








ADwin

TestPoint-Treiber



Inhaltsverzeichnis

1	BESONDERE FÄHIGKEITEN DER ADWIN-SYSTEME.....	3
2	SOFTWARE INSTALLATION	4
2.1	Installation der <i>ADwin</i> -Treiber	4
2.2	Laden des <i>ADwin</i> -Betriebssystems	5
2.3	Einbinden der <i>ADwin</i> Objekte.....	6
3	A/D OBJECT 	7
3.1	Start A/D	7
3.2	Stop A/D	7
3.3	Acquire A/D.....	8
3.4	Set A/D trigger analog (nur für Prozess 1).....	8
3.5	Set A/D trigger immediate (nur für Prozess 1).....	9
3.6	Sample A/D.....	9
3.7	Stop A/D	9
4	D/A OBJECT 	11
4.1	Start D/A	11
4.2	Stop D/A	11
4.3	Output D/A	12
5	DIO OBJECT 	13
5.1	Output to	13
5.2	Set bits of.....	13
6	PID OBJECT 	14
6.1	Start PID	14
6.2	Stop PID	14
6.3	Get PID	15
6.4	Set PID	15
6.5	Config PID	16
7	OSZI OBJECT 	17
7.1	Start Oszi Startet das schnelle Grafikfenster.....	17
7.2	Stop Oszi Stoppt und löscht das schnelle Grafikfenster.....	17
8	REALTIME OBJECT 	19
8.1	Start Prozess	19
8.2	Stop Prozess	19
8.3	Set Parameter.....	19
8.4	Read Parameter	20
8.5	Set Data	20
8.6	Read Data.....	20
8.7	Set Fifo	21
8.8	Read Fifo	21
8.9	Clear Fifo	21
8.10	Load Prozess.....	22
8.11	Data to File	22
9	SYSTEM OBJECT 	24
9.1	Boot	24
9.2	Auslastung	24
9.3	Memory.....	24
9.4	Net Connect.....	24
9.5	Net Disconnect	25
10	ADBASIC BEISPIELPROGRAMME :	26
10.1	Online-Auswertung von Meßwerten	26
10.2	Digitaler P-Regler	27
10.3	Beispiel zur Datenübertragung mit Data.....	28
10.4	Beispiel zur kontinuierlichen Datenübertragung mit Fifo	29
10.5	Beispiel PID-Regler	30

1 Besondere Fähigkeiten der ADwin-Systeme.

Die **ADwin** Regel- Steuer- und Mess-Systeme enthalten einen vollständigen Prozessrechner zur schnellen Regelung , Steuerung und Meßwerterfassung. Dadurch wird Testpoint zu einem sehr schnellen echtzeitfähigen Meßsystem, mit sicheren Antwortzeiten im Mikrosekundenbereich erweitert.

- Das **A/D-Object** wurde um eine **Pretriggerfunktion** erweitert, die auch den Signalverlauf vor dem Triggerzeitpunkt erfassen kann. Es können bis zu **vier A/D-Objects gleichzeitig** eingesetzt werden.
- Das **D/A-Object** erlaubt die Ausgabe von Signalverläufen mit bis zu 100 kHz Ausgabefrequenz. Es können **zwei D/A-Objects gleichzeitig** mit unterschiedlicher Ausgabefrequenz laufen.
- Das **PID-Object** ermöglicht digitale Regelungen mit Abtastraten bis zu 30 kHz. Während die Regelung läuft können die Regelabweichungen und die Stellwerte von Testpoint abgefragt werden. Es können bis zu **vier Regler parallel** mit unterschiedlichen Abtastfrequenzen auf den **ADwin**-Systemen laufen.
- Das **DIO-Object** beinhaltet Funktionen zur Ein- und Ausgabe von digitalen Werten über das **ADwin**-System.
- Das **REALTIME-Object** bietet Funktionen zum Datenaustausch zwischen TestPoint und Prozessen, die mit dem Echtzeit-Entwicklungstool ADbasic erstellt wurden.
- Das **SYSTEM-Object** erlaubt u. a. die Fernsteuerung u. -überwachung von **ADwin**-Systemen über ein Netzwerk (z. B. LAN, ISDN, Internet, ...).

Da die **ADwin**-Systeme einen eigenen Prozessor haben, der speziell für die schnelle Bearbeitung parallel laufender Prozesse entwickelt wurde, **können all diese Objecte auch gleichzeitig benutzt werden. Auf den ADwin-Systemen können gleichzeitig mehrere Regler, Signalgeneratoren, Zähler und getriggerte Meßwerterfassungen laufen.** Dadurch ist es möglich, mit dem **ADwin**-System komplette Anlagen zu steuern und zu überwachen.

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.

Andernfalls 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:

ADWIN2.BTL	Betriebssystem für die ADwin -Systeme mit T225-Prozessor
ADWIN4.BTL	Betriebssystem für die ADwin -Systeme mit T400-Prozessor
ADWIN5.BTL	Betriebssystem für die ADwin -Systeme mit T450-Prozessor
ADWIN8.BTL	Betriebssystem für die ADwin -Systeme mit T805-Prozessor
ADWIN9.BTL	Betriebssystem für die ADwin -Systeme mit ADSP-Prozessor
TESTVE16.EXE	zeigt die Version der installierten 16Bit- ADwin -DLLs an
TESTVE32.EXE	zeigt die Version der installierten 32Bit- ADwin -DLLs an
ADTEST.EXE	Programm zum Testen von ADwin -PC-Einsteckkarten und ADwin-GOLD
ADPRO.EXE	Programm zum Testen des ADwin-Pro -Systems
ADWINSET.EXE	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 TestPoint Action Boot oder über **ADbasic** geladen. Zusätzlich bietet auch das Testprogramm **ADtest** die Möglichkeit das System zu laden.

a) Unter TestPoint

Mit der Action Boot aus dem **ADwin** System Objekt. Achten Sie darauf, das die Settings des System-Objekts mit der Ausführung Ihres **ADwin**-Systems übereinstimmen. Nähere Informationen hierzu in Kapitel 3.



b) Mittels ADbasic

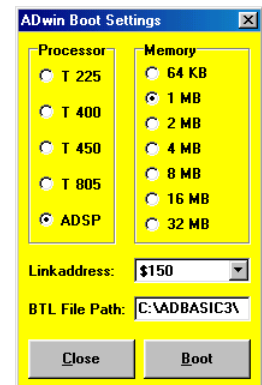
- ♦ Starten Sie **ADbasic** indem Sie im Windows-Menü Start ⇒ Programme ⇒ ADwin ⇒ **ADbasic 3.0** auswählen.
- ♦ Überprüfen Sie, ob die Einstellungen im **ADbasic**-Menü Options ⇒ Compiler mit Ihrem **ADwin**-System übereinstimmen.
- ♦ Klicken Sie in der **ADbasic** Toolbar auf das folgende Symbol:  (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 das **ADwin-GOLD** System. **ADtest** bietet ebenfalls die Möglichkeit das Betriebssystem zu laden. 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.

2.3 Einbinden der ADwin Objekte

Mit Hilfe der Actions aus den **ADwin** Objekten können Ihre TestPoint-Programme Befehle an das **ADwin**-Systeme schicken und Daten abholen bzw. senden.

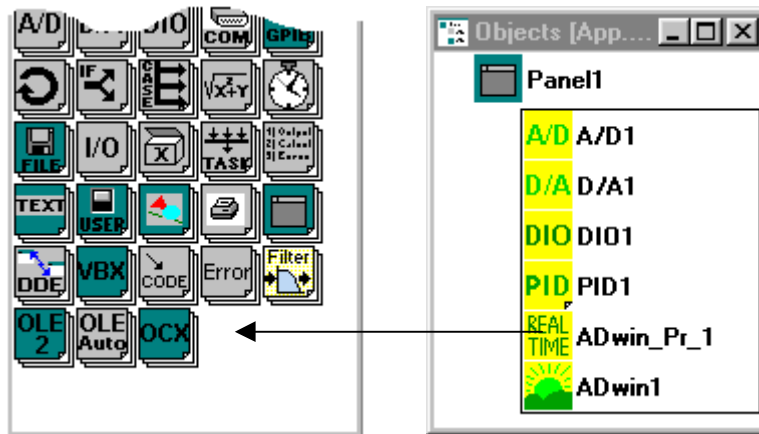
Die **ADwin** Objekte heben sich durch ihre grün/gelbe Farbgebung deutlich von den übrigen TestPoint Objekten ab und sind dadurch leicht zu erkennen.

Zum Einbinden der **ADwin** Objekte in den TestPoint Stock benötigen Sie die Datei:

ADwin.tst

Gehen Sie wie folgt vor:

- Laden Sie die Datei **ADwin.tst** in Testpoint. Im Objects Fenster erscheinen daraufhin alle **ADwin** Objekte.



- Jetzt können Sie jedes einzelne Objekt mit der Maus aus dem Objekt Fenster an einen freien Platz im Stock Fenster verschieben.

WICHTIG: Während Sie die linke Maustaste loslassen, um das verschobene Objekt im Stock zu platzieren, müssen Sie die '**Strg**' bzw. '**Ctrl**' Taste auf der Tastatur Ihres Rechners gedrückt halten.

- Wenn Sie TestPoint beenden, werden die neuen Objekte automatisch im Stock gespeichert. Sie können dann die **ADwin** Objekte wie jedes andere TestPoint Objekt in allen Applikationen verwenden.

3 A/D Object



Das A/D Object ermöglicht den Zugriff auf die analogen Eingänge des **ADwin**-Systems. Auf dem **ADwin**-System können bis zu vier A/D Prozesse gleichzeitig ablaufen. Jeder Prozess wird von einem eigenen A/D Object gesteuert. Die Nummer des Prozesses wird im Settings-Menue des A/D Objects festgelegt.

Actions

3.1 Start A/D

Die Action **Start A/D** startet die Datenerfassung auf einem oder mehreren analogen Eingangskanälen mit einer festgelegten Abtastrate. Die Messung läuft im Hintergrund ab. Das Programm arbeitet die nächsten Zeilen der Action List ab, ohne auf das Ende der Messreihe zu warten. Deshalb kann auf die Meßwerte erst in der Action-List des A/D-Objects zugegriffen werden und nicht in den folgenden Zeilen der Action-List, in der die Start-Action steht. **Das A/D-Object kann auch gleichzeitig mit getakteten analogen Ausgaben (mit dem D/A Object) und dem digitalen Regler betrieben werden.** Auf den **ADwin**-Systemen können mehrere Prozesse mit unterschiedlichen Taktraten parallel ablaufen.

Start A/D A/D1 #samples=1000 ▾, rate=1000 Hz, channel(s)=1 ▾, event after "ALL" ▾ sample(s)

#samples - Anzahl der Messungen, die pro Kanal ausgeführt werden soll
Minimalwert = 2 **Maximalwert** = 500000 oder CONTINUOUS (falls die Messung erst mit dem Stop A/D Befehl angehalten werden soll)

rate - Abtastrate in Hz.
Minimalwert = 1 Hz **Maximalwert** = 60 kHz

channel(s) - A/D-Kanäle, die abgetastet werden sollen
 Wenn nur auf einem Kanal gemessen werden soll, ist die Nummer dieses Kanals einzusetzen (die Kanalnummern beginnen bei 1 !). Falls mehrere Kanäle gemessen werden sollen, sind die Nummern durch Kommas getrennt einzugeben.
Minimalwert = 1 **Maximalwert** = 12

event after - Anzahl der Messungen, nach der die Action-List des A/D-Objects ausgeführt werden soll
 Bei schnellen Messungen ist es meist sinnvoll, die Action-List erst am Ende der Meßreihe auszuführen.
Minimalwert = 1 **Maximalwert** = 500000 oder ALL

3.2 Stop A/D

Die Action **Stop A/D** hält die analoge Meßreihe an.

Stop A/D A/D1

3.3 Acquire A/D

Die Action **AcquireA/D** startet die Datenerfassung auf einem oder mehreren analogen Eingangskanälen mit einer festgelegten Abtastrate und **wartet auf das Ende der Messung. Die restliche Action-List wird erst ausgeführt wenn die Meßdaten erfaßt sind. Das Acquire-Object kann auch gleichzeitig mit getakteten analogen Ausgaben (mit dem D/A Object) und dem digitalen Regler betrieben werden.** Auf den **ADwin**-Systemen können verschiedene Prozesse mit unterschiedlichen Taktraten parallel ablaufen.

Acquire A/D A/D1 #samples=**1000** , rate=**1000** Hz, channel(s)=**1**

#samples - Anzahl der Messungen, die pro Kanal ausgeführt werden soll
Minimalwert = 2 **Maximalwert** = 4000

rate - Abtastrate in Hz.
Minimalwert = 1 Hz **Maximalwert** = 30 kHz

channel(s) - A/D-Kanäle, die abgetastet werden sollen
 Wenn nur auf einem Kanal gemessen werden soll, ist die Nummer dieses Kanals einzusetzen (die Kanalnummern beginnen bei 1 !). Falls mehrere Kanäle gemessen werden sollen, sind die Nummern durch Kommas getrennt einzugeben.
Minimalwert = 1 **Maximalwert** = 12

3.4 Set A/D trigger analog (nur für Prozess 1)

Die Action **Set A/D trigger analog** setzt den Triggermode, der bei der nächsten Start-A/D Action benutzt werden soll, auf analog. Es muß festgelegt werden, welcher Eingangskanal triggern soll, wie hoch die Schwelle ist, und wie lange vor dem Trigger gemessen werden soll.

Diese Funktion ist nur wirksam, wenn im Setting des A/D Objects die Prozessnummer 1 gewählt wurde. Für alle anderen Prozessnummern ist der Triggermode unveränderbar auf immediate festgelegt.

Set A/D trigger analog A/D1 channel=, level=, polarity=, pretrigger= mS

channel - analoger Eingangskanal der triggern soll
 Dieser Kanal muß auch in der Liste der zu messenden Kanäle stehen.
Minimalwert = 1 **Maximalwert** = 12

level - Triggerschwelle, die unter bzw. überschritten werden muß
Minimalwert = -10 V **Maximalwert** = 10 V

polarity - **Positive level:** Triggert, falls der gemessene Wert größer ist als unter **level** angegeben
Negative level: Triggert, falls der gemessene Wert kleiner ist als unter **level** angegeben
Rising Slope: Triggert auf die Überschreitung des unter **level** angegebenen Wertes
Falling Slope: Triggert auf die Unterschreitung des unter **level** angegebenen Wertes

pretrigger - Meßdauer in Millisekunden, die vor dem Triggersignal angezeigt wird
Minimalwert = 0.0 **Maximalwert** = gesamte Meßdauer

3.5 Set A/D trigger immediate (nur für Prozess 1)

Die Action **Set A/D trigger immediate** setzt den Triggermode, der bei der nächsten Start-A/D Action benutzt werden soll, auf immediate. Die Messung beginnt sofort nach der Start-A/D Action, ohne auf irgendwelche Triggerschwellen zu warten.

Set A/D trigger immediate A/D1

3.6 Sample A/D

Die Action **Sample A/D** führt eine Einzelmessung auf einem oder mehreren analogen Eingangskanälen aus. Das Meßergebnis wird im Datenbuffer des A/D-Objects übergeben. **Laufende Messungen, Analogausgaben oder Regelungen werden von dieser Messung nicht abgebrochen oder gestört.**

Sample A/D A/D1 **once, channel(s)=** 

channel(s) - A/D-Kanäle, die gemessen werden sollen

Soll nur auf einem Kanal gemessen werden, so ist die Nummer dieses Kanals einzusetzen (die Kanalnummern beginnen bei 1 !). Falls mehrere Kanäle benutzt werden sollen, sind die Nummern durch Kommas getrennt einzugeben.

Minimalwert = 1

Maximalwert = 12

3.7 Stop A/D

Die Action **Stop A/D** hält die analoge Meßreihe an.

Stop A/D A/D1

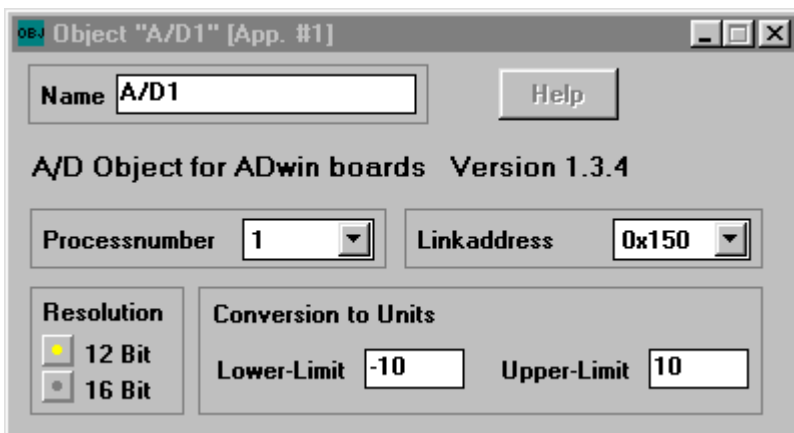
Data

Die Daten des A/D Objects sind die Ergebnisse der letzten Messungen. Falls zuletzt eine Sample Action ausgeführt wurde, ist im Datenbuffer eine Zahl oder eine Liste von Zahlen (falls mehrere Eingänge gemessen wurden). Falls eine Start A/D Action ausgeführt wurde, steht im Datenbuffer für jeden ausgewählten Eingangskanal ein Vektor mit der bei Event after angegebenen Länge in einer Liste.

Action List

Die Action wird ausgeführt, wenn eine Meßreihe mit dem Start A/D Befehl gestartet und die unter event after gewählte Anzahl von Messungen ausgeführt wurde. Bei jedem Aufruf der Action List stehen im Datenbuffer die neu gemessenen Meßwerte.

Settings



Prozessnummer Auf dem **ADwin**-System können völlig unabhängig voneinander vier A/D Prozesse parallel ablaufen. Anhand der Prozessnummer wird festgelegt, welcher Prozess vom A/D Objekt angesprochen wird.

Linkaddress Die Linkadresse, die auf dem **ADwin**-System eingestellt ist. Durch die variable Linkadresse können gleichzeitig mehrere **ADwin**-Systeme in einem Rechner betrieben werden.

Resolution Die Auflösung der A/D-Wandler Ihres **ADwin**-Systems. Stellen Sie 12 Bit für **ADwin-PC** Karten und 16 Bit für das **ADwin-GOLD** System ein.

Lower-Limit, Upper-Limit

Alle vom ADC gemessenen Werte werden unter Zuhilfenahme von Lower- und Upper-Limit in den Bereich umgerechnet, der durch diese beiden Werte vorgegeben wird. Um die an den analogen Eingängen anliegenden Spannungswerte zu erhalten, müssen für Lower- und Upper-Limit die Grenzwerte des Eingangsspannungsbereiches eingegeben werden, der auf dem ADwin-Systemen eingestellt ist.

Defaultmäßig ist der Eingangsspannungsbereich der DACs auf -10V bis +10V eingestellt. Damit TestPoint die gemessenen Spannungswerte in diesem Fall richtig anzeigt, müssen die Werte für Lower- und Upper-Limit auf -10 und +10 gesetzt werden.

4 D/A Object



Das D/A Object dient zur Ausgabe von Spannungen auf den analogen Ausgängen des **ADwin**-Systems. Auf dem **ADwin**-System können zwei D/A Prozesse gleichzeitig ablaufen. Jeder Prozess wird von einem eigenen D/A Object gesteuert. Die Nummer des Prozesses wird im Settings-Menue des D/A Objects festgelegt.

Actions

4.1 Start D/A

Die Action **Start D/A** startet die Ausgabe von Werten auf einem analogen Ausgabekanal mit einer vorgegebenen Ausgaberate. **Diese Funktion kann auch gleichzeitig mit getakteten Messungen (mit dem A/D Object) und mit Regelungen (mit dem PID Object) betrieben werden.**

Auf den **ADwin**-Systemen können verschiedene Prozesse mit unterschiedlichen Taktraten parallel ablaufen.

Start D/A D/A1 rate=10000 Hz, channel=1, values=rampe, repeat=20

- rate** - Taktrate in Hz
Minimalwert = 1 Hz Maximalwert = 100 kHz
- channel** - D/A-Kanal auf dem die Werte ausgegeben werden sollen
Minimalwert = 1 Maximalwert = 6
- values** - Spannungen (in Volt), die ausgegeben werden sollen
Die auszugebenden Werte müssen in einem Vektor stehen.
Minimalwert = -10 V Maximalwert = 10 V
Minimalvektorlänge = 1 Maximalvektorlänge = 1000000
- repeat** - Gibt an, wie oft die Ausgabe der gesamten Datenreihe wiederholt werden soll
Minimalwert = 1 Maximalwert = 1000000000

4.2 Stop D/A

Hält die laufende analoge Ausgabe an.

Stop D/A D/A1

4.3 Output D/A

Die Action **Output D/A** gibt einen analogen Wert auf einem analogen Ausgabekanal sofort aus. **Laufende Messungen, Analogausgaben oder Regelungen werden von dieser Messung nicht abgebrochen oder gestört.**

Output D/A D/A1 `once,channel=1,value=0`

channel - D/A-Kanal, auf dem der Werte ausgegeben werden soll
Minimalwert = 1 **Maximalwert** = 6

value - Spannung (in Volt), die ausgegeben werden soll
Minimalwert = -10 V **Maximalwert** = 10 V

Settings



Prozessnummer Auf dem **ADwin**-System können völlig unabhängig voneinander zwei D/A Prozesse parallel ablaufen. Anhand der Prozessnummer wird festgelegt, welcher Prozess vom D/A Objekt angesprochen wird.

Linkaddress Die Linkadresse, die auf dem **ADwin**-System eingestellt ist. Durch die variable Linkadresse können gleichzeitig mehrere **ADwin**-Systemen in einem Rechner betrieben werden.

Resolution Die Auflösung der D/A-Wandler Ihres **ADwin**-Systems. Stellen Sie 12 Bit für **ADwin-PC** Karten und 16 Bit für das **ADwin-GOLD** System ein.

5 DIO Object



Das DIO Object ermöglicht digitale Ein- und Ausgaben mit dem **ADwin**-System. Die **ADwin**-Karte hat je 16 digitale Ein- und Ausgänge, die jeweils in einem 16-Bit Wort zusammengefaßt sind.

Actions

5.1 Output to

Die Action **Output to** setzt die digitalen Ausgänge auf die gewünschten Werte.

Output to **DIO1** **value=**

Value Dezimales Ausgabewort (0 = Lowpegel, 1 = Highpegel)

Input from

Die Action **Input from** liest die digitalen Eingänge ein

Input from **DIO1**

5.2 Set bits of

Die Action **Set bits of** setzt die digitalen Ausgänge auf die gewünschten Werte. Dabei werden nur die Bits verändert, die im Mask-Parameter auf 1 gesetzt sind, alle anderen Bits bleiben unverändert.

Set bits of **DIO1** **value=** , **mask=**

Value Dezimales Ausgabewort (0 = Lowpegel, 1 = Highpegel)

mask Dezimale Ausgangemaske (0 = nicht ändern, 1 = ändern)

6 PID Object



Das PID Object ermöglicht schnelle digitale Regelungen mit der **ADwin**-Karte.

Auf der **ADwin**-Karte können bis zu vier Regelprozesse gleichzeitig laufen. Jeder Prozess wird von einem eigenen PID Object gesteuert. Die Nummer des Prozesses wird im Settings-Menue des PID Objects festgelegt.

Actions

6.1 Start PID

Die Action **Start PID** startet einen PID-Regler auf der **ADwin**-Karte. Die Reglerparameter werden vorgegeben. Der Regler protokolliert die Regelabweichungen und die Stellwerte der ersten Regelzyklen in einen Buffer. Wenn der Buffer voll ist (die Defaultgröße des Buffers beträgt 256 Werte), wird die Action-List des PID-Objects ausgeführt. Die protokollierten Werte stehen als Liste von 2 Vektoren im Datenbuffer des PID-Objects. **Der PID-Regler kann auch gleichzeitig mit getakteten Messungen (mit dem A/D Object) und mit getakteten analogen Ausgaben(mit dem D/A Object) betrieben werden.**

Auf den **ADwin**-Karten können mehrere Prozesse mit unterschiedlichen Taktraten parallel ablaufen.

Start PID PID Sollwert=**0** V, k=**5** , t0=**1** mS, td=**0.2** mS, ti=**50** mS

- Sollwert** - Sollwert für den PID-Regler in Volt
 Minimalwert = -10 V **Maximalwert** = 10 V
- k** - Verstärkungsfaktor für den PID-Regler
 Minimalwert = 0.1 **Maximalwert** = 1000
- t0** - Zyklusdauer für den PID-Regler in Millisekunden
 Minimalwert = 0.033 mS **Maximalwert** = 1000 mS
- td** - Differentiationszeit für den PID-Regler in Millisekunden
 Minimalwert = 0.001 **Maximalwert** = 1000000
- ti** - Integrationszeit für den PID-Regler in Millisekunden
 Minimalwert = 0.01 **Maximalwert** = 1000000

6.2 Stop PID

Die Action **Stop PID** Hält den laufenden PID-Regler an.

Stop PID PID1

6.3 Get PID

Nach der **Get-PID** Action wird der Protokollbuffer gelöscht und mit aktuellen Werten aufgefüllt. Wenn der Buffer voll ist, wird die Action-List des PID-Objects ausgeführt.

Get PID PID1

6.4 Set PID

Die Action **Set PID** gibt neue Regelparameter für einen gerade laufenden PID-Regler vor. Nach der Änderung der Parameter protokolliert der Regler die Regelabweichungen und die Stellwerte der ersten Regelzyklen in einen Buffer. Wenn der Buffer voll ist (die Defaultgröße des Buffers beträgt 256 Werte), wird die Action-List des PID-Objects ausgeführt. Die protokollierten Werte stehen als Liste von 2 Vektoren im Datenbuffer des PID-Objects.

Set PID PID Sollwert=0 V, k=5, t0=1 mS, td=0.2 mS, ti=50 mS

- Sollwert** - Sollwert für den PID-Regler in Volt
Minimalwert = -10 V **Maximalwert** = 10 V
- k** - Verstärkungsfaktor für den PID-Regler
Minimalwert = 0.1 **Maximalwert** = 1000
- t0** - Zyklusdauer für den PID-Regler in Millisekunden
Minimalwert = 0.033 mS **Maximalwert** = 1000 mS
- td** - Differentiationszeit für den PID-Regler in Millisekunden
Minimalwert = 0.001 **Maximalwert** = 1000
- ti** - Integrationszeit für den PID-Regler in Millisekunden
Minimalwert = 0.01 **Maximalwert** = 1000

6.5 Config PID

Die Action **Config PID** ermöglicht die Festlegung von Messeingang, Stellausgang und Buffergröße des PID-Reglers. Diese Funktion wird nur ausgeführt, wenn der Regler nicht läuft.

Config PID **PID1** **Messeingang=** , **Stellausgang=** , **Buffergroesse=**

Messeingang - Nummer des Eingangskanals, den der Regler benutzen soll
Minimalwert = 1 **Maximalwert** = 12

Stellausgang - Nummer des Ausgangskanals, den der Regler benutzen soll
Minimalwert = 1 **Maximalwert** = 6

Buffergroesse - Länge des Buffers zur Protokollierung der Regelverhaltens
Minimalwert = 8 **Maximalwert** = 500000

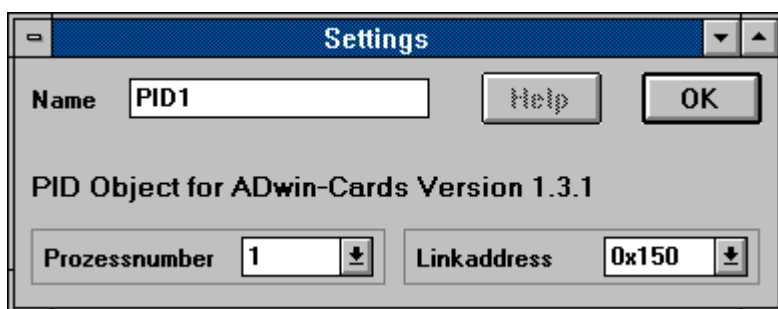
Data

Die Daten des PID-Objects sind die protokollierten Regelabweichungen und Stellwerte nach der zuletzt ausgeführten Start-, Set- oder Get-Anweisung. Die Daten stehen in einer Liste aus zwei Vektoren mit der Länge des Buffers.

Action List

Die Action List wird nach jeder Start-, Set-, oder Getanweisung ausgeführt, sobald der Buffer mit Protokollwerten gefüllt ist.

Settings



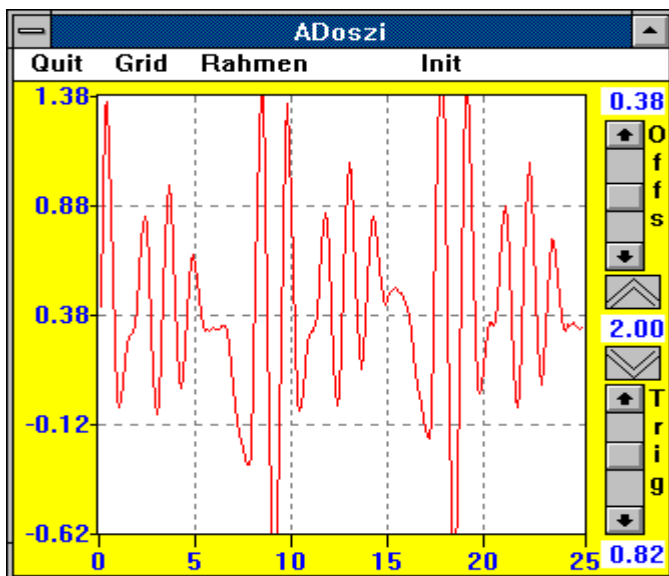
Prozessnummer Auf der **ADwin** Karte können völlig unabhängig voneinander zwei D/A Prozesse parallel ablaufen. Anhand der Prozessnummer wird festgelegt, welcher Prozess vom D/A Objekt angesprochen wird.

Linkaddress Die Linkadresse, die auf der **ADwin**-Karte eingestellt ist. Durch die variable Linkadresse können gleichzeitig mehrere **ADwin**-Karten in einem Rechner betrieben werden.

7 Oszi Object*



Das Oszi Object enthält ein schnelles Grafikfenster für **ADwin-4**, **ADwin-8** und **ADwin-9** Karten.



Das Oszi-Object ermöglicht eine kontinuierliche, verzögerungsfreie Darstellung von bis zu zwei analogen Eingängen ohne störendes Flackern beim Bildaufbau.

Das Oszi-Object hat flankengesteuerte Triggerfunktionen mit einstellbarem Pretriggeranteil.

Das Oszi-Object kann mit **ADwin-8** und **ADwin-9** Karten auch schnelle Online Frequenzspektren berechnen und darstellen. (Frequenzbereich bis 15 KHz, Updaterate 10 Hz)

Das Oszi-Object kann gleichzeitig mit anderen Messungen unter Testpoint auf der gleichen **ADwin**-Karte betrieben werden. z. B. :

- Signalerzeugung mit D/A-Object und Messung mit Oszi-Object
- langsame Messung mit AD-Object und schnelle Kontrollanzeige mit Oszi-Object
- Regelung mit PID-Object und Anzeige der Abweichung mit Oszi-Object

Actions

7.1 Start Oszi Startet das schnelle Grafikfenster

7.2 Stop Oszi Stoppt und löscht das schnelle Grafikfenster

* Das Oszi-Object ist standardmäßig nicht im Lieferumfang enthalten. Es kann optional für 600,- DM + MWSt erworben werden.

Installation der ADoszi-Treiber

Sie benötigen dazu die Dateien:

- **ADwin8.btl** Version 1.1
- **ADoszi.dll**
- **ADoszis.exe**
- **ADpoint.dll**
- **iserver.exe**
- **run.bat**

Die neueste Version des Programms **ADwin8.btl** muß immer vor dem Starten von Testpoint oder einer Testpoint-Anwendung mit der Anweisung **run ADwin8** auf die **ADwin**-Karte geladen werden. Wenn das Programm richtig geladen wurde, gibt es die Meldung **ADwin8 V1.3.1 started** ! aus.

Kopieren Sie die Dateien **ADoszi.dll**, **ADoszis.exe** und **ADpoint.dll** in Ihr Windows-Subdirectory.

Laden Sie die Datei **ADwin.tst** in Testpoint und kopieren Sie das User defined object **Oszi**, in den Stock.

Zum Testen der Funktion laden und starten Sie das Programm udo_oszi.tst.

INIT

Zeit
Messzeit: 25 ms
Updatezeit: 100 ms

Eingänge
KANAL 1: 1
KANAL 2: 0

FFT
☐ An ☒ Aus
max. Freq: 5000 Hz
☐ Log ☒ Linear

Trigger
☐ Aus ☒ Steigende Flanke ☐ Fallende Flanke
Pretrigger: 0 ms

Cancel OK

8 REALTIME Object



Das REALTIME-Object ermöglicht den Datenaustausch mit schnelle Echtzeitprozessen, die mit **ADbasic** geschrieben wurden. Auf dem **ADwin**-System können bis zu 10 Echtzeitprozesse gleichzeitig ablaufen. Jeder Prozess wird von einem eigenen Realtime-Object gesteuert. Die Nummer des zugehörigen Prozesses wird im Settings-Menue des Realtime Objects festgelegt. Die mit **ADbasic** entwickelten Prozesse können in kompilierter Form abgespeichert werden und dann mit der Action Load Prozess auf das **ADwin**-System geladen werden.

Actions

8.1 Start Prozess

Die Action **Start Prozess** startet auf dem **ADwin**-System einen Prozess, der mit **ADbasic** geschrieben wurde. Auf den **ADwin**-Systemen können bis zu 10 **ADbasic**-Prozesse (die mit 10 verschiedenen Realtime-Objects gestartet werden) gleichzeitig ablaufen.

Start Prozess ADwin1

8.2 Stop Prozess

Die Action **Stop Prozess** stoppt einen Prozess auf dem **ADwin**-System.

Stop Prozess ADwin_Pr_1

8.3 Set Parameter

Die Action **Set Parameter** setzt einen Parameter für einen mit **ADbasic** geschriebenen Prozess. Es stehen 80 Parameter zur Verfügung, die von 1 bis 80 durchnummeriert sind. Jeder Parameter ist eine long-Integer Variable. In **ADbasic** kann man mit Hilfe der Variablen Par_1 bis Par_80 auf die Parameter zugreifen.

Auf dem **ADwin-8** und **ADwin-9** System gibt es zusätzlich 80 Floating-Point Parameter mit den Namen Fpar_1 bis Fpar_80, auf die man von Testpoint aus zugreifen kann, wenn Type = Float eingestellt wird.

Set Parameter ADwin_Pr_1 Parameter-Nr = , Value = , Type = **"Integer"**

Parameter-Nr Nummer des zu setzenden Parameters. Über die Parameter-Nr Pull-Down Liste können auch das Delay zwischen 2 vom internen Prozessortimer erzeugten Events (**Delay**) und die max. Anzahl der Durchläufe (**nSteps**) gesetzt werden. Die Einstellung für **nSteps** wird vom ADSP nicht beachtet

Value zu übergebender 32-Bit Integerwert

Type Parameter-Typ Integer = Par, Float = Fpar

8.4 Read Parameter

Die Action **Read Parameter** liest einen Parameter der von einem **ADbasic**-Prozess gesetzt wurde. Es stehen 80 Parameter zur Verfügung, die von 1 bis 80 durchnummeriert sind. Jeder Parameter ist eine long-Integer Variable. In **ADbasic** können die Parameter mit Hilfe der Variablen Par_1 bis Par_80 gesetzt werden.

Auf dem **ADwin-8** und **ADwin-9** System gibt es zusätzlich 80 Floating-Point Parameter mit den Namen Fpar_1 bis Fpar_80, auf die man von Testpoint aus zugreifen kann, wenn Type = Float eingestellt wird.

Read Parameter ADwin_Pr_1 Parameter-Nr = 1, Type = "Integer" 

Parameter-Nr Nummer des zu lesenden Parameters

Type Parameter-Typ Integer = Par, Float = Fpar

8.5 Set Data

Die Action **Set Data** schickt einen Datensatz an das **ADwin**-System. Es stehen 200 Datensätze zur Verfügung die von 1 bis 200 durchnummeriert sind. Die Datensätze müssen vor dem Beschreiben von einem **ADbasic**-Prozess mindestens mit der hier benutzten Länge definiert worden sein. In **ADbasic** kann man mit Hilfe des Arrays "Data" auf diese Datensätze zugegriffen werden.

Set Data ADwin_Pr_1 Dataset-Nr = 1, length = , values =

Dataset-Nr Nummer des zu setzenden Datensatzes

length - Länge des Datensatzes

Value- zu übergebender Datensatz

8.6 Read Data

Die Action **Read Data** liest einen Datensatz der von einem **ADbasic**-Prozess geschrieben wurde. Es stehen 200 Datensätze zur Verfügung die von 1 bis 200 durchnummeriert sind. In **ADbasic** können diese Datensätze mit Hilfe des Arrays DATA beschrieben werden.

Read Data ADwin_Pr_1 Dataset-Nr. = 1, length = , start at 1


Dataset-Nr Nummer des zu lesenden Datensatzes

length - Länge des Datensatzes

Start at Startindex beim Lesen

8.7 Set Fifo

Um die kontinuierliche Übertragung von Daten zu vereinfachen können die Data-Strukturen auch als Fifo-Speicher organisiert werden. Bei wiederholtem Schreiben in den gleichen Datensatz werden die neuen Daten dabei automatisch immer hinten angehängt. Die benutzten Datensätze müssen vor dem Beschreiben von einem **ADbasic**-Prozess als Fifo definiert worden sein. Die Funktion „Set Fifo“ hat als Rückgabewert die Anzahl der noch freien Speicherplätze im Fifo.

Set Fifo **ADwin_Pr_1** **Dataset-Nr. = 1** , **length = 0** , **values =** 

Dataset-Nr Nummer des zu setzenden Datensatzes

length - Länge des anzuhängenden Datensatzes. Bei Länge =0 wird nichts geschrieben, aber die Anzahl der freien Speicherplätze zurückgegeben, so daß man dies Funktion auch zum Abfragen des freien Speicherplatzes im Fifo benutzen kann.

Values - anzuhängender Datensatz

8.8 Read Fifo

Um die kontinuierliche Übertragung von Daten zu vereinfachen können die Data-Strukturen auch als Fifo-Speicher organisiert werden. Bei wiederholtem Lesen des gleichen Datensatzes werden immer die nächsten Daten übergeben, die von **ADbasic** genauso kontinuierlich aufgefüllt werden können. Die benutzten Datensätze müssen vor dem Lesen von einem **ADbasic**-Prozess als Fifo definiert worden sein .

Read Fifo **ADwin_Pr_1** **Dataset-Nr. = 1** , **length = "all"** 

Dataset-Nr Nummer des zu lesenden Datensatzes

length - Anzahl der zu lesenden Daten. Beim Eintrag „all“ werden alle abholbereiten werte gelesen. Sind im Fifo weniger Daten enthalten, als angefordert wurden, so werden nur so viele Werte geliefert wie vorhanden waren.

8.9 Clear Fifo

Die Funktion „Clear Fifo“ löscht den Inhalt eines Fifo-Datensatzes. Dieser Befehl sollte vor Beginn einer Messung aufgerufen werden um eventuell vorhandene alte Werte aus dem Fifo zu löschen..

Clear Fifo **ADwin_Pr_1** **Dataset-Nr. = 1** 

Dataset-Nr Nummer des zu löschenden Datensatzes

8.10 Load Prozess

Mit der Funktion „Load Prozess“ kann man einen bereits compilierten ADbasic-Prozess laden, ohne daß ADbasic benötigt wird. Compilierte Prozesse können unter **ADbasic** mit der Funktion: 'Make Bin File' aus dem Menü: 'Project' erstellt werden. (Siehe auch **ADbasic** Handbuch)

Load Prozess **ADwin_Pr_1** **Prozess-Name =**

Prozess-Name Der Dateiname des zu ladenden ADbasic Prozesses.
Der Name muß vollständig mit Pfad und der Extension „.T81“ oder
„.T82“, „.T83“ beim Prozess Nr. 2 oder 3 angegeben werden.

8.11 Data to File

Die Funktion „Data to File“ speichert die Daten aus einem ADbasic Datensatz direkt in binärer Form auf der Festplatte. Dadurch können je nach Rechner und Betriebssystem 150-200 kByte/s gespeichert werden.

Data to file **ADwin_Pr_1** **File-Name =** "C:\TEST.DAT", **Dataset-Nr. =** , **length =** , **start at** , **mode =** "append" ▼

File-Name Der Dateiname unter dem die Daten gespeichert werden.
Der Name muß vollständig mit Pfad und Extension angegeben werden.

Dataset-Nr Nummer des zu speichernden Datensatzes

length - Anzahl der Datensatz-Elemente die gespeichert werden.

Start at Index des ersten Elementes das gespeichert wird.

mode Speichermodus: append = Daten werden an eine bereits existierende Datei angehängt.
 overwrite = Gleichnamige Datei wird gelöscht.

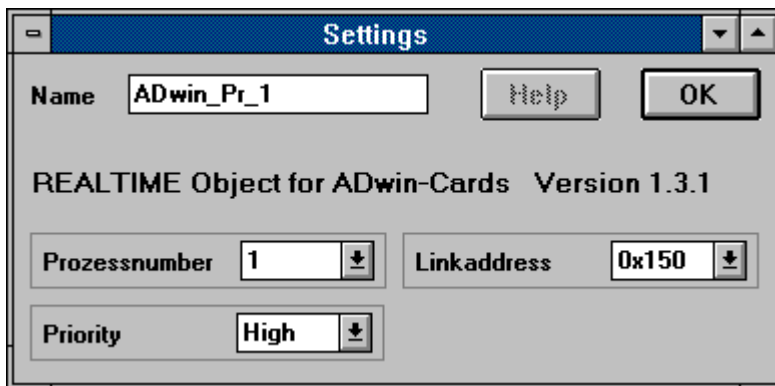
Action List

Die Action List wird ausgeführt, wenn im ADbasic-Programm die Funktion "**activate_pc**" aufgerufen wird.



Die Action List wird nur dann von TestPoint ausgeführt, wenn der auslösende **ADbasic** Prozess von TestPoint aus mit der Action: **Start Process** gestartet wurde. Falls der **ADbasic** Prozess von einem anderen Programm oder von einem anderen **ADbasic** Prozess gestartet wird, kann TestPoint nicht auf die activate_pc Funktion reagieren.

Settings



Prozessnummer Die Nummer des ADbasic-Prozesses der angesprochen werden soll (1 - 10). Falls gleichzeitig zwei oder mehr ADbasic-Prozesse auf dem **ADwin**-System laufen, werden auch zwei oder mehr REALTIME Objects benötigt. Die Actions: Load Prozess, Start Prozess und Stop Prozess wirken sich nur auf den ADbasic-Prozess aus, der hier eingestellt ist.

Linkaddress Die Linkadresse, die auf dem **ADwin**-System eingestellt ist. Durch die variable Linkadresse können gleichzeitig mehrere **ADwin**-Systeme in einem Rechner betrieben werden.

Priority Die Priorität des Prozesses. Diese Einstellung hat nur Buchführungswert, da die Prozesse immer mit der Priorität ablaufen, die beim Erzeugen der Binärdatei unter ADbasic eingestellt war.

9 System Object



Mit dem **ADwin**-System Object wird der Prozessor auf dem **ADwin**-System initialisiert und das Betriebssystem auf das System geladen. Außerdem kann die Auslastung (in %) und der freie Speicher des **ADwin**-Systems abgefragt werden. Zusammen mit dem Echtzeit-Entwicklungstool **ADbasic** kann mittels des System Objects - über ein Netzwerk - eine Verbindung zu einem **ADwin**-System in einem anderen Rechner hergestellt werden.

Actions

9.1 Boot

Die Action **Boot** initialisiert das **ADwin**-System und lädt das Betriebssystem, das die Grundlage zur Kommunikation zwischen TestPoint und **ADwin** bildet.

9.2 Auslastung

Die Action **Auslastung** liefert die aktuelle Prozessorauslastung des **ADwin**-Systems. Der ermittelte Wert liegt zwischen 0 und 100 Prozent.

9.3 Memory

Die Action **Memory** gibt die Größe des freien Speichers auf dem **ADwin**-System zurück.

9.4 Net Connect

Diese Action stellt über ein Netzwerk (z. B. LAN, ISDN, Internet, ...) eine Verbindung zu einem **ADwin**-System in einem anderen Rechner (Server) her. Nach erfolgreichem Verbindungsaufbau werden alle TestPoint Funktionen, welche das **ADwin**-System betreffen, über das Netzwerk zum System im Server weitergeleitet.

Bevor diese Action aufgerufen wird, muß auf dem Server das Programm ADserver gestartet und aktiviert werden, welches im Lieferumfang des Echtzeit-Entwicklungstools **ADbasic** enthalten ist.

Net Connect ADwin1 Protocol="TCP over IP" ☒ Address = "PC_Stefan" Endpoint = 200 Password =

Protocol Netzwerk Protokoll. Das gewählte Protokoll muß auf Ihrem Rechner ordnungsgemäß installiert und mit der Einstellung im Programm ADserver identisch sein.

Address Adresse bzw. Kennung des Rechners, zu dem die Verbindung aufgebaut wird

Endpoint Endpunkt für die Netzwerk Kommunikation.
Muß mit der Einstellung im Programm ADserver identisch sein.

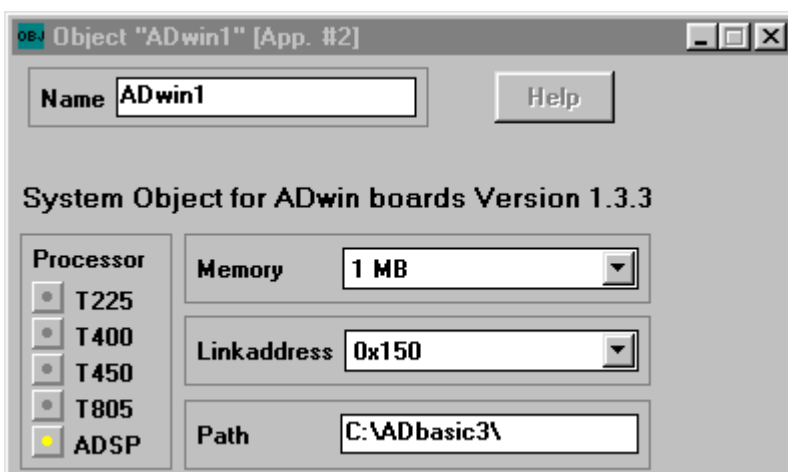
Password Falls im Programm ADserver ein Paßwort angegeben wurde, muß das selbe Paßwort auch hier eingetragen werden. Es wird auf Groß- und Kleinschreibung geachtet.

Die Funktion: 'NetConnect' wird nur von Windows 95, 98 und Windows NT unterstützt !

9.5 Net Disconnect

Beendet die Netzwerk Verbindung zu einem **ADwin**-System in einem anderen Rechner.

Settings



Prozessor Prozessor auf dem verwendeten **ADwin**-System.

Memory Speichergröße dem verwendeten **ADwin**-System.

Linkaddress Die Linkadresse, die auf dem **ADwin**-System eingestellt ist. Durch die variable Linkadresse können gleichzeitig mehrere **ADwin**-Systeme in einem Rechner betrieben werden.

Path Der Pfad, unter dem die Treiberdatei (je nach Prozessor: adwin2.btl, adwin4.btl, adwin8.btl oder adwin9.btl) steht.

10 ADbasic Beispielprogramme :

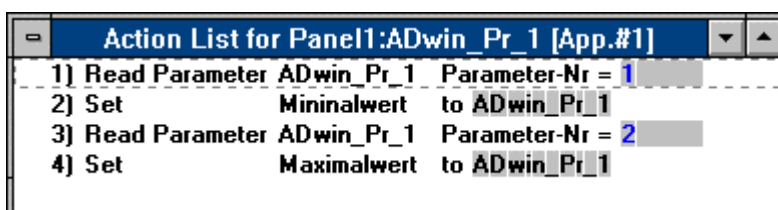
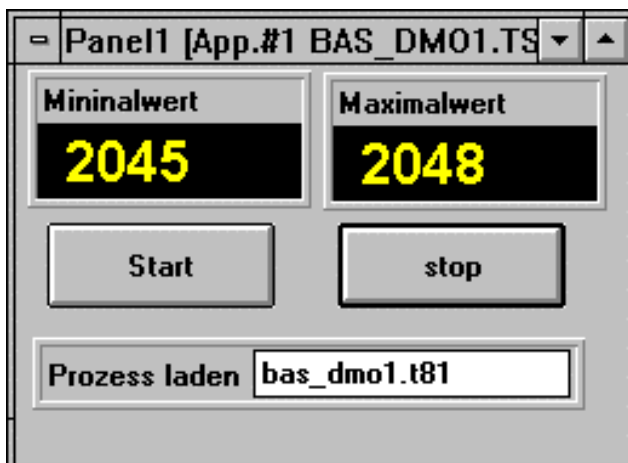
10.1 Online-Auswertung von Meßwerten

```
REM Das Programm BAS_DMO1 sucht den Maximal-
rem und Minimalwert aus 1000 Messungen von ADC 1
rem und schreibt das Ergebnis in par_1 und Par2

dim il, iw, max, min as integer

init:
  il = 1
  max = 0
  min = 4095

event:
  iw = adc(1)
  if (iw>max) then max = iw
  if (iw<min) then min = iw
  il = il+1
  if (il>1000) then
    il = 1
    par_1 = min : rem Minimalwert in Parameter 1 schreiben
    par_2 = max : rem Maximalwert in Parameter 2 schreiben
    max = 0
    min = 4095
    activate_pc : rem Damit wir ein Testpoint-Event ausgelöst
  endif
```

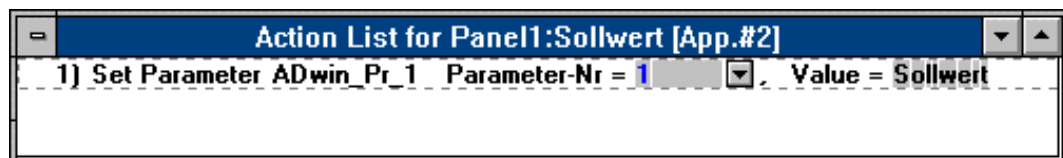
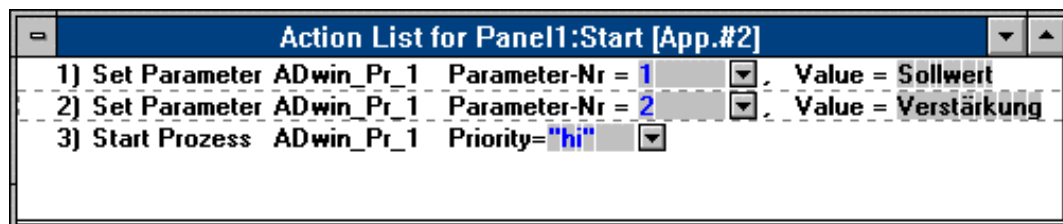
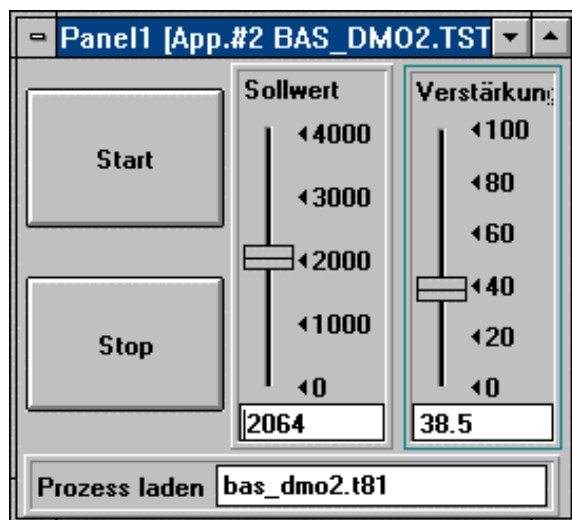


10.2 Digitaler P-Regler

```
rem Das Programm BAS_DM02 ist ein digitaler
rem P-Regler. Der Sollwert wird mit par_1
rem vorgegeben, die Verstärkung mit Par_2
rem Dieses Programm kann mit bis zu 35 kHz ablaufen
```

```
dim abweichung, stell as integer
```

```
abweichung = par_1 - adc(1)
stell = abweichung * par_2 + 2048
dac 1, stell
```



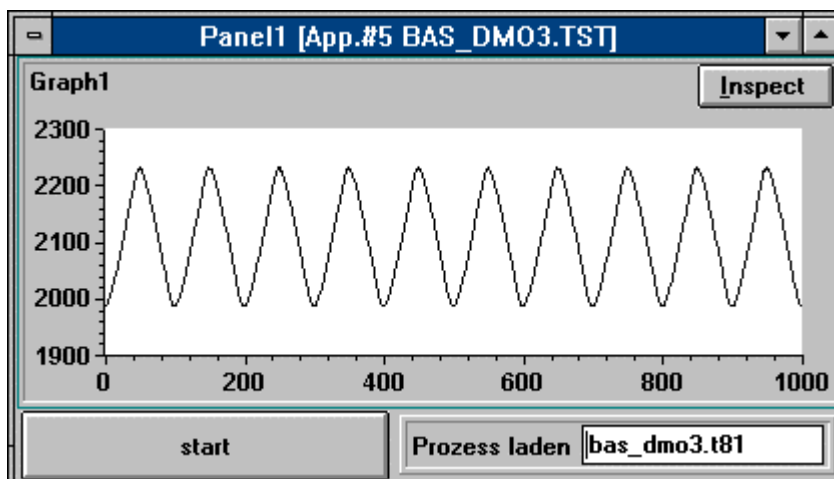
10.3 Beispiel zur Datenübertragung mit Data

```
rem  Das Programm bas_dmo3 mißt den analogen Eingang 1
rem  und löst nach 1000 Messungen ein Testpoint-Event aus
rem  um die Meßwerte abzuholen.
rem  Die Daten werden mit Hilfe eines DATA-Arrays übertragen.
```

```
dim data_1[1000] as integer
dim index as integer
```

```
init:
index = 0
```

```
event:
index = index+1
if (index>1000) then
    activate_pc
end
endif
data_1[index] = adc(1)
```



Action List for Panel1:ADwin_Pr_1 [App.#5]				
1) Read Data	ADwin_Pr_1	Dataset-Nr. = 1	length = 1000	start at 1
2) Draw graph	Graph1	with ADwin_Pr_1 ,		

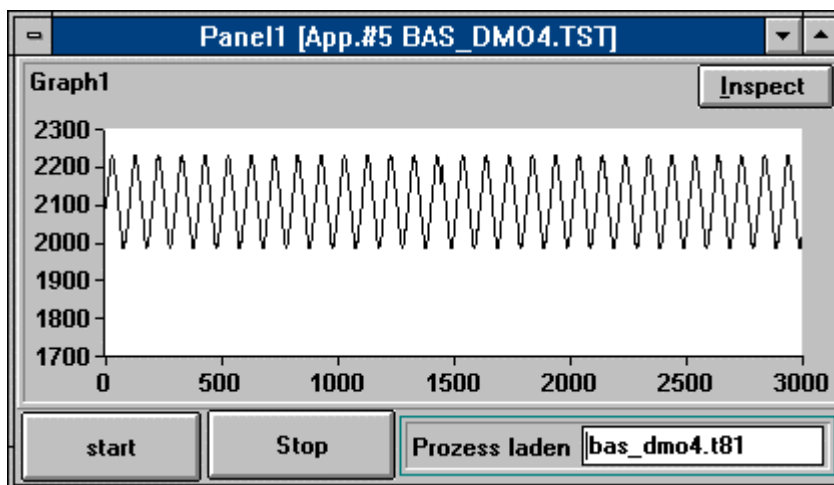
10.4 Beispiel zur kontinuierlichen Datenübertragung mit Fifo

```
rem Das Programm bas_dmo4 mißt den analogen Eingang 1
rem kontinuierlich und löst nach jeweils 1000 Messungen
rem ein Testpoint-Event aus, um die Meßwerte abzuholen.
rem Die Daten werden mit Hilfe eines Fifos übertragen.
```

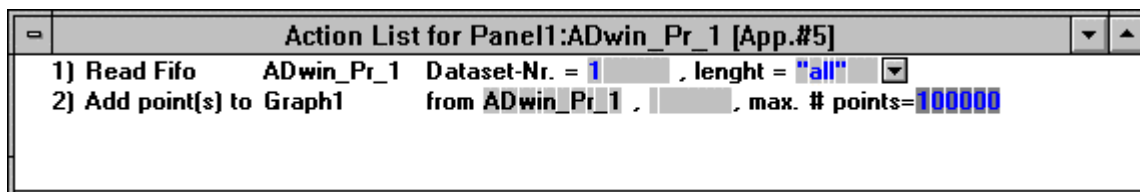
```
dim data_1[3000] as long as fifo
dim index as integer
```

```
init:
index = 0
```

```
event:
index = index+1
if (index>1000) then
    activate_pc
    index = 0
endif
data_1 = adc(1)
```



Nach jeweils 1000 Messungen wird die Action-List des Realtime-objects ausgelöst und der Fifo von Testpoint ausgelesen. Die Messung läuft dabei ungestört weiter. Das Graph-object ist im Strip-Chart Mode und zeigt immer die letzten Werte an



10.5 Beispiel PID-Regler

```
rem Das Programm BAS_DM06 ist ein digitaler PID-Regler.
rem Damit das Programm auch auf ADwin-2 und ADwin-4 Karten
rem benutzt werden kann arbeitet der Regler nur mit
rem Integer-Variablen. Die Reglerkoeffizienten werden auf
rem dem PC berechnet, mit 1000 multipliziert und als Integer-
rem Parameter an die ADwin-Karte uebergeben.
rem Par_1 = Sollwert in Digits
rem Par_2 =  $q_0 * 1000 = k * (1+t_0/(2*t_i)+t_d/t_0) * 1000$ 
rem Par_3 =  $q_1 * 1000 = -k * (1+2*t_d/t_0-t_0/(2*t_i)) * 1000$ 
rem Par_4 =  $q_2 * 1000 = k * t_d/t_0 * 1000$ 
rem Par_5 = Bufferindex fuer dei Regelabweichung

dim iw,abweichung as long
dim Sollwert, Verstärkung as long
dim ek, ek1, ek2, uk, uk1 as long
dim q0,q1,q2 as long
dim data_1[600] as short

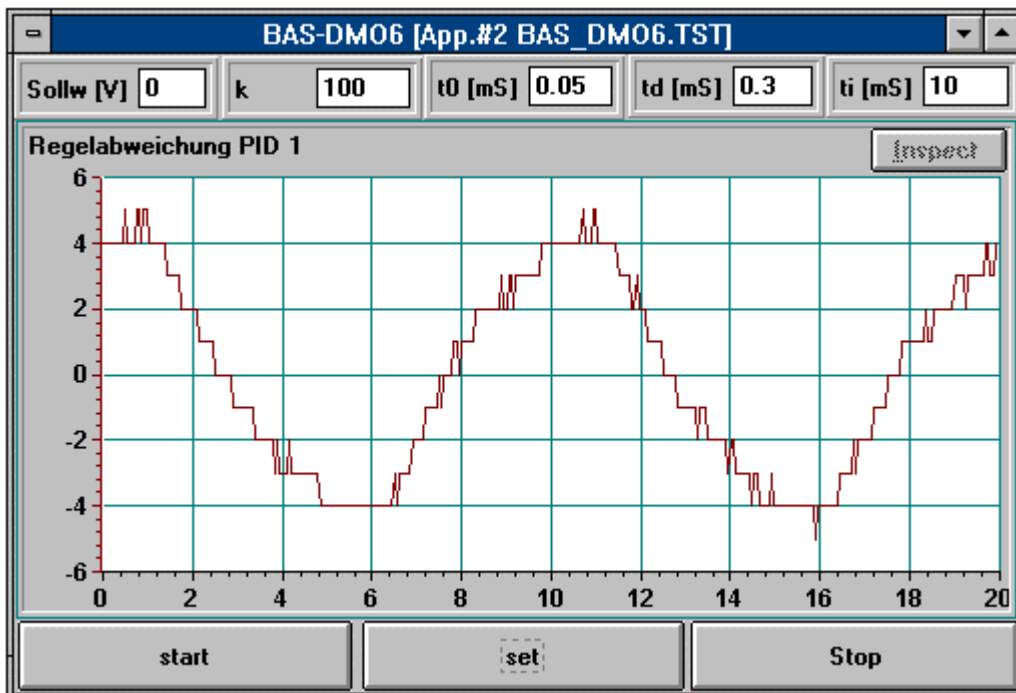
init:
sollwert = par_1
q0=par_2
q1=par_3
q2=par_4
par_5 = 0
set_mux(0)

event:
set_mux(0)           'Multiplexer auf den Eingang setzen
uk1 = uk + q0*ek
start_conv(1)        'Messung starten
dac 1, (uk1/1000)     'Stellwert ausgeben
                    'Da alle Parameter mit 1000 multipliziert wurden,
                    'muß er durch 1000 dividiert werden

ek2=ek1
ek1=ek
uk = uk1+q1*ek1+q2*ek2
wait_eoc(1)          'Warten bis der ADC ausgelesen werden kanm
ek = sollwert - readadc(1) 'Auslesen und Regelabweichung berechnen

    rem Die Regelabweichung in einen Buffer schreiben
if (par_5<500) then
    data_1[par_5] = ek
    par_5=par_5+1
    if (par_5=500) then activate_pc
endif

    rem Falls die Parameter geaendert wurden
    rem die muessen die neuen Parameter uebernommen werden.
if (par_9=1) then
    sollwert = par_1
    q0=par_2
    q1=par_3
    q2=par_4
    par_9=0
    par_5=0
endif
```



Das Testpointprogramm berechnet aus den Reglerparametern die Koeffizienten q_0 , q_1 und q_2 , multipliziert Sie mit 1000, damit sie als Integer übergeben werden können und schreibt sie in die Parameter 2, 3 und 4

Action List for BAS-DM06:start [App.#2]		
1) Calculate	to_digits	with u=Sollw [V]
2) Calculate	to_delay	with t0=t0 [mS]
3) Calculate	q0	with k=k t0=t0 [mS] ti=ti [mS] td=td [mS]
4) Calculate	q1	with k=k td=td [mS] t0=t0 [mS] ti=ti [mS]
5) Calculate	q2	with k=k td=td [mS] t0=t0 [mS]
6) Set Parameter ADwin_Pr_1	Parameter-Nr = 1	Value = to_digits, Type = "Integer"
7) Set Parameter ADwin_Pr_1	Parameter-Nr = Delay	Value = to_delay, Type = "Integer"
8) Calculate	mal1000	with x=q0
9) Set Parameter ADwin_Pr_1	Parameter-Nr = 2	Value = mal1000, Type = "Integer"
10) Calculate	mal1000	with x=q1
11) Set Parameter ADwin_Pr_1	Parameter-Nr = 3	Value = mal1000, Type = "Integer"
12) Calculate	mal1000	with x=q2
13) Set Parameter ADwin_Pr_1	Parameter-Nr = 4	Value = mal1000, Type = "Integer"
14) Set Parameter ADwin_Pr_1	Parameter-Nr = 9	Value = 1, Type = "Integer"
15) Start Prozess	ADwin_Pr_1	