

SDC_Zero – komputer retro do zmontowania w pół godziny

Oprogramowanie mikrokontrolera niezbędne do działania komputera SDC_Zero jest dostępne na portalu EP w pliku SDC0.zip.

W cyklu artykułów publikowanych w ciągu minionego roku przedstawiliśmy projekt komputera SDC_One, złożonego z jednego z kilku typów zabytkowych mikroprocesorów i mikrokontrolera STM32L476 umieszczonego na płytce uruchomieniowej Nucleo-64. Po zrealizowaniu tego projektu powstał pomysł wykonania podobnej w założeniach konstrukcji, którą łatwo można byłoby zmontować w warunkach domowych, używając wyłącznie tanich i łatwo dostępnych elementów. W ten sposób powstał projekt SDC_Zero – minimalistyczny, lecz w pełni funkcjonalny komputer zdefiniowany programowo, który można zmontować w domu w ciągu ok. pół godziny, kosztem około 30 złotych.

SDC_Zero składa się z taniej i popularnej płytki uruchomieniowej STM32F103C8T6 Minimum Development Board, znanej pod nazwą BluePill, oraz mikroprocesora typu 65C02 lub 80C85. Elementy te są umieszczone na popularnej płytce stykowe typu „830”. Do wykonania komputera potrzeba ponadto ok. 40 przewodów połączeniowych (w tym jednego zakończonego z jednej strony wtykiem, a z drugiej – gniazdem) i dwóch kondensatorów ceramicznych 100 nF. W wersji z procesorem 65C02 potrzebny jest ponadto rezystor 10 kΩ, a przy użyciu procesora 8085 w wersji NMOS – kilka rezystorów 4,7 kΩ i dowolnych diod prostowniczych (np. 1N4148 lub 1N5819). Wygląd zmontowanego komputera SDC_Zero pokazano na fotografii 1.

Ograniczenie repertuaru dostępnych procesorów w porównaniu do SDC_One wynika z niewielkiej liczby wyprowadzeń użytego na płytce BluePill mikrokontrolera STM32F103C8T6. Do dyspozycji jest tu tylko 28 sygnałów, które mogą być wykorzystane do współpracy z szyną procesora. Taka liczba sygnałów wystarcza do obsługi bardzo prostej szyny mikroprocesora 65C02 lub bardziej złożonej, lecz multipleksowanej szyny 80C85.

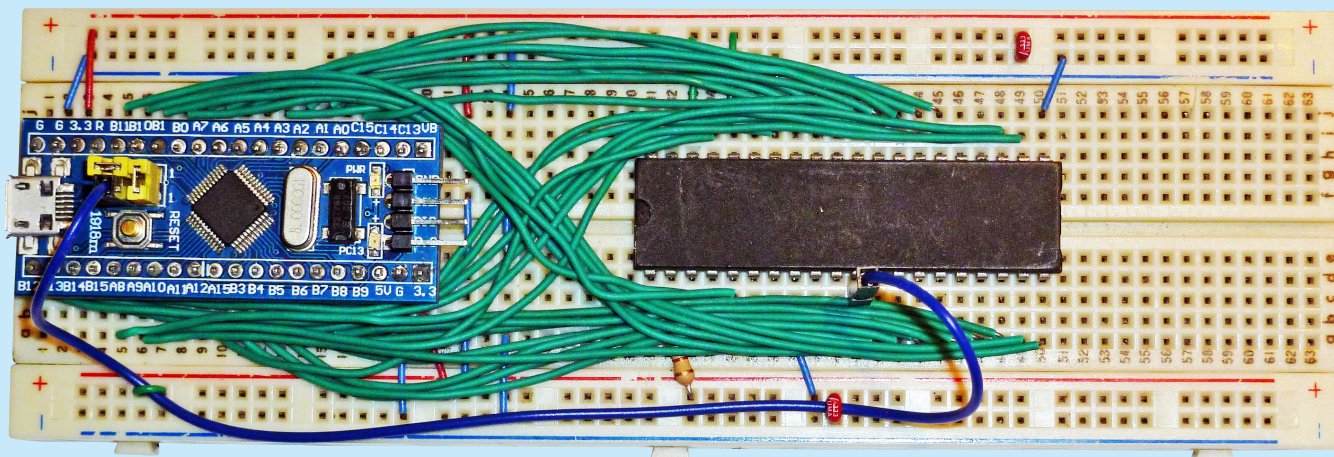
Zbudowany w ten sposób komputer jest wyposażony w 16 kB pamięci RAM i 64 kB pamięci ROM, co umożliwia uruchomienie na nim np. interpretera języka Basic. Jako terminal służy komputer PC z programem TeraTerm lub podobnym. SDC_Zero jest połączony

z komputerem PC przy użyciu kabla USB, zapewniającego zasilanie oraz komunikację. Oprogramowanie urządzenia realizuje złożone urządzenie USB, udostępniające dwa wirtualne porty szeregowy – jeden do współpracy z monitorem sprzętu, drugi do komunikacji z komputerem docelowym.

Monitor sprzętu umożliwia pełne sterowanie komputerem na poziomie szyny procesora, w tym:

- inicjowanie (RESET) i generowanie przebiegu zegarowego,
- wyświetlanie i edycję zawartości pamięci,
- ładowanie do pamięci programów w postaci plików Intel HEX i Motorola S-record,
- pracę krokową na poziomie cykli szyny lub instrukcji z wyświetlaniem wykonywanych transferów i deasemblacją instrukcji,
- pracę ciągłą z opcjonalnym pułapkowaniem i gromadzeniem śladu wykonania,
- programowanie pamięci ROM komputera docelowego poprzez kopiowanie zawartości pamięci RAM.

Dzięki takiej funkcjonalności komputer może być używany jako narzędzie dydaktyczne – do demonstrowania działania procesora (w tym przebiegu wykonania instrukcji, adresowania pamięci, operacji stosowych i obsługi przerwań) oraz do nauki programowania assemblerowego.



Fotografia 1. Zmontowany komputer SDC_Zero z mikroprocesorem 65C02

Schematy

Linie portów mikrokontrolera STM32F103C umieszczonego na płytce BluePill, które mogą być użyte do komunikacji z procesorem komputera SDC_One, to PA0..10, PA15, PB0..15 i PC14..15. Pozostałe linie służą do programowania mikrokontrolera i do komunikacji przez interfejs USB. Linia PB2 mikrokontrolera jest dostępna tylko na złączu BOOT1, a dostęp do niej wymaga wprowadzenia drobnej modyfikacji na płytce.

Na schematach na rysunkach 2 i 3 pokazano połączenia linii mikroprocesorów. Komputer został zaprojektowany głównie do współpracy z mikroprocesorami wykonanymi w technologii CMOS i zasilanymi napięciem 3,3 V. Połączenia z portami mikrokontrolera zostały zaplanowane tak, by zminimalizować liczbę elementów dopasowujących w przypadku zastosowania w komputerze mikroprocesora z zasilaniem 5 V. Niektóre linie mają wymuszone stałe poziomy

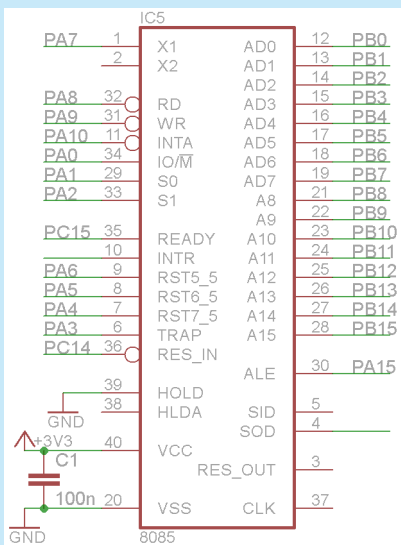
logiczne poprzez połączenie z linią zasilania bezpośrednio lub przez rezystor. Linia RDY mikrokontrolera 65C02 musi być dołączona do zasilania przez rezystor rzędu 10 kΩ.

W komputerze nie może być użyty mikrokontroler 6502 wykonany w technologii NMOS z powodu zastosowania w nim struktur dynamicznych uniemożliwiających dłuższe wstrzymanie cyklu szyny.

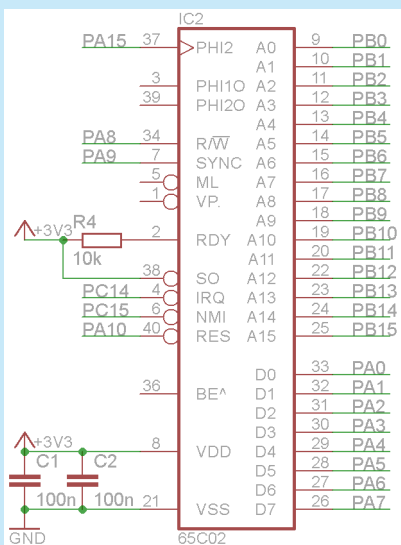
W wersji z procesorem 80C85, w przypadku niedostępności wersji CMOS, mogącej pracować z zasilaniem 3,3 V, można użyć mikroprocesora NMOS z zasilaniem 5 V, jednak niezbędne będzie wtedy dodanie szeregowych rezystorów zabezpieczających o wartości 4k7 na połączeniach procesora z liniami PA0, PA1, PA2, PB0, PB1 i PB5 mikrokontrolera (pozostałe linie portów albo są używane jako wyjścia, albo tolerują poziomy logiczne 5 V).

Modyfikacja płytki BluePill

Przed przystąpieniem do montażu komputera należy poddać niewielkim modyfikacją płytkę BluePill. Ponieważ komputer używa linii PB2, która na płytce BluePill jest połączona przez rezystor szeregowy R4 (100 kΩ, rozmiar 0603) ze złączem zwór konfiguracyjnych, niezbędne jest zastąpienie go rezystorem o wartości nieprzekraczającej 10 kΩ lub zwarcie go krótkim odcinkiem cienkiego drutu (np. kyanar). Ponadto dla wygody użytkownika warto wymienić rezystory



Rysunek 2. Schemat SDC_Zero z mikroprocesorem 80C85



Rysunek 3. Schemat SDC_Zero z mikroprocesorem 65C02

szeregowe diod LED R1 i R5 (typowo 510R) na rezystory o wartości 2..3 kΩ, by zredukować jasność świecenia diod.

Programowanie mikrokontrolera

Mikrokontroler STM32F103 umieszczony na płytce BluePill może być programowany przez interfejs SWD przy użyciu interfejsu ST-Link (np. z płytki serii Nucleo lub Discovery) lub innego interfejsu SWD. Jeśli nie dysponujemy programatorem z interfejsem SWD, możemy wgrać oprogramowanie do pamięci mikrokontrolera, korzystając z wbudowanego bootloadera mikrokontrolera.

Programowanie przy użyciu interfejsu ST-Link

W celu zaprogramowania mikrokontrolera przy użyciu interfejsu SWD z płytki Nucleo należy:

- Usunąć z płytki Nucleo zwory łączące interfejs SWD z mikrokontrolerem umieszczonym na płytce Nucleo.
- Połączyć cztery linie dostępne na złączu programowania płytki BluePill z odpowiednimi złączami płytki Nucleo; linię 3,3 V należy połączyć z jedną z dostępnych linii zasilania 3,3 V na płytce Nucleo, a GND, SWDIO i SWCLK – ze złączem ST-Link.
- Uruchomić program ST-Link Utility, a następnie załadować plik `sdc0_65.bin` lub `sdc0_85.bin`, stosownie do typu używanego mikroprocesora.
- Nacisnąć przycisk Program-Verify.

Programowanie mikrokontrolera przy użyciu wbudowanego bootloadera

Ponieważ bootloader korzysta z linii PA9 i PA10, programowanie musi nastąpić przed połączeniem płytki BluePill z mikroprocesorem naszego komputera. Do zaprogramowania płytki potrzebujemy interfejsu USB-UART z liniami UART na poziomach logicznych 3,3 V. Podczas programowania możemy zasilić płytkę z interfejsu USB-UART, doprowadzając napięcie 3,3 V do jednego ze styków 3,3 V płytki BluePill. Do programowania mikrokontrolera należy użyć programu Flash Loader Demonstrator, udostępnianego przez ST Microelectronics na witrynie internetowej (plik zawierający program nosi nazwę `STSW-MCU005.zip`).

W celu zaprogramowania mikrokontrolera przy użyciu bootloadera należy:

- Połączyć masę z wyjście zasilania 3,3 V modułu USB-UART z masą i zasilaniem 3,3 V płytki BluePill.
- Połączyć wyjście TX modułu z linią PA10 mikrokontrolera, a wejście RX – z linią PA9.
- Ustawić zworę BOOT0 w pozycję 1, a zworę BOOT1 w pozycję 0.
- Po uruchomieniu programu Flash Loader Demonstrator wybrać właściwy interfejs szeregowy (numer portu COM przypisany do używanego interfejsu możemy sprawdzić w menedżerze urządzeń), pozostawiając wartości domyślne pozostałych ustawień. Po pomyślnym nawiązaniu komunikacji przechodzimy przez kilka kolejnych dialogów z informacjami o stanie układu, a następnie w dialogu programowania należy wybrać plik `sdc0_65.bin` lub `sdc0_85.bin`, stosownie do typu używanego mikroprocesora i nacisnąć przycisk Next.
- Po zaprogramowaniu mikrokontrolera należy przestawić zworę BOOT0 w pozycję 0 i usunąć zworę Boot1, tak by nie wpływała ona na stan linii PB2, używanej do współpracy z mikroprocesorem

Montaż komputera

Montaż SDC_Zero rozpoczynamy od umieszczenia na skraju płytki stykowej płytki BluePill, orientując ją złączem microUSB na zewnątrz płytki stykowej. Wyprowadzenia GND i 3,3 V płytki BluePill

łączymy z obiema szynami zasilania płytki stykowej. Jeżeli używamy mikrokontrolera zasilanego napięciem 5 V, zamiast wyjścia 3,3 V łączymy z szyną zasilania wyjście 5 V płytki BluePill.

Wersja z 65C02

Aktualna dokumentacja mikroprocesora jest dostępna u producenta – westerndesigncenter.com. Mikrokontroler umieszczamy na płytce stykowej, pozostawiając ok. 10 rzędów styków wolnych pomiędzy płytką BluePill i mikrokontrolerem, orientując mikrokontroler wycięciem (wyprowadzeniami 1 i 40) w stronę płytki BluePill. Wykonujemy połączenia zasilania i wejść mikrokontrolera zwartych z liniami zasilania lub masy i łączymy wejście RDY poprzez rezystor 10 kΩ z dodatnim biegunem zasilania. Na szynie zasilania, w pobliżu doprowadzenia bieguna dodatniego i masy do mikrokontrolera, umieszczamy kondensatory 100 nF. Następnie wykonujemy połączenia pomiędzy mikroprocesorem i BluePill w następującej kolejności (wynikającej z topologii połączeń i wygody ich prowadzenia):

- szyna danych D0...D7 z PA0...PA7,
- IRQB z PC14, NMIB z PC15,
- linie adresu A3...A9 z PB3...PB9,
- linie adresu A12...A15 z PB12...PB15,
- linie adresu A0, A1, A10, A11 z PB0, PB1, PB10, PB11,
- linie RWB, SYNC, RESB, PHI2 z PA8, PA9, PA10 i PA15,
- linia adresu A2 do PB2 – środkowy styk zwory BOOT1, przy przycisku RESET na płytce BluePill.

Wersja z 80C85

Mikrokontroler umieszczamy na płytce stykowej, pozostawiając ok. 10 rzędów styków wolnych pomiędzy płytką BluePill i mikrokontrolerem, orientując mikrokontroler wyprowadzeniami 20 i 21 w stronę płytki BluePill (wycięciem w kierunku przeciwnym do płytki). Wykonujemy połączenia zasilania i wejść mikrokontrolera zwartych z liniami masy (INTR, HOLD). Na szynie zasilania, w pobliżu doprowadzenia bieguna dodatniego i masy do mikrokontrolera, umieszczamy kondensatory 100 nF. Następnie wykonujemy połączenia pomiędzy mikroprocesorem i BluePill w następującej kolejności:

- szyna adresowa A8...A15 z PB8...PA15,
- linie adresu/danych AD0, AD1, AD3...AD7 z PB0, PB1, PB3...PB7,
- wszystkie linie sterowania i stanu wg tabeli 1,
- linia adresu A2 do PB2 – środkowy styk zwory BOOT1, przy przycisku RESET na płytce BluePill.

Użytkowanie SDC_Zero

Do działań z komputerem potrzebny jest program emulator terminalu działający na komputerze PC. W roli tej dobrze sprawdza się np. popularny TeraTerm. Komputer jest widoczny jako dwa wirtualne porty szeregowo, z których pierwszy służy do komunikacji z monitorem sprzętu, a drugi – do komunikacji z komputerem docelowym.

Monitor sprzętu

Monitor sprzętu służy do sterowania pracą komputera. Umożliwia on m.in.:

- inicjowanie procesora sygnałem RESET,
- pracę krokową mikroprocesora w cyklach szyny lub cyklach instrukcyjnych,
- pracę ciągłą z opcjonalnym pułapkowaniem i gromadzeniem śladu,
- przeglądanie i edycję zawartości pamięci,
- ładowanie do pamięci programów poprzez przeciągnięcie pliku HEX do okna monitora sprzętu,
- programowanie dostępnej dla procesora pamięci Flash zawartością pamięci RAM.

Tabela 1. Zestawienie połączeń mikroprocesorów 65C02 i 80C85 z płytką mikrokontrolera STM32F103 w komputerze SDC_Zero

	STM32F103		W65C02S		80C85	
			nazwa	nr	nazwa	nr
PA	0	D0	33	IO/-M	34	
	1	D1	32	S0	29	
	2	D2	31	S1	33	
	3	D3	30	TRAP	6	
	4	D4	29	RST7.5	7	
	5	D5	28	RST6.5	8	
	6	D6	27	RST5.5	9	
	7	D7	26	X1	1	
	8	RWB	34	-RD	32	
	9	SYNC	7	-WR	31	
	10	RESB	40	-INTA	11	
	11					
	12					
	13					
	14					
	15	PHI2	37	ALE	30	
PB	0	A0	9	AD0	12	
	1	A1	10	AD1	13	
	2	A2	11	AD2	14	
	3	A3	12	AD3	15	
	4	A4	13	AD4	16	
	5	A5	14	AD5	17	
	6	A6	15	AD6	18	
	7	A7	16	AD7	19	
	8	A8	17	A8	21	
	9	A9	18	A9	22	
	10	A10	19	A10	23	
	11	A11	20	A11	24	
	12	A12	22	A12	25	
	13	A13	23	A13	26	
	14	A14	24	A14	27	
	15	A15	25	A15	28	
PC	13					
	14	IRQB	4	RES_IN	36	
	15	NMIB	6	READY	35	
Zasilanie						
	3V3	Vdd	8	Vcc	40	
	GND	Vss	21	GND	20	
Stałe wejścia						
	3V3	BE	36			
	3V3	SOB	38			
	3V3 p.10k	RDY	2			
	GND			HOLD	39	
	GND			INTR	10	

Tabela 2. Ważniejsze polecenia monitora sprzętu SDC_Zero

Argumenty	Polecenie	Opis
	?	Wyświetlenie listy poleceń z opisami
	?clk	Wyświetlenie bieżących wartości okresu i częstotliwości zegara procesora
<period>	clkp	Ustawienie okresu zegara w nanosekundach
	run	Uruchomienie w trybie pracy ciągłej
<addr>	r2	Uruchomienie w trybie pracy ciągłej z zatrzymaniem przy dostępie do podanego adresu
	ct	Uruchomienie w trybie pracy ciągłej na 1 sekundę i wyświetlenie liczby wykonanych cykli szyny (test wydajności komputera)
<intnum>	is	Zgłoszenie przerwania
<intnum>	ic	Wycofanie zgłoszenia przerwania
	mi	Załadowanie domyślnej zawartości pamięci
	rst	Sprzętowe inicjowanie procesora, inicjowanie peryferii i ustawienie trybu pracy krokowej w cyklach szyny
	sc	Wykonanie jednego cyklu szyny
	si	Wykonanie jednego cyklu instrukcyjnego z zatrzymaniem przy rozpoczęciu pobierania następczej instrukcji
	s	Wykonanie jednego cyklu (takiego jak ostatni – szyny lub instrukcyjnego)
<num>	ns	Wykonanie zadanej liczby cykli
<val>	dbz	Wymuszenie zawartości szyny danych w cyklu szyny, niezależnie od zawartości pamięci lub portu wejściowego
<ioaddr>	ior	Wyświetlenie bieżących zawartości portów wejściowych bez zmiany stanu peryferii wynikającej z odczytu
<ioaddr>	iot	Wyświetlenie bieżących zawartości portów wejściowych z ew. zmianą stanu peryferii wynikającej z odczytu, jak przy odczycie przez procesor
<data> <ioaddr>	iow	Zapis wartości do portu wyjściowego
<addr>	md	Wyświetlenie zawartości 128 bajtów pamięci i modyfikacja adresu na stosie argumentów do kolejnej wielokrotności 128
<len> <val> <addr>	mf	Wypełnianie obszaru pamięci stałą zawartością
	mr	Zainicjowanie pamięci RAM zawartością pamięci Flash
<val> <addr>	mw	Zapis bajtu do pamięci
<val> <addr>	mww	Zapis słowa 16-bitowego do pamięci
<target> <addr>	wj	Zapis do pamięci instrukcji skoku do zadanego adresu
<0/1>	mmm	Wielokrotne odwzorowanie pamięci RAM w całej przestrzeni adresowej procesora
<addr> <size>	ms	Programowanie pamięci Flash na podstawie zawartości pamięci RAM od zadanego adresu. Rozmiar zaokrąglony w górę do wielokrotności 1 kB
<addr> <size>	mse	Kasowanie pamięci Flash od zadanego adresu. Rozmiar zaokrąglony w górę do wielokrotności 1 kB

Interakcja z monitorem odbywa się przez terminal przy użyciu poleceń wpisywanych z klawiatury. Monitor umożliwia przywołanie kilku ostatnich poleceń i ich edycję. Monitor jest napisany w konwencji maszyny stosowej – argumenty poprzedzają nazwę

```

COM14:115200bps - Tera Term VT
File Edit Setup Control Window Help

SDC_Zero 65C02 Software-Defined Computer
16 KiB RAM, fv.ib1 (? - command help)

0F d390:ea NOP >si
RD d390:ea
RD 01fa:ff
RD 01f9:ff
RD 01f8:ff
RD fffc:00
RD fffd:ff
0F ff00:a2 LDX #fc >
RD ff01:fc
0F ff02:9a TXS >
RD ff03:a0
0F ff03:a0 LDY #00 >
RD ff04:00
0F ff05:b9 LDA ff4f,Y >ct
256687 bus cycles/second
0F ff32:ad LDA fe03 >run
###:### >

```

Rysunek 4. Przykładowa sesja z monitorem sprzętu – start procesora 65C02 w trybie pracy krokowej, test wydajności, uruchomienie programu z pełną szybkością

polecenia. Argumenty poleceń są liczbami; mogą one być podawane w postaci dziesiętnej, ósemkowej, szesnastkowej lub binarnej, zgodnej ze składnią języka C. Najważniejsze polecenia monitora sprzętu zestawiono w tabeli 2.

Wybrane polecenia monitora, w tym polecenie przeglądania pamięci (md) i pracy krokowej (si, sc), są automatycznie powtarzane po naciśnięciu klawisza Enter, bez potrzeby powtórnego wpisywania polecenia (rysunek 4).

Pamięć

Komputer docelowy dysponuje pamięcią RAM o pojemności 16 kB i pamięcią Flash o pojemności 64 kB. Obie pamięci są udostępniane przez mikrokontroler STM32F103. Ponieważ mikroprocesory używane w SDC_Zero mają przestrzeń adresową o wielkości 64 kB, równocześnie mogą one mieć dostęp do 16 kB RAM i 48 kB ROM. W SDC_Zero przyjęto następujące zasady dostępu do pamięci:

- Jeśli pamięć Flash nie jest pusta, przy starcie komputera zawartość pierwszych 16 kB pamięci Flash jest kopiowana do pamięci RAM.
- Podczas normalnej pracy w przestrzeni adresowej procesora pamięć RAM jest widoczna pod adresami 0x0000..0x3FFF, a ostatnie 48 kB pamięci Flash – pod adresami 0x4000..0xFFFF.

SDC_One Mini_IO registers										Function	
Name	Addr	7	6	5	4	3	2	1	0		
LED	+0	FUN								LEDR	LED on/off control
CON	+1	r:RD w:TD									console data
TIM	+2	DIV8			PERIOD						timer period
CSR	+3							TIF	TXE	RXNE	status

Rysunek 5. Rejestry sterownika wejścia-wyjścia SDC_Zero

```

COM22:115200bps - Tera Term VT
File Edit Setup Control Window Help

TINY BASIC
OK
>10 FOR I=1 TO 6
>20 PRINT I
>30 NEXT I
>RUN
1
2
3
4
5
6
OK
>

```

Rysunek 6. Sesja 80C85 z interpreterem Tiny Basic

```

COM22:115200bps - Tera Term VT
File Edit Setup Control Window Help

Cold [C] or warm [W] start?
MEMORY SIZE?
TERMINAL WIDTH?

19871 BYTES FREE

OSI 6502 BASIC VERSION 1.0 REV 3.2
COPYRIGHT 1977 BY MICROSOFT CO.

OK
10 FOR I=1 TO 5
20 PRINT I
30 NEXT I
RUN
1
2
3
4
5
OK
>

```

Rysunek 7. Sesja 65C02 z interpreterem OSI Basic

- Używając polecenia monitora sprzętu (mmm), można odwzorować pamięć RAM w całej przestrzeni adresowej – ten sam blok o rozmiarze 16 kB jest wtedy widoczny pod adresami 0x0000, 0x4000, 0x8000 i 0xC000.
- Po załadowaniu zawartości pamięci RAM może ona zostać skopiowana do pamięci Flash przy użyciu polecenia monitora sprzętu. W ten sposób można zaprogramować całą pamięć Flash, ładując zawartość poszczególnych jej bloków do pamięci RAM.

Monitor umożliwia ładowanie do pamięci komputera dowolnych programów zapisanych w postaci plików Intel HEX lub Motorola S-record. Programy mogą być pisane i translowane na komputerze PC. Załadowania programu do pamięci polega na przeciągnięciu ikony pliku do okna monitora sprzętu w TeraTerm i potwierdzeniu operacji przesłania pliku. Dzięki temu można szybko i wygodnie uruchomić na komputerze dowolny program.

Wejście-wyjście

SDC_Zero jest wyposażony w sterownik wejścia-wyjścia, zawierający interfejs terminalu, timer i jednobitowy port wyjściowy sterujący diodą LED umieszczoną na płytce BluePill. Sterownik zajmuje 4 lokacje w przestrzeni adresowej. W wersji z 80C85 jest on umieszczony pod adresem 0 w przestrzeni wejścia-wyjścia; w wersji z 65C02 sterownik jest położony pod adresem 0xFE00 w jednolitej przestrzeni adresowej.

Rejestry sterownika przedstawiono na **rysunku 5**. Sterownik stanowi podzbiór bloku wejścia-wyjścia komputera SDC_One. Rejestr LED służy do sterowania świeceniem diody. Rejestr CON zawiera dane przysyłane do terminalu i odbierane z terminalu. Rejestr TIM służy do programowania okresu timera. Okres może być zadawany w milisekundach (DIV8 = 0) lub w jednostkach 8 ms (DIV8 = 1). Rejestr CSR zawiera trzy znaczniki: TIF – znacznik końca okresu timera, TXE – gotowość do transmisji danych terminalu, RXNE – obecność danych odebranych.

Oprogramowanie

Do translacji programów assemblerowych użyto popularnego assemblera TASM (Table-driven Assembler).

Na obu wersjach komputera SDC_Zero uruchomiono interpretry języka BASIC z lat 70. Dla wersji z 80C85 jest to TinyBasic, a wersji z 65C02 – OSI Basic. Oba interpretry są dostępne w sieci w postaci

źródłowej. Wersje programów dostosowane dla SDC_One zostały udostępnione w pliku towarzyszącym artykułowi. Adaptacja polegała głównie na napisaniu procedur związanych z obsługą terminalu.

Tiny BASIC 8080. Interpreter TinyBasic dla mikroprocesora 8080 został napisany w 1976 roku i zajmuje zaledwie 2 kB pamięci. Może on działać na procesorach zgodnych z 8080, w tym na 8085 i Z80. Jest on napisany w taki sposób, że musi być umieszczony w pamięci, począwszy od adresu 0. Interpreter został poddany niezbędnym modyfikacjom i doprowadzony do postaci translowalnej przy użyciu assemblera TASM (**rysunek 6**). Po zresetowaniu procesora, w stanie zatrzymania, należy załadować pamięć RAM zawartością pliku .HEX interpretera. Interpreter można uruchomić z pamięci RAM poleceniem run monitora sprzętu. Można go również umieścić na stałe w pamięci Flash, skąd będzie automatycznie kopiowany do pamięci RAM przy każdym uruchomieniu komputera. W celu zapisania interpretera do pamięci flash należy w monitorze sprzętu wpisać polecenie `0 0x800 ms`.

Interpreter OSI BASIC dla 6502 z 1977 roku jest jednym z pierwszych produktów firmy Microsoft. Z powodu błędów w aktualnej wersji programu TeraTerm (4.100) interpreter nie może być załadowany w edytorze tekstu na dwa pliki o rozmiarze poniżej 16 kB, bas65-1.hex i bas65-2.hex, które należy kolejno załadować do pamięci. Interpreter jest umieszczony w pamięci Flash, począwszy od adresu 0xd000 i używa pamięci RAM od adresu 0. W celu zapisania interpretera do pamięci Flash należy:

- Zresetować procesor i pozostawić go w stanie zatrzymania – polecenie rst.
- Włączyć odwzorowanie pamięci RAMN w całej przestrzeni adresowej – polecenie 1 mmm.
- Załadować do pamięci oba pliki HEX, przeciągając je kolejno do okna terminalu monitora sprzętu i odczekując na zakończenie ładowania (czas rzędu 1...2 sekund).
- Zaprogramować pamięć Flash – polecenie 0xd000 0x3000 ms.
- Wyłączyć wielokrotne odwzorowanie RAM – polecenie 0 mmm.
- Uruchomić komputer – polecenie run (**rysunek 7**).

Grzegorz Mazur
Instytut Informatyki
Politechnika Warszawska