

Z MIKROKOMPUTEREM NA TY

NR INDEKSU 353965  
PL ISSN 0860-1674

# Bajtek

MIESIĘCZNY DODATEK DO SZTANDARU MŁODYCH

NR 4(40)

KWIECIEŃ 1989

CENA 220 ZŁ

KONKURS  
TYLKO DLA GRACZY

ATARI  
SUPER  
TURBO

KTO  
ZBUDOWAŁ  
PIERWSZY  
KOMPUTER?

BLIŻEJ  
ŚWIATA?

BUG/65

PUSZKA PANDORY





# COŚ NA ZIMNO!

A jednak nie ma nic wspanialszego niż niespodzianki, jakich może dostarczyć nauka!

Raptem dwa lata temu otworzył się zaczarowany kocioł z nadprzewodnictwem, a oto znowu z naukowych laboratoriów nadeszły informacje o odkryciu, które z całą pewnością wpisane zostanie do „złotej dziesiątki” sensacji naukowych XX wieku. Mam oczywiście na myśli „zimną” syntezę jądrową.

Sensacja wybuchła 23 marca br., gdy „Financial Times” jako pierwszy poinformował, że dwaj naukowcy, profesorowie Martin Fleischmann z Uniwersytetu Southampton w Anglii i Stanley Pons z Uniwersytetu Utah w USA, przeprowadzili w „śmiesznie prosty” sposób reakcję kontrolowanej syntezy jądrowej — coś, nad czym od przeszło 30 lat mozola się bez większego powodzenia potężne instytucje naukowe całego świata.

Na czym ten „śmiesznie prosty” sposób polega? Do tej pory główny wysiłek urzeczywistnienia kontrolowanej syntezy termojądrowej kierowany był w stronę gigantycznych ciśnień i super wysokich temperatur. Bo wprawdzie rozpracowywano również od strony teoretycznej warunki zaistnienia „zimnej” syntezy jądrowej (np. w Instytucie Energii Jądrowej im. Kurczatowa w Moskwie), ale jednak w nauczaniu się wytwarzania i utrzymywania przez odpowiednio długi okres czasu rozgrzanej do milionów stopni plazmy widziano przyszłość.

W rozwój badań nad „gorącą” syntezą termojądrową włożone zostały olbrzymie pieniądze. Stać na nie było tylko najbogatszych: USA, ZSRR, plus niektóre kraje zachodnie, w tym Japonię. Ale okazało się w ostatnich latach, że budowanie potężnych „tokamaków” i innych urządzeń zdolnych do wytwarzania i utrzymywania przez ok. 1 sekundę plazmy, przekracza możliwości finansowe nawet światowych gigantów. Dlatego właśnie pojawiła się idea zbudowania międzynarodowego gigantycznego „tokamaka” o kryptonimie INTOR. Było to nawet tematem rozmów prowadzonych przez Michaiła Gorbaczowa i Ronalda Reagana podczas spotkania na szczycie w Genewie.

Przypominam te fakty, aby tym wyraźniej uwidocznic genialność metody zaproponowanej przez Fleischmanna i Ponsa. Ich baza doświadczalna okazała się niezwykle prosta: szklana wanienska napełniona „ciężką wodą”, dwie elektrody — z platy-

ny i palladu, zasilacz oraz odpowiednia aparatura pomiarowa (tak to w każdym razie przedstawili w pierwszych relacjach, pełny opis odkrycia zapowiedziano bowiem w majowym numerze miesięcznika popularno-naukowego „Nature”). Pod wpływem przepływu prądu zachodzi w tym zestawie proces elektrolizy i syntezy jąder, gromadzących się w elektrodzie z palladu. Świadczy o tym wydzielanie się dużych ilości ciepła oraz pojawienie się wolnych neutronów. Zdaniem autorów, jedynym wyjaśnieniem tych zjawisk może być zachodząca pod wpływem elektrolizy deuteru reakcja syntezy jąder atomowych.

Tę sensacyjną wiadomość, światowa społeczność uczonych przyjęła z olbrzymim aplauzem — ale i z pewną dozą rezerwy. W nauce bowiem o tym, że jakieś zjawisko zachodzi obiektywnie i może być uznane za prawidłowość decyduje jego powtarzalność. Zaczęło się nerwowe oczekiwanie — czy komuś uda się powtórzyć eksperyment Fleischmanna-Ponsa? Po kilku dniach wreszcie odetchnięto z ulgą — sypnęły się informacje o odtworzeniu eksperymentu w różnych laboratoriach świata, w tym w katedrze fizyki doświadczalnej Uniwersytetu w Debreczynie (Węgry).

Oczywiście, od pierwszego eksperymentu w laboratorium uniwersytetu w stanie Utah (Fleischmann jest na nim dziekanem wydziału chemicznego) do zbudowania elektrowni przemysłowej wykorzystującej energię powstającą z elektrolizy „ciężkiej wody” droga jest daleka. Ale rokuje ona (jeśli zjawisko wykryte przez pñów F-P zostanie ostatecznie potwierdzone) perspektywy wprost oszałamiające.

Łatwo też wykazać, że łączny wpływ trzech głównych przełomów naukowo-technologicznych drugiej połowy XX wieku — komputerów osobistych, nadprzewodnictwa w temperaturze pokojowej i „zimnej” syntezy termojądrowej — tworzy podstawy pod nowy, skokowy etap rozwoju cywilizacyjnego. A to wszystko, drogi Czytelniku, stało się (a raczej powiedzmy ostrożniej: staje się) nie w zamierzczłej historii lat 60-tych, tylko w czasach, które świetnie pamiętasz. Być świadkiem przełomu w historii nauki i historii cywilizacji — czyż to nie jest fascynujące?

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

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

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

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

Cena 150 zł.  
Skład techniki CRT-200, przygotowalnia offsetowa i druk: PRASOWE ZAKŁADY GRAFICZNE RSW „PRASA-KSIAZKA-RUCH” w Ciechanowie, ul. Sienkiewicza 51.

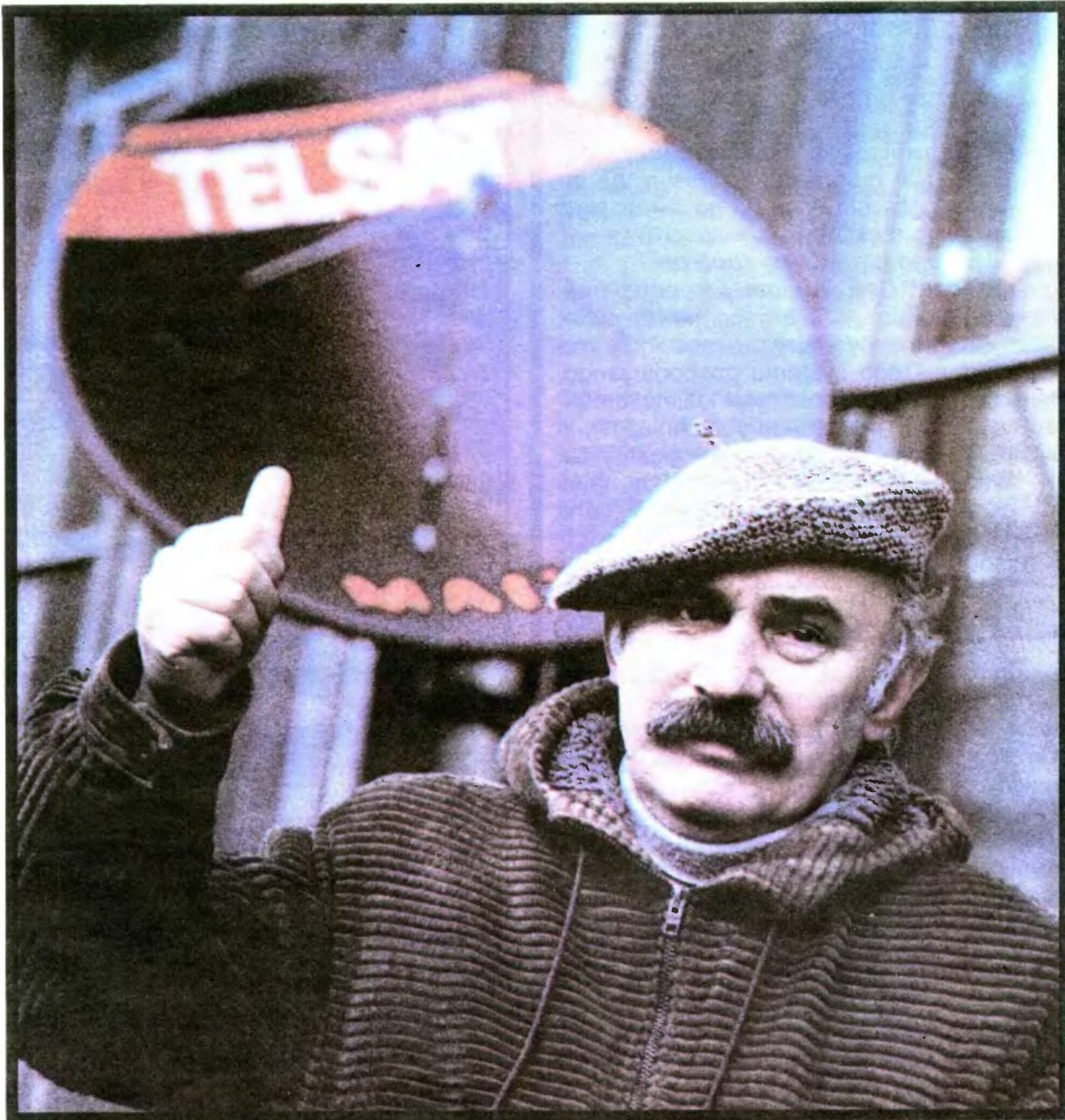
Nr zlecenia 12439 n. 150 000 egz. A-111

## ZA MIESIĄC:

- CeBIT '89
- Monitory na Atari i Commodore
- kilka uwag o Kyan Pascalu
- weryfikator i MicroDraft dla właścicieli Amstrada
- klan kalkulatora
- przedstawiamy pierwsze kluby w konkursie o „Złotą Dyskietkę Bajtka”
- nowy konkurs, gry, mikrociekawostki







**ROZMOWA  
z doc. dr hab. Lechem  
Zacherem  
kierownikiem Zespołu  
Cybernetyki Politycznej  
i Prognozowania  
w Międzyuczelnianym  
Instytucie Nauk  
Politycznych UMCS.**

— *Panie docencie, informatyka, elektronika i telekomunikacja nie znalazły się wśród przyjętych przez rząd priorytetów rozwojowych Polski. Jakie konsekwencje może to za sobą pociągnąć? Czy możliwy jest w ogóle dzisiaj rozwój kraju bez rozwoju tych dziedzin?*

— Jest to kwestia wizji przyszłości jaką dany kraj przed sobą widzi. Czy wizja społeczeństwa informacyjnego sprawdziła się na świecie? Oczywiście! Kraje najwyżej rozwinięte rozbudowują sektor „informacyjny”, jest on dzisiaj u nich większy od sektora przemysłowego.

My natomiast, przypomnę, rozbudowaliśmy ponad miarę sektor przemysłowy, podpierając się przy tym cytatami z klasyków, że przemysł jest ważniejszy od sektora usług. Konsekwencją była więc monstrualna rozbudowa struktur biurokratyczno-politycznych związanych z przemysłem paliwowo-wydobyczym i jednocześnie ochraniających ten przemysł. Struktury te w dalszym ciągu dominują w naszym kraju i nie dopuszczają do rozwoju tego wszystkiego, co nazywamy przemysłami informacyjnymi.

— *Czy rzeczywiście wszystkim jest winien kompleks paliwowo-surowcowy?*

— Może nie wszystkim, ale ci panowie eksploatują i wyniszczają nasz kraj bez litości. Przecież to właśnie między innymi dochodowa telekomunikacja — przez podział budżetu państwa — dofinansowuje deficytowe górnictwo i niszczone przez nie środowisko. Reforma gospodarcza musi to zmienić. Na razie jednak ugrupowania przemysłowe i decydenckie nie chcą uwierzyć w to, że dzięki oszczędnościom i zrationalizowaniu zużycia energii można zmniejszyć jej zużycie o jedną czwartą, a nawet jedną trzecią. Zapytajmy bowiem jakie konkretne działania podjęto w zakresie racjonalizacji zużycia energii. Jest z tym trochę tak jak z ochroną środowiska: trochę się mówi, rozkłada się ręce i... nic się nie dzieje. Tak jest w wielu dziedzinach, choćby w budownictwie — upada ono w gwałtownym

tempie, a resort nie chce zrezygnować z wielkiej — zresztą bardzo energochłonnej — płyty!

— Tymczasem trzeba wyraźnie zdać sobie sprawę z tego, że nie można osiągnąć nowych celów przy zachowaniu starej struktury. W gruncie rzeczy prowadzi się politykę zaduszania elektroniki, informatyki i innych nowoczesnych dziedzin przemysłu, gdyż nie pozwala im się osiągnąć „masy krytycznej”, czyli takiej skali, przy której rozpoczyna się przyspieszony rozwój.

— *No cóż, elektronika i informatyka znalazły się poza priorytetem!*

— Może nawet nie warto mówić o priorytetach, bo przecież z priorytetu ochrony środowiska nic dla środowiska nie wynika! Mówmy o tym, aby przy pomocy elektroniki, informatyki, telekomunikacji osiągnąć wzrost wydajności pracy, osiągnąć wyższy standard życia, zacząć wreszcie skutecznie ochraniać środowisko, w którym żyjemy. Pamiętajmy też, że tylko dzięki tym nowoczesnym przemysłom możemy nawiązać kontakt ze światem. A bez tego kontaktu dalszy rozwój nie jest po prostu możliwy. Może być tylko pogłębianie kryzysu i dalszy dryf, tak jak dryfuje pozbawiona motoru łódź. Nie można w nieskończoność podtrzymywać deficytowych gałęzi i branż, bo to nic dobrego nie daje. Nie można też liczyć na żadne poważniejsze kredyty, bo nikt, choćby z przyczyn politycznych nam ich nie da. Duże kredyty zagraniczne skończyły się dla nas definitywnie w końcu lat 70-tych i trzeba wreszcie przyjąć to do wiadomości.

A współpracy między krajami socjalistycznymi brak. Potrafiliśmy kiedyś krytykować rzekome rozbieżności pomiędzy krajami zachodnimi, przeciwstawiając im naszą jedność i współpracę, tymczasem Europa Zachodnia jednoczy się w stopniu nigdzie jeszcze na świecie nie spotykanym, a my musimy cały ten proces rozpoczynać niemal od nowa.

— *Niedawno naukowcy radzieccy sformułowali program tworzenia budowy „społeczeństwa informacyjnego” w ZSRR.*

— Bo to jest prawidłowość cywilizacyjna. Cały świat podąża w tym kierunku.

— *Cały?*

— Cały świat rozwinięty. U nas tymczasem niektórzy wyobrażają sobie, iż możemy się rozwijać w oderwaniu od świata: inwestować po staremu, przestać na małym, tu i tam coś ulepszyć, trochę tylko rozwinąć prywatną przedsiębiorczość... Byłoby to możliwe, gdyby cały świat stał w miejscu lub się cofał, jak to wyobrażali sobie nasi ideologowie w latach 50-tych i 60-tych, a nawet w latach 70-tych. Zakładano, że świat kapitalistyczny pogrzeży

BLIŻEJ  
ŚWIATA?



się w kryzysie strukturalnym, my natomiast utrzymamy ówczesne tempo rozwojowe, wyjdziemy na czoło... i zostaniemy drugą Japonią. W praktyce mimo przepowiedni naszych ideologów, wszystko odbyło się inaczej. Dzięki rozwijaniu nowoczesnych technologii (high technology), świat zachodni bardzo poszedł do przodu. Nie tylko Europa Zachodnia i Japonia, ale nawet kraje, które uważaliśmy za zacofane — myślę o tzw. młodych smokach z basenu Pacyfiku. To od nich dzisiaj kupujemy po niezwykle niskiej cenie niezawodne układy scalone, komputery, telewizory, telefaksy, to wszystko co składa się na pojęcie rewolucji mikroelektronicznej.

— **Singapur, Korea Południowa i Tajwan to jeszcze nie cały świat!**

— Proszę pana, dzisiaj nawet kraje w Afryce telefonizują się i informatyzują, przynajmniej enklawowo unowocześniają wybrane dziedziny swoich gospodarek. Sensację wzbudził niedawno w Uniwersytecie Warszawskim pewien stypendysta z Afryki, gdyż poszedł do Biblioteki Uniwersyteckiej i zaczął wszystkich pytać, gdzie tu znajdują się komputery, gdyż chciałby wypożyczyć książkę. Gdy zaczęto mu tłumaczyć, że u nas zamiast w komputerowym banku danych, informacji o książkach należy szukać w katalogach składających się z setek drewnianych pudełeczek i tysięcy papierowych fiszek, ów przedstawiciel Czarnego Łądu był święcie przekonany, że próbuje się z niego kpić, obraził się i podobno wrócił do swoich komputerów w Afryce!

Dzisiaj trzeba informatyzować się nie dlatego, że jest to modne, tylko dlatego, że tak jest wygodniej i taniej. Jest to też zgodne z trendami politycznymi na świecie, bowiem społeczeństwa chcą mieć szeroki dostęp do informacji. Bo na czym polega demokracja? Na tym, że obywatele uczestniczą w podejmowaniu decyzji. A mogą to czynić tylko pod warunkiem, że mają dostęp do odpowiednich informacji.

— **Mówimy o polskim zacofaniu w nowoczesnych na świecie dziedzinach przemysłu, a przecież jesteśmy jednocześnie krajem, w którym nastąpiła eksplozja zainteresowań informatyką, coraz więcej widać na dachach i balkonach naszych domów anten satelitarnych... Jak te dwa fakty pogodzić ze sobą?**

— Jest to rzeczywiście paradoks: w Polsce jest obecnie więcej komputerów i urządzeń wideo niż we wszystkich krajach socjalistycznych razem wziętych. Jest to wynikiem naszej fascynacji informatyką, a mówiąc szerzej — nowoczesnością. Informatykom się to nie podoba, bo oni chcieliby, żeby to wszystko było systemowe, kompatybilne itp. Jednak to nasycenie sprzętem od Sasa do Lasa posuwa nas ku nowoczesności, choć struktura tego sprzętu jest rzeczywiście nieprawdopodobna. Część tego sprzętu jest bardzo wysokiej jakości, nie można jednak wykorzystać jego możliwości, gdyż nie ma u nas sieci umożliwiających dostęp do banków danych z prawdziwego zdarzenia.

Ważne jednak, że młode pokolenie uczy się kultury technicznej, uczy się nowoczesności. I to uczy skutecznie, o czym świadczą sukcesy naszych licznych elektroników i informatyków, którzy zdecydowali się pracować na Zachodzie.

— **Czy nie sądzi pan, że jest kolejnym paradoksem iż tak podstawowym obecnie sprzętem służącym komunikowaniu się, jak telefaks, wchodzi do naszych przedsiębiorstw dzięki firmom prywatnym, natomiast ociężały przemysł państwowy nie potrafi uruchomić jego produkcji?**

— Oczywiście. Dlatego trzeba dokonać takich zmian, żeby duży przemysł państwowy zaczął to wreszcie robić. Musi on zacząć pra-

cować na potrzeby rynku. Musimy zdać sobie wreszcie sprawę z tego, że przemysł państwowy jest ciągle podstawą gospodarki. Dlatego nie może on być poprzez podatki ograbiany z zysku, musi mieć możliwości inwestowania w swój rozwój. Fachowcy w tym dużym przemyśle muszą mieć zapewnione takie warunki, aby nie oplotało im się uciekać za granicę lub do spółek prywatnych.

— **Jest pan przeciwko spółkom?**

— Skądże! One są niezwykle potrzebne, wszędzie na Zachodzie tysiące małych dynamicznych firm decyduje o prężności i żywotności tamtejszego systemu gospodarczego, ale przecież o nowoczesności krajów, o obliczu tamtejszej infrastruktury technicznej, o wyglądzie rynku decydują takie giganty jak np. IBM. U nas funkcje tych gigantów musi pełnić silny przemysł państwowy. Żadna ustawa o przedsiębiorczości nie może zwolnić rządu z dbałości o rozwój — i to efektywnej i szybkiej — głównych gałęzi przemysłu.

— **Profesor Jarosław Swiderski z Instytutu Technologii Elektronowej alarmował niedawno na łamach prasy, że bez zbudowania wielkiej wytwórni układów scalonych wielkiej skali integracji zostaniemy wkrótce wypchnięci poza obręb krajów cywilizowanych!**

— Zbudowanie nowoczesnej infrastruktury informacyjno-telekomunikacyjnej jest bowiem dzisiaj podstawą rozwoju współczesnego świata. A do tego potrzebne są układy scalone. Może ktoś powiedzieć, że nie musimy się wcale do tego świata tak bardzo pchać, wystarczy spokojnie dreptać sobie z tyłu... Ale to nieprawda. Dlaczego skomputeryzowaliśmy w Polsce system rezerwacji biletów lotniczych? Gdyż nie mieliśmy innego wyjścia. Gdybyśmy tego nie zrobili — nie moglibyśmy latać po świecie, ani nikt nie miałby okazji zarezerwować biletu do nas. To samo dotyczy dzisiaj choćby telefaksów. To nie jest już żadna techniczna nowinka, czy też czyjaś fanaberia. Nie masz telefaksu — nie możesz już obecnie brać udziału w handlu światowym. Takie są realia!

— **A więc w co powinni inwestować dzisiaj młodzi czytelnicy „Bajtki”? Jakie kierunki aktywności powinni wybrać, aby jak najlepiej przygotować się do dorostego życia?**

— Powinni to robić, co już robią — uczyć się nowoczesności. Nauka i technika są obecnie najważniejszymi czynnikami rozwoju, nie mają one żadnych zamienników, nic nie jest w stanie ich zastąpić. Jednocześnie wiedza jest najważniejszym czynnikiem rozwoju człowieka. Przy tym ważne jest, że wiedza nie ma granic. To nie jest kopalnia węgla, gdzie kopujemy i w pewnym momencie uderzamy łopatą w kamień. W pokładach wiedzy można kopać bez końca. Nie powinni również bać się nowych problemów, kroczenia nieprzetartymi ścieżkami. Zespół japońskich naukowców pracujących nad skonstruowaniem komputera V generacji składa się z osób, które nie przekroczyły 28 roku życia. Nie mają oni może takiej erudycji i takiego doświadczenia, jak ich profesorowie, ale dysponują za to zasadniczą przewagą: nie wiedzą, że pewnych rzeczy nie da się zrobić!

Jeśli ktoś chce dokonać czegoś w przyszłości, powinien też przekroczyć barierę programu szkolnego, nastawić się na samokształcenie, bo wiedza szkolna jest zawsze wiedzą przestarzałą. Żeby iść naprzód, trzeba odbić się od tego, co już jest znane. A żeby dokonać w nauce i technice rzeczy nowych, trzeba też zazwyczaj zakwestionować to, co było do tego momentu. Oznacza to, że tworząc nowe społeczeństwo trzeba odważnie zakwestionować dotychczasowy, industrialny etap rozwoju.

Rozmawiał:

Waldemar Siwiński



**Własności operacyjne kalkulatora wyznaczają możliwości, jakie stwarza on osobie liczącej.**

Zależą one od kilku elementów związanych z budową kalkulatora i organizacją jego systemu operacyjnego, a przede wszystkim od:

- przyjętego systemu algebraicznego;
- wielkości i rodzaju dodatkowych pamięci czy specjalnych rejestrów;
- liczby i rodzaju wbudowanych instrukcji matematycznych (i nie tylko);
- rodzaju programowalności.

Wszystkie wymienione elementy zostaną omówione w kolejnych artykułach, gdyż poznanie możliwości kalkulatora umożliwi nam wybór modelu, który będzie najlepiej spełniał postawione przed nim wymagania. W dzisiejszym artykule zajmiemy się pierwszym punktem, czyli systemami algebraicznymi kalkulatorów.

Trzeba przyznać, że jeżeli nie brać pod uwagę np. przyzwyczajenia liczącego, to sam system algebraiczny ma najmniejszy wpływ na walory eksploatacyjne kalkulatora. Zrozumienie działania systemu algebraicznego pozwoli jednak na pełniejsze wykorzystanie kalkulatora i usprawnienie prowadzenia procesu obliczeniowego.

SYSTEM ALGEBRAICZNY kalkulatora to zbiór reguł dotyczący wykonywania za jego pomocą różnych operacji i funkcji. Określa on kolejność wprowadzania argumentów i wykonywania operacji. Z każdym systemem algebraicznym kalkulatora związana jest pewna minimalna liczba rejestrów pamięci (komórek pamięci) służących do zapamiętywania argumentów, na których wykonywane są operacje np. matematyczne. Pod pojęciem operacje matematyczne rozumiemy działania typu mnożenie, dodawanie itd. Z punktu widzenia układu logicznego kalkulatora instrukcja operacji matematycznej jest rozkazem wykonania operacji matematycznej. Z punktu widzenia użytkownika równoważne jest to wciśnięciu odpowiedniego przycisku lub kilku przycisków.

Po tym wstępie konieczne staje się wprowadzenie opisu dwóch rejestrów pamięci.



Rejestry te oznaczane są najczęściej przez X i Y (tak jak w komputerach). Występują one w każdym kalkulatorze i stanowią one podstawę praktycznego działania systemu algebraicznego. Każda liczba wprowadzana do kalkulatora trafia najpierw do rejestru X. Wszystkie działania jednoargumentowe wykonywane są bezpośrednio na argumentach znajdujących się w tym rejestrze. W operacjach dwuargumentowych drugi z argumentów przechowywany jest w rejestrze Y. W operacjach, w których ważna jest kolejność argumentów (np. przy potęgowaniu  $Y^X \neq X^Y$ ) należy zadbać o to by wprowadzać argumenty w określonej kolejności. Na ogół zawartość rejestru X wskazywana jest przez wyświetlacz kalkulatora. Wyniki kolejnych, pośrednich i ostatecznych operacji są zawsze umieszczane w rejestrze X, czyli spełnia on również rolę rejestru wyjściowego.

Rejestr Y jest drugim rejestrem służącym do prowadzenia operacji dwuargumentowych. W większości kalkulatorów nie istnieje bezpośredni dostęp do tego rejestru. Działanie tego rejestru wynika jedynie z zaprogramowanego w kalkulatorze algorytmu pracy — rejestr pracuje automatycznie.

Pierwszym z omawianych systemów algebraicznych będzie system ODWROTNEJ NOTACJI POLSKIEJ (RPN — Reverse Polish Notation). Podstawowe zasady systemu są następujące:

- operator wprowadzany jest po argumentach np. mnożenia  $a \times b$  wprowadzane jest jako  $a, b, \times$
- operacja jest wykonywana po wprowadzeniu operatora

W kalkulatorach o tym systemie algebraicznym zawsze jest używana pamięć w postaci stosu, zbudowana z kilku rejestrów. Rejestrów tych jest na ogół stała ilość (zwykle 4) choć w kalkulatorach HP-28 i późniejszych stos jest dynamiczny i zmienia się w miarę potrzeb od 0 aż do wyczerpania dostępnej pamięci. Stos umożliwia techniczną realizację odwrotnej notacji polskiej. Z RPN związane są następujące rodzaje instrukcji:

Instrukcja wprowadzania liczb na stos (ENTER). Służy ona do rozdzielania kolejnych liczb wprowadzanych na stos. W wypadku gdy po naciśnięciu klawisza funkcyjnego chcemy wprowadzić jakąś kolejną liczbę na stos, użycie tego klawisza staje się zbędne. Jak się za chwilę okaże klawisz ten nie jest tak często używany jakby się zdawać mogło na pierwszy rzut oka. Wykonanie dodawania czterech kolejnych liczb  $a, b, c, d$  może być wykonane następująco:  $a$  ENTER  $b$  ENTER  $c$  ENTER  $d$  ENTER  $+$   $+$   $+$  (operatorzy wprowadzane po argumentach). Zadanie to może być jednak wykonane znacznie prościej przy jednokrotnym użyciu klawisza ENTER:  $a$  ENTER  $b + c + d +$ .

Zakładając, że mamy doczynienia ze stosem składającym się z czterech rejestrów, oznaczenia kolejnych poziomów stosu są X Y Z T. Drugą grupą instrukcji związaną z RPN są instrukcje zmieniające między sobą zawartości tych rejestrów. Za pomocą instrukcji ROLL można dokonać cyklicznej wymiany rejestrów  $X - Y - Z - T - X$ . Istnieją instrukcje wykonujące te operacje w drugą stronę jak również parami np. zamiana X i Y (SWAP).

System odwrotnej notacji polskiej ma wiele zalet. Przede wszystkim jest on bardzo uniwersalny. Te same ciągi działań mogą być wykonane na różne sposoby, przy czym zawsze znane są częściowe wyniki operacji. Można więc zawsze wybrać najdogodniejszy sposób prowadzenia obliczeń. Stwierdzono również, że jest to system, w którym dla przeprowadzenia złożonych obliczeń wymagana jest najmniejsza liczba wykonywanych instrukcji (łącznie z manipulacją danymi i wynikami pośrednimi). Tak więc w kalkulatorach programowalnych, w których dostępna pamięć przeznaczona na przechowywanie programu jest ograniczona np. do 100 kroków cecha ta odgrywa często decydującą rolę.

System odwrotnej notacji polskiej stosowany jest w wielu kalkulatorach naukowych i programowalnych np. w kalkulatorach firmy Hewlett-

Packard. Ta ostatnia firma w najnowszej generacji swoich kalkulatorów wzięła pod uwagę przyzwyczajenia użytkowników kalkulatorów i wypuściła model umożliwiający wymienne korzystanie z RPN i systemu z algebraicznym wprowadzaniem argumentów (o których za chwilę). Zaletą takiego rozwiązania, oprócz indywidualnych przyzwyczajzeń do tego czy innego systemu, jest fakt, że wzory książkowe można wprowadzać tak jak występują one w książce bez żadnych przekształceń związanych ze specyfiką RPN, jak również umożliwia prowadzenie obliczeń RPN i korzystanie z wyników pośrednich itp.

Systemy z algebraicznym wprowadzaniem argumentów (AES — Algebraic Entry System). Systemy te niekiedy nazywane są „prawdziwymi systemami algebraicznymi”. Określenie to bierze się stąd, że zasadnicza idea tych systemów zakłada maksymalne podobieństwo procedury wprowadzania argumentów i operatorów do postaci szkolnego, algebraicznego zapisu działań, czyli w operacjach dwuargumentowych obowiązuje następująca kolejność: argument, operator, argument, operator... Jak wynika z takiego ciągu ostatni jest zawsze operator dlatego staje się konieczne wprowadzenie dodatkowej instrukcji zamykającej ciąg działań i powodującej wykonanie ostatniej operacji dwuargumentowej. Taką rolę pełni instrukcja  $=$ . W ten sposób uzyskuje się całkowite podobieństwo do zwykłego zapisu stosowanego w algebrze. Dodanie czterech liczb  $a, b, c, d$ , wygląda więc następująco:  $a + b + c + d =$ . W systemach tych chodzi po prostu o to, aby wartość wyrażenia matematycznego zapisanego na kartce papieru można było obliczyć przepisując na klawiaturę bez przekształceń i zmian kolejności działań. Oczywiście najprostsza wersja systemu algebraicznego nie nadaje się do szybkiego wykonywania złożonych obliczeń. Powstaje konieczność wprowadzenia odpowiedniej liczby rejestrów mogących przechowywać pośrednie wyniki obliczeń. Prowadzi to do rozbudowy samego systemu i wprowadzenia dodatkowych instrukcji manipulacji danymi itp.

Wśród produkowanych obecnie minikalkulatorów można spotkać następujące wersje omawianego systemu:

- podstawowy system algebraiczny
- podstawowy system z preferencją działań
- zwykły system nawiasowy
- system nawiasowy z preferencją działań (AOS — Algebraic Operating System)

Systemy nawiasowe wprowadzają ściśle określoną liczbę par nawiasów  $()$  lub  $[\ ]$  umożliwiających wydzielenie operacji, które mają być wykonane przed pozostałymi. Nawiasy te mogą być wykonane przed pozostałymi. Nawiasy te mogą być otwierane kilkakrotnie i tyleżkrotnie zamykane (wielopoziomowo). Wyniki operacji w nawiasach traktowane są jako nowe argumenty dla ciągu nie wykonanych jeszcze operacji. Warto przy kupnie kalkulatora zainteresować się jaka jest dopuszczalna liczba par nawiasów, gdyż często decyduje ona o ułatwieniach związanych z prowadzeniem żmudnych obliczeń. Warto podkreślić, że kolejnym poziomom nawiasów w RPN odpowiadają kolejne rejestry stosu, a instrukcji otwierania nawiasu odpowiada instrukcja ENTER w RPN odpowiadają kolejne rejestry stosu, a instrukcji otwierania nawiasu odpowiada instrukcja ENTER w RPN. Instrukcja  $=$  zamyka automatycznie wszystkie do tej pory otwarte nawiasy (niedomknięte). Nawiasy zamykające umieszczane są na końcu ciągu obliczeń. Tak więc wynika stąd nowy wniosek, że programując pewne zagadnienie można dla oszczędzenia pamięci pominąć zamykające nawiasy gdyż system sam się nimi zajmie. Systemy te nie są jednak konsekwentnie zrealizowane od początku do końca. Sprzeczność ujawnia się w operacjach jednoargumentowych, w których operator wprowadzany jest po argumentach (tak jak w RPN). Jedynym systemem konsekwentnie realizującym zbliżenie wprowadzania ciągu działań do kalkulatora z działaniami zapisanymi na kartce papieru jest system AER (Algebraic Expression Reserve) stosowany w wie-

lu kalkulatorach doskonałej firmy japońskiej SHARP.

W systemach z preferencją działań (naukowe kalkulatory Texas Instruments, Casio, SHARP) wprowadzono dodatkowe reguły określające zasady pierwszeństwa wykonywania operacji tzn. matematyczną hierarchię działań. (W zwykłych systemach bez preferencji kolejność działań zgodna jest z kolejnością zapisu). Kolejność ta jest zgodna z zasadami przyjętymi w matematyce: najpierw wykonywane są wszystkie operacje jednoargumentowe (obliczanie funkcji trygonometrycznych, zmiana znaku wyrażenia), następnie pierwiastkowanie i potęgowanie, później dzielenie i mnożenie, a na końcu dodawanie i odejmowanie. Jeśli w systemie występują dodatkowo nawiasy to obliczenia rozpoczynane są od wewnętrznej pary nawiasów. W nawiasach występuje identyczna preferencja działań. Stąd ciąg działań  $2 + 3 \times 4^2$  przy uwzględnieniu preferencji działań wyglądać będzie następująco  $2 + 3 \times 4^2 =$  i obliczona wartość wyniesie 50. Należy jednak pamiętać, że system z preferencją działań jak i pozostałe systemy z algebraicznym wprowadzaniem argumentów, jakkolwiek łatwe do opanowania charakteryzują się mniejszą elastycznością niż system odwrotnej notacji polskiej. Jeśli w trakcie obliczeń stosuje się zasadę książkową prowadzenia obliczeń „od lewej do prawej strony” zapisu wzoru obliczeniowego, konieczne staje się wprowadzenie nawiasów lub przekształcenie wzorów. Tę ostatnią uwagę należy traktować jako sprzeczną z ideą systemu — zakładającą bezpośredni przekład zapisu wzoru na ciąg operacji kalkulatorowych.

Na zakończenie omawiania systemów algebraicznych wypada dokonać ich porównania. Zadanie to nie jest proste, brak jest bowiem jednego kryterium porównawczego. Największe szanse uproszczenia procesu prowadzenia obliczeń stwarzają systemy najbardziej złożone, w których działaniem zaangażowana jest maksymalna liczba dodatkowych rejestrów, funkcjonujących w sposób automatyczny, a więc algebraiczny system nawiasowy z rejestrami stosu. Z drugiej strony, jeśli założyc maksymalnie skomplikowany charakter obliczeń, to również rozstrzygnięcie nie jest proste. Jak już wspomniano RPN umożliwia przeprowadzenie obliczeń przy mniejszej ilości wciskanych klawiszy. Niemniej na etapie skomplikowanych obliczeń w grę wchodzi psychologiczne względy przyzwyczajenia. Systemy zbliżone do „szkolnego” sposobu zapisu są łatwiejsze do opanowania, szczególnie przy pierwszym zetknięciu z kalkulatorem. Ponadto korzyści z bardziej oszczędnego systemu odwrotnej notacji polskiej stają się dopiero zauważalne przy odpowiednim wyborze sekwencji operacji co z kolei wymaga od liczącego pewnej minimalnej rutyny w postępowaniu się systemem, inaczej bowiem zastanawianie się nad kolejnymi operacjami może zniweczyć osiągnięte oszczędności płynące z lepszego algorytmu obliczeń (inaczej wygląda ten problem przy pisaniu programu i następnie wielokrotnym jego użyciu). Aby na koniec wyrobić sobie pogląd na efektywność poszczególnych typów systemów algebraicznych, porównajmy ciąg operacji kalkulatorowych potrzebnych do obliczenia wyrażenia

$$\sqrt{((a+b)(c+d)^e)(f+g)^h - i}$$

Dla RPN:  $c$  ENTER  $d + e$   $Y^X$   $b \times a + f$  ENTER  $g + h$   $Y^X$   $\times i$   $Y^X$   $- j$

W tym ciągu obliczeń należało wykonać 12 instrukcji operacji matematycznych (nie uwzględniając operacji wprowadzania argumentów pośrednich dla obu systemów).

Dla AOS:  $[[a + b \times [c + d] Y^X e] \times [f + g] Y^X h] Y^X i - j =$  Tutaj potrzebowano aż 19 instrukcji operacji. Liczbę tę można zmniejszyć do 15 zmieniając kolejność działań i używając  $=$  zamiast  $]$

#### Literatura:

Kozarski; Szurmak „Minikalkulatory w Obliczeniach Naukowych i Technicznych”  
Instrukcje obsługi kalkulatorów SHARP, Hewlett-Packard i Texas Instruments.

*Dominik Falkowski*



**Biblioteka Action!** zawiera wprawdzie dziewięć procedur wejścia/wyjścia, lecz taka ich liczba nie może zadowolić wytrawnego programisty. Wypada więc zająć się jej uzupełnieniem.

Znajdujące się w **Action!** procedury to: Print, Input, Put, Get, Open, Close, Note, Point i XIO. Działają one identycznie jak ich odpowiedniki w Basicu. Brakuje tu procedur służących do pracy ze stacją dysków — Basic zresztą też ich nie posiada, są natomiast w Turbo Basicu.

## Procedury podstawowe

Pierwsze pięć procedur uzyskamy bardzo łatwo wykorzystując istniejącą już procedurę XIO. Sięgnijmy w tym celu do opisu tej instrukcji dla Basicu, który opublikowany został w „Bajtku” 3/88. Podane tam parametry dla XIO obowiązują we wszystkich językach programowania Atari, w których ta instrukcja (lub procedura) się znajduje.

W ten sposób tworzymy procedury zmiany nazwy pliku (Rename), skasowania pliku (Delete), zabezpieczenia (Lock) i odbezpieczenia pliku (UnLock) oraz formatowania dyskietki (Format). W każdym przypadku do przeprowadzenia operacji wykorzystywany jest kanał 5 IOCB, który jest rzadko używany w programach. Uważny Czytelnik zauważył zapewne, że napisana w ten sposób procedura formatowania dyskietki daje zawsze rozszerzoną gęstość (1010 sektorów). Aby uzyskać pojedynczą gęstość należy zmienić w procedurze Format 254 na 253. Oczywiście można też wprowadzić wybieranie gęstości przy wywoływaniu procedury, ale to zostawiam już Czytelnikom (niech się również wykażą).

```
;Input/Output w Action!
;Wojciech Zientara
;Copyright (c) Bajtek
```

```
;Zmiana nazwy pliku dyskowego
;np. Rename("D1:STARY.ACT, NOWY.ACT")
```

```
PROC Rename(BYTE ARRAY plik)
```

```
    XIO(5,0,32,0,0,plik)
RETURN
```

```
;Skasowanie pliku dyskowego
;np. Delete("D2:PROGRAM.ACT")
```

```
PROC Delete(BYTE ARRAY plik)
```

```
    XIO(5,0,33,0,0,plik)
RETURN
```

```
;Zabezpieczenie pliku dyskowego
;np. Lock("D1:PROGRAM.ACT")
```

```
PROC Lock(BYTE ARRAY plik)
```

```
    XIO(5,0,35,0,0,plik)
RETURN
;Odbezpieczenie pliku dyskowego
;np. UnLock("D1:PROGRAM.ACT")
```

```
PROC UnLock(BYTE ARRAY plik)
```

```
    XIO(5,0,36,0,0,plik)
RETURN
```

```
;Formatowanie dyskietki
;np Format("D1:")
```

```
PROC Format(BYTE ARRAY stacja)
```

```
    XIO(5,0,254,0,0,stacja)
RETURN
```

## Transmisja bloku

Często występuje konieczność zapisania lub odczytania znacznej liczby bajtów umieszczonych kolejno w pamięci (bloku pamięci). Przykładem może tu być zapis lub odczyt zawartości pamięci obrazu. Najprościej jest to wykonać w pętli For/Next przy wykorzystaniu par procedur Peek/Put (zapis) lub Get/Poke (odczyt). Sposób ten jest jednak mało efektywny. W Basicu stosuje się więc w tym celu krótkie procedury maszynowe. Struktura **Action!** pozwala na uzyskanie tego samego efektu w znacznie prostszy sposób.

Najpierw trzeba zdefiniować procedurę CIO, która umożliwi dostęp do systemu operacyjnego komputera. Następnie napiszemy procedurę Block. Służy ona do umieszczenia w bloku IOCB wartości sterujących transmisją danych (zob. „Bajtek” 2/87) i wywołania procedury CIO.

Teraz można już ułożyć procedury przesyłania bloków danych BGet i BPut. Proszę zwrócić uwagę, że BGet została napisana jako funkcja. Dzięki temu po jej wywołaniu otrzymamy długość odczytanego bloku, a ponadto jest to zgodne z konwencją **Action!**, w którym wszystkie procedury odczytu są funkcjami (np. Input i Get). W obu przypadkach

```
;Podstawowa procedura I/O
```

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

```
;Pomocnicza procedura transmisji
```

```
CARD FUNC Block(BYTE kan,tryb,
    CARD adr,dlbuf)
```

```
    TYPE IOCB=[BYTE id,num,cmd,stat
        CARD badr,padr,blen
        BYTE a1,a2,a3,a4,a5,a6]
    IOCB POINTER iptr
```

```
    kan==&$07
    iptr=$340+(kan LSH 4)
    iptr.cmd=tryb
    iptr.blen=dlbuf
    iptr.badr=adr
    CIO(0,kan LSH 4)
RETURN(iptr.blen)
```

```
CARD FUNC BGet(BYTE kan CARD adr,dlug)
```

```
CARD temp
```

```
    temp=Block(kan,7,adr,dlug)
RETURN(temp)
```

```
;Zapis bloku danych
;np. BPut(1,32768,1024)
```

```
PROC BPut(BYTE kan CARD adr,dlug)
```

```
    Block(kan,11,adr,dlug)
RETURN
```

parametrami podawanymi przy wywołaniu są: numer kanału IOCB, adres przesyłanego bloku oraz jego długość. Oczywiście kanał musi być przedtem otwarty przez Open.

## Procedury dyskowe

Dzięki łatwemu dostępowi do procedur systemowych **Action!** pozwala także na proste przeprowadzanie operacji bezpośrednio na całych sektorach dyskietki. W tym celu należy zdefiniować procedurę DSK, która wywołuje DSKINT. Normalne wywołanie przez tablicę skoków OS (adres \$E453) pozwala na pracę tylko ze stacją dysków numer 1. Zastosowałem więc wywołanie bezpośrednie (adres \$C6B3), lecz może ono nie działać na komputerach z inną wersją systemu operacyjnego. Drugą procedurą pomocniczą jest Sector, która ustala parametry, w bloku kontroli urządzeń BCD (analogicznie do procedury Block dla IOCB).

```
;Podstawowa procedura dyskowa
```

```
PROC DSK=$C6B5(BYTE akum)
```

```
;Pomocnicza procedura dyskowa
```

```
BYTE FUNC Sector(BYTE drv,tryb,
    CARD adr,secnum)
```

```
    TYPE DCB=[BYTE dev,num,cmd,stat
        CARD badr BYTE tout,unus
        CARD cntr,aux1]
    DCB POINTER dptr=$300
```

```
    drv==&$07
    dptr.cmd=tryb
    IF tryb=$50 OR tryb=$57
        THEN dptr.stat=$80
        ELSE dptr.stat=$40
```

```
    FI
    dptr.aux=secnum
    dptr.badr=adr
    drv==+$30
    DSK(drv)
RETURN(dptr.stat)
```

```
;Odczyt sektorów dyskowych
;np. GetSec(1,$169,1536,2)
```

```
BYTE FUNC GetSec(BYTE stacja
    CARD nsek,adr,lsek)
```

```
    CARD i
    BYTE temp
```

```
    FOR i=nsek TO nsek+lsek-1
        DO
            temp=Sector(stacja,82,adr,i)
            adr==+128
```



```

OD
RETURN(temp)

;Zapis sektorow bez weryfikacji
;np. PutSec(1,4,$4000,5)

PROC PutSec(BYTE stacja
            CARD nsek,adr,lsek)

CARD 1

FOR i=nsek TO nsek+lsek-1
DO
Sector(stacja,80,adr,i)
adr==+128
OD
RETURN

;Zapis sektorow z weryfikacja
;np. WrtSec(1,4,$4000,5)

PROC WrtSec(BYTE stacja
            CARD nsek,adr,lsek)

CARD 1

FOR i=nsek TO nsek+lsek-1
DO
Sector(stacja,87,adr,i)
adr==+128
OD
RETURN

;Odczyt statusu stacji dyskow
;np. DrvStat(1)

BYTE FUNC DrvStat(BYTE stacja)

BYTE temp

temp=Sector(stacja,83,0,0)
RETURN(temp)
    
```

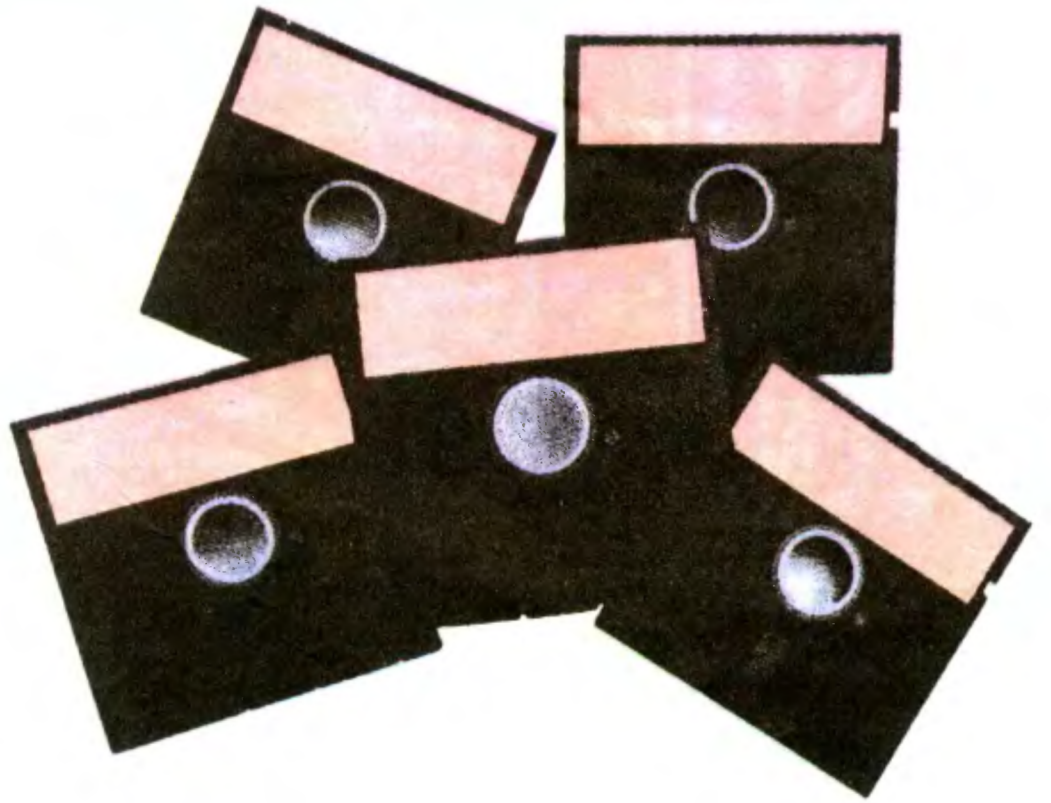
Teraz łatwo już napisać procedury zapisu i odczytu sektorów. Trzeba w nich zastosować pętle, które umożliwią transmisję kilku kolejnych sektorów przy jednym wywołaniu procedury. Procedura zapisu sektorów występuje w dwóch wariantach: z weryfikacją zapisu (WrtSec) i bez weryfikacji (PutSec). Parametrami tych trzech procedur są: numer stacji dysków, numer pierwszego sektora, adres bufora w pamięci i liczba przesyłanych sektorów.

Na końcu znajduje się jeszcze jedna prosta procedura, która służy do odczytu statusu stacji. **NIE** jest ona odpowiednikiem instrukcji STATUS w Basicu i dlatego ma inną nazwę. Po jej wywołaniu czterobajtowy status stacji dysków umieszczany jest w rejestrze DVSTAT (adres 746) i można go odczytać funkcją Peek. Parametrem tej procedury jest numer stacji. W odróżnieniu od procedur przesyłania bloków GetSec i DrvStat zwracają zawsze status operacji czyli liczbę 1.

Wojciech Zientara



# KATALOG DYSKIETKI



Oto program umożliwiający listowanie oraz drukowanie nazw plików zapisanych na dyskietce bez potrzeby wchodzenia do DOS-u.

```

10 DIM PL$(13)
20 OPEN # 1,6,0,"D:*.*)"
30 TRAP 60:INPUT # 1,PL$
40 PRINT PL$
50 GOTO 30
60 CLOSE # 1
    
```

Po przepisaniu i zapisie na dyskietce (SAVE) program uruchamiamy instrukcją RUN. I już mamy wykaz wszystkich plików znajdujących się na dyskietce.

Program ten można oczywiście rozwinąć — zależnie od potrzeb i pomysłowości. Podam dwa przykłady:

1. Aby nie liczyć „piechotą — palcem po ekranie” ile jest plików na dyskietce i ile możemy jeszcze dopisać — tworzymy licznik. Dopisujemy wiersz:

```
35 N=N+1
```

Wiersz 40 po małej zmianie będzie wyglądać następująco:

```
40 PRINT N, PL$
```

Po uruchomieniu programu każdy plik ma przypisaną kolejną liczbę (N) oraz nazwę wraz z informacją czy jest zabezpieczony.

2. Mając do dyspozycji drukarkę i chcąc otrzymać wydruk plików z dyskietki — tworzymy wiersz dodatkowy:

```
45 LPRINT N, PL$
```

Kasia Lekawska

## PRZEWIJANIE NAPISU

Głównym mankamentem programu „Płynący napis” z „Bajtka” 7/87 jest fakt, że przewijany napis może mieć długość jedynie 40 znaków.

Przedstawiony przeze mnie program pozwala wydłużyć go do 255 znaków i umieszcza tekst u dołu ekranu w trybie GRAPHICS 2. Jeśli komuś nie odpowiada ten tryb, może użyć trybu GRAPHICS 1, zmieniając w wierszu 20 POKE DL+25,87 na POKE DL+25,86. Procedura współpracuje z przerwaniami VBLK, wobec czego może się wykonywać w tym czasie program w Basicu. Należy uważać, aby po każdej instrukcji GRAPHICS 0 powtórzyć sekwencję rozkazów zawartą w wierszu 20.

Procedura odczytuje tekst począwszy od adresu 16384 gdzie jest on umieszczony w kodach ICODE (kod wewnętrzny ANTIC-a). Przed uruchomieniem należy więc go tam wpisać.

W tym celu należy wykonać POKE 82 0; wyczyścić ekran (klawisze SHIFT+CLE-

```

EB 1 REM PRZEWIJANIE NAPISU
UU 2 REM JAKUB GAWLIKOWSKI
XO 3 REM COPYRIGHT (C) BAJTEK
NJ 4 REM
NL 10 GRAPHICS 0:DL=PEEK(560)+256*PEEK(561)
KT 20 POKE DL+25,87:POKE DL+26,0:POKE DL+27,64:POKE DL+28,0
XN 30 FOR I=1700 TO 1709:READ A:POKE I,A:NEXT I
DK 40 FOR I=1536 TO 1577:READ A:POKE I,A:NEXT I
IP 50 POKE 8192,15:POKE 9193,0:POKE 8194,64:A=USR(1700)
ZH 60 DATA 104,162,6,160,0,169,7,76,92,228
MT 70 DATA 173,1,32,141,58,156,173,2,32,141,59,156,173,0,32,141,4,212,201,0,240,6,206,0,32
QK 80 DATA 76,98,228,169,15,141,0,32,238,1,32,238,1,32,76
    
```

AR); w 6-ciu pierwszych wierszach ekranu wpisać, co tylko na myśl przyjdzie (najlepiej jeśli pierwsze 21 znaków będzie spacjami); zjechać kursorem do około 6-ciu wierszy przed końcem ekranu i wpisać FOR I=0 TO 255: POKE 1638+I, PEEK(40000+I):NEXT I. Gdy pojawi się komunikat READY, uruchomić program (RUN).

Jakub Gawlikowski  
(lat 14)



ZOSTAŃ  
NIEŚMIE-  
RTELNYM!

(9)

UWAGA panowie! Ogłaszam wspaniałą do nasze: ubruk dofarł pierwszy list napisany przez dziewczynę!!! Jego autorka jest Ania Gabryszewska uczennica 8 klasy Szkoły Podstawowej z Kielc. Ona jest wnie potrawki.

**AMAROUTE**  
Poszukujemy następujące sekwencji rozkazów: ADC #01 STA \$63AC (\$69,01,8D,AC,65; "CTRL-A" "CTRL-M" w neg. "c") zamieniamy drugi bajt ciągu na: \$00; "CTRL-." Jak pisze Ania, poprawka ta powoduje, że bohater gry nie traci energii w kontakcie z krzemowcami.

**CAVE LORD**  
Szukamy sekwencji: LDA \$0609, SEC, SBC #S0A, STA \$0609 (\$AD,09,06,38,E9,0A,8D,09,06; "w neg. "CTRL-I" "CTRL-F" "8" "i" w neg. "CTRL-J" "CTRL-M" w neg. "CTRL-L" "CTRL-F") zamieniamy szósty bajt na: \$00; "CTRL-." Szukamy kolejnej sekwencji: LDA \$0609, SEC, SBC #SOF, STA \$0609 (\$AD,09,06,38,E9,0F,8D,09,06; "w neg. "CTRL-I" "CTRL-F" "8" "i" w neg. "CTRL-O" "CTRL-M" w neg. "CTRL-L" "CTRL-F") zamieniamy szósty bajt sekwencji na: \$00; "CTRL-." Wynikiem tych zmian jest całkowite unieszkodliwienie ptaków.

Michał Wayda z Krakowa przysłał nam następujące POKE-I:

**RIVER RAID**  
W sekwencji rozkazów: DEY, STY \$6A (\$88,84,64; "CTRL-H" w neg., "CTRL-D" w neg., "i") zamieniamy pierwszy bajt na: \$EA, "j" w neg.

**ZEPPELIN**  
W rozkazie: DEC \$2584 (\$CE,84,25; "N" w neg., "CTRL-H" w neg., "SHIFT-5") zamieniamy pierwszy bajt na: \$AD; "w neg., lub zamieniamy całość na: trzy razy NOP (\$EA, "j" w neg.)

Od Dariusza Grabarczyka, ucznia I klasy LO w Białej Podlaskiej dowiedzieliśmy się, jak poprawić grę:

**MORKY**  
Rozkaz: DEC \$0615 (\$CE,15,06; "N" w neg., "CTRL-U" "CTRL-F") zamieniamy na: trzy razy NOP (\$EA; "j" w neg.)

A teraz porawki z własnych poszukiwań:

**H.E.R.O.**  
W rozkazie: DEC \$39 (\$C6,39; "F" w neg., "9") zamieniamy pierwszy bajt na: \$A5; "SHIFT-5" w neg. i mamy nieskończoną liczbę "życ".

Odszukujemy ciąg liczb: \$8D,C6,38; "CTRL-M" w neg., "F" w neg., "8") i zamieniamy w nim drugi bajt na: \$A5; "SHIFT-5" w neg. i mamy dowolną ilość "bomb".

**LOCO**  
Odszukujemy ciąg rozkazów: LDA \$4FB0, SBC #S01, STA \$4FB0 (\$AD,B0,4F,E9,01,8D,B0,4F; "w neg., "0" w neg., "0" "i" w neg., "CTRL-A" "CTRL-M" w neg., "0" w neg., "0") i zamieniamy bajty czwarty i piąty na: NOP (\$EA; "j" w neg.).

**PANIC EXPRESS**  
"Zycia": DEC \$62E (\$CE,2E,06; "N" w neg., "9" "CTRL-F") zamieniamy na trzy razy NOP (\$EA; "j" w neg.).  
"Czas": DEC \$9FA6 (\$CE,A6,9F; "N" w neg., "SHIFT-6" w neg., "CTRL-\*" w neg.). Zamieniamy jak wyżej.

**ZYBEX**  
"Zycia" gracza 1: DEC \$3C7F (\$CE,7F,3C; "N" w neg., "ESC-TAB") zamieniamy na trzy razy NOP (\$EA; "j" w neg.).

"Zycia" gracza 2: DEC \$3C88 (\$CE,88,3C; "N" w neg., "CTRL-H" w neg., "c"). Zamieniamy jak wyżej.

**BASIL**  
DEC \$13FD (\$CE,FD,13; "N" w neg., "ESC-CTRL-2", "CTRL-S") zamieniamy na: LDA \$D01F (\$AD,1F,D0; "w neg., "ESC-CTRL-\*", "P" w neg.). Poprawka ta da nam nieskończoną energię i pozwoli przerwać grę w dowolnym momencie przez jednoczesne wciśnięcie klawiszy "OPTION+SELECT+START".

Tomasz Wiśniewski

monitor

BUG/65

W odróżnieniu od assemblerów, monitory języka maszynowego w zasadzie nie służą do pisania programów, lecz do ich uruchamiania. Polega to na możliwości wykonywania badanego programu rozkaz po rozkazie oraz śledzenia uzyskiwanych wyników. Dzięki bezpośredniemu dostępowi do pamięci komputera monitory pozwalają nie tylko na poprawianie istniejących programów, lecz także na pisanie od początku krótkich procedur w języku maszynowym.

Istnieje wiele różnych monitorów (zwanymi także debuggerami), lecz są one do siebie bardzo podobne, więc przedstawię tylko trzy z nich. Pierwszym będzie doskonały program uruchamiający i monitor firmy OSS: **BUG/65**, który stanowi komplet z assemblerem **MAC/65**. Jest on dostępny jedynie w wersji dyskowej.

Jedną z najważniejszych cech **BUG/65** jest symulowanie przez niego rejestrów procesora i części pamięci komputera. Użytkownik i jego program pracują na „kopii” strony zerowej i samego procesora, ponieważ w rzeczywistości są one wykorzystywane przez sam monitor. Zarówno użytkownik, jak i badany program nie są w stanie rozpoznać różnicy bez zastosowania specjalnych sztuczek. Wynikiem tego jest przede wszystkim duży komfort pracy z **BUG/65** oraz możliwość odzyskania kontroli nad komputerem nawet w przypadku zawieszenia się programu.

Druga ważna uwaga dotyczy sposobu przedstawienia liczb, który jest odmienny od powszechnie stosowanego. W zasadzie **BUG/65** operuje na liczbach szesnastkowych, nie wymagają one więc specjalnego oznaczania. Natomiast liczby dziesiętne muszą być poprzedzone kropką (.), co pozwala komputerowi odróżnić je od liczb szesnastkowych.

ZAPIS I ODCZYT

Zanim przystąpimy do pracy nad programem, musimy go odczytać. Do tego celu służy polecenie **R** (Read), które posiada trzy odmiany. Wpisanie litery **R** i nazwy pliku poprzedzonej znakiem # powoduje odczyt wskazanego pliku do pamięci w miejsce określone przez nagłówek tego pliku. Jednocześnie adres wczytywania pliku jest umieszczony w rejestrze **PC** (Program Counter) użytkownika. W przypadku programu złożonego z kilku bloków w rejestrze **PC** znajdzie się adres ostatniego bloku. Możliwe jest także odczytanie programu do innego miejsca pamięci. W tym celu należy dodatkowo podać „odstęp”, który określa, o ile program ma być przesunięty w stosunku do oryginalnego adresu wczytywania.

Gdy mamy do czynienia z programem „całodyskowym”, to znaczy nie zapisanym w pliku, lecz w kolejnych, pełnych sektorach dyskietki, to do jego odczytania wykorzystujemy jedną z dwóch pozostałych odmian polecenia **R**. Umieszczenie bezpośrednio po literze **R** apostrofu (') oznacza odczyt sektora w pojedynczej gęstości, zaś umieszczenie cudzysłowu (") — w podwójnej gęstości. W obu przypadkach parametrami polecenia są kolejno: numer pierwszego odczytywanego sektora, adres bufora (miejsca umieszczenia w pamięci odczytanej informacji) i liczba odczytywanych sektorów. Jeśli pominiemy jeden z parametrów, to trzeba pominąć także parametry występujące po nim. Ich standardowymi wartościami (przyjmowanymi w przypadku braku parametru w poleceniu) są odpowiednio: sektor numer 1, adres początku **BUG/65** zwiększony o \$2000 oraz 1 sektor.

Zapis zawartego w pamięci programu jest wykonywany na polecenie **W** (Write). Dla operacji na całych sektorach polecenia (**W** i **W'**) są analogiczne jak przy odczycie. Podczas zapisywania pliku binarnego można dodatkowo

(poza specyfikacją pliku) określić adres początkowy i/lub końcowy zapisywanego obszaru pamięci. Dużym udogodnieniem jest możliwość dopisania pliku do pliku już istniejącego na dyskietce. Uzyskuje się to przez napisanie: **A** (Append) po literze **W**.

Pomocniczą funkcję przy operacjach zapisu i odczytu pełni polecenie **I** (Index). Służy ono do odczytania katalogu dyskietki umieszczonej w stacji dysków numer 1 (tylko) i nie wymaga żadnego parametru.

OPERACJE NA PAMIĘCI

Podstawowym zadaniem każdego monitora jest umożliwienie użytkownikowi „grzebania” w pamięci komputera, a więc także w umieszczonym tam programie. **BUG/65** jest wyposażony w liczne polecenia służące do wykonania takich funkcji.

Wyświetlenie zawartości komórek pamięci uzyskamy po podaniu polecenia **D** (Display) z następującym po nim adresem początkowym wyświetlania. Jeśli nie podamy adresu końcowego to zostanie wyświetlona zawartość ośmiu komórek, poczynając od adresu początkowego. Na ekranie otrzymamy przy tym adres początkowy, szesnastkowe wartości bajtów oraz odpowiadające im znaki kodu ATASCII. Gdy wyświetlana jest jednocześnie zawartość więcej niż ośmiu komórek, to na początku każdego wiersza podawany jest adres pierwszego bajtu w tym wierszu.

Do odzyskania w pamięci określonego ciągu bajtów wykorzystuje się polecenie **L** (Locate). Jego parametrami są: adres początkowy i końcowy przeszukiwanego obszaru oraz poszukiwany ciąg bajtów. Ciąg ten musi być wpisany w postaci dwucyfrowych liczb szesnastkowych oddzielonych spacjami. Ponadto w ciągu mogą występować „jokery”, czyli znaki zastępujące każdy inny bajt. Podobnie jak w DOS-ie znakiem tym jest gwiazdka (\*).

Gdy zachodzi potrzeba wypełnienia pewnego obszaru pamięci jedną wartością, to należy zastosować polecenie **F** (Fill). Po adresach początku i końca wypełnianego obszaru wpisujemy w nim żadaną wartość, a jeśli ją pominiemy, to cały obszar zostanie wypełniony zerami.



# KSIĄŻKI O ST (2)

**Przedstawione w 2/89 numerze „Bajtka” książki o Atari ST to tylko niektóre publikacje Data Becker, które poświęcono tym komputerom. Dla wszystkich interesujących się ST zamieszczamy krótkie opisy innych tytułów wydanych przez to wydawnictwo.**

### „BASIC Training Guide”

Bardzo interesujący podręcznik języka Basic na Atari ST. Uczy podstaw programowania, analizy rozwiązywanego problemu i budowy algorytmu. Czytelnik, który uważnie przeczyta tę książkę, bez trudu zbuduje sprawnie działający program posługując się komendami języka. „BASIC Training Guide” podpowiada, w jaki sposób należy tworzyć schematy blokowe, ułatwia poznanie podstawowych technik programowania. Opisano w niej procedury obsługi menu, procedury sortujące, tablice wielowymiarowe, sposób zarządzania zbiorami. Książka ma 312 stron.

### „Introduction to MIDI Programming”

Lata 80 przyniosły gwałtowny rozwój techniki komputerowej syntezy dźwięku i elektronicznych instrumentów muzycznych. Najtańsze urządzenia syntetyzujące dźwięk można kupić już za 250 \$. Nawet dziecko bez trudu połączy je z komputerem. Urządzeniem sprzęgającym instrument z Atari ST jest interfejs MIDI (Musical Instrument Digital Interface). Komputer w konfiguracji z syntetyzerem może zdziałać cuda: naśladować całą orkiestrę, zapamiętać melodię zagrąną na instrumencie, wydrukować jej zapis nutowy na pięciolinii, zamienić się w dowolny instrument muzyczny. W książce zamieszczono wykład podstaw programowania interfejsu MIDI i syntetyzera, opisano standard MIDI i zasady posługiwania się procedurami BIOS. Znajdziemy tu też listingi wielu programów muzycznych. Książka ma 256 stron.

### „BASIC to C”

Książka ta jest przeznaczona dla osób, które poznały już Basic i chciałyby wzbogacić swoje umiejętności programowania. Uczy ona posługiwania się językiem C, który dzięki licznym zaletom zdobył sobie duże uznanie u zawodowych programistów. Metoda wtajemniczenia czytelnika w arkanę C jest bardzo prosta. Obok przykładów w Basicu znajdują się odpowiadające im procedury w C. Zwięzłe porównanie zasadniczych różnic charakteryzujących oba rozwiązania pozwala, jak zapewniają autorzy, opanować podstawy C już po jednym dniu. Książka ukazuje korzyści wynikające z wyboru C jako narzędzia programowania, opisuje instrukcje, strukturę programu, podstawowe typy danych, metody wykonywania obliczeń arytmetycznych i błędy najczęściej popełniane przez osoby przyzwyczajone do Basicu.

### „3D Graphics”

Książka ta jest podręcznikiem programowania trójwymiarowej grafiki w języku maszynowym mikroprocesora 68000 Atari ST. Należy ją polecić wszystkim, których interesuje programowanie w asemblerze oraz grafika i animacja obrazu. Znajdujące się tu procedury można zastosować we własnych programach. Przyda się szczególnie matematykom zajmującym się trójwymiarowym opisem obiektów w różnych układach współrzędnych. Omówiono w niej, w jaki sposób należy radzić sobie w programie ze skalowaniem osi, 2 i 3-wymiarowymi transformacjami obrazu, przedstawianiem zakrytych krawędzi i niewidocznych płaszczyzn przedmiotu. Sporo miejsca poświęcono także animacji obrazu, obrotom obiektów oraz grafice w komputerowym wspomaganium projektowania CAD. Książka ma 351 stron.

### „ST Disk Drives: Inside and Out”

Podręcznik ten jest niezastąpionym źródłem informacji o dyskach elastycznych, twardym dysku i RAM dysku Atari ST. Znajduje się w nim kilkanaście programów użytkowych pozwalających efektywniej wykorzystywać te urządzenia. W książce podano definicje związane ze zbiorami sekwencyjnymi, o dostępie bezpośrednim, przykłady zarządzania zbiorami danych w Basicu, Pascalu, C i Fortranie. Dla programistów cennym materiałem jest dokładny opis boot sektora, parametrów BIOS, tablicy alokacji zbiorów, oraz konstrukcji, sygnałów i rozkazów kontrolera twardego dysku. Książka ma 403 strony.

Polecenie **M** (Move) służy do przemieszczania w pamięci bloków informacji (programu lub danych). Wymaga ono podania trzech adresów: początku i końca przemieszczanego obszaru oraz początku obszaru docelowego. **BUG/65** automatycznie rozpoznaje pokrywanie się obszarów i wykonuje przemieszczenie bezbłędnie zarówno w górę, jak i w dół pamięci.

Dokonywanie zmian w poszczególnych komórkach pamięci umożliwiają dwa polecenia **S** (Set number) i **A** (set Ascii). Różnica między nimi polega tylko na rodzaju danych — dla **A** są to znaki ASCII, a dla **S** liczby szesnastkowe. Po wpisaniu polecenia z adresem komórki i naciśnięciu spacji wyświetlana jest zawartość tej komórki. Teraz podkreślenie ( \_ ) powoduje przejście do poprzedniej komórki pamięci, spacja — do następnej, a RETURN — ponownie do tej samej. W każdym z wymienionych przypadków wyświetlany jest adres komórki i jej zawartość. Opuszczenie tych funkcji następuje po naciśnięciu klawisza ESC. Każdy inny klawisz naciśnięty po poleceniu **A** jest traktowany jako znak i wpisywany do podanej komórki pamięci. Dla wpisania znaków odpowiadających wymienionym wcześniej klawiszom funkcyjnym należy je poprzedzić znakiem @ (znak @ wpisujemy do pamięci jako @@). Po poleceniu **S** wpisywana jest do pamięci każda poprawna wartość liczbową (szesnastkowa).

Dodatkowym poleceniem, rzadko spotykanym w monitorach jest **C** (Compare). Wymaga ono podania adresu początku i końca jednego obszaru pamięci oraz początku drugiego. Polecenie to porównuje wskazane obszary pamięci i wyświetla wszystkie różnice się bajty wraz z ich adresami.

## ŚLEDZENIE PROGRAMU.

Przy uruchamianiu napisanego programu ważna jest możliwość dokładnego śledzenia jego przebiegu. I w tym wypadku **BUG/65** jest wyposażony w szeroki repertuar odpowiednich poleceń.

Podstawowym poleceniem jest tu **G** (Goto). Rozpoczyna ono wykonywanie programu użytkownika od podanego adresu lub od aktualnego stanu licznika **PC** (jeśli adres nie został podany). Przerwanie wykonywania tego programu jest możliwe przez ustawienie punktu zatrzymania (adres punktu poprzedzony znakiem @). Ponadto można ustalić dodatkowe warunki zatrzymania. **R** (Register) oznacza zatrzymanie we wskazanym punkcie, jeśli zawartość podanego rejestru będzie równa określonej wartości, np. RY = 00. **I** (Iteration) pozwala na zatrzymanie przy kolejnym wykonaniu wskazanego rozkazu, np. I=5 oznacza przerwanie programu przy

piątym przekroczeniu punktu zatrzymania.

Szczegółowe badanie przebiegu programu umożliwia polecenie **T** (Trace). Podana przy nim wartość licznika oznacza liczbę rozkazów programu, które mają być wykonane (brak = 1). Po każdym z nich wyświetlane są zawartości rejestrów użytkownika. Dodatkowo litera **S** (Subroutine) pozwala na pominięcie procedur wywoływanych rozkazem JSR i potraktowanie ich jako pojedynczych rozkazów. Przyspiesza to śledzenie i umożliwia skupienie się na zasadniczej treści programu, bez zwracania uwagi na procedury (pod warunkiem, że nie zawierają one błędów).

Wykonanie procedury użytkownika osiąga się poleceniem **U** (User), po którym podajemy adres procedury. Do procedury tej można przekazać parametr liczbowy, którego starszy bajt jest przepisywany do rejestru Y, a młodszy do rejestru X. Procedura musi kończyć się rozkazem RTS, co zapewnia powrót do **BUG/65**.

Polecenie **V** (View) nie wymaga parametrów i wyświetla aktualny stan rejestrów użytkownika. Kolejno są to zawartości akumulatora, rejestru X, rejestru Y, wskaźnika stosu, rejestru statusu i licznika programu. Na końcu podawany jest rozkaz znajdujący się w komórce wskazanej przez licznik programu.

Do zmiany zawartości jednego z rejestrów użytkownika służy polecenie **X**, po którym należy podać jednoliterowy skrót nazwy rejestru (A, X, Y, S, P i F). Działanie tego polecenia jest analogiczne do poleceń **A** i **S**.

Gdy zaistnieje konieczność zdisasembrowania (zamiany na kod źródłowy) fragmentu programu, to użyjemy polecenia **Y**. Powoduje ono wyświetlenie rozkazów programu zawartego w podanym obszarze.

Funkcję odwrotną pełni polecenie **Z**, które wywołuje podręczny asembler. Umożliwia on wpisanie do pamięci krótkiego programu. Można przy tym stosować etykiety postaci „Ln”, gdzie n jest cyfrą od 0 do 9. Poza normalnymi rozkazami asembler rozpoznaje znak „/” z adresem jako polecenie zmiany zawartości licznika programu (jak „\*=” w **MAC/65**) oraz znak „+” jako polecenie wpisania jednego bajtu bezpośrednio do pamięci (jak „.BYTE”).

## FUNKCJE POMOCNICZE

Dla ułatwienia pracy **BUG/65** jest wyposażony w różne polecenia pomocnicze. Nie będziemy omawiać wszystkich, gdyż wykracza to poza ramy naszego opracowania. Zostaną one opisane, gdy zaistnieje potrzeba ich wykorzystania.

Przy pracy z liczbami szesnastkowymi często zachodzi koniecz-

ność przeliczenia ich na wartości dziesiętne lub wykonania prostego obliczenia. Polecenie **K** (Konvert) zamienia podaną po nim liczbę szesnastkową na dziesiętną. Odwrotna operacja jest niemożliwa i trzeba zastosować sztuczkę, np. podać polecenie **D** z adresem dziesiętnym, co spowoduje wyświetlenie zawartości pamięci i adresu w postaci szesnastkowej. Drugim użytecznym poleceniem jest **H** (Hex), po którym podajemy dwie liczby szesnastkowe. W wyniku otrzymamy kolejno wartości sumy i różnicy tych liczb.

Polecenie = wyświetla ostatnio wprowadzone polecenie i umożliwia wykonanie go, np. z innymi parametrami. Samo naciśnięcie klawisza RETURN powoduje powtórne wykonanie ostatnio wprowadzonego polecenia (lub poleceń, jeśli było ich kilka w jednym wierszu).

Podczas pracy z **BUG/65** możliwe jest utrwalenie jej przebiegu i wyników na drukarce. Służy do tego celu polecenie **P** (Print), które ustala urządzenie do wyprowadzania wszystkich komunikatów i innych informacji. Litera **S** (Screen) oznacza tu ekran, a **P** (Printer) drukarkę. Można oczywiście uruchomić oba urządzenia jednocześnie.

Ostatnim poleceniem jest **Q** (Quit), które powoduje zakończenie pracy **BUG/65** i powrót do DOS XL. Ponowne uruchomienie monitora wymaga odczytania go z dyskiety.

Wojciech Zientara

## SKŁADNIA POLECEŃ BUG/65:

- A adres
- C adr\_pocz\_1 adr\_końc\_1  
adr\_pocz\_2
- D adr\_pocz [adr\_końc]
- F adr\_pocz adr\_końc [wartość]
- G [adr\_pocz] [@ adr\_stop  
[Rn=wartość] [I=iteracja]]
- H liczba\_hex\_1 liczba\_hex\_2
- K liczba\_hex
- L adr\_pocz adr\_końc bajty
- M adr\_pocz adr\_końc adr\_docelowo
- P [S] [P]
- R [odstęp] #spec\_pliku
- R' [num\_sekt [adr\_bufora [liczba\_sekt]]]
- R'' [num\_sekt [adr\_bufora [liczba\_sekt]]]
- S adres
- T [S] [licznik]
- U adres [parametr]
- W [:A] [adr\_pocz] [adr\_końc] #spec\_pliku
- W' [num\_sekt [adr\_bufora [liczba\_sekt]]]
- W'' [num\_sekt [adr\_bufora [liczba\_sekt]]]
- X [A] [X] [Y] [S] [P] [F]
- Y adr\_pocz adr\_końc
- Z adres



# ATARI SUPER TURBO

**ATARI SUPER TURBO jest specjalnym systemem bardzo szybkiej transmisji danych z magnetofonu do komputera, umożliwiającym bezbłędną komunikację i bardzo szybkie wczytywanie (praktycznie poniżej 3 minut dla najdłuższych programów, których czas wczytywania w normalnym systemie sięgał 20—30 minut).**

**System AST został zainstalowany w naszym redakcyjnym magnetofonie XC12 przez jedynego prawnego dystrybutora systemu: firmę AS — ATARI STUDIO (Warszawa tel. 125-123, aktualny adres ul. Józwiaka 4, poprzednio al. Bieruta 5/34, adres korespondencyjny: Warszawa 33, skr. poczt. 80). Chcielibyśmy w poniższym artykule przekazać Państwu nasze uwagi na temat systemu.**

## KORZENIE

Pod koniec 1986 roku początkujący student Sławek postawił sobie za cel „uszlachetnienie” denerwującego systemu pobierania informacji z magnetofonu w 8-bitowych komputerach Atari. Ponieważ Commodore i Spectrum miały swoje systemy TURBO wielokrotnie zmniejszające czas wczytywania programów więc doszedł do wniosku, że Atari też na to stać. Po półrocznych próbach i badaniach, zbudował prototypowy egzemplarz magnetofonu Atari 1010 i napisał pierwszy program do obsługi systemu AST. Po kilku następnych miesiącach powstały pierwsze egzemplarze XC12 pracujące w tym samym systemie, choć z trochę mniejszą prędkością. Zastosowanie przędkości transmisji użytej w 1010 było dla tych magnetofonów bardzo ryzykowne. Cały system został dopracowany i w swojej pierwszej wersji zaczął być rozprowadzany pod koniec 1987 roku. Ponieważ autor nie opierał się na żadnym istniejącym rozwiązaniu, bo takich po prostu nie było, zgłosił swój wynalazek w Urzędzie Patentowym (nadal posiada tylko numer zgłoszenia patentowego, choć otrzymanie numeru patentu wydaje się już być bardzo bliskie). Niestety znalazło się sporo osób nieuczciwych, które wykorzystywały luki w naszych przepisach i zaczęły podrabiać system zmieniając wianietki programów, nazwy systemu i w kilku miejscach pojawiły się nowe „oryginalne” Turbo, będące po prostu „podróżkami i przeróbkami” systemu AST (bez względu na nazwę HARD, SUPER, SPE-

CIAL, BLIZZARD itp...). Niestety firmy bądź osoby „grzebiące” w programach często je uszkodziły i systemy pracowały nie najlepiej, brakowało im dodatkowego, nowego oprogramowania. Często „twórcy” wybierali się oczywiście do źródeł czyli firmy **AS-ATARI STUDIO** gdzie — za w sumie niewielkie pieniądze — mogli nabyć wszystko, co najnowsze do systemu **AST**.

W chwili obecnej jest wielu użytkowników systemu **AST** i właściwie wszyscy są z systemu zadowoleni, czy tak jest z „pirackimi” podróbkami systemu, to trudno powiedzieć. Firma **AS** dba o swoich klientów i stara się stopniowo i oczywiście w miarę możliwości spełnić ich wszystkie prośby dotyczące systemu.

Autorem **ATARI SUPER TURBO** jest Sławomir Nienaltowski, zaś wszelkie prawa do systemu **AST** posiada firma **AS-ATARI STUDIO** (Warszawa 33 skr. poczt. 80).

## ZESTAW PODSTAWOWY

Podstawowa wersja systemu to:

- modyfikacja firmowego magnetofonu Atari (1010, XC11, XC12, a także TurboData, PhoneMark, itp.);
- kasetę z programami narzędziowymi do systemu **AST**;
- kasetę z 16 gramami zapisanymi w standardzie **AST**;
- instrukcja obsługi.

Magnetofon po modyfikacji może nadal bez zmian pracować w standardowym systemie, natomiast po wczytaniu programu obsługującego automatycznie przełą-

czy się na **AST**. Kasetę z programami narzędziowymi zawiera program ładujący, system operacyjny, dwa programy kopiujące do standardowych zbiorów z systemu normalnego na **AST** oraz tester ustawienia głowicy. Loader **AST** służy do uruchamiania programów zapisanych w tym systemie. Systemem Operacyjnym można programy dodatkowo przegrywać, redagować tytuł, robić przerwy na kasetach, wczytywać inne zbiory itp. Programy kopiujące służą do translacji z postaci normalnej na **AST**.

Programy nagrane w tym systemie posiadają tytuły, nie wymagają ustawiania taśmy na początek programu, procent błędów przy wczytywaniu jest znikomy, a najdłuższe z nich wczytują się w granicach 3 minut. Obsługa **AST** jest bardzo prosta.

## OPROGRAMOWANIE DODATKOWE

Jak już wspominaliśmy firma **AS** nie zapomina o swoich klientach i stara się spełnić ich życzenia. Dlatego oprócz podstawowego pakietu programów stara się udostępnić nowe programy ułatwiające pracę z **AST**. Ponieważ wszystkie programy sprzedawane były jako niezabezpieczone więc ułatwiło to pracę wszelkim „piratom” systemu, choć na pewno w zapamiętaliśmy przerabianiu choćby napisów nie ustrzegli się oni błędów. Powstały oczywiście translatory między systemem dla 1010 i XC12, który z uwagi na zamontowanie pewnej ilości **AST** do magnetofonów 1010, pozostał w swojej starej bardzo szybkiej formie (najdłuższe programy ok. 2 minuty). Umożliwiają one 100 proc. wymienności oprogramowania pomiędzy systemami. Kolejne programy powstały od momentu rozpowszechniania systemu to:

**AST BASIC** — program instaluje w standardowym edytorze Basicu dodatkowe urządzenie o nazwie „T:”, oznaczające magnetofon pracujący w systemie **AST**, co umożliwi używanie dodatkowych instrukcji dotyczących urządzenia zewnętrznego **AST**. Pozwala to na przegranie 100 proc. programów w Basicu na ten system, po oczywiście uprzednim ich zabezpieczeniu.

**AST INIT** — program kopiujący z systemu normalnego na **AST**, do tych programów, które uruchamiały się za standardowego kopiera dołączonego do systemu.

**AST->NORMAL** — program pozwalający na skopiowanie większości programów zapisanych w **AST** na normalną transmisję.

**AST SPECIAL PACKAGE** — zestaw programów do kopiowania na system **AST** programów nie-

standardowych, wieloblokowych, itp. W skład zestawu wchodzi dwa loadery do takich programów, monitor pamięci z wbudowaną opcją Trace (pojemność ok. 50 KB), trzy programy translujące na system **AST**, Turbo Basic XL zapisany w standardzie **AST** oraz bardzo obszerna instrukcja z dokładnymi informacjami na temat systemu, obszarów pamięci **AST**, sposobu przerabiania programów, adaptacji itp.

**SPECTRUS PACKAGE** — pakiet programów do „wyciągania” grafiki z programów na komputer ZX-Spectrum, adaptowania jej do standardu Atari i możliwości dołączania do programów Atari.

**AST CARTRIDGE** — najmłodsze dziecko systemu. Niby zwykły cartridge, jednak różny od znanych Państwu, bardzo szybki, nie wymagający wyjmowania z komputera (odłączalny w dowolnym momencie). Duża pojemność pozwoliła na umieszczenie na nim wszystkich programów niezbędnych do zaawansowanej pracy z systemem: wszystkie trzy loadery, system operacyjny, **AST BASIC**, programy kopiujące do programów standardowych i niestandardowych oraz opcja pozwalająca na odłączenie cartridge'a bez potrzeby wyjmowania go.

Oprócz wymienionych programów w firmie **AS** znajdują Państwo także przerobione już na wersję **AST** programy bardzo skomplikowane i niestandardowe, których nie można przerobić programami kopiującymi, np. „Goonies”, „Tomahawk”, „Spy vs Spy” I i II, „Beach Head” I i II, „Basil”, „Kennedy Approach”, „Zybex”, „Mercenary” oraz wiele innych.

Trudności w translacji polegają głównie na tym, że zmieniono system obsługi magnetofonu, więc niektóre programy mogą „nie zgadzać” się z tym nowym systemem, jednak wszystkie programy można przerobić na system **AST**.

## OCENA

System **AST** spisuje się bardzo dobrze i właściwie każdy użytkownik magnetofonu powinien być zadowolony z takiego systemu, tym bardziej, że bez względu na rozmiary szaleństwa cenowego na rynku komputerowym cena systemu jest kilkakrotnie niższa od ceny stacji dyskiety. Jeżeli zamierzają Państwo używać komputera głównie jako narzędzia do zabawy, czy zabawy połączonej z edukacją to znacznie rozsądniejsze jest zainstalowanie **AST** niż kupno stacji dyskiety, przy zastosowaniach bardziej profesjonalnych należałoby sprawę przemyśleć. Jeżeli natomiast chcemy przy użyciu „małego” Atari poprowadzić swoją firmę, to stacja dyskie-



tek okaże się nieodzowna — najwspanialszy i najszybszy nawet magnetofon nie jest w stanie jej zastąpić. Jeżeli chodzi o zalety systemu **AST** jest ich bardzo wiele — do najważniejszych powinniśmy zaliczyć prostą, szybką i bezawaryjną obsługę i wczytywanie programów. Jedną z niewielu wad systemu jest to, że nie wszystkie programy można przenieść na **AST** przy użyciu programów kopiujących, choć używając **AST SPECIAL PACKAGE** i dokładając do tego sporą porcję niezbędnej wiedzy można każdy program dostosować do **AST**. Jeżeli chodzi o nośnik, to można stosować takie same kasety jak przy normalnej transmisji, pamiętając oczywiście, aby nie miały zbyt dużych oporów mechanicznych.

Liczba reklamacji w firmie **AS** świadczy, że tak ganięne powszechnie magnetofony XC12 wcale nie są złe.

## WNIOSKI

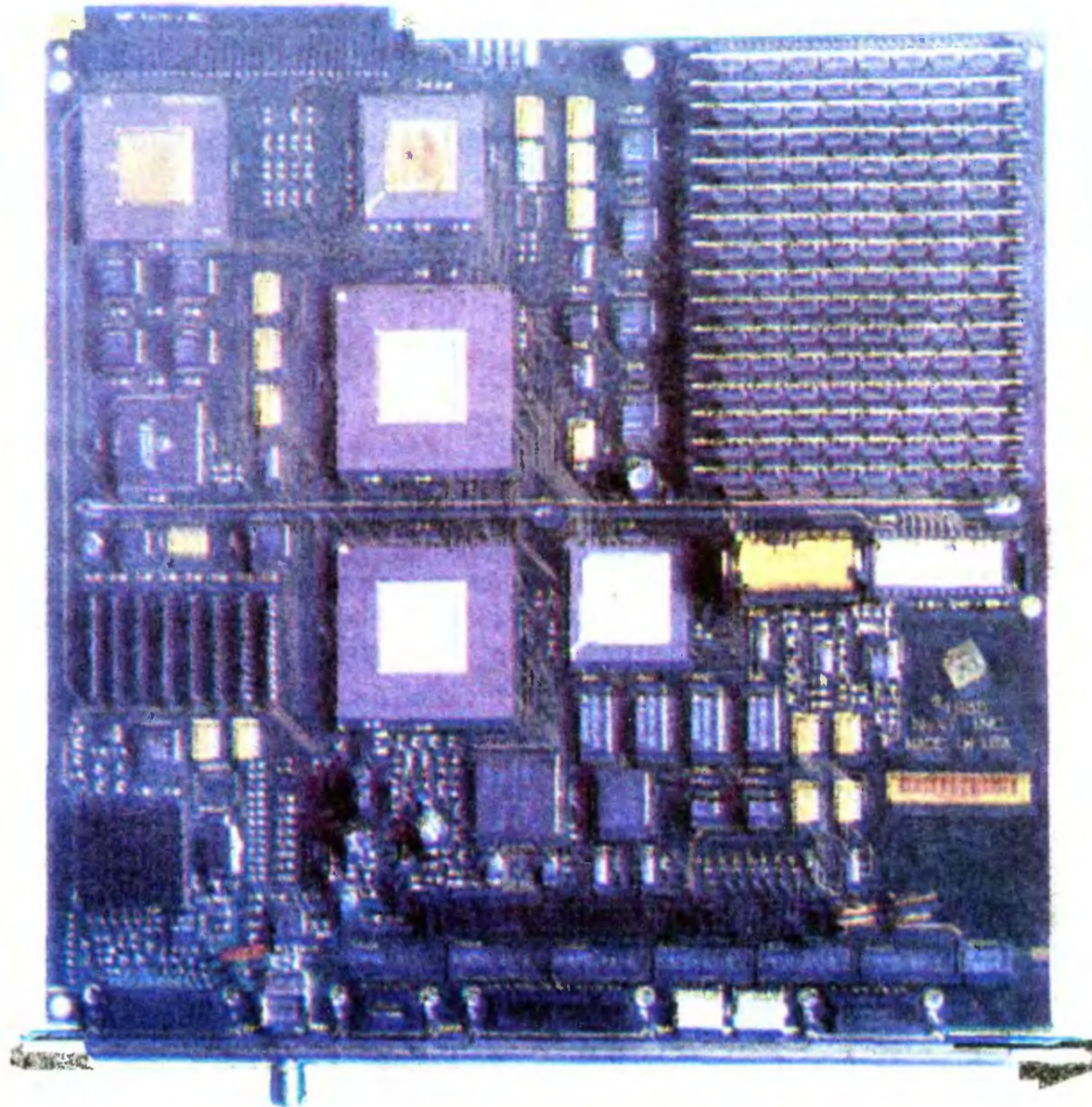
Pomysł i wynalazek tego urządzenia wydaje się iskierką nadziei na naszym „pirackim”, krajowym rynku komputerowym. Okazuje się, że potrafimy czasami coś wymyślić, a nie jedynie ściągać z za miedzy. Szkoda tylko, że nasi krajowi konkurenci nie potrafią tego docenić i wszelkimi dostępnymi metodami próbują omamić i naciągnąć klientów, a zarazem nabić dystrybutora i autora patentu w przysłowiową „butelkę”. Dlatego należy się liczyć z tym, że wszelkie nowości do systemu będą zabezpieczone i chronione bardziej niż dotychczas, a w momencie uzyskania numeru patentu (nie wykluczone, że kiedy Państwo czytają ten artykuł jest on już znany) firma w sposób prawny i bardziej konsekwentny zacznie dochodzić swoich praw.

Podsumowując, wydaje się nam, że system **AST** jest bardzo dobrą alternatywą dla wszystkich posiadaczy magnetofonów, pracuje bardzo poprawnie, a firma **AS** dba o to, aby się stale rozwijał. Niestety, jak dotychczas był on instalowany wyłącznie w magnetofonach firmowych, niewykluczone jednak, że zostanie wprowadzony do sprzedaży także interfejs do zwykłego magnetofonu (pojawił się już taki w „pirackim” wykonaniu, jednak firma **ATARI STUDIO** nie bierze odpowiedzialności za ich prawidłową pracę). Magnetofon z wbudowanym systemem **AST** (z uwagi na powyższe) pracuje we wszystkich dostępnych w naszym kraju systemach **TURBO**, także w „przywiezionych” z Czechosłowacji **TURBO** serii 2000 i 5000.

Jeżeli zdecydowałoby się Państwo na instalację systemu to mogą Państwo skontaktować się z firmą **AS-ATARI STUDIO** osobiście, telefonicznie lub listownie, dla osób z Warszawy i okolic usługa wykonywana jest w ciągu jednego lub dwóch dni, dla osób z odległych miejscowości po uprzednim umówieniu się, w miarę możliwości w ciągu jednego dnia.

Przypominamy! **ATARI STUDIO „AS”** zmieniło swoją siedzibę z al. Bieruta 5/34 na ul. Józwiaka 4 (przystanek autobusowy bliżej Trasy Łazienkowskiej), tel. 125-123, adres korespondencyjny: Warszawa 33 skr. poczt. 80.

*Tomasz Mazur*



# 1750 RAM EXPANSION MODULE

**Dzięki uprzejmości jednego ze stałych Czytelników BAJTKA otrzymałem do testowania kartkę rozszerzającą pamięć o 512 KB o nazwie „1750 RAM EXPANSION MODULE” przeznaczoną dla Commodore 128.**

Moduł 1750 pozwala zwiększyć pamięć Commodore 128 do całkowitej pojemności 640 KB. Jest on przyłączany do portu **EXPANSION** komputera. Odpowiednie wyprofilowanie dolnej części obudowy umożliwia przyłączenie modułu zarówno do C-128 jak i C-128D; w tym drugim wypadku moduł opiera się na prowadnicy przewodu uniemożliwiającej przy okazji zbyt bliskie dosunięcie komputera do ściany co mogłoby spowodować wykrzywienie i ewentualne uszkodzenie wtyków. Wygląd zewnętrzny modułu oraz płytę główną przedstawiono na ilustracji.

Wraz z samą kartą użytkownik otrzymuje kilkunastostronicową instrukcję obsługi oraz dwie dyskietki. Na jednej z nich znajduje się program testujący kartę oraz programy demonstracyjne w **BASIC** druga dyskietka zawiera odpowiednio zmodyfikowany system operacyjny **CP/M** umożliwiający wykorzystanie karty pod kontrolą tego systemu.

Moje odczucia na temat przydatności tego urządzenia i jego ceny są dość mieszane. Tak w **BASIC** jak i w **CP/M** moduł ten można wykorzystać przede wszystkim jako szybki **RAM-DISK**. **BASIC C-128** jest wyposażony w trzy instrukcje umożliwiające szybkie (1 MB na sekundę) przenoszenie danych do/z

pamięci dodatkowej — jako pamięć masową w której zapisywanych jest kilka lub kilkadziesiąt obrazów graficznych; dzięki ich szybkiemu przenoszeniu użytkownik otrzymuje w efekcie krótki film animowany z dość sugestywną grafiką.

Jak wynika z instrukcji obsługi, transfer obrazów czy danych do modułu może być przeprowadzany wyłącznie z banku 0 co w znacznym sposób ogranicza moim zdaniem uniwersalność karty. W lepszej sytuacji są programujący w języku maszynowym choć większe możliwości wykorzystania karty zawdzięczają raczej samemu językowi programowania. Dużym plusem instrukcji obsługi jest stosunkowo dokładnie omówiona mapa pamięci procesora 8726 sterującego pracą modułu.

Po przetestowaniu karty za pomocą odpowiedniego programu zawartego na dyskietce firmowej, użytkownik może wczytać do dodatkowej pamięci **RAM** swoje własne dane. W zależności od ilości tych danych czas transferu wynosi zwykle w granicach 5—10 minut (w czas ten należy wliczyć również czas tworzenia np. rysunku graficznego). Późniejsze wczytywanie danych do pamięci operacyjnej komputera jest realizowane za pośrednictwem linii **DMA**. Taki transfer jest możliwy w **BASIC** dzięki trzem wyżej wspomnianym instrukcjom. Po wciśnięciu przycisku **RESET** skasowaniu ulegają dane znajdujące się jedynie w pamięci operacyjnej komputera — programy/dane zawarte w pamięci dodatkowej pozostają nieknięte co ma swoje i złe i dobre strony.

Nieco inaczej wygląda praca pod kontrolą systemu **CP/M**. Po wczyta-

niu systemu ze sprzedawanej wraz z kartą dyskietki systemowej **CP/M** należy skopiować potrzebne nam zbiory i/lub programy za pomocą instrukcji **PIP**. W **CPM/** przyznano tej karcie literę **M** (**Memory drive**). Kopiowanie przeprowadza się w następujący sposób:

\*M:=A.\*

co pozwala przenieść do dodatkowej **RAM** zawartość całej dyskietki (mogą to być także określone i wybrane przez użytkownika zbiory). Następnie wykonujemy:

A>M:

M>

i w ten sposób pracujemy teraz z kartą zamiast dyskietki.

Najbardziej zauważalną różnicą jest czas dostępu do poszczególnych programów — w porównaniu z dyskietką są one wczytywane rzeczywiście dość szybko. Tym niemniej szybkość ta nie jest aż tak zachwycająca jak jest to często opisywane w reklamach — winą za to należy obarczyć przede wszystkim dość małą częstotliwość zegara **Z80** (ok. 2MHz) oraz niezbyt fortunne rozwiązanie współpracy procesora z kontrolerem wizji. W efekcie szybkość ta jest nieco mniejsza aniżeli spodziewana (zwłaszcza po cenie karty).

Podczas pracy pod kontrolą **CP/M** dysk **M** ma pojemność użytkową 508 KB. Niestety pełne wykorzystanie tej pamięci nie jest takie proste. Przypuśćmy, że masz duży zbiór danych utworzony za pomocą programu **dBASE II**. Zanim jednak rozpoczniesz jego uaktualnianie, musisz go przenieść do pamięci dodatkowej. Po dokonaniu transferu uaktualnianie odbywa się w karcie. Gdy praca zostaje zakończona poprawiony zbiór trzeba przenieść z powrotem na dyskietkę (za pomocą **PIP COM**), co może być już znacznie utrudnione jeżeli zbiór ma po uzupełnieniu większą pojemność aniżeli sama dyskietka...

Commodore dostarcza użytkownikowi dyskietkę z systemem **CP/M** datowanym 8 DEC 1985 ponieważ wersje wcześniejsze mogą nie chcieć współpracować z dyskiem o znaczniku **M** (tak dzieje się m.in. z wersją systemu datowaną 1 AUG 85). Uaktualnienie wcześniejszych wersji (o ile użytkownik takie posiada) leży już w gestii samego użytkownika przy czym na testowanej przeze mnie dyskietce znajdowała się tylko część zbiorów systemowych. Wynika z tego, że wymianie należy podać tylko te zbiory których nazwy powtarzają się w obu wersjach. Dodatkowo na dyskietce znajduje się program **SCREEN40.COM** wyłączający ekran 40-znakowy co nieznacznie przyspiesza pracę systemu. Podobnie jak w **BASIC**, po wciśnięciu **RESET** lub **CONTROL** i **ENTER** kasowana jest tylko zawartość pamięci operacyjnej komputera.

Reasumując, moduł 1750 jest na pewno modułem przydatnym lecz przede wszystkim dla bardziej zaawansowanych użytkowników znających zasady programowania w języku maszynowym lub korzystających na co dzień z systemu **CP/M**. Jeżeli weźmiemy pod uwagę, że za podobną cenę 150-200 \$ można nabyć drugą stację dysków (np. 1581 o pojemności ponad 700 KB) to karta ta traci na swej atrakcyjności. Z mojego punktu widzenia zdecydowałbym się chyba prędzej na ten drugi wydatek.

*Klaudiusz Dybowski*



# MONITORY ML CZ. V

**Rejestr słowa stanu (status register), często określany skrótem .P jest jednym z najważniejszych rejestrów mikroprocesora. Dostarcza on procesorowi wielu informacji dotyczących np. wyników ostatnio wykonanych operacji, pozwala na zmianę trybu pracy, wyłączenie i włączanie przerw IRQ itp. Dzisiejsza część będzie zatem poświęcona w całości temu rejestrowi.**

Rejestr słowa stanu jest rejestrem ośmiobitowym, w którym jednak wykorzystano tylko 7 bitów (bez bitu 5 licząc od bitu 0 numerze 0). W wielu monitorach jego zawartość jest podawana w postaci konkretnej liczby zapisanej szesnastkowo pod skrótem SR; bywa i tak, że rejestr ten jest obrazowany bit po bicie pod nagłówkiem NV-BDIZC.

Te tajemnicze literki oznaczają poszczególne znaczniki (flags) rejestru słowa stanu. Po nabraniu wprawy będziesz w stanie po ich konfiguracji zorientować się jakiego rodzaju wydarzenia zaszyły podczas wykonywania programu; wykonywanie instrukcji skoków warunkowych (odpowiedników IF...THEN w BASIC) byłoby bez tego rejestru niemożliwe.

Znacznik jako taki nie bierze zwykle udziału w operacji (są wyjątki!) lecz przyjmuje stan logiczny zależny od jej wyniku. Innymi słowy dany znacznik jest modyfikowany odpowiednio poprzez wykonanie nawet pojedynczego rozkazu. Ponieważ część znaczników (np. C) jest objętych niektórymi poleceniami procesora (np. SBC, ROL itp.) ich stan może mieć pewien wpływ na wynik.

Aby dobrze zrozumieć informacje zawarte w rejestrze słowa stanu musimy zacząć od podstaw logiki. Podstawową jednostką informacji jest bit (Binary digit) czyli twór, który może przyjmować tylko dwa stany logiczne: 0 i 1. W praktyce chodzi to o pewne poziomy napięcia np. na nóżkach jakiegoś układu scalonego; gdy dane napięcie występuje określa się to za pomocą cyferki „1”, natomiast gdy napięcie takie nie występuje — oznaczamy to cyferką „0”.

Commodore 64, 128 czy C-16 to komputery ośmiobitowe co oznacza, że ich procesor może jednocześnie przetwarzać informacje złożone z 8 bitów. Grupa ośmiu bitów stanowiących jakąkolwiek informację nazywa się zwykle SŁOWEM — wszystkie wspomniane powyżej modele Commodore pracują więc w oparciu o słowo ośmiobitowe.

Jeden bit może reprezentować sobą tylko dwa stany — 0 i 1. Dwa bity mają już większe możliwości — reprezentują bowiem cztery stany:

BIT PIERWSZY	BIT DRUGI
0	0
1	0
0	1
1	1

Liczba możliwych kombinacji oczywiście zwiększa się w miarę dodawania następnych bitów. Osiem bitów w bajcie daje nam łącznie 256 kombinacji (od 00000000 do 11111111). Liczba kombinacji podwaja się przy dodaniu następnego bitu.

- Bit o numerze 7—128 Bit o numerze 3—0
- Bit o numerze 6— 64 Bit o numerze 2—4
- Bit o numerze 5— 32 Bit o numerze 1—2
- Bit o numerze 4— 16 Bit o numerze 0—1\*

\* należy tu również uwzględnić stan logiczny 0 bitu.

Podstawy tego liczenia zostały wytłumaczone

przez Dominika Falkowskiego w serii artykułów poświęconej programowaniu w języku maszynowym. Do naszych potrzeb niezbędna jest znajomość, iż liczba zapisana w dowolnej komórce pamięci komputera czy w dowolnym rejestrze mikroprocesora określa nie konkretną liczbę dziesiętną lecz konfigurację bitów w danym bajcie.

Przypuśćmy, że z rejestru słowa stanu odczytałeś liczbę 192. Co to oznacza? Że w rejestrze słowa stanu zapisana jest następująca konfiguracja bitów:

Numer bitu:	7	6	5	4	3	2	1	0
Znacznik:	N	V	—	B	D	I	Z	C
Stan logiczny bitów:	1	1	0	0	0	0	0	0

gdyż  $2^{17} + 2^{16} = 128 + 64 = 192$ . Ponieważ zero podniesione do jakiegokolwiek potęgi da nam zawsze 0 stan bitów o numerach od 5 do 0 nie jest brany pod uwagę podczas obliczeń. Liczą się tylko bity o stanie logicznym 1...

Jednakże nawet to wyliczenie nie jest jeszcze odpowiedzią na pytanie „co oznacza liczba 192 w rejestrze słowa stanu”. Prawidłową odpowiedzią będzie stwierdzenie, że znaczniki N i V są ustawione (stan logiczny 1), natomiast pozostałe są wyzerowane (stan logiczny 0). Tego rodzaju przekaz natomiast oznacza bardzo konkretne rzeczy, a ściślej — rezultaty wykonywanych ostatnio operacji. Zanim jednak do tego przejdziemy zapoznajmy się ze znaczeniem poszczególnych znaczników rejestru słowa stanu.

N — znacznik ujemności (Negative flag)

Znacznik ten przyjmuje stan logiczny 1 wyłącznie wtedy, gdy wykonano lub wykonuje się operacje w wyniku której bit 7 (np. wyniku) przyjmuje stan logiczny 1. Ponadto ma on pewne znaczenie w arytmetyce liczb zamiennoprzecinkowych.

Rozkazy zmieniające stan znacznika: N=1 — brak, N=0 — brak.

UWAGA. Przed wykonaniem jakiegokolwiek z podanych tu przykładów powinieneś najpierw wyzerować całkowicie rejestr słowa stanu za pomocą następującej instrukcji:

```
; 0000 00 00 00 00 00
lub
;      00 00 00 00 00
```

Jeżeli Twój monitor wyświetla zawartość rejestru słowa stanu jako liczbę, to badania konfiguracji bitów w tym rejestrze musisz przeprowadzać poprzez zamianę tej liczby szesnastkowej na binarną. Sprawdź w instrukcji obsługi monitora czy i jakim poleceniem można taką zamianę przeprowadzić.

Po wyzerowaniu rejestru .P wpisz i wykonaj:

```
A 2710 LDA    $80
A 2712 BRK
```

uruchom program za pomocą G 2710 i po ukazaniu się na ekranie czołówki monitora sprawdź zawartość rejestru słowa stanu. Liczba ta będzie na pewno większa niż \$80 (128 dziesiętnie) np. ze względu na wykonaną instrukcję BRK (patrz znacznik B). V — znacznik przepiętlenia (o Verflow flag)

Znacznik ten jest wykorzystywany w arytmetyce liczb zamiennoprzecinkowych i przyjmuje on stan logiczny 1, gdy wynik dowolnej operacji matematycznej przekracza wartości +127 i -128. W przeciwnym wypadku bit ten przyjmuje stan logiczny 0.

Przykład (wykonaj go po wyzerowaniu rejestru .P):

```
A 2710 LDA    $7F
A 2712 CLC
A 2713 ADC    $7F
A 2715 STA    $2720
A 2718 BRK
```

W wyniku działania tego programu (dodawanie 127 do 127) znacznik V przyjmuje stan logiczny 1. Wynik operacji jest zapisywany w komórce \$2720.

Rozkazy zmieniające stan znacznika: V=1 — brak, V=0 — CLY.

Bit 5 rejestru słowa stanu

Projektanci rodziny mikroprocesorów 65XX nie

przypisali temu znacznikowi żadnego znaczenia (oficjalnie).

B — znacznik przerwania wykonywania programu (Break flag)

Znacznik ten przyjmuje stan logiczny i zawsze wtedy, gdy mikroprocesor wykona instrukcję BRK. W chwili wznowienia lub rozpoczęcia wykonywania nowego programu jest on samoistnie zerowany przez mikroprocesor.

Przykład:

```
A 2710 BRK
```

Po ukazaniu się czołówki monitora okaże się, że znacznik B ma stan logiczny 1.

Rozkazy zmieniające stan znacznika: B=1 — BRK, B=0 — brak.

D — znacznik włączenia dziesiętnego trybu pracy mikroprocesora (Decimal mode on flag).

Znacznik ten przyjmuje stan logiczny 1 wyłącznie po wykonaniu polecenia SED i jest zerowany poleceniem CLD. Pózwala on na włączenie dziesiętnego trybu pracy mikroprocesora (w rzeczywistości BCD — Binary Coded Decimal) czyli takiego w którym liczby są zapisywane jako poszczególne cyfry w formie binarnej — np. 123 jest zapisywana w trzech bajtach jako 00000001 00000010 00000011. Rozkazy zmieniające stan znacznika: D=1 — SED, D=0 — CLD.

I — znacznik wyłączenia przerw IRQ (Interrupt disable flag)

Podobnie jak w wypadku znacznika D, znacznik I przyjmuje stan logiczny 1 po wykonaniu instrukcji SEI, której zadaniem jest wyłączenie przerw IRQ. Może to być potrzebne np. podczas włączania własnego programu do procedury obsługi przerw, zmian niektórych wektorów itp. I jest zerowany poprzez wykonanie polecenia CLI.

Rozkazy zmieniające stan znacznika: I=1 — SEI, I=0 — CLI.

Z — znacznik zera (Zero flag)

Znacznik ten przyjmuje stan logiczny 1 gdy wynik operacji wynosi zero. Wyzerowanie tego znacznika następuje gdy wynik jest różny od zera.

Przykład:

```
A 2710 SEC
A 2711 LDA    $02
A 2713 SBC    $02
A 2715 STA    $2720
A 2718 BRK
```

Program ten to odejmowanie 2—2. Wynik zostanie wpisany do komórki \$2720.

Rozkazy zmieniające stan znacznika: Z=1 — brak, Z=0 — brak.

C — znacznik przeniesienia (Carry flag)

Znacznik ten przyjmuje stan logiczny 1 w wielu sytuacjach, np. po wykonaniu rozkazu SEC, gdy w wyniku dodawania otrzymamy wynik większy niż 255, podczas przesuwania bitów za pomocą rozkazów ROR, ASL, LSR itp. Po wyzerowaniu rejestru słowa stanu wpisz następujący przykład:

```
A 2710 CLC
A 2711 LDA    $80
A 2713 ADC    $80
A 2715 STA    $2720
A 2718 BRK
```

Po wykonaniu programu przekonasz się, że znacznik C przyjął stan logiczny 1 gdyż  $128 + 128 = 256$ . Wynik jest zapisany w komórce \$2720 choć być może nie w takiej postaci o jakiej myślisz.

Rozkazy zmieniające stan znacznika: C=1 — SEC, C=0 — CLC.

W następnej części „Monitorów” zajmiemy się bliżej analizą wykonanych w programie operacji na podstawie otrzymanego wyniku oraz zawartości rejestru słowa stanu, omówimy także w zarysie ogólnym zależność instrukcji warunkowych od tego rejestru.

(cdn)

Klaudiusz Dybowski



## PORADY SPOD LADY

Po dwóch latach używania mojego C-64 komputer reaguje dość dziwnie na wciśnięcie klawisza RETURN. Czasami zdarza się, że nie reaguje wogóle, czasami natomiast jedno wciśnięcie powoduje przesunięcie kursora nawet o trzy, cztery linie w dół (...)

Objaw ten jest bardzo charakterystyczny dla silnie zanieczyszczonej klawiatury. Klawisze są montowane w plastikowych prowadnicach, które w miarę upływu czasu wycierają się na skutek tarcia powstającego przy wciskaniu klawisza. Wskutek tego tworzy się drobnieutęki pyłek opadający na zestyki znajdujące się dokładnie pod spodem. Pył izoluje zestyk co następnie objawia się właśnie w opisany przez Pana sposób. Radziłbym zatem oczyścić dokładnie klawiaturę.

W literaturze fachowej poświęconej Commodore ciągle wspomina się o niekorzystnym porcie SZEREGOWYM przedstawiając jako wzór doskonałości port RÓWNOLEGŁY. Czy mogłbym prosić o wyjaśnienie o co tu właściwie chodzi?

Chodzi tu o sposób transmisji danych do/z urządzeń peryferyjnych. Transmisja SZEREGOWA polega na zamianie pojedynczego bajtu na 8 bitów i wysłaniu ich kolejno w danej jednostce czasu np. do drukarki. Transmisja RÓWNOLEGŁA natomiast pozwala na jednoczesne wysłanie 8 bitów w tej samej jednostce czasu co znacznie zwiększa prędkość transmisji.

Oba rozwiązania mają swoje wady i zalety. Wadą transmisji szeregowej jest jej powolność, zaletą natomiast stosunkowo niewielka ilość linii potrzebnych do tego celu; transmisja równoległa jest szybsza lecz wymaga stosowania znacznie większej ilości linii transmisyjnych.

Od niedawna zajmuję się programowaniem w języku maszynowym C-128. Ostatnio w mapie pamięci napotkałem określenie KERNAL JUMP TABLE (tabela skoków KERNAL-u). Do czego służy ta tabela i jak można ją wykorzystać?

Systemy operacyjne C-128 i C-64 zawierają szereg standardowych procedur, przeznaczonych m.in. do wykorzystania przez programujących w języku wewnętrznym. Jednakże ze względu na odmienne założenia projektowe dla obu tych komputerów dana procedura może być dostępna w C-64 od adresu E000 a w C-128 od adresu E8CA. Gdyby teraz uruchomił Pan program napisany dla C-64 na C-128 to prawdopodobnie zablokowałby się on.

Ponieważ zestaw rozkazów mikroprocesora w obu komputerach jest identyczny projektanci wpadli na pomysł aby w określony i STAŁY we wszystkich komputerach obszar pamięci ROM wpisać adresy poszczególnych procedur standardowych (CLALL, RAMTAS, CHROUT itp.). Jeżeli teraz w programie układanym dla C-64 zamiast fizycznego adresu damy na to procedury CHROUT wpisze Pan adres \$FFD2 będący jednym z adresów tabeli skoków KERNAL, to może Pan dany program wykorzystać i na C-64 i na C-128. Po wykonaniu skoku do komórki \$FFD2 procesor znajdzie tam rozkaz JMP i aktualny adres procedury CHROUT zależy od typu komputera. Dzięki temu, jeśli korzysta Pan wyłącznie z procedur standardowych systemu operacyjnego (bez odwołań do BASIC ROM) Pana program będzie działał na wszystkich komputerach Commodore- C-64, C-128 czy C-16. Szerzej omówimy tabelę skoków w cyklu poświęconym programowaniu w języku maszynowym.

# JĘZYK MASZYNOWY (CZ.3)

## Miesiąc temu naszą lekcję zakończyliśmy na pracy domowej. Zgodnie z daną obietnicą dziś odpowiedź i dalszy ciąg wykładu.

Zarówno Listing 1 jak i Listing 2 przedstawiają rozwiązanie naszej pracy domowej. Różnica między nimi polega na tym, że program drugi jest ogólniejszy, ale o tym za chwilę. Listing 1 przedstawia rozwiązanie problemu w sposób jaki zademonstrowałem ostatnio tzn. pobierane są z pamięci i wczytywane do rejestrów wartości spod bezpośrednio określonych adresów (kolejno \$2800, \$2801 ... \$2804) i następnie zawartości rejestrów przesyłane są do komórek pamięci zwiększonych o jeden w stosunku do tych, z których zostały pobrane. Na koniec zawartość komórki o adresie \$2804 zostaje przesyłana do komórki o adresie \$2800. Jak widać takie rozwiązanie ma jedną kolosalną wadę. Im więcej komórek pamięci chcemy powymieniać zawartościami pomiędzy sobą, tym program staje się dłuższy, gdyż musimy za każdym razem określać konkretne adresy komórek, a w związku z tym prawdopodobieństwo popełnienia błędu wzrasta bardzo szybko. Drugi program pozbawiony jest tej wady. Zanim jednak omówię sam program wspomnę o kilku nowych elementach języka maszynowego jakie się w nim pojawiły. Po pierwsze znajduje się w nim nowy typ instrukcji wczytywania: np. **LDY \$2800,X**. Taka instrukcja oznacza, że zawartość rejestru .X zostanie dodana do adresu \$2800 i dopiero spod tak określonego adresu (\$2800+X) zostanie pobrana wartość i wczytana do rejestru .Y np. programik

```
LDX #000
LDA $2800, X
INX
LDA $2800, X
INX
LDA $2800, X
BRK
```

będzie działał następująco: do rejestru .X zostanie wczytana wartość zero; następnie 0 zostanie dodane do \$2800 (otrzymujemy adres \$0+\$2800=\$2800) i zawartość komórki o adresie \$2800 zostanie wczytana do rejestru .A; następnie zawartość rejestru .X zostanie zwiększona o 1 i będzie wynosić \$0+\$1=\$1; kolejna instrukcja spowoduje dodanie zawartości rejestru .X do \$2800 (otrzymujemy \$1+\$2800=\$2801) i spod tego adresu \$2801 zostanie wczytana zawartość komórki do rejestru .A; rejestr .X zwiększy swoją zawartość o 1 itd. Taki sposób postępowania ma jedną wielką zaletę: musimy jedynie wyróżnić pierwszy adres komórki, a kolejne adresy uzyskujemy przez dodanie odpowiedniej wartości do tego adresu. Instrukcje **TYA** zostały omówione podczas ostatniego wykładu, ale przypomnę, że chodzi o przesyłanie zawartości rejestrów pomiędzy sobą. W tym przypadku zawartość rejestru .Y zostanie wczytana do rejestru .A.

Instrukcje **INX, DEX** powodują odpowiednio zwiększenie i zmniejszenie zawartości rejestru .X o jeden. Istnieją analogiczne instrukcje dla rejestru .Y (**INY, DEY**). Instrukcje **INC** i **DEC** natomiast powodują zwiększenie lub zmniejszenie zawartości dowolnej komórki pamięci (po podaniu adresu komórki np. **INC \$2800**). Instrukcja **CPX#\$06** należy do instrukcji porównań.

W tym wypadku sprawdzamy czy zawartość rejestru .X jest równa liczbie #\$06 (znak # oznacza, że chodzi o wartość a nie o adres). W zależności od relacji pomiędzy zawartością .X i wartością #\$06 ustawiane są znaczniki w rejestrze słowa stanu (m.in. znacznik Z omówiony w cyklu wykładów poświęconych monitorom). Następna instrukcja **BNE \$2719** bazując na stanie logicznym znaczników steruje odpowiednio dalszym wykonywaniem programu. **BNE \$2717** sprawdza czy ustawiony jest znacznik Z. Jeżeli nie, to wykonywanie programu zostanie wznowione od adresu \$2717 w przeciwnym wypadku zostanie wykonana instrukcja po **BNE**. Ponieważ grupy instrukcji związanych z rozgałęzieniami, pętlami, warunkami będą jeszcze szeroko omawiane w cyklu o monitorach i w niniejszym kursie języka maszynowego poprzestaną na wyjaśnieniu działania odmian tych instrukcji występujących w programie 2. Jak zdążyliśmy zauważyć połączenie adresowania absolutnego indeksowanego (**LDY \$2800,X**) ze sprawdzaniem warunków (**CPX#\$06**) i pseudolicznikiem (**INX**) daje możliwość konstruowania pętli a co za tym idzie nasz program będzie zamieniał między sobą tyle komórek ile tylko zechcemy (do 255). W przypadku zmiany ilości komórek jakie chcemy pozamieniać wartościami jedyna zmiana w programie będzie dotyczyć linii 271F, w której to po instrukcji **CPX** umieszczamy ilość komórek+1. Tak więc niezależnie od ilości komórek program nie będzie się wydłużał w odróżnieniu od programu 1, który dla 255 komórek skonsumowałby sporo cennej pamięci, byłby nieznosnie długi i prawdopodobnie zawierałby gdzieś trudny do wychycenia błąd.

Listing 2 jak widać jest prosty do przetestowania. dokładne przeszedzenie tych programów pozostawiam Wam, my natomiast zajmiemy się ugruntowaniem świeżo nabytej wiedzy o pętlach.

Jak wiemy system operacyjny komputera (KERNAL) składa się z szeregu procedur realizujących różne mniej lub bardziej skomplikowane działania. Wynika z tego, że w wielu wypadkach nasze programy mogą korzystać z procedur systemu operacyjnego czy nawet ROM BASIC. Zasady korzystania z tych procedur zostaną omówione szczegółowo podczas jednego z wykładów niniejszego kursu. Jedną z takich procedur jest procedura o nazwie **CHROUT** (lub **BSOUT** w C-128) zaczynająca się od

## KLAN COMMODORE

adresu \$FFD2. Procedura ta powoduje wysłanie kopii rejestru .A w formie znaku w kodzie ASCII na ekran, drukarkę, magnetofon czy stację dysk. W naszym wypadku wysłamy znak na ekran monitora. Działanie tej procedury jest bardzo podobne do działania instrukcji PRINT i PRINT# z tą różnicą, że wysyłany jest tylko pojedynczy znak. Zawartość akumulatora .A to nic innego jak kod ASCII znaku jaki chcemy by pojawił się na ekranie. Wszystkie specjalne znaki: graficzne, zmieniające kolor honorowane są również. I tak aby wyświetlić literę X (o kodzie ASCII 58, dziesiętnie 88) musimy:

1. wczytać wartość 58 do akumulatora czyli do rejestru .A

2. wywołać procedurę CHROUT.

Analogicznie do języka BASIC, w którym występuje instrukcja GOSUB również w języku maszynowym istnieje taka instrukcja: **JSR** adres (Jump to SubRoutine). Każda z procedur w pamięci komputera zakończona jest instrukcją **RTS** (analogia do instrukcji RETURN w języku BASIC) więc nie musimy się obawiać, że komputer nie znajdzie drogi powrotnej do naszego programu.

Spróbujmy teraz wpisać i wykonać przedstawiony powyżej program.

```
$2710 LDA #$58
$2712 JSR $FFD2
$2715 RTS
```

Dlaczego RTS a nie BRK na końcu tego programu zapytacie zapewne? Może oczywiście być BRK, ale mam zamiar uruchomić ten program pod kontrolą BASIC-a: **FOR I=1 TO 10**  
**SYS 10000**  
**NEXT I**

i dlatego RTS spowoduje przekazanie kontroli do BASIC-a po wykonaniu naszego programu. W ten sposób poznaliśmy sposób na łączenie języka maszynowego z językiem BASIC. Taki sposób postępowania daje nam możliwość łączenia zalet BASIC-a z np. szybkością i uniwersalnością procedur maszynowych.

W domu spróbujcie wyświetlić na ekranie monitora parę różnych liter i znaków. Dobrze by było gdyby te literki układały się w jakiś miły tekst np. NAJLEPSZY W BAJTKU JEST KLAN COMMODORE. Jeżeli Wam się to uda to gratuluje. Podpowiem od razu, że warto sobie przypomnieć to co zostało powiedziane na początku tego wykładu o pętlach... chyba, że lubicie dużo pisać. Za miesiąc rozwiązanie i ciąg dalszy.

Dominik Falkowski

### LISTING 1

READY.						
MONITOR	PC	SR	AC	XR	YR	SP
:	B000	00	00	00	00	FB
02710	AD	00	28	LDA	\$2800	
02713	AE	01	28	LDX	\$2801	
02716	AC	02	28	LDY	\$2802	
02719	BD	01	28	STA	\$2801	
0271C	BE	02	28	STX	\$2802	
0271F	AD	03	28	LDA	\$2803	
02722	BC	03	28	STY	\$2803	
02725	AE	04	28	LDX	\$2804	
02728	BD	04	28	STA	\$2804	
0272B	BE	00	28	STX	\$2800	
0272E	00			BRK		

### LISTING 2

READY.						
MONITOR	PC	SR	AC	XR	YR	SP
:	B000	00	00	00	00	FB
02710	A2	00		LDX	#\$00	
02712	BC	00	28	LDY	\$2800, X	
02715	98			TYA		
02716	EB			INX		
02717	BC	00	28	LDY	\$2800, X	
0271A	9D	00	28	STA	\$2800, X	
0271D	98			TYA		
0271E	EB			INX		
0271F	E0	06		CPX	#\$06	
02721	D0	F4		BNE	\$2717	
02723	CA			DEX		
02724	BD	00	28	LDA	\$2800, X	
02727	BD	00	28	STA	\$2800	
0272A	00			BRK		



# KOPIOWANIE EKRANU NIE JEST TAKIE PROSTE

Chciałbym opowiedzieć Wam historię o pożyczonej drukarce i pilnej pracy do wykonania. Morał wyciągnijcie sami.

Pożyczyłem od znajomego drukarkę. Nie był to Epson ani Star, lecz zwykła stara Seiksha GP-500AS. Miałem zamiar przy jej pomocy przygotować różne, pilnie mi potrzebne graficzne materiały.

Podłączyłem wszystko zgodnie z instrukcją i zdrowym rozsądkiem, napisałem tak jak trzeba **OPEN#3; "t": FORMAT "t"; 9600**, i spróbowałem wydrukować jakiś program. Udało się. Pokrzepiony tym sukcesem, napisałem **COPY**. Nie udało się. Spróbowałem z innej beczki — napisałem **OPEN#4; "b": FORMAT "b"; 9600. COPY** jak nie dawało żadnego efektu, tak nie daje dalej. W tej sytuacji postanowiłem wziąć się do sprawy systematycznie, wyciągnąłem różne mądre książki i zacząłem szukać. Odpowiedź znalazłem w ROM-ie. **COPY** działa tylko wtedy, gdy korzysta się ze specjalnej drukarki, przystosowanej do Spectrum. Co ma w takiej sytuacji zrobić człowiek, któremu zależy na skopiowaniu ekranu — to samo, co zawsze się robi, kiedy trzeba — napisać samemu odpowiedni program. Tak też zrobiłem. (Przy wszystkich programach zakładam, że tak samo jak ja, otworzyliście strumienie trzeci i czwarty).

Zacząłem od programu kopiującego ekran w trybie tekstowym (wydruk 1). W końcu od czasu do czasu zależy nam na przestaniu na drukarkę jakichś wyników — wtedy wystarczy skopiowanie znaków AS-CII, znajdujących się na ekranie.

Niestety, czasem trzeba przemieścić na papier widoczną na ekranie grafikę. Wtedy pierwszy program jest bezużyteczny, bo funkcja **SCREEN\$** potrafi rozpoznać tylko znaki których wzorce są zapamiętane w ROM-ie, bez grze-

## PROGRAM 1

```
10 FOR i=0 TO 21: REM pętla dla wierszy
20 FOR j=0 TO 31: REM pętla dla kolumn
30 LPRINT SCREEN$(i, j);: REM j-ty znak i-tego wiersza
40 NEXT j
50 LPRINT: REM przejście do następnego wiersza
60 NEXT i
```

bania w zmiennych systemowych nie można przy jej pomocy rozpoznać nawet UDG. W takich wypadkach jedyne co może nam pomóc, to program kopiujący ekran w trybie graficznym (wydruk 2).

Cały problem polega na odpowiednim podzieleniu ekranu na poziome paski o grubości siedmiu pikseli i sprawdzaniu stanu pikseli (są czy ich nie ma) w odpowiedniej kolejności — od dołu do góry. To, co się dzieje w linii 150, najlepiej sobie rozpisać — jest tam przygotowywany bajt, którego poszczególne bity odpowiadają stanowi odpowiednich punktów na ekranie. Taki program wprowadzić działa bardzo skutecznie, ale ma jedną wadę. Jest przeraźliwie wolny. Skopiowanie ekranu trwa ponad 14 minut czyli więcej niż przygotowanie skomplikowanego rysunku, o czym niedawno pisałem. Należy go więc przyspieszyć czyli „ręcznie” skompilować. Uzyskamy wtedy to samo, ale w kodzie maszynowym (wydruk 3).

Teraz jest już szybko — to nie drukarka czeka na komputer ale odwrotnie — komputer czeka, aż drukarka skończy poprzedni wiersz, żeby jej przestać dalszy ciąg danych. (Program uruchamia się instrukcją **PRINT#4; CHR\$ USR adres**, i nie radzę robić tego inaczej, bo nic się nie skopiuje!).

Znając użytkowników Spectrum wiem, że niektórzy patrzą podejrzliwie na program w assemblerze. Mam dla nich niespodziankę — nie musicie z tego wydruku korzystać! Popatrzcie na następny — jest to program ładujący wersją, którą on ładuje jest wygodniejsza — bo relokowalna (wydruk 4)!

Podejrzewam, że naraziłem się tym wszystkim, którzy mają inne drukarki niż pożyczona przeze mnie. Radzę im dobrze przeanalizować wszystkie programy — i spróbować napisać własne. Powodzenia!

Marcin Borkowski

## PROGRAM 2

```
100 FOR y=175 TO 7 STEP -7: REM głowica ma siedem igieł
110 PRINT #4;CHR$ 8;: REM wywołanie trybu graficznego
120 FOR x=0 TO 255: REM od lewej do prawej
130 LET b=0: REM bity tej zmiennej sterują igłami
140 FOR i=6 TO 0 STEP -1: REM od dołu do góry
150 LET b=2*x+POINT(x, y-1): REM najlepiej to rozpisac
160 NEXT i
170 PRINT #4;CHR$ (128+b);: REM 7 bit musi być równy 1
180 NEXT x
190 PRINT #4;CHR$ 10: REM koniec tego wiersza
200 NEXT y
210 PRINT #4;CHR$ 8;: REM zostało jeszcze 256 punktów
220 FOR x=0 TO 255: REM o współrzędnej y=0
230 PRINT #4;CHR$ (128+POINT(x, 0));
240 NEXT x
250 PRINT #4;CHR$ 10
```

## PROGRAM 3

```
10 ORG 40000
20 LD IX, DANE
30 LD (IX+0), 25; (IX+0) - kuzyn zmiennej y
40 LP3 LD A, 8 ; LP3 - pętla dla y
50 RST #10 ; uaktywnia tryb graficzny
60 LD A, (IX+0)
70 SLA A ; weź (IX+0)
80 SLA A ; pomnóż przez 8
90 SLA A ; i odejmij (IX+0)
100 SUB (IX+0)
110 INC A ; w efekcie A=7*(IX+0)+1
120 LD E, A ; w E - ta sama wartość co y
130 LD (IX+1), 0 ; (IX+1) - to samo co x
140 LP2 LD D, 1 ; D - to samo co zmienna b
150 LD (IX+2), 7 ; (IX+2) - to samo co i
160 LP1 LD A, E ; LP2, LP1 - pętla dla x i i
170 SUB (IX+2) ; w efekcie A=y-i
180 LD B, A ; teraz z A do B
190 LD C, (IX+1) ; w C - współrzędna x
200 CALL PUNKT ; w A będzie 0 lub 1
210 ADD A, D
220 ADD A, D ; w efekcie A=2*D+A
230 LD D, A ; zapamiętaj w D
240 DEC (IX+2)
250 JR NZ, LP1 ; NEXT i
260 RST #10 ; wydrukuj odpowiednik b
270 INC (IX+1) ; NEXT x,
280 JR NZ, LP2 ; zero będzie po 255
290 LD A, 10
300 RST #10 ; koniec linii
310 DEC (IX+0)
320 JR NZ, LP3 ; NEXT y
330 LD A, 8
340 RST #10 ; znowu tryb graficzny
350 LD D, 0 ; w D - współrzędna x
360 LP4 LD C, D
370 LD B, 0 ; w B - współrzędna y
380 CALL PUNKT ; sprawdzamy najniższy rząd
390 ADD A, 128 ; punktów
400 RST #10
410 INC D ; NEXT x, zero będzie po 255
420 JR NZ, LP4
430 LD A, 10
440 RST #10
450 LD BC, 15 ; wróć do normalnego tekstu
460 RET
470 PUNKT CALL #22AA ; ten podprogram wprowadzi do
480 LD B, A ; HL adres bajtu zawierającego
490 INC B ; punkt o współrzędnych
500 LD A, (HL) ; x=C, y=B
510 PKT RLCA
520 DJNZ PKT
530 AND 1 ; teraz A=1 jeśli punkt ma
540 RET ; kolor atramentu, 0 jeśli tła
550 DANE DEFB 0, 0, 0 ; miejsce na zmienne x, y, i
```

## PROGRAM 4

```
10 INPUT "Podaj adres kodu? ";r
20 FOR i=r TO r+110 STEP 10
30 LET s=0
40 FOR j=0 TO 9
50 READ a: POKE i+j, a
60 LET s=s+a
70 NEXT j
80 READ a: IF s<>a THEN PRINT "Błąd w linii ";100+i-r
90 NEXT i
100 DATA 205, 82, 0, 59, 59, 221, 225, 1, 113, 0, 965
110 DATA 221, 9, 221, 54, 0, 25, 62, 8, 215, 221, 1036
120 DATA 126, 0, 203, 39, 203, 39, 203, 39, 60, 221, 1133
130 DATA 150, 0, 95, 221, 54, 1, 0, 22, 1, 221, 765
140 DATA 54, 2, 7, 123, 221, 150, 2, 71, 221, 78, 929
150 DATA 1, 205, 170, 34, 71, 4, 126, 7, 16, 253, 887
160 DATA 230, 1, 130, 130, 87, 221, 53, 2, 32, 229, 1115
170 DATA 215, 221, 52, 1, 32, 217, 62, 10, 215, 221, 1246
180 DATA 53, 0, 32, 188, 62, 8, 215, 22, 0, 74, 654
190 DATA 6, 0, 205, 170, 34, 71, 4, 126, 7, 16, 639
200 DATA 253, 230, 1, 198, 128, 215, 20, 32, 236, 62, 1375
210 DATA 10, 215, 1, 15, 0, 201, 0, 0, 0, 0, 442
```



# C.ITOH

# ABC Data®

## DRUKARKI 9-IGŁOWE

Typ	Pr. dr. (norm./NLQ)	DM
CI-4000-10	400/87 zn/sek	4.430
CI-4000-20P	400/87 zn/sek	4.270
CI-4000-20S	400/87 zn/sek	4.270

W cenę drukarki wliczony jest interfejs oraz moduł znakowy.

Kasety barwiące	DM
Opakowanie 6 sztuk	80
Moduły znakowe	DM
Model 10	518
Model 20P	448
Model 20S	448



Instrukcje obsługi	DM
W jęz. angielskim	140
W jęz. niem.	40

## Drukarki 24-IGŁOWE

Typ	Pr. dr. (norm./NLQ)	DM
T161	300/100 zn/sek	3.758



Urządzenia dodatkowe	DM
Automatyczny podajnik kart:	
pierwsza komora	581
druga komora	385
traktor	371

Powyższe modele drukują na papierze o szerokości do 406 mm.



Przedstawiamy Państwu nowy produkt, bestseller w zastosowaniach przemysłowych i wszędzie tam, gdzie liczy się szybkość, jakość, wytrzymałość. Dzięki zastosowanym rozwiązaniom (różne rodzaje druku i różne emulacje zapisane na wymiennych kartach) drukarki firmy C.ITOH spełniają wymagania wielu różnych klientów.

Na uwagę zasługuje bardzo wysoka prędkość (do 570 zn/sek. w modelu C 815). Godna podkreślenia jest wysoka żywotność głowicy (do 100 mln znaków w C 715, 1 miliard znaków w C-4000). Zastosowana konstrukcja prowadzenia papieru umożliwia także drukowanie etykiet samoprzylepnych.

Nowością na rynku drukarek jest 24-igłowa, bardzo szybka 7-kolorowa drukarka z szerokim 15" wałkiem C.ITOH 715 C.

Zainteresowanych innymi produktami prosimy o bezpośredni kontakt



## C.ITOH

## houston

## Roland DIGITAL GROUP

## AMPEX

# ABC Data

**Ditmar-Koel-Str. 13**  
**2000 Hamburg 11, RFN**  
**tel. 040 314003/3195874**  
**tlix. 2166 002 abch d**  
**fax. 040 3191783**

### ZAMÓWIENIE

Niniejszym zamawiam następujące artykuły:

..... sztuk..... DM.....  
 ..... sztuk..... DM.....  
 ..... sztuk..... DM.....  
 ..... sztuk..... DM.....  
 Transport (1 szt. DM 40 .....  
 Kwota pobierana przez bank ..... DM 10,-  
 Razem: ..... DM.....

W załączeniu kopia zlecenia bankowego na przelew w/w sumy na konto ABC Data GmbH w Dresdner Bank, Hamburg, RFN, BLZ: 200.800.00, konto nr: 061 612 61.

W/w sprzęt proszę wysłać na adres:  
 Nazwisko i imię .....  
 Kod pocztowy i miejscowość .....  
 Ulica i numer domu ..... tel. ....  
 Nazwisko, imię i adres zamawiającego (jeśli inne niż odbiorcy): .....  
 data ..... podpis .....

### JAK ZAMAWIAĆ ARTYKUŁY OFEROWANE PRZEZ ABC DATA?

- Po dokonaniu przelewu prosimy o wystanie do nas załączonego zamówienia lub krótkiego listu (najlepiej na odwrocie kserokopii dowodu wpłaty) z dokładną informacją, co Państwo zamawiają i na jaki adres towar ma być wysłany.
- Wyroby nasze możecie Państwo również kupić osobiście w Berlinie lub na wysytkę z Bonn.
 

ABC Computersystems	ABC Data Bonn
Alt-Moabit 80	Augustastr. 40
1000 Berlin 21	5300 Bonn 2
tel. (030) 3915090/99	tel. (0228) 354480/90
tix. 18 13 65 abc d	tix. 885566
fax. (0049-30) 3936483	fax. (0049-228)355635
- Na wszystkie zakupione u nas produkty udzielamy 1-letniej gwarancji z obsługą serwisową prowadzoną w Polsce przez zakłady państwowe i prywatne.  
 ABC Data GmbH jest firmą zarejestrowaną w Amtsgericht Hamburg, HRB 38590.



Oto prawdziwa polska adventure — „Puszka Pandory”. Dwa i pół roku temu można ją było we wrześniu kupić na Bajtkowej giełdzie, w 1987 roku latem była w Bajtku jej reklama, ale do tej pory niemal nikt nic o niej nie wie, a to pierwsza tego typu gra u nas, w dodatku napisana w assemblerze (na Spectrum 48K). Napisał ją M. Borkowski — może to ta sama osoba która niedawno pisała o trójwymiarowych wykresach?

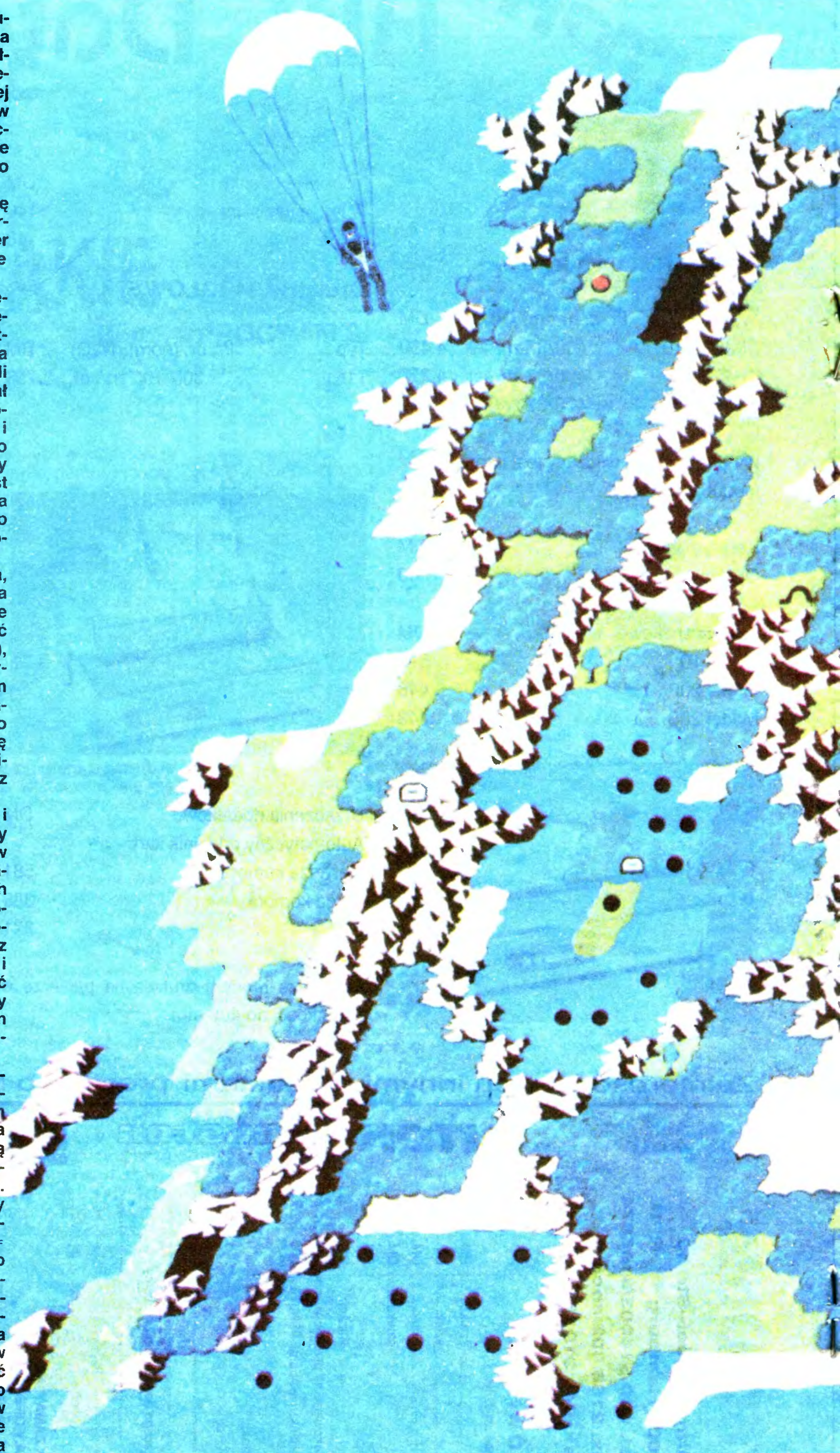
Nie jest to może gra o jakiej marzy się przez całe życie, ale na pewno nie jest gorsza od wielu sprowadzonych do nas gier brytyjskich, a ma tę niewątpliwą zaletę, że nie wymaga znajomości angielskiego.

Naszym zadaniem jest rozbrojenie systemu raketowego znajdującego się na niewielkiej wyspie. Nie mamy żadnych dodatkowych informacji oprócz tego, że rakietą na pewno gdzieś tu jest. W ostatniej chwili przed lądowaniem nasz śmigłowiec został zestrzelony przez automatyczne działko zostawione przez konstruktorów systemu, i tylko dzięki skokowi na spadochronie udało nam się ocalić życie. Używać możemy wszystkiego, co uda się znaleźć, a niej jest to tak mało jak by się mogło zdawać — na wyspie znajduje się sporo różnorodnego sprzętu, wyrzuconego przez nas ze śmigłowca tuż przed skokiem.

Jak to zwykle w grach tekstowych bywa, polecenia wydajemy pisząc krótkie zdania (np. „WZIĄĆ MŁOTEK”) albo pojedyncze słowa (np. „SKOCZYĆ”). Możemy poruszać się w czterech kierunkach, pływać (w H<sub>2</sub>O), czasami wspinać się na drzewa, a także używać przedmiotów, rozrzuconych po całym terenie wyspy, często w dość nieoczekiwanych miejscach. Program rozpoznaje około dwudziestu słów, toteż komunikowanie się z komputerem jest możliwe w raczej ograniczonym zakresie, ale na ogół przebiega bez zakłóceń.

Scenariusz nie jest zbyt rozbudowany, i występują w nim niemal wyłącznie elementy potrzebne, nie ma dodatkowych wątków rozpraszających uwagę. Nie znaczy to wcale, że gra jest nudna, i nie szykuje żadnych niespodzianek. Podłoże pod napięciem, komora gazowa przez którą trzeba się przedostać, zalany labirynt — to tylko niektóre z pułapek. I ostatnia przeszkoda, trudna albo i łatwa, zależnie od gracza — konieczność odgadnięcia hasła unieruchamiającego cały system. (Jeżeli ktoś się zaciął właśnie w tym miejscu, niech poszuka natchnienia w wakacyjnych numerach Fantastyki z 1983 r.)

Do wad gry przede wszystkim trzeba zaliczyć niemożność nagrania na kasetę jej aktualnego stanu, co powoduje, że za każdym razem trzeba zaczynać od początku. Na szczęście wszystkie przedmioty zawsze leżą w tych samych miejscach, toteż nie ma problemu z ich ponownym skompletowaniem. Druga sprawa to edytor — bardzo wygodny (polskie litery otrzymuje się naciskając odpowiedni klawisz z shiftem — np. (ł = shift + L) dopóki nie popełni się błędu, bo użycie klawisza DELETE powoduje skasowanie nie ostatniej litery, a całego wprowadzonego tekstu. No i na koniec — mapa kolos. Obszar gry to kwadrat czterdzieści na czterdzieści czyli tysiąc sześćset pozycji w których można się znaleźć. Żeby narysować mapę trzeba zużyć kilkanaście godzin, co może być zabawne tylko dla kartografów (inna sprawa, że są gry z mapami znacznie większymi — np. „Lords of Midnight”). Ta



# PUSZKA PANDORY





wersja mapy którą widzicie obok nie jest całkiem kompletna — brak na niej kilku elementów, a przede wszystkim nie są zaznaczone te miejsca, w których warunki terenowe nie pozwalają przejść na sąsiednie pole. Dotyczy to zwłaszcza gór i skał.

Teraz kilka słów na temat grafiki i dźwięku, bo w każdej poważnej recenzji zamieszcza się tego typu informacje. W zasadzie ani jednego ani drugiego w „Puszczy Pandory” nie ma, chociaż nie od razu się to zauważa. Ekran jest tak zaprojektowany, że w jego centralnej części przesuwa się fragment krajobrazu — lasu, wydm, gór itd. w zależności od tego na jakim terenie aktualnie się znajdujemy. Wszystkim naszym poczynaniom towarzyszą krótkie piknięcia — gorzej jest z muzyką, która pojawia się tylko w jednej sytuacji — jako krótki marsz pogrzebowy. Trochę mało, ale lepsze to od ciągłego, nie dającego się wyłączyć brzęczenia — a takie gry też istnieją.

Na koniec dla chętnych kilka wskazówek ułatwiających grę. Swoje działania na terenie wyspy trzeba skoncentrować na trzech punktach — kłapie w górach, drzewie na północnym brzegu jeziora i bunkrze na wysepce na jeziorze. Wchodzenie głęboko w las bez kompasu mija się z celem, na szczęście kompas leży na północny zachód od miejsca w którym zaczyna się gra, i łatwo go odnaleźć. Z kompasem czy bez — wejście w obszar anomalii magnetycznej kończy się zabłądzeniem, ale żeby grę ukończyć trzeba tam niestety zajrzeć. Warto rozejrzeć się po mieliznach — można tam znaleźć coś przydatnego.

Do bunkra w górach najlepiej wejść na samym początku gry, kiedy nie ma się jeszcze nic do stracenia. Próba pokonania zalanych korytarzy bez akwalungu jest z góry skazana na niepowodzenie. I ostatnia uwaga — posiadanie sprzętu to jeszcze nie wszystko — trzeba go uruchomić lub umiejętnie używać.

Ogólnie rzecz biorąc, gra jest całkiem udana i potrafi wciągnąć. Szkoda tylko że jest tak mało rozpropagowana, od jej ukończenia minęło już co najmniej dwa lata, i do dziś znana jest najwyżej kilkudziesięciu, może kilkuset osobom, a na pewno zasługuje na szersze rozpowszechnienie.

Karol B. Mirowski

ZYGMUNT ZAKRĘKIEWICZ

# DRY

- - LICZBA GRY
- - POLE MINOWE
- - DRZEWO W SKALE
- - GŁAZY SKAŁY
- | - MIELIZNA
- A - ANOMALIA MAGNETYCZNA
- ⌒ - JASKINIA
- ▩ - KLAPA
- - JEZIORO, MOKRE
- - POLA CZARNE SA NIEDOSTĘPNE
- ☐ - BUNKIER
- 💡 - DUŻE DRZEWO
- ▩ - LASY
- ▩ - WADMA
- ▩ - LAZAR
- ▩ - SCHODY



# 10

BAJTKOWA LISTA PRZEBOJÓW (4/89)

W tym miesiącu, podobnie jak w poprzednim, Listę opanowały gry wydane w 1988 roku. Mimo to, obok nich trzymają się produkty zaliczane raczej do „staroci” w świecie natógowych graczy. Ciągłe czekamy na grę „Double Dragon”, która jeszcze (a jest luty) nie dotarła do Polski. Czy jest aż tak świetna, jak o niej piszą — przekonamy się wkrótce. Na kwietniowe notowania Listy nadeszło 1918 propozycji. Czytelnicy głosowali na 107 tytułów.

1	GRYZOR		x	x	x
2	MOUSE TRAP		x	x	x
3	PLATOON		x	x	x
4	ARMY MOVES	↑		x	x
5	UNIVERSAL HERO	↑	x	x	x
6	GREEN BERET	↓	x	x	x
7	STREET FIGHTER	!		x	x
8	VIRUS	!		x	x
9	STRONG MAN	↓		x	x
10	SKATE CRAZY	↓		x	x



## THE TRAIN

**Gra ta** przenosi nas w czasy kampanii hitlerowskiej we Francji. Prawie cały kraj jest już opanowany przez wojska niemieckie, tylko niektóre miasta są jeszcze w rękach francuskich. Ty, wcielając się w postać jednego z członków ruchu oporu, uczestniczysz w akcji porwania pociągu pancernego.

Zaczynasz, atakując stację METZ. Twój kolega Pierre, musi przestawić zwrotnicę ustalając tor (jeden z trzech), którym będziesz jechał. Twoje zadanie natomiast polega na ośmianiu Pierra przed ogniem esesmanów, strzelających z okien dworca. Jesteś uzbrojony w karabin maszynowy oraz w granaty.

Gdy wybierzesz już trasę, którą będziesz podążał, zaczynają się kłopoty. Wchodzisz do parowozu i widzisz mnóstwo dźwigni, przetłączników, liczników. Trudno domyślić się, do czego służą, podaję więc ich umiejscowienie wraz z funkcją:

- 1 — THROTTLE (przepustnica) — znajduje się w lewym górnym rogu, służy do regulowania prędkości pociągu,
- 2 — FURNACE DOOR (drzwi do pieca) — umieszczone pod przepustnicą, palisz w nich węglem, zwiększając temperaturę w kotle,
- 3 — FORWARD REVERSE LEVER (skrzynia biegów) — znajduje się w prawym dolnym rogu, są dwa biegi — do przodu i do tyłu.
- 4 — STEAM BLOWOFF (zawór parowy) — leży tam, gdzie skrzynia biegów, służy do zmniejszania ciśnienia pary,
- 5 — BRAKE (hamulec ręczny) — umieszczony po lewej stronie skrzyni biegów,
- 6 — WHISTLE (gwizdek) — znajduje się w prawym górnym rogu.

Obok gwizdka, mamy skalę informującą nas o ilości wody i kierunku ustawienia zwrotnicy. Poniżej są liczniki:

- PSI — ciśnienie pary
- MPH — prędkość pociągu (w milach/godzinnie)
- TEMP — temperatura w kotle

Pamiętaj o następujących rzeczach:  
— hamulec ma ograniczoną wytrzymałość  
— przelączenie biegów podczas jazdy powoduje zablokowanie skrzyni (GEAR STRIPPED)

— zwrotnica (SWITCH) jest przestawiana po użyciu gwizdka (WHISTLE).

Gdy jedziesz dalej, na twojej drodze staje most. Jest on zajęty przez wojska wroga — chronią go okręty niemieckie. Zniszczenie ich nie jest jednak trudne (toną po pierwszym trafieniu), lecz twój pociąg wybuchą po kilku trafieniach przez statki.

Stacje nie muszą być przez ciebie zajęte. Uzupełniasz na nich zapas wody oraz wysyłasz telegraficzny meldunek. Odpowiedź otrzymasz prawie natychmiast. Możesz rozkazać, aby zbombardowano następną stację (TAKE STATION) lub most (TAKE BRIDGE) — zajmiesz je wtedy bez walki. Jeżeli twój pociąg jest zniszczony, możesz zażądać naprawy. W każdym z tych przypadków centrala podaje ci czas, w którym będzie to wykonane. Jeśli nie masz nic do przekazania, nie wysyłasz wiadomości (NO MESSAGE).

Podczas jazdy atakują cię Messershmitty 109, przelatując nisko nad tobą i ostrzeliwując ze swoich karabinów. Jesteś oczywiście uzbrojony w działka przeciwlotnicze oraz w karabiny maszynowe, mimo to zestrzelenie samolotu graniczy z cudem.

Gra kończy się, gdy uda ci się dotrzeć do stacji RIVIERE. Przegrywasz w przypadku, gdy twój pociąg zostanie zniszczony lub stracisz jednego ze swoich ludzi. Zbyt długi postój na torach również kończy się napisem „GAME OVER”.

Pod ekranem graficznym komputer wyświetla aktualny czas (nie biegnie on normalnie lecz jest przyspieszony około 10 razy). Oprócz tego jesteś informowany o ewentualnych zagrożeniach pociągu, np: wrogie samoloty atakują z przodu (ENEMY FIGHTERS ATTACKING FRONT), wrogi most — lepiej zatrzymać się (ENEMY BRIDGE, BETTER STOP) i inne. Najlepsze wyniki są nagrywane na dysk, a jeśli ktoś przekroczy 2000, komputer pisze bardzo przyjemną informację.

Klawisze sterujące:

- 1 — widok z przodu
- 2 — widok z tyłu
- 3 — maszynownia
- 4 — mapa

Komputer: Commodore 64/128  
Atari XL/XE

Luke

## JAPOŃCZYCY I ANGLICY KUPILI PRAWA DO TETRIS

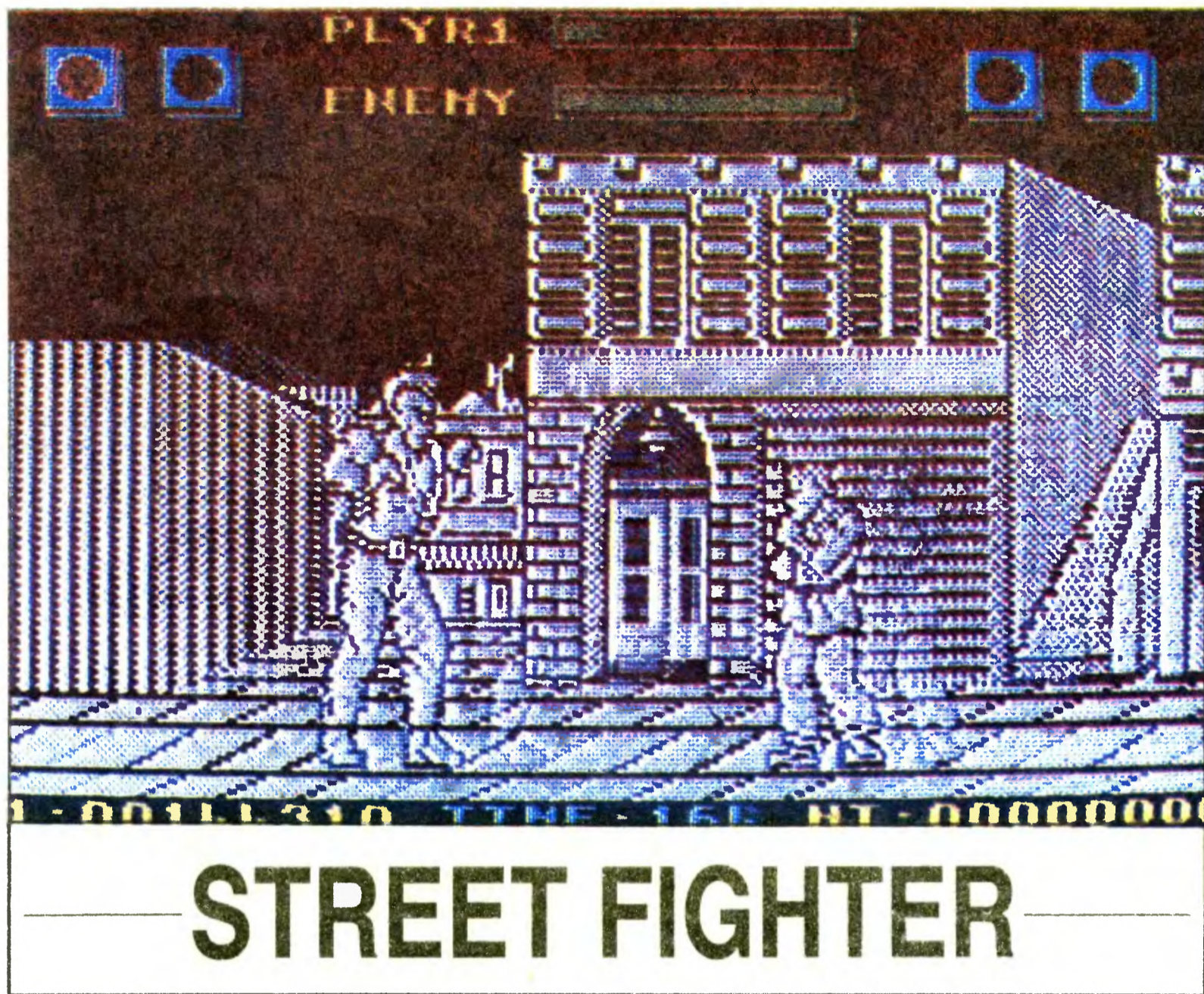
TETRIS, znana w Polsce radziecka gra komputerowa, zdobywa coraz większe uznanie na całym świecie. Radzieckie Zjednoczenie Eksportowe „Elektronorgtechnika” zawarło w końcu lutego br. kontrakt z japońską firmą „Bullet-prof-software” oraz an-

gielską „Andkameda Software”, na mocy którego uzyskały one prawo do wykorzystania TETRISU w automatach z grami komputerowymi, zabawkach elektronicznych i zegarkach.

Warto przypomnieć, że po raz pierwszy ta efektowna gra rozwijająca wyobraźnię przestrzenną i szybkość reakcji, sprzedana została za granicę w roku 1988. Autorem TETRIS jest Andriej Pažitnow, specjalista w Centrum Obliczeniowym Akademii Nauk ZSRR. (ws)







## Ostatnio

w środkach masowego przekazu coraz częściej mówi się o tym, co dzieje się na ulicach wielu miast całego świata po zapadnięciu zmroku. Dziennikarze zaczynają bić na alarm: to już nie pojedyncze incydenty ale prawdziwe, zorganizowane potyczki. O co chodzi? Ktoś nie obeznany z tematem zbagatelizowałby sprawę: kilku chłopców pokłóciło się i pobiło. Tak jednak nie jest. Bardzo często liczba walczących jest duża a w użyciu są nie tylko pięści i nogi ale także pałki, łańcuchy i kastety. Prawie zawsze musi interweniować policja, a niektórzy uczestnicy tych bójek lądują w szpitalu. Na szczęście w naszym kraju takie wydarzenia mają miejsce stosunkowo rzadko. Dla wszystkich łaknących emocji mam alternatywę: grę „Street Fighter”. A więc joystick zamiast pałki, monitor zamiast ulicy i można zacząć całkowicie bezpieczną zabawę.

Gra składa się z pięciu oddzielnie doładowywanych części, z których każda dzieje się w innym kraju. Nasze zadanie jest proste — należy pokonać przeciwnika (nota bene wyglądającego charakterystycznie dla kraju, w którym toczy się dana walka). Dokonać tego możemy po wygraniu trzech rund; rundę przegrywa ten, który straci całą energię lub, w przypadku skończenia się czasu — ten, który ma jej mniej. Jako broń służą nam (a także przeciwnikowi) ręce i nogi a ilość technik, jakie możemy wykonać, jest zbliżona do tych, jakie mamy w „The Way of the Exploding Fist”. Szkoda jednak, że nie mamy do dyspozycji jakiegokolwiek innej broni, gdyż niewątpliwie uatrakcyjniłoby to grę.

Gdy już uporaliśmy się ze swoim przeciwnikiem (co nie jest trudne, szczególnie dla tych, którzy dobrze znają wspomnianą „The Way of the Exploding Fist”), czeka nas na danym etapie jeszcze jedna próba. Otóż nasz bohater pojawia się przed stertą cegieł, które musi rozbić. Nad jego głową znajduje się wskaźnik energii, wahający się szybko między pozycjami maximum i minimum. Wciśnięcie strzału blokuje wskaźnik i można odczytać, ile cegieł jesteśmy w stanie rozbić jednym uderzeniem.

Jak już wspomniano, gra nie należy do trudnych — można ją skończyć już po 2—3 godzinach zabawy. Jest też bardzo podobna do innych gier karate; taki sam zestaw ciosów, przeciwnik jakby łatwiejszy do pokonania. Jest jednak coś, co przemawia na korzyść „Street Fighter” — tym czymś jest grafika. Figurki walczących są duże (dochodzą nawet do wysokości 1/3 ekranu) i narysowane z dużą ilością szczegółów. To samo dotyczy scenerii, w której toczy się walka. Widać dbałość autorów o dokładne odtworzenie rzeczywistości. Niestety, istotną wadą tej gry jest kiepska animacja oraz (gdy gramy z komputerem) schematyczność ataków przeciwnika. Ale i tak chyba lepsze to niż prawdziwa walka na ulicy z realnym przeciwnikiem.

**Firma: U.S. Gold**  
**Komputer: ZX Spectrum 48/+, Commodore 64/128, Amstrad/Schneider, Amiga, Atari ST**

(mz)

## KRÓL I KRÓLOWA GIER



**Mateusz Kołasa**, lat 12  
 chodzi do szkoły 130 w Warszawie  
 ulubione gry: Bruce Lee, Strip Fire  
 hobby: rower, narty, języki obce.



**Kasia Wyporska**, lat 11  
 chodzi do szkoły 333 w Warszawie  
 ulubione gry: Boulder Dash,  
 Matterhorn Screamer  
 hobby: plastyka, ceramika,  
 filmy z różową panterą.

# S.O.S.

Nie wiem, jak wystartować w grze ACE, a w grze GHOST BUSTERS — jak ją rozpocząć. ZX Spectrum+.

**Arkadiusz Stelwach**, ul. Spółdzielcza 29 m 37,  
 18-400 Łomża

Szukam dwóch klawiszy do magnetofonu Atari 1010 i opisu do UNIVERSAL HERO. W zamian opisy do innych gier.

**Bartosz Grabski**, ul. Celulozowa 14 m 12,  
 71-869 Szczecin, tel. 239-298

Od dłuższego czasu próbuję rozgryźć następujące gry: FREAKY FACTORY, MEGAGUN, WAR GAMES, NECROMANDER, SPY STRIKES BACK. Nie wiem też, jak dostać się do skarbcza w Montezumie. Proszę o pomoc. Mam Atari 800 XL. W zamian opisy do innych programów.

**Sebastian Szafran**, ul. Nałęczowska 9 m 3,  
 24-320 Poniatowa, woj. lubelskie

Mam Spectrum. Szukam opisów do gier: SCPTRE OF BAGDAD, DEATH WISH 3. W zamian inne na ten komputer.

**Wojtek Kubisz**, ul. Hawajska 17 m 62,  
 02-776 Warszawa, tel. 641-11-88

Uprzejmie proszę o dokładny opis gry TRAP DOOR. W zamian inne opisy.

**Maciej Szewczyk**, ul. Kopernika 9,  
 05-220 Zielonka k/W-wy

Poszukuję opisów do gier: INFILTRATOR, UNIVERSAL HERO i CORONIS RIFT na Atari 65 XE. W zamian inne.

**Janusz Wojtyna**, ul. Oświęcimska 85 m 22,  
 41-707 Ruda Śląska 7

Jak uzyskać nieśmiertelność w grze COBRA GAME? Nie wiem, co zrobić po zebraniu trzech złotych strzał w grze ROBIN OF WOOD. Obie na C-64.

**Wojciech Figurski**, ul. Sucharskiego 21b m8,  
 81-157 Gdynia

Pilnie szukam gier: BARBARIAN, HARD BALL, WINTER GAMES, HOBBIT w wersji kasetowej na Atari 65 XE. W zamian oferuję około 100 gier i 15 programów użytkowych.

**Maciej Sobera**, ul. Zygm. Starego 20,  
 26-900 Kozenice

WINTER GAMES i BARBARIAN na Atari 65 XE — szukam.

**Tomasz Męzydło**, ul. Migaly 35 m 9,  
 81-355 Gdynia

O co chodzi w grach QUARK IX i CAVERNS OF KHAFKA? Potrzebuję nieśmiertelności do gier BRIGLES 1, BOMB JACK, HUNCH BACK?

**Bartłomiej Walczak**, ul. Leśnej Płanki 63/65 m 2,  
 03-142 Warszawa

Od kilku miesięcy bezskutecznie poszukuję instrukcji do TURBO KOPIERA R and R Software ze Szczecina. Komputer Atari 65 XE.

**Tomasz Kozubek**, ul. Kusocińskiego 7 m 2,  
 48-303 Nysa

Bardzo proszę o dokładny opis do gry TARZAN. Atari 65 XE.

**Arkadiusz Śliwa**, ul. Zamiejska 9 m 18,  
 03-580 Warszawa

Mam Atari 130 XE i XC 12. Szukam gier: BOMB JACK, MIKIE, WINTER GAMES, MASTER KUNGFU. W zamian 300 innych oraz wiele opisów.

**Piotr Seweryński**, ul. Reja 2 a, 82-300 Elbląg

Na czym polega gra BOMB JACK na Commodore 116?

**Przemysław Karolczuk**, Os. Sikorskiego 11e m 6,  
 44-244 Żory, woj. katowickie

Nie mogę zdobyć w mojej okolicy gry WINTER GAMES. W zamian za nią oferuję wiele innych z nieśmiertelnościami.

**Tomasz Cała**, ul. 22 Lipca 22 m 3,  
 67-100 Nowa Sól

Szukam opisu do gry BARBARIAN na Atari 65 XE. W zamian opisy do innych gier.

**Krzysztof Mazur**, ul. Akacja 8 m 20,  
 40-102 Katowice, tel. 542-619

COMMANDO COBRA, KUNGFU, RAMBO na Spectrum — szukam planów i opisów.

**Paweł Dąbski**, ul. Kalinowa 12 m 73,  
 21-040 Świdnik, woj. lubelskie

Pilne! Szukam dwóch gier na Atari XE: MONTEZUMA'S REVENGE i CHUCKIE EGG.

**Bartek Brzozowski**, ul. Moczyńskiego 3 m 11,  
 87-100 Toruń



# KONKURS TYLKO DLA GRACZY!

Proponujemy Wam dzisiaj nietypowy konkurs. Do rozpoznania jest 16 postaci wyjętych „żywcem” ze znanych i popularnych gier komputerowych. Tytuł gry należy wpisać do diagramu, a litery oznaczonych pól czytane z góry do dołu utworzą rozwiązanie.

Gdy tytuł składa się z kilku słów, do krzyżówki należy wpisać je bez spacji, myślników i podobnych separatorów, rozwiązanie będzie w podobnej postaci.

Rozwiązania nadsyłajcie na adres redakcji: „Bajtek”, ul. Wspólna 61, 00-687 Warszawa, w terminie do 31 maja br.

Wśród tych, którzy nadeślą poprawne rozwiązanie, rozlosujemy joysticki i zestawy programów komputerowych.



1 3



16



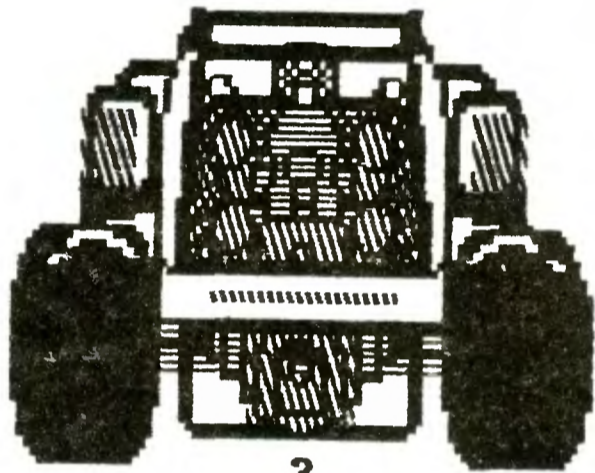
9



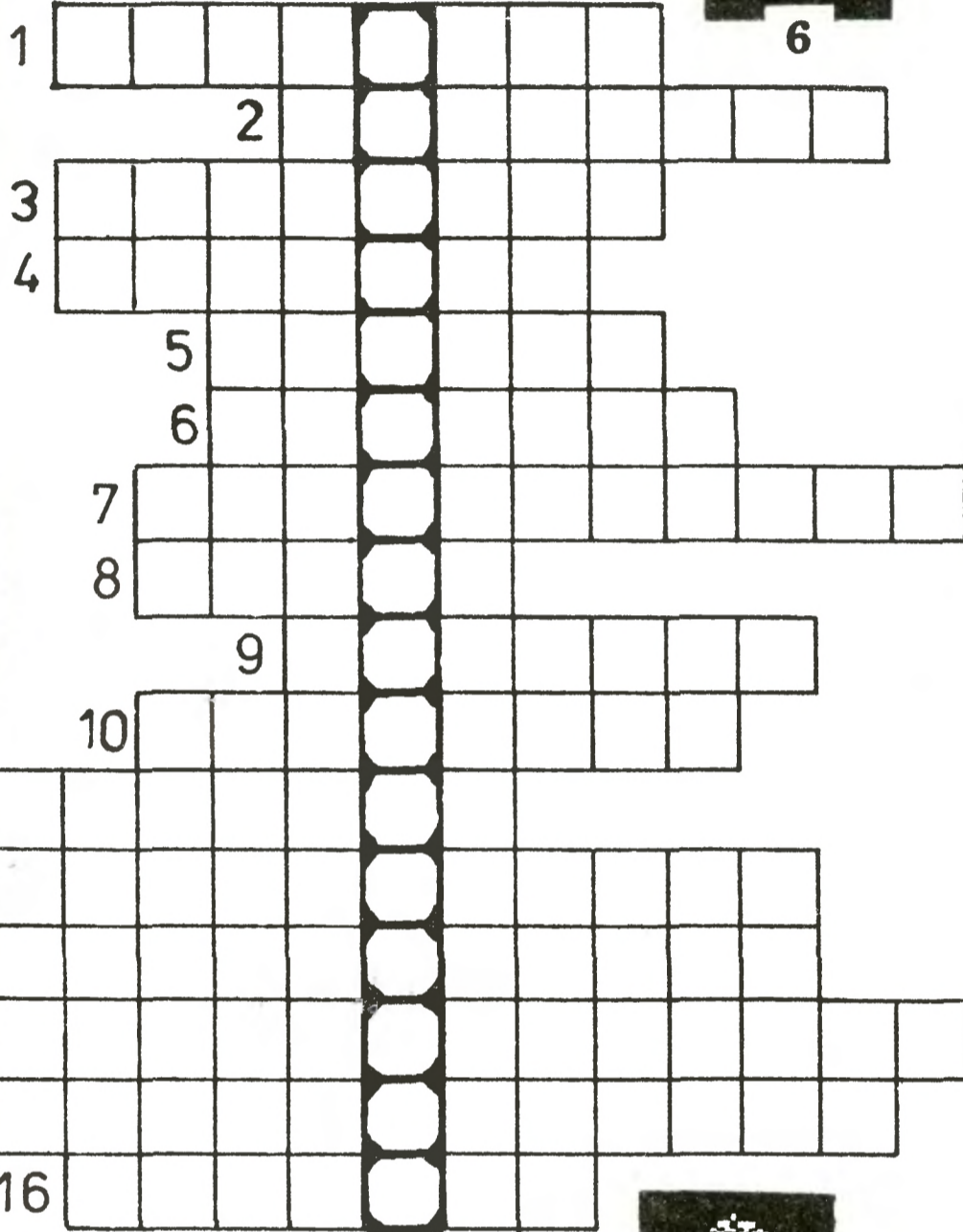
6



7



3



5



2



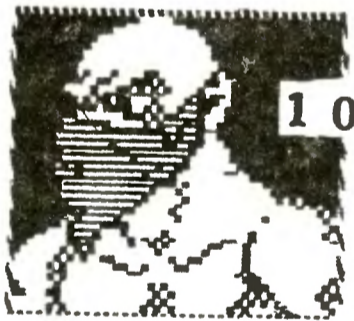
12



4



14



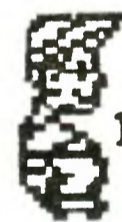
10



1



8



11



15

## ZNÓW POKE-rzysta

Dziś większa porcja POKE-ów, dla niestrudzonych i nienasyconych:

Basil — 41296,0: 41968,201  
 Batman — 36798,0  
 Batty — 47633,0  
 Bionic Commando — 34690,0  
 Booty — 58294,0  
 Bounder — 36610,0  
 Bubbler — 57517,0  
 Bobby Bearing — 28094,36  
 Bubble Bobble — 43871,52  
 Bombjack II — 25379,0  
 Barbarian (Palace) — 51005,255  
 Barbarian (Psygnosis) — 27680,0

Car Wars — 32337,0  
 Catch 23 — 46813,0: 61635,0  
 Cauldron — 40060,0  
 Cauldron II — 52133,0  
 Chiller — 34025,0

Chuckie Egg — 42873,0  
 Chuckie Egg II — 35453,0  
 Cobra — 36515,183  
 Critical Mass — 56879,0  
 Cybernoid — 24917,255  
 Cybernoid II — 32202,0

Dan Dare — 23974,168  
 Dan Dare 2 — 23453,237: 23450,212  
 Death Wish 3 — 38678,183:  
 39353,183: 43301,183  
 Deflektor — 34473,8: 41784,0:  
 42707,201  
 Dizzy — 54216,0  
 Driller — 48246,0  
 Druid — 24890,201  
 Druid 2 — 30012,58  
 Dragon's Lair — 51867,0  
 Elite — 46848,201

Enduro Racer — 43542,0: 43643,0  
 Exolon — 38120,201: 31646,0:  
 35456,0: 42338,0: 36845,0: 38221,0  
 Eskimo Eddie — 24686,24: 24687,76

Fat Worm — 30624,201  
 Fire Lord — 34509,0  
 Freddy Hardest 1 — 64011,24  
 Freddy Hardest 2 — 61607,183  
 Futre Knight — 31683,0  
 Fawilight 2 — 30429,0: 32027,24:  
 31341,0  
 Firefly — 44997,255  
 Flying Shark — 54462,201: 54379,3

Ghosts'N Goblins — 33352,201:  
 33433  
 Goonies — 33400,183  
 Greet Gurianos — 34962,0  
 Green Beret — 42076,0  
 Gryzor — 33015,255  
 Great Escape — 41953,183  
 Garfield — 33595,0  
 Gyroscope — 53887,201: 59149,0  
 Gothik — 43934,58: 42110,58

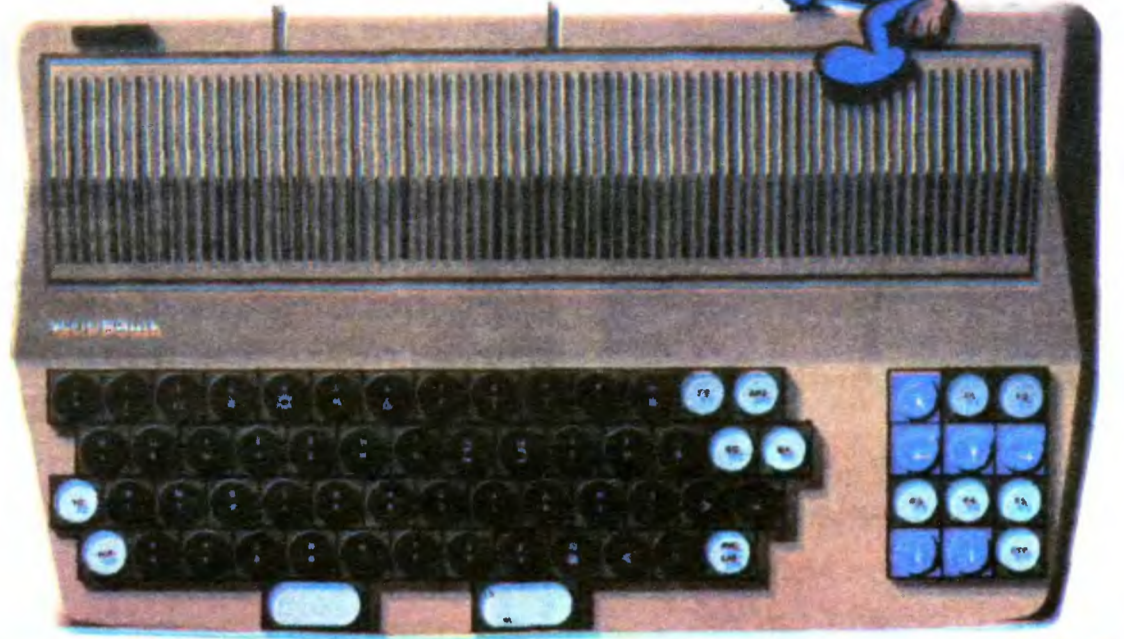
Head Over Heels — 43132,0: 42195,0:  
 35315,0  
 Hysteria — 44588,201  
 Heist 2012 — 36106,12: 36190,12

Indiana Jones — 33948,0  
 I Ball 2 — 43384,0: 49483,0  
 Ikari Warriors — 40272,0  
 Impossaball — 41185,0: 37706,0:  
 37539,0  
 Into the Eagle's Nest — 36640,0:  
 36641,0: 40512,0: 40513,0

Jack the Nipper — 44278,58: 44285,58  
 Jack the Nipper 2 — 43251,0  
 Jail Break — 50651,0  
 Joe Blade — 65032,50: 65029,50  
 Judge Dredd — 24936,24

Karnov — 32972,0  
 Knightmare — 38686,16: 38693,16  
 Kokotoni Wilf — 42214,255  
 Krakout — 46565,0  
 Kai Temple — 47783,0: 47824,0  
 Killer Ring — 33636,0  
 Kosmic Kanga — 36212,0





## Korespondencja własna z Moskwy

Znajomy widok: duża sala, na kilkunastu stolikach komputery, tłum dzieci wpatrzonych w kolorowe ekrany monitorów... Może tylko klawiatury wyglądają nieco inaczej, niż te, do których się przyzwyczailiśmy. I gwar dzieci, choć przecież taki sam jak wszędzie na świecie, to jednak trochę inny... Nic zresztą dziwnego — jesteśmy w Moskwie, przy ulicy Dmitrowskiej, w klubie komputerowym — „Agat”.

Przedstawiciel „Bajtka”, korzystając z praw gościa, wszedł do klubu bezpłatnie, ale wszyscy inni muszą wykupić bilet, gdyż „Agat” działa na zasadzie odpłatności. Za godzinę przebywania w sali, w której znajdują się najprostsze radzieckie komputery domowe o nazwie „Mikrosza” należy zapłacić 50 kopiejek. Za godzinę w sali „Agatów”, czyli radzieckich komputerów osobistych z prawdziwego zdarzenia, należy zapłacić 1 rubel 20 kopiejek.

Tyle właśnie zapłacił **Denis Tichonow**, uczeń V klasy, który przyszedł tu po 4 lekcjach. Nie żałuje tego wydatku, gdyż w zamian ma do dyspozycji komputer, fachowego instruktora i dostęp do wszystkich programów, jakie znajdują się w klubie. Na razie Denis „testuje” kolejne gry.

— Widocznie tak już musi być na całym świecie — mówi oprowadzający mnie po klubie **Siergiej Pawłowski** — że nie da się tego etapu przeskoczyć!

Siergiej Pawłowski, lat 22, pracuje jako technik — programista w Lianazowskich Zakładach Elektrotechnicznych. Są to te właśnie zakłady, które montują „Agaty” (3 tysiące sztuk rocznie). Są one również patronem, a właściwie właścicielem klubu „Agat”, gdyż mieści się on w pomieszczeniach należących do Zakładów. Ich własnością są komputery i cały sprzęt klubu, a instruktorzy pobierają pieniądze z kasy Zakładów, co zresztą zaczyna już obecnie stawać się barierą w rozwoju klubu, ale o tym za chwilę.

Klub powstał w 1986 roku. Początkowo dysponował dwoma salami komputerowymi i pracownią techniczną. Obecnie istnieje również w klubie silna grupa programistów, wyodrębniona w oddzielną pracownię. Niedawno uruchomiono też pierwszą w Moskwie wypożyczalnię komputerów. Co tydzień — wyjaśnia szefowa wypożyczalni, **Marina Carkowa**, lat 19 — przyjeżdża do nas ktoś z innej dzielnicy, a nawet z innej republiki, aby dowiedzieć się na jakich zasadach działamy i czy to się opłaca.

Opłaca się! Miesięcznie wypożyczalnia daje do kasy klubu 6—7 tysięcy rubli, a w czasie wakacji nawet 9—10 tysięcy. Stawki nie są wysokie — tak przynajmniej uważa Marina. 80 kopiejek za dzień wypożyczenia „Mikroszy” i 6,5 rubla za dzień wypożyczenia „Agata”. Różnica w stawkach jest oczywista, bo przecież „Mikrosza” kosztuje w sklepie (jeśli już tam trafi) około 500 rubli, natomiast „Agaty” sprzedawano niedawno oficjalnie po 4900 rubli. Poza tym — Marina już tego nie powie, ale tak twierdzą znajomi komputerowcy z Moskwy — „Agaty” częściej się psują, co oznacza, że więcej kosztuje ich obsługa.

Wypożyczalnia to jednak zaledwie „działalność społeczna” przy zyskach jakie daje klubowi grupa programistów. W ubiegłym roku klub zarobił dzięki programom 160 tysięcy rubli. Program obsługi bazy danych kosztuje na przykład w „Agacie” 2000 rubli. Ta działalność softwarowa zaczyna się zresztą stawać

## KSZTAŁCENIE NARYBKU

przyczyną kłopotów klubu z patronem. Opiekuńcze skrzydła Lianazowskich Zakładów Elektrotechnicznych są niezastąpione, bo fabryka daje serwis. Ale jednocześnie fabryka zaczyna stawiać przeszkody, gdy tylko trzeba zatrudnić w klubie (i nieźle zapłacić) specjalistę z zewnątrz. W Związku Radzieckim zachodzą jednak tak duże zmiany w systemie gospodarczym, że Siergiej ma nadzieję, iż również ten problem uda się wkrótce, ku zadowoleniu obu stron, rozwiązać. Zwłaszcza, że i fabryka zyskuje sporo na sławie klubu. Bo przecież dzięki temu, że „Agat” był jednym z pierwszych klubów komputerowych w ZSRR, i to klubem prężnie działającym, dobrą prasę ma również patron. Dyrektor Lianazowskich Zakładów Elektrotechnicznych świetnie zdaje sobie z tego sprawę!

Rozmawiałem z wieloma bywalcami „Agata” o tym, co daje im klub, jak powinny działać takie placówki itp. Główny sens takiego spędzania wolnego czasu widzą w podnoszeniu poziomu swej wiedzy informatycznej, w możliwości nawiązania kontaktów z podobnymi fanatykami komputerów, w wymianie informacji na temat nowości na tym rynku. Odpowiedzi te pokrywają się z wynikami badań socjologicznych na temat klubów komputerowych, przeprowadzonych niedawno w Związku Radzieckim przez naukowców z Wyższej Szkoły Komsomolskiej.

Oto niektóre rezultaty tych badań: przeważająca część (91,3 proc.) bywalców radzieckich klubów komputerowych to ludzie w wieku do 30 lat, przede wszystkim uczniowie szkół podstawowych (66,1 proc.), techników (10,5 proc.) i studenci (6,8 proc.).

Wielkość klubów: 36 proc. respondentów odpowiedziało, że ich klub liczy ponad 100 osób; 20 procent respondentów jest członkami klubów liczących 30 do 50 osób. Małych klubów, grupujących do 10 osób, jest 24 procent.

Są to w większości kluby bardzo młode. 2 lub 3-letni staż ma zaledwie 20 proc. klubów. 40 procent klubów powstało przed rokiem, a 36 procent — zaledwie kilka miesięcy temu.

Jeśli chodzi o wyposażenie klubu, to żaden z ankietowanych nie stwierdził, iż jego klub posiada wystarczającą ilość sprzętu.

Tylko 14,6 procent ankietowanych posiada komputery w domu. Najczęściej są to radzieckie BK-0010 i japońskie Yamahy. Dostęp ankietowanych

do komputerów wygląda wg wyników badań następująco:

Nazwa komputera	Mam w domu (procent ankietowanych)	Mam dostęp w pracy (proc. ankietowanych)
BK-0010	11,2	24,2
Yamaha	1,9	28,9
Mikrosza	1,2	1,6
IBM	0,6	5,3
DWK	0,3	23,3
Agat	0,3	10,6

**Ilia Wołkow**, lat 22, z którym przegadaliśmy 1,5 godziny w klubie o zaletach programowania w Pascalu, nie ma jeszcze w domu żadnego komputera. Ale za to w pracy ma dostęp do „Agata” i to mu na razie wystarcza. Jeśli natomiast chodzi o oprogramowanie, to zgadza się, iż potrzeby są w tym zakresie olbrzymie. Przecież w radzieckich szkołach zaczęto od 1985 roku wdrażać nowy przedmiot: „Podstawy informatyki i techniki obliczeniowej”. Brakuje jeszcze do niego zadowalającej bazy programów dydaktycznych. Również coraz liczniejsza obecność różnego typu komputerów osobistych w radzieckich biurach, instytutach i w przemyśle stwarza lawinowy wzrost zapotrzebowania na dobry software.

Sekcja programowania w „Agacie” nie musi się więc przez najbliższych kilkadziesiąt lat obawiać o klientelę. Członkowie sekcji coraz więcej czasu czynią natomiast spędzać w salach komputerowych klubu. Obserwują przychodzących tu chłopców (bo oni przeważają wśród gości), rozmawiają z nimi, podsuwają im ciekawe problemy do rozwiązania. Jednym słowem: kształcą informatyczny narybek. Zarówno z myślą o rozwoju swojej sekcji, jak i z szerszymi planami. Bo chociaż trochę odstawiamy od świata w zakresie sprzętu — zapewnia Siergiej — to przecież nikt nie może nam przeszkodzić być najlepszymi w programowaniu!

Siergiej zapewnił też, że „agatowcy” chętnie nawiążą kontakt z polskimi klubami komputerowymi. Podaję więc ich adres: **Klub Komputerowy „Agat”. 127 411 MOSKWA, Dmitrowskoje Szosse 115.**

Waldemar Siwiński

## KOMPUTERY W SZKOŁACH USA

W ubiegłym roku szkolnym 94,9 proc. amerykańskich szkół w większym lub mniejszym stopniu korzystało z komputerów w procesie nauczania.

Dla porównania: siedem lat temu komputerami dysponowało zaledwie 16,5 proc. tamtejszych szkół. Obecnie w szkołach USA przypada średnio jeden komputer na 32 uczniów. Entuzjaści komputerowego nauczania oświadczają jednakże, iż jest to za mało. Uważają, że komputer powinien stać się centralnym elementem procesu nauczania.

„Jesteśmy obecnie zadowoleni, gdy w szkole znajduje się jedna sala — a w niej 24 komputery — mówi John Martin, specjalista w dziedzinie tworzenia komputerowych programów edukacyjnych. — Ale, tak naprawdę, gdy mamy do czynienia tylko z jedną skomputeryzowaną klasą, to po staremu znajdujemy się na prymitywnym poziomie”.

„Nowa technika powinna stać się w szkołach takim samym zwykłym zjawiskiem, jak ołówki” — stwierdził w wywiadzie udzielonym gazecie „USA Today” Garri Peterson, kierownik wydziału oświaty

w okręgu Baldrige, gdzie komputery stoją już na ławce każdego ucznia.

Oczywiście, niezależnie od robiących wrażenie średnich cyfr, sytuacja nie jest wcale we wszystkich szkołach jednakowa. Płatne szkoły prywatne, przykładowo, znacznie wyprzedzają pod względem nasycenia komputerami zwykłe placówki oświatowe. Większość szkół, jak wynika z wniosków raportu opublikowanego jesienią ub. roku przez Kongres USA, odczuwa niedobór nowej techniki.

„Eksplozja” komputeryzacji szkół USA w tym dziesięcioleciu stała się możliwa dzięki pojawieniu się stosunkowo niedrogich modeli komputerów osobistych, a także opracowaniu specjalnych komputerowych programów edukacyjnych dla szkół i przedszkoli. Obecnie istnieje w USA 900 firm, przeważnie małych, przygotowujących programy edukacyjne. 800 z nich powstało w ciągu ostatnich trzech lat.

(ws)



## W SYSTEMIE CP/M DLA CPC I PCW

Przedstawiony poniżej program pozwala na sformatowanie dyskietki na 210 KB. Ustawia także odpowiednie parametry stacji dysków, aby komputer wiedział, że pracuje z niestandardową dyskietką. Program ustawia parametry dla aktualnie obsługiwanej stacji, ale formatuje tylko stację A. Program korzysta

bezpośrednio z procedur w języku maszynowym, dlatego skutki niepoprawnego wprowadzenia programu mogą być nieobliczalne. Aby uchronić się przed nieopatrznie „wstukanym” błędem proponuję trochę poeksperymentować na czystej dyskietce, a przy przepisywaniu szczególną uwagę zwrócić na stałą alfaXDPB i zawartość instrukcji inline(...). Podstawowym tekstem jest wypełnienie całej dyskietki danymi tekstowymi, a następnie odczytywanie i sprawdzanie poprawności poszczególnych plików np. za pomocą programu COMPARE.COM.

*Marcin Skóra*

```

($
short formater 1.1
SK0600T SOFTWARE 89
18.02.89
$)

program formater;
(#####)
($ program pobiera opcję z commandline lub z klawiatury i w zależności $)
($ od niej formatuje dyskietkę lub ustawia parametry stacji dysków $)
(#####)

type buffer = array(1..512.) of byte;
   XDPBtype = array(0..26.) of byte;

var error      : byte;
    ok         : boolean;
    buf        : buffer;
    header     : array(1..16,0..3.) of byte;
    XDPB       : XDPBtype;

procedure DD_Format(track,filler:byte);
(#####)
($ procedura formatuje podaną ścieżkę wypełniając ją zawartością zmiennej $)
($ filler korzysta przy tym z danych umieszczonych w zmiennych track i $)
($ header; po pomyślnym sformatowaniu ścieżki zmienna ok przybiera wartość $)
($ TRUE w przeciwnym razie ok=FALSE $)
(#####)

begin
inline($OE/$1F/$CD/$05/$00/$E5/$DD/$E1/$3A/track/
      $57/$3A/filler/$5F/$21/header/$06/$01/
      $0E/$00/$CD/$5A/$FC/$BF/$00/$32/error/
      $3E/$00/$32/ok/$30/$05/$3E/$FF/$32/ok);
end;

procedure GetXDPB;
(#####)
($ procedura pobiera parametry stacji dysków i umieszcza je w zmiennej $)
($ XDPB $)
(#####)
begin
inline($OE/$1F/$CD/$05/$00/$11/XDPB/$01/$1B/$00/$ED/$B0);
end;

procedure SetXDPB;
(#####)
($ procedura ustawia parametry stacji dysków korzystając z danych w zmiennej $)
($ XDPB $)
(#####)

```

Trzycalowe dyskietki stosowane w Amstradach serii CPC i oraz PCW poza wieloma zaletami posiadają także wady. Jedną z nich jest wysoka cena w stosunku do niezbyt dużej pojemności w standardowych formatach. Na cenę nie mamy niestety wpływu, co się jednak tyczy sformatowania dyskietki, okazuje się, że bez większych problemów możemy „zarobić” dodatkowe 30 KB formatując dyskietkę na 210 KB (z czego 2 KB zajmuje katalog). Poniżej przedstawiamy dwa sposoby realizacji tego pomysłu.

```

begin
inline($OE/$1F/$CD/$05/$00/$54/$5D/$21/XDPB/$01/$1B/$00/$ED/$B0);
end;

(#####)
const copyright : string(32.)='Copyright by SK0600T SOFTWARE 89';
type optiontyp  = 1..2;
const AlfaXDPB  : XDPBtype=(40, 0,3, 7, 0,209, 0,63, 0,192,
                           0,16,0, 0, 0, 2, 3, 0,42, 10,
                           16, 0,2,22,42, 96,255);

var option      : byte;
    cl_option_ok : boolean;
    lastXDPB    : XDPBtype;
    ch           : char;

function Key:char;
(#####)
($ funkcja podaje literę odpowiadającą klawiszowi, który został naciśnięty $)
($ litera zostaje zamieniona na dużą jeśli była mała $)
(#####)
var ch:char;
begin
read(KBD,ch);
Key:=upcase(ch);
end;

function GetComLine:optiontyp;
(#####)
($ funkcja podająca numer opcji (format lub ustaw) wprowadzonej po nazwie $)
($ programu $)
(#####)
begin ($ GetComLine $)
cl_option_ok:=TRUE;
if paramstr(1)='FORMAT' then GetComLine:=1
    else
    if paramstr(1)='USTAW' then GetComLine:=2
        else begin
            writeln(paramstr(1),'nieznana opcja',chr(7));
            cl_option_ok:=FALSE;
        end;
end; ($ GetComLine $)

function GetKeyParam:optiontyp;
(#####)
($ funkcja podająca numer opcji wybranej z klawiatury przez obsługującego $)
(#####)
begin ($ GetKeyParam $)
writeln;
writeln('1 - formatowanie');
writeln('2 - ustawianie');
writeln('3 - koniec pracy');
writeln;
case Key of
    '1':GetKeyParam:=1;
    '2':GetKeyParam:=2;
    '3':GetKeyParam:=3;
    else writeln(chr(7),'Nie ma takiej opcji!!!');
    end;
end; ($ GetKeyParam $)

procedure Format;
(#####)
($ procedura przygotowuje formatowanie dyskietki a następnie formatuje $)
($ 41 ścieżek $)
(#####)
var no_track:byte;
    procedure FormatTrack(track:byte);
(#####)
($ procedura przygotowująca parametry i formatująca jedną ścieżkę $)
(#####)

```



# 210 KB

```
var qu_sect,no_sect,no_f_sect:byte;
begin ($ format track $)
  GetXDPB;
  qu_sect:=XDPB(.19.);
  no_f_sect:=XDPB(.20.);
  for no_sect:=1 to qu_sect do
    begin
      header(.no_sect,0.):=track;
      header(.no_sect,1.):=0;
      header(.no_sect,2.):=no_f_sect+no_sect-1;
      header(.no_sect,3.):=XDPB(.22.);
    end;
  DD_Format(track,229);
  if ok then write(' O.K. ');
  else write(' Błąd: ',error);
end;($ format track $)

begin ($ format $)
  GetXDPB;          ($ zapamiętanie przez program $)
  lastXDPB:=XDPB;  ($ parametrów stacji dysków $)

  repeat
    writeln('Przygotuj dyskietkę do formatowania i wciśnij "F"');
    writeln;
    ch:=Key;
  until ch='F';

  XDPB:=AlfaXDPB;
  SetXDPB;
  for no_track:=0 to 41 do
    begin
      write(chr(13),no_track);
      FormatTrack(no_track);
    end;
  writeln;
  XDPB:=lastXDPB;  ($ przywrócenie parametrów stacji dysków $)
  SetXDPB;         ($ sprzed działania procedury formatującej $)
end; ($ format $)

procedure Ustaw;
(#####)
($ procedura ustawia parametry stacji dysków na pracę w zwiększonej pojemn. $)
(#####)
begin
  XDPB:=AlfaXDPB;
  SetXDPB;
  writeln('Format ALFA ustawiony. ');
end; ($ ustaw $)

begin ($ formater $)

  writeln(' $$$ Formater v 1.1 $$$ ');
  writeln(copyright);
  writeln;

  if paramcount>0 then begin
    option:=GetComLine;
    if not cl_option_ok then option:=GetKeyParam;
    end
    else option:=GetKeyParam;

  repeat
    case option of
      1:Format;
      2:Ustaw;
      3:begin writeln('Dziękuję'); halt; end;
    end;
    option:=GetKeyParam;
  until FALSE
end.
```

## W CP/M I AMSDOSIE DLA CPC

Poniższy pakiet programów przeznaczony jest wyłącznie dla komputerów serii CPC. Ma jednak tę zaletę, że umożliwia wykorzystywanie dyskietek sformatowanych na 208 KB także w systemie Amsdos. Już na pierwszy rzut oka można zauważyć, że zdecydowano się w tym przypadku na całkowicie inną filozofię rozwiązania problemu. Zamiast jednego programu wykonującego wszystkie funkcje zastosowano dla każdej funkcji osobny niewielki program.

Do formatowania służy BASIC-owy program KUNYFORM.BAS (wydruk 1). Programy O.BAS (wydruk 2) i O.PAS (wydruk 3) wymuszają współpracę systemów AMSDOS i CP/M z tak sformatowaną dyskietką w stacji A lub B, albo też w obu stacjach (opcja R).

Niestety przy użyciu tego formatu nie można korzystać z normalnych formatów Amstrada. W celu powrotu do nich można wykorzystać programy RET.BAS (w Amsdosie) i RET.PAS (w CP/M).

Wszystkie programy z rozszerzeniami "PAS" należy oczywiście skompilować przy pomocy kompilatora TURBO PASCAL.

I jeszcze kilka uwag praktycznych. Formatowanie dyskietki programem KUNYFORM. można przerwać przy pomocy klawisza ESC, co pozwala — w razie pomyłki — uratować choć część zapisanych zbiorów.

Z pewnością staniecie przed problemem kopiowania zbiorów z dyskietki sformatowanej standardowo na dyskietkę w nowym formacie. W tym celu wykonajcie następujące czynności:

- \* Inicjujemy system CP/ M (lcpm)
- \* Uruchamiamy program O.COM i wybieramy opcję B
- \* Uruchamiamy program pip z dyskietki systemowej
- \* piszemy A:=B:\*(C)
- \* Wkładamy dyskietkę źródłową, naciskamy ENTER i kopujemy kolejne zbiory.

Jacek Kunowski

```
1 * #####
2 * $
3 * $ Program formatujacy dysk $
4 * $
5 * $ 208 KB $
6 * $
7 * $ AMSTRAD CPC 6128 $
8 * $
9 * $ autor J. KUNOWSKI $
10 * $
11 * $ Warszawa 1989 $
12 * $
13 * #####
14 *

100 MEMORY &BFFF:MODE 1
110 DATA &21,&1d,&90,&cd,&d4,&bc,&22,&1e

120 DATA &90,&79,&32,&20,&90,&1e,&00,&1b
130 DATA &00,&0e,&00,&21,&30,&90,&df,&1e

140 DATA &90,&32,&21,&90,&c9,&a0,&a5,&a1
150 DATA &a6,&a2,&a7,&a3,&a8,&a4,&a9,&28

160 DATA &00,&03,&07,&00,&d1,&00,&3f,&00
170 DATA &c0,&00,&10,&00,&00,&00,&a0,&0a

180 DATA &20,&2e,&e5,&02,&04,&01,&ff,&ff
185 PRINT "FORMATOWANIE - 208 KB":PRINT

190 FOR i=&9000 TO &9000+28
200 READ zm:POKE i,zm:NEXT
210 POKE &9010,&36
220 low=PEEK(&BE40):high=PEEK(&BE41)
230 adpn=high*256+low+10
240 low=PEEK(adpn):high=PEEK(adpn+1)
250 adpb=high*256+low
260 FOR i=&9030 TO &9030+39 STEP 4
270 POKE i+1,0:POKE i+3,2
280 READ nr:POKE i+2,nr:NEXT
290 a$="":PRINT"Stacja (A/B)?";
300 WHILE NOT(a$="A" OR a$="B")
310 a$=INKEY$:a$=UPPER$(a$)
320 IF a$="A" THEN GOSUB 420
330 IF a$="B" THEN GOSUB 470

340 WEND
350 PRINT":Formatować (T/N)?";
360 WHILE NOT(a$="N" OR a$="T")
370 a$=INKEY$:a$=UPPER$(a$)
380 IF a$="N" THEN END
390 IF a$="T" THEN GOSUB 510

400 WEND
410 PRINT:PRINT:END
420 POKE &900E,0:PRINT"A":PRINT
430 FOR I=adpb TO adpb+24
440 READ zm:POKE i,zm

450 NEXT
460 RETURN
470 POKE &900E,1:PRINT"B":PRINT
480 FOR i=adpb+64 TO adpb+87
490 READ zm:POKE i,zm:NEXT

500 RETURN
510 PRINT a$:PRINT
520 FOR i=41 TO 0 STEP -1
530 FOR j=&9030 TO &9030+39 STEP 4

540 POKE j,1:NEXT j
550 POKE &9010,1
560 CALL &9000

570 PRINT USING("###");i;:PRINT" - ";
580 ER=PEEK(&9021)

590 IF ER=0 THEN PRINT"D.K.";ELSE 630
600 NEXT i
610 PRINT:PRINT:PRINT"#FORMATOWANE#"

620 RETURN
630 PRINT
640 PRINT SPC(11)"A W A R I A !!!"
650 BORDER 27
660 PRINT CHR$(7);:GOTO 660
```



## O.BAS

```

1 '
2 'System AMSDOS
3 'Zmiana dpb - format 208KB
4 '
10 DATA 40,0,3,7,0,209,0,63,0
20 DATA &c0,0,16,0,0,0,&a0,10
30 DATA 32,45,&e5,2,4,0,0,&ff
40 low=PEEK(&BE40):high=PEEK(&BE41)
50 adph=high*256+low+10:MODE 1
60 low=PEEK(adph):high=PEEK(adph+1)
70 adpb=high*256+low
80 PRINT"Zmiana formatu dysku - 208KB"
90 PRINT:A$="":PRINT"Stacja (A/B/R)?"
100 WHILE NOT(a$="A"OR a$="B"OR a$="R")
110 a$=INKEY$:a$=UPPER$(a$)
120 IF a$="A" THEN GOSUB 200
130 IF a$="B" THEN GOSUB 230
140 IF a$(">"R" THEN 160
150 GOSUB 200:RESTORE:GOSUB 230
160 WEND:PRINT
170 PRINT"Format 208KB - stacja ";
180 IF a$="A"OR a$="B"THEN PRINT a$
ELSE PRINT"A i B"
190 PRINT:END
200 FOR i=adpb TO adpb+24
210 READ zm:POKE i,zm:NEXT
220 RETURN
230 FOR i=adpb+64 TO adpb+88
240 READ zm:POKE i,zm:NEXT
250 RETURN

```

## O.PAS

```

program kuny;
var
  l,adr,z,s,zm : integer;
  stacja : char;
  mark,mark2 : boolean;
const
  dpb : array[0..26h] of integer =
    (40,0,3,7,0,209,0,63,0,192,0,
     16,0,0,0,2,3,0,42,10,$A0,0,
     2,32,45,$60,255);
begin
  clrscr;gotoxy(1,1);
  z:=bdos(25);
  writeln('Zmiana formatu - 208KB');
  writeln(' ');
  adr:=bdoshl(31);
  write('Stacja (A/B/R)');
  stacja:= ' ';
  write('Format 208KB - stacja ');
  repeat
    mark2:=true;
    readln(stacja);
    case stacja of
      'a','A' : adr:=adr-27*z;
      'b','B' : adr:=adr+27*abs(z-1);
      'r','R' : begin
        adr:=adr-27*z;
        mark:=true;
        end;
      else mark2:=false;
    end;
  until mark2;
  repeat
    mark:=not(mark);
    zm:=0;
    while zm<27 do
      begin
        memzadr+zm:=opbzmz;
        zm:=zm+1;
      end;
    if mark=false then adr:=adr+26;
  until mark;

```

```

write(' ');
write('Format 208KB - stacja ');
case stacja of
  'a','A' : write('A');
  'b','B' : write('B');
  'r','R' : write('A i B');
end;
write(' ');
end.

```

## RET.BAS

```

1 'program wiadca,ady wybor formatu
2 '
10 low=PEEK(&BE40):high=PEEK(&BE41)
20 adph=high*256+low+10
30 low=PEEK(adph):high=PEEK(adph+1)
40 adpb=high*256+low:MODE 1
50 PRINT"Zmiana formatu dysku":PRINT
60 A$="":PRINT"Stacja (A/B/R)?"
70 WHILE NOT(a$="A"OR a$="B"OR a$="R")
80 a$=INKEY$:a$=UPPER$(a$)
90 IF a$="A" THEN POKE adpb+24,0
100 IF a$="B" THEN POKE adpb+88,0
110 IF a$(">"R" THEN 130
120 POKE adpb+24,0:POKE adpb+88,0
130 WEND:PRINT
140 PRINT"Format normalny - stacja ";
150 IF a$="A"OR a$="B"THEN PRINT a$
ELSE PRINT"A i B"
160 PRINT:END

```

## RET.PAS

```

program ret;
var
  adr,z,s,zm : integer;
  stacja : char;
  mark : boolean;
begin
  clrscr;gotoxy(1,1);
  writeln('Zmiana formatu ');
  writeln(' ');
  z:=bdos(25);
  adr:=bdoshl(31);
  write('Stacja (A/B/R)');
  stacja:= ' ';
  repeat
    mark:=true;
    readln(stacja );
    case stacja of
      'a','A' : begin
        adr:=adr-27*z;
        memzadr+26h:=0;
        end;
      'b','B' : begin
        adr:=adr+27*abs(z-1);
        memzadr+26h:=0;
        end;
      'r','R' : begin
        adr:=adr-27*z;
        memzadr+26h:=0;
        memzadr+53h:=0;
        end;
      else mark:=false;
    end;
  until mark;
  writeln(' ');
  write('Normalny format - stacja ');
  case stacja of
    'a','A' : write('A');
    'b','B' : write('B');
    'r','R' : write('A i B');
  end;
end.

```

# Drogi Bajtku!

**Na listy czytelników odpowiada Dominik Falkowski, student IV roku Politechniki Warszawskiej:**

Mam 13 lat, od blisko roku posiadam mikrokomputer Timex 2048.

Przed paroma dniami nabyłem w księgarni zestaw programów firmy Coda. Przy oglądaniu przyszłego nabytku uderzyło mnie podobieństwo dwóch tytułów: *The Way of The Exploding Fist*, którą już posiadałem w swoich zbiorach i *Droga Wybuchowej Pięści* nagranej na kasecie. Mimo to jednak opis oraz hasła typu: „Pamiętaj!!! programy firmy Coda są najlepszą rozrywką” skusiły mnie. Ponadto na kasecie było wyraźnie napisane, Producent: Coda.

Po przyjeździe do domu okazało się, że miałem rację. Gry okazały się bliskimi wersjami gier „The Way...” oraz „Fairlight” W grach wszystko zgodne było z pierwowzorem, jedynie napisy były polskie i widniał na nich napis Coda i znak firmowy.

Obecnie traktuję dwie pierwsze gry, jako polskie wersje gier już posiadanych. Żałuję tylko, że zamiast tej kasety mogłem kupić zestaw Polbriku — 8 gier z oryginalnymi angielskimi tytułami.

Myszę, że należałoby przestrzec innych posiadaczy tego komputera przed kupnem tych zestawów. Trzeba najpierw dokładnie sprawdzić, czy nie posiada się już tych gier w swojej kolekcji.

**Slawomir Mróz  
Bydgoszcz**

Jak jest Ci pewnie wiadomo, w naszym kraju wciąż nie ma przepisów regulujących handel programami. To, co skutecznia (!) Polbriku oraz wiele innych firm (w szczególności wzmiankowana CODA), w innych krajach jest ciężkim przestępstwem. Dlatego zachowanie pseudohandlarzy, np. giełdowych lub w kramikach z programami uważam za co najmniej żenujące. Tworzą oni cenniki, biją się i handlują programami, które po prostu ukradli (tak, gdyż nie napisali ich sami, a zezwolenia od producenta nie mają). Apeluje więc o zbratanie się komputerowców w celu wymiany programów,

gdyż odpłatne ich „odstępowanie” jest nonsensem. Jest bardzo wiele osób, dla których jedynym źródłem oprogramowania jest kupno — na giełdzie lub w kramiku. Nie powinno się ich oszukiwać!

**Czy magnetofon MK 232P nadaje się do współpracy z Commodore C16/116.**

**Czy myszka i piórko świetlne mogą być przyłączone do tych komputerów.**

**nazwisko i adres  
do wiadomości redakcji**

Tak jak wszystkie komputery firmy Commodora tak i C16/116 wymagają do współpracy specjalnych, firmowych magnetofonów. Są oczywiście elektrony, którzy we własnym zakresie potrafią dokonać odpowiednich przeróbek, nie jest to najprostsza sprawa nawet dla specjalisty.

W systemie operacyjnym tych komputerów nie przewidziano możliwości obsługi ani myszy ani pióra świetlnego czy też modemu. Nie ma więc odpowiedniej obsługi portów. Nie istnieje User port niezbędny np. do podłączenia do niego modemu. Istnieje wszakże na rynku pióro świetlne specjalnie przystosowane do współpracy z C116.

**Do napisania tego listu skłoniły mnie dwie sprawy: Po pierwsze chciałbym się dowiedzieć w jakiej cenie są ogłoszenia w Indywidualnym Banku Danych oraz w rubryce SOS. Po drugie chcę napisać kilka słów na temat firmy Pana Andrzeja Zakutynskiego „Progres” działającej w Koszalinie. Otóż po wcześniejszym listowym umówieniu się z wyżej wymienioną firmą w dniu 9 stycznia 1989 roku wysłałem kasetę razem z 12 programami na wymianę. Gdy po upływie 3 tygodni od wysłania, paczka nie przychodziła wysłałem ekspresem list z prośbą o szybki zwrot kasety. Odpowiedzi jednak na ten list nie otrzymałem. Gdy piszę ten list jest 20 lutego 1989 r., więc minął już dawno miesiąc, a ja wciąż nie otrzymuję paczki. W katalogu gier firmy 6 punkt brzmi następująco: „Termin realizacji zamówień tylko 12 godzin!!!” Wydaje mi się, że w tym punkcie mocno przesadzono. Niech list ten stanie się przestroga dla tych wszystkich, którzy zamierzają skorzystać z usług „Progresu”.**

**Andrzej Józewczuk  
Lublin**

Ogłoszenia zamieszczane we wspomnianych rubrykach są całkowicie bezpłatne. Są to miejsca w Bajtku przeznaczone dla tych, którzy pomagają nam redagować pismo, a więc dla Was wszystkich. W drugiej sprawie osądzać nie będziemy. Zamieszczamy ten list jako ostrzeżenie Czytelnika dla Czytelników.

**Jaki telewizor może współpracować z komputerami Commodore C116 i PLUS/4, a także z ATARI 65XE? Jaki joystick należy kupić do PLUS/4?**

**nazwisko i adres  
do wiadomości redakcji**

Wszystkie komputery wyposażone w modulator TV przystosowane są do współpracy z telewizorami posiadającymi wejścia antenowe 750. Nawet gdy wartość ta jest inna (np. 2400 dla starych niemieckich telewizorów) to potrzebny jest jedynie odpowiedni symetryzator dopasowujący impedancję falową. Jest to jednak przypadek niesłychanie rzadki, gdyż większość telewizorów posiada już wspomniane wejścia 750. Od czasu do czasu można na różnego rodzaju wyprzedazach nabyć specjalne joysticki do PLUS/4. Najwygodniej jest jednak kupić dowolny joystick np. Quickshot z 9-bolcowym gniazdem i zastosować złącza pośrednie. Takie złącza występowały kiedyś wraz z pakietem oprogramowania (3 kasety) przeznaczonego dla C116.

**Słyszałem, że istnieje możliwość kupna dekodera, dzięki któremu na PLUS/4 mógłbym skorzystać z programów pisanych dla C-64.**

**Hubert Lubecki  
Olecko**



# PAKIET ZINTEGROWANY

**Do widoku komputera stojącego w biurze lub w domu przyzwyczailiśmy się do tego stopnia, że traktujemy go podobnie jak każdy inny przedmiot codziennego użytku. Nie zajmuje on honorowego miejsca i nie jest otaczany szczególną czcią. Nikt też nie zastanawia się, jakie ogromne zmiany zaszły w wielu instytucjach od chwili pojawienia się komputera.**

## ● EDYTOR ZAMIAST MASZINY DO PISANIA

Jeszcze niedawno większość dokumentów powstających w urzędach była pisana na zwykłych maszynach do pisania. Ma to jeden istotny mankament. Wykreślenie jakiegoś zdania lub poprawka stylistyczna zmuszają do przepisania całego tekstu. Wymyślono więc program tzw. **edytor tekstu**, który znakomicie ułatwia pisaćemu jego pracę: tekst można wielokrotnie poprawiać, kasować i przenosić jego fragmenty, zmieniać odstęp między wierszami i szerokość strony. Mądry edytor potrafi nawet sam przenosić wyrazy i poprawić błędy ortograficzne. Wszystko to odbywa się na ekranie monitora, bez zmarnowania nawet jednego arkusza papieru!

W niektórych biurach edytory tak bardzo przypadły pracownikom do gustu, że począwszy maszynę do pisania oddano do lamusa. Najpopularniejsze edytory, które zaważowały cały świat, to WordStar, Tasword i ChWriter.

## ● ELEKTRONICZNE ARCHIWUM

Komputer wspiera człowieka nie tylko w pisaniu tekstów. Niegdyś półki archiwów aż uginęły się od tysięcy zakurzonych skoroszytów. Odszukanie czegośkolwiek w takich stertach papieru było niezwykle trudne. Dziś rolę archiwisty przejął program zwany **bazą danych**. Odpowiednikiem teczki z aktami w takim elektronicznym archiwum jest dyskietka. Na dyskietce można zapisać dowolne informacje, na przykład imię, nazwisko, adres, numer telefonu i kod pocztowy znajomych. Taka porcja danych o jednej osobie tworzy **rekord**. Każda informacja zajmuje zawsze określone miejsce w rekordzie zwane **polem**. Kilkadziesiąt rekordów zawierających dane o różnych osobach to **plik**. System zarządzający bazą danych może błyskawicznie posortować rekordy pliku według danego klucza, np. alfabetycznie, lub natychmiast wyszukać wszystkie osoby zamieszkałe w Warszawie.

Oczywiście, bazy danych wykorzystuje się w praktyce do rozwiązywania o wiele poważniejszych zagadnień. Przy ich pomocy prowadzi się ewidencje ludności. Katalogowanie zbiorów biblioteczných, ewidencję obrotów w przedsiębiorstwach, identyfikuje się przestępców. Oprócz sortowania i wyszukiwania informacji, mogą one przetwarzać zmagazynowane informacje, wykonywać obliczenia, prezentować graficznie uzyskane wyniki. Najbardziej znanym programem do tworzenia baz danych jest dBase.

## ● KONIEC UDRĘKI Z ZESTAWIENIAMI

Nie tylko konieczność wertowania stosów skoroszytów w poszukiwaniu jakiejś informacji spędza sen z oczu urzędnikom. Równie dokuczliwe jest sporządzanie nierzadziej tabel i zestawień, w

których trzeba sumować, przemnażać i obliczać średnie z setek liczb.

Programiści opracowali więc program, który służy wyłącznie do sporządzania obliczeń tabelarycznych. **Spreadsheet**, czyli arkusz kalkulacyjny do złudzenia przypomina zwykłe formularze stosowane w biurach do przygotowania nierzadziej zestawień. Ekran podzielony jest na pola pogrupowane w ponumerowanych rzędach i kolumnach, tak jak w zwykłej tabelce. Do każdej z takich komórek można wpisać tekst, liczbę lub formułę matematyczną (np. sumowanie wszystkich liczb w kolumnie). Po zmianie liczby w dowolnej komórce natychmiast przeliczane są wszystkie formuły i zawsze mamy tabelę z aktualnymi wynikami. Arkusz kalkulacyjny może przedstawić najistotniejsze dane z każdej tabelki w postaci przejrzystych wykresów lub diagramów.

## ● LEKTURA NA PARĘ WIECZORÓW

Tak więc widzimy, że informatyka stworzyła znakomite narzędzia, pozwalające unowocześnić nasz warsztat pracy. Nie jest to jednak takie proste. Decydując się na wprowadzenie elektroniki do biura, stajemy przed niebagatelnym problemem. Musimy poznać co najmniej trzy programy: edytor tekstów, bazę danych i arkusz kalkulacyjny (spreadsheet). Nie jest to wcale łatwe. Instrukcja do każdego z tych programów może składać się nawet z kilkuset stron. Samo przeczytanie opisów jest lekturą na parę wieczorów.

Niestety, to nie koniec trudności. Każdy z programów będzie miał inny system komunikacji z użytkownikiem, inaczej opracowane menu. Te same funkcje będą realizowane po naciśnięciu innej kombinacji klawiszy. W sumie nauczenie się sprawniej obsługi trzech programów może okazać się równie trudne, jak opanowanie podstaw języka obcego.

Załóżmy, że pokonaliśmy te wszystkie przeszkody. Umieemy już napisać pismo przy pomocy edytora tekstu, baza danych pomaga w prowadzeniu podręcznego banku danych, rachunki, dzięki arkuszowi kalkulacyjnemu, nie sprawiają już żadnych kłopotów. Któregoś dnia okazało się, że musimy napisać 40 identycznych listów i wysłać je do 40 różnych osób. Natychmiast wpadliśmy na pomysł, żeby wykorzystać adresy zapisane w bazie danych i umieścić je w tekście napisanym przy pomocy edytora. Nasza radość była, niestety, przedwczesna. Okazało się, że nasze programy pochodzą z różnych firm, które nie przewidziały możliwości wymiany danych między nimi. Musimy więc adresy i nazwiska dopisać ręcznie.

## ● PAKIET ZINTEGROWANY

Twórcy oprogramowania zdawali sobie od dawna sprawę z kłopotów, jakie może przysporzyć użytkownikowi korzystanie z oddzielnie opracowanych programów. Brak możliwości przesyłania danych z jednego programu do drugiego, stosowanie różnych formatów zapisu na dys-

kach, nieznormalizowane systemy komend skutecznie utrudniają użytkownikom pełne wykorzystanie zalet komputera. Istnieją oczywiście programy narzędziowe, które pozwalają przenosić zbiory np. z programu kalkulacyjnego do edytora tekstu lub dopasowują format zapisu. Nie jest to jednak sposób satysfakcjonujący przeciętnego użytkownika.

Wyjście z sytuacji okazało się na szczęście bardzo proste. Programiści po prostu połączyli bazę danych, edytor tekstu i spreadsheet w jeden program. Tak właśnie powstał **pakiet zintegrowany**. Składa się on z trzech podstawowych bloków. Użytkownik przystępujący do pracy z pakietem zintegrowanym styka się z jednolitym systemem komend. Podstawowe funkcje są wspólne dla wszystkich bloków, dzięki czemu użytkownik bardzo łatwo uczy się obsługi programu. Najważniejszą zaletą pakietu jest możliwość przesłania danych pomiędzy poszczególnymi modułami. Można na przykład bardzo szybko zaadresować korespondencję adresami z bazy danych, tabelkę opracowaną w programie kalkulacyjnym dołączyć do tekstu napisanego w edytorze.

Współczesne pakiety zintegrowane zawierają również inne bloki. Jednym z nich jest tak zwana grafika ekonomiczna (ang. business graphics, niem. Geschäftsgrafik). Procedury graficzne umożliwiają utworzenie wykresów prezentujących dane przetwarzane w programie kalkulacyjnym lub bazie danych. Przejrzyste diagramy kołowe lub słupkowe pomagają szybko zorientować się w ogólnej tendencji danego procesu lub wychwycić proporcje pomiędzy kilkoma parametrami. Kolejnym programem często umieszczanym w pakiecie zintegrowanym jest program komunikacyjny. Przydaje się użytkownikom komputerów podłączonych do sieci komputerowej. Odpowiednie procedury ustalają wymagane parametry transmisji danych, dbają o uzyskanie połączenia z odbiorcą. Korzystając z modemu i właściwie dobranego programu komunikacyjnego możemy wysłać informację poprzez linie telekomunikacyjne nawet na inny kontynent. Dzięki temu nasz komputer staje się prawdziwym oknem na świat. Na naszym biurku stoi urządzenie, które zastępuje teleks, umożliwia uzyskanie danych o aktualnych notowaniach giełdowych, najnowszego serwisu prasowego i programów.

W pakiecie zintegrowanym umieszczony jest czasem Time Manager. Jest to po prostu elektroniczny kalendarz, który ułatwia właściwe gospodarowanie czasem i pamięcią o umówionych spotkaniach. Profesjonalne pakiety zintegrowane zawierają własne języki programowania. Umożliwiają to pisanie skomplikowanych programów sortujących, przeszukujących i dokonujących złożonych obliczeń matematycznych na podstawie zgromadzonych danych.

## ● PAKIET TYLKO NA IBM?

Nikogo nie powinno dziwić, że program dysponujący tak ogromnymi możliwościami zajmuje dużo miejsca i wymaga szybkiego komputera. Najpopularniejsze pakiety zintegrowane Lotus 1-2-3, Symphony, Framework opracowane dla komputerów IBM. Mieszczą się one na kilku dyskietkach i wymagają co najmniej kilkuset kilobajtów pamięci operacyjnej. Praca z nimi staje się naprawdę komfortowa dopiero przy zastosowaniu twardego dysku. Nie znaczy to, że całkowicie zapomniano o użytkownikach komputerów 8 bitowych. Na Atari XL/XE powstał całkiem przyzwoity pakiet, który nosi nazwę Mini Office II. Zawiera on bazę danych, edytor tekstu, spreadsheet, blok procedur grafiki ekonomicznej i program komunikacyjny do pracy w kilku popularnych sieciach.

Janusz Jarmoch

Dekoder taki nie istnieje. Powód jest następujący. W komputerach C16/116 jak i PLUS/4 funkcje kontrolera dźwięku i obrazu spełnia jeden jedyny układ, znany pod nazwą TED. W komputerze C-64 funkcje te są jak wiadomo rozdzielone pomiędzy dwa wysoko wyspecjalizowane układy VIC (zajmujący się grafiką) i SID (zajmujący się dźwiękiem). Ponadto podczas pracy C+4 intensywnie bankuje pamięć, podczas gdy w C-64 mechanizm ten jest stosowany (przynajmniej) teoretycznie.

**Chciałbym zasięgnąć rady na temat rozbudowy pamięci w komputerze Atari 130XE. Jaka jest maksymalna granica takiego rozszerzenia i gdzie można tego dokonać?**

Michał Speier  
Toruń.

Możliwe jest rozszerzenie pamięci we wszystkich modelach Atari XL/XE. Różny jest tylko stopień skomplikowania takiej operacji. W Atari 65XE i 130XE jest to trudniejsze niż w 800XL z powodu zastosowania „ulepszonoego” układu zarządzania pamięcią. Trudno jest mówić o maksymalnej wielkości takiego rozszerzenia, ponieważ jest ona płynna. Kiedyś sukcesem było uzyskanie 64 KB RAM, a obecnie w Stanach Zjednoczonych wykonywane są rozszerzenia do 1 MB. W Polsce rozszerzenia do 256 KB lub 192 KB wykonują liczne firmy reklamujące się w „Bajtku”.

**Zdecydowałem się napisać do Was ponieważ mam kłopot z wyborem komputera. Dotychczas miałem do czynienia najwięcej z komputerami ZX Spectrum, więc chciałbym pozostać przy tym typie. Teraz gdy pokazały się na rynku polskim Spectrumy 128+2 i +3 mam zamiar wybrać jeden z nich. Niestety oprócz tego, że mają 128K pamięci i stację dysków, bądź magnetofon, nic o nich nie wiem. Publikacji na temat tych komputerów jest bardzo mało i dlatego chcę skorzystać z waszej pomocy.**

1. Czy wymienione komputery są kompatybilne z wersją + i wcześniejszym ZX Spectrum
2. Jaka jest rozdzielczość grafiki i jaką dysponują paletą barw?
3. Jakie są różnice między Basic-iem tych komputerów a np. Spectrum+, w szczególności chodzi mi o instrukcję BEEP.
4. Czy nadal jest tylko jeden port manipulatora?
5. Jeżeli nie są kompatybilne z rodziną ZX to jak wygląda sprawa z oprogramowaniem na te typy na polskim rynku.

Krzysztof Nocuń  
Dąbrowa Górnicza

O problemach związanych z wyborem komputera pisaliśmy już w „Bajtku” niejednokrotnie. Jeżeli decydujesz się na zakup jednego z licznych „Spectrum”, musisz pamiętać, iż w jego przypadku stosowanie stacji dysków jest ekstrawagancją. Spectrum nie ma bowiem żadnego systemu operacyjnego, musi więc zostać on podłączony w postaci interface. Także struktura zapisu plików na dyskietki oraz niewielka ich pojemność sprawiają, że stacja do Spectrum jest raczej kosztowną zabawką. Istnieją oczywiście przypadki, że staje się niemal niezbędną, lecz to sporadycznie.

Co do różnic między modelami Spectrum 128 a wcześniejszym Spectrum 48, to jest ona niewielka. Wszystkie Spectrum 128 mogą pracować w trybie 48 i działają wtedy wszystkie programy dla 48. Oprócz tego dostępny jest tzw. Basic 128, rozbudowany w stosunku do Basica 128 o dwie komendy i pracujący z ciekawym rozwiązaniem edytorem ekranowym. Rozdzielczość pozostaje bez zmian, zaś generator pisków zosał zastąpiony trzykanałowym generatorem porządnego dźwięku. We wszystkie Spectrum 128 wbudowano porty joysticków, lecz używać można jedynie joysticków SJS1 specjalnie produkowanych przez firmę Sinclair. W sumie, wydaje mi się, że naprawdę lepiej jest kupić wystuzoną „Spectrumnę” Plus i borykać z magnetofonem. Jeżeli jednak potrzebny jest krok dalej, proponuję Amstrada 6128 lub nawet IBM.





Adres: Przedsiębiorstwo "PRO-INFO"

ul. Sikorskiego 18/38  
40-001 KATOWICE  
skrytka pocztowa 1347  
tel. 53-42-88

## U W A G A !

### Originalna dokumentacja w języku angielskim na IBM PC XT/AT

Word Perfect 5.0; Quatro 1.0; Turbo Pascal 5.0  
Turbo C 2.0; Turbo Assembler 1.0; Turbo Debugger 1.0

### XENIX - w języku polskim

podręcznik dla programistów i operatorów  
10 tomów, 700 stron.

### Wkrótce:

- podstawowa dokumentacja do dBase IV
- podstawowa dokumentacja Clipper '88
- zbiorcze opracowanie nt. baz danych.

### Oraz programy użytkowe:

- KATALOG kartoteka silników elektrycznych
- ZBYT kalkulacja, fakturowanie i rozliczenie sprzedaży
- PROJEKT ewidencja twórców i projektów racjonalizatorskich
- ANALIZA analiza awaryjności i niezawodności wyrobów

SB17

### 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)

AWASOFT — Spectrum, Atari, Timex, IBM, Atari ST  
Naprawy i wysyłka programów, skr. poczt. 124,  
00-950 Warszawa. (SB 32)

SDS ATARI, programy tanio, szybko. Opole, ul. Krajewskiego 19A/603. (SB 31)

## INTERSOFT

00-443 Warszawa ul. Górnośląska 9/11 tel 21-56-08 tlix 81-72-45

### Programy Edukacyjne

	cen	ZX SPECTRUM	AMSTRAD CPC	ATARI XL/XE
1. DIODA-1 Animowany film ilustrujący zjawiska na łączu p-n.	19.000			X
2. ELEKTRONIKA Ilustrowany test z elektroniki, uczący podstawowych pojęć i symboli elektronicznych.	10.000			X
3. GRAWITACJA Symulacja wzajemnego oddziaływania dwóch ciał w ich polu grawitacyjnym.	8.000			X
4. KINEMATYKA 2 Pakiet trzech programów na temat pola grawitacyjnego.	8.000			X
5. SOCZEWKI Powstawanie obrazów rzeczywistych i pozornych w soczewkach wklęsłych i wypukłych.	5.000			X
6. FUNKCJA LINIOWA Własności wykresu funkcji liniowej oraz wpływ współczynników na wykres.	6.000			X
7. PRZEKROJE Graficzne przedstawienie przekrojów brył geometrycznych. Projektowanie własnych brył.	8.000			X X
8. STOLICE EUROPY Gra dydaktyczna ucząca położenia stolic państw Europy.	11.000			X
9. GRAMATYKA 1 Ćwiczenia z logicznego rozbioru zdań.	7.000			X
10. JĘZYK ANGIELSKI Tłumaczenie i wywoda zdań w oparciu o podręcznik L.Szutnika.	8.000			X X
11. KRÓLÓWIE POLSKI Nowe połączenia faktów historycznych z okresu panowania powstających królów Polski.	8.000			X

JAK KUPIWAĆ PROGRAMY ? - indywidualnie za gotówkę lub za zaliczeniem pocztowym - instytucje za gotówkę lub czekiem. Ceny programów nie zawierają ceny nosisników. Na życzenie zamówienia realizujemy na nośnikach klientów. Pełen katalog wraz z krótkimi opisami wysyłamy gratis.

## INTERSOFT

SB19

Programy na ATARI i SPECTRUM  
pisz do:  
STUDIO ATARBIT  
skr. poczt. 7  
42-550 SOSNOWIEC 18  
Co 10 program GRATIS  
Za kopertę + znaczek  
otrzymasz katalog.  
**ZAPRASZAMY**  
641

## MICROMAN

Programy na Atari XL/XE, Spectrum 48 kB, Commodore 16/116/+4 na miejscu lub za zaliczeniem pocztowym. Informacje za załączeniem koperty i znaczka pocztowego.  
40-181 Katowice, ul. Osikowa 66, tel. 58-51-06.  
D-204

#### Interface

do ATARI i do SPECTRUM drukarki, joysticki, do dowolnego magnetofonu również — TURBO. Sterowniki dzwonek szkolnych, makiet, reklam i inne. PAWTRONIK Warszawa tel. 659-38-44  
D-206

Dyskietki firmowe PRECISION, BASF, 3M, MAXELL, BONUS, DYSAN, VERBATIM — najtaniej od dostawcy z U.S.A. Informacje, cenniki: „Elektronika”, Kraków, Proszowicka 9, tel. 34-19-10.  
D-207

## ZX SPECTRUM

Naprawy klawiatur

Interfejsy:  
KEMPSTON 9.600 zł  
SINCLAIR 11.000 zł  
KEMPSTON + SINCLAIR 16.000 zł  
reset dopłata 1.000 zł  
wyjście monitora dopłata 5.000 zł

## ATARI

interfejs do każdego magnetof. 12.000 zł  
Roczna gwarancja  
Błyskawiczna wysyłka pocztą  
WINUE ul. Meissnera 14 m 1  
03-982 WARSZAWA  
tel. 15-93-38 godz. 18—20

## ZX SPECTRUM!

Nowe polskie programy do zabawy, eksperymentów oraz ciekawych zastosowań:

- zestaw TOTO (DL, SL, Ex, ZS)
  - LIGA POLSKA — dla każdego kibica piłki nożnej!
  - KAMIL — nowa gra tekstowa
  - inne atrakcyjne programy
- Informacje kopertą zwrotną.

#### MASTER BIT

61-660 Poznań 31 skr. p. 56

Commodore +4, C16, C116  
Programy, największy wybór

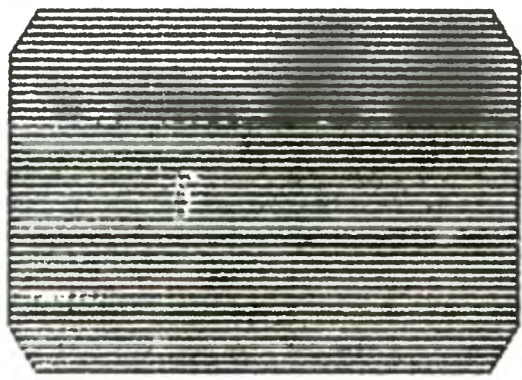
#### ASSTROCO

Podwłocze 24/53 35-310 Rzeszów  
G-19

## JOYSTICK SERVICE

Zgłoszenia: Studio komputerowe SEZAM  
D.H. "SEZAM" Ilp. - czwartki 16" - 19"  
Korespondencja: JOYSTICK SERVICE  
02-770 Warszawa 130 skr. poczt. 102  
tel. grzechn. 41-22-22 D-241





Impreza, na której winienes być obecny

# INFORMACJA '89



10 - 14 październik '89

Hala Widowiskowo-Sportowa "Spodek"-Katowice

- Informatyka w zarządzaniu - INFO '89
- Informatyka w medycynie - INFOMED
- Międzynarodowe sesje z udziałem Stowarzyszenia Dziennikarzy Nauki i Techniki
- Seminarium o tematyce: informatyka audio-video telewizja satelitarna

*Jeśli chcesz być obecny wytnij poniższy kupon i wyślij*

INF '89

Jestem zainteresowany:.....stoisko pokazowe

.....stoisko informacyjne

Nazwisko.....

Stanowisko.....

Firma.....

Dziedzina zainteresowań.....

Adres.....

telefon..... telex.....

Adresat:

Janusz Gołuch

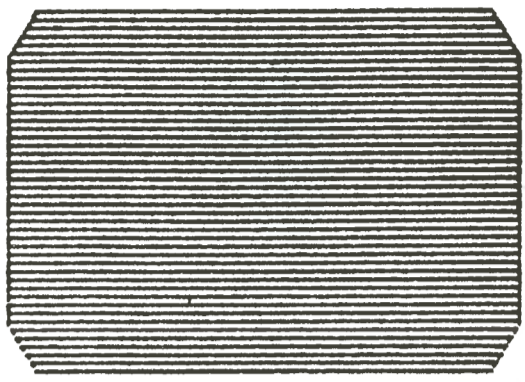
"PRO - INFO"

40-001 Katowice 1 skr. poczt. 1347

tel. (832) 53-42-88

tlx. 312401 info pl





Let your firm not be absent at

# INFORMACJA '89

10 - 14 october 1989

Show Hall "Spodek", Katowice, Poland

All the firms electronics or informatics and willing to promote and present their products in Poland are welcome to participate in our Fair.

Clearly then, the Show Hall "Spodek" in October 89 is the place to be at if new business is your goal. For further information please complete and return the coupon or telephone (832) 5342-88 or telex 312401 info pl.

Now we'd like to hear from you.....

I am interested in: .....exhibition stand  
.....information stand

Name.....

Position.....

Company.....

Business Category.....

Address.....

.....

Telephone.....Telex.....

to: Janusz Goluch  
"PRO-INFO" Co. Ltd.  
40-001 Katowice 1 P.O.Box 1347  
tel. (832)5342-88 tlx. 312401 info pl.



## GIEŁDA NA DZIEŃ 9 IV 1989

	Giełda Bajtka	Sklep Bajtka	Komis	Pewex	RFN	Wlk. Brytania	CSH i inne
	tys. zł	tys. zł	tys. zł	\$	DM	#	tys. zł
<b>SINCLAIR</b>							
ZX 81	90	80	—	—	—	—	—
ZX Spectrum 48	330	280	350	115	80	40	—
ZX Spectrum +	360	350	400	—	90	55	—
Timex 2048	320	330	400	146	—	—	240
ZX Spectrum 128+	430	—	—	—	—	85	—
ZX Spectrum 128+2	—	—	—	—	140	100	—
ZX Spectrum 128+3	—	—	—	—	280	150	—
drukarka Seikosha GP 50s	—	220	—	—	—	—	—
Interface Kempston	15	10-15	—	—	10	9	9.3

## COMMODORE

Commodore 64	600	650	710	219	290	120	—
VC 20	100	180	—	—	—	—	—
C 16	180	250	—	—	80	—	—
C 116	220	240	—	—	70	40	—
C Plus 4	310	360	380	—	150	—	—
C 128	900	910	—	299	399	170	—
C 128 D	1700	—	—	—	820	250	—
Amiga 500	2200	2900	—	—	899	280	—
Magnetofon 1531	100	120	170	48	30	49	—
Stacja dysków 1541	530	680	—	—	355	—	—
Stacja dysków 1571	780	800	—	299	460	160	—
Drukarka MPS 803	500	600	—	—	260	100	—

## ATARI

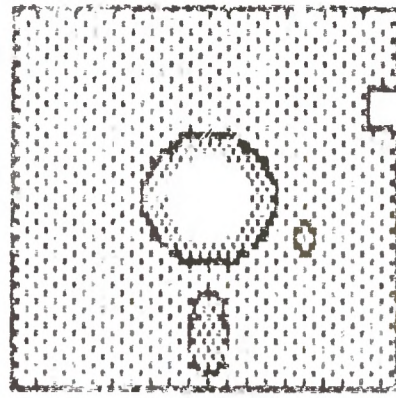
Atari 800 XL	380	500	470	—	—	—	—
Atari 65 XE	400	510	520	114	100	80	—
Atari 130 XE	500	680	—	199	220	110	—
Atari 520 ST	1500	1800	—	—	—	260	—
Atari 1040 ST	—	—	—	—	1140	499	—
Magnetofon XC 12	120	130	—	36	40	42	—
Stacja dysków 1050	700	700	750	185	300	100	—
Stacja dysków 520 STM	—	500	—	—	—	120	—
Drukarka 1029	400	650	—	—	—	—	—

## AMSTRAD

Amstrad 464 mono.	700	700	—	—	350	150	—
Amstrad 664 mono.	—	900	—	—	—	—	—
Amstrad 6128 mono.	110	—	—	—	670	220	—
Amstrad PCW 8256	—	—	—	—	—	300	—
Amstrad PCW 8512	—	—	—	—	—	390	—
Amstrad PCW 9512	—	—	—	—	—	450	—
Stacja dysków do 464	—	—	—	—	380	100	—

## SHARP

Sharp MZ 700	—	360	—	—	—	—	—
Sharp MZ 800	—	420	—	—	—	—	—
Dyskiety 5.25 cala	2	2-2.5	3	1	0.7	0.5	4-9
Dyskiety 3.5 cala	7.5	7.5-9	7-9	—	5	1-2	9-10
Dyskiety 3 cale	11	—	—	—	6	2	10
Joystick	21	16-26	20-30	5	10	5	14
Monitor Neptun	60	80	—	—	—	—	68



**Przemysław Gwoździowski** — jest posiadaczem ZX Spectrum+2. Prosi o kontakt posiadaczy sprzętu z rodziny 128k Spectrum. Proponuje wymianę oprogramowania, opisów programów oraz doświadczeń. Zainteresowany jest głównie programami muzycznymi. Oferuje opis przeróbki joysticka Quick Shot V do Sinclair Interface 2. Adres: 39-300 Mielec, ul. Warszawska 6/44.

**Radostaw Wołosz**, uczeń lat 16. Prosi posiadaczy mikrokomputera TANDY TRS-80 Color Computer 2 o pomoc w zdobyciu oprogramowania. Mikrokomputer ten jest kompatybilny z Texas instruments. Adres: 23-210 Kraśnik, ul. Makuszyńskiego 9.

**Przemysław Orłowski**, lat 15. Jest posiadaczem Atari 130 XE, stacji LDW 2000 oraz magnetofonu XC 12. Oprogramowanie: około 500 programów użytkowych oraz gier. Proponuje wymianę oprogramowania, literatury oraz opisów. Adres: 71-401 Szczecin, ul. Wyzwolenia 75a/41.

**Andrzej Hofman**, komputer Atari 800 XL, stacja dysków 1050, magnetofon polski i interfejs. Oprogramowanie: gry i programy użytkowe. Nawiąże kontakt z posiadaczami tego komputera. Adres: 43-300 Bielsko-Biała, ul. I Armii Wojska Polskiego 4/41.

**Zdenek Habla**, posiada mikrokomputer ZX-Spectrum+. Pragnie nawiązać korespondencję z posiadaczami tego mikrokomputera w celu wymiany oprogramowania oraz doświadczeń. Korespondencja w języku angielskim. Adres: 627 00 Brno, Langrova 23, Czechosłowacja.

**Arkadiusz Dawkrzyc** prosi posiadaczy komputera VIDEOPAC G 7000 firmy PHILIPS o kontakt. Adres: 66-460 Witnica, ul. Traugutta 2a/3, woj. Gorzów Wlkp.

**Mariusz Kowalik**, uczeń lat 14. Posiada Timex 2048 z magnetofonem MK 450. Opro-

## INDYWIDUALNY BANK DANYCH

gramowanie: około 350 gier oraz programy użytkowe. Proponuje wymianę gier oraz literatury. Adres: 24-100 Puławy, ul. Słowackiego 24/26 m 8.

**Robert Sard**, lat 15. Posiada mikrokomputer Commodore C-128, kolorowy monitor oraz stację dysków 1571. Oprogramowanie: gry i programy użytkowe. Proponuje wymianę gier i programów użytkowych na dyskietkach. Adres: 41-800 Zabrze, ul. Waryńskiego 2/9.

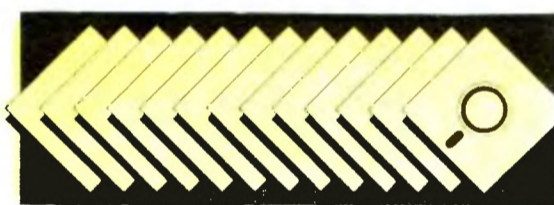
**Monika Dobosz** lat 14. Posiada Atari 800 XL prosi o kontakt z użytkownikami tego mikrokomputera w celu wymiany programów i gier. Adres: 95-100 Zgierz, ul. Witkacego 19 m 15 bl 62.

**Dominik Kowalczewski** lat 17. Posiada Amstrada PC 1512 SD. Chce nawiązać kontakt z posiadaczami tego komputera lub kompatybilnego. Zainteresowany jest głównie grami oraz językami programowania. Adres: 28-200 Staszów, ul. Karasia 18/20, woj. Tarnobrzeg.

**Rafał Makowiecki** prosi o pomoc w zdobyciu programów oraz literatury na komputer SPECTRAVIDO SVI-738 MSX X'PRESS. Adres: 65-091 Zielona Góra, ul. Lisia 39 m 43.

**Jarosław Pysz**, lat 17. Posiada Atari 800 XL, stację dysków 1050, drukarkę 1029 oraz monitor NEC. Programy: około 350 gier oraz programy użytkowe. Nawiąże korespondencję w celu wymiany programów oraz instrukcji. Adres: 78-400 Szczecinek, ul. 28 Lutego 10 B/5.

**Tomasz Kucharczyk**, lat 16. Posiada Timex 2048, oraz około 400 programów (większość gry). Proponuje wymianę oprogramowania. Adres: 26-030 Suchedniów, ul. Bugaj 10 m 15.



## O ŻŁOTĄ DYSKIETKĘ BAJTKA...

ubiegać się będą w tym roku następujące kluby komputerowe (dzisiaj prezentujemy ich listę, a począwszy od następnego numeru przedstawimy na naszych łamach same kluby):

**„Amstrad-Schneider Club”** przy Zespole Szkół Górniczo-Energetycznych, ul. Czaplinska 100, 97-400 Bełchatów.

**„ATARI”** przy Leśnictwie Wręczyca, ul. Śląska 32, 42-130 Wręczyca Wlk.

**„ATARI TEST”** przy Wojewódzkim Domu Kultury w Piotrkowie Trybunalskim, Al. Bieruta 12, 97-300 Piotrków Trybunalski.

**„BOBAS-ATARI”**, klub nieformalny, Gliwice.

**„C. C. ATARI”** przy Choszczeńskim Domu Kultury, ul. Bohaterów Warszawy 17, 73-200 Choszczno.

**„CELBIT”** przy Zakładach Papierniczych Kwidzyn, ul. Warszawska 18, 82-500 Kwidzyn.

**„Future”**, klub nieformalny, Oleśnica.

**Harcerski Klub Komputerowy „HarcBajt”** przy Domu Harcerza, ul. Za Murami 2—10, 80-823 Gdańsk.

**Harcerskie Centrum Komputerowe** przy Komenazie Chorągwi ZHP Lublin, POK „Dom Chemika” w

Puławach i Wydziale Oświaty i Wychowania, ul. Wojska Polskiego 4, 24-100 Puławy.

**Inspektorat Mikrokomputerowy Komendy Chorągwi ZHP, Klub Komputerowy „WITEZ”**, ul. Karczówkowska 20, 25-731 Kielce.

**Klub Komputerowy przy Robotniczym Centrum Kultury WSK „PZL-Mielec”**, ul. 22 Lipca, 39-300 Mielec.

**Klub Komputerowy DK „Masters”** przy Domu Kultury w Rawiczu, ul. Targowa 1, 63-900 Rawicz.

**Klub Komputerowy MMT „ALFA”** przy Spółdzielni Mieszkaniowej w Stargardzie Szczecińskim, ul. Wieniawskiego 32, 73-110 Stargard Szczeciński.

**Klub Komputerowy MMT „BIN”** przy ZSMP Kopalni Węgla Brunatnego „Adamów” w Turku, ul. Uniejowska 9, 62-700 Turek.

**Klub Mikrokomputerowy przy Pałacu Młodzieży w Szczecinie**, Al. Wojska Polskiego 80, 70-482 Szczecin.

**„MAD BOYS ATARI”**, klub nieformalny, Żyrardów.

**„MNEMONIK”** przy zakładach Elektronicznych ELWRO we Wrocławiu, ul. Ostrowskiego 30, 53-238 Wrocław.

**Młodzieżowy Klub Komputerowy „MIKROBAJT”** przy Międzyzakładowym Klubie Pracowników Poligrafii „Chochlik”, ul. Rewolucji 1905 r. nr 10, Łódź.

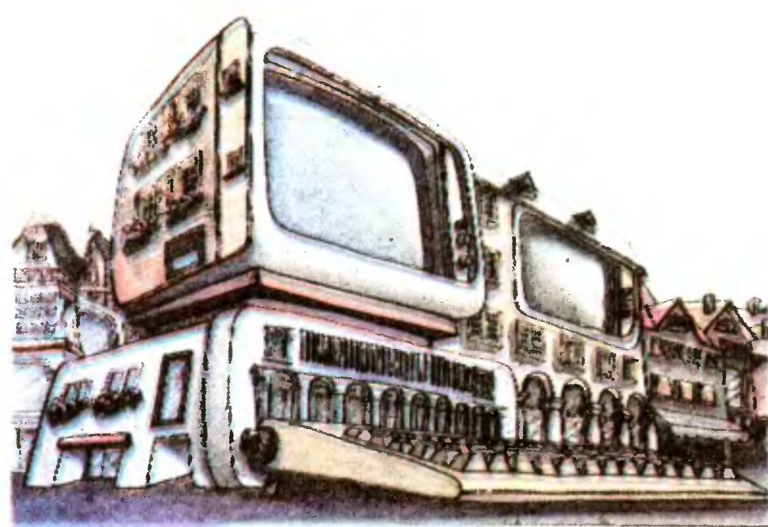
**„SPECTRUM”** przy Zespole Szkół Mechaniczno-Energetycznych, ul. Malinowa 8, 20-143 Lublin.

**„TIME-Studio”** przy Zespole Szkół Ogólnokształcących w Łęcznej, ul. Szkolna 3, 21-010 Łęczna.

**„Złoty Amstrad”** przy Zakładach Tworzyw i Farb Złoty Stok, ul. Rynek 1, 57-250 Złoty Stok.

(b)





## PROGRAM

## DLA BIANECZKI

Przemek ma 13 lat, jego młodsza siostra Bianka ukończyła 10 i chodzi do czwartej klasy. Dla niej właśnie napisał Przemek na Amstradzie 464 program, który Wam dzisiaj przedstawiam. Służy on do ćwiczenia szybkiego obliczania w pamięci wyników czterech podstawowych działań arytmetycznych. Nie jest to jednak zwykły test, jakich otrzymujemy dziesiątki, ale prawdziwa gra komputerowa.

*Romek*

Wyobraźcie sobie, że nad naszym miastem zawisło niebezpieczeństwo. Przybysze z kosmosu postanowili je zniszczyć zrzucając burzące bomby. Spadającą bombę można rozbroić wykonując napisane na niej działanie. Jeśli uczynimy to bezbłędnie, bomba zawisnie w powietrzu i będziemy mogli ją zestrzelić. Jeśli jednak pomylimy się lub nie starczy nam czasu, bomba spadnie na miasto, a kilka domów legnie w gruzach.

Grający ma możliwość ustawiania prędkości i wyboru działań arytmetycznych. Tak więc w miarę doskonalenia swoich możliwości, można podnosić poprzeczkę.

**autor programu:**

**Przemek Siwiński lat 13**

```
1 SYMBOL AFTER 237:SYMBOL 237,0,0,159,1
96,126,63,15,6
2 SYMBOL 238,63,63,49,49,49,49,63,63
3 SYMBOL 239,252,252,140,140,140,140,252,
252
4 ENV 1,5,-1,2
5 MODE 1:INK 0,0:INK 1,26:INK 2,16:INK 3
,10,24
```

```
6 GOSUB 54:GOSUB 64
7 MODE 1:PAPER 0:CLS
8 BORDER 18:PEN 1:LOCATE 1,22
9 q$=CHR$(222)+CHR$(223):e$=CHR$(238)+CHR$(239):w$=" "
10 LOCATE 1,20:PEN 1:PRINT w$;q$;w$;w$;q$;w$;q$;w$;q$;w$;q$;w$;q$;w$;q$;w$;
q$;w$
11 LOCATE 1,21:PEN 2:PRINT w$;e$;:PEN 1:
PRINT q$;w$;:PEN 2:PRINT e$;w$;e$;w$;e$;
w$;w$;e$;w$;e$;w$;w$;e$;w$;e$;w$
12 LOCATE 1,22:PEN 1:PRINT w$;e$;e$;w$;e$;
w$;e$;w$;e$;w$;q$;e$;w$;e$;w$;q$;e$;w$;
e$;w$
13 LOCATE 1,23:PRINT w$;e$;e$;q$;e$;q$;e$;
q$;e$;q$;e$;e$;q$;e$;q$;e$;e$;q$;e$
14 LOCATE 1,24:PEN 2:PRINT w$;e$;e$;e$;e$;
e$;e$;e$;e$;e$;e$;e$;e$;e$;e$;e$;e$;e$;
e$;
15 LOCATE 1,25:PEN 1:PRINT CHR$(214);STR
ING$(38,143):CHR$(215);
16 y=INT(RND*28)+4;z=4;a=10:IF wo=1 THEN
lo=INT(RND*4)+1
17 ON lo GOSUB 19,20,21,22
18 GOTO 23
19 zn$="+":c=INT(RND*60)+10:d=INT(RND*60)
)+10:w=c+d:RETURN
20 zn$="-":c=INT(RND*90)+10:d=INT(RND*50)
)+10:IF c<d THEN 20 ELSE w=c-d:RETURN
21 zn$="*":c=INT(RND*7)+4:d=INT(RND*7)+4
:w=c*d:RETURN
22 zn$="/":c=INT(RND*70)+20:d=INT(RND*10)
)+5:IF c/d=c\d THEN w=c/d:RETURN ELSE 2
2
23 PEN 1:LOCATE 1,a:PRINT CHR$(237)
24 h=h+1:IF kod=0 AND h/s=h\s THEN GOSUB
30
25 k$=INKEY$
26 IF VAL(k$)<>0 OR k$="0" THEN GOSUB 34
27 IF k$=CHR$(13) THEN GOTO 79
28 PEN 3:LOCATE 2,1:PRINT a$:PEN 1
29 GOTO 24
30 IF z=21 THEN GOTO 35
31 PEN 2:LOCATE y,z:PRINT CHR$(24);c;zn$
;d;CHR$(24)
32 LOCATE y,z-1:PRINT " "
33 z=z+1:SOUND 1,INT(RND*400)+100,2,15,1
,1,0:RETURN
34 a$=a$+k$:RETURN
35 SOUND 1,1000,50,15,,30
36 LOCATE y-1,20:PRINT " "
37 LOCATE y-1,21:PRINT " "
38 LOCATE y-1,22:PRINT " "
39 LOCATE y,23:PRINT " "
40 LOCATE y+1,24:PRINT " "
41 u=u+1:LOCATE 1,a:PRINT " ":GOTO 88
42 DATA 142,30,142,15,142,15,142,30,142,
15,142,15,142,30,213,30,179,30,142
43 DATA 30,159,30,159,15,159,15,159,30,1
59,15,159,15,159,30,239,30,190,30
44 DATA 159,30,142,30,142,15,142,15,142,
30,142,15,142,15,142,30,127,30,119
45 DATA 30,106,30,119,30,142,30,159,30,1
90,30,213,60,213,60
46 FOR x=1 TO 1000:NEXT
47 RESTORE 42:FOR a=1 TO 36:READ x,y
48 SOUND 1,x,y,15:SOUND 2,x/1.5,y,15
49 SOUND 1,0,1:SOUND 2,1,1
50 NEXT a
51 PEN 3:IF u=0 THEN LOCATE 1,1:PRINT "
!Brawo! Uratowałeś Miasteczko !!!0":
GOTO 53
```

```
52 LOCATE 1,1:PRINT " Na Miasteczko sp
adli0";u;"bomb !"
53 WHILE INKEY$<>"":WEND:WHILE INKEY$=""
:WEND:RUN
54 MODE 1:PEN 1:CLS
55 PRINT:PRINT:PRINT:PRINT" Klawisze ste
rujace : "
56 PRINT:PRINT:PRINT:PRINT:PRINT" ENTER
--> zatwierdzenie lub skasowanie"
57 PRINT:PRINT" wyniki":PRINT:
PRINT
58 PRINT"1 SPACE --> strzał do bomby0"
59 PRINT:PRINT:PRINT:PRINT:PEN 2
60 PRINT"0 Szybkosc spadania bomby ?":PR
INT
61 INPUT" 1-szybko 30-wolno ",s
62 IF s<1 OR s>30 THEN 59
63 RETURN
64 PEN 1:MODE 1
65 LOCATE 5,6:PRINT"Dodawanie"
66 LOCATE 5,8:PRINT"Odejmowanie"
67 LOCATE 5,10:PRINT"Mnozenie"
68 LOCATE 5,12:PRINT"Dzielenie"
69 LOCATE 5,14:PRINT"Wszystko naraz"
70 PEN 2:LOCATE 5,20:PRINT "a - ";CHR$(2
40);" ";z - ";CHR$(241)
71 PEN 2:LOCATE 5,22:PRINT"SPACE - zatwi
erdzenie"
72 x=1:PEN 3:WHILE INKEY$<>"":WEND
73 LOCATE 3,x*2+4:PRINT CHR$(243)
74 k$=INKEY$
75 IF UPPER$(k$)="Z" AND x<5 THEN LOCATE
3,x*2+4:PRINT " :x=x+1:GOTO 73
76 IF UPPER$(k$)="A" AND x>1 THEN LOCATE
3,x*2+4:PRINT " :x=x-1:GOTO 73
77 IF k$=" " THEN IF x<5 THEN lo=x:RETUR
N ELSE wo=1:RETURN
78 GOTO 74
79 IF VAL(a$)<>wy THEN LOCATE 1,1:FOR x=
1 TO LEN(a$)+1:PRINT " ";NEXT:a$="":GOTO
24
80 g=-1:FOR p=1 TO 150
81 a=a+g
82 PEN 1:LOCATE 1,a:PRINT CHR$(237)
83 LOCATE 1,a-g:PRINT " "
84 IF a=2 OR a=20 THEN g=g*-1
85 FOR x=1 TO 10:IF INKEY$=" " THEN GOTO
91 ELSE NEXT
86 NEXT p
87 LOCATE y,z-1:PRINT " ":LOCATE
1,a:PRINT " "
88 SOUND 1,2000,50,15,,30:SPEED INK 1,1
:BORDER 0,26
89 FOR t=1 TO 300:NEXT:BORDER 18:SPEED I
NK 12,12
90 WHILE INKEY$<>"":WEND:WHILE INKEY$=""
:WEND:GOSUB 98:GOTO 16
91 FOR x=2 TO y+10
92 PEN 1:LOCATE x,a:PRINT CHR$(237):LOCA
TE x-1,a:PRINT " "
93 NEXT
94 IF a<>z-1 THEN LOCATE y+10,a:PRINT " "
:GOTO 87
95 SOUND 1,239,5,15:SOUND 1,311,5,15:SOU
ND 1,371,5,15:SOUND 1,311,5,15:SOUND 1,3
71,5,15:SOUND 1,239,10,15
96 do=do+1:IF do=13 THEN GOTO 46
97 LOCATE y+10,a:PRINT " ":GOSUB 98:GOTO
16
98 LOCATE 2,1:FOR x=1 TO LEN(a$):PRINT"
":NEXT:a$="":RETURN
```



# TRZY NACZYNIA

## Cześć Maluchy!

Bywa czasem, że nawet najwięksi ludzie, najtęższe umysły zmuszeni są poświęcić nieco czasu rzeczom zwykłym i codziennym. Zdarza się wówczas, że proste, nieciekawe czynności stają się pretekstem do niezwykle interesujących rozważań. Anegdota z życia Simeona Denisa Poisson (1781—1840), francuskiego matematyka i fizyka, profesora Sorbony stanowi doskonały przykład takiego zdarzenia.

Pewnego razu Poisson wybrał się wraz ze swym przyjacielem, również matematykiem, na długą, pieszą wycieczkę. Późnym wieczorem zdrożeni wędrowcy dotarli do samotnej karczmy. Obydwaj przyjaciele byli już solidnie głodni i spragnieni. Zamówili więc dzban wina w zamiarze osuszenia go nim karczmarz przygotuje posiłek. Matematycy postanowili podzielić się winem po połowie, tu jednak pojawiła się pewna trudność. Gospodarz przyniósł wino w jednym, pełnym naczyniu ośmiolitrowym. Na prośbę gości użyczył im jeszcze dwóch naczyń, o pojemności 5 i 3 litrów, dodając, że w karczmie nie ma już innych naczyń.

Po kilku próbach przelewania wina z jednego dzbana do drugiego, profesorowie mieli ochotę zrezygnować ze sprawiedliwego podziału na rzecz bezzwłocznego zwilżenia wyschniętych gardel, lecz przekora i upór właściwe wszystkim wielkim naukowcom wzięły górę...

Anegdota nie mówi jak długo przyjaciele trudzili się tym problemem, nie wspomina też o tym, co myślał sobie karczmarz widząc co ci dwaj spragnieni podróżni wyprawiają z jego winem. Jako fakt podaje jednak, że w końcu każdy z nich wypił swoje 4 litry wina (dla wyjaśnienia dodam, że było to bardzo słabe wino, rozcieńczane wodą).

Wynika stąd, że istnieje rozwiązanie tego problemu. Nie jest ono jednak proste, o czym możesz się sam przekonać. Nie biegnij jednak od razu do kuchni i nie wyciągaj matki garnków z szafki. O ile pamiętam, mamy zwykle nie przepadają za robieniem z kuchni basenu kąpielowego. Znacznie bezpieczniej będzie jeśli usiądziesz przed komputerem i wpiszesz ten program.

Rolę naczyń spełniają w naszym programie tablice **pojemnosc(3)** i **zawartosc(3)**. Już same nazwy

tablic określają jakie wartości będą w nich przechowywane. I tak na przykład **pojemnosc(2)=5** oznacza, że pojemność naczynia drugiego wynosi 5 litrów, a **zawartosc(2)=3** oznacza, że w tym naczyniu znajdują się 3 litry wina.

W liniach 30—50 następuje wczytanie początkowych wartości tablic. W trakcie gry zmianie ulegają jedynie wartości umieszczone w tablicy **zawartosc(3)**, natomiast tablica **pojemnosc(3)** zachowuje wartości początkowe.

Zasadnicza pętla programu mieści się w liniach 100—340. Na początek drukowana jest na ekranie informacja o napełnieniu kolejnych butelek (linie 120—150). Przy okazji sprawdzane jest, czy w którymś z naczyń nie pozostały przypadkiem dokładnie 4 litry wina, co oznaczałoby spełnienie warunków gry (linia 140). Pętla wyposażona jest także w licznik kolejnych sytuacji (linie 70, 100 i 330), który pozwala określić po ilu przelewaniach uda nam się osiągnąć sukces.

W następnej kolejności komputer pyta gracza kolejno o to, z której butelki i do której chce on przelać wino (linie 200—260). Numery butelek zapamiętuje odpowiednio jako zmienne **z** i **d**. Przy okazji sprawdza, czy wprowadzone liczby mogą być numerami butelek (linie 220 i 250), oraz czy gracz nie chce przelać wina do butelki, z której właśnie wylewa (linia 260). W każdym z tych przypadków komputer rozpoczyna pytanie od nowa.

Dysponując wszystkimi danymi komputer może przystąpić do „przelewania” (linie 300—320 i 1000—1020). Zawartość butelki **d** — **zawartosc(z)** zwiększana jest o zawartość butelki **z** — **zawartosc(z)** (linia 300). Zawartość butelki **z** — **zawartosc(z)** przyjmuje wartość 0 (linia 310). Może się jednak zdarzyć, że zawartość butelki **d** — **zawartosc(d)** jest większa niż pojemność tej butelki — **pojemnosc(d)** (linia 320). Komputer wykonuje wówczas dodatkowe operacje (podprogram w liniach 1000—1020). Zawartość butelki **z** — **zawartosc(z)** przyjmuje wartość równą różnicy zawartości butelki **d** — **zawartosc(d)** i jej pojemności — **pojemnosc(d)** (linia 1000). Zawartość butelki **z** — **zawartosc(z)** przyjmuje wartość jej pojemności — **pojemnosc(z)**.

Teraz pozostało jedynie zwiększenie wartości licznika **s** o 1 i pętla rozpoczyna się od nowa. Program trwa aż do momentu, gdy zadanie nie zostanie wypełnione lub... aż nie zostanie on przerwany.

\* \* \*

Proponuję abyś do rozwiązania tego zadania przygotował się w szczególny sposób. Przez cały dzień nie bierz do ust ani kropli wody. Wieczorem przygotuj sobie dzbanek pysznego, zimnego soku pomarańczowego. Postaw go obok komputera i... uruchom ten program. Przekonasz się wówczas, czy nadajesz się — jak profesor Poisson — na wielkiego naukowca.

Czego życzy ci z całego serca

Romek

```

9 REM ***** dane początkowe *****
10 DIM pojemnosc(3)
20 DIM zawartosc(3)
30 FOR i=1 TO 3
40 READ pojemnosc(i),zawartosc(i)
50 NEXT i
60 DATA 8,5,0,3,0
70 LET s=0
99 REM * wydruk informacji o butelkach *
100 PRINT "sytuacja";s
110 PRINT
120 FOR i=1 TO 3
130 PRINT i;"butelka";pojemnosc(i);"-1
itrowa zawiera";zawartosc(i);"litrów."
140 IF zawartosc(i)=4 THEN GOTO 2000
150 NEXT i
160 PRINT
199 REM **** komunikacja z grającym ****
200 PRINT "Z której butelki chcesz przelać?"
210 INPUT z
220 IF z<>1 AND z<>2 AND z<>3 THEN GOTO 200
230 PRINT "Do której butelki chcesz przelać?"
240 INPUT d
250 IF d<>1 AND d<>2 AND d<>3 THEN GOTO 200
260 IF z=d THEN GOTO 200
299 REM ***** przelewanie *****
300 LET zawartosc(d)=zawartosc(d)+zawartosc(z)
310 LET zawartosc(z)=0
320 IF zawartosc(d)>pojemnosc(d) THEN GOSUB 1000
330 LET s=s+1
340 GOTO 100
999 REM **** brak miejsca w butelce ****
1000 LET zawartosc(z)=zawartosc(d)-pojemnosc(d)
1010 LET zawartosc(d)=pojemnosc(d)
1020 RETURN
1999 REM ***** koniec programu *****
2000 PRINT "Brawo!"
    
```



## 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)

## „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

Programy na Commodore 16, 116, +4 wysyłamy pocztą. Koperta zwrotna i znaczek — otrzymasz katalog. Nagrywanie programów komputerowych Tadeusz Mieczkowski ul. Lenina, 104/3, 58-304 Wałbrzych.

(SB 28)

## JAK REKLAMOWAĆ SIĘ W BAJTKU?

Reklamy przyjmuje Biuro Reklamy „Sztandaru Młodych”, Warszawa 1, skr. pocztowa 363, ul. Wspólna 61, tel. 28-02-56.

Cena reklamy czarno-białej wynosi 750 zł za cm<sup>2</sup>. Do ceny podstawowej doliczane jest 30% za dodatkowy kolor i 100% w przypadku reklamy wielobarwnej. Ogłoszenie drobne kosztuje 450 zł za 1 słowo.

Wyżej wymienione kwoty można również wpłacać bezpośrednio na konto MAW: Państwowy Bank Kredytowy w Warszawie, III Oddział Warszawa nr 370015-5757.

W miejscu na korespondencję prosimy zaznaczyć, że wpłata dotyczy ogłoszenia w Bajtku. Ogłoszenia drobne oddawane są do druku po otrzymaniu dowodu wpłaty z NBP.

**Ceny te obowiązują od stycznia 1989 roku.**





John V. Atanasoff

**W 1942 roku prace nad komputerem „ABC” zostały przerwane...**

# KTO ZBUDOWAŁ PIERWSZY KOMPUTER

**W roku 1967 przed sądem okręgowym w Minneapolis rozpoczął się proces o naruszenie praw autorskich przez koncern Honeywell Inc. Sprawę wniosła korporacja Sperry Rand będąca w posiadaniu patentu na komputer „ENIAC” i czerpiąca z tego tytułu zyski.**

Honeywell Inc. nie chciał płacić. Przed trybunałem zarzucił przeciwnikom naruszenie ustaw antytrustowych i... posługiwanie się nieważnym patentem. Sprawa zapowiadała się sensacyjnie nie tylko z tego powodu, że w grę wchodziły ogromne pieniądze, ale też dlatego, że sąd miał rozstrzygnąć kwestię, kto zbudował pierwszy komputer. W tamtych latach powszechnie przyjmowano, że pierwszą maszyną liczącą zbudowaną w oparciu o lampy elektronowe był „ENIAC”, skonstruowany w roku 1946 przez zespół naukowców z Moore School of Engineering w University of Pensylwanii kierowany przez J.P. Eckerta i I.W. Mauchly'ego. To oni właśnie w roku 1950 stali się właścicielami patentu na koncepcję komputera.

Prawnicy Honeywell oświadczyli, że w trakcie przygotowań do procesu natrafili na wzmiankę o pracach, prowadzonych na przełomie lat trzydziestych i czterdziestych w Uniwersytecie Stanu Iowa przez doktora Johna V. Atanasoffa. Po rozmowie z nim okazało się, że już w roku 1939 zbudował on komputer przeznaczony do rozwiązywania 30 równań z 30 niewiadomymi. Pracował, podobnie jak współczesne komputery, w systemie dwójkowym: jedynka i zero były reprezentowane przez odpowiednią polaryzację kondensatorów. W roku 1942 sprawnie działała część elektroniczna, trwały też prace nad udoskonaleniem czytnika kart. Wezwany w charakterze świadka John V. Atanasoff zeznał przed sądem, że w grudniu 1940 roku przedstawił J. Mauchly'emu wyniki swych badań. Inni świadkowie potwierdzili to.

19 października 1973 roku sędzia E.A. Larsen wydał wyrok. Uznał, że — „Mauchly i Eckert nie byli pierwszymi odkrywcami automatycznego, opartego o układy elektroniczne komputera, a jedynie posłużyli się pomysłami Johna V. Atanasoffa” — w związku z tym ich patent tracił swą ważność. Sperry Rand przyjęła wyrok do wiadomości i wbrew zwyczajom panującym w Stanach Zjednoczonych nie wniosła apelacji do sądu wyższej instancji.

Historię tę opisał w jednym z ubiegłorocznych numerów „Scientific American” Allan R. Mackintosh, profesor fizyki eksperymentalnej ciała stałego, dyrektor Skandynawskiego Instytutu Fizyki Teoretycznej NORDITA,

który w roku 1983 w czasie wizyty w Uniwersytecie Iowa zapoznał się z odkryciami Atanasoffa.

W końcu lat dwudziestych, pisząc pracę doktorską poświęconą strukturze atomów helu, Atanasoff zmuszony był do wykonania ogromnej ilości obliczeń matematycznych. Jak wspomina dziś — marzył o wyeliminowaniu tej męczącej pracy przez automatyczne maszyny liczące. W roku 1930 świeżo upeczętowany doktor Atanasoff podejmuje pracę w Uniwersytecie Stanu Iowa. W siedem lat później dokonuje pierwszych odkryć teoretycznych w dziedzinie maszyn cyfrowych. Jego zdaniem proces zapamiętywania danych w maszynie powinien być oddzielony od procesu obliczeniowego, zaś metoda obliczania powinna być cyfrowa, a nie analogowa. Największym problemem był system dziesiętny, w którym zapisywano liczby. Możliwość zastosowania go w urządzeniu, nad którym pracował Atanasoff, była praktycznie żadna. Wiedział o tym, jednak, jak sam przyznał, żaden pomysł nie przychodził mu do głowy.

...Pewnego wczoraz — wspomina Atanasoff — postanowiłem odpocząć od pracy i wybrałem się samochodem na przejażdżkę. Po kilku godzinach znalazłem się w sąsiednim stanie Illinois — ponad dwieście mil od domu. Postanowiłem się upić w przydrożnym barze. W tym dniu było szczególnie chłodno, miałem na sobie ciężki zimowy płaszcz. Zdjąłem go, powiesiłem i zamówiłem u barmana podwójną whisky. Kiedy podał trunek, ożywiłem się. Myślałem wrócić do maszyn obliczeniowych. Nie wiem dlaczego w t e d y mój umysł zaczął pracować. Wszystko stało się jasne, myślałem precyzyjnie i spokojnie... Zapewne wypitem ze dwie kolejki zanim zrozumiałem, że przychodzą mi do głowy bardzo ciekawe myśli. W istocie pomysły zrodzone pod wpływem alkoholu okazały się znakomite.

Atanasoff doszedł do wniosku, że jego maszyna powinna pracować w systemie dwójkowym, zaś operacje liczbowe powinny odbywać się zgodnie z zasadami logiki, a nie na zasadzie prostych podliczeń. Zamiast mechanicznych elementów trzeba zastosować przełączniki elektroniczne, z pomocą których winny być realizowane arytmetyczne operacje komputera. Być może najciekawszym pomysłem tamtego dnia była idea zastosowania kondensatorów jako pamięci. Otwierało to możliwość skorzystania z systemu dwójkowego w pracy maszyny.

Znaczenie rozwiązań zaproponowanych przez Atanasoffa najłatwiej ocenić porównując jego komputer, który otrzymał nazwę „ABC”, ze stosowanymi wtedy mechanicznymi urządzeniami obliczeniowymi. Był on 1000 razy szybszy.

W roku 1942 przerwane zostały prace nad komputerem ABC. Jedną z przyczyn był brak funduszy. Nie było też specjalnego zainteresowania ze strony władz uczelni, na której pracował Atanasoff. On sam stracił chęci do dalszej pracy.

Gdyby jednak kontynuowano wysiłki — pisze Allan R. Mackintosh — z całą pewnością w roku 1943 urządzenie byłoby gotowe i w pełni sprawne. Tak się nie stało, a to co pozostało z komputera „ABC” zostało rozmontowane na części. Idee Atanasoffa zostały przejęte przez Mauchly'ego i zastosowane przy budowie „ENIAC”, który był już maszyną znacznie większą. Następnym krokiem był „EDVAC” — jeszcze doskonalsze urządzenie obliczeniowe, którego twórcą był John von Neumann, uważamy za jednego z „ojców” informatyki.

Atanasoff nie uzyskał patentu na komputer. Czy jednak sędzia Larsen miał rację wydając swój wyrok? Jesienią 1975 roku, po z górą trzydziestu latach milczenia, rząd brytyjski zezwolił archiwom państwowym na udostępnienie zdjęć pierwszego w świecie komputera noszącego nazwę „COLOSSUS”. Okazało się, że podczas II wojny światowej zbudowano w Anglii serię programowanych elektronicznych maszyn cyfrowych, z których pierwsza powstała w grudniu 1943 roku. Zespołem specjalistów kierowali Max H. Newman i T.H. Flowers. Budowę oparto na teoretycznych pracach Alana M. Turinga, który już parę lat wcześniej udowodnił, że możliwy jest uniwersalny automat, zdolny do wykonywania wszelkich obliczeń.

„Colossus” został zbudowany dla potrzeb Intelligence Service, a konkretnie centrali dekryptażu mieszczącej się w Bletchley, gdzie służył łamaniu szyfru „Enigmy”. Warto przypomnieć, że od roku 1943 Brytyjczycy przy odczytywaniu depesz niemieckich posługiwali się metodami opracowanymi przez Polaków jeszcze przed wojną. Tajemnica „Enigmy” była jedną z najściślej strzeżonych przez Niemców. Polskim kryptologom jeszcze przed wybuchem wojny udało się złamać mechaniczny szyfr „Enigmy”. Po klęsce wrześniowej pracodawcy Biura Szyfrów Oddziału II Sztabu Głównego przekazali zachodnim aliantom wyniki swych prac. Dzięki temu możliwy stał się dokładny wgląd w sekrety III Rzeszy. Polscy matematycy Marian Rejewski, Jerzy Różycki i Henryk Zygalski byli tymi, którzy na długo przed wrześniem prowadzili wojnę z Niemcami, dodajmy, że byli w tej wojnie zwycięzcami.

Elektroniczne maszyny cyfrowe opracowane dla celów dekryptażu otoczone zostały tajemnicą. Do dziś nie wiemy ile ich zbudowano, nie wiadomo też, co się z nimi stało po zakończeniu wojny. Wiadomo tylko, że „Colossus” został zbudowany z 1500 lamp zastępujących przekładniki elektromagnetyczne. Ulepszony „Colossus” („Mark 2”) zawierał 2400 lamp i działał z prędkością 5000 impulsów na sekundę.

Brytyjski historyk techniki komputerowej prof. B. Randall napisał w artykule „The Colossi” — Britain's Wartime Secret Computers, że urządzenia z serii „Colossus” okazały się niezawodne i służyły dobrze swym użytkownikom.

A co stało się z Atanasoffem? Nic szczególnego. Do emerytury pracował jako wykładowca fizyki na Uniwersytecie Iowa. Ma obecnie 86 lat i cieszy się dobrym zdrowiem. Historia jego odkryć może być przykładem tego jak ważną rzeczą, nie tylko w nauce, jest przypadek. Ktoż mógłby przypuszczać czym zakończy się dwustumilowy rajd do Illinois! Intuicja? Ośnienie? Nikt nie wie, w jaki sposób genialne rozwiązania pojawiają się w umysłach uczonych. Przypadek Atanasoffa jest jednym z najbardziej niezwykłych. I nie ma znaczenia fakt, że tak NAPRAWDĘ to pierwsi byli Brytyjczycy. Sędzia sądu okręgowego w Minneapolis E.A. Larsen też zresztą o tym nie wiedział.

Marek Czarkowski

