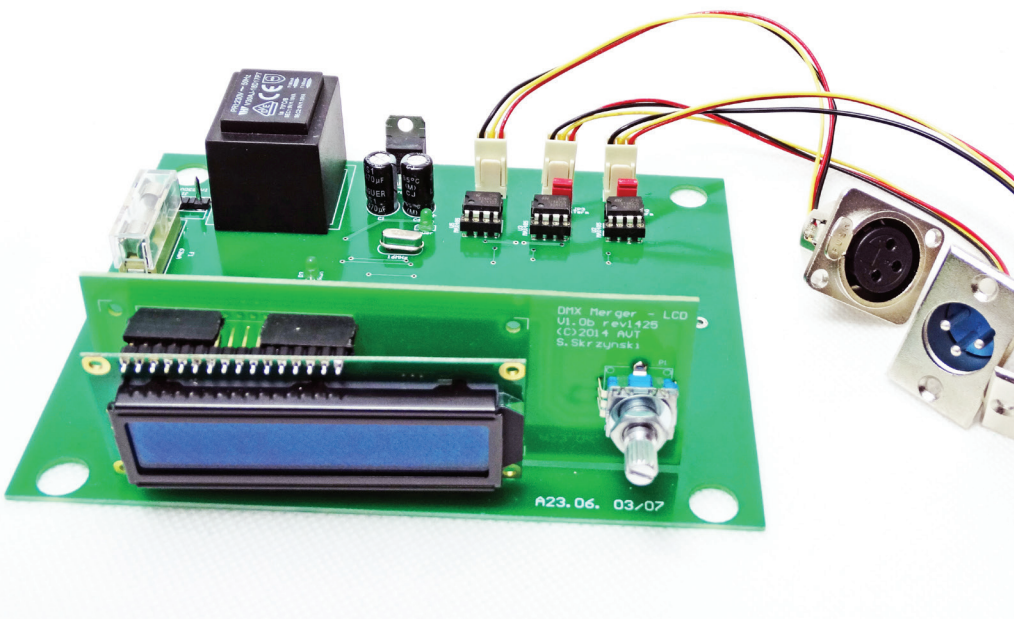
Merger DMX

AVT 5481

W systemach DMX nierzadko jest konieczne dołączenie do sieci więcej niż jednej konsoli, choćby ze względu na zwiększenie liczby obsługiwanych urządzeń. Nie można po prostu połączyć równolegle ich zacisków wyjściowych, bo konsole będą się wzajemnie zakłócały. Konieczne jest zastosowanie mergera.

***Rekomendacje:** urządzenie jest kontynuacją publikacji związanych z systemem DMX – przyda się osobom zajmujących się aranżacją i obsługą techniczną sceny.*



Budowa mergera nie jest tak prosta, jak opisywanego w poprzednim numerze EP splittera, w którym wystarczyło „rozdzielić" sygnał na kilka wyjść, co daje się łatwo zrobić z użyciem kilku driverów MAX485. W mergerze trzeba zsumować dwa niezsynchronizowane ze sobą strumienie DMX. Każdy ze strumieni może przenosić od 24 do 512 bajtów danych. Funkcjonalność taką można zrealizować z użyciem mikrokontrolera mającego dwa interfejsy UART. Wskazane, aby można było ustalić, jaką część strumienia wykorzystują poszczególne wejścia, co umożliwi połączenie kaskadowe wielu mergerów. Do skonfigurowania takiego urządzenia, co prawda, wystarczyłyby zworki, ale większe możliwości daje impulsator i wyświetlacz alfanumeryczny.

Budowa i zasada działania

Schemat ideowy projektu mergera DMX pokazano na **rysunku 1**. Napięcie zasilające 230 V AC jest obniżane w transformatorze, następnie prostowane za pomocą diod D1 i D2, po czym stabilizowane przez układ U1. Dane wejściowe konwertowane są w układach U2 i U3, a następnie trafiają na wejścia UART mikrokontrolera. Przeważnie merger będzie jedynym urządzeniem podłączonym do konsoli czy interfejsu komputera, dlatego zworki JP2 i JP3 powinny być założone. Przekształcone dane wyjściowe z mikrokontrolera są konwertowane za pomocą U5. Mikrokontroler steruje także alfanumerycznym wyświetlaczem LCD oraz odczytuje informacje z impulsatora. Wyświetlacz i impulsator umieszczono na osobnej płytce, zamontowanej pod kątem 90 stopni w stosunku do płyty głównej.

Jak widać, budowa urządzenia jest nieskomplikowana, program już niestety taki nie jest. Oprogramowanie mikrokontrolera odbiera dane z obu interfejsów USART i zapamiętuje je w tablicy. Obsługa USART odbywa się z użyciem przerwań. Realizuje to procedura (dla USART0) pokazana na **listingu 1**.

Przy odbiorze danych z USART z użyciem mechanizmu przerwań należy pamiętać, aby zadeklarować przerwanie jako „Signal". W przeciwnym wypadku dojdzie do przepełnienia stosu, gdyż znacznik przerwania jest kasowany po zakończeniu jego obsługi, a nie po wejściu w przerwanie, tak jak to się dzieje w wypadku przerwań od timerów i innych peryferii. Jeśli konieczne byłoby przerwanie wielopoziomowe, należy postąpić w następujący sposób:

- Zadeklarować przerwanie jako „Signal".
- Odblokować przerwania za pomocą „sei()".
- Obsłużyć przerwanie.

Przerwania nie trzeba odblokowywać przy wychodzeniu z procedury jego obsługi. Dzieje się to automatycznie przy odtwarzaniu rejestru CREG.

Procedura odbiorcza jest zabezpieczona przed przepełnieniem bufora. Jak wspomniano, konsola może wysyłać od 24 do 512 bajtów danych. W zmiennej *curLenDmx* znajduje się liczba odebranych danych. Gdy dwa razy zostanie odebrana taka sama ich liczba (zmienna *prevLenDmx*), jest ona zapamiętywana w *LenDmx0* dla USART0 lub *LenDmx1* dla USART1.

W ofercie AVT*

AVT-5481 A	AVT-5481 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: 63172, pass: 428ofq53

- wzory płytek PCB

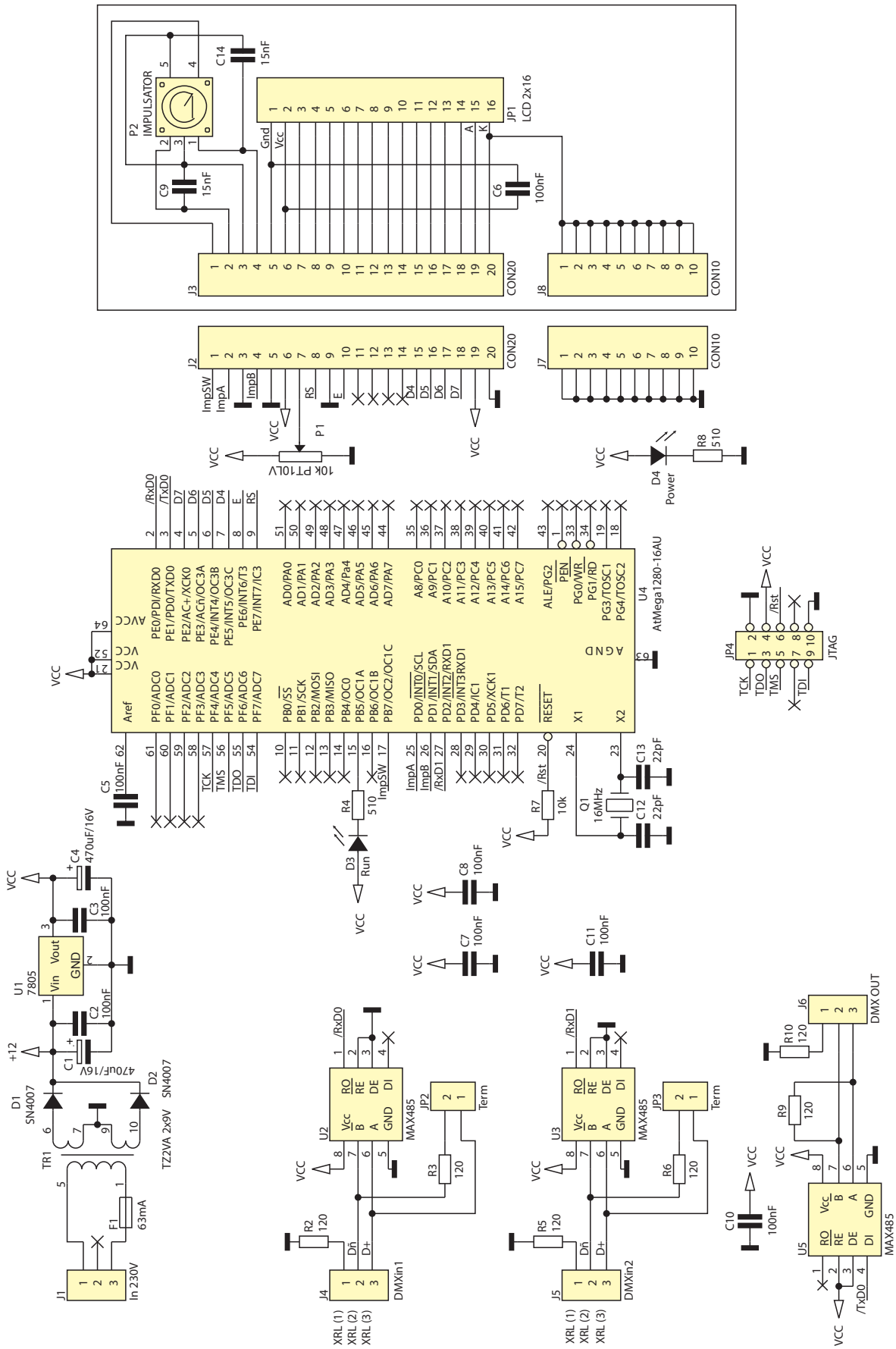
Projekty pokrewne na FTP:
(wymienione artykuły są w całości dostępne na FTP)

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)

* 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 wlutowane 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*

Procedura odbioru danych z USART1 jest taka sama, jak z USART0. Różni się tylko nazwami rejestrów. Odebrane dane są kopiowane do bufora nadawczego. Realizuje to procedura pokazana na **listingu 2**.

Zależnie od zmiennej *podzial* jest kopiowana część pierwszego bufora odbiorczego do nadawczego. Następnie, drugi bufor odbiorczy jest kopiowany do nadawczego za danymi pierwszego bufora. Jeśli długość przekroczyłaby maksymalna liczbę da-



Rysunek 1. Schemat ideowy Mergera DMX

Listing 1. Procedura obsługi przerwania od USART0
```
// Obsługa IRQ od odbiornika USART
SIGNAL(USART0_RX_vect)
{
    byte UsartStatus, UsartDana;
    word static CntDanych, curLenDmx, prevLenDmx;

    UsartStatus = UCSR0A;
    UsartDana = UDR0;
    UCSR0A = 0;

//Jeśli wykryto błąd ramki, to skończył się sygnał BREAK
    if ((UsartStatus & (1<<FE0)))
    {
        TimerDmx0 = OVERTIME_DMX;
        DmxStatus0 = DMX_BREAK;  //Ustaw status
        prevLenDmx = curLenDmx;  //Zapamiętaj poprzednią długość
        curLenDmx = CntDanych;  // Zapamiętaj aktualną
        CntDanych = 0;  // Zeruj licznik danych
// Zapamiętaj długość danych gdy dwa wyniki takie same
        if (prevLenDmx==curLenDmx) LenDmx0 = curLenDmx;
    }
        else
// Jesli byl BREAK a pojawila sie nowa dana:
    if(DmxStatus0 == DMX_BREAK)
    {
        // nastepny kanal
        if (CntDanych < LENBUFDMX) RxBuf0[CntDanych++] = UsartDana;
    }
}
```

Listing 2. Kopiowanie danych do bufora nadawczego
```
//--- Wysłanie ramki DMX ---//
if ( TxEmpty )  // Jeśli poprzednia ramka wysłana
{
    d = 1;  // Kopiujemy od bajta 1, bo 0 zawsze równy 0
    newLen = podzial + LenDmx1;
    if ( newLen > 513 ) newLen = 513;  // Wyliczmy nową długość
    for (s=1; s<podzial; s++)
        TxBuf0[ d++ ] = RxBuf0[ s ];
    for (s=1; s<513; s++)
    {
        TxBuf0[ d++ ] = RxBuf1[ s ];
        if ( d > LENBUFDMX) break;  // Kontrola długości
    }
    if((DmxStatus0 != DMX_OFFLINE)||(DmxStatus1 != DMX_OFFLINE))
        //Nadajemy tylko, gdy jest transmisja na którymś porcie
        SendDmx( newLen );
    else newLen = 0;
}
```

Listing 3. Generowanie sygnałów BREAK i MAB, wysłanie danych
```
void SendDmx( word len )
{
    LenTxRs = len;
    if ( LenTxRs > LENBUFDMX ) LenTxRs = LENBUFDMX;
    TxEmpty = FALSE;
// Program główny generuje BREAK i MAB po czym wyśle dane (wpis do UDR0)
    DdrTx();  // Port wyjsciem
    UCSR0B &= ~(1<<TXEN0);  // wyłącz TX UARTA
    ClrTx(); _delay_us(200);  // wygeneruj BREAK
    SetTx(); _delay_us(88);  // wygeneruj MAB
    UCSR0B |= (1<<TXEN0);  // włącz TX UARTA
    UDR0 = TxBuf0[ PtrTxRs=0 ];  // Wpisz do nadajnika pierwszy znak (start
transmisji)
}
```

Listing 4. Obsługa przerwania USART – wysłanie danych
```
//  Przerwanie nadawcze od USART
#define USART_UDRE_vect    _VECTOR(19)  /* USART, Data Register Empty */
#define USART_TX_vect      _VECTOR(20)  /* USART Tx Complete */
SIGNAL(USART0_TX_vect)  // Musi byc ISR lub SIGNAL a nie INTERUPT
{
    PtrTxRs++;  //Wskaźnik na następny znak
    if (PtrTxRs < LenTxRs)  //Jeśli są jeszcze znaki do wysłania
        UDR0 = TxBuf0[ PtrTxRs ];  // Wpisz do nadajnika
    else
        TxEmpty = TRUE;  // Wszystkie znaki wysłane (ostatnie IRQ)
}
```

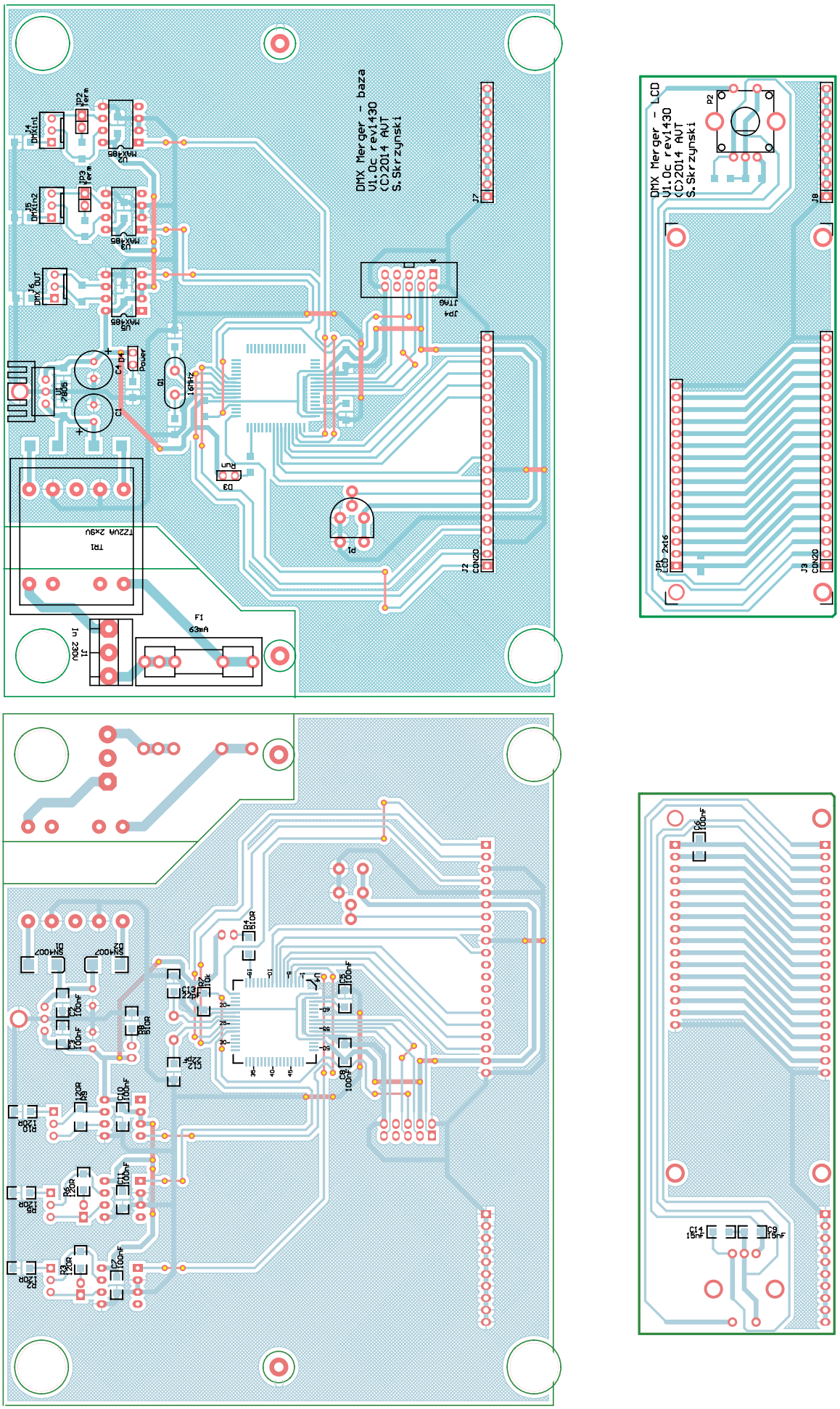
Listing 5. Procedura obsługi impulsatora
```
//Obsługa IRQ od wejścia INT 0
SIGNAL( INT0_vect )  // NIEMOZLIWE przerwanie wielopoziomowe (brak instrukcji
SEI)
{
    byte data;
    data = IMPdata;
    if ( RdImpB ) data++;
    else data--;
    data &= IMPdataMASK;  // ograniczenie zakresu
    IMPdata = data;
}
```

nych, jest ona ograniczana. Jeśli jest mniejsza, to po wyliczeniu jest zapamiętywana w zmiennej *newLen*. Procedura transmisyjna korzysta z tej zmiennej, dzięki czemu nie wysyła niepotrzebnych danych, co zwiększa częstotliwość odświeżania. Dane wysyłane są z użyciem przerwań, jednak najpierw generowane są sygnały BREAK i MAB (**listing 3**).

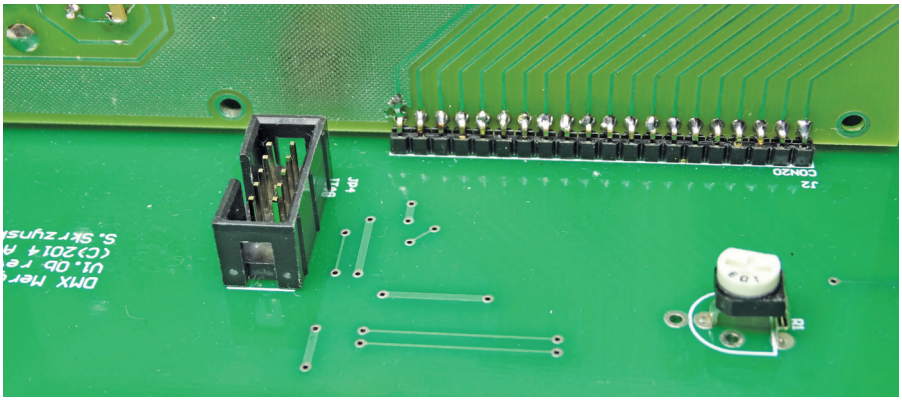
Ponadto, ta procedura wpisuje pierwszy bajt do USART, co powoduje jego wysłanie

Wykaz elementów

Rezystory: (SMD 1206)
R2, R3, R5, R6, R9, R10: 120 Ω
R4, R8: 510 Ω
R7: 10 kΩ
Kondensatory:
C1, C4: 470 µF/16 V (elektrolit.)
Kondensatory SMD 1206:
C2, C3, C5…C8, C10, C11: 100 nF (SMD 1206)
C9, C14: 15 nF (SMD 1206)
C12, C13: 22 pF (SMD 1206)
Półprzewodniki:
U1: 7805
U2, U3, U5: MAX485
U4: ATmega1280-16AU
D1, D2: SN4007
D3: dioda LED żółta, 3 mm
D4: dioda LED, zielona, 3 mm
Inne:
P1: 10 kΩ (PT10LV pot. montażowy)
P2: ED1212S-24C24-30F (impulsator z włącznikiem, długa oska)
Q1: 16 MHz (kwarc HC49/HC49S)
F1: F63 mA (bezpiecznik 5×20)
J1: TB-5.0-PP-2P, TB-5.0-PIN (złącze TB z listwą kołkową)
J3: listwa kątowa goldpin 1×20
J8: listwa kątowa goldpin 1×10
J4, J5: NS25-W3 (gniazdo NS25 3 pin), NS25-G3 (wtyk NS25 3 pin), NS25-T (3 szt. terminali do wtyku NS25)
XLR-3G-C: wtyk XRL-3 do obudowy
J6: NS25-W3 (gniazdo NS25 3 pin)
NS25-G3: wtyk NS25 3 pin
NS25-T: 3 szt. terminali do wtyku NS25
XLR-3G-CH: gniazdo XRL-3 do obudowy
JP4: IDC10MLP (gniazdo „wannowe" 2×5)
JP1: LCD 2×16 (wyświetlacz z listwą goldpin + gniazdo do płytki)
TR1: TZ2VA 2×9 V (transformator sieciowy, zalewany 2×9 V/100 mA)
PPIN8: podstawka precyzyjna 8 pin (3 szt.)



Rysunek 2. Schemat montażowy Mergera DMX



Fotografia 3. Sposób zamontowania płytek drukowanych

i wygenerowanie przerwania. Pozostałe dane są wysyłane z wykorzystaniem mechanizmu przerwań (**listing 4**).

Procedura obsługi impulsatora została zrealizowana z użyciem mechanizmu przerwań, dzięki czemu jest nieskomplikowana. Pokazano ją na **listingu 5**.

Montaż i uruchomienie

Schemat montażowy Mergera DMX pokazano na **rysunku 2**. Montaż jest typowy i nie wymaga omawiania. Jeśli płytki wykonujemy samodzielnie, to płytę główną można wykonać jako jednowarstwową. Wtedy jest konieczne zamontowanie kilkunastu zwór z drutu. Płytkę wyświetlacza i impulsatora montujemy pod kątem 90 stopni w stosunku do płyty bazowej, za pośrednictwem goldpina kątowego. Należy zwrócić uwagę na jego typ. Szczegóły można zobaczyć na **fotografii 3**. Złącze J8 zastosowano tylko w celu wzmocnienia mechanicznego płytki. Wyświetlacz montujemy na gnieździe goldpin. Po włączeniu zasilania dioda D4 powinna świecić natomiast D3 migać z częstotliwością około 1 Hz. Na wyświetlaczu powinien pojawić się ekran powitalny. Jeśli nic nie widać lub kontrast jest za mały, regulujemy go potencjometrem P1. Pod układy MAX485 warto zastosować podstawki. Jeśli procesor nie jest zaprogramowany można to zrobić przez złącze JP4 (JTAG). Ustawienie bitów konfiguracyjnych pokazano na **rysunku 4**. Płytka Mergera DMX jest przystosowana do zamontowania w obudowie KM-60.

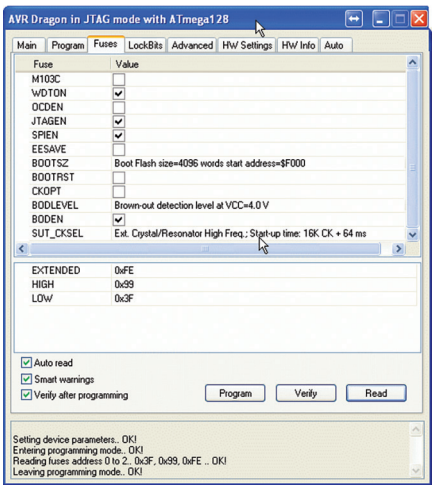
Obsługa

Po włączeniu zasilania pojawi się ekran powitalny, po czym ukaże się ekran roboczy, a na nim komunikat:

```
1: aaa*  2: bbb*
P: ccc   N: ddd
```

Poszczególne symbole oznaczają:

aaa – liczba odebranych danych na pierwszym wejściu DMX, gwiazdka za liczbą oznacza trwającą transmisję (on-line), znak minus, brak transmisji przez co najmniej 1 sekundę (off-line).



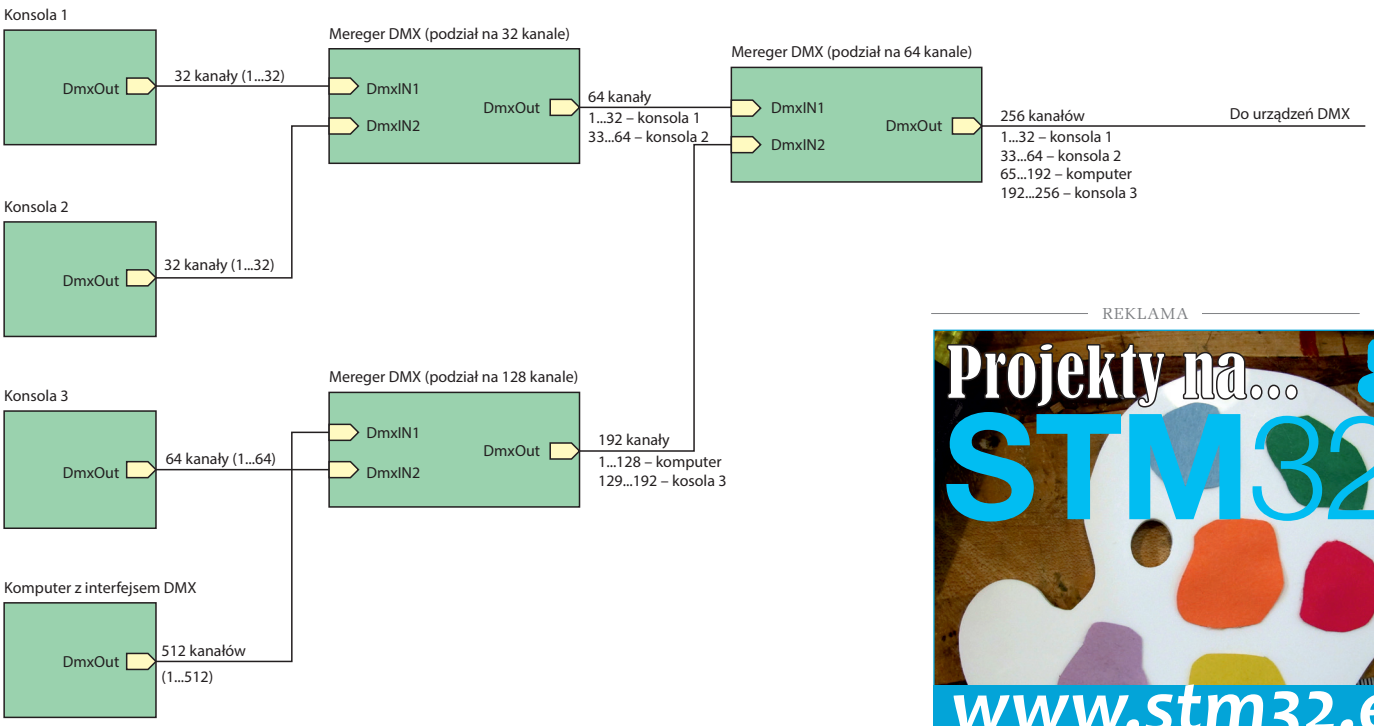
Rysunek 4. Ustawienie ważniejszych fusebitów

bbb – jak wyżej, ale dla wejścia drugiego.
ccc – liczba danych, które zostaną wysłane z pierwszego kanału na wyjście; za nimi zostaną przesłane dane z drugiego kanału (miejsce podziału bufora nadawczego).
ddd – liczba danych wysyłanych na wyjście.

Porada: urządzenie, które wysyła dłuższe ramki powinno być włączone do pierwszego kanału. Dzięki temu, zależnie od ustawienia mergera i liczby danych odbieranych w drugim kanale, ‚merger będzie wysyłał mniej niż 512 bajtów danych, co zwiększy częstotliwość odświeżania urządzeń odbiorczych.

Kręcąc impulsatorem zmieniamy liczbę danych wysyłanych z pierwszego kanału. Tę liczbę można zmieniać z krokiem 16. Aby zapamiętać nastawę należy nacisnąć impulsator. Przykładowy sposób włączenia meregerów można zobaczyć na **rysunku 5**.

Sławomir Skrzyński, EP



Rysunek 5. Przykładowy sposób włączenia mergerów