

# ATAPI

## specyfikacja standardu



## Sterowanie CD-ROM-ów i dysków twardych w praktyce, część 2

W drugiej części artykułu publikujemy resztę podstawowych informacji związanych z programową obsługą urządzeń dołączonych do interfejsu ATA. Dzięki nim samodzielne przygotowanie oprogramowania dla mikrokontrolera współpracującego z dyskiem twardym będzie znacznie łatwiejsze niż dotychczas.

### Tryby adresowania

Wyróżniamy dwa, wcześniej już wspomniane, tryby adresowania:

1. CHS - adresowanie odbywa się za pomocą trzech pól:
  - Cylinder (1...255),
  - Head (1...15),
  - Sector (0...65535).
2. LBA, w którym adres jest wyliczany ze wzoru:  $LBA = ((cylinder * heads\_per\_cylinder + heads) * sectors\_per\_track) + sector - 1$ .

Gdy urządzenie posiada mniej niż 16515072 sektorów, domyślnym trybem adresowania staje się tryb CHS. W prze-

ciwnym przypadku - LBA. Sterownik wbudowany w napęd, bez względu na pojemność nośnika danych, powinien obsługiwać oba tryby adresowania. Host wybiera tryb adresowania poprzez ustawienie bitu *LBA* w rejestrze *Device/Head*. Po włączeniu zasilania urządzenie IDE przechodzi automatycznie na tryb adresowania CHS.

### Sterowanie CD-ROM-em

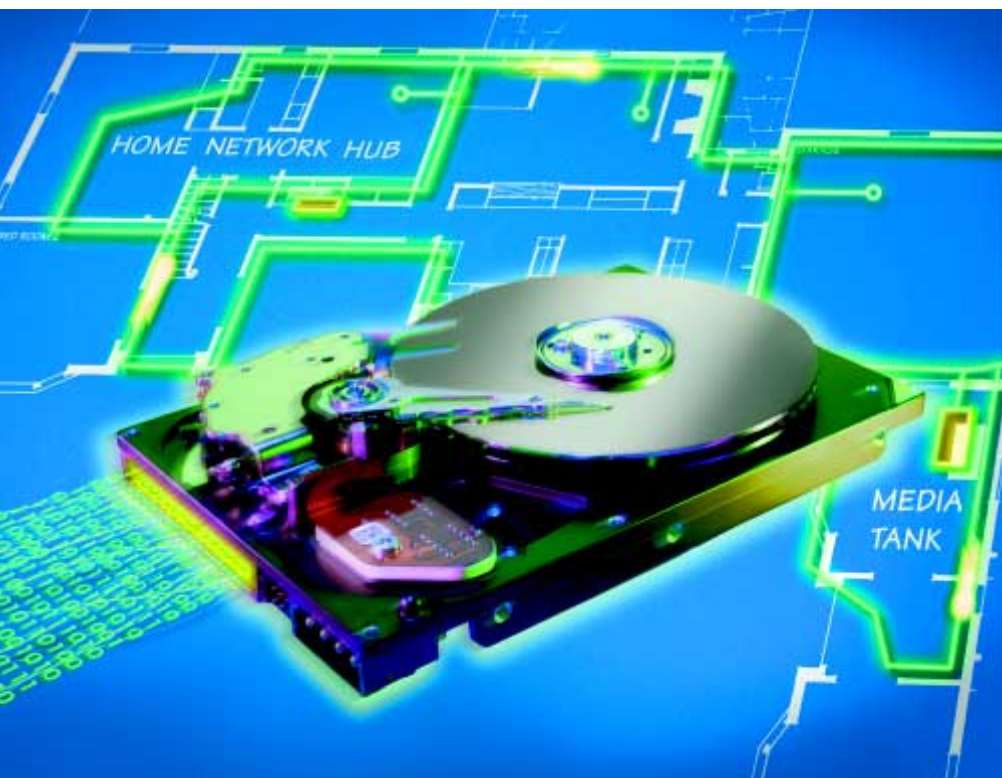
Sterowanie CD-ROM-em odbywa się za pomocą 12-bajtowych ramek zwanych rozkazami. Pierwszy bajt w ramce jest jednocześnie kodem roz-

kazu. Aby przesłać 12-bajtową ramkę sterującą należy przedtem - każdorazowo - ustawić odpowiednie rejestry.

Zaczynamy od sprawdzenia rejestru *Alternate Status* i stanu bitu *BSY*. Odczytujemy cyklicznie w pętli rejestr dopóki bit *BSY* nie będzie równy „0”. Odczytujemy rejestr *Status* - spowoduje to skasowanie ewentualnego przerwania. Do rejestru *Cylinder Low* wpisujemy wartość 12 (dziesięć) - jest to liczba danych, jaką chcemy przesłać do urządzenia. Zerujemy rejestry *Cylinder High*, *Features*, *Sector Count*, *Sector Number*. Do rejestrów *Cylinder Low* i *Cylinder High* wpisujemy liczbę bajtów, jaką jesteśmy w stanie odebrać od napędu. Sprawdzamy wartość bitu w rejestrze *Alternate Status* i czekamy dopóki bit *BSY* = 1. Odczytujemy rejestr *Status* - spowoduje to skasowanie zadania obsługi przerwania.

Wpisujemy do rejestru *Command* tzw. *Atapi Packet Command* - bajt o wartości 0xA0. Bajt ten informuje urządzenie, że będziemy przysyłać ramkę sterującą, a następnie:

- odczytujemy zawartość rejestru *Alternate Status*;
- czekamy, aż bit *BSY* będzie równy „0”;
- odczytujemy zawartość rejestru *Status*;
- sprawdzamy wartość bitu *ERR* - powinna być równa „0”;
- sprawdzamy wartość bitu *DRQ* w rejestrze *Status* - powinien być teraz ustawiony.
- teraz możemy wpisywać 12-bajtową ramkę sterującą;
- dane wprowadzamy na magistralę danych DD0...15; bajty parzyste ramki (0, 2, 4, 6, 8, 10) wpisuje-



my na mniej znaczącą część magistrali danych DD0...7 a bajty nieparzyste (1, 3, 5, 7, 9, 11) na bardziej znaczącą część magistrali DD8...15; każdorazowo musimy wpisać dwa bajty, czyli sumarycznie dokonujemy sześć cykli zapisu; obecnie tylko bardzo stare urządzenia obsługują transmisję danych w trybie ośmiobitowym.

Gdy porcja danych jest już gotowa na magistrali danych DD0...15, zerujemy linię *DLOW*. Odczekujemy ok. 1 μs i ustawiamy *DLOW* (poziom wysoki). Dane wpisywane są do urządzenia z boczem narastającym. Cykl zapisu powtarzamy sześć razy. Po wpisaniu 12 bajtów ramki urządzenia ustawia bit *BSY*. Odczytując stan rejestru *Alternate Status* czekamy, aż bit *BSY* będzie równy zero. Gdy tak się stanie, odczytujemy rejestr *Status*. Sprawdzamy czy bit *ERR* = 0 oraz czy bit *DRQ* = 1. Następnie musimy odczytać ile bajtów danych ma do przekazania CD-ROM. Wartość ta składana jest z młodszego bajtu pobieranego z rejestru *Cylinder Low*, oraz starszego bajtu z rejestru *Cylinder High*. Dane z CD-ROM-u do hosta są transmitowane w trybie 16-bitowym. Każde opadające zbocze sygnału na linii *DIOR* powoduje wystawienie dwóch bajtów danych na magistralę DD0...15. Po odczytanej liczbie bajtów powinniśmy sprawdzić czy bit *ERR* nie jest ustawiony w rejestrze *Status*. Gdy bit *ERR*=0 możemy przyjąć, że dane odebrano prawidłowo.

Wyżej wymienione czynności powtarzamy każdorazowo, gdy chcemy przesłać porcje danych do CD-ROM-u.

**Formaty ramek sterujących napędem**

*Eject CD* - wysunięcie płyty z napędu wymaga wysłania następującej sekwencji bajtów: 0x1B, 0x01, 0x00, 0x00, 0x02, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00.

*Load CD* - załadowanie krążka do napędu wymaga wysłania następującej sekwencji bajtów: 0x1B, 0x01, 0x00, 0x00, 0x03, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00.

*Start CD* - wystartowanie napędu i przeczytanie TOC wymaga wysłania następującej sekwencji bajtów: 0x1B, 0x01, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00.

*Stop CD* - zatrzymanie napędu wymaga wysłania następującej sekwencji bajtów: 0x1B, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00.

Po wykonaniu powyższych poleceń CD-ROM nie zwraca żadnych da-

nych. Jak widać kod komendy (najmłodszy bajt) jest w tych trzech przypadkach taki sam. Zmienia się tylko wartość czwartego bajtu. W bajcie pierwszym ustawiony jest bit zerowy - oznacza to, że *Status* ma być zwrócony natychmiast po tym, jak sprawdzona będzie poprawność przesłanej ramki przez CD-ROM, stanu tacki (czy wysunięta, czy nie) oraz status płyty. W przeciwnym przypadku *Status* jest zwracany po skończeniu całej operacji.

**Napędy CD-ROM są wyposażone w interfejs ATAPI, który jest rozszerzeniem klasycznego IDE (ATA). Charakteryzuje się on zastosowaniem protokołu niemal identycznego ze SCSI, przy zachowaniu mechanicznej i elektrycznej kompatybilności z IDE.**

TOC - *Table of Contents* - to bardzo ważna tablica. Zawiera adresy początkowe poszczególnych ścieżek, sesji, itp. Jej odczyt przebiega następująco:

*Zatrzymanie odtwarzania* płyty CD audio (*Stop*) wymaga wysłania następującej sekwencji bajtów: 0x4E, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00.

*Test gotowości* urządzenia wymaga wysłania następującej sekwencji bajtów: 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00.

Następnie wysyłamy komendę *Request Sense Key*, która składa się z następującej sekwencji bajtów:

0	0x03	W wyniku odpowiedzi na tę ramkę otrzymamy z CD-ROM-u ASC i ASCQ.
		Znajdują się one odpowiednio w bajcie 12 i 13.
		ASC = 0x00 oznacza płytę CD w napędzie.
		ASC = 0x3A brak płyty w napędzie.
		ASC = 0x02 głowice nie ustawiły się jeszcze - musimy czekać.
		ASC = 0x04 brak gotowości napędu - musimy czekać.

1	0x00
2	0x00
3	0x00
4	0x60
5	0x00
6	0x00
7	0x00
8	0x00
9	0x00
10	0x00
11	0x00

Odczyt TOC wymaga wysłania następującej sekwencji bajtów:

0	0x43	
1	0x00	CD zwróci informację w LBA (0x00)- nie w MSF (0x02)
2	0x00	
3	0x00	

4	0x00	
5	0x00	
6	0x01	Numer początkowej ścieżki od której CD-ROM ma zwrócić informację
7	0x00	
8	0x64	100 bajtów na dane z CD-ROM-u
9	0x00	
10	0x00	
11	0x00	

Poniżej pokazano przykładową ramkę wysłaną przez CD-ROM w odpowiedzi na zapytanie o TOC. Pierwsze dwa bajty określają ilość przesłanej informacji w bajtach. Bajt czwarty określa liczbę utworów na płycie. Bajty: 5, 13, 20 itd. są znacznikami początku adresu - mają stałą wartość 0x10. Mają charakter czysto informacyjny. Ważną ścieżką jest ścieżka o numerze 0xAA (tzw. *Lead Track*) - adres tej ścieżki oznacza ostatni sektor na płycie CD.

Numer utworu	Adres utworu			
	MSB	...	...	LSB
1	0x00	0x00	0x20	0x00
2	0x00	0x20	0x38	0x1C
3	0x00	0x2C	0x3B	0x21
4	0x00	0x37	0x33	0x03
0xAA	0x00	0x4A	0x38	0x1C

Nr bajtu	Wartość	Komentarz
0	0x00	MSB TOC Data Length
1	0x2A	LSB TOC Data Length
2	0x01	First Track Number
3	0x04	Last Track Number
4	0x00	Reserved
5	0x10	ADR (4 bity) CONTROL (4 bity)
6	0x01	TrackNumber
7	0x00	Reserved
8	0x00	MSB
9	0x00	
10	0x02	
11	0x00	LSB
12	0x00	Reserved
13	0x10	ADR (4 bity) CONTROL (4 bity)
14	0x02	TrackNumber
15	0x00	Reserved
16	0x00	
17	0x20	
18	0x38	
19	0x1C	
20	0x00	Reserved
21	0x10	ADR(4 bity) CONTROL(4 bity)
22	0x03	Track Number
23	0x00	Reserved
24	0x00	
25	0x2C	
26	0x3B	
27	0x21	
28	0x00	Reserved
29	0x10	ADR(4 bity) CONTROL(4 bity)

30	0x04	Track Number
31	0x00	Reserved
32	0x00	
33	0x37	
34	0x33	
35	0x03	
36	0x00	
37	0x10	
38	0xAA	Lead OUT
39	0x00	
40	0x00	
41	0x4A	
42	0x38	
43	0x1C	

Odtwarzanie płyty CD audio wymaga wysłania następującej sekwencji bajtów:

0	0xA5	
1	0x00	
2	0x00	
3	0x00	Adres początkowy ulworu (tutaj track 1)
4	0x20	
5	0x00	
6	0x00	
7	0x4A	Ilość danych - nie adres końcowy!!!
8	0x38	Czyli do odtworzenia cała płyta
9	0x1C	
10	0x00	
11	0x00	

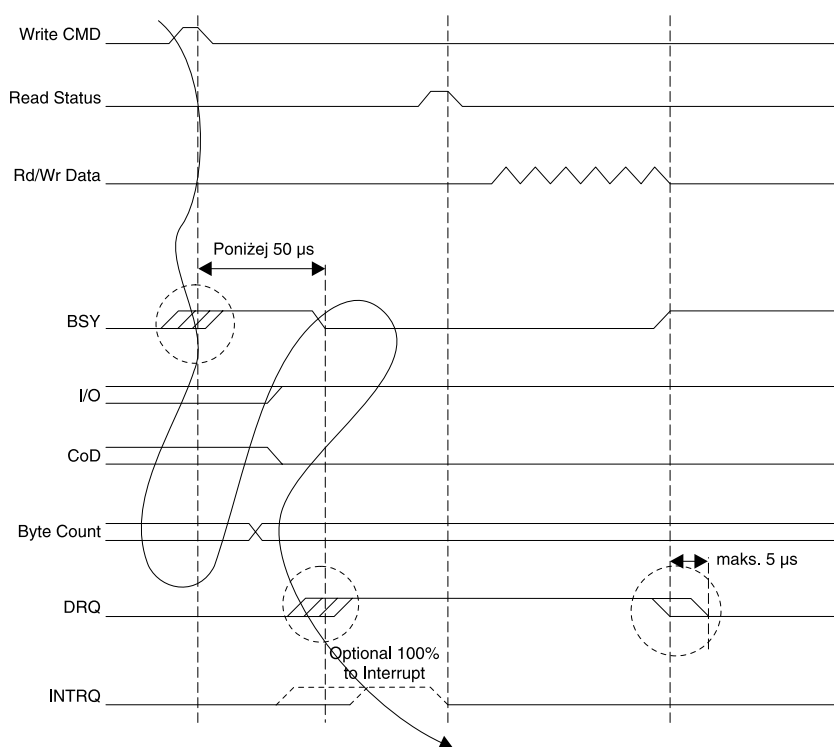
*Wstrzymanie lub wznowienie odtwarzania* (PAUSE) następuje po wysłaniu następującej sekwencji bajtów: 0x4B, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, xxx = 0x01 (odtworzenie lub wznowienie) xxx = 0x00 (odtworzenie wstrzymane), 0x00, 0x00, 0x00.

*Zatrzymanie odtwarzania płyty CD audio* (STOP) następuje po wysłaniu następującej sekwencji bajtów: 0x4E, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00.

## Odczyt danych z sektorów

### Dysk twardy w trybie CHS

Sprawdzamy stan bitu *Busy* w rejestrze *Alternate status*. Gdy BSY = 1, czekamy na BSY=0 i odczytujemy rejestr *Status*. W rejestrze *Device Head* zerujemy bity *LBA* (odczyt będzie w trybie CHS) oraz bit *DEV* (gdy chcemy odczytać dane z *Mastera* - w przeciwnym wypadku ustawiamy ten bit). Bity 7 i 5 domyślnie ustawiamy. Na mniej znaczące bity tego rejestru wpisujemy adres *Head*. W rejestrze *Alternate Status* sprawdzamy stan bitów *DRDY* i *BSY*. Pierwszy musi być ustawiony, drugi wyzerowany. Odczytujemy rejestr *Status*. Do rejestrów *Cylinder High* i *Cylinder Low* wpisujemy starszy bajt i młodszy bajt początkowego adresu *Cylinder*. Do rejestru *Sector Number* wpisujemy liczbę sektorów do odczytu.



Rys. 4. Przebiegi czasowe charakterystyczne dla zapisu polecenia do sterownika

Do rejestru *Command* wpisujemy wartość 0x20 - kod komendy *ATA\_READ\_SECTS*. Następnie czekamy ok. 10 µs, odczytujemy rejestr *Alternate Status* i sprawdzamy bit *BSY*. Czekamy aż bit *BSY* będzie wyzerowany. Odczytujemy rejestr *Status*, aby skasować przerwanie. Sprawdzamy stan bitów: *ERR* (powinien być wyzerowany), *DRQ* (powinien być ustawiony - urządzenie chce przesłać dla nas dane).

Dane odczytujemy jak w poprzednim przykładzie. Opadające zbocze na linii *DIOR* powoduje wystawienie dwóch kolejnych bajtów na magistralę *DD0...15*. Ilość danych, które musimy odebrać jest określona liczbą sektorów, jaką wcześniej wpisaliśmy do rejestru *Sector Number* pomnożoną przez zmienną, która określa liczbę bajtów w sektorze - tzw. *BytesPerSector*. Wartość ta może wynosić 4096 lub 8192 B.

### Dysk twardy w trybie LBA

Dane odczytujemy bardzo podobnie jak w trybie CHS z tą różnicą, że adres początkowy podajemy w *LBA*, następnie sprawdzamy stan bitu *BSY*. Gdy jest wyzerowany - ustawiamy bit *LBA* w rejestrze *Device/Head*. Do mniej znaczącej połowy rejestru wpisujemy bity 27...24 adresu początkowego w trybie *LBA*. Do rejestru *Cylinder High* wpisujemy bity 23...16 adresu *LBA*, do rejestru *Cylinder Low* bity 15...8, a do rejestru *Sector Number* bity 0...7. Do rejestru *Sector*

*Count* wpisujemy liczbę sektorów do odczytu.

### Zależności czasowe

Na rys. 4 przedstawiono przykładowe zależności czasowe przebiegu sygnałów na magistrali danych i liniach sterujących CD-ROM-u. Na rysunku pominięto dokładne specyfikacje czasów pomiędzy zmianami sygnałów - ma to być tylko ilustracja działania interfejsu.

### Aplikacja

Na podstawie przedstawionego opisu przygotowałem oprogramowanie sterujące zaimplementowane w mikrokontrolerze z rodziny AVR. Program dla niego jest napisany w C - wszystkie źródła są dostępne na stronie internetowej [http://www.pelos.pl/cd\\_free/](http://www.pelos.pl/cd_free/) oraz na płycie CD-EP8/2002B.

**Pawel Dienwebel**  
pelos@pelos.pl

### Literatura

- [1] INF-8020.pdf - AATA Packet Interface for CDROM rev. 2.6 lub wyższe
- [2] CD3610 - Command Specification
- [3] [http://www.pelos.pl/cd\\_free/](http://www.pelos.pl/cd_free/)
- [4] <http://www.yampp.com>
- [5] Standard ECMA -130 „Data interchange on read-only 120 mm optical data disks (CD-ROM)“
- [6] SFF8090i v4 - Fuji Commands for Multimedia Devices
- [7] [http://www.republika.pl/romek\\_by/index.html](http://www.republika.pl/romek_by/index.html)