

Tak narodził się Bascom

Urodziłem się w 1963 roku w Amsterdamie. Moja fascynacja elektroniką zaczęła się gdy miałem 5 lat. Na temat elektroniki wiedziałem niewiele, ale nie przeszkodziło mi to w zabawach z żaróweczkami, bateriami i przełącznikami. W wieku lat 10 byłem pewien, że elektroniką chcę się zająć zawodowo. Konsekwentnie, gdy nadszedł czas, podjąłem studia związane z elektroniką. Podczas praktyk studenckich w firmie Philips po raz pierwszy w życiu miałem możliwość pracy na prawdziwym komputerze przemysłowym – był to ogromny (mocą obliczeniową, a zwłaszcza wymiarami) PM3200. Był on wyposażony w niezbyt szybki 8-bitowy procesor i miał do tego jedną zaletę: wbudowany interpreter BASIC-a. Ten „superkomputer”, pomimo wielu niedogodności związanych z pracą na nim, polubiłem, bo dzięki BASIC-owi szybko nauczyłem się wykorzystywać oferowane przez niego możliwości.

Po ukończeniu studiów i odbyciu służby wojskowej rozpocząłem pracę w firmie PTT/KPN (holenderski narodowy operator telekomunikacyjny), gdzie nie miałem zbyt dużych możliwości zajmowania się „prawdziwą” elektroniką. Postanowiłem jednak nie zrywać z nią kontaktu i w ten sposób elektronika stała się moim hobby.

W 1984 roku odpowiednik IBM PC kosztował około 5000 USD. Było to zbyt dużo w stosunku do moich możliwości, więc za ok. 1500 USD kupiłem komputer Atari 1040ST (procesor 68000, taktowany 8 MHz), za pomocą którego bardzo szybko nauczyłem się rysować obwody drukowane, co znacznie skracало czas realizacji projektów. Do tego czasu płytki drukowane rysowałem... piórkiem kreslarskim za pomocą werniksu lub lakieru do paznokci. Mój domowy komputer działał doskonale, więc po niedługim czasie zdobyłem interpreter BASIC-a na Atari, który umożliwił mi zautomatyzowanie wykonywania wielu żmudnych zadań. W ten sposób zboczyłem ze ścieżki „elektronicznej” i rozpocząłem nową przygodę – z programowaniem komputerów. Pierwsze kroki w programowaniu stawiałem wykorzystując Quick Basic 2.0, 3.0 i 4.0, następnie Visual Basic.

Wymuszony obowiązkami zawodowymi brak czasu ograniczył liczbę projektów wykonywanych przeze mnie „po godzinach”, lecz cały czas śledziłem rynkowe nowości. W pewnym momencie w moje ręce wpadł mikrokontroler 8052AH z wbu-



Mark Alberts

dowanym w pamięć ROM interpreterem BASIC-a, co po kilku eksperymentach spowodowało, że bardzo zainteresowały mnie mikrokontrolery. Początki nie były łatwe, bowiem do pracy z ówczesnymi mikrokontrolerami niezbędny był spory zestaw kosztownych narzędzi: kasownik UV, programator pamięci EPROM i kompilator asemblera. Tak więc, zanim napisałem pierwszy program, musiałem dużo przeczytać, przeprowadziłem wiele czasochłonnych prób i wiele rzeczy i urządzeń musiałem kupić. Mikrokontroler 8052AH sprawdzał się w praktyce, szczególnie ujęła mnie łatwość jego oprogramowywania, ale był on do wielu zadań zbyt wolny i zdecydowanie zbyt kosztowny. Zacząłem więc szukać alternatywy dla niego.

Ponieważ moje wcześniejsze doświadczenia z BASIC-iem były bardzo dobre, postanowiłem napisać własny kompilator BASIC-a dla mikrokontrolerów '51. Pierwszy kompilator (jeszcze dla DOS) powstał specjalnie dla mikrokontrolera 89C2051. Dlaczego? Ponieważ nie wymagał on specjalnego osprzętu, zwłaszcza kasownika UV, którego nie lubiłem (naczytałem się, jakie są one niebezpieczne dla zdrowia), a ten jeden niewielki układ umożliwił zastąpienie wielu klasycznych układów TTL i CMOS. Dzięki temu płytki drukowane mogły być mniejsze, prostsze i tańsze w wykonaniu.

W chwili gdy napisany przeze mnie kompilator zaczął w miarę dobrze działać, okazało się, że świat jest podbijany przez Windows. Przepisałem więc stworzone oprogramowanie w Visual Basicu 3.0, co pozwalało zastosować je w środowisku Windows 3.1. Wtedy po raz pierwszy pomyślałem, że mój kompilator może okazać się atrakcyjny także dla innych elektroników, którzy mieli problemy podobne do moich. Dodałem więc do kompilatora programowy symulator, obsługę programatorów i kilka innych drobiazgów, które uprościły pracę z programem. Najważniejszym i najbardziej czasochłonnym zadaniem okazało się napisanie pomocy do programu. Pierwsza komercyjna wersja Bascoma LT ujrzała światło dzienne w 1995 roku.

Bardzo szybko okazało się, że moje przewidywania były słuszne – Bascom LT zainteresował wielu użytkowników, którzy zasugerowali mi pewne poprawki i uzupełnienia uwzględnione w kolejnych wersjach (dodałem m.in. obliczenia zmiennoprzecinkowe i obsługę tablic). Wprowadzenie przez Microsoft na rynek 32-bitowej wersji Windows (Windows 95) spowodowało konieczność ponownego zmodyfikowania kodu źródłowego kompilatora, co zaowocowało powstaniem jego nowej wersji. Nazwałem ją Bascom 8051. Szybko rosnąca wśród elektroników popularność mikrokontrolerów AVR zachęciła mnie do wprowadzenia do sprzedaży drugiej wersji kompilatora, przeznaczonej specjalnie dla nich: Bascom AVR.

Teraz pracuję nad nowymi wersjami obydwu kompilatorów, w których uwzględnię wiele uwag i sugestii użytkowników z całego świata. Trwa to dość długo, ponieważ pracuję jako programista w firmie informatycznej, a własnej firmie – MCS Electronics, którą prowadzę z moją żoną Sharon – poświęcam czas tylko wieczorami. Cały czas moim działaniom przyświeca myśl, której owocem jest Bascom: najlepsze narzędzie to takie, które umożliwia skupienie uwagi konstruktora na rozwiązywanym zadaniu, co mam nadzieję – przynajmniej częściowo – udało mi się.

*Things should be made as simple
as possible – but no simpler.*

Albert Einstein

Zbigniew Raabe – wspomnienie

22 grudnia 2001 roku zakończył się jeden z etapów współczesnej historii amatorskiej elektroniki w Polsce: niespodziewanie zmarł Zbigniew Witold Raabe, niestrudzony popularyzator Bascoma, twórca wielu narzędzi do tego kompilatora, wieloletni współpracownik największych polskich pism elektronicznych: Elektroniki Praktycznej i Elektroniki dla Wszystkich. W swojej karierze opracował ponad 200 projektów urządzeń elektronicznych, spośród których blisko połowa ilustrowała niebagatelne możliwości Bascoma. Z większością z tych projektów mogli zapoznać się Czytelnicy EP i EdW, cztery wdrożono na skalę przemysłową (jeden w Polsce, jeden w Niemczech, dwa w Holandii), a niektóre z jego projektów służą do dziś w miejscach tak szacownych jak sala posiedzeń Senatu RP.



Zbigniew Raabe

Jako autor artykułów – ich pisanie zajmowało mu większość czasu ostatnich lat życia – Zbyszek zasłynął z ciętego języka, żartobliwego podejścia do często bardzo poważnych zagadnień, jakie w swych artykułach poruszał (jak choćby opis timera dla osób zażywających lekarstwa), umiejętności wychwytywania nietypowych i pozornie mało sensownych oczekiwań Czytelników swoich artykułów (w ten sposób narodził się m.in. słynny „Pipek dręczyciel”), a także z wielu złotych myśli (powstawały one zazwyczaj podczas pracy nad projektami), które skrzętnie zapisywał i które miały ukazać się drukiem jako „Zbiór praw Raabego dla elektroniki” w Elektronice Praktycznej. Podjął także trud przygotowania książki o Bascomie, lecz nie udało mu się doprowadzić tego projektu, jak i niestety wielu innych, do końca.

Urodził się w Lublinie w 1946 roku. Jego dzieciństwo, przebiegające w trudnych latach powojennych, nie sugerowało niezwykle dalszego przebiegu życia zwłaszcza, że inteligencka rodzina z jakiej pochodził (jego dziadek prof. Henryk Waclaw Raabe był m.in. organizatorem i pierwszym rektorem Uniwersytetu Marii Curie-Skłodowskiej w Lublinie, jego ojciec – Zdzisław Raabe był profesorem PAN, słynnym zoologiem i działaczem politycznym w PRL) nie zawsze akceptowała wybryki młodego, pełnego zaskakujących pomysłów człowieka. Jednym z takich „wybryków” było podjęcie studiów w Wyższej Szkole Filmowej w Łodzi, na Wydziale Operatorskim. Na uczelni realizował część swoich zainteresowań fotografią, miał także okazję spotkać wielkich świata filmu, w tym Romana Polańskiego, z którym nakręcił kilka etiud filmowych. Trudności z dostępem do nowoczesnego sprzętu wspomagającego realizację filmów i robienia zdjęć spowodowały, że do bogatej



Zestaw ewaluacyjny do Bascoma opracowany przez Zbigniewa Raabe w roku 1997

gamy zainteresowań Zbyszka dołączyła elektronika i – we wczesnym stadium swojego rozwoju – informatyka, które to dziedziny z czasem pochłonęły go bez reszty. Kolejne losy rzucały go w różne strony świata (sporo czasu spędził m.in. w Indiach), co nie sprzyjało stabilizacji życia rodzinnego ani zawodowego. Pomimo trudności w 1996 roku dołączył do zespołu Elektroniki Praktycznej, co zapoczątkowało nową kartę w życiu jego, Elektroniki Praktycznej, później Elektroniki dla Wszystkich i oczywiście Czytelników obydwu miesięczników. Byli tacy, którzy nie lubili jego nowatorskich pomysłów i stylu artykułów, byli także tacy, którzy go z wielkim żalem żegnali. Niewielu było natomiast takich, którzy pozostali obojętni, zgodnie z zasadą pięknie ujętą w słowa przez Juliana Tuwima: „Żyj tak, aby twoim znajomym zrobiło się nudno, gdy umrzesz”. Udało mu się!

*Piotr Zbysiński
Redaktor Naczelny
miesięcznika
Elektronika Praktyczna*

Wstęp

Język Basic pamięta każdy, kto zetknął się z pierwszymi mikrokontrolerami 8-bitowymi: Spectrum, Atari, Commodore. Ze względu na prostotę składni oraz niewielką liczbę instrukcji, był implementowany w tych mikrokomputerach. Ja miałem Commodore i pierwsze, proste programy pisałem właśnie w języku Basic. Choć język ten stracił w ostatnich latach sporo na znaczeniu, to jednak jeszcze nie wyszedł z użycia. Stosowany jest nadal w różnych odmianach, jak między innymi Visual Basic. Jedną z odmian tego popularnego niegdyś języka jest język programowania mikrokontrolerów, którego kompilatory są zawarte w pakietach Bascom AVR i Bascom 51. Są to pakiety programów służących do programowania mikrokontrolerów produkowanych przez firmę ATMEL. Oferowanie dwóch oddzielnych pakietów Bascom jest spowodowane różną architekturą mikrokontrolerów AVR i 51. Zastosowany w Bascom język przygotowywania programów trochę przypomina właśnie Basic. Choć podstawowe instrukcje i składnia są podobne jak w Basicu, jednak zwiększono znacznie liczbę instrukcji związanych z samymi mikrokontrolerami. Ułatwiają one pracę programiście i przyspieszają przygotowywanie programów.

Ze względu na zwiększającą się popularność mikrokontrolerów AVR oraz niewielkie różnice między Bascom AVR a Bascom 51, książka została poświęcona właśnie bardziej perspektywnemu pakietowi Bascom AVR. Jednak poznawszy go, bez większych trudności będzie można pisać programy także w Bascom 51.

Dużą zaletą mikrokontrolerów AVR jest łatwość ich programowania przez SPI. Nie są potrzebne zewnętrzne – zazwyczaj drogie – programatory. Aby zaprogramować mikrokontroler wystarczy jedynie dołączyć programator z kilkoma przewodami do portu równoległego komputera.

Pakiet Bascom AVR umożliwia szybkie i łatwe przygotowywanie programów dla większości mikrokontrolerów AVR bez dokładnej znajomości ich peryferiów oraz rejestrów, co jest niezbędne przy pisaniu w assemblerze czy w C. W Bascomie mikrokontrolery można traktować jako czarne skrzynki.

Bascom umożliwia szybkie programowanie, gdyż ma wbudowanych wiele gotowych instrukcji (procedur i funkcji) do obsługi wyświetlaczy LCD (alfanumerycznych oraz graficznych), interfejsów komunikacyjnych (I²C, 1-Wire czy RS232), stosu TCP/IP i wielu innych układów peryferyjnych. Na przykład, aby wyświetlić tekst (stałą) na wyświetlaczu LCD wystarczy napisać instrukcję `Lcd "tekst"`. Po zaprogramowaniu mikrokontrolera na wyświetlaczu będzie wyświetlony odpowiedni tekst wpisany w cudzysłowie. Chyba prościej już nie można, co skłania do wniosku, że programowanie mikrokontrolerów może być proste, łatwe i przyjemne.

Z każdym miesiącem, w uaktualnianych wersjach pakietu, przybywa wiele nowych instrukcji umożliwiających realizację coraz bardziej wymyślnych projektów. W skład pakietu wchodzi wiele narzędzi programowych, które mają na celu ułatwienie przygotowywania własnych programów i zaprogramowania mikrokontrolerów jak: symulator, program komunikacyjny (terminal) czy program obsługujący różne programatory SPI.

Do spopularyzowania Bascoma w Polsce przyczynił się ś.p. redaktor Zbigniew Raabe. Zafascynowały go możliwości i prostota tego pakietu. To on na łamach miesięczników „Elektronika Praktyczna” oraz „Elektronika dla Wszystkich” prezentował możliwości Bascoma na przykładzie kilkunastu rewelacyjnych projektów z wykorzystaniem AVR i 51. Jego eksperymenty z tym oprogramowaniem zaowocowały kilkumiesięcznym kursem na łamach „Elektroniki dla Wszystkich”, który zainteresował Bascomem wielu Czytelników i powiększył grono jego entuzjastów, w tym również mnie.

Ponieważ dotąd napisałem tylko o zaletach Bascoma, to należy wspomnieć także o jego wadach. Jedną z głównych wad jest generowanie zazwyczaj długiego kodu wynikowego, który w przypadku zastosowania języka C jest o wiele krótszy. Jednak przy współczesnych mikrokontrolerach, które mają sporo pamięci Flash nie jest to aż tak duża wada. Kod wynikowy można znacznie zmniejszyć poprzez umiejętnie wykorzystywanie instrukcji, a także poprzez wstawki assemblerowe.

Książka jest przeznaczona dla praktyków elektroników i hobbystów, którzy słyszeli lub czytali coś o Bascomie, wiedzą co to jest mikrokontroler i chcieliby szybko opanować sztukę programowania tych układów. Książka może z pewnością zainteresować także tych, którzy chcą rozwijać swoje umiejętności programowania w tym pakiecie. W książce przedstawiono – krok po kroku – posługiwanie się oprogramowaniem pakietu Bascom, etapy symulacji programowej i – co bardzo ważne – sprzętowej (nie jest potrzebny drogi emulator sprzętowy), sposób przygotowywania programów, obsługę przerw oraz konfigurowanie peryferiów. Nie opisano dokładnie wszystkich instrukcji, po których opis znaczenia można sięgnąć do rewelacyjnego *Helpa* (w wersji spolszczonej jest dostępny w Internecie pod adresem www.mcselec.com/download/avr/bascomavr.pl.zip), lecz skupiono się na przedstawieniu Bascoma na przykładach. Zamieszczono wiele przykładów poświęconych obsłudze bloków peryferyjnych mikrokontrolera oraz najczęściej dołączanych urządzeń zewnętrznych (wyświetlacze, klawiatury itp.). Aby umożliwić samodzielną realizację przykładów i innych ćwiczeń, zaprojektowano układ uruchomieniowy ZL2AVR. Zastosowany w nim mikrokontroler ATmega8 ma wiele bloków peryferyjnych, co pozwala na sprawdzenie wszystkich przykładów. Zestaw nie jest dedykowany tylko dla Bascoma, ale może być dowolnie wykorzystywany do ćwiczeń z innymi językami programowania.

Jak wspomniano, różnicowanie pakietów Bascom AVR i Bascom 51 wynika z różnej budowy mikrokontrolerów rodziny 51 i nowoczesnych AVR. Chociaż różnice pomiędzy budową mikrokontrolerów 51 i AVR są spore, to różnice w możliwościach i obsłudze Bascom AVR i Bascom 51 są niewielkie, a więc bez większych trudności można zacząć programować mikrokontrolery z rodziny 51 w Bascom 51 znając jedynie Bascom AVR. Pakiet Bascom AVR został bowiem tak opracowany, aby był w największym stopniu kompatybilny z Bascom 51. Jednak niektóre instrukcje musiały być usunięte, zmienione lub dodane. Największe zmiany dotyczą konfigurowania portów, bloków peryferyjnych oraz wprowadzenia instrukcji potrzebnych dla procesorów rodziny 51 a usunięcia zbędnych. Dla tych mikrokontrolerów nie trzeba wcześniej konfigurować portów. Na przykład, aby ustawić linię portu jako wejściową, wystarczy w Bascom 51 ustawić ją na 1. Dla mikrokontrolerów AVR

należy wcześniej skonfigurować porty. Różnice występują także w konfigurowaniu bloków peryferyjnych.

Dużą zaletą mikrokontrolerów AVR jest ich szybkość działania, o wiele większa od szybkości mikrokontrolerów 51, przy tej samej częstotliwości zegara. Mają one także więcej wewnętrznych bloków peryferyjnych. Dużą szybkość działania tych procesorów zapewnia przetwarzanie potokowe, dzięki któremu większość rozkazów jest wykonywana w jednym cyklu zegarowym. Ponadto mikrokontrolery AVR mają niewielką listę rozkazów (architektura RISC), a brak ścisłego sprecyzowania akumulatora, którego rolę może pełnić dowolnie wybrany rejestr, zapewnia znaczny wzrost szybkości wykonywania programów.

Dużym udogodnieniem dla projektantów jest wbudowanie pamięci EEPROM, którą można przeznaczyć do zapamiętywania danych, które muszą być odtworzone po zaniku napięcia zasilania. Większość mikrokontrolerów ma wbudowany generator RC, dlatego w układach, w których precyzyjne odmierzenie czasu nie jest konieczne, można zrezygnować z zewnętrznego rezonatora kwarcowego. W mikrokontrolerach AVR jest dostępnych wiele trybów pracy z obniżonym poborem mocy, dlatego nadają się one do zastosowania w urządzeniach zasilanych bateryjnie.