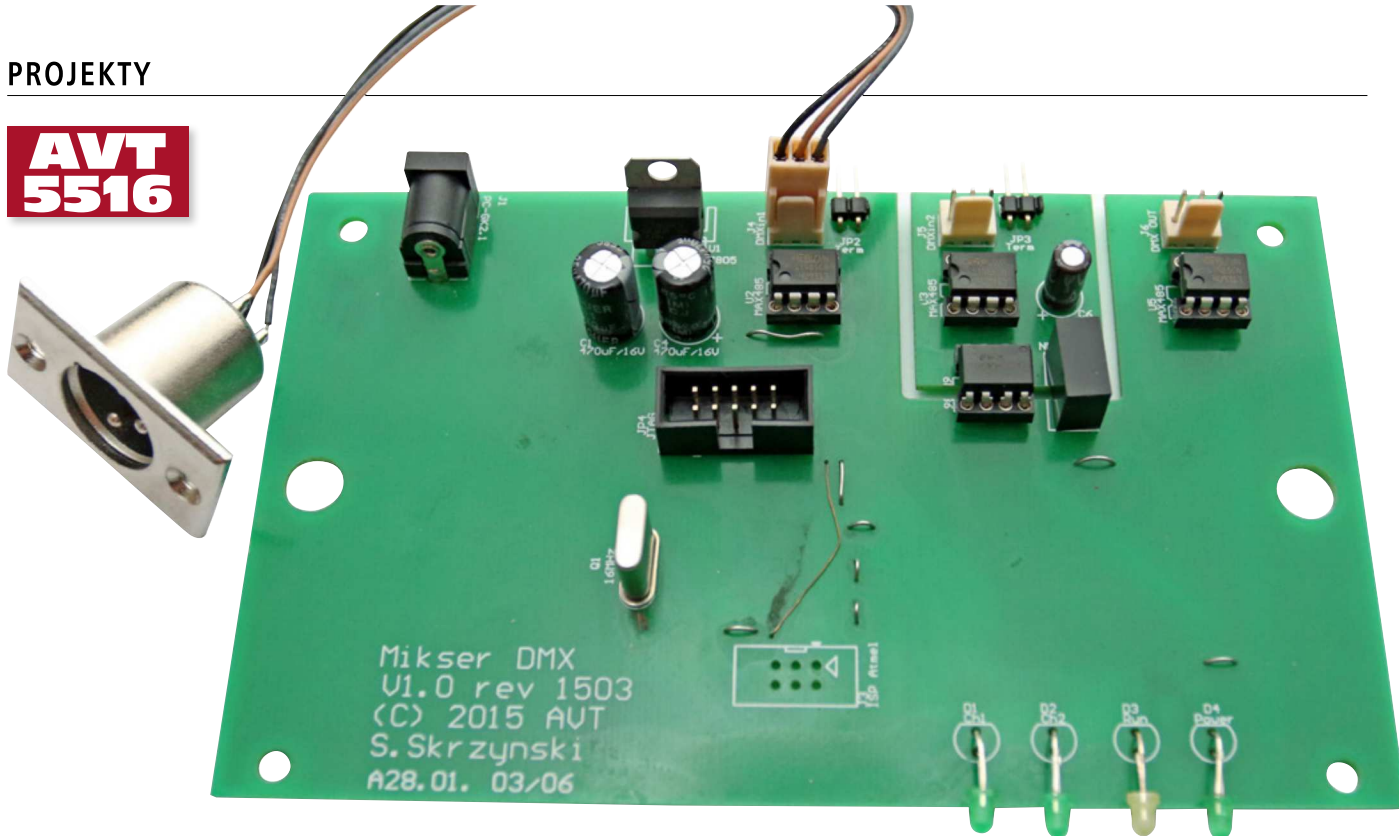


**AVT  
5516**


# Mikser DMX

Mikser odbiera strumień DMX z dwóch konsol, a następnie analizuje dane i transmituje zgodnie z zasadą: jeśli wartość kanału X konsoli A jest większa od B, to zostaną wysłane dane z konsoli A; jeśli wartość kanału X konsoli A jest mniejsza od B, to zostaną wysłane dane z konsoli B. Ze względu na zastosowany mikrokontroler jest obsługiwane „tylko” 256 kanałów.

**Rekomendacje:** kolejny projekt urządzenia, który jest przeznaczony dla osób zajmujących się techniczną oprawą imprez oraz scenografią. Umożliwia on dołączenie konsoli zapasowej lub zapewni równoległą obsługę wielu urządzeń za pomocą dwóch konsol.

Miksera jest uproszczoną wersją opisywanego w EP 12/2014 Mergera DMX. Podobnie jak w tamtym urządzeniu, są sumowane są dwa DMX, ale według innych reguł, dzięki czemu nie jest potrzebny impulsator i wyświetlacz LCD.

## Budowa i zasada działania

Napięcie zasilające 12 V jest stabilizowane za pomocą układu U1. Dzięki zastosowaniu mostka prostowniczego M1 polaryzacja napięcia nie ma znaczenia. Urządzenie można też zasilć napięciem przemiennym. Dane wejściowe konwertowane są w układach U2 i U3, po czym trafiają na wejścia UART a mikrokontrolera. Układ U3 jest zasilany za pośrednictwem przetwornicy DC/DC zbudowanej w oparciu o U7, a dane wysyłane do UART-a są separowane transoptorem U6. Dzięki temu jest zapewniona izolacja galwaniczna pomiędzy konsolami.

Przeważnie mikser będzie jedynym urządzeniem dołączonym do konsoli czy interfejsu komputera, dlatego zworki JP2 i JP3 powinny być założone. Dane wyjściowe z mikrokontrolera są konwertowane za pomocą U5. Jak widać, budowa urządzenia jest nieskomplikowana. Niestety, nie można tego powiedzieć o oprogramowaniu.

Procedury odbioru i nadawania pochodzą z mergera z małymi zmianami. Na **listingu 1** pokazano procedurę wysyłającą ramkę DMX. Istotne zmiany zaznaczono pogrubieniem.

Nie wiedzieć dlaczego, co bym nie wpisywał do rejestru UBRR, UART działał z jakąś dziwnie małą prędkością. Podglądałem wartość UBRR za pomocą debugera JTAG i była prawidłowa. Przypadkowo, manipulując w trybie debug rejestrem, czasem udawało się uzyskać prawidłową prędkość, ale tylko do restartu mikrokontrolera.

### W ofercie AVT\*

**AVT-5516 A**
**AVT-5516 UK**

### Podstawowe informacje:

- Zasilanie 230 V AC/maks. 2 VA.
- Dwa wejścia interfejsu DMX, jedno wyjście.
- Przyjmuje ramki DMX o różnych długościach.
- Przystosowany do zamontowania w obudowie KM-60.

### Dodatkowe materiały na FTP:

<ftp://ep.com.pl>, user: 82218, pass: aagt5gj6

- wzory płytek PCB

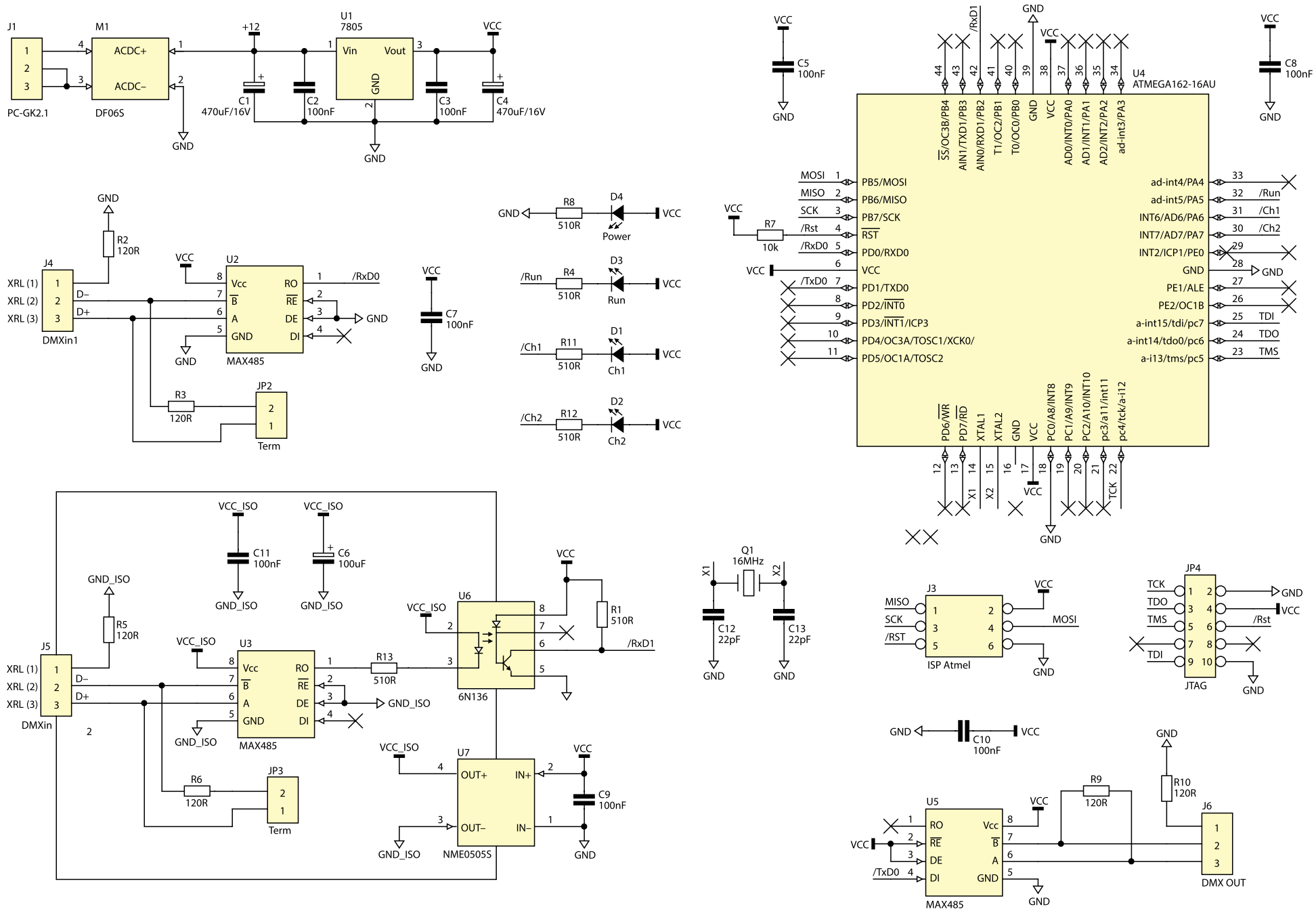
### Projekty pokrewne na FTP:

(wymienione artykuły są w całości dostępne na FTP)

- AVT-5512 Skaner DMX (EP 9/2015)
- AVT-5506 Lampa RGB z interfejsem DMX (EP 6/2015)
- AVT-5481 Merger DMX (EP 12/2014)
- AVT-5474 Demultiplexer DMX (EP 11/2014)
- AVT-5473 Multiplexer DMX (EP 11/2014)
- AVT-5462 DMX-owy sterownik serwomechanizmów (EP 8/2014)
- AVT-5456 Miniaturowa konsola z interfejsem DMX (EP 7/2014)
- AVT-5435 Sterownik DMX-RGB (EP 2/2014)
- AVT-5429 Transmisja DMX512 przez sieć Ethernet (EP 1/2014)
- AVT-5400 DMX Dimmer & Relay (EP 6/2013)
- AVT-3045 Switch DMX (EdW 12/2012)
- AVT-1632 Tester serwomechanizmów modelarskich (EP 8/2011)
- AVT-1605 Dwustanowy sterownik serwomechanizmu (EP 2/2011)
- AVT-5181 Sześciokanałowy dimmer z DMX512 (EP 4/2009)
- AVT-5129 Cyfrowy sterownik DMX512 (EP 4/2008)
- AVT-930 Konwerter USB-DMX512 (EP 5-6/2006)

\* Uwaga: Zestawy AVT mogą występować w następujących wersjach:  
 AVT xxxx UK to zaprogramowany układ. Tylko i wyłącznie. Bez elementów dodatkowych.  
 AVT xxxx A płytka drukowana PCB (lub płytki drukowane, jeśli w opisie wyraźnie zaznaczono), bez elementów dodatkowych.  
 AVT xxxx A+ płytka drukowana i zaprogramowany układ (czyli połączenie wersji A i wersji UK) bez elementów dodatkowych.  
 AVT xxxx B płytka drukowana (lub płytki) oraz komplet elementów wymienionych w załączniku pdf  
 AVT xxxx C to nic innego jak zmontowany zestaw B, czyli elementy wlotowane w PCB. Należy mieć na uwadze, że o ile nie zaznaczono wyraźnie w opisie, zestaw ten nie ma obudowy ani elementów dodatkowych, które nie zostały wymienione w załączniku pdf  
 AVT xxxx CD oprogramowanie (nieczęsto spotykana wersja, lecz jeśli występuje, to niezbędne oprogramowanie można ściągnąć, klikając w link umieszczony w opisie kitu)  
 Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik pdf! Podczas składania zamówienia upewnij się, którą wersję zamawiasz! (UK, A, A+, B lub C). <http://sklep.avt.pl>

Rysunek 1. Schemat ideowy Miksera DMX



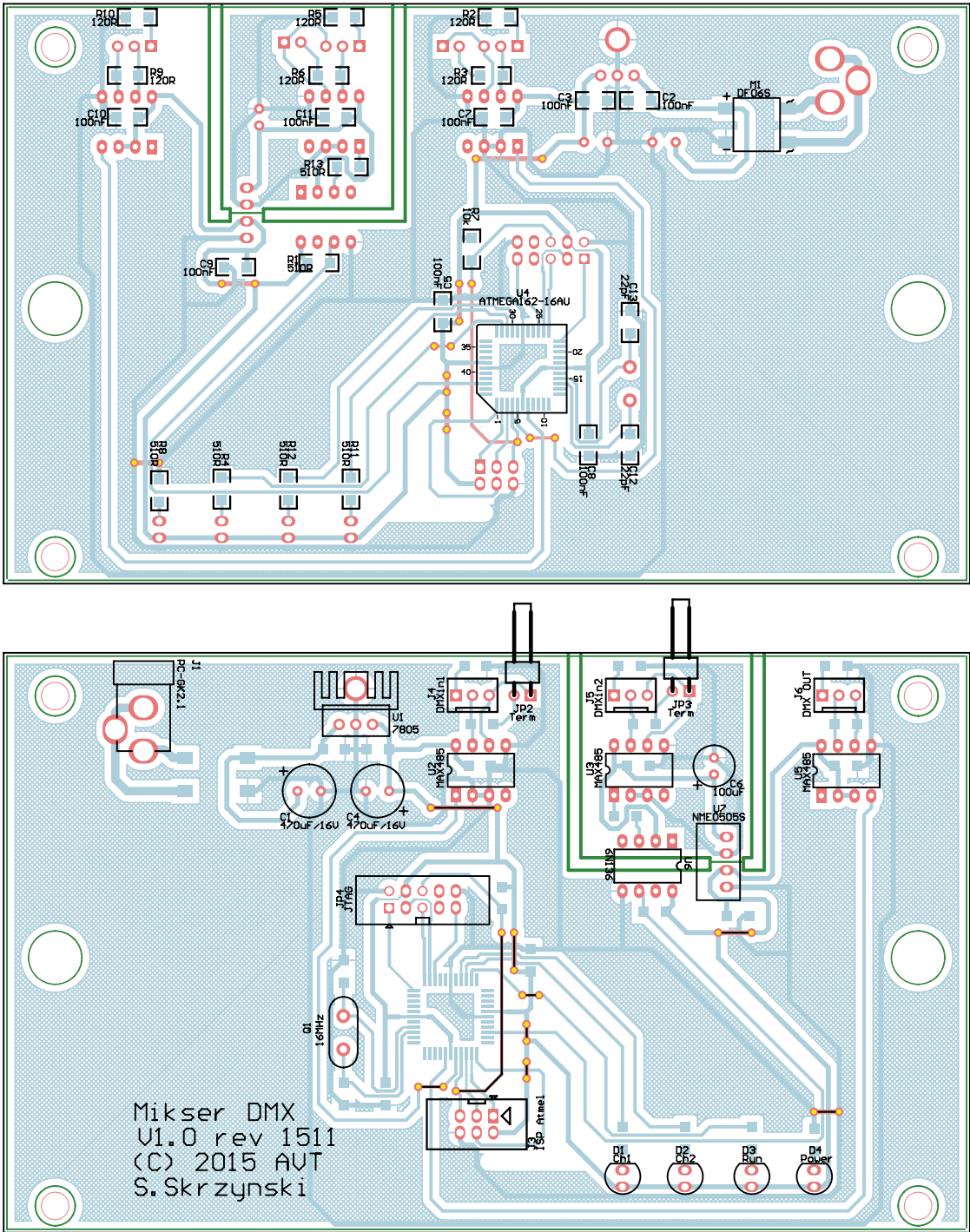
Początkowo uaktualniałem rejestr w przetwarzaniu, ale wtedy odbiorniki DMX (nie wicie jak pomocny okazał się być skaner DMX) nie synchronizowały się poprawnie. Przeniesienie uaktualniania rejestru do procedury generującej BREAK i MAB rozwiązało problem. Zapytacie: a co z UBRR w drugim kanale, gdzie jest tylko procedura odbioru? O tym za chwilę, bo chciałem dodać, że podobnie jak w Mergerze wysyłane jest jedynie tyle danych, ile potrzeba. Jeśli jedna konsola generuje dane dla 24 kanałów, a druga dla 128, wysłane zostanie 128 kanałów. Oczywiście, że modyfikacji zostaną poddane dane z 24 kanałów. Ponadto, przy braku

Listing 1. Procedura wysyłająca ramkę DMX

```
void SendDmx( word len )
{
    if (!TxEmpty){ return; }
    LenTxRs = len;
    if ( LenTxRs > LENBUFDMX ) LenTxRs = LENBUFDMX;
    TxEmpty = FALSE;
    UBRR0H = 0;
    UBRR0L = 3; // Dla 16Mhz
    // Program główny generuje BREAK i MAB po czym wyśle dane (wpisz do UDR0)
    UCSRB &= ~(1<<TXEN0); // wyłącz TX UARTA
    DdrTx(); // port wyjściem
    ClrTx(); _delay_us(200); // wygeneruj BREAK
    SetTx(); _delay_us(100); // wygeneruj MAB
    UCSRB |= (1<<TXEN0); // włącz TX UARTA
    UDR0 = TxBuf0[ PtrTxRs=0 ]; // wpisz do nadajnika pierwszy znak (start transmisji)
}
```

sygnału na którymś z kanałów, jest zerowana tablica, w której zapamiętane są dane. Dzięki temu wysyłane są dane z drugiego

aktywnego kanału. Jeśli nie będzie transmisji w żadnym kanale, to mikser przestanie wysyłać dane.



Rysunek 2. Schemat montażowy Miksera DMX

**Listing 2. Procedura odbierająca dane**

```

//-----
// Obsługa IRQ od odbiornika USART-a
//-----
//#define USART_RX_vect VECTOR(18) /* USART Rx Complete */
//#define USART_UDRE_vect VECTOR(19) /* USART, Data Register Empty */

//INTERRUPT(USART_RX_vect) // mozliwe przerwanie wielopoziomowe
//SIGNAL(USART1_RXC_vect) // niemozliwe przerwanie wielopoziomowe
{
    byte UsartStatus, UsartDana;
    word static CntDanych, curLenDmx, prevLenDmx;
    UsartStatus = UCSRIA;
    UsartDana = UDR1;
    if ( (UsartStatus & (1<<FE1)) ) // Jesli wykryto blad ramki to skonczyl
sie sygnal BREAK
    {
        TimerDmx1 = OVERTIME_DMx;
        DmxStatus1 = DMx_BREAK; // Ustaw status
        prevLenDmx = curLenDmx; // Zapamietaj poprzednia dlugosc
        curLenDmx = CntDanych; // Zapamietaj aktualna
        CntDanych = 0; // Zeruj licznik danych
        // Zapamietaj dlugosc danych gdy dwa wyniki takie same
        if ( prevLenDmx==curLenDmx ) LenDmx1 = curLenDmx-1;
        UBRR1H = 0;
        UBRR1L = 3; // Dla 16MHz
    }
    // Jesli byl BREAK a pojawila sie nowa dana:
    else if (DmxStatus1 == DMx_BREAK)
    {
        if ( CntDanych < LENBUFDmx ) RxBuf1[ CntDanych++ ] = UsartDana; //
nastepny kanal
    }
}

```

**Listing 3. Procedura analizująca dane przed wysłaniem**

```

//--- Wysłanie ramki DMX ---//
if ( TxEmpty ) // Jeśli poprzednia ramka wysłana
{
    for( cnt=0; cnt<LENBUFDmx; cnt++)
    {
        TxBuf0[ cnt ] = RxBuf0[ cnt ];
        if (RxBuf1[cnt]>RxBuf0[cnt]){TxBuf0[cnt]=RxBuf1[cnt];}
    }
    cnt = LenDmx0; if ( LenDmx1 > LenDmx0 ) { cnt = LenDmx1; }
    // Nadajemy tylko gdy jest transmisja na którymś porcie
    if ( (DmxStatus0 != DMx_OFFLINE) || (DmxStatus1 != DMx_OFFLINE) )
    {
        SendDmx( cnt+1 );
    }
    else
    {
        cnt = 0;
    }
}
}

```

Na listingu 2 pokazano procedurę odbierającą dane – pogrubiono na nim zmiany w porównaniu do tej z Mergera DMX. Na listingu 3 zamieszczono procedurę analizującą dane przed wysłaniem. Pogrubienie ukazuje istotny fragment analizy danych.

**Montaż i uruchomienie:**

Schemat montażowy Miksera DMX pokazano na rysunku 2. Montaż jest typowy i nie wymaga omawiania. Należy pamiętać o zamontowaniu kilkunastu zworek. Po włączeniu

zasilania dioda D4 powinna świecić, natomiast D3 migać. Pod układy MAX485 warto zastosować podstawki. Jeśli procesor nie jest zaprogramowany można to zrobić przez złącze JP4 (JTAG) lub J3 (SPI). Ustawienie bitów konfiguracyjnych pokazano rysunku 3. Płytkę miksera jest przystosowana do obudowy KM-50.

**Obsługa**

Urządzenie nie wymaga regulacji ani ustawiania. Wystarczy podłączyć konsolę

**Wykaz elementów****Rezystory:** (SMD 1206)

R1, R4, R8, R11...R13: 510 Ω  
R2, R3, R5, R6, R9, R10: 120 Ω  
R7: 10 kΩ

**Kondensatory:**

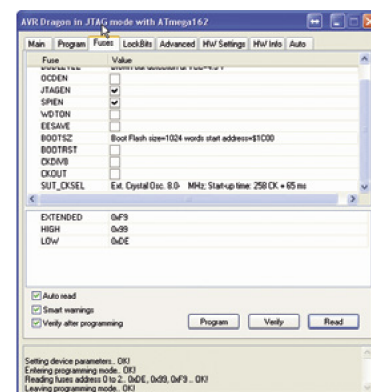
C1, C4: 470 μF/16 V (elektrolit.)  
C6: 100 μF/16 V (elektrolit.)  
C2, C3, C5, C7...C11: 100 nF  
C12, C13: 22 pF

**Półprzewodniki:**

U1: 7805  
U2, U3, U5: MAX485  
U4: ATmega162-16AU  
U6: 6N136  
U7: NME0505S  
M1: DF06S (mostek prostowniczy)  
D1, D2, D4: dioda LED 5 mm, zielona  
D3: dioda LED 5 mm, żółta

**Pozostałe:**

Q1: 16 MHz kwarc  
J1: gniazdo zasilające 5,5/2,1 kątowne  
J4, J5: gniazdo NS25 3 pin, wtyk NS25 3 pin, 3 szt. terminali do wtyku NS25, wtyk XRL-3 do obudowy  
J6: gniazdo NS25 3 pin, wtyk NS25 3 pin, 3 szt. terminali do wtyku NS25, gniazdo XRL-3 do obudowy  
JP4: IDC10 (gniazdo „wannowe” 2×5)  
JP2, JP3: listwa kątowna goldpin + zworki  
PPIN8: podstawka precyzyjna 8 pin – 3 szt.



Rysunek 3. Ustawienie bitów konfiguracyjnych mikrokontrolera

do wejść, a odbiorniki do wyjścia. Aktywność kanałów jest sygnalizowana diodami LED. Na serwerze FTP EP dostępna jest wersja źródłowa i wynikowa programu.

**Sławomir Skrzyński, EP**

REKLAMA

# ELEKTRONIKA PRAKTYCZNA

Zaprenumeruj na stronie AVT.pl, e-mail: [prenumerata@avt.pl](mailto:prenumerata@avt.pl)  
lub telefonicznie pod numerem: 22 257 84 99  
Bieżący numer zamów na [www.ulubionykiosk.pl](http://www.ulubionykiosk.pl)

