

ATARI - COMMODORE - ZX SPECTRUM - AMSTRAD CPC - AMIGA - APPLE

RIETROKOMP

Mikrokomputery wczoraj i dziś





Spełnienie marzeń

Pamiętam dobrze rozmowy, które prowadziłem 20 lat temu. Było to w czasach, gdy na rynek miały wejść procesory Transmeta i rozpisywano się o ich hipotetycznych możliwościach. Mówiono, że mają mieć możliwość programowej zmiany "wsadu", dzięki czemu ten sam sprzęt będzie mógł pełnić rolę Amigi, Atari czy ZX Spectrum. Oczywiście po odpowiednim dopracowaniu sterowników i reszty oprogramowania. Wtedy wydawało mi się, że jest to iście kosmiczna technologia, bo przecież uruchomić programy z różnych platform "bez emulacji" było nie do pomyślenia.

Dzisiaj mamy urządzenia oparte o technologię FPGA, które realizują te same funkcje. Możemy też uruchomić sprzęt z procesorem ARM, który może zachowywać się podobnie, choć jest to wykonywane w inny sposób. Obsługa ograniczona jest do minimum i od razu możemy cieszyć się w pełni funkcjonalnym, choć w pewien sposób "udawanym" komputerem. Praktycznie plotki o możliwościach platformy Transmeta zostały wcielone w życie. Z tego stanu rzeczy powinniśmy cieszyć się jak nigdy wcześniej, bo przecież mało kto ma na tyle dużo samozaparcia, aby w swojej kolekcji mieć kilkanaście czy kilkadziesiąt maszyn. W zamian sta-

wiamy małe "pudełko", na którym uruchamiamy oprogramowanie dla większości lub wręcz wszystkich komputerów jakie nas interesują.

Jednak wraz z upływem czasu inaczej patrzymy na swoje własne wybory sprzed lat. Dzisiaj nie wystarczy uruchomić program czy grę, chcemy mieć idealną jakość obrazu, najlepszy joystick albo też... sfatygowany joystick używany w "dawnych czasach" i koniecznie telewizor kinoskopowy zapewniający zniekształcenia obrazu.

Co kraj to obyczaj, chciałoby się powiedzieć. Dzieli nas spojrzenie na sprzęt retro, poszczególne platformy i oprogramowanie, ale łączy nas dużo więcej. Na przykład chęć poznawania nowych możliwości, a w tym swój udział ma zarówno oryginalny sprzęt, emulatory, jak i implementacje w technologii FPGA. Jesteśmy jedną wielką rodziną.

Tym optymistycznym akcentem zapraszam Was do lektury jubileuszowego, dziesiątego numeru magazynu. Tym razem trochę się zmieniany, bowiem zaczynamy pisać o systemie RISC OS oraz Raspberry Pi. Jest to doskonałe urządzenie dla wszystkich osób, które chcą poznać smak dobrze działającego oprogramowania dla Commodore czy Atari bez konieczności wydawania większych kwot na sprzęt oraz pokonywania trudności w jego konfiguracji. Nie jest to rozwiązanie idealne, ale zdecydowanie bardziej zachęcające niż "zwykłe" emulatory. Dlatego będziemy o nim pisać szerzej, także w kolejnych numerach RetroKompa.

Oprócz tego, numer jak zwykle wypełniony jest treścią użytkową, listingami oraz poradami jak wykorzystać swój ulubiony sprzęt w lepszy lub bardziej nietypowy sposób. Mało reklam, za to dużo tekstu, z którego można skorzystać od razu po uruchomieniu Basica, Asemblera lub tylko cartridge'a z grą. Szczególnie polecam dział Apple, który w naszych warunkach jest nietypowy, bo raczej mało kto miał kiedyś 8-bitowe modele typu Apple II. A jeśli zmęczycie się wpisywaniem programów, zajrzyjcie do artykułów o grach, które pokazują nietypowe możliwości między innymi komputerów Amstrad CPC.

Adam Zalepa

RETROKOMP

Mikrokomputery wczoraj i dziś

NUMER 10/2018
(maj)

ISSN 2450-5862

Redaktor naczelny: Adam Zalepa
retrokomp@amiga.net.pl

Autorzy tekstów:

Mateusz "Tfardy" Eckert
Krzysztof Kłiś
Piotr „Piter” Krużycki
Marcin Libicki
Piotr "Kroll" Mietniowski
Michał "stRing" Radecki-Mikulicz
Piotr "Sachy" Sachanowicz
Marcin Skawiński
Kamil Stokowski
Mariusz Wasilewski

Projekt okładki:

Marzena Bukowska

Korekta: Renata Gralak
Skład: Andrzej Wilczyński

Wydawca:

"A2" Aleksandra Zalepa, Łódź

Wszystkie nazwy oraz znaki handlowe należą do ich właścicieli i zostały użyte wyłącznie w celach informacyjnych.



RETROKOMP 10:

Spełnienie marzeń
Doniesienia

INTEGRACJA 8

Giełda Pixela
Raport z JagNES Fest Level 2
Nowa książka od Retronics - Poradnik programisty

ZX SPECTRUM 16

Czterdzieści lat minęło - Sinclair Basic
Warunki i porównania

AMSTRAD 22

Instalujemy ParaDOS
Obudź w sobie bohatera - gry przygodowe

COMMODORE 27

W cieniu wielkiego brata - VIC-20
Efekty w Hardtrack Composer
Black Box i Asembler

ATARI 40

Grafika w trybie zerowym
Poprawiamy systemowy Reset
Kolory i dźwięki
Wykopaliska
Czasomierz inaczej
Gdy Atari nie słucha - błędy w Basicu

APPLE 54

Uniwersalne wprowadzanie danych
Konflikty pamięci graficznej
Specjalne funkcje klawiszy
Crisis Mountain

ACORN 63

Wprowadzenie do systemu RiscOS
Pierwsze kroki na Raspberry Pi (cz. 1)

AMIGA 75

IOMega Zip
Workbench i 24-bity
Ukryte funkcje karty graficznej

Co nowego?

Sterowniki Pendrive dla Netusbee

Od kwietnia bieżącego roku, dostępne są już sterowniki do urządzenia o nazwie Netusbee. Każdy użytkownik dużego „Atari” mający podłączonego Atari do Internetu, wie o czym piszę. Urządzenie to oprócz możliwości podłączenia komputera do sieci posiada dwa złącza USB.

Niestety z dobrodziejstw Netusbee mogli dotychczas korzystać jedynie użytkownicy systemu MiNT, do tego wyłącznie posiadacze komputera Atari Falcon z CT60. Sterowniki są do ściągnięcia ze strony autora dla systemu TOS dla każdego Atari 16/32-bit. Urządzenie działa całkowicie Plug and Play.

Po zainstalowaniu i uruchomieniu komputera, wystarczy włożyć pendrive, a system wykryje go automatycznie i będzie widziany jako dodatkowy napęd. Wystarczy przygotować odpowiednią partycję np. w HDDriverze, a umożliwi to bardzo łatwe przenoszenie danych między PC a Atari.

Szybkość odczytu zależy od komputera, sprawdziliśmy to dla Atari ST - ok. 35 KB/s, natomiast na TT rezultat wynosi ok. 150 KB/s, Falcon z akceleratorem CT6x działa szybciej - 300 Kb/s.

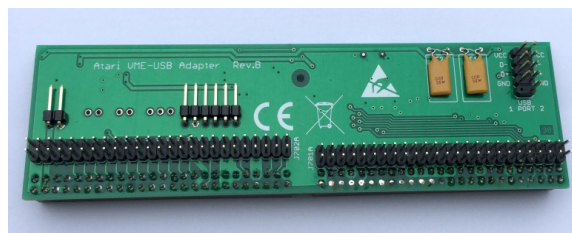
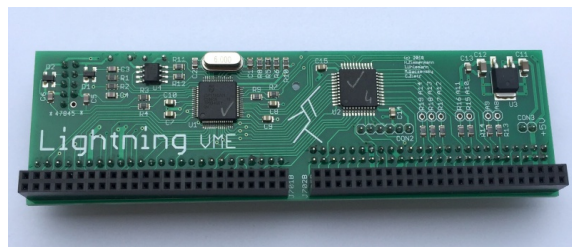
Biorę czynny udział (Kroll - przyp. red.) w fazie testów, powstały także sterowniki dla systemu Magic. Ponadto sterowniki umożliwiające podłączenie myszki i klawiatury na USB są już w przygotowaniu.

USB dla komputerów Atari MegaSTE i TT

Od jakiegoś czasu można było wyczytać na różnych forach informacje, że dla MegaSTE oraz TT powstaje podobne urządzenie umożliwiające podłączenie np. klawiatury, myszki czy pendrive na złączu USB. Więcej informacji można znaleźć na stronie projektu:

http://wiki.newtosworld.de/index.php?title=Lightning_VME

Urządzenie podłącza się bezpośrednio między płytą główną a złączem VME. Oczywiście nie przeszkadza to użytkownikom korzystającym np. z kart graficznych. Niestety nie znana jest jeszcze cena, ale w chwili obecnej są już dostępne sterowniki umożliwiające podłączenie myszki, klawiatury i wspomnianego pendrive.



Ze strony projektu można przeczytać, że szybkość odczytu jest o wiele większa. I tak - dla Mega STe jest to ok. 360 Kb/s, a dla TT - ok. 600 KB, czyli wydajność jest 4 razy większa.

Więcej informacji w kolejnym numerze, gdy będę miał możliwość dokładnego przetestowania tego urządzenia.

Nowy HDDriver wersja 10.12

Ukazała się po raz kolejny najnowsza wersja sterownika pamięci zewnętrznej dla naszych Atari. Dla zarejestrowanych użytkowników wersji 10.x dostępna jest za darmo ze strony autora.

W tej wersji, poprawiony błąd w emulacji SCSI dla napędów IDE, związany z obsługą nieprawidłowych jednostek LUN, ponadto mamy kosmetyczną zmianę w interfejsie programu. Więcej informacji można znaleźć na stronie projektu. UWAGA! Od 30 marca br. strona zmieniła adres na:

<https://www.hddriver.net>

WAPNIAK 2018

Zbliża się najnowsze party dla 8-bitowych komputerów Atari. Od kilku dni, po wielu tygodniach niepewności i problemach z miejscówką, wiemy, że cykliczne party o nazwie WAPNIAK odbędzie się w terminie 1-3 czerwca.

Po raz pierwszy zmianie uległo miejsce, bowiem party odbędzie się w ROD Bemowo II, przy ul. Księżycowej 1A. Zbliża się ono szybkimi krokami. Szczegóły należy śledzić na forum Atari oraz przez FB.

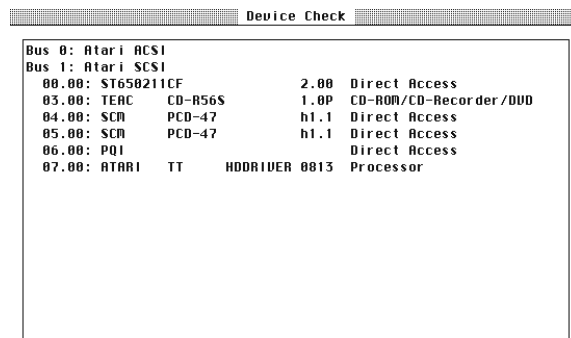
Ironia 2018

Informacja dosłownie z ostatniej chwili: czwarta edycja Ironii, zlotu użytkowników retrokomputerów Atari, odbędzie w tradycyjnym miejscu - Góra Św. Anny w terminie 3 – 5 sierpnia. Więcej informacji będzie można znaleźć oczywiście w Internecie na forum Atari oraz FB.

Time Pilot na małe Atari

Kilka dni temu ukazała się finalna wersja tej długo oczekiwanej gry - znanej przede wszystkim z automatów arcade - pod tytułem Time Pilot.

Wcześniejsze wersje pojawiały się tu i ówdzie, ale autorzy postanowili wzbogacić grę o możliwość wykorzystania karty Rapidus i zrobili to wzorcowo! Teraz mamy wersję finalną dla wszystkich.

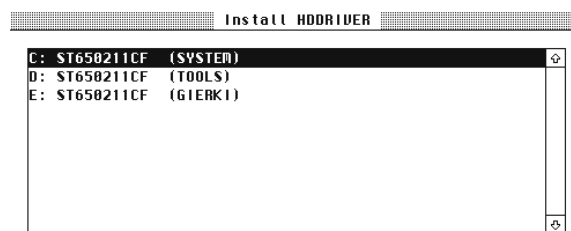


Press ESC to stop

Clipboard

Again

Cancel



OK

Cancel



Gra działa na standardowym Atari z 64 KB pamięci RAM, a dodatkowo, jak podają autorzy, ładuje się z każdym SIO (a przynajmniej powinna), automatycznie rozpoznaje kartę Rapidus i skaluje silnik pod lepszy procesor. Po szczegółowe informacje wraz z możliwością pobrania odsyłam na stronę projektu:

timepilot.atari.pl

Gra Train na Atari 8-bit

Podczas Atariady 2017 można było przeczytać o powstaniu nowej grupy BAHA Software, którą założyli Michael „Baktra” Kalous oraz Petr „HardCore” Postava. Właśnie w czasie tegorocznej edycji imprezy opublikowali oni swoją nową prostą grę Train, będącą konwersją pecetowej darmowej produkcji Vlak (czyli pociąg) z 1993 roku autorstwa Miroslava Němečka. Train jest klonem klasycznej gry „w węża”. Naszym zadaniem jest zebranie wszystkich przedmiotów na planszy i załadowanie ich do wagonów. Każdy zebrany towar wydłuża nasz pociąg o jeden wagonik. Po zebraniu kompletu przedmiotów pozostawionych na planszy otwiera się brama, przez którą możemy przejść do następnego poziomu (jest ich łącznie 50).

Utopia na... Atari 2600!

Ta gra naprawdę się ukaże, ale pod nazwą Distopia. Więcej informacji szukajcie w kolejnym RetroKompie. Na razie prezentujemy pierwsze ekrany z gry.

Paski Grozy wygrywają

Na ostatnim Global Game Jam, w gdańskiej lokalizacji spotkali się: Krzysztof „KK” Kluczek, Mariusz „Jelcyn” Górski oraz Michał „TraQ” Żuchowski i... wygrali ten konkurs wybierając do tego platformę Atari 2600. Gra napisana jest w assemblerze i traktuje o aktualnej sytuacji politycznej w naszym kraju. Jest zdecydowanie warta uruchomienia, namawiam nawet do użycia emulatora VCS, bo przecież nie wszyscy mają oryginalny sprzęt w domu. Brawa dla autorów!

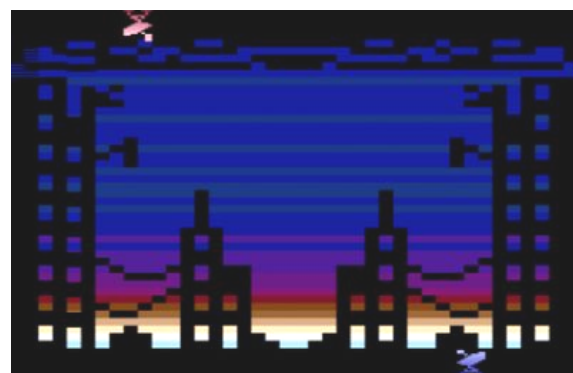
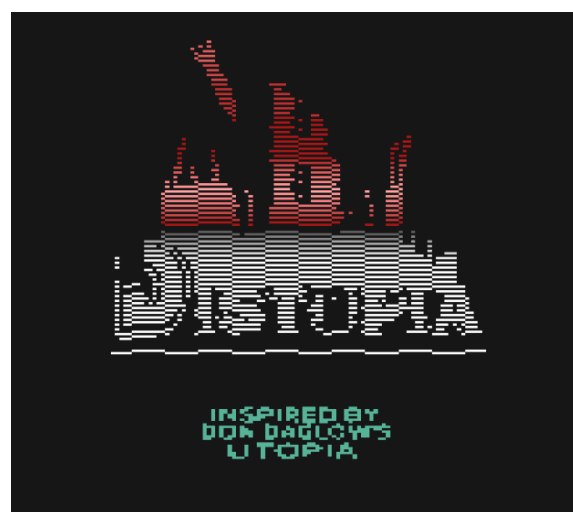
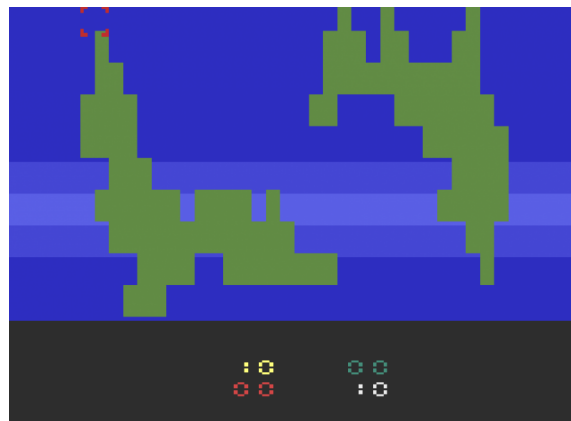
Koniec Calamusa

Ulf Dunkel, szef Invers Software zdecydował, że z dniem 31 marca zamyka działalność. W związku z tym kończy się kilkadziesiąt lat żywota flagowego narzędzia DTP na Atari. Tą smutną wiadomością, pogrążeni w żalu żegnają się:

Piotr „Kroll” Mietniowski oraz Piotr „Piter” Krużycki

ale nie kończymy pracować na Calamusie!

Przygotował: Piotr „Piter” Krużycki



Giełda Pixela

14 kwietnia w klubie River Remont w Warszawie odbyła się trzecia już w tym roku edycja Giełdy Pixela, tym razem połączona ze spectrumowym Speccy Party 2018.1.

Wstęp na imprezę kosztował 20zł, pod warunkiem wcześniejszego zakupu biletu przez Internet. Kupno wejściówki na miejscu wiązało się już niestety z wydatkiem 30zł. Dzieci pod opieką osób dorosłych wchodziły na giełdę za darmo. W cenie biletu wstępu otrzymywaliśmy kupon na dowolny, wybrany przez siebie archiwalny numer Pixela, który można było zrealizować na stoisku magazynu.

Na tym samym stoisku można też było zakupić pojedyncze egzemplarze numerów specjalnych gazet wydawanych z okazji kolejnych edycji Pixel Heaven: Bajtka, Commodore & Amiga i Top Secret. Była ich jednak bardzo ograniczona ilość.

Na miejsce trafiłem bez większego problemu. Klub zlokalizowany jest w centrum miasta, przy skrzyżowaniu ulic Ludwika Waryńskiego i Lecha Kaczyńskiego. Można do niego bez

problemu dojechać zarówno komunikacją miejską, jak i metrem.

Ze względu na to, że do Warszawy jechałem pociągiem, wybrałem tę drugą możliwość. Po wyjściu ze stacji Politechnika przy budynku Zebra Tower od razu zauważyłem po drugiej stronie ulicy plakaty i stojaki z napisem PIXEL. Organizatorzy wydarzenia zadbali o to, żeby potencjalni goście nie musieli zbyt długo szukać wejścia na giełdę.

W środku do dyspozycji wystawców i zwiedzających oddano dwa poziomy: -1 i -2. Niestety na tym pierwszym oświetlenie nie było zbyt mocne, co powodowało, że panował tam półmrok. Z drugiej strony dzięki temu bardzo dobrze oglądało się wyświetlane na projektorze prezentacje. Miąłem okazję obejrzeć dwie z nich.

Pierwszą był wykład Rafała Wiosny o tym, jak w ZX-81 tworzony jest obraz wideo. Stanowił on dla mnie niemałe



COBRA 1 - polski komputer do samodzielnego montażu.

zaskoczenie, gdyż Rafał kojarzy mi się głównie z Amigą. Drugim była prezentacja Fausta o tym, jak powstają jego Reduksy starych czasopism komputerowych.

Obydwa wystąpienia były niezwykle ciekawe, jednak ze względu na fakt, że odbywały się w tym samym pomieszczeniu co reszta giełdy, to mimo nagłośnienia trzeba było przysunąć się bardzo blisko sceny i mocno wyęźać słuch, żeby dobrze słyszeć prowadzących.

Obok sceny, na której odbywały się prelekcje, rozłożyli się Spectromowcy. Mogliśmy więc podziwiać różne modele ZX Spectrum w akcji, była również gablotka z różnym sprzętem spod znaku Sinclaira - od kalkulatorów kieszonkowych i pierwszych modeli ZX-80 i ZX-81, poprzez legendarnego "gumiaka" Spectrum 48k, aż po stworzony już po przejęciu firmy przez Amstrada model +3.

Była również niepowtarzalna okazja zagrać w Defenders of Earth na SAMie Coupe. Gra wyglądała na tyle dobrze, że niektórzy nie mogli



Stoisko ze sprzętem.

uwierzyć, że SAM to komputer 8-bitowy a nie 16-bitowy.

Na mnie jednak największe wrażenie zrobił działający egzemplarz polskiego komputera COBRA 1. Dla osób niezorientowanych w temacie krótkie wyjaśnienie. COBRA 1 była polską konstrukcją opartą o procesor

Zilog Z-80, zaprojektowaną przez Andrzeja Sirko i Grzegorza Gancarza. Jej głównym założeniem była możliwość samodzielnego zbudowania komputera w warunkach domowych, na bazie dostępnych powszechnie części elektronicznych.

W latach 80-tych było to nie lada wyzwaniem, gdyż jak zapewne większość czytelników pamięta, półki sklepowe świeciły wówczas pustkami, a na Polsce i innych krajach bloku wschodniego ciążyło embargo COCOM, mocno ograniczające import nowych technologii. Pomimo tych trudności, w latach 1984-1986 w kwartalniku Audio Video opublikowany został cykl artykułów pokazujących jak krok po kroku zbudować komputer COBRA. Obecny na miejscu imprezy egzemplarz był wersją poprawioną i nieco zmodyfikowaną w stosunku do oryginału, jednak bez problemu uruchamiał gry przeznaczone dla swojego pierwowzoru. Na tym samym stoisku można było również zobaczyć prototyp komputera COBRA 2.



Stoisko z czasopismami

Moim zdaniem jest to konstrukcja absolutnie rewelacyjna, ponieważ opiera się na 80-pinowej "płyce głównej", stanowiącej szynę danych, do której podłączamy poszczególne moduły z procesorem, pamięcią RAM, ROM, układami wejścia/wyjścia, itp.

Pod względem koncepcyjnym konstrukcja COBRY 2 mocno przypomina słynny wśród hobbystów, szczególnie w Wielkiej Brytanii, komputer RC-2014. Jest to jednak przede wszystkim projekt edukacyjny, dokładnie tak jak założyli to jego twórcy w 1984r. Trzymam mocno kciuki za projekt i mam nadzieję, że zostanie niedługo upubliczniony.

Oczywiście oprócz podziwiania sprzętu retro, można było też dokonać zakupów - w końcu na giełdzie powinno się handlować! Na poziomie, na którym odbywało się Speccy Party, można było zakupić nieco akcesoriów, starych czasopism, a także gier planszowych wydawnictwa Paladynat w promocyjnych cenach.

Jednak serce giełdy biło piętro niżej. Mogliśmy tam znaleźć mnóstwo klasycznych programów i gier na oryginalnych nośnikach, a także praktycznie każdy popularny w latach 80-tych i 90-tych komputer, włącznie z tak rzadkimi egzemplarzami, jak ZX Spectrum 128k czy Atari Falcon.

Oświetlenie również było o wiele lepsze niż w położonej wyżej sali, więc można było dokładnie obejrzeć wystawiony sprzęt. Mogliśmy też dokonać zakupu nowoczesnych akcesoriów do swoich komputerów, których próżno szukać na portalach aukcyjnych, takich jak na przykład kartridż Side 2 do małego Atari czy interfejs DDI-1 Zaxona dla Amstrada CPC-464. Ceny nie różniły się poziomem od tych, które spotkamy w Internecie - widać wyraźnie, że punk-

tem odniesienia dla sprzedających jest Allegro.

Jednak zakup na giełdzie ma jedną zasadniczą przewagę - sprzęt możemy przed zakupem dokładnie obejrzeć, a niejednokrotnie również sprawdzić. Nie ma więc obawy, że kupimy kota w worku od przysłowiowego Janusza, który wciśnie nam niedziałający złom jako "niesprawdzony", a po wysłaniu paczki przestanie odpowiadać na maile i odbierać telefon.

Tutaj co prawda też na swój zakup nie dostaniemy gwarancji jak w sklepie, ale możemy liczyć na fachowe wsparcie i pomoc w razie jakichkolwiek problemów, co z pewnością stanowi olbrzymią zaletę giełdy.

Dla osób, które chciały chwilę odpocząć, czekały wolne stoliki w górnej sali, a także siedzące miejsca przy barze, gdzie można było zamówić zimny napój, również w kolorze złocistym. Jeśli ktoś natomiast chciał zaspokoić głód, to również nie było z tym problemu. Na miejscu

można było zjeść całkiem niezły kebab, a jeżeli taki posiłek nam nie odpowiadał, w okolicy znajduje się bardzo dużo dobrych i niedrogich restauracji oraz sklepów spożywczych.

Na Giełdzie Pixela byłem pierwszy raz, ale na pewno jeszcze nie raz na nią wrócę. Jest to nie tylko doskonałe miejsce na dokonanie zakupów i uzupełnienie kolekcji, ale też świetnie zorganizowana impreza i niepowtarzalna okazja, żeby spotkać ludzi o podobnych zainteresowaniach, nawiązać nowe, ciekawe znajomości, czy chociażby chwilę porozmawiać. Połączenie giełdy i Speccy Party również moim zdaniem znakomicie się sprawdziło.

Party było nie tylko ciekawym urozmaicheniem części komercyjnej, ale dodało całej imprezie niepowtarzalnego klimatu. Giełda odbywa się regularnie i bardzo często łączy się z imprezami scenowymi - więc jeśli ktoś jeszcze na niej nie był, koniecznie musi te braki nadrobić.

Krzysztof Kliś



Ceny nie odbiegają poziomem od portali aukcyjnych.

JagNES Fest

Level 2

O imprezie JagNES Fest dowiedziałem się ze strony AtariOnLine, choć już wcześniej słyszałem o niej. W podjęciu decyzji o wyjeździe pomogły mi zapowiedzi meteorologów o zbliżającym się ciepłym długim weekendzie. To właśnie w tych dniach, czyli 1-3 maja odbyła się konsolowa „nasiadówka”.

Jako że jechałem PKP, nie zabrałem nic ze sprzętu, co okazało się dobrym posunięciem, gdyż na miejscu wszystkiego było aż nadto. Ruszałem z Mazowsza, więc podróż miałem dwuetapową, z przesiadką w Tczewie (chyba), a następnie pojechałem - tak jak mnie Jesionen poprowadził - do Zblewa. Tam już czekała „limuzyna” w postaci auta niemieckiej produkcji, która bezpiecznie zawiozła mnie na party place.

Po drodze, poznając Jesionena (nie znałem go wcześniej), miło rozmawialiśmy głównie o Playstation. Na miejscu byliśmy szybciej niż się tego spodziewałem. Od razu zrobiło się sielankowo, było południe, słońce wysoko, prawie upał, a partyzanci leniwie leżący na leżakach, czy to samotnie, czy w grupkach popijali

(zapewne) zimne piwko. Było spokojnie. Po przywitaniu się z większością osób wszedłem do niewielkiej salki wypełnionej różnego rodzaju konsolami.

Piękny widok! NESy, SNESy, jakaś spora kolekcja gier z Jaguarem CD, Amstradowa konsola, którą wcześniej na oczy nie widziałem, a także mini stanowisko z handheldami różnej maści.

Z komputerów zauważyłem jedną Amigę 600 i Atari ST, ale na nich także odpalone były gry, więc nie psuło to całości - konsolowo-growej atmosfery.

Na sali było mało osób, ale pewnie ze względu na fakt, iż pogoda nie sprzyjała siedzeniem przed maszynami, a wyjściem na zewnątrz lub na spacer. Okolice są przepiękne, dookoła lasy, przejeżdżaliśmy też obok kilku jezior. Ogólnie świetny klimat. Zostałem zakwaterowany w przyczepie kempingowej, gdzie było sporo miejsca i miałem blisko do party place.

Do wieczora czas spędziłem na rozmowach, m.in. z Iratą4. Dowiedziałem się też, że większość osób jest tu już kilka dni. Faktycznie, termin majówki w tym roku pozwalał na zrobienie sobie naprawdę długiego weekendu.



Rodzinne granie na konsolach.



Na imprezie były dostępne konsole różnej maści.

Wieczorem nastąpiło oficjalne otwarcie, rozdanie koszulek, a także wyświetlenie invitek, wcześniej przygotowanych i publikowanych na stronach o retro sprzętach, z tego co pamiętam m.in. na AtariOnLine.

Następnie kolega Rafał RJ1307 rozpoczął swoją prezentację konsoli Amstrad GX4000. Na początku wszyscy sądzili, że będzie to po prostu opis konsoli, lecz okazało się że jest on elektronikiem i dokonał sporych przeróbek i udoskonaleń tej maszyny, co doprowadziło do tego, iż stała się ona niemalże komputerem Amstrad CPC z możliwością podłączenia klawiatury i karty SD.

Bardzo ciekawy wykład trwający chyba 2 godziny zakończył się prezentacją na tej właśnie konsoli kultowego już (mimo że powstało ono w 2011 roku) dema pod tytułem „Batman Forever”, pierwotnie uruchamianego na ciężko dostępnym, a co za tym idzie - drogim sprzęcie CPC. Po wykładzie nastąpiły luźne formy integracji połączone ze spożyciem różnych napojów, których łączył jeden wspólny mianownik – mianowicie „%”.

Na drugi dzień wieczorem musiałem powoli wracać do domu, ale w ciągu dnia zrobiliśmy sobie jeszcze wycieczkę nad jezioro oraz do pobliskiego, oddalonego o 3 km, pubu czy raczej baru, w którym stały automaty arcade! Co za klimat, tam czas się zatrzymał się na latach '80-tych czy '90-tych. Zjedliśmy frytki, pograliśmy i wrócimy na party place. Wieczorem Jesionen ponownie odwiózł mnie na dworzec w Zblewie, skąd po kilku godzinach trafiłem do Warszawy. Bardzo żałuję, że nie zaplanowałem wyprawy na kilka dni, bo nie zdążyłem ze wszystkimi rozmawiać, a także wypróbować wszystkich sprzętów.

Za rok, jeśli JagNES Fest się odbędzie, zarezerwuję sobie więcej czasu, bo naprawdę warto. Spędziłem miło czas wśród ludzi pozytywnie zakręconych na punkcie konsol, było także kilka znanych osób ze środowiska demoscenowego, co mnie bardzo ucieszyło. Szczerze polecam to party wszystkim, którzy lubią las, kiełbasę z ogniska, musztardę oraz granie na konsolach. Do zobaczenia (mam nadzieję) za rok!

Mariusz Wachowicz

JAGNES



Nie zabrakło czysto jaguaro-
wych zbiorów.



Atari Lynx MK1 w rękach MotionRide'a.

Daewoo DPC-200

Jednym z dość zaskakujących producentów komputerów w historii była firma Daewoo, u nas znana głównie od strony motoryzacyjnej. Gdy sprawdzimy, że produkowała także klony komputerów MSX w pierwszej połowie lat '80-tych, zaskoczenie może być zdecydowanie większe.

Oczywiście sprzęt był przeznaczony na rynek koreański, ale tytułowy model 200 nie był pierwszym w serii. Wszystko zaczęło się od komputera DCP-100, który został wyposażony w 16 KB pamięci ROM, tyle samo pamięci RAM i dialekt języka Basic o egzotycznej nazwie Hangul. Była to wersja dostosowana do warunków w Korei. Powstało kilka wersji, a ostatnia została wydana jeszcze w 2015 roku, co jest prawdziwą ciekawostką.

Nowsze wersje Hangul Basica przede wszystkim rozszerzają instrukcję SCREEN o możliwość uruchomienia trybu z obsługą znaków koreańskich (SCREEN 9). Jednak programiści zachowali zgodność „w dół”, dlatego tej wersji Basica można także używać na nie-koreańskich komputerach.

Kolejne modele komputerów DCP o oznaczeniach 180 i 200 różniły się ilością pamięci – rozszerzoną ją do 32 i 64 KB, a także bardziej

nowoczesną obudową. Do Basica wprowadzono też szereg nowych instrukcji, które są związane z obsługą specyficznych cech nowej serii komputerów.

Model DCP-200 był sprzedawany nie tylko w Korei, lecz także w wielu europejskich krajach, na przykład w Hiszpanii, Francji, Niemczech, Włoszech i Wielkiej Brytanii. W związku z tym przygotowano specjalne edycje dla tych krajów, a sprzęt był sprzedawany pod innymi symbolami, jak MCP-80 czy Bawareth Perfect.

Do Polski komputerowe produkty Daewoo nie dotarły oficjalnie i przyznam, że w latach '80-tych Korea była pod tym względem dla mnie pra-

wdziwą „biała plamą”. Jak się jednak okazuje, także w tym regionie świata starano się komputeryzować świat, a firma Daewoo wydała szereg oprogramowania w latach 1987-1990. Ponadto produkowany przez pewien czas chipset zgodny ze standardem Intel 8255, a komputery DCP były przeznaczone do rozbudowy.

W późniejszym okresie firma starała się wypromować za pomocą komputerów edukacyjnych, na przykład modelu CPC-330K, który posiadał już całe 128 KB pamięci. Historia zakończyła się na przełomie lat 1989-1990, kiedy wydano ostatnie aktualizacje oprogramowania i zaprzestano produkcji klonów w standardzie MSX.

Mariusz Wasilewski



Nowa książka od Retronics

Firma Retronics nie zaprzestaje działalności wydawniczej. Kolejną pozycją w jej ofercie jest reedycja książki Wojciecha Zientary pod tytułem „Poradnik Programisty Atari”.

Do książki dorzucono także dodatek w formie poradnika o tym, jak napisać 1KB grę w assemblerze. Autorem dodatku jest Marcin Sochacki, znany pod pseudonimem Montezuma. Oryginał, wydany w 1987 roku, był drukowany na słabej jakości papierze i posiadał miękką okładkę. Obecne wydanie to dużo lepsza jakość. W numerze 7 magazynu Retrokomp opisaliśmy różnice pomiędzy starym wydaniem innej książki „Assembler 6502” wydawnictwa SOETO, a jej wznowieniem od Retronics.

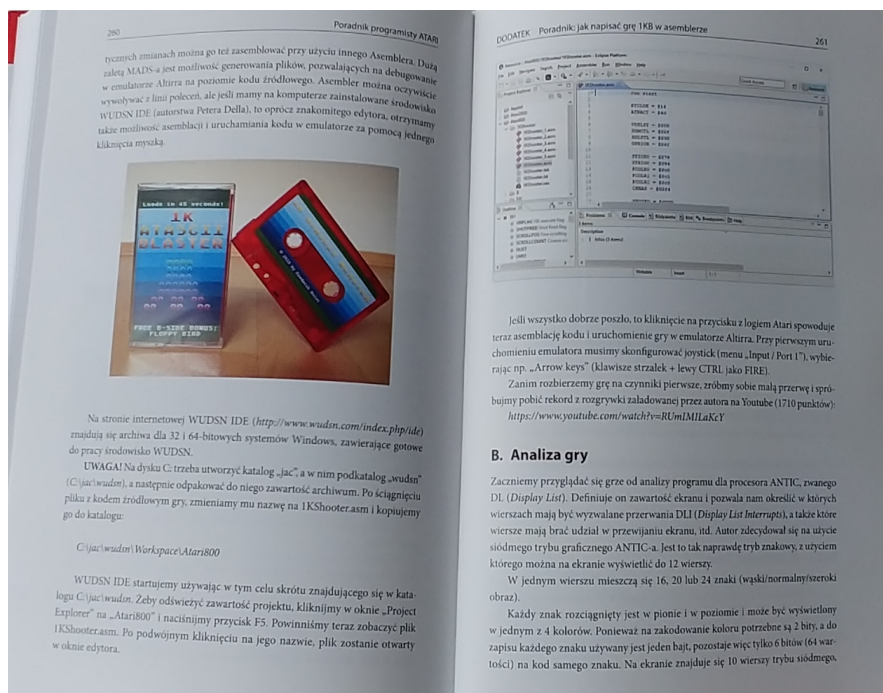
W przypadku „Poradnika Programisty Atari” sytuacja się powtarza i tak jak poprzednio otrzymujemy profesjonalnie wydaną książkę z twardą laminowaną okładką i zaokrąglonym grzbietem, papier kredowy i sztywne strony. Tutaj również poprawiono skład, wyeliminowano błędy i stworzono na nowo tabele i rysunki. Dodatek, który zawiera 50 stron, jest wydrukowany w kolorze. We właściwej części książki znajdziemy kolejno rozdziały o programowaniu w Basicu,

oraz w języku maszynowym, a także o programowaniu grafiki i dźwięku, oraz opis urządzeń zewnętrznych komputera.

Ta część stanowi solidny zasób wiedzy, wraz z przykładami i jest odpowiednia dla każdego użytkownika, zarówno początkującego, jak i zaawansowanego. Jak czytamy we wprowadzeniu: podręcznik stanowi kompendium wiedzy o językach programowania, a zadaniem książki jest głównie pomoc w rozwiązywaniu problemów pojawiających się podczas pisania programów. Nie należy

więc traktować tego poradnika jako podręcznika do nauki języka Basic, czy języka maszynowego.

Napiszmy kilka słów na temat dodatku Marcina, który w naszej ocenie jest najciekawszą częścią tego wznowienia. Na konkurs programistyczny Abbuc w 2016 roku została zgłoszona gra 1K ATASCII Blaster. I chociaż zajęła 7 miejsce to stanowiła swego rodzaju wyzwanie, ponieważ jej autor Frederik Holst postanowił zmieścić swoją grę w jednym kilobajcie pamięci.

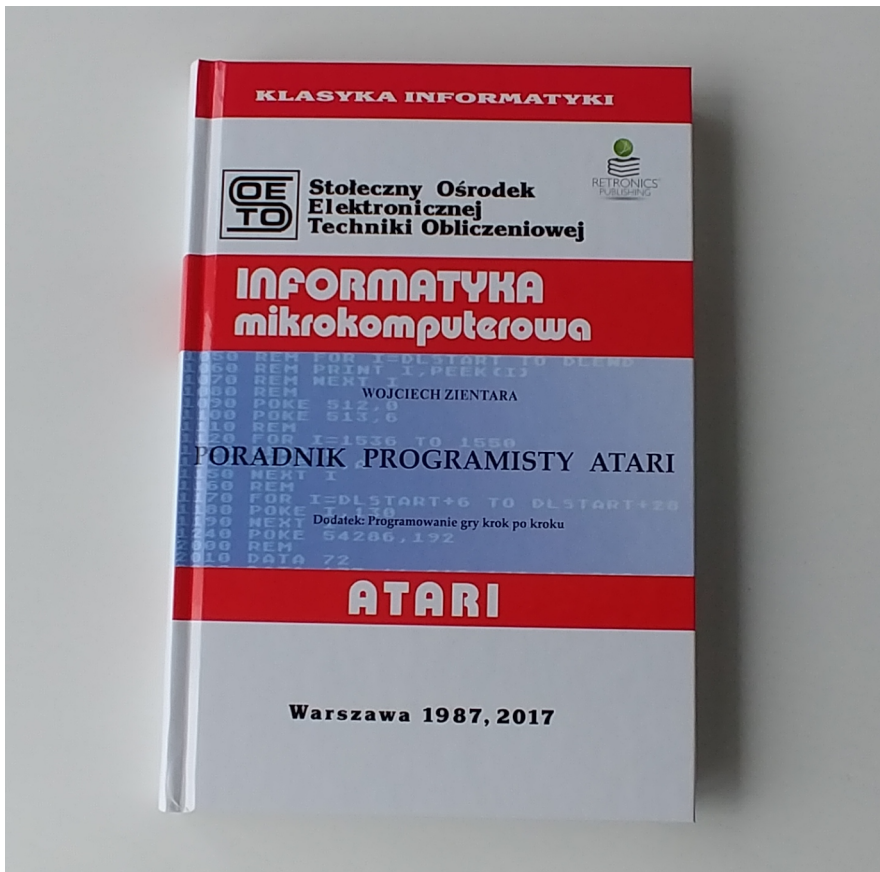


B. Analiza gry

Zacznijmy przyglądać się grze od analizy programu dla procesora ANTIC, zwanego DL (Display List). Definiuje on zawartość ekranu i pozwala nam określić w których wierszach mają być wywołane przerwy (menu, Input / Port I), a także które wiersze mają brać udział w przewijaniu ekranu, itd. Autor zdecydował się na użycie siódmego trybu graficznego ANTIC-a. Jest to tak naprawdę tryb znakowy, z użyciem którego można na ekranie wyświetlić do 12 wierszy.

W jednym wierszu mieszczą się 16, 20 lub 24 znaki (wąski/normalny/szeroki obraz).

Każdy znak rozciągany jest w pionie i w poziomie i może być wyświetlony w jednym z 4 kolorów. Ponieważ na zakodowanie koloru potrzebne są 2 bity, a do zapisu każdego znaku używany jest jeden bajt, pozostaje więc tylko 6 bitów (64 wartości) na kod samego znaku. Na ekranie znajduje się 10 wierszy trybu siódmego.



Książka wydana jest w solidnej, twardej oprawie.

Dodajmy, że sama gra została wydana także na kasecie magnetofonowej. Marcin poddał analizie kod gry (którego źródła dostępne są na stronie autora:

<http://phobotron.de/1K-Shooter.asm.txt>

oraz opisał szczegółowo kolejne jego fragmenty, wyjaśniając jak działają poszczególne rozkazy i co w danym momencie wykonuje program. Wszystko opisane jest bardzo czytelnie, a tekst uzupełniono wieloma ilustracjami i schematami blokowymi.

Dodatek podzielony został na podrozdziały takie jak: Display List dla procesora ANTIC, opis przerw i generowanie dźwięku, grafika graczy i pocisków, strzelanie, oraz przewijanie ekranu gry. Taki poradnik to idealna pomoc, dla tych którzy chcą



rozpocząć programowanie w assemblerze. Mając namacalny produkt – gotową grę 1K ATASCII Blaster, możemy na nim swobodnie eksperymentować i go testować, zmieniając i rozbudowując jego kod o dodatkowe elementy, jednocześnie ucząc się programowania i tego jak działa komputer Atari.

Uważamy, że początkującym programistom łatwiej jest zrozumieć pewne działania i zależności na krótkim listingu. Jednocześnie pro-

gram jest jak na swój rozmiar dość rozbudowany pod względem zawartości jak i możliwości (mamy tu zarówno kolizje, oraz strzelanie, ruch statku, przesuw ekranu, a nawet proste dźwięki), przez co można się z niego wiele nauczyć.

Na koniec dodajmy, iż firma Retronics wprowadziła właśnie do swojej oferty dodruk wznowienia książki „Assembler 6502” w cenie 95 PLN. Jest więc jeszcze szansa na zakup dla tych, dla których ostatnio zabrakło egzemplarzy. Zbliża się też reedycja wyczekiwanej książki „Atari Basic”, także w cenie 95 PLN. Jej premiera nastąpi w czerwcu 2018 roku.

Bardzo jesteśmy ciekawi, jak będzie wyglądało to wznowienie, którego wersja z 1987 roku wydana została w formacie A4, przez co jest dość czytelna i wygodna w korzystaniu. Niestety, przez częste wertowanie stron, w wielu przypadkach nietrwały papier nie przetrwał próby czasu, a okładka potrafiła odpaść od reszty.

Dodajmy, że nowa książka zostanie wydana z opcjonalnym dodatkiem w postaci kasety magnetofonowej, na której znajdą się nagrane programy, których listingi zawarte są w tejże książce. Jak zapewnia producent, dane ze starej kasety zostały poddane remasteringowi, więc spodziewamy się, że jakość nagrania będzie porównywalna z jakością jaką oferują nam reedycje gier - także od Retronics.

Michał Radecki-Mikulicz

Zapraszamy na stronę wydawcy, na której można znaleźć także kolekcjonerskie edycje gier oraz pokrywę przeciwkurzową na nasze komputery:

<http://retronics.eu>



Czterdzieści lat minęło

Basic był podstawowym "trybem pracy" dla każdego, kto zdobył ZX Spectrum. Pisząc nawet proste programy można było zorientować się, że nie jest to demon szybkości. Ba, skutecznie zwolnić komputer można po prostu wciskając... jeden klawisz. Poczekajmy, aż ekran zostanie wypełniony większą ilością znaków. Różnica jest mocno zauważalna, delikatnie mówiąc. Niezależnie od wielu wad Basica, warto zastanowić się nad jego historią, bowiem sięga ona okresu, w którym u nas nie można było mówić o jakimkolwiek rynku komputerowym. Musimy cofnąć się do lat '70-tych.

Sinclair BASIC został oryginalnie napisany dla komputerów ZX-81 oraz ZX-80. Później pojawił się na innych maszynach. Ciekawostką jest fakt, że już w latach '80-tych pisano o możliwościach uruchomienia programów za pomocą emulacji, na przykład na Commodore 64. Jednak historia Sinclair Basica jest ciekawa i wbrew pozorom, dość skomplikowana.

W lipcu 1975 roku, firma Micro-Soft (tak się wówczas nazywała) stworzyła interpreter Basic dla komputera Altair 8800. Całość mieściła się w ok. 4 KB pamięci i była zapisana na taśmie perforowanej. Była to pierwsza komercyjna wersja języka programowania dla początkujących, oparta na Dartmouth Basic.

Dodajmy, że pierwotnie został on opracowany przez J.G. Kemeny'ego i T.E. Kurtza w 1964 r. w Stanach Zjednoczonych. Język spotkał się od początku z krytyką, co nie dziwi również dzisiaj. Mówiono, że zachęca to złych nawyków programistycznych, a także o nieprawidłowej strukturze programów.

Wersja Micro-Softu stała się tak popularna, że uczyniła założycieli, czyli panów Gatesa i Allena milionerami. W późniejszych latach MS Basic był instalowany na większości 8-bitowych komputerów.

W maju 1979 roku inżynierowie Clive Sinclaira rozpoczęli pracę nad maszyną, która ostatecznie stała się komputerem ZX80. Sinclair został zainspirowany do jego stworzenia po tym, gdy zobaczył, jak bardzo jego syn cieszył się z używania TRS-80, czyli wyprodukowanego w 1977 roku pierwszego komputera domowego amerykańskiej firmy Tandy Radio

Shack. Jednak wielu ludzi było zniechęconych do zakupu ze względu na wysoką cenę wynoszącą prawie 500 funtów. Oczywiście przykładając to do ówczesnych polskich warunków, była to fortuna.

Sinclair chciał obniżyć koszty, a także uniknąć opłat licencyjnych w stosunku do Microsoftu. Dlatego spotkał się z Johnem Grantem, szefem firmy Nine Tiles, aby omówić wymagania oprogramowania dla komputera ZX-80. Planem było zastosowanie Basica w standardzie ANSI.

Biorąc pod uwagę mały budżet na badania i rozwój, firma Nine Tiles nie była w stanie zarobić na projekcie dużych pieniędzy, ale uznano, że jest wart zachodu i przyniesie firmie zysk pozafinansowy.

Aby osiągnąć cenę startową w wysokości 79,95 funtów za zestaw, pamięć RAM w ZX-80 była ograniczona do 1 KB, a cały Basic musiał zmieścić się w pamięci ROM o rozmiarze 4 KB. Grant napisał większość ROM-u od czerwca do lipca 1979 roku. Jednak powstały program miał 5 KB długości, więc Grant uzyskał ostateczną wersję dopiero w sierpniu, gdy dokonał optymalizacji kodu.

Trzeba powiedzieć, że Grant wyznaczył ścieżkę dla przyszłych wy-

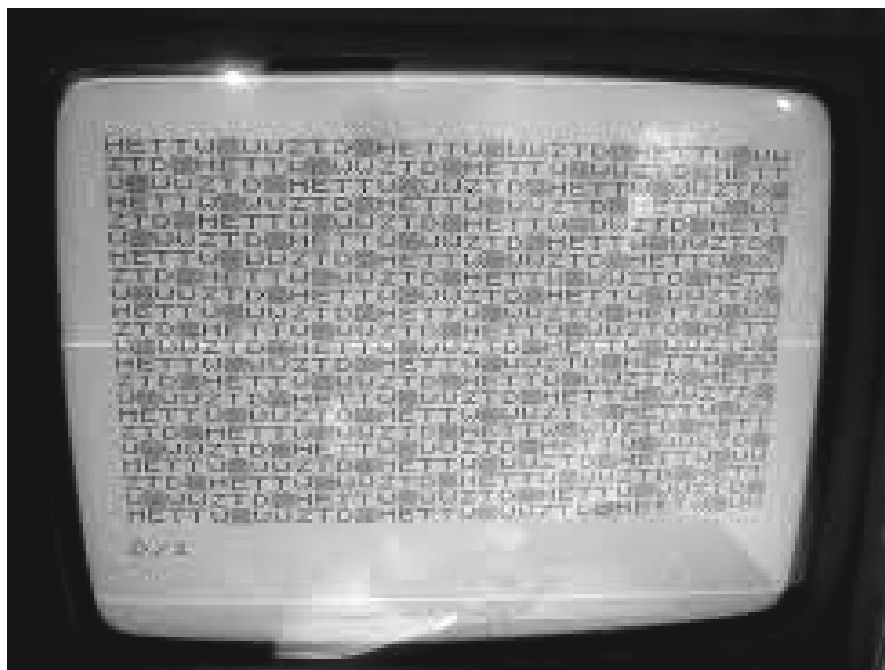
darzeń, wprowadzając wiele wyjątkowych cech do Sinclair Basica, między innymi sposób wskazywania błędów składniowych, co spowodowało, że uczenie się tego języka było łatwiejsze niż u konkurencji. Myślę, że każdy posiadasz ZX Spectrum wie, co mam na myśli.

Komputer został zaprezentowany na targach komputerowych w Wembley, w pierwszym tygodniu lutego 1980 roku. Nie okazał się wielkim sukcesem w porównaniu z późniejszym ZX Spectrum, ale szybko okazało się, że sprzedał się na całkiem dobrze, co przekonało Sinclaira do opracowania kolejnego komputera, czyli ZX-81.

W zasadzie prace nad sprzętem rozpoczęto we wrześniu 1979 roku, czyli jeszcze przed wprowadzeniem na rynek ZX-80. Jednak początkowo ograniczyło się to do stworzenia układu logicznego ULA. Pozwoliło to po raz kolejny zmniejszyć koszty, dzięki czemu komputer mógł być oferowany w cenie 49,95 funtów.

Firma Nine Tiles została znowu poproszona o dostarczenie nowej wersji Basica, ale tym razem do dyspozycji było aż 8 KB pamięci. Steve Vickers, który dołączył do Nine Tiles w styczniu 1980 roku, dodał funkcje arytmetyczne, trygonometryczne i inne. Sinclair chciał umieścić w pamięci jak najwięcej elementów, zwracając szczególną uwagę na matematykę. Chodziło więc o rozbudowę funkcji znanych z ZX-80.

Basic został prawie ukończony do jesieni 1980 roku, ale do trzeba było jeszcze dodać obsługę drukarki. W tym czasie w programie pojawił się błąd skutkujący nieprawidłowymi obliczeniami pierwiastków. Firma szybko go naprawiła, ale mimo wszystko niektóre egzemplarze komputera posiadały pamięć ROM bez poprawek.



Tak może wyglądać błąd wywołany w Basicu.

Mimo tego problemu, ZX-81 został dobrze przyjęty i stał się ogromnym sukcesem. Stworzono wiele klonów, zarówno nielegalnych, jak i licencjonowanych przez firmę Timex, która produkowała brytyjskie modele dla Sinclaira w zakładzie w Dundee.

Sinclair z jednej strony był zadawalony z sukcesu, niestety z drugiej - nie udało mu się podpisać umowy na zaprojektowanie komputera dla BBC. Dlatego zdecydował, że wprowadzi na rynek kolejną, ulepszoną wersję swojego komputera.

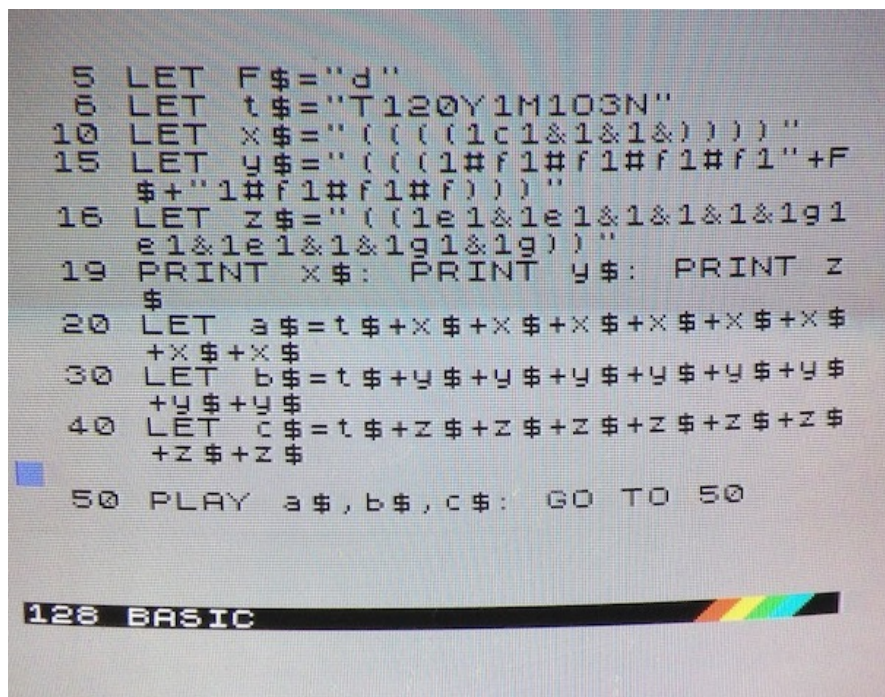
Sprzęt ZX-80 i ZX-81 był przede wszystkim dziełem jednego człowieka - Jima Westwooda. Został on jednak przeniesiony do działu związanego z telewizorami, tak więc zadanie zaprojektowania sprzętu, który stał się później ZX Spectrum, zostało przekazane Richardowi Altwasserowi.

Jednocześnie firma Nine Tiles została znowu poproszona o dostarczenie nowego Basica.

Projekt rozpoczął się jako rozszerzenie funkcji ZX-81, jednak szybko przekształcił się w duży program zajmujący całe 16 KB. Sinclair chciał jak najmniej zmian w kodzie ZX-81, ale pracownicy firmy Nine Tiles mieli przekonanie, że oprogramowanie zaprojektowane wcześniej dla maszyny z 1 KB pamięci będzie nieodpowiednie dla komputera wyposażonego w 16 KB.

Po latach w wywiadach stwierdzali, że ich zamierzeniem było przepisanie kodu, natomiast brakowało na to czasu. Oskarżano też Sinclara o pewnego rodzaju cięcie kosztów, bowiem, jak powiedział Grant w wywiadzie z 1985 roku: "Clive chciał mieć maksymalną ilość nowych urządzeń za minimalną ilość pieniędzy".

Po roku pracy Basic był praktycznie ukończony. Został znacząco rozbudowany, niestety wraz z mnogością zmian stał się także bardzo powolny. Głównym problemem stała się obsługa wielu planowanych urządzeń



Edytor w trybie 128.

peryferyjnych, ponieważ żadne działające prototypy nie były dostępne do końca 1981 roku.

W 1982 roku firma zaczęła mieć kłopoty w związku ze sporami finansowymi związanymi z opłatami licencyjnymi Sinclaira. Na domiar złego, Vickers i Altwasser złożyli rezygnację w celu utworzenia własnej firmy, Cantab, która wyprodukowała mało znany u nas komputer Jupiter Ace, który różnił się od ZX-80 głównie wbudowanym językiem Forth zamiast Basica.

Rezultatem opóźnień spowodowanych tymi problemami był fakt, że Sinclair otrzymał niekompletny ROM. Nine Tiles przez trzy miesiące, po rozpoczęciu sprzedaży ZX Spectrum, poprawiało program, ale do tego czasu została sprzedana duża ilość maszyn.

Zawartość ROM-u w zasadzie nigdy nie została ukończona według założonego planu. Sinclair chciał sprzedać ograniczoną ilość komputerów i zapewnić im aktualizację

pamięci, podobnie jak w przypadku błędu w ROM-ie ZX-81. Jednak, gdy okazało się, że zostało sprzedanych, około 75 tys. egzemplarzy, plan stał się niewykonalny. Z tego powodu na standardowej pamięci ROM nie działają na przykład polecenia do obsługi Microdrive.

Osoby zainteresowane gotowym ROMem zapraszam na następującą stronę:

<https://8bit.yarek.pl/upgrade/zx.roms/index-pl.html>

gdzie można pobrać najnowszą wersję, a także wiele innych ciekawych plików. Wersja, o której mówię zawiera między innymi obsługę portu RS-232 i funkcje sieciowe, chociaż niestety są one kompatybilne z większością komercyjnego oprogramowania Spectrum.

Na przestrzeni lat do Basica zostały wprowadzone różne ulepszenia, na przykład w 1983 roku, gdy firma Timex rozpoczęła sprzedaż komputera o oznaczeniu 2068. Wersja

ROM wciąż była niekompletna, a komputer nie był w stanie bezpośrednio uruchomić większości oprogramowania przeznaczonego dla Spectrum ze względu na inną organizację ROM.

W 1985 roku, w ramach współpracy z hiszpańskim dystrybutorem In-vestrónica, Sinclair wprowadził na rynek ZX Spectrum 128 o nazwie kodowej "Derby", z nowym edytorem Basica. Całość była bardziej kompatybilne niż wersja Timexa, ale edytor nie zawsze działał prawidłowo, a niektóre programy odmawiały pracy nawet w trybie 48K, ponieważ puste miejsce na końcu oryginalnej pamięci ROM, używane były przez niektóre programy jako obszar tymczasowy. Powodowało to nadpisywanie pamięci dodatkowym kodem.

Wprowadzono kilka nowych, przydatnych poleceń i wbudowany edytor tekstu, jednak kolejna wersja została okrojona i posiadała mniejszą ilość funkcji w menu. Trzeba jednak dodać, że programiści w tym przypadku musieli dokonać deasemblacji programu, bowiem oryginalne źródła zostały utracone.

Edytor 128 posiada więc kilka wersji różniących się między sobą, ale można je łatwo rozpoznać. W początkowej lub końcowej części ROM-u można znaleźć inicjały autorów, co przekłada się na MB, KM i AT (wersja hiszpańska) oraz MB, SB, AC, RG i KM (wersja angielska).

Według Ruperta Goodwina, autora instrukcji dla ZX Spectrum +2 i osoby odpowiedzialnej za logo Spectrum w menu, programiści Sinclaira nie zdawali sobie sprawy, że niewykorzystane bajty w oryginalnej pamięci ROM były używane jako tabela przez programistów gier. Produkt był testowany za pomocą obrazu na telewizorze oraz innego, pomocniczego kodu.

Gdy ZX Spectrum schodził z produkcji, sprawdzana była klawiatura, magnetofon, porty, kolor oraz dźwięk. Większość kodu testowego znalazła się w pamięci ROM, bo było to najtańsze i teoretycznie najszybsze rozwiązanie.

Było też kilka "dziwnych" funkcji, które zostały zaplanowane dla modelu 128, ale zostały usunięte przed produkcją, ponieważ nie mogły działać prawidłowo. Planowano wprowadzić więcej funkcji związanych z klawiaturą, a także obsługę myszki. Nad kodem edytora dla 128 pracował też Kevin Males, który twierdził, że pracował nad wieloma fragmentami, które finalnie nie trafiły do ROM-u, na przykład funkcjami związanymi z obsługą dźwięku i Microdrive. Ponadto pracował nad zautomatyzowanym oprogramowaniem testowym i diagnostycznym dla procesorów stosowanych w Spectrum i QL. Zaczął też tworzyć oprogramowanie dla nowej konsoli do gier, ale firma została sprzedana i tych planów nie zrealizowano.

Co ciekawe, Sinclair nigdy nie posiadała praw do ROM-u. Amstrad, po

przejęciu firmy, musiał je kupić osobno od Nine Tiles w 1986 roku. Kiedy klony ZX Spectrum zaczęły pojawiać się na rynku pod koniec 1984 roku, szef Sinclaira, Nigel Searle odkrył, że nie można nic z tym zrobić, bowiem jedyną naprawdę unikatową częścią Spectrum był ROM. W efekcie sporów, o których pisałem wcześniej, firma nie zdołała uzyskać praw, które pierwotnie miały być zakupione za 5 tys. funtów. Do tego czasu sprzedano ok. 2,5 miliona sztuk komputerów.

Amstrad uzyskał jednak prawa do ROM-ów komputerów Spectrum i QL. Firma Nine Tiles zachowała prawa do programów dla ZX-80 i ZX-81. Jest to ważne, bowiem nieco później twórcy komputera SAM Coupé chcieli uzyskać licencję dotyczącą procedur zmiennoprzecinkowych od Nine Tiles, jednak w tym czasie proponowana cena była zbyt wysoka.

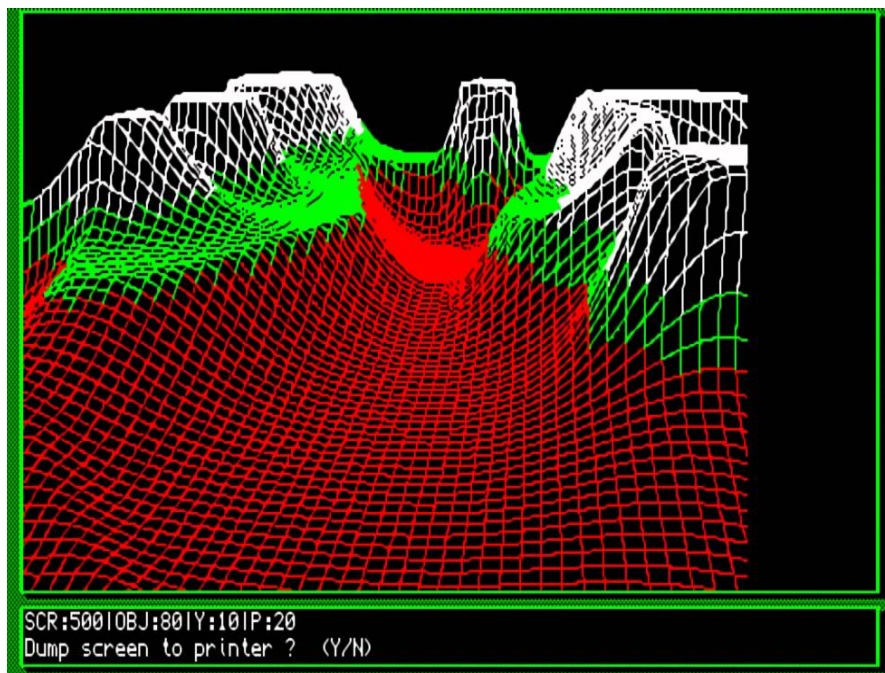
Pod koniec 1986 roku, gdy Amstrad chciał stworzyć ZX Spectrum z wbudowanym dyskiem twardym, po prostu zastosował DOS z modelu PCW. Poprawiono też edytor 128, aby umożliwić prosty dostęp do

funkcji dyskowych. System DOS, napisany przez Cliffa Lawsons, był bardzo dobry, chociaż jego pełna moc pozostała niewykorzystana w modelu Spectrum +3. Niestety w pierwszej wersji nie zostały poprawione błędy, a jednocześnie pojawiły się nowe. Miało to związek z problemami w dziale Sinclair, gdzie pracowano na ograniczonej dokumentacji, a organizacja pracy była daleka od ideału.

Amstrad przestał sprzedawać ostatni model ZX Spectrum dopiero na początku lat dziewięćdziesiątych. Później przez pewien czas wydawało się, że SAM Coupé może być interesującym sposobem na upgrade sprzętowy, ale maszyna nie uzyskała planowanej popularności i szybko została zapomniana. Renesans nastąpił dopiero wraz z nadejściem możliwości emulacji. Również firma Amstrad wyraziła zgodę na dystrybucję swojego materiału chronionego prawami autorskimi do użytku z emulatorami, tak więc możemy dzisiaj cieszyć się programem takim jak Sim Coupe. Działa on bardzo przyzwoicie na różnych systemach i umożliwia zapoznanie się z szeregiem oprogramowania dla "Sama", którego mało kto ma na swoim biurku.

Licencja Amstrada pomogła także Paulowi Dunnowi, który opracował zintegrowane środowisko programistyczne o nazwie "BASin". Piliśmy o nim w poprzednich numerach RetroKompa. Oprócz doskonałego edytora, program zawiera całą zawartość oryginalnego podręcznika dla Spectrum Basic, modelu "plus" oraz 128 w formie pomocy online. Program jest co prawda rozwijany tylko dla Windows, ale jest open source, więc może być przeniesiony na inne systemy. Działa też pod kontrolą Wine, dlatego w praktyce może być używany z większością nowoczesnych komputerów.

Opracował: Kamil Stokowski





Warunki i porównania

Instrukcje warunkowe są typowe dla wszystkich dialektów języka Basic i ich podstawowy sposób użycia jest łatwy do zrozumienia. Jednak wprowadzane warunki można połączyć lub zamienić z odpowiadającymi im funkcjami logicznymi. Ścisłe mówiąc są to funkcje, których operacje dają ten sam wynik. Daje to możliwość uproszczenia programu, szybszego działania lub zaoszczędzenia cennej pamięci. Przyjrzyjmy się temu bliżej.

Wiadomo, że polecenie IF przyjmuje postać struktury IF ... THEN, gdzie pomiędzy należy podać warunek. Jest to bardzo proste do stosowania, jednak jeżeli chcemy napisać bardziej rozbudowany program, musimy użyć wiele podobnych linii w celu porównania różnych wartości.

Tutaj zaczyna się pewien problem, gdyż porównywania typu "mniejsze" czy "większe" można ze sobą łączyć za pomocą operacji logicznych, a one są to zrozumienia trudniejsze.

Bez zbędnych ceregieli, zacznijmy od porównania AND, które jest prawdziwe tylko wtedy, gdy oba łączone warunki są prawdziwe. Na przykład możemy wpisać poniższą linię:

```
IF A$="OK" AND X>1 THEN  
PRINT X
```

Wartość zmiennej X zostanie wypisana tylko wtedy, gdy jednocześnie - zmienna A\$ będzie zawierała tekst "ok" oraz X przyjmie wartość większą od jednośc. Jeśli jeden z tych warunków wystąpi osobno, program nie wywoła instrukcji PRINT. Przy pomocy funkcji AND można połączyć ze sobą wiele porównań, a całość będzie prawdziwa, jeśli prawdziwe jest każde z nich.

Czasem jest to niewystarczające, dlatego mamy do dyspozycji porównanie OR, które będzie prawdziwe wtedy, gdy przynajmniej jeden warunków będzie spełniony.

Kolejną opcją jest słowo NOT, które odwraca wszystko do góry nogami. Takie porównanie jest prawdziwe tylko wtedy, gdy warunki nie zostają spełnione, natomiast będzie fałszywe wtedy, gdy warunki są spełnione. Można powiedzieć, że jest to "inwersja" w stosunku do poprzednich możliwości.

Wyrażenia logiczne mogą być tworzone na podstawie warunków tak samo, jak wyrażenia liczbowe, dlatego w razie potrzeby można umieszczać się w nawiasach. Ponadto, w wypadku użycia bardziej skomplikowanych formuł, posiadają one analogiczne priorytety jak operacje dodawania, odejmowania, mnożenia, dzielenia i potęgowania. Najniższy priorytet ma funkcja OR, następnie AND, NOT, a następnie zwykłe operacje.

Należy zauważyć, że NOT jest w rzeczywistości funkcją posiadającą argument oraz wynik, ale jej priorytet jest dużo niższy od priorytetów innych funkcji.

Dlatego wpis w tym wypadku nie wymaga nawiasów, o ile nie zawiera AND lub OR (lub obu). Zapis:

```
NOT A=B
```

oznacza to samo co:

```
NOT (A=B)
```

Natomiast porównanie "większe-równe" (czyli "<=") jest negacją równości, co można zapisać jako:

```
A < B
```

ale także inaczej:

NOT a=b

Odwracając wynik możemy zastosować też wpis "a=b" oraz "NOT a<>b" i oba są równorzędne. Podobnie porównania "większe-równe" i "mniejsze-równe" (czyli ">=" i "<=") są negacjami porównania "większe" lub "mniejsze".

Łatwo o tym zapomnieć lub pomylić, dlatego osobiście lubię przyrównywać ich działanie do przedziału zamkniętego i otwartego w matematyce. W ten sposób można również pozbyć się słowa NOT na początku warunku.

Teraz chciałbym poruszyć zagadnienie tak zwanych praw De Morgana. Są to twierdzenia w logice matematycznej sformułowane przez angielskiego matematyka Augustusa De Morgana. Oczywiście nie będziemy się zagłębiać w ten temat, jednak warto wiedzieć, że zgodnie z tymi prawami możemy zapisać zamiennie poniższe porównania:

NOT(pierwsze wyrażenie AND drugie wyrażenie)

i

NOT(pierwsze wyrażenie) OR NOT(drugie wyrażenie)

oraz

NOT(pierwsze wyrażenie OR drugie wyrażenie)

i

NOT(pierwsze wyrażenie) AND NOT(drugie wyrażenie)

W wyniku działania obu grup funkcji otrzymamy ten sam wynik, bowiem działają one identycznie. Mówiąc bardziej ogólnie, możemy zmienić miejsce słowa NOT i umieścić się przed wyrażeniami, a nawet pozbyć się ich przez odpowiednią zmianę warunku. Mówiąc językiem logiki,



Odwracamy kolory - będzie wygodniej pisać program.

funkcja NOT jest zbędna, chociaż jej używanie może znacząco uprościć cały program.

W linii ze słowem IF, gdzie nie podamy wartości porównania, według tego schematu:

IF warunek THEN ...

Wartość może przyjąć dowolną liczbę. Jeśli wyniesie "zero", warunek zostanie potraktowany jako fałsz - to jest oczywiste. Jednak jeśli uzyskamy tu jakąkolwiek inną wartość (nie tylko "jeden"), uzyskamy logiczną "prawdę".

Jest to to kolejna rzecz, którą można przegapić. W związku z powyższym powyższa linia oznacza:

IF warunek <> 0 THEN ..

Wyrażenia używające funkcje AND i OR mogą być stosowane też w inny

sposób. Zawsze jednak pozostają warunkami, nawet jeśli wpisujemy linię jak poniżej:

LET a=b*(1.5 OR a\$="mniejsze")

Zwróćmy uwagę, że standardowa wartość funkcji AND wynosi 0, a domyślna wartość dla OR to 1. W związku z tym możemy je wykorzystać podczas mnożenia.

Dodatkowo, funkcję AND można użyć w stosunku do zmiennych o różnych rodzajach i nadal będzie działać prawidłowo. Na przykład:

X\$ AND Y

Wszystko funkcjonuje bez błędów, bowiem jeśli ciąg tekstowy będzie pusty (""), jego wartość zostanie potraktowana jako "zerowa".

**Opracował:
Mariusz Wasilewski**

Instalujemy ParaDOS

Każdy szczęśliwy posiadacz komputera takiego jak Amstrad CPC 6128 doświadczy kiedyś awarii, na przykład stacji dyskietek. Sprzęt nie jest produkowany od wielu lat i brakuje do niego części zamiennych. Wszystko to powoduje, że naprawy nie są łatwe, lekkie i przyjemne. Zupełnie innym problemem jest dostępność dyskietek w egzotycznym dzisiaj formacie 3-calowym. Bardzo trudno znaleźć nowe, a jeśli już nawet trafimy ofertę, z pewnością nie będzie ona miała okazyjnej ceny. Czy jest jakieś wyjście z tej sytuacji?

Oczywiście - przerobić stację dyskietek 3,5 cala od peceta. Była o tym mowa w jednym z poprzednich numerów RetroKompas, ale teraz chciałbym skupić się bardziej na oprogramowaniu. Powiedzmy jednak kilka słów o kwestiach sprzętowych. Na portalach aukcyjnych można kupić napędy bez problemu, niektóre

modele są łatwe do przeróbki, inne niestety wymagają znacznie więcej pracy. W zasadzie dostosować można każdą stację, musimy tylko umieć posługiwać się lutownicą.

Najłatwiej przerobić stacje firm Sony, Samsug i Mitsumi. Mają one dość dobrze opisaną płytkę, chociaż w większości przypadków nie posiadają zworek, za pomocą których można zmieniać numer stacji. Jeżeli macie możliwość znalezienia stacji ze

zworkami, polecam ich zakup, bo upraszcza to temat.

Standardowo stacja w PC jest bowiem ustawiona jako "druga", czyli DS1 ("B:" w DOS-ie). W sieci znajdziemy wiele stron odnośnie przeróbek stacji dla Amigi i Atari ST, dlatego jest to temat szeroko znany. Dla tamtych komputerów przeróbka polega na zamianie numeru stacji, czyli z pozycji DS1 musimy uzyskać DS0. Ponadto należy połączyć sygnał





Dwa układy ROM na płycie głównej komputera.

READY na pin numer 2 stacji. Niestety w przypadku Amstrada pracy mamy trochę więcej. Po pierwsze, kable we wtyczce zasilania są odwrócone, czyli zamiast 5V mamy 12V i odwrotnie. Dlatego pierwszą rzeczą jest zamiana dwóch skrajnych kabli miejscami.

Poza tym należy odpowiednio przerobić taśmę sygnałową, na szczęście nie jest to skomplikowane. Piny 33 i 34 muszą być połączone, na przykład za pomocą zworki, bowiem CPC potrzebuje zawsze ustawionego sygnału gotowości. Tak samo połączone powinny być piny 11 i 12, co spowoduje, że komputer będzie używał naszej stacji jako napędu podstawowego. Natomiast po połączeniu pinów 31 i 32 napęd będzie używał 2 głowic. Przeróbkę można wykonać lutując kable lub - tak jak wcześniej wspomniałem - montując goldpiny i zworki w odpowiednich miejscach na gniazdach taśmy sygnałowej.

W tym miejscu zaczyna się tytułowy temat, czyli ParaDOS. Jest to system obsługi dyskietek, który poprawia

błędy w oryginalnym oprogramowaniu. Normalnie, modele CPC 664 i 6128 mają dwa układy ROM. Pierwszy zawiera Basicm, a drugi AM-DOS. Są to właśnie procedury do obsługi stacji dyskietek. ParaDOS zastępuje drugi AMDOS-a, co przydaje się na przykład podczas formatowania dyskietek. Ponadto mamy do dyspozycji program narzędziowy uruchamiany za pomocą polecenia:

|DRIVE

System obsługuje różne formaty dyskietek oraz różne pojemności, od ok. 200 do ok. 800 KB pojemności. Najlepiej działają formaty o nazwach ROMDOS D1 i ROMDOS D2. Możliwe staje się również zapisywanie pojedynczych plików o maksymalnej objętości 128 KB lub 256 KB. W związku z powyższym za pomocą ParaDOS-u bez problemu użyjemy dyskietek o typowej pojemności 720 KB. Praca z takimi nośnikami to czysta przyjemność, bowiem gry i programy zajmują zwykle maksymalnie 64 KB, czasem więcej. Tak więc na jednej dyskietce zapiszemy kilka-

naście pozycji. najczęściej max 64. Na dysku mamy możliwość zapisu kilku/kilkunastu gier, dem bądź użytków. Raj.

Proces przygotowania Amstrada to instalacji ParaDOS-u nie jest trudny, chociaż wymaga sprawnego posługiwania się lutownicą. Należy rozkręcić komputer, odłączyć wszystkie urządzenia i wyjąć płytę główną. Pod procesorem Z80, zobaczymy dwa układy ROM. Najbliższy (z oznaczeniem 40015) to AMDOS. Należy go usunąć, a w jego miejsce przylutować podstawkę pod nowy układ. Dalej należy zapisać "wsad", czyli obraz pamięci ROM. Można tu wykorzystać pamięć 27C128, co nie zajmuje wiele czasu. Można też użyć adaptera PLCC, trzeba mieć tylko dostęp do programatora.

Jeżeli nie mamy odpowiedniej kości Eprom możliwe jest też użycie innej pamięci jak np. 27C256. Niestety powoduje to, że komputer uruchamia się w trybie bez obsługi stacji dyskietek. Aby temu zapobiec, podczas programowania należy zmienić parametr Offset na wartość 4000. To wszystko. Przypicie, że instalacja nie jest trudna. Obsługa nowego ROM-u także nie należy do skomplikowanych. Po uruchomieniu komputera będziemy mieli informację o ParaDOS-ie, a później możemy korzystać ze standardowych komend CPC oraz nowych.

Obraz pamięci do zaprogramowania można znaleźć tutaj:

<http://www.winape.net/download/ParaDOS%201.2.zip>

Wiele przydatnych informacji jest też dostępnych na stronach CPCWiki oraz CPCTech, ale jest to dokumentacja w języku angielskim. Dlatego postanowiłem przybliżyć temat ParaDOS-u naszym Czytelnikom.

Mariusz Wasilewski

Obudź w sobie bohatera

Gry na Amstrada nie są tak dobrze znane, jak dla innych komputerów. Często są traktowane jako gorsze odpowiedniki tytułów dla Commodore czy Atari. Dlatego postanowiłem pokazać, że poza ciekawą grafiką czy muzyką mają one często naprawdę wciągającą fabułę. Mój opis jest inny niż suche recenzje gier, niektóre gry podobają mi się bardziej, inne mniej. Mam nadzieję, że taki sposób przedstawienia przypadnie Czytelnikom do gustu.

Baba's Palace

Shaolin Kid and Little Indian podróżowali do pałacu Baby Jagi w poszukiwaniu informacji na temat eliksiru życia. Jednak Baba Jaga przekazuje temu duetowi informację tylko wtedy, gdy pokonani zostaną wszyscy popleczników, na 100 poziomach o różnym stopniu trudności.

Na pierwszych kilku poziomach kontrolujesz bohatera i musisz biegać, wspinać się po drabinach, aby

znaleźć sposoby na zabicie wszystkich wrogów. Później Shaolin Kid i Little Indian występują razem i muszą współpracować ze sobą, co jest jeszcze bardziej ciekawe.

Ta gra wygrała konkurs Creation Creation 2017 CPC RetroDev i jest bardzo dopracowana. Posiada świetną, kolorową grafiką, a także animację i fantastyczną muzykę. Aby przejść wiele poziomów potrzeba mocno pomyśleć, ale kiedy wypracu-

jesz rozwiązanie, możesz mieć naprawdę wiele satysfakcji. Po ukończeniu każdego poziomu otrzymujemy specjalny kod, więc nie trzeba powtarzać wcześniejszych za każdym razem. Dzięki temu gra nie jest zniechęcająca i szybko się nie nudzi.

Baby Jo

Tytułowe Baby Jo gubi się w parku, a jego matka martwi się o niego, ale



będąc silnym i odważnym małym dzieckiem, udaje się samodzielnie do domu. Mamy tu cztery poziomy, które prowadzą Jo przez park i kilka jaskiń oraz szybów kopalni.

Jest to wesoła gra platformowa, grafika jest naprawdę kolorowa, a sceneria i rodzaje potworów do pokonania są również bardzo sympatyczne. Poziomy są duże i podzielone na kilka sekcji. Jeśli stracisz życie, wracasz do początku sekcji, w której zacząłeś grę. Są też również hasła dla każdego poziomu, co jest bardzo pomocne. Jedyne problem z grą polega na tym, że w ogóle nie ma dźwięku. Mimo to można spędzić przy niej dużo czasu.

Bachou

Teraz wracamy do początków Amstrada CPC. Firma Central Solutions była znana z tego, że zawsze wydawała gry naprawdę fatalne i Bachou niestety nie stanowi tu wyjątku. Kosmici zaatakowali Ziemię i niszczą miasta, a Ty musisz ich zestrzelić. Niespecjalnie oryginalne, prawda?

Każdy poziom rozgrywa się na tym samym pojedynczym ekranie z czterema miastami na dole. Kosmici latają dookoła ekranu zrzucając bomby na miasta, a gdy wszystkie są już zniszczone, gra się kończy. Jeśli zestrzelisz wystarczająco dużo kosmitów, możesz przejść do następnego poziomu, który jest mniej więcej taki sam jak poprzedni.

Efekty graficzne i dźwiękowe są prymitywne i chyba nie ma powodu, aby wracać do tej gry. Jedyne, co może się podobać, to szybkość rozgrywki. Musimy reagować naprawdę dynamicznie, aby pokonać kosmitów. Gdyby ekran gry był większy, być może byłaby to pozycja bardziej godna do polecenia.

Back to Reality

Statek, w którym podróżujesz wpadł do czarnej dziury i wynurzył się w równoległym wszechświecie. Musisz go skierować z powrotem do prawdziwego „naszego” wszechświata, tworząc antymaterię i łącząc ją z materią. Jakkolwiek dziwnie to

brzmi, taka jest fabuła gry Back to Reality.

Jest to gra przygodowa polegająca na zbieraniu przedmiotów i przekształcaniu ich w nowe obiekty, z których ostatecznie powstaje antymateria. Będziesz musiał być naprawdę dobry w nauce, aby rozwiązać wszystkie zagadki. Masz również ograniczony zapas tlenu, który trzeba często uzupełniać.

Grafika jest w porządku, podoba mi się szczególnie animacja. Niestety kuleje muzyka (jeśli można to tak w ogóle nazwać), natomiast gra w dalszych poziomach jest mało różnorodna i przez to mniej interesująca. Jednak jeśli nigdy nie próbowałeś tego tytułu, być może przypadnie Ci do gustu.

Back to the Future Part II

Przypuszczam, że wszyscy widzieliśmy film pod tym tytułem, więc nie ma potrzeby wyjaśniania fabuły. Gra rozpoczyna się od pościgu na deskolotkach, ale wygląda to dość słabo w trybie monochromatycznym. Drugi poziom wygląda bardziej atrakcyjnie, ponieważ pomagasz Jennifer wyjść z domu tak, aby nikt Cię nie zauważył. Po tym następuje kolejny monochromatyczny poziom typu bijatyka, który prowadzi do animowanej i przede wszystkim – bardziej kolorowej gry logicznej

Ostateczny poziom jest podobny do pierwszego, a zakończenie to odlot DeLoreanem w siną dal. Trzeba powiedzieć, że grafika jest wykonana na bardzo różnym poziomie, w zależności od lokacji. Rozgrywka jest dość powolna, ale z drugiej strony pozwala to na podziwianie oprawy graficznej. Na szczęście na podstawie trzeciej części filmu opracowano lepszą grę.



Back to the Future Part III

Tę grę opracowała firma Image Works i zdecydowanie lepiej wykorzystano możliwości CPC. Ponownie mamy świetne intro po załadowaniu, a potem – w całej grze – dobre efekty wizualne. Na początku gonisz pociąg, aby uratować ukochaną doktora Browna. Zaczynasz na koniu i podróżujesz na nim do różnych miast. Kolejno ekran zmienia się na dwuwymiarową strzelaninę z kowbojami i Indianami. Poza tym, walczysz z gangiem Wściekłego Psa w rzucie izometrycznym.

Gra jest dużo bardziej urozmaicona od poprzedniej części. Mamy bardziej kolorową grafikę i zmieniające się poziomy, które powodują, że rozgrywka nie nudzi. Również dźwięk stoi na dużo lepszym poziomie.

Back to the Golden Age

Wielki Mag niegdyś rządził światem, wykorzystując moc czterech magicznych kryształowych kul. Mag trzymał jedną z nich, a resztą rozporządzali kapłani. Pewnego dnia jeden z kapłanów ukradł kule próbując przejąć kontrolę nad światem. Twojego bohaterowi powierzono zadanie, aby odzyskać brakujące magiczne kule.

Jest to typowo arcadowa gra przygodowa, podczas której można eksplorować zamki i lochy, zbierać różne mikstury, walczyć z rycerzami, kupować zapasy i rzucać zaklęcia. Opisy mikstur są dość tajemnicze, ale to powoduje, że grę trzeba naprawdę odkrywać, więc szybko się nie znudzi.

Grafika i animacja zostały wykonane znakomicie, chociaż dominuje jednolita kolorystyka. Świat gry jest rozbudowany i można się w niego wciągnąć na dłużej. Jest to jednak z

moich ulubionych gier ze względu na stopień dopracowania grafiki.

Bacterik Dream

Badacze z armii amerykańskiej stworzyli wirusa, niestety niedługo później wymknął się on spod kontroli. Nasi bracia zza oceanu nie mieli wyjścia, jak tylko poprosić pomoc w zwalczeniu śmiertelnej broni biologicznej.

Każdy poziom ten gry składa się z pojedynczego ekranu zawierającego wirusy i białe krwinki. Musisz zniszczyć wszystkie wirusy zanim zainfekują one wszystkie komórki krwi. Aby to zrobić, możesz użyć specjalnego narzędzia do „splaszczania” wirusów, pary laserów do ich likwidacji poprzez ciepło lub też kombinacji tych dwóch rozwiązań. Jednak metodę musisz wybrać na samym początku, co – jak się okazuje później – jest sporym utrudnieniem.

Każdy poziom zawiera również wiele kolorowych płytek, które wpływają na rozgrywkę na różne sposoby. W zasadzie jest to prosta gra zręcznościowa, ale bardzo przyjemna. Akcja jest wręcz szalona, a zasady mało skomplikowane. Wydane mi się, że to stanowi siłę Bacterik Dream.

Bactron

Pacjent jest chory, znajduje się w stanie krytycznym, a wirusy wciąż rozprzestrzeniają się po jego ciele. Bactron, czyli żółta kropla antybiotyków, musi aktywować enzymy wysterylizowane przez wirusy.

Gra wymaga niezłego główkowania, aby zlokalizować enzymy w ciele pacjenta unikając wirusów. Będą one wyczerpywać Twoją energię, jeśli je dotkniesz. To także wyścig z czasem; bo ciągle wytwarza się większa ilość wirusów i wzrasta temperatura pac-

jenta. Trzeba się więc bardzo spieszyć!

Enzymy są pokazane jako żółte kostki, a ich aktywacja zwiększa energię bohatera. Pod żadnym pozorem nie należy dotykać jasnoniebieskich kostek. Lokacje są widoczne z perspektywy izometrycznej, a grafika jest naprawdę wspaniała, szczególnie biorąc pod uwagę rok, w którym gra została wydana – 1986. Bardzo sympatyczna jest też muzyka, choć nigdy nie zwróciłem na nią specjalnej uwagi.

Bad Cat

Zainspirowane Igrzyskami Olimpijskimi, bezpieczne koty w Los Angeles połączyły siły, by zorganizować własną konkurencję sportową. Maksymalnie czterech graczy może rywalizować w czterech głównych poziomach. Istnieją dwie przeszkody do pokonania. Musisz wyskoczyć w powietrze i łapać geometryczne kształty, starając się nie wpaść w kałużę wody. Druga część to dziesięciopunktowa gra w kręgle przeciwko psu. Między każdą „imprezą” musisz podróżować przez miasto na motorze.

Osobiście, gdy po raz pierwszy zobaczyłem ekran tytułowy Bad Cat, zacząłem posądzać autorów o rysowanie grafiki „po spożyciu”. Jednak później przekonałem się, że grafika jest naprawdę bardzo dobra, a cała gra skierowana jest dla młodszych użytkowników. Jest łatwa i prawdziwa zabawa zaczyna się dopiero wtedy, gdy zagramy w co najmniej dwóch graczy. Jest to dobra pozycja na rodzinną imprezę, gdy chcemy pokazać ciekawy tytuł naszym pociechom. O ile oczywiście uda się je przykuć do komputera z pierwszej połowy lat '80-tych.

Opracował:
Kamil Stokowski

VIC-20

W cieniu wielkiego brata

VIC-20 którego popularność nie była aż tak duża, jak jego młodszego brata - Commodore C64, został wypuszczony na rynek w 1980 roku. VIC-20 był z kolei następcą Commodore PET, pierwszego domowego komputera z prawdziwego zdarzenia, który miał dość ograniczone możliwości. PET posiadał jeden monochromatyczny znakowy tryb graficzny, bez możliwości zmiany fontów. Oba komputery posiadają procesor 6502, bardzo popularny w tych czasach układ, który napędzał także inne 8-bitowce jak Atari XL/XE czy Apple II.

VIC-20 to już dużo bardziej zwarta i zupełnie inaczej zaprojektowana konstrukcja, niż w starszym bracie PET. Elegancka, zaokrąglona obudowa (wykorzystana także w pierwszych modelach C64), brak wbudowanego monitora i wbudowanego magnetofonu (choć późniejsze modele PET też go nie posiadały). To już komputer, który

można wygodniej użytkować, nie zajmuje dużo miejsca i wydaje się być bardziej przyjazny dla przeciętnego użytkownika. Do nowego modelu dodano port joystick'a i kartridża. Zmieniono złącze stacji dysków z równoległego na szeregowo – mniejsze, ale z wolniejszym przesyłem danych. Poprzez minimalizację względem poprzednika, koszty produkcji takiego modelu też były dużo niższe, więc jego końcowa niewysoka cena (początkowo wynosiła 299,95 USD)

spopularyzowała komputer, bo stać było na niego każdego przeciętnego użytkownika. Do początku 1983 roku sprzedano milion egzemplarzy, a w 1985 roku zakończono produkcję. Według różnych źródeł, łącznie sprzedano od 2,5 do 3 milionów egzemplarzy.

W odróżnieniu od PET'a, jego następca posiadał dodatkowo zintegrowany układ generujący grafikę i muzykę, o nazwie Video Interface Chip, w skrócie VIC (stąd





też nazwa komputera), który był zaprojektowany przez firmę MOS i planowano go montować m.in. w terminalach z kolorowymi wyświetlaczami. Możliwości graficzne tego układu to wyświetlanie grafiki na 256 redefiniowanych znakach, w dwóch trybach tekstowych: 22 na 23 lub 22 na 13 pól. Szesnaście dostępnych kolorów (z pewnymi ograniczeniami), oraz brak sprite'ów.

oryginalnie 16KB pamięci (z możliwością rozszerzenia do 48KB). Na szczęście wiele oprogramowania na VIC'a (głównie gier) było wydanych na kartridżach, które posiadały własną pamięć ROM (głównie były to moduły 8KB i 16KB) więc 5KB w komputerze w zupełności wystarczało. W czasach, gdy VIC-20 był popularny, sprzedawano także kartridże zwiększające pamięć

wadzenia ciepła (stąd obecność sporego radiatora wewnątrz obudowy komputera). Przez to starsza płyta główna jest szersza i wypełnia całą przestrzeń obudowy komputera. Późniejsze modele posiadały już zasilacze, które dostarczały napięcia 9VAC i 5VDC (są to zasilacze z 7-pinową okrągłą wtyczką, znaną z modelu C64). W tej wersji płyta jest węższa i mniej rozbudowana.



Obie wersje płyt nie posiadały wbudowanego modulatora TV. Był on podłączany do komputera jako osobne urządzenie. Komputer można było także podłączyć do telewizora przez złącze kompozytowe (popularny cinch). Niestety uzyskanie sygnału chroma-luma poprzez złącze S-VHS, bez odpowiedniej przeróbki, nie było możliwe (w przeciwieństwie do Commodore C64, który taki sygnał ma w standardzie). Obecnie VIC'a ciężko znaleźć na polskich aukcjach internetowych. Jest dużo mniej popularny niż C64. Warto rozejrzeć się więc za granicą, przykładowo na:

<https://www.ebay.co.uk>

Od strony możliwości muzycznych, jest też dużo lepiej niż w modelu PET, który de facto fabrycznego układu dźwiękowego nie posiadał. VIC może generować dźwięk na czterech kanałach, z czego 3 potrafią wygenerować falę prostokątną, a ostatni kanał - biały szum. Niestety ograniczeniem jest regulacja głośności, która działa na wszystkie kanały naraz, nie pozwalając wyciszać z osobna każdego z nich. Można dany kanał jedynie wyłączyć.

VIC-20 oryginalnie posiadał 5KB pamięci (z możliwością rozbudowy do 32KB), wówczas z poziomu BASIC'a, dla użytkownika dostępne było jedynie 3,5KB pamięci. Dla porównania Atari 400 wprowadzone na rynek w 1979 roku posiadało

komputera. Dostępne były m.in. rozszerzenia VIC-1111 dające do dyspozycji dodatkowo 16KB pamięci RAM, VIC-1110 z 8KB na pokładzie, jak i VIC-1211A, który zwiększał pamięć RAM o 3KB i dodatkowo rozszerzał możliwości BASIC'a. Do VIC-20 można było podłączyć magnetofon, drukarkę, stację dysków, a nawet modem.

Istnieją dwie wersje płyt głównych. W starszej wersji, napięcie jakie dochodziło z zasilacza wynosiło 9VAC (2-pinowa wtyczka zasilania o prostokątnym kształcie). Powodowało to konieczność uzyskania napięcia 5VDC do zasilania układów scalonych bezpośrednio na płycie i wymagało dodatkowych elementów konwertujących napięcie, jak i odpro-

za około 200-300 PLN znajdziemy działający egzemplarz w dobrym stanie, z dodatkami czy nawet kartonem. Po zakupie, możemy próbować zdobyć oprogramowanie na kartridżach, kasetach lub dyskietkach, ale znacznie lepiej jest zaopatrzyć się w jedno z rozszerzeń, które opisujemy poniżej. Chociaż do pełnej funkcjonalności i uruchomienia całej bazy oprogramowania na ten komputer, najlepiej byłoby posiadać je wszystkie.

Penultimate to pierwsze z opisywanych urządzeń rozbudowujące skromne możliwości VIC-20. Można je zakupić na stronie:

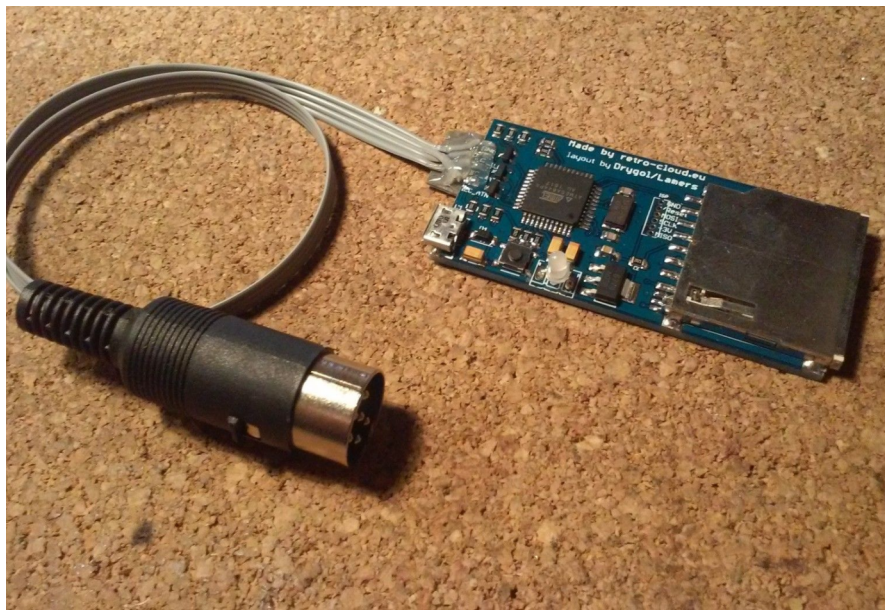
<https://www.thefuture-was8bit.com>

w cenie £54.99. Ten moduł wpinany jest do złącza kartridża i zawiera kilka ciekawych funkcji. Jedną z nich jest możliwość rozszerzenia pamięci - możemy ją ustawić m.in. na 3, 8, 16, 24, 32 i maksymalnie 35KB. Jest to dość istotne, bo wiele nowszych produkcji (zarówno gier jak i dem), ma problem z uruchomieniem na niektórych konfiguracjach i działa na przykład jedynie z 8KB pamięci. Na tym kartridżu w pamięci ROM zapisano też 40 gier, które są dostępne z poziomu menu Penultimate'a. Przywołujemy je naciskając lewy przycisk na rozszerzeniu. W naszej ocenie nie wszystkie gry są dobre, ale znajdzie się tu kilka perełek.

Najciekawsze i najładniej wykonane, to według nas: Choplifter, Defender, Donkey Kong, Galaxian, Lunar Leeper. Dig, Dug, LodeRunner i Ms Pac-Man. Kolejną funkcją jaką daje moduł jest możliwość szybkiego dostępu do FileBrowsera, który jest programem do obsługi i montowania plików w wirtualnej stacji SD2IEC. Oczywiście przeglądarka (w tym wypadku jest to FB20-8K) musi znajdować się na karcie SD. Uruchomimy ją wciskając F7, będąc w menu Penultimate'a. Jednocześnie zostaje ustawiona pamięć na 35KB.

Przed uruchomieniem browsera, możemy także ustawić inną pamięć w zaawansowanych opcjach, wciskając klawisz A i jedną z cyfr od 1 do 8, które włączą nam odpowiednią konfigurację pamięci. Przy zapełnianiu karty SD produkcjami musimy pamiętać, aby nie używać dużych liter w nazwach plików i ograniczać się do ośmiu znaków w opisie, w innym wypadku nazwy widziane z poziomu browsera staną się nieczytelne.

Warto jeszcze dodać, że na kartridżu oprócz wspomnianych gier, znajdziemy także ROM'y z BASI-C'iem w wersji v4 (oryginalnie w VIC'u zgłasza



nam się wersja v2), oraz Super Expander'a – wspomniane już 3KB rozszerzenie z wieloma dodatkowymi opcjami.

Kolejnym urządzeniem jest interfejs SD2IEC – sprzętowy emulator stacji dyskietek, z opcją montażu wewnątrz jak i zewnątrz komputera. Możemy go zakupić m.in. tu:

<https://www.retro-cloud.eu/>

Niestety to urządzenie nie jest tym, które pozwala na pełną emulację stacji dysków, więc uruchomienie niektórych produkcji (głównie dem) jest niemożliwe. Uruchomimy na nim głównie produkcje jedno-plikowe o rozszerzeniu PRG.

Do uruchomienia wszystkiego, musimy zatem posiadać oryginalną stację dyskietek, lub inny sprzętowy emulator: 1541 Ultimate-II (obecnie sprzedawana wersja to Ultimate II+ i można ją zakupić tutaj:

<http://www.1541ultimate.net>

Jest jednak problem, ponieważ emulator pasuje jedynie do portu kartridża od Commodore C64. Jak więc podłączyć go do VIC-20? Otóż, podłączamy emulator Ultimate-II

właśnie do C64 (rozszerzenie wkłada się do portu kartridża i jednocześnie podłącza się je przewodem z portem serial w komputerze). Teraz wybieramy odpowiedni plik do załadowania dedykowany dla VIC'a (w tym wypadku będzie to plik z rozszerzeniem D64 – jest to cyfrowy obraz dyskietki) i montujemy go z poziomu menu Ultimate'a pod stację np. z numerem 8. Nie wyłączając komputera, odłączamy kabel serial z C64 i podłączamy jego końcówkę do portu serial w VIC-20.

W ten sposób C64 stał się niejako zasilaczem i urządzeniem obsługującym nasz sprzętowy emulator stacji dyskietek. W tym momencie pozostaje nam wpisać komendę LOAD"nazwa pliku",8 z poziomu BASIC'a na VIC-20 i poczekać na wczytanie się produkcji. Może jest to mało wygodne, bo musimy mieć dwa komputery, każdy zasilany osobno, lecz doskonale spełnia swoje zadanie. Szczególnie gdy nie jesteśmy w posiadaniu oryginalnej stacji dyskietek.

Co zatem oprócz dawnych gier z lat 80-tych uruchomimy na VIC'u? Wydawałoby się, że na ten komputer o ograniczonych możliwościach nie powstają już żadne produkcje. Na



szczęście znajdują się pasjonaci, którzy piszą zarówno gry jak i dema. Powstają produkcje nowe, jak i porty innych gier. Z najciekawszych możemy wymienić Doom'a z 2013 roku – dość udany port gry z PC, który jest całkiem grywalny i zawiera nawet muzykę brzmiącą jak w oryginale, chociaż z wiadomych względów - uproszczoną.

Podczas rozprawiania się z potworami mamy do dyspozycji kilka rodzajów broni i musimy się przebić przez osiem poziomów, błędząc po korytarzach bazy i wykańczając przeciwników. Gra wymaga 35KB pamięci i uruchomi się z SD2IEC. Kolejną produkcją jest Pentagorath, wydany niedawno na kasecie magnetofonowej. Jest to gra przygodowa w rzucie izometrycznym, wizualnie przypominająca Knight Lore. Można ją zakupić tutaj:

<https://www.thefuturewas-8bit.com/pg-vic20.html>

Gra również wciąga i jest bardzo przyzwoicie wykonana. Kolejną grą jest Cheese and Onion – bardzo dobrze wyglądająca platformówka, której niestety nie mogliśmy przetestować w redakcji, bo jej nie posiadamy. Grę można zakupić w wersji na kartridżu (póki co nie

istnieje wersja darmowa) za £24.99 na wspomnianej wyżej stronie internetowej. Dalej mamy darmowe, niezależne produkcje takie jak: Dragonwing, Astro Nell, Boulder Dan, niedokończony, ale ulepszony względem oryginału z VIC'a Q*bert i wiele innych. Polecamy ich szukać na stronach z demoscenowymi produkcjami jak na przykład:

<https://demozoo.org>

Jeśli już jesteśmy w temacie dem, to polecamy obejrzenie kilku najciekawszych z nich na tę platformę. Przypominamy, że nie wszystko uruchomi się z SD2IEC. Jedną z najnowszych produkcji jest Gnork! z tegorocznego Revision. Demo zostało wystawione w kategorii „Oldskool Demo” i zajęło 3. miejsce. W naszej ocenie zasłużyło spokojnie na podium, a może nawet na wyższe miejsce. Inne ciekawe produkcje demoscenowe z poprzednich lat to: OMD - Orb MegaDemo, Impossiblator 2, Back in the Good Old Days, Robotic Liberation czy The Next Level. Oczywiście wszystkich produkcji na ten komputer nie wymienimy, napisaliśmy tylko o tych najciekawszych.

Mój VIC-20 wymaga jeszcze trochę pracy od strony wizualnej (czeka go m.in. wybielanie), jednakże, od strony

elektroniki prace zostały póki co zakończone. Prezentowany egzemplarz posiada następujące przeróbki i rozszerzenia:

**Michał "stRing"
Radecki-Mikulicz**

1. Gniazdo S-VHS, co daje obraz o dużo lepszej jakości niż standardowe złącze kompozytowe. Przeróbki dokonano na podstawie opisu ze strony:

http://sleepingelephant.com/denial/wiki/index.php?title=S-Video_output

2. Wyprowadzone złącze audio (w postaci gniazda cinch-mono), które podpięte zostało do portu audio/video wewnątrz komputera.

3. Wlutowany kołek do pinu drugiego w gnieździe User Port, z którego zasilany jest zewnętrzny emulator SD2IEC napięciem 5VDC:

http://www.hardware-book.info/VIC-20_User_I/O

4. Kartridż Penultimate:

<https://www.thefuturewas8bit.com/pu-vic-20.html>

5. Sprzętowy emulator stacji dyskiek SD2IEC, który współpracuje także z C64:

<https://www.retro-cloud.eu/>

6. Doraźnie Ultimate-II + C64 jako pełna emulacja stacji dyskiek:

<http://www.1541ultimate.net/>



Hardtrack Composer

Programy muzyczne dla C64 mają dość charakterystyczną obsługę. Jednak osoby, które nie miały wcześniej do czynienia z tego typu oprogramowaniem mogą czuć się zagubione. Dlatego zobaczymy czego możemy spodziewać się polskim programom. Okazuje się, że najbardziej chyba rozpoznawalny Voice Tracker nie jest jedyny i niezastąpiony.

Po uruchomieniu edytora na ekranie pojawi się główny ekran edycyjny. Wszystkie operacje odbywają się na jednym ekranie, pomijając obsługę magnetofonu. Dzięki temu nie musimy przechodzić do kolejnych opcji i wszystkie elementy są od razu dostępne. Główne części to trzy okna edycji ścieżek (zawierające osiem pozycji), trzy okna do edycji makro-rozkazów (zawierające dwie pozycje), okno do edycji paternów oraz linia informacyjna.

Myślę, że wszyscy wiedzą, że poszczególne ścieżki utworu odpowiadają kanałom dźwiękowym układu SID. Składają się maksymalnie z 256 pozycji. W poszczególnych

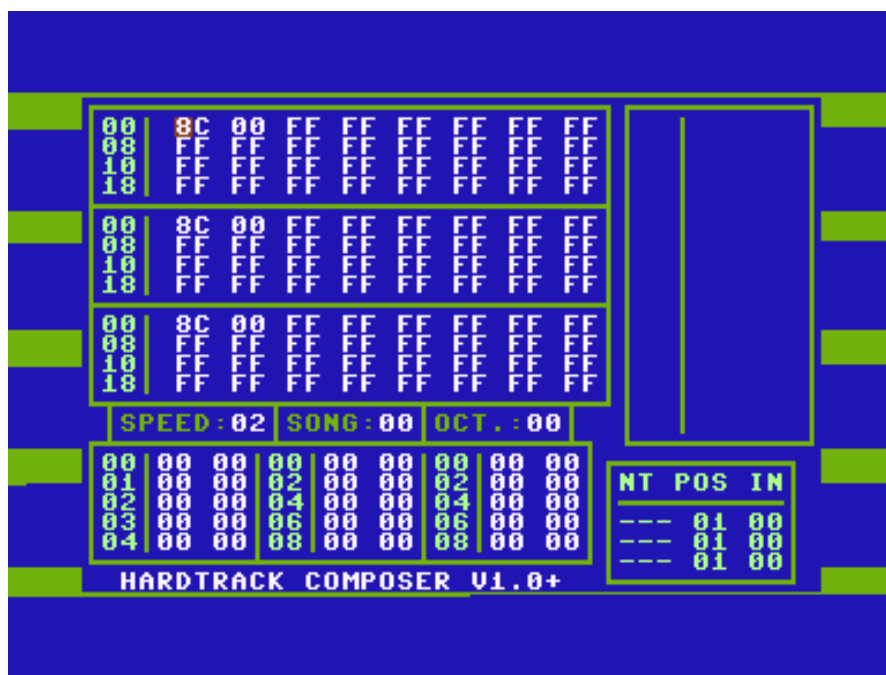
pozycjach możemy umieszczać numery paternów lub proste komendy dla programu odtwarzającego muzykę.

Umieszczając kolejne patterny w ścieżce, ustalamy ich kolejność w utworze. Komendy są natomiast wykonywane natychmiast i zaraz po nich pobierany jest numer kolejnego patternu do odtwarzania. Używanie komend nie powoduje żadnych opóźnień w odtwarzaniu utworu, a przynajmniej nie powinno być to zauważalne.

Do dyspozycji mamy kilka komend powodujących zmianę sposobu odtwarzania utworu. Najważniejsze to:

- FF - skok do początku ścieżki,
- FE - wyłączenie kanału,
- FD x - skok do podanej pozycji w ścieżce (numer pozycji podajemy jako "x").

Warto dodać, że komenda FE nie powoduje wyczyszczenia kanału przed wyłączeniem, w związku z tym,





jeśli jakkolwiek dźwięk wybrzmiewa w tej samej chwili, będzie to robił dalej, mimo że ścieżka nie będzie już odtwarzana.

Trzy dolne okna to tablice makrorozkazów. Każda z nich składa się z dwóch kolumn danych oraz z kolumny zawierającej numery porządkowe pozycji. Makrorozkazy to, najprościej mówiąc, tablice, dzięki którym możemy kształtować parametry dźwięku w czasie jego odtwarzania. Pozwala to tworzyć własne układy filtrów, kształtów fal oraz wypełnień.

Liczby od 00 do 7F to numery patternów do odtwarzania. Należy pamiętać, że wszystkie liczby w edytorze są podawane w systemie szesnastkowym (heksadecymalnym).

Liczby od 80 do FC to tak zwane transpozycje. Przyrost wartości transpozycji o jednostkę podnosi wysokość patternu o jeden półton. Transpozycja wpływa na wszystkie patterny umieszczone dalej, aż do napotkania następnej instrukcji transpozycji. Dzięki tym komendom możemy odegrać pattern o dowolną liczbę półtonów wyżej, niż jest on odtwarzany normalnie, czyli zgodnie z wpisanymi dźwiękami. Transpozycja 80 jest najniższa, a transpozycja FC jest wartością największą.

Pierwszą tablica dotyczy transpozycji i kształtu fal. W lewej kolumnie wpisujemy kształty fal w następującym formacie:

- 1x - fala trójkątna (np. brzmienie fletu),
- 2x - fala piłokształtna (często używana przy tworzeniu akordów),
- 4x - fala prostokątna (instrumenty perkusyjne lub bas),
- 8x - szum.



Sumując te wartości, możemy łączyć kształty fal, na przykład: liczba 3 da w rezultacie połączenie przydatne przy tworzeniu instrumentów perkusyjnych oraz efektów specjalnych.

Każda pozycja składa się z dwóch cyfr (od 0 do F), więc należy tu używać drugiej cyfry z pierwszej kolumny, która ma następujące znaczenie:

- 0 - zanikanie fali,
- 1 - narastanie, podtrzymanie i opadanie fali,
- 2 - włączenie synchronizacji kanałów
- 4 - włączenie modulacji,
- 8 - wyłączenie kanału

W ostatnim przypadku pamiętajmy, że program odtwarzający będzie działał dalej, mimo że w danym kanale nie będzie nic słychać. Powyższe komendy również możemy łączyć, na przykład: liczba 6 spowoduje włączenie modulacji połączonej z synchronizacją.

W prawej kolumnie umieszczane są transpozycje. Dzięki nim możemy płynnie sterować wysokością dźwięku w czasie odtwarzania. Wyróżniamy tutaj dwa rodzaje transpozycji, o następujących wartościach:

- od 00 do 5F - transpozycje względne,
- od 80 do DF - transpozycje bezwzględne.

W pierwszym przypadku wysokość dźwięku jest obliczana ze wzoru: wysokość nuty w patternie + transpozycja x 1 półton. Dostępne są też dwie komendy, które nie wpływają bezpośrednio na rejestry dźwiękowe. Składają się z dwóch cyfr:

- FE - powoduje zakończenie pobierania informacji z makrorozkazu i od tej chwili dźwięk będzie posiadał takie parametry, jakie zostały ustalone za pomocą ostatniej komendy w makrorozkazie,
- FF x - powoduje skok do pozycji podanej w makrorozkazie (numer podawany jest jako zmienna "x").

Następna tablica dotyczy zmian wypełnienia dźwięku. W lewej kolumnie podajemy o ile ma przyrastać lub opadać wypełnienie. Wartości parzyste odpowiadają przyrastaniu, a nieparzyste - opadaniu. Jest to nie do końca logiczne, ale musimy się przyzwyczaić do takiej obsługi. W prawej kolumnie podajemy natomiast czas, w ciągu którego włączone będzie narastanie lub opadanie wypełnienia.

Dostępna jest także jedna dodatkowa komenda: "FF x", która powoduje skok do pozycji w makrorozkazie (podownie numer podajemy jako zmi-

tego na ekranie znajdują się trzy wskaźniki wskazujące na:

- SPEED - tempo utworu,
- SONG - numer utworu,
- OCT - aktualnie wybraną oktawę.

Dodajmy, że dla każdego utworu możemy ustalić inne tempo odtwarzania, co pozwala na zaoszczędzenie pamięci, która będzie wykorzystywana tym bardziej, im bardziej skomplikowane wpisy w patternach będziemy robić. W prawym dolnym rogu znajduje się mniejsze okno, w którym w czasie odtwarzania utworu pokazywane są informacje dla

- F1 - rozpoczęcie odtwarzania aktualnego utworu od początku,

- F2 - zatrzymanie odtwarzania utworu,

- F3 - kontynuacja odtwarzania od miejsca, gdzie utwór został zatrzymany,

- F5 - spowolnienie odtwarzania od aktualnego miejsca,

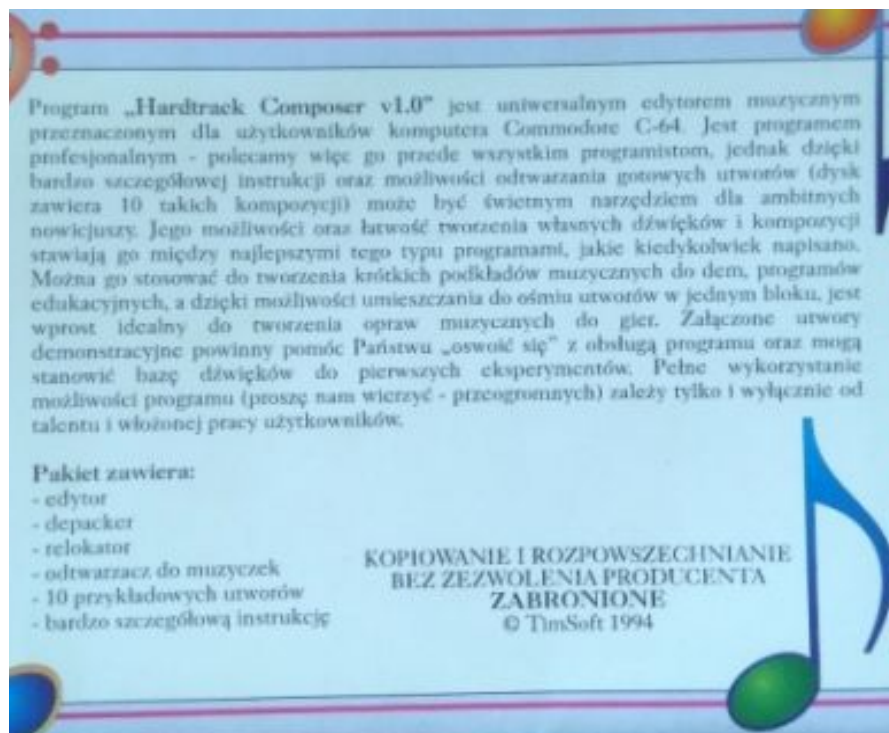
- F7 - jak wyżej, lecz przyspieszenie odtwarzania.

Za pomocą klawisza "<" (strzałka w lewo) możemy przejść do menu operacji dyskowych. Klawisze "+" (plus) i "-" (minus) powodują zmianę aktualnie edytowanego utworu. Za pomocą kombinacji klawisza CONTROL oraz 1, 2 lub 3 możemy zmieniać edycję ścieżki, natomiast te same cyfry wraz z klawiszem COMMODORE wyłączamy ścieżki o określonych numerach. Aby je włączyć ponownie należy rozpocząć odtwarzanie od nowa za pomocą klawisza F1.

Jeżeli chcemy przejść do edycji patternu o numerze, którego numer znajduje się aktualnie "pod" kursorem, należy nacisnąć jednocześnie klawisze SHIFT i RETURN. Klawisze K oraz L służą do zwiększania oraz zmniejszania prędkości odtwarzania utworu. Jeśli chcemy skasować całą zawartość w edytorze, należy użyć kombinacji klawiszy SHIFT oraz "funt".

Możemy również skasować aktualnie edytowaną ścieżkę (COMMODORE + CLR), zawartość wszystkich patternów (COMMODORE + P) lub wszystkie makrorozkazy (COMMODORE + M). Ostatnie operacje dodatkowo wyświetlają okna z pytaniem o potwierdzenie. Oczywiście, należy tu nacisnąć klawisz Y lub N.

Mariusz Wasilewski



enną "x"). Dalej mamy tablicę filtrów. Tym razem w lewej kolumnie należy podać wartość, która będzie dodawana do rejestru układu SID o adresie \$D417 - jest to stopień filtrowania. W prawej kolumnie ustalamy, w ciągu jakiego czasu zmiany mają zachodzić.

Tu również dodatkowo mamy jedną komendę "80 x", która powoduje skok do pozycji w makrorozkazie. Oprócz

poszczególnych kanałów. Trzy kolumny będą tu zawierać:

- aktualną nutę i oktawę,
- aktualną pozycję w ścieżce,
- aktualną pozycję w patternie,
- aktualny instrument.

Podczas edycji dostępnych jest są także wiele funkcji klawiszy, dzięki temu możemy pracować szybko i wygodnie. Oto najważniejsze z nich:



Black Box i Asembler

Cartridge Black Box, poza funkcjami związanymi z magnetofonem, zawiera też monitor Asemblera, dzięki któremu możemy wykonywać dość zaawansowane operacje.

Asembler włącza się za pomocą instrukcji "<-MO" lub poprzez klawisz funkcyjny F2. Powrót do systemu uzyskujemy po wpisaniu X i naciśnięciu klawisza RETURN. Tak najkrócej można zacząć pracować, jednak należy zwrócić uwagę na kilka szczegółów. Przy wprowadzaniu adresów, początkowym jest pierwszy adres w interesującym nas obszarze pamięci. Adres końcowy to natomiast pierwszy następny adres poza wspomnianym obszarem. Wszystkie parametry mogą być rozdzielone dowolną liczbą Spacji, ale nie odnosi się to parametrów w rozkazach: FA, I oraz S.

Za pomocą rozkazu A wywołujemy Asembler, dlatego obok należy podać adres startowy programu. Pewnym ułatwieniem jest fakt, że przy wpisywaniu adresów przy rozkazach nie trzeba używać symbolu oznaczającego liczbę w systemie szesnastkowym, na przykład:

LDA 1C2B

Gdy wystąpi błąd, kursor przeskakuje z powrotem na miejsce "za" adres. Jeżeli chcemy zmienić wpisaną wcześniej linię, na przykład zmienić adres lub instrukcję, musimy ustawić kursor w miejsce, które ma być zmodyfikowane, a po dokonaniu poprawki trzeba nacisnąć RETURN wielokrotnie tak, aby kursor znalazł się w miejscu, z którego przerwaliśmy edycję programu.

Jeżeli mamy wykonać skok, ale nie znamy adresu docelowego, możemy bez problemu wstawić etykietę. Składa się ona z litery Z i z liczby szesnastkowej z zakresu od 00 do 3E, przykładowo:

2005 JMP Z03

Poszczególne bajty nie będące rozkazami, ani adresami rozkazów, przykładowo krótkie tabele, Asembler przyjmuje poprzez podstawienie kropki przed bajtem. Dlatego musimy zachować zapis podobny do takiego:

357A.FF

Etykiety można tworzyć nie tylko dla skoków bezwzględnych jak JSR czy JMP, ale także dla względnych, czyli

dla następujących instrukcji: BPL, BMI, BNE, BVS, BCC, BCS, BEQ.

Gotowy program należy zakończyć znakiem F, po czym naciskamy klawisz RETURN. W tym momencie komputer przeliczy wszystkie adresy i długości skoków.

Za pomocą kolejnej instrukcji B możemy sprawić, że bajty zawarte w określonym obszarze pamięci zostaną umieszczone jako linie programu napisanego w Basicu, w połączeniu z rozkazem DATA. Obok instrukcji B trzeba jako argumenty wpisać początkowy i końcowy adres pamięci, rozdzielając je Spacją.

Odpowiedni program jest tworzony automatycznie. Posiada on charakterystyczną cechę polegającą na tym, iż pierwsza linia danych zawiera numer 32000. Dodajmy, że wszystkie dane są przedstawione jako liczby dziesiętne.

Dane w takiej formie mogą być dołączone do innego programu w Basicu i po jego uruchomieniu można je wczytać za pomocą instrukcji READ. Aby umieścić je w odpowiednim obszarze pamięci wykorzystujemy z kolei polecenie POKE.

SPACJA	-	SYSTEM BLACK BOX
RUN/STOP	-	SYNTEZA POLSKIEJ MOWY I ROZKAZY MUZYCZNE
RETURN	-	PROGRAM DEMONSTRACYJNY
P	-	POMOC

ROM-BIT PRZEDSTAWIA

Dalej mamy instrukcję C, która przyjmuje aż pięć parametrów. Służy do zmiany lokalizacji podprogramów lub całych programów w języku maszynowym. Realizuje dwie funkcje jednocześnie: po pierwsze przesuwa program z jednego obszaru pamięci do drugiego, po drugie - przelicza

wszystkie adresy skoków bezwzględnych JSR lub JMP dostosowując je do nowego obszaru pamięci. Znaczenie kolejnych argumentów jest następujące:

- adres początku obszaru programu, który ma być przesunięty,

- adres końca obszaru,
- adres początku obszaru, do którego ma być przesunięty program,
- adres początkowy całego programu,
- adres końca całego programu.

Należy uważać, aby adres początkowy obszaru, który ma być przesunięty był rzeczywiście adresem pierwszego bajtu rozkazu maszynowego. Jeżeli się pomylimy i pozostawimy na przykład znak Spacji (czyli 120 w kodzie ASCII), zostanie to zinterpretowane jako rozkaz JSR i cała dalsza praca, jak również wszystkie przeliczenia adresów będą nieprawidłowe.

Bardzo ważną instrukcją jest D, która służy do wyświetlania rozkazów Asemblera, pokazuje ona również sposób adresowania. Jako argument należy podać początkowy adres pamięci lub zakres, czyli adres początku i końca obszaru, jaki chcemy sprawdzić.

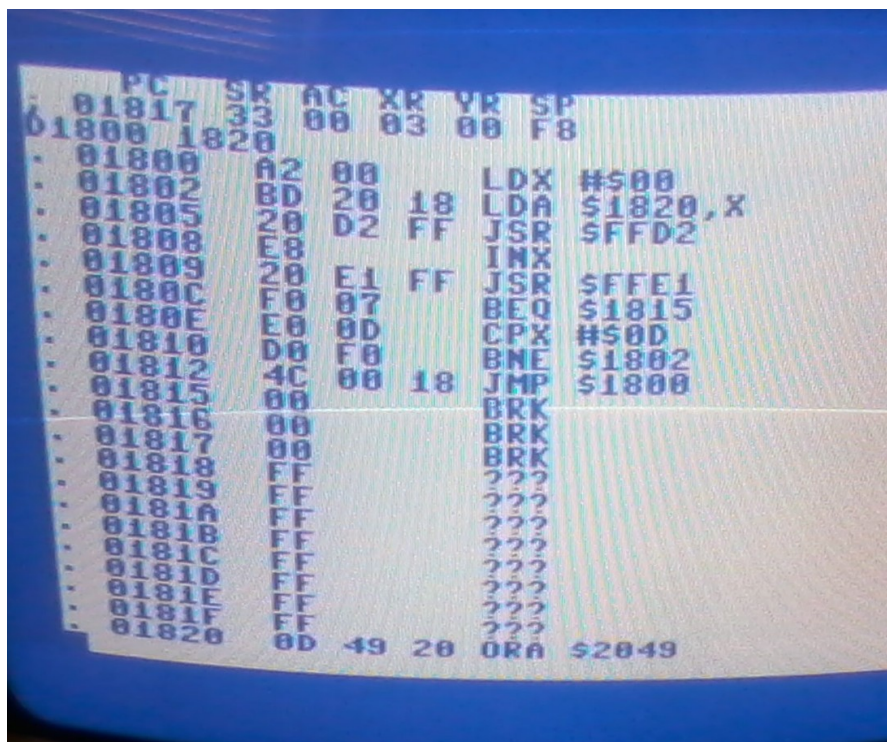
Jeżeli chcemy wyświetlać program w sposób ciągły, naciskamy klawisz RETURN. Dla pojedynczych linii należy użyć klawisza Spacji, natomiast pracę kończymy za pomocą klawisza RUN/STOP.

Następna instrukcja na symbol F i służy do wyszukania określonej kombinacji bajtów w obszarze podawanym analogicznie do poprzedniego polecenia. Po znalezieniu interesujących nas danych, pokazywany jest adres pierwszego bajtu. Gdy przeszukujemy pamięć według wielu kryteriów, nieistotne części bajtów (lub całe bajty) można zastąpić gwiazdką lub dwoma gwiazdkami), czyli na przykład wpisać:

F A*B*AC*

aby wyszukać poniższe bajty:

A0 FF B4 AC A5 78 B5 AC





Możemy też wyszukiwać 3-bajtowe rozkazy, których operandami jest określony adres w podanym obszarze pamięci. Do tego celu służy instrukcja FA, której najpierw podajemy adres, a następnie początek i koniec obszaru. Może to wyglądać tak:

FA1234 2000 2100

Powyższa linia spowoduje wyszukanie następujących rozkazów w obszarze od 2000 do 2100:

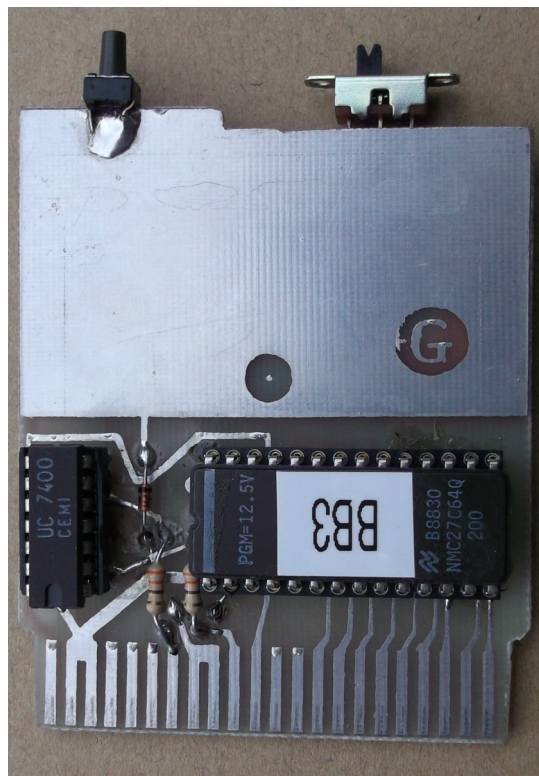
LDA 1234,X
CMP 1234,Y
JSR 1234

Po symbolu FA tym razem nie należy umieszczać znaku Spacji. Dodatkowo, część bajtów ze wskazanego adres można zastąpić gwiazdką, podobnie jak wcześniej. Instrukcja FT wyświetla wszystkie bajty, które nie poddają się operacji deasemblacji. Używamy jej tak samo jak FA, lecz ma tylko dwa argumenty oznaczające adresy - początkowy i końcowy.

Wybrany obszar pamięci możemy wyświetlić w postaci liczb szesnastkowych. Wystarczy użyć instrukcji M, która posiada argumenty podobne do FT.

W tej sytuacji, po prawej stronie pokazywane są dodatkowo znaki ASCII, ale tylko wówczas gdy kod oznacza literę, liczbę lub znak matematyczny. Po wywołaniu instrukcji M na ekranie zobaczymy tylko jeden wiersz.

Aby wyświetlić kolejne należy nacisnąć Spację, natomiast klawisz RETURN powoduje ciągłe listowanie, aż do ponownego przyciśnięcia tego samego klawisza. Po przyciśnięciu klawisza RUN/STOP, na ekranie zobaczymy migający kursor. W tym trybie można wprowadzać modyfikacje w poszczególnych liniach.



Wnętrze cartridge'a.

Instrukcja O pozwala wypełnić obszar pamięci określoną wartością. Jako argumenty należy tu podać zakres pamięci, czyli początkowy i końcowy obszar oraz wartość, na przykład linia:

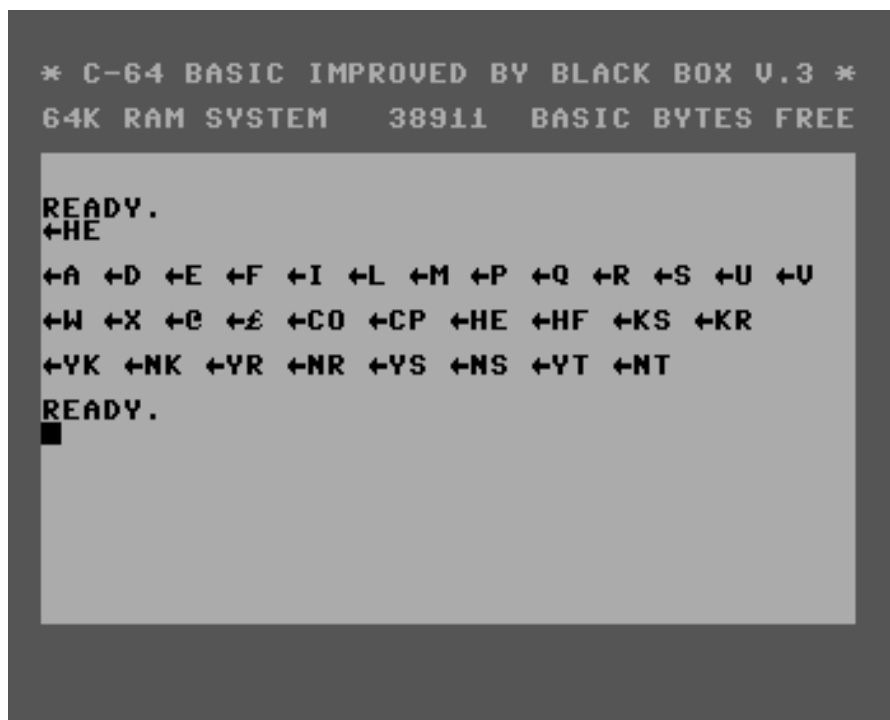
O 2031 2109 EA

oznacza wypełnienie adresów od 2031 do 2109 rozkazem NOP.

Przydatną instrukcją jest także B, która wyświetla zawartość rejestrów procesora, czyli następujących wartości:

- PC - licznika programu,
- SR - rejestru stanu,
- AC - akumulatora,
- XR - rejestru X,
- YR - rejestru Y,
- SP - wskaźnika stosu.

Mariusz Wasilewski



Grafika w trybie 0

Zestaw znaków Atari zawiera liczne symbole graficzne. Można je wykorzystywać w nietypowy sposób, mianowicie w trybie 0, który zwykle jest używany dla programów operujących na tekście. Dlaczego tego nie zmienić?

W przeciwieństwie do trybów od 3 do 11, tryb graficzny 0 ma własny zestaw wbudowanych znaków, które można użyć do tworzenia bardzo elegancko wyglądających projektów. Przypomina to trochę tworzenie własnego zestawu znaków na ZX Spectrum, ale mamy jednak większe możliwości.

Poniższy program powoduje wyświetlenie większości znaków graficznych dostępnych w trybie 0:

```
10 POKE 82,9:POKE 83,25
20 GRAPHICS 0:CHAR=0:LINE=0:PRINT
30 PRINT CHR$(CHAR);CHR$(32);
40 CHAR=CHAR+1:LINE=LINE+1
50 IF LINE=8 THEN LINE=0:PRINT :PRINT
60 IF (CHAR>26 AND CHAR<32) OR (CHAR>155) THEN
PRINT CHR$(27);
70 IF CHAR=32 THEN CHAR=128:PRINT
80 IF CHAR=155 THEN GOTO 40
90 IF CHAR=160 THEN PRINT :GOTO 110
100 GOTO 30
110 POKE 82,2:POKE 83,39:END
```

To, co widać na ekranie, gdy ten program jest uruchomiony, to znaki zestawu ATASCII od 0 do 31, a także od 128 do 159. Kompletną listę zestawów znaków można znaleźć w dodatku do oryginalnego podręcznika Atari Basic. Warto zwrócić uwagę, że znaki od 128 do 159 to znaki wyświetlone po odwróceniu kolorów..

Ekran został wyśrodkowany z dwóch powodów: wygląda lepiej i przy okazji ilustruje sposób umieszczania znaków w innym miejscu niż na lewym marginesie.

W zasadzie pierwszy znak w każdej linii znajduje się na lewym marginesie. Jednak spójrzmy na linię numer 10. Adres pamięci 82 przechowuje informacje o lokalizacji kolumny w stosunku do lewej części ekranu. Po włączeniu komputera pod tym adresem zapisywana jest wartość 2.

Podobnie adres pamięci 83 przechowuje kolumnę prawego marginesu. Wszystko, co tutaj robimy, to zmieniamy marginesy na liczby 9 i 25, natomiast w linii 110 marginesy są przywracane do wartości 2 i 39.

Istnieje wiele innych sposobów umieszczania znaków na ekranie w określonym miejscu, w tym za pomocą instrukcji POSITION, PLOT i DRAWTO. Powiem o tym w dalszej części artykułu.

Jeśli chcemy wyświetlić znak możemy zrobić to na dwa sposoby. Wypisanie litery A można wywołać za pomocą linii:

PRINT "A"

lub

PRINT CHR\$(65)

Obie metody dają takie same wyniki. Druga metoda może być bardzo nieefektywna w przypadku drukowania tekstu na ekranie, ale jest bardzo wydajna w przypadku drukowania znaków z kolejnymi numerami ATASCII i znakami graficznymi.

Jako przykład możemy zastosować poniższą linię:

**FOR I = 65 TO 90:PRINT
CHR\$(I),:NEXT I**



Wyświetlenie znaków od 27 do 31, od 125 do 127, od 156 do 159 i od 253 do 255 wymaga specjalnych instrukcji. Zwróćmy uwagę, że linia 60 sprawdza niektóre z tych znaków. Po odnalezieniu, wypisuje znak 27 (Escape) przed wyświetleniem kolejnego znaku.

Można tu użyć również alternatywnej metody. Spróbuj na przykład usunąć linię 60 i dodać nową linię 15 o następującej treści:

15 POKE 766,2

Dalej zmieniamy linię 110 w ten sposób:

**110 POKE 82,2:POKE
83,39:POKE 766,0:
END**

Po uruchomieniu na ekranie powinny się pojawić cztery krótkie linie. Pierwsza linia powinna mieć symbol Escape. Druga i trzecia linia powinna pokazywać na strzałki do tyłu, z czego jedna będzie odwrócona). W ostatnich dwóch liniach powinny być widoczne symbole kasowania i cztery strzałki.

Podobne efekty można uzyskać w łatwiejszy sposób. Poniższy program wyświetla te same znaki, co poprzedni. Instrukcje POKE 766 powodują, że nie trzeba już naciskać klawisza Escape trzy razy dla każdego znaku, ale tylko raz.

5 POKE 766,2

```
10 PRINT "<ESC>"
20 PRINT "<CLEAR>"
30 PRINT "<BELL>"
40 PRINT "<DELETE>"
50 PRINT "<UP><DOWN>
<LEFT><RIGHT>"
```

60 POKE 766,0



Tryb tekstowy nie musi być nudny!

Oczywiście zamiast wpisów w klamrach „<>” należy naciskać odpowiednie klawisze.

Dowolny znak może być pokazany także za pomocą polecenia PLOT i DRAWTO. Poniższy program poka-

zuje, jak należy używać tych poleceń w trybie graficznym 0.

Każdy znak może zostać umieszczony na ekranie w dowolnym wybranym miejscu. Instrukcja POSITION jest łatwa w użyciu, jeśli

```
10 GRAPHICS 0: POKE 752,1
15 REM USTAWIAMY TRYB I WYLACZAMY KURSOR
```

```
20 PRINT "<DOWN>": COLOR 148
30 PLOT 2,4
35 DRAWTO 36,4
40 DRAWTO 36,18:DRAWTO 2,18
50 DRAWTO 2,3
```

```
60 POSITION 5,7:PRINT "TUTAJ UMIESC SWOJ
NAPIS."
70 POSITION 5,9:PRINT "UPEWNIJ SIE, ZE NIE
BEDZIE MIAL"
80 POSITION 5,11:PRINT "ZBYT WIELU ZNAKOW W
LINII."
90 POSITION 2,20:REM PRZESUWAMY KURSOR
```

```
100 POKE 752,0:REM WLACZAMY KURSOR Z POWROTEM
```

przestrzegamy pewnych zasad: Format tej instrukcji to:

POSITION c, r

gdzie „c” jest żadaną kolumną (liczba od 0 do 39), a „r” jest żadany wiersz (od 0 do 23).

Instrukcja POSITION spowoduje, że następujące po niej polecenie PRINT wypisze wszystkie znaki od wskazanego miejsca.

Znaki po 39 pozycji w kolumnie należy wyświetlać ostrożnie, bo powoduje to „owinięcie” się linii. Przez to kolejne znaki zobaczymy w kolejnym wierszu. Tak samo należy uważać przy ustawianiu pozycji 23 w linii, bo może spowodować przewinięcie ekranu i utratę informacji w pierwszej górnej linii.

Dobrym pomysłem jest też najpierw skasować spacje, które mają być użyte. Następny program (w ramce obok) również używa instrukcji POSITION, ale tym razem pętla służy do zmiany lokalizacji.

Teraz możemy dodać kilka linii, aby zobaczyć, co stanie się z naszym trójkątem:

```
250 FOR C=1 TO 25
260 A=1:B=2
265 GOSUB 300
270 A=2:B=1
275 GOSUB 300
280 NEXT C
290 END

300 SETCOLOR B,7,4
305 SETCOLOR A,7,14
310 FOR D=1 TO 50
320 NEXT D:RETURN
```

Opracował:
Marcin Libicki

```
100 DIM TRIANGLE$(34)
110 GRAPHICS 0:POKE 752,91
115 SETCOLOR 2,7,14:SETCOLOR 1,7,4
120 C=18:R=1
130 TRIANGLE$="<H><J>"
140 POSITION C,R:PRINT TRIANGLE$

150 FOR B=2 TO 32 STEP 2
160 FOR A=2 TO B
170 TRIANGLE$(A,A)="(inverse blank)"
180 NEXT A

190 TRIANGLE$(A,A+1)="(inverse blank)<J>"
200 C=C+1:R=R+1
210 POSITION C,R:PRINT TRIANGLE$
220 NEXT B
230 POKE 752,0
```


atari
online.pl

Serwis informacyjny **24h**
o komputerach ATARI
proponujemy:

- gry i programy użytkowe
- opisy, instrukcje
- historia ATARI, wywiady
- naprawy, rozszerzenia
- programowanie 6502

a także:

- organizujemy stoiska,
- majówki
- odczyty

NAJWIĘKSZY

wyślij dyskietkę
kopertę ze
znaczką





odeślemy
katalog

W Y B Ó R

SERWIS ATARI
monitory przerobki



telefon
696 030 914



Poprawiamy systemowy Reset

Program, który zaprezentuję został opracowany w celu wyłączenia przycisku resetowania systemu. Zmienia on wektor przycisku Reset, aby po jego naciśnięciu program w pamięci został ponownie uruchomiony. Program wykorzystuje zdolność Atari do wskazywania miejsca, do którym powinien zostać wykonany „skok” w takiej sytuacji.

Osiąga się to poprzez instrukcję POKE wskazującą na adres 12 i 13. Najmniej znaczący bajt trafia do lokalizacji pamięci 12, a najbardziej znaczący bajt – do adresu 13. Program również wykorzystuje możliwość Atari do odczytania informacji z ekranu. Instrukcja POKE 842,13 spowoduje, że kursor będzie przesuwany się w dół ekranu, wykonując napotkane instrukcje. Można to przyrównać do ciągłego naciskania klawisza Return.

Pewnym problemem jest umieszczenie na ekranie instrukcji RUN, aby kursor wrócił na swoje miejsce. In-

strukcja POKE 842,12 wyłącza poprzednie zachowanie kursora. Ta dynamiczna technika programowania może mieć wiele zastosowań.

Postanowiłem użyć jej do konwersji instrukcji DATA do pojedynczego ciągu tekstowego, a następnie usunięcia polecenia DATA. W ten sposób trzy linie można łatwo wprowadzić na początku dowolnego programu. Są inne sposoby na wykonanie programu, ale zaletą mojego jest także fakt, że kod języka maszynowego jest relokowalny.

Aby umieścić komunikat RUN na ekranie, program najpierw ukrywa informacje, ustawiając kolor i znaku tak samo jak tło (w liniach 100 i 110). Następnie kasuje ekran (linie 120 i 130), ustawia pozycję kolumny kursora (linie 140 i 150), odczytuje instrukcję RUN oraz dwukrotnie strzałki w górę i umieszcza je na ekranie (linie od 160 do 210). Wreszcie włącza wspomniany ruch kursora poprzez instrukcję POKE 842,13 (linie 220 i 230).

Na końcu instrukcja GRAPHICS 0 przywraca normalny kolor znaków. Jeśli nie chcesz, aby program usuwał instrukcje DATA, nie wpisuj linii od

330 do 370, a w linii 320 zamiast „GOTO 330” wpisz:

RUN

Pamiętaj, że po naciśnięciu przycisku resetowania systemu nie można od razu użyć polecenia SAVE. Gdy zostanie podjęta próba wykonania dowolnej operacji na dysku, zostanie wywołany błąd 130 (nieistniejące urządzenie).

Aby zapisać program, wektor inicjalizacji należy zresetować za pomocą poniższych instrukcji:

POKE12,64

POKE13,21

Następnie naciskamy przycisk resetowania i wykonujemy:

SAVE 'D: nazwa pliku'

Pamiętaj też, że jeśli natychmiast po uruchomieniu wyświetlisz listę programów, a następnie naciśniesz przycisk resetowania systemu, program się zawiesi. Przed uruchomieniem programu polecam zawsze najpierw jego zapisanie.

Opracował:
Kamil Stokowski



```
0 DIM R$(60)
1 GOTO 250
2 I=ADR(R$):H=INT(I/256):L=I!H*256:POKE 12,L:POKE 13,H:POKE
  842,12:GRAPHICS 0
5 ? "WITAJ!":END
10 POKE 12,64:POKE 13,21:? "NACISNIJ PRZYCIISK RESET":
  ? "I ZAPISZ PROGRAM":END

100 DATA 169,148
110 DATA 141,197,2
120 DATA 169,125
130 DATA 32,164,246
140 DATA 169,2
150 DATA 133,84
160 DATA 169,0
170 DATA 32,164,246
180 DATA 169,28
190 DATA 32,164,246
200 DATA 169,28
210 DATA 32,164,246
220 DATA 169,13
230 DATA 141,74,3
240 DATA 76,0,160

250 DIM INSTR$(3):INSTR$="RUN"
260 S=LEN(INSTR$):FOR I=1 TO 16:READ T:R$(I,I)=CHR$(T):NEXT I
270 K=0:FOR I=17 TO 17+5*(5!1) STEP 5:RESTORE 160
280 FOR J=0 TO 4:READ T:R$(I+J,I+J)=CHR$(T):NEXT J
290 K=K+1:R$(I+1,I+1)=INSTR$(K,K):NEXT I
300 FOR I=22+5*(5!1) TO 39+5*(5!1):READ T:R$(I,I)=CHR$(T):NEXT I
310 ? CHR$(125):POKE 766,1:POSITION 2,5:? "1 R$(1,";I1;")=";
  CHR$(34);R$;CHR$(34):POKE 766,0
320 ? "POKE 842,12:GOTO 330":POKE 766,0:POSITION 0,0: POKE
  842,13:END
330 IMAX=90:GOLINE=340:LMAX=360
340 IMIN=IMAX+10:IMAX=IMAX+140:IF IMAX>=LMAX THEN IMAX=LMAX
  :GOLINE=370
350 ? CHR$(125):? :FOR I=IMIN TO IMAX STEP 10:? I:NEXT I
360 ? "POKE 842,12:? :? :GOTO";GOLINE:POSITION 0,0:POKE 842,13
  :END
370 ? CHR$(125):? :? "370":? "POKE 842,12:? :RUN":POSITION 0,0
  :POKE 842,13: END
```


noczesnego naciśnięcie klawisza CTRL i klawisza "↑". Ostatnia linia powoduje skok do linii 40, gdzie program czeka na zmianę położenia wiosełek, a więc wartości zmiennych A i B.

Program nr 2

```
10 N = INT (1.12 *  
    PADDLE (0))  
20 D = 2 * INT  
    (PADDLE (1)/30)  
30 PRINT "NUTA = " ;  
    N ; "ZNIEKSZT.= "  
    ; D  
40 SOUND 0, N, D, 8  
50 IF INT (1.12 *  
    PADDLE (0)) < > N  
    THEN 10  
70 IF 2 * INT (PADDLE  
    (1)/30) < > D THEN  
    10  
80 GOTO 50
```

W programie numer 2 jeden kontroler zmienia nuty, a drugi - poziom zniekształcenie dźwięku. Wartości liczbowe używane przy instrukcji SOUND wyświetlane są na ekranie. Klawisz RESET wyłącza dźwięk. Wiosełka dają tylko 228 pozycji, a wszystkich nut mamy 256, dlatego w ten sposób nie możemy uzyskać dostępu do każdej nuty.

Współczynnik 1,12 w linii 10 pozwala usłyszeć pełny zakres nut, ale jednocześnie pomija niektóre dźwięki. Linia 20 generuje liczby parzyste od 0 do 14 dla wartości zniekształceń. Linie 50 i 60 powodują oczekiwanie na zmiany wartości wiosełek.

Moje programy są proste, ale mogą być całkiem przydatne w pisaniu bardziej rozbudowanych algorytmów. Poza tym pomagają zorientować się w możliwościach uzyskania określonego koloru lub dźwięku, bez potrzeby wpisywania liczb i każdorazowego uruchamiania programu na nowo.

Opracował: Mariusz Wasilewski

Atari VCS

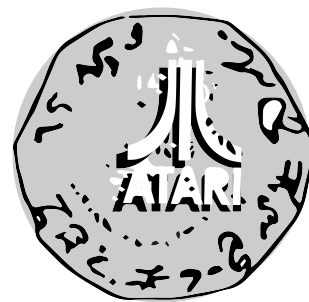
Już 39 maja uruchomiony ma być preorder na nową konsolę do gier (i nie tylko) ze znaczkami Atari! W kolejnym numerze RetroKompabyć może będziemy wiedzieć wiedzieć i jak najszybciej prześlemy Wam te informacje.

Czy sprzęt ze znaczkami Atari będzie znowu w sprzedaży? Wszystko wskazuje, że tak! Konsola została już wstępnie zaprezentowana i ma to być nowoczesny komputer, a nie typowy sprzęt retro. Zastanawiamy się, czy jest dzisiaj możliwe wypromowanie nowej konsoli? Czas pokaże. Z pewnością wszystko zależy od oprogramowania i podejścia producenta do użytkowników.





WYKOPALISKA



Mistrzowski „Team” na Mistrzostwa Świata

Wkrótce zasiądziemy przed telewizorami oglądając naszą i inne drużyny w rozgrywkach Mistrzostw Świata w "nodze" rozgrywanych w Rosji. Tym artykułem chciałbym zmusić Was do odejścia sprzed ekranu TV i przełączania w nim kanału na... STe lub ewentualnie Falcona030 lub TT030.

To właśnie na tych wspaniałych maszynach odpalimy naszego i TYLKO naszego Sensible'a, czyli grę pod tytułem "Team" napisaną przez firmę Impact Software w 1995 roku.



Ekran tytułowy gry.

Jak już wspomniałem i możecie się domyśleć, gra została specjalnie opracowana dla Atari STe, w pełni wykorzystując ulepszony sprzęt drzemający w tej maszynie. Nowe możliwości były często ignorowane przez inne komercyjne firmy zajmujące się produkcją gier.

Mam tu na myśli przede wszystkim dodatkowe kolory, przewijanie ekranu w 50 fps, użycie blittera i dźwięku DMA. Brzmi dobrze? Jeszcze lepiej wygląda. Trzeba dodać, że gra powstała u schyłku popularności maszyn 16-bitowych. Na Atari wychodziły już takie znakomite produkcje, a wspomnieć należy o takich jak:

- Pinball Obsession z kolorowymi i szybkimi stołami,
- opisywany przez Yermieya, kilka numerów wcześniej w RetroKompie, klimatyczny Substation czy Stardust gdzie dogoniliśmy Amigę!

Team był wtedy notowany w ST Formacie bardzo wysoko, bo na poziomie 94%. Dziś nie zmieniam tej oceny.

W grze istnieje kilka trybów gry, możesz wyświetlić, obejrzeć i ustalić wiele stylów gry, takich jak przekazywanie piłki, taktyki i rozgrywka.

Po uruchomieniu właściwej gry, czyli boiska, Team prezentuje się podobnie do kultowego Sensible Soccer. Rozgrywka jest podobna, nie ma co się rozpisywać - po prostu należy to włączyć.

W tym miejscu mam gorący apel - Atarowcy, stwórzmy własną ligę rozgrywk, jak to się robi na Amigach w Sensible Soccer na różnych retro imprezach.

Nie jestem zazdrosny, wszak mamy SS także na ST i na Jaguarze, ale byłoby świetnie zorganizować turniej także w Team. Ostatnio na "kociaku" możemy grać już joystickiem, co jest



Wyjście na murawę.

w przypadku tej gry wygodniejsze. Choć na padzie jaguarowym gra się równie dobrze.

Użytkownicy Atari niezbyt często cieszą się efektem Overscanu w innych grach, a w tej grze jest on zastosowany i służy do drastycznego zwiększenia obszaru gry. Udało się to znakomicie.

Pole gry jest obszerne i wystarczające do wygodnej zabawy. Sprite'y, czyli zawodnicy są dość mali, mniejsi niż w Sensible Soccer, ale ich ruchy są niezwykle szybkie, a ekran przesuwa się płynnie i działa w animacji 50 klatek na sekundę. Robi to wrażenie i gra się bez stresu, że "coś" nam przeskakuje i nie mamy wpływu na nasze plany względem kopania piłki.

Efekty dźwiękowe też są świetne, ale wydaje mi się, że trochę za ciche, co jest raczej dziwne w przypadku gry na stadionie wypełnionym tysiącami ludzi. Na pewno brakuje tu klimatu "wielkostadionowego". Na szczęście reszta oprawy dźwiękowej, czyli muzyka jest doskonała.

To wszystko jest możliwe dzięki procedurom odtwarzania muzyki z odtwarzacza stereo opracowanym przez znaną osobę z demosceny Atari ST, człowieka o ksywie Griff.

Przyznaję wysokie noty grze i zachęcam do jej uruchomienia. Amigowcy mający coraz częściej swoje własne STe, a także Falcons i myślę, że nie będą zawiedzeni "Team". Natomiast każdy szanujący się piłkarz z logiem Atari na koszulce powinien już dawno znać tę pozycję.

(poniżej) Wysoka ocena gry w magazynie ST Format.

simulate the unfairness of real life. You can also change each player's controls (Team can be played from the keyboard or with the Jaguar powerpad).

In the dressing room
Once you've configured the game, you're ready to play. Go to the dressing room screen and pick your side. Here you can alter your team's tactics, depending on whether you want to play a defensive or attacking formation. From there, it's on to the pitch and the game itself.

Once the game kicks off you can dribble, pass, shoot and (more often than not) chase the ball, utilising whichever one of your players is closest. The computer-generated sides are designed to be as realistic as possible, so don't expect Leicester City to compete effectively against Newcastle United. In the predictable world of computer-controlled football, Leicester's victories will be few

and far between when the computer competes against itself. You, of course, bring an entirely new element into the equation. Once you're good enough, you'll be able to make lesser sides trounce previously superior opponents.

One bug that's been fixed since the demo version graced issue 70's Cover Disk is that the screen centres on any player about to be cautioned by the ref, rather than on the ball. This means you no longer lose players without being aware of it because the ball was up the other end of the pitch while your player was being booked for the second time.

At the touchline
You can substitute players, switch them around (useful when a player has been sent off and you want to plug a gap in your defence) and even alter the tactics whenever the ball goes out of play. There's nothing like going 1-0 ahead after five minutes and switching to a 5-4-1 formation to frustrate your friend's efforts to get an equaliser.

Conversely, you can always throw caution to the wind and pick an attacking tactic when your team is struggling to keep on level terms. Just as in real football, this can leave gaps at the back which can be exploited in a breakaway move by the opposition. If you've seen it in real life, chances are you'll encounter it in Team.

If you want to emulate your favourite players' silky skills or crunching tackles, you'll be able to – with practice. The ball responds according to real-life physics and the players and their environment are solid, so you can't just ghost through opponents and other objects as in Sensible Soccer.

It's not easy to calmly place the ball when two defenders are bearing down on you

Team even manages to bring luck into the equation. Just as in real football, the game can be decided by the woodwork or a mix-up in defence. Missing sitters is easy – it's not easy to calmly place the ball in the right place when two defenders are bearing down on you. The atmosphere can certainly affect your judgement – and it's easy to see how that 'player' blasted the ball wide when it would have been so easy to score.

But is it football?
Team succeeds because it's realistic. The graphics are superb (try going back to Sensible Soccer after Team), and make the game a coherent whole, but ultimately Team wins out because of the gameplay. Controlling your players may seem nigh on impossible at first, but practice will enable you to turn 360 degrees just like your opponents. The sensitivity of the fire controls ensures that, just as in the real game, you'll never completely master anything other than short sharp passes. The movement that produces a spectacular dipping 40-yard strike on one occasion will send the next ball high and wide, all because you didn't quite manage to judge the angle and height correctly the second time.

Gripes? It's difficult to find anything to complain about in a game that's been written by someone who's not only a genuine football fan, but also knows more about the STE and Falcon hardware than most. Team oozes class

You can tell what pitch you're playing on by its colour. This is the dry variety. Look out for the extra bounce on the ball.

Team even manages to bring luck into the equation. Just as in real football, the game can be decided by the woodwork or a mix-up in defence. Missing sitters is easy – it's not easy to calmly place the ball in the right place when two defenders are bearing down on you. The atmosphere can certainly affect your judgement – and it's easy to see how that 'player' blasted the ball wide when it would have been so easy to score.

But is it football?
Team succeeds because it's realistic. The graphics are superb (try going back to Sensible Soccer after Team), and make the game a coherent whole, but ultimately Team wins out because of the gameplay. Controlling your players may seem nigh on impossible at first, but practice will enable you to turn 360 degrees just like your opponents. The sensitivity of the fire controls ensures that, just as in the real game, you'll never completely master anything other than short sharp passes. The movement that produces a spectacular dipping 40-yard strike on one occasion will send the next ball high and wide, all because you didn't quite manage to judge the angle and height correctly the second time.

Gripes? It's difficult to find anything to complain about in a game that's been written by someone who's not only a genuine football fan, but also knows more about the STE and Falcon hardware than most. Team oozes class

CREATE A TEAM
If you don't like the idea of Chris Sutton leaving Norwich for Blackburn, or you fancy Martin Tyler's chances against Brazil, you'll be glad to know that you can alter teams or even create your own from scratch. You can not only change the teams' names, manager and players, but also alter their home and away strips, player hair and skin colours and even the team's default tactics. This option includes the facility to load and save team sets (each set takes up a minuscule 15K or so). Look out for future data disks that will ensure Team

slays fully in the picture as the contemporary football game.

Hurrah! Team enables you to recreate your own favourite footballing side. Here's a team we picked completely at random.

Team
Price: £24.95
Contact: Impact Software
Tel: 01280 850450
Min system: STE/Falcon required, 1MByte

Team
HIGHS
■ Excellent graphics
■ Realistic
■ Superb gameplay
LOWS
■ That damn music gets annoying after a while

In short...
Team scores highly in every area and relegates Sensible Soccer to the fourth division.

94%



Product: Team
Price: £24.95
Contact: Impact Software
Tel: 01280 850450
Min system: STE/Falcon
required, 1MByte

Team

HIGHS

- Excellent graphics
- Realistic
- Superb gameplay

LOWS

- That damn music gets annoying after a while

In short...
Team scores highly in every area and relegates Sensible Soccer to the fourth division. **94%**

STIFF VERDICT

ISSUE 72 JULY ST FORMAT **39**

Ocena w magazynie ST Format, wydanie 72.

TEAM: CHelsea
MANAGER: GLENN HODDLE

● G KEVIN HITCHCOCK
● G DMITRI KHARINE
● D STEVE CLARKE
● D ANDY DOW
● D GARETH HALL
● D SCOTT MINTO
● D ANY MYERS
● M MICHAEL DUBERRY
● M GLEN HODDLE
● M ERLAND JOHNSEN
● M JAKOB KJELDBERG
● M DAVID LEE
● M FRANK SINCLAIR
● M DARREN BARNARD
● M CRAIG BURLEY
● M DAVID HOPKIN
● M EDDI NEWTON
● F GAVIN PEACOCK
● F NIGEL SPACKMAN
● F DENNIS WISE
● F ROBERT FLECK
● F PAUL FURLONG

HOME KIT

1ST COLOUR

2ND COLOUR

TACTICS:

4-4-2

DONE

Wybory, wybory... skąd my to znamy?



Sytuacja podbramkowa.

Wszystko świetnie, ale pół roku po wydaniu gry wyszedł kolejny sezon o nazwie - Team: New Season Edition. Zostały w nim poprawione niedoróbki, o których pisałem wyżej.

Usprawniony został dźwięk, a piłkarze nabrali więcej życia i realizmu. Przykładowo teraz potrafią się

cieszyć z udanej akcji, przewracając się o piłkę itd. Dodane zostały też nowe możliwości ustawień taktycznych. Natomiast dla osób, które wolą tak zwane "szmery naturalne", ważną zmianą jest możliwość wyłączenia... muzyki.

I tak z dużą odwagą mogę napisać że po tych zmianach Team: New Season Edition jest prawdopodobnie najlepszą piłką na 16- i 32-bitowe Atari.

Piotr "Piter" Krużycki

TEAM

Wymagania:

Atari STe, Falcon 030,
Atari TT030 z min. 1 MB RAM

Autorzy:

Impact Software

Status:

gra komercyjna

Ocena:

grafika:



dźwięk:



grywalność:



Czasomierz inaczej

Pętla FOR ... NEXT nie stanowi dokładnego licznika czasu, który może być potrzebny w wielu programach. Rozwiązaniem jest użycie wewnętrznych liczników, aby uzyskać bardziej precyzyjne lub opóźnione odliczanie.

Jeśli napiszemy program, który wymaga określonego opóźnienia i oczywiście można to osiągnąć za pomocą wspomnianej pętli. Na przykład tak:

```
10 PRINT "START"
20 FOR X=1 TO 1000
30 NEXT X
40 PRINT "STOP"
```

Jednak zatrzymanie za pomocą komunikatu „STOP” trwa ok. 3 sekundy. W sytuacji, gdy program będzie bardziej rozbudowany, opóźnienie to zwiększy się do 5-6 sekund. Dlaczego tak się dzieje? Pętla FOR ... NEXT jest wykonywana dłużej, jeśli do programu dopiszesz dodatkowe linie.

Dlatego do pomiaru czasu jest lepszy sposób - trzeba użyć procedur języka maszynowego. Będziemy korzystać z adresów od 18 do 20. Zegary te dzi-

alają jak wskaźnik przebiegu na prędkościomierzu samochodu: jeden licznik odlicza czas, a następnie ustawia obok niego kolejny, który znowu tworzy kolejny licznik. Każdy z nich zwiększa wartość, gdy ten z prawej strony przekracza 10. Po uzyskaniu wartości 255 (\$FF) licznik wraca do zera.

Numer rejestru 20 odlicza czas w tempie 60 cyfr na sekundę, aż do liczby 255, a następnie zwiększa rejestr 19 i zaczyna od nowa. Gdy rejestr 19 osiąga 255, zwiększany jest z kolei rejestr 18 o jedną wartość. Jeśli mamy wyzerowane rejestry, zmiana wartości w adresie 18 będzie miała miejsce dopiero po 18 minutach. Aby w lokalizacji pamięci 18 uzyskać wartość 255, musimy odczekać ponad 77 godzin. Jest to więc bardzo uniwersalne rozwiązanie.

Program korzystający z tych funkcji można napisać w następujący sposób:

```
10 PRINT "BEGIN":
   POKE 20,0
20 IF PEEK(20)<180
   THEN 20
```

Procedura ta będzie wykonana przez trzy sekundy, prawie bez względu na

długość programu. Jest to nadal program w Basicu i dłuższy program będzie działał wolniej, ale wpływ długości programu jest w tym wypadku praktycznie pomijalny.

Jak można to wszystko wykorzystać bardziej praktycznie? Spójrzmy na program zaprezentowany w ramce. Informuje on użytkownika o odstępie czasu pomiędzy naciśnięciem przycisku RETURN po wpisaniu RUN i naciśnięciu przycisku RETURN po raz drugi. Zauważmy, że jeśli naciśniemy inny klawisz, komputer powróci do linii 130.

Ten krótki program demonstruje kilka użytecznych funkcji. Najpierw komputer szuka określonego „wejścia”, w tym przypadku klawisza RETURN (czyli ATASCII o wartości 155). Linia 150 odczytuje wartości umieszczone w rejestrach 18, 19, 20.

Natomiast linia 160 zawiera ważną formułę do konwersji informacji zapisanych w tych samych adresach tak, aby otrzymać wynik wyrażony w sekundach.

Na końcu, linie od 170 do 190 konwertują całkowitą liczbę sekund na minuty i sekundy, aby wyświetlona informacja była jak najbardziej czytelna dla użytkownika.



Ten sposób pomiaru czasu powinien być przydatny, niezależnie od tego, czy chcesz dodać limit czasu do swojego algorytmu, czy po prostu potrzebujesz ograniczyć czas odpowiedzi użytkownika. Liczniki mogą mieć zastosowanie zarówno w grach, jak i programach użytkowych.

Polecam także własne eksperymenty na podstawie mojego programu. Wiele rzeczy można jeszcze udoskonalić, mimo że wydają się wykonywać proste operacje. Moja przygoda z Atari dopiero się zaczyna, a już widzę, że ciekawych tematów na kolejne artykuły nie zabraknie.

*Opracował:
Marcin Libicki*

```
10 REM PROGRAM 1
15 REM
20 REM PROGRAM PREZENTUJE MOZLIWOSCI
30 REM WYKORZYSTANIA ZEGARA W ATARI

100 OPEN #1,4,0,"K:"
110 FOR Z=18 TO 20:POKE Z,0:NEXT Z
130 GET #1,D:IF D=155 THEN 150
140 GOTO 130

150 A=PEEK(20):B=PEEK(19):C=PEEK(18)
160 SEC=INT((4.267*256*C)+(B*4.267)+(A/60))
170 MIN=INT(SEC/60)
180 M=MIN*60
190 SEC=SEC-M
200 PRINT MIN;" MIN. ":"SEC;" SEK."
```

 **AMIGA.net.pl**

OFERUJEMY:

Nową literaturę przeznaczoną dla użytkowników komputerów:
COMMODORE, AMSTRAD CPC, ZX SPECTRUM, ATARI, AMIGA

POSZUKUJEMY:

Autorów artykułów pragnących podzielić się wiedzą oraz
doświadczeniami z zakresu użytkowania mikrokomputerów.

WSPÓŁPRACUJEMY:



Gdy Atari nie słucha

Firma Atari swojego czasu wykonała ogromną pracę polegającą na eliminacji większości błędów w pamięci ROM. Tak to wygląda w porównaniu do innych platform. Jednak mimo wszystko w różnych wersjach pozostawiono kilka błędów, zarówno w samym systemie operacyjnym, jak i Basicu. Chciałbym przedstawić kilka z najbardziej znaczących, bowiem pozwalają one zrozumieć zachowanie Atari, gdy pojawiają się sytuacje trudne do wyjaśnienia.

Zacznijmy od mniejszych błędów. Instrukcja wprowadzania danych, której nie podany nazwy zmiennej nie zawsze zostanie oznaczona jako błąd. Tak więc możemy wpisać liczbę lub tekst, a program nie zapisze ich w ramach żadnej zmiennej.

Pętla zawierająca instrukcje LPRINT nie będzie możliwa do zatrzymania za pomocą klawisza Break. Jest to jeden z typowych błędów systemu operacyjnego. Kolejny to blokowanie klawiatury po wywołaniu linii:

```
PRINT A = NOT B
```

Jeśli natomiast zadeklarujemy następującą tablicę:

DIM L (10)

Kod zostanie zapamiętany jako „DIM L10)”, co oczywiście również spowoduje nieprawidłowe działanie programu. Tak się dzieje, bowiem pomiędzy nazwą zmiennej, a rozmiarem tablicy został wpisany odstęp (znak Spacji), przez co linia będzie ona interpretowana niepoprawnie. Tego rodzaju błąd można bardzo często przeoczyć.

Nieprawidłowe wartości przyjmują także funkcje:

```
LOG(0)
CLOG(0)
LOG(1)
CLOG(1)
```

Są to proste błędy podczas zaokrąglania wartości, które widać również przy potęgowaniu. Aby je wyeliminować należy dodatkowo zaokrąglić wynik lub korzystać z liczb całkowitych. Jak wiadomo, nie zawsze jest to wygodne czy możliwe, ale jeśli traficie na wersję pamięci ROM zachowującą się w ten sposób, warto wiedzieć jak sobie poradzić.

Altirra 8K BASIC 0.5

Ready

```

EERRERROERRERERROERERRERERRERRE
ERROR- 128

```

Ready
: 



Czasami podczas edycji linii system zostaje zablokowany. Zwykle występuje to podczas usuwania wielu linii z programu, dlatego tego błędu nie zaobserwujemy przy pisaniu mniejszych programów.

Innym problemem jest fakt, że każdy ciąg tekstowy, który jest dokładną wielokrotnością 256 bajtów, znajdzie się w niewłaściwym miejscu, jeśli zostanie przeniesiony do pamięci. Powoduje to bardzo dziwne błędy i niewytłumaczalne zachowanie programu.

Nie można korzystać z funkcji w linii zawierającej wywołanie USR. Zwraca ona wartość liczbową z zakresu 0 – 65535 i wskazuje na miejsce w pamięci, gdzie znajduje się kod maszynowy. Ten błąd powoduje to, że nie można użyć funkcji do zdefini-

owania adresu. Oczywiście możesz to zrobić w osobnej linii i problem nie wystąpi.

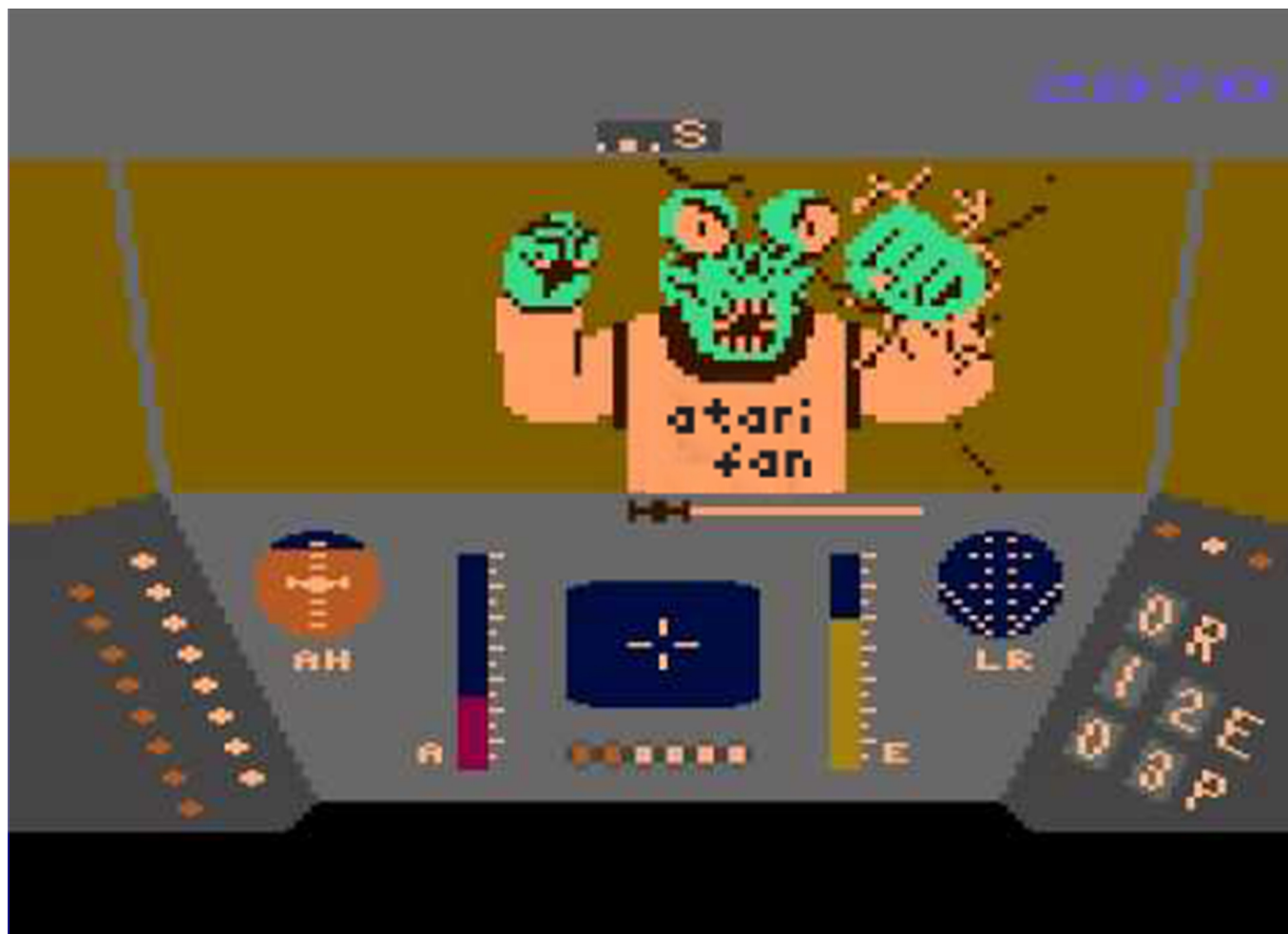
Kolejna sprawa to możliwość deklarowania tablicy większych niż dostępny rozmiar pamięci bez tworzenia błędu. Oczywiście podczas próby uruchomienia programu pojawiają się problemy i komunikaty o błędach, ale sam listing zostanie przyjęty bez system.

Inny błąd jest związany z obsługą stacji dyskietek. Czasami zdarza się, że system przesyła dane do drukarki, przez co stacja przechodzi w dziwny stan bezczynności na kilka sekund. Jeżeli będziemy korzystać z Interface 850, możemy nawet uzyskać część zduplikowanych danych właśnie na drukarce. Z tego samego powodu drukarka może także czasem przes-

tać drukować bez wskazania wyraźnej przyczyny.

Oczywiście wspomniane błędy nie występują we wszystkich wersjach pamięci ROM. Nowsze wersje są większości z nich pozbawione, jednak dzisiaj, gdy wiele osób kolekcjonuje różne modele Atari, z pewnością możemy trafić na bardzo „dziwne” wersje ROM. Dlatego musimy się przygotować na konieczność wymiany kości pamięci, o ile będziemy mieli zamiennik lub zaprogramujemy nową kość. Mnie osobiście ciekawi rozwój systemów operacyjnych, dlatego po natrafieniu na nietypowy sposób zachowania, robię zrzut ekranu lub w inny sposób „archiwizuję” wywołany problem.

Opracował:
Marcin Libicki



Uniwersalne wprowadzanie danych

Za pomocą poniższego programu możliwe jest usprawnienie procedur wprowadzania danych. Działa on na każdym komputerze z serii Apple II z DOS-em w wersji 3.3.

Applesoft Basic nie pozwala posługiwać się znakami przecinków lub dwukropków, gdy wpisujemy dane w odpowiedzi na instrukcję INPUT. Komputer odrzuca wszystko po otrzymaniu znaku przestankowego, co może być bardzo niewygodne. Przykładem jest sytuacja, w której chcesz wprowadzić czas w formacie typu GODZINA:MINUTA:SEKUNDA.

Mój program rozwiązuje ten problem i może być stosowany jako procedura w dowolnym programie w Basicu. Po zainstalowaniu pozwala wprowadzać ciągi zawierające przecinki i dwukropki.

Po wpisaniu programu, proponuję go od razu zapisać. Następnie uruchom go i wpisz dowolny ciąg zawierający przecinki lub dwukropki. Program wypisze ciąg znaków, aby pokazać, że dane wejściowe zostały zaakceptowane bez błędów. Jak widać, funkcję wprowadzania danych można poprawić dość łatwo, a cały program ma tylko ok. 20 linii.

Zauważ, że teraz pytanie jest znakiem „większym niż” (czyli „>”), a nie znakiem zapytania. Oczywiście oznacza to, że standardowe polecenie INPUT wbudowane w Basic nie jest używane.

Jeśli nie podoba ci się używany znak, teraz możesz łatwo zmienić go na inny. Wystarczy odnaleźć kod ASCII dla wybranego znaku, dodać liczbę 128 do jego wartości, a następnie wynik wpisać jako drugą liczbę w linii 270.

Aby użyć tej procedury w swoim programie, powinieneś prześledzić jak działa. Na początku, linia 100 definiuje zmienną tekstową T\$. Jest to ważne, aby była to pierwsza zmienna zdefiniowana przez program i musi być pusta.

Instrukcja POKE w linii 110 zapisuje krótką procedurę języka maszynowego do pamięci. Dane dla tej procedury zawarte są w liniach od 270 do 300. Są to typowe wiersze z instrukcjami DATA.

```

100  CLEAR : HOME
110  GOSUB 4000
120  CLEAR : HOME
130  HCR = UTAB 24
140  GOSUB 3000
150  INPUT "ENTER SUBJECT NAME: ";
160  A$
170  INPUT "BIRTHDAY (MMDDYY): ";
180  B$
190  IF LEN (B$) < 8 GOTO 40
200  INPUT "BIORHYTHM DATE (MMDDYY
210  YY): "; C$
220  IF LEN (C$) < 8 GOTO 50
230  PRINT "STAND BY, WHILST I PLO
240  T =
250  VAL ( LEFT$ (B$,2))
260  VAL ( MID$ (B$,3,2))
270  VAL ( RIGHT$ (B$,4))
280
290
300

```

Linie 120 i 130 wyświetlają monit na ekranie i wywołują nową procedurę wprowadzania danych za pomocą instrukcji GOSUB 190.

Używając tej procedury w swoich własnych programach, powinienś używać podobnego podprogramu, gdy chcesz wprowadzić nowy ciąg. Zwróć uwagę, że ciąg znaków jest zwracany jako zmienna A\$ (patrz linia 140).

Podprogram wywołuje procedurę języka maszynowego (CALL 768), aby wprowadzić ciąg wejściowy do pamięci komputera. Używając procedury GETLN, ciąg jest przenoszony do bufora wejściowego.

Następnie program zapisuje długość ciągu w adresie 798, odejmuje 128 od wartości każdego znaku, aby uzyskać prawidłowe kody ASCII i zwraca wynik do Basic.

Linie 200-260 kopiują ciąg z bufora wejściowego do bezpiecznego miejsca w pamięci, do którego może uzyskać dostęp główny program. Będzie do tego celu używana zmienna tekstowa T\$, którą przywołamy jako pierwszą zmienną zdefiniowaną w programie. W ten sposób możemy znaleźć identyfikator dla zmiennej poprzez użycie instrukcji PEEK oraz adresów 105 i 106.

Jak wiadomo, prosty identyfikator zmiennej składa się z pięciu bajtów w takiej formie, jak w ramce obok.

Poprzez manipulację identyfikatora dla zmiennej T\$, możesz w prosty sposób skopiować ciąg z bufora wejściowego (gdzie zostałby on szybko nadpisany) do innej zmiennej tekstowej (na przykład: A\$).

Po znalezieniu identyfikatora (patrz wiersz 210), jego trzecim bajtem jest długości ciągu (wiersz 220), a

```

100 T$ = " "
110 FOR I = 768 TO 798: READ A
115 POKE I,A: NEXT
120 HOME

130 PRINT "WPISZ COS":GOSUB 190
140 PRINT "=> ";A$
150 PRINT

160 INPUT "DALEJ? (T LUB N) ";YT$
170 IF YT$ = "T" THEN 120
180 END

190 CALL 768
200 B1 = PEEK (798)
210 B2 = PEEK (106) * 256 + PEEK (105)
220 POKE B2 + 2,B1
230 POKE B2 + 3,0
240 POKE B2 + 4,2

230 A$ = MID$(T$,1)
260 RETURN

270 DATA 169, 190, 133, 51, 32, 106, 253,
        142
280 DATA 30, 3, 164, 0, 204, 30, 3, 240
290 DATA 12, 185, 0, 2, 41, 127, 153, 0
300 DATA 2, 200, 76, 12, 3, 96, 0, 0

```

Bajt	Funkcja
1	Pierwsza litera nazwy zmiennej
2	Druga litera nazwy
3	Długość zmiennej
4	Niski bajt adresu pamięci zmiennej
5	Wysoki bajt adresu pamięci zmiennej

czwarty i piąty bajt to instrukcja POKE z adresem bufora wejściowego (linie 230-240). Wartość T\$ jest teraz ustawiona na prawidłową długość, a jej identyfikator wskazuje na bufor wejściowy.

Ostatnim krokiem (w linii 250) jest skopiowanie zawartości zmiennej T\$ do kolejnej - A\$ za pomocą typowej funkcji MID\$, która jest w stanie wyodrębnić każdy znak z T\$.

Oczywiście symbole zmiennych możesz zastąpić zupełnie innymi, ale zasada działania pozostaje niezmienną.

Dzięki temu omijamy standardową instrukcję INPUT i możemy cieszyć się bardziej uniwersalnym sposobem wprowadzania danych do naszych programów. A wszystko dzięki tylko kilku liniom DATA.

Opracował: Kamil Stokowski

Konflikty pamięci graficznej

W tym artykule chcę pokazać kilka sposobów eliminacji konfliktów związanych z używaniem pamięci, które mogą wystąpić, gdy program w Basicu staje się wystarczająco duży, aby zacząć zajmować miejsce zajmowane przez zawartość ekranu w trybie Hires. Oczywiście problemy pojawiają się tylko wtedy, gdy wykonywany jest program wykorzystujący grafikę Hires.

Najpierw musimy zrozumieć układ pamięci, na którym będziemy operować. Programy napisane w Basicu mogą znajdować się w dowolnym miejscu w pamięci od adresu \$800 do \$BFFF. Jeżeli używany jest napęd dyskowy, wtedy górna granica będzie niższa, co praktycznie oznacza adres \$9600.

Dwa ekrany w trybie Hires znajdują się w ustalonych pozycjach w pamięci. Pierwszy to obszar od \$2000 do \$3FFF, natomiast drugi zawiera się w przedziale od \$4000 do \$5FFF. Jeżeli program ogranicza się do obszaru od \$800 do \$1FFF, nie ma żadnego konfliktu.

Pozwala to na uzyskanie wolnej przestrzeni dla programu wynoszącej całe 6 KB.

Jednak program zajmuje nie tylko miejsce przeznaczone dla instrukcji, ale także definiowane są zmienne, które zajmują dodatkowy obszar w pamięci.

Najmniej problemów jest ze zmiennymi tekstowymi, bo są przydzielane z górnej części pamięci, gdzie dostępne jest wiele niewykorzystanego miejsca.

Zwykle zmienne oraz tablice liczbowe i tekstowe zaczynają się na końcu programu. Jeśli będzie ich wystarczająco dużo, mogą zająć obszar grafiki Hires. Jeśli tak się stanie, za każdym razem, gdy ekran w trybie Hires zostanie zmieniony przez HGR lub inne polecenie, jednocześnie zostaną zmodyfikowane lub usunięte zmienne znajdujące się w obszarze ekranu.

W tym momencie można zauważyć, że ze wspomnianej pamięci 6 KB dla programu powinniśmy zmieścić także zmienne i tablice. Szybko pamięci może zacząć brakować. Co z tym można zrobić?

Pierwsze rozwiązanie, które przychodzi do głowy to przełączenie

ekranów i używanie drugiej strony ekranu. Pozwoli to zwolnić dodatkowe 8 KB pamięci. Nie jest to zły pomysł, ale niektóre funkcje trybu Hires nie są w pełni obsługiwane na drugim ekranie, na przykład w trybie mieszanym, gdzie mamy dostępny jednocześnie tekst i grafikę.

W tym przypadku, cztery linie tekstu na dole są zawsze wypełnione "dziwną" zawartością.

Ponadto zmienne możemy przenieść do innego miejsca w pamięci. Aby to zrobić, jako pierwszą linię programu w Basicu należy wpisać:

0 LOMEN:16384

Linia musi zostać wykonana przed przydzieleniem jakichkolwiek zmiennych. W ten sposób zmienne są przechowywane powyżej strony Hires. Jeśli wymagane jest użycie obu ekranów, można zastosować linie:

0 LOMEN:24576

aby rozpocząć przydzielanie zmiennych powyżej drugiej strony ekranu. Należy się także upewnić, że dodatkowa linia programu nie spowodowała rozszerzenia programu w taki sposób, że zajmuje on miejsce poza granicą obszaru Hires.



Tak mogą objawiać się problemy z pamięcią graficzną.

Jeśli sam program jest zbyt duży, należy go przenieść do bardziej obszernego obszaru pamięci, czyli w tym przypadku - powyżej stron Hires. Istnieją dwie strony zerowe, które są kontrolowane, gdy program rozpoczyna się od następujących adresów:

\$67
\$68

Zmieniając te lokalizacje i ponownie ładując program, można go uruchomić w inny sposób. Zmiany te mogą być dokonywane w trybie bezpośrednim. Dopóki Basic nie zostanie zresetowany, zmiany będą zachowane.

Warto dodać, że generalnie tylko adres \$68 musi zostać zmieniony, bowiem adres \$67 będzie ustawiany na 1. Niezbędnym poleceniem jest tutaj:

POKE 104,96

Oznacza on adres startowy \$6001 dla programu. Alternatywnie można użyć linii:

POKE 104,64

aby zmienić lokalizację programu na adres \$4001. Dzięki powyższym liniom ustawiamy nowy adres dla programu, który może być załadowany i uruchomiony. Jednak aby wszystko działało poprawnie, musimy użyć jeszcze jednej instrukcji POKE, bowiem miejsce poprzedzające program musi być ustawione na "zero".

Będzie to adres \$4000 lub \$6000, w zależności od wybranej lokalizacji.

Odpowiednio trzeba tu zastosować linię:

POKE 24576,0
lub
POKE 16384,0

Po wykonaniu tych operacji, w pamięci od adresu \$800 do \$1FFF powstanie puste miejsce, które nie zostanie użyte przez program, ani zdefiniowane zmienne.

Aby program mógł być ponownie wczytany i uruchomiony powyżej pierwszej strony Hires, należy wpisać poniższe instrukcje:

POKE 104,64
POKE 16384,0

Natomiast, aby program mógł zostać uruchomiony powyżej strony drugiej, wpisujemy:

POKE 104,96
POKE 24576,0

Należy pamiętać, że dopiero po wykonaniu tych zmian programy będą znajdować się w innych miejscach pamięci. Zresetowanie Basica spowoduje także przywrócenie domyślnego stanu wskaźników na adres \$800.

Mariusz Wasilewski



Przykładowa grafika Hires w powiększeniu.

Specjalne funkcje klawiszy

Dzięki procedurze podanej w tym artykule możliwe jest przeprogramowanie klawisza Escape lub nawet dowolnego innego. Dzięki temu możemy wykonywać szereg specjalnych funkcji w Basicu. Program działa z każdym komputerem z serii Apple II pod kontrolą i DOS-u 3.2 lub 3.3.

Jeśli nasz program używa instrukcji INPUT, w wielu przypadkach przydatne jest przeglądanie katalogu dyskowego. Może być to nawet konieczne, gdy chcemy, aby użytkownik wprowadził nazwę pliku. W normalnych warunkach nie jest to możliwe, bo podczas oczekiwania na odpowiedź, nie można wpisać i uruchomić polecenia CATALOG.

Mój program pozwala jednak dodać specjalną funkcję klawisza Escape, aby po jego naciśnięciu wyświetlić automatycznie katalog. Dość łatwo, po małych przeróbkach, możesz użyć tej samej procedury, aby nadać specjalne funkcje innym klawiszom lub nawet sekwencjom klawiszy.

Po wpisaniu program numer 1 należy go uruchomić, aby zobaczyć jak działa. Pierwsze dziesięć linii tworzy krótką procedurę języka maszy-

nowego, natomiast linia 110 wykonuje zwykłe polecenie INPUT. Po wyświetleniu monitu należy nacisnąć klawisz Escape. Komputer natychmiast wyświetli katalog dysków, tak jakbyśmy na chwilę "wyszli" poza tryb oczekiwania programu i wpisali polecenie CATALOG w trybie bezpośrednim.

Oczywiście nie jesteś ograniczony do wyświetlania katalogu, bo przez podstawienie różnych poleceń po instrukcji THEN w linii 120, możesz wykonywać inne polecenia, wykonać skok do podprogramu lub wywołać dowolne inne instrukcje.

Klawisz Escape jest jednak idealnie dostosowany do tego celu, ponieważ znajduje się w pobliżu lewego górnego rogu klawiatury i jest rzadko używany. Dlatego program numer 1 sprawdza Escape po instrukcji INPUT, lecz ta technika działa równie dobrze po każdym poleceniu GET. Jeśli chcesz, aby program reagował szybciej reagował na naciśnięcie klawisza, nie musisz czekać na użycie instrukcji INPUT lub GET. Równie dobrze można okresowo sprawdzać stan klawiatury w regularnych odstępach czasu. Taki podprogram może wyglądać na przykład tak:

```
900 IF PEEK (-16384) < 128  
THEN RETURN  
910 POKE -16368,0
```

```
920 IF PEEK (-16384) = 27  
THEN PRINT CHR$(4) "CATALOG"  
930 RETURN
```

Pierwsza linia tej procedury sprawdza, czy naciśnięty został klawisz. Tylko wtedy, gdy odpowiedź brzmi twierdząco, procedura przechodzi do testowania klawisza Escape. Zapobiega to dwukrotnemu reagowaniu komputera na to samo naciśnięcie klawisza. Dzięki temu możesz sprawdzić stan Escape, kiedy tylko chcesz, wstawiając linię:

GOSUB 900

w odpowiednim miejscu w programie. Oczywiście numery linii mogą być zmienione.

Sprawdzanie innych klawiszy jest też proste. Spójrz na linię 60 programu numer 1. Druga wartość w instrukcji DATA reprezentuje CHR\$(27), czyli wartość ASCII klawisza Escape. W celu użycia innego klawisza po prostu zamień liczbę 27 w linii 60 na kod ASCII wybranego klawisza, a następnie dokonaj takiego samego podstawienia w linii 120. Na przykład, aby przetestować literę A, wpisz 65 w liniach 60 i 120. Jeśli ponownie uruchomisz program i naciśniesz A w odpowiedzi na instrukcję INPUT, komputer wyświetli katalog dysku dokładnie tak, jak poprzednio.

Można też zaprogramować podobne działanie, gdy naciśnięty zostanie więcej niż jeden klawisz jednocześnie. Przykładowo, mogą to być następujące kombinacje:

- Escape - wyświetli katalog dla napędu dyskowego nr 1,

- Control + A - wyświetli katalog dla napędu nr 2,

- Control + C - przywróci kursor w domyślne miejsce.

Pierwszym krokiem jest dodanie poniższych linii do programu numer 1:

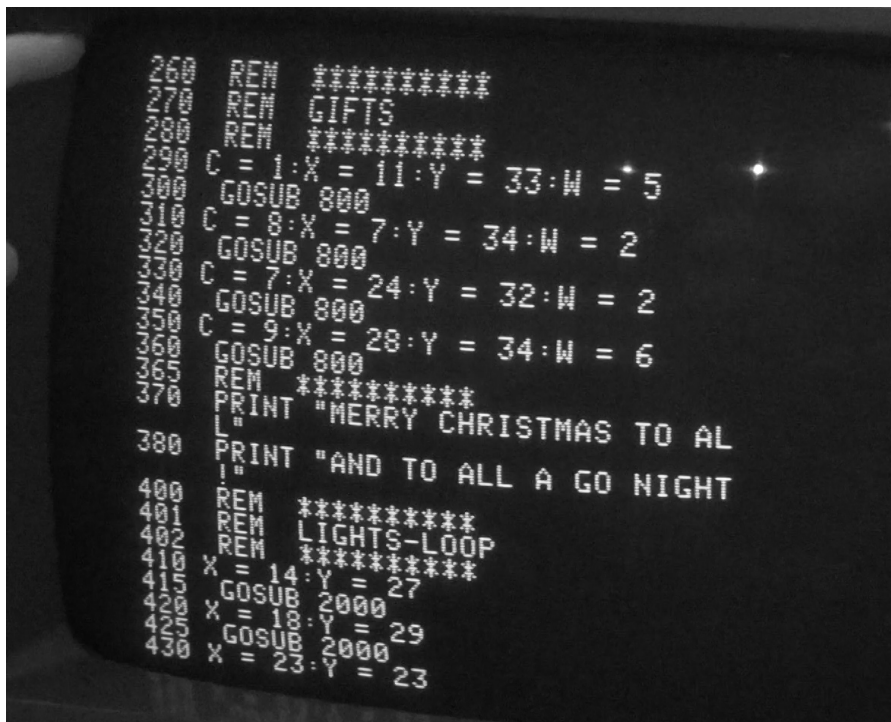
```
58 DATA 201,1,240,10
59 DATA 201,3,240,6
120 IF PEEK (-16384) = 27
    THEN PRINT CHR$(4)
    "CATALOG,D1": GOTO 110
130 IF PEEK (-16384) = 1
    THEN PRINT CHR$(4)
    "CATALOG,D2": GOTO 110
140 IF PEEK (-16384) = 3
    THEN HOME: GOTO 110
```

Zwróć uwagę, że linia 120 jest zamieniana na zupełnie nową. Druga liczba w każdej instrukcji DATA reprezentuje kod ASCII dla danego naciśnięcia klawisza. Ostatnia liczba w każdym ciągu DATA musi być równa ostatniej cyfrze w poniższym wierszu, ale trzeba dodać wartość 4. Linia 59 kończy się wartością 6, a więc wiersz 58 musi kończyć się wartością 10. Linie 130 i 140 dodają instrukcje IF potrzebne do przetestowania kombinacji klawiszy:

CONTROL+A
CONTROL+C

Zgodnie z tym wzorem możesz dodawać własne definicje klawiszy.

Zaprogramowane klawisze zachowują się tak samo jak klawisz RETURN,



powodując natychmiastową odpowiedź na instrukcję INPUT. Istnieją jednak drobne ograniczenia. Po pierwsze, w wielu liniach, takich jak:

```
INPUT A$, B$, C$
```

komputer wykrywa nowe funkcje klawiszy dopiero po ostatnim wpisie - w tym przypadku będzie to zmienna C\$. Po drugie, jeśli polecenie INPUT oczekuje wprowadzenia liczby, przed użyciem klawisza specjalnego należy zawsze wprowadzić dowolną liczbę.

Jedną z wad jest także to, że normalne funkcje klawisza Escape są wyłączone. Normalne zachowanie klawiszy można przywrócić naciskając CONTROL+RESET. Lepszym rozwiązaniem jest jednak, aby program pozostawił komputer w stanie "fabrycznym" dopiero po zakończeniu swojej pracy. Aby to osiągnąć, należy dodać poniższe polecenia na końcu programu:

```
POKE 56,27
POKE 57,253
CALL 1002
```

Teoretycznie można korzystać z tej procedury bez wiedzy o jej działaniu, jednak osoby znające język maszynowy mogą ją zoptymalizować. Aby to lepiej zrozumieć, należy wiedzieć jak standardowo jest sprawdzana klawiatura w komputerze.

Apple odczytuje klawiaturę za pomocą procedury znanej jako RDCHAR (zlokalizowanej pod adresem \$FD35). Wywołuje ona z kolei wywołuje inną procedurę o nazwie RDKEY, która pobiera kod ASCII z klawiatury i zapisuje go w akumulatorze procesora, czyli w rejestrze, gdzie przechowywane są wyniki operacji. Następnie komputer dodaje 128 do kodu klawisza i przechowuje wynikową wartość w adresie \$C000.

Dalej, procedury RDKEY i RDCHAR sprawdzają, czy klawisz został naciśnięty. Jeśli tak, wykonywany jest skok do specjalnej procedury. Naciśnięcie klawisza Escape jest ignorowane, za wyjątkiem sygnału wskazującego, że następne naciśnięcie powinno być rozpoznane w specjalny sposób - jako kod "wyjścia"

zamiast normalnego naciśnięcia klawisza.

W tej sytuacji, zamiast normalnego zachowania, procedura RDKEY przeskakuje do jeszcze jednej procedury - KEYIN, która faktycznie wykonuje całą pracę. Zwykle rozpoczyna się ona od adresu \$FD1B. Ostatni fragment kodu nie jest wywoływany jako podprogram, zamiast tego komputer wykonuje pośrednią instrukcję skoku, przekazując kontrolę do adresu zapisanego w adresach \$38 i \$39. Dokumentacja Apple traktuje tę parę adresów jako przełącznik klawiatury.

Zastąpienie zwykłej procedury KEYIN wersją niestandardową jest proste. Linie od 10 do 70 programu numer 1 wprowadzają nową procedurę do pamięci, zaczynającą się od adresu 768. Linia 80 sprawdza adres pamięci 48640.

Jeśli w tej lokalizacji mamy wartość 76, standardowa procedura wprowadzania danych jest zmieniana za pomocą polecenia INA768, która "mówi", aby uzyskać dane wejściowe z procedury rozpoczynającej się od adresu 768. Procedura ta rozpoczyna się od bajtu 216.

Linie 90 i 100 wykonują przekierowanie wejścia dla DOS-u. Procedura jest relokowalna, a więc możesz umieścić ją w innym miejscu, które nie jest używane przez główny program. Następnie instrukcja CALL 1002 w linii 100 powoduje, że DOS użyje nowej procedury KEYIN zamiast zwykłego kodu. Kod maszynowy ten procedury przedstawiam w programie numer 2.

Pierwsza instrukcja w programie numer 2 jest niezbędna dla systemu ProDOS. Następne osiem linii naśladuje normalną procedurę KEYIN. Te instrukcje w sposób ciągły generują losową liczbę, zwiększając

wartości w adresach od \$4E do \$4F, dopóki nie zostanie naciśnięty klawisz. Po jego naciśnięciu komputer przenosi kod klawisza znaleziony do rejestru - akumulatora, a następnie usuwa bajt stanu klawiatury.

W tym momencie normalna wersja procedury KEYIN powróciłaby do

RDCHAR, która z kolei sprawdziłaby kod, aby zobaczyć, czy naciśnięto Escape. Nasza niestandardowa procedura KEYIN sprawdza to od razu. Jeśli klawisz Escape został faktycznie naciśnięty, zastępujemy zawartość akumulatora kodem ASCII z ustawionym wysokim bitem.

Mariusz Wasilewski

Program nr 1

```

10 J = 768
20 READ A: IF A < > 256 THEN POKE J,A:J = J + 1: GOTO
  20
30 DATA 216,230,78,208,2,230,79,44,0
40 DATA 192,16,245,145,40,173,0,192
50 DATA 44,16,192,72,41,127
60 DATA 201,27,240,2
70 DATA 104,96,104,169,141,96,256
80 IF PEEK (190 * 256) = 76 THEN PRINT CHR$ (4);
  "IN#A768": GOTO 110
90 POKE 56,0: POKE 57,3
100 CALL 1002
110 INPUT "WYBIERZ ";A$
120 IF PEEK ( - 16384) = 27 THEN PRINT CHR$ (4)"CATA
  LOG": GOTO 110

```

Program nr 2

```

          CLD
START     INC $4E
          BNE NEXT
          INC $4F
NEXT      BIT $C000
          BPL START
          STA ($28),Y
          LDA $C000
          BIT $C010
          PHA
          AND #$7F
          CMP #$1B
          BEQ RETURN
          PLA
          RTS
RETURN    PLA
          LDA #$8D
          RTS

```

Crisis Mountain

Gra pod takim tytułem została zaprogramowana w języku maszynowym przez Rona Aldricha i Davida Schroedera w 1982 roku. Jest to świetna produkcja zręcznościowo - logiczna wymagająca Apple II lub Apple II Plus z 48K RAM i napędu dyskowego. Podczas rozgrywki możemy używać joysticka lub pad-dle (czyli wiosełka), które jest dzisiaj raczej zapomnianym kontrolerem. W tym numerze RetroKompapiszemy o nim więcej tak w dziale Atari. Z tego względu warto dodać, że Crisis Mountain jest dostępny również w wersji dlatego komputera.

Scenariusz gry polega na tym, że grupa terrorystów ukrywa się w jaskiniach uśpionego wulkanu na północno-zachodnim Pacyfiku. Wulkan wybuchł niespodziewanie, tym samym zmuszając terrorystów do opuszczenia kryjówki. Gdy uciekali, zostawili tak zwane "łupy", zaopatrzenie oraz - niespodzianka - kilka bomb nuklearnych. Aby ocalić Zachodnie Wybrzeże przed nad-

chodzącą katastrofą, musisz zapuścić się w Górę Kryzysu, wykopać i rozbroić bomby, unikając licznych zagrożeń.

Gra toczy się na kilku scenach w jaskiniach, gdzie musimy przechodzić przez dziewięć poziomów. Na początku otrzymujesz trzy życia. Jeśli jesteś wystarczająco sprawny, możesz zarobić odpowiednio - 10.000, 30.000 i 50.000 punktów.

Na każdym poziomie znajduje się labirynt z wieloma przejściami,

przepaściami i ognistymi dołami lawy, które sporadycznie wypluwają skały i gruz.

Wokół jaskini, oprócz nieszkodliwych przedmiotów pozostawionych przez terrorystów, rozproszone są bomby ustawione losowo w jednym z pięciu miejsc. Każda wyświetla czas pozostały do detonacji, również losowo wybrany.

W miarę przechodzenia od jednego poziomu do drugiego, mamy do czynienia z większą liczbą bomb i



mniejszym czasem, aby je rozbroić. W związku z tym wybieranie odpowiedniej trasy przez teoretycznie znany już labirynt korytarzy, staje się coraz bardziej trudne. Gra wygląda na prostą, ale taka jest naprawdę tylko na początkowym poziomie trudności.

Punkty są przyznawane za wykonanie kilku zadań. Nominalne wyniki są przyznawane za zbieranie łupów, schowków na broń i skrzynek pozostawionych przez terrorystów. Po zebraniu wszystkich przedmiotów, niektóre przedmioty bonusowe pojawiają się w losowych pozycjach w jaskini. Innym sposobem na zdobywanie punktów jest przeskakiwanie głazów. Im większy głaz, tym więcej punktów otrzymasz. Z drugiej strony uderzenie głazem zmniejsza twoją siłę, ale nie od razu tracisz życie.

Poziom siły jest oznaczony numerem od jednego (najsłabszy) do trzech (najsilniejszy). Kiedy jesteś osłabiony, twoje zdolności do punktowania są znacznie osłabione, co jest kolejnym

ciekawym urozmaicheniem. Ogólnie rzecz biorąc, na poziomie najmniejszej siły punktacja staje się druzgordna w stosunku do przetrwania, ponieważ rzadko można przeskoczyć głazy w tym stanie. Na szczęście w jaskini jest kilka bezpiecznych zakamarków, w których można się zatrzymać i nabrać więcej siły.

Mamy inne sposoby na utratę życia w Crisis Mountain. Możesz spaść lub zostać wepchniętym do lawy przez głaz, może też zostać zdetonowana bomba. Na koniec zostawiam utratę życia przez ugryzienie śmiercionośnego nietoperza. Prawdziwy horror.

Nietoperz jest najbardziej niezwykłym elementem i przy okazji - najbardziej denerwującym. Porusza się po jaskini w sposób przypominający prawdziwego nietoperza. Jeśli głaz zostanie wysadzony z pobliskiego dołu lawy, nietoperz rzuci się w jego kierunku, by sprawdzić, czy kamień nie jest ofiarą, aby zaraz odlecieć do innej części jaskini.



Obecność nietoperza to nie tylko prosta "przeszkadzajka". Jego obecność powoduje, że gra jest trudniejsza, bowiem ma on prawdziwy talent do określania miejsca gracza w praktycznie dowolnym momencie gry. Czasami możemy uniknąć spotkania, ale w wielu przypadkach ucieczka jest niemożliwa.

Gra jest nietypowa, chociaż na pierwszy rzut oka na taką nie wygląda. Autorzy gry włożyli ogromny wysiłek w zaprojektowanie grafiki wysokiej rozdzielczości. Wygląda ona lepiej niż wersje dla innych maszyn, posiadają wiele skomplikowanych szczegółów.

Podobnie na pozytywną ocenę zasługuje animacja, która jest realistyczna jak na warunki 8-bitowego komputera. Dzisiaj możemy tego nie dostrzegać, ale postawmy się w roli entuzjasty tego sprzętu sprzed lat.

Ponadto mamy tu kilka ciekawych funkcji dodatkowych. Po pierwsze, klawisz ESC pozwala zatrzymać lub wznowić grę w dowolnym momencie. Rozgrzywka może potrwać długo, nawet ponad godzinę, dlatego pauza jest wygodna lub nawet niezbędna do utrzymania koncentracji. Niestety nie ma możliwości zapisania stanu gry, chyba że będzie korzystał z dostępnych emulatorów, czego osobiście nie polecam.

Efekty dźwiękowe są dobre, ale z czasem mogą być przeszkadzające. Na szczęście możesz je wyłączyć za pomocą kombinacji klawiszy CTRL + S. Na dysku można zapisać najwyższy osiągnięty wynik, dzięki czemu Crisis Mountain jest świetną pozycją do turniejów na imprezach retro. Gra jest bardzo wciągająca i pokazuje szerokie możliwości komputerów Apple.

Opracował:
Mariusz Wasilewski

Wprowadzenie do RiscOS

Gdyby chcieć opisać RISC OS jednym zdaniem, to wystarczyłoby zapewne powiedzieć, że jest to system operacyjny przeznaczony dla komputerów z procesorem ARM. Nie sposób jednak pisać o tym fantastycznym środowisku bez choćby krótkiego wprowadzenia w historię komputerów spod szyldu Acorn Computers.

RISC OS został opracowany pierwotnie przez brytyjską firmę Acorn na potrzeby własnej linii komputerów Archimedes, opartych o autorski projekt procesora, nazwanego ARM - co jest akronimem od słów Advanced RISC Machine. Była to niezwykle zaawansowana konstrukcja, wprowadzająca wiele innowacyjnych rozwiązań, jak na przykład przyrostki rozkazów, warunkujące wykonanie instrukcji procesora.

Przykładowo, instrukcja "MOV R0, R1" przesyłająca dane z rejestru R1 do rejestru R0 może mieć postać "MOV CC R0, R1" co oznacza, że zostanie ona wykonana jedynie wtedy, gdy flaga przeniesienia będzie wygaszona (Carry Clear). Podobnie możemy zażyczyć sobie wykonania operacji tylko w przypadku, gdy po-

jawi się nadmiar - wówczas instrukcja przyjmie postać "MOV VS R0, R1".

Przyrostki warunkowe można dodawać do większości rozkazów, można je również dodatkowo łączyć z przyrostkiem "S", który mówi, czy flagi rejestru stanu powinny być aktualizowane po wykonaniu instrukcji przez procesor. I tak na przykład "SUB R0, R1, R2" odejmie od siebie zawartość rejestrów R1 i R2, a wynik umieści w R0, ale nie dowiemy się, czy wynikiem operacji jest zero. Dowiemy się tego natomiast w przypadku, gdy zamiast "SUB" użyjemy "SUB S".

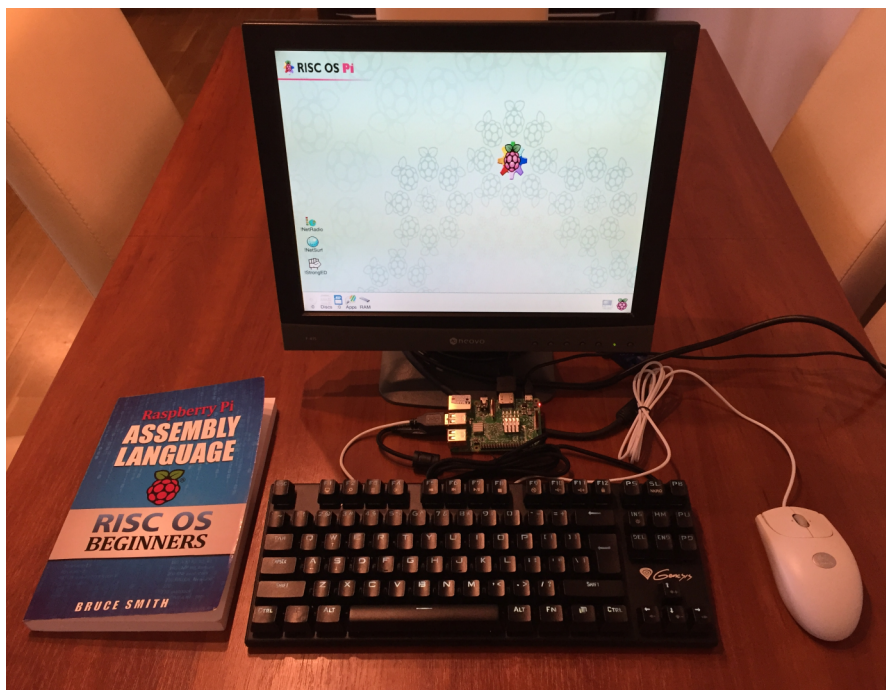
Możemy również wykonać odejmowanie i ustawić flagi tylko wtedy, gdy wynik wcześniejszej operacji był dodatni - wówczas skorzystamy z instrukcji w postaci "SUB PL S R0, R1, R2". Dzięki tak ogromnej elastyczności rozkazów programy pisane w języku maszynowym mogą być o wiele krótsze na ARMie niż na innych procesorach, a przez to działać o wiele szybciej.

Kolejną zaletą procesorów ARM, którą konkurencyjni producenci wprowadzili dopiero wiele lat później, jest potokowe przetwarzanie rozkazów. Potok we wczesnych wersjach procesora ARM był trójstopniowy: w czasie gdy jeden rozkaz był wykonywany, następny w kolejce był

w tym samym czasie dekodowany, a kolejny pobierany z pamięci. Architektura potokowa, w połączeniu z elastyczną listą rozkazów, pozwalającą użyć mniejszej liczby rozkazów do osiągnięcia tego samego efektu, czyniły z ARMa najszybszy istniejący wówczas na świecie procesor przeznaczony dla komputerów domowych.

W 1987 roku Amiga 500 z procesorem MC68000 7 MHz mogła się poszczycić (według programu Sys-Info) wydajnością niewiele przekraczającą 0,5 MIPSa. Wprowadzony w tym samym roku do sprzedaży Archimedes A310 z procesorem ARM2 8 MHz uzyskiwał wynik 4,8 MIPSa!

Niestety pomimo nowoczesnej, 32-bitowej architektury, Archimedes nie zdobył większej popularności. Przyczyną takiego stanu rzeczy była przede wszystkim cena - maszyny te były wielokrotnie droższe od innych komputerów domowych, w tym pecetów. Firmie Acorn udało się jednak wykorzystać swoją pozycję na brytyjskim rynku edukacyjnym, którą zdobyła dzięki kontraktowi z telewizją BBC i komputerom z serii BBC Micro, do promocji swojej nowej konstrukcji. Ostatecznie Archimedes znalazł miejsce prawie wyłącznie w angielskich szkołach i na uczelniach, natomiast w Polsce pozostał praktycznie nieznany.



Raspberry Pi z systemem RISC OS 5.

Jedynymi komputerami firmy Acorn, które dotarły do naszego kraju, były ośmiobitowe Electrony, sprzedawane na Śląsku w sklepach dla górników. Pochodziły one z wyprzedaży magazynów, kiedy to po przegranej wojnie z firmą Sinclair o rynek komputerów domowych, Acorn pozostał z dziesiątkami tysięcy niesprzedanych egzemplarzy. Historia tej fascynującej rywalizacji stała się kanwą dla fabularyzowanego dokumentu "Micro Men", nakręconego w 2009 roku przez BBC. Można go bez problemu odnaleźć na Youtube, a dla osób nie znających biegle języka angielskiego na forum speccy.pl dostępne są napisy w języku polskim.

Film charakteryzuje się nie tylko doskonałym scenariuszem i świetną obsadą aktorską, ale także obecnością prawdziwych bohaterów tamtych wydarzeń, którzy pojawiają się jako postacie drugoplanowe w pojedynczych epizodach. Na przykład Sophie Wilson (w filmie pokazana jeszcze przed zmianą płci, jako Roger Wilson) gra w ostatniej scenie

filmu właścicielkę pubu, w którym spotykają się Chris Curry (założyciel Acorna) i Sir Clive Sinclair.

Wróćmy jednak do systemu RISC OS. Jest to system wielozadaniowy, ale oparty na wielozadaniowości kooperatywnej, podobnie jak Windows 3.x czy wcześniejsze wersje MacOS. Ma to zarówno swoje wady, jak i zalety. Wadą jest z pewnością fakt, że cały system może zostać całkowicie zablokowany przez jeden proces, który przez dłuższy czas nie zwalnia czasu procesora. Stawia to również dodatkowe wymagania przed programistą, który podczas tworzenia programu musi cały czas pamiętać, aby raz na jakiś czas zwracać sterowanie do systemu.

Z drugiej strony, ten typ wielozadaniowości jest mniej kosztowny (nie jest konieczne częste przełączanie kontekstu pomiędzy trybem użytkownika, a trybem nadzorczy), więc system działa wydajniej na słabszych procesorach. Możliwe jest również celowe zajęcie całego czasu procesora przez

pojedynczy program, co pozwala na tworzenie oprogramowania czasu rzeczywistego. Jest to niezbędne np. w automatyce czy systemach pomiarowych, gdzie komputer musi odpowiedzieć w zadanym czasie i niedopuszczalna jest sytuacja, że system "zamyśli się" nad innym zadaniem. RISC OS nadaje się więc znakomicie jako baza dla programów sterujących na przykład robotami. Jeżeli dodamy do tego możliwość uruchomienia RISC OS na Raspberry Pi to sami przyznacie, że zaczyna robić się naprawdę ciekawie.

Historia systemu RISC OS pod wieloma względami przypomina historię systemu Amiga OS, ponieważ jest równie zawiła. Acorn rozwijał RISC OS aż do wersji 3.7, która ujrzała światło dzienne w 1996 roku. Po upadku firmy Acorn prawa do systemu operacyjnego wykupiły trzy spółki: Castle Technology, Pace Micro Technology oraz RISCOS Ltd. Ta ostatnia wypuściła w 1999 roku zaktualizowaną wersję systemu, oznaczoną numerem 4.02. Nowa wersja systemu RISC OS została wydana w postaci kości ROM oraz płyty CD z oprogramowaniem. Wersja 4.03 była już montowana fabrycznie w konstrukcjach będących następcami komputerów z serii Archimedes: RISC PC, RiscStation, A9Home i Ionix.

Producentem tych ostatnich była wspomniana wcześniej firma Castle, która po wykupieniu praw do RISC OS od firmy Pace, wydała wersję 5 systemu. W październiku 2006 roku Castle postanowiło udostępnić kod źródłowy RISC OS 5 społeczności, w wyniku czego powstała dedykowana tej inicjatywie spółka RISC OS Open Limited (w skrócie ROOL), która w krótkim czasie doprowadziła do opracowania wersji systemu na nowe urządzenia bazujące na procesorach ARM, takie jak BeagleBoard, PandaBoard, czy Raspberry Pi. Na-

nowsza wersja tej gałęzi systemu nosi oznaczenie 5.23 i można ją za darmo pobrać ze strony ROOL pod adresem: <https://www.riscosopen.org> w postaci obrazu do nagrania na kartę SD.

Jednocześnie firma RISCOS Ltd. kontynuuje prace nad następcą wersji 4 systemu. Efektem tych prac jest wersja 6, znana także jako RISC OS Six. Jej celem nie jest, jak w przypadku wersji 5, obsługa najnowszego sprzętu, ale dostarczenie najnowszego oprogramowania użytkownikom komputerów klasy RISC PC oraz ich emulatorów, takich jak np. Virtual Acorn. Dlatego RISC OS 6 nie jest już dostarczany (jak jego poprzednik) w postaci fizycznych kości ROM, ale w postaci ROM-ów ładowanych programowo do pamięci.

Przypomina to nieco mechanizm MapROM znany z kart turbo do Amigi. Tak więc mamy dwie gałęzie tego samego systemu, których numeracja wcale nie odzwierciedla chronologii jego rozwoju. Na dodatek oba warianty systemu rozwijane są przez dwie różne firmy, których interesy nie zawsze są do końca zbieżne.

Oprócz podobieństw do Amigi jest też jednak wiele różnic. Oprócz wspomnianej już wielozadaniowości kooperatywnej (AmigaOS wykorzystuje model wielozadaniowości z wyłączeniem) RISC OS stosuje nieco inną filozofie interfejsu użytkownika. Przede wszystkim wykorzystywane są trzy klawisze myszki, które mają nieco inne funkcje niż w większości innych systemów operacyjnych. Lewy przycisk myszy nosi nazwę "select" i jego działanie jest najbardziej podobne do tego, do czego jesteśmy przyzwyczajeni - wybieramy nim elementy lub uruchamiamy aplikacje podwójnym kliknięciem.

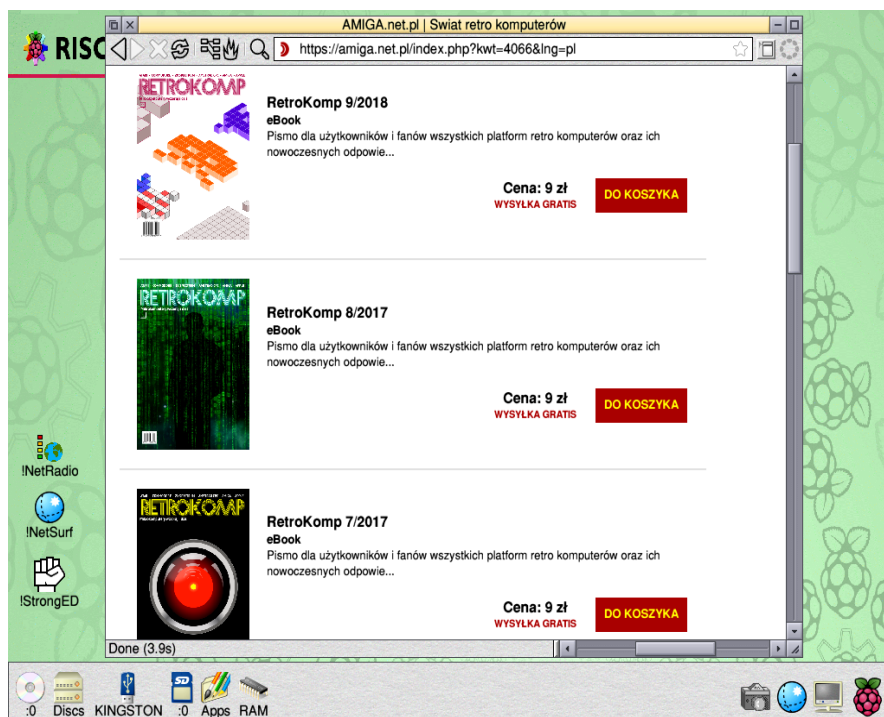
Prawy przycisk nosi nazwę "adjust" i stanowi swego rodzaju uzupełnienie przycisku "select" - możemy nim zaznaczać wiele elementów po prostu klikając na nie (w innych systemach aby osiągnąć taki sam efekt przytrzymujemy jakiś klawisz, na przykład SHIFT lub CTRL), możemy nim również "odwracać" działanie przycisku "select". Na przykład kliknięcie "select" na ikonie znajdującej się w lewym górnym rogu okna powoduje jego schowanie za pozostałymi oknami, natomiast ta sama akcja wykonana przyciskiem "adjust" przywraca je na pierwszy plan.

Podobnie otwierając nowy folder za pomocą "select" otworzymy go w nowym oknie, natomiast gdy tę samą akcję wykonamy przyciskiem "adjust", żądany folder otworzy się w aktualnym oknie. Jak zatem powrócić do poprzedniego katalogu? To proste - wystarczy zamknąć okno przyciskiem "adjust" zamiast "select". Komplementarne funkcje obu przycisków możemy też wykorzystać np. do szybkiego zaznaczania bloków tekstu w różnego rodzaju edytorach - "selectem" oznaczamy początek

bloku, a "adjustem" jego koniec. Środkowy przycisk myszki nosi nazwę "menu" i służy do wywoływania menu kontekstowego. Jeżeli korzystamy z myszki dwuprzyciskowej z rolką, menu wywołujemy naciskając rolkę.

Jeśli jednak mamy myszkę, która posiada tylko dwa przyciski, możemy się posilkować programem !WinMenu. W systemie RISC OS 5 znajdziemy go w katalogu "Utilities". Po jego uruchomieniu menu kontekstowe wywołamy naciskając klawisz "windows" (lub jego odpowiednik) na klawiaturze. Obsługa trzech przycisków myszy zamiast dwóch może się początkowo wydawać skomplikowana, ale po jej opanowaniu bardzo szybko staje się drugą naturą. Muszę przyznać, że wracając do innych systemów często brakuje mi możliwości, które daje w tym zakresie RISC OS.

Kolejnym ważnym elementem systemu RISC OS jest tzw. Icon Bar. Jest to belka znajdująca się na dole ekranu, pełniąca podobne funkcje jak w innych systemach operacyjnych.



Jednak w RISC OS pojawiła się ona znacznie wcześniej niż u konkurencji. Znajdziemy na niej symbole oznaczające podłączone napędy, skrót do folderu z najważniejszymi aplikacjami, a także ikony uruchomionych programów.

Najważniejsza dla systemu ikona znajduje się na prawym krańcu belki i jest oznaczona symbolem systemu, na którym uruchomiony jest RISC OS - w przypadku Raspberry Pi ma ona oczywiście postać czerwonej malinki. Klikając na nią przyciskiem "select" lub "adjust" możemy zobaczyć listę uruchomionych procesów wraz z ilością zajmowanej przez nie pamięci. Z kolei poprzez przycisk "menu" dostaniemy się do opcji konfiguracyjnych systemu, będziemy go też mogli zrestartować lub wyłączyć.

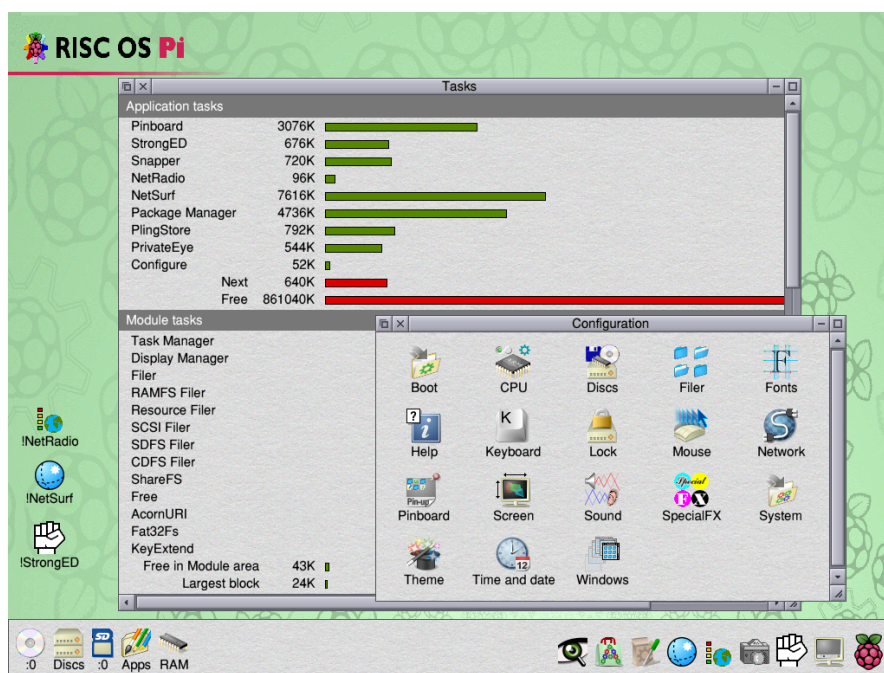
Równie ciekawie wygląda w RISC OS uruchamianie aplikacji. Po dwukrotnym kliknięciu na ikonę programu przyciskiem "select" możemy na pierwszy rzut oka nie zauważyć żadnego efektu. Program uruchamia się bowiem "w tle", a jego ikona pojawia się na belce Icon Bara. Jeżeli program posiada interfejs użytkownika, możemy go aktywować klikając na ikonie przyciskiem "select". Możemy również wykorzystać przycisk "menu" myszki, aby dostać się do opcji konfiguracyjnych (nazywanych często w nomenklaturze RISC OS "choices") lub go wyłączyć. Zastosowana jest więc tu filozofia znana z systemu MacOS, gdzie program działa niezależnie od swojego interfejsu graficznego, a zamknięcie okna nie jest jednoznaczne z zakończeniem działania aplikacji.

Aplikacje łatwo odróżnimy od folderów, ponieważ ich nazwy rozpoczynają się od znaku wykrzyknika: !WinMenu, !StrongEd, itp. Również podobnie jak w systemie MacOS, aplikacje są tak naprawdę katalogami, w których znajdziemy sam program

oraz różnego rodzaju skrypty, biblioteki, sprajty i inne niezbędne do jego działania pliki. Jeżeli chcemy podejrzeć zawartość katalogu aplikacji, wystarczy kliknąć na niego dwukrotnie przyciskiem "select", trzymając jednocześnie wciśnięty klawisz SHIFT.

W systemie RISC OS bardzo duży nacisk położono na działanie w myśl filozofii "przeciągnij i upuść". Jest ona

Cóż, wystarczy po prostu przeciągnąć jego ikonę do wybranego przez nas folderu. Czy można prościej? Przeciągać i upuszczać w RISC OS możemy prawie wszystko, chcąc na przykład skopiować adres URL z przeglądarki do edytora po prostu go tam przeciągamy. Nie mamy natomiast do dyspozycji kosza na śmieci. Pliki usuwamy bardziej tradycyjnie - klikając na nie przyciskiem "menu" i wybierając opcje "delete".



Systemowy menadżer zadań.

do tego stopnia zakorzeniona w świadomości użytkowników tego systemu, że nawet największy obecnie magazyn skupiający społeczność RISC OS nosi nazwę "Drap 'N Drop".

Wiele operacji możemy wykonać tylko poprzez przeciąganie ikon, dlatego możemy się mocno zdziwić, próbując na przykład po napisaniu tekstu w edytorze zapisać plik na dysku. Po otwarciu okna zapisu zobaczymy okienko z ikoną i nazwą pliku, którą możemy oczywiście zmienić. Jak jednak zapisać plik w konkretnym katalogu?

Skoro jesteśmy już przy plikach, w tym zakresie RISC OS również jest indywidualistą. Pliki mogą posiadać nie tylko atrybuty (protected, unprotected, private, public, etc), ale także typy, które pozwalają systemowi operacyjnemu powiązać plik z konkretną aplikacją. O ile w systemach firmy Microsoft takie powiązanie następuje na podstawie trzyliterowego rozszerzenia, tutaj stosowane są trzyliterowe kody szesnastkowe. I tak na przykład plik typu "FFF" to plik tekstowy. Większość typów posiada swoje przyjazne odpowiedniki, więc jeśli chcemy aby utworzony przez nas plik był trak-

towany jako program w Basicu, możemy mu po prostu ustawić typ "Basic" zamiast "FFB". Takie przyjazne nazwy są oczywiście o wiele prostsze do zapamiętania niż trzyliterowe kody czy rozszerzenia. System typów i atrybutów sprawia jednak, że przy kopiowaniu plików na przykład na kartę SD z systemem Fat32, dane te są tracone. O ile w przypadku pojedynczych plików można sobie jeszcze jakoś poradzić, o tyle odtwarzanie atrybutów plików wchodzących w skład aplikacji może być bardzo trudne. Pamiętajmy, że programy w systemie RISC OS są tak naprawdę katalogami zawierającymi zbiory plików.

Dlatego jeżeli pobieramy z Internetu oprogramowanie dla systemu RISC OS, to należy to robić albo bezpośrednio z poziomu samego RISC OS (o tym za chwilę) albo ściągnąć archiwum zip i rozpakować je dopiero w docelowym środowisku RISC OS. Dodatkową trudnością jest fakt, że w systemie RISC OS kropka "." jest separatorem ścieżki, natomiast znak "/" oddziela nazwę pliku od rozszerzenia - a więc dokładnie odwrotnie niż w zdecydowanej większości innych systemów operacyjnych.

Dlatego rozpakowanie archiwum z plikami przeznaczonymi dla RISC OS w innym systemie spowoduje nie tylko utratę metadanych dotyczących plików, ale najprawdopodobniej zupełnie niepoprawnie odtworzy ich hierarchię i nazwy.

Cechy RISC OS, które opisałem do tej pory, są wspólne dla wszystkich wersji systemu. W wersji 5 doszła obsługa sieci TCP/IP (oprócz natywnego dla komputerów firmy Acorn protokołu Econet), a co za tym idzie - Internetu. W związku z tym w RISC OS 5 oprócz przeglądarki Netsurf znajdziemy też wbudowany menedżer pakietów o nazwie !Pack-

Man, w którym dostępne są głównie pakiety przeportowane z open-source'owych narzędzi GNU, a także sklep !Store, z którego możemy pobrać zarówno aplikacje płatne, jak i freeware. Zawartość obu nie jest zbytnio imponująca, dlatego na końcu artykułu podaję listę stron, które warto odwiedzić szukając oprogramowania dla systemu RISC OS. Ściągając pliki za pomocą przeglądarki pamiętajmy, że aby zapisać je na dysku, należy po pobraniu pliku przeciągnąć ich ikonę do folderu docelowego.

Jak już wcześniej wspomniałem, RISC OS może być używany nie tylko na klasycznych komputerach klasy Archimedes czy też RISC PC, ale również na całkiem współczesnym sprzęcie. Pod tym względem użytkownicy RISC OS są w zdecydowanie bardziej komfortowej sytuacji niż na przykład użytkownicy AmigaOS 4 czy FreeMiNTa, ponieważ mają do dyspozycji bardzo przystępną cenowo i popularną platformę sprzętową w postaci Raspberry Pi. Jest jeszcze MorphOS, którego możemy zainstalować na starszych Macach, ale w tym wypadku jesteśmy z kolei skazani na sprzęt używany, którego dostępność z roku na rok maleje. W przypadku Raspberry Pi nie musimy się tego obawiać jeszcze przez długie lata.

Nie ma jednak róży bez kolców. Po pierwsze Raspberry Pi jest względnie słabym sprzętem. Został on zaprojektowany jako komputer edukacyjny, a nie stacja robocza, w związku z czym jego konstrukcja jest wynikiem wielu kompromisów pomiędzy wydajnością a ceną. Niedawno miałem okazję przekonać się o tym na własnej skórze, pisząc ten właśnie tekst. Mój komputer nie polubił nowo zakupionej klawiatury mechanicznej - niektóre znaki gubiły się podczas pisania, inne powtarzały kilkukrotnie po jednokrotnym naciśnięciu klawisza. Nie poma-

gała zmiana ustawień systemu, za to problem ustąpił w momencie, kiedy wyłączyłem podświetlenie klawiszy.

Pokazało to wyraźnie, że problemem jest niewystarczające zasilanie klawiatury. Sytuacji nie poprawiła jednak sama wymiana zasilacza do Raspberry Pi na mocniejszy. Sprawność prądowa portów usb jest w Raspberry Pi niestety na tyle niska, że konieczne okazało się podłączenie klawiatury przez aktywny hub usb. Wydajność innych komponentów systemu, takich jak interfejs kart SD czy wbudowana karta sieciowa, również nie powala.

Dlatego dla bardziej profesjonalnych zastosowań zdecydowanie bardziej nadają się komputery ARMv6 czy Titanium, wyposażone w mocne dyski, wydajne interfejsy usb i szybkie, gigabitowe karty sieciowe. Cena tych sprzętów to już jednak wydatek rzędu kilku tysięcy złotych.

Po drugie nie wszystkie wersje Raspberry Pi są równie dobrze wspierane. Wynika to z faktu, że firmie ARM nie zależy aż tak bardzo na zachowaniu wstecznej kompatybilności jak na przykład Intelowi. W związku z tym kolejne generacje procesorów nie są ze sobą w stu procentach zgodne.

Na przykład w ostatniej, ósmej wersji architektury, zrezygnowano z obsługi rozkazu "SWP", pozwalającego na zamianę miejscami zawartości wybranych rejestrów. Na skutek tego wyposażone w ARMv8 Raspberry Pi 3, pomimo tego, że jest obecnie najszybsze, to jednocześnie nie pozwoli nam na uruchomienie niektórych programów (na przykład Milky Tracker). Przeportowywanie software'u na nową architekturę idzie niestety bardzo powoli, ponieważ do sporej ilości programów autorzy nie udostępniają kodu źródłowego.

W związku z tym użytkownicy skazani są na czekanie, aż znajdą oni czas i chęci aby wypuścić ich poprawioną wersję. Sytuacji nie poprawia też fakt, że spora część programów napisana jest w assemblerze, który na platformie RISC OS wciąż jest bardzo popularnym językiem programowania. Dostosowanie ich do nowych wersji procesora wiąże się często nie tylko ze zmianą kilku instrukcji, ale z przepisaniem całych fragmentów kodu.

W przypadku programów napisanych w językach wysokiego poziomu sytuacja jest zazwyczaj lepsza - oczywiście pod warunkiem, że sam kompilator został wcześniej zreporowany. Dlatego paradoksalnie starsze wersje Raspberry Pi są aktualnie lepszym wyborem dla osób chcących rozpocząć swoją przygodę z RISC OS. System ten zresztą jest na tyle szybki i ma tak niewielkie wymagania sprzętowe, że w codziennej pracy trudno nam będzie zauważyć różnicę.

Dotychczas korzystałem z RISC OS na Raspberry Pi B+ (ARMv6), Raspberry Pi 2 v1.1 (ARMv7) i Raspberry Pi 3 (ARMv8) i z całej trójki zdecydowanie polecam dwa pierwsze. Raspberry Pi 3 nie dość, że ma problemy z kompatybilnością, to na dodatek strasznie się grzeje i konieczne jest używanie na nim programu !CPUClock, który pilnuje, żeby procesor się nie przegrzał i nie uległ uszkodzeniu. Warto też pamiętać, że nowsza wersja Raspberry Pi 2, oznaczona symbolem v1.2, posiada ten sam procesor co Raspberry Pi 3 i powyższe uwagi odnoszą się również do niej.

Jak już wspomniałem, dla systemu RISC OS istnieje całkiem pokaźna baza dobrej jakości oprogramowania. Nie jest ona co prawda tak imponu-

jąca jak choćby dla Amigi, ale do dyspozycji dostajemy całkiem przyzwoity system, który może stanowić doskonałą alternatywę dla Windowsa, Linuxa czy Mac OS X. Zresztą analogie do Amigi nasuwają się same. Podobnie jak w przypadku Amigi NG, system nadaje się bardzo dobrze do codziennej pracy, ale ma też podobne bolączki, jak na przykład brak nowoczesnej przeglądarki internetowej.

Podstawową przeglądarką jest, znany również z Amigi, Netsurf, którego podstawową wadą jest brak sensownej obsługi Javascriptu. Kolejnym poważnym brakiem RISC OS z punktu widzenia polskiego użytkownika jest niestety brak obsługi naszego ojczystego języka w samym systemie. Na szczęście polskie strony internetowe z poziomu Netsurfa wyświetlają się poprawnie.

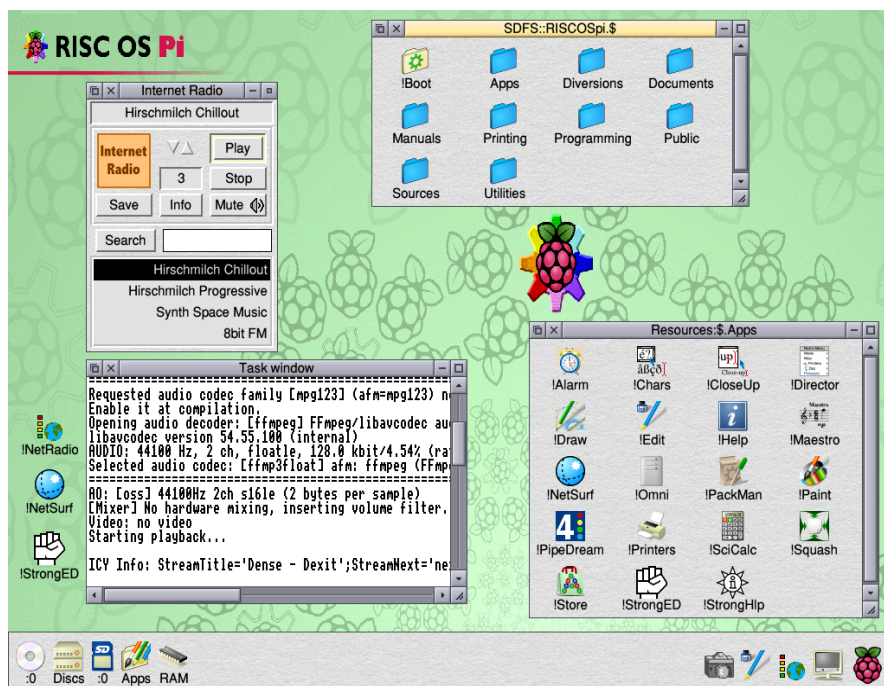
Do dyspozycji mamy też szeroki wachlarz języków programowania, poczynwszy od wbudowanego w system nowoczesnego dialektu Basica i zintegrowanego z nim świetnego assemblera, poprzez kompilator GCC w

wersji 4.7, aż po języki skryptowe, takie jak Python, Perl czy Lua. Możemy też spróbować swoich sił w języku Charm, który jest dostępny wyłącznie w systemie RISC OS.

Jeżeli znudzi nam się praca i będziemy chcieli się trochę rozzerwać, to możemy odprężyć się nieco przy Doomie albo Quake'u, a jeśli lubimy klasyki to dzięki pakietowi ADFFS, będącemu odpowiednikiem amigowego WHDLoad, zagramy w znane także na innych platformach tytuły, takie jak na przykład Alone in the Dark, Cannon Fodder, Gods, Speedball, Lotus Turbo Challenge czy Lemmings.

Możemy też spróbować swoich sił w grach, które nigdy nie pojawiły się na Amidze czy Atari, a były popularne wśród właścicieli Archimedesów, jak na przykład Starfighter 3000. Warto również wspomnieć, że na RISC OS wciąż powstają i są wydawane także nowe gry. Można je zakupić między innymi za pośrednictwem !Store.

Zachęcam każdego, komu znudziło się "pecetowanie", do spróbowania



swoich sił ze wspaniałą spuścizną komputerów Archimedes, jaką jest RISC OS. Nie znajdziemy tu oczywiście wodotrysków obecnych w tzw. nowoczesnych systemach operacyjnych, ale w zamian otrzymamy coś - moim zdaniem - o wiele bardziej wartościowego.

Otrzymamy system zaprojektowany przez inżynierów z pasją, a nie przez wielkie korporacje, dla których liczy się wyłącznie zysk. System przyjazny użytkownikowi, który nie zabiera całej dostępnej pamięci dla siebie, ale zajmuje zaledwie kilkanaście megabajtów, a całą resztę pozostawia do naszej dyspozycji. System, który nie próbuje nas ograniczać ani kontrolować "dla naszego dobra", lecz oddaje nam pełną władzę nad sprzętem. System, dzięki któremu obcowanie z komputerem staje się znowu wspaniałą przygodą.

Krzysztof Kliś

Strony warte odwiedzenia:

<https://www.riscosopen.org/> - strona RISC OS Open Limited, stąd pobierzemy najnowszą wersję RISC OS 5 dla Raspberry Pi, tu również znajdziemy największe forum użytkowników tego systemu.

<http://www.riscos.com/> - strona domowa systemów RISC OS 4 i 6, a jednocześnie niezwykle bogate źródło dokumentacji (polecam szczególnie e-booki dostępne w sekcji Support).

<http://www.iconbar.com/> - działający od ponad 18 lat portal poświęcony wszystkim wersjom RISC OS, najlepsze źródło wiadomości i ciekawostek ze świata komputerów z procesorami ARM.

<https://www.riscosblog.co.uk/> - zbiór bardzo ciekawych artykułów i poradników, skierowanych głównie do użytkowników nowszego sprzętu, w tym Raspberry Pi.

<http://www.riscos.org/links/> - bogata baza oprogramowania, zarówno dla starszych jak i nowszych wersji RISC OS.

<http://sendiri.co.uk/aemulor/> - moduł rezydentny dla RISC OS 5, pozwalający na uruchamianie w nim oprogramowania przeznaczonego dla starszych wersji systemu.



Archiwalne numery
teraz w niższej cenie
z darmową
edycją eBook

Kup prenumeratę
ze zniżką
do 30%

Raspberry Pi

Pierwsze kroki

Dla osoby, która wcześniej nie miała do czynienia z RISC OS, przygodę z tym systemem najłatwiej rozpocząć od Raspberry Pi. Jest to sprzęt niedrogi i szeroko dostępny, a dzięki złączu HDMI możemy go podłączyć do telewizora, nie jest więc potrzebny dedykowany monitor. Na początek wystarczy nam dowolny model tego mikrokomputera, RISC OS obsługuje bowiem wszystkie dotychczasowe wersje: począwszy od A i B, poprzez B+, 2 i 3, aż po Raspberry Pi Zero. Oczywiście najlepiej wybrać wersję z portem Ethernet, żeby nie odbierać sobie możliwości podłączenia do Internetu. Jeżeli jednak dysponujemy tylko modelem A lub Zero, poszukajmy adaptera USB -> Ethernet z układem ASIX AX-88772, na przykład Maplin A31GY, jest on bowiem wspierany przez system RISC OS.

W przypadku kiedy dopiero rozważamy zakup Raspberry Pi, polecam dwa rozwiązania. Pierwsze, to zakup modelu B+, który w zupełności wystarczy do rozpoczęcia przygody z RISC OS. Charakteryzuje się on przy tym niską ceną i bardzo wysokim stopniem kompatybilności z istniejącym oprogramowaniem.

Jest to opcja moim zdaniem najbardziej "bezpieczna" - nie tracimy wiele, jeśli system nie przypadnie nam do gustu. Drugą opcją jest zakup Raspberry Pi 2 rev 1.1, wyposażonego w procesor Cortex-A7 900MHz. Jest to opcja droższa, ale ma tę zaletę, że nadaje się również do pracy z systemem Linux, przede wszystkim ze względu na dwa razy większą pamięć RAM (1GB) i mocniejszy procesor.

Nie polecam natomiast zakupu pod system RISC OS zarówno nowszego modelu Raspberry Pi 2 (rev 1.2), jak i Raspberry Pi 3. Oba modele wykorzystują procesor Cortex-A53, w którym usunięto niektóre rozkazy obecne w starszych generacjach procesorów ARM.

Oznacza to, że nie będzie na nich działać spora część starszych programów, głównie tych napisanych w assemblerze czy językach kompilowanych do kodu maszynowego.

Również adaptacja nowszego oprogramowania nie przebiega zbyt sprawnie, przede wszystkim z powodu niewielkiego rozmiaru środowiska aktywnych programistów - co jest zresztą problemem nie tylko w świecie RISC OS. Drugim problemem, nie związanym jednak bezpośrednio z RISC OS, jest temperatura.

Procesor Cortex-A53 jest taktowany zegarem 1,2GHz i lubi się mocno nagrzewać, co powoduje, że jeśli nie mamy aktywnego chłodzenia, to po kilku minutach pracy jego prędkość jest automatycznie obniżana do 600MHz. Prowadzi to do paradoksalnej sytuacji, w której otrzymujemy komputer wolniejszy niż niektóre starsze modele. W przypadku Raspberry Pi 3 nie skorzystamy też z wbudowanego wifi, gdyż RISC OS nie potrafi go obsługiwać.

To tyle jeśli chodzi o wybór sprzętu. Przejdźmy teraz do instalacji samego systemu. Do tego celu będziemy potrzebowali karty SD (lub microSD - w zależności od posiadanej przez nas wersji Raspberry Pi) o pojemności 2GB lub większej. Niestety RISC OS nie wspiera automatycznego rozszerzania partycji, tak jak ma to miejsce w przypadku np. Raspbiana. Dlatego pozostała po instalacji systemu wolna przestrzeń

(powyżej 2GB) pozostanie niewykorzystana. Jednak zarówno sam RISC OS, jak i przeznaczone dla niego oprogramowanie zajmują na tyle mało miejsca, że nie stanowi to większego problemu.

Jeśli natomiast będziemy potrzebowali więcej miejsca na pliki z danymi, możemy bez problemu podłączyć zewnętrzny dysk twardy lub pamięć flash, sformatowaną w systemie FAT32. Pamiętajmy tylko, żeby nie kopiować na taki nośnik aplikacji, gdyż FAT nie wspiera używanego przez RISC OS systemu atrybutów. Oznacza to, że program przeniesiony na napęd z innym systemem plików niż ADFS (Acorn Disc Filing System) nie zadziała.

RISC OS w wersji 5 dla Raspberry Pi możemy pobrać ze strony:

<https://www.riscosopen.org/content/downloads/raspberry-pi>

w postaci obrazu do nagrania na kartę SD. Z kolei na stronie:

<https://www.riscosopen.org/content/sales/risc-os-pi>

możemy zakupić kartę z już nagrany systemem w najnowszej wersji. Dzięki temu nie tylko oszczędzimy sobie pracy, ale przy okazji wesprzemy rozwój RISC OS. Jeżeli zdecydujemy się na samodzielne nagranie obrazu, pobieramy plik "RC15 SD card image", a następnie rozpakowujemy go na dysku swojego komputera.

W wyniku tej operacji uzyskamy 2GB plik z rozszerzeniem "img", który musimy teraz nagrać na kartę. Do tego celu polecam wykorzystanie bezpłatnego programu Etcher. Znajdziemy go na stronie <https://etcher.io> w wersjach dla systemów Linux, Mac OS X i Windows.

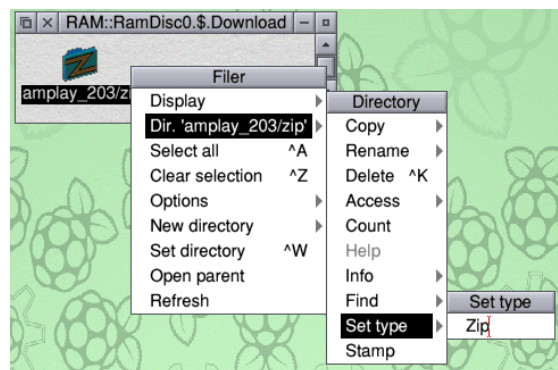
Po uruchomieniu programu wskazujemy plik z obrazem systemu, następnie wybieramy kartę, na którą chcemy go nagrać i naciskamy przycisk "Flash!". Po zakończeniu nagrywania Etcher automatycznie zweryfikuje poprawność zapisanych danych.

Możemy teraz umieścić kartę w Raspberry Pi. Może być ono podłączone do telewizora za pomocą przewodu HDMI lub do monitora SVGA za pomocą przejściówki HDMI->VGA, przy czym ważne aby nie była to zwykła przelotka, ale adapter zawierający układ konwertujący sygnał cyfrowy na analogowy. Adapter taki możemy zakupić w Internecie w cenie ok. 50zł.

Dodatkowo potrzebna nam będzie klawiatura oraz mysz, najlepiej z trzema przyciskami. Jeżeli mamy myszkę tylko z dwoma przyciskami i rolką sprawdzimy, czy rolkę można naciskać i w ten sposób nią klikać. Jeśli tak - myszka taka nada się doskonale.

Warto pamiętać, że w przypadku Raspberry Pi najlepiej sprawdzają się niedrogie, popularne peryferia. Im bardziej "wypasiony" sprzęt, tym większych możemy spodziewać się problemów, szczególnie z zasilaniem.

Jeżeli system zachowuje się niestabilnie - zainwestujmy w mocniejszy zasilacz. Jeżeli urządzenia zewnętrzne "wariują" - klawiatura nie rejestruje wszystkich naciśnień klawiszy, a zewnętrzny napęd CD zaczyna pokazywać błędy odczytu w momencie, kiedy poruszamy intensywnie myszką - podłączmy je za pomocą aktywnego huba, który uzupełni niedostatek mocy na portach usb naszego mikrokomputera.



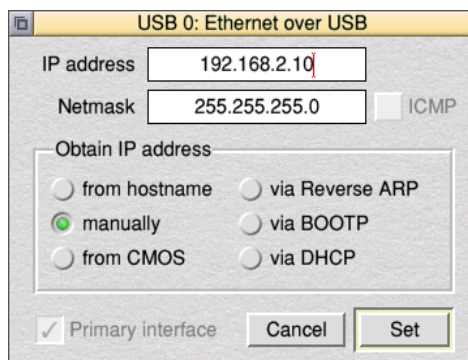
Ustawienie typu pliku wygląda może nietypowo, ale robi się to wygodnie.

Pierwsze uruchomienie RISC OS zajmuje około 30 sekund. W tym czasie następuje wstępna konfiguracja systemu i kolejne uruchomienia przebiegają już zdecydowanie szybciej. Podczas startu następuje również próba automatycznej konfiguracji sieci za pomocą protokołu DHCP.

Jeżeli nasze Raspberry Pi nie jest podpięte do żadnego routera, to wówczas start systemu wydłuży się o kolejne kilkanaście sekund, podczas których RISC OS będzie oczekiwał na uzyskanie adresu IP. Możemy skrócić ten czas, ustawiając w konfiguracji systemu statyczną adresację. Wrócimy do tego za chwilę.

Po uruchomieniu systemu zobaczymy pulpit, który w nomenklaturze RISC OS nazywa się "Pinboard", a także pasek systemowy, znajdujący się przy dolnej krawędzi ekranu, zwany "Icon Bar". Jeżeli posiadamy myszkę wyposażoną jedynie w dwa przyciski, musimy najpierw uruchomić program, który umożliwi nam emulację brakującego przycisku.

W tym wypadku kliknijmy dwukrotnie lewym przyciskiem myszki (nazywanym Select) na symbol karty SD, znajdujący się w lewej części Icon Bara. Otworzy nam się okno zawierające



Konfiguracja adresów IP nie odbiega od standardów znanych z innych systemów.

rające kilka folderów. Odszukajmy folder Utilities i wejdźmy do niego, ponownie klikając dwukrotnie przycisk Select. W środku znajdziemy program o nazwie !WinMenu. U uruchomimy go, klikając na jego ikonę dwukrotnie Selectem. Od tego momentu będziemy mogli używać klawisza Windows (lub jego odpowiednika) na klawiaturze do emulacji środkowego przycisku myszki, zwanego Menu.

Uzupełniając wprowadzone nazewnictwo - prawy przycisk myszki nosi nazwę Adjust. Tak więc korzystając z myszy w systemie RISC OS do dyspozycji mamy trzy kolejne przyciski: Select, Menu i Adjust - i takim właśnie nazewnictwem będę się dalej posługiwał.

Wróćmy jednak do konfiguracji sieci. Jeżeli korzystamy z DHCP to oczywiście nie musimy nic zmieniać, niemniej moim zdaniem warto wiedzieć, jak to zrobić. Kliknijmy więc przyciskiem Menu myszki na symbolu czerwonej maliny, znajdującym się po prawej stronie paska Icon Bar i wybierzmy opcję Configure, klikając na niej Selectem.

Otworzy nam się okno, w którym znajdziemy najważniejsze opcje konfiguracyjne RISC OS. Większość z

nich jest podobna do ustawień w innych systemach, różni się jednak nieco nazewnictwem.

Odszukajmy opcję Network i wybierzmy ją Selectem - tym razem wystarczy jedno kliknięcie, ponieważ poruszamy się wewnątrz aplikacji, a nie okna z folderami (choć wygląda podobnie). Konfigurację połączenia internetowego znajdziemy oczywiście w sekcji Internet. Aby ustawić adres i maskę podsieci klikamy na ikonę Interfaces, następnie na przycisk Configure, wybieramy opcję Manually i wypełniamy pola IP Address oraz Netmask. Zmiany zatwierdzamy przyciskiem Set. Następnie przechodzimy do ikony Routing i w polu Gateway wpisujemy adres IP naszego routera.

Na koniec w oknie znajdującym się pod ikoną Hostnames zaznaczamy opcję "Try name server too" i w pola "Primary name server" oraz "Secondary name server" wpisujemy adresy IP serwerów DNS naszego operatora. Jeżeli ich nie znamy możemy tam wpisać adresy serwerów DNS Google'a: 8.8.8.8 i 8.8.4.4.

Po zamknięciu okna konfiguracji połączenia sieciowego RISC OS zapyta nas, czy chcemy zrestartować system. Odpowiadamy twierdząco. Jeżeli będziemy chcieli przywrócić dynamiczne pobieranie adresów IP, należy usunąć wpisy dotyczące DNS i Gatewaya, a następnie w oknie Interfaces wybrać opcję DHCP w miejsce Manually i ponownie zrestartować komputer.

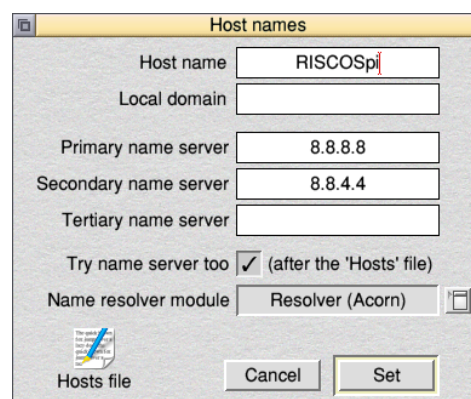
Kolejnym ważnym elementem konfiguracji są napędy. Po wybraniu ikony Discs w panelu konfiguracyjnym RISC OS możemy określić ile dysków danego typu

chcemy obsługiwać. Warto przy tym nadmienić, że chodzi tu zarówno o napędy fizyczne, jak i wirtualne. Dlatego jeżeli planujemy grać w starsze gry, wykorzystując obrazy dyskielek, powinniśmy przy liczbie napędów typu Floppy wybrać minimum jeden, pomimo tego, że przecież Raspberry Pi nie posiada stacji dyskielek.

Podobnie jeśli chcemy podłączyć zewnętrzny CD-ROM lub zamontować obraz ISO (służy do tego darmowy program !CDFaker) to również musimy zarezerwować co najmniej jeden napęd.

W systemie RISC OS możemy również włączyć ramdysk i ustalić jego wielkość w megabajtach. Co prawda rozmiar ramdysku musi zostać określony z góry i nie może się zmieniać dynamicznie, jednak jego włączenie i wyłączenie oraz zmiana wielkości nie wymaga restartu systemu. RISC OS wymaga jedynie, aby podczas tych operacji ramdysk był pusty.

Z opcji konfiguracyjnych warto wymienić również ustawienia klawiatury. Domyślnym układem klawiszy w RISC OS jest UK, czyli brytyjski. Jeżeli naciskając przycisk @ na ekranie pojawia cudzysłów, należy zmienić go na USA. Możemy też skonfigurować szybkość automatycznego powtarzania klawiszy. Niestety na liście dostępnych układów



klawiatury nie znajdziemy polskiego. Bardzo słabe wsparcie dla języków narodowych jest z pewnością jedną z większych bolączek RISC OS.

Po wstępnej konfiguracji systemu przyjrzymy się jak działa GUI w RISC OS. Pierwszą rzeczą, która rzuca się w oczy, jest belka u dołu ekranu, czyli Icon Bar. Znajdziemy na nim nie tylko listę napędów czy główne menu systemowe pod postacią symbolu Raspberry Pi, ale także ikony aktualnie uruchomionych aplikacji.

Aplikacje w systemie RISC OS uruchamiane są w tle i dopiero kliknięcie na ich ikonę ulokowaną na Icon Barze powoduje ich aktywację. Spróbujmy na przykład uruchomić przeglądarkę !Net-Surf. Po dwukrotnym kliknięciu przyciskiem Select na jej ikonie na Pinboardzie (czyli pulpicie) zauważymy, że ta sama ikona pojawiła się na pasku Icon Bar. Oznacza to, że program został uruchomiony i jest gotowy do pracy.

Możemy teraz kliknąć w ikonę !Net-Surf na Icon Barze którymś z przycisków myszki i obserwować efekt swojego działania. Szybko zauważymy, że w zależności od tego, którego przycisku użyjemy, wywołamy różne akcje. Select spowoduje otwarcie nowego okna, Menu da nam dostęp do ustawień przeglądarki, a Adjust otworzy listę ulubionych adresów.

Podobnie podczas przeglądania stron, kliknięcie na odnośnik przyciskiem Select spowoduje przejście pod zadany adres, natomiast Adjust otworzy go w nowym oknie.

Okna w RISC OS zachowują się podobnie jak w innych systemach operacyjnych. Możemy je przenosić, zmieniać ich rozmiar, zamykać, itp. Różnicą, która rzuca się w oczy jest to, że okna są minimalizowane do Pinboardu, a nie Icon Bara, jak można by się spodziewać. Pinboard w RISC OS służy jednak nie tylko do minimalizowania do niego okien. Możemy na nim umieszczać skróty do aplikacji przeciągając na niego ich ikony.

Musimy jednak pamiętać o zapisaniu ustawień, inaczej ikony znikną po restarcie systemu. W tym celu klikamy przyciskiem Menu w dowolnym pustym miejscu Pinboardu, wybieramy opcję Save i klikamy OK.



Okno konfiguracji systemu jest przejrzyste i może przypominać preferencje znane z innych platform.

W lewym górnym rogu każdego okna, oprócz przycisku służącego do jego zamknięcia, mamy również przycisk znany m.in. z Amigowego Workbench, który umożliwia schowanie okna w tle. Takie okno pozostaje jednak cały czas aktywne, dopóki nie klikniemy myszką na inne. Schowane okno możemy w każdej chwili przywrócić, klikając na ten sam przycisk Adjustem zamiast Selectem.

Na tym jednak możliwości ukrywania okien się nie kończą. W ramach

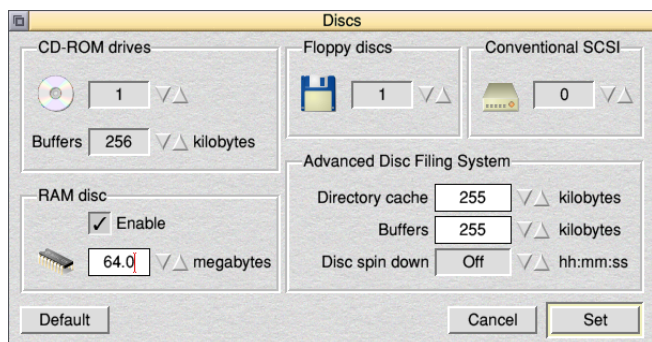
ćwiczenia proponuję otworzyć kilka okien, ustawić jedno nad drugim i obserwować jak zachowują się one, kiedy odsyłamy je w tło przyciskiem Select, tym razem jednak trzymając jednocześnie wciśnięty klawisz Shift.

Komplementarne funkcje przycisków Select i Adjust możemy też wykorzystać poruszając się pomiędzy folderami. Dwukrotne kliknięcie Selectem na folderze powoduje jego otwarcie w nowym oknie, co może w krótkim czasie spowodować bałagan na ekranie. Dlatego możemy otwierać foldery Adjustem, co spowoduje wyświetlenie zawartości nowo otwartego katalogu w tym samym oknie. Aby cofnąć się do poprzedniego folderu wystarczy do zamknięcia okna zamiast przycisku Select wykorzystać Adjust.

Na koniec jeszcze krótka informacja o tym, jak w RISC OS instalować oprogramowanie. W systemie tym, podobnie jak w MacOS, aplikacja jest tak naprawdę katalogiem o określonej strukturze, zawierającym pliki wykonywalne, skrypty rozruchowe, ikony, itp. Od zwykłych katalogów różni się nazwą, rozpoczynającą się zawsze od znaku "!", a także często własną ikoną.

Poza nielicznymi wyjątkami, program może być umieszczony w dowolnym miejscu w systemie plików, w tym również w ramdysku. W starszych komputerach, wyposażonych jedynie w stację dyskietek, kopiowanie aplikacji do ramdysku było popularnym sposobem na przyspieszenie ich działania.

Programy zazwyczaj pobieramy z Internetu w postaci plików zip. RISC OS 5 posiada wbudowane oprogramowanie !SparkFS, które pozwala nam zaglądać do wnętrza skompresowanych plików. Tak więc po pobraniu archiwum i zapisaniu go na dysku



Okno konfiguracji napędów jest bardziej rozbudowane.

wystarczy kliknąć w nie dwukrotnie przyciskiem Select, a następnie przeciągnąć ikonę programu do wybranego folderu.

Jeśli archiwum z programem znajduje się na nośniku wymiennym, na przykład pendrivie, może ono nie zostać rozpoznane poprawnie - pamiętajmy, że RISC OS rozpoznaje typy plików po ich atrybutach, a nie rozszerzeniach. W takiej sytuacji archiwum należy skopiować na systemową kartę SD lub do ramdysku, a następnie kliknąć na nim przyciskiem Menu, wybrać opcję File -> Set Type i zmienić wpisany tam ciąg znaków na "zip".

Jeżeli program się nie uruchamia może to oznaczać, że brakuje mu jakichś zależności. W tej sytuacji przydatne może być przeczytanie pliku pomocy. Zgodnie z wytycznymi systemowymi firmy Acorn, każda aplikacja musi posiadać plik pomocy, do którego dostęp uzyskujemy klikając na jej ikonie przyciskiem menu i wybierając opcję App -> Help.

Problemy z zależnościami rozwiązuje obecny w najnowszej wersji RISC OS menedżer pakietów, zwany !PackMan. Jego obsługa jest bardzo prosta, więc nie będę jej tu szczegółowo opisywał. !PackMan korzysta z repozytorium pakietów znajdujących się na stronie:

<http://www.riscos.info>

jednak nie znajdziemy w nim wszystkich dostępnych dla RISC OS 5 programów, w szczególności komercyjnych.

Dlatego w systemie obecny jest

również sklep z aplikacjami, zwany !Store. Za jego pośrednictwem możemy zakupić oprogramowanie, płacąc kartą kredytową. Niestety nie jest obsługiwany żaden z systemów płatności online, a dane karty podajemy bezpośrednio w aplikacji.

Również realizacja transakcji nie jest natychmiastowa, a zakupione oprogramowanie jest dostępne do pobrania zazwyczaj dopiero następnego dnia roboczego po zakupie. Nie dotyczy to aplikacji dostępnych w sklepie jako freeware - te można pobrać w każdej chwili, bez konieczności podawania danych karty kredytowej i oczekiwania na finalizację płatności.

Na tym kończymy pierwszą część wprowadzenia do systemu RISC OS. W kolejnej poznamy bliżej system plików i polecenia systemowe, a także dowiemy się jak korzystać z napędów wirtualnych.

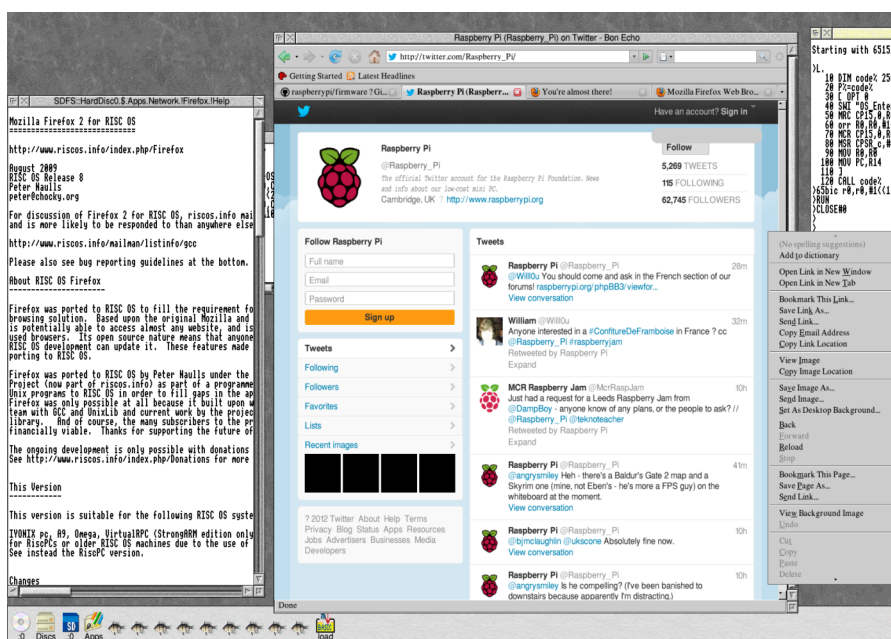
Tymczasem gorąco zachęcam do samodzielnego poznawania i odkrywania możliwości systemu. W poszukiwaniu wiedzy bardzo przydatna będzie również wizyta na forum użytkowników RISC OS, znajdujące się pod adresem:

<https://www.riscos-open.org/forum/>

Krzysztof Kliś

Przydatne informacje o systemie RiscOS można znaleźć także na poniższej stronie:

<https://programowo.net/2012/11/raspberry-pi-i-risc-os,23.html>





IOmega Zip

Korzystanie z nietypowych napędów na Amidze z zasady wymaga dwóch rzeczy: sterownika (.device), który zapewni transmisję danych oraz odpowiedniego systemu plikowego. W przypadku IOmega Zip sytuacja wygląda bardzo podobnie. Nie będzie możliwe używanie napędów w wersjach podłączanych do portu równoległego, możemy jednak z powodzeniem użyć Zipów w wersji SCSI.

Metoda instalacji jest bardzo prosta. Jeśli mamy napęd podłączony do kontrolera SCSI, trzeba tylko utworzyć odpowiedni plik tzw. mountlisty. Jest to zwykły plik tekstowy, który musi być zapisany w odpowiedniej formie.

Dodam, że napęd będzie montowany dopiero po wczytaniu systemu, a więc nie będzie możliwości bootowania, czyli rozruchu komputera z dyskiety Zip. Możemy jednak bez problemu składować dane na dyskietykach o pojemności 100 MB, zarówno sformatowanych w systemie plikowym Amigi (Fast File System), jak i peceta (MS-DOS).

Jako system plikowy wykorzystamy CrossDOS lub Fat95, który można

znaleźć na Aminecie (katalog disk/misc). Teoretycznie można spróbować wykorzystać nawet CrossDOS dołączany do Workbench, ale nie będzie on w stanie przekroczyć ograniczeń związanych z maksymalną pojemnością zwykłych dyskietek oraz nazewnictwem plików w standardzie 8+3.

Tak więc mija się to z celem, chyba że do celów testowych. Nowsze wersje CrossDOS-a działają bardzo dobrze. Zawartość pliku mountlisty przedstawiam w ramce na następnej stronie. Aby jej użyć należy skopiować plik do poniższego katalogu:

SY\$S:Devs

a następnie użyć polecenia:

mount ZIPC:

lub

mount ZP0:

w oknie Shell. Pakiet CrossDOS wymaga, aby zakończeniem nazwy symbolu urządzenia była litera C lub D, natomiast w przypadku Fast File System nazwa może być dowolna. Proponuję jednak zachować typowo amigowe nazewnictwo. Oczywiście po sformatowaniu dyskiety w FFS, zachowane zostaną atrybuty plików i komentarze.

Na dyskiecie Zip nie należy tworzyć partycji. Należy ustawić jeden obszar w maksymalnym rozmiarze, czyli ok.

95 MB. Teoretycznie można to zrobić i dzięki temu uzyskać nawet kilka oddzielnych urządzeń sformatowanych w różnych systemach plikowych, ale zajmuje to większą ilość pamięci, a operacje na takiej dyskiecie przebiegają dużo wolniej z uwagi na konieczność aktualizacji różnych partycji. Napędy Zip nie są bardzo szybkie i praktycznie możemy spodziewać się transmisji danych na poziomie ok. 1 MB/s.

Ten sam sposób obsługi IOmega Zip może być stosowany dla systemów operacyjnych w wersjach od 2.0 począwszy. Wpis "scsi.device" musisz zamienić na nazwę sterownika obsługującego Twój kontroler, czyli na przykład scsi-apollo.device albo 2060scsi.device. Ostatnie moje zastrzeżenie dotyczy kart Apollo, które są bardzo wybredne zarówno w samym działaniu, jak i obsłudze urządzeń SCSI.

Osobiście nie udało mi się uruchomić Zipa na karcie Apollo w mojej Amidze 1200, natomiast działa on bez problemu na karcie Blizzard w Amidze 2000. Jest to znana sprawa, ale wszystkie moje próby obsługi na karcie Apollo kończyły się zawieszeniem komputera. Jeżeli uda się Wam wyeliminować ten problem, proszę o przesłanie mi informacji o wersji oprogramowania, z której korzystacie.

Marcin Libicki



```
ZIPC:      Device = scsi.device
           Unit = 5
           Flags = 0
           Surfaces = 1
           BlocksPerTrack = 60
           Reserved = 0
           Interleave = 0
           LowCyl = 0; HighCyl = 2890
           FileSystem = L:CrossDOSFileSystem
           Stacksize = 4000
           Priority = 4
           GlobVec = -1
           DosType = 0x4D534800
           Mask = 0x7ffffffe
           MaxTransfer = 0x100000
           Buffers = 5
           BufMemType = 1
           Mount = 1
```

#

```
ZP0:      Device = scsi.device
           Unit = 5
           Flags = 0
           Surfaces = 1
           BlocksPerTrack = 60
           Reserved = 0
           Interleave = 0
           LowCyl = 0; HighCyl = 2890
           FileSystem = L:FastFileSystem
           Stacksize = 4000
           Priority = 4
           GlobVec = -1
           Mask = 0x7ffffffe
           MaxTransfer = 0x100000
           DosType = 0x444F5301
           Buffers = 5
           BufMemType = 1
           Mount = 1
```

#



KA plus
Magazyn użytkowników komputerów Commodore

Z pasją o kulturze retro gier komputerowych:
■ recenzje ■ felietony ■ tutoriale ■ dyskusje
■ demoscena ■ sztuka ■ programowanie
■ wywiady ■ emulacja ■ nowości

Komoda & Amiga plus to pismo poświęcone szerokokopijnej kulturze retro gier komputerowych. Mimo że nasze rozważania poświęcone są głównie publikowanym na sprzęt spod znaku marki Commodore to jednak celem nadrzędnym jest odświeżenie tego niesamowitego klimatu lat 80/90-tych, czyli czasów naszego dzieciństwa wypełnionego pikselami i dźwiękiem chipowym.

www.ka-plus.pl team@ka-plus.pl
Amiga 68k, AmigaOS 4.x, AROS, MorphOS
Plus/4, VIC20, C16, C64, C128



Workbench i 24-bit

Gdy chcemy korzystać z 24-bitowych trybów wyświetlania, musimy zainstalować sterowniki do karty graficznej. Oczywiście najpierw należy rozbudować sprzęt, chyba że korzystamy z emulatora Amigi. Jednak sama instalacja pakietu CyberGraphX lub Picasso 96 to nie wszystko, co należy zrobić.

Po zainstalowaniu sterowników karty uzyskałeś możliwość uruchamiania nowych trybów wyświetlania. Z pewnością szybkość operacji graficznych jest dużo większa, jednak Workbench nie zawiera standardowo komponentów pozwalających na wyświetlanie grafiki w większej ilości kolorów. Wyjątkiem jest tryb „HAM”,

Gdy chcesz korzystać z plików zawierających grafikę 15-, 16- lub 24-bitową, musisz zainstalować odpowiednie „datatype”. Teraz zajmiemy się tylko rozszerzeniem funkcji tak, aby można było wykorzystać wszystkie cechy sprzętu.

Datatype zarządzającym wyświetlanie wszystkich formatów grafiki jest „picture”. Dla karty graficznej należy zainstalować inną wersję, ale musi

uwzględniać ona również zainstalowany sterownik, czyli system „CyberGraphX” lub „Picasso96”. W przypadku tego drugiego odpowiednie pliki są automatycznie kopiowane podczas instalacji pakietu. Zwróć uwagę, że podczas wyboru komponentów jedną z pozycji jest:

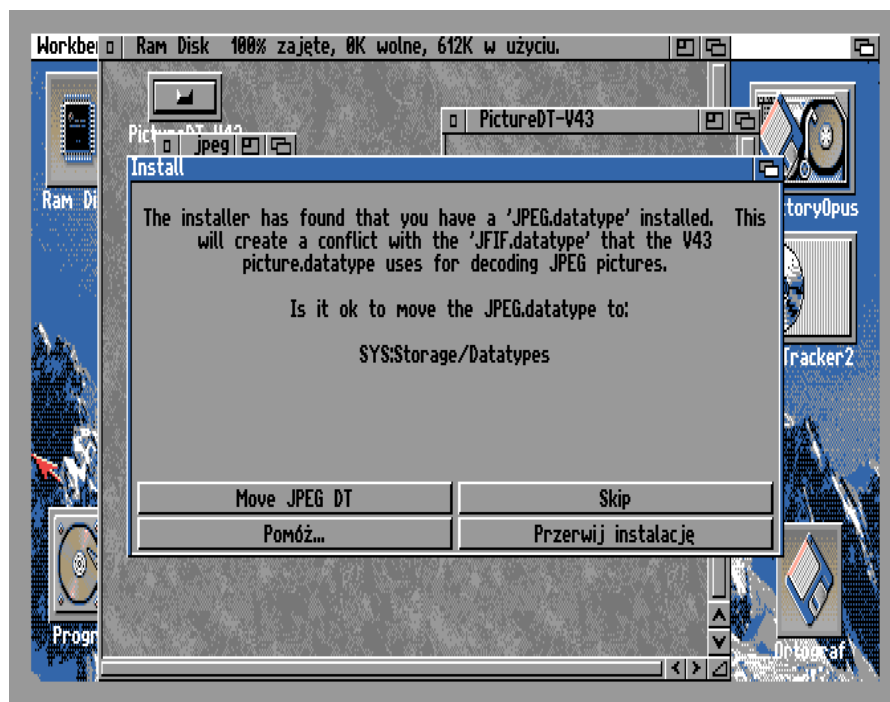
24bit picture.datatype

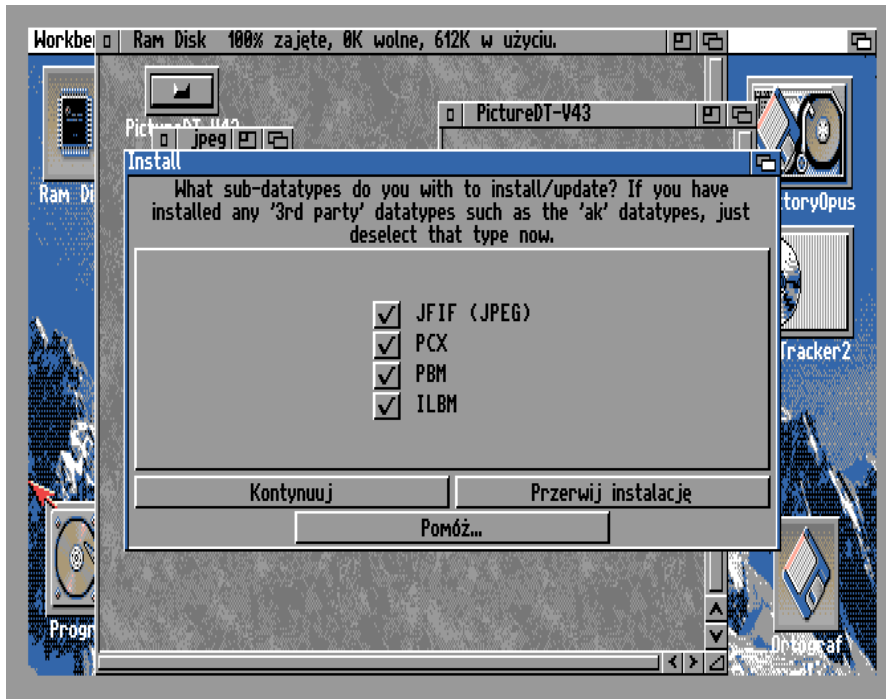
Dlatego nie musisz ręcznie kopiować dodatkowych plików. Jeśli jednak korzystasz z pakietu „CyberGraphX” należy zainstalować „picture.datatype” w wersji oznaczonej numerem

43. Znowu oprzemy się na wersji dostępnej na Aminecie. Przejdź do katalogu „util/dtype” i pobierz plik o nazwie „PictDT43.lha”. Zwróć uwagę, aby nie pomylić nazwy. Następnie rozpakuj archiwum w zwykły sposób, czyli wykonując linię polecenia:

`lha x PictDT43.lha RAM:`

aby umieścić wszystkie pliki w „Ram Dysku”. Odszukaj tam ikonę katalogu „PictureDT-V43” i wykonaj na niej „dwuklik”. Gdy pojawi się nowe okno uruchom program „Install”. Na ekranie powinieneś zobaczyć stan-





Wybór obsługiwanych formatów podczas instalacji.

dardowe okno „instalatora” wraz z krótką informacją na temat pakietu. Wybierz przycisk „Kontynuuj”. Następne dwa kroki wymagają takiego samego działania, jednak zobaczysz w kolejności – tzw. opcje instalacji oraz komunikat na temat zastosowanych patentów związanych z formatem „GIF”.

Dalej program spyta o miejsce, w którym mają zostać zainstalowane nowe pliki. Domyślnie wskazany będzie katalog „Classes” na dysku systemowym i jest to dobry wybór. Dlatego nic nie zmieniaj, tylko wskaż jeszcze raz opcję „Kontynuuj”.

W podobny sposób zostaniesz zapytany o to, gdzie znajdują się biblioteki systemu „CyberGraphX”. „Instalator” znowu samoczynnie ustawi katalog „Libs”, dlatego bez wykonywania zmian potwierdź wybór za pomocą takiego samego przycisku „Kontynuuj”.

Dotrzesz do możliwości wykonania kopii bezpieczeństwa aktualnego

stanu plików związanych z datatypami.

Nie musisz uruchamiać tej funkcji, ale uważamy, że zawsze warto posiadać kopie ważnych plików systemowych, dlatego pole „BackUp Current Datatypes” należy pozostawić aktywne. Następnie kolejny raz wskaż przycisk „Kontynuuj”. Teraz program spyta o to, gdzie zapisać kopie plików.

Domyślnie wybrany będzie katalog „Storage” na dysku systemowym, który sprawdzi się znakomicie. Nie rób żadnych modyfikacji, tylko skorzystaj z przycisku „Kontynuuj”, aby przejść dalej. Kolejna informacja dotyczy dodatkowych formatów grafiki, które mogą zostać zainstalowane. Zobaczysz kilka pól oznaczonych symbolami:

Taki stan może pozostać, bowiem uzyskasz możliwość odczytywania plików w wymienionych formatach – także wtedy, gdy będzie to grafika zapisana w pełnej 24-bitowej palecie

barw. Wskaż znowu przycisk „Kontynuuj”. Jeśli jednak posiadasz zainstalowany wcześniej datatyp dotyczący jednego z tych formatów zobaczysz nowy komunikat.

W naszym przypadku pokrywają się pliki rozpoznające format „JPEG”. Wcześniej nie korzystaliśmy z datatypów przeznaczonych dla kart graficznych, dlatego należy użyć przycisku „Move JPEG DT”. Spowoduje to sporządzenie kopii aktualnych plików w systemowym katalogu „Storage” i zainstalowanie w systemie nowego datotypu. Jeśli jesteś pewien, że poprzednie pliki obsługują 24-bitowe tryby wyświetlania, możesz skorzystać z pola „Skip”, dzięki czemu pominiesz tę część instalacji.

Program skopiuje niezbędne pliki na dysk, po czym wyświetli kolejne pytanie. Tym razem będzie dotyczyć chęci instalacji programu „preferencji” nowych datatypów.

Nie jest to konieczne, ale za pomocą programu o nazwie „PCDT” można zmieniać sposób tzw. „roztrząsania” (ang. dithering). Plik nie zajmie wiele miejsca na dysku, dlatego bez wykonywania zmian wybierz pole „Kontynuuj”.

Zostaniesz zapytany o miejsce instalacji programu, przy czym znowu domyślnym wyborem będzie systemowy katalog „Prefs”. Wskaż od razu przycisk „Kontynuuj”. „Instalator” skopiuje teraz pozostałe ważne pliki na dysk, a dalej w oknie wypisany zostanie następujący komunikat:

Instalacja zakończona!

Na dole dostępny będzie tylko jeden przycisk – znajoma opcja „Kontynuuj”. Użyj jej, a okno powinno zniknąć. Po zresetowaniu Amigi i ponownym wczytaniu Workbenchu nowe datatypy będą obsługiwać



wyświetlanie grafiki z dużą ilością kolorów. Pamiętaj, że w każdej chwili możesz zainstalować kolejne pliki, lecz muszą być przeznaczone dla pakietu „CyberGraphX” lub „Picasso96”. Polecamy szczególnie pakiet o nazwie „WarpDT”, który można znaleźć na stronie:

www.warpgdt.co.uk

Jest on rozwijany od wielu lat, a ostatnie wersje datowane są na październik 2014 roku. Dobrym rozwiązaniem jest także skorzystanie z datatypów o nazwach rozpoczynających się od „ak”, które można znaleźć na Aminecie w katalogu „util/dtype”, na przykład „akGIF” lub „akJFIF”. W przeciwieństwie do poprzednich są darmowe.

File	Version	Release Date	File size
WarpBMP.datatype	45.7	15 March 2011	70K
WarpJPEG.datatype	45.13	20 April 2016	323K
WarpPCX.datatype	45.5	25 February 2011	64K
WarpPNG.datatype	45.21	24 March 2017	181K
WarpPSD.datatype	45.5	11 March 2011	67K
WarpTIFF.datatype	45.9	4 May 2016	517K
WarpDTPrefs	45.9	4 May 2016	153K

Strona internetowa WarpDT jest aktywna także dzisiaj.



MysticView - szybka przeglądarka grafiki obsługująca datatypy.

Oczywiście są to tylko przykłady, bowiem z sieci możesz pobrać przynajmniej kilkaset różnych datatypów. W każdym przypadku przed instalacją przeczytaj dokumentację, aby

zorientować się czy obsługiwane są karty graficzne, a jeśli tak – czy jest to odpowiednia wersja dla Twojego sterownika. Jeśli tego nie zrobisz może okazać się, że po zainstalowa-

niu pakietu stracisz możliwość rozpoznawania grafiki albo będzie ona pokazywana w nieprawidłowy sposób.

Na Aminecie możesz także spotkać wiele programów służących do przeglądania plików graficznych. Wiele z nich posiada dopiski oznaczające obsługę pakietu „CyberGraphX” lub „Picasso96”. Można z nich korzystać, bowiem posiadają zwykle o wiele większe możliwości niż standardowe programy dostępne w systemie.

Pamiętaj jednak, że najłatwiejszym sposobem sprawdzenia czy obsługiwane są konkretne pliki jest wywołanie fabrycznego programu „Multiview”. Nie jest zbyt szybki, posiada ograniczoną ilość funkcji, ale zawsze odpowie na pytanie czy Twój Workbench działa poprawnie i czy zainstalowałeś prawidłowy datatyp oraz powiązane z nim pliki.

Adam Zalepa



Ukryte funkcje karty graficznej

Obsługa karty graficznej to także odpowiednie ustawienie tak zwanych zmiennych środowiskowych. Podczas instalacji sterownika ustawialiśmy je, bowiem mają bezpośredni wpływ na działanie karty graficznej. Raz ustawiona konfiguracja nie oznacza braku możliwości dokonania poprawek. Może się bowiem okazać, że jest to wymagane do działania przynajmniej części oprogramowania.

Aby je później zmienić należy zajrzeć do wpisów umieszczonych w urządzeniu logicznym „ENV:” lub „ENVARC:”. Nie jest to wygodne, ponadto źle wykonana modyfikacja może doprowadzić do awarii systemu. Na szczęście możesz skorzystać z dodatkowych programów, które ułatwiają konfigurację.

Tak jak wcześniej – trzeba użyć programu przeznaczonego dla pakietu, który został zainstalowany. Wraz z „Picasso96” do systemowego katalogu „Prefs” kopiowane są „preferencje” o nazwie „PVS” (jest to skrót od „Picasso96 Variable Setter”). Po uruchomieniu pojawi się mniejsze

okno. Program korzysta z „Magic User Interface” i można swobodnie zmieniać jego wygląd.

Pierwsza zakładka „Environment” zawiera ustawienia różnych funkcji karty graficznej, natomiast za pomocą drugiej - „Tooltypes” - mamy dostęp do parametrów ikony sterownika.

Na pierwszej „stronie” zwróć uwagę na przycisk cykliczny „Keep Amiga Video”. Gdy ustawisz go w pozycji „31kHz”, obraz generowany przez układy Amigi będzie miał parametry trybu „Multiscan”. Dzięki temu możliwe będzie odtwarzanie dźwięku z lepszą jakością, bowiem układ dźwiękowy „Paula” będzie w stanie uzyskać częstotliwość większą niż 30 kHz. Nie możesz jednak wtedy używać trybów telewizyjnych PAL. Dlatego funkcja ta jest odpowiednia jeśli korzystasz tylko z karty graficznej lub dla trybów wyświetlania typu VGA przez złącze monitorowe na płycie głównej.

W tym pierwszym przypadku możesz również włączyć opcję oznaczoną jako „Black Switching”. Spowoduje ona wygaszenie obrazu Amigi, gdy używasz tylko ekranu karty graficznej. Zwykle pokazywane jest wtedy szare tło i jest to niepotrzebne. Jak już wspomnieliśmy, lista trybów

wyświetlania dostępna w programie „ScreenMode” po zainstalowaniu sterowników karty graficznej jest dużo dłuższa. Można ją nieco skrócić wybierając funkcję „Essential” na przycisku o nazwie „Show Modes”. Dzięki temu lista zawierać będzie tylko te pozycje, które zasadniczo różnią się między sobą. Jeśli chcesz przywrócić poprzedni stan, wskaż w tym samym polu funkcję „All”.

Gdy korzystasz z procesora Motorola 68040 lub 68060, system używa układu „MMU” (ang. Memory Management Unit) zarządzającego pamięcią. Niektóre programy mogą jednak działać zbyt szybko lub powodować inne problemy. Dlatego za pomocą pola „Do Not Set MMU” możesz wyłączyć funkcje związane z układem „MMU”. Zwróć uwagę, że podobna opcja znajduje się w innych programach, na przykład popularnym „WHDLoad”. Pozwala to zachować zgodność ze starszym oprogramowaniem.

Gdy używasz programów służących do odtwarzania animacji przydatne może przydać się również pole o nazwie „Planar Only Dbuf”. Po „włączeniu” system będzie korzystał z funkcji tzw. podwójnego buforowania (ang. double buffering) tylko na ekranach układów Amigi, a nie karty



graficznej. Operacje przyspieszane przez układ „blitter” mogą być też wykonywane przez procesor główny. Gdy Twoja Amiga posiada szybki procesor, spowoduje to wzrost prędkości pracy, o ile będziesz używał zwykłych trybów graficznych. Aby tak się stało pozostaw aktywną opcję „Disable Amiga Blitter”.

Za pomocą pola o nazwie „Planes To Fast” możliwa jest natomiast aktywacja funkcji, która powoduje umieszczanie danych graficznych w pamięci typu „Fast” zamiast „Chip”, co także przyspiesza działanie. Ponadto niektóre programy są wrażliwe na ilość pamięci „graficznej”. O tym mechanizmie pisaliśmy już przy okazji instalacji pakietu „Cyber-GraphX”.

Druga zakładka w oknie „PVS” nosi tytuł „Tooltypes”. Ułatwia ona zmianę parametrów ikony sterownika karty graficznej. Pamiętaj, że cały czas odnosi się to do pakietu „Picasso96”. W górnej części umieszczona jest lista „Graphic Boards”, na której zobaczysz nazwę aktywnego sterownika. Dalej zwróć uwagę na funkcje poniżej.

Pierwsza ważna opcja ukryta jest pod przyciskiem cyklicznym „Border Blank”. Napis „System” spowoduje, że ekran będzie zawierał ramkę, podobnie jak w przypadku obrazu generowanego przez układy na płycie głównej komputera. Ramkę możesz wyłączyć przez ustawienie pozycji „Never”. Jeśli Twoja Amiga posiada układy „ECS”, pole o nazwie „Big Sprite” umożliwi zmianę rozmiaru wskaźnika myszki. Będzie on dwukrotnie większy po „włączeniu” funkcji.

Układ „blitter” Amigi pozwala wyświetlać obiekty zwane „spritami” (lub „bobami”). Na ekranach karty graficznej trzeba je emulować i jest to

jedną z cech sterownika. „Sprity” mogą być jednak emulowane na każdym z ekranów, niezależnie od wybranego trybu. Pozwala to przyspieszyć lub zwolnić ich obsługę, w zależności od konfiguracji sprzętowej Amigi. W tym celu należy zmienić stan opcji oznaczonej jako „Soft Sprite”. Pamiętaj, że nie zawsze będzie to działać prawidłowo, szczególnie podczas korzystania ze starszego oprogramowania.

Kolejne pole „No Blitter” pozwala wyłączyć układ „blitter”, lecz tym razem ten, który posiada karta graficzna. Spowoduje to drastyczne zmniejszenie szybkości pracy, dlatego używaj tej opcji tylko, gdy jest to absolutnie konieczne. Możesz również zadziałać w drugą stronę, czyli przyspieszyć operacje na kartach wyposażonych w układ firmy Cirrus Logic, na przykład „Picasso II”. Aby tak się stało „włącz” pole opisane jako „Ignore Mask”. Funkcja ta nie będzie działać prawidłowo na innych kartach, poza tym nie jest w pełni zgodna z systemem operacyjnym

Amigi. Pozwala jednak uzyskać maksymalną szybkość przewijania ekranu na wolniejszym sprzęcie. Z powyższych względów, zanim użyjesz tej opcji sprawdź na jakim układzie oparta jest Twoja karta.

Ciekawą możliwością jest także wykonywanie niektórych funkcji za pomocą procesora głównego, zamiast „blittera” karty graficznej. Może to dotyczyć wyświetlania czcionek oraz sposobu zapisywania informacji o grafice. W tym celu skorzystaj z dwóch opcji: „Fonts Fix” oraz „No Planar-to-Chunky Blits”. Jeżeli będą włączone, system będzie korzystał z procesora Amigi zamiast układu na karcie. Może to prowadzić do przyspieszenia działania. Wszystko jednak zależy od wydajności sprzętu, dlatego najlepiej samodzielnie sprawdzić która konfiguracja będzie pracować szybciej.

Mówiąc o wydajności warto wspomnieć o tym, że program pozwala dokonać zwiększenia częstotliwości tzw. „zegara obrazu” (ang. pixel





clock). Dzięki temu możemy uzyskać wyższą rozdzielczość ekranu wraz z większą częstotliwością odświeżania. Jeżeli obecne tryby wyświetlania nie są wystarczające, spróbuj „włączyć” pole o nazwie „Overclock”. Jest to funkcja obciążająca sprzęt, bowiem będzie on pracował nieco szybciej niż zakładał to producent.

Dlatego na ekranie pojawi się okno zatytułowane „Warning!” zawierające ostrzeżenie o potencjalnej możliwości uszkodzenia karty. Są to przypadki rzadkie, niemniej trzeba o tym pamiętać. W oknie wybierz przycisk „OK”. Opcja ta będzie działać inaczej w zależności od zastosowanej karty. Gdyby po jej aktywacji obraz zaczął być wyświetlany nieprawidłowo, oznacza to, że nie powinienś z niej korzystać.

W takiej sytuacji od razu zmień ustawienia na poprzednie, aby nie ryzykować dalszymi problemami. W podobny sposób możesz przyspieszyć pamięć karty graficznej, lecz należy przestawić wartość widoczną na suwaku „Memory Clock” w dolnej części okna. Jeśli jest nieaktywny – tak jak na naszej ilustracji – oznacza to, że Twój sterownik nie obsługuje tej możliwości.

Gdy chcesz używać dwóch monitorów podłączonych jednocześnie do złącza wideo na płycie głównej i gniazda karty graficznej, przydatnym polem jest „Display Chain”. Musisz je „włączyć”, w przeciwnym razie obraz nie będzie mógł być wyświetlany na raz na obu wyświetlaczach. Możesz też spowodować, aby na liście trybów wyświetlania widoczne były pozycje bliźniacze do sterowników układów Amigi. W tym celu skorzystaj z funkcji opisanej jako „Fake Native Modes”.

Jednocześnie musisz wtedy usunąć wszystkie inne sterowniki – poza tymi z pakietu „Picasso96” - z katalogu „Monitors” umieszczonego w „Devs”

na dysku systemowym. Możesz je oczywiście przenieść do „Storage”, tak jak na pewno robiłeś to wcześniej.

Jeżeli podczas wczytywania Workbench'a i uruchamiania sterownika karty wystąpi błąd, zwykle pojawia się okno informacyjne. Może to zatrzymać rozruch systemu lub zablokować możliwość przełączenia pulpitu na ekran karty graficznej. Aby tego uniknąć użyj innego pola widocznego w oknie „PVS” - oznaczonego jako „Quiet”. Pamiętaj jednak, że wraz z wyłączeniem komunikatów pozbawiasz się możliwości diagnozowania ewentualnych usterek. Oczywiście zawsze możesz przywrócić poprzedni stan i ponownie załadować system, aby odczytać informacje.

Na dole umieszczone są trzy typowe przyciski „Save”, „Use” oraz „Cancel”, które odpowiadają funkcjom zapisania konfiguracji na dysku, zapamiętania do czasu ponownego rozruchu komputera oraz zrezygnowania z wprowadzania zmian. Nie jest to żadna nowość, działają one analogicznie do przycisków w systemowych programach „preferencji”.

Jeśli korzystasz z „CyberGraphX” ustawianie zmiennych środowiskowych wygląda trochę inaczej. Musisz zainstalować oddzielny program, bowiem wraz z pakietem nie jest dostarczane żadne narzędzie podobne do poprzednio omówionego. Proponuję skorzystać z „CyberEnvPrefs” znajdującego się na Aminecie w katalogu „driver/video”. Pobierz stamtąd archiwum o nazwie „CyberPrefs4065.lha” i rozpakuj je do „Ram Dysku” za pomocą standardowej formuły:

**1ha x CyberPrefs40-
65.1ha RAM:**

Odczytaj zawartość katalogu docelowego i wykonaj „dwuklik” na ikonie podpisanej „CyberEnvPrefs”.

W nowym oknie zobaczysz kolejną pozycję o takiej samej nazwie, lecz będzie to już główny plik, który należy przenieść myszką do systemowego katalogu „Prefs”. Nie wymaga żadnych innych czynności instalacyjnych. Następnie najedź na ikonę wskaźnikiem i naciśnij dwukrotnie lewy klawisz myszki. Pojawi się okno jak poniżej:

Program jest łatwy w obsłudze i zawiera tylko najważniejsze funkcje. Za pomocą przycisku cyklicznego w lewym górnym rogu należy ustawić model karty, którą posiadamy. Lista poniżej - o nazwie „Environment” - zawiera pozycje, które umożliwiają zmianę podstawowych parametrów sterownika. Wiele z nich pokrywa się z tymi, które są dostępne w pakiecie „Picasso96”, ale niektóre mają nieco inne znaczenie.

Pierwszą ważną opcją jest „Hires Pointer”. Dzięki niej grafika wskaźnika myszki będzie wyświetlana w wyższej rozdzielczości. Aby ją uaktywnić należy najechać wskaźnikiem na nazwę i nacisnąć lewy klawisz myszki. Zwróć uwagę, że obok pojawi się znak gwiazdki „*”. Gdy wykonasz tę samą czynność jeszcze raz gwiazdka zniknie i opcja zostanie „wyłączona”. Niektóre pozycje mogą posiadać także mały znak „x”, co uwzględniliśmy na powyższej ilustracji. Takich pozycji nie można „włączyć”, bowiem ich stan jest ustalony na stałe lub sterownik ich nie obsługuje.

Za pomocą funkcji opisanej jako „CPU for P2C” możesz spowodować, aby konwersja grafiki typu „Planar To Chunky” była wykonywana przez procesor główny. Więcej na ten temat mówiliśmy w części „Instalacja sterowników”, w punkcie związanym ze zmienną „CPUP2C”. Kilka następnych opcji dotyczy zarządzania pamięcią komputera.



Po aktywacji pozycji „No Screens in CHIP” oraz „Bitmaps to FAST” spowodujesz, że wszystkie informacje na temat używanych ekranów systemowych oraz dane grafiki bitmapowej będą zapisywane w pamięci typu „Fast”, zamiast – domyślnie – w pamięci „graficznej”.

Pozwala to zmniejszyć wykorzystanie pamięci typu „Chip”, a w wielu przypadkach również przyspieszyć wykonywanie operacji. Jeśli Ci na tym zależy „włącz” również pozycję „Save Memory”, bowiem system będzie wtedy oszczędzał pamięć w celu pozostawienia wolnego jak największego obszaru przeznaczonego na obsługę ekranów.

Systemowe komunikaty o awariach (na przykład ramka „Guru Meditation”) nie są automatycznie wyświetlane na ekranie karty graficznej. Można to jednak łatwo zmienić, wystarczy skorzystać z opcji „Redirect Alerts” widocznej na liście nieco wyżej. Kolejne dwie pozycje związane są z używaniem kilku monitorów lub kart graficznych jednocześnie.

Pierwsza z nich - „Disable Mon Switch” - powoduje wyłączenie funkcji samoczynnego przełączania obrazu na ten wyświetlacz, gdzie nastąpiła zmiana. Jest wykorzystywana rzadko, gdyż mało kto ma podłączonych więcej kart w jednej Amidze. Natomiast za pomocą opcji „Keep video DMA” sprawisz, że sygnał pochodzący z gniazda wideo na płycie głównej będzie utrzymywany cały czas, nawet jeśli aktualnie obraz wyświetlany będzie tylko za pomocą karty graficznej.

Jest to odpowiednik zmiennej środowiskowej „KEEPAMIGAVIDEO”. Zwróć uwagę, że po prawej stronie okna „CyberEnvPrefs” dostępne są dwa suwaki związane z konfiguracją monitora.

Dzięki nim możliwe jest ustalenie minimalnej i maksymalnej częstotliwości odświeżania poziomego. Wartości te powinien znaleźć w dokumentacji monitora. Jeśli na dysku masz zapisaną definicję monitora, możesz w zamian użyć przycisku „Select” i w oknie wyboru wskazać plik. Wiele z nich znajdziesz m.in. na Aminecie w katalogu „driver/moni”. Ich stosowanie jest łatwiejsze niż ręczna konfiguracja, ale rezultaty będą takie same.

Poniżej grupy „Monitor” widoczna jest ramka oznaczona jako „CyberGraphX Versions”, która pokazuje wersje kilku bibliotek zainstalowanych na dysku. Wymienione pozycje są związane z obsługą grafiki, a całe pole ma znaczenie wyłącznie informacyjne. Obok umieszczono klasyczne przyciski „Save”, „Use” i „Cancel”.

W przypadku pakietu „Picasso96” należy skorzystać z trybu „Multiscan”, aby możliwe było odtwarzanie dźwięku o wyższej częstotliwości.

To samo możesz uzyskać również za pomocą „CyberGraphX”, ale zamiast ustawiać odpowiednią opcję jako zmienną środowiskową, należy użyć oddzielnego programu „Cyber56kHz”. Nazwa pochodzi od teoretycznej funkcji odtwarzania dźwięku o maksymalnej częstotliwości 56 kHz.

Program jest dostępny na Aminecie w katalogu „driver/audio”. Pobierz archiwum o nazwie „Cyber56kHz.lha” i rozpakuj je, na przykład, do „Ram Dysku”:

lha x Cyber56kHz.lha RAM:

Następnie odczytaj jego zawartość i włącz funkcję wyświetlania wszystkich plików. Wykonaj „dwuklik” na ikonie podpisanej jako „Cyber56kHz”, a w oknie „Wykonaj polecenie” wpisz poniższą linię:

copy Cyber56kHz C:

Na koniec musisz jeszcze przenieść ikonę sterownika „Multiscan” z systemowego katalogu „Storage” do „Devs”. Oczywiście w obu przypadkach plik ten jest umieszczony w dodatkowym katalogu o nazwie „Monitors”.

Jest to konieczne nawet, gdy nie chcesz używać trybu wyświetlania „Multiscan”, bowiem korzysta z niego polecenie „Cyber56kHz”. Aby je uruchomić wystarczy wpisać nazwę w oknie „Wykonaj polecenie” lub oknie „Shell” bez żadnych argumentów, czyli:

cyber56kHz

Potem naciśnij klawisz ENTER. Taki wpis możesz również dodać do pliku rozruchowego „user-startup” w katalogu „S” na dysku systemowym, lecz powinien być poprzedzony poleceniem RUN i dodatkowym symbolem „>NIL:”, czyli razem:

run >NIL: cyber56kHz

Dzięki temu nie będziesz musiał uruchamiać go za każdym razem, a linia ta nie spowoduje blokady wczytywania systemu.

Oba pakiety obsługi kart graficznych mają podobne możliwości, choć uzyskujemy je na kilka sposobów. Pamiętaj, że podstawowe różnice kryją się samych sterownikach kart graficznych, a pozostałe funkcje są dodatkami, których nie trzeba koniecznie używać.

Najważniejsze, aby zdecydować się na jeden pakiet, a potem dostosować cały system do jego obsługi. W przeciwnym razie doprowadzisz do sytuacji, gdzie tylko część programów będzie działać prawidłowo.

Adam Zalepa