

Komponent Delphi do tworzenia paneli cyfrowych

Wielu konstruktorów miewa kłopoty z napisaniem efektywnego oprogramowania współpracującego z opracowanym urządzeniem.

Ze względu na rozpowszechnienie się systemu operacyjnego Windows, skala trudności podczas pisania programów nieustannie rośnie i to pomimo powstawania coraz doskonalszych narzędzi programistycznych.

Aby nieco ułatwić konstruktorom rozwiązywanie najczęściej spotykanych problemów, przedstawiamy pierwszy z serii artykułów, prezentujących gotowe, typowo „elektroniczne” komponenty, które można wykorzystać we własnych programach.

Dylemat - czy używać Windows, czy też nie - został już dawno rozstrzygnięty. Nowe opracowania oprogramowania dla elektroniki pracują tylko w środowisku Windows. Idąc zatem śladem tego trendu proponujemy opracowania, które nam - elektronikom, lecz nie najlepszym programistom - mogą ułatwić budowanie własnych aplikacji okienkowych, służących do sterowania z komputera naszych urządzeń.

Autor wybrał środowisko Delphi, jako dosyć wygodne do tworzenia aplikacji okienkowych, zwalniające z konieczności wglębiania się w wiele programistycznych szczegółów i pozwalające na skupienie się na rozwiązywaniu tylko zasadniczych problemów.

W Delphi okno konstruowanej aplikacji wypełniają się komponentami, które posiadają własności, czyli cechy charakterystyczne kształtowane zależnie od potrzeb. Ponadto, z komponentami są związane metody, czyli procedury wykonujące określone czynności na zasobach komponentu oraz obsługę zdarzeń, czyli możliwość reakcji komponentu na zachowanie się jego otoczenia.

Komponent TSevenSegment

Komponent *TSevenSegment* umożliwia konstruowanie w oknie aplikacji paneli cyfrowych, składających się z wyświetlaczy siedmiosegmentowych. Każdy wskaźnik jest wyposażony w kropkę dziesiętną. Trzy tryby pracy wyświetlacza pozwalają na zobrazowanie cyfr dowolnego systemu liczbowego, od dwójkowego po szesnastkowy, wyświetlenie zamiast cyfr znaku dwukropka, tudzież dowolnego układu segmentów. Dodatkowe własności umożliwiają wirtualne połączenie wielu komponentów *TSevenSegment* w panele cyfrowe i wprowadzanie wielocyfrowej liczby na wejście

najmłodszej cyfry, inkrementację bądź dekrementację wyświetlanej wartości o dowolną liczbę.

Własności komponentu TSevenSegment

Własność *BlankWhenZero*

Własność *BlankWhenZero* określa sposób wyświetlania cyfry 0. Przyjmuje ona wartości *True* albo *False*. Kiedy *BlankWhenZero=True*, wszystkie segmenty cyfry są wygaszone, zaś gdy *BlankWhenZero=False*, zero jest wyświetlane normalnie. Własność ta może być zmieniana, kiedy własność *DisplayMode=dmDigit* i domyślnie przyjmuje wartość *False*.

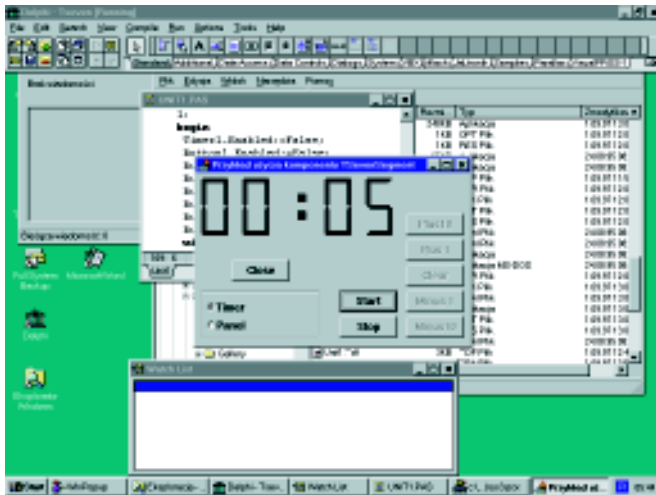
Własność *Cursor*

Własność *Cursor* jest dziedziczona z poprzednich, standardowych obiektów i przyjmuje wartość *crDefault*.

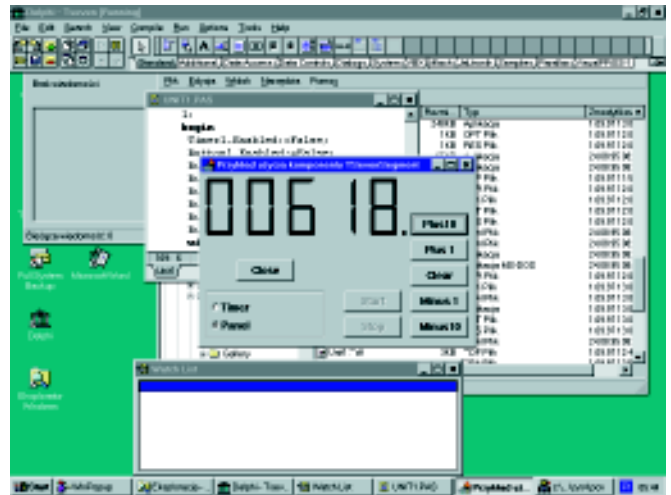
Własność *DisplayMode*

Własność *DisplayMode* określa tryb pracy komponentu i przyjmuje następujące wartości:

- ✓ *dmBin* - w trybie *dmBin* komponent jest traktowany jako zbiór ośmiu segmentów, przy czym numeracja segmentów odpowiada typowej numeracji spotykanej w katalogach wyświetlaczy siedmiosegmentowych. Stan segmentu jest zależny od wartości najmłodszego bitu własności *Value*, a segmentowi *h* odpowiada wartość najstarszego bitu najmłodszego bajtu *Value*. Zero na danym bicie wygasza segment, a jedynka go zapala.
- ✓ *dmColon* - w trybie *dmColon* jest wyświetlany dwukropek, którego stan zapalenia jest zależny od własności *Dot*.
- ✓ *DmDigit* - w trybie *dmDigit* komponent interpretuje wszystkie cyfry szesnastkowe. W zależności od własności *MaxValue*, zakres interpretowanych cyfr może być skrócony.



Rys. 1. Panel cyfrowy jako timer.



Rys. 2. Panel cyfrowy jako licznik.

Własność Dot

Własność *Dot* określa stan kropki dziesiętnej albo dwukropka i przyjmuje wartości *True* albo *False*. W zależności od stanu własności *DisplayMode*, komponent różnie reaguje na własność *Dot*.

DisplayMode=dmBin

W trybie *dmBin* własność *Dot* jest ignorowana, wtedy kropka dziesiętna jest wyświetlana niezależnie od stanu siódmego bitu własności *Value*.

DisplayMode=dmColon

W trybie *dmColon* własność *Dot* odpowiada za stan wyświetlanego dwukropka. Dwukropek jest wyświetlany, jeśli *Dot=DotOn*, a jest wygaszany, kiedy *Dot=DotOff*.

DisplayMode=dmDigit

W trybie *dmDigit* własność *Dot* odpowiada za stan wyświetlanej kropki dziesiętnej. Kropka jest wyświetlana, jeśli *Dot=DotOn*, a jest wygaszana, kiedy *Dot=DotOff*.

Własność ForwardConnect

Własność *ForwardConnect* pozwala łączyć komponenty *TSevenSegment* w panelu cyfrowym. Jeśli w danym komponencie zostanie wybrany inny komponent typu *TSevenSegment*, to względem danego komponentu stanowi on starszą pozycję znakową tworzonego panelu.

Dla przykładu, w oknie aplikacji możemy mieć trzy komponenty: *SevenSegment1*, *SevenSegment2* i *SevenSegment3*. We własności *ForwardConnect* kom-

ponentu *SevenSegment3* zapiszemy *SevenSegment2*, a we własności *ForwardConnect* komponentu *SevenSegment2* zapiszemy *SevenSegment1*. Jeśli własność *ParallelLoad* jest we wszystkich komponentach ustawiona jako *True*, to zapis liczby 567 do własności *Value* komponentu *SevenSegment3*, przy domyślnych ustawieniach własności *MaxValue* spowoduje, że *Value* komponentu *SevenSegment1* wyniesie 5, *Value* komponentu *SevenSegment2* wyniesie 6, a *Value* komponentu *SevenSegment3* wyniesie 7. Ponadto, własność *ForwardConnect* pozwala na poprawne działanie metod *Decrement* i *Increment*. Domyślną wartością *ForwardConnect* jest *nil*.

Własność Height

Własność *Height* określa wysokość komponentu wyrażoną w pikselach. Domyślną wartością *Height* jest 131.

Własność Hint

Własność *Hint* jest dziedziczna i domyślnie jest pustym łańcuchem znakowym.

Własność Left

Własność *Left* jest dziedziczna i określa ona składową X położenia lewego górnego narożnika komponentu w oknie aplikacji.

Własność MaxValue

Własność *MaxValue* określa zakres dostępnych cyfr. *MaxValue* przybiera wartości z zakresu 1..15. Jest ona dostępna tylko w trybie *DisplayMode=dmDigit*.

Domyślną wartością tej własności jest 9.

Własność Name

Własność *Name* występuje dla każdego komponentu i jest oryginalną nazwą komponentu w aplikacji-właścicielu.

Własność OffColor

Własność *OffColor* określa kolor wygaszonego segmentu. Domyślną wartością tej własności jest *clBtnFace*.

Własność OnColor

Własność *OnColor* określa kolor wygaszonego segmentu. Domyślną wartością tej własności jest *clRed*.

Własność ParallelLoad

Własność *ParallelLoad* przyjmuje wartości *True* albo *False*. Jeśli jest ona równa *True*, działa właściwość komponentu do przenoszenia nadmiaru na wyższe pozycje znakowe, w oparciu o własność *ForwardConnect*, przy zapisie do *Value* liczby większej niż zapisana w *MaxValue*. Własność *ParallelLoad* jest ignorowana w metodach *Decrement* i *Increment*.

Własność Proportional

Własność *Proportional* przyjmuje wartości *True* albo *False*. Jeśli jest ona równa *True*, stosunek wysokości (własność *Height*) do szerokości (własność *Width*) komponentu jest stały i wynosi 26/16. Z pary obu własności - *Height* i *Width* - *Height* jest niezależna, czyli jej zmiana

spowoduje zmianę wymiarów komponentu.

Jeśli własność *Proportional* jest równa *False*, stosunek wysokości do szerokości może być dowolny.

Własność *Tag*

Własność *Tag* jest dziedziczona.

Własność *Top*

Własność *Top* jest dziedziczona i określa ona składową Y położenia lewego górnego narożnika komponentu w oknie aplikacji.

Własność *Visible*

Własność *Visible* jest dziedziczona i określa, czy dany komponent będzie widoczny, czy nie.

Własność *Width*

Własność *Width* określa wysokość komponentu wyrażoną w pikselach. Domyślną wartością *Width* jest 81.

Zdarzenia wykrywane przez komponent

Poniżej są przedstawione zdarzenia wykrywane przez komponent *TSevenSegment*. Odpowiedzią na dane zdarzenie jest wywoływana metoda, wcześniej zdefiniowana w inspektorze obiektów i ręcznie wypełniona treścią. Wszystkie zdarzenia są dziedziczone.

Zdarzenie *OnClick*

Zdarzenie *OnClick* jest rejestrowane, jeśli kursor myszki jest nad komponentem i zostanie kliknięty jeden z przycisków.

Zdarzenie *OnDbClick*

Zdarzenie *OnDbClick* jest rejestrowane, jeśli kursor myszki jest nad komponentem i zostanie dwukrotnie kliknięty lewy przycisk.

Zdarzenie *OnMouseDown*

Zdarzenie *OnMouseDown* jest rejestrowane, jeśli kursor myszki jest nad komponentem i jeden z klawiszy myszki jest wciśnięty.

Zdarzenie *OnMouseMove*

Zdarzenie *OnMouseMove* jest rejestrowane, jeśli kursor myszki jest przesuwany nad komponentem.

Zdarzenie *OnMouseUp*

Zdarzenie *OnMouseUp* jest

rejestrowane, jeśli kursor myszki jest nad komponentem i został puszczony naciśnięty klawisz myszki.

Metody komponentu

Metoda *Clear*

Metoda *Clear* zeruje wartość *Value*. Żeby za pomocą tej metody wyzerować cały panel należy we wszystkich komponentach ustawić wartość *ParallelLoad* równą *True* i wypełnić odpowiednio własność *ForwardConnect*.

Metoda *Decrement*

Metoda *Decrement(Dec)* pozwala zmniejszyć wartość *Value* o zadaną liczbę *Dec*. Przy odpowiednio ustawionej własności *ForwardConnect* odejmowanie odbywa się na liczbie wielopozycyjnej. Metoda *Decrement* nie korzysta z własności *ParallelLoad*.

Metoda *Increment*

Metoda *Increment(Inc)* pozwala zwiększyć wartość *Value* o zadaną liczbę *Inc*. Przy odpowiednio ustawionej własności *ForwardConnect* dodawanie odbywa się na liczbie wielopozycyjnej. Metoda *Increment* nie korzysta z własności *ParallelLoad*.

Dostępność oprogramowania

Komponent jest rozpowszechniany w wersji shareware. Na stronie AVT, w pliku **SEVENSE.DCU** znajduje się wersja nie zarejestrowana, która jest wersją freeware. Do tego jest dołączony plik **SEVENSE.DCR** z ikoną, którą można umieścić w palecie komponentów. Warunki rejestracji i opis komponentu znajdują się w pliku **SEVENSE.DOC**. Dodatkowo proponujemy prostą aplikację (plik **TSEVEN.EXE**), której efekty działania przedstawione zostały na **rys.1** i **rys.2**.

Mirosław Lach, AVT

Oprogramowanie jest dostępne w Internecie pod adresem
<http://www.atm.com.pl/~avt/ep/download.htm>