

M2 Lego-Roboter im Informatik-Unterricht der Sekundarstufe I



RCX-Programmierung mit Delphi

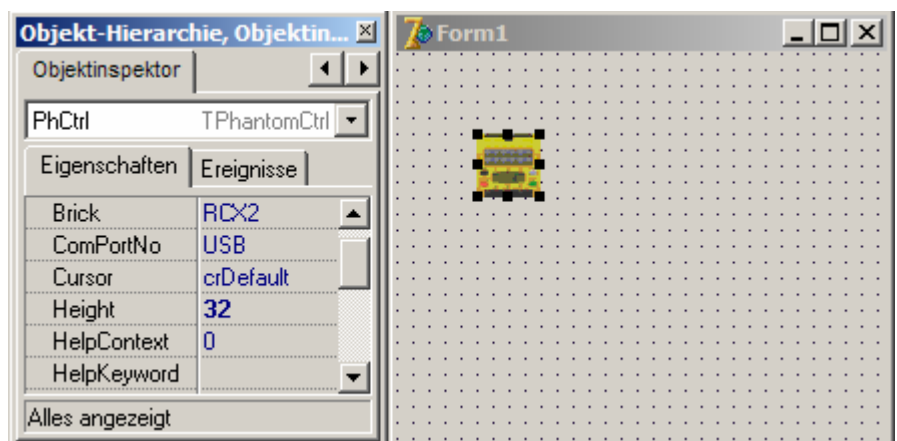
Neben Robolab und dem Robotics Invention System gibt es eine Vielzahl von Programmierungsumgebungen für Lego Mindstorms Roboter, z. B. NQC oder LEJOS. Des Weiteren gibt es ActiveX-Elemente, mit denen die Lego-Roboter angesprochen werden können, so etwa die PBrick-Komponente Spirit.OCX, die sowohl eine Programmierung unter Visual Basic bzw. Microsoft Visual Studio als auch unter Delphi ermöglicht. Diese Komponente arbeitet aber nur mit der seriellen Schnittstelle zusammen, die bei neueren Computern kaum noch anzutreffen ist. Daher soll im Folgenden mit der *Phantom*-Komponente gearbeitet werden, da diese auch auf die USB-Schnittstelle zugreifen kann; im Übrigen ist die Funktionalität beider Komponenten gleich.

Installation der Phantom-Komponente:

Für die Installation sind i. a. Administrator-Rechte notwendig. Kopieren Sie zunächst die Phantom.dll in ein Delphi-Verzeichnis, etwa *C:\Programme\Borland\Delphi\Ocx\Lego*. Starten Sie Delphi. Wählen Sie im *Komponente-Menü* den Punkt *ActiveX importieren*. Wählen Sie dort *Hinzufügen* und navigieren Sie zum Verzeichnis, in das Sie die Phantom.dll kopiert haben. Klicken Sie auf dann *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*. (Installationsanleitung nach [1])

Ein erstes Programm:

Dazu legen wir zunächst ein neues Delphi-Projekt an und speichern das leere Projekt unter dem Namen *RCX1* ab. Danach wird die Phantom-Komponente auf das Formular gezogen und die Einstellungen kontrolliert. Auf Druck eines Buttons soll nur ein Verbindungstest durchgeführt werden: Bei Erfolg spielt der RCX eine kleine, vorgegebene Melodie. Dazu wird ein Button erzeugt mit der zugehörigen Button-Click-Methode:



```
procedure TForm1.Button1Click(Sender: TObject);
begin
  PhCtrl.InitComm; //Schnittstelle initialisieren
  PhCtrl.PlaySystemSound(3); // Sound abspielen
end;
```

M2 Lego-Roboter im Informatik-Unterricht der Sekundarstufe I



Bei der Programmierung mit Delphi gibt es - wie bei den anderen Programmierumgebungen auch – zwei Möglichkeiten:

- Der RCX-Baustein kann vom PC aus „ferngesteuert“ werden. Dazu überträgt ein Delphi-Programm ständig über eine bestehende Infrarotverbindung Befehle auf den RCX, die dieser sofort ausführt.
- Mittels eines Delphi-Programms wird ein RCX-Programm auf den Roboter übertragen, das dann manuell oder vom PC aus aufgerufen werden kann.

„Fernsteuerung“ des RCX-Bausteins:

Schon beim Start des Delphi-Programms wird geprüft, ob sich ein RCX in der Nähe befindet. Ist das der Fall, so lässt das Programm den RCX eine Melodie spielen und es erscheint im Programmfenster eine Erfolgsmeldung. Außerdem werden noch einige Einstellungen (siehe auch unten) vorgenommen:

Mit dem Befehl *SetSensorType* werden die Sensoren an den Ausgängen festgelegt. Der erste Parameter gibt den Ausgang (von 0 bis 2) an, der zweite den Sensortyp:

- 1: Tastsensor
- 2: Temperatursensor
- 3: Lichtsensor
- 4: Rotationssensor

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  Zaehler := false;
  hz := 0;
  with PhCtrl do begin
    InitComm;
    if PBAliveOrNot
    then begin
      PlaySystemSound(1);
      Label1.caption := 'Verbindung: OK';
      SetSensorType(0, 1);
      SetSensorType(1, 3);
      SetSensorType(2, 4);
    end
    else Label1.caption := 'Verbindung: Fehler!';
  end;
end;
```

Kommt keine Verbindung zustande, so gibt das Delphi-Programm eine entsprechende Fehlermeldung aus.

Bei Beendigung des Programms können mit *PhCtrl.Off('012')* alle Motoren angehalten und mit *PhCtrl.CloseComm* die Verbindung geschlossen werden.

Das Abfragen der Sensoren erfolgt mit dem Befehl *PhCtrl.Poll(SRC_SENVAL, Source)*. Die Darstellung erfolgt mit Standard-Delphi-Komponenten.

```
procedure TForm1.Timer2Timer(Sender: TObject);
var sens1, sens2, sens3: integer;
begin
  sens1 := PhCtrl.Poll(SRC_SENVAL, 0);
  sens2 := PhCtrl.Poll(SRC_SENVAL, 1);
  sens3 := PhCtrl.Poll(SRC_SENVAL, 2);
  Label3.caption := IntToStr(sens1);
  Label3.refresh;
  Label4.caption := IntToStr(sens2);
  Label4.refresh;
  Label5.caption := IntToStr(sens3);
  Label5.refresh;
end;
```

M2 Lego-Roboter im Informatik-Unterricht der Sekundarstufe I



RCX-Programme übertragen und starten:

Mit Hilfe der Prozedur rechts wird ein Programm auf dem Rcx-Baustein gespeichert. Zunächst wird mit *selectprgm()* der Programmplatz ausgewählt. *BeginOfTask()* und *EndOfTask()* markieren Anfang und Ende des zu speichernden Programms (es können mehrere Tasks auf einem Programmplatz abgelegt werden.) die Bedeutung der anderen Befehle für die Motorsteuerung ist der Tabelle unten zu entnehmen.

Das Starten eines Programms vom PC aus mit Hilfe von Delphi erfolgt dann mittels der Befehle *selectprgm(3)* und *starttask(0)*, siehe auch das Beispielprogramm).

```
procedure TForm1.BTaskLoadClick(Sender:
TObject);
begin
  with PhCtrl do
    begin
      selectprgm(3);
      beginoftask(0);
      setfwd('0');
      on_('0');
      wait(2,150);
      off('0');
      setrwd('0');
      on_('0');
      wait(2,100);
      off('0');
      endoftask;
    end;
end;
```

Vorschläge für Programmierübungen:

- Probieren Sie die „Fernsteuerung“ für den RCX aus, in dem Sie ihn kurze Befehlssequenzen ausführen lassen.
- Fragen Sie die Sensoren ab. Bauen Sie – falls vorhanden – einen Rotationssensor ein und fragen Sie seine Daten ab. Was geschieht, wenn man einen Sensoreingang falsch initialisiert?
- Überlegen Sie sich eine Aufgabe, die der Roboter erledigen soll, etwa: geradeaus fahren bis er auf ein Hindernis trifft. Dann soll er wieder ein Stück zurückfahren.
- Experimentieren Sie mit Schleifen, die Befehle dazu finden Sie in der Tabelle unten. Lassen Sie den Roboter eine Sequenz mehrfach wiederholen. Es kann eine Schleife mit vorgegebener Anzahl an Wiederholungen sein oder auch eine Schleife mit einer Abbruchbedingung, die durch einen Sensor gesteuert wird.

Literatur:

- [1]: Eckart Modrow: Lego Mindstorms-Roboter im Informatik-Unterricht – mit Delphi (<http://www.vlin.de/sek1/LEGOMitDelphi.pdf>)
- [2]: Dr. Jürgen Poloczec: Lego Mindstorms Steuerung mit Delphi (Hess. Bildungsserver: <http://www.bildung.hessen.de/abereich/inform/skii/material/delphi/lego/index.htm>)

M2 Lego-Roboter im Informatik-Unterricht der Sekundarstufe I



Übersicht der RCX-Befehle:

Befehl	Bedeutung
Schnittstelle ein- und ausschalten:	
PhCtrl.InitComm	Öffne die USB Schnittstelle
PhCtrl.CloseComm	schließe die USB Schnittstelle
Motoren ein- und ausschalten:	
"Motor0"	Motor an Ausgang A
"Motor1"	Motor an Ausgang B
"Motor2"	Motor an Ausgang C
"Motor0Motor2"	Motor an Ausgang A und B
PhCtrl.SetFwd "Motor0"	schalte MotorA in den Vorwärtsgang
PhCtrl.On "Motor0"	schalte MotorA ein
PhCtrl.Off "Motor0"	schalte MotorA aus
PhCtrl.SetPower "Motor0",SRC_CON,1	Stärke des Motors A = 7 (möglich:1-7)
PhCtrl.AlterDir "Motor0"	schalte MotorA in die entgegen gesetzte Richtung
Sensoren:	
0 = Eingang 1 1 = Eingang 2 2 = Eingang 3	1 = Tastsensor 2 = Temperatursensor 3 = Lichtsensor 4 = Rotationssensor
PhCtrl. SetSensorType 0, 3	Sensor definieren: Eingang 1, Lichtsensor
PhCtrl. Poll (SRC_SENVAL, 0)	Wert abfragen: sende Wert, Eingang 1
Befehle um ein Programm auf dem RCX zu speichern:	
PhCtrl.SelectPrg 4	Speichere das folgende Programm auf Speicherplatz 5 des RCX
PhCtrl.BeginOfTask main	Beginn des Programms, welches gespeichert werden soll
PhCtrl.EndOfTask	Ende des Programms, welches gespeichert werden soll
Sound spielen:	
PhCtrl.PlaySystemSound 3	Sound Nr. 3 abspielen
Variablen einen Wert zuweisen:	
PhCtrl.SetVar 6, SRC_CON, 1	Variable Nr. 6 erhält den Wert 1

M2 Lego-Roboter im Informatik-Unterricht der Sekundarstufe I



Befehl	Bedeutung
Schleifen:	
Zählschleife:	
PhCtrl.Loop SRC_CON, 4 PhCtrl.EndLoop	wiederhole 4 x die Befehle in der Schleife
Warteschleife:	
PhCtrl.Wait SRC_CON, 200	Warte zwei Sekunden.
Schleife mit Bedingung am Anfang:	
EQ = equal = GT = greater than > LT = less than < NE = not equal to <>	
PhCtrl.While SRC_SENVAL, 0, EQ, SRC_VAR, 6 PhCtrl.EndWhile	Wiederhole solange die Ausgabe von Eingang 1 gleich dem Wert der Variablen 6 ist.
Fallunterscheidung:	
PhCtrl.If SRC_SENVAL, 0, EQ, SRC_VAR, 6 PhCtrl.On "Motor0" PhCtrl.Else PhCtrl.On "Motor2" PhCtrl.EndIf	wenn Eingang1 = Variable5 dann MotorA an sonst MotorB an Ende der Fallunterscheidung