

Dipl.-Chem.  
Thomas Meinike  
W. v. Klewiz-Str. 3  
D-06132 Halle

02. 02. 1995

ANGULAR.DLL - Bibliothek der Winkelfunktionen - Registrierte Vollversion  
-----

Beim Programmieren mit Microsoft Visual Basic 3.0 stellte ich fest, daß von den Winkelfunktionen nur ATN, COS, SIN und TAN implementiert sind. Es ist aber möglich mit diesen und einigen Zusatzformeln die fehlenden Funktionen nachzubilden. Ich beschränke jedoch einen anderen Weg und erinnere mich an FORTRAN, die Sprache der Mathematiker und Naturwissenschaftler. Die Vollversion enthält 24 in FORTRAN geschriebene Winkelfunktionen. Der Code ist für die Nutzung des mathematischen Coprozessors kompiliert worden und wird bei Abwesenheit desselben durch die Windows-interne Emulation interpretiert.

Die DLL-Funktionen sind jeweils als Single- und Double-Version ausgeführt. Dazu sind sie an der ersten Stelle des Funktionsnamens mit 'S' bzw. 'D' gekennzeichnet. Die folgende Liste enthält alle Funktionen geordnet nach Klassen und ohne die Angabe des ersten Buchstabens.

Trigonometrische Funktionen  
-----

Sinus()  
Cosinus()  
Tangens()  
Cotangens()  
Secans()  
Cosecans()

Inverse Trigonometrische Funktionen  
-----

ArcSinus()  
ArcCosinus()  
ArcTangens()  
ArcCotangens()  
ArcSecans()  
ArcCosecans()

Hyperbolische Funktionen  
-----

HypSinus()  
HypCosinus()  
HypTangens()  
HypCotangens()  
HypSecans()  
HypCosecans()

Inverse Hyperbolische Funktionen  
-----

ArcHypSinus()  
ArcHypCosinus()  
ArcHypTangens()  
ArcHypCotangens()  
ArcHypSecans()  
ArcHypCosecans()

In der oben bezeichneten Nomenklatur existiert also beispielsweise die

Funktion 'Sinus Hyperbolicus' als:

DHypSinus() und SHypSinus()

Die Übergabeparameter beim Aufruf der Funktion sind 'Argument' und 'Result', wiederum in Single- bzw. Double-Notation, z.B.

DHypSinus(DArgument, DResult) und SHypSinus(SArgument, SResult)

In dieser Form ist bei allen enthaltenen Funktionen vorzugehen.

Anwendung der DLL-Funktionen in eigenen Visual Basic-Programmen

-----  
Das Prinzip wird wiederum an der HypSinus-Funktion verdeutlicht.

Zunächst müssen die verwendeten Funktionen aus der DLL im Deklarationsteil des VB-Projektes vereinbart werden:

```
Declare Sub DHypSinus Lib "Angular.dll" (DArgument As Double, DResult As Double)
Declare Sub SHypSinus Lib "Angular.dll" (SArgument As Single, SResult As Single)
```

In der Praxis ist es einfacher alle DLL-Funktionen in der Datei 'GLOBAL.BAS' zu deklarieren und zum Projekt hinzuzuladen. Diese Datei ist in der Shareware- bzw. Vollversion mit den jeweils unterstützten Funktionen enthalten.

Bei den nachfolgenden 3 Beispielen befinden sich auf der VB-Form 2 Textfelder (Text1 und Text2) für Ein- und Ausgaben sowie ein Button (Command1) für das Ausführen der Berechnung.

Beispiel 1:

-----

```
Sub Command1_Click ()

    Dim DArgument As Double
    Dim DResult As Double

    DArgument = Val(Text1.Text)
    Call DHypSinus(DArgument, DResult)
    Text2.Text = Str$(DResult)

End Sub
```

Beispiel 2:

-----

```
Sub Command1_Click ()

    Dim DArgument As Double
    Dim DResult As Double

    DArgument = Val(Text1.Text)
    If DArgument = 0 Then Text2.Text = " nicht def.": Exit Sub
    Call DCotangens(DArgument, DResult)
    Text2.Text = Str$(DResult)

End Sub
```

Beispiel 3:

-----

```
Sub Command1_Click ()

    Dim SArgument As Single
    Dim SResult As Single
```

```

SArgument = Val(Text1.Text)
If Abs(SArgument) > 1 Then Text2.Text = " nicht def.": Exit Sub
Call SArcCosinus(SArgument, SResult)
Text2.Text = Str$(SResult)

End Sub

```

In den Beispielen 2 und 3 wird vor dem Aufrufen der DLL-Winkelfunktion zunächst eine Prüfung des Argumentes bezüglich des Definitionsbereiches der Funktion vorgenommen. Diese Aufgabe muß vom Programmierer übernommen werden, da in der DLL keine Fehlerbehandlung enthalten ist. Dieses Vorgehen wurde gewählt, da die aufgerufene Funktion einen Rückgabewert besitzt und dieser nicht beliebig gesetzt werden kann (würde man die Result-Variable bei einem Fehler z.B. auf Null setzen, so stellt sich die Frage wie der Anwender der Funktion auswertet ob Result=0 einen Fehler oder einen wirklichen Funktionswert darstellt). Um den Code der DLL und die Aufrufkonventionen für den VB-Programmierer nicht zu kompliziert zu gestalten, wurde dieser Kompromiß gewählt. In den obigen Beispielen wird beim Nichtgenügen des Argumentes die Prozedur ohne Aufruf der DLL-Funktion verlassen. Der Anwender der DLL möge sich in den Tabellenwerken der mathematischen Funktionen über den Definitionsbereich der gewünschten Funktionen informieren, da das den Rahmen an dieser Stelle sprengen würde. Wie im Beispiel 1 zu sehen ist, sind einige Funktionen auch für alle reellen Argumente gültig. Wichtiger Hinweis: alle Winkelfunktionen der DLL erwarten die Argumente im Bogenmaß!

#### Beispielprogramm FULLDEMO.EXE

Das mitgelieferte Programm FULLDEMO.EXE wurde mit VB3 geschrieben und demonstriert alle Funktionen in der Vollversion der DLL. Für ein eingegebenes Argument können alle 24 Funktionen in Single- bzw. Double-Ausführung berechnet werden. Bei ungeeigneten Argumenten wird wie oben beschrieben verfahren und 'nicht definiert' als Funktionsergebnis gemeldet.

#### Copyright und Nutzungsrecht

Die registrierte DLL darf vom Käufer in eigenen Programmen verwendet und mit diesen ohne Beschränkung weitergegeben werden. Es fallen keine zusätzlichen Lizenzgebühren an. Erwirbt ein Anwender ein Programm mit integrierter DLL, so darf er diese nur im Zusammenhang mit dieser Software nutzen. Das Copyright der 'ANGULAR.DLL' bleibt beim Autor. Der mitgelieferte FORTRAN-Code darf von registrierten Benutzern verändert oder erweitert werden, wobei der Ursprungs-Copyright-Vermerk in der Software / Dokumentation einzutragen ist.

#### Anwendungshinweise

Für die Anwendung der ANGULAR.DLL wird seitens des Autors keine Haftung in Bezug auf Schäden an Hardware oder Software übernommen. Die DLL funktioniert mit Visual Basic 3.0 in der beschriebenen Weise. Es wird darauf hingewiesen den Funktionsaufruf auf Zulässigkeit in Bezug auf das Argument zu prüfen und die VB-Projekte in der Testphase stets vor dem Start in der Entwicklungsumgebung abzuspeichern. Ein fehlerhafter Aufruf beendet Visual Basic sofort! Das Beispielprogramm FULLDEMO.EXE wurde unter Einhaltung dieser Regeln entwickelt und funktioniert ohne bekannte Probleme. Die DLL kann prinzipiell auch mit anderen DLL-fähigen Windows-Entwicklungssystemen verwendet werden. Dafür liegen dem Autor bisher keine Anwendermitteilungen vor.

#### Dateien der Programmdiskette

01 ANGULAR.BAS ... Deklarationen für Visual Basic

02 ANGULAR.DLL ... Bibliothek der Winkelfunktionen  
03 ANGULAR.BAT ... Aufruf von FORTRAN-Compiler und -Linker  
04 ANGULAR.DEF ... Linker-Definitionsdatei  
05 ANGULAR.FOR ... FORTRAN-Code der ANGULAR.DLL  
06 FULLDEMO.EXE ... Demo-Programm mit allen Funktionen  
07 VBRUN300.DLL ... Visual Basic Runtime Library  
08 DOKUMENT.TXT ... diese Dokumentation im ASCII-Format  
09 DOKUMENT.WRI ... diese Dokumentation im Windows-Write-Format  
10 FUNCTION.EXE ... alle Funktionsgraphen im PCX-Format  
11 FUNCTION.VBX ... Grafik-Ressourcen für FULLDEMO.EXE

**Anmerkung:**

Die Dateien 03 bis 05 sind zur Anwendung der DLL nicht notwendig,  
sie enthalten die Entwicklungsinformation in MS-FORTRAN 5.1.  
Für eigene Projekte sind die Dateien 01 und 02 zu verwenden, mit der  
erstellten EXE-Datei muß dann nur noch die 'ANGULAR.DLL' geliefert werden!

Viel Erfolg wünscht  
Thomas Meinike  
Basic ist professionell!

E-Mail: [meiniket@mailserv.rz.fh-merseburg.de](mailto:meiniket@mailserv.rz.fh-merseburg.de)  
Telefon: 0345 / 7764468