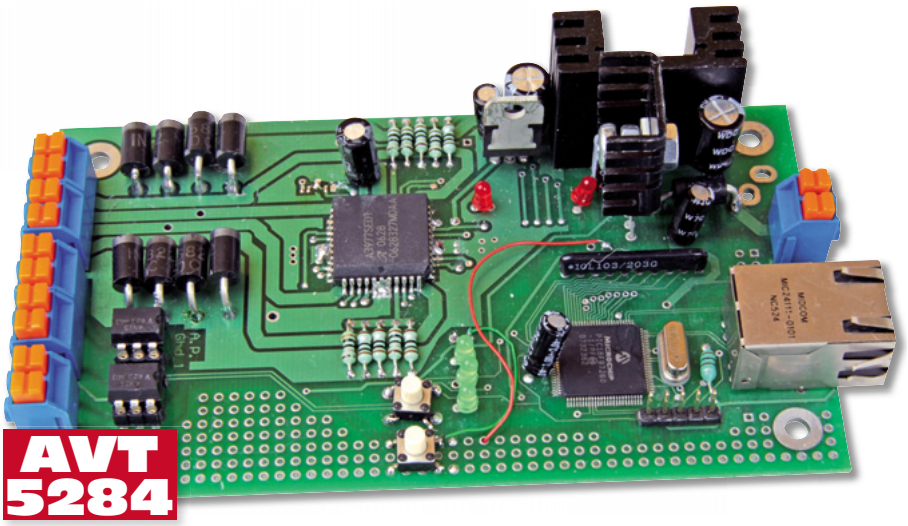
MechaNet (1)
Sterownik silnika krokowego kontrolowany przez Ethernet

Każdy elektronik hobbysta poszukuje jakiś zastosowań dla swoich projektów. Radość z dzieła jest większa gdy nowe elektro-cudo nie jest tylko ozdobnym, świecącym bajerem. Jednym z takich bardziej praktycznych projektów może być układ pozwalający na wykonanie mechanicznej pracy. O ile ciekawiej może być jeżeli ta praca może być sterowana zdalnie z komputera PC poprzez sieć LAN. Proponuję więc ethernetowy sterownik bipolarnego silnika krokowego z protokołem TCP/IP – MechaNet.

Rekomendacje: *ciekawy projekt będący przykładem implementacji i użycia stosu TCP/IP w mikrokontrolerze PIC firmy Microchip; może przydać się w automatyce, robotyce, inteligentnym budynku itd.*



Schemat sterownika pokazano na **rysunku 1** i **rysunku 2**. Jądrem konstrukcji jest mikrokontroler PIC18F97J60. Układ ten jest przeznaczony specjalnie dla aplikacji ethernetowych. Ma wbudowany mostek Ethernet PHY (10BaseT) i sprzętowy kontroler MAC, dzięki którym do dołączenia magistrali sieciowej wystarcza gniazdo RJ-45 i kilka elementów dyskretnych (w układzie zastosowano gniazdo ze zintegrowanymi transformatorkami separującymi). Programista ma do dyspozycji 128 kB pamięci programu, 3808 B RAM i 8 kB bufor ethernetowy. Do programowania MCU przewidziane jest listwa ICSP.

Głównym układem wykonawczym jest A3977 firmy Allegro MicroSystems. Jest to sterownik silnika krokowego zintegrowany z kontrolerem. Pozwala on na realizację sterowania mikrokrokowego z podziałką do 1/8. Maksymalny prąd pracy układu to 2,5 A na cewkę, a maksymalne napięcie zasilania obwodu mocy 35 V. Tranzystory mocy układu są wykonane w technologii DMOS, dzięki czemu układ podczas pracy nie wytwarza dużo ciepła.

Sterowanie układem odbywa się poprzez sygnały wejść logicznych:

- STEEP, którego każde narastające zbocze powoduje wykonanie jednego skoku.
- DIR determinujące kierunek obrotów silnika.
- MS1 i MS2 określające wielkość skoków wykonywanych przez silnik.
- /RESET służące wyzerowania kontrolera i wyłączenie stopnia mocy.
- /SLEEP służące do wprowadzenie układu w stan oszczędzania energii.

Ponadto poziom na wejściu SR i napięcie na wyprowadzeniu PFD determinują sposób gaszenia prądów zwrotnych, generowanych przez SEM zasilanego silnika. Do wyboru jest wykorzystanie do tego celu zewnętrznych diod, tranzystorów z wewnętrznego mostka mocy lub praca mieszana. Powoduje ona polepszenie parametrów pracy napędu przy średnich prędkościach obrotowych.

Podłączenie do wyprowadzenia REF sterownika napięcia generowanego przez przetwornik C/A oparty o drabinkę R-2R i port MCU pozwala na programową regulację maksymalnych prądów w cewkach sterowanego silnika.

Sterownik ma dwa wejścia odizolowane za pomocą transoptorów 4N25. Są to wejścia dla sygnałów krańcówek.

Cały układ może być zasilany napięciem stałym (12...32 V), które jest jednocześnie źródłem mocy dla kontrolera silnika. Pozostałe elementy są zasilane napięciem 9 V ze stabilizatora 7809. Jego napięcie jest też podawane na stabilizatory napięcia 3,3 V. Takie rozwiązanie zastosowano ze względu na niski pułap dopuszczalnego napięcia wejściowego dla stabilizatorów LF33CV (max 18 V) oraz w celu ograniczenia strat mocy. Dodatkowo, napięcie 9 V jest wyprowadzane przez rezystory na złącze J2, więc może być użyte do zasilania elementów krańcowych.

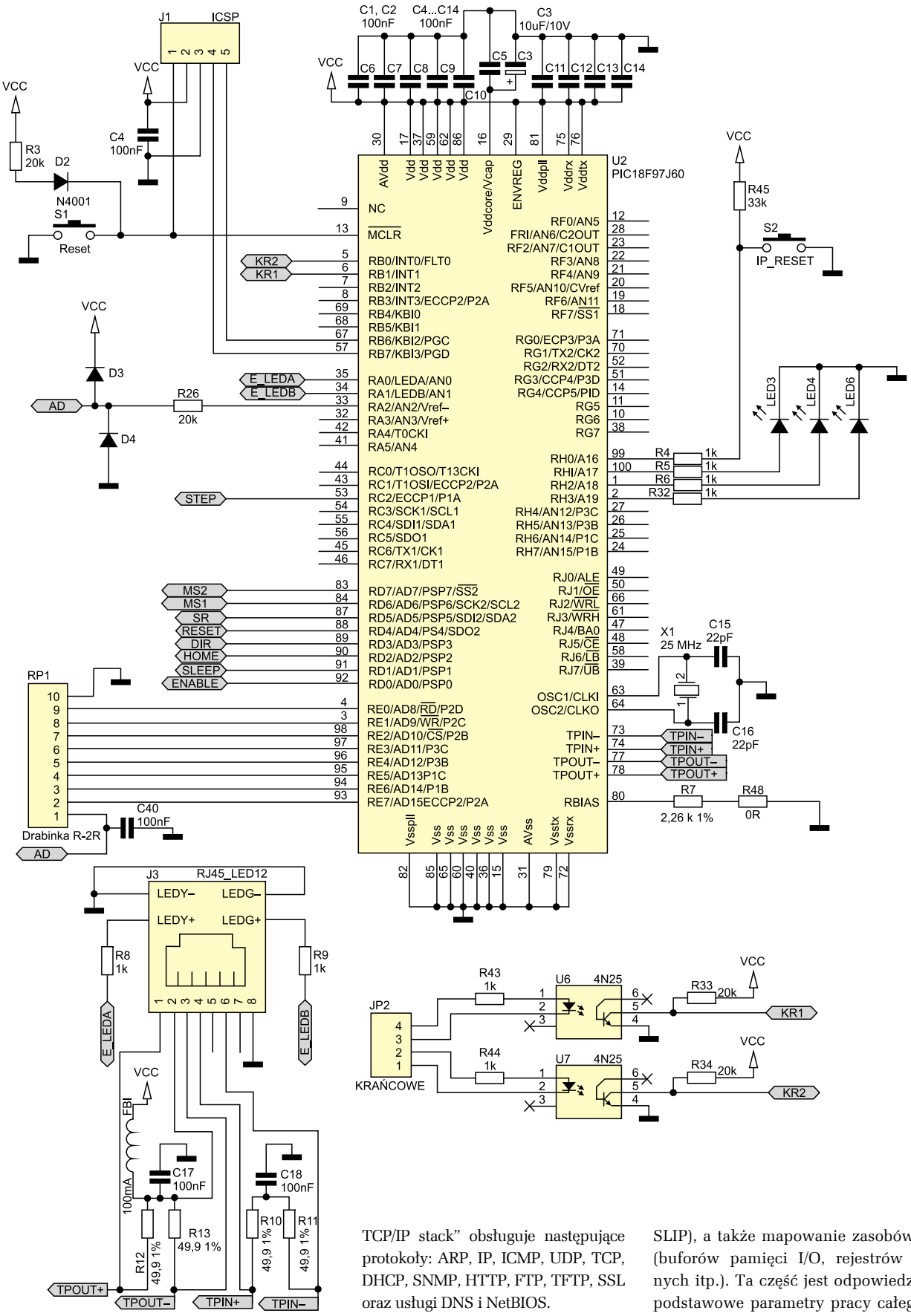
AVT-5284 w ofercie AVT:
AVT-5284A – płytka drukowana

Podstawowe informacje:
- Mikrokontroler PIC18F97J60 z wbudowanym Ethernet PHY (10BaseT) i sprzętowym kontrolerem MAC
- Układ wykonawczy A3977 firmy Allegro MicroSystems
- Maksymalny prąd obciążenia 2,5 A na cewkę
- Napięcie zasilania 12...32 VDC
- Sterowanie układem odbywa się poprzez Ethernet za pomocą TCP/IP

Dodatkowe materiały na CD/FTP:
ftp://ep.com.pl, user: 14039, pass: 4p80b5b5
- wzory płytek PCB
- karty katalogowe i noty aplikacyjne elementów oznaczonych w Wykazie elementów kolorem czerwonym

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

AVT-1585	Sterownik bipolarnego silnika krokowego (EP 8/2010)
AVT-2933	Sterownik silnika krokowego USB (EdW 2/2010)
AVT-1525	Sterownik unipolarnego silnika krokowego (EP 6/2009)
AVT-5137	Sterownik silnika krokowego z interfejsem MODBUS (EP 6-7/2008)
AVT-1314	Najprostszy sterownik silnika krokowego (EP 8/2001)

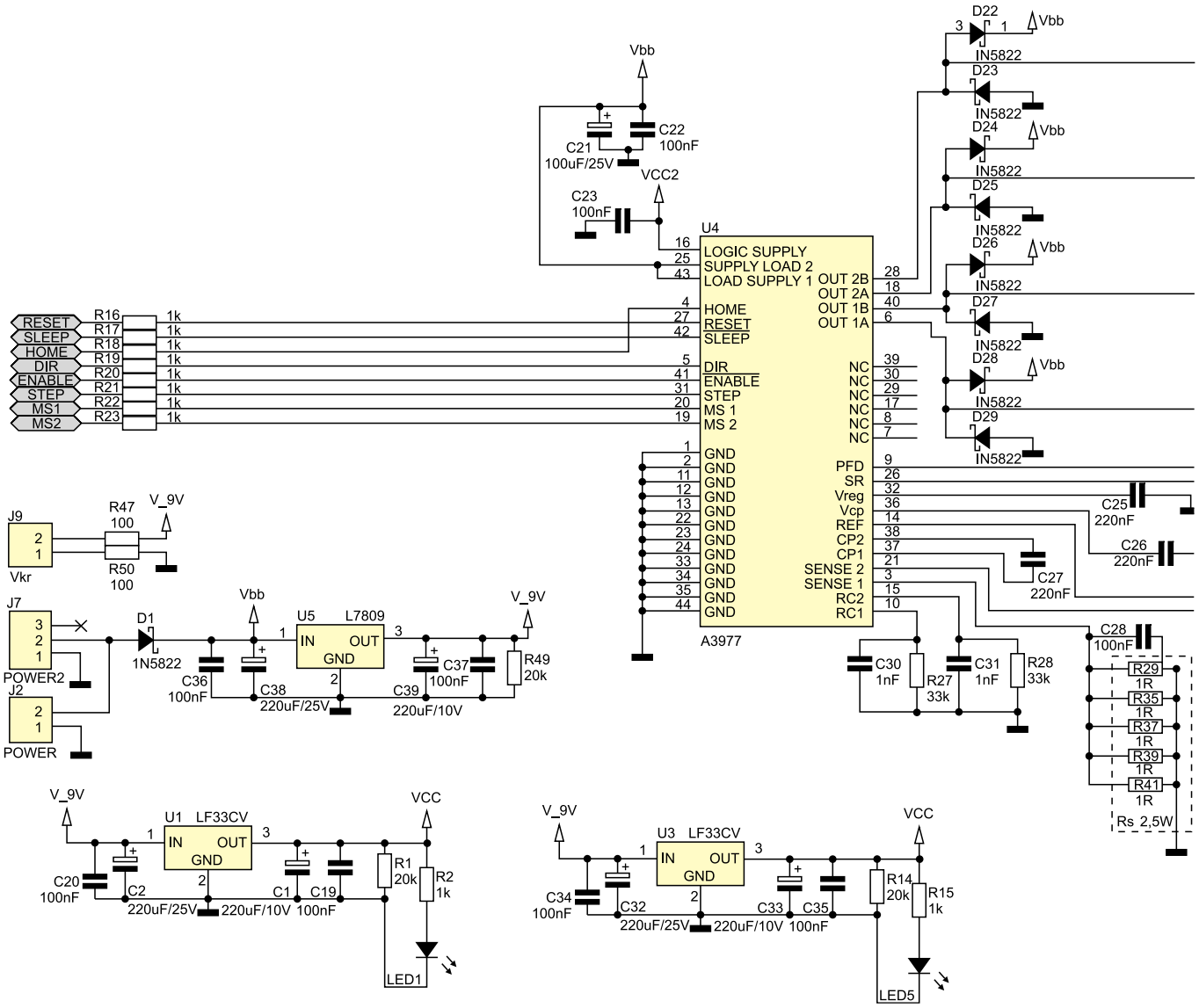


Rysunek 1. Schemat części mikroprocesorowej

Oprogramowanie

Punktem wyjścia dla stworzenia oprogramowania był „Microchip TCP/IP stack". Jest to wsparcie ze strony producenta PIC dla implementacji stosu protokołów TCP/IP. „Microchip TCP/IP stack" obsługuje następujące protokoły: ARP, IP, ICMP, UDP, TCP, DHCP, SNMP, HTTP, FTP, TFTP, SSL oraz usługi DNS i NetBIOS.

Oprogramowanie stosu można podzielić na następujące części: MAC, TCP/IP i Aplikacyjną. Część MAC (*Media Acces Control*) odpowiada za sprzęgnięcie oprogramowania stosu TCP/IP ze sprzętowym modułem komunikacyjnym (na ogół Ethernet PHY, ale może być też SLIP), a także mapowanie zasobów układu (buforów pamięci I/O, rejestrów kontrolnych itp.). Ta część jest odpowiedzialna za podstawowe parametry pracy całego stosu. Tu właśnie należy podać liczbę i rodzaj używanych gniazd, ich bufory, co jest uzależnione od charakteru projektu. Należy pamiętać, że ten element jest odpowiedzialny za udostępnienie zasobów fizycznych pozostałej części oprogramowania i jeżeli go nieodpowiednio skonfigurujemy, to niektó-



Rysunek 2. Sterownik silnika i zasilacz

re zadania aplikacji mogą być wykonywane mniej efektywnie.

Stos TCP/IP to właściwa część oprogramowania. Za jego pomocą są realizowane poszczególne warstwy protokołów sieciowych. Do programisty należy jego skonfigurowanie. Ze stosu można wyeliminować niepotrzebne elementy (np.: protokół HTTP), włączyć lub wyłączyć określone usługi. Ta część udostępnia swoje zasoby warstwie aplikacyjnej za pomocą dobrze opisanego API.

Warstwa aplikacyjna to miejsce gdzie programista realizuje funkcjonalne zadanie urządzenia. Może ona zawierać moduły sprzęgające z otoczeniem sieciowym: serwery i klientów usług sieci TCP/IP, moduły wykonawcze procesów realizujących specyficzne dla urządzenia zadania, komunikujące się poprzez niższe warstwy stosu z otoczeniem.

Poszczególne zadania w aplikacji bazującej na stosie Microchip'a korzystają z czasu procesora na zasadzie zadań kooperacyjnych. Każde zadanie samo w sobie decyduje kiedy zwalnia zasoby dla kolejnego. Stąd na twórcy oprogramowania spoczywa odpowiedzialność dostosowania poszczególnych części aplikacji do tej konwencji. Niedostosowanie się do tego rygoru prowadzi do zatrzymywania wykonywania innych zadań i w konsekwencji nie pozwala na prawidłową obsługę sieci. Na **listingu 1** pokazano program główny, który sekwencyjnie wywołuje funkcje obsługi głównych zadań. Faktycznie, poszczególne procesy są realizowane najczę-

Wykaz elementów

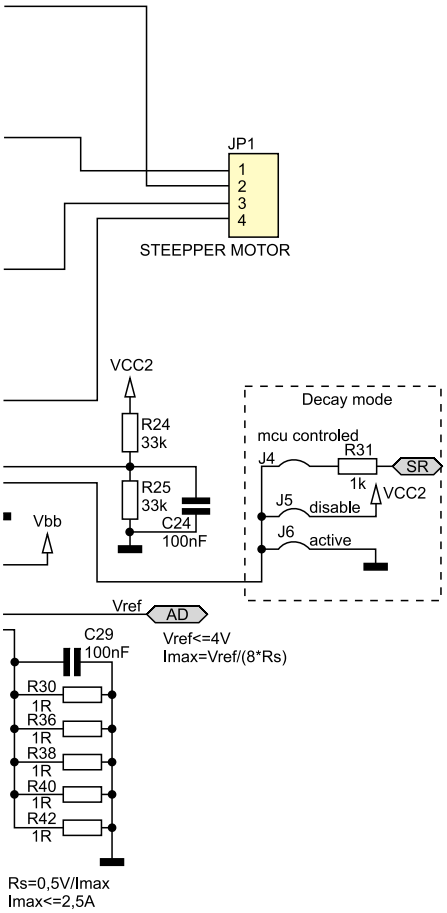
Kondensatory:
C1, C33, C39: 220 µF/10 V
C2, C21, C32, C38: 220 µF/25 V
C3: 10 µF/10 V
C4...C20, C22...C24, C28, C29, C34...C37, C40: 100 nF (SMD 0805)
C15, C16: 22 pF (SMD 0805)
C25...C27: 220 nF (SMD 0805)
C30, C31: 1 nF (SMD 0805)

Rezystory: (SMD 0805)
R1, R3, R14, R26, R33, R34, R49: 20 kΩ
R2, R4...R6, R8, R9, R15...R23, R31, R32, R43, R44: 1 kΩ
R7: 2,26 kΩ 1%
R10...R13: 49,91 kΩ 1%
R24, R25, R27, R28: 33 kΩ
R29, R30, R35...R42: 1 Ω
R47, R50: 100 Ω
R48: 0 Ω
RP1: drabinka R-2R

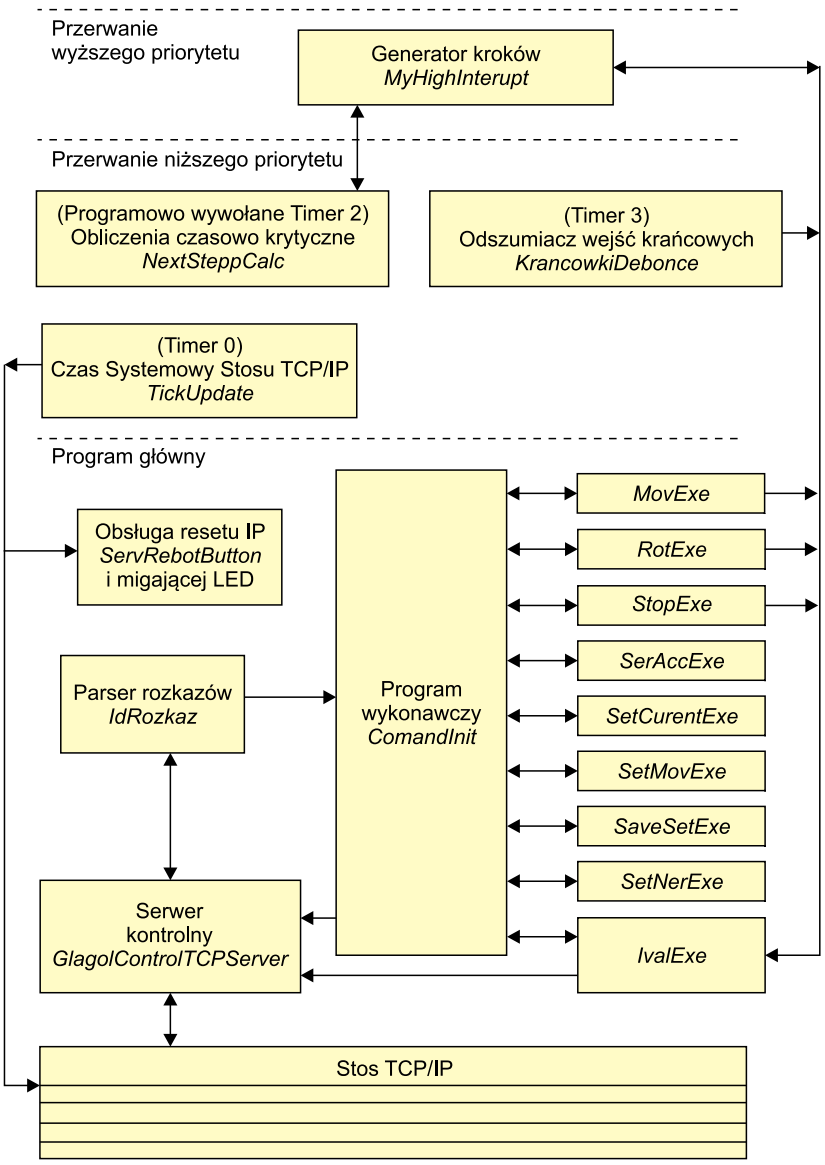
Półprzewodniki:
D1, D22...D29: 1N5822
D2...D4: 1N4001
LED1...LED6: diody LED
U1, U3: LF33CV
U2: PIC18F97J60
U4: A3977
U5: L7809
U6, U7: 4N25

Inne:
FB1: bezpiecznik 100 mA
J1: złącze 5-pin (dla PICkit)
J2, J7, J9, JP1, JP2: np. ARK5
J3: RJ45 (z diodami LED i transformatorem)
J4...J6: zwora SMD
S1: przycisk
X1: kwarc 25 MHz

Na CD: karty katalogowe i noty aplikacyjne elementów oznaczonych w wykazie elementów kolorem czerwonym



Rysunek 2. c.d.



Rysunek 3. Architektura oprogramowania

ściej za pomocą maszyn stanów, a poszczególne stany są zmieniane w kolejnych wywołaniach.

Część interfejsowa

Napisanie aplikacji polega na dopisaniu jej jako ostatniej warstwy stosu. Ogólną strukturę aplikacji pokazano na **rysunku 3**. Oprogramowanie sterownika jest zintegrowane ze stosem Microchip'a za pomocą *Kontrolnego Serwera TCP GlagolControlTCPServer*. Pośredniczy on w komunikacji całej aplikacji wbudowanej z otoczeniem sieciowym. Pozwala na podłączenie się jednego klienta, który jest zdalnym kontrolerem sterownika.

Dane z połączenia sieciowego (obsługiwanego przez serwer) przekazywane są do *Interpretera Zadań*. Przetwarza on dane otrzymane z procesu serwera kontrolnego TCP na rozkazy wewnętrzne sterownika. Tę operację przeprowadza podprogram *IdRozkaz*. Interpretuje on ciąg znaków ASCII na rozkazy.

Rozkazy można podzielić ze względu na sposób ich realizacji na dwie grupy:

- Sekwencyjne. Rozkazy te są wykonywane w kolejności przyjęcia.
- Bezzwłoczne. Są wykonywane najszybciej, jak to możliwe.

W kolejce mieści się 8 poleceń. Opis poszczególnych rozkazów znajduje się w dalszej części artykułu.

Funkcja *IdRozkaz* dla większości rozkazów generuje potwierdzenia ich przyjęcia (za co odpowiada funkcja *ReciveAcknovledge*). Rozwiązanie takie jest ukłonem w stronę modelu komunikacji sieciowej klient-serwer.

Listing 1. Program główny

```
void main(void)
{
  static TICK t = 0;
  InitGlagoltempSetings();  //inicjalizacja ustawień aplikacji sterownika
  InitializeBoard();  //inicjalizacja peryferialna etc.
  TickInit();  //inicjalizacja czasomierza (Microchip)
  InitAppConfig();  //inicjalizacja ustawień sieciowych
  StackInit();  //inicjalizacja stosu (MAC, ARP, TCP, UDP; Microchip)
  DebonuceIni();  //włączenie „odszumiacza" wejść dyskretnych
  while(1)
  {
    // miganie diodą
    if(TickGet() - t >= TICK_SECOND/2ul)
    {
      t = TickGet();
      LED0_IO ^= 1;
    }
    StackTask();  //zadanie stosu TCP/IP
    #if defined(STACK_USE_NBNS)
    NBNSTask();  //opcjonalne zadanie usługi nazw NetBios
    #endif
    GlagolControlTCPServer();  //zadanie serwera kontrolnego
    ComandInit();  //zadanie wykonawcze sterownika
    ServRebotButton();  //Zadanie ręcznego resetu ustawień sieciowych
  }
}
```

Andrzej Puzdrowski
a.m.puzdrowski@interia.pl