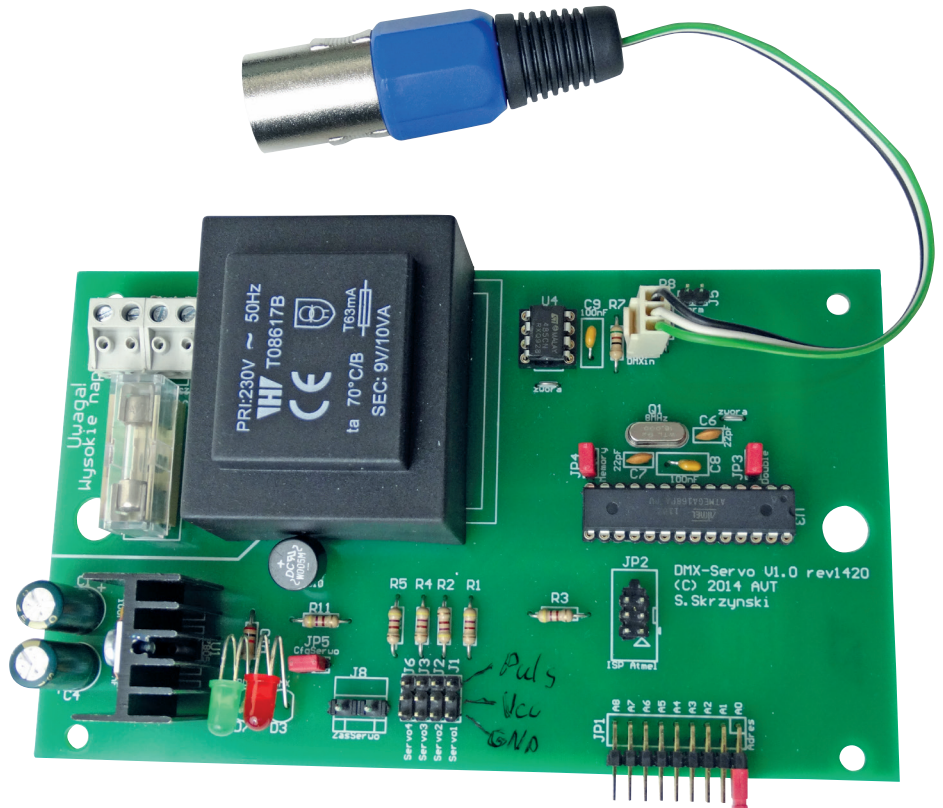


# DMX-owy sterownik serwomechanizmów



Projekt jest kontynuacją urządzeń z interfejsem DMX przeznaczonych dla scenografów, osób zajmujących się oprawą imprez, muzyków itp. Nasz sterownik potrafi kontrolować cztery serwomechanizmy, za pomocą których można na przykład sterować oświetleniem lub animowanymi elementami scenografii. Po zaniku transmisji serwomechanizmy mogą zostać w ostatnim stanie lub przyjąć wcześniej zaprogramowaną pozycję domyślną.

**Rekomendacje:** serwo-roboty przyda się do poruszania obiektów na scenie, przełączania filtrów lub przesłon w reflektorach, kierowania nimi lub sterowaniem zaworami.



Schemat ideowy sterownika serwomechanizmów z interfejsem DMX pokazano na rysunku 1. Napięcie sieciowe jest obniżane w transformatorze, następnie prostowane za pomocą mostka prostowniczego. Na potrzeby zasilania układów scalonych zastosowano stabilizator U1 dający na wyjściu napięcie +5 V. Może on rów-

nież zasilac serwomechanizmy (założona zwora JP5), ale trzeba mieć świadomość, że zależnie od typu i obciążenia, **serwomechanizmy mogą pobierać prąd większy niż 1 A**. W takiej sytuacji należy podać odpowiednie napięcie zasilające, na przykład z zakresu 4...6 V, o odpowiedniej wydajności prądowej do złącza J8 oraz zdjąć

**W ofercie AVT\***  
 AVT-5462 A  
 AVT-5462 B

- Podstawowe informacje:**
- Napięcie zasilające: 230 V AC/10 VA.
  - Możliwość zasilenia serwomechanizmów z zewnętrznego źródła napięcia zasilającego lub z wbudowanego stabilizatora +5 V.
  - Możliwość kontrolowania 4 serwomechanizmów.
  - Położenie serwomechanizmów w wypadku zaniku transmisji: domyślne lub ostatnio ustawione (konfigurowane).
  - Interfejs DMX, adres konfigurowany za pomocą zworek.
  - Mikrokontroler ATmega168.

**Dodatkowe materiały na FTP:**  
<ftp://ep.com.pl>, user: 26526, pass: 841uhx54

• wzory płytek PCB

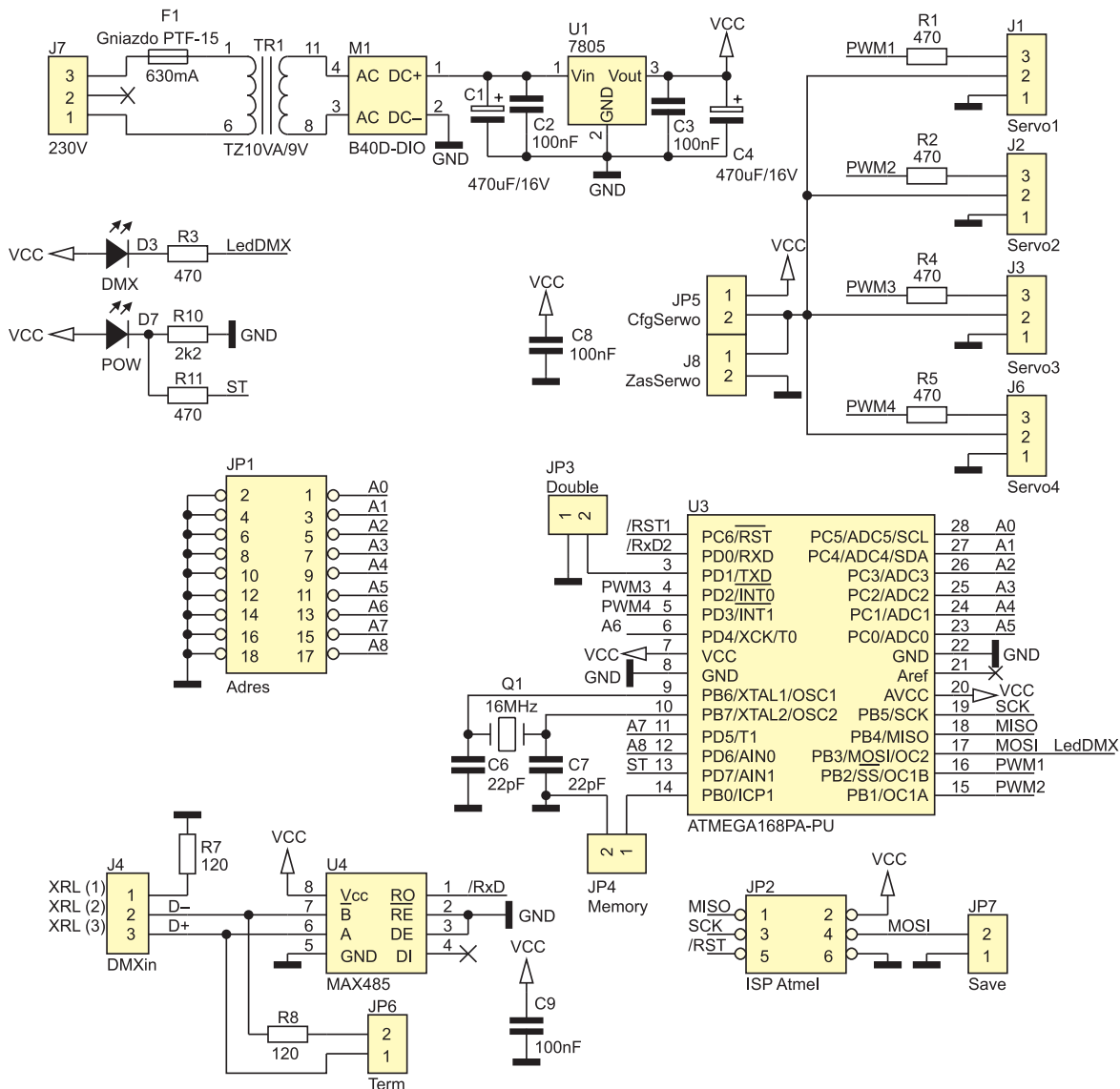
Projekty pokrewne na FTP:

- (wymienione artykuły są w całości dostępne na FTP)
- 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-930 Konwerter USB-DMX512 (EP 5-6/2006)

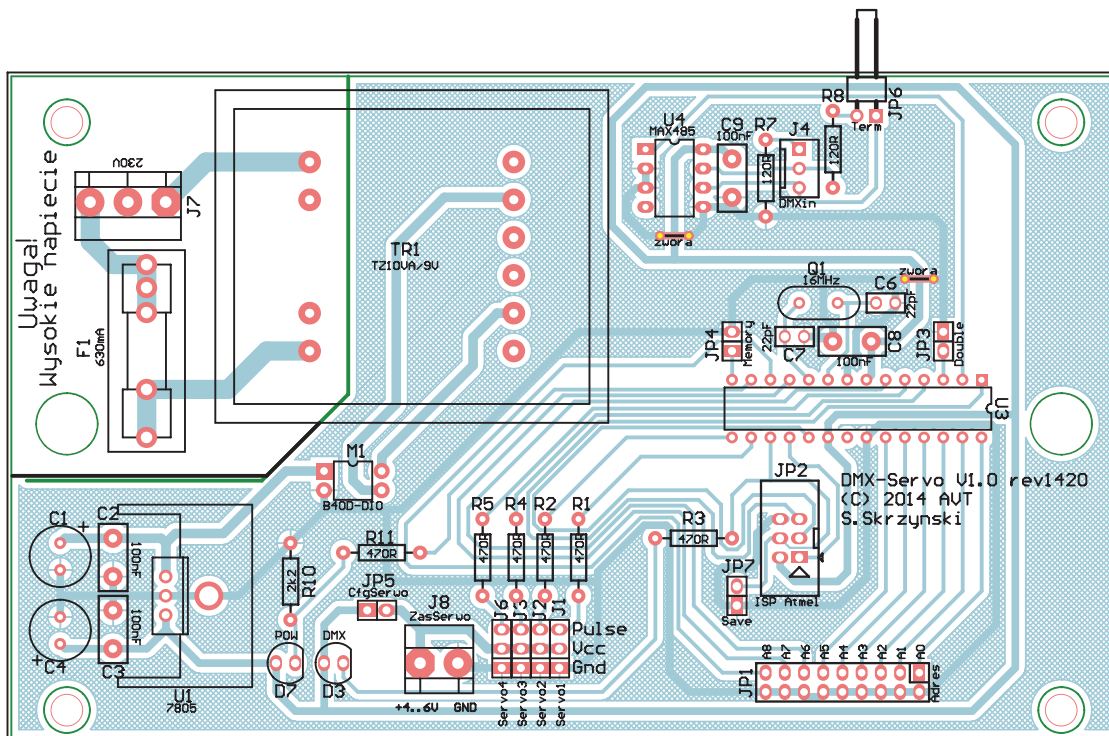
\* Uwaga: Zestawy AVT mogą występować w następujących wersjach:  
 AVT xxxx UK to zaprogramowany układ. Tylko i wyłącznie. Bez elementów dodatkowych.  
 AVT xxxx A płytka drukowana PCB (lub płytki drukowane, jeśli w opisie wyraźnie zaznaczono), bez elementów dodatkowych.  
 AVT xxxx A+ płytka drukowana 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>

Oznaczenie elementu	Oznaczenie	Funkcja
D7	POW	Słabym świeceniem sygnalizuje obecność napięcia zasilającego. Rozbłyska jasno po co drugim odebrany BREAK i SC Miga wolno po zapisie ustawień serw w eeprom
D3	DMX	Świeci jeśli poprawne zdekodowanie DMX (wykryto BREAK i SC)

Oznaczenie	Funkcja
JP1 (ADRES)	Adresy od 1 do 512 (512-brak zworek)
JP4 (Memory)	Brak zworki: przy braku transmisji serwomechanizmy pamiętają ostatni stan Zwórka założona: przy braku transmisji serwomechanizmy zostaną ustawione zgodnie z danymi zapamiętanymi w EEPROM
JP3 (Double)	Brak zworki: dwie identyczne transmisje wywołują zmianę (większa odporność na błędy, ale wolniejsza reakcja na zmiany) Zwórka założona: pojedyncza transmisja jest akceptowana (mniejsza odporność na błędy, ale szybsza reakcja na zmiany)
JP7 (Save)	Założenie zworki zapisuje aktualne ustawienie serwomechanizmów do EEPROM. Podczas zapisu EEPROM serwomechanizm nie są obsługiwane. Po zaniku transmisji, gdy zwora JP4 (Memory) jest założona, ustawienia serwomechanizmów zostaną odtworzone z EEPROM.



Rysunek 1. Schemat ideowy sterownika serwomechanizmów z interfejsem DMX



Rysunek 2. Schemat montażowy sterownika serwomechanizmów z interfejsem DMX

Wykaz elementów

Rezystory:

R1...R5, R11: 470 Ω  
R7, R8: 120 Ω  
R10: 2,2 kΩ

Kondensatory:

C1, C4: 470 μF/16 V (elektrolit.)  
C2, C3, C8, C9: 100 nF (ceram.)  
C6, C7: 22 pF (ceram.)

Półprzewodniki:

U1: 7805  
U3: ATmega168PA-PU (DIP-28)  
U4: MAX485 (DIP-8)  
D3: dioda LED, żółta, 5 mm  
D7: dioda LED, zielona, 5 mm  
M1: B40D-DIO (mostek prostowniczy, DIP4)  
**Inne:**  
Q1: 16 MHz (kwarc, obudowa HC49/HC49S)  
F1: bezpiecznik 630 mA z gniazdem  
JP1: kątowa listwa goldpin  
JP2: goldpin prosty 2×3  
JP3...JP5: goldpin prosty 1×2  
JP6: goldpin kątowy 1×2  
J1...J3, J6: goldpin prosty 1×3  
J4: gniazdo NS25 (3 piny), wtyk NS25 (3 piny), 3 szt. terminali do wtyku NS25, wtyk XRL-3 do obudowy  
J7: złącze ARK3  
J8: złącze ARK2  
TR1: TZ10VA/9V (transformator zalewany do druku 9 V/10 VA)  
PPIN28: podstawa 28 pin, wąska  
PPIN8: podstawa 8 pin  
KM-50: obudowa

Listing 1. Najbardziej istotny fragment procedury inicjującej mikrokontroler

```
// Preskaler na 256, F = 16000000/64 = 250kHz, t = 4us
TCCR1B = ( CNT1_PRESCALER_64 );
TCNT1 = 0xFFFF - 2;
// Zezwolenie na IRQ od przepełnienia timera
TIMSK1 |= (1<<TOIE1);
```

Listing 2. Fragment procedury generującej impulsy dla serwomechanizmów i przerwy synchronizująca

```
/* Ustawiamy timer na czas w 4 * us, co da zakres: */
TCNT1 = 0xFFFF - ( TabServ[servo] + (1000/4) );
switch( servo )
/* ( 1000us / 4 + 0 ) * 4 = 1,000ms, ( 1000us / 4 + 128 ) * 4 = 1,512ms,
(1000us / 4 + 256 ) * 4 = 2,024ms */
{
    case ( 0 ) : SetS1(); break; // Ustawiamy Serwo1
    case ( 1 ) : ClrS1(); SetS2(); break; // Ustawiamy Serwo2, kasujemy 1
    case ( 2 ) : ClrS2(); SetS3(); break; // Ustawiamy Serwo3, kasujemy 2
    case ( 3 ) : ClrS3(); SetS4(); break; // Ustawiamy Serwo4, kasujemy 3
    case ( 4 ) : ClrS4(); // Kasujemy serwo
    TCNT1 = 0xFFFF - (2000 / 4); // ustawiamy Sync 20ms
    break;
}
servo++; if ( servo > 4 ) servo = 0;
```

Ustawienie zworek      Wartość

	1
	2
	4
	8
	16
	32
	64
	128
	256
	8+4+1=13
	128+64+8=200

Rysunek 4. Ustawienie adresu sterownika

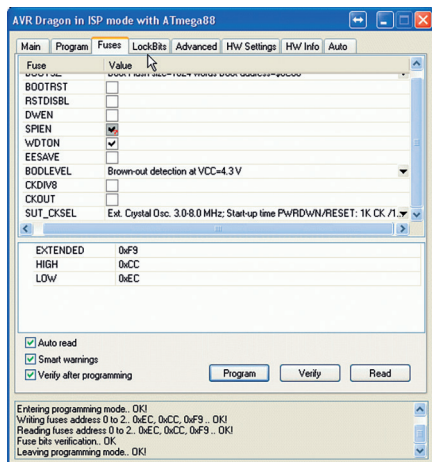
Montaż i uruchomienie

Montaż elementów jest typowy i nie wymaga szczegółowego omawiania. Schemat montażowy sterownika z serwomechanizmów z interfejsem DMX pokazano na rysunku 2. Płytkę jest 1-warstwowa, co spowodowało konieczność zastosowania dwóch zworek. Pod mikrokontroler warto zastosować podstawkę.

Uruchomienie rozpoczynamy od sprawdzenia napięcia zasilającego mikrokontroler. Gdy napięcie zasilające jest poprawne, umieszczamy mikrokontroler w podstawce. Jeśli nie jest on zaprogramowany, możemy to zrobić za pomocą złącza JP2. Ustawienie bitów konfiguracyjnych zaprezentowano na rysunku 3. W kolejnym kroku podłączamy urządzenie do sygnału DMX. Może to być konsola DMX lub np. moduł AVTDMX512 (<http://goo.gl/szSKZR>) z odpowiednim oprogramowaniem sterującym. Zworkami na urządzeniu ustawiamy adres zgodnie z rysunkiem 4. Regulując manipulatorami na konsoli lub w programie na komputerze powodujemy ruchy serwomechanizmu. Pierwszy adres steruje serwomechanizmem numer 1, kolejny numer 2 itd. W czasie pracy status sterownika jest sygnalizowany za pomocą diod LED – sposób sygnalizacji opisano w tabeli 1. Funkcje sterownika konfigurowane ze pomocą zworek opisano w tabeli 2.

Płytkę przystosowano do umieszczenia w obudowie KM-50.

Sławomir Skrzyński, EP



Rysunek 3. Ustawienie bitów konfiguracyjnych

zworę JP5. Dane DMX są konwertowane do poziomów TTL za pomocą układu U4. Zwora JP6 umożliwia załączenie terminali linii.

JP1 ustala adres DMX. Należy pamiętać, że sterownik zajmuje cztery kolejne adresy. U3 dekoduje dane DMX i generuje

impulsy dla serwomechanizmów. Impulsy są generowane w przerwanach Timera 1. Najistotniejszy fragment procedury inicjującej pokazano na listingu 1. Na listingu 2 zamieszczono procedurę generującą impulsy dla serwomechanizmów i przerwę synchronizującą.

W programie, za pomocą Timera 0 zrealizowano liczenie przekroczenia czasu (timeout) braku transmisji. Jeśli komendy nie docierają przez ponad 1 sekundę, serwomechanizmy są ustawiane w ostatniej pozycji lub przyjmują pozycję domyślną, zachowaną w pamięci EEPROM.

REKLAMA