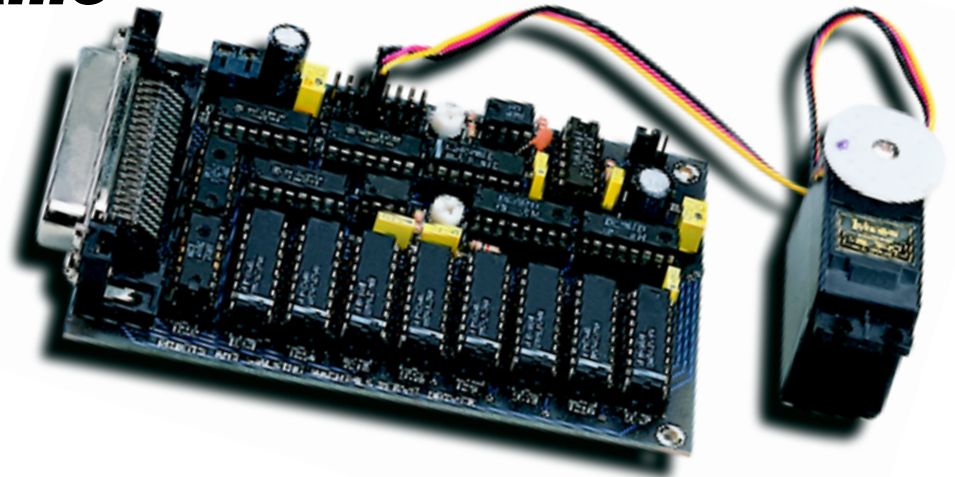


Robot, część 3

Oprogramowanie

AVT-821

W trzeciej, ostatniej części artykułu przedstawiamy opis oprogramowania sterującego „ręką” robota. Opis procedur przygotowanych przez autora ułatwi Czytelnikom przygotowanie oprogramowania spełniającego ich indywidualne wymagania.



Informacje ogólne

Program *Raabot* jest dostarczany z kitem AVT-821. Służy do edytowania, zapisywania i odtwarzania kolejnych położenia serwo-mechanizmów stanowiących elementy wykonawcze robota. Program przeznaczony jest do pracy w systemie operacyjnym Windows 95 lub 98. Nie będzie działał prawidłowo z wcześniejszymi wersjami Windows i z Windows NT. Został napisany za pomocą Delphi 3, z wykorzystaniem funkcji API32. Niemożliwa jest więc jego współpraca ze starszymi, 16-bitowymi wersjami Windows. Nie pomoże nawet zainstalowanie Win32 do Windows wersji 3.1 i 3.11. Ponieważ program odwołuje się bezpośrednio

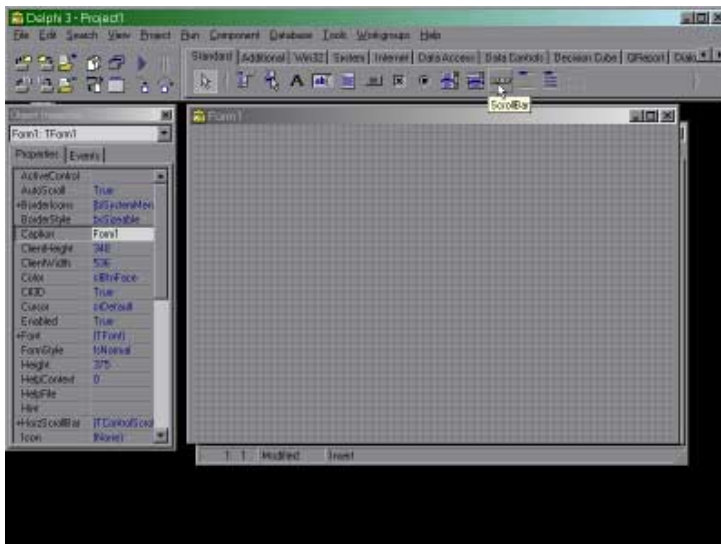
do portów za pomocą wstawek assemblerowych, co jest niedozwolonym rozwiązaniem w systemie Windows NT, może działać niepoprawnie również w tym systemie.

Porty, adresy

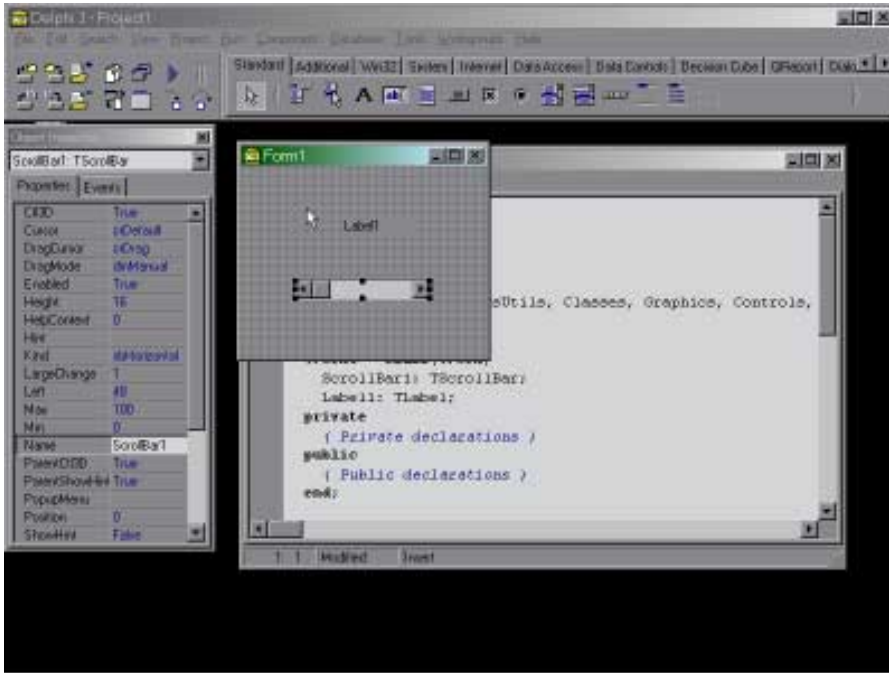
Program *Raabot* steruje układem z kitu AVT-821 za pośrednictwem łącza równoległego. W programie przewidziana jest współpraca ze standardowymi portami LPT1 (adres 378h) i LPT2 (adres 278h). Zmiany portu można dokonać wybierając odpowiednią pozycję w menu Konfiguracja. W programie nie przewidziano współpracy z portami ustawionymi niestandardowo.

Instalowanie i odinstalowywanie

Aby zainstalować to oprogramowanie należy uruchomić program *setup.exe* z dyskietki instalacyjnej. Program instalacyjny skopiuje pliki oprogramowania na dysk twardy i utworzy grupę programu *Raabot* w Menu Start. Program zapisuje w systemie dane konfiguracyjne w pliku rejestrów. Dlatego, aby program odinstalować całkowicie, należy skorzystać z polecenia *Dodaj/Usuń Programy* z Panelu Sterowania lub z programu *Usuń Raabota*, znajdującego się w grupie programu *Raabot*. Nie zaleca się ręcznego usuwania programu. Nie stanowi to żadnego niebezpieczeństwa dla systemu, a nawet nie zaśmieca go w istot-



Rys. 5. Otwarte okno nowej aplikacji w Delphi.



Rys. 6. Sposób lokalizowania suwaka.

nym stopniu. Jest to po prostu nieeleganckie.

Programy instalacyjny i odinstalowywujący są całkowicie zgodne ze standardami obowiązującymi w systemach Windows 95 i 98.

Uwagi dla użytkowników

Program służy do ustawiania, modyfikacji, zapisywania i odtwarzania kolejnych położenia każdego z ośmiu serwomechanizmów oddzielnie. Przewidziana jest moż-

liwość zapisania do 10000 kolejnych położenia każdego z serwomechanizmów. Położenie każdego z suwaków odpowiada aktualnemu położeniu odpowiedniego serwomechanizmu. Aby zapamiętać bieżące położenia serwomechanizmów i przejść do kolejnej pozycji naciśnij *Następny*. Przyciski: *Poprzedni*, *Pierwszy* i *Ostatni* umożliwiają zmianę aktualnej pozycji. Przyciski *Wstaw* i *Usuń* umożliwiają dodawanie i usuwanie pozycji.

Pola znajdujące się po lewej stronie suwaków sterujących serwami pozwalają na wpisanie własnej nazwy danego serwomechanizmu. W konkretnym modelu pozwala to nadawać nazwy poszczególnym serwom, np. szczytce czy obrót przedramienia. Umożliwia to łatwą identyfikację aktualnie edytowanego stopnia swobody Raabota. Pola znajdujące się po prawej stronie suwaków wskazują aktualne położenie serwa w przedziale od 0 do 255.

Dla każdej pozycji można indywidualnie ustawić prędkość poruszania się serwomechanizmów za pomocą wskaźników wyboru (radio button), znajdujących się po prawej stronie w grupie *Prędkość*. Przewidziano trzy różne prędkości.

Włączenie odtwarzania całej sekwencji położenia można dokonać klikając na przełącznik *On/Off*. Przewidziano również odtwa-

rzanie całej sekwencji w pętli.

Szybka edycja położenia serwomechanizmów w kolejnych pozycjach możliwa jest przy wykorzystaniu klawisza TAB i kursorów.

Całą utworzoną sekwencję ruchów można zapisać w pliku w sposób standardowy dla systemu Windows 95/98.

Jak działa program

Delphi jest wyjątkowo wdzięcznym narzędziem programistycznym. Stanowi bardzo inteligentne i intuicyjne połączenie starego, powszechnie lubianego Pascala, środowiska graficznego i obiektowych technik programowania. Programowanie w nim jest bardzo proste, nawet jeżeli ktoś od Delphi zaczyna programowanie pod Windows.

Serce aplikacji Raabot stanowi króciutka, standardowa procedura zapisująca do portu o podanym adresie wartość podaną jako parametr:

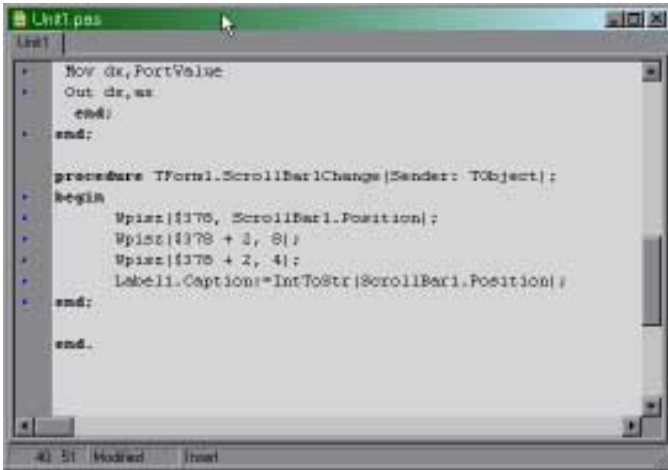
```
procedure Wpisz(PortValue, DataValue:
word);
begin
  DataValue := (DataValue * 256) + DataValue;
asm
  Mov ax,DataValue
  Mov dx,PortValue
  Out dx,ax
end;
end;
```

To co znajduje się pomiędzy słowami kluczowymi asm i end, to właśnie wspomniana wcześniej wstawka assemblerowa zapisująca do portu o adresie *PortValue* wartość *DataValue*. Jednak, aby poinformować urządzenie AVT-821 o położeniu kolejnego serwa, należy do portu zapisać cały ciąg informacji.

```
function DoPortu(P: Pointer): integer;
var
  i: integer;
begin
  repeat
    for i := 1 to 8 do
      // dla każdego serwa
    begin
      Wpisz(nrportu, a[i]);
      // podaj położenie
      Wpisz(nrportu + 2, 7 + i);
      // wraz z informacją o którym
      // serwie mowa
    end;
  end;
end;
```



Rys. 7. Okno definiujące parametry suwaka.



Rys. 8. Okno edytora tekstu z programem obsługi suwaka.

```
Wpisz(nrportu + 2, 4);
// koniec przekazu :-)
end;
until blad=true;
end;
```

W macierzy $a[8]$ są zapisane aktualne położenia wszystkich ośmiu serw.

I to wszystko!

Tym, którzy chcieliby się przekonać, jak proste jest dziś programowanie pod Windows proponuję następujący eksperyment: spróbujcie napisać własny program do sterowania Raabotem w 10 minut! Najpierw powinniście rozejrzeć się za darmową wersją Delphi. Na przykład Delphi 2 było wielokrotnie publikowane na krążkach dołączanych do czasopism o tematyce "komputerowej".

Po otwarciu nowego projektu aplikacji (Menu File | New Application) powinniśmy zobaczyć na ekranie komputera sytuację jak na rys. 5.

Okno zatytułowane Form1, to nasza aplikacja. Powinniśmy dodać do niego jakieś elementy. Na przykład suwak (ScrollBar) i opis tekstowy (Label). W tym celu powinniśmy na pasku narzędzi znaleźć potrzebne nam elementy i dodać je do naszego projektu poprzez dwukrotne kliknięcie. Elementy te możemy na naszym oknie dowolnie przesuwać za pomocą myszy. Ustawmy je mniej więcej tak, jak to widać na rys. 6.

W oknie *Object Inspector* musimy teraz zmienić jedną z właści-

wości naszego paska. Właściwość *Max* odpowiedzialną za to, jaką maksymalną wartość zwraca pasek, zmieniamy z domyślnej wartości 100 na 255 (rys. 7).

Trzecie okno, jakie mamy do dyspozycji, to edytor kodu źródłowego. Zawiera on w tej chwili jedynie generowany au-

tomatycznie kod, który musimy teraz uzupełnić o napisane przez nas procedury.

Pod słowem kluczowym *implementation* wpiszmy naszą procedurkę *Wpisz(PortValue, DataValue: word)* podaną wyżej.

Potem kliknijmy dwukrotnie na pasku *ScrollBar1*. Pojawi się kod źródłowy tworzonej aplikacji z gotowym szkieletem procedury, która będzie wywoływana za każdym razem, gdy przesuniesz pasek *ScrollBar1*, czyli procedurę *TForm1.ScrollBar1Change(Sender: TObject)*. Pomiedzy słowa kluczowe *begin* i *end* wpiszmy:

```
Wpisz($378, ScrollBar1.Position);
Wpisz($378 + 2, 8);
Wpisz($378 + 2, 4);
Label1.Caption:=IntToStr
  (ScrollBar1.Position);
```

Całość, wraz z kodem generowanym automatycznie, powinna wyglądać następująco:

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils,
  Classes, Graphics, Controls, Forms,
  Dialogs,
  StdCtrls;

type
  TForm1 = class(TForm)
    ScrollBar1: TScrollBar;
    Label1: TLabel;
    procedure ScrollBar1Change(Sender: TObject);
  private
```

```
{ Private declarations }
public
  { Public declarations }
end;

var
  Form1: TForm1;

implementation

{$R *.DFM}

procedure Wpisz(PortValue, DataValue:
word);
begin
  DataValue := (DataValue * 256) +
DataValue;
  asm
  Mov ax,DataValue
  Mov dx,PortValue
  Out dx,ax
  end;
end;

procedure TForm1.ScrollBar1Change(Sender: TObject);
begin
  Wpisz($378, ScrollBar1.Position);
  Wpisz($378 + 2, 8);
  Wpisz($378 + 2, 4);
  Label1.Caption:=IntToStr(Scroll-
  Bar1.Position);
end;
end.
```

I to wszystko! Pozostało nam nacisnąć F9, czyli skompilować projekt i cieszyć się z własnego programu, który na ruch suwaczka reaguje ruchem pierwszego serwa naszego Raabota. To naprawdę proste. Zachęcam do eksperymentowania. Jeżeli jesteście zainteresowani tą tematyką napiszcie do nas.

Być może przygotujemy publikację wyjaśniającą sposoby zapisywania i odczytywania portów z poziomu Delphi.

Adam Dębowski, AVT



Rys. 9. Okno działającego programu.