

11
Z MIKROKOMPUTEREM NA TY

NR INDEKSU 353965
PL ISSN 0860-1674

Bajtek

MIESIĘCZNIK SZTANDARU MŁODZIEŻY R 11(47) LISTOPAD 1989 CENA 1000 ZŁ



W JOYSTICKLANDZIE

NAJLEPSZY

DOS

**PREZENTY
I RÓZGI**

**DOSTAŁEM
AMIGĘ!**

**MODYFIKACJA
TASWORDA**

**KONKURS
ŚWIĄTECZNY!**

TETRIS

● ZNAK FIRMOWY ●

Na początku listopada otwarty został w Krakowie drugi w Polsce sklep firmowy „Bajtki” (ul. Pstrowskiego 9, Klub Sportowy Korona). Będzie można w nim nabyć aktualne i archiwalne numery naszego miesięcznika, ale przede wszystkim sprzęt komputerowy. Sześć sklepu, Stefan Opalski, chciałby, żeby był to sprzęt dla naszych Czytelników — to znaczy tani i dobry. Zobaczymy na ile się to Stefanowi uda. Bo gdyby się nie udało — to musielibyśmy przemyśleć w redakcyjnym gronie, czy w dalszym ciągu ten salon firmować. Sądziemy, oczywiście, że do tego nie dojdzie i że kultura obsługi i jakość proponowanego towaru będą na miarę naszych oczekiwań.

Miesiąc temu postawiliśmy ultimatum przed organizatorami giełdy „Bajtki” w Warszawie. Wyjaśniliśmy dokładnie co nas zmusiło do tego kroku. Mówiąc najkrócej: nie chcemy i nie będziemy firmować podejrzanych interesów, na które przysmykają oko organizatorzy giełdy i nie będziemy tolerować braku elementarnej kultury w szkole przy ul. Grzybowskiej. Wprawdzie termin ultimatum jeszcze nie minął, ale niepokoi nas brak postulowanych zmian...

Może się ktoś zapytać, skąd ta nasza troska o wysoką jakość wszystkiego, co wiąże się z hasłem „Bajtek”? Nie jest to, otóż, nic nowego — chcemy po prostu, aby, tak jak to było do tej pory, nasz „bajtkowy” znak firmowy był jednocześnie znakiem najwyższej jakości. Uważamy, że jest to szczególnie ważne w chwili obecnej, gdy powszechny zalew tandety (sprzedawanej za duże pieniądze!) zaczyna stanowić coraz większe zagrożenie dla przyszłości naszego kraju.

Nie jest to oczywiście zagrożenie jedyne. Wspólnie odczuwamy dotkliwie gonitwę cen, której ofiarą pada również nasz miesięcznik. Broniliśmy się i bronimy nadal przed zbyt wielkimi podwyżkami ceny, ale istnieje przecież granica ekonomicznej opłacalności, poniżej której nie możemy zejść. Optymiści redakcyjni przeliczają cenę jednego numeru „Bajtki” na chleb czy też dekagramy masła — i wchodzi im, że stajemy się relatywnie tańsi, niż przed rokiem czy dwoma. Ale realności takich porównań nie przyjmują. Skoro bowiem ludzie rezygnują z masła — argumentują — to tym łatwiej rezygnują z prasy...

Sytuacja jest trudna i chemy to bez ogródek powiedzieć. Ponieważ stale rosną ceny papieru, a w ślad za nimi również ceny usług drukarskich — więc musi również rosnąć cena „Bajtki”. Oczywiście nie tak

szybko, jak ceny papieru, ale jednak... Co możemy w tej sytuacji zrobić?

Możemy przekonywać wydawcę, że pismo takie jak „Bajtek” nie powinno być traktowane jako „maszynka do robienia pieniędzy”, co oznacza w praktyce, że tak należy kalkulować jego cenę, aby przynosił minimalny zaledwie zysk.

Założmy, że nasze argumenty o edukacyjnej i cywilizacyjnej roli „Bajtki” przekonają wydawcę, i że świadomie zrezygnuje on z pokusy zbijania milionów na „bajtkowcach”. Ale przecież nie zatrzyma to wzrostu ceny, tylko spowoduje zmniejszenie tempa tego procesu. Jest jednak zagrożenie innego rodzaju, na które trudniej znaleźć antidotum. Mam na myśli różnorodne akcje i konkursy organizowane przez nasze pismo.

Otrzymujemy na przykład pytania, dlaczego tak długo trwa rozstrzygnięcie Konkursu Klubów Komputerowych „Złotą Dyskietkę „BAJTKA”? Otóż dlatego właśnie, że naszego wydawcy nie stać już na fundowanie nagród, a sponsorzy z pierwszej edycji naszego konkursu też jakby wykryli dziury w swych kieszeniach. Taka jest prawda — czy się to nam podoba, czy nie. To samo dotyczy różnych atrakcyjnych dla naszych Czytelników pomysłów współpracy międzynarodowej. Umówiliśmy się otóż niedawno z partnerami ze Związku Radzieckiego i FRN, że będziemy wspólnie organizować wymienne kursy uzdolnionych informatycznie uczniów. Ale też naszego wydawcy już na to nie stać...

Wszyscy się zgadzają, owszem, że upowszechnianie komputeryzacji wśród młodzieży to sprawa bardzo ważna, ale nie idą za tymi deklaracjami wymierne w złotówkach czyny! Dlatego stoimy obecnie przed koniecznością znalezienia innych form i innych sponsorów współfinansowania naszej edukacyjnej działalności. Będziemy takich form szukać (mamy kilka niezłych chyba pomysłów w tym zakresie), gdyż uważamy, że kupując „Bajtki”, Czytelnik płaci nie tylko za papier i za treść wydrukowanych na nim artykułów — ale również za to, że dzięki „Bajtkowi” może uzyskać dostęp do najnowszego sprzętu... Szerzej na ten temat — za miesiąc.

A już teraz zapraszam do rozwiązania wielkiego konkursu zamieszczonego w tym numerze. Główną nagrodą w nim jest komputer „Commodore” ufundowany przez... sklep „Bajtki” w Krakowie!

Wesołych Świąt!

Waldemar Siwiński

Bajtek

**„BAJTEK” — MIESIĘCZNY
DÓDATEK DO „SZTANDARU
MŁODYCH”**

ADRES: 00-687 Warszawa, ul. Wspólna 61. Tel. 21-12-05 Przewodniczący Rady Redakcyjnej: Jerzy Domanski — redaktor naczelny „Sztandaru Młodych”.

ZESPÓŁ REDAKCYJNY:

Waldemar Siwiński — kierownik zespołu „Bajtki”
Marek Czarkowski — sekretarz redakcji „Bajtki”
Roman Pożnański — kierownik działu klanów
Wanda Roszkowska — opr. graficzne

KLANY REDAGUJĄ:

Atari — Wojciech Zientara
Amstrad — Jonasz Mayer
Commodore — Klaudiusz Dybowski, Dominik Falkowski
Spectrum — Marcin Przasnyski

STAŁE WSPÓŁPRACUJĄ:

Andrzej Piłaszek, Janusz Jarmoch, Marcin Borkowski, Łukasz Czekajewski, Waldemar Nowak, Michał Sobieszuk, Maciej Pietras, Marcin Bójko, Jacek Kunowski, Michał Karkuciński

Zdjęcia w numerze Leopold Dzikowski

Fotoskład — Tadeusz Olczak,
Montaż offsetowy — Grażyna Ostaszewska,
Korekta — Maria Krajewska, Zofia Wóltńska

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

Skład techniką CRT-200, przygotowalnia offsetowa i druk: PRASOWE ZAKŁADY GRAFICZNE RSW „PRASA-KSIAZKA-RUCH” w Ciechanowie, ul. Sienkiewicza 51.

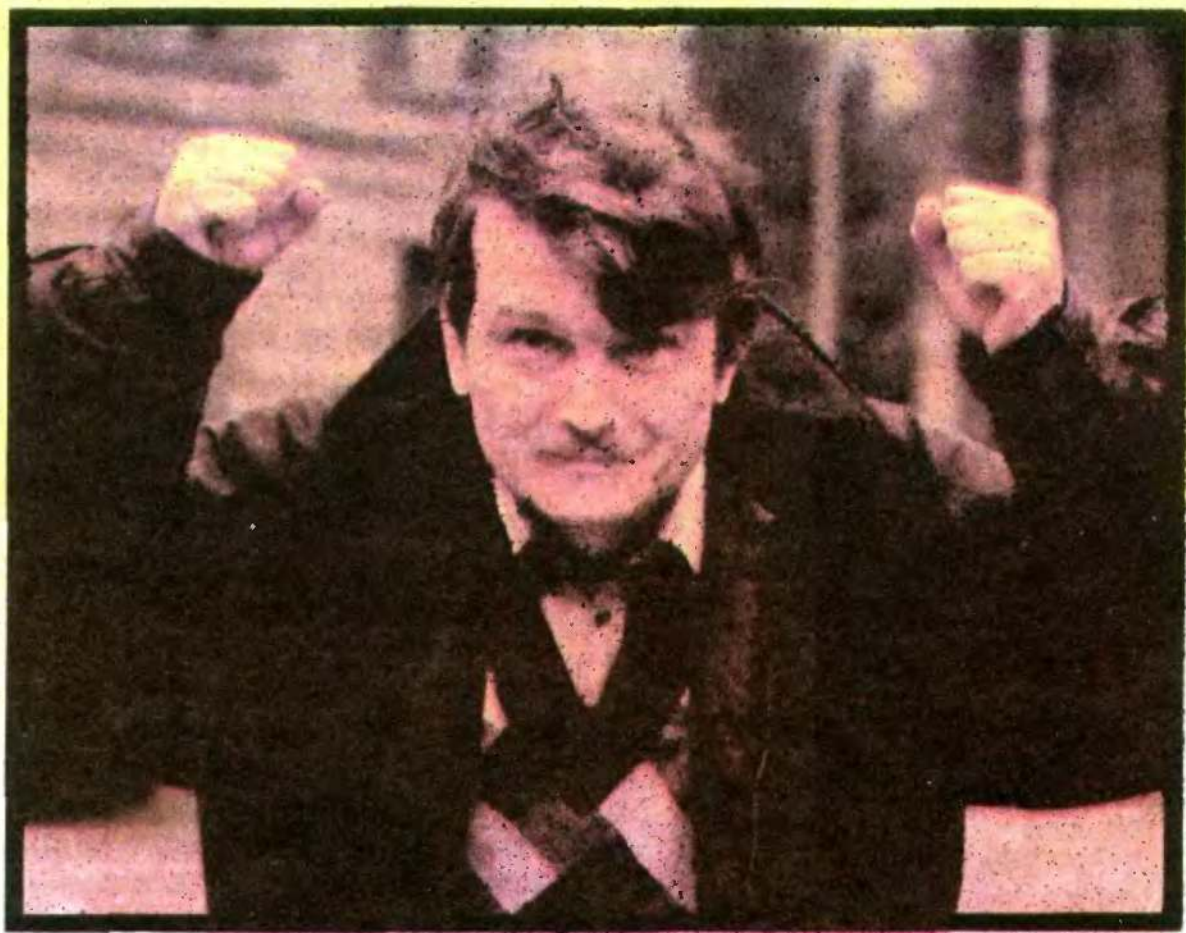
Nr. zlecenia 084919 n. 120 000 egz. A-111

ZA MIESIĄC:

- poznamy dalsze szczegóły multiputera
- uzupełnimy bibliotekę Action!
- narysujemy kreskę w AZTEC C
- zabijemy wirusa!
- zgłębimy wszystkie tajemnice CHIWRITERA
- ożenimy Smoka z Księżniczką Ośmiornicą
- przypomnimy sobie Kubusia Puchatka
- wypróbujemy stację dysków F DD 3000



Ich piekielna maszyna



FASCYNUJE MNIE GŁÓWNIIE PROCES NADAWANIA KOMPUTEROM CECH LUDZKICH.

przed negatywnymi zjawiskami komputeryzacji, takimi jak możliwość naruszenia tajemnic osobistych jednostki, nadmierną koncentrację informacji o obywatelach czy wreszcie możliwością awarii systemów odpowiedzialnych np. za funkcjonowanie elektrowni atomowych, lotnisk, sieci energetycznych kraju, banków itp.

Mógłbym powiedzieć, że literatura s-f tymi zagrożeniami zajmuje się od dawna, ale nie to jest ważne. Obawy, o których mówisz są dla mnie dowodem klęski bezgranicznej wiary w postęp techniczny i nieomyślność człowieka. To jest problem także filozoficzny.

Uczeni Zachodni często argumentowali, jakby nie mieli pojęcia o filozofii — „Dajcie nam więcej pieniędzy na badania, a rozwiążemy każdy problem. Nowoczesne technologie wymagają nakładów”. Problem moim zdaniem leży nie tylko w pieniądzu. Ducha nie da się odtworzyć w maszynie. Człowiek jest tak skomplikowany i bogaty wewnętrznie, że nie sposób go np. „symulować” w komputerze. Tej bariery po prostu nie da się przeskoczyć. Dostrzegam wiele samozadowolenia wśród teoretyków komputeryzacji i jest to bardzo niebezpieczne. Czekać — a może śmieszne?

Rozumiem, że jesteś przeciwnikiem komputerów.

Człowiek ma dziwną potrzebę otaczania się rzeczami, które nie są mu do życia niezbędne. Nie tak dawno stoczyłem prawdziwą batalię z moim synem, który oczywiście chciał mieć w domu ten cud techniki końca XX wieku. Wytłumaczyłem mu, że poza tą rzeczą jest wiele innych ciekawych i wartościowych zjawisk, które warto poznać, np. teatr, film, literatura. Maszyna nie przemówi do niego w języku tych mediów. Osobiście nie umiałbym pracować z komputerem. Lubię stukot maszyny do pisania, szelest kartek papieru i przyjemność lektury. Monitor tej szansy nie daje. Z mojego punktu widzenia komputer jest rzeczą niekonieczną.

Być może twoje poglądy wynikają z tego, że nie miałeś okazji dokładnego poznania możliwości tego narzędzia.

Mam wielu przyjaciół, którzy zajmują się nimi zawodowo — albo oszaleli na ich punkcie jako amatorzy — i to oni uświadomili mi niebezpieczeństwo płynące z wiary w ogromne możliwości tego urządzenia.

A czy to nie jest czasem konserwatyzm?

U człowieka zawodowo parającego się literaturą s-f? Raczej ostrożność. Nie jestem zdeklarowanym przeciwnikiem komputeryzacji, ja tylko wskazuję na zagrożenia. To coś zupełnie innego. Myślę, że dojrzelśmy już do tego, aby ludzi mających odmienne poglądy traktować wyrozumiale. Przeciwnik to nie znaczy wcale wróg, tym bardziej, że być może ma rację.

Jestem pewien, że wielu Czytelników „Bajtki” nie zgodzi się z twoimi obawami.

Przynajmniej niech się zastanowią.

Czy wśród pisarzy uprawiających gatunek science-fiction jest wielu myślących podobnie.

pieczniejsza, z drugiej zaś pojawiło się zjawisko coraz silniejszego uzależnienia od narzędzia jakim jest komputer.

Współpracujący z maszyną człowiek musi zmienić swój język tak, by był on dla niej zrozumiały. A więc w pewnym sensie — cofnąć się, zredukować intelektualnie. Błąd w pracy komputera kontrolującego wyrzutnie rakiety z głowicami nuklearnymi może spowodować zagładę naszej cywilizacji. Wszystko to pobudza wyobraźnię autorów.

Twoją też?

Trochę mniej niż innych. Fascynuje mnie głównie proces nadawania komputerom cech ludzkich. Mam wrażenie, że w tym przypadku człowiek chętnie wchodzi w rolę Boga i powołuje do życia twórcę „na kształt i podobieństwo swoje”. Zbliżamy się do Biblii, do Genesis. To szalenie ciekawe zjawisko — przy czym mamy z nim do czynienia i w sztuce i w technice. Jeszcze dziesięć lat temu „sztuczna inteligencja” możliwa była jedynie w filmie lub na kartach książek s-f, a dziś jest to jeden z kierunków badań naukowych.

Komputery mają „myśleć” jak ludzie, a więc czuć i przeżywać, być może nawet mieć świadomość własnego istnienia. Gdyby nie przekonanie, że to się nie powiedzie, byłbym przerażony.

Obawiasz się, że wkraczamy na niebezpieczne ścieżki dotychczas zastrzeżone dla Boga lub Natury?

Człowiek zawsze grzeszył pychą, kiedy próbował zastąpić jedno albo drugie. Lecz ta postawa jest źródłem niebywałego postępu w dziejach naszego gatunku. Badacze częściej ponoszą klęski niż świętują chwile triumfu. Jednak nawet porażka nie idzie na marne, bowiem uświadamia nam, kim właściwie jesteśmy. Sukces zaś często bywa przyczyną zmartwień.

Zbudowanie bomby atomowej jest tego najlepszym przykładem. W chwili, kiedy nad Ziemią pojawił się pierwszy błysk atomu uczeni poczuli strach, zrozumieli bowiem, że stworzyli coś potwornego. Ci najwybitniejsi: Albert Einstein, Robert Oppenheimer, Enrico Fermi czy Andriej Sacharow stali się przeciwnikami własnego naukowego triumfu.

Wiem, że w Stanach Zjednoczonych coraz częściej pojawiają się głosy uczonych ostrzegających

Rozmowa z Maciejem Pająkiem, literackim mikrofilmem „Fantsystyka”

PROGRAM NAME: ŻEGNAJ ŚWIECIE
DATE: 8/20
CONTENTS: TESTAMENT: DROBI-AZGI
PROGRAMMER: "CHARLES KLU-GE"

Czy poznajesz ten fragment?

Oczywiście, ten program znajduje się w opowiadaniu Johna Varley'a „Naciśnij Enter”, które otrzymało prestiżową nagrodę „Nebula” za najlepsze opowiadanie science-fiction opublikowane w Stanach Zjednoczonych w roku 1985. Była to całkiem prawdopodobna historia faceta, który przy pomocy komputerów chciał opanować świat.

Podobnych opowiadań pojawiło się w pierwszej połowie lat osiemdziesiątych bardzo wiele. Dlaczego twórcy s-f zaczęli tak bardzo eksplorować ten temat?

Zyjemy w epoce silikonowego chipa. Byłbym zdziwiony gdyby pisarze uprawiający ten gatunek literacki ominięli zjawisko masowej komputeryzacji i społecznych konsekwencji, jakie ono niesie. W krajach zachodnich stało się możliwe zdobywanie informacji praktycznie na każdy temat dzięki sieciom komputerowym. Coraz częściej inżynierowie, lekarze, nauczyciele i policjanci w swej pracy korzystają z ekranu i klawiatury.

Ich życie zmieniło się. Z jednej strony praca stała się wygodniejsza i bez-

Tego nie wiem. W każdym razie motyw komputera w literaturze s-f jest do dziś mocno eksploatowany. Wprowadzenie do akcji inteligentnej maszyny daje autorom możliwość szukania atrakcyjnych rozwiązań. Komputery jako system ogarniający rzeczywistość, wzmacniacz dobrych lub złych cech człowieka, symulacja społeczeństwa czy wreszcie jako mit. Być może powie ktoś, że jest to jednak kręcenie się wokół starych tematów. Oczywiście publiczność oczekuje nowych pomysłów. Ja także chętnie przeczytam powieść ujmującą ten temat w zupełnie nowym świetle. Być może podejmiecie się tego zadania?

Obawiam się, że tworzenie dzieł literackich jest o wiele bardziej skomplikowane od pisania programów.

Wiem, że walka z komputerami jest bezsensowna i skazana na przegraną. Przemysł komputerowy musi się rozwijać. Do tego potrzebne są pieniądze — pieniąż to skondensowana energia naszych czasów. Kiedy w połowie lat siedemdziesiątych pojawiły się komputery osobiste, giganci przemysłu elektronicznego tacy jak IBM czy Hewlett-Packard nie dostrzegali szansy, ale pięć lat później pojawiają się pierwsze IBM PC. Można zarobić bardzo dużo sprzedając superkomputery, ale miliardy najłatwiej wyciągnąć od masowego użytkownika i to się udało.

Miliony szeregowych nabywców, z których wielu po miesiącu odłożyło komputer na szafę, sfinansowały Molocha. To jest system doskonalony się napędzający. Im więcej sprzedanych komputerów, tym więcej pieniędzy na rozwój badań nad nowymi generacjami, tym doskonalsze produkty na rynku. Dokąd to prowadzi?

Oczywiście nie czujesz się częścią tego systemu?

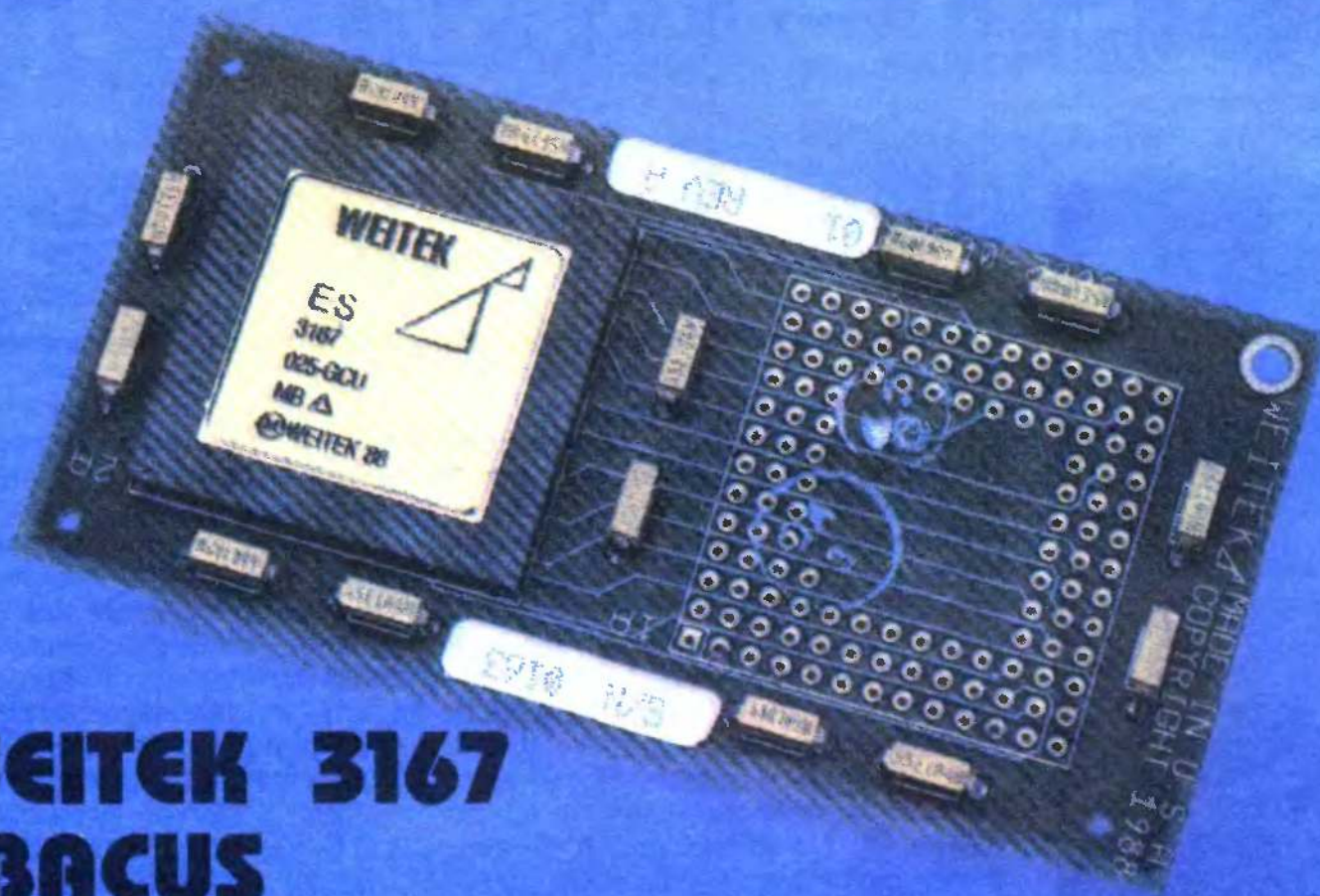
Zachowuję się nie po obywatelsku. Nie używam komputera w pracy, nie prenumeruję „Bajtki”, nie kupiłem synowi tej zabawki, nie lubię gier, dobrowolnie nie zasilam więc systemu swą gotówką. Jednak korzystam z jego usług. Kupuję bilety w „skomputeryzowanych” kasach kolejowych i lotniczanych, moje pieniądze w banku obliczane są jak sądzę przez komputery, moje nazwisko figuruje w pamięci tych maszyn nawet wbrew mojej woli.

Moje obawy płyną z obserwacji rzeczywistości. Literatura, która mnie pasjonuje oddaje się penetracji światów wewnętrznych, wymyślonych być może tylko na potrzeby autorów — sądzę, że jest to bardziej ludzkie od prymitywnej fascynacji technologiami. Dwadzieścia MHz — czterdzieści — sto! Technologiami należy się postugiwać a nie służyć im.

Jedyne co mnie zachwyca to, jak pisał Immanuel Kant, „Niebo gwiazdzone nade mną i prawo moralne we mnie” Bardzo was proszę, niech ta rozmowa nie zostanie zbyt wyprana z „antykomputerowego jadu”.

Pozostawmy ocenę czytelnikom.

Rozmawiali:
Marcin Borkowski
i Marek Czarkowski



WEITEK 3167 ABACUS

Abacus to na nazwa nowego koprocesora firmy Weitek. W kilka miesięcy po wprowadzeniu na rynek koprocesora mW1167 (pisałmy o nim niedawno) wszystkie układy scalone znajdujące się na płytce włożono do wspólnej obudowy i zwiększono maksymalną częstotliwość zegara z 20 do 25 MHz. Daje to szybkość porównywalną z dużym komputerem VAX 8600, czyli około 3.6 miliona operacji zmiennoprzecinkowych na sekundę. Ponieważ Abacus (tak samo jak i mW 1167) ma więcej wyprowadzeń niż koprocesor 80387, potrzebuje specjalnego gniazda. Niektóre komputery oparte o procesor

80386 takim gniazdem dysponują, ale nie wszystkie. Dla tych, które go nie mają, można dokupić kartę rozszerzającą, zawierającą miejsce na oba możliwe do użycia koprocesory — Abacus i 80387. Ponieważ każdy z nich dysponuje odrębnym zestawem komend, nie przeszkadzają sobie nawzajem. Cena szybszej wersji Abacus-a (25MHz) — wg cennika firmy MicroWay — 2300 \$, czyli dwa dobrze wyposażone XT z twardymi dyskami, ale jeśli wziąć pod uwagę koszt komputera VAX 8600 (kilkaset tysięcy dolarów), cena przestanie robić tak duże wrażenie.

(mb)

WEITEK
CACHE
286

Za cenę 400 dolarów można zmienić spokojne wolniutkie XT w szalejący z zegarem 12 MHz AT. Tak przynajmniej twierdzi firma MicroWay, reklamując swój hit — SuperCACHE-286. Jest to karta wkładana do jednego z gniazd płyty głównej, zawierająca procesor 80286, 32 KB szybkiej pamięci podręcznej o czasie dostępu 55ns, („cache memory”) i gniazdo koprocesora 80287. Urządzenie na ogół działa jak asynchroniczny emulator procesora 8088. Asynchroniczny — czyli cała płyta główna może być taktowana zegarem o dowolnej własnej częstotliwości, a procesor 80286 żądający np. dostępu do pamięci RAM poczeka na odpowiedź tyle ile będzie trzeba. Poza emulacją 8088 możliwy jest także inny tryb pracy, w którym SuperCACHE zachowuje się jak osobny komputer, korzystając wyłącznie z pamięci podręcznej, w której zapamiętany jest program i dane. Można wreszcie całkowicie zrezygnować z pomocy szybszego procesora i wrócić do oryginalnego 8088. Co ważne, można to zrobić na drodze programowej, bez konieczności restartowania systemu. Czy mimo wszystko nie lepiej sprzedać stare XT i kupić nowe AT, dokładając wymienioną na początku sumę? Na to pytanie każdy odpowiada sobie sam, ale widocznie nie zawsze lepiej, skoro sporo osób decyduje się na prezentowane rozwiązanie.

(mb)

INTEL RZUCA WYZWANIE IBM-owi

Pojawiają się sygnały, że IBM może przegrać batalię o utrzymanie pełnej kontroli nad standardem szyny dla następnej generacji mikrokomputerów personalnych. Konkurenci na rynku tych mikrokomputerów preferują nowe rozwiązania zamiast IBM-owskiej struktury mikrokanałowej. Przypomnijmy podstawowe fakty. Szyna, to ta część komputera, która realizuje przesłanie danych między wszystkimi elementami np. między dyskami a pamięcią operacyjną. Jej sprawność decyduje o mocy obliczeniowej całej maszyny. Wprowadzając pomyślnie na światowe rynki mikrokomputery PC, firma IBM stworzyła standard sprzętowy, do którego dostosowali się prawie wszyscy wielcy konkurenci aby utrzymać zgodność oprogramowania swoich produktów z oprogramowaniem maszyn IBM. Wprowadzając na rynek nowe modele, zbudowane w oparciu o nowy standard szyny, IBM miał prawdopodobnie nadzieję na umocnienie swojej pozycji właściciela faktycznie realizowanych standardów. Tym razem jednak sztuka może się nie udać. W czasie gdy przygotowujemy materiały do tego numeru (lipiec), czołowy producent układów scalonych, firma Intel przygotowuje prezentację kompletu kości do mikrokomputera opartego o tzw. extended industry standard architecture (Eisa). Jeśli to rozwiązanie zyska popularność, to będzie

alternatywą dla, będącą zastrzeżoną własnością IBM-a, architektury mikrokanałowej. A zainteresowanie nowym rozwiązaniem jest bardzo duże. Główny konkurent IBM-a firma Compaq — zapowiada wypuszczenie na rynek maszyny opartej o Eise jeszcze przed końcem bieżącego roku. Produkcję sprzętu w oparciu o ten standard zapowiadają także m.in. Hewlett-Packard, Zenith Data System, Olivetti, Tandy, NEC i Epson.

Intel nie opowiedział się po żadnej stronie — dostarcza kości i jednym i drugim, gdyż oferuje także elementy do architektury mikrokanałowej. Przedstawiciele firmy poinformowali jednak, że w tej chwili maszyny oparte o tę strukturę, na dużą skalę, produkuje jedynie IBM, choć kilkanaście firm zapowiada budowę maszyn w/g tego standardu.

Powstała sytuacja jest trochę kłopotliwa dla użytkowników mikrokomputerów, szczególnie dla tych, którzy planują dalsze inwestycje. Trudno jest bowiem odpowiedzieć, który standard zwycięży. Zdaniem Paula Otellini, dyrektora oddziału mikrokomputerowego w Intelu, żadne ostateczne rozwiązanie nie nastąpi w ciągu najbliższych trzech, czterech lat. Oba standardy przeżyją, tworząc w ten sposób rozłam na rynku mikrokomputerów personalnych.

(pan)



PODRĘCZNY SKANER

To proste w obsłudze urządzenie umożliwi wczytanie do komputera Atari ST grafiki lub tekstu, zreprodukowanie ich na ekranie monitora, lub wydrukowanie na drukarce. Maksymalna szerokość reproduktowanego obrazu wynosi 105 mm, a rozdzielczość 400 punktów/cal. GENISCANER GS4000 ST jest szczególnie przydatnym partnerem dla osób korzystających z oprogramowania Desktop Publishing. Producent dostarcza wraz z urządzeniem kabel i odpowiedni program użytkowy. GENISCANER jest bardzo prosty w obsłudze. Użytkownik może regulować kontrast i jasność obrazu. Program użytkowy zapisuje kopiowaną grafikę w postaci zbioru, który można potem wykorzystać w innych programach jako gotowy rysunek. GENISCANER jest przystosowany do współpracy z drukarkami kompatybilnymi z Epson.

(jj)

DOSTAŁEM

AMIGĘ

Dostałem Amigę. Dokładniejsze zapoznanie się z nią opóźniło kilka misji, które musiałem wypełnić: uwolnić ambasadę zaprzyjaźnionego państwa od terrorystów („Hostages”), uratować świat od zagłady („Obiliterator”) oraz zestrzelić kilkanaście MIG-ów 29 („F-18 interceptor”). Teraz gdy joystick odmówił posłuszeństwa, mogę zająć się wnętrzem Amigi, a jest czym.

Sercem komputera jest 16-to bitowy procesor Motorola 68000 pracujący z częstotliwością 7,2 MHz. Procesor ma do dyspozycji 512 K RAM i nadzoruje pracę czterech układów specjalizowanych nazwanych: GARY, PAULA, FAT AGNUS oraz DENISE.

Fat Agnus (gruba Agnus), nazwany tak zapewne dlatego, że jest to duży 84-nóżkowy układ w kwadratowej obudowie, kontroluje 25 kanałów danych, m.in. operacje dyskowe i dźwięk minimalizując tym samym interwencje CPU. Poza tym może przenosić szybko obszary pamięci bez udziału procesora co umożliwia bardzo szybką animację.

Denise jest odpowiedzialna za grafikę. Generuje obraz w różnych rozdzielczościach od 320×200 punktów do 640×400, dysponuje paletą 4096 barw, kontroluje 8 sprite'ów oraz wyświetla tekst w 60 lub 80 kolumnach. Ilość kolorów na ekranie ograniczona jest jedynie dostępną pamięcią RAM. Normalnie jest to 32 (5 bitów na piksel) przy rozdzielczości 320×200, 16 lub 4 przy 640×200 i 2 przy rozdzielczości 640×400.

Paula kontroluje dźwięk. Ma cztery głosy z zakresem 9 oktaw każdy, skonfigurowane w dwa kanały stereofoniczne. Używa modulacji amplitudowej i częstotliwościowej. Pozwala na Amidze na naśladowanie wszystkiego co wydaje jakiś dźwięk od Ferrari do Sabriny. Ponadto Paula kontroluje stację dysków i porty wejścia/wyjścia.

Gary spełnia mniej widoczne funkcje — kontroluje szybną danych, porządkuje dane dla CPU i steruje niektórymi funkcjami stacji dysków.

Ze światem zewnętrznym łączy Amigę interface'y Centronics, RS 232, monitor, klawiatura i mysz.

Amigę można podłączyć praktycznie do każdego monitora. Polecane są monitory firmowe 1082 i 1084. Model 1084 jest monitorem bardziej uniwersalnym — ma wejścia TTL RGB (cyfrowe) i RGB analogowe (do Amigi) oraz Composite Video do (Commodore 64 lub magnetowidu) ostatni model 1084 S jest stereofoniczny.

Klawiatura, rozplanowana podobnie do klawiatury IBM AT, ma około 95 klawiszy (zależy to od układu) działa bez zarzutu. Jest dobrze rozplanowana i wyprofilowana. Jednak układ mechaniczny klawiatury AT stwarza większy komfort pracy.

Do Amigi skonstruowano wiele urządzeń peryferyjnych: dodatkowa stacja dysków elastycznych 5.25" lub 3.5", twardy dysk, drukarkę kolorową, modem, baterijny zegar czasu rzeczywistego z dodatkową pamięcią RAM, MIDI, oraz przetworniki dźwięku i obrazu. Działanie tego ostat-



niego jest imponujące gdyż Amiga jest w stanie przetwarzać obraz z magnetowidu lub kamery video w czasie rzeczywistym.

W pudełku z Amigą znajdują się trzy dyskietki: „Workbench”, „Amiga Extras” i „The Very First”. „The Very First” to program firmowy program demonstracyjny dla „bardzo początkujących”. Program pokazuje jak obchodzić się ze stacją dysków, dyskietkami i uczy posługiwania się myszą.

„Workbench” to dyskietka systemowa. Znajdują się na niej zestawy znaków dla Pauli, dane dotyczące konfiguracji systemu oraz program pozwalający na ustawienie tejsze programy obsługujące modem i kilkanaście popularnych drukarek i „drivery” w żargonie komputerowym), zegar, kalkulator, notatnik, program do edycji ikon, syntezer mowy oraz dla nie lubiących ikon i czujących wstręt do myszy CLI (Command Line Interface) pozwalający na komunikowanie się z komputerem w sposób klasyczny (tzn. za pomocą klawiatury).

Zegar normalnie analogowy lub opcjonalnie cyfrowy, zrobiony jest w bardzo ciekawy sposób. Sam rysunek zegara tj. tarczy ze wskazówkami zdefiniowany jest wektorowo. Można dowolnie zmienić rozmiary okna a tarcza zegara wypełnia zawsze całe okno. Ciekawą funkcją zegara jest alarm. Można nastawić sobie budzik, który wyrwie nas z twórczego zapędu o określonej porze. Jedyną wadą budzika jest jego mała radykalność. Pojawia się tylko raz i daje się łatwo usunąć z widoku przez zamknięcie lub schowanie okna z zegarem. Budzik powinien zapisywać ostatnio przetwarzane dane na dysk i zawieszać bezpowrotnie system. Byłby to o wiele skuteczniejszy sposób na oderwanie człowieka od komputera.

Kalkulator nie oferuje nic specjalnego, a notatnik jest bardziej programem do wydruku ładnie wyglądających tekstów niż podręcznym notatnikiem. Oferuje kilkanaście krójów pisma o różnych matrycach (od 8×8 do 20×20 punktów). Każdy krój może być pochylony, podkreślony lub wytłuszczony a na dodatek można to wszystko wydrukować na drukarce. Sam proces uruchomienia programu trwa ok. minutę a jak na podręczny notatnik to zbyt długo, zwłaszcza że zajmuje dużo pamięci.

Program do edycji ikon pozwala na zaprojektowanie własnych ikon, lecz jest to czasochłonne bo nie oferuje funkcji graficznych takich jak rysowanie linii, okręgów itp.

CLI, oprócz zdublowania funkcji osiągalnych przy pomocy operowania ikonami, takich jak formatowanie dysku, zmiana nazwy czy skopiowanie pliku, daje dostęp do bardzo przydatnych procedur. Przede wszystkim do Disk Doctor'a — programu „naprawiającego” dyskietki, oraz poz-

wala na edycję zbioru startup-sequence (sekwencja startowa — zbiór rozkazów które wykonuje komputer po wgraniu DOS'a).

Trzecia dyskietka — „Amiga Extras” zawiera Amiga Basic i kilka przydatnych programów.

Amiga Basic nie jest rewelacją, przynajmniej w moim odczuciu, mimo, że zawiera wiele funkcji pozwalających na obsługiwaniu grafiki i dźwięku z syntezerem mowy włącznie. Jednak edytor, a więc ta część z którą użera się normalny człowiek jest strasznie powolny, i nie daje komfortu pracy. Sam Basic dalece odbiega od standardu znanego mi z mniejszych komputerów (konkretnie z Atari 800 XL). Pozwala na programowanie, nazwijmy to strukturalne, nie potrzebna jest numeracja linii, można definiować własne procedury itp.

Szufflady „Tools” ma ciekawszą zawartość. Są tam: Micro Emace — pełnoekranowy edytor liniowy („full-screen line editor”), Fed — edytor znaków, Key Toy — ściągawka z układu klawiatury — (przydatna gdy przeddefiniujemy sobie klawiaturę np. na klawiaturę Dvoraka i chcemy wrócić do normalnego układu), Print Files — program drukujący pliki, Icon Merge — program który łączy dwie ikony w ten sposób, że po wybraniu danego narzędzia zamiast wyświetlenia ikony w inwersji wyświetlana jest druga ikona (można uzyskać efekt otwierania się szufflady). Program More wyświetla na ekranie zawartość dowolnego pliku, Free Map — wielkość wolnego RAM a PM z zabawnym rysunkiem zajączka demonstruje jedną z ciekawszych możliwości systemu operacyjnego Amigi: multitasking (wykonywanie na raz kilku programów).

Programik ten rysuje wykres zajętości RAM i stopień zapracowania procesora w czasie. Dopiero na tym wykresie widać jak słodkie życie ma Motorola w Amidze. Uruchomiłem jednocześnie program muzyczny (Sound studio), programiki z szufflady Demos, zegar i PM'a, a nie udało mi się zająć procesora tak aby pracował pełną mocą dłużej niż sekundę. Doświadczenie skończyło się zawieszeniem systemu. Zabrakło RAM-u.

W krajach cywilizowanych (czytaj: tam gdzie technika komputerowa jest na wysokim poziomie) Amiga zaliczana jest do klasy Home Computer tzn. można przy jej pomocy pisać teksty, przysyłać dane, bawić się w przetwarzanie dźwięku, lub obrazu i od czasu albo nawet częściej, postrelać do wrednych kosmitów. Jest nowoczesnym następcą wiernych Commodore 64 i ZX Spectrum.

Marcin Bójko

SOUND HACKER

Wielu amigantów lubi pochwalić się muzyką ze swojego komputera. Trzymają gry tylko ze względu na dobrą oprawę dźwiękową. Gry zajmują dużo miejsca, a dysków szkoda. Problem ten pomoże rozwiązać Sound Hacker.

Program pozwala na wydobycie z pamięci komputera modułów SOUND TRACKERA. Jest to aktualnie najlepszy program tego typu dostępny na krajowym rynku (jeżeli rynek w ogóle istnieje). W celu uzyskania ścieżki dźwiękowej z jakiegos gry należy:

1. Wgrać program z wybranym utworem i zrestartować komputer w czasie gdy gra muzyka.

2. Bezpośrednio po tym wgrać

SOUND HACKERA i wybrać opcję wyszukiwania (L — obszar 0.5 MB i dodatkowo — dla właścicieli rozszerzonej pamięci — opcję H). UWAGA! W przypadku podania opcji H przy posiadaniu pamięci poniżej 1 MB następuje zawieszenie systemu.

Jeżeli utwór okazał się zgodny ze standardem SOUND TRACKERA, program zgłosi to sam, podając jego długość, nazwę i listę

instrumentów. Istnieje również możliwość weryfikacji oraz ewentualnej korekty utworu i odtwarzania utraconych fragmentów. Muzykę można nagrać jako „MODULE” lub „SONG”.

Należy dodać, że wydobycie utworu z pamięci nie będzie możliwe po tzw. długim resecie (kiedy pamięć zostaje wyzerowana).

Mr. MatSOFT

HARDCOPY

Na 8-bitowe ATARI istnieje wiele ciekawych programów graficznych pracujących w 8 trybie graficznym np. TOPS GRAPH, DESIGN MASTER. Niestety, żaden z nich nie daje możliwości drukowania więcej niż dwóch obrazków naraz.

Czasami przy tworzeniu różnego rodzaju dokumentacji dobrze jest połączyć np. cztery obrazki, które zajmowałyby cały arkusz A4. Między innymi taką możliwość daje program **HARDCOPY** napisany w Turbo Basicu XL. Program ten tworzony był na drukarkę firmy STAR typu NL-10, ale oczywiście istnieje możliwość przystosowania go do innych drukarek przez zmianę odpowiednich kodów sterujących.

W linii 1350 procedury INIT znajduje się ciąg znaków ustawiających drukarkę w tryb graficzny 27, 65, 8, 27, 50. Pierwsze trzy bajty to definicja odstępu międzyliniowego 8/72 cala, a kolejne dwa bajty to ustawienie zdefiniowanego odstępu. W procedurach LINE1 i LINE2 znajdują się kody formatujące wydruk jednej linii w trybie graficznym. Wygląda to tak 27, 42, 0x, n1, n2, m1, m2,...

gdzie pierwsze trzy bajty określają tryb graficzny:

- dla x=0 — 60 punktów/cal
- x=1 — 120 punktów/cal
- x=2 — 120 punktów/cal z podwójną prędkością
- x=3 — 240 punktów/cal

- x=4 — 80 punktów/cal (CRT graphics)
- x=5 — 72 punktów/cal (plotter graphics)
- x=6 — 90 punktów/cal (CRT graphics type II)

kolejne dwa bajty (n1 i n2) określają ilość bajtów do wydrukowania w trybie graficznym:

ilość bajtów = n2x256+n1
następnie bajty (m1, m2,...) w ilości jw. określają drukowaną informację, przy czym znaczenie poszczególnych bitów jest następujące:

- 128
- 64 najstarszy bit określa punkt
- 32 górny, a najmłodszy — dolny
- 16 (gdy bit ustawiony — punkt
- 8 będzie drukowany)
- 4 drukowany)
- 2
- 1

Należy również pamiętać, że po każdej wydrukowanej linii, należy wystąpić do drukarki znaki CR i LF (13,10)!

KOLEJNOŚĆ INFORMACJI WYSYŁANYCH DO DRUKARKI

- 1) Ustawienie odstępu międzyliniowego.
- 2) Początek pętli (dla linii graficznych).
- 3) Wystąpienie do drukarki linii graficznej.
- 4) Wystąpienie znaków powrotu karetki i przejścia nowej linii.
- 5) Koniec pętli (dla linii graficznych).

Mam nadzieję, że zamieszczone tu informacje będą pomocne w zrozumieniu działania programu przed jego przeróbkami, bądź też przy pisaniu własnego oprogramowania.

Paweł Miasojedow

```

HG 0 REM HARDCOPY
LI 1 REM Pawel Miasojedow
UP 2 REM Copyright (c) Bajtek
NI 3 REM
DI 10 POKE 82,0:POKE 752,255:DIM L$(4):L$="ABCD"
SW 20 DIM S1$(7800),S2$(7800),A$(14),B$(14),C$(14),D$(14),N$(14)
NC 30 DIM X(4),Y(4),T$(14),P$(14)
NC 40 S1$(7800)="X":S2$(7800)="Y"
AG 50 S1=ADR(S1$):S2=ADR(S2$)
YG 60 EXEC MENU
YZ 70 TRAP #BAD
MY 80 GET KEY
ZS 90 M=KEY-48
PL 100 IF M<>1 AND M<>2 AND M<>4 AND M<>5 THEN 60
CC 110 IF M=5 THEN DOS
II 120 EXEC TAB
YG 130 POSITION 3,20:? "CZY CHCESZ KATALOG DYSKOW? (Y/N)"
MO 140 GET KEY
MQ 150 IF KEY<>84 THEN POSITION 3,20:? "":GOTO 170
KM 160 CLS:DIR:?:?:? "KONTynuacja"
GB 170 FOR I=%1 TO M
BL 180 POSITION 10,20:? "":POSITION 10,20
ZZ 190 ? L$(I,I);"-";:INPUT " ";N$
FV 200 IF LEN(N$)>12 THEN 180
PJ 210 IF I=1:A$=N$:ELSE:IF I=2:B$=N$:ELSE:IF I=3:C$=N$:ELSE:D$=N$:ENDIF:ENDIF
BX 220 POSITION X(I)+2,Y(I):? N$
UK 230 POSITION 10,20
GA 240 NEXT I
UH 250 POSITION 10,20:? "AKCEPTUJESZ? (Y/N)"
MT 260 GET KEY
    
```

```

IQ 270 IF KEY=78 THEN 60
EY 280 SPR=1
GY 290 FOR K=%1 TO M
JM 300 IF K=1 THEN T$=A$:EXEC INP
KS 310 IF K=2 THEN T$=B$:EXEC INP
LY 320 IF K=3 THEN T$=C$:EXEC INP
NE 330 IF K=4 THEN T$=D$:EXEC INP
GV 340 NEXT K
EK 350 SPR=0
UL 360 CLS:?:?:? "WYBOR SPOSOBU Drukowania"
JU 370 POSITION 8,6:? "PROPORCJE 1/2"
LM 380 POSITION 8,8:? "1/2 1/2 1/2 1/2"
MU 390 GET KEY:POSITION 29,6:IF KEY=49:G=1:?"1":ELSE:G=2:?"2":ENDIF
PV 400 POSITION 8,10:? "WNIOSKI"
VR 410 GET KEY:IF KEY=84:INW=1:?"N":ELSE:INW=0:?"N":ENDIF
QM 420 POSITION 8,12:? "DRUKOWANIE"
NH 430 GET KEY:IF KEY=84:TL=1:?"N":ELSE:TL=0:?"N":ENDIF
AG 440 IF M=1 THEN 470
ZR 450 POSITION 8,14:? "MARGINESY"
KF 460 GET KEY:IF KEY=84:MAR=1:?"N":ELSE:MAR=0:?"N":ENDIF
YA 470 POSITION 8,18:? "AKCEPTUJESZ? (Y/N)"
XJ 480 GET KEY:IF KEY=78 THEN 360
AR 490 POSITION 1,22:? "
CV 500 EXEC INIT
DH 510 ON M EXEC OBR1,OBR2,OBR3,OBR4
AS 520 D=10:EXEC OUT
OA 530 END
HC 540 PROC MENU
SH 550 SETCOLOR 2,0,0:CLS:POSITION 15,2:? "HARDCOPY"
FO 560 ? :? "Program umożliwia drukowanie obrazków"
CO 570 ? " pojedynczych lub połączonych"
AL 580 POSITION 13,8:? "1"
HW 590 POSITION 13,9:? "WYBIERZ"
AS 600 POSITION 13,10:? "1"
BU 610 POSITION 8,12:? "1 obrazek pojedynczy"
VS 620 POSITION 8,14:? "2 dwa obrazki połączone"
US 630 POSITION 8,16:? "3 cztery obrazki połączone"
AQ 640 POSITION 8,18:? "5 D05"
HX 650 POSITION 14,20:? "(1,2,4,5) ";
WA 660 ENDPROC
MI 670 PROC TAB
RR 680 CLS
DB 690 ?
RU 700 IF M=1:?"BRAK OBRAZKOW":ELSE:?"PODAJ NAZWE OBRAZKA":ENDIF
DR 710 IF M=4:POKE 82,4:ELSE:POKE 82,13:ENDIF
CO 720 ?
ZE 730 ? "_____";
IO 740 IF M=4:?"_____":ELSE:?"_____":ENDIF
HH 750 EXEC TAB1
GC 760 IF M=1:?"L":ELSE:?"T":ENDIF
GZ 770 ? "_____";
LM 780 IF M=1 THEN ? "4"
MS 790 IF M=2 THEN ? "4"
WH 800 IF M=4 THEN ? "+";
EF 810 IF M=4 THEN ? "_____";
ZX 820 IF M>1 THEN EXEC TAB1
FP 830 IF M=2:?"_____":ELSE:?"_____":ENDIF:ENDIF
HT 840 POKE 82,0
JP 850 IF M=1:RESTORE 2060:ELSE:IF M=2:RESTORE 2070:ELSE:RESTORE 2080:ENDIF:ENDIF
GY 860 FOR K=%1 TO M
OT 870 READ X,Y:X(K)=X:Y(K)=Y
WX 880 POSITION X(K),Y(K):? L$(K,K);"-";
HK 890 NEXT K
VR 900 ENDPROC
KM 910 PROC TAB1
PR 920 FOR K=%1 TO 3
TA 930 ? "]"
UB 940 IF M=4:?"|":ELSE:?"|":ENDIF
HD 950 NEXT K
WD 960 ENDPROC
UE 970 PROC INP
KN 980 P$="D":P$(3)=T$
CO 990 OPEN #1,4,0,P$
MU 1000 IF SPR=1 THEN 1020
RR 1010 BGET #1,AD,7680
MP 1020 CLOSE #1
YL 1030 ENDPROC
WT 1040 PROC OBR1
AQ 1050 T$=A$:AD=S1:EXEC INP
LF 1060 EXEC DRUK
RV 1070 D=13:EXEC OUT
UZ 1080 EXEC POWT
DL 1090 IF KEY=80:GOTO 1060:ELSE:GOTO 60:ENDIF
YE 1100 ENDPROC
XA 1110 PROC OBR2
    
```


Wielokrotnie wspominaliśmy już o bibliotece Action! Zawiera ona wiele gotowych procedur oraz funkcji i umożliwia korzystanie z nich bez konieczności wcześniejszego definiowania przez użytkownika. System Action! posiada zdefiniowane 47 procedur, 22 funkcje i 5 zmiennych bibliotecznych. Planowaliśmy opublikowanie ich opisu w numerach specjalnych „Bajtki” — „Tylko o Atari”. Jednak ze względu na kłopoty z papierem zostaliśmy zmuszeni do umieszczenia tego materiału w odcinkach w zwykłym „Bajtku”.

KANAŁY IOCB

Wiele procedur i funkcji korzysta z urządzeń zewnętrznych. Dlatego przed opisem biblioteki Action! musimy wyjaśnić dwa terminy: „kanał” i „IOCB”.

IOCB jest to skrót nazwy „Input/Output Control Block” (blok kontroli wejścia/wyjścia). Określa on zespół rejestrów w pamięci RAM służących do przechowywania parametrów komunikacji komputera z urządzeniem zewnętrznym. Rejestry te stanowią jednocześnie połączenie między językiem programowania i systemem operacyjnym komputera. Po przypisaniu jakiegось urządzenia jednego z bloków IOCB, możemy później komunikować się z tym urządzeniem podając jedynie numer bloku. Jest to zbliżone do wybierania programu telewizyjnego przez wybranie odpowiedniego kanału. Kanał jest tu odpowiednikiem IOCB i takie właśnie (w uproszczeniu) jest znaczenie tego słowa.

Atari posiada osiem kanałów I/O (bloków IOCB) ponumerowanych od 0 do 7, czyli jednocześnie może komunikować się z ośmioma urządzeniami. Tak jest jedynie teoretycznie, gdyż IOCB 0 jest wykorzystywany przez edytor, zaś kanał 6 służy do operacji graficznych. Pozostają więc do wykorzystania kanały 1-5 i 7.

PROCEDURY WYJŚCIA

Biblioteka Action! zawiera dwa podstawowe rodzaje procedur służących do wyprowadzenia informacji, czyli do jej wyświetlania, drukowania i zapisywania na kasecie lub dyskietce. Są to procedury **Print** i **Put**, przy czym każda z nich ma kilka odmian.

PROCEDURY PRINT

Najpierw zajmiemy się procedurami, których nazwy rozpoczynają się od słowa „Print”. Pozwalają one na zapisywanie dowolnej informacji przez dowolny kanał. Jaka to informacja i jaki kanał? O tym informują pojedyncze litery dodawane na końcu nazwy procedury (po słowie „Print”). Można więc powie-

zieć, że ogólny format nazwy takiej procedury jest następujący:

Print<typ>[D][E]<parametry>

gdzie: **Print** jest podstawową nazwą procedury;
<typ> wskazuje rodzaj danej, która jest wyprowadzana. Są tu możliwe:
B dana typu **BYTE**,
C dana typu **CARD**,
I dana typu **INT**,
<nic> ciąg znaków;
D oznacza urządzenie (*Device*) i jest stosowane, gdy chcemy wskazać, przez który kanał ma nastąpić wyjście;
E oznacza koniec wiersza (*End of line*) i powoduje zapisanie znaku <RETURN> na końcu danych;
<parametry> są parametrami wymaganymi przez procedurę.

UWAGA: Zarówno „D”, jak i „E” można stosować dowolnie, lecz typ danej musi być zawsze określony, ponieważ brak takiego określenia oznacza ciąg znaków.

Korzystając z podanego wyżej formatu dochodzimy do wniosku, że możliwe są następujące procedury **Print**:

	ciąg	BYTE	CARD	INT
bez wariantów				
Print	PrintB	PrintC	PrintI	
z EOL	PrintE	PrintBE	PrintCE	PrintIE
z IOCB	PrintD	PrintBD	PrintCD	PrintID
oba warianty	PrintDE	PrintBDE	PrintCDE	PrintIDE

Procedury te zostały celowo pogrupowane według typów zapisywanych danych. Dla podzielonych w ten sposób procedur zostaną teraz opisane zastosowania, formaty, parametry i warianty.

W podanym wykazie nie ma jednej procedury **Print**, ponieważ jej działanie jest bardzo specyficzne. Nazywa się ona **PrintF** i pozwala na zapisywanie według ustalonego formatu zarówno liczb, jak i ciągów.

ZAPIS CIĄGÓW

Istnieją cztery procedury, które służą do wyprowadzania ciągów znaków, i utworzone przez wymienione wcześniej warianty.

formaty: **PROC Print(<ciąg>)**
PROC PrintE(<ciąg>)
PROC PrintD(BYTE kanał,<ciąg>)
PROC PrintDE(BYTE kanał,<ciąg>)

parametry: <ciąg> może być stałą

tekstową zamkniętą w cudzysłowy lub nazwą tablicy typu **BYTE ARRAY**

kanał jest poprawnym numerem IOCB (0—7)

Działanie tych czterech procedur jest następujące:

Print wyprowadza przez ustalony kanał IOCB ciąg znaków bez znaku <RETURN> na końcu.

PrintE wyprowadza przez ustalony kanał IOCB ciąg znaków ze znakiem <RETURN> na końcu.

PrintD wyprowadza przez wskazany kanał IOCB ciąg znaków bez znaku <RETURN> na końcu.

PrintDE wyprowadza przez wskazany kanał IOCB ciąg znaków ze znakiem <RETURN> na końcu.

Użycie tych procedur jest bardzo proste, lecz trzeba pamiętać, że dla procedur wykorzystujących jakiś kanał, kanał ten musi być przedtem otwarty (procedura **Open**).

Zapis liczb BYTE

Kolejne cztery procedury służą do wyprowadzania liczb typu **BYTE** w formie dziesiętnej. Podstawą ich nazwy jest „PrintB”, a dozwolone są wszystkie możliwe warianty.

formaty: **PROC PrintB (BYTE liczba)**

PROC PrintBE (BYTE liczba)

PROC PrintBD (BYTE kanał, liczba)

PROC PrintBDE (BYTE kanał, liczba)

parametry: **liczba** jest wyrażeniem arytmetycznym
kanał jest poprawnym numerem IOCB (0—7)

Działanie tych czterech procedur jest następujące:

PrintB wyprowadza przez ustalony kanał IOCB liczbę **BYTE** bez znaku <RETURN> na końcu.

PrintBE wyprowadza przez ustalony kanał IOCB liczbę **BYTE** ze znakiem <RETURN> na końcu.

PrintBD wyprowadza przez wskazany kanał IOCB liczbę **BYTE** bez znaku <RETURN> na końcu.

PrintBDE wyprowadza przez wskazany kanał IOCB liczbę **BYTE** ze znakiem <RETURN> na końcu.

Zapis liczb CARD

Procedury o nazwach rozpoczynających się od „PrintC” służą do wyprowadzania liczb typu **CARD**.

formaty: **PROC PrintC(CARD li-**

czba)

PROC PrintCE(CARD liczba)

PROC PrintCD(CARD kanał, liczba)

PROC PrintCDE(CARD kanał, liczba)

parametry: **liczba** jest wyrażeniem arytmetycznym
kanał jest poprawnym numerem IOCB (0—7)

Działanie tych czterech procedur jest następujące:

PrintC wyprowadza przez ustalony kanał IOCB liczbę **CARD** bez znaku <RETURN> na końcu.

PrintCE wyprowadza przez ustalony kanał IOCB liczbę **CARD** ze znakiem <RETURN> na końcu.

PrintCD wyprowadza przez wskazany kanał IOCB liczbę **CARD** bez znaku <RETURN> na końcu.

PrintCDE wyprowadza przez wskazany kanał IOCB liczbę **CARD** ze znakiem <RETURN> na końcu.

Zapis liczb INT

Ostatnie cztery procedury wyprowadzają liczby typu **INT**. Ich nazwy zaczynają się od „PrintI”.

formaty: **PROC PrintI(INT liczba)**

PROC PrintIE(INT liczba)

PROC PrintID(INT kanał, liczba)

PROC PrintIDE(INT kanał, liczba)

parametry: **liczba** jest wyrażeniem arytmetycznym
kanał jest poprawnym numerem IOCB (0—7)

Działanie tych czterech procedur jest następujące:

PrintI wyprowadza przez ustalony kanał IOCB liczbę **INT** bez znaku <RETURN> na końcu.

PrintIE wyprowadza przez ustalony kanał IOCB liczbę **INT** ze znakiem <RETURN> na końcu.

PrintID wyprowadza przez wskazany kanał IOCB liczbę **INT** bez znaku <RETURN> na końcu.

PrintIDE wyprowadza przez wskazany kanał IOCB liczbę **INT** ze znakiem <RETURN> na końcu.

Zapis według formatu

Procedura **PrintF** pozwala na wyprowadzanie liczb i ciągów w jednokowy sposób poprzez użycie ciągu określającego format. Ciąg ten wskazuje procedurze, jak powinny wyglądać wyprowadzone dane.

format: **PrintF(„<format>”,<dana>I;,<dana>I)**

parametry: <format> ciąg utworzony ze znaków sterujących i tekstu. Tekst jest wyprowadzany dokładnie w podanej formie, a znaki sterujące (maksy-

mum 5) informują o sposobie wyprowadzania parametrów <dana>. <dana> wyrażenie arytmetyczne, które jest wyprowadzane w formie określonej przez odpowiednie znaki sterujące. Pierwszy znak określa format pierwszej <danej>, drugi — drugiej <danej> itd.

Jest to specjalna procedura umożliwiająca zapisanie ustalonej formie przez ustalony kanał (tylko). W zapisywanym ciągu można umieścić do pięciu różnych danych, przy czym każda z nich może mieć odrębną formę zapisu. Znaki sterujące tą formą są następujące:

znak	forma wyprowadzanej danej
%S	ciąg cyfr
%I	liczba typu INT
%U	liczba typu CARD bez znaku
%C	znak kodu ATASCII
%H	liczba szesnastkowa bez znaku
%E	znak <RETURN> (końiec wiersza)
%%	znak „%”

Proszę zauważyć, że dwa ostatnie znaki sterujące nie formatują danych i nie wymagają żadnej danej. Służą one wyłącznie do ustalenia pożądanego formatu zapisywanej strony, a nie danych.

Dozwolone jest umieszczenie w ciągu określającym format maksymalnie pięciu znaków sterujących, a każda dana wymaga własnego znaku sterującego. Wszystkie pozostałe znaki znajdujące się w ciągu są wyprowadzane bezpośrednio, czyli dokładnie w takiej postaci, w jakiej zostały umieszczone w ciągu.

Procedury Put

Grupa bibliotecznych procedur „Put” służy do wyprowadzania pojedynczych znaków, czyli danych typu BYTE lub znaków kodu ATASCII. Procedury te są bardzo podobne do procedur „Print” i dzięki temu opis ich może być skrócony.

formaty: **PROC Put(CHAR znak)**
PROC PutE()
PROC PutD(BYTE kanał, CHAR znak)
PROC PUTDE(BYTE kanał, CHAR znak)

parametry: znak jest wyrażeniem arytmetycznym
 kanał jest poprawnym numerem IOCB (0—7)

Działanie tych czterech procedur jest następujące:

Put wyprowadza przez ustalony kanał IOCB znak bez znaku <RETURN> na końcu.

PutE wyprowadza przez ustalony kanał IOCB znak <RETURN>.

PutD wyprowadza przez wskazany kanał IOCB znak bez znaku <RETURN> na końcu.

PutDE wyprowadza przez wskazany kanał IOCB znak ze znakiem <RETURN> na końcu.

Wojciech Zientara

TURBO FREEZER KILLER

Od pewnego czasu na giełdach komputerowych w kraju zaczęły pojawiać się programy całodyskowe zabezpieczone tzw. Turbo-Freezerem.

Zabezpieczenie to polega na umieszczeniu na dyskietce sformatowanej w gęstości podwyższonej kilku sektorów CRC w ostatniej ścieżce oraz procedury sprawdzającej ich obecność. Dotychczas powszechnie dostępne oprogramowanie i sprzęt nie były w stanie tego skopiować (czyt. zapisać na dysku w gęstości średniej sektorów CRC z własnej woli).

Zabezpieczenie to — choć bardzo dobre, ma zasadniczą wadę: każdy program zabezpieczony jest TAK SAMO! Wystarczy więc odbezpieczyć jeden, by widzieć jak poradzić sobie z następnymi.

Jak poznać, że mamy do czynienia z taką właśnie dyskietką? Otóż programy z nich czytują się w bardzo charakterystyczny sposób — dwueta-powo. W pierwszej części wczytuje się czołówka: w górnej części ekranu

elementy graficzne w grafice 9 (zwykle znak firmowy jakiejś firmy komputerowej). Poniżej napisy: zwykle w stylu „Loading Game”. Całość pojawia się na ekranie powoli, wytaniając się po kawałku, czemu towarzyszy wiele szumu. Dopiero teraz doczytywany jest właściwy program — jeśli na dysku są wadliwe sektory...

Oto jak można sobie poradzić z tym problemem:

1. Skopiować zabezpieczoną dyskietkę (zwaną dalej numer 1) pomijając błędy (pozwala na to np. US SECTOR COPY 4 — w momencie pojawienia się błędu naciskamy klawisz „SELECT” i dyskietka jest czytana dalej). U nas będzie tych błędów kilka.

2. Wczytać zamieszczony obok program i uruchomić.

3. Włożyć do stacji dyskietkę-kopię (numer 2).

4. Postępować zgodnie z poleceniami zawartymi w programie.

Teraz możemy już wczytywać swój program, pomimo że nie ma już na dysku sektorów CRC.

(wist)

```
FK 10 REM TURBO-FREEZER-KILLER
QW 20 REM *** (wist) ***
AK 30 REM Copyright (c) Bajtek
BB 40 REM
AW 100 DIM A$(53):OPEN #1,4,0,"K":M=1
TC 110 READ D:IF D=-1 THEN 130
UY 120 A$(M)=CHR$(D):M=M+1:GOTO 110
RY 130 ? CHR$(125):? :? :? :? "Wloz dysk
zabezpieczony":? " Turbo-Freezerem":?
KK 140 ? "Nacisnij dowolny klawisz.":? :?
CB 150 GET #1,A
LZ 160 X=USR(ADR(A$))
CO 170 ? :? :? :? "GOTOWE !!!":? :? :?
OG 180 END
GX 200 DATA 104,169,73,141,0,3,169,1
JB 210 DATA 141,1,3,169,82,141,2,3
TU 220 DATA 169,19,141,10,3,169,3,141
QD 230 DATA 11,3,169,0,141,4,3,169
WS 240 DATA 64,141,5,3,32,83,228,169
DX 250 DATA 156,141,18,64,169,80,141,2
DX 260 DATA 3,32,83,228,96,-1
```

TAJEMNICE NIEŚMIERTELNYCH (2)

Gra „Universal Hero”, do której opis, rozwiązani i mapa publikowane były w Bajtku nr 1/89, w wersji na 8-bitowe ATARI jest fabrycznie (!) wadliwa. Program zawiesz się w momencie wpisywania swego imienia (identyfikacji) w pomieszczeniu z komputerem. Osobiście miałem przyjemność skończyć tę grę na wspomnianym komputerze dzięki firmie AGASOFT & Z. W., która program poprawiła. Wszystkich zainteresowanych poprawną wersją tej gry odsyłam na Giełdę Bajtka, gdzie podobno jest już do kupienia.

A teraz przedstawię poprawki pomagające ukończyć tę barwną, wciągającą i wszak niełatwą przygodę:

nieograniczony czas uzyskamy po zamianie pierwszego bajtu w rozkazie: DEC \$0608 (\$CE,08,06; „N” w neg., „CTRL-H” „CTRL-F”) na: \$AD; „—” w neg.

Nieograniczoną ilość tlenu zapewni nam zamiana pierwszego bajtu w rozkazie: DEC \$0645 (\$CE,45,06; „N” w neg., „E”, „CTRL-F”) na: \$AD; „—” w neg.

HENRY'S HOUSE
 Od Andrzeja Neumana z Poznania dowiedzieliśmy się, że w tej grze wystarczy przed jej rozpoczęciem wpisać z klawiatury litery „CPM”, co spowoduje pojawienie się na ekranie małego robaczka i nieśmiertelnie bohatera.

W siedmiu kolejnych grach zamieniamy pierwszy bajt (\$CE; „N” w neg. lub \$DE; „SHIFT-” w neg.) na: \$AD; „—” w neg.

BILBO
 DEC \$1678, X (\$DE, 78,16; „SHIFT-” w neg., „X”, „CTRL-V”).

TALES OF DRAGON
 DEC \$0664, X (\$DE,64,06; „SHIFT-” w

neg., „d”, „CTRL-F”).

DEATH RACE
 DEC \$061E (\$CE,1E,06; „N” w neg., „ESC-CTRL-+”, „CTRL-F”).

ENCOUNTER
 DEC \$2B13 (\$CE,13,2B; „N” w neg., „CTRL-S”, „+”).

QUASIMODO
 DEC \$7E8D (\$CE,8D,7E; „N” w neg., „CTRL-M”, „ESC-DELETE”).

ZONE-X
 DEC \$059A (\$CE,9A,05; „N” w neg., „CTRL-Z” w neg., „CTRL-E”).

LABIRYNTH
 DEC \$7018 (\$CE,18,70; „N” w neg., „CTRL-X”, „p”).

DEC \$7023 (\$CE,23,70; „N” w neg., „SHIFT-3”, „p”).

Pięć kolejnych poprawek polega na zamianie pierwszego bajtu w rozkazie (\$C6; „F” w neg., lub \$D6; „V” w neg.) na: \$A5; „SHIFT-5” w neg.)

ABRACADABRA
 DEC \$86 (\$C6,86; „F” w neg., „CTRL-F” w neg.).

LORD OF THE ORB
 DEC \$C4 (\$C6,C4; „F” w neg., „D” w neg.).

DEC \$C2 (\$C6,C2; „F” w neg., „B” w neg.).

JOUST
 DEC \$A8,X (\$D6,A8; „V” w neg., „SHIFT-6” w neg.).

TAPPER
 DEC \$1C,X (\$D6,1C; „V” w neg., „ESC-CTRL-”).

KARATEKA
 DEC \$B6 (\$C6, B6; „F” w neg., „6” w neg.).

P.S. W imieniu redakcji zwracam się do czytelników, którzy posiadają oryginalną instrukcję do gry tekstowo-przygodowej „THE PAWN” o wypożyczenie nam jej. Z góry bardzo dziękuję.

Tomasz Wiśniewski

ATARI NA ZACHODZIE

Uważni Czytelnicy „Bajtka” z pewnością zauważyli już, że poza komputerami zajmują się także żeglarstwem. Podczas minionych wakacji brałem udział w rejsie do krajów Europy Zachodniej, podczas którego odwiedziłem RFN, Holandię, Belgię i Wielką Brytanię. Oczywiście interesowały mnie tam komputery, przede wszystkim Atari. Chciałbym teraz przekazać Czytelnikom wnioski wynikające z dokonanych obserwacji.

Rynek komputerów domowych został już niemal całkowicie zdominowany przez komputery 16-bitowe. Królują one niepodzielnie zarówno we wszystkich domach towarowych, jak i w specjalistycznych sklepach komputerowych. Tylko gdzieś tam można spotkać jeszcze ósmiobitowe. Są to przeważnie Commodore 64 i 128 oraz Atari XL i XE. Zupełnie zniknęły komputery standardu MSX, a znalezienie pocziwego Spectrum wymaga detektywistycznych zdolności. Podobnie, choć nie tak drastycznie, przedstawia się sprawa oprogramowania i literatury.

SPRZĘT

Spośród różnych wersji małego Atari można spotkać wszystkie popularne modele: 800XL, 130XE i 65XE. Dwie pierwsze odmiany występują najczęściej, a niekiedy bywa także XE Game System. Znacznie gorzej jest jednak z urządzeniami peryferyjnymi. Bez kłopotów dostępne są jedynie magnetofony XC12 oraz stacje dysków XF551. Tylko w dwóch sklepach widziałem Atari 1050, a nigdzie nie było stacji Indus GT (zachodnia wersja LDW 2000).

Dodatkowe wyposażenie komputerów można nabyć wyłącznie w niewielkich, prywatnych firmach, które ogłaszają się w piśmie poświęconych Atari. Wybór jest stosunkowo duży, choć znalezienie odpowiedniego urządzenia może sprawić pewien kłopot. Takiego zakupu znacznie łatwiej jest dokonać w Wielkiej Brytanii niż w pozostałych krajach. Oto wykaz najczęściej oferowanych peryferii: Atari 850, 1050 Super Archiver, US Doubler, XEP-80 oraz różne interfejsy Centronics.

OPROGRAMOWANIE

Programów użytkowych dla Atari należy również poszukiwać w prywatnych firmach. W sklepach bywają zwykle tylko gry. Wybór programów jest dosyć obfity i ciągle pojawiają się nowe, a także poprawione i zmodyfikowane wersje starych. Ponadto dynamicznie rozwija się oprogramowanie „public domain”. Pomimo lawinowej inwazji lepszego sprzętu producenci oprogramowania nie zapominają więc o użytkownikach starszych komputerów.

CZASOPISMA

Dla użytkowników ósmiobitowych Atari wydawane jest na Zachodzie pięć pism, przy czym wszystkie zawierają mniej lub bardziej rozbudowane działy poświęcone Atari ST. Najciekawszym wydaje mi się brytyjski dwumiesięcznik „New Atari User”, który powstał z połączenia „Atari User” i „Page 6”. Dwa następne miesięczniki — „Analog Computing” i „Antic Magazine” — są wydawane w Stanach Zjednoczonych. Większą część pozostałych pism zajmuje już ST. Są to: niemiecki miesięcznik „Atari Magazin” oraz oficjalny dwumiesięcznik Atari — „Atari Explorer”. Bliższe informacje o wymienionych czasopismach będą opublikowane w późniejszych numerach „Bajtka”.

KSIĄŻKI

Najgorzej przedstawia się literatura dla małych Atari. W zasadzie nie publikuje się już nowych książek, a w sklepach komputerowych i księgarniach można spotkać jedynie dodruki starszych wydań. Większość z tych książek trafiła już wcześniej do Polski przywieziona przez prywatnych nabywców.

Wojciech Zientara

WSPÓŁPRACA BASIC-a

Prawie każdy użytkownik mikrokomputera przechodzi przez kolejne charakterystyczne fazy. Początkowo ulega fascynacji grami, które doskonale oddają możliwości komputera w zakresie grafiki, dźwięku i animacji. Następnie zaczyna korzystać z gotowych programów użytkowych, a kiedy i tą fazę ma za sobą, przystępuje do pisania własnych prostych programów w języku Basic.

Kolejny etap to zainteresowanie początkującego „programisty” językiem assemblera zwanym również językiem maszynowym, językiem wewnętrznym lub po prostu assemblerem. Niniejszy artykuł przeznaczony jest głównie dla tych Czytelników, którzy pierwsze trzy fazy mają już za sobą i właśnie odkryli, jak wielkie możliwości dają programy wykorzystujące assembler w stosunku do programów pisanych w języku Basic.

Autor pragnie pomóc tym wszystkim, którzy chcieliby wykorzystać gotowe procedury w assemblerze (można je znaleźć w wielu czasopiśmie poświęconych komputerom i informatyce) we własnych programach napisanych w Basicu i nie bardzo wiedzą jak zapewnić właściwą współpracę obu tych języków. A więc zajmujemy się problemem przekazywania parametrów między tymi dwoma językami. Oczywiście zadaniem autora nie jest pełne opracowanie tego tematu i pokazanie wszystkich możliwych „dróg współpracy” assemblera z językami Basic. Chciałbym jedynie na konkretnym przykładzie przekazać Czytelnikom „przeróbkę” procedury w języku assembler i przystosowanie jej do wykorzystania we własnych programach. Ale na początek trochę „teorii”, bez której raczej trudno byłoby wykorzystać przedstawione w artykule wiadomości.

Inicjacja wykonania programu w kodzie maszynowym — funkcja USR.

Chcąc z poziomu Basica wywołać podprogram napisany w kodzie maszynowym posłużymy się funkcją USR. Jest to funkcja, która wywołana

z argumentem arytmetycznym (z przedziału 0-65535) dokonuje m.in. wpisania tego argumentu do rejestru licznika rozkazów PC, czyli wykonuje skok do adresu będącego argumentem. Tak samo jak każda inna funkcja w języku Basic, funkcja USR nie może być użyta samodzielnie. Aby więc uruchomić podprogram w assemblerze używamy najczęściej zestawień:

RANDOMIZE USR adr, PRINT USR adr, LET x = USR adr itp., gdzie **adr** jest adresem startowym podprogramu. Każde z powyższych zestawień ma swój efekt uboczny, np. **PRINT USR adr** może spowodować wydruk wartości funkcji USR na ekranie (wartością funkcji USR jest zawartość pary rejestrów BC), **LET x = USR adr** spowoduje podstawienie wartości funkcji USR pod zmienną **x**, a **RANDOMIZE USR adr** ustawia zmienną systemową o nazwie SEED. Użycie funkcji USR i jej argumentu musi być przemyślane, gdyż możemy zarówno skasować całą pamięć komputera (np. **PRINT USR 0**) jak też stracić nad nim kontrolę (**RANDOMIZE USR 1000**). Jeżeli program w assemblerze oblicza jakąś wartość a następnie przed powrotem do Basica umieszcza ją w parze rejestrów BC, to możemy również użyć funkcji USR w programie w następującej postaci:

```
.....
100 IF USR 30000 = 5 THEN GOTO 1000
.....
```

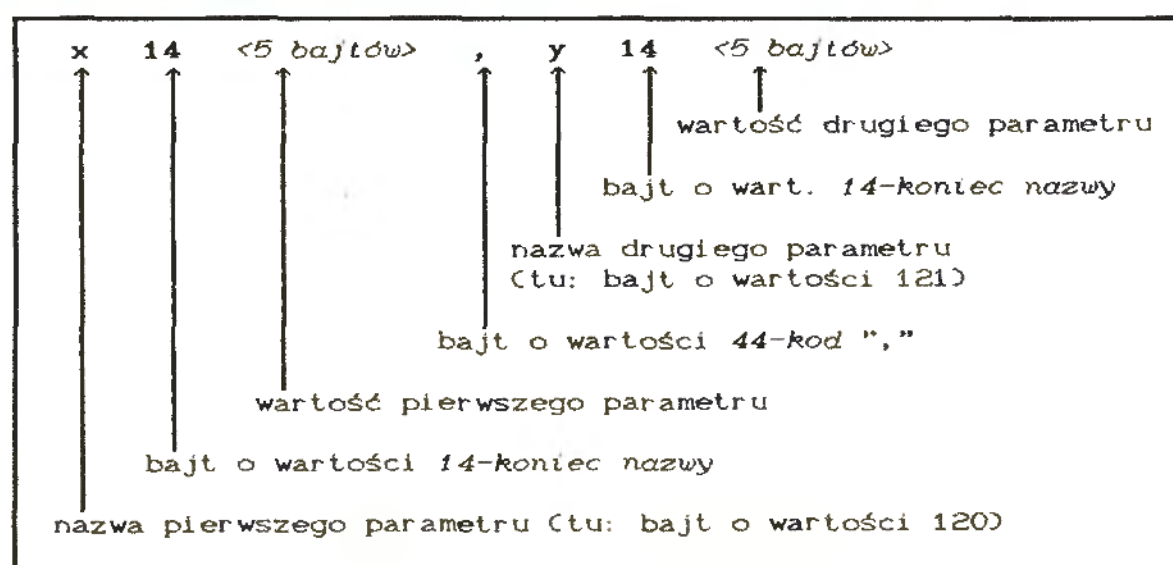
Ważne jest abyśmy zapamiętali, że **wartością funkcji USR jest liczba umieszczona w parze rejestrów BC bezpośrednio przed powrotem do Basic.**

Przekazywanie parametrów z assemblera do Basica.

Jak już wspomniano wyżej, najprostszym sposobem przekazywania wyniku z poziomu assemblera do Basica jest jego umieszczenie w parze rejestrów BC. Wtedy wynik „odnajdziemy” jako wartość funkcji USR. Oczywiście z tego sposobu możemy skorzystać jedynie wtedy, gdy przekazywana wartość „mieści się” na dwóch bajtach (np. młodszy bajt w rejestrze C a starszy bajt w rejestrze B). W wielu programach to ograniczenie jest całkowicie do przyjęcia i dlatego warto zapamiętać ten sposób przekazywania parametrów. Przedstawionym sposobem możemy posłużyć się również w przypadku, gdy chcemy przekazać do Basica dwa parametry będące liczbami całkowitymi z przedziału od 0 do 255. Wtedy jeden parametr wpisujemy do rejestru B (np. **LD B,n**) a drugi do rejestru C (np. jeżeli aktualnie parametr do przekazania znajduje się w akumulatorze (A), to bezpośrednio przed instrukcją powrotu **RET** wpisujemy go do rejestru C instrukcją **LD C,A**). Aby przekazywane parametry znalazły się w odpowiednio w zmiennych **x** i **y**, wystarczy umieścić w programie następujący zestaw instrukcji Basica:

```
.....
100 LET p = USR 30000 (wywołanie podprogramu)
110 LET x = INT (p) * 256
120 LET y = p - 256 x
.....
```

W przypadku gdy przekazywany wynik zajmuje więcej niż 2 bajty, posłużymy się innym sposobem. Zarezerwujemy w pamięci komputera pewne komórki pamięci do przekazywa-



Rys.1. Opis rysunku w tekście

nia parametrów. Ale musimy to zrobić w taki sposób, aby przypadkowo Basic nie mógł nam zmienić zawartości tych komórek (oczywiście nie mam na myśli działania instrukcji **POKE** lecz przypadkowe pokrycie się obszaru Basica z wybranymi przez nas komórkami). Możemy więc wybrać komórki powyżej RAMTOP-u. Następnie przed powrotem do Basica przekazywany parametr (parametry) musimy umieścić w wybranych przez nas komórkach np. przy pomocy instrukcji typu **LD (adr),A**, gdzie **adr** jest jednym z adresów wybranych komórek, a akumulator zawiera parametr do przekazania. Po powrocie do Basica posłużymy się funkcją **PEEK adr**, która to funkcja przyjmuje wartość równą liczbie umieszczonej w komórce o adre-

nie **adr**, np. **LET x = PEEK 65000**. W ten sposób możemy przekazać dowolną liczbę parametrów.

Przekazywanie parametrów z Basica do assemblera

Chcąc przekazać parametry z Basica do assemblera możemy posłużyć się sposobem analogicznym do przedstawionego wyżej. Na poziomie Basica umieszczamy więc poszczególne parametry w wybranych komórkach pamięci (uwagi co do wyboru komórek jak wyżej) przy pomocy instrukcji **POKE adr, par** gdzie **adr** — adres wybranej komórki, **par** — przekazywany (lub jego część) będący liczbą z przedziału 0—255. Następnie już na poziomie assemblera odczytujemy za-

RED	LD	B, 01
	BIT	0, L
	JR	NZ, CHKDE
	BIT	0, E
	JR	NZ, HLRED
	SRL	H
	RR	L
	SRL	D
	RR	E
	INC	B
	JR	RED
HLRED	EX	DE, HL
DERED	SRL	DE
	RR	E
CHKDE	BIT	0, E
	JR	Z, DERVED
	SBC	HL, DE
	JR	Z, EXIT
	JR	NC, HLRED
	ADD	HL, DE
	EX	DE, HL
	CCF	
	SBC	HL, DE
	JR	HLRED
EXIT	EX	DE, HL
LOOP	DEC	B
	RET	Z
	ADD	HL, HL
	JR	LOOP

Rys.2. Procedura obliczania najw. wspólnego dzielnika

	LD	IX, (23563)	221, 42, 11, 92
	LD	H, (IX+5)	221, 102, 5
	LD	L, (IX+4)	221, 110, 4
	LD	D, (IX+13)	221, 86, 13
	LD	E, (IX+12)	221, 94, 12
	LD	B, 01	6, 1
RED	BIT	0, L	203, 69
	JR	NZ, CHKDE	32, 20
	BIT	0, E	203, 67
	JR	NZ, HLRED	32, 11
	SRL	H	203, 60
	RR	L	203, 29
	SRL	D	203, 58
	RR	E	203, 27
	INC	B	4
	JR	RED	24, 237
HLRED	EX	DE, HL	235
DERED	SRL	D	203, 58
	RR	E	203, 27
CHKDE	BIT	0, E	203, 67
	JR	Z, DERVED	40, 248
	SBC	HL, DE	237, 82
	JR	Z, EXIT	40, 9
	JR	NC, HLRED	48, 241
	ADD	HL, DE	25
	EX	DE, HL	235
	CCF		63
	SBC	HL, DE	237, 82
	JR	HLRED	24, 234
EXIT	EX	DE, HL	235
LOOP	DEC	B	5
	JR	Z, KONIEC	40, 3
	ADD	HL, HL	41
	JR	LOOP	24, 250
KONIEC	LD	B, H	68
	LD	C, L	77
	RET		201

Rys.3. Procedura po przeróbce

Z ASEMBLEREM

wartość komórek np. instrukcją **LD A, adr**.

Istnieje jednak bardziej elegancki sposób na przekazywanie parametrów z Basica do assemblera. Jego opis znajduje się w książce K. Kuryłowicza, D. Madeja i K. Maraska — „Przewodnik po ZX Spectrum”. Jednak autorzy tej naprawdę bardzo cennej książki popełnili przy opisie pewien błąd (pominięcie przecinka między parametrami), w związku z czym zamieszczony tam program nie będzie działał poprawnie. Pozwól więc sobie jeszcze raz przedstawić ten sposób „naprawiając” przy okazji powstały błąd.

Posłużymy się tym razem instrukcją **DEF FN** (definicja własnych funkcji w języku Basic). Naszą funkcję o nazwie np. **p** (od słowa parametry) zdefiniujemy w następujący sposób:

```
10 DEF FN p(x,y) = USR adr
gdzie x, y — przekazywane parametry
(oczywiście może ich być większa ilość, gdyż Sinclair Basic dopuszcza taką możliwość),
```

adr — adres startowy podprogramu w języku assemblera

Jeżeli teraz w programie umieścimy np. instrukcję:

```
.....
100 LET x = FN p (10,46)
.....
```

to nastąpi uruchomienie programu w języku maszynowym, a jednocześnie zostaną przekazane parametry (w tym przypadku liczby 10 i 46). Musimy jednak pamiętać, że przekazanie parametrów nie nastąpi automatycznie. Jednak przy wywołaniu tak zdefiniowanej funkcji zmienna systemowa o nazwie **DEF ADD** (2 bajty umieszczone w komórkach o adresie 23563 i 23564) wskazuje na obszar pamięci, gdzie znajdują się parametry funkcji. Obszar ten będzie miał postać taką jak na rys. 1.

A więc kolejne parametry o jednoliterowych nazwach umieszczone są począwszy od 3, 11, 19 itd. bajtu pola wskazywanego przez zmienną **DEF ADD**.

Wartości poszczególnych parametrów zapisywane są w formie liczb zmiennoprzecinkowych na pięciu bajtach. Sytuacja znacznie upraszcza się, gdy wartość parametru jest liczbą całkowitą z przedziału od 0 do 65535. Wtedy wartość parametru zapamiętywana jest w formie uproszczonej: **0,0, bajt mniej znaczący, bajt bardziej znaczący, 0**

W dalszych rozważaniach przyjmiemy właśnie taki przypadek, gdzie parametry są liczbami całkowitymi z przedziału od 0 do 65535. A więc wartości poszczególnych parametrów należy szukać począwszy od 4, 12, 20 itd. bajtu pola wskazywanego przez zmienną **DEF ADD**. Aby „odzyskać” parametry na poziomie assemblera posłużymy się następującą sekwencją instrukcji:

```
LD IX,(23563) wpisujemy zawartość
DEF ADD do rej. IX
```

```
LD C,(IX+4) wartość pierwszego
parametru wpisujemy
```

```
LD B,(IX+5) do pary rejestrów BC
```

```
LD L,(IX+12) wartość drugiego
parametru wpisujemy
```

```
LD H,(IX+13) do pary rejestrów HL
Jeżeli parametry byłyby liczbami z zakresu 0—255, to niektóre z powyższych instrukcji będą zbędne (w tym przypadku pominiemy LD B,(IX+5) oraz LD H,(IX+13) a instrukcję LD L,(IX+12) zastąpimy instrukcją LD B,(IX+12), co spowoduje, że przekazywane parametry znajdą się odpo-
```

wiednio w rejestrach **C** i **B**).

Najwyższy czas na praktyczne wykorzystanie poznanych wiadomości.

Procedura obliczenia największego wspólnego dzielnika dwóch dodatnich dwubajtowych liczb całkowitych.

Taką procedurę napisaną w assemblerze znajdziemy w książce J. Karczmarczuka — Mikroprocesor Z 80 (uwaga: jedną z instrukcji **RR E** zastąpiono instrukcją **RR D**, co z pewnością jest błędem drukarskim). Na rys. 2 przedstawiono właśnie tę procedurę, pomijając komentarze, które w naszych rozważaniach są zbędne. Ważne jest tylko to, że parametry (liczby dla których szukamy największego wspólnego dzielnika) muszą się znaleźć w rejestrach **HL** i **DE**, a wynik znajdzie się również w parze rejestrów **HL** (Czytelników zainteresowanych algorytmem tej procedury odsyłam do wyżej wymienionej książki). Wykorzystując poznaną przed chwilą „teorię” dopiszmy do przedstawionej procedury początek składający się z następującej sekwencji instrukcji:

```
LD IX,(23563)
LD H,(IX+5)
LD L,(IX+4)
LD D,(IX+13)
LD E,(IX+12)
```

w ten sposób zapewniamy sobie, że pierwszy parametr znajdzie się w parze rejestrów **HL**, a drugi w parze **DE**. Teraz musimy jeszcze „przerobić” koniec procedury tak, aby wynik znalazł się w parze rejestrów **BC** (wartość funkcji **USR**). W tym celu instrukcję **RET Z** zastąpimy instrukcją **JR Z, KONIEC** a następnie dopiszemy na końcu procedury:

```
KONIEC LD B, H przepisanie wartości
HL do BC
```

```
LD C,L
RET
```

Procedura po „przeróbce” wyglądać będzie tak jak na rys. 3. Dodatkowo przedstawione kody dziesiętne poszczególnych instrukcji assemblera, co pozwoli dołączyć ją do własnych programów korzystając z instrukcji **READ, DATA** i **POKE**. Założmy jeszcze, że przedstawioną procedurę umieściliśmy w pamięci pod adresem 65000 (procedura jest relokowalna, a więc można ją umieścić pod dowolnym adresem). Teraz możemy już wrócić do Basica.

Na początku programu umieścimy instrukcję **CLEAR 64 999** co odpowiednio obniży RAMTOP i pozwoli w sposób „bezpieczny” umieścić naszą procedurę pod adresem **65000**. Następnie zdefiniujemy funkcję: **10 DEF FN p(x,y) = USR 65000** Chcąc teraz wyliczyć największy wspólny dzielnik liczb np. 24135 i 525 i umieścić go w zmiennej o nazwie **NWP** wystarczy użyć instrukcji: **20 LET NWP = FN p(24135,525)**

W ten sposób przerobiona procedura stała się bardzo wygodnym podprogramem obliczającym największy wspólny dzielnik.

Myślę, że przedstawione wiadomości oraz przykład pozwolą mniej zaawansowanym Czytelnikom na samodzielne wykorzystanie różnych pożytecznych procedur w assemblerze celem dopasowania ich do własnego programu w języku Basic. Bardziej dociekliwych Czytelników odsyłam do wyżej wspomnianych książek.

Zbigniew Krauze

SLANG

Zjawisko stare jak komputer. Najnowszy język. Informatyczny slang. Elektroniczne narzecze.

Te i jeszcze wiele innych sformułowań używa się do określenia pewnego zjawiska ściśle związanego z rozwojem komputeryzacji, języka, którego używają ludzie związani z komputerami.

Ci, którzy nie zetknęli się jeszcze z informatyką, wiadomości podobne traktowali pewnie z przymrużeniem oka. „Pogadają sobie i znudzi im się” — mówiono ze złośliwym uśmiechem na ustach. Na szczęście tak jednak się nie stało i język ten jest aktualnie w rozkwicie.

Od kilkunastu lat, kiedy to pojawiły się pierwsze komputery, a nawet już wcześniej, gdy wszystko istniało jeszcze w teorii, zaczęły tworzyć się pewne określenia ułatwiające porozumiewanie się. Były to krótkie, lecz bardzo „nasączone” słówka, zawierające w sobie bardzo dużo treści. Ich pochodzenie związane było bezpośrednio z dziedziną, w której się pojawiły. Skróty komend, nazw, słów kluczowych — słowa dotąd nieznanne, lecz zrozumiałe od razu dla każdego z „branży”.

Słów tych nie można było znaleźć w żadnych słownikach, encyklopediach lub innych temu podobnych, co więcej, nawet w poważnych komputerowych książkach czy gazetach nie wspomniano o nich.

Nasuwa się więc pytanie — czemu tak się działo? Dlaczego wokół tego zjawiska było tak cicho. Czy jedynym powodem było lekceważenie?

Nie ma na te pytania jednoznacznych odpowiedzi. Wydaje się, iż nikt na serio nie podjął tego tematu, ponieważ nie zaistniała taka potrzeba. Początki lat osiemdziesiątych — pierwsze komputery ośmiobitowe, to lata bardzo z naszej perspektywy zacofane. Pocziwe Spectrums czy inne temu podobne skrzyneczki nie stwarzały jeszcze poważniejszych problemów. Pisanie instrukcji i nauka obsługi odbywała się przy użyciu normalnego słownika. Początki, kietki późniejszego slangu zaczęły pojawiać się tylko wśród młodzieży użytkującej te maszyny.

Wprowadzenie komputerów szesnasto- i trzydziestodwubitowych, oraz nieprawdopodobne skomputeryzowanie całego prawie globu wywołało lawinową reakcję. Ołbrzymie kręgi naukowców-informatyków porozumiewać zaczęły się wyłącznie swoim wewnętrznym językiem, niezrozumiałym dla większości „normalnych” ludzi. Było to zjawisko z gruntu normalne, natychmiastowa reakcją na rewolucyjne zmiany w każdej prawie dziedzinie życia.

Wywołało to odzew w wielu poważnych pismach a czasem nawet w codziennych brukowcach — niezliczone artykuły poruszające ten problem, często nawet ironicznie podchodzące jeszcze do sprawy, stały się normalne. Nadal nikt jeszcze nie sądził, że przybierze to tak kolosalne formy z jakimi spotykamy się dzisiaj. Poważni dziennikarze śmiali się wszystkim w nos, pisząc w swoich ukochanych pismach o beznadziejnych, ich zdaniem, formach porozumiewania się między skomputeryzowanymi ludźmi, porównując ich slang do indiańskiego narzecza. Nie muszę chyba dodawać, że niedługo później poglądy ich uległy szybkim zmianom (przynajmniej na papierze) i śmiech zamieniał się w powagę, a uśmiech w mądrą minę.

Bez kozery można już stwierdzić, że narodził się naprawdę nowy język. Laik nie jest w stanie zrozumieć co mówi do niego

informatyk, podobnie jak technolog żywności nic nie pojmie z wyjaśnień mechanika samolotowego.

Większa część określeń zaliczanych do slangu zaciągnięta została z języka angielskiego, jako bazy wszystkich języków programowych. Oczywiście spolszczono je przedtem i odpowiednio dostosowano do naszych warunków. Teraźniejsza ich forma jest już często nie do poznania, tak więc stworzenie jakiegokolwiek słowniczka wydaje się bez sensu.

Inne rodzaje słów zaliczanych do slangu miały swoje źródło w naszym rodzimym języku. Wymyślane przez nudzących się hackerów, którzy akurat nie mieli żadnego programu do „złamania” lub po raz kolejny „wieszał” się im komputer. Z reguły słówka takie przyjmowały się raczej bezboleśnie i szybko. Czasami jednak zdarzały się określenia zgoła nietrafne i bezużyteczne — te zanikały w naturalnej, komputerowej selekcji. Widać też z praktyki, że na rynku tym istnieje trochę większa konkurencja niż np. w Społem, tak że lepsze pomysły wypierają gorsze, a nie że do sklepu kupuje się to, co lubi wielki Bóg zwany kierownikiem placówki. Z powodzeniem więc, możemy się spotkać czasem ze słowami, które po kilku tygodniach będą już poza użyciem, nie będzie to bynajmniej znacząco, iż slang jest w zaniku. Najprawdopodobniej nie zdarzy to się dotąd, póki istnieć będą komputery, więc jeszcze bardzo długo.

Wiadomym także jest, że aktualnie w większości poważnych czasopism używa się języka bardzo dopracowanego, w którym wplecione są tylko wyrażenia slangowe, w ogólnie akceptowanych mutacjach. Spowodowane jest to najczęściej obawą o to, czy przeciętny czytelnik, który jeszcze nie posiada komputera zrozumie, o co autorowi danego artykułu chodzi. Dlatego też nikt nie ryzykuje stworzenia pisma o bardzo luźnym, lub całkowicie komputerowo-slangowym języku.

Co bardziej codzienny kontakt ze sprzętem żartują, że bez żadnych kart ani innych rozszerzeń komunikują się z komputerem za pomocą języków naturalnych. W praktyce taka komunikacja wygląda w ten sposób, że facet klaruje drukarce, żeby się nie wygłupiała, tylko wyłączyła ten tryb graficzny. Bardzo często też ponagla się twardy dysk, krzyczy na edytor lub ochrzania klawiaturę.

Na zakończenie pragnę powiedzieć, że nie przytaczam żadnych szczegółowych przykładów z dość prozaicznej przyczyny. Otóż nikt nie może dokładnie powiedzieć, które wyrażenia są slangowe, a które po prostu komputerowymi określeniami. Z pewnością słowa „flaki” (wnętrze komputera) lub „drajw” (stacja dysków) można do slangu zaliczyć, lecz większość podobnych określeń nie jest niestety aż tak wyrazista. Pozostaje więc wszystkim chętnym nauki spędzić resztę życia poznając ten „tajemny język”, co jednak stanowczo odradzam. Myślę, że dużo bardziej pożyteczna będzie nauka jakiegoś konkretnego języka programowania, byleby tylko nie był to Turbo Pascal pod CP/M'em

Lukasz Czekaewski

UJARZMIANIA SMOKA CZĘŚĆ DRUGA

Każdego smoka można owoić, nawet takiego, który posiada dwie głowy.

Będziemy mieli do czynienia ze wspaniałym stworzeniem — groźnym dla obcych, lecz przyjaznym i pożytecznym dla wtajemniczonych. Tym smokiem jest rewelacyjny napęd trzyczalowy firmy Timex, a jego sercem system operacyjny TOS.

System ten jest bardzo elastyczny i wygodny w obsłudze. Kłopoty zaczynają się dopiero wtedy, gdy chcemy korzystać ze specjalistycznego oprogramowania. Oprogramowanie takie najczęściej zawiera moduł współpracy tylko z magnetofonem kasetowym. Tak jest i w przypadku asemblera GENS 3M21, którym zajmiemy się tu bliżej.

Przedstawiony obok krótki program jest jedną z wielu „dziewic dla smoka”. Uzupełnia on GENS o opcję współpracy z TOS-em.

Po uważnym wpisaniu programu powinno się go zapisać na dyskietce zleceniem **SAVE * „GENS”LINE 1000** oraz zmienić posiadany kod GENS-a: **LET * „staranazwa” TO „GENS.1”**.

Używanie programu jest proste — po przejściu komendą B do Basic-a ukazuje się menu, które pozwala na obsługę dyskietki lub przejście do edytora Basic-a lub powrót do GENS-a.

Działanie opcji współpracy z dyskietką jest analogiczne do działania komendy P GENS-a. Tam trzeba jednak podawać parametry — początek i koniec nagrywanego na taśmę tekstu źródłowego. Nasza „dziewica” nagrywa cały tekst.

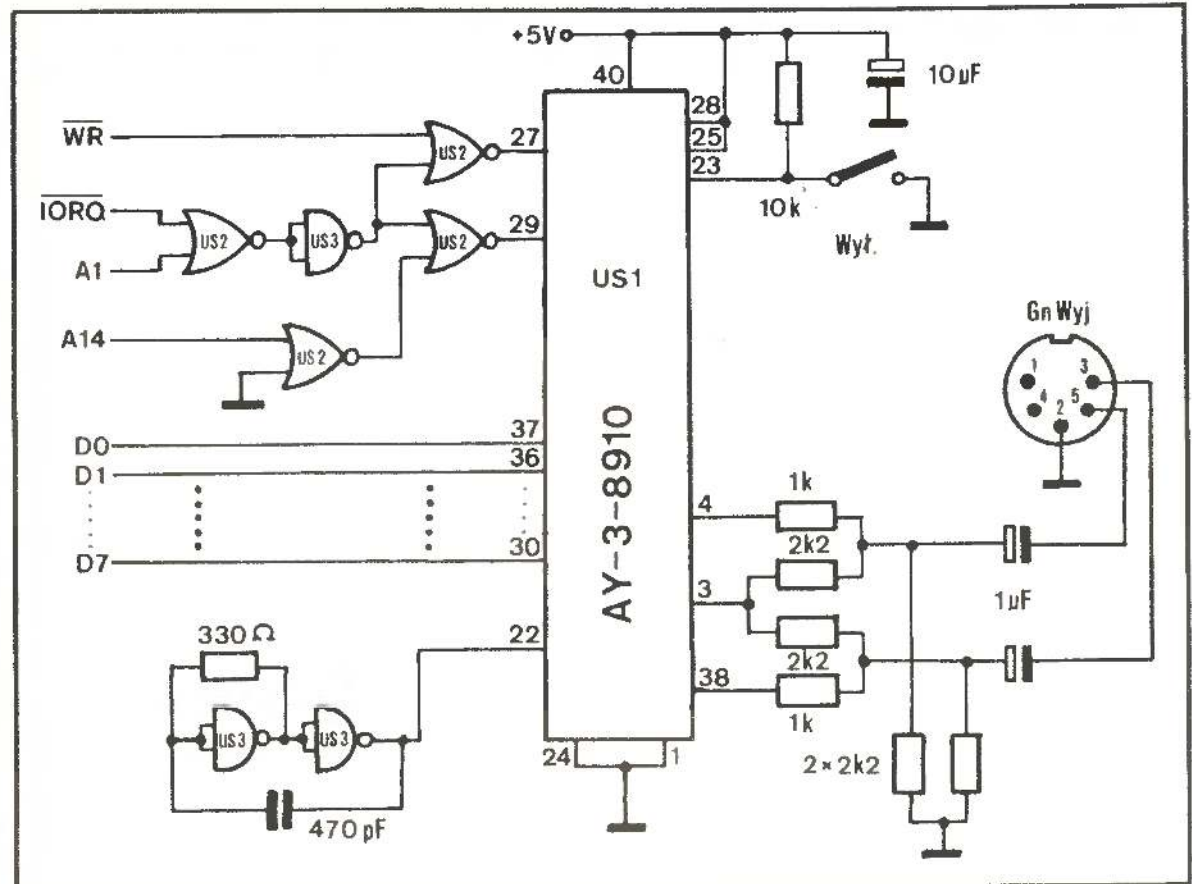
W przypadku łączenia renumracji w edytorze GENS — N jak Natalia.

Kod GENS-a można wgrać pod inny adres (standardowo 32768), lecz zmienić należy linię 1000.

Teraz smok jest bardziej przyjazny, podaje już tapę. W następnych artykułach będę starał się uczyć smoka warowania, szczekania oraz merdania ogonkiem...

Maciej Pietras

GENERATOR TRÓJKANAŁOWY DO SPECTRUM



Rys. 1 Schemat ideowy przystawki

Pojawienie się w Bajtku schematu trójkanałowego generatora dźwięku do ZX SPECTRUM wzbudziło ogromne zainteresowanie Czytelników.

Niestety schemat nie był pozbawiony kilku błędów. Podajemy niniejszym prawidłowy schemat i — miejmy nadzieję — pełny opis proponowanego usprawnienia.

Generator skonstruowany jest w oparciu o układ scalony AY-3-8910 SOUND CHIP. Na schemacie (Rys.1) widzimy sposób podłączenia tego układu do magistrali SPECTRUM. Obwód dekodujący adres urządzenia zbudowany jest z 5 bramek: 4 typu NOR (74LS02) i 1 typu NAND (7400). Jako US2 zastosowano układ w technologii TTL LS, bowiem nie powoduje on zbyt dużego obciążenia szyn mikroprocesora. Jako US3 wystarczy zwykła „zerówka” — jest nawet korzystniejsza, gdyż pewniej pracuje wówczas generator zegarowy. Generator ten zastosowano ze względu na fakt, iż układ ULA w SPECTRUM ma możliwość zatrzymywania systemowego zegara, co powoduje — przy sterowaniu nim układu AY-3-8910 — nieprzyjemne efekty dźwiękowe (dźwięk staje się „metaliczny”). Zewnętrzny sygnał taktujący otrzymywany z tego generatora okazał się dostatecznie stabilny i dokładny. Podane wartości R i C (330 omów i 470 pF) zostały dobrane doświadczalnie.

Układ US1 wytwarza dźwięk w trzech oddzielnych kanałach, do nas należy więc takie jego zmiksowanie, aby otrzymać sygnał stereofoniczny, „przestrzenny”. Temu celowi służy układ zbudowany z 4 rezystorów 2.2 kilooma i 2 rez. 1 kiloom. Z tego układu poprzez kondensatory separujące dźwięk doprowadzany jest do nóżek 3 i 5 gniazdka typu DIN. Do nóżki 2 doprowadzona jest masa. Gniazdko to pozwala na podłączenie naszego generatora do dowolnego wzmacniacza lub magnetofonu za pomocą zwykłego „kabelka stereo”. Jeśli komuś (na razie) nie zależy na dźwięku stereo, to może połączyć końcówki 3 i 5 gniazda i korzystać z sygnału mono.

Na schemacie widzimy wyłącznik. Pozwala on zablokować działanie generatora, co w pewnych przypadkach jest użyteczne (wyl. zwarty — układ zablokowany).

Przystawka nasza zasilana jest z napięcia +5V dostępnego na złączu krawędziowym. Kondensator 10uF (najlepiej tantalowy) służy do odsprężenia zasilania.

Całość należy zmontować na płytce drukowanej z przylutowanym złączem krawędziowym. Świadomie nie podajemy rysunku płytki, jak również położenia odpowiednich sygnałów na złączu, bowiem budowy przystawki powinny się podejmować tylko te osoby, które mają już pewne doświadczenie w konstruowaniu układów elektronicznych. ZX SPECTRUM został skonstruowany w taki sposób, że każda nieostrożność przy korzystaniu z szyny krawędziowej może spowodować uszkodzenie komputera.

Teraz nieco informacji na temat korzystania z przystawki. Jak już wielokrotnie wspomniano na tych łamach, istnieje dużo programów wytwarzających efekty dźwiękowe poprzez nasz generator. Namawiamy jednak do samodzielnego eksperymentowania.

Układ SOUND CHIP wytwarza dźwięk niezależnie od procesora. Wystarczy załadować odpowiednie parametry do wewnętrznych rejestrów tego układu i dźwięk

już płynie, a mikroprocesor w tym czasie realizuje ciąg dalszy własnego programu. Dostęp do wspomnianych rejestrów jest bardzo prosty:

w BASICu OUT (65533), NrRej: OUT (49149), DANA

w asemblerze LD BC,65533
LD A,NrRej
OUT (C),A
LD BC,49149
LD A,DANA
OUT (C),A

Znaczenie i zakres dopuszczalnych wartości danych dla poszczególnych rejestrów (z lewej — numer rejestru):

- 0 — Dokładna wysokość tonu w kanale A (0-255);
- 1 — Orientacyjna wysokość tonu w kanale A (0-15);
- 2 — Dokładna wysokość tonu w kanale B (0-255);
- 3 — Orientacyjna wysokość tonu w kanale B (0-15);
- 4 — Dokładna wysokość tonu w kanale C (0-255);
- 5 — Orientacyjna wysokość tonu w kanale C (0-15);
- 6 — „Wysokość” szumu — wspólna dla 3 kanałów (0-31);
- 7 — Rejestr sterujący. Poszczególne bity bajtu wpisywanego do tego rejestru włączają i wyłączają poszczególne kanały dla szumu i tonu. Włączenie danego kanału dla tonu bądź szumu następuje poprzez wyzerowanie odpowiedniego bitu:

bit 0 — ton A, bit 1 — ton B, bit 2 — ton C,
bit 3 — szum A, bit 4 — szum B, bit 5 — szum C.
Pozostałe nie mają dla nas znaczenia. Oczywiście wszystkie bity numerowane są od najmłodszego do najstarszego.

8 — Amplituda (głośność) sygnału w kanale A (0-15 lub 16). Przy wartości 16 amplitudą steruje wbudowany generator obwiedni, wspólny dla wszystkich kanałów);

9 — Amplituda sygnału w kanale B (0-15 lub 16);
10 — Amplituda sygnału w kanale C (0-15 lub 16);
11 — Dokładny czas obwiedni (0-255);
12 — Orientacyjny czas obwiedni (0-255);

13 — Wybór typu obwiedni i jej wyzwolenie (0-15);
Należy poeksperymentować z typami obwiedni — to najlepszy sposób na ich poznanie. Dodamy tylko, że obwiednia typu 0 to typowa obwiednia opadająca — głośność dźwięku opada do zera przez ustawiony rejestrami 11 i 12 czas.

Jeszcze raz zachęcamy do samodzielnego eksperymentowania z generatorem. Na koniec przytaczamy przykład najprostszego efektu dźwiękowego — „wyststrzału”:

10 OUT 65533,6 : OUT 49149,31 : REM Szum „NIS-
KI”

20 OUT 65533,7 : OUT 49149,7 : REM Szum w kana-
łach A,B i C

30 OUT 65533,8 : OUT 49149,16 : REM Kanał A —
obwiednia

40 OUT 65533,9 : OUT 49149,16 : REM Kanał B —
obwiednia

50 OUT 65533,10 : OUT 49149,16 : REM Kanał C —
obwiednia

60 OUT 65533,12 : OUT 49149,18 : REM Czas opada-
nia dźwięku

70 OUT 65533,13 : OUT 49149,0 : REM Wyzwolenie
obw.type 0

Stanisław Winiecki

```

10 REM © BROMBA
20 RANDOMIZE USR st
35 LET s=st+VAL "8966"
40 PRINT #NOT PI;TAB VAL "4";"
SAVE LOAD MERGE EXIT GENS"
50 BEEP SGN PI;NOT PI; PAUSE N
OT PI: IF INKEY$="e" OR INKEY$="
E" THEN STOP
60 IF INKEY$="S" OR INKEY$="s"
THEN GO TO CODE "d"
65 IF INKEY$="M" OR INKEY$="m"
THEN GO TO VAL "300"
70 IF INKEY$="L" OR INKEY$="l"
THEN GO TO CODE "="
75 IF INKEY$="G" OR INKEY$="g"
THEN INPUT : GO TO NOT PI
80 GO TO CODE "2"
100 LET l=PEEK (st+CODE "6")+VA
L "256"*PEEK (st+CODE "7"): LET
le=l-s
110 GO SUB CODE " COPY ": SAVE
*#CODE s,le:CLS:CAT #n$:GO
TO CODE "("
200 CLS:CAT #"+.gen"
210 GO SUB CODE " COPY "
220 GO SUB VAL "260"
240 LOAD #n$CODE s
245 LET en=CODE a$(VAL "2")+VAL
"256"*CODE a$(VAL "3"):LET end
=s+en:RANDOMIZE end
250 POKE (st+CODE "6"),PEEK VAL
"23670":POKE (st+CODE "7"),PEE
K VAL "23671"
254 GO TO CODE "("
255 INPUT "Name
: LINE n$:LET n$=n$+".gen":R
ETURN
260 OPEN #SGN PI;n$;i;VAL "5"
270 INPUT #SGN PI;a$:CLOSE #
:RETURN
300 CLS:CAT #"+.gen":GO SUB
CODE " COPY "
310 GO SUB VAL "260":LOAD #n$C
ODE END
320 LET en=CODE a$(VAL "2")+VAL
"256"*CODE a$(VAL "3"):LET end
=s+en:RANDOMIZE end:GO TO CO
DE "("
1000 CLEAR VAL "32767":LET st=V
AL "32768"
1010 LOAD #gens.1"CODE st:GO T
O NOT PI
    
```


TETRIS

Do tej gry nie napisano żadnej wymyślnej legendy. Co więcej, nie ma jej w ogóle. Dlatego wolni od spojrzeń całej ludzkości możemy oddawać się tej grze bez sensu.

Pomysł narodził się wśród radzieckich, studentów informatyki. Zadaniem grającego jest układanie i dopasowywanie do siebie spadających z góry elementów. Pozornie jest to bardzo proste, wymaga jednak koncentracji i niesłychanej szybkości kojarzenia. Gra wciąż przyspiesza i zmierza do punktu, gdy nie nadąży się wzrokiem za spadającymi elementami. Po zapchaniu całej wieży — koniec.

Tetris stał się niezwykle popularny dzięki wysokim wymaganiom sprawnościowym, jakie stawia program przed użytkownikiem. Choć pierwsza gra przeznaczona była na IBM i pochodne, to dziś można spotkać ją na każdym modelu mikro- i komputera, w różnych modyfikacjach i odmianach. Istnieje osobna wersja na Spectrum 128 ze wspaniałą oprawą dźwiękową i ogromnie urozmaiconą grafiką. Microsoft Tetris na IBM ma 9 różnych planszy jako tło. Tetris na Commodore posiada dwa wspaniałe ekrany graficzne. Szał Tetris wciąż nie mija.

Obok zamieszczamy listing tej gry na ZX Spectrum. Została ona okrojona z nieistotnych efektów i w 972 bajtach i 20 liniach Basic-a mieści cały Tetris!

Opis programu:

2 — 28 — wpisanie pod adres 60000 kodu maszynowego,

30 — 500 — program w Basic-u rysujący ramkę i ustawiający zmienne niezbędne do działania kodu.

Po przepisaniu programu uruchamiamy go przez RUN. Przez kilka chwil komputer będzie przepisywał kod z DATA do pamięci, a następnie rozpocznie się gra. Z prawej strony widnieje studnia do której spadają klocki, zaś z lewej odpowiedź, jaki będzie następny spadający element. W dolnej linii widoczny będzie zakres poziomów trudności, do wyboru klawiszami 1-9. Sterowanie: 8,9,0 na klawiaturze.

Dobrej zabawy i wielu godzin przy Tetrisie życzy

Michał Sobieszuk

```

2 LET S=0: FOR I=60000 TO 60971
4 READ A: LET S=S+A: POKE I,A: NEXT I
6 IF S<>95297 THEN PRINT 'BLAD' : STOP
10 DATA 243,175,17,255,255,195,203,17,
,42,93,92,34,95,92,24,67,195,242,21,255,
255,255,255,255,42,93,92,126,205,125,0,2
08,205,116,0,24,247,255,255,255,195,91,5
1,255,255,255,255,255,197,42,97,92,229,1
95,158,22,245,229,42,120,92,35,34,120,92
,124,181,32,3,253,52,64,197,213,205,191,
2,209,193,225,241,251,201,225,110,253,11
7,0,237,123,61,92,195,197,22,255,255,255
,255,255
12 DATA 255,255,245,229,42,176,92,124
,181,32,1,233,225,241,237,69,42,93,92,35
,34,93,92,126,201,254,33,208,254,13,200,
254,16,216,254,24,63,216,35,254,22,56,1,
35,55,34,93,92,201,191,82,78,196,73,78,7
5,69,89,164,80,201,70,206,80,79,73,78,21
2,83,67,82,69,69,78,164,65,84,84,210,65,
212,84,65,194,86,65,76,164,67,79,68,197,
86,65,204,76,69,206,83,73
14 DATA 206,67,79,211,84,65,206,65,83
,206,65,67,211,65,84,206,76,206,69,88,20
8,73,78,212,83,81,210,83,71,206,65,66,21
1,80,69,69,203,73,206,85,83,210,83,84,82
,164,67,72,82,164,78,79,212,66,73,206,79
,210,65,78,196,60,189,62,189,60,190,76,7
3,78,197,84,72,69,206,84,207,83,84,69,20
8,68,69,70,32,70,206,67,65,212,70,79,82,
77,65,212,77,79,86,197
16 DATA 69,82,65,83,197,79,80,69,78,3
2,163,67,76,79,83,69,32,163,77,69,82,71,
197,86,69,82,73,70,217,66,69,69,208,67,7
3,82,67,76,197,73,78,203,80,65,80,69,210
,70,76,65,83,200,66,82,73,71,72,212,73,7
8,86,69,82,83,197,79,86,69,210,79,85,212
,76,80,82,73,78,212,76,76,73,83,212,83,8
4,79,208,82,69,65,196,68,65,84,193,82,69
,83,84,79
18 DATA 82,197,78,69,215,66,79,82,68,
69,210,67,79,78,84,73,78,85,197,68,73,20
5,82,69,205,70,79,210,71,79,32,84,207,71
,79,32,83,85,194,73,78,80,85,212,76,79,6
5,196,76,73,83,212,76,69,212,80,65,85,83
,197,78,69,88,212,80,79,75,197,80,82,73,
78,212,80,76,79,212,82,85,206,83,65,86,1
97,82,65,78,68,79,77,73,90,197,73,198,67
,76,211,68,82
20 DATA 65,215,67,76,69,65,210,82,69,
84,85,82,206,67,79,80,217,66,72,89,54,53
,84,71,86,78,74,85,55,52,82,70,67,77,75,
73,56,51,69,68,88,14,76,79,57,50,87,83,9
0,32,13,80,48,49,81,65,227,196,224,228,1
80,188,189,187,175,176,177,192,167,156,1
90,173,178,186,229,165,194,225,179,185,1
93,184,126,220,218,92,183,123,125,216,19
1,174,170,171,221,222,223,127,181,214
22 DATA 124,213,93,219,182,217,91,215
,12,7,6,4,5,8,10,11,9,15,226,42,63,205,2
00,204,203,94,172,45,43,61,46,44,59,34,1
99,60,195,62,197,47,201,96,198,58,208,20

```

```

6,168,202,211,212,209,210,169,207,46,47,
17,255,255,1,254,254,237,120,47,230,31,4
0,14,103,125,20,192,214,8,203,60,48,250,
83,95,32,244,45,203,0,56,230,122,60,200,
254,40,200,254,25,200,123,90,87
24 DATA 254,24,201,205,142,2,192,33,0
,92,203,126,32,7,35,53,43,32,2,54,255,12
5,33,4,92,189,32,238,205,30,3,208,33,0,9
2,190,40,46,235,33,4,92,190,40,39,203,12
6,32,4,235,203,126,200,95,119,35,54,5,35
,58,9,92,119,35,253,78,7,253,86,1,229,20
5,51,3,225,119,50,8,92,253,203,1,238,201
,35,54,5,35,53,192,58,10,92,119,35,126,2
4,234,66,22
26 DATA 0,123,254,39,208,254,24,32,3,
203,120,192,33,5,2,25,126,55,201,123,254
,58,56,47,13,250,79,3,40,3,198,79,201,33
,235,1,4,40,3,33,5,2,22,0,25,126,201,33,
41,2,203,64,40,244,203,90,40,10,253,203,
48,94,192,4,192,198,32,201,198,165,201,2
54,48,216,13,250,157,3,32,25,33,84,2,203
,104,40,211,254,56,48,7,214,32,4,200,198
,8,201,214,54
28 DATA
0,0,0,0,3,3,0,3,0,1,2,0,0,0,0,4,2,0,0,0,
0,0,0,4,2,0,0,1,0,0,0,4,0,2,2,1,0,0,0,4,
2,0,1,0,0,0,0,4,0,2,1,2,0,0,0,5,2,0,0,2,
1,0,0,0,0,0,0,8,23,1,179,237
30>DIM N$(10,15):DIM W(10)
32 BORDER 6: PRINT AT 3,1:"Wynik: 000
00"
36 PRINT AT 5,1:"Nast. element"
38 PRINT AT 7,4:" ";AT 8,4:" "
;AT 9,4:" "
40 FOR i=1 TO 10: PRINT AT 11+i,0: PAPER
4;" " AND i<10;i: PAPER 7;" "
";" " : NEXT i
50 PRINT AT 0,18:" " : FOR i=
1 TO 20: PRINT AT i,18:" " : NE
XT i
60 PRINT AT 21,18:" "
100 PRINT #0;AT 1,10;"(1-9)"
110 LET i$=INKEY$: IF i$<"1" OR i$>"9"
THEN GO TO 110
115 PRINT AT 1,13: BRIGHT 1;I$: POKE 60
003,CODE I$-CODE "0"
200 LET W=USR 6E4: IF W<=W(10) THEN PR
INT #0;AT 1,5: FLASH 1;" K O N I E C !
! ! " : PAUSE 0: GO TO 50
210 LET P=1
220 IF W<=W(P) THEN LET P=P+1: GO TO 2
20
260 INPUT #0;AT 0,0:"Pozycja nr ";(P);A
T 1,0: LINE a$
300 FOR J=10 TO P+1 STEP -1: LET N$(J)=
N$(J-1): LET W(J)=W(J-1): PRINT AT 11+J,
2: BRIGHT 8;N$(J): NEXT J
310 LET N$(P, TO 10)=A$: LET W(P)=W: LE
T N$(P,11 TO )="00000": LET N$(P,15-LEN
STR$ W TO )=STR$ W: PRINT AT 11+P,2: BRI
GHT 8;N$(P)
500 GO TO 50

```

W•E•R•Y•F•I•K•A•T•O•R

JUŻ PO RAZ CZWARTY

Po raz ostatni przedstawiamy kolejną wersję programu do kontroli wpisywania listingów. Program ten wyświetla sumę kontrolną linii, w której znajduje się kursor programu, suma kontrolna powstaje przez dodanie moduło 256 wszystkich kodów danej linii. Ta suma jest wyświetlana w prawym górnym rogu ekranu w postaci czarnej liczby w zapisie szesnastkowym na białym tle z włączonym BRIGHT-em można zmienić te kolory przez POKE 65239,n:POKE 65242,n — gdzie n to atrybut pół ekranu z sumą kontrolną. Program działa na przerwaniach w trybie IM 2. RAMTOP musi być ustawiony poniżej adresu 64768 FD00. Włączenie wyświetlania sum

```

10 INK 7: PAPER 0: BORDER 0: CLS
20 CLEAR 64767
30 POKE 65100,237: POKE 65101,86: POKE 65102,207: POKE 65103,255
40 LET s=0: FOR f=65145 TO 65318: READ a: POKE f,a: LET s=s+a: NEXT f
50 IF S<>18219 THEN PRINT "BLAD W LINIACH DATA!": STOP
60 RANDOMIZE USR 65290
1000 DATA 245,229,213,197,42,73,92,58,98,254,188,32,13,58,99
1010 DATA 254,189,32,7,58,4,92,254,13,32,30,175,50,97,254
1020 DATA 205,110,25,32,21,126,35,134,35,94,134,35,86,134,35
1030 DATA 27,20,40,4,21,134,24,247,50,97,254,58,97,254,103
1040 DATA 230,15,79,124,230,240,15,15,15,15,111,205,224,254,17
1050 DATA 30,64,205,243,254,105,205,224,254,17,31,64,205,243,254
1060 DATA 33,30,88,54,120,35,54,120,193,209,195,58,0,38,0
1070 DATA 62,9,189,48,4,62,7,133,111,17,128,61,41,41,41
1080 DATA 25,201,6,8,126,235,119,36,19,235,16,248,201,195,121
1090 DATA 254,0,0,0,0,0,0,0,219,2,237,91,83,92,42
1100 DATA 75,92,205,229,25,33,0,253,62,254,119,6,0,35,119
1110 DATA 16,252,61,237,71,237,94,207,255

```

kontrolnych: RANDOMIZE USR 65300, wyłączenie: RANDOMIZE USR 65100. W programie włączenie wyświetlania sum

kontrolnych zrealizowałem za pomocą RANDOMIZE USR 65290, gdyż wywołanie programu od tego miejsca powoduje

dotatkowo skasowanie programu w Basicu.

Witold Kołodziejczyk

SEKRETY TWOJEJ DRUKARKI

Jednym z najpopularniejszych urządzeń peryferyjnych jest niewątpliwie drukarka. Wielu użytkowników często nie zdaje sobie sprawy z możliwości, jakie drzeźnią w tym sprzęcie. Wśród przyczyn tej nieświadomości można wymienić nieznaną języków obcych (a w takich są pisane instrukcje obsługi), korzystanie z drukarki pożyczonej od kolegi, itp. Niniejszy artykuł ma na celu przedstawienie sposobów efektywnego wykorzystania drukarki współpracującej z komputerami Amstrad serii CPC.

Na początek wyjaśnimy sobie zasady sterowania drukarką. Jak wiadomo komputer może przysyłać przez interfejs Centronics kody ASCII (w postaci bajtów). Kodom ASCII odpowiada ściśle określony zestaw znaków. Zestaw ten składa się ze znaków sterujących (o kodach od 0 do 31), znaków alfanumerycznych (liter i cyfr) oraz opcjonalnie znaków specjalnych (np. semigraficznych). Znaki sterujące służą do sterowania urządzeniami wyjściowymi (w naszym przypadku drukarki) a pozostałe znaki do wizualnego przekazywania informacji (tworzenia napisów, diagramów itp.). Istnieje kilka standardów sterowania drukarką, najpopularniejsze z nich to standard Epson (tzw. ESC/P) oraz IBM. W przypadku komputerów ośmiobitowych stosuje się najczęściej standard Epson. Na czym on polega? Otóż, aby ustawić w drukarce żądany krój pisma, marginesy itp. wysyła się do niej odpowiednią sekwencję kodów sterujących. W standardzie Epsona taką sekwencję rozpoczyna kod 27 (tzw. ESC). I tak na przykład aby ustawić w drukarce druk pochyły należy wysłać kolejno kody: 27 52. Powrót do druku prostego otrzymujemy przez wysłanie: 27 53. Oczywiście druk pochyły nie jest jedyną możliwością jaką oferują współczesne drukarki igłowe. Na wydrukach przedstawiono sposób korzystania z tych możliwości z poziomu dwóch najpopularniejszych języków programowania: Basic i Pascala (kompilator Hisoft). W przypadku Basicu sekwencje kodów sterujących najwygodniej jest przypisać zmiennym łańcuchowym, których nazwy odpowiadają funkcjom realizowanym przez te sekwencje. Natomiast w Hisoft Pascalu, ze względu na brak typu łańcuchowego zastosowano procedury. Do zalet tego rozwiązania należy przejrzystość programu oraz łatwe wykorzystanie sekwencji sterujących z parametrami (np. wysuw papieru o n wierszy: procedura Feed). Dla porównania różnic i analogii przyjętych rozwiązań w Basicu i Pascalu proszę porównać na wydrukach ustawianie druku pochyłego (Italic) oraz inicjowanie drukarki (Reset). Kilka słów o specyfice Hisoft Pascala. Otóż aby uzyskać wydruk na drukarce należy wysłać kod 16 (przy pomocy instrukcji write (chr(16)));). Ponowne wysłanie kodu 16 powoduje przywrócenie wydruku na ekranie. W trakcie drukowania może się okazać, że nie następuje wysuw papieru i wszystko jest drukowane w jednej linii. Wynika to z faktu, że Hisoft Pascal na końcu drukowanego wiersza nie wysyła kodu 10 (tzw. Line Feed czyli wysuń papier o jeden wiersz). Aby temu zapobiec należy w drukarce włączyć tryb Auto Line Feed (Auto LF) lub po każdej instrukcji WriteLn wysyłać kod 10 (Write(chr(10));). Dla

przykładu w drukarkach Star LC- 10 tryb Auto LF włącza się przez ustawienie ósmego wewnętrznego mikroprzełącznika w pozycji OFF (przy wyłączonym zasilaniu). Aby efektywnie wykorzystać przedstawione możliwości drukarki należy stworzyć zestaw procedur (Pascal) i zmiennych (Basic) w postaci oddzielnego zbioru zapisanego na taśmie lub dyskietce (tzw. biblioteka). Zbiór taki można bez problemu dołączyć do tworzonego programu i uniknąć w ten sposób konieczności ponownego wpisywania procedur i zmiennych, co jest czynnością żmudną i mało efektywną. W Basicu bibliotekę można dołączyć instrukcją MERGE lub CHAIN MERGE, a w Pascalu należy wczytać bibliotekę po wpisaniu części deklarującej zmienne w danym programie.

Mirostlaw Rybiński

```

10 ESC$=CHR$(27)
20 Reset$=ESC$+"@"
30 Draft$=ESC$+"x0"
40 Nlq$=ESC$+"x1"
50 Italic$=ESC$+"4"
60 NoItalic$=ESC$+"5"
70 Emph$=ESC$+"E"
80 NoEmph$=ESC$+"F"
90 DStrike$=ESC$+"G"
100 NoDStrike$=ESC$+"H"
110 Underline$=ESC$+"-1"
120 NoUnderline$=ESC$+"-0"
130 Overline$=ESC$+"_1"
140 NoOverline$=ESC$+"_0"
150 Superscript$=ESC$+"S0"
160 Subscript$=ESC$+"S1"
170 NoScript$=ESC$+"T"
180 Pica$=ESC$+"P"
190 Elite$=ESC$+"M"
200 Cond$=CHR$(15)
210 NoCond$=CHR$(18)
220 Expand$=ESC$+"W1"
230 NoExpand$=ESC$+"W0"
240 DHeight$=ESC$+"w1"
250 NormalHeight$=ESC$+"w0"
260 Space18$=esc$+"0"
270 Space772$=esc$+"1"
280 Space16$=esc$+"2"
290 PRINT#8,reset$:
300 PRINT#8,nlq$:"czcionka typu NLQ ";
310 PRINT#8,draft$:"czcionka typu DRAFT ";
320 PRINT#8,italic$:"druk pochyly ";
330 PRINT#8,noitalic$:"druk prosty ";
340 PRINT#8,Emph$:"druk wyrazisty ";NoEmph$;
350 PRINT#8,DStrike$:"podwójne uderzenie ";NoDStrike$;
360 PRINT#8,Underline$:"podkreślenie ";NoUnderline$;
370 PRINT#8,Overline$:"Nadkreślenie ";NoOverline$;
380 PRINT#8,Superscript$:"Wykładnik ";
390 PRINT#8,Subscript$:"Indeks dolny ";NoScript$;
400 PRINT#8,Pica$:"Czcionka typu Pica ";
410 PRINT#8,Elite$:"czcionka typu Elite ";
420 PRINT#8,Cond$:"czcionka typu Elite skondensowana ";NoCond$;
430 PRINT#8,Expand$:"Druk poszerzony ";noexpand$;
440 PRINT#8,DHeight$:"podwójna wysokość ";NormalHeight$;
450 PRINT#8
460 PRINT#8.Space18$:"odstęp między liniami 1/8 cala"
470 PRINT#8
    "odstęp między liniami 1/8 cala"
480 PRINT#8
490 PRINT#8.Space772$:"odstęp między liniami 7/72 cala"
500 PRINT#8.
    "odstęp między liniami 7/72 cala"
510 PRINT#8
520 PRINT#8.Space16$:"odstęp między liniami 1/6 cala"
530 PRINT#8.
    "odstęp między liniami 1/6 cala"
540 PRINT#8
550 PRINT#8,Nlq$;Italic$;Expand$;DHeight$;Underline$:" Druk mieszany "
560 PRINT#8,Reset$

```

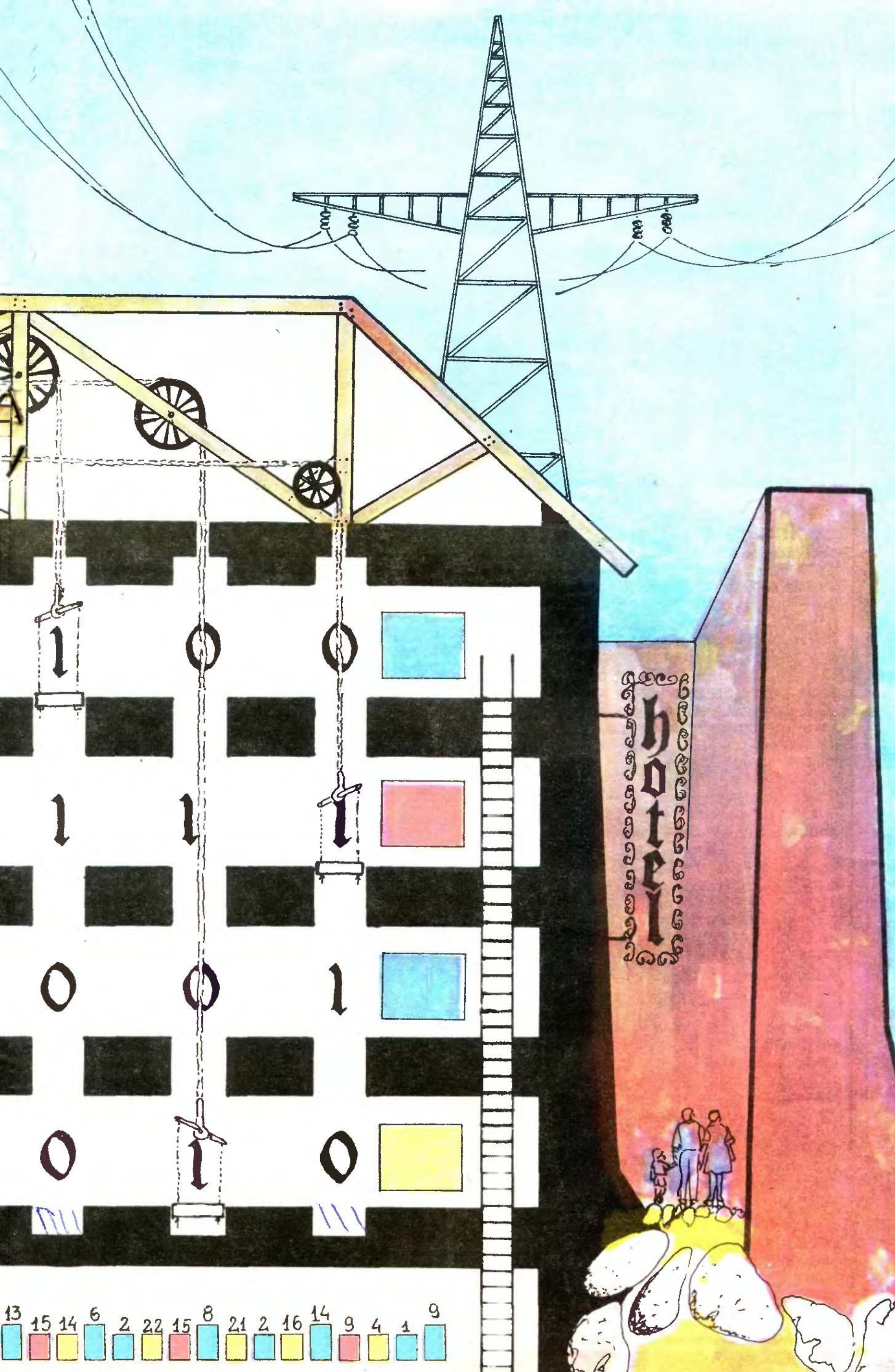
```

1 PROGRAM Printer;
3
5 PROCEDURE Italic;
7 BEGIN write(chr(27),'4') END;
9
11 PROCEDURE Reset;
13 BEGIN write(chr(27),'@') END;
15
17 PROCEDURE Feed(n:integer);
19 BEGIN write(chr(27),'f1',chr(n)) END;
21
23 PROCEDURE HMargins(l,r:integer);
25 BEGIN write(chr(27),'X',chr(l),chr(r)) END;
27
29 PROCEDURE Tab(n:integer);
31 BEGIN write(chr(27),'f0',chr(n)); END;
33
35 (* ----- D E M O ----- *)
37 BEGIN
39 write(chr(16)); (* przełączenie wydruku
41                      na drukarke *)
43 Reset;
45 Italic;
47 Write('Wysuw papieru o 5 wierszy : Feed(5)');
49 Feed(5);
51 Tab(10);
53 WriteLn('Drukowanie od 10 kolumny : Tab(10)');
55 WriteLn;
57 WriteLn('Ustawianie marginesu ');
59 WriteLn('Lewy margines = 20');
61 WriteLn('Prawy margines = 30');
63 HMargins(20,30);
65 WriteLn('Procedura HMargins(20,30)');
67 Reset;
69 END.

```

Listing 1. Procedury pascalowe

Listing 2. Program w BASIC'u



Dyskietyusz nigdy nie przypuszczał, że terminowanie u Mistrza Komputeryka będzie tak trudne, ale cel — tytuł Informatyusza — był zbyt ponętny, by cofnąć się w pół drogi. Ten dzień okazał się jednak wyjątkowo ciężki. Po długich nocnych swawolach z dżojstikiem Dyskietyusz zasnął, i nie dość, że nie zdążył sprzątnąć pracowni Mistrza, to jeszcze zapomniał nakarmić myszy. Nic dziwnego, że Komputeryk był wściekły, i całą swą złość wyładowywał na czeladniku. Dopiero po obiedzie humor mu się nieco poprawił. Zawołał Dyskietyusza do siebie, i stojąc w drzwiach powiedział:

— Jadę dziś na zjazd Informatyusza. Miałem zamiar wziąć Cię ze sobą w nagrodę za szybkie postępy, jakie czynisz, ale musisz ponieść karę za dzisiejsze zachowanie. Zostaniesz w domu i uporządkujesz magazyn. Na podłodze leżą różnokolorowe paczki, które trzeba po kolei przynieść do odpowiednich korytarzy. Wiem, że lubisz sięgać po dżojstik, więc by mieć pewność, że zrobisz, co Ci kazałem, zamknąłem go u siebie w pokoju. Dostaniesz go, gdy powiesz głośno odpowiednie hasło.

Tu Mistrz uśmiechnął się pod wąsem — żeby je poznać, musisz po położeniu każdej paczki na miejscu odczytać z ustawienia wind w magazynie liczbę dwójkową i po zamianie na dziesiętną dodać ją do liczby na paczce. Reszta z dzielenia sumy tych liczb przez 25 wskaże na literę hasła, którą trzeba napisać na paczce.

W tym momencie rozległo się głośne łomotanie do drzwi, i okrzyk „Panie, karetka zajechała!”. Komputeryk wyszedł przed dom. Zegnął Dyskietyusza, mam nadzieję, że poradziś sobie z zadaniem — po czym wsiadł do karetki i odjechał. Gdy uciął stukot kół, czeladnik zgięty w ukłonie wyprostował się nagle.

— Na Świętego McIntosha! Mistrz nie powiedział mi, jaka liczba odpowiada jakiej literze! A to pech! Oj, będzie trzeba się napracować. Liter są na pewno po kolei, według naszego alfabetu, ale jak są ponumerowane? Może ktoś mi pomoże odgadnąć?

* * *

Pomóżcie Dyskietyuszowi w jego kłopotcie! W Magazynie Mistrza Komputeryka znajduje się osiem wind, połączonych parami wspólną liną tak, że gdy jedziemy jedną z wind, druga też się przesuwa. Oczywiście z windy do windy wolno przechodzić tylko wtedy, gdy obie są na tym samym poziomie. Do pierwszej windy trzeba wejść z paczką po schodach, po odstawieniu paczki do korytarza odpowiedniego koloru, na dół schodzi się po drabince. Spośród tych, którzy mu pomogą, Dyskietyusz wylosuje jedną osobę, która dostanie w nagrodę komputer (schowany przed Mistrzem Komputerykiem). Odpowiedzi — czyli hasło otwierające pokój Mistrza — nadsyłajcie do redakcji na kartach pocztowych z naklejonym kuponem. Dyskietyusz czeka na Waszą pomoc!



KUPON KONKURSOWY

Pegaz Ass '89

10

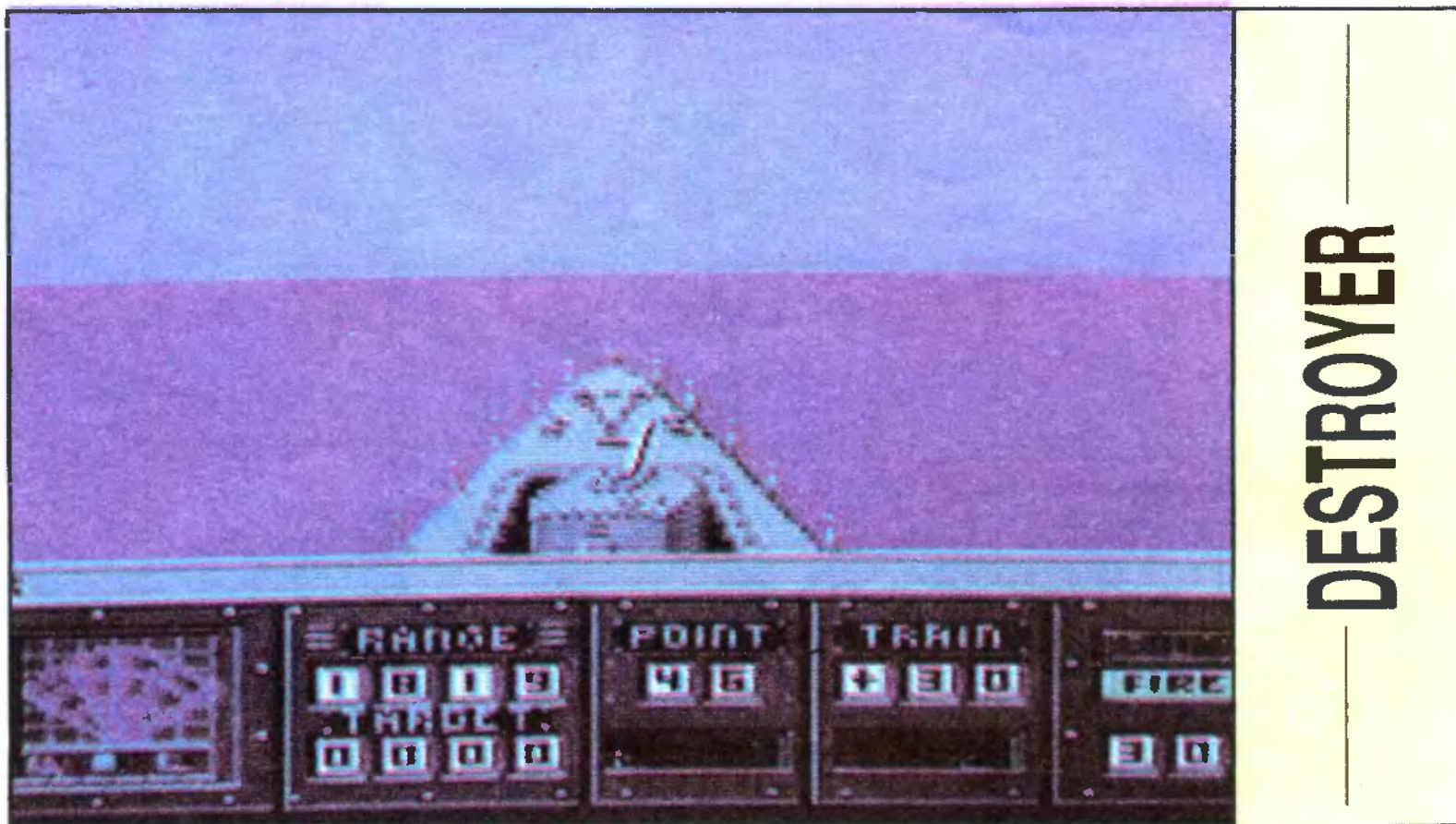
BAJTKOWA LISTA PRZEBOJÓW (11/89)

Ogólna tendencja spadkowa. Dwie nowości — R-Type i Batman — są być może nowościami dzisiaj (29.08.89), ale w listopadzie, gdy to czytacie, nie są rewelacjami. Tym niemniej na Liście budzą pewne zainteresowanie.

Renegade 3 też zaczęło spadać, z dołu podchodzą znane tytuły: Led Storm, Chicago the 30's, Afterburner, Night Raider, Hostages, Navy Moves a nawet... F-19 Stealth Fighter! Czas rozstrzygnie.

Na listopadowe notowanie otrzymaliśmy 3215 propozycji, Czytelnicy głosowali na 203 tytuły.

	ATARI	AMSTRAD	COMMODORE	SPECTRUM
1 R-TYPE	>	<	x	<
2 TOTAL ECLIPSE		<	<	<
3 OPERATION WOLF	>	<	x	x
4 ROBOCOP	>	<	x	x
5 RENEGADE 3	>	<	<	x
6 TRAZ	>	<	<	<
7 CAPTAIN BLOOD	>	x	x	x
8 GAME OVER II		x	<	<
9 CYBERNOID	>	<	x	x
10 BATMAN II	>	<	x	x



DESTROYER

Olbrzymie fale zalewały pokład niszczyciela, który prul wodę z prędkością 40 węzłów. Statek mimo tej oszałamiającej prędkości wykonywał jeszcze kilka rzeczy. Pierwszą było ciągle ostrzeliwanie nieprzyjacielskiego krążownika, drugą — przygotowywanie się do ataku torpedowego. Nagle potężny huk targnął okrętem — to źle przymocowana bomba głębinowa eksplodowała. Szybkość niszczyciela od razu spadła do połowy, wszystkie działa ucichły. Statek był zdany na łaskę krążownika, który zaczął obracać główną artylerię w kierunku uszkodzonego okrętu. W ogłuszającym ryku syren alarmowych spuszczano szalupy. Nic już nie mogło uratować żywego jeszcze wraku, ostrzeliwanego z dział o kalibrze 120 mm, chyba, że... Ty, przyszły admirał!

Jedynym sposobem, aby tego dokonać, jest załadowanie gry o twardym, łamiącym zęby tytule — „DESTROYER”. Program ten, to jedna z najlepszych morskich symulacji, która oprócz niezapomnianych wrażeń podczas godzinnego ładowania (w wersji na Commodore) daje nam jeszcze dużo przydatnej wiedzy. Sterowanie statkiem jest możliwe tylko, gdy znasz odpowiednie kody wywołujące różne części statku. Oto one:

- AB (abandon ship) — opuszczenie statku
- AP (anti-planes) — działo przeciwlotnicze dziobowe
- AS — działo przeciwlotnicze rufowe
- BR (bridge) — mostek
- DA (damage) — zniszczenie statku
- DC (depth charges) — bomby głębinowe
- GA (gun aimed) — ciężkie działo dziobowe
- GF — ciężkie działo rufowe
- NA (navigation) — mapa
- RA (radar) — radar
- SO (sonar) — sonar
- TP (torpedo) — lewa wyrzutnia torped
- TR (track) — kierowanie radarowe
- TS — prawa wyrzutnia torped

Przed wyruszeniem na pełne morze należy wybrać jedną z siedmiu misji. Komputer udziela Ci informacje o miejscu, dokąd musisz się udać, nie pisze jednak, co należy zrobić. Jestem więc zmuszony naprawić ten błąd:

- SUBHUNTER — polowanie na samotne łodzie podwodne. Szukaj ich z dala od wysp, po znalezieniu zbliż się i rzuć bomby.
- SCREEN — walka z kilkoma dywizjonami samolotów torpedowych. Znajdą Cię same, lecz nie oczekuj ich stojąc w miejscu.
- SCOUT — musisz odebrać ściśle tajne dokumenty ze zbombardowanej wyspy. Atak możliwy jest zewsząd, nawet z wysp.
- BOMBARDMENT — jedna ze strategicznych wysp została zajęta przez wojska wroga. Twoje zadanie polega na zniszczeniu dział groźnych dla wojsk, które wylądują tam wkrótce. Liczy się czas!!
- BLOCKADE RUNNER — silna flota nieprzyjaciela zablokowała wyspę. Musisz zniszczyć wszystkie okoliczne jednostki nawodne, samoloty, a potem zapolować na okręty podwodne.
- CONVOY ESCORT — ochrona konwoju tankowców. Atakują Cię samoloty i statki. Nie wpyń na własne okręty.
- RESCUE — ratunek rozbitków. Musisz odnaleźć i uratować pilota samolotu nie dając się zabić.

Po ukończeniu zadania cały dom opanowuje ekstaza. Ojciec biega szczęśliwy, gdyż zwalnia się gniazdko, co pozwala na zgolenie tygodniowego zarostu. Mama cieszy się, że może Cię nareszcie nakarmić. Brat natychmiast zasypia. A Ty płaczesz, ponieważ wciąż widzisz rękę, mianującą Cię admirałem.

Firma: EPYX
Komputer: Commodore 64, Amstrad/Schneider, IBM PC, Atari ST

Luke

KRÓL I KRÓLOWA GIER



Ewa Ostrowska, lat 14
chodzi do klasy 8 szk. nr 96
komputer: Commodore
gry: Defender of the Crown After Burner
hobby: książki, język niemiecki



Robert Drewniak, lat 11
szkoła nr 309
komputer: Atari
gry: Bruce Lee
hobby: sport, piłka nożna





— BMX SIMULATOR + —

Wielki stadion jak zwykle pod wieczór, opustoszał. Dzisiejsze wyścigi motorowe trwały niezwykle długo. Chwilę po tym, jak ostatni z kibiców opuścił trybuny, na horyzoncie pojawiły się cztery osoby, prowadzące ze sobą wspaniałe rowery zwane BMX-ami. Prawie codziennie przychodzili tu, aby poprawić swe i tak już znakomite umiejętności w jeździe po najgorszych nawet trasach. Wielkie pagórki, rowy, kamienie, to tylko nieliczne z wielu atrakcji serwowanych na stadionie.

Tym razem po zewnętrznej miała startować Larissa. Zawodnik ruszający z tej pozycji, na mecie zawsze meldował się na szarym końcu. Dzisiejszy dzień miał być jednak inny. Na sygnał gwizdka wszyscy natarli na pedały. Błoto zmieszane z piaskiem wyleciało spod małych kół, lądując na prostokątnych reklamach. Wszyscy pochylili się do przodu zmuszając rowery do błyskawicznego startu z miejsca.

Pierwszy zakręt był jak zwykle rozstrzygający o

losach wyścigu. Gdy prowadząca — Larissa — zamierzała powiększyć przewagę nad przeciwnikami, jej rower przewrócił się, a ona wylądowała na pobliskiej kupie śmieci. Ten, który w tym momencie zajmował drugie miejsce, po chwili był pierwszym... czarnym od błota. Trzeci z zawodników pomyślnie przebył zakręt, lecz nagle przekoziołkował, przebił jedną z reklam i zaszczycił swoją obecnością pobliski rów. Po tym zakręcie „prawdy” ostatni zawodnik stał się pierwszym.

Atmosfera staje się coraz bardziej gorąca. Po przejechaniu kilku wyścigów na różnych trasach okazało się, że cała czwórka ma szansę na pierwsze miejsce, o końcowej klasyfikacji więc miał zdecydować ostatni wyścig.

Faworytem była Larissa, która ciągle jeszcze miała wszystkie zęby i rower z okrągłymi kołami. Przewidywania sprawdziły się. To, co zostało z roweru Larissy, pierwsze przekroczyło metę. Wręczono medale i po czterech godzinach żaartej walki udano się do domów. Oczywiście oprócz tych, którzy jeszcze przez kilka tygodni wygniali szpitalne łóżka.

Grając w tę grę, należy uważać bardziej od tych, którzy uprawiają ten sport na codzień. Po przegranym wyścigu większość ludzi wykonuje niebezpieczne ruchy, machając pięściami, czasami używają bardziej skutecznych ciosów — łokciami w brzuch kolegi. Spotkałem się też z jeszcze groźniejszym odruchem u jednego z moich przeciwników — wcisnął on po prostu klawisz RESTORE, kończąc niepomyślną dla niego jazdę. Jedynym sposobem na takiego „gentlemana” jest zakrycie prawej części klawiatury grubą dyktą, ewentualnie wyjęcie przycisku. Życzę przyjemnej i owocnej walki.

Firma Code Masters
Komputer: ZX Spectrum 48/+, Commodore 64, Atari XL/XE

Luke



— GRAND PRIX CIRCUIT —

Nie tak dawno bez opamiętania szaleliśmy sportowymi samochodami po górskich szosach za pośrednictwem gry **Test Drive** firmy Accolade, a już możemy wziąć udział w kolejnym samochodowym szaleństwie.

Tym razem jest to prawdziwy (?) rajd samochodowy po ośmiu najtrudniejszych torach Formuły I. Do wyboru są tylko trzy samochody, mianowicie MacLaren, Williams i Ferrari. Każdy z nich ma swe zalety i

wady, lecz zaawansowani kierowcy zawsze wybierają MacLarena. Jest on najszybszy i ma najlepsze parametry, zaś inne cechy właściwie nie liczą się. Tak samo wymaga zmiany biegów (na poziomie trudności wyższym od drugiego) i jest równie podatny na rozbięcie. Pewnym plusem MacLarena są też opony, nieco trwalsze od opon pozostałych dwóch maszyn.

Po obejrzeniu początkowych ilustracji gracz otrzymuje menu, z którego wybiera jeden z trzech

trybów gry: praktykę, rajd na jednej z tras lub Grand Prix na wszystkich trasach. Możliwe do wyboru opcje to wpisanie imienia, wybór poziomu trudności oraz liczby okrążeń toru.

Praktyka jest praktyką i kończy się wypadkiem lub upływem czasu. Rajd po jednej z tras to mała wprawka przed Grand Prix. Ale i tu na każdej trasie następuje tzw. kwalifikacja w celu ustalenia pozycji startowych rajdu. Nie muszę chyba mówić, iż najkorzystniejszą pozycją jest Pole Position, tzn. pierwsza w kolejności i po wewnętrznej stronie pierwszego zakrętu.

Grand Prix to niezwykle trudny maraton. Nasz samochód stawał będzie na starcie aż szesnaście razy (jeżeli się przedtem nie rozbije). Kwalifikacja następuje bowiem na każdej z tras i ma charakter decydujący.

W przerwie między rajdami na kolejnych trasach gracz informowany będzie o aktualnym rankingu kierowców. Trzeba więc starać się wypadać zawsze jak najlepiej, gdyż o wyniku końcowym decyduje liczba uzyskanych punktów.

W trakcie gry notowane są również najlepsze czasy na każdej z tras, z uwzględnieniem samochodu i kierowcy, który ten czas osiągnął.

Niektóre tory wyposażone są w tunele. To właściwie największe utrudnienie, gdyż często trudno jest się w nie zmieścić. A uderzenie w betonową ścianę z prędkością 200 mil na godzinę nie kończy się nawet pogrzebem.

Pamiętać należy również o oponach, które ścierają się bardzo łatwo, szczególnie na ostrej nawierzchni pobocza. Ale w pogotowiu czeka zawsze sprawna brygada, która wymienia koła w 3 sekundy, jeśli zatrzymać się w pit-stopie.

Firma: Accolade
Komputer: ZX Spectrum 48/+, Commodore 64, Amiga, Atari ST, IBM PC

Gen

S.O.S.

Jestem posiadaczem TIMEX'a 2048. Poszukuję gier: TURBO ESPRIT, KUNG-FU MASTER, BARBARIAN. Bardzo proszę, pomóżcie.

Sebastian Kucik, ul. Polna 34/8, 85-163 Bydgoszcz

Ludzie! Pomocy! Proszę o nieśmiertelność do gier: GYRUS NINJA, JET SET WILLY, WRESTLING. W zamian oddam opisy gier: JUMBO JET PILOT, AGENT U.S.A., PITFALL II. Posiadam Atari 65XE.

Michał Bartkowiak, ul. Rusa 3/11, 61-245 Poznań

Poszukuję gier: SOLO FLIGHT, COMMANDO, ZORRO i SPY vs SPY. W zamian oferuję gry LAST NINJA i COMMANDO LYBIA. Posiadam komputer Commodore 64 i magnetofon DR 1535.

Przemysław Wąsik, os. Wiśłana 1/15, 08-520 Dęblin

Jestem posiadaczem Schneidera CPC 464. Jak wpisuje się nieśmiertelność na ten komputer? Proszę o nieśmiertelności i opis gry RAMBO.

Błażej Begger, ul. Gdańska 57, 89-600 Chojnice

Szukam opisów do gier: ELITE, SNODGITY, BARBARIAN 2 na C64. W zamian służę opisami gier m.in. DEFENDER OF THE CROWN.

Adam Jagiełło, ul. Trzcicka 17/4, 96-100 Skierzwice

Posiadam Spectrum 48. Poszukuję opisów do gier: GLIDER RIDER, THE GONIES, TRAP DOOR, ZORRO.

Paweł Kopyt, ul. Przejazdowa 29, 05-805 Otrębusy

Jestem użytkownikiem Spectrum 48. Poszukuję informacji o grach: TECHNO-COP, FOX FIGHT BACK. Liczę na waszą pomoc.

Michał Szopa, ul. Zakątek 7, 43-100 Tychy

Szukam opisu do gry SILENT SERVICE na Atari 65XE. W zamian oferuję inne opisy i gry.

Rafał Kunowski, ul. Kazimierska 29/17, 51-657 Wrocław

Poszukuję map i opisów do następujących gier: ASTERIX, ALCHEMIST, LIVINGSTONE, HERBERTS DUMMY RUN, MICRONAUT. Proszę także pomóc zabić smoka w GHOSTS AND GOBLINS.

Maciej Samak, ul. Przybyszewskiego, 66/13

Proszę o udostępnienie mi gier: BARBARIAN, PLATOON, KING'S QUEST, BLUE THUNDER, IKARI WARRIORS, NEMESIS i WONDER BOY w wersji na Atari 65XE.

Marcin Słowik, ul. Brzeg 17a, 34-460 Szczawnica

Odkupię gry i programy na ZX-81.

Marcin Wandzel, ul. Górnicza 21/8, 41-400 Mysłowice tel. 22-37-17

Szukam gry MONTY ON THE RUN w wersji na ZX Spectrum. W zamian oferuję inne gry.

Grzegorz Tannenberg, ul. Kokoszeki 8/42, 44-100 Gliwice

Służę nieśmiertelnościami do gier: URIDIUM, JAMES BOND 007, PAC MAN II, MISS PACMAN, SPECJAL DELIVERY, KANGAROO, MOUSE TRAP, MONTEZUMA s REVENGE, DONKEY KONG JR oraz BRUCE LEE na Atari 65XE.

Ireneusz Rykała, ul. Polna 1/18, 98-100 Łask woj. sieradzkie

Bardzo proszę o opisy do gier: WINTER GAMES, EQUI-NOX. W zamian oferuję opisy do innych gier. Posiadam Atari 65XE.

Marcin Plewik, ul. Zubrzyckiego 3/14, 44-100 Gliwice

Jak uzyskać nieśmiertelności w grach na Atari 800XL.
Jarosław Ostrowski, ul. Kędzierskiego 12/10, 01-432 Warszawa

W JOYSTICK LANDZIE

Termin „joystick” figuruje w słowniku angielsko-polskim jako „joy-stick — lotn. drążek sterowy”. Mimo niewątpliwej trafności tego określenia w stosunku do komputerowego joysticka, można odgadnąć właściwą etymologię tego słowa. Pierwsza część „joy” oznacza uradowanie, zachwyty, rozkosz itp. Druga część, „stick” to po prostu patyk, drążek, kijek. Tak więc po „komputerowym” joystick oznacza „radujący drążek”.

Takie drążki radują nas już kilka lat, od początku mikrokomputerowej ery. Już w pierwszym numerze „Bajtka” (1/85) podany był przepis, jak samemu wykonać joystick.

Co to właściwie jest i do czego służy, wyjaśnić może pierwszy lepszy ośmiolatek. Ja jednak przypomnę.

Umieszczony jednym końcem w plastikowej puszcze drążek przy wychylaniu w cztery strony włącza odpowiednie styki. Kombinacje połączeń odczytuje komputer przez kilkunastokrotny kabel zakończony dziewięciobolcowym złączem „D”. W ten sposób, poruszając drążkiem, komunikować się można z komputerem.

Ten sposób komunikacji wykorzystywany jest najczęściej przy programach graficznych oraz grach. Poza nimi spotykany jest sporadycznie.

Ze względu na to, że gry cieszą się największą popularnością, joystick stał się podstawowym urządzeniem dokupowanym do komputera (ośmiobitowego raczej, choć joysticki do IBM są ostatnio w modzie). On też najszybciej się zużywa, na wyścigi z folią klawiatury Spectrum.

Rynek joysticków na świecie jest przeogromny. Co miesiąc producenci szokują nowymi modelami. Okruchy tego szaleństwa docierają i do nas, lecz mijają bez wybuchów zachwyty.

Dwa numery wcześniej ogłosiliśmy konkurs na najwytrzymalszy joystick z oferowanych w kraju. Otrzymał już pierwsze zgłoszenia, lecz zawody jeszcze się nie zaczęły. Postanowiliśmy do istniejących już konkurencji dotrzeć próbą ognia i próbą zgniatania, gdyż przedstawione nam konstrukcje wyglądają bardzo solidnie.

Spójrzmy teraz na modele oferowane w krajach zachodnich. Można podzielić je na trzy grupy: mocnych, wytrzymałych joysticków wyposażonych w mikroprzełączniki i auto fire, do obłądnych gier w stylu Decathlon lub Space Harrier; wygodnych, niewielkich konstrukcji, do spokojnych gier labiryntowych itp. oraz delikatnych, niezwykle czułych, do programów graficznych.

Micro Blaster to jeden z pogromców kosmitów. Posiada wszystko, co potrzeba narzędziom masowego mord. Podobnie Star Fighter, lecz ten posiada wygodny uchwyt, dzięki któremu rajd przez kolejne galaktyki będzie mniej męczący.

Cruiser i jego bracia oraz Quickshot Turbo są wymarzeni jako broń TechnoCopa i RoboCopa. Przyssawki, dokładne wykonanie, kilka przycisków Fire, wygodna obudowa to niewątpliwe zalety.

Speedking i Super Professional to przykład joysticków, których nie wolno zbyt zmęczyć. Są za to niezwykle dokładne i wygodne.

Tak bawią się na Zachodzie. Jakie pomysły mają nasi konstruktorzy, przekonacie się niebawem. Z listów do redakcji wyłowiliśmy dosłownie niecodzienny pomysł kol. Marcina Klika. Zmontował on joystick na wodę. W puszcze znajduje się roztwór soli, w którym zanurzane są styki. Nie ma więc mowy o mechanicznym zużyciu joysticka. Błazniana puszka przetrzewie jednak szybko, a sól nie działa zbyt dobrze i na metalowe styki. Niemniej jednak pomysł jest bardzo ciekawy.

Obok przedstawiamy też, kilkuletni już, wytwór jednego z naszych kolegów. Joystick ten wygląda jak połączenie lejka, młota pneumatycznego i detonatora, nie mówiąc już o przyssawkach od wiatraczka elektrycznego. Spełnia on swą funkcję bardzo dobrze i służy już długo.

Joystick bez drążka, czyli jeszcze jeden oryginalnie sterowany manipulator przedstawiliśmy w „Bajtku” 12/86.

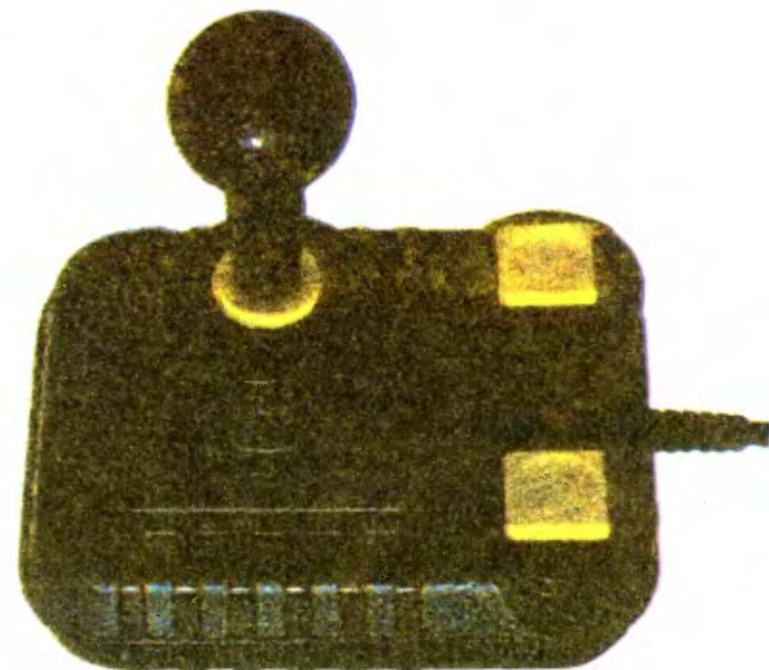
W dalszym ciągu, do końca roku kalendarzowego czekamy na zgłoszenia.

Adres redakcji: Bajtek, ul. Wspólna 61, 00-687 Warszawa

Gen Martinez



MICRO BLASTER



SUPER PROFESIONAL



QUICKSHOT TURBO

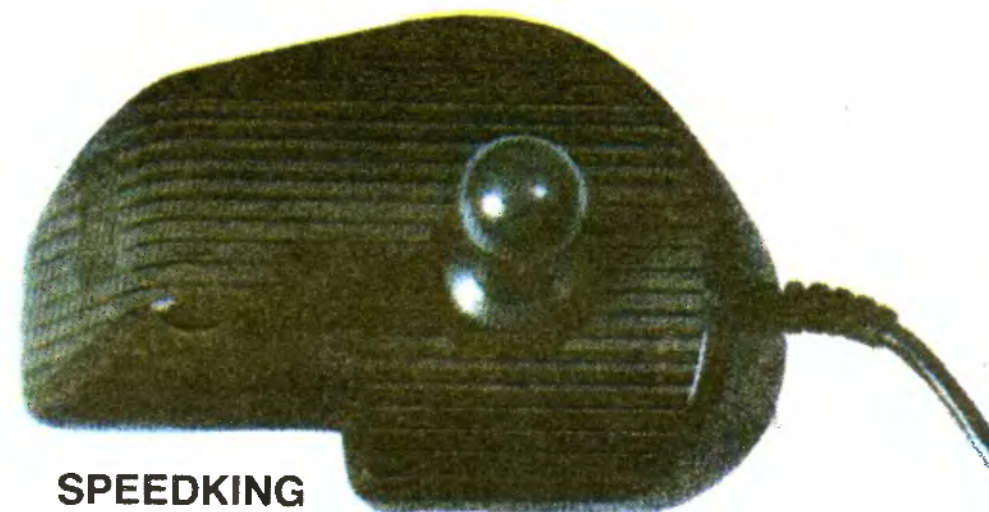
CRUISERS



JOYSTICK — DETONATOR!

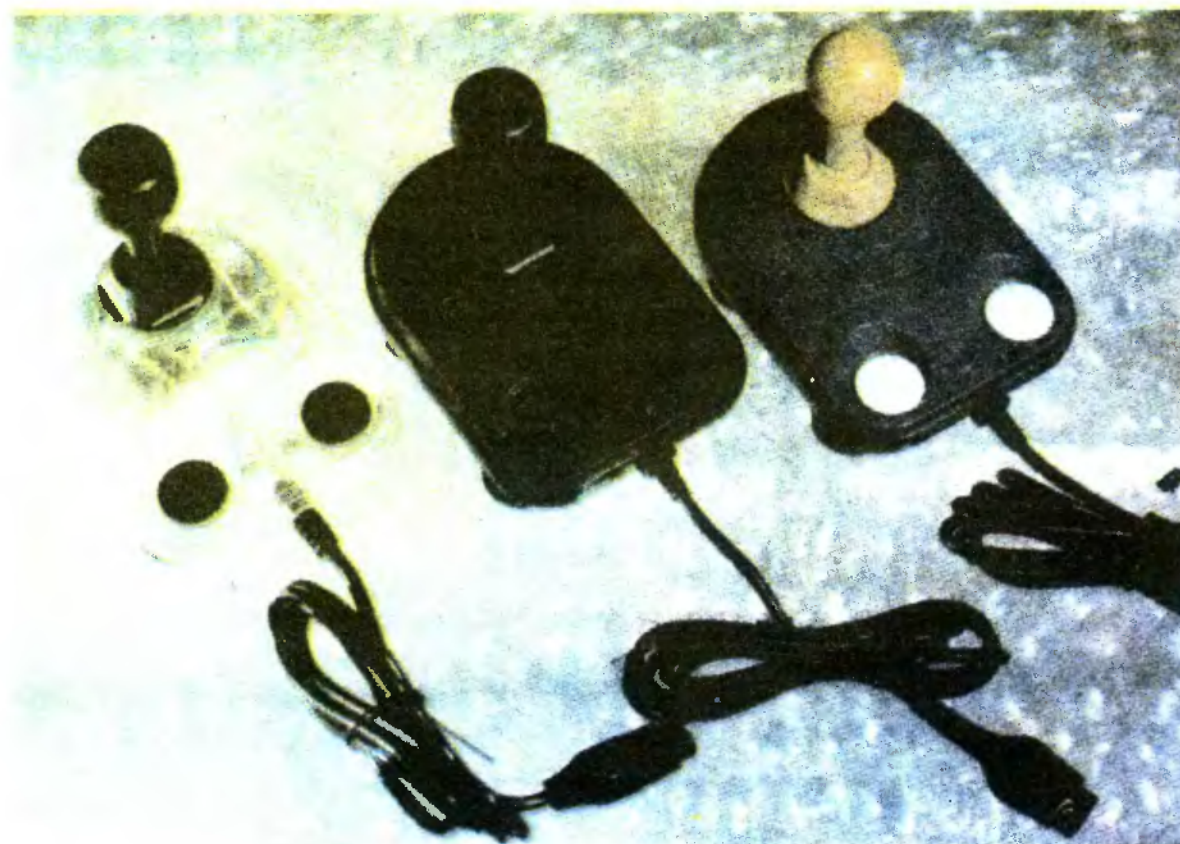


SINCLAIR



SPEEDKING

STAR FIGHTER





Sanki, narty i łyżwy zimą, zaś wrotki i deskorolka latem to najlepsza zabawa dla młodzieży. Te tendencje widać również w grach komputerowych, naśladujących już prawie każdy element naszego życia.

Nie wiem, skąd dziwaczna nazwa 720° dla gry oddającej jazdę na deskorolce. Faktem jest to, że gra ta jest bardzo dobra, zarówno pod względem graficznym, muzycznym, jak i rozrywkowym. Najlepszym tego dowodem jest przyznanie jej nagrody „CLASSIC” miesięcznika Sinclair User i oczywiście cały czas nie słabnące powodzenie u graczy.

720° to składanka czterech najpopularniejszych konkurencji, rozgrywanych za pomocą deskorolki. Są to: slalom, zjazd, skoki i dziwna konkurencja — połączenie skoków ze slalomem.

W każdej z tych konkurencji można wziąć udział, wjeżdżać na odpowiedni teren z położonego w centrum Deskorolkomiasteczka skweru. Radzę jednak zacząć od konkurencji najłatwiejszych, zarobić tam trochę pieniędzy, zakupić potrzebne do jazdy akcesoria

i dopiero wtedy udać się po mistrzostwo na trasach najtrudniejszych. Potrzebne bilety wstępu zakupić należy w budce przed bramą.

Główną dewizą początkującego deskorolkisty powinno być opanowanie. Na ukończenie objazdu całego parku i wzięcie udziału w finałowej konkurencji przeznaczony jest określony czas. Na pierwszy rzut oka jest go niewiele, wystarczy jednak na wszystko i nie warto się spieszyć.

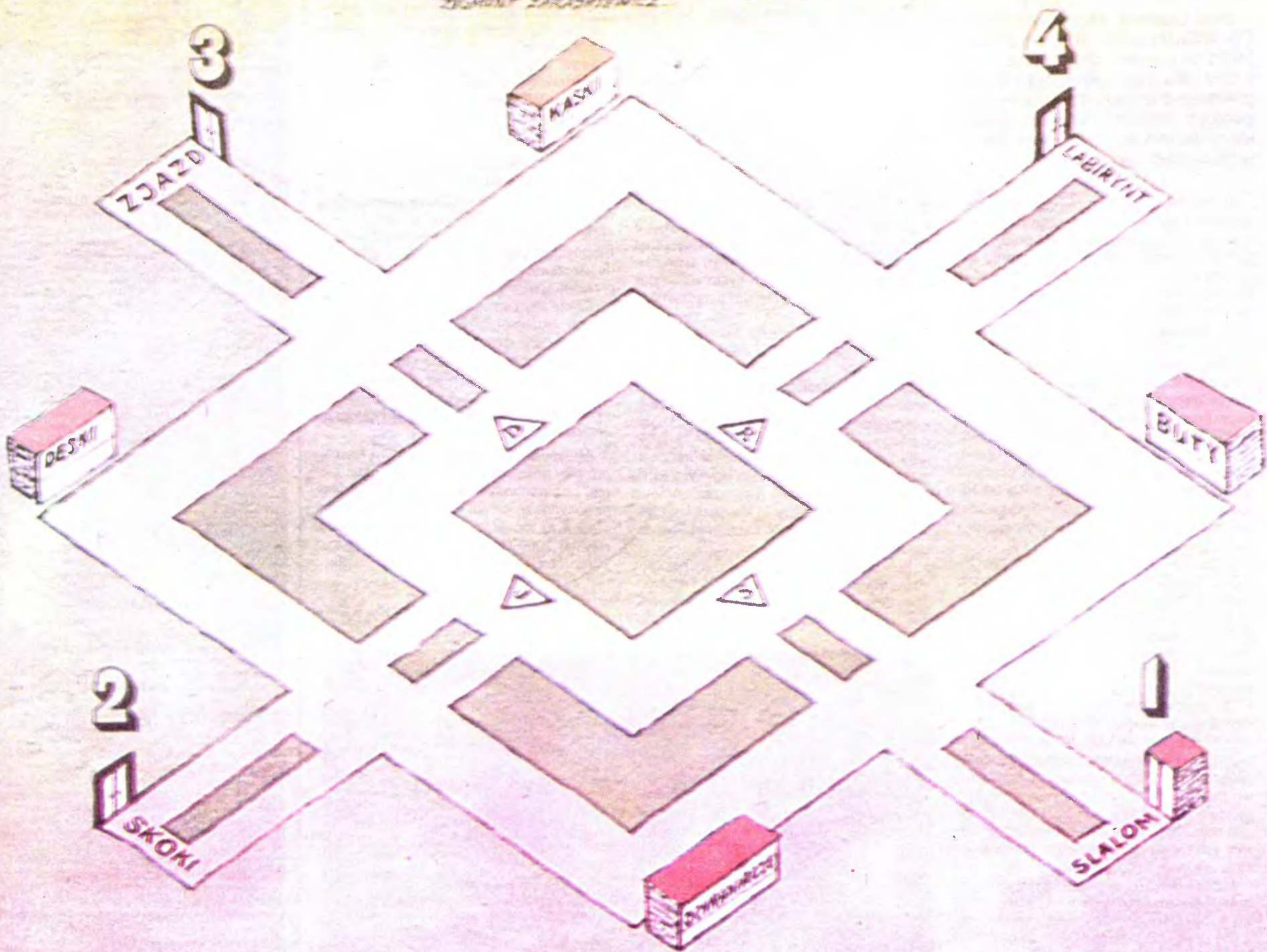
Przed udaniem się na rozgrywkę proponuję nabierać jak najwięcej punktów na ćwiczebnych trasach. Co do konkurencji, to bierzcie udział najpierw w tych, w których jesteście najlepsi.

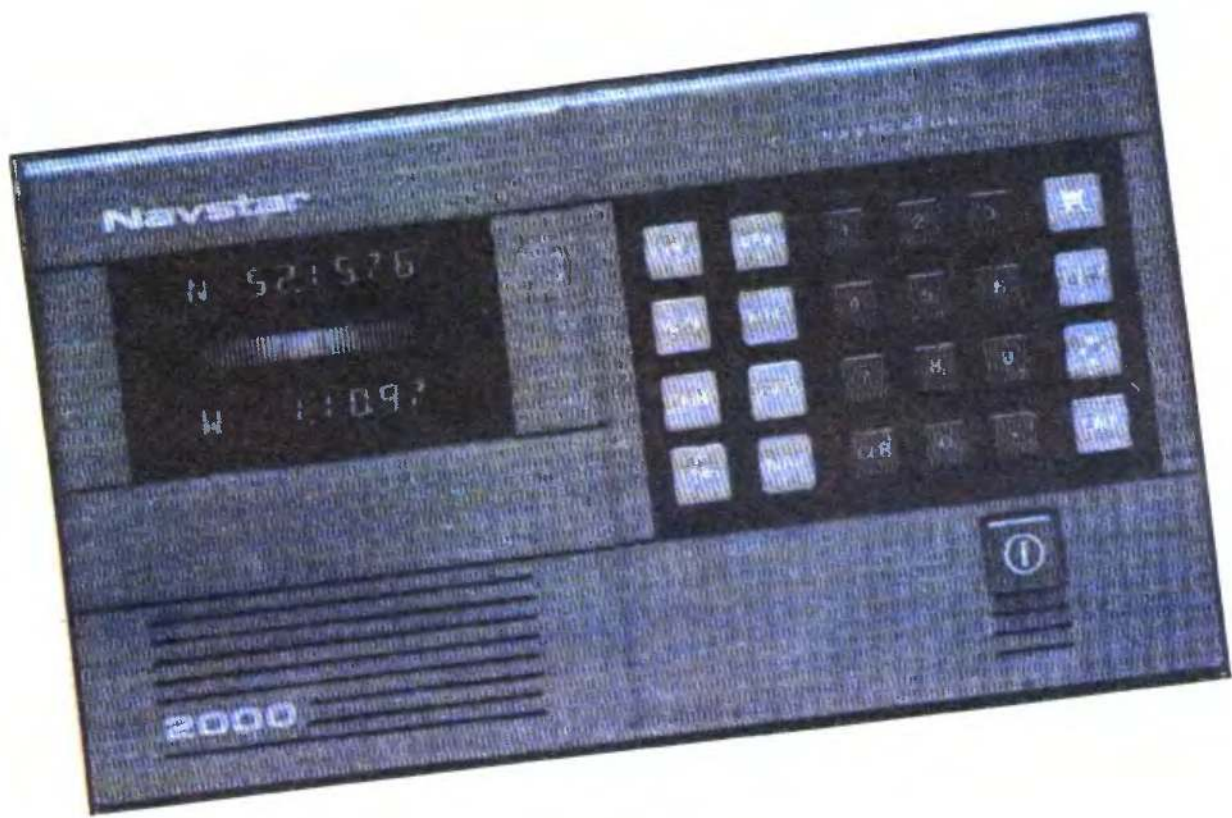
W sumie żadna z konkurencji nie jest zbyt trudna. Należy tylko rozsądnie gospodarować pieniędzmi, biletami i własnymi umiejętnościami.

Firma: U.S. Gold.
Komputer: ZX Spectrum 48, Commodore 64, Amstrad CPC, Atari ST, IBM PC

Gen

720°





NAWIGATORZY NA EMERYTURĘ?

Jest ciemna, sztormowa noc. Po wzburzonym morzu płynie jacht z trudem przebijając się przez fale. Na zalewanym wodą pokładzie oficer wachtowy wypatruje świateł latarni morskich. Jedno jest! Po kilku minutach widać także drugie.

Według błysków i przerw, czyli charakterystyki światła określona zostaje nazwa latarni. Teraz najtrudniejsza część pracy. Stojąc na pokładzie jachtu skaczącego po falach jak piłka trzeba przy pomocy kompasu i namiernika określić kierunek, w którym widoczne są latarnie. Aby dokładnie wyznaczyć pozycję jachtu na morzu kierunek musi być zmierzony z dokładnością do 2°. W takich warunkach jest to duża sztuka.

Następnie oficer schodzi pod pokład i oblicza błędy kompasu oraz odchylenie północy magnetycznej od geograficznej. Po uwzględnieniu poprawek zmierzone namiary rysuje na mapie i dopiero teraz wiadomo, gdzie znajduje się jacht. Cała ta operacja trwa od 5 do 15 minut, w zależności od doświadczenia oficera i panujących warunków atmosferycznych, a musi być wykonywana co najmniej raz na godzinę.

A co robić, gdy jacht płynie po pełnym morzu, z dala od brzegu? Trzeba określić przebytą drogę oraz przybliżony kierunek i na tej podstawie obliczyć pozycję. Oczywiście, taki sposób (zwany zliczeniem) nie może być dokładny. Błąd określenia pozycji zliczonej dochodzi czasami do kilku mil morskich (1 Mm = 1852 m).

Od dokładnego określenia pozycji na morzu zależy bezpieczeństwo statku i znajdujących się na nim ludzi. Wraz z rozwojem techniki wymyślano więc różne urządzenia mające ułatwić pracę nawigatora. Początkowo były to radiolatarnie wysyłające sygnały radiowe zamiast świetlnych. Dzięki nim można obliczyć pozycję nawet wtedy, gdy latarnia jest ukryta za horyzontem lub niewidoczna we mgle.

Przez połączenie kilku radiostacji w zsynchronizowany system tworzy się łańcuch. Mierząc specjalnym odbiornikiem różnicę czasu lub fazy nadawanych przez nie sygnałów można określić odległość od poszczególnych nadajników i na tej podstawie wyznaczyć pozycję na morzu. Takich systemów radionawigacyjnych powstało kilka, a najbardziej znanymi są Decca, Loran i Omega. Podstawową wadą tych systemów była konieczność posiadania specjalnych map oraz specjalnych odbiorników, dużych i drogich. Instalowano je więc na statkach,

zaś żeglarze nadal nawigowali tak, jak przed trzema wiekami.

Technika wkroczyła jednak również do żeglarstwa i to także na polskie jachty. Opisana na początku scena wygląda teraz zupełnie inaczej. Oficer wachtowy naciska przycisk małej szarozielonej skrzynki umocowanej nad stołem nawigacyjnym, odczytuje pozycję i nanosi ją na mapę. Określenie pozycji w ten sposób trwa niecałą minutę i jest możliwe w dowolnej chwili. Ta cudowna skrzynka to Navstar 2000 Navigator.

Navstar 2000 jest produkowany w dwóch odmianach: do odbioru sygnałów systemu Decca (2000-D) oraz do odbioru sygnałów z satelitów nawigacyjnych systemu Satnav (2000-S). W obu wersjach urządzenie to składa się z odbiornika i małej, niespełna 40-centymetrowej anteny. Ponadto możliwe jest uzupełnienie zestawu o dodatkowe wyposażenie.

Odbiornik Navstar 2000-D jest w rzeczywistości połączeniem radia z komputerem, a jego pracą steruje procesor Z80. Działanie jego jest w pełni automatyczne, a dzięki wbudowanej baterii raz wprowadzone parametry pozostają w pamięci po wyłączeniu zasilania. Obsługiwany jest przez gumową klawiaturę zawierającą 24 klawisze i umożliwiającą zarówno wybór funkcji, jak i wprowadzanie danych.

Po uruchomieniu odbiornika najpierw przeprowadzana jest kontrola jego elementów. Gdy cały system jest sprawny, operator może wprowadzić niezbędne parametry.

datę, godzinę, jednostki (mile lub kilometry), poprawki kompasu itd. oraz wybrać język, w którym będą wyświetlane komunikaty: angielski, niemiecki lub francuski. Po zakończeniu tego etapu odbiornik sam wyszukuje sygnał najbliższego łańcucha Decca, a następnie określa pozycję. Po upływie czterech minut, niezbędnych do zmierzenia przemieszczenia jachtu, można także odczytać prędkość i kurs jachtu.

Navstar 2000-D ma ponadto wiele innych funkcji. Potrafi zapamiętać 99 punktów o podanych współrzędnych i oblicza kierunek i odległość w jakiej się znajdują. Punkty te można połączyć w łańcuchy (trasy). Takich tras może być dziewięć, a każda z nich może składać się z 25 punktów. Po wybraniu trasy Navstar pokazuje kierunek i odległość do kolejnego punktu, a po jego osiągnięciu do następnego. Zbliżenie się jachtu do zapamiętanego punktu na odległość mniejszą od 3 Mm jest sygnalizowane alarmem dźwiękowym.

Specjalny przycisk klawiatury powoduje zapamiętanie aktualnej pozycji jachtu jako punktu o numerze 0. Jest to bardzo użyteczne w przypadku wypadnięcia za burtę człowieka, gdyż umożliwia powrót do tego miejsca i odnalezienie człowieka nawet w bardzo niekorzystnych warunkach.

Do prowadzenia nawigacji przy pomocy odbiornika Navstar 2000-D niezbędne jest posiadanie map akwenu, po której się pływa i nanoszenie na nie aktualnej pozycji jachtu. Jednak i to można wyeliminować stosując dodatkowe urządzenie — Navstar 2000-P Chart Plotter.

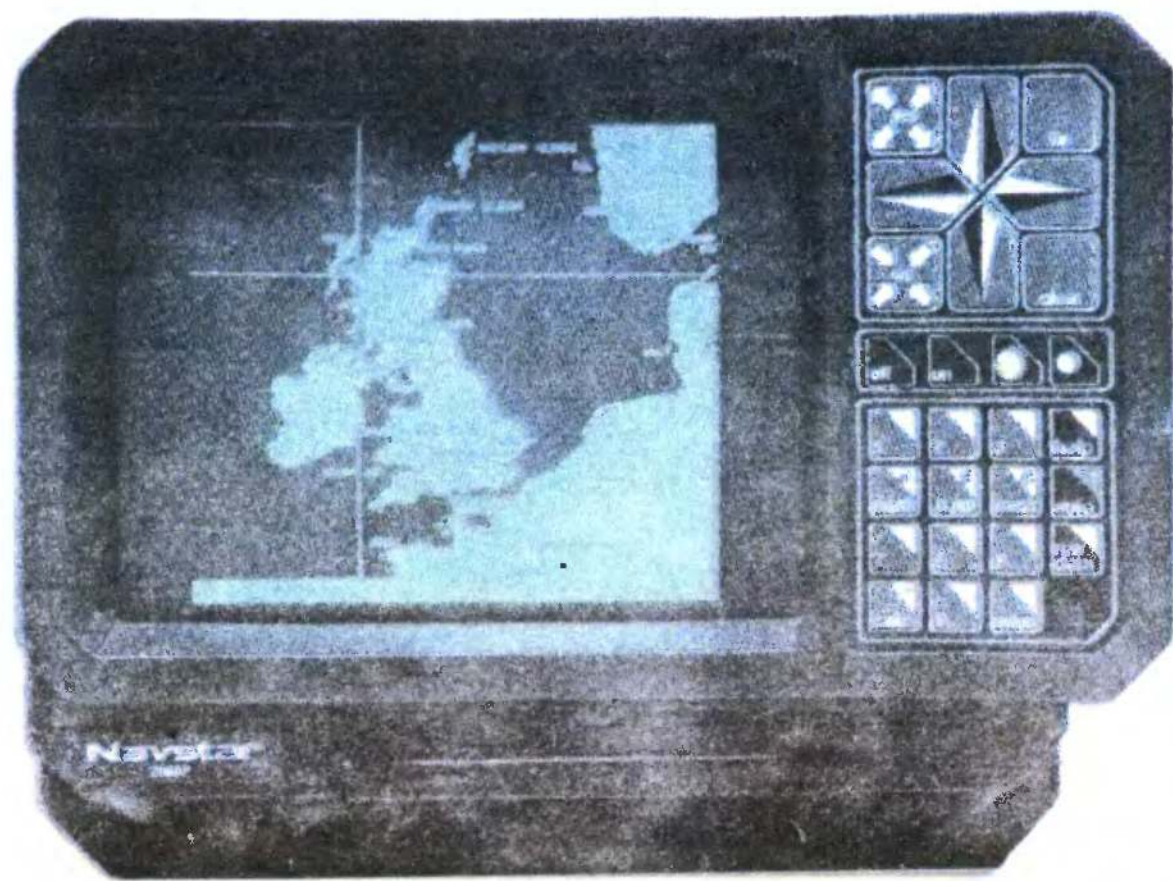
Ploter Navstar wyposażony jest w ośmiocalowy ekran o wysokiej rozdzielczości, który pozwala na przedstawienie obszaru o rozmiarach d 0,5 Mm do 4000 Mm. Jednemu punktowi ekranu odpowiada w rzeczywistości odległość od 4 metrów do 32 kilometrów. Wybrany fragment mapy można więc dowolnie powiększać i zmniejszać w takim zakresie. Wyświetlana mapa przesuwa się płynnie po ekranie w miarę ruchu jachtu.

Mapy różnych rejonów świata wprowadzane są na modułach umieszczanych w specjalnym gnieździe plotera. Jedna mapa może zawierać do 350 tys. punktów kartograficznych. Moduł mapy trzeba oczywiście zakupić oddzielnie, lecz wraz z ploterem dostarczana jest zawsze jedna, wybrana przez nabywcę mapa.

Także i tu istnieje wiele funkcji pomocniczych. Można obliczyć i wyświetlić przebytą drogę, obliczyć kierunek i odległość do dowolnego punktu mapy oraz czas niezbędny do jej przebycia i zużycie paliwa podczas żeglugi na silniku. Navstar 2000-D posiada także możliwość sterowania autopilotem i współpracy z radarem.

Zastosowanie całego systemu Navstar pozwala więc na prowadzenie żeglugi niemal bez wychodzenia z ciepłego i przytulnego wnętrza jachtu. A wszystko to jest dostępne za stosunkowo nieduże pieniądze: Navstar 2000-D kosztuje 1500 marek, a za 2000-P trzeba zapłacić 2500 DM.

Wojciech Zientara



Jednym z najciekawszych zastosowań komputerów, przynajmniej dla postronnych obserwatorów, nie żyjących komputerami, jest niewątpliwie grafika. Tym też należy tłumaczyć popularność tego tematu i dużą ilość książek i artykułów na ten temat. Jak to jednak często bywa, ilość nie idzie w parze z jakością — do tej pory nie wydano u nas książki, która połączyłaby w sobie przystępny dla nieinformatyka opis metod stosowanych przy tworzeniu grafiki z gotowymi programami do „wstukiwania”. Dziurę tę próbowali w zeszłym roku zapełnić Wydawnictwa Naukowo-Techniczne, przy pomocy tłumaczonej z jęz. angielskiego książki „Grafika mikrokomputerowa”. Niestety — bezskutecznie.

Oryginalnie książka ta była wydana w Stanach Zjednoczonych w 1983 r. Od tamtego czasu minęło sześć lat, co w tak szybko zmieniającej się dziedzinie jak komputery i ich oprogramowanie jest niemal całą epoką. Niestety widać to na każdej niemal stronie — książka jest po prostu nudna, bo wszystko co się w niej znajduje jest bardzo dobrze znane — nawet tym którzy nie mają z komputerami do czynienia na codzień. (Komputery tak się już zdomowały w naszym życiu, że pewnych rzeczy z nimi nie kojarzymy, choć powinniśmy). Ponadto brak (bo i skąd miały by się znaleźć) najmniejszych nawet wzmianek o tak istotnych dzisiaj koncepcjach jak ikony, pull-up menu i mysz, mimo, że książka zawiera rozdział poświęcony „konwersacji” między użytkownikiem i programem.

Zostawmy narazie braki i zajmijmy się tym co jest. Pierwsze trzy rozdziały poświęcone są wprowadzeniu w zagadnienie — jest trochę informacji o sprzęcie (zupełnie u nas nieznanym), o zastosowaniach i instrukcjach graficznych, z których korzysta się w dalszej części książki. Następnie omówione zostały wykresy słupkowe i sposoby rysowania podstawowych figur geometrycznych — prostokątów, kolek i niektórych krzywych — np. sinusoidy. Szósty rozdział zawiera informacje o przesuwaniu, skalowaniu i obracaniu rysunków przy pomocy przeliczania współrzędnych końców odcinków z których obraz został skonstruowany. Siódmy rozdział to animacja — przesuwanie punktów i odcinków po ekranie. Ośmy rozdział mówi o oknach — takich jakie występują w programach graficznych i pozwalają na powiększanie i przesuwanie fragmentów większych rysunków. Następne dwa rozdziały poświęcone są trzeciemu wymiarowi, czyli rozdziały zawierają propozycje zastosowań tego, o czym wcześniej była mowa — w tym między innymi zastosowań tak wspaniałych, jak graficzna prezentacja budżetu domowego czy graficzna kontrola składu spożywanego posiłków.

Wymienione powyżej zagadnienia mogą sugerować, że książka mimo swego wieku, jest ciekawa — niestety, wiek nie jest jej jedyną wadą. Trzy czwarte książki to wydruki programów — napisanych w BASIC-u i zawierających błędy składni (np. program 5.6, linia 500). Zastosowane w nich algorytmy praktycznie nie są opisane w tekście, i ich zrozumienie wymaga przegrzywania się przez wydruki — wprawdzie opatrzone komentarzami, ale to jednak trochę za mało. Tych, którzy liczą na wstukiwanie gotowych programów i zastosowanie ich do własnych celów muszą ostrzec — programy są mało elastyczne i przystosowane do bardzo konkretnych i wąskich zadań, poza tym brak gwarancji że nie zawierają innych błędów niż proste do wykrycia — takie jak prakujący cudzysłowy.

Podsumowując — nie polecam tej książki nikomu, zwłaszcza że nie jest zbyt tania. Nie sądzę, by komuś udało się z niej naprawdę skorzystać i dużo się nauczyć. Nie pozostaje nam nic innego jak dalej czekać — może w końcu pojawi się taka książka, która wypełni włącznie pustą dziurę, której nie udało się zatkać.

(mb)

Donald Hearn, M. Pauline Baker, „Grafika mikrokomputerowa”, Wydawnictwa Naukowo-Techniczne, Warszawa 1988, wyd. 1, nakład 30000, cena 1100 zł.

KOMUNIKACJA POTRZEBUJE PAMIĘCI

Tym razem proponuję, abymy rozważyli wymagania pamięciowe procesu komunikacyjnego, w którym brak synchronizacji i różnice w prędkościach pracy wymagają dodatkowego przechowywania przesyłanych danych.

Zajmijmy się problemem obsługi łącza komunikacyjnego współpracującego z naszym komputerem. Nadchodzące za pośrednictwem tego łącza komunikaty są obrabiane przez program (nazwijmy go np. ODBIORCA). Sposób obróbki nie ma tu większego znaczenia, dla ustalenia uwagi możemy przyjąć, że program przegląda komunikaty i w zależności od treści rozsyła je dalej, do abonentów sieci lokalnej. Ważne jest dla dalszej analizy, że komunikat raz przetworzony nie musi być dalej przechowywany — miejsce w którym został zapisany przy odbiorze z łącza może być użyte do zapisu nowych danych. Ważne jest również, że ODBIORCA nie może odmówić przyjmowania komunikatu, nie może wydać łączu polecenia „CZEKAJ” — działa ono niezależnie od tego czy ktoś jest gotów do odbioru czy też nie. Przetwarzanie komunikatu zajmuje trochę czasu, niestety wyraźnie więcej niż jego przesłanie. Stąd wynika podstawowy problem: łącze jest w stanie dostarczać dużo więcej danych niż program ODBIORCA przetworzyć. Ponieważ jednak komunikatów w ciągu doby napływa niewiele, więc program jest w stanie poradzić sobie z nimi, jeśli tylko są równomiernie rozłożone w czasie. Niestety, zwykle w życiu jest tak, że komunikaty napływają losowo, może więc zdarzyć się kilka napływających bezpośrednio jeden za drugim. Co wtedy? Dopóki program jest zajęty nie może pobierać następnego, tak więc część danych nie zostanie odebrana i jest stracona. Jeśli chcemy tego uniknąć, to rozwiązanie jest proste: ODBIORCA nie może współpracować bezpośrednio z łączem. Szybki program (nazwijmy go KOMUNIK) będzie odbierał dane z łącza i umieszczał je w pamięci, skąd ODBIORCA będzie mógł je pobierać w miarę wolnych mocy przerobowych, przez nikogo nie poganiany. Oczywiście musimy do realizacji tego rozwiązania wydzielić odpowiedni obszar pamięci, w którym komunikaty będą zapisywane i składowane. Odpowiedni to znaczy przede wszystkim odpowiednio duży — jego wielkość należy dobrać w zależności od maksymalnej i średniej długości komunikatu, szybkości pracy łącza, szybkości pracy programu ODBIORCA, i wreszcie od liczby oczekiwanych komunikatów. Nie chcemy dopuścić do sytuacji, że łącze zaczyna „wyrzucać z siebie” komunikat, a my nie mamy wolnego miejsca w pamięci i nie możemy komunikatu zapisać — jego treść byłaby stracona. Taki obszar pamięci, przeznaczony do chwilowego przechowywania informacji, która nie może być od razu przetworzona z powodu różnicy w prędkości pracy, nazywamy buforem (patrz „Bajtek” 2/88).

Ale samo wydzielenie w pamięci obszaru przeznaczonego na bufor nie wystarcza. Musimy jeszcze zaprojektować mechanizm zapewniający dostęp do tego bufora. Konkretnie, gdy pojawia się nowy komunikat to program KOMUNIK musi wiedzieć w którym miejscu bufora go zapisać i czy jest dosyć wolnego miejsca, zaś program ODBIORCA musi wiedzieć skąd pobrać następny komunikat do przetwarzania. Podkreślam, że to wszystkie informacje, które

są potrzebne, nie jest np. potrzebna żadna „mapa” bufora, zawierająca informacje o położeniu każdej porcji danych.

Spróbujmy zorganizować dostęp do pamięci tak, aby w prosty sposób zaspokoić potrzeby obu programów. Przyjmujemy, że komunikaty mogą być różnej długości i każdy zakończony jest znacznikiem rozróżnianym przez programy. Nasze programy umieszczają i pobierają dane w takiej kolejności w jakiej dostarcza je łącze. Jako bufor przydzielamy spójny obszar pamięci o długości N komórek, do którego (dla wygody zapisu) będziemy odwoływać się jak do tablicy BUF(1..N).

Każdy komunikat wpisujemy w kolejnych, następujących bezpośrednio po sobie komórkach, od pierwszego wolnego miejsca poczynając, jeden znak w jednej komórce. Na początku cała tablica jest wolna, więc pierwszą porcję danych wpisujemy od komórki nr 1, czyli w komórkach: BUF(1), BUF(2), BUF(3), itd. Jeśli jego długość wynosiła np. dziesięć znaków, to zajęte zostały pozycje od BUF(1) do BUF(10) włącznie i oczywiście następny zapis będziemy rozpoczynali od BUF(11). Jedyne co musimy zapamiętać po zapisaniu porcji danych to numer pierwszej wolnej komórki — tej od której zaczniemy następny. Przeznaczmy na to zmienną całkowitą o nazwie WOLNY. Początkowo ustalamy WOLNY=1 (tablica pusta), zaś każde wpisanie komunikatu o długości k znaków powoduje zwiększenie

$WOLNY = WOLNY + k$

Wpisanie, realizowane w programie KOMUNIK, może wyglądać tak:

Dopóki nie koniec komunikatu wykonuj:

BUF(WOLNY) = następny znak z łącza;

WOLNY = WOLNY + 1;

Koniec dopóki.

Zanim zastanowimy się, czy o niczym nie zapomnieliśmy, popatrzmy na przykładową zawartość bufora w czasie pracy, po wpisaniu trzech komunikatów: K1, K2, K3, odpowiednio o długości 4, 7 i 3 znaków. Poszczególne komunikaty zaznaczymy różnymi kolorami. Wartość zmiennej WOLNY przedstawimy w postaci strzałki, pokazującej komórkę o numerze zapisanym w tej zmiennej, N=25.



Pobieranie z bufora będzie wyglądało analogicznie. Poczynając od pierwszego znaku komunikatu pobieramy znak po znaku z kolejnych komórek pamięci. Musimy zawsze znać miejsce (numer komórki), w którym jest zapisany pierwszy znak oczekującego na pobranie komunikatu. Pobierając dane przesuwamy go zawsze na pierwszą nie pobraną komórkę. Zapamiętywać go będziemy w zmiennej NASTĘPNY. Początkowo, gdy tablica jest pusta, NASTĘPNY ustawimy na 1, w miarę czytania danych z BUF będziemy posuwać go do przodu. Popatrzmy na ustawienie wskaźników, gdy z tablicy pobierzemy komunikat K1:



a następnie dopiszemy jeszcze K4 i K5 długości 2 i 4, oraz pobierzemy K2 i K3:



Na powyższych rysunkach widać (mam nadzieję), nie tylko omówioną zasadę gospodarowania pamięcią, ale i pewne prawidłowości wynikające z tej zasady. Otóż,

$NASTĘPNY \leq WOLNY$

przy czym równość zachodzi wtedy i tylko wtedy gdy bufor jest pusty — żadne dane nie oczekują na przetwarzanie. Gdy

$NASTĘPNY < WOLNY$

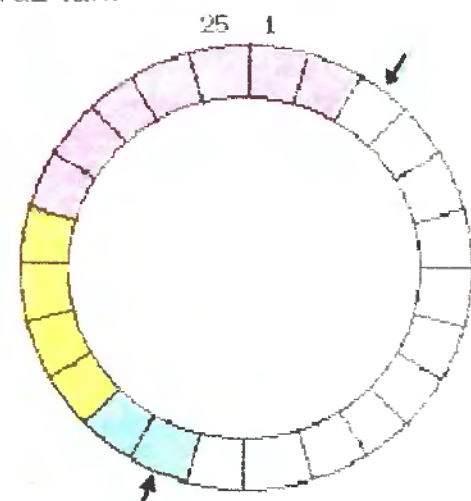
to komunikaty zajmują komórki o numerach od NASTĘPNY do WOLNY-1 włącznie. Pozostałe komórki są wolne. Ich ilość łatwo obliczyć wzorem:

$IŁOŚĆ = N - (WOLNY - NASTĘPNY)$.

Potrąfimy więc właściwie wszystko co nam jest potrzebne: wpisywać i pobierać dane, określić ilość wolnego miejsca. Niestety, pominieliśmy dotąd jeden ważny szczegół techniczny. Spróbujmy do tablicy z ostatniego rysunku dopisać komunikat K6, o długości 7 znaków. Ilość wolnego miejsca, obliczona wzorem wynosi $25 - (21 - 15) = 19$, ale nasz prosty program padnie, gdyż spróbuje pisać dane poza tablicą:



Problemu nie byłoby, gdybyśmy potrafili „skleić” koniec tablicy z jej początkiem i zamiast mówić o „wpisywaniu do komórek o rosnących numerach” mówili np. o „wpisywaniu do kolejnych z prawej strony”. Jedyne ograniczeniem jest abyśmy, przesuwając przy wpisywaniu strzałką WOLNY nie przekroczyli strzałki NASTĘPNY. Nasz bufor wygląda teraz tak:



Rola strzałki WOLNY i NASTĘPNY pozostała bez zmian i teraz już wszystko jest pięknie: dwa procesy, wpisywania i pobierania danych „gonią się” po okręgu, a cała informacja potrzebna do właściwej gospodarki pamięcią mieści się w dwóch zmiennych. Właśnie takie spojrzenie na gospodarkę pamięcią dało nazwę BUFOR CYRKULARNY lub BUFOR CYKLICZNY, często używaną na określenie opisanej struktury.

Aby szczęście było pełne musimy jeszcze zrealizować bufor cyrkularny w naszej zwykłej tablicy BUF(1..N). Nie jest to wcale zadanie trudne, wystarczy rozszerzyć kod programu o instrukcje sprawdzające dla każdego ze wskaźników, czy nie wykracza on za koniec tablicy, i ew. przestawić wartość na początek. Np. przy wpisywaniu komunikatu:

Dopóki nie koniec komunikatu wykonuj:

BUF(WOLNY) = następny znak z łącza;

WOLNY = WOLNY + 1;

IF WOLNY > N THEN WOLNY = 1

Koniec dopóki.

Na zakończenie jedno ważne zastrzeżenie. Do naszego bufora odwołują się dwa różne procesy. Oprócz gospodarki pamięcią istnieje więc drugi poważny problem, synchronizacji dostępu do tablicy, tak by programy ODBIORCA i KOMUNIK nie zakłócały nawzajem swojej pracy. Jednak jego rozwiązanie nie zmieści się w naszych rozważaniach o strukturach danych.

Andrzej Krul

PRZETWARZANIE RÓWNOLEGŁE

Zadanie 1.

Jeśli jeden robotnik kopie duży dół w ciągu dwóch dni, ile czasu taki sam dół będzie kopac dwóch robotników?

Zadanie 2.

Jeżeli jeden komputer wykonuje pewne obliczenia w ciągu jednej godziny, ile czasu te same obliczenia będą wykonywać dwa komputery?

Rozwiązaliście oba zadania? Sprawdźmy, czy wyszło Wam to samo co mi. W pierwszym przypadku odpowiedź jest oczywista — dół powinien być wykopany w ciągu jednego dnia (chyba że obaj panowie razem pójda na piwo — jednemu samemu mogło się nie chcieć). Drugie zadanie — widzę że sprawa jest bardziej skomplikowana, niektórzy twierdzą, że godzinę, inni, że pół. Żeby ułatwić sobie poszukiwanie prawidłowego rozwiązania zrobimy proste doświadczenie myślowe. Niech obliczenia o których mowa w zadaniu polegają na dodaniu do siebie dwóch liczb — na przykład e i π , ale z dokładnością do kilkuset cyfr po przecinku. Weźmy małe Spectrum, wprowadźmy do jego pamięci odpowiedni program, który wykona obliczenia i uruchomimy go. Zanim skończą się obliczenia minie (powiedzmy) godzina. Weźmy teraz drugie Spectrum, postawmy obok pierwszego wprowadźmy ten sam program i uruchomimy oba programy naraz. I co? Zamiast pół godziny znowu musimy czekać całą. Czy wszyscy rozumieją?

Coś tu nie gra. Absurdalność uzyskanego w ten sposób wyniku jest oczywista, ale znalezienie błędu nie jest takie proste. Poszukajmy go — to nam pomoże wykorzystać fakt posiadania dwóch komputerów do przyspieszenia obliczeń.

Odpowiedzmy na pytanie, czy rzeczywiście

oba komputery wykonywały te same obliczenia? Otóż nie, choć może się to wydawać niezrozumiałe. Jeśli się jednak głębiej nad tym zastanowić, okaże się, że oba komputery wykonywały takie same obliczenia, a nie te same. Zarówno liczba e jak i π były liczone po dwa razy — przez każdy komputer osobno. Tymczasem robotnicy kopający ten sam dół nie wbijają łopaty po dwa razy w to samo miejsce, bo i po co.

Teraz już wiemy, jak przyspieszyć nasze rachunki. Możemy nasz program rozbić na dwa mniejsze — jeden liczący liczbę e , drugi liczący π . Każdy z nich możemy uruchomić na jednym komputerze, a kiedy oba skończą uruchomimy trzeci program (na którymkolwiek komputerze), który dokona sumowania. Oczywiście żeby sumowanie było możliwe, trzeba najpierw przekazać wyniki pracy jednego komputera do drugiego — w tym celu najlepiej jest je połączyć w sieć. Tym razem na pewno wynik uzyskamy po krótszym czasie, choć wcale nie tak krótkim jak jest to możliwe. Kiedy bowiem jeden komputer skończy liczyć, musi czekać na koniec pracy drugiego beczynnie stojąc. Można oczywiście napisać programy tak, żeby pracę podzielić sprawiedliwie, to znaczy że ich wykonywanie będzie trwało mniej więcej tyle samo — wcale nie musi być tak, że jeden komputer liczy cyfry e a drugi π , zadanie może się udać podzielić inaczej, ale jest to bardzo trudne zadanie. Taki sposób postępowania zawiedzie jednak zupełnie gdy zakupimy trzeci komputer — będziemy musieli cały program pisać od nowa (a właściwie trzy programy).

Przyjrzyjmy się przez chwilę pracy naszych wzorowych robotników. Ponieważ mogą się między sobą porozumiewać, czyli wymieniać informacje, mogą odpowiednio zorganizować swoje działanie. Jeżeli każemy im wykopać dół oszalować deskami, nie muszą wcale szalować każdy jednej strony dołu (e i π). Jeden z nich może przynieść deski i gwoździe, a drugi robić resztę.

Tak doszliśmy do drugiego sposobu przyspieszania obliczeń. Jest on bardzo trudny w realizacji, ale daje znacznie lepsze rezultaty. Wyposażmy oba komputery w program zarządzający nimi, czyli w odpo-

wiedni system operacyjny. System powinien działać tak — jeżeli jeden z komputerów sygnalizuje że skończył swoje zadanie i stoi beczynnie, trzeba znaleźć w tym co zostało do zrobienia odpowiedni kawałek i przez łącze wysłać fragment programu z danymi. Możliwe jest też inne rozwiązanie — w momencie uruchamiania programu system dzieli go na części które będą mogły pracować równolegle. W ten sposób można ze sobą połączyć kilka komputerów, nie tylko dwa, więc znika problem pisania programu od nowa po zakupie kolejnego komputera. Pomysł jest bardzo obiecujący, ale zwróćcie uwagę na użyte przed chwilą słowo „odpowiedni”. Jeżeli do wykonania obliczeń następnymi instrukcjami programu są potrzebne rezultaty obliczeń prowadzonych właśnie w tej chwili, nie można ich wykonywać równolegle. W tym leży problem — system operacyjny musi znać powiązania między różnymi częściami programu i musi potrafić tak dzielić zadanie, żeby nic nie zginięło i wszystkie obliczenia zostały wykonane we właściwej kolejności. Nie jest to wcale łatwe, choć nie niemożliwe.

A jak to wygląda w praktyce? Oba sposoby prowadzenia obliczeń na kilku komputerach naraz, czyli przetwarzania równoległego, są już stosowane. Oczywiście żeby móc z nich korzystać potrzebne są nowe narzędzia, jakim są nowe języki programowania. Pierwszym takim językiem był Occam, lecz — ze względu na swoje braki — nie stał się standardem. Bardzo szybko pojawiły się kompilatory innych języków — dobrze znanych — lecz „poprawionych”, tak by można było korzystać z możliwości oferowanych przez przetwarzanie równoległe. Pojawiły się więc kompilatory Parallel C, Parallel Pascal i Parallel Fortran. Asembler, mimo że istnieje, nie jest prawie wcale stosowany — tylko tam gdzie bardzo zależy nam na szybkości. Trwają prace nad programami, które będą potrafiły podzielić (dysponując tekstem źródłowym) istniejące już programy na takie części, które dadzą się wykonywać równoległe. Jest również przygotowywany system operacyjny o którym była mowa — będzie się on nazywać Mercury.

Przetwarzanie równoległe, rozwijające się dziś bardzo szybko, ma za sobą długą historię. Jego najlepszy okres przypada — paradoksalnie — na lata trzydzieste naszego wieku. Poważne obliczenia numeryczne były wówczas przeprowadzane przy pomocy wyposażonych w mechaniczne sumatory rachmistrzów, dokonujących obliczeń cząstkowych i przekazujących sobie nawzajem wyniki według ustalonego wcześniej schematu. W zależności od rangi problemu (i możliwości finansowych sponsora), obliczenia prowadziło od kilku do kilkuset osób. Zainteresowanie przetwarzaniem równoległym spadało pod wpływem komputerów, które liczyły znacznie szybciej i dokładniej, wykonując obliczenia w sposób sekwencyjny, czyli po kolei, jedna operacja za drugą. Wprawdzie już w latach sześćdziesiątych nastąpiły nieśmiało próby budowy komputerów o architekturze równoległej, ale ich szybki rozwój obserwujemy dopiero ostatnio. Zadziałało tu jedno z praw Murphy'ego: każdy program rozrasta się aż komputer z którego korzysta programista przestaje być wystarczający. Innymi słowy — nie ma tak dużego komputera, żeby nie można było wymyślić zadania dla którego komputer ten będzie za mały. Dlatego też nigdy nie doczekamy się końca wyścigu w stronę Mega — bajtów, herców itd. Przetwarzanie równoległe jest jednym z etapów tego wyścigu — na tyle ciekawym, że warto mu poświęcić kilka chwil.

Czy kupowanie kilku komputerów po to, żeby móc coś szybciej policzyć ma sens? Czy nie lepiej kupić jeden duży i bardzo szybki? Oczywiście jest stawianie obok siebie szesnastu Atari mija się z celem, ale jeśli tą cegiełką z której będziemy korzystać jest transputer, to sytuacja wygląda zupełnie inaczej. Zajmiemy się tą sprawą za miesiąc.

Marcin Borkowski

Powyższy tekst powstał na podstawie materiałów nadesłanych nam przez europejski oddział firmy MicroWay, za co serdecznie dziękujemy.



Wrześniowy numer Sinclair User oprócz comiesięcznej porcji opisów nowych gier zawiera również opis nowego komputera firmy Miles Gordon Technology, zapowiadane już od lutego.

SAM COUPE, czyli nowy twór MGT (znanej m.in. jako producenta ciekawych interfejsów do Spectrum) jest próbą pójścia dalej ze Spectrum, kontynuowania nurtu zapoczątkowanego dawno temu przez model Spectrum LOKI. U nas „pójście dalej” oznacza 80 KB RAM, stacje FDD3000, przystawkę AY. Co proponują Anglicy?

Tytuł artykułu — „Sam Coupe — przyszłość gier na Spectrum?” mówi już, „co jest grane”. W chwili, gdy wydawało się, że Spectrum +3 będzie ostatnim komputerem sir Sinclaira, MGT przedstawiła jego następcę.

Sam Coupe oprócz swych szerokich możliwości posiada również tryb emulacji „starej” Trumny, więc wszystkie programy (gry!) napisane dotąd dla Spectrum mogą być uruchomione na Samie.

Nowy komputer sprzedawany będzie w dwóch podstawowych konfiguracjach: z magnetofonem i stacją dysków (do dwóch napędów). Sam posiada 61-klawiszową klawiaturę w układzie QWERTY i wydzielony blok 10 klawiszy funkcyjnych.

Na tylnej ścianie zastajemy bogactwo wszelkiego rodzaju gniazd. Znajduje się tam wyjście TV i VIDEO, gniazdo pióra świetlnego, magnetofonu, joysticka, port myszy, I/O interface-u MIDI oraz złącze krzewdziowe.

Wewnątrz — niespodzianka — tylko 8 kości: kontroler obrazu, generator dźwięku, układ specjalizowany SAM, dwa układy po 128 KB RAM (+ miejsce na rozszerzenie do 512 KB). ROM z systemem operacyjnym i interpreterem Basic-a, procesor Z80 i układ obsługi wejścia/wyjścia MIDI. Na osobną uwagę zasługuje układ SAM, zarządzający pamięcią, interface-em MIDI i funkcjami DMA.

Możliwości muzyczne SAM-a dobiją nawet Amigę, a zadowolą najwybredniejszych. Scalony generator dźwięku SAA 1099 firmy Philips pozwala na uzyskanie dźwięku w sześciu kanałach stereo. Możliwości graficzne nie należą do najgorszych. Dostępne są 4 tryby graficzne: 256x192 w 16 kolorach z palety 64, tryb Spectrum używany przy jego emulacji oraz pozostałe dwa, o których brak informacji.

Jednostką centralną jest Z80 taktowany zegarem 6 MHz.

Aby możliwości Sama Coupe mogły być wykorzystane, firma MGT ma zamiar skłonić największe stawy komputerowego światka do tworzenia odpowiedniego oprogramowania. Muzyką ma zająć się nikt inny, jak sam David Whittaker, grafiką Bo Jangebord. Obaj znani są z najlepszych programów na obecne Spectrum. Pozostaje tylko mieć nadzieję, że wielcy świata gier zainteresują się nowym komputerem i nie zaginie on w tłumie innych konstrukcji, jak np. Sinclair QL. Nie oczekujemy jednak sukcesu; nie znajdzie się drugi Microsoft, który upowszechni Sama, jak niegdyś IBM-a.

Na koniec sprawa ceny. Sam Coupe ma być sprzedawany za ok. 150 £. Biorąc pod uwagę jego możliwości, wydaje się on być interesującą i godną przemyślenia ofertą.

Maciej Kasatkin

SAM COUPE, CZYLI KOCHAMY WUJKA CLIVE'A



ABC Data

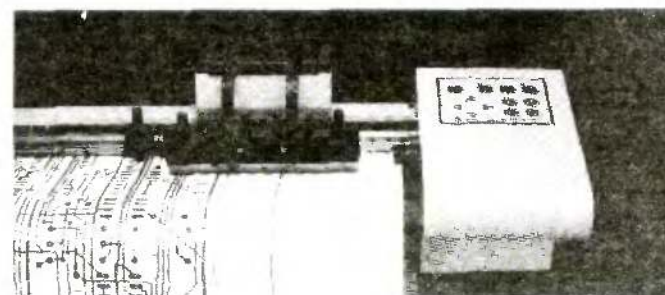
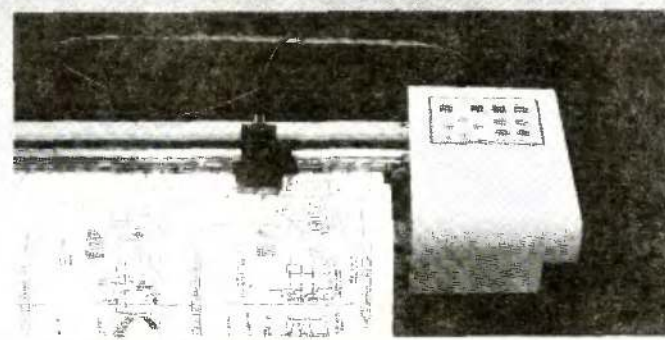
proponuje



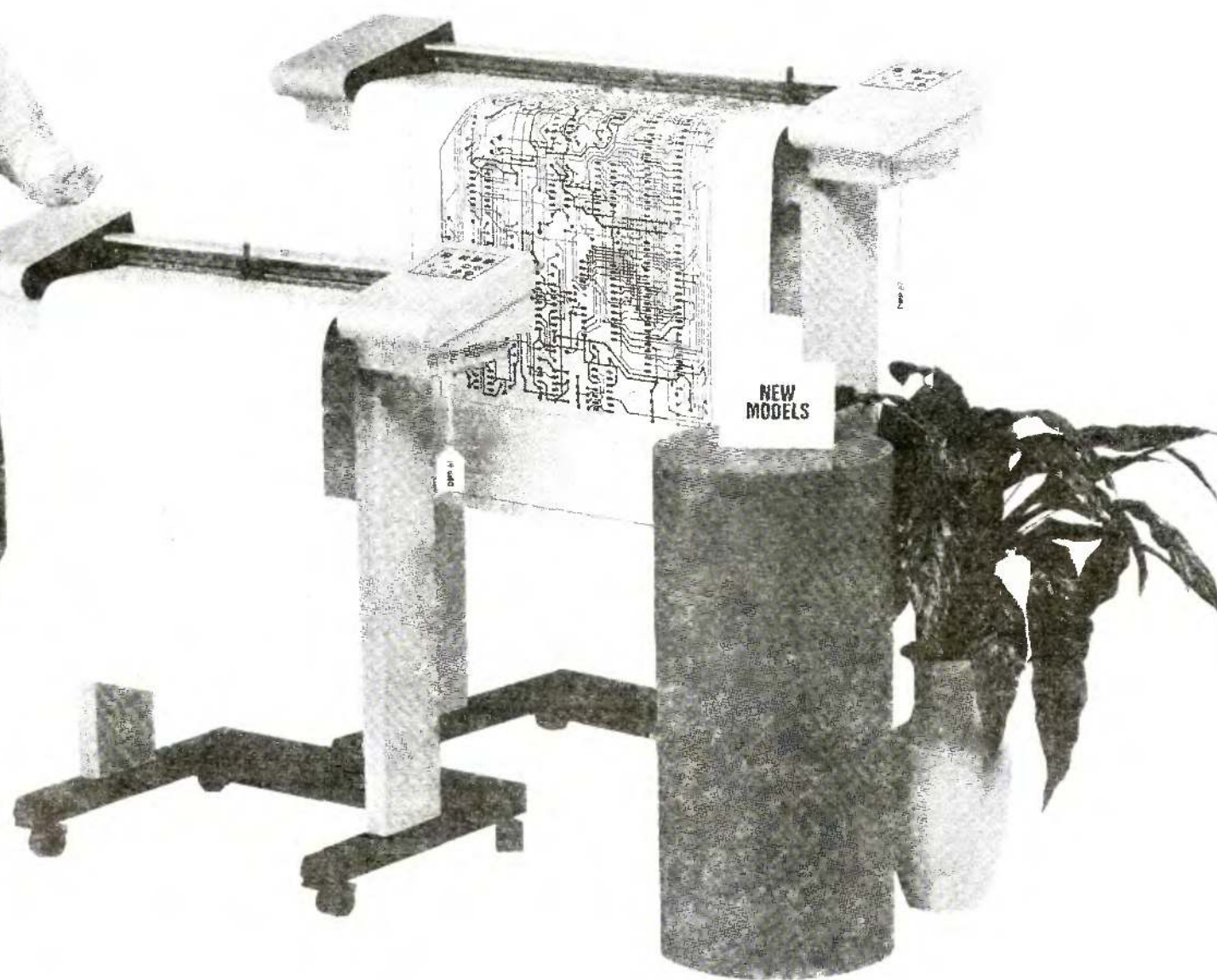
- * plotery tablicowe
- * wielkoformatowe plotery bębnowe
- * digitizery najnowszej generacji
firmy Houston Instrument

Ditmar-Koel-Str. 13
2000 Hamburg 11
tel. (040) 31 40 03
tel. (040) 319 5874
tlx. 21 66 002
fax. (040) 31 91 783

- * atrakcyjne ceny
- * szybkie dostawy
- * gwarancja jakości



HOUSTON
INSTRUMENT
A DIVISION OF AMETEK



PRO-INFO

Adres: Przedsiębiorstwo "PRO-INFO"
 ul. Sikorskiego 12/38
 40-001 KATOWICE
 skrytka pocztowa 1347
 tel. 53-42-88

OFERTA

Oprogramowanie oraz dokumentacja

IBM, Atari 800/65 XE/ST, Amstrad, Commodore, Amiga

IBM, Clipper 86/87 wersja polska, Turbo Pascal v4.0, Turbo C,
 Turbo Basic, dBase III+, Pro-Desin, DOS-3.3, Xenix

oraz organizacja imprez promocyjnych

Masz pomysł - napisz

Masz dokumentację w wersji polskiej - napisz

Masz ciekawy program - napisz

SB-13

SERWIS KOMPUTERÓW

TEST

40-164 Katowice ul. Modrzewiowa 24/33
 poleca naprawy:

- ATARI 600, 800, 65, 130 XL, XE
 - COMMODORE 16, 116, +4, 64, 128
 - DISC DRIVE 1541, 1570, 1571, 1050
 - MAGNETOFONY COMMODORE
 - DRUKARKI
- godz. 9—11, 16—18

SB 30

ZX SPECTRUM, TIMEX, ATARI
 ● programy użytkowe, edukacyjne, gry
 ● programy dla rzemiosła (receptury, kalkulacje, remanenty)
 ● instrukcje do programów
 ● informacje po nadstaniu koperty zwrotnej ze znaczkiem
 ● wysyłka na cały kraj — rachunki „P.K.T.S.” Studio Komputerowe
 00-103 Warszawa
 ul. Królewska 43 m 25 (SB-59)

Sprzedam Commodore 64 + drukarka. Tel. Zakopane 66-364.
 (SB 66)

Własne gry na Atari sprzedam. Informacje Tomasz Bielewicz Reymonta 6, Żywiec.
 (SB 65)

UWAGA użytkownicy CA 80

Programy użytkowe oraz gry
 Koperta zwrotna + znaczek
 Bogdan Chmiela
 32-087 Zielonki 264 (SB 40)

MICROMAN Sp. z o.o.
 prowadzi naprawę klawiatur ZX Spectrum
 Rybnik 44-200 ul. Wiejska 19 tel. 233-56
 Nałęczów 24-140 ul. Kolejowa 13 tel. 114-222 (Lublin) (SB 69)

„Spectrum-interface” umożliwiający grę joystickiem w każdej grze. Nowak. Wielka Skotnica 90c/1, 41-400 Mysłowice.
 (SB 67)

SUPER OFERTA SUPER KATALOG ATARI XL/XE

Największy wybór opisów do gier i programów użytkowych: literatura gry, programy narzędziowe, użytkowe, polskie programy edukacyjne — także w systemie AST
 interfejsy do magnetofonów w systemie standardowym i ATARI SUPER TURBO, przeróbki magnetofonów firmowych na system AST
 TOP DRIVE do samodzielnego montażu w stacjach dysków ATARI 1050
 interfejsy CENTRONICS do drukarek polecamy tanie opisy gier dla odbiorców hurtowych (druk offsetowy)
 SPRZEDAŻ WYSYŁKOWA — GWARANCJA

STUDIO KOMPUTEROWE
MEGABAJT
 03-945 Warszawa 33
 skr. poczt. 28, tel. 17-76-16 D-169

COMPUTER SERVICE

MS elektronik naprawy komputerów:
 Spectrum 48k, +, 128, +2, +3
 Amstrad- Schneider Sharp
 Drukarki, Interfejsy
 Wyjścia monitorowe
 Czynne: od 9.00—16.00
 MSelektronik Legionowa 23, 00-343 Warszawa
 Dojazd: 105, 305, F (Jelonki)
 tel. 37-76-65. (K-118)

MICROMAN

oferuje:
 1. Programy i literaturę dla komputerów Atari XL/XE, Commodore 16/116/+4, Spectrum/Timex na miejscu lub za zaliczeniem pocztowym.
 2. Dla użytkowników ELWRO 800 program kopiujący taśma — dysk.
 3. Przystawki UNIVERSAL TURBO dla magnetofonów Atari XC12, umożliwiające zapis i odczyt programów zarówno w systemie BLIZZARD jak i TURBO 2000.
 UNIVERSAL TURBO to połączenie dwóch najpopularniejszych systemów transmisji w Polsce, to przy tym samym koszcie możliwość wyboru.
 4. Naprawę klawiatur dla komputerów ZX Spectrum.
 5. Dodatkowe akcesoria dla komputerów domowych.
 Informacje na miejscu lub za załączeniem koperty zwrotnej i znaczka.
 Adres: **MICROMAN**
 40-181 Katowice
 ul. Osikowa 66
 tel. 585-106 (SB 68)

STUDIO KOMPUTEROWE

HIT - BIT

programy, literatura, osprzęt komputerowy
ATARI COMMODORE SPECTRUM AMSTRAD
Warszawa: D.H. „Merkury”
 przy Pl. Komuny Paryskiej,
 tel. 39-92-51 w. 69
 Wysyłka: HIT — BIT
 00-967 Warszawa 86
 skr. poczt. 168
 katalog po załączeniu koperty. D-51

REKLAMUJ SIĘ W BAJTKU!

WPHW Dąbrowa Górnicza

ELEKTROIN

SKLEP Nr 163 41-300 DĄBROWA GÓRNICZA, ul. Sobieskiego 17
tel. 622 371 w godz. 10 - 18

OFERUJE:

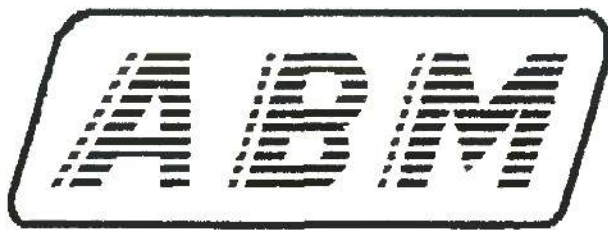
- mikrokomputery 8-bitowe (Amstrad, Commodore, Atari)
- mikrokomputery klasy PC XT/AT
- urządzenia peryferyjne (drukarki, plotery, stacje dysków)
- akcesoria i materiały eksploatacyjne (dyskiety, kable, kasety barwiące, pudełka,...)
- sprzęt video i CTU
- sprzęt elektroakustyczny klasy Hi-Fi
- sprzęt estradowy
- drobny sprzęt elektroniczny i gospodarstwa domowego

SKLEP PROWADZI SPRZEDAŻ POZARYNKOWĄ

PROWADZIMY KOMIS I SKUP W/W ARTYKUŁÓW

W TYM ROKU o 5% TANIEJ!

SERWIS 12-MIESIĘCZNY PROWADZI:



PRZEDSIĘBIORSTWO
POSTĘPU TECHNICZNEGO
SPÓŁKA z o.o.
41-303 DĄBROWA GÓRNICZA
Czerwonych Sztandarów 94
tel. 647 148 tlx. 031 28 99

SB 33

Wojewódzkie Przedsiębiorstwo
Handlu Wewnętrznego
Oddział w Tychach

VIDEOBIT

43-100 Tychy, Al. ZMP 77
tel. 276975

poleca między innymi

- sprzęt komputerowy
Atari ● Commodore ● Amstrad ●
● IBM PC XT/AT/PS 2
- drukarki STAR, EPSON, AMSTRAD
- Sprzęt audiowizualny
- magnetowidy
- OTV PAL/SECAM
- Videoskopy
- kamery
- anteny satelitarne
- aparaturę badawczo-naukową

Udzielamy gwarancji, prowadzimy naprawy pogwarancyjne. Zapewniamy o atrakcyjnych cenach.

SB 18,

COMMODORE 16,116,4/PLUS ATARI XE, XL, ST, SHARP Studio komputerowe „CANON” Proponuje bogatą ofertę oprogramowania Koperta zwrotna Chorzów 41-506 ul. Karłowicza 23/12

D-87

Naprawa komputerów ATARI, COMMODORE, IBM, SPECTRUM oraz urządzeń peryferyjnych. Warszawa tel. 22-07-85

D-84

Programy na C-16, C-116, C plus 4 wysyłam pocztą. Katalog gratis po otrzymaniu koperty zwrotnej. Nagrywanie programów komputerowych. ul. Lenina 104/3 58-304 Wałbrzych

K-261

BIURO HANDLOWO-USŁUGOWE

TECHMAX

KUPNO-SPRZEDAŻ
I POŚREDNICTWO:

- KOMPUTERÓW, DRUKAREK,
DYSKIETEK, PROGRAMÓW,
- SPRZĘTU AUDIO-VIDEO,

OFERUJEMY
OPROGRAMOWANIE
NA KOMPUTER
ATARI XE/XL

81-969 GDYNIA 2
SKR. POCZT. 150

G 156,

STUDIO „TAL-ONERTY”, SKR. POCZT. 51
02-105 - Warszawa 21

POLECANY PROGRAMY NA KOMPUTERY :

- ATARI 600,800,65,130 XE i XL
- ATARI 260,520,1040 ST
- COMMODORE 64,128
- COMMODORE AMIGA 500,1000
- SPECTRUM - TIMEX
- AMSTRAD - SCHNEIDER 464

Polecamy także : opisy i literaturę

Wysokie zniżki dla stałych klientów
sięgające nawet 40% !!!

D 35

NIE CZEKAJ !

ATARI

Posiadasz stację dysków
LWD?

Po zainstalowaniu
TOMS TURBO DRIVE

uzyskasz 3-krotnie szybszy zapis i odczyt, przenoszenia plików na IBM, kopiowanie zabezpieczonych dysków!
Warszawa tel. 641-54-29,
635-41-49, 46-02-01

D-166

SPECTRUM! TIMEX!

SUPERNOWOŚĆ!

Komplet mikroprogramów układających krzyżówki, zadania szaradziarskie.

koperta zwrotna.
Bogdan CHMIELA
32-087 Zielonki 264.

(SB 64)

„BETA B”

AGENCJA INFORMATYCZNA

41-200 Sosnowiec, skrytka 254

Telef. 632-935 690-385

oferuje również wysyłkowo:
Programy, Instrukcje, Literaturę dla komputerów
ACORN AMSTRAD ATARI
COMMODORE SHARP IBM

K-19

ZX SPECTRUM

Polskie programy do zabawy, eksperymentów oraz ciekawych zastosowań:

- zestaw TOTO (DL, SL, Ex, ZS)
 - LIGA POLSKA
 - LITERKI — dla przedszkolaków, rodziców i dziadków!
 - oraz inne atrakcyjne programy
- Informacje kopertą zwrotną

MASTER BIT

61-660 Poznań 31 skr. p. 56 D-23

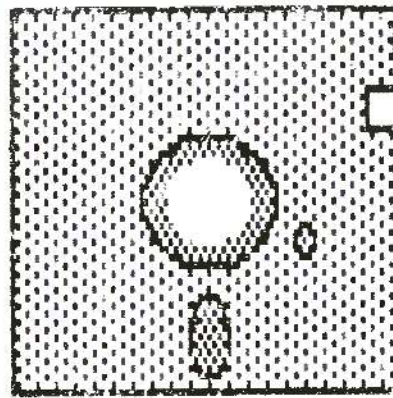
MIKROSERVICE COMMODORE — 64/128, AMIGA, ATARI, SPECTRUM, IBM — PC/XT/AT. INTERFEJSY — CENTRONICS, RS 232, DIGITIZER, FINAL, SPEEDDOS, PROGRAMATORY. EUROKARTY — Z80, 6502, 68000. A/C, CP/M. 01-911 WARSZAWA, ANDERSENA 3/103. D-52

PRODAM KOMPUTER PC DIGITAL RAINBOW 100, RAM 256Kb, MS-DOS/CP-M, 2 DISK. PO 400 KB + MONITOR, TISKARNA LA-50 MNOHO PROGRAMU — WORDSTAR, SCAL, M BASIC, HRY + DOKUMENTACE, TURBO PASCAL — DVORAKOVA MARIE, PRAHA 4 — CHURNAJEVOVO 16 TEL. 4014747 — CSSR.

(SB 71)

Ceny mogą być wyższe ze względu na inflację, a poza tym obowiązują tzw. „przelicznik dolarowy”

	Giełda Bajtka	Sklep Bajtka	Komis	Pewex	RFN	Baltona	CSH i inne
	tys. zł	tys. zł	tys. zł	\$	DM	\$	tys. zł
SINCLAIR							
ZX 81	120	120	—	—	—	—	—
ZX Spectrum 48	400	350	550	—	80	—	—
ZX Spectrum +	450	550	700	—	90	—	—
Timex 2048	400	400	400	—	—	—	240
ZX Spectrum 128+	550	—	—	—	—	—	—
ZX Spectrum 128+2	—	—	—	—	140	—	—
ZX Spectrum 128+3	—	—	—	—	280	—	—
drukarka Seikosha GP 50s	—	280	—	—	—	—	—
Interface Kempston	25	20	—	—	10	—	9.3
COMMODORE							
Commodore 64	900	650	710	199	290	155	950
VC 20	200	180	—	—	—	—	—
C 16	300	250	—	—	80	—	—
C 116	350	240	—	—	70	—	—
C Plus 4	450	450	380	—	150	—	—
C 128	1100	1100	—	—	399	—	—
C 128 D	1900	—	—	450	820	—	—
Amiga 500	2300	2900	—	—	899	—	—
Magnetofon 1531	150	170	170	48	30	225	—
Stacja dysków Oceanic	700	700	—	—	320	170	—
Stacja dysków 1571	1000	1000	—	199	460	—	—
Drukarka LCIOC	1300	1300	—	—	260	230	—
ATARI							
Atari 800 XL	600	700	470	—	160	—	—
Atari 65 XE	700	650	520	127	—	—	—
Atari 130 XE	800	850	—	199	220	—	—
Atari 520 ST	2000	2000	—	—	—	—	—
Atari 1040 ST	—	—	—	—	1140	—	—
Magnetofon XC 12	140	140	—	36	40	—	—
Stacja dysków 1050	900	900	750	185	300	—	—
Stacja dysków 520 STM	—	700	—	—	—	—	—
Drukarka 1029	400	650	—	—	—	—	—
AMSTRAD							
Amstrad 464 mono.	900	900	—	—	350	—	—
Amstrad 664 mono.	—	1200	—	—	—	—	—
Amstrad 6128 mono.	—	—	—	—	670	—	—
Amstrad PCW 8256	—	—	—	—	—	—	—
Amstrad PCW 8512	—	—	—	—	—	—	—
Amstrad PCW 9512	—	—	—	—	—	—	—
Stacja dysków do 464	—	—	—	—	380	—	—
SHARP							
Sharp MZ 700	—	360	—	—	—	—	—
Sharp MZ 800	—	420	—	—	—	—	—
Dyskietki 5.25 cala	2	2-2.5	3	1	0.7	11	4-9
Dyskietki 3.5 cala	7.5	7.5-9	7-9	—	5	2.5	9-10
Dyskietki 3 cale	11	—	—	—	6	3	10
Joystick	21	16-26	20-30	5	10	10	14
Monitor Neptun	60	80	—	—	—	—	68



Piotr Butowski, posiada Amstrada PCW 8256, zainteresowany jest maksymalnym wykorzystaniem jego możliwości. Szuka kontaktu z innymi użytkownikami tego komputera. Adres: 80-471 Gdańsk 45, skr. poczt. 79.

Jarosław Pawłowski, 17 lat. Posiada Atari 1040 STF, monitor kolorowy Atari SC 1224 RGB, drukarkę STAR LC-10. Proponuje wymianę oprogramowania, literatury i doświadczeń. Adres: 30-686 Kraków, ul. Podedworze 4/61.

Władimir Janowicz, lat 19. Posiada mikrokomputer Sharp MZ 821. Zainteresowania: informatyka, muzyka oraz fantastyka. Chciałby nawiązać kontakt w sprawie wymiany informacji o programach. Korespondencja w języku polskim. Adres: CSSR, Dolny Kubin 02601, ul. Febi vii 1170/14.

Bartosz Muszyński, lat 17. Posiada mikrokomputer Commodore C-64, magnetofon Datasette Unit, monitor Neptun 471 oraz joysticki. Nawiąże kontakt w celu wymiany programów oraz literatury. Adres: Na Slovance 20, Praha 8, Libeň 182-00.

Piotr Walaszek, lat 15. Posiada Atari 65 XE, stację dysków LDW 2000 wraz z programem TURBO. Pragnie wymienić doświadczenia i oprogramowanie z innymi posiadaczami Atari ze stacją dysków. 57-100 Strzelin, ul. Kościelna 3/12.

Paweł Kwiecień, lat 12. Posiada Commodore C/16, interesuje się informatyką oraz programowaniem w kodzie maszynowym. Nawiąże kontakt z rówieśnikami w celu wymiany doświadczeń i oprogramowania. Adres: 85-711 Bydgoszcz, ul. Polczyńska 2/198.

Piotr Jodel, uczeń lat 16. Posiada Atari 65 XE, magnetofon XC 12 pracujący w systemie AS TURBO. Pragnie nawiązać kontakt z posiadaczami tego mikrokomputera, proponuje wymianę oprogramowania w systemie TURBO. Adres: 16-100 Sokółka, Os. Centrum 16/20.

Janusz Bójnicki, uczeń 15 lat. Posiada Commodore C+4 z magnetofonem 1531. Oprogramowanie: gry, programy muzyczne oraz graficzne. Zainteresowania: szachy, biologia, komputery. Chętnie nawiąże kontakt z posiadaczami C 16, C 116, C+4 w celu wymiany oprogramowania i doświadczeń. Adres: 34-100 Wadowice, ul. Poprzeczna 1/6.

Paweł Nowodziński lat 15. Posiada Spectrum + magnetofon, joystick oraz około 400 gier i programów użytkowych. Proponuje wymianę oprogramowania oraz doświadczeń. Interesuje się informatyką, sportem, fantastyką. Adres: 42-200 Częstochowa, ul. Siemradzka 9/G.

Krzysztof Wojas, lat 15. Posiada komputer Master Compact z kolorowym monitorem oraz joystick Quick Shot II. Prosi o kontakt posiadaczy tego komputera. Adres: 41-813 Zabrze, ul. Thaimanna 27.

Janusz Gacek, lat 16. Posiada ZX Spectrum + i około 150 programów. Chętnie wymieni oprogramowanie. Adres: 32-400 Myślenice, ul. Słowackiego 23/27.

Jarosław Mikula, lat 15. Posiada komputer AMSTRAD CPC 464, monitor CTM 640, stację dysków DDI-1 (3"). Pragnie nawiązać kontakt z posiadaczami tego komputera. Adres: 44-100 Gliwice, ul. Słowackiego 30/1.

Zapamiętaj adresy nowych punktów przyjęć ogłoszeń

MŁODIEŻOWEJ AGENCJI WYDAWNICZEJ W WARSZAWIE

SALON WYDAWNICZY MAW, ul. Wilcza 71, tel. 28-07-82.
Tu można zamówić w godz. 11.00—18.00 ogłoszenia do: „SZTANDARU MŁODYCH”, „MOTORU”, „BAJTKA”, „ŚWIATA MŁODYCH”, „RAZEM”, „NOWEJ WSI”, „NA PRZEŁAJ”, „ITD”, „WALKI MŁODYCH”, „GAZETY MŁODYCH”, „ŻAGLI”, „FILATELISTY”.

REDAKCJA „MOTORU”, ul. Pankiewicza 3, tel. 21-32-37,
— przyjmuje ogłoszenia bezpośrednio do tygodnika „MOTOR” w godz. 9.00—15.00.

oraz przypominamy
adres stałego punktu przyjmowania ogłoszeń do wszystkich tytułów MŁODIEŻOWEJ AGENCJI WYDAWNICZEJ w godz. 9.00—16.00. 04-026 Warszawa, ul. Aleja Stanów Zjednoczonych 53, tel. 10-56-82.

Drogi Bajtku!

Na listy Czytelników odpowiada Dominik Falkowski

Dość dużym zaskoczeniem była dla mnie duża grupa listów, jaką przyniosła ostatnio redakcyjna poczta. Listy te bez wyjątku dotyczyły komputerów Commodore VIC-20 czyli praszczura C-64.

„Chcę kupić Commodore VIC-20, nie wiem jednak jakie oprogramowanie mam do niego stosować [...]”

Daniel S.

Obawiam się, że ze względu na wiek komputera będziesz musiał korzystać przede wszystkim z programów napisanych przez siebie, gdyż firmowych znajdziesz już niewiele. Główną przeszkodą jest tu po pierwsze — niewielka pamięć RAM (ok. 3.5 KB), a po drugie — niezgodność programowa z następnymi modelami. Niewielką pociechą może być fakt, że niektóre programy napisane w BASIC-u dla C-64 będą działały na VIC-20 pod warunkiem, że nie zawierają one instrukcji POKE, SYS, WAIT, PEEK, nie obejmują grafiki i dźwięku.

„Czy Redakcja poleciłaby VIC-20 jako pierwszy komputer? Czy VIC-20 jest zgodny programowo z innymi modelami Commodore? Jak wygląda sytuacja z literaturą? [...]”

Jacek R., Łódź

VIC-20 NIE JESET ZGODNY programowo z żadnym modelem Commodore — ani wcześniejszymi, ani późniejszymi. Jedyne BASIC wszystkich modeli jest w pewnym stopniu zbliżony do siebie, co umożliwia przenoszenie krótkich procedur z dużymi ograniczeniami (patrz list powyżej). Literatura do VIC-20 praktycznie w Polsce nie istnieje — w księgarniach nie ma co szukać, a na giełdzie spotyka się ją okazjonalnie. Mój stosunek do kupowania **PIERWSZE-GO** komputera (takiego na początek) jest negatywny. Czy przed zakupem motocykla

warto kupować hulajnogę? Wydaje mi się, że nie...

„Zacząłem wymieniać programy ze znajomym, który ma VIC-20 z modułem 3KB, ja natomiast posiadam VIC-20 z RAM rozszerzoną do 16 KB. Niestety, czasami niektóre programy nie chcą się wczytywać, pojawiają się komunikaty błędów, litery zmieniają się w znaki graficzne [...]. Co może być przyczyną tych zaburzeń?”

Tomasz H., Świdnica

Powodów może być kilka. Po pierwsze — brudna głowica magnetofonu lub zniszczona kasetka. Po drugie — (bardziej prawdopodobne) — programy dłuższe przeznaczone dla VIC-20 z odpowiednim rozszerzeniem nie będą NA PEWNO pracowały na wersji podstawowej tego komputera (z 3,5 KB RAM). Opisane obawy przypominają mi najbardziej to właśnie zjawisko. Jedyne ratunkiem będzie więc wyposażenie komputera znajomego kolegi w odpowiednie rozszerzenie pamięci RAM.

„Posiadam komputer Commodore CBM-610 i stację dysków SFD-1001. Pracuje on w języku BASIC V4.0 i zgłasza się jako COMMODORE 128 BASIC V4.0. Czy istnieje możliwość przejścia na język BASIC V2.0 i czy mój komputer może współpracować (instrukcja mówi, że tak) z C-64 [...]”

Dorota Ł., Rumia

Commodore CBM-610 to starsza generacja komputerów tej firmy mieszcząca się w klasyfikacji pomiędzy komputerami PET i VIC-20. Komputery te były tak zwaną serią biurową Commodore i ich znaczna ilość trafiła przed laty do Europy. W chwili obecnej są one wyprzedawane.

BASIC V4.0 zawiera w sobie w zasadzie wszystkie instrukcje BASIC V2.0 wywodzącego się zresztą z tego dialektu. Jednakże istnieje kilka instrukcji (SYS, WAIT, POKE, PEEK), które odnoszą się do specyficznych komórek pamięci i powodują, że programy w BASIC-u nie będą działały.

Współpraca obu komputerów obejmuje właściwie jedynie BASIC. Żadne wymiany układów scalonych nie zamienią CBM-610 w C-64. Co do wczytywania programów z dyskietek to SFD-1001 ma niewątpliwą przewagę nad 1541 — oferuje 4133 bloki wolne (ok. 1.06 MB pojemności) i jest w stanie odczytywać dyskietki z innych stacji Commodore.

„Jak można wpisać program do pamięci C-16 w taki sposób, aby nie zdążył się on uruchomić?”

Marek N., Wałbrzych

Rozumiem, że chodzi tu o zapobieżenie samouruchomianiu się programu wczytanemu do komputera z dyskietki lub kasety firmowej. Zadanie to nie jest wcale takie proste i bez dobrej znajomości programowania w języku wewnętrznym lub odpowiedniego urządzenia bardzo trudne do zrealizowania. Problem polega na fakcie, że taki program jest wyposażony w procedurę kontrolującą od początku do końca cały proces wczytywania. Gdy do pamięci zostanie wczytany ostatni bajt, procedura ta automatycznie uruchamia program, sama się przy tym deaktywując. Rozwiązanie polega na zmianie fragmentu kodu tej procedury lub jej całkowitemu wyeliminowaniu.

„Mam problem, niedawno kupiłem C-64 i myszkę. Jednakże myszka działa tylko z programem firmowym, a nie chce dzia-

łać z programem GEOS [...]”

Marcin S., Katowice

Z opisu wynika, że Twoja wersja GEOS nie jest wyposażona w program sterujący do myszki. Jeżeli jednak znajdziesz na dyskietce systemowej program o nazwie COMM MOUSE to oznacza to, że nie jest on pierwszym programem sterującym zapisanym na dyskietce (aby myszka była aktywowana automatycznie, program ten powinien być umieszczony jako pierwszy program obsługujący sterownik). Możesz również wcisnąć klawisz Commodore Logo (C=) i I; na ekranie ukaże się okienko zawierające wszystkie programy obsługujące sterowniki. Przesuń klawiszami kursora wskaźnik tak, aby podświetlony został tytuł COMM MOUSE, następnie wcisnij RETURN.

„Moim problemem jest to, że w wielu grach nie mogę znaleźć nieśmiertelności [...]”

Urban W., Włocławek

Uzyskiwanie „nieśmiertelności” wcale nie jest takie proste, jakby się to mogło wydawać. Najczęściej wyszukuje się w tym celu (np. za pomocą monitora języka maszynowego) wszystkie instrukcje odnoszące się do rejestrów kontrolera graficznego VIC (przykład dla C-64), rozpoczynających wystąpienie kolizji typu sprite-sprite lub sprite-podkład. Po odpowiednich przeróbkach (wystarczy w tym celu zmienić kilka bajtów) albo osiągasz „pełną nieśmiertelność”, albo też masz do dyspozycji coraz więcej ludzików czy dajmy na to rakiet. Tym niemniej dla osób nie znających dobrze języka maszynowego jest to dość trudne. Dobre wyniki może dać również cykliczne sprawdzanie zawartości „podejrzanych” komórek pamięci.



JOYSTICKI do Atari Commodore, Spectrum, Amstrad precyzyjny mechanizm specjalne styki 9800,— zł, 6 m-cy gwarancja interface do Spectrum

Wysyłka natychmiastowa za zaliczeniem pocztowym

Dla instytucji rachunki płatne przelewem

ELEKTROMECHANIKA

ul. Cegielniana 17
32-410 DOBCZYCE

G-129

TOMBAT
XL ATARI XE

- gry i programy użytkowe
- co piąty program gratis
- inne bonifikaty
- opisy gier i instrukcje
- pomoc dla początkujących
- wysyłka na cały kraj
- katalog gratis

WYPOŻYCZALNIA
TOMBAT
XL ATARI XE

Nasz adres:

ul. Magistracka 27 m 26
01-413 Warszawa
Tel. 363-078 godz. 12-20

Zapraszamy!

(SB 12)

ATASERW

43-100 TYCHY ul. Lencwicza
46/3
tel. 27 69 66

oferuje świetne rozwiązania sprzętowe

do ATARI XL/XE:

1. TURBO DOS—wspaniały DOS na kartridżu
 2. TOP DRIVE—do stacji 1050, LDW 2000, CALIFORNIA samodzielny montaż—wysyłkowo (rec. INFORMIK III/88)
 3. INTERFEJS CENTRONIKS
 4. ROZSZERZENIA PAMIĘCI
 5. BASIC XE—kartridż
 6. TURBO DOS + BUG65—kartridż
- 12 miesięcy gwarancji. Informacje i zamówienia telefonicznie (wtorek 8—12, środa, czwartek 16—18) i listownie po otrzymaniu koperty zwrotnej

K—236

ATARI
TURBO
2000 F

Nowy system transmisji danych z magnetofonem przyspieszony do 6700 bodów

- Komplet:
- cartridge
 - kasetka z 5 programami kopiującymi
 - przeróbka magnetofonu
 - 12 m-cy gwarancji za 29 000 zł oraz interfejs do standardowego magnetofonu za 22 000 zł
- oferuje firma MUEL ul. Cząstkowska 30, 01-678 Warszawa tel. 33-40-91

D-121



JOYSTICK
SERVICE

GUN SHOT
QUICK SHOT (I-V)
VG-125 , CX 40
200 X i inne

- * wymiana standardowych styków na mikrolaczniki
- ** naprawa

Zgłoszenia: Studio komputerowe SEZAM
D.H. "SEZAM" lip. — czwartki 16" — 19"
Rachunki, Zniżki cen dla szkół i klubów.
Prowadzimy ekspedycję pocztową. Szczegółowe informacje po nadesłaniu koperty zwrotnej.

Korespondencja : JOYSTICK SERVICE
02-770 Warszawa 130 skr. poczt. 102

D-175



Ten program Przemka Siwińskiego (lat 14) — ucznia 8 klasy Szkoły Podstawowej nr 1 imienia Gustawa Morcin-ka — musiał czekać na swoją kolej prawie rok. Po prostu nie zdążył „załapać się” w zeszyt rocznym numerze gwiazdkowym. Przemek współpracuje z naszą rubryką od dawna, przysyłając nam coraz to nowe i coraz to doskonalsze programy. Ten, który prezentujemy poniżej jest już zupełnie „dorosły” i mógłby się znaleźć w Klanie Amstrada, bo na ten właśnie komputer został napisany. Gratuluję Ci Przemku!

Romek

Jesteś Świętym Mikołajem i Twój Zwierzchnik (ciekawe kto taki może być zwierzchnikiem Świętego Mikołaja — przyp. redakcji) kazał Ci odwiedzić dzieci mieszkające przy pewnej, bardzo długiej ulicy. Mieszkają tam różne dzieci: grzeczne i nieposłuszne. Grzecznym trzeba wręczyć ładne prezenty, a nieposłusznym — różgi. Musisz się bardzo spieszyć, a w pośpiechu łatwo się pomylić. Im lepiej wyko-

nasz swoje zadanie, tym więcej otrzymasz punktów: 50 i więcej, to już bardzo dobry wynik.

Powodzenia!
Kierowanie odbywa się za pomocą klawiszy kursora.

→ rzut prezentem w prawo
← rzut prezentem w lewo
[COPY] + → rzut różgą w prawo
[COPY] + ← rzut różgą w lewo

Przemek Siwiński

```

10 REM *****
20 REM *
30 REM * PREZENTY I RÓZGI *
40 REM *
50 REM * P. Siwinski'1988 *
60 REM *
70 REM *****
80 DIM d1(300),d2(300),c$(9),a1$(308),a2$(308)
90 GOSUB 10000:GOSUB 10500:GOSUB 17500
100 GOSUB 11000:GOSUB 19000:GOSUB 11500
110 MODE 0:PAPER 5:CLS:PEN 3:LOCATE 2,23
:PRINT"PLUNKTY:"
120 h$=STRING$(20,228):LOCATE 1,1:PRINT h$;h$;
130 LOCATE 1,17:PRINT h$;h$;
140 h$=STRING$(2,228):FOR k=3 TO 16:LOCA TE 1,k:PRINT h$
150 LOCATE 19,k:PRINT h$;:NEXT
160 PEN 1:LOCATE 10,3:PRINT CHR$(238);:P EN 11:PRINT CHR$(239)
170 LOCATE 10,4:PEN 1:PRINT CHR$(240);CH R$(241);CHR$(242)
180 LOCATE 10,5:PRINT CHR$(243);CHR$(244 );CHR$(245)
190 LOCATE 10,6:PRINT CHR$(246);CHR$(247 );CHR$(248)
200 LOCATE 10,7:PRINT CHR$(249);CHR$(250 );CHR$(251)
210 WINDOW #1,3,7,3,16:WINDOW #2,14,18,3 ,16:WINDOW #3,10,12,8,16
220 WINDOW #4,11,13,20,24:WINDOW #5,15,1 7,20,24:PEN #1,0:PEN #2,0:PAPER #1,5
230 a=1:pkt=0:LOCATE #3,1,8:GOSUB 12000: PAPER #2,5:PAPER #3,5:BORDER 12
250 REM * GRA *
260 PAPER #4,5:PAPER #5,5
270 WHILE a<308:GOSUB 12500:GOSUB 13000: IF a<12 THEN 320
280 IF INKEY(1)<>-1 AND INKEY(9)=-1 THEN GOSUB 13500:GOTO 320
290 IF INKEY(8)<>-1 AND INKEY(9)=-1 THEN GOSUB 14000:GOTO 320
300 IF INKEY(1)<>-1 AND INKEY(9)<>-1 THE N GOSUB 15000:GOTO 320
310 IF INKEY(8)<>-1 AND INKEY(9)<>-1 THE N GOSUB 15500
320 WEND
1000 WHILE INKEY$<>"":WEND
1010 WHILE INKEY$="":WEND
1020 FOR s=1 TO 5:GOSUB 19500:NEXT:ERASE d1,d2:DIM d1(300),d2(300):GOSUB 11000:G OSUB 11500:CLS #1:CLS #2:GOTO 230
9999 REM Znak "■" to CHR$(143)
10000 DATA "■■■■ ■■ ■■ ■■■■"
10010 DATA " ■ ■■■ ■ ■■"
10020 DATA "■■■ ■■■■ ■■■■"
10030 DATA "■■■ ■■■ ■■■■"
10040 DATA "■ ■■ ■■■■ ■ ■■"
10050 DATA "■■■■ ■■■ ■■■■"
10060 DATA "■■■■ ■■■■ ■■■■"
10070 DATA "■■■ ■ ■■ ■■"
10080 DATA "■■■■ ■■■■ ■■■■"
10090 DATA "■■■■ ■■■■ ■■■■"
10100 RESTORE 10000
10110 FOR k=0 TO 9:READ c$(k):NEXT:RETUR N
10500 SYMBOL AFTER 231
10505 SYMBOL 231,170,170,255,255,255,255 ,170,170
10510 SYMBOL 232,85,85,255,255,255,255,8 5,85
10520 SYMBOL 233,126,24,252,26,25,36,36, 195
10530 SYMBOL 234,0,126,129,165,129,165,1 53,129
10540 SYMBOL 235,36,126,129,165,129,153, 165,129
10550 SYMBOL 236,0,127,127,127,127,127,1
27,0
10560 SYMBOL 237,0,254,254,254,254,254,2 54,0
10570 SYMBOL 238,0,0,0,28,34,34,76,64
10580 SYMBOL 239,24,60,60,24,24,60,126,2 55
10590 SYMBOL 240,65,66,135,136,137,147,1 45,144
10600 SYMBOL 241,255,0,255,0,0,129,24,60
10610 SYMBOL 242,128,64,224,16,144,200,1 36,8
10620 SYMBOL 243,160,160,160,160,224,229 ,164,166
10630 SYMBOL 244,126,126,126,126,126,60, 129,126
10640 SYMBOL 245,4,4,4,4,7,167,36,100
10650 SYMBOL 246,163,163,163,163,163,161 ,160,32
10660 SYMBOL 247,0,255,255,255,255,1 26,24
10670 SYMBOL 248,196,196,196,196,196,132 ,4,4
10680 SYMBOL 249,64,81,213,255,3,9,17,15
10690 SYMBOL 250,0,17,85,255,195,129,129 ,129
10700 SYMBOL 251,2,18,85,255,192,144,136 ,240
10710 SYMBOL 252,96,144,136,136,132,66,6 3,1
10720 SYMBOL 253,14,17,33,33,65,130,252, 128
10730 SYMBOL 254,136,80,82,37,36,36,24,2 4
10740 SYMBOL 255,255,129,129,129,129,129 ,129,255
10750 RETURN
11000 FOR k=1 TO 303 STEP 2
11010 a1$(k)=CHR$(236)+CHR$(237)+CHR$(23 6)+CHR$(237)+CHR$(236)
11020 a2$(k)=CHR$(237)+CHR$(236)+CHR$(23 7)+CHR$(236)+CHR$(237)
11030 a2$(k+1)=a1$(k):a1$(k+1)=a2$(k):NE XT k
11040 FOR k=304 TO 308 STEP 2:a1$(k-1)=" META ":a2$(k-1)=" META"
11045 a1$(k)=" ":a2$(k)=a1$(k):NEXT: h$=" "
11050 FOR k=20 TO 288 STEP 14:n=INT(RND* 10)-2:o=INT(RND*10)+2
11060 p=k+n:r=k+o
11070 MID$(a1$(p),2,4)=h$:MID$(a1$(p+1), 2,4)=h$:MID$(a1$(p-1),2,4)=h$
11080 d1(p)=INT(RND*2)+1:d1(p+1)=d1(p)+2
11090 MID$(a2$(r),1,4)=h$:MID$(a2$(r+1), 1,4)=h$:MID$(a2$(r-1),1,4)=h$
11100 d2(r)=INT(RND*2)+1:d2(r+1)=d2(r)+2
11110 NEXT:RETURN
11500 FOR k=18 TO 300
11510 IF d1(k)=1 THEN MID$(a1$(k),2,1)=C HR$(234)
11520 IF d1(k)=2 THEN MID$(a1$(k),2,1)=C HR$(235)
11530 IF d1(k)=3 OR d1(k)=4 THEN MID$(a1 $(k),2,1)=CHR$(233)
11540 IF d2(k)=1 THEN MID$(a2$(k),4,1)=C HR$(234)
11550 IF d2(k)=2 THEN MID$(a2$(k),4,1)=C HR$(235)
11560 IF d2(k)=3 OR d2(k)=4 THEN MID$(a2 $(k),4,1)=CHR$(233)
11570 NEXT:RETURN
12000 IF pkt<0 THEN pkt=0
12010 IF pkt<10 THEN n=0 ELSE n=INT(pkt/ 10)
12020 PRINT #4,c$(n);:PRINT #5,c$(pkt-n* 10);:RETURN
12500 a=a+1:PRINT #1,a1$(a);:PRINT #2,a2 $(a);:RETURN
13000 FOR k=1 TO 3:PLOT INT(RND*48)+308,
154,3:NEXT
13010 PRINT #3:RETURN
13500 IF d2(a-10)=0 OR d2(a-10)=3 OR d2( a-10)=4 THEN RETURN
13510 x=16:y=6:IF d2(a-10)=2 THEN GOSUB 16700:RETURN
13530 IF d2(a-10)=1 THEN GOSUB 16000:RET URN
14000 IF d1(a-10)=0 OR d1(a-10)=3 OR d1( a-10)=4 THEN RETURN
14010 x=4:y=6:IF d1(a-10)=2 THEN GOSUB 1 6700:RETURN
14030 IF d1(a-10)=1 THEN GOSUB 16000:RET URN
15000 IF d2(a-10)=0 OR d2(a-10)=3 OR d2( a-10)=4 THEN RETURN
15010 x=17:y=6:IF d2(a-10)=1 THEN GOSUB 16500:RETURN
15030 IF d2(a-10)=2 THEN GOSUB 16300:RET URN
15500 IF d1(a-10)=0 OR d1(a-10)=3 OR d1( a-10)=4 THEN RETURN
15510 x=4:y=6:IF d1(a-10)=1 THEN GOSUB 1 6500:RETURN
15530 IF d1(a-10)=2 THEN GOSUB 16300:RET URN
16000 PEN 10:LOCATE x,y:PRINT CHR$(252); CHR$(253)
16010 LOCATE x,y+1:PRINT CHR$(255);CHR$( 255):pkt=pkt+2:GOSUB 12000
16020 GOSUB 19500:RETURN
16300 PEN 7:LOCATE x,y:PRINT CHR$(254):L OCATE x,y+1:PRINT CHR$(124)
16310 pkt=pkt+1:GOSUB 12000:GOSUB 19500: RETURN
16500 PEN 9:LOCATE x,y:PRINT CHR$(254):L OCATE x,y+1:PRINT CHR$(124)
16510 FOR k=12 TO 1 STEP -1
16520 SOUND 1,k*50,2,15:SOUND 2,k*45,2,1 5:SOUND 4,k*40,2,15:NEXT
16530 pkt=pkt-3:GOSUB 12000:RETURN
16700 PEN 9:LOCATE x,y:PRINT CHR$(252);C HR$(253):LOCATE x,y+1
16710 PRINT CHR$(255);CHR$(255)
16720 FOR k=1 TO 12:SOUND 1,k*50,2,15:SO UND 2,k*45,2,15:SOUND 4,k*40,2,15
16730 NEXT:pkt=pkt-4:GOSUB 12000:RETURN
17500 MODE 1:PEN 3:PAPER 0:CLS
17510 LOCATE 20,2:PRINT"*":LOCATE 20,3:P RINT CHR$(143):FOR k=1 TO 8
17520 LOCATE 20,k*2+3:PRINT CHR$(143)
17530 LOCATE 20-k,k*2+2:PRINT STRING$(k, 232);CHR$(143);STRING$(k,231)
17540 NEXT k
17550 FOR k=5 TO 35 STEP 5:PEN INT(RND*3 )+1:IF INT(RND*2)=1 THEN 17585
17560 LOCATE k+1,21:PRINT CHR$(252);CHR$( 253)
17570 LOCATE k+1,22:PRINT CHR$(255);CHR$( 255)
17580 LOCATE k+1,23:PRINT CHR$(255);CHR$( 255):GOTO 17590
17585 LOCATE k+2,22:PRINT CHR$(254):LOCA TE k+2,23:PRINT CHR$(124)
17590 NEXT
17600 PEN 1:LOCATE 12,25:PRINT"Prezenty i rózgi 0":RETURN
19000 RESTORE 19000:DATA 0,26,24,18,16,1 2,8
19010 FOR k=0 TO 6:READ 1:INK k,1:NEXT
19020 FOR k=7 TO 10:READ 1,m:INK k,1,m:NE XT:DATA 6,20,2,24,0,26,10,20
19030 INK 11,5:INK 12,4:INK 13,20:INK 14 ,22:INK 15,14:RETURN
19500 RESTORE 19500:DATA 119,95,106,119, 127,142
19510 FOR k=1 TO 6:READ i:SOUND 1,i,4,15 :SOUND 2,i/2,4,15:NEXT:RETURN

```


TYLKO DLA PRZEDSZKOLAKÓW

CHOINKA RAZ JESZCZE

Cześć Maluchy!

Stało się już naszą tradycją — tak, tak, jesteśmy już pismem z tradycjami; „Bajtek” dawno już skończył cztery lata — że na każde święto Bożego Narodzenia Przedszkolaki programują choinkę. I tym razem nie może być inaczej.

Tegoroczna choinka jest wyjątkowo prosta do zaprogramowania. Nie znaczy to wcale, że z tego powodu program nie jest interesujący. Może nawet odwrotnie... Program ma narysować na ekranie choinkę ze świeczkami, a następnie spowodować, aby płomienie świec zmieniały się, migocząc jak prawdziwe świeczki.

W moim programie jest pięć różnych kształtów płomienia. Oczywiście, Wy możecie wymyślić ich znacznie więcej. Znaki, które wybrałem, umieszczam w zmiennych tekstowych o nazwach 01\$, 02\$... 05\$ (linie 10—50). Podczas rysowania drzewka (linie 100—290) komputer nad każdą świeczką drukuje wartość jednej z tych zmiennych.

Oczywiście nad konkretną świeczką drukuje zawsze wartość tej samej zmiennej. Aby więc osiągnąć wrażenie migotania, należy zmieniać te wartości. Komputer robi to po zakończeniu rysowania drzewka (linie 500—550). Pojawia się nam tutaj jeszcze jedna zmienna: 00\$. Potrzebna jest ona jedynie do chwilowego przechowania wartości pierwszej zmiennej.

Po zamianie wartości zmiennych komputer przechodzi do ponownego rysowania choinki. Rozkaz w linii 90 powoduje ustawienie kursora w lewym górnym rogu bez kasowania ekranu.

Mam nadzieję, że nie poprzestaniecie na przepisaniu tego programu, ale sami zaprogramujecie o wiele piękniejsze i bogatsze świąteczne drzewka. Może na nich znaleźć się o wiele więcej świeczek, mogą także zawisnąć inne ozdoby — bombki, gwiazdki, łańcuchy. A pod choinką — wiadomo — powinien leżeć cały stos wspianych zabawek, które też mogą się ruszać, błyszczeć.

A gdyby tak jeszcze komputer zagrał piękną kolędę. Bo cóż to za święta bez kolędy. Jeśli sami nie potraficie, może poproście o pomoc starszego brata, kolegę, a może tatę, jeśli znajdzie czas. Myślę, że taka komputerowa choinka będzie mogła stanąć obok prawdziwej w piękny, wigilijny wieczór.

Wesołych Świąt, Wam i Waszym Rodzicom życzy

Romek

```

5 CLS
9 REM ** dane początkowe ***
10 LET 01$=""
20 LET 02$="o"
30 LET 03$="O"
40 LET 04$="@ "
50 LET 05$="&"
90 LOCATE 1,1
99 REM **** rysunek ****
100 PRINT
110 PRINT "          !"
120 PRINT "          /%\"
130 PRINT "          "; 01$; " )*( "
; 02$
140 PRINT "          I # I"
150 PRINT "          +++#+++"
160 PRINT "          "; 03$; " #
"; 04$
170 PRINT "          I # I"
180 PRINT "          +++#+++"
190 PRINT "          "; 05$; " #
"; 01$
200 PRINT "          I # I"
210 PRINT "          +++#+++"
220 PRINT "          "; 02$; " #
"; 03$
230 PRINT "          I # I"
240 PRINT "          +++#+++"
250 PRINT "          "; 04$; " #
"; 05$
260 PRINT "          I # I
"
270 PRINT "          +++#+++++
+"
280 PRINT "          #"
290 PRINT "          #"
499 REM *** zmiana danych ***
500 LET 00$=01$
510 LET 01$=02$
520 LET 02$=03$
530 LET 03$=04$
540 LET 04$=05$
550 LET 05$=00$
600 GOTO 90
    
```

Uzupełnienia

Atari 800XL/65XE

```

1 DIM 00$(1)
2 DIM 01$(1)
3 DIM 02$(1)
4 DIM 03$(1)
5 DIM 04$(1)
6 DIM 05$(1)
7 POKE 82,0
8 PRINT CHR$(125)
90 POSITION 0,0
    
```

Spectrum 48/128

```
90 PRINT AT 0,0
```

Commodore 64/128

```
5 PRINT CHR$(147)
90 PRINT CHR$(19)
```

dokończenie ze str. 32

wania. Programuję samodzielnie w języku GFA Basic. Z biegiem czasu zdobyłem niezbędną wiedzę, ponieważ musiałem samodzielnie opracować program do malowania. Naturalnie nie ma żadnego oprogramowania przeznaczonego specjalnie do rysowania komiksów. Dlatego też napisałem na własne potrzeby graficzny program „Pixel”. Było to bardzo uciążliwe, ponieważ nie mogłem znaleźć nikogo, kto mógłby mi pomóc i udzielić fachowych porad.

Ile czasu zajmuje Panu zaprojektowanie komiksu takiego jak „Imperium robotów”?

M.G.: Z samego tylko „Imperium robotów” nie mógłbym się utrzymać. Rysuję również seriale komiksowe „Fred Feuerstein” lub „Heidi”, naturalnie odręcznie. Popołudnia i czasami noce mam zarezerwowane dla komputerowego projektowania. Jedna część zajmuje mi około jednego roku, przy codziennej pracy cztery do pięciu godzin.

Tak długo mimo zastosowania ST, czy tak krótko dzięki ST?

M.G.: Ręczna praca trwa dokładnie tak samo długo. Więcej czasu zajmują kolorowe komiksy, ponieważ kolory nie są widoczne na monochromatycznym monitorze. Pomagają jedynie próbne wydruki, bardzo wiele próbnych wydruków. W najbliższej przyszłości mam zamiar kupić sobie kolorowy monitor.

Druga część „Imperium robotów” ma ukazać się w kolorze. Czy stosuje Pan do tego kolorową drukarkę?

M.G.: Nie, pozostałem przy FX-85. Zestaw kolorowych taśm zrobiłem sobie sam. Przepitowałem i skróciłem kasetę z czterokolorową taśmą, a potem ponownie ją sklepiłem. Obraz jest najpierw drukowany w jednym kolorze, następnie papier przewijam z powrotem i drukuję następne kolory: żółty, niebieski, czerwony, czarny. Znakomita byłaby kolorowa drukarka laserowa. Niestety jeszcze trochę potrwa, zanim na rynku ukażą się takie urządzenia po odpowiednio korzystnej cenie. Opłacałaby się kolorowa drukarka termiczna, gdyby nie zbyt wysokie koszty materiałów.

Opr. Janusz Jarmoch

Czy komputer może być dla rysownika takim samym narzędziem pracy jak ołówek i farby? To pytanie zadawano sobie już nie raz. Pierwsze próby wyprodukowania komputerowych komiksów podjęto już przed czterema laty. Amerykański komiks „Shatter”, powstały przy pomocy Apple Macintosh, nie zyskał sobie jednak zbyt wielkiego uznania. Zbyt grube ziarno, wynikające z małej rozdzielczości komputerowego obrazu, nie wywołało dobrego wrażenia na Czytelnikach.

Wśród artystów przeważają nadal sceptycy. Niektórzy z nich uważają komputer za narzędzie znacznie mniej uniwersalne niż pędzel, inni obawiają się, że elektroniczne urządzenia pozbawią ich miejsc pracy.

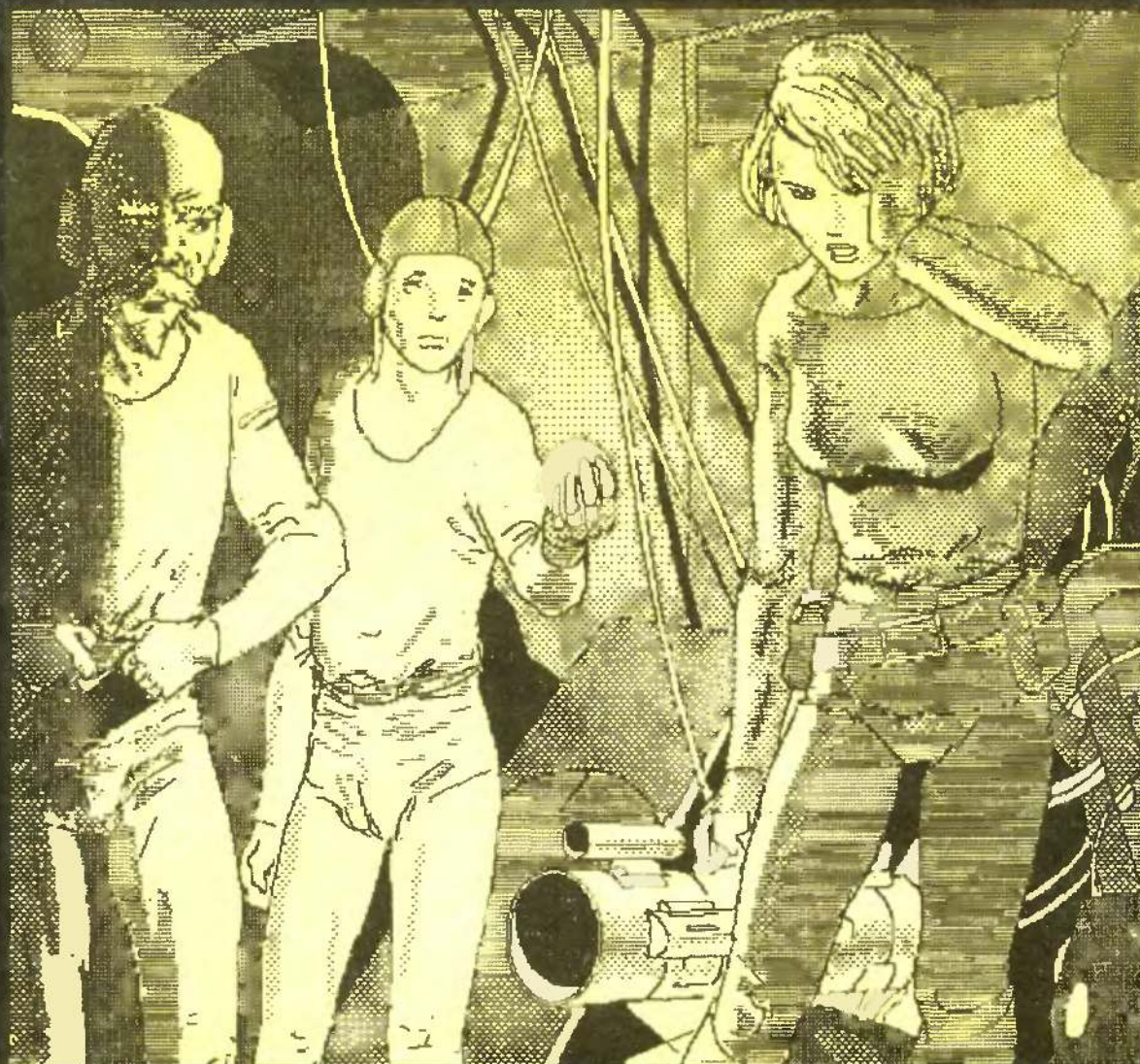
W czerwcowym numerze „ST Magazin” ukazał się wywiad z rysownikiem Michaeliem Götze z RFN, który zamienił pędzel i ołówek na komputer Atari ST i drukarkę. Jest on autorem komiksu „Imperium robotów” (Robot-Imperium), pierwszego dokorowego komiksu stworzonego całkowicie przy pomocy komputera. W wywiadzie tym Michael Götze odsłania kulisy swojego warsztatu pracy.

Rysuje Pan przy pomocy komputera. Czy nie obawia się Pan, że komputer może kiedyś całkowicie zastąpić rysownika?

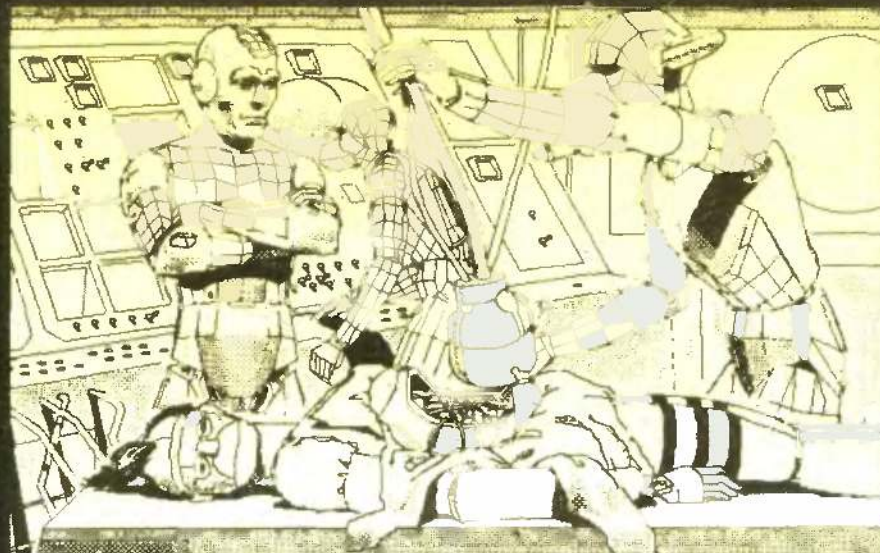
Michael Götze: Obawy te są nieuzasadnione. Komputer nie potrafi sam wymyślić akcji i zilustrować jej rysunkami. To czyni człowiek siedzący przy klawiaturze.

Dlaczego w ogóle rysować przy pomocy komputera? Czy ludzie, którzy w ogóle nie potrafią rysować, nie uznali zbyt wcześnie, że komputer może zastąpić rękę artysty?

M.G.: Na litość boską nie. Kto nie potrafi rysować własnoręcznie, nie stworzy przy po-



Nowe niebezpieczeństwo dla bojowników z podziemia, Higgena Dau-Dau i jego towarzysza Scharlocka.



Panujące na Ziemi roboty odnalazły swoich wrogów...

mocy komputera nic wartościowego, również żadnego dobrego komiksu.

Dlaczego rysuje Pan przy pomocy komputera, skoro można się obejść bez niego?

M.G.: Zdecydowała o tym audycja telewizyjna, w której



...ostatnich ludzi i planują ich zniszczenie.

pokazano, jak wspaniałe efekty uzyskuje się w studiu zajmującym się komputerową animacją. To mnie zafascynowało. Siedziałem jak oniemiały przed telewizorem i powiedziałem sobie, że to byłoby coś nowego w branży komiksowej. Muszę tego spróbować.

Czy jest to naprawdę takie łatwe?

M.G.: No tak. Było trochę problemów zanim wszystko ułożyło się tak, jak to sobie wyobrażałem. Przy pomocy Atari ST i drukarki Epson FX-85 pracuje się całkiem nieźle. Uważam komputerowe komiksy za środek wyrazu z przyszłością. Opowiadam się za nimi całym sercem, zwłaszcza że stawiamy dopiero pierwsze kroki. W przyszłości najtańsze komputery będą miały o wiele większą rozdzielczość niż komputery obecnie dostępne na rynku. Kiedyś wreszcie komputerowe rysunki osiągną taką doskonałość, że nikt nie odróżni ich od narysowanych odręcznie.

Czy związał się Pan z Atari ST już na stałe, czy też może Pan sobie wyobrazić rysowanie komiksów przy pomocy jakiegoś innego systemu?

M.G.: Nie jestem przywiązany ani do ST, ani do żadnego innego komputera. Kupiłem ST, ponieważ przy jego pomocy można uzyskać zadowalające efekty. Przedtem miałem komputer Schneider CPC, który miał zbyt małą rozdzielczość, pamięć i niewystarczającą szybkość działania.

Nie jest więc Pan komputerowym fanem przywiązanym wyłącznie do Atari?

M.G.: Nie jestem użytkownikiem komputera, który zaczął rysować komiksy, lecz twórcą komiksów, który odkrył komputer jako znakomite narzędzie pracy. To oczywiście nie oznacza, że używam komputera wyłącznie jako narzędzia do rysowania.

dokonczenie na str. 31

4. Michael Götze, autor pierwszego w świecie komiksu, stworzonego przy pomocy komputera

