

# ***ADwin***

## **LabVIEW-Treiber**



## Inhaltsverzeichnis

<b>1</b>	<b>ADwin-Treiberkonzept .....</b>	<b>3</b>
1.1	ADwin Betriebssystem .....	3
1.2	Windows-ADwin-Schnittstelle .....	4
<b>2</b>	<b>Software Installation .....</b>	<b>5</b>
2.1	Installation der ADwin-Treiber .....	5
2.2	Laden des ADwin-Betriebssystems .....	6
2.1	Einbinden der ADwin VIs .....	7
<b>3</b>	<b>LabVIEW VIs für ADwin-Systeme .....</b>	<b>9</b>
3.1	ADbasic-Funktionen .....	9
3.1.1	ADboot.VI .....	9
3.1.2	ADbload.VI .....	9
3.1.3	ADbstart.VI .....	9
3.1.4	ADbstop.VI .....	9
3.1.5	SetPar_i.VI .....	10
3.1.6	SetPar_f.VI .....	10
3.1.7	GetPar_i.VI .....	10
3.1.8	GetPar_f.VI .....	10
3.1.9	SetiData.VI .....	11
3.1.10	SetfData.VI .....	11
3.1.11	GetiData.VI .....	11
3.1.12	GetfData.VI .....	11
3.1.13	GetiFifo.VI .....	12
3.1.14	GetfFifo.VI .....	12
3.1.15	Auslast.VI .....	12
3.1.16	Memory.VI .....	12
3.2	System- und einfache I/O-Funktionen .....	13
3.2.1	Get_adc.VI .....	13
3.2.2	Set_dac.VI .....	13
3.2.3	GetDigin.VI .....	13
3.2.4	Dout_bit.VI .....	13
3.2.5	Dout_val.VI .....	13
3.2.6	Net_Connect.VI .....	14
3.2.7	Net_Disconnect.VI .....	14
3.3	Messprozess .....	15
3.3.1	Acquire.VI .....	15
3.4	Prozess zur analogen Ausgabe .....	15
3.4.1	StartDAC.VI .....	15
3.4.2	StopDAC.VI .....	15
3.5	Hilfsmittel .....	16
3.5.1	V_to_Val.VI .....	16
3.5.2	Val_to_V.VI .....	16
<b>4</b>	<b>Beispielprogramm „MinMax.VI“ .....</b>	<b>17</b>
4.1	Programmaufbau .....	17
<b>5</b>	<b>Befehlsindex .....</b>	<b>18</b>

## 1 ADwin-Treiberkonzept

### 1.1 ADwin Betriebssystem

Das **ADwin**-Betriebssystem befindet sich in der Datei mit der Bezeichnung **adwin9.btl**<sup>1</sup> (ADSP bzw. T9). Nach jedem Einschalten der Spannungsversorgung muß zunächst das Betriebssystem auf das **ADwin**-System geladen werden. Erst nach dessen erfolgreicher Übertragung ist das System in der Lage, Befehle vom PC entgegzunehmen und Daten mit ihm auszutauschen.

Die Aufgaben des **ADwin**-Betriebssystems sind:

- Verwaltung von bis zu 10 **ADbasic**-Prozessen, wobei zwischen zwei frei wählbaren Prioritätsstufen unterschieden wird (siehe nachfolgende Tabelle).

Priorität	Merkmal	Verwendungszweck
niedrig	kann von hoch priorisierten Prozessen unterbrochen werden.	für zeitunkritische Berechnungen und langsame Messungen.
hoch	kann von keinem anderen Prozess unterbrochen werden.	für zeitgenaue Messungen, Steuerungen und Regelungen.

- Bereitstellung von 80 vordefinierten **ADbasic**-Integer-Variablen (PAR\_1 bis PAR\_80) und 80 vordefinierten **ADbasic**-Float-Variablen (FPAR\_1 bis FPAR\_80). Außerdem werden 200 Datensätze mit frei definierbarer Länge bereitgestellt. Die Werte dieser Variablen bzw. Datensätze können vom PC jederzeit gelesen und geändert werden.
- Organisation und Durchführung der Kommunikation zwischen **ADwin**-System und PC.

Einen wesentlichen Bestandteil des **ADwin**-Betriebssystems bildet der Kommunikationsprozess. Dieser Prozess läuft mit niedriger Priorität auf dem **ADwin**-System und interpretiert bzw. bearbeitet alle Befehle, die der PC an das **ADwin**-System richtet. Die wichtigsten Befehle lassen sich in zwei Gruppen unterteilen. In den folgenden Tabellen sind aus jeder Gruppe einige Beispiele aufgelistet.

Steuerbefehle	
<b>ADBload</b>	überträgt einen <b>ADbasic</b> Prozeß auf das <b>ADwin</b> -System
<b>ADBstart</b>	startet einen <b>ADbasic</b> -Prozeß

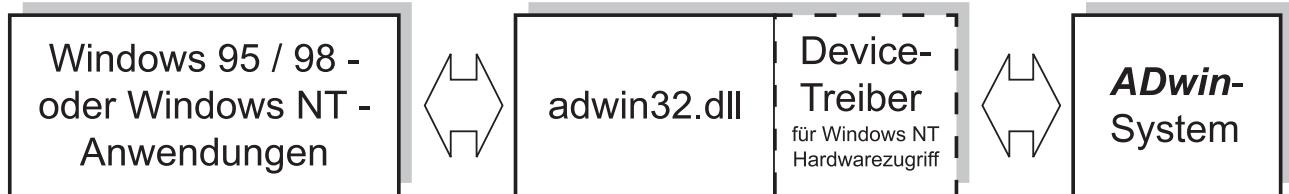
Befehle für den Datenaustausch	
<b>get_par</b>	liefert den aktuellen Wert eines <b>ADbasic</b> -Parameters
<b>set_par</b>	ändert den Wert eines <b>ADbasic</b> -Parameters
<b>get_data</b>	liefert die Werte aus einem <b>ADbasic</b> -Datensatz

Der Kommunikationsprozess sendet niemals unaufgefordert Daten an den PC. Dadurch wird sichergestellt, daß nur dann Daten zum PC übertragen werden, wenn diese vorher explizit angefordert wurden.

<sup>1</sup> Bei Verwendung eines **ADwin**-2, -4, -5 oder -8 Systems wird die Datei adwin2.btl, adwin4.btl, adwin5.btl bzw. adwin8.btl benötigt.

## 1.2 Windows-ADwin-Schnittstelle

Die Schnittstelle zwischen Windows und dem **ADwin**-System bildet die **adwin32.dll** (Dynamic Link Library). Alle Windows Programme, die in der Lage sind DLL-Funktionen aufzurufen, können mit dem **ADwin**-System kommunizieren (siehe nachfolgende Abbildung).



ADwin-Kommunikationsschnittstelle zu Windows-Anwendungen

Die Schnittstelle bewältigt dabei folgende Aufgaben:

- **Befehlsweiterleitung**  
Alle Befehle, die an das **ADwin**-System gerichtet sind, laufen über die **adwin32.dll**. Dadurch ist die Schnittstelle für alle Windows Programme identisch. Die **adwin32.dll** ist das einzige Modul, das mit dem **ADwin**-System in Verbindung treten kann.
- **Datentransfer**  
Der Datentransfer zwischen **ADwin**-System und PC wird in der **adwin32.dll** durchgeführt. Unter Windows NT erfolgt der Zugriff über den **ADwin**-Device-Treiber. Alle anderen Windows Versionen erlauben den direkten Zugriff auf das **ADwin**-System.  
Da weder Interrupts noch DMA-Kanäle benötigt werden, ist ein zuverlässiger Betrieb des **ADwin**-Systems in jedem PC problemlos möglich und aufwendige Konfigurationsprozeduren entfallen.
- **Multitasking-Verwaltung**  
Mit dem **ADwin**-System können mehrere Windows Programme gleichzeitig kommunizieren (Zeitscheiben-Verfahren). Wenn ein Programm Daten mit dem **ADwin**-System austauscht, wird der Zugriff für alle anderen Programme gesperrt. Sofort nach der Beendigung des Austausches gibt die **adwin32.dll** das **ADwin**-System für folgende Anforderungen frei. Der Zeitaufwand für den Datenaustausch liegt im Bereich von wenigen Millisekunden.

## 2 Software Installation

### 2.1 Installation der *ADwin*-Treiber

Wenn Sie die Treiber schon im Zuge der **ADbasic** Installation aufgespielt haben, ist eine erneute Installation nicht erforderlich.

Andernefalls legen Sie die mitgelieferte CD-ROM in das Laufwerk Ihres Rechners ein. Das Setup-Programm wird normalerweise automatisch gestartet.

Sollte dies nicht der Fall sein, dann wechseln Sie bitte in das Unterverzeichnis ...\\Driver\\Disk1 auf der CD-ROM und starten das dort befindliche Programm **setup.exe**.

Nachdem Sie die Sprache und das Zielverzeichnis - es wird empfohlen das angebotene Verzeichnis C:\\ADbasic3 zu bestätigen - ausgewählt haben, kopiert das Setup-Programm die folgenden Dateien in das Verzeichnis Ihres Rechners:

<b>ADWIN2.BTL</b>	Betriebssystem für die <b>ADwin</b> -Systeme mit T225-Prozessor
<b>ADWIN4.BTL</b>	Betriebssystem für die <b>ADwin</b> -Systeme mit T400-Prozessor
<b>ADWIN5.BTL</b>	Betriebssystem für die <b>ADwin</b> -Systeme mit T450-Prozessor
<b>ADWIN8.BTL</b>	Betriebssystem für die <b>ADwin</b> -Systeme mit T805-Prozessor
<b>ADWIN9.BTL</b>	Betriebssystem für die <b>ADwin</b> -Systeme mit ADSP-Prozessor
<b>TESTVE16.EXE</b>	zeigt die Version der installierten 16Bit- <b>ADwin</b> -DLLs an
<b>TESTVE32.EXE</b>	zeigt die Version der installierten 32Bit- <b>ADwin</b> -DLLs an
<b>ADTEST.EXE</b>	Programm zum Testen der <b>ADwin</b> -PC-Einsteckkarten
<b>ADPRO.EXE</b>	Programm zum Testen des <b>ADwin-Pro</b> -Systems
<b>ADWINSET.EXE</b>	Programm zum Anmelden der Linkadresse unter Windows NT.  Wenn Sie eine andere Linkadresse als \$150 verwenden möchten, muss die neue Adresse mit diesem Programm angemeldet werden !



Falls Sie unter Windows NT arbeiten, müssen Sie bei der Installation über Administrator Rechte verfügen. Nach der Treiber-Installation muß Ihr Rechner neu gestartet werden.


### 2.2 Laden des ADwin-Betriebssystems

Der PC kann erst mit dem **ADwin**-System kommunizieren, nachdem das Betriebssystem geladen wurde. Das Betriebssystem wird wahlweise durch die Visual-Basic Funktion Boot oder über **ADbasic** geladen. Zusätzlich bietet auch das Testprogramm **ADtest** die Möglichkeit das System zu laden.

#### a) Unter LabView

Mit dem **ADwin** ADBoot VI. Nähere Informationen hierzu in Kapitel 3.

#### b) Über ADbasic

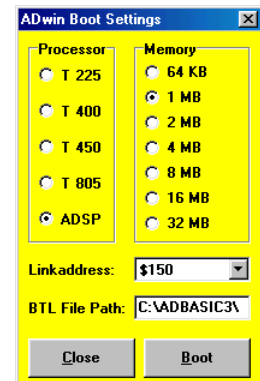
- ♦ Starten Sie **ADbasic** indem Sie im Windows-Menü Start ⇒ Programme ⇒ ADwin ⇒ **ADbasic 3.0** auswählen.
- ♦ Überprüfen Sie, ob die Einstellungen im **ADbasic**-Menü Options ⇒ Compiler mit Ihrem **ADwin**-System übereinstimmen.
- ♦ Klicken Sie in der **ADbasic** Toolbar auf das folgende Symbol:  (alternativ: Project ⇒ Boot ADwin)

Das erfolgreiche Laden des Betriebssystems wird in der Statuszeile des **ADbasic**-Editors durch die Meldung „**ADwin** is booted“ bestätigt.

#### c) Mit dem Windows Programm ADtest

Das Programm **ADtest** dient als Funktionskontrolle für **ADwin**-PC-Einsteckkarten und bietet ebenfalls eine Möglichkeit zum Laden des Betriebssystems. Starten Sie dazu das Programm **ADtest.exe**. Falls das Betriebssystem seit dem Einschalten des PCs noch nicht auf das **ADwin** System geladen wurde, erscheint die folgende Dialogbox:

Stimmen Sie alle Einstellungen auf dem von Ihnen eingesetzten **ADwin**-System ab, und betätigen Sie anschließend die OK-Taste. Falls keine Fehlermeldung ausgegeben wird, konnte das Betriebssystem erfolgreich geladen werden. Das Programm **ADtest** zeigt dann die momentan an den analogen Eingängen 1-12 anliegenden Spannungen und die Pegel an den digitalen Eingängen an. Außerdem können die analogen Ausgänge und die digitalen Ausgänge gesetzt werden. Auf den analogen Ausgängen kann zum Testen ein Sinus- oder Dreiecksignal mit 10 Hz ausgegeben werden. Falls das Betriebssystem nicht geladen werden konnte, läßt sich der Vorgang durch Betätigen der Settings-Taste mit geänderten Einstellungen wiederholen.



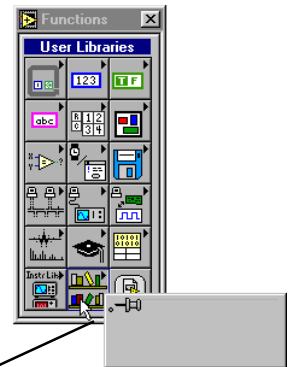
Durch das Laden des Betriebssystems werden alle Prozesse auf dem **ADwin**-System gelöscht und alle globalen Variablen auf den Wert 0 gesetzt. Der Wert für das Globaldelay ist nach dem Laden des Betriebssystems beim ADSP auf 25000 ns und bei allen anderen Prozessoren auf 1000 µs voreingestellt.

### 2.1 Einbinden der ADwin VIs

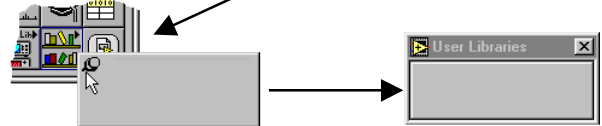
In LabVIEW Programmen können Sie mit Hilfe vorgefertigter VIs, Befehle und Daten an das ADwin-System schicken und Daten vom System abrufen. Die VIs nutzen hierzu Funktionen die in der DLL: ADwin32.dll implementiert sind.

Zum Einbinden der **ADwin** VIs in LabVIEW befolgen Sie bitte die folgenden Anweisungen:

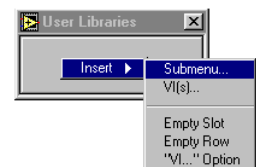
- Kopieren Sie den Ordner **ADwin.lib** inclusive allem Inhalt in Ihr LabVIEW Verzeichnis
- Starten Sie LabVIEW und aktivieren Sie das weiße Diagram Fenster
- Drücken Sie im Fenster: 'Functions' mit der linken Maustaste auf das Symbol 'User Libraries'



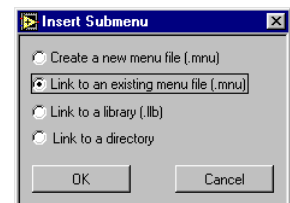
- Das neu erscheinende Fenster muß durch anklicken der Reiszwecke auf dem Desktop verankert werden. (s. rechts)



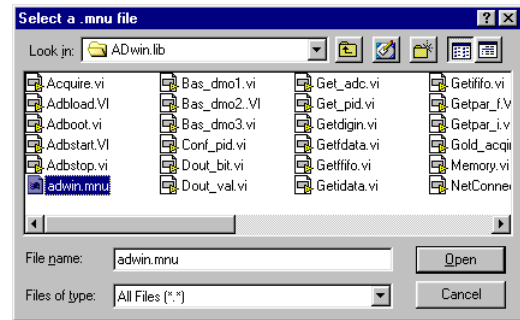
- Wählen Sie im Menü des Diagram Fensters die Option: 'Edit ⇒ Edit Control & Function Palettes'
- Drücken Sie mit der rechten Maustaste auf die graue Fläche im User Libraries Fenster und wählen im daraufhin erscheinenden Menü den Punkt: Insert ⇒ Submenu



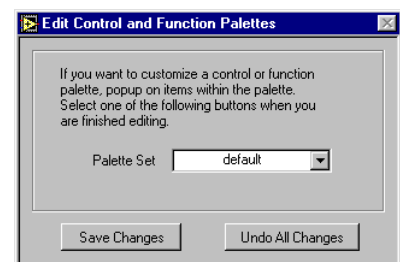
- Im Insert Submenu Fenster wählen Sie die Einstellung: Link to an existing menu file und betätigen anschließend die Taste: OK



- Wählen Sie in der Dateiauswahlbox im Verzeichnis **ADwin.lib** die Datei **ADwin.mnu**.



- Speichern Sie die Änderung durch die Betätigung der Taste: 'Save Changes' im Fenster Edit Control and Functions Palettes.



- Alle **ADwin** VIs erscheinen jetzt in der Gruppe User Libraries und können direkt von dort in Ihre LabVIEW Diagramme gezogen werden. Die Beschreibung der einzelnen VIs folgt im nächsten Kapitel.



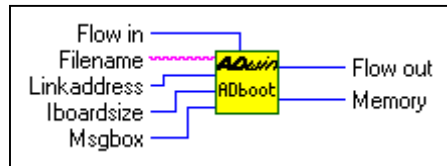


### 3 LabVIEW VIs für ADwin-Systeme

#### 3.1 ADbasic-Funktionen

##### 3.1.1 ADboot.VI

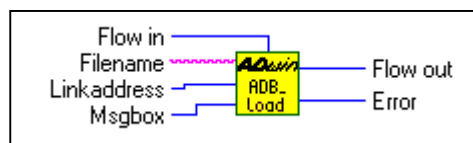
Booten des ADwin-Systems mit der Datei ADwin9.btl (default). Wenn Sie einen anderen Prozessor benutzen, müssen Sie den Eingang „Dateiname“ mit dem korrekten Wert (z.B. ADwin8.btl) beschalten.



**Hinweis:** Durch öffnen der VIs erhalten Sie weitere Hilfe zur Beschaltung.

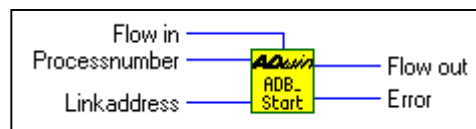
##### 3.1.2 ADbload.VI

Dieses VI lädt die Binärdatei (z.B. \*.t91) eines **ADbasic**-Programms auf das **ADwin**-System.



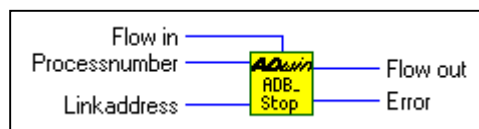
##### 3.1.3 ADbstart.VI

Dieses VI startet einen geladenen Prozeß.



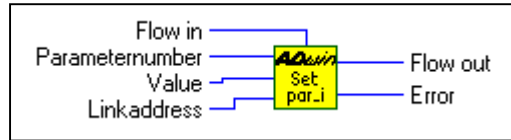
##### 3.1.4 ADbstop.VI

Dieses VI stoppt einen laufenden Prozeß.



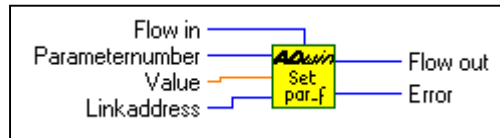
### 3.1.5 SetPar\_i.VI

Das SetPar\_i.VI übergibt einen **long integer** Wert an einen der 80 **ADbasic** Integer Parameter (Par\_1 bis Par\_80).



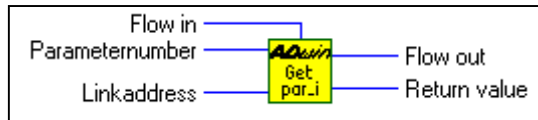
### 3.1.6 SetPar\_f.VI

Das SetPar\_f.VI übergibt einen floating point Wert an einen der 80 ADbasic float Parameter (FPar\_1 bis FPar\_80).



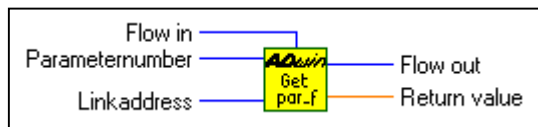
### 3.1.7 GetPar\_i.VI

Dieses VI liest den aktuellen Wert von einem der 80 **ADbasic** integer Parameter (Par\_1 bis Par\_80) aus. Der Rückgabewert ist vom Typ **long integer**.



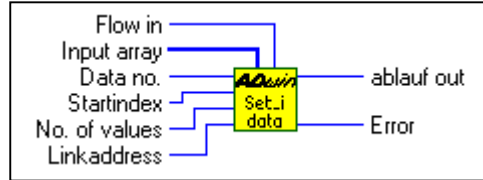
### 3.1.8 GetPar\_f.VI

Dieses VI liest den aktuellen Wert von einem der 80 **ADbasic** float Parameter (FPar\_1 bis FPar\_80) aus. Der Rückgabewert ist vom Typ floating point.



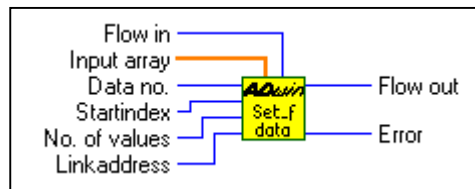
### 3.1.9 SetiData.VI

Dieses VI übergibt ein Array mit **integer** Daten an den gewählten **ADbasic** Datensatz (data\_1...data\_200).



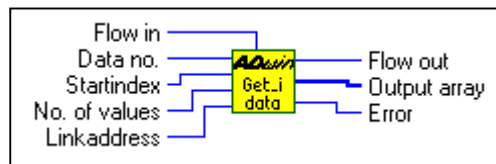
### 3.1.10 SetfData.VI

Dieses VI übergibt ein Array mit **floating point** Daten an den gewählten **ADbasic** Datensatz (data\_1...data\_200).



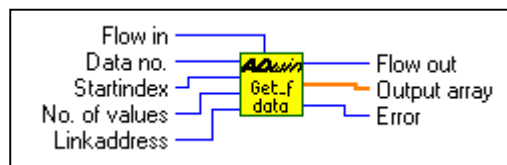
### 3.1.11 GetiData.VI

Das GetiData.VI liest einen **ADbasic** Datensatz vom Typ **integer** aus und gibt dieses Array auf den Ausgang.



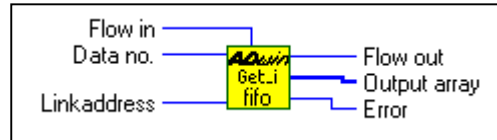
### 3.1.12 GetfData.VI

Das GetfData.VI liest einen **ADbasic** Datensatz vom Typ **floating point** aus und gibt dieses Array auf den Ausgang.



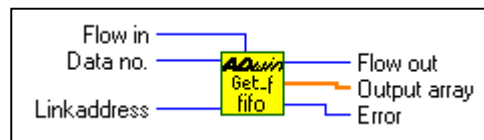
### 3.1.13 GetiFifo.VI

Das GetiFifo.VI liest alle Werte aus einem ADbasic Fifo vom Typ integer aus und gibt dieses Array auf den Ausgang.



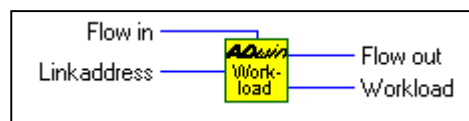
### 3.1.14 GetfFifo.VI

Das GetfData.VI liest alle Werte aus einem **ADbasic** Fifo vom Typ **floating point** aus und gibt dieses Array auf den Ausgang.



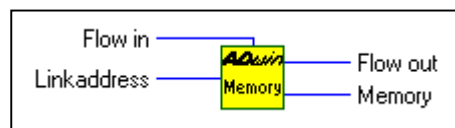
### 3.1.15 Auslast.VI

Dieses VI gibt die Auslastung (in %) des Prozessors Ihres **ADwin**-Systems aus.



### 3.1.16 Memory.VI

Das Memory.VI zeigt den freien Speicher (in Byte) des **ADwin**-Systems an.

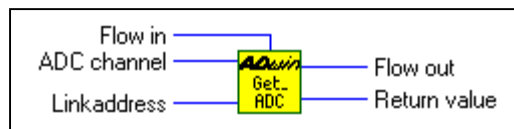


### 3.2 System- und einfache I/O-Funktionen

In diesem Abschnitt werden VIs beschrieben, mit denen man direkt auf die analogen und digitalen Aus- und Eingänge des **ADwin**-Systems zugreifen kann.

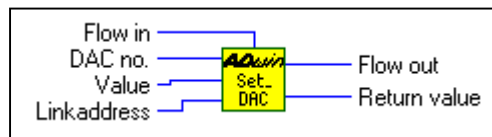
#### 3.2.1 Get\_adc.VI

Dieses VI liest einen analogen Eingang des **ADwin**-Systems aus. Der return Wert ist vom Typ **integer**. Für die Umrechnung in Volt kann das **Val\_to\_V.VI** genutzt werden.



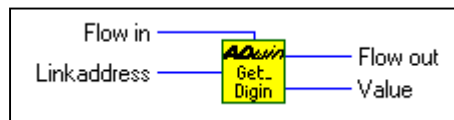
#### 3.2.2 Set\_dac.VI

Dieses VI setzt einen analogen Ausgang auf einen **integer** Wert.



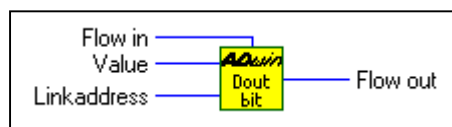
#### 3.2.3 GetDigin.VI

Dieses VI zeigt den Zustand der digitalen Eingänge. Der Ausgabewert ist vom Typ **long integer**, in **dezimaler** Darstellung.



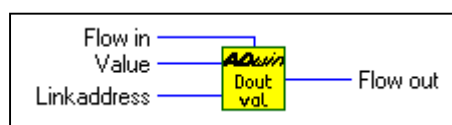
#### 3.2.4 Dout\_bit.VI

Dieses VI setzt die digitalen Ausgänge des **ADwin**-Systems. Die Eingabe erfolgt hier in **Binärdarstellung** (0=löschen, 1=setzen, für 16 Bit [15.....0]).



#### 3.2.5 Dout\_val.VI

Dieses VI setzt ebenfalls die digitalen Ausgänge des **ADwin**-Systems. Die Eingabe ist jedoch **dezimal** (wird binär ausgewertet).



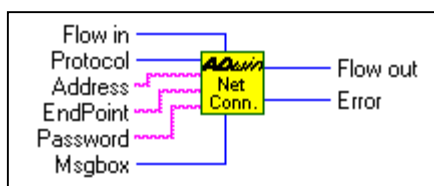
### Hinweis zur Netzwerkverbindung:

Das VI *Net\_Connect* baut über Netzwerk eine Verbindung zu einem Rechner (Wirtsrechner) auf, an dem ein **ADwin**-System angeschlossen ist.

Auf dem Wirtsrechner muß vorher das Programm **ADServer.exe** gestartet werden. Nach erfolgreichem Verbindungsaufbau gehen alle Zugriffe auf das **ADwin**-System automatisch über das Netzwerk zum Wirtsrechner.

#### 3.2.6 Net\_Connect.VI

Dieses VI baut eine Netzwerkverbindung zu einem Rechner mit ADwin-System auf

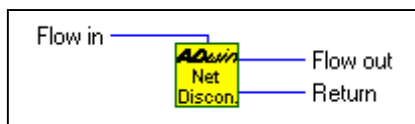


Auswahl des Netzwerkprotokolls:

Protocol	Netzwerkprotokoll
0	NetBIOS / NetBEUI
1	IPX / NetBEUI
2	NetBIOS / TCP/IP
3	TCP/IP
4	Named pipes
5	SPX
6	DECnet

#### 3.2.7 Net\_Disconnect.VI

Dieses VI beendet eine Netzwerkverbindung die mit dem VI Net\_Disconnect aufgebaut wurde.

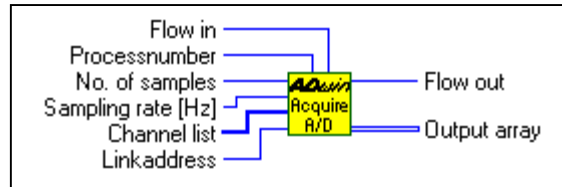


### 3.3 Messprozess

In diesem Abschnitt werden VI's beschrieben, mit denen man direkt auf die **analogen Eingänge** (ADC 1-16) zugreifen kann.

#### 3.3.1 Acquire.VI

Das Acquire.VI führt eine A/D-Wandlung auf einem oder mehreren analogen Eingangskanälen mit einer festgelegten Abtastrate durch. Mit diesem VI kann man alle 16 analogen Eingangskanäle zyklisch messen.

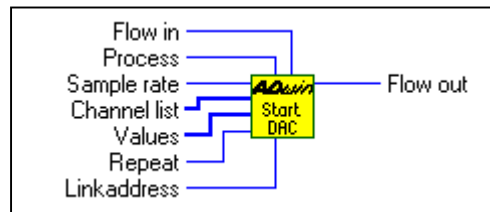


### 3.4 Prozess zur analogen Ausgabe

Die VIs in diesem Abschnitt eignen sich dazu, direkt auf die analogen Ausgangskanäle (DAC 1-6) zuzugreifen, und Werte zyklisch auszugeben. Das **ADwin**-System bietet die Möglichkeit, zwei verschiedene Prozesse mit unterschiedlichen Taktraten parallel ablaufen zu lassen.

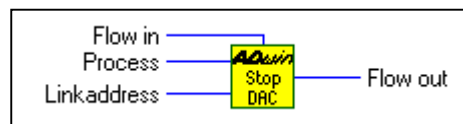
#### 3.4.1 StartDAC.VI

Das StartDAC.VI startet die zyklische Ausgabe mit einer vorgegebenen Ausgaberate auf bis zu 6 analogen Kanälen gleichzeitig.



#### 3.4.2 StopDAC.VI

Dieses VI hält einen gestarteten und noch nicht beendeten Ausgabe-Prozess vorzeitig an.

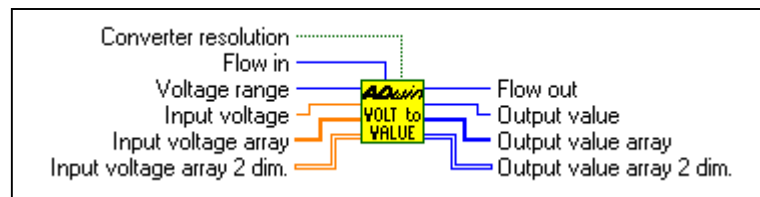


### 3.5 Hilfsmittel

Da manche der Ein-/Ausgabe VIs Werte im „Raw-Data-Format“ (Integer-Werte von 0 – 4095 [12-Bit], bzw. 0 – 65535 [16-Bit]) benötigen und liefern, gibt es zwei VIs, die die Umrechnung von Volt in Value bzw. umgekehrt von Value in Volt übernehmen. Die Eingänge sind für einfache Werte, Arrays und 2 dimensionale Arrays ausgelegt und können nach Bedarf beschaltet werden (Hinweis: Wenn die Ausführungszeiten für besondere Fälle zu lang sein sollten, können die passenden Umrechnungssequenzen einfach aus den VIs herauskopiert und direkt im Programm verwendet werden.).

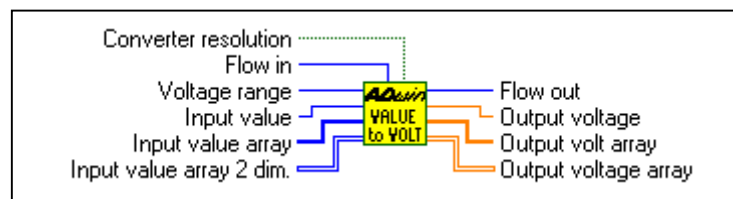
#### 3.5.1 V\_to\_Val.VI

Umrechnung von Volt -Werten in Values (0 – 4095 [12-Bit], bzw. 0 – 65535 [16-Bit]), abhängig vom eingestellten Bereich auf dem **ADwin**-System (0...+10V, -5...+5V, -10...+10V).



#### 3.5.2 Val\_to\_V.VI

Umrechnung von Values (0 – 4095 [12-Bit], bzw. 0 – 65535 [16-Bit]) in Volt -Werte, abhängig vom eingestellten Bereich auf dem **ADwin**-System (0...+10V, -5...+5V, -10...+10V).



**Hinweis:** Sollte eines Ihrer VIs nicht ausführbar sein (zu sehen am unterbrochenen Run-Pfeil), findet das VI wahrscheinlich die DLL nicht. Überprüfen Sie dann bitte zuerst, ob sich die **ADwin32.dll** in Ihrem Windows-Verzeichnis befindet. Sollte dies der Fall sein, dann öffnen Sie zunächst das fehlerhafte VI. Im Diagramm klicken Sie dann 2x auf den DLL-Aufruf. Danach öffnet sich das Fenster zum Editieren des Aufrufs. In der ersten Zeile finden Sie unter „*Library name or path*“ den Eintrag ADwin32.dll. Geben Sie hier nun den kompletten Pfad der ADwin32.dll an, (z.B. C:\Windows\ADwin32.dll) und speichern Sie das VI mit dieser Einstellung ab. Es sollte nun problemlos funktionieren.



### 4 Beispielprogramm „MinMax.VI“

Das MinMax.VI arbeitet mit der Binärdatei des BAS\_DMO1 Ihres **ADbasic** Beispielprogramms. Damit das Programm einwandfrei läuft, müssen Sie sicherstellen, daß der Name der Binärdatei, deren Pfad sowie die Prozeßnummer mit dem im Programm verwendeten Parameter übereinstimmt. Passen Sie gegebenenfalls diese Parameter in dem LabVIEW-Programm an.

#### 4.1 Programmaufbau

Auf der folgenden Seite wird das Icon, das Front-Panel, sowie das Diagramm dargestellt. Das Programm enthält eine eigene Bootsequenz, d.h. Sie müssen die Datei ADwin9.btl nicht gesondert auf Ihrem **ADwin**-System laden.

Icon:



Front-Panel:

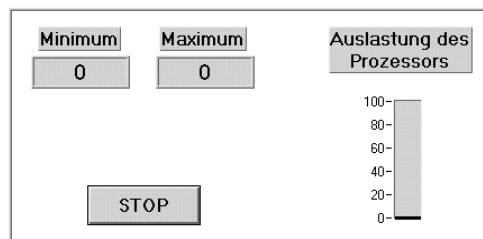
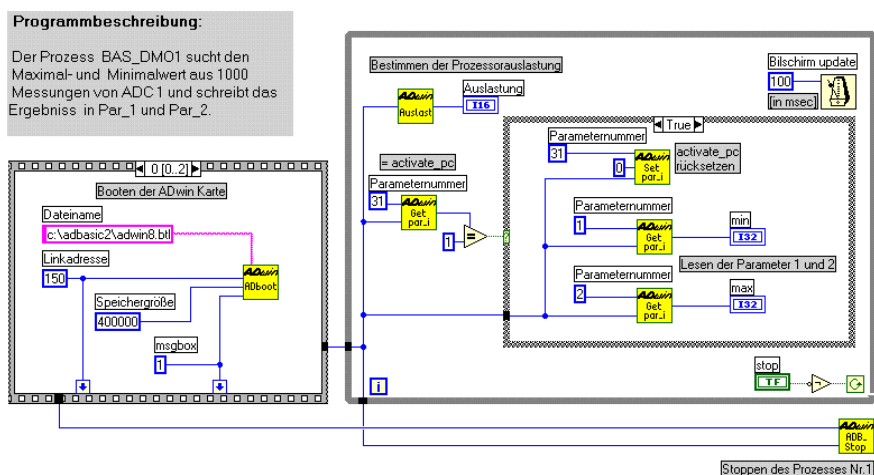


Diagramm:



## 5 Befehlsindex

Acquire.VI.....	15
ADbload.VI.....	9
ADboot.VI.....	9
ADbstart.VI.....	9
ADbstop.VI.....	9
Auslast.VI.....	12
Dout_bit.VI.....	13
Dout_val.VI.....	13
Get_adc.VI.....	13
GetDigin.VI.....	13
GetfData.VI.....	11
GetfFifo.VI.....	12
GetiData.VI.....	11
GetiFifo.VI.....	12
GetPar_f.VI.....	10
GetPar_i.VI.....	10
Memory.VI.....	12
Net_Connect.VI.....	14
Net_Disconnect.....	14
Set_dac.VI.....	13
SetfData.VI.....	11
SetiData.VI.....	11
SetPar_f.VI.....	10
SetPar_i.VI.....	10
StartDAC.VI.....	15
StopDAC.VI.....	15
V_to_Val.VI.....	16
Val_to_V.VI.....	16