

# Zdalny system pomiarowy z interfejsem Ethernet,

## część 1

### AVT-5111

Układy monitoringu i zdalnego sterowania zyskały nową jakość wraz z upowszechnieniem szerokopasmowego dostępu do Internetu. Podobnie stało się z rozproszonymi systemami pomiarowymi pozwalającymi sterować nawet złożonymi procesami produkcyjnymi z poziomu jednej aplikacji.

W artykule prezentujemy przykładowy projekt systemu pomiarowego, w którym sterowanie odbywa się poprzez Internet.

**Rekomendacje:** projekt mimo sporych możliwości i dość rygorystycznych założeń technicznych stanowi doskonale pole doświadczalne dla projektantów właśnie ujarzmiających Ethernet.

#### PODSTAWOWE PARAMETRY

- Płytko o wymiarach: 136x128 (151 z wypustkami) mm
- Podstawowe cechy systemu pomiarowego:
  - liczba modułów połączonych wspólną magistralą RS485: od 1 do 32,
  - liczba wejść/wyjść jednego modułu: 4 wejścia z pętlą prądową 0/4...20 mA, 2 wejścia stykowe oraz 2 wyjścia przekaźnikowe,
  - możliwość zapisywania kolejnych pomiarów w wewnętrznej pamięci EEPROM (w każdym module),
  - stempel czasowy dodawany do każdego pomiaru,
  - możliwość odczytu mierzonych wartości na wbudowanym wyświetlaczu,
  - zabezpieczenie zapisanych parametrów i nastaw zegara czasu rzeczywistego przed zanikami napięcia zasilania,
  - zdalny podgląd stanu czujników oraz ustawianie wyjść przekaźnikowych,
  - obsługa systemu przez aplikację uruchomioną na komputerze PC



Opisany niżej projekt był tworzony z myślą o bardzo konkretnym zastosowaniu: miał to być system pomiarowy o charakterze przemysłowym, w którym czujnikami są różnego rodzaju przetworniki pomiarowe pracujące w standardzie pętli prądowej 0/4...20 mA. Mimo konkretnego zastosowania lektura artykułu może być przydatna podczas budowy wielu innych podobnych urządzeń współpracujących z siecią Ethernet. Porusza również problematykę interfejsu RS485 oraz protokołu MODBUS.

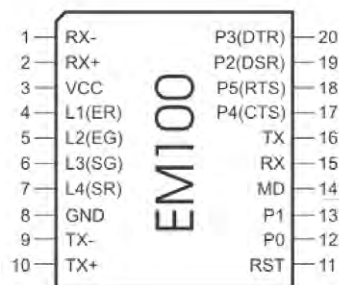
Podstawową trudnością w implementacji interfejsu sieciowego Ethernet jest duży stopień skomplikowania protokołu TCP/IP. Producenci elektroniki nie śpią jednak i na rynku ukazują się co raz to nowsze rozwiązania mające ułatwić życie projektantom. Jedną z wiodących firm w tej tematyce jest Tibbo Technology – firma pochodząca z Tajwanu. Sedno działania modułów ethernetowych Tibbo polega na zamianie trudnego w obsłudze interfejsu oraz protokołu TCP/IP na banalnie prosty i znany każdemu elektronikowi interfejs szeregowy typu RS232. Właśnie moduł ethernetowy tej firmy został wykorzystany w opisywanym urządzeniu.

#### Założenia projektowe

Głównym założeniem autora było zbudowanie systemu uniwersalnego, z możliwością łatwej rozbudowy o kolejne wejścia pomiarowe. Duży nacisk został również położony na niezawodność systemu,

tak aby było możliwe jego zastosowanie w środowisku przemysłowym. Podłączenie systemu do sieci lokalnej lub do Internetu ma ułatwić dostęp do mierzonych parametrów oraz umożliwić zdalne monitorowanie i sterowanie obiektu.

Należy także zauważyć, że mimo iż system był projektowany z myślą o instalacjach przemysłowych, nie ograniczyło to w żaden sposób możliwości jego pracy w warunkach mniej wymagających, np. domowych. Można sobie tu wyobrazić chociażby kontrolę temperatury lub wilgotności w pomieszczeniach, a przy użyciu wyjść przekaźnikowych, programowe lub ręczne sterowanie różnymi urządzeniami.



Rys. 1. Moduł ethernetowy EM100 i jego wyprowadzenia

## Moduł ethernetowy Tibbo EM100

Moduł EM100 jest uniwersalnym interfejsem Ethernet – port szeregowy (rys. 1). Jest on przeznaczony do bezpośredniego montażu na płytce drukowanej (istnieją również moduły z własną obudową i złączami RJ-45 i szeregowym). Moduł należy do grupy najprostszych, pracujących z prędkością 10 Mbit/s. Podłączenie do sieci Ethernet wymaga jedynie zewnętrznego złącza RJ-45 – w swoim wnętrzu integruje również wymagany w sieciach tego typu transformator izolujący. Moduł zawiera własny mikrokontroler z wewnętrznym oprogramowaniem (*firmware*). Oprogramowanie to realizuje całość zadań potrzebnych do pracy modułu jako pośredniczącego między interfejsami Ethernet i szeregowym. Są to obsługa stosu TCP/IP, programowanie modułu poprzez port szeregowy lub sieciowy, sygnalizacja błędów. Port szeregowy modułu umożliwia bezpośrednie podłączenie do mikrokontrolera (np. do jego portu UART), a za pomocą transceiverów RS232 lub RS485 do odpowiednich magistral. Interfejs szeregowy EM100 może pracować z prędkościami od 150 do 115200 bitów/s. W typowych zastosowaniach moduł jest połączony z urządzeniem posiadającym interfejs szeregowy, natomiast port Ethernet łączy się z wewnętrzną siecią komputerową ewentualnie Internetem. W zasadzie więc każdy potencjalny host sieciowy może się komunikować z modułem i urządzeniem będącym niejako „za nim”.

Głównym zadaniem opisywanego modułu jest zapewnienie przezroczystej transmisji danych między hostem a urządzeniem szeregowym podłączonym do modułu. Transmisja danych odbywa się za pośrednictwem dwóch pamięci buforowych – w module EM100-03 użytym w prototypie pamięć ta ma pojemność 2x510 bajtów. Użycie buforów staje się konieczne wobec różnych prędkości z jakimi pracuje łącze ethernetowe i szeregowe. Dane przesyłane z jednego bądź drugiego

kierunku, muszą być w jakiś sposób buforowane i gromadzone, tak aby przy nadarzającej się okazji mogły być wysłane w jednym lub kilku pakietach do sieci Ethernet lub – z drugiej strony – być sukcesywnie przesyłane do powolnego portu szeregowego.

Tak jak każde urządzenie sieciowe, moduł posiada własny adres fizyczny MAC oraz musi mieć zdefiniowany adres IP, aby mógł się komunikować z innymi urządzeniami w sieci. Numer IP można zdefiniować ręcznie, bądź też wykorzystać protokół DHCP do nadania automatycznego. Aby przesyłać dane pomiędzy hostem sieciowym i urządzeniem szeregowym podłączonym do modułu EM100, należy wcześniej zestawić połączenie. Moduł EM100 pozwala na pracę tylko z jednym połączeniem. Nie jest to żadne ograniczenie, wystarczy tylko sobie uzmysłowić, że port szeregowy nie jest elementem współdzielonym i w danej chwili może z niego korzystać jedynie jeden program/urządzenie.

Możliwa jest praca z protokołem połączeniowym TCP (*Transmission Control Protocol*) oraz bezpołączeniowym UDP (*User Datagram Protocol*). Chyba najogólniej różnice między tymi protokołami można ująć w następujący sposób: TCP jest protokołem pewnym i niezawodnym, natomiast bezpołączeniowy UDP nieco szybszym, ale nie dającym pewności że dane dotarły do celu. Wynika to z prostego faktu: urządzenia (np. komputery) wymieniające się danymi, w przypadku TCP potwierdzają sobie fakt poprawnego odebrania danych. Generują przy tym dodatkowy ruch w sieci, ale jest to cena jaką się płaci za niezawodność. Protokół TCP jest szeroko wykorzystywany w sieci Internet przez protokoły wyższej warstwy: FTP, HTTP, POP3.

W przypadku UDP sprawa wygląda nieco inaczej. Każdy host może w dowolnej chwili wysłać pakiet UDP do wielu hostów bez wcześniejszego nawiązywania po-

łączenia. Dodatkowo, nadawca nie otrzymuje żadnego potwierdzenia, że dane zostały poprawnie odebrane przez odbiorcę. Mimo tych cech protokół UDP jest szeroko wykorzystywany chociażby w przypadku mediów strumieniowych, takich jak wideo czy dźwięk. Strata kilku pakietów często nie będzie problemem, a transmisja może odbywać się z dużą prędkością.

W dokumentacji firmy Tibbo określenie „połączenie” jest także używane w stosunku do protokołu UDP, co nie do końca jest prawdziwe. Wynika to z pewnej modyfikacji protokołu UDP zaimplementowanego w modułach firmy. W przypadku odebrania pakietu UDP przez moduł, zachowuje się on tak, jakby zostało nawiązane połączenie i wie komu przesłać własne pakiety UDP (czyli ewentualne odpowiedzi). W tym momencie staje się oczywiste, że w opisywanym urządzeniu wykorzystano protokół TCP – pewność działania systemu jest jednym z jej priorytetów.

Moduł EM100 obsługuje następujące protokoły: TCP, UDP, DHCP, ARP, ICMP. Oferuje on 3 tryby pracy dotyczące *routingu* (jest to angielskie określenie opisujące trasowanie pakietów, czyli odpowiednie ich kierowanie do hosta docelowego). Definiują one czy DS (*Device Server* – określenie zamiennie używane w stosunku do modułu EM100) będzie akceptował nadchodzące połączenia oraz czy, i w jaki sposób będzie możliwe otwieranie własnych połączeń.

W trybie pierwszym (serwer), moduł odpowiada na wszystkie połączenia wywoływane przez dowolnego hosta sieciowego. Połączenie musi odbywać się na ustawionym w konfiguracji porcie TCP (0...65534). W trybie tym moduł nigdy nie próbuje otwierać własnych połączeń.

W trybie drugim (klient i serwer) oba rodzaje połączeń są dozwolone. Praca jako klient umożliwia inicjowanie połączeń ze wskazanym hostem w sieci.

Istnieje jeszcze tryb trzeci, czyli praca tylko jako klient. Moduł odrzuca wszystkie przychodzące żądania połączenia. Tylko sam może je inicjować.

Oczywiście wybór trybu pracy nie jest sprawą obojętną dla działania systemu. Każdy z dwóch głównych rodzajów pracy (klient/serwer) ma swoje zalety i wady. Przykładowo w trybie serwera odczytywanie stanu czujników musi być inicjowane przez aplikację uruchomioną na zdalnym hoście. W trybie tym utrudnione jest natychmiastowe powiadomienie hosta np. o zmianie stanu czujnika. Rozwiązaniem, ale niestety niedoskonałym, jest cykliczne odpytywanie serwera, który dopiero w odpowiedzi na to zapytanie może przesłać informację o zaistniałej sytuacji. Oczywiście takie cykliczne odpytywanie niepotrzebnie generuje ruch sieciowy, choć gdy nie jest ono zbyt szybkie nie stanowi to zbyt dużego problemu.

Inicjowanie przez DS połączenia w trybie klienta również ma pewną niedogodność. Jest nią, po pierwsze, konieczność wcześniejszego zapisania w jego konfiguracji adresu IP komputera hosta, co utrudnia dość mocno możliwość obsługi systemu z różnych hostów (oczywiście nie jednocześnie – moduł umożliwia tylko

jedno połączenie w tym samym czasie), a niekiedy wręcz uniemożliwia. Ta ostatnia sytuacja występuje, gdy komputer hosta znajduje się za routerem pełniącym także funkcję translatora adresów (tzw. NAT – Network Address Translation). Wtedy również host będzie miał (najczęściej) adres IP z puli niepublicznej (np. 192.168.x.x). Sprowadza się to do tego, że taki host nie będzie „widoczny” w sieci Internet i nie będzie możliwe odpowiednie skonfigurowanie modułu w celu połączenia z hostem zdalnym. W omawianym urządzeniu zdecydowano o użyciu bardziej uniwersalnego trybu serwera. Odpytywanie można z powodzeniem przeprowadzać np. co sekundę, co pozwala na stosunkową szybką reakcję obsługi.

Moduł EM100 posiada wewnętrzną pamięć nieulotną, w której zapisanych jest szereg ustawień i parametrów definiujących sposób pracy urządzenia. Ustawienia te można zmieniać po wejściu w tryb programowania. Programowanie można przeprowadzić zdalnie, poprzez port ethernetowy, bądź też lokalnie

STX	ROZKAZ/ODPOWIEDZ	CR
-----	------------------	----

Rys. 2. Format komunikatu modułu EM100 w trybie programowania

Tab. 1. Zestawienie rozkazów DS

Kod operacji	Nazwa rozkazu	Opis
L	Login	Uwierzytelnia hosta hasłem i otwiera sesję programowania zdalnego przez sieć
O	Logout	Kończy sesję programowania (sieciowego lub przez port szeregowy)
E	Reboot	Powoduje ponowne uruchomienie modułu
I	Initialize	Przywraca ustawienia do fabrycznych (z kilkoma wyjątkami)
S	Set Setting	Ustawia wybrany parametr
G	Get Setting	Odczytuje wybrany parametr
P	Parameter	Ustawia wybrany parametr (ale bez zapisu do pamięci nieulotnej)
X	Echo	Zwraca status modułu, pełni funkcję informacyjną
U	Status	Zwraca dodatkowe, rozszerzone informacje o statusie
B	Buzz	Powoduje szybkie miganie diod świecących podłączonych do modułu (pełniących normalnie funkcję kontrolną); używane do rozpoznawania modułu
R	Reset Overflow Flags	Kasuje flagi przepełnienia buforów sieciowego i szeregowego
A	Assign IP-address	Powoduje natychmiastową zmianę IP modułu (autoryzacja na podstawie adresu MAC i/lub hasła)
W	Select in Broadcast Mode	Wybiera moduł, zdefiniowany adresem MAC, do dalszego programowania w trybie rozgłoszeniowym UDP
V	Get Firmware Version	Zwraca numer oprogramowania wewnętrznego (firmware'u)
N	Jump To Net-loader	Powoduje uruchomienie NetLoadera – fragmentu firmware'u odpowiedzialnego za ładowanie nowej wersji firmware'u przez sieć
J	Notification	Komunikat wysyłany przez moduł, raportujący zmianę stanu linii I/O modułu

Tab. 2. Zestawienie możliwych odpowiedzi

Kod odpowiedzi	Nazwa	Opis
A	OK	Rozkaz wykonany poprawnie
C	Error	Błędna komenda
R	Rejected	Rozkaz odrzucony przez DS
D	Denied	Dostęp zabroniony
F	Failed	Wykonanie rozkazu nie powiodło się

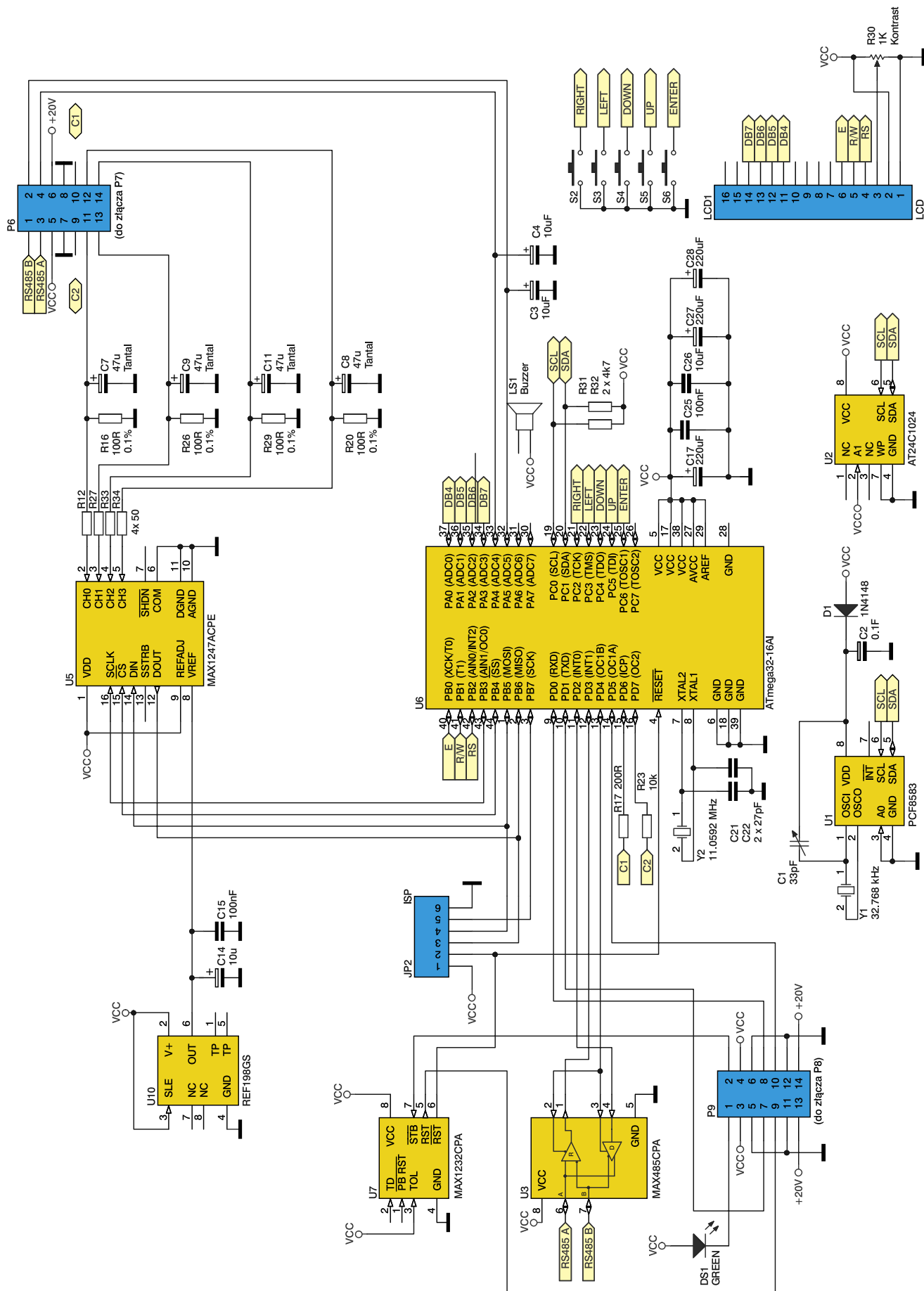
poprzez port szeregowy. Wejście w tryb programowania od strony interfejsu szeregowego można zrealizować na dwa sposoby. Pierwszy to wymuszenie na wyprowadzeniu MD modułu stanu niskiego przez co najmniej 100 ms. Drugi sposób to wysłanie portem szeregowym specjalnej sekwencji programującej. Drugi sposób może być również programowo zablokowany. Odpowiedzialny jest za to jeden z parametrów modułu. Programowanie zdalne (poprzez sieć) jest również możliwe w kilku trybach zarówno przy użyciu protokołu TCP i UDP.

W prototypie używano do programowania tylko portu szeregowego i sterowania linią MD. Wszystkie rozkazy i odpowiedzi modułu EM100 przesyłane przez port szeregowy mają format przedstawiony na rys. 2. Każdy komunikat rozpoczyna się bajtem STX (jest to znak ASCII o kodzie 2), natomiast kończy się znakiem CR – powrót karetki (*Carriage Return*), kod 13.

Jako przykład zostaną pokazane komunikaty przesyłane podczas zapytania o numer IP modułu i jego odpowiedź:

```
Urządzenie szeregowo --->
DS: STXGIPCR
DS --->  Urządzenie szeregowo: STXA192.168.12.10CR
```

Zapytanie o numer IP ma postać trzech znaków ASCII „GIP”, litera G, będąca kodem operacji oznacza skrót od angielskiego słowa GET, czyli pobierz, następnie dwa znaki „IP” określające konkretnie o jaki parametr chodzi. Pierwszy znak odpowiedzi (pomijając znak [STX]) sygnalizuje status odpowiedzi: w tym przypadku „A” oznacza, że komenda została poprawnie zrealizowana, a następnie przesłany jest adres IP w notacji dziesiętnej z kropkami. W tab. 1 zestawiono wszystkie możliwe rozkazy, a w tab. 2 możliwe odpowiedzi.



Rys. 3. Schemat ideowy - procesor



Nie wszystkie podane rozkazy można wykonać w sesji programowania szeregowego, z kolei część z nich wymaga spełnienia dodatkowych warunków, jak np. wcześniejszego załogowania się. Pełne informacje na ten temat można znaleźć w dokumentacji modułu.

W omawianym projekcie programowanie modułu odbywa się tylko poprzez port szeregowy – nie wykorzystano możliwości programowania przez sieć, gdyż nie było to potrzebne. Jeśli konieczna staje się zdalna zmiana jakiegoś parametru, jest to osiągane w ten sposób, że najpierw specjalny komunikat jest przesyłany przez moduł do mikrokontrolera, a następnie mikrokontroler rozpoczyna sesję programowania szeregowego i przesyła do DS wymagane rozkazy. Takie rozwiązanie wymaga wszakże ustawienia (przy użyciu klawiatury i wyświetlacza modułu MASTER) dwóch parametrów komunikacyjnych, jakimi są adres IP oraz port TCP modułu EM100. Po skonfigurowaniu tych parametrów będzie możliwa dalsza, już zdalna praca.

## Budowa

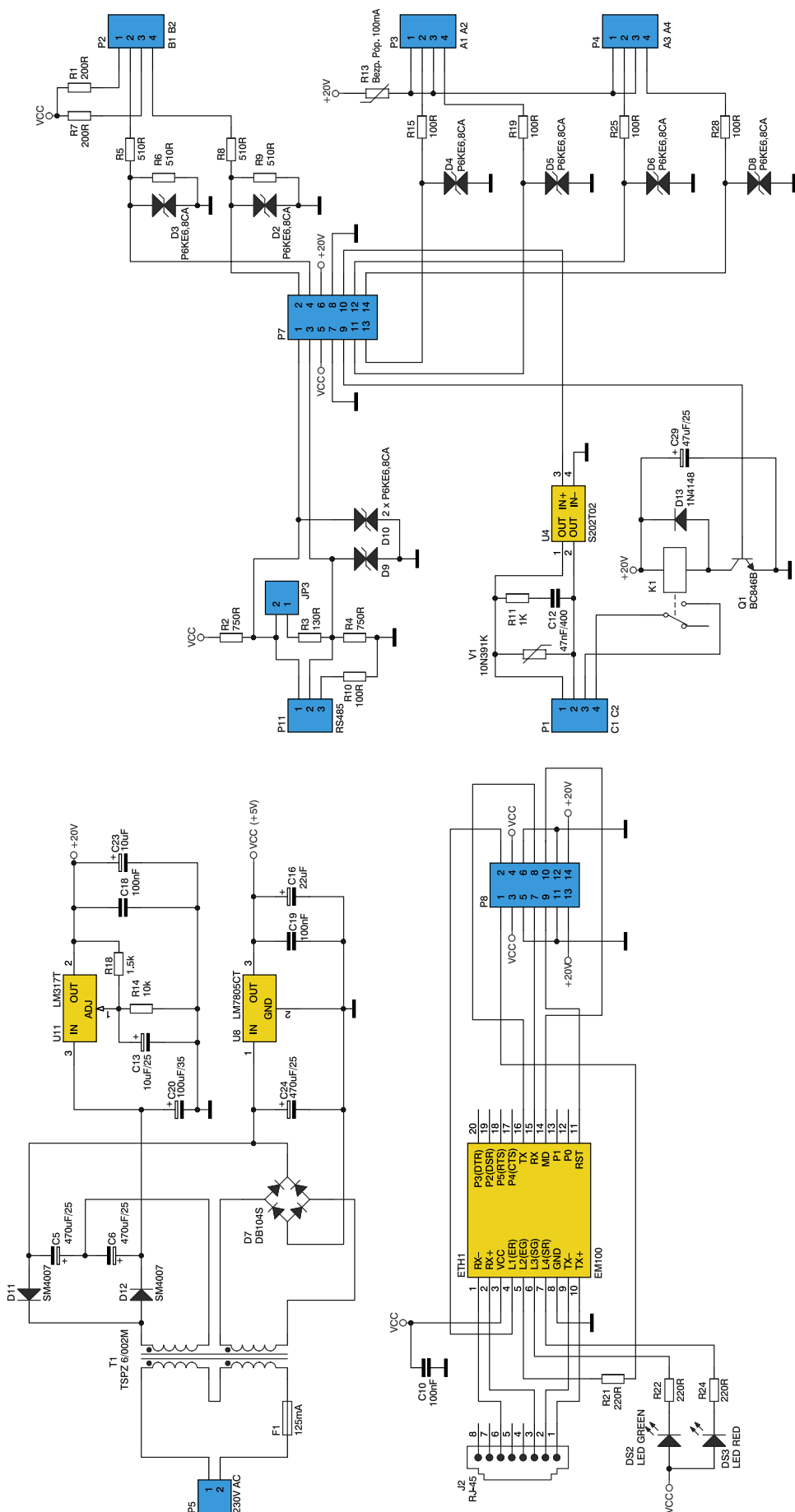
Schemat ideowy systemu pomiarowego przedstawiono na rys. 3 (procesor) i rys. 4 (zasilacz i moduły we/wy). Poniżej zostanie omówiona budowa interfejsów pomiarowych.

**Analogowe wejścia pętli prądowej (kanały A1 do A4).** Typowe przetworniki pomiarowe stosowane w układach automatyki przemysłowej wykorzystują kilku standardów sygnału wyjściowego. Można do nich zaliczyć:

- sygnał prądowy 0/4...20 mA,
- sygnał napięciowy 0...1/5/10 V.

Urządzenie będące przedmiotem tego projektu przystosowano do współpracy z najbardziej rozpowszechnionymi przetwornikami z pętlą prądową 0/4...20 mA. Ich popularność jest efektem kilku niezaprzeczalnych zalet:

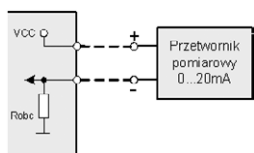
- można je zasilac tymi samymi przewodami, którymi przesyłana jest informacja pomiarowa,



Rys. 4. Schemat ideowy – interfejs ethernetowy i interfejsy we/wy

- ze względu na prądowy charakter sygnału informacyjnego, przetworniki są niewrażliwe na wartość rezystancji przewodów doprowadzających, a co z tego wynika, możliwe jest uzyskanie dużych odległości,
- są w dużym stopniu niewrażliwe na zakłócenia (również ze względu na prądowy charakter i typowo małą impedancję wyjściową i obciążenia) oraz na ewentualne wahania napięcia zasilającego,
- mają szeroki zakres napięć zasilających.

Należy jednak zauważyć, że przyjęcie przetworników z pętlą prądową nie zamyka możliwości zastosowania w razie potrzeby prze-



Rys. 5. Sposób podłączenia przetwornika pomiarowego z pętlą prądową

tworników o wyjściu napięciowym. Niewielka modyfikacja interfejsu pomiarowego polegająca na zmianie wartości dwóch rezystorów oraz jednego kondensatora pozwoli na przyłączenie również wszystkich wymienionych przetworników napięciowych.

Przetworniki pomiarowe typu prądowego mają różne zakresy dopuszczalnych napięć zasilających i impedancji obciążenia. Zwykle napięcie zasilające może się mieścić w zakresie 18...30 V (lub 15...35 V), natomiast impedancja wyjściowa nie może być wyższa od 500  $\Omega$  dla napięcia zasilającego 24 V. Sposób zasilania i podłączenia takiego przetwornika pokazano na rys. 5. Przetwornik do swojej pracy wymaga jedynie interfejsu dwuprzewodowego. W zależności od rodzaju, przetwornik może wymagać oddzielnego zasilania z sieci energetycznej lub też może być zasilany bezpośrednio prądem pętli. Spadek napięcia na rezystorze R0bc spowodowa-

ny przepływem prądu pętli może być następnie podany na wejście przetwornika analogowo-cyfrowego w celu dalszej obróbki. Przy maksymalnej dozwolonej rezystancji rezystora R0bc równej 500  $\Omega$  i prądzie pętli 20 mA, spadek napięcia wynosi 10 V. W praktyce stosuje się mniejsze wartości R0bc, np. 100  $\Omega$ . Tak też uczyniono w wykonanym prototypie. Taka wartość rezystancji daje napięcie na wejściu przetwornika w zakresie 0...2 V, impedancja wejściowa kanału pomiarowego jest niższa, co jest wskazane ze względu na mniejsze przenikanie zakłóceń oraz daje jeszcze dodatkowy margines bezpieczeństwa przy przesterowaniu i typowym napięciu zasilającym przetwornik A/C równym 5 V. W tak prostym rozwiązaniu toru wejściowego, dokładność pomiaru jest uzależniona od trzech czynników: tolerancji rezystancji R0bc, dokładności źródła napięcia odniesienia przetwornika A/C oraz samego przetwornika.

#### WYKAZ ELEMENTÓW

##### moduł Master oraz Slave

##### Rezystory

R1, R7, R17: 200  $\Omega$  (1206)  
 R3: 130  $\Omega$   
 R5, R6, R8, R9: 510  $\Omega$  (1206)  
 R11: 1 k $\Omega$   
 R12, R27, R33, R34, R50: 50  $\Omega$  (1206)  
 R14, R23: 10 k $\Omega$  (1206)  
 R10, R15, R19, R25, R28: 100  $\Omega$   
 R16, R20, R26, R29: 100  $\Omega$ /0,1%/0,6 W  
 R18: 1,5 k $\Omega$  (1206)  
 R31, R32: 4,7 k $\Omega$  (1206)  
 R30: potencjometr montażowy 1 k $\Omega$

##### Kondensatory

C3, C4, C14, C26: 10  $\mu$ F/16 V tantalowy (obudowa B)  
 C5, C6, C24: 470  $\mu$ F/25 V  
 C7, C8, C9, C11: 47  $\mu$ F/25 V tantalowy (obudowa D)  
 C12: 47 nF/400 V  
 C13: 10  $\mu$ F/25 V  
 C15, C18, C19, C25: 100 nF (1206)  
 C16: 22  $\mu$ F/16 V tantalowy (obudowa C)  
 C20: 100  $\mu$ F/35 V  
 C21, C22: 27 pF (1206)  
 C23: 10  $\mu$ F/25 V tantalowy (obudowa D)  
 C17, C27, C28: 220  $\mu$ F/10 V

C29: 47  $\mu$ F/25 V

##### Półprzewodniki

D13: 1N4148  
 D2...D6, D8...D10: transil P6KE6.8CA  
 D7: mostek Graetza DB104S (SMD)  
 D11, D12: SM4007  
 Q1: BC846B SMD  
 U2: AT24C1024 (DIP-8)  
 U3: MAX485 (DIP-8)  
 U4: przekaźnik półprzewodnikowy S202T02 (Sharp)  
 U5: MAX1247 (DIP-16)  
 U6: ATmega32-16AI (TQFP-44, zaprogramowany)  
 U8: LM7805  
 U10: REF198  
 U11: LM317T

##### Inne

V1: warystor 10N391K  
 Y2: kwarc 11,0592 MHz  
 F1: bezpiecznik 125 mA  
 R13: bezpiecznik polimerowy 100 mA  
 K1: przekaźnik 24 V/6 A (Finder)  
 LCD1: wyświetlacz LCD 2x16 znaków PLED (zgodny z HD44760)  
 LS1: buzzer 5 V  
 P1, P2, P3, P4: złącze typu Terminal Block 4 pin.  
 P11: złącze typu Terminal Block 3 pin.  
 P5: złącze typu Terminal Block 2 pin.

P6, P9: gniazdo do złącza na taśmę 2x7 pin. kątowe  
 P7, P8: gniazdo do złącza na taśmę 2x7 pin. proste  
 JP2: listwa goldpin 6 pin.  
 JP3: listwa goldpin 2 pin.  
 S2, S3, S4, S5, S6: przyciski chwilowe  
 T1: transformator sieciowy TSPZ 6/0002M (INDEL)

##### tylko moduł MASTER

##### Rezystory

R2, R4: 750  $\Omega$  (1206)  
 R21, R22, R24: 220  $\Omega$  (1206)

##### Kondensatory

C1: 33 pF trymer  
 C2: 0,1 F (kondensator podtrzymujący)  
 C10: 100 nF (1206)

##### Półprzewodniki

U1: PCF8583P (DIP-8)  
 U7: MAX1232 lub odpowiednik np. TC1232 (DIP-8)  
 D1: 1N4148  
 DS1: LED zielony 5 mm  
 DS2: LED SMD zielony (1206)  
 DS3: LED SMD czerwony (1206)

##### Inne

ETH1: moduł EM100 (Tibbo)  
 Y1: kwarc 32,768 kHz  
 J2: gniazdo RJ45

Ważnym aspektem przy opracowywaniu toru wejściowego jest jego zabezpieczenie przed sytuacjami udarowymi, takimi jak impulsy wysokiego napięcia, wynikające z przenikania zakłóceń pochłanianych przez kable o znacznych długościach (sytuacja często występująca w obiektach przemysłowych) oraz przypadkowe zwarcie przewodów pętli prądowej. Tor wejściowy opisywanego systemu został zabezpieczony przed wymienionymi wyżej udarami. Przed szpilkami wysokiego napięcia przetwornik analogowo-cyfrowy jest chroniony przez dwukierunkowe diody transil. Zabezpieczają one wejścia przetwornika przed napięciem przekraczającym ok. 6 V, a w impulsie o czasie trwania 1 ms mogą przyjąć moc 600 W. Zabezpieczeniem przed zwarciami jest bezpiecznik polimerowy o prądzie 100 mA (zabezpieczający wszystkie 4 kanały wejściowe pętli prądowych). Przed zakłóceniami impulsowymi nałożonymi na sygnał prądowy chroni filtr dolnoprzepustowy z kondensatorem tantalowym o stosunkowo dużej pojemności 47  $\mu$ F. Przyjęto, że sygnał z przetwornika pomiarowego jest wolnozmienny – stałą czasową filtru ustalono na ok. 5 ms.

**Wejścia stykowe (kanały B1 i B2).** Oprócz czterech wejściowych kanałów przystosowanych do podłączenia pętli prądowych, moduł wzbogacono o dwa dodatkowe wejścia dwustanowe. Można do nich podłączyć czujniki stykowe. Ponieważ zdecydowano się na bezpośrednie dołączenie wejść kanałów do mikrokontrolera, pętla prądowa ma obniżone napięcie zasilające

(+5 V) i zmniejszony prąd zwarcia (do ok. 4 mA). Mikrokontroler został zabezpieczony przez szpilki wysokiego napięcia diodami przeciwprzepięciowymi, podobnie jak to było w przypadku kanałów A1 do A4. Chociaż stan zwarcia/rozwarcia mógłby być z powodzeniem wykrywany przez zmianę stanu zwykłego cyfrowego wejścia portu mikrokontrolera, autor zdecydował żeby wykorzystać wewnętrzny przetwornik A/C układu, a tym samym zwiększyć uniwersalność wejść stykowych. Pozwala to znacznie lepiej rozpoznawać stan linii, np. wykrywać nietypową rezystancję zwarcia, uszkodzenie, itp.

**Elementy wyjściowe.** Urządzenie tu omawiane jest systemem pomiarowym. W zasadzie nie było żadnej potrzeby, aby dodawać do niego elementy pełniące funkcję wyjść. Jednak względy praktyczne przeważały i zdecydowano się uzupełnić moduł o dwa niezależnie sterowane wyjścia przekaźnikowe. Dodatkowo, po napisaniu odpowiedniej aplikacji, można w bardzo prosty sposób zbudować przy użyciu modułów tu opisywanych nieskomplikowany układ regulacyjny. Aplikacja wtedy przejmuje funkcję regulatora – analizuje stan wejść i odpowiednio steruje wyjściami przekaźnikowymi. Część elektryczna systemu pomiarowego nie wymaga żadnych modyfikacji, a korzystając z zawartej w poniższym opisie dokumentacji można napisać dowolny program sterujący.







Wyjścia oznaczone C1 i C2 nie są identyczne. Wyjście C1 jest sterowane przekaźnikiem elektromechanicznym, pozwalającym na podłączenie obciążenia 230 V/5 A.

Wyjście C2 jest sterowane przekaźnikiem półprzewodnikowym typu triakowego, które przy takim samym napięciu nominalnym 230 V, posiada mniejszą obciążalność prądową: 2 A. Każde z tych rozwiązań ma swoje wady i zalety o czym później.

**Przetwornik analogowo-cyfrowy.** W każdym systemie pomiarowym ważnym elementem jest przetwornik analogowo-cyfrowy. W omawianym projekcie jest to tym ważniejsze, iż każdy moduł ma niezależne cztery wejścia analogowe. Oczywiście nie ma tutaj potrzeby, aby konwersja na postać cyfrową była bardzo szybka, tym bardziej, że wcześniej założono wolne zmiany sygnałów wejściowych. Nie ma też potrzeby, aby stosować dedykowane przetworniki dla każdego z kanałów.

Rozdzielczość zastosowanego przetwornika powinna wystarczać do osiągnięcia założonej dokładności rzędu 0,1% dla pomiaru przy pełnej skali. W projekcie zastosowano źródło napięcia odniesienia o wartości 4,096 V. Aby dokładność nie była niższa od założonej, rozdzielczość przetwornika A/C nie może być niższa niż 11 bitów. Spośród co najmniej kilku dostępnych na rynku przetworników A/C autor wybrał układ firmy MAXIM – MAX1247, będący 4-kanałowym 12-bitowym przetwornikiem o sukcesywnej aproksymacji. Nowoczesny, wielokanałowy układ eliminuje konieczność stosowania zewnętrznego multipleksera. Szczegóły obsługi użytego przetwornika można znaleźć w dokumentacji firmy Maxim.

**Andrzej Piernikarczyk**  
andrzej.piernikarczyk@gmail.com

R	E	K	L	A	M	A
 <p><b>Renomowany producent drukarek INK-JET oferuje wysokiej klasy</b></p> <p><b>Aktywny detektor podczerwieni do zastosowań w układach automatyki i zabezpieczeń</b></p> <ul style="list-style-type: none"> <li>małe wymiary budowy (M18x1)</li> <li>duża odporność na zakłócenia</li> <li>wbudowany wskaźnik zadziałania</li> <li>wyjście odporne na zwarcie</li> <li>wykonania PNP, NPN</li> </ul> <p><b>EBS Ink-Jet Systems Poland Sp. z o.o.</b> ul. Tarnogajska 13, 50-512 Wrocław tel. (071) 367 04 11, fax (071) 373 32 69</p>				<ul style="list-style-type: none"> <li>montaż powierzchniowy SMD</li> <li>programowanie, testy ICT i funkcjonalne</li> <li>kompleksowe przygotowanie dokumentacji</li> <li>montaż przewlekany THT uzupełniający</li> <li>projektowanie obwodów drukowanych</li> <li>zakup i kompletacja podzespołów</li> </ul>  <p><b>ente</b></p> <p>ul. Gaudiego 7 44-100 Gliwice</p> <p>tel: +48 32 33 82 200 fax: +48 32 33 82 210</p> <p>e-mail: produkcja@ente.com.pl http://www.ente.com.pl</p> <p>   </p>		