



# Turm von Hanoi

ftComputing : Programme für die fischertechnik-Interfaces und -konstruktionskästen

[NEU](#)
[Computing](#)
[DLLs](#)
[Modelle](#)
[Downloads](#)
[English Pages](#)

ftComputing.de

[Home](#)
[Back](#)
[Sitemap](#)
[Index](#)
[Links](#)
[Impressum](#)
[Mail](#)

## Über das Scheibenschichten

Ein Stapel von immer kleiner werdenden Scheiben soll über eine Zwischenstation auf einen neuen Platz gestapelt werden. Immer eine auf einmal und stets die kleinere Scheibe auf die größere.

Die Lösung dieses Problems erfreut sich im Informatik-Unterricht großer Beliebtheit (bei den Lehrern). Es gibt deswegen auch immer wieder Lösungen zu diesem Problem. fischertechnik hat bereits im Kasten Computing von 1984 das Modell eines Hanoi-Roboters angeboten. Auf dieser Site finden sich ebenfalls einige Lösungen für das Problem :

- [WinLogo](#) Lösung für den Hanoi Robot von 1984
- [VB6](#) Lösung für den Industry Robot von 1998
- VB6 Lösung für den [LynxMotion](#) Robot

Hier Lösungen für **VBA** mit [vbaFish](#), **JScript** mit mscFish, **VB.NET**, **C#.NET**, **Python** und **Delphi**, alle nutzen FishFa30/umFish30.DLL :

## Die manuelle Lösung

Das nachfolgende kleine Programm Hanoi.ftC (es wurde im PC Magazin Nr. 22 von 1988 als Quick Basic Lösung veröffentlicht) gibt im Debug-Fenster eine Lösungsanweisung für das manuelle Umschichten aus :

```
Sub Main
Dim n&
  n = InputBox("Anzahl Scheiben")
  Debug.Clear
  Debug.Print "Spiel mit" & n & " Scheiben"
  Hanoi n, "Links ", "Mitte ", "Rechts"
End Sub

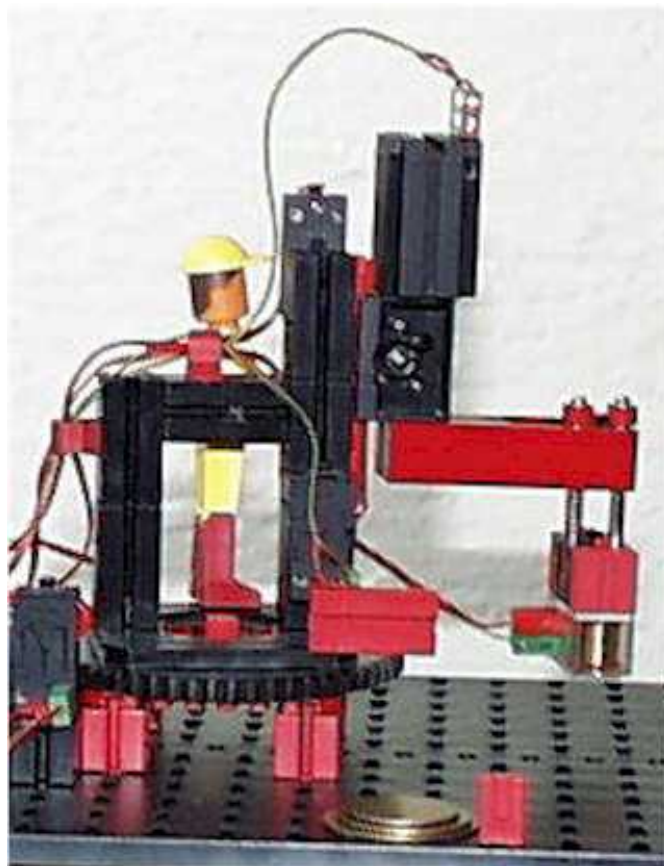
Sub Hanoi(n&, TAnf$, TMit$, TEnd$)
  If n = 1 Then
    Debug.Print "Scheibe von " & TAnf & " -> " & TEnd
  Else
    Hanoi n-1, TAnf, TEnd, TMit
    Debug.Print "Scheibe von " & TAnf & " -> " & TEnd
    Hanoi n-1, TMit, TAnf, TEnd
  End If
End Sub
```

Kern ist das Unterprogramm Hanoi, das aus einer einzigen If-Anweisung besteht. Wenn die Anzahl der Scheiben auf dem Stapel bei 1 angekommen ist, wird direkt vom aktuellen Quellstapel zum Zielstapel geschichtet, sonst wird dreimal hin und her geschichtet. Trick dabei ist der rekursive Aufruf von Hanoi (das Unterprogramm ruft sich selber auf).

Achtung : Zur Anzeige der Lösung ist im Menü Ansicht die Option 'Immer teilen' einzustellen

Die Anweisung Debug.Print "Scheibe von " & TAnf & " -> " & TEnd, die zweimal im Programm auftaucht, ist nun durch eine Anweisung Ziehe zu ersetzen, die einen Robot überredet tätig zu werden, ganz wie im manuellen Verfahren. Dazu ist dann erstmal ein Robot erforderlich :

## Der Hanoi Robot



Der Hanoi Robot ist ein Eigenbau der auf dem Schweißroboter-Chassis des Computing Starter Kits basiert. Er entspricht hier weitgehend dem Hanoi Robot von 1984. Allerdings wurde die Technik modernisiert :

M1 : Säulenmotor mit Impulsrad und Endtaster an E1, Impulstaster an E1

M2 : Armmotor mit Endtaster oben E3 und unten E4

M3 : Gefedert gelagerter Magnet (neuere Version bei [Knobloch](#) : 32363 - 17,20 Euro)

Gestapelt werden die Eisenscheiben, die im Bild vorn (PosA) neben dem Winkel liegen. Verdrahtet wurde "fliegend".

## Die Routinen

-----	Hanoi-Logik
Ziehe(Von, Nach)	Ersatz für die manuelle Arbeitsanweisung : Zerlegt in Hole(Von) und Bringe(Nach)
Bringe(Pos)	Ablegen einer Scheibe auf der gewünschten Zielposition : SaulaNach(Pos), ArmSenken, ScheibeGreifen, ArmHeben
Hole(Pos)	Aufgreifen einer Scheibe von der angegebenen Quellposition SaulaNach(Pos), ArmSenken, ScheibeLegen, ArmHeben
-----	Robot-Ansteuerung
SaulaNach(ZielPos)	Fahren auf die Zielposition. Die Position wird durch eine Impulsrad bestimmt
ArmSenken	Senken des Greiferarms bis zum Endtaster
ArmHeben	Heben des Greiferarms bis zum Endtaster
ScheibeLegen	Einfach : Magnet aus
ScheibeGreifen	Auch einfach : Magnet ein
-----	DrumRum
Grundstellung	Fahren auf Grundstellung ( Robot auf PosA, Armhoch, Magnet aus)
Init	Besetzen der globalen Variablen. Dabei auch festlegen der Werte für die drei Stapelpositionen PosA, PosB und PosC

## Die Sources

Das komplette [VBA](#) Programm gibts gleich zweimal :

- HanoiRobot.ftC : Eine all-in-one Lösung. Alle beschriebenen Routinen sind in einer Source enthalten
- HanoiMain.ftC/HanoiRobot.CLS : Die Robot-Ansteuerung wurde in eine separate Klasse ausgelagert. Hier wohl kaum erforderlich. Mehr um zu zeigen, wie man mit vbaFish auch Klassen einsetzen kann. Die Programme sind funktional gleich

dann gleich noch einmal für [Python](#) :

- HanoiRobot.PY : Ein File, aber wie bei HanoiMain/HanoiRobot wurde die Robot-Ansteuerung in eine Klasse ausgelagert. Getestet wurde mit PythonWin. Ablauf auch als reine Konsolapplikation.

für [Delphi](#) :

- poorHanoiRob.DPR : Eine all-in-one Lösung auf Basis des Templates poorFish30. Alle beschriebenen Routinen sind in einer Source enthalten, Konsolprogramm.

und für [JScript](#) :

- HanoiRob.JS : Eine Lösung auf Basis von Unterprogrammen (function)
- HanoiRobC.JS : Eine Lösungsvariante unter Verwendung eines Objects (Klasse), das die Robot-Methoden kapselt.

und dann noch [VB.NET](#) und [C#.NET](#):

- HanoiRobot : Eine Windows Lösung unter Nutzung der Klasse HanoiRob.

Alles zusammengefaßt in [HanoiRob.ZIP](#). Zusätzlich ist noch [vbaFish30Setup.EXE](#) (VBA) bzw. [PythonFish30.ZIP](#) (Python) oder [delphiFish30Setup.EXE](#) (Delphi) erforderlich (Auf der jeweiligen Seite wird das genauer erklärt). Außerdem die zugehörnden Entwicklungssysteme - Ausnahme : vbaFish30, da ist schon alles dabei.

Stand : 13.10.2003