

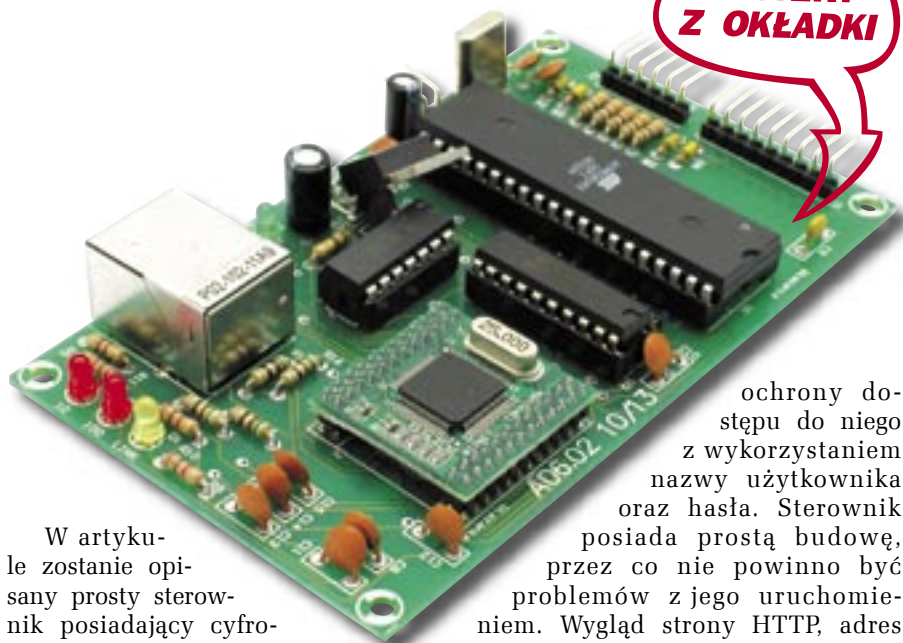
Sterownik z interfejsem TCP/IP

AVT-974

**PROJEKT
Z OKŁADKI**

Powszechnie dostępne specjalizowane układy interfejsów sieciowych stały się powodem lawinowo powstających projektów najróżniejszych sterowników ethernetowych. Wpływa na to również łatwość ich stosowania, a możliwość zbudowania internetowej aplikacji, która jeszcze nie dawno była tylko marzeniem, dziś jest już „chlebem powszednim”.

Rekomendacje: głównym zastosowaniem sterownika będzie zbieranie danych z różnorodnych czujników wejściowych i przekazywanie ich do komputerów pracujących w lokalnej sieci ethernetowej lub komunikujących się przez Internet.



W artykule zostanie opisany prosty sterownik posiadający cyfrowe wejścia i wyjścia, wyposażony w interfejs sieciowy. Linie wejściowe sterownika mogą być wykorzystywane do odczytu stanu wszelkiego rodzaju czujników oraz sygnałów pochodzących np. z innych urządzeń. Linie wyjściowe mogą służyć do sterowania różnymi urządzeniami wykonawczymi. W przypadku, gdy potrzebne jest sterowanie urządzeniami zasilanymi z sieci 230 V, należy dodatkowo zastosować przekaźnik lub triak. Stan linii wyjściowych sterownika jest zapamiętywany w nieulotnej pamięci, więc nie jest tracony po zaniku napięcia zasilającego. Komunikacja ze sterownikiem jest bardzo prosta i odbywa się poprzez stronę internetową z wykorzystaniem protokołu TCP/IP. Jednym z najbardziej popularnych układów z zaimplementowanymi stosami TCP/IP jest układ W3100A firmy Wiznet. Oferuje on 4 gniazda, co oznacza, że do układu można dołączyć jednocześnie czterech klientów. Ma także wbudowaną pamięć RAM będącą buforem dla otrzymywanych i wysyłanych danych. Układ W3100A jest dostępny w obudowie SMD, ale jest również odpowiedni moduł IIM7000A, który bez większych problemów można zastosować we własnych aplikacjach. Tego typu moduł wykorzystano w sterowniku. Niewątpliwą zaletą urządzenia jest możliwość

ochrony dostępu do niego z wykorzystaniem nazwy użytkownika oraz hasła. Sterownik posiada prostą budowę, przez co nie powinno być problemów z jego uruchomieniem. Wygląd strony HTTP, adres IP, maskę oraz bramę można łatwo modyfikować, wystarczy do tego celu podstawowa znajomość języka HTML oraz oprogramowania Bascom AVR, w którym napisano oprogramowanie sterownika. W artykule nie będą przedstawiane informacje o działaniu protokołów TCP/IP oraz HTTP. Dodatkowych informacji na temat tych protokołów należy szukać w dostępnej literaturze.

Opis działania układu

Na rys. 1 przedstawiono schemat ideowy sterownika TCP/IP. Zastosowano w nim mikrokontroler AVR typu ATmega162, gdyż posiada on magistralę przystosowaną do komunikacji z zewnętrzną pamięcią RAM (umożliwia jej zaadresowanie). Za jej pomocą odbywa się komunikacja z modułem sieciowym U2. Układ U3 służy do zatraskiwania mniej znaczącego bajtu adresowego (linie A0...A7) wystawianego na magistralę danych. Moduł U2 zajmuje adres z zakresu &H8000 do &HFFFF. Większość tego obszaru jest używana przez wewnętrzne bufony, a pozostała część przez rejestry statusowe i kontrolne modułu U2. Linia A15 zanegowana przez U4D steruje linią wyboru CS modułu U2. Dioda D1 sygnalizuje nawiązane połączenie z innym urządzeniem sieciowym, natomiast diody D2 i D3 sygnalizują prędkość transmisji (od-



PODSTAWOWE PARAMETRY

- Płytką o wymiarach 109x64 mm
- Liczba wejściowych linii cyfrowych: 5
- Liczba cyfrowych linii wyjściowych: 8
- Zapamiętywanie stanu linii w pamięci nieulotnej
- Ochrony dostępu do sterownika: nazwa użytkownika oraz hasło
- Obsługa sterownika przez stronę HTTP

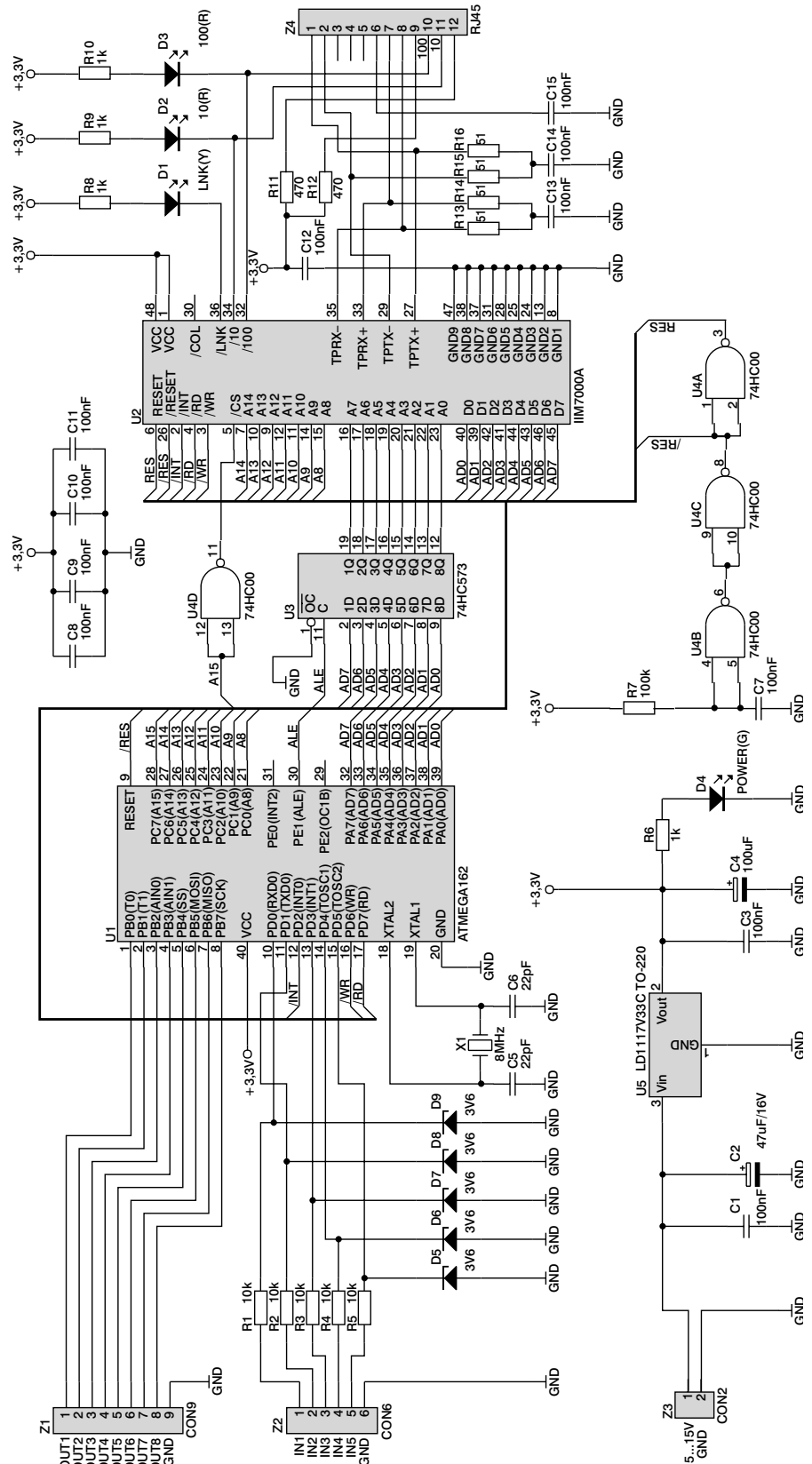
powiednio 10 Mb/s i 100 Mb/s). Do linii TPR i TPT modułu U2 zostało dołączone gniazdo RJ45 (Z4) z wbudowanym już transformatorkiem. Moduł U2 wymaga dwóch sygnałów zerowania o przeciwnych fazach. Obwód zerowania został zbudowany w oparciu o elementy R7, C7 oraz bramki U4A, U4B i U4C. Moduł U2, jak i mikrokontroler U1 zerowany jest zawsze po włączeniu zasilania. Czas zerowania jest zależny od wartości elementów R7 i C7. Mikrokontroler jest taktowany zewnętrznym rezonatorem kwarcowym o częstotliwości 8 MHz. Moduł U2 wymaga jednej linii przerwania i do tego celu zostało wykorzystane przerwanie INT0. Linie wejściowe sterownika zostały zabezpieczone przed wyższymi napięciami od napięcia zasilania mikrokontrolera. Do tego celu wykorzystano elementy R1...R5 oraz diody Zenera D5...D9. Dzięki tym elementom możliwe jest podawanie napięć na wejścia sterownika wyższych od napięcia zasilania mikrokontrolera. Wszystkie linie portu PB zostały skonfigurowane jako wyjściowe. Elementy sterownika są zasilane napięciem +3,3 V uzyskiwanym na wyjściu stabilizatora U5. Napięcie +3,3 V zasilające elementy sterownika wynika z tego, że moduł U2 nie może być zasilany wyższym napięciem. Dioda D4 sygnalizuje napięcie zasilania sterownika.

Montaż i uruchomienie

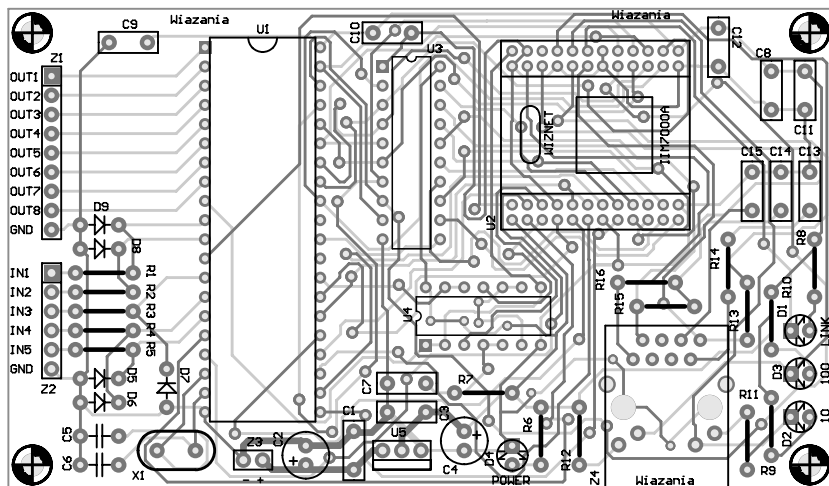
Schemat montażowy sterownika pokazano na rys. 2. W sterowniku nie wykorzystano żadnych elementów SMD. Montaż należy rozpocząć od elementów najmniejszych. Moduł sieciowy U2 można włutować, ale można również zastosować dla niego odpowiednie gniazdo o rastrze 2 mm. Po zmontowaniu całości wraz z zaprogramowanym mikrokontrolerem, sterownik jest gotowy do uruchomienia. Można go zasilć napięciem stałym z zakresu od 5 V do 15 V. Sterownik można dołączyć do routera lub bezpośrednio do komputera za pomocą skrętki. W drugim przypadku potrzebny będzie przewód skrosowany. Sterownik posiada domyślny numer IP: 192.168.1.5, maskę 255.255.255.0 oraz bramę 192.168.1.1. Po włączeniu zasilania powinna zaświecić się dioda D2 lub D3 w zależności od możliwej prędkości transmisji. Poprawność konfigu-

racji i komunikację można sprawdzić programem *ping*, który jest dostępny w prawie każdym systemie operacyjnym. Program *ping* wysyła pakiet danych pod określony adres IP. Ad-

resowane urządzenie powinno zwrócić otrzymane dane. Aby sprawdzić poprawność komunikacji ze sterownikiem, należy wysłać następujące polecenie:



Rys. 1. Schemat elektryczny sterownika



Rys. 2. Schemat montażowy płytki sterownika

ping.exe 192.168.1.5

W przypadku braku odpowiedzi od sterownika, problem może być ze źle skonfigurowaną kartą sieciową w komputerze, w złym przewodzie połączeniowym lub z samym układem wykorzystującym moduł sieciowy. Należy wtedy w pierwszej kolejności sprawdzić poprawność montażu oraz napięcia zasilania. Ze sterownikiem można się połączyć poprzez przeglądarkę stron HTTP. Należy w niej wpisać adres `http://192.168.1.5/index.htm`.

Na rys. 3 pokazano wygląd wyświetlanej strony HTTP sterownika. Zawarto tu informacje o stanie linii wejściowych i wyjściowych sterownika. Za pomocą przycisków ON i OFF można zmieniać stan danej linii wyjściowej. Wyświetlana strona jest odświeżana automatycznie co kilka sekund. Jeśli włączona została autoryzacja, przed załadowaniem strony HTTP zostanie wyświetlone okienko (rys. 4), w którym należy podać nazwę użytkownika oraz hasło. Domyśl-

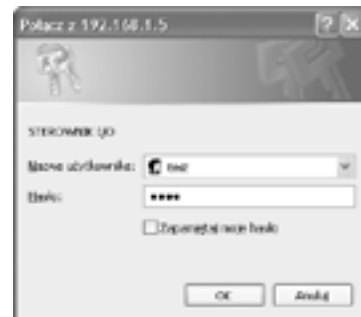


Rys. 3. Strona WWW zaimplementowana w sterowniku

na nazwa użytkownika, jak i hasło do „test”. Włączenie autoryzacji umożliwia ochronę dostępu do sterownika przed niepowołanymi osobami. Gdyby sterownik miał sterować urządzeniami zasilanymi z sieci 230 V, należy do wyjść sterownika dołączyć układ wykonawczy, którym może być przekaźnik lub triak. Na rys. 5 pokazano przykładowy układ wykonawczy z przekaźnikiem, a na rys. 6 z wykorzystaniem triaka. Oprogramowanie sterownika można dowolnie modyfikować. Można w nim zmienić nie tylko numer IP, maskę, bramę, nazwę użytkownika, hasło ale również wygląd strony HTTP. W EP prezentowany był kurs, w którym przedstawiono przykłady programów współpracujących z modułem sieciowym firmy Wiznet. Informacje tam zawarte będą pomocne w ewentualnych przeróbkach programu tak, aby na przykład obsłużyć wiadomości email. Pomocne przeróbkom będzie również krótki opis działania programu sterownika.

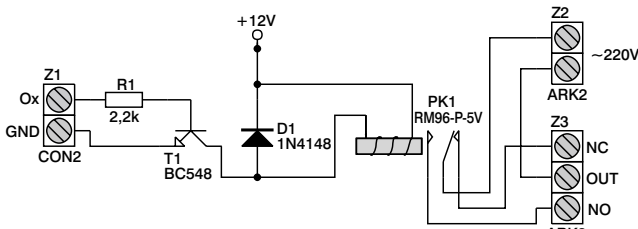
Oprogramowanie sterownika

Oprogramowanie sterujące sterownikiem, zostało napisane w Bascom AVR i realizuje obsługę protokołu HTTP, który jest wykorzystywany do przeglądania stron www. Jest on oparty na mechanizmie pytanie-odpowiedź. Klient żąda przesłania strony z serwera HTTP, odpowiedzią jest prze-

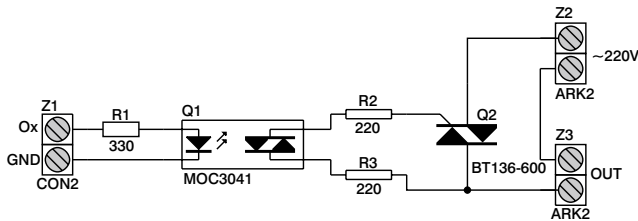


Rys. 4. W przypadku aktywnej autoryzacji dostęp do strony jest możliwy po zalogowaniu

ślanie strony do klienta. Protokół HTTP wykorzystuje port o numerze 80. Po udzieleniu odpowiedzi na zapytanie klienta, połączenie jest zrywane do czasu otrzymania następnego zapytania. W zapytaniach wykorzystuje się kilka metod, takich jak: *GET*, *HEAD*, *POST*, *PUT* itp. Metoda *GET* jest stosowana w zapytaniu o żądanie przesłania strony HTTP o określonym adresie. Metoda *POST* (wykorzystywana podczas przesyłania danych z formularzy) pozwala żądać od serwera przyjęcia załączonych danych jako nowej, skierowanej do niego publikacji. W programie sprawdzany jest rodzaj zapytanie, na które jest przesyłana odpowiedź. Zmianę numeru IP, maski i bramy można zrealizować tylko w kodzie programu. Po każdej zmianie program należy poddać kompilacji. Otrzymaoną zawartość pliku należy zapisać do mikrokontrolera. Numer IP, maskę i bramę można zmienić w instrukcji *config TCP/IP*. Można również w niej zmienić adres MAC sterownika. W programie występuje stała *Authenticate*, od wartości której zależy czy dostęp do strony HTTP sterownika będzie chroniony. Nadanie stałej *Authenticate* wartości 1 spowoduje włączenie chronionego dostępu do sterownika. Nazwę użytkownika, jak i login można zmodyfikować w linii *If Base64dec(s) = "test:test" Then...*. Nazwa użytkownika i hasło są oddzielone znakiem dwukropka. W tym przypadku nazwa użytkownika i hasło składają się ze znaków *test*. Strona HTTP została zapisana w pamięci programu za pomocą instrukcji *data*. Jest ona identyfikowana etykietą *www*. Do odczytania tych danych wykorzystywane są instrukcje *restore* oraz *read*. Po wysłaniu do przeglądarki ilości przesyłanych danych,



Rys. 5. Wyjściowy obwód wykonawczy z przekaźnikiem



Rys. 6. Wyjściowy obwód wykonawczy z triakiem

WYKAZ ELEMENTÓW

Rezystory

R1...R5: 10 kΩ
 R6, R8...R10: 1 kΩ
 R7: 100 kΩ
 R11, R12: 470 Ω
 R13...R16: 51 Ω

Kondensatory

C1, C3, C7...C15: 100 nF
 C2: 47 μF/16 V
 C4: 100 μF/16 V
 C5, C6: 22 pF

Półprzewodniki

U1: ATmega162 DIP-40
 U2: moduł IIM7000A
 U3: 74HC573
 U4: 74HC00
 U5: LD1117V33C TO-220
 X1: kwarc 8 MHz
 D1: LED 3mm żółta
 D2, D3: LED 3mm czerwona
 D4: LED 3mm zielona
 D5...D9: dioda Zenera 3,6 V

Inne

Z1: Goldpin 1x9
 Z2: Goldpin 1x6
 Z3: Goldpin 1x2
 Z4: złącze RJ45 z trafo typu J0026D21B

następuje wysłanie strony HTTP odczytywanej kolejno z pamięci programu. W kod wyświetlanej strony HTTP, podczas jej wysyłania, wpłata jest informacja o stanach linii wejściowych i wyjściowych. Miejsca wstawiania informacji o wejściach i wyjściach są rozpoznawane poprzez zliczanie liczby linii *data*. Na przykład w przypadku wysyłania linii

o numerze 67 program realizuje następujące instrukcje:

```
Elseif Count = 67 Then
  'jeśli count=67 to
    If Out4 = 1 Then
  'jeśli Out4 = 1 to
    S = "ON"
  'zapis do S tekstu ON
    Else
  'w przeciwnym razie
    S = "OFF"
  'zapis do S tekstu OFF
    End If
```

W zależności od stanu linii Out4 zostanie wysłany tekst *ON* lub *OFF*. W przypadku otrzymania strony HTTP o nazwie innej niż */index.htm*, wysłana jest strona HTTP, która powoduje przekierowanie strony na adres */index.htm*. Instrukcja HTML:

```
Data "<meta HTTP-EQUIV={034}refresh{034} CONTENT={034}5{034}>"
```

w kodzie powoduje, że strona HTTP będzie automatycznie odświeżana co 5 sekund. Znak o kodzie ASCII {034} to cudzysłów ("). Jak widać odczyt i wyświetlenie na stronie HTTP informacji o wyjściach i wejściach sterownika jest bardzo proste. Trudniejsza do zrealizowania jest możliwość sterowania wyjściami za pomocą przycisków umieszczonych na stronie HTTP. Do tego celu wykorzystano przyciski wysłania formularza przy pomocy metody *POST*. Przykładowo

instrukcja zapisane w HTML realizująca wysłanie formularza z komunikatem ustawienia wyjścia Out1 jest następująca:

```
Data "<form method={034}post{034}>"
Data "<input type={034}hidden{034} name={034}on1{034}>"
Data "<input type={034}submit{034} value={034}ON {034}></form></td>"
```

Dla pozostałych przycisków ustawiających i zerujących linie Out, instrukcje będą podobne. Naciśnięcie przycisku spowoduje wysłanie do serwera zapytania z wykorzystaniem metody *POST*. Po wykryciu przez serwer tej metody, wykonywany jest podprogram *page* oraz zmiennej *Bcmd* zostaje przypisana wartość 2. W przypadku metody *POST* wykrywany jest komunikat *Content-Length*, z którym związana jest ilość wysłanych przez przeglądarkę danych. Ilość wysłanych danych z przeglądarki jest zapisywana do zmiennej *Bcontent*. Przy metodzie *POST* jeśli zmienna *Bcmd* wynosi 2, następuje odczyt danych wysłanych po naciśnięciu przycisku wysyłającego formularz. Wysłane dane są odczytywane do zmiennej *Buf*. Ilość odbieranych danych wskazuje zmienna *Bcontent*. Warunek ustawienia wyjścia Out1 jest następujący:

```
If Left(s , 3) = "on1" Then 'jeśli otrzymano znaki on1 to
  Set Out1
'ustawienie linii Out1
```

Jeśli otrzymano znaki *on1* wyjście Out1 zostanie ustawione. Warunki ustawiania i zerowania dla pozostałych wyjść są podobne.

Jak pokazano umieszczenie na stronie stanów linii wejściowych i wyjściowych nie jest trudne. Stosunkowo łatwo można zmodyfikować program tak, aby w przypadku jakiegoś wykrytego przez wejścia zdarzenia była wysyłana wiadomości email. Zastosowań przedstawionego sterownika może być bardzo wiele. W dużej mierze jego funkcje będą zależały od oprogramowania.

Marcin Wiązania, EP
marcin.wiazania@ep.com.pl

forum.ep.com.pl

forum.ep.com.pl

forum.ep.com.pl

forum.ep.com.pl

forum.ep.com.pl

forum.ep.com.pl