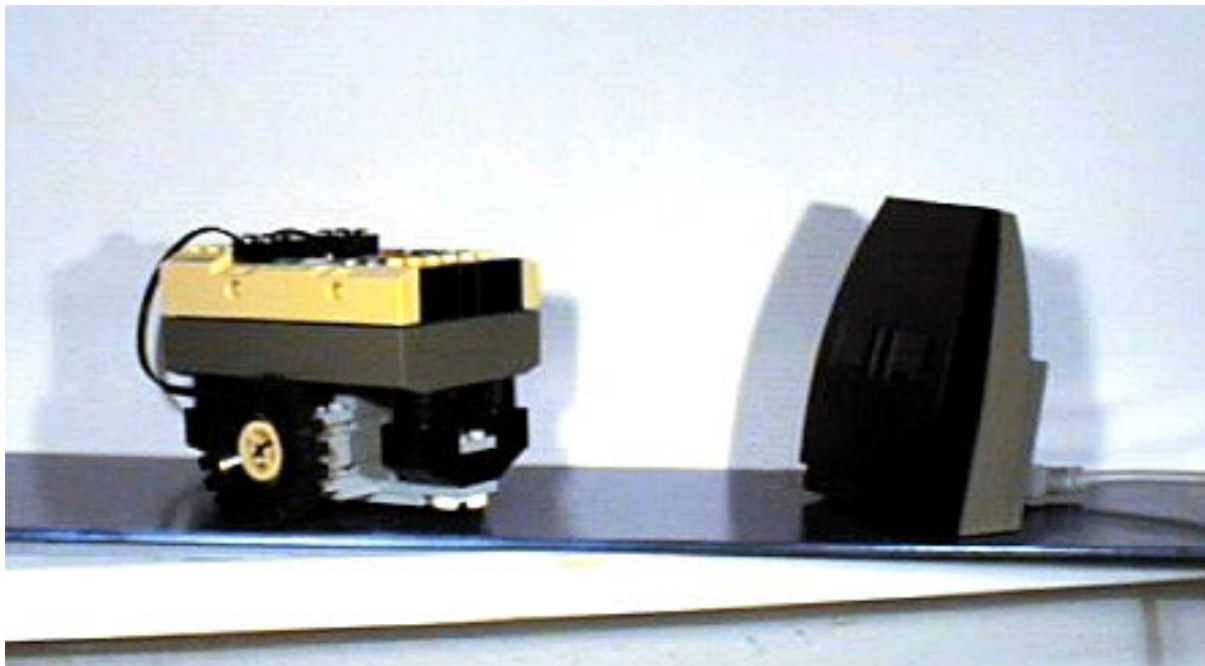


LEGO-Mindstorms-Roboter im Informatikunterricht -mit Delphi-



Benutzung in einer Programmierumgebung

Für die LEGO-Roboter stehen unter allen gängigen Betriebssystemen unterschiedliche Entwicklungsumgebungen und Programmiersprachen zur Verfügung, z. B. NQC (Not Quite C), LEGOS und andere. Dazu gibt es die grafischen Programmierertools von LEGO (Mindstorms Invention System) und das Robolab. Auf diese Systeme wird hier nicht weiter eingegangen, weil gut dokumentierte Software im Internet zu finden ist. Hier werden die LEGO-Roboter mit Hilfe von ActiveX-Komponenten angesprochen, die unter Windows mit allen gängigen Entwicklungsumgebungen nutzbar ist, also z.B. mit Microsofts VisualStudio oder Borlands Delphi. Damit muss die aus dem Unterricht bekannte Entwicklungsumgebung nicht verlassen werden. Statt völlig neu zu beginnen wird auf Bekanntem aufgebaut.

Arbeiten mit der PBRICK-Komponente

Die benötigte Bibliothek findet man nach Installation der LEGO-Software im Verzeichnis \Programme\LEGO MINDSTORMS\system unter dem Namen SPIRIT.OCX. Alternativ dazu befindet sie sich bei den beigegeführten Materialien. Als Beispiele wollen wir die Software unter Borland Delphi benutzen.

Einbinden der ActiveX-Komponenten unter Delphi

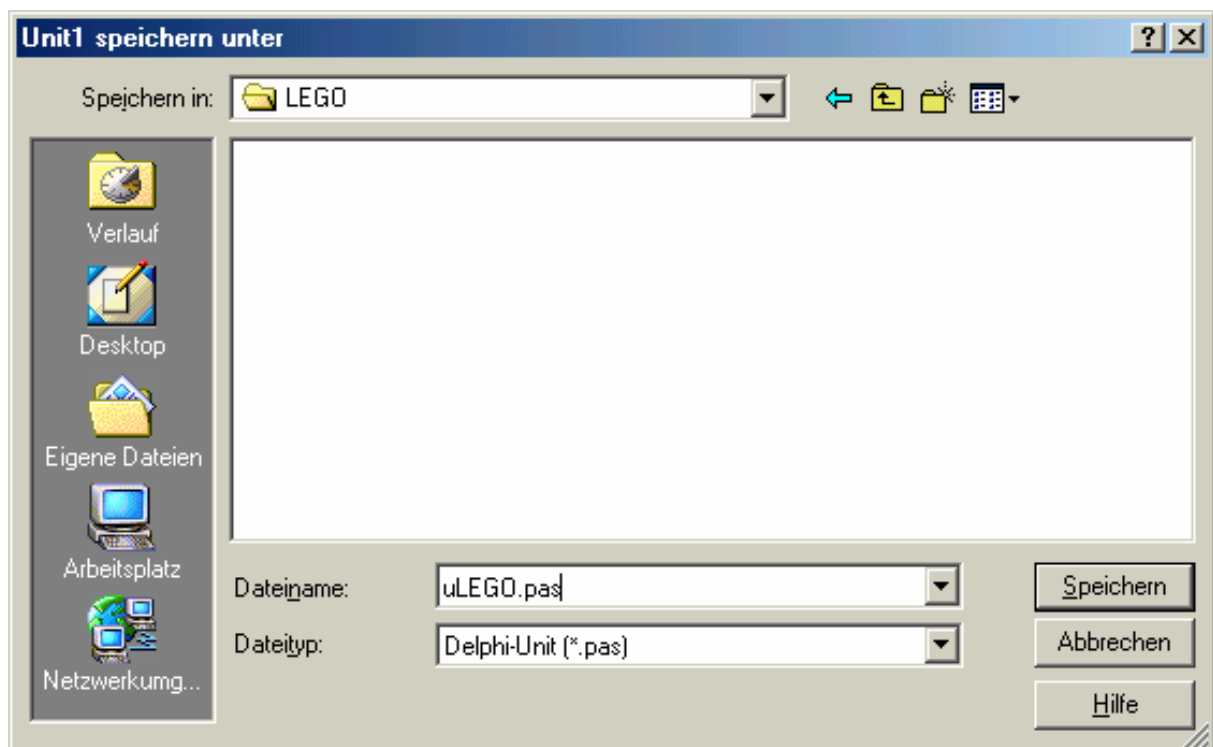
a: Registrieren von Steuerelementen

Falls Sie die LEGO-Mindstorms-Software installiert haben ist die Spirit-Komponente im System registriert. Andernfalls gehen Sie wie folgt vor:

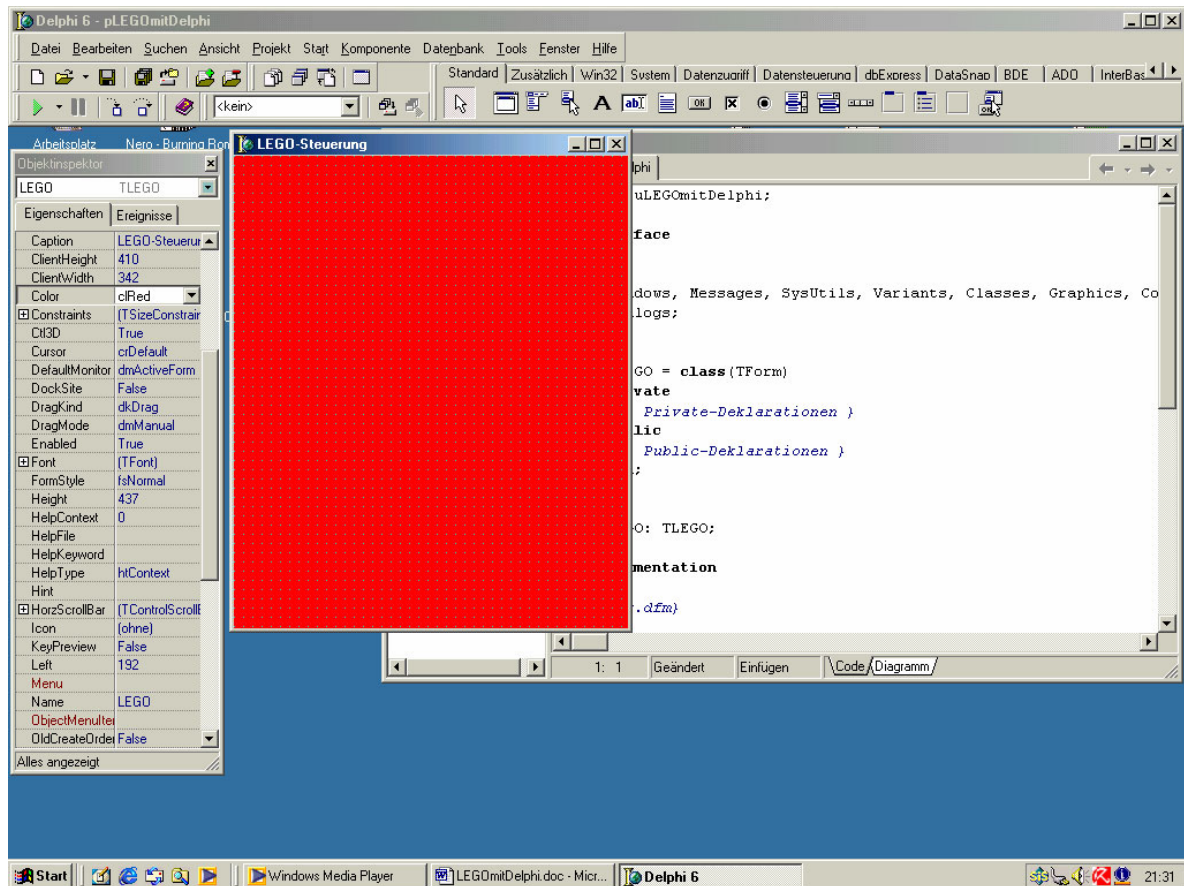
- Klicken Sie auf die Schaltfläche Start und dann auf Ausführen.
- Geben Sie im Feld Öffnen folgendes ein: Regsvr32.exe <Steuerelement-Pfad und -Dateiname>, wobei <Steuerelement-Pfad und -Dateiname> für den Pfad und den Dateinamen des Steuerelements stehen. Also in unserem Fall:
Regsvr32.exe c:\progra~1\LEGOMI~1\system\spirit.ocx
- Klicken Sie auf OK.
- Falls eine Fehlermeldung ausgegeben wird, dass die Registrierung fehlgeschlagen ist, stellen Sie sicher, dass der Pfad zum Steuerelement korrekt ist und die Datei existiert.

b: Erstellen einer neuen Delphi-Anwendung

Jetzt wird eine neue Delphi-Anwendung erstellt. Starten Sie dazu Delphi. Dabei wird automatisch ein neues Projekt eröffnet. Speichern Sie die Projektdateien am besten gleich in einem eigenen Verzeichnis, indem Sie die Unit mit uLEGO.pas und die Projektdatei mit pLEGO.dpr bezeichnen. Wählen Sie dazu im Datei-Menü den Punkt Alles speichern:

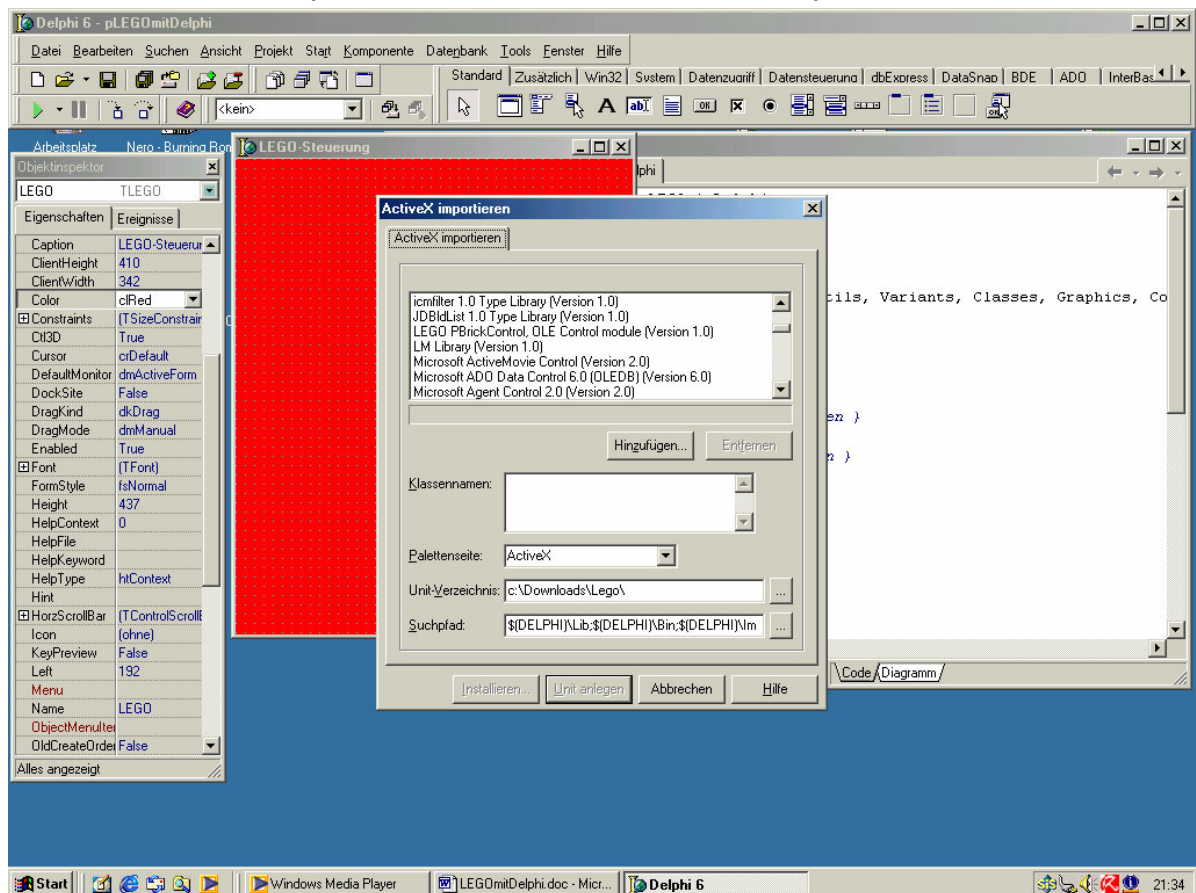


Danach steht Ihnen die übliche Delphi-Entwicklungsumgebung mit Werkzeugleiste, Objektinspektor, Editorfenster usw. zur Verfügung. Ändern Sie im Objektinspektor den Namen des Formulars auf LEGO und die Caption auf LEGO-Steuerung.



c: Einfügen des ActiveX-Steuerelements in die Werkzeugleiste

- Wählen Sie im Komponente-Menü den Punkt ActiveX importieren.



- Wählen Sie aus der Liste der ActiveX-Steuerelemente das LEGO PBrick Control OLE Control Modul aus.
- Klicken Sie auf Hinzufügen.
- Wählen Sie die zu installierende Bibliothek SPIRIT.OCX aus, die sich z. B. im Verzeichnis \Programme\LEGO Mindstorms\system befindet.
- Klicken Sie auf Installieren.

Danach finden Sie das Steuerelement in der Werkzeugleiste unter dem Reiter ActiveX.

Arbeiten mit der PHANTOM-Komponente

Im Unterschied zur PBrick-Komponente arbeitet die Phantom-Bibliothek auch mit dem USB-Tower zusammen. Sie ersetzt in ihrer Funktionalität die PBRICK vollständig. Zu bekommen ist die freie Phantom-Bibliothek unter

<http://www.geocities.com/pbrickdev/downloadphantom.html>

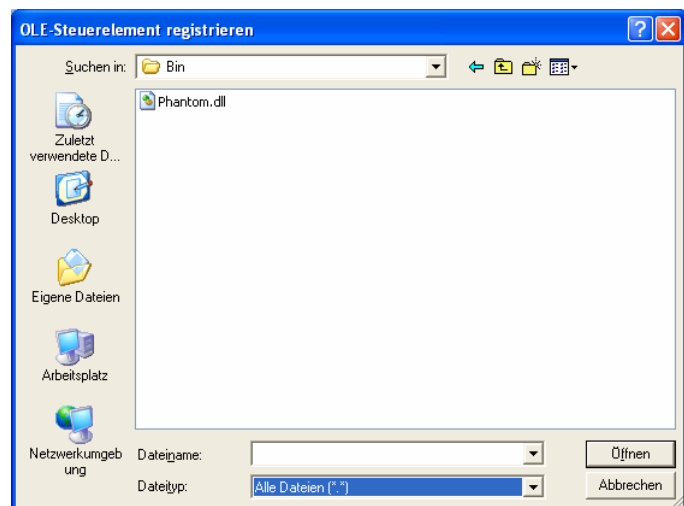
Nach der Installation befindet sich die benötigte Bibliothek z.B. im Verzeichnis \Programme\Phantom/bin unter dem Namen Phantom.dll.

Einbinden der ActiveX-Komponenten unter Delphi

Wir wählen diesmal den direkten Weg unter Delphi:

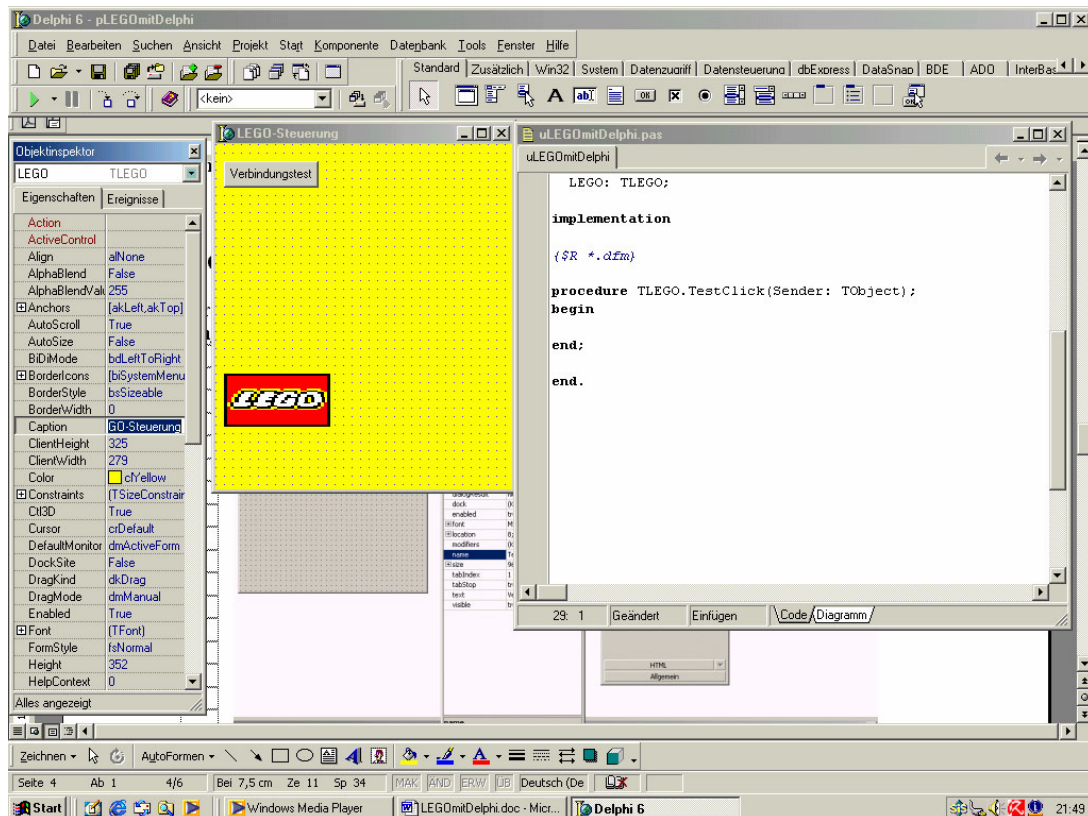
- Starten Sie Delphi. Es wird eine neue Delphi-Anwendung erstellt. Speichern Sie die Projektdateien am besten gleich in einem eigenen Verzeichnis, indem Sie die Unit mit uLEGO.pas und die Projektdatei mit pLEGO.dpr bezeichnen. Wählen Sie dazu im Datei-Menü den Punkt Alles speichern. Danach steht Ihnen die übliche Delphi-Entwicklungsumgebung mit Werkzeugleisten, Objektinspektor, Editorfenster usw. zur Verfügung. Ändern Sie im Objektinspektor den Namen des Formulars auf LEGO und die Caption auf LEGO-Steuerung.
- Wählen Sie im Komponente-Menü den Punkt ActiveX importieren. Wählen Sie dort Hinzufügen und navigieren Sie zum Verzeichnis mit der Phantom.dll. Um die zu finden, stellen Sie unter Dateityp „alle Dateien (*.*)“ ein.
- Klicken Sie auf Hinzufügen.
- Wählen Sie die zu installierende Bibliothek Fenestra Software Phantom ... aus. Klicken Sie auf Installieren.

Danach finden Sie das Steuerelement in der Werkzeugleiste unter dem Reiter ActiveX.



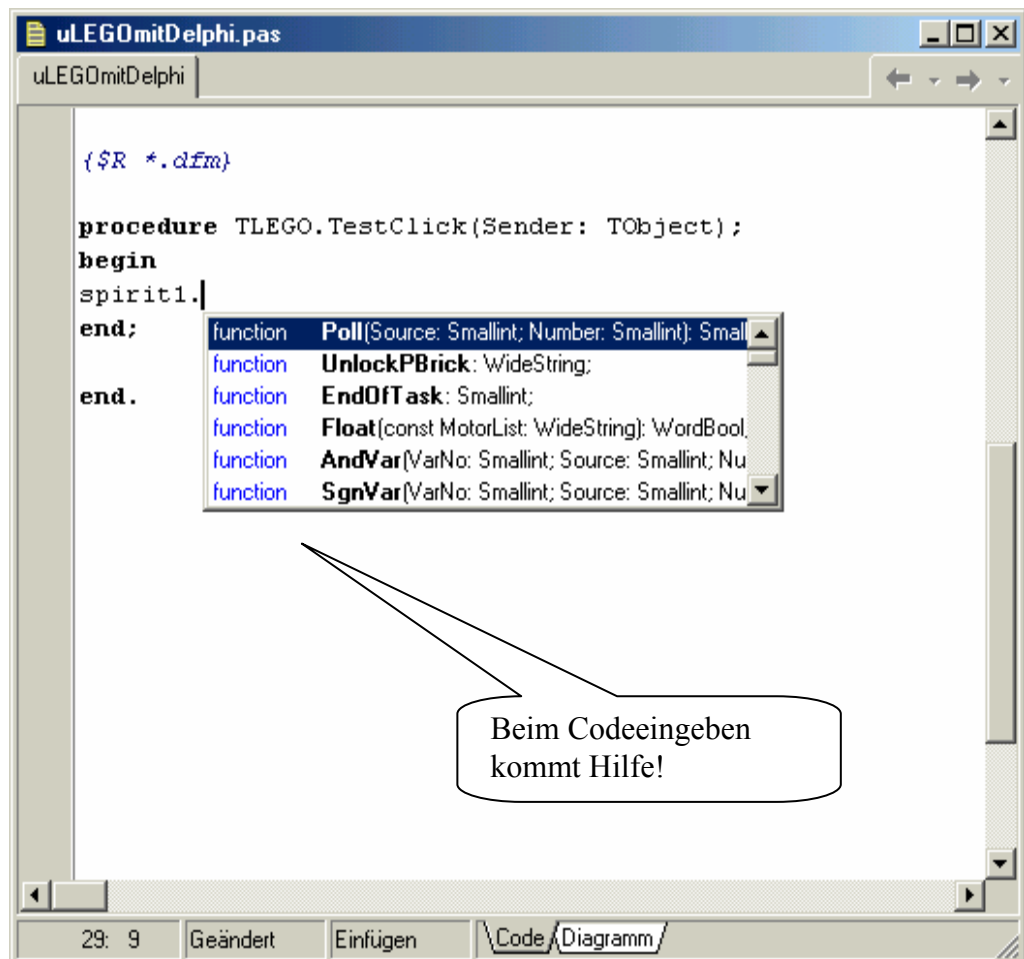
Steuern des Roboters unter Delphi

Wir fügen die Spirit- oder Phantom-Komponente unter dem Namen Spirit1 und einen weiteren Knopf ein, mit dessen Hilfe die Verbindung zum Roboter (RCX) getestet wird:



Doppelklicken wir auf den Knopf, den wir **Test** nennen, dann wird ein entsprechender Event-Handler **TLEGO.TestClick** automatisch erzeugt. In diesen können wir Code eingeben.

Weiter oben im Quelltext finden wir den Namen der eingebundenen Spirit-Komponente: Sie heißt **spirit1**. Geben wir diesen Namen und danach einen Punkt ein, dann erscheinen alle Methoden und Felder des Objekts.



Für den Leitungstest müssen wir zuerst die serielle Schnittstelle initialisieren:

```
spirit1.InitComm;
```

Danach prüfen wir die Verbindung mit `spirit1.PBAliveOrNot` und geben im Erfolgsfall eine Tonfolge am RCX mit `spirit1.PlaySystemSound(2)` aus:

```
procedure TLEGO.TestClick(Sender: TObject);
begin
  //bei Phantom: zusätzlich Spirit1.ComPortNo := 1;
  spirit1.InitComm;
  spirit1.PlaySystemSound(2);
end;
```

Jetzt ergänzen wir das Formular um eine Label-Komponente, die den Zustand der Verbindung zum Roboter anzeigt, sowie eine Steuerknöpfe, um den Roboter direkt anzusprechen. An Befehlen stehen dafür zur Verfügung: (wobei `<Motorliste>` eine Zeichenkette ist, die die Motornummern ("0" bis "2") enthält, z. B. "02")

<code>spirit1.On(<Motorliste>);</code>	startet die angegebenen Motoren
<code>spirit1.Off(<Motorliste>);</code>	stoppt die angegebenen Motoren.
<code>spirit1.AlterDir(<Motorliste>);</code>	ändert die Bewegungsrichtung der angegebenen Motoren.

```
procedure TLEGO.TestClick
  (Sender: TObject);
begin
  if spirit1.PBAliveOrNot
  then begin
    spirit1.PlaySystemSound(2);
    Label1.Caption := 'Verbindung: ok!'
  end
  else Label1.Caption :=
    'Verbindung: FEHLER!'
end;
```

```
procedure TLEGO.FormCreate
  (Sender: TObject);
begin
  spirit1.InitComm;
  if spirit1.PBAliveOrNot
  then begin
    spirit1.PlaySystemSound(2);
    Label1.Caption := 'Verbindung: ok!'
  end
  else Label1.Caption := 'Verbindung: FEHLER!'
end;
```



```
procedure TLEGO.vorClick(Sender: TObject);
begin spirit1.On_('02') end;
```

Beide Motoren „an“.

```
procedure TLEGO.zurueckClick(Sender: TObject);
begin spirit1.On_('02'); spirit1.AlterDir('02') end;
```

Beide Motoren „an“ und dann die Richtung umkehren.

```
procedure TLEGO.linksClick(Sender: TObject);  
begin spirit1.On_('0'); spirit1.Off('2') end;
```

rechter Motor „aus“, linker „an“.

```
procedure TLEGO.rechtsClick(Sender: TObject);  
begin spirit1.On_('2'); spirit1.Off('0') end;
```

linker Motor „aus“, rechter „an“.

```
procedure TLEGO.stopClick(Sender: TObject);  
begin spirit1.Off('02') end;
```

Beide Motoren „aus“.

```
procedure TLEGO.EndeClick(Sender: TObject);  
begin  
spirit1.Off('02');  
spirit1.CloseComm;  
halt  
end;
```

Motoren abschalten, Schnittstelle freigeben und Programm beenden.