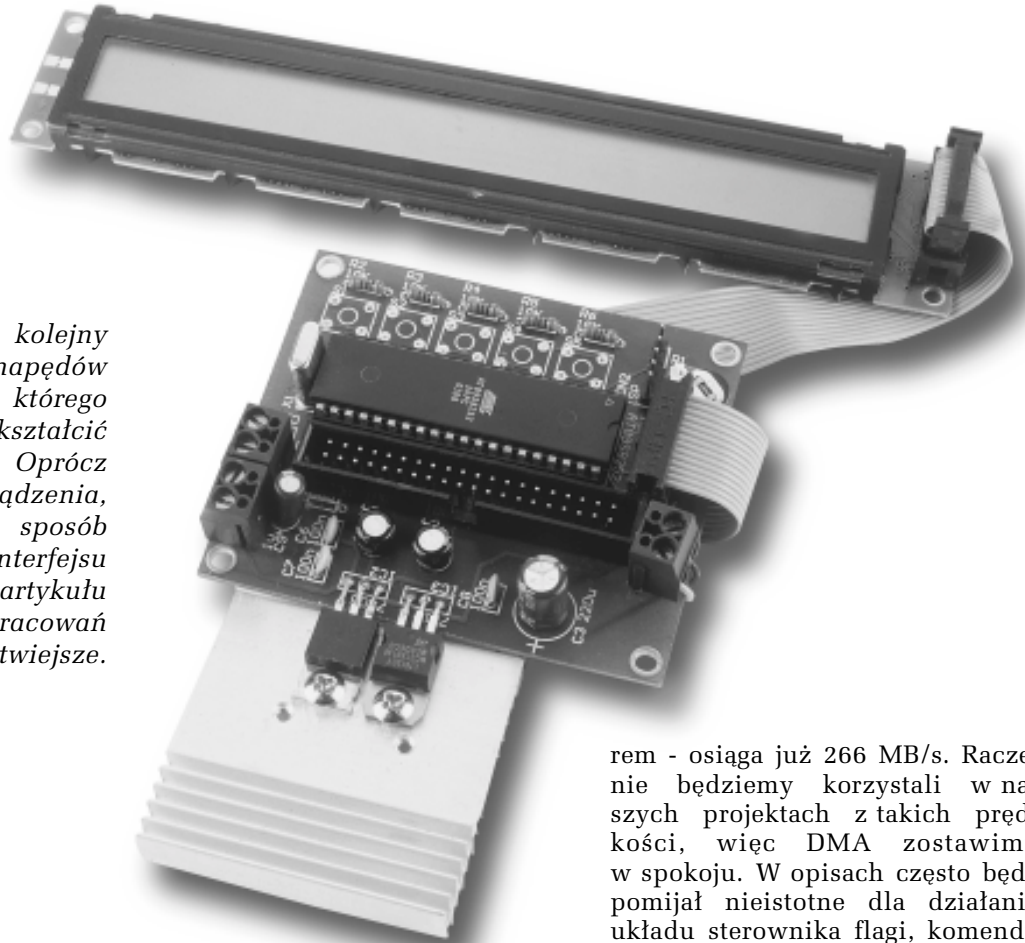


Sterownik napędu CD-ROM, część 1

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**.

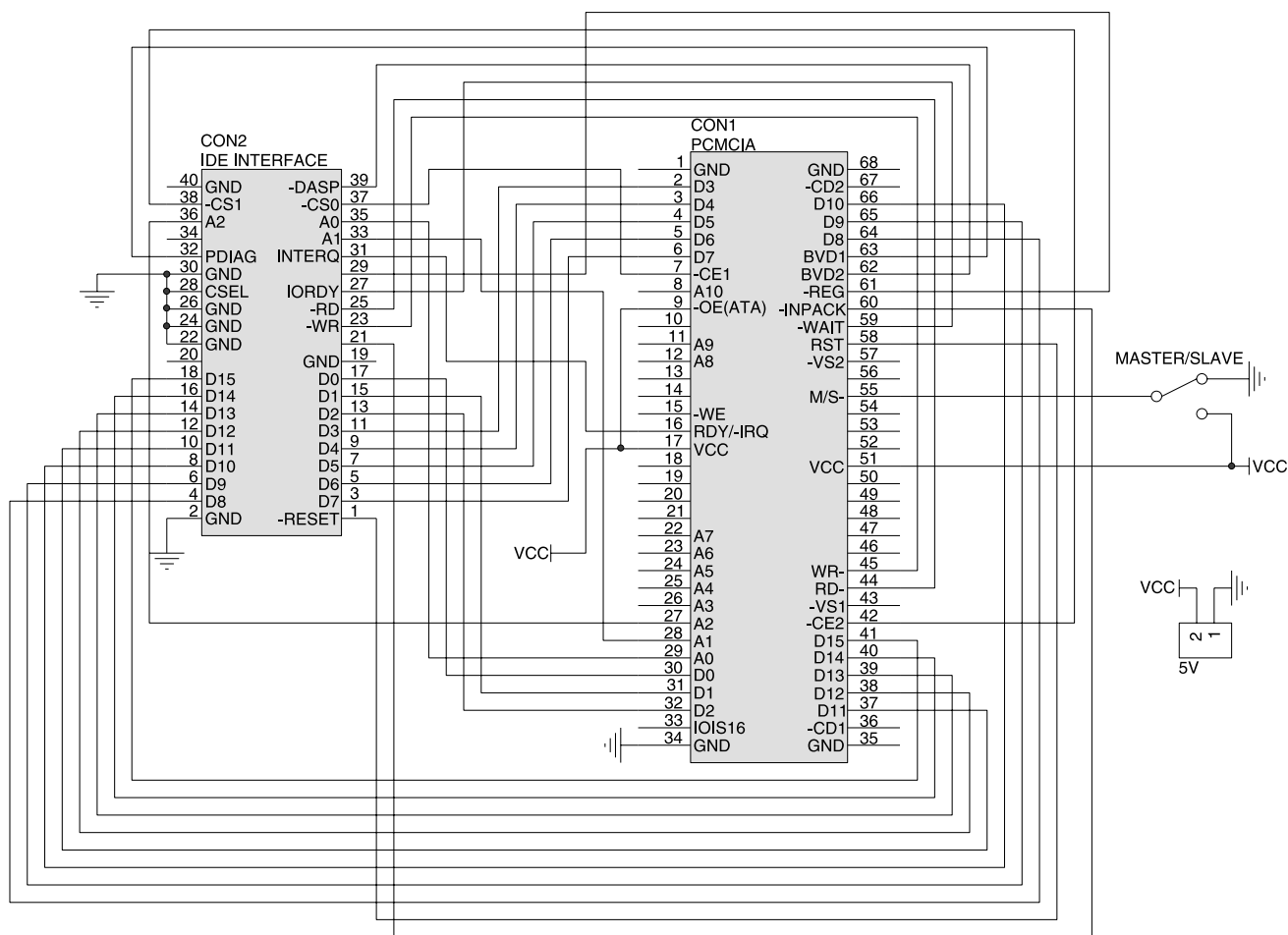
Możliwości i ograniczenia sterownika opisanego w artykule

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.

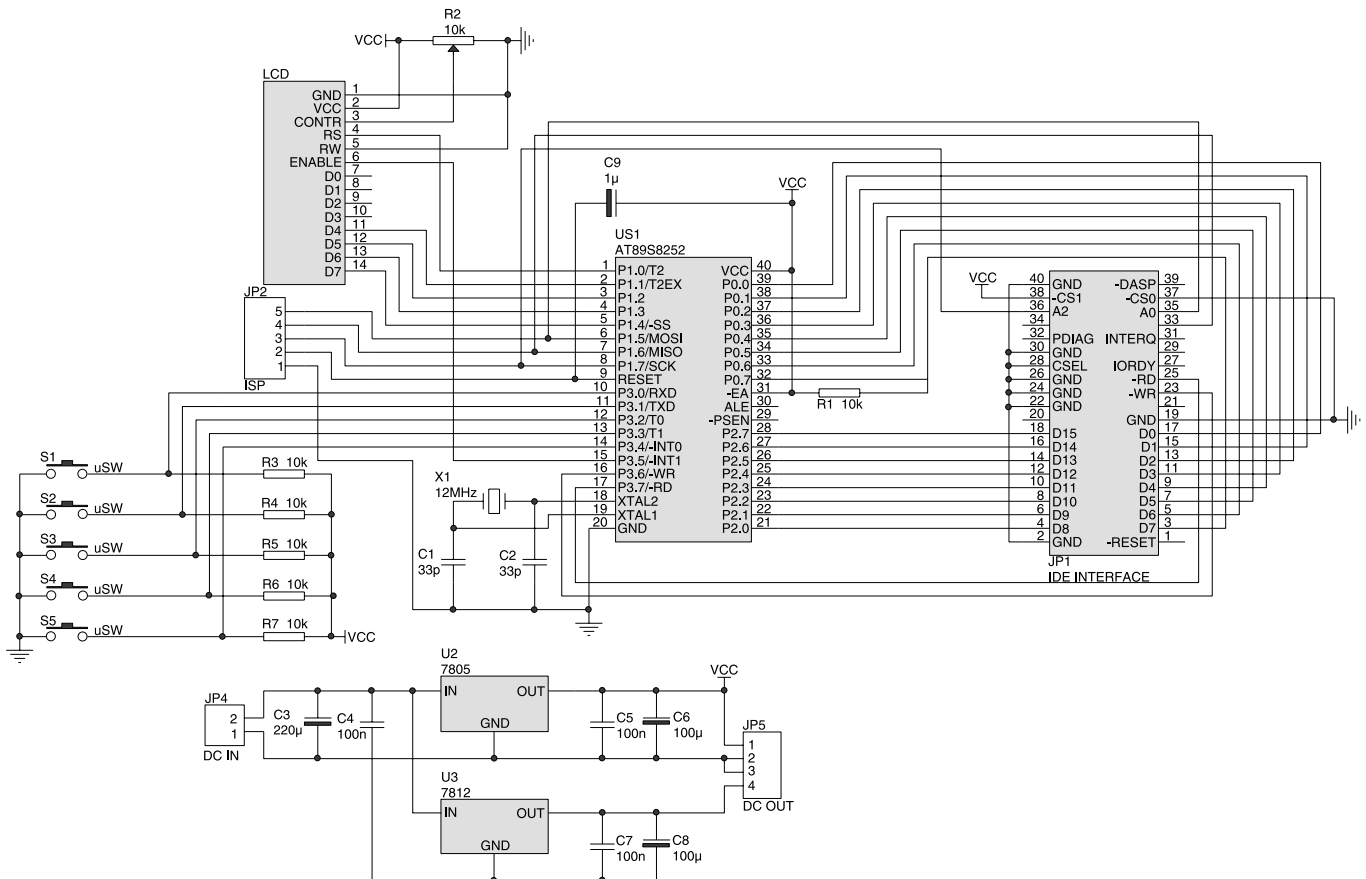
Mikrokontroler sterownika pracuje w typowej aplikacji z zegarem kwarcowym 12 MHz. Zerowanie po włączeniu zasilania zapewnia kondensator 1 μ F

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.

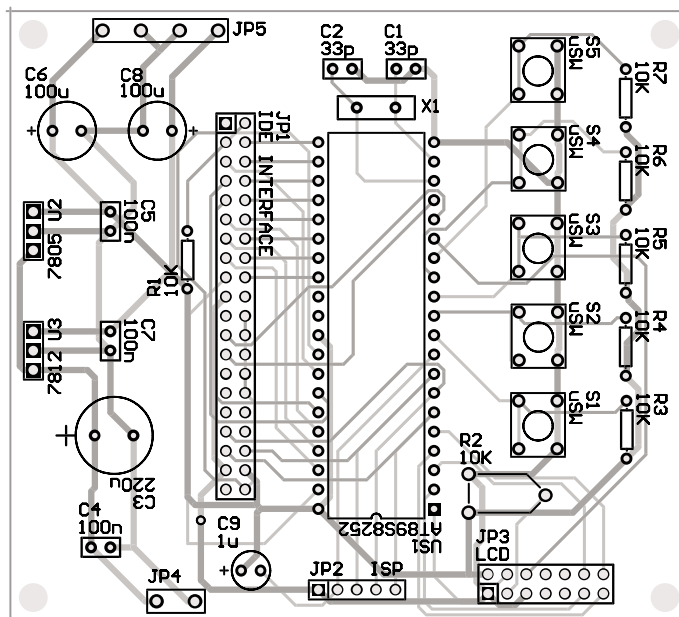
To tyle na temat struktury programu - plik źródłowy jest dostępny na stronie www.ep.com.pl w dziale *Download>Dokumentacje* oraz na płycie CD-EP8/2002B.

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:

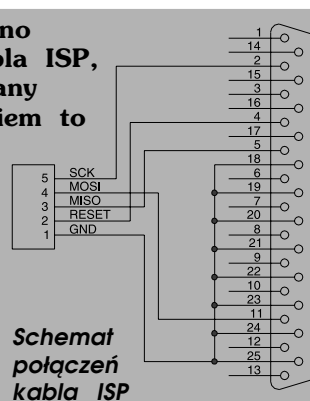
```
Templcd = Makebcd(X)
Shift Templcd, Right, 4
Lcd Templcd

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.