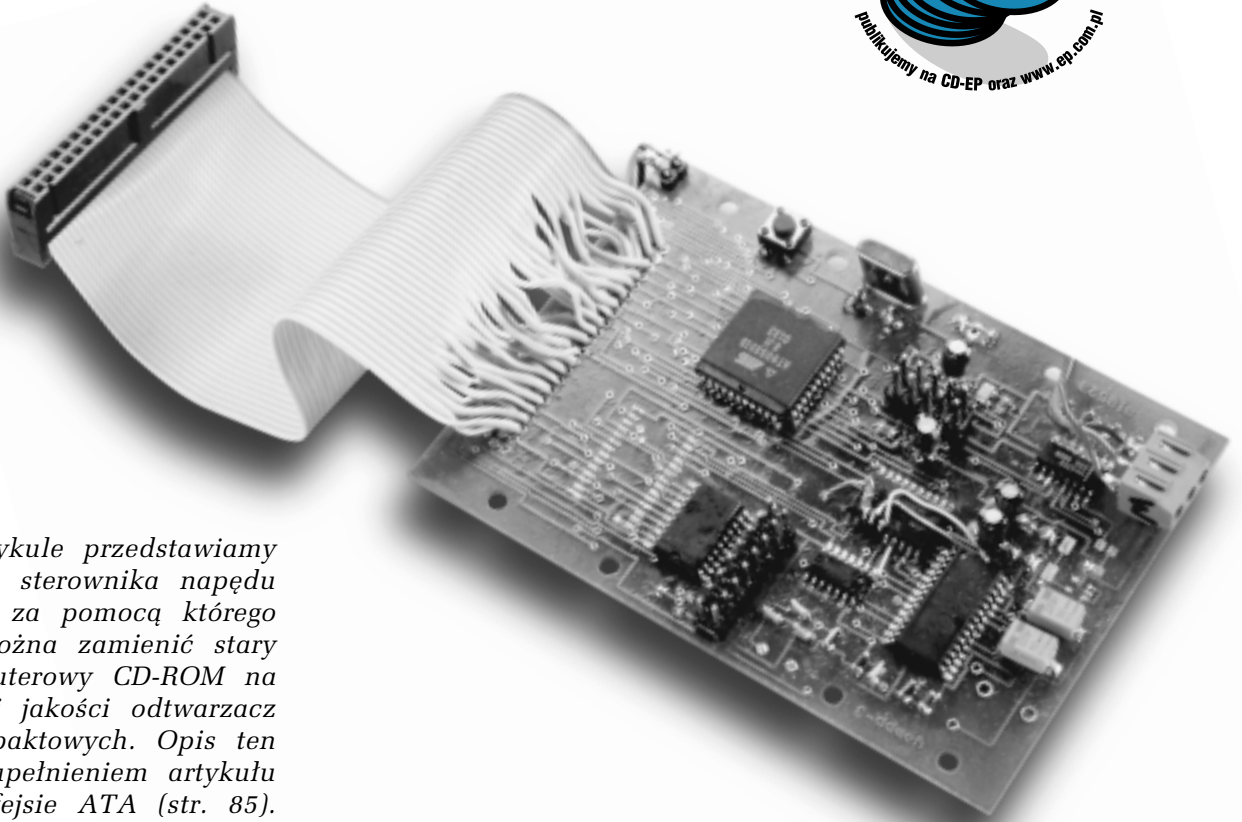


Sterownik odtwarzacza CD



W artykule przedstawiamy projekt sterownika napędu CD, za pomocą którego można zamienić stary komputerowy CD-ROM na niezłej jakości odtwarzacz płyt kompaktowych. Opis ten jest uzupełnieniem artykułu o interfejsie ATA (str. 85). Na płycie CD-EP7/2002B publikujemy kod źródłowy programu dla mikrokontrolera AVR.

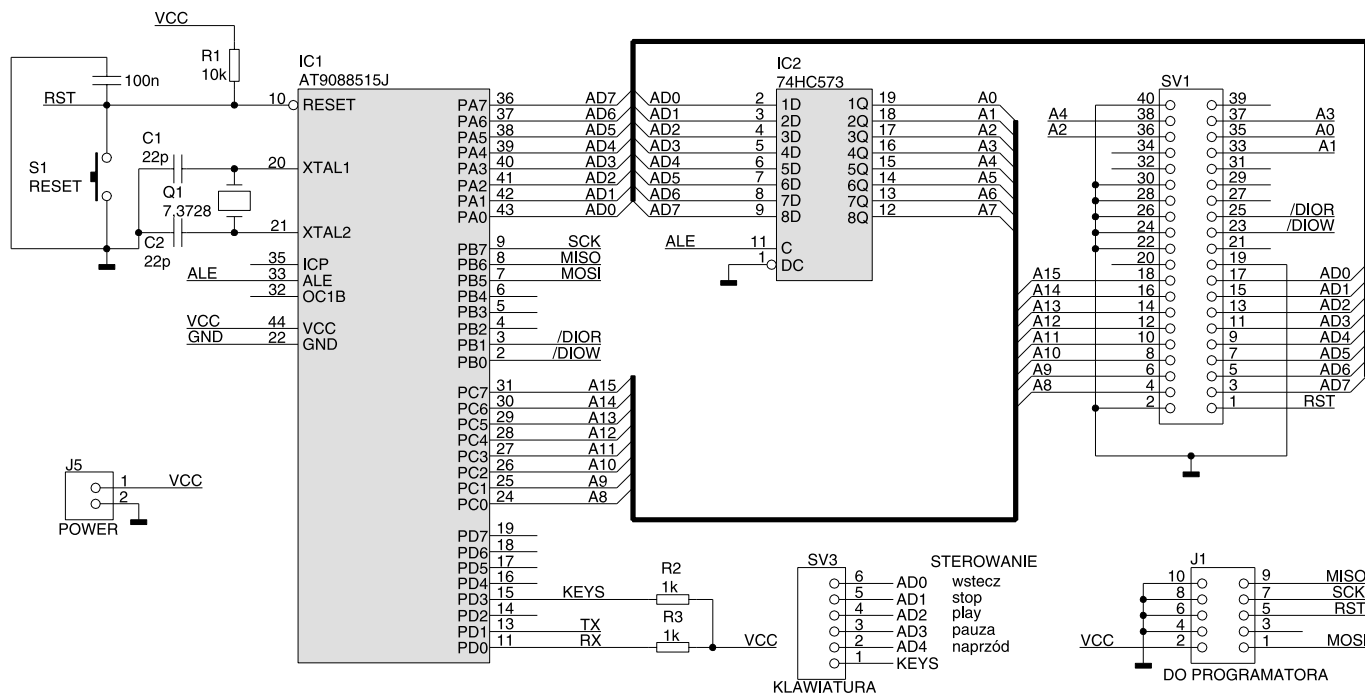
Do budowy własnego odtwarzacza CD wykorzystałem moduł YAMPP-3 w okrojonej wersji. Może na początku słów kilka o tym układzie. Jest to odtwarzacz plików MP3 zbudowany na mikrokontrolerze AVR AT90S8515 oraz sprzętowym dekodерze MP3 VS1001 (układ scalony firmy VLSI). Autorem tego projektu jest Jesper Hansen, a jego opis i dokumentacja są dostępne na stronie WWW o adresie <http://www.yampp.com> (jego budowę opiszemy we wrześniowym wydaniu EP).

Do przechowywania plików MP3 układ wykorzystuje 2,5" dyski twarde ze złączem 44-stykowym. Sterownik YAMPP'a potrafi obsługiwać systemy plików FAT32-LBA oraz FAT16. Jeżeli dobrze pamiętam, rekordzista YAMPP-3 pracował z dyskiem twardym o pojemności 40 GB. Ja bez problemu używałem dysk twardy o pojemności 10 GB.

Po mojej adaptacji na potrzeby odtwarzacza płyt audio CD, z płytki YAMPP'a-3 został jedynie procesor AVR z rezonatorem kwarcowym, układ zatrasku 74HC573 (choć po pewnych modyfikacjach w kodzie programu, można również zrezygnować i z niego) oraz 40-stykowe złącze IDE.

Opis układu

Schemat elektryczny sterownika przedstawiono na **rys. 1**. Procesor AT90S8515 jest procesorem typu RISC. W układzie odtwarzacza CD zastosowano konfigurację układu z zatraskiem 8-bitowym 74HC573. W normalnej sytuacji, gdy do procesora jest dołączana zewnętrzna pamięć, służy on do zatrzaśnięcia mniej znaczącej części adresu danej - w tym przypadku A0...A7. Adres ten jest dostępny w porcie A podczas pierwszego cyklu odczytu pamięci. Bardziej znacząca część adresu wy-



Rys. 1. Schemat elektryczny sterownika CD

stawiana jest na wyprowadzeniach portu C (rys. 2).

Trzy linie adresowe interfejsu IDE - DAO (styk 35), DA1 (styk 33) i DA2 (styk 36) połączono odpowiednio z liniami adresowymi A0, A1 i A2 systemu. Linie CS0 i CS1, służące do wyboru bloku rejestrów *Command block* lub *Control block*, dołączono do linii adresowych A3 i A4 systemu mikroprocesorowego. Można łatwo zauważyć, że adres na liniach adresowych A0...A7 systemu mikroprocesorowego uzyskujemy poprzez zatrzaśnięcie danej z portu A. Do zatrzaśnięcia danej służy linia ALE mikroprocesora. W układzie sterownika napędu CD-ROM zastosowano pewien trick związany z obsługą pamięci zewnętrznej:

otóż sposób obsługi pamięci zewnętrznej zależy od bitu SRE w rejestrze MCUCR. Informuje on procesor o obecności (bit ustawiony) lub braku (bit wyzerowany) zewnętrznej pamięci SRAM. Gdy bit SRE jest ustawiony, linie portów A i C stają się liniami magistrali adresowej bez względu na to, jak wcześniej skonfigurowaliśmy porty A i C. Dokonując odczytu z komórki należącej do zewnętrznej pamięci, mniej znaczący bajt adresu A0...A7 jest generowany poprzez zatrzaśnięcie danej z portu A. Generując odpowiedni adres na liniach A0...A7 wybieramy odpowiedni rejestr sterownika wbudowanego w CD-ROM. Następnie możemy wyzerować bit SRE. Porty A i C stają się wtedy

List. 1. Fragment kodu źródłowego odpowiedzialny za obsługę klawiatury

```

event_e get_key_event(void)
{
    static u08 keydown = 0;
    u08 i;
    event_e event = EV_IDLE;

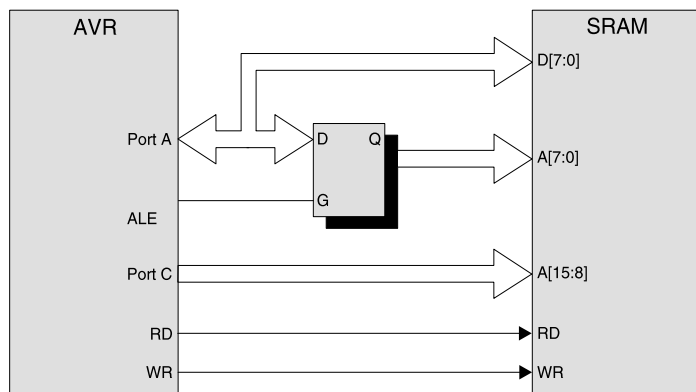
    cbi(MCUCR, SRE); // disable EXTRAM

    outp(0xff, PORTA-1); // set port as output
    outp(0, PORTA); // set all pins to 0
    WDR; // allow time for port
    // to activate
    delay (500); //odczekanie 1 ms
    // w przypadku dłuższych przewodów

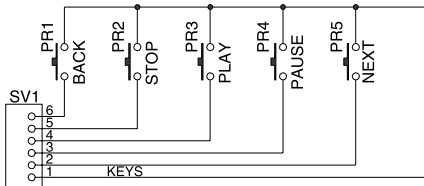
    if (keydown) // if a key was pressed
    // earlier
    {
        // see if it has been released
        if (bit_is_set(PIND,PD3)) // check bit
        {
            // no key is down
            // return last key seen
            switch (keydown)
            {
                case 1 : event = EV_PREV; break;
                case 2 : event = EV_STOP; break;
                case 3 : event = EV_PLAY; break;
                case 4 : event = EV_PAUSE; break;
                case 5 : event = EV_NEXT; break;
            }
            keydown = 0;
        }
    }
    else
    {
        // check if a key is down
        if (bit_is_clear(PIND,PD3))
        {
            // a key is active, check which one
            for (i=0;i<8;i++)
            {
                // write bit mask
                outp(~(1<<i), PORTA);
                // wait a while
                asm volatile("nop");
                WDR;
                // opoznienie w przypadku
                // dłuższych przewodów
                delay (500);

                // this bit ?
                if (bit_is_clear(PIND,PD3))
                {
                    // store the key
                    keydown = i+1;
                    // and exit
                    break;
                }
            }
        }
    }

    sbi(MCUCR, SRE); // enable RAM
    return event;
}
    
```



Rys. 2. Ilustracja zasady rozdzielania adresu i danej z magistrali multipleksowej

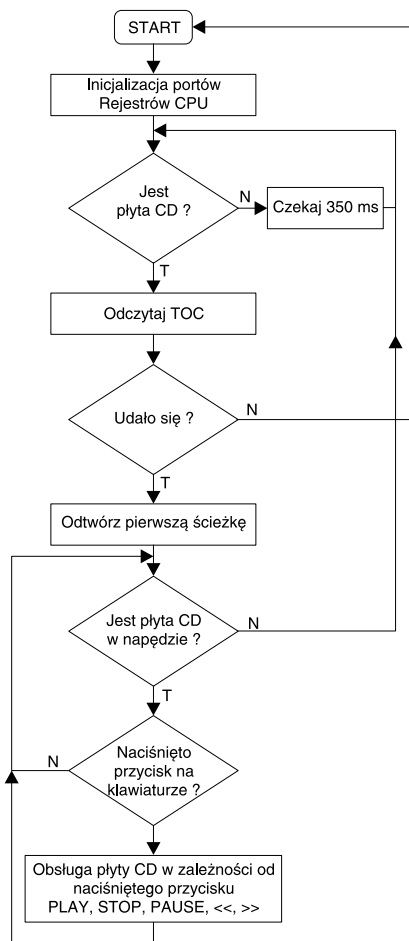


Rys. 3. Schemat elektryczny układu klawiatury

normalnymi portami. Dzięki temu mamy do dyspozycji magistralę 16-bitową. Zmiany konfiguracji portów A i C nie mają żadnego wpływu na stan CD-ROM-u, dopóki nie ustawimy DIOR lub DIOW na poziomie niskim.

Napęd jest sterowany za pomocą prostej (rys. 3), pięcioprzyciskowej klawiatury. Umożliwia ona wybranie: następnego lub poprzedniego utworu, rozpoczęcie odtwarzania, pauzy oraz zatrzymania odtwarzania płyty CD. Klawiatura jest podłączana za pomocą złącza SV3.

Zastosowany w projekcie sposób dołączenia klawiatury jest



Rys. 4. Algorytm działania programu sterującego

Środowisko pracy wykorzystane podczas realizacji prezentowanego sterownika

Kompilator - do tworzenia i kompilowania programów na procesory z rodziny AVR - używam świetnego i darmowego kompilatora AVR-GCC w wersji 3.0.2.

Edytor - jako edytor z podświetlaną składnią wykorzystuję darmową wersję AVR-studio - ver. 3.53 (niestety, najnowsza wersja 4.0 tego programu ogranicza edycję jedynie do plików assemblera).

Programator - jedną z największych zalet procesorów z rodziny AVR jest możliwość ich programowania poprzez interfejs ISP, czyli bez demontowania procesora z płytki. Jako programowego sterownika programatora używam darmowego programu YAAP ver 0.2 autorstwa Jespera Hansena (krótki jego opis przedstawiamy w Miniprojektach, na str. 75). Można go pobrać ze strony WWW - <http://www.myplace.nu/avr/yaap/index.htm>.

bardzo czytelny, lecz zasługuje na kilka słów komentarza. Linia KEYS jest skonfigurowana jako wejściowa. Funkcja sprawdzająca stan klawiatury znajduje się w pętli głównej programu. Przed przystąpieniem do sprawdzenia stanu przycisków „odłączana” jest pamięć zewnętrzna poprzez skasowanie bitu SRE w rejestrze MCUCR. Jest to konieczne, ponieważ za odczytu klawiatury jest wykorzystywany PORT A, który normalnie służy do przesyłania adresu. Następnie program sprawdza czy naciśnięty został jakikolwiek klawisz - wtedy są zerowane wszystkie linie portu A. W kolejnym kroku jest sprawdzany stan wejścia KEYS. Gdy jest „1” - wychodzimy z procedury - oznacza to, że żaden klawisz nie jest naciśnięty (przedtem ustawiamy bit SRE). Gdy jest „0” - za pomocą przesuwanej jedynek sprawdzamy, która z linii portu A jest połączona z KEYS. Zapamiętujemy numer linii portu A oraz fakt wciśnięcia przycisku. Wychodzimy z procedury odczytu klawiatury, po czym jest ona znowu wywoływana. Wejście w procedurę spowoduje (przedtem zerujemy bit SRE) przejście do sprawdzenia, czy zapamiętany przycisk został już zwolniony. W tym celu zerujemy zapamiętaną linię portu A i sprawdzamy stan linii KEYS. Gdy na linii KEYS jest „1” - przycisk został już zwolniony. Funkcja zwraca numer zapamiętanej linii portu A. Gdy na linii

KEYS jest „0” - wtedy przycisk jest nadal wciśnięty. Wychodzimy z funkcji (ustawiając przed wyjściem bit SRE).

Na list. 1 znajduje się fragment kodu źródłowego odpowiedzialnego za obsługę klawiatury. Cały program oraz wzory płytek drukowanych (w wersjach SMD i przewlekanej) publikujemy na CD-EP7/2002B. Na rys. 4 zilustrowano algorytm działania programu sterującego.

Paweł Dienwebel
pelos@pelos.pl

Dodatkowe informacje można znaleźć w Internecie pod adresem: http://www.pelos.pl/cd_free/.

WYKAZ ELEMENTÓW

Rezystory

R1: 1kΩ M0805
R2: 10kΩ M1206
R12: 10kΩ M0805

Kondensatory

C1: 22pF C0805
C2: 22pF C0805
C3: 100n C1206

Półprzewodniki

IC1: AT90S8515J (PLCC44) - zaprogramowany
IC2: 74HC573 SO20L

Różne

J1: programator MA05-2
J5: POWER MA02-1
Q1: 7,3728 MHz HC49U-V
S1: RESET ALPS_45
SV1: ML40
SV3: MA06-1