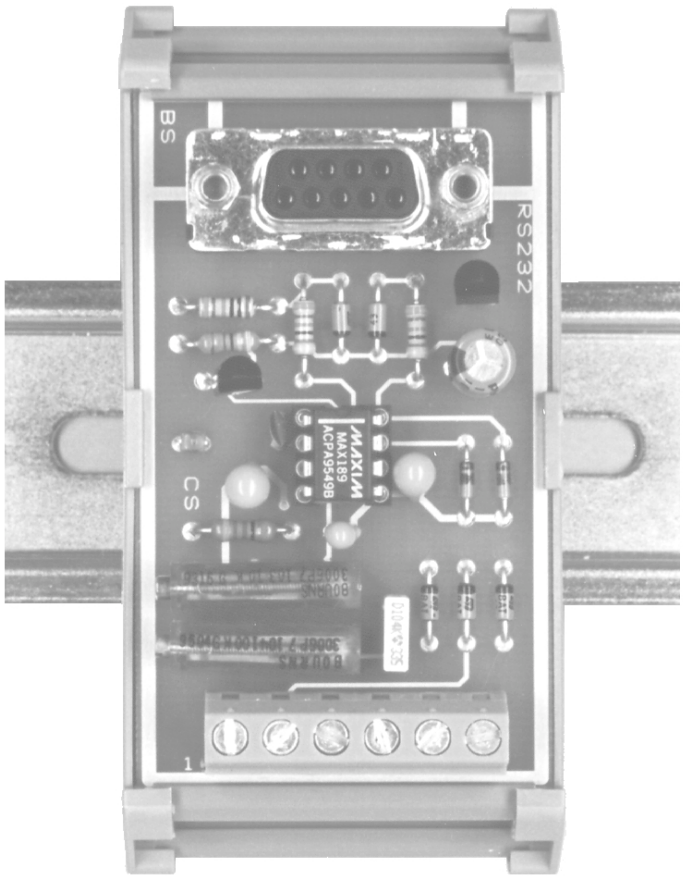
MAX 12

A/D-Wandler-Modul für 10 V= mit 12 bit Auflösung

12-bit A/D-Wandler mit RS 232 Schnittstelle

zur Montage in Schaltschränken



Industrie-Datenerfassung mit dem PC

KOLTER ELECTRONIC

Tel.: 02235-76707 Fax.: 02235-72048 e-mail: service@pci-card.com Internet: www.pci-card.com

Inhaltsverzeichnis

Willkommen

Sehr geehrter Kunde,
wir bedanken uns für das Interesse oder den Kauf des MAX-12 12-Bit A/D RS232 Modul.

Mit diesem Modul haben Sie ein Produkt erworben, welches nach dem heutigen Stand der Technik gebaut wurde. Dieses Produkt erfüllt die Anforderungen der geltenden europäischen und nationalen Richtlinien. Die EMV-Konformität wurde nachgewiesen, die entsprechenden Erklärungen und Unterlagen sind beim Hersteller hinterlegt. Um diesen Zustand zu erhalten und einen gefahrlosen Betrieb sicherzustellen müssen Sie als Anwender diese Betriebsanleitung sowie weitere Sicherheitsdokumente s.u. beachten.

Bei technischen Fragen wenden Sie sich bitte an unsere Technische Beratung. Rufnummern und Adressen finden Sie dazu unten auf dem Titelblatt und/oder hinten im Anhang.

Diese Bedienungsanleitung gehört zu diesem Produkt. Sie enthält wichtige Hinweise zur Inbetriebnahme und Handhabung bei der Installation. Achten Sie hierauf, auch wenn Sie dieses Produkt an Dritte weitergeben. Das Produkt hat den Hersteller in sicherheitstechnisch einwandfreiem Zustand verlassen. Um diesen Zustand zu erhalten und einen gefahrlosen Betrieb sicherzustellen, muß der Anwender alle Sicherheitshinweise und Warnvermerke beachten, die in dieser Gebrauchsanweisung enthalten sind. Ggf. müssen weitere Hinweise beachtet werden, die Sie jedoch nur online von unserer Webseite herunterladen können. Beipielsweise haben wir eine FAQ-Seite eingerichtet, um wiederkehrende Fragen ausführlich zu beantworten, die diese Betriebsanleitung vom Umfang her sicher sprengen würde.

Achtung:
Eine andere Verwendung als die beschriebene führt zur Beschädigung dieses Produktes, darüber hinaus ist dies mit Gefahren, wie z.B. Kurzschluß, Brand, elektrischer Schlag etc. verbunden. Das gesamte Produkt darf nicht geändert bzw. umgebaut und die Gehäuse nicht geöffnet werden. Die nachfolgenden Sicherheits- und Gefahrenhinweise ergeben sich zu diesem Produkt in der Form, dass der Einbau in/an einem Industrie-PC in industrieller Umgebung als Anlage erfolgt. Somit sind möglicherweise auch übergeordnete Sicherheits- und Gefahrenhinweise relevant, die unser Produkt zwar nicht unmittelbar betreffen, jedoch in ihrer Gesamtheit als industrielle Anlage beachtet werden müssen. Der Einbau, sowie die Inbetriebnahme darf daher nur durch geschultes Fachpersonal, oder durch einen ausgebildeten Techniker erfolgen. Aus Gründen der ständigen Gesetzesänderungen und EU-Richtliniennovellen haben wir uns entschlossen, diese Hinweise als Zusammenfassung in einem separaten Dokument halbjährlich zu aktualisieren und online zu stellen.

Die aktuellen Sicherheits- und Gefahrenhinweise finden Sie auf unserer Webseite unter:

http://www.pci-card.com/SiGef-Hinweise.PDF

Vielen Dank.

Allgemeines zu I/O-Modulen

Wenn ein PC zeitlich festgelegte Abläufe innerhalb einer Produktion steuern oder komplexe Prozesse regeln soll, muß man ihn zuerst in die Lage versetzen, die nötigen analogen oder digitalen Meßsignale aufnehmen und ausgeben zu können. Dazu verwendet man am besten eine möglichst exakt auf die jeweilige Aufgabenstellung zugeschnittene Peripherikarte, auf der alle nötigen Ein- und Ausgänge vorhanden sind und mit der auch noch gleich die Pegel anpaßt werden.

Da man, angesichts der Menge der zu automatisierenden Abläufe, diese Karte in der Praxis kaum finden wird, bietet sich als zweitbeste Lösung die Verwendung mehrerer Karten an, die jeweils einen Teilbereich der Aufgabenstellung abdecken.

Häufig werden beispielsweise TTL-I/O-Karten genutzt, die oft viele Signale ein- und ausgeben können, aber nur solche, die im TTL-Pegelbereich von 0...5 V angesiedelt sind. Oder es kommen Timer-Karten zum Einsatz, wenn Taktzeiten leicht zu verändern, aber präzise einstellbar sein müssen.

Optokoppler- und Relais-Karten dienen zur Potentialtrennung zwischen dem PC und der Anlagenseite und können sowohl TTL als auch andere Spannungswerte verarbeiten. Um auch größere Ströme bis zu einigen Ampére schalten zu können, setzt man Karten mit elektro-mechanisch arbeitenden Relais oder sogenannte Halbleiter-Relais ein.

Zur Erfassung physikalischer Größen braucht man analog-/digital-Wandlerkarten, die mit Auflösungen zwischen 8 Bit und 24 Bit und Wandlungsraten von einigen kHz bis zu mehreren MHz verfügbar sind. Mit den in gleicher Variationsbreite lieferbaren digital-/analog-Umsetzern kann man die Steuerspannungen erzeugen, mit denen beispielsweise Sollwertvorgaben an analogen Reglern verändert werden können.

Zur Nutzung einer beliebigen I/O-Karte braucht man immer ein speziell auf die jeweilige Karte zugeschnittenes Steuerprogramm, welches für die Einbindung der Karte in das Betriebssystem des Computers sorgt. Im einfachsten Fall ist das ein mehr oder weniger kleines Treiberprogramm, das beim Booten des Rechners geladen und gestartet wird, während des Betriebs aber nicht mehr weiter in Erscheinung tritt.

Aufwendigere Lösungen beinhalten einen oder mehrere Treiber und ein Anwendungsprogramm, das auf eine spezielle Aufgabenstellung zugeschnitten ist. Der Rechner wird dann üblicherweise auch nur für diese eine Anwendung genutzt.

Beschreibung des Moduls

Das Modul MAX 12 digitalisiert Gleichspannungen bis maximal 10 V mit einer Auflösung von 12 bit und stellt die digitalisierten Werte an einer normgerechten RS-232C-Schnittstelle, zur weiteren Verarbeitung im PC, zur Verfügung.

Die Handhabung in der Praxis ist unkompliziert, da das Modul seine Versorgungsspannung direkt aus der Schnittstelle des PCs beziehen kann. Dank der äußerst geringen Stromaufnahme ist auch beim mobilen Einsatz am Notebook nicht mit wesentlichen Einschränkungen der Betriebsdauer zu rechnen.

Mit einem Mehrgangtrimmer ist eine präzise Anpassung an die zu erwartenden Eingangsspannungen möglich. Bei korrektem Abgleich ist dadurch in jedem Fall die volle Auflösung von 12 bit nutzbar.

Der Signalanschluß kann alternativ auf zwei Arten beschaltet werden: entweder als reiner DC-Analogeingang oder als Analogeingang für aktive Sensoren, wie etwa den LM35. Die stabilisierte Versorgungsspannung für den Sensor erzeugt das Modul intern.

Die Messung ist immer massebezogen. Der Anschluß ***Masse***, beziehungsweise ***Masse analog*** am Signaleingang, ist identisch mit der Masse an der Schnittstelle des verwendeten PCs.

Zum Lieferumfang des Moduls gehört ein Pascal-Beispielprogramm, dessen Quelltext die Ansteuerung verdeutlicht. Die ebenfalls mitgelieferte ausführbare Programmdatei erlaubt die sofortige Inbetriebnahme.

Aufbau und Anschluss

Der an der sechspoligen Lüsterklemme zugängliche Signalanschluss erlaubt zwei unterschiedliche Belegungen:

a) Zwischen Pin 4 und Pin 5 (Analogeingang und Masse) kann eine Gleichspannung im Bereich von 0...10 V angeschlossen werden

b) Zwischen Pin 1, 2 und 3 kan direkt ein aktiver Temperatursensor angeschlossen werden (beispielsweise einer vom Typ LM35). Pin 1 liefert die stabilisierte Versorgungsspannung für den Sensor (+5 V), Pin 2 bildet den Sensoreingang und Pin 3 ist der Masseanschluss.

Die Eingangsspannung wird über den als Spannungsteiler geschalteten Mehrgangtrimmer (Gain) zum Messeingang des A/D-Wandlers geführt. Dadurch kann das Modul auch Spannungen verarbeiten, die höher sind als die Referenzspannung. Standardmäßig sind kalibrierte Module in zwei Spannungsbereichen lieferbar: 0...5 V und 0...10 V.

Das Modul wird, je nach Verfügbarkeit am Markt, mit dem A/D-Wandler MAX187 oder MAX189 geliefert. Von der Funktion her unterscheiden sich beide Typen nicht. Der A/D-Wandler MAX187 erzeugt ledigliche zusätzlich intern eine gepufferte Referenzspannung von 4,096 V. Diese wird für unsere Anwendung allerdings nicht benötigt, da auf eine modulintern erzeugte und stabilisierte Referenz im Bereich von 0... 5 V zurückgeriffen wird. Die Einstellung des Mehrgangtrimmers Range legt daher die Messbereichsobergrenze fest.

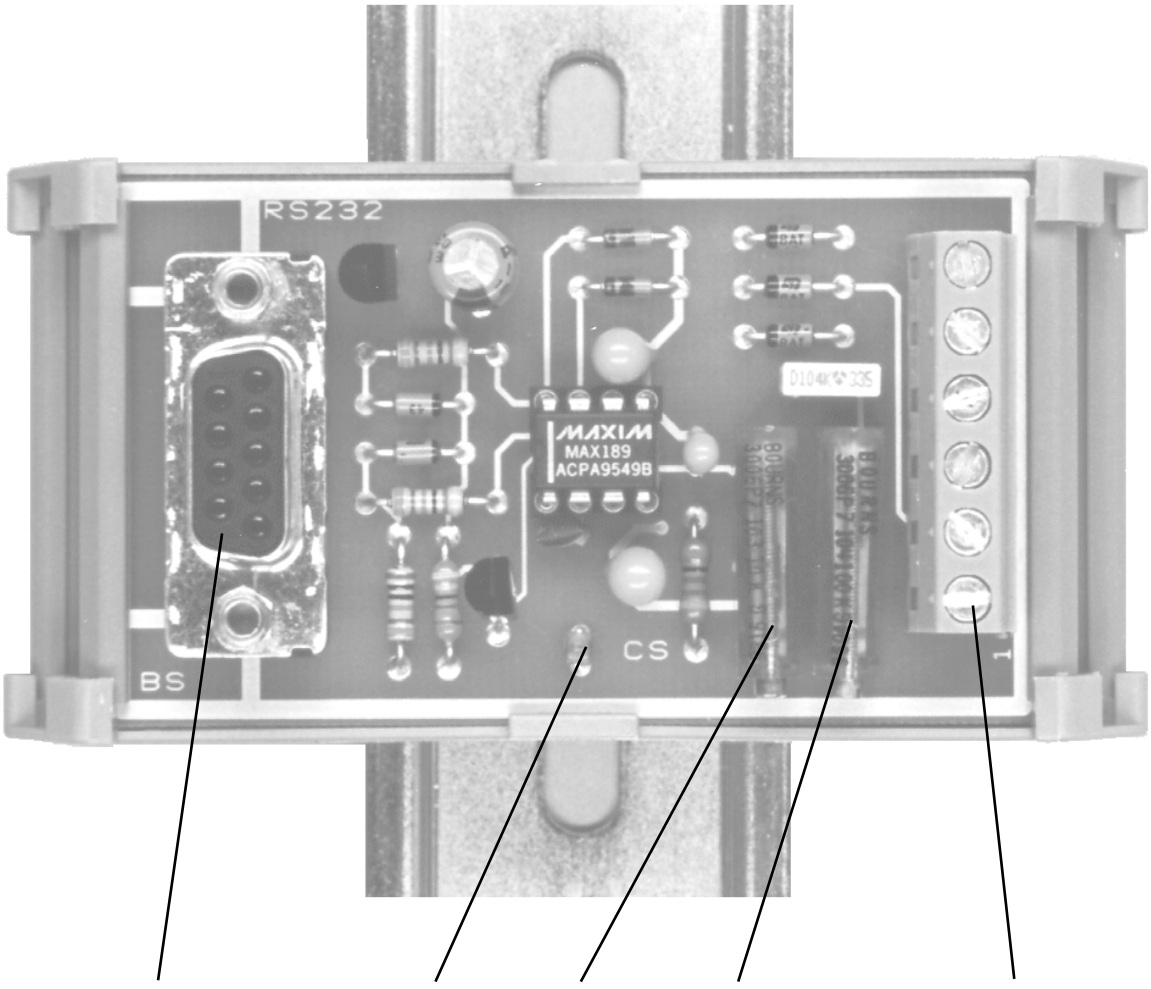
Zum Betrieb am PC beziehungsweise Notebook (12 V) ist keine separate Stromversorgung nötig, das die da die Betriebsspannung direkt aus der seriellen Schnittstelle erzeugt wird. Ein LED zeigt an,

das die Spannung anliegt. Das Modul benötigt zur einwandfreien Funktion einen High-Pegel von mindestens 7,5 V an den Anschlüssen DTR und RTS. Die Stromaufnahme liegt bei knapp 2 mA.

Bei Notebooks mit einer 5 V-Stromversorgung ist allerdings eine externe Spannungsversorgung von 12 V erforderlich, da diese die serielle Schnittstelle mit TTL-Pegeln betreiben. Die Sromversorgung kann dann mit einem Steckernetzteil erfolgen. Der Masseanschluss wird an Pin 5 (Masse) und die Spannung an Pin 6 (Spannungsversorgung extern) gelegt. Dieser Eingang ist zwar durch eine Diode gegen Verpolung geschützt, die angelegte Spannung sollte dennoch nicht wesentlich über 12 V liegen.

Die 9 polige Sub-D-Buchse ist so belegt, das man das Modul mit einem ungekreuzten Kabel (1:1) an die serielle Schnittstelle anschließen kann.

Kartenansicht und Bauteile



RS 232C-Anschluss Kontroll-LED Gain Range Eingangsanschlussleiste

Technische Daten

Meßbereiche	:	0...5 V, 0...10 V
Eingangswiderstand	:	1 MΩ
A/D-Wandlerauflösung	:	12 bit
Unlinearität	:	±1/2 LSB
Abtastrate	:	max. 75 kHz entsprechend 13,3 µs
Referenzspannung	:	Der A/D-Wandler MAX187 erzeugt intern eine gepufferte Referenz spannung von 4,096 V aus einer Bandgap-Quelle. Der Wandler MAX 189 arbeitet mit einer modulintern erzeugten und stabilisierten Referenz im Bereich von 0...5 V. Der Wert ist mit dem Mehrgangtrimmer ***Range*** einstellbar.
Versorgungsspannung	:	mindestens 7,5 V, intern auf 5,0 V stabilisiert
Stromaufnahme	:	< 2 mA im Betrieb < 2 µA im Shutdown-Modus
Temperaturbereich	:	0...70 °C
Abmessungen	:	45 x 90 x 40 mm (Breite x Höhe x Tiefe)

Programmierung in (GW)BASIC

Das folgende BASIC-Programmbeispiel verdeutlicht die Steuerung des Moduls MAX 12. Gleichzeitig kann hiermit der gewünschte COM-Port eingestellt werden. Beachten Sie bitte den Zugriff auf die Leitung RTS der seriellen Schnittstelle in den Zeilen 140 und 220. Hiermit wird sichergestellt, daß der Pufferelko der modulinternen Spannungsversorgung ausreichend geladen ist.

```
100 REM 12 BIT ADC-MODUL an COM1:DTR (4) = 1 Clock
                                 RTS (7) = 2 CS
                                 CTS (8)= DATA

110 CLS : PRINT "Wähle COM1 oder COM2 1,2 :"; : INPUT ADR
120 IF ADR = 1 THEN COMC = &H3FC : COME = COMC + 2
130 IF ADR = 2 THEN COMC = &H2FC : COME = COMC + 2
140 OUT COMC,3 : FOR T=0 TO  50 : NEXT T            : REM Lade Elko 50 ms vor
150 Y=2048 : R=0 : F=5                              : REM Auflösung + Parameter
160 OUT COMC,2 : OUT COMC,0 : OUT COMC,2            : REM CS = 1/0 ADC wandeln
170 FOR BIT=1 TO 12                                 : REM Schleife für 12 Bit
180 OUT COMC,1 : OUT COMC,0                         : REM clocken bei CS = Low
190 IF (INP(COME) AND 16)=16 THEN D=0 ELSE D=1      : REM Input invert Databit
200 OUT COMC,1
210 R=R+(D*Y) : Y=Y/2 : NEXT BIT                    : REM 12-Bit-Wert bilden
220 OUT COMC,3                                      : REM wieder Elko laden
230 LOCATE 3,1
240 PRINT "Bit/dez. = ";:PRINT USING " ##### ";R    : REM Digit Ausgabe
250 VO=(R/4095) * F                                 : REM Umrechnen in Volt
260 PRINT "Volt 0-5 = ";:PRINT USING " #.### ";VO   : REM Ausgabe Spannung
270 V2 = VO * 2
280 PRINT "Volt 0-10= ";:PRINT USING " #.### ";V2   : REM Ausgabe 0-10 Volt
290 TEMP = R/10
300 PRINT "Temperatur ";:PRINT USING " ###.# ";TEMP : REM Ausgabe Grad
310 GOTO 140
```

Programmbeispiel in Pascal

```
{ For low-cost AD12BIT-SERIELL-Modul }
program SPEC;
USES Crt,Dos,Graph,Printer;

type    parms = record
        end;

var     param   : parms;
        fi      : file;
        parm    : file of parms;
        fil     : file of char;
        wmem    : array[0..650] of word;
        fname   : String[20];
        ii      : integer;
        strh    : string;
        charc   : char;


CONST   Bgipath       = '';   {''=im Selben, sonst:'c:\TP7\BGI'; }
        AUFL           = 2048;

var  COMMBASE          : integer;
     STG         : string;
       driver,Gmode      : integer;
     key         : char;
     mode,i,COM       : integer;
     ok,stop          : boolean;
        Y,R,F             : integer;
        BIT,D             : integer;
       VOLT,V2,TEMP       : integer;
        TB                : integer;
      STD,MIN,SEC,SEK100 : word;
        VS                : real;
        TT,SAVE           : integer;

{=========================================================================}
procedure SelectCOM(com:integer; var ok:boolean);
  begin
    if com=1 then
      begin
       CommBase:=$3f8;
        ok:=true;
      end
    else if com=2 then
      begin
       CommBase:=$2f8;
        ok:=true
      end
  end;
{=========================================================================}
```

```
procedure InitComm; (* Initialize communication port *)
begin
  port[CommBase+3]:=$03;
  port[CommBase+3]:=$83;
  port[CommBase  ]:=$60;
  port[CommBase+1]:=$0 ;
  port[CommBase+3]:=$03;
  port[CommBase+1]:=$0 ;
  port[CommBase+4]:=$09;    { set DTR=H, RTS=L, OUT2=H }
end;
{=============================================================================}
{$R-}{$S-}
procedure LOG_FILE;  { Speichern im ASCII-Format / Binaer 0...4096 }
Label 11;

begin
charc:= ' '; { Trennzeichen zwischen den Werten }
assign(fil,fname+'.LOG');
 rewrite(fil);
  if IOresult <> 0 then goto 11;

   strh[1]   := 'A';  { Header schreiben }
   strh[2]   := 'D';
   strh[3]   := '1';
   strh[4]   := '2';
   strh[5]   := 'L';
   strh[6]   := 'O';
   strh[7]   := 'G';
   strh[8]   := '-';
   strh[9]   := 'B';
   strh[10]  := 'I';
   strh[11]  := 'N';
   strh[12]  := '•';
   strh[13]  := 'R';
   strh[14]  := ':';
   strh[15]  := ' ';
   strh[16]  := ' ';

  for i := 1 to 16 do
   begin
   write(fil,strh[i]);
    end;

  for ii := 3 to 650 do begin
 { write(fil,wmem[ii]); }
   R := wmem[ii];
   Str(R:4,Strh);
   write(fil,strh[1],strh[2],strh[3],strh[4],charc);  { disk schreiben }
 end;
```

```
   strh[1]  := ' ';
   strh[2]  := 'K';
   strh[3]  := 'O';
   strh[4]  := 'L';
   strh[5]  := 'T';
   strh[6]  := 'E';
   strh[7]  := 'R';
   strh[8]  := ' ';

  for i := 1 to 8 do
   begin
   write(fil,strh[i]);
   end;

close(fil);
11:
end;

{=============================================================================}

Procedure PTIME;     { Uhrzeit ausgeben }
begin
 SetFillStyle(0,0);
   SetColor(01);
GETTIME(STD,MIN,SEC,SEK100);
Bar(530,55,620,66);
SetColor(15);
Str(STD,STG);
OutTextXY(535,57,STG);
OutTextXY(555,57,':');
Str(MIN,STG);
OutTextXY(565,57,STG);
OutTextXY(585,57,':');
Str(SEC,STG);
OutTextXY(595,57,STG);
end;

Procedure Raster;
begin
 SetFillStyle(01,08); { loesche ges. spec. }
  Bar(2,74,637,450);
 SetColor(03);
 SetLineStyle(0,0,1);
  Rectangle(2,75,600,450);
  Rectangle(614,75,621,450);

   LINE( 2 ,150,600,150);
   LINE( 2 ,225,600,225);
   LINE( 2 ,300,600,300);
```

```
  LINE( 2 ,375,600,375);
  LINE(100,75 ,100,450);
  LINE(200,75 ,200,450);
  LINE(300,75 ,300,450);
  LINE(400,75 ,400,450);
  LINE(500,75 ,500,450);
  LINE(600,75 ,600,450);
 SetColor(06);
 SetLineStyle(1,1,1);
  LINE(  3,113,599,113);
  LINE(  3,187,599,187);
  LINE(  3,262,599,262);
  LINE(  3,337,599,337);
  LINE(  3,412,599,412);
 SetLineStyle(0,0,1);
 SetColor(02);
IF (VS=0) then
 begin
  OutTextXY(5,440,'0V');
  OutTextXY(5,400,'1V');
  OutTextXY(5,418,'0,5V');
  OutTextXY(5,250,'5V');
  OutTextXY(5,268,'2,5V');
  OutTextXY(5,100,'9V');
  OutTextXY(5,118,'4,5V');
  OutTextXY(5,440,'0V');
  OutTextXY(5,365,'2V');
  OutTextXY(5,290,'4V');
  OutTextXY(5,215,'6V');
  OutTextXY(5,140,'8V');
 end;

IF (VS=5.5) then
 begin
  OutTextXY(5,440,'0V');
  OutTextXY(5,365,'1V');
  OutTextXY(5,290,'2V');
  OutTextXY(5,215,'3V');
  OutTextXY(5,140,'4V');
 end;

IF (VS=9.91) then
 begin
  OutTextXY(5,440,'0V');
  OutTextXY(5,365,'0.2V');
  OutTextXY(5,290,'0.4V');
  OutTextXY(5,215,'0.6V');
  OutTextXY(5,140,'0.8V');
 end;
end;
```

```
procedure DisplayCLS;
begin
 SetFillStyle(01,08); { loesche ges. spec. }
  Bar(2,51,637,478);
 SetFillStyle(01,07); { style, farbe }
  Bar(2, 2,637, 48);
 SetFillStyle(0,0);
   SetColor(01);
   Rectangle(0,0,639,479);
 SetTextStyle(0,0,0);

if SAVE=0 then
 begin
  SetColor(10);
  OutTextXY(380,57,'S:Log OFF');
 end;

IF SAVE=1 then
 begin
  SetColor(12);
  OutTextXY(380,57,'S:Log ON ');
 end;

SetColor(10);
if COM=1 then
 OutTextXY(190,57,'C:COM1-03F8');
if COM=2 then
 OutTextXY(190,57,'C:COM2-02F8');
    OutTextXY(100,57,'ESC:Ende ');
    OutTextXY(310,57,'T:Stop ');

  SetColor(01);
    OutTextXY(400,04,'Tasten:');
    OutTextXY(400,15,'1..3  Darstellung Volt');
    OutTextXY(400,25,'Space Clear-Screen ');
    OutTextXY(400,37,'+ -   Zeitbasis:');

  SetFillStyle(01,03); { loesche style,col. }
  Bar(05,07,380,44);
  SetTextStyle(10,0,1);
  OutTextXY(10,1,'AD-12Bit Seriell-Modul');
  SetFillStyle(01,00); { loesche style,col. }
  SetTextStyle(0,0,0);

BAR(580,13,620,25);
SetColor(15);
IF (VS =   0 ) then OutTextXY(585,15,'0-10');
IF (VS = 5.5 ) then OutTextXY(585,15,'0-5');
IF (VS = 9.91) then OutTextXY(585,15,'0-1');
```

```
BAR(530,35,620,46);
SetColor(15);

IF TB = 1 then
 begin
 OutTextXY(535,37,'0-6 sec.');
   SetColor(10);
   OutTextXY( 03,458,'0 sec');
   OutTextXY( 80,458,'1 sec');
   OutTextXY(185,458,'2 sec');
   OutTextXY(285,458,'3 sec');
   OutTextXY(385,458,'4 sec');
   OutTextXY(485,458,'5 sec');
   OutTextXY(585,458,'6 sec');
  SetColor(11);
   OutTextXY( 80,468,'1000');
   OutTextXY(185,468,'2000');
   OutTextXY(285,468,'3000');
   OutTextXY(385,468,'4000');
   OutTextXY(485,468,'5000');
   OutTextXY(585,468,'-> ms');
 end;

IF TB = 2 then
 begin
 OutTextXY(535,37,'0-36 sec.');
   SetColor(10);
   OutTextXY( 03,458,'0 sec');
   OutTextXY( 80,458,'6 sec');
   OutTextXY(185,458,'12 sec');
   OutTextXY(285,458,'18 sec');
   OutTextXY(385,458,'24 sec');
   OutTextXY(485,458,'30 sec');
   OutTextXY(585,458,'36 sec');
  SetColor(11);
   OutTextXY( 80,468,'0,1 ');
   OutTextXY(185,468,'0,2 ');
   OutTextXY(285,468,'0,3 ');
   OutTextXY(385,468,'0,4 ');
   OutTextXY(485,468,'0,5 ');
   OutTextXY(585,468,'-> min.');
 end;

IF TB = 3 then
 begin
 OutTextXY(535,37,'0-2 min. ');
   SetColor(10);
   OutTextXY( 03,458,'0    ');
   OutTextXY( 80,458,'20 sec');
```

```
   OutTextXY(185,458,'40 sec');
   OutTextXY(285,458,'60 sec');
   OutTextXY(385,458,'80 sec');
   OutTextXY(485,458,'100 sec');
   OutTextXY(585,458,'120 s.');
 end;

IF TB = 4 then
 begin
 OutTextXY(535,37,'0-12 min.');
   SetColor(10);
   OutTextXY( 03,458,'0     ');
   OutTextXY( 80,458,'2 min. ');
   OutTextXY(185,458,'4 min. ');
   OutTextXY(285,458,'6 min. ');
   OutTextXY(385,458,'8 min. ');
   OutTextXY(485,458,'10 min.');
   OutTextXY(585,458,'12 m.');
 end;

  SetFillStyle(01,00); { loesche style,col. }
SetColor(0);
end;


procedure INITEXT;
begin
  SetColor(14);
  SetTextStyle(7,0,3);
    OutTextXY(40, 80,'AD-12-Bit Seriell-Modul');
    OutTextXY(40,120,'COPYRIGHT by KOLTER ELECTRONIC');
    OutTextXY(40,160,'(c) 1995/96 ');
   SetTextStyle(0,0,0);
SOUND(3000); Delay(100); NOSOUND;
DELAY(1500);
DisplayCLS;
end;

(* ========================================================================== *)

procedure ELKO; { Ladungspumpe min. 6ms aufladen }
begin
PORT[COMMBASE+4] := 3;
Delay(6);
end;

procedure Wandel;
begin
PORT[COMMBASE+4] := 2;
```

```
Delay(1);
PORT[COMMBASE+4] := 1;
Delay(1);
PORT[COMMBASE+4] := 2;
end;

procedure Lesen;
begin
Y:= AUFL;
R := 0;
FOR BIT := 1 to 12 do
begin
For TT := 0 to 8 do
 begin
PORT[COMMBASE+4] := 1;
 end;
For TT := 0 to 5 do
 begin
PORT[COMMBASE+4] := 0;
 end;

IF ((PORT[COMMBASE+6] AND 16) = 16)
  then R:=R     { D=0 }
   else R:=R+Y;  { D=1 }
Y:=Y div 2;
end;
PORT[COMMBASE+4] := 3;
{VOLT := (R div 4096) * F;}
end;

procedure ADMESS;
var OLD_R  : integer;
    YY     : integer;
    DD     : integer;
   STD,MIN,SEC,SEK100 : word;
begin
{ -- Ab hier jetzt die Graphik !!! ---}
DD := 0;  { Anpassung - Zeitachse falls erforderlich }
RASTER;
SetFillStyle(01,12); { 01 style, 09 farbe }
SetColor(14);
for i:= 3 to 599 do
begin
ELKO;                       { Beispiel fuer AT-286/20MHz}
DELAY(DD);                  { ca.-Werte je nach Rechner neu eingeben}
IF (TB = 2) then delay(   50); { 49 Zeitbasis 36 sekunden }
IF (TB = 3) then delay(  190); { 188 fuer 2min. }
IF (TB = 4) then delay( 1225); { 1225 fuer 12min.}
```

```
Wandel;
Lesen;
YY := 449-(ROUND(R / (11-VS)));    { Anzeige Y verstaerken mit VS }
If (YY<76) then YY:= 76 ;         { Begrenzung der Anzeige }
If (i>4) then Line(i-1,OLD_R,i,YY); { Graph zeichnen }
If KeyPressed then exit;
wmem[i] := R;          { Werte in Array fuer LOG-File zwischenspeichern }
OLD_R := YY;
BAR(615,YY-1,620,YY+1);
end;
IF SAVE=1 then LOG_FILE; { wenn S dann Werte in AD12LOG schreiben }
end;

(* ========================================================================== *)

Procedure PORTSEL;      { COM-Port waehlen }
begin
IF COM=1 then
begin
  COM:=2;
SelectCOM(com,ok);
  Stop := false;
  DISPLAYCLS;
  EXIT;
end;
IF COM=2 then
begin
  COM:=1;
SelectCOM(com,ok);
  Stop := false;
  DISPLAYCLS;
  EXIT;
end;
end;

Procedure SAVESEL;
begin
IF SAVE=1 then
begin
  SAVE:=0;
  DISPLAYCLS;
  SetColor(10);
  OutTextXY(380,57,'S:Log OFF');
  EXIT;
end;

IF SAVE=0 then
begin
```

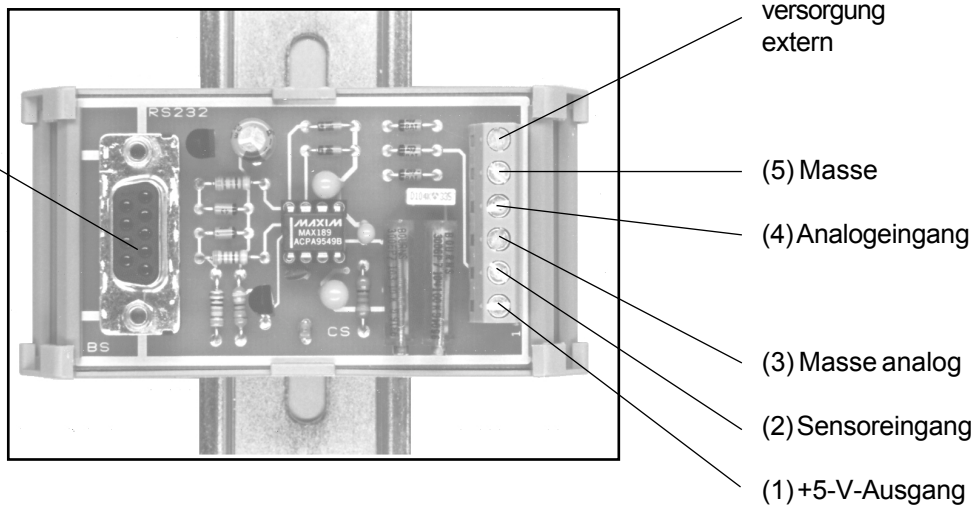
```
  SAVE:=1;
  DISPLAYCLS;
  SetColor(12);
  OutTextXY(380,57,'S:Log ON ');
  EXIT;
end;
end;


(* ========================================================================== *)
begin
DetectGraph(driver,Gmode);
InitGraph(driver,Gmode,bgipath);
DISPLAYCLS;
INITEXT;

fname := 'AD12LOG';

F    := 5; { 5Volt Range }
TB   := 1; { Time-Base }
COM  := 1;
VS   := 0;
SAVE := 0;

SelectCOM(com,ok);
Stop := false;
PORTSEL;

PORT[COMMBASE+4] := 3;
Delay(200);

Repeat
 PTIME;
 ADMESS;
   if KeyPressed then
     begin
      key:=Readkey;  { Kurztasten abfragen }
       case key of
       #27: Stop := true;
     { #49: DISPLAYCLS;}   { Loeschen und neu }

   't','T': begin
             SetColor(12);
            OutTextXY(310,57,'T:Stop ');
             key:=ReadKey;
             SetColor(10);
            OutTextXY(310,57,'T:Stop ');
              end;
```

```
    'c','C': PORTSEL;

        '+': begin
             TB:=TB+1;
            DISPLAYCLS;
             end;

        '-': begin
            IF (TB >1) then TB:=TB-1;
            DISPLAYCLS;
             end;

        '1': begin
             VS:=0;
            DISPLAYCLS;
             end;

        '2': begin
             VS:=5.5;
            DISPLAYCLS;
             end;

        '3': begin
             VS:= 9.91;
            DISPLAYCLS;
             end;

    'S','s': SAVESEL;

       end;
      end;
until stop;
ClrScr;
TextMode(CO80);
end.
```

KOLTER ELECTRONIC
High-Tech Manufacturing in EUROPE Import-Export

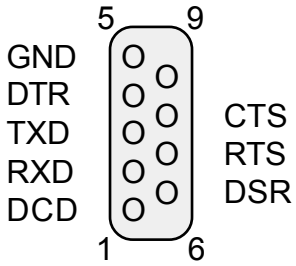
Steckerbelegungen

Belegung der Klemmleiste



Belegung der 9 poligen Sub-D Buchse

Sub-D-Anschluß



Erklärung der Abkürzungen

DCD = Data Carrier Detect
DSR = Data Set Ready
GND = Signal Ground
DTR = Data Terminal Ready
CTS = Clear To Send
RTS = Request To Send
TXD = Transmitt Data
RXD = Recive Data

KOLTER ELECTRONIC
High-Tech Manufacturing in EUROPE Import-Export

Anschriften und Rufnummernverzeichnis

Anschriften

Postfach 1127 D-50362 Erftstadt
Steinstraße 22 D-50374 Erftstadt

Ruf- und Faxnummern

Auslandsvorwahl	++49 22 35
Inlandsvorwahl	0 22 35
Telefon Vertrieb und Service	7 67 07
Fax	7 20 48
Werkstatt und Prüffeld	69 18 52
Pressestelle	95 37 31
Geschäftsleitung	95 37 32

Internet

E-Mail - Service	service@pci-card.com
Haupt-Domains	http//www.pci-card.com
	http//www.kolter.de

CE

EMV-Konformität:

Die EMV-Konformität gilt für industielle Einrichtungen bzw. ortsfeste Anlagen.
Der Einsatz im priv. Haushalt ist auf Grund der Prüfungsvorschriften untersagt.

Die elektromagnetische Verträglichkeit wurde nach 2004/108/EG
(vormals 89/336/EWG) nachgewiesen.

Folgende Fachgrundnormen wurden bei der EMV-Prüfung angewandt:

- DIN EN 50 081-2 (EMV Störaussendung - Industrie)
- DIN EN 50 082-2 (EMV Störfestigkeit - Industrie)
- und weitere IEC-Prüfungen

Die komplette EG-Konformitätserklärung können Sie auch unter folgender
URL als PDF-Dokument herunterladen: http://www.pci-card.com/ce-non-pci.pdf

Diese Erklärung bescheinigt die Übereinstimmung mit den genannten Richtlinien, ist jedoch keine Zusicherung von Eigenschaften im Sinne des Produkthaftungsgesetzes. Die Sicherheitshinweise auf unserer Webseite, sowie in der mitgelieferten Produktinformation sind zu beachten. Weitere Informationen unter: http://www.pci-card.com/faq015.html