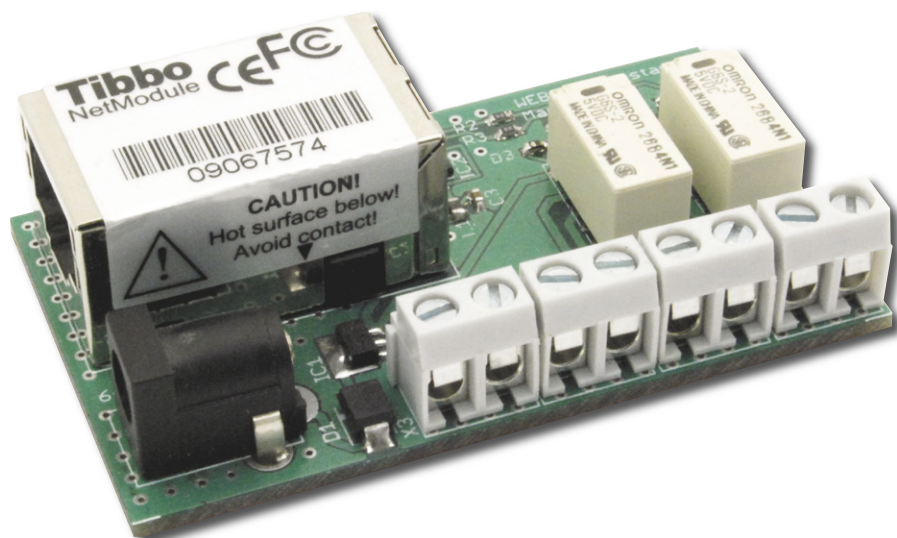




Przełącznik internetowy

W artykule przedstawiamy unowocześnioną wersję LAN-owego restartera, który opisywaliśmy w EP1/2007. Jest to urządzenie, które może być przydatne do zdalnego zarządzania routerami i switchami w małych (na przykład domowych lub osiedlowych) sieciach LAN.

Rekomendacje: przełącznik internetowy polecamy osobom administrującym małymi sieciami ethernetowymi, pozwala on zdalnie (poprzez przeglądarkę WWW) przywrócić prawidłową pracę routera lub switcha w sytuacji, gdy uległ on zawieszeniu.



Początkowo przełącznik internetowy miał stanowić unowocześnioną wersję LAN-owego restartera opisanego w EP 1/2007. W pierwotnym wzorze wykorzystywano linię telefoniczną do chwilowego odcięcia zasilania urządzeń odpowiedzialnych za rozdzielanie sygnałów w sieci LAN (routery, switchy) w przypadku, gdy urządzenie takie przestanie działać poprawnie (po prostu zawiesi się). Zaletą, a jednocześnie wadą układu było to, iż był on sterowany niezależnym od sieci LAN kanałem – linią telefoniczną. Linia taka musiała być dostępna w sąsiedztwie infrastruktury LAN.

Opisany niżej przełącznik z założenia miał posiadać takie same funkcje, jednak jego kanałem kontrolnym miała być sama sieć LAN. Autor miał na myśli rozbudowane sieci LAN, łącznie z elementami sieci bezprzewodowych. Urządzenie sterowane z poziomu przeglądarki WWW potrafiłoby odłączyć zasilanie np. Access Pointa,

który przestał poprawnie pracować. Warunkiem byłoby jednak to, iż po stronie kablowej wszystkie urządzenia (np. switchy), które umożliwiają komunikację z samym restartem musiałyby działać bez zarzutu. Na rys. 1 przedstawiono przykładową sytuację (autentyczną), w której opisywany układ jest przydatny z punktu widzenia użytkownika sieci. W budynku wielokondygnacyjnym do istniejącej infrastruktury LAN dołączono Access Point łączący budynek z dostawcą Internetu. Urządzenie było jednak zawodne. Co jakiś czas następowała przerwa w dostawie Internetu. Jediną metodą naprawy sytuacji było ręczne wyzerowanie urządzenia przy pomocy lokalnego przycisku, bądź proste odłączenie zasilania. Czynność ta była jednak uciążliwa z punktu widzenia użytkowników pracujących na poszczególnych kondygnacjach. Szybkim rozwiązaniem problemu było zastosowanie przełącznika internetowego w obwodzie zasilania Access Pointa. Użytkownik w chwili dostrzeżenia problemu z dostawą Internetu mógł zdalnie odciąć, a następnie dołączyć zasilanie AP, powodując jego restart.

Podczas tworzenia schematu urządzenia autor doszedł do wniosku ze spektrum zastosowań może być dużo większe. Ponieważ układ jest w stanie sterować (z poziomu interfejsu WWW) niezależnie dwoma przełącznikami, to dlaczego nie używać go do ogólnych zastosowań automatyki domowej, czy nawet przemysłowej. Z myślą o tym, w dalszej części artykułu zostanie zaprezentowana prosta aplikacja (napisana w LabVIEW) sterująca układem.

Podsumowując: układ przełącznika internetowego jest urządzeniem sterującym w sposób niezależny dwoma przełącznikami. Włączanie i wyłączanie przełączników może odbywać się z poziomu interfejsu WWW urządzenia (przy użyciu przeglądarki), jak również przy pomocy

AVT-5157

W ofercie AVT:
AVT-5157A – płytką drukowaną
AVT-5157B – płytką + elementy

PODSTAWOWE PARAMETRY

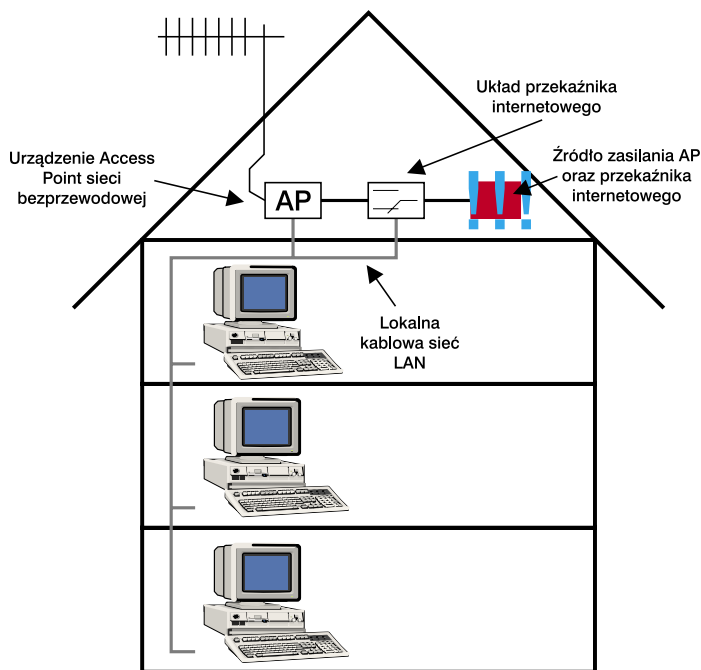
- Płytką o wymiarach 76x50 mm
- Napięcie zasilania: 6...9 VDC
- Pobór prądu przy załączonych obu przełącznikach: ok. 350 mA
- Niezależne sterowanie dwoma przełącznikami poprzez sieć Ethernet
- Sterowanie z poziomu interfejsu WWW, jak również z poziomu niezależnej aplikacji (LV)



PROJEKTY POKREWNE

wymienione artykuły są w całości dostępne na CD

Tytuł artykułu	Nr EP/EdW	Kit
Karta przełączników sterowana przez Internet	EP 2/2007	AVT-966
Uniwersalny interfejs ethernetowy	EP 1/2007	AVT-1443
Internetowy sterownik urządzeń	EdW 3/2008	AVT-2859
Internetowy interfejs dla mikrokontrolera	EP 3-5/2002	AVT-5055
Uniwersalny interfejs internetowy	EP 4-5/2006	AVT-927
Karta wejść z interfejsem Ethernet	EP 10/2006	AVT-953



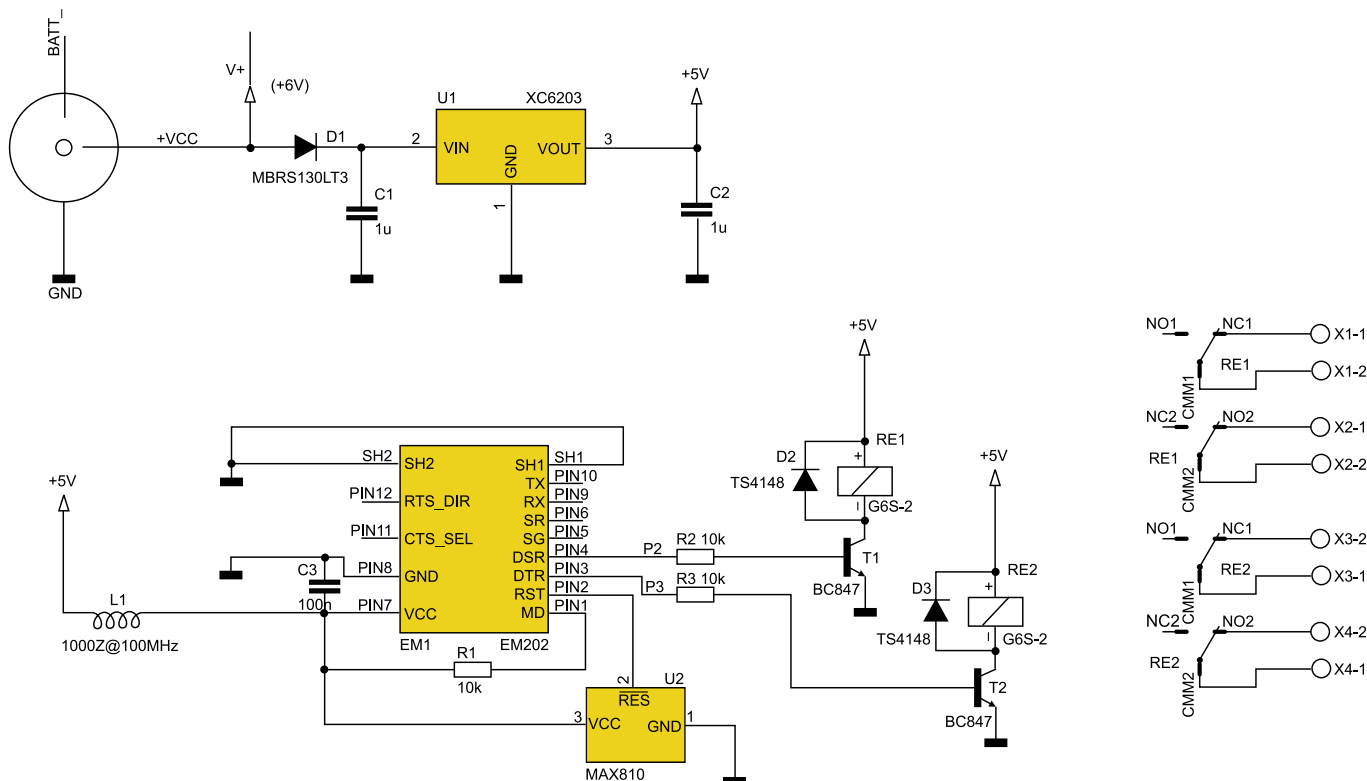
Rys. 1. Budynek z infrastrukturą LAN

samodzielnej aplikacji. Zasilany może być z dowolnego niestabilizowanego źródła dostarczającego napięcie stałe z zakresu 6...9 V. Pobór prądu układu wynosi max 300 mA.

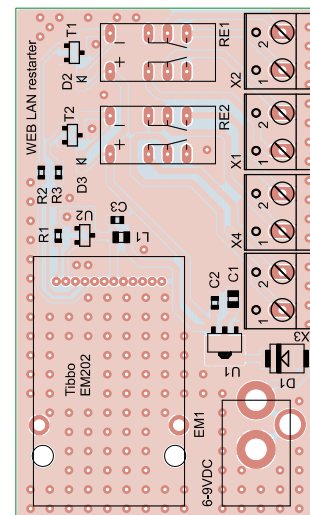
Budowa

Schemat ideowy przełącznika internetowego przedstawiono na rys. 2. Głównym elementem jest moduł ethernetowy firmy Tibbo – EM202. Do poprawnej pracy wymaga on zewnętrznego układu zerowania. Funkcję tę realizuje układ MAX810 (U2). W obwodzie zasilania modułu

zastosowano filtr składający się z dławika L1 i kondensatora C3. Jako elementy wykonawcze zastosowano miniaturowe przełączniki firmy Omron G6S-2 (RE1, RE2), a do ich sterowania użyto kluczy zbudowanych z tranzystorów T1 i T2. Tranzystory są zabezpieczone przed przepięciami wywołanymi pracą przełączników (diody D2, D3). Bazy tranzystorów są podłączone (poprzez rezystor ograniczający prąd) do linii GPIO modułu. Linie te to (za dokumentacją Tibbo): P3 (pin 3) oraz P2 (pin 4). Na schemacie ideowym linie sterujące przełącznikami są opisane jako



Rys. 2. Schemat ideowy przełącznika internetowego

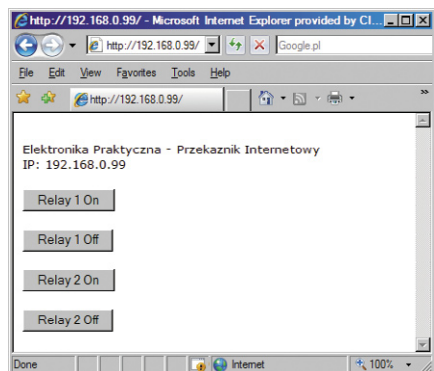


Rys. 3. Schemat montażowy płytki drukowanej przełącznika internetowego

DSR, DTR. Wynika to z faktu, iż w przypadku stosowania modułu pełniącego funkcję mostu Ethernet <-> RS232 linie te są odpowiedzialne za generowanie tych właśnie sygnałów.

Blok zasilania zrealizowano na układzie U1 (XC6203). Jest to regulator LDO o stałym napięciu wyjściowym (+5 V). Jego minimalne napięcie wejściowe jest równe 5,3 V, a wydajność prądowa to 400 mA. Układ potrzebuje do poprawnej pracy jedynie dwóch kondensatorów (C1 oraz C2). W obwodzie zasilania zastosowano też diodę D1 chroniącą układ przed napięciem o odwróconej polaryzacji.

Na rys. 3 przedstawiono schemat montażowy płytki drukowanej układu. Płytką została wyposażona w złącze zasilające DC (2,5 mm)



Rys. 4. Główna strona przełącznika internetowego

oraz w złącza ARK podłączone do wyprowadzeń przełączników. Część ethernetową (wraz z gniazdem RJ45) całkowicie realizuje moduł EM202. Moduł posiada wbudowane dwie diody sygnalizacyjne, które mogą być dowolnie wykorzystywane przez użytkownika. Nie było zatem potrzeby umieszczania dodatkowych, zewnętrznych diod sygnalizacyjnych.

Oprogramowanie

Program przełącznika internetowego został napisany w darmowym środowisku Tide (*Tibbo Integrated Development Environment*) w wersji 2.5.19. Skorzystano z dostarczanego przez środowisko szablonu aplikacji, który implementuje klienta DHCP oraz konfiguruje moduł. Konfiguracja obejmuje włączenie serwera WWW oraz przydzielenie gniazd TCP do jego obsługi. Nazwa szablonu to „DHCP+setup”.

Po włączeniu moduł próbuje odnaleźć w sieci serwer DHCP w celu pobrania konfiguracji IP. Gdy próba ta nie przynosi rezultatu, moduł zostaje skonfigurowany z domyślnym adresem: 192.168.0.99. Adres ten jest zdefiniowany w kodzie szablonu w pliku *main.tbs* (list. 1).

Innym interesującym miejscem w kodzie szablonu jest fragment (również z pliku *main.tbs*) przedstawiony na list. 2. W przeciwieństwie do poprzedniego przykładu, kod ten dotyczy sytuacji, w której komunikacja z serwerem DHCP zakończyła się sukcesem i moduł otrzymał

List. 1. Przydzielenie domyślnego adresu IP w przypadku niepowodzenia pobrania konfiguracji IP z serwera DHCP

```
'Now set the NET parameters
if dhcp_result=OK then
  `DHCP session was successful
  net.ip=dhcp_ip
  net.gatewayip=dhcp_gateway
  net.netmask=dhcp_netmask
else
  `could not complete DHCP- you decide what to do here
  `For example, can set fixed IP OR DO SOMETHING ELSE
  net.ip="192.168.0.99"

`rel test
io.num = PL_IO_NUM_2_DSR
io.state = LOW
io.num = PL_IO_NUM_3_DTR
io.state = LOW

end if
```

List. 2. Fragment kodu szablonu odpowiedzialny za odświeżenie konfiguracji IP z serwera DHCP

```
sub on_sys_timer
`Count down to the time when DHCP lease expires. We only count if DHCP is NOT
at 0
if dhcp_lease_time>0 then
  dhcp_lease_time=dhcp_lease_time-1
  `When DHCP lease expires reboot (and obtain IP again)
  if dhcp_lease_time=0 then sys.reboot
end if

`TO DO: put other stuff that you want to do on timer here

end sub
```

List. 3. Fragment głównego pliku html projektu – *index.html*

```
...
<form action="relay1on.html" method="get">
<input type="submit" value="Relay 1 On">
</form>

<form action="relay1off.html" method="get">
<input type="submit" value="Relay 1 Off">
</form>

<form action="relay2on.html" method="get">
<input type="submit" value="Relay 2 On">
</form>

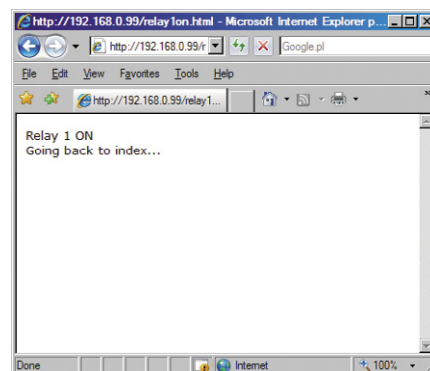
<form action="relay2off.html" method="get">
<input type="submit" value="Relay 2 Off">
</form>
...
```

poprawną konfiguracją IP. Fragment kodu dotyczy parametru *Lease time*, który informuje urządzenie o tym, po jakim czasie powinno ono odświeżyć swoją konfigurację. W przypadku przekroczenia czasu, urządzenie wykonuje funkcję systemową *sys.reboot* powodującą restart urządzenia.

Podstawowym interfejsem użytkownika urządzenia jest strona WWW. Po wpisaniu w oknie przeglądarki adresu IP urządzenia zobaczymy interfejs taki, jak na rys. 4. Mamy do dyspozycji cztery przyciski. Pierwsze dwa są odpowiedzialne za włączanie/wyłączanie przełącznika RE1, pozostałe obsługują przełącznik RE2. Po kliknięciu na wybrany przycisk, w odpowiedzi otrzymamy podstronę potwierdzającą wybraną akcję (rys. 5). Po upływie jednej sekundy zostaniemy z powrotem przekierowani na główną stronę. Dodatkowo fakt wybrania którejś z opcji jest sygnalizowany za pomocą diod LED. Akcja włączenia przełącznika RE1 jest sygnalizowana trzema szybkimi mrugnięciami zielonej diody modułu. Wyłączenie jest sygnalizowane przez pojedyncze mrugnięcie zielonej diody. Za sygnalizację

akcji przełącznika RE2 jest odpowiedzialna dioda czerwona.

Cała funkcjonalność urządzenia została wbudowana w pliki html dołączone do projektu. Opiera się ona o znaczniki *<form>*. Na list. 3. przedstawiono fragment głównego pliku – *index.html*, który jest odpowiedzialny za wyświetlanie czterech przycisków. Kliknięcie na przycisk powoduje wywołanie metody *get* oraz przekazi-



Rys. 5. Podstrona potwierdzająca wykonanie akcji włączenia przełącznika RE1

WYKAZ ELEMENTÓW

Rezystory

R1...R3: 10 kΩ (0603)

Kondensatory

C2: 1 μF/6,3 V (0603)

C1: 1 μF/16 V (0805)

C3: 100 nF/16 V (0603)

Półprzewodniki

D1: MBR5130LT3

D2, D3: TS4148

T1, T2: BC847

U1: XC6203

U2: MAX810

Inne

X1...X4: gniazdo ARK-5.02

L1: dławik 1000Z@100MHz (0805)

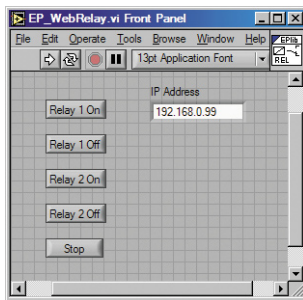
Gn1: gniazdo zasilania 2,5 mm

RE1, RE2: przełącznik G6S

EM1: moduł ethernetowy EM202

Na CD karty katalogowe i noty aplikacyjne elementów oznaczonych na Wykazie Elementów kolorem czerwonym





Rys. 6. Panel frontowy aplikacji sterującej przełącznikiem internetowym

zanie do odpowiedniej podstrony parametrów. Ich wartości są zupełnie nieistotne. Istotne jest to, iż w wyniku tej akcji serwer WWW modułu będzie musiał zwrócić do użytkownika odpowiednią podstronę (*relay1on.html*, *relay1off.html*, *relay2on.htm* lub *relay2off.html*) i co za tym idzie wykonać wbudowany w nią kod. Przyjrzyjmy się obecnie podstronie *relay1on.html*. Na list. 4 przedstawiono kod w Basicu realizujący funkcję włączenia przełącznika RE1 oraz sygnalizację przy pomocy zielonej diody LED. Za manipulację linią DSR modułu (pin 4, do którego podłączony jest tranzystor sterujący przełącznikiem RE1) odpowiada obiekt *io*. W pierwszej kolejności ustawiamy jego własność *num* na odpowiednią wartość tak, aby przy odwołaniu się do własności *io.state* manipulować odpowiednim pinem. W kolejnej linii wywołujemy funkcję systemową *pat.play*, czym sygnalizujemy wykonaną akcję. I to w zasadzie wszystko. Po zwróceniu strony do użytkownika (rys. 5), po 1 sekundzie następuje przekierowanie do strony *index.html*. Odpowiedzialny jest za to kod html (znacznik *<meta>*, zaznaczony na list. 4).

Aplikacja sterująca

Mając wiedzę na temat budowy interfejsu WWW urządzenia możemy przystąpić do utworzenia niezależnej aplikacji sterującej. Taka przykładowa aplikacja została napisana w LabVIEW. Przyjrzyjmy się najpierw panelowi użytkownika (rys. 6). Podobnie jak interfejs WWW, składa się z czterech przycisków odpowiedzialnych za akcje przełączników. Ponadto widoczne jest pole adresu IP urządzenia. Obsługa programu nie wymaga komentarza.

Struktura aplikacji jest oparta o pojedynczą pętlę *while*, w której umieszczono strukturę *Event* odpowiedzialną za przechwytywanie zdarzeń na panelu frontowym. Przyjrzyjmy się akcji przypisanej do przycisku *Relay 1 On* (rys. 7).

List. 4. Podstrona *relay1on.html* odpowiedzialna za realizację akcji włączenia przełącznika RE1

```
<?
  include "global.tbh"
?>

<!doctype html public "-//w3c//dtd wc html//en">
<html>
<body>
<meta http-equiv="Refresh" content="1 url=index.html">
<font size="2" face="Verdana">

  Relay 1 ON

<br>

<?
  `turn on 1st relay
  io.num = PL_IO_NUM_2_DSR
  io.state = HIGH
  `turn on" green led pattern
  pat.play("G-G", PL_PAT_NOINT)
?>

  Going back to index...
</font>
</body>
</html>
```

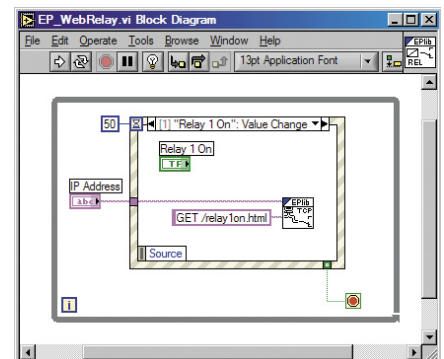
Po zmianie wartości przycisku zostaje wywołany *subvi* przedstawiony na rys. 8. W pierwszej kolejności nawiązujemy próbę połączenia z przełącznikiem internetowym używając adresu IP przekazanego przez kontrolkę *IP Address* (*TCP Open Connection.vi*). Po udanej akcji następuje wywołanie *TCP Write.vi*, który w argumencie przyjmuje referencję do nawiązanego połączenia oraz jako dane do wysłania, zawartość kontrolki *Data*:

```
GET /relay1on.html
```

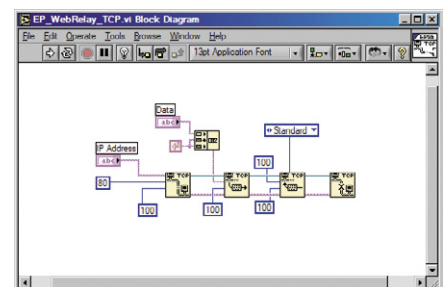
Ponieważ urządzenie eksponuje interfejs http, nasza aplikacja musi zachowywać się tak, jak przeglądarka internetowa w momencie, kiedy naciskamy przycisk *Relay 1 On*. Przeglądarka wysyła żądanie przesłania strony *relay1on.html*, a to, jak już wiemy, wymusi na serwerze WWW wykonanie kodu tej strony i w rezultacie włączenie przełącznika oraz sygnalizację diodą LED. W dalszej części kodu LabVIEW wywołujemy *TCP Read.vi*, który ma za zadanie odebrać stronę WWW zwróconą przez moduł. Odebrane znaki są jednak dla nas zupełnie bezużyteczne, zatem nie wyświetlamy ich na panelu użytkownika. Na sam koniec wywołujemy *TCP Close Connection.vi* w celu zakończenia połączenia z modułem. Pozostałe funkcje (wyłączanie przełączników) są zrealizowane w analogiczny sposób.

Podsumowanie

Prezentowane urządzenie pomimo swej prostoty może być przydatne w systemach automatyki domowej. Fakt możliwości programowania modułu można wykorzystać do utworzenia bardziej zaawansowanej aplikacji. Można np. rozbu-



Rys. 7. Akcja przypisana do przycisku włączania przełącznika RE1



Rys. 8. Kod *subvi EP_WebRelay_TCP.vi* odpowiedzialny za komunikację z modułem

dować urządzenie o autonomiczne sterowanie przełącznikami, np. w celu utworzenia symulatora obecności domowników, bądź sterownika lamp ogrodowych podłączonych do Internetu. Jak zwykle, możliwości są nieograniczone.

Marcin Chruściel, EP
marcin.chrusciel@ep.com.pl

R E K L A M M A

Coraz więcej ciekawych materiałów na
www.ep.com.pl