

### Eingabemodul (digital)

Das digitale Eingabemodul dient der Aufzeichnung von Rückmeldungen des plan & simulation-Modells, bei denen zwischen zwei Zuständen unterschieden wird (z.B. Schalter geschlossen - Schalter offen, Teil vorhanden - Teil nicht vorhanden).

Das digitale Eingabemodul ist kompatibel zu den übrigen Baugruppen und Modulen des plan & simulation Interface-Systems und kann daher an allen durch Computer-Anbindungssets unterstützten Personal-Computern eingesetzt werden. Die Programmierung erfolgt durch unkomplizierte E/A-Registerabfrage und -steuerung (s. Software-Module).

Das Eingabemodul besitzt 16 potentialfreie Eingänge. Die Eingangsspannung darf bis zu 30V Gleichspannung betragen. Spannungen zwischen 0V und 2V werden als Low-Pegel, Spannungen zwischen 5V und 30V werden als High-Pegel erkannt. Der maximale Eingangsstrom bei High-Signal beträgt ca. 30mA. Diese Werte entsprechen weitgehend den Eigenschaften üblicher SPS-Steuerungen, so daß die bei SPS-gesteuerten Anlagen üblichen Sensoren an das Eingabemodul angeschlossen werden können.

Wegen der Potentialtrennung kann das Eingabemodul auch in "rauer" Umgebung betrieben werden, wenn z.B. die Gefahr von Störspitzen durch das Schalten induktiver Lasten besteht oder die Gefahr elektrostatische Aufladung gegeben ist.

Zur Funktionskontrolle des plan & simulation-Modells besitzt jeder Eingang eine Leuchtdioden-anzeige. Ein Aufleuchten der Leuchtdiode bedeutet High-Pegel.

### Technische Daten des Eingabemoduls (digital)

Eingabemodul für 16 digitale Eingangssignale (0V bis 30V).  
Die 16 Eingangsstufen sind identisch.

Schaltpegel:           0V bis 2V   - Low-Pegel  
                              5V bis 30V   - High-Pegel  
Bei Verwendung von Treiber-IC kann das Eingabemodul auch mit TTL-Bausteinen gesteuert werden.

Eingangsstrom:       ca. 30 mA bei High-Pegel

Eingang potentialfrei

Abmessungen:       249 mm x 35 mm x 129 mm (Tiefe x Breite x Höhe)  
Leiterplatte:       220 mm x 100 mm  
Frontplatte         7 Teileinheiten breit

Gewicht:             260 g

```

{ Industrie - Interfacetreiber Pascal 5.0   Vers.   3.1 }
CONST Basisadresse = $300; { Basisadresse der Einsteckkarte.
    Weitere mögliche Adressen: $308, $310, $318 }
Port_A1 = Basisadresse + $0; { Ausgabe low Byte }
Port_A2 = Basisadresse + $1; { Ausgabe high Byte }
Port_B1 = Basisadresse + $2; { Eingabe low Byte }
Port_B2 = Basisadresse + $3; { Eingabe high Byte }
Port_C1 = Basisadresse + $4; { Kontrolle }
Port_C2 = Basisadresse + $5; { Modul-Adresse }
Port_Steuerregister_1 = Basisadresse + $6; { Steuerregister 1 }
Port_Steuerregister_2 = Basisadresse + $7; { Steuerregister 2 }
An = TRUE;
Aus = FALSE;
Bitwert : ARRAY [1..16] OF Longint = [1,2,4,8,16,32,64,128,
    256,512,1024,2048,4096,8192,16384,32768];
TYPE t_Bitmuster = ARRAY [0..15] OF Longint;
VAR Aktive_Karte, Low_Byte, High_Byte : BYTE;
    Bitmuster : Longint;
    Bitmuster_Speicher : t_Bitmuster;
PROCEDURE Karte (Karte: BYTE);
BEGIN
    PORT [Port_C1] := 0;
    PORT [Port_C2] := 24;
    PORT [Port_A1] := LO[Bitmuster_Speicher[Karte]];
    PORT [Port_A2] := HI[Bitmuster_Speicher[Karte]];
    PORT [Port_C1] := 0;
    PORT [Port_C2] := Karte;
    Aktive_Karte := Karte;
END; { Karte }
FUNCTION Eingang_Bitmuster : Longint;
BEGIN
    Bitmuster := PORT[Port_B2];
    Bitmuster := Bitmuster SHL 8 + PORT[Port_B1];
    Bitmuster_Speicher[Aktive_Karte] := Bitmuster;
    Eingang_Bitmuster := Bitmuster;
END; { Eingang_Bitmuster }
FUNCTION Eingang(Eingang_Nr : BYTE) : BOOLEAN;
BEGIN
    Bitmuster_Speicher[Aktive_Karte] := PORT[Port_B2];
    Bitmuster_Speicher[Aktive_Karte] :=
        Bitmuster_Speicher[Aktive_Karte] SHL 8 + PORT[Port_B1];
    Eingang := Bitmuster_Speicher[Aktive_Karte] =
        (Bitmuster_Speicher[Aktive_Karte] OR Bitwert[Eingang_Nr]);
END; { Eingang }
FUNCTION Karte_Eingang(KarteNr, Eingang_Nr : BYTE) : BOOLEAN;
BEGIN { Eingang }
    Karte(KarteNr);
    Bitmuster_Speicher[Aktive_Karte] := PORT[Port_B2];
    Bitmuster_Speicher[Aktive_Karte] :=
        Bitmuster_Speicher[Aktive_Karte] SHL 8 + PORT[Port_B1];
    Karte_Eingang := Bitmuster_Speicher[Aktive_Karte] =
        (Bitmuster_Speicher[Aktive_Karte] OR Bitwert[Eingang_Nr]);
END; { Karte_Eingang }
PROCEDURE Interface_initialisieren;
VAR i : INTEGER;
BEGIN
    PORT [Port_Steuerregister_1] := $82; { = Baustein 8255 }
    PORT [Port_Steuerregister_2] := $82; { = Baustein 8255 }
    FOR i := 0 TO 15 DO { 0-7 für Interface Nr. 1; 8-15 für Nr.2 }
    BEGIN
        Bitmuster_Speicher[i] := 0;
        Karte(i);
        Ausgang_Bitmuster(0);
    END; { FOR i }
END; { Interface_initialisieren }

```

## Programmierung des Eingabemoduls

Zur Programmierung des Eingabemoduls werden nebenstehende Pascal-Prozeduren empfohlen, die Sie (neben anderen) in der Include-Datei INDTREIB.INC auf der Diskette vorfinden.

Der Vorspann der Datei enthält verschiedene Konstanten-, Typen- und Variablendefinitionen. Diese sollten global, also auch in dem Benutzerprogramm, gelten.

Als erstes sollte das Benutzerprogramm die Prozedur **Interface\_initialisieren** aufrufen. Danach muß mit der Prozedur **Karte** der Adressierlogik die Nummer des Steckplatzes mitgeteilt werden, in dem das Eingabemodul eingeschoben ist.

Anschließend können die drei Funktionsprozeduren zur Eingabe je nach Anwendung benutzt werden.

Die Funktion **Eingang\_Bitmuster** liefert als Ergebnis ein 16-Bit-Datenwort, das die logischen Zustände der 16 Eingangsleitungen an den zugeordneten Bitpositionen enthält.

Die Funktion **Eingang** nimmt als Argument eine Zahl, die die Eingangsleitung (nicht Bitposition!) bezeichnet. Der Funktionswert ist **TRUE**, wenn der Eingang High-Pegel führt, und **FALSE**, wenn der Eingang Low-Pegel führt.

Die Funktionsprozedur **Karte\_Eingang** vereint die Prozedur **Karte** mit der Funktion **Eingang** und ist besonders bei Installationen mit mehreren Interfacemodulen zu empfehlen. Sie nimmt die Nummer des Steckplatzes und die Nummer der Eingangsleitung und liefert den Funktionswert **TRUE** für High-Pegel und **FALSE** für Low-Pegel.

```

1 DIM KLOBY[15], KHIBY[15]
10 GOTO 10000
99 :
100 REM 8255 initialisieren
110 DEFSEG = 0
120 ADR = &H300
130 OUT ADR + 6, &H82
140 OUT ADR + 7, &H82
150 FOR I = 0 TO 15
160 OUT ADR + 5, I
170 OUT ADR + 4, 0
180 OUT ADR, 0 : OUT ADR + 1, 0
185 KLOBY[I] = 0
186 KHIBY[I] = 0
190 NEXT I
195 RETURN
199 :
200 REM Kartenauswahl
205 OUT ADR + 4, 0
206 OUT ADR + 5, 24
207 OUT ADR, KLOBY[KARTE]
208 OUT ADR + 1, KHIBY[KARTE]
210 OUT ADR + 5, KARTE
220 OUT ADR + 4, 0
230 RETURN
399 :
400 REM Eingang
410 EINGANG1 = INP(ADR+2)
420 EINGANG2 = INP(ADR+3)
430 RETURN
999 :
1000 REM Biteingang
1010 GOSUB 400
1020 IF BIT > 8 THEN 1100
1030 BITWERT = 2^(BIT-1)
1040 IF EINGANG1 = (EINGANG1 OR BITWERT) THEN AN = 1
      ELSE AN = 0
1050 RETURN
1100 REM bit > 8
1110 BITWERT = 2^(BIT - 9)
1120 IF EINGANG2 = (EINGANG2 OR BITWERT) THEN AN = 1
      ELSE AN = 0
1130 RETURN
9999 :
10000 REM BEGINN DES HAUPTPROGRAMMS...

```

## Programmierung des Eingabemoduls

Zur Programmierung des Eingabemoduls werden nebenstehende BASIC-Unterprogramme empfohlen, die Sie (neben anderen) in der Datei INDTREIB.BAS auf der Diskette vorfinden.

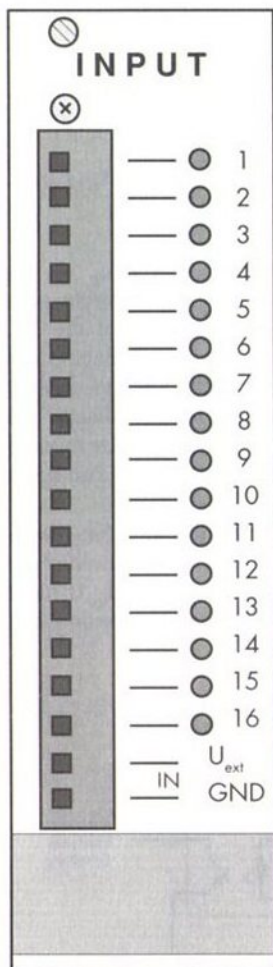
Als erstes sollte das Benutzerprogramm das Unterprogramm **8255 initialisieren** ab Zeile 100 aufrufen. Danach muß mit dem Unterprogramm **Kartenauswahl** (ab Zeile 200) der Adressierlogik die Nummer des Steckplatzes mitgeteilt werden, in dem das Eingabemodul eingeschoben ist.

Anschließend können die beiden Unterprogramme zur Eingabe je nach Anwendung benutzt werden.

Das Unterprogramm **Eingang** (ab Zeile 400) liefert als Ergebnis die beiden Variablen EINGANG1 und EINGANG2, die die Zustände der 16 Eingangsleitungen in Form zweier Zahlenwerte zwischen 0 und 255 enthalten.

Das Unterprogramm **Biteingang** (ab Zeile 1000) nimmt den Inhalt der Variablen BIT, der die Eingangsleitung (nicht Bitposition!) bezeichnet. Die Rückgabe erfolgt mit der Variablen AN. Deren Wert ist 1, wenn der Eingang High-Pegel führt, und 0, wenn der Eingang Low-Pegel führt.

## Steckerbelegung des Eingabemoduls



Eingang 1	-	Bit 0	Eingaberegister Low-Byte
Eingang 2	-	Bit 1	Eingaberegister Low-Byte
Eingang 3	-	Bit 2	Eingaberegister Low-Byte
Eingang 4	-	Bit 3	Eingaberegister Low-Byte
Eingang 5	-	Bit 4	Eingaberegister Low-Byte
Eingang 6	-	Bit 5	Eingaberegister Low-Byte
Eingang 7	-	Bit 6	Eingaberegister Low-Byte
Eingang 8	-	Bit 7	Eingaberegister Low-Byte
Eingang 9	-	Bit 0	Eingaberegister High-Byte
Eingang 10	-	Bit 1	Eingaberegister High-Byte
Eingang 11	-	Bit 2	Eingaberegister High-Byte
Eingang 12	-	Bit 3	Eingaberegister High-Byte
Eingang 13	-	Bit 4	Eingaberegister High-Byte
Eingang 14	-	Bit 5	Eingaberegister High-Byte
Eingang 15	-	Bit 6	Eingaberegister High-Byte
Eingang 16	-	Bit 7	Eingaberegister High-Byte

nicht verbunden, kann für U<sub>ext</sub> benutzt werden.  
Masseanschluß, gemeinsame Signalerückleitung.

## Kombination des digitalen Eingabemoduls mit anderen Komponenten

Zum Betrieb des digitalen Eingabemoduls ist erforderlich:

- ein Baugruppenträger, Art. Nr. 30 915 (enthalten in Grundkonfiguration Art. Nr. 30 914),
- ein Netzteil, Art. Nr. 30 916 (enthalten in Grundkonfiguration Art. Nr. 30 914),
- ein Computer-Anbindungsset:  
z.B.: IBM-PC-Anbindungsset für Computer nach dem IBM-XT® bzw. IBM-AT®-Industriestandard (Art. Nr. 30 917),  
weitere Computer-Anbindungssets befinden sich in Entwicklung.

Eingangsseitig können jegliche Art von Signalgeber angeschlossen werden, die eine Ausgangsspannung zwischen 0V und 30V liefern. Für einen High-Pegel muß das Signal über 5V liegen und mit mindestens 10mA belastbar sein. Die Ausgangsspannung für Low-Pegel muß unter 2V liegen. Im Übergangsbereich zwischen 2V und 5V ist das Schaltverhalten nicht definiert. Allerdings schalten auch Treiber-IC nach dem TTL-Standard in aller Regel die Eingangstufe noch einwandfrei durch. Passive Sensoren wie z.B. Schalter oder Taster müssen mit einer externen Spannungsquelle versorgt werden. In begrenztem Umfang eignet sich dazu der 24-V-Spannungsausgang des Netzteils, Art. Nr. 30 916.

Die Schaltungsskizze zeigt den Innenaufbau einer Eingangsstufe des Eingabemoduls sowie den beispielhaften Anschluß verschiedener Signalgeber. Beachten Sie, daß die Signalerückleitung aller Eingänge über die gemeinsame Massebuchse GND erfolgt. Der Anschluß  $U_{ext}$  ist nicht mit der Schaltung verbunden; er kann als Knotenpunkt für die Versorgungsspannung der passiven Signalgeber dienen.

