

ADwin

Treiber für DIAdem 4+6

im Modus

- Einzelwertverarbeitung
- Paketverarbeitung
- Autosequenz



Dokument Version 1.001 /Januar 1999

© Jäger Computergesteuerte Meßtechnik GmbH / Lorsch

Inhaltsverzeichnis

1	DAS ADWIN-KONZEPT	4
1.1	Das ADwin System und DIAdem	5
1.2	Typische Anwendungen	5
1.2.1	Steuerung schneller Prüfstände	5
1.2.2	Signale generieren	5
1.2.3	Intelligent Messen, Erfassung von Daten mit komplexen Triggerbedingungen	5
1.2.4	Regeln und Steuern	6
1.2.5	Online-Verarbeitung, Datenreduzierung	6
1.2.6	Hardware-in-the-Loop, Simulation von Sensordaten	6
2	INSTALLATION	7
2.1	Installation des ADwin-Treibers	7
2.2	Installation des DIAdem-Treibers (Einzelwertverarbeitung und Autosequenzen)	7
2.3	Installation des DIAdem-Treibers (Paketverarbeitung) für ADwin-Systeme	8
3	BOOTEN DES ADWIN-SYSTEMS, LADEN VON ADBASIC-PROGRAMMEN	9
3.1.1	Unter DIAdem in der Paketverarbeitung	9
3.1.2	In den Autosequenzen	9
3.1.3	Im DIAdem Einzelwertmodus	9
4	ADWIN-OBJEKTE FÜR DIADEM-DAC (EINZELWERTMODUS)	10
4.1	DAC-Objekte zur Kommunikation mit ADbasic-Prozessen	11
4.1.1	ADwin System Settings / General	12
4.1.2	ADwin System Settings / Inputs bzw. ADwin System Settings / Outputs	13
4.1.3	PAR-Objekt	14
4.1.4	FPAR-Objekt	15
4.1.5	Data-Objekt	15
4.1.6	FData-Objekt	16
4.1.7	FIFO-Objekt	16
4.1.8	FFIFO-Objekt	17
4.2	Takt	18
4.2.1	Softwaretakt	19
4.2.2	Hardwaretakt	19
4.2.3	Unterschiedliche Abtastraten in einem ADwin-System	20
4.3	DAC-Objekte zur direkten Ein-/Ausgabe auf analogen und digitalen Kanälen	21
4.3.1	ADwin System Settings / General	21
4.3.2	AD-Objekt	22
4.3.3	DA-Objekt	22

4.3.4	DIG-Bit-Objekt	22
4.3.5	DIG-Byte-Objekt	23
5	ADWIN-DIADEM-AUTO - BEFEHLE FÜR AUTOSEQUENZEN	24
5.1	ADwin-DIAdem-Auto Variablen	24
5.2	ADwin-DIAdem-Auto Befehle	25
5.2.1	ADboot - Booten des <i>ADwin</i> -Systems	25
5.2.2	ADBlod - Laden eines <i>ADbasic</i> -Prozesses	25
5.2.3	Start_Process - Starten eines <i>ADbasic</i> -Prozesses	25
5.2.4	Stop_Process - Stoppen eines <i>ADbasic</i> -Prozesses	26
5.2.5	Get_Par - Lesen eines <i>ADbasic</i> -Integer-Parameters	26
5.2.6	Get_FPar - Lesen eines <i>ADbasic</i> -Floating-Point-Parameters	26
5.2.7	Set_Par - Schreiben eines <i>ADbasic</i> -Integer-Parameters	27
5.2.8	Set_FPar - Schreiben eines <i>ADbasic</i> -Floating-Point-Parameters	27
5.2.9	AD_Connect / AD_Disconnect - Netzwerkkommunikation mit <i>ADwin</i> -Systemen	28
6	ADWIN-DIADEM-PAKET-TREIBER	30
6.1	ADwin-DIAdem-DAC-Objekte (Paketverarbeitung) zur Kommunikation mit <i>ADbasic</i> -Prozessen	30
6.1.1	<i>ADwin</i> System Process	31
6.1.2	<i>ADwin</i> System Autolnit	32
6.1.3	<i>ADwin</i> System Parameter:	33
	<i>ADwin</i> System Data	34
6.1.5	<i>ADwin</i> System Network	35
7	DEMONSTRATIONSBEISPIELE IM DIADEM-DAC EINZELWERTMODUS	36
7.1	Globale <i>ADbasic</i> -Datensätze und Variablen von <i>ADbasic</i> nach DIAdem laden	36
7.1.1	Übertragen von Parametern	36
7.1.2	Übertragen von Datensätzen	37
7.1.3	Übertragen von FIFO-Datensätzen im Hardwaretakt	38
8	DEMONSTRATIONSBEISPIELE IM DIADEM-PAKETMODUS	39
8.1	Globale <i>ADbasic</i> -Datensätze und Variablen von <i>ADbasic</i> nach DIAdem laden	39
8.1.1	FIFO-Version 2, multiplexte Daten	39
8.1.2	FIFO-Version 1, nichtmultiplexte Daten	40
8.1.3	Version 3, mit Variablen zur Synchronisation	41
9	SUPPORT	42

1 Das ADwin-Konzept

Der Einsatz der intelligenten **ADwin**-Systeme eröffnet ein weites Spektrum an Anwendungen für den Bereich des schnellen und mikrosekundengenauen Regelns, Steuerns und Messens unter Windows und **DIAdem**.

Die **ADwin**-Systeme haben einen eigenen Prozessor und lokalen Speicher sowie analoge und digitale Ein-/Ausgänge. Der Prozessor führt die komplette Meßwerterfassung, Online-Verarbeitung und Reaktion durch. Jeder Meßwert wird durch den Prozessor des **ADwin**-Systems sofort nach der Erfassung weiterverarbeitet. Das Echtzeit-Entwicklungstool **ADbasic** erlaubt die Programmierung beliebiger mathematischer Operationen und Funktionen, die sofort nach jedem Abtastschritt ausgeführt werden. Als Reaktion könnten z.B. Ausgänge neu gesetzt oder die aufgenommenen Meßwerte plus die weiterverarbeiteten Meßwerte gespeichert werden.

Das ADwin-System und der Windows-PC. Obwohl das **ADwin**-System mit dem PC verbunden ist, laufen alle Anwendungen auf dem **ADwin**-System eigenständig ab. Der PC kann jedoch jederzeit auf das **ADwin**-System zugreifen, um Daten auszutauschen oder neue Programme auf das **ADwin**-System zu laden. Programme auf dem **ADwin**-System können unabhängig voneinander gestartet und gestoppt werden. Die Kommunikation zwischen dem PC und **ADwin-Gold** hat keinerlei Einfluß auf die schnelle und zeitlich exakte Abarbeitung der Programme auf dem System **ADwin-Gold**.

ADwin-Systeme machen DIAdem/Windows-PCs echtzeitfähig! Echtzeit: "Was ist das?"

Mit dem Begriff "Echtzeit" wird zunächst nur festgelegt, daß ein System innerhalb einer definierten Zeit auf eine Anforderung reagiert. Für eine vergleichbare Aussage muß also eine Zeitspanne angegeben werden. **ADwin**-Systeme reagieren garantiert in weniger als **300 Nanosekunden**, völlig unabhängig vom PC und dessen Auslastung.

ADwin-Systeme sind schnell! **ADwin**-Systeme verarbeiten jeden Meßwert sofort nach der Erfassung, und dies sogar mit Abtastraten von 1MHz. Die Kombination eines leistungsstarken SHARC-DSPs mit schnellen AD-/DA-Wandlern macht dies möglich.

Absolute Betriebssicherheit! Unabhängig von der Auslastung des PCs, unabhängig von der Stabilität eines PC-Programms: Das **ADwin**-System verrichtet alle Aufgaben kontinuierlich und zuverlässig. Stürzt der PC ab, das **ADwin**-System läuft weiter, alle Regelungen und Steuerungen laufen stabil, alle Meßwerte sind gerettet. Nach dem Neustart des PCs kann auf alle zwischenzeitlich aufgenommenen Daten zugegriffen werden.

Das ADwin-System ist frei programmierbar! Das Echtzeit-Entwicklungstool **ADbasic** ermöglicht die einfache und schnelle Erstellung von zeitkritischen Programmen für die **ADwin**-Systeme. **ADbasic** hat eine integrierte Entwicklungsumgebung unter Windows mit Möglichkeiten zum Online-Debugging. Es bietet dem Entwickler eine gewohnte BASIC-Befehlssyntax mit erweiterten Funktionen für den Zugriff auf analoge und digitale Ein- und Ausgänge sowie Funktionen zur Prozeßsteuerung und zum Datenaustausch mit dem PC. Bei der Compilierung der Prozesse durch **ADbasic** wird auf die minimale Laufzeit der Programme sehr großer Wert gelegt.

1.1 Das ADwin System und DIAdem

Die Kombination von **DIAdem** mit einem **ADwin**-System bietet dem Anwender völlig neue Möglichkeiten. Die Intelligenz und Rechenleistung des **ADwin**-Systems zum einen und die Funktionen zum Verwalten, Analysieren und Dokumentieren von Meßdaten zum anderen bieten ein leistungsstarkes Konzept.

1.2 Typische Anwendungen

1.2.1 Steuerung schneller Prüfstände

Prüfstandsanwendungen erfordern vielseitige Funktionen, wie z. B.:

- Meßdatenerfassung
- Signalerzeugung
- Steuerungs- und Regelfunktionen
- Digitale I/O-Steuerung

Alle diese Funktionen und weitere können auf einem einzigen **ADwin**-System parallel und mit hoher Geschwindigkeit zuverlässig ablaufen. Komplette Prüfstände werden durch ein **ADwin**-System gesteuert und geregelt, völlig unabhängig von dem PC-Betriebssystem.

Mit **ADwin**-Systemen werden Prüfstände realisiert für: Motoren, Getriebe, Kupplungen, Stoßdämpfer, Kugellager, Motorlager, Autoabgasanlagen, ABS, Lenkung, Schalter, Relais, etc.

1.2.2 Signale generieren

Viele Anwendungen benötigen Signalgeneratoren mit analogen oder digitalen Ausgängen. Mit einem **ADwin**-System können vielkanalige **periodische** oder **nicht-periodische** Signalformen erzeugt und bei Bedarf beliebige **Störsignale** aufgeschaltet werden. Frequenz, Phase, Amplitude und Offset können jederzeit korrigiert werden. Diese Korrektur kann gleitend oder als Sprung ausgeführt werden. Der SHARC-DSP des **ADwin**-Systems erlaubt online eine Korrektur des Ausgabewertes jeder einzelnen Stützstelle sowie des zeitlichen Abstands zwischen den Stützstellen. Die neuen Werte werden auf analogen und digitalen Kanälen ausgegeben und/oder stehen in Variablen für andere **ADbasic**-Programme (z.B. PID-Regler) zur Verfügung.

1.2.3 Intelligent Messen, Erfassung von Daten mit komplexen Triggerbedingungen

Jeder Meßwert wird durch das **ADwin**-System sofort nach der Erfassung verarbeitet, sogar mit Abtastraten von 1MHz. Signale an analogen und digitalen Eingängen werden gemessen, Zähler und interne Timer ausgelesen. **ADbasic** erlaubt die Programmierung beliebiger mathematischer Operationen und Funktionen, die sofort nach jedem Abtastschritt ausgeführt werden. Für jeden aufgenommenen Meßwert wird so überprüft, ob die geforderten Triggerbedingungen eingehalten sind und entschieden, ob der Meßwert gespeichert oder verworfen werden soll.

1.2.4 Regeln und Steuern

Digitale Regler sind mit **ADwin**-Systemen leicht realisierbar. Verschiedene Reglerstrukturen (PI, PID, Kaskadenregler, adaptive Regler, Zustands-Regler, ... etc.) arbeiten auf einem **ADwin**-System mit hohen Geschwindigkeiten von bis zu 500KHz zuverlässig, unabhängig von der PC-Auslastung. Auf einem **ADwin**-System können gleichzeitig verschiedene Regler mit unterschiedlichen Regelalgorithmen und Zykluszeiten parallel ablaufen.

Das Echtzeit-Entwicklungstool **ADbasic** bietet dem Anwender alle Möglichkeiten, um Regler auf **ADwin**-Systemen schnell und übersichtlich zu entwickeln. Standard-Regelalgorithmen können nach den Erfordernissen der Anlage erweitert, angepaßt oder durch andere Regler ersetzt werden.

1.2.5 Online-Verarbeitung, Datenreduzierung

Die Auswertung der Meßdaten durch den SHARC-DSP auf den **ADwin**-Systemen macht es erst möglich, Meßkanäle mit bis zu 1MHz über Stunden zu erfassen! Alle Auswertungen laufen online auf dem **ADwin**-System ab. Die in der Datenrate deutlich reduzierten Ergebnisse werden vom **DIAdem** protokolliert und dargestellt.

Typische Online-Berechnungen: Effektivwerte, Min/Max, Grenzwerte, Integrale, Ableitungen, statistische Auswertungen, Korrelationen, Signalanalysen, ...

1.2.6 Hardware-in-the-Loop, Simulation von Sensordaten

Hier gibt es zwei verschiedene Anwendungen:

- **ADwin**-Systeme sind in den Signalzweig eingekoppelt. Über analoge/digitale Eingänge werden Signale gelesen, über frei definierbare Funktionen abgebildet und über analoge/digitale Ausgänge wieder in den Signalzweig eingespeist.
- **ADwin**-Systeme werden am Anfang des Signalzweigs zur direkten Simulation von Sensordaten eingesetzt. Die Ausgabewerte werden entweder online berechnet oder stehen als Datensätze zur Verfügung.

Typische Simulationen von:

- frei definierbaren Störungen aufschalten
- Sensoren, Sensordaten
- Inkrementalgebern mit:
 - Gleichlaufschwankung
 - Störspitzen
 - Offsetdrift, etc.

2 Installation

2.1 Installation des ADwin-Treibers

Zur Installation des **ADwin**-Treibers legen Sie bitte die dem **ADwin**-System beigelegte **ADwin**-Treiber-CD bzw. **ADbasic**-CD in das CD-ROM Laufwerk und starten Sie die Installation.

2.2 Installation des DIAdem-Treibers (Einzelwertverarbeitung und Autosequenzen)

Alle Funktionen des **DIAdem** Einzelwerttreibers für **ADwin**-Systeme sind in der DLL **ADADwin.dll** verfügbar und können je nach Bedarf ausgewählt und geladen werden. Diese Treiberfunktionen können nur in der Verbindung mit der entsprechenden **ADwin**-Hardware genutzt werden. Die Treiber melden das Fehlen der Hardware erst beim Start der Messung.

Schritt	Was ist zu tun ?
1	Sind bereits ADwin -Treiber DIAdem -DAC Einzelwerttreiber installiert, dann stoppen Sie an dieser Stelle mit der Installation und deinstallieren Sie die alte Treiberversion. Gehen Sie hierzu in der umgekehrten Reihenfolge vor, wie im folgenden beschrieben und löschen Sie sämtliche Einträge und Dateien.
2	Sollte DIAdem bereits laufen, beenden Sie das Programm bitte.
3	Im Verzeichnis für zusätzliche Treiber auf der ADwin -CD bzw auf der DIAdem -CD (CD:\AddOn\ADwin\Driver\DIAdem6\ ; bzw. ...DIAdem4\) liegen die notwendigen Dateien, kopieren Sie diese in die DIAdem -Verzeichnisse: *.G5D → ...DIAdem\ , *.bmp → ..DIAdem\symbols, *.dll → ..DIAdem\addinfo, adwdia46.t40, adwdia46.t50, adwdia46.t80, adwdia46.t90 → c:\ADbasic3\
4	Starten Sie DIAdem
5	Schließen Sie in DIAdem alle offenen Fenster und gehen Sie in das Menü: Einstellungen → GPI-DLL-Registrierung..., es erscheint eine Liste mit registrierten GPI Erweiterungen. Durch Klicken auf Hinzufügen können Sie die Dateien ADADwin.DLL und ADinit.DLL auswählen. Schließen Sie anschließend alle Fenster und bestätigen Sie die Registrierung in der Datei Desktop.DDD. Anschließend startet DIAdem neu.
7	Gehen Sie bitte im Fenster DIAdem -DAC: MSR-Beschreibung auf Einstellungen → Einzelwert-Verarbeitung → Treiber konfigurieren
8	Klicken Sie bitte auf neuer Eintrag und wählen Sie den Treiber ADwin -Treiber bzw. ADwinInit-Treiber aus. Die von dem Treiber zur Verfügung gestellten Funktionen erscheinen in der rechten Auswahlliste. Laden Sie bitte nacheinander alle Funktionen durch Anklicken. Achten Sie hierbei bitte darauf, diese in der Reihenfolge der angebotenen Funktionen zu laden. Wird die Reihenfolge nicht beachtet, stimmen die Icons in der Aktionsliste nicht mit den Funktionen überein.
9	Hiermit ist die Installation beendet

In der Aktionsliste finden Sie hinter den Knöpfen
die ausgewählten Funktionen zum Gebrauch wieder.



2.3 Installation des DIAdem-Treibers (Paketverarbeitung) für ADwin-Systeme

Alle Funktionen der Paketverarbeitung für **ADwin**-Systeme sind in der DLL **ADwinpak.dll** verfügbar und können je nach Bedarf ausgewählt und geladen werden. Diese Treiberfunktionen funktionieren nur in der Verbindung mit der entsprechenden **ADwin**-Hardware.

Die Treiber melden das Fehlen der Hardware erst beim Start der Messung.

Schritt	Die Installation - "Was ist zu tun ?"
1	Sind bereits ADwin -Treiber DIAdem -DAC Pakettreiber installiert, dann stoppen Sie an dieser Stelle mit der Installation und deinstallieren Sie die alte Treiberversion. Gehen Sie hierzu in der umgekehrten Reihenfolge vor, wie im folgenden beschrieben und löschen Sie sämtliche Einträge und Dateien.
2	Sollte DIAdem bereits laufen, beenden Sie das Programm bitte.
3	Im Verzeichnis für zusätzliche Treiber auf der ADwin -CD bzw auf der DIAdem -CD (CD:\AddOn\ADwin\Driver\ DIAdem6 \ ; bzw. ...\ DIAdem4 \) liegen die notwendigen Dateien, kopieren Sie diese in die DIAdem -Verzeichnisse: Kopieren Sie bitte die ADwinpak.DLL in den DIAdem -Pfad ..\ DIAdem \addinfo ; ADwinpak.BMP in den DIAdem -Pfad ..\ DIAdem \symbols
4	Starten Sie DIAdem
5	Schließen Sie in DIAdem alle offenen Fenster und gehen Sie auf Einstellungen → GPI-DLL-Registrierung..., es erscheint eine Liste mit registrierten GPI Erweiterungen. Durch Klicken auf Hinzufügen können Sie die Datei ADwinpak.DLL auswählen. Schließen Sie anschließend alle Fenster und bestätigen Sie die Registrierung in der Datei Desktop.DDD. Anschließend startet DIAdem neu.
7	Gehen Sie bitte im Fenster DIAdem -DAC: MSR-Beschreibung auf Einstellungen → Paket-Verarbeitung → Treiber konfigurieren
8	Klicken Sie bitte auf neuer Eintrag und wählen Sie den Pakettreiber ADPAKET aus. Die von dem Treiber zur Verfügung gestellten Funktionen erscheinen in der rechten Auswahlliste. Laden Sie die gewünschten Funktionen bitte einzeln durch Anklicken. Achten Sie hierbei bitte darauf, diese in der Reihenfolge der angebotenen Funktionen zu laden. Wird die Reihenfolge nicht beachtet, stimmen die Icons in der Aktionsliste nicht mit den Funktionen überein.
9	Hiermit ist die Installation beendet.

In der Aktionsliste finden Sie hinter dem Knopf
die ausgewählten Funktionen zum Gebrauch wieder.



3 Booten des **ADwin**-Systems, Laden von **ADbasic**-Programmen

Der PC und **DIAdem** kann erst mit einem **ADwin**-System kommunizieren, nachdem der **ADwin**-Treiber auf das **ADwin**-System geladen wurde. Der **ADwin**-Treiber muß in jedem Fall nach jeder Unterbrechung der Spannungsversorgung des **ADwin**-Systems geladen werden.

Das Booten löscht den gesamten Speicher des **ADwin**-Systems, so daß alle bereits auf dem System befindlichen Programme und Daten dabei verloren gehen.

Der Treiber wird wahlweise durch eine der drei nachfolgenden Möglichkeiten geladen.

3.1.1 Unter **DIAdem** in der Paketverarbeitung

In der Paketverarbeitung stehen zwei Objekte für die Prozeßverwaltung zur Verfügung.

Das Objekt **ADwin System Autoinit** dient zum Booten, dem Programmdownload, sowie dem Starten von **ADbasic**-Programmen.

Das Objekt **ADwin System Process** dient dem Programmdownload, dem Starten/Stoppen von **ADbasic**-Programmen, der Überwachung des Software Triggers **Activate_PC** und dem Einstellen des Prozeß-Delays (Zykluszeit eines **ADbasic**-Prozesses).

3.1.2 In den Autosequenzen

Über die Autosequenzfunktion **ADboot** kann ein **ADwin**-System gebootet, über die Funktion **ADBload** kann ein **ADbasic** Prozeß auf das System geladen werden.

3.1.3 Im **DIAdem** Einzelwertmodus

Die DAC-Objekte testen beim Start des **DIAdem**-Programms, ob das **ADwin**-System gebootet wurde. Ist dies nicht der Fall, wird die Karte automatisch gebootet. Die Kartenadresse, der Prozessortyp und der Treiberpfad müssen für die **DIAdem**-Objekte unter **ADwin System Settings / General** angegeben werden. Weiterhin gibt das Objekt **ADwin System Process** eine übersichtliche Möglichkeit, das System zu booten und einzelne Dateien zu starten.

4 ADwin-Objekte für DIAdem-DAC (Einzelwertmodus)

Zur Kommunikation mit **ADwin**-Systemen über den **ADwin-DIAdem**-DAC Treiber stehen zwei Möglichkeiten zur Verfügung:

- **Kommunikation mit ADbasic-Prozessen**

Diese Betriebsart ist im Einzelwertmodus die Standardvariante und nutzt die Fähigkeiten des **ADwin**-Systems voll aus. Auf dem **ADwin**-System laufen verschiedene Regelungen, Datenerfassungen, online-Verarbeitungen, Signalerzeugungen, etc. ab. Diese Funktionen nutzen globale Variablen und Datensätze, auf welche **DIAdem** über verschiedene Objekte zugreifen kann. Zusätzlich gibt es Funktionen zum Starten, Stoppen, Booten, Laden von Programmen.

- **Direktes Schreiben und Lesen von digitalen & analogen Werten**

Diese Funktionen dienen vorwiegend dazu, während der Inbetriebnahme des Systems schnell einige Eingänge zu lesen und darzustellen. Diese Funktionen nutzen die Fähigkeiten des **ADwin**-Systems nur zu einem geringen Bruchteil aus.

Durch einen Doppelklick auf das jeweilige Objekt des **ADwin-DIAdem**-DAC Treibers öffnet sich ein Fenster zum Einstellen von verschiedenen Grundfunktionen:

Erfassung-Signalliste

Name:

Signaltyp:

Signalleitungen:

Nr	Signalname	Anschl.-Nr	Aktiv
1	DLL_In1_1	0	Ja

Buttons: OK, Abbrechen, Signaltyp..., Gerät..., Listenlänge..., Hilfe

Objekteigenschaften wie Signalname, Anschlußnummer, Name, Listenlänge, etc. können eingestellt werden. Spezifische Einstellungen für **ADwin-Objekte** müssen unter **Anschluß-Nr.**, **Listenlänge** und **Geräte** durchgeführt werden.

Die Einstellungen bei **Anschluß-Nr.** und **Listenlänge** sind für verschiedene **ADwin-Objekte** unterschiedlich, diese Eigenschaften werden in jedem Objekt einzeln erläutert.

Bei Betätigung der Schaltfläche **Gerät** erscheint ein Dialog des **DIAdem-ADwin**-Treibers zur Einstellung von Treiberparameter, die zur korrekten Kommunikation des Treibers mit dem **ADwin**-System benötigt werden.

4.1 DAC-Objekte zur Kommunikation mit ADbasic-Prozessen

Folgende Objekte stehen zum Datenaustausch zwischen dem in **ADbasic** programmierten **ADwin**-System und **DIAdem** zur Verfügung, pro Objekt je zwei Varianten zum Einlesen und zum Ausgeben:

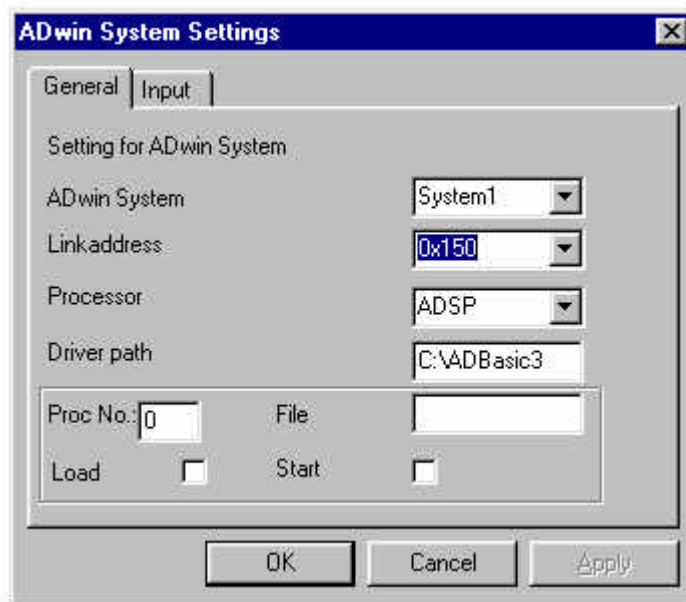
Objekt	Funktion
PAR	Austausch von Parametern (globale Integer-Variable in ADbasic)
FPAR	Austausch von Floating-Point-Parametern (globale Floating-Point-Variable in ADbasic)
DATA	Austausch von Datensätzen (DATAs, globale Integer-Arrays in ADbasic)
FDATA	Austausch von Floating-Point-Datensätzen (DATAs, globale Floating-Point-Arrays in ADbasic)
FIFO	Austausch von FIFOs (globale Integer-FIFOs in ADbasic)
FFIFO	Austausch von Floating-Point-FIFOs (globale Floating-Point-FIFOs in ADbasic)

Folgendes Objekt steht zum Booten und zum Laden von **ADbasic**-Prozessen zur Verfügung.

Objekt	Funktion
Init	Booten des ADwin -Systems, Laden und Starten von Programmen

4.1.1 ADwin System Settings / General

Hier können für das spezifizierte System **Linkaddress**, **Processor** und **Driver Path** angegeben werden. Alle Einstellungen, die auf der Karte **General** getroffen werden, gelten für das unter **ADwin-System** ausgewählte System. Sinnvoll ist es mit mehreren Systemen zu arbeiten, wenn verschiedene **ADwin**-Hardware genutzt wird, oder wenn auf einem **ADwin**-System mehrere Prozesse gleichzeitig ablaufen sollen und diese über die **Load** Funktion des **ADwin-DIAdem**-Blocks geladen und gestartet werden.



Sind mehrere **ADwin**-DAC-Objekte mit dieser System-Nr. bereits geladen, so werden Änderungen in einem Objekt automatisch in alle anderen Objekte mit der gleichen System-Nr. übernommen. Wird ein weiteres Objekt mit der gleichen System-Nr. hinzugeladen, so erhalten alle bereits geladenen Objekte mit dieser System-Nr. die gleichen Werte, wie das neu geladene Objekt (also die Standardwerte).

Es existieren zehn verschiedene Standardkonfigurationen. Diese Standardkonfigurationen können bei Bedarf angepaßt werden. Werden z.B. an einem PC verschiedene **ADwin**-Systeme gleichzeitig oder abwechselnd betrieben, müssen für die verschiedenen Systeme die Standardeinstellungen im Menü **Einstellungen** → **Einzelwert-Verarbeitung** → **Treiber konfigurieren** angepaßt werden. Hierzu öffnen Sie bitte mit einem Doppelklick auf **Process-ADwin-Treiber** eine Liste der zur Verfügung stehenden Funktionen. Nun können Sie für jedes Objekt die Standardwerte neu zuordnen.

Die Standardwerte des Treibers (Lieferzustand) für Process1 bis 10 sind:

Linkaddress = 0x150h

Processor = ADSP

Driver Path = c:\adbadic3\

Die Objekte erlauben das Laden einer **ADbasic**-Programmdatei-Datei (Binärdatei).

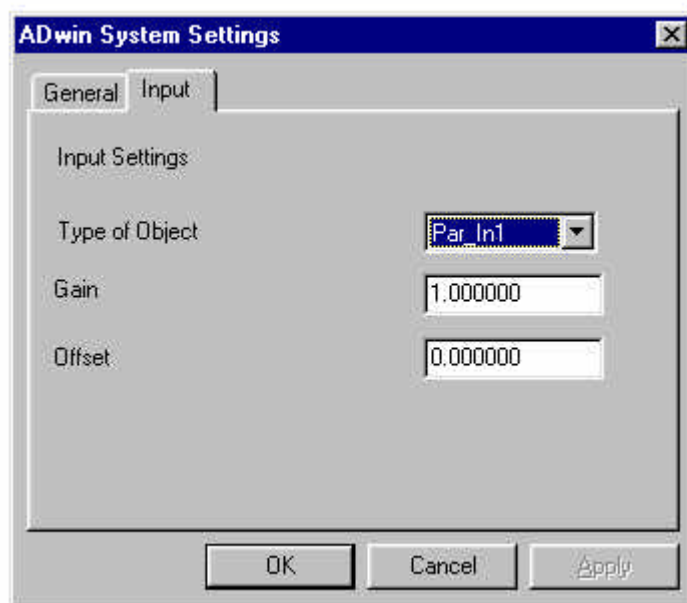
Hierzu sind die folgenden Angaben erforderlich:

Process No.:	Nummer des zu ladenden Prozesses (der Process muß in ADbasic ebenfalls unter dieser Prozeßnummer erzeugt werden)
File	Die zu ladende Binärdatei
Load	Auswählen, wenn die Datei geladen werden soll.
Start	Auswählen, wenn die Datei gestartet werden soll. Hierzu muß diese Datei auch über diesen Block geladen worden sein.

4.1.2 ADwin System Settings / Inputs bzw. ADwin System Settings / Outputs

Hier erfolgt eine Skalierung der eingelesenen Werte.

Zunächst wird unter **Type of Object** eine Eingangskonfiguration festgelegt. Pro Objekttyp stehen vier verschiedene Eingangskonfigurationen zur Verfügung. In den Eingangskonfigurationen können für den ausgewählten Block **Gain** (Verstärkungsfaktor) und **Offset** angegeben werden. Die Einstellungen für **Gain** und **Offset** gelten für alle **ADwin-DIADEM**-Objekte, auf welchen dieselbe Objektkonfiguration **Type of Object** gewählt ist.



Über **Gain** und **Offset** werden die Werte des **ADwin**-Systems wie folgt umgerechnet:

$$X = \text{Wert} \times \text{Gain} + \text{Offset}$$

Mögliche Eingangskonfigurationen für **Gain** und **Offset**:

Eingangsspg. Bereich	ADwin-Wertebereich	Gain	Offset	DIAdem-Wertebereich
+10V	0 .. 65535	1	0	0 .. 65535
+10V	0 .. 65535	0.000305180	-10	-10 ... 10
+10V	0 ... 4095	0.004882812	-10	-10 ... 10
+5V	0 .. 65535	0.000305180	-5	-5 ... 5
+5V	0 ... 4095	0.004882812	-5	-5 ... 5
0 .. 10V	0 .. 65535	0.000305180 / 2	0	0 ... 10
0 .. 10V	0 ... 4095	0.004882812 / 2	0	0 ... 10

Einige Objekte verfügen über zusätzliche Möglichkeiten der Eingangskonfigurationen, diese werden in den jeweiligen Abschnitten beschrieben.

4.1.3 PAR-Objekt

Die PAR-Objekte dienen zum Datenaustausch mit **ADbasic**-Parametern (globale Integer-Variablen). Es gibt zwei verschiedene Objekte, zur Erfassung und zur Ausgabe. Das Erfassungs-Objekt liefert als Ergebnis den aktuellen Inhalt eines **ADbasic**-Parameters. Das Ausgabe-Objekt schreibt in die spezifizierten Parameter die übergebenen Werte.

Spezifische Einstellungen in der Erfassungs-Signalliste bzw. Ausgabe-Signalliste

Anschluß-Nr.	Gibt die Nummern der zu übertragenden Parameter an, gültige Werte sind 1..80. Der Standardwert ist 0, dieser muß vom Bediener auf einen gültigen Wert gesetzt werden. Wird die Listenlänge > 1 angegeben, können mehrere Parameter gleichzeitig übertragen werden. Bei einer Listenlänge >1 wird von DIAdem eine aufsteigende Reihenfolge der Parameternummern angeboten, diese kann angepaßt werden.
Listenlänge	Gibt die Anzahl der zu übertragenden Parameter an, der Standardwert ist 1

4.1.4 FPAR-Objekt

Die FPAR-Objekte dienen zum Datenaustausch mit **ADbasic**-Floating-Point-Parametern (globale Floating-Point-Variablen). Es gibt zwei verschiedene Objekte, zur Erfassung und zur Ausgabe. Das Erfassungs-Objekt liefert als Ergebnis den aktuellen Inhalt eines **ADbasic**- Floating-Point-Parameters. Das Ausgabe-Objekt schreibt in die spezifizierten Floating-Point-Parameter die übergebenen Werte.

Spezifische Einstellungen in der Erfassungs-Signalliste bzw. Ausgabe-Signalliste

Anschluß-Nr.	Gibt die Nummern der zu übertragenden Floating-Point-Parameter an, gültige Werte sind 1..80. Der Standardwert ist 0, dieser muß vom Bediener auf einen gültigen Wert gesetzt werden. Wird die Listenlänge > 1 angegeben, können mehrere Floating-Point-Parameter gleichzeitig übertragen werden. Bei einer Listenlänge >1 wird von DIADEM eine aufsteigende Reihenfolge der Parameternummern angeboten, diese kann angepaßt werden.
Listenlänge	Gibt die Anzahl der zu übertragenden Floating-Point-Parameter an, der Standardwert ist 1

4.1.5 Data-Objekt

Die Data-Objekte dienen zum Datenaustausch mit **ADbasic**-Datensätzen (globale Integer-Arrays). Es gibt zwei verschiedene Objekte, zur Erfassung und zur Ausgabe. Das Erfassungs-Objekt liefert als Ergebnis den aktuellen Inhalt eines **ADbasic**-Datensatzes. Das Ausgabe-Objekt schreibt in den spezifizierten Datensätzen die übergebenen Werte. Für Lesen oder Schreiben gilt: Es werden die Datensatzelemente übertragen, die unter **Anschluß-Nr.** spezifiziert sind.

Wichtig! Die korrekte Nummer des zu übertragenden Datensatzes muß in den **ADwin System Settings** unter **Data No.** eingetragen werden.

Spezifische Einstellungen in der Erfassungs-Signalliste bzw. Ausgabe-Signalliste

Anschluß-Nr.	Es werden die Datensatzelemente übertragen, die unter Anschluß-Nr. spezifiziert sind. Achtung! Gültige Werte sind 1..Datensatzlänge. Der Standardwert ist 0, dieser muß vom Bediener auf einen gültigen Wert gesetzt werden. Wird die Listenlänge > 1 angegeben, können mehrere Datensatzeinträge gleichzeitig übertragen werden. Bei einer Listenlänge >1 wird von DIADEM eine aufsteigende Reihenfolge der Parameternummern angeboten, diese kann angepaßt werden.
Listenlänge	Gibt die Anzahl der zu übertragenden Datensatzeinträge an, der Standardwert ist 1.

4.1.6 FData-Objekt

Die FData-Objekte dienen zum Datenaustausch mit **ADbasic**-Floating-Point-Datensätzen (globale Floating-Point-Arrays). Die weiteren Einstellungen sind mit denen des Data-Objektes identisch, *siehe Data-Objekt*.

4.1.7 FIFO-Objekt

Die FIFO-Objekte dienen zum Datenaustausch mit **ADbasic**-FIFOs (globale Integer-Arrays als FIFO deklariert). Es gibt zwei verschiedene Objekte, zur Erfassung und zur Ausgabe. Das Erfassungs-Objekt liefert als Ergebnis den aktuellen Inhalt eines **ADbasic**-FIFO-Datensatzes. Das Ausgabe-Objekt schreibt in den spezifizierten FIFO-Datensätzen die übergebenen Werte. Die Übertragung der Werte zwischen dem FIFO und **DIAdem** geschieht synchron mit den **DIAdem**-Einstellungen für den Software-Takt oder Hardware-Takt. Für das Lesen gilt: Die FIFO Werte werden von **DIAdem** ausgelesen. Die Listenlänge gibt die Anzahl der Kanäle an, die in einem FIFO nacheinander abgelegt sind. Für das Schreiben: **DIAdem** schreibt die Werte so schnell wie möglich in den FIFO-Datensatz. Die Listenlänge gibt die Anzahl der Kanäle an, die in einem FIFO nacheinander abgelegt werden.

Wichtig! Die korrekte Nummer des zu übertragenden FIFO-Datensatzes muß in den **ADwin System Settings** unter **Data No.** eingetragen werden.

Spezifische Einstellungen in der Erfassungs-Signalliste bzw. Ausgabe-Signalliste, sowie für die ProzeßNr (ProcNo.)

Anschluß-Nr.	Achtung! Gültige Werte sind 1..Datensatzlänge. Der Standardwert ist 0, dieser muß vom Bediener auf einen gültigen Wert gesetzt werden. Wird die Listenlänge > 1 angegeben, können mehrere Datensatzeinträge gleichzeitig übertragen werden. Bei einer Listenlänge >1 wird von DIAdem eine aufsteigende Reihenfolge der Parameternummern angeboten, diese kann angepaßt werden.
Listenlänge	Gibt die Anzahl der zu übertragenden multiplexenden oder demultiplexenden Daten an, der Standardwert ist 1. Wird die Listenlänge > 1 angegeben, können aufeinanderfolgende FIFO-Einträge beim Lesen nach DIAdem demultiplext werden.
ProcNo.	Wird das Objekt im Modus Hardwaretakt betrieben, muß hier die Prozeßnummer des ADbasic -Prozesses angegeben werden. Beim Start wird automatisch die für den Hardwaretakt eingestellte Abtastrate an den ADbasic -Prozeß auf dem ADwin -System übergeben. Hierbei ist darauf zu achten, daß die gewählte Abtastrate durch den ADbasic -Prozeß ausgeführt werden kann.

Hinweis: Dieses Objekt kann sowohl im Softwaretakt als auch im Hardwaretakt betrieben werden. Für langsame Messungen (<100Hz Summenabtastrate) ist die Übertragung im Softwaretakt ausreichend. Für schnellere Messungen wird der Hardwaretakt empfohlen.

ADbasic-Prozesse deren Daten im **DIAdem**-Hardwaretakt gelesen werden, müssen die Werte in ein FIFO schreiben, auch dann, wenn mehrere Kanäle gemessen werden.

Weitere Einzelheiten finden Sie im Kapitel „Takt“.

4.1.8 FFIFO-Objekt

Die FFIFO-Objekte dienen zum Datenaustausch mit **ADbasic** Floating-Point-FIFO-Datensätzen (globale Floating-Point-Arrays als FIFO deklariert).

Die weiteren Einstellungen sind mit denen des FIFO-Objektes identisch, *siehe FIFO-Objekt*.

Hinweis: Dieses Objekt kann sowohl im Softwaretakt als auch im Hardwaretakt betrieben werden. Einzelheiten finden Sie im Kapitel „Takt“.

4.2 Takt

Das Takt-Objekt kann zusammen mit den **ADwin**-Objekten für den **DIAdem-DAC**-Einzelwertmodus in zwei möglichen Betriebsarten genutzt werden:

- Softwaretakt

In diesem Modus werden die Daten mit dem in **DIAdem** eingestellten Default-Takt übertragen.

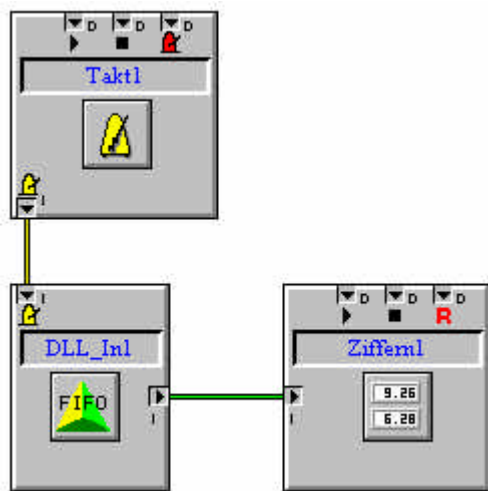
- Hardwaretakt

In diesem Modus werden die Daten des **ADwin**-Systems mit der maximalen Geschwindigkeit von **DIAdem** abgeholt. Das Timing wird durch das **ADwin**-System vorgegeben. Bei jedem Zugriff von **DIAdem** werden alle aufgenommenen Daten übertragen. **ADbasic**-Prozesse, deren Daten im **DIAdem**-Hardwaretakt gelesen werden, müssen die Werte in ein FIFO schreiben. Werden mehrere Kanäle gemessen, so werden die Daten multiplext in einem FIFO abgelegt.

Für langsame Messungen (<100Hz Summenabtastrate) ist die Übertragung im Softwaretakt ausreichend. Für schnellere Messungen wird der Hardwaretakt empfohlen.

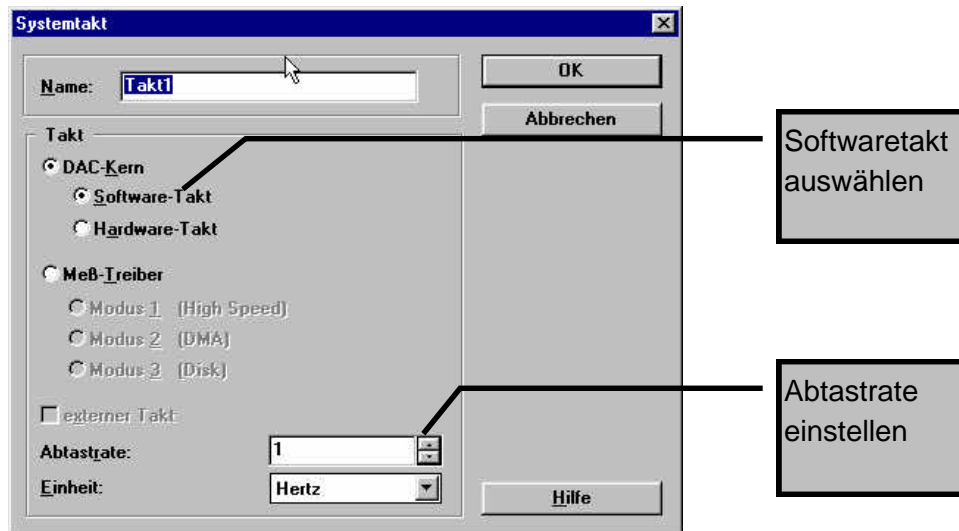
Das Takt-Objekt muß mit einer Systemleitung an das **ADwin**-Objekt angeschlossen werden.

Folgende Abbildung zeigt als Beispiel die Verbindung zu einem **ADwin-FIFO**-Objekt:



4.2.1 Softwaretakt

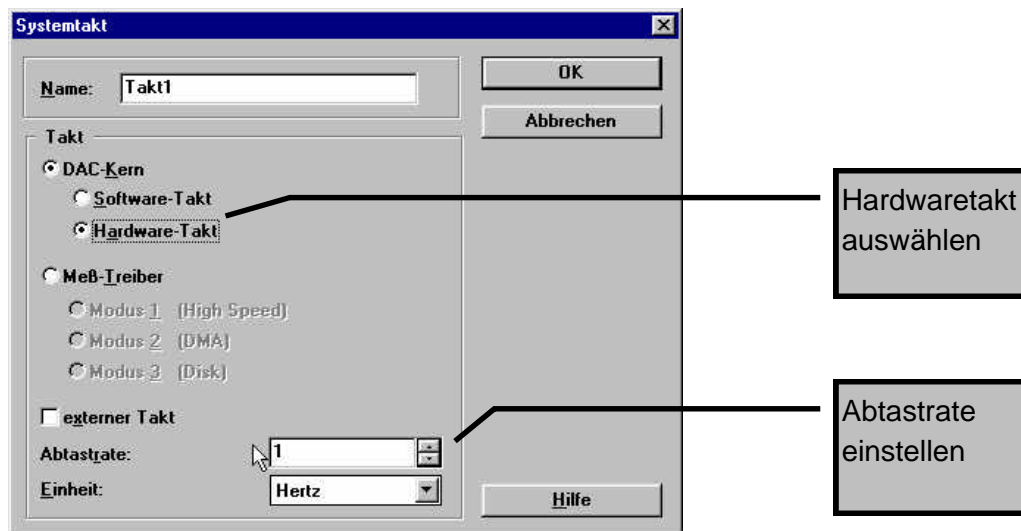
Der Softwaretakt kann mit allen **ADwin-DIAdem**-DAC-Objekten genutzt werden.



4.2.2 Hardwaretakt

Der Hardwaretakt kann mit den **ADwin-DIAdem**-DAC-Objekten:

FIFO und **FFIFO** genutzt werden.



Die ausgewählte Abtastrate wird als Abtastrate (Delay) auf dem **ADwin**-System eingestellt. Hierzu ist die Angabe der **ProcessNo** in dem **DIAdem** FIFO-Objekt notwendig.

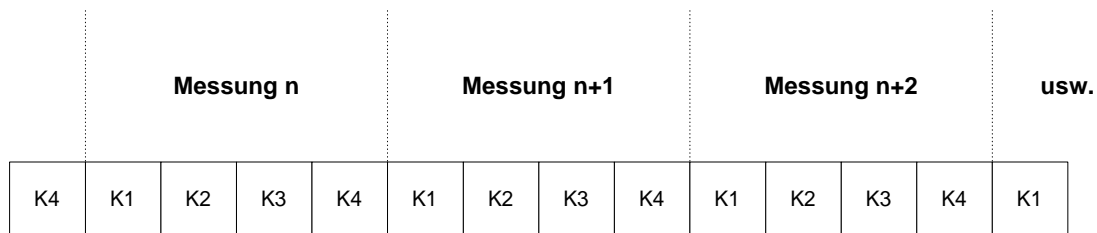
Bei jedem Zugriff von **DIAdem** auf das **ADwin**-System werden alle zur Verfügung stehenden Werte abgeholt und entsprechend der Abtastrate über die Zeit skaliert. **DIAdem** kann also asynchron und wesentlich langsamer auf ein **ADbasic**-Programm des **ADwin**-Systems zugreifen, bei jedem Zugriff werden alle zwischenzeitlich erfaßten Werte abgeholt. Diese Werte werden mit der eingestellten Skalierung (Abtastrate) weiterverarbeitet.

4.2.3 Unterschiedliche Abtastraten in einem ADwin-System

In einem **ADwin**-System können Eingänge mit unterschiedlichen Abtastraten erfaßt werden, also schnelle Signale (Drücke, Drehzahlen, Momente, ...) und langsame Signale (Temperaturen, Füllstände, ...) gleichzeitig.

Realisiert wird dies durch verschiedene **ADbasic**-Prozesse, die gleichzeitig auf dem **ADwin**-System ablaufen, mit unterschiedlichen Abtastraten (Delays/Zykluszeiten). Die Realisierung dieser Meßprozesse sollte günstigerweise mit FIFOs oder FFIFOs aufgebaut werden, die **DIADEM** im Hardwaretakt ausliest.

Pro Abtastrate nimmt ein FIFO die Meßergebnisse auf. Die Erfassung verschiedener Kanäle mit der gleichen Abtastrate erfolgt durch multiplextes Ablegen der Meßergebnisse im FIFO (alle Kanäle werden reihum im FIFO abgelegt). Beim Auslesen in **DIADEM** werden die einzelnen Kanäle durch die Angabe der Listenlänge wieder getrennt (demultiplext).



K1 - ADC1
K2 - ADC2
K3 - Counter1
K4 - Digital Inputs (Word)

**Organisation der Werte im FIFO,
bei Aufnahme mehrerer Kanäle**

Insgesamt stehen gleichzeitig vier FIFOs und vier FFIFOs zur Verfügung.

4.3 DAC-Objekte zur direkten Ein-/Ausgabe auf analogen und digitalen Kanälen

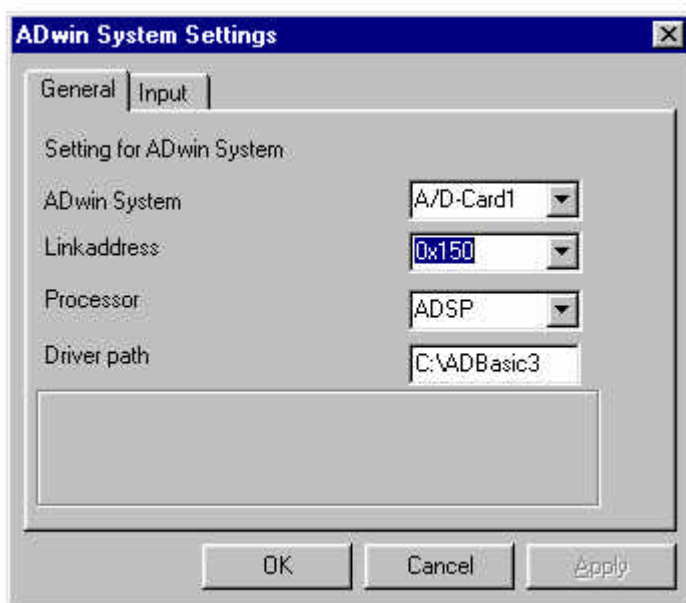
Folgende Objekte gestatten den direkten Zugriff auf analoge und digitale Ein-/Ausgänge der Systeme **ADwin**, **ADwin-light** und **ADwin-Gold**. Von den folgenden Objekten können in einem Schaltplan jeweils nur ein Objekt des gleichen Typs betrieben werden.

Zum Betrieb dieser Objekte muß sichergestellt sein, daß in den Pfad c:\ADbasic3\ die Dateien *adwdia46.t90*, *adwdia46.t80*, *adwdia46.t50*, *adwdia46.t40* kopiert wurden. Alternativ zu diesem Pfad können die Dateien auch in ein anderes Verzeichnis kopiert werden, jedoch müssen dann die Standardeinstellungen für diese Objekte korrigiert werden.

Objekt	Funktion
AD	Einlesen von bis zu 16 analogen Eingängen
DA	Ausgabe von bis zu 6 analogen Ausgängen
DIG Bit	Bitweises Einlesen/Ausgeben von bis zu 16 digitalen Kanälen
DIG Byte	Wortweises Einlesen/Ausgeben von allen 16 digitalen Kanälen

Diese Funktionen dienen vorwiegend dazu, während der Inbetriebnahme des Systems schnell einige Eingänge zu lesen und darzustellen. Diese Funktionen nutzen die Fähigkeiten des **ADwin**-Systems nur zu einem geringen Bruchteil aus.

4.3.1 ADwin System Settings / General



Hier können für das spezifizierte System **Linkaddress**, **Prozessortyp** und **Driver Path** angegeben werden. Die Dateien *adwdia46.t90*, *adwdia46.t80*, *adwdia46.t50*, *adwdia46.t40* müssen in das Verzeichnis **Driver Path** kopiert worden sein.

4.3.2 AD-Objekt

Das AD-Objekt mißt einen oder mehrere analoge Eingänge.

Spezifische Einstellungen in der Erfassungs-Signalliste

Anschluß-Nr.	Anschlußnummer entspricht dem analogen Eingangskanal, gültige Werte 1..16. Soll mehr als ein Kanal erfaßt werden, muß die Anzahl der Kanäle bei Listenlänge eingetragen werden (zuvor sollte aber die Anschlußnummer für den ersten Kanal gesetzt werden, da die weiteren Kanäle automatisch um eins erhöht in die Liste eingetragen werden).
Listenlänge	Gibt die Anzahl der zu erfassenden analogen Eingänge an, der Standardwert ist 1.

4.3.3 DA-Objekt

Das DA-Objekt gibt auf einen oder mehreren analogen Ausgängen Werte aus.

Spezifische Einstellungen in der Ausgabe-Signalliste

Anschluß-Nr.	Anschlußnummer entspricht dem analogen Ausgangskanal, gültige Werte 1..6. Soll mehr als ein Kanal ausgegeben werden, muß die Anzahl der Kanäle bei Listenlänge eingetragen werden (zuvor sollte aber die Anschlußnummer für den ersten Kanal gesetzt werden, da die weiteren Kanäle automatisch um eins erhöht in die Liste eingetragen werden).
Listenlänge	Gibt die Anzahl der auszugebenden analogen Ausgänge an, der Standardwert ist 1.

4.3.4 DIG-Bit-Objekt

Die DIG-Bit-Objekte erfassen einen digitalen Eingangs- bzw. setzen einen digitalen Ausgangskanal.

Spezifische Einstellungen in der Signalliste

Anschluß-Nr.	Anschlußnummer entspricht dem digitalen Ein-/Ausgangskanal, gültige Werte 1..16. Soll mehr als ein Kanal angesprochen werden, muß die Anzahl der Kanäle bei Listenlänge eingetragen werden (zuvor sollte aber die Anschlußnummer für den ersten Kanal gesetzt werden, da die weiteren Kanäle automatisch um eins erhöht in die Liste eingetragen werden).
Listenlänge	Gibt die Anzahl der auszugebenden analogen Ausgänge an, der Standardwert ist 1.

4.3.5 DIG-Byte-Objekt

Die DIG-Byte-Objekte erfassen die digitalen Eingänge bzw. setzen die digitalen Ausgänge. Der Zugriff geschieht wortweise, d.h. alle Eingänge bzw. Ausgänge 16 Bit auf einmal.

Spezifische Einstellungen in der Signalliste

Anschluß-Nr.	Die Anschlußnummer ist auf 0 zu setzen.
Listenlänge	Die Anzahl ist auf den Standardwert 1 zu setzen.

5 ADwin-DIAdem-Auto - Befehle für Autosequenzen

Mit Hilfe von **DIAdem-Auto** können wiederkehrende Arbeitsabläufe, Standardauswertungen und Serienauswertung in Form von Autosequenzen automatisiert werden. Die automatische Abarbeitung mehrerer Befehle nacheinander kann mit Hilfe einer Autosequenz realisiert werden.

Für **ADwin**-Systeme stehen verschiedene Funktionen zur Verfügung, mit denen im Modus **DIAdem-Auto** auf das System zugegriffen werden kann.

5.1 ADwin-DIAdem-Auto Variablen

Die im folgenden genannten Variablen werden vom **ADwin**-Treiber bei der Ausführung von Autosequenzen genutzt und sind somit belegt.

Name	Typ	Kommentar
ADwin_Link	eGPIVARTYPELONG	I/O-Adresse an der das ADwin -System in den PC eingebunden ist.
ADwin_Btl	eGPIVARTYPESTRING	Pfad/Dateiname der BTL-Datei
ADwin_File	eGPIVARTYPESTRING	Pfad/Dateiname der ADbasic -Binär-Datei
ADwin_Proc	eGPIVARTYPELONG	Nummer eines zu startenden oder stoppenden Prozesses.
ADwin_Par	eGPIVARTYPELONG	Nummer des zu übertragenden Parameters.
ADwin_LValue	eGPIVARTYPELONG	Wert des Integer-Parameters
ADwin_FValue	eGPIVARTYPEDOUBLE	Wert des Floating-Point-Parameters
ADwin_Prot	eGPIVARTYPESTRING	Variablen zur Netzwerkkommunikation
ADwin_End	eGPIVARTYPESTRING	
ADwin_Server	eGPIVARTYPESTRING	
ADwin_Pword	eGPIVARTYPESTRING	

5.2 ADwin-DIAdem-Auto Befehle

5.2.1 ADBoot - Booten des ADwin-Systems

ADBoot()

Diese Funktion bootet ein **ADwin**-System. Genutzt werden die Variablen **ADwin_Btl** und **ADwin_Link**.

Beispiel:

```
ADwin_Btl:='c:\ADbasic3\ADwin9.btl'  
ADwin_Link:=$150;  
ADBoot();
```

5.2.2 ADBload - Laden eines ADbasic-Prozesses

ADBload(ADwin_File)

Diese Funktion lädt einen **ADbasic**-Prozeß auf das **ADwin**-System. Genutzt wird die Variable **ADwin_Link**. Beim Befehlsaufruf wird das Argument **ADwin_File** übergeben.

Beispiel:

```
ADwin_File:='c:\ADbasic3\test.t91'  
ADwin_Link:= $150;  
ADBload(ADwin_File);
```

5.2.3 Start_Process - Starten eines ADbasic-Prozesses

Start_Process(ADwin_Proc)

Diese Funktion startet einen **ADbasic**-Prozeß. Genutzt wird die Variable **ADwin_Link**. Beim Befehlsaufruf wird das Argument **ADwin_Proc** übergeben.

Beispiel:

```
Start_Process(1);
```

5.2.4 Stop_Process - Stoppen eines *ADbasic*-Prozesses

Stop_Process (ADwin_Proc)

Diese Funktion stoppt einen **ADbasic**-Prozeß. Genutzt wird die Variable **ADwin_Link**. Beim Befehlsaufruf wird das Argument **ADwin_Proc** übergeben.

Beispiel:

```
Stop_Process(1);
```

5.2.5 Get_Par - Lesen eines *ADbasic*-Integer-Parameters

Get_Par(ADwin_Par, ADwin_LValue)

Diese Funktion lädt einen Integer-Parameter des **ADwin**-Systems nach **DIAdem**. Genutzt wird die Variable **ADwin_Link**. Beim Befehlsaufruf wird das Argument **ADwin_Par** übergeben. Das Ergebnis wird in der Variablen **ADwin_LValue** abgelegt.

Beispiel 1:

```
get_par(1, ADwin_lvalue);  
l1:= ADwin_lvalue;
```

Beispiel 2:

```
get_par(1);  
l1:= ADwin_lvalue;
```

5.2.6 Get_FPar - Lesen eines *ADbasic*-Floating-Point-Parameters

Get_FPar (ADwin_Par, ADwin_FValue)

Diese Funktion lädt einen Floating-Point-Parameter des **ADwin**-Systems nach **DIAdem**. Genutzt wird die Variable **ADwin_Link**. Beim Befehlsaufruf wird das Argument **ADwin_Par** übergeben. Das Ergebnis wird in der Variablen **ADwin_FValue** abgelegt.

Beispiel 1:

```
get_fpar(1, ADwin_fvalue);  
r1:= ADwin_fvalue;
```

Beispiel 2:

```
get_fpar(1);  
r1:= ADwin_fvalue;
```

5.2.7 Set_Par - Schreiben eines *ADbasic*-Integer-Parameters

`Set_Par(ADwin_Par, ADwin_LValue)`

Diese Funktion lädt einen Wert von **DIAdem** in einen Integer-Parameter des **ADwin**-Systems. Diese Funktion nutzt die Variable **ADwin_Link**. Beim Befehlsaufruf werden die Argumente **ADwin_Par** und optional **ADwin_LValue** übergeben.

Beispiel 1:

```
set_par(1,12345);
```

Beispiel 2:

```
ADwin_lvalue:=12345;  
set_par(1, ADwin_lvalue);
```

Beispiel 3:

```
l1:=12345;  
set_par(1,l1);
```

5.2.8 Set_FPar - Schreiben eines *ADbasic*-Floating-Point-Parameters

`Set_FPar(ADwin_Par, [ADwin_FValue])`

Diese Funktion lädt einen Wert von **DIAdem** in einen Floating-Point-Parameter des **ADwin**-Systems. Diese Funktion nutzt die Variable **ADwin_Link**. Beim Befehlsaufruf werden die Argumente **ADwin_Par** und optional **ADwin_FValue** übergeben.

Beispiel 1:

```
set_fpar(1,12.345);
```

Beispiel 2:

```
ADwin_fvalue:=12.345;  
set_fpar(1, ADwin_fvalue);
```

Beispiel 3:

```
r1:=12.345;  
set_fpar(1,r1);
```

5.2.9 AD_Connect / AD_Disconnect - Netzwerkkommunikation mit ADwin-Systemen

Ein **ADwin**-System, das mit einem Netzwerkrechner (z. B. LAN, ISDN, Internet, ...) verbunden ist, kann von jedem beliebigen Rechner des Netzwerkes (auch von mehreren gleichzeitig) ferngesteuert und überwacht werden. Der Rechner, mit dem das **ADwin**-System direkt verbunden ist, wird im folgenden Server genannt. Ein Rechner, der über das Netzwerk auf das **ADwin**-System zugreifen will, wird als Arbeitsplatzrechner bezeichnet.

Damit ein Arbeitsplatzrechner auf ein **ADwin**-System in einem Serverrechner über ein Netzwerk zugreifen kann, muß auf dem Server das - im Lieferumfang von **ADbasic** enthaltene - Programm **ADserver** gestartet werden.

Auf dem Arbeitsplatzrechner kann nun unter **DIAdem** über den Autosequenz-Befehl **AD_Connect** eine Netzwerkverbindung hergestellt werden.

```
AD_Connect(ADwin_Prot,ADwin_Server,ADwin_End, ADwin_Pword)
```

ADwin_Prot

Tragen Sie aus dieser Liste das in Ihrem Netzwerk installierte Protokoll aus (siehe Tabelle).

ADwin_End

Angabe des Endpunktes des verwendeten Netzwerkprotokolls (siehe Tabelle).

Protocol-Typ	ADwin_Prot	ADwin_End
NetBios über NetBEUI	ncacn_nb_nb	Zahl zwischen 32 und 255
IPX über NetBEUI	ncacn_nb_ipx	Zahl zwischen 32 und 255
TCP über NetBEUI	ncacn_nb_tcp	Zahl zwischen 32 und 255
TCP über IP	ncacn_ip_tcp	16 Bit Zahl
Named Pipes	ncacn_np	Name
SPX	ncacn_spx	16 Bit Zahl
DECnet	ncacn_dnet_nsp	Objekt-Name oder Nummer mit der Kennzeichnung # (z. B. #27)

ADwin_Pword

Dieses Kennwort wird bei jedem Verbindungsaufbau abgefragt, zu beachten ist die Eingabe der Groß- und Kleinschreibung.

ADwin_Server

Der Name des Server-PCs im Netzwerk.

AD_Disconnect()

Dieser Befehl dient zum Beenden der Verbindung.

Beispiel:

```
AD_Connect("ncacn_ip_tcp" , "PentPro" , "200" , "xyz" );  
...  
...  
AD_DisConnect( ) ;
```


6 ADwin-DIAdem-Paket-Treiber

Zur Kommunikation mit **ADwin**-Systemen steht ein **DIAdem-DAC** Pakettreiber zur Verfügung:

Diese Betriebsart nutzt die Fähigkeiten des **ADwin**-Systems voll aus. Auf dem **ADwin**-System laufen verschiedene Regelungen, Datenerfassungen, online-Verarbeitungen, Signalerzeugungen, etc. ab. Diese Funktionen nutzen globale Variablen und Datensätze, auf welche **DIAdem** über verschiedene Objekte zugreifen kann. Zusätzlich gibt es Funktionen zum Starten, Stoppen, Booten, Laden von Programmen.

In der Aktionsliste finden Sie hinter dem Knopf



die möglichen Funktionen, um auf **ADwin**-Systeme zuzugreifen.

6.1 ADwin-DIAdem-DAC-Objekte (Paketverarbeitung) zur Kommunikation mit ADbasic-Prozessen

Diese Paketfunktionen erlauben den Datenaustausch mit **ADbasic**-Programmen.

Funktion	Beschreibung
ADwin - Autoinit	Booten, Programmdownload, Starten von ADbasic -Programmen
ADwin - Prozeß	Booten, Programmdownload, Starten und Stoppen von ADbasic -Programmen, Überwachung des Software Triggers Activate_PC, sowie der Einstellung des Process-Delays (Zykluszeit eines ADbasic -Prozesses)
ADwin - Parameter	Austausch von globalen Variablen
ADwin - Data	Austausch von globalen Datensätzen (Arrays)
ADwin - Netz	Betrieb von ADwin -Systemen im Netz

6.1.1 ADwin System Process

Das Objekt **ADwin System Process** dient zur Steuerung der **ADbasic**-Prozesse über **DIADEM**.

Dieses Objekt erlaubt das Laden, Starten und Stoppen von Prozessen, das Setzen des Delays (Zykluszeit eines **ADbasic**-Prozesses), sowie die Auswertung des **ACTIVATE_PC**-Flags.

Die Linkadresse und die Prozeßnummer müssen immer angegeben werden, ein Dateiname nur bei Bedarf.

Folgende Einstellmöglichkeiten sind vorhanden:

The screenshot shows a dialog box titled "ADwin System Process". It has the following controls:

- Name:** A text input field.
- Load:** A checkbox.
- Activate:** A checkbox.
- Start:** A checkbox.
- Stop:** A checkbox.
- Delay:** A label above a text input field.
- Enable:** A checkbox.
- Linkaddress:** A dropdown menu showing "0x150".
- Process:** A dropdown menu.
- OK:** A button at the bottom.

Funktion	erlaubter Wert	Kommentar
Name	gültiger Pfad/Dateiname	Die folgen Funktionseigenschaften beziehen sich auf die unter Name angegebene Datei und/oder den unter Process angegebenen Prozeß.
Process	Auswahlliste (Wert 0 entspricht Prozeß 10)	
BOOT	ja/nein	Booten des ADwin -Systems
Load	ja/nein	Laden eines ADbasic -Prozesses
Start	ja/nein	Starten eines ADbasic -Prozesses
Stop	ja/nein	Stoppen eines ADbasic -Prozesses
Activate	ja/nein	Auslesen des Activate_PC Flags eines ADbasic -Prozesses. Das Flag wird durch einen ADbasic -Befehl gesetzt. Wird dieser Zustand erkannt, wird das Flag zurückgesetzt und der in DIADEM folgende (angeschlossene) Block ausgeführt.
Delay	ja/nein; Wert	Setzen der Zykluszeit für einen ADbasic -Prozeß
Enable; Wert		
Linkaddress	0x150, 0x190, ... (aus Liste)	Hardware, Adresse des ADwin -Systems

6.1.2 ADwin System Autolnit

Das Objekt **ADwin System Autolnit** wird mit jedem Start einer **DIAdem-DAC**-Messung einmal ausgeführt.

Das Objekt dient zum Booten eines **ADwin**-Systems, sowie zum Laden und Starten von Prozessen.

The dialog box 'ADwin System Autolnit' features the following controls:

- Name:** A text input field followed by a **Boot:** checkbox.
- Process 1 through 10:** Each has a text input field, a **Load:** checkbox, and a **Start:** checkbox.
- Linkaddress:** A dropdown menu currently showing '0x150'.
- Buttons:** An 'OK' button and a green triangle icon.

Funktion	erlaubter Wert	Kommentar
Name	gültiger Pfad/Dateiname	Booten des ADwin -Systems mit der unter Name angegebenen Datei.
Process 1..10	gültiger Pfad/Dateiname	Laden eines ADbasic -Prozesses auf ein ADwin -System mit der unter Process angegebenen Datei.
Load/Boot	ja/nein	Laden der spezifizierten Datei
Start	ja/nein	Starten der spezifizierten Datei
Linkaddress	0x150, 0x190, ... (aus Liste)	Hardware, Adresse des ADwin -Systems

6.1.3 ADwin System Parameter:

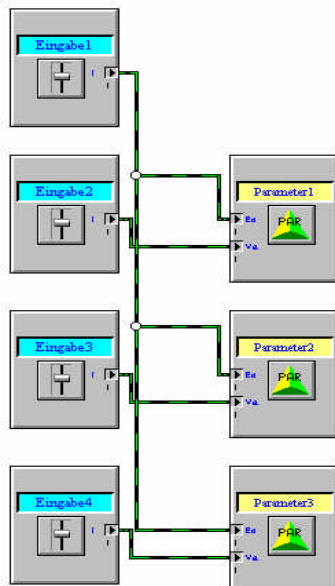
Das Objekt **ADwin System Parameter** dient zum Austausch von globalen Variablen (in **ADbasic**) mit **DIAdem**.

Es können sowohl Integer- als auch Floating-Point-Variablen übertragen werden.

Folgende Einstellmöglichkeiten sind vorhanden:



Funktion	erlaubter Wert	Kommentar
Par_No.	1 bis 80	Nummer des zu übergebenden Parameters
Write	ja/nein	Wahlmöglichkeit ob der Parameter beschrieben (Richtung ADwin) oder gelesen (Richtung DIAdem) werden soll
Float	ja/nein	Auswahl ob Float- oder Integervariablen übertragen werden. Der Typ muß mit dem der zu übertragenden globalen Variable übereinstimmen.
Linkaddress	0x150, 0x190, ... (aus Liste)	Hardware, Adresse des ADwin -Systems



Zum Laden eines Parameters von **DIAdem** auf das **ADwin**-System erhält der Parameter-Block einen Enable-Eingang. Dieser Eingang ist mit einem Taster zu verbinden. Dies ermöglicht, eine Anzahl von Parametern gleichzeitig auf die Karte zu senden. Die Parameter werden solange gesendet, bis der Taster zurückgesetzt wird. Sollen die Parameter kontinuierlich gesendet werden, so kann an der Stelle des Tasters ein Schalter ausgewählt werden.

6.1.4 ADwin System Data

Das Objekt **ADwin System Data** dient zum Austausch von globalen Datensätzen (in **ADbasic**) mit **DIAdem**.

Es können sowohl Integer- als auch Floating-Point-Datensätze übertragen werden.

Die Übertragung kann über SYNC-Einstellungen mit dem **ADbasic** Programm synchronisiert werden.

Die Angabe der Blocklänge gestattet eine variable Blocklänge, unabhängig von der Standard-Blocklänge in **DIAdem**.

Werden multiplexte Kanäle übertragen (d.h. mehrere Datenreihen werden in **ADbasic** in einem Data abgelegt), dann gibt dieses Objekt die Anzahl der Kanäle als Information in den Datenstrom.

Folgende Einstellmöglichkeiten sind vorhanden:

Funktion	erlaubter Wert	Kommentar
Number	1 bis 200	Nummer des zu übergebenden Datensatzes
Write	ja/nein	Wahlmöglichkeit ob der Datensatz beschrieben (Richtung ADwin) oder gelesen (Richtung DIAdem) werden soll
FIFO	ja/nein	Auswahl ob der Datensatz in ADbasic als FIFO deklariert wurde. Beide Einstellungen müssen übereinstimmen. Wenn der Datensatz als FIFO deklariert ist, muß DIAdem die Daten schneller auslesen, als diese in den FIFO geschrieben werden. Es muß in ADbasic überprüft werden, ob noch freie Stellen in dem FIFO sind, nur dann darf von ADbasic in das FIFO geschrieben werden.
Linkaddress	0x150, 0x190, ... (aus Liste)	Hardware, Adresse des ADwin -Systems

Erweiterte Einstellmöglichkeiten im Feld SYNC: Die Funktionen im Feld **SYNC** erlauben eine synchronisierte Datenübertragung mit **ADbasic**-Programmen. Der mit **PAR_Nr** spezifizierte Integer-Parameter wird ständig überprüft, ob der Wert **Value** eintritt. Wenn ja, wird der unter **Nummer** spezifizierte Datensatz übertragen. Anschließend wird der Parameter (**PAR_Nr**) auf den Wert **Reset** (zurück-) gesetzt.

Enable	ja/nein	aktiviert die Funktion SYNC
PAR_No.	1 bis 80	

Value	Integer-Bereich	
Reset	Integer-Bereich	
Block Length	1 .. max. global eingestellte DIAdem Blocklänge; Bei dem Wert "-1" wird die global eingestellte DIAdem -Blocklänge genutzt.	<p>Der Data-Block kann mit einer variablen Blocklänge arbeiten. Die Standardeinstellung der Blocklänge (wie in den Menüoptionen eingestellt.) entspricht dem Wert "-1" .</p> <p>Achtung! Die eingestellte Blocklänge darf maximal die selbe Länge haben wie die global eingestellte DIAdem-Blocklänge. Wird ein größerer Wert eingestellt wartet der Block auf weitere Daten, die nicht ankommen. Als Ergebnis werden keine Daten übertragen.</p> <p>Wird eine Blocklänge kleiner der global eingestellte DIAdem-Blocklänge angegeben, so wird der Rest des ankommenden Blockes verworfen.</p>
Channels	Integer-Bereich	Anzahl der multiplexten Kanäle. In ADbasic können in einem DATA die Anzahl N Signale multiplext geschrieben werden. Wird diese Anzahl N im Feld <i>Channels</i> eingetragen, separiert der DATA-Block die gelesenen Daten und stellt diese auf N Ausgängen zur Verfügung.

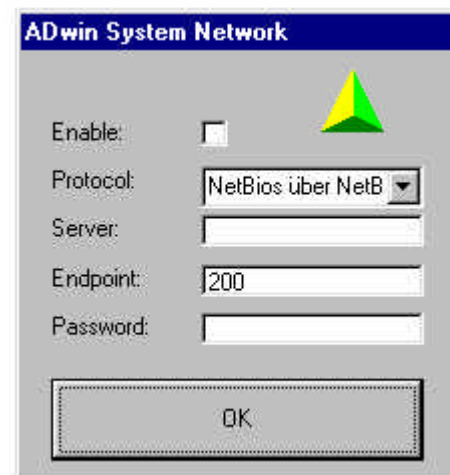
6.1.5 ADwin System Network

Ein **ADwin**-System, das mit einem Netzwerkrechner (z. B. LAN, ISDN, Internet, ...) verbunden ist, kann von jedem beliebigen Rechner des Netzwerkes (auch von mehreren gleichzeitig) ferngesteuert und überwacht werden. Der Rechner, mit dem das **ADwin**-System direkt verbunden ist, wird im folgenden Server genannt. Ein Rechner, der über das Netzwerk auf das **ADwin**-System zugreifen will, wird als Arbeitsplatzrechner bezeichnet.

Damit ein Arbeitsplatzrechner auf ein **ADwin**-System in einem Serverrechner über ein Netzwerk zugreifen kann, muß auf dem Server das - im Lieferumfang von **ADbasic** enthaltene - Programm **ADserver** gestartet werden.

Auf dem Arbeitsplatzrechner kann nun im Modus **DIAdem-DAC** (Paketverarbeitung) mit dem Objekt eine Netzwerkverbindung hergestellt werden.

Die Informationen zu den Einstellungen für Protocol, Server, Endpoint und Password entnehmen Sie bitte dem Abschnitt **ADwin-DIAdem-Auto** für den Befehl **AD_Connect**.



7 Demonstrationsbeispiele im DIAdem-DAC Einzelwertmodus

Die im folgenden beschriebenen Beispiele veranschaulichen den Umgang und die Art des Datenaustauschs zwischen **DIAdem-DAC** im Einzelwertmodus und dem in **ADbasic** programmierten **ADwin**-Systems.

Die Beispiele werden kontinuierlich ergänzt. Wir laden Sie als Anwender zu einem Besuch auf unserer Homepage ein (siehe Kapitel "Support"). Dort finden Sie neben der aktuellen Treiberversion auch weitere Programmier- und Anwendungsbeispiele.

7.1 Globale ADbasic-Datensätze und Variablen von ADbasic nach DIAdem laden

7.1.1 Übertragen von Parametern

ADbasic-Beispielprogramm minmax.BAS:

Dieser **ADbasic**-Prozeß erfaßt den analogen Eingang Nr.1 und errechnet die Minima und Maxima des Eingangssignals über die Anzahl von Abtastungen, die in Par_4 angegeben sind. Der aktuelle Eingangswert wird in Par_1 abgelegt, Min. in Par_2 und Max. in Par_3.

```
DIM n, istwert, max, min AS INTEGER

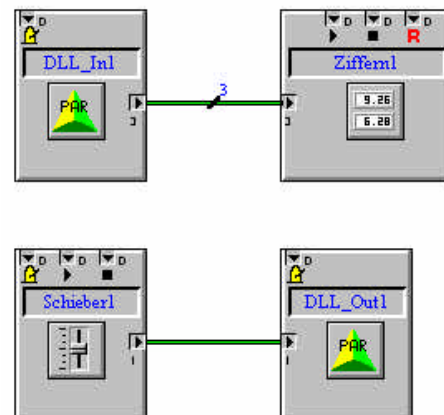
INIT:
n = 1                                'Zählvariable
min = 4095                           'Initialisierungswert für min
max = 0                              'Initialisierungswert für max

EVENT:
istwert = ADC(1)                     'Istwert ermitteln
IF (istwert < min) THEN min = istwert 'Minimalwert?
IF (istwert > max) THEN max = istwert 'Maximalwert?
n = n + 1                            'Zählvariable inkrementieren
IF (n>par_4) THEN                    'Anzahl der ADC-Werte ermittelt?
    PAR_2 = min                      'Minimalwert PAR_2 zuweisen
    PAR_3 = max                      'Maximalwert PAR_3 zuweisen
    n = 1                            'Zählvariable zurücksetzen
    min = 4095                       'Minimalwert zurücksetzen
    max = 0                           'Maximalwert zurücksetzen
ENDIF

par_1=istwert
```

DIAdem-Schaltplan MinMax.DAC:

Diese Schaltung dient zum Auslesen und Anzeigen der Parameter PAR_1 (Abtastwert), PAR_2 (Minimum) und PAR_3 (Maximum). In **DIAdem** wird nur ein Par-Objekt genutzt, um drei Parameter zu übertragen. Über den Schieber wird die Anzahl der Abtastungen an Par_4 übergeben.



7.1.2 Übertragen von Datensätzen

ADbasic-Beispielprogramm ADC_Data.BAS:

Dieser **ADbasic**-Prozeß erfaßt die analogen Eingänge Nr.1 bis 16 und legt die Ergebnisse der Abtastungen in Data_1 Index 1 bis 16.

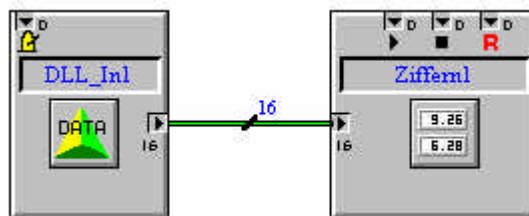
```
dim data_1[16] as integer

event:

'Einlesen der 16 analogen Eingänge in data_1
data_1[1]=adc(1)
data_1[2]=adc(2)
data_1[3]=adc(3)
data_1[4]=adc(4)
data_1[5]=adc(5)
data_1[6]=adc(6)
data_1[7]=adc(7)
data_1[8]=adc(8)
data_1[9]=adc(9)
data_1[10]=adc(10)
data_1[11]=adc(11)
data_1[12]=adc(12)
data_1[13]=adc(13)
data_1[14]=adc(14)
data_1[15]=adc(15)
data_1[16]=adc(16)
```

DIAdem-Schaltplan ADC_Data.DAC:

Diese Schaltung dient zum Einlesen der Data_1 Einträge 1 bis 16.



7.1.3 Übertragen von FIFO-Datensätzen im Hardwaretakt

ADbasic-Beispielprogramm FIFO_Data.BAS:

Dieser **ADbasic**-Prozeß erfaßt die analogen Eingänge Nr.1 bis 4, alle digitalen Eingänge und die Zähler 1, 2 und legt die Ergebnisse der Abtastungen in den FIFO-Datensatz 1 ab.

```
dim data_1[100000] as integer as fifo

init:
fifo_clear(1)

event:
if (fifo_full(1)<99990)THEN
'Einlesen von 4 analogen Eingänge in data_1
data_1=adc(1)
data_1=adc(3)
data_1=adc(5)
data_1=adc(7)

'Einlesen der 16 digitalen Eingänge als Word
data_1=digin_word()

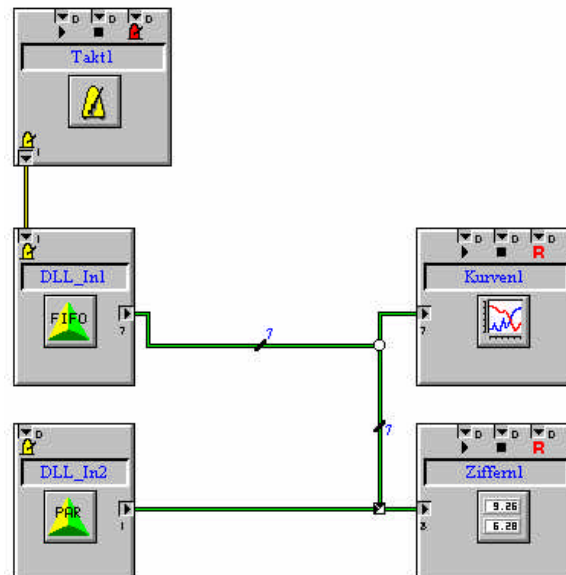
'Einlesen von 2 Zählern
data_1=co4_read(1)
data_1=co4_read(2)

'Anzahl der freien Plätze im FIFO
par_1=fifo_full(1)

ENDIF
```

DIAdem-Schaltplan FIFO_Data.DAC:

Die Messung der Eingänge geschieht im Modus-Hardwaretakt. Das FIFO-Objekt liest den Datensatz ein. Das Par-Objekt liest die Anzahl der freien Plätze des FIFOs.



8 Demonstrationsbeispiele im DIAdem-Paketmodus

Die im folgenden beschriebenen Beispiele veranschaulichen den Umgang und die Art des Datenaustauschs zwischen **DIAdem-DAC** im Paketmodus und dem in **ADbasic** programmierten **ADwin**-System.

8.1 Globale ADbasic-Datensätze und Variablen von ADbasic nach DIAdem laden

8.1.1 FIFO-Version 2, multiplexte Daten

ADbasic-Beispielprogramm adc_data_02.BAS:

Data_1 wird als FIFO deklariert. Im INIT-Abschnitt, (wird nur einmal nach dem Start ausgeführt) , wird das FIFO zurückgesetzt und Parameter PAR_1 gelöscht. Im EVENT-Abschnitt, (wird zyklisch, timergesteuert ausgeführt), wird zunächst überprüft, ob noch min. 4 Speicherplätze im FIFO frei sind. Wenn ja, werden 4 analoge Eingänge gemessen und nacheinander im FIFO abgelegt, PAR_1 um 1 erhöht und die Anzahl der verbleibenden freien Speicherstellen in PAR_2 abgelegt.

```
REM      adc_data_02.BAS
REM      Dimensionieren eines Datensatzes als FIFO
dim data_1[40000] as integer as fifo

init:
fifo_clear(1) 'Reset des FIFOs
par_1=0

event:
if (fifo_empty(1)>3) then      'Abfrage, ob noch ausreichend Speicherstellen frei im FIFO
    data_1 = adc(1)           'Analog Eingang 1 konvertieren und im FIFO 1 speichern
    data_1 = adc(3)           'Analog Eingang 3 konvertieren und im FIFO 1 speichern
    data_1 = adc(5)           'Analog Eingang 5 konvertieren und im FIFO 1 speichern
    data_1 = adc(7)           'Analog Eingang 7 konvertieren und im FIFO 1 speichern

    par_1=par_1+1              'Globale Variable PAR_1 wird mit jedem Abtastschritt um 1 erhöht
    par_2=fifo_empty(1)        'Globale Variable PAR_2 enthält die Anzahl freier Stellen im FIFO
endif
```

DIAdem-Schaltplan ADC_DATA_02.DAC:

Diese Schaltung dient zum Auslesen und Anzeigen von FIFO-1 und von PAR_1 (Abtastschritt-Nr.) und PAR_2 (Anzahl freier Stellen im FIFO). Nach dem Auslesen von FIFO-1 werden die Daten entsprechend der Kanalanzahl in 4 Kanäle getrennt. In dieser Variante der Datenübertragung wird nur ein Data-Block in **DIAdem** genutzt, um eine höhere Anzahl von Kanälen zu übertragen. Dieser Block trägt wesentlich zur Übersichtlichkeit der Schaltpläne bei. Dieser Block kann überall dort genutzt werden, wo mehrere Kanäle (analog und/oder digital) im **ADwin**-System erfaßt werden.

8.1.2 FIFO-Version 1, nichtmultiplexte Daten

ADbasic-Beispielprogramm `adc_data_01.BAS`:

`Data_1` bis `Data_4` als FIFO deklariert. Im INIT-Abschnitt, (wird nur einmal nach dem Start ausgeführt), werden die FIFOs zurückgesetzt und Parameter `PAR_1` gelöscht. Im EVENT-Abschnitt, (wird zyklisch, timergesteuert ausgeführt), wird zunächst überprüft, ob noch min. ein Speicherplatz im FIFO frei ist. Wenn ja, werden 4 analoge Eingänge gemessen und in den 4 FIFOs abgelegt, `PAR_1` um 1 erhöht und die Anzahl der verbleibenden freien Speicherstellen in `PAR_2` abgelegt.

```
REM    Dimensionieren von Datensätzen als FIFO
dim data_1[10000] as integer as fifo
dim data_2[10000] as integer as fifo
dim data_3[10000] as integer as fifo
dim data_4[10000] as integer as fifo

init:
fifo_clear(1) 'Reset des FIFOs 1
fifo_clear(2) 'Reset des FIFOs 2
fifo_clear(3) 'Reset des FIFOs 3
fifo_clear(4) 'Reset des FIFOs 4

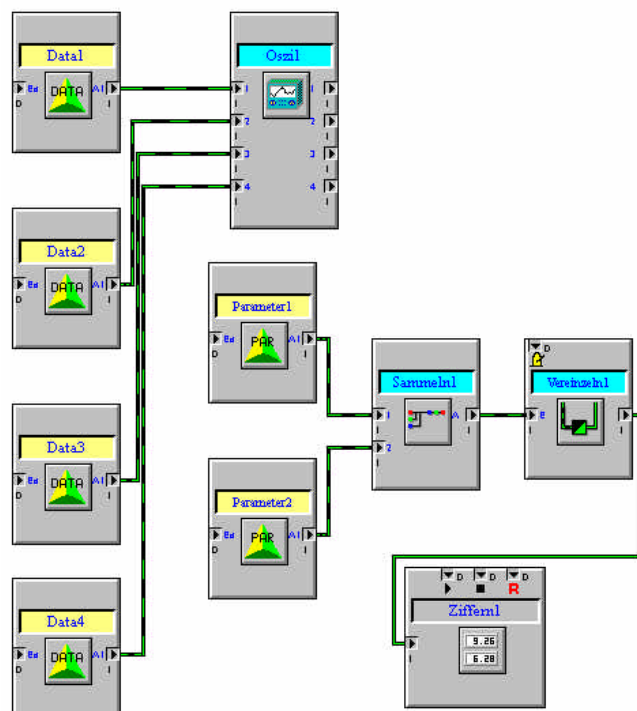
par_1=0

event:
if (fifo_empty(1)>1) then      'Abfrage, ob noch ausreichend Speicherstellen frei im FIFO
    data_1 = adc(1)           'Analog Eingang 1 konvertieren und im FIFO speichern
    data_2 = adc(3)           'Analog Eingang 3 konvertieren und im FIFO speichern
    data_3 = adc(5)           'Analog Eingang 5 konvertieren und im FIFO speichern
    data_4 = adc(7)           'Analog Eingang 7 konvertieren und im FIFO speichern

    par_1=par_1+1             'Globale Variable PAR_1 wird mit jedem Abtastschritt um 1 erhöht
    par_2=fifo_empty(1)       'Globale Variable PAR_2 enthält die Anzahl freier Stellen im FIFO
endif
```

DIAdem-Schaltplan `ADC_DATA_01.DAC`:

Diese Schaltung dient zum Auslesen und Anzeigen der 4 Datensätze und von `PAR_1` (Abtastschritt-Nr.) und `PAR_2` (Anzahl freier Stellen im FIFO). Zum Auslesen der 4 Datensätze werden 4 einzelne Data-Blöcke in **DIAdem** genutzt.



8.1.3 Version 3, mit Variablen zur Synchronisation

ADbasic-Beispielprogramm adc_data_03.BAS:

Data_1 und Data_2 wird deklariert (hier nicht als FIFO). Im INIT-Abschnitt, (wird nur einmal nach dem Start ausgeführt), wird Laufindex i auf 1 zurückgesetzt und Parameter PAR_1 gelöscht. PAR_1 dient als Flag, ob der Puffer voll ist. Im EVENT-Abschnitt, (wird zyklisch, timergesteuert ausgeführt), wird zunächst überprüft, ob das Flag PAR_1 gesetzt wurde. Wenn nein, werden 2 analoge Eingänge gemessen und in den Datensätzen abgelegt, der Laufindex i um 1 erhöht. Erreicht der Laufindex die vorgegebene Marke von 1001, d.h. die Anzahl der vorgegebenen Abtastungen ist erreicht, der Puffer ist voll, dann wird Flag PAR_1 auf 1 gesetzt.

(Hinweis: PAR_1 wird hier zwar als Flag genutzt, ist aber in **ADbasic** eine globale Integer-Variable)

```
REM adc_data_03.BAS
REM Dimensionieren von Datensätzen
dim data_1[10000] as integer
dim data_2[10000] as integer

REM i wird als Laufindex genutzt
dim i as integer

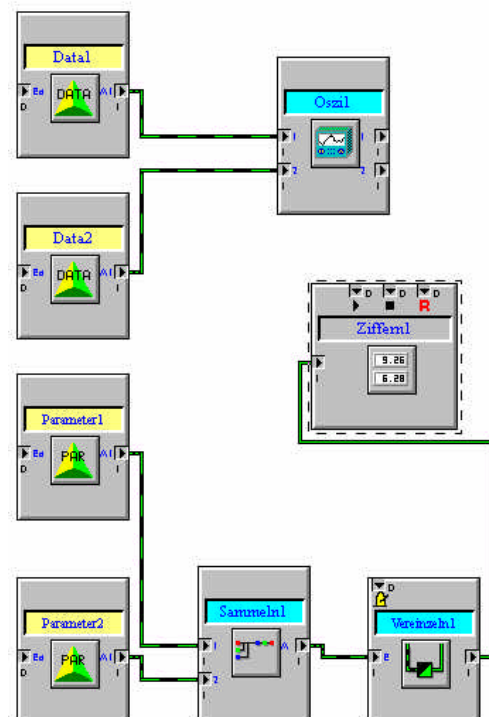
init:
i=1
par_1=0

event:
if (par_1=0) then
    data_1[i] = adc(1)      'Abfrage, ob Flag PAR_1 gesetzt
    data_2[i] = adc(3)      'Analog Eingang 1 konvertieren und im Puffer-1 speichern
                           'Analog Eingang 3 konvertieren und im Puffer-2 speichern

    i=i+1                  ' Laufindex erhöhen
    if (i=1001) then
        i=1                ' wenn i=1001 dann ist Puffer-1 bis 4 voll
        par_1=1            ' Flag PAR_1 setzen (wird von DIAdem nach dem Auslesen
                           ' zurückgesetzt)
    endif
    par_2=i                ' PAR_2 zeigt den Laufindex an
endif
endif
```

DIAdem-Schaltplan ADC_DATA_03.DAC:

Diese Schaltung dient zum Auslesen und Anzeigen der beiden Datensätze und von PAR_1 (Flag). Zum Auslesen der Datensätze werden einzelne Data-Blöcke in **DIAdem** genutzt.



9 Support

Wenn Sie bereits bei der Installation Probleme haben, ziehen Sie bitte die Dokumentation zu Ihrem **ADwin**-System zu Rate. Überprüfen Sie, ob alle Einstellungen richtig und vollständig durchgeführt wurden. Kontrollieren Sie bitte, ob die Software richtig installiert wurde. Prüfen Sie auch, ob die Basisadresse, der Prozessortyp etc. der **ADwin**-Systeme richtig in der Software angegeben sind..

Sollten Ihre Probleme weiterhin bestehen, wenden Sie sich bitte an den Lieferanten Ihres ADwin-Systems. Dort erhalten Sie umgehende und kompetente Hilfe.

Jäger Computergesteuerte Meßtechnik GmbH
Rheinstraße 4
D-64653 Lorsch
Tel.: 0 62 51 - 9 63 20
Fax: 0 62 51 - 5 68 19
E-Mail: support@adwin.de
Homepage: www.ADwin.de

GfS mbH Aachen
Pascalstraße 17
D-52076 Aachen
Tel.: 0 2408 - 6011
Fax: 0 2408 - 6019
E-Mail: gfs@gfs-ac.de
Homepage: www.gfs-ac.de

Benötigen Sie die neuesten Treiber oder suchen Sie neue Anwendungsbeispiele für **ADwin**-Systeme im Zusammenarbeit mit **DIAdem**, so besuchen Sie doch einmal unsere Homepage.

Sollten Sie Fragen oder Anregungen haben, so senden Sie uns bitte ein e-mail oder FAX, oder rufen Sie uns einfach an.

Mit freundlichen Grüßen

Ihre Jäger Computergesteuerte Meßtechnik GmbH