ALLE IM PAKET "PID-EASY" ENTHALTENEN FUNKTIONS-BAUSTEINE:

SYMBOL	ADRESSE	KOMMENTAR
INIT	FC 127	FIRST_SCAN, UR_SCAN, Zeit-PULSE
MUL_MINUS1_INT	FC 130	INTEGER-WERT * (-1)
DATA_SWITCH	FC 140	Datenumschalter "INT"
MIN_MAX_ALARM	FC 141	MIN-/MAX-Alarm
MIN_MAX_ALARM_tV	FC 142	MIN-/MAX-Alarm + "tV"
MIN_MAX_ALARM_HY	FC 143	MIN- / MAX-Alarm mit HYSTERESE
MIN_MAX_ALARM_HY_tV	FC 144	MIN- / MAX-Alarm mit HYSTERESE + "tV"
P	FC 150	P-Regler
PID	FC 151	PID-Regler
ZUSATZSEQUENZ	FC 152	Zusatzsequenz 0-1000%; DW und UW
AH	FC 153	BuB: Stellgröße AUTO<-->HAND
AD/TAKTGEBER	FC 154	AD-Wandler: Analoginput -> 3-Punkt-Takt
AD/SM	FC 155	AD-Wandler/Stellmotor mit Endlagenerfassung
HYST_SWITCH	FC 156	Hystereseschalter=2-Punkt-Regler
HYST_SWITCH_tV	FC 157	Hystereseschalter=2-Punkt-Regler mit "tV"
3PUNKT	FC 158	Dreipunktregler
SSW	FC 159	Stufen- / Schrittschaltwerk, n-Stufen n{0,1,2,...16}
NORMSIGNAL->ANALOG_OUT	FC 164	Wandlung: 0-1000 % -> Analogausgangssignal
SKAL_LINEAR+LIMIT_INT	FC 165	Lineare Skalierung + Begrenzung INTEGER
SKAL_QUADRAT+LIMIT_INT	FC 166	Quadratische Skalierung + Begrenzung INTEGER
SKAL_WURZEL2+LIMIT_INT	FC 167	Quadratwurzel Skalierung + Begrenzung INTEGER
SKAL_EXPONENT+LIMIT_INT	FC 168	Exponential-Skalierung + Begrenzung INTEGER
SKAL_LOGARITH+LIMIT_INT	FC 169	Logarithmische Skalierung + Begrenzung INTEGER
RAMP_INT	FC 172	Die INT-Variable "y" wird an den INT-Wert "x" mit Rampe geführt
DAMP_INT	FC 173	Dämpfung eines INTEGERSIGNALES durch Mittelwertbildung
MIN_MAX_LIMIT	FC 174	Variable begrenzen auf OG/UG
MIN_MAX_LIMIT+TOTBAND	FC 175	MIN-/MAX-Limit + Totband für eine Variable
AVERAGE_2	FC 176	Durchschnitt aus 2 INT-Werten
AVERAGE_2-8	FC 177	Durchschnitt aus 2-8 INT-Werten
MIN_MAX_SELEKT_2	FC 178	Minimum + Maximum aus 2 INT-Werten
MIN_MAX_SELEKT_2-8	FC 179	Minimum + Maximum aus 2-8 INT-Werten
SWG_TIME	FC 180	Sollwertgeber mit Zeitplanführung
SWG_INT	FC 181	Sollwertgeber mit Führung durch INTEGERWERT

UDT's MIT STRUKTUREN, DIE IN DEN "DBxxx" ABGELEGT WERDEN:

SYMBOL	ADRESSE	KOMMENTAR
UDT_MIN_MAX_ALARM	UDT141	Speicherbereich für MIN-/MAX-Grenzwertüberwachung
UDT_MIN_MAX_ALARM_tV	UDT142	Speicherbereich für MIN-/MAX-Grenzwertüberwachung+tV
UDT_MIN_MAX_ALARM_HY	UDT143	Speicherbereich für MIN-/MAX-Grenzwertüberwachung+Hysterese
UDT_MIN_MAX_ALARM_HY_tV	UDT144	Speicherbereich für MIN-/MAX-Grenzwertüberwachung+Hysterese+tV
UDT_P	UDT150	P-Regler-Daten
UDT_PID	UDT151	PID-Regler-Daten
UDT_AH	UDT153	Speicher für AUTO-HAND-Umschaltung
UDT_TKT	UDT154	Takt-Regler-Daten
UDT_SM	UDT155	SM-Daten
UDT_HY	UDT156	Hystereseschalterdaten
UDT_HY_tV	UDT157	Hystereseschalterdaten für Hystereseschalter mit "tV"
UDT_3PUNKT	UDT158	3-Punkt-Regler Daten
UDT_SSW	UDT159	Schrittschaltwerk-Daten
UDT_RAMP_INT	UDT172	Daten + Speicher für INTEGER-RAMPE
UDT_DAMP_INT	UDT173	Daten-FIFO: Dämpfung eines INTEGERSIGNALES durch Mittelwertbildung
UDT_SWG_TIME	UDT180	Daten für Sollwertgeber mit Zeitplanführung
UDT_SWG_INT	UDT181	Daten für Sollwertgeber mit Führung durch INTEGERWERT

VORSCHLAG FÜR ADRESSEN WICHTIGER MERKER+TIMER:

ADRESSE	SYMBOL	KOMMENTAR
M 0.0	PERM_FALSE	Permanent:=0, für FUP- + KOP-Programme
M 0.1	PERM_TRUE	Permanent:=1, für FUP- + KOP-Programme
M 1.0	UR_SCAN	Im 1.OB1-Zyklus:=1, nach OB100/101+Remanenzverlust
M 1.1	FIRST_SCAN	Im 1.OB1-Zyklus:=1, stets nach OB100/101
M 1.2	START_tv	Nach OB100/101/102 mit "tV_START":=1
M 2.0	PLS_100ms	PULS aller 100ms
M 2.1	PLS_250ms	PULS aller 250ms
M 2.2	PLS_500ms	PULS aller 500ms
M 2.3	PLS1_1000ms	PULS aller 1000ms mit Versatz von 500ms zu "PLS2_1000ms"
M 2.4	PLS2_1000ms	PULS aller 1000ms mit Versatz von 500ms zu "PLS1_1000ms"
MB 3	TAKT_MERKER_BYTE	Hardwarekonfiguration = CPU-Taktmerker
T 0	PERMANENT_TIMER	TIMER, der im FC "INIT" permanent neu gestartet wird

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
FC127, INIT: UR_SCAN/FIRST_SCAN/WAIT/START_tV NACH FIRST_SCAN/PULSE
################################################################################
Kurzbeschreibung:
Der FC "INIT" aktiviert das für die Initialisierung der meisten Bausteine der
PID-Familie benötigte "FIRST_SCAN"-FLAG. Nach Datenverlust, Urlöschen oder einem
Programm-Erst-Start wird ein "UR_SCAN"-FLAG gesetzt. Aus den CPU-Taktmerkern
werden einige wichtige Zeitpulse gebildet. Neben den Ausgängen "PERMANENT =TRUE
und =FALSE" und einem "PERMANENTEN TIMER", wird jedesmal im ersten OB1-Zyklus
eine Verzögerungszeit gestartet und nach Ablauf dieser Zeit eine Speicherstelle
auf TRUE gesetzt.
Über den FC "INIT" kann in allen Anlauf OB's eine Wartezeit bis max 99.9s pro-
grammiert werden. (Die Zykluszeitüberwachung ist in diesen OB's unwirksam!)
Diese Wartezeit verhindert nach Netz "AUS-EIN" Peripheriezugriffsfehler, die
dadurch entstehen, daß dezentrale Peripherie oder Erweiterungseinheiten zeitver-
zögert an das Netz gehen.
################################################################################
!!Dieser FC verändert nicht die Adressregister AR1/AR2 UND versorgt für FUP- UND
  KOP-Anwendungen die "ENO"-Box, indem das VKE=1 im BIE-BIT gespeichert wird.
!!Der FC "INIT" muß jeweils die 1.Anweisung im OB1, OB100, OB101 und OB102 sein.
  Er ist STETS IM OB1 AUFZURUFEN, sein Aufruf in den OB'S 100 bis 102 ist nicht
  notwendig. Der Aufruf in den Anlauf-OB's OB100, OB101 und OB102 beeinflußt
  nicht die Funktion des FC "INIT" im OB1. Der FC "INIT" erkennt intern, in wel-
  chem OB er aufgerufen wird, denn er wertet das 4. LOKALDATENBYTE aller OB's
  und das 2.LOKALDATENBYTE des OB1 aus. Der Aufruf des FC "INIT" muß deshalb
  direkt im OB1, OB100, OB101 oder OB102 erfolgen.
!!Erfolgt der Aufruf des FC "INIT" aus einem anderen Baustein als aus den o.g.
  Organisationsbausteinen, so führt dies zu unkontrollierbaren Fehlfunktionen im
  gesamten Programm.
!!Die Ausgänge "UR_SCAN" + "FIRST_SCAN" + "PERM_FALSE" + "PERM_TRUE" sind
  bereits im OB100/OB101/OB102 verfügbar!
!!Im FC "INIT" wird der SFC22 "CREAT_DB" aufgerufen.
!!Wird der FC in einem anderen OB als im OB1, OB100, OB 101, OB102, aufgerufen
  ODER der DB konnte nicht erzeugt werden ODER ist im Programm mit einer
  Länge <> 6 BYTE vorhanden, so liegt ein Fehler vor. Dann werden lediglich die
  Ausgänge "PERM_FALSE" und "PERM_TRUE" aktiviert. Alle anderen Ausgänge:=0 UND
  das BIE-BIT:=0.

in:
DB_NR   [INT]: Beliebige Datenbausteinnummer  DB1,...,DBxxx (xxx=CPU-abhängig!)
!!Dieser DB wird mit dem SFC22="CREAT DB" mit der Länge von 6BYTE erzeugt und
  darf im Anwenderprogramm nicht bereits enthalten sein. Bei Remanenzverlust
  wird dieser "DB" neu erzeugt. Wenn der Rückgabewert "RET_VAL"=0 ist - nur dann
  wurde der "DB" neu erzeugt - wird das "UR-SCAN"-Flag gesetzt.
  Die 6BYTES des "DB" werden FC intern wie folgt genutzt und dürfen vom Anwender
  nicht beschrieben werden:
     BYTE 0   = 8 BITS mit folgender Bedeutung
                  BIT0  : Speicher für UR_SCAN
                  BIT1  : Frei (Reserve)
                  BIT2  : FP-Speicher für PULS  100ms
                  BIT3  : FN-Speicher für PULS  100ms
                  BIT4  : FP-Speicher für PULS  250ms
                  BIT5  : FN-Speicher für PULS  250ms
                  BIT6  : FP-Speicher für PULS  500ms
                  BIT7  : FN-Speicher für PULS  500ms
     BYTE 1   = Frei (Reserve)
     BYTE 2+3 = Speicher für den Altwert des TIMERS
     BYTE 4+5 = Speicher für den tV_START-TIMER
!!IST DER DB MIT DER NUMMER "DB_NR" IM PROGRAMM BEREITS ENTHALTEN UND WIRD VOM
  ANWENDER ANDERWEITIG BENUTZT, DANN TRETEN FEHLFUNKTIONEN DES FC-INIT AUF.
PERM_TIMER [TIMER]: Beliebiger TIMER       T0,...,Txxx (xxx=CPU-abhängig!)
!!Dieser TIMER wird nach folgenden Kriterien permanent gestartet:
  Im ERSTEN OB1-Zyklus nach OB100/101/102 mit der Zeit "t0=800*[10ms]". Der
  Aktualwert des TIMERS="tAKT" wird in jedem Zyklus abgefragt. Der "PERM_TIMER"
  wird stets dann mit dem WERT "tNEU=(400+tAKT)*[10ms]" neu gestartet, wenn der
  Aktualwert des TIMERS "tAKT<400*[10ms]" ist.
  Da die TIMER im Interrupt bearbeitet werden, haben diese an verschiedenen
  Stellen des OB1-Zyklusses verschiedene Werte. Wird nun mit "FIRST-SCAN=1" eine
  beliebige Speicherstelle für den ALTWERT des TIMERS mit "tALT=800*[10ms]"
  initialisiert, dann stellt die die Differenz "tALT-tAKT" die Zeit "DELTA_t"
  dar, die an der "MEßSTELLE" zwischen 2-CPU-Zyklen vergangen ist. Wird das
  "DELTA_t" aufsummiert, dann kann man mit einer Genauigkeit von ca 0,1% belie-
```

```
  big viele Zeiten im RASTER von "10ms" bilden.
     Folgendes Programm muß dafür geschrieben werden:
              L  t_ALT       //SPEICHER für tALT          ZYKLUS n-1
              L  PERM_TIMER  //Aktualwert   TIMER=tAKT    ZYKLUS n
              T   t_ALT      //SPEICHER für tALT:=tAKT    ZYKLUS n
              -I             //AKKU1:=(tALT-tAKT)
              SPPZ POSI
              L  400         //Ist die Differenz negativ, dann "+400"
              +I
        POSI: T DELTA_t      //DELTA_t:=(tALT-tAKT)
!!Das beschriebene Programm gilt nur für CPU-Zykluszeiten "tZYK<=400*10ms=4s".
t_WAIT    [S5TIME]: Wartezeit im OB100, OB101, OB102    {0.00,0.01,...,99.9}*s
                    Nach STOP-RUN wird in Anlauf-OB's der FC "INIT" erst nach
                    der programmierten Wartezeit "t_WAIT" durchlaufen. Dadurch
                    verlängert sich die Bearbeitungszeit der Anlauf OB's um min-
                    destens die Zeit "t_WAIT".
                    Damit können nach Netz AUS-EIN Peripheriezugriffsfehler zur
                    dezentralen Peripherie und zu Erweiterungseinheiten verhind-
                    dert werden.
!!Wenn Zeiteingabe fehlerhaft: tWAIT > 99.9s -> Korrektur: t_WAIT:=99.9s
tV_START  [S5TIME]: Verzögerungszeit nach FIRST_SCAN    {0.00,0.01,...,99.9}*s
                    Diese Zeit wird im ERSTEN OB1-Zyklus stets neu gestartet.
!!Wenn Zeiteingabe fehlerhaft: tV_START > 99.9s -> Korrektur: tV_START:=99.9s

TAKT     [POINTER]: Pointer auf die Adresse des CPU-Taktmerker-Bytes
!!Jedem Bit des Taktmerkerbytes ist eine Periodendauer/Frequenz zugeordnet:
           BIT         |    Frequenz  |           Periodendauer
          -------------------------------------------------------------------
           BIT0 = TAKT1 |    10,00 Hz  |  100ms ->   50ms=AUS +   50ms=EIN
           BIT1 = TAKT2 |     5,00 Hz  |  200ms ->  100ms=AUS +  100ms=EIN
           BIT2 = TAKT3 |     2,50 Hz  |  400ms ->  200ms=AUS +  200ms=EIN
           BIT3 = TAKT4 |     2,00 Hz  |  500ms ->  250ms=AUS +  250ms=EIN
           BIT4 = TAKT5 |     1,25 Hz  |  800ms ->  400ms=AUS +  400ms=EIN
           BIT5 = TAKT6 |     1,00 Hz  | 1000ms ->  500ms=AUS +  500ms=EIN
           BIT6 = TAKT7 |     0,625Hz  | 1600ms ->  800ms=AUS +  800ms=EIN
           BIT7 = TAKT8 |     0,50 Hz  | 2000ms -> 1000ms=AUS + 1000ms=EIN

out:
PERM_FALSE[BOOL]: FÜR KOP-/FUP-ANWENDUNGEN PERMANENT:= FALSE -> 0
PERM_TRUE [BOOL]: FÜR KOP-/FUP-ANWENDUNGEN PERMANENT:= TRUE  -> 1
UR_SCAN   [BOOL]: =1 ->  im 1.Zyklus nach OB100/101/102 im OB1 high=PULS, aber
                  nur, wenn
                  - ein Programm zum 1.Mal gestartet wird
                  - ein Programm nach Urlöschen neu geladen wird
                  - bei Remanenzverlust
FIRST_SCAN[BOOL]: =1 ->  stets im 1.Zyklus nach OB100/101/102 im OB1 high=PULS
START_tV  [BOOL]: =1 nach FIRST SCAN UND wenn der TIMER "tV_START" = high ist
!!Diese Speicherstelle kann für zeitverzögerte Freigaben nach  OB100/101/102
  verwendet werden.

PLS100ms   [BOOL]: PULS:=1, aller  100ms
PLS250ms   [BOOL]: PULS:=1, aller  250ms
PLS500ms   [BOOL]: PULS:=1, aller  500ms
PLS1_1000ms[BOOL]: PULS:=1, aller 1000ms, um 500ms versetzt zu PLS2_1000ms
PLS2_1000ms[BOOL]: PULS:=1, aller 1000ms, um 500ms versetzt zu PLS1_1000ms

!!Diese Pulse sind nur brauchbar, wenn die CPU-Zykluszeit "t_ZYK" kleiner ist,
  als der Zeitabstand zwischen 2 Pulsen. Die maximale Zykluszeit muß jedoch
  im Bereich "t_ZYK<500ms" liegen, wenn der 500ms-PULS und die 1000ms-PULSE
  benutzt werden sollen.
!!PLS1_1000ms UND PLS2_1000ms sind um 500ms zueinander versetzt:

        PLS1_1000ms="|"
   ^
   |
  1+     |           |           |           |           |
   |     |           |           |           |           |
   |     |           |           |           |           |
   +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+----->
   0    500   1000  1500  2000  2500  3000  3500  4000  4500  5000  t[ms]

        PLS2_1000ms="!"
```

```
   ^
   |
  1+          !          !          !          !          !
   |          !          !          !          !          !
   |          !          !          !          !          !
   +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+---->
   0    500   1000  1500  2000  2500  3000  3500  4000  4500  5000  t[ms]
```

Diese versetzten Pulse können in KOP-Anwendungen dann benutzt werden, wenn garantiert werden muß, daß 2 Schaltvorgänge einen zeitlichen Abstand voneinander haben müssen. Z.B. wird mit "E_AZ" eine Klappe "AUF" und "ZU" gefahren. Am Klappenstellmotor darf das Signal "AUF" und "ZU" nie gleichzeitig anliegen. Da dieser Vorgang nicht zeitkritisch ist, kann folgende KOP-Schaltung benutzt werden:

```
NETZWERK: Motorklappe "AUF"
  |      E_AZ    PLS1_1000ms       ZU         AUF
  +------| |-------| |------+------|/|--------( )-|
  |      AUF     PLS1_1000ms|
  +------| |-------|/|------+
  |
NETZWERK: Motorklappe "ZU"
  |      E_AZ    PLS2_1000ms       AUF        ZU
  +------|/|-------| |------+------|/|--------( )-|
  |      ZU      PLS2_1000ms|
  +------| |-------|/|------+
  |
```

Diese Schaltung garantiert, daß die CPU-Ausgänge "AUF" UND "ZU" bei jedem Umschaltvorgang mindestens 500ms lang gleichzeitig "AUS" sind und dann erst der geforderte Ausgang "EINSCHALTEN" kann.

!!
FC130, MUL_MINUS1_INT : MULTIPLIKATION MIT "(-1)"
##
Kurzbeschreibung:
Der FC "MUL_MINUS1_INT" multipliziert einen INTEGERWERT mit "-1". Dabei wird das Produkt (-1)*(-32768) auf den WERT =+32767 gesetzt. Diese Multiplikation wird zur Invertierung von Signalen benötigt.
##
!!Dieser FC verändert nicht die Adressregister AR1/AR2 UND versorgt für FUP- UND KOP-Anwendungen die "ENO"-Box, indem das VKE=1 im BIE-BIT gespeichert wird.

```
in:
x [INT]: Eingangssignal                            {-32768,...,0,...+32767}

out:
y [INT]: Ausgangssignal = x*(-1)                   {+32767,...,0,...-32767}
```

!!! Da die INTEGER-Multiplikation von (-32768)*(-1)= -32768 ist, wird FC-intern
!!! der negative Wert "x=-32768" auf "x:=-32767" gesetzt.

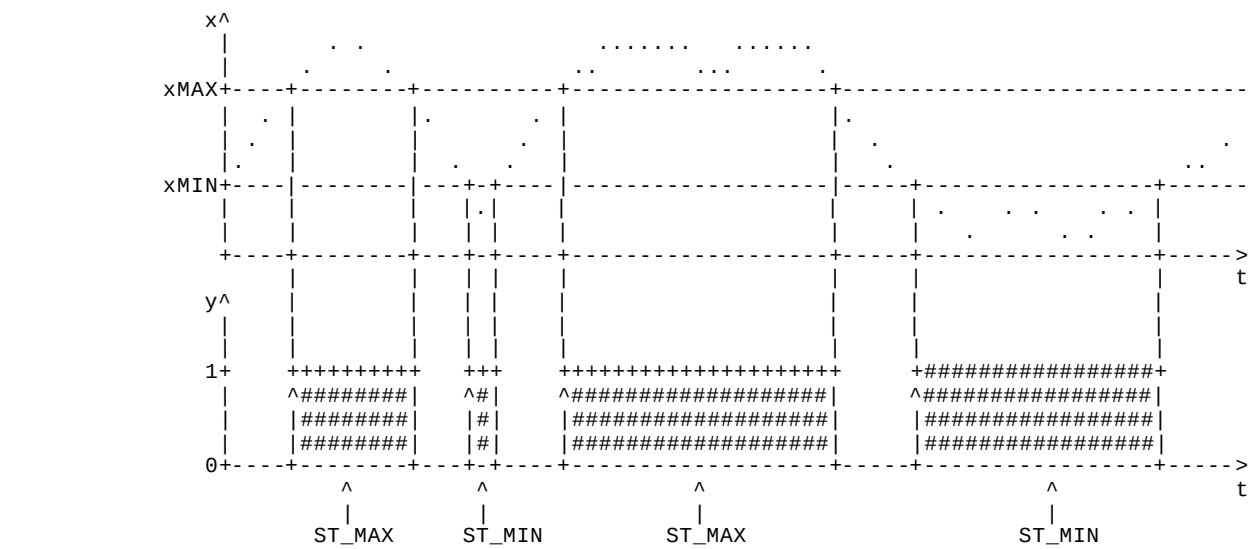
!!
FC140, DATA_SWITCH : DATA-SWITCH für INT-WERTE
##
Kurzbeschreibung:
Mit dem FC "DATA_SWITCH_INT" ist es möglich, abhängig vom Zustand einer binären Speicherstelle einen von 2-Integerwerten auszuwählen.
##
!!Dieser FC verändert nicht die Adressregister AR1/AR2 UND versorgt für FUP- UND KOP-Anwendungen die "ENO"-Box, indem das VKE=1 im BIE-BIT gespeichert wird.

```
in:
SW        [BOOL]: =0 -> y:=x0 ODER =1 -> y:=x1
x0         [INT]: Integerwert                        {-32768,...,0,...,+32768}
x1         [INT]: Integerwert                        {-32768,...,0,...,+32768}
out:
y          [INT]: x0 ODER x1
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
FC141, MIN_MAX_ALARM : MIN/MAX ALARM
################################################################################
Kurzbeschreibung:
Der FC "MIN_MAX_ALARM" aktiviert jeweils einen MIN- UND MAX-Alarm, wenn eine
INTEGER-Variable frei parametrierbare MIN-/MAX-Grenzwerte erreicht bzw. unter-
oder überschreitet. Gleichzeitig wird ein ALARM-NEUWERT generiert.
Die Parameter des FC "MIN_MAX_ALARM" müssen in einen Datenbausteinbereich ein-
gegeben werden, der durch den "UDT_MIN_MAX_ALARM" definiert wird. Sie lassen
sich vom BuB-System verändern und alle Zustände der Ein- und Ausgänge des FC
sind aus diesem DB-Bereich lesbar.
Die Quittierung der Alarme ist nach 2 MODI möglich: Der Alarm geht nur, wenn
er quittiert wurde und die Störungsursache nicht mehr anliegt ODER der Alarm
geht sofort, ohne Quittung, nach Beseitigung der Störungsursache.
Die OUTPUTS des FC "MIN_MAX_ALARM" können auch aus einer Universalschnittstel-
le, dem INFO-BYTE, gelesen werden.
################################################################################
!!Dieser FC verändert nicht die Adressregister AR1/AR2 UND versorgt für FUP- UND
  KOP-Anwendungen die "ENO"-Box, indem das VKE=1 im BIE-BIT gespeichert wird.

in:
QUITT       [BOOL]: =0->1 die "STÖRUNG" wird mit internem PULS quittiert,
                    liegt aber solange an, bis die Ursache beseitigt ist. STÖ-
                    RUNGSURSACHEN (Siehe MODE!) gelten als beseitigt, wenn die
                    zu überwachende Variable wieder im zulässigen Wertebereich
                    liegt.
                      MODE=0: Störungen gehen nur, wenn sie vorher quittiert
                              wurden und die Störungsursache nicht mehr anliegt.
                      MODE=1: Die Störungen gehen ohne Quittierung sofort dann,
                              wenn die Störungsursache beseitigt ist. Eine Neu-
                              wertmeldung wird nicht generiert. Der Eingang
                              "QUITT" ist wirkungslos.
PRV12BYTE[POINTER]: Pointer auf die Adresse eines  12 BYTE langen DB-Speicher-
                    bereiches, dessen Struktur im "UDT_MIN_MAX_ALARM" festge-
                    legt ist und der für jeden FC "MIN_MAX_ALARM" separat im
                    Datenbaustein "DB_MIN_MAX_ALARM" vorhanden sein muß.
x            [INT]: Zu überwachende Integervariable {-32768,...,0,...,+32767}
                      x<= xMIN -> MINIMUM-ALARM
                      x>= xMAX -> MAXIMUM-ALARM
out:
ST_MIN      [BOOL]: =1 -> der MINIMUM-ALARM wurde als Störung gespeichert
ST_MAX      [BOOL]: =1 -> der MAXIMUM-ALARM wurde als Störung gespeichert
ST_NW       [BOOL]: =1 -> die Störungen sind noch unquittiert - NEUWERT
SST         [BOOL]: =1 -> SAMMELSTÖRUNG = MIN- ODER MAX-ALARM
INFO       [BYTE]: Informationsbyte mit dem BIT-Muster "0sq0_0gk0", wobei die
                    BITS folgende Bedeutung haben:
                      BIT 0   = 0 : ohne Bedeutung
                      BIT 1   = k : =1 -> Störung MINIMUM-ALARM
                      BIT 2   = g : =1 -> Störung MAXIMUM-ALARM
                      BIT 3+4 = 0 : ohne Bedeutung
                      BIT 5   = q : =1 -> Störung=NEUWERT, nach QUITTIERUNG:=0
                      BIT 6   = s : =1 -> SAMMELSTÖRUNG, ein ALARM ist high
                      BIT 7   = 0 : ohne Bedeutung
!!Das "q"-BIT wird nach Betätigung der Quittiertaste stets auf  "=0" gesetzt,
!!auch, wenn die Störung noch anliegt. Bei MODE=1 ist das "q"-BIT permanent "=0"
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

################################################################################
Im Datenbaustein "DB_MIN_MAX_ALARM" müssen folgende Parameter eingegeben oder
können gelesen werden:
Alle mit "[W]"     gekennzeichneten Parameter sind zwingend einzugeben
Die  mit "[WO]"    gekennzeichneten Parameter sind optional einzugeben
Alle mit "[R]"     gekennzeichneten Parameter können nur gelesen werden
Alle mit "[I]/[O]" gekennzeichneten Parameter sind identisch mit In- und Outputs
                   des zugeordneten FCxxx und können nur gelesen werden
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
MODE   [BOOL]: MODE=0: Störungen gehen nur, wenn sie vorher quittiert wurden
                       und die Störungsursache nicht mehr anliegt.
               MODE=1: Die Störungen gehen ohne Quittierung sofort dann, wenn
                       die Störungsursache beseitigt ist. Eine Neuwertmeldung
                       wird nicht generiert. Der Eingang QUITT ist wirkungslos.
```

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
xMIN    [INT]: GRENZWERT FÜR MINIMUM-ALARM         {-32768,...,0,...,+32767}
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
xMAX    [INT]: GRENZWERT FÜR MAXIMUM-ALARM         {-32768,...,0,...,+32767}
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

Beispiel:




```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
FC142, MIN_MAX_ALARM_tV : MIN/MAX ALARM + tV
################################################################################
Kurzbeschreibung:
Der FC "MIN_MAX_ALARM_tv" aktiviert jeweils einen MIN- UND MAX-Alarm, wenn eine
INTEGER-Variable frei parametrierbare MIN-/MAX-Grenzwerte erreicht bzw. unter-
oder überschreitet UND diese Grenzwertverletzung permanent bis zum Ablauf einer
Verzögerungszeit anliegt. Gleichzeitig wird ein ALARM-NEUWERT generiert.
Die Parameter des FC "MIN_MAX_ALARM_tV" müssen in einen Datenbausteinbereich
eingegeben werden, der durch den "UDT_MIN_MAX_ALARM_tV" definiert wird. Sie las-
sen sich vom BuB-System verändern und alle Zustände der Ein- und Ausgänge des
FC sind aus diesem DB-Bereich lesbar.
Die Quittierung der Alarme ist nach 2 MODI möglich: Der Alarm geht nur, wenn
er quittiert wurde und die Störungsursache nicht mehr anliegt ODER der Alarm
geht sofort, ohne Quittung, nach Beseitigung der Störungsursache.
Die OUTPUTS des FC "MIN_MAX_ALARM_tV" können auch aus einer Universalschnitt-
stelle, dem INFO-BYTE, gelesen werden.
################################################################################
!!Dieser FC verändert nicht die Adressregister AR1/AR2 UND versorgt für FUP- UND
  KOP-Anwendungen die "ENO"-Box, indem das VKE=1 im BIE-BIT gespeichert wird.

in:
FIRST_SCAN  [BOOL]: Im 1.OB1-ZYKLUS nach OB100/101/102=TRUE, sonst FALSE
                    FIRST_SCAN wird zur Initialisierung unbedingt benötigt!
QUITT       [BOOL]: =0->1 die "STÖRUNG" wird mit internem PULS quittiert,
                    liegt aber solange an, bis die Ursache beseitigt ist. STÖ-
                    RUNGSURSACHEN (Siehe MODE!) gelten als beseitigt, wenn die
                    zu überwachende Variable wieder im zulässigen Wertebereich
                    liegt.
                      MODE=0: Störungen gehen nur, wenn sie vorher quittiert
                              wurden und die Störungsursache nicht mehr anliegt.
                      MODE=1: Die Störungen gehen ohne Quittierung sofort dann,
                              wenn die Störungsursache beseitigt ist. Eine Neu-
                              wertmeldung wird nicht generiert. Der Eingang
                              "QUITT" ist wirkungslos.
PRV24BYTE[POINTER]: Pointer auf die Adresse eines  24 BYTE langen DB-Speicher-
                    bereiches, dessen Struktur im "UDT_MIN_MAX_ALARM_tV" fest-
                    gelegt ist und der für jeden FC "MIN_MAX_ALARM_tV" separat
                    im Datenbaustein "DB_MIN_MAX_ALARM_tV" vorhanden sein muß.
```

```
x            [INT]: Zu überwachende Integervariable {-32768,...,0,...,+32767}
                    x<= xMIN UND tV=HIGH -> MINIMUM-ALARM
                    x>= xMAX UND tV=HIGH -> MAXIMUM-ALARM
out:
ST_MIN      [BOOL]: =1 -> der MINIMUM-ALARM wurde als Störung gespeichert
ST_MAX      [BOOL]: =1 -> der MAXIMUM-ALARM wurde als Störung gespeichert
ST_NW       [BOOL]: =1 -> die Störungen sind noch unquittiert - NEUWERT
SST         [BOOL]: =1 -> SAMMELSTÖRUNG = MIN- ODER MAX-ALARM
INFO       [BYTE]: Informationsbyte mit dem BIT-Muster "0sq0_0gk0", wobei die
                   BITS folgende Bedeutung haben:
                     BIT 0   = 0 : ohne Bedeutung
                     BIT 1   = k : =1 -> Störung MINIMUM-ALARM
                     BIT 2   = g : =1 -> Störung MAXIMUM-ALARM
                     BIT 3+4 = 0 : ohne Bedeutung
                     BIT 5   = q : =1 -> Störung=NEUWERT, nach QUITTIERUNG:=0
                     BIT 6   = s : =1 -> SAMMELSTÖRUNG, ein ALARM ist high
                     BIT 7   = 0 : ohne Bedeutung
!!Das "q"-BIT wird nach Betätigung der Quittiertaste stets auf  "=0" gesetzt,
!!auch, wenn die Störung noch anliegt. Bei MODE=1 ist das "q"-BIT permanent "=0"
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

################################################################################
Im Datenbaustein "DB_MIN_MAX_ALARM_tV" müssen folgende Parameter eingegeben
oder können gelesen werden:
Alle mit "[W]"     gekennzeichneten Parameter sind zwingend einzugeben
Die  mit "[WO]"    gekennzeichneten Parameter sind optional einzugeben
Alle mit "[R]"     gekennzeichneten Parameter können nur gelesen werden
Alle mit "[I]/[O]" gekennzeichneten Parameter sind identisch mit In- und Outputs
                   des zugeordneten FCxxx und können nur gelesen werden
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
MODE   [BOOL]: MODE=0: Störungen gehen nur, wenn sie vorher quittiert wurden
                       und die Störungsursache nicht mehr anliegt.
               MODE=1: Die Störungen gehen ohne Quittierung sofort dann, wenn
                       die Störungsursache beseitigt ist. Eine Neuwertmeldung
                       wird nicht generiert. Der Eingang QUITT ist wirkungslos.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
xMIN    [INT]: GRENZWERT FÜR MINIMUM-ALARM          {-32768,...,0,...,+32767}
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
xMAX    [INT]: GRENZWERT FÜR MAXIMUM-ALARM          {-32768,...,0,...,+32767}
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
tV_ST  [TIME]: VERZÖGERUNGSZEIT             {0,1ms,...,max=24d20h31m23s647ms}
               LIEGT DER ZU ÜBERWACHENDE INTEGERWERT IM ALARMBEREICH, DANN
               WIRD DIE ÜBERWACHUNSZEIT GESTARTET. EIN ALARM WIRD ERST DANN
               GENERIERT, WENN DIE ZEIT ABGELAUFEN IST UND DIE GRENZWERTVER-
               LETZUNG WEITERHIN ANLIEGT.
               !!IST xMIN >= xMAX , DANN LIEGT FÜR JEDEN WERT "x" EIN MIN-
                 UND/ODER MAX-ALARM AN.IN DIESEM FALL WIRD DER TIMER SOFORT
                 NACH FIRST_SCAN GESTARTET.
               !!DER TIMER WIRD BEI JEDEM WECHSEL VON "x<=xMIN <--> x>=xMAX"
                 NEU GESTARTET.
               !!BEI FEHLEINGABE tV_ST<0 -> tV_ST:=0
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

Beispiel:

```
   x^
    |        . .              .......   ......
    |     .     .           ..       ...      .
xMAX+----+--------+--------+-------------------+------------------------------
    |  . |        |.      .|                   |.
    | .  |        |  . ... |                   |  .                         .
    |.   |        |        |                   |   .                     ..
xMIN+----|--------|--------|-------------------|-----+-----------------+------
    |    |        |        |                   |     | .    . .    . . |
    |    |        |        |                   |     |   .      . .    |
    +----+--------+--------+-------------------+-----+-----------------+----->
         |        |        |                   |     |                 |     t
TIMER    |        |        |                   |     |                 |
    ^    |        |        |                   |     |                 |
    |    |        |        |                   |     |                 |
high+    +--------+>>+     +-----------+++++++++++++++------------++++++++++
    |    |        |  |     |           |#######|     |            |####|
    |    |        |  |     |           |#######|     |            |####|
    |    |        |  |     |           |#######|     |            |####|
   0+----+--------+--+-----+-----------+-------+-----+------------+----+----->
         |    tV     |     |    tV     |       |     |     tV     |    |     t
         +<--------->+     +<--------->+       |     +<---------->+    |
         |                 |           |       |                  |    |
         |                 |           |       |                  |    |
   y     |                 |           |       |                  |    |
    ^    |                 |           |       |                  |    |
    |    |                 |           |       |                  |    |
   1+    |                 |           +++++++++                  ++++++
    |    |                 |           ^#######|                  ^####|
    |    |                 |           |#######|                  |####|
    |    |                 |           |#######|                  |####|
   0+----+-----------------+-----------+-------+------------------+----+---->
                                           ^                         ^      t
                                           |                         |
                                         ST_MAX                    ST_MIN
```

!!
FC143, MIN_MAX_ALARM_HY : MIN/MAX ALARM MIT HYSTERESE
##
Kurzbeschreibung:
Der FC "MIN_MAX_ALARM_HY" aktiviert jeweils einen MIN- UND MAX-Alarm, wenn eine INTEGER-Variable frei parametrierbare MIN-/MAX-Grenzwerte erreicht bzw. unter- oder überschreitet. Gleichzeitig wird ein ALARM-NEUWERT generiert. Der Alarm wird erst dann zurückgesetzt, wenn die Variable den mit einer frei vorgebbaren HYSTERESE versehenen GRENZWERTBEREICH wieder verläßt.
Die Parameter des FC "MIN_MAX_ALARM_HY" müssen in einen Datenbausteinbereich eingegeben werden, der durch den "UDT_MIN_MAX_ALARM_HY" definiert wird. Sie lassen sich vom BuB-System verändern und alle Zustände der Ein- und Ausgänge des FC sind aus diesem DB-Bereich lesbar.
Die Quittierung der Alarmeist nach 2 MODI möglich: Der Alarm geht nur, wenn er quittiert wurde und die Störungsursache nicht mehr anliegt ODER der Alarm geht sofort, ohne Quittung, nach Beseitigung der Störungsursache.
Die OUTPUTS des FC "MIN_MAX_ALARM_HY" können auch aus einer Universalschnittstelle, dem INFO-BYTE, gelesen werden.
##
!!Dieser FC verändert nicht die Adressregister AR1/AR2 UND versorgt für FUP- UND KOP-Anwendungen die "ENO"-Box, indem das VKE=1 im BIE-BIT gespeichert wird.

in:

QUITT [BOOL]: =0->1 die "STÖRUNG" wird mit internem PULS quittiert, liegt aber solange an, bis die Ursache beseitigt ist. STÖRUNGSURSACHEN (Siehe MODE!) gelten als beseitigt, wenn die zu überwachende Variable wieder im zulässigen Wertebereich liegt.

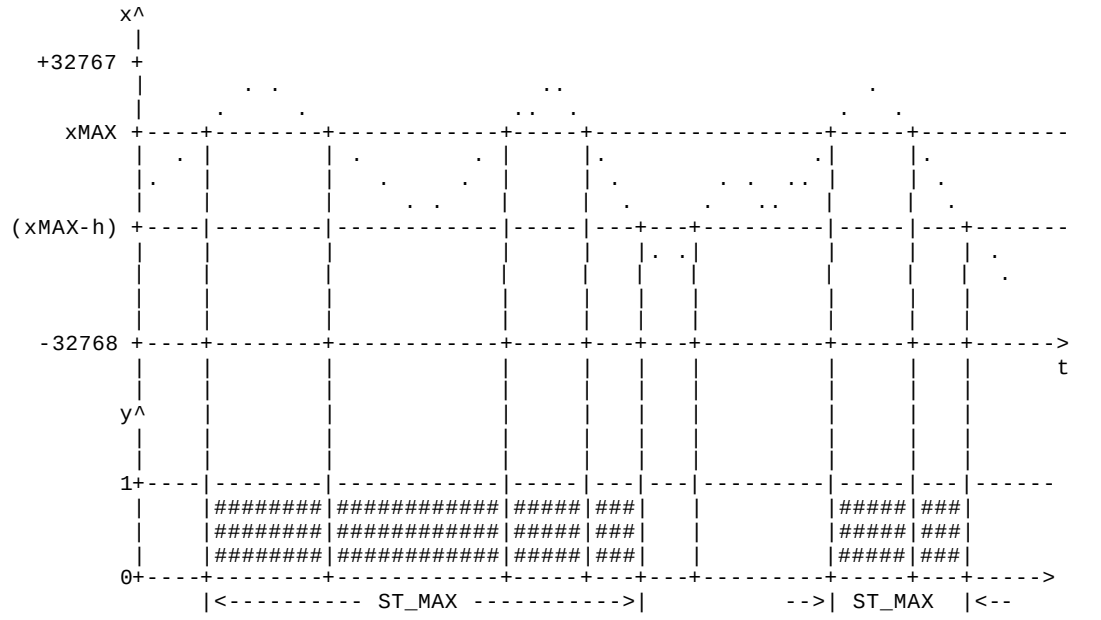
MODE=0: Störungen gehen nur, wenn sie vorher quittiert wurden und die Störungsursache nicht mehr anliegt.

MODE=1: Die Störungen gehen ohne Quittierung sofort dann, wenn die Störungsursache beseitigt ist. Eine Neuwertmeldung wird nicht generiert. Der Eingang "QUITT" ist wirkungslos.

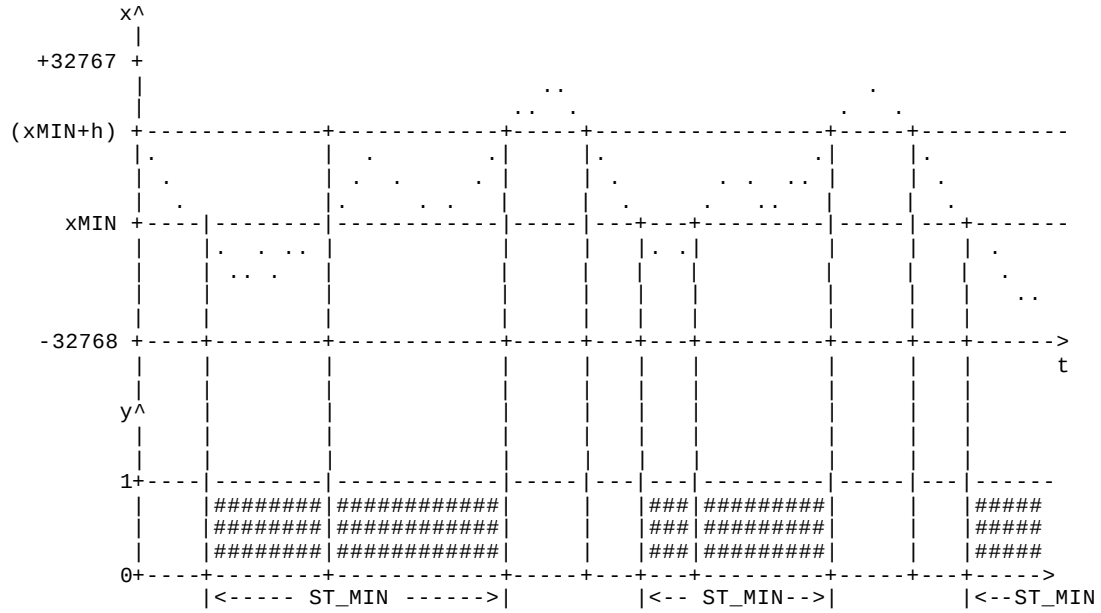
```
PRV14BYTE[POINTER]: Pointer auf die Adresse eines  14 BYTE langen DB-Speicher-
                    bereiches, dessen Struktur im "UDT_MIN_MAX_ALARM_HY" fest-
                    gelegt ist und der für jeden FC "MIN_MAX_ALARM_HY" separat
                    im Datenbaustein "DB_MIN_MAX_ALARM_HY" vorhanden sein muß.
x            [INT]: Zu überwachende Integervariable {-32768,...,0,...,+32767}
                      x<= xMIN -> MINIMUM-ALARM
                      ist der MINIMUM-ALARM HIGH, dann geht er wieder, wenn
                      x>=(xMIN+h)wird und quittiert wurde

                      x>= xMAX -> MAXIMUM-ALARM
                      ist der MAXIMUM-ALARM HIGH, dann geht er wieder, wenn
                      x<=(xMAX-h)wird und quittiert wurde

out:
ST_MIN      [BOOL]: =1 -> der MINIMUM-ALARM wurde als Störung  gespeichert
ST_MAX      [BOOL]: =1 -> der MAXIMUM-ALARM wurde als Störung  gespeichert
ST_NW       [BOOL]: =1 -> die Störungen sind noch unquittiert - Neuwert
SST         [BOOL]: =1 -> SAMMELSTÖRUNG = MIN- ODER MAX-ALARM
INFO       [BYTE]: Informationsbyte mit dem BIT-Muster "0sq0_0gk0", wobei die
                    BITS folgende Bedeutung haben:
                      BIT 0   = 0 : ohne Bedeutung
                      BIT 1   = k : =1 -> Störung MINIMUM-ALARM
                      BIT 2   = g : =1 -> Störung MAXIMUM-ALARM
                      BIT 3+4 = 0 : ohne Bedeutung
                      BIT 5   = q : =1 -> Störung=NEUWERT, nach QUITTIERUNG:=0
                      BIT 6   = s : =1 -> SAMMELSTÖRUNG, ein ALARM ist high
                      BIT 7   = 0 : ohne Bedeutung

!!FC-INTERN werden die Vergleiche zwischen x,(xMIN+h) und (xMAX-h) im DINT-
  FORMAT vorgenommen, um Überlaufprobleme des INT-Zahlenbereiches zu verhindern.
  Es sind Eingabewerte möglich, die eine Störung aktivieren, aber nicht mehr
  deaktivieren können.
###############################################################################
Im Datenbaustein "DB_MIN_MAX_ALARM_HY" müssen folgende Parameter eingegeben
oder können gelesen werden:
Alle mit "[W]"     gekennzeichneten Parameter sind zwingend einzugeben
Die  mit "[WO]"    gekennzeichneten Parameter sind optional einzugeben
Alle mit "[R]"     gekennzeichneten Parameter können nur gelesen werden
Alle mit "[I]/[O]" gekennzeichneten Parameter sind identisch mit In- und Outputs
                   des zugeordneten FCxxx und können nur gelesen werden
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
MODE   [BOOL]: MODE=0: Störungen gehen nur, wenn sie vorher quittiert wurden
                       und die Störungsursache nicht mehr anliegt.
               MODE=1: Die Störungen gehen ohne Quittierung sofort dann, wenn
                       die Störungsursache beseitigt ist. Eine Neuwertmeldung
                       wird nicht generiert. Der Eingang QUITT ist wirkungslos.
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
xMIN    [INT]: GRENZWERT FÜR MINIMUM-ALARM         {-32768,...,0,...,+32767}
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
xMAX    [INT]: GRENZWERT FÜR MAXIMUM-ALARM         {-32768,...,0,...,+32767}
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
h       [INT]: Hysterese                                  {+1,+2,...,+32767}
               !!h<+1 -> FC-intern Korrektur h:=+1
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

Beispiel:

ST_MAX

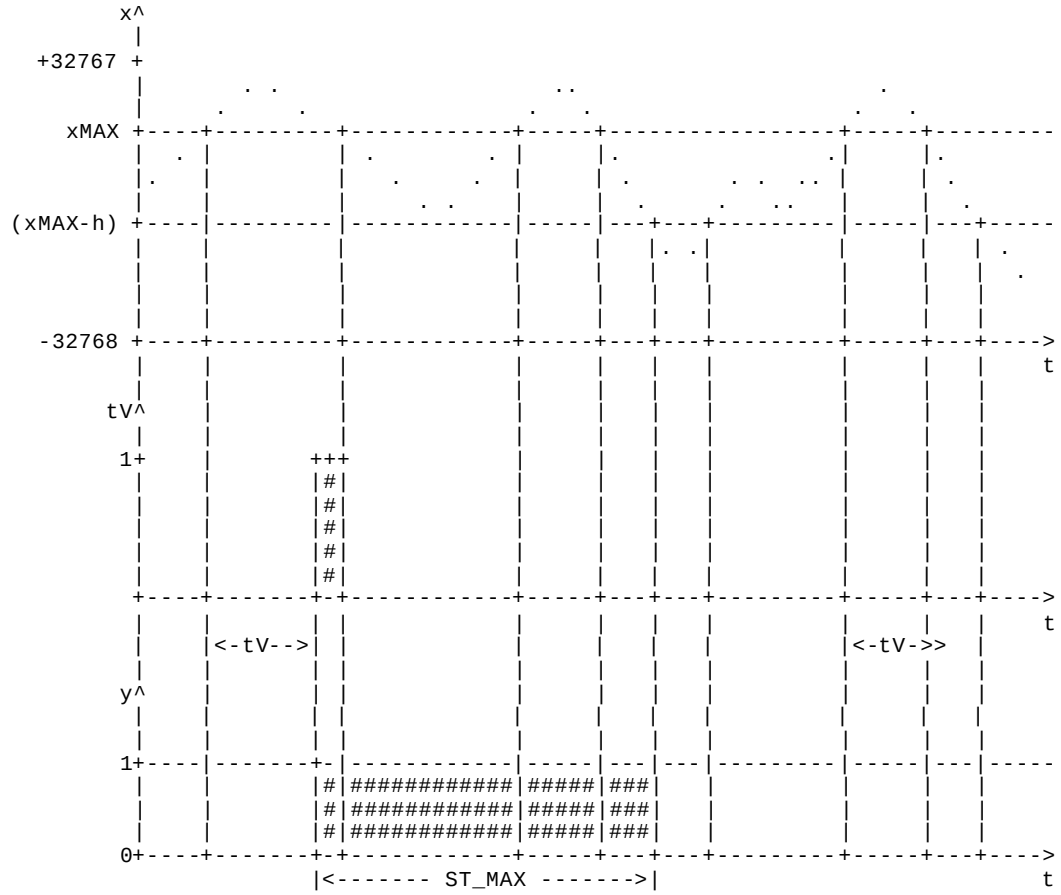


ST_MIN



```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
FC144, MIN_MAX_ALARM_HY_tV : MIN/MAX ALARM MIT HYSTERESE + tV
################################################################################
Kurzbeschreibung:
Der FC "MIN_MAX_ALARM_HY_tV" aktiviert jeweils einen MIN- UND MAX-Alarm, wenn
eine INTEGER-Variable frei parametrierbare MIN-/MAX-Grenzwerte erreicht bzw.
unter oder überschreitet UND diese Grenzwertverletzung permanent bis zum Ablauf
einer Verzögerungszeit anliegt.  Gleichzeitig wird ein ALARM-NEUWERT generiert.
Der Alarm wird erst dann zurückgesetzt, wenn die Variable den mit einer frei
vorgebbaren HYSTERESE versehenen GRENZWERTBEREICH wieder verläßt.
Die Parameter des FC "MIN_MAX_ALARM_HY_tV" müssen in einen Datenbausteinbereich
eingegeben werden, der durch den "UDT_MIN_MAX_ALARM_HY_tV" definiert wird. Sie
lassen sich vom BuB-System verändern und alle Zustände der Ein- und Ausgänge
des FC sind aus diesem DB-Bereich lesbar.
Die Quittierung der Alarme ist nach 2 MODI möglich: Der Alarm geht nur, wenn
er quittiert wurde und die Störungsursache nicht mehr anliegt ODER der Alarm
geht sofort, ohne Quittung, nach Beseitigung der Störungsursache.
Die OUTPUTS des FC "MIN_MAX_ALARM_HY_tV" können auch aus einer Universal-
schnittstelle, dem INFO-BYTE, gelesen werden.
################################################################################
!!Dieser FC verändert nicht die Adressregister AR1/AR2 UND versorgt für FUP- UND
  KOP-Anwendungen die "ENO"-Box, indem das VKE=1 im BIE-BIT gespeichert wird.

in:
FIRST_SCAN  [BOOL]: Im 1.OB1-ZYKLUS nach OB100/101/102=TRUE, sonst FALSE
                    FIRST_SCAN wird zur Initialisierung unbedingt benötigt!
QUITT       [BOOL]: =0->1 die "STÖRUNG" wird mit internem PULS quittiert,
                    liegt aber solange an, bis die Ursache beseitigt ist. STÖ-
                    RUNGSURSACHEN (Siehe MODE!) gelten als beseitigt, wenn die
                    zu überwachende Variable wieder im zulässigen Wertebereich
                    liegt.
                      MODE=0: Störungen gehen nur, wenn sie vorher quittiert
                              wurden und die Störungsursache nicht mehr anliegt.
                      MODE=1: Die Störungen gehen ohne Quittierung sofort dann,
                              wenn die Störungsursache beseitigt ist. Eine Neu-
                              wertmeldung wird nicht generiert. Der Eingang
                              "QUITT" ist wirkungslos.
PRV26BYTE[POINTER]: Pointer auf die Adresse eines  26 BYTE langen DB-Speicher-
                    bereiches, dessen Struktur im "UDT_MIN_MAX_ALARM_HY_tV"
                    festgelegt ist und der für jeden FC "MIN_MAX_ALARM_HY_tV"
                    separat im Datenbaustein "DB_MIN_MAX_ALARM_HY_tV" vorhanden
                    sein muß.
x            [INT]: Zu überwachende Integervariable {-32768,...,0,...,+32767}
                      x<= xMIN UND tV=HIGH -> MINIMUM-ALARM
                      ist der MINIMUM-ALARM HIGH, dann geht er wieder, wenn
                      x>=(xMIN+h)wird und bei MODE=0 quittiert wurde

                      x>= xMAX UND tV=HIGH -> MAXIMUM-ALARM
                      ist der MAXIMUM-ALARM HIGH, dann geht er wieder, wenn
                      x<=(xMAX-h)wird und bei MODE=0 quittiert wurde

out:
ST_MIN      [BOOL]: =1 -> der MINIMUM-ALARM wurde als Störung  gespeichert
ST_MAX      [BOOL]: =1 -> der MAXIMUM-ALARM wurde als Störung  gespeichert
ST_NW       [BOOL]: =1 -> die Störungen sind noch unquittiert - Neuwert
SST         [BOOL]: =1 -> SAMMELSTÖRUNG = MIN- ODER MAX-ALARM
INFO        [BYTE]: Informationsbyte mit dem BIT-Muster "0sq0_0gk0", wobei die
                    BITS folgende Bedeutung haben:
                      BIT 0   = 0 : ohne Bedeutung
                      BIT 1   = k : =1 -> Störung MINIMUM-ALARM
                      BIT 2   = g : =1 -> Störung MAXIMUM-ALARM
                      BIT 3+4 = 0 : ohne Bedeutung
                      BIT 5   = q : =1 -> Störung=NEUWERT, nach QUITTIERUNG:=0
                      BIT 6   = s : =1 -> SAMMELSTÖRUNG, ein ALARM ist high
                      BIT 7   = 0 : ohne Bedeutung

!!FC-INTERN werden die Vergleiche zwischen x,(xMIN+h) und (xMAX-h) im DINT-
  FORMAT vorgenommen, um Überlaufprobleme des INT-Zahlenbereiches zu verhindern.
  Es sind Eingabewerte möglich, die eine Störung aktivieren, aber nicht mehr
  deaktivieren können.
################################################################################
Im Datenbaustein "DB_MIN_MAX_ALARM_HY_tV" müssen folgende Parameter eingegeben
oder können gelesen werden:
```

```
Alle mit "[W]"     gekennzeichneten Parameter sind zwingend einzugeben
Die  mit "[WO]"    gekennzeichneten Parameter sind optional einzugeben
Alle mit "[R]"     gekennzeichneten Parameter können nur gelesen werden
Alle mit "[I]/[O]" gekennzeichneten Parameter sind identisch mit In- und Outputs
                   des zugeordneten FCxxx und können nur gelesen werden
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
MODE   [BOOL]: MODE=0: Störungen gehen nur, wenn sie vorher quittiert wurden
                       und die Störungsursache nicht mehr anliegt.
               MODE=1: Die Störungen gehen ohne Quittierung sofort dann, wenn
                       die Störungsursache beseitigt ist. Eine Neuwertmeldung
                       wird nicht generiert. Der Eingang QUITT ist wirkungslos.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
xMIN    [INT]: GRENZWERT FÜR MINIMUM-ALARM         {-32768,...,0,...,+32767}
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
xMAX    [INT]: GRENZWERT FÜR MAXIMUM-ALARM         {-32768,...,0,...,+32767}
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
h       [INT]: Hysterese                                {+1,+2,...,+32767}
               !!h<+1 -> FC-intern Korrektur h:=+1
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
tV_ST  [TIME]: VERZÖGERUNGSZEIT             {0,1ms,...,max=24d20h31m23s647ms}
               LIEGT DER ZU ÜBERWACHENDE INTEGERWERT IM ALARMBEREICH, DANN
               WIRD DIE ÜBERWACHUNSZEIT GESTARTET. EIN ALARM WIRD ERST DANN
               GENERIERT, WENN DIE ZEIT ABGELAUFEN IST UND DIE GRENZWERTVER-
               LETZUNG WEITERHIN ANLIEGT.
               !!IST xMIN >= xMAX , DANN LIEGT FÜR JEDEN WERT "x" EIN MIN-
                 UND/ODER MAX-ALARM AN.IN DIESEM FALL WIRD DER TIMER SOFORT
                 NACH FIRST_SCAN GESTARTET.
               !!DER TIMER WIRD BEI JEDEM WECHSEL VON "x<=xMIN <--> x>=xMAX"
                 NEU GESTARTET.
               !!BEI FEHLEINGABE tV_ST<0 -> tV_ST:=0
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Beispiel:

ST_MAX
```



ST_MIN

```
        x^
         |
  +32767 +
         |                             ..                      .
         |                            .   .                   .   .
(xMIN+h) +--------------+------------+-----+-----------------+-----+----------
         |.             |  .        .|     |.               .|     |.
         | .            | .  .     . |     | .       . .  .. |     | .
         |  .           |.     . .   |     |  .     .   ..   |     |  .
    xMIN +----|---------|------------|-----|---+---+---------|-----|---+------
         |    |.  . ... |            |     |   |. .|         |     |   | .
         |    | .. .    |            |     |   |   |         |     |   |  .
         |    |         |            |     |   |   |         |     |   |   ..
         |    |         |            |     |   |   |         |     |   |
  -32768 +----+---------+------------+-----+---+---+---------+-----+---+------->
         |    |         |            |     |   |   |         |     |   |      t
         |    |         |            |     |   |   |         |     |   |
       tV^    |         |            |     |   |   |         |     |   |
         |    |         |            |     |   |   |         |     |   |
        1+    |       +++            |     |   |   |         |     |   |
         |    |       |#|            |     |   |   |         |     |   |
         |    |       |#|            |     |   |   |         |     |   |
         |    |       |#|            |     |   |   |         |     |   |
         |    |       |#|            |     |   |   |         |     |   |
         |    |       |#|            |     |   |   |         |     |   |
         |    |       |#|            |     |   |   |         |     |   |
         +----+-------+-+------------+-----+---+---+---------+-----+---+----->
         |    |       | |            |     |   |   |         |     |   |     t
         |    |<-tV-->| |            |     |   +-tV->>       |     |   +-tV->>
         |    |       | |            |     |   |   |         |     |   |
        y^    |       | |            |     |   |   |         |     |   |
         |    |       | |            |     |   |   |         |     |   |
         |    |       | |            |     |   |   |         |     |   |
        1+----|-------+++------------|-----|---|---|---------|-----|---|------
         |    |       |#|############|     |   |   |         |     |   |
         |    |       |#|############|     |   |   |         |     |   |
         |    |       |#|############|     |   |   |         |     |   |
        0+----+-------+-+------------+-----+---+---+---------+-----+---+----->
                      |<--ST_MIN -->|                                        t
```

!!
FC150, P : P-REGLER
##
Kurzbeschreibung:
Der FC "P" stellt einen Proportionalregler dar, dessen Parameter in einen durch den "UDT_P" definierten Datenbausteinbereich im INTEGERFORMAT eingegeben werden müssen und die sich vom BuB-System verändern lassen. Der FC-Input"SH" ermöglicht Sollwertschiebungen für Kaskadenregelungen. Ober- und Untergrenze der Sollwertschiebung sind parametrierbar.
Der Regler besitzt folgende 3 Norm-Ausgangssignale im Bereich {-1000,...,+1000}:

```
  "y"   {-1000,...,0,...,+1000}, als direktwirkendes Signal, dessen Vorzeichen
        durch das Vorzeichen der Regelabweichung "(x-w)" bestimmt wird.
  "yUW" {+1000,...,0} als umgekehrtwirkendes Signal für "(x-w)<=0"
  "yDW" {0,...,+1000} als direktwirkendes    Signal für "(x-w)>=0"
```

Alle Parameter des Reglers sind Festkommazahlen. Aus diesem Grund wird statt der Proportionalverstärkung "KP" das Proportionalband "XP" als Parameter des Reglers verwendet. (XP=1/KP!)
##
!!Dieser FC verändert nicht die Adressregister AR1/AR2 UND versorgt für FUP- UND KOP-Anwendungen die "ENO"-Box, indem das VKE=1 im BIE-BIT gespeichert wird.

in:

```
ENABLE      [BOOL]: =0 -> DISABLE: y:=yUW:=yDW:=0;  =1 -> ENABLE -> Berechnung
PRV22BYTE[POINTER]: Pointer auf die Anfangsadresse eines 22 BYTE langen DB-
                    Speicherbereiches, dessen Struktur im "UDT_P" festgelegt
                    ist und der für jeden P-Regler separat im "DB_P" vorhan-
                    den sein muß.Weiter unten sind die einzelnen Inputs in den
                    "DB_P" näher erläutert.
x            [INT]: Istwert der Regelgröße        {-32768,...,0,...,+32767}*10^m
```

```
SH          [INT]: Sollwertschiebung/Kaskadeninp.{-32768,...,0,...,+32767}*10^m
                   Die Schiebeweite wird durch die Eingabewerte "w_SH_MIN" und
                   und "w_SH_MAX" im "DB_P"wie folgt begrenzt:
                   SH>0:= SH(+) -> [w+SH(+)]<=wSH_MAX;   w > wSH_MAX -> SH:=0
                   SH<0:= SH(-) -> [w+SH(-)]>=wSH_MIN;   w < wSH_MIN -> SH:=0

out:
y           [INT]: Stellgröße  = Normsignal            {-1000,...,0,...,+1000}
yUW         [INT]: Stellgröße umgekehrt wirkend               {+1000,...,+1,0}
                   y>0 -> yUW:=0 UND y<=0 -> yUW:=-y
yDW         [INT]: Stellgröße direkt wirkend                  {0,+1,...,+1000}
                   y<0 -> yDW:=0 UND y>=0 -> yDW:=y
!!Alle Eingabewerte in den "DB_P" werden im "FC_P" auf die angegebenen
  Grenzen geprüft. Sind Eingaben fehlerhaft, so werden sie auf die zuläs-
  sige Untergrenze=UG bzw. Obergrenze=OG FC-intern korrigiert.
################################################################################
Im "DB_P" müssen folgende Parameter eingegeben oder können gelesen werden:
Alle mit "[W]"     gekennzeichneten Parameter sind zwingend einzugeben
Alle mit "[R]"     gekennzeichneten Parameter können nur gelesen werden
Alle mit "[I]/[O]" gekennzeichneten Parameter sind identisch mit In- und Outputs
                   des zugeordneten FCxxx und können nur gelesen werden
################################################################################
Für jeden P-Regler sind das folgende Parameter:
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
w       [INT]: Sollwert, Führungsgröße           {-32768,...,0,...,+32767}*10^m
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
w_SH_MIN[INT]: MINIMUM für "w := [w+SH(-)]"      {-32768,...,0,...,+32767}*10^m
               Gilt bereits "w < w_SH_MIN", dann kann "w" durch SH(-) nicht noch
               zu kleineren Werten geschoben werden.
               !! w_SH_MIN ist keine Begrenzung für die Sollwerteingabe "w" !!
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
w_SH_MAX[INT]: MAXIMUM für "w := [w+SH(+)]"      {-32768,...,0,...,+32767}*10^m
               Gilt bereits "w > w_SH_MAX", dann kann "w" durch SH(+) nicht noch
               zu größeren Werten geschoben werden.
               !! w_SH_MAX ist keine Begrenzung für die Sollwerteingabe "w" !!
!! Relationen, wie w_SH_MIN >= w_SH_MAX verursachen lediglich, und das ist ab-
   hängig von der Lage von "w", daß nur SH(-) ODER nur SH(+) ODER keine Soll-
   wertschiebung möglich ist.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
XP      [INT]: Proportionalband                    {1,2,...,32767}*10^m
               Ist "Kp" die Proportionalverstärkung, so gilt: "XP = 1/Kp"
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Begriffe:

x     = Istwert  der Regelgröße        x  {-32768,...,0,...,+32767}*10^m
w     = Sollwert der Regelgröße        w  {-32768,...,0,...,+32767}*10^m
w_eff = wirkender Sollwert   w_eff=(w+SH) {-32768,...,0,...,+32767}*10^m
SH    = Sollwertschiebung             SH {-32768,...,0,...,+32767}*10^m
SH(-) = negativer Bereich von SH    SH(-) {-32768,...,-1,0}*10^m
SH(+) = positiver Bereich von SH    SH(+) {0,+1,...,+32767}*10^m
w_SH_MIN= MINIMUM für w:=[w+SH(-)]  w_SH_MIN {-32768,...,0,...,+32767}*10^m
        Gilt bereits "w < w_SH_MIN", dann kann "w" durch SH(-) nicht noch
        zu kleineren Werten geschoben werden.
        !! w_SH_MIN ist keine Begrenzung für die Sollwerteingabe "w" !!
w_SH_MAX= MAXIMUM für w:=[w+SH(+)]  w_SH_MAX {-32768,...,0,...,+32767}*10^m
        Gilt bereits "w > w_SH_MAX", dann kann "w" durch SH(+) nicht noch
        zu größeren Werten geschoben werden.
        !! w_SH_MAX ist keine Begrenzung für die Sollwerteingabe "w" !!
        !!Relationen, wie w_SH_MIN >= w_SH_MAX verursachen lediglich, und
        das ist abhängig von der Lage von "w", daß nur SH(-) ODER nur SH(+)
        ODER keine Sollwertschiebung möglich ist.
XP    = Proportionalband               XP {1,2,...,32767}*10^m
xW    = (x-w) = Regelabweichung        xW {-65536,...,0,...,+65534}*(10^m)
y     = Stellgröße                      y {-1000,...,0,...,+1000}*0.1%
yUW   = Stellgröße umgekehrt wirkend  yUW {1000,...,1,0}*0.1%
yDW   = Stellgröße direkt    wirkend  yDW {0,1,...,1000}*0.1%
10^m  = gemeinsamer Koeffizient aller damit gekennzeichneter Größen, der sich
        bei der Berechnung herauskürzt. Der korrekte Wert ist nur als fiktive
        Kommastelle oder angehängte Nullen beim BuB wichtig.
m     = Exponent zur Basis 10          m {-t,...,-3,-2,-1,0,1,2,3,...,+t}
```

Darstellung der Zusammenhänge "w+SH" mit "y", "yUW" und "yDW"

```
     Y^                             +XP
      |              y            |----|y                   y
 +1000+             +-            |    +-                 +-
      |            .              |   /                  .
      |           .               |  /                  .
      |          .                | /                  .
      |         .                 |/                  .
     0+--------+------------------+------------------+----------->
      |    w:=w+SH(-)            /w              w:=w+SH(+)      x
      |      .                  / |                .
      |     .                  /  |               .
      |    .                  /   |              .
 -1000+  -+                 -+    |            -+
      |                      |----|
                              -XP
```

yUW: y<=0 -> yUW:=|y| UND y>0 -> yUW:=0; yDW: y<0 -> yDW:0 UND y=>0 -> yDW:=y

```
  yDW ^
  yUW |                       -XP   +XP
      |yUW           yDW  yUW|----|----|yDW  yUW            yDW
 +1000+  -+         +-      -+    |    +-      -+           +-
      |    .       .          \   |   /          .       .
      |     .     .            \  |  /            .     .
      |      .   .              \ | /              .   .
      |       . .                \|/                . .
     0+--------+------------------+------------------+----------->
      |    w:=w+SH(-)             w              w:=w+SH(+)      x
```

Die Stellgröße wird wie folgt berechnet:

```
       1
   y=---*xW(n)*1000;
       Xp
```

und begrenzt:
   -1000<=y<=+1000 (Das gilt damit auch für yDW, yUW!)

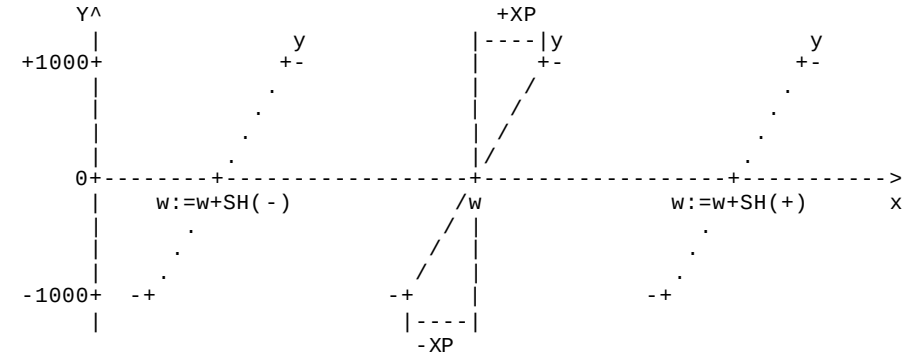
Wobei gilt:

   xW=(x-w); w:=w_eff=(w+SH); (Siehe aber auch wSH_MIN und wSH_MAX !)

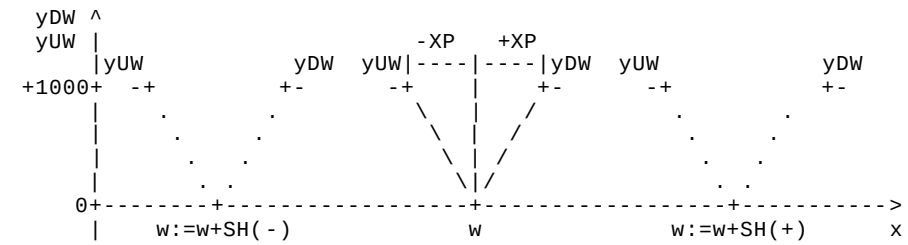
!!
FC151, PID : PID-REGLER
##
Kurzbeschreibung:
Der FC "PID" stellt einen Proportional- Integral- und Differentialregler dar, dessen Parameter in einen durch den "UDT_PID" definierten Datenbausteinbereich im INTEGERFORMAT eingegeben werden müssen und die sich vom BuB-System verändern lassen. Der FC-Input "SH" ermöglicht Sollwertschiebungen für Kaskadenregelungen. Ober- und Untergrenze der Sollwertschiebung sind parametrierbar.
Der Regler besitzt folgende 3 Norm-Ausgangssignale im Bereich {-1000,...,+1000}:
  "y"   {-1000,...,0,...,+1000}, als direktwirkendes Signal, dessen Vorzeichen durch das Vorzeichen der Regelabweichung "(x-w)" bestimmt wird.
  "yUW" {+1000,...,0} als umgekehrtwirkendes Signal für "(x-w)<=0"
  "yDW" {0,...,+1000} als direktwirkendes   Signal für "(x-w)>=0"
Alle Parameter des Reglers sind Festkommazahlen. Aus diesem Grund wird statt der Proportionalverstärkung "K" das Proportionalband "XP" als Parameter des Reglers verwendet - (XP=1/K!). Die Nachstellzeit "TN" und Vorhaltezeit "TV" sind ebenfalls Integerzahlen mit der Maßeinheit [0.1s]. Die einzelnen Komponenten der Regelgröße, der Proportional-, der Integral- und der Differentialanteil sind separat ein- und ausschaltbar. Der Integralanteil wird begrenzt und damit das "HOCHINTEGRIEREN" des Reglers verhindert.
Für den D-Anteil ist eine Haltezeit "tH_yD" konfigurierbar. Der D-Anteil wirkt als Trenderkennung und zeigt kein Sprungverhalten. Die Ansteuerung von Stellgliedern wird dadurch optimal mit PID-Verhalten möglich.
##

```
!!Dieser FC verändert nicht die Adressregister AR1/AR2 UND versorgt für FUP- UND
  KOP-Anwendungen die "ENO"-Box, indem das VKE=1 im BIE-BIT gespeichert wird.

in:
FIRST_SCAN  [BOOL]: Im 1.OB1-ZYKLUS nach OB100/101/102=TRUE, sonst FALSE
                    FIRST_SCAN wird zur Initialisierung unbedingt benötigt!
ENABLE      [BOOL]: =0 -> DISABLE: y:=yUW:=yDW:=0;  =1 -> ENABLE -> Berechnung
PRV54BYTE[POINTER]: Pointer auf die Anfangsadresse eines 54 BYTE langen DB-
                    Speicherbereiches, dessen Struktur im "UDT_PID" festgelegt
                    ist und der für jeden PID-Regler separat im "DB_PID" vorhan-
                    den sein muß. Weiter unten sind die einzelnen Inputs in den
                    "DB_PID" näher erläutert.
x           [INT]: Istwert der Regelgröße        {-32768,...,0,...,+32767}*10^m
SH          [INT]: Sollwertschiebung/Kaskadeninp.{-32768,...,0,...,+32767}*10^m
                    Die Schiebeweite wird durch die Eingabewerte "w_SH_MIN" und
                    und "w_SH_MAX" im "DB_PID"wie folgt begrenzt:
                    SH>0:= SH(+) -> [w+SH(+)]<=wSH_MAX;   w > wSH_MAX -> SH:=0
                    SH<0:= SH(-) -> [w+SH(-)]>=wSH_MIN;   w < wSH_MIN -> SH:=0

out:
y           [INT]: Stellgröße  = Normsignal             {-1000,...,0,...,+1000}
yUW         [INT]: Stellgröße umgekehrt wirkend                {+1000,...,+1,0}
                    y>0 -> yUW:=0 UND y<=0 -> yUW:=-y
yDW         [INT]: Stellgröße direkt wirkend                   {0,+1,...,+1000}
                    y<0 -> yDW:=0 UND y>=0 -> yDW:=y
!!Alle Eingabewerte in den "DB_PID" werden im FC "PID" auf die angegebenen Gren-
  zen geprüft. Sind Eingaben fehlerhaft, so werden sie auf die zulässige Unter-
  grenze=UG bzw. Obergrenze=OG FC-intern korrigiert.
################################################################################
Im DB_PID müssen folgende Parameter eingegeben oder können gelesen werden:
Alle mit "[W]"     gekennzeichneten Parameter sind zwingend einzugeben
Die  mit "[WO]"    gekennzeichneten Parameter sind optional einzugeben
Alle mit "[R]"     gekennzeichneten Parameter können nur gelesen werden
Alle mit "[I]/[O]" gekennzeichneten Parameter sind identisch mit In- und Outputs
                   des zugeordneten FCxxx und können nur gelesen werden
################################################################################
Für jeden PID-Regler sind das folgende Parameter:
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
w       [INT]: Sollwert, Führungsgröße             {-32768,...,0,...,+32767}*10^m
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
w_SH_MIN[INT]: MINIMUM für "w := [w+SH(-)]"        {-32768,...,0,...,+32767}*10^m
               Gilt bereits "w < w_SH_MIN", dann kann "w" durch SH(-) nicht noch
               zu kleineren Werten geschoben werden.
               !! w_SH_MIN ist keine Begrenzung für die Sollwerteingabe "w" !!
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
w_SH_MAX[INT]: MAXIMUM für "w := [w+SH(+)]"        {-32768,...,0,...,+32767}*10^m
               Gilt bereits "w > w_SH_MAX", dann kann "w" durch SH(+) nicht noch
               zu größeren Werten geschoben werden.
               !! w_SH_MAX ist keine Begrenzung für die Sollwerteingabe "w" !!
!! Relationen, wie w_SH_MIN >= w_SH_MAX verursachen lediglich, und das ist ab-
   hängig von der Lage von "w", daß nur SH(-) ODER nur SH(+) ODER keine Soll-
   wertschiebung möglich ist.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

XP      [INT]: Proportionalband                          {1,2,...,32767}*10^m
               Ist "Kp" die Proportionalverstärkung, so gilt: "XP = 1/Kp"
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
TN      [INT]: Nachstellzeit                             {0,1,...,32767}*0.1s
               TN=0 -> der Integral-Anteil "yI" wird nicht berechnet.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
TV      [INT]: Vorhaltezeit                              {0,1,...,32767}*0.1s
               TV=0 -> der Differential-Anteil "yD" wird nicht berechnet.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
tH_yD   [INT]: Haltezeit                                 {1,2,...,10000}*0.01s
               Für diese Zeit wird ein berechneter "yD-Anteil" konstant gehalten
               und erst nach Ablauf der Zeit neu berechnet. Dadurch wirkt "yD"
               nicht als sprunghafte Änderung. Es geht als "Trenderkennung" in
               die berechnete Stellgröße ein und berücksichtigt das Totzeit-
               verhalten von Sensoren und Wandlungszeiten von Analogeingangs-
               baugruppen.
               Empfohlene Einstellung: "200*0.01s = 2s".
               Siehe auch Anmerkung zu den Berechnungsgrundlagen!
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

```
NSW    [BOOL]: =1 -> Wenn die Werte für "w_eff, XP und TN" durch Eingaben ver-
               ändert werden, dann wird "yI":=0 und "yD":=0 und mit den geän-
               derten Werten neu berechnet. Auch alle zur Berechnung gespeicher-
               ten Altwerte :=0!
               Wird lediglich "TN" UND/ODER "XP" verändert, dann wird nur der
               Integralanteil "yI":=0 neu berechnet.
               =0 -> Werden neue Sollwerte eingegeben, dann werden diese Werte
               übernommen, aber eine neue Stellgröße auf der Basis der bishe-
               rigen Altwerte berechnet.
               !!Für Folgeregler bei Kaskadenregelungen muß "NSW:=0".
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
yP_aus [BOOL]: =0 -> yP ist eingeschaltet
               =1 -> yP ist ausgeschaltet
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
yI_aus [BOOL]: =0 -> yI ist eingeschaltet, wenn TN>0 ist !
               =1 -> yI ist ausgeschaltet, unabhängig von "TN"!
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
yD_aus [BOOL]: =0 -> yD ist eingeschaltet, wenn TV>0 ist !
               =1 -> yD ist ausgeschaltet, unabhängig von "TV"!
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Begriffe:

x     = Istwert  der Regelgröße        x  {-32768,...,0,...,+32767}*10^m
w     = Sollwert der Regelgröße        w  {-32768,...,0,...,+32767}*10^m
w_eff = wirkender Sollwert   w_eff=(w+SH) {-32768,...,0,...,+32767}*10^m
SH    = Sollwertschiebung              SH {-32768,...,0,...,+32767}*10^m
SH(-) = negativer Bereich von SH    SH(-) {-32768,...,-1,0}*10^m
SH(+) = positiver Bereich von SH    SH(+) {0,+1,...,+32767}*10^m
w_SH_MIN= MINIMUM für w:=[w+SH(-)]  w_SH_MIN {-32768,...,0,...,+32767}*10^m
        Gilt bereits "w < w_SH_MIN", dann kann "w" durch SH(-) nicht noch
        zu kleineren Werten geschoben werden.
        !! w_SH_MIN ist keine Begrenzung für die Sollwerteingabe "w" !!
w_SH_MAX= MAXIMUM für w:=[w+SH(+)]  w_SH_MAX {-32768,...,0,...,+32767}*10^m
        Gilt bereits "w > w_SH_MAX", dann kann "w" durch SH(+) nicht noch
        zu größeren Werten geschoben werden.
        !! w_SH_MAX ist keine Begrenzung für die Sollwerteingabe "w" !!
        !!Relationen, wie w_SH_MIN >= w_SH_MAX verursachen lediglich, und
        das ist abhängig von der Lage von "w", daß nur SH(-) ODER nur SH(+)
        ODER keine Sollwertschiebung möglich ist.
XP    = Proportionalband               XP {1,2,...,32767}*10^m
TN    = Nachstellzeit                  TN {0,1,...,32767}*0.1s;   TN=0 -> yI:=0
TV    = Vorhaltezeit                   TV {0,1,...,32767}*0.1s;   TV=0 -> yD:=0
xW    = (x-w) = Regelabweichung        xW {-65536,...,0,...,+65534}*(10^m)
y     = Stellgröße                      y {-1000,...,0,...,+1000}*0.1%
yUW   = Stellgröße umgekehrt wirkend  yUW {1000,...,1,0}*0.1%
yDW   = Stellgröße direkt wirkend     yDW {0,1,...,1000}*0.1%
yP    = Proportionalanteil in y        yP {-1000,...,0,...,+1000}
yI    = Integralanteil     in y        yI {-1000,...,0,...,+1000}
yD    = Differentialanteil in y        yI {-1000,...,0,...,+1000}
n     = CPU-Zyklusnummer
dTZ   = Die an derselben Programm-     dTZ_max=4000ms
        stelle gemessene, zwischen
        2 CPU-Zyklen vergangene Zeit.
dTI   = Zeit "dT" für das Integra-     dTI=dTZ
        tionsintervall
dTD   = Zeit "dT" für das Differen-    dTD=tH_yD
        tationsintervall.
T(n)  = Timer-Neuwert im CPU-Zyklus (n)
T(n-1)= Timer-Altwert im CPU-Zyklus (n-1)
T(n-k)= Timer-Altwert im CPU-Zyklus (n-k)
10^m  = gemeinsamer Koeffizient aller damit gekennzeichneter Größen, der sich
        bei der Berechnung herauskürzt. Der korrekte Wert ist nur als fiktive
        Kommastelle oder angehängte Nullen beim BuB wichtig.
m     = Exponent zur Basis 10          m {-t,...,-3,-2,-1,0,1,2,3,...,+t}
tH_yD = Haltezeit yD              tH_yD {1,2,...,10000}*0.01s
```



yUW: y<=0 -> yUW:=|y| UND y>0 -> yUW:=0; yDW: y<0 -> yDW:0 UND y=>0 -> yDW:=y



Die Stellgröße berechnet sich aus 3 Anteilen:
   y=yP+yI+yD;

Begrenzungen der Anteile:
   yP: -1000<=yP<=+1000 (!!Nur in y, yDW, yUW)
   yI: yP>=0: -1000<=yI<=(1000-yP); yP<0: (-1000-yP)<=yI<=+1000
   yD: -1000<=yD<=+1000

Unbegrenzt:
   yP_UNLIM: -65535000<=yP<=+65535000

Der Wert "yP" ist repräsentativ für die tatsächlich vorhandene Regelabweichung.
Deshalb wird das unbegrenzte "yP" wird zu dem begrenzten "yD" und dem begrenzten
"yI" addiert und anschließend zur Stellgröße "y" begrenzt!

Stellgröße
   y=yP_UNLIM+yI+yD; nach Addition begrenzt auf: -1000<=y<=+1000

Berechnungsgrundlagen:
 Proportionalanteil:
       1
   yP=---*xW(n)*1000;
       Xp

 Integralanteil:
       1 i=n
   yI=---*S xW(n)*[T(n)-T(n-1)]*1000;
      TI i=0

 Differentialanteil:
          [xW(n)-xW(n-k)]
   yD=TD*----------------*1000
          [T(n)-T(n-k)]

!! Der berechnete Wert "yD" wird bis zur nächsten, durch die Zeit "tH_yD"
   bestimmten, neuen Berechnung, als konstant betrachtet und geht in "y"
   ein. Dadurch wirkt "yD" wie eine Trenderkennung. Der Wert für "tH_yD"
   ist hinreichend groß einzugeben, daß überhaupt eine Änderung des Wer-
   tes "[xW(n)-xW(n-k)]" über Sensor und AI-Baugruppe erfaßt werden kann.
   Damit wird ein ständiges Springen von "yD" zwischen "0" und dem berech-
   neten Wert verhindert.

Mit den Substitutionen:

```
        TI
   TN:=----   UND  TV:=TD*XP
        XP
```

Berechnet sich die Stellgröße y:

```
      1000          1000  i=n           1000*[xW(n)-xW(n-k)]*TV
   y =----*xW(n) + ------*S xW(n)*dTI + -----------------------
       XP           XP*TN i=0                  XP*dTD
```

Wobei gilt:

```
   dTI=dTZ=[T(n)-T(n-1)];  dTD=[T(n)-T(n-k)]=tH_yD;

   xW(n)=[x(n)-w(n)]; w:=w_eff=(w+SH); (Siehe aber auch wSH_MIN und wSH_MAX !)
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
FC152, ZUSATZSEQUENZ : ZUSATZSEQUENZ ALS NORMSIGNAL
################################################################################
```

Kurzbeschreibung:
Mit dem FC "ZUSATZSEQUENZ" kann ein beliebiger Bereich einer im Integerformat vorliegenden Variablen in ein Normstellsignal {0,...,+1000} gewandelt werden. Zusatzsequenzen werden häufig in der Klimatechnik benötigt, wo es z.B außen-temperaturabhängig möglich ist, eine energetisch günstigere Form der Luftküh-lung zu wählen.

```
################################################################################
```

!!Dieser FC verändert nicht die Adressregister AR1/AR2 UND versorgt für FUP- UND KOP-Anwendungen die "ENO"-Box, indem das VKE=1 im BIE-BIT gespeichert wird.

```
in:
x     [INT]: Abszisse des beliebigen Punktes P        {-32768,...,0,...,+32767}
x0    [INT]: Abszisse Punkt P0 mit der Ordinate=0     {-32768,...,0,...,+32767}
x1000 [INT]: Abszisse Punkt P1 mit der Ordinate=1000  {-32768,...,0,...,+32767}

out:
y     [INT]: Ordinate des beliebigen Punktes P                    {0,...,+1000}

                   1000
             y = ----------*(x-x0); y<0 -> y:=0; y>1000 -> y:=1000;
                 (x1000-x0)



!!x1000 > x0 -> Zusatzsequenz direkt    wirkend =DW
!!x1000 < x0 -> Zusatzsequenz umgekehrt wirkend =UW
!!x1000 = x0 -> ERROR:   y:=0  UND BIE-BIT :=0
```

Beispiele:

```
 x0 < x1000 -> Zusatzsequenz ist direkt wirkend =DW

   y ^
     |            P1(x1000;1000)
 1000+--------------+--------
     |             /|
     |            / |
     |           /  |
     |          /   |
     |         /    |
     |P0(x0;0)/     |
    0+-------+------+------->
             x0     x1000   x
```

x0 > x1000 -> Zusatzsequenz ist umgekehrt wirkend =UW

```
   y ^
     |  P1(x1000;1000)
 1000+--+
     |  |\
     |  | \
     |  |  \
     |  |   \
     |  |    \
     |  |     \ P0(x0;0)
    0+--+------+----------->
       x1000  x0          x
```

!!
FC153, AH : DATEN-UMSCHALTER AUTO-HAND FÜR BuB
##
Kurzbeschreibung:
Mit dem FC "AH" kann vom BuB-System ein beliebiges INTEGER-Signal von AUTO auf HAND über eine Tastfunktion und einem FC-internen FLIP-FLOP umgeschaltet werden. Dazu muß dem FC "AH" in einem beliebigen Datenbaustein mit dem "UDT_AH" ein Speicherbereich zugeordnet werden, auf den das BuB-System zugreifen kann. In diesem Speicherbereich ist der AUTO- und HAND-WERT des Signales hinterlegt. Im Moment der Umschaltung ist das HAND- mit dem AUTOSIGNAL identisch und kann dann frei gewählt werden. Die Begrenzung des HANDSIGNALES auf einen zulässigen Werte-bereich muß im BuB-System erfolgen.
##
!!Dieser FC verändert nicht die Adressregister AR1/AR2 UND versorgt für FUP- UND KOP-Anwendungen die "ENO"-Box, indem das VKE=1 im BIE-BIT gespeichert wird.

```
in:
PRV8BYTE [POINTER]: Pointer auf die Anfangsadresse eines 8 BYTE langen
                    DB-Speicherbereiches, dessen Struktur im "UDT_AH" fest-
                    gelegt ist und der für jeden AH-Umschalter separat im
                    "DB_AH" enthalten sein muß. Weiter unten sind die
                    einzelnen Inputs in den "DB_AH" näher erläutert.
x            [INT]: DATEN = AUTO                     {-32768,...,+32767}

out:
y            [INT]: DATEN AUTO ODER HAND             {-32768,...,+32767}
                    SG_AH =0  -> y:=x
                    SG_AH =1  -> y:=xAH
```

!!Im Automatikbetrieb, SG_AH=0, gilt "xAH:=x".
!!Bei Umschaltung von AUTO nach HAND, SG_AH=0->1, hat "xAH" den Anfangswert "x".
!!Nach der Umschaltung, SG_AH=1, ist "xAH" beliebig vom BuB veränderbar.

```
################################################################################
Im DB_AH müssen folgende Parameter eingegeben oder können gelesen werden:
Alle mit "[W]"     gekennzeichneten Parameter sind zwingend einzugeben
Die  mit "[WO]"    gekennzeichneten Parameter sind optional einzugeben
Alle mit "[R]"     gekennzeichneten Parameter können nur gelesen werden
Alle mit "[R/W]"   gekennzeichneten Parameter können je nach Schaltzustand
                   nur gelesen oder auch am BuB-System eingegeben werden
Alle mit "[I]/[O]" gekennzeichneten Parameter sind identisch mit In- und Outputs
                   des zugeordneten FCxxx und können nur gelesen werden
!!FC-intern werden fehlerhaft eingegebene Parameter auf den angegebenen Werte-
  bereich begrenzt.
################################################################################
Für jeden AH-Umschalter sind das folgende Parameter:
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
xAH     [INT]: Je nach Schaltzustand des FLIP-FLOPs "SG_AH" nimmt "xAH" folgende
               Werte an:
                 SG_AH=0 -> xAH:=x, vom BuB-System sind keine Eingaben möglich
                 SG_AH=1 -> xAH hat den Anfangswert "x" und kann vom BuB-System
                            verändert werden
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
TAST   [BOOL]: Softkey-Taste =0/1 vom BuB-System für FLIP-FLOP "SG_AH"
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
SG_AH  [BOOL]: FLIP-FLOP, dessen Schaltzustand vom Taster "TAST" bestimmt wird
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
FC154, AD/TAKTGEBER : ANALOG-DIGITAL-WANDLER MIT PULSDAUERMODULATION
################################################################################
Kurzbeschreibung:
Mit dem FC "AD/TAKTGEBER" kann ein analoges Normsignal {-1000,...,0,...+1000}
in die periodisch taktenden, impulsdauermodulierten digitalen Stellsignale
"yPLUS" und "yMINUS" gewandelt werden.
Die Periodendauer, eine minimale Ein-/Ausschaltzeit und eine Totzeit bei Schalt-
richtungswechsel muß parametriert werden. Dazu muß dem FC "AD/TAKTGEBER" in
einem beliebigen Datenbaustein mit dem "UDT_TKT" ein Speicherbereich zugeordnet
werden, auf den auch das BuB-System zugreifen kann.
Mit dem 3-Punkt-Analog-Digitalwandler können impulsdauermoduliert Stellglieder
vom  Typ "AUF-ZU" oder "AUS-EIN" über analoge Reglerausgangssignale angesteuert
werden.
Z.B. lassen sich damit Magnetventile, Dosierpumpen, Thyristorschalter für Elek-
troheizungen usw. als Stellglieder in Regelkreisen einsetzen.
################################################################################
!!Dieser FC verändert nicht die Adressregister AR1/AR2 UND versorgt für FUP- UND
  KOP-Anwendungen die "ENO"-Box, indem das VKE=1 im BIE-BIT gespeichert wird.

in:
FIRST_SCAN  [BOOL]: Im 1.OB1-ZYKLUS nach OB100/101/102=TRUE, sonst FALSE
PRV18BYTE [POINTER]: Pointer auf die Anfangsadresse eines 18 BYTE langen
                     DB-Speicherbereiches, dessen Struktur im "UDT_TKT" fest-
                     gelegt ist und der für jeden AD-Taktgeber separat im
                     "DB_TKT" enthalten sein muß. Weiter unten sind die
                     einzelnen Inputs in den "DB_TKT" näher erläutert.
x             [INT]: Stellgröße              {-1000,...,0,...,1000}
                     FC-intern wird "x" wie folgt begrenzt:
                     x<-1000 -> x:=-1000 ODER x>+1000 -> x:=+1000
out:
yPLUS        [BOOL]: Takt(+) =0/1, wenn x > 0 UND "tEIN > t_min_E_A" !
yMINUS       [BOOL]: Takt(-) =0/1, wenn x < 0 UND "tEIN > t_min_E_A" !

################################################################################
Im DB_TKT müssen folgende Parameter eingegeben oder können gelesen werden:
Alle mit "[W]"     gekennzeichneten Parameter sind zwingend einzugeben
Die  mit "[WO]"    gekennzeichneten Parameter sind optional einzugeben
Alle mit "[R]"     gekennzeichneten Parameter können nur gelesen werden
Alle mit "[I]/[O]" gekennzeichneten Parameter sind identisch mit In- und Outputs
                   des zugeordneten FCxxx und können nur gelesen werden
!!FC-intern werden fehlerhaft eingegebene Parameter auf den angegebenen Werte-
  bereich begrenzt.
################################################################################
Für jeden AD-Taktgeber sind das folgende Parameter:
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
tPER        [INT]: Dauer einer Periode            {0,1,...,32767}* 10ms
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
tMIN_E_A    [INT]: Minimale Ein- / Ausschaltzeit  {0,1,...,32767}* 10ms
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
tAUS_SRW    [INT]: Ausschaltzeit bei SRW          {0,1,...,32767}* 10ms
                   SRW = Signalrichtungswechsel, d.h. von Signal "yPLUS=1"
                   soll direkt auf Signal "yMINUS=1" geschaltet werden.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Berechnungsgrundlagen:

!!Periodendauer
  Die Periodendauer besteht aus einer EIN- und AUS-SCHALTZEIT, die sich wie
  folgt berechnen:

           |x| * tPER
   tEIN = -----------    UND   tAUS = (tPER - tEIN)
              1000

!!Ist "x > 0" , dann wird "yPLUS" oder  wenn "x < 0", dann wird "yMINUS"
  eingeschaltet.
!!Gilt "t_PER=0" ODER "x=0" ODER "tEIN <= tMIN_E_A, dann "yPLUS UND yMINUS :=0"!
!!Ist  "tAUS <= tMIN_E_A, dann "yPLUS ODER yMINUS permanent :=1" !
!!Jede Periode wird mit der Einschaltzeit gestartet.
!!Bei Signalrichtungswechsel wird die Ausschaltzeit "tAUS_SRW" gestartet. Läuft
  gerade die Auszeit "tAUS", dann ist die aufgelaufene Ausschaltzeit automatisch
  Bestandteil der Ausschaltzeit "tAUS_SRW".
```

Grafische Darstellung des Zusammenhanges zwischen der analogen Stellgröße "x"
und der Dauer der taktenden Stellgrößen "yPLUS" / "yMINUS":

```
    x ^
      |
 +1000+     x=800
      +------------------------+
      |                        |    x=400
      |                        +------------------+
      |                                           |
     0+-------------------------------------------|----------------------->
      |                                           +-------------------   t
      |                                                 x=-200
      |
      |
 -1000+
      |

    y ^
      |
yPLUS +--------+ +--------+ +---+      +---+
      |########| |########| |###|      |###|
      |########| |########| |###|      |###|
     0+--------+-+--------+-+---+------+---+------+--+-+-----+--+-+-----+->
      |          1          2          3   |      4  |#|     5  |#|     6*tPER
      |                                    |         |#|        |#|
yMINUS+                                    |         +-+        +-+
      |                                 -->+  tSRW   +<--
                                           |         |
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
FC155, AD/SM : ANALOG-DIGITAL-WANDLER FÜR STELLMOTOREN "AUF/ZU"
################################################################################
Kurzbeschreibung:
Mit dem FC "AD/SM" können digitale Stellantriebe von einem analogen Normsignal
mit dem Wertebereich {0,...,1000} angesteuert werden. Aus dem absoluten Wert
dieses Signales und dem Stellungs-Istwert {0,...,1000}*0.1% des Stellantriebes
wird die Stellrichtung und die Zeitdauer des Stellbefehles "AUF"/"ZU" berech-
net. Ist die Zeitdauer kleiner als das parametrierte Totband, so erfolgt keine
Verstellung des Stellantriebes.
Die Basis für die Berechnung der Stellzeit und des Stellungsistwertes ist die
Laufzeit des Stellantriebes gemäß technischer Dokumentation. Alle Parameter des
FC "AD/SM" müssen in einen durch den "UDT_SM" definierten Datenbausteinbereich
im INTEGERFORMAT eingegeben werden und sie lassen sich vom BuB-System
verändern.
Vom BuB-System kann der Stellantrieb von der Betriebsart "AUTO" nach "HAND"
umgeschaltet und im Bereich {0,...,1000}*0.1% angesteuert werden. Über den
FC-Input "FREEZE" ist eine Sequenzverriegelung mehrerer Stellantriebe möglich.
Der FC "AD/SM" läßt eine Laufzeit- = Endlagenüberwachung nach 3 MODIS zu. Neben
dem direkten Lesezugriff auf Speicher des "DB_SM" kann aus dem "INFO-Byte" der
Zustand des Stellantriebes "AUF", "ZU", "STÖRUNG" (=Störung Laufzeitüberwa-
chung!), "NEUWERT STÖRUNG" und "HAND" jederzeit ermittelt werden.
################################################################################

!!Dieser FC verändert nicht die Adressregister AR1/AR2 UND versorgt für FUP- UND
  KOP-Anwendungen die "ENO"-Box, indem das VKE=1 im BIE-BIT gespeichert wird.
!!Der FC "AD/SM" kann im Weckalarm aufgerufen werden.

in:
FIRST_SCAN  [BOOL]: Im 1.OB1-ZYKLUS nach OB100/101/102=TRUE, sonst FALSE
                    FIRST_SCAN wird zur Initialisierung unbedingt benötigt!

QUITT       [BOOL]: =0->1 die "STÖRUNG" wird mit internem PULS quittiert,
                    liegt aber solange an, bis die Ursache beseitigt ist. STÖ-
                    RUNGSURSACHEN (Siehe MOD !) gelten als beseitigt, wenn bei
                    MOD 0: Es wird keine Störung generiert
                    MOD 1: Die geforderte Endlage muß anliegen ODER der SM fährt
                           aus den Endlagenstellungen heraus -> EA=0 UND EZ=0.
                    MOD 2: Die Endlage EZ (=EZ ODER =EA) muß anliegen ODER der
                           SM fährt aus den Endlagenstellungen heraus UND der
                           SIGNALVERLAUF von "EZ" wird "1-0-1".
```

```
FREEZE      [BOOL]: Sequenzverriegelung:
                    =0 -> der Schaltzustand von "yZ/yA" wird wie beschrieben
                          berechnet.
                    =1 -> der momentane Stellungs-Iswert "y_PRZ"/"y_TIME" wird
                          eingefroren. Die Ausgänge "yZ/yA" nehmen dann folgende
                          Schaltzustände an:
                          0 < y_PRZ < 1000  ->  yZ:=yA:=0
                          y_PRZ=0           ->  yZ:=1 + Endlagenüberwachung
                          y_PRZ=1000        ->  yA:=1 + Endlagenüberwachung
PRV38BYTE [POINTER]: Pointer auf die Anfangsadresse eines  38 BYTE langen DB-
                     Speicherbereiches, dessen Struktur im "UDT_SM" festgelegt
                     ist und der für jeden Stellmotor separat im "DB_SM" vor-
                     handen sein muß.
x            [INT]: Analoge  Stellgröße                  {0,1,...,1000}*0.1%
EZ          [BOOL]: Rückmeldung Endlage "ZU"
                    MOD=0: Rückmeldung Endlage =0/1-> keine Bedeutung
                    MOD=1: Rückmeldung Endlage =1  -> Endlage   =   ZU
                                               =0  -> Endlage nicht ZU
                    MOD=2: Rückmeldung Endlage =1  -> Endlage   =   ZU ODER AUF
                                               =0  -> Endlage weder ZU noch AUF
EA          [BOOL]: Rückmeldung Endlage "AUF"
                    MOD=0: Rückmeldung Endlage =0/1-> keine Bedeutung
                    MOD=1: Rückmeldung Endlage =1  -> Endlage   =   AUF
                                               =0  -> Endlage nicht AUF
                    MOD=2: Rückmeldung Endlage =0/1-> keine Bedeutung
MS          [BOOL]: =0 -> STÖRUNG: Die Motorschutzeinrichtung hat angesprochen

out:
yZ          [BOOL]: Digitale Stellgröße         =1  -> Stellmotor = ZU
yA          [BOOL]: Digitale Stellgröße         =1  -> Stellmotor = AUF
ST_LZ       [BOOL]: =1 -> Störung Laufzeit-/Endlagenüberwachung
                     !!AUS DER STÖRUNG "ST_LZ" kann diagnostiziert werden, ob
                       der Stellantrieb einen mechanischen oder elektrischen
                       Defekt hat, die Endlagenschalter nicht mehr funktionieren
                       oder die elektrischen Verbindungen unterbrochen sind.
ST_MS      [BOOL]: =1 -> Störung Motorschutzeinrichtung
SST        [BOOL]: =1 -> Sammelstörung
                     !!Sammelstörung "SST=1" bewirkt die sofortige Ausschaltung
                       des Stellantriebes -> "yZ=yA:=0"!
INFO       [BYTE]: Informationsbyte über den Zustand des SM, mit dem BIT-Muster
                   "0sqAZaz0, wobei die BITS folgende Bedeutung haben:
                   BIT 0   = 0 : ohne Bedeutung
                   BIT 1   = z : =1  -> der SM = ZU
                   BIT 2   = a : =1  -> der SM = AUF
                   BIT 3   = Z : =1  -> der SM fährt ZU  / schließt
                   BIT 4   = A : =1  -> der SM fährt AUF / öffnet
                   BIT 5   = q : =1  -> Störung = Neuwert,nach Quittierung :=0
                   BIT 6   = s : =1  -> Störung Laufzeitüberwachung/Motorschutz
                   BIT 7   = 0 : ohne Bedeutung

!!Die Rückmeldungen der Endlagen beeinflussen NICHT den Zustand der Ausgänge .
!!Den Schaltzustand der Ausgänge beeinflussen nur "x", "FREEZE" UND "SST".
!!Bei Erstinbetriebnahme einer Anlage ODER Remanenzverlust ODER nach Austausch
  eines defekten Stellmotors ist es möglich, daß der BERECHNETE STELLUNGSISTWERT
  und die für MOD 0+2 BERECHNETEN ENDLAGENSTELLUNGEN mit den tatsächlichen HARD-
  WAREPOSITIONEN nicht übereinstimmen. Es kann zu einmaligen Störmeldungen kom-
  men, deren Ursache in undefinierten Speicherwerten des "DB_SM" zu suchen ist.
  Nach Quittieren der Störung ist diese Störungsursache dauerhaft beseitigt.
  Die berechneten Positionen korrigieren sich automatisch, weil der SM in den
  ENDLAGENSTELLUNGEN permanent das Signal "AUF=1" ODER "ZU=1" erhält.

!!Der Input "x" und alle Eingabewerte in den "DB_SM" werden "FC_SM" intern
  auf zulässige Werte überprüft und auf die angegebenen Grenzen korrigiert.
################################################################################
Im "DB_SM" müssen folgende Parameter eingegeben oder können gelesen werden:
Alle mit "[W]"     gekennzeichneten Parameter sind zwingend einzugeben
Die  mit "[WO]"    gekennzeichneten Parameter sind optional einzugeben
Alle mit "[R]"     gekennzeichneten Parameter können nur gelesen werden
Alle mit "[I]/[O]" gekennzeichneten Parameter sind identisch mit In- und Outputs
                   des zugeordneten FCxxx und können nur gelesen werden
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
MOD     [INT]: MOD=0 -> Keine Rückmeldung der Endlagen, keine Überwachung
               MOD=1 -> Rückmeldung der Endlagen UND Überwachung
```

```
              MOD=2 -> Rückmeldung "EZ"("EA" ODER "EZ") UND Überwachung
              Fehleingaben: MOD<0 ODER MOD>2 -> MOD:=0 !
!!Auch bei MOD=0 muß eine Stellzeit, das ist die Zeit in welcher der Stellan-
  trieb von einer Endlage zur anderen fährt, eingegeben werden. Nach Ablauf
  dieser Stellzeit werden im "DB_SM" die Endlagen aktualisiert, die entspre-
  chend Stellrichtung erreicht würden:

  Die erreichte ENDLAGE wird bei MOD=0 im DB_SM in den Speicherstellen
  "E_ZU_INFO" und "E_AUF_INFO" nach folgenden Kriterien hinterlegt und
  analog dazu in das INFOBYTE geschrieben:
    E_ZU_INFO [BOOL]: =1, wenn STELLBEFEHL=ZU  UND STELLUNGSISTWERT yPRZ=0
    E_AUF_INFO[BOOL]: =1, wenn STELLBEFEHL=AUF UND STELLUNGSISTWERT yPRZ=1000

!!Bei MOD=1 muß beim Stellbefehl "y_ZU" die Rückmeldung der Endlage "EZ" bzw.
  beim SB "y_AUF" die Rückmeldung der Endlage "EA" erfolgen. Liegen beide End-
  lagensignale gleichzeitig an ODER liegt nach dem Erreichen derEndlagenposi-
  tion UND nach Ablauf der Überwachungszeit "tUE" nicht das der Endlagenposition
  entsprechende Endlagensignal an, dann wird "STÖRUNG:=1" gesetzt.
  Die "STÖRUNG" wird zurückgesetzt, WENN QUITTIERT WURDE UND KEINE STÖRUNGS-
  SITUATION ANLIEGT.
  Die erreichte ENDLAGE wird bei MOD=1 im DB_SM in den Speicherstellen
  "E_ZU_INFO" und "E_AUF_INFO" hinterlegt und analog dazu in das INFOBYTE
  geschrieben:
    E_ZU_INFO [BOOL]: =EZ, ist also identisch mit INPUT ENDLAGE "EZ"
    E_AUF_INFO[BOOL]: =EA, ist also identisch mit INPUT ENDLAGE "EA"

!!Bei MOD=2 sind die Endlagenschalter hardwareseitig parallel zu schalten.
  Es tritt dann am Eingang "EZ" folgender Signalverlauf auf:
  ---------------------------------------------------------+----------------------
  Stellbefehl=SB  UND  Ausgangstellung=AS                  |Signalverlauf an "EZ"
  ---------------------------------------------------------+----------------------
  SB  x=100%             ->"AUF" + AS "ZU"                 |  1 -> 0 -> 1
  ---------------------------------------------------------+----------------------
  SB  x=100%             ->"AUF" + AS "ZU<y<AUF"           |  0 -> 1
  ---------------------------------------------------------+----------------------
  SB  0%<(x>y_PRZ)<100% ->"AUF" + AS "ZU<y<AUF"            |  0 -> 0
  ---------------------------------------------------------+----------------------
  SB  0%<(x<y_PRZ)<100% ->"ZU"  + AS "ZU<y<AUF"            |  0 -> 0
  ---------------------------------------------------------+----------------------
  SB  x=0%               ->"ZU"  + AS "ZU<y<AUF"           |  0 -> 1
  ---------------------------------------------------------+----------------------
  SB  x=0%               ->"ZU"  + AS "AUF"                |  1 -> 0 -> 1
  ---------------------------------------------------------+----------------------
  Tritt der Signalverlauf "1-0-1" nicht ein ODER bleibt nach Ablauf der Über-
  wachungszeit die Rückmeldung "EZ"=0, dann wird der Antrieb als gestört gemel-
  det.
  Die "STÖRUNG" wird zurückgesetzt, WENN QUITTIERT WURDE UND KEINE STÖRUNGS-
  SITUATION ANLIEGT.

  Die erreichte ENDLAGE wird bei MOD=2 im DB_SM in den Speicherstellen
  "E_ZU_INFO" und "E_AUF_INFO" nach folgenden Kriterien hinterlegt und
  analog dazu in das INFOBYTE geschrieben:
    E_ZU_INFO [BOOL]: :=1, wenn STELLBEFEHL=ZU  UND STELLUNGSISTWERT yPRZ=0
                           UND  EZ=1 MIT  DEM SIGNALVERLAUF "?-0-1"!

    E_AUF_INFO[BOOL]: :=1, wenn STELLBEFEHL=ZU  UND STELLUNGSISTWERT yPRZ=1000
                           UND  EZ=1 MIT DEM SIGNALVERLAUF "?-0-1" !
!!"E_ZU_INFO" UND/ODER "E_AUF_INFO" werden immer dann zurückgesetzt, wenn "EZ=0"
  IST ODER die STÖRUNG QUITTIERT wird.
!!Für den FALL, daß "EZ=1" permanent anliegt, nehmen die Stellungsrückmeldungen
  NACH QUITTIERUNG der STÖRUNG immer den oben beschriebenen Zustand an.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
tLZ     [INT]: Laufzeit=Stellzeit "ZU <->"AUF"        {1,2,...,32767}*0.1s
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
tUE     [INT]: Überwachungszeit                       {1,2,...,32767}*0.1s
              Bei den SB "xZ" und "xA" wird nach Ablauf der Stellzeit die
              Überwachungszeit "tUE" gestartet. Wenn nach Ablauf von "tUE"
              nicht die oben beschriebene Rückmeldung der Endlagen "EZ" bzw.
              "EA" anliegt, wird das Störungsbit "STÖRUNG" gesetzt. Die End-
              lagenüberwachung ist bei MOD=0 unwirksam.
!!Liegt die Störung Laufzeit an, so wird nach Betätigung der Quittiertaste die
  Störung zurückgesetzt und die Überwachungszeit neu gestartet.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

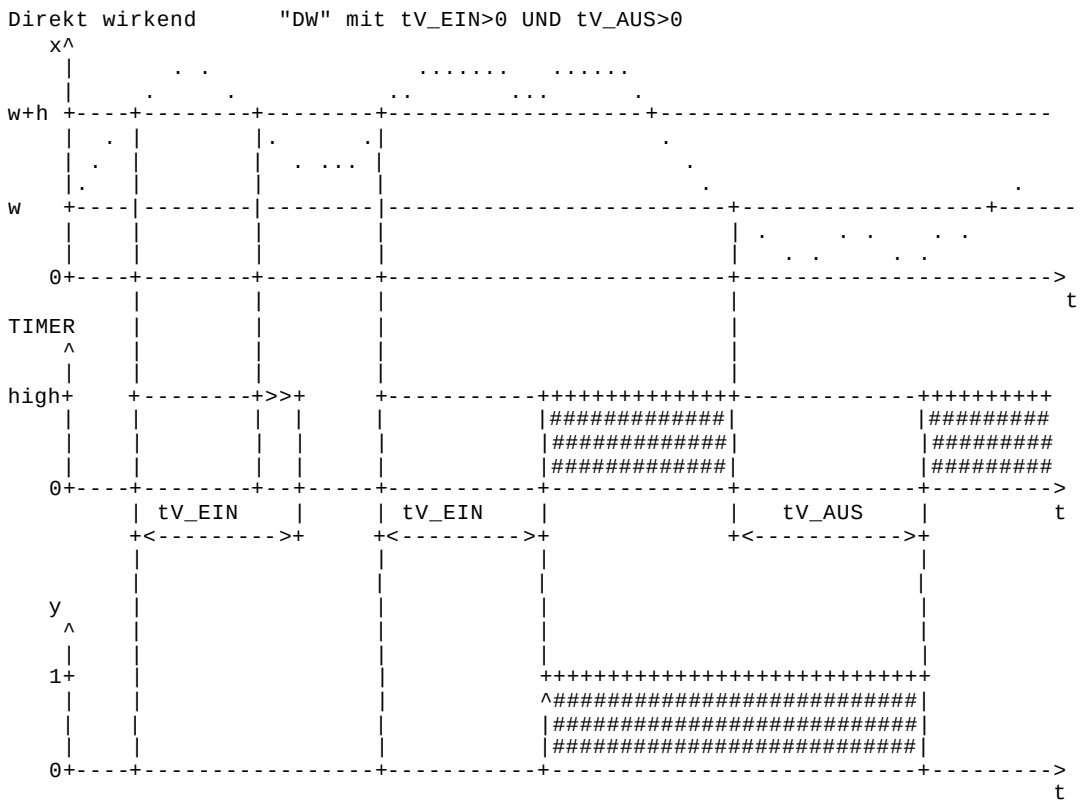
```
tTB     [INT]: Totband                                    {0,1,...,32767}*0.01s
               Der analoge Stellbefehl "x" wird in eine Stellzeit
                              x*tLZ*100
                     x_TIME= ---------- [ms]
                               1000%
               umgerechnet und der Stellungs-Istwert in "y_TIME" gespeichert.
               Verändert sich "x_TIME" in den Grenzen

                       (y_TIME-tTB)<= x_TIME <= (y_TIME+tTB)

               wobei alle Werte in [ms] umgerechnet sind, dann bleiben die
               digitalen Stellungssignale "yZ" UND "yA" ausgeschaltet!

               !!Das Totband ist unwirksam für die Endstellungen:
                 x=0%   -> SB "ZU" :=1
                 x=100% -> SB "AUF":=1
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
tMIN_AUS [INT]: Minimale Auszeit für den SB ZU/AUF    {0,1,...,32767}*0.01s
               Bei jedem Wechsel des Stellbefehles von AUF->ZU oder ZU->AUF
               wird die Auszeit aktiviert.
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
x_HAND   [INT]: Handstellbefehl                          {0,1,...,1000}*0.1%
               Von der BuB-Ebene kann der Stellmotor von "HAND" angesteuert
               werden. Dabei sind Handstellbefehle "TAST_ZU" bzw. "TAST_AUF"
               ODER eine HAND-Stellungsvorgabe in [%] möglich.
!!Das Handstellsignal "x_HAND" nimmt folgende Werte an:
    HAND = 0                                      ->  x_HAND:=0
    HAND = Flanke(0-1) ODER HAND=ZU ODER HAND=AUF ->  x_HAND:=Istwert "y_PRZ"
    HAND = 1  UND NICHT HAND=ZU/AUF               ->  x_HAND:=Eingabewert
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
TAST_AH       [BOOL]: Taste vom BuB: FLIP-FLOP-Umschaltung AUTO/HAND
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
TAST_ZU       [BOOL]: Taste vom BuB: Mit PULS(+) Umschaltung HAND=ZU
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
TAST_AUF      [BOOL]: Taste vom BuB: Mit PULS(+) Umschaltung HAND=AUF
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
TAST_STOP_GO  [BOOL]: Taste vom BuB mit folgender Doppelfunktion:
                      = 1 -> PERMANENT RESET HAND-STELLBEFEHL ZU/AUF
                      = 0 -> die HAND-STELLBEFEHLE ZU/AUF können wieder per
                             Taster ausgelöst werden.
                      = mit FLIP-FLOP-FUNKTION können die über die Eingabe von
                        x_HAND ausgelösten Stellbefehle gestoppt oder wieder
                        freigegeben werden.
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
QUITT_BuB     [BOOL]: =0->1 die "STÖRUNG" wird mit internem PULS quittiert
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Begriffe:

x_TIME  = Der analoge Stellbefehl "x" wird in eine Stellzeit
                    x*tLZ*100
          x_TIME = ---------- [ms]  umgerechnet      {0,1,...,3276700}*0.001s
                    1000%
y_TIME  = Istwert der Stellzeit                      {0,1,...,3276700}*0.001s
          Der Istwert der Stellzeit nähert sich immer dem Wert x_TIME, wenn sich
          x_TIME nicht ändert.
y_PRZ   = Die Umrechnung des Stellungsistwertes in [%]  {0,1,...,1000}*0.1%
                   y_TIME*1000%
          y_PRZ = --------------
                      tLZ*100
          Der Istwert der Stellzeit in [%], also "y_PRZ", nähert sich immer dem
          Wert "x", wenn sich "x" nicht ändert.
tTB     = Totband                                     {0,1,...,32767}*0.01s
          (y_TIME-tTB)<= x_TIME <= (y_TIME+tTB)
          dann bleiben die digitalen Stellungssignale "yZ"+"yA" ausgeschaltet!

!!Die Formulierung "nähert sich" resultiert daraus, daß die kleinste Stellzeit
  von der Zykluszeit der CPU abhängt.Es ist Zufall, wenn der Zustand "y_PRZ=x"
  bzw. "y_TIME=x_TIME" bei einer Stellungsänderung sofort eintritt. Das Programm
  ist so aufgebaut, daß es für den Fall Totband "tTB=0", unbedingt den Aus-
  gleich "y_PRZ=x" herstellen will, wenn ein Unterschied zwischen beiden Werten
  vorhanden ist. Das Totband "tTB>0" verhindert ein eventuelles Pendeln des di-
  gitalen Stellsignales.
```

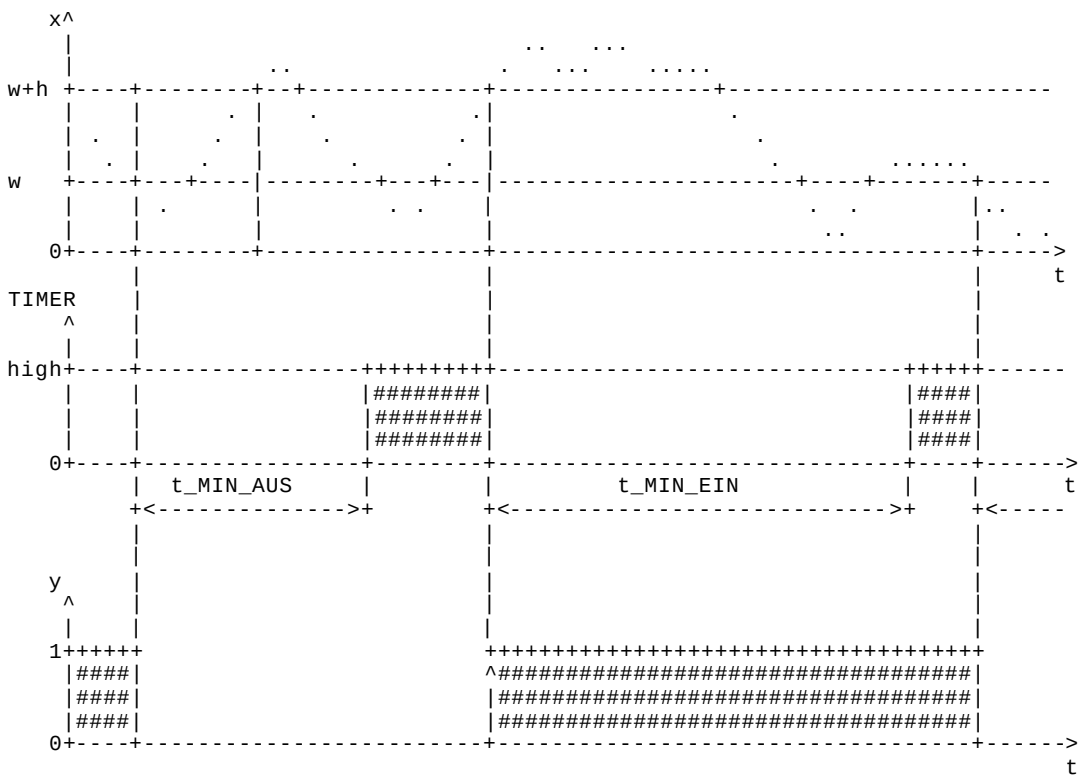
```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
FC156, HYST_SWITCH : HYSTERESESCHALTER
################################################################################
Kurzbeschreibung:
Der FC "HYST_SWITCH" stellt einen Hystereseschalter dar, dessen Parameter in
einen durch den "UDT_HY" definierten Datenbausteinbereich eingegeben werden
müssen und die vom BuB-System veränderbar sind. Der FC-Input "SH" ermöglicht
Sollwertschiebungen für Kaskadenregelungen. Ober- und Untergrenze der Sollwert-
schiebung sind parametrierbar. Mit den Inputs "ENABLE" und "FREEZE" können Se-
quenzverriegelungen programmiert werden. Die Wirkungsrichtung des Hysterese-
stereseschalters, direkt- oder umgekehrt wirkend, ist konfigurierbar.
################################################################################
!!Dieser FC verändert nicht die Adressregister AR1/AR2 UND versorgt für FUP- UND
  KOP-Anwendungen die "ENO"-Box, indem das VKE=1 im BIE-BIT gespeichert wird.


in:
ENABLE      [BOOL]: =0 -> DISABLE: y:=yUW:=yDW:=0;  =1 -> ENABLE: Berechnung "y"
                    !!"ENABLE" hat Vorrang vor "FREEZE"
FREEZE      [BOOL]: Sequenzverriegelung:
                    =0 -> der Schaltzustand von "y" ist wie beschrieben von
                          den Parametern "ENABLE", "x", w_eff und "h" abhängig
                    =1 -> der momentane Schaltzustand von "y" wird eingefroren
                          und ist von "x" unabhängig. "ENABLE" ist wirksam!
PRV18BYTE[POINTER]: Pointer auf die Anfangsadresse eines 18 BYTE langen DB-
                    Speicherbereiches, dessen Struktur im "UDT_HY" festgelegt
                    ist und der für jeden Hystereseschalter separat im "DB_HY"
                    enthalten sein muß. Weiter unten sind die einzelnen Inputs
                    in den "DB_HY" näher erläutert.
x            [INT]: Istwert  der Regelgröße          {-32768,...,0,...,+32767}
SH           [INT]: Sollwertschiebung w:=w+SH        {-32768,...,0,...,+32767}
                    Die Schiebeweite wird durch die Eingabewerte "w_SH_MIN" und
                    und "w_SH_MAX" im "DB_HY" wie folgt begrenzt:
                    SH>0:= SH(+) -> [w+SH(+)]<=wSH_MAX;   w > wSH_MAX -> SH:=0
                    SH<0:= SH(-) -> [w+SH(-)]>=wSH_MIN;   w < wSH_MIN -> SH:=0

out:
y           [BOOL]: Stellgröße UW  x <= w-h UND  ENABLE=1 -> y:=1
                                   x >= w   ODER ENABLE=0 -> y:=0
                    Stellgröße DW  x >= w+h UND  ENABLE=1 -> y:=1
                                   x <= w   ODER ENABLE=0 -> y:=0

!!FC-intern werden alle Werte im "DINT-Format" verglichen. Bei ungünstigen
  Wertekombinationen  kann es vorkommen, daß "w-h" oder "w+h" außerhalb
  des Wertebereiches von "x" liegen. In solchen Fällen ist "y" permanent :=0.
!!Alle Eingabewerte in den "DB_HY" werden im "FC_HY_SWITCH" auf die angegebenen
  Grenzen geprüft. Sind Eingaben fehlerhaft, so werden sie auf die zulässige und
  angegebene Untergrenze=UG bzw. Obergrenze=OG FC-intern korrigiert.
################################################################################
Im "DB_HY" müssen folgende Parameter eingegeben oder können gelesen werden:
Alle mit "[W]"     gekennzeichneten Parameter sind zwingend einzugeben
Alle mit "[R]"     gekennzeichneten Parameter können nur gelesen werden
Alle mit "[I]/[O]" gekennzeichneten Parameter sind identisch mit In- und Outputs
                   des zugeordneten FCxxx und können nur gelesen werden
################################################################################
Für jeden HYSTERESESCHALTER sind das die Parameter:
UW_DW  [BOOL]: =0 -> umgekehrt wirkend; =1 -> direkt wirkend
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
w       [INT]: Sollwert, Führungsgröße            {-32768,...,0,...,+32767}
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
w_SH_MIN[INT]: MINIMUM für  "w := [w+SH(-)]"      {-32768,...,0,...,+32767}
               Gilt bereits "w < w_SH_MIN", dann kann "w" durch SH(-) nicht noch
               zu kleineren Werten geschoben werden.
               !! w_SH_MIN ist keine Begrenzung für die Sollwerteingabe "w" !!
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
w_SH_MAX[INT]: MAXIMUM für "w := [w+SH(+)]"       {-32768,...,0,...,+32767}
               Gilt bereits "w > w_SH_MAX", dann kann "w" durch SH(+) nicht noch
               zu größeren Werten geschoben werden.
               !! w_SH_MAX ist keine Begrenzung für die Sollwerteingabe "w" !!
!! Relationen, wie w_SH_MIN >= w_SH_MAX verursachen lediglich, und das ist ab-
   hängig von der Lage von "w", daß nur SH(-) ODER nur SH(+) ODER keine Soll-
   wertschiebung möglich ist.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

```
h       [INT]: Hysterese                                        {1,2,...,32767}
         Fehleingabe h<=0 -> FC-intern h:=1
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Begriffe:
UW_DW    = Wirkrichtung des Hystereseschalters UW_DW  {0,1}
x        = Istwert  der Regelgröße                x  {-32768,...,0,...,+32767}
w        = Sollwert der Regelgröße                w  {-32768,...,0,...,+32767}
w_eff    = wirkender Sollwert           w_eff=(w+SH) {-32768,...,0,...,+32767}
SH       = Sollwertschiebung                     SH {-32768,...,0,...,+32767}
SH(-)    = negativer Bereich von SH             SH(-) {-32768,...,-1,0}
SH(+)    = positiver Bereich von SH              SH(+) {0,+1,...,+32767}
w_SH_MIN = MINIMUM für w:=[w+SH(-)]          w_SH_MIN {-32768,...,0,...,+32767}
           Gilt bereits "w < w_SH_MIN", dann kann "w" durch SH(-) nicht noch
           zu kleineren Werten geschoben werden.
           !!"w_SH_MIN" ist keine Begrenzung für die Sollwerteingabe "w" !!
w_SH_MAX = MAXIMUM für w:=[w+SH(+)]          w_SH_MAX {-32768,...,0,...,+32767}
           Gilt bereits "w > w_SH_MAX", dann kann "w" durch SH(+) nicht noch
           zu größeren Werten geschoben werden.
           !!"w_SH_MAX" ist keine Begrenzung für die Sollwerteingabe "w" !!
           !!Relationen, wie w_SH_MIN >= w_SH_MAX verursachen lediglich, und
           das ist abhängig von der Lage von "w", daß nur SH(-) ODER nur SH(+)
           ODER keine Sollwertschiebung möglich ist.
h        = Hysterese  (h<=0 -> h:=1)               h {1,2,...,+32767}
y        = Stellgröße umgekehrt ODER direkt wirkend y {0,1}

Beispiele:

Umgekehrt wirkend   "UW"
                    w1:=w+SH1 UND SH1<0 !
                    w2:=w+SH2 UND SH2>0 !
   Y^
    |
   1+     +--------+--           +--------+--           +--------+--
    |     ^        .            ^        |            ^        .
    |     .        .            |        |            .        .
    |     .        .            |        |            .        .
    |     .        .            |        |            .        .
   0+-----+--------+-------------+--------+-------------+--------+------->
          w1-h     w1            w-h      w             w2-h    w2      x
          |<-- h --|            |<-- h --|             |<-- h --|


Direkt wirkend      "DW"
                    w1:=w+SH1 UND SH1<0 !
                    w2:=w+SH2 UND SH2>0 !
   Y^
    |
   1+     +--------+--           +--------+--           +--------+--
    |     .        ^            |        ^            .        ^
    |     .        .            |        |            .        .
    |     .        .            |        |            .        .
    |     .        .            |        |            .        .
   0+-----+--------+-------------+--------+-------------+--------+------->
          w1       w1+h          w        w+h          w2       w2+h     x
          |-- h -->|            |-- h -->|             |-- h -->|

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
FC157, HYST_SWITCH_tV : HYSTERESE-SCHALTER MIT VERZÖGERUNGSZEITEN
################################################################################
Kurzbeschreibung:
Der FC "HYST_SWITCH_tV" stellt einen Hystereseschalter dar, dessen Parameter in
einen durch den "UDT_HY_tV" definierten Datenbausteinbereich eingegeben werden
müssen und die vom BuB-System veränderbar sind. Der FC-Input "SH" ermöglicht
Sollwertschiebungen für Kaskadenregelungen. Ober- und Untergrenze der Sollwert-
schiebung sind parametrierbar. Mit den Inputs "ENABLE" und "FREEZE" sind Se-
quenzverriegelungen möglich.
Die EIN- und AUS-Schaltungen können zeitverzögert werden. Dabei sind die Varian-
ten EINSCHALTZEITVERZÖGERUNG/MINDESTAUSCHALTZEIT bzw. AUSSCHALTZEITVERZÖGERUNG/
MINDESTEINSCHALTZEIT frei wählbar. Die Wirkungsrichtung des Hystereseschalters,
direkt- oder umgekehrt wirkend, ist konfigurierbar.
################################################################################
```

```
!!Dieser FC verändert nicht die Adressregister AR1/AR2 UND versorgt für FUP- UND
  KOP-Anwendungen die "ENO"-Box, indem das VKE=1 im BIE-BIT gespeichert wird.

in:
FIRST_SCAN  [BOOL]: Im 1.OB1-ZYKLUS nach OB100/101/102=TRUE, sonst FALSE
                    !!Mit FIRST_SCAN wird nur der UNI_TIMER initialisiert
ENABLE      [BOOL]: =0 -> DISABLE: y:=0;  =1 -> ENABLE: Berechnung von "y"
                    !!"ENABLE" hat Vorrang vor "FREEZE"
FREEZE      [BOOL]: Sequenzverriegelung:
                    =0 -> der Schaltzustand von "y" ist wie beschrieben von
                          den Parametern "ENABLE", "x", w_eff und "h" UND dem
                          Zustand der TIMER abhängig
                    =1 -> der momentane Schaltzustand von "y" wird eingefroren
                          und ist von "x" unabhängig. "ENABLE" ist wirksam!
                          Die TIMER laufen ab entsprechend MOD_tx und Schalt-
                          zustand ab.
PRV32BYTE[POINTER]: Pointer auf die Anfangsadresse eines 32 BYTE langen DB-Spei-
                    cherbereiches, dessen Struktur im "UDT_HY_tV" festgelegt ist
                    und der für jeden Hystereseschalter separat im "DB_HY_tV"
                    enthalten sein muß. Weiter unten sind die einzelnen Inputs
                    in den "DB_HY_tV" näher erläutert.
x            [INT]: Istwert  der Regelgröße          {-32768,...,0,...,+32767}
SH           [INT]: Sollwertschiebung w:=w+SH        {-32768,...,0,...,+32767}
                    Die Schiebeweite wird durch die Eingabewerte "w_SH_MIN" und
                    und "w_SH_MAX" im "DB_HY" wie folgt begrenzt:
                    SH>0:= SH(+) -> [w+SH(+)]<=wSH_MAX;   w > wSH_MAX -> SH:=0
                    SH<0:= SH(-) -> [w+SH(-)]>=wSH_MIN;   w < wSH_MIN -> SH:=0

out:
y           [BOOL]: Stellgröße UW x <= w-h UND  ENABLE=1 UND tV_EIN=HIGH -> y:=1
                                   x >= w   ODER ENABLE=0 UND tV_AUS=HIGH -> y:=0
                    Stellgröße DW x >= w+h UND  ENABLE=1 UND tV_EIN=HIGH -> y:=1
                                   x <= w   ODER ENABLE=0 UND tV_AUS=HIGH -> y:=0

!!Ist MOD_tV_xxx=0, dann wird der TIMER immer dann neu gestartet, wenn sich der
  Schaltbefehl für "y" geändert hat - sonst wird er zurückgesetzt.
  Ist MOD_tV=1 wird der TIMER sofort gestartet, wenn sich der Schaltzustand von
  "y" ändert.Das gilt auch wenn FREEZE=1 ist!
!!FC-intern werden alle Werte im "DINT-Format" verglichen. Bei ungünstigen
  Wertekombinationen  kann es vorkommen, daß "w-h" oder "w+h" außerhalb
  des Wertebereiches von "x" liegen. In solchen Fällen ist "y" permanent :=0.
!!Alle Eingabewerte in den "DB_HY_tV" werden im "FC_HY_SWITCH_tv" auf die ange-
  gebenen Grenzen geprüft. Sind Eingaben fehlerhaft, so werden sie auf die zu-
  lässige und angegebene Untergrenze=UG bzw. Obergrenze=OG FC-intern korrigiert.
################################################################################
Im "DB_HY_tV" müssen folgende Parameter eingegeben oder können gelesen werden:
Alle mit "[W]"     gekennzeichneten Parameter sind zwingend einzugeben
Alle mit "[R]"     gekennzeichneten Parameter können nur gelesen werden
Alle mit "[I]/[O]" gekennzeichneten Parameter sind identisch mit In- und Outputs
                   des zugeordneten FCxxx und können nur gelesen werden
################################################################################
Für jeden HYSTERESESCHALTER mit "tV" sind das die Parameter:
UW_DW  [BOOL]: =0 -> umgekehrt wirkend; =1 -> direkt wirkend
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
w       [INT]: Sollwert, Führungsgröße            {-32768,...,0,...,+32767}
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
w_SH_MIN[INT]: MINIMUM für  "w := [w+SH(-)]"      {-32768,...,0,...,+32767}
               Gilt bereits "w < w_SH_MIN", dann kann "w" durch SH(-) nicht noch
               zu kleineren Werten geschoben werden.
               !! w_SH_MIN ist keine Begrenzung für die Sollwerteingabe "w" !!
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
w_SH_MAX[INT]: MAXIMUM für "w := [w+SH(+)]"       {-32768,...,0,...,+32767}
               Gilt bereits "w > w_SH_MAX", dann kann "w" durch SH(+) nicht noch
               zu größeren Werten geschoben werden.
               !! w_SH_MAX ist keine Begrenzung für die Sollwerteingabe "w" !!
!! Relationen, wie w_SH_MIN >= w_SH_MAX verursachen lediglich, und das ist ab-
   hängig von der Lage von "w", daß nur SH(-) ODER nur SH(+) ODER keine Soll-
   wertschiebung möglich ist.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
h       [INT]: Hysterese                                {1,2,...,32767}
               Fehleingabe h<=0 -> FC-intern h:=1
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
MOD_tV_EIN[BOOL]: MODUS von tV_EIN
```

```
                =0 -> Einschalt-Zeitverzögerung = tV_EIN      für "y"
                =1 -> Minimale Ausschaltzeit    = tMIN_AUS    für "y"
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
MOD_tV_AUS[BOOL]: MODUS von tV_AUS
                =0 -> Ausschalt-Zeitverzögerung = tV_AUS      für "y"
                =1 -> Minimale Einschaltzeit    = tMIN_EIN    für "y"
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
tV_EIN     [INT]: tV_EIN ODER t_MIN_AUS                  {0,1,...,32767}*0.1s
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
tV_AUS     [INT]: tV_AUS ODER t_MIN_EIN                  {0,1,...,32767}*0.1s
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Begriffe:
UW_DW    = Wirkrichtung des Hystereseschalters UW_DW  {0,1}
x        = Istwert  der Regelgröße                x  {-32768,...,0,...,+32767}
w        = Sollwert der Regelgröße                w  {-32768,...,0,...,+32767}
w_eff    = wirkender Sollwert           w_eff=(w+SH) {-32768,...,0,...,+32767}
SH       = Sollwertschiebung                     SH {-32768,...,0,...,+32767}
SH(-)    = negativer Bereich von SH             SH(-) {-32768,...,-1,0}
SH(+)    = positiver Bereich von SH             SH(+) {0,+1,...,+32767}
w_SH_MIN = MINIMUM für w:=[w+SH(-)]          w_SH_MIN {-32768,...,0,...,+32767}
           Gilt bereits "w < w_SH_MIN", dann kann "w" durch SH(-) nicht noch
           zu kleineren Werten geschoben werden.
           !!"w_SH_MIN" ist keine Begrenzung für die Sollwerteingabe "w" !!
w_SH_MAX = MAXIMUM für w:=[w+SH(+)]          w_SH_MAX {-32768,...,0,...,+32767}
           Gilt bereits "w > w_SH_MAX", dann kann "w" durch SH(+) nicht noch
           zu größeren Werten geschoben werden.
           !!"w_SH_MAX" ist keine Begrenzung für die Sollwerteingabe "w" !!
           !!Relationen, wie w_SH_MIN >= w_SH_MAX verursachen lediglich, und
           das ist abhängig von der Lage von "w", daß nur SH(-) ODER nur SH(+)
           ODER keine Sollwertschiebung möglich ist.
h        = Hysterese  (h<=0 -> h:=1)                 h {1,2,...,+32767}
y        = Stellgröße umgekehrt ODER direkt wirkend   y {0,1}
tV_EIN   = Einschalt-Zeitverzögerung   = tV_EIN   für y {0,1,...,32767}*0.1s
           ODER minimale Ausschaltzeit = tMIN_AUS für y
tV_AUS   = Ausschalt-Zeitverzögerung   = tV_AUS   für y {0,1,...,32767}*0.1s
           ODER Minimale Einschaltzeit = tMIN_EIN für y



Beispiele:

Direkt wirkend      "DW" UND tV_EIN=tV_AUS=0
                    w1:=w+SH1 UND SH1<0 !
                    w2:=w+SH2 UND SH2>0 !
   Y^
    |
   1+     +--------+--           +--------+--           +--------+--
    |     .        ^            |        ^            .        ^
    |     .        .            |        |            .        .
    |     .        .            |        |            .        .
    |     .        .            |        |            .        .
   0+-----+--------+-------------+--------+-------------+--------+------->
          w1       w1+h          w        w+h          w2       w2+h     x
          |-- h -->|            |-- h -->|             |-- h -->|

Umgekehrt wirkend   "UW" UND tV_EIN=tV_AUS=0
                    w1:=w+SH1 UND SH1<0 !
                    w2:=w+SH2 UND SH2>0 !
   Y^
    |
   1+     +--------+--           +--------+--           +--------+--
    |     ^        .            ^        |            ^        .
    |     .        .            |        |            .        .
    |     .        .            |        |            .        .
    |     .        .            |        |            .        .
   0+-----+--------+-------------+--------+-------------+--------+------->
          w1-h     w1            w-h      w             w2-h     w2      x
          |<-- h --|            |<-- h --|             |<-- h --|
```


Direkt wirkend "DW" mit tV_EIN>0 UND tV_AUS>0
x^
w+h
w
0
TIMER
high
tV_EIN
tV_EIN
tV_AUS
t
y
1

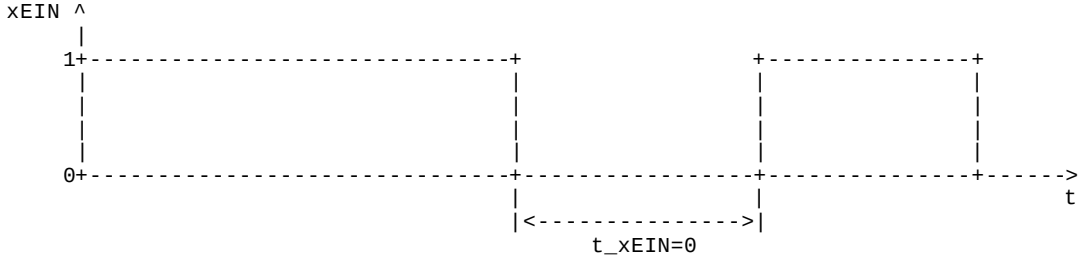

Direkt wirkend "DW" mit t_MIN_EIN>0 UND t_MIN_AUS>0
x^
w+h
w
0
TIMER
high
t_MIN_AUS
t_MIN_EIN
t
y
1

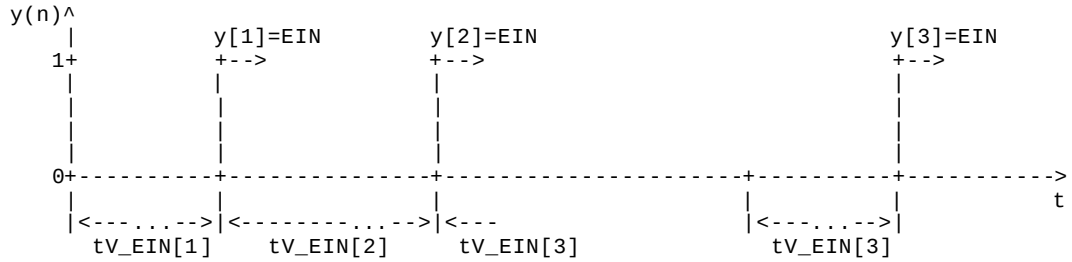
```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
FC158, 3PUNKT : 3-PUNKT-REGLER
################################################################################
Kurzbeschreibung:
Der FC "3PUNKT" stellt einen klassischen 3-PUNKT-REGLER dar, dessen Parameter in
einen durch den "UDT_3PUNKT" definierten Datenbausteinbereich eingegeben werden
müssen und die vom BuB-System veränderbar sind. Der FC-Input "SH" ermöglicht
Sollwertschiebungen für Kaskadenregelungen. Ober- und Untergrenze der Sollwert-
schiebung sind parametrierbar.
################################################################################
!!Dieser FC verändert nicht die Adressregister AR1/AR2 UND versorgt für FUP- UND
  KOP-Anwendungen die "ENO"-Box, indem das VKE=1 im BIE-BIT gespeichert wird.

in:
ENABLE      [BOOL]: =0 -> DISABLE: y:=yUW:=yDW:=0;  =1 -> ENABLE: Berechnung "y"
PRV18BYTE[POINTER]: Pointer auf die Anfangsadresse eines 18 BYTE langen DB-
                    Speicherbereiches, dessen Struktur im "UDT_3PUNKT" festge-
                    legt ist und der für jeden 3-Punkt-Regler separat im Daten-
                    baustein "DB_3PUNKT" enthalten sein muß. Weiter unten sind
                    die einzelnen Inputs in den "DB_3PUNKT" näher erläutert.
x            [INT]: Istwert  der Regelgröße          {-32768,...,0,...,+32767}
SH           [INT]: Sollwertschiebung w:=w+SH        {-32768,...,0,...,+32767}
                    Die Schiebeweite wird durch die Eingabewerte "w_SH_MIN" und
                    und "w_SH_MAX" im "DB_3PKT"wie folgt begrenzt:
                    SH>0:= SH(+) -> [w+SH(+)]<=wSH_MAX;   w > wSH_MAX -> SH:=0
                    SH<0:= SH(-) -> [w+SH(-)]>=wSH_MIN;   w < wSH_MIN -> SH:=0

out:
yUW         [BOOL]: Stellgröße UW  x <= w-h UND  ENABLE=1 -> y:=1        {0,1}
                                   x  = w   ODER ENABLE=0 -> y:=0
yDW         [BOOL]: Stellgröße DW  x >= w+h UND  ENABLE=1 -> y:=1        {0,1}
                                   x  = w   ODER ENABLE=0 -> y:=0
!!Wird "w_eff" UND/ODER "h" geändert, dann wird der Schaltzustand von yUW und
  yDW neu berechnet.
!!FC-intern werden alle Werte im "DINT-Format" verglichen. Bei ungünstigen
  Wertekombinationen  kann es vorkommen, daß "w-h" oder "w+h" außerhalb
  des Wertebereiches von "x" liegen. In solchen Fällen ist "yUW" ODER "yDW"
  permanent :=0.
!!Alle Eingabewerte in den "DB_3PKT" werden im "FC_3PUNKT_REGLER" auf die
  angegebenen Grenzen geprüft. Sind Eingaben fehlerhaft, so werden sie auf die
  zulässige Untergrenze=UG bzw. Obergrenze=OG FC-intern korrigiert.
################################################################################
Im "DB_3PUNKT" müssen folgende Parameter eingegeben oder können gelesen werden:
Alle mit "[W]"     gekennzeichneten Parameter sind zwingend einzugeben
Alle mit "[R]"     gekennzeichneten Parameter können nur gelesen werden
Alle mit "[I]/[O]" gekennzeichneten Parameter sind identisch mit In- und Outputs
                   des zugeordneten FCxxx und können nur gelesen werden
################################################################################
Für jeden 3-Punkt-Regler sind das die Parameter:
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
w       [INT]: Sollwert, Führungsgröße           {-32768,...,0,...,+32767}
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
w_SH_MIN[INT]: MINIMUM für "w := [w+SH(-)]"       {-32768,...,0,...,+32767}
               Gilt bereits "w < w_SH_MIN", dann kann "w" durch SH(-) nicht noch
               zu kleineren Werten geschoben werden.
               !! w_SH_MIN ist keine Begrenzung für die Sollwerteingabe "w" !!
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
w_SH_MAX[INT]: MAXIMUM für "w := [w+SH(+)]"       {-32768,...,0,...,+32767}
               Gilt bereits "w > w_SH_MAX", dann kann "w" durch SH(+) nicht noch
               zu größeren Werten geschoben werden.
               !! w_SH_MAX ist keine Begrenzung für die Sollwerteingabe "w" !!
!! Relationen, wie w_SH_MIN >= w_SH_MAX verursachen lediglich, und das ist ab-
   hängig von der Lage von "w", daß nur SH(-) ODER nur SH(+) ODER keine Soll-
   wertschiebung möglich ist.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
h       [INT]: Hysterese                                  {1,2,...,32767}
               Fehleingabe h<=0 -> FC-intern h:=1
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Begriffe:

x        = Istwert  der Regelgröße                   x  {-32768,...,0,...,+32767}
w        = Sollwert der Regelgröße                   w  {-32768,...,0,...,+32767}
```

```
w_eff    = wirkender Sollwert            w_eff=(w+SH) {-32768,...,0,...,+32767}
SH       = Sollwertschiebung                       SH {-32768,...,0,...,+32767}
SH(-)    = negativer Bereich von SH             SH(-) {-32768,...,-1,0}
SH(+)    = positiver Bereich von SH             SH(+) {0,+1,...,+32767}
w_SH_MIN = MINIMUM für w:=[w+SH(-)]          w_SH_MIN {-32768,...,0,...,+32767}
           Gilt bereits "w < w_SH_MIN", dann kann "w" durch SH(-) nicht noch
           zu kleineren Werten geschoben werden.
           !!"w_SH_MIN" ist keine Begrenzung für die Sollwerteingabe "w" !!
w_SH_MAX = MAXIMUM für w:=[w+SH(+)]          w_SH_MAX {-32768,...,0,...,+32767}
           Gilt bereits "w > w_SH_MAX", dann kann "w" durch SH(+) nicht noch
           zu größeren Werten geschoben werden.
           !!"w_SH_MAX" ist keine Begrenzung für die Sollwerteingabe "w" !!
           !!Relationen, wie w_SH_MIN >= w_SH_MAX verursachen lediglich, und
           das ist abhängig von der Lage von "w", daß nur SH(-) ODER nur SH(+)
           ODER keine Sollwertschiebung möglich ist.

h        = Hysterese  (h<=0 -> h:=1)                h {1,2,...,+32767}
yUW      = Stellgröße umgekehrt wirkend           yUW {0,1}
yDW      = Stellgröße direkt    wirkend           yDW {0,1}


Beispiel:

   w1:=w+SH1 UND SH1<0 !
   w2:=w+SH2 UND SH2>0 !


   Y^
    |  yUW         yDW       yUW         yDW            yUW         yDW
   1+ -<-+----+----+->-     -<-+----+----+->-          -<-+----+----+--
    |    ^    .    ^           ^    |    ^                ^    .    ^
    |    .    .    .           |    |    |                .    .    .
    |    .    .    .           |    |    |                .    .    .
    |    .    .    .           |    |    |                .    .    .
   0+----+----+----+-----------+----+----+------------------+----+----+------->
       w1-h   w1  w1+h        w-h   w   w+h              w2-h   w2  w2+h      x
         |<-h-|-h->|           |<-h-|-h->|                |<-h-|-h->|


!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
FC159, SSW : STUFEN- / SCHRITTSCHALTWERK MIT 1-16 STUFEN
################################################################################
Kurzbeschreibung:
Mit dem FC "SSW" kann wahlweise ein STUFEN- oder SCHRITTSCHALTWERK mit maximal
16 STUFEN/SCHRITTEN generiert werden. Es ist möglich, jedem SCHRITT ein belie-
biges BITMUSTER, bestehend aus einem WORT=16BIT, zuzuordnen. Dabei ist auch
wählbar, ob die Weiterschaltung zur/zum nächsten STUFE/SCHRITT FLANKEN- oder
ZEITGESTEUERT erfolgen soll. Für die ZEITSTEUERUNG ist für jede(m) STUFE/SCHRITT
Schritt eine EIN- UND AUSSCHALTVERZÖGERUNGSZEIT parametrierbar.
Alle Parameter des FC "SSW" müssen in einen durch den "UDT_SSW" definierten Da-
tenbausteinbereich eingegeben werden und sie lassen sich vom BuB-System verän-
dern. Vom BuB-System können in der Betriebsart "HAND" die STUFEN/SCHRITTE ge-
gezielt ein oder ausgeschaltet werden. Über den FC-Input "FREEZE" läßt sich der
Schaltzustand einfrieren bzw. ist damit eine Sequenzveriegelung mehrerer "SSW"
möglich.
Über die Sequenzverriegelung können Schaltwerke mit 32,48,usw. STUFEN/SCHRITTEN
aufgebaut werden.
################################################################################

!!Dieser FC verändert nicht die Adressregister AR1/AR2 UND versorgt für FUP- UND
  KOP-Anwendungen die "ENO"-Box, indem das VKE=1 im BIE-BIT gespeichert wird.

in:
FIRST_SCAN  [BOOL]: Im 1.OB1-ZYKLUS nach OB100/101/102=TRUE, sonst FALSE.
                    = TRUE -> Nur Initialisierung Altwertspeicher TIMER
ENABLE      [BOOL]: =0 -> DISABLE(=RESET): Alle y[n]:=0 und Speicher :=0;
                    =1 -> ENABLE:          y[n] EIN/AUS
                    !!"ENABLE" hat Vorrang vor "FREEZE"
FREEZE      [BOOL]: Sequenzverriegelung:
                    =0 -> der Schaltzustand von "y[n]" ist wie beschrieben von
                          den Parametern "ENABLE", "RESET", "HAND", "x_EIN",
                          "x_AUS" UND dem Zustand der TIMER abhängig
```

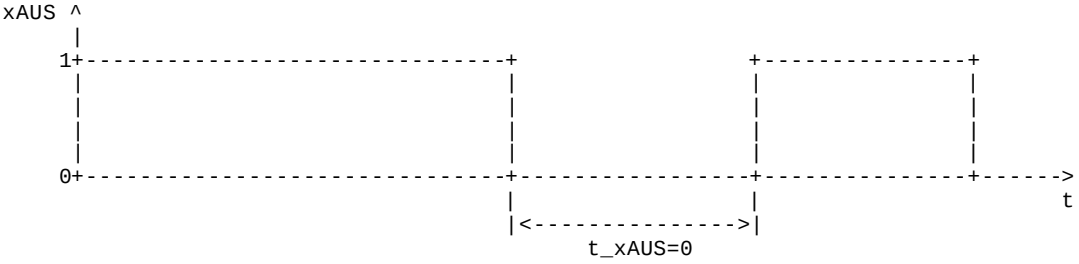
```
                    =1 -> der momentane Schaltzustand von "y[n]" wird eingefro-
                          ren und ist von "x_EIN" UND "x_AUS" unabhängig.
                          "ENABLE", "RESET" UND "HAND" sind wirksam!
                          Die TIMER laufen entsprechend dem Schaltzustand von
                          "x_EIN" UND "x_AUS" ab.
xEIN         [BOOL]: =1 -> Einschaltung der Stufen zeit-/flankengesteuert {0,1}
xAUS         [BOOL]: =1 -> Ausschaltung der Stufen zeit-/flankengesteuert {0,1}
                     Mit "CONT=0/1" wird parametriert, ob die Ein-/Ausschaltung
                     der Stufen zeit- oder flankengesteuert erfolgt.
                    !!Sind "xEIN" UND "xAUS" gemeinsam "=1", werden FC-intern
                      beide Signale als "=0" bewertet.
PRV116BYTE[POINTER]:Pointer auf die Anfangsadresse eines 116 BYTE langen DB-
                    Speicherbereiches, dessen Struktur im "UDT_SWW" festgelegt
                    ist und der für jeden Stufenschalter separat im "DB_SSW"
                    enthalten sein muß. Weiter unten sind die einzelnen Inputs
                    in den "DB_SSW" näher erläutert.
out:
y_n          [WORD]: Stufen "y[n]" des Schrittschaltwerkes         n{1,2,...,16}
                     Die Stufen sind auf das HIGH- und LOW-BYTE des Wortes wie
                     verteilt:
                     H-BYTE:    BIT  0 -> STUFE  1 = y[1]
                                BIT  1 -> STUFE  2 = y[2]
                                BIT  2 -> STUFE  3 = y[3]
                                BIT  3 -> STUFE  4 = y[4]
                                BIT  4 -> STUFE  5 = y[5]
                                BIT  5 -> STUFE  6 = y[6]
                                BIT  6 -> STUFE  7 = y[7]
                                BIT  7 -> STUFE  8 = y[8]

                     L-BYTE:    BIT  0 -> STUFE  9 = y[9]
                                BIT  1 -> STUFE 10 = y[10]
                                BIT  2 -> STUFE 11 = y[11]
                                BIT  3 -> STUFE 12 = y[12]
                                BIT  4 -> STUFE 13 = y[13]
                                BIT  5 -> STUFE 14 = y[14]
                                BIT  6 -> STUFE 15 = y[15]
                                BIT  7 -> STUFE 16 = y[16]

                     !!Im "DB_SSW" sind alle Stufen in einem ARRAY[1..16] mit dem
                       Namen "y" abgelegt. Der Schaltzustand der Stufen kann auch
                       aus den "y[n]" dieses DB-ARRAYS wie oben angegeben gelesen
                       werden.
################################################################################
Im "DB_SSW" müssen folgende Parameter eingegeben oder können gelesen werden:
Alle mit "[W]"     gekennzeichneten Parameter sind zwingend einzugeben
Die  mit "[WO]"    gekennzeichneten Parameter sind optional einzugeben
Alle mit "[R]"     gekennzeichneten Parameter können nur gelesen werden
Alle mit "[I]/[O]" gekennzeichneten Parameter sind identisch mit In- und Outputs
                   des zugeordneten FCxxx und können nur gelesen werden

!!Alle Eingabewerte in den "DB_SSW" werden "FC_SSW"-intern auf zulässige
Werte überprüft und auf die angegebenen Grenzen korrigiert.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
n          [INT]: Anzahl der benutzten Schritte            {0,1,2,...,16}
                  n=0 -> Alle Schritte y[n]:=0
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
MODE      [BOOL]: =0 -> Stufenschaltwerk mit folgendem, fixen Bit-Muster der y[n]
                        STP[1] : B{0000_0001_0000_0000}
                        STP[2] : B{0000_0011_0000_0000}
                        STP[3] : B{0000_0111_0000_0000}
                        STP[4] : B{0000_1111_0000_0000}
                        STP[5] : B{0001_1111_0000_0000}
                        STP[6] : B{0011_1111_0000_0000}
                        STP[7] : B{0111_1111_0000_0000}
                        STP[8] : B{1111_1111_0000_0000}
                        STP[9] : B{1111_1111_0000_0001}
                        STP[10]: B{1111_1111_0000_0011}
                        STP[11]: B{1111_1111_0000_0111}
                        STP[12]: B{1111_1111_0000_1111}
                        STP[13]: B{1111_1111_0001_1111}
                        STP[14]: B{1111_1111_0011_1111}
                        STP[15]: B{1111_1111_0111_1111}
                        STP[16]: B{1111_1111_1111_1111}
```

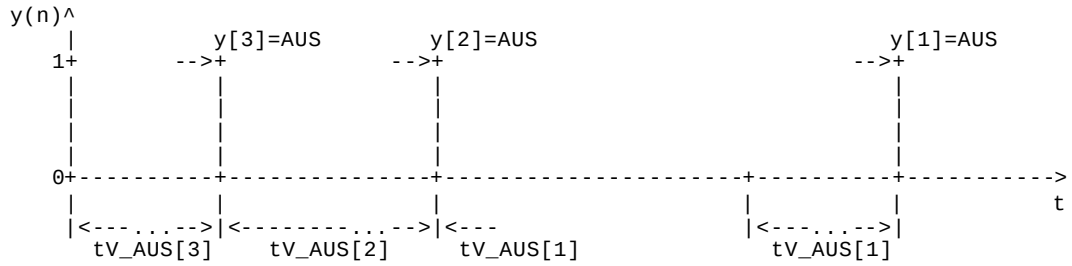
```
                !!Das BIT-Muster der ARRAYS "B" ist bei MODE=0 ohne Bedeutung

                =1 -> Schrittschaltwerk. Das Bit-Muster der y[n] muß für
                      jeden Schritt STP[n] im "DB_SSW" in den ARRAYS "B"
                      festgelegt werden.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
CONT     [BOOL]: =0 -> Die Stufen werden zeitgesteuert    EIN- + AUSgeschaltet.
                      Im "DB_SSW" sind die Zeiten "tv_EIN" und "tv_AUS" einzu-
                      geben.
                =1 -> Die Stufen werden flankengesteuert EIN- + AUSgeschaltet.
                      FC-intern wird die steigende Flanke an den Eingängen
                      "x_EIN" UND "x_AUS" ausgewertet.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
n_H_EIN   [INT]: Anzahl der Schritte HAND-EIN            {0,1,2,...,16}
                Abhhängig von der gewählten Betriebsart lassen sich vom
                BuB-System die Schritte manuell EIN- und AUSSCHALTEN.
                STUFENSCHALTWERK:
                   Beginnend mit Schritt 1 werden bei HAND=1 soviele Schritte
                   eingeschaltet, wie mit "n_H_EIN" angefordert.
                SCHRITTSCHALTWERK:
                   Bei HAND=1 wird der Schritt eingeschaltet, der mit "n_H_EIN"
                   festgelegt ist.

                n_H_EIN = 0 -> Alle Schritte werden ausgeschaltet
                n_H_EIN > n -> n_H_EIN :=n
                !!Beim Umschalten von "AUTO" auf "HAND" gilt "n_H_EIN:=n_A_EIN"
                !!In der BA "AUTO" ist "n_H_EIN:=0"
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
RESET    [BOOL]: =1 -> Alle y[n]:=0;  =0 -> y[n] EIN/AUS, analog Schaltbefehl.
                !!"RESET=1" ODER "ENABLE=0" haben identische Wirkungen und
                  heben sich nicht gegeneinander auf. Der Befehl "RESET" ist
                  für "BuB"-Funktionen vorgesehen.
                !!"RESET" hat Vorrang vor "FREEZE".
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
TAST_AH  [BOOL]: Ändert sich der Schaltzustand "0-1-0", wird FC-intern mit
                einem FLIP-FLOP der Schaltzustand der Stufen von "AUTO" auf
                "HAND" umgeschaltet. Die AUTOFUNKTION des Stufenschalers bleibt
                dabei weiterhin aktiv und beim Rückschalten auf "AUTOMATIK"
                werden dann soviele Stufen eingeschaltet, wie im "DB_SSW" in
                dem Speicher "n_A_EIN" hinterlegt sind.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Für jeden Schritt STP[n]                          n {1,2,...,16}
tV_EIN    [INT]: Einschaltzeitverzögerung        {0,1,...,+32767}*0.1s
tV_AUS    [INT]: Ausschaltzeitverzögerung        {0,1,...,+32767}*0.1s
B  [ARRAY1..16]: BITMUSTER für den Schaltzustand der "y[n]" im Schritt "STP[n]"
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

BEISPIEL 1:
n    = 3 -> 3 Stufen werden benutzt
MODE = 0 -> zeitgesteuert
CONT = 0 -> Stufenschaltwerk
Bitmuster für den Schaltzustand der Stufen:
STP[1] : B{0000_0001_0000_0000} -> y[1]          =1 UND y[2] bis y[16] =0
STP[2] : B{0000_0011_0000_0000} -> y[1] bis y[2] =1 UND y[3] bis y[16] =0
STP[3] : B{0000_0111_0000_0000} -> y[1] bis y[3] =1 UND y[4] bis y[16] =0


EINSCHALTUNG der STUFEN y[n]; n{1,2,3}: xAUS{0} UND xEIN{0,1}
Einschaltung y[n+1] nur, wenn y[n]=1 UND xEIN=1, solange, bis tV_EIN[n+1]=HIGH
```





AUSSCHALTUNG der STUFEN y[n]; n{1,2,3}: xAUS{0,1} UND xEIN{0}
Ausschaltung y[n] nur, wenn y[n+1]=0 UND xAUS=1, solange bis tV_AUS[n]=HIGH





BEISPIEL 2:
n = 5 -> 5 Schritte werden benutzt
MODE = 1 -> flankengesteuert
CONT = 1 -> Schrittschaltwerk
Bitmuster für den Schaltzustand der Stufen:
STP[1] : B{1000_0001_0000_0001} -> y[1] + y[8] + y[9] sind eingeschaltet
STP[2] : B{1001_0000_1100_0000} -> y[5] + y[8] + y[15] + y[16]sind eingeschaltet
STP[3] : B{0000_0011_0000_0000} -> y[1] + y[2] sind
eingeschaltet
STP[4] : B{0000_0000_0001_1100} -> y[11]+ y[12]+ y[13] sind eingeschaltet
STP[5] : B{0100_1000_0000_0000} -> y[4] + y[5] sind eingeschaltet

EINSCHALTUNG der STUFEN y[n]; n{1,2,...,5}: xAUS{0} UND xEIN{0,1}
Einschaltung STP[n+1] nur, wenn STP[n]=1 UND xEIN= 0->1,

```
xEIN ^
     |
    1+    +---+      +----------+   +-+        +---------------+    +-----
     |    |   |      |          |   | |        |               |    |
     |    |   |      |          |   | |        |               |    |
     |    |   |      |          |   | |        |               |    |
     |    |   |      |          |   | |        |               |    |
     |    |   |      |          |   | |        |               |    |
    0+----+---+------+----------+---+-+--------+---------------+----+------->
                                                                            t
y[n] ^
     |    STP[1]     STP[2]         STP[3]      STP[4]               STP[5]
STP=1+    +----      +----          +----       +----                +----
     |    |          |              |           |                    |
y[16]+    |          +-->=EIN       |           |                    |
     |    |          |              |           |                    |
y[15]+    |          +-->=EIN       |           |                    |
     |    |          |              |           |                    |
y[14]+    |          |              |           |                    |
     |    |          |              |           |                    |
y[13]+    |          |              |           +-->=EIN             |
     |    |          |              |           |                    |
y[12]+    |          |              |           +-->=EIN             |
     |    |          |              |           |                    |
y[11]+    |          |              |           +-->=EIN             |
     |    |          |              |           |                    |
y[10]+    |          |              |           |                    |
     |    |          |              |           |                    |
y[9] +    +-->=EIN   |              |           |                    |
     |    |          |              |           |                    |
y[8] +    +-->=EIN   +-->=EIN       |           |                    |
     |    |          |              |           |                    |
y[7] +    |          |              |           |                    |
     |    |          |              |           |                    |
y[6] +    |          |              |           |                    |
     |    |          |              |           |                    |
y[5] +    |          +-->=EIN       |           |                    +-->=EIN
     |    |          |              |           |                    |
y[4] +    |          |              |           |                    +-->=EIN
     |    |          |              |           |                    |
y[3] +    |          |              |           |                    |
     |    |          |              |           |                    |
y[2] +    |          |              +-->=EIN    |                    |
     |    |          |              |           |                    |
y[1] +    +-->=EIN   |              +-->=EIN    |                    |
     |    |          |              |           |                    |
     +----+----------+--------------+-----------+--------------------+------->
     |    1          2              3           4                    5    STP[n]
```

AUSSCHALTUNG der STUFEN y[n]; n{1,2,...,5}: xAUS{0,1} UND xEIN{0}
Ausschaltung STP[n] nur, wenn STP[n+1]=0 UND xAUS= 0->1,
!!DIE AUSSCHALTUNG IST NICHT DARGESTELLT: NACH AUSSCHALTUNG VON STP[n+1] WERDEN
IMMER DIE STUFEN "y[n]" DES SCHRITTES STP[n] EINGESCHALTET.

!!!
FC164, NORMSIGNAL->ANALOG_OUT : WANDLUNG EINES NORMSIGNALES IN ANALOGAUSGANGSIGNAL
###
Kurzbeschreibung:
Mit dem FC "NORMSIGNAL_ANALOG_OUT" wird der Bereich {-1000,...,0,...,+1000}, in dem sich der INTEGERWERT eines NORMSIGNALES bewegt, in das SPS-interne Analog-ausgangssignal {-27648,...,0,...,+27648} umgewandelt. Liegt das Normsignal nicht im o.g. Bereich, so wird das SPS-interne Analogausgangssignal begrenzt.
###
!!Dieser FC verändert nicht die Adressregister AR1/AR2 UND versorgt für FUP- UND KOP-Anwendungen die "ENO"-Box, indem das VKE=1 im BIE-BIT gespeichert wird.

in:
x [INT]: NORMSIGNAL (STELLGRÖßE)   {-1000,...,0,...,+1000}

out:
y [INT]: SPS-internes ANALOGAUSGANGSSIGNAL   {-27684,...,0,...,+27648}

Berechnung:

```
          x*27648
     y = ----------
            1000
```

Begrenzung:

Für x>1000 ODER x<-1000 wird "y" FC-intern auf den angegebenen Bereich begrenzt.

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
FC165, SKAL_LINEAR+LIMIT_INT : SKALIERUNG LINEAR, "y=a*x+b", VOM TYP INTEGER -> INTEGER
####################################################################################
```

Kurzbeschreibung:
Mit dem FC "SKAL_LINEAR+LIMIT_INT" kann ein beliebiger Bereich einer im INTEGER-FORMAT vorliegenden Punktmenge {X} linear auf die Punktmenge {Y} abgebildet wer-den. Die lineare Zuordnung zwischen den Punktmengen wird in Form der Geraden-gleichung "y=a*x+b" hergestellt. Die Punktmenge {Y} wird im INTEGERFORMAT be-rechnet und kann auf einen MIN-/MAX-Wert begrenzt werden.
Der FC "SKAL_LINEAR+LIMIT_INT" ist für die Skalierung beliebiger Analogeingangs-werte in physikalische Größen vorgesehen. Es können auch beliebige Bereiche eines Integerwertes in ein Analogausgangssignal gewandelt werden.

```
##############################################################################
```

!!Dieser FC verändert nicht die Adressregister AR1/AR2 UND versorgt für FUP- UND KOP-Anwendungen die "ENO"-Box, indem das VKE=0 ODER 1 im BIE-BIT gespeichert wird.

```
in:
x    [INT]:    Abszisse Punkt P(x;y)                        {-32768,...,+32767}
yMIN [INT]:    MIN-Limit für y                               {-32768,...,<yMAX}
yMAX [INT]:    MAX-Limit für y                               {yMIN<,...,+32767}
x0   [INT]:    Abszisse Punkt P0(x0;y0)                     {-32768,...,+32767}
y0   [INT]:    Ordinate Punkt P0(x0;y0)                     {-32768,...,+32767}
x1   [INT]:    Abszisse Punkt P1(x1;y1)                     {-32768,...,+32767}
y1   [INT]:    Ordinate Punkt P1(x1;y1)                     {-32768,...,+32767}

out:
y    [INT]:    Ordinate Punkt P(x;y)                            {yMIN,...,yMAX}

!!!Für "y" gilt:
       Wenn  y <= yMIN,                    dann y:=yMIN
       Wenn  y >= yMAX,                    dann y:=yMAX
       Wenn  yMIN >= yMAX,                 dann y unbegrenzt
       Wenn  x0=x1,                        dann y:=0 + ERROR -> BIE-BIT:=0
       Wenn  y0=y1,                        dann y:=y0 + Begrenzung auf yMIN/MAX

!!!Überlauf INTEGER-Bereich -> Korrektur "y" auf MIN/MAX-INTEGER UND BIE-BIT:=0
```

In diesem FC wird eine lineare Zuordnung zwischen den Punktmengen {X} und {Y} in Form der Geradengleichung "y=a*x+b" hergestellt.
FC-intern werden anhand von 2 beliebig vorgebbaren Punkten P0(x0;y0) und P1(x1;y1) die Konstanten "a" und "b" der Gleichung bestimmt.
Dann kann für jeden Wert "x" der zugeordnete Wert "y" berechnet wer-den.

Durch die Eingabe von yMIN-/yMAX kann der berechnete Wert von "y" auf einen unteren und oberen BEREICH begrenzt werden.
Damit werden ÜBER- UND UNTERsteuerungsbereiche der Analog-EIN- und AUSgangskarten ausgeblendet.(Z.B. wird erreicht, daß Analogausgangs-werte immer im Bereich {0,1,...,27648} liegen.)

Die Menge, {X} ODER {Y}, welche die PHYSIKALISCHE GRÖßE darstellt, muß im gesam-ten Wertebereich mit dem Koeffizienten Z=10^m; m{-t,...,-1,0,+1,...,+t} multi-plizierbar sein, damit für das BuB-System eine genormte Dezimalzahldarstellung der "INTEGERWERTE" als "FESTKOMMAZAHL" möglich ist.

"y" wird wie folgt berechnet:

```
      (y1-y0)
  y = --------*(x-x0) + y0
      (x1-x0)
```

```
 y0=y1        ->  y:=y0 für alle x
 x0=x1        ->  y:=0 UND ERROR, das BIE-BIT:=0
 yMIN>=yMAX   ->  Die Begrenzung ist unwirksam und es gilt:
                  yMIN:=-32768; yMAX:=+32767


                             y ^
                               |
                        +32767 +
                               |
                               |                               yMAX
                          yMAX +                                 +--
                               |                              .
                               |                P1(x1;y1)  .
                             y1+------------------------+
                               |             P(x;y)  .  |
                             y +------------------+     |
                               |               .  |     |
                               |  P0(x0;y0) .     |     |
                             y0+----------+       |     |
                               |       .  |       |     |
 ------------------------------+----+-----+-------+-----+--------->
                               | .        x0     x     x1       x
                               +
                            .  |
                         .     |
                      .        |
                   .           |
                .              |
             .                 |
yMIN      .                    |
-------+                  yMIN +
                               |
                        -32768 +
                               |
```

Beispiele:

1.)
Analogeingangskarte:
An der Analogeingangskarte liegt ein Signal {4,...,20}mA an. Dieses wird zu dem SPS-internen Signal {0,...,27648} gewandelt. Damit soll ein Füllstand gemessen werden. Es ist bekannt, daß "4mA" dem Füllstand "0,0m" und "20mA" dem Füllstand 1,700m entsprechen. Daraus ergibt sich folgende Parametrierung:

```
P0(0;0)                          x0=      0 ->     y0= +   0  *[0,001m]
P1(27648;1700)                   x1= +27648 ->     y1= +1700  *[0,001m]
Aus dem o.g. ergibt sich:          yMIN= +   0
                                   yMAX= +1700
```

2.)
Analogausgangskarte:
Das SPS-interne Reglerausgangssignal liegt im Bereich {-1000,...,+1000} vor. Dieses soll in das SPS-interne Analogausgangs-Signal {-27648,...,0,...,+27648} so gewandelt werden, daß der gesamte Bereich des Reglerausgangssignales wirksam wird.
Daraus ergibt sich folgende Parametrierung:

```
P0(-1000; -27648)                   x0=  -1000      und     y0= -27648
P1(+1000; +27648)                   x1=  +1000      und     y1= +27648
Aus dem o.g. ergibt sich:           yMIN= -27648
                                    yMAX= +27648
```

!!
FC166, SKAL_QUADRAT+LIMIT_INT: SKALIERUNG "y=a*x^2 + b*x + c" VOM TYP INTEGER -> INTEGER
##
Kurzbeschreibung:
Mit dem FC "SKAL_QUADRAT+LIMIT_INT" kann ein beliebiger Bereich einer im INTEGERFORMAT vorliegenden Punktmenge {X} nach einer quadratischen Funktion auf die Punktmenge {Y} abgebildet werden. Die Zuordnung zwischen den Punktmengen wird in Form der quadratischen Gleichung "y=a*x^2 + b*x + c" hergestellt. Die Punktmenge {Y} wird im INTEGERFORMAT berechnet und kann auf einen MIN-/MAX-Wert begrenzt werden.
Der FC "SKAL_QUADRAT+LIMIT_INT" wird benötig, wenn eine physikalische Größe "y" aus einem Gebersignal = Analogeingangswert "x" nach der o.g. quadratischen Glei-

chung berechnet werden muß.
Es können auch beliebige Bereiche eines Integerwertes in ein Analogausgangssignal mit quadratischer Kennlinie gewandelt werden. (Ventil- + Klappensteuerung!)
##
!!Dieser FC verändert nicht die Adressregister AR1/AR2 UND versorgt für FUP- UND KOP-Anwendungen die "ENO"-Box, indem das VKE=0 ODER 1 im BIE-BIT gespeichert wird.

in:

x	[INT]:	Abszisse Punkt P(x;y)	{-32768,...,+32767}
yMIN	[INT]:	MIN-Limit für y	{-32768,...,<yMAX}
yMAX	[INT]:	MAX-Limit für y	{yMIN<,...,+32767}
x0	[INT]:	Abszisse Punkt P0(x0;y0)	{-32768,...,+32767}
y0	[INT]:	Ordinate Punkt P0(x0;y0)	{-32768,...,+32767}
x1	[INT]:	Abszisse Punkt P1(x1;y1)	{-32768,...,+32767}
y1	[INT]:	Ordinate Punkt P1(x1;y1)	{-32768,...,+32767}
x2	[INT]:	Abszisse Punkt P2(x2;y2)	{-32768,...,+32767}
y2	[INT]:	Ordinate Punkt P2(x2;y2)	{-32768,...,+32767}

out:

y	[INT]:	Ordinate Punkt P(x;y)	{yMIN,...,yMAX}

!!!Für "y" gilt:

Wenn	y <= yMIN,	dann y:=yMIN
Wenn	y >= yMAX,	dann y:=yMAX
Wenn	yMIN >= yMAX	dann y unbegrenzt
Wenn	x0=x1 ODER x0=x2 ODER x1=x2	dann y:=0 + ERROR -> BIE-BIT:=0
Wenn	y0=y1=y2	dann y:=0 + ERROR -> BIE-BIT:=0

!!!Überlauf INTEGER-Bereich -> Korrektur "y" auf MIN/MAX-INTEGER UND BIE-BIT:=0

In diesem FC wird eine Zuordnung zwischen den Punktmengen {X} und {Y} in Form der quadratischen Gleichung "y=a*x^2 + b*x + c" hergestellt.
FC-intern werden anhand von 3 beliebig vorgebbaren Punkten P0(x0;y0), P1(x1;y1) und P2{x2;y2} die Konstanten "a", "b" und "c" der Gleichung bestimmt. Dann kann für jeden Wert "x" der zugeordnete Wert "y" berechnet werden.

Durch die Eingabe von yMIN-/yMAX wird der berechnete Wert von "y" auf einen unteren und oberen BEREICH begrenzt.

Damit werden ÜBER- UND UNTERsteuerungsbereiche der Analog-EIN- und AUSGÄNGE ausgeblendet.(Z.B. wird erreicht, daß Analogausgangswerte immer im Bereich {0,1,...,27648} liegen.)

Die Menge, {X} ODER {Y}, welche die PHYSIKALISCHE GRÖßE darstellt, muß im gesamten Wertebereich mit dem Koeffizienten Z=10^m; m{-t,...,-1,0,+1,...,+t} multiplizierbar sein, damit für das BuB-System eine genormte Dezimalzahldarstellung der "INTEGERWERTE" als "FESTKOMMAZAHL" möglich ist.

Die Koeffizienten werden wie folgt berechnet:

$$a = \frac{\frac{(y0-y1)}{(x0-x1)} - \frac{(y0-y2)}{(x0-x2)}}{(x1-x2)}$$

$$b = -a*(x0+x1) + \frac{(y0-y1)}{(x0-x1)}$$

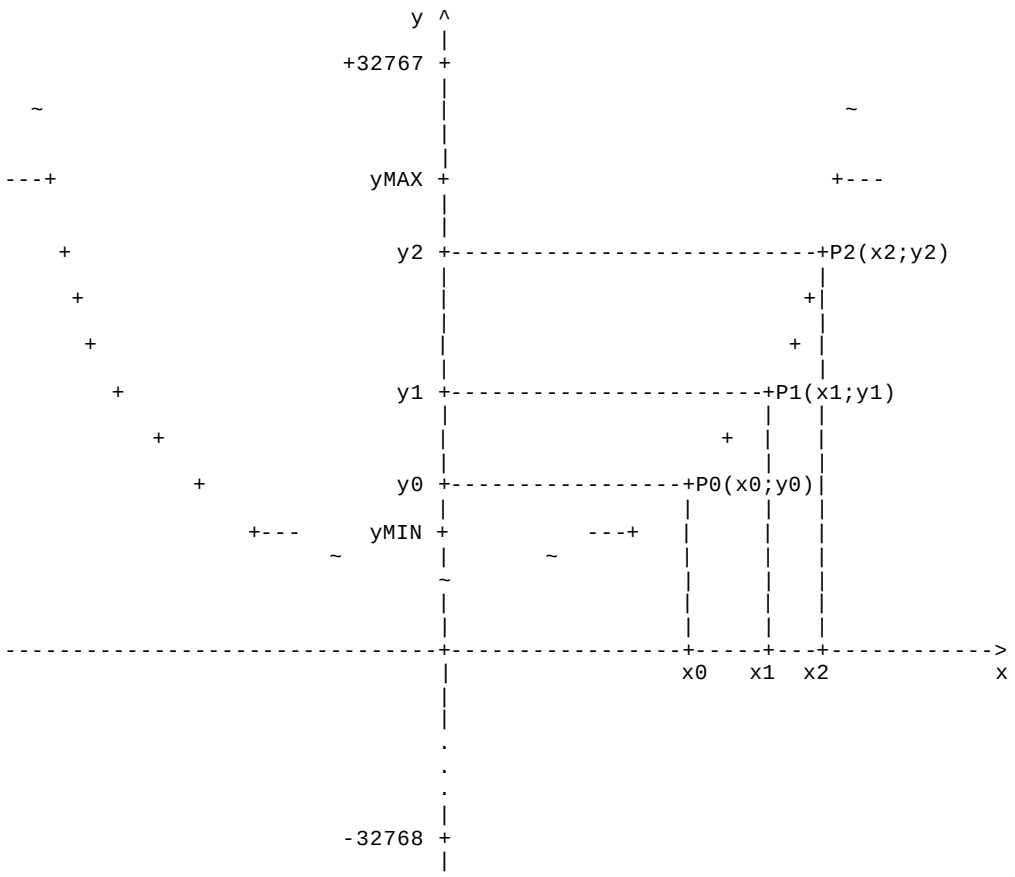
$$c = y0 - a*x0^2 - b*x0$$

"y" wird dann wie folgt berechnet:

$$y = a*x^2 + b*x + c$$

y0=y1=y2	->	y:=0 -> ERROR UND das BIE-BIT:=0 !
x0=x1 ODER x0=x2 ODER x1=x2	->	y:=0 -> ERROR UND das BIE-BIT:=0 !
yMIN>=yMAX	->	Die Begrenzung ist unwirksam und es gilt:

yMIN:=-32768; yMAX:=+32767



Beispiele:
!!!Folgende Abkürzungen werden in den Beispielen verwendet:
   DEE = Einheiten des digitalisierten Analog-EINgangssignales (PEWxxx)
   DEA = Einheiten des digitalisierten Analog-AUSgangssignales (PAWxxx)
   PHE = Einheiten der skalierten physikalischen Größe
1.)
Analogeingangskarte:
An der Analogeingangskarte liegt ein Signal {4,...,20}mA an. Dieses wird zu dem SPS-internen Signal {0,...,27648} gewandelt. Damit wird eine physikalische Größe gemessen die in einem quadratischem Zusammenhang zum Analogeingangssignal steht und von der folgende 3 Punkte bekannt sind:
(Die dem quadratischem Zusammenhang zugrunde liegende Parabel ist zur y-Achse symmetrisch und ihr Scheitelpunkt hat die Koordinaten (0;500).)
Analogeingangssignal = 4[mA]:
P0(0;500)   -> x0=   0*[DEE]; y0= +500*[0,001PHE]
Analogeingangssignal =20[mA]:
P1(+27648;+10000)   -> x1=+27648*[DEE]; y1=+10000*[0,001PHE]
Aus der Symmetrie zur y-Achse folgt:
P2(-27648;-10000)   -> x2=-27648*[DEE]; y2=+10000*[0,001PHE]

"yMIN" und "yMAX" ergeben sich aus dem o.g. wie folgt:
yMIN=   0*[0,001PHE]
yMAX=+10000*[0,001PHE]
(y_MIN und y_MAX, können auch beliebige andere, zulässige Werte sein!)

!!!Der Analogeingangswert muß mit dem FC "MIN_MAX_LIMIT" auf den Wert MIN=0 begrenzt werden, da negative Werte fehlerhafte, positive Werte der physikalischen Größe vortäuschen würden.
2.)
Analogausgangskarte:
Das SPS-interne Reglerausgangssignal liegt im Bereich {0,...,+1000}*[0,1%PHE] vor. Dieses soll in das SPS-interne Analogausgangs-Signal {0,...,+27648}*[DEA]

```
so gewandelt werden, daß
    das Analogausgangssignal einer quadratischen Ventilkennlinie entspricht.
    (Die als Normalparabel durch den Punkt P0(0;0) geht und symmetrisch zur
    y-Achse liegt.)

Daraus ergibt sich folgende Parametrierung:
P0(0;0)                                    x0=      0*[PHE]; y0=      0*[DEA]
P1(+1000; +27648)                          x1=  +1000*[PHE]; y1= +27648*[DEA]
P2(-1000; +27648) = Symmetrie zur y-Achse   x2=  -1000*[PHE]; y2= +27648*[DEA]

Aus dem o.g. ergibt sich:                                   yMIN=      0*[DEA]
                                                            yMAX= +27648*[DEA]

Die nachstehenden, tabellarisch aufgelisteten Stellgrößen werden berechnet:

  Reglerausgang*0,1%PHE |0|100| 200| 300| 400| 500| 600|  700|  800|  900| 1000|

----------------------+-+---+----+----+----+----+----+-----+-----+-----+-----+
  Stellgröße   *0,1%PHE |0| 10|  40|  90| 160| 250| 360|  490|  640|  810| 1000|

----------------------+-+---+----+----+----+----+----+-----+-----+-----+-----+
  Analogausgang*DEA     |0|276|1106|2488|4424|6912|9953|13548|17695|22395|27648|

!!!Die nach der quadratischen Kennlinie verlaufende Stellgröße wurde hier aus
   dem Analogausgangssignal nach folgender Gleichung berechnet:
                        Analogausgang * 1000
           Stellgröße = -------------------  [0,1%PHE]
                              27648
   Sie kann aber auch mittels linearer Skalierung aus dem Analogausgangssignal
   gebildet werden.


!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
FC167, SKAL_WURZEL2+LIMIT_INT : SKALIERUNG "y=a +/- (bx + c)^1/2" VOM TYP INTEGER -> INTEGER
########################################################################################
Kurzbeschreibung:
Mit dem FC "SKAL_WURZEL2+LIMIT_INT" kann ein beliebiger Bereich einer im INTE-
GERFORMAT vorliegenden Punktmenge {X} nach einer Quadratwurzel-Funktion auf die
Punktmenge {Y} abgebildet werden. Die Zuordnung zwischen den Punktmengen wird
mit der allgemeinen Gleichung "y=a +/- (bx + c)^1/2" hergestellt. Die Punkt-
menge {Y} wird im INTEGERFORMAT berechnet und kann auf einen MIN-/MAX-Wert be-
grenzt werden.
Der FC "SKAL_WURZEL2+LIMIT_INT" wird benötig, wenn eine physikalische Größe "y"
aus einem Gebersignal = Analogeingangswert "x" nach der o.g. Wurzel-Funktion
berechnet werden muß.
################################################################################
!!Dieser FC verändert nicht die Adressregister AR1/AR2 UND versorgt für FUP- UND
  KOP-Anwendungen die "ENO"-Box, indem das VKE=0 ODER 1 im BIE-BIT gespeichert
  wird.

in:
x    [INT]:    Abszisse Punkt P(x;y)                        {-32768,...,+32767}
yMIN [INT]:    MIN-Limit für y                               {-32768,...,<yMAX}
yMAX [INT]:    MAX-Limit für y                                {yMIN<,...,+32767}
x0   [INT]:    Abszisse Punkt P0(x0;y0)                     {-32768,...,+32767}
y0   [INT]:    Ordinate Punkt P0(x0;y0)                     {-32768,...,+32767}
x1   [INT]:    Abszisse Punkt P1(x1;y1)                     {-32768,...,+32767}
y1   [INT]:    Ordinate Punkt P1(x1;y1)                     {-32768,...,+32767}
x2   [INT]:    Abszisse Punkt P2(x2;y2)                     {-32768,...,+32767}
y2   [INT]:    Ordinate Punkt P2(x2;y2)                     {-32768,...,+32767}

out:
y_P  [INT]:    Positiv gerichtete Ordinate P(x;y)               {yMIN,...,yMAX}
y_N  [INT]:    Negativ gerichtete Ordinate P(x;y)               {yMIN,...,yMAX}

!!!Für "y_P/N" gilt:
       Wenn  y_P/N <= yMIN,             dann y_P/N:=yMIN
       Wenn  y_P/N >= yMAX,             dann y_P/N:=yMAX
       Wenn  yMIN >= yMAX               dann y_P/N unbegrenzt
       Wenn  x0=x1=x2                   dann y_P/N:=0  +  ERROR -> BIE-BIT:=0
       Wenn  y0=y1 ODER y0=y2 ODER y1=y2 dann y_P/N:=0  +  ERROR -> BIE-BIT:=0
!!!Überlauf INTEGER-Bereich -> Korrektur "y_P/N" auf MIN/MAX-INT UND BIE-BIT:=0
!!!Radikand NEGATIV         -> ERROR:     y_P/N:=0               UND BIE-BIT:=0
```

In diesem FC wird eine Zuordnung zwischen den Punktmengen {X} und {Y} in Form der allgemeinen Funktion der 2.Wurzel (Das ist die Umkehrfunktion der quadratischen Gleichung "y=a*x^2 + b*x + c") hergestellt.

Diese Funktion lautet allgemein "y=a +/- (bx + c)^1/2" und besteht aus dem positiv gerichteten Teil der Ordinate:

"y_P=a+(bx + c)^1/2"

Und dem negativ gerichteten Teil:

"y_N=a-(bx + c)^1/2"

FC-intern werden anhand von 3 beliebig vorgebbaren Punkten P0(x0;y0), P1(x1;y1) und P2{x2;y2} die Konstanten "a", "b" und "c" der Gleichung bestimmt. Dann kann für jeden Wert "x" der zugeordnete Wert "y_P" und "y_N" berechnet werden.

Durch die Eingabe von yMIN-/yMAX wird der berechnete Wert von "y" auf einen unteren und oberen BEREICH begrenzt.

Damit werden ÜBER- UND UNTERsteuerungsbereiche der Analog-EIN- und AUSGÄNGE ausgeblendet.(Z.B. wird erreicht, daß Analogausgangswerte immer im Bereich {0,1,...,27648} liegen.)

Die Menge, {X} ODER {Y}, welche die PHYSIKALISCHE GRÖßE darstellt, muß im gesamten Wertebereich mit dem Koeffizienten Z=10^m; m{-2t,...,-2,0,+2,...,+2t} multiplizierbar sein, damit für das BuB-System eine genormte Dezimalzahldarstellung der "INTEGERWERTE" als "FESTKOMMAZAHL" möglich ist.

Die Koeffizienten werden wie folgt berechnet:

$$a = \frac{(x0-x1)*(y0-y2)*(y0+y2) - (x0-x2)*(y0-y1)*(y0+y1)}{2[(x0-x1)*(y0-y2) - (x0-x2)*(y0-y1)]}$$

$$b = \frac{(y0-a)^2 - (y1-a)^2}{(x0-x1)} \text{ , wenn } x0 \neq x1$$

ODER

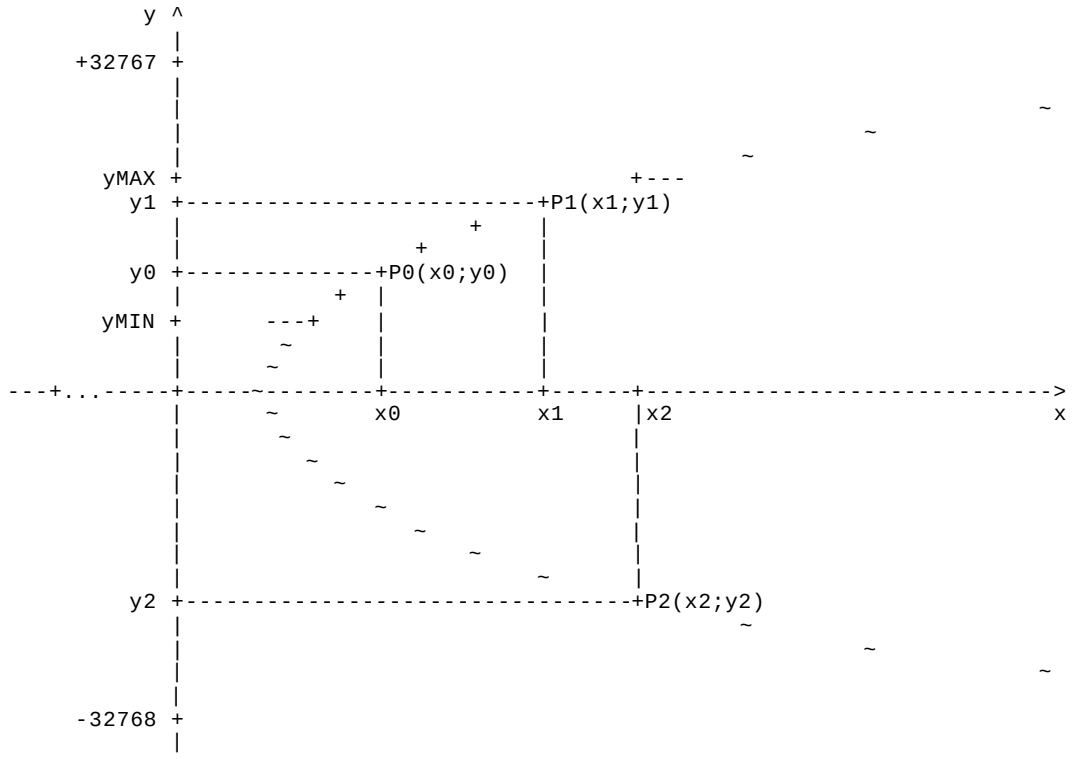
$$b = \frac{(y0-a)^2 - (y2-a)^2}{(x0-x2)} \text{ , wenn } x0 \neq x2$$

$$c = (y0-a)^2 - b*x0$$

"y" wird mit diesen Koeffizienten wie folgt berechnet:

y_P=a+(bx + c)^1/2
y_N=a-(bx + c)^1/2

y0=y1 ODER y0=y2 ODER y1=y2	->	y:=y0 für alle x
x0=x1=x2	->	y:=0 -> ERROR UND das BIE-BIT:=0 !
yMIN>=yMAX	->	Die Begrenzung ist unwirksam und es gilt: yMIN:=-32768; yMAX:=+32767



Beispiel:

Ein Meßwertgeber wandelt eine physikalische Größe in ein Spannungssignal 0-10V nach einer quadratischen Funktion um. SPS-intern entsprechen 0-10V Analogeingangssignal dem Wertebereich 0-27648.

Beispiel:
!!!Folgende Abkürzungen werden in dem Beispiel verwendet:
DEE = Einheiten des digitalisierten Analog-EINgangssignales (PEWxxx)
PHE = Einheiten der skalierten physikalischen Größe

Analogeingangskarte:
An der Analogeingangskarte liegt ein Meßwertgebersignal {0,00V-10,00V} an. Dieses wird zu dem SPS-internen Signal {0,...,27648} gewandelt. Damit wird eine physikalische Größe gemessen, die nach einer Quadratwurzel-Funktion aus dem Ana-
logeingangssignal berechnet werden muß und von der folgendes bekannt ist:
Die physikalische Größe hat für das Meßwertgebersignal {1.00V-10.00V}, das sind SPS-intern {2765,...,27648}*[DEE] (1.00V ist SPS-intern 2765!), den Wertevorrat {0,0 bis 5,000} und soll als Festpunktzahl SPS-intern wie folgt dargestellt werden: "{0,...,5000}*[0,01PHE]".
(Die dem mathematischen Zusammenhang zugrunde liegende Parabel ist zur x-Achse symmetrisch und ihr Scheitelpunkt hat die Koordinaten (2765;0).)

Analogeingangssignal = 0,00[V]:
P0(+ 2765;0) -> x0= 2765*[DEE]; y0=   0*[0,001PHE]
Analogeingangssignal =10,00[V]:
P1(+27648,+5000) -> x1=+27648*[DEE]; y1=+5000*[0,001PHE]
Wegen Symmetrie zur x-Achse folgt:
P2(+27648;-5000) -> x2=+27648*[DEE]; y2=-5000*[0,001PHE]

Aus dem o.g. ergibt sich:
yMIN=   0*[0,001PHE]
yMAX=+5000*[0,001PHE]

(y_MIN und y_MAX, können auch beliebige andere, zulässige Werte sein!)

Damit lassen sich SPS-intern alle Werte der physikalischen Größe aus dem Analogeingangssignal wie folgt linearisiert darstellen:

x * [V]	1,00	2,00	3,00	4,00	5,00	6,00	7,00	8,00	9,00	10,00
PEWxxx * [DEE]	2765	5530	8294	11059	13824	16589	19354	22118	24883	27648
y_P * [PHE]	0	1667	2357	2887	3333	3727	4083	4410	4714	5000

!!!
FC168, SKAL_EXPONENT+LIMIT_INT : SKALIERUNG "y=a^(x+c) + d" VOM TYP INTEGER -> INTEGER
###
Kurzbeschreibung:
Mit dem FC "SKAL_EXPONENT+LIMIT_INT" kann ein beliebiger Bereich einer im INTEGERFORMAT vorliegenden Punktmenge {X} nach einer Exponentialfunktion auf die Punktmenge {Y} abgebildet werden. Die Zuordnung zwischen den Punktmengen wird in Form der Exponentialfunktion "y=a^(x+c)+d" hergestellt. Die Punktmenge {Y} wird im INTEGERFORMAT berechnet und kann auf einen MIN-/MAX-Wert begrenzt werden.
Der FC "SKAL_EXPONENT+LIMIT_INT" wird benötigt, wenn eine physikalische Größe "y" aus einem Gebersignal = Analogeingangswert "x" nach der o.g. Exponentialfunktion berechnet werden muß.
Es können auch beliebige Bereiche eines Integerwertes in ein Analogausgangssignal mit logarithmischer Kennlinie gewandelt werden. (Ventil-/Klappensteuerung!)
###
!!Dieser FC verändert nicht die Adressregister AR1/AR2 UND versorgt für FUP- UND KOP-Anwendungen die "ENO"-Box, indem das VKE=0 ODER 1 im BIE-BIT gespeichert wird.

```
in:
x    [INT]:    Abszisse Punkt P(x;y)                      {-32768,...,+32767}
yMIN [INT]:    MIN-Limit für y                            {-32768,...,<yMAX}
yMAX [INT]:    MAX-Limit für y                            {yMIN<,...,+32767}
x0   [INT]:    Abszisse Punkt P0(x0;y0)                   {-32768,...,+32767}
y0   [INT]:    Ordinate Punkt P0(x0;y0)                   {-32768,...,+32767}
x1   [INT]:    Abszisse Punkt P1(x1;y1)                   {-32768,...,+32767}
y1   [INT]:    Ordinate Punkt P1(x1;y1)                   {-32768,...,+32767}
k   [REAL]:    k=ln(a) > 0, Umrechnungskoeffizient von der Basis "a" zur Basis
               der natürlichen Logarithmen  "e=2,718..."

out:
y    [INT]:    Ordinate Punkt P(x;y)           {yMIN,...,yMAX}

!!!Für "y" gilt:
       Wenn  y <= yMIN,                  dann y:=yMIN
       Wenn  y >= yMAX,                  dann y:=yMAX
       Wenn  yMIN >= yMAX                dann y unbegrenzt
       Wenn  x0=x1 ODER y0=y1            dann y:=0 + ERROR -> BIE-BIT:=0
       Wenn  k <=0                       dann y:=0 + ERROR -> BIE-BIT:=0
!!!Überlauf INTEGER-Bereich -> Korrektur "y" auf MIN/MAX-INTEGER UND BIE-BIT:=0
```

In diesem FC wird eine Zuordnung zwischen den Punktmengen {X} und {Y} in Form der Exponentialfunktion "y=a^(x+c) + d" hergestellt. Die beliebige Basis "a" wird in die Basis der natürlichen Logarithmen "e=2,718..." umgerechnet. Die Exponentialfunktion nimmt dann folgende Form an: "y=e^k*(x+c)+d" mit k=ln(a).

FC-intern werden anhand von 2 beliebig vorgebbaren Punkten P0(x0;y0), P1(x1;y1) die Konstanten "c" und "d" der Gleichung bestimmt. Damit kann für jeden Wert "x" der zugeordnete Wert "y" berechnet werden.

Durch die Eingabe von yMIN-/yMAX wird der berechnete Wert von "y" auf einen unteren und oberen BEREICH begrenzt.

Damit werden ÜBER- UND UNTERsteuerungsbereiche der Analog-EIN- und AUSGÄNGE ausgeblendet.(Z.B. wird erreicht, daß Analogausgangswerte immer im Bereich {0,1,...,27648} liegen.)

Die Menge, {X} ODER {Y}, welche die PHYSIKALISCHE GRÖßE darstellt, muß im gesamten Wertebereich mit dem Koeffizienten Z=10^m; m{-t,...,-1,0,+1,...,+t} multiplizierbar sein, damit für das BuB-System eine genormte Dezimalzahldarstellung der "INTEGERWERTE" als "FESTKOMMAZAHL" möglich ist.

Die Koeffizienten werden wie folgt berechnet:

$$c = \frac{\ln(y0-d)}{k} - x0, \text{ für } y0>d$$

ODER

$$c = \frac{\ln(y1-d)}{k} - x1, \text{ für } y1>d$$

$$d = \frac{y1*e^{k*(x0-x1)} - y0}{e^{k*(x0-x1)} - 1}$$

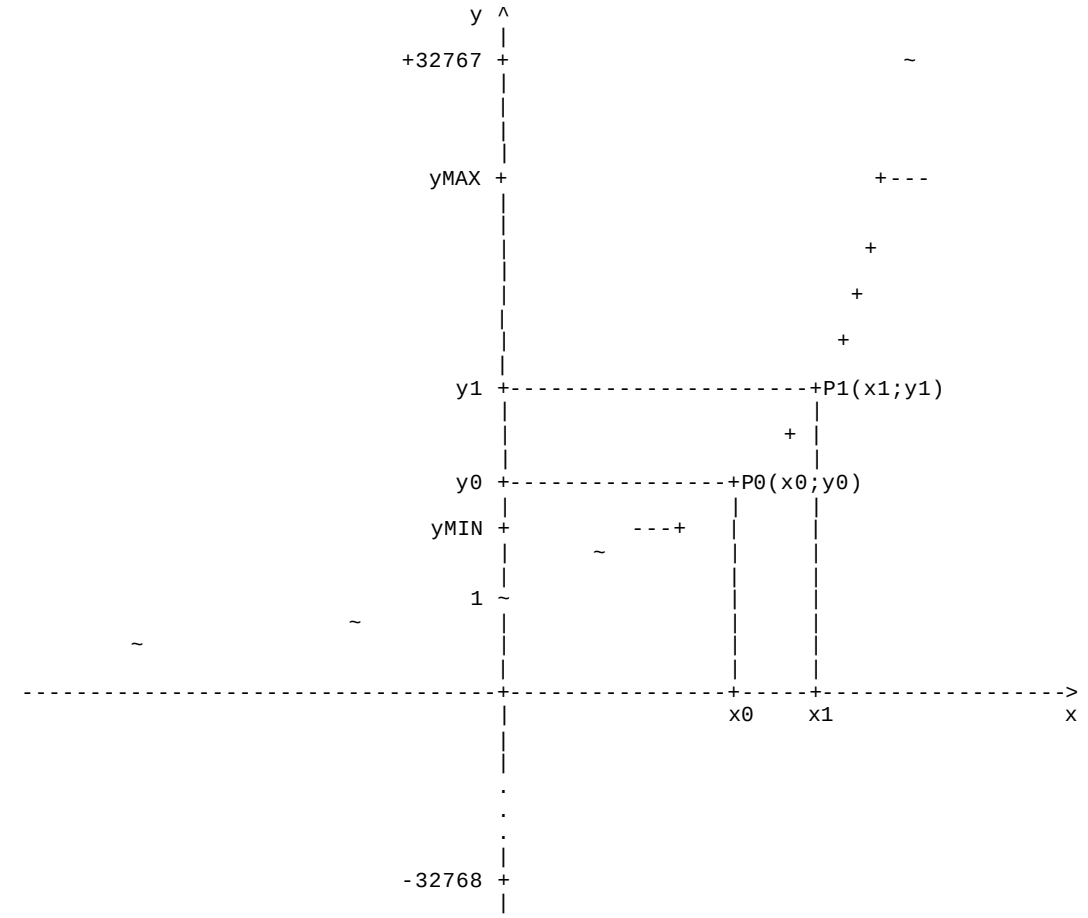
!! "k=ln(a)" gilt nur im Sonderfall. Der Wert für "k" muß in der Regel empirisch ermittelt werden. Beim Ermitteln von "k" sollte die MIN-/MAX-Begrenzung für "y" "-32768/+32767" sein.

für x0=x1 ODER y0=y1 ist keine Berechnung der Koeffizienten möglich -> ERROR, y:=0 UND das BIE-BIT:=0 !

"y" wird dann wie folgt berechnet:

$$y = e^{k*(x+c)} + d$$

x0=x1 ODER y0=y1 -> y:=0 + ERROR UND das BIE-BIT:=0 !
yMIN>=yMAX -> Die Begrenzung ist unwirksam und es gilt:
yMIN:=-32768; yMAX:=+32767



Beispiele:
!!!Folgende Abkürzungen werden in den Beispielen verwendet:
   DEE = Einheiten des digitalisierten Analog-EINgangssignales (PEWxxx)
   DEA = Einheiten des digitalisierten Analog-AUSgangssignales (PAWxxx)
   PHE = Einheiten der skalierten physikalischen Größe
1.)
Analogeingangskarte:
An der Analogeingangskarte liegt ein Signal {4,...,20}mA an. Dieses wird zu dem SPS-internen Signal {0,...,27648} gewandelt. Damit wird eine physikalische Größe gemessen, die nach einer Exponentialfunktion im Zusammenhang zum Analogeingangssignal steht und von der folgends bekannt ist:

Analogeingangssignal = 4 [mA]:
P0(0;10000) -> x0= 0*[DEE]; y0=+10000*[0,001PHE]
Analogeingangssignal =20 [mA]:
P1(+27648;+11000) -> x1=+27648*[DEE]; y1=+11000*[0,001PHE]

Weiterhin ist der Wert "k" empirisch so zu bestimmen, daß sich für das analoge Signal = 10mA -> 10368DEE, eine physikalische Größe von 10250*[0,001PHE] ergibt. (Es soll also der Punkt P(10368;10250) auf der Kurve vorhanden sein! - die Abszisse 10368DEE errechnet sich aus [27648DEE:16mA]*6mA!)

Mit dem vorgegebenen Wert "x=10368" wird für "y=10250" durch gezielte Versuche k =0.0000415 emittelt.

Aus dem o.g. ergibt sich weiterhin:
yMIN=+10000*[0,001PHE]
yMAX=+11000*[0,001PHE]

(y_MIN und y_MAX, können auch beliebige andere, zulässige Werte sein!)

2.)
Analogausgangskarte:
Das SPS-interne Reglerausgangssignal liegt im Bereich {0,...,+1000}*[0,1%PHE] vor. Dieses soll in das SPS-interne Analogausgangs-Signal {0,...,+27648}*[DEA] so gewandelt werden, daß:
  1. Das Analogausgangssignal einer logarithmischen Ventilkennlinie entspricht.
  2. Der Wert "k" muß empirisch so bestimmt werden, daß sich für das Reglerausgangssignal "y=500*0,1%" eine Stellgröße von "x=200*0,1%" ergibt.
     (Es soll also der Punkt P(500;5535) auf der Kurve vorhanden sein! - die Ordinate 5535 errechnet sich aus [27648*200]:1000!)
Daraus ergibt sich folgende Parametrierung:
P0(0;0) x0= 0*[PHE]; y0= 0*[DEA]
P1(+1000; +27648) x1= +1000*[PHE]; y1= +27648*[DEA]

Aus dem o.g. ergibt sich:
yMIN= 0*[DEA]
yMAX= +27648*[DEA]

Mit dem vorgegebenen Wert "x=500" wird durch gezielte Versuche für "k=0.00277" die gewünschte Stellgröße von "y=5535" berechnet.

Die nachstehenden, tabellarisch aufgelisteten Stellgrößen werden berechnet:

Reglerausgang*0,1%PHE	0	100	200	300	400	500	600	700	800	900	1000
Stellgröße *0,1%PHE	0	21	49	87	136	200	285	398	546	742	1000
Analogausgang* DEA	0	590	1368	2395	3749	5535	7892	11001	15102	20511	27648

!!!Die nach der logarithmischen Kennlinie verlaufende Stellgröße wurde hier aus dem Analogausgangssignal nach folgender Gleichung berechnet:

$$\text{Stellgröße} = \frac{\text{Analogausgang} * 1000}{27648}\ [0,1\%PHE]$$

Sie kann aber auch mittels linearer Skalierung aus dem Analogausgangssignal gebildet werden.

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
FC169, SKAL_LOGARITH+LIMIT_INT : SKALIERUNG "y=logm(x+c) + d" VOM TYP INTEGER -> INTEGER
########################################################################################
Kurzbeschreibung:
Mit dem FC "SKAL_LOGARITH+LIMIT_INT" kann ein beliebiger Bereich einer im INTE-
GERFORMAT vorliegenden Punktmenge {X} nach einer logarithmischen Funktion auf
die Punktmenge {Y} abgebildet werden. Die Zuordnung zwischen den Punktmengen
wird nach der Logarithmusfunktion "y=logm(x+c) + d" hergestellt. Die Punktmen-
ge {Y} wird im INTEGERFORMAT berechnet und kann auf einen MIN-/MAX-Wert begrenzt
werden.
Der FC "SKAL_LOGARITH+LIMIT_INT" wird benötigt, wenn eine physikalische Größe
"y" aus einem Gebersignal = Analogeingangswert "x" nach der o.g. logarithmischen
Funktion berechnet werden muß.
################################################################################
!!Dieser FC verändert nicht die Adressregister AR1/AR2 UND versorgt für FUP- UND
  KOP-Anwendungen die "ENO"-Box, indem das VKE=0 ODER 1 im BIE-BIT gespeichert
  wird.

in:
x    [INT]:    Abszisse Punkt P(x;y)                          {-32768,...,+32767}
yMIN [INT]:    MIN-Limit für y                                 {-32768,...,<yMAX}
yMAX [INT]:    MAX-Limit für y                                 {yMIN<,...,+32767}
x0   [INT]:    Abszisse Punkt P0(x0;y0)                       {-32768,...,+32767}
y0   [INT]:    Ordinate Punkt P0(x0;y0)                       {-32768,...,+32767}
x1   [INT]:    Abszisse Punkt P1(x1;y1)                       {-32768,...,+32767}
y1   [INT]:    Ordinate Punkt P1(x1;y1)                       {-32768,...,+32767}
k   [REAL]:    k=1/ln(m) > 0, Umrechnungskoeffizient von der Basis "m" zur Basis
               der natürlichen Logarithmen  "e=2,718..."

out:
y    [INT]:    Ordinate Punkt P(x;y)           {yMIN,...,yMAX}

!!!Für "y" gilt:
       Wenn  y <= yMIN,                 dann y:=yMIN
       Wenn  y >= yMAX,                 dann y:=yMAX
       Wenn  yMIN >= yMAX               dann y unbegrenzt
       Wenn  x0=x1 ODER y0=y1           dann y:=0 + ERROR -> BIE-BIT:=0
       Wenn  k <=0                      dann y:=0 + ERROR -> BIE-BIT:=0
!!!Überlauf INTEGER-Bereich -> Korrektur "y" auf MIN/MAX-INTEGER UND BIE-BIT :=0



In diesem FC wird eine Zuordnung zwischen den Punktmengen {X} und {Y} in Form
der Logarithmusfunktion "y=logm(x+c) + d" hergestellt. Die beliebige Basis "m"
wird in die Basis der natürlichen Logarithmen "e=2,718..." umgerechnet. Die
Logarithmusfunktion nimmt dann folgende Form an: "y=k*ln(x+c)+d" mit k=1/ln(a).

FC-intern werden anhand von 2 beliebig vorgebbaren Punkten P0(x0;y0), P1(x1;y1)
die Konstanten "c" und "d" der Gleichung bestimmt. Damit kann für jeden Wert "x"
der zugeordnete Wert "y" berechnet werden.

Durch die Eingabe von yMIN-/yMAX wird der berechnete Wert von "y" auf einen un-
teren und oberen BEREICH begrenzt.

Damit werden ÜBER- UND UNTERsteuerungsbereiche der Analog-EIN- und AUSGÄNGE
ausgeblendet.(Z.B. wird erreicht, daß Analogausgangswerte immer im Bereich
{0,1,...,27648} liegen.)

Die Menge, {X} ODER {Y}, welche die PHYSIKALISCHE GRÖßE darstellt, muß im gesam-
ten Wertebereich mit dem Koeffizienten Z=10^m; m{-t,...,-1,0,+1,...,+t} multi-
plizierbar sein, damit für das BuB-System eine genormte Dezimalzahldarstellung
der "INTEGERWERTE" als "FESTKOMMAZAHL" möglich ist.

Die Koeffizienten werden wie folgt berechnet:


          x1*e^[1/k*(y0-y1)] - x0
     c =  ----------------------
             1-e^[(1/k*(y0-y1)]


     d = y0 - k*ln(x0+c), für x0>c
```

ODER

d = y1 - k*ln(x1+c), für x1>c

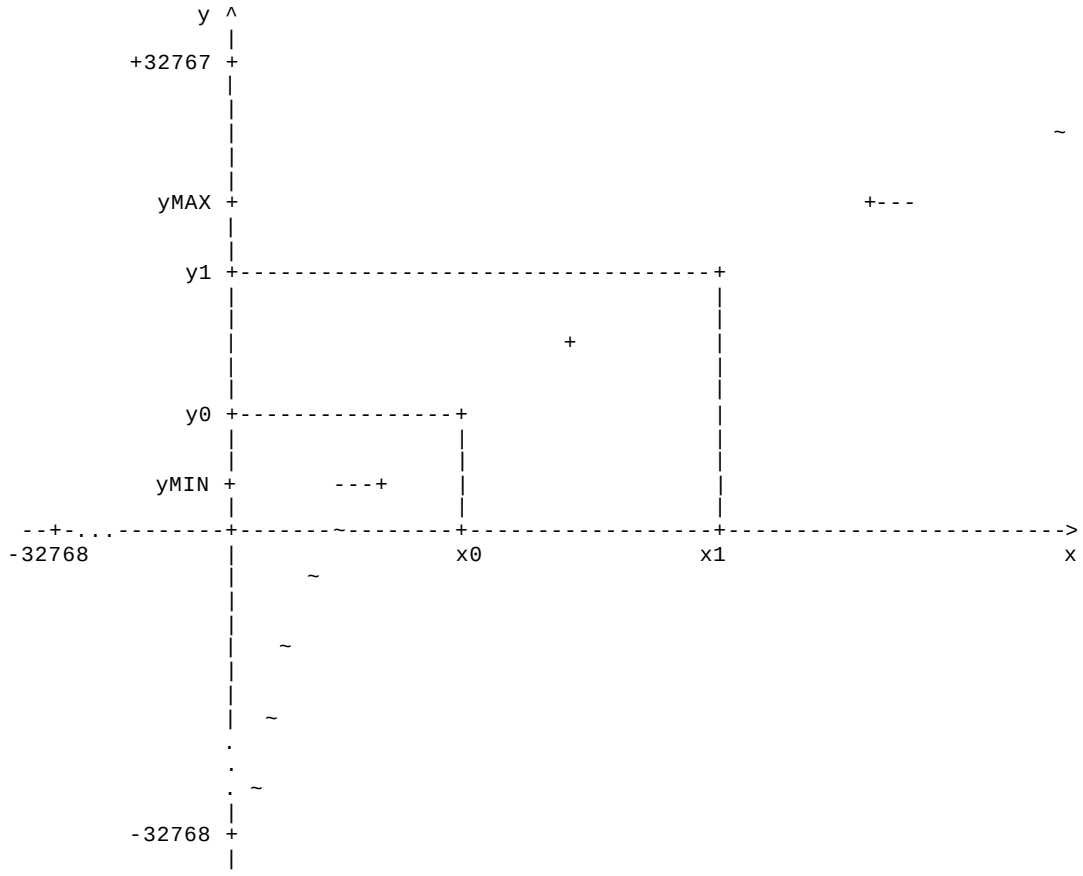
!! "k=1/ln(a)" gilt nur im Sonderfall. Der Wert für "k" muß in der Regel empirisch ermittelt werden. Beim Ermitteln von "k" sollte die MIN-/MAX-Begrenzung für "y" "-32768/+32676" sein.

für x0=x1 ODER y0=y1 ist keine Berechnung der Koeffizienten möglich -> ERROR, y:=0 UND das BIE-BIT:=0 !

"y" wird dann wie folgt berechnet:

y = k*ln(x+c) + d

x0=x1 ODER y0=y1 -> y:=0 + ERROR UND das BIE-BIT:=0 !
yMIN>=yMAX -> Die Begrenzung ist unwirksam und es gilt: yMIN:=-32768; yMAX:=+32767



Beispiel:
!!!Folgende Abkürzungen werden in den Beispielen verwendet:
DEE = Einheiten des digitalisierten Analog-EINgangssignales (PEWxxx)
PHE = Einheiten der skalierten physikalischen Größe

Analogeingangskarte:
An der Analogeingangskarte liegt ein Signal {0,...,10,00}V an. Dieses wird zu dem SPS-internen Signal {0,...,27648} gewandelt. Damit wird eine physikalische Größe gemessen, die nach einer Logarithmusfunktion im Zusammenhang zum Analogeingangssignal steht. Sie hat den Wertevorrat {0.000 bis 5.000} und soll als Festpunktzahl SPS-intern wie folgt dargestellt werden: {0,...,5000}*[0,000PHE].

```
Folgende Daten sind dadurch bekannt:

Analogeingangssignal = 0,00[V]:
P0(0;0)                   -> x0=     0*[DEE];  y0=    0*[0,001PHE]
Analogeingangssignal =10,00[V]:
P1(+27648;+5000)          -> x1=+27648*[DEE];  y1=+5000*[0,001PHE]

Weiterhin ist der Wert "k" empirisch so zu bestimmen, daß sich für das analoge
Signal = 3,114V -> 8610DEE, eine physikalische Größe von 3000*[0,001PHE] ergibt.
(Es soll also der Punkt P(8610;3000) auf der Kurve vorhanden sein! - die Abs-
zisse 8610DEE errechnet sich aus [27648DEE:10V]*3,114V! )

Der Wert "x=8610" wird eingegeben und der Wert "k" solange verändert, bis
"y=3000" ist. Nach wenigen Versuchen findet man für "k=2000.0"

Aus dem o.g. ergibt sich weiterhin:
yMIN=    0*[0,001PHE]
yMAX=+5000*[0,001PHE]

(y_MIN und y_MAX, können auch beliebige andere, zulässige Werte sein!)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
FC172, RAMP_INT : UP/DOWN RAMPE FÜR INTEGER-SIGNAL
################################################################################
Kurzbeschreibung:
Der FC "RAMP_INT" führt permanent den momentanen SOLLWERT eines INTEGER-SIGNALES
mit einer vorgebbaren RAMPE an den berechneten ISTWERT dieses Signales heran.
Z.B. wird ein berechnetes Reglerausgangssignal mit einer Rampe auf den Analog-
ausgang für einen Frequenzumformer eines Antriebes abgebildet.
################################################################################
!!Dieser FC verändert nicht die Adressregister AR1/AR2 UND versorgt für FUP- UND
  KOP-Anwendungen die "ENO"-Box, indem das VKE=1 im BIE-BIT gespeichert wird.

in:
FIRST_SCAN   [BOOL]: Im 1.OB1-ZYKLUS nach OB100/101/102=TRUE, sonst FALSE
                     =1 -> y:=x ohne Rampenfunktion!
SET_y        [BOOL]: =1 -> y:=x ohne Rampenfunktion!
x             [INT]: SOLLWERT der INT-Variablen      {-32768,...,0,...,+32767}
PRV16BYTE [POINTER]: Pointer auf die Anfangsadresse eines 16 BYTE langen DB-
                     Speicherbereiches, dessen Struktur im "UDT_RAMP_INT" fest-
                     gelegt ist und der für jede Rampe im "DB_RAMP_INT" enthal-
                     ten sein muß. Weiter unten sind die einzelnen Inputs in den
                     "DB_RAMP_INT" näher erläutert.
out:
y             [INT]: ISTWERT der INT-Variablen      {-32767,...,0,...,+32767}
                     "y" wird mit einer RAMPE an den Wert "x" angeglichen.

!!Alle Eingabewerte in den "DB_RAMP_INT" werden im "FC_RAMP_INT" auf die ange-
  gebenen Grenzen geprüft. Sind Eingaben fehlerhaft, so werden sie auf die zu-
  lässige Untergrenze=UG bzw. Obergrenze=OG FC-intern korrigiert.

################################################################################
Im "DB_RAMP_INT" müssen folgende Parameter eingegeben oder können gelesen
werden:
Alle mit "[W]"      gekennzeichneten Parameter sind zwingend einzugeben
Alle mit "[R]"      gekennzeichneten Parameter können nur gelesen werden
Alle mit "[I]/[O]" gekennzeichneten Parameter sind identisch mit In- und Outputs
                    des zugeordneten FCxxx und können nur gelesen werden
################################################################################
Für jeden RAMPENGENERATOR sind das die Parameter:
RMP_UP     [INT]: Inkremente pro Sekunde                 {1,2,...,32767}*1/s
                  Gilt als RAMPE für x>y
!!Fehleingabe: RMP_UP   <1 wird FC-intern wie folgt korrigiert: RMP_UP    :=1
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
RMP_DOWN   [INT]: Dekremente pro Sekunde                 {1,2,...,32767}*1/s
                  Gilt als RAMPE für x<y
!!Fehleingabe: RMP_DOWN <1 wird FC-intern wie folgt korrigiert: RMP_DOWN  :=1
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
SET_BUB_y [BOOL]: =1 -> ohne Rampe vom BuB  "y:=x"
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


Beispiel:
```

```
    y ^
      |       x1+------+               x3+--+
      |       .         .              .     .
      |     .            .           .        .
      |   .               .        .           .
    x0+-+                x2+-----+              .
      |                                          .
      |                                           .              x6+-----
      |                                            .               .
      |                                             .              .
      |                                              .             .
     0+-----------------------------------------------.------------.------>
      |                                                .           .      t
      |                                                 .          .
      |                                                x4+---------+
      |                                                            SET_y
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
FC173, DAMP_INT : DÄMPFUNG VON INTEGERSIGNALEN DURCH MITTELWERTBILDUNG
################################################################################
Kurzbeschreibung:
Der FC "DAMP_INT" dämpft ein INTEGER-SIGNAL, z.B. ein Analogeingangssignal,
durch Mittelwertbildung über "n"{1,2,...20} Werte. Für die Mittelwertbildung
wird jedesmal nach Ablauf eines Abtastintervalles der momentane Wert des Sig-
nales einem FIFO gespeichert. Dabei wird vor der Berechnung des Mittelwertes
stets das älteste Signal aus dem FIFO heraus- und das neue Signal in das FIFO
hineingeschoben. Die ZEIT für das ABTASTINTERVALL ist frei parametrierbar.
################################################################################

!!Dieser FC verändert nicht die Adressregister AR1/AR2 UND versorgt für FUP- UND
  KOP-Anwendungen die "ENO"-Box, indem das VKE=1 im BIE-BIT gespeichert wird.

in:
FIRST_SCAN   [BOOL]: Im 1.OB1-ZYKLUS nach OB100/101/102=TRUE, sonst FALSE
x             [INT]: veränderliches INTEGERSIGNAL {-32768,...,0,...,+32767}
PRV54BYTE [POINTER]: Pointer auf die Anfangsadresse eines 54 BYTE langen DB-
                     Speicherbereiches, dessen Struktur im "UDT_DAMP_INT"
                     festgelegt ist und der für jedes zu dämpfende Integer-
                     signal "DB_DAMP_INT" enthalten sein muß.
                     Weiter unten sind die einzelnen Inputs in den "DB_DAMP_INT"
                     näher erläutert.

out:
y             [INT]: MITTELWERT                  {-32767,...,0,...,+32767}


Die INTEGERSIGNALE werden in einem FIFO so gespeichert, daß immer der letzte
Wert heraus- und der NEUWERT des INTEGERSIGNALES als Erstwert in das FIFO hi-
neingeschoben wird. Dann wird aus den "n"-Werten der Mittelwert "y" berechnet.
Ein neues INTEGERSIGNAL wird jedoch immer nur dann im im FIFO abgelegt, wenn
die ABTASTINTERVALLZEIT abgelaufen ist.
Ist "FIRST_SCAN"=TRUE, so werden "n" Elemente des FIFO's mit dem momentan
anliegenden INTEGERSIGNAL "x" aufgefüllt. Für den Mittelwert gilt dann "y=x"!


Der Mittelwert wird wie folgt berechnet
```

$$y = \frac{\mathrm{SUMME}_{k=1}^{n}\, x(k)}{n} ; \qquad n\{1,2,...,20\}$$

```
!!Alle Eingabewerte in den "DB_DAMP_INT" werden im "FC_DAMP_INT" auf die
angegebenen Grenzen geprüft. Sind Eingaben fehlerhaft, so werden sie auf die
zulässige Untergrenze=UG bzw. Obergrenze=OG FC-intern korrigiert.
################################################################################
Im "DB_DAMP_INT" sind Parameter eingegeben oder können gelesen werden:
Alle mit "[W]"     gekennzeichneten Parameter sind zwingend einzugeben
Alle mit "[R]"     gekennzeichneten Parameter können nur gelesen werden
Alle mit "[I]/[O]" gekennzeichneten Parameter sind identisch mit In- und Outputs
                   des zugeordneten FCxxx und können nur gelesen werden
```

```
################################################################################
Für jeden DÄMPFER für INTEGERSIGNALE sind das die Parameter:

n       [INT]: Anzahl der INTEGERSIGNALE                              {1,2,...,20}
               die zur Berechnung des Mittelwertes im FIFO gespeichert werden.
               Für n=1 -> y=x!
               Fehleingabe:   n<1 -> n:=1 ODER n>20 -> n:=20
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
tABTAST [INT]: Abtastintervall-Zeit                    {0,1,...,+32767}*0.001s
               nach deren Ablauf ein neues Signal im FIFO abgelegt wird.
               Fehleingabe:   tABTAST<0 -> tABTAST:=0
!!Wird der FC "DAMP_INT" in einem Weckalarm aufgerufen und soll dessen Zeit-
  intervall als ABTASTZEIT wirksam sein, so ist tABTAST=0 zu wählen.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++



!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
FC174, MIN_MAX_LIMIT : MIN-/MAX-LIMIT
################################################################################
Kurzbeschreibung:
Der FC "MIN_MAX_LIMIT" begrenzt ein INTEGERSIGNAL auf eine frei parametrierbare
UNTER- UND OBERGRENZE.
################################################################################
!!Dieser FC verändert nicht die Adressregister AR1/AR2 UND versorgt für FUP- UND
  KOP-Anwendungen die "ENO"-Box, indem das VKE=1 im BIE-BIT gespeichert wird.

in:
x    [INT]: x                                        {-32768,...,0,...,+32767}
yMIN [INT]: MIN-Limit für y                          {-32768,...,0,...,+32767}
yMAX [INT]: MAX-Limit für y                          {-32768,...,0,...,+32767}

out:
y    [INT]: Wenn  yMIN < x < yMAX,            dann y:=x
            Wenn  x <= yMIN, UND yMIN < yMAX, dann y:=yMIN
            Wenn  x >= yMAX, UND yMIN < yMAX, dann y:=yMAX
            Wenn  x beliebig UND yMIN = yMAX, dann y:=yMIN=yMAX
            Wenn  x beliebig UND yMIN > yMAX, dann ERROR: BIE-BIT:=0 UND  y:=x

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
FC175, MIN_MAX_LIMIT+TOTBAND : MIN-/MAX-LIMIT + TOTBAND
################################################################################
Kurzbeschreibung:
Der FC "MIN_MAX_LIMIT" begrenzt ein INTEGERSIGNAL=EINGANGSSIGNAL auf eine frei
parametrierbare UNTER- UND OBERGRENZE. Durch ein TOTBAND kann das AUSGANGSSIGNAL
stabilisiert werden. Das Ausgangssignal folgt den Änderungen des Eingangssigna-
les nur dann, wenn sich das Eingangssignal um einen größeren Wert ändert, als
der Betrag des TOTBANDES ist ODER das Eingangssignal die Unter-/Obergrenze er-
reicht hat.
################################################################################
!!Dieser FC verändert nicht die Adressregister AR1/AR2 UND versorgt für FUP- UND
  KOP-Anwendungen die "ENO"-Box, indem das VKE=1 im BIE-BIT gespeichert wird.

in:
x            [INT]: x                          {-32768,...,0,...,+32767}
d            [INT]: d = Totband                          {0,...,+32767}
                    !! Ist d<0, dann FC-intern d:=0
yMIN         [INT]: MIN-Limit für y            {-32768,...,0,...,+32767}
yMAX         [INT]: MAX-Limit für y            {-32768,...,0,...,+32767}

PRV2BYTE [POINTER]: Pointer auf Speicherbereich mit der Länge von 2 BYTE
                !!FC-intern werden dieser Speicher benötigt - es dürfen nur
                  Pointer der Form "P#DBrrr.DBXsss.0" ODER "P#Msss.0" sein.
                  Liegt der Pointer im Bereich der temporären Variablen, z.B.
                  P##TEMPsss.0, so muß diese TEMP-Variable vor dem Aufruf
                  des FC aus Speichern versorgt und anschließend in diese rück-
                  gespeichert werden.

out:
y    [INT]: Wenn [x>(y+d) ODER x<(y-d)] UND [yMIN < x < yMAX], dann y:=x
            Wenn  x <= yMIN,            UND     [yMIN < yMAX], dann y:=yMIN
            Wenn  x >= yMAX,            UND     [yMIN < yMAX], dann y:=yMAX
            Wenn  x beliebig UND yMIN > yMAX, dann ERROR, BIE-BIT:=0 UND y:=x
```

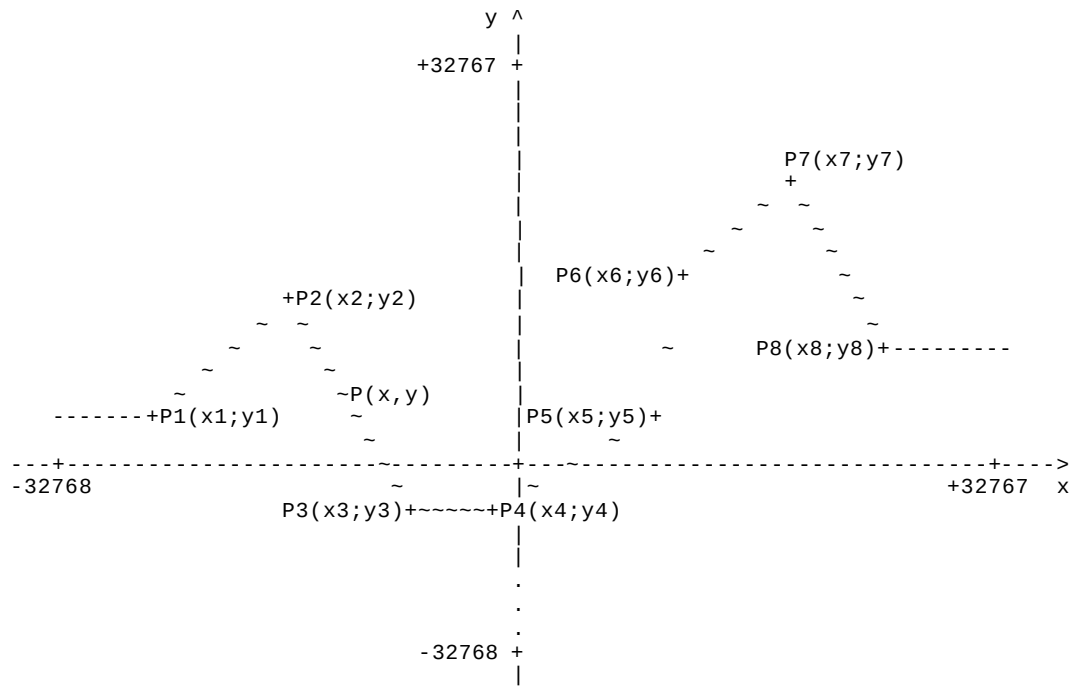
```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
FC176, AVERAGE_2 : AVERAGE 2
################################################################################
Kurzbeschreibung:
Der FC "AVERAGE_2" berechnet aus 2 beliebigen INTEGERVARIABLEN einen MITTELWERT
im INTEGERFORMAT.
################################################################################
!!Dieser FC verändert nicht die Adressregister AR1/AR2 UND versorgt für FUP- UND
  KOP-Anwendungen die "ENO"-Box, indem das VKE=1 im BIE-BIT gespeichert wird.

in:
x1       [INT]:  INT-Variable 1                       {-32768,...,0,...,+32767}
x2       [INT]:  INT-Variable 2                       {-32768,...,0,...,+32767}

out:
y        [INT]:  Mittelwert                           {-32768,...,0,...,+32767}


Berechnungsgrundlagen:

         x1 + x2
     y = -------
            2

!!Ist der REST der DIVISION "REST>=|1|" dann wird "y" auf die nächste, größere
  bzw. kleinere Zahl gerundet.

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
FC177, AVERAGE_2-8 : AVERAGE 2 bis 8
################################################################################
Kurzbeschreibung:
Der FC "AVERAGE_2-8" berechnet aus 2 bis 8 beliebigen INTEGERVARIABLEN einen
MITTELWERT im INTEGERFORMAT. Die Anzahl der in die Berechnung des Mittelwertes
eingehenden Variablen ist im Bereich {2,3,...,8} frei wählbar.
################################################################################
!!Dieser FC verändert nicht die Adressregister AR1/AR2 UND versorgt für FUP- UND
  KOP-Anwendungen die "ENO"-Box, indem das VKE=1 im BIE-BIT gespeichert wird.

in:
n        [INT]:  Anzahl der Variablen                               {2,3,...,8}
                 Durch den Wert von "n" wird die Anzahl und Reihenfolge der
                 Variablen bestimmt, die in die Durchschnittsberechnung ein-
                 gehen.
!!Fehleingabe: n<2 -> n:=2 ODER n>8 -> n:=8   ->   ERROR, das BIE-BIT:=0

x1       [INT]:  INT-Variable 1                       {-32768,...,0,...,+32767}
x2       [INT]:  INT-Variable 2                       {-32768,...,0,...,+32767}
x3       [INT]:  INT-Variable 3                       {-32768,...,0,...,+32767}
x4       [INT]:  INT-Variable 4                       {-32768,...,0,...,+32767}
x5       [INT]:  INT-Variable 5                       {-32768,...,0,...,+32767}
x6       [INT]:  INT-Variable 6                       {-32768,...,0,...,+32767}
x7       [INT]:  INT-Variable 7                       {-32768,...,0,...,+32767}
x8       [INT]:  INT-Variable 8                       {-32768,...,0,...,+32767}

out:
y        [INT]:  Mittelwert                           {-32768,...,0,...,+32767}

Berechnungsgrundlagen:

         x1 + x2 ... + x(n)
     y = ------------------ ;    n{2,3,...,8}
               n

Beispiel: n=5

         x1 + x2 + x3 + x4 + x5
     y = ----------------------
                  5

!!Ist der REST der DIVISION "REST>=|n/2|" dann wird "y" auf die nächste, größere
  bzw. kleinere Zahl gerundet.
```

```
!!Die Inputs x6, x7 UND x8 sind bei n=5 OHNE BEDEUTUNG.

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
FC178, MIN_MAX_SELEKT_2 : MINIMUM UND MAXIMUM AUS 2 VARIABLEN
################################################################################
Kurzbeschreibung:
Der FC "MIN_MAX_SELEKT_2" ermittelt aus 2 beliebigen INTEGERVARIABLEN das
MINIMUM und MAXIMUM .
################################################################################
!!Dieser FC verändert nicht die Adressregister AR1/AR2 UND versorgt für FUP- UND
  KOP-Anwendungen die "ENO"-Box, indem das VKE=1 im BIE-BIT gespeichert wird.

in:
x1       [INT]:  INT-Variable 1                         {-32768,...,0,...,+32767}
x2       [INT]:  INT-Variable 2                         {-32768,...,0,...,+32767}

out:
yMIN     [INT]:  MINIMUM von                            {x1,x2}
yMAX     [INT]:  MAXIMUM von                            {x1,x2}

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
FC179, MIN_MAX_SELEKT_2-8 : MINIMUM UND MAXIMUM AUS 1 bis 8 VARIABLEN
################################################################################
Kurzbeschreibung:
Der FC "MIN_MAX_SELEKT_2-8" ermittelt aus 2 bis 8 beliebigen INTEGERVARIABLEN
das MINIMUM und MAXIMUM.
Die Anzahl der Variablen ist im Bereich {2,3,...,8} frei wählbar.
################################################################################
!!Dieser FC verändert nicht die Adressregister AR1/AR2 UND versorgt für FUP- UND
  KOP-Anwendungen die "ENO"-Box, indem das VKE=0 ODER 1 im BIE-BIT gespeichert
  wird.

in:
n    [INT]:  Anzahl der Variablen                                   {2,3,...,8}
             Durch den Wert von "n" wird die Anzahl der Variablen bestimmt, die
             in die MIN- / MAX-AUSWAHL eingehen. Die in die Auswahl eingehenden
             Variablen sind dadurch festgelegt, daß die 1.Variable immer mit
             "x1" beginnt und die letzte in arithmetischer Folge "x(n)" ist.
!!Fehleingabe: n<2 -> n:=2 ODER n>8 -> n:=8   ->   ERROR, das BIE-BIT:=0


x1   [INT]:  INT-Variable 1                             {-32768,...,0,...,+32767}
x2   [INT]:  INT-Variable 2                             {-32768,...,0,...,+32767}
x3   [INT]:  INT-Variable 3                             {-32768,...,0,...,+32767}
x4   [INT]:  INT-Variable 4                             {-32768,...,0,...,+32767}
x5   [INT]:  INT-Variable 5                             {-32768,...,0,...,+32767}
x6   [INT]:  INT-Variable 6                             {-32768,...,0,...,+32767}
x7   [INT]:  INT-Variable 7                             {-32768,...,0,...,+32767}
x8   [INT]:  INT-Variable 8                             {-32768,...,0,...,+32767}

out:
yMIN [INT]:  MINIMUM von  x(n)                            {x1,x2,...,x(n),...,x8}
yMAX [INT]:  MAXIMUM von  x(n)                            {x1,x2,...,x(n),...,x8}


!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
FC180, SWG_TIME : SOLLWERTGEBER: SOLLWERTFÜHRUNG DURCH ZEITPLAN
################################################################################
Kurzbeschreibung:
Der FC "SWG_TIME" stellt eine Sollwert-Zeitplanführung = "ZPF" dar, deren Para-
meter in einen durch den "UDT_SWG_TIME" definierten Datenbausteinbereich einge-
gegeben werden müssen und die vom BuB-System veränderbar sind. Die "ZPF" be-
steht aus maximal 16 Sollwertstützpunkten = "SSP". Zwischen 2 "SSP" wird der
Sollwert zeitabhängig nach einer Geradengleichung "w = a*t + b" berechnet. Die
"Fahrkurve" des Sollwertes besteht also aus 16 Geradenabschnitten. Beim Umschal-
ten von einem "SSP" auf den nächsten oder am Ende der "ZPF" werden für stati-
stische Auswertungen, separat für jeden "SSP", der Istwert der Regelgröße, die
benötigte Zeit und ein Zeitstempel im Format "DATE_AND_TIME" abgespeichert. Alle
Parameter werden im INTEGER-FORMAT eingegeben. Die Zeiteingaben sind im Fest-
kommazahlen mit der Maßeinheit [0.01h].Damit ist die maximal eingebbare Zeit für
jeden "SSP" = 327.67[h].
```

```
Die Zeitplanführung unterscheidet 2 Betriebsarten:
    1. Der Sollwert wird ohne Rücksicht auf den vorhandenen Istwert geführt.
    2. Ist der geführte Sollwert "w" dem Istwert der Regelgröße "x" um einen
       vorgebbaren Wert, "MAX_DRIFT", vorausgeeilt, dann wird die Sollwertfüh-
       rung solange eingefroren, bis die Abweichung "|x-w|<=MAX_DRIFT" ist.
Mit der Funktion "FREEZE" kann die Sollwertführung über eine Tastfunktion vom
BuB-System angehalten und wieder freigegeben werden. Ebenso sind die Funktionen
"START" und "RESET" über Taster vom BuB-Sytem bedienbar. Der FC "SWG_TIME" hat
auch die Inputs "FREEZE", "START" und "RESET". Damit kann vom Programm die Zeit-
planführung gesteuert werden.
################################################################################


!!Dieser FC verändert nicht die Adressregister AR1/AR2 UND versorgt für FUP- UND
  KOP-Anwendungen die "ENO"-Box, indem das VKE=0 ODER 1 im BIE-BIT gespeichert
  wird.
!!Im FC "SWG_TIME" wird der SFC1 "READ_CLK" aufgerufen.

in:
FIRST_SCAN   [BOOL]: Im 1.OB1-ZYKLUS nach OB100/101/102=TRUE, sonst FALSE
START        [BOOL]: =1 mit FC-internem PLS(+) wird der Zeitplangeber gestartet.
                        Zum Zeitpunkt des Startes werden alle ISTWERTSPEICHER
                        gelöscht.
FREEZE       [BOOL]: =1 der momentane Zustand des Zeitplangebers wird eingefro-
                        ren, die Zeitabläufe werden gestoppt und der im Moment
                        berechnete Sollwert wird permanent ausgegeben.
                        !!FREEZE hat Vorrang vor START
RESET        [BOOL]: =1 der Zeitplangeber wird permanent zurückgesetzt und kann
                        mit "START=1" nur dann neu gestartet werden, wenn der
                        Input "RESET=0" ist. (Das alternative RESET vom BuB-Sy-
                        stem mit dem Parameter "TAST_RESET" setzt den Zeitplan-
                        geber nur mit einem PULS(+) zurück!). Mit "RESET=1" wird
                        "START:=0" UND "n_AKT":=0", die ISTWERTE der vorangegan-
                        genen Zeitplanführung werden jedoch nicht gelöscht.
                        !!RESET  hat Vorrang vor FREEZE
!!Vom BuB-Sytem lassen sich über Eingaben in den "DB_SWG_TIME" auch die Steuer-
  kommandos "START", "FREEZE" und "RESET" realisieren. Sie sind gleichrangig zu
  den gleichnamigen FC-Inputs und schließen sich nicht gegeneinander aus.

PRV348BYTE[POINTER]: Pointer auf die Anfangsadresse eines 348 BYTE langen
                     DB-Speicherbereiches, dessen Struktur im "UDT_SWG_TIME"
                     festgelegt ist und der für jeden Sollwertgeber separat im
                     "DB_SWG_TIME" enthalten sein muß. Weiter unten sind die
                     einzelnen Inputs in den "DB_SWG_TIME" näher erläutert.
x             [INT]: Istwert der Regelgröße              {-32767,...,+32767}
                     !!Die Rückführung der Regelgröße "x" ist für "MOD_ZPF=1"
                       zwingend notwendig. Bei "MOD_ZPF=0" wird "x" nur für
                       statistische Auswertungen erfaßt.
out:
w             [INT]: Aktueller Sollwert nach Zeitplan     {-32767,...,+32767}
################################################################################
Im "DB_SWG_TIME" müssen folgende Parameter eingegeben oder können gelesen
werden:
Alle mit "[W]"     gekennzeichneten Parameter sind zwingend einzugeben
Die  mit "[WO]"    gekennzeichneten Parameter sind optional einzugeben
Alle mit "[R]"     gekennzeichneten Parameter können nur gelesen werden
Alle mit "[I]/[O]" gekennzeichneten Parameter sind identisch mit In- und Outputs
                   des zugeordneten FCxxx und können nur gelesen werden

!!Alle Eingabewerte in den "DB_SWG_TIME" werden "FC_SWG_TIME"-intern auf zuläs-
  sige Werte überprüft und korrigiert.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
n_ENDE       [INT]: Nummer des letzten benötigten SSP          {1,2,...,16}
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
w_START      [INT]: Startsollwert der Zeitplanführung   {-32767,...,+32767}
                    !!Die Zeit t(1) ist auf die Führung von w_START nach w(1)
                      bezogen. Für "MOD_w_START=1" wird die Zeit t(1) für den
                      vorhandenen Istwert "x" wie folgt interpoliert:
                                                |t(1)*[w(1)-x_START]|
                          t(1_INTERPOLIERT)= |--------------------|
                                                |  [w(1)-w_START]   |
                      Ist "[w(1)-w_START]=0", dann kann diese Berechnung nicht
                      ausgeführt werden und es gilt dann für alle "x_START":
```

```
                        t(1_INTERPOLIERT):=t(1)
                      Falls "w(1)=x_START" ist, dann ist t(1)=0 und es wird so-
                      fort auf SSP(2) umgeschaltet.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
MAX_DRIFT    [INT]: MAX|x-w|>0 FÜR MOD_ZPF=1                {0,...,+32767}
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
MOD_w_START [BOOL]: =0 -> START der Zeitplanführung mit "w:=w_START"
                    =1 -> START der Zeitplanführung mit "w:=x"
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
MOD_ZPF     [BOOL]: MODUS DER    ZEITPLANFÜHRUNG=ZPF                  {0,1}
                    =0 -> die Sollwertführung erfolgt unabhängig vom Istwert.
                          Nach Ablauf der Zeit "t" des aktuellen Stützpunktes
                          SSP[n] ist der Zielsollwert "w" erreicht und es wird
                          auf den nächste Stützpunkt umgeschaltet.
                    =1 -> die Sollwertführung wird sofort angehalten, wenn der
                          Betrag der Abweichung zwischen Ist- und Sollwert grös-
                          ser oder gleich "MAX_DRIFT" ist. Ebenso erfolgt die
                          Umschaltung auf den nächsten Stützpunkt nur dann, wenn
                          der Zielsollwert "w" ererreicht ist UND die Abweichung
                          "|x-w|<MAX_DRIFT" ist.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
TAST_START  [BOOL]: BuB = TASTER-START  ZEITPLANFÜHRUNG mit PLS(+)
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
TAST_FREEZE [BOOL]: BuB = TASTER-FREEZE ZEITPLANFÜHRUNG mit FLIP-FLOP
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
TAST_RESET  [BOOL]: BuB = TASTER-RESET  ZEITPLANFÜHRUNG mit PLS(+)
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Für jeden Stützpunkt SSP[n]                                  n {1,2,...,16}
müssen folgend Werte in das "SSP-ARRAY" eingegeben werden:
   t1   [INT]: ZEIT FÜR w(START)-> w(1)   SSP[1]         {0,1,...,32767}*0.01h
   w1   [INT]: SOLLWERT w(1)    AM ENDE   SSP[1]     {-32767,...,+32767}

   t2   [INT]: ZEIT FÜR w(1)    -> w(2)   SSP[2]         {0,1,...,32767}*0.01h
   w2   [INT]: SOLLWERT w(1)    AM ENDE   SSP[2]     {-32767,...,+32767}

    :               :            :        :                :
    :               :            :        :                :

   t16  [INT]: ZEIT FÜR w(15)   -> w(16)  SSP[16]        {0,1,...,32767}*0.01h
   w16  [INT]: SOLLWERT w(16)   AM ENDE   SSP[16]    {-32767,...,+32767}

!!Abhängig von "MOD_ZPF" hat die Eingabe von "t(n)=0" unterschiedliche Wirkung:
  MOD_ZPF=0 -> "w" wird sprunghaft von w(n-1) auf w(n) geändert, n_AKT:=n+1, und
               die ZPF wird mit den Daten des SSP[n+1] fortgesetzt.
  MOD_ZPF=1 -> "w" wird sprunghaft von w(n-1) auf w(n) geändert. Danach wird so-
               lange mit der Umschaltung auf die Daten des SSP[n+1] gewartet,
               bis "|[x-w(n)]|<= MAX_DRIFT" ist.

Berechnungsgrundlagen:
"w(n)" wird im aktuellen Sollwertstützpunkt "SSP[n] wie folgt berechnet:

          [w(n)-w(n-1)]*t_IST
  w(n) = ---------------------- + w(n-1)
                 t(n)

Für n=1 UND MOD_w_START=1, wird t(1), wie unter "w_START" beschrieben, inter-
poliert.


Beispiel:
Sollwertzeitplangeber mit 5 Stützpunkten {SSP1,...,SSP5}
MOD_w_START=0 -> "w" beginnt mit dem Wert "w_START"
MOD_ZPF    =0 -> nach Ablauf von "t(n)" gilt stets "w=w(n)"
```

```
    w  ^
       |                               SSP3(t3;w3) SSP4(t4;w4)
  w3=w4+                                  +-----------+
       |                              .   |           |.
       |                           .      |           | .
       |                        .         |           |  .
       |    SSP1(t1;w1) SSP2(t2;w2)       |           |   .
  w1=w2+       +-----------+              |           |    .
       |     . |           |              |           |     .SSP5(t5;w5)
     w5+   .   |           |              |           |      +---------
       | .     |           |              |           |      |
w_START+       |           |              |           |      |
       +-------+-----------+--------------+-----------+------+------------>
     t=0      t1          t2             t3          t4     t5            t

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
FC181, SWG_INT : SOLLWERTFÜHRUNG DURCH EINEN INTEGERWERT
################################################################################
Kurzbeschreibung:
Der FC "SWG_INT" stellt eine lineare Zuordnung zwischen der Führungsgröße "x"
und dem Sollwert "y" in Form von Geradengleichungen "y=a*x+b" her. FC-intern
wird für jeden Wert "x" ein Punktpaar Pi UND P[i+1] aus den n{1,2,...,8} vorge-
gebenen Stützpunkten bestimmt, für das gilt "xi <= x < x(i+1)".Aus diesen beiden
Punkten werden die Konstanten "a" und "b" der Geradengleichung berechnet und
dann kann für jeden Wert "x" der zugeordnete Wert "y" ermittelt werden, wenn "x"
im Bereich dieser Stützpunkte liegt. Die Anzahl der benutzten Stützpunkte und
deren Koordinaten müssen in einen durch den "UDT_SWG_INT" definierten Datenbau-
steinbereich eingegeben werden.
Damit kann eine Sollwertführung durch eine aus bis aus 8 Geradenabschnitten be-
stehende Kurve (Ein typischer Einsatzfall ist die in Abhängigkeit von der Außen-
temperatur geführte Vorlauftemperatur einer Warmwasserheizung = Heizkurve. Die
Heizkurve wird durch Aufteilung in Geradenabschnitte linearisiert!) realisiert
werden. Der Anfangs- und Endpunkt der Geradenstützpunkte stellt gleichzeitig
einen Grenzwert für den Sollwert dar.
################################################################################
!!Dieser FC verändert nicht die Adressregister AR1/AR2 UND versorgt für FUP- UND
  KOP-Anwendungen die "ENO"-Box, indem das VKE=1 im BIE-BIT gespeichert wird.

in:
x            [INT]: Führungsgröße=Abszisse beliebiger Punkt P{-32768,...,+32767}
PRV38BYTE[POINTER]: Pointer auf die Anfangsadresse eines 38 BYTE langen DB-Spei-
                    cherbereiches, dessen Struktur im "UDT_SWG_INT" festgelegt
                    ist und der für jeden Sollwertgeber separat im "DB_SWG_INT"
                    enthalten sein muß. Weiter unten sind die einzelnen Inputs
                    in den "DB_SWG_INT" näher erläutert.
out:
y            [INT]: Sollwert     =Ordinate beliebiger Punkt P


In diesem FC wird eine lineare Zuordnung zwischen der Führungsgröße "x" und
dem Sollwert "y" in Form der Geradengleichung "y=a*x+b" hergestellt. FC-intern
wird für jeden Wert "x" ein Punktpaar Pi UND P[i+1] aus den "n" Stützpunkten,
die im "DB_SWG_AWIN" vorgegeben sind, bestimmt, für das gilt "xi <= x < x(i+1)".
Aus diesen beiden Punkten werden die Konstanten "a" und "b" der Geradengleichung
berechnet und dann kann für jeden Wert "x" der zugeordnete Wert "y" ermittelt
werden, wenn "x" im o.g. Bereich liegt.


 "y" wird für den Wert "x" im Bereich "xi <= x < x(i+1)" wie folgt berechnet:

      [y(i+1)-yi]
  y = -----------*(x-xi) + yi
      [x(i+1)-xi]

 Für alle  x<x1   -> y:=y1
 Für alle  x>xn   -> y:=yn; n{2,3,...,8}
 Gilt yi = y(i+1) -> y:=yi für alle x im Bereich "xi <= x < x(i+1)"
 Wird die Bedingung "x1<x2<...<xn" verletzt -> ERROR, y:=0 UND BIE-BIT:=0
```



!!Alle Eingabewerte in den "DB_SWG_INT" werden "FC_SWG_INT"-intern auf zuläs-
sige Werte überprüft.
##
Im "DB_SWG_AWIN" müssen folgende Parameter eingegeben oder können gelesen
werden:
Alle mit "[W]"   gekennzeichneten Parameter sind zwingend einzugeben
Die mit "[WO]"   gekennzeichneten Parameter sind optional einzugeben
Alle mit "[R]"   gekennzeichneten Parameter können nur gelesen werden
Alle mit "[I]/[O]" gekennzeichneten Parameter sind identisch mit In- und Outputs
                   des zugeordneten FCxxx und können nur gelesen werden
++
Für jeden Sollwertgeber "SWG_INT" sind das die Parameter:

n   [INT]: Anzahl der benutzten Stützpunkte   {2,...,8}
                 n<2 ODER n>8 -> ERROR: y:=0 UND BIE-BIT:=0
++
Für jeden Stützpunkt Pi   i {1,2,...,8}
Die Koordinaten (xi; yi):

x1   [INT]:   Punkt P1 = Abszisse   {-32768,...,+32767}
y1   [INT]:   Punkt P1 = Ordinate   {-32768,...,+32767}

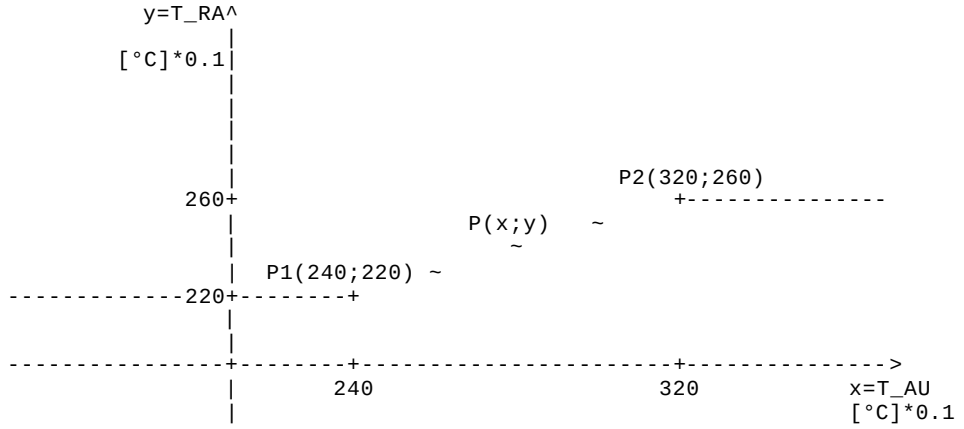
x2   [INT]:   Punkt P2 = Abszisse   {-32768,...,+32767}
y2   [INT]:   Punkt P2 = Ordinate   {-32768,...,+32767}

 :   :   :
 :   :   :
x8   [INT]:   Punkt P8 = Abszisse   {-32768,...,+32767}
y8   [INT]:   Punkt P8 = Ordinate   {-32768,...,+32767}

Um jedem Wert "x" einen eindeutigen Wert "y" zuordnen zu können, muß für alle
verwendeten Stützpunkte folgende Bedingungen eingehalten werden:
x1 < x2 < ...< xi; i{1,2,...,n}
Wird diese Bedingung nicht eingehalten -> ERROR y:=0 UND BIE-BIT:=0
(Die "yi" können beliebige Werte annehmen)
++

BEISPIELE:
1.Anhebung der Raumtemperatur in Abhängigkeit von der Außentemperatur: n=2
  (Bei hohen Außentemperaturen wird in klimatisierten Arbeitsräumen die Raum-

temperatur in abhängigkeit von der Außentemperatur angehoben!)



2.Außentemperaturabhängige Führung der Heißwassertemperatur (Heizkurve): n=6
(Die Heißwassertemperatur für Raumheizungen wird mit sinkender Außentemperatur nach einer Heizkurve angehoben. Diese Heizkurve kann mit ausreichender Genauigkeit durch einen Polygonzug linearisiert werden !)