

# Mikser audio ze sterowaniem cyfrowym, część 2

## kit AVT-490

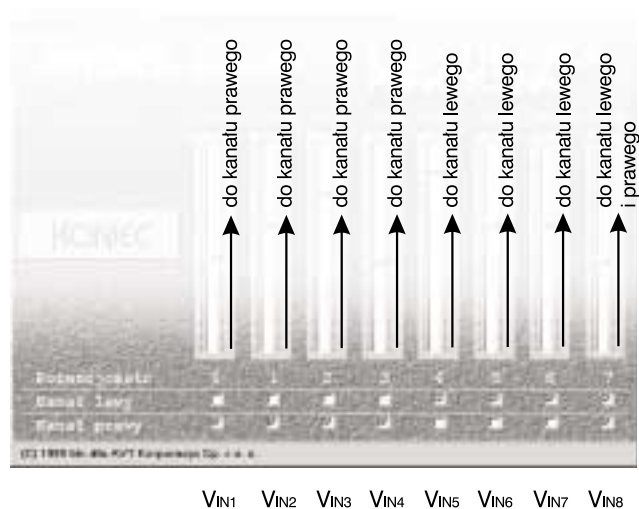


*Druga część artykułu dotyczy komputerowego sterowania pracą miksera.*

*Przedstawiona zostanie przykładowa procedura wysyłania danych do miksera oraz sposób połączenia komputera z mikserem.*

Przedstawiony w pierwszej części artykułu sposób sterowania mikserem jest naturalny, jednak rozwiązanie takie nie w pełni wykorzystuje możliwości układu SSM2163. Układ SSM2163 zapewnia elastyczne konfigurowanie wejść audio (tab. 1 w pierwszej części artykułu), jednak wyposażenie w te funkcje konsoli mikserskiej znacznie skomplikowałoby budowę urządzenia. Przykładem ilustrującym przewagę sterowania komputerowego nad tradycyjnym (oczywiście w przypadku, gdy istotnym czynnikiem jest cena urządzenia) niech będzie następująca sytuacja: w kanale wyjściowym prawym należy uzyskać zmiksowane sygnały audio z wejść VIN1..VIN4, w kanale wyjściowym

lewym zmiksowane sygnały audio z wejść VIN5..VIN7 natomiast sygnał audio z wejścia VIN8 ma być dodawany do prawego i lewego kanału wyjściowego. Opisana konfiguracja wejść jest dosyć zawiła - sytuację taką ilustruje rys. 13. Wyraźnie widać, że uzyskanie takiej konfiguracji bez komputera byłoby skomplikowane, więc sterowanie komputerowe



Rys. 13. Przykładowa konfiguracja wejść audio miksera możliwa do uzyskania w programie sterującym.

List. 2. Procedura wysłania danych do miksera.

```

TabBajt: array[0..7] of Byte; {tablica z jedynekami }
.....

TabBajt[0] := 1;
TabBajt[1] := 2;
TabBajt[2] := 4;
TabBajt[3] := 8;
TabBajt[4] := 16;
TabBajt[5] := 32;
TabBajt[6] := 64;
TabBajt[7] := 128;
.....

const
  CentrPort = $378; { adres portu drukarkowego }
.....

procedure OutPort( PortWy: Word; BajtWy: Byte );
begin
  BajtWy := not BajtWy;
  asm
    mov dx,PortWy
    mov al,BajtWy
    out dx,al
  end; { asm }
end; { OutPort }

.....

procedure Send( kanal: Integer; wartosc: Integer );
{ D0 = CLK; D1 = DATA }
var
  B: Byte;
  i: Integer;
begin
  { obliczenie adresu }
  B := kanal or TabBajt[7]; { 1 na D7 }

  B := B or TabBajt[5]; { 1 na D5 }
  B := B or TabBajt[6]; { 1 na D6 }

  kanal := B;
  { wartość adresu do wysłania jest już obliczona }
  OutPort( CentrPort, 3 ); { 0 na D2 rozpoczyna zapis }
  for i:=7 downto 0 do
  begin
    B := kanal;

    if (B and TabBajt[i]) <> 0 then { czyli bit jest ustawiony }
      B := TabBajt[i] { 00000010 }
    else
      B := 0; { 00000000 }

    OutPort( CentrPort, B ); { wartość wysłana }

    B := B or TabBajt[0]; { 000000x1 - CLK zapis }
    OutPort( CentrPort, B ); { zbocze narasta }

    OutPort( CentrPort, 0 ); { zbocze opada }

  end;

  OutPort( CentrPort, TabBajt[2] ); { 1 na D2 kończy zapis bajtu }

  { obliczenie wartości do wysłania }
  B := wartosc and (not TabBajt[7]); { 0 na D7 }
  wartosc := B;
  { wartość do wysłania jest już obliczona }
  OutPort( CentrPort, 1 ); { 0 na D2 rozpoczyna zapis }
  for i:=7 downto 0 do
  begin
    B := wartosc;

    if (B and TabBajt[i]) <> 0 then { czyli bit jest ustawiony }
      B := TabBajt[i] { 00000010 }
    else
      B := 0; { 00000000 }

    OutPort( CentrPort, B ); { wartość wysłana }

    B := B or TabBajt[0]; { 00000011 - CLK zapis }
    OutPort( CentrPort, B ); { zbocze narasta }

    OutPort( CentrPort, 0 ); { 00000000 }
    { zbocze opada }

  end;
  OutPort( CentrPort, TabBajt[2] ); { 1 na D2 kończy zapis bajtu }

end; { wyslij }

```

mikserem jest w pełni uzasadnione zarówno ze względu na koszt urządzenia, jak i wygodę sterowania.

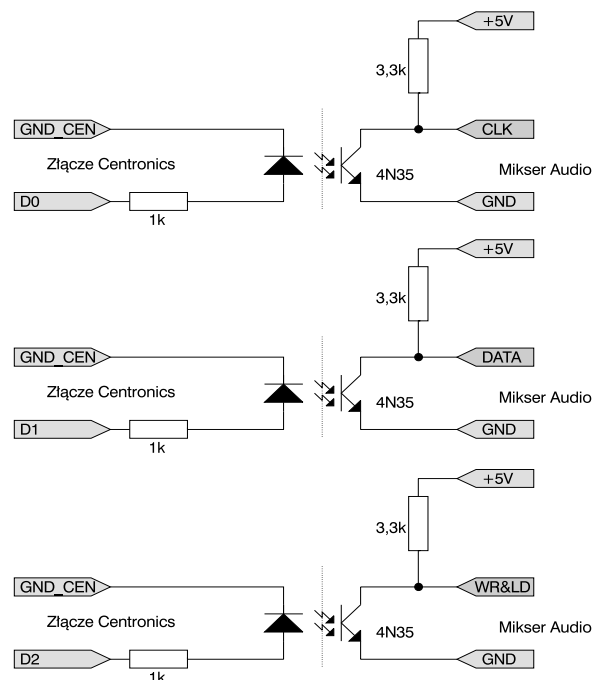
### Połączenie miksera z komputerem

Mikser jest sterowany komputerem poprzez port drukarkowy. Zdecydowano się na ten port ze względu na to, że nie wymagało to przebudowania układu elektrycznego miksera a zaimplementowanie protokołu wpisywania danych do układu SSM2163 jest łatwe. Przed połączeniem miksera z komputerem należy wyjąć z podstawki procesor US1. W połączeniu wykorzystano trzy bity portu drukarkowego, które są dołączone w następujący sposób:

- bit 0 rejestru danych - do wejścia CLK układu US2;
- bit 1 rejestru danych - do wejścia DATA układu US2;
- bit 2 rejestru danych - do wejścia WR&LD układu US2. Dodatkowo jako zabezpiecze-

- bit 2 rejestru danych - do wejścia !WRITE & !LOAD układu US2;

Znaczenie bitów 0..2 jest takie samo jak w przypadku sterowania układem SSM2163 za pomocą procesora US1. Ponieważ początkowo nie przewidywano komputerowego sterowania pracą miksera, na płycie drukowanej nie wyznaczono miejsc, do których można dołutować przewody łączące z portem drukarkowym. Z tego względu przewody te należy dołutować do odpowiednich wejść układu US2. Dodatkowo jako zabezpiecze-



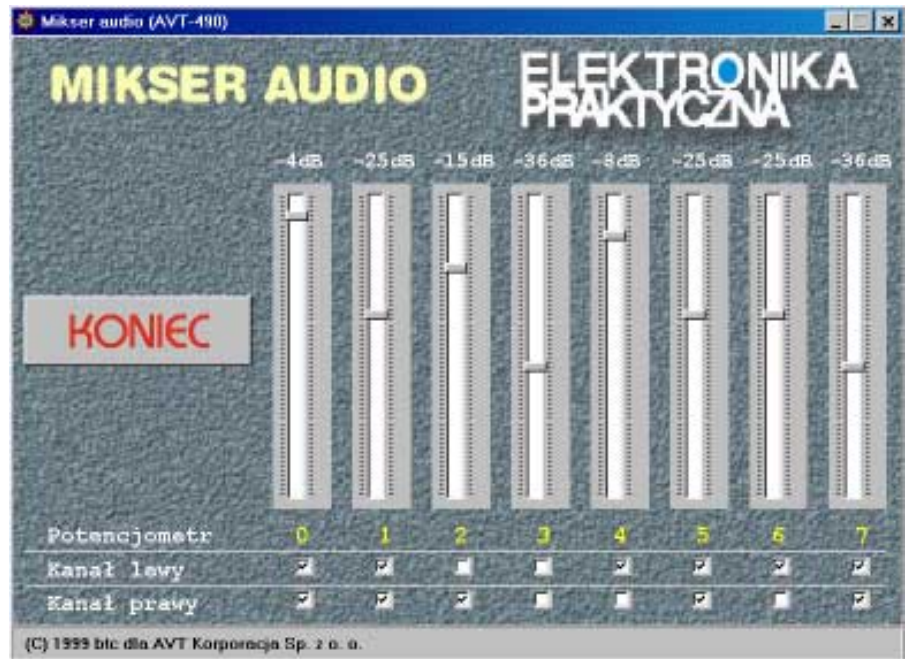
Rys. 14. Schemat elektryczny układu zabezpieczającego z transoptorami.

nie należy zastosować dodatkowy interfejs z transoptorami - schemat elektryczny takiego układu pokazano na rys. 14.

### Procedura wysyłania danych

Algorytm wysyłania danych jest taki sam jak w przypadku sterowania mikserem za pomocą mikroprocesora US1. Na listingu 2 pokazano przykładową implementację tego algorytmu w języku Delphi. W procedurze *Send* jest wykorzystywana procedura *Out-Port*, umożliwiająca wpisywanie dowolnej wartości do określonego portu sprzętowego komputera. Parametrami procedury *Send* są: numer kanału miksera (od 0 do 7) oraz wartość wzmacnienia w danym kanale (od 0 do 63).

Procedurę *Send* można podzielić na dwie części. Zadaniem pierwszej jest wysłanie do układu SSM2163 adresu kanału którego ustawienia chcemy zmienić. Najpierw jest obliczana wartość zmiennej *kanal*, zgodnie z tab. 1. Następnie wartość tej zmiennej jest szeregowo wysyłana poprzez port drukarkowy do układu US2. Druga część procedury wykonuje podobne operacje, lecz dotyczą one wartości wzmacnienia w danym kanale. Wartości te są zgodne z przedstawionymi w tab. 1 i tab. 2. Listing jest opatrzony obszernymi komentarzami, tak więc jego dokładniejsze omówienie nie jest konieczne. Wyjaśnienia może wymagać tylko sposób wykorzystania w procedurze tablicy *TabBajt*. W tablicy tej są przechowywane wartości, odpowiadające ustawieniu pojedynczych bitów w ramach bajtu (odpowiednie wartości widać na list. 2). Wartości te są wykorzystywane w operacjach na pojedynczych bitach. Przed wykonaniem połączenia miksera z komputerem należy wyjąć procesor US1 z podstawki.



Rys. 15. Widok okna przykładowej aplikacji sterującej pracą miksera.

### Opis przykładowej aplikacji

Do kitów w wersji „A” i „B” jest dołączana dyskietka z przykładową aplikacją umożliwiającą sterowanie mikserem za pomocą komputera. Na rys. 15 przedstawiono widok okna tej aplikacji. Przedstawiona na list. 2 procedura wysyłania danych do miksera jest fragmentem tego programu. Poziom wzmacnienia w poszczególnych kanałach ustala się przesuwając suwaki wirtualnych potencjometrów. Największe wzmacnienie uzyskuje się w górnym położeniu suwaków. Na list. 3 znajduje się procedura obsługująca zdarzenie *OnChange*, która jest wywoływana po każdym przesunięciu suwaka standardowego komponentu Delphi *TrackBar*. Położenie suwaka można odczytać z własności *Position* komponentu *TrackBar* i ta właśnie wartość (ustawiony zakres od 0 do 63) jest wykorzystywana do ustawiania

wartości wzmacnienia. Dodatkowo można ustalić, które z wejść audio ma być dołączone do wybranego kanału wyjściowego.

Dołączany do kitu program jest tylko przykładową aplikacją, ale pozwala w pełni sterować pracą miksera. Oczywiście programista może, korzystając z procedury *Send*, napisać inny program dostosowany do indywidualnych potrzeb. Ciekawy efekt (pojawiania się i zanikania dźwięku) można uzyskać ustalając wartość wzmacnienia w kanale audio za pomocą funkcji sinus. Wszystko zależy od inwencji programisty!

**Paweł Zbysiński**

Opisana w artykule przykładowa aplikacja sterująca pracą miksera jest dołączana do kitów w wersji „A” i „B”. Program ten jest również dostępny w internecie pod adresem: <http://www.ep.com.pl/ftp>.

List. 3. Procedura obsługująca zdarzenie *OnChange*.

```
procedure TMiks.TB0Change(Sender: TObject);
begin
  if TB0.Position = 0 then
    L_DB0.Caption := " 0dB"
  else
    L_DB0.Caption := "-" + IntToStr(TB0.Position) + "dB";

  Send( 0, TB0.Position );
end;
```

