

# Kurs programowania mikrokontrolerów PIC (10)

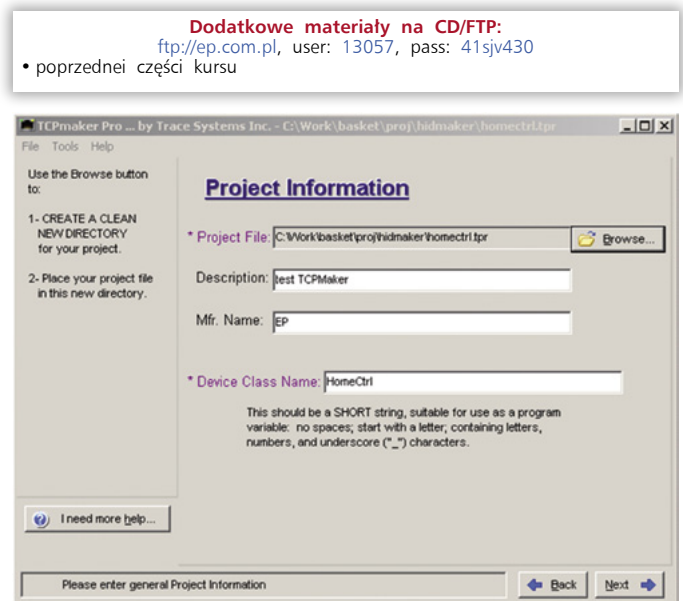


## TCPMaker – Łatwiej już się nie da

*Aplikacje, w których sterownik zbudowany w oparciu o mikrokontroler pracuje z interfejsem Ethernet i stosem TCP/IP są coraz bardziej popularne. Zaletą stosowania takiego rozwiązania jest zestandaryzowany interfejs fizyczny o dużej przepustowości oraz standardowe protokoły komunikacyjne. Pozwala to bez większych problemów dołączyć sterownik do lokalnej sieci Ethernet i dalej, do Internetu. Możliwość sterowania z dowolnego miejsca na Ziemi jest nie do przecenienia, ale równie ważna jest możliwość wykorzystania jako interfejsu sterowania i nadzoru dowolnego urządzenia wyposażonego w przeglądarkę internetową. Wystarczy wyposażyć nasz sterownik w funkcję serwera HTTP i można go kontrolować za pomocą strony internetowej z komputera, laptopa, czy telefonu komórkowego. Gotowe są wszystkie mechanizmy zapewniające identyfikację w sieci i bezbłędne przesyłanie danych. Co ważne, mamy również możliwość łatwego tworzenia interfejsu graficznego dla użytkownika.*

Jednym z bardziej znanych, kompletnych i dystrybuowanych bezpłatnie stosów komunikacyjnych TCP/IP jest produkt firmy Microchip. Można go pobrać ze strony producenta i używać bez ograniczeń, zarówno do zastosowania amatorskiego, jak i profesjonalnego. Oczywiście, z użyciem mikrokontrolerów PIC Micro. Oprócz stosu firma dostarcza dodatkowe narzędzie MPFS2.exe pozwalające na konwersję kodu źródłowego strony napisanej w języku HTML do wartości binarnej wpisywanej do pamięci mikrokontrolera. Wydaje się, że mamy wszystko, co potrzebne aby znając zasady programowania w języku C i w HTML móc zaprojektować prosty serwer HTTP. W rzeczywistości, szczególnie dla początkujących, nie jest to jednak takie łatwe. Ten problem dostrzegła firma Microchip i udostępniła na swoich stronach 3 częściowy kurs *Web-Based Status Monitoring* pokazujący dokładnie, krok po kroku drogę, którą trzeba pokonać, by zaprojektować swoją warstwę aplikacji. Przerobienie tego kursu daje pojęcie o skali problemu i pozwala ukierunkować dalsze poszukiwania niezbędnej wiedzy. W efekcie można projektować przydatne strony, ale jest to proces dość żmudny i wymaga sporo pracy, jeżeli interfejs nie ma sprawiać wrażenia zrobionego ad hoc „na kolanie”.

Ponieważ często przeglądam strony Microchipsa, a szczególnie fragmenty dotyczące rozwiązań sieciowych, to bez trudu zauważyłem reklamowany tam program TCPMaker firmy Trace Systems Inc. Z krótkiego (z oczywistych względów) opisu wynikało, że TCPMaker może za mnie wykonać większą część procesu tworzenia strony przeznaczonej do sterowania moim sterownikiem internetowym i że ta będzie wyglądać atrakcyjnie. Program zainteresował mnie na tyle,



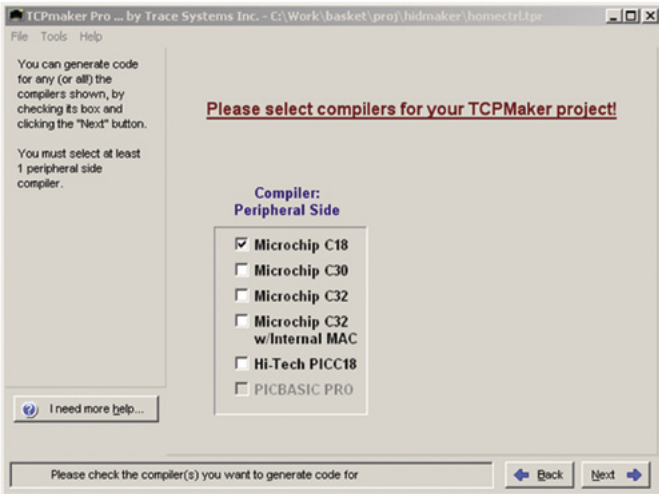
Rysunek 1. Tworzenie projektu

że odwiedziłem stronę producenta [www.tracesystemsinc.com/tcpmaker/tcpmaker.html](http://www.tracesystemsinc.com/tcpmaker/tcpmaker.html). Po lekturze zawartych tam informacji mój apetyt wzrósł na tyle, że postanowiłem protestować jakąś wersję próbną. Niestety okazało się, że producent nie udostępnia takich wersji i tłumaczy to obroną przez piractwem komputerowym. Trudno się temu dziwić zważywszy, że program jest atrakcyjny, a firma nie jest wielka. Rozprzestrzenienie się pirackich kopii mogłoby zapewne zagrozić jej istnieniu. Jednak po krótkiej korespondencji z właścicielem firmy dostałem wyłącznie do celów testowania i opisania wyników testów pełną wersję programu. Wyniki pokazały, że jest to rewelacyjny program dla każdego, kto chciałby się zająć problematyką sterowania z wykorzystaniem Ethernetu i Internetu. Projektanci Trace Systems Inc. zrobili wszystko, a nawet więcej, by proces projektowania skrócić tak bardzo jak to tylko możliwe. Jak tego dokonali postaram się pokazać na przykładzie sterownika nadzoru domu z układem PIC18F67J60.

Założmy, że sterownik będzie serwerem HTTP zdalnie nadzorującym temperaturę wewnętrzną i zewnętrzną w domu, włączającym zasilanie główne oraz oświetlenie na zewnątrz i mającym możliwość zmiany jasności oświetlenia w ogrodzie. Dodatkowo, niech włączać system alarmowy, mierzy napięcie baterii zasilającej ten system i ostrzega o włączeniu się alarmu. Przykładowe funkcje sterownika zostały tak wybrane, by pokazać większość możliwości TCPmaker'a.

### Projekt

Jak w większości tego typu programów typu IDE wszystkie elementy projektowanego serwera są zapisane w pliku projektu. Nowy projekt jest tworzony z menu *File->New* (rysunek 1). Korzystając z tego samego menu można otworzyć wcześniej zapisany projekt (*File->Open Project*). Na stronie startowej



Rysunek 2. Wybór kompilatora

można otworzyć plik pomocy klikając na belkę *I need more help*. Plik pomocy zawiera dosyć dobry opis i warto z niego korzystać.

Po wybraniu opcji tworzenia nowego projektu trzeba kliknąć na przycisk *Next*. Projekt zapisuje się we wcześniej utworzonym katalogu (rysunek 2). W kolejnym kroku trzeba wybrać kompilator. Program jest na tyle uniwersalny, że może generować pliki dla różnych rodzin mikrokontrolerów: PIC18, PIC24/dsPIC30 i PIC32. Co ciekawe, można zaznaczyć jeden, lub kilka kompilatorów. W wypadku zaznaczenia *Microchip C18* i *Microchip C30* zostaną wygenerowane pliki dla PIC18 i PIC24 lub dsPIC30. Ja użyłem kompilatora MPLAB C18 i dlatego zazaczyłem opcję wyboru *Microchip C18* (rysunek 3).

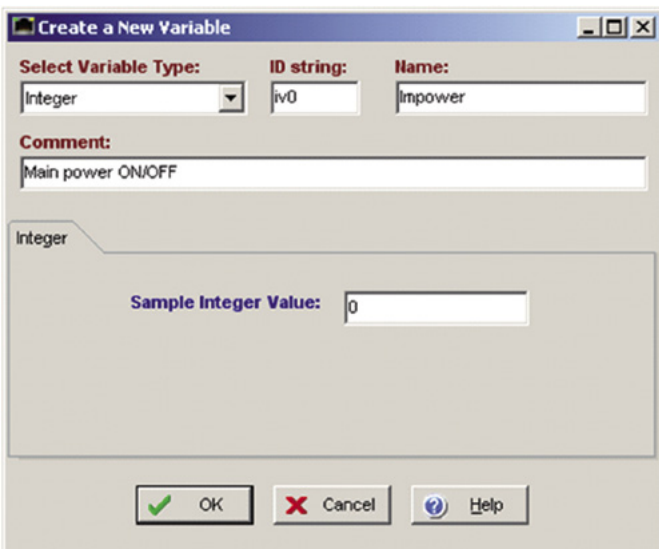
Po zapisaniu nowego projektu i wyborze kompilatora przechodzimy do zasadniczej części projektowania strony serwera http. Na proces projektowania składają się 3 etapy:

- Definicji zmiennych (zakładka *Step1 – Define Your Variables*),
- Zaprojektowania strony (zakładka *Step2 – Lay out Page(s)*),
- Zapisania, testowania i zakończenia pracy (zakładka *Step3 – Save, Test and Exit*).

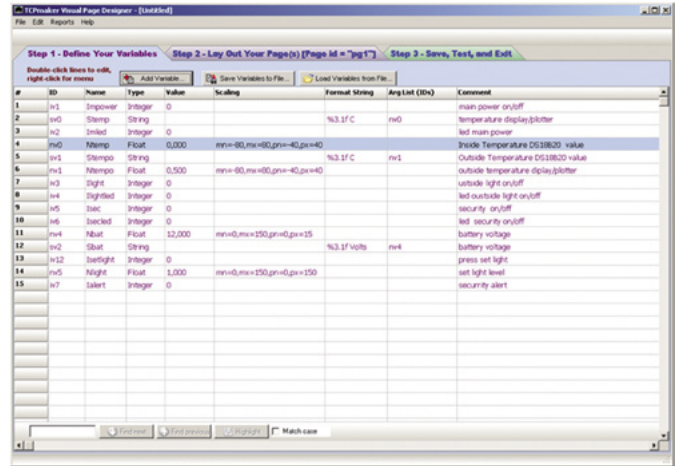
### Zmienne

Przed przejściem do omawiania projektu sterownika trzeba powiedzieć parę słów o zmiennych używanych w projekcie. Zmienne są niezbędne do wymiany danych pomiędzy sterownikiem a przeglądarką. Zależnie od potrzeb definiowane są trzy typy zmiennych:

- **Integer.** Podstawowy typ zmiennej używany do szybkiego przesyłania danych w obu kierunkach.



Rysunek 3. Okno dodawania zmiennej do listy

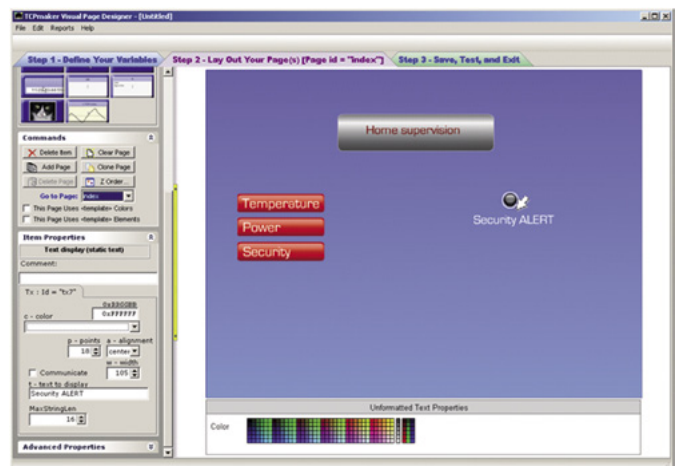


Rysunek 4. Lista zmiennych projektu sterownika

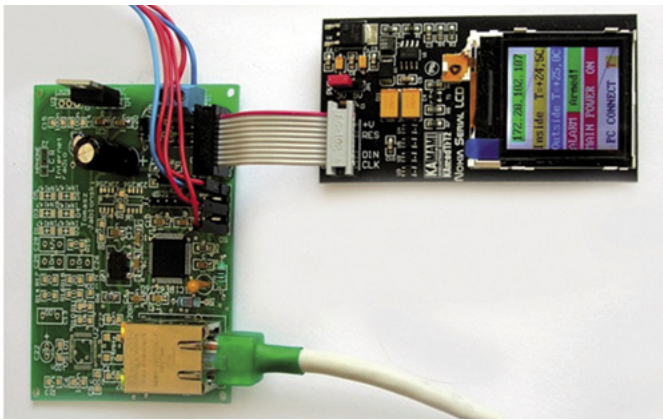
- **Numeric.** Zmienna również przeznaczona do przesyłania danej od sterownika do przeglądarki, ale przeglądarka może ją wyświetlić w postaci zmiennoprzecinkowej. Dodatkowo, wyświetlenie może być skalowane. Do każdej ze zmiennych typu *Numeric* są przypisane definiowane 4 parametry: *mx*, *mn*, *px* i *pn*. Parametry *mx* i *mn* określają maksymalną i minimalną wartość przesyłaną przez sterownik do przeglądarki. Parametry *px* i *pn* określają w jaki sposób te dane mają być przeskalowane i wyświetlanie na ekranie. Dokładnie to opiszę przy okazji wyświetlania danych z termometru DS18B20.

- **String.** Zmienna używana do formatowanego wyświetlania zawartości zmiennej *Numeric*. Do tej zmiennej jest przypisywana w trakcie definiowania zmienna *Numeric*.

Jak już wspomniałem, do definiowania zmiennych jest przeznaczona zakładka *Define Your Variables*. Nowa zmienna jest dodawana do listy po kliknięciu na przycisk *Add Variable* (rysunek 4). W oknie *Select Variable Type* wybiera się typ zmiennej (tutaj *Integer*). W oknie *ID string* jest umieszczona nazwa identyfikatora, a w oknie *Name* nazwa zmiennej. Nazwa jest później używana w trakcie dodawania zmiennej do elementu manipulacyjnego umieszczonego na stronie, a także w deklaracji zmiennych w plikach źródłowych programu sterowania serwerem. Z tych powodów warto nazywać zmienne według wybranej zasady aby potem można było je skojarzyć z wykonywanymi czynnościami. Ja nazwałem swoją zmienną *Impower*, bo jest to zmienna typu *Integer* i przeznaczona do sterowania przyciskiem *Main Power*. Pole komentarza pomaga odnaleźć na liście potrzebną zmienną, jeżeli jest ich więcej. Po kliknięciu na *OK* edytowana zmienna jest dodawana do listy. Można ją potem w dowolnym momencie edytować. Na rysunku 4 pokazano listę zmiennych projektu naszego sterownika.



Rysunek 5. Okno projektowania strony



Fotografia 6. Układ sterownika

## Projekt strony

Po zakończeniu pierwszego kroku, czyli zdefiniowaniu listy zmiennych można przejść do następnego, czyli do projektowania strony serwera. W dowolnym momencie projektowania strony można wrócić do listy zmiennych i zależnie od potrzeb ją modyfikować: edytować istniejące zmienne, usuwać z listy, lub dodawać nowe.

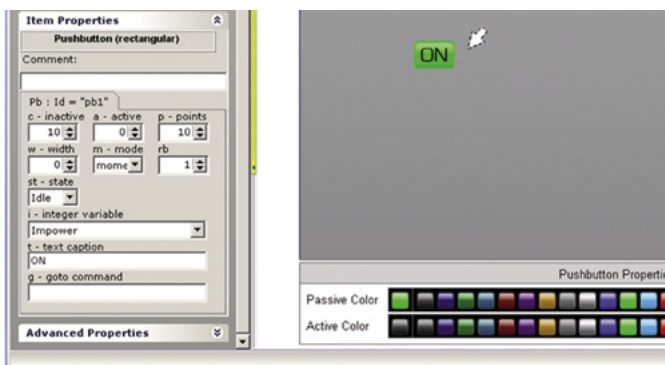
Zakładka projektu strony jest najbardziej rozbudowana (rysunek 5). Okno można podzielić na kilka obszarów:

- **Palette.** Paleta wyboru elementów umieszczanych na stronie.
- **Commands.** Obszar przeznaczony do pracy z podstronami.
- **Item Properties.** Obszar definiowania właściwości edytowanego elementu.
- **Color Properties.** Obszar definiowania kolorów definiowanych elementów (*Text, Pushbutton, Background* itp.).
- **Obszar roboczy** projektu strony. Na nim umieszcza się elementy manipulacyjne (*items*).

Projektowanie polega głównie na wyborze elementów a palety, umieszczeniu ich na obszarze roboczym strony i definiowaniu ich właściwości za pomocą okna *Item Properties*.

## Projekt sterownika

Najlepiej będzie pokazać proces projektowania w programie TCPMaker na konkretnym przykładzie. Do projektu wspomnianego sterownika użyłem mojej płytki z projektu Web radia z układem PIC18F67J60. Ten mikrokontroler ma wbudowany kompletny układ PHY (10 BASE-T), dość dużą pamięć programu (128 kB) i wystarczającą ilość pamięci RAM. Na płytce poza mikrokontrolerem jest umieszczony transformator ze złączem 8P8C (RJ45), układ zasilania i scalony generator 25 MHz. Moduł kolorowego graficznego wyświetlacza KamodTFT2 (Nokia 6610) jest elementem interfejsu użytkownika i będą na nim wyświetlane istotne informacje. Dodatkowe układy: dwa termometry DS18B20 i dwa przekaźniki umieszczono na dodatkowej płytce uniwersalnej (fotografia 6). Układ jest ze źródła napięcia stałego lub przemiennego 6...7 V i pobiera prąd o natężeniu ok. 240 mA.



Rysunek 7. Element Pushbutton

## Koncepcja sterownika

Sterownik ma za zadanie nadzorować stan domowej centrali alarmowej, oraz umożliwić jej zdalne uzbrajanie i rozbrajanie. Mierzy też i zdalnie przesyła wartość napięcia baterii zasilającej centralkę. Poza tym, mierzy temperaturę wewnątrz i symuluje załączenie grzejnika na podstawie zapisanych progów temperatur. Drugi termometr mierzy temperaturę na zewnątrz. Jest też możliwe zdalne włączenie i wyłączenie głównego wyłącznika mocy (napięcia sieci energetycznej) oraz oświetlenia zewnętrznego. Ostatni element sterujący symuluje sterowanie jasnością oświetlenia ogrodu (na przykład przez zmianę wypełnienia przebiegu PWM).

Projekt strony sterującej podzieliłem na stronę główną i 3 podstrony. Podział na podstrony wynika z wykonywanych przez nie funkcji: sterowania alarmem, sterowania mocą i odczytywania temperatury.

## Sterowanie przyciskiem

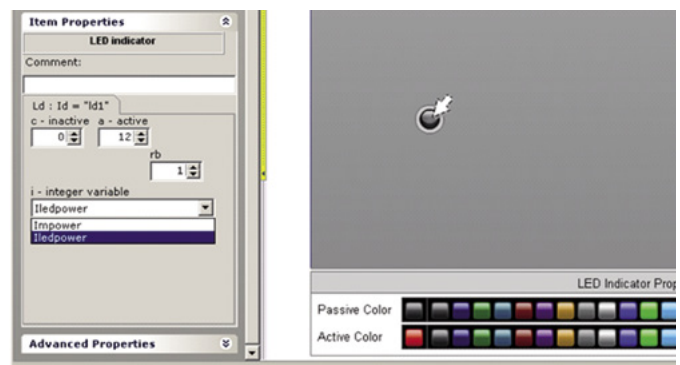
Na początek pokażę jak przygotować sterowanie załączeniem i wyłączeniem urządzeń wykorzystując element *Pb (push button, rysunek 7)*. Polecenie sterownicze jest przesyłane z komputera (strony) do sterownika. Po kliknięciu na niego przenosimy wskaźnik myszki na obszar roboczy i ponownie klikamy. Przycisk zostanie umieszczony na obszarze z domyślnym opisem *\_Title\_*. W oknie *t-text caption* możemy zmienić ten opis na przykład na *ON*. Pod obszarem roboczym pojawi się okno *Pushbutton Properties* pozwalające stanowi aktywnemu (naciśnięciu przycisku) i stanowi nieaktywnemu (zwolnienie przycisku) przypisać kolory z dostępnej palety. W oknie *i-integer variable* do przycisku trzeba przyporządkować wcześniej zdefiniowaną zmienną typu *Integer* – w naszym wypadku będzie to *Impower*. Teraz każde przytoczenie tak zdefiniowanego przycisku spowoduje, że przeglądarka prześle do sterownika zmienną *Impower* z wartością 0x01. W tym momencie to nam wystarczy, a później zobaczymy jak na to zareaguje sterownik. Przycisk *Pb* można wykorzystywać do nawigowania po podstronach wykorzystując do tego celu okno *g- Goto command*. Ta funkcjonalność również zostanie wykorzystana w naszym projekcie.

## Sygnalizacja stanu – dioda LED

Element LED (rysunek 8) jest przeznaczony do jednobitowej sygnalizacji stanu przesyłanej ze sterownika do komputera. Tak jak w wypadku *Pushbutton* i z tym elementem jest związana zmienna *Integer*. Po wybraniu z palety i umieszczeniu w obszarze roboczym w oknie *i-integer (Item properties)* wybieramy wcześniej zdefiniowaną zmienną *Iledpower*, a z palety *LED Indicator Properties* kolor aktywny (dioda świeci się) i kolor nieaktywny (dioda jest „zgaszona”). Gdy sterownik przesyła zmienną *Iledpower* z wartością 0x01, to dioda się „zaświeca się” kolorem aktywnym.

## Wyświetlanie formatowanych danych – Numeric Indicator

*Pushbutton* i *dioda LED* to elementy bardzo przydatne i jednocześnie proste w działaniu. Następny w kolejności element *Numeric Indicator* jest trochę bardziej złożony w edycji, ale daje możliwość



Rysunek 8. Element LED

# KONKURS

## Do wygrania oscylloskop i multimetr!



Redakcja Elektroniki Praktycznej wspólnie z firmą NDN, wyłącznym, autoryzowanym dystrybutorem i serwisem przyrządów pomiarowych firmy Rigol w Polsce, ogłaszają konkurs, w którym nagrodą główną jest zestaw:

### Oscylloskop Rigol DS1052E i multimetr Sanwa PC5000a

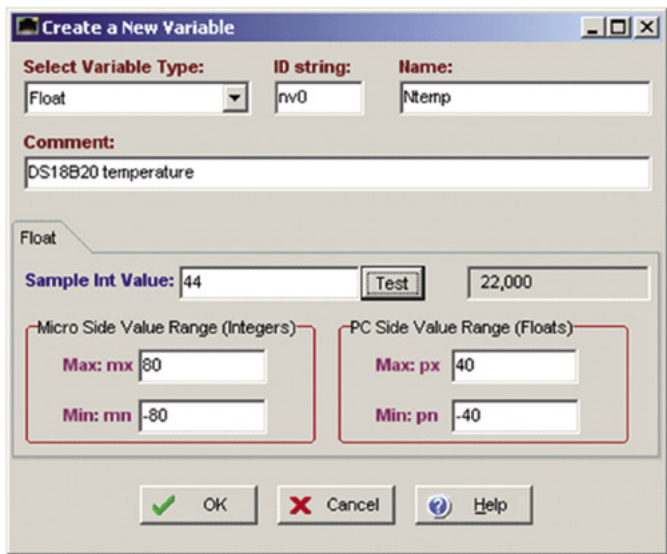
Aby wziąć udział w konkursie należy w terminie do dnia 31 marca 2012 r. wypełnić ankietę umieszczoną na stronie internetowej [www.ep.com.pl](http://www.ep.com.pl) udzielając odpowiedzi na pytanie:

**W którym roku Rigol wprowadził na rynek nowoczesne oscylloskopy cyfrowe o częstotliwości próbkowania 1 GSa/s i paśmie 200 MHz?**

Wyniki konkursu ogłosimy w majowym wydaniu EP oraz na stronie internetowej. Szczegółowe informacje dotyczące nagrody ufundowanej przez firmę NDN: [http://www.ndn.com.pl/index.php?id\\_dzialu=14&id\\_firmy=30#1884](http://www.ndn.com.pl/index.php?id_dzialu=14&id_firmy=30#1884)

**NDN**

**ELEKTRONIKA  
PRAKTYCZNA**



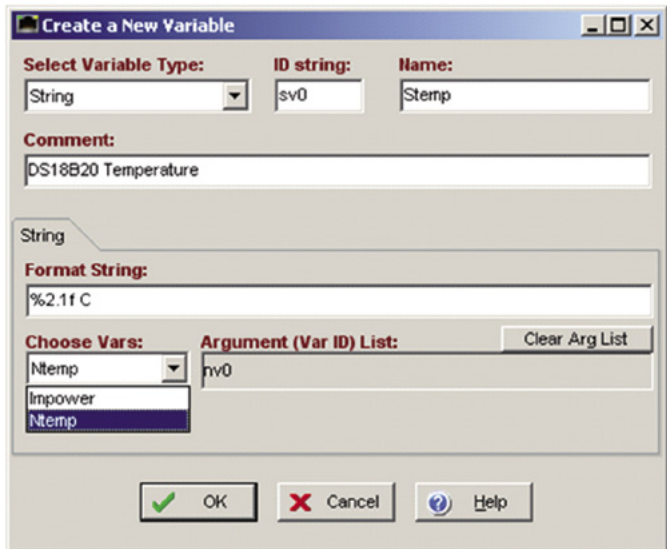
Rysunek 9. Definiowanie zmiennej *Ntemp*

formatowanego wyświetlania zawartości zmiennej *Numeric*. Aby ten element działał prawidłowo trzeba zdefiniować dwie zmienne: pierwszą typu *Numeric*, a drugą typu *String*. Jak to się robi, pokażę na przykładzie wyświetlania temperatury odczytanej z termometru DS18B20.

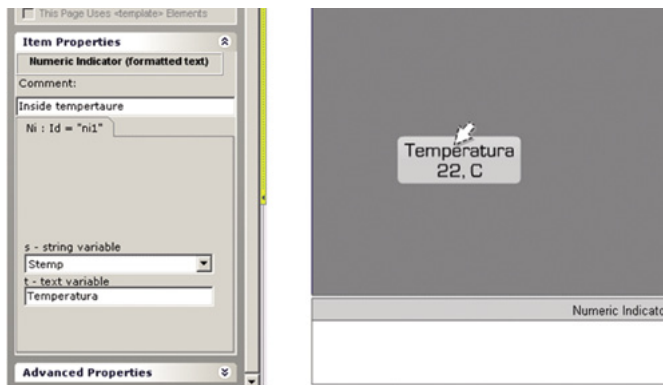
Na początek definiujemy zmienną *Numeric Ntemp*. Do tej zmiennej sterownik będzie wpisywał 9-bitową wartość odczytaną z termometru (rysunek 9). Dla zmiennej *Numeric* w oknie *Select Variable Type* wybieramy *Float*, a w oknie *Name* zmieniamy nazwę na *Ntemp*. Pole *ID string* może pozostać niezmienione.

Teraz jest dobry moment, żeby wyjaśnić jak można ustawić parametry *mx*, *mn*, *px* i *pn*. Liczby *mx* i *mn* określają, w jakich zakresach zmienna *Ntemp* będzie zmieniana w sterowniku. Termometr DS18B20 odczytujący temperaturę z rozdzielczością 9-bitową, czyli zmiana na najmłodszym bicie odpowiada zmianie temperatury o 0,5°C. Wpisujemy *mn*=-80 i *mx*=+80, bo zakładamy, że wartość odczytana z DS18B20 będzie się zmieniała od -80 do +80°C, co przy rozdzielczości 0,5°C daje temperatury z zakresu -40,0 do +40,0°C. Wartość wysłana przez sterownik ma być w komputerze interpretowana jako temperatury od -40,0 do +40,0°C i dlatego wpisujemy *pn*=-40 i *px*=+40. Skalowanie jest elastyczne i bez trudu można dobrać odpowiednie współczynniki. Poprawność skalowania sprawdza się w oknie *Sample Int Value*.

Żeby wyświetlić przesyłaną do komputera wartość zmiennej *Ntemp* trzeba ją skojarzyć ze zmienną typu *string*. W tym celu dodaje-



Rysunek 10. Definiowanie zmiennej typu *String*

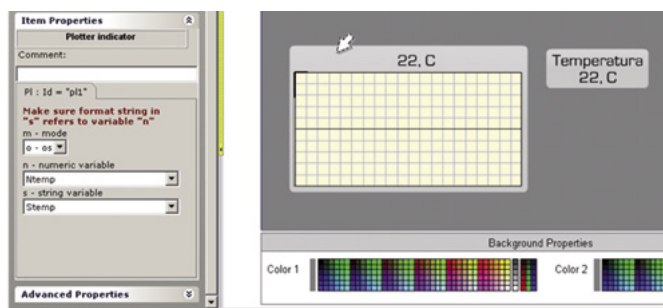


Rysunek 11. Dodanie elementu *Numeric Indicator*

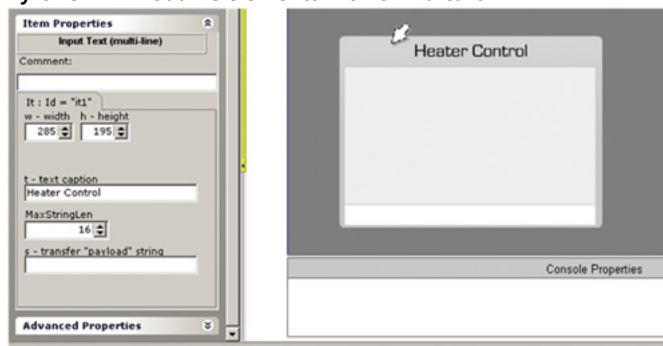
my do listy zmienną typu *string* o nazwie *Stemp* (jak na rysunku 10). Zmienna będzie reprezentować wartość *Ntemp* po dodaniu jej do listy argumentów w oknie *Choose Var*. Pozostaje jeszcze ustalić format wyświetlania za pomocą okna *Format String*. W przykładzie z rys. 11 będą wyświetlane 2 cyfry części całkowitej i 1 cyfra po przecinku oraz litera C (stopnie Celsjusza). Po zdefiniowaniu obu zmiennych można wybrać z palety element *Numeric Indicator* (rysunek 11). W oknie *Properties Item* dodajemy zmienną *Stemp* typu *string*. Wcześniejsze jej powiązanie w definicji ze zmienną *Ntemp* powoduje, że element *Numeric Indicator* będzie wyświetlał sformatowaną wartość *Ntemp* przesyłaną ze sterownika. Żeby było wiadomo co jest wyświetlane, w oknie *t-text variable* wpisujemy „Temperatura”.

### Plotter Indicator

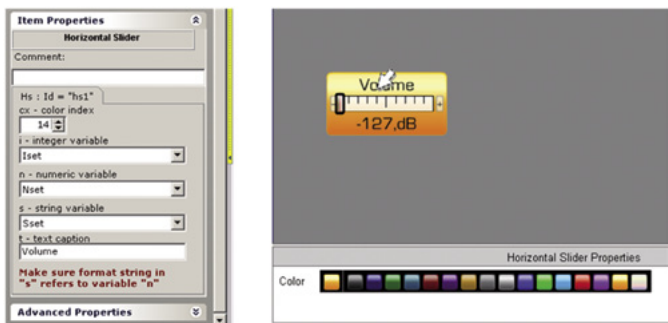
Do graficznego przedstawienia zmian jakiejś wielkości w czasie jest przeznaczony element *Plotter Indicator* (rysunek 12). W naszym projekcie użyjemy go do pokazania przebiegu zmian temperatury przesyłanej w *Ntemp*. Oś pionowa jest skalowana jak zmienna *Temp*, czyli ma zakres -40°C do +40°C. Oś pozioma to interwały czasu, z którymi są przesyłane dane o temperaturze ze sterownika. Do celów testowych przesyłam pomiar co około 3 sekundy. W rzeczywistości można go przesyłać na przykład co godzinę lub dwie. Na ekranie elementu mieści się 20 pomiarów, a po zapelnieniu najstarszy pomiar jest przesuwany poza lewą krawędź. Przebieg temperatury jest rysowany jak sygnał napięcia na ekranie oscyloskopu.



Rysunek 12. Dodanie elementu *Plotter Indicator*



Rysunek 13. Dodanie elementu *Input Text*



Rysunek 14. dodanie elementu *Horizontal Slider*

Po ustawieniu elementu na obszarze roboczym, w oknie *Item Properties* w *n-numeric* wybieramy zmienną *Ntemp*. Wartość tej zmiennej jest graficznie rysowana w polu plotera. Jeżeli dodatkowo w *s-string* wybierzemy *Stemp*, to na górze okna będzie wyświetlana wartość temperatury, dokładnie tak samo, jak w elemencie *Numeric Indicator*.

### Input Text

Sterownik oprócz stanu diody LED może przysyłać do komputera komunikaty tekstowe. Do tego celu jest używany element *Input Text* (rysunek 13). Po wybraniu z palety i postawieniu w obszarze roboczym, w oknie *Item properties* można nadać nazwę nagłówka na przykład *Heater Control*, lub *Alarm Status*. W oknie *MaxStringLength* ustawia się maksymalną długość komunikatu, a okno *s-transfer* może pozostać puste. Do tego elementu nie przypisuje się żadnej zmiennej z listy zmiennych, ale TCPmaker w pliku źródłowym sterownika tworzy automatycznie tablicę o długości równej *MaxStringLength*. Każdy nowy element typu *Input Text* ma swoją tablicę, w której zapisuje komunikaty wysyłane do komputera.

### Nastawa wartości – element Slider

Slider jest jednym z ciekawszych elementów TCPMaker'a. Jest wykorzystywany do przesyłania danej, której wartość jest zależna od położenia suwaka. Slider może być poziomy lub pionowy i przypomina graficzne przedstawienie potencjometru suwakowego. Z tym elementem mogą być związane 3 zmienne:

- Zmienna typu Integer. Jest używana do przesyłania informacji, czy suwak został „przyciśnięty”. Po najechaniu kursora myszki, żeby móc przesunąć suwak trzeba przycisnąć i przytrzymać lewy klawisz. Wtedy ta zmienna zmienia wartość na 0x01.
- Zmienna typu Numeric. W czasie przesuwania suwaka na ekranie komputer przesyła do sterownika ciąg zmieniających się wartości tej zmiennej.
- Zmienna typu String. Jeżeli jest skojarzona ze zmienną Numeric, to wyświetla jej wartość.

Na **rysunku 14** pokazano regulator poziomu sygnału w zdalnie sterowanym przedwzmacniaczu audio. Zmienna *Nset* ma ustawione  $mn=0$  oraz  $mx=255$ , czyli przesunięcie suwaka w całym zakresie spowoduje przesłanie wartości od 0 do 255. Współczynniki skalowania po stronie komputera są równe  $pn=-127$ , a  $px=0$ , co umożliwia wyświetlanie od  $-127$  dB (dla  $Nset=0$ ) do 0 dB (dla  $Nset=255$ ). Zmienna *Sset* ma w liście argumentów wpisaną zmienną *Nset*.

### Mierzenie wartości – element Gauge

Ten element pokazuje graficznie wartość zmiennej przesyłanej przez sterownik do komputera. Może mieć kilka reprezentacji graficznych: kolorowane bargrafy pionowe i poziome oraz mierniki wskaźkowe. Oprócz pokazywania wartości mogą też sygnalizować kolorami zakresy prawidłowe (kolor zielony), ostrzegawcze (kolor żółty) i alarmowe (kolor czerwony). Zakresy można dowolnie dostosowywać do potrzeb.

Z elementem tym są związane 2 zmienne:

- Numeric w której jest zwarta przesyłana dana ze sterownika do komputera.



## Przekładniki prądowe

### Szeroka gama czujników do pomiaru prądów DC i AC

- Technologia Fluxgate i Hall-Effect
- Rozwiązania z otwartą i zamkniętą pętlą
- Szeroki zakres mierzonych prądów: od 5 do 2000 A RMS
- Bardzo wysoka dokładność
- Podłączenie PCB przez konektory lub kable przyłączeniowe

Dystrybutor:

**Semicon Sp. z o.o.**

ul. Zwoleńska 43

04-761 Warszawa

tel. 22 615-73-71

fax 22 615-73-75

info@semicon.com.pl

www.semicon.com.pl

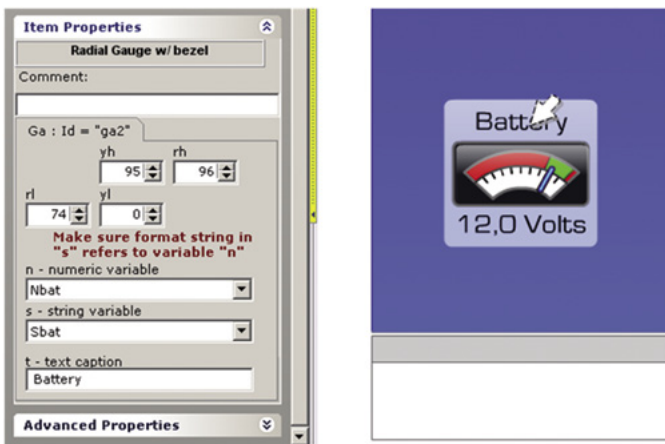


Zapraszamy na nasze stoiska na targach



**CONTROL-STOM**  
Kielce 27-29 marca

20-23 marca  
Hala 3, stoisko F14



Rysunek 15. Dodanie elementu *Radial Gauge*

- String przeznaczona do wyświetlania wartości tej danej.

W projekcie sterownika użyłem tego elementu do przesyłania i wyświetlania napięcia baterii zasilającej centralkę alarmową (rysunek 15).

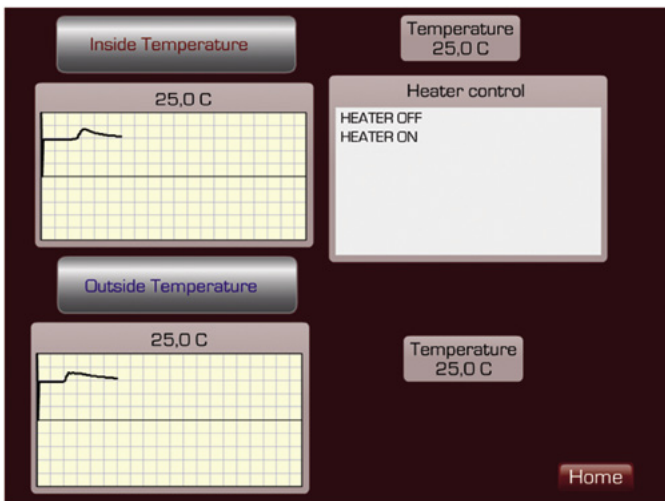
Zakresy ostrzegawcze są zmieniane przez zapisywanie ustawień *yh*, *rh*, *rl* i *yl*. Jeżeli przesyłana wartość jest z zakresu alarmowego, to cały element jest obwiedziony czerwoną ramką dodatkowo sygnalizującą odstępstwo od normy.

### Strona główna i podstrony

Projekt strony może się składać z jednej lub kilku stron. W przypadku bardziej rozbudowanych projektów wygodniej jest podzielić



Rysunek 16. Strona główna (index)



Rysunek 17. strona pomiaru temperatury



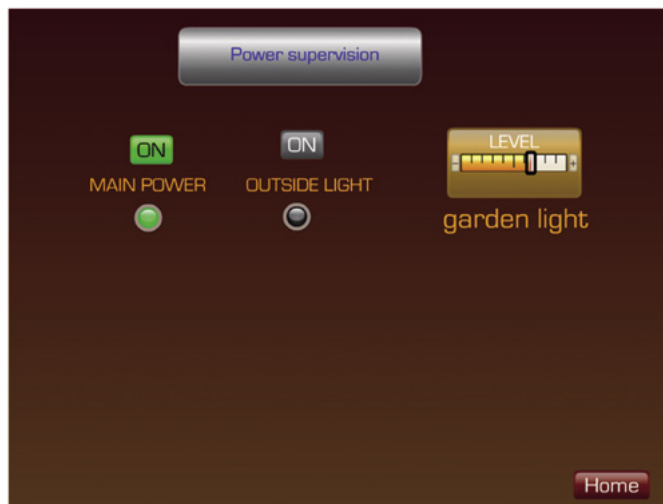
Rysunek 18. Strona nadzoru alarmu

sterowanie na grupy i umieścić je na oddzielnych stronach. W sterowniku podzieliłem zadanie na 3 grupy: sterowanie zasilaniem, pomiar temperatury i nadzór nad systemem alarmowym. Na stronie głównej umieściłem sygnalizację zadziałania alarmu i przyciski nawigacyjne dostępu do podstron.

Operacje na stronach wykonuje się w oknie *Commands*. Można dodać nową stronę, skopiować istniejącą, wyczyścić, i wykasować. Każda ze stron ma swój identyfikator: strona główna *index*, a pozostałe *pg1*, *pg2* itd. Na kolejnych stronach umieszczamy niezbędne elementy z palety, ale musimy mieć możliwość nawigowania pomiędzy stronami. Do tego celu jest przeznaczony element *Pushbutton*. Wykorzystanie go w celu poruszania się pomiędzy stronami projektu nie wymaga definiowania zmiennej *i-integer variable*, ale trzeba użyć pola *g-goto command* w oknie *Item Properties*. Na stronie głównej można umieścić przyciski wywołujące podstrony projektu. Przycisk powodujący przejście do strony 1 ma wpisane w pole *g-goto commands* `<pg1/>`, do *strony 2* `<pg2/>` itd. Każda z podstron ma przycisk HOME z komendą `<index/>`. Na **rysunku 16** pokazano projekt strony głównej nadzoru domu z przyciskami nawigacji po pozostałych stronach i z diodą LED sygnalizująca zadziałanie alarmu. Kolejne strony projektu są pokazane na **rysunkach 17...19**. Każda z nich ma przycisk powrotu do strony głównej *Home*.

W kolejnym odcinku kursu dokończymy nasz projekt sterownika i opiszemy sposób jego działania. Pokażemy też, że nie „taki TCP/IP straszny, jak go malują”. A wszystko dzięki TCP Maker'owi.

**Tomasz Jabłoński, EP**



Rysunek 19. Strona sterowania zasilaniem