JÄGER
Computergesteuerte
Messtechnik GmbH

ADwin-Pro II

System- und Hardware-Beschreibung

ADwin-Pro II Hardware, Handbuch Feb. 2020

Hier finden Sie immer einen Ansprechpartner für Ihre Fragen:

Hotline:	(0 62 51) 9 63 20
Fax:	(0 62 51) 5 68 19
E-Mail:	info@ADwin.de
Internet	www.ADwin.de

JÄGER
Computergesteuerte
Messtechnik GmbH

Jäger Computergesteuerte
Messtechnik GmbH
Rheinstraße 2-4
D-64653 Lorsch

Inhaltsverzeichnis

Typografische Konventionen

Das „Achtung“-Zeichen steht bei Informationen, die auf Folgeschäden durch Fehlbedienung an der Hard- oder Software, am Messaufbau oder an Personen hinweisen.

Einen „Hinweis“ finden Sie bei

– Informationen, die für einen fehlerfreien Betrieb unbedingt beachtet werden müssen.

– Tipps und Ratschlägen für einen effizienten Betrieb.

Das Zeichen „Information“ verweist auf weiterführende Informationen in dieser Dokumentation oder andere Quellen wie Handbücher, Datenblätter, Literatur etc.

C:\ADwin\…

Dateinamen und -verzeichnisse sind in spitzen Klammern und im Schrifttyp Courier New angeben.

Programmtext

Programmanweisungen und Benutzer-Eingaben sind durch den Schrifttyp Courier New gekennzeichnet.

Var_1

Elemente eines Quelltextes wie Befehle, Variablen, Kommentar und sonstiger Text werden im Schrifttyp Courier New und farbig dargestellt.

In einem Datenwort (hier: 16 Bit) werden die Bits wie folgt nummeriert:

Bit-Nr.	15	14	13	…	1	0
Wert des Bits	2^{15}	2^{14}	2^{13}	…	2^1=2	2^0=1
Bezeichnung	MSB	-	-	-	-	LSB

Das ADwin-Pro II-System

1 Das *ADwin-Pro II*-System

Das *ADwin-Pro II*-System ist ein modular erweiterbares Prozessrechner-System. Je nach Anforderung können die verschiedenen Gehäuseformen mit bisherigen *ADwin-Pro*- und neuen *ADwin-Pro II*-Modulen bestückt werden.

Bei der Entwicklung des *ADwin-Pro II*-Systems wurde großer Wert auf die EMV-Verträglichkeit gelegt. Das *ADwin-Pro II*-System hat mit allen lieferbaren Ein- / Ausgabemodulen das CE-Zeichen und kann deshalb bei Bedarf auch nachträglich beliebig umkonfiguriert werden.

Ausrüstung mit Modulen

Jedes *ADwin-Pro II*-System benötigt ein Prozessormodul. Das Prozessormodul kommuniziert über Ethernet mit dem PC oder Notebook.

Um den vielfältigen Anforderungen bei Mess- und Steuerungsaufgaben zu entsprechen, kann das System mit folgenden Modulen ausgerüstet werden:

- analoge Eingabemodule und analoge Ausgabemodule
- digitale Eingabemodule und digitale Ausgabemodule
- Zähler
- Filter, Trennverstärker
- Verstärker für Thermoelemente und Temperaturfühler
- serielle Kommunikations-Schnittstellen: CAN, LIN, RSxxx, Feldbus, EtherCAT, FlexRay
- Speicher-/Lese-Modul für PCMCIA-Datenträger

Revisionsbezeichnung

Alle Module haben eine Revisionsbezeichnung auf der Frontseite, z. B. Rev. A2, Rev. B3, Rev. C3. Früher gelieferte Module sind nicht gekennzeichnet; sie besitzen den Revisionsstand „Rev. A“. *ADwin-Pro II*-Module tragen die Revisionsbezeichnung Rev. E1 oder höher.

Unterschiedliche Revisions-Buchstaben bedeuten unterschiedliche Modul-Eigenschaften und sind separat dokumentiert.

Der Revisionsbezeichnung angehängt ist eine untergeordnete Zählnummer, die für interne Zwecke der Firma Jäger Computergesteuerte Messtechnik GmbH verwendet wird.

2 Installation des *ADwin-Pro II*-Systems

Halten Sie bitte unbedingt die folgende Reihenfolge ein:

1. Beginnen Sie mit dem Handbuch „*ADwin*-Installation“:
   - *ADwin*-Software-Paket mit Schnittstellen-Treibern installieren.
   - Datenverbindung vom PC zum *ADwin*-System in Betrieb nehmen und auf korrekte Funktion prüfen.
     Die Anschlüsse für die Stromversorgung sind in Kapitel 4 “Gehäuse für das ADwin-Pro II-System“ beschrieben.
   - Beachten Sie die Hinweise in Kapitel 3 “Betriebliche Umgebung“.
2. Moduladressen einstellen siehe Kapitel 5.2 auf Seite 11.
3. Machen Sie die ersten Schritte mit dem *ADbasic*-Tutorial.
4. Programmieren in *ADbasic*:

   Das Handbuch *ADbasic* beschreibt die Echtzeit-Entwicklungsumgebung, den Aufbau eines *ADbasic*-Programms und gibt Hinweise für Optimierungen.

   Die *ADbasic*-Befehle finden Sie in in der Online-Hilfe der Entwicklungsumgebung oder in diesen Dokumenten:
   - Handbuch *ADbasic*: Grundlegende Befehle für Berechnungen, Programmstruktur und Prozesssteuerung.
   - Handbuch *ADwin-Pro II* Systembeschreibung: Befehle und Hinweise zum Ansprechen der Pro-Module.

Beachten Sie auch die Hinweise zu den Modulen in diesem Handbuch.

Bitte beachten Sie folgende Hinweise

Damit Ihr *ADwin*-System sicher arbeitet, halten Sie sich an die Informationen dieser und weiterführender Dokumentationen, auf die hier verwiesen wird.

Einschränkung der Anwendergruppe

Der Hersteller des in dieser Dokumentation beschriebenen Systems geht davon aus, dass an dem Gerät nur qualifiziertes Personal arbeitet.

> *Qualifiziertes Personal sind Personen, die aufgrund ihrer Ausbildung, Erfahrung und Unterweisung sowie ihrer Kenntnisse über einschlägige Normen, Bestimmungen, Unfallverhütungsvorschriften und Betriebsverhältnisse von dem für die Sicherheit der Anlage Verantwortlichen berechtigt worden sind, die jeweils erforderlichen Tätigkeiten auszuführen und die dabei mögliche Gefahren erkennen und vermeiden können. (Definition für Fachkräfte nach VDE 105 und IEC 60364)*.

Verfügbarkeit der Unterlagen

Diese Produktdokumentation und Unterlagen, auf die verwiesen wird, müssen stets verfügbar sein und konsequent beachtet werden. Für Schäden, die durch Missachtung der Informationen in dieser bzw. der weiterführenden Dokumentation entstehen, übernimmt die Firma *Jäger Computergesteuerte Messtechnik GmbH*, Lorsch, keine Haftung.

Rechtliche Grundlagen

Diese Dokumentation ist einschließlich aller Abbildungen urheberrechtlich geschützt. Reproduktion, Übersetzung sowie elektronische und fotografische Archivierung und Veränderung bedürfen der schriftlichen Genehmigung der Firma *Jäger Computergesteuerte Messtechnik GmbH*, Lorsch.

Fremdprodukte werden ohne Vermerk auf mögliche Patentrechte genannt, deren Existenz nicht auszuschließen ist.

Änderungen vorbehalten.

Hotline-Adresse siehe vordere Umschlagseite, innen.

Betriebliche Umgebung

3 Betriebliche Umgebung

Erdung

Das *ADwin-Pro II*-Gerät muss geerdet werden, um

– einen Massebezugspunkt für die Elektronik herzustellen und

– Störungsenergie auf die Erde ableiten zu können.

Verbinden Sie dazu die GND-Klemme / Buchse über ein kurzes impedanzarmes Masseband mit dem zentralen Erdungspunkt Ihrer Anlage. Die GND-Klemme / Buchse ist im Gerät mit der Masse und dem Gehäuse verbunden.

Galvanische Kopplung

Beim Ethernet-Kabel sind die Datenleitungen galvanisch entkoppelt, die Massepotenziale sind jedoch gekoppelt, weil die Schirmung des Ethernet-Steckers (RJ-45) mit GND verbunden ist.

Ausgleichströme ausschließen

Ausgleichsströme, die über das Gehäuse oder die Schirmung abfließen, beeinflussen das Messsignal.
Wenn Sie Ausgleichströme vermindern wollen, müssen Sie darauf achten, dass die Wirkung des Schirmes erhalten bleibt, indem Sie geeignete Maßnahmen zur Ableitung von Störungen treffen, wie z.B. das Auflegen des Schirms kurz vor dem Eintritt in den Schaltschrank. Je häufiger Sie die Schirmung auf dem Weg zur Maschine erden, desto besser ist die Schirmwirkung.

Verwenden Sie für die **Signalleitungen** möglichst Kabel mit beidseitig aufgelegtem Schirm. Auch hier sollte das Ableiten von Störungen über das Gehäuse mit der Verwendung von Schirmklemmen reduziert werden.

Netzspannung

Betreiben Sie das Gerät nur mit der passenden Netzspannung. Für den Betrieb mit einem externem Netzteil gelten die Angaben des Herstellers. Betreiben Sie das Gerät nur im geschlossenen Zustand, schließen Sie Lücken zwischen den eingebauten Modulen mit Abdeckplatten.

Umgebungsklima

ADwin-Pro II ist für den Betrieb in trockenen Räumen konzipiert und muss daher vor Feuchtigkeit und Kondenswasser geschützt werden. Das Gehäuse kann in Schaltschränken eingebaut oder mobil betrieben werden (z.B. im Kfz). Am Einbauort sollen eine Umgebungstemperatur von +5 °C … +50 °C und eine relative Luftfeuchte von 0 … 80 % (nicht kondensierend) vorhanden sein.

Gehäusetemperatur

Die Gehäusetemperatur (Oberflächentemperatur) darf auch unter extremen betrieblichen Bedingungen, z.B. im Schaltschrank oder bei direkter Sonneneinstrahlung, +60 °C nicht überschreiten. Es besteht sonst die Gefahr, dass Schäden am Gerät entstehen oder nicht definierte Daten (Werte) ausgegeben werden, die unter ungünstigen Umständen zu Schäden in ihrer Anlage führen können.

Beachten Sie insbesondere beim Schaltschrank-Einbau:

– Das *ADwin-Pro II*-Gerät soll nicht über starken Wärmequellen stehen wie z.B. Leistungstransformatoren.

– Die Be- und Entlüftung im Schaltschrank bis zum und vom *ADwin-Pro II*-Gerät muss gewährleistet sein.
Insbesondere müssen die Lüftungsschlitze des Geräts frei bleiben, so dass die vom Gerät erzeugte Wärme vollständig abgeführt wird.

– Wenn das Gerät an der Vorderseite mit Befestigungswinkeln Pro II-Mount (siehe Kapitel 7.4) montiert wird, beispielsweise in einem 19“-Schrank, empfehlen wir, das Gehäuse sicherheitshalber zusätzlich auf der Rückseite auf einer Lastaufnahme aufzulagern.

4 Gehäuse für das *ADwin-Pro II*-System

Die Gehäusevarianten für das *ADwin-Pro II*-System unterscheiden sich durch die Anzahl der Steckplätze und die Art der Stromversorgung:

Gehäuse	Anzahl Steckplätze	Stromversorgung	
ADwin-Pro II	16	100V…240V	AC
ADwin-Pro II-DC	16	10V…35V	DC
ADwin-Pro II-BM	15	100V…240V	AC
ADwin-Pro II-light	7	100V…240V	AC
ADwin-Pro II-light-DC	7	10V…35V	DC
ADwin-Pro II-mini	5	10V…36V	DC

Die Anzahl der Steckplätze gilt für Pro II-Module. Wenn Pro I-Module verwendet werden – auch gemischt mit Pro II-Modulen – passen weniger Module in das Gehäuse.

Für die Abmessungen des Einschubbereichs (inklusive Netzteileinschub) gelten folgende Maßeinheiten:

1 TE = 1/5 inch = 5,08mm 1 HE = 1¾ inch = 44,45mm

Die Einsteckmodule haben meistens eine Breite von 5 TE = 1 inch.

Modul einstecken

So stecken Sie ein Modul in das Gehäuse ein:

– Schalten Sie das Gerät aus! Ein Modul kann beschädigt werden, wenn Sie es bei eingeschalteter Stromversorgung einstecken oder herausziehen.

– Entfernen Sie ein oder mehr Abdeckbleche an der gewünschten Position, so dass am linken Rand die Führungsschienen zu sehen sind: je eine oben und eine unten.

  - Achten Sie auf die Farbe der Schienen. Es gibt für Pro I- und Pro II-Module verschiedene, leicht versetzte Führungsschienen: Weiße Schienen: Pro II-Module. Schwarze Schienen: Pro I-Module.
  - Das Prozessormodul hat einen festen Steckplatz, andere Positionen sind nicht möglich.

– Führen Sie die Platine mit dem Stecker voran oben und unten sorgfältig in die Führungsschienen ein. Bei korrekter Positionierung lässt sich das Modul nicht schräg stellen.

– Schieben Sie das Modul ganz nach hinten. Am Ende spüren Sie einen leichten Widerstand, wenn Sie den Modulstecker in die Buchse der Rückwand einschieben.

  Das Deckblech des Moduls sollte ganz am Gehäuse anliegen.

– Drehen Sie die Schrauben oben und unten am Deckblech fest.

– Schließen Sie eventuelle Lücken zwischen den eingebauten Modulen mit den Abdeckplatten. Es gibt Abdeckplatten mit 2, 3 oder 5 TE Breite.

Gehäuse für das ADwin-Pro II-System

4.1 *ADwin-Pro II*

16 Steckplätze

Das Standard-Gehäuse für *ADwin-Pro II*-Systeme. Die Rückwand (Backplane) des Gehäuses verbindet das Prozessormodul mit den anderen Modulen über einen internen Bus.

Die Gerätesicherung befindet sich an in einem Einschub im Netzteil, oberhalb der Buchse für den Netzstecker (Gehäuserückseite).

Anzahl Steckplätze	16
Außenabmessungen (L × B × H) Einschubbereich (B × H)	336 mm × 447,5 mm × 146 mm (mit Füßen) 84 TE × 3 HE
Netzteil	100 V…240 VAC bei 50/60 Hz Schaltnetzteil Leistungsobergrenze Sekundärseite >70 W
Sicherung	5 A, träge

Abb. 1 – Gehäuse *ADwin-Pro II*: Spezifikation

Auf der Gehäuserückseite befindet sich über dem Netzstecker ein Aufkleber mit der Revisionsbezeichnung des Gehäuses:

Revision	Ausgabe	Änderung zur Vorgänger-Version
E1	Jun. 2005	*ADwin-Pro II*: Neue Gehäusekonstruktion und neue Backplane mit Pro I- und Pro II-Bus.

Gehäuse Pro II

Im Pro II-Gehäuse können sowohl Pro I-Module als auch Pro II-Module eingesteckt werden. Die Gehäuserückwand enthält sowohl den Pro I-Bus als auch den Pro II-Bus; das Prozessormodul arbeitet parallel mit beiden Bussen.

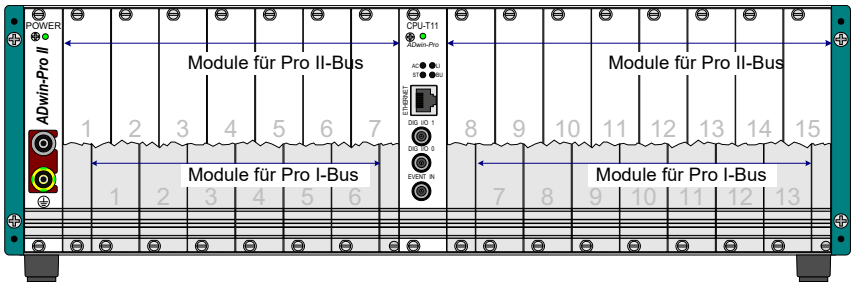


Abb. 2 – Gehäuse *ADwin-Pro II*

Bitte beachten Sie, dass Module für Pro I-Bus (im Bild grau) und Pro II-Bus leicht versetzte Einsteckpositionen haben. Die Positionen sind an der Farbe der Schienen leicht zu unterscheiden:

– Weiße Schienen: Module für Pro II-Bus.

– Schwarze Schienen: Module für Pro I-Bus.

  Ausgangsmodule Pro-AOut-x mit Rev. A können aus technischen Gründen nicht verwendet werden.

Das Prozessormodul wird immer in der Mittelposition eingesteckt (weiße Schienen).

Zwischen dem Prozessormodul und Modulen für Pro I-Bus bleibt ein halber Steckplatz frei (Abdeckbleche liegen bei), während Module für Pro II-Bus ohne Abstand neben das Prozessormodul passen.

Pro II-DC
16 Steckplätze

4.2 ADwin-Pro II-DC

Das Gehäuse *ADwin-Pro II-DC* entspricht vollständig dem Standardgehäuse *ADwin-Pro II*, ist aber mit einem Gleichstromnetzteil ausgerüstet.

Wenn zur Spannungsversorgung ein strombegrenzendes Netzteil verwendet wird, sollte dies in der Lage sein, beim Einschalten ein Mehrfaches des Ruhestroms zur Verfügung zu stellen, um einen einwandfreien Betrieb zu gewährleisten.

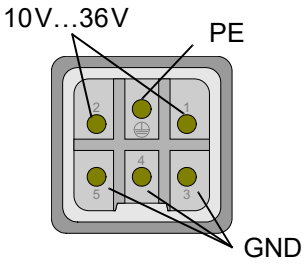


Abb. 3 – Gehäuse *ADwin-Pro II-DC*:
Detailansicht der Steckerbelegung

Anzahl Steckplätze	16
Außenabmessungen (L × B × H) Einschubbereich (B × H)	336mm × 447,5mm × 146mm (mit Füßen) 84 TE × 3 HE
Netzteil	DC-DC-Wandler 10V…35V Leistungsobergrenze Sekundärseite >80W
Sicherung	25A, 58V

Abb. 4 – Gehäuse *ADwin-Pro-DC*: Spezifikation

Auf der Gehäuserückseite befindet sich über dem Netzstecker ein Aufkleber mit der Revisionsbezeichnung des Gehäuses:

Revision	Ausgabe	Änderung zur Vorgänger-Version
E1	Jun. 2005	*ADwin-Pro II*: Neue Gehäusekonstruktion und neue Backplane mit Pro I- und Pro II-Bus. Neuer Stromversorgungsstecker.

Gehäuse für das ADwin-Pro II-System

4.3 *ADwin-Pro II-BM*

Pro II „backmounted“ mit 15 Steckplätzen

Das Gehäuse entspricht vollständig dem Standardgehäuse, aber die Module werden auf der Rückseite eingesteckt (BM = back mounted).

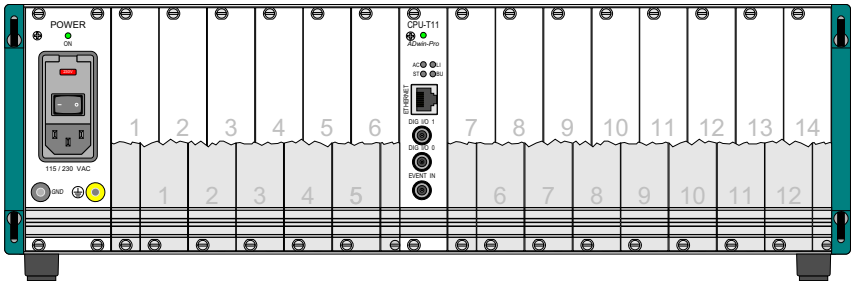


Abb. 5 – Gehäuse *ADwin-Pro II-BM* (Rückseite)

Die Gerätesicherung befindet sich in einem Einschub im Netzteil, oberhalb der Buchse für den Netzstecker.

Anzahl Steckplätze	16
Außenabmessungen (L × B × H) Einschubbereich (B × H)	336 mm × 447,5 mm × 146 mm (mit Füßen) 84 TE × 3 HE
Netzteil	100…240 VAC bei 50/60 Hz Schaltnetzteil Leistungsobergrenze Sekundärseite >70 W
Sicherung	5 A, träge

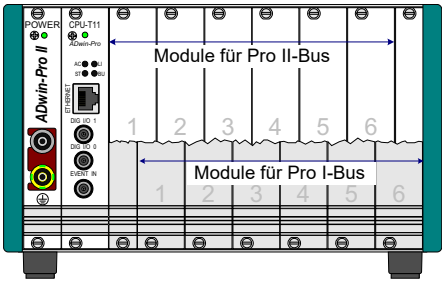
Abb. 6 – Gehäuse *ADwin-Pro II-BM*: Spezifikation

Auf der Gehäuserückseite befindet sich über dem Netzstecker ein Aufkleber mit der Revisionsbezeichnung des Gehäuses:

Revision	Ausgabe	Änderung zur Vorgänger-Version
E1	Jun. 2005	*ADwin-Pro II*: Neue Gehäusekonstruktion und neue Backplane mit Pro I- und Pro II-Bus.

Pro II-light mit 7 Steckplätzen

4.4 *ADwin-Pro II-light*



Gehäuse *ADwin-Pro II*-light

Die Rückwand (Backplane) des Gehäuses verbindet das Prozessormodul mit den *Pro*-Modulen.

Anzahl Steckplätze	7
Außenabmessungen (L × B × H) Einschubbereich (B × H)	336 mm × 234 mm × 146 mm (mit Füßen) 42 TE × 3 HE
Netzteil	100…240 VAC bei 50/60 Hz Schaltnetzteil Leistungsobergrenze Sekundärseite >40 W
Sicherung	2 A, träge

Abb. 7 – Gehäuse *ADwin-Pro II-light*: Spezifikation

Auf der Gehäuserückseite befindet sich über dem Netzstecker ein Aufkleber mit der Revisionsbezeichnung des Gehäuses:

Revision	Ausgabe	Änderung zur Vorgänger-Version
E1	Jun. 2005	*ADwin-Pro II*: Neue Gehäusekonstruktion, schmaler Netzteileinschub und neue Backplane mit Pro I- und Pro II-Bus.

Gehäuse für das ADwin-Pro II-System

4.5 ADwin-Pro II-light-DC

Pro II-light-DC mit 7 Steckplätzen

Das Gehäuse *ADwin-Pro-light-DC* entspricht vollständig dem Gehäuse *ADwin-Pro II-light*, ist aber mit einem Gleichstromnetzteil ausgerüstet.

Wenn zur Spannungsversorgung ein strombegrenzendes Netzteil verwendet wird, sollte es in der Lage sein, beim Einschalten ein Mehrfaches des Ruhestroms zur Verfügung zu stellen, um einen einwandfreien Betrieb zu gewährleisten.

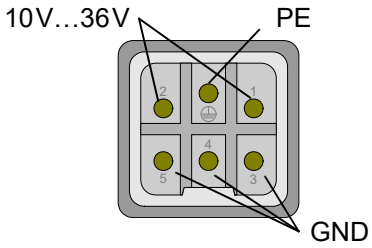


Abb. 8 – Gehäuse *ADwin-Pro II-light-DC*: Anschlussbuchse für die Stromversorgung

Anzahl Steckplätze	7
Außenabmessungen (L × B × H) Einschubbereich (B × H)	336mm × 234mm × 146mm (mit Füßen) 42 TE × 3 HE
Netzteil	DC-DC-Wandler 10V…35V Leistungsobergrenze Sekundärseite >80W
Sicherung	15A, 58V

Abb. 9 – Gehäuse *ADwin-Pro II-light-DC*: Spezifikation

Auf der Gehäuserückseite befindet sich über dem Netzstecker ein Aufkleber mit der Revisionsbezeichnung des Gehäuses:

Revision	Ausgabe	Änderung zur Vorgänger-Version
E1	Okt. 2006	Erst-Version, nur für Pro II.

Das Gehäuse wird mit einer Anschlussbuchse für die Stromversorgung ausgeliefert. Die Buchse kann bei Bedarf wie folgt nachbestellt werden:

Bezugsadresse: Regional-Electronic-Distribution Handelsgesellschaft mbH
Postfach 1250, 63084 Rodgau

Bezeichnungen: Steckergehäuse Harting, Serie HA.3.XX.X,
Best.nr. HA.3.STO.1.11

Buchseneinsatz Harting, 5+PE, 400V, 16A,
Best.nr. HE.Q.5.BU.C

4 Stk. Buchsenkontaktstift 2.5mm^2, vergoldet,
Best.nr. HE-HA.C.BU.2,5.AU

Pro II-mini mit 5 Steckplätzen

4.6 *ADwin-Pro II-mini*

Das kleinste Gehäuse für *ADwin-Pro II*-Systeme hat 5 Steckplätze und benötigt ein externes Netzteil. Der Anschluss für das Netzteil ist auf der Rückseite des Gehäuses.

Anzahl Steckplätze	5
Außenabmessungen (L × B × H) Einschubbereich (B × H)	253 mm × 147,3 mm × 146 mm (mit Füßen) 20 TE x 3 HE
Sicherung	10 A, 58 V
Externes Netzteil	externes Netzteil erforderlich DC-DC-Wandler 10 V…36 V DC Leistungsobergrenze Sekundärseite 50 W

Abb. 10 – Gehäuse *ADwin-Pro II-mini*: Spezifikation

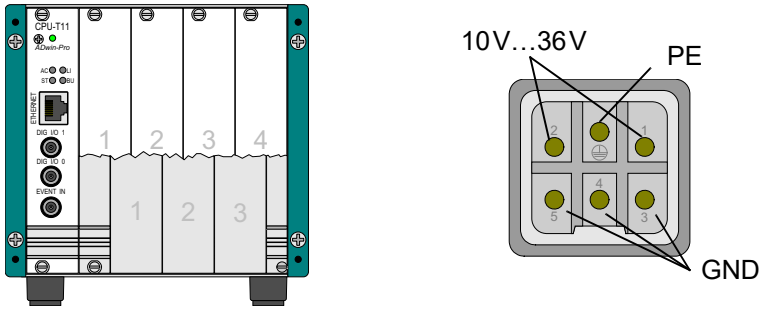


Abb. 11 – Gehäuse *ADwin-Pro II-mini* und Anschlussbuchse für die Stromversorgung

Auf der Gehäuserückseite befindet sich ein Aufkleber mit der Revisionsbezeichnung des Gehäuses:

Revision	Ausgabe	Änderung zur Vorgänger-Version
E1	Dez. 2005	*ADwin-Pro II*: Neue Gehäusekonstruktion und neue Backplane mit Pro I- und Pro II-Bus. Variable Eingangsspannung.

5 *ADwin-Pro*-Module

Ein *ADwin-Pro*-Modul belegt in der Regel einen Steckplatz (5 TE) in einem *ADwin-Pro II*-System, manche Module auch 2 Steckplätze.

Alle technischen Daten der Module beziehen sich auf das eingeschaltete Gerät.

Beachten Sie beim Einstecken eines Moduls in das Gehäuse die Beschreibung auf Seite 4.

5.1 Nicht unterstützte Pro I-Module

Wenn das Prozessormodul Pro-CPU-T11-ENET verwendet wird, sind grundsätzlich alle Pro I-Module auch in einem *ADwin-Pro II*-System einsetzbar.

Dagegen werden die folgenden Module in *ADwin-Pro II* in keinem Fall unterstützt. Sie können nur in einem *ADwin-Pro I*-System eingesetzt werden:

– Pro-AOut-4/16 Rev. A

– Pro-AOut-8/16 Rev. A

– Pro-AO-16/8-12 Rev. A

Das Prozessormodul Pro-CPU-T12 arbeitet nur mit Pro II-Modulen.

5.2 Moduladresse einstellen (auch Pro I-Module)

Jedes eingesteckte Pro-Modul (ausgenommen CPU-Module) wird in einem *ADbasic*-Programm über seine Moduladresse angesprochen. Die Moduladresse ist weitgehend frei wählbar.

Moduladresse wählen

Für das Wählen der Moduladresse gelten folgende Regeln:

– Eine Moduladresse muss innerhalb der Modulgruppe eindeutig sein.

  Jedes Modul gehört zu einer Modulgruppe:
  - Pro II-Module.
  - Pro I-Module, Funktionsgruppe CPU: Prozessormodule.
  - Pro I-Module, Funktionsgruppe ADC: analoge Eingangsmodule.
  - Pro I-Module, Funktionsgruppe DAC: analoge Ausgangsmodule.
  - Pro I-Module, Funktionsgruppe DIO: digitale Ein- / Ausgangs-, Relais- und Zählermodule.
  - Pro I-Module, Funktionsgruppe EXT: Sondermodule aller Art.

– Die Moduladresse kann in folgenden Grenzen frei gewählt werden:
  - Pro II-Module: 1 … 15.
  - Pro I-Module: 1 … 255.

Es ist zwar möglich, Modulen aus verschiedenen Gruppen die gleiche Moduladresse zu geben. Um Verwechslungen zu vermeiden, empfehlen wir aber, eindeutige Adressen zu vergeben.

Moduladresse einstellen (auch Pro I-Module)

Moduladresse einstellen: Pro II-Module

Bei Pro II-Modulen stellen Sie die Moduladresse mit dem Programm *ADpro* ein. Das Einstellen setzt gleichzeitig alle Pro II-Module in den Anfangszustand zurück. Gehen Sie vor wie folgt:

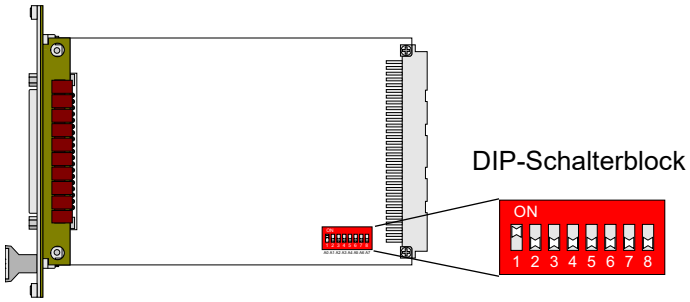
– Schalten Sie die Stromversorgung des *ADwin*-Systems aus und stekken die Pro II-Module in das Gehäuse ein; beachten Sie die Beschreibung auf Seite 4. Schalten Sie die Stromversorgung wieder ein.

– Anschließend starten Sie das Programm *ADpro* aus dem Windows Startmenü unter `Programs` ▸ `ADwin`. Unter dem Menüpunkt `Edit` ▸ `Set module adresses` können Sie die gewählten Moduladressen einstellen.

Moduladresse einstellen: Pro I-Module

Bei Pro I-Modulen stellen Sie die Moduladresse manuell an einem DIP-Schalterblock ein. Der Schalterblock befindet sich auf der Platine rechts unten.

Mit den 8 DIP-Schaltern ist eine Adresse zwischen 1 und 255 einstellbar (siehe Abb. 12). Wie oben beschrieben, muss jedes Modul einer Gruppe eine eindeutige Adresse haben.



Moduladresse	Einstellung der DIP-Schalter							
	1	2	3	4	5	6	7	8
1	1	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0
3	1	1	0	0	0	0	0	0
4	0	0	1	0	0	0	0	0
5	1	0	1	0	0	0	0	0
6	0	1	1	0	0	0	0	0
7	1	1	1	0	0	0	0	0
8	0	0	0	1	0	0	0	0
…				…				
254	0	1	1	1	1	1	1	1
255	1	1	1	1	1	1	1	1

Abb. 12 – Adressierung der *ADwin-Pro I*-Module mit den DIP-Schaltern

Beachten Sie bitte:

– Ein RSxxx-Modul mit 4 seriellen Schnittstellen belegt 2 Adressen (Gruppe EXT): die eingestellte Adresse und die nächsthöhere.

– Ein Feldbus-Modul belegt 32 Adressen (Gruppe EXT); die Adressverteilung ist wie folgt:

Moduladresse einstellen (auch Pro I-Module)

Eingestellte Moduladresse	Zusätzl. belegte Adressen	Einstellung der DIP-Schalter							
		1	2	3	4	5	6	7	8
1	160…191	1	0	0	0	0	0	0	0
2	192…223	0	1	0	0	0	0	0	0
3	224…255	1	1	0	0	0	0	0	0
4	128…159	0	0	1	0	0	0	0	0

5.3 Prozessormodule

Für jedes *ADwin-Pro II*-System wird ein Prozessormodul benötigt. Das Prozessormodul ist die zentrale Recheneinheit des Pro II-Systems und hat folgende Aufgaben:

- Kommunikation mit dem PC oder Laptop. Die Verbindung wird über Ethernet hergestellt.
- Kommunikation mit allen anderen Pro II-Modulen über den internen Bus.
- Kommunikation mit eventuell vorhandenen TiCo-Prozessoren auf Pro II-Modulen über den internen Bus.
- Aufnehmen und Ausführen der benutzerdefinierten Prozesse.
- Synchron-Taktgeber für zeitgesteuerte Abläufe in allen Modulen: Prozessortakt, Ablaufsteuerungen, Zähler, Zeitstempel

  Für besondere Anforderungen (nur als Bestelloption) kann ein externes Synchronsignal eingespeist werden. Damit können mehrere *ADwin-Pro II*-Systeme mit gleicher Zeittaktung und ohne gegenseitige Drift arbeiten.

Auf dem Prozessormodul ist der Speicher für Daten und Programme untergebracht. Derzeit stehen für *ADwin-Pro II* folgende Prozessormodule zur Verfügung:

- Pro-CPU-T11-ENET
- Pro-CPU-T12

Das Modul kann ergänzt werden durch die Bestelloptionen Pro II-Boot mit Speichermedium (zusätzliches externes Speichermedium) und Pro II-Boot (selbstständiger Start des Prozessormoduls).

5.3.1 Pro-CPU-T11-ENET

Das Prozessormodul kann nur in einem Pro II-Gehäuse eingesetzt werden und arbeitet sowohl mit Pro I- als auch mit Pro II-Modulen.

Beachten Sie, dass manche Pro I-Module nicht unterstützt werden, siehe Kapitel 5.1.

Es gibt für das Modul mehrere Bestelloptionen Pro II-Boot mit Speichermedium, die den Zugriff auf ein Speichermedium erlauben. Dadurch wird z. B. die Datenspeicherung im stand-alone-Betrieb das *ADwin*-Systems bei Langzeit-Messungen möglich.
Die zusätzlichen Eigenschaften der Optionen sind unter Pro II-Boot mit Speichermedium beschrieben.

Geeignet für Gehäusetyp	Pro II
Prozessor	ADSP TS101S
Taktrate Genauigkeit und Drift	300 MHz ±20 ppm
Rechenauflösung Float-Werte	40 Bit
Datenleitung	Ethernet: Schnittstelle ENET-2, bis 100 MbBit/s Schnittstelle ENET-3 (ab Rev. E10), bis 1000 MbBit/s
Interner Speicher	PM: 256 KiB, DM: 256 KiB, EM: 256 KiB
Externer Speicher	256 MiB
TTL-Signaleingänge	Event In, mit 4,7 kΩ Pull-down-Widerstand Dig I/O 0, mit 4,7 kΩ Pull-up-Widerstand Dig I/O 1, mit 4,7 kΩ Pull-up-Widerstand

Abb. 13 – Pro-CPU-T11-ENET: Spezifikation

Das Prozessormodul hat im Pro II-Gehäuse eine feste Position. Beachten Sie zum Einstecken des Moduls auch die Hinweise auf Seite 4.

Der interne Speicher des Prozessors ist aufgeteilt in Programmspeicher (PM), Datenspeicher (DM) und frei verwendbaren Zusatzspeicher (EM).

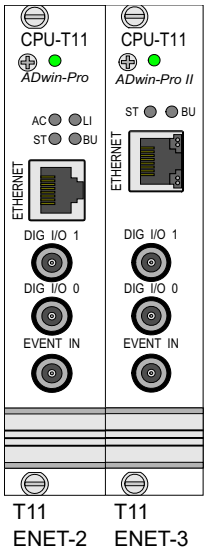


Abb. 14 – Pro-CPU-T11-ENET: Frontplatte

Das Prozessormodul signalisiert seinen Betriebszustand durch LEDs am Ethernet-Anschluss. Die Bedeutung der LEDs ist im Handbuch ADwin Installation, Kapitel 10.5 beschrieben.

Eingang Event In

Mit dem externen Trigger-Eingang (`Event In`) kann das Prozessormodul ein externes TTL-Signal (Trigger) als Event-Signal erkennen und einen Prozess auslösen, der sofort und vollständig abgearbeitet wird (siehe Handbuch *ADbasic*, Kapitel: Die Programmabschnitte).

Das Event-Signal muss 50 ns lang anstehen, um erkannt zu werden.

Alternativ kann ein Event-Eingang eines anderen Moduls im Pro II-Gehäuse verwendet werden. Alle Event-Signale gelangen beim Prozessormodul auf die gleiche Signalleitung wie der Eingang `Event In`.

Der Event-Eingang kann in *ADbasic* mit dem Befehl **`CPU_Event_Config`** konfiguriert werden.

Digitalkanäle DIG I/O

Die Digitalkanäle `DIG I/O 0` und `DIG I/O 1` arbeiten mit TTL-Signalen und können als Eingang oder Ausgang programmiert werden.

Nach einem Neustart sind die `DIG I/O`-Kanäle als Eingang für fallende Flanken konfiguriert.

Programmierung in ADbasic

Die Digitalkanäle des CPU-Moduls werden komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind in der Online-Hilfe *ADbasic* und im Handbuch Pro II-Software erläutert.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Digitalkanäle konfigurieren	**`CPU_Dig_IO_Config`**
Flanke am Digitaleingang abfragen	**`CPU_Digin`**
Pegel am Digitalausgang setzen	**`CPU_Digout`**
Event-Eingang konfigurieren	**`CPU_Event_Config`**

Software-Änderung beim Umstieg von T9 / T10

Bestehende *ADbasic*-Programme für den Prozessor T9 oder T10 können – soweit die Programme mit Pro I-Modulen arbeiten – mit wenigen, aber notwendigen Änderungen weiter verwendet werden:

– Für den Prozessor T11 muss die Include-Datei `<ADwinPro II_All.inc>` neu eingebunden werden. Gleichzeitig sollten alle anderen Include-Dateien für Pro II-Module aus dem Programm gelöscht werden.

PROCESSDELAY

– Die Zeiteinheit des **`Processdelay`** (Zykluszeit) beträgt $3,\overline{3}$ ns für hochpriore wie für niederpriore Prozesse.

  Alle entsprechenden Werte und Berechnungen müssen angepasst werden. Das größtmögliche **`Processdelay`** entspricht 7,1 s; für größere Zykluszeiten muss eine Hilfsvariable verwendet werden.

READ_TIMER

– Die Zeiteinheit von $3,\overline{3}$ ns gilt auch für den internen Zähler, d.h. Abfragen des Zählers mit **`Read_Timer`** müssen ebenfalls angepasst werden.

  Beachten Sie bitte: Das Zeitverhalten von Prozessen im Zusammenhang mit Ein-/Ausgabebefehlen ist komplexer geworden (s.u.). Mit **`Read_Timer`** bestimmte Zeitdifferenzen spiegeln jetzt nur noch einen Teilaspekt des gesamten Zeitverhaltens wider.

SLEEP

– Der Befehl **SLEEP** muss durch einen der folgenden neuen Befehle ersetzt werden:
  - **CPU_SLEEP** lässt den Prozessor warten. Der Befehl **SLEEP** hatte bei den Prozessoren T9 und T10 die gleiche Funktion.
  - **P1_SLEEP** lässt den Pro I-Bus warten, z. B. um Ein-/ Ausgabebefehle aufeinander abzustimmen. Der Befehl lässt auch Pro II-Bus und externen Speicher warten.
  - **P2_SLEEP** lässt den Pro II-Bus warten, z. B. um Ein-/ Ausgabebefehle aufeinander abzustimmen.

  Die neuen Befehle haben eine Zeiteinheit von 10 ns (bei **Sleep**: 100 ns).

  Welcher Befehl ist der richtige? In der Regel wird **Sleep** verwendet, um die Wartezeit eines Ein-/ Ausgabebefehls zu überbrücken, z. B. das Einschwingen des Multiplexers bei **Set_Mux**. In diesem Fall ist für bisherige Module (Pro I-Bus) **P1_Sleep** der passende Befehl, für Pro II-Module ist es **P2_Sleep**.

  Beachten Sie die Hinweise im *ADbasic*-Handbuch, wie Sie die Wartezeit genau einstellen (Kapitel 5.2.4).

  Warum gibt es neue Befehle? Der Prozessor T11 unterscheidet zwischen Prozessorbefehlen einerseits und Ein-/Ausgabebefehlen andererseits. Die Prozessorarchitektur erlaubt oft eine quasi-parallele Bearbeitung[1] der beiden Befehlsgruppen und damit eine deutlich schnellere Bearbeitung der *ADbasic*-Prozesse. Das heißt gleichzeitig, dass die Befehlsgruppen (weitgehend) zeitlich unabhängig bearbeitet werden. Um das Zeitverhalten durch Warten zu beeinflussen, sind deshalb den Gruppen zugeordnete Befehle notwendig. Die Unterscheidung nach Bussen ergibt sich, weil für das Warten bei den Ein-/Ausgabebefehlen der jeweilige Bus angehalten wird.

1. Die Prozessorstruktur unterscheidet sich in diesem Punkt von T9 und T10: Dort wurden beide Befehlsgruppen sequentiell bearbeitet. Ein Anhalten des Prozessors mit **SLEEP** beeinflusste deshalb auch folgende Ein-/Ausgabebefehle.

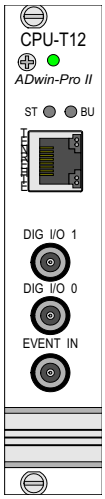
5.3.2 Pro-CPU-T12

Das Prozessormodul kann nur in einem Pro II-Gehäuse eingesetzt werden und arbeitet nur mit Pro II-Modulen.

Es gibt für das Modul mehrere Bestelloptionen Pro II-Boot mit Speichermedium (Details siehe Seite 20), die den direkten Zugriff auf ein externes Speichermedium erlauben. Dadurch wird z. B. die Datenspeicherung im standalone-Betrieb das *ADwin*-Systems bei Langzeit-Messungen möglich.

Geeignet für Gehäusetyp	Pro II
Prozessor	XILINX ZYNQ™ mit Dual-Core ARM Cortex-A9
Taktrate Genauigkeit und Drift	1000 MHz ±50 ppm
Rechenauflösung Float-Werte	64 Bit
Datenleitung	Schnittstelle ENET-3, bis 1000 MbBit/s
Speicher	1000 MiB, davon bis zu 6 MiB cacheable
TTL-Signaleingänge	Event In, mit 4,7 kΩ Pull-down-Widerstand Dig I/O 0, mit 4,7 kΩ Pull-up-Widerstand Dig I/O 1, mit 4,7 kΩ Pull-up-Widerstand

Abb. 15 – Pro-CPU-T12: Spezifikation



Das Prozessormodul hat im Pro II-Gehäuse eine feste Position. Beachten Sie zum Einstecken des Moduls auch die Hinweise auf Seite 4.

Das Prozessormodul hat eine Speicher mit 1000 MB. Intern verwendet die *ADwin*-CPU einen Cache-Speicher, in dem Programmteile besonders schnell bearbeitet werden.

Das Prozessormodul signalisiert seinen Betriebszustand und die Kommunikationsaktivität durch LEDs am Ethernet-Anschluss. Die Bedeutung der LEDs ist im Handbuch ADwin Installation, Kapitel 10.5 beschrieben.

Eingang Event In

Mit dem externen Trigger-Eingang (`Event In`) kann das Prozessormodul ein externes TTL-Signal (Trigger) als Event-Signal erkennen und einen Prozess auslösen, der sofort und vollständig abgearbeitet wird (siehe Handbuch *ADbasic*, Kapitel: Die Programmabschnitte).

Das Event-Signal muss 50 ns lang anstehen, um erkannt zu werden.

Alternativ kann ein Event-Eingang eines anderen Moduls im Pro II-Gehäuse verwendet werden. Alle Event-Signale gelangen beim Prozessormodul auf die gleiche Signalleitung wie der Eingang `Event In`.

Der Event-Eingang kann in *ADbasic* mit dem Befehl **`CPU_Event_Config`** konfiguriert werden.

Digitalkanäle DIG I/O

Die Digitalkanäle `DIG I/O 0` und `DIG I/O 1` arbeiten mit TTL-Signalen und können als Eingang oder Ausgang programmiert werden.

Nach einem Neustart sind die `DIG I/O`-Kanäle als Eingang für fallende Flanken konfiguriert.

Programmierung in ADbasic

Die Digitalkanäle des CPU-Moduls werden komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind in der Online-Hilfe *ADbasic* und im Handbuch Pro II-Software erläutert.

Die Include-Datei `ADwinPro_All.inc` enthält folgende Befehle:

Bereich	Befehle
Digitalkanäle konfigurieren	`CPU_Dig_IO_Config`
Flanke am Digitaleingang abfragen	`CPU_Digin`
Pegel am Digitalausgang setzen	`CPU_Digout`
Event-Eingang konfigurieren	`CPU_Event_Config`

C-Code in *ADbasic* einbinden

Sie können in den *ADbasic*-Code auch eigenen C-Code einbinden. Näheres ist im Handbuch „ADbasic und C-Code“ beschrieben.

Software-Änderung beim Umstieg von T9 / T10 / T11

Bestehende *ADbasic*-Programme für frühere Prozessoren können ohne Änderung weiter verwendet werden, soweit die Programme auf Pro II-Module zugreifen. Der Zugriff auf Pro I-Module ist nicht möglich.

Beachten Sie:

PROCESSDELAY

– Die Zeiteinheit des **`Processdelay`** (Zykluszeit) beträgt 1 ns für hochpriore wie für niederpriore Prozesse.

  Alle entsprechenden Werte und Berechnungen müssen angepasst werden. Das größtmögliche **`Processdelay`** entspricht 2,1 s; für größere Zykluszeiten muss eine Hilfsvariable verwendet werden.

READ_TIMER

– Die Zeiteinheit von 1 ns gilt auch für den internen Zähler, d. h. Abfragen des Zählers mit `Read_Timer` müssen ebenfalls angepasst werden.

  Beachten Sie bitte: Im Vergleich zu früheren Prozessoren ist das Zeitverhalten von Prozessen komplexer geworden. Mit `Read_Timer` bestimmte Zeitdifferenzen spiegeln nur einen Teilaspekt des gesamten Zeitverhaltens wider. Deswegen gibt es für Hardware-Zugriffe den alternativ den Befehl `Read_Timer_Sync`.

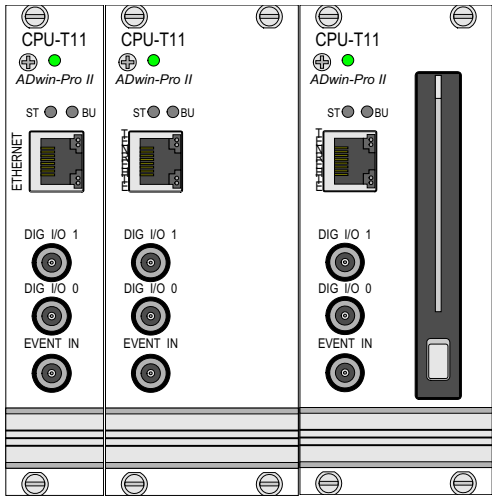
5.3.3 Pro II-Boot mit Speichermedium

Die Prozessormodule Pro-CPU-T12 und Pro-CPU-T11-ENET können optional für den Zugriff auf ein Speichermedium ausgerüstet werden. Die Option muss bestellt und kann nicht nachgerüstet werden. Der Zugriff auf das Speichermedium ermöglicht z. B. die Datenspeicherung im stand-alone-Betrieb das *ADwin*-Systems bei Langzeit-Messungen.

Bei jedem zusätzlichen Speichermedium ist der Bootloader Pro II-Boot für eigenständigen Betrieb mit enthalten, Beschreibung siehe Seite 23. Im folgenden ist nur der Umgang mit dem zusätzlichen Speichermedium beschrieben.

Das Prozessormodul wird je nach Bestellung mit einem der folgenden Speichermedien geliefert:

Option	Medium	min. Größe	Breite	
Pro II-Boot-USB	USB-Flash	16 GB	5 TE	
Pro II-Boot-SSD	SSD	240 GB	10 TE	fest eingebaut
Pro II-Boot-SSD-RMV	SSD	240 GB	10 TE	herausnehmbar auf Wechselrahmen
Pro II-Boot-HDD	Festplatte	1 TB	10 TE	fest eingebaut
Pro II-Boot-HDD-RMV	Festplatte	1 TB	10 TE	herausnehmbar auf Wechselrahmen



Speichermedium entnehmen / in den Wechselrahmen einlegen

In den Wechselrahmen passen Speichermedien mit dem Formfaktor 2,5“, max. Bauhöhe 9,5 mm.

So entnehmen Sie ein Speichermedium aus dem Wechselrahmen oder setzen es ein:

– Bei einem Speichermedium mit offener Platine kleben Sie zunächst die mitgelieferte Folie auf die Platinenseite. Ohne die Folie kann ein Kontakt der Platine mit dem Gehäuse entstehen und dadurch Fehler erzeugen.

  Eine Anleitung zur Befestigung der Folie bzw. des Abstandshalters (siehe unten) liegt bei.

  Bei einem schmalen Speichermedium nehmen Sie den mitgelieferten Abstandshalter und befestigen ihn an der linken Seite des Speichermediums.

Manche Speichermedien besitzen Jumper neben der Steckerleiste. In diesem Fall nehmen Sie die beigelegte Schutzkappe und schieben die Kappe über die Jumper. Die Schutzkappe verhindert einen Kontakt zwischen der Einschiebemechanik und den Jumpern.

– Öffnen Sie die Klappe am Modul, indem Sie den „Knopf“ etwas nach oben schieben. Ziehen Sie die Klappe nach vorn, bis sie nach oben zeigt (etwa waagerecht).

  Ein bereits eingestecktes Speichermedium wird dadurch nach geschoben und kann entnommen werden.

– Halten Sie das Speichermedium so, dass die Steckerleiste des Speichermediums in Richtung zum Modul und nach rechts zeigt.

– Schieben Sie das Speichermedium in den Schacht ganz ein.

– Drücken Sie die Modulklappe vorsichtig nach unten, bis die Klappe einrastet.

  Achtung: Wenn Sie die Klappe nicht oder nur mit Kraftaufwand schließen können, ist das Speichermedium nicht richtig eingelegt.
  Entnehmen Sie das Speichermedium und prüfen Sie die Position der Steckerleiste (siehe oben). Wiederholen Sie den Einschiebevorgang.

Dateiformate

Das Modul kann Speichermedien mit den Dateiformaten EXT3 oder FAT32 formatieren, lesen und schreiben. Im Dateiformat EXT3 wird Groß-/Kleinschreibung unterschieden, im Dateisystem FAT32 nicht.

Beachten Sie, dass *ADbasic* nur Dateien mit einer Größe von maximal 2 GiB lesen kann. Im Dateisystem EXT3 ist es möglich, aus *ADbasic* durch Anhängen von Daten eine Datei mit mehr als 2 GiB zu erzeugen (lesbar auf einem PC, aber nicht in ADbasic). Beim Dateisystem FAT32 erzeugt das Anhängen an eine Datei einen Fehler, wenn die maximale Dateigröße überschritten wird.

Ein auswechselbares Speichermedium kann auch an anderen Rechnern mit geeigneter SATA-Schnittstelle gelesen und beschrieben werden.

Modulinterne Uhr

Prozessormodule mit der Option Pro II-Boot-Storage enthalten eine modulinterne Uhr, die mit einer austauschbaren Batterie gepuffert ist. Das Modul nutzt die Uhrzeit, um Dateioperationen mit einem Zeitstempel zu versehen. Sie können die Uhrzeit mit einem *ADbasic*-Befehl abfragen. Als Zeitstempel für Messergebnisse ist die Uhr ungeeignet.

Die Uhrzeit wird ab Werk gestellt, danach synchronisiert sich das Modul automatisch mit einem Zeitserver im Internet, sobald es Internet-Zugang hat. Manuell kann die Uhrzeit derzeit nicht gesetzt werden.

Programmierung in ADbasic

Der Zugriff auf die Speichermedien wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind in der Online-Hilfe *ADbasic* und im Handbuch *ADwin*-Bootloader erläutert.

Die Include-Datei `ADwinPro_All.inc` enthält folgende Befehle:

Bereich	Befehle
Speichermedium erstmalig einrichten	`Storage_Create_Partitions_xxx` `Storage_Format_xxx`
Dateisystem für den Betrieb anmelden	`Storage_Mount_xxx` `Storage_Unmount_xxx`

Dateioperationen	`Storage_File_Create_xxx`
	`Storage_File_Delete_xxx`
	`Storage_File_Exists_xxx`
	`Storage_File_Status_xxx`
	`Storage_File_Move_xxx`

Datei lesen und schreiben	`Storage_File_Write_xxx`
	`Storage_File_Read_xxx`
	`Storage_File_Append_xxx`

Verzeichnisoperationen	`Storage_Dir_Create_xxx`
	`Storage_Dir_Delete_xxx`
	`Storage_Dir_Exists_xxx`
	`Storage_Dir_Status_xxx`

5.3.4 Pro II-Boot

Mit Pro II-Boot, auch Bootloader genannt, steht Ihnen eine Erweiterung zur Verfügung, mit der nach dem Einschalten

– das *ADwin-Pro II*-System gebootet wird.
– bis zu 10 Prozesse geladen werden können.
– Prozess 10 automatisch gestartet wird (falls vorhanden).
– Daten gespeichert werden können.

Pro II-Boot ist eine Bestelloption und kann nicht nachgerüstet werden. Der Bootloader ist bei der Bestelloption Pro II-Boot mit Speichermedium enthalten.

Der Bootloader wird mit dem Programm `ADethflash` (im Windows Startmenü unter `Programs\ADwin`) programmiert. Hinweise zur Bedienung sind im Programm enthalten.

Software

Mit der Installation von *ADbasic* und der *ADwin*-Treiber aus dem *ADwin*-Software-Paket sind bereits die für die Bootloader-Option nötigen Dateien/Programme auf den PC kopiert worden.

Wenn Sie den Bootloader benutzen, darf eine Anwendung, die Sie z.B. zur Visualisierung der Messdaten geschrieben haben, das *ADwin*-System nicht neu booten.

5.3.5 T11: Modul-Überwachung mit Watchdog

Sie können Ihr Prozessormodul T11 mit einem Watchdog überwachen. Der Watch-dog ist ein Zähler, der regelmäßig über den Programmcode zurückgesetzt werden muss. Geschieht das nicht, erzeugt der Watchdog ein Reset-Signal (siehe auch *ADwin-Pro I* Systembeschreibung „Programmierung in *ADbasic*“).

Das Reset-Signal des Watchdogs setzt die digitalen und analogen Ausgänge an Pro-Modulen auf diejenigen Werte, die der Konfiguration nach dem Einschalten entsprechen, im Normalfall digital 0 bzw. 0 Volt. Beachten Sie, dass ProII-Module vom Watchdog-Signal nur dann angesprochen werden, wenn der Prozessor T11 mindestens die Revision E06 hat.

Programmierung in ADbasic

Der Watchdog wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind in der Online-Hilfe *ADbasic* und im Handbuch Pro I-Software erläutert.

Die Include-Datei `ADwinPro_All.inc` enthält folgende Watchdog-Befehle:

Bereich	Befehle
Watchdog starten und stoppen	`StartWatchdog`, `StopWatchdog`
Watchdog-Zähler zurücksetzen	`ResetWatchdogTimer`

Hinweise im Zusammenhang mit Pro II-Flash-Boot:

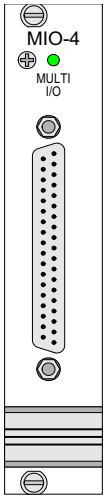
– Bitte achten Sie darauf, dass der Watchdog spätestens alle 1,6 s zurückgesetzt werden muss. Ein längerer Zeitraum zwischen zwei Impulsen wird ansonsten als Fehler interpretiert.
– Der Watchdog kann auch mit dem Bootloader Pro II-Flash-Boot verwendet werden, sorgt dann aber nicht für das automatische Laden und Starten der Software.
– Testen Sie Ihre Programme immer mit ausgeschaltetem Watchdog. Aktivieren Sie den Watchdog erst, wenn Ihre Programme zuverlässig arbeiten!

5.4 Pro II: Multi-IO-Module

Modulname	Rev.	Typ	Kanäle	Kenndaten	Seite
MIO-4	E	TTL-Ein-/Ausgang	8	U_{Ein} 5V TTL	25
		Analogeingänge	16 s.e. 8 diff.	1 ADC, 18 Bit, ±10V Wandlungszeit max. 2µs Abtastrate max. 500 kSample/s	
		Analogausgänge	4	4 DAC, 16 Bit, ±10V Einschwingzeit 9µs	
		TiCo1-Prozessor	–	128 KiByte interner Speicher 4 MiByte externes SRAM	
MIO-4-ET1	E	TTL-Ein-/Ausgang	8	U_{Ein} 5V TTL	32
		Analogeingänge	16 s.e. 8 diff.	1 ADC, 18 Bit, ±10V Wandlungszeit max. 2µs Abtastrate max. 500 kSample/s	
		Analogausgänge	4	4 DAC, 16 Bit, ±10V Einschwingzeit 9µs	
		Transistor-Ausgänge	4	5…30 V_{DC} (extern), 200mA	
		Optokoppler-Eingänge	4 s.e.	einstellbar 5V, 12V, 24V	
		Zählerblock	1	Universal, 32 Bit, 5V diff.	
		SSI-Decoder	1	max. 12,5MHz	
		EtherCAT-Schnittstelle	1	Slave-Schnittstelle	
		TiCo1-Prozessor	–	128 KiByte interner Speicher 4 MiByte externes SRAM	
MIO-D12	E	Transistor-Ausgänge	12	5…30 V_{DC} (extern), 200mA	45
		Optokoppler-Eingänge	12 s.e.	einstellbar 5V, 12V, 24V	
		Zählerblock	2	Universal, 32 Bit, 5V 1 diff., 1 über Optokoppler	
		SSI-Decoder	1	max. 12,5MHz	
		TiCo1-Prozessor	–	56 KiByte interner Speicher	

5.4.1 Pro II-MIO-4 Rev. E

Das Modul Pro II-MIO-4 Rev. E ist mit folgender Hardware ausgerüstet:

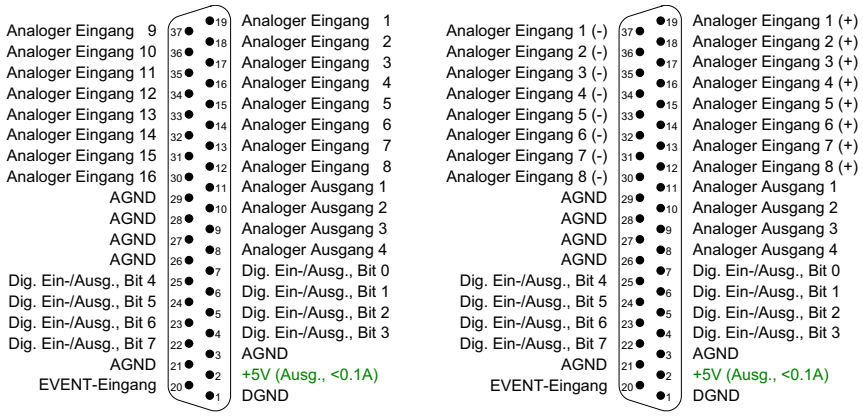


- 16 analoge Eingänge (16 single-ended oder 8 differentiell), über Multiplexer mit 18 Bit ADC, Abtastrate bis 500 kHz
- 4 analoge Ausgänge mit 16 Bit DAC.
- 8 digitale Ein- und Ausgangskanäle mit TTL-Pegeln
- 1 Event-Eingang
- *TiCo*-Prozessor (TiCo1) mit 128 KiByte internem Speicher und 4 MiB externem SRAM-Speicher

  Der *TiCo*-Prozessor hat Zugriff auf alle Ein- und Ausgänge des Moduls. Nähere Hinweise zur Anwendung und Programmierung des *TiCo*-Prozessors entnehmen Sie bitte dem Handbuch *TiCoBasic*.

Wenn der *TiCo*-Bootloader programmiert ist, kann das Modul eigenständig und unabhängig vom CPU-Modul des *ADwin-Pro II*-Systems arbeiten.

Es gibt eine erweiterte Modulvariante Pro II-MIO-4-ET1 Rev. E (Seite 32).



Analogeingänge single-ended Analogeingänge differentiell

Abb. 16 – Pro II-MIO-4 Rev. E: Pinbelegung

Die Funktionen des Moduls sind in folgenden Abschnitten beschrieben:

- Analoge Eingänge
- Analoge Ausgänge
- Digitale Ein-/Ausgänge
- TiCo-Prozessor
- Technische Daten
- Programmierung mit *ADbasic* und *TiCoBasic*

Analoge Eingänge

Das Modul Pro II-MIO-4 Rev. E hat 16 single ended-Eingänge oder 8 differentielle Eingänge (über Software einstellbar). Nach dem Einschalten ist das Modul auf 8 differentielle Eingänge eingestellt.

Die Eingänge sind auf eine 37-polige D-Sub-Buchse geführt; Pinbelegung siehe Abb. 16.

Die Eingänge sind über einen Multiplexer mit dem ADC verbunden. Der ADC hat eine Auflösung von 18 Bit und kann mit einer Abtastrate von bis zu 500 kSamples/s arbeiten.

Das Modul besitzt einen Eingangs-Spannungsbereich von ±10V und eine programmierbare Verstärkung von 1, 2, 4 oder 8. Der Abgleich der Verstärkung und des Offsets erfolgt per Software (siehe Kapitel 6 “Kalibrierung“).

Das Modul beinhaltet eine Ablaufsteuerung, die Messwerte an mehreren oder allen Eingangskanälen nacheinander einlesen kann.

Das Modul kann jeden Eingangskanal daraufhin überwachen, ob ein oberer oder unterer Grenzwert – die Grenzwerte sind für jeden Eingangskanal separat einstellbar – über- oder unterschritten wurde.

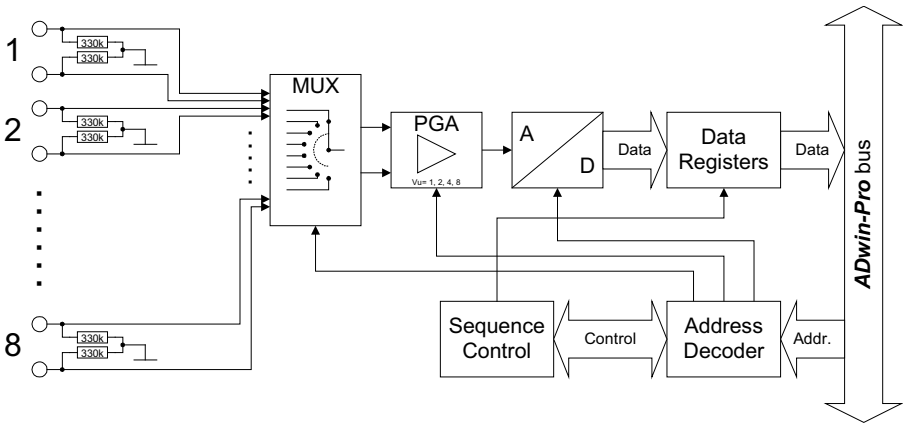


Abb. 17 – Pro II-MIO-4 Rev. E: Blockschaltbild der Analogeingänge

Analoge Ausgänge

Das Modul Pro II-MIO-4 Rev. E hat 4 analoge Ausgänge mit einem DAC zu 16 Bit. Jeder Ausgang hat einen festen Tiefpass 1. Ordnung, um Störungen zu unterdrücken (f_g = 10 MHz).

Der Ausgangs-Spannungsbereich der DAC ist fest auf ±10 V bipolar eingestellt und lässt sich nicht verändern. Der Abgleich des Offset erfolgt per Software (siehe Kapitel 6 “Kalibrierung“).

Die Ausgänge sind auf die 37-polige D-Sub-Buchse geführt; Pinbelegung siehe Abb. 16.

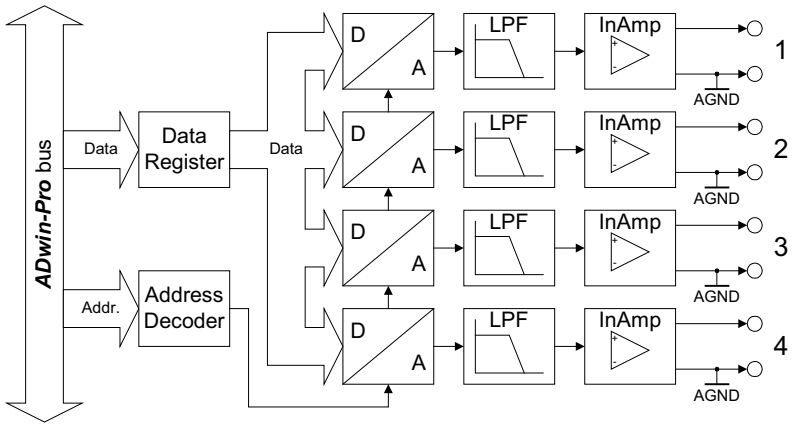


Abb. 18 – Pro II-MIO-4 Rev. E: Blockschaltbild der Analogausgänge

Digitale Ein-/Ausgänge

Das Modul Pro II-MIO-4 Rev. E stellt 8 programmierbare Ein- und Ausgangskanäle mit TTL-Pegeln bereit. Die Kanäle können in Blöcken zu jeweils 4 mit

ADbasic-Befehlen als Ein- oder Ausgänge konfiguriert werden. Nach dem Einschalten sind alle Kanäle als Eingänge konfiguriert.

Über den Trigger-Eingang EVENT kann ein Signal (Trigger) einen Prozess auslösen, der dann sofort und vollständig abgearbeitet wird (siehe *ADbasic*-Handbuch).

Die Digitalkanäle sind auf die 37-polige D-Sub-Buchse geführt; Pinbelegung siehe Abb. 16.

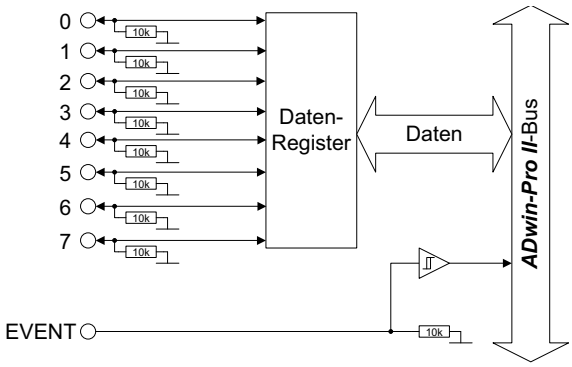


Abb. 19 – Pro II-MIO-4 Rev. E: Blockschaltbild

TiCo-Prozessor

Das Modul besitzt einen frei programmierbaren *TiCo*-Prozessor (TiCo1) mit 64 KiByte internem Programmspeicher, 64 KiByte internem Datenspeicher und 4 MiByte externem SRAM-Speicher. Sie programmieren den *TiCo*-Prozessor in *TiCoBasic*.

Der *TiCo*-Prozessor hat – wie auch die *ADwin*-CPU – Zugriff auf alle Ein- und Ausgangskanäle, analog wie digital. Nähere Hinweise zur Anwendung und Programmierung des *TiCo*-Prozessors entnehmen Sie bitte dem Handbuch *TiCoBasic*.

Wenn Sie ein *TiCoBasic*-Programm im *TiCo*-Bootloader ablegen, wird das Programm beim Einschalten der Stromversorgung in den *TiCo*-Prozessor geladen und gestartet. Auf diese Weise kann das Modul eigenständig und unabhängig vom CPU-Modul des *ADwin-Pro II*-Systems arbeiten.

Technische Daten

Analoge Eingänge	
Eingangskanäle	16 single ended / 8 differentiell, Multiplexer
Auflösung	18 Bit
Wandlungszeit	max. 2 µs
Abtastrate	max. 500 ksps
Multiplexer Einschwing-zeit	5 µs (innerhalb des Messbereichs)
Messbereich	±10 V
Verstärkung	1, 2, 4, 8 per Software einstellbar
Genauigkeit INL	typisch ±4 LSB
Genauigkeit DNL	max. ±1 LSB
Eingangswiderstand	330 kΩ, ±2%
Spannungsfestigkeit	±20 V
Offsetfehler	abgleichbar
Offsetdrift	±30 ppm/°C
Analoge Ausgänge	
Ausgangskanäle	4
Auflösung	16 Bit
Einschwingzeit	9 µs (auf 0,01% FSR)
Ausgangsspannung	±10 V
Max. Ausgangsstrom	±5 mA pro Kanal für optimale Funktion
Genauigkeit INL	±2 LSB typisch
Genauigkeit DNL	±1 LSB typisch
Offsetfehler	abgleichbar
Verstärkungsfehler	abgleichbar
Digitale Ein-/Ausgänge	
Digitale Eingänge	8 Kanäle mit TTL-Logik, konfigurierbar in Gruppen zu 4 Kanälen
Pull-Down-Widerstand	10 kΩ
V_{IH}	min. 2 V
V_{IL}	max. 0,8 V
I_{IH}	max. 1 µA
I_{IL}	max. 0,01 mA
Spannungsbereich	-0,5 V … +5,5 V
Ausgangsstrom	max. ±24 mA pro Kanal, max. ±50 mA je Block (4 Kanäle) über V_{CC} oder GND
Event-Eingang	TTL-Logik
Power-Up-Status	Alle digitalen Kanäle als Eingänge
Allgemein	
TiCo-Prozessor	Prozessortyp: TiCo1 Taktfrequenz: 50 MHz Speichergröße: 64 kiB PM intern, 64 kiB DM intern, 4 MiB externes SRAM
Steckerverbindung	37-polige D-Sub-Buchse

Abb. 20 – Pro II-MIO-4 Rev. E: Spezifikation

Programmierung

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Analoge Eingänge	
Eingänge einstellen auf single-ended oder differentiell	`P2_SE_Diff`

Einzelmessung durchführen – vollständig oder schrittweise	`P2_ADC`, `P2_ADC24`
	`P2_Set_Mux`, `P2_Start_Conv`
	`P2_Wait_EOC`
	`P2_Read_ADC`, `P2_Read_ADC24`

Messwert lesen und neue Messung starten	`P2_Read_ADC_SConv` `P2_Read_ADC_SConv24`

Ablaufsteuerung anwenden	`P2_Seq_Init`, `P2_Seq_Start`
	`P2_Seq_Read`, `P2_Seq_Read24`
	`P2_SEQ_Read_Packed`
	`P2_Seq_Wait`

Grenzwerte überwachen	`P2_ADC_Read_Limit` `P2_ADC_Set_Limit`
Analoge Ausgänge	
Ausgabe durchführen	`P2_DAC`, `P2_DAC4`, `P2_DAC4_Packed`

Ausgabe schrittweise durchführen	`P2_Write_DAC`, `P2_Write_DAC4`
	`P2_Write_DAC4_Packed`
	`P2_Write_DAC32`
	`P2_Start_DAC`

Digitale Ein-/Ausgänge	
Ein- und Ausgänge konfigurieren	`P2_MIO_DigProg`
Eingangssignale abfragen	`P2_MIO_Digin_Long`

Latch-Register nutzen	`P2_MIO_Dig_Latch`
	`P2_MIO_Dig_Read_Latch`
	`P2_MIO_Dig_Write_Latch`

Ausgangssignale setzen und rücklesen	`P2_MIO_Digout`
	`P2_MIO_Digout_Long`
	`P2_MIO_Get_Digout_Long`

Allgemein	
LEDs einstellen	`P2_Check_LED`, `P2_Set_LED`
Abläufe synchronisieren	`P2_Sync_All`, `P2_Sync_Enable` `P2_Sync_Stat`
Interrupts und Event-Eingang einstellen	`P2_Event_Enable`, `P2_Event_Read` `P2_Event_Config`

Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe erläutert.

Programmierung in ADbasic

Programmierung in TiCoBasic

Das Modul kann mit *TiCoBasic*-Befehlen programmiert werden. Die Befehle sind in der Online-Hilfe *TiCoBasic* erläutert.

Die Include-Datei `MIO_TiCo.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Analoge Eingänge	
Eingänge einstellen auf single-ended oder differentiell	`SE_Diff`

Einzelmessung durchführen – vollständig oder schrittweise	`ADC`, `ADC24`
	`Set_Mux`, `Start_Conv`, `Wait_EOC`
	`Read_ADC`, `Read_ADC24`

Messwert lesen und neue Messung starten	`Read_ADC_SConv` `Read_ADC_SConv24`

Ablaufsteuerung anwenden	`Seq_Init`, `Seq_Start`
	`Seq_Read`, `Seq_Read24`
	`Seq_Wait`

Grenzwerte überwachen	`ADC_Read_Limit` `ADC_Set_Limit`
Analoge Ausgänge	
Ausgabe durchführen	`DAC`
Ausgabe schrittweise durchführen	`Write_DAC`, `Write_DAC32` `Start_DAC`
Digitale Ein-/Ausgänge	
Ein- und Ausgänge konfigurieren	`MIO_DigProg`
Eingangssignale abfragen	`MIO_Digin_Long`

Latch-Register nutzen	`MIO_Dig_Latch`
	`MIO_Dig_Read_Latch`
	`MIO_Dig_Write_Latch`

Ausgangssignale setzen und rücklesen	`MIO_Digout`
	`MIO_Digout_Long`
	`MIO_Get_Digout_Long`

Allgemein	
LEDs einstellen	`Check_LED`, `Set_LED`
Interrupts und Event-Eingang einstellen	`Event_Enable`, `Event_Read` `Event_Config`, `Trigger_Event`

Programmierung TiCo-Zugriff

Für den Zugriff auf den *TiCo*-Prozessor von der ADwin CPU sind die folgenden *ADbasic*-Befehle in der Datei `ADwinPro_All.inc` definiert. Die Befehle sind im Handbuch *TiCoBasic* und in der Online-Hilfe *ADbasic* erläutert.

Bereich	Befehle

Datenaustausch mit dem *TiCo*-Prozessor über globale Variablen	`P2_TDrv_Init`
	`P2_GetData_Long`, `P2_Get_Par`, `P2_Get_Par_Block`
	`P2_SetData_Long`, `P2_Set_Par`, `P2_Set_Par_Block`
	`P2_Get_TiCo_RingBuffer`, `P2_Set_TiCo_RingBuffer`
	`P2_RingBuffer_Empty`
	`P2_RingBuffer_Full`

Bereich	Befehle

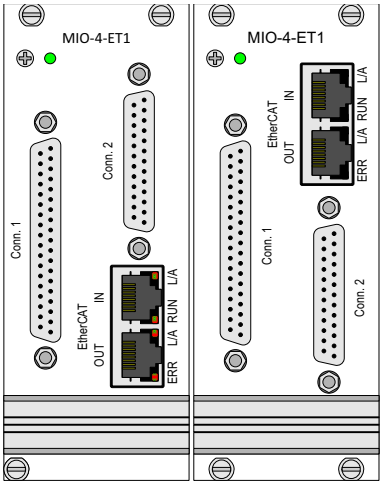
TiCo-Prozessor steuern	`P2_TiCo_Reset`, `P2_TiCo_Start`, `P2_TiCo_Stop`
	`P2_Get_TiCo_Bootloader_Status`
	`P2_Get_TiCo_Status`, `P2_Workload`

TiCo-Prozesse steuern	`P2_Process_Status`
	`P2_TiCo_Get_Processdelay`
	`P2_TiCo_Set_Processdelay`
	`P2_TiCo_Start_Process`
	`P2_Tico_Stop_Process`

TiCo-Programme übertragen	`P2_TiCo_Flash`, `P2_TiCo_Load`

5.4.2 Pro II-MIO-4-ET1 Rev. E

Das Modul Pro II-MIO-4-ET1 Rev. E ist mit folgender Hardware ausgerüstet (Variante siehe Pro II-MIO-4 Rev. E, Seite 25).:

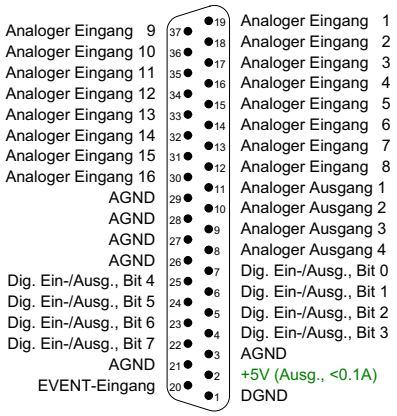


Rev. E02 Rev. E01

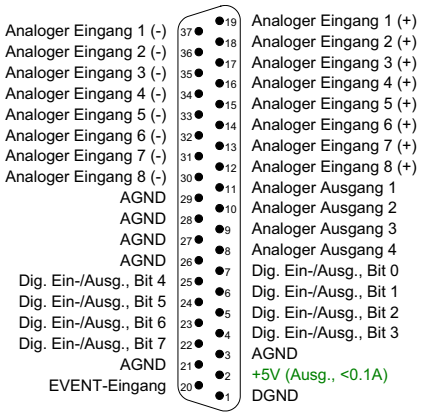
- 16 analoge Eingänge (16 single-ended oder 8 diff.), über Multiplexer mit 18 Bit ADC, Abtastrate bis 500 kHz
- 4 analoge Ausgänge mit 16 Bit DAC.
- 8 digitale Ein- und Ausgangskanäle mit TTL-Pegeln
- 1 Event-Eingang
- 4 Transistor-Ausgänge (TRA)
- 4 optisch isolierte Eingänge (OPT)
- 1 Zählerblock mit zwei 32 Bit-Zählern:
  - Ein Vor-/ Rückwärtszähler mit Takt/Richtungs-Auswertung oder mit Vierflankenauswertung zum Anschluss von Encodern.
  - Ein Zähler zur Periodendauer- und Tastverhältnismessung.
- 1 SSI-Decoder zum Anschluss eines Inkremental-Encoders
- 1 Ethercat-Schnittstelle (Slave)
- *TiCo*-Prozessor (TiCo1) mit 128 KiByte internem Speicher und 4 MiB externem SRAM-Speicher

  Der *TiCo*-Prozessor hat Zugriff auf alle Ein- und Ausgänge des Moduls. Nähere Hinweise zur Anwendung und Programmierung des *TiCo*-Prozessors entnehmen Sie bitte dem Handbuch *TiCoBasic*.

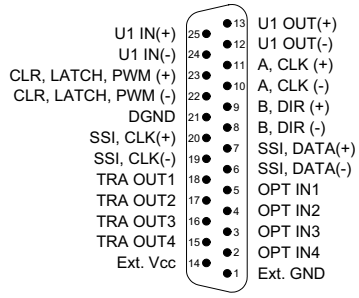
Wenn der *TiCo*-Bootloader programmiert ist, kann das Modul eigenständig und unabhängig vom CPU-Modul des *ADwin-Pro II*-Systems arbeiten.



Buchse `Conn1`, single-ended



Buchse `Conn1`, differentiell



Buchse `Conn2`

Abb. 21 – Pro II-MIO-4-ET1 Rev. E: Pinbelegungen

Modul-Revisionen

Die Unterschiede der Revisionsstände sind nachfolgend dargestellt:

Revision	Ausgabe-datum	Änderungen zur Vorgänger-Version
E1		Erst-Version
E2	Apr 2020	Steckeranordnung geändert

Die Revisionsbezeichnung befindet sich auf der Frontseite.

Unterschiedliche Revisions-Buchstaben bedeuten unterschiedliche Modul-Eigenschaften und sind separat dokumentiert. Der Revisionsbezeichnung angehängt ist eine Zählnummer für interne Zwecke.

Die Funktionen des Moduls sind in folgenden Abschnitten beschrieben:

- Analoge Eingänge
- Analoge Ausgänge
- Digitale Ein-/Ausgänge
- Transistor-Ausgänge
- Optokoppler-Eingänge
- SSI-Decoder
- Zählerblock
- EtherCAT-Schnittstelle
- TiCo-Prozessor
- Technische Daten
- Programmierung mit *ADbasic* und *TiCoBasic*

Analoge Eingänge

Das Modul Pro II-MIO-4-ET1 Rev. E hat 16 single ended-Eingänge oder 8 differentielle Eingänge (über Software einstellbar). Nach dem Einschalten ist das Modul auf 8 differentielle Eingänge eingestellt.

Die Eingänge AIN 1…AIN 16 bzw. AIN 1(+)/(-)…AIN 8(+)/(-) sind auf eine 37-polige D-Sub-Buchse `Conn1` geführt; Pinbelegung siehe Abb. 21.

Die Eingänge sind über einen Multiplexer mit dem ADC verbunden. Der ADC hat eine Auflösung von 18 Bit und kann mit einer Abtastrate von bis zu 500kSamples/s arbeiten.

Das Modul besitzt einen Eingangs-Spannungsbereich von ±10V und eine programmierbare Verstärkung von 1, 2, 4 oder 8. Der Abgleich der Verstärkung und des Offsets erfolgt per Software (siehe Kapitel 6 “Kalibrierung“).

Das Modul beinhaltet eine Ablaufsteuerung, die Messwerte an mehreren oder allen Eingangskanälen nacheinander einlesen kann.

Das Modul kann jeden Eingangskanal daraufhin überwachen, ob ein oberer oder unterer Grenzwert – die Grenzwerte sind für jeden Eingangskanal separat einstellbar – über- oder unterschritten wurde.

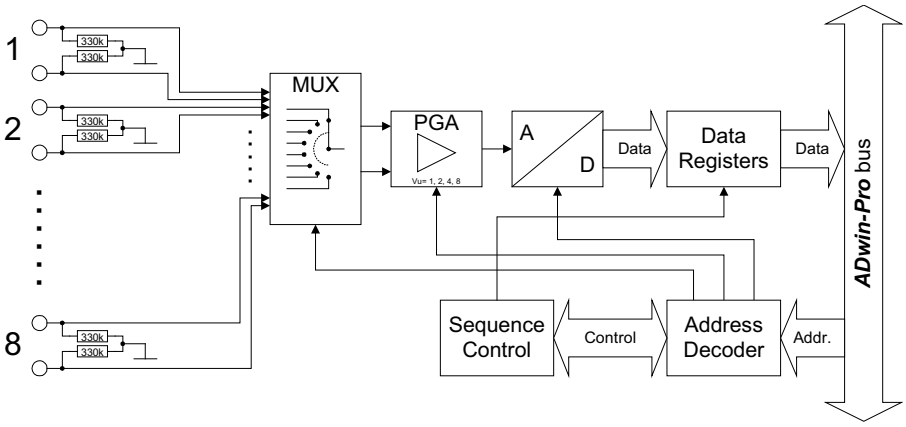


Abb. 22 – Pro II-MIO-4-ET1 Rev. E: Blockschaltbild Analogeingänge

Analoge Ausgänge

Das Modul Pro II-MIO-4-ET1 Rev. E hat 4 analoge Ausgänge mit einem DAC zu 16 Bit. Jeder Ausgang hat einen festen Tiefpass 1. Ordnung, um Störungen zu unterdrücken (f_g = 10MHz).

Der Ausgangs-Spannungsbereich der DAC ist fest auf ±10V bipolar eingestellt und lässt sich nicht verändern. Der Abgleich des Offset erfolgt per Software (siehe Kapitel 6 “Kalibrierung“).

Die Ausgänge sind auf die 37-polige D-Sub-Buchse `Conn1` geführt; Pinbelegung siehe Abb. 21.

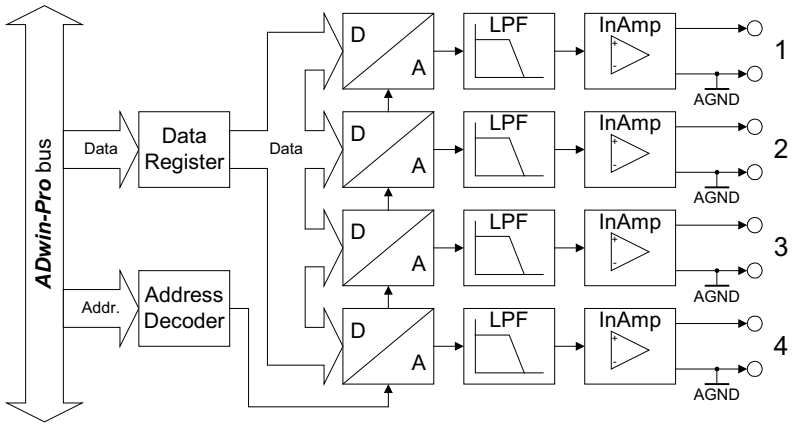


Abb. 23 – Pro II-MIO-4-ET1 Rev. E: Blockschaltbild Analogausgänge

Digitale Ein-/Ausgänge

Das Modul Pro II-MIO-4-ET1 Rev. E stellt 8 programmierbare Ein- und Ausgangskanäle mit TTL-Pegeln bereit. Die Kanäle können in Blöcken zu jeweils 4 mit *ADbasic*-Befehlen als Ein- oder Ausgänge konfiguriert werden. Nach dem Einschalten sind alle Kanäle als Eingänge konfiguriert.

Über den Trigger-Eingang EVENT kann ein Signal (Trigger) einen Prozess auslösen, der dann sofort und vollständig abgearbeitet wird (siehe *ADbasic*-Handbuch).

Die Digitalkanäle sind auf die 37-polige D-Sub-Buchse `Conn1` geführt; Pinbelegung siehe Abb. 21.

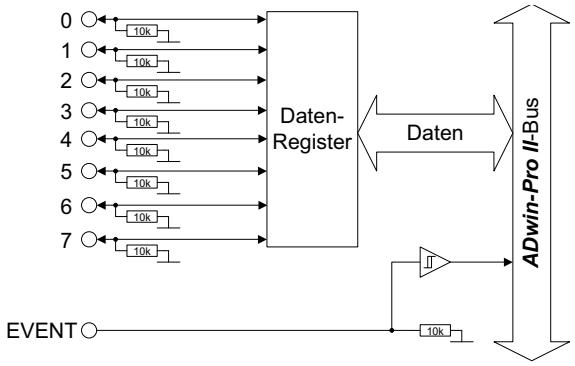


Abb. 24 – Pro II-MIO-4-ET1 Rev. E: Blockschaltbild Digitalkanäle

Transistor-Ausgänge

Das Modul Pro II-MIO-4-ET1 Rev. E stellt 4 galvanisch getrennte Transistor-Schaltausgänge bereit. Die Ausgänge schalten nach V_{CC}.

Die gemeinsame Schaltspannung V_{CC} muss durch eine externe Spannungsversorgung an den Pins `EXT VCC` und `EXT GND` zugeführt werden. Beachten Sie, dass `EXT GND` auch für die Optokoppler-Eingänge genutzt wird.

Die Kanäle wie auch der Event-Eingang sind optisch vom System-Stromkeis isoliert.

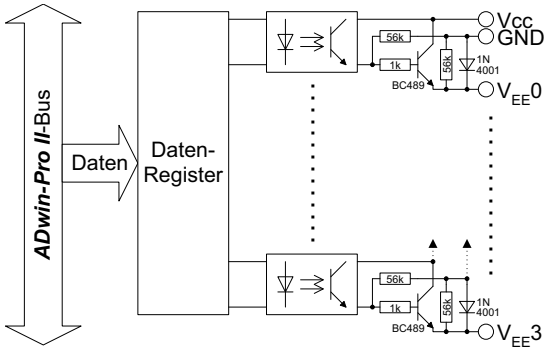


Abb. 25 – Pro II-MIO-4-ET1 Rev. E: Blockschaltbild Transistor-Ausgänge

Die Ausgänge sind auf die 25-polige D-Sub-Buchse `Conn2` geführt; Pinbelegung siehe Abb. 21.

Optokoppler-Eingänge

Das Modul Pro II-MIO-4-ET1 Rev. E stellt 4 Kanäle mit optisch isolierten digitalen Eingängen bereit. Die Eingangs-Spannungsbereiche sind für jeden Eingang separat über Jumper einstellbar (5V, 12V, 24V). Die Voreinstellung ist 24V. Die Schaltzeit von nur 100ns erlaubt das Einlesen von schnellen digitalen Signalen.

Jeder Kanal ist vom Systemstromkreis und von den anderen Eingängen optisch isoliert, wie auch der Event-Eingang.

Beachten Sie, dass `EXT GND` auch für die Transistor-Ausgänge genutzt wird.

Die Eingänge sind auf die 25-polige D-Sub-Buchse `Conn2` geführt; Pinbelegung siehe Abb. 21.

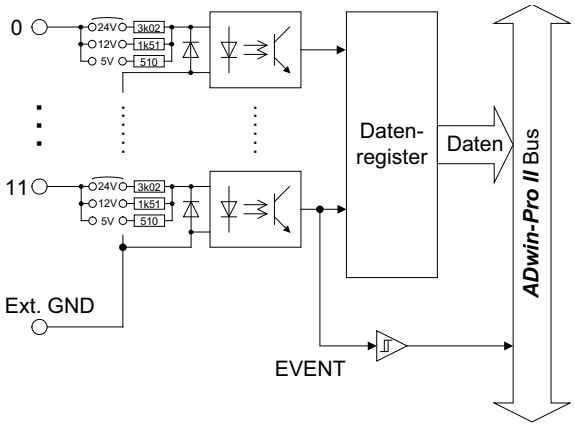


Abb. 26 – Pro II-MIO-4-ET1 Rev. E: Blockschaltbild Optokoppler-Eingänge

SSI-Decoder

An den SSI-Decoder kann ein Inkremental-Encoder mit SSI-Schnittstelle angeschlossen werden. Die Signale sind differentiell und haben RS422/485-Pegel.

Programmierbar sind die Taktraten über einen Vor-Teiler (von ca. 6 kHz bis 12,5 MHz) ebenso wie die Auflösung des Decoders (bis 32 Bit).

Die Decoder-Eingänge sind auf die 25-polige D-Sub-Buchse `Conn2` geführt; Pinbelegung siehe Abb. 21.

Über die beiden Pins `U1 IN` können Sie eine Spannung einspeisen, die an den Pins `U1 OUT` z. B. für einen angeschlossenenen Encoder zur Verfügung steht. Die Stromstärke soll 0,1 A nicht überschreiten; zwischen Ein- und Ausgangs-Pins ist jeweils eine entsprechende Sicherung eingebaut.

Zählerblock

Das Modul Pro II-MIO-4-ET1 Rev. E stellt einen konfigurierbaren Zählerblock zur Verfügung. Der Zählerblock enthält zwei 32 Bit-Zähler: Einerseits einen Vor-/ Rückwärtszähler mit Takt/Richtungs-Auswertung oder Vierflankenauswertung zum Anschluss von Encodern. Zum anderen einen Zähler zur Periodendauer- und Tastverhältnismessung. Beide Zähler können parallel genutzt werden.

Die Zählereingänge arbeiten differentiell.

Vor-/Rückwärtszähler

Der Vor-/ Rückwärtszähler kann in 2 Betriebsarten arbeiten:

– Takt/Richtungs-Auswertung (CLK- und DIR-Signale)

  Eine negative Flanke am CLK-Eingang löst einen Zählimpuls am 32 Bit-Zähler aus. Das DIR-Signal bestimmt die Zählrichtung des Zählers, TTL high bedeutet Hochzählen, TTL low bedeutet Herunterzählen.

  Sie können den Zählerstand programmgesteuert ins Latch übernehmen oder den Zähler durch ein externes CLR-/LATCH-Signal beeinflussen.

  Das CLR-/LATCH-Signal kann je nach Programmierung ein Löschen (CLR) des Zählerstands oder die Übernahme des Zählerstands ins Latch (LATCH) bewirken. Diese Funktion wird erst wirksam, wenn sie durch **`P2_Cnt_Clear_Enable`** oder **`P2_Cnt_Latch_Enable`** freigegeben ist.

  Das Löschen oder Latchen des Zählers erfolgt bei einem High-Pegel am Eingang CLR/LATCH. Beim Latchen lässt sich aus der Differenz von zwei gelesenen Latch-Werten die Frequenz der Messung ermitteln,

denn die Differenz gibt die Anzahl der Impulse zwischen den beiden Lesevorgängen an.

– Vierflankenauswertung (A- und B-Signale)

  Die Vierflankenauswertung wandelt die (möglichst um 90° phasenverschobenen) Signale eines angeschlossenen Inkremental-Encoders an A- und B-Eingang in ein CLK- und DIR-Signal um. Hierzu sind die Eingänge in *ADbasic* entsprechend zu programmieren (siehe „*ADwin-Pro* Systembeschreibung, Programmierung in *ADbasic* „).

  Da jede Flanke des A- und B-Signales einen Zählimpuls erzeugt, wird die Auflösung um den Faktor 4 vergrößert. Besitzt der Encoder ein Referenz-Signal, so kann dies (nach Freigabe des CLR- bzw. LATCH-Einganges) zum Löschen oder Latchen des Zählers genutzt werden. Das Löschen des Zählers erfolgt, wenn die Signale A, B und CLR auf logisch „1“ stehen (über Software umstellbar: Löschen, wenn nur das CLR-Signal auf logisch „1“ steht).

PWM-Zähler

Der PWM-Zähler des Zählerblocks wertet die Signale an den PWM-Eingängen aus. Mit Standard-Befehlen können folgende Daten direkt ausgelesen werden:

– Frequenz und Tastverhältnis (**`P2_Cnt_Get_PW`**)

– Eintastzeit und Austastzeit (**`P2_Cnt_Get_PW_HL`**)

Die Zähler-Eingänge sind auf die 25-polige D-Sub-Buchse `Conn2` geführt; Pinbelegung siehe Abb. 21.

Wenn Sie die PWM-Zähler wie im Beispiel mit den Standard-Befehlen auswerten, benötigen Sie keine Kenntnisse über die PWM-Register.
Wenn Sie aber andere Auswertungs-Möglichkeiten benötigen, stehen Ihnen für jeden PWM-Zähler mehrere Register zur Verfügung, mit denen Sie eine Lösung erzielen können. Sie finden eine Beschreibung der PWM-Register beim Modul Pro II-CNT-x Rev. E (Seite 141).

EtherCAT-Schnittstelle

Das Modul stellt einen Feldbusknoten mit der Funktionalität eines EtherCAT-Slave zur Verfügung. Alle Einstellungen werden per Software vorgenommen.

Nach dem Einschalten müssen Sie den Feldbusknoten in *TiCoBasic* initialisieren. Mit der Initialisierung wird die Größe der Ein- und Ausgangsbereiche festgelegt.

Es gibt je einen Bereich für eingehende und für ausgehende Daten; jeder Bereich hat eine Größe von 16 Long (= 64 Byte). Die Begriffe „Eingang“ und „Ausgang“ sind aus Sicht des Feldbus-Controllers zu sehen.

Die Schnittstelle hat je eine Buchse vom Typ RJ45 für den Dateneingang (`IN`) und für den Datenausgang (`OUT`). An jeder Buchse ist oben eine LED „Link / Activity“ (`L/A`), die den Betriebszustand des Knotens im EtherCAT-Bus anzeigt. Die beiden weiteren LEDs zeigen den Status der EtherCAT-Zustandsmaschine (`RUN`) und das Auftreten von Kommunkationsfehlern (`ERR`) an.

LED	Status	Bedeutung
Link / Activity	Aus	Nicht online (oder keine Stromversorgung).
	An	Feldbusknoten online, kein Datenaustausch.
	flackernd	Feldbusknoten online, mit Datenaustausch.
RUN	Aus	Status INIT: Die Schnittstelle wird initialisiert (oder keine Stromversorgung).
	blinkt grün	Status PRE-OP: Schnittstelle hat Kontakt zum Bus-Master.
	leuchtet einmal grün	Status SAFE-OP: Schnittstelle kann Daten vom Bus lesen, aber nicht senden.
	leuchtet grün	Status OP: Schnittstelle ist vollständig eingerichtet, Ein- und Ausgänge sind aktiv.
	leuchtet rot	Status EXCEPTION: Ausnahmesituation.
ERR	Aus	Kommunikation arbeitet ohne Fehler (oder keine Stromversorgung).
	blinkt rot	Fehler bei der Konfiguration.
	leuchtet einmal rot	Lokaler Fehler in der Schnittstelle; der EtherCAT-Status wurde geändert.
	leuchtet doppelt rot	Fehler durch Zeitüberschreitung (timeout).
	leuchtet rot	Kritischer Kommunikationsfehler.

Abb. 27 – Pro II-MIO-4-ET1 Rev. E: Bedeutung der EtherCAT-LED

Wenn beide LEDs `RUN` und `ERR` rot leuchten, ist ein gravierender Fehler in der Schnittstelle aufgetreten. Melden Sie sich dann bitte beim Support von Jäger Messtechnik; die Adresse finden Sie auf der vorderen Umschlagseite des Handbuchs, innen.

Der Zugriff auf den *ADwin*-EtherCAT-Slave ist nur in *TiCoBasic* möglich. Sie konfigurieren den Slave mit dem Befehl **`MIO_ECAT_Init`**. Anschließend projektieren Sie den EtherCAT-Bus mit einem – zum Bus-Master passenden – Konfigurations-Tool, beispielsweise mit dem Programm „TwinCAT System Manager“ der Firma Beckhoff als EtherCAT-Master. Kopieren Sie dazu die Beschreibungsdatei `ADwin-EtherCAT.xml` des EtherCAT-Moduls aus dem Ordner `C:\ADwin\Fieldbus\EtherCAT\` in das Verzeichnis des Konfigurations-Tools.

TiCo-Prozessor

Das Modul besitzt einen frei programmierbaren *TiCo*-Prozessor (TiCo1) mit 64 KiByte internem Programmspeicher, 64 KiByte internem Datenspeicher und 4 MiByte externem SRAM-Speicher. Sie programmieren den *TiCo*-Prozessor in *TiCoBasic*.

Der *TiCo*-Prozessor hat – wie auch die *ADwin*-CPU – Zugriff auf alle Ein- und Ausgangskanäle, analog wie digital. Nähere Hinweise zur Anwendung und Programmierung des *TiCo*-Prozessors entnehmen Sie bitte dem Handbuch *TiCoBasic*.

Wenn Sie ein *TiCoBasic*-Programm im *TiCo*-Bootloader ablegen, wird das Programm beim Einschalten der Stromversorgung in den *TiCo*-Prozessor geladen und gestartet. Auf diese Weise kann das Modul eigenständig und unabhängig vom CPU-Modul des *ADwin-Pro II*-Systems arbeiten.

Technische Daten

Analoge Eingänge	
Eingangskanäle	16 single ended / 8 differentiell, Multiplexer
Auflösung	18 Bit
Wandlungszeit	max. 2 µs
Abtastrate	max. 500 ksps
Multiplexer Einschwing-zeit	5 µs
Messbereich	±10 V
Verstärkung	1, 2, 4, 8 per Software einstellbar
Genauigkeit INL	typisch ±4 LSB
Genauigkeit DNL	max. ±1 LSB
Eingangswiderstand	330 kΩ, ±2%
Spannungsfestigkeit	±20 V
Offsetfehler	abgleichbar
Offsetdrift	±30 ppm/°C
Analoge Ausgänge	
Ausgangskanäle	4
Auflösung	16 Bit
Einschwingzeit	9 µs (auf 0,01% FSR)
Ausgangsspannung	±10 V
Max. Ausgangsstrom	±5 mA pro Kanal für optimale Funktion
Genauigkeit INL	±2 LSB typisch
Genauigkeit DNL	±1 LSB typisch
Offsetfehler	abgleichbar
Verstärkungsfehler	abgleichbar
Digitale Ein-/Ausgänge	
Digitale Eingänge	8 Kanäle mit TTL-Logik, konfigurierbar in Gruppen zu 4 Kanälen
Pull-Down-Widerstand	10 kΩ
V_{IH}	min. 2 V
V_{IL}	max. 0,8 V
I_{IH}	max. 1 µA
I_{IL}	max. 0,01 mA
Spannungsbereich	-0,5 V … +5,5 V
Ausgangsstrom	max. ±24 mA pro Kanal, max. ±50 mA je Block (4 Kanäle) über V_{CC} oder GND
Event-Eingang	TTL-Logik
Power-Up-Status	Alle digitalen Kanäle als Eingänge
Transistor-Ausgänge	
Ausgangskanäle	4
Schaltspannung V_{CC}	5…30 V DC durch externe Spannungsversorgung
Schaltstrom	200 mA max. pro Kanal
Spannungsabfall	0,5 V

Abb. 28 – Pro II-MIO-4-ET1 Rev. E: Spezifikation

Schaltzeit	2,5µs
Isolation	42V Kanal zu Kanal / Kanal zu Masse gemeinsame Masse für alle OPT- und TRA-Kanäle
Optokoppler-Eingänge	
Eingangskanäle	4
Eingangsstrom	typ. 3,5mA / max. 7,5mA
Eingangs-Spannungs-bereich (über Jumper wählbar)	0…5V / 0…12V / 0…24V
Schaltschwelle für 0-low	0…0,8V / 0…1,6V / 0…3,2V
Schaltschwelle für 1-high	4,5…5V / 10…12V / 20…24V
Spannungsfestigkeit	-5V … 8V / -5V … 16V / -5V … 30V /
Schaltzeit	100ns
Isolation	42V Kanal zu Kanal / Kanal zu Masse gemeinsame Masse für alle OPT- und TRA-Kanäle
Zähler	
Anzahl	1 Universalzähler
Zählerbreite	32 Bit
Referenztakt	50MHz
Taktfrequenz Vierflankenauswertung	12,5MHz max. (bei 90° Phasenverschiebung der Signale)
Taktfrequenz Vor- / Rückwärtszähler	15MHz max.
Referenzfrequenz PWM-Analyse	100MHz
Eingangspegel	RS422/485 kompatibel (5V differentiell,120 Ω Bus-Abschlusswiderstand)
Isolation	keine
SSI-Decoder	
Anzahl	1
Taktfrequenz SSI-Deco-der (CLK)	6,1kHz … 12,5MHz
Ethercat-Schnittstelle	
Anzahl	1
Allgemein	
TiCo-Prozessor	Prozessortyp: TiCo1 Taktfrequenz: 50MHz Speichergröße: 64kiB PM intern, 64kiB DM intern, 4MiB externes SRAM
Steckerverbindungen	37-polige D-Sub-Buchse `Conn1` 25-polige D-Sub-Buchse `Conn2` 2 Buchsen Typ RJ45 für EtherCAT-Dateneingang (`IN`) und -Datenausgang (`OUT`)
Modulbreite	10 TE

Abb. 28 – Pro II-MIO-4-ET1 Rev. E: Spezifikation

Programmierung

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Programmierung in ADbasic

Bereich	Befehle
Analoge Eingänge	
Eingänge einstellen auf single-ended oder differentiell	`P2_SE_Diff`

Einzelmessung durchführen – vollständig oder schrittweise	`P2_ADC`, `P2_ADC24`
	`P2_Set_Mux`, `P2_Start_Conv`
	`P2_Wait_EOC`
	`P2_Read_ADC`, `P2_Read_ADC24`

Messwert lesen und neue Messung starten	`P2_Read_ADC_SConv` `P2_Read_ADC_SConv24`

Ablaufsteuerung anwenden	`P2_Seq_Init`, `P2_Seq_Start`
	`P2_Seq_Read`, `P2_Seq_Read24`
	`P2_Seq_Read_Packed`
	`P2_Seq_Wait`

Grenzwerte überwachen	`P2_ADC_Read_Limit` `P2_ADC_Set_Limit`
Analoge Ausgänge	
Ausgabe durchführen	`P2_DAC`, `P2_DAC4`, `P2_DAC4_Packed`

Ausgabe schrittweise durchführen	`P2_Write_DAC`, `P2_Write_DAC4`
	`P2_Write_DAC4_Packed`
	`P2_Write_DAC32`
	`P2_Start_DAC`

Digitale Ein-/Ausgänge	
Ein- und Ausgänge konfigurieren	`P2_MIO_DigProg`
Eingangssignale abfragen	`P2_MIO_Digin_Long`

Latch-Register nutzen	`P2_MIO_Dig_Latch`
	`P2_MIO_Dig_Read_Latch`
	`P2_MIO_Dig_Write_Latch`

Ausgangssignale setzen und rücklesen	`P2_MIO_Digout`
	`P2_MIO_Digout_Long`
	`P2_MIO_Get_Digout_Long`

Transistor-Ausgänge	

Ausgangssignale setzen und rücklesen	`P2_Digout`, `P2_Digout_Long`
	`P2_Digout_Bits`
	`P2_Get_Digout_Long`

Latch-Register nutzen	`P2_Dig_Latch`, `P2_Sync_All` `P2_Dig_Write_Latch`
Optokoppler-Eingänge	
Eingangssignale abfragen	`P2_Digin_Edge` `P2_Digin_Long`
Latch-Register nutzen	`P2_Dig_Latch`, `P2_Sync_All` `P2_Dig_Read_Latch`
Zählerblock	
Zähler konfigurieren	`P2_Cnt_Enable`, `P2_Cnt_Mode`

Bereich	Befehle
Zähler ansteuern	P2_Cnt_Clear, P2_Cnt_Get_Status P2_Cnt_Latch, P2_Cnt_Sync_Latch P2_Cnt_Read, P2_Cnt_Read4 P2_Cnt_Read_Latch P2_Cnt_Read_Latch4
PWM-Zähler ansteuern	P2_Cnt_PW_Enable, P2_Cnt_PW_Latch P2_Cnt_Get_PW, P2_Cnt_Get_PW_HL
SSI-Decoder	
SSI-Decoder ansteuern	P2_SSI_Mode, P2_SSI_Set_Bits P2_SSI_Set_Clock P2_SSI_Set_Delay, P2_SSI_Read P2_SSI_Read2 P2_SSI_Start, P2_SSI_Status
EtherCat-Schnittstelle	
Schnittstelle ansteuern	nur in *TiCoBasic*
Allgemein	
LEDs einstellen	P2_Check_LED, P2_Set_LED
Abläufe synchronisieren	P2_Sync_All, P2_Sync_Enable P2_Sync_Stat
Interrupts und Event-Eingang einstellen	P2_Event_Enable, P2_Event_Read P2_Event_Config

Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe erläutert.

Programmierung in TiCoBasic

Das Modul kann mit *TiCoBasic*-Befehlen programmiert werden. Die Befehle sind in der Online-Hilfe *TiCoBasic* erläutert.

Die Include-Datei `MIO_TiCo.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Analoge Eingänge	
Eingänge einstellen auf single-ended oder differentiell	`SE_Diff`

Einzelmessung durchführen – vollständig oder schrittweise	`ADC`, `ADC24`
	`Set_Mux`, `Start_Conv`, `Wait_EOC`
	`Read_ADC`, `Read_ADC24`

Messwert lesen und neue Messung starten	`Read_ADC_SConv` `Read_ADC_SConv24`

Ablaufsteuerung anwenden	`Seq_Init`, `Seq_Start`
	`Seq_Read`, `Seq_Read24`
	`Seq_Wait`

Grenzwerte überwachen	`ADC_Read_Limit` `ADC_Set_Limit`
Analoge Ausgänge	
Ausgabe durchführen	`DAC`
Ausgabe schrittweise durchführen	`Write_DAC`, `Write_DAC32` `Start_DAC`
Digitale Ein-/Ausgänge	
Ein- und Ausgänge konfigurieren	`MIO_DigProg`
Eingangssignale abfragen	`MIO_Digin_Long`

Latch-Register nutzen	`MIO_Dig_Latch`
	`MIO_Dig_Read_Latch`
	`MIO_Dig_Write_Latch`

Ausgangssignale setzen und rücklesen	`MIO_Digout`
	`MIO_Digout_Long`
	`MIO_Get_Digout_Long`

Transistor-Ausgänge	
Ausgangssignale setzen und rücklesen	`MIO_Digout`, `MIO_Digout_Long` `MIO_Get_Digout_Long`
Latch-Register nutzen	`MIO_Dig_Latch` `MIO_Dig_Write_Latch`
Optokoppler-Eingänge	
Eingangssignale abfragen	`MIO_Digin_Long`
Latch-Register nutzen	`MIO_Dig_Latch` `MIO_Dig_Read_Latch`
Zählerblock	
Zähler konfigurieren	`Cnt_Enable`, `Cnt_Mode`

Zähler ansteuern	`Cnt_Clear`, `Cnt_Get_Status`
	`Cnt_Latch`, `Cnt_Read`
	`Cnt_Read_Int_Register`
	`Cnt_Sync_Latch`, `Cnt_Read_Latch`

PWM-Zähler ansteuern	`Cnt_PW_Enable`, `Cnt_PW_Latch` `Cnt_Get_PW_HL`

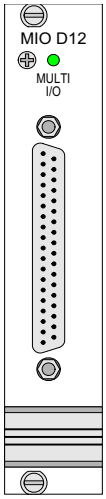
Bereich	Befehle
SSI-Decoder	
SSI-Decoder ansteuern	SSI_Mode, SSI_Set_Bits SSI_Set_Clock SSI_Set_Delay, SSI_Read SSI_Start, SSI_Status
EtherCat-Schnittstelle	
Schnittstelle ansteuern	MIO_ECAT_Init MIO_ECAT_Get_Version MIO_ECAT_Check_State_Transition MIO_ECAT_Read_Data_16L MIO_ECAT_Write_Data_16L
Allgemein	
Aktionen synchron starten	P2_Sync_All, P2_Sync_Enable P2_Sync_Stat
LEDs einstellen	Check_LED, Set_LED
Interrupts und Event-Eingang einstellen	Event_Enable, Event_Read Event_Config, Trigger_Event

Programmierung TiCo-Zugriff

Für den Zugriff auf den *TiCo*-Prozessor von der ADwin CPU sind die folgenden *ADbasic*-Befehle in der Datei `ADwinPro_All.inc` definiert. Die Befehle sind im Handbuch *TiCoBasic* und in der Online-Hilfe *ADbasic* erläutert.

Bereich	Befehle
Datenaustausch mit dem *TiCo*-Prozessor über globale Variablen	P2_TDrv_Init P2_GetData_Long, P2_Get_Par, P2_Get_Par_Block P2_SetData_Long, P2_Set_Par, P2_Set_Par_Block P2_Get_TiCo_RingBuffer, P2_Set_TiCo_RingBuffer P2_RingBuffer_Empty P2_RingBuffer_Full
TiCo-Prozessor steuern	P2_TiCo_Reset, P2_TiCo_Start, P2_TiCo_Stop P2_Get_TiCo_Bootloader_Status P2_Get_TiCo_Status, P2_Workload
TiCo-Prozesse steuern	P2_Process_Status P2_TiCo_Get_Processdelay P2_TiCo_Set_Processdelay P2_TiCo_Start_Process P2_Tico_Stop_Process
TiCo-Programme übertragen	P2_TiCo_Flash, P2_TiCo_Load

5.4.3 Pro II-MIO-D12 Rev. E



Das Modul Pro II-MIO-D12 Rev. E ist mit folgender Hardware ausgerüstet:

- 12 Transistor-Ausgänge (TRA)
- 12 optisch isolierte Eingänge (OPT)
- 1 Event-Eingang (siehe Optokoppler-Eingänge)
- 2 Zählerblöcke mit je 2 parallel nutzbaren 32 Bit-Zählern:
  - Ein Vor-/ Rückwärtszähler mit Takt/Richtungs-Auswertung oder mit Vierflankenauswertung zum Anschluss von Encodern.
  - Ein Zähler zur Periodendauer- und Tastverhältnismessung.
- 1 SSI-Decoder zum Anschluss eines Inkremental-Encoders
- *TiCo*-Prozessor (TiCo1) mit 56 KiByte internem Speicher

  Der *TiCo*-Prozessor hat Zugriff auf alle Ein- und Ausgänge des Moduls. Nähere Hinweise zur Anwendung und Programmierung des *TiCo*-Prozessors entnehmen Sie bitte dem Handbuch *TiCoBasic*.

Wenn der *TiCo*-Bootloader programmiert ist, kann das Modul eigenständig und unabhängig vom CPU-Modul des *ADwin-Pro II*-Systems arbeiten.

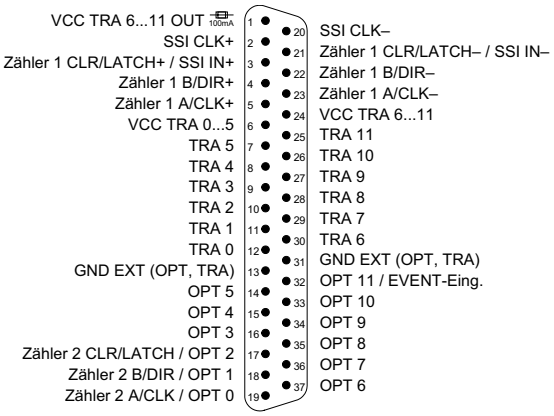


Abb. 29 – Pro II-MIO-D12 Rev. E: Pinbelegungen

Einige Pins sind doppelt belegt, aber die damit verbundenen Funktionen sind gleichzeitig nutzbar.

Die Funktionen des Moduls sind in folgenden Abschnitten beschrieben:

- Transistor-Ausgänge (TRA)
- Optokoppler-Eingänge (OPT)
- SSI-Decoder
- Zählerblock
- TiCo-Prozessor
- Technische Daten
- Programmierung mit *ADbasic* und *TiCoBasic*

Transistor-Ausgänge

Das Modul Pro II-MIO-D12 Rev. E stellt 12 galvanisch getrennte Transistor-Schaltausgänge bereit, Pinbelegung siehe Abb. 29. Die Ausgänge schalten

nach V_{CC}. Alle TRA- uind OPT-Kanäle verwenden eine gemeinsame GND-Leitung.

Die Schaltspannungen V_{CC} muss durch eine externe Spannungsversorgung zugeführt werden. Es können 2 Spannungen, je eine Spannung für 6 Schaltausgänge (`TRA 0` … `TRA 5` und `TRA 6` … `TRA 11`), eingespeist werden.

Die Schaltspannung für die Ausgänge `TRA 6` … `TRA 11` steht außerdem auf dem Pin `VCC TRA 6…11 OUT` zur Versorgung externer Geräte zur Verfügung. Der Ausgang ist mit 100 mA abgesichert.

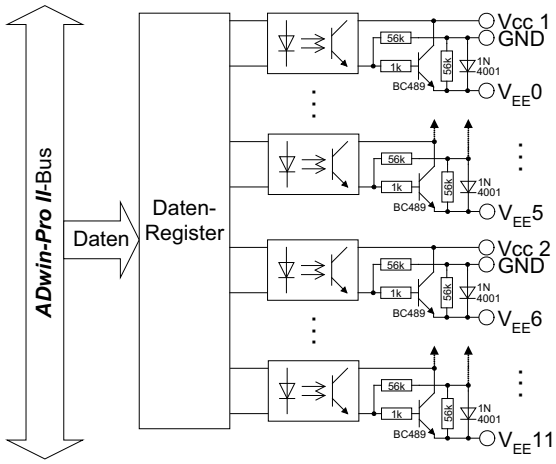
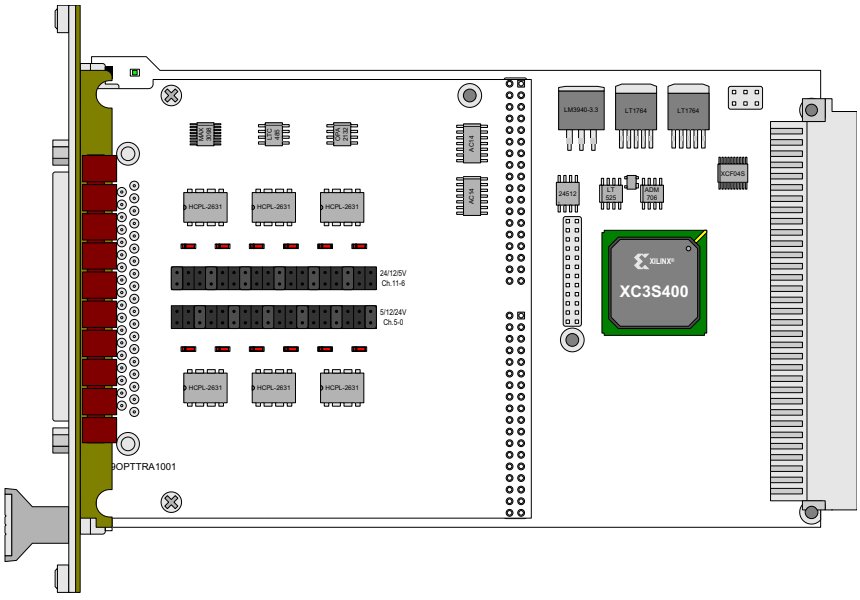


Abb. 30 – Pro II-MIO-D12 Rev. E: Blockschaltbild Transistor-Ausgänge

Das Modul kann selbstständig Pegel zu bestimmten Zeitpunkten auf den TRA-Ausgängen ausgeben. Ein FIFO dient als Zwischenspeicher für die vom Benutzer festgelegten Pegel und Zeitpunkte, maximal 511 Wertepaare. Der Ausgabezeitpunkt kann auf 5 ns genau festgelegt werden.
Das FIFO kann entweder für die Flankenüberwachung von OPT-Eingängen oder für die Pegelausgabe an TRA-Ausgängen verwendet werden.

Optokoppler-Eingänge

Das Modul Pro II-MIO-D12 Rev. E stellt 12 Kanäle mit optisch isolierten digitalen Eingängen bereit. Der Eingangs-Spannungsbereich ist für jeden Eingang separat über Jumper auf der Platine einstellbar (5 V, 12 V, 24 V).
Die Schaltzeit von nur 100 ns erlaubt das Einlesen von schnellen digitalen Signalen.



Das Modul kann automatisch die Flanken an Eingangskanälen überwachen. Bei einer Änderung wird der aktuelle Pegelstand gemeinsam mit einem Zeitstempel in einem FIFO zwischengespeichert. Die FIFO-Daten können ausgelesen und weiter verarbeitet werden.
Das FIFO kann entweder für die Flankenüberwachung von OPT-Eingängen oder für die Pegelausgabe an TRA-Ausgängen verwendet werden.

Jeder OPT-Kanal ist vom Systemstromkreis und von den anderen Eingängen optisch isoliert. Alle OPT- uind TRA-Kanäle verwenden eine gemeinsame GND-Leitung.

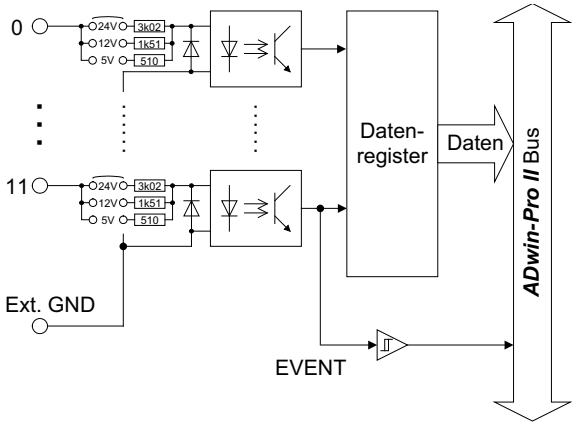


Abb. 31 – Pro II-MIO-D12 Rev. E: Blockschaltbild Optokoppler-Eingänge

Der optisch isolierte Eingang `OPT 11` kann auch als Event-Eingang verwendet werden. Wenn der Event-Eingang dafür freigegeben ist, startet ein Trigger-Signal am Event-Eingang den extern getriggerten *ADbasic*-Prozess.

Die Eingänge `OPT 0`…`OPT 2` können zusätzlich als Eingänge für den Zähler 2 genutzt werden. Technisch ist es erlaubt, beide belegten Funktionen gleichzeitig zu nutzen.

SSI-Decoder

An den SSI-Decoder kann ein Inkremental-Encoder mit SSI-Schnittstelle angeschlossen werden. Die Signale sind differentiell und haben RS422/485-Pegel.

Programmierbar sind die Taktraten über einen Vor-Teiler (von ca. 6kHz bis 12,5MHz) ebenso wie die Auflösung des Decoders (bis 32 Bit).

Pinbelegung der Decoder-Eingänge siehe Abb. 29. Der Eingang `SSI IN+/-` kann zusätzlich als Eingang `CLR/LATCH` für den Zähler 1 genutzt werden. Technisch ist es erlaubt, beide belegten Funktionen gleichzeitig zu nutzen.

Der Decoder kann entweder (auf Anforderung) einen einzelnen Wert auslesen oder aber kontinuierlich den aktuellen Wert bereit stellen.

Der SSI-Decoder ist per Software einstellbar:

– Taktrate ist einstellbar von 6,1kHz…12,5MHz mit **`SSI_Set_Clock`**.

– Auflösung ist einstellbar bis 32 Bit mit **`SSI_Set_Bits`**.

Zählerblock

Das Modul Pro II-MIO-D12 Rev. E stellt 2 konfigurierbare Zählerblöcke zur Verfügung. Ein Zählerblock enthält zwei 32 Bit-Zähler: Einerseits einen Vor-/Rückwärtszähler mit Takt/Richtungs-Auswertung oder Vierflankenauswertung zum Anschluss von Encodern. Zum anderen einen Zähler zur Periodendauer- und Tastverhältnismessung (PWM). Beide Zähler eines Blocks können parallel genutzt werden.

Zähler 1 hat differentielle Eingänge, Zähler 2 hat optisch isolierte Eingänge (single ended).

Beachten Sie, dass die Pins für die Zählereingänge mehrfach belegt sind. Technisch ist es erlaubt, alle belegten Funktionen gleichzeitig zu nutzen, es ist aber in der Regel nicht sinnvoll.

Vor-/Rückwärtszähler

Der Vor-/ Rückwärtszähler kann in 2 Betriebsarten arbeiten:

– Takt/Richtungs-Auswertung (CLK- und DIR-Signale)

  Eine negative Flanke am Eingang CLK löst einen Zählimpuls am 32 Bit-Zähler aus. Das Signal an DIR bestimmt die Zählrichtung des Zählers, TTL high bedeutet Hochzählen, TTL low bedeutet Herunterzählen; das Signal kann per Software invertiert werden.

  Sie können die Signale an den Eingängen CLK und DIR per Software (Befehl **P2_Cnt_Mode**) invertieren und damit sowohl die auslösende Flanke als auch die Zählrichtung ändern.

  Sie können den Zählerstand programmgesteuert ins Latch übernehmen oder den Zähler durch ein externes Signal an CLR/LATCH beeinflussen.

  Das CLR-/LATCH-Signal kann je nach Programmierung ein Löschen (CLR) des Zählerstands oder die Übernahme des Zählerstands ins Latch (LATCH) bewirken. Diese Funktion wird erst wirksam, wenn sie durch **P2_Cnt_Clear_Enable** oder **P2_Cnt_Latch_Enable** freigegeben ist.

  Das Löschen oder Latchen des Zählers erfolgt bei einem High-Pegel am Eingang CLR/LATCH; das Signal kann per Software invertiert werden. Beim Latchen lässt sich aus der Differenz von zwei gelesenen Latch-Werten die Frequenz der Messung ermitteln, denn die Differenz gibt die Anzahl der Impulse zwischen den beiden Lesevorgängen an.

– Vierflankenauswertung (A- und B-Signale)

  Die Vierflankenauswertung wandelt die (möglichst um 90° phasenverschobenen) Signale eines angeschlossenen Inkremental-Encoders an den Eingängen A und B in ein CLK- und DIR-Signal um. Hierzu sind die Eingänge in *ADbasic* entsprechend zu programmieren (siehe „*ADwin-Pro* Systembeschreibung, Programmierung in *ADbasic* „).

  Da jede Flanke des A- und B-Signales einen Zählimpuls erzeugt, wird die Auflösung um den Faktor 4 vergrößert. Besitzt der Encoder ein Referenz-Signal, so kann dies (nach Freigabe des CLR- bzw. LATCH-Einganges) zum Löschen oder Latchen des Zählers genutzt werden. Das Löschen des Zählers erfolgt, wenn die Signale A, B und CLR auf logisch „1“ stehen (über Software umstellbar: Löschen, wenn nur das CLR-Signal auf logisch „1“ steht).

PWM-Zähler

Der PWM-Zähler des Zählerblocks wertet die Signale am gewählten PWM-Eingang aus. Sie legen mit **P2_Cnt_Mode** den Eingangs-Pin (A/CLK, B/DIR oder CLR/LATCH) und die auslösende Flanke fest.

Mit Standard-Befehlen können folgende Daten direkt ausgelesen werden:

– Frequenz und Tastverhältnis (**P2_Cnt_Get_PW**)

– Eintastzeit und Austastzeit (**P2_Cnt_Get_PW_HL**)

Wenn Sie die PWM-Zähler wie im Beispiel mit den Standard-Befehlen auswerten, benötigen Sie keine Kenntnisse über die PWM-Register.
Wenn Sie aber andere Auswertungs-Möglichkeiten benötigen, stehen Ihnen für jeden PWM-Zähler mehrere Register zur Verfügung, mit denen Sie eine Lösung erzielen können. Sie finden eine Beschreibung der PWM-Register bei Modul Pro II-CNT-x Rev. E (Seite 141).

TiCo-Prozessor

Das Modul besitzt einen frei programmierbaren *TiCo*-Prozessor (TiCo1) mit 28 KiByte Datenspeicher und 28 KiByte Programmspeicher. Den *TiCo*-Prozessor programmieren Sie in *TiCoBasic*.

Der *TiCo*-Prozessor hat – wie auch die *ADwin*-CPU – Zugriff auf alle Ein- und Ausgangskanäle, analog wie digital. Nähere Hinweise zur Anwendung und Programmierung des *TiCo*-Prozessors entnehmen Sie bitte dem Handbuch *TiCoBasic*.

Wenn Sie ein *TiCoBasic*-Programm im *TiCo*-Bootloader ablegen, wird das Programm beim Einschalten der Stromversorgung in den *TiCo*-Prozessor geladen und gestartet. Auf diese Weise kann das Modul eigenständig und unabhängig vom CPU-Modul des *ADwin-Pro II*-Systems arbeiten.

Technische Daten

Transistor-Ausgänge	
Ausgangskanäle	12
Schaltspannung V_{CC}	5…30 V DC durch externe Spannungsversorgung
Schaltstrom	200 mA max. pro Kanal
Spannungsabfall	0,5 V
Schaltzeit	2,5 µs
Isolation	42 V Kanal zu Kanal / Kanal zu Masse gemeinsame Masse für alle OPT- und TRA-Kanäle
Optokoppler-Eingänge	
Eingangskanäle	12
Eingangsstrom	typ. 3,5 mA / max. 7,5 mA
Eingangs-Spannungs-bereich (über Jumper wählbar)	0…5 V / 0…12 V / 0…24 V
Schaltschwelle für 0-low	0…0,8 V / 0…1,6 V / 0…3,2 V
Schaltschwelle für 1-high	4,5…5 V / 10…12 V / 20…24 V
Spannungsfestigkeit	-5 V … 8 V / -5 V … 16 V / -5 V … 30 V /
Schaltzeit	100 ns
Isolation	42 V Kanal zu Kanal / Kanal zu Masse gemeinsame Masse für alle OPT- und TRA-Kanäle
Zähler	
Anzahl	2 Universalzähler
Zählerbreite	32 Bit
Referenztakt	50 MHz
Taktfrequenz Vierflankenauswertung	12,5 MHz max. (bei 90° Phasenverschiebung der Signale)
Taktfrequenz Vor- / Rückwärtszähler	15 MHz max.
Referenzfrequenz PWM-Analyse	100 MHz

Abb. 32 – Pro II-MIO-D12 Rev. E: Spezifikation

Eingangspegel	1 Zähler RS422/485 kompatibel (5V differentiell,120 Ω Bus-Abschlusswiderstand) 1 Zähler über Optokoppler
Isolation	über Optokoppler: 42V Kanal zu Kanal / Kanal zu Masse
SSI-Decoder	
Anzahl	1
Taktfrequenz SSI-Decoder (CLK)	6,1kHz … 12,5MHz
Allgemein	
TiCo-Prozessor	Prozessortyp: TiCo1 Taktfrequenz: 50MHz Speichergröße: 28KiB PM intern, 28KiB DM intern
Eingangs- / Ausgangs-Fifo für TRA oder OPT	Größe: 511 Wertepaare Frequenz: 100MHz
Steckerverbindungen	37-polige D-Sub-Buchse

Abb. 32 – Pro II-MIO-D12 Rev. E: Spezifikation

Programmierung

Programmierung in ADbasic

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Transistor-Ausgänge	
Ausgangssignale setzen und rücklesen	`P2_MIO_Digout`, `P2_MIO_Digout_Long` `P2_MIO_Get_Digout_Long`
Latch-Register nutzen	`P2_MIO_Dig_Latch`, `P2_Sync_All` `P2_MIO_Dig_Write_Latch`

Ausgangssignale automatisch setzen	`P2_Dig_FIFO_Mode`
	`P2_Digout_FIFO_Clear`
	`P2_Digout_FIFO_Empty`
	`P2_Digout_FIFO_Enable`
	`P2_Digout_FIFO_Read_Timer`
	`P2_Digout_FIFO_Start`
	`P2_Digout_FIFO_Write`

Optokoppler-Eingänge	
Eingangssignale abfragen	`P2_Digin_Edge` `P2_MIO_Digin_Long`
Latch-Register nutzen	`P2_MIO_Dig_Latch`, `P2_Sync_All` `P2_MIO_Dig_Read_Latch`

Flanken an Eingangskanälen überwachen	`P2_Dig_FIFO_Mode`
	`P2_Digin_FIFO_Enable`
	`P2_Digin_FIFO_Read`
	`P2_Digin_FIFO_Read_Fast`
	`P2_Digin_FIFO_Read_Timer`
	`P2_Digin_FIFO_Clear`
	`P2_Digin_FIFO_Full`

Zählerblock	

Bereich	Befehle
Zähler konfigurieren	P2_Cnt_Enable, P2_Cnt_Mode
Zähler ansteuern	P2_Cnt_Clear, P2_Cnt_Get_Status P2_Cnt_Latch, P2_Cnt_Sync_Latch P2_Cnt_Read, P2_Cnt_Read4 P2_Cnt_Read_Latch P2_Cnt_Read_Latch4
PWM-Zähler ansteuern	P2_Cnt_PW_Enable, P2_Cnt_PW_Latch P2_Cnt_Get_PW, P2_Cnt_Get_PW_HL
SSI-Decoder	
SSI-Decoder ansteuern	P2_SSI_Mode, P2_SSI_Set_Bits P2_SSI_Set_Clock P2_SSI_Set_Delay, P2_SSI_Read P2_SSI_Start, P2_SSI_Status
Allgemein	
LEDs einstellen	P2_Check_LED, P2_Set_LED
Abläufe synchronisieren	P2_Sync_All, P2_Sync_Enable P2_Sync_Stat
Interrupts und Event-Eingang einstellen	P2_Event_Enable, P2_Event_Read P2_Event_Config

Programmierung in TiCoBasic

Das Modul kann mit *TiCoBasic*-Befehlen programmiert werden. Die Befehle sind in der Online-Hilfe *TiCoBasic* erläutert.

Die Include-Datei `MIO-D12_TiCo.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Transistor-Ausgänge	
Ausgangssignale setzen und rücklesen	MIO_Digout, MIO_Digout_Long MIO_Get_Digout_Long
Latch-Register nutzen	MIO_Dig_Latch MIO_Dig_Write_Latch
Optokoppler-Eingänge	
Eingangssignale abfragen	MIO_Digin_Long
Latch-Register nutzen	MIO_Dig_Latch MIO_Dig_Read_Latch
Zählerblock	
Zähler konfigurieren	Cnt_Enable, Cnt_Mode
Zähler ansteuern	Cnt_Clear, Cnt_Get_Status Cnt_Latch, Cnt_Read Cnt_Read_Int_Register Cnt_Sync_Latch, Cnt_Read_Latch
PWM-Zähler ansteuern	Cnt_PW_Enable, Cnt_PW_Latch Cnt_Get_PW_HL
SSI-Decoder	
SSI-Decoder ansteuern	SSI_Mode, SSI_Set_Bits SSI_Set_Clock SSI_Set_Delay, SSI_Read SSI_Start, SSI_Status
Allgemein	
LEDs einstellen	Check_LED, Set_LED

Bereich	Befehle
Interrupts und Event-Eingang einstellen	`Event_Enable`, `Event_Read` `Event_Config`, `Trigger_Event`

Programmierung TiCo-Zugriff

Für den Zugriff auf den *TiCo*-Prozessor von der ADwin CPU sind die folgenden *ADbasic*-Befehle in der Datei `ADwinPro_All.inc` definiert. Die Befehle sind im Handbuch *TiCoBasic* und in der Online-Hilfe *ADbasic* erläutert.

Bereich	Befehle

Datenaustausch mit dem *TiCo*-Prozessor über globale Variablen	`P2_TDrv_Init`
	`P2_GetData_Long`, `P2_Get_Par`, `P2_Get_Par_Block`
	`P2_SetData_Long`, `P2_Set_Par`, `P2_Set_Par_Block`
	`P2_Get_TiCo_RingBuffer`, `P2_Set_TiCo_RingBuffer`
	`P2_RingBuffer_Empty`
	`P2_RingBuffer_Full`

TiCo-Prozessor steuern	`P2_TiCo_Reset`, `P2_TiCo_Start`, `P2_TiCo_Stop`
	`P2_Get_TiCo_Bootloader_Status`
	`P2_Get_TiCo_Status`, `P2_Workload`

TiCo-Prozesse steuern	`P2_Process_Status`
	`P2_TiCo_Get_Processdelay`
	`P2_TiCo_Set_Processdelay`
	`P2_TiCo_Start_Process`
	`P2_Tico_Stop_Process`

TiCo-Programme übertragen	`P2_TiCo_Flash`, `P2_TiCo_Load`

5.5 Pro II: Analoge Eingangsmodule

Dieser Abschnitt beschreibt analoge Eingangsmodule für *ADwin-Pro II*.

Sie finden analoge Eingangsmodule für *ADwin-Pro I* im Handbuch „ADwin-Pro-Hardware ab Seite 19.

Modulname	AIn 8/18	AIn 8/18-TiCo	AIn 32/18-D	AIn 32/18-D-TiCo	AIn 16/18-C	AIn 8/18-8B	AIn 16/18-8B
Revision	E	E	E	E	E	E	E
Anzahl ADC	1	1	1	1	1	1	1
Auflösung [Bit]	18	18	18	18	18	18	18
max. Wandlungs-zeit [µs]	2	2	2	2	2	2	2
max. Abtastrate [ksample/s]	500	500	500	500	500	500	500
Kanäle diff.	8	8	16	16	16	8	–
Kanäle s.e.	–	–	32	32	–	–	–
Kanäle 8B	–	–	–	–	–	8	16
Messbereich	±10V				±20mA	±10V	
Verstärkung	1, 2, 4, 8						
Kalibrierung	per Software						
TiCo-Prozessor	–	TiCo1	–	TiCo1	–	–	–
Seite	54	54	58	58	62	65	68

Modulname	AIn F-4/14	AIn F-8/14	AIn F-4/16	AIn F-8/16	AIn F-4/18	AIn F-8/18
Revision	E	E	E	E	E	E
Anzahl ADC	4	8	4	8	4	8
Auflösung [Bit]	14	14	16	16	18	18
max. Wandlungs-zeit [µs]	0,02	0,02	0,25	0,25	2	2
max. Abtastrate [ksample/s]	50000	4×50000 8×25000	4000	4000	500	500
Kanäle diff.	4	8	4	8	4	8
Messbereich	±10V					
Verstärkung	1		1, 2, 4, 8		1	
Kalibrierung	per Software					
Seite	71	75	79	83	87	90

Hinweis zur Eingangsverschaltung

Offene Eingänge können zu Fehlern führen, vor allem in einer nicht störungsfreien Umgebung. Sie vermeiden offene Eingänge folgendermaßen:

- Trennen Sie nicht benutzte Eingänge von offenen Leitungen.
- Legen Sie nicht benutzte Eingänge auf einen definierten Pegel (z.B. GND). Der Anschluss sollte möglichst nah an Stecker oder Buchse des Moduls liegen.

5.5.1 Pro II-AIn-8/18 Rev. E

Analoges Eingangsmodul Pro II-AIn-8/18 Rev. E mit einem 18 Bit-ADC, 8 differentiellen Eingängen und einem programmierbaren Verstärker (PGA). Das Modul kann mit Ver1stärkern, Pro-TC und Pro-PT-Modulen kombiniert werden.

Das Modul kann in folgenden Varianten bestellt werden:

	ohne Filter	Filter 5 kHz	Filter 50 kHz	Filter 10 kHz, Bereich ±30 V
LEMO 1-polig	AIn-8/18	AIn-8/18-LP5	–	AIn-8/18-LP-30V
LEMO 1-polig, TiCo-Prozessor	–	–	AIn-8/18-LP50-TiCo	AIn-8/18-LP-30V-TiCo
D-Sub	AIn-8/18-D	AIn-8/18-LP5-D	–	AIn-8/18-LP-30V-D
D-Sub, TiCo-Prozessor	–	–	AIn-8/18-LP50-D-TiCo	AIn-8/18-LP-30V-D-TiCo

TiCo-Prozessor

Die Varianten Pro II-AIn-8/18-xxx-TiCo besitzen zusätzlich einen frei programmierbaren *TiCo*-Prozessor mit 28 KiByte Datenspeicher und 28 KiByte Programmspeicher, der Zugriff auf alle Eingänge des Moduls hat. Nähere Hinweise zur Anwendung und Programmierung des *TiCo*-Prozessors entnehmen Sie bitte dem Handbuch *TiCoBasic*.

Wenn Sie ein *TiCoBasic*-Programm im *TiCo*-Bootloader ablegen, wird das Programm beim Einschalten der Stromversorgung in den *TiCo*-Prozessor geladen und gestartet. Auf diese Weise kann das Modul eigenständig und unabhängig vom CPU-Modul des *ADwin-Pro II*-Systems arbeiten.

Tiefpass-Filter

Die Varianten Pro II-AIn-8/18-LPxxx enthalten einen Eingangsfilter, einen Tiefpass 4. Ordnung vom Typ Butterworth. Je nach Variante ist die Eckfrequenz auf 5 kHz, 10 kHz oder 50 kHz festgelegt.

Die Eingänge verfügen über geschirmte LEMO-Buchsen (1-polig, CAMAC Europanorm, siehe Abb. 36) oder eine 37-polige D-Sub-Buchse.

Das Modul Pro II-AIn-8/18 Rev. E hat einen Eingangs-Spannungsbereich von ±10V (Variante LP-30V: ±30V) und per Software programmierbare Verstärkung von 1, 2, 4 oder 8. Der Abgleich der Verstärkung und des Offsets erfolgt per Software (siehe Kapitel 6 “Kalibrierung“).

Das Modul beinhaltet zusätzlich eine Ablaufsteuerung, die Messwerte an mehreren oder allen Eingangskanälen nacheinander einlesen kann.

Das Modul kann für jeden Kanal getrennt das Übertreten eines oberen und eines unteren Grenzwerts überwachen.

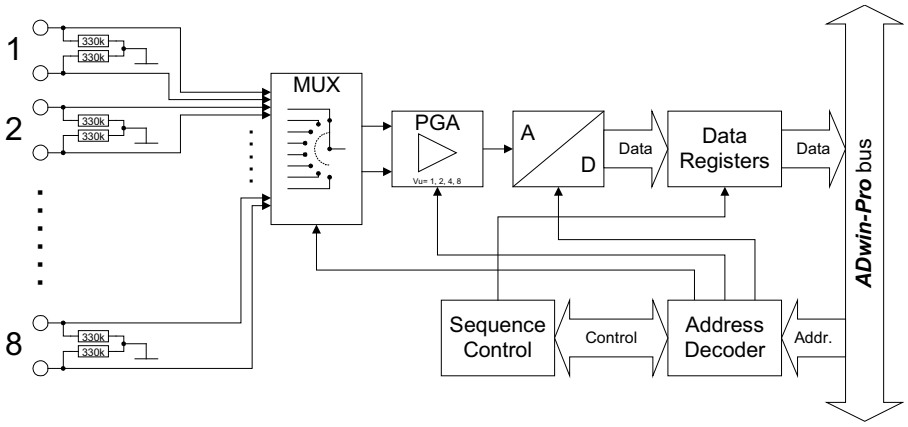
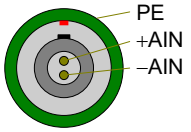


Abb. 33 – Pro II-AIn-8/18 Rev. E: Blockschaltbild

Eingangskanäle		8 differentiell über Multiplexer
Auflösung		18 Bit
Wandlungszeit		max. 2µs
Abtastrate		max. 500ksps
Multiplexer Einschwingzeit		2,5µs
Messbereich		±10V oder ±30V
Verstärkung		1, 2, 4, 8 per Software einstellbar
Eingangsfilter		je nach Variante: ohne Filter, 5kHz, 10kHz oder 30kHz
Genauigkeit	INL	typisch ±4 LSB
	DNL	max. ±1 LSB
Eingangswiderstand		330kΩ, ±2%
Spannungsfestigkeit		±35V
Offsetfehler		abgleichbar
Offsetdrift		±30ppm/ºC
TiCo-Prozessor, je nach Variante		Prozessortyp: TiCo1 Taktfrequenz: 50MHz Speichergröße: 28KiB PM intern, 28KiB DM intern
Steckverbindung		8 LEMO-Buchsen, 1-polig oder D-Sub-Buchse, 37-polig

Abb. 34 – Pro II-AIn-8/18 Rev. E: Spezifikation



Detailansicht 2-polige LEMO-Buchse; Serie 00 Multikontakt, Bauform EGG

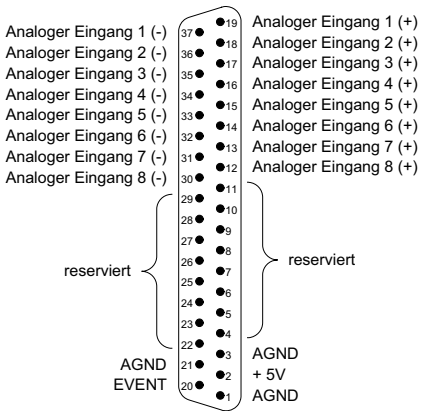


Fig. 35 – Pro II-AIn-8/18 Rev. E: Pinbelegung

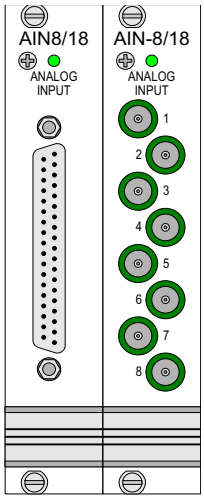


Abb. 36 – Pro II-AIn-8/18 Rev. E: Frontplatten

Programmierung in ADbasic

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Im Verzeichnis `C:\ADwin\ADbasic\samples_ADwin_ProII` finden Sie zussäzliche Beispielprogramme.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Einzelmessung durchführen – vollständig oder schrittweise	**P2_ADC**, **P2_ADC24** **P2_Set_Mux**, **P2_Start_Conv** **P2_Wait_EOC** **P2_Read_ADC**, **P2_Read_ADC24**
Messwert lesen und neue Messung starten	**P2_Read_ADC_SConv** **P2_Read_ADC_SConv24**
Ablaufsteuerung anwenden	**P2_Seq_Init**, **P2_Seq_Start** **P2_Seq_Read**, **P2_Seq_Read24** **P2_Seq_Read24_Packed** **P2_Seq_Wait**
Grenzwerte überwachen	**P2_ADC_Read_Limit** **P2_ADC_Set_Limit**
Wandlung synchronisieren	**P2_Sync_All**
LEDs einstellen	**P2_Check_LED**, **P2_Set_LED**
Interrupts und Event-Eingang einstellen	**P2_Event_Enable**, **P2_Event_Read** **P2_Event_Config**, **P2_Event2_Config**

Programmierung in TiCoBasic

Das Modul kann mit *TiCoBasic*-Befehlen programmiert werden. Die Befehle sind in der Online-Hilfe *TiCoBasic* erläutert.

Die Include-Datei `AInTiCo.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Einzelmessung durchführen – vollständig oder schrittweise	**ADC**, **ADC24** **Set_Mux**, **Start_Conv**, **Wait_EOC** **Read_ADC**, **Read_ADC24**
Messwert lesen und neue Messung starten	**Read_ADC_SConv** **Read_ADC_SConv24**
LEDs einstellen	**Check_LED**, **Set_LED**

Bereich	Befehle
Interrupts und Event-Eingang einstellen	`Event_Enable`, `Event_Read` `Event_Config`, `Trigger_Event`

Programmierung TiCo-Zugriff

Für den Zugriff auf den *TiCo*-Prozessor von der ADwin CPU sind die folgenden *ADbasic*-Befehle in der Datei `ADwinPro_All.inc` definiert. Die Befehle sind im Handbuch *TiCoBasic* und in der Online-Hilfe *ADbasic* erläutert.

Bereich	Befehle

Datenaustausch mit dem *TiCo*-Prozessor über globale Variablen	`P2_TDrv_Init`
	`P2_GetData_Long`, `P2_Get_Par`, `P2_Get_Par_Block`
	`P2_SetData_Long`, `P2_Set_Par`, `P2_Set_Par_Block`
	`P2_Get_TiCo_RingBuffer`, `P2_Set_TiCo_RingBuffer`
	`P2_RingBuffer_Empty`
	`P2_RingBuffer_Full`

TiCo-Prozessor steuern	`P2_TiCo_Reset`, `P2_TiCo_Start`, `P2_TiCo_Stop`
	`P2_Get_TiCo_Bootloader_Status`
	`P2_Get_TiCo_Status`, `P2_Workload`

TiCo-Prozesse steuern	`P2_Process_Status`
	`P2_TiCo_Get_Processdelay`
	`P2_TiCo_Set_Processdelay`
	`P2_TiCo_Start_Process`
	`P2_Tico_Stop_Process`

TiCo-Programme übertragen	`P2_TiCo_Flash`, `P2_TiCo_Load`

5.5.2 Pro II-AIn-32/18-D Rev. E

Analoges Eingangsmodul Pro II-AIn-32/18-D Rev. E mit einem 18-Bit ADC, 32 Eingängen und einem programmierbaren Verstärker (PGA). Das Modul kann mit Verstärkern, Pro-TC und Pro-PT-Modulen kombiniert werden.

Es gibt eine Modulvariante mit Stromeingängen Pro II-AIn-16/18-C Rev. E (Seite 62).

Die Variante Pro II-AIn-32/18-D-TiCo Rev. E besitzt zusätzlich einen frei programmierbaren *TiCo*-Prozessor mit 28 KiByte Datenspeicher und 28 KiByte Programmspeicher, der Zugriff auf alle Eingänge des Moduls hat. Nähere Hinweise zur Anwendung und Programmierung des *TiCo*-Prozessors entnehmen Sie bitte dem Handbuch *TiCoBasic*.

Wenn Sie ein *TiCoBasic*-Programm im *TiCo*-Bootloader ablegen, wird das Programm beim Einschalten der Stromversorgung in den *TiCo*-Prozessor geladen und gestartet. Auf diese Weise kann das Modul eigenständig und unabhängig vom CPU-Modul des *ADwin-Pro II*-Systems arbeiten.

Das Modul hat 32 single ended-Eingänge oder 16 differentielle Eingänge (über Software wählbar). Die Eingänge sind auf eine 37-polige D-Sub-Buchse geführt; Pinbelegung siehe Abb. 39 und 40.

Nach dem Einschalten ist das Modul auf 16 differentielle Eingänge eingestellt.

Das Modul Pro II-AIn-32/18-D Rev. E besitzt einen Eingangs-Spannungsbereich von ±10V und eine programmierbare Verstärkung von 1, 2, 4 oder 8. Der Abgleich der Verstärkung und des Offsets erfolgt per Software (siehe Kapitel 6 “Kalibrierung“).

Ab Werk sind die Eingänge mit der Masse des Pro-Geräts verbunden. Alternativ kann ein Massepotenzial an einem der AGND-Pins angeschlossen werden. Die Masseverbindung zum Pro-Gerät sollte dann aufgetrennt werden, indem der DIP-Schalter auf die Position `GND LIFT` gestellt wird (Abb. 41).

Betreiben Sie das Modul in keinem Fall ohne Masseverbindung.

Das Modul beinhaltet zusätzlich eine Ablaufsteuerung, die Messwerte an mehreren oder allen Eingangskanälen nacheinander einlesen kann.

Das Modul kann für jeden Kanal getrennt das Übertreten eines oberen und eines unteren Grenzwerts überwachen.

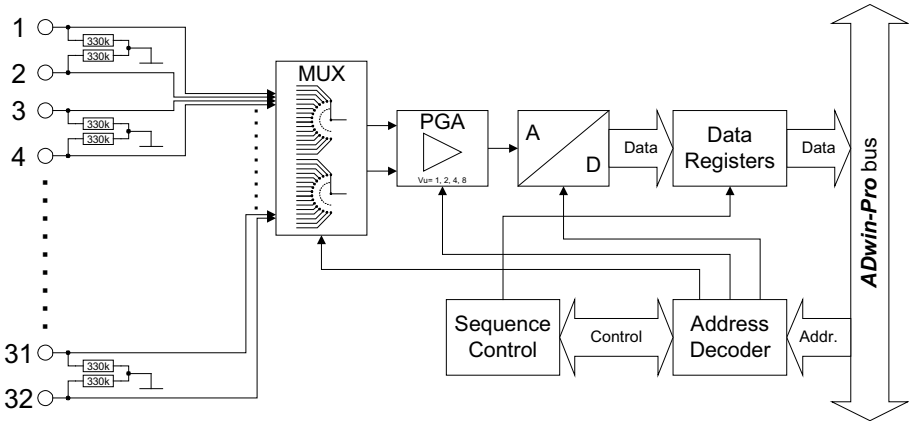


Abb. 37 – Pro II-AIn-32/18-D Rev. E: Blockschaltbild

Eingangskanäle:	32 single ended / 16 differentiell, Multiplexer
Auflösung:	18 Bit
Wandlungszeit:	max. 2 µs

Abb. 38 – Pro II-AIn-32/18-D Rev. E: Spezifikation

Abtastrate:		max. 500 ksps
Multiplexer Einschwingzeit:		2,5 µs
Messbereich:		±10 V
Verstärkung:		1, 2, 4, 8 per Software einstellbar
Genauigkeit	INL	typisch ±4 LSB
	DNL	max. ±1 LSB
Eingangswiderstand:		330 kΩ, ±2%
Spannungsfestigkeit:		±35 V
Offsetfehler:		abgleichbar
Offsetdrift:		±30 ppm/°C
Steckverbindung:		37-polige D-Sub-Buchse
TiCo-Prozessor, nur Variante Pro II-AIn-32/18-D-TiCo Rev. E		Prozessortyp: TiCo1 Taktfrequenz: 50 MHz Speichergröße: 28 KiB PM intern, 28 KiB DM intern

Abb. 38 – Pro II-AIn-32/18-D Rev. E: Spezifikation

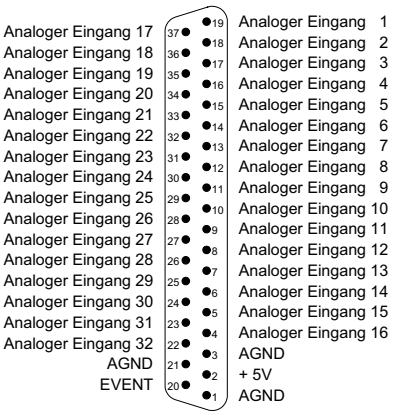


Abb. 39 – Pro II-AIn-32/18-D Rev. E: Pinbelegung single ended

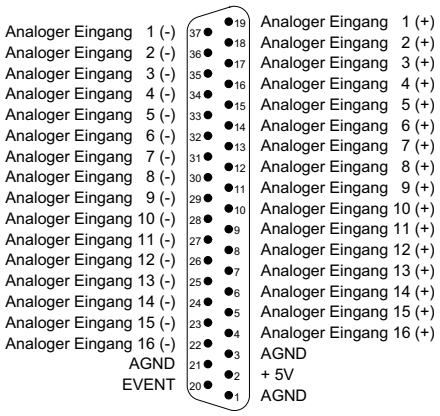


Abb. 40 – Pro II-AIn-32/18-D Rev. E: Pinbelegung differentiell

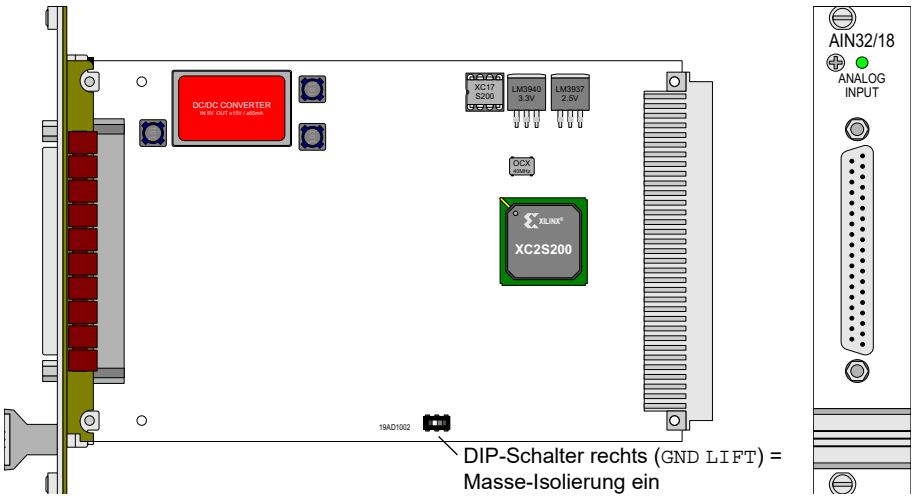


Abb. 41 – Pro II-AIn-32/18-D Rev. E: Platine und Frontplatte

Programmierung in ADbasic

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Im Verzeichnis `C:\ADwin\ADbasic\samples_ADwin_ProII` finden Sie zusätzliche Beispielprogramme (`ProII-AIn-8-18xxx.bas`).

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Eingänge s.e. oder differentiell	`P2_SE_Diff`

Einzelmessung durchführen – vollständig oder schrittweise	`P2_ADC`, `P2_ADC24`
	`P2_Set_Mux`, `P2_Start_Conv`
	`P2_SE_Diff`, `P2_Wait_EOC`
	`P2_Read_ADC`, `P2_Read_ADC24`

Messwert lesen und neue Messung starten	`P2_Read_ADC_SConv` `P2_Read_ADC_SConv24`

Ablaufsteuerung anwenden	`P2_Seq_Init`, `P2_Seq_Start`
	`P2_Seq_Read`, `P2_Seq_Read24`
	`P2_Seq_Read24_Packed`
	`P2_Seq_Wait`

Grenzwerte überwachen	`P2_ADC_Read_Limit` `P2_ADC_Set_Limit`
Wandlung synchronisieren	`P2_Sync_All`
LEDs einstellen	`P2_Check_LED`, `P2_Set_LED`
Interrupts und Event-Eingang einstellen	`P2_Event_Enable`, `P2_Event_Read` `P2_Event_Config`, `P2_Event2_Config`

Programmierung in TiCoBasic

Das Modul kann mit *TiCoBasic*-Befehlen programmiert werden. Die Befehle sind in der Online-Hilfe *TiCoBasic* erläutert.

Die Include-Datei `AInTiCo.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle

Einzelmessung durchführen – vollständig oder schrittweise	`SE_Diff`
	`ADC`, `ADC24`
	`Set_Mux`, `Start_Conv`
	`Wait_EOC`
	`Read_ADC`, `Read_ADC24`

Messwert lesen und neue Messung starten	`Read_ADC_SConv` `Read_ADC_SConv24`
LEDs einstellen	`Check_LED`, `Set_LED`
Interrupts und Event-Eingang einstellen	`Event_Enable`, `Event_Read` `Event_Config`, `Trigger_Event`

Programmierung TiCo-Zugriff

Für den Zugriff auf den *TiCo*-Prozessor von der ADwin CPU sind die folgenden *ADbasic*-Befehle in der Datei `ADwinPro_All.inc` definiert. Die Befehle sind im Handbuch *TiCoBasic* und in der Online-Hilfe *ADbasic* erläutert.

Bereich	Befehle

Datenaustausch mit dem *TiCo*-Prozessor über globale Variablen	`P2_TDrv_Init`
	`P2_GetData_Long`, `P2_Get_Par`, `P2_Get_Par_Block`
	`P2_SetData_Long`, `P2_Set_Par`, `P2_Set_Par_Block`
	`P2_Get_TiCo_RingBuffer`, `P2_Set_TiCo_RingBuffer`
	`P2_RingBuffer_Empty`
	`P2_RingBuffer_Full`

Bereich	Befehle
TiCo-Prozessor steuern	P2_TiCo_Reset, P2_TiCo_Start, P2_TiCo_Stop P2_Get_TiCo_Bootloader_Status P2_Get_TiCo_Status, P2_Workload
TiCo-Prozesse steuern	P2_Process_Status P2_TiCo_Get_Processdelay P2_TiCo_Set_Processdelay P2_TiCo_Start_Process P2_Tico_Stop_Process
TiCo-Programme übertragen	P2_TiCo_Flash, P2_TiCo_Load

5.5.3 Pro II-AIn-16/18-C Rev. E

Analoges Eingangsmodul Pro II-AIn-16/18-C Rev. E mit einem 18-Bit ADC, 16 Eingängen und einem programmierbaren Verstärker (PGA). Das Modul kann mit Verstärkern, Pro-TC und Pro-PT-Modulen kombiniert werden.

Das Modul hat 16 differentielle Strom-Eingänge. Die Eingänge sind auf eine 37-polige D-Sub-Buchse geführt; Pinbelegung siehe Abb. 44.

Es gibt eine Modulvariante mit Spannungseingängen Pro II-AIn-32/18-D Rev. E (Seite 58).

Das Modul Pro II-AIn-16/18-C Rev. E besitzt einen Eingangsbereich von ±20mA und eine programmierbare Verstärkung von 1, 2, 4 oder 8. Der Abgleich der Verstärkung und des Offsets erfolgt per Software (siehe Kapitel 6 “Kalibrierung“).

Ab Werk sind die Eingänge mit der Masse des Pro-Geräts verbunden. Alternativ kann ein Massepotenzial an einem der AGND-Pins angeschlossen werden. Die Masseverbindung zum Pro-Gerät sollte dann aufgetrennt werden, indem der DIP-Schalter auf die Position `GND LIFT` gestellt wird (Abb. 45).

Betreiben Sie das Modul in keinem Fall ohne Masseverbindung.

Das Modul beinhaltet zusätzlich eine Ablaufsteuerung, die Messwerte an mehreren oder allen Eingangskanälen nacheinander einlesen kann.

Das Modul kann für jeden Kanal getrennt das Übertreten eines oberen und eines unteren Grenzwerts überwachen.

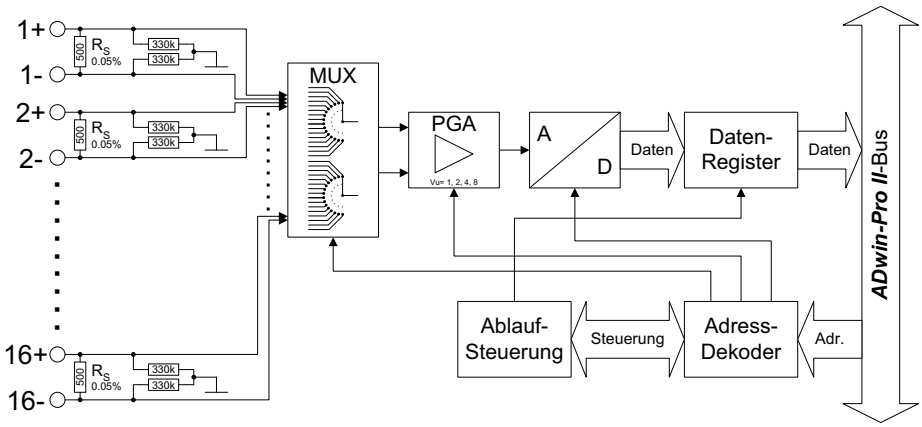


Abb. 42 – Pro II-AIn-16/18-C Rev. E: Blockschaltbild

Eingangskanäle:		16 differentiell, Multiplexer
Auflösung:		18 Bit
Wandlungszeit:		max. 2µs
Abtastrate:		max. 500ksps
Multiplexer Einschwingzeit:		2,5µs
Messbereich:		±20mA
Verstärkung:		1, 2, 4, 8 per Software einstellbar
Genauigkeit	INL	typisch ±4 LSB ±0,05% der gemessenen Spannung durch den Eingangswiderstand
	DNL	max. ±1 LSB ±0,05% der gemessenen Spannung durch den Eingangswiderstand

Abb. 43 – Pro II-AIn-16/18-C Rev. E: Spezifikation

Eingangswiderstand:	500 Ω, ±0,05%
Spannungsfestigkeit:	±15V
Offsetfehler:	abgleichbar
Offsetdrift:	±30 ppm/°C
Steckverbindung:	37-polige D-Sub-Buchse

Abb. 43 – Pro II-AIn-16/18-C Rev. E: Spezifikation

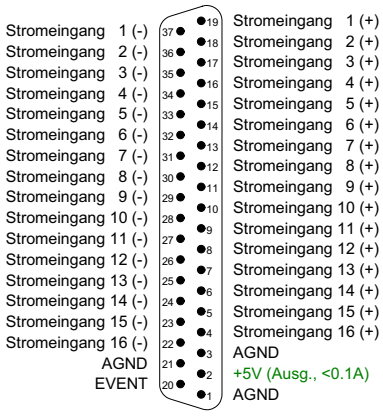


Abb. 44 – Pro II-AIn-16/18-C Rev. E: Pinbelegung

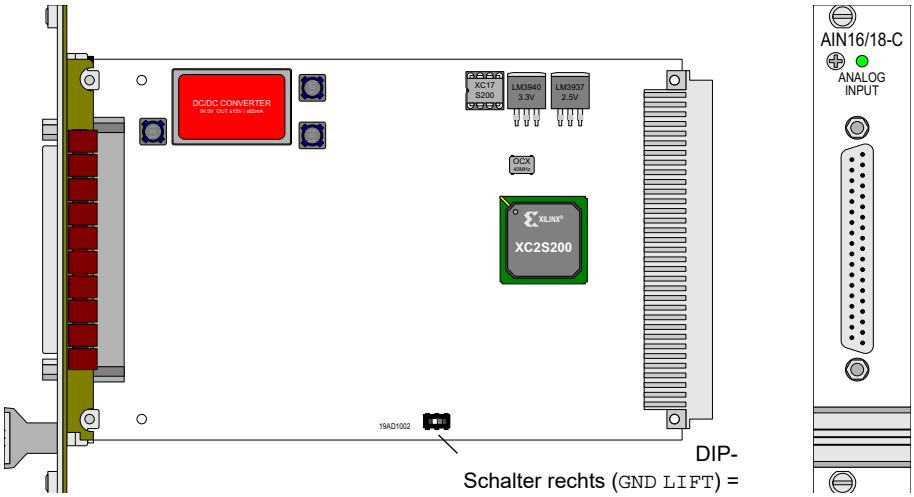


Abb. 45 – Pro II-AIn-16/18-C Rev. E: Platine und Frontplatte

Programmierung in ADbasic

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Im Verzeichnis `C:\ADwin\ADbasic\samples_ADwin_ProII` finden Sie zusätzliche Beispielprogramme (`ProII-AIn-8-18xxx.bas`).

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Eingänge s.e. oder differentiell	`P2_SE_Diff`

Einzelmessung durchführen – vollständig oder schrittweise	`P2_ADC`, `P2_ADC24`
	`P2_Set_Mux`, `P2_Start_Conv`
	`P2_SE_Diff`, `P2_Wait_EOC`
	`P2_Read_ADC`, `P2_Read_ADC24`

Messwert lesen und neue Messung starten	`P2_Read_ADC_SConv` `P2_Read_ADC_SConv24`

Bereich	Befehle
Ablaufsteuerung anwenden	P2_Seq_Init, P2_Seq_Start P2_Seq_Read, P2_Seq_Read24 P2_Seq_Read24_Packed P2_Seq_Wait
Grenzwerte überwachen	P2_ADC_Read_Limit P2_ADC_Set_Limit
Wandlung synchronisieren	P2_Sync_All
LEDs einstellen	P2_Check_LED, P2_Set_LED
Interrupts und Event-Eingang einstellen	P2_Event_Enable, P2_Event_Read P2_Event_Config, P2_Event2_Config

5.5.4 Pro II-AIn-8/18-8B Rev. E

Analoges Eingangsmodul Pro II-AIn-8/18-8B Rev. E mit einem 18-Bit ADC, 16 analogen Eingängen und einem programmierbaren Verstärker (PGA). Das Modul basiert auf Pro II-AIn-32/18-D Rev. E und hat eine Zusatzplatine mit 8 Steckplätzen für 8B-Module.

Von den 16 Moduleingängen sind 8 Eingänge für 8B-Module und 8 differentielle Eingänge. Die Eingänge stehen auf zwei 37-poligen D-Sub-Buchsen zur Verfügung; Pinbelegung siehe Abb. 47.

Der Eingangs-Spannungsbereich nach den 8B-Modulen beträgt ±10V. Die Verstärkung ist programmierbar als 1, 2, 4 oder 8. Der Abgleich der Verstärkung und des Offsets erfolgt per Software (siehe Kapitel 6 “Kalibrierung“).

Ab Werk sind die Eingänge mit der Masse des Pro-Gehäuses verbunden. Alternativ kann an den differentiellen Eingängen ein Massepotenzial an einem der AGND-Pins angeschlossen werden. Die Masseverbindung zum Pro-Gehäuse sollte dann aufgetrennt werden, indem der DIP-Schalter auf die Position `GND LIFT` gestellt wird (siehe Abb. 41, Pro II-AIn-32/18-D Rev. E).

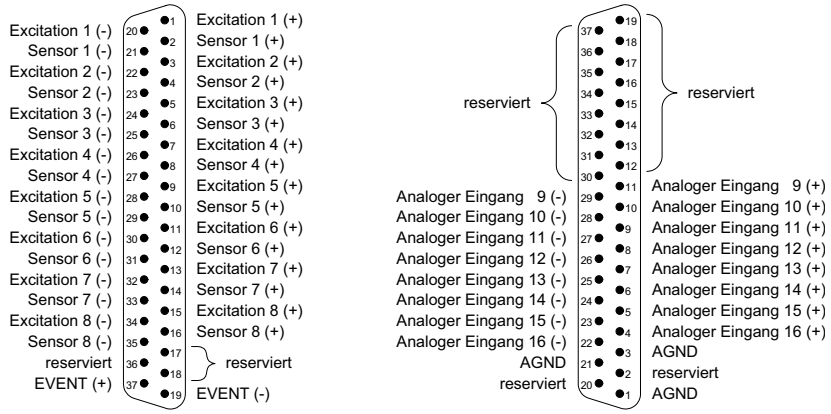
Betreiben Sie das Modul in keinem Fall ohne Masseverbindung.

Das Modul beinhaltet zusätzlich eine Ablaufsteuerung, die Messwerte an mehreren oder allen Eingangskanälen nacheinander einlesen kann.

Das Modul kann für jeden Kanal getrennt das Übertreten eines oberen und eines unteren Grenzwerts überwachen.

Eingangskanäle:		16 über Multiplexer: 8 für MB8-Module, 8 differentiell
Auflösung:		18 Bit
Wandlungszeit:		max. 2µs
Abtastrate:		max. 500ksps
Multiplexer Einschwingzeit:		2,5µs
Messbereich:		±10V
Verstärkung:		1, 2, 4, 8 per Software einstellbar
Genauigkeit	INL	typisch ±4 LSB
	DNL	max. ±1 LSB
Eingangswiderstand:		330kΩ, ±2%
Spannungsfestigkeit:		±35V
Offsetfehler:		abgleichbar
Offsetdrift:		±30ppm/°C
Steckverbindung:		37-polige D-Sub-Buchse
Modulbreite:		15 TE

Abb. 46 – Pro II-AIn-8/18-8B Rev. E: Spezifikation



Eingänge 8B 1…8 Eingänge AIn 9…16

Abb. 47 – Pro II-AIn-8/18-8B Rev. E: Pinbelegungen

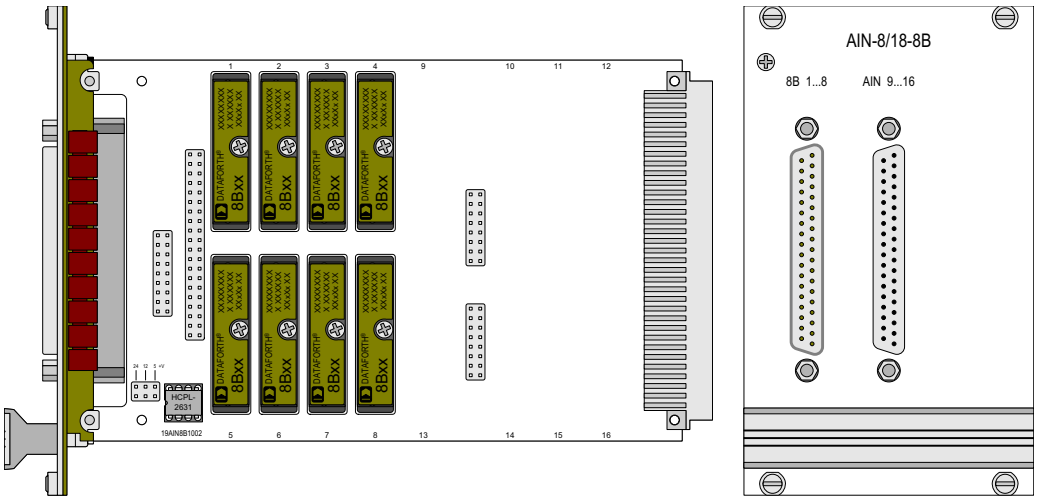


Abb. 48 – Pro II-AIn-8/18-8B Rev. E: Platine und Frontplatte

Programmierung

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Im Verzeichnis `C:\ADwin\ADbasic\samples_ADwin_ProII` finden Sie zusätzliche Beispielprogramme (`ProII-AIn-8-18xxx.bas`).

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Eingänge s.e. oder differentiell	`P2_SE_Diff`

Einzelmessung durchführen – vollständig oder schrittweise	`P2_ADC`, `P2_ADC24`, `P2_Wait_EOC`
	`P2_Set_Mux`, `P2_Start_Conv`
	`P2_Read_ADC`, `P2_Read_ADC24`

Messwert lesen und neue Messung starten	`P2_Read_ADC_SConv` `P2_Read_ADC_SConv24`

Ablaufsteuerung anwenden	`P2_Seq_Init`, `P2_Seq_Start`
	`P2_Seq_Read`, `P2_Seq_Read24`
	`P2_Seq_Read24_Packed`, `P2_Seq_Wait`

Grenzwerte überwachen	`P2_ADC_Read_Limit` `P2_ADC_Set_Limit`
Wandlung synchronisieren	`P2_Sync_All`
LEDs einstellen	`P2_Check_LED`, `P2_Set_LED`

Bereich	Befehle
Interrupts und Event-Eingang einstellen	`P2_Event_Enable`, `P2_Event_Read` `P2_Event_Config`, `P2_Event2_Config`

5.5.5 Pro II-AIn-16/18-8B Rev. E

Analoges Eingangsmodul Pro II-AIn-16/18-8B Rev. E mit einem 18-Bit ADC, 16 analogen Eingängen und einem programmierbaren Verstärker (PGA). Das Modul basiert auf Pro II-AIn-32/18-D Rev. E und hat eine Zusatzplatine mit 16 Steckplätzen für 8B-Module.

Das Modul hat 16 Eingänge (über 8B-Module) auf zwei 37-poligen D-Sub-Buchsen; Pinbelegung siehe Abb. 50.

Der Eingangs-Spannungsbereich nach den 8B-Modulen beträgt ±10V. Die Verstärkung ist programmierbar als 1, 2, 4 oder 8. Der Abgleich der Verstärkung und des Offsets erfolgt per Software (siehe Kapitel 6 “Kalibrierung“).

Ab Werk sind die Eingänge mit der Masse des Pro-Gehäuses verbunden. Alternativ kann ein Massepotenzial an einem der AGND-Pins angeschlossen werden. Die Masseverbindung zum Pro-Gehäuse sollte dann aufgetrennt werden, indem der DIP-Schalter auf die Position `GND LIFT` gestellt wird (siehe Abb. 41, Pro II-AIn-32/18-D Rev. E).

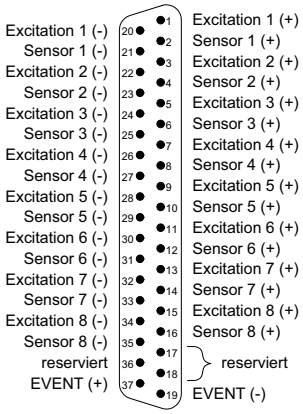
Betreiben Sie das Modul in keinem Fall ohne Masseverbindung.

Das Modul beinhaltet zusätzlich eine Ablaufsteuerung, die Messwerte an mehreren oder allen Eingangskanälen nacheinander einlesen kann.

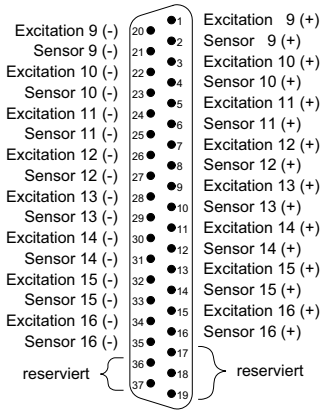
Das Modul kann für jeden Kanal getrennt das Übertreten eines oberen und eines unteren Grenzwerts überwachen.

Eingangskanäle:		16 über Multiplexer
Auflösung:		18 Bit
Wandlungszeit:		max. 2µs
Abtastrate:		max. 500ksps
Multiplexer Einschwingzeit:		2,5µs
Messbereich:		±10V
Verstärkung:		1, 2, 4, 8 per Software einstellbar
Genauigkeit	INL	typisch ±4 LSB
	DNL	max. ±1 LSB
Eingangswiderstand:		330kΩ, ±2%
Spannungsfestigkeit:		±35V
Offsetfehler:		abgleichbar
Offsetdrift:		±30ppm/ºC
Steckverbindung:		37-polige D-Sub-Buchse
Modulbreite		10 TE

Abb. 49 – Pro II-AIn-16/18-8B Rev. E: Spezifikation



Eingänge 8B 1…8



Eingänge 8B 9…16

Abb. 50 – Pro II-AIn-16/18-8B Rev. E: Pinbelegungen

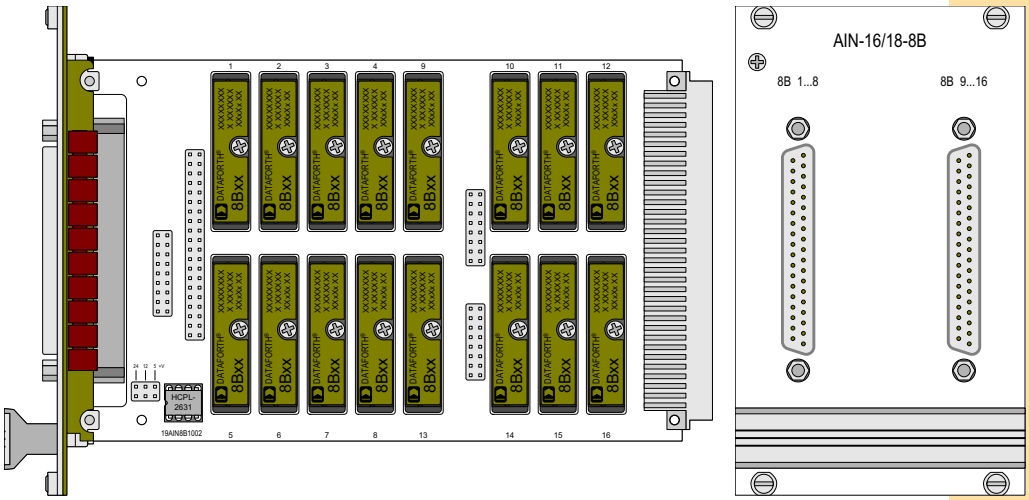


Abb. 51 – Pro II-AIn-16/18-8B Rev. E: Platine und Frontplatte

Programmierung

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Im Verzeichnis `C:\ADwin\ADbasic\samples_ADwin_ProII` finden Sie zusätzliche Beispielprogramme (`ProII-AIn-8-18xxx.bas`).

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Eingänge s.e. oder differentiell	`P2_SE_Diff`

Einzelmessung durchführen – vollständig oder schrittweise	`P2_ADC`, `P2_ADC24`, `P2_Wait_EOC`
	`P2_Set_Mux`, `P2_Start_Conv`
	`P2_Read_ADC`, `P2_Read_ADC24`

Messwert lesen und neue Messung starten	`P2_Read_ADC_SConv` `P2_Read_ADC_SConv24`

Ablaufsteuerung anwenden	`P2_Seq_Init`, `P2_Seq_Start`
	`P2_Seq_Read`, `P2_Seq_Read24`
	`P2_Seq_Read24_Packed`, `P2_Seq_Wait`

Grenzwerte überwachen	`P2_ADC_Read_Limit` `P2_ADC_Set_Limit`
Wandlung synchronisieren	`P2_Sync_All`
LEDs einstellen	`P2_Check_LED`, `P2_Set_LED`

Bereich	Befehle
Interrupts und Event-Eingang einstellen	`P2_Event_Enable`, `P2_Event_Read` `P2_Event_Config`, `P2_Event2_Config`

5.5.6 Pro II-AIn-F-4/14 Rev. E

Analoges Eingangsmodul Pro II-AIn-F-4/14 Rev. E mit 4 Fast-ADC zu 14 Bit und 4 differentiellen Eingängen.

Für die Eingänge sind folgende Steckverbindungen verfügbar:

– Pro II-AIn-F-4/14: geschirmte LEMO-Buchsen 1-polig, CAMAC Europanorm.

– Pro II-AIn-F-4/14-L2: geschirmte LEMO-Buchsen 2-polig, CAMAC Europanorm.

– Pro II-AIn-F-4/14-D: D-Sub-Buchse 37-polig.

– Pro II-AIn-F-4/14-B: BNC-Buchsen.

Die Wandler des Moduls arbeiten eigenständig mit einer festen Abtastrate von 50 MHz; im Programm werden nur noch die zuletzt gewandelten Messwerte abgeholt. Der moduleigene Speicher erlaubt, die große Menge anfallender Daten bei Burst-Messungen zwischenzuspeichern.

Das Modul Pro II-AIn-F-4/14 Rev. E hat einen Eingangs-Spannungsbereich von bipolar ±10V. Der Abgleich der Verstärkung und des Offsets erfolgt per Software (siehe Kapitel 6 “Kalibrierung“).

Das Modul verfügt über mehrere Betriebsarten für die Wandlung von analogen Signalen:

– Einzelmessung: Jede Messung wird im *ADbasic*-Programm einzeln abgefragt und weiter verarbeitet; die Wandlung führt das Modul selbstständig durch.

– Einfache Burst-Messreihe: Das *ADbasic*-Programm startet eine vollständige Messreihe, also eine definierte Anzahl von Einzelmessungen.

– Kontinuierliche Burst-Messreihe: Das *ADbasic*-Programm startet eine Messreihe, die kontinuierlich Einzelmessungen ausführt, bis die Messreihe gestoppt wird. Die Daten werden in einem Ringspeicher abgelegt.

Das Modul kann optional anstelle von Einzelmessungen jeweils den Mittelwert aus 2…32 Messwerten zurückgeben. Für jeden Mittelwert wandelt das Modul separat die eingestellte Anzahl an Messwerten.

Das Modul kann für jeden Kanal getrennt das Übertreten eines oberen und eines unteren Grenzwerts überwachen.

Burst-Messreihen

Das Modul führt Burst-Messreihen unabhängig vom Prozessormodul des *ADwin*-Systems aus. Die Messwerte – Anzahl und Messfrequenz sind vorab im Programm zu definieren – werden im Burst-Speicher auf dem Modul abgelegt. Das Prozessormodul liest nur noch die gespeicherten Messwerte (auch während der laufenden Messreihe) und verarbeitet sie.

Bei einer kontinuierlichen Burst-Messreihe muss die Messfrequenz mit der Auslesegeschwindigkeit abgestimmt werden. Hierbei sind verschiedene Aspekte von Bedeutung:

– Das Auslesen der Messwerte geschieht in Blöcken. Je größer der Datenblock ist, um so schneller ist der Lesevorgang im Mittel.

  Beachten Sie: Das blockweise Lesen kann andere, auch hochpriore Prozesse verzögern. Je größer der gelesene Block ist, umso eher kann eine Verzögerung entstehen.

– Der Zeitversatz zwischen kontinuierlicher Wandlung und blockweisem Lesen erfordert einen Datenpuffer. Hierzu muss bei der Initialisierung der Burst-Messung ein genügend großer Speicherbereich reserviert werden (`P2_Burst_Init`, Parameter `samples`).

Event-Eingänge

Bei der Modulversion Pro II-AIn-F-4/14-D Rev. E (mit D-Sub-Buchse) kann eine Burst-Messreihe über externe Event-Signale gesteuert werden, so dass für jedes (resultierende) Event-Signal ein Messwert gespeichert wird.

Optional kann ein Kanal der Burst-Messreihe als Zeitkanal genutzt werden, in dem bei jedem Event-Signal der Zählerstand des moduleigenen Timers gespeichert wird.

Das Modul hat 3 differentielle Event-Eingänge: `EVENT/A`, `B`, `ENABLE`, deren Signale das Modul zum resultierenden Event-Signal verarbeitet. Diese Vorverarbeitung der Signale ist konfigurierbar mit **`P2_Event2_Config`**.

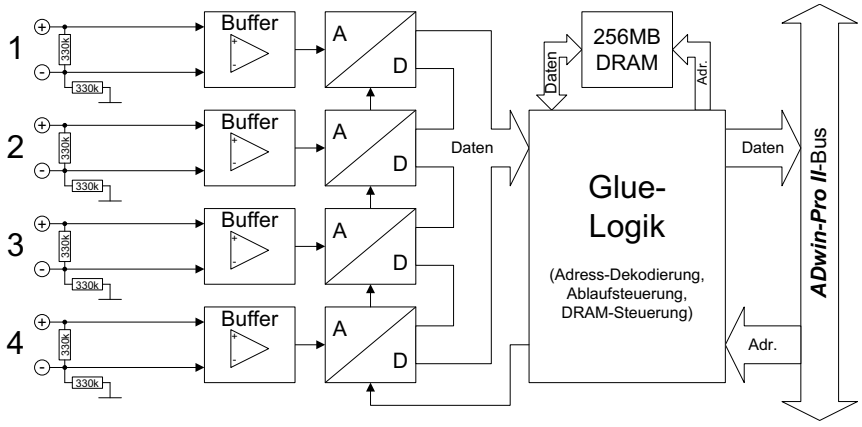


Abb. 52 – Pro II-AIn-F-4/14 Rev. E: Blockschaltbild

Eingangskanäle		4 differentiell
Auflösung		14 Bit
Konvertierungszeit		0,02 µs (je ADC)
Abtastrate		50000 ksps (je ADC)
Eingangs-Bandbreite		0 … 4 MHz
Speichergröße		256 MiB oder 2^{27} = 134217728 Messwerte insgesamt
Messbereich		±10 V mit max. Offset 3,5 V
Genauigkeit	INL	typisch ±1,2 LSB, max. ±5 LSB
	DNL	typisch ±1 LSB, max. ±1,5 LSB
Eingangswiderstand		330 kΩ, ±2%
Spannungsfestigkeit:		±35 V
Offsetfehler		abgleichbar
Offsetdrift		±30 ppm/°C vom Endwert
Event-Eingang (nur D-Sub-Buchse)		3 differentiell; RS422/485 kompatibel (5 V differentiell, 120 Ω Bus-Abschlusswiderstand) max. Signalfrequenz 16 MHz
Steckerverbindung		4 LEMO-Buchsen einpolig oder 4 LEMO-Buchsen zweipolig oder 37-polige D-Sub-Buchse oder 4 BNC-Buchsen

Abb. 53 – Pro II-AIn-F-4/14 Rev. E: Spezifikation

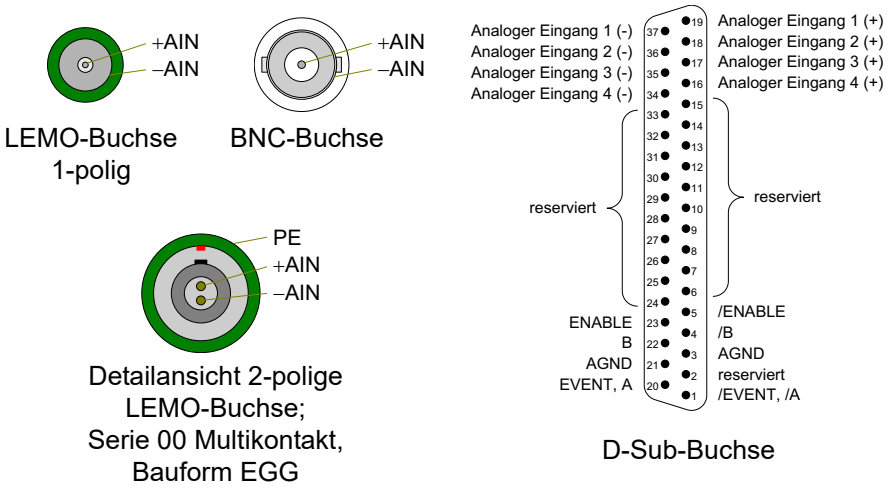


Abb. 54 – Pro II-AIn-F-4/14 Rev. E: Pinbelegung differentiell

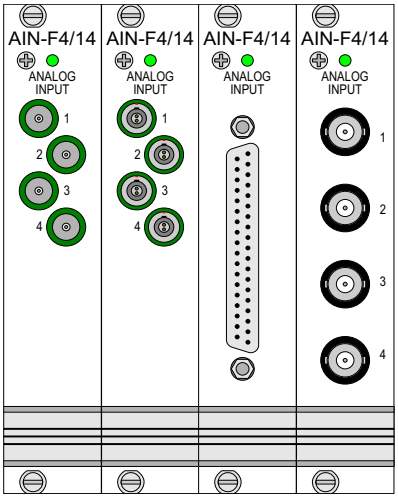


Abb. 55 – Pro II-AIn-F-4/14 Rev. E: Frontplatten

Programmierung

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Im Verzeichnis `C:\ADwin\ADbasic\samples_ADwin_ProII` finden Sie zusätzliche Beispielprogramme.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Ablaufsteuerung konfigurieren	`P2_ADCF_Mode`

Einzelmessung durchführen – vollständig oder schrittweise	`P2_ADCF`
	`P2_Start_ConvF`, `P2_Wait_EOCF`
	`P2_Read_ADCF`

Mehrere Messwerte lesen	`P2_Read_ADCF32`, `P2_Read_ADCF4` `P2_Read_ADCF4_Packed`

Bereich	Befehle
Burst-Messung durchführen	P2_Burst_Init P2_Burst_Start, P2_Burst_Stop P2_Burst_Status, P2_Burst_Reset P2_Burst_Read P2_Burst_Read_Index, P2_Burst_Read_Index P2_Burst_Read_Unpacked1/2/4 P2_Burst_CRead_Unpacked1/2/4 P2_Burst_CRead_Pos_Unpacked1/2/4
Wandlung synchronisieren	P2_Sync_All, P2_Sync_Enable P2_Sync_Stat
Grenzwerte überwachen	P2_ADCF_Read_Limit P2_ADCF_Set_Limit
LEDs einstellen	P2_Check_LED, P2_Set_LED
Interrupts und Event-Eingang einstellen	P2_Event_Enable, P2_Event_Read P2_Event_Config, P2_Event2_Config

5.5.7 Pro II-AIn-F-8/14 Rev. E

Analoges Eingangsmodul Pro II-AIn-F-8/14 Rev. E mit 8 Fast-ADC zu 14 Bit und 8 differentiellen Eingängen.

Für die Eingänge sind folgende Steckverbindungen verfügbar:

– Pro II-AIn-F-8/14: geschirmte LEMO-Buchsen 1-polig, CAMAC Europanorm.

– Pro II-AIn-F-8/14-L2: geschirmte LEMO-Buchsen 2-polig, CAMAC Europanorm.

– Pro II-AIn-F-8/14-D: D-Sub-Buchse 37-polig.

– Pro II-AIn-F-8/14-B: BNC-Buchsen.

Die Wandler des Moduls arbeiten eigenständig mit einer festen Abtastrate von 50 MHz. Wenn auf allen 8 Kanälen Messwerte gewandelt werden, ist die Abtastrate wegen Erreichens der maximalen Speicherzugriffsrate auf 25 MHz je Kanal begrenzt.

Das Modul Pro II-AIn-F-8/14 Rev. E hat einen Eingangs-Spannungsbereich von bipolar ±10V. Der Abgleich der Verstärkung und des Offsets erfolgt per Software (siehe Kapitel 6 “Kalibrierung“).

Der moduleigene Speicher erlaubt, die große Menge anfallender Daten bei Burst-Messungen zwischenzuspeichern.

Das Modul verfügt über mehrere Betriebsarten für die Wandlung von analogen Signalen:

– Einzelmessung: Das Modul führt die Wandlung selbstständig (mit fester Abtastrate) durch. Das *ADbasic*-Programm fragt bei Bedarf den aktuellsten Messwert ab und verarbeitet ihn weiter.

– Einfache Burst-Messreihe: Das *ADbasic*-Programm startet eine vollständige Messreihe, also eine definierte Anzahl von Einzelmessungen.

– Kontinuierliche Burst-Messreihe: Das *ADbasic*-Programm startet eine Messreihe, die kontinuierlich Einzelmessungen ausführt, bis die Messreihe gestoppt wird. Die Daten werden in einem Ringspeicher abgelegt.

Das Modul kann optional anstelle von Einzelmessungen jeweils den Mittelwert aus 2…32 Messwerten zurückgeben. Für jeden Mittelwert wandelt das Modul separat die eingestellte Anzahl an Messwerten.

Das Modul kann für jeden Kanal getrennt das Übertreten eines oberen und eines unteren Grenzwerts überwachen.

Burst-Messreihen

Das Modul führt Burst-Messreihen unabhängig vom Prozessormodul des *ADwin*-Systems aus. Die Messwerte – Anzahl und Messfrequenz sind vorab im Programm zu definieren – werden im Burst-Speicher auf dem Modul abgelegt. Das Prozessormodul liest nur noch die gespeicherten Messwerte (auch während der laufenden Messreihe) und verarbeitet sie.

Bei einer kontinuierlichen Burst-Messreihe muss die Messfrequenz mit der Auslesegeschwindigkeit abgestimmt werden. Hierbei sind verschiedene Aspekte von Bedeutung:

– Das Auslesen der Messwerte geschieht in Blöcken. Je größer der Datenblock ist, um so schneller ist der Lesevorgang im Mittel.

  Beachten Sie: Das blockweise Lesen kann andere, auch hochpriore Prozesse verzögern. Je größer der gelesene Block ist, umso eher kann eine Verzögerung entstehen.

– Der Zeitversatz zwischen kontinuierlicher Wandlung und blockweisem Lesen erfordert einen Datenpuffer. Hierzu muss bei der Initialisierung

der Burst-Messung ein genügend großer Speicherbereich reserviert werden (`P2_Burst_Init`, Parameter `samples`).

Event-Eingänge

Bei der Modulversion Pro II-AIn-F-8/14-D Rev. E (mit D-Sub-Buchse) kann eine Burst-Messreihe über externe Event-Signale gesteuert werden, so dass für jedes (resultierende) Event-Signal ein Messwert gespeichert wird.

Optional kann ein Kanal der Burst-Messreihe als Zeitkanal genutzt werden, in dem bei jedem Event-Signal der Zählerstand des moduleigenen Timers gespeichert wird.

Das Modul hat 3 differentielle Event-Eingänge: `EVENT/A`, `B`, `ENABLE`, deren Signale das Modul zum resultierenden Event-Signal verarbeitet. Diese Vorverarbeitung der Signale ist konfigurierbar mit `P2_Event2_Config`.

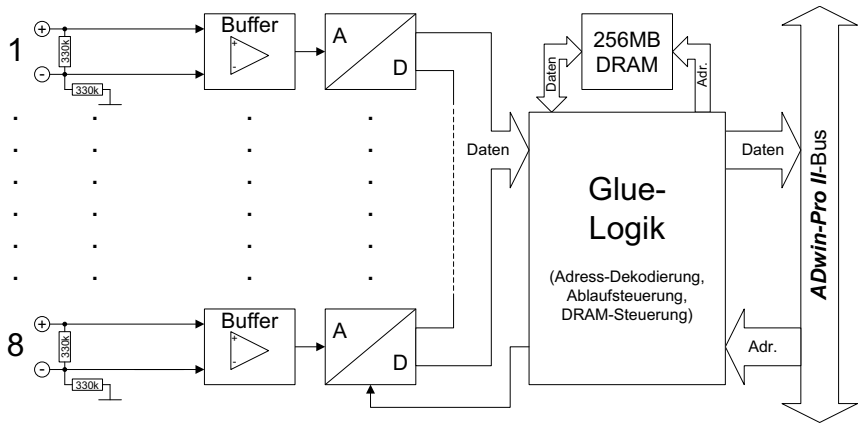


Abb. 56 – Pro II-AIn-F-8/14 Rev. E: Blockschaltbild

Eingangskanäle		8 differentiell
Auflösung		14 Bit
Konvertierungszeit		0,02 µs (je ADC)
Abtastrate		4×50000 ksps (je ADC) oder 8×25000 ksps (je ADC)
Eingangs-Bandbreite		0 … 4 MHz
Speichergröße		256 MiB oder 2^{27} = 134217728 Messwerte insgesamt
Messbereich		±10 V mit max. Offset 3,5 V
Genauigkeit	INL	typisch ±1,2 LSB, max. ±5 LSB
	DNL	typisch ±1 LSB, max. ±1,5 LSB
Eingangswiderstand		330 kΩ, ±2%
Spannungsfestigkeit:		±35 V
Offsetfehler		abgleichbar
Offsetdrift		±30 ppm/°C vom Endwert
Event-Eingang (nur D-Sub-Buchse)		3 differentiell; RS422/485 kompatibel (5 V differentiell, 120 Ω Bus-Abschlusswiderstand) max. Signalfrequenz 16 MHz
Steckerverbindung		8 LEMO-Buchsen einpolig oder 8 LEMO-Buchsen zweipolig oder 37-polige D-Sub-Buchse oder 8 BNC-Buchsen

Abb. 57 – Pro II-AIn-F-8/14 Rev. E: Spezifikation

Modulbreite	10 TE

Abb. 57 – Pro II-AIn-F-8/14 Rev. E: Spezifikation

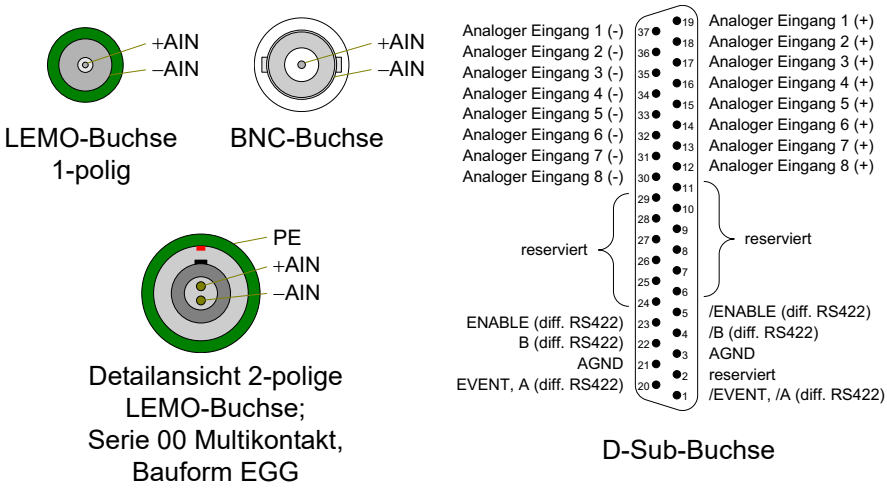


Abb. 58 – Pro II-AIn-F-8/14 Rev. E: Pinbelegung differentiell

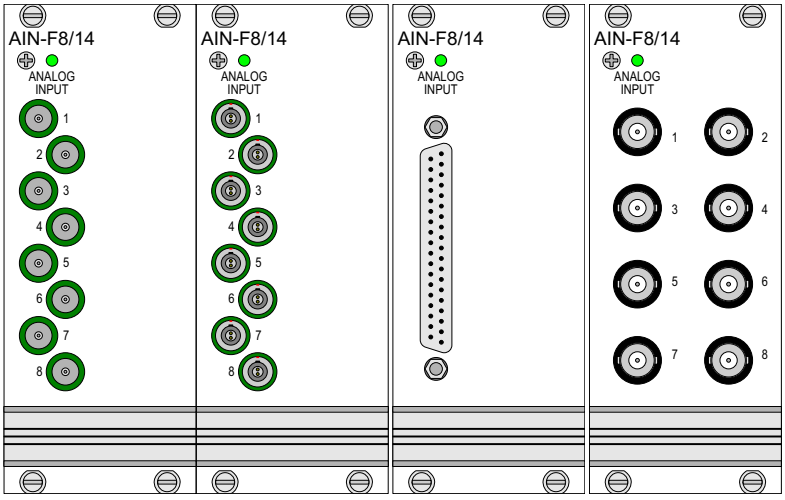


Abb. 59 – Pro II-AIn-F-8/14 Rev. E: Frontplatten

Programmierung

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Im Verzeichnis `C:\ADwin\ADbasic\samples_ADwin_ProII` finden Sie zusätzliche Beispielprogramme.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Ablaufsteuerung konfigurieren	`P2_ADCF_Mode`

Einzelmessung durchführen – vollständig oder schrittweise	`P2_ADCF`
	`P2_Start_ConvF`, `P2_Wait_EOCF`
	`P2_Read_ADCF`

Mehrere Messwerte lesen	`P2_Read_ADCF32`
	`P2_Read_ADCF4`, `P2_Read_ADCF8`
	`P2_Read_ADCF4_Packed`
	`P2_Read_ADCF8_Packed`

Bereich	Befehle
Burst-Messung durchführen	P2_Burst_Init P2_Burst_Start, P2_Burst_Stop P2_Burst_Status, P2_Burst_Reset P2_Burst_Read P2_Burst_Read_Index, P2_Burst_Read_Index P2_Burst_Read_Unpacked1/2/4/8 P2_Burst_CRead_Unpacked1/2/4/8 P2_Burst_CRead_Pos_Unpacked1/2/4/8
Wandlung synchronisieren	P2_Sync_All, P2_Sync_Enable P2_Sync_Stat
Grenzwerte überwachen	P2_ADCF_Read_Limit P2_ADCF_Set_Limit
LEDs einstellen	P2_Check_LED, P2_Set_LED
Interrupts und Event-Eingang einstellen	P2_Event_Enable, P2_Event_Read P2_Event_Config, P2_Event2_Config

5.5.8 Pro II-AIn-F-4/16 Rev. E

Analoges Eingangsmodul Pro II-AIn-F-4/16 Rev. E mit 4 Fast-ADC zu 16 Bit und 4 differentiellen Eingängen.

Für die Eingänge sind folgende Steckverbindungen verfügbar:

– Pro II-AIn-F-4/16: geschirmte LEMO-Buchsen 1-polig, CAMAC Europanorm.

– Pro II-AIn-F-4/16-L2: geschirmte LEMO-Buchsen 2-polig, CAMAC Europanorm.

– Pro II-AIn-F-4/16-D: D-Sub-Buchse 37-polig.

– Pro II-AIn-F-4/16-B: BNC-Buchsen.

Die Wandler des Moduls arbeiten mit einer Abtastrate von bis zu 4 MHz. Ab Rev. E04 erlaubt der moduleigene Speicher, die anfallenden Daten bei Burst-Messungen zwischenzuspeichern.

Das Modul Pro II-AIn-F-4/16 Rev. E hat einen Eingangs-Spannungsbereich von ±10V und per Software programmierbare Verstärkung von 1, 2, 4 oder 8. Der Abgleich der Verstärkung und des Offsets erfolgt per Software (siehe Kapitel 6 “Kalibrierung“).

Ab Rev. E04 verfügt das Modul über mehrere Betriebsarten für die Wandlung von analogen Signalen:

– Einzelmessung: Jede Messung wird im *ADbasic*-Programm einzeln gestartet, abgefragt und weiter verarbeitet.

– Einfache Burst-Messreihe: Das *ADbasic*-Programm startet eine vollständige Messreihe, also eine definierte Anzahl von Einzelmessungen.

– Kontinuierliche Burst-Messreihe: Das *ADbasic*-Programm startet eine Messreihe, die kontinuierlich Einzelmessungen ausführt, bis die Messreihe gestoppt wird. Die Daten werden in einem Ringspeicher abgelegt.

Alternativ zu den genannten Betriebsarten kann das Modul mit einer Ablaufsteuerung die Wandlung auf allen Kanälen gleichzeitig durchführen. Dadurch wird – im Vergleich zu Einzelmessungen mit dem Befehl `P2_ADCF` – das Prozessormodul entlastet, das im Prozess nur noch die gewandelten Werte liest und verarbeitet. Die Wandlung kann entweder zyklisch in regelmäßigen Zeitabständen oder durch externe Event-Signale ausgelöst werden.

Das Modul kann optional anstelle von Einzelmessungen jeweils den Mittelwert aus 2…32 Messwerten zurückgeben. Für jeden Mittelwert wandelt das Modul separat die eingestellte Anzahl an Messwerten.

Das Modul kann für jeden Kanal getrennt das Übertreten eines oberen und eines unteren Grenzwerts überwachen.

Das Modul sammelt für jeden Messkanal getrennt die Maximal- und Minimalwerte und stellt sie per Software-Befehl zur Verfügung.

Burst-Messreihen

Das Modul führt Burst-Messreihen unabhängig vom Prozessormodul des *ADwin*-Systems aus. Die Messwerte – Anzahl und Messfrequenz legen Sie vorab im Programm fest – werden im Burst-Speicher auf dem Modul abgelegt. Das Prozessormodul liest nur noch die gespeicherten Messwerte (auch während der laufenden Messreihe) und verarbeitet sie.
Burst-Messreihen sind ab Rev. E04 verfügbar.

Bei einer kontinuierlichen Burst-Messreihe müssen Sie die Messfrequenz mit der Auslesegeschwindigkeit abstimmen. Hierbei sind verschiedene Aspekte von Bedeutung:

– Das Auslesen der Messwerte geschieht in Blöcken. Je größer der Datenblock ist, um so schneller ist der Lesevorgang im Mittel.

Beachten Sie: Das blockweise Lesen kann andere, auch hochpriore Prozesse verzögern. Je größer der gelesene Block ist, umso eher kann eine Verzögerung entstehen.

– Der Zeitversatz zwischen kontinuierlicher Wandlung und blockweisem Lesen erfordert einen Datenpuffer. Hierzu müssen Sie bei der Initialisierung der Burst-Messung einen genügend großen Speicherbereich reserviert (**P2_Burst_Init**, Parameter samples).

Event-Eingänge

Bei der Modulversion Pro II-AIn-F-4/16-D Rev. E (mit D-Sub-Buchse) kann eine Burst-Messreihe über externe Event-Signale gesteuert werden, so dass für jedes (resultierende) Event-Signal ein Messwert gespeichert wird.

Optional kann ein Kanal der Burst-Messreihe als Zeitkanal genutzt werden, in dem bei jedem Event-Signal der Zählerstand des moduleigenen Timers gespeichert wird.

Das Modul hat 3 differentielle Event-Eingänge: EVENT/A, B, ENABLE, deren Signale das Modul zum resultierenden Event-Signal verarbeitet. Diese Vorverarbeitung der Signale ist konfigurierbar mit **P2_Event2_Config**.

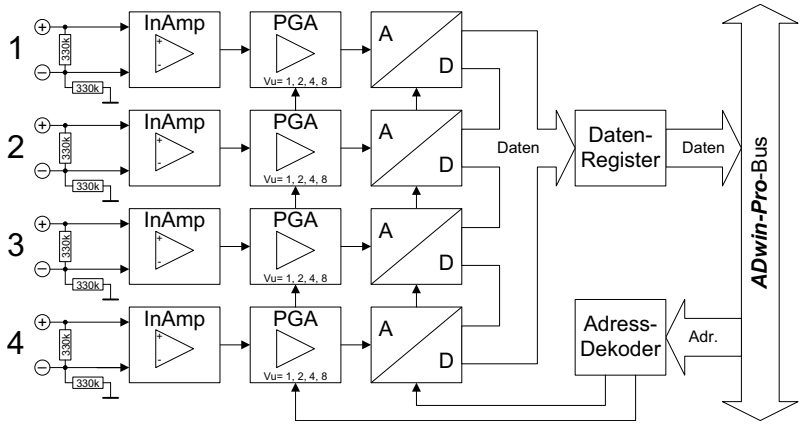


Abb. 60 – Pro II-AIn-F-4/16 Rev. E: Blockschaltbild

Eingangskanäle		4 differentiell
Auflösung		16 Bit
Konvertierungszeit		0,25 µs (je ADC)
Eingangs-Bandbreite		0 … 600 kHz
Speichergröße (ab Rev. E04)		256 MiB oder 2^{27} = 134217728 Messwerte insgesamt
Messbereich		±10 V mit max. Offset 3,5 V
Verstärkung:		1, 2, 4, 8 per Software einstellbar
Genauigkeit	INL	typisch ±1,2 LSB, max. ±5 LSB
	DNL	typisch ±1 LSB, max. ±1,5 LSB
Eingangswiderstand		330 kΩ, ±2%
Spannungsfestigkeit:		±20 V
Offsetfehler		abgleichbar
Offsetdrift		±30 ppm/°C vom Endwert
Event-Eingang (nur D-Sub-Buchse)		3 differentiell; RS422/485 kompatibel (5 V differentiell, 120 Ω Bus-Abschlusswiderstand)

Abb. 61 – Pro II-AIn-F-4/16 Rev. E: Spezifikation

Steckerverbindung	4 LEMO-Buchsen einpolig oder 4 LEMO-Buchsen zweipolig oder 37-polige D-Sub-Buchse oder 4 BNC-Buchsen

Abb. 61 – Pro II-AIn-F-4/16 Rev. E: Spezifikation

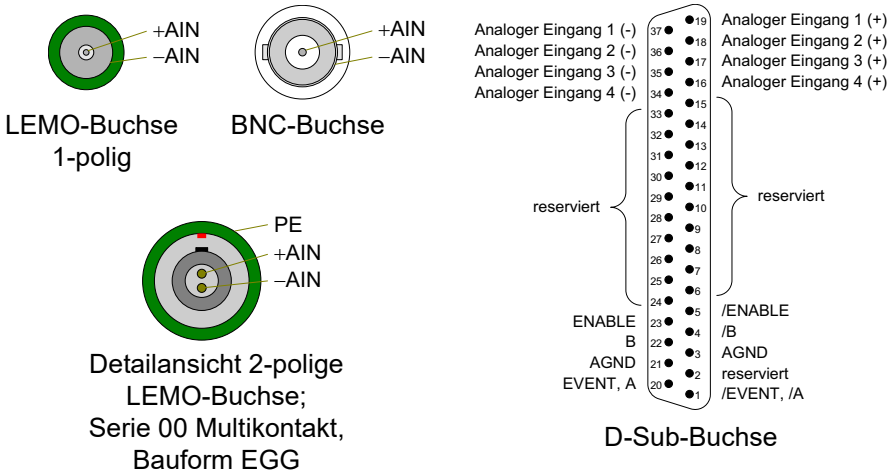


Abb. 62 – Pro-AIn-F-4/16-D Rev. E: Pinbelegung

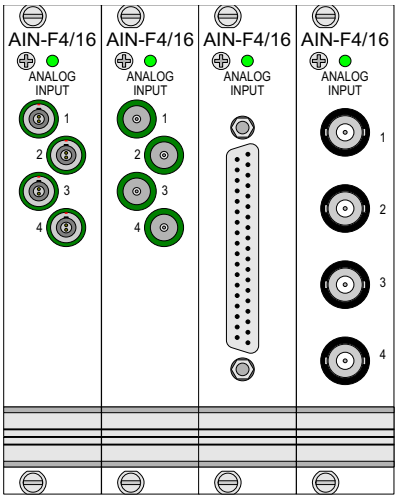


Abb. 63 – Pro II-AIn-F-4/16 Rev. E: Frontplatten

Programmierung

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Im Verzeichnis `C:\ADwin\ADbasic\samples_ADwin_ProII` finden Sie das Beispielprogramm `ProII-AIn-F.bas`.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Ablaufsteuerung konfigurieren	`P2_ADCF_Mode`, `P2_Set_Average_Filter`, `P2_Set_Gain`

Einzelmessung durchführen – vollständig oder schrittweise	`P2_ADCF`
	`P2_Start_ConvF`, `P2_Wait_EOCF`
	`P2_Read_ADCF`

Bereich	Befehle
Mehrere Messwerte lesen	P2_Read_ADCF32, P2_Read_ADCF4 P2_Read_ADCF4_Packed
Messwert lesen und neue Messung starten	P2_Read_ADC_SConv P2_Read_ADC_SConv32
Burst-Messung durchführen	P2_Burst_Init P2_Burst_Start, P2_Burst_Stop P2_Burst_Status, P2_Burst_Reset P2_Burst_Read P2_Burst_Read_Index, P2_Burst_Read_Index_Limit P2_Burst_Read_Unpacked1/2/4 P2_Burst_CRead_Unpacked1/2/4 P2_Burst_CRead_Pos_Unpacked1/2/4
Wandlung synchronisieren	P2_Sync_All, P2_Sync_Enable P2_Sync_Mode, P2_Sync_Stat
Grenzwerte überwachen	P2_ADCF_Read_Limit P2_ADCF_Set_Limit
Maximal- und Minimalwerte bereitstellen	P2_ADCF_Reset_Min_Max P2_ADCF_Read_Min_Max4
LEDs einstellen	P2_Check_LED, P2_Set_LED
Interrupts und Event-Eingang einstellen	P2_Event_Enable, P2_Event_Read P2_Event_Config, P2_Event2_Config

5.5.9 Pro II-AIn-F-8/16 Rev. E

Analoges Eingangsmodul Pro II-AIn-F-8/16 Rev. E mit 8 Fast-ADC zu 16 Bit und 8 differentiellen Eingängen.

Das Modul kann in folgenden Varianten bestellt werden:

	ohne Filter	Filter 50 kHz	Filter 10 kHz, Messbereich ±30 V
LEMO 1-polig	AIn-F-8/16	AIn-F-8/16-LP50	AIn-F-8/16-LP-30V
LEMO 2-polig	AIn-F-8/16-L2	AIn-F-8/16-LP50-L2	AIn-F-8/16-LP-30V-L2
D-Sub	AIn-F-8/16-D	AIn-F-8/16-LP50-D	AIn-F-8/16-LP-30V-D
BNC-Buchsen	AIn-F-8/16-B	AIn-F-8/16-LP50-B	AIn-F-8/16-LP-30V-B

Die Varianten Pro II-AIn-8/18-xxx-LP enthalten einen Eingangsfilter, einen Tiefpass 4. Ordnung vom Typ Butterworth. Je nach Variante ist die Eckfrequenz auf 10 kHz oder 50 kHz festgelegt.

Die Eingänge liegen auf geschirmten LEMO-Buchsen (1-polig oder 2-polig, CAMAC Europanorm, siehe Abb. 66), BNC-Buchsen oder auf einer 37-polige D-Sub-Buchse.

Die Wandler des Moduls arbeiten mit einer Abtastrate von bis zu 4 MHz. Ab Rev. E04 erlaubt der moduleigene Speicher, die anfallenden Daten bei Burst-Messungen zwischenzuspeichern.

Das Modul Pro II-AIn-F-8/16 Rev. E hat einen Eingangs-Spannungsbereich von ±10V (Variante LP-30V: ±30V) und per Software programmierbare Verstärkung von 1, 2, 4 oder 8. Der Abgleich der Verstärkung und des Offsets erfolgt per Software (siehe Kapitel 6 “Kalibrierung“).

Ab Rev. E04 verfügt das Modul über mehrere Betriebsarten für die Wandlung von analogen Signalen:

– Einzelmessung: Jede Messung wird im *ADbasic*-Programm einzeln gestartet, abgefragt und weiter verarbeitet.
– Einfache Burst-Messreihe: Das *ADbasic*-Programm startet eine vollständige Messreihe, also eine definierte Anzahl von Einzelmessungen.
– Kontinuierliche Burst-Messreihe: Das *ADbasic*-Programm startet eine Messreihe, die kontinuierlich Einzelmessungen ausführt, bis die Messreihe gestoppt wird. Die Daten werden in einem Ringspeicher abgelegt.

Alternativ zu den genannten Betriebsarten kann das Modul mit einer Ablaufsteuerung eine Wandlung auf allen Kanälen gleichzeitig durchführen. Dadurch wird – im Vergleich zu Einzelmessungen mit dem Befehl `P2_ADCF` – das Prozessormodul entlastet, das im Prozess nur noch die gewandelten Werte liest und verarbeitet. Die Wandlung kann entweder zyklisch in regelmäßigen Zeitabständen oder durch externe Event-Signale ausgelöst werden.

Das Modul kann optional anstelle von Einzelmessungen jeweils den Mittelwert aus 2…32 Messwerten zurückgeben. Für jeden Mittelwert wandelt das Modul separat die eingestellte Anzahl an Messwerten, siehe.

Das Modul kann für jeden Kanal getrennt das Übertreten eines oberen und eines unteren Grenzwerts überwachen.

Das Modul sammelt für jeden Messkanal getrennt die Maximal- und Minimalwerte und stellt sie per Software-Befehl zur Verfügung.

Burst-Messreihen

Das Modul führt Burst-Messreihen unabhängig vom Prozessormodul des *ADwin*-Systems aus. Die Messwerte – Anzahl und Messfrequenz sind vorab im Programm zu definieren – werden im Burst-Speicher auf dem Modul abgelegt. Das Prozessormodul liest nur noch die gespeicherten Messwerte (auch während der laufenden Messreihe) und verarbeitet sie.

Bei einer kontinuierlichen Burst-Messreihe muss die Messfrequenz mit der Auslesegeschwindigkeit abgestimmt werden. Hierbei sind verschiedene Aspekte von Bedeutung:

– Das Auslesen der Messwerte geschieht in Blöcken. Je größer der Datenblock ist, um so schneller ist der Lesevorgang im Mittel.

  Beachten Sie: Das blockweise Lesen kann andere, auch hochpriore Prozesse verzögern. Je größer der gelesene Block ist, umso eher kann eine Verzögerung entstehen.

– Der Zeitversatz zwischen kontinuierlicher Wandlung und blockweisem Lesen erfordert einen Datenpuffer. Hierzu muss bei der Initialisierung der Burst-Messung ein genügend großer Speicherbereich reserviert werden (**`P2_Burst_Init`**, Parameter `samples`).

Event-Eingänge

Bei der Modulversion Pro II-AIn-F-8/16-D Rev. E (mit D-Sub-Buchse) kann eine Burst-Messreihe über externe Event-Signale gesteuert werden, so dass für jedes (resultierende) Event-Signal ein Messwert gespeichert wird.

Optional kann ein Kanal der Burst-Messreihe als Zeitkanal genutzt werden, in dem bei jedem Event-Signal der Zählerstand des moduleigenen Timers gespeichert wird.

Das Modul hat 3 differentielle Event-Eingänge: `EVENT/A`, `B`, `ENABLE`, deren Signale das Modul zum resultierenden Event-Signal verarbeitet. Diese Vorverarbeitung der Signale ist konfigurierbar mit **`P2_Event2_Config`**.

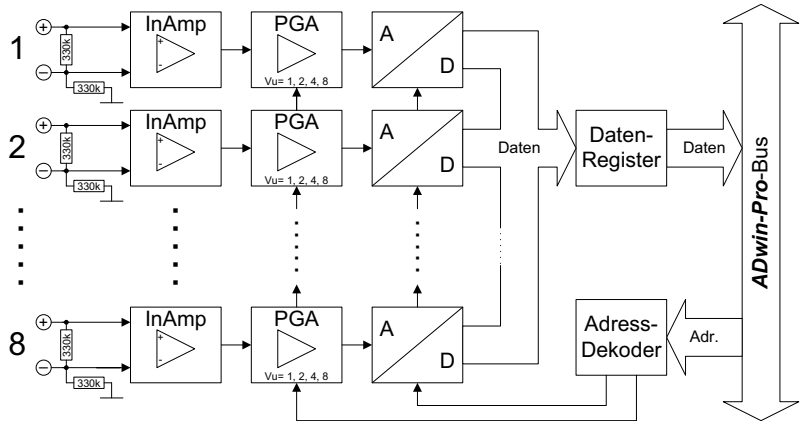


Abb. 64 – Pro II-AIn-F-8/16 Rev. E: Blockschaltbild

Eingangskanäle	8 differentiell
Auflösung	16 Bit
Konvertierungszeit	0,25µs (je ADC)
Eingangs-Bandbreite	0 … 600kHz
Speichergröße (ab Rev. E04)	256MiB oder 2^{27} = 134217728 Messwerte insgesamt
Messbereich	±10V mit max. Offset 3,5V
Verstärkung:	1, 2, 4, 8 per Software einstellbar

Abb. 65 – Pro II-AIn-F-8/16 Rev. E: Spezifikation

Genauigkeit	INL	typisch ±1,2 LSB, max. ±5 LSB
	DNL	typisch ±1 LSB, max. ±1,5 LSB
Eingangswiderstand		330 kΩ, ±2%
Spannungsfestigkeit:		±20 V
Offsetfehler		abgleichbar
Offsetdrift		±30 ppm/ºC vom Endwert
Event-Eingang (nur D-Sub-Buchse)		3 differentiell; RS422/485 kompatibel (5 V differentiell, 120 Ω Bus-Abschlusswiderstand)
Steckerverbindung		8 LEMO-Buchsen einpolig oder 8 LEMO-Buchsen zweipolig oder 37-polige D-Sub-Buchse oder 8 BNC-Buchsen
Modulbreite		5 TE; mit BNC-Buchsen: 10 TE

Abb. 65 – Pro II-AIn-F-8/16 Rev. E: Spezifikation

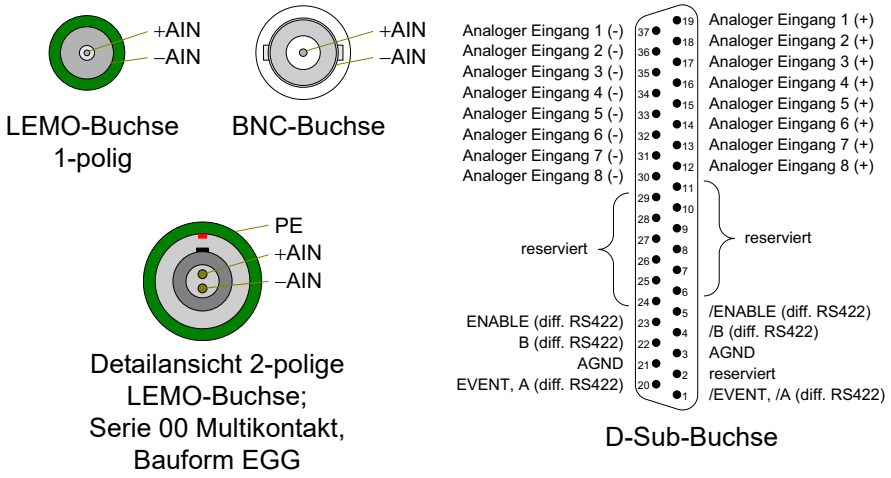


Abb. 66 – Pro II-AIn-F-8/16 Rev. E: Pinbelegung

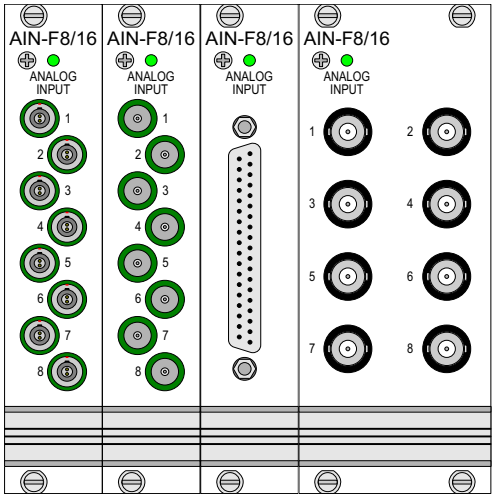


Abb. 67 – Pro II-AIn-F-8/16 Rev. E: Frontplatten

Programmierung

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Im Verzeichnis `C:\ADwin\ADbasic\samples_ADwin_ProII` finden Sie das Beispielprogramm `ProII-AIn-F.bas`.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Ablaufsteuerung konfigurieren: Modus, Verstärkung, Mittelwert	**P2_ADCF_Mode**, **P2_Set_Gain** **P2_Set_Average_Filter**
Einzelmessung durchführen – vollständig oder schrittweise	**P2_ADCF** **P2_Start_ConvF**, **P2_Wait_EOCF** **P2_Read_ADCF**
Mehrere Messwerte lesen	**P2_Read_ADCF32** **P2_Read_ADCF4**, **P2_Read_ADCF8** **P2_Read_ADCF4_Packed** **P2_Read_ADCF8_Packed**
Messwert lesen und neue Messung starten	**P2_Read_ADC_SConv** **P2_Read_ADC_SConv32**
Burst-Messung durchführen	**P2_Burst_Init** **P2_Burst_Start**, **P2_Burst_Stop** **P2_Burst_Status**, **P2_Burst_Reset** **P2_Burst_Read** **P2_Burst_Read_Index**, **P2_Burst_Read_Index_Limit** **P2_Burst_Read_Unpacked1/2/4/8** **P2_Burst_CRead_Unpacked1/2/4/8** **P2_Burst_CRead_Pos_Unpacked1/2/4/8**
Wandlung synchronisieren	**P2_Sync_All**, **P2_Sync_Enable** **P2_Sync_Mode**, **P2_Sync_Stat**
Grenzwerte überwachen	**P2_ADCF_Read_Limit** **P2_ADCF_Set_Limit**
Maximal- und Minimalwerte bereitstellen	**P2_ADCF_Reset_Min_Max** **P2_ADCF_Read_Min_Max4** **P2_ADCF_Read_Min_Max8**
LEDs einstellen	**P2_Check_LED**, **P2_Set_LED**
Interrupts und Event-Eingang einstellen	**P2_Event_Enable**, **P2_Event_Read** **P2_Event_Config**, **P2_Event2_Config**

5.5.10 Pro II-AIn-F-4/18 Rev. E

Analoges Eingangsmodul Pro II-AIn-F-4/18 Rev. E mit 4 Fast-ADC zu 18 Bit und 4 differentiellen Eingängen. Die Eingänge sind voneinander und von anderen Modulen galvanisch getrennt.

Für die Eingänge sind folgende Steckverbindungen verfügbar:

– Pro II-AIn-F-4/18: geschirmte LEMO-Buchsen, CAMAC Europanorm.

– Pro II-AIn-F-4/18-L2: geschirmte LEMO-Buchsen 2-polig, CAMAC Europanorm.

– Pro II-AIn-F-4/18-D: D-Sub-Buchse 37-polig.

– Pro II-AIn-F-4/18-B: BNC-Buchsen.

Das Modul Pro II-AIn-F-4/18 Rev. E hat einen Eingangs-Spannungsbereich von bipolar ±10V. Der Abgleich des Offsets erfolgt per Software (siehe Kapitel 6 “Kalibrierung“).

Das Modul kann die Wandlung auf allen Kanälen gleichzeitig durchführen. Dadurch wird – im Vergleich zu Einzelmessungen mit dem Befehl **P2_ADCF** – das Prozessormodul entlastet, das im Prozess nur noch die gewandelten Werte liest und verarbeitet. Die Wandlung kann entweder zyklisch in regelmäßigen Zeitabständen oder durch externe Event-Signale ausgelöst werden.

Das Modul kann für jeden Kanal getrennt das Übertreten eines oberen und eines unteren Grenzwerts überwachen.

Um Kanäle als single-ended-Eingänge zu nutzen, verbindet man das Eingangssignal mit dem Plus-Eingang und die Masse mit dem Minus-Eingang.

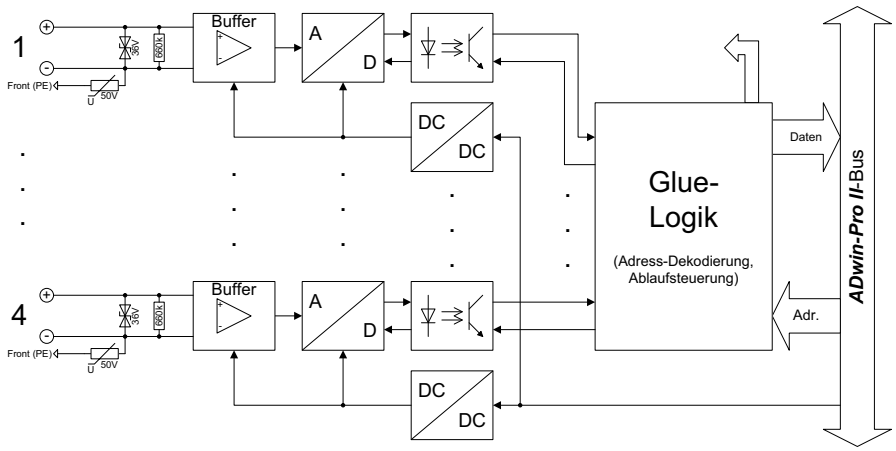


Abb. 68 – Pro II-AIn-F-4/18 Rev. E: Blockschaltbild

Als Masseverbindung steht die Gehäusemasse (PE) an der Frontplatte zur Verfügung. Alle Analogeingänge sind von der Gehäusemasse galvanisch getrennt.

Bei den Modulversionen -L2 (2-polige Lemo-Buchsen) und -D (D-Sub-Buchse) steht die Gehäusemasse PE auch am Schirm der Buchse zur Verfügung.

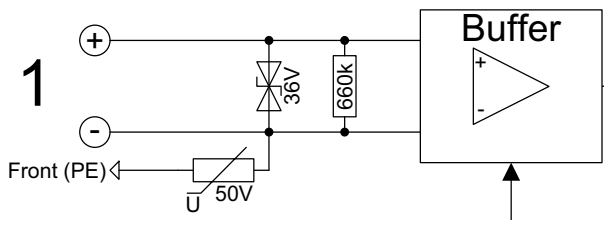


Abb. 69 – Pro II-AIn-F-4/18 Rev. E: Eingangsverschaltung

Eingangskanäle		4 differentiell, galvanisch getrennt
Auflösung		18 Bit
Wandlungszeit		max. 2µs (je ADC)
Abtastrate		max. 500ksps (je ADC)
Eingangs-Bandbreite		0 … 0,6MHz
Messbereich		±10V
Genauigkeit	INL	max. ±4 LSB
	DNL	max. ±3 LSB
Eingangswiderstand		660kΩ, ±2%
Spannungsfestigkeit:		±35V
Offsetfehler		abgleichbar
Offsetdrift		±30ppm/°C vom Endwert
Steckerverbindungen		4 LEMO-Buchsen einpolig oder 4 LEMO-Buchsen zweipolig oder 37-polige D-Sub-Buchse oder 4 BNC-Buchsen

Abb. 70 – Pro II-AIn-F-4/18 Rev. E: Spezifikation

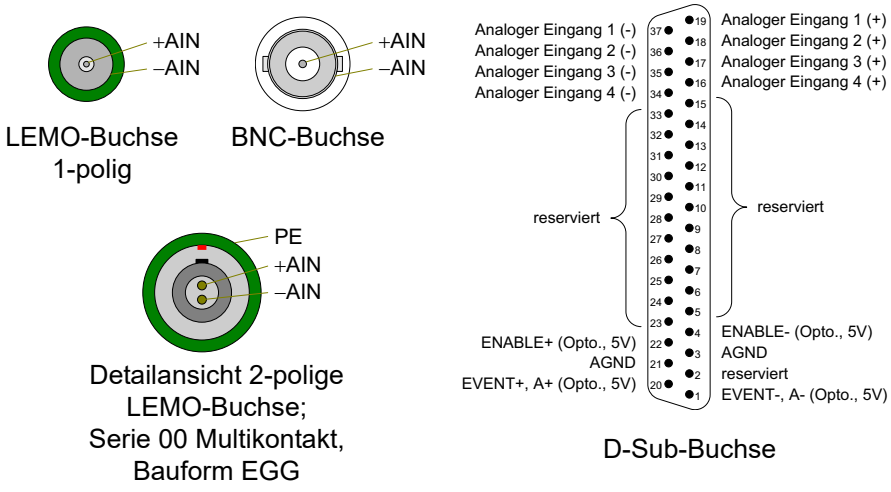


Abb. 71 – Pro II-AIn-F-4/18 Rev. E: Pinbelegung differentiell

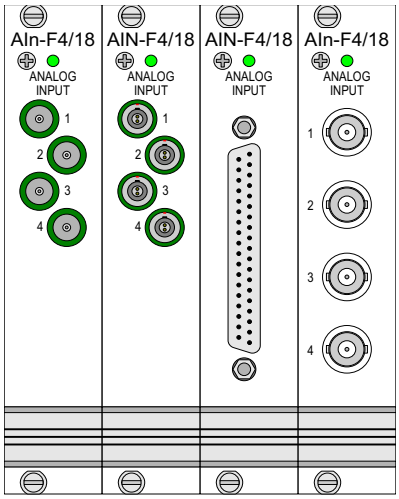


Abb. 72 – Pro II-AIn-F-4/18 Rev. E: Frontplatten

Programmierung

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Im Verzeichnis `C:\ADwin\ADbasic\samples_ADwin_ProII` finden Sie das Beispielprogramm `ProII-AIn-F.bas`.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Ablaufsteuerung konfigurieren	**P2_ADCF_Mode**
Einzelmessung durchführen – vollständig oder schrittweise	**P2_ADCF**, **P2_ADCF24** **P2_Start_ConvF**, **P2_Wait_EOCF** **P2_Read_ADCF**, **P2_Read_ADCF24**
Mehrere Messwerte lesen	**P2_Read_ADCF32** **P2_Read_ADCF4**, **P2_Read_ADCF4_24B** **P2_Read_ADCF4_Packed**
Messwert lesen und neue Messung starten	**P2_Read_ADC_SConv** **P2_Read_ADC_SConv24** **P2_Read_ADC_SConv32**
Wandlung synchronisieren	**P2_Sync_All**, **P2_Sync_Enable** **P2_Sync_Stat**
Grenzwerte überwachen	**P2_ADCF_Read_Limit** **P2_ADCF_Set_Limit**
LEDs einstellen	**P2_Check_LED**, **P2_Set_LED**
Interrupts und Event-Eingang einstellen	**P2_Event_Enable**, **P2_Event_Read** **P2_Event_Config**, **P2_Event2_Config**

5.5.11 Pro II-AIn-F-8/18 Rev. E

Analoges Eingangsmodul Pro II-AIn-F-8/18 Rev. E mit 8 Fast-ADC zu 18 Bit und 8 differentiellen Eingängen. Die Eingänge sind voneinander und von anderen Modulen galvanisch getrennt.

Für die Eingänge sind folgende Steckverbindungen verfügbar:

– Pro II-AIn-F-8/18: geschirmte LEMO-Buchsen, CAMAC Europanorm.

– Pro II-AIn-F-8/18-L2: geschirmte LEMO-Buchsen 2-polig, CAMAC Europanorm.

– Pro II-AIn-F-8/18-D: D-Sub-Buchse 37-polig.

– Pro II-AIn-F-8/18-B: BNC-Buchsen.

Das Modul Pro II-AIn-F-8/18 Rev. E hat einen Eingangs-Spannungsbereich von bipolar ±10V. Der Abgleich des Offsets erfolgt per Software (siehe Kapitel 6 “Kalibrierung“).

Das Modul kann die Wandlung auf allen Kanälen gleichzeitig durchführen. Dadurch wird – im Vergleich zu Einzelmessungen mit dem Befehl **P2_ADCF** – das Prozessormodul entlastet, das im Prozess nur noch die gewandelten Werte liest und verarbeitet. Die Wandlung kann entweder zyklisch in regelmäßigen Zeitabständen oder durch externe Event-Signale ausgelöst werden.

Das Modul kann für jeden Kanal getrennt das Übertreten eines oberen und eines unteren Grenzwerts überwachen.

Um Kanäle als single-ended-Eingänge zu nutzen, verbindet man das Eingangssignal mit dem Plus-Eingang und die Masse mit dem Minus-Eingang.

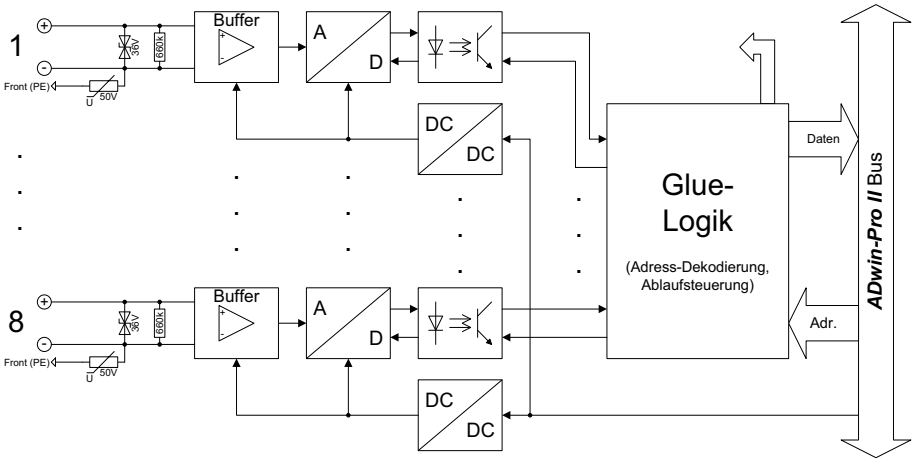


Abb. 73 – Pro II-AIn-F-8/18 Rev. E: Blockschaltbild

Als Masseverbindung steht die Gehäusemasse (PE) an der Frontplatte zur Verfügung. Alle Analogeingänge sind von der Gehäusemasse galvanisch getrennt.

Bei den Modulversionen -L2 (2-polige Lemo-Buchsen) und -D (D-Sub-Buchse) steht die Gehäusemasse PE auch am Schirm der Buchse zur Verfügung.

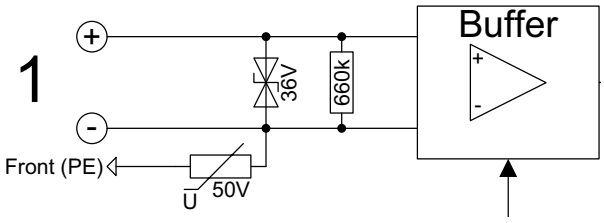


Abb. 74 – Pro II-AIn-F-4/18 Rev. E: Eingangsverschaltung

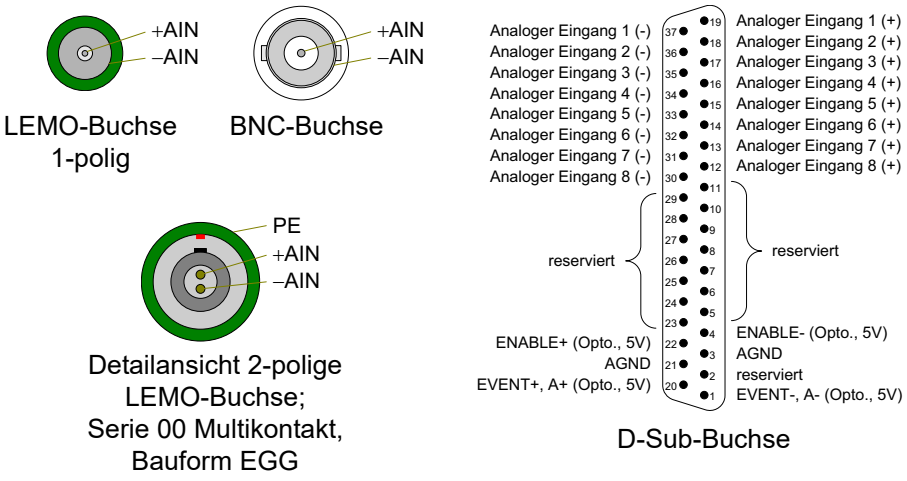


Abb. 75 – Pro II-AIn-F-8/18-D Rev. E: Pinbelegung differentiell

Eingangskanäle		8 differentiell, galvanisch getrennt
Auflösung		18 Bit
Wandlungszeit		max. 2µs (je ADC)
Abtastrate		max. 500ksps (je ADC)
Eingangs-Bandbreite		0 … 0,6MHz
Messbereich		±10V
Genauigkeit	INL	max. ±4 LSB
	DNL	max. ±3 LSB
Eingangswiderstand		660kΩ, ±2%
Spannungsfestigkeit:		±35V
Offsetfehler		abgleichbar
Offsetdrift		±30ppm/ºC vom Endwert
Steckerverbindungen		8 LEMO-Buchsen einpolig oder 8LEMO-Buchsen zweipolig oder 37-polige D-Sub-Buchse oder 8 BNC-Buchsen
Modulbreite		5 TE; mit BNC-Buchsen: 10 TE

Abb. 76 – Pro II-AIn-F-8/18 Rev. E: Spezifikation

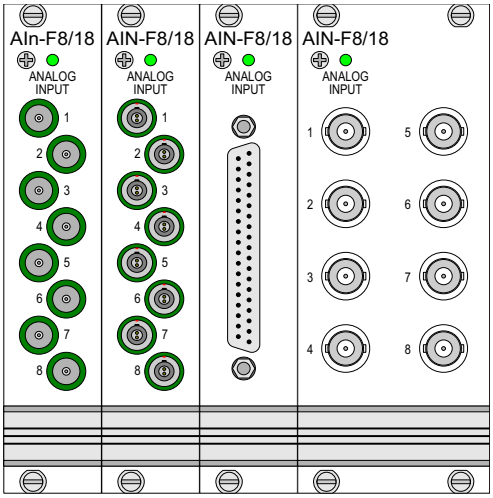


Abb. 77 – Pro II-AIn-F-8/18 Rev. E: Frontplatten

Programmierung

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Im Verzeichnis `C:\ADwin\ADbasic\samples_ADwin_ProII` finden Sie das Beispielprogramm `ProII-AIn-F.bas`.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Ablaufsteuerung konfigurieren	**P2_ADCF_Mode**
Einzelmessung durchführen – vollständig oder schrittweise	**P2_ADCF**, **P2_ADCF24** **P2_Start_ConvF**, **P2_Wait_EOCF** **P2_Read_ADCF**, **P2_Read_ADCF24**
Mehrere Messwerte lesen	**P2_Read_ADCF32** **P2_Read_ADCF4**, **P2_Read_ADCF4_24B** **P2_Read_ADCF4_Packed** **P2_Read_ADCF8**, **P2_Read_ADCF8_24B** **P2_Read_ADCF8_Packed**
Messwert lesen und neue Messung starten	**P2_Read_ADC_SConv** **P2_Read_ADC_SConv24** **P2_Read_ADC_SConv32**
Wandlung synchronisieren	**P2_Sync_All**, **P2_Sync_Enable** **P2_Sync_Stat**
Grenzwerte überwachen	**P2_ADCF_Read_Limit** **P2_ADCF_Set_Limit**
LEDs einstellen	**P2_Check_LED**, **P2_Set_LED**
Interrupts und Event-Eingang einstellen	**P2_Event_Enable**, **P2_Event_Read** **P2_Event_Config**, **P2_Event2_Config**

5.6 Pro II: Analoge Ausgangsmodule

Dieser Abschnitt beschreibt analoge Ausgangsmodule für ***ADwin-Pro II***.

Sie finden analoge Ausgangsmodule für ***ADwin-Pro I*** im Handbuch „ADwin-Pro-Hardware ab Seite 65.

Modulname	AOut 4/16	AOut 4/16-TiCo	AOut 8/16	AOut 8/16-TiCo	AOut 1/16
Revision	E	E	E	E	E
Anzahl DAC	4	4	8	8	1
Auflösung [Bit]	16	16	16	16	16
max. Einschwingzeit [µs]	< 3	< 3	< 3	< 3	0,015
Kanäle sng. end.	4	4	8	8	1
Ausgangsspannung	±10V	±10V	±10V	±10V	±2V
Kalibrierung per Software	ja	ja	ja	ja	ja
TiCo-Prozessor	–	TiCo1	–	TiCo1	TiCo1
Seite	94	94	97	97	100

5.6.1 Pro II-AOut-4/16 Rev. E

Das analoge Ausgangsmodul Pro II-AOut-4/16 Rev. E hat 4 DAC zu 16 Bit mit festem Tiefpass 1. Ordnung, um Störungen zu unterdrücken (f_g = 10MHz).

Das Modul kann in folgenden Varianten bestellt werden:

	ohne TiCo-Prozessor	mit TiCo-Prozessor
geschirmte LEMO-Buchsen 1-polig, CAMAC Europanorm	Pro II-AOut-4/16	Pro II-AOut-4/16-TiCo
geschirmte LEMO-Buchsen 2-polig, CAMAC Europanorm	Pro II-AOut-4/16-L2	Pro II-AOut-4/16-L2-TiCo
D-Sub-Buchse 37-polig	Pro II-AOut-4/16-D	Pro II-AOut-4/16-D-TiCo
BNC-Buchsen	Pro II-AOut-4/16-B	Pro II-AOut-4/16-B-TiCo

Der Ausgangs-Spannungsbereich der DAC ist fest auf ±10V bipolar eingestellt und lässt sich nicht verändern. Der Abgleich der Verstärkung und des Offset erfolgt per Software (siehe Kapitel 6 “Kalibrierung“).

Module mit D-Sub-Buchse haben einen Event-Eingang; ein anliegendes Signal kann als Triggersignal an das Prozessormodul geleitet werden.

TiCo-Prozessor

Die Varianten Pro II-AOut-4/16-xxx-TiCo besitzen zusätzlich einen frei programmierbaren *TiCo*-Prozessor mit 28KiByte Datenspeicher und 28KiByte Programmspeicher, der Zugriff auf alle Ausgänge des Moduls hat. Nähere Hinweise zur Anwendung und Programmierung des *TiCo*-Prozessors entnehmen Sie bitte dem Handbuch *TiCoBasic*.

Wenn Sie ein *TiCoBasic*-Programm im *TiCo*-Bootloader ablegen, wird das Programm beim Einschalten der Stromversorgung in den *TiCo*-Prozessor geladen und gestartet. Auf diese Weise kann das Modul eigenständig und unabhängig vom CPU-Modul des *ADwin-Pro II*-Systems arbeiten.

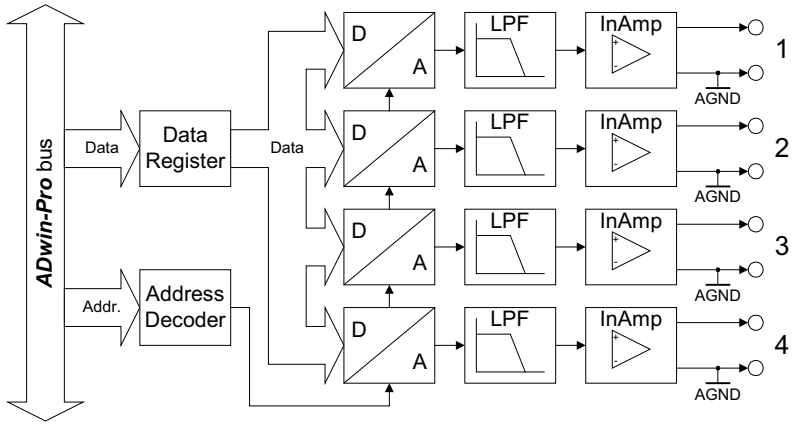


Abb. 78 – Pro II-AOut-4/16 Rev. E: Blockschaltbild

Ausgangskanäle	4 single ended
Auflösung	16 Bit
Einschwingzeit auf 0,01% FSR	< 3µs
Ausgangsspannung	±10V

Abb. 79 – Pro II-AOut-4/16 Rev. E: Spezifikation

Maximaler Ausgangsstrom		±5 mA pro Kanal für optimale Funktion ±35 mA technisch möglich, aber mit eingeschränkter Funktion kurzschlussfest
Genauigkeit	INL	±2 LSB typisch
	DNL	±1 LSB typisch
Offsetfehler		abgleichbar
Verstärkungsfehler		abgleichbar
Offsetdrift		±10 µV/°C
TiCo-Prozessor, je nach Variante		Prozessortyp: TiCo1 Taktfrequenz: 50 MHz Speichergröße: 28 KiB PM intern, 28 KiB DM intern
Steckerverbindung		4 LEMO-Buchsen einpolig oder 4LEMO-Buchsen zweipolig oder 37-polige D-Sub-Buchse oder 4 BNC-Buchsen

Abb. 79 – Pro II-AOut-4/16 Rev. E: Spezifikation

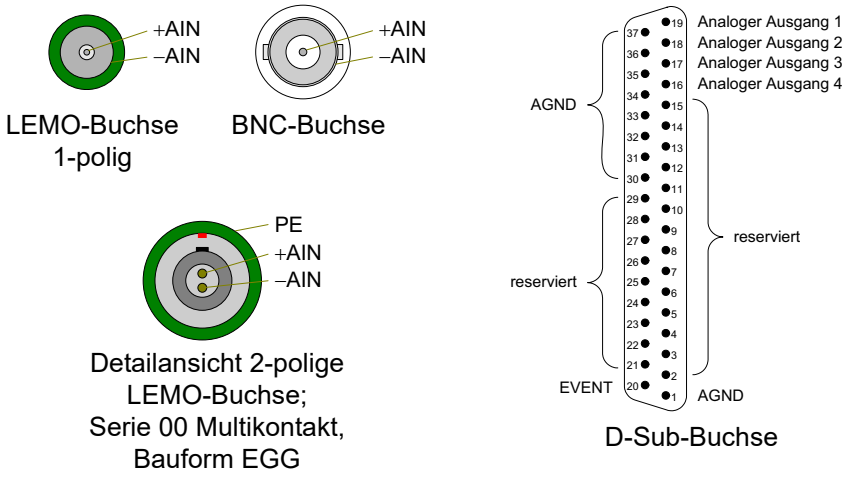


Abb. 80 – Pro II-AOut-4/16 Rev. E: Pinbelegung

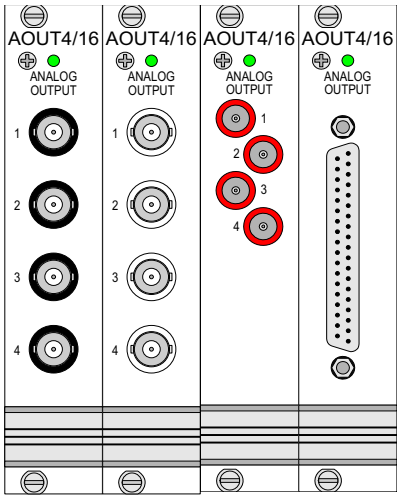


Abb. 81 – Pro II-AOut-4/16 Rev. E: Frontplatten

Programmierung

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Ausgabe durchführen	`P2_DAC`, `P2_DAC4`, `P2_DAC4_Packed`

Ausgabe schrittweise durchführen	`P2_Write_DAC`, `P2_Write_DAC4`
	`P2_Write_DAC4_Packed`
	`P2_Write_DAC32`
	`P2_Start_DAC`

Wandlung synchronisieren	`P2_Sync_All`, `P2_Sync_Enable` `P2_Sync_Stat`
LEDs einstellen	`P2_Check_LED`, `P2_Set_LED`
Interrupts und Event-Eingang einstellen	`P2_Event_Enable`, `P2_Event_Read` `P2_Event_Config`

Programmierung in TiCoBasic

Das Modul kann mit *TiCoBasic*-Befehlen programmiert werden. Die Befehle sind in der Online-Hilfe *TiCoBasic* erläutert.

Die Include-Datei `AOut_TiCo.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Ausgabe durchführen	`DAC`
Ausgabe schrittweise durchführen	`Write_DAC`, `Write_DAC32`, `Start_DAC`
LEDs einstellen	`Check_LED`, `Set_LED`

Programmierung TiCo-Zugriff

Für den Zugriff auf den *TiCo*-Prozessor von der ADwin CPU sind die folgenden *ADbasic*-Befehle in der Datei `ADwinPro_All.inc` definiert. Die Befehle sind im Handbuch *TiCoBasic* und in der Online-Hilfe *ADbasic* erläutert.

Bereich	Befehle

Datenaustausch mit dem *TiCo*-Prozessor über globale Variablen	`P2_TDrv_Init`
	`P2_GetData_Long`, `P2_Get_Par`, `P2_Get_Par_Block`
	`P2_SetData_Long`, `P2_Set_Par`, `P2_Set_Par_Block`
	`P2_Get_TiCo_RingBuffer`, `P2_Set_TiCo_RingBuffer`
	`P2_RingBuffer_Empty`
	`P2_RingBuffer_Full`

TiCo-Prozessor steuern	`P2_TiCo_Reset`, `P2_TiCo_Start`, `P2_TiCo_Stop`
	`P2_Get_TiCo_Bootloader_Status`
	`P2_Get_TiCo_Status`, `P2_Workload`

TiCo-Prozesse steuern	`P2_Process_Status`
	`P2_TiCo_Get_Processdelay`
	`P2_TiCo_Set_Processdelay`
	`P2_TiCo_Start_Process`
	`P2_Tico_Stop_Process`

TiCo-Programme übertragen	`P2_TiCo_Flash`, `P2_TiCo_Load`

5.6.2 Pro II-AOut-8/16 Rev. E

Das analoge Ausgangsmodul Pro II-AOut-8/16 Rev. E hat 8 DAC zu 16 Bit mit festem Tiefpass 1. Ordnung, um Störungen zu unterdrücken (f_g = 10MHz).

Das Modul kann in folgenden Varianten bestellt werden:

	ohne TiCo-Prozessor	mit TiCo-Prozessor
geschirmte LEMO-Buchsen 1-polig, CAMAC Europanorm	Pro II-AOut-8/16	Pro II-AOut-8/16-TiCo
geschirmte LEMO-Buchsen 2-polig, CAMAC Europanorm	Pro II-AOut-8/16-L2	Pro II-AOut-8/16-L2-TiCo
D-Sub-Buchse 37-polig	Pro II-AOut-8/16-D	Pro II-AOut-8/16-D-TiCo
BNC-Buchsen	Pro II-AOut-8/16-B	Pro II-AOut-8/16-B-TiCo

Der Ausgangs-Spannungsbereich der DAC ist fest auf ±10V bipolar eingestellt und lässt sich nicht verändern. Der Abgleich der Verstärkung und des Offset erfolgt per Software (siehe Kapitel 6 “Kalibrierung“).

Module mit D-Sub-Buchse haben einen Event-Eingang; ein anliegendes Signal kann als Triggersignal an das Prozessormodul geleitet werden.

TiCo-Prozessor

Die Varianten Pro II-AOut-8/16-xxx-TiCo besitzen zusätzlich einen frei programmierbaren *TiCo*-Prozessor mit 28KiByte Datenspeicher und 28KiByte Programmspeicher, der Zugriff auf alle Ausgänge des Moduls hat. Nähere Hinweise zur Anwendung und Programmierung des *TiCo*-Prozessors entnehmen Sie bitte dem Handbuch *TiCoBasic*.

Wenn Sie ein *TiCoBasic*-Programm im *TiCo*-Bootloader ablegen, wird das Programm beim Einschalten der Stromversorgung in den *TiCo*-Prozessor geladen und gestartet. Auf diese Weise kann das Modul eigenständig und unabhängig vom CPU-Modul des *ADwin-Pro II*-Systems arbeiten.

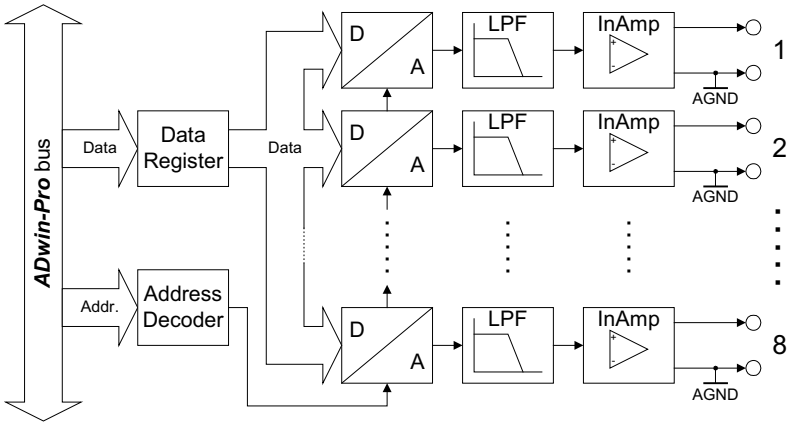


Abb. 82 – Pro II-AOut-8/16 Rev. E: Blockschaltbild

Ausgangskanäle	8 single ended
Auflösung	16 Bit
Einschwingzeit auf 0,01% FSR	< 3µs
Ausgangsspannung	±10V
Maximaler Ausgangsstrom	±5mA pro Kanal für optimale Funktion ±35mA technisch möglich, kurzschlussfest

Abb. 83 – Pro II-AOut-8/16 Rev. E: Spezifikation

Genauigkeit	INL	±2 LSB typisch
	DNL	±1 LSB typisch
Offsetfehler		abgleichbar
Verstärkungsfehler		abgleichbar
Offsetdrift		±10 µV/ºC
TiCo-Prozessor, je nach Variante		Prozessortyp: TiCo1 Taktfrequenz: 50 MHz Speichergröße: 28 KiB PM intern, 28 KiB DM intern
Steckerverbindung		8 LEMO-Buchsen einpolig oder 8 LEMO-Buchsen zweipolig oder 37-polige D-Sub-Buchse oder 8 BNC-Buchsen
Modulbreite		5 TE; mit BNC oder LEMO zweipolig: 10 TE

Abb. 83 – Pro II-AOut-8/16 Rev. E: Spezifikation

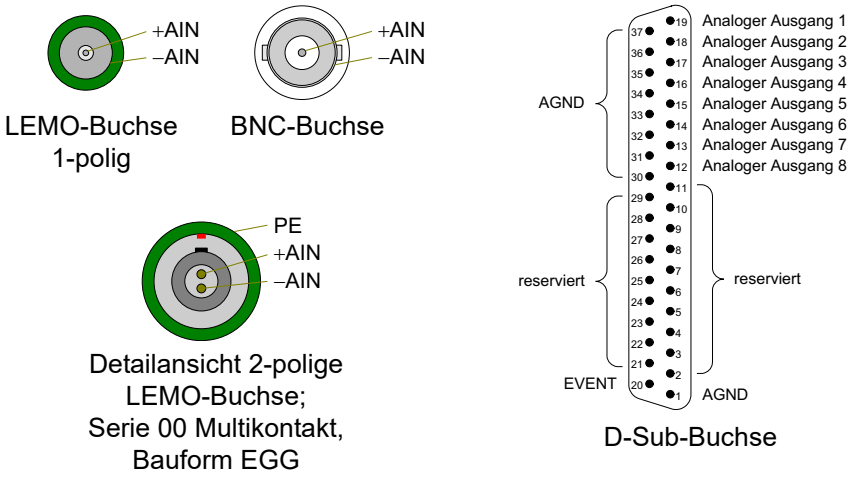


Abb. 84 – Pro II-AOut-8/16 Rev. E: Pinbelegung

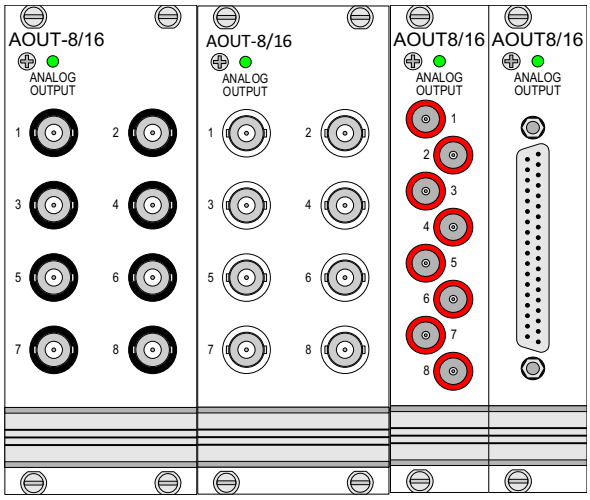


Abb. 85 – Pro II-AOut-8/16 Rev. E: Frontplatten

Programmierung

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Ausgabe durchführen	`P2_DAC`, `P2_DAC4`, `P2_DAC4_Packed` `P2_DAC8`, `P2_DAC8_Packed`

Ausgabe schrittweise durchführen	`P2_Write_DAC`, `P2_Write_DAC4`
	`P2_Write_DAC4_Packed`
	`P2_Write_DAC8`
	`P2_Write_DAC8_Packed`
	`P2_Write_DAC32`
	`P2_Start_DAC`

Wandlung synchronisieren	`P2_Sync_All`, `P2_Sync_Enable` `P2_Sync_Stat`
LEDs einstellen	`P2_Check_LED`, `P2_Set_LED`
Interrupts und Event-Eingang einstellen	`P2_Event_Enable`, `P2_Event_Read` `P2_Event_Config`

Programmierung in TiCoBasic

Das Modul kann mit *TiCoBasic*-Befehlen programmiert werden. Die Befehle sind in der Online-Hilfe *TiCoBasic* erläutert.

Die Include-Datei `AOut_TiCo.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Ausgabe durchführen	`DAC`
Ausgabe schrittweise durchführen	`Write_DAC`, `Write_DAC32`, `Start_DAC`
LEDs einstellen	`Check_LED`, `Set_LED`

Programmierung TiCo-Zugriff

Für den Zugriff auf den *TiCo*-Prozessor von der ADwin CPU sind die folgenden *ADbasic*-Befehle in der Datei `ADwinPro_All.inc` definiert. Die Befehle sind im Handbuch *TiCoBasic* und in der Online-Hilfe *ADbasic* erläutert.

Bereich	Befehle

Datenaustausch mit dem *TiCo*-Prozessor über globale Variablen	`P2_TDrv_Init`
	`P2_GetData_Long`, `P2_Get_Par`, `P2_Get_Par_Block`
	`P2_SetData_Long`, `P2_Set_Par`, `P2_Set_Par_Block`
	`P2_Get_TiCo_RingBuffer`,
	`P2_Set_TiCo_RingBuffer`
	`P2_RingBuffer_Empty`
	`P2_RingBuffer_Full`

TiCo-Prozessor steuern	`P2_TiCo_Reset`, `P2_TiCo_Start`, `P2_TiCo_Stop`
	`P2_Get_TiCo_Bootloader_Status`
	`P2_Get_TiCo_Status`, `P2_Workload`

TiCo-Prozesse steuern	`P2_Process_Status`
	`P2_TiCo_Get_Processdelay`
	`P2_TiCo_Set_Processdelay`
	`P2_TiCo_Start_Process`
	`P2_Tico_Stop_Process`

TiCo-Programme übertragen	`P2_TiCo_Flash`, `P2_TiCo_Load`

5.6.3 Pro II-AOut-1/16 Rev. E

Das Modul Pro II-AOut-1/16 Rev. E ist mit folgender Hardware ausgerüstet:

- 1 analoger Ausgang mit 16 Bit DAC, Ausgaberate 50 MHz, galvanisch getrennt.
- Je 16 digitale Eingangs- und Ausgangskanäle mit TTL-Pegeln
- 1 Event-Eingang
- *TiCo*-Prozessor mit 56 KiB internem Speicher und 256 MiB externem DRAM-Speicher

  Der *TiCo*-Prozessor hat Zugriff auf alle Ein- und Ausgänge des Moduls. Nähere Hinweise zur Anwendung und Programmierung des *TiCo*-Prozessors entnehmen Sie bitte dem Handbuch *TiCoBasic*.

Wenn der *TiCo*-Bootloader programmiert ist, kann das Modul eigenständig und unabhängig vom CPU-Modul des *ADwin-Pro II*-Systems arbeiten.

Analogausgang

Der DAC arbeitet mit 50 MHz. Der Ausgangs-Spannungsbereich des DAC ist auf ±2 V bipolar eingestellt und lässt sich nicht verändern. Der Abgleich der Verstärkung und des Offset erfolgt per Software (siehe Kapitel 6 “Kalibrierung“).

Der DAC-Ausgang liegt auf einer separaten SMB-Buchse und ist galvanisch von der Gehäusemasse und von anderen Modulen getrennt.

Beachten Sie: Der DAC-Ausgang hat einen Widerstand von 50 Ω und muss mit einem 50 Ω-System betrieben werden. Sobald der Ausgang kalibriert wurde, kann jede Änderung des Widerstandswerts – beispielsweise durch andere Kabel oder Steckverbindungen – zu starken Verfälschungen beim ausgegebenen Spannungswert führen.

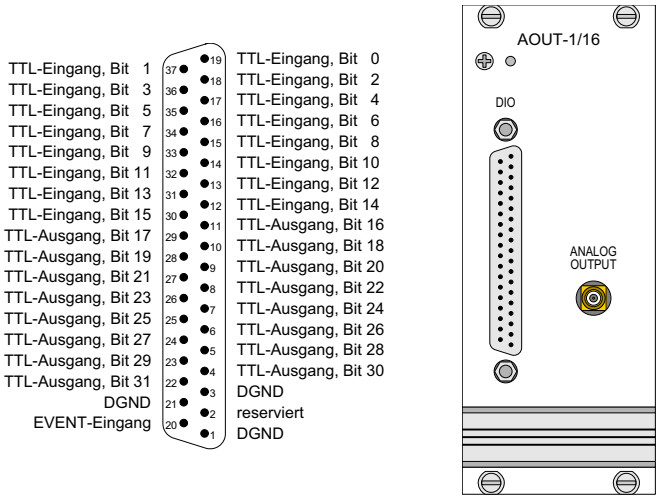


Abb. 86 – Pro II-AOut-1/16 Rev. E: Pinbelegung und Frontplatte

Für die Ausgabe von DAC-Werten gibt es 3 Varianten:

1. Einzelwertausgabe
2. Kontinuierliche Ausgabe über einen Ausgangs-Fifo. Die Ausgabewerte werden entweder regelmäßig nachgefüllt oder in einer Dauerschleife wiederholt.

   Mit jedem ausgegebenen Spannungswert können außerdem gleichzeitig 14 digitale Ausgänge gesetzt werden.

3. Ausgabe von Rampen, die über Startwert des DAC, Endwert des DAC und Zeitspanne definiert werden. Durch Kombination aufeinander folgender Rampen können Kurven ausgegeben werden.

   Zusätzlich zur Rampe können am Anfang und/oder am Ende der Rampe TTL-Signale auf den 16 Digitalausgängen ausgegeben werden.

Die 3 Ausgabevarianten können nur nacheinander, nicht gleichzeitig eingesetzt werden. Wenn Sie die Ausgabevariante wechseln, gibt in jedem Fall einen Zeitverzug, bis der nächste Spannungswert ausgegeben wird.

Digitale Ein-/Ausgänge

Auf einer 37-poligen D-Sub-Buchse stehen 16 Eingänge und 16 Ausgänge mit TTL-Pegeln zur Verfügung, Pinbelegung siehe oben. Die Kanalrichtung ist nicht umschaltbar.

Das Modul kann mit einer Frequenz von 100 MHz die Flanken an Eingangskanälen überwachen. Bei einer Änderung wird der aktuelle Pegelstand gemeinsam mit einem Zeitstempel in einem FIFO zwischengespeichert; es können bis zu 511 solcher Wertepaare (Pegelstand und Zeitstempel) gespeichert werden. Die FIFO-Daten können ausgelesen und weiter verarbeitet werden.
Außerdem kann abgefragt werden, an welchen Eingangskanälen eine positive oder negative Flanke aufgetreten ist.

Das FIFO kann entweder für die Flankenüberwachung von Eingängen oder für die Ausgabe von DAC-Werten und Pegeln an Digitalkanälen verwendet werden.

Über den Trigger-Eingang EVENT kann ein Signal (Trigger) einen Prozess auslösen, der dann sofort und vollständig abgearbeitet wird (siehe *ADbasic*-Handbuch).

TiCo-Prozessor

Das Modul besitzt einen frei programmierbaren *TiCo*-Prozessor mit 28 KiByte internem Programmspeicher, 28 KiByte internem Datenspeicher und 256 MiByte externem DRAM-Speicher. Sie programmieren den *TiCo*-Prozessor in *TiCoBasic*.

Der *TiCo*-Prozessor hat – wie auch die *ADwin*-CPU – Zugriff auf alle Ein- und Ausgangskanäle, analog wie digital. Nähere Hinweise zur Anwendung und Programmierung des *TiCo*-Prozessors entnehmen Sie bitte dem Handbuch *TiCoBasic*.

Wenn Sie ein *TiCoBasic*-Programm im *TiCo*-Bootloader ablegen, wird das Programm beim Einschalten der Stromversorgung in den *TiCo*-Prozessor geladen und gestartet. Auf diese Weise kann das Modul eigenständig und unabhängig vom CPU-Modul des *ADwin-Pro II*-Systems arbeiten.

Technische Daten

Analoger Ausgang	
Ausgangskanäle	1 single ended
Auflösung	16 Bit
Einschwingzeit auf 0,01% FSR	15 ns
Ausgangsspannung	±2 V
Maximaler Ausgangsstrom	±40 mA bei 50Ω Last kurzzeitig kurzschlussfest

Abb. 87 – Pro II-AOut-1/16 Rev. E: Spezifikation

Genauigkeit	INL	±8 LSB max.
	DNL	±1 LSB typisch
Offsetfehler		abgleichbar
Verstärkungsfehler		abgleichbar
Offsetdrift		±10 µV/°K
Digitale Ein-/Ausgänge		
Digitale Ein-/Ausgänge		16 Eingänge und 16 Ausgänge mit TTL-Logik
Pull-Down-Widerstand		10 kΩ
V_{IH}		min. 2 V
V_{IL}		max. 0,8 V
I_{IH}		max. 500 µA
I_{IL}		max. 10 µA
Spannungsbereich		-0,5 V … +5,5 V
Ausgangsstrom		max. ±24 mA pro Kanal über V_{CC} oder GND, max. ±70 mA je Block (8 Kanäle) über VCC oder GND
Event-Eingang		TTL-Logik
Allgemein		
Ein-/Ausgangs-Fifo		Größe: 511 Wertepaare Frequenz: 100 MHz
TiCo-Prozessor		Prozessortyp: TiCo1 Taktfrequenz: 50 MHz Speichergröße: 28 KiB PM intern, 28 KiB DM intern, 256 MiByte DRAM extern
Steckerverbindung		37-polige D-Sub-Buchse, SMB-Buchse
Modulbreite		10 TE

Abb. 87 – Pro II-AOut-1/16 Rev. E: Spezifikation

Programmierung

Programmierung in ADbasic

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Analoger Ausgang	
Einzelwert ausgeben	`P2_DAC`, `P2_DAC1_DIO`
Ausgabe schrittweise durchführen	`P2_Write_DAC`, `P2_Start_DAC`

Ausgabe des DAC-Werts und eines digitalen Bitmusters über Ausgangs-Fifo	`P2_Dig_Fifo_Mode`
	`P2_Digout_Fifo_Clear`
	`P2_Digout_Fifo_Read_Timer`
	`P2_Digout_Fifo_Empty`
	`P2_Digout_Fifo_Start`
	`P2_Digout_Fifo_Enable`
	`P2_Digout_Fifo_Write`

Bereich	Befehle
Rampenmodus	P2_DAC_Ramp_Write P2_DAC_Ramp_Status P2_DAC_Ramp_Buffer_Free P2_DAC_Ramp_Stop
Digitale Ein-/Ausgänge	
Eingangssignale abfragen	P2_Digin_Long
Abfrage über Eingangs-Fifo	P2_Digin_Fifo_Clear P2_Digin_Fifo_Read P2_Digin_Fifo_Read_Fast P2_Digin_Fifo_Read_Timer P2_Digin_Fifo_Full P2_Digin_Fifo_Enable
Ausgangssignale setzen	P2_Digout, P2_Digout_Long P2_Get_Digout_Long P2_Digout_Bits P2_Digout_Set, P2_Digout_Reset
Latch-Register nutzen	P2_Dig_Latch P2_Dig_Read_Latch P2_Dig_Write_Latch
Allgemein	
Aktionen synchron starten	P2_Sync_All, P2_Sync_Enable P2_Sync_Stat
LEDs einstellen	P2_Check_LED, P2_Set_LED
Interrupts und Event-Eingang einstellen	P2_Event_Enable, P2_Event_Read P2_Event_Config

Programmierung in TiCoBasic

Das Modul kann mit *TiCoBasic*-Befehlen programmiert werden. Die Befehle sind in der Online-Hilfe *TiCoBasic* erläutert.

Die Include-Datei `AOUT1_TiCo.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Analoger Ausgang	
Einzelwert ausgeben	DAC, DAC1_DIO
Ausgabe schrittweise durchführen	Write_DAC, Start_DAC
Ausgabe über Ausgangs-Fifo	Digout_Fifo_Clear Digout_Fifo_Read_Timer Digout_Fifo_Empty Dig_Fifo_Mode Digout_Fifo_Start Digout_Fifo_Enable Digout_Fifo_Write
Rampenmodus	DAC_Ramp_Write DAC_Ramp_Status DAC_Ramp_Buffer_Free DAC_Ramp_Stop
Digitale Ein-/Ausgänge	
Eingangssignale abfragen	Digin_Long
Abfrage über Eingangs-Fifo	Digin_Fifo_Clear Digin_Fifo_Read_Timer Digin_Fifo_Full Digin_Fifo_Enable

Bereich	Befehle

Ausgangssignale setzen	`Digout`, `Digout_Long`
	`Get_Digout_Long`
	`Digout_Bits`
	`Digout_Set`, `Digout_Reset`

Latch-Register nutzen	`Dig_Latch`
	`Dig_Read_Latch`
	`Dig_Write_Latch`

Allgemein	
LEDs einstellen	`Check_LED`, `Set_LED`
Interrupts und Event-Eingang einstellen	`Event_Enable`, `Event_Read` `Event_Config`

Programmierung TiCo-Zugriff

Für den Zugriff auf den *TiCo*-Prozessor von der ADwin CPU sind die folgenden *ADbasic*-Befehle in der Datei `ADwinPro_All.inc` definiert. Die Befehle sind im Handbuch *TiCoBasic* und in der Online-Hilfe *ADbasic* erläutert.

Bereich	Befehle

Datenaustausch mit dem *TiCo*-Prozessor über globale Variablen	`P2_TDrv_Init`
	`P2_GetData_Long`, `P2_Get_Par`, `P2_Get_Par_Block`
	`P2_SetData_Long`, `P2_Set_Par`, `P2_Set_Par_Block`
	`P2_Get_TiCo_RingBuffer`, `P2_Set_TiCo_RingBuffer`
	`P2_RingBuffer_Empty`
	`P2_RingBuffer_Full`

TiCo-Prozessor steuern	`P2_TiCo_Reset`, `P2_TiCo_Start`, `P2_TiCo_Stop`
	`P2_Get_TiCo_Bootloader_Status`
	`P2_Get_TiCo_Status`, `P2_Workload`

TiCo-Prozesse steuern	`P2_Process_Status`
	`P2_TiCo_Get_Processdelay`
	`P2_TiCo_Set_Processdelay`
	`P2_TiCo_Start_Process`
	`P2_Tico_Stop_Process`

TiCo-Programme übertragen	`P2_TiCo_Flash`, `P2_TiCo_Load`

5.7 Pro II: Digital-IO-Module

Digital-IO-Module

Modulname	Rev.	Typ	Kanäle	U_{Ein} [V]		High Pegel [mA]	Isolation [V]	Seite
DIO-32	E	TTL-Ein-/Ausgang	32	5	TTL	–	–	106
DIO-32-TiCo	E	TTL-Ein-/Ausgang, mit TiCo1-Prozessor	32	5	TTL	–	–	108
DIO-32/1-TiCo	E	TTL-Ein-/Ausgang einzeln einstellbar, mit TiCo1-Prozessor	32	5	TTL	–	–	112
DIO-32-TiCo2	E	TTL-Ein-/Ausgang, mit TiCo2-Prozessor	32	5	TTL	–	–	116
DIO-8-D12	E	8 TTL-Kanäle, 12 diff. Kanäle mit TiCo2-Prozessor	20	5	TTL	–	–	120
OPT-16	E	Optokoppler-Eingang	16	5, 12, 24	DC	–	42	124
OPT-32-24V	E	Optokoppler-Eingang	32	24	DC	–	42	127
REL-16	E	Relais-Ausgang	16	max. 30	AC / DC	500	42	129
TRA-16	E	Transistor-Ausgang	16	5…30	DC	200	42	131
PWM-16	E	PWM-Ausgangssignal	16	5	TTL	–	–	133
PWM-16-I	E	PWM-Ausgangssignal	16	5…30	DC	100	42	133
COMP-16	E	Digitaleingänge mit Komparatoren	16	-2…32	DC	–	–	136
MIO-D12	E	Transistor-Ausgänge	12	5…30	DC	200		45
		Optokoppler-Eingänge	12 s.e.	5, 12, 24	DC	–		
		Zählerblock	2	Universal, 32 Bit, 5V 1 diff., 1 über Optokoppler				
		SSI-Decoder	1	max. 12,5MHz				
		TiCo1-Prozessor	–	56 KiByte interner Speicher				

Zähler-Module

Modulname	Rev.	Kanäle	Zähler			Eingangsspg. U_{Ein}		Isolation [V]	Seite
			Anzahl	Typ	Auflösg. [Bit]	[V]	Typ		
CNT-T	E	4, mit TiCo1-Prozessor	1	U	32	5	TTL	–	140
CNT-I	E	4, mit TiCo1-Prozessor	1	U	32	5, 12, 24	DC	42	140
CNT-D	E	4 + 2 SSI, mit TiCo1-Prozessor	1	U	32	5 diff.	RS422/ RS485	–	140

Der Zählertyp „Universalzähler“ (U) beinhaltet Vor-/Rückwärtszähler, Vierflankenauswertung und PWM-Zähler.

5.7.1 Pro II-DIO-32 Rev. E

Das digitale Ein-/Ausgangsmodul Pro II-DIO-32 Rev. E stellt 32 programmierbare Ein- und Ausgangskanäle mit TTL-Pegeln bereit. Die Kanäle können in Blöcken zu jeweils 8 Bit mit *ADbasic*-Befehlen als Ein- oder Ausgänge konfiguriert werden. Nach dem Einschalten sind alle Kanäle als Eingänge konfiguriert.

Das Modul kann mit einer Frequenz von 100 MHz die Flanken an Eingangskanälen überwachen. Bei einer Änderung wird der aktuelle Pegelstand gemeinsam mit einem Zeitstempel in einem FIFO zwischengespeichert; es können bis zu 511 solcher Wertepaare (Pegelstand und Zeitstempel) gespeichert werden. Die FIFO-Daten können ausgelesen und weiter verarbeitet werden.
Außerdem kann abgefragt werden, an welchen Eingangskanälen eine positive oder negative Flanke aufgetreten ist.

Die Variante Pro II-DIO-32-TiCo Rev. E besitzt zusätzlich einen frei programmierbaren *TiCo*-Prozessor und kann selbstständig Pegel zu bestimmten Zeitpunkten auf Digitalausgängen ausgeben. Weiteres siehe Seite 108.

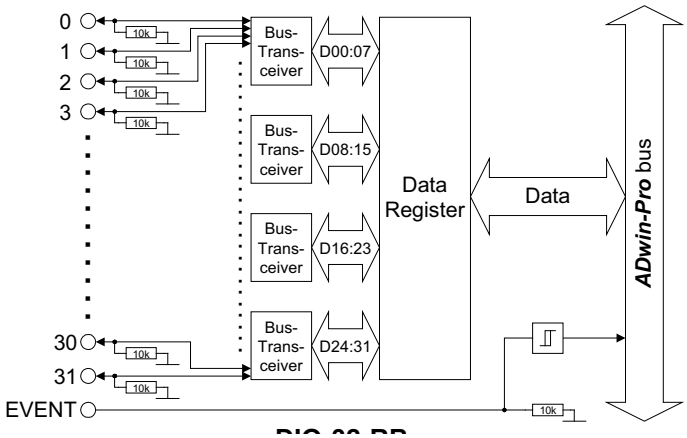


Abb. 88 – Pro II-DIO-32 Rev. E: Blockschaltbild

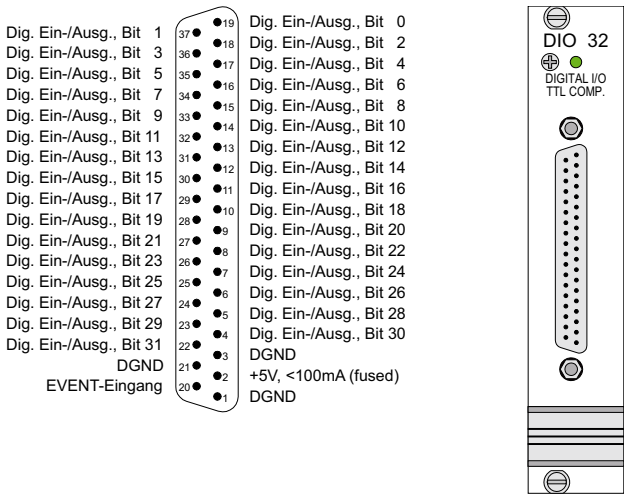


Abb. 89 – Pro II-DIO-32 Rev. E: Frontplatte und Pinbelegung

Ein-/Ausgangskanäle	32; in Blöcken zu 8 Bit als Ein-/Ausgang mittels Software einstellbar
Digitale Eingänge	TTL-Logik
Pull-Down-Widerstand	10 kΩ

Abb. 90 – Pro II-DIO-32 Rev. E: Spezifikation

V_{IH}	min. 2V
V_{IL}	max. 0,8V
I_{IH}	max. 1µA
I_{IL}	max. 0,01mA
Spannungsbereich	-0,5V … +5,5V
Ausgangsstrom	max. ±35mA pro Kanal, max. ±70mA je Block (8 Kanäle) über VCC oder GND
Event-Eingang	TTL-Logik
Power-Up-Status	Alle Kanäle als Eingänge
Steckerverbindung	37-polige D-Sub-Buchse

Abb. 90 – Pro II-DIO-32 Rev. E: Spezifikation

Programmierung

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Ein- und Ausgänge konfigurieren	`P2_DigProg`
Eingangssignale abfragen	`P2_Digin_Long`

Latch-Register nutzen	`P2_Dig_Latch`
	`P2_Dig_Read_Latch`
	`P2_Dig_Write_Latch`

Flanken an Eingangskanälen überwachen	`P2_Digin_FIFO_Enable`
	`P2_Digin_FIFO_Read`
	`P2_Digin_FIFO_Read_Fast`
	`P2_Digin_FIFO_Read_Timer`
	`P2_Digin_FIFO_Clear`
	`P2_Digin_FIFO_Full`

Flankenstatus abfragen	`P2_Digin_Edge`

Ausgangssignale setzen und rücklesen	`P2_Digout`, `P2_Digout_Bits`
	`P2_Digout_Long`
	`P2_Get_Digout_Long`

Abläufe synchronisieren	`P2_Sync_All`, `P2_Sync_Enable` `P2_Sync_Stat`
LEDs einstellen	`P2_Check_LED`, `P2_Set_LED`
Interrupts und Event-Eingang einstellen	`P2_Event_Enable`, `P2_Event_Config` `P2_Event_Read`

5.7.2 Pro II-DIO-32-TiCo Rev. E

Das digitale Ein-/Ausgangsmodul Pro II-DIO-32-TiCo Rev. E stellt 32 programmierbare Ein- und Ausgangskanäle mit TTL-Pegeln bereit. Die Kanäle können in Blöcken zu jeweils 8 Bit mit *ADbasic*-Befehlen als Ein- oder Ausgänge konfiguriert werden. Nach dem Einschalten sind alle Kanäle als Eingänge konfiguriert.

Flankenüberwachung

Das Modul kann mit einer Frequenz von 100 MHz die Flanken an Eingangskanälen überwachen. Bei einer Änderung wird der aktuelle Pegelstand gemeinsam mit einem Zeitstempel in einem FIFO zwischengespeichert; es können bis zu 511 solcher Wertepaare (Pegelstand und Zeitstempel) gespeichert werden. Die FIFO-Daten können ausgelesen und weiter verarbeitet werden.
Außerdem kann abgefragt werden, an welchen Eingangskanälen eine positive oder negative Flanke aufgetreten ist.

Spike-Filter

An den Eingangskanälen können einzelne Fehlpulse (Spikes) mit einem einstellbaren Filter unterdrückt werden. Jeder Kanal hat seinen eigenen Filter, aber die Filtereinstellungen gelten für alle Kanäle gleichermaßen. Nach dem Einschalten sind die Filter deaktiviert.

TiCo-Prozessor

Im Unterschied zur Variante Pro II-DIO-32 Rev. E besitzt das Modul zusätzlich einen frei programmierbaren *TiCo*-Prozessor mit 28 kiB Programmspeicher, 28 kiB Datenspeicher und 256 MiB externem Speicher. Der *TiCo*-Prozessor hat Zugriff auf alle digitalen Ein- und Ausgangskanäle. Nähere Hinweise zur Anwendung und Programmierung des *TiCo*-Prozessors entnehmen Sie bitte dem Handbuch *TiCoBasic*.

Wenn Sie ein *TiCoBasic*-Programm im *TiCo*-Bootloader ablegen, wird das Programm beim Einschalten der Stromversorgung in den *TiCo*-Prozessor geladen und gestartet. Auf diese Weise kann das Modul eigenständig und unabhängig vom CPU-Modul des *ADwin-Pro II*-Systems arbeiten.

Zeitgesteuerte Pegelausgabe

Ab Revision E 03 kann das Modul selbstständig Pegel zu bestimmten Zeitpunkten auf Digitalausgängen ausgeben. Ein FIFO dient als Zwischenspeicher für die vom Benutzer festgelegten Pegel und Zeitpunkte.
Das FIFO kann entweder für die Flankenüberwachung von Eingängen oder für die Pegelausgabe an Ausgängen verwendet werden.

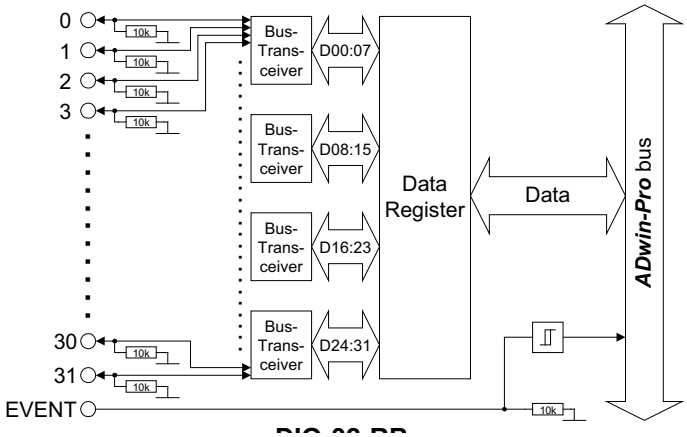


Abb. 91 – Pro II-DIO-32-TiCo Rev. E: Blockschaltbild

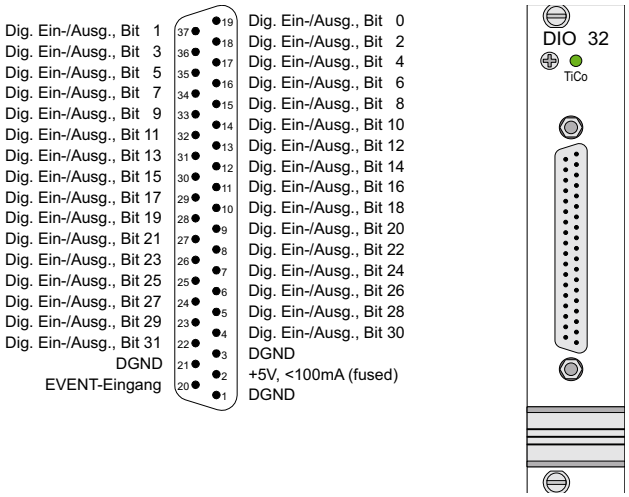


Abb. 92 – Pro II-DIO-32-TiCo Rev. E: Frontplatte und Pinbelegung

Ein-/Ausgangskanäle	32; in Blöcken zu 8 Bit als Ein-/Ausgang mittels Software einstellbar
Digitale Eingänge	TTL-Logik
Pull-Down-Widerstand	10 kΩ
V_{IH}	min. 2 V
V_{IL}	max. 0,8 V
I_{IH}	max. 1 µA
I_{IL}	max. 0,01 mA
Spannungsbereich	-0,5 V … +5,5 V
Ausgangsstrom	max. ±35 mA pro Kanal, max. ±70 mA je Block (8 Kanäle) über VCC oder GND
Event-Eingang	TTL-Logik
Power-Up-Status	Alle Kanäle als Eingänge
Eingangs- / Ausgangs-Fifo	Größe: 511 Wertepaare Frequenz: 100 MHz
TiCo	Prozessortyp: TiCo1 Taktfrequenz: 50 MHz Speichergröße: 28 KiB PM intern, 28 KiB DM intern, 256 MiByte DRAM extern
Steckerverbindung	37-polige D-Sub-Buchse

Abb. 93 – Pro II-DIO-32-TiCo Rev. E: Spezifikation

Programmierung

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Ein- und Ausgänge konfigurieren	`P2_DigProg` `P2_Dig_FIFO_Mode`
Eingangssignale abfragen	`P2_Digin_Long`

Latch-Register nutzen	`P2_Dig_Latch`
	`P2_Dig_Read_Latch`
	`P2_Dig_Write_Latch`

Bereich	Befehle
Flanken an Eingangskanälen überwachen	P2_Digin_FIFO_Enable P2_Digin_FIFO_Read P2_Digin_FIFO_Read_Fast P2_Digin_FIFO_Read_Timer P2_Digin_FIFO_Clear P2_Digin_FIFO_Full
Spike-Filter setzen	P2_Digin_Filter_Init
Flankenstatus abfragen	P2_Digin_Edge
Ausgangssignale setzen und rücklesen	P2_Digout, P2_Digout_Bits P2_Digout_Long P2_Digout_Set P2_Digout_Reset P2_Get_Digout_Long
Ausgangssignale automatisch setzen	P2_Digout_FIFO_Clear P2_Digout_FIFO_Empty P2_Digout_FIFO_Enable P2_Digout_FIFO_Read_Timer P2_Digout_FIFO_Start P2_Digout_FIFO_Write
Abläufe synchronisieren	P2_Sync_All, P2_Sync_Enable P2_Sync_Stat
LEDs einstellen	P2_Check_LED, P2_Set_LED
Interrupts und Event-Eingang einstellen	P2_Event_Enable, P2_Event_Config P2_Event_Read

Programmierung in TiCoBasic

Das Modul kann mit *TiCoBasic*-Befehlen programmiert werden. Die Befehle sind in der Online-Hilfe *TiCoBasic* erläutert.

Im Verzeichnis `C:\ADwin\TiCoBasic\samples_ADwin_ProII` finden Sie zusätzliche Beispielprogramme.

Die Include-Datei `DIO32TiCo.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Ein- und Ausgänge konfigurieren	DigProg
Eingangssignale abfragen	Digin_Long
Flanken an Eingangskanälen überwachen	Digin_FIFO_Enable Digin_FIFO_Read Digin_FIFO_Read_Timer Digin_FIFO_Clear Digin_FIFO_Full
Spike-Filter setzen	Digin_Filter_Init
Flankenstatus abfragen	Digin_Edge
Ausgangssignale setzen und rücklesen	Digout, Digout_Bits Digout_Set, Digout_Reset Digout_Long Get_Digout_Long
Ausgangssignale automatisch setzen	Digout_FIFO_Clear Digout_FIFO_Empty Digout_FIFO_Enable Digout_FIFO_Read_Timer Digout_FIFO_Start Digout_FIFO_Write

Bereich	Befehle
LEDs einstellen	`Check_LED`, `Set_LED`
Interrupts und Event-Eingang einstellen	`Event_Enable`, `Trigger_Event` `Event_Config`

Programmierung TiCo-Zugriff

Für den Zugriff auf den *TiCo*-Prozessor von der ADwin CPU sind die folgenden *ADbasic*-Befehle in der Datei `ADwinPro_All.inc` definiert. Die Befehle sind im Handbuch *TiCoBasic* und in der Online-Hilfe *ADbasic* erläutert.

Bereich	Befehle

Datenaustausch mit dem *TiCo*-Prozessor über globale Variablen	`P2_TDrv_Init`
	`P2_GetData_Long`, `P2_Get_Par`, `P2_Get_Par_Block`
	`P2_SetData_Long`, `P2_Set_Par`, `P2_Set_Par_Block`
	`P2_Get_TiCo_RingBuffer`, `P2_Set_TiCo_RingBuffer`
	`P2_RingBuffer_Empty`
	`P2_RingBuffer_Full`

TiCo-Prozessor steuern	`P2_TiCo_Reset`, `P2_TiCo_Start`, `P2_TiCo_Stop`
	`P2_Get_TiCo_Bootloader_Status`
	`P2_Get_TiCo_Status`, `P2_Workload`

TiCo-Prozesse steuern	`P2_Process_Status`
	`P2_TiCo_Get_Processdelay`
	`P2_TiCo_Set_Processdelay`
	`P2_TiCo_Start_Process`
	`P2_Tico_Stop_Process`

TiCo-Programme übertragen	`P2_TiCo_Flash`, `P2_TiCo_Load`

5.7.3 Pro II-DIO-32/1-TiCo Rev. E

Das digitale Ein-/Ausgangsmodul Pro II-DIO-32/1-TiCo Rev. E stellt 32 programmierbare Ein- und Ausgangskanäle mit TTL-Pegeln bereit. Die Kanäle können einzeln mit *ADbasic*-Befehlen als Ein- oder Ausgänge konfiguriert werden. Nach dem Einschalten sind alle Kanäle als Eingänge konfiguriert.

Flankenüberwachung

Das Modul kann mit einer Frequenz von 100 MHz die Flanken an Eingangskanälen überwachen. Bei einer Änderung wird der aktuelle Pegelstand gemeinsam mit einem Zeitstempel in einem FIFO zwischengespeichert; es können bis zu 511 solcher Wertepaare (Pegelstand und Zeitstempel) gespeichert werden. Die FIFO-Daten können ausgelesen und weiter verarbeitet werden.
Außerdem kann abgefragt werden, an welchen Eingangskanälen eine positive oder negative Flanke aufgetreten ist.

Spike-Filter

An den Eingangskanälen können einzelne Fehlpulse (Spikes) mit einem einstellbaren Filter unterdrückt werden. Jeder Kanal hat seinen eigenen Filter, aber die Filtereinstellungen gelten für alle Kanäle gleichermaßen. Nach dem Einschalten sind die Filter deaktiviert.

TiCo-Prozessor

Das Modul besitzt zusätzlich einen frei programmierbaren *TiCo*-Prozessor mit 28 kiB Programmspeicher und 28 kiB Datenspeicher. Der *TiCo*-Prozessor hat Zugriff auf alle digitalen Ein- und Ausgangskanäle. Nähere Hinweise zur Anwendung und Programmierung des *TiCo*-Prozessors entnehmen Sie bitte dem Handbuch *TiCoBasic*.

Wenn Sie ein *TiCoBasic*-Programm im *TiCo*-Bootloader ablegen, wird das Programm beim Einschalten der Stromversorgung in den *TiCo*-Prozessor geladen und gestartet. Auf diese Weise kann das Modul eigenständig und unabhängig vom CPU-Modul des *ADwin-Pro II*-Systems arbeiten.

Zeitgesteuerte Pegelausgabe

Das Modul kann selbstständig Pegel zu bestimmten Zeitpunkten auf Digitalausgängen ausgeben. Ein FIFO dient als Zwischenspeicher für die vom Benutzer festgelegten Pegel und Zeitpunkte.
Das FIFO kann entweder für die Flankenüberwachung von Eingängen oder für die Pegelausgabe an Ausgängen verwendet werden.

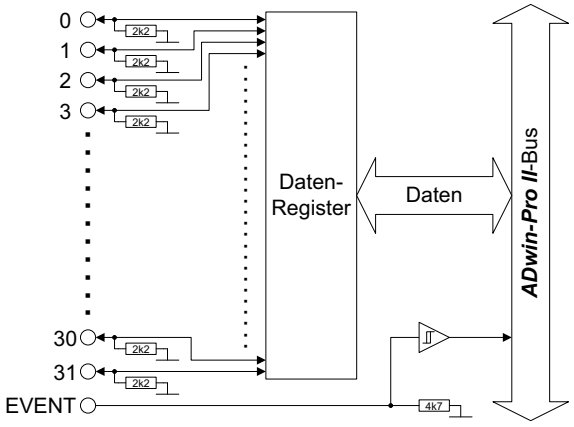


Abb. 94 – Pro II-DIO-32/1-TiCo Rev. E: Blockschaltbild

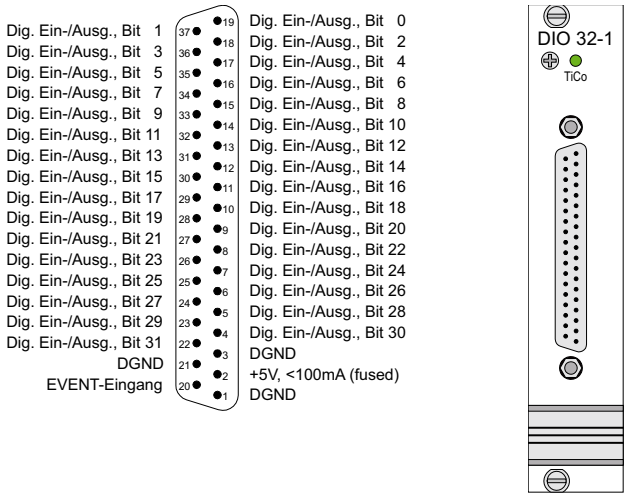


Abb. 95 – Pro II-DIO-32/1-TiCo Rev. E: Frontplatte und Pinbelegung

Ein-/Ausgangskanäle	32; einzeln als Ein-/Ausgang mittels Software einstellbar
Digitale Eingänge	TTL-Logik
Pull-Down-Widerstand	10 kΩ
V_{IH}	min. 3,5 V
V_{IL}	max. 1,5 V
I_{IH}	max. 1 µA
I_{IL}	max. 0,01 mA
Spannungsbereich	-0,5 V … +5,5 V
Ausgangsstrom	max. ±32 mA pro Kanal über VCC oder GND
Event-Eingang	TTL-Logik
Power-Up-Status	Alle Kanäle als Eingänge
Eingangs- / Ausgangs-Fifo	Größe: 511 Wertepaare Frequenz: 100 MHz
TiCo	Prozessortyp: TiCo1 Taktfrequenz: 50 MHz Speichergröße: 28 KiB PM intern, 28 KiB DM intern
Steckerverbindung	37-polige D-Sub-Buchse

Abb. 96 – : Spezifikation

Programmierung

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle

Ein- und Ausgänge konfigurieren	`P2_DigProg`
	`P2_DigProg_Bits`
	`P2_Dig_FIFO_Mode`

Eingangssignale abfragen	`P2_Digin_Long`

Latch-Register nutzen	`P2_Dig_Latch`
	`P2_Dig_Read_Latch`
	`P2_Dig_Write_Latch`

Bereich	Befehle
Flanken an Eingangskanälen überwachen	P2_Digin_FIFO_Enable P2_Digin_FIFO_Read P2_Digin_FIFO_Read_Fast P2_Digin_FIFO_Read_Timer P2_Digin_FIFO_Clear P2_Digin_FIFO_Full
Spike-Filter setzen	P2_Digin_Filter_Init
Flankenstatus abfragen	P2_Digin_Edge
Ausgangssignale setzen und rücklesen	P2_Digout, P2_Digout_Bits P2_Digout_Long P2_Digout_Set P2_Digout_Reset P2_Get_Digout_Long
Ausgangssignale automatisch setzen	P2_Digout_FIFO_Clear P2_Digout_FIFO_Empty P2_Digout_FIFO_Enable P2_Digout_FIFO_Read_Timer P2_Digout_FIFO_Start P2_Digout_FIFO_Write
Abläufe synchronisieren	P2_Sync_All, P2_Sync_Enable P2_Sync_Stat
LEDs einstellen	P2_Check_LED, P2_Set_LED
Interrupts und Event-Eingang einstellen	P2_Event_Enable, P2_Event_Config P2_Event_Read

Programmierung in TiCoBasic

Das Modul kann mit *TiCoBasic*-Befehlen programmiert werden. Die Befehle sind in der Online-Hilfe *TiCoBasic* erläutert.

Im Verzeichnis `C:\ADwin\TiCoBasic\samples_ADwin_ProII` finden Sie zusätzliche Beispielprogramme.

Die Include-Datei `DIO32TiCo.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Ein- und Ausgänge konfigurieren	DigProg, DigProg_Bits Dig_Fifo_Mode
Eingangssignale abfragen	Digin_Long
Flanken an Eingangskanälen überwachen	Digin_FIFO_Enable Digin_FIFO_Read Digin_FIFO_Read_Timer Digin_FIFO_Clear Digin_FIFO_Full
Spike-Filter setzen	Digin_Filter_Init
Flankenstatus abfragen	Digin_Edge
Ausgangssignale setzen und rücklesen	Digout, Digout_Bits Digout_Set, Digout_Reset Digout_Long Get_Digout_Long

Bereich	Befehle
Ausgangssignale automatisch setzen	Digout_FIFO_Clear Digout_FIFO_Empty Digout_FIFO_Enable Digout_FIFO_Read_Timer Digout_FIFO_Start Digout_FIFO_Write Digout_FIFO_Write_Burst
LEDs einstellen	Check_LED, Set_LED
Interrupts und Event-Eingang einstellen	Event_Enable, Trigger_Event Event_Config

Programmierung TiCo-Zugriff

Für den Zugriff auf den *TiCo*-Prozessor von der ADwin CPU sind die folgenden *ADbasic*-Befehle in der Datei `ADwinPro_All.inc` definiert. Die Befehle sind im Handbuch *TiCoBasic* und in der Online-Hilfe *ADbasic* erläutert.

Bereich	Befehle
Datenaustausch mit dem *TiCo*-Prozessor über globale Variablen	P2_TDrv_Init P2_GetData_Long, P2_Get_Par, P2_Get_Par_Block P2_SetData_Long, P2_Set_Par, P2_Set_Par_Block P2_Get_TiCo_RingBuffer, P2_Set_TiCo_RingBuffer P2_RingBuffer_Empty P2_RingBuffer_Full
TiCo-Prozessor steuern	P2_TiCo_Reset, P2_TiCo_Start, P2_TiCo_Stop P2_Get_TiCo_Bootloader_ Status P2_Get_TiCo_Status, P2_Workload
TiCo-Prozesse steuern	P2_Process_Status P2_TiCo_Get_Processdelay P2_TiCo_Set_Processdelay P2_TiCo_Start_Process P2_Tico_Stop_Process
TiCo-Programme übertragen	P2_TiCo_Flash, P2_TiCo_Load

5.7.4 Pro II-DIO-32-TiCo2 Rev. E

Das digitale Ein-/Ausgangsmodul Pro II-DIO-32-TiCo2 Rev. E stellt 32 programmierbare Ein- und Ausgangskanäle mit einstellbaren Spannungspegeln bereit. Die Kanäle können in Blöcken zu jeweils 8 Bit mit *ADbasic*-Befehlen als Ein- oder Ausgänge konfiguriert werden. Nach dem Einschalten sind alle Kanäle als Eingänge konfiguriert.

Die Nenn-Spannungspegel der Digitalkanäle können im Bereich von 1,6V … 4,7V für Gruppen von je 8 Kanälen per Software eingestellt werden. Die Tabelle zeigt typische Spannungspegel mit den zugehörigen Schaltschwellen und Ausgangsströmen.

Spannungspegel	$V_{IL\ max}$	$V_{IH\ min}$	I_{OL}
1,65V	0,6V	1,1V	4mA
2,5V	0,7V	1,7V	8mA
3,3V	0,8V	2,0V	24mA
4,7V	1,4V	3,3V	32mA

Abb. 97 – Pro II-DIO-32-TiCo2 Rev. E: Typische Spannungspegel

Der Spannungspegel des Event-Eingangs arbeitet mit TTL-Logik. Der Pegel ist nicht einstellbar.

Flankenüberwachung

Das Modul kann mit einer Frequenz von 200MHz die Flanken an Eingangskanälen überwachen. Bei einer Änderung wird der aktuelle Pegelstand gemeinsam mit einem Zeitstempel in einem FIFO zwischengespeichert; es können bis zu 2048 solcher Wertepaare (Pegelstand und Zeitstempel) gespeichert werden. Die FIFO-Daten können ausgelesen und weiter verarbeitet werden.
Außerdem kann abgefragt werden, an welchen Eingangskanälen eine positive oder negative Flanke aufgetreten ist.

Spike-Filter

An den Eingangskanälen können einzelne Fehlpulse (Spikes) mit einem einstellbaren Filter unterdrückt werden. Jeder Kanal hat seinen eigenen Filter, aber die Filtereinstellungen gelten für alle Kanäle gleichermaßen. Nach dem Einschalten sind die Filter deaktiviert.

TiCo-Prozessor

Im Unterschied zur Variante Pro II-DIO-32 Rev. E besitzt das Modul zusätzlich einen frei programmierbaren *TiCo*-Prozessor (Typ TiCo2) mit 128kiB internem Programmspeicher und 512kiB internem Datenspeicher. Der *TiCo*-Prozessor hat Zugriff auf alle digitalen Ein- und Ausgangskanäle. Nähere Hinweise zur Anwendung und Programmierung des *TiCo*-Prozessors entnehmen Sie bitte dem Handbuch *TiCoBasic*.

Wenn Sie ein *TiCoBasic*-Programm im *TiCo*-Bootloader ablegen, wird das Programm beim Einschalten der Stromversorgung in den *TiCo*-Prozessor geladen und gestartet. Auf diese Weise kann das Modul eigenständig und unabhängig vom CPU-Modul des *ADwin-Pro II*-Systems arbeiten.

Zeitgesteuerte Pegelausgabe

Das Modul kann selbstständig Pegel zu bestimmten Zeitpunkten auf Digitalausgängen ausgeben. Ein FIFO dient als Zwischenspeicher für die vom Benutzer festgelegten Pegel und Zeitpunkte, maximal 2048 Wertepaare. Der Ausgabezeitpunkt kann auf 5ns genau festgelegt werden.
Das FIFO kann entweder für die Flankenüberwachung von Eingängen oder für die Pegelausgabe an Ausgängen verwendet werden.

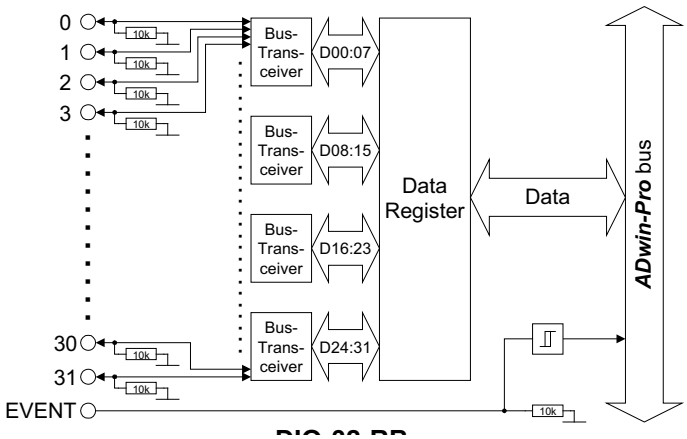


Abb. 98 – Pro II-DIO-32-TiCo2 Rev. E: Blockschaltbild

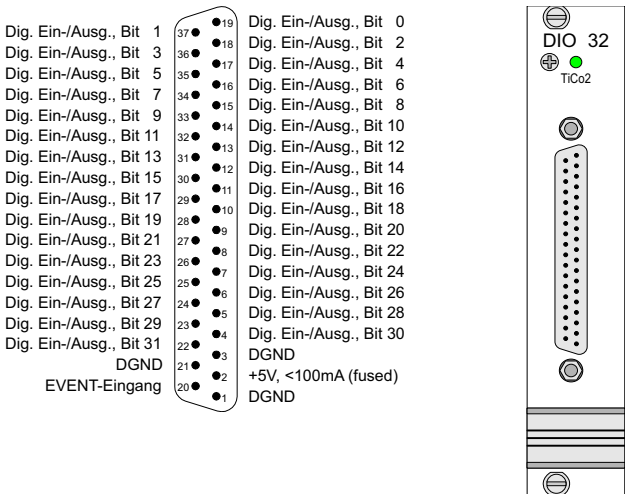


Abb. 99 – Pro II-DIO-32-TiCo2 Rev. E: Pinbelegung

Ein-/Ausgangskanäle	32; in Blöcken zu 8 Bit als Ein-/Ausgang mittels Software einstellbar
Pull-Down-Widerstand	10 kΩ
Spannungs-Schaltschwellen	$V_{IL\ max}$: 0,3×V_{cc}; $V_{IH\ min}$: 0,7×V_{cc} mit V_{cc} = eingestellter Spannungspegel
Spannungsbereich Digitalkanäle	-0,5 V … +5,5 V
Ausgangsstrom	max. ±35 mA pro Kanal, max. ±70 mA je Block (8 Kanäle) über V_{CC} oder GND
Event-Eingang	TTL-Logik
Power-Up-Status	Alle Kanäle als Eingänge
Eingangs- / Ausgangs-Fifo	Größe: 2048 Wertepaare Frequenz: 200 MHz
TiCo	Prozessortyp: TiCo2 Taktfrequenz: 100 MHz Speichergröße: 128 kiB PM intern, 512 kiB DM intern
Steckerverbindung	37-polige D-Sub-Buchse

Abb. 100 – Pro II-DIO-32-TiCo2 Rev. E: Spezifikation

Programmierung

Das Modul wird komfortabel mit *ADbasic*-Befehlen oder *TiCoBasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software sowie in den Online-Hilfen *ADbasic* und *TiCoBasic* erläutert.

Die Include-Datei `ADwinPro_All.inc` enthält folgende *ADbasic*-Befehle:

Bereich	Befehle
Ein- und Ausgänge konfigurieren	**P2_DigProg**, **P2_Dig_FIFO_Mode** **P2_DigProg_Set_IO_Level**
Eingangssignale abfragen	**P2_Digin_Long**
Latch-Register nutzen	**P2_Dig_Latch** **P2_Dig_Read_Latch** **P2_Dig_Write_Latch**
Flanken an Eingangskanälen überwachen	**P2_Digin_FIFO_Enable** **P2_Digin_FIFO_Read** **P2_Digin_FIFO_Read_Fast** **P2_Digin_FIFO_Read_Timer** **P2_Digin_FIFO_Clear** **P2_Digin_FIFO_Full**
Spike-Filter setzen	**P2_Digin_Filter_Init**
Flankenstatus abfragen	**P2_Digin_Edge**
Ausgangssignale setzen und rücklesen	**P2_Digout**, **P2_Digout_Bits** **P2_Digout_Long** **P2_Digout_Set** **P2_Digout_Reset** **P2_Get_Digout_Long**
Ausgangssignale automatisch setzen	**P2_Digout_FIFO_Clear** **P2_Digout_FIFO_Empty** **P2_Digout_FIFO_Enable** **P2_Digout_FIFO_Read_Timer** **P2_Digout_FIFO_Start** **P2_Digout_FIFO_Write**
Abläufe synchronisieren	**P2_Sync_All**, **P2_Sync_Enable** **P2_Sync_Stat**
LEDs einstellen	**P2_Check_LED**, **P2_Set_LED**
Interrupts und Event-Eingang einstellen	**P2_Event_Enable**, **P2_Event_Config** **P2_Event_Read**

Programmierung in TiCoBasic

Das Modul kann mit *TiCoBasic*-Befehlen programmiert werden. Die Befehle sind in der Online-Hilfe *TiCoBasic* erläutert.

Im Verzeichnis `C:\ADwin\TiCoBasic\samples_ADwin_ProII` finden Sie zusätzliche Beispielprogramme.

Die Include-Datei `DIO32TiCo.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Ein- und Ausgänge konfigurieren	**DigProg** **DigProg_Set_IO_Level**
Eingangssignale abfragen	**Digin_Long**
Flanken an Eingangskanälen überwachen	**Digin_FIFO_Enable** **Digin_FIFO_Read** **Digin_FIFO_Read_Timer** **Digin_FIFO_Clear** **Digin_FIFO_Full**

Bereich	Befehle
Spike-Filter setzen	Digin_Filter_Init
Flankenstatus abfragen	Digin_Edge
Ausgangssignale setzen und rücklesen	Digout, Digout_Bits Digout_Set, Digout_Reset Digout_Long Get_Digout_Long
Ausgangssignale automatisch setzen	Digout_FIFO_Clear Digout_FIFO_Empty Digout_FIFO_Enable Digout_FIFO_Read_Timer Digout_FIFO_Start Digout_FIFO_Write
LEDs einstellen	Check_LED, Set_LED
Interrupts und Event-Eingang einstellen	Event_Enable, Trigger_Event Event_Config

Programmierung TiCo-Zugriff

Für den Zugriff auf den *TiCo*-Prozessor von der ADwin CPU sind die folgenden *ADbasic*-Befehle in der Datei `ADwinPro_All.inc` definiert. Die Befehle sind im Handbuch *TiCoBasic* und in der Online-Hilfe *ADbasic* erläutert.

Bereich	Befehle
Datenaustausch mit dem *TiCo*-Prozessor über globale Variablen	P2_TDrv_Init P2_GetData_Long, P2_Get_Par, P2_Get_Par_Block P2_SetData_Long, P2_Set_Par, P2_Set_Par_Block P2_Get_TiCo_RingBuffer, P2_Set_TiCo_RingBuffer P2_RingBuffer_Empty P2_RingBuffer_Full
TiCo-Prozessor steuern	P2_TiCo_Reset, P2_TiCo_Start, P2_TiCo_Stop P2_Get_TiCo_Bootloader_Status P2_Get_TiCo_Status, P2_Workload
TiCo-Prozesse steuern	P2_Process_Status P2_TiCo_Get_Processdelay P2_TiCo_Set_Processdelay P2_TiCo_Start_Process P2_Tico_Stop_Process
TiCo-Programme übertragen	P2_TiCo_Flash, P2_TiCo_Load

5.7.5 Pro II-DIO-8-D12 Rev. E

Das digitale Ein-/Ausgangsmodul Pro II-DIO-8-D12 Rev. E stellt 20 programmierbare Ein- und Ausgangskanäle bereit, 8 Kanäle mit TTL-Pegeln und 12 differentielle Kanäle. Die Spannungspegel sind nicht einstellbar.

Die Kanäle mit TTL-Pegeln sind mit `P2_DigProg` in Gruppen zu je 4 als Eingänge oder Ausgänge umschaltbar. Die differentiellen Digitalkanäle sind mit `P2_DigProg_Bits` als Eingang oder Ausgang umschaltbar, und zwar jeweils einzeln. Nach dem Einschalten sind alle Kanäle als Eingänge konfiguriert.

Der Event-Eingang arbeitet differentiell.

Flankenüberwachung

Das Modul kann mit einer Frequenz von 200 MHz die Flanken an Eingangskanälen überwachen. Bei einer Änderung wird der aktuelle Pegelstand gemeinsam mit einem Zeitstempel in einem FIFO zwischengespeichert; es können bis zu 2048 solcher Wertepaare (Pegelstand und Zeitstempel) gespeichert werden. Die FIFO-Daten können ausgelesen und weiter verarbeitet werden.
Außerdem kann abgefragt werden, an welchen Eingangskanälen eine positive oder negative Flanke aufgetreten ist.

Spike-Filter

An den Eingangskanälen können einzelne Fehlpulse (Spikes) mit einem einstellbaren Filter unterdrückt werden. Jeder Kanal hat seinen eigenen Filter, aber die Filtereinstellungen gelten für alle Kanäle gleichermaßen. Nach dem Einschalten sind die Filter deaktiviert.

TiCo-Prozessor

Das Modul besitzt einen frei programmierbaren *TiCo*-Prozessor (Typ TiCo2) mit 128 kiB internem Programmspeicher und 512 kiB internem Datenspeicher. Der *TiCo*-Prozessor hat Zugriff auf alle digitalen Ein- und Ausgangskanäle. Nähere Hinweise zur Anwendung und Programmierung des *TiCo*-Prozessors entnehmen Sie bitte dem Handbuch *TiCoBasic*.

Wenn Sie ein *TiCoBasic*-Programm im *TiCo*-Bootloader ablegen, wird das Programm beim Einschalten der Stromversorgung in den *TiCo*-Prozessor geladen und gestartet. Auf diese Weise kann das Modul eigenständig und unabhängig vom CPU-Modul des *ADwin-Pro II*-Systems arbeiten.

Zeitgesteuerte Pegelausgabe

Das Modul kann selbstständig Pegel zu bestimmten Zeitpunkten auf Digitalausgängen ausgeben. Ein FIFO dient als Zwischenspeicher für die vom Benutzer festgelegten Pegel und Zeitpunkte, maximal 2048 Wertepaare. Der Ausgabezeitpunkt kann auf 5 ns genau festgelegt werden.
Das FIFO kann entweder für die Flankenüberwachung von Eingängen oder für die Pegelausgabe an Ausgängen verwendet werden.

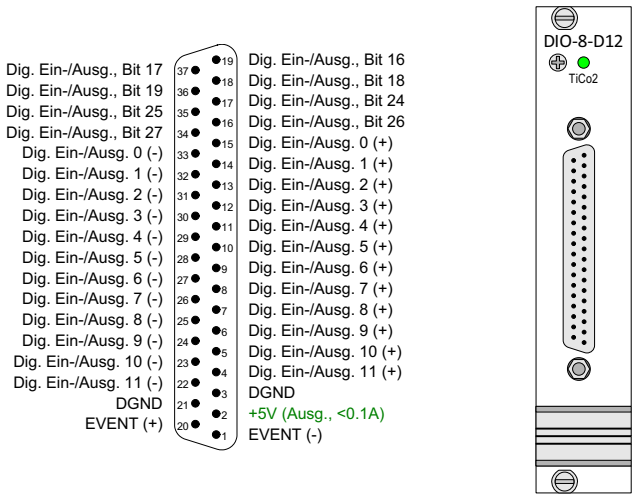


Abb. 101 – Pro II-DIO-8-D12 Rev. E: Pinbelegung

Ein-/Ausgangskanäle	8 Kanäle TTL-Logik (5V) 12 differentielle Kanäle
Eingangswiderstand	TTL-Eingänge: 10 kΩ, pull-down Diff. Eingänge:120 Ω
Ein-/Ausgangspegel TTL	Spannungsbereich 0 V … +5 V V_{IH} min. 2 V V_{IL} max. 0,8 V I_{IH} max. 1 µA
Ein-/Ausgangspegel differentiell	RS422/485 kompatibel (5V differentiell, 120 Ω Bus-Abschlusswiderstand)
Ausgangsstrom	max. ±35mA pro Kanal, max. ±70mA je TTL-Block (4 Kanäle) über V_{CC} oder GND
Event-Eingang	1 differentiell
Power-Up-Status	Alle Kanäle als Eingänge
Eingangs- / Ausgangs-Fifo	Größe: 2048 Wertepaare Frequenz: 200MHz
TiCo	Prozessortyp: TiCo2 Taktfrequenz: 100MHz Speichergröße: 128kiB PM intern, 512kiB DM intern
Steckerverbindung	37-polige D-Sub-Buchse

Abb. 102 – Pro II-DIO-8-D12 Rev. E: Spezifikation

Programmierung

Das Modul wird komfortabel mit *ADbasic*-Befehlen oder *TiCoBasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software sowie in den Online-Hilfen *ADbasic* und *TiCoBasic* erläutert.

Die Include-Datei `ADwinPro_All.inc` enthält folgende *ADbasic*-Befehle:

Bereich	Befehle
Ein- und Ausgänge konfigurieren	`P2_DigProg` `P2_DigProg_Bits`
Eingangssignale abfragen	`P2_Digin_Long`

Latch-Register nutzen	`P2_Dig_Latch`
	`P2_Dig_Read_Latch`
	`P2_Dig_Write_Latch`

Flanken an Eingangskanälen überwachen	`P2_Digin_FIFO_Enable`
	`P2_Digin_FIFO_Read`
	`P2_Digin_FIFO_Read_Fast`
	`P2_Digin_FIFO_Read_Timer`
	`P2_Digin_FIFO_Clear`
	`P2_Digin_FIFO_Full`

Spike-Filter setzen	`P2_Digin_Filter_Init`
Flankenstatus abfragen	`P2_Digin_Edge`

Ausgangssignale setzen und rücklesen	`P2_Digout`, `P2_Digout_Bits`
	`P2_Digout_Long`
	`P2_Digout_Set`
	`P2_Digout_Reset`
	`P2_Get_Digout_Long`

Bereich	Befehle
Ausgangssignale automatisch setzen	P2_Digout_FIFO_Clear P2_Digout_FIFO_Empty P2_Digout_FIFO_Enable P2_Digout_FIFO_Read_Timer P2_Digout_FIFO_Start P2_Digout_FIFO_Write
Abläufe synchronisieren	P2_Sync_All, P2_Sync_Enable P2_Sync_Stat
LEDs einstellen	P2_Check_LED, P2_Set_LED
Interrupts und Event-Eingang einstellen	P2_Event_Enable, P2_Event_Config P2_Event_Read

Programmierung in TiCoBasic

Das Modul kann mit *TiCoBasic*-Befehlen programmiert werden. Die Befehle sind in der Online-Hilfe *TiCoBasic* erläutert.

Im Verzeichnis `C:\ADwin\TiCoBasic\samples_ADwin_ProII` finden Sie zusätzliche Beispielprogramme.

Die Include-Datei `DIO32TiCo.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Ein- und Ausgänge konfigurieren	DigProg DigProg_Bits
Eingangssignale abfragen	Digin_Long
Flanken an Eingangskanälen überwachen	Digin_FIFO_Enable Digin_FIFO_Read Digin_FIFO_Read_Timer Digin_FIFO_Clear Digin_FIFO_Full
Spike-Filter setzen	Digin_Filter_Init
Flankenstatus abfragen	Digin_Edge
Ausgangssignale setzen und rücklesen	Digout, Digout_Bits Digout_Set, Digout_Reset Digout_Long Get_Digout_Long
Ausgangssignale automatisch setzen	Digout_FIFO_Clear Digout_FIFO_Empty Digout_FIFO_Enable Digout_FIFO_Read_Timer Digout_FIFO_Start Digout_FIFO_Write
LEDs einstellen	Check_LED, Set_LED
Interrupts und Event-Eingang einstellen	Event_Enable, Trigger_Event Event_Config

Programmierung TiCo-Zugriff

Für den Zugriff auf den *TiCo*-Prozessor von der ADwin CPU sind die folgenden *ADbasic*-Befehle in der Datei `ADwinPro_All.inc` definiert. Die Befehle sind im Handbuch *TiCoBasic* und in der Online-Hilfe *ADbasic* erläutert.

Bereich	Befehle
Datenaustausch mit dem *TiCo*-Prozessor über globale Variablen	P2_TDrv_Init P2_GetData_Long, P2_Get_Par, P2_Get_Par_Block P2_SetData_Long, P2_Set_Par, P2_Set_Par_Block P2_Get_TiCo_RingBuffer, P2_Set_TiCo_RingBuffer P2_RingBuffer_Empty P2_RingBuffer_Full
TiCo-Prozessor steuern	P2_TiCo_Reset, P2_TiCo_Start, P2_TiCo_Stop P2_Get_TiCo_Bootloader_Status P2_Get_TiCo_Status, P2_Workload
TiCo-Prozesse steuern	P2_Process_Status P2_TiCo_Get_Processdelay P2_TiCo_Set_Processdelay P2_TiCo_Start_Process P2_Tico_Stop_Process
TiCo-Programme übertragen	P2_TiCo_Flash, P2_TiCo_Load

5.7.6 Pro II-OPT-16 Rev. E

Das Eingangsmodul Pro II-OPT-16 Rev. E stellt 16 Kanäle mit optisch isolierten digitalen Eingängen bereit. Die Eingangs-Spannungsbereiche sind für jeden Eingang separat über Jumper einstellbar (5V, 12V, 24V). Die Voreinstellung ist 24V. Die Schaltzeit von nur 100ns erlaubt das Einlesen von schnellen digitalen Signalen.

Das Modul kann mit einer Frequenz von 100MHz die Flanken an Eingangskanälen überwachen. Bei einer Änderung wird der aktuelle Pegelstand gemeinsam mit einem Zeitstempel in einem FIFO zwischengespeichert; es können bis zu 511 solcher Wertepaare (Pegelstand und Zeitstempel) gespeichert werden. Die FIFO-Daten können ausgelesen und weiter verarbeitet werden.
Außerdem kann abgefragt werden, an welchen Eingangskanälen eine positive oder negative Flanke aufgetreten ist.

Jeder Kanal ist vom Systemstromkreis und von den anderen Eingängen optisch isoliert, wie auch der Event-Eingang.

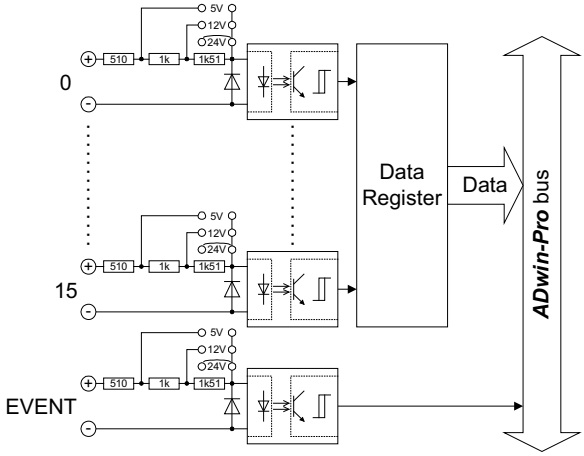


Abb. 103 – Pro II-OPT-16 Rev. E: Blockschaltbild

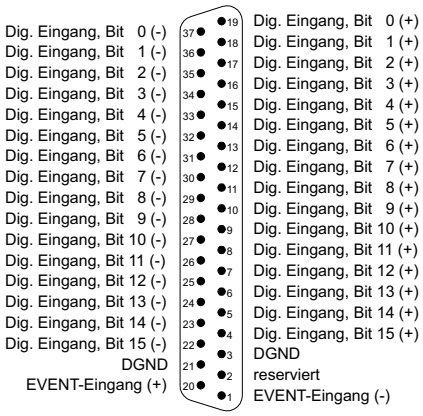


Abb. 104 – Pro II-OPT-16 Rev. E: Pinbelegung

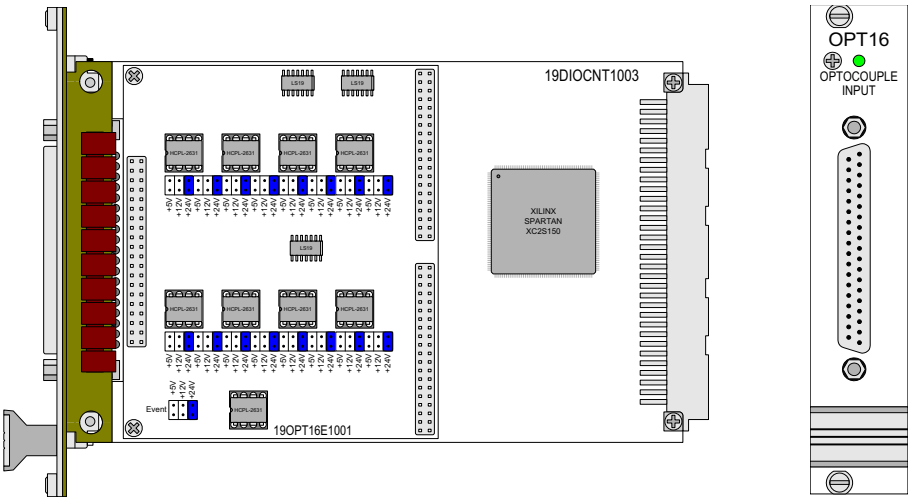


Abb. 105 – Pro II-OPT-16 Rev. E: Platine und Frontplatte

Eingangskanäle	16		
Event-Eingänge	1		
Eingangsstrom	typ. 3,5mA / max. 7,5mA		
Eingangs-Spannungsbereich (über Jumper wählbar)	0…5V	0…12V	0…24V
Schaltschwelle für 0-low	0…0,8V	0…1,6V	0…3,2V
Schaltschwelle für 1-high	4,5…5V	10…12V	20…24V
Spannungsfestigkeit	-5V … 8V	-5V … 16V	-5V … 30V
Schaltzeit	100ns		
Eingangs-Fifo	Größe: 511 Wertepaare Frequenz: 100MHz		
Isolation	42V Kanal zu Kanal / Kanal zu Masse		
Steckerverbindung	37-polige D-Sub-Buchse		

Abb. 106 – Pro II-OPT-16 Rev. E: Spezifikation

Programmierung

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Eingangssignale abfragen	`P2_Digin_Edge` `P2_Digin_Long`
Latch-Register nutzen	`P2_Dig_Latch` `P2_Dig_Read_Latch`

Flanken an Eingangskanälen überwachen	`P2_Digin_FIFO_Enable`
	`P2_Digin_FIFO_Read`
	`P2_Digin_FIFO_Read_Fast`
	`P2_Digin_FIFO_Read_Timer`
	`P2_Digin_FIFO_Clear`
	`P2_Digin_FIFO_Full`

Abläufe synchronisieren	`P2_Sync_All`, `P2_Sync_Enable` `P2_Sync_Stat`
LEDs einstellen	`P2_Check_LED`, `P2_Set_LED`

Bereich	Befehle
Interrupts und Event-Eingang einstellen	P2_Event_Enable P2_Event_Config P2_Event_Read

5.7.7 Pro II-OPT-32-24V Rev. E

Das Eingangsmodul Pro II-OPT-32-24V Rev. E stellt 32 Kanäle mit optisch isolierten digitalen Eingängen bereit.

Der Eingangs-Spannungsbereich ist fest eingestellt auf 24 V und gilt auch für den Event-Eingang. Die Schaltzeit von nur 100 ns erlaubt das Einlesen von schnellen digitalen Signalen.

Das Modul kann automatisch die Flanken an Eingangskanälen überwachen. Bei einer Änderung wird der aktuelle Pegelstand gemeinsam mit einem Zeitstempel in einem FIFO zwischengespeichert. Die FIFO-Daten können ausgelesen und weiter verarbeitet werden.

Jeder Kanal ist vom Systemstromkreis und von den anderen Eingängen optisch isoliert, wie auch der Event-Eingang.

Die Kanäle sind als single-ended ausgeführt und haben ein gemeinsames Massepotenzial (ext. GND), das auf der D-SUB-Buchse bereitgestellt werden muss.

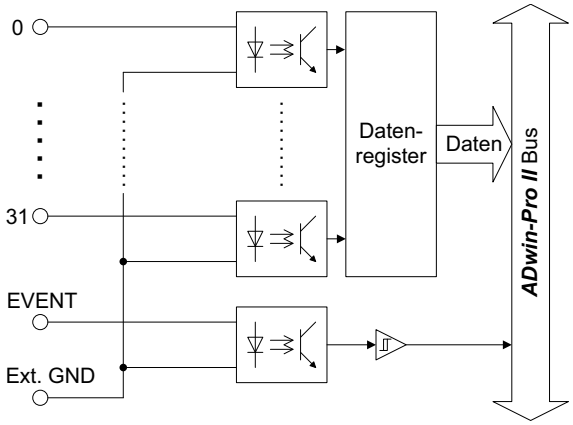


Abb. 107 – Pro II-OPT-32-24V Rev. E: Blockschaltbild

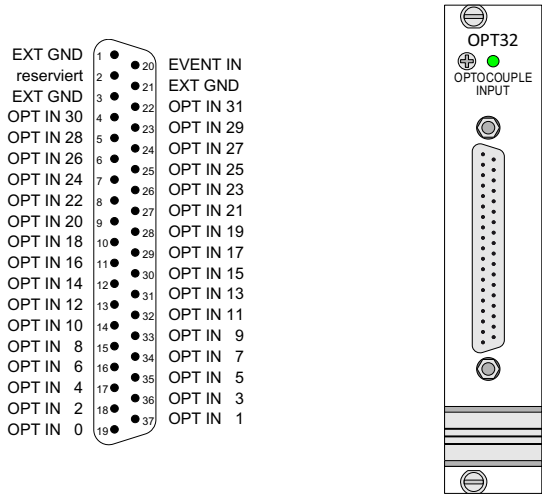


Abb. 108 – Pro II-OPT-32-24V Rev. E: Pinbelegung und Frontplatte

Eingangskanäle	32 single-ended
Event-Eingänge	1
Eingangsstrom	typ. 3,5mA / max. 7,5mA
Eingangs-Spannungsbereich (Bestelloption)	0…24V
Schaltschwelle für 0-low	0…3,2V
Schaltschwelle für 1-high	20…24V
Spannungsfestigkeit	-5V … 30V
Schaltzeit	100ns
Eingangs-Fifo	Größe: 511 Wertepaare Frequenz: 100MHz
Isolation	42V Kanal zu Kanal / Kanal zu Masse
Steckerverbindung	37-polige D-Sub-Buchse

Abb. 109 – Pro II-OPT-32-24V Rev. E: Spezifikation

Programmierung

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert.

Die Datei `ADwinPRO_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Eingangssignale abfragen	`P2_Digin_Edge` `P2_Digin_Long`
Latch-Register nutzen	`P2_Dig_Latch` `P2_Dig_Read_Latch`

Flanken an Eingangskanälen überwachen	`P2_Digin_FIFO_Enable`
	`P2_Digin_FIFO_Read`
	`P2_Digin_FIFO_Read_Fast`
	`P2_Digin_FIFO_Read_Timer`
	`P2_Digin_FIFO_Clear`
	`P2_Digin_FIFO_Full`

Abläufe synchronisieren	`P2_Sync_All`, `P2_Sync_Enable` `P2_Sync_Stat`
LEDs einstellen	`P2_Check_LED`, `P2_Set_LED`

Interrupts und Event-Eingang einstellen	`P2_Event_Enable`
	`P2_Event_Config`
	`P2_Event_Read`

Die Befehle sind im Handbuch Pro-Software und in der Online-Hilfe erläutert.

5.7.8 Pro II-REL-16 Rev. E

Das Ausgangsmodul Pro II-REL-16 Rev. E stellt 16 isolierte Relaisausgänge bereit. Jeder Kanal ist vom System und den anderen Kanälen getrennt. Der Event-Eingang ist optisch vom Systemstromkreis isoliert.

Das Modul ist mit Schließern bestückt. Optional ist das Modul auch mit Öffnern erhältlich.

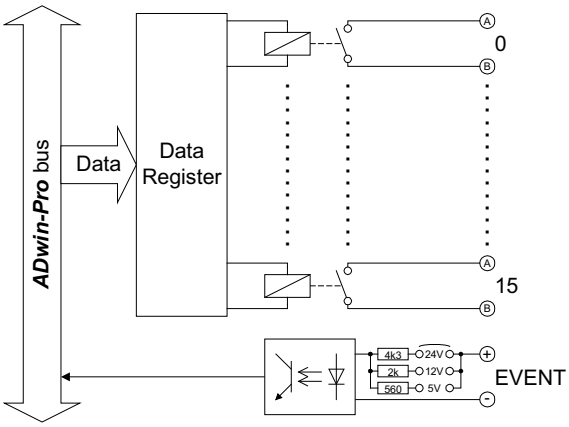


Abb. 110 – Pro II-REL-16 Rev. E: Blockschaltbild

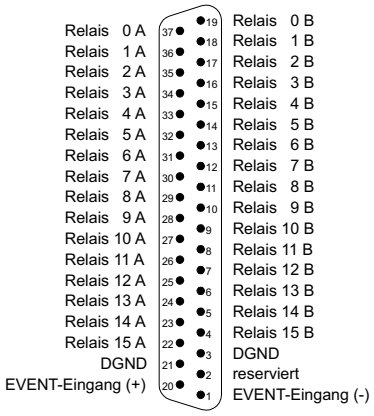


Abb. 111 – Pro II-REL-16 Rev. E: Pinbelegung

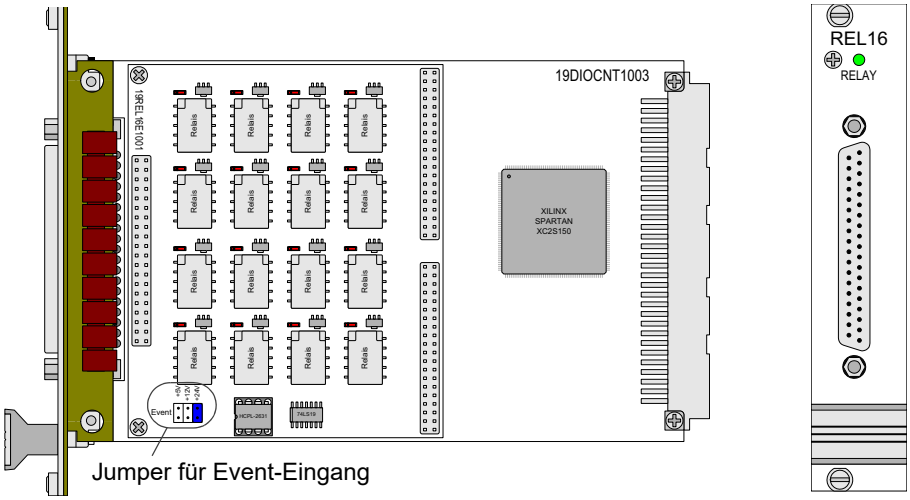


Abb. 112 – Pro II-REL-16 Rev. E: Platine und Frontplatte

Ausgangskanäle	16
Kontaktspannung	30 V AC/DC Maximum
Kontaktstrom	max. 500 mA pro Kanal
Kontaktausführung	1 Schließer pro Kanal, (optional: Öffner)
Ansprechzeit	4 ms
Abfallzeit	3 ms
Prellzeit	2 ms
Event-Eingänge	1
Isolation	42 V Kanal zu Kanal/ Kanal zu Masse
Event-Eingangsspannung	5 V, 12 V, 24 V (über Jumper wählbar)
Power-Up-Status	low (mit Schließern: offen / mit Öffnern: geschlossen)
Steckerverbindung	37-polige D-Sub-Buchse

Abb. 113 – Pro II-REL-16 Rev. E: Spezifikation

Programmierung

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle

Ausgangssignale setzen und rücklesen	`P2_Digout`, `P2_Digout_Long`
	`P2_Digout_Bits`
	`P2_Get_Digout_Long`

Latch-Register nutzen	`P2_Dig_Latch` `P2_Dig_Write_Latch`

Ausgangssignale setzen und rücklesen	`P2_Digout`, `P2_Digout_Bits`
	`P2_Digout_Long`
	`P2_Digout_Set`
	`P2_Digout_Reset`
	`P2_Get_Digout_Long`

Abläufe synchronisieren	`P2_Sync_All`, `P2_Sync_Enable` `P2_Sync_Stat`
LEDs einstellen	`P2_Check_LED`, `P2_Set_LED`

Interrupts und Event-Eingang einstellen	`P2_Event_Enable`
	`P2_Event_Config`
	`P2_Event_Read`

5.7.9 Pro II-TRA-16 Rev. E

Das Ausgangsmodul Pro II-TRA-16 Rev. E stellt 16 galvanisch getrennte Transistor-Schaltausgänge bereit. Es gibt 2 Modulvarianten:

– Pro II-TRA-16 Rev. E: Die Ausgänge schalten nach V_{CC}.

– Pro II-TRA-16-G Rev. E: Die Ausgänge schalten nach Masse.

Die Schaltspannung V_{CC} muss durch eine externe Spannungsversorgung zugeführt werden.

Die Kanäle wie auch der Event-Eingang sind optisch vom System-Stromkeis isoliert.

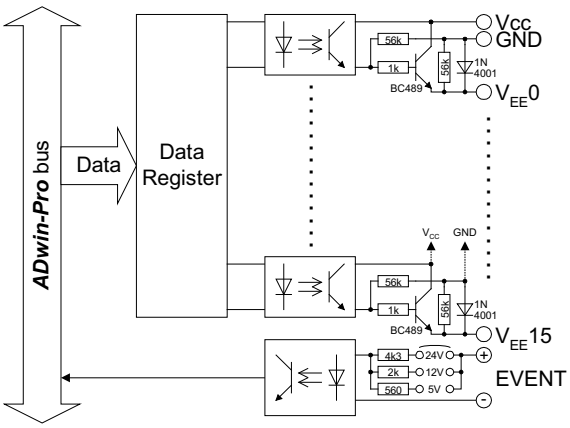


Abb. 114 – Pro II-TRA-16 Rev. E: Blockschaltbild

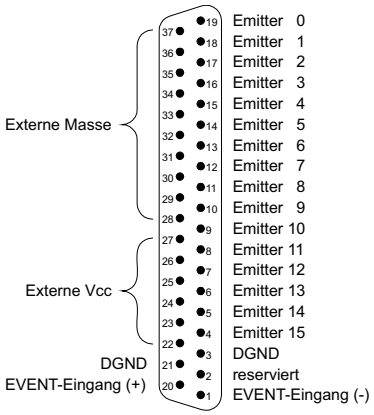


Abb. 115 – Pro II-TRA-16 Rev. E: Pinbelegung

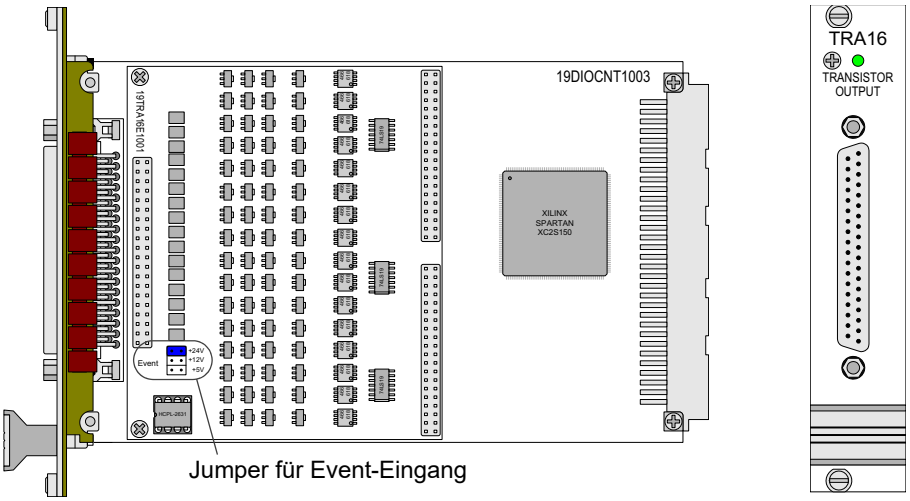


Abb. 116 – Pro II-TRA-16 Rev. E: Platine und Frontplatte

Ausgangskanäle	16
Schaltspannung V_{CC}	5…30V DC durch externe Spannungsversorgung
Schaltstrom	Pro II-TRA-16: 200mA max. pro Kanal Pro II-TRA-16-G: 100mA max. pro Kanal
Spannungsabfall	0,5V
Schaltzeit	2,5µs
Event-Eingang	1
Isolation	42V Kanal zu Kanal / Kanal zu Masse
Event-Eingangsspannung	5V, 12V, 24V (über Jumper wählbar)
Power-Up-Status	low (GND extern)
Steckerverbindung	37-polige D-Sub-Buchse

Abb. 117 – Pro II-TRA-16 Rev. E: Spezifikation

Programmierung

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle

Ausgangssignale setzen und rücklesen	`P2_Digout`, `P2_Digout_Long`
	`P2_Digout_Bits`
	`P2_Get_Digout_Long`

Latch-Register nutzen	`P2_Dig_Latch` `P2_Dig_Write_Latch`
Abläufe synchronisieren	`P2_Sync_All`, `P2_Sync_Enable` `P2_Sync_Stat`
LEDs einstellen	`P2_Check_LED`, `P2_Set_LED`

Interrupts und Event-Eingang einstellen	`P2_Event_Enable`
	`P2_Event_Config`
	`P2_Event_Read`

5.7.10 Pro II-PWM-16 Rev. E

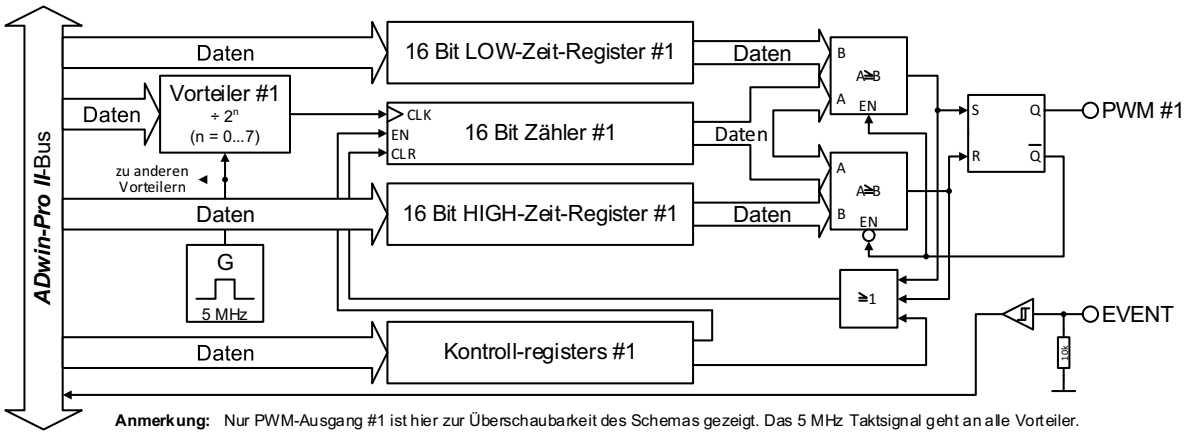
Das Modul Pro II-PWM-16 Rev. E gibt auf 16 Ausgängen pulsweitenmodulierte Signale (PWM-Signale) aus. Die (PWM-)Signale sind getrennt voneinander über Software konfigurierbar, d.h. sie können getrennt voneinander eingestellt werden.

Es gibt das Modul auch in der Variante Pro II-PWM-16-I Rev. E. Dort sind die Ausgänge gegen den Systemstromkreis und die anderen Ausgänge optisch isoliert. Auch der Event-Eingang ist gegen den Systemstromkreis isoliert.

Die Ausgabesignale werden mit einem Referenztakt von 100 MHz getaktet.

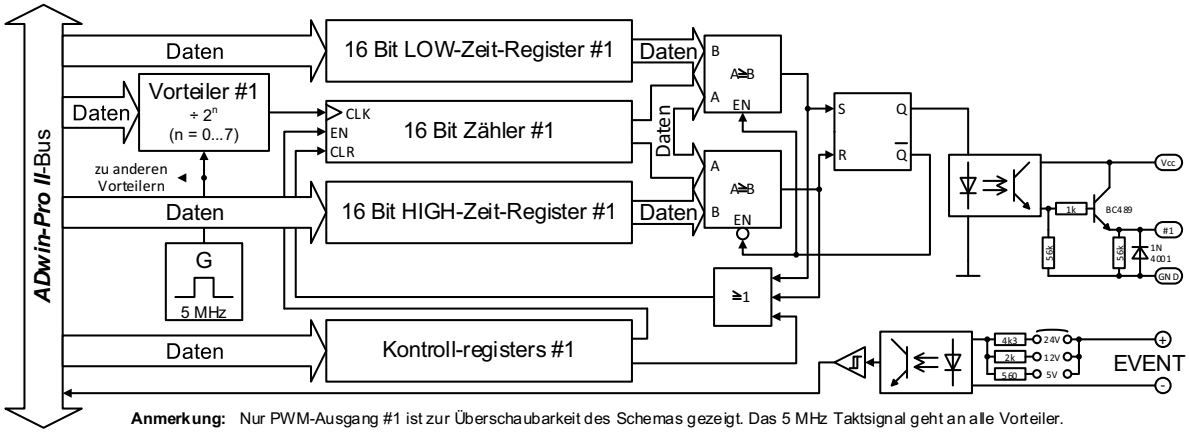
Die niedrigste Ausgangsfrequenz beträgt ca. 0,6 Hz.

Die höchste Ausgangsfrequenz, bei der das Tastverhältnis noch in 1%-Schritten einstellbar ist, beträgt 1000 kHz.



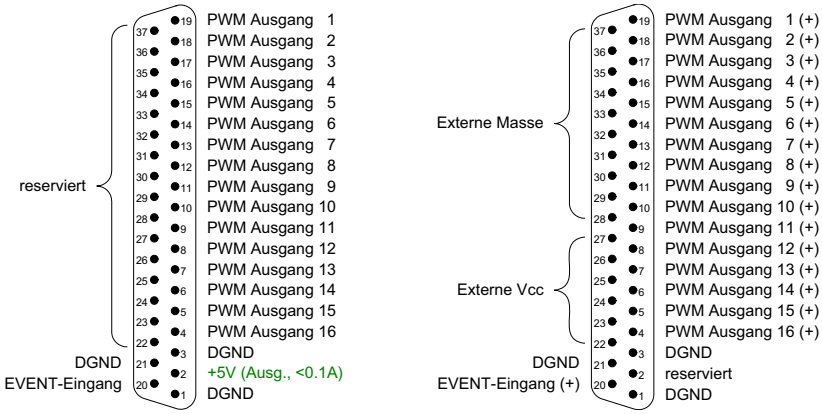
Anmerkung: Nur PWM-Ausgang #1 ist hier zur Überschaubarkeit des Schemas gezeigt. Das 5 MHz Taktsignal geht an alle Vorteiler.

oben: Pro II-PWM-16 Rev. E, unten: Pro II-PWM-16-I Rev. E



Anmerkung: Nur PWM-Ausgang #1 ist zur Überschaubarkeit des Schemas gezeigt. Das 5 MHz Taktsignal geht an alle Vorteiler.

Abb. 118 – Pro II-PWM-16 Rev. E: Blockschaltbild



Pro II-PWM-16 Rev. E Pro II-PWM-16-I Rev. E

Abb. 119 – Pro II-PWM-16 Rev. E: Pinbelegung

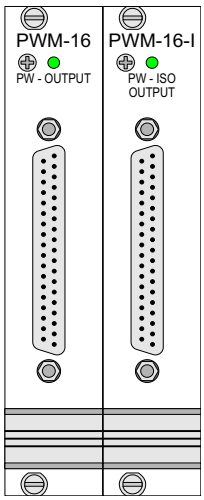


Abb. 120 – Pro II-PWM-16 Rev. E: Frontplatte

Ausgangskanäle	16 PWM-Kanäle
Ausgänge	TTL
Zähler-/Registerbreite	32 Bit
Referenztakt	100 MHz
Event-Eingang	1 Eingang, positiv TTL
Steckerverbindung	37-polige D-Sub-Buchse
Pro II-PWM-16 Rev. E	
V_{OH}	2,4 V min.
V_{OL}	0,8 V max.
Ausgangsstrom	5 mA pro Kanal max.
Isolation	Nein
Pro II-PWM-16-I Rev. E	
Ausgangs-Spannung	5…30 V DC durch externe Spannungsversorgung
Ausgangsstrom	100 mA max. pro Kanal
Spannungsabfall	0,5 V max.

Abb. 121 – Pro II-PWM-16 Rev. E: Spezifikation

Schaltzeit	ohne Last: 3µs mit Last: 1…2µs
Event-Eingangsspannung	5V, 12V, 24V (über Jumper wählbar)
Isolation	42V Kanal zu Kanal / Kanal zu Masse

Abb. 121 – Pro II-PWM-16 Rev. E: Spezifikation

Programmierung

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Im Verzeichnis `C:\ADwin\ADbasic\samples_ADwin_ProII` finden Sie das Beispielprogramm `ProII-PWM-16.bas`.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Modul initialisieren	**P2_PWM_Init** **P2_PWM_Standby_Value** **P2_PWM_Enable** **P2_PWM_Reset**
PWM-Signale ausgeben und rücklesen	**P2_PWM_Write_Latch** **P2_PWM_Latch**, **P2_Sync_All** **P2_PWM_Get_Status**
Abläufe synchronisieren	**P2_Sync_All**, **P2_Sync_Enable** **P2_Sync_Stat**
LEDs einstellen	**P2_Check_LED**, **P2_Set_LED**
Interrupts und Event-Eingang einstellen	**P2_Event_Enable** **P2_Event_Config** **P2_Event_Read**

5.7.11 Pro II-COMP-16 Rev. E

Das Eingangsmodul Pro II-COMP-16 Rev. E stellt 16 Kanäle mit jeweils eigenem Komparator bereit. Anliegende Signale werden analog mit einer Schaltschwelle verglichen, die per Software eingestellt wird. Je nachdem, ob die Schaltschwelle auf einem Kanal über- oder unterschritten ist, wird das entsprechende Bit auf 1 oder 0 gesetzt.

Zusätzlich verfügbar sind eine symmetrische Hysterese um eine Schaltschwelle, das Halten von Komparatorwerten, eine frei wählbare Hysterese (durch Verwenden von 2 Kanälen mit unterschiedlicher Schaltschwelle) sowie das Filtern von Fehlpulsen.

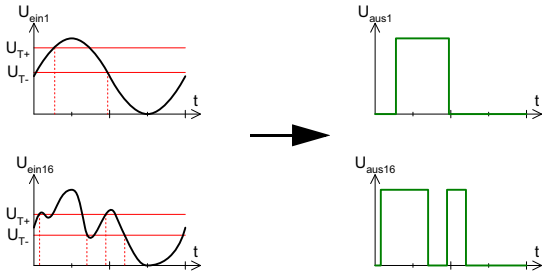


Abb. 122 – Pro II-COMP-16 Rev. E: Komparator-Prinzip mit Hysterese

Der Eingangs-Spannungsbereich und der Einstellbereich für die Schaltschwelle ist -1 V … +30 V.

Sie geben die Schaltschwelle digital per Software vor; daraus erzeugt ein DAC mit 16 Bit Auflösung das Analogsignal für den Vergleich im Komparator. Auf dem Modul stehen 4 DAC zur Verfügung, so dass je 4 Kanäle mit der gleichen Schaltschwelle arbeiten. Die Zuordnung der DAC zu den Kanälen ist wie folgt:

DAC	Kanäle
1	1, 3, 5, 7
2	2, 4, 6, 8
3	9, 11, 13, 15
4	10, 12, 14, 16

Die Komparatoren arbeiten mit 50 MHz.

Flankenüberwachung

Das Modul kann automatisch die Flanken an den Eingangskanälen (= Signale nach den Komparatoren) überwachen. Eine positive Flanke ist also das Überschreiten, eine negative das Unterschreiten der Schaltschwelle. Bei einer Änderung wird der aktuelle Pegelstand gemeinsam mit einem Zeitstempel in einem FIFO (Speicherprinzip: First in, first out) zwischengespeichert. Die FIFO-Daten können ausgelesen und weiter verarbeitet werden.

Zusatzfunktionen

Das Modul erlaubt mehrere zusätzliche Einstellungen das Halten des Komparators (Hold), die Einstellung einer Hysterese um die Schaltschwelle (Standard) und eine frei wählbare Hysterese durch Verwenden von 2 Kanälen mit unterschiedlicher Schaltschwelle.

– Komparator halten

  Sobald der anliegende Analogwert die Schaltschwelle erreicht, wird der Komparator für eine definierte Haltezeit deaktiviert und arbeitet erst danach weiter. Die Haltezeit kann auf folgende Zeiten eingestellt werden: 1 µs, 10 µs oder 100 µs.

  Sie können das Halten jeweils für eine Gruppe von 4 Kanälen (1…4, 5…8, 9…12, 13…16) einstellen. Halten und Standard-Hysterese können nicht kombiniert werden.

– Standard-Hysterese

  Die Hysterese bezieht sich auf die eingestellte Schaltschwelle. Das Bit des Kanals wird auf 1 gesetzt, wenn die Schaltschwelle plus Hysterese überschritten wird, und wieder auf 0 gesetzt, wenn die Schaltschwelle minus Hysterese unterschritten wird.

  Die Hysterese kann auf die folgenden Werte eingestellt werden. Bis Rev. E02: 0 mV, ±17 mV, ±33 mV, ±55 mV, ±100 mV; ab Rev. E03: 0 mV, ±34 mV, ±66 mV, ±110 mV, ±200 mV..

  Sie können die Standard-Hysterese jeweils für eine Gruppe von 4 Kanälen (1…4, 5…8, 9…12, 13…16) einstellen. Halten und Standard-Hysterese können nicht kombiniert werden.

– Frei wählbare Hysterese

  Um andere, insbesondere größere Hysteresen als die Standard-Hysterese zu nutzen, kombinieren Sie jeweils ein Kanalpaar (1,2 / … / 15, 16). Sie legen das Signal an beide Kanäle an. Das Komparator-Bit des Kanalpaars wird auf 1 gesetzt, wenn das Signal die Schwellenwerte auf beiden Kanälen überschreitet. Ebenso wird das Komparator-Bit auf 0 gesetzt, wenn das Signal die Schwellenwerte auf beiden Kanälen unterschreitet.

  Die für frei wählbare Hysterese genutzten Kanäle können zusätzlich mit Halten oder Standard-Hysterese kombiniert werden.

– Komparator-Filter

  Sie können nach den Komparatoren einen Filter aktivieren, um einzelne Fehlpulse (Spikes) zu unterdrücken. Die Anzahl der Fehlpulse sollte im Verhältnis zur Pulsbreite des Signals klein sein.

  Die Filtereinstellungen gelten für alle Kanäle gleichermaßen; jeder Kanal hat seinen eigenen Filter. Die Prüfdauer des Filters ist einstellbar von 20 ns bis 1,31 ms und sollte etwas länger sein als die erwartete Breite der Fehlpulse.

  Die Prüfdauer gibt an, wie lange ein Komparator eine 1 liefern muss, damit das Bit des Kanals auf 1 gesetzt wird. Der Zeitfilter zählt in 20 ns-Takten, bis der Zielwert erreicht ist; wenn der Komparator eine 0 liefert, wird rückwärts gezählt. Für das Unterschreiten des Schwellenwerts gilt das Gleiche entsprechend umgekehrt.

  Der Filter kann mit Standard-Hysterese und frei wählbarer Hysterese kombiniert werden. Der Filter ist nicht geeignet für die Kombination mit dem Komparator-Modus „Hold“.

  Die folgende Abbildung zeigt das Filtern von 2 Beispielsignalen, links ohne, rechts mit Fehlpuls. Der Filter verzögert die Flanke des resultierenden Signals um die Prüfdauer plus der doppelten Dauer der Fehlpulse.

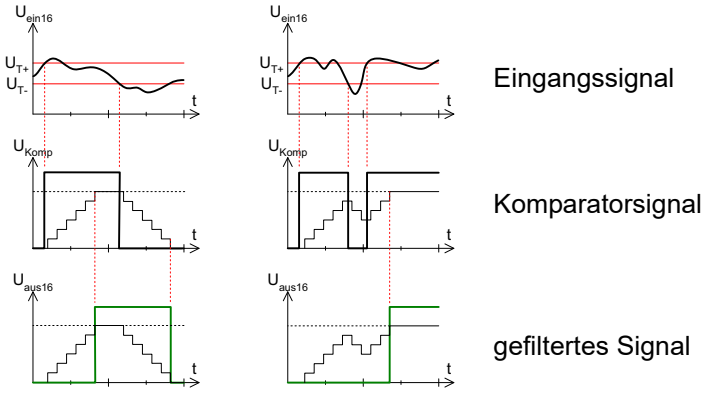


Abb. 123 – Pro II-COMP-16 Rev. E: Komparator-Filter

Komparator-Bits lesen

Sie lesen die Komparator-Bits mit Befehlen für Digitaleingänge wie z.B. `P2_Digin_Long`.

Bei Halten / Standard-Hysterese ist die Zuordnung der Bits zu den Einzelkanälen wie folgt:

Bit-Nr.	31…16	15	14	…	2	1	0
Kanalnr.	–	16	15	…	3	2	1

Die Komparator-Bits für die Kanalpaare der frei wählbaren Hysterese sind die Bits 16…23; die Zuordnung ist wie folgt:

Bit-Nr.	31…24	23	22	…	17	16	15…0
Kanalpaar	–	15, 16	13, 14	…	3, 4	1, 2	–

Event-Eingang

Der Kanal 16 arbeitet auch als Event-Eingang, es gibt für diese Funktion also keinen separaten Pin. Als Event-Signal wird das Signal nach dem Komparator verwendet.

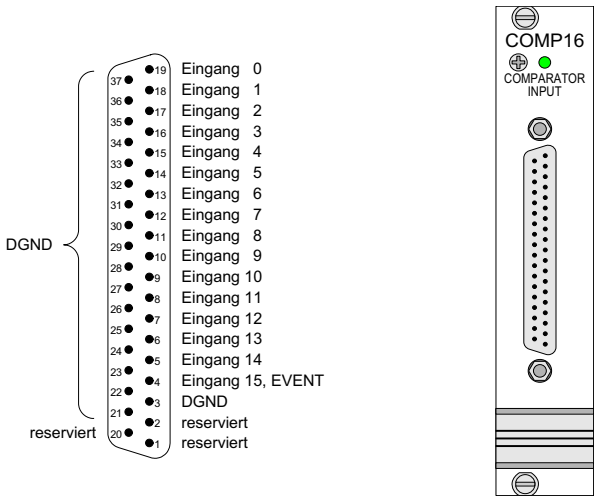


Abb. 124 – Pro II-COMP-16 Rev. E: Pinbelegung und Frontplatte

Eingangskanäle	16 Eingänge single-ended mit jeweils eigenem Komparator Schaltschwelle für Gruppen von je 4 Kanälen frei einstellbar
Event-Eingänge	1
Spannungsfestigkeit	-2V…+32V
Eingangs-Spannungsbereich	-1V … +30V
Abtastrate	50MHz
Auflösung Schaltschwelle	16 Bit
DAC-Einschwingzeit	30µs
Eingangs-Fifo	Größe: 511 Wertepaare Frequenz: 100MHz
Steckerverbindung	37-polige D-Sub-Buchse

Abb. 125 – Pro II-COMP-16 Rev. E: Spezifikation

Programmierung

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle

Modus Halten / Hysterese einstellen
Komparator-Filter einstellen
Komparator-Schaltschwelle einstellen
	`P2_Comp_Init`
	`P2_Comp_Filter_Init`
	`P2_Comp_Set`

Komparator-Bits abfragen	`P2_Digin_Edge` `P2_Digin_Long`
Latch-Register nutzen	`P2_Dig_Latch` `P2_Dig_Read_Latch`

Flanken an Eingangskanälen überwachen	`P2_Digin_FIFO_Enable`
	`P2_Digin_FIFO_Read`
	`P2_Digin_FIFO_Read_Fast`
	`P2_Digin_FIFO_Read_Timer`
	`P2_Digin_FIFO_Clear`
	`P2_Digin_FIFO_Full`

LEDs einstellen	`P2_Check_LED`, `P2_Set_LED`

Interrupts und Event-Eingang einstellen	`P2_Event_Enable`
	`P2_Event_Config`
	`P2_Event_Read`

5.7.12 Pro II-CNT-x Rev. E

Das Modul Pro II-CNT-x Rev. E stellt 4 konfigurierbare Zählerblöcke zur Verfügung. Ein Zählerblock enthält zwei 32 Bit-Zähler: Einerseits einen Vor-/Rückwärtszähler mit Takt/Richtungs-Auswertung oder Vierflankenauswertung zum Anschluss von Encodern. Zum anderen einen Zähler zur Periodendauer- und Tastverhältnismessung. Beide Zähler eines Blocks können parallel genutzt werden.

Es gibt folgende Modulvarianten:

– *Pro II-CNT-T*: Zählereingänge mit TTL-Logik. Das Modul besitzt einen *TiCo*-Prozessor.

– *Pro II-CNT-D*: Zählereingänge differentiell; das Modul enthält zusätzlich 2 SSI-Decoder und einen *TiCo*-Prozessor.

– *Pro II-CNT-I*: Zählereingänge sind optisch isoliert. Das Modul besitzt einen *TiCo*-Prozessor.

  Die Zähler-Eingänge sind gegen den Systemstromkreis und gegeneinander optisch isoliert. Auch der Event-Eingang ist gegen den Systemstromkreis isoliert.

  Der Eingangs-Spannungsbereich der Zählereingänge und des Event-Eingangs kann mit Hilfe von Jumpern auf 0…5V, 0…12V oder 0…24V eingestellt werden. Die Voreinstellung ist 0…24V.

Der *TiCo*-Prozessor ist frei programmierbar und hat Zugriff auf alle Zählerfunktionen. Nähere Hinweise zur Anwendung und Programmierung des *TiCo*-Prozessors entnehmen Sie bitte dem Handbuch *TiCoBasic*.

Wenn Sie ein *TiCoBasic*-Programm im *TiCo*-Bootloader ablegen, wird das Programm beim Einschalten der Stromversorgung in den *TiCo*-Prozessor geladen und gestartet. Auf diese Weise kann das Modul eigenständig und unabhängig vom CPU-Modul des *ADwin-Pro II*-Systems arbeiten.

Vor-/Rückwärtszähler

Der Vor-/ Rückwärtszähler eines Blocks kann in 2 Betriebsarten arbeiten:

– Takt/Richtungs-Auswertung (CLK- und DIR-Signale)

  Eine negative Flanke am CLK-Eingang löst einen Zählimpuls am 32 Bit-Zähler aus. Das DIR-Signal bestimmt die Zählrichtung des Zählers, TTL high bedeutet Hochzählen, TTL low bedeutet Herunterzählen.

  Sie können die Signale an den Eingängen CLK und DIR per Software (Befehl `P2_Cnt_Mode`) invertieren und damit sowohl die auslösende Flanke als auch die Zählrichtung ändern.

  Sie können den Zählerstand programmgesteuert ins Latch übernehmen oder den Zähler durch ein externes CLR-/LATCH-Signal beeinflussen.

  Ein High-Pegel am CLR-/LATCH-Eingang kann je nach Programmierung ein Löschen (CLR) des Zählerstands oder die Übernahme des Zählerstands ins Latch (LATCH) bewirken. Die Funktion wird erst wirksam, wenn sie mit `P2_Cnt_Mode` freigegeben ist.

  Beim Latchen lässt sich aus der Differenz von zwei gelesenen Latch-Werten die Frequenz der Messung ermitteln, denn die Differenz gibt die Anzahl der Impulse zwischen den beiden Lesevorgängen an.

– Vierflankenauswertung (A- und B-Signale)

  Die Vierflankenauswertung wandelt die (möglichst um 90° phasenverschobenen) Signale eines angeschlossenen Inkremental-Encoders an A- und B-Eingang in ein CLK- und DIR-Signal um. Hierzu sind die Eingänge in *ADbasic* entsprechend zu programmieren (siehe „*ADwin-Pro* Systembeschreibung, Programmierung in *ADbasic* „).

Da jede Flanke des A- und B-Signales einen Zählimpuls erzeugt, wird die Auflösung um den Faktor 4 vergrößert. Besitzt der Encoder ein Referenz-Signal, so kann dies (nach Freigabe des CLR- bzw. LATCH-Einganges) zum Löschen oder Latchen des Zählers genutzt werden. Das Löschen des Zählers erfolgt, wenn die Signale A, B und CLR auf logisch „1“ stehen (über Software umstellbar: Löschen, wenn nur das CLR-Signal auf logisch „1“ steht).

PWM-Zähler

Der PWM-Zähler des Zählerblocks wertet die Signale an den PWM-Eingängen aus. Sie können mit `P2_Cnt_Mode` den Eingangs-Pin und die auslösende Flanke festlegen.

Beachten Sie: Auf dem Modul *Pro II-CNT-T* sind die PWM-Eingangs-Pins A und B fest mit der Betriebsart Vierflankenauswertung des Vor-/ Rückwärtszählers verknüpft, die Pins CLK und DIR mit der Betriebsart Takt/Richtungs-Auswertung.

Mit Standard-Befehlen können folgende Daten direkt gelesen werden:

– Frequenz und Tastverhältnis (`P2_Cnt_Get_PW`)

– Eintastzeit und Austastzeit (`P2_Cnt_Get_PW_HL`)

Ausnahme: PWM-Register selbst auswerten

Zu jedem PWM-Zähler gehören mehrere Register, die im folgenden beschrieben werden. Wenn Sie die PWM-Zähler wie im Beispiel mit den Standard-Befehlen auswerten, benötigen Sie keine Kenntnisse über die PWM-Register. Nur für spezielle Lösungen ist es sinnvoll, wenn Sie die PWM-Register selbst auswerten.

Um PWM-Signale auszuwerten zu können, werden in Latch-Registern die Zählerstände für das aktuelle und zwei vorhergehende PWM-Signale gespeichert, sowohl für steigende als auch für fallende Flanken. Für die Auswertung gibt es für jedes der 6 Register ein sogenanntes „Schattenregister“: die Werte aller 6 Register können gleichzeitig in die 6 Schattenregister kopiert werden und stehen dort für Berechnungen zur Verfügung.

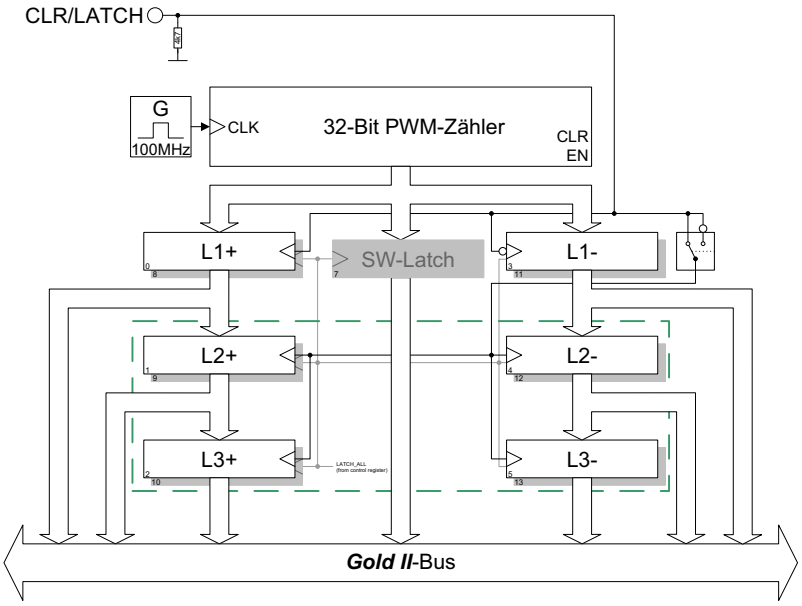
Register	Latch	Schatten-register
Latch 1 für positive Flanken (aktuell)	L1+	SL1+
Latch 2 für positive Flanken	L2+	SL2+
Latch 3 für positive Flanken	L3+	SL3+
Latch 1 für negative Flanken (aktuell)	L1–	SL1–
Latch 2 für negative Flanken	L2–	SL2–
Latch 3 für negative Flanken	L3–	SL3–

Die Registerwerte werden bei einer Flanke wie folgt geändert:

– Steigende Flanke:
  • Zählerstand nach L1+ kopieren
  • Wenn die steigende Flanke als Referenzflanke eingestellt ist:
    Register L2+ nach L3+ kopieren
    Register L1+ nach L2+ kopieren
    Register L2– nach L3– kopieren
    Register L1– nach L2– kopieren

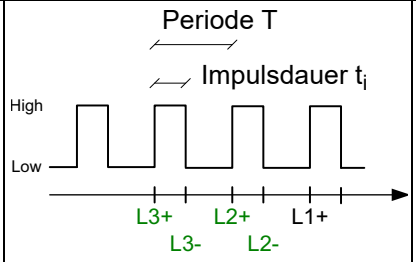
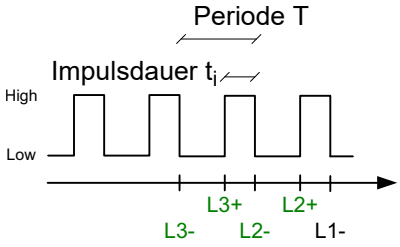
– Fallende Flanke:
  • Zählerstand nach L1– kopieren
  • Wenn die fallende Flanke als Referenzflanke eingestellt ist:
    Register L2– nach L3– kopieren
    Register L1– nach L2– kopieren
    Register L2+ nach L3+ kopieren
    Register L1+ nach L2+ kopieren

Zusätzlich gibt es ein einzelnes Latch-Register, in das der Zählerstand per Software (Befehl **P2_Cnt_PW_Latch**) kopiert wird.



Bei der Auswertung werden immer die PWM-Register der Ebenen 2 und 3 verwendet. Zunächst werden alle Registerwerte mit **P2_Cnt_Sync_Latch** in die Schattenregister kopiert und anschließend ausgewertet.

Die Berechnung ist abhängig von der eingestellten Referenzflanke:

Parameter	steigende Flanke	fallende Flanke
Schema		
Periode	T = L2+ – L3+	T = L2- – L3-
Impulsdauer	t_H = L3- – L3+	t_H = L2- – L3+
Pausendauer	$t_L = T - t_H$ = L2+ – L3-	$t_L = T - t_H$ = L3+ – L3-
Frequenz	f = 1 / T = 1 / (L2+ – L3+)	f = 1 / T = 1 / (L2- – L3-)
Tastverhältnis	g = t_H / T = (L3- – L3+) / (L2+ – L3+)	g = t_H / T = (L2- – L3+) / (L2- – L3-)

SSI-Decoder

Die Modulvariante *Pro II-CNT-D* enthält 2 SSI-Decoder zum Anschluss jeweils eines Inkremental-Encoders mit SSI-Schnittstelle. Die Signale sind differentiell und haben RS422/485-Pegel.

Ein Decoder kann entweder (auf Anforderung) einen einzelnen Wert auslesen oder aber kontinuierlich den aktuellen Wert bereit stellen.

Die SSI-Decoder sind per Software einstellbar:

– Taktraten sind einstellbar von 6,1kHz…12,5MHz mit **SSI_Set_Clock**.

– Auflösung ist einstellbar bis 32 Bit mit **SSI_Set_Bits**.

```
REM Beispiel: Umrechnung von Gray- in Binär-Code
REM PAR_1 = zu wandelnder Gray-Wert
REM PAR_2 = Flag für einen neuen Gray-Wert
REM PAR_9 = Ergebnis der Gray-zu-Binär-Wandlung

Dim m, n As Long

Event:
  If (Par_2 = 1) Then           'Start der Wandlung
    m = 0                       'Variable initialisieren
    Par_9 = 0                   ' -"-
    For n = 1 To 32             'Alle 32 Bits durchgehen
      m = (Shift_Right(Par_1,(32-n)) And 1) XOr m
      Par_9 = (Shift_Left(m,(32-n))) Or PAR_9
    Next n
    Par_2 = 0                   'Nächste Wandlung ermöglichen
  EndIf
```

EVENT-Eingang

Dieser Eingang kann, sofern er freigeben wurde, einen extern getriggerten *ADbasic*-Prozess starten.

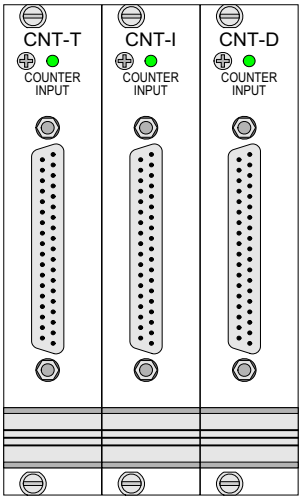


Abb. 126 – Pro II-CNT-x Rev. E: Frontplatten

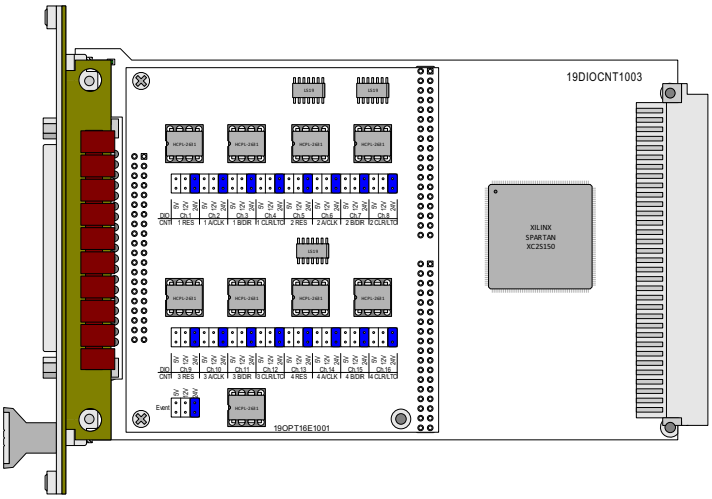
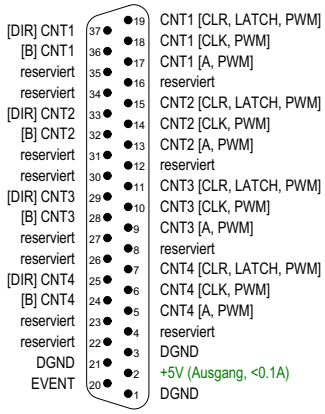
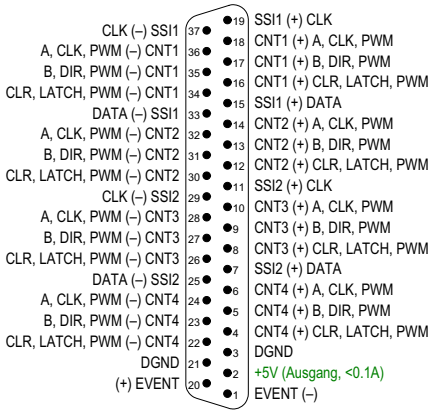


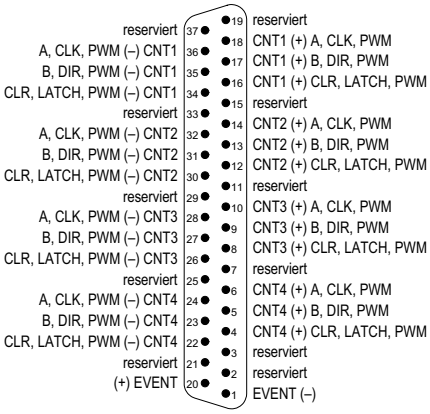
Abb. 127 – Pro II-CNT-I Rev. E: Platine mit Jumpern



Pinbelegung Pro II-CNT-T



Pinbelegung Pro II-CNT-D



Pinbelegung Pro II-CNT-I

Zähler	4 Universalzähler CNT-D: zusätzl. 2 SSI-Decoder
Zählerbreite	32 Bit
Eventeingang	1
Referenztakt	50 MHz
Taktfrequenz Vierflankenauswertung	12,5 MHz max. (bei 90° Phasen-verschiebung der Signale)
Taktfrequenz Vor- / Rückwärtszähler	CNT-T: 25 MHz max. CNT-I und CNT-D: 15 MHz max.
Referenzfrequenz PWM-Analyse	100 MHz
Steckerverbindung	37-pol. D-Sub-Buchse
Strombedarf des Moduls	CNT-T ca. 150 mA CNT-I ca. 200 mA CNT-D ca. 200 mA
TiCo-Prozessor	Prozessortyp: TiCo1 Taktfrequenz: 50 MHz Speichergröße: 28 KiB PM intern, 28 kiB DM intern

Abb. 128 – Pro II-CNT-x Rev. E: Allg. Spezifikation

Pro II-CNT-T			
Ein-/Ausgangspegel	TTL-Logik		
Event-Eingang	TTL-Logik		
Isolation	keine		
Pro II-CNT-D			
Ein-/Ausgangspegel	RS422/485 kompatibel (5V differentiell, 120 Ω Bus-Abschlusswiderstand)		
Event-Eingang	1 differentiell (single ended-Betrieb möglich)		
Taktfrequenz SSI-Decoder (CLK)	6kHz … 12,5MHz		
Isolation	keine		
Pro II-CNT-I			
Eingangsstrom	typ. 3,5mA / max. 7,5mA		
Eingangs-Spannungsbereich (über Jumper wählbar)	0…5V	0 … 12V	0…24V
sichere Schaltschwelle[1] für 0 (low)	0…0,8V	0…1,6V	0…3,2V
sichere Schaltschwelle[1] für 1 (high)	4,5…5V	10…12V	20…24V
Spannungsfestigkeit	8V	16V	30V
negative Spannung	-5V für alle Bereiche		
Schaltzeit	100ns		
Isolation	42V Kanal zu Kanal / Kanal zu Masse		

Abb. 129 – Spezifikationen der Modulvarianten

Programmierung

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Im Verzeichnis `C:\ADwin\ADbasic\samples_ADwin_ProII` finden Sie das Beispielprogramm `ProII-CNT_reciprocal_counting.bas`.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Zähler konfigurieren	**P2_Cnt_Enable**, **P2_Cnt_Mode**
Zähler ansteuern	**P2_Cnt_Clear**, **P2_Cnt_Get_Status** **P2_Cnt_Latch** **P2_Cnt_Read**, **P2_Cnt_Read4** **P2_Cnt_Read_Latch** **P2_Cnt_Read_Latch4**
PWM-Zähler ansteuern	**P2_Cnt_PW_Enable**, **P2_Cnt_PW_Latch** **P2_Cnt_Get_PW**, **P2_Cnt_Get_PW_HL**
SSI-Decoder ansteuern (nur CNT-D)	**P2_SSI_Mode**, **P2_SSI_Set_Bits** **P2_SSI_Set_Clock** **P2_SSI_Set_Delay**, **P2_SSI_Read** **P2_SSI_Read2** **P2_SSI_Start**, **P2_SSI_Status**
Event-Eingang ansteuern	**P2_Event_Enable**, **P2_Event_Config** **P2_Event_Read**
LED ansteuern	**P2_Check_LED**, **P2_Set_LED**

1. Innerhalb der angegebenen Spannungsbereiche wird ein low-/high-Signal sicher erkannt. Der Schaltvorgang kann jedoch bereits außerhalb dieser Spannungsbereiche erfolgen.

Bereich	Befehle
Synchronisieren	P2_Sync_All

Programmierung in TiCoBasic

Das Modul kann mit *TiCoBasic*-Befehlen programmiert werden. Die Befehle sind in der Online-Hilfe *TiCoBasic* erläutert.

Die Include-Datei `Cnt_TiCo.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Zähler konfigurieren	Cnt_Enable, Cnt_Mode
Zähler ansteuern	Cnt_Clear, Cnt_Get_Status Cnt_Latch, Cnt_Sync_Latch Cnt_Read, Cnt_Read_Latch Cnt_Read_Int_Register
PWM-Zähler ansteuern	Cnt_PW_Enable, Cnt_PW_Latch Cnt_Get_PW_HL
SSI-Decoder ansteuern (nur CNT-D)	SSI_Mode, SSI_Set_Bits SSI_Set_Clock SSI_Set_Delay, SSI_Read SSI_Start, SSI_Status
Event-Eingang ansteuern	Event_Enable, Event_Config Trigger_Event
LED ansteuern	Check_LED, Set_LED

Programmierung TiCo-Zugriff

Für den Zugriff auf den *TiCo*-Prozessor von der ADwin CPU sind die folgenden *ADbasic*-Befehle in der Datei `ADwinPro_All.inc` definiert. Die Befehle sind im Handbuch *TiCoBasic* und in der Online-Hilfe *ADbasic* erläutert.

Bereich	Befehle
Datenaustausch mit dem *TiCo*-Prozessor über globale Variablen	P2_TDrv_Init P2_GetData_Long, P2_Get_Par, P2_Get_Par_Block P2_SetData_Long, P2_Set_Par, P2_Set_Par_Block P2_Get_TiCo_RingBuffer, P2_Set_TiCo_RingBuffer P2_RingBuffer_Empty P2_RingBuffer_Full
TiCo-Prozessor steuern	P2_TiCo_Reset, P2_TiCo_Start, P2_TiCo_Stop P2_Get_TiCo_Bootloader_Status P2_Get_TiCo_Status, P2_Workload
TiCo-Prozesse steuern	P2_Process_Status P2_TiCo_Get_Processdelay P2_TiCo_Set_Processdelay P2_TiCo_Start_Process P2_Tico_Stop_Process
TiCo-Programme übertragen	P2_TiCo_Flash, P2_TiCo_Load

5.8 Pro II: Zusatz- und Schnittstellenmodule

Modul	RTD-8	TC-8-ISO
Revision	E	E
Funktion	Pt-/Ni-Temperaturmessung	Thermoelement-Schnittstelle
Messtechnik	2-Leiter, 3-Leiter oder 4-Leiter	–
Temperaturbereich	-200°C…+700°C	B: 250°C … 1820°C E: -200°C…1000°C J: -210°C…1250°C K: -200°C … 1372°C N: -200°C…1300°C R: -50°C … 1768°C S: -50°C … 1768°C T: -270°C … 400°C
Genauigkeit	±0,05…0,1K	±1K
Kanäle	8	8
Seite	149	153

Modul	CAN-2	CAN-2-LS	LIN-2	CAN-FD-2
Revision	E	E	E	E
Funktion	CAN-Schnittstelle, mit TiCo1-Prozessor		Schnittstelle LIN-Bus	CAN FD-Schnittstelle mit TiCo1-Prozessor
Typ	High speed	Low speed	–	–
Schnittstellen	2		2	2
Seite	158		178	164

Modul	Profi-SL	Profi-SL-40	PROFI-IRT-40
Revision	E	E	E
Funktion	Schnittstelle Profibus	Schnittstelle Profibus	Schnittstelle Profinet-IRT
Schnittstellen	1	1	1
Anschlüsse	–	–	Cu (Kupfer)/ FO (LWL)
TiCo-Prozessor	–	TiCo1	TiCo1
Seite	181	185	189

Modul	RSxxx-2	RSxxx-4	RS422-4	SG-4/18
Revision	E	E	E	E
Funktion	RSxxx-Schnittstelle, mit TiCo1-Prozessor		RS422-Schnittstelle	DMS-Schnittstelle
Typ	RS232 RS485	RS232 RS485	RS422	DMS
Schnittstellen	2	4	4	4
Übertragungsrate [kBaud]	0,035 … 2304		0,035… 2304	–
Seite	171		175	155

Modul	MIL-1553	ARINC-429	SPI-2-T	SPI-2-D
Revision	E	E	E	E
Funktion	Schnittstelle MIL-Bus	Schnittstelle ARINC-Bus, mit TiCo1-Prozessor	SPI-Master/ SPI-Slave, mit TiCo1-Prozessor	SPI-Master/ SPI-Slave, mit TiCo1-Prozessor
Signaltyp	–	–	TTL	differentiell
Schnittstellen	2	3	2	2
Übertragungsrate [kBaud]	1 MBit/s	100 oder 12,5 kBit/s	–	–
Seite	194	197	226	226

Modul	FlexRay-2	EtherCAT-SL	EtherCAT-SL-40	LS-Bus
Revision	E	E	E	E
Funktion	Schnittstelle FlexRay	Schnittstelle EtherCAT	Schnittstelle EtherCAT	Schnittstelle LS-Bus
Schnittstellen	2	1	1	2
TiCo-Prozessor	–	–	TiCo1	–
Seite	201	203	207	234

Modul	SENT-4	SENT-6	SENT-4-Out
Revision	E	E	E
Funktion	Schnittstelle SENT	Schnittstelle SENT	Schnittstelle SENT
Schnittstellen	4 Eingänge	6 Eingänge	4 Ausgänge
Digitalkanäle	–	4 Ein, 4 Aus	4 Ein, 4 Aus
Seite	212	217	222

5.8.1 Pro II-RTD-8 Rev. E

Das Modul Pro II-RTD-8 Rev. E hat 8 Eingänge zum Anschluss von Platin-Temperaturfühlern vom Typ Pt 100, Pt 500, Pt 1000 oder Ni 100. Die Eingänge sind über einen Multiplexer an einen ADC angebunden. Der mögliche Messbereich ist -200°C…+700°C, je nach eingesetztem Temperaturfühler (siehe Datenblatt des Herstellers).

Es gibt 2 Modulvarianten:

– Pro II-RTD-8 Rev. E: Moduleingänge auf geschirmten Lemo-Buchsen.

– Pro II-RTD-8-D Rev. E: Moduleingänge auf der 37-poligen D-Sub-Buchse.

Messungen können für jeden Kanal separat in 2-, 3- oder 4-Leitertechnik durchgeführt werden; Eingangsbeschaltung siehe Abb. 130. Die Messmethode und der Fühlertyp werden per Software eingestellt.
Die Messmethoden und die Verkabelung zwischen Sensor und Modul sind ab Seite 150 beschrieben; für LEMO-Stecker siehe Abb. 131.

Messungen werden entsprechend den Einstellungen automatisch durch eine Ablaufsteuerung durchgeführt, dabei verbinden Multiplexer die Messleitungen mit dem jeweiligen Sensor und trennen sie nach der Messung wieder. Danach liegen die Messwerte im Modul bereit, so dass die *ADwin* CPU die Messwerte nur noch abholen muss.

Die Eingangssignale durchlaufen einen Tiefpassfilter 2. Ordnung mit 25 kHz. Zusätzlich können Sie eine einzelne Frequenz aus den Digitalsignalen mit dem Befehl **`P2_RTD_Channel_Config`** herausfiltern; hierbei wird der Messwert als Mittelwert aus mehreren, in definierten Zeitabständen erfassten Messungen berechnet.

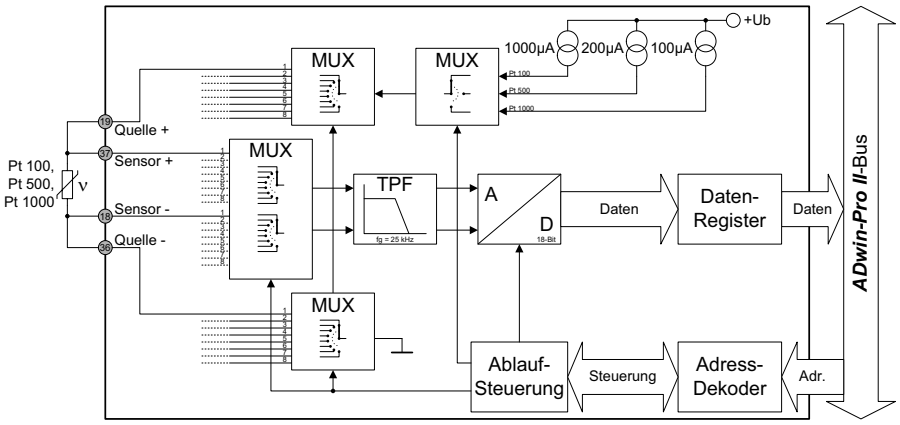


Abb. 130 – Pro II-RTD-8 Rev. E: Blockschaltbild

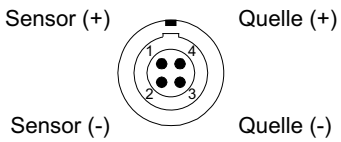


Abb. 131 – Pro II-RTD-8-L Rev. E: Pinbelegung LEMO-Stecker

Eingänge	8
Messtechnik	2-Leiter, 3-Leiter oder 4-Leiter
Multiplexer Einschwingzeit	100 µs
Eingangsfilter	25 kHz (2. Ordnung)
max. Messbereich	-200°C…+700°C

Abb. 132 – Pro II-RTD-8 Rev. E: Spezifikation

Genauigkeit	PT100: ±0,05K PT500: ±0,1K PT1000: ±0,1K Ni100: ±0,05K
Auflösung	0,015K
Temperaturdrift	±10ppm/K
$I_1 = I_2$	PT100: 1mA PT500: 0,2mA PT1000: 0,1mA Ni100: 1mA
Steckerverbindung	37-polige D-Sub-Buchse oder geschirmte Lemo-Buchsen

Abb. 132 – Pro II-RTD-8 Rev. E: Spezifikation

Programmierung

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Modul konfigurieren	`P2_RTD_Config` `P2_RTD_Channel_Config`

Temperatureingänge ansprechen	`P2_RTD_Start`
	`P2_RTD_Status`
	`P2_RTD_Read`
	`P2_RTD_Read8`

Messwert umrechnen	`P2_RTD_Convert`
LEDs einstellen	`P2_Check_LED`, `P2_Set_LED`

Messmethoden

Sie haben per Software 3 Messmethoden zur Auswahl: 2-Leiter-Messung, 3-Leiter-Messung oder 4-Leiter-Messung. Wir empfehlen die 4-Leiter-Messung, weil sie die genaueste Messmethode ist.

– 2-Leiter-Messung

  Achten Sie auf eine sehr kurze und niederohmige Verbindung zwischen dem Pt-Sensor und dem Moduleingang, weil der Spannungsabfall über die Messleitungen additiv in die gemessene Spannung eingeht.

  Aus diesem Grunde ist diese Messmethode für präzise Messungen generell nicht zu empfehlen.

  Für eine 2-Leiter Messung am Kanal `n` verbinden Sie den Sensor mit den Eingängen `Quelle +` und `Quelle -`. Für Kanal 1 wären dies die Pins 19 und 36 (siehe Abb. 133 bzw.) auf der D-Sub-Buchse.

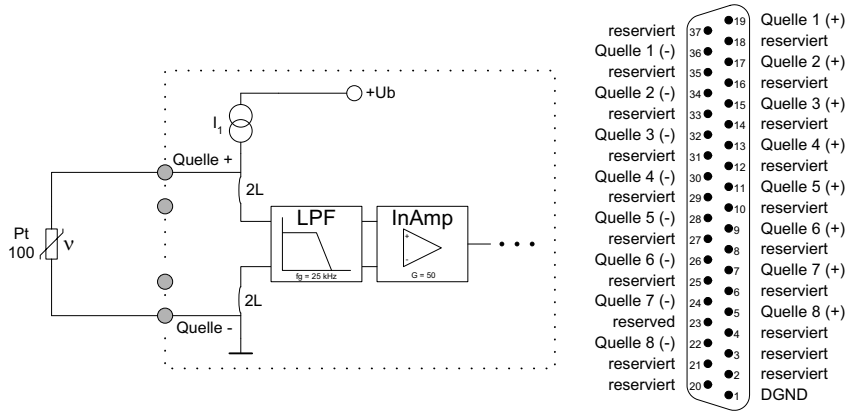


Abb. 133 – Pro II-RTD-8 Rev. E:
Schaltung und Pinbelegung bei 2-Leiter-Messung

– 3-Leiter-Messung

Um die Nachteile der 2-Leiter-Messung zu umgehen, wird hier der Spannungsabfall auf den Messleitungen mittels der zweiten Stromquelle I2 kompensiert.

Um den Messfehler so gering wie möglich zu halten, sollte der Widerstandswert der 3 Messleitungen vom Pt-Sensor zum Moduleingang identisch sein.

Bei der 3-Leiter-Messung werden immer 2 Messungen durchgeführt, um einen Messwert zu erhalten. Daher braucht eine 3-Leiter-Messung doppelt so viel Zeit wie die 2-Leiter- oder 4-Leiter-Messung.

Für eine 3-Leiter Messung am Kanal `n` verbinden Sie den Sensor mit den Eingängen `Quelle +`, `Sensor -` und `Quelle -`. Für Kanal 1 wären dies die Pins 18, 19 und 36 (siehe Abb. 134) auf der D-Sub-Buchse.

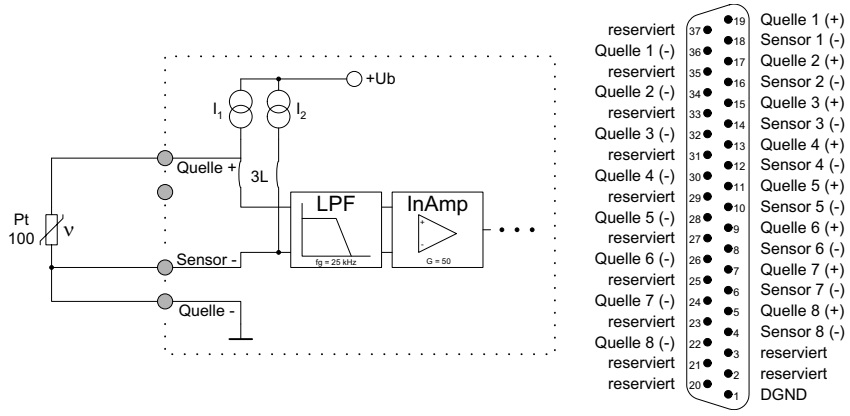


Abb. 134 – Pro II-RTD-8 Rev. E:
Schaltung und Pinbelegung bei 3-Leiter-Messung

– 4-Leiter-Messung

Der Spannungsabfall über den Pt-Sensor wird direkt am Platinelement durch die beiden „Sensor“-Eingänge hochohmig abgegriffen. Die Widerstände der Messleitungen gehen hier nicht mehr in die Messung ein und bedürfen damit auch keiner Kompensation.

Für eine 4-Leiter Messung am Kanal `n` verbinden Sie den Sensor mit den Eingängen `Quelle +`, `Sensor +`, `Sensor -` und `Quelle -`. Für Kanal 1 wären dies die Pins 18, 19, 36 und 37 (siehe Abb. 135) auf der D-Sub-Buchse.

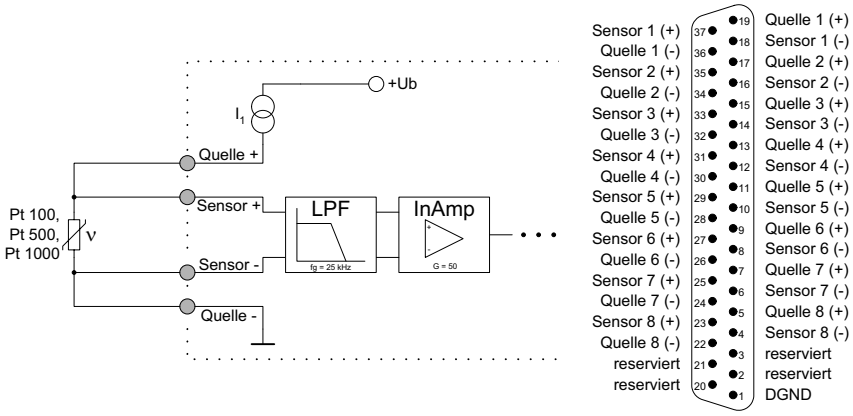


Abb. 135 – Pro II-RTD-8 Rev. E:
Schaltung und Pinbelegung bei 4-Leiter-Messung

5.8.2 Pro II-TC-8 ISO Rev. E

Das Modul Pro II-TC-8 ISO Rev. E hat 8 Eingänge für Thermoelemente und kann mit den Thermoelement-Typen B, E, J, K, N, R, S und T betrieben werden. Über Software-Befehle können Sie die Thermospannung (mit oder ohne Kaltstellenkompensation) oder die Temperatur jedes Kanals separat abfragen.

Jeder Eingang ist mit einem separaten ADC ausgerüstet. Das Modul hat eine gemeinsame Kaltstellenkompensation für alle Temperatureingänge.

Sie können die Kanaltrennung über die Position der Steckbrücken (Jumper, siehe Abb. 138 links) für jeden Kanal getrennt einstellen:

– Position rechts: Die Potentiale eines Eingangs sind voneinander getrennt (Voreinstellung).

– Position links: Der negative Eingang ist mit Erde verbunden.

Die Signalerfassung an den ADC erfolgt mit einer schrittweise einstellbaren Abtastrate. Sobald per Software ein Messwert abgefragt wird, wird aus dem zuletzt erfassten Messsignal die Thermospannung oder den Temperaturwert in Grad Celsius oder in Grad Fahrenheit berechnet. Die Berechnung basiert auf der Grundwertreihe der Norm IEC 584-1. Die Thermospannung ohne Kaltstellenkompensation kann auch zurückgegeben werden.

Die Eingänge haben einen Butterworth-Filter mit 5 Hz als Tiefpass. Das Modul kann auch ohne Tiefpass bestellt werden.

Die Kalibrierung des Moduls erfolgt beim Hersteller. Senden Sie das Modul hierzu an die Lieferanschrift auf der Rückseite der Titelseite.

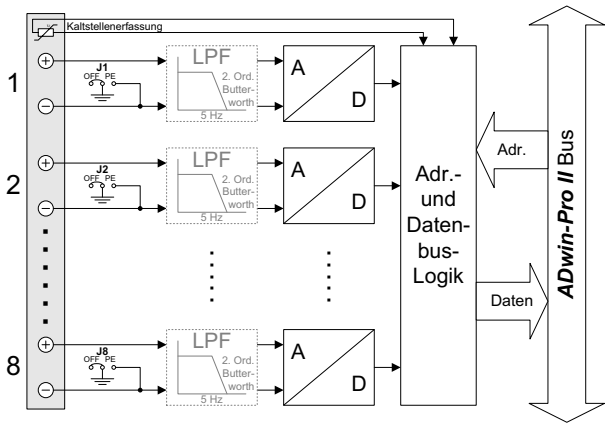


Abb. 136 – Pro II-TC-8 ISO Rev. E: Blockschaltbild

Eingangskanäle	8
Abtastrate	7 Hz … 3500 Hz
Thermoelement-Typen, Messbereiche und Genauigkeit	B: 250 °C … 1820 °C; ±5 °C E: -200 °C…1000 °C; ±1 °C J: -210 °C…1250 °C; ±1 °C K: -200 °C … 1372 °C; ±1 °C N: -200 °C…1300 °C; ±2 °C R: -50 °C … 1768 °C; ±3 °C S: -50 °C … 1768 °C; ±3 °C T: -270 °C … 400 °C; ±1 °C
Auflösung	0,1 °C
Eingangswiderstand	10 MΩ
Eingangsfilter	5 Hz Butterworth

Abb. 137 – Pro II-TC-8 ISO Rev. E: Spezifikation

Spannungsfestigkeit	±20 V
Offsetdrift	±30 ppm/°C vom Endwert
Steckerverbindung	Omega Subminiature Connector, Typ SMP

Abb. 137 – Pro II-TC-8 ISO Rev. E: Spezifikation

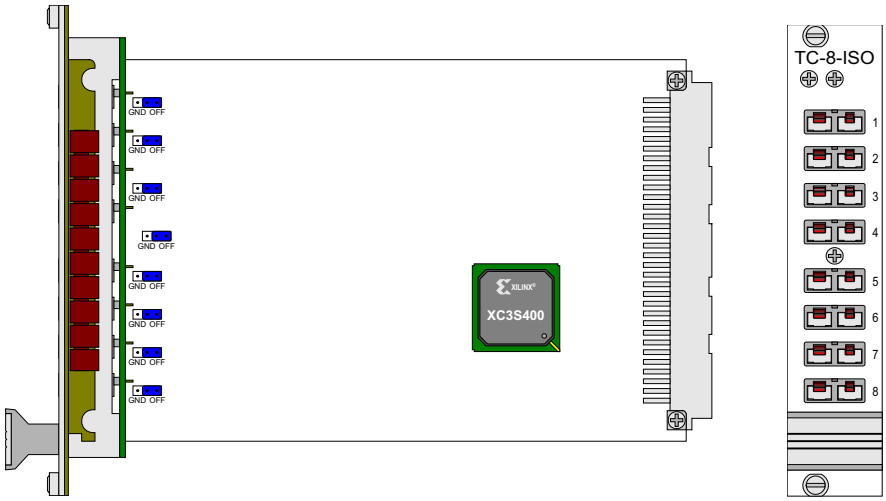


Abb. 138 – Pro II-TC-8 ISO Rev. E: Platine und Frontplatte

Programmierung

Mit folgenden Befehlen wird das Modul programmiert:

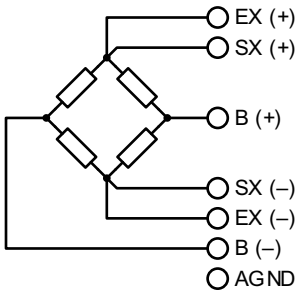
Abtastrate einstellen	`P2_TC_Set_Rate`
Eingangswerte in Latches kopieren	`P2_TC_Latch`, `P2_Sync_All`
Messwerte lesen	`P2_TC_Read_Latch` `P2_TC_Read_Latch4`, `P2_TC_Read_Latch8`

5.8.3 Pro II-SG-4/18 Rev. E

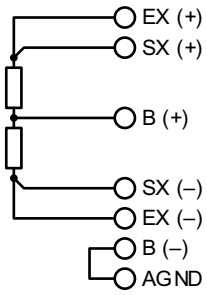
Das Modul Pro II-SG-4/18 Rev. E hat 4 Eingänge für Dehnungsmessstreifen (DMS) sowie 2 Analogeingänge und ist für Druck- und Zugkräfte geeignet. Alle Eingänge sind differentiell und über einen Multiplexer mit einem 18 Bit-ADC verbunden.

Es können DMS als Halbbrücke oder Vollbrücke (nicht als Viertelbrücke) angeschlossen werden sowie als 6-Leiter- oder als 4-Leiter-Schaltung.

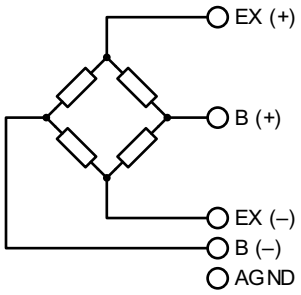
Der Mindest-Nennwiderstand der DMS hängt von der Versorgungsspannung EX ab; bei EX=9,80 V ist der Mindestwiderstand 400 Ω, bei EX=1,25 V sind es 50 Ω..



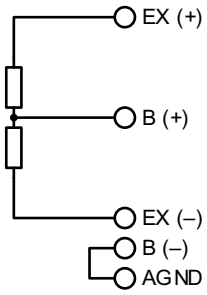
Vollbrücke, 6-Leiter



Halbbrücke, 6-Leiter



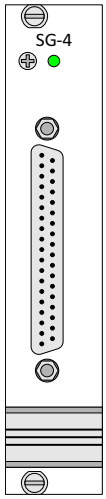
Vollbrücke, 4-Leiter



Halbbrücke, 4-Leiter

Abb. 139 – Pro II-SG-4/18 Rev. E: DMS-Schaltungen

Das Modul Pro II-SG-4/18 Rev. E stellt die Versorgungsspannung EX für die DMS-Schaltung zur Verfügung; die Versorgungsspannung kann – für jeden DMS-Kanal separat – per Software eingestellt werden. An jeder der 4 DMS-Schaltungen können bis zu 3 Spannungen gemessen werden: Versorgungsspannung EX (am Modul), Sense-Leitung SX und Brückenspannung B. Zusätzlich können die 2 Analogeingänge AIN13/AIN14 gemessen werden.



Alle Kontakte sind auf eine 37-polige D-Sub-Buchse geführt; Pinbelegung siehe Abb. 140. Die AGND-Kontakte sind intern miteinander verbunden.

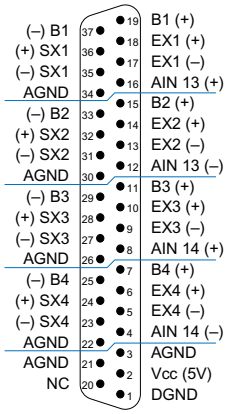


Abb. 140 – Pro II-SG-4/18 Rev. E: Pinbelegung

Der Verstärkungsfaktor für die Brückenspannung B wird per Software stufenweise (20 … 640) eingestellt; damit sind auch Messbereich und Signalauflösung festgelegt. Andere Signale werden mit Verstärkungsfaktor 1 gemessen.

Verstärkung	Messbereich Brückenspannung	Auflösung
20	±500 mV	3,82 µV
40	±250 mV	1,92 µV
80	±125 mV	0,96 µV
160	±62,5 mV	0,48 µV
320	±31,25 mV	0,24 µV
640	±15,625 mV	0,12 µV

Für den Abgleich des Nullsignals einer DMS kann die Brückenspannung per Software um bis zu ±10 mV angepasst werden. Zusätzlich kann der Verstärkungsfaktor einer DMS per Software abgeglichen werden; dazu muss auf die DMS eine definierte Last aufgebracht wird.

Die Brückensensitivität hängt von der Versorgungsspannung EX und dem Verstärkungsfaktor ab:

Spannung EX	Verstärkung 20	Verstärkung 640
1,25 V	400 mV/V	12,50 mV/V
2,50 V	200 mV/V	6,25 mV/V
3,75 V	133 mV/V	4,17 mV/V
5,00 V	100 mV/V	3,13 mV/V
6,25 V	80 mV/V	2,50 mV/V
7,50 V	66 mV/V	2,08 mV/V
8,75 V	57 mV/V	1,79 mV/V
9,80 V	51 mV/V	1,59 mV/V

Für jeden Kanal sind mehrere Filter vorhanden:

– am Eingang 1 MHz
– nach der ersten Verstärkerstufe 2 kHz, 2.Ordnung
– nach der zweiten Verstärkerstufe 2 kHz oder 200 Hz (2. Ordung), umschaltbar per Software.

Die Kalibrierung des Moduls erfolgt beim Hersteller. Falls erneut erforderlich, senden Sie das Modul an die Lieferanschrift auf der Rückseite der Titelseite.

DMS-Eingangskanäle Analogeingänge	4 (differentiell) 2 (differentiell)
Mindest-Nennwiderstand DMS	je nach EX: 50Ω…400Ω
Auflösung Analogmessung ADC-Wandlerzeit	18 Bit 2µs
Messbereich DMS Messbereich Analogeingänge	max. ±500mV; min. ±15,625mV ±10V
Multiplexer Einschwingzeit	5µs
Brückensensitivität	1,59mV/V … 400mV/V
Eingangswiderstand	>10MΩ
Eingangsfilter	2kHz oder 200Hz
Spannungsfestigkeit	±5V
Versorgungsspannung	max. 25mA je Kanal, kurzschlussfest
Steckerverbindung	D-Sub-Buchse, 37-polig

Abb. 141 – Pro II-SG-4/18 Rev. E: Spezifikation

Programmierung

Mit folgenden Befehlen wird das Modul programmiert:

Bereich	Befehle
Initialisieren	`P2_SG_Mode`, `P2_SG_Init`, `P2_SG_Zero`, `P2_SG_Set_Gain`
Messung durchführen	`P2_SG_Start`, `P2_SG_Wait`, `P2_SG_Read`
Messwert umrechnen	`P2_SG_Convert`
LEDs einstellen	`P2_Check_LED`, `P2_Set_LED`

5.8.4 Pro II-CAN-2 Rev. E

Das Modul Pro II-CAN-2 besitzt 2 CAN-Schnittstellen, die als „high speed“ oder als „low speed“-Variante ausgeführt sind. Die Bezeichnungen der Modulvarianten lauten:

– Pro II-CAN-2 Rev. E: CAN-Schnittstelle high speed

– Pro II-CAN-2-LS Rev. E: CAN-Schnittstelle low speed

Jedes Modul besitzt einen frei programmierbaren *TiCo*-Prozessor, der vollen Zugriff die CAN-Schnittstellen hat. Nähere Hinweise zur Anwendung und Programmierung des *TiCo*-Prozessors entnehmen Sie bitte dem Handbuch *TiCoBasic*.

Die Beschreibung des CAN-Moduls ist in folgende Abschnitte gegliedert:

CAN-Controller

Die CAN-Schnittstelle ist mit dem CAN-Controller AN82527 von Intel® bestückt und arbeitet nach der Spezifikation „CAN 2.0 part A+B“ sowie ISO 11898. Sie programmieren die Schnittstelle mit *ADbasic*-Befehlen, die direkt auf die Register des Controllers zugreifen.

Über den CAN-Bus verschickte Nachrichten sind Datentelegramme mit bis zu 8 Bytes, die durch sogenannte „Identifier“ gekennzeichnet sind. Der CAN-Controller unterstützt Identifier mit 11 Bit und 29 Bit Länge. Die eigentliche Kommunikation, d.h. die Verwaltung der Bus-Nachrichten, erfolgt über 15 „Message-Objekte“.

Zur Konfiguration und Statusanzeige des CAN-Controllers dienen die in ihm enthaltenen Register. Hier werden Busgeschwindigkeit, Interrupt handling usw. eingestellt (siehe separate Dokumentation „82527 - Serial Communications Controller, Architectural Overview“ von Intel®).

Der CAN-Bus (high speed) ist auf Frequenzen bis 1 MHz einstellbar und wird standardmäßig mit 1 MHz betrieben; bei CAN low speed beträgt die max. Frequenz 125 kHz. Der CAN-Bus ist durch Optokoppler vom *ADwin*-System galvanisch getrennt.

Der Eingang einer Nachricht kann einen Interrupt auslösen, der sofort einen Event am Prozessor erzeugt. Dadurch kann eine sofortige Bearbeitung der Nachrichten gewährleistet werden.

TiCo-Prozessor

Das Modul besitzt einen frei programmierbaren *TiCo*-Prozessor mit 28 KiB Programmspeicher und 28 KiB Datenspeicher. Den *TiCo*-Prozessor programmieren Sie in *TiCoBasic*.

Der *TiCo*-Prozessor hat – wie auch die *ADwin*-CPU – Zugriff auf den CAN-Controller. Nähere Hinweise zur Anwendung und Programmierung des *TiCo*-Prozessors entnehmen Sie bitte dem Handbuch *TiCoBasic*.

Wenn Sie ein *TiCoBasic*-Programm im *TiCo*-Bootloader ablegen, wird das Programm beim Einschalten der Stromversorgung in den *TiCo*-Prozessor geladen und gestartet. Auf diese Weise kann das Modul eigenständig und unabhängig vom CPU-Modul des *ADwin-Pro II*-Systems arbeiten.

Hardware-Aufbau

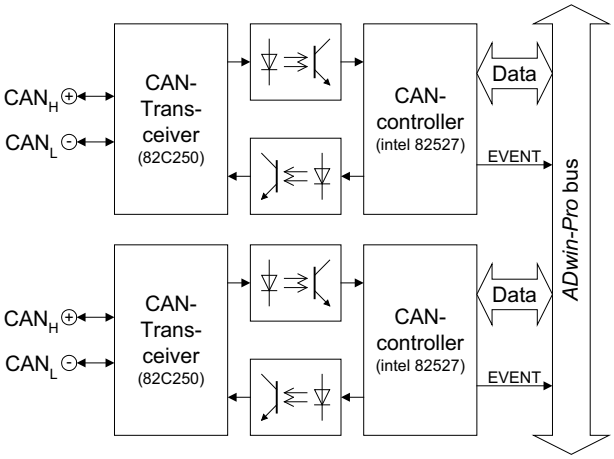


Abb. 142 – Pro II-CAN-2 Rev. E: Blockschaltbild

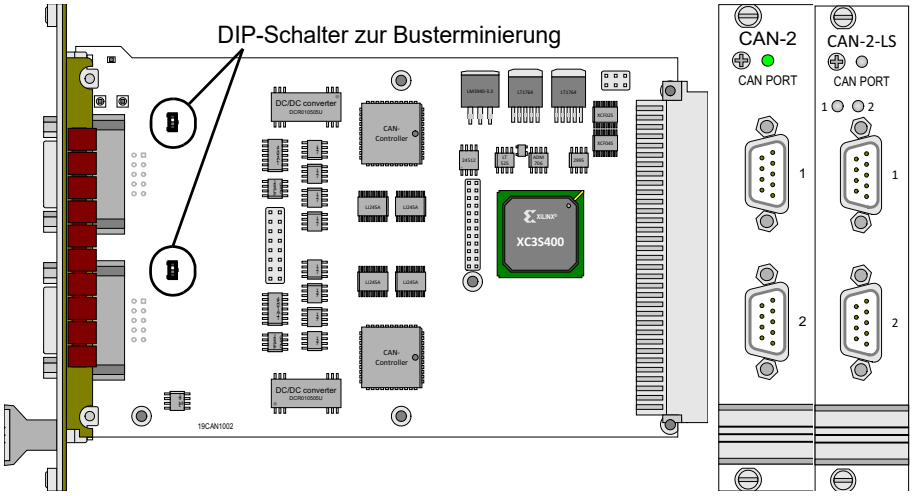


Abb. 143 – Pro II-CAN-2 Rev. E: Platine und Frontplatte

Die Anschlüsse der CAN-Bus Schnittstelle stehen auf einem 9-poligen D-Sub-Verbinder zur Verfügung; die Pin-Belegung ist unten dargestellt.

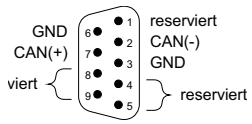
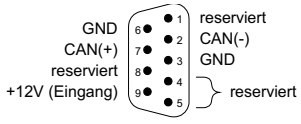
CAN-2	CAN-2-LS
	

Abb. 144 – Pro II-CAN-2: Pinbelegungen (Stecker)

Spannungsversorgung (nur Low speed)

Die „low speed“-Variante Pro II-CAN-2-LS benötigt eine externe Spannungsversorgung mit 12V Gleichstrom. Die Spannung muss für jeden Controller separat eingespeist werden.

Bei CAN „low-speed“ müssen die DIP-Schalter zur Busterminierung (siehe Abb. 143) nach unten gestellt sein, also ausgeschaltet.

Bus-Terminierung (nur High speed)

Wenn das CAN-Modul das physikalische Ende eines CAN-Bus vom Typ „High speed“ bildet, muss es mit einem Abschlusswiderstand 120Ω terminiert werden (also nur am ersten oder letzten CAN-Knoten). An CAN-Knoten, die sich nicht an einem physikalischen Ende der Kette befinden, darf nicht terminiert werden.

Wenn die Terminierung erforderlich ist, legen Sie den entsprechenden DIP-Schalter (siehe Abb. 143) nach oben um.

Nachrichten verwalten

Identifier

Der CAN-Controller unterscheidet über den Bus verschickte Nachrichten durch „Identifier“, das sind Kennzahlen mit einer definierten Bitlänge. Aus der Bitlänge ergeben sich hier die möglichen Kennzahlen 0...2^{11}-1 bzw. 0...2^{29}-1.

Message-Objekte

Jede Nachricht (zu sendende oder zu empfangende) speichert der Controller in einem von 15 „Message-Objekten“. Die Message-Objekte können jeweils entweder zum Senden oder zum Empfangen konfiguriert werden. Als Ausnahme kann das Message-Objekt 15 nur zum Empfangen genutzt werden. Nach der Initialisierung des CAN-Controllers sind sämtliche Message-Objekte nicht konfiguriert und beteiligen sich nicht am Busverkehr.

Jedes Message-Objekt erhält einen Identifier, der die Zuordnung einer Nachricht zu einem Message-Objekt ermöglicht.

Nachricht übergeben

In *ADbasic* übergeben Sie eine Nachricht an ein Message-Objekt über das Feld **`can_msg`**, das 8 Datenbytes plus die Anzahl der Datenbytes aufnehmen kann (9 Elemente). Ebenso wird eine Nachricht beim Auslesen aus einem Message Objekt in das Feld **`can_msg`** übertragen.

Nachricht senden

Das Versenden einer Nachricht läuft in folgenden Schritten ab:

– Sie konfigurieren ein Message-Objekt zum Senden und definieren den Identifier des Objekts (Befehl **`En_Transmit`**).

– Sie speichern die Nachricht im Feld **`can_msg`**.

– Sie senden die Nachricht (Befehl **`Transmit`**). Die Nachricht im Feld **`can_msg`** wird an das Message-Objekt übergeben. Sobald der Bus frei ist, wird die Nachricht gesendet (mit dem Identifier des Message-Objekts).

Nachricht empfangen

Das Empfangen einer Nachricht läuft in folgenden Schritten ab:

– Sie konfigurieren ein Message-Objekt für Empfang und definieren den Identifier des Objekts (Befehl **`En_Receive`**).

– Der Controller überwacht den CAN-Bus auf eingehende Nachrichten und speichert Nachrichten mit dem richtigen Identifier in dem Message-Objekt.

– Sie übertragen die Nachricht aus dem Message-Objekt in das Feld **`can_msg`** (Befehl **`Read_Msg`**) und lesen den zugehörigen Identifier aus.

Eine eingehende Nachricht überschreibt die alten Daten in dem Message-Objekt, die dadurch unwiderruflich verloren sind. Achten Sie daher beim Programmieren darauf, dass die Daten schneller ausgelesen als empfangen werden. Ein Datenverlust wird durch ein Flag angezeigt.

Bei dem Message Objekt 15 existiert ein zusätzlicher interner Zwischenspeicher, so dass dort 2 Nachrichten gespeichert werden können.

Nachricht zuordnen

Die Zuordnung einer eingehenden Nachricht zu einem Message-Objekt wird automatisch durch einen Vergleich ihrer Identifier gesteuert. Die globale Maske (CAN-Register 6...7 bzw. 6...9) steuert diesen Vergleich:

– Der Identifier der Nachricht wird bitweise mit dem Identifier des Message-Objekts verglichen. Wenn die relevanten Bits gleich sind, wird die Nachricht in das Message-Objekt übernommen. Nicht relevante Bits

werden nicht verglichen, d.h. die Nachricht wird (sofern es von diesem Bit abhängt) in das Objekt übernommen.

– Relevante Bits werden in der globalen Maske festgelegt, indem sie dort gesetzt werden.

Globale Maske

Durch die globale Maske kann ein Message-Objekt für den Empfang von Nachrichten mit **verschiedenen Identifiern** (ID) genutzt werden. Das folgende Beispiel zeigt die Zuordnung der Nachrichten-ID 1...4 zu den Message-Objekt-ID 1...4, wenn alle Bits der globalen Maske gesetzt sind bis auf die beiden niederwertigsten (bei einem 11-Bit-Identifier also `11111111100b`).

Nachrichten-ID	ID des Message-Objekts			
	1 …001b	2 …010b	3 …011b	4 …100b
1 (…001b)	x	x	x	0
2 (…010b)	x	x	x	0
3 (…011b)	x	x	x	0
4 (…100b)	0	0	0	x

x: Nachricht wird übernommen
0: Nachricht wird nicht übernommen

In diesem Beispiel entscheidet nur der Vergleich des Bits 2 über die Zuordnung, denn die Bits 3...10 der hier verglichenen Identifier sind identisch (= 0) und die Bits 0 und 1 werden nicht verglichen, weil sie in der globalen Maske auf Null gesetzt sind (= nicht relevant).

Busfrequenz einstellen

Die **CAN-Bus-Frequenz** hängt von der Konfiguration des Controllers ab.

Bei der Initialisierung mit `Init_CAN` wird der Controller automatisch so konfiguriert, dass die CAN-Bus-Frequenz 1 MHz beträgt. Soll der CAN-Bus mit einer anderen Frequenz betrieben werden, geschieht dies am einfachsten mit dem Befehl `Set_CAN_Baudrate`.

Bei CAN low speed muss die Busfrequenz auf Werte ≤125 kBit/s eingestellt werden.

Busfrequenz für Sonderfälle

In Sonderfällen kann es vorteilhaft sein, die Einstellungen anders zu wählen, als es mit `Set_CAN_Baudrate` möglich ist. Zu diesem Zweck müssen bestimmte Register mit dem Befehl `Poke` gesetzt werden. Der Registeraufbau ist in der Dokumentation des Controllers beschrieben.

Interrupt freigeben / Event auslösen

Sie können bei einem Message-Objekt freigeben, ob es beim Eingang einer Nachricht einen Interrupt auslöst. Der Interrupt-Ausgang des CAN-Controllers ist intern mit dem Event-Eingang des Prozessors verbunden. Dadurch kann der Prozessor sofort auf eingehende Nachrichten reagieren, ohne den Nachrichteneingang kontrollieren zu müssen (Polling).

Sie können die Interrupts mehrerer Message-Objekte freigeben. Welches Objekt den Interrupt ausgelöst hat, kann aus dem Interrupt-Register (`5Fh`) ersehen werden: Es enthält die Nummer des auslösenden Message-Objekts. Wird das Interrupt-Flag (new message flag) im Message-Objekt zurückgesetzt, wird das Interrupt-Register aktualisiert. Wenn kein Interrupt mehr ansteht, wird das Register auf „0“ gesetzt. Ist während der Bearbeitung des ersten Interrupts ein weiterer aufgetreten, so wird dessen Quelle nun im Interrupt-Register angezeigt. Ein weiterer Hardware-Interrupt erfolgt in diesem Fall nicht.

Modul-Revisionen

Die Unterschiede der Revisionsstände sind nachfolgend dargestellt:

Revision	Ausgabe-datum	Änderungen zur Vorgänger-Version
E1		Erst-Version

Die Revisionsbezeichnung befindet sich auf der Frontseite.

Unterschiedliche Revisions-Buchstaben bedeuten unterschiedliche Modul-Eigenschaften und sind separat dokumentiert. Der Revisionsbezeichnung angehängt ist eine Zählnummer für interne Zwecke.

Programmierung

Programmierung in ADbasic

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
CAN-Controller initialisieren, Baudrate einstellen	`P2_Init_CAN` `P2_Set_CAN_Baudrate`
Message-Objekte übertragen	`P2_En_Receive`, `P2_Read_Msg` `P2_En_Transmit`, `P2_Transmit`
Register setzen und lesen	`P2_Set_CAN_Reg`, `P2_Get_CAN_Reg`
LEDs einstellen	`P2_Check_LED`, `P2_Set_LED`, `P2_CAN_Set_LED`
Interrupts und Event-Eingang einstellen	`P2_En_Interrupt`, `P2_Event_Enable`, `P2_Event_Config`, `P2_Event_Read`

Programmierung in TiCoBasic

Das Modul kann mit *TiCoBasic*-Befehlen programmiert werden. Die Befehle sind in der Online-Hilfe *TiCoBasic* erläutert.

Die Include-Datei `CAN_TiCo.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
CAN-Controller initialisieren, Baudrate einstellen	`Init_CAN` `Set_CAN_Baudrate`

Message-Objekte übertragen	`En_Receive`, `Read_Msg`
	`Read_Msg_Con`, `CAN_Msg`
	`En_Transmit`, `Transmit`
	`Transmit_Status`

Register setzen und lesen	`Set_CAN_Reg`, `Get_CAN_Reg`
LEDs einstellen	`Check_LED`, `Set_LED` `CAN_Set_LED`
Interrupts einstellen	`En_Interrupt`

Programmierung TiCo-Zugriff

Für den Zugriff auf den *TiCo*-Prozessor von der ADwin CPU sind die folgenden *ADbasic*-Befehle in der Datei `ADwinPro_All.inc` definiert. Die Befehle sind im Handbuch *TiCoBasic* und in der Online-Hilfe *ADbasic* erläutert.

Bereich	Befehle

Datenaustausch mit dem *TiCo*-Prozessor über globale Variablen	`P2_TDrv_Init`
	`P2_GetData_Long`, `P2_Get_Par`, `P2_Get_Par_Block`
	`P2_SetData_Long`, `P2_Set_Par`, `P2_Set_Par_Block`
	`P2_Get_TiCo_RingBuffer`, `P2_Set_TiCo_RingBuffer`
	`P2_RingBuffer_Empty`
	`P2_RingBuffer_Full`

TiCo-Prozessor steuern	`P2_TiCo_Reset`, `P2_TiCo_Start`, `P2_TiCo_Stop`
	`P2_Get_TiCo_Bootloader_Status`
	`P2_Get_TiCo_Status`, `P2_Workload`

TiCo-Prozesse steuern	`P2_Process_Status`
	`P2_TiCo_Get_Processdelay`
	`P2_TiCo_Set_Processdelay`
	`P2_TiCo_Start_Process`
	`P2_Tico_Stop_Process`

TiCo-Programme übertragen	`P2_TiCo_Flash`, `P2_TiCo_Load`

5.8.5 Pro II-CAN-FD-2 Rev. E

Das Modul Pro II-CAN-FD-2 besitzt 2 CAN FD-Schnittstellen.

Jedes Modul besitzt einen frei programmierbaren *TiCo*-Prozessor, der vollen Zugriff die CAN FD-Schnittstellen hat. Nähere Hinweise zur Anwendung und Programmierung des *TiCo*-Prozessors entnehmen Sie bitte dem Handbuch *TiCoBasic*.

Die Beschreibung des CAN FD-Moduls ist in folgende Abschnitte gegliedert:

CAN FD-Controller

Die CAN-Schnittstelle ist mit dem CAN-Controller MCP2517FD von Microchip® bestückt und arbeitet nach der Spezifikation ISO 11898-1.Sie programmieren die Schnittstelle mit *ADbasic*-Befehlen, die auf die Register des Controllers zugreifen.

Der Controller unterstützt sowohl CAN 2.0B (high speed) als auch CAN FD (auch im Mischbetrieb mit CAN 2.0B).

Über den CAN FD-Bus verschickte Nachrichten sind Datentelegramme mit bis zu 64 Byte Nutzdaten, die durch sogenannte „Identifier“ gekennzeichnet sind. Der Controller unterstützt Identifier mit 11 Bit, 12 Bit und 29 Bit Länge. Die eigentliche Kommunikation, d.h. die Verwaltung der Bus-Nachrichten, erfolgt über 32 Message-Objekte. Die Message-Objekte 1…31 arbeiten als Fifo (First in, first out) und können als Eingang oder Ausgang konfiguriert werden; das Message-Objekt mit der Nummer 0, der Ausgabepuffer (transmit queue), verarbeitet die Nachrichten in der Reihenfolge ihrer Priorität.

Der Controller ist auf Busfrequenzen bis 8 Mbps (Daten) bzw. bis 1 Mbps (Nachrichten-Header) einstellbar. Bei CAN 2.0B (high speed) sind Busfrequenzen bis 1 MHz möglich. Der CAN-Bus ist durch Optokoppler vom *ADwin*-System galvanisch getrennt.

TiCo-Prozessor

Das Modul besitzt einen frei programmierbaren *TiCo*-Prozessor mit 28 KiB Programmspeicher und 28 KiB Datenspeicher. Den *TiCo*-Prozessor programmieren Sie in *TiCoBasic*.

Der *TiCo*-Prozessor hat – wie auch die *ADwin*-CPU – Zugriff auf den CAN FD-Controller. Nähere Hinweise zur Anwendung und Programmierung des *TiCo*-Prozessors entnehmen Sie bitte dem Handbuch *TiCoBasic*.

Wenn Sie ein *TiCoBasic*-Programm im *TiCo*-Bootloader ablegen, wird das Programm beim Einschalten der Stromversorgung in den *TiCo*-Prozessor geladen und gestartet. Auf diese Weise kann das Modul eigenständig und unabhängig vom CPU-Modul des *ADwin-Pro II*-Systems arbeiten.

Hardware-Aufbau

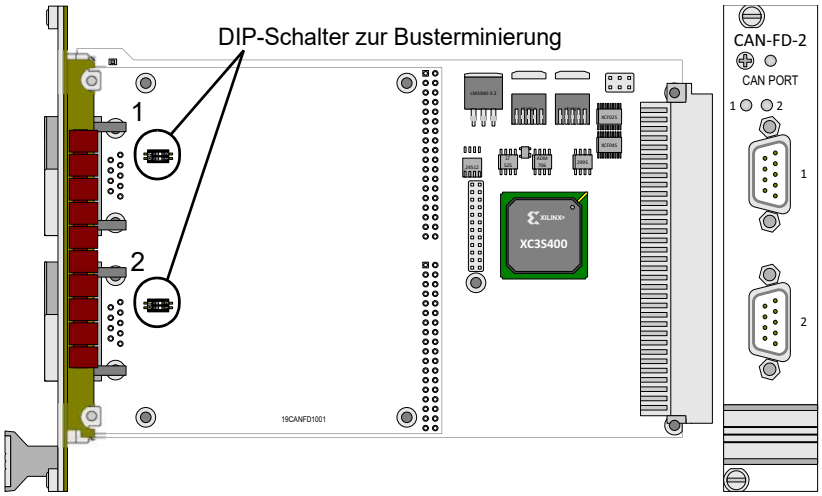


Abb. 145 – Pro II-CAN-FD-2 Rev. E: Platine und Frontplatte

Die Anschlüsse der CAN FD-Bus Schnittstelle stehen auf einem 9-poligen D-Sub-Verbinder zur Verfügung; die Pin-Belegung ist unten dargestellt.

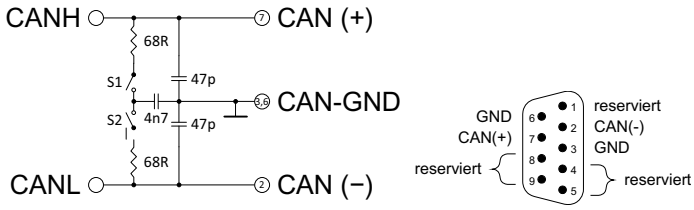


Abb. 146 – Pro II-CAN-FD-2 Rev. E: Schema und Pinbelegungen (Stecker)

Bus-Terminierung

Wenn das Modul das physikalische Ende eines CAN FD-Bus oder eines CAN-Bus vom Typ „High speed“ bildet, muss der Bus an dieser Stelle terminiert werden (also nur am ersten oder letzten CAN-Knoten). An CAN-Knoten, die sich nicht an einem physikalischen Ende der Kette befinden, darf nicht terminiert werden.

Sie schalten die Busterminierung ein, indem Sie beide DIP-Schalter für die betreffende CAN-Schnittstelle nach links legen (siehe Abb. 145).

Nachrichten verwalten

Identifier

Der CAN FD-Controller unterscheidet über den Bus verschickte Nachrichten durch „Identifier“, das sind Kennzahlen mit einer definierten Bitlänge. Aus der Bitlänge ergeben sich hier die möglichen Kennzahlen 0...2^{11}-1, 0...2^{12}-1 und 0...2^{29}-1.

Message-Objekte

Jede Nachricht (zu sendende oder zu empfangende) speichert der Controller in einem von 32 Message-Objekten. Jedes Message-Objekt kann bis zu 32 Nachrichten speichern. Die Message-Objekte 1…31 sind als Fifo aufgebaut und können jeweils entweder zum Senden (Ausgangs-Fifo) oder zum Empfangen (Eingangs-Fifo) konfiguriert werden. In einem Fifo (First in, first out) werden Nachrichten in der Reihenfolge abgerufen, in der sie gespeichert wurden.

Das Message-Objekt 0 arbeitet immer als Ausgabepuffer (transmit queue). Er verarbeitet die Nachrichten in der Reihenfolge ihrer Priorität. Die Nachricht mit dem niedrigsten Identifier hat die höchste Priorität.

Jedes Message-Objekt erhält einen Identifier, der die Zuordnung von Nachrichten ermöglicht. Über eine Maske (siehe Nachricht zuordnen) können einem Message-Objekt auch Gruppen von Identifiern zugeordnet werden.

Zur Kontrolle des Sendevorgangs kann ein Überwachungs-Fifo eingerichtet werden. Für jede gesendete Nachricht wird der Nachrichten-Header sowie optional ein Zeitstempel gespeichert.

Message-Objekte benötigen Speicherplatz im Controller. Es stehen insgesamt 2kB Speicher zur Verfügung, die frei auf die Message-Objekte und den Überwachungs-Fifo verteilt werden können.

Initialisierung

Für die Initialisierung wird ein Datenfeld angelegt, in dem die Eigenschaften des CAN FD-Controllers (Betriebsmodus, Busfrequenz, Zeitabgleich beim Senden) und die Eigenschaften aller verwendeten Message-Objekte (z.B. Größe, Identifier) abgelegt werden. Mit `P2_CANFD_Init_Controller` werden die Informationen aus dem Datenfeld in die entsprechenden Register des Controllers übertragen. Danach ist der Controller betriebsbereit.

Nachricht senden

Das Versenden einer Nachricht läuft in folgenden Schritten ab:

– Sie konfigurieren ein Message-Objekt zum Senden mit dem Befehl `P2_CANFD_Enable_Transmit_Fifo` oder `P2_CANFD_Enable_Transmit_Queue`. Anschließend initialisieren Sie den CAN FD-Controller mit `P2_CANFD_Init_Controller` (siehe Initialisierung).

– Sie übergeben eine Nachricht und den Identifier an den CAN-Controller (Befehl `P2_CANFD_Write_TMO`).

– Sie geben das Message-Objekt zum Senden frei (Befehl `P2_CANFD_Transmit_MSG`). Sobald der Bus frei ist, wird die Nachricht mit dem Identifier gesendet.

Beachten Sie: Wenn in mehreren Message-Objekten Nachrichten zum Senden bereitstehen, wird die Nachricht mit dem niedrigsten Identifier zuerst gesendet.

Beim Senden liest der Controller die Daten auf dem Bus zurück, um auf Konflikte gemäß Busprotokoll reagieren zu können. Beim Zurücklesen kann eine Verzögerung gegenüber den gesendeten Daten auftreten. Eine solche Verzögerung kann bei der Initialisierung des Controllers ausgeglichen werden (`P2_CANFD_Set_TDC`).

Gesendete Nachricht kontrollieren

Zur Kontrolle gesendeter Nachrichten können Sie einen Überwachungs-Fifo (transmit event fifo) einrichten. Für jede gesendete Nachricht werden im Überwachungs-Fifo der Header der gesendeten Nachricht sowie optional ein Zeitstempel gespeichert. Sie lesen ein solches Nachrichtenobjekt EFO aus mit `P2_CANFD_Read_EFO`.

Um auch Nachrichten mit gleichem Header nachverfolgen zu können, geben Sie beim Senden einer Nachricht einen Kennwert SEQ an. Nach dem Senden finden Sie den Kennwert im gespeicherten Header im Überwachungs-Fifo wieder.

Nachricht empfangen

Das Empfangen einer Nachricht läuft in folgenden Schritten ab:

– Sie konfigurieren ein Message-Objekt zum Empfangen und definieren den Identifier des Objekts (`P2_CANFD_Enable_Receive_Fifo`). Anschließend initialisieren Sie den CAN FD-Controller mit `P2_CANFD_Init_Controller` (siehe Initialisierung).

– Der Controller überwacht den CAN-Bus auf eingehende Nachrichten und speichert Nachrichten mit dem richtigen Identifier in dem Eingangs-Fifo.

– Sie lesen die Nachricht aus dem Eingangs-Fifo (Befehl `P2_CANFD_Read_RMO`).

Wenn alle Plätze des Eingangs-Fifos voll sind, überschreibt die nächste eingehende Nachricht die älteste Nachricht, die dadurch unwiderruflich verloren geht. Achten Sie daher beim Programmieren darauf, dass die Daten schneller ausgelesen als empfangen werden. Ein Datenverlust kann separat abgefragt werden.

Nachricht zuordnen

Die Zuordnung einer eingehenden Nachricht zu einem Eingangs-Fifo wird automatisch durch einen Vergleich der Identifier gesteuert. Für jeden Eingangs-Fifo gibt es eine Maske, die diesen Vergleich steuert:

– Der Identifier der Nachricht wird bitweise mit dem Identifier des Eingangs-Fifos verglichen. Wenn die relevanten Bits gleich sind, wird die Nachricht in das Eingangs-Fifo übernommen. Nicht relevante Bits werden nicht verglichen, d.h. die Nachricht wird (sofern es von diesem Bit abhängt) in das Eingangs-Fifo übernommen.

– Relevante Bits werden in der Maske festgelegt, indem sie dort gesetzt werden.

Durch die Maske kann ein Eingangs-Fifo für den Empfang von Nachrichten mit verschiedenen Identifiern (ID) genutzt werden. Das folgende Beispiel zeigt die Zuordnung der Nachrichten-ID 1...4 zu einem Eingangs-Fifo, bei dem alle Bits der Maske gesetzt sind bis auf die beiden niederwertigsten, bei einem 11-Bit-Identifier also `11111111100b`.

Nachrichten-ID	ID des Eingangs-Fifos			
	1 …001b	2 …010b	3 …011b	4 …100b
1 (…001b)	x	x	x	0
2 (…010b)	x	x	x	0
3 (…011b)	x	x	x	0
4 (…100b)	0	0	0	x

x: Nachricht wird übernommen
0: Nachricht wird nicht übernommen

In diesem Beispiel entscheidet nur der Vergleich des Bits 2 über die Zuordnung, denn die Bits 3...10 der hier verglichenen Identifier sind identisch (= 0) und die Bits 0 und 1 werden nicht verglichen, weil sie in der Maske auf Null gesetzt sind (= nicht relevant).

Modul-Revisionen

Die Unterschiede der Revisionsstände sind nachfolgend dargestellt:

Revision	Ausgabe-datum	Änderungen zur Vorgänger-Version
E1	April 2018	Erst-Version

Die Revisionsbezeichnung befindet sich auf der Frontseite.

Unterschiedliche Revisions-Buchstaben bedeuten unterschiedliche Modul-Eigenschaften und sind separat dokumentiert. Der Revisionsbezeichnung angehängt ist eine Zählnummer für interne Zwecke.

Programmierung in ADbasic

Programmierung

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle

CAN-Controller initialisieren, Baudrate einstellen	`P2_CANFD_Init_Datatable`
	`P2_CANFD_Init_Controller`
	`P2_CANFD_Enable_Receive_Fifo`
	`P2_CANFD_Enable_Transmit_Fifo`
	`P2_CANFD_Enable_Transmit_Queue`
	`P2_CANFD_Enable_Transmit_Event_Fifo`
	`P2_CANFD_Set_Baudrate_Nominal`
	`P2_CANFD_Set_Baudrate_Data`
	`P2_CANFD_Set_SID11`
	`P2_CANFD_Set_TDC`
	`P2_CANFD_Set_Mode`

Status prüfen	`P2_CANFD_Get_Fifo_State`

Nachricht empfangen	`P2_CANFD_Read_RMO`
	`P2_CANFD_Get_Header_Parts`
	`P2_CANFD_Get_ID`
	`P2_CANFD_Get_ESI`
	`P2_CANFD_Get_FDF`
	`P2_CANFD_Get_BRS`
	`P2_CANFD_Get_RTR`
	`P2_CANFD_Get_IDE`
	`P2_CANFD_Get_DLC`

Nachricht senden und überprüfen	`P2_CANFD_Write_TMO`
	`P2_CANFD_Transmit_MSG`
	`P2_CANFD_Transmit_Multi_MSG`
	`P2_CANFD_Read_EFO`
	`P2_CANFD_Get_SEQ`

Register lesen	`P2_CANFD_Get_TREC`
	`P2_CANFD_Get_BDIAG0`
	`P2_CANFD_Get_BDIAG1`

LEDs einstellen	`P2_Check_LED`, `P2_Set_LED` `P2_CANFD_Set_LED`

Programmierung in TiCoBasic

Das Modul kann mit *TiCoBasic*-Befehlen programmiert werden. Die Befehle sind in der Online-Hilfe *TiCoBasic* erläutert.

Die Include-Datei `CAN_FD_TiCo.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle

CAN FD-Controller initialisieren, Baudrate einstellen	`CANFD_Init_Datatable`
	`CANFD_Init_Controller`
	`CANFD_Enable_Receive_Fifo`
	`CANFD_Enable_Transmit_Fifo`
	`CANFD_Enable_Transmit_Queue`
	`CANFD_Enable_Transmit_Event_Fifo`
	`CANFD_Set_Baudrate_Nominal`
	`CANFD_Set_Baudrate_Data`
	`CANFD_Set_SID11`
	`CANFD_Set_TDC`
	`CANFD_Set_Mode`

Status prüfen	`CANFD_Get_Fifo_State`

Nachricht empfangen	`CANFD_Read_RMO`
	`CANFD_Get_Header_Parts`
	`CANFD_Get_ID`
	`CANFD_Get_ESI`
	`CANFD_Get_FDF`
	`CANFD_Get_BRS`
	`CANFD_Get_RTR`
	`CANFD_Get_IDE`
	`CANFD_Get_DLC`

Nachricht senden	`CANFD_Write_TMO`
	`CANFD_Transmit_MSG`
	`CANFD_Transmit_Multi_MSG`
	`CANFD_Read_EFO`
	`CANFD_Get_SEQ`

Register setzen und lesen	`CANFD_Get_TREC`
	`CANFD_Get_BDIAG0`
	`CANFD_Get_BDIAG1`

LEDs einstellen	`Check_LED`, `Set_LED` `CANFD_Set_LED`

Programmierung TiCo-Zugriff

Für den Zugriff auf den *TiCo*-Prozessor von der ADwin CPU sind die folgenden *ADbasic*-Befehle in der Datei `ADwinPro_All.inc` definiert. Die Befehle sind im Handbuch *TiCoBasic* und in der Online-Hilfe *ADbasic* erläutert.

Bereich	Befehle

Datenaustausch mit dem *TiCo*-Prozessor über globale Variablen	`P2_TDrv_Init`
	`P2_GetData_Long`, `P2_Get_Par`, `P2_Get_Par_Block`
	`P2_SetData_Long`, `P2_Set_Par`, `P2_Set_Par_Block`
	`P2_Get_TiCo_RingBuffer`, `P2_Set_TiCo_RingBuffer`
	`P2_RingBuffer_Empty`
	`P2_RingBuffer_Full`

TiCo-Prozessor steuern	`P2_TiCo_Reset`, `P2_TiCo_Start`, `P2_TiCo_Stop`
	`P2_Get_TiCo_Bootloader_Status`
	`P2_Get_TiCo_Status`, `P2_Workload`

TiCo-Prozesse steuern	`P2_Process_Status`
	`P2_TiCo_Get_Processdelay`
	`P2_TiCo_Set_Processdelay`
	`P2_TiCo_Start_Process`
	`P2_Tico_Stop_Process`

TiCo-Programme übertragen	`P2_TiCo_Flash`, `P2_TiCo_Load`

5.8.6 Pro II-RSxxx Rev. E

Das Modul Pro II-RSxxx besitzt 2 oder 4 Schnittstellen vom Typ RS-232 oder RS-485. Der Schnittstellentyp wird mit dem Befehl `P2_RS485_Send` eingestellt.

Die Bezeichnungen der Modulvarianten lauten:

– Pro II-RSx-2 Rev. E: 2 Schnittstellen RS232/485

– Pro II-RSx-4 Rev. E: 4 Schnittstellen RS232/485

Alle Modulvarianten der RSxxx-Module sind mit dem Controller „Quad Universal Asynchronous Receiver/Transmitter“ (UART) vom Typ TL16C754 der Firma Texas Intruments® bestückt. Die Funktionalität und Programmierung der Schnittstellen beruhen auf diesem Controller.

Jedes Modul besitzt einen frei programmierbaren *TiCo*-Prozessor, der vollen Zugriff die RSxxx-Schnittstellen hat. Nähere Hinweise zur Anwendung und Programmierung des *TiCo*-Prozessors entnehmen Sie bitte dem Handbuch *TiCoBasic*.

Der physikalische Unterschied zwischen den Schnittstellen-Typen liegt in den Pegeln der Signale, die auf dem „Bus“ durch entsprechende Treiber-Bausteine bereitgestellt werden.

Die Beschreibung ist in folgende Abschnitte gegliedert:

TiCo-Prozessor

Das Modul besitzt einen frei programmierbaren *TiCo*-Prozessor mit 28 KiB Programmpeicher und 28 KiB Datenspeicher. Den *TiCo*-Prozessor programmieren Sie in *TiCoBasic*.

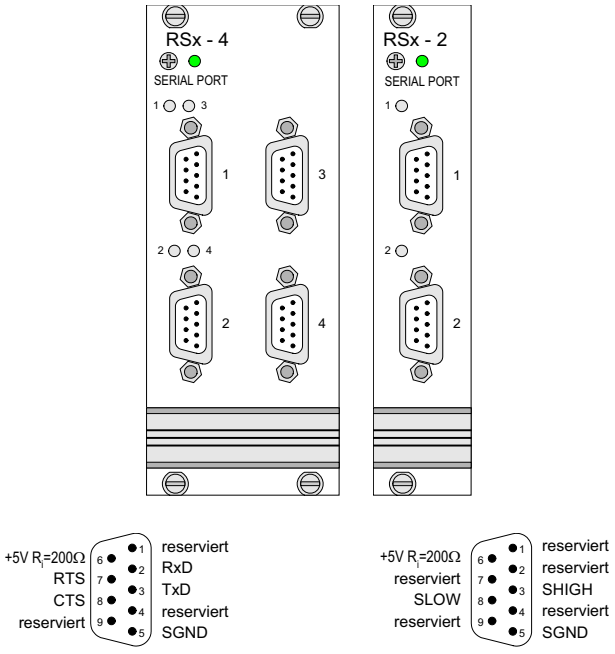
Der *TiCo*-Prozessor hat – wie auch die *ADwin*-CPU – Zugriff auf die RSxxx-Schnittstellen. Nähere Hinweise zur Anwendung und Programmierung des *TiCo*-Prozessors entnehmen Sie bitte dem Handbuch *TiCoBasic*.

Wenn Sie ein *TiCoBasic*-Programm im *TiCo*-Bootloader ablegen, wird das Programm beim Einschalten der Stromversorgung in den *TiCo*-Prozessor geladen und gestartet. Auf diese Weise kann das Modul eigenständig und unabhängig vom CPU-Modul des *ADwin-Pro II*-Systems arbeiten.

Hardware

Nachfolgend sind Frontplatten und Pin-Belegungen der Module Pro II-RSxxx gezeigt. Die Pinbelegung wird mit dem Schnittstellentyp umgeschaltet.

Die Modulvariante mit 2 Schnittstellen ist 5 TE breit, mit 4 Schnittstellen ist die Breite 10 TE.



RS-232 (Stecker) RS-485 (Stecker)

Abb. 147 – Pro II-RSxxx: Frontplatten / Pinbelegungen

Einstellbare Schnittstellen-Parameter

Jede Schnittstelle verfügt über einen Eingangs- und einen Ausgangs-FIFO mit einer Länge von jeweils 64 Byte. Die Einstellung der Schnittstellen-Parameter wird mit Hilfe der Controller-Register vorgenommen, und zwar getrennt für jeden Kanal. Im Folgenden werden die Einstellmöglichkeiten dargestellt:

Handshake

– Handshake: Die Schnittstelle kann in 2 Modi betrieben werden:

1. Ohne Handshake
2. Software-Handshake
3. Hardware-Handshake (nur RS232).
   Beim Hardware-Handshake müssen die Signale RTS und CTS angeschlossen sein.

Parität

– Parität: Um einen Fehler bei der Übertragung und damit fehlerhafte Daten erkennen zu können, kann ein Paritätsbit mit übertragen werden. Die Parität kann gerade oder ungerade sein, oder es kann auf das Paritätsbit verzichtet werden.

Daten-Bits

– Datenbits: Die Nutzdaten, die übertragen werden sollen, können aus 5…8 Bits bestehen.

Stopp-Bits

– Stopp-Bits: Die Anzahl der Stopp-Bits kann auf 1, 1½ oder 2 eingestellt werden. Dabei ist die Anzahl der Stoppbits von der Anzahl der Datenbits abhängig:
  - 5 Datenbits: 1 oder 1½ Stoppbits.
  - 6…8 Datenbits: 1 oder 2 Stoppbits.

Baudrate

– Baudrate: Die physikalisch erreichbaren Werte liegen zwischen 35 Baud und 2,304 MBaud; bei einer RS-232 Schnittstelle liegt die max. Baudrate laut Spezifikation bei 115,2 kBaud.

  Die einstellbaren Baudraten werden vom moduleigenen Taktgeber abgeleitet; der Grundtakt hat eine Frequenz von 2,304 MHz. Davon ausgehend ist jede Baudrate möglich, die sich durch ganzzahlige Division dieses Grundtakts ergibt. Der Teiler kann Werte im Bereich von `1…0FFFFh` annehmen. Die nachfolgende Tabelle zeigt einige gängige Baudraten und die zugehörigen Teiler.

Baudrate	Teiler	
	dez.	hex.
2304000	1	0001h
1152000	2	0002h
460800	5	0005h
230400	10	000Ah
115200	20	0014h
57600	40	0028h
38400	60	003Ch

Baudrate	Teiler	
	dez.	hex.
19200	120	0078h
9600	240	00F0h
4800	480	01E0h
2400	960	03C0h
1200	1920	0780h
600	3840	0F00h
300	7680	1E00h

Abb. 148 – Pro II-RSxxx: Gängige Baudraten

Besonderheiten RS485

Über eine RS485-Schnittstelle können – im Gegensatz zu RS232 – mehr als 2 Teilnehmer miteinander kommunizieren. So kann mit Hilfe von RS485-Schnittstellen ein Bus aufgebaut werden.

Daraus ergeben sich folgende Hinweise:

– Bei RS485 gibt es keinen Handshake, da ein Handshake immer nur zwischen zwei Teilnehmern funktioniert.

– Jeder Schnittstelle muss mit **RS485_SEND** mitgeteilt werden, ob sie auf den Bus schreiben soll oder nur Daten vom Bus übernehmen darf.

Modul-Revisionen

Die Unterschiede der Revisionsstände sind nachfolgend dargestellt:

Revision	Ausgabe-datum	Änderungen zur Vorgänger-Version
E	11/2007	Erst-Version

Programmierung

Programmierung in ADbasic

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Im Verzeichnis `C:\ADwin\ADbasic\samples_ADwin_ProII` finden Sie zusätzliche Beispielprogramme.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Initialisierung	**P2_RS_Init**, **P2_RS_Reset**
Empfangen und Senden von Daten	**P2_Read_Fifo**, **P2_Write_Fifo**
RS485-Kanal konfigurieren	**P2_RS485_Send**
Schreib- / Lesezugriff auf Controller-Register	**P2_Get_RS**, **P2_Set_RS**
LED ansteuern	**P2_Check_LED**, **P2_Set_LED** **P2_RS_Set_LED**

Programmierung in TiCoBasic

Das Modul kann mit *TiCoBasic*-Befehlen programmiert werden. Die Befehle sind in der Online-Hilfe *TiCoBasic* erläutert.

Die Include-Datei `RS_LIN_TiCo.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Initialisierung	**RS_Init**, **RS_Reset**

Bereich	Befehle
Empfangen und Senden von Daten	Read_Fifo, Write_Fifo Check_Shift_Reg
RS485-Kanal konfigurieren	RS485_Send
Schreib- / Lesezugriff auf Controller-Register	Get_RS, Set_RS
LED ansteuern	Check_LED, Set_LED RS_Set_LED

Programmierung TiCo-Zugriff

Für den Zugriff auf den *TiCo*-Prozessor von der ADwin CPU sind die folgenden *ADbasic*-Befehle in der Datei `ADwinPro_All.inc` definiert. Die Befehle sind im Handbuch *TiCoBasic* und in der Online-Hilfe *ADbasic* erläutert.

Bereich	Befehle
Datenaustausch mit dem *TiCo*-Prozessor über globale Variablen	P2_TDrv_Init P2_GetData_Long, P2_Get_Par, P2_Get_Par_Block P2_SetData_Long, P2_Set_Par, P2_Set_Par_Block P2_Get_TiCo_RingBuffer, P2_Set_TiCo_RingBuffer P2_RingBuffer_Empty P2_RingBuffer_Full
TiCo-Prozessor steuern	P2_TiCo_Reset, P2_TiCo_Start, P2_TiCo_Stop P2_Get_TiCo_Bootloader_Status P2_Get_TiCo_Status, P2_Workload
TiCo-Prozesse steuern	P2_Process_Status P2_TiCo_Get_Processdelay P2_TiCo_Set_Processdelay P2_TiCo_Start_Process P2_Tico_Stop_Process
TiCo-Programme übertragen	P2_TiCo_Flash, P2_TiCo_Load

5.8.7 Pro II-RS422-4 Rev. E

Das Modul Pro II-RS422-4 Rev. E besitzt 4 Schnittstellen vom Typ RS-422.

Das Modul ist mit dem Controller „Quad Universal Asynchronous Receiver/Transmitter“ (UART) vom Typ TL16C754 der Firma Texas Intruments® bestückt. Die Funktionalität und Programmierung der Schnittstellen beruhen auf diesem Controller.

Das Modul besitzt einen frei programmierbaren *TiCo*-Prozessor, der vollen Zugriff die RS422-Schnittstellen hat. Nähere Hinweise zur Anwendung und Programmierung des *TiCo*-Prozessors entnehmen Sie bitte dem Handbuch *TiCoBasic*.

Die Beschreibung ist in folgende Abschnitte gegliedert:

- TiCo-Prozessor
- Hardware
- Einstellbare Schnittstellen-Parameter
- Modul-Revisionen
- Programmierung

TiCo-Prozessor

Das Modul besitzt einen frei programmierbaren *TiCo*-Prozessor mit 28 KiB Programmpeicher und 28 KiB Datenspeicher. Den *TiCo*-Prozessor programmieren Sie in *TiCoBasic*.

Der *TiCo*-Prozessor hat – wie auch die *ADwin*-CPU – Zugriff auf die RS422-Schnittstellen. Nähere Hinweise zur Anwendung und Programmierung des *TiCo*-Prozessors entnehmen Sie bitte dem Handbuch *TiCoBasic*.

Wenn Sie ein *TiCoBasic*-Programm im *TiCo*-Bootloader ablegen, wird das Programm beim Einschalten der Stromversorgung in den *TiCo*-Prozessor geladen und gestartet. Auf diese Weise kann das Modul eigenständig und unabhängig vom CPU-Modul des *ADwin-Pro II*-Systems arbeiten.

Hardware

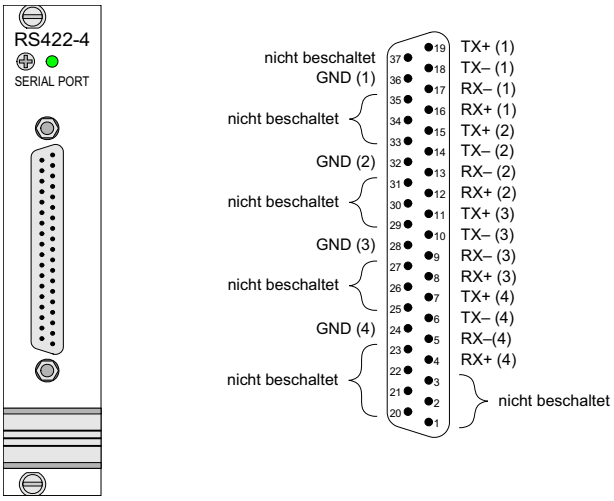


Abb. 149 – Pro II-RS422-4 Rev. E: Frontplatte / Pinbelegung

Einstellbare Schnittstellen-Parameter

Initialisieren Sie die Schnittstellen zuerst mit `P2_RS_Init` für den RS422-Betrieb.

Jede Schnittstelle verfügt über einen Eingangs- und einen Ausgangs-FIFO mit einer Länge von jeweils 64 Byte. Die Einstellung der Schnittstellen-Parameter wird mit Hilfe der Controller-Register vorgenommen, und zwar getrennt für jeden Kanal. Folgende Einstellungen sind möglich:

Handshake

– Handshake: Die Schnittstelle kann in 2 Modi betrieben werden:

1. Ohne Handshake
2. Software-Handshake

Parität

– Parität: Um einen Fehler bei der Übertragung und damit fehlerhafte Daten erkennen zu können, kann ein Paritätsbit mit übertragen werden. Die Parität kann gerade oder ungerade sein, oder es kann auf das Paritätsbit verzichtet werden.

Daten-Bits

– Datenbits: Die Nutzdaten, die übertragen werden sollen, können aus 5…8 Bits bestehen.

Stopp-Bits

– Stopp-Bits: Die Anzahl der Stopp-Bits kann auf 1, 1½ oder 2 eingestellt werden. Dabei ist die Anzahl der Stoppbits von der Anzahl der Datenbits abhängig:
  - 5 Datenbits: 1 oder 1½ Stoppbits.
  - 6…8 Datenbits: 1 oder 2 Stoppbits.

Baudrate

– Baudrate: Die physikalisch erreichbaren Werte liegen zwischen 35 Baud und 2,304 MBaud.

Die einstellbaren Baudraten werden vom moduleigenen Taktgeber abgeleitet; der Grundtakt hat eine Frequenz von 2,304 MHz. Davon ausgehend ist jede Baudrate möglich, die sich durch ganzzahlige Division dieses Grundtakts ergibt. Der Teiler kann Werte im Bereich von `1…0FFFFh` annehmen. Die nachfolgende Tabelle zeigt einige gängige Baudraten und die zugehörigen Teiler.

Baudrate	Teiler	
	dez.	hex.
2304000	1	0001h
1152000	2	0002h
460800	5	0005h
230400	10	000Ah
115200	20	0014h
57600	40	0028h
38400	60	003Ch

Baudrate	Teiler	
	dez.	hex.
19200	120	0078h
9600	240	00F0h
4800	480	01E0h
2400	960	03C0h
1200	1920	0780h
600	3840	0F00h
300	7680	1E00h

Abb. 150 – Pro II-RS422-4 Rev. E: Gängige Baudraten

Modul-Revisionen

Die Unterschiede der Revisionsstände sind nachfolgend dargestellt:

Revision	Ausgabe-datum	Änderungen zur Vorgänger-Version
E	11/2011	Erst-Version

Programmierung

Programmierung in ADbasic

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Initialisierung	`P2_RS_Init`, `P2_RS_Reset`

Bereich	Befehle
Empfangen und Senden von Daten	P2_Read_Fifo, P2_Write_Fifo
Schreib- / Lesezugriff auf Controller-Register	P2_Get_RS, P2_Set_RS
LED ansteuern	P2_Check_LED, P2_Set_LED P2_RS_Set_LED

Programmierung in TiCoBasic

Das Modul kann mit *TiCoBasic*-Befehlen programmiert werden. Die Befehle sind in der Online-Hilfe *TiCoBasic* erläutert.

Die Include-Datei `RS_LIN_TiCo.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Initialisierung	RS_Init, RS_Reset
Empfangen und Senden von Daten	Read_Fifo, Write_Fifo Check_Shift_Reg
Schreib- / Lesezugriff auf Controller-Register	Get_RS, Set_RS
LED ansteuern	Check_LED, Set_LED RS_Set_LED

Programmierung TiCo-Zugriff

Für den Zugriff auf den *TiCo*-Prozessor von der ADwin CPU sind die folgenden *ADbasic*-Befehle in der Datei `ADwinPro_All.inc` definiert. Die Befehle sind im Handbuch *TiCoBasic* und in der Online-Hilfe *ADbasic* erläutert.

Bereich	Befehle
Datenaustausch mit dem *TiCo*-Prozessor über globale Variablen	P2_TDrv_Init P2_GetData_Long, P2_Get_Par, P2_Get_Par_Block P2_SetData_Long, P2_Set_Par, P2_Set_Par_Block P2_Get_TiCo_RingBuffer, P2_Set_TiCo_RingBuffer P2_RingBuffer_Empty P2_RingBuffer_Full
TiCo-Prozessor steuern	P2_TiCo_Reset, P2_TiCo_Start, P2_TiCo_Stop P2_Get_TiCo_Bootloader_Status P2_Get_TiCo_Status, P2_Workload
TiCo-Prozesse steuern	P2_Process_Status P2_TiCo_Get_Processdelay P2_TiCo_Set_Processdelay P2_TiCo_Start_Process P2_Tico_Stop_Process
TiCo-Programme übertragen	P2_TiCo_Flash, P2_TiCo_Load

5.8.8 Pro II-LIN-2 Rev. E

Das Modul Pro II-LIN-2 Rev. E besitzt 2 LIN-Schnittstellen, die unabhängig voneinander als LIN-Master oder LIN-Slave konfiguriert werden können.

LIN-Schnittstelle

Die LIN-Schnittstellen des Moduls sind nach der Spezifikation "LIN 2.1" (Local Interconnect Network) vom November 2006 implementiert. Sie programmieren die LIN-Schnittstelle mit *ADbasic*-Befehlen.

LIN ist ein serielles Kommunikationsprotokoll auf einem Eindrahtbus mit einer Übertragungsgeschwindigkeit bis zu 20 KiBit/s. Der Bus ist besonders kostengünstig und wird für die Vernetzung mechatronischer Knoten in Fahrzeugen eingesetzt.

Über den LIN-Bus verschickte Nachrichten sind Datenpakete mit bis zu 8 Bytes Nutzdaten, die durch sogenannte "Identifier" gekennzeichnet sind. Die Verwaltung der Bus-Nachrichten auf dem LIN-Modul erfolgt über 64 "Message-Boxen".

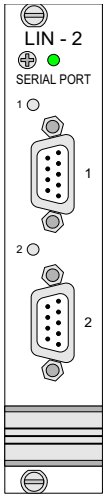
Das Buskonzept sieht einen einzelnen Master-Knoten und mehrere Slave-Knoten vor. Der Master kontrolliert den gesamten Datenverkehr auf dem Bus: Bei jedem Datenpaket sendet der Master zunächst einen Header mit dem Identifier des nächsten Datenpakets. Daraufhin reagieren diejenigen Knoten (das kann auch der Master selbst sein), die eine Messagebox mit diesem Identifier verwalten, indem sie ein Datenpaket senden oder ein auf dem Bus anstehendes Datenpaket empfangen.

Hardware-Aufbau

Die Anschlüsse der LIN-Bus Schnittstelle stehen auf zwei 9-poligen D-Sub-Verbindern zur Verfügung; die Pin-Belegung ist unten dargestellt.

Die Betriebsspannungs-Bereich für `+Bat` ist 8V…18V.

Für jede Schnittstelle ist eine programmierbare LED vorhanden.



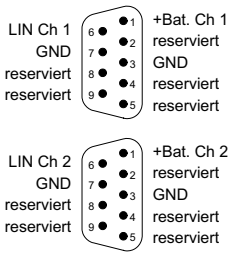


Abb. 151 – Pro II-LIN-2: Pinbelegungen

Mit LIN arbeiten

Die 2 LIN-Schnittstellen des Moduls können mit **`P2_LIN_Init_Write`** unabhängig voneinander als LIN-Master oder LIN-Slave konfiguriert werden; dies gilt auch für die Einstellung der Baudrate. Der Busabschluss wird je nach Konfiguration automatisch geschaltet.

Das Zeitverhalten des Masters (Netzwerkmanagement NMT) wird in *ADbasic* programmiert.

Beim Konfigurieren einer Messagebox mit **`P2_LIN_Msg_Write`** geben Sie den Identifier und den Sendestatus (Senden, Empfangen) der Messagebox an. Einer LIN-Schnittstelle können beliebig viele Messageboxen zugeordnet werden, jedoch darf auf dem LIN-Bus jeder Identifier nur einmal zum Senden und einmal zum Empfangen vergeben werden; anderenfalls entstehen Datenkollisionen.

Nach dem Konfigurieren einer Messagebox ist diese sofort auf dem LIN-Bus aktiv, d.h. es können Daten empfangen oder gesendet werden.

Wenn die LIN-Schnittstelle als LIN-Master konfiguriert ist, senden Sie mit **P2_LIN_Msg_Transmit** einen Header und den Identifier einer Messagebox auf den LIN-Bus. Die mit diesem Identifier konfigurierten Messageboxen am LIN-Bus reagieren darauf automatisch.

Die Messagebox eines Master-Knotens verhält sich anders als die eines Slave-Knotens:

- **Master-Knoten, Senden**: Der Master sendet sowohl den Header (siehe **P2_LIN_Msg_Transmit**) als auch gleich anschließend das Datenpaket der Messagebox.
- **Master-Knoten, Empfangen**: Der Master sendet den Header (siehe **P2_LIN_Msg_Transmit**) auf den LIN-Bus und wartet auf die Antwort des passenden Slaves. Das empfangene Datenpaket wird in die Messagebox eingetragen.
- **Slave-Knoten, Senden**: Der Slave wartet, bis der Master den Header mit dem zur Messagebox passenden Identifier sendet. Erst dann sendet der Slave-Knoten das Datenpaket.
- **Slave-Knoten, Empfangen**: Der Slave wartet, bis der Master den Header mit dem zur Messagebox passenden Identifier sendet, empfängt anschließend das Datenpaket und trägt es in die Messagebox ein.

Ab Rev. E04 kann das Modul Wakeup-Signale mit **P2_LIN_Msg_Transmit** senden (nur als LIN-Slave) und mit **P2_LIN_Read_Dat** erkennen (nur als LIN-Master).

Modul-Revisionen

Die Unterschiede der Revisionsstände sind nachfolgend dargestellt:

Revision	Ausgabe-datum	Änderungen zur Vorgänger-Version
E01		Erst-Version
E04	10/2019	Wakeup-Signale senden und erkennen

Die Revisionsbezeichnung befindet sich auf der Frontseite.

Unterschiedliche Revisions-Buchstaben bedeuten unterschiedliche Modul-Eigenschaften und sind separat dokumentiert. Der Revisionsbezeichnung angehängt ist eine Zählnummer für interne Zwecke.

Programmierung

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
LIN-Schnittstellen initialisieren und zurücksetzen	**P2_LIN_Init** **P2_LIN_Init_Write** **P2_LIN_Init_Apply** **P2_LIN_Reset**
Schnittstellenversion abfragen	**P2_LIN_Get_Version**
Daten lesen und senden LIN-Header senden Wakeup-Signale senden und erkennen	**P2_LIN_Read_Dat** **P2_LIN_Ch_Read_Cnt** **P2_LIN_Msg_Read_Status** **P2_LIN_Msg_Write** **P2_LIN_Msg_Transmit**
Modul-LED einstellen	**P2_Check_LED** **P2_Set_LED** **P2_LIN_Set_LED**

5.8.9 Pro II-Profi-SL Rev. E

Das Modul Pro II-Profi-SL Rev. E stellt einen Feldbusknoten mit der Funktionalität eines Profibus-Slave zur Verfügung. Alle Einstellungen werden per Software vorgenommen.

Das Modul Pro II-Profi-SL-40 Rev. E bietet einen größeren Eingangs- und Ausgangsbereich.

Funktionsbeschreibung

Nach dem Einschalten muss der Feldbusknoten initialisiert werden. Mit der Initialisierung werden die Stationsadresse (Slave-Knotenadresse) auf dem Profibus und die Größe der Ein- und Ausgangsbereiche festgelegt.

Es gibt je einen Bereich für eingehende und für ausgehende Daten; jeder Bereich hat eine maximale Größe von 76 Byte. Die Begriffe „Eingang“ und „Ausgang“ sind aus Sicht des Feldbus-Controllers zu sehen.

Sie legen für beide Bereiche separat fest, wieviele Datenbereiche in jedem Bereich vorhanden sind und wie groß die Datenbereiche sind.

Hardware

Die Pinbelegung der 9-poligen D-Sub-Buchse DP-V1 entspricht der DIN E 19245, Teil 3.

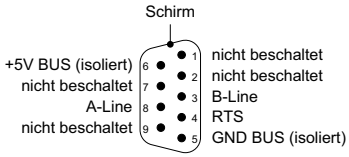


Abb. 152 – Pro II-Profi-SL Rev. E: Pinbelegung

Der Profibus muss am physikalischen Anfang und Ende der Busleitung mit einem Abschlusswiderstand abgeschlossen werden. Falls erforderlich, müssen Sie den Abschlusswiderstand selbst an den entsprechenden Datenleitungen des Feldbusknotens anbringen oder einen entsprechenden Steckverbinder verwenden.

Über und unter der D-Sub-Buchse befinden sich zwei LEDs, die den Betriebszustand des Knotens im Profibus anzeigen, nämlich Betriebsmodus (OP) und Status (ST).

LED	Status	Bedeutung
OP	aus	Nicht online oder keine Stromversorgung.
	grün	Feldbusknoten online, Datenaustausch.
	blinkt grün	Feldbusknoten online, Status Clear.
	blinkt rot, einfach	Fehler: Ein/Ausgangskonfiguration stimmt nicht mit der Masterkonfiguration überein.
	blinkt rot, doppelt	Fehler bei der Profibus-Konfiguration.
ST	aus	Nicht online oder keine Stromversorgung.
	grün	Initialisiert.
	blinkt grün	Initialisiert, Diagnosemeldung liegt an.
	rot	Ausnahmefehler.

Abb. 153 – Pro II-Profi-SL Rev. E: Bedeutung der LED

Profibus projektieren

Sie projektieren den Profibus mit einem – zum Bus-Master passenden – Konfigurations-Tool. Für das folgende Beispiel wurden ein Profibus-Master der Firma Hilscher und das zugehörige Programm `SyCon` verwendet.

Für andere Konfigurations-Tools gilt die folgende Ablaufbeschreibung entsprechend. Entnehmen Sie die genaue Vorgehensweise bei der Busprojektierung der Dokumentation Ihres Konfigurations-Tools.

GSD-Datei kopieren

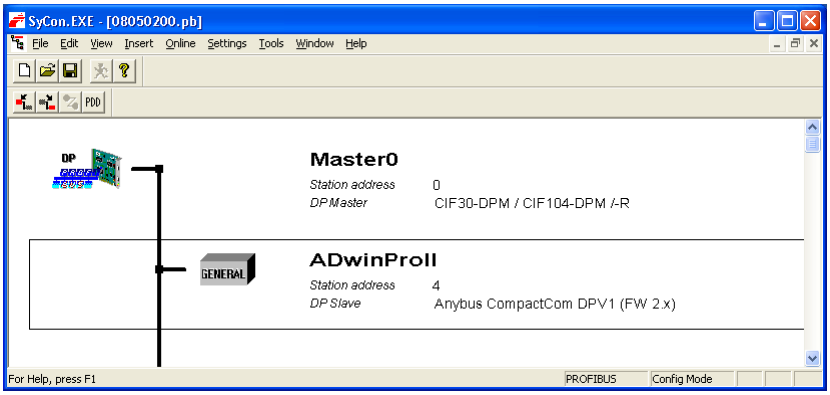
– Kopieren oder importieren Sie die GSD-Datei `hmsb1811.gsd` (Gerätestammdaten-Datei) des Feldbusknotens von `C:\ADwin\Fieldbus\Profibus` in das Quellverzeichnis des Konfigurations-Tools.

  Das Konfigurations-Tool lädt die benötigten Informationen über einen neuen Slave aus der zugehörigen GSD-Datei; der Dateiinhalt ist durch die EN 50170 festgelegt. Anschließend kann der Slave von jedem Master angesprochen werden.

Slave einbinden

– Fügen Sie im Konfigurations-Tool den Slave, also den Feldbusknoten zum Profibus hinzu, in dem Sie die GSD-Datei `hmsb1811.gsd` auswählen. Die Stationsadresse muss die gleiche sein wie bei der Initialisierung in *ADbasic* mit **`P2_Init_Profibus`**.

  Danach könnte das Profibus-Layout wie folgt aussehen:



Slave konfigurieren

– Konfigurieren Sie die Anzahl und Größe der Datenbereiche für eingehende und für ausgehende Daten jeweils einzeln.
Eventuelle Standard-Konfigurationen sind in aller Regel nicht sinnvoll verwendbar.

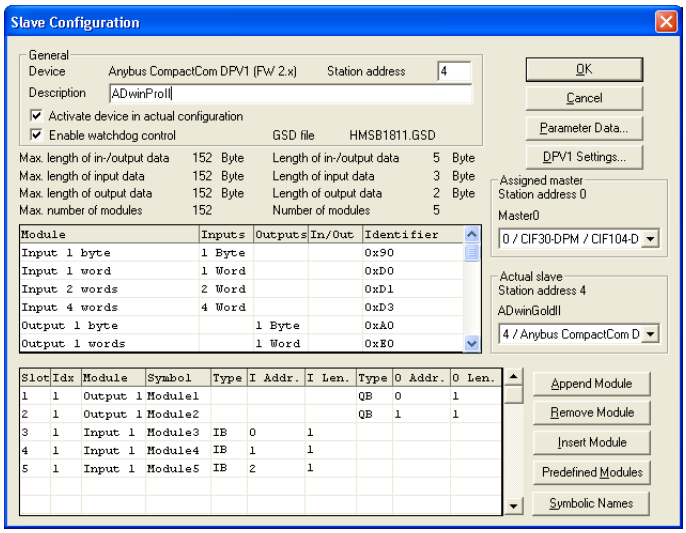
  Beachten Sie dabei folgende Regeln:

  - Die Begriffe „Eingang“ und „Ausgang“ in *ADbasic* (Slave) und im Konfigurations-Tool (Master) sind vertauscht.
  - Konfigurieren Sie (aus Sicht des Masters) zuerst die Ausgänge und dann die Eingänge.
    Wenn also in *ADbasic* Eingänge initialisiert wurden, müssen dafür im Master Ausgänge konfiguriert werden.
  - Anzahl und Größe der Datenbereiche müssen die gleichen sein wie bei der Initialisierung in *ADbasic* mit **`P2_Init_Profibus`**.
  - Verwenden Sie für Eingang und Ausgang im Speicher des Feldbusknotens die gleiche Datenbereichsgröße. Datenbereiche können in Größen von 1, 2, 4 oder 8 Byte angelegt werden (2 byte = 1 word).

  Mit der folgenden Beispielzeile wird der Slave in *ADbasic* mit 2 Eingängen mit 1 Byte und 3 Ausgängen mit 1 Byte initialisiert.

```
PAR_31 = P2_Init_Profibus(2, 2, 1, 3, 1, conf_Arr, DATA_1)
```

  Um den Slave im Konfigurations-Tool richtig einzurichten, müssen nun zuerst 2 Ausgänge und dann 3 Eingänge angelegt werden (jeweils einzeln mit 1 Byte). Die unten stehende Grafik zeigt die Konfiguration beispielhaft.



Modul-Revisionen

Die Unterschiede der Revisionsstände sind nachfolgend dargestellt:

Revision	Ausgabe-datum	Änderungen zur Vorgänger-Version
E1	Juli 2008	Erst-Version

Die Revisionsbezeichnung befindet sich auf der Frontseite.

Unterschiedliche Revisions-Buchstaben bedeuten unterschiedliche Modul-Eigenschaften und sind separat dokumentiert. Der Revisionsbezeichnung angehängt ist eine Zählnummer für interne Zwecke.

Programmieren in *ADbasic*

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Stationsadresse und Datenbereiche initialisieren	`P2_Init_Profibus`
Daten schreiben und lesen	`P2_Run_Profibus`

Die Initialisierung muss mit niedriger Priorität ablaufen, da sie einige Sekunden in Anspruch nimmt; bei hoher Priorität würde der PC nach einer bestimmten Zeit (time-out) die Kommunikation abbrechen. Aus dem gleichen Grund sollte auch das Schreiben und Lesen von Daten mit niedriger Priorität ablaufen.

Spezifikationen

Der Feldbusknoten entspricht dem europäischen Standard EN 50170 Volume 2. Dieser kann von der Profibus-Nutzerorganisation bezogen werden:

Profibus Nutzerorganisation e.V.
Haid-und-Neu-Str.7
76131 Karlsruhe
Tel.: +497219658590
Fax : +497219658589
Bestellnummer: 0.042

Betriebszustände des Feldbusknotens

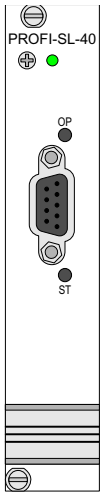
Die nachfolgende Tabelle zeigt die Betriebszustände, die der Feldbusknoten unterstützt und welches Verhalten er in den verschiedenen Zuständen zeigt.

Betriebs-Zustand	Verhalten
Operate	Der Profibusslave nimmt am zyklischen Datenverkehr teil. Eingangsdaten werden von einem Master über den Bus übernommen und Ausgangsdaten werden für den Master zum Abholen bereitgestellt.
Clear	Die Eingänge werden weiterhin aktualisiert und die Ausgänge werden auf Null gesetzt.
Stop	Der Slave nimmt nicht an der Buskommunikation teil.

Abb. 154 – Pro II-Profi-SL Rev. E: Betriebszustände

5.8.10 Pro II-Profi-SL-40 Rev. E

Das Modul Pro II-Profi-SL-40 Rev. E stellt einen Feldbusknoten mit der Funktionalität eines Profibus-Slave zur Verfügung. Alle Einstellungen werden per Software vorgenommen.



Funktionsbeschreibung

Nach dem Einschalten muss der Feldbusknoten initialisiert werden. Mit der Initialisierung werden die Stationsadresse (Slave-Knotenadresse) auf dem Profibus und die Größe der Ein- und Ausgangsbereiche festgelegt.

Es gibt je einen Bereich für eingehende und für ausgehende Daten; die Größe jedes Bereichs kann separat festgelegt werden und zwar auf 1, 2, 4, 8, 16, 32 oder 61 Doppelworte. Die Begriffe „Eingang“ und „Ausgang“ sind aus Sicht des Feldbus-Controllers zu sehen.

TiCo-Prozessor

Das Modul besitzt einen frei programmierbaren *TiCo*-Prozessor, der vollen Zugriff den Feldbusknoten hat, so wie der *ADwin*-Prozessor. Nähere Hinweise zur Anwendung und Programmierung des *TiCo*-Prozessors entnehmen Sie bitte dem Handbuch *TiCoBasic*.

Der *TiCo*-Prozessor (Typ TiCo1) hat eine Taktfrequenz von 50 MHz und ist mit 56 KiB Speicher ausgerüstet: 28 KiB Datenspeicher und 28 KiB Programmspeicher.

Wenn Sie ein *TiCoBasic*-Programm im *TiCo*-Bootloader ablegen, wird das Programm beim Einschalten der Stromversorgung in den *TiCo*-Prozessor geladen und gestartet. Auf diese Weise kann das Modul eigenständig und unabhängig vom CPU-Modul des *ADwin-Pro II*-Systems arbeiten.

Hardware

Die Pinbelegung der 9-poligen D-Sub-Buchse DP-V1 entspricht der DIN E 19245, Teil 3.

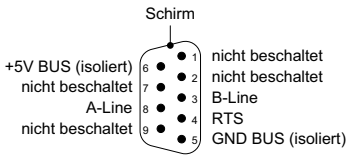


Abb. 155 – Pro II-Profi-SL-40 Rev. E: Pinbelegung

Der Profibus muss am physikalischen Anfang und Ende der Busleitung mit einem Abschlusswiderstand abgeschlossen werden. Falls erforderlich, müssen Sie den Abschlusswiderstand selbst an den entsprechenden Datenleitungen des Feldbusknotens anbringen oder einen entsprechenden Steckverbinder verwenden.

Über und unter der D-Sub-Buchse befinden sich zwei LEDs, die den Betriebszustand des Knotens im Profibus anzeigen, nämlich Betriebsmodus (OP) und Status (ST).

LED	Status	Bedeutung
OP	aus	Nicht online oder keine Stromversorgung.
	grün	Feldbusknoten online, Datenaustausch.
	blinkt grün	Feldbusknoten online, Status Clear.
	blinkt rot, einfach	Fehler: Ein/Ausgangskonfiguration stimmt nicht mit der Masterkonfiguration überein.
	blinkt rot, doppelt	Fehler bei der Profibus-Konfiguration.
ST	aus	Nicht online oder keine Stromversorgung.
	grün	Initialisiert.
	blinkt grün	Initialisiert, Diagnosemeldung liegt an.
	rot	Ausnahmefehler.

Abb. 156 – Pro II-Profi-SL-40 Rev. E: Bedeutung der LED

Profibus projektieren

Sie projektieren den Profibus mit einem – zum Bus-Master passenden – Konfigurations-Tool. Für das folgende Beispiel wurden ein Profibus-Master der Firma Siemens und das zugehörige Programm `SIMATIC` verwendet.

Für andere Konfigurations-Tools gilt die folgende Ablaufbeschreibung entsprechend. Entnehmen Sie die genaue Vorgehensweise bei der Busprojektierung der Dokumentation Ihres Konfigurations-Tools.

GSD-Datei installieren

– Installieren Sie im Programm `SIMATIC` die GSD-Datei `hmsb1815.gsd` (Gerätestammdaten-Datei), Verzeichnis `C:\ADwin\Fieldbus\Profibus`.

  Das Konfigurations-Tool lädt die benötigten Informationen über einen neuen Slave aus der zugehörigen GSD-Datei; der Dateiinhalt ist durch die EN 50170 festgelegt. Anschließend erscheint der Slave in `SIMATIC` im Profil-Baum und kann angesprochen werden.

– Stellen Sie die Stationsadresse des Slaves in `SIMATIC` genauso ein wie bei der Initialisierung in *ADbasic* mit **`P2_Init_Profibus_M40`**.

Slave konfigurieren

– Konfigurieren Sie die Größe der Datenbereiche für eingehende und für ausgehende Daten jeweils einzeln.
  Eventuelle Standard-Konfigurationen sind in aller Regel nicht sinnvoll verwendbar.

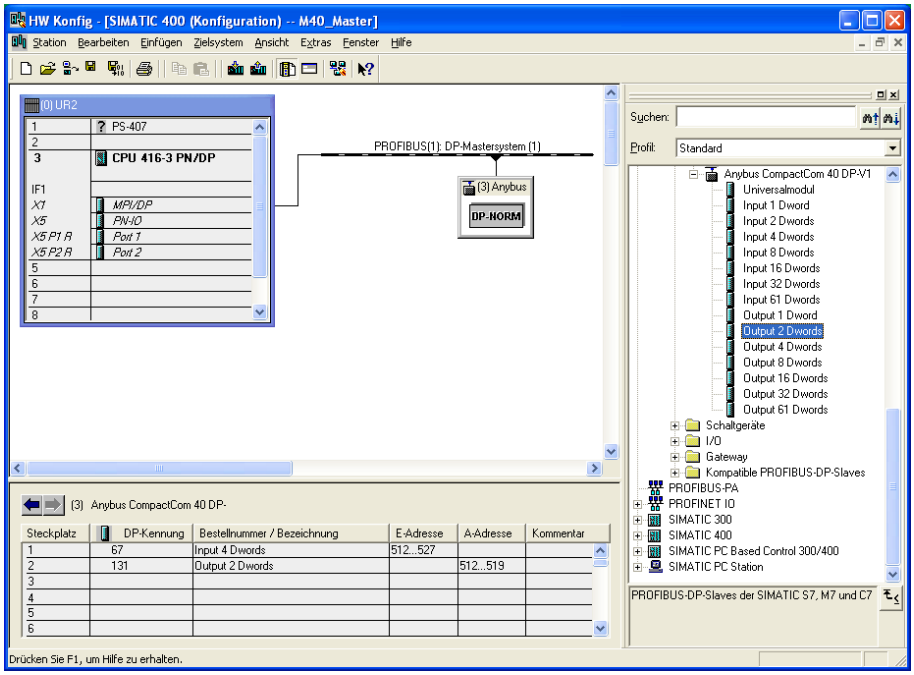
  Beachten Sie dabei folgende Regeln:
  - Die Begriffe „Eingang“ und „Ausgang“ in *ADbasic* (Slave) und im Konfigurations-Tool (Master) sind vertauscht.
  - Konfigurieren Sie (aus Sicht des Konfigurations-Tools) zuerst den Eingang und dann den Ausgang.
    Wenn also in *ADbasic* die Eingangsgröße initialisiert wurde, muss dafür im Master der Ausgang mit der gleichen Größe konfiguriert werden.
  - Die Größe der Datenbereiche muss die gleiche sein wie bei der Initialisierung in *ADbasic* mit **`P2_Init_Profibus_M40`**.
  - Datenbereiche können in Größen von 1, 2, 4, 8, 16, 32, 61 Doppelworten (dword) angelegt werden.

  Beispielhaft wird der Slave in *ADbasic* initialisiert mit Stationsadresse 5, Eingang 2 Doppelworte und Ausgang 4 Doppelworte:

```
Par_31 = P2_Init_Profibus_M40(module, 5, 2, 4, conf_Arr)
```

  Um den Slave im Konfigurations-Tool richtig einzurichten, müssen nun zuerst der Eingang mit 4 Doppelworten und dann der Ausgang mit 2

Doppelworten angelegt werden. Die unten stehende Grafik zeigt die Beispielkonfiguration.



Modul-Revisionen

Die Unterschiede der Revisionsstände sind nachfolgend dargestellt:

Revision	Ausgabe-datum	Änderungen zur Vorgänger-Version
E1	Jan 2020	Erst-Version

Die Revisionsbezeichnung befindet sich auf der Frontseite.

Unterschiedliche Revisions-Buchstaben bedeuten unterschiedliche Modul-Eigenschaften und sind separat dokumentiert. Der Revisionsbezeichnung angehängt ist eine Zählnummer für interne Zwecke.

Programmieren in *ADbasic*

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Stationsadresse und Datenbereiche initialisieren	`P2_Init_Profibus_M40`
Daten schreiben und lesen	`P2_Run_Profibus_M40`

Die Befehle müssen mit niedriger Priorität ablaufen, da sie einige Sekunden in Anspruch nehmen; bei hoher Priorität würde der PC nach einer bestimmten Zeit (time-out) die Kommunikation abbrechen.

Programmierung in TiCoBasic

Das Modul kann mit *TiCoBasic*-Befehlen programmiert werden. Die Befehle sind in der Online-Hilfe *TiCoBasic* erläutert.

Die Include-Datei `Fieldbus_TiCo.inc` enthält folgende Befehle:

Bereich	Befehle
Stationsadresse und Datenbereiche initialisieren	`Init_Profibus_M40`
Daten schreiben und lesen	`Run_Profibus_M40`

Programmierung TiCo-Zugriff

Für den Zugriff auf den *TiCo*-Prozessor von der ADwin CPU sind die folgenden *ADbasic*-Befehle in der Datei `ADwinPro_All.inc` definiert. Die Befehle sind im Handbuch *TiCoBasic* und in der Online-Hilfe *ADbasic* erläutert.

Bereich	Befehle
Datenaustausch mit dem *TiCo*-Prozessor über globale Variablen	**P2_TDrv_Init** **P2_GetData_Long**, **P2_Get_Par**, **P2_Get_Par_Block** **P2_SetData_Long**, **P2_Set_Par**, **P2_Set_Par_Block** **P2_Get_TiCo_RingBuffer**, **P2_Set_TiCo_RingBuffer** **P2_RingBuffer_Empty** **P2_RingBuffer_Full**
TiCo-Prozessor steuern	**P2_TiCo_Reset**, **P2_TiCo_Start**, **P2_TiCo_Stop** **P2_Get_TiCo_Bootloader_Status** **P2_Get_TiCo_Status**, **P2_Workload**
TiCo-Prozesse steuern	**P2_Process_Status** **P2_TiCo_Get_Processdelay** **P2_TiCo_Set_Processdelay** **P2_TiCo_Start_Process** **P2_Tico_Stop_Process**
TiCo-Programme übertragen	**P2_TiCo_Flash**, **P2_TiCo_Load**

Spezifikationen

Der Feldbusknoten entspricht dem europäischen Standard EN 50170 Volume 2. Dieser kann von der Profibus-Nutzerorganisation bezogen werden:

Profibus Nutzerorganisation e.V.
Haid-und-Neu-Str.7
76131 Karlsruhe
Tel.: +497219658590
Fax : +497219658589
Bestellnummer: 0.042

Betriebszustände des Feldbusknotens

Die nachfolgende Tabelle zeigt die Betriebszustände, die der Feldbusknoten unterstützt und welches Verhalten er in den verschiedenen Zuständen zeigt.

Betriebs-Zustand	Verhalten
Operate	Der Profibusslave nimmt am zyklischen Datenverkehr teil. Eingangsdaten werden von einem Master über den Bus übernommen und Ausgangsdaten werden für den Master zum Abholen bereitgestellt.
Clear	Die Eingänge werden weiterhin aktualisiert und die Ausgänge werden auf Null gesetzt.
Stop	Der Slave nimmt nicht an der Buskommunikation teil.

Abb. 157 – Pro II-Profi-SL-40 Rev. E: Betriebszustände

5.8.11 Pro II-PROFI-IRT-40 Rev. E

Das Modul Pro II-PROFI-IRT-40 Rev. E stellt einen Feldbusknoten mit der Funktionalität eines Profinet-IRT-Slave zur Verfügung. Alle Einstellungen werden per Software vorgenommen.

Es gibt das Modul mit verschiedenen Anschlüssen:

– *Pro II-PROFI-IRT-CU-40* Rev. E: Schnittstelle mit Kupferleiter, 2 Buchsen RJ-45, handelsübliche Steckverbinder.

– *Pro II-PROFI-IRT-FO-40* Rev. E: Schnittstelle mit Lichtwellenleiter, 2 Duplex-Buchsen SC-RJ (Lichtwellenleiter).

Funktionsbeschreibung

Nach dem Einschalten muss der Feldbusknoten initialisiert werden. Mit der Initialisierung wird die Größe der Ein- und Ausgangsbereiche festgelegt.

Es gibt je einen Bereich für eingehende und für ausgehende Daten; jeder Bereich hat eine maximale Größe von 1280 Byte. Die Begriffe „Eingang“ und „Ausgang“ sind aus Sicht des Feldbus-Masters zu sehen.

Bei der Initialisierung legen Sie für beide Bereiche separat fest, wieviele Datenbereiche in jedem Bereich vorhanden sind und wie groß die Datenbereiche sind. Im Betrieb kann jedoch nur eine Bereichsgröße genutzt werden.

TiCo-Prozessor

Das Modul besitzt einen frei programmierbaren *TiCo*-Prozessor, der vollen Zugriff den Feldbusknoten hat, so wie der *ADwin*-Prozessor. Nähere Hinweise zur Anwendung und Programmierung des *TiCo*-Prozessors entnehmen Sie bitte dem Handbuch *TiCoBasic*.

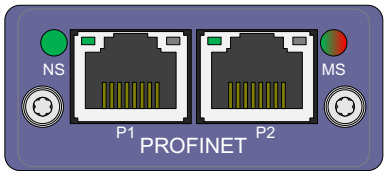
Der *TiCo*-Prozessor (Typ TiCo1) hat eine Taktfrequenz von 50 MHz und ist mit 56 KiB Speicher ausgerüstet: 28 KiB Datenspeicher und 28 KiB Programmspeicher.

Wenn Sie ein *TiCoBasic*-Programm im *TiCo*-Bootloader ablegen, wird das Programm beim Einschalten der Stromversorgung in den *TiCo*-Prozessor geladen und gestartet. Auf diese Weise kann das Modul eigenständig und unabhängig vom CPU-Modul des *ADwin-Pro II*-Systems arbeiten.

Hardware

An die Buchsen werden handelsübliche Stecker angeschlossen:

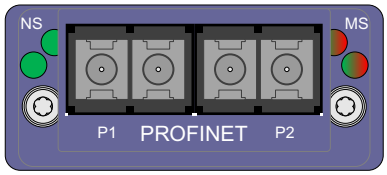
– Ethernet-Stecker RJ-45 (IRT-Cu)



Neben den beiden Buchsen befinden sich zwei LEDs, die den Betriebszustand des Knotens im Profinet anzeigen: Netzwerk-Status (NS) und Modul-Status (MS).

In jeder Buchse befindet sich jeweils eine LINK-LED.

– Glasfaser-Duplexstecker SC-RJ (IRT-FO)



Neben den Buchsen befinden sich vier LEDs. Die halb verdeckten LED zeigen den Betriebszustand des Knotens im Profinet anzeigen: Netzwerk-Status (NS, links) und Modul-Status (MS, rechts).

Die weiter außen liegenden LED für jede Buchse den LINK-Status an.

LED	Status	Bedeutung
NS	Aus	Offline: Nicht online oder keine Stromversorgung.
	Grün	Online (RUN): Feldbusknoten online, IO-Master arbeitet.
	Grün blinkt 1-fach	Online (STOP): Feldbusknoten online, aber IO-Master ruht, Datenfehler oder IRT-Synchronisierung nicht abgeschlossen.
	Grün blinkend	Blink-Modus: wird von Prüfwerkzeugen zur Identifizierung des Knotens im Netzwerk genutzt.
	Rot (mit MS rot)	Ausnahmefehler: Schwerer interner Fehler; die MS LED ist ebenfalls rot.
	Rot blinkt 1-fach	Stationsname ist nicht gesetzt.
	Rot blinkt 2-fach	IP-Adresse ist nicht gesetzt.
	Rot blinkt 3-fach	Konfigurationsfehler: Erwartete unterscheidet sich von der tatsächlichen Identifizierung.
MS	Aus	Nicht initialisiert: Keine Stromversorgung oder Modul im Modus SETUP / NW_INIT.
	Grün	Normaler Betrieb.
	Grün blinkt 1-fach	Diagnosemeldung liegt an.
	Rot	Ausnahmefehler, Gerät im Status EXCEPTION.
	Rot (mit NS rot)	Ausnahmefehler: Schwerer interner Fehler; die MS LED ist ebenfalls rot.
	Rot / Grün wechselnd	Firmware-Update. Schalten Sie das Gerät nicht ab. Abschalten während dieser Phase könnte dauerhafte Schäden verursachen.
LINK	Aus	Kein Verbindung vorhanden.
	Grün	Ethernet-Verbindung vorhanden, keine Kommunikation.
	Grün flackernd	Ethernet-Verbindung vorhanden, Kommunkation findet statt.

Abb. 158 – Profinet: Bedeutung der LED

Profinet projektieren

Sie projektieren den Profinet-Bus mit einem – zum Bus-Master passenden – Konfigurations-Tool. Für das folgende Beispiel wurden ein Profinet-Master der Firma Siemens und das zugehörige Programm `SIMATIC-Manager` verwendet.

Für andere Konfigurations-Tools gilt die folgende Ablaufbeschreibung entsprechend. Entnehmen Sie die genaue Vorgehensweise bei der Busprojektierung der Dokumentation Ihres Konfigurations-Tools.

GSD-Datei installieren

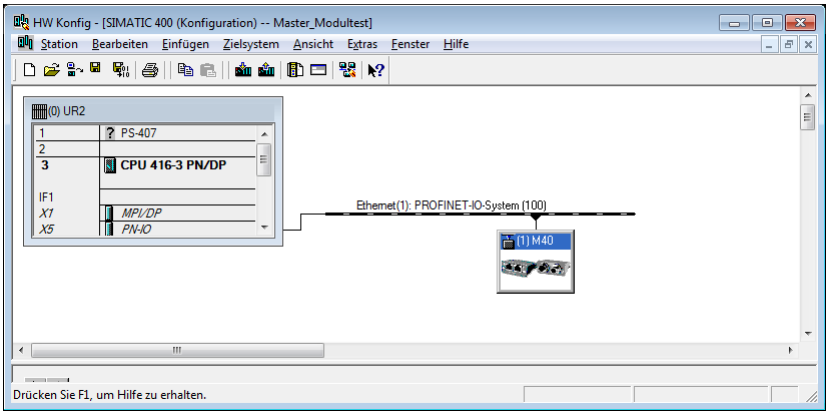
– Installieren Sie im Programm `SIMATIC-Manager` die GSD-Datei `GSDML-V2.33-HMS-ABCC40-PIR-ADwin-20180620.xml` (Gerätestammdaten-Datei), Verzeichnis `C:\ADwin\Fieldbus\Profinet`.

  Das Konfigurations-Tool lädt die benötigten Informationen über einen neuen Slave aus der zugehörigen GSD-Datei. Anschließend kann der Slave von jedem Master angesprochen werden.

Slave einbinden

– Fügen Sie im Konfigurations-Tool den Slave, also den Feldbusknoten zum Profinet hinzu.

  Danach könnte das Profinet-Layout wie folgt aussehen:



Slave konfigurieren

– Konfigurieren Sie die Größe der Datenbereiche für eingehende und für ausgehende Daten jeweils einzeln.
  Eventuelle Standard-Konfigurationen sind in aller Regel nicht sinnvoll verwendbar.

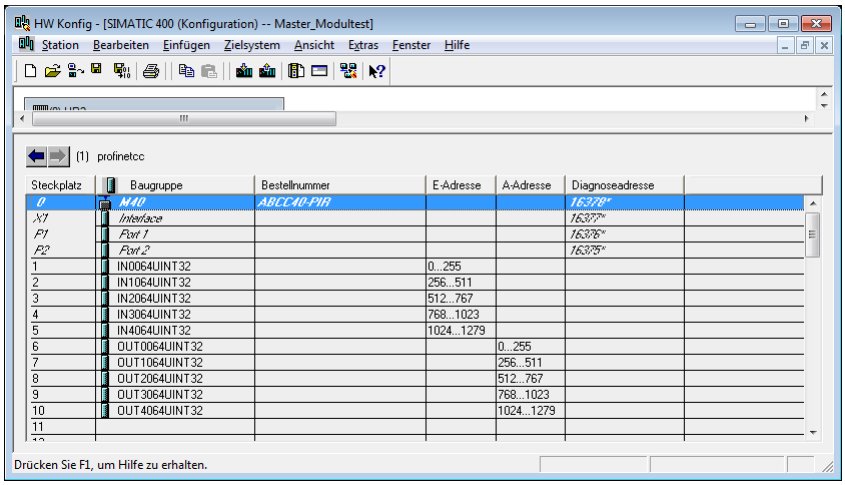
  Beachten Sie dabei folgende Regeln:
  - Die Begriffe „Eingang“ und „Ausgang“ in *ADbasic* (Slave) und im Konfigurations-Tool (Master) sind vertauscht.
    Wenn also in *ADbasic* Eingänge initialisiert wurden, müssen dafür im Master Ausgänge konfiguriert werden.
  - Die Größe der Datenbereiche müssen so eingestellt sein wie bei der Initialisierung in *ADbasic* mit **P2_Init_ProfinetIO**.
  - Die Datenbereichsgrößen für Eingang und Ausgang können voneinander unabhängig initialisiert werden. Datenbereiche können angelegt werden in einer der folgenden Größen (1 Doppelwort = 4 Byte):
    1, 2, 4, … 64 Doppelworte; 64, 128, … 320 Doppelworte
    Die 64 Doppelwort-Blöcke sind nummeriert und dürfen nur in aufsteigender Reihenfolge IN0…IN5 / OUT0…OUT5 angelegt werden.

Beispiel-Konfiguration

  Mit der folgenden Zeile werden in *ADbasic* sowohl der Eingangsbereich als auch der Ausgangsbereich mit 320 Doppelworten (=1280 Byte) initialisiert:

```
P2_Init_ProfinetIO(module, 320, 320, work_arr)
```

  Um den Slave im Konfigurations-Tool richtig einzurichten, müssen im Eingangs- und im Ausgangsbereich jeweils fünf Blöcke zu 256 Byte (=64 Doppelworte) angelegt werden. Beachten Sie die Reihenfolge der 64 Doppelwort-Blöcke.



Programmieren in *ADbasic*

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Reset, Datenbereiche initialisieren	`P2_Init_ProfinetIO`
Daten schreiben und lesen, Nachrichtenverwaltung	`P2_Run_ProfinetIO`

Programmierung in TiCoBasic

Das Modul kann mit *TiCoBasic*-Befehlen programmiert werden. Die Befehle sind in der Online-Hilfe *TiCoBasic* erläutert.

Die Include-Datei `Fieldbus_TiCo.inc` enthält folgende Befehle:

Bereich	Befehle
Reset, Datenbereiche initialisieren	`Init_ProfinetIO`
Daten schreiben und lesen, Nachrichtenverwaltung	`Run_ProfinetIO`

Programmierung TiCo-Zugriff

Für den Zugriff auf den *TiCo*-Prozessor von der ADwin CPU sind die folgenden *ADbasic*-Befehle in der Datei `ADwinPro_All.inc` definiert. Die Befehle sind im Handbuch *TiCoBasic* und in der Online-Hilfe *ADbasic* erläutert.

Bereich	Befehle

Datenaustausch mit dem *TiCo*-Prozessor über globale Variablen	`P2_TDrv_Init`
	`P2_GetData_Long`, `P2_Get_Par`, `P2_Get_Par_Block`
	`P2_SetData_Long`, `P2_Set_Par`, `P2_Set_Par_Block`
	`P2_Get_TiCo_RingBuffer`, `P2_Set_TiCo_RingBuffer`
	`P2_RingBuffer_Empty`
	`P2_RingBuffer_Full`

TiCo-Prozessor steuern	`P2_TiCo_Reset`, `P2_TiCo_Start`, `P2_TiCo_Stop`
	`P2_Get_TiCo_Bootloader_Status`
	`P2_Get_TiCo_Status`, `P2_Workload`

Bereich	Befehle

TiCo-Prozesse steuern	`P2_Process_Status`
	`P2_TiCo_Get_Processdelay`
	`P2_TiCo_Set_Processdelay`
	`P2_TiCo_Start_Process`
	`P2_Tico_Stop_Process`

TiCo-Programme übertragen	`P2_TiCo_Flash`, `P2_TiCo_Load`

Spezifikationen

Der Feldbusknoten entspricht dem Standard IEC 61158 (Feldbus). Dieser kann von der Profibus-Nutzerorganisation bezogen werden:

Profibus Nutzerorganisation e.V.
Haid-und-Neu-Str.7
76131 Karlsruhe
Tel.: +497219658590
Fax : +497219658589
www.profibus.com

Betriebszustände des Feldbusknotens

Die nachfolgende Tabelle zeigt die Betriebszustände, die der Feldbusknoten unterstützt und welches Verhalten er in den verschiedenen Zuständen zeigt.

Betriebs-Zustand	Verhalten
Setup	Initialisierung der Schnittstelle.
Wait	Slave wartet auf Busstart durch den Master.
Active	Der Profinet-Slave nimmt am zyklischen Datenverkehr teil.
Error	Fehler. Slave nimmt nicht am Busverkehr teil.
Exception	Schwerer interner Fehler. Slave nimmt nicht am Busverkehr teil.

Abb. 159 – Profinet: Betriebszustände

5.8.12 Pro II-MIL-1553 Rev. E

Das Modul Pro II-MIL-1553 Rev. E stellt eine Multifunktions-Schnittstelle für den Feldbus MIL-STD-1553 an 2 Anschlüssen Bus A und Bus B zur Verfügung. Derzeit kann die MIL-Schnittstelle als Busmonitor im Modus SMT 16 Bit (simple monitoring terminal) konfiguriert werden.
Andere Funktionen als Buscontroller, Gerät (remote terminal) oder Monitor können durch Setzen der entsprechenden Register eingestellt werden.

Schnittstelle MIL-STD-1553

Die MIL-Schnittstelle des Moduls entspricht der Spezifikation MIL-STD-1553B. Das Modul basiert auf dem MIL Multi-Terminal HI-6130.

Sie programmieren den Modus SMT der MIL-Schnittstelle mit *ADbasic*-Befehlen. Für andere Modi setzen Sie die entsprechenden Register entsprechend der separaten Dokumentation "HI-6130 / MIL-STD-1553 / BC/MT/RT Multi-Terminal Device" von Holt Integrated Circuits Inc.

MIL-Bus

MIL-STD-1553 ist ein serieller Bus, der für die Datenverarbeitung in Flugzeugen weit verbreitet ist, sowohl militärisch wie zivil. Die Bitübertragungsrate ist 1,0 Megabit pro Sekunde.

Zum Buskonzept gehört ein einzelner Bus-Controller (BC) mit mehreren angesteuerten Geräten (remote terminal, RT) als Slaves. Zusätzlich können ein oder mehrere Busmonitore angeschlossen sein, die nicht am Datenverkehr teilnehmen, sondern nur Daten erfassen oder speichern.

Alle Nachrichten auf dem Datenbus sind sowohl dem BC als auch den angeschlossenen Geräten zugänglich. Nachrichten bestehen aus einem oder mehreren 16 Bit-Worten (Befehlswort, Datenwort oder Statuswort). Die 16 Bits eines Worts werden im Manchester-Code übertragen.

Der BC kontrolliert allen Datenverkehr über den Bus. Dabei sendet der BC Befehle zum Senden oder Empfangen an die RTs.

Vor jedem Datenpaket sendet der BC einen Header mit der Kennung (Identifier) des nächsten Datenpakets. Es reagiert nur derjenige Busknoten (das kann auch der BC selbst sein), der die Nachrichten mit der angegebenen Kennung verwaltet. Danach sendet der Knoten ein Datenpaket über den MIL-Bus oder er empfängt ein Datenpaket von dort.

In den Busteilnehmern gibt es Untersysteme (z.B. RT1), die über eine Unteradresse angesprochen werden.

Modus SMT

Wenn das Modul als Busmonitor im Modus SMT arbeitet, kann das Modul alle Befehle, Geräteantworten und Nachrichtendaten speichern, die über die beiden MIL-Bus-Anschlüsse A und B gehen.

Der SMT-Busmonitor kann 16 Bit-Zeitmarken mit verschiedenen Taktoptionen verwenden.

Nach dem Einschalten speichert der SMT Monitor jede Nachricht auf den beiden MIL-Bus-Anschlüssen. Sie können Filter einstellen, um bestimmte Nachrichten auszuwählen. Die Filter basieren auf der Geräteadresse und -Unteradresse der Befehle sowie auf dem Statusbit für Senden / Empfangen.

Der Monitor speichert eine Nachricht in zwei Zwischenspeichern:

– Zwischenspeicher für Befehle: enthält für jede MIL-Nachricht 4 Worte zu je 16 Bit. Die Worte sind:
  - Befehlswort des BC, mit dem die Nachricht beginnt.
    Bei einer RT-RT-Nachricht, enthält das Wort den Befehl zum Empfangen; der zweite Befehl zum Senden ist das erste Wort im Zwischenspeicher für Daten.

  - Zeiger auf die Worte im Zwischenspeicher für Daten; der Zeiger wird nicht benötigt, weil der *ADbasic*-Befehl diese Daten bereits zurückgibt.
  - Zeitmarke (16 Bit) der Nachricht.
  - Blockstatus des SMT: zusätzliche Informationen über Nachrichtenstatus, den verwendeten Bus A/B und eventuell aufgetretene Fehler.

– Zwischenspeicher für Daten: Datenworte der Nachricht.
  Die Anzahl der Worte für eine Nachricht hängt vom MIL-Nachrichtentyp ab und reicht von 0 (Broadcast-Befehl ohne Daten) bis zu 35 Worten (RT-RT-Befehl mit 32 Datenworten).

Die zuletzt empfangene Nachricht kann mit **`P2_MIL_SMT_Message_Read`** gelesen werden. Der Befehl gibt die Daten der Nachricht in 2 Feldern zurück: im ersten Feld die 4 Worte des Zwischenspeichers für Befehle, im zweiten Feld die Datenworte aus dem Zwischenspeicher für Daten.

Sie können nur die Daten der zuletzt empfangenen Nachricht lesen. Sobald eine neue MIL-Nachricht vollständig empfangen und gespeichert ist, sind die Informationen früherer Nachrichten verloren.

Hardware

Auf der Frontseite befinden sich 2 konzentrische Triax-Bajonett-Anschlüsse, die als Bus A und Bus B bezeichnet sind. Für jeden Anschluss gibt es 2 programmierbare LEDs.

Auf der Platine sind mehrere DIP-Schalter angeordnet (S1…S8, S13), die nur verwendet werden, wenn die Schnittstelle als BC oder RT genutzt wird. Im Modus SMT-Monitor sollen alle DIP-Schalter in der Position OFF stehen.

Ungeeignete Einstellungen der DIP-Schalter können das Modul schädigen. Wenn Sie die Einstellungen ändern müssen, wenden Sie sich bitte an den Support von Jäger Messtechnik; Sie finden die Adresse auf der Innenseite des Deckblatts.

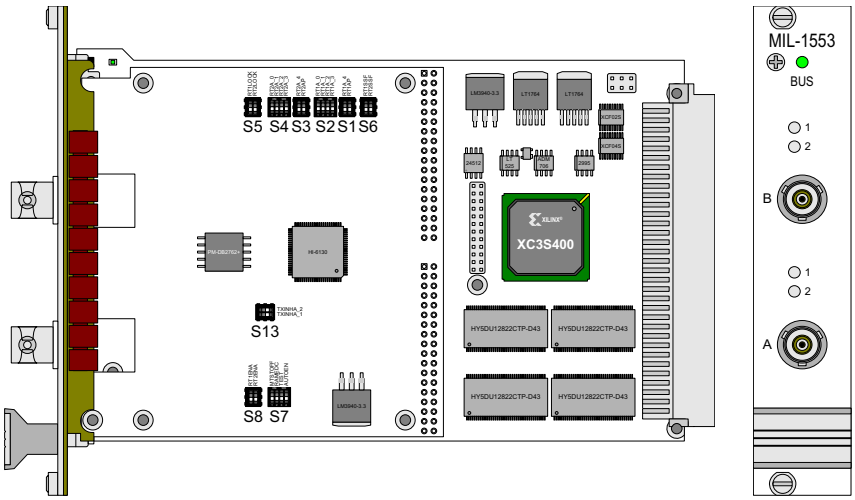


Abb. 160 – Pro II-MIL-1553 Rev. E: Platine und Frontplatte

DIP-Schalter	DIP	Stellung für Modus SMT	DIP-Schalter	DIP	Stellung für Modus SMT
S5	RT1LOCK	OFF	S6	RT1SSF	OFF
S5	RT2LOCK	OFF	S6	RT2SSF	OFF
S2	RT1A_0	OFF	S4	RT2A_0	OFF

DIP-Schalter	DIP	Stellung für Modus SMT
S2	RT1A_1	OFF
S2	RT1A_2	OFF
S2	RT1A_3	OFF
S1	RT1A_4	OFF
S1	RT1AP	OFF
S8	RT1ENA	OFF
S8	RT2ENA	OFF
S7	MTSTOFF	OFF
S7	RAMEDC	OFF
S7	TEST	OFF
S7	AUTOEN	OFF

DIP-Schalter	DIP	Stellung für Modus SMT
S4	RT2A_1	OFF
S4	RT2A_2	OFF
S4	RT2A_3	OFF
S3	RT2A_4	OFF
S3	RT2AP	OFF
S13	TXINHA_1	OFF
S13	TXINHA_2	OFF

Sie finden eine Beschreibung der DIP-Schalter in der separaten Dokumentation "HI-6130 / MIL-STD-1553 / BC/MT/RT Multi-Terminal Device" by Holt Integrated Circuits Inc.

Modul-Revisionen

Revision	Ausgabe	Änderungen
E1	Dec. 2012	Erstversion

Programmierung

Programmieren mit ADbasic

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Area	Instructions
Rücksetzen und initialisieren der MIL-Schnittstelle	`P2_MIL_Reset` `P2_MIL_SMT_Init`
SMT: Busnachricht lesen	`P2_MIL_SMT_Message_Read`
SMT-Filter setzen	`P2_MIL_Set_All_Filters` `P2_MIL_Set_Filter`

LEDs einstellen	`P2_Check_LED`
	`P2_Set_LED`
	`P2_MIL_Set_LED`

Register der MIL-Schnittstelle setzen und lesen	`P2_MIL_Set_Register` `P2_MIL_Get_Register`

5.8.13 Pro II-ARINC-429 Rev. E

Das Modul Pro II-ARINC-429 Rev. E stellt eine Schnittstelle für den seriellen Bus ARINC 429 mit einem Sender (Transmitter) und 2 Empfängern (Receiver) zur Verfügung. Sender und Empfänger können gleichzeitig betrieben werden.

Schnittstelle ARINC 429

Die Schnittstelle des Moduls entspricht der Spezifikation für ARINC-429. Das Modul basiert auf der Schnittstelle HI-3582A von Holt Integrated Circuits Inc.

Sie programmieren die ARINC-Schnittstelle mit *ADbasic*- oder *TiCoBasic*-Befehlen.

ARINC 429 ist ein serieller 2-Draht-Bus und ist der am meisten verwendete Datenbus-Standard in der Luftfahrt. Der Bus arbeitet mit einem Sender und bis zu 20 Empfängern. Die Bitübertragungfsrate beträgt 100 kBit/s (high speed) oder 12,5 kBit/s (low speed).

ARINC 429 verwendet ein selbsttaktendes und selbstsynchronisierendes Datenbus-Protokoll; die Leitungen Tx (senden) und Rx (empfangen) liegen auf unterschiedlichen Anschlüssen. Die physikalischen Verbindungskabel sind verdrillte Leitungen und werden mit A und B bezeichnet.

Daten könnne nur in eine Richtung übertragen werden – Simplexbetrieb – so dass bidirektionale Übertragung 2 Kanäle oder Busse erfordert. Die Busteilnehmer, auch als LRU (line replaceable units) bezeichnet, werden meist als Stern- oder Bustopologie angeordnet. Jeder Teilnehmer kann mehrere Sender und Empfänger enthalten, die auf verschiedenen Bussen kommunizieren.

Datenworte auf dem Bus haben eine Länge von 32 Bit und die meisten Nachrichten bestehen aus einem einzigen Datenwort. Der Sender überträgt dauernd entweder 32 Bit-Datenworte oder den NULL-Status, wöhrend die Empfänger die Busnachrichten mithören.

Ein Datenwort für ARINC 429 besteht aus 32 Bit in 5 Bereichen:

– Label, 8 bits
– Source/destination identifier (SDI), 2 bits
– Daten, 19 bits
– Sign/status matrix (SSM), 2 bits
– Parity-Bit, 1 bit

Moduleigenschaften

Das Modul Pro II-ARINC-429 Rev. E stellt einen Sender und 2 Empfänger zur Verfügung, die jeweils einen eigenen Fifo verwenden. Jeder Fifo kann bis zu 32 Datenworte aufnehmen.

Eigenschaften des Moduls Pro II-ARINC-429 Rev. E:

– Paritätsprüfung beim Senden

  Die Paritätsprüfung beim Senden kann abgeschaltet werden. Wenn sie eingeschaltet ist, kann auf gerade oder auf ungerade Parität geprüft werden.

  Beachten Sie, dass die Empfänger des Moduls immer eine Prüfung auf ungerade Parität durchführen. Die Paritätsprüfung beim Empfangen kann nicht abgeschaltet werden.

– Label-Erkennung

  Jeder der beiden Empfänger kann mit bis zu 16 Labels programmiert werden. Wenn die Label-Erkennung eingeschaltet ist, akzeptiert der

Empfänger nur solche ARINC-Nachrichten, die zu einem der programmierten Label passen. Nur akzeptierte Nachrichten werden in den Empfangs-Fifo kopiert.

Die Label-Erkennung kann für jeden Empfänger separat eingerichtet werden.

– SDI-Erkennung

Jeder der beiden Empfänger kann mit einem SDI-Wert programmiert werden. Wenn die SDI-Erkennung eingeschaltet ist, akzeptiert der Empfänger nur solche ARINC-Nachrichten, die zum programmierten SDI-Wert passen. Nur akzeptierte Nachrichten werden in den Empfangs-Fifo kopiert.

Die Label-Erkennung kann für jeden Empfänger separat eingerichtet werden.

SDI-Erkennung und Label-Erkennung können unabhängig voneinander eingeschaltet werden. Wenn beide aktiv sind, muss ein eingehender Wert beide Bedingungen erfüllen, damit er akzeptiert wird.

Hardware

Auf der Frontseite steht eine 25-polige D-Sub-Buchse zur Verfügung.

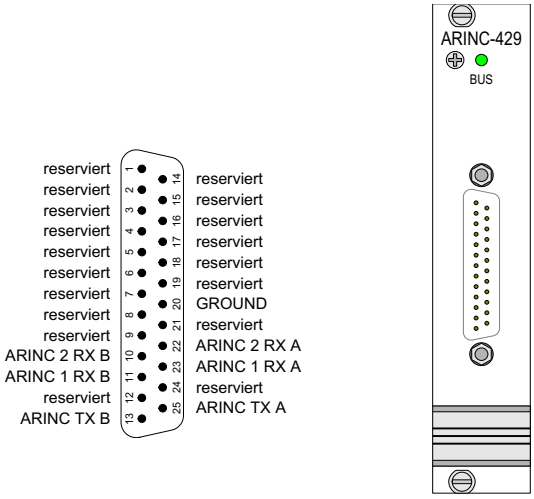


Abb. 161 – Pro II-ARINC-429 Rev. E: Pin-Belegung und Frontseite

Funktion	Schnittstelle für ARINC 429 mit 1 Sender und 2 Empfängern
Bitübertragungsrate	Wählbar: 100 kBit/s (high speed) oder 12,5 kBit/s (low speed)
Schnittstelle basiert auf	HI-3582A von Holt Integrated Circuits Inc.
TiCo	Prozessortyp: TiCo1 Taktrate: 50 MHz Speichergröße: 28 kiB PM intern, 28 kiB DM intern.
Steckerverbindung	25-polige D-Sub-Buchse

Abb. 162 – Pro II-ARINC-429 Rev. E: Spezifikation

Modul-Revisionen

Revision	Ausgabe	Änderungen
E1	März 2013	Erstversion

Programmierung

Programmieren mit ADbasic

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle

Rücksetzen und konfigurieren der Schnittstelle ARINC-429	`P2_ARINC_Reset`
	`P2_ARINC_Config_Transmit`
	`P2_ARINC_Config_Receive`
	`P2_ARINC_Set_Labels`

Eine Nachricht senden	`P2_ARINC_Transmit_Enable`
	`P2_ARINC_Transmit_Fifo_Full`
	`P2_ARINC_Transmit_Fifo_Empty`
	`P2_ARINC_Write_Transmit_Fifo`
	`ARINC_Create_Value32`

Eine Nachricht empfangen	`P2_ARINC_Receive_Fifo_Empty`
	`P2_ARINC_Read_Receive_Fifo`
	`ARINC_Split_Value32`

LEDs einstellen	`P2_Check_LED` `P2_Set_LED`

Programmieren mit TiCoBasic

Das Modul kann mit *TiCoBasic*-Befehlen programmiert werden. Die Befehle sind in der Online-Hilfe *TiCoBasic* erläutert.

Die Include-Datei `ARINC_TiCo.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle

Rücksetzen und konfigurieren der Schnittstelle ARINC-429	`ARINC_Reset`
	`ARINC_Config_Transmit`
	`ARINC_Config_Receive`
	`ARINC_Set_Labels`

Eine Nachricht senden	`ARINC_Transmit_Enable`
	`ARINC_Transmit_Fifo_Full`
	`ARINC_Transmit_Fifo_Empty`
	`ARINC_Write_Transmit_Fifo`
	`ARINC_Create_Value32`

Eine Nachricht empfangen	`ARINC_Receive_Fifo_Empty`
	`ARINC_Read_Receive_Fifo`
	`ARINC_Split_Value32`

LEDs einstellen	`Check_LED` `Set_LED`

Programmierung TiCo-Zugriff

Für den Zugriff auf den *TiCo*-Prozessor von der ADwin CPU sind die folgenden *ADbasic*-Befehle in der Datei `ADwinPro_All.inc` definiert. Die Befehle sind im Handbuch *TiCoBasic* und in der Online-Hilfe *ADbasic* erläutert.

Bereich	Befehle
Datenaustausch mit dem *TiCo*-Prozessor über globale Variablen	P2_TDrv_Init P2_GetData_Long, P2_Get_Par, P2_Get_Par_Block P2_SetData_Long, P2_Set_Par, P2_Set_Par_Block P2_Get_TiCo_RingBuffer, P2_Set_TiCo_RingBuffer P2_RingBuffer_Empty P2_RingBuffer_Full
TiCo-Prozessor steuern	P2_TiCo_Reset, P2_TiCo_Start, P2_TiCo_Stop P2_Get_TiCo_Bootloader_Status P2_Get_TiCo_Status, P2_Workload
TiCo-Prozesse steuern	P2_Process_Status P2_TiCo_Get_Processdelay P2_TiCo_Set_Processdelay P2_TiCo_Start_Process P2_Tico_Stop_Process
TiCo-Programme übertragen	P2_TiCo_Flash, P2_TiCo_Load

5.8.14 Pro II-FlexRay-2 Rev. E

Das Modul Pro II-FlexRay-2 Rev. E besitzt 2 FlexRay-Schnittstellen, wobei jede Schnittstelle einen kompletten FlexRay-Bus mit 2 Kanälen repräsentiert.

Das Modul ist so konfigurierbar, dass es ohne weitere Busteilnehmer einen FlexRay-Bus starten kann. Dazu müssen beide FlexRay-Schnittstellen als Coldstarter-Knoten arbeiten und die Schnittstellen per DIP-Schalter zu einem FlexRay-Bus zusammengeschlossen werden.

Ebenfalls per DIP-Schalter kann für jede Schnittstelle kanalweise die Bus-Terminierung eingestellt werden.

Die Beschreibung des FlexRay-Moduls ist in folgende Abschnitte gegliedert:

- FlexRay-Controller
- Hardware-Aufbau
- Modul-Revisionen
- Programmierung

FlexRay-Controller

Das Modul Pro II-FlexRay-2 ist mit zwei FlexRay-Controllern MFR4310 von FreeScale® (NXP) bestückt und arbeitet nach der „FlexRay Communications System Protocol Specification V2.1“. Sie programmieren die Schnittstelle mit *ADbasic*-Befehlen, die direkt auf die Register des Controllers zugreifen.

Zur Konfiguration und Statusanzeige der FlexRay-Controller dienen die in ihnen enthaltenen Register. Hier werden alle FlexRay-Parameter wie z. B. Busgeschwindigkeit, Bustiming etc. eingestellt.

Weitere Informationen finden Sie auf der Webseite https://www.nxp.com in folgenden Dokumentationen von FreeScale®:

- Engineering Bulletin EB683: MFR4310 and MFR4310 differences
- Data sheet MFR4300: MFR4300 FlexRay Communication Controller

Hardware-Aufbau

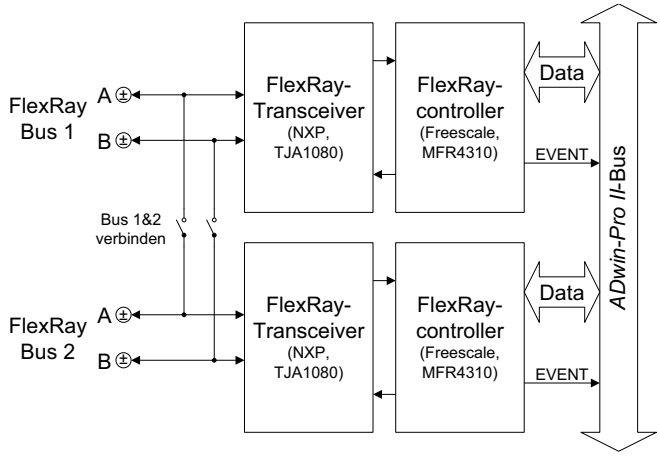


Abb. 163 – Pro II-FlexRay-2 Rev. E: Blockschaltbild

Die Anschlüsse der FlexRay-Schnittstelle stehen auf einem 9-poligen D-Sub-Verbinder zur Verfügung; die Pin-Belegung ist unten dargestellt.

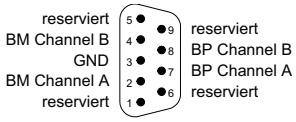


Abb. 164 – Pro II-FlexRay-2 Rev. E: Pinbelegung

Wenn eine Schnittstelle des FlexRay-Moduls das physikalische Ende eines FlexRay-Busses bildet, muss der Bus an an dieser Schnittstelle mit dem entsprechenden DIP-Schalter terminiert werden (siehe Abb. 165); die Kanäle A und B können separat terminiert werden. Da die Kanäle differentiell betrieben werden, müssen Sie zum Terminieren immer beide (!) DIP-Schalter eines Kanals nach rechts umlegen.
Wenn sich das FlexRay-Modul nicht an einem physikalischen Ende des Busses befinden, darf nicht terminiert werden.

Die beiden FlexRay-Schnittstellen können per DIP-Schalter zu einem startfähigen FlexRay-Cluster zusammengeschlossen werden. Für den Zusammenschluss der Kanäle A und B müssen Sie immer beide (!) DIP-Schalter eines Kanals nach oben umlegen (siehe Abb. 165).

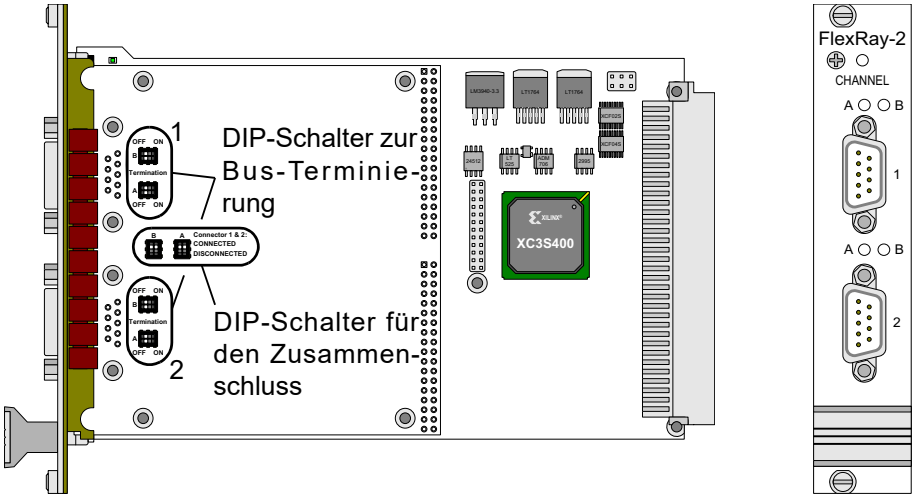


Abb. 165 – Pro II-FlexRay-2 Rev. E: Platine und Frontplatte

Modul-Revisionen

Die Unterschiede der Revisionsstände sind nachfolgend dargestellt:

Revision	Ausgabe-datum	Änderungen zur Vorgänger-Version
E1	08 / 2009	Erst-Version

Die Revisionsbezeichnung befindet sich auf der Frontseite.

Unterschiedliche Revisions-Buchstaben bedeuten unterschiedliche Modul-Eigenschaften und sind separat dokumentiert. Der Revisionsbezeichnung angehängt ist eine Zählnummer für interne Zwecke.

Programmierung

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

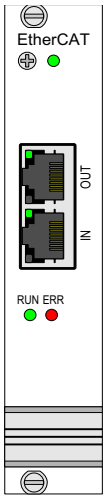
Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
FlexRay-Schnittstelle initialisieren	`P2_FlexRay_Init`
FlexRay-Controller zurücksetzen	`P2_FlexRay_Reset`
Schnittstellenversion abfragen	`P2_FlexRay_Get_Version`
Register setzen und lesen	`P2_FlexRay_Read_Word` `P2_FlexRay_Write_Word`
Modul-LED einstellen	`P2_Check_LED`, `P2_Set_LED` `P2_FlexRay_Set_LED`

5.8.15 Pro II-EtherCAT-SL Rev. E

Das Modul Pro II-EtherCAT-SL Rev. E stellt einen Feldbusknoten mit der Funktionalität eines EtherCAT-Slave zur Verfügung. Alle Einstellungen werden per Software vorgenommen.

Das Modul Pro II-EtherCAT-SL-40 Rev. E bietet einen größeren Eingangs- und Ausgangsbereich.



Funktionsbeschreibung

Nach dem Einschalten müssen Sie den Feldbusknoten in *ADbasic* initialisieren. Mit der Initialisierung wird die Größe der Ein- und Ausgangsbereiche festgelegt.

Es gibt je einen Bereich für eingehende und für ausgehende Daten; jeder Bereich hat eine Größe von 16 Long (= 64 Byte). Die Begriffe „Eingang“ und „Ausgang“ sind aus Sicht des Feldbus-Controllers zu sehen.

Bei der Datenübertragung mit einem einzelnen Datentyp (Long, Float) wird eine Geschwindigkeit von 10 kHz erreicht. Es können auch beide Datentypen nebeneinander übertragen werden, dann werden 5 kHz erreicht.

Hardware

Die Schnittstelle hat je eine Buchse vom Typ RJ45 für den Dateneingang (`IN`) und den Datenausgang (`OUT`). An jeder Buchse ist oben rechts eine LED „`Link / Activity`“, die den Betriebszustand des Knotens im EtherCAT-Bus anzeigt. Die beiden weiteren LEDs (jeweils unten an der Buchse) sind ohne Funktion.

Unterhalb der Buchsen befinden sich LEDs, die den Status der EtherCAT-Zustandsmaschine (`RUN`) und das Auftreten von Kommunkationsfehlern (`ERR`) anzeigen.

LED	Status	Bedeutung
`Link / Activity`	Aus	Nicht online (oder keine Stromversorgung).
	An	Feldbusknoten online, kein Datenaustausch.
	flackernd	Feldbusknoten online, mit Datenaustausch.
`RUN`	Aus	Status INIT: Die Schnittstelle wird initialisiert (oder keine Stromversorgung).
	blinkt grün	Status PRE-OP: Schnittstelle hat Kontakt zum Bus-Master.
	leuchtet einmal grün	Status SAFE-OP: Schnittstelle kann Daten vom Bus lesen, aber nicht senden.
	leuchtet grün	Status OP: Schnittstelle ist vollständig eingerichtet, Ein- und Ausgänge sind aktiv.
	leuchtet rot	Status EXCEPTION: Ausnahmesituation.
`ERR`	Aus	Kommunikation arbeitet ohne Fehler (oder keine Stromversorgung).
	blinkt rot	Fehler bei der Konfiguration.
	leuchtet einmal rot	Lokaler Fehler in der Schnittstelle; der EtherCAT-Status wurde geändert.
	leuchtet doppelt rot	Fehler durch Zeitüberschreitung (timeout).
	leuchtet rot	Kritischer Kommunikationsfehler.

Abb. 166 – Pro II-EtherCAT-SL Rev. E: Bedeutung der LED

Wenn beide LEDs RUN und ERR rot leuchten, ist ein gravierender Fehler in der Schnittstelle aufgetreten. Melden Sie sich dann bitte beim Support von Jäger Messtechnik; die Adresse finden Sie auf der vorderen Umschlagseite des Handbuchs, innen.

EtherCAT projektieren

Sie projektieren den EtherCAT mit einem – zum Bus-Master passenden – Konfigurations-Tool. Für das folgende Beispiel wurde das Programm „TwinCAT System Manager“ der Firma Beckhoff als EtherCAT-Master verwendet.

Für andere Konfigurations-Tools gilt die folgende Ablaufbeschreibung entsprechend. Entnehmen Sie die genaue Vorgehensweise bei der Busprojektierung der Dokumentation Ihres Konfigurations-Tools.

– Kopieren Sie die Beschreibungsdatei ADwin-EtherCAT.xml des Feldbusknotens von C:\ADwin\Fieldbus\EtherCAT in das Quellverzeichnis des Konfigurations-Tools.

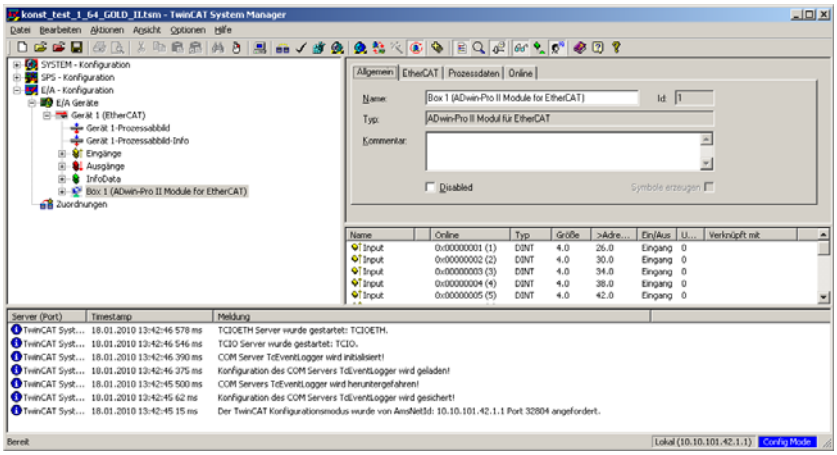
  Beim Starten lädt das Konfigurations-Tool die benötigten Informationen über den neuen Slave aus der zugehörigen Beschreibungsdatei.

– Fügen Sie die *ADwin*-EtherCAT-Slave als Busteilnehmer zum EtherCAT-Bus hinzu.

  Im TwinCAT System Manager markieren Sie dazu den EtherCAT-Master und wählen im Kontextmenü (rechte Maustaste) den Menüpunkt Boxen scannen.
  Ihnen wird eine Liste aller Busteilnehmer angezeigt.

– Wählen Sie aus der Liste den *ADwin*-EtherCAT-Slave; damit ist der Slave als Busteilnehmer bestätigt.

  Danach könnte das EtherCAT-Layout wie folgt aussehen:



– Konfigurieren Sie den *ADwin*-EtherCAT-Slave in einem *ADbasic*-Programm mit dem Befehl **P2_ECAT_Init**.

– Lesen Sie die Konfiguration im Konfigurations-Tool aus.

  Im TwinCAT System Manager markieren Sie dazu den *ADwin*-EtherCAT-Slave und klicken auf die Schaltfläche Lade PDO Info aus dem Gerät.

  Die Konfiguration des Slave muss der Konfiguration im *ADbasic*-Programm entsprechen; einstellbar ist die Datenübertragung mit nur einem Datentyp (Long oder Float) oder mit beiden Datentypen. Für jeden verwendeten Datentyp sind 16 Werte (4 Byte) für Eingänge und 16 Werte (4 Byte) für Ausgänge einzurichten.

Modul-Revisionen

Die Unterschiede der Revisionsstände sind nachfolgend dargestellt:

Revision	Ausgabe-datum	Änderungen zur Vorgänger-Version
E1	Dezember 2009	Erst-Version
E2	Dezember 2012	Firmware 2.0: Übertragung auch mit Datentyp Float

Die Revisionsbezeichnung befindet sich auf der Frontseite.

Unterschiedliche Revisions-Buchstaben bedeuten unterschiedliche Modul-Eigenschaften und sind separat dokumentiert. Der Revisionsbezeichnung angehängt ist eine Zählnummer für interne Zwecke.

Programmieren in *ADbasic*

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Datenbereiche initialisieren	`P2_ECAT_Init`, `P2_Set_Mode`
Version der Schnittstelle lesen	`P2_ECAT_Get_Version`
Betriebsstatus der Schnittstelle lesen	`P2_ECAT_Get_State`
Daten schreiben und lesen, Format Long	`P2_ECAT_Read_Data_16L` `P2_ECAT_Write_Data_16L`
Daten schreiben und lesen, Format Float	`P2_ECAT_Read_Data_16F` `P2_ECAT_Write_Data_16F`

Die Initialisierung muss mit niedriger Priorität ablaufen, da sie einige Sekunden in Anspruch nimmt; bei hoher Priorität würde der PC nach einer bestimmten Zeit (time-out) die Kommunikation abbrechen. Aus dem gleichen Grund sollte auch das Schreiben und Lesen von Daten mit niedriger Priorität ablaufen.

Spezifikationen

Der Feldbusknoten entspricht den internationalen Standards IEC 61158 (Protokolle und Dienste) und IEC 61784-2 (Kommunikationsprofile für die spezifischen Geräteklassen). Nähere Informationen erhalten Sie von der EtherCAT-Nutzerorganisation:

EtherCAT Technology Group
Ostendstraße 196
90482 Nürnberg
Tel.: +499115405620
Fax : +499115405629
https://www.ethercat.org/

Betriebszustände der EtherCAT-Schnittstelle

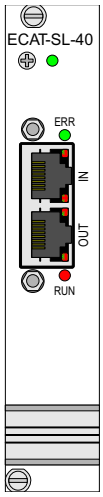
Die nachfolgende Tabelle zeigt die Betriebszustände, die die EtherCAT-Schnittstelle unterstützt.

Betriebs-Zustand	Verhalten
Init	Der EtherCAT-Slave wird vom Bus-Master initialisiert.
Boot	Der EtherCAT-Slave befindet sich im Boot-Modus.
PreOp	Die Schnittstelle nimmt am Datenverkehr teil, Ein- und Ausgänge sind noch inaktiv.
SafeOp	Die Schnittstelle kann Daten empfangen, die Ausgänge sind noch inaktiv.
Op	Die Schnittstelle ist vollständig betriebsbereit; Ein- und Ausgänge sind aktiv.

Abb. 167 – Pro II-EtherCAT-SL Rev. E: Betriebszustände

5.8.16 Pro II-EtherCAT-SL-40 Rev. E

Das Modul Pro II-EtherCAT-SL-40 Rev. E stellt einen Feldbusknoten mit der Funktionalität eines EtherCAT-Slave zur Verfügung. Alle Einstellungen werden per Software vorgenommen.



Funktionsbeschreibung

Nach dem Einschalten müssen Sie den Feldbusknoten in *ADbasic* initialisieren. Mit der Initialisierung wird die Größe der Ein- und Ausgangsbereiche festgelegt.

Es gibt je einen Bereich für eingehende und für ausgehende Daten; jeder Bereich hat eine maximale Größe von 1440 Byte (=360 Doppelworte). Die Begriffe „Eingang“ und „Ausgang“ sind aus Sicht des Feldbus-Masters zu sehen.

Bei der Initialisierung legen Sie fest, wie groß die Datenbereiche für EIngang und Ausgang sind.

TiCo-Prozessor

Das Modul besitzt einen frei programmierbaren *TiCo*-Prozessor mit 28 KiB Programmspeicher und 28 KiB Datenspeicher. Den *TiCo*-Prozessor programmieren Sie in *TiCoBasic*.

Der *TiCo*-Prozessor hat – wie auch die *ADwin*-CPU – Zugriff auf den CAN-Controller. Nähere Hinweise zur Anwendung und Programmierung des *TiCo*-Prozessors entnehmen Sie bitte dem Handbuch *TiCoBasic*.

Wenn Sie ein *TiCoBasic*-Programm im *TiCo*-Bootloader ablegen, wird das Programm beim Einschalten der Stromversorgung in den *TiCo*-Prozessor geladen und gestartet. Auf diese Weise kann das Modul eigenständig und unabhängig vom CPU-Modul des *ADwin-Pro II*-Systems arbeiten.

Hardware

Die Schnittstelle hat zwei Buchsen vom Typ RJ45, die mit `IN` und `OUT` bezeichnet sind.

In jeder Buchse befindet sich rechts oben eine LED „`Link / Activity`“, die den Betriebszustand des Knotens im EtherCAT-Bus anzeigt. Die LED rechts unten ist ohne Funktion.

Ober- und unterhalb der Buchsen befinden sich LEDs, die den Status der EtherCAT-Zustandsmaschine (`RUN`) und das Auftreten von Kommunikationsfehlern (`ERR`) anzeigen.

LED	Status	Bedeutung
Link / Activity	Aus	Nicht online (oder keine Stromversorgung).
	An	Feldbusknoten online, kein Datenaustausch.
	flackernd	Feldbusknoten online, mit Datenaustausch.
RUN	Aus	Status INIT: Die Schnittstelle wird initialisiert (oder keine Stromversorgung).
	blinkt grün	Status PRE-OP: Schnittstelle hat Kontakt zum Bus-Master.
	leuchtet einmal grün	Status SAFE-OP: Schnittstelle kann Daten vom Bus lesen, aber nicht senden.
	leuchtet grün	Status OP: Schnittstelle ist vollständig eingerichtet, Ein- und Ausgänge sind aktiv.
	leuchtet rot	Status EXCEPTION: Ausnahmesituation.
ERR	Aus	Kommunikation arbeitet ohne Fehler (oder keine Stromversorgung).
	blinkt rot	Fehler bei der Konfiguration.
	leuchtet einmal rot	Lokaler Fehler in der Schnittstelle; der EtherCAT-Status wurde geändert.
	leuchtet doppelt rot	Fehler durch Zeitüberschreitung (timeout).
	leuchtet rot	Kritischer Kommunikationsfehler.

Abb. 168 – Pro II-EtherCAT-SL-40 Rev. E: Bedeutung der LED

Wenn beide LEDs RUN und ERR rot leuchten, ist ein gravierender Fehler in der Schnittstelle aufgetreten. Melden Sie sich dann bitte beim Support von Jäger Messtechnik; die Adresse finden Sie auf der vorderen Umschlagseite des Handbuchs, innen.

EtherCAT projektieren

Sie projektieren den EtherCAT mit einem – zum Bus-Master passenden – Konfigurations-Tool. Für das folgende Beispiel wurde das Programm „TwinCAT System Manager“ der Firma Beckhoff (Version 3.1) als EtherCAT-Master verwendet.

Für andere Konfigurations-Tools gilt die folgende Ablaufbeschreibung entsprechend. Entnehmen Sie die genaue Vorgehensweise bei der Busprojektierung der Dokumentation Ihres Konfigurations-Tools.

– Konfigurieren Sie den *ADwin*-EtherCAT-Slave in einem *ADbasic*-Programm mit dem Befehl **P2_Init_EtherCAT**.

– Kopieren Sie die Beschreibungsdatei des Feldbusknotens HMS CompactCom 40 EtherCAT 2_08.xml aus dem Ordner C:\ADwin\Fieldbus\EtherCAT in das Quellverzeichnis des Konfigurations-Tools.

  Beim Starten lädt das Konfigurations-Tool die benötigten Informationen über den neuen Slave aus der zugehörigen Beschreibungsdatei.

– Fügen Sie den *ADwin*-EtherCAT-Slave als Busteilnehmer zum EtherCAT-Bus hinzu.

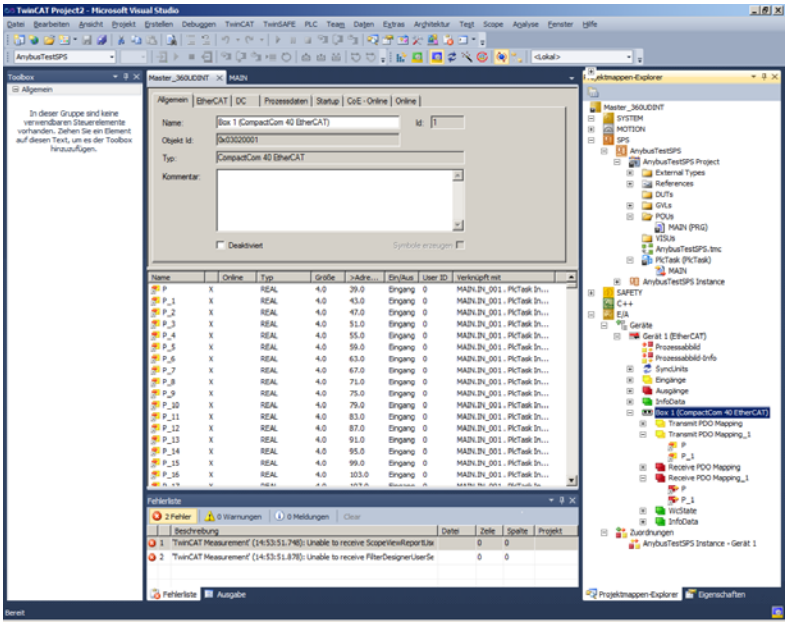
  Im TwinCAT System Manager markieren Sie dazu den EtherCAT-Master und wählen im Kontextmenü (rechte Maustaste) den Menüpunkt

`Scan`.
Ihnen wird eine Liste aller Busteilnehmer angezeigt.

– Wählen Sie aus der Liste den *ADwin*-EtherCAT-Slave; damit ist der Slave als Busteilnehmer bestätigt.

– Lesen Sie die Konfiguration im Konfigurations-Tool aus.

  Im TwinCAT System Manager markieren Sie dazu den *ADwin*-EtherCAT-Slave und klicken auf die Schaltfläche `Lade PDO Info aus dem Gerät`.



– Damit ist die Busprojektierung abgeschlossen und das Modul betriebsbereit.

Modul-Revisionen

Die Unterschiede der Revisionsstände sind nachfolgend dargestellt:

Revision	Ausgabe-datum	Änderungen zur Vorgänger-Version
E1	Dezember 2019	Erst-Version

Die Revisionsbezeichnung befindet sich auf der Frontseite.

Unterschiedliche Revisions-Buchstaben bedeuten unterschiedliche Modul-Eigenschaften und sind separat dokumentiert. Der Revisionsbezeichnung angehängt ist eine Zählnummer für interne Zwecke.

Programmieren in *ADbasic*

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Reset, Datenbereiche initialisieren	`P2_Init_EtherCAT`
Daten schreiben und lesen, Nachrichtenverwaltung	`P2_Run_EtherCAT`

Programmierung in TiCoBasic

Das Modul kann mit *TiCoBasic*-Befehlen programmiert werden. Die Befehle sind in der Online-Hilfe *TiCoBasic* erläutert.

Die Include-Datei `Fieldbus_TiCo.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Reset, Datenbereiche initialisieren	`Init_EtherCAT`
Daten schreiben und lesen, Nachrichtenverwaltung	`Run_EtherCAT`

Programmierung TiCo-Zugriff

Für den Zugriff auf den *TiCo*-Prozessor von der ADwin CPU sind die folgenden *ADbasic*-Befehle in der Datei `ADwinPro_All.inc` definiert. Die Befehle sind im Handbuch *TiCoBasic* und in der Online-Hilfe *ADbasic* erläutert.

Bereich	Befehle

Datenaustausch mit dem *TiCo*-Prozessor über globale Variablen	`P2_TDrv_Init`
	`P2_GetData_Long`, `P2_Get_Par`, `P2_Get_Par_Block`
	`P2_SetData_Long`, `P2_Set_Par`, `P2_Set_Par_Block`
	`P2_Get_TiCo_RingBuffer`, `P2_Set_TiCo_RingBuffer`
	`P2_RingBuffer_Empty`
	`P2_RingBuffer_Full`

TiCo-Prozessor steuern	`P2_TiCo_Reset`, `P2_TiCo_Start`, `P2_TiCo_Stop`
	`P2_Get_TiCo_Bootloader_Status`
	`P2_Get_TiCo_Status`, `P2_Workload`

TiCo-Prozesse steuern	`P2_Process_Status`
	`P2_TiCo_Get_Processdelay`
	`P2_TiCo_Set_Processdelay`
	`P2_TiCo_Start_Process`
	`P2_Tico_Stop_Process`

TiCo-Programme übertragen	`P2_TiCo_Flash`, `P2_TiCo_Load`

Spezifikationen

Der Feldbusknoten entspricht den internationalen Standards IEC 61158 (Protokolle und Dienste) und IEC 61784-2 (Kommunikationsprofile für die spezifischen Geräteklassen). Nähere Informationen erhalten Sie von der EtherCAT-Nutzerorganisation:

EtherCAT Technology Group
Ostendstraße 196
90482 Nürnberg
Tel.: +499115405620
Fax : +499115405629
https://www.ethercat.org/

Betriebszustände der EtherCAT-Schnittstelle

Die nachfolgende Tabelle zeigt die Betriebszustände, die die EtherCAT-Schnittstelle unterstützt.

Betriebs-Zustand	Verhalten
Init	Der EtherCAT-Slave wird vom Bus-Master initialisiert.
Boot	Der EtherCAT-Slave befindet sich im Boot-Modus.
PreOp	Die Schnittstelle nimmt am Datenverkehr teil, Ein- und Ausgänge sind noch inaktiv.
SafeOp	Die Schnittstelle kann Daten empfangen, die Ausgänge sind noch inaktiv.
Op	Die Schnittstelle ist vollständig betriebsbereit; Ein- und Ausgänge sind aktiv.

Abb. 169 – Pro II-EtherCAT-SL-40 Rev. E: Betriebszustände

5.8.17 Pro II-SENT-4 Rev. E

Das Modul Pro II-SENT-4 Rev. E besitzt 4 Eingangskanäle für das SENT-Protokoll (Single Edge Nibble Transmission) und wertet die eintreffenden SENT-Nachrichten auf allen Kanälen automatisch aus.

Die Beschreibung des SENT-Moduls ist in folgende Abschnitte gegliedert:

- Hardware-Aufbau
- Mit SENT arbeiten
- Modul-Revisionen
- Programmierung

SENT-Kanäle	4 Eingänge
Signalspannung	Low-Pegel < 0,5V, High-Pegel > 4,1V
Versorgungsspannung	5V
Steckerverbindung	37-polige D-Sub-Buchse

Abb. 170 – Pro II-SENT-4 Rev. E: Spezifikation

Hardware-Aufbau

Das Modul Pro II-SENT-4 Rev. E arbeitet nach der der Norm SAE J2716, Stand 2010. Weitere Informationen finden Sie auf der Webseite von SAE International®: https://standards.sae.org.

Die Anschlüsse der SENT-Schnittstelle stehen auf einem 37-poligen D-Sub-Verbinder zur Verfügung; die Pin-Belegung ist unten dargestellt.

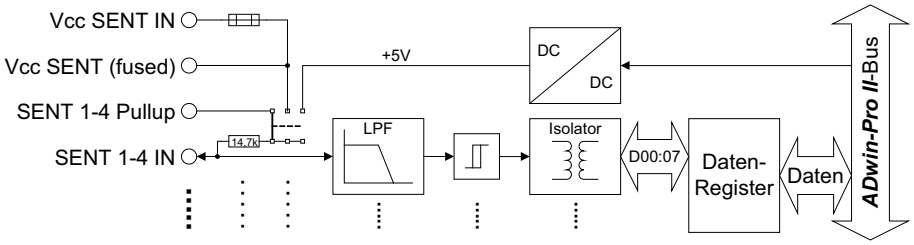


Abb. 171 – Pro II-SENT-4 Rev. E: Blockschaltbild

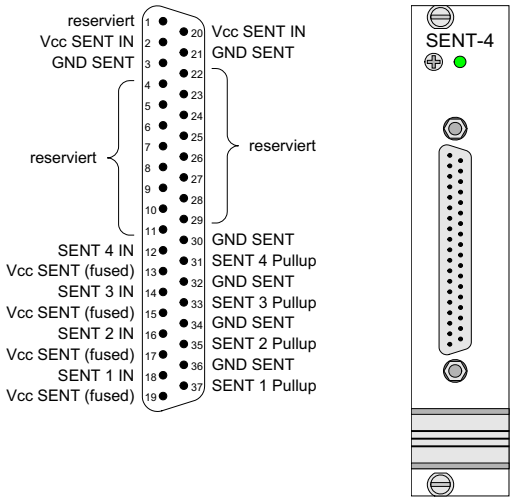


Abb. 172 – Pro II-SENT-4 Rev. E: Pinbelegung und Frontplatte

Versorgungsspannung

Über die Pins `Vcc SENT In` können Sie eine Versorgungsspannung von 5 Volt mit bis zu 400 mA einspeisen; die Pins `Vcc SENT In` sind miteinander verbunden. Die Spannung wird an den 4 Spannungsausgängen `Vcc SENT (fused)`

ausgegeben und kann (siehe unten) als Pull-up-Spannung genutzt werden. Zwischen Spannungseingängen und -ausgängen ist eine Sicherung von 500 mA geschaltet.

Beachten Sie: In Rev. E01 ist in der Eingangsbeschaltung für die Versorgungsspannung ein Widerstand 51 kΩ bestückt, ab Rev. E02 ein Widerstand 14,7 kΩ.

Pull-up-Signal

Die SENT-Kanäle können auf verschiedene Arten mit einem Pull-up-Signal beschaltet werden. Sie wählen die Beschaltung aus, indem Sie einen Jumper auf der Platine umstecken:

– Ein modulinterner DC/DC-Wandler liefert eine gemeinsame Pull-up-Spannung von 5 Volt.
– Die externe Versorgungsspannung `Vcc SENT In` wird gleichzeitig als gemeinsame Pull-up-Spannung verwendet.
– Externe Einspeisung der Pull-up-Spannungen über die Eingänge `SENT 1…4 Pullup`. Für jeden SENT-Kanal gibt es eine separate Einspeisung.

  Die Pull-up-Spannungen sollen 5 Volt betragen.

Auf der Platine sind 3 Reserve-Jumper gesteckt (unten links, siehe Abb. 173), falls einer der Pull-up-Jumper verloren geht.

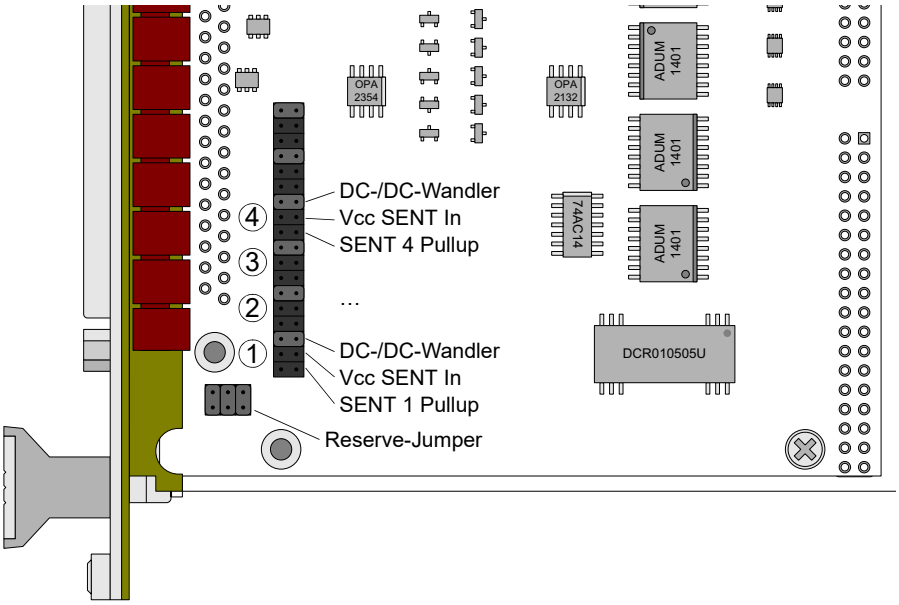


Abb. 173 – Pro II-SENT-4 Rev. E: Jumperpositionen auf der Platine

Mit SENT arbeiten

Begriffsbestimmung

Das SENT-Protokoll arbeitet als Direktverbindung, die Signalwerte von einem Sensor an einen Controller überträgt. Eine SENT-Nachricht besteht aus 9 oder 10 Pulsen:

– 1 Kalibrierpuls zur Synchronisierung
– 1 Nibble-Puls (1 Nibble = 4 Bit): Status und Kommunikation, enthält u. a. 1…2 Bits der seriellen Nachricht (= slow channel).
– 3 Nibble-Pulse: erster 12 Bit-Wert eines Sensors (Signal 1 / fast channel 1)
– 3 Nibble-Pulse: zweiter 12 Bit-Wert eines Sensors (Signal 2 / fast channel 2)
– 1 Nibble-Puls: CRC-Prüfsumme über die Datenpulse
– optional 1 Pausenpuls

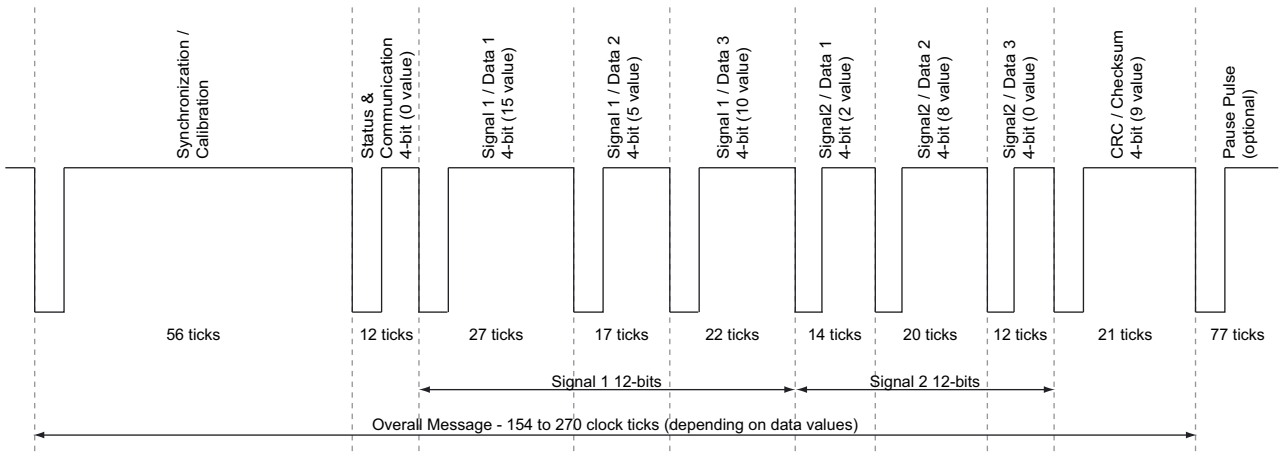


Abb. 174 – Beispiel SENT-Nachricht (Quelle: SAE J2716)

Die Länge von Pulsen und SENT-Nachrichten wird in Basistakten gemessen. Der Basistakt kann 1,5µs…90µs betragen.

Eine serielle Nachricht besteht aus der Kennung (ID), einem Datenwert und einer CRC-Prüfsumme. Das Sendeformat (standard, enhanced 1, enhanced 2) legt fest, wie viele Bits für ID-Nummer und Datenwert verwendet werden. Serielle Nachrichten werden über mehrere SENT-Nachrichten verteilt gesendet, und zwar nur 1…2 Bits je Nachricht (im Nibble Status & Kommunikation).

Mehrere (bis zu 32) serielle Nachrichten bilden zusammen einen Nachrichtensatz. Der Nachrichtensatz wird zyklisch übertragen.

Erkennungsmodus

Nach dem Einschalten sind alle Eingangskanäle im Erkennungsmodus, in dem eingehende SENT-Nachrichten analysiert werden. Sobald das Modul den Basistakt und die Pulsanzahl der SENT-Nachrichten erkannt hat, schaltet es den entsprechenden Eingangskanal in den Lesemodus.

Man kann alternativ Pulsanzahl und Basistakt eines Eingangskanals auch per Software festlegen (`P2_SENT_Set_PulseCount`, `P2_SENT_Set_ClockTick`). Mit dem Festlegen des Basistakts wird der Empfangskanal ohne weitere Prüfung in den Lesemodus geschaltet.

Man kann einen Eingangskanal aus dem Lesemodus per Software (Befehl `P2_SENT_Set_Detection`) wieder in den Erkennungsmodus schalten. Das ist beispielsweise sinnvoll, wenn der SENT-Sensor gewechselt wurde.

Lesemodus

Im Lesemodus wertet das Modul die SENT-Nachrichten der Eingangskanäle kontinuierlich aus und stellt die jeweils enthaltenen Daten zum Abruf bereit. Die Einstellungen Basistakt, Pulsanzahl und CRC-Modus werden berücksichtigt. Folgende Daten sind abrufbar:

– Signale 1 und 2 (fast channel) und das Ergebnis der CRC-Prüfung:
  `P2_SENT_Get_Fast_Channel1`
  `P2_SENT_Get_Fast_Channel2`
  `P2_SENT_Get_Fast_Channel_CRC_OK`

– Kennung (ID) und Datenwert der seriellen Nachricht (slow channel) und das Ergebnis der CRC-Prüfung.
  `P2_SENT_Get_Serial_Message_Id`
  `P2_SENT_Get_Serial_Message_Data`
  `P2_SENT_Get_Serial_Message_CRC_OK`
  `P2_SENT_Get_Serial_Message_Array`

  Das Format der seriellen Nachricht wird automatisch erkannt:
  - short serial message format: 4 Bit ID + 8 Bit Daten.
  - enhanced serial message format:
    8 Bit ID + 12 Bit Daten oder 4 Bit ID + 16 Bit Daten.

Man kann serielle Nachrichten entweder einzeln in der empfangenen Reihenfolge lesen oder einen vollständigen Nachrichtensatz auf einmal. Zu einem Nachrichtensatz gehört für jede verwendete Kennung jeweils die zuletzt empfangene Nachricht sowie die Anzahl der empfangenen Nachrichten je Kennung.

– Nur auf Anforderung (`P2_SENT_Request_Latch`) sammelt das Modul alle Daten der nächsten SENT-Nachricht in einem Latch-Zwischenspeicher, von wo die Daten anschließend ausgelesen werden können (`P2_SENT_Get_Latched_Data`). Damit ist sichergestellt, dass alle Daten zur gleichen SENT-Nachricht gehören.

  Der Datensatz enthält unter anderem alle 8 Nibbles sowie den Empfangszeitpunkt der SENT-Nachricht.

Timeout prüfen

Sie können abfragen, ob an einem Eingangskanal kontinuierlich SENT-Nachrichten eintreffen oder nicht: Solange sich der Rückgabewert von `P2_SENT_Get_Msg_Counter` ändert, treffen neue SENT-Nachrichten am Eingangskanal ein.

CRC-Modus

Mit `P2_SENT_Set_CRC_Implementation` kann die CRC-Prüfung vom Berechnungsalgorithmus „Legacy“ auf „Recommended“ umgeschaltet werden.

Der Berechnungsalgorithmus gilt sowohl für die Signale 1 und 2 als auch für die serielle Nachricht.

Sensortyp einstellen

Mit `P2_SENT_Set_Sensor_Type` wird der erwartete Sensortyp eines SENT-Kanals eingestellt. Das Modul prüft daraufhin die eingehenden Nachrichten, ob sie auch den Kriterien für den eingestellten Sensortyp gemäß SENT-Spezifikation entsprechen. Das Prüfergebnis kann mit einem Fehlercode abgefragt werden.
Derzeit werden die ty unterstützt.

Modul-Revisionen

Die Unterschiede der Hardware-Revisionen sind nachfolgend dargestellt. Die Revisionsbezeichnung befindet sich auf der Frontseite.

Revision	Ausgabe-datum	Änderungen zur Vorgänger-Version
E01	06 / 2012	Erstversion
E02	10 / 2013	Änderung der Eingangsbeschaltung von „Legacy SENT System Interface Circuit Topology“ (SENT Spec. Fig. 6.3.3) zu „Recommended Sent System Interface Circuit Topology“ (SENT Spec. Fig. 6.3.4)
E03	11 / 2013	Aufsatzkarte mit Beschaltungsvarianten.

Unterschiedliche Revisions-Buchstaben bedeuten unterschiedliche Modul-Eigenschaften und sind separat dokumentiert. Der Revisionsbezeichnung angehängt ist eine Zählnummer für interne Zwecke.

Die Schnittstellenversion ist von außen nicht ablesbar, sondern kann per Software abgefragt werden:

Software-Version	Ausgabe-datum	Änderungen zur Vorgänger-Version
10	06 / 2012	Erste offizielle Version
11	07 / 2012	CRC-Modus einstellen.
12	08 / 2012	Sensortyp eines Kanals gemäß SENT-Spezifikation einstellen.
13	08 / 2012	Pausenpuls für eine SENT-Nachricht einstellen.
14	09 / 2013	Alle Serial Messages auf einmal lesen.

Software-Version	Ausgabe-datum	Änderungen zur Vorgänger-Version
16	09 / 2013	Gesammelte serielle Nachrichten zurücksetzen.
17	11 / 2013	Einführung des Latch-Zwischenspeichers.
18	01 / 2014	Änderungen am Latch-Zwischenspeicher.
19	04 / 2014	Include-Dateien überarbeitet.
20	11 / 2014	Performance-Optimierung.

Programmierung

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Beachten Sie: Bei manchen Befehlen müssen Sie erst mit **P2_SENT_Command_Ready** prüfen, ob die SENT-Schnittstelle schon wieder für die Verarbeitung des Befehls bereit ist. Solche Befehle sind in der unten stehenden Tabelle mit * markiert.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Versionsabfrage	**P2_SENT_Get_Version**
Schnittstelle initialisieren	**P2_SENT_Init**
Befehlsverarbeitung	**P2_SENT_Command_Ready**
Auf Timeout prüfen	**P2_SENT_Get_Msg_Counter**
SENT-Parameter lesen	**P2_SENT_Get_ChannelState** **P2_SENT_Get_ClockTick** **P2_SENT_Get_PulseCount**
SENT-Parameter setzen	**P2_SENT_Set_CRC_Implementation*** **P2_SENT_Set_Detection*** **P2_SENT_Set_ClockTick*** **P2_SENT_Set_PulseCount*** **P2_SENT_Set_Sensor_Type***
Signale lesen	**P2_SENT_Get_Fast_Channel1** **P2_SENT_Get_Fast_Channel2** **P2_SENT_Get_Fast_Channel_CRC_OK**
Serielle Nachricht lesen	**P2_SENT_Get_Serial_Message_Id** **P2_SENT_Get_Serial_Message_Data** **P2_SENT_Get_Serial_Message_CRC_OK** **P2_SENT_Get_Serial_Message_Array** **P2_SENT_Clear_Serial_Message_Array***
SENT-Nachricht mittels Latch sammeln	**P2_SENT_Request_Latch*** **P2_SENT_Check_Latch** **P2_SENT_Get_Latched_Data**
Modul-LED einstellen	**P2_Check_LED**, **P2_Set_LED**

5.8.18 Pro II-SENT-6 Rev. E

Das Modul Pro II-SENT-6 Rev. E besitzt 6 Eingangskanäle für das SENT-Protokoll (Single Edge Nibble Transmission) und wertet die eintreffenden SENT-Nachrichten auf allen Kanälen automatisch aus. Zusätzlich stehen 4 digitale Eingangs- und 4 digitale Ausgangskanäle mit TTL-Pegeln bereit.

Die Beschreibung des SENT-Moduls ist in folgende Abschnitte gegliedert:

SENT-Schnittstelle	
SENT-Kanäle	6 Eingänge
Signalspannung	Low-Pegel < 0,5V, High-Pegel > 4,1V
Versorgungsspannung	5V
Digitalkanäle	
Digitalkanäle	4 Eingänge, 4 Ausgänge; TTL-Logik
Pull-Down-Widerstand	10kΩ
V_{IH}	min. 2V
V_{IL}	max. 0,8V
I_{IH}	max. 1µA
I_{IL}	max. 0,01mA
Spannungsbereich	-0,5V … +5,5V
Ausgangsstrom	max. ±35mA pro Kanal, max. ±70mA je Block (4 Kanäle) über VCC oder GND
Allgemeines	
Steckerverbindung	37-polige D-Sub-Buchse

Abb. 175 – Pro II-SENT-6 Rev. E: Spezifikation

Hardware-Aufbau

Das Modul Pro II-SENT-6 Rev. E arbeitet nach der der Norm SAE J2716, Stand 2010. Weitere Informationen finden Sie auf der Webseite von SAE International®: https://standards.sae.org.

Die Anschlüsse der SENT-Schnittstelle stehen auf einem 37-poligen D-Sub-Verbinder zur Verfügung; die Pin-Belegung ist unten dargestellt.

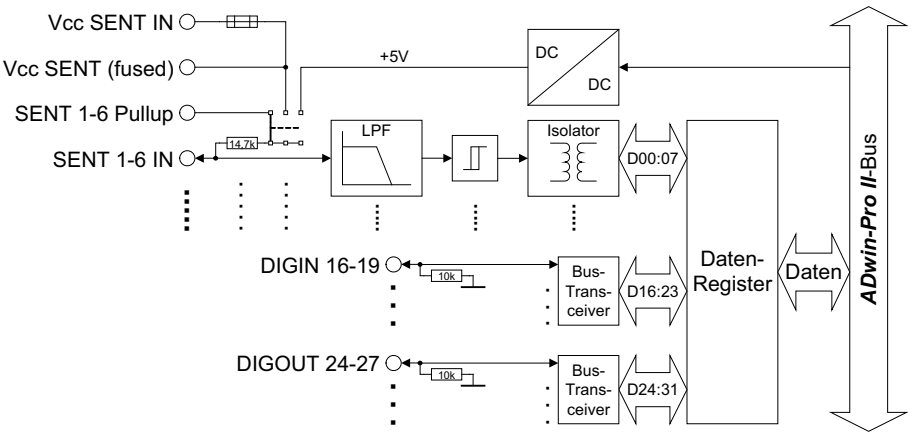


Abb. 176 – Pro II-SENT-4 Rev. E: Blockschaltbild

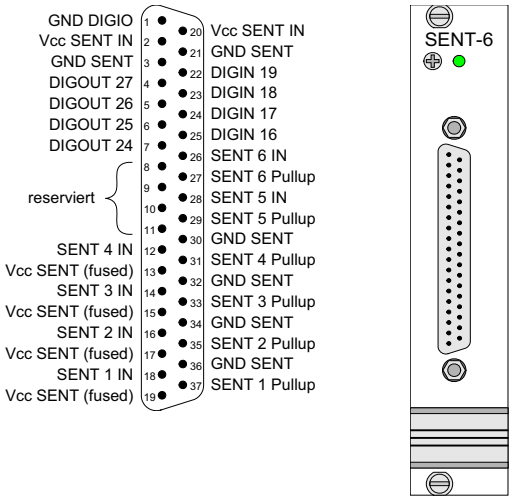


Abb. 177 – Pro II-SENT-6 Rev. E: Pinbelegung und Frontplatte

Versorgungsspannung

Über die beiden miteinander verbundenen Pins `Vcc SENT In` können Sie eine Versorgungsspannung von 5 Volt mit bis zu 400 mA einspeisen. Die Spannung wird an den 4 Spannungsausgängen `Vcc SENT (fused)` ausgegeben und kann außerdem als Pull-up-Spannung genutzt werden (siehe unten). Zwischen Spannungseingängen und -ausgängen ist eine Sicherung von 500 mA geschaltet.

Pull-up-Signal

Die SENT-Kanäle können auf verschiedene Arten mit einem Pull-up-Signal beschaltet werden. Sie wählen die Beschaltung aus, indem Sie einen Jumper auf der Platine umstecken:

- Ein modulinterner DC/DC-Wandler liefert eine gemeinsame Pull-up-Spannung von 5 Volt.
- Die externe Versorgungsspannung `Vcc SENT In` wird gleichzeitig als gemeinsame Pull-up-Spannung verwendet.
- Externe Einspeisung der Pull-up-Spannungen über die Eingänge `SENT 1…6 Pullup`. Für jeden SENT-Kanal gibt es eine separate Einspeisung.

  Die Pull-up-Spannungen sollen 5 Volt betragen.

Auf der Platine sind 3 Reserve-Jumper gesteckt (in Abb. 178 unten links), falls einer der Pull-up-Jumper verloren geht.

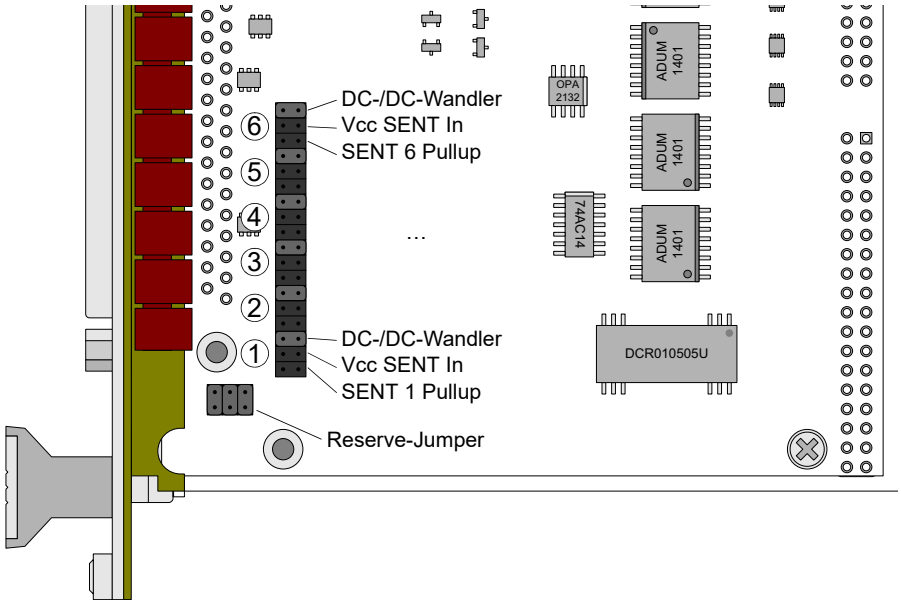


Abb. 178 – Pro II-SENT-6 Rev. E: Jumperpositionen auf der Platine

Mit SENT arbeiten

Begriffsbestimmung

Das SENT-Protokoll arbeitet als Direktverbindung, die Signalwerte von einem Sensor an einen Controller überträgt. Eine SENT-Nachricht besteht aus 9 oder 10 Pulsen:

– 1 Kalibrierpuls zur Synchronisierung
– 1 Nibble-Puls (1 Nibble = 4 Bit): Status und Kommunikation, enthält u. a. 1…2 Bits der seriellen Nachricht.
– 3 Nibble-Pulse: erster 12 Bit-Wert eines Sensors
– 3 Nibble-Pulse: zweiter 12 Bit-Wert eines Sensors
– 1 Nibble-Puls: CRC-Prüfsumme über die Datenpulse
– optional 1 Pausenpuls

Die Länge von Pulsen und SENT-Nachrichten wird in Basistakten gemessen. Der Basistakt kann 3 µs…90 µs betragen.

Eine serielle Nachricht besteht aus der Kennung (ID), einem Datenwert und einer CRC-Prüfsumme. Das Sendeformat (standard, enhanced 1, enhanced 2) legt fest, wie viele Bits für ID-Nummer und Datenwert verwendet werden. Serielle Nachrichten werden über mehrere SENT-Nachrichten verteilt gesendet, und zwar nur 1…2 Bits je Nachricht.

Mehrere (bis zu 32) serielle Nachrichten bilden zusammen einen Nachrichtensatz. Der Nachrichtensatz wird zyklisch übertragen.

Erkennungsmodus

Nach dem Einschalten sind alle Eingangskanäle im Erkennungsmodus, in dem eingehende SENT-Nachrichten analysiert werden. Sobald das Modul den Basistakt und die Pulsanzahl der SENT-Nachrichten erkannt hat, schaltet es den entsprechenden Eingangskanal in den Lesemodus.

Man kann alternativ den Basistakt und die Pulsanzahl eines Eingangskanals auch per Software festlegen. Damit wird der Empfangskanal ohne weitere Prüfung in den Lesemodus geschaltet.

Man kann einen Eingangskanal aus dem Lesemodus per Software wieder in den Erkennungsmodus schalten. Das ist beispielsweise sinnvoll, wenn der SENT-Sensor gewechselt wurde.

Lesemodus

Im Lesemodus wertet das Modul die SENT-Nachrichten der Eingangskanäle kontinuierlich aus und stellt die jeweils enthaltenen Daten zum Abruf bereit:

– Signale 1 und 2 (fast channel) und das Ergebnis der CRC-Prüfung.

  Es wird automatisch erkannt, ob die SENT-Nachricht einen Pausenpuls enthält.

– Kennung (ID) und Datenwert der seriellen Nachricht (slow channel) und das Ergebnis der CRC-Prüfung.

  Folgende Formate der seriellen Nachricht werden erkannt:
  - short serial message format: 4 Bit ID + 8 Bit Daten.
  - enhanced serial message format:
    8 Bit ID + 12 Bit Daten oder 4 Bit ID + 16 Bit Daten.

  Man kann serielle Nachrichten entweder einzeln in der empfangenen Reihenfolge lesen oder einen vollständigen Nachrichtensatz auf einmal. Zu einem Nachrichtensatz gehört für jede verwendete Kennung jeweils die zuletzt empfangene Nachricht sowie die Anzahl der empfangenen Nachrichten je Kennung.

– Nur auf Anforderung sammelt das Modul alle Daten der nächsten SENT-Nachricht in einem Latch-Zwischenspeicher, von wo die Daten anschließend ausgelesen werden können. Damit ist sichergestellt, dass alle Daten zur gleichen SENT-Nachricht gehören.

Der Datensatz enthält unter anderem alle 8 Nibbles sowie den Empfangszeitpunkt der SENT-Nachricht.

Timeout prüfen

Sie können abfragen, ob an einem Eingangskanal kontinuierlich SENT-Nachrichten eintreffen oder nicht: Solange sich der Rückgabewert von `P2_SENT_Get_Msg_Counter` ändert, treffen neue SENT-Nachrichten am Eingangskanal ein.

CRC-Modus

Mit `P2_SENT_Set_CRC_Implementation` kann die CRC-Prüfung vom Berechnungsalgorithmus „Legacy“ auf „Recommended“ umgeschaltet werden.

Der Berechnungsalgorithmus gilt sowohl für die Signale 1 und 2 als auch für die serielle Nachricht.

Sensortyp einstellen

Mit `P2_SENT_Set_Sensor_Type` wird der erwartete Sensortyp eines SENT-Kanals eingestellt. Das Modul prüft daraufhin die eingehenden Nachrichten, ob sie auch den Kriterien für den eingestellten Sensortyp gemäß SENT-Spezifikation entsprechen. Das Prüfergebnis kann mit einem Fehlercode abgefragt werden.
Derzeit werden die Sensortypen „Throttle position sensor“ und „Single secure sensor“ unterstützt.

Digitalkanäle

Es stehen 4 Digitaleingänge (DIGIN 16…19) und 4 Digitalausgänge (DIGOUT 24…27) auf einem 37-poligen D-Sub-Verbinder zur Verfügung; Pin-Belegung siehe Abb. 177.

Die Kanalnummern 0…15, 20…23 und 28…31 sind nicht belegt.

Beachten Sie: Bei den Digitalkanälen dürfen keine Befehle für Eingangs- und Ausgangs-Fifo verwendet werden. Anderenfalls arbeitet die SENT-Schnittstelle fehlerhaft.

Modul-Revisionen

Die Unterschiede der Hardware-Revisionen sind nachfolgend dargestellt. Die Revisionsbezeichnung befindet sich auf der Frontseite.

Revision	Ausgabe-datum	Änderungen zur Vorgänger-Version
E03	11 / 2013	Erstversion.

Unterschiedliche Revisions-Buchstaben bedeuten unterschiedliche Modul-Eigenschaften und sind separat dokumentiert. Der Revisionsbezeichnung angehängt ist eine Zählnummer für interne Zwecke.

Die Schnittstellenversion ist von außen nicht ablesbar, sondern kann per Software abgefragt werden:

Software-Version	Ausgabe-datum	Änderungen zur Vorgänger-Version
17	11 / 2013	Erste offizielle Version.
18	01 / 2014	Änderungen am Latch-Zwischenspeicher.
19	04 / 2014	Include-Dateien überarbeitet.
20	11 / 2014	Performance-Optimierung.

Programmierung

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Beachten Sie: Bei manchen Befehlen müssen Sie erst mit **P2_SENT_Command_Ready** prüfen, ob die SENT-Schnittstelle schon wieder für die Verarbeitung des Befehls bereit ist. Solche Befehle sind in der unten stehenden Tabelle mit * markiert.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
SENT-Schnittstelle	
Versionsabfrage	**P2_SENT_Get_Version**
Schnittstelle initialisieren	**P2_SENT_Init**
Befehlsverarbeitung	**P2_SENT_Command_Ready**
Auf Timeout prüfen	**P2_SENT_Get_Msg_Counter**
SENT-Parameter lesen	**P2_SENT_Get_ChannelState** **P2_SENT_Get_Period** **P2_SENT_Get_PulseCount**
SENT-Parameter setzen	**P2_SENT_Set_CRC_Implementation*** **P2_SENT_Set_Detection_Mode*** **P2_SENT_Set_ClockTick*** **P2_SENT_Set_PulseCount*** **P2_SENT_Set_Sensor_Type***
Signale lesen	**P2_SENT_Get_Fast_Channel1** **P2_SENT_Get_Fast_Channel2** **P2_SENT_Get_Fast_Channel_CRC_OK**
Serielle Nachricht lesen	**P2_SENT_Get_Serial_Message_Id** **P2_SENT_Get_Serial_Message_Data** **P2_SENT_Get_Serial_Message_CRC_OK** **P2_SENT_Get_Serial_Message_Array** **P2_SENT_Clear_Serial_Message_Array***
SENT-Nachricht mittels Latch sammeln	**P2_SENT_Request_Latch*** **P2_SENT_Check_Latch** **P2_SENT_Get_Latched_Data**
Digitalkanäle	
Digitaleingänge abfragen	**P2_Digin_Long**
Latch-Register der Digitalkanäle nutzen	**P2_Dig_Latch** **P2_Dig_Read_Latch** **P2_Dig_Write_Latch** **P2_Sync_All**
Digitalausgänge setzen und rücklesen	**P2_Digout**, **P2_Digout_Bits** **P2_Digout_Long** **P2_Get_Digout_Long** **P2_Digout_Set**, **P2_Digout_Reset**
Modul-LED einstellen	**P2_Check_LED**, **P2_Set_LED**

5.8.19 Pro II-SENT-4-Out Rev. E

Das Modul Pro II-SENT-4-Out Rev. E besitzt 4 Ausgangskanäle für das SENT-Protokoll (Single Edge Nibble Transmission). Zusätzlich stehen 4 Eingangs- und 4 Ausgangskanäle mit TTL-Pegeln bereit.

Die Beschreibung des SENT-Moduls ist in folgende Abschnitte gegliedert:

- Hardware-Aufbau
- Mit SENT arbeiten
- Digitalkanäle
- Modul-Revisionen
- Programmierung

SENT-Schnittstelle	
SENT-Kanäle	4 Ausgänge
Digitalkanäle	
Digitalkanäle	4 Eingänge, 4 Ausgänge; TTL-Logik
Pull-Down-Widerstand	10 kΩ
V_{IH}	min. 2 V
V_{IL}	max. 0,8 V
I_{IH}	max. 1 µA
I_{IL}	max. 0,01 mA
Spannungsbereich	-0,5 V … +5,5 V
Ausgangsstrom	max. ±35 mA pro Kanal, max. ±70 mA je Block (4 Kanäle) über VCC oder GND
Allgemeines	
Steckerverbindung	37-polige D-Sub-Buchse

Abb. 179 – Pro II-SENT-4-Out Rev. E: Spezifikation

Hardware-Aufbau

Das Modul Pro II-SENT-4-Out Rev. E arbeitet nach der der Norm SAE J2716, Stand 2010. Weitere Informationen finden Sie auf der Webseite von SAE International®: https://standards.sae.org.

Die Anschlüsse der SENT-Schnittstelle stehen auf einem 37-poligen D-Sub-Verbinder zur Verfügung; die Pin-Belegung ist unten dargestellt.

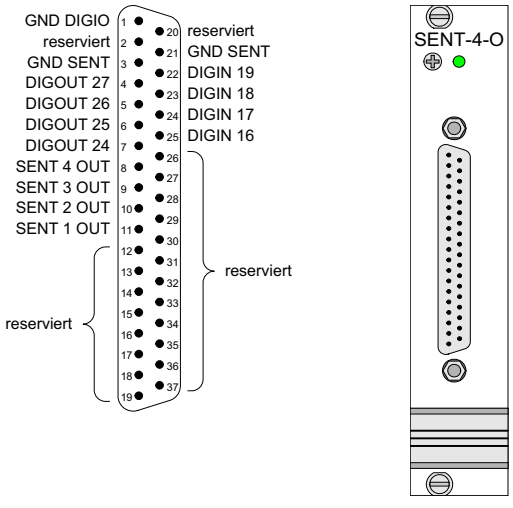


Abb. 180 – Pro II-SENT-4-Out Rev. E: Pinbelegung und Frontplatte

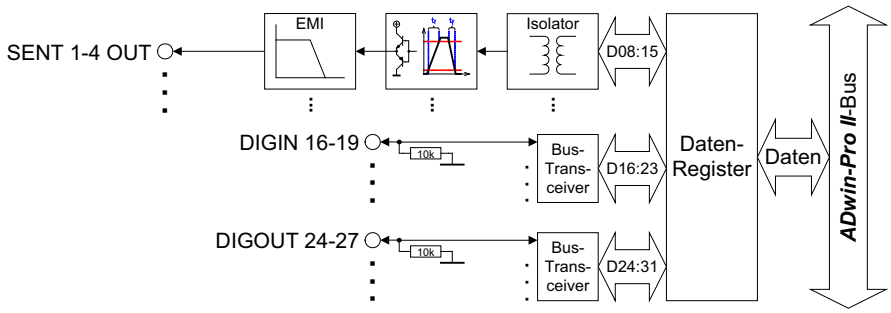


Abb. 181 – Pro II-SENT-4-Out Rev. E: Blockschaltbild

Mit SENT arbeiten

Begriffsbestimmung

Das SENT-Protokoll arbeitet als Direktverbindung, die Signalwerte von einem Sensor an einen Controller überträgt. Eine SENT-Nachricht besteht aus 9 oder 10 Pulsen:

– 1 Kalibrierpuls zur Synchronisierung
– 1 Nibble-Puls (1 Nibble = 4 Bit): Status und Kommunikation, enthält u. a. 1…2 Bits der seriellen Nachricht.
– 3 Nibble-Pulse: erster 12 Bit-Wert eines Sensors
– 3 Nibble-Pulse: zweiter 12 Bit-Wert eines Sensors
– 1 Nibble-Puls: CRC-Prüfsumme über die Datenpulse
– optional 1 Pausenpuls

Die Länge von Pulsen und SENT-Nachrichten wird in Basistakten gemessen. Der Basistakt kann 3 µs…90 µs betragen.

Eine serielle Nachricht besteht aus der Kennung (ID), einem Datenwert und einer CRC-Prüfsumme. Das Sendeformat (standard, enhanced 1, enhanced 2) legt fest, wie viele Bits für ID-Nummer und Datenwert verwendet werden. Serielle Nachrichten werden über mehrere SENT-Nachrichten verteilt gesendet, und zwar 1…2 Bits je Nachricht.

Mehrere (bis zu 32) serielle Nachrichten bilden zusammen einen Nachrichtensatz. Der Nachrichtensatz wird zyklisch übertragen.

Betriebsmodi

Das Modul unterscheidet zwei Betriebsmodi beim Senden:

– Continuous mode: Das Modul sendet kontinuierlich die gerade definierte SENT-Nachricht.

  Sie können die SENT-Nachricht per Software ändern: Signale 1/2 (fast channel), serielle Nachricht und reservierte Bits. Die CRC-Prüfsummen werden automatisch berechnet.

– FIFO mode: Sie füllen ein FIFO-Feld mit SENT-Nachrichten (max. 400), das Modul liest das FIFO-Feld aus und gibt die SENT-Nachrichten nacheinander aus. Sobald der FIFO leer läuft, wird der SENT-Kanal abgeschaltet.

  Sie stellen die Bitfolge der SENT-Nachrichten selbst zusammen. Dies betrifft insbesondere auch das Setzen der Bits für serielle Nachrichten.

Für die Ausgabe der SENT-Nachrichten können folgende Parameter für jeden Kanal eingestellt werden:

– Basistakt.
– Feste Gesamtlänge der SENT-Nachricht mittels Pausenpuls bzw. variable Gesamtlänge ohne Pausenpuls.
– Berechnungsalgorithmus der CRC-Prüfsumme „Legacy“ oder „Recommended“. Der Algorithmus gilt für die Signale 1 und 2 wie auch für die serielle Nachricht.
– Dauer des Low-Pegels am Anfang eines Pulses.
– Sendeformat der seriellen Nachricht:
  • short serial message format: 4 Bit ID + 8 Bit Daten.
  • enhanced serial message format:
    8 Bit ID + 12 Bit Daten oder 4 Bit ID + 16 Bit Daten.

Sie können abfragen, wie viele SENT-Nachrichten an einem Ausgangskanal gesendet wurden.

Digitalkanäle

Es stehen 4 Digitaleingänge (DIGIN 16…19) und 4 Digitalausgänge (DIGOUT 24…27) auf einem 37-poligen D-Sub-Verbinder zur Verfügung; Pin-Belegung siehe Abb. 180.

Die Kanalnummern 0…15, 20…23 und 28…31 sind nicht belegt.

Beachten Sie: Bei den Digitalkanälen dürfen keine Befehle für Eingangs- und Ausgangs-Fifo verwendet werden (Flankenüberwachung, Pegelausgabe). Anderenfalls arbeitet die SENT-Schnittstelle fehlerhaft.

Modul-Revisionen

Die Unterschiede der Hardware-Revisionen sind nachfolgend dargestellt. Die Revisionsbezeichnung befindet sich auf der Frontseite.

Revision	Ausgabe-datum	Änderungen zur Vorgänger-Version
E01	11 / 2013	Erstversion

Unterschiedliche Revisions-Buchstaben bedeuten unterschiedliche Modul-Eigenschaften und sind separat dokumentiert. Der Revisionsbezeichnung angehängt ist eine Zählnummer für interne Zwecke.

Die Schnittstellenversion ist von außen nicht ablesbar, sondern wird per Software mit `P2_SENT_Get_Version` abgefragt:

Software-Version	Ausgabe-datum	Änderungen zur Vorgänger-Version
1	11 / 2013	Erste offizielle Version.
2	1 / 2014	besseres Handling im Betriebsmodus Fifo mode.
3	2 / 2014	Wiedereinschalten von SENT-Kanälen verbessert.
4	3 / 2014	Höherer Datendurchsatz.
5	3 / 2014	Verbesserte PWM-Ausgabe

Programmierung

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Beachten Sie: Bei manchen Befehlen müssen Sie erst mit `P2_SENT_Command_Ready` prüfen, ob die SENT-Schnittstelle schon wieder für die Verarbeitung des Befehls bereit ist. Solche Befehle sind in der unten stehenden Tabelle mit * markiert.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
SENT-Schnittstelle	
Versionsabfrage	`P2_SENT_Get_Version`
Schnittstelle initialisieren	`P2_SENT_Init`
Befehlsverarbeitung	`P2_SENT_Command_Ready`

Ausgabekanäle initialisieren	`P2_SENT_Set_Output_Mode`
	`P2_SENT_Get_Output_Mode`
	`P2_SENT_Config_Output`*
	`P2_SENT_Config_Serial_Messages`*
	`P2_SENT_Invert_Channel`

Kanäle sperren und freigeben	`P2_SENT_Enable_Channel`

Continuous mode:
SENT-Nachricht setzen
	`P2_SENT_Set_Fast_Channel1`
	`P2_SENT_Set_Fast_Channel2`
	`P2_SENT_Set_Reserved_Bits`
	`P2_SENT_Clear_Serial_Message_Array`*
	`P2_SENT_Set_Serial_Message_Pattern`*
	`P2_SENT_Set_Serial_Message_Data`*

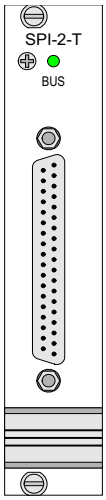
Fifo mode: SENT-Nachrichten einstellen	`P2_SENT_Fifo_Empty` `P2_SENT_Fifo_Clear`* `P2_SENT_Set_Fifo`*
Anzahl Nachrichten abfragen	`P2_SENT_Get_Msg_Counter`
Digitalkanäle	
Digitaleingänge abfragen	`P2_Digin_Long` `P2_Digin_Edge`

Latch-Register der Digitalkanäle nutzen	`P2_Dig_Latch`
	`P2_Dig_Read_Latch`
	`P2_Dig_Write_Latch`
	`P2_Sync_All`

Digitalausgänge setzen und rücklesen	`P2_Digout`, `P2_Digout_Bits`
	`P2_Digout_Long`
	`P2_Get_Digout_Long`
	`P2_Digout_Set`, `P2_Digout_Reset`

Modul-LED einstellen	`P2_Check_LED`, `P2_Set_LED`

5.8.20 Pro II-SPI-2 Rev. E



Das Modul Pro II-SPI-2 Rev. E bietet zwei unabhängige SPI-Schnittstellen mit Master- oder Slave-Funktionalität sowie mehrere Digitalkanäle.

Es gibt das Modul in 2 Varianten:

– ADwin-Pro II-SPI-2-T: SPI-Signale verwenden TTL-Pegel.

– ADwin-Pro II-SPI-2-D: SPI-Signale sind differentiell.

Alternativ kann das Modul auch als Digitalmodul allein mit digitalen Ein-/ Ausgängen genutzt werden (ähnlich Pro II-DIO32-TiCo Rev. E).

Jedes Modul besitzt einen frei programmierbaren *TiCo*-Prozessor, der vollen Zugriff auf die SPI-Schnittstellen hat. Nähere Hinweise zur Anwendung und Programmierung des *TiCo*-Prozessors entnehmen Sie bitte dem Handbuch *TiCoBasic*.

Die Beschreibung ist in folgende Abschnitte gegliedert:

TiCo-Prozessor

Das Modul besitzt einen frei programmierbaren *TiCo*-Prozessor mit 28 KiB Programmpeicher und 28 KiB Datenspeicher. Den *TiCo*-Prozessor programmieren Sie in *TiCoBasic*.

Der *TiCo*-Prozessor hat – wie auch die *ADwin*-CPU – Zugriff auf die SPI-Schnittstellen. Nähere Hinweise zur Anwendung und Programmierung des *TiCo*-Prozessors entnehmen Sie bitte dem Handbuch *TiCoBasic*.

Wenn Sie ein *TiCoBasic*-Programm im *TiCo*-Bootloader ablegen, wird das Programm beim Einschalten der Stromversorgung in den *TiCo*-Prozessor geladen und gestartet. Auf diese Weise kann das Modul eigenständig und unabhängig vom CPU-Modul des *ADwin-Pro II*-Systems arbeiten.

SPI-Betrieb

Die beiden SPI-Schnittstellen können in folgenden Betriebsmodi eingesetzt werden:

– Beide Schnittstellen als SPI-Master

– Beide Schnittstellen als SPI-Slave

– Schnittstelle 1 als SPI-Master, Schnittstelle 2 als SPI-Slave

– Zum Betrieb als Digitalmodul siehe Betrieb mit Digitalsignalen, Seite 228.

Der Betriebsmodus wird per Software mit **`P2_SPI_Mode`** eingestellt.

Eine Schnittstelle ist jeweils über die Leitungen `SCLK`, `SS`, `DATAIN` und `DATAOUT` mit dem SPI-Bus verbunden. Die Pinbelegungen für die verschiedenen Modi sind auf unter Hardware-Aufbau (Seite 229) dargestellt.

Im SPI-Betrieb stellt das Modul mehrere Digitalkanäle zur Verfügung:

– SPI-2-T: je nach Betriebsmodus zwischen 10 und 22 Kanäle mit TTL-Pegeln.

– SPI-2-D: 8 Kanäle mit TTL-Pegeln und 4 differentielle Kanäle.

SPI-Signale

Die SPI-Signale des Moduls ADwin-Pro II-SPI-2-T haben einen 5 V TTL-Pegel, die Signale des Moduls ADwin-Pro II-SPI-2-D sind differentiell.

Die SPI-Schnittstellen arbeiten mit einer Busfrequenz von max. 12,5 MHz.

Es können theoretisch beliebig viele Teilnehmer an den SPI-Bus angeschlossen werden, wobei es immer exakt einen Master geben muss. Der Master legt über die Slave-Select-Leitung(en) fest, mit welchen Slaves er kommunizieren will und erzeugt – solange das Select-Signal aktiv ist – das Taktsignal an SPI Clock (SCLK). Sobald der Master eine Slave-Select-Leitung auf den passenden Pegel zieht, wird der jeweilige Slave aktiv, "lauscht" an SPI MOSI, empfängt die vom Master gesendete Nachricht und legt seine eigene Nachricht - in der Regel noch während der gleichen Datenübertragung - im Takt von SPI CLK an SPI MISO.

Das Taktsignal wird mit den Parametern CPOL (Clock Polarity) und CPHA (Clock Phase) auf einen von 4 Modi festgelegt. CPOL bestimmt das Ruhesignal (Low / High), während CPHA angibt, ob die Daten bei der ersten oder der zweiten Flanke übernommen werden.

Es wird somit eine SPI-Nachricht vom Master zum Slave (MOSI: Master out, Slave in) und eine andere SPI-Nachricht vom Slave zum Master (MISO: Master in, Slave out) transportiert. Der Datenaustausch kann gleichzeitig in beide Richtungen stattfinden.

Die Länge der SPI-Nachrichten ist frei konfigurierbar; eine SPI-Nachricht kann maximal 64 Bit (= 8 Byte) lang sein.

Mit jeder Taktperiode wird ein Bit der SPI-Nachricht übertragen. Beim üblichen Byte-Transfer sind also 8 Taktperioden für eine vollständige Übertragung nötig. Es ist einstellbar, ob die Übertragung mit dem höchstwertigen Bit (MSB) oder mit dem niedrigstwertigen Bit (LSB) beginnt. Eine Übertragung ist beendet, wenn der Master die Slave-Select-Leitung endgültig deaktiviert.

Sie können in das Taktsignal `SCLK` des SPI-Masters Wartezeiten einfügen. Ebenso können Sie beim Einlesen des MISO-Signals eine Zeitverzögerung einstellen.

Grundsätzlich beginnt das Slave-Select-Signal `SS out` des Masters eine bestimmte Zeit vor dem Start des Taktsignals `SCLK` und endet um die gleiche Zeit nach dem Ende des Taktsignals. Diesen Zeitabstand können Sie per Software verlängern.

Schnittstelle SPI-Master

Das Modul Pro II-SPI-2 Rev. E stellt zur Kommunikation als SPI-Master die folgenden Leitungen / Signale bereit. Die Ziffern 1 und 2 beziehen sich auf die beiden Schnittstellen:

Signal	Beschreibung
`SCLK1 out` `SCLK2 out`	Ausgehendes Taktsignal (SPI Clock), das der jeweilige Master erzeugt. Der maximal erzeugbare SPI-Takt ist 12,5 MHz. Der maximal verwendbare Takt ist abhängig von der verwendeten Kabellänge.
`SS1 out` `SS2 out`	Ausgehendes Slave-Select-Signal zur Auswahl der angeschlossenen SPI-Slaves für die aktuell gültige SPI-Nachricht. Die einzelne Slave-Select-Leitung SS out wird im Automatik-Modus genutzt; bei manueller Aktivierung können weitere Slave-Select-Leitungen verwendet werden.
`DATAOUT1` `DATAOUT2`	Datenleitung vom Master zum Slave: Master Out, Slave In (MOSI).
`DATAIN1` `DATAIN2`	Datenleitung vom Slave zum Master: Master In, Slave Out (MISO).

Schnittstelle SPI-Slave

Das Modul Pro II-SPI-2 Rev. E stellt zur Kommunikation als SPI-Slave die folgenden Leitungen / Signale bereit. Die Ziffern 1 und 2 beziehen sich auf die beiden Schnittstellen:

Signal	Beschreibung
`SCLK1 in` `SCLK2 in`	Eingehendes Taktsignal (SPI Clock), das der jeweilige Master erzeugt. Der maximal erzeugbare SPI-Takt ist 12,5 MHz. Der maximal verwendbare Takt ist abhängig von der verwendeten Kabellänge.
`SS1 in` `SS2 in`	Slave-Select-Signal vom SPI-Master zur Aktivierung des SPI-Slaves für die aktuell gültige SPI-Nachricht.
`DATAIN1` `DATAIN2`	Datenleitung vom Master zum Slave: Master Out, Slave In (MOSI).
`DATAOUT1` `DATAOUT2`	Datenleitung vom Slave zum Master: Master In, Slave Out (MISO).

Am Ein- und Ausgang des SPI-Slave werden die Daten jeweils über einen Fifo-Ringspeicher geführt, der 18 Werte zu 32 Bit fasst.

Wenn der Slave eine vom Master gesendete Nachricht über SPI MOSI empfängt, beginnt der Slave noch während der gleichen Datenübertragung, die nächste Nachricht aus dem Ausgangs-Fifo über SPI MISO an den Master zu senden.

Sobald eine eingehende Nachricht vollständig empfangen ist, speichert der SPI-Slave sie im Eingangs-Fifo. Von dort werden die Nachrichten per Software gelesen. Im Fifo können bis zu 18 Nachrichten von 32 Bit Länge zwischengespeichert werden; wenn die SPI-Nachrichtenlänge größer als 32 Bit konfiguriert ist, können nur 9 Nachrichten (in jeweils 2 Fifo-Plätzen) gespeichert werden.

Der Ausgangs-Fifo des SPI-Slaves hat die gleiche Größe wie der Eingangs-Fifo, fasst also 18 Nachrichten von 32 Bit Länge bzw. 9 Nachrichten mit mehr als 32 Bit Länge. Achten Sie darauf, rechtzeitig neue Nachrichten in den Ausgangs-Fifo zu schreiben, damit er nicht leer läuft. Wenn keine Nachrichten mehr im Ausgangs-Fifo enthalten sind, sendet der SPI-Slave die zuletzt gesendete Nachricht so lange, bis wieder Nachrichten in den Ausgangs-Fifo geschrieben werden.

Betrieb mit Digitalsignalen

In allen Betriebsmodi stellt das Modul 8 Digitalkanäle mit TTL-Pegeln (siehe Spezifikation) zur Verfügung. Für die weiteren Kanäle ist zwischen den Modulversionen Pro II-SPI-2-T (TTL-Pegel) und Pro II-SPI-2-D (differentielle Signale) zu unterscheiden, siehe unten.

Im Betriebsmodus Digitalmodul (siehe **`P2_SPI_Mode`**) arbeitet das Modul ganz ohne SPI-Schnittstellen, d.h. alle Kanäle arbeiten als digitale Ein- oder Ausgänge.

Das Modul stellt einen EVENT-Eingang zur Verfügung. Über den Trigger-Eingang EVENT kann ein Signal (Trigger) einen Prozess auslösen, der dann sofort und vollständig abgearbeitet wird (siehe ADbasic-Handbuch).

Pro II-SPI-2-T (TTL-Pegel)

In den SPI-Modi stehen manche Pins, die nicht für SPI-Signale genutzt werden, als zusätzliche digitale Ein- oder Ausgänge zur Verfügung. Bei diesen Pins ist bereits festgelegt, ob sie als Eingang oder als Ausgang arbeiten.

Im Betriebsmodus Digitalmodul stehen 32 digitale Ein- oder Ausgänge mit TTL-Pegeln zur Verfügung (wie beim Digitalmodul Pro II-DIO 32-TiCo). Die Kanäle sind mit **P2_DigProg** in Gruppen zu je 8 als Eingänge oder Ausgänge umschaltbar. Beachten Sie die entsprechende Hardware- und Software-Dokumentation bzw. die Online-Hilfe.

Pro II-SPI-2-D (differentielle Signale)

Im Betriebsmodus Digitalmodul stehen 8 Digitalkanäle mit TTL-Pegeln und 12 differentielle Digitalkanäle zur Verfügung. In den SPI-Modi sind die 8 Digitalkanäle mit TTL-Pegeln sowie 4 differentielle Digitalkanäle weiter verfügbar.

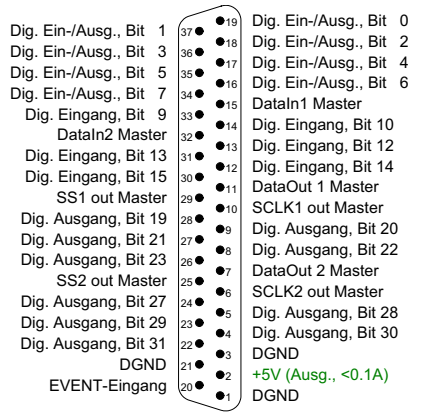
Die Kanäle mit TTL-Pegeln sind mit **P2_DigProg** in Gruppen zu je 4 als Eingänge oder Ausgänge umschaltbar. Die differentiellen Digitalkanäle sind mit **P2_DigProg_Bits** als Eingang oder Ausgang umschaltbar, und zwar jeweils einzeln. Nach dem Einschalten sind alle Kanäle als Eingänge konfiguriert.

Beachten Sie, dass die Befehle für Digitalkanäle beim Modul Pro II-SPI-2-D sowohl für die TTL-Signale als auch für die differentiellen Signale gelten. Beispielsweise setzt **P2_Digout_Long** mit den Bits 0…11 differentielle Signale und mit den Bits 16…19 / 24…27 TTL-Signale (soweit sie als Ausgänge konfiguriert sind). Die Bits 12…15, 20…23 und 28…31 sind nicht belegt.

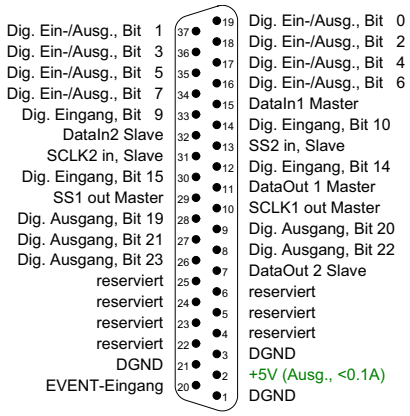
Hardware-Aufbau

Die Pinbelegung hängt vom Betriebsmodus des Moduls ab (**P2_SPI_Mode**).

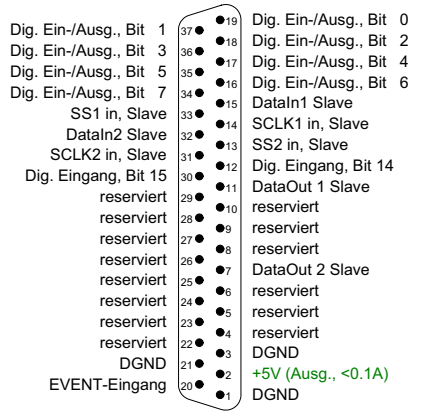
Wenn Sie Probleme mit der Abschirmung von ungenutzten DataIn/DataOut-Leitungen haben, kann es helfen, diese Pins mit logisch 0 oder DGND zu belegen.



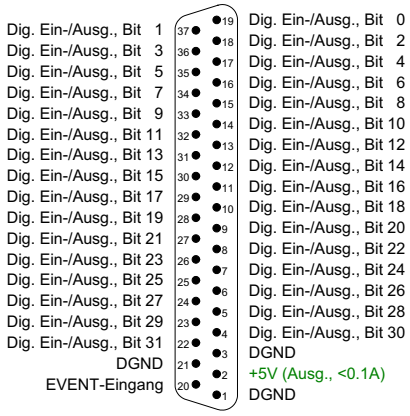
SPI-2-T: Master 1 + Master 2



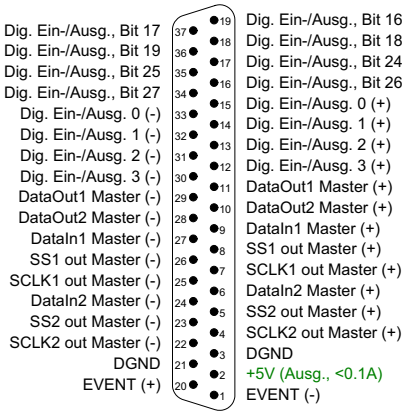
SPI-2-T: Master 1 + Slave 2



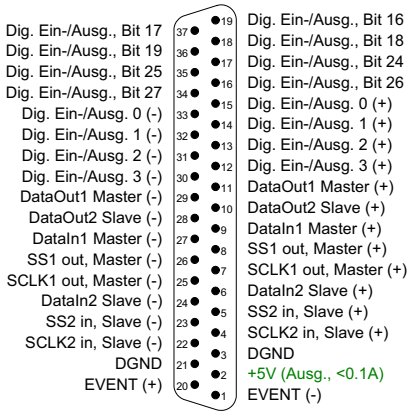
SPI-2-T: Slave 1 + Slave 2



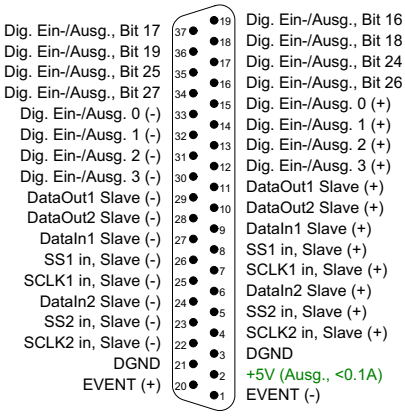
SPI-2-T: Digital



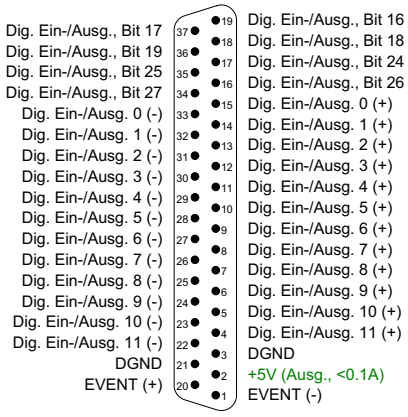
SPI-2-D: Master 1 + Master 2



SPI-2-D: Master 1 + Slave 2



SPI-2-D: Slave 1 + Slave 2



SPI-2-D: Digital

Spezifikation

Digitale Ein-/Ausgänge	Pro II-SPI-2-T: TTL-Logik (5V) Pro II-SPI-2-D: differentielle Signale, teilweise TTL-Logik (5V)
Eingangswiderstand	TTL-Eingänge: 10 kΩ, pull-down Diff. Eingänge:120 Ω zwischen (+) und (-) sowie 1,5 kΩ zu +5V und zu Masse
V_{IH} min.	2 V
V_{IL} max.	0,8 V
I_{IH} max.	1 µA
Spannungsbereich	0 V … +5 V
Ausgangsstrom	max. ±35 mA pro Kanal, max. ±70 mA je SPI-Schnittstelle
Steckerverbindung	37-polige Sub-D-Buchse

Modul-Revisionen

Die Unterschiede der Hardware-Revisionen sind nachfolgend dargestellt. Die Revisionsbezeichnung befindet sich auf der Frontseite.

Revision	Ausgabe-datum	Änderungen zur Vorgänger-Version
E01	01 / 2013	Erst-Version

Unterschiedliche Revisions-Buchstaben bedeuten unterschiedliche Modul-Eigenschaften und sind separat dokumentiert. Der Revisionsbezeichnung angehängt ist eine Zählnummer für interne Zwecke.

Programmierung

Programmierung in ADbasic

Das Modul wird komfortabel mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch Pro II-Software und in der Online-Hilfe *ADbasic* erläutert.

Die Include-Datei `ADwinPro_All.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle

Schnittstelle konfigurieren	`P2_SPI_Mode`
	`P2_SPI_Config`
	`P2_SPI_Master_Config`
	`P2_SPI_Slave_Config`

SPI-Master: spezielle Konfiguration	`P2_SPI_Master_Set_Clk_Wait`
SPI-Master: Ausgabedaten bereitstellen	`P2_SPI_Master_Set_Value32` `P2_SPI_Master_Set_Value64`

SPI-Master:
Slave-Select setzen
Datenübertragung starten
	`P2_Digout, P2_Digout_Bits`
	`P2_Digout_Long`
	`P2_SPI_Master_Start`

SPI:Master: Status lesen	`P2_SPI_Master_Status`

SPI-Master:
Nachricht lesen
	`P2_SPI_Master_Get_Value32`
	`P2_SPI_Master_Get_Value64`
	`P2_SPI_Master_Get_Static_Input`

SPI-Slave:
Eingangs-/Ausgangs-Fifo verwenden
	`P2_SPI_Slave_OutFifo_Write`
	`P2_SPI_Slave_OutFifo_Empty`
	`P2_SPI_Slave_InFifo_Full`
	`P2_SPI_Slave_InFifo_Read`
	`P2_SPI_Slave_Clear_Fifo`

TTL-Digitalkanäle konfigurieren	`P2_DigProg`
Digitalkanäle mit differentiellen Signalen konfigurieren	`P2_DigProg_Bits` (nur Pro II-SPI-2-D)
Eingangssignale abfragen	`P2_Digin_Long`
Latch-Register nutzen	`P2_Dig_Latch`, `P2_Dig_Read_Latch` `P2_Dig_Write_Latch`, `P2_Sync_All`

Flanken an Eingangskanälen überwachen	`P2_Dig_Fifo_Mode`
	`P2_Digin_Fifo_Enable`
	`P2_Digin_Fifo_Read`
	`P2_Digin_Fifo_Read_Fast`
	`P2_Digin_Fifo_Read_Timer`
	`P2_Digin_Fifo_Clear`
	`P2_Digin_Fifo_Full`

Flankenstatus abfragen	`P2_Digin_Edge`

Ausgangssignale setzen und rücklesen	`P2_Digout`, `P2_Digout_Bits`
	`P2_Digout_Long`
	`P2_Digout_Reset`
	`P2_Digout_Set`
	`P2_Get_Digout_Long`

Bereich	Befehle
Ausgangssignale automatisch setzen	P2_Digout_Fifo_Clear P2_Digout_Fifo_Empty P2_Digout_Fifo_Enable P2_Digout_Fifo_Read_Timer P2_Digout_Fifo_Start P2_Digout_Fifo_Write
Modul-LED einstellen	P2_Check_LED, P2_Set_LED
Interrupts und Event-Eingang einstellen	P2_Event_Enable, P2_Event_Config P2_Event_Read

Programmierung in TiCoBasic

Das Modul kann mit *TiCoBasic*-Befehlen programmiert werden. Die Befehle sind in der Online-Hilfe *TiCoBasic* erläutert.

Die Include-Datei `SPI_TiCo.inc` enthält Befehle für folgende Bereiche:

Bereich	Befehle
Schnittstelle konfigurieren	SPI_Mode SPI_Config SPI_Master_Config SPI_Slave_Config
SPI-Master: spezielle Konfiguration	SPI_Master_Set_Clk_Wait
SPI-Master: Ausgabedaten bereitstellen	SPI_Master_Set_Value32 SPI_Master_Set_Value64
SPI-Master: Slave-Select setzen Datenübertragung starten	Digout, Digout_Bits Digout_Long SPI_Master_Start
SPI-Master: Status lesen	SPI_Master_Status
SPI-Master: Nachricht lesen	SPI_Master_Get_Value32 SPI_Master_Get_Value64 SPI_Master_Get_Static_Input
SPI-Slave: Eingangs-/Ausgangs-Fifo verwenden	SPI_Slave_OutFifo_Write SPI_Slave_OutFifo_Empty SPI_Slave_InFifo_Full SPI_Slave_InFifo_Read SPI_Slave_Clear_Fifo
TTL-Digitalkanäle konfigurieren	DigProg
Digitalkanäle mit differentiellen Signalen konfigurieren	DigProg_Bits (nur Pro II-SPI-2-D)
Eingangssignale abfragen	Digin_Long
Latch-Register nutzen	Dig_Latch, Dig_Read_Latch Dig_Write_Latch
Flanken an Eingangskanälen überwachen	Dig_Fifo_Mode Digin_Fifo_Enable Digin_Fifo_Read Digin_Fifo_Read_Timer Digin_Fifo_Clear Digin_Fifo_Full
Flankenstatus abfragen	Digin_Edge

Bereich	Befehle
Ausgangssignale setzen und rücklesen	Digout, Digout_Bits Digout_Long Digout_Reset Digout_Set Get_Digout_Long
Ausgangssignale automatisch setzen	Digout_Fifo_Clear Digout_Fifo_Empty Digout_Fifo_Enable Digout_Fifo_Read_Timer Digout_Fifo_Start Digout_Fifo_Write
Modul-LED einstellen	Check_LED, Set_LED
Interrupts und Event-Eingang einstellen	Event_Enable, Event_Config Trigger_Event

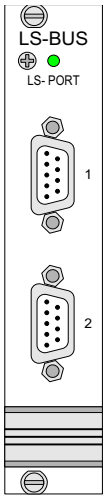
Programmierung TiCo-Zugriff

Für den Zugriff auf den *TiCo*-Prozessor von der ADwin CPU sind die folgenden *ADbasic*-Befehle in der Datei `ADwinPro_All.inc` definiert. Die Befehle sind im Handbuch *TiCoBasic* und in der Online-Hilfe *ADbasic* erläutert.

Bereich	Befehle
Datenaustausch mit dem *TiCo*-Prozessor über globale Variablen	P2_TDrv_Init P2_GetData_Long, P2_Get_Par, P2_Get_Par_Block P2_SetData_Long, P2_Set_Par, P2_Set_Par_Block P2_Get_TiCo_RingBuffer, P2_Set_TiCo_RingBuffer P2_RingBuffer_Empty P2_RingBuffer_Full
TiCo-Prozessor steuern	P2_TiCo_Reset, P2_TiCo_Start, P2_TiCo_Stop P2_Get_TiCo_Bootloader_Status P2_Get_TiCo_Status, P2_Workload
TiCo-Prozesse steuern	P2_Process_Status P2_TiCo_Get_Processdelay P2_TiCo_Set_Processdelay P2_TiCo_Start_Process P2_Tico_Stop_Process
TiCo-Programme übertragen	P2_TiCo_Flash, P2_TiCo_Load

5.8.21 Pro II-LS-2 Rev. E

Das Modul Pro II-LS-2 Rev. E stellt 2 Schnittstellen für den LS-Bus auf 9-poligen D-Sub-Verbindern (Buchse) zur Verfügung.



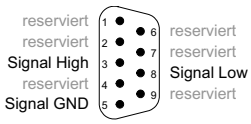


Abb. 182 – Pro II-LS-2 Rev. E: Pinbelegung

Der LS-Bus (Low Speed) ist ein bidirektionaler, serieller Bus mit 5 MHz Taktrate. Der Bus ist eine Eigenentwicklung für den Anschluss externer Module. Es steht der Modultyp HSM-24V zur Verfügung, mit dem 24 V-Signale auf 32 digitalen Kanälen verarbeitet werden können.

Der Bus ist als Linienverbindung aufgebaut, d. h. die *ADwin*-Schnittstelle und bis zu 15 LS-Bus-Module sind jeweils über Zweipunktverbindungen miteinander verbunden. Am letzten LS-Bus-Modul muss der Busabschluss aktiviert sein. Die maximale Buslänge beträgt 5 m.

Die Module am LS-Bus werden mit *ADbasic*-Befehlen programmiert, die über die LS-Bus-Schnittstelle am *ADwin*-System geschickt werden. Die Befehle für die Module am LS-Bus sind im Handbuch des LS-Bus-Moduls beschrieben und in der Online-Hilfe.

TiCo-Prozessor

Das Modul besitzt zusätzlich einen frei programmierbaren *TiCo*-Prozessor mit 28 KiByte Datenspeicher und 28 KiByte Programmspeicher, der Zugriff auf alle Kanäle des Moduls hat. Nähere Hinweise zur Anwendung und Programmierung des *TiCo*-Prozessors entnehmen Sie bitte dem Handbuch *TiCoBasic*.

Wenn Sie ein *TiCoBasic*-Programm im *TiCo*-Bootloader ablegen, wird das Programm beim Einschalten der Stromversorgung in den *TiCo*-Prozessor geladen und gestartet. Auf diese Weise kann das Modul eigenständig und unabhängig vom CPU-Modul des *ADwin-Pro II*-Systems arbeiten.

Programmierung in ADbasic

Am Modul selbst kann nur die LED programmiert werden mit den Befehlen `P2_Check_LED`, `P2_Set_LED`.

Die LS-Bus-Module werden mit *ADbasic*-Befehlen programmiert. Die Befehle sind im Handbuch HSM-24V und in der Online-Hilfe *ADbasic* erläutert.

Die Include-Datei `ADwinPro_All.inc` enthält folgende LS-Bus-Befehle:

Bereich	Befehle
Initialisierung eines Moduls am LS-Bus.	`P2_LS_DIO_Init`
Digitale Kanäle des Moduls HSM-24V als Ein- der Ausgänge konfigurieren.	`P2_LS_DigProg`
Digitalkanäle am Modul HSM-24V abfragen.	`P2_LS_Dig_IO`, `P2_LS_Digin_Long`
Wert ausgeben	`P2_LS_Digout_Long`, `P2_LS_Get_Output_Status`
Watchdog am LS-Bus initialisieren.	`P2_LS_Watchdog_Init`, `P2_LS_Watchdog_Reset`

Programmierung in TiCoBasic

Am Modul selbst kann nur die LED programmiert werden mit den Befehlen **Check_LED**, **Set_LED**.

Die LS-Bus-Module werden mit *TiCoBasic*-Befehlen programmiert. Die Befehle sind in der Online-Hilfe *TiCoBasic* erläutert.

Die Include-Datei LSbus_TiCo.inc enthält folgende LS-Bus-Befehle:

Bereich	Befehle
Initialisierung eines Moduls am LS-Bus.	**LS_DIO_Init**
Digitale Kanäle des Moduls HSM-24V als Ein- der Ausgänge konfigurieren.	**LS_DigProg**
Digitalkanäle am Modul HSM-24V abfragen.	**LS_Dig_IO**, **LS_Digin_Long**
Wert ausgeben	**LS_Digout_Long**, **LS_Get_Output_Status**
Watchdog am LS-Bus initialisieren.	**LS_Watchdog_Init**, **LS_Watchdog_Reset**

Programmierung TiCo-Zugriff

Für den Zugriff auf den *TiCo*-Prozessor von der ADwin CPU sind die folgenden *ADbasic*-Befehle in der Datei ADwinPro_All.inc definiert. Die Befehle sind im Handbuch *TiCoBasic* und in der Online-Hilfe *ADbasic* erläutert.

Bereich	Befehle
Datenaustausch mit dem *TiCo*-Prozessor über globale Variablen	**P2_TDrv_Init** **P2_GetData_Long**, **P2_Get_Par**, **P2_Get_Par_Block** **P2_SetData_Long**, **P2_Set_Par**, **P2_Set_Par_Block** **P2_Get_TiCo_RingBuffer**, **P2_Set_TiCo_RingBuffer** **P2_RingBuffer_Empty** **P2_RingBuffer_Full**
TiCo-Prozessor steuern	**P2_TiCo_Reset**, **P2_TiCo_Start**, **P2_TiCo_Stop** **P2_Get_TiCo_Bootloader_Status** **P2_Get_TiCo_Status**, **P2_Workload**
TiCo-Prozesse steuern	**P2_Process_Status** **P2_TiCo_Get_Processdelay** **P2_TiCo_Set_Processdelay** **P2_TiCo_Start_Process** **P2_Tico_Stop_Process**
TiCo-Programme übertragen	**P2_TiCo_Flash**, **P2_TiCo_Load**

6 Kalibrierung

6.1 Allgemeine Hinweise

Die Digital/Analog- (DAC) und Analog/Digitalwandler (ADC) der ***ADwin***-Systeme sind werkseitig kalibriert. Entsprechend der Vorschriften zur Einhaltung der Messgenauigkeit für Ihr Anwendungsgebiet sind die Geräte in regelmäßigen Abständen zu kalibrieren.

Einschränkung der Anwendergruppe

Der Hersteller des in dieser Dokumentation beschriebenen Systems geht davon aus, dass an dem Gerät nur qualifiziertes Personal arbeitet.

> *Qualifiziertes Personal sind Personen, die aufgrund ihrer Ausbildung, Erfahrung und Unterweisung sowie ihrer Kenntnisse über einschlägige Normen, Bestimmungen, Unfallverhütungsvorschriften und Betriebsverhältnisse von dem für die Sicherheit der Anlage Verantwortlichen berechtigt worden sind, die jeweils erforderlichen Tätigkeiten auszuführen und die dabei mögliche Gefahren erkennen und vermeiden können. (Definition für Fachkräfte nach VDE 105 und IEC 60364)*.

Verfügbarkeit der Unterlagen

Diese Produktdokumentation und Unterlagen, auf die verwiesen wird, müssen stets verfügbar sein und konsequent beachtet werden. Für Schäden, die durch Missachtung der Informationen in dieser bzw. der weiterführenden Dokumentation entstehen, übernimmt die Firma *Jäger Computergesteuerte Messtechnik GmbH*, Lorsch, keine Haftung.

Hilfsmittel

Zur Kalibrierung benötigen Sie folgende Hilfsmittel:

- Eine Referenzspannungsquelle mit einer Genauigkeit von
  - 10 µV bei 18 Bit Wandlern
  - 30 µV bei 16 Bit Wandlern
  - 100 µV bei 14 Bit Wandlern
- Ein Digital-Multimeter mit einer Genauigkeit von
  - 10 µV bei 18 Bit Wandlern
  - 30 µV bei 16 Bit Wandlern
  - 100 µV bei 14 Bit Wandlern
- Verbindungskabel von den Ein/Ausgängen zur Referenzspannungsquelle und zum Messgerät

Warnung: Gefahr des elektrischen Schlags.

ADwin-Pro II-Systeme verfügen über ein Netzteil, das bei geöffnetem Gerät Zugang zu hochspannungsführenden Leitungen bzw. Anschlüssen ermöglicht. Die Lüftungsschlitze lassen die Durchführung eines Abgleichbestecks mit einem Durchmesser von 2,5 mm zu.

Kalibrieren Sie nur bei geschlossenem Gerät!
Führen Sie keine stromleitenden Objekte durch die Lüftungsschlitze!

6.2 Berechnungsgrundlagen

Spannungsbereich

Die ***ADwin***-Systeme arbeiten bei den analogen Ein- und Ausgängen in der Standardeinstellung mit einem Spannungsbereich von -10V…+10V (bipolar ±10V).

Zuordnung von Digits zu Spannung

Die 65536 (2^{16}) Digits sind den jeweiligen Spannungsbereichen der ADC und DAC so zugeordnet, dass der Wert für

- 0 (Null) Digit der maximalen negativen Spannung
- 65535 Digit der maximalen positiven Spannung entspricht.

Der Wert für 65536 Digit, genau 10 Volt, liegt damit gerade außerhalb des Messbereiches, womit sich für die 16 Bit AD- bzw. DA-Wandlung ein maximaler Spannungswert von 9,999695 Volt und für die 18 Bit AD-Wandlung von 9,999923706 Volt ergibt.

Nullpunktverschiebung

In der Einstellung bipolar ±10V entsteht damit eine Nullpunktverschiebung, die im folgenden auch als Offset bezeichnet wird. Die Verschiebung beträgt U_{OFF}=-10V.

Least Significant Bit U_{LSB}

Die Quantisierungsstufe (U_{LSB}) gibt die Spannung des niederwertigsten Bit an (Least Significant Bit). In der Standardeinstellung entspricht

- eine Quantisierungsstufe bei einem 18 Bit-Wandler dem 2^{18}-ten Teil von 20V gleich 76,294µV.
- eine Quantisierungsstufe bei einem 16 Bit-Wandler dem 2^{16}-ten Teil von 20V gleich 305,175µV.
- eine Quantisierungsstufe bei einem 14 Bit-Wandler dem 2^{14}-ten Teil von 20V gleich 1220,7µV.

Verstärkung k_V

Bei Eingangsmodulen mit einem programmierbaren Verstärker (PGA) können Sie die Eingangsspannung um die Faktoren 2, 4, und 8 verstärken. Damit verkleinert sich der Messbereich um den jeweiligen Verstärkungsfaktor k_V.

Beachten Sie bei Anwendungen mit k_V >1, dass auch die Störsignale entsprechend mit verstärkt werden. Diese können Sie mit der Programmierung von digitalen Filtern im ***ADbasic*** vermindern.

Zuordnung der Bits

Um bei Messungen mit einem 14 Bit-ADC und einem16 Bit-ADC dieselbe Zuordnung der Bits zu erreichen, wird der gewandelte Wert beim 14 Bit-ADC linksbündig in einem Wort (16 Bit) zurückgeliefert, wobei die untersten 2 Bits stets 0 sind (siehe Abb. 183).

Ein 18 Bit-Messwert wird als 24 Bit-Wert zurückgegeben, d.h. der Wert wird um 6 Bits nach links verschoben, die Bits 0…5 sind immer Null.

Bit-Nr.	31…24	23…16	15…6	5…2	1…0
Inhalt	0	18-Bit Messwert in den Bits 6…23		0	0
	0	0	16 Bit-Messwert in den Bits 0…15		
	0	0	14 Bit-Messwert in den Bits 2…15		0
	oberes Wort		unteres Wort		

Abb. 183 – Bit-Zuordnung bei verschiedenen Auflösungen

Die 16384 Digits des 14 Bit-ADC werden auf die 65535 Digits des 16 Bit-ADC abgebildet. Damit entsprechen 4 Digits des 16 Bit-ADC einem Digit des 14 Bit-ADC.

DAC

Für einen DAC gilt:

$$U_{OUT} = Digits \cdot U_{LSB} + U_{OFF}$$

$$Digits = \frac{U_{OUT} - U_{OFF}}{U_{LSB}}$$

ADC

Für einen ADC gilt:

$$Digits = \frac{k_V \cdot U_{IN} - U_{OFF}}{U_{LSB}}$$

$$U_{IN} = \frac{Digits \cdot U_{LSB} + U_{OFF}}{k_V}$$

Toleranzbereiche

Geringe Abweichungen zu den rechnerischen Werten können innerhalb der Toleranzbereiche einzelner Bauteile liegen. Es gibt 2 charakteristische Abweichungsarten, die in diesem Handbuch (in LSB) angegeben sind:

INL

– Die integrale Nicht-Linearität (INL) beschreibt die maximale Abweichung von der Geraden über den gesamten Eingangsspannungsbereich.

DNL

– Die differentielle Nicht-Linearität (DNL) beschreibt die maximale Abweichung von der Breite einer Quantisierungsstufe.

6.3 Kalibrieren

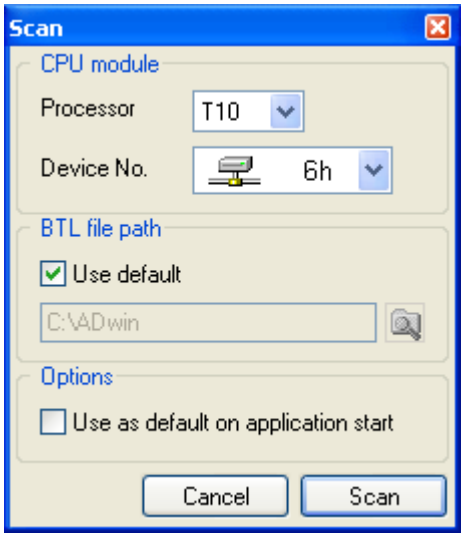
Vorbereitung der Hardware

Legen Sie den Spannungsbereich des Moduls fest.

Die Kalibrierung muss bei Betriebstemperatur erfolgen. Bei einer Temperatur des Gerätes von ca. 20…25 Grad Celsius (Raumtemperatur) ist die Betriebstemperatur etwa 30 Minuten nach dem Einschalten erreicht.

Beachten Sie bitte die allgemeinen Hinweise in Kapitel 6.1.

Rufen Sie das Programm `ADpro.exe` aus dem Windows-Startmenü auf unter `Programme\ADwin`. Es erscheint das Einstellungsfenster `Scan`.



Wenn beim Programmstart eine Fehlermeldung auftritt, prüfen Sie bitte, ob auf Ihrem Rechner das Programmpaket `.NET Framework 2.0` installiert ist.

Beachten Sie: Mit dem nächsten Schritt stoppen Sie alle Prozesse und setzen alle Moduleinstellungen zurück!

Stellen Sie die Daten für das zu kalibrierende *ADwin*-System ein. Mit der Schaltfläche `Scan` wird eine Verbindung zum *ADwin*-System aufgebaut und

Systeminformationen werden gelesen. Hierbei initialisiert das Programm `ADpro.exe` das *ADwin*-System, d.h. es beendet und löscht noch laufende Prozesse.

Wenn Ihr *ADwin*-System korrekt initialisiert ist, erscheint das Fenster `ADpro`.

Sie erhalten eine Fehlermeldung, wenn

– Ihr ***ADwin***-System nicht erfolgreich gebootet wurde.

– die Angaben im Fenster `Scan` nicht zutreffen.

  Überprüfen und – falls erforderlich – korrigieren Sie die Angaben im Fenster. Starten Sie Sie Ihr System nun nochmals mit der Schaltfläche „`Scan`“.

Markieren Sie das zu kalibrierende Modul im Fenster `ADpro` und wählen Sie den Menüeintrag `Calibration` im Menü `Module`. Wenn der Menüeintrag `Calibration` fehlt, kann das gewählte Modul nicht kalibriert werden.

Kalibrieren

Schließen Sie das Messgerät und die Referenzspannungsquelle am Modul an.

Folgen Sie nun den Anweisungen, die im Eingabefenster auf englisch angezeigt werden. Beachten Sie bitte den Unterschied zwischen analogen Eingangs-Modulen mit und ohne Multiplexer:

– Analoge Eingangs-Module mit Multiplexer (AIn-…): Die Kalibrierung des ADC erfolgt über den Eingangskanal 1.

– Analoge Eingangs-Module ohne Multiplexer (AIn-F-…): Der jeweils angeschlossene Kanal wird im Fenster „`Input channel`“ angewählt.

7 Zubehör

7.1 LEMO-Kabelsätze für *ADwin-Pro*-Systeme

1-polig

Pro-CS-1	4 x Kabel 200 mm (7,8 inch) und 4 x Kabel 400 mm (15,7 inch)
Pro-CS-2	4 x Kabel 400 mm (15,7 inch) und 4 x Kabel 800 mm (31,5 inch)
Pro-CS-3	4 x Kabel 1000 mm (39,4 inch) und 4 x Kabel 1500 mm (59 inch)
Pro-CS-4	4 x Kabel 5000 mm (196,8 inch)
Pro-CS-5	8 x Kabel 400 mm (15,7 inch)
Pro-CS-6	8 x Kabel 1000 mm (39,4 inch)
Pro-CS-7	8 x Kabel 2000 mm (78,7 inch)

Alle Kabel haben LEMO-Stecker an beiden Enden.

2-polig

Pro-CS-8	4 x Kabel 2000 mm (78,7 inch): LEMO-Stecker 2-polig - Kabel - offenes Ende
Pro-CS-9	4 x Kabel 1000 mm (39,4 inch) und 4 x LEMO-Buchse (lose) für Frontplattenmontage
Pro-CS-10	4 x Kabel 500 mm (19,7 inch) und 4 x LEMO-Buchse (lose) für Frontplattenmontage
Pro-CS-11	4 x Kabel 2000 mm (78,7 inch) und 4 x LEMO-Buchse (lose) für Frontplattenmontage

Soweit nicht anders angegeben, haben alle Kabel 2-polige LEMO-Stecker an beiden Enden.

7.2 LEMO-Adaptersätze

Pro-AS-1	4 Adapter: LEMO-Buchsen auf BNC-Stecker
Pro-AS-3	4 LEMO-Buchsen T-Stück (1 x Stecker, 2 x Buchse)
Pro-AS-4	4 Adapter: LEMO-Buchse - LEMO-Buchse
Pro-AS-5	4 LEMO-Buchse mit 50 Ohm-Abschluss
Pro-AS-6	4 Adapter, Länge: 150 mm: LEMO-Stecker - Kabel - BNC-Buchse
Pro-AS-7	4 Adapter, Länge: 1000 mm: LEMO-Stecker - Kabel - BNC-Buchse
Pro-AS-8	4 Adapter, Länge: 2000 mm: LEMO-Stecker - Kabel - BNC-Buchse
Pro-AS-9	4 Adapter, Länge: 1000 mm: LEMO-Stecker - Kabel - BNC-Stecker
Pro-AS-10	4 Adapter, Länge: 2000 mm: LEMO-Stecker - Kabel - BNC-Stecker

7.3 Kabel / Klemmblöcke für OPT-16 und TRA-16

ADwin-Cable-1	Verlängerungskabel 1000 mm geschirmt, für 37-polige DSub-Anschlüsse; eine Seite Buchse, eine Seite Stecker
ADwin-Cable-2	Verlängerungskabel 500 mm geschirmt, für 37-polige DSub-Anschlüsse; eine Seite Buchse, eine Seite Stecker
ADwin-Cable-3	Verlängerungskabel 250 mm geschirmt, für 37-polige DSub-Anschlüsse; eine Seite Buchse, eine Seite Stecker
ADwin-AT-37M	Klemmleistenblock für 37-polige D-Sub-Stecker

7.4 Gehäuse-Befestigung

Pro-Mount	Befestigungsblech (mit den Gehäusegriffen zu verschrauben) zum Befestigen eines Pro I-Gehäuses z.B. in einem 19“-Schrank.
Pro II-Mount	Befestigungswinkel (an der Gehäuseseite zu montieren) zum Befestigen eines Pro II-Gehäuses z.B. in einem 19“-Schrank.

7.5 Bezugsadressen

7.5.1 LEMO-Stecker

Pro-Module sind mit folgenden LEMO-Steckverbindern ausgerüstet:

– Buchsen/Stecker der Serie 00 NIM-CAMAC, 1-polig
  - Kabelstecker: Bauform FFS (straight cable plug)
  - Einbaubuchse: Bauform ERN

– Buchsen/Stecker der Serie 00 Multikontakt, 2-polig
  - Kabelstecker: Bauform FGG
  - Einbaubuchse: Bauform EGG

– Modul Pt100/RTD-8: Buchsen/Stecker der Serie 0B
  - Kabelstecker: Bauform FGG
  - Einbaubuchse: Bauform EGG

Hersteller der LEMO-Steckverbinder:

LEMO GmbH
Hanns-Schwindt-Straße 6
Postfach 820529
D-81829 München

Tel.: +49 89 42770-3
Fax: +49 89 4202192
E-Mail: lemo@info.de
Internet: www.lemo.com

7.5.2 Stromversorgungs-Stecker Pro-Mini

Der Anschlussstecker für die externe Stromversorgung des Gehäuses Pro-Mini stammt von Phoenix Contact GmbH:

Combicon-Steckerteil, Raster 5,0 mm, Typ MSTB 2,5/ 3-STF;
Artikelnr. 1786844 (Stand Dez. 2005)

Hersteller des Steckers:

Phoenix Contact GmbH & Co. KG
Flachsmarktstraße 8
D-32825 Blomberg

Tel.: +49 5235 300
Fax: +49 5235 341200
E-Mail: info@phoenixcontact.com
Internet: www.phoenixcontact.de

Anhang

A.1 RoHS Konformitätserklärung

Die RoHS-Richtlinie 2011/65/EU der Europäischen Union zur Beschränkung und Verwendung gefährlicher Stoffe in elektrischen und elektronischen Geräten ist am 3. Januar 2013 in Kraft getreten. Die Vorgaben wurden durch die Richtlinie 2015/863/EU ergänzt.

Die Richtlinie betrifft folgende Substanzen:

- Blei (Pb)
- Cadmium (Cd)
- Hexavalentes Chrom (Cr VI)
- Polybromierte Biphenyle (PBB)
- Polybromierte Diphenylether (PBDE)
- Quecksilber (Hg)
- Bis(2-ethylhexyl)phthalat (DEHP)
- Benzylbutylphthalat (BBP)
- Dibutylphthalat (DBP)
- Diisobutylphthalat (DIBP)

Die Produktlinie *ADwin-Pro II* erfüllt die Voraussetzungen der RoHS-Richtlinie in allen gelieferten Varianten.

A.2 Liste der Module

Die Module sind nach Seitenzahl aufgeführt.