

Sterownik silnika krokowego z interfejsem MODBUS, część 1

AVT-5137

Sterowanie silnikami krokowymi nie jest wielką sztuką, jednak na naszych łamach pojawia się raczej niewiele artykułów poruszających tę tematykę.

Nadrabiając zaległości prezentujemy projekt sterownika wykorzystującego protokół MODBUS.

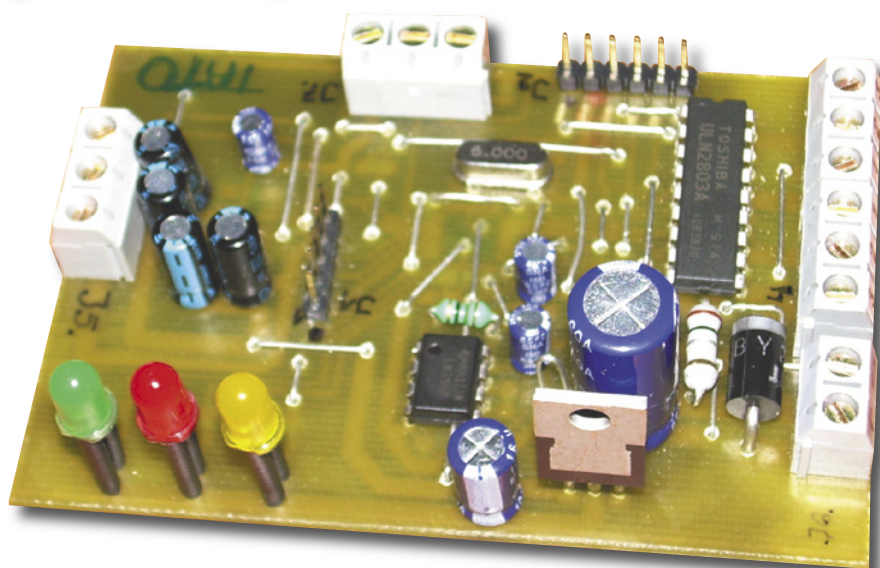
Rekomendacje:

to projekt, który powinien zainteresować głównie automatyków i nie tylko dlatego, że do komunikacji z komputerem wykorzystano protokół MODBUS.



PODSTAWOWE PARAMETRY

- Płytko o wymiarach 105x149 mm
- Zasilanie: 8...35 V
- Maksymalny pobór prądu jednej cewki silnika: 1 A
- Sterowanie silników 2- i 4-fazowych
- Praca w trybie pół-krokowym i pełno-krokowym
- Zatrzymanie silnika po dojściu do określonej pozycji lub otrzymaniu sygnału z wyłącznika krańcowego
- Pomiar napięcia zasilającego silnik
- Pomiar prądu pobieranego przez silnik
- Możliwość użycia czujników krańcowych typu NC/NO
- Komunikacja: RS232 9600,n,8,1 protokół MODBUS RTU
- Sygnalizacja podstawowych stanów pracy za pomocą diod LED



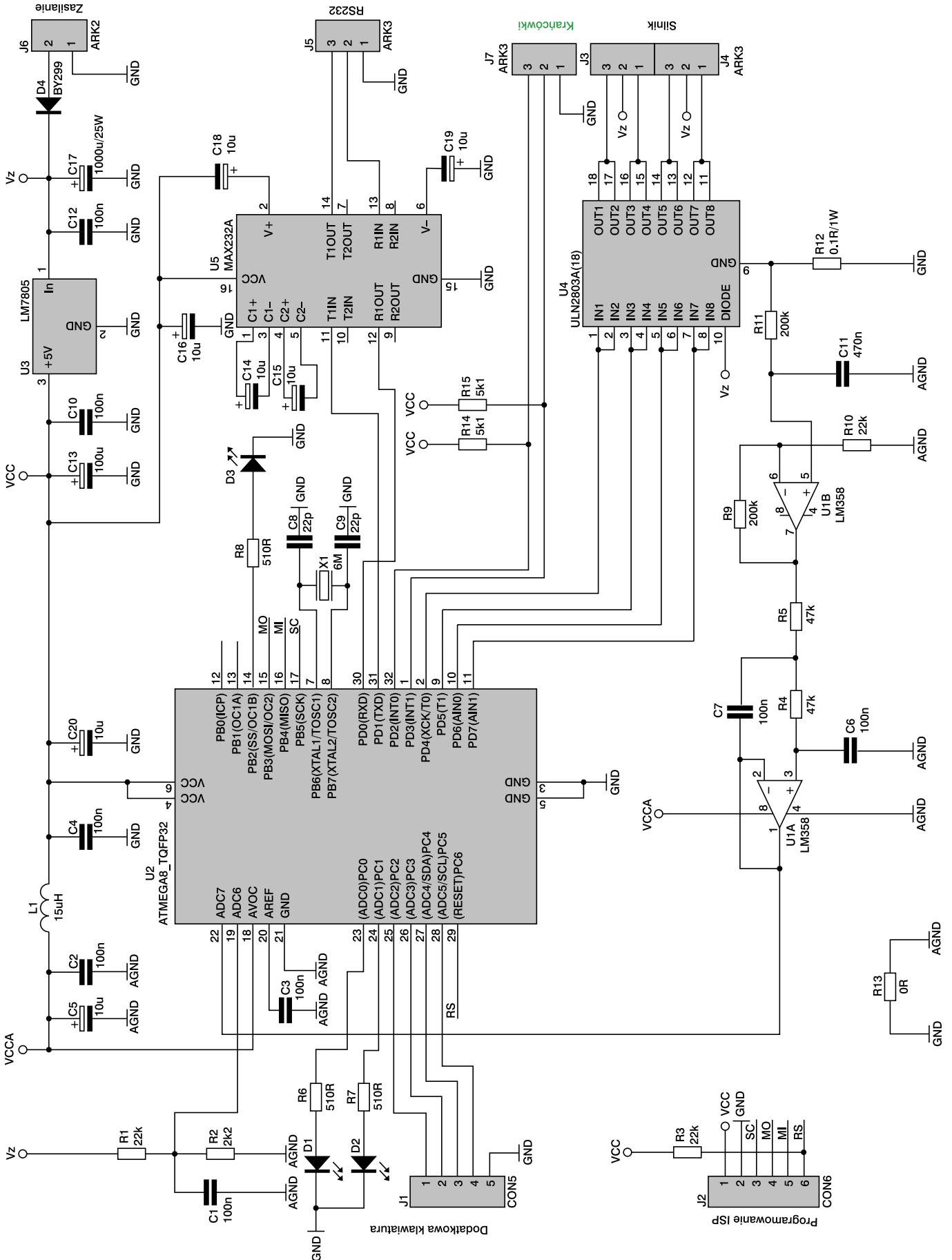
W artykułach o sterowaniu silnikami krokowymi jakie ukazały się w EP wykorzystywano głównie układy o małej skali integracji, natomiast w bardziej rozbudowanych projektach stosowano mikrokontrolery. Żadne z przedstawionych rozwiązań nie umożliwiała jednak sterowania pracą silnika z wykorzystaniem protokołu MODBUS – popularnego w układach automatyki przemysłowej.

W niniejszym artykule zostanie zaprezentowany prosty mikroprocesorowy sterownik silników krokowych małej mocy. Pełne możliwości sterowania silnikiem uzyskujemy za pomocą interfejsu szeregowego, ale sterownik jest również wyposażony w złącze prostej klawiatury pozwalającej na sterowanie silnikiem krokowym bez użycia komputera. Sterowanie za pomocą klawiatury umożliwia uruchomienie i zatrzymanie silnika oraz zmianę kierunku i prędkości. Pomysł zrealizowania sterownika o takiej funkcjonalności powstał w momencie, gdy zaistniała potrzeba sterowania dodatkowym posuwem maszyny za pomocą jej sterownika głównego, którym jest komputer przemysłowy PC104.

Budowa

Schemat elektryczny urządzenia przedstawiono na rys. 1. Całością steruje dobrze znany czytelnikom mikrokontroler ATmega8 (U2) posiadający 8 kB pamięci Flash oraz 1 kB pamięci RAM. Dodatkowo zawiera on wewnątrz 10-bitowy przetwornik A/C, który jest wykorzystywany do pomiaru napięcia zasilającego oraz prądu pobieranego przez silnik. Mikrokontroler pracuje w konfiguracji z zewnętrznym rezonatorem kwarcowym 6 MHz (X1), ponieważ potrzebny jest stabilny sygnał zegarowy wykorzystywany do transmisji szeregowej oraz wyznaczenia prędkości silnika.

Do linii PC0 jest podłączona czerwona dioda (D1), która informuje użytkownika o wystąpieniu w sterowniku sytuacji awaryjnej, np. przekroczenia maksymalnego prądu pobieranego przez silnik. Do linii PC1 podłączona jest dioda zielona, która informuje o wirowaniu silnika w lewo, natomiast do linii PB2 podłączona jest dioda zielona, której zadaniem jest informowanie o wirowaniu silnika w prawo. Ponieważ w urządzeniu wykorzystano zamontowany na stałe mikrokontroler w obudowie TQFP32, konieczne stało się umieszczenie złącza szpilkowego J2 umożliwiającego zaprogramowanie



Rys. 1. Schemat elektryczny sterownika

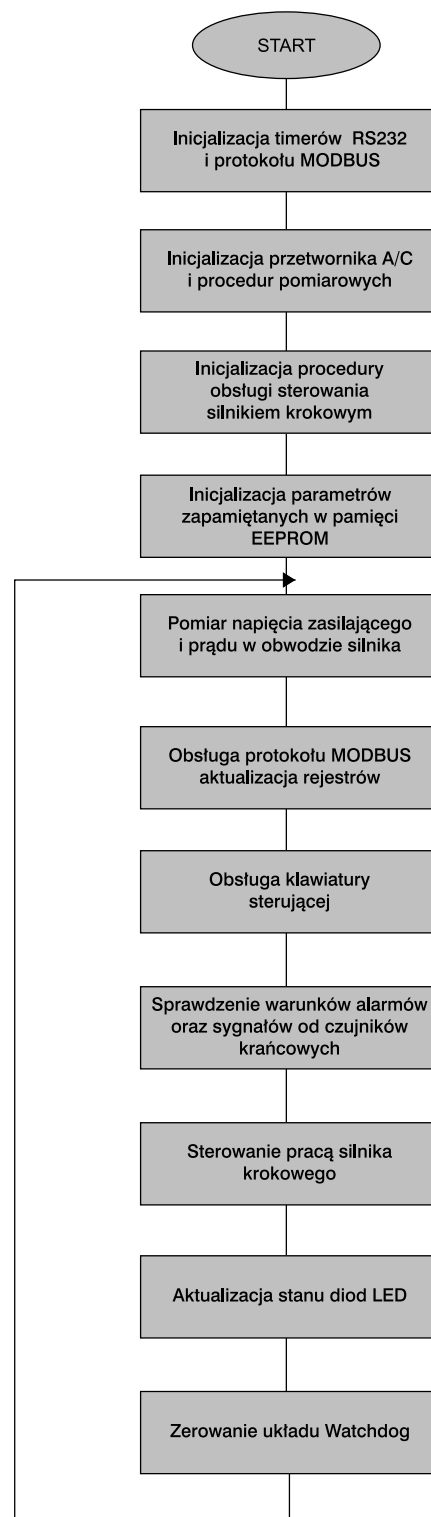
układu w systemie docelowym. Układ jest wyposażony także w dodatkowe złącze J1 umożliwiające opcjonalnie dołączenie 3–klawiszowej klawiatury sterującej urządzeniem. Przycisk podłączony do linii PC2 służy do wyboru kierunku ruchu oraz zatrzymania silnika, natomiast przyciski podłączone do linii PC3 i PC4 służą odpowiednio do zwiększania i zmniejszania obrotów silnika. Interfejs portu szeregowego RS232 rozwiązano w sposób klasyczny, podłączając do linii RXD i TXD portu szeregowego mikrokontrolera układ MAX232 (U5), który pracuje w swojej podstawowej aplikacji. Linie PD2 i PD3 mikrokontrolera wyprowadzono na złącze ARK3 (J7) służące do podłączenia czujników krańcowych. Dodatkowe rezystory podciągające służą do wymuszenia prądu o wartości kilkunastu miliamperów w obwodzie czujników. Napięcie zasilające układ (łącznie z silnikiem) jest doprowadzone za pomocą diody BY299 (D4), zabezpieczającej przed dołączeniem napięcia o odwrotnej polaryzacji. Napięcie zasilające mikrokontroler i część analogową jest stabilizowane za pomocą klasycznego stabilizatora liniowego 7805 (U3). Dodatkowo napięcie zasilające część analogową mikrokontrolera oraz całego układu jest filtrowanie za pomocą prostego filtra LC (L1, C4, C2), eliminując w ten sposób zakłócenia, jakie mogą się przedostać z części cyfrowej układu oraz silnika. Do sterowania uzwojeniami silnika wykorzystano układ ULN2803 (U4) zawierający w swojej strukturze 8 tranzystorów Darlingtona o maksymalnym prądzie $I_c=0,5$ A wyposażonych w dodatkowe diody zabezpieczające. Do sterowania silnikiem krokowym wystarczą nam 4 sygnały, dlatego poszczególne tranzystory połączono parami równolegle, uzyskując w ten sposób wyjścia o maksymalnej wydajności prądowej około 1 A. Z uwagi na to, że tranzystory te są umieszczone w jednej strukturze – producent zapewnia jednakowe parametry poszczególnych tranzystorów – wyjścia układu możemy połączyć bezpośrednio ze sobą bez wykorzystywania dodatkowych rezystorów wyrównujących prądy w poszczególnych gałęziach. Pomiar napięcia zasilającego silnik dokonywany jest poprzez wejście ADC6 z wykorzystaniem dzielnika napięcia 1:10, dzięki któremu możliwy jest pomiar napięcia do wartości 25,6 V. Pomiar prądu zasilającego silnik jest realizowany z wykorzystaniem bocznika (rezystor R12) o wartości

0,1 Ω . Napięcie wyjściowe z bocznika jest podawane na prosty filtr dolno-przepustowy R11, C11, a następnie na układ wzmacniacza nieodwracającego o wzmacnieniu ok. 10 z wykorzystaniem połówki układu LM358 (U1). Sygnał prądowy po odfiltrowaniu w prostym filtrze RC nadal może zawierać składową zmienną, więc z wyjścia wzmacniacza jest on następnie podawany na drugą połowę układu LM358 (U1), na której zrealizowano prosty aktywny filtr dolno-przepustowy o częstotliwości granicznej około 10 Hz. Po wyjściu z filtru sygnał jest podawany bezpośrednio na wejście przetwornika A/C mikrokontrolera (ADC7).

Oprogramowanie

Oprogramowanie sterujące pracą mikrokontrolera zostało napisane w języku C z wykorzystaniem kompilatora GCC. Program działa w pętli nieskończonej, zgodnie z klasycznym algorytmem: czytaj wejścia, przetwarzaj dane, ustaw wyjścia. Zasadę działania programu zilustrowano na **rys. 2**.

Po włączeniu zasilania sterownik rozpoczyna pracę od inicjalizacji interfejsu RS232 z parametrami 9600,n,8,1, timera systemowego wykorzystywanego do odmierzania czasu oraz procedur związanych obsługą protokołu MODBUS. Ustawiany jest także przetwornik analogowo–cyfrowy, którego kanał 7 służy do pomiaru prądu pobieranego przez silnik, a kanał 8 do pomiaru napięcia zasilającego. Po tej czynności wykonywany jest kod inicjalizujący moduł sterujący pracą silnika krokowego. W efekcie uruchamiany jest m.in. 16-bitowy Timer1 sterujący pracą silnika krokowego. Następnie są ustawiane linie PD3...PD7 służące do generowania sekwencji sterujących dla silnika. Kolejną czynnością wykonywaną w procesie inicjalizacji jest odtworzenie wszystkich parametrów konfiguracyjnych z pamięci EEPROM sterownika. W pamięci nieulotnej są zapisywane następujące parametry konfiguracyjne: typ używanego silnika krokowego (2– lub 4-fazowy), tryb sterowania (pełno krokowy lub pół krokowy), sposób zachowania sterownika po zatrzymaniu silnika (włączenie lub wyłączenie prądu zasilającego silnik), aktualną pozycję w jakiej znajduje się silnik. Po odczytaniu parametrów konfiguracyjnych program wchodzi do nieskończonej pętli, w której najpierw jest odczytywana wartość z kanałów 7 i 8 przetwornika



Rys. 2. Algorytm pracy sterownika

A/C, a następnie na podstawie parametrów korekcyjnych jest wyznaczane bieżące napięcie zasilające oraz prąd silnika. Po tym następuje obsługa protokołu MODBUS, gdzie na podstawie danych odebranych poprzez interfejs RS232 wykonywana jest odpowiednia akcja. Polega ona na wydaniu rozkazu dla silnika lub zapisaniu, bądź odczytaniu parametrów konfi-

List. 1. Procedura sterowania liniami załączającym poszczególne uzwojenia silnika

```

/* Przerwanie od porównania dla Timera 0 */
SIGNAL(SIG_OUTPUT_COMPARE1A)
{
    if( ToPos && CurrPos==TargetPos)
    {
        IsRun = FALSE;
        StopMotorHw();
        return;
    }
    //Generowanie wektora wyjściowego
    u08 mport = MOTOR_PORT;
    mport &= ~MOTOR_MASK;
    mport |= pgm_read_byte(&OutVector[StepCnt]);
    MOTOR_PORT = mport;
    if(StepDir!=MOTOR_POS_LEFT)
    {
        StepCnt++;
        if(StepCnt==OutVectorSize) StepCnt=0;
        CurrPos++;
    }
    else
    {
        StepCnt--;
        if(StepCnt==0xff) StepCnt = OutVectorSize-1;
        CurrPos--;
    }
}

```

guracyjnych oraz odesłaniu odpowiedzi zgodnie ze specyfikacją protokołu MODBUS do urządzenia nadrzędnego. Kolejną czynnością podejmowaną przez sterownik jest odczyt zmiennej zawierającej stan wciśniętych klawiszy na klawiaturze zewnętrznej i na podstawie tego zostaje zaktualizowana zmienna określająca prędkość silnika oraz jego aktualny stan (zatrzymany, kierunek lewo, kierunek prawo). Procedura odczytywania stanu klawiatury jest wykonywana tylko wtedy, gdy wcześniej za pomocą odpowiedniego rozkazu protokołu MODBUS zostało wybrane sterowanie za pomocą klawiatury. Natomiast jeżeli wybrano sterowanie pracą urządzenia za pomocą komputera, wówczas procedura nie wykonuje żadnych czynności, co uniemożliwia wydawanie sterownikówi sprzecznych rozkazów z komputera i zewnętrznej klawiatury. Kolejną czynnością wykonywaną przez sterownik jest analiza sytuacji alarmowych, których wystąpienie powoduje ustawienie odpowiednich flag alarmów. W procedurze obsługi sygnałów alarmowych analizowany jest stan wyłączników krańcowych oraz napięcie zasilające i prąd silnika. W przypadku wystąpienia odpowiedniego sygnału z czujnika krańcowego lub przekroczenia wcześniej ustawionych dopuszczalnych parametrów silnika, ustawiane są odpowiednie znaczniki. Na podstawie flag alarmów i sygnałów wejściowych wysyłane są właściwe rozkazy do modułu odpowiedzialnego za bezpośrednie sterowanie silnika. Ostatnią czynnością jest aktualizacja stanu urządzenia za pomocą diod LED. Jeżeli flaga alarmu od przekroczenia napięcia lub prądu

jest ustawiona, wówczas wywoływana jest funkcja, która aktywuje miganie czerwonej diody informującej o wystąpieniu sytuacji awaryjnej. Gdy silnik obraca się w lewo wówczas uaktywniane jest błyskanie diody żółtej, natomiast w przypadku obrotu w prawo włączana jest dioda czerwona. W przypadku uaktywnienia odpowiedniej flagi oznaczającej zatrzymanie silnika na czujniku krańcowym, zaświecana jest na stałe dioda zielona lub żółta oznaczająca kierunek ruchu. Sterowanie liniami załączającym poszczególne uzwojenia silnika odbywa się z wykorzystaniem procedury przerwania od układu porównującego licznika, który został przedstawiony na **list. 1**.

Wirowanie silnika następuje w wyniku podawania wektorów sterujących zawartych w odpowiednich tablicach, które następnie są bezpośrednio wysyłane do portu wyjściowego. W sterowniku zdefiniowano po dwie tablice wektorów dla silnika 4-fazowego oraz 2-fazowego, dla trybu pełno oraz półkrokowego. Prędkość wirowania silnika zależy od szybkości zmian wektorów sterujących, więc jest zależny od częstotliwości zgłaszania przerwania, a kierunek obrotów zależy od kolejności wystawiania wektorów. Oprócz sterowania pracą silnika w procedurze przerwania obliczana jest także liczba kroków wykonanych przez silnik, a w przypadku, gdy silnik wykonuje ruch do określonej pozycji w momencie zrównania się licznika pozycji z wartością zadaną następuje zatrzymanie silnika w wyniku wywołania funkcji *StopMotorHw()*. Procedura ta zatrzymuje układ czasowo-licznikowy oraz w zależności od flagi określają-

cej czy silnik ma zachować moment obrotowy, następuje wyzerowanie lub nie linii portów wyjściowych, co powoduje wyłączenie prądu zasilającego uzwojenia silnika.

Montaż i uruchomienie

Schemat montażowy sterownika został przedstawiony na **rys. 3**. Całość została wykonana na płycie jednostronnej z wykorzystaniem elementów przewlekanych oraz SMD. Montaż rozpoczynamy od wlutowania zworek z kawałków drutu, a następnie rezystorów SMD oraz kondensatorów od strony druku. Po stronie elementów montujemy teraz elementy dyskretnie od najmniejszych do największych, a po nich złącza. Kolejny etap to montaż układów scalonych od strony elementów. Są to: wzmacniacz operacyjny U1 oraz układ ULN2803 (U4), a na koniec układ scalony MAX232 (U5) oraz procesor (U2). W urządzeniu zastosowano złącza ARK. Złącze ISP (J2) służące do programowania oraz złącze klawiatury (J1) to typowe goldpiny. Jeżeli nie planujemy stosowania zewnętrznej klawiatury, złącza J2 można nie montować. W analogowym torze pomiarowym zaleca się

WYKAZ ELEMENTÓW

Rezystory

R12: 0,1 Ω/1 W
 R9, R11: 200 kΩ
 R1, R3, R10: 22 kΩ
 R2: 2,2 kΩ
 R4, R5: 47 kΩ
 R6, R7, R8: 510 Ω
 R14, R15: 5,1 kΩ

Kondensatory

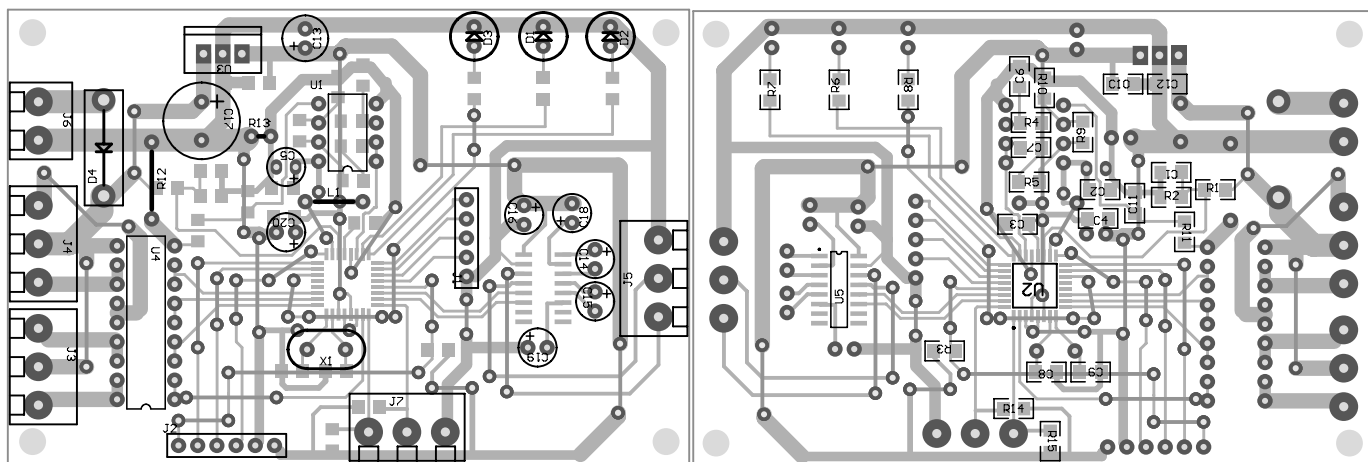
C17: 1000 μF/25 V
 C12: 100 nF
 C13: 100 μF/25 V
 C5, C14...C16, C18...C20: 10 μF/25 V
 C8, C9: 22 pF
 C11: 470 nF

Półprzewodniki

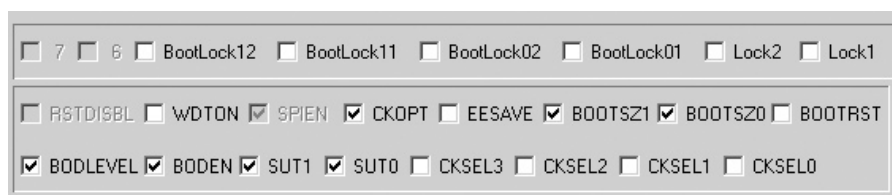
U2: ATmega8 (TQFP32)
 D4: BY299
 U1: LM358
 U3: LM7805
 U5: MAX232A
 U4: ULN2803A(18)
 D1, D2, D3: dioda LED 5 mm

Inne

X2: rezonator kwarcowy 6 MHz
 J6: złącze
 J3, J4, J5, J7: złącze ARK3
 J1, J2: złącze szpilkowe 2,54 mm
 L1: dławik 15 μH



Rys. 3. Schemat montażowy



Rys. 4. Ustawienie fusebitów

użyć rezystorów o tolerancji 1%, jednak w ostateczności można użyć rezystorów 5%, ponieważ urządzenie umożliwia automatyczną korekcję zmierzonych wartości. W tym przypadku należy się liczyć z tym, że rezystory 5% mają gorszą stabilność temperaturową, ale z uwagi na orientacyjną wartość pomiarów nie powinno to mieć większego znaczenia. Uruchomienie układu rozpoczynamy od sprawdzenia poprawności montażu oraz podłączenia do złącza J6 napięcia za-

silającego o wartości 8...35 V. Następnie należy zmierzyć wartość napięcia zasilającego doprowadzonego do procesora oraz wzmacniacza operacyjnego, które powinny wynosić 5 V. Gdy już będziemy pewni o prawidłowości zasilania wszystkich układów możemy przystąpić do wgrania oprogramowania do mikrokontrolera za pomocą programatora ISP. W celu zaprogramowania mikrokontrolera, do złącza J2 należy podłączyć programator ISP, a na komputerze PC uruchomić pro-

gram umożliwiające zaprogramowanie procesora, np. PonyProg. Po wybraniu odpowiedniego typu procesora (ATmega8) należy go zaprogramować plikami *ssk.hex* oraz *ssk.eep* zawierającymi obraz pamięci Flash oraz EEPROM. Następnie weryfikujemy poprawność programowania. Po zaprogramowaniu procesora pozostało nam jeszcze odpowiednie ustawienie *fusebitów*, tak jak przedstawiono to na **rys. 4**.

Po zakończeniu programowania możemy odłączyć programator i odłączyć oraz podłączyć napięcie zasilające. Jeżeli cały proces programowania przebiegł pomyślnie, w momencie dołączenia napięcia zasilającego wszystkie diody LED powinny się zapalić na 0,5 s, a następnie zgasnąć.

Lucjan Bryndza, EP
lucjan.bryndza@ep.com.pl

R E K L A M A

Wstęp do Klubu AVT



Prenumerujesz więcej niż jedno z powyższych pism?

To znaczy, że jesteś już **Członkiem Klubu AVT** uprawnionym do comiesięcznego zamawiania bezpłatnych egzemplarzy naszych czasopism, wydanych przed 2 miesiącami.

Jeśli prenumerujesz *n* czasopism, możesz zamówić *n-1* darmowych egzemplarzy (np. Prenumeratorem 2 tytułów może otrzymać za darmo 1 egzemplarz, zaś Prenumeratorem 6 tytułów ma prawo do 5 darmowych egzemplarzy).

Prezentacje aktualnie oferowanych numerów wszystkich czasopism znajdziesz na stronach

www.Klub.AVT.pl. Tam również możesz złożyć bezpłatne zamówienie.

Jeszcze nie prenumerujesz?

Zaprenumeruj! Zajrzyj na str. 152 lub skontaktuj się z Działem Prenumeraty: tel. 022 2578422, e-mail prenumerata@avt.pl