

Programmierung des CompuLAB unter Visual Basic 5/6

Wer mit Visual Basic externe Hardware wie z.B. das CompuLB ansteuern möchte, sollte eine DLL verwenden, die die Verbindung zur seriellen Schnittstelle des PC herstellt. Mögliche Kandidaten sind die MAPI.DLL von Modul-Bus, die RSAPI.DLL von Franzis und die PORT.DLL von Elektor. Hier soll beispielhaft die Verwendung der RSAPI.DLL erläutert werden.

Die RSAPI.DLL wurde von H.-J.Berndt für das Buch „MSR mit Word und Excel“ geschrieben. Es gibt eine 16-Bit-Version für Win3.1 und eine 32-Bit-Version für Win95/98. Das Buch verwendet die DLL in Word- oder Excel-Makros zur direkten Ansteuerung der Hardware aus Office-Programmen. Man kann z.B. eine Tabelle mit Messwerten direkt in Excel erzeugen. Die Makrosprache VBA (Visual Basic for Applications) ist weitgehend identisch mit Visual Basic, wobei der Standard in Word97 und Excel97 recht genau dem von Visual Basic 5 entspricht. Natürlich gibt es spezifische Befehle z.B. für das Einfügen von Werten in Excel-Tabellenblätter, die in VB fehlen. Andererseits kann man nur mit VB autonom lauffähige EXE-Programme erzeugen.

Die Beschreibung in diesem Aufsatz stützt sich auf Visual Basic 5, das in Schulen sehr verbreitet ist. Erste Versuche kann man auch mit der über das Internet von Microsoft frei erhältlichen Version VB5CCE unternehmen. Es handelt sich dabei um ein voll einsatzfähiges VB5, allerdings ohne einen Compiler für EXE-Programme. Alle Quelltexte lassen sich später mit einem vollwertigen VB5 weiter verarbeiten. Auch das Einlesen in die inzwischen herausgegebene Version VB6 ist problemlos.

Für ein neues Projekt wählt man zunächst die Dateiform Standard-EXE. Man erhält ein leeres Programmfenster, in das sich per Drag and Drop einzelne Control-Elemente wie Buttons, Schieber oder Textfenster ziehen lassen. Ein Doppelklick auf ein Element öffnet eine zugehörige Prozedur, in die man passenden Programmcode eintragen kann. Die Verwendung von Visual Basic ist deshalb so einfach und bequem, weil man eigentlich kein Gesamtprogramm, sondern nur einzelne Prozeduren schreiben muss.

Ein Doppelklick in die Form selbst öffnet die Prozedur (Im VB-Sprachgebrauch: Sub) Form_Load. Der hier eingetragene Code wird beim Start des Programm ausgeführt. Hier hinein gehört also die Initialisierung der seriellen Schnittstelle zum Zugriff auf ein Interface. Nur woher nimmt man einen solchen Befehl, wenn er in VB5 selbst nicht vorhanden ist? Zum Beispiel aus einer DLL!

```
Private Sub Form_Load()
```

```
End Sub
```

Listing 1 Die leere Prozedur Form_Load

Ein neuer Befehl wird mit dem Schlüsselwort DECLARE in VB verfügbar gemacht. Es müssen der Name der Prozedur oder Funktion, der Name der DLL und alle Übergabeparameter angegeben werden. Man könnte meinen, der richtige Ort für diese Deklaration wäre der allgemeine Teil eines VB-Programms, in dem ja auch globale Variable deklariert werden. Aber das funktioniert nicht, sondern man muss erst ein externes Modul in das Projekt einfügen, in das man dann die Deklarationen der DLL-Funktionen schreibt. Das Modul trägt die Endung BAS und ist nichts anderes als ein Stück Basic-Quelltext.

```
Declare Function OPENCOM Lib "RSAPI" (ByVal A$) As Integer
Declare Function INITCOMPULAB Lib "RSAPI" (ByVal COM%) As Integer
Declare Function FINDHARD Lib "RSAPI" (ByVal Meldung%) As Integer
Declare Sub CLOSECOM Lib "RSAPI" ()
Declare Sub SENDBYTE Lib "RSAPI" (ByVal b%)
Declare Function READBYTE Lib "RSAPI" () As Integer
Declare Sub DOUT Lib "RSAPI" (ByVal Wert%)
Declare Function DIN Lib "RSAPI" () As Integer
Declare Function AIN Lib "RSAPI" (ByVal Eingang%) As Integer
Declare Sub DELAY Lib "RSAPI" (ByVal b%)
Declare Sub TIMEINIT Lib "RSAPI" ()
Declare Function TIMEREAD Lib "RSAPI" () As Long
```

Listing 2 Alle Deklarationen im Modul RSAPI.BAS

Listing 2 zeigt das externe Modul mit den wesentlichen Deklarationen für die Ansteuerung des CompuLAB. Alle Funktions- und Prozedurnamen müssen mit Großbuchstaben geschrieben werden. Ganz wichtig ist auch die Übergabe der Parameter mit ByVal. Dieser Zusatz bewirkt, dass die Variablenwerte selbst und nicht ein Zeiger auf sie übergeben wird, wie VB es sonst handhaben würde. So mache Stunde mühseliger Fehlersuche wurde schon damit vertan, dass jemand ByVal in den Deklarationen vergessen hat.

Nun kann endlich die Prozedur Form_Load gefüllt werden. Hier wird „INITCOMPULAB 2“ zur Inialisierung von COM2 verwendet. Genauso gut geht auch „FINDHARD 1“. Man kann dann verschiedene Interfaces einsetzen, die automatisch erkannt werden, außerdem findet das Programm die verwendete Schnittstelle selbst.

```
Private Sub Form_Load()  
    INITCOMPULAB 2  
    ' FINDHARD 1  
End Sub  
  
Private Sub Form_Unload(Cancel As Integer)  
    CLOSECOM  
End Sub
```

Listing 3 Öffnen und Schließen der Schnittstelle.

Nun sind alle Vorbereitungen abgeschlossen, so dass man nun endlich die Hardware ansteuern kann. In einem ersten Versuch sollen alle Ein- und Ausgänge verwendet werden. Werte der digitalen Ausgänge sollen hier, zwar etwas unüblich aber weil es so einfach ist, mit einem Schieberegler eingestellt werden. Alle Eingänge sollen ausgelesen werden und in Textfenstern angezeigt werden.

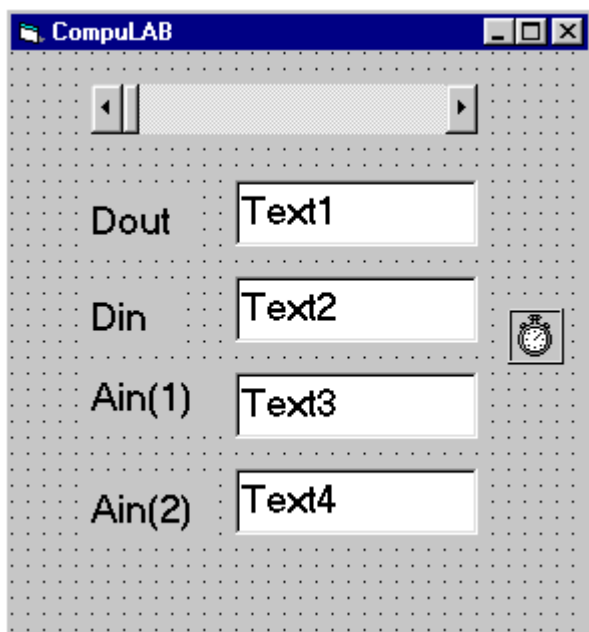


Abb. 1 Das Programm zur Entwicklungszeit

Abb. 1 zeigt den Aufbau des Programmfensters. Man erkennt den Schieberegler und alle Ausgabefenster. Zusätzlich ist noch ein Timer eingebaut, der als Uhr dargestellt wird. Diesen Timer sieht man zur Laufzeit des Programms nicht mehr. Aber er soll hier den gesamten Ablauf steuern. Die entscheidende Eigenschaft „Timer1.Interval“ wird auf den Wert 100 eingestellt, so dass alle 100 ms die Inhalte der Fenster erneuert werden. Alle wesentlichen Programmzeilen stehen daher in der Prozedur „Timer1.Timer“. Listing 4 zeigt das komplette Programm.

```
Private Sub Form_Load()  
    INITCOMPULAB 2  
    ' FINDHARD 1  
End Sub
```

```

Private Sub Form_Unload(Cancel As Integer)
    CLOSECOM
End Sub

Private Sub Timer1_Timer()
    Ausgabe = HScroll1.Value
    Text1.Text = Str$(Ausgabe)
    DOUT Ausgabe
    Text2.Text = Str$(DIN())
    Text3.Text = Str$(AIN(1))
    Text4.Text = Str$(AIN(2))
End Sub

```

Listing 4 Das Programm

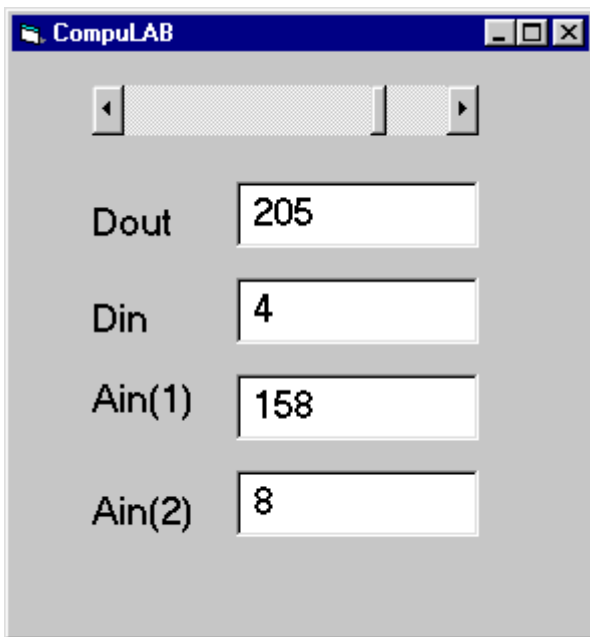


Abb. 2 Das Programm zur Laufzeit

Wer zum ersten Mal ein solches Programm mit Visual Basic erstellt, ist meist erstaunt, mit wie wenig Aufwand man ein brauchbares Ergebnis erzielen kann. Der spielerische Umgang mit einer Programmiersprache macht hier wirklich Spaß. Mit dem vorliegenden Programm ist noch eine zusätzliche Spielerei möglich: Klickt man den Schieberegler an einer Seite an und hält die Maustaste gedrückt, dann erhält man einen frei laufenden Binärzähler mit 8 Ausgängen. Die LEDs am CompuLAB blinken lustig vor sich hin und ermöglichen nebenbei noch tiefeschürfende Betrachtungen zum Aufbau von Binärzahlen.

Hinweis: Eine Übersicht zur Problematik der Schnittstellenansteuerung unter Windows gibt auch die Homepage des Autors: <http://home.t-online.de/home/B.Kainka>