Phybox.dll Pflichtenheft

% = Integer , & = Long Integer, ! = Single, $ * len = String mit der Länge len
True = -1%
False = 0%

Ok% = PB_Init (ByVal hWnd%) %
SerNr& = PB_SerNr()&
FWVersion! = PB_FirmwareVersion()!
PB_SetChannel (PBChannelStructure)
PB_CreateChannel (ChannelStructure)
ErrorPos% = PB_Compile (ByVal ChanNr%, ByVal ErrorCode%) %
PB_Start (ByVal StartByte%)
Ok% = PB_ProcessData()%
Wert! = PB_Werte (ByVal Page%,ByVal Kanal%,ByVal Index&) !
PB_SetWerte (ByVal Page%, ByVal Kanal%, ByVal Index&, ByVal Wert!)
PB_SetPage (ByVal Page%)
PB_PCTrigger (ByVal TriggerNr%)
Ok% = PB_FROMUpload (ByVal File$*120)%
Ok% = PB_FROMDownload (ByVal File$*120, ByVal ADRStart&, ByVal ADRStop&)%
PB_Calibrate (ByVal Channel%, ByVal CalRange!, ByVal CalPoint!)

Das Exception Handling von Delphi muß innerhalb der ganzen DLL so verwendet werden, daß ein Fehler an einer beliebigen Stelle im Treiber nicht zu einem Absturz des Systems führt. Treten Fehlerzustände auf, so werden diese in der Datei PHYX.ERR fortlaufend protokolliert.

z.B.	14.12.1995 Treiber:	GP in PB_Calibrate
	15.12.1995 Treiber:	Stack Overflow in ProcessData

PB_Init (hWnd%) %

% = Integer;
Inizialisiert den Treiber und übergiebt den WindowHandle des Fensters an das die Messages gepostet werden sollen.

Rückgabewert: True OK,
False DLL lässt sich nich starten

PB_SerNr () &

& = Long Integer
Liefert die Phybox-SerienNr

Rückgabe-Wert: LongInteger
0 wenn Phybox nicht aktiv
PB SerienNr wenn erfolgreiche Übertragung

PB_SetSerNr(ByVal Seriennummer&)

& = Long Integer
setzt die Seriennummer der Phybox

PB_FirmwareVersion () !

! = Single
Liefert die Phybox Firmware-Version

Rückgabe-Wert: Single
0 wenn Phybox nicht mehr aktiv oder keine lauffähige Firmware geladen
sonst Versions-Nr der Phybox-Firmware z.B. 1,1 für Version 1.1

PB_SetChannel (PBChannelStructure)

Parametriert einen Pybox-Kanal mit der PBCahnnelStructure.

```
Type TPBChannel
    ChannelNr%              'Nummer des Phybox-Kanals
    Enabled%                'Kanal-Einstellung für ungepufferte und gepufferte Wandlung
    Mode%                   'Integer Parameter meist für Mode verwendet
    NrParamSingle%          'Anzahl der folgenden Single-Parameter
    ParamSingle![]          'Array von Single Parametern
End Type
```

PB_Start (StartByte%)

Setzt das StartByte der Phybox. Startet und stoppt die Meßwert-Aufnahme, DA, Zähler etc.

PB_ProcessData() %

Teilt der DLL mit, daß der Seriell-Puffer abgearbeitet werden soll.

Rückgabewert: Integer in dem ChemboxReset gemeldet wird
0 OK
1 Reset

PB_CreateChannel (ChannelStructure)

Erzeugt einen neuen Softwarekanal gemäß den Angaben in der Channel-Structure.

```
Type TChannel
    DatenNr As Integer          'Kanal Datenverwaltungsnummer (für Source Bezugsangaben)
    KanalNr As Integer          'Meßwerte Spalten-Index vom Werte-Array
    Source As Integer           '0 Formel speist,
                                'DatenNr Kanal speist über Mul,Add ein
                                'negativ Phybox speist über Mul Add ein
                                '       -1 TimeBase
                                '       -2 ΔTimeBase
                                '       -3 Sensor1
                                '       -4 Sensor1B
                                '       -5 Sensor1C
                                '       -6 Sensor2
                                '       -7 Sensor2B
                                '       -x   usw.
    Formel As String * 250      'Verrechnugs-Formel, die den Kanal speist
    VarName As String * 10      'Variablenname des Kanals in Verrechnungs-Formeln
    Mul As Single               'mit Quellwerten multiplizieren
    Add As Single               'dazu addieren und in Kanal einspeisen
    Wert As Single              'ungepufferter Wert = letzer in Kanal gespeister Wert
    Start As Long               'Index des ersten Wertes im Werte-Array
    Stop As Long                'Index des letzten Wertes im Werte-Array
    Tiefe As Integer            'max Werteanzahl des Kanals (=Tiefe des Werte-Arrays)
    Gepuffert As Integer        'legt fest ob das Feld einen Puffer im Werte-Array hat
    Triggermode As Integer
    Triggerlevel As Single
    Triggerlevel2 As Single      'unterer Triggerlevel für Fenstertrigger
    Pretrigger As Single
    PBChanEnable As Integer     'für den Kanal relevanter Teil des PB Startbytes
End Type
```

- `Mul`, `Add` oder `Formel` dienen als Eingangsweg in das Feld je nach `Source`. Ist `Source` gleich einer `DatenNr` eines anderen Kanals so speist dieser Kanal über `Mul`, `Add`, ist `Source` negativ, so speist die Hardware über `Mul`, `Add`. Ist Source nicht definiert, so wird der Kanal nicht gespeist. Ist `Source=0`, wird der Kanal über den Formel-Parser gespeist. Dieser benützt die in `Formel` angegebene Verrechnungsformel. Der Variablennamen der zu verrechnenden Kanäle ist in `VarName` dieser Kanäle gespeichert. Ist die Formel auf eine einzelne Variable mit multiplikativem und additiven Anteil zu reduzieren, setzt der Parser `Mul` und `Add` und stellt `Source` auf den speisenden Kanal.
- Ist die Variable `Gepuffert` gesetzt, so liegt hinter dem Kanal eine Spalte des Werte-Arrays die mit `KanalNr` fesgelegt wird. Ist die `Gepuffert=False` so werden die Daten nur temporär gespeichert um die Formeln abarbeiten zu können.
- `Start` und `Stop` sind Index des ersten bzw. letzten Wertes im Werte-Array
- `Tiefe` definiert die Tiefe des Werte-Arrays

PB_Compile (ByVal ChanNr%, ByVal ErrorCode%) %

Compiliert alle Formeln in Feld-Struktur

PB_Werte (ByVal Page%,ByVal Kanal%,ByVal Index&) !

% = Integer, & = Long, ! = Single

Liest Werte aus dem Werte-Array

PB_SetWerte (ByVal Page%, ByVal Kanal%, ByVal Index&, ByVal Wert!)

% = Integer, & = Long, ! = Single

Schreib die Variable Wert in das Werte-Array. Falls der spezifizierte Kanal bzw. die Page bisher nicht benützt wurden, erfolgt automatisch ein entsprechendes MemAlloc. Die Tiefe des Arrays (MaxIndex) kann in ChannelStructure.Tiefe abgefragt werden

PB_SetPage (ByVal Page%)

% = Integer

Schaltet die aktuelle Schreiber-Page um

PB_PCTrigger (TriggerNr%)

% = Integer

Sendet ein Trigger-Ereignis an die Phybox

PB_FROMUpload (File$*120) %

% = Integer, $ * len String mit der Länge len

Firmware wird aus der Datei File$ gelesen und in die Phybox übertragen. Nach Übertragung des Codes in die Phybox erfolgt automatisch ein Verify.

Rückgabe-Wert: Integer
-1 wenn erfolgreiche Übertragung
0 wenn nicht erfolgreich

File$ String mit Länge 120 Zeichen z.B. „D:\PHYX\FIRMWARE.HEX “

PB_FROMDownload(File$*120,ADRStart&,ADRStop&)%

% = Integer, $ * len String mit der Länge len

Firmware wird aus der Phybox gelesen und in der Datei File$ mit dem Header „Phybox Firmware ^Z“, ADRStart&,ADRStop& gespeichert. Nach Download des Codes erfolgt automatisch ein Verify.

Rückgabe-Wert: Integer
-1 wenn erfolgreich
0 wenn nicht erfolgreich

File$ String mit Länge 120 Zeichen z.B. „D:\PHYX\FIRMWARE.HEX “

PB_Calibrate (ByVal Channel%, ByVal CalRange!, ByVal CalPoint!)

% = Integer, ! = Single

Dient zur Kalibrierung der Phybox. Nimmt einen Kalibrierpunkt bei CalPoint! im Meßbereich CalRange! auf. Sind pro Kanal zwei Kalibrierpunkte erfasst, verwendet die Phybox diese als Kalibrierwerte.