

 Friedrich-Ebert-Schule Esslingen	µC-Steuerung mit Visual Basic	Name:
3.3.2	Übung 5: Fehlerbehandlung mit "Try...Catch..."	Datum:

Beim Arbeiten mit der seriellen Schnittstelle können Fehler auftreten, die im schlimmsten Fall zum Absturz des Programms oder gar zur Blockade des PC führen können. Solche Fehler sind z.B.:

- Ein Com-Port ist bereits von einem anderen Programm belegt.
- Der Com-Port existiert nicht mehr (z.B. USB-RS232 Konverter entfernt)
- Ein Sende- oder Empfangs-Timeout tritt auf!

In Visual-Basic ist es möglich diese (und viele andere) Fehler bzw. Ausnahmen (=Exceptions) abzufangen. Dazu wird die Try-Catch-Finally-Anweisung verwendet.

Syntax:

Try

Test-Anweisungen (abgesichert)

Catch Fehlervariable As ExceptionTyp

Anweisungen zur Fehlerbehandlung

Finally

Anweisungen die immer ausgeführt werden (auch nach Fehlerbehandlung)

End Try

Beschreibung:

In den **Test-Anweisungen-Block** werden alle Anweisungen geschrieben, die abgesichert werden sollen. Tritt eine Ausnahme (Fehler) in diesem Block auf, wird in den zugehörigen **Catch-Block** verzweigt. Dabei werden in der *Fehlervariablen* Informationen zum aufgetretenen Fehler übergeben. *ExceptionTyp* definiert den Typ des Ausnahmeobjekts (z.B. System.IO.IOException), auf das geprüft wird. Im **Finally-Block** stehende Anweisungen werden immer ausgeführt, auch wenn ein Fehler aufgetreten ist.

Mit dem *ExceptionTyp* kann die Fehlerbehandlung auf bestimmte Fehler begrenzt werden. Es dürfen auch mehrere Catch-Blöcke vorhanden sein. Die System.Exception-Klasse definiert alle möglichen Ausnahmeobjekte. Wird sie als *ExceptionTyp* angegeben, werden alle (vordefinierten) Fehler behandelt.

Arbeitsauftrag

1. Ergänzen Sie die Ereignisprozedur btnSend_Click() um die abgebildete Fehlerbehandlung durch Try...Catch...Finally
2. Testen Sie die Fehlerbehandlung! Blockieren Sie den seriellen Port mit Hyperterminal!

VB-Quellcode

```
Private Sub btnSend_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSer
    Try
        InitSerialPort()
        If Not ComPort.IsOpen Then ComPort.Open()
        ComPort.Write(Chr(2))           ' STX = 2h senden
        ComPort.WriteLine(tbSend.Text)  ' Text senden
    Catch IOerr As System.Exception    ' Ausnahmenbehandlung
        MsgBox(IOerr.Message + vbNewLine + "Der ComPort wird geschlossen!", MsgBoxStyle.Exclamation)
    Finally
        If ComPort.IsOpen Then ComPort.Close()
    End Try
End Sub
```