

Embedded ethernet, część 3

Dwie poprzednie części cyklu poświęciliśmy na prezentację podstawowych zasad obowiązujących w sieci ethernet oraz przybliżenie komponentów, możliwych do wykorzystania przy budowie własnego układu.

Teraz nadszedł czas na praktyczną realizację systemu.

Rekomendacje:

Artykuł polecemy wszystkim zainteresowanym łącznością poprzez ethernet, którzy chcieliby zapoznać się z podstawami działania sieci i zdobycie tej wiedzy uwieńczyć samodzielnym wykonaniem mini-serwera sieciowego

Trudny wybór

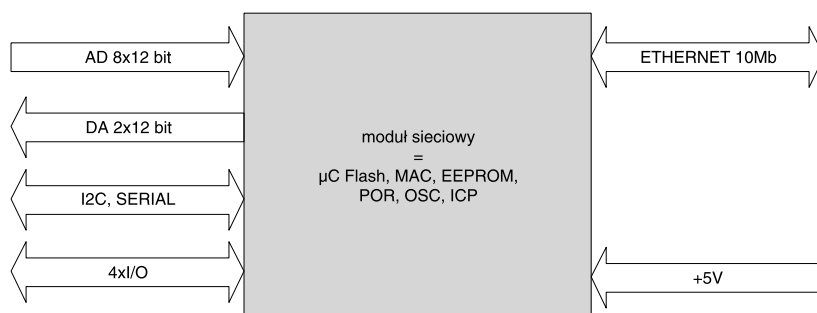
Muszę przyznać, że trochę się zastanawiałem nad funkcjonalnością projektu wykorzystującego łącze Ethernet. Większość dostępnych i oferowanych układów przyjmuje jako jedno ze standardowych rozwiązań funkcję bramki Ethernet-serial, w którym strumień danych warstwy TCP lub UDP jest kierowany wprost na port szeregowy. Jest to wygodne podejście, zwłaszcza, że oferowane sterowniki umożliwiają zazwyczaj „ukrycie” łącza Ethernet pod „maską” wirtualnego portu COM. Pewna część modułów umożliwia z kolei użycie aplikacji HTTP do zarządzania i monitorowania stanu linii we/wy. Przydatność tego rozwiązania ujawnia się zwłaszcza wtedy, gdy możemy samodzielnie modyfikować zawartość strony WWW, a tym samym uatrakcyjnić jej wygląd i funkcjonalność. Często chcielibyśmy pójść krok dalej – móc samodzielnie stworzyć aplikację, która wykonywałaby nasze specyficznie zadania, a niejako „przy okazji” udostępniała interfejs sieciowy. Dokładając do tego możliwość obsługi nie tylko prostych linii we/wy, ale także przetworników A/C i C/A moglibyśmy tworzyć zwarte systemy pomiarowe, swobodnie adaptowalne do zmieniających się potrzeb i wymagań. Niebagatelną sprawą jest także dostęp do tanich i prostych narzędzi uruchomieniowych, które w przypadku bardzo specjalizowanych rozwiązań mogą pozostawać poza zasięgiem większości konstruktorów.

Konstrukcja

Po sprecyzowaniu wymagań został zaprojektowany i praktycznie zrealizowany moduł sieciowy, którego schemat blokowy przedstawiono na **rys. 8**. Moduł komunikuje się ze światem zewnętrznym za pomocą asynchronicznego portu szeregowego, interfejsu I2C, uniwersalnych linii we/wy oraz wejść i wyjść analogowych. Jako interfejs sieciowy zastosowano możliwie najtańsze rozwiązanie, wykorzystujące zawsze obsługiwaną szybkość 10 Mbd. Schemat elektryczny modułu przedstawiono na **rys. 9**.

RTL8019AS

Podstawowym przeznaczeniem modułu jest realizacja dostępu do sieci Ethernet, w oparciu o tanie i łatwo dostępne rozwiązanie. Kierując się tym wymogiem wybrano scalony kontroler sieci RTL8019AS, produkowany przez firmę Realtek. Układ ten, nazywany skrótowo MAC (*Media Access Controller*), zawiera wszystkie składniki niezbędne do wykonania podłączenia do sieci Ethernet o szybkości 10 Mbd. Pomimo, że z założenia jest on przeznaczony do połączenia z magistralą ISA, to jednak nie ma specjalnych problemów, aby wykorzystać go także w systemach nie wyposażonych w takie łącze. Układ ma możliwość pracy w trybie TBase5 (AUI), TBase2 (BNC) lub TBaseT (UTP), który przyjęto w module jako najbardziej praktyczny. Do komunikacji z mikrokontrolerem wybrano 8-bitowy tryb, który jest ustalany za pomo-



Rys. 8. Schemat blokowy modułu interfejsowego

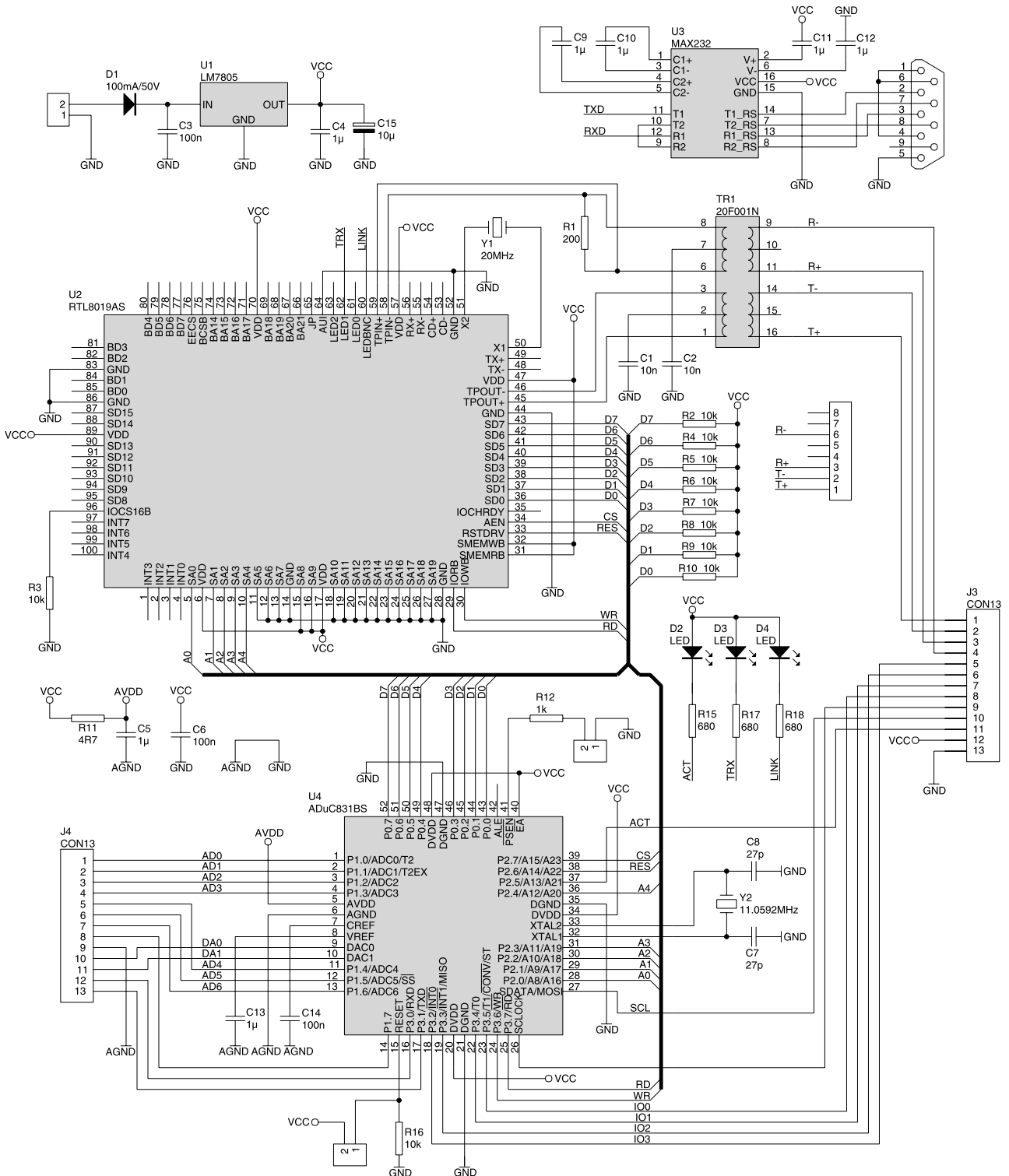
czą rezystora R3. Jako adres bazy, od którego są adresowane wewnętrzne rejestry układu przyjęto wartość 300h, ustaloną stanem linii SA5...19. Pozostałe linie SA0...SA4 (oznaczone na schemacie etykietami A0...A4) służą do wyboru właściwego rejestru RTL8019AS. Wewnętrzny blok MAC kontrolera umożliwia prawidłowe rozpoznanie odebrane-

go strumienia bitów (ciąg w kodzie Manchester) i umieszczenie ramki w buforze kołowym, zrealizowanym w wewnętrznej pamięci SRAM o pojemności 16 kB. MAC dokonuje także weryfikacji sumy kontrolnej CRC32 (przy odbiorze) i jej przeliczenia (przy nadawaniu).

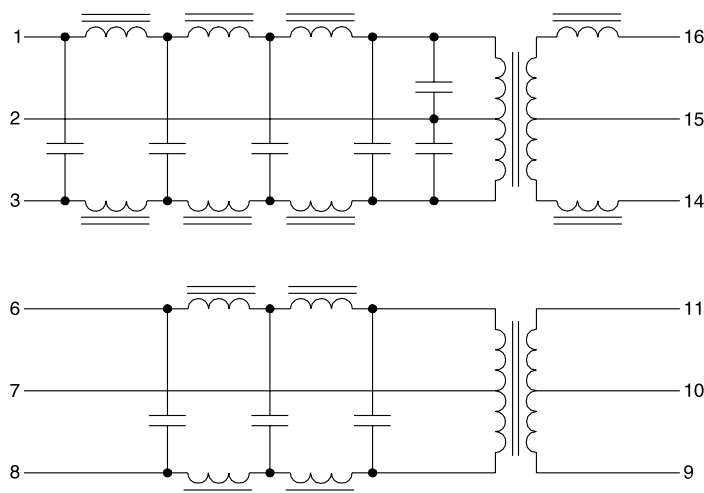
W układzie modelowym zrealizowano z wielu dodatkowych

cech, które posiada w sobie układ RTL8019AS. Nie jest wykorzystywany tryb Plug and Play, obsługa zewnętrznej pamięci konfiguracyjnej EEPROM oraz możliwość zdalnego bootowania poprzez równoległą pamięć BOOT-ROM.

Ze względu na wymaganą izolację elektryczną podłączenie kontrolera do sieci Ethernet musi być



Rys. 9. Schemat elektryczny modułu sieciowego



Rys. 10. Budowa transformatora 20F001N

zrealizowane za pomocą transformatora. W układzie prototypowym wykorzystano model 20F001N, produkowany przez firmę YCL. Jego wewnętrzna struktura jest przedstawiona na rys. 10. Do tłumienia sygnałów poza pasmem działania układu RTL obwód wewnętrzny transformatora uzupełniony jest filtrami 5. i 7. rzędu, odpowiednio dla części odbiorczej i nadawczej. Prawidłowe dopasowanie impedancji wejściowej zapewnia rezystor R1 (na schemacie elektrycznym), którego wartość producent zaleca utrzymywać z dokładnością ok. 0,5%.

ADuC831

Układem integrującym znaczną część funkcjonalności modułu jest mikrokontroler ADuC831, produkowany przez firmę Analog Devices. Jest to układ oparty o rdzeń 8052, wyposażony w 12-bitowy tor przetwarzania A/C i C/A, źródło napięcia referencyjnego oraz dużą pamięć wewnętrzną Flash, EEPROM i RAM. Posiada wbudowany układ zerowania, bootloader, a także sprzętowy port I2C oraz SPI. Uzupełnieniem jest możliwość adresowania do 16 MB zewnętrznej pamięci danych oraz praca z zewnętrznym emulatorem za pomocą tylko jednego wprowadzenia.

Przy konstrukcji modułu przyjęto, że będzie on zawsze stanowił część większego systemu, realizującego specyficzne funkcje aplikacji użytkownika. Dzięki takiemu podejściu można było zrezygnować z bloku dopasowania sygnałów oraz stosowania szeregu zabezpieczeń, normalnie wymaganych w rzeczywistym systemie. Pozwoliło to zna-

cząco uprościć konstrukcję modułu, a zwłaszcza jego części pomiarowej. Analogowe sygnały wejściowe AD0...7 oraz wyjściowe DA0...1 są doprowadzone bezpośrednio do złącza szpilkowego J4. Są one odniesione do linii masy analogowej AGND, podłączonej do J4-9. Jako napięcie referencyjne przyjęto wewnętrzne źródło ADuC831 o wartości nominalnej 2,5 V. Jego stosunkowo niska stabilność termiczna (± 100 ppm) może być w pewnym zakresie skorygowana poprzez wykorzystanie sygnału informującego o temperaturze, pochodzącego z wbudowanego w strukturę czujnika. Do filtrowania napięcia zasilania części analogowej AVDD wykorzystano dwójnik R11, C5.

Wejścia/wyjścia cyfrowe IO0...3 wykorzystują linie P3.2...P3.5 mikrokontrolera. Wyprowadzone linie magistrali I2C – SDA i SCL, wymagają dołączenia zewnętrznych rezystorów podciągających (pull-up). Do złącza J3 doprowadzono także linię ACT, wykorzystywaną jako wskaźnik aktywności modułu. W pewnych warunkach może być ona użyta także w standardowym trybie I/O, o czym będzie mowa w dalszej części.

Mikrokontroler ADuC831 posiada wbudowany układ POR (power-on-reset), co eliminuje potrzebę dołączenia zewnętrznego układu zerującego. Dla potrzeb uruchomieniowych przewidziano jednak doprowadzenie sygnału zerowania, który można wymuszać zwierając styki złącza J7. Jeśli w trakcie zerowania linia PSEN będzie zwarta do GND za pośrednictwem rezystora 1 k Ω (R12 i J5), to mikrokontroler uruchomi wewnętrzny bootloader, za

pomocą którego będzie można aktualizować oprogramowanie. Nowy firmware będzie przesyłany liniami asynchronicznego portu szeregowego RXD i TXD, doprowadzonymi do złącza J4.

Komunikacja uC - MAC

Mikrokontroler łączy się z układem MAC za pośrednictwem linii danych D0...7, adresowych A0...A4 oraz sterujących CS, RD, WR i RES. Rezystory R2...R10 zapewniają prawidłową pracę portu P0 mikrokontrolera (open drain). Przyjęty sposób podłączenia umożliwia komunikację między ADuC831 a RTL8019AS w dwu trybach.

Tryb I/O

W trybie tym wszystkie sygnały są obsługiwane programowo. Za ich właściwą generację odpowiada sekwencyjny sposób sterowania linii portu P0 (dane), P2 (adres) oraz P3 (stroby odczytu i zapisu). Należy zwrócić uwagę, że przy blokowym ustawianiu P2 modyfikowana jest także linia zerowania RES oraz aktywności ACT. Można także oczywiście sterować indywidualnie liniami A0...4 (zakres 0...1Fh), co jednak niepotrzebnie wydłuża czas komunikacji z układem MAC. Przykładowa sekwencja odczytu bajtu może wyglądać następująco (wynik w ACC):

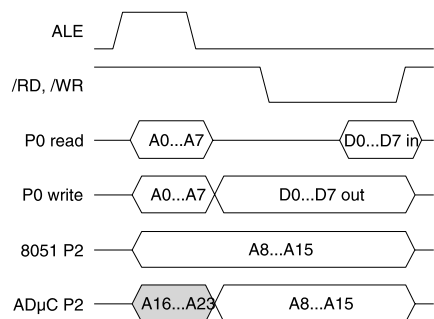
```

mov P0, #0FFh ; P0 na wejście
mov P2, adres ; CS=RES=ACT=0
clr RD ;strob odczytu
mov A, P0 ;odczyt RTL
setb RD ;strob nieaktywny
setb CS ;RTL nieaktywny
    
```

Cykl zapisu danej jest bardzo zbliżony, przy czym aktywny jest sygnał WR. Zasadniczo można zrezygnować z przełączania linii CS, pozostawiając ją w ciągłym stanie wysterowania (ustawiona na 0). Przyjęty sposób umożliwia jednak prawidłową współpracę także z innymi układami dzielącymi port P0. Tryb powyższy można wprost wykorzystać w dowolnym mikrokontrolerze, posiadającym wystarczającą liczbę wolnych linii I/O.

Tryb sprzętowy

W układzie rodziny 8051 podczas dostępu do zewnętrznej pamięci danych generowane są sygnały WR i RD, młodszy adres na P0 (gdy ALE aktywny) oraz starszy na P2. Dane do zapisu i odczytu są wymieniane na P0. Ponieważ RTL wykorzystuje tylko 5 linii adresowych, można je podłączyć do do-



Rys. 11. Cykl zapisu/odczytu ze wewnętrznej pamięci przez mikrokontroler

wolnego portu. Jednakże adres na P0 jest dostępny tylko przez czas aktywnego ALE, co wymagałoby użycia dodatkowego układu zatraskowego. Dlatego też do przesyłania adresu wykorzystano port P2, na którym w 8051 adres pozostaje stabilny przez cały cykl dostępu do pamięci. I tutaj mała niespodzianka – w ADuC tak nie jest. Ze względu na to, że ADuC831 ma możliwość adresowania do 16 MB zewnętrznej pamięci danych, do przesyłania pełnego adresu wykorzystuje się 24 bity (rys. 11). Brakujące 8 bitów adresu, tzn. A16...23, jest przesyłanych na porcie P2, w sposób analogiczny jak A0...7. Do ich zapamiętania wykorzystuje się dodatkowy, drugi układ zatraskowy. Jak więc sobie z tym poradzić? Linie A16...23 są nieistotne w kontakcie z RTL tak długo, dopóki linia A22 pozostaje w stanie niskim. Jest ona bowiem jednocześnie linią zerowania RES, więc jej ustawienie na 1 powoduje restart układu MAC. Należy więc jedynie zadbać o to, aby A22 była cały czas na poziomie 0. Na szczęście wartość adresu A16...23 jest pobierana z rejestru DPP, który po zerowaniu mikrokontrolera przyjmuje wartość 00h. Jeśli tylko nie przekreślimy rejestru DPTR więcej niż 32 razy, wszystko

będzie działać poprawnie (rozkaz *inc DPTR* inkrementuje także wartość rejestru DPP).

Korzystając z powyższych uwag do odczytu z RTL możemy użyć następującej sekwencji rozkazów (pamiętając, że $DPL=A0...7$, $DPH=A8...15$):

```
mov DPH, adres ;DPL nieważny
movx A, @DPTR ;CS=RES=0
```

Zapis wygląda analogicznie. Jeśli tylko linia CS była wcześniej ustawiona na 1, to nie jest wymagane jej powtórne ustawianie po zakończeniu cyklu.

O tym, z którego sposobu obsługi należy korzystać decydują także inne względy, niż tylko minimalizacja czasu dostępu. Jak się okaże przy okazji omawiania oprogramowania modułu, w przyjętej konfiguracji tryb sprzętowy będzie wolniejszy niż I/O.

Elementy dodatkowe

Celem umożliwienia szybkiego uruchomienia i przetestowania modułu, na płytce drukowanej urządzenia zastosowano stabilizator +5 V, interfejs RS232, za pomocą którego programuje się ADuC831 oraz 3 diody sygnalizacyjne LED.

Montaż modułu

Obwód drukowany (schemat montażowy pokazano na rys. 12) został zaprojektowany z myślą o zastosowaniu możliwie jak najłatwiej dostępnych elementów. Z kolei dążenie do miniaturyzacji modułu i rodzaj obwodów podstawowych układów narzuciły potrzebę jego wykonania w postaci montażu mieszanego (przewlekane i SMT). W przypadku układu prototypowego nie stanowi to większego problemu, natomiast w produkcji seryjnej należałoby użyć możliwie jak największej liczby elementów SMD.

Ręczny montaż modułu najlepiej

WYKAZ ELEMENTÓW

Rezystory SMD rozmiar 0805

- R1: 200Ω/0,5%
- R2...R10, R16: 10kΩ
- R11: 4,7Ω
- R12: 1kΩ
- R15, R17, R18: 680Ω

Kondensatory SMD rozmiar 0805

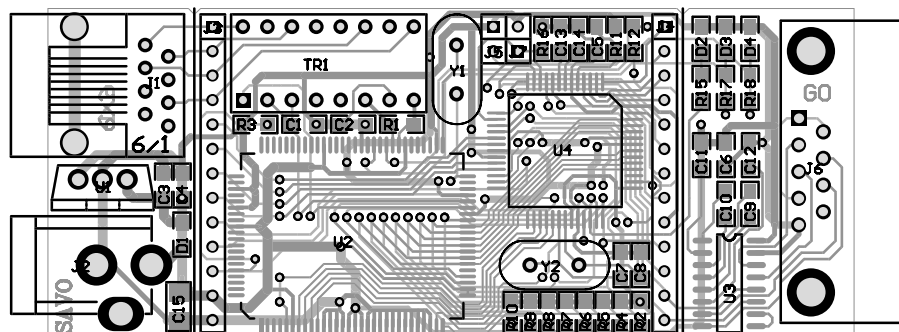
- C1, C2: 10nF
- C3, C6, C14: 100nF
- C4, C5, C9...C13: 1μF
- C7, C8: 27pF
- C15: 10μF/6V rozmiar A

Półprzewodniki

- U1: L7805 TO220
- U2: RTL8019AS QFP100
- U3: MAX232 SO16
- U4: ADuC831BS PQFP52
- D1: LL4148 MINIMELF
- D2, D3, D4: LED 0805 czerwone

Inne

- J1: RJ45 gniazdo 8p8s kątowe
- J2: CON2 gniazdo zasilające
- J3, J4: CON13 listwa jednorzędowa x13
- J5, J7: CON2 listwa jednorzędowa x2
- J6: DB9 gniazdo żeńskie do druku
- TR1: 20F001N transformator DIP14
- Y1: 20MHz
- Y2: 11,0592MHz



Rys. 12. Schemat montażowy płytki drukowanej

rozpocząć od wlutowania układów U2, U3 i U4, a następnie pozostałych elementów powierzchniowych. TR1, złącza J1, J2, J3, J4 i J6 powinny być zamontowane na samym końcu. Przyjęcie innej kolejności może utrudnić bądź wręcz uniemożliwić złożenie całego układu. Aby było możliwe stworzenie „kanapki” z modułem sieciowym, listwy J3 i J4 powinny być zamontowane od spodniej strony obwodu drukowanego.

Elementy wspomagające uruchomienie mogą być montowane opcjonalnie, jeśli tylko układ docelowy, w którym będzie pracował moduł, posiada własny interfejs szeregowy, gniazdo RJ45 oraz źródło napięcia +5 V. W takim przypadku płytka drukowana może zostać zmniejszona poprzez jej przycięcie w miejscach zaznaczonych na rys. 12.

Sposób uruchomienia i szczegóły oprogramowania zostaną przedstawione za miesiąc w ostatniej części artykułu.

Grzegorz Oleszek
grzegorz@savo.pl