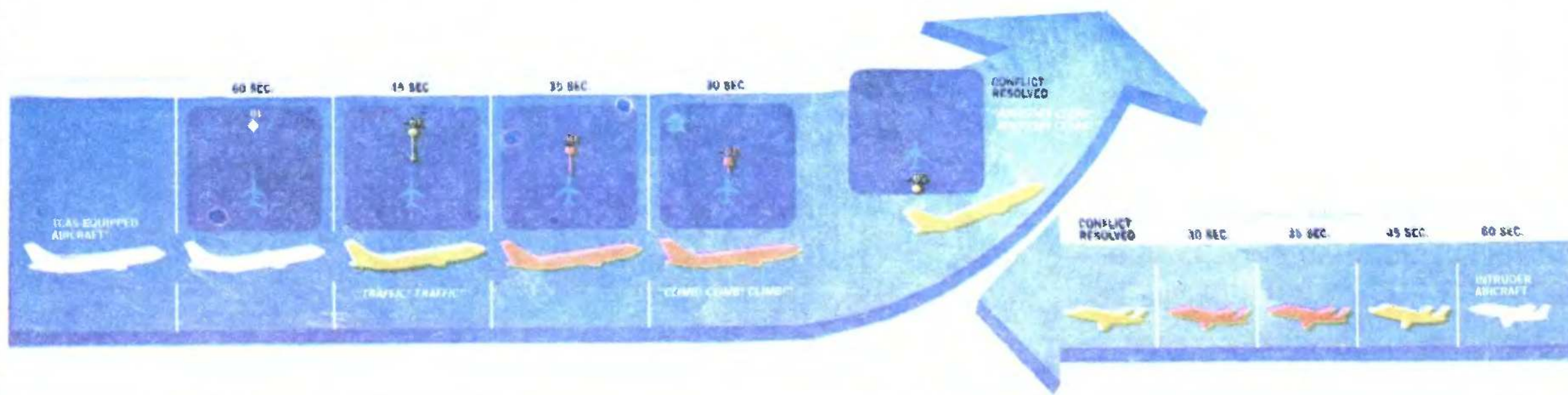
```
make "minuta rq
```

działają one dokładnie tak jak BASIC-owe INPUT. Czekają na wprowadzenie wartości i potwierdzenie klawiszem RETURN/ENTER.

* * *

Zegar chodzi, żółw uwija się jak w ukropie a wy spróbujcie, może znajdziecie jakieś rozsądne zastosowanie takiego czasomierza. Jeśli Wam się to uda, najbarziej zdziwioną osobą będzie z pewnością... Wasz

Romek



KOMPUTEROWY ANIOŁ STRÓŻ

Kurczaki — tak nazywano tę grę w czasach mojego dzieciństwa: dwóch kierowców prowadziło swoje samochody pędzące naprzeciwko siebie. Pierwszy, który zrefterował skręcając w bok przegrywał; jeśli skręcili obaj, stawka przechodziła na następną parę ryzykantów.

Dzisiaj gra jest inna. 12 tysięcy stóp nad kasynami gry w Atlantic City lecieliśmy naprzeciwko siebie dwoma samolotami: odrzutowym Boeingiem 727 i turbosmigłowcem Convair 580, testując TCAS — elektroniczny system ostrzegający przed kolizją. Maksymalną prędkość ustalono na 650 km/h. Siedziałem w kokpicie Boeinga w fotelu odrzutowym przygotowany na najgorsze, tj. katapultę; gdyby gra przestała być zabawą. Obie leżące naprzeciwko siebie załogi zdały się bowiem w tej grze na decyzję komputera — ostatniego ognia w testowanym systemie unikania kolizji.

— Convair powinien być akurat przed nami — stwierdził John Ryan, pierwszy pilot Boeinga, po rozmowie radiowej z wieżą. Spięty obserwowałem uważnie zachmurzone niebo, nie dostrzegając jednak niczego. Kiedy wyraziłem swoje uwagi o ewentualnym błędzie, pilot odwrócił się do mnie mówiąc z uśmiechem:

— Bez obawy, nawet gdyby ktoś tam błędnie odczytał wskazania wysokościomierza, przeleciał 100-200 metrów nad nami.

Znow zacząłem uważnie przeszukiwać pusty horyzont rzucając okiem na pusty jeszcze ekran monitora, umieszczony centralnie przed pilotem.

Za chwilę pojawił się na nim mały, biały romb (patrz rys. 1)

— To właśnie jest Convair — poinformowali mnie piloci. Z położenia rombu — w górnej części ekranu (rys. 1) wynikało, że „obcy” samolot znajduje się w odległości 5 mil od Boeinga.

A więc zaczęło się — miałem nadzieję, że system w pierwszej próbie nie zawiedzie.

— **TRAFFIC. TRAFFIC. TRAFFIC.** — (Tfoki!) nagle odezwał się z głośnika nad konsolą z monitorem matowy, syntetyzowany głos komputera.

Nadal nic nie widziałem na bezpiecznym, zdawało się, niebie. I wtedy Ryan, odwracając się, rzucił przez ramię w moim kierunku:

— Spójrz, symbol „intruza” zmienił się na ekranie w żółtą kulkę!

Znaczyło to, że radarowy sygnał z Convaira otworzył naszą „kopertę bezpieczeństwa” i TCAS uruchomił komputerowy system analizujący sytuację pod kątem wyboru najlepszego wariantu uniknięcia kolizji.

Mijały powoli sekundy; w końcu inżynier pokładowy Lou De-Stefano spostrzegł na horyzoncie małą czarną plamkę, która zbliżając się, początkowo wolno, z każdą chwilą rostała w oczach. Minęło jeszcze kilka sekund i żółta plamka na ekranie monitora zamieniła się w pulsujący czerwony prostokąt. W tym samym czasie czarny punkt, leżąc na kursie „727”, zbliżał się już jak pocisk do celu.

Nagle głośnik wyrzucił ostrym tonem mechanicznie ostrzeżenie:

— **CLIMB. CLIMB. CLIMB.** (W górę!)

Ryan lekko przesunął drążek do siebie i olbrzymi nos Boeinga podniósł się w górę. W trakcie manewru dotychczasowa plamka na kursie przeszła w ryczący silnikami Convair, który przeleciał pod nami w niewielkiej, ale już bezpiecznej odległości.

— Advisory clear — ciepłym, przy najmniej tak mi się wydawało, głosem zaanonsował w chwilę później komputer.

Poddawany obecnie testom TCAS (elektroniczny system antykolizyjny) przeszedł już długą koncepcyjną i konstrukcyjną drogę w kierunku eliminacji niebezpieczeństwa kolizji statków powietrznych.

Jego powstanie i skonstruowanie nie było pomysłem nowym; prace nad konstrukcją sprawnego systemu zapobiegającego coraz częstszym przypadkom prawie-kolizji w powietrzu zaczęły się już ponad 30 lat temu. Z roku na rok problem ten nabierał jednak znaczenia, zmuszając, zwłaszcza w ostatnich latach, do zwielokrotnienia wysiłków. Łatwo zrozumieć dlaczego, jeśli tylko w ubiegłym roku piloci amerykańscy raportowali o 1056 prawdopodobnych kolizjach w przestrzeni powietrznej USA. W każdej z nich brały udział przynajmniej dwa samoloty; a o ilu nie wspomniano w raportach? Mówi się o przynajmniej raz tylu przypadkach.

To jeśli chodzi o liczby i statystyki

— które łatwo przeliczyć na żywą gotówkę, miliony dolarów nie tylko bezpośredniej straty w przypadku katastrofy, ale daleko większe konsekwencje w postaci zmniejszenia zaufania do samolotu, linii czy danego towarzystwa lotniczego. Stąd naciski i zwiększone fundusze na badania i testy, do których włączone zostały również programy rządowe.

TCAS pojawi się na rynku w trzech konfiguracjach. Wydaje się, że najwcześniej zakończone zostaną prace nad wersją TCAS II, której model laboratoryjny przeszedł już testy praktyczne (m.in. w opisywanym locie); produkcja seryjna rozpocznie się za kilka miesięcy. System TCAS II ostrzega pilotów maszyn znajdujących się na kursie kolizyjnym dyktując im manewr ucieczki: w górę lub w dół (jednemu z nich) lub ostrzega przed pojsciem w górę lub w dół (drugiego). Ulepszona wersja tego systemu — TCAS III — powinna znaleźć się na pokładach samolotów w kilka lat później. Z prób w laboratorium wynika, że najistotniejsze wzbogacenie tego systemu polegać będzie na zwiększeniu „koncepcyjnych” możliwości komputera, który w razie prawdopodobieństwa kolizji dyktował będzie pilotom prócz zmiany kursu — w

górę lub w dół — jeszcze skręt w prawo lub w lewo.

TCAS I zaś jest (będzie) najprostszym (i najtańszym) urządzeniem tej rodziny systemów antykolizyjnych, którego zadanie polega tylko na ostrzeżeniu pilota w przypadku niebezpiecznego „tłoku” w powietrzu.

W miarę zaawansowania prac nad systemami TCAS ich autorzy musieli odpowiadać na pytania, które tylko w części dotyczyły technicznej strony przedsięwzięcia. Żądano od nich na przykład określenia niezawodnej pracy systemu, pytano czy jest ona wystarczająca do powszechnego instalowania jej w kokpitach. Kiedy? Za jaką cenę? Czy wszystkie statki powietrzne powinny obowiązywać jego zainstalowanie? etc.

Nie na wszystkie z tych pytań znaleziono już odpowiedzi. Niektórzy z ekspertów są sceptyczni co do możliwości rozładowania przez TCAS II tłoku nad lotniskami np. w Los Angeles, gdzie zdarza się, że 25 samolotów czeka na lądowanie w kręgu o 5-milowym promieniu. Dlatego też niektóre zarządy towarzystw lotniczych wątpią, czy decyzja władz w kwestii instalowania systemu we wszystkich typach samolotów będzie tak jednoznaczna, jak tego chcą autorzy rozwiązania.

Mimo dyskusji jednak i wątpliwości pewnym jest fakt, że zastosowanie TCAS dodatkowo ubezpieczy podróż samolotami; może także zmniejszyć opóźnienia samolotów, którym kontrolerzy lotniczy pozwolą zmniejszyć bezpieczną odległość. Jednym słowem, system ten przyspieszy lotniczą karuzelę, biorąc odpowiedzialność za jej bezpieczeństwo. Jedynym ogniwem, które w tym systemie może zawieść jest człowiek — pilot wykonujący polecenie komputera.

na podst. Popular Science

opr. F.P.



PLAMKA NA KURSIE PRZESZŁA W RYCZĄCY SILNIKAMI CONVAIR