

Beispielprogramm für das Modul: MBS

Anwendung des I/O-Expanders 82C55 an der parallelen Schnittstelle

Voraussetzungen ... :

- parallele Schnittstelle im BIOS auf EPP-Modus eingestellt

Zusatzinformationen, Anfragen und Vorschläge bitte unter:

- Adresse : Ingolf Bauer
An der Wetterseite 2a
38889 Blankenburg/ Harz
Deutschland
- Email : ingolf.bauer@nexgo.de

Abkürzungen:

- ADR : Adresse
- BS : Bildschirm
- HG : Hintergrundfarbe ("Papier")
- IC01 ... : 1. Schaltkreis 82C55
- IC02 ... : 2. Schaltkreis 82C55
- PSS : parallele Schnittstelle
- VG : Vordergrundfarbe ("Schrift")

Variablen:

- ADRIC01 : Adreßbyte Vorbereitung Ausgabe Kontrolbyte IC01 (EB Hex)
- ADRIC02 : Adreßbyte Vorbereitung Ausgabe Kontrolbyte IC02 (DB Hex)
- BENEIN . : benutzerdefinierter Eingänge (Pin 12, 13, 15 der PSS)
- BETART . : Betriebsart für IC01/IC02 82C55 (80 Hex)
- BETMOD . : Betriebsmodus für die Ports von IC01/IC02 82C55 (Feld)
- ~ (1) . : IC01-Port A: Eingabe = 1, Ausgabe = 0 lt. Tabelle Bit 4
- ~ (2) . : IC01-Port B: Eingabe = 1, Ausgabe = 0 lt. Tabelle Bit 1
- ~ (3) . : IC01-Port C: Eingabe = 1, Ausgabe = 0 lt. Tabelle Bit 0/3
- ~ (4) . : IC02-Port A: Eingabe = 1, Ausgabe = 0 lt. Tabelle Bit 4
- ~ (5) . : IC02-Port B: Eingabe = 1, Ausgabe = 0 lt. Tabelle Bit 1
- ~ (6) . : IC02-Port C: Eingabe = 1, Ausgabe = 0 lt. Tabelle Bit 0/3
- DATWER . : Datenwerte der Ports (Feld)
- ~ (1) . : Datenwert IC01-Port A
- ~ (2) . : Datenwert IC01-Port B
- ~ (3) . : Datenwert IC01-Port C
- ~ (4) . : Datenwert IC02-Port A
- ~ (5) . : Datenwert IC02-Port B
- ~ (6) . : Datenwert IC02-Port C
- DATBYT. : Datenbyte (Datenregister der PSS)
- DBIC01 . : Port-Funktion IC01; Ein- und Ausgabe Datenbyte
- DBIC02 . : Port-Funktion IC02; Ein- und Ausgabe Datenbyte
- HS01\$.. : Hilfsstring 01
- HV01 ... : Hilfsvariable 01
- HV02 ... : Hilfsvariable 02
- HV03 ... : Hilfsvariable 03
- KONBYT . : Kontrolbyte (Steuerungsregister der PSS)
- LPT : Basisadresse der parallelen Schnittstelle (378 Hex)
- INIIC01 : Portinitialisierung IC01
- INIIC02 : Portinitialisierung IC02
- STABYT . : Statusbyte (Eingangsregister der PSS)

```

***** Programmstart *****

'Deklaration der Funktionen
'-----
DECLARE FUNCTION HIB$ (HEXWER AS INTEGER)      'wandelt HEX- in BIN-Anzeige um

'Deklaration der Subroutinen
'-----
DECLARE SUB Lesen ()                          'Datenbyte von Port lesen
DECLARE SUB Schreiben ()                      'Datenbyte auf Port ausgeben
DECLARE SUB Status ()                        'benutzerdef. Eingänge lesen

'Deklaration der globalen Variablen
'-----
COMMON SHARED ADRIC01 AS INTEGER              'ADR für Kontrolbyte IC01
COMMON SHARED ADRIC02 AS INTEGER              'ADR für Kontrolbyte IC02
COMMON SHARED BENEIN AS INTEGER               'benutzerdef. Eingangsbyte
COMMON SHARED HV01 AS INTEGER                 'Hilfsvariable 01
COMMON SHARED HV02 AS INTEGER                 'Hilfsvariable 02
COMMON SHARED HV03 AS INTEGER                 'Hilfsvariable 03
COMMON SHARED LPT AS INTEGER                  'ADR der PSS
COMMON SHARED STABYT AS INTEGER               'Statusbyte (Eingangsregister)

'Deklaration der globalen Felder
'-----
DIM SHARED DATWER(1 TO 6) AS INTEGER          'Datenwerte der Ports
DIM SHARED BETMOD(1 TO 6) AS INTEGER          'Betriebsarten der Port

TYPE Variablen
    BETART AS INTEGER                        'Betriebsart 82C55
    KONBYT AS INTEGER                        'Kontrolbyte (Steuerungsregister)
    DATBYT AS INTEGER                        'Datenbyte (Datenregister)
    DBIC01 AS INTEGER                        'IC01 Port-Funktion
    DBIC02 AS INTEGER                        'IC02 Port-Funktion
    INIIC01 AS INTEGER                       'Port-Initialisierung IC01
    INIIC02 AS INTEGER                       'Port-Initialisierung IC02
END TYPE

'Initialisierung
'-----
    ADRIC01 = &HEB                          'IC01 ADR für Kontrolbyte
    ADRIC02 = &HDB                          'IC02 ADR für Kontrolbyte
    BETART = &H80                            'laut Datenblatt IC-82C55 Bytemodus
    LPT = &H378                              'ADR der PSS
    FOR HV01 = 1 TO 6
        BETMOD(HV01) = 1                    'alle Ports: Modus Eingabe
    NEXT HV01

'Bildschirmfarben festlegen
'-----
    COLOR 15, 1                              'VG: weiß; HG: blau
    CLS                                       'BS loeschen

'Menü
'-----
'Betriebsarteneinstellung der Ports (Ein- oder Ausgabe)
'-----

MSP:
'Statusbyte von IC01 und IC02
'    die vorgesehenen Betriebsarten der einzelnen Ports werden abgefragt
'    und daraus ein Steuerbyte für den jeweiligen 82C55 ermittelt

    INIIC01 = BETART OR (BETMOD(1) * 16) OR (BETMOD(2) * 2) OR (BETMOD(3) * 8) OR
BETMOD(3)
    INIIC02 = BETART OR (BETMOD(4) * 16) OR (BETMOD(5) * 2) OR (BETMOD(6) * 8) OR
BETMOD(6)

'Darstellung
'-----

```

' durch Eingabe einer jedem Port zugeordneten Ziffer (1-6) kann die Betriebsart von Ausgabe in Eingabe (bzw. umgekehrt) geschaltet werden

```
CLS
LOCATE 3, 16: PRINT "
LOCATE 4, 16: PRINT "
LOCATE 5, 16: PRINT "
LOCATE 6, 16: PRINT "
LOCATE 7, 16: PRINT "
|| "
|| "
|| "
LOCATE 8, 16: PRINT "
|| "
|| "
LOCATE 9, 16: PRINT "
|| "
LOCATE 10, 16: PRINT "
LOCATE 11, 16: PRINT "
" || "
" || "
" || "
LOCATE 12, 16: PRINT "
LOCATE 13, 16: PRINT "
" || "
LOCATE 14, 16: PRINT "
LOCATE 15, 16: PRINT "
LOCATE 16, 16: PRINT "
LOCATE 17, 16: PRINT "
LOCATE 18, 16: PRINT "
LOCATE 18, 29: PRINT "
LOCATE 18, 35: PRINT "
LOCATE 18, 44: PRINT "
LOCATE 18, 44: PRINT "
LOCATE 18, 49: PRINT "
LOCATE 19, 16: PRINT "
LOCATE 20, 19: PRINT "

```

'Umstellung

'-----

' Tastaturabfrage zur Umschaltung der Betriebsart der einzelnen Ports

MTA:

HS01\$ = UCASE\$(INKEY\$)

MSC:

```
SELECT CASE HS01$
CASE " "
GOTO MTA
CASE "1"
IF BETMOD(1) = 0 THEN BETMOD(1) = 1
BETMOD(1) = 0
CASE "2"
IF BETMOD(2) = 0 THEN BETMOD(2) = 1
BETMOD(2) = 0
CASE "3"
IF BETMOD(3) = 0 THEN BETMOD(3) = 1
BETMOD(3) = 0
CASE "4"
IF BETMOD(4) = 0 THEN BETMOD(4) = 1
BETMOD(4) = 0
CASE "5"
IF BETMOD(5) = 0 THEN BETMOD(5) = 1
BETMOD(5) = 0
CASE "6"
IF BETMOD(6) = 0 THEN BETMOD(6) = 1
BETMOD(6) = 0
CASE "I"
GOTO MINIT
CASE "Q"
GOTO ENDE
CASE ELSE
CLS
COLOR 4, 1
LOCATE 2, 2: PRINT "
LOCATE 3, 2: PRINT "

```

```

LOCATE 4, 2: PRINT "
LOCATE 5, 2: PRINT "
LOCATE 6, 2: PRINT "
LOCATE 7, 2: PRINT "
LOCATE 8, 2: PRINT "
LOCATE 9, 2: PRINT "
COLOR 15, 1

```

'VG: grün; HG: blau

```

LOCATE 3, 4: PRINT "      Falsche Eingabe!"
LOCATE 5, 4: PRINT "Gültige Optionen sind:"
LOCATE 6, 4: PRINT "1-6: Portumschaltung"
LOCATE 7, 6: PRINT "I: Initialisierung"
LOCATE 8, 6: PRINT "Q: Programmende"
LOCATE 10, 7: PRINT "Z - zurück zum Menü"

```

'Tastaturabfrage: zurück zum Menü

MTW:

```

HS01$ = UCASE$(INKEY$)
SELECT CASE HS01$
    CASE "Z"
        GOTO MSP
    CASE ELSE
        GOTO MTW
END SELECT

```

END SELECT

MPP:

```

LOCATE 1, 1: PRINT STRING$(42, " ")
LOCATE 2, 1: PRINT STRING$(42, " ")
LOCATE 22, 3: PRINT "Port"
LOCATE 22, 8: PRINT HS01$ + " für Statusumstellung gewählt"
GOTO MSP

```

MINIT:

'Reset am Modul MBS auslösen

'Kontrolbyte ausgeben (/RESET=0-1-0, Bit 2, Pin 16 der PSS)

```

OUT (LPT + 2), 0
OUT (LPT + 2), 4
COLOR 10, 1
CLS

```

'VG: weiß; HG: blau

```

PRINT : PRINT " Reset der Ports von IC01/02 wurde durchgeführt"
PRINT " -----"

```

```

PRINT " Basis-Adresse LPT ... : " + HEX$(LPT + 2) + "    Datenbyte Steuerregis-
ter: " + HEX$(0) + "    (/RESET = 0)"
PRINT " Basis-Adresse LPT ... : " + HEX$(LPT + 2) + "    Datenbyte Steuerregis-
ter: " + HEX$(4) + "    (/RESET = 1)"
PRINT : PRINT

```

'Ports initialisieren

'IC01

'Adressierung für Empfang Kontrolbyte (In- und Output Codierung)

'Kontrolbyte ausgeben

```

PRINT " Initialisierung der Ports für Aus- oder Eingabe laut Tabelle"
PRINT " -----"

```

```

PRINT " Ports von IC01 ..... : " + "    Adreßbyte: " + HEX$(ADRIC01) + "
Kontrolbyte: " + HEX$(INIIC01)
OUT (LPT + 3), ADRIC01
OUT (LPT + 4), INIIC01

```

'IC02

'Adressierung für Empfang Kontrolbyte (In- und Output Codierung)

'Kontrolbyte ausgeben

```

PRINT " Ports von IC02 ..... : " + "    Adreßbyte: " + HEX$(ADRIC02) + "
Kontrolbyte: " + HEX$(INIIC02)
PRINT
OUT (LPT + 3), ADRIC02
OUT (LPT + 4), INIIC02

```

'Abfrage, ob neue Initialisierung erforderlich

Initialisierung wiederholen? (J/N)
Q - Quit

MNA :

```

HS01$ = UCASE$(INKEY$)
SELECT CASE HS01$
    CASE "J"                                'Init wiederholen
        CLS : GOTO MSP
    CASE "N"                                'Init in Ordnung
        GOTO MDLS
    CASE "Q"                                'Programm beenden
        GOTO ENDE
    CASE ELSE
        GOTO MNA
END SELECT

```

'Menü Daten Lesen und Schreiben

```
'für Input : aktivierte Ports werden automatisch ausgelesen
'für Output: bei aktivierten Ports erfolgt die Ausgabe nach der Eingabe
'           eines Datenbytes
```

CLS

LOCATE 3, 8: PRINT

```

"||"
LOCATE 4, 8: PRINT "||"
||"
LOCATE 5, 8: PRINT
"||"
LOCATE 6, 8: PRINT "||" PN Port" + " Status" + " Datenwert" + "
Bit 7 6 5 4 3 2 1 0 ||"
LOCATE 7, 8: PRINT "||"
..... ||"
LOCATE 8, 8: PRINT "||" 1 IC01 Port A": LOCATE 8, 72: PRINT "||"
LOCATE 9, 8: PRINT "||" 2 IC01 Port B": LOCATE 9, 72: PRINT "||"
LOCATE 10, 8: PRINT "||" 3 IC01 Port C": LOCATE 10, 72: PRINT "||"
LOCATE 11, 8: PRINT "||" ": LOCATE 11, 72: PRINT "||"
LOCATE 12, 8: PRINT "||" 4 IC02 Port A": LOCATE 12, 72: PRINT "||"
LOCATE 13, 8: PRINT "||" 5 IC02 Port B": LOCATE 13, 72: PRINT "||"
LOCATE 14, 8: PRINT "||" 6 IC02 Port C": LOCATE 14, 72: PRINT "||"
LOCATE 15, 8: PRINT "||"
..... ||"
LOCATE 16, 8: PRINT "||" 7 benutzerdef. Eingang": LOCATE 16, 72: PRINT "||"
LOCATE 17, 8: PRINT
"||"
LOCATE 18, 8: PRINT "||" Für Datenein-/ ausgabe entsprechende Portnummer (PN)
wählen ||"
LOCATE 19, 8: PRINT
"||"
LOCATE 20, 8: PRINT "||" Q - Quit I - Initialisierung L - Anzeige lö-
schen": LOCATE 20, 72: PRINT "||"
LOCATE 21, 8: PRINT
"||"

```

HV02 = 0

```

FOR HV01 = 1 TO 6
  IF HV01 > 3 THEN HV02 = 1
  IF BETMOD(HV01) = 1 THEN
    LOCATE 7 + HV01 + HV02, 29: PRINT "Eingabe"
  ELSE
    LOCATE 7 + HV01 + HV02, 29: PRINT "Ausgabe"
  END IF
NEXT HV01

```

'Tastatur-Abfrage der Portnummer

MTB:

```
HS01$ = UCASE$(INKEY$)
SELECT CASE HS01$
CASE " "                                'Tastatur neu abfragen
    GOTO MTB
CASE "1"                                'IC01-Port A gewählt
    GOTO DAT
CASE "2"                                'IC01-Port B gewählt
    GOTO DAT
CASE "3"                                'IC01-Port C gewählt
    GOTO DAT
CASE "4"                                'IC02-Port A gewählt
    GOTO DAT
CASE "5"                                'IC02-Port B gewählt
    GOTO DAT
CASE "6"                                'IC02-Port C gewählt
    GOTO DAT
CASE "7"                                'benutzerdef. Eingänge abfragen
    GOTO MST
CASE "I"                                'neu initialisieren
    CLS : GOTO MSP
CASE "L"                                'Anzeige loeschen
    GOTO MDLS
CASE "Q"                                'Programm beenden
    GOTO ENDE
CASE ELSE
    GOTO MTB                                'Tastatur neu abfragen
END SELECT
```

DAT:

'Daten Lesen oder Schreiben

'Port und Status ermitteln

```
FOR HV03 = 1 TO 6
    IF MID$(STR$(HV03), 2) = HS01$ THEN
        IF BETMOD(HV03) = 0 THEN
            CALL Schreiben                'Datenbyte schreiben
        ELSE
            CALL Lesen                    'Datenbyte lesen
        END IF
    END IF
NEXT HV03
```

GOTO MTB 'Zurück zur Tastaturabfrage

'Daten der benutzerdefinierten Statusleitungen ermitteln

' Bit 3-5 des Eingangsregisters

MST:

CALL Status

GOTO MTB 'zurück zur Tastaturabfrage

ENDE:

END

' ***** Programmende *****

```

FUNCTION HIB$ (HEXWER AS INTEGER)
'***** Funktionsanfang *****

'wandelt eine Zahl (Hex-Format, 0-255) in das BIN-Format um
'Trennzeichen nach jedem Bit, Blocktrennung nach 4 Bits

'Deklarationen der Variablen
'-----
    DIM n, Digit AS INTEGER, Temp AS STRING

'Wandlung
'-----
    FOR n = 7 TO 0 STEP -1
        Digit = INT(HEXWER / 2 ^ n)
        Temp = Temp + STR$(Digit)
        HEXWER = HEXWER - Digit * 2 ^ n
    NEXT n

'Zeichenkette generieren
'-----
    HIB$ = LEFT$(Temp, 8) + " " + RIGHT$(Temp, 8)

'***** Funktionsende *****
END FUNCTION

```

SUB Lesen

'***** Routinenanfang *****

'Routine zum Lesen eines Datenbytes vom Port

'-----

'Adreßbyte für "Datenbyte einlesen" generieren und übertragen

' IC01-Port A = E4, IC01-Port B = E5, IC01-Port C = E6

' IC02-Port A = D4, IC02-Port B = D5, IC02-Port C = D6

IF HV03 < 4 THEN

OUT (LPT + 3), &HE3 + HV03 'Port von IC01 selektieren

ELSE

OUT (LPT + 3), &HD3 + HV03 - 3 'Port von IC02 selektieren

END IF

'Datenbyte einlesen

'-----

DATWER(HV03) = INP(LPT + 4) 'Leseoperation

HV02 = 0

FOR HV01 = 8 TO 13

'aktuelle Portmarkieerung loeschen

IF HV01 > 10 THEN HV02 = 1 'Zeilensprung bei Anzeige beachten

LOCATE HV01 + HV02, 13: PRINT " "

NEXT HV01

HV02 = 0

'Zeilensprung bei Anzeige beachten

IF HV03 > 3 THEN HV02 = 1

LOCATE HV03 + 7 + HV02, 13: PRINT "X"

LOCATE HV03 + 7 + HV02, 37: PRINT " "

LOCATE HV03 + 7 + HV02, 37: PRINT DATWER(HV03)

IF LEN(HEX\$(DATWER(HV03))) < 2 THEN

LOCATE HV03 + 7 + HV02, 43: PRINT "0" + HEX\$(DATWER(HV03)) + " HEX"

ELSE

LOCATE HV03 + 7 + HV02, 43: PRINT HEX\$(DATWER(HV03)) + " HEX"

END IF

LOCATE HV03 + 7 + HV02, 53: PRINT HIB(DATWER(HV03))

'***** Routinenende *****

END SUB

SUB Schreiben

'***** Routinenanfang *****'

'Routine zum Schreiben eines Datenbytes auf einen Port

'-----'

```
HV02 = 0
FOR HV01 = 8 TO 13
    IF HV01 > 10 THEN HV02 = 1      'aktuelle Portmarkierung loechen
    LOCATE HV01 + HV02, 13: PRINT " " 'Zeilensprung bei Anzeige beachten
NEXT HV01
```

```
HV02 = 0      'Zeilensprung bei Anzeige beachten
IF HV03 > 3 THEN HV02 = 1
```

```
LOCATE HV03 + 7 + HV02, 13: PRINT "X"
```

MEW:

```
LOCATE 22, 8: INPUT "Datenbyte für den Port eingeben (0-255 oder &H[HEX]): ",
```

HS01\$

```
IF LEN(HS01$) < 1 THEN
    LOCATE 22, 60: PRINT STRING$(20, " ")
    GOTO MEW
END IF
```

```
IF UCASE$(HS01$) = "Q" THEN
    LOCATE 22, 5: PRINT STRING$(70, " ")
    EXIT SUB      'Abbruch Sub
END IF
```

```
IF ASC(UCASE$(HS01$)) > 64 AND ASC(UCASE$(HS01$)) < 91 THEN
    BEEP
    LOCATE 22, 60: PRINT STRING$(20, " ")
    GOTO MEW
END IF
```

```
IF VAL(HS01$) < 0 OR VAL(HS01$) > 255 THEN
    BEEP
    LOCATE 22, 60: PRINT STRING$(20, " ")
    GOTO MEW
END IF
```

```
LOCATE 22, 5: PRINT STRING$(70, " ")
DATWER(HV03) = VAL(HS01$)      'Datenbyte eintragen
LOCATE HV03 + 7 + HV02, 37: PRINT " "
LOCATE HV03 + 7 + HV02, 37: PRINT DATWER(HV03)
```

```
IF LEN(HEX$(DATWER(HV03))) < 2 THEN
    LOCATE HV03 + 7 + HV02, 43: PRINT "0" + HEX$(DATWER(HV03)) + " HEX"
ELSE
    LOCATE HV03 + 7 + HV02, 43: PRINT HEX$(DATWER(HV03)) + " HEX"
END IF
```

```
HV01 = DATWER(HV03)
LOCATE HV03 + 7 + HV02, 53: PRINT HIB(DATWER(HV03))
DATWER(HV03) = HV01
```

'Daten ausgeben

'-----'

'Adreßbyte für "Datenbyte ausgeben" generieren und übertragen

```
'IC01-Port A = E8, IC01-Port B = E9, IC01-Port C = EA
'IC02-Port A = D8, IC02-Port B = D9, IC02-Port C = DA
```

```
IF HV03 < 4 THEN
    OUT (LPT + 3), &HE7 + HV03      'Port von IC01 selektieren
ELSE
    OUT (LPT + 3), &HD7 + HV03 - 3  'Port von IC02 selektieren
END IF
```

'Datenbyte ausgeben

'-----'

```
OUT (LPT + 4), DATWER(HV03)      'Schreiboperation
```

```
'***** Routinenende *****  
END SUB
```

SUB Status

'***** Routinenanfang *****'

'Routine zum Lesen der benutzerdefinierbaren Steuerleitungen

'-----'

'Eingang 1, 2 ,3 (Pin 12/15/13 der PSS) werden für die Anzeige herausgefiltert

'Datenbyte einlesen

STABYT = INP(LPT + 1)

BENEIN = STABYT AND &H38

LOCATE 16, 37: PRINT BENEIN

IF LEN(HEX\$(BENEIN)) < 2 THEN

LOCATE 16, 43: PRINT "0" + HEX\$(BENEIN) + " HEX"

ELSE

LOCATE 16, 43: PRINT HEX\$(BENEIN) + " HEX"

END IF

LOCATE 16, 53: PRINT HIB(BENEIN)

'***** Routinenende *****'

END SUB

Bildschirmdarstellungen

Menü: Auswahl der Betriebsart

Betriebsart der Ports			
Status: 1 Eingabe 0 Ausgabe			

1 -	IC01 Port A	Status	1
2 -	IC01 Port B	Status	1
3 -	IC01 Port C	Status	1
.....			
4 -	IC02 Port A	Status	1
5 -	IC02 Port B	Status	1
6 -	IC02 Port C	Status	1
.....			
I - Initialisierung			
Q - Quit			

DW-IC01: 9B		DW-IC02: 9B	

Portnummer entsprechend wählen

Menü: Fehlermeldung bei falscher Auswahl

Falsche Eingabe!

Gültige Optionen sind:
1-6: Portumschaltung
I: Initialisierung
Q: Programmende

Z - zurück zum Menü

Menü: Initialisierung wiederholen

Reset der Ports von IC01/02 wurde durchgeführt

Basis-Adresse LPT ... : 37A Datenbyte Steuerregister: 0 (/RESET = 0)
Basis-Adresse LPT ... : 37A Datenbyte Steuerregister: 4 (/RESET = 1)

Initialisierung der Ports fuer Aus- oder Eingabe laut Tabelle

Ports von IC01 : Adreßbyte: EB Kontrolbyte: 8B
Ports von IC02 : Adreßbyte: DB Kontrolbyte: 9B

Initialisierung wiederholen? (J/N)

Q - Quit

Menü: Datenübertragung (X hinter der PN kennzeichnet die letzte Auswahl)

Datenübertragung der Ports												
PN	Port	Status	Datenwert		Bit 7	6	5	4	3	2	1	0
1	IC01 Port A	Eingabe	155	9B HEX	1	0	0	1	1	0	1	1
2	IC01 Port B	Eingabe	155	9B HEX	1	0	0	1	1	0	1	1
3	IC01 Port C	Eingabe	155	9B HEX	1	0	0	1	1	0	1	1

4	IC02 Port A	Eingabe	155	9B HEX	1 0 0 1	1 0 1 1
5	IC02 Port B	Eingabe	155	9B HEX	1 0 0 1	1 0 1 1
6 X	IC02 Port C	Eingabe	155	9B HEX	1 0 0 1	1 0 1 1
.....						
7	benutzerdef.	Eingang	56	38 HEX	0 0 1 1	1 0 0 0

Für Datenein-/ ausgabe entsprechende Portnummer (PN) waehlen						

Q - Quit		I - Initialisierung		L - Anzeige loeschen		