

# ***ADbasic***

## ***Tutorial Programmierbeispiele***

**Version 1.3**

**August 1998**

***Mikrosekundengenaue Echtzeitverarbeitung***

***- schnell und einfach programmiert***

## Inhaltsverzeichnis

<b>1 Wichtige Hinweise</b>	<b>3</b>
1.1 Aufbau dieser Dokumentation	3
1.2 Gebrauch der Beispielprogramme	3
<b>2 Test der Installation und erste Schritte mit <i>ADbasic</i></b>	<b>4</b>
<b>3 Beispielprogramme</b>	<b>6</b>
3.1 A/D- und D/A-Wandlung	6
3.2 Online-Auswertung von Meßwerten	7
3.3 Digitaler P-Regler	8
3.4 Datenaustausch mit DATA	10
3.5 Kontinuierliche Datenübertragung mit FIFO	11
3.6 Digitaler PID-Regler	12
3.6.1 Digitaler PID-Regler für <b>ADwin</b> -Systeme mit T8- oder T9-(ADSP-)Prozessor	12
3.6.2 Digitaler PID-Regler für <b>ADwin</b> -Systeme mit T4- oder T5-Prozessor	14
<b>4 Index</b>	<b>17</b>

## 1 Wichtige Hinweise

### 1.1 Aufbau dieser Dokumentation

Die auf den folgenden Seiten aufgeführten Beispielprogramme sollen Ihnen bei den ersten Schritten beim Programmieren mit **ADbasic** helfen. Beachten Sie bitte, daß Sie den vollen Leistungsumfang von **ADbasic** erst im alltäglichen Umgang mit **ADbasic** unter Zuhilfenahme des **ADbasic**-Handbuchs erlernen werden.

In Kapitel 2 finden Sie eine Anleitung, welche prinzipiellen Schritte Sie bei der Programmierung durchführen müssen. Das Programmierbeispiel ist bewußt einfach gehalten, damit Sie sich zunächst einmal auf die Bedienung von **ADbasic** konzentrieren können.

Kapitel 3 enthält mehrere Unterkapitel in denen jeweils ein Beispielprogramm erklärt wird. Sehen Sie diese Beispielprogramme als eine erweiterte Erläuterung zum Einsatz der in **ADbasic** eingesetzten Programmierbefehle an.

Durch den Index haben wir versucht, Ihnen den Zugang zu dem jeweiligen Beispielprogramm, in dem „Ihre“ Programmieraufgabe behandelt wird, schnell zu ermöglichen.

### 1.2 Gebrauch der Beispielprogramme

- Diese Beispielsammlung wird ständig erweitert. Von fast jedem Programm gibt es in dem Verzeichnis `...ADbasic\Samples` den Programmcode. Da der Umfang einiger Programme so groß ist, daß sie in gedruckter Form nicht mehr übersichtlich sind, haben wir für diese Programme kein Programmlisting abgedruckt.
- Beachten Sie bitte, daß nicht jedes Beispielprogramm mit exakt der gleichen Syntax auf jedem **ADwin**-System funktioniert. Für welches **ADwin**-System das jeweilige Programm direkt übernommen werden kann, ist zu Anfang jedes Programms aufgeführt.
- Für das Austesten eines Beispielprogramms benötigen Sie in jedem Fall einen Prozessor der entweder mit einer **ADwin**-PC-Einsteckkarte verbunden ist, oder ein **ADwin-Pro**-System in dem sich ein Prozessormodul aus der **ADwin-Pro**-Serie befindet.  
Außerdem muß **ADbasic** zum Editieren der Programme und die **ADwin**-Treiber auf Ihrem Rechner installiert sein, sowie der **ADwin**-Treiber auf das **ADwin**-System geladen worden sein.
- Die Befehle zum Ansprechen eines **ADwin-Pro**-Systems mit **ADbasic** werden in speziellen INCLUDE-Dateien zur Verfügung gestellt. Diese Include-Dateien müssen Sie in Ihr **ADbasic**-Programm einbinden. Lesen Sie hierzu bitte unbedingt die Dokumentation mit dem Titel „**ADwin-Pro** : Systembeschreibung ; Programmierung in **ADbasic**“.

## 2 Test der Installation und erste Schritte mit **ADbasic**

Sie können die Funktion des **ADwin**-Systems und die korrekte Installation der **ADbasic**-Software sehr leicht überprüfen indem Sie das folgende **ADbasic**-Programm ausprobieren.

### Programmbeschreibung:

In dem Programm zum Testen der Installation wird eine Variable (PAR\_1) periodisch erhöht. Die Variable kann im Parameterfenster von **ADbasic** beobachtet werden.

- a) Bei **ADwin**-Systemen mit den Prozessoren T225, T405, T450, T805 wird der Prozeß alle 1000 µs einmal ausgeführt; d. h. die Variable PAR\_1 wird alle 1000 µs um den Wert 1 erhöht.
- b) Bei einem **ADwin**-System mit dem Prozessor ADSP21062 wird der Prozeß alle 25 µs einmal ausgeführt; d. h. die Variable PAR\_1 wird alle 25 µs um den Wert 1 erhöht.

### Programmierung:

- ◆ Starten Sie **ADbasic** durch einen Doppelklick auf das Programmsymbol

- ◆ Geben Sie das folgende Programm ein:

```
EVENT:  
PAR_1 = PAR_1 + 1
```

- ◆ Öffnen Sie das Fenster zum Einstellen der Prozeß-Optionen mit **Options** ⇒ **Process**.

Überprüfen Sie die Einstellung für die Prozeß-Priorität (Priority: High).

- ◆ Öffnen Sie das Fenster zum Einstellen der Compiler-Optionen mit **Options** ⇒ **Compiler**.

Nehmen Sie hier die folgenden Einstellungen vor:

Processor: Prozessortyp, den Sie in Ihrem **ADwin**-System einsetzen.


Autostart: Yes

Debug Mode: No

Linkaddress: Basisadresse für den Linkadapter Ihres **ADwin**-Systems (standardmäßig ist auf Ihrem **ADwin**-System die Adresse 150H eingestellt).

Memory: Größe des Speichers mit dem Ihr **ADwin**-System ausgerüstet ist.


Übernehmen Sie Ihre Einstellungen indem Sie die OK-Taste betätigen.

- ◆ Klicken Sie auf dieses Symbol in der Symbolleiste:  (alternativ: **Project** ⇌ **Boot ADwin**)  
➔ Der **ADwin**-Treiber wird auf den Prozessor Ihres Systems geladen (Statuszeile: **ADwin** is booted).

## Fehlerbeseitigung:

Wenn der **ADwin**-Treiber nicht auf Ihr System geladen wurde und Sie anstatt dessen eine Fehlermeldung bekommen, überprüfen Sie bitte, ob

- das **ADwin**-System hardwaremäßig mit dem PC verbunden ist,
  - das **ADwin**-System richtig konfiguriert ist (Prozessortyp, Speichergröße, Linkadresse),
  - die **ADwin**-Treibersoftware installiert wurde, und in dem Menü **Directory** (**Options** ⇌ **Directory**) der korrekte Pfad für die BTL-Datei angegeben ist,
  - die richtigen Dateien für das entsprechende Betriebssystem installiert wurden (die DLL-Dateien im Windows-Verzeichnis vorhanden sind).
- ◆ die Einstellungen im Menü **Options** den beschriebenen Einstellungen entsprechen. Öffnen Sie das Parameterfenster mit **Options** ⇌ **Parameter**.  
Überprüfen Sie den Eintrag in dem 'Delay'-Feld (Delay: 1000).

- ◆ Klicken Sie auf dieses Symbol in der Symbolleiste:  (alternativ: **Project** ⇌ **Compile**)  
➔ Das aktuelle Programm wird übersetzt (kompiliert), auf das **ADwin**-System übertragen und gestartet.

## 3 Beispielprogramme

### 3.1 A/D- und D/A-Wandlung

#### Hardwarevoraussetzungen:

Für dieses **ADbasic**-Programm benötigen Sie eine **ADwin**-Karte oder eine **ADwin-light**-Karte.

#### Programmbeschreibung:

Dieses Programm gibt über den D/A-Wandler (DAC) eine Spannung aus. Diese Spannung wird anschließend vom A/D-Wandler (ADC) gemessen. Verbinden Sie dazu den analogen Ausgang 1 mit dem analogen Eingang 1 auf der 37-poligen D-SUB-Buchse auf der Rückseite der **ADwin**-Karte.

**! Achtung:** Bei den analogen Eingängen der **ADwin-light**-Karte handelt es sich um differentielle Eingänge. Es muß für jeden Eingang auch der entsprechende Masseeingang angeschlossen werden.

#### Programmierung:

♦ *Geben Sie folgendes Programm ein:*

```
EVENT :  
DAC(1, PAR_1)  
PAR_2 = ADC(1)
```

Der Befehl `DAC(1, PAR_1)` gibt den Wert der globalen Variablen 1 auf dem analogen Ausgang 1 aus. Die Programmzeile `PAR_2 = ADC(1)` mißt den Wert, der am analogen Eingang 1 anliegt und speichert ihn in der globalen Variablen 2.

Der ADC setzt die anliegende Spannung in einen Wert zwischen 0 und 4095 um. Der Meßbereich des ADC ist werkseitig auf -10 Volt bis +10 Volt eingestellt. In diesem Fall entspricht eine Spannung von 0 Volt dem Wert 2048. Werden z. B. 5 Volt vom ADC gemessen, so liefert dieser den Wert 3072. Dieselbe Umrechnung gilt auch für den DAC.

♦ *Klicken Sie in der Toolbar auf das Symbol zum Start des Compilers. Rufen Sie im Menü **Options** das Fenster **Parameter** auf.*

Wenn Sie jetzt den Wert der globalen Variablen 1 ändern, spiegelt sich diese Änderung sofort im Wert der Variablen 2 wider. ('Senden' nicht vergessen.)

## 3.2 Online-Auswertung von Meßwerten

### Hardwarevoraussetzungen:

Für dieses **ADbasic**-Programm benötigen Sie eine **ADwin**-Karte oder eine **ADwin-light**-Karte.

### Programmbeschreibung:

Das Programm sucht den Maximal- und Minimalwert aus 1000 Messungen von ADC1 und schreibt das Ergebnis in die Variablen `PAR_1` (Maximalwert) und `PAR_2` (Minimalwert).

### Programmierung:

- ♦ Öffnen Sie von **ADbasic** aus das Programm `BAS_DMO1.BAS` in dem Verzeichnis `...ADbasic\Samples`:

➔ Das beschriebene Programm zur Online-Auswertung von Meßwerten wird geöffnet.

- ♦ Führen Sie die Einstellungen für den von Ihnen verwendeten Prozessor durch, booten Sie Ihr **ADwin**-System, compilieren Sie das Programm und starten Sie es.

Hinweis: Das Programm `BAS_DMO1.BAS` enthält einen Befehl zur Kommunikation mit TestPoint. Wenn TestPoint nicht auf Ihrem Rechner installiert ist, dann löschen Sie diese Zeile aus dem Beispielpogramm und geben den entsprechenden Befehl für Ihr Visualisierungsprogramm ein (siehe jeweilige Treiberdokumentation).

```
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>1000) THEN                   '1000 Werte ermittelt?
    PAR_1 = min                    'Minimalwert PAR_1 zuweisen
    PAR_2 = max                    'Maximalwert PAR_2 zuweisen
    n = 1                          'Zählvariable zurücksetzen
    min = 4095                     'Minimalwert zurücksetzen
    max = 0                        'Maximalwert zurücksetzen
    ACTIVATE_PC                    'Test-Point aktivieren
ENDIF
```

## 3.3 Digitaler P-Regler

### Hardwarevoraussetzungen:

Für dieses **ADbasic**-Programm benötigen Sie eine **ADwin**-Karte oder eine **ADwin-light**-Karte.

### Programmbeschreibung:

Das Programm ist ein digitaler P-Regler. Der Sollwert wird mit `PAR_1` vorgegeben, die Verstärkung mit `PAR_9`. Dieses Programm kann mit bis zu 35 kHz ablaufen.

Parameter	Beschreibung
PAR_1	Sollwert in Digits
PAR_2	Istwert in Digits
PAR_3	Regeldifferenz in Digits
PAR_4	Stellgröße in Digits
PAR_9	Verstärkung des Reglers

Tabelle 1: Verwendete Parameter

### Programmierung:

- ♦ Öffnen Sie von **ADbasic** aus das Programm `BAS_DMO2.BAS` in dem Verzeichnis `...ADbasic\Samples`:

→ Das beschriebene Programm zur Realisierung eines digitalen P-Reglers wird geöffnet.

- ♦ Führen Sie die Einstellungen für den von Ihnen verwendeten Prozessor durch, booten Sie Ihr **ADwin**-System, compilieren Sie das Programm und starten Sie es.

```
DIM index AS INTEGER           'Array-Index
DIM w_alt AS INTEGER           'Speicher für alten Sollwert
DIM DATA_1[2000] AS INTEGER   'Zwischenspeicher der Regeldifferenz

INIT:
PAR_1 = 2048                   'Initialisierungswert für den Sollwert
PAR_9 = 100                    '          -"-          für die Verstärkung

EVENT:
PAR_2 = ADC(1)                 'Ermittlung des Istwertes
PAR_3 = PAR_1 - PAR_2          'Regeldifferenz ermitteln (Sollwert - Istwert)
PAR_4 = PAR_3 * PAR_9 + 2048    'Stellgröße berechnen ...
DAC(1,PAR_4)                   'und an DAC-1 ausgeben
```



```
' _____
'/Dieser Programmteil dient der Beobachtung und gehört nicht zum Regler!  \
  IF (PAR_1<>w_alt) THEN                                'Sollwertänderung?
    DATA_1[index] = PAR_3                              'Regelabweichung in Array schreiben
    index=index+1                                       'Index erhöhen
    IF (index>=2000) THEN                              'Array voll?
      ACTIVATE_PC                                       'Test-Point aktivieren
      index = 1                                         'Index zurücksetzen
      w_alt=PAR_1                                       'Alter Sollwert = Neuer Sollwert
    ENDIF
  ENDIF
'\_____/'
```

## 3.4 Datenaustausch mit DATA

### Hardwarevoraussetzungen:

Für dieses **ADbasic**-Programm benötigen Sie eine **ADwin**-Karte oder eine **ADwin-light**-Karte.

### Programmbeschreibung:

Das Programm mißt den analogen Eingang 1 und informiert nach 1000 Messungen den PC, der dann die Meßwerte abholen kann. Die Daten werden mit Hilfe eines DATA-Arrays übertragen. Dieser liest die 1000 Meßwerte aus.

Beachten Sie bitte, daß dem PC genügend Zeit gegeben werden muß, um alle Werte auslesen zu können; d. h. wenn mit dem Programm nach 1000 Messungen ein zweites Mal eine Messung durchgeführt werden soll, muß das Array größer gewählt werden.

### Programmierung:

- ♦ Öffnen Sie von *ADbasic* aus das Programm *BAS\_DMO3.BAS* in dem Verzeichnis *...ADbasic\Samples*.

→ Das beschriebene Programm zum Datenaustausch mit DATA wird geöffnet.

- ♦ Führen Sie die Einstellungen für den von Ihnen verwendeten Prozessor durch, booten Sie Ihr **ADwin**-System, compilieren Sie das Programm und starten Sie es.

Hinweis: Das Programm *BAS\_DMO3.BAS* enthält einen Befehl zur Kommunikation mit TestPoint. Wenn TestPoint nicht auf Ihrem Rechner installiert ist, dann löschen Sie diese Zeile aus dem Beispielprogramm und geben den entsprechenden Befehl für Ihr Visualisierungsprogramm ein (siehe jeweilige Treiberdokumentation).

```
DIM DATA_1[1000] AS INTEGER
```

```
DIM index AS INTEGER
```

```
INIT:
```

```
index = 1                                'Initialisierungswert der Zählvariable
```

```
EVENT:
```

```
DATA_1[index] = ADC(1)                   'ADC-Wert in Array schreiben
```

```
index = index + 1                         'Zählvariable inkrementieren
```

```
IF (index > 1000) THEN                    '1000 ADC-Werte ermittelt
```

```
    ACTIVATE_PC                           'Test-Point aktivieren
```

```
    END                                    'Prozeß beenden
```

```
ENDIF
```

## 3.5 Kontinuierliche Datenübertragung mit FIFO

### Hardwarevoraussetzungen:

Für dieses **ADbasic**-Programm benötigen Sie eine **ADwin**-Karte oder eine **ADwin-light**-Karte.

### Programmbeschreibung:

Das Programm mißt den analogen Eingang 1 kontinuierlich und informiert nach 1000 Messungen den PC. Das Programm wird dann beendet und der PC liest die 1000 Meßwerte aus.

Beachten Sie bitte, daß dem PC genügend Zeit gegeben werden muß, um alle Werte auslesen zu können; d. h. wenn mit dem Programm nach 1000 Messungen ein zweites Mal eine Messung durchgeführt werden soll, muß der FIFO größer dimensioniert werden.

### Programmierung:

- ♦ Öffnen Sie von **ADbasic** aus das Programm **BAS\_DMO4.BAS** in dem Verzeichnis **...ADbasic\Samples**.

→ Das beschriebene Programm zur kontinuierlichen Datenübertragung mit FIFO wird geöffnet.

- ♦ Führen Sie die Einstellungen für den von Ihnen verwendeten Prozessor durch, booten Sie Ihr **ADwin**-System, compilieren Sie das Programm und starten Sie es.

Hinweis: Das Programm **BAS\_DMO4.BAS** enthält einen Befehl zur Kommunikation mit TestPoint. Wenn TestPoint nicht auf Ihrem Rechner installiert ist, dann löschen Sie diese Zeile aus dem Beispielprogramm und geben den entsprechenden Befehl für Ihr Visualisierungsprogramm ein (siehe jeweilige Treiberdokumentation).

```
DIM DATA_1[1000] AS INTEGER AS FIFO
```

```
DIM index AS INTEGER
```

```
INIT:
```

```
index = 1                                'Initialisierungswert der Zählvariable
```

```
EVENT:
```

```
DATA_1 = ADC(1)                          'ADC-Wert in FIFO-Array schreiben
```

```
index = index + 1                        'Zählvariable inkrementieren
```

```
IF (index > 1000) THEN                    '1000 ADC-Werte ermittelt
```

```
    ACTIVATE_PC                           'Test-Point aktivieren
```

```
    END                                    'Prozeß beenden
```

```
ENDIF
```

## 3.6 Digitaler PID-Regler

### Hardwarevoraussetzungen:

Für diese **ADbasic**-Programme benötigen Sie eine **ADwin**-Karte oder eine **ADwin-light**-Karte. Das in Kapitel 3.6.1 beschriebene Programm ist für **ADwin**-Systeme mit T8- oder T9-(ADSP-)Prozessor und das in Kapitel 3.6.2 beschriebene Programm für **ADwin**-Systeme mit T2-, T4 oder T5-Prozessor geeignet.

Hinweis: In dem Programm wird davon ausgegangen, daß der Eingangsspannungsbereich auf  $\pm 10$  V eingestellt ist.

### 3.6.1 Digitaler PID-Regler für ADwin-Systeme mit T8- oder T9-(ADSP-)Prozessor

#### Programmbeschreibung:

Dieses Programm benutzt Floating-Point-Parameter und kann nur mit T8- und T9-CPU's benutzt werden. Obwohl auch T5-CPU's mit Floating-Point-Parameter rechnen können, diese aber bei diesem Prozessortyp vorher emuliert werden, benutzen Sie für die maximale Reglergeschwindigkeit bitte die Integer-Version BAS\_DM6I.BAS.

**ADwin**-Systeme mit T2- und T4-CPU's können ausschließlich mit der Integer-Version BAS\_DM6I.BAS programmiert werden.

Hinweis: Vor dem Starten des Reglers müssen die Reglereinstellungen in den Parametern stehen. Ansonsten werden die Reglerparameter auf die in dem Programm vorgesehenen Werte voreingestellt.

Parameter	Beschreibung
PAR_9	Sollwert in Digits
PAR_10	Istwert in Digits
PAR_11	Regeldifferenz in Digits
PAR_12	Stellgröße in Digits
PAR_13	Zykluszeit([1 $\mu$ s] für T8 und [25 ns] für ADSP)
FPAR_1	Verstärkung des Reglers
FPAR_2	Integrationszeit des Reglers (ti)
FPAR_3	Differentiationszeit des Reglers (td)

Tabelle 2: Verwendete Parameter

#### Programmierung:

♦ Öffnen Sie von **ADbasic** aus das Programm BAS\_DM6f.BAS in dem Verzeichnis ...ADbasic\Samples.

➔ Das beschriebene Programm zur Realisierung eines digitalen PID-Reglers wird geöffnet.

♦ Führen Sie die Einstellungen für den von Ihnen verwendeten Prozessor durch, booten Sie Ihr **ADwin**-System, compilieren Sie das Programm, stellen Sie im Parameterfenster von **ADbasic** die Reglerparameter ein und starten Sie das Programm.

Hinweis: Das Programm BAS\_DMO6f.BAS enthält einen Befehl zur Kommunikation mit TestPoint. Wenn TestPoint nicht auf Ihrem Rechner installiert ist, dann löschen Sie diese Zeile aus dem Beispielprogramm und geben den entsprechenden Befehl für Ihr Visualisierungsprogramm ein (siehe jeweilige Treiberdokumentation).

```
DIM yd_alt, w_alt AS INTEGER      'Zwischenspeicherung von div. Werten
DIM sum AS INTEGER                'Integrationswert
DIM d_yd AS FLOAT                'Deltawert der Regeldifferenz
DIM index AS INTEGER             'Array-Index
DIM DATA_1[2000] AS INTEGER     'Zwischenspeicher der Regeldifferenz

INIT:
SET_MUX(0)                       'Multiplexer auf Eingang 1 setzen
sum = 0                          'Integralwert auf Null setzen
PAR_11 = ADC(1)                  'Initialwert der Regelabweichung
index = 1                       'Initialwert für den Array-Index
'----- ACHTUNG: ti und zyklus dürfen nicht 0 sein. -----
IF (FPAR_2<1) THEN FPAR_2 = 10000 'FPAR_2 < 1 nicht zulassen
IF (PAR_13<40) THEN PAR_13 = 40   'GLOBALDELAY nicht kleiner als 40µs
GLOBALDELAY = PAR_13             'GLOBALDELAY setzen

EVENT:
START_CONV(1)                   'Messung starten

'----- Der eigentliche Regler: -----
'y =  KR      *( P      +      I      +      D      )
'|      |      |      |      |      |
'|      |      |      |      |      |
'|      |      |      |      |      |
PAR_12=FPAR_1*(PAR_11 + sum/(FPAR_2) + d_yd*FPAR_3) 'Stellgröße berechnen

DAC(1, PAR_12 + 2048)            'Stellwert ausgeben
yd_alt = PAR_11                  'Regelabweichung zwischenspeichern
WAIT_EOC(1)                     'Konvertierungsende abwarten
PAR_10 = READADC(1)              'Istwert ermitteln
PAR_11 = PAR_9 - PAR_10          'Regelabw. = Sollwert - Istwert
sum = sum + PAR_11               'Integralanteil berechnen

'-- Die Grenze für die Summe sollte ca. 5000 * FPAR_2 / FPAR_1 betragen. ---
IF (sum>2000000) THEN sum = 2000000 'Begrenzung für den Integralteil
IF (sum<-2000000) THEN sum = -2000000 'Begrenzung für den Integralteil

d_yd = PAR_11 - yd_alt           'Deltawert der Regeldiff. berechnen
```

```

'
'/ Dieser Programmteil dient der Beobachtung und gehört nicht zum Regler! \
  FPAR_9 = FPAR_9*0.9999 + PAR_11*0.0001 'Filter für Regelabw.
                                     '(nur Anzeige)
  IF (PAR_9<>w_alt) THEN              'Sollwertänderung?
    DATA_1[index] = PAR_11          'Regelabweichung in Array schreiben
    index=index+1                    'Index erhöhen
    IF (index>=2000) THEN             'Array voll?
      ACTIVATE_PC                     'Test-Point aktivieren
      index = 1                      'Index zurücksetzen
      w_alt=PAR_9                    'Alter Sollwert = Neuer Sollwert
    ENDIF
  ENDIF
' \

FINISH:
DAC 1, 2048                          'Ausgangsspannung von 0V ausgeben

```

### 3.6.2 Digitaler PID-Regler für ADwin-Systeme mit T2-, T4- oder T5-Prozessor

Dieses Programm benutzt Integer-Parameter und ist daher für **ADwin**-Systeme mit T2-, T4- und T5-CPU's vorgesehen.

Hinweis: Vor dem Starten des Reglers müssen die Reglereinstellungen in den Parametern stehen. Ansonsten werden die Reglerparameter auf die in dem Programm vorgesehenen Werte voreingestellt.

Parameter	Beschreibung
PAR_9	Sollwert in Digits
PAR_10	Istwert in Digits
PAR_11	Regeldifferenz in Digits
PAR_12	Stellgröße in Digits
PAR_13	Zykluszeit([1 µs] für T8 und [25 ns] für ADSP)
PAR_1	Verstärkung des Reglers
PAR_2	Integrationszeit des Reglers (ti)
PAR_3	Differentiationszeit des Reglers (td)

**Tabelle 3: Verwendete Parameter**

#### Programmierung:

- ♦ Öffnen Sie von **ADbasic** aus das Programm *BAS\_DM6I.BAS* in dem Verzeichnis *...ADbasic\Samples*.
  - ➔ Das beschriebene Programm zur Realisierung eines digitalen PID-Reglers wird geöffnet.
- ♦ Führen Sie die Einstellungen für den von Ihnen verwendeten Prozessor durch, booten Sie Ihr **ADwin**-System, compilieren Sie das Programm, stellen Sie im Parameterfenster von **ADbasic** die Reglerparameter ein und starten Sie das Programm.

Hinweis: Das Programm BAS\_DMO6I.BAS enthält einen Befehl zur Kommunikation mit TestPoint. Wenn TestPoint nicht auf Ihrem Rechner installiert ist, dann löschen Sie diese Zeile aus dem Beispielprogramm und geben den entsprechenden Befehl für Ihr Visualisierungsprogramm ein (siehe jeweilige Treiberdokumentation).

```
DIM yd_alt, w_alt AS INTEGER      'Zwischenspeicherung von div. Werten
DIM sum AS INTEGER                'Integrationswert
DIM d_yd AS INTEGER               'Deltawert der Regeldifferenz
DIM index AS INTEGER              'Array-Index
DIM DATA_1[2000] AS INTEGER      'Zwischenspeicher der Regeldifferenz

INIT:
SET_MUX(0)                        'Multiplexer auf Eingang 1 setzen
sum = 0                           'Integralwert auf Null setzen
PAR_11 = ADC(1)                   'Initialwert der Regelabweichung
index = 1                         'Initialwert für den Array-Index
'----- ACHTUNG: PAR_2 und PAAR_13 dürfen nicht 0 sein. -----
IF (PAR_2<1) THEN PAR_2 = 10000    'PAR_2 < 1 nicht zulassen
IF (PAR_13<100) THEN PAR_13 = 100  'GLOBALDELAY nicht kleiner als 100µs
GLOBALDELAY = PAR_13              'GLOBALDELAY setzen

EVENT:
START_CONV(1)                     'Messung starten

'----- Der eigentliche Regler: -----
' y = KR * ( P + I + D )
' |   |   |   |   |   |   |   |   |   |   |   |   |
' |   |   |   |   |   |   |   |   |   |   |   |   |
PAR_12=PAR_1*(PAR_11+sum*1000/PAR_2+d_yd*PAR_3/1000) 'Stellgröße berechnen

DAC(1, PAR_12/1000 + 2048)         'Stellwert ausgeben
yd_alt = PAR_11                    'Regelabweichung zwischenspeichern
WAIT_EOC(1)                       'Konvertierungsende abwarten
PAR_10 = READADC(1)                'Istwert ermitteln
PAR_11 = PAR_9 - PAR_10             'Regelabw. = Sollwert - Istwert
sum = sum + PAR_11                 'Integralanteil berechnen

'--- Die Grenze für die Summe sollte ca. 5000 * PAR_2 / PAR_1 betragen. ---
IF (sum>2000000) THEN sum = 2000000 'Begrenzung für den Integralteil
IF (sum<-2000000) THEN sum = -2000000 'Begrenzung für den Integralteil

d_yd = PAR_11 - yd_alt             'Deltawert der Regeldiff. berechnen
```

```
' _____
'/  Dieser Programmteil dient der Beobachtung und gehört nicht zum Regler! \
  IF (PAR_9<>w_alt) THEN                'Sollwertänderung?
    DATA_1[index] = PAR_11             'Regelabweichung in Array schreiben
    index=index+1                       'Index erhöhen
    IF (index>=2000) THEN                'Array voll?
      ACTIVATE_PC                       'Test-Point aktivieren
      index = 1                         'Index zurücksetzen
      w_alt=PAR_9                       'Alter Sollwert = Neuer Sollwert
    ENDIF
  ENDIF

'\_____

FINISH:
DAC 1, 2048                            'Ausgangsspannung von 0V ausgeben
```



## 4 Index

A/D und D/A-Wandlung 6  
BAS\_DM6f.BAS 12  
BAS\_DM6l.BAS 14  
BAS\_DMO1.BAS 7  
BAS\_DMO2.BAS 8  
BAS\_DMO3.BAS 10

BAS\_DMO4.BAS 11  
Datenaustausch mit DATA 10  
Datenübertragung mit FIFO 11  
digitaler PID-Regler 12, 14  
Online-Auswertung 7  
P-Regler 8