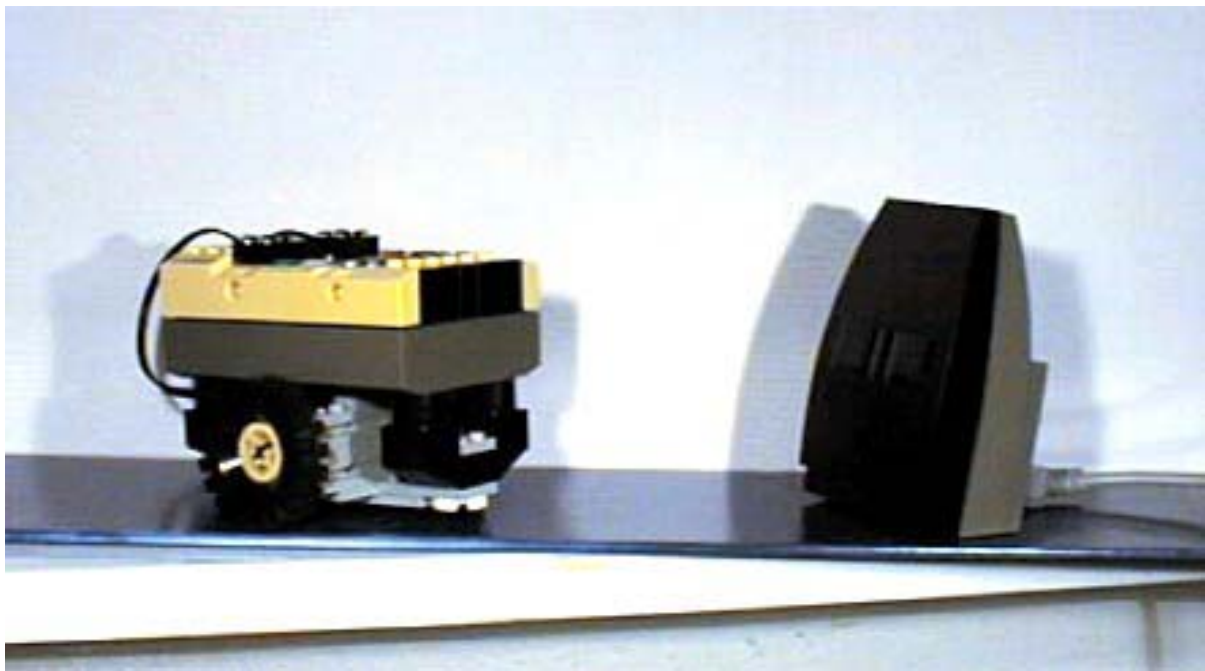


LEGO-Mindstorms-Roboter im Informatikunterricht¹ - mit Java-



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 einer 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 Java

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 beigelegten Materialien. Als Beispiele wollen wir die Software unter Microsoft Visual J++ benutzen.

¹ gefördert von der Initiative N21

Einbinden der ActiveX-Komponenten unter Java

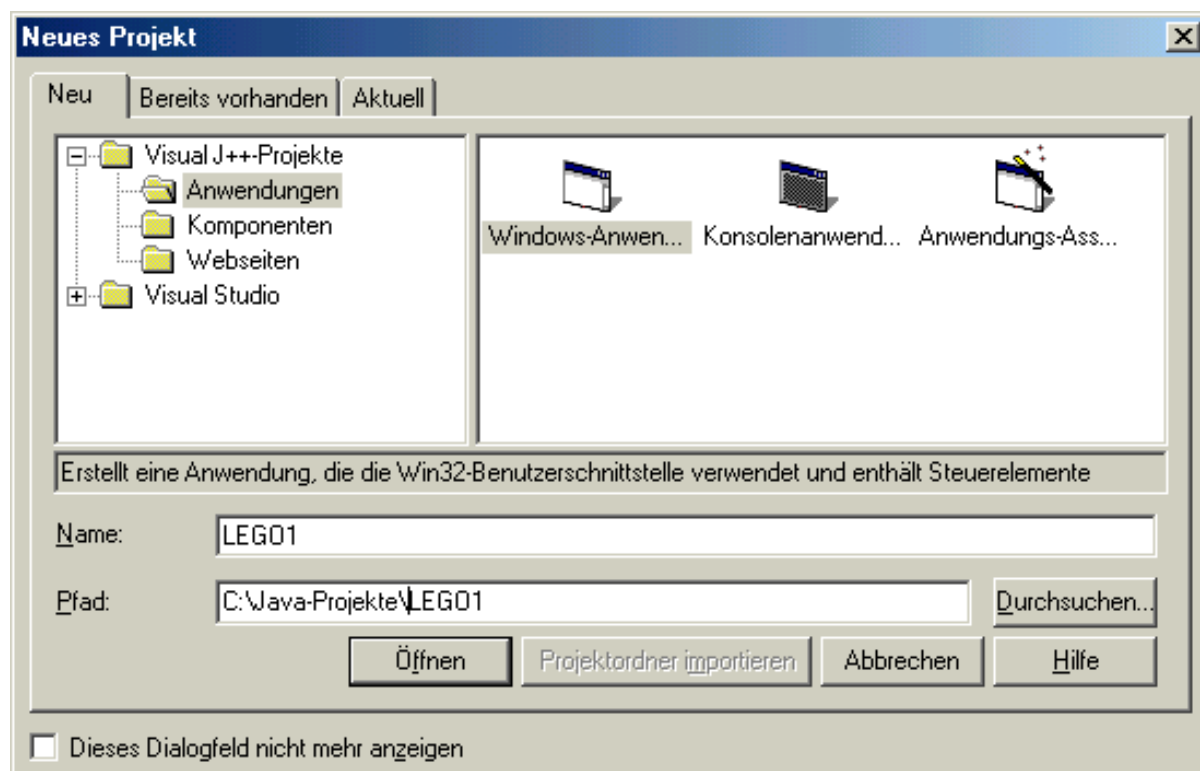
a: Registrieren von Steuerelementen

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.

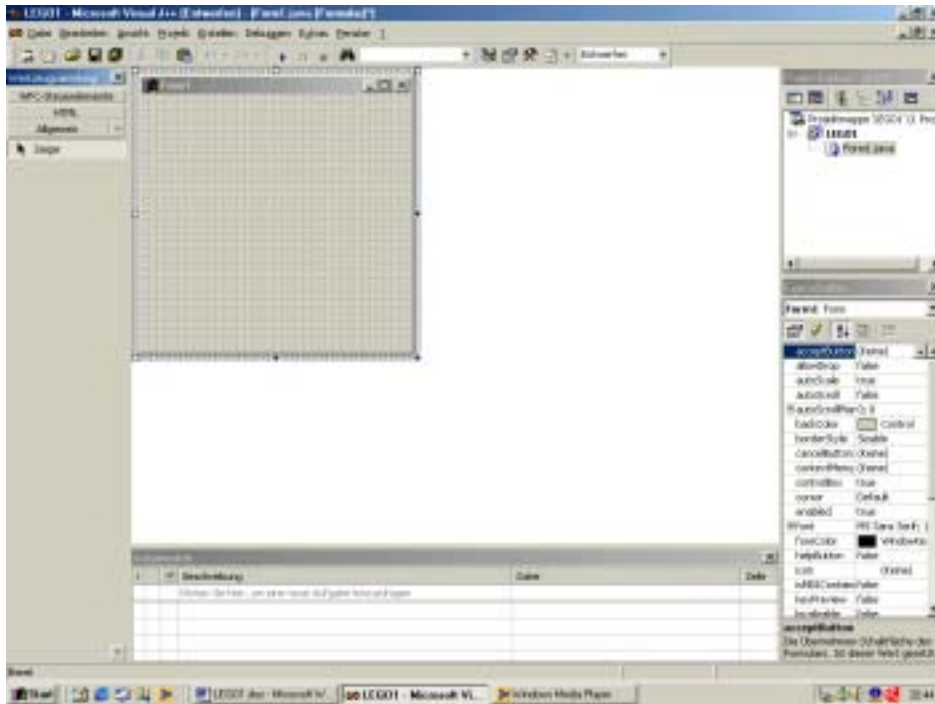
b: Erstellen einer neuen Java-Anwendung

Jetzt wird eine neue Java-Anwendung erstellt. Starten Sie dazu VisualJ++ und wählen Sie im Datei-Menü den Punkt Neues Projekt:



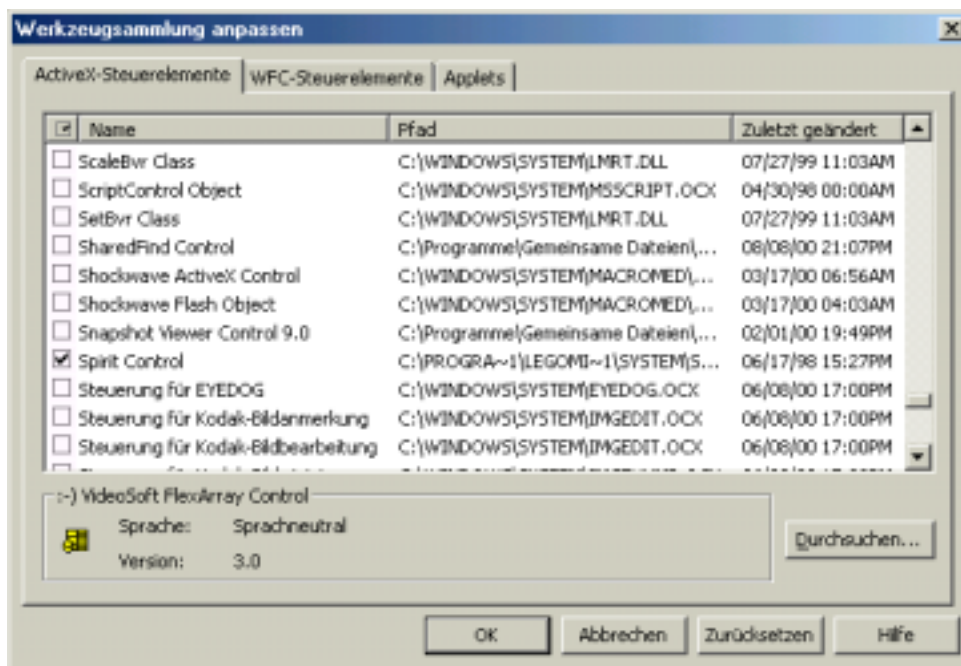
Wählen Sie Windows-Anwendung aus und vergeben Sie einen Namen. Danach klicken Sie auf Öffnen. Vergrößern Sie den Projektknoten („+“ vor dem Projektnamen rechts im Fenster), so dass die Java-Datei Form1.java erscheint. Doppelklicken Sie auf diesen Namen. Es erscheint ein Fenster mit dem Formular designer, mit dessen Hilfe Sie die Oberfläche des Programms erstellen können.

In der Werkzeugsammlung befinden sich jetzt die üblichen Windows-Kontrollelemente wie Buttons, Editierfelder, Label etc. Es fehlt noch das benötigte LEGO-Mindstorms-Steuerelement.

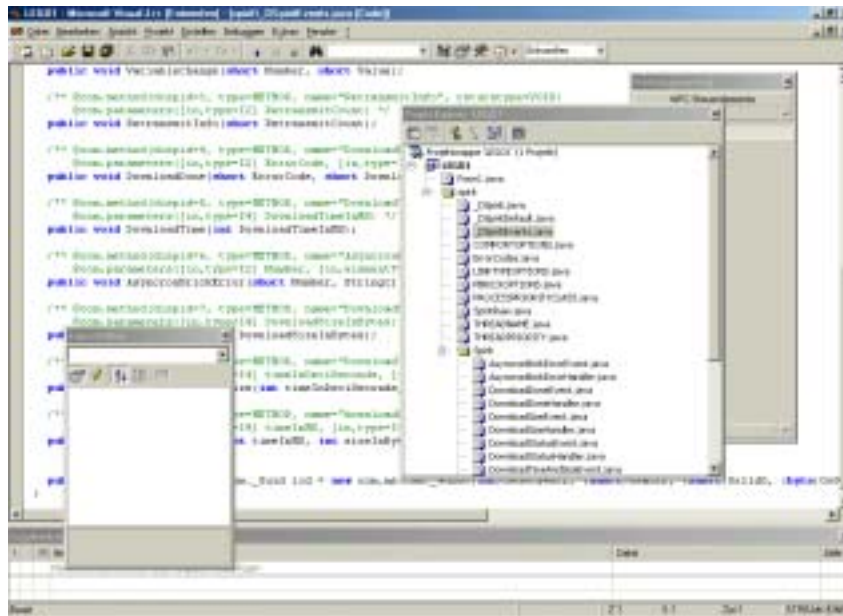


c: Einfügen des ActiveX-Steuererelementes in die Werkzeugleiste

- Klicken Sie in der Werkzeugsammlung auf die Registerkarte Allgemein. (Falls die Werkzeugsammlung nicht angezeigt wird, klicken Sie im Menü Ansicht auf Werkzeugsammlung.)
- Klicken Sie mit der rechten Maustaste auf die Werkzeugsammlung und klicken Sie dann auf Werkzeugsammlung anpassen.
- Klicken Sie im Dialogfeld Werkzeugsammlung anpassen auf die Registerkarte ActiveX-Steuererelemente.
- Wählen Sie aus der Liste der ActiveX-Steuererelemente das Spirit-Steuererelement aus.
- Klicken Sie auf OK.

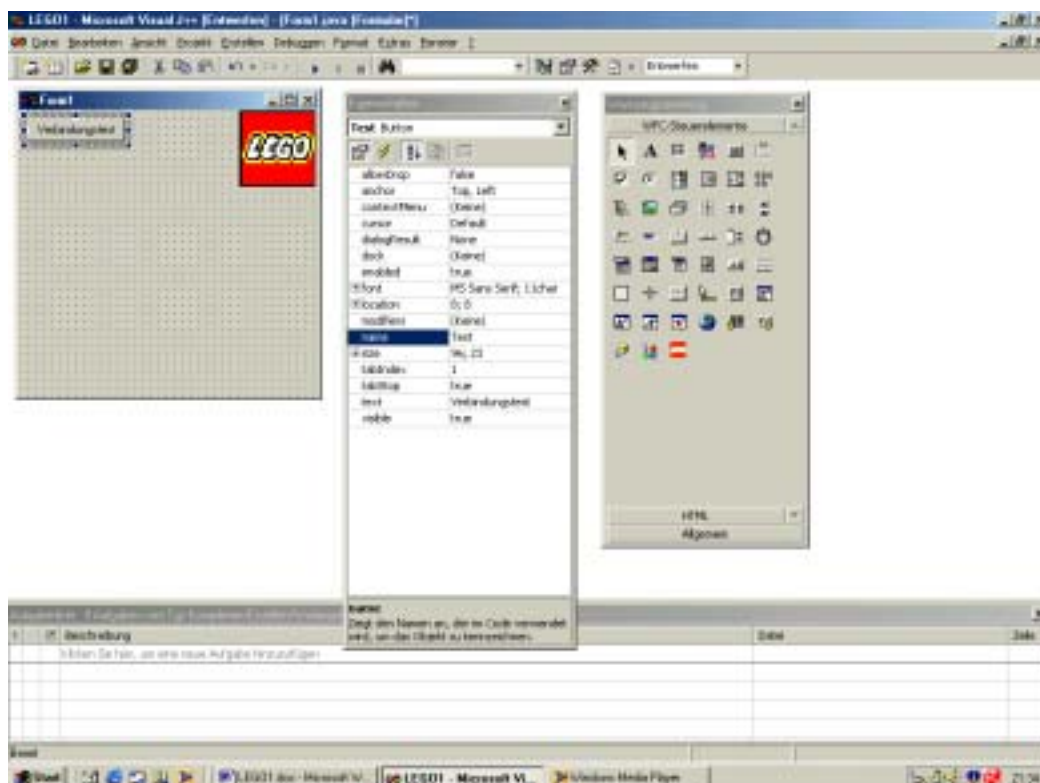


In der Werkzeugsammlung finden Sie auf der Registrierkarte WFC-Steuerelemente jetzt das LEGO-Symbol. Unter Java stehen Wrapper-Klassen und Interfaces bereit, die die Funktionalität des Steuerelements kapseln. Bei Bedarf finden wir im Projektexplorer im Ordner Spirit alle zur Verfügung stehenden Methoden.



Steuern des Roboters unter Java

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




```
private void Test_click
    (Object source, Event e)
{
    spirit1.InitComm();
    if(spirit1.PBAliveOrNot())
    {
        spirit1.PlaySystemSound((short)2);
        label1.setText("Verbindung: ok!");
    }
    else
    {
        label1.setText
            ("Verbindung: FEHLER!");
    }
}
```



```
private void ende_click(Object source, Event e)
{
    spirit1.Off("02");
    spirit1.CloseComm();
    Application.exit();
}
```

Motoren abschalten, Schnittstelle freigeben und Programm beenden.

```
private void vor_click(Object source, Event e)
{
    spirit1.On("02");
}
```

Beide Motoren „an“.

```
private void stopp_click(Object source, Event e)
{
    spirit1.Off("02");
}
```

Beide Motoren „aus“.

```
private void zurueck_click(Object source, Event e)
{
    spirit1.On("02"); spirit1.AlterDir("02");
}
```

Beide Motoren „an“ und dann die Richtung umkehren.

```
private void links_click(Object source, Event e)
{
    spirit1.On("0"); spirit1.Off("2");
}
```

rechter Motor „aus“, linker „an“.

```
private void rechts_click(Object source, Event e)
{
    spirit1.On("2"); spirit1.Off("0");
}
```

linker Motor „aus“, rechter „an“.