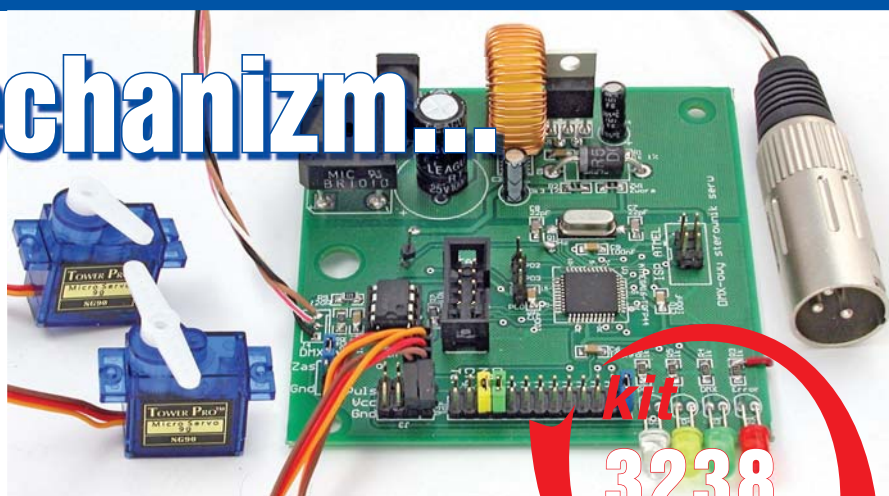


Serwomechanizm... DMX

Charakterystyka:

- Napięcie zasilania: 8..37V
- Liczba serwomechanizmów: 4
- Rozdzielczość regulacji: co najmniej 6 bitów.
- Zakres czasu impulsu sterującego 0,5...2ms
- Napięcie zasilania serwomechanizmów: 5V lub więcej
- Maksymalny prąd serwomechanizmów: 2,9A lub więcej
- Liczba adresów: 512
- Zapamiętanie ostatniej pozycji serwomechanizmu po zaniku transmisji; powrót do pozycji zerowej lub neutralnej,
- Akceptacja pojedynczej transmisji lub podwójnej z eliminacją błędów.

Zacznę od wstępu do artykułu, jaki ukazał się ponad 20 lat temu (w EdW 4/1997). Nieżyjący już Autor, Zbigniew Raabe, wprowadzając w tematykę sterowania serwomechanizmów modelarskich, napisał: *Tym razem Autor zaprasza Czytelników na niebezpieczną wyprawę. Udamy się na teren bagnisty i grząski, gdzie słońce nigdy nie wschodzi, a ziemia usiana jest kośćmi śmiatłków, którzy niebacznie zapuścili się na ten opuszczony przez bogów i ludzi obszar. (...) Kłopoty zaczynają się w momencie, kiedy efektem pracy układu elektronicznego ma być działanie mechaniczne – poruszenie „czegoś”. (...) chcemy zdalnie sterować kamerą video podczas wykonywania zdjęć przyrodniczych lub też poruszać kamerą w systemie dozoru. Samodzielne wykonanie potrzebnych nam układów mechanicznych raczej nie wchodzi w grę, chyba że ktoś jest fanatycznym wielbiciele mechaniki precyzyjnej i posiada odpowiednio wyposażoną pracownię.*



Na szczęście takie układy mechaniczne są od dawna produkowane seryjnie i można je bez większych problemów nabyć za niewygórowaną cenę. Są to serwomechanizmy modelarskie. (...) wysyłane są do serwomechanizmu w odstępach ok. 20 msec krótkie impulsy dodatnie o czasie trwania od 1 do 2 msec. Właśnie czas trwania tych impulsów decyduje (...) o położeniu, dokładniej o kącie obrotu tak zwanego orczyka. Cały ten artykuł wprowadzający w tematykę serwomechanizmów można znaleźć w Internecie pod adresami:

https://elportal.pl/pdf/k13/16_11.pdf

<https://serwis.avt.pl/manuals/AVT2217.pdf>

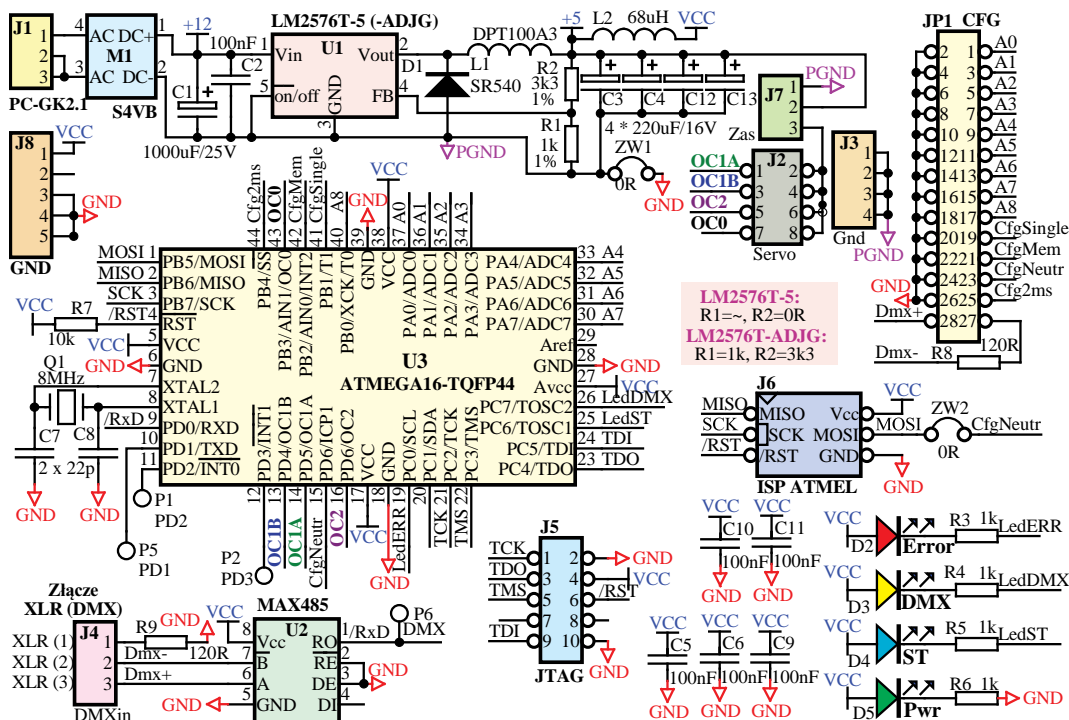
Współczesne serwomechanizmy modelarskie mogą zmieniać położenie różnych elementów, na przykład lampy LED, której barwa i moc światła jest sterowana sygnałem DMX. Wtedy jasność lamp jest sterowana przez linię DMX, a do sterowania

klasycznym serwomechanizmem potrzebny jest specyficzny sygnał impulsowy i co najgorsze, najtrudniejsze – oddzielny kanał transmisyjny. Przedstawiany projekt rozwiązuje problem oddzielnego kanału transmisyjnego: do sterowania typowym analogowym serwomechanizmem modelarskim wykorzystuje dostępny sygnał DMX. Pozwala niezależnie sterować czterema serwomechanizmami, wykorzystując cztery kanały (sloty) spośród 512 dostępnych w systemie DMX.

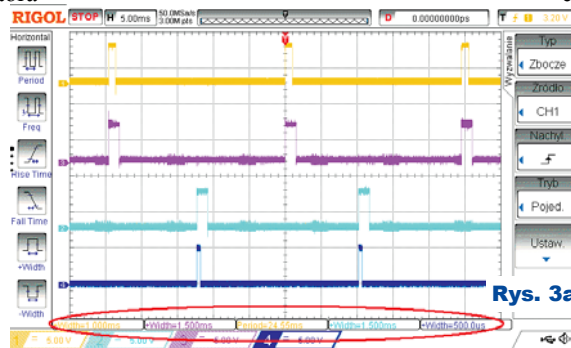
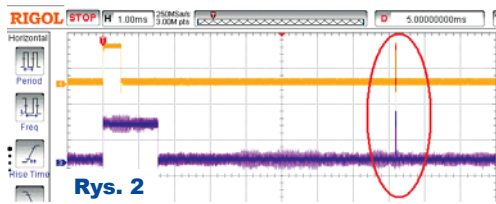
Opis układu

Schemat układu pokazany jest na rysunku 1. Sterownik jest zasilany z zasilacza o wydajności dostosowanej do

Rys. 1



liczby i poboru prądu zastosowanych serwomechanizmów. Napięcie zasilające jest ograniczone w stabilizatorze impulsowym U1 o maksymalnej wydajności 3A. Stabilizator zasilą zarówno mikrokontroler, jak i serwomechanizmy. Napięcie wyjściowe stabilizatora można zwiększyć do 5,5V, uzyskując większą moc serwomechanizmów. Wyższe napięcie zasilające może być groźne, ponieważ producent zaleca zasilanie mikrokontrolera maksymalnym napięciem 5,5V. Serwomechanizmy najczęściej akceptują zasilanie 4...6V, czasem więcej. Jeśli istnieje konieczność zasilania serwomechanizmu wyższym napięciem, można je podać na J7, w przeciwnym wypadku należy zewrzeć jumperem piny 2-3 tegoż złącza. Sygnał DMX jest konwertowany w układzie U2. Zwierając piny 27-28 złącza JP1, można włączyć terminowanie linii. Dane transmitowane po DMX są dekodowane w U3, który generuje także przebiegi za pomocą „PWM-ów” sterowanych timerem 0 i 1. Zakładając zworki A0...A8 na JP1, ustalamy adres bazy interfejsu (DMX). Kolejne serwomechanizmy będą miały kolejne adresy DMX: +1, +2 i +3. Zworka na pinach 19...20 włącza tryb akceptacji pojedynczej transmisji. W przeciwnym wypadku, aby dana została uznana za prawidłową, odczyty dwóch kolejnych ramek muszą



dać tę samą wartość. Serwomechanizmy reagują wtedy wolniej, ale ewentualne błędy są eliminowane. Zwierając 21 z 22 („Memory”), spowodujemy, że w przypadku zaniku komunikacji na

ponad sekundę, serwomechanizmy zachowają swoje ostatnie położenie. Brak zworki spowoduje powrót serwomechanizmów do pozycji zero. Po zwarceniu pinów 24-25 „Neutral” (także piny 4-6 J6 „ISP po SPI”) serwomechanizmy wrócą do pozycji neutralnej (środkowej) – impuls 1ms dla trybu 0,5...1,5ms (1,25ms dla trybu 0,5...2ms). Zworka na „2ms” (piny 25-26) wydłuża impuls maksymalnego położenia do 2ms. Należy pamiętać, że nie wszystkie serwomechanizmy akceptują taką wartość.

Protokół DMX był już omawiany wielokrotnie, więc go tu pominę. Danymi z wybranych slotów należyysterować PWM-y. Okazuje się, że to nie takie proste. Impuls trwa 0,5...1,5(2ms) a odstęp pomiędzy impulsami około 20...30ms. Aby zapewnić rozdzielczość 8 bit, timer musiałby być w przybliżeniu 12-bitowy. Niestety AVR posiadają głównie timery 8-bit, małą ilość (przeważnie jeden) 10-bit. Czasem możliwe jest uzyskanie timera 16-bit ale najczęściej jednego. Pozostaje zmniejszyć rozdzielczość do np. 4-bitów, co jest raczej nieakceptowalne. Można też zmniejszyć częstotliwość pojawiania się impulsów z 20ms do np. 5, ale nie ma gwarancji, że wszystkie sterowniki serwomechanizmów będą akceptować takie parametry. Zdecydowano się więc na to, aby po wygenerowaniu impulsu wyłączyć PWM na 20ms. Pierwsza próba polegająca na tym, aby w przerwaniu od przepełnienia timera wyłączyć PWM-a, zaowocowało skutkiem pokazanym na **rysunku 2**. Mianowicie po pierwszym przerwaniu pojawiał się krótki impuls o długości 125us. Impuls spowodowany jest tym, że PWM, w trybie FAST PWM, rozpoczyna generowanie impulsu w chwili przepełnienia timera, który później wyłączamy w obsłudze przerwań. Serwomechanizmy różnie mogą reagować na taki impuls i wyłączenie PWM należy zrealizować w przerwaniu od zrównania się licznika CNT z rejestrem OCR.

A dlaczego pierwotnie chciałem wykorzystać przerwanie od przepełnienia? Ze względu na to, że użyłem timera z dwoma PWM-ami, chciałem jednym przerwaniem włączyć i wyłączyć oba te PWM-y. Trzeba było niestety każdy

Nazwa	Numer pinów złącza JP1	Funkcja
Adres 0..8	1..18	Ustawienie adresu urządzenia (zajmuje 4 kolejne adresy). Adresy ponad 508 nie mają sensu, bo część serwomechanizmów nie będzie działać, ponieważ w DMX nie istnieją adresy >512
Single	19-20	Założona: Pojedyncza transmisja ustawia PWM Brak: Dwie identyczne transmisje ustawiają PWM
Mem	21-22	Założona: Przy braku transmisji serwa pamiętają ostatnie położenie Brak: Brak transmisji ponad 1 sekundę spowoduje powrót serwa do pozycji zerowej lub neutralnej zależnie od stanu zworki „Mem”
Neutral	23-24 i 4-6 J6	Założona: Powoduje przyjęcie pozycji neutralnej (połowa zakresu ruchu serwa) po resecie lub przy braku transmisji gdy zworka „mem” zdjęta Brak: Serwo przyjmuje pozycję zerową
2ms	25-26	Założona: Impulsy serwomechanizmu w granicach 0,5...1,5ms Brak: Impulsy serwomechanizmu w granicach 0,5...2ms
Term	27-28	Włącza terminowanie linii

Nazwa	Symbol	Funkcja
Power	D5	Świeci, gdy jest zasilanie
Status	D4	Świeci przy braku transmisji, miga podczas odbioru danych
DMX	D3	Świeci, gdy jest transmisja
Error	D2	Zaświeca się na dwie sekundy po resecie i na 50ms w przypadku wykrycia błędu transmisji: - bajt CS różny od zera. - trzy kolejne ramki mają różne wartości kanałów wybranych jako PWM. Dioda może więc zaświecać się podczas szybkich zmian wartości kanału

timer włączać i wyłączać w osobnym przerwaniu od rejestru OCRxA i OCRxB. Po takiej zmianie generowany przebieg wygląda wzorcowo, jak na **rysunku 3a** (w innej skali czasu na **rysunku 3b**).

A co w sytuacji, gdy wypełnienie PWM-a będzie maksymalne? Do takiej sytuacji nie dojdzie, bo zakres wartości PWM waha się w granicach od 64 do 128 (1,5ms) lub 192 (2ms). Daje to rozdzielczość 6 bitów, czyli 64 różne pozycje serwa, a można ją zwiększyć, taktując mikrokontroler zegarem 16MHz, używając innego podzielnika lub używając innego mikrokontrolera, w którym zakres podzielników jest większy niż w ATmega16, jak np. ATtiny25/45/85, ARM.

W **tabeli 1** podane są funkcje zwojek, a w **tabeli 2** – funkcje diod LED.

Uwagi dotyczące programu

W materiałach dodatkowych w Elportalu dostępne są kody źródłowe. Przy ich modyfikacji warto wiedzieć, ile czasu mikrokontroler może poświęcić na dodatkowe działania. Na **rysunku 4** pokazany jest oscylogram przedstawiający czas transmisji bajtu DMX (zielona obwódka) w stosunku do czasu obsługi przerwania funkcją SIGNAL (żółty) oraz całkowitego czasu obsługi IRQ (niebieski) łącznie z wejściem, operacjami na stosie i wyjściem z przerwania.

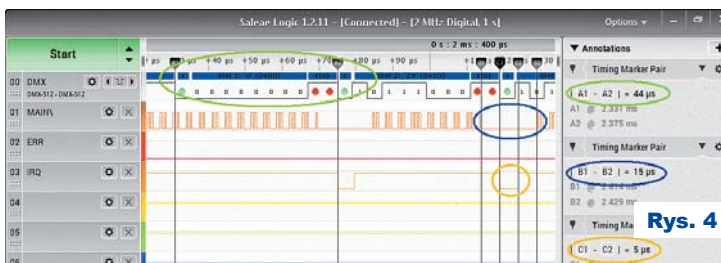
Czas obsługi IRQ można by zoptymalizować, pisząc wstawkę w assemblerze, a przerwanie zadeklarować z flagą NAKED. Chwila wejścia w IRQ jasno informuje o tym, że bez względu na liczbę bitów stopu przerwanie jest wywołane po pierwszym bicie stopu. Ustawienie liczby bitów stopu ma znaczenie tylko w czasie nadawania. Na **rysunku 5** jest zbliżenie pokazujące liczbę rozkazów wykonanych przez pętlę główną:

```
while(1) { PORTx ^= _BV(Py); }
```

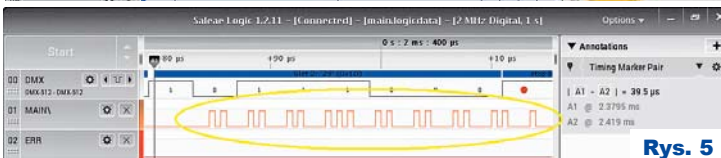
składającej się z czterech elementarnych rozkazów:

```
LOOP: LD      PORTx
      EOR      y
      ST      PORTx
      BNE     LOOP
```

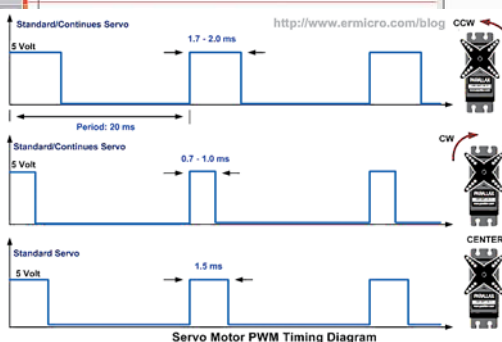
Brązowy przebieg to wynik działania tej pętli. Łatwo policzyć, że podczas transmisji bajtu mikrokontroler wykonał 72 rozkazy (18*4). Nierówne odstępy pomiędzy rozkazami spowodowane są



Rys. 4

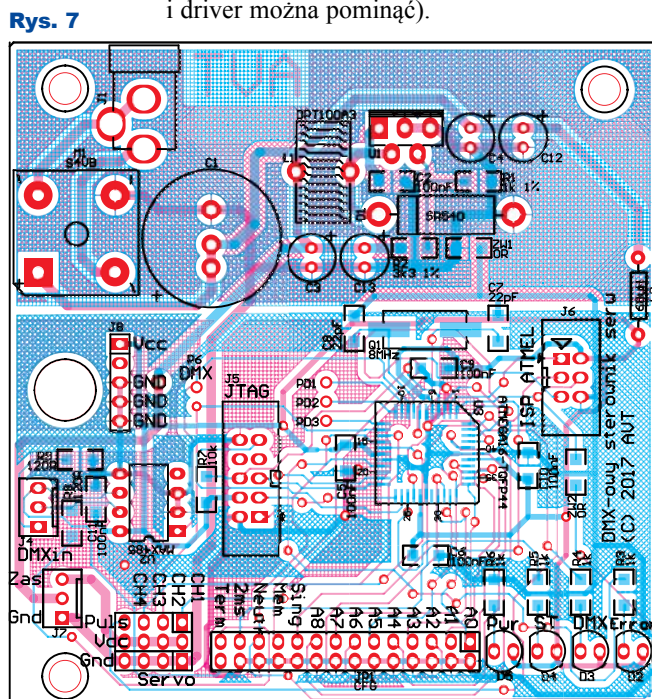


Rys. 5



Rys. 6

przerwaniem od timerów. Przy zegarze 16MHz mikrokontroler wykona, co oczywiste, dwa razy więcej rozkazów. Informacja ta jest o tyle cenna, że przerwanie od innych źródeł nie mogą wykonywać się zbyt długo, najbezpieczniej deklarować je jako INTERRUPT lub ISR z atrybutem NOBLOCK. Jeśli muszą koniecznie być zadeklarowane jako SIGNAL, to należy zadbać, aby nie wykonywały się dłużej niż 88us (AVR ma dwubajtowe FIFO) łącznie z czasem operacji na stosie. Takie postępowanie jest jednak ryzykowne i wskazane aby nie przekraczać czasu transmisji jednego bajtu. Trzeba bowiem mieć na uwadze, że przerwanie wykonuje się w różnym czasie zależnie od spełnionych bądź nie różnych warunków. Odczytanie bajtu z FIFO przy standardowej obsłudze realizowane jest przez wyjście z IRQ, wykonanie rozkazu programu głównego i wejście w IRQ. Operacja wejścia/wyjścia z IRQ trwa dwa razy dłużej niż sama obsługa IRQ. Można tę operację przyspieszyć przez sprawdzenie przed wyjściem z IRQ, czy nie ma bitu do odczytania, jeśli jest to skok na początek obsługi IRQ. Tym sposo-



Rys. 7

bem przyspieszymy obsługę IRQ przez brak wychodzenia i ponownego wchodzenia w przerwanie. W tym przypadku zamiast 15us na odczyt 2 bajtów z FIFO zużyjemy tylko 20, więc aż o 33% mniej.

Przy implementacji protokołu DMX na inne mikrokontrolery należy pamiętać, że niektóre, takie jak ATmega16, 161, 162, 32, ale już nie ATmega164, **wymagają wyboru**, (podobnie jak UART-y rodziny 16C45x i 16C55x) najstarszym bitem rejestru UCSRC pomiędzy rejestrem statusu a rejestrem UBRRH rejestru podzielnika.

Słowo ostrzeżenia: o sterowaniu serwowymi mechanizmami w sposób programowy (bez timerów) opisanym np. na stronie <http://hobby.abxyz.bplaced.net/index.php?pid=3&aid=20> można w zasadzie zapomnieć ze względu na stosunkowo szybką transmisję DMX.

I jeszcze jedna kwestia: różne źródła podają różne zakresy czasów impulsu, od 0,5...1ms, poprzez 1...2ms do 0,2...2,3ms oraz różny czas dla położenia neutralnego: 1 albo 1,5ms. Przykład na **rysunku 6**. W programie zadeklarowano stałe definiujące czasy, więc w razie potrzeby łatwo je zmodyfikować.

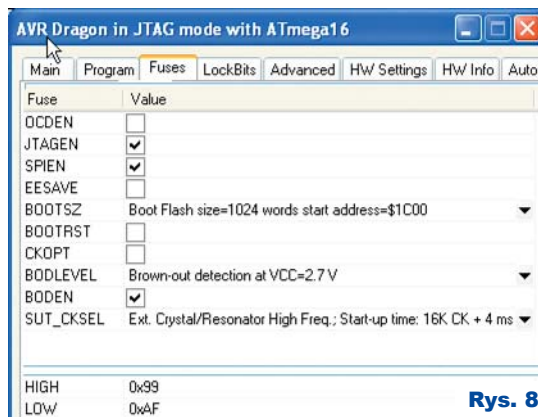
Jeśli potrzebne jest więcej wyjść PWM, należy wykorzystać ATmega128 (6 PWM) albo po prostu połączyć ze sobą dwa sterowniki (w jednym zasilacz i driver można pominąć).

Montaż i uruchomienie

Układ można zmontować na płytce drukowanej, pokazanej na rysunku 7.

Montaż jest typowy i nie wymaga omawiania. Pod U2 warto zastosować podstawkę. U1 może wymagać radiatora przy dużym prądzie obciążenia. Na początek nie zalecam montażu mikrokontrolera i drivera U2. Uruchomienie należy bowiem rozpocząć od zasilacza. Zależnie od użytego stabilizatora (LM2567T-5 lub LM2576T-ADJG) wartości rezystorów R1 i R2 będą różne. Dopiero wtedy, gdy napięcie zasilania jest prawidłowe, można włutować mikrokontroler i umieścić U2 w podstawce.

Jeśli układy te zostały włutowane przed uruchomieniem zasilacza, najbezpieczniej będzie się posłużyć zasilaczem laboratoryjnym. Należy ustawić napięcie 6V, prąd ograniczyć do 100 mA. Powoli zwiększać napięcie i kontrolować je po przejściu przez stabilizator na złączu J8. Jeśli przekroczy 5,5V to został popełniony błąd. Wartość napięcia większa niż 7V może uszkodzić układy U2 i U3 (nieprawidłowe wartości R1 i R2, złe luty w ich pobliżu). Jeśli przy napięciu na zasilaczu laboratoryjnym 8V na złączu J8 napięcie mieści się w granicach 5V $\pm 5\%$, zasilacz pracuje poprawnie. Jeśli mikrokontroler nie był zaprogramowany, można to zrobić JTAG-em lub przez SPI (ISP via SPI). Należy pamiętać



Rys. 8

o odpowiednim ustawieniu bitów konfiguracyjnych jak na rysunku 8 lub użyć pliku *.ELF, w którym są zawarte dane o bitach konfiguracyjnych.

Najważniejsze jest poprawne ustawienie źródła sygnału zegarowego. Zdecydowanie nie zalecam używania wewnętrznego generatora RC 8MHz. Dwuprocentowa dokładność generatora RC jest za mała aby bezbłędnie odbierać transmisję 11 bitów (start, 8 danych, dwa bity stopu). Do bezbłędnej transmisji UART wymagana jest jednoprocetowa dokładność. BOD powinien być nastawiony na 2,7V. Przy ustawieniu 4V może się resetować podczas pracy serwomechanizmu (dioda error zaświeca się na 2 sekundy). Aby się przed tym zabezpieczyć, należałoby zasilac serwomechanizmy przez filtr LC (łatwo go wstawić do złącza J7) albo z pomocą zewnętrznego zasilacza. Zasilanie mikrokontrolera jest filtrowane dławikiem L1.

Wykaz elementów

ZW1 ZW2	0Ω SMD 1206
R1	1kΩ 1% SMD 1206
R2	3,3kΩ 1% SMD 1206
R3 R4 R5 R6	1kΩ SMD 1206
R7	10kΩ SMD 1206
R8 R9	120Ω SMD 1206
C1	1000uF/25V ce18/7.5
C2 C5 C6 C9 C10 C11 C13	100nF SMD 1206
C4 C3 C12 C13	220uF/16V THT ce6.3/2.5
C7 C8	22pF SMD 1206
L1	DPT100A5 dławik THT
L2	.68uH dławik THT
Q1	8MHz kwarc SMD
D1	SR540 dioda THT
D2	LED Czerwona THT 5mm
D3	LED Żółta THT 5mm
D4	LED niebieska THT 5mm
D5	LED zielona THT 5mm
M1	S4VB mostek prostowniczy
U1	LM2576T-5 (-ADJG) TO-220V-5
U2	MAX485
U3	ATMEGA16-TQFP44
J1	PC-GK2.1 Gn. Zas.JACK2.1/5.5
J2	listwa Goldpin 2x4
J3	listwa Goldpin 1x4
J4	SN25-3W + SN25-3G + piny + gniazdo XRL3
J5	ZL201-10G IDC10MLP
J6	ZL201-06G IDC6MLP
J7	SN25-3W + jumper
JP1	ZL201-14G listwa goldpin + jumpery
DIP8	PPIN8 podstawka pod U2

Komplet podzespołów z płytką jest dostępny w Sklepie AVT jako zestaw AVT3238

Sterownik można umieścić w obudowie KM-50, do której przystosowano otwory mocujące.

SaS
sas@elportal.pl