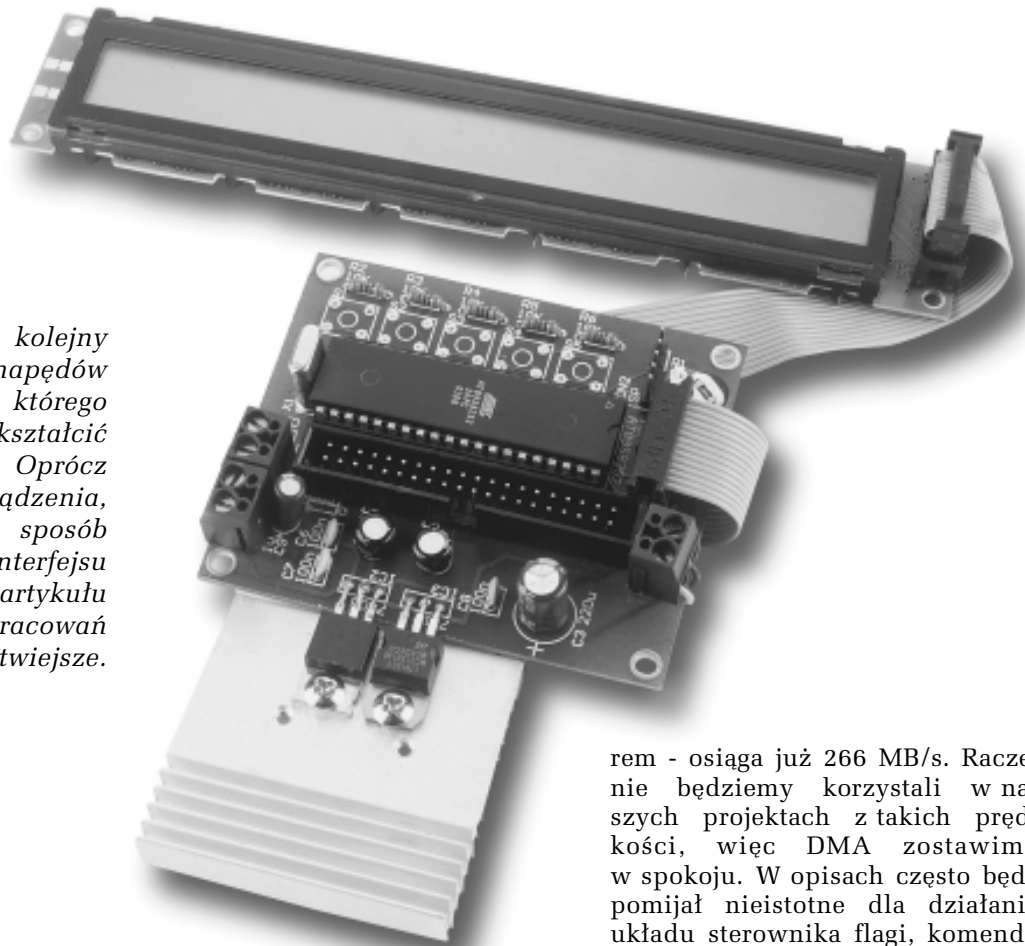


Sterownik napędu CD-ROM, część 1

AVT-5078

Przedstawiamy kolejny projekt sterownika napędów CD-ROM, za pomocą którego można je przekształcić w odtwarzacze audio. Oprócz opisu budowy urządzenia, szczegółowo opisano sposób programowej obsługi interfejsu ATAPI. Po lekturze artykułu tworzenie własnych opracowań będzie z pewnością łatwiejsze.



Zapewne wielu „komputerowców“ myślało już kiedyś o wykorzystaniu starego, niepotrzebnego CDROM-a jako odtwarzacza audio. O ile właściciele napędów z dodatkowym przyciskiem *Play* nie mają problemów z adaptacją tego rodzaju, to w zwykłych napędach nie jest to już takie proste. W typowym CDROM-ie z interfejsem ATA komunikacja z otoczeniem odbywa się z zastosowaniem protokołu ATAPI.

Standard ATA ujrzał światło dzienne wraz z powstaniem komputerów klasy AT (stąd jego nazwa - *AT Attachment*). Dziś powstało wiele jego odmian wykorzystujących m.in. transfery DMA, rozszerzoną obsługę dostępu równoległego PIO itp. Ciągłe zwiększana jest szybkość przesyłu danych pomiędzy HDD a kompute-

rem - osiąga już 266 MB/s. Raczej nie będziemy korzystali w naszych projektach z takich prędkości, więc DMA zostawimy w spokoju. W opisach często będę pomijał nieistotne dla działania układu sterownika flagi, komendy itp., aby nie przytłoczyć Cię, drogi Czytelniku, nadmiarem informacji, z których znaczna część jest mało przydatna konstruktorowi. Ponadto, gdy będą wątpliwości, zawsze można skorzystać z łatwo dostępnej dokumentacji.

Część sprzętowa

Zgodnie z normą interfejsu można podłączyć maksymalnie 2 urządzenia do jednej magistrali - *master* i *slave*. Magistrala danych w ATA ma 16 bitów, co zapewne zmartwi wielu amatorów mikrokontrolerów 8-bitowych. Niestety, nie da się w żaden sposób zmusić napędu do pracy w trybie 8-bitowym. Tym samym nie możemy „wpiąć“ napędu do przestrzeni adresowej procesora 51 czy AVR. Rozkład sygnałów w magistrali ATA przedstawiono w **tab. 1**.

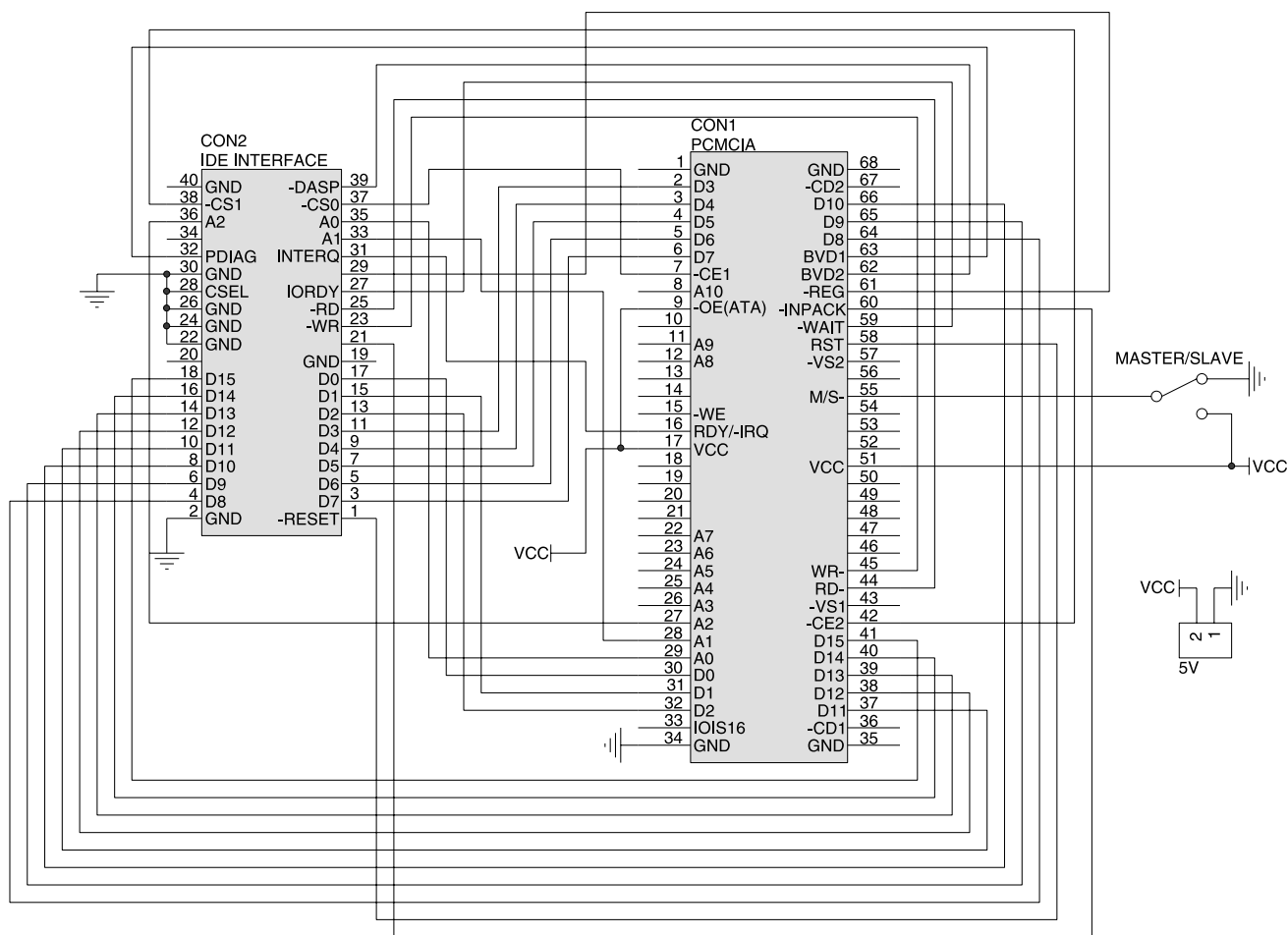
Możliwości i ograniczenia sterownika opisanego w artykule

Możliwości:

- rozpoczęcie/zatrzymanie odtwarzania,
- pauza,
- zmiana aktualnej ścieżki,
- przewijanie (zmienny skok od 3 do 30 sekund),
- wyświetlanie aktualnej pozycji/długości ścieżki,
- wyświetlanie numeru ścieżki/liczby ścieżek,
- wyświetlanie paska postępu,
- możliwość włączenia/wyłączenia powtarzania całości (w przyszłości zmiana trybu „jedna ścieżka”/„całość”),
- wyświetlanie nazwy podłączonego napędu,

Ograniczenia:

- brak obsługi płyt mieszanych (dane + audio),
- brak obsługi płyt wielosecyjnych,
- brak obsługi wszystkich kodów błędów.



Rys. 1. Schemat połączeń (przejściówki) między ATA a PCMCIA

Do podstawowej komunikacji z napędem wystarczą linie danych, adresowe, impulsy zapisu/odczytu oraz wybór bazy rejestrów. Należy pamiętać, że linia D7 jest wewnętrznie „ściągnięta” (pull down) do masy. Służy to wykryciu podłączenia napędu (wszystkie pozostałe linie są podciągnięte do +). Jak się w praktyce okazuje, dość często „ściągnięcie” działa zbyt skutecznie i trzeba zastosować wyrównujące *pull-upa*. W zależności od aplikacji, przydatne może być żądanie przerwania oraz wskaźnik aktywności !DASP. Zazwyczaj w kieszeniach napędów do tej linii jest podłączona dioda LED. Gdy napęd jest ustawiony w trybie *slave*, informuje za pomocą tego sygnału napęd *master* o swojej obecności. Wybór napędu jest całkowicie programowy, więc nie musimy przejmować się stanem tej linii. Nie trzeba także troszczyć się o zerowanie (napęd wykonuje go automatycznie po uruchomieniu). Oczywiście, możemy wykonać

„ręczne” zerowanie, ale zazwyczaj nie jest ono potrzebne.

Bardzo ważne jest, na co radzę zwrócić uwagę, zaznaczenie negacji sygnałów „!”. Przez przeoczenie tego drobiazgu miałem duże problemy z uruchomieniem sterownika.

Zgodnie z normą maksymalna długość kabla może wynosić 46 cm. Przy niezbyt dużych prędkościach przesyłania danych można tę wartość nieco „rozciągnąć”, ale nie zalecam takiego postępowania.

Każdy napęd ATA posiada 8 rejestrów podstawowych (wybierany sygnałem !CS0) oraz rejestr kontrolny (wybierany sygnałem !CS1). W moim projekcie nie używam drugiej bazy rejestrów, gdyż jedyne, co możemy za jej pomocą zrobić, to włączyć/wyłączyć zewnętrzne przzerwiania oraz programowo wyzerować napęd (nie dotyczy to CDROM-ów). Napędy ATAPI posiadają także rejestry do kontroli stacji dysków (np. napędów LS120 lub streamerów).

To w zasadzie wszystkie informacje, jakie są nam potrzebne do „rozmowy” z CDROM-em lub dyskiem twardym. Znacznie bardziej skomplikowana jest jego programowa obsługa. Na koniec tego wstępu chciałem przedstawić jeszcze małą ciekawostkę.

Nie wszyscy pewnie wiedzą, że karty *ATA Flash* (w formie np. kart PCMCIA lub jej wersji mini - *Compact Flash*) są całkowicie zgodne pod względem interfejsu z klasycznym ATA. Na **rys. 1** pokazano schemat połączeń przejściówki między zwykłym ATA a kartą PCMCIA. Do wyboru trybu pracy jako ATA służy wejście !OE. Karty *ATA Flash* posiadają także wyprowadzenie służące do wyboru między pracą jako *master* oraz *slave*. Dobrym zastosowaniem dla takiej karty, udającej HDD, może być np. praca w sterowniku zrobionym na komputerze PC. Taki dysk jest bezgłośny i znacznie mniej awaryjny niż klasyczne, mechaniczne dyski twarde. Przejściówka ta nie zоста-

ła przeze mnie sprawdzona w praktyce, więc nie gwarantuję jej poprawnej pracy.

Opis układu

Schemat elektryczny sterownika pokazano na **rys. 2**. „Sercem“ układu jest procesor AT89S8252. Jest to mikrokontroler całkowicie zgodny z '51. Posiada m.in.:

- 8 kB pamięci Flash,
- 2 kB pamięci EEPROM,
- możliwość programowania w systemie (ISP),
- sprzętowy interfejs SPI.

Główną przyczyną wyboru tego właśnie procesora jest możliwość programowania go w systemie. Programator składa się z dwóch wtyczek i 5 przewodów, może go więc wykonać każdy i zaprogramować procesor bez dostępu do specjalistycznych urządzeń. Oczywiście można zastosować w układzie dowolny procesor z rodziny '51 np. 89C55.

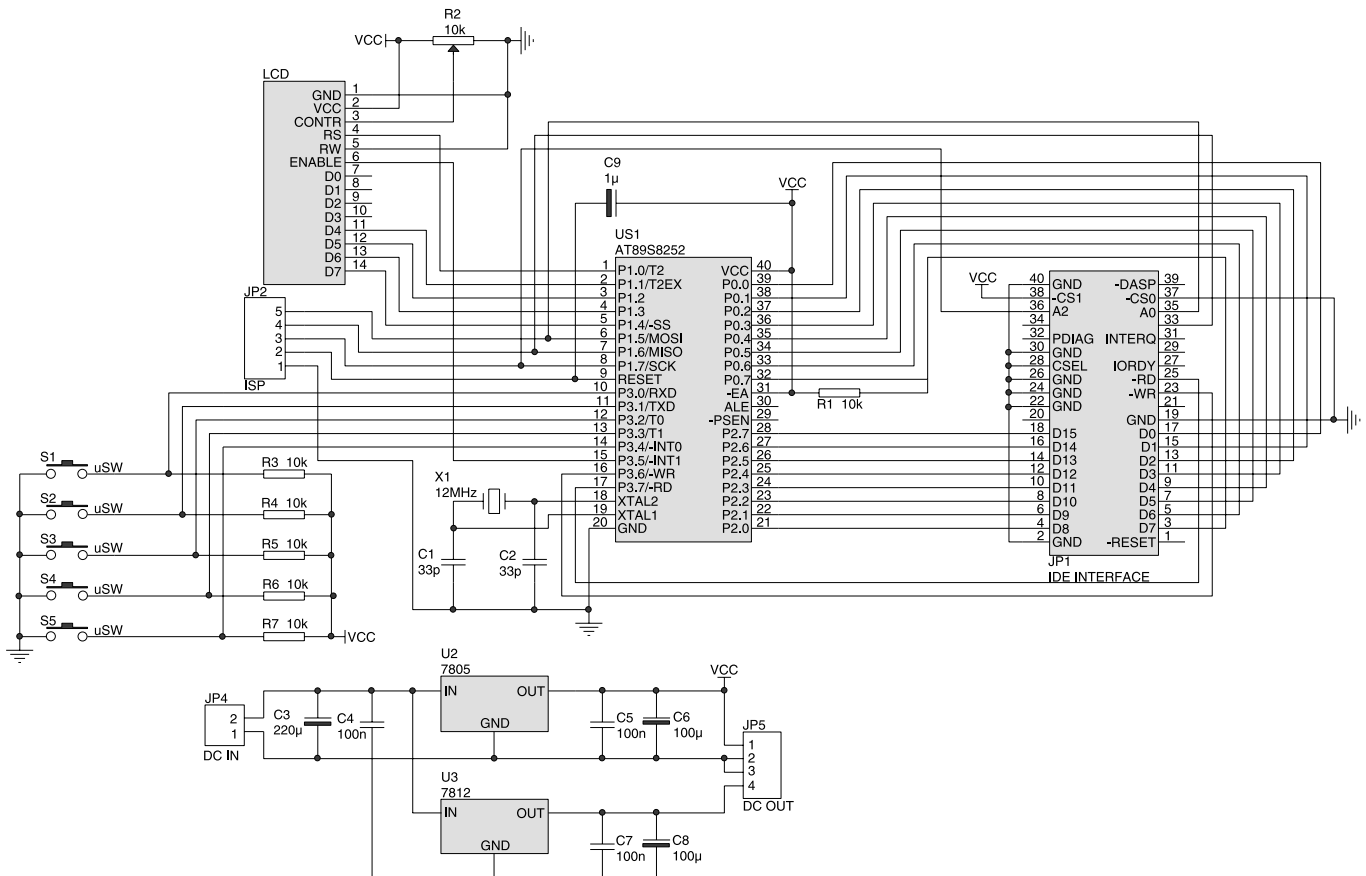
Mikrokontroler sterownika pracuje w typowej aplikacji z zegarem kwarcowym 12 MHz. Zerowanie po włączeniu zasilania zapewnia kondensator 1 μ F

włączony między VCC a wejście zerujące mikrokontrolera. Wejście EA podłączone do VCC wymusza pracę procesora z wewnętrzną pamięcią programu. Port 1 zajmują linie adresowe interfejsu ATA oraz linie sterujące LCD. Wejście Enable LCD zostało podłączone do wyprowadzenia 5 portu 3, aby zapobiec uaktywnianiu LCD podczas zapisywania adresu. Okazuje się, że podczas pracy konieczne jest odłączenie programatora. Wyprowadzenia P3.6 oraz P3.7 służą do wysyłania sygnałów sterujących zapisem oraz odczytem. Porty 0 oraz 1 są całkowicie wykorzystane przez szynę danych. Rezystor R1 „podciąga“ linię D7 do VCC (przyczyny wyjaśniłem wcześniej). Linie P3.0...P3.4 obsługują klawiaturę zbudowaną na mikroprzełącznikach. Wejścia procesora są na wszelki wypadek podciągnięte do „+“, aby uniknąć zakłóceń (procesor posiada wbudowane *pull-upy*, ale jak wynika z praktyki nie są one zawsze skuteczne). W razie potrzeby można

podłączyć w miejsce dwóch przycisków interfejs RS232 dla łatwiejszej komunikacji z urządzeniem. Potencjometr służy do regulacji kontrastu LCD. Układ posiada wbudowany zasilacz dla napędu CD-ROM, zbudowany na dwóch typowych stabilizatorach - 7805 i 7812. Wbrew temu, co jest napisane na tabliczce znamionowej napędu, przy pracy z prędkością 1x (odtwarzanie audio) pobór prądu nie przekracza 0,5A (5V) i 250mA (12V).

Program sterujący napisałem w BASCOM-ie. Posiada on modułową strukturę, więc można go bez problemu zaadaptować do innych urządzeń, modyfikując tylko procedury RD oraz WR. Listing ilustrujący sposób skonfigurowania mikrokontrolera i wykorzystanie procedur zapisu i odczytu przedstawiono poniżej:

```
Io_rd Alias P3.7
Io_wr Alias P3.6
Io_hi Alias P2
Io_lo Alias P0
Io_adr Alias P1
```



Rys. 2. Schemat elektryczny sterownika

Tab. 1.

Numer wyprow.	Opis		PIN
1	!Reset	Masa	2
3	D7	D8	4
5	D6	D9	6
7	D5	D10	8
9	D4	D11	10
11	D3	D12	12
13	D2	D13	14
15	D1	D14	16
17	D0	D15	18
19	Gnd	(blokada)	20
21	!DMARQ	Gnd	22
23	!WR	Gnd	24
25	!RD	Gnd	26
27	!ORDY	CSEL	28
29	!DMACK	Gnd	30
31	INTRQ		32
33	A1	!PDIAG	34
35	A0	A2	36
37	!CS0	!CS1	38
39	!DASP	Gnd	40

- D0...D15: linie danych,
- A0...A2: linie adresowe,
- !CS0 i !CS1: wybór bazy rejestrów,
- !DASP: wskazuje na aktywność, służy także do wykrywania obecności drugiego napędu (slave),
- !INTRQ: sygnał przerwania,
- !ORDY: informacja, czy napęd jest gotowy do rozpoczęcia operacji zapisu/ odczytu,
- !RD i !WR: sygnały odczytu oraz zapisu słowa danych,
- !DMACK i !DMARQ: sygnały do kontroli transferów DMA - przez nas nie będą wykorzystywane,
- !PDIAG: sygnał informujący o zakończeniu testów diagnostycznych,
- CSEL: obecnie prawie niewykorzystywane - wybór napędu master/slave zewnętrznym sygnałem.

```
Sub Rd(adr As Byte)
    ' ustawienie adresu
    Shift Adr, Left, 5
    Io_adr = Adr
    ' impuls rd
    Io_rd = 0
    ' odczyt danych
    Lo = Io_lo
    Hi = Io_hi
    Io_rd = 1
End Sub
```

```
Sub Wr(adr As Byte, Datlo As Byte, Dathi As Byte)
    ' ustawienie adresu
    Shift Adr, Left, 5
    ' wystawienie danych
    Io_adr = Adr
    Io_hi = Dathi
```

```
Io_lo = Datlo
' impuls wr
Io_wr = 0
Nop
nop
Nop
Io_wr = 1
End Sub
```

Program korzysta tylko z tych procedur podczas komunikacji, więc można je dowolnie dostosować. Wszystkie numery wyprowadzeń posiadają przypisane aliasy na początku, więc nie musimy zmieniać nazw w całym kodzie w przypadku jego modyfikacji. Wszystkie komendy oraz numery rejestrów mają swoje nazwy z przypisanymi na stałe wartościami. Dzięki temu w programie wszystko ma swoją definicję i nie ma liczb o niewiadomym znaczeniu.

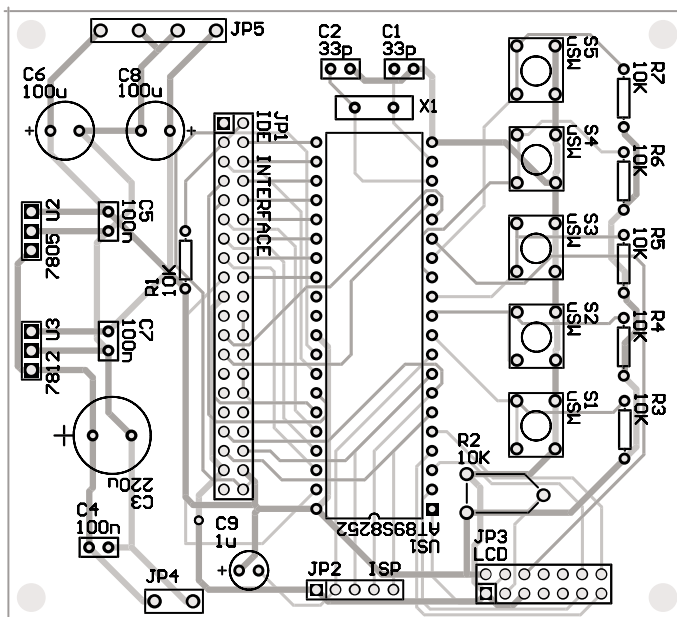
To tyle na temat struktury programu - plik źródłowy jest dostępny na stronie www.ep.com.pl w dziale *Download>Dokumentacje* oraz na płycie CD-EP8/2002B.

Obsługa sterownika jest bardzo łatwa i intuicyjna. Mamy do dyspozycji 5 klawiszy: *Prev*, *Pause*, *Next*, *Stop*, *Config*. Jeśli po uruchomieniu w napędzie znajduje się płyta, automatycznie rozpocznie się odtwarzanie. Za pomocą klawiszy *Prev* oraz *Next* możemy zmieniać ścieżkę. Mają one także inną funkcję - jeśli

przytrzymamy klawisz dłużej niż ok. 250 ms, to rozpoczniemy przewijanie. Klawisz *Stop* oczywiście zatrzymuje odtwarzanie. Ponadto, w przeciwieństwie do klawisza *Pause*, zeruje licznik ścieżek. Jeśli naciśniemy *Stop* ponownie, napęd wysunie tackę z płyty. Za pomocą klawisza *Config* możemy wejść do menu opcji. Na razie dostępne są tylko dwie opcje: powtarzanie oraz zmiana skoku przewijania.

Działanie sterownika

Na początku sterownik dokonuje programowego zerowania dołączonego napędu. Następnie odczytuje sygnaturę, wyświetla ją i czeka na gotowość napędu (sprawdza stan flag ASC i ASCQ). Po wykonaniu inicjalizacji przechodzi do głównego programu. Podczas pierwszej pętli jest sprawdzane, czy jest włożona płyta. Jeżeli nie, to wyświetlany jest odpowiedni komunikat. Po włożeniu CD sterownik blokuje możliwość wysunięcia tacki, odczytuje pozycję końca płyty, liczbę zapisanych ścieżek, sprawdza, czy włożona płyta zawiera tylko dane audio i rozpoczyna odtwarzanie. Następnie przechodzi do kolejnej pętli. Najpierw zostaje sprawdzona aktualna pozycja oraz status odtwarzania. W następnym kroku następuje odczyt klawiszy. W przypadku naciśnięcia *Prev* lub *Next* jest sprawdzane, po jakim



Rys. 3. Rozmieszczenie elementów na płycie drukowanej

czasie puszczono klawisz. W zależności od wyniku, program zmienia numer ścieżki, odczytuje jej pozycję i wydaje komendę *play*. Gdy przytrzymano klawisz odpowiednio długo, dodaje do aktualnej pozycji odpowiednią wartość, rozpoczyna odtwarzanie od nowej pozycji i po ok. 500 ms sprawdza ponownie stan klawisza. Gdy jest on nadal wciśnięty, to przechodzi do początku procedury przewijania. Aktualny numer ścieżki jest pobrany za pomocą komendy *Read Subchannel*. Z tego samego źródła jest określana aktualna pozycja. Jednak nie korzystam z wartości względnej do początku ścieżki, tylko wyliczam ją, odejmując od aktualnej pozycji pozycję początku ścieżki. Jak wynika z moich doświadczeń, jest to konieczne, bo różne napędy bardzo różnie wyliczają sobie tę wartość i występują duże problemy podczas sterowania. Pod koniec sprawdzane jest, czy odtwarzanie zakończyło się - jeśli tak, to program przechodzi do pętli *Stop*. Możliwe jest teraz wysunięcie płyty lub ponowne rozpoczęcie odtwarzania. Informacje o statusie odtwarzania można uzyskać jednocześnie z pozycją dzięki komendzie *Read Subchannel*. W programie używam komendy *Play MSF*. Nie jest to zbyt wygodne, ponieważ musimy przeliczyć adres z postaci LBA na MSF. Wynika to z faktu, że komenda *Play* (45h) umożliwia odtworzenie maksymalnie 65536 sektorów (ok. 14 minut). Z kolei komenda *Play CD* (BCh) nie działała z żadnym z testowanych napędów. Poniżej przedstawiono procedurę przeliczania LBA na MSF:

```
F = LBA Mod 75
LBA = LBA - F
LBA = LBA/75
S = LBA Mod 60
LBA = LBA - S
M = LBA/60
```

Komenda MOD zwraca resztę z dzielenia dwóch liczb. Wartość LBA zwrócona np. przez komendę *Read TOC* składa się z 4 bajtów. Potrzebna jest jedna liczba

całkowita. Aby uniknąć mnożenia bajtów przez dość spore liczby, zastosowałem pewną sztuczkę. Bascom po kompilacji zwraca listę, na której są adresy wszystkich zmiennych w wewnętrznej pamięci. Odczytane bajty zapisuje za pomocą komendy bascoma *Poke <adres>, <dana>* bezpośrednio do pamięci. Dalej kompilator musi się martwić, aby te 4 bajty były widziane w programie w postaci liczby całkowitej. Podczas wyświetlania informacji na LCD potrzebne jest sformatowanie wszystkich liczb w taki sposób, aby zawsze miały format dwucyfrowy (czyli w pustych miejscach widniały zera). Gdybyśmy o to nie zadbali, to długość zapisu na LCD ciągle by się zmieniała. Aby wyeliminować ten problem, zastosowałem kolejną sztuczkę programową:

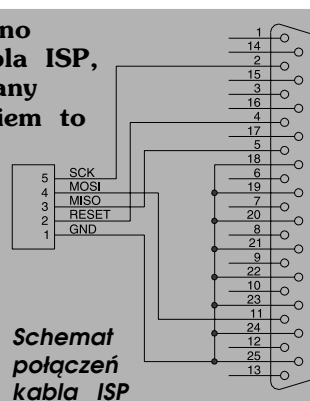
```
Templcd = Makebcd(X)
Shift Templcd, Right, 4
Lcd Templcd

Templcd = Makebcd(X)
Templcd = Templcd And &B00001111
Lcd Templcd
```

Liczba (x) zostaje przekształcona do formatu BCD, czyli pierwsze 4 bity zawierają jednostki,

Na rysunku przedstawiono schemat elektryczny kabla ISP, który może być stosowany m.in. z Bascomem, bowiem to w języku Basic powstał cały program sterujący pracą napędu. Bascom posiada wbudowane oprogramowanie do tego interfejsu pod nazwą *Simple electronic ISP programmer*.

Rozwiązanie to jest bardzo użyteczne przy dopracowywaniu oprogramowania - pewnie wielu Czytelników zna "ból" ciągłego przekładania układu do programatora.



Schemat połączeń kabla ISP

a kolejne dziesiątki. Po odpowiedniej konwersji zawsze będziemy mieli zapis dwucyfrowy. Program został przygotowany do współpracy z LCD 2x40, ponieważ tylko taki miałem. Bardziej praktyczne

WYKAZ ELEMENTÓW

Rezystory

R1, R3...R7: 10kΩ
R2: potencjometr 10kΩ

Kondensatory

C1, C2: 33pF
C3: 220μF/25V
C4, C5, C7: 100nF
C6, C8: 100μF/16V
C9: 1μF/16V

Półprzewodniki

U2: 7805
U3: 7812
US1: AT89S8252

Różne

JP1: IDE INTERFACE
JP2: ISP
JP3: LCD
JP4: DC IN
JP5: DC OUT (złącze 4 pin)
S1...S5: mikroswitch
X1: kwarc 12MHz

jest zastosowanie LCD 4x20, ponieważ ma rozmiary zbliżone do rozmiarów płytki i jest możliwe stosunkowo łatwe wykonanie sterownika z rys. 2. Użytkownik może dowolnie modyfikować sposób wyświetlania informacji. Obsługa

wyświetlacza jest zawarta w procedurze *Show*.

Układ był testowany z napędem Teac CDW54E (nagrywarka 4X, CD 32X), DVD Samsung (32X) oraz AOpen (50X). Sterownik najbardziej niezawodnie działał z tym pierwszym. Nie chcę, aby ktoś mnie posądził o kryptoreklamę, ale moim zdaniem Teac robi najmniej awaryjne i najcichsze napędy.

Montaż i uruchomienie

Sterownik wykonałem w formie niewielkiego modułu z dołączonym wyświetlaczem. Można go zastosować jako sterownik autonomicznego odtwarzacza lub jako część większego urządzenia (komunikacja poprzez RS232, o której wcześniej wspominałem).

Schemat montażowy płytki znajduje się na rys. 3. Montaż

większości elementów nie wymaga komentarza. Mikroprzełączniki najlepiej jest wlutować od strony druku, ponieważ po drugiej stronie większość elementów jest dość wysoka, i w przypadku zabudowy trudno byłoby „wyprowadzić“ je na zewnątrz.

Prototyp wyposażono w stabilizatory przykręcone do niedużego radiatora. Lepiej jest wykorzystać metalową obudowę napędu do odprowadzania ciepła. Wyświetlacz LCD oraz płytką są przymocowane do płyty z pleksiglasu, przytwierdzonej do napędu za pomocą trzech wsporników (pozostałe ot-

wory wykorzystano do przykręcenia 78xx).

Po zaprogramowaniu sterownik powinien działać od razu. Jediną regulacją jest ustawienie odpowiedniego kontrastu na wyświetlaczu. Program domyślnie pracuje z urządzeniem *master*.

Ważne jest przypisanie odpowiednim aliasom odpowiednich klawiszy. Przykładowo, po obróceniu płytki o 180 stopni klawisze *Next* i *Prev* znajdują się po przeciwnych stronach - jest to trochę nienaturalne.

Oprogramowanie (kody źródłowe i gotowy plik HEX do zapisu

w procesorze) jest dostępne w Internecie pod adresem <http://www.mwsoft.prv.pl> oraz na płycie CD-EP8/2002B.

Do wykonania sterownika wykorzystałem dokumentację: SFF-8020i (ATAPI) oraz X3T13 (ATA), które także publikujemy na tej płycie.

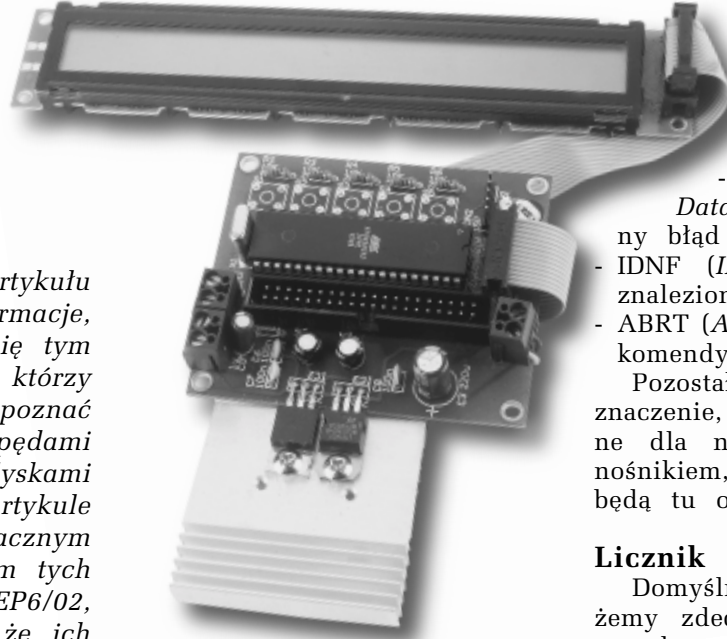
Michał Wysocki
mwsoft@satkabel.com.pl

Wzory płytek drukowanych w formacie PDF są dostępne w Internecie pod adresem: <http://www.ep.com.pl/?pdf/sierpien02.htm> oraz na płycie CD-EP08/2002B w katalogu PCB.

Sterownik napędu CD-ROM, część 2

AVT-5078

W drugiej części artykułu przedstawiamy informacje, które przydadzą się tym spośród Czytelników, którzy zamierzają dokładnie poznać sposób sterowania napędami CD-ROM oraz dyskami twardymi. Zawarte w artykule informacje są w znacznym stopniu powtórzeniem tych opublikowanych w EP6/02, ale uważamy, że ich „nadmiar“ z pewnością nie zaszkodzi.



Oprogramowanie interfejsu ATA

Jak wspomniałem w pierwszej części artykułu, do sterowania dyskiem twardym wykorzystuje się 9 rejestrów, których wykaz zawarto w tab. 2.

Rejestr danych

Jest to dwukierunkowy rejestr służący do wysyłania i odbierania danych. Jest uniwersalny, więc jego zastosowanie ściśle zależy od operacji, jaką wykonujemy.

Rejestr błędów

7	6	5	4	3	2	1	0
-	UNC	-	IDNF	-	ABRT	-	-

Tab. 2.				
!CS0	!CS1	Adres	Odczyt	Zapis
Rejestr kontrolny				
0	1	6	Alternatywny rejestr statusu	Rejestr kontroli napędu
Rejestry komend i danych				
1	0	0	Rejestr danych	Rejestr danych
1	0	1	Rejestr błędów	Rejestr dodatkowych opcji
1	0	2	Licznik sektorów	Licznik sektorów
1	0	3	Numer sektora/LBA (7:0)	Numer sektora/LBA (7:0)
1	0	4	Numer cylindra (7:0)/LBA(15:8)	Numer cylindra (7:0)/LBA(15:8)
1	0	5	Numer cylindra (15:8)/LBA(23:16)	Numer cylindra (15:8)/LBA(23:16)
1	0	6	Numer głowicy/wyбір napędu/LBA(27:24)	Numer głowicy/wyбір napędu/LBA(27:24)
1	0	7	Rejestr statusu	Rejestr komend

Liczby w nawiasach oznaczają zakresy bitów.

Znaczenie bitów:
 - UNC: (*Uncorrectable Data Error*) - nienaprawialny błąd danych,
 - IDNF (*ID Not Found*) - nie znaleziono sektora,
 - ABRT (*Aborted*) - nie wykonano komendy.

Pozostałe bity także posiadają znaczenie, ale są charakterystyczne dla napędów z wymiennym nośnikiem, w związku z czym nie będą tu omawiane.

Licznik sektorów

Domyślnie za jego pomocą możemy zdecydować, ile sektorów napęd ma odczytać lub zapisać. Niektóre komendy wykorzystują go także do innych celów.

Numer sektora

Zależnie od trybu adresowania służy do ustawienia numeru sektora, jaki chcemy odczytać lub pierwszych 8 bajtów adresu LBA.

Numer cylindra Lo oraz Hi

To samo co przy sektorach, ale określamy numer cylindra lub następne 16 bitów adresu LBA.

Numer głowicy

7	6	5	4	3	2	1	0
1	LBA	1	DEV	Numer głowicy/LBA (27:23)			

Bity oznaczone „1“ powinny być zawsze ustawione na logiczną „1“. Za pomocą bitu LBA wybieramy sposób adresowania, o którym później.

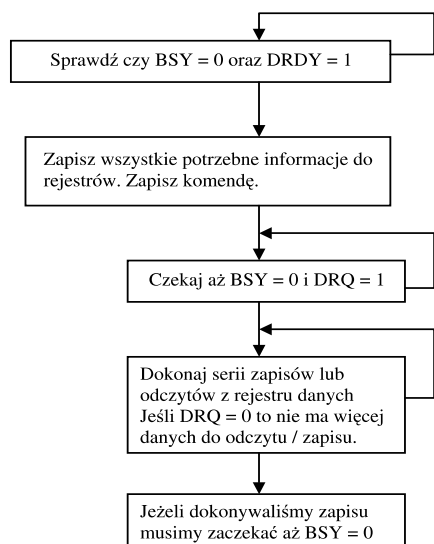
Bit DEV służy do wyboru napędu:

0 = MASTER

1 = SLAVE

Po zapisie bitu urządzenie wybrane reaguje na wszystkie komendy i operacje IO, a drugie staje się „niewidzialne“ dopóki go nie uaktywnimy.

Młodsze 4 bity zawierają numer głowicy lub pozostałe 4 bity adresu LBA.



Rys. 4. Algorytm dostępu do rejestrów kontrolera napędu

Rejestr poleceń

Jest to rejestr tylko do zapisu. Po wpisaniu do niego kodu komendy kontroler natychmiast przystępuje do jej wykonywania.

Rejestr statusu

- | | | | | | | | |
|-----|------|---|-----|-----|---|---|-----|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BSY | DRDY | - | DSC | DRQ | - | - | ERR |
- BSY: jak sama nazwa wskazuje, sygnalizuje on zajętość kontrolera. Napęd ustawia tę flagę gdy wykonuje komendę lub po zerowaniu, aby zasignalizować brak gotowości.
 - DRDY: wskazuje na gotowość napędu.
 - DSC: wskazuje, że głowica znajduje się nad żadaną ścieżką.
 - DRQ: flaga informuje, że napęd jest gotowy do rozpoczęcia zapisu lub odczytu danych.
 - ERR: wskazuje na błąd i w takich wypadkach wypadałoby odczytać rejestr błędów, aby dowiedzieć się o szczegółach.

Rejestr kontroli

- | | | | | | | | |
|---|---|---|---|---|------|--------|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | - | - | - | - | SRST | -INTEN | - |
- SRST: programowe zerowanie kontrolera.
 - -INTEN: służy do włączenia/wyłączenia generacji przerwania.

Przykładowy algorytm wykonania komendy przedstawiono na rys. 4. Przesyłanie danych w trybie PIO odbywa się poprzez kolejne odczyty lub zapisy rejestru danych. Zawsze pierwszy bajt znajduje się w młodszej części odczytanego słowa.

Adresowanie

Obecnie stosuje się dwa sposoby adresowania sektorów na dysku twardym. W obu przypadkach rozmiar sektora wynosi 512 bajtów.

CHS

System ten jest domyślny we wszystkich napędach i kompatybilny wstecz. Skrót pochodzi od *Cylinder Head Sector*. Aby zaadresować sektor w tym systemie, musimy podać numer sektora, numer cylindra oraz numer głowicy. Liczba sektorów na ścieżkę, liczba głowic (ścieżek) na cylinder oraz liczba cylindrów są różne dla różnych napędów. Dla sektorów zakres ten wynosi 1...255, dla głowic 0...15, a dla cylindrów 0...65535. Za pomocą następującego wzoru możemy obliczyć adres LBA:

$$LBA = (((cylinder * liczba głowic) + głowica) * liczba sektorów na głowicę) + sektor - 1$$

Potrzebne do tego informacje można zdobyć za pomocą polecenia *Identify Device*, którą omówimy później. Taki system jest raczej niewygodny, więc powstał nowy system adresowania:

LBA

Jest to akronim od *Logical Block Addressing*. System jest niesamowicie prosty - do zaadresowania sektora używa się jednej 28-bitowej liczby. Policzymy: $2^{28} = 268435456$, $268435456 * 512$ (bajtów w sektorze) = 137438953472 , co daje: $137438953472/1024/1024 = 131072$ MB, czyli możemy zaadresować 130GB danych - wydaje się to dużo, ale biorąc pod uwagę to, że dziś produkuje się już dyski o takiej pojemności, producenci będą musieli niedługo wymyślić coś nowego.

Polecenia

Kontrolery dysków twardych mają ich sporo, ale my zajmiemy się tylko czterema najważniejszymi. Reszta poleceń to głównie odmiany tych najważniejszych, komendy do obsługi haseł dostępu, trybów zasilania itp.

Standby

Rejestr	7	6	5	4	3	2	1	0
1	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-
6	1	-	1	DEV	-	-	-	-
7	94h	lub	E0h	-	-	-	-	-

Polecenie to służy do wprowadzenia napędu w tryb *standby*, czyli obniżonego poboru mocy, który jest uzyskiwany poprzez zatrzymanie talerzy i uśpienie napędu aż do czasu próby dostępu do danych.

Read sector

Rejestr	7	6	5	4	3	2	1	0
1	-	-	-	-	-	-	-	-
2	liczba sektorów do odczytania	-	-	-	-	-	-	-
3	numer sektora lub LBA	-	-	-	-	-	-	-
4	numer cylindra lub LBA	-	-	-	-	-	-	-
5	numer cylindra lub LBA	-	-	-	-	-	-	-
6	1 LBA 1 DEV numer głowicy lub LBA	-	-	-	-	-	-	-
7	21h lub 20h (z powtórzeniami)	-	-	-	-	-	-	-

Jest to najbardziej interesujące dla nas polecenie. Pozwala na odczyt maksymalnie 256 sektorów. Jeśli wartość rejestru 2 wynosi zero, wtedy także dostaniemy 256 sektorów. Rejestry 3...6 zawierają adres początkowego sektora. Jeśli bit LBA = 1, wtedy stosujemy adresowanie LBA. Gdy LBA = 0 - CHS. Po wydaniu komendy trzeba poczekać aż kontroler napędu wyzeruje flagę BSY i ustawi DRQ. Komenda posiada dwie odmiany - wersja z powtórzeniami nakazuje napędowi w przypadku natrafienia na uszkodzony sektor próbować odczytać go ponownie. Liczba prób jest nieokreślona i specyficzna dla różnych napędów.

Write sector

Rejestr	7	6	5	4	3	2	1	0
1	-	-	-	-	-	-	-	-
2	liczba sektorów do zapisania	-	-	-	-	-	-	-
3	numer sektora lub LBA	-	-	-	-	-	-	-
4	numer cylindra lub LBA	-	-	-	-	-	-	-
5	numer cylindra lub LBA	-	-	-	-	-	-	-
6	1 LBA 1 DEV numer głowicy lub LBA	-	-	-	-	-	-	-
7	31h lub 30h (z powtórzeniami)	-	-	-	-	-	-	-

Tab. 3.

Nr słowa	Opis
0	bit 15 0 = ATA, 1 = ATAPI
1	Liczba cylindrów
3	Liczba głowic
6	Liczba sektorów na ścieżkę (głowicę)
10...19	Numer seryjny (20 znaków ASCII)
23...26	Wersja oprogramowania (8 znaków ASCII)
27...46	Nazwa urządzenia (40 znaków ASCII)
57...58	Pojemność wyrażona w sektorach
60...61	Całkowita liczba możliwych do zaadresowania sektorów (tryb LBA)

Zasada działania jest identyczna jak w przypadku *Read Sector*, ale oczywiście dane zapisujemy. Po przesłaniu wszystkich danych do napędu musimy poczekać aż kontroler napędu wyzeruje flagę BSY.

Identify Device

Rejestr	7	6	5	4	3	2	1	0
1	-							
2	-							
3	-							

4	-							
5	-							
6	1	-	1	DEV	-			
7	Device							

Polecenie służy do odczytu informacji o napędzie. Po jej wydaniu otrzymujemy 256 słów, z których większość nie jest wykorzystywana. W **tab. 3** znajduje się zestawienie najważniejszych i najbardziej użytecznych.

To są w zasadzie wszystkie informacje, jakich potrzebujemy, aby operować na danych z dysku twardego. Za miesiąc zajmiemy się omówieniem interfejsu ATAPI.

Michał Wysocki
mwssoft@satkabel.com.pl

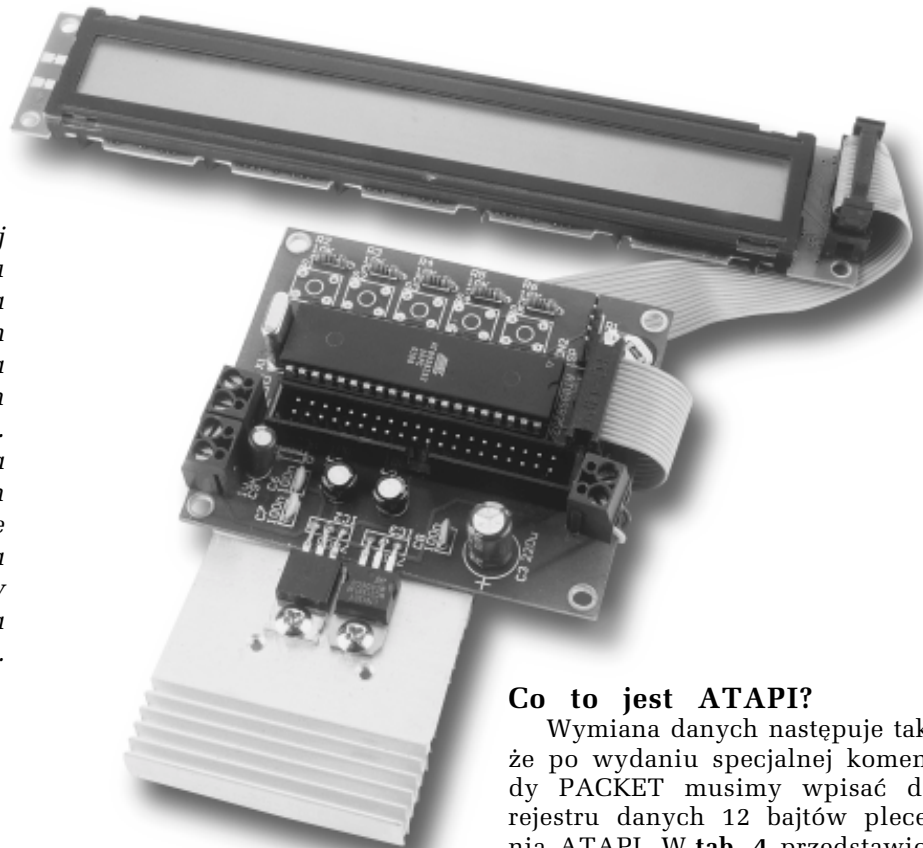
Wzory płytek drukowanych w formacie PDF są dostępne w Internecie pod adresem: <http://www.ep.com.pl/?pdf/wrzesien02.htm>.

Sterownik napędu CD-ROM, część 3

AVT-5078

W trzeciej - przedostatniej - części artykułu przedstawiamy zagadnienia związane z protokołem sterowania pracą kontrolera w napędach CD-ROM.

Rekomendacje: prezentacja o niebagatelnych walorach praktycznych - na podstawie tego artykułu można zaprojektować własny sterownik HDD/CD-ROM na mikrokontrolerze AVR.



Zazwyczaj mówi się, że sterownik lub napęd wyposażono w interfejs ATAPI. Jest to jednak uproszczenie. ATAPI jest bowiem protokołem wymiany danych pomiędzy sterownikiem zintegrowanym na płycie komputera a sterownikiem wbudowanym w napęd. Z elektrycznego punktu widzenia, sterowniki ATAPI i ATA są identyczne.

Co to jest ATAPI?

Wymiana danych następuje tak, że po wydaniu specjalnej komendy PACKET musimy wpisać do rejestru danych 12 bajtów plecienia ATAPI. W **tab. 4** przedstawiono funkcje rejestrów w urządzeniu z ATAPI.

Jak widać część rejestrów nie zmieniła swoich funkcji, ale całkowicie identyczne z ATA są tylko: rejestr kontroli napędu, rejestr danych oraz rejestr wyboru napędu (bity wyboru głowicy i LBA są zastrzeżone).

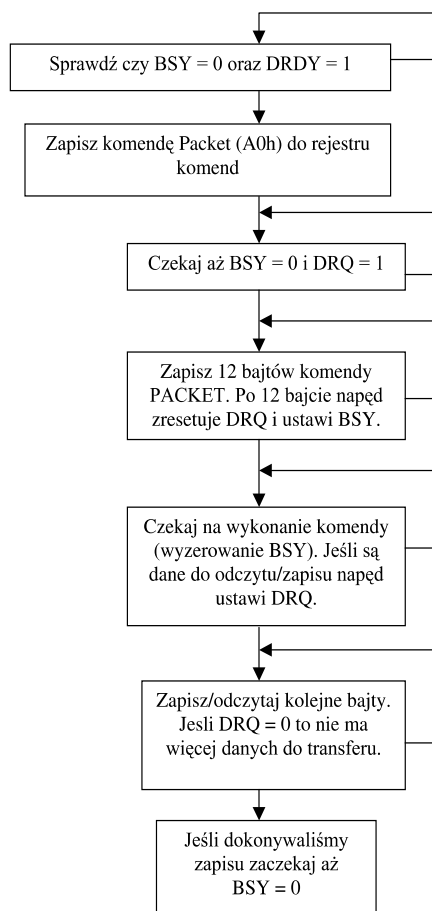
Rejestr błędów

7	6	5	4	3	2	1	0
Sense key				-	ABRT	EOM	-

W standardzie ATAPI jest bardzo rozbudowany system obsługi błędów. Kody błędów pogrupowano w 3 zestawy - *Sense Key*, *Additional Sense Code* oraz *Additional Sense Code Qualifier*. Dwa ostatnie są dostępne poprzez komendę *Request Sense*. Kod błędu *Sense Key* znajduje się w 4 starszych bitach rejestru błędów. W **tab. 5** zawarto opis kodów *Sense Key*.

Tab. 4. Funkcje rejestrów ATAPI

!CS0	!CS1	Adres	Odczyt	Zapis
Rejestr kontrolny				
0	1	6	Alternatywny rejestr statusu	Rejestr kontroli napędu
Rejestry komend i danych				
1	0	0	Rejestr danych	Rejestr danych
1	0	1	Rejestr błędów	-
1	0	2	Rejestr kontroli przesyłu danych	-
1	0	3	-	-
1	0	4	Licznik bajtów (7:0)	Licznik bajtów (7:0)
1	0	5	Licznik bajtów (15:8)	Licznik bajtów (15:8)
1	0	6	Wybór aktywnego napędu	Wybór aktywnego napędu
1	0	7	Rejestr statusu	Rejestr komend



Rys. 5. Przykładowy algorytm wykonywania komendy

Każda komenda ma swój zestaw rekomendowanych kodów błędów: SK, ASC oraz ASCQ. W przypadku ostrzeżenia (6h) komenda także nie zostaje wykonana:

- ABRT: znaczenie identyczne jak w ATA - komenda nie wykonana,
- EOM: wykryto koniec nośnika.

Rejestr kontroli przesyłania danych

7	6	5	4	3	2	1	0
-	-	-	-	-	-	IO	CoD

Służy do programowej kontroli wymiany danych. Istotne są w nim dwa najmłodsze bity:

- IO - wskazuje czy transfer danych ma odbyć się do napędu (0) czy z napędu (1)
- CoD - *Command_or_Data* - wskazuje czy napęd oczekuje na bajty komendy (1), czy na przesłanie danych (0).

W rzeczywistości nie korzysta się z tego rejestru - do poprawnej komunikacji wystarczy kontrola flag DRQ oraz BSY.

Licznik bajtów

Rejestr ten zawiera informacje o liczbie bajtów przekazanych z/do napędu. Jeśli nie chcemy określić limitu danych, MUSIMY wpisać do rejestru wartość 0. W innym przypadku napęd może wykorzystać do następnej komendy wartość, jaką sam zapisał przy wykonywaniu poprzedniej (w specyfikacji nie ma informacji na ten temat - sprawdzono to praktycznie).

Rejestr statusu

7	6	5	4	3	2	1	0
BSY	DRDY	-	-	DRQ	CORR	-	ERR

Znaczenie poszczególnych bitów:

- BSY - znaczenie identyczne jak w przypadku ATA,
- DRDY - jw. ale po uruchomieniu urządzenia flaga ta ma wartość 0 - służy do odróżnienia napędu ATA od ATAPI,
- DRQ - żądanie transferu danych,
- CORR - wykryto naprawialny błąd danych,
- ERR - błąd podczas wykonywania komendy.

Komendy ATAPI

Interfejs ATAPI dopuszcza tylko kilka komend wykonywanych w klasycznym trybie ATA. Są to:

ATAPI Soft reset

Komenda jest odpowiednikiem bitu SRST w rejestrze kontrolnym. Napęd CD posiada ten bit w rejestrze kontrolnym, ale nie zaleca się używania go.

Kod komendy: 08h

Identify ATAPI device

Odpowiednik komendy ECh. Większość słów nie jest wykorzystywana. Zwraca informacje o typie urządzenia, możliwych trybach pracy (PIO, DMA itp.) oraz nazwę i wersję firmware'u (informacje te mają taki sam układ jak w ATA). Szerszych informacji może dostarczyć komenda ATAPI Inquiry.

Kod komendy: A1h

Standby

Działanie identyczne jak w ATA.

Kod komendy: E0h

Packet Command

Jest to ta komenda, dzięki której możemy wysłać do napędu

Tab. 5. Kody Sense Key

Kod	Opis
0h	Brak dodatkowych informacji o błędzie/brak błędu
1h	Podczas wykonywania komendy wystąpiły naprawialne błędy danych
2h	Nie gotowy do wykonania tej operacji
3h	Błąd nośnika
4h	Błąd sprzętu
5h	Błąd w komendzie
6h	Ostrzeżenie o zmianie nośnika lub zerowaniu urządzenia
7h	Próba dostępu do danych chronionych
Bh	Komenda anulowana przez napęd

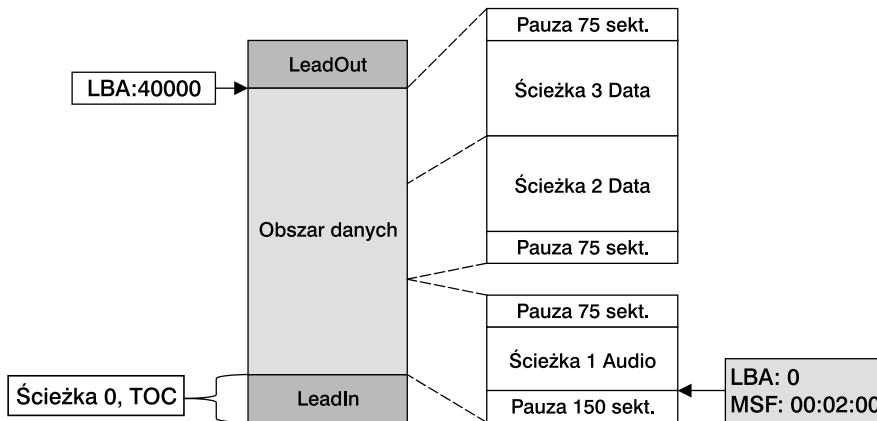
komendę ATAPI. Po jej wydaniu zapisujemy do rejestru danych 12 bajtów (6 słów 2-bajtowych). Oto przykład komendy Play:

Bajt	7	6	5	4	3	2	1	0
0	47h	-	-	-	-	-	-	-
1	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-
3	początkowe M	-	-	-	-	-	-	-
4	początkowe S	-	-	-	-	-	-	-
5	początkowe F	-	-	-	-	-	-	-
6	końcowe M	-	-	-	-	-	-	-
7	końcowe S	-	-	-	-	-	-	-
8	końcowe F	-	-	-	-	-	-	-
9	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-

Jak widać w standardzie ATAPI nie ma ograniczenia liczby parametrów komendy. Zawsze pierwszy bajt zawiera kod komendy. Dalej panuje znaczenie bajtów jest zależne od konkretnej komendy. Zanim jednak przejdę do ich opisu, omówię sposób składowania danych na płytach CD-ROM. Na rys. 5 przedstawiono przykładowy algorytm wykonania komendy. Ważne jest, aby nie wydać kolejnej komendy zanim nie odczytamy/zapiszemy wszystkich bajtów, które wymagała poprzednia.

Organizacja danych na płytach CD

Organizacja danych na płytach nie jest taka sama, jak w przypadku dysków twardych. Standard dopuszcza kilka możliwych typów sektorów. W CD używa się dwóch sposobów adresowania sektorów - LBA oraz MSF. LBA jest jedną 32-bitową liczbą (w rzeczywistość-



Rys. 6. Schemat organizacji zapisu na płycie

ci w CDROM-ach wykorzystuje się tylko 24 bity), która określa numer sektora. Adres MSF składa się z 3 liczb: Minut, Sekund i Ramek (*Frames*). Na jedną sekundę przypada 75 ramek, a na minutę oczywiście 60 sekund. Standard ten pochodzi z czasów, gdy płyty używano wyłącznie do przechowywania dźwięku. Adres fizycznie zapisany w nagłówku każdego sektora ma format MSF, a nie LBA. Dopiero napęd przelicza tę wartość. Większość komend operujących na adresie sektora posiada znacznik, dzięki któremu możemy wybrać jakiego rodzaju adresowania chcemy używać. Na rys. 6 przedstawiono blokową budowę przykładowej płyty.

Wszystkie dane są zawarte pomiędzy znacznikami początku i końca płyty, tzn. *LeadIn* oraz *LeadOut*. Cała płyta jest podzielona na ścieżki. Numer ścieżki musi się zawierać między 1 a 99. Minimalna długość ścieżki to 300 sektorów. *LeadIn* jest ścieżką 0 i zawiera tzw. TOC (*Table of Contents*) - tabelę zawartości. Nie jest ona dostępna z poziomu adresowania sektorów. TOC możemy odczytać za pomocą komendy *Read TOC*. Zawsze pierwsze 150 sektorów jest pauza. W adresie LBA pozycję 0 ma pierwszy sektor po obowiązkowej pauzie. W MSF jest ona uwzględniona, więc adres początkowy to 00:02:00. Zawsze przed zmianą typu ścieżki musi wystąpić pauza. W przypadku ścieżek tego samego typu pauza jest nieobowiązkowa i zazwyczaj jest zawarta w końcówce lub początku ścieżki. W tabeli zawartości nie ma informacji

o pauzach. Każdy sektor posiada kilka subkanałów. Na 24 bajty przypada 1 bajt informacji subkanałowych. Najważniejszy jest kanał Q zawierający informacje o aktualnej pozycji oraz pozycji relatywnej do początku ścieżki. W subkanale P jest flaga informująca o pauzie. Inne raczej nie są bezpośrednio przydatne użytkownikowi.

Najważniejsze formaty sektorów to: *Audio*, *Mode 1* oraz *Mode 2*. Ten ostatni jest rzadko używany. W trybie 1 sektor składa się z:

- 12-bajtowej ramki synchronizacji,
- 4-bajtowego nagłówka zawierającego informacje o typie sektora i pozycji (MSF),
- 2048 bajtów danych,
- oraz 288 bajtów zawierających informacje do korekcji błędów oraz subkanały.

Tryb drugi jest podobny, ale nie zawiera ostatnich 288 bajtów - do dyspozycji mamy teoretycznie 2336 bajtów danych.

Istnieją jeszcze tryby: *Mode 2 Form 1* oraz *Mode 2 Form 2*, o których dostępna dokumentacja milczy. Pierwszy posiada 2048 bajtów, drugi 2324 + 4 wolne. Oba tryby posiadają także 8-bajtowy dodatkowy nagłówek, którego zastosowania nie potrafiłem rozszyfrować. W trybie audio sektor ma 2352 bajty. Kolejność bajtów jest następująca:

- lewy kanał - młodszy bajt,
- lewy kanał - starszy bajt,
- prawy kanał - młodszy bajt,
- prawy kanał - starszy bajt.

W tab. 6 zawarto zestawienie wszystkich komend - zarówno tych narzuconych przez standard jak i opcjonalnych. Nie będę opisywał wszystkich, ponieważ część

Tab. 6. Zestawienie poleceń ATAPI wraz z kodami

Komendy statusu i informacji		
1	TEST UNIT READY	00h
2	REQUEST SENSE	03h
3	INQUIRY	12h
4	MECHANISM STATUS	BDh
5	READ CDROM CAPACITY	25h
6	READ TOC	43h
7	MODE SENSE	5Ah
Komendy Audio		
8	PLAY AUDIO	45h
9	PLAY AUDIO MSF	47h
10	PLAY CD	BC h
11	STOP PLAY	4Eh
12	PAUSE/RESUME	4Bh
Komendy sterujące		
13	MODE SELECT	55h
14	PREVENT/ALLOW MEDIUM REMOVAL	1Eh
15	SEEK	2Bh
16	START/STOP UNIT/EJECT	1Bh
Komendy danych		
17	READ	28h
18	READ CD	BEh
19	READ CD MSF	B9h
	READ HEADER	44h
20	READ SUB-CHANNEL	42h
Komendy opcjonalne		
	LOAD/UNLOAD CD	A6h
	SCAN	BAh
	SET CD SPEED	BBh

z nich jest mało przydatna. Opis pól zarezerwowanych w komendach także zostanie pominięty - powinny zawsze mieć wartość 0.

Należy pamiętać, że bez względu na liczbę bajtów niosących informację, polecenie musi zawsze mieć 12 bajtów. Ustawiony na „1” bit MSF oznacza, że będziemy dokonywać adresowania w systemie MSF. Adres zapisuje się w miejsce adresu LBA z tym, że najstarszy bajt jest w tym wypadku nieużywany. Również zwracane dane będą miały postać MSF. Nie wszystkie komendy podporządkowują się tej regule - niektóre, jak np. *Play MSF*, mają własny układ bajtów. Ich znaczenie i realizowane funkcje przedstawimy w artykule za miesiąc.

Michał Wysocki
mwssoft@satkabel.com.pl

Wzory płytek drukowanych w formacie PDF są dostępne w Internecie pod adresem: <http://www.ep.com.pl/?pdf/pazdziernik02.htm> oraz na płycie CD-EP10/2002B w katalogu PCB.

Sterownik napędu CD-ROM, część 4

AVT-5078

W ostatniej części artykułu przedstawiamy dokładny opis poleceń ATAPI, za pomocą których można sterować napędami CD-ROM.

Rekomendacje: prezentacja o niebagatelnych walorach praktycznych - na podstawie tego artykułu można zaprojektować własny sterownik HDD/CD-ROM na mikrokontrolerze AVR.

Opis poleceń ATAPI

Prezentujemy opis poleceń ATAPI. Należy pamiętać o tym, że są one zawsze 12-bajtowe, a ze względu na oszczędność miejsca niewykorzystane bajty zostały w tablicach pominięte (powinny mieć zawsze wartość 00h).

1. TEST UNIT READY

Komenda bardzo przydatna. Nie wykonuje ona żadnego działania, ale po jej wydaniu SK, ASC i ASCQ przyjmują wartości adekwatne do aktualnego stanu urządzenia.

2. REQUEST SENSE

Komenda zwraca dane dotyczące aktualnego statusu/błędu.

Bajt	Opis
0	03h

4	Maksymalna długość transferu (16)
---	-----------------------------------

Ze zwróconych danych nas interesują tylko pola ASC oraz ASCQ. Napęd zwraca nieokreśloną ilość bajtów więc dobrze jest określić limit.

Bajt	Opis
2	- Sense Key
12	Additional Sense Code
13	Additional Sense Code Qualifier

W **tab. 7** przedstawiono najbardziej użyteczne i najczęściej spotykane kody ASC i ASCQ.

3. INQUIRY

Zwraca informacje o urządzeniu. Nie są to zbyt przydatne dane więc zainteresowanych odsyłam do dokumentacji.

4. MECHANICAL STATUS

Zwraca informacje o aktualnej pozycji mechanizmu, oraz statusu zmieniarkei (rzadko występują CDROM-y obsługujące więcej niż 1 płytę).

Bajt	7	6	5	4	3	2	1	0
0	BDh							

7	MSB	Maksymalna długość transferu	LSB
---	-----	------------------------------	-----

Zwracane dane:

Bajt	7	6	5	4	3	2	1	0	
2	MSB	Aktualna pozycja LBA					LSB		

tywane w normalnym napędzie. Komenda zwraca 8 bajtów.

5. READ CDROM CAPACITY

Pozwala na odczytanie objętości płyty w sektorach. Ten sam efekt można uzyskać za pomocą komendy *Read TOC*. Jest wywoływana bez parametrów.

Bajt	7	6	5	4	3	2	1	0
0	MSB	Całkowita liczba sektorów						LSB
4	MSB	Rozmiar jednego bloku w bajtach						LSB

Obecnie nie używane - wartość powinna zawsze mieć 2048

6. READ TOC

Pozwala na odczyt tabeli adresów ścieżek. Nadaje się także do odczytu informacji o płytach wielosekcyjnych. Biorąc pod uwagę rzadkość występowania takich płyt omówię tylko tradycyjny sposób.

Bajt	7	6	5	4	3	2	1	0
0	43h							
1								MSF
2								Tryb (0!)
6	Początkowy nr ścieżki							
7	MSB	Maksymalna liczba bajtów do wystania						LSB
9	Tryb (0!)							

Numer ścieżki określa od której ścieżki chcemy rozpocząć odczyt tabeli. Zwrócone dane składają się z nagłówka:

Bajt	7	6	5	4	3	2	1	0
0	MSB	Całkowita długość danych liczona od 3 bajtu						LSB

2	Numer pierwszej ścieżki							
3	Numer ostatniej ścieżki							

oraz 8 bajtowych bloków zawierających dane kolejnych ścieżek:

Bajt	7	6	5	4	3	2	1	0
0								
1	ADR							- A/D Copy
2	Numer ścieżki							
4	MSB	Adres pierwszego sektora ścieżki						LSB

Bit A/D wskazuje czy jest to ścieżka audio (0) czy z danymi (1). Bit *Copy* wskazuje na zabronienie kopiowania (0) lub pozwolenie (1). *ADR* wskazuje na dane zakodowane w subkanale Q ñ w tym przypadku mało użyteczne. Każda płyta posiada specjalną ścieżkę oznaczoną numerem AAh. Jej adres jest początkiem *Lead-Outu*, czyli jednocześnie określa przestrzeń adresową na jakiej możemy operować.

Tab. 7. Najczęściej spotykane kody ASC i ASCQ

ASC	ASCQ	Opis
04	01	Przygotowywanie do gotowości mechaniki
05	01	Błąd przy ładowaniu/usuwanie płyty
1A	00	Błędna długość listy parametrów
20	00	Błędny kod komendy
21	00	Adres poza zakresem
24	00	Błąd w którymś z bajtów komendy
26	02	Błędna wartość parametru
28	00	Zmieniono nośnik
29	00	Wykonano zerowanie
30	00	Niekompatybilny nośnik
30	01	Nieznany format nośnika
30	02	Błędny format nośnika
3A	00	Brak nośnika w napędzie
44	00	Wewnętrzny błąd urządzenia
53	02	Dozwolono usunięcie nośnika
57	00	Nie można odczytać TOC
64	00	Błędna komenda dla tej ścieżki

7. MODE SENSE

Bardzo rozbudowana komenda. W połączeniu z *Mode Select* służy do odczytu parametrów urządzenia oraz jego konfiguracji. Posiada 4 strony ustawień, my zajmujemy się tylko jedną: *Capabilities and Mechanical Status*. Jest to strona tylko do odczytu więc nie możemy w niej nic modyfikować za pomocą *Mode Select*. Interesująca może być jeszcze strona *Audio Status*, ale służy ona tylko do wyboru wyjścia sygnału (mało który napęd posiada więcej niż jedno wyjście).

Bajt	7	6	5	4	3	2	1	0
0								5Ah
2	0							Kod strony
7	MSB							LSB
								Maksymalna ilość danych jakie komenda może wysłać
Kod strony	Opis							
01h	Obsługa błędów odczytu							
0Dh	Parametry CDR0M-u							
0Eh	Parametry Audio CDR0M-u							
2Ah	Możliwości + aktualny status CDR0M-u							

Bajty strony wysyłanej przez komendę zawsze są poprzedzone 8-bajtowym nagłówkiem:

Bajt	7	6	5	4	3	2	1	0
0	MSB							LSB
								Całkowita długość danych (bez nagłówka)
2								Rodzaj nośnika

Bajt	7	6	5	4	3	2	1	0
0	-							Kod strony (2Ah)
1								Długość strony (12h)
2								Odczyt Odczyt CDRW CDR
3								Zapis Zapis CDRW CDR
4	-							Multi Mode 2 Mode 2 Cyfrowe Cyfrowe Wyjście CD sesja Form Form wyjście wyjście analogowe AUDIO
6								Typ mechanizmu ładowania - Komenda - Status Komenda Eject blokady blokady
8	MSB							LSB
12	MSB							LSB
14	MSB							LSB

W bajtach 2...4 ustawienie danego bitu równa się obsłudze danej właściwości. Bit CD Audio wskazuje na możliwość odtwarzania audio. W bajcie 6 bity 0 i 3 wskazują na obsługę komend. Trzy najstarsze bity zawierają typ mechanizmu ładowania nośnika:

Kod (bin)	Typ
000	Szczelina do wsuwania płyty
001	Wysuwana tacka (najpopularniejsze)
010	Otwierane wieko (mechanizm stosowany powszechnie w Discmanach)

101 Mechanizm ładowania kartridży (już nie spotykany)

Prędkość jest wyrażona w kilobitach na sekundę. Aby przeliczyć tą wartość na prędkość w „rachach“ należy skorzystać z prostego wzoru: prędkość = x/8/175. Wynika z tego że prędkość 1x jest równa 175 kB/s.

8. PLAY AUDIO

Rozpoczyna odtwarzanie audio. Działa bezproblemowo w większości urządzeń. Dziwactwem ze strony konstruktorów jest umożliwienie odtworzenia tylko 65536 sektorów (ok. 14 minut). Eliminuje to ją w niemal wszystkich zastosowaniach.

Bajt	7	6	5	4	3	2	1	0
0								45h
2	MSB							LSB
								Adres bloku od którego rozpoczniemy odtwarzanie
7	MSB							LSB
								Liczba sektorów do odtworzenia

9. PLAY AUDIO MSF

Zasada działania jest identyczna jak w poprzedniej, lecz komenda wymaga podania adresu początku i końca odtwarzania w formacie MSF.

Bajt	7	6	5	4	3	2	1	0
0								47h
3								początkowe M
4								początkowe S
5								początkowe F
6								końcowe M
7								końcowe S
8								końcowe F

10. PLAY CD

Komenda ta jest bardzo rozbudowana i posiada sporo opcji, takich jak możliwość wyboru wyjścia audio itp. Niestety żaden z testowanych napędów nie obsługiwał tej komendy.

Bajt	7	6	5	4	3	2	1	0
0								BCh
1	-							Oczekiwany typ MSF sektora
2	MSB							LSB
								Adres bloku od którego rozpoczniemy odtwarzanie
7	MSB							LSB
								Liczba sektorów do odtworzenia
11	SPEED							Port 2 Port 1 - Audio
								SPEED: 0 odtwarzanie z prędkością 1x, 1 - z największą możliwą. Port 1/2: włącz wyjścia cyfrowe. Audio: włącz wyjścia analogowe. Typ sektora: patrz komenda Read CD.

11. STOP PLAY

Zatrzymuje odtwarzanie. Nie ma żadnych parametrów.

12. PAUSE

Bajt	7	6	5	4	3	2	1	0
0								4Bh

Tab. 8. Kody rodzaju nośnika

Kod rodzaju nośnika	Opis
00h	Nieznany typ nośnika
01h	120mm DANE
02h	120mm AUDIO
03h	120mm DANE + AUDIO
05h	80mm DANE
06h	80mm AUDIO
07h	80mm DANE + AUDIO
10h	CDR nieznanego rozmiaru
11h	120mm CDR DANE
12h	120mm CDR AUDIO
13h	120mm CDR DANE + AUDIO
15h	80mm CDR DANE
16h	80mm CDR AUDIO
17h	80mm CDR DANE + AUDIO
20h	CDRW nieznanego rozmiaru
21h	120mm CDRW DANE
22h	120mm CDRW AUDIO
23h	120mm CDRW DANE + AUDIO
25h	80mm CDRW DANE
26h	80mm CDRW AUDIO
27h	80mm CDRW DANE + AUDIO
30h	Nieznany typ nośnika
70h	Tacka wsunięta, brak nośnika
71h	Tacka wysunięta
72h	Błędny format nośnika

8 - RESUME
Gdy Resume = 1, odtwarzanie jest kontynuowane, gdy = 0 - wstrzymane.

14. PREVENT/ALLOW MEDIUM REMOVAL

Pozwala zabezpieczyć nośnik przed usunięciem:

Bajt	7	6	5	4	3	2	1	0
0								1Eh
4								LOCK

Gdy LOCK = 1 zapęd blokuje nośnik.

15. SEEK

Ustawia mechanikę nad określoną pozycją. Komenda umożliwia np. odczytanie danych subkanałów określonego sektora.

Bajt	7	6	5	4	3	2	1	0
0								2Bh
2	MSB							LSB
								Numer LBA sektora

16. START/STOP/EJECT

Pozwala na zatrzymanie/rozpędzenie płyty oraz na operowanie tray'em.

Bajt	7	6	5	4	3	2	1	0
0								1Bh
1								Tryb
4								Eject Start

Możliwe operacje do wykonania:

Eject	Start	Wykonana operacja
0	0	Zatrzymaj dysk
0	1	Rozpędź dysk, odczytaj TOC
1	0	Wysuń tackę
1	1	Załaduj płytę (wsuń tackę)

17. READ

Pozwala na odczyt dowolnej liczby sektorów:

Bajt	7	6	5	4	3	2	1	0
0	A8h							

2 MSB Adres bloku od którego rozpoczniemy odczyt LSB

6 MSB Liczba sektorów do odczytania LSB

Jeśli liczba sektorów = 0 wtedy nie zostaną odczytane żadne dane. Ilość bajtów w sektorze jest zawsze równa 2048. Gdy zechcemy odczytać sektor o innej długości komenda wygeneruje błąd. Dane są wysyłane w postaci ciągu, nie są niczym rozdzielone.

18. READ CD

Rozbudowana wersja komendy READ. Pozwala odczytać nie tylko dane użytkowe ale i pozostałe informacje takie jak nagłówki czy pola synchronizacji.

Bajt	7	6	5	4	3	2	1	0
0	BEh							

1 - Oczekiwany typ sektora

2 MSB Adres bloku od którego rozpoczniemy odczyt LSB

6 MSB Liczba sektorów do odczytania LSB

9 Synch. Nagł. Dane EDC -

10 - Sub kanały

Dla nas takie dane jak nagłówki, synchronizacja oraz pola korekcji błędów są bezużyteczne więc nie będą ich omawiać. Podobnie nie ma sensu odczytywanie danych subkanałowych w czystej formie, skoro mamy do tego celu specjalne komendy. Przydatne natomiast może być pole typu sektora. Jeśli któryś z odczytanych sektorów nie jest zgodny z podanym przez nas typem to zostanie wygenerowany błąd. Poniżej przedstawiam wartości dla określonych typów:

Wartość (bin)	Typ
000	Wszystkie sektory są właściwe
001	CD AUDIO
010	Mode 1
011	Mode 2
100	Mode 2 Form 1
101	Mode 2 Form 2

19. READ CD MSF

Działanie identyczne jak w przypadku poprzedniej komendy z tym że do adresowania używamy MSF i określamy zakres sektorów do odczytu, a ich nie liczbę.

Bajt	7	6	5	4	3	2	1	0
0	47h							

2 -
3 początkowe M

4 początkowe S
5 początkowe F
6 końcowe M
7 końcowe S
8 końcowe F

Pozostałe pola mają identyczne znaczenie.

20. READ SUBCHANNEL

Odczytuje informacje subkanałowe aktualnego sektora. Dostarcza informacji o aktualnej pozycji, pozycji relatywnej do początku ścieżki, numerze ścieżki, indeksie, statusie odtwarzania audio oraz dane o ścieżce (rok nagrania, kraj, numer seryjny).

Bajt	7	6	5	4	3	2	1	0
0	42h							

1 - MSF -

2 - SubQ -

3 Typ subkanału do odczytania

6 Numer ścieżki

7 MSB Ilość danych do odebrania LSB

8

SubQ wskazuje czy komenda ma wysłać dane subkanałowe czy tylko nagłówek. Typy danych subkanałowych przedstawiono poniżej:

Kod Opis

01h Aktualna pozycja

02h Numer katalogowy

03h Informacje o ścieżce

Numer ścieżki jest brany pod uwagę tylko gdy typ danych = 03h.

Komenda zawsze zwraca nagłówek przed właściwymi danymi:

Bajt	7	6	5	4	3	2	1	0
1	Status Audio							

2 MSB Długość danych LSB

(nie uwzględnia nagłówka)

3

W tabelicy poniżej pokazano znaczenie kodów Status Audio:

Kod statusu audio Opis

00h Status audio nie obsługiwany

11h Trwa odtwarzanie

12h Odtwarzanie wstrzymane (pauza)

13h Odtwarzanie pomyślnie zakończone

14h Odtwarzanie zakończone z powodu błędu

15h Brak operacji audio

Jeśli pole długości danych jest równe 0 oznacza to brak dalszych danych do wysłania (bit SubQ = 0).

Tryb pozycji (01h)

Bajt	7	6	5	4	3	2	1	0
4	01h							

5 ADR 4/2 A/D Copy -

6 Numer ścieżki

7 Index

8 MSB Aktualna pozycja relatywna do początku płyty LSB

12 MSB Aktualna pozycja relatywna do początku ścieżki LSB

4/2: dźwięk czterokanałowy (1)/dwukanałowy (0)

A/D: audio (0)/dane (1)

Copy: zabronione kopiowanie (0)/dozwolone (1)

Ścieżka: numer ścieżki aktualnie odtwarzanej

Indeks: Ścieżka może być podzielona na indeksy. Pauza rozdzielająca umieszczona na początku ścieżki zawsze posiada indeks 0. Domyślny numer indeksu to 1. Zakres wynosi od 1 do 99. Rzadko spotyka się ścieżki nagrane z użyciem indeksów. Tablica zawartości w ogóle ich nie uwzględnia.

Aktualna pozycja: Nie wymaga komentarza. Gdy aktualny sektor jest pauzą wartość ta jest ujemna. Jak zapisuje się taką liczbę? Przykład liczby 8 bitowej (-128 do 127). Liczby od 0 do 127 są wartościami dodatnimi a od 255 do 128 wartościami ujemnymi. Tak więc ñ1 = 255, -2 = 254 itd. Wartość ujemna naszego adresu nie jest określona. Jeśli MSF = 1 to wartości zawsze są dodatnie.

Pozycja relatywna do początku ścieżki: jak wyżej z tym, że pozycja 0 jest początkiem aktualnej ścieżki a nie płyty.

Z doświadczeń wynika, że ta druga wartość potrafi być bardzo różnie wyliczana przez różne napędy co powoduje kłopoty sterownika z kompatybilnością. Natomiast obie liczby potrafią także przybrać na moment w zupełnie losowym czasie bardzo dziwne wartości (np. 11017340). W dokumentacji nic nie ma na ten temat, a przysporzyło to mi wielu problemów.

Tryb informacji o ścieżce (03h)

Bajt	7	6	5	4	3	2	1	0
4	03h							

5 - 4/2 A/D Copy -

8 OK -

9 Kod kraju

11 Kod właściciela

14 Rok nagrania

16 Numer seryjny

Bit OK sygnalizuje że dane są poprawne. OK = 0 oznacza że napęd nie potrafi odczytać danych lub subkanał ich nie zawiera.

Michał Wysocki

mwssoft@satkabel.com.pl