

Beschreibung aller Unterprogramme der S7-200 Bibliothek PID-EASY

Welche Unterprogramme sind im Softwarepaket PID-Easy für S7-200 enthalten ?

Unterprogrammname	Unterprogrammfunktion
INIT_TIMER_ACK	Permanenztimer, Zeittakte, Quittiertakt + Quittierpuls
DELTA_t	Zeitdifferenz zwischen 2-Aufrufen des lms-TIMERS + Speicherung ISTWERT
TIMER_tD_ON_OFF	TIMER tv_EIN, tvAUS, tmin_AUS, tmin_EIN; tv: {0,...,32767*0.1min + 0,...,32767*0.1s}
MUL_MINUS1_INT	Multiplikation mit *(-1) ohne Überlauf
DATA_SWITCH_INT	Datenumschalter für 2 Integerwerte
AH_SWITCH_INT	AUTO-HAND-Umschaltung für Integerwert (Stellgröße AUTO-HAND!)
MIN_MAX_LIMIT	Begrenzung eines Integerwertes auf eine Ober- + Untergrenze
MIN_MAX_LIMIT_TOTBAND	Begrenzung eines Integerwertes auf eine Ober- + Untergrenze mit Totband
RAMP_INT	Abbildung einer Eingangsgröße mit UP- + DOWN-Rampe auf eine Ausgangsgröße
DAMP_INT	Dämpfung eines Signales über Zeittakt + FIFO + Mittelwertbildung
SWG_INT	Sollwertführung über max 10 Geradenstützpunkte: Pi(xi,yi)
SWG_TIME	Zeitplanführung für Sollwerte mit max 10 Stützpunkten: Pi(wi,ti)
NORMSIGNAL_ANALOGOUTPUT	Skalierung eines Normsignales {-1000,...,0,...,+1000} in Analogausgangssignal
SCALE_LINEAR_LIMIT_INT	Lineare Skalierung eines beliebigen Signales mit Ober- + Untergrenze
AVERAGE_2_INT	Mittelwert von 2-Integerwerten
AVERAGE_2_8_INT	Mittelwert von 2 bis 8 Integerwerten
MIN_MAX_SELECT_2_INT	Auswahl von Minimum + Maximum aus 2 Integerwerten
MIN_MAX_SELECT_2_4_INT	Auswahl von Minimum + Maximum aus 2 bis 8 Integerwerten
MIN_MAX_ALARM	Alarm, wenn ein Integerwert ein Minimum bzw. Maximum erreicht
HY_SWITCH_hw	Hystereseschalter mit : Hysterese + Sollwert, direkt- + inverswirkend
HY_SWITCH_xON_OFF	Hystereseschalter mit : Einschalt- + Ausschaltpunkt, direkt- + inverswirkend
P_CONT	Proportionalregler mit Kaskadeneingang
PID_CONT	PID-Regler mit Kaskadeneingang
ZUSATZSEQUENZ	Zusatzsequenz
THREE_STEP_CONT	Dreipunktregler
AD_TAKTGEBER	Analog- Digitalwandler mit puls- / pausenmodulierten Digitalausgängen
AD_SM	Analog- Digitalwandler mit Stellgliedfunktion 'AUF / ZU'

In der Bibliothek benutzte MERKER + TIMER und ihre SYMBOLE

SYMBOL	ADRESSE	KOMMENTAR
PLS_QUITT	M20.0	BASIC: Quittiertaste als CPU-interner PULS
PLS_500ms	M20.1	BASIC: PULS aller 500ms
PLS1_1s	M20.2	BASIC: PULS1 aller 1s: Immer 500ms vor PLS2
PLS2_1s	M20.3	BASIC: PULS2 aller 1s: Immer 500ms nach PLS1
tD_RUN	M20.4	BASIC: NACH FIRST_SCAN UND TIMER tD_RUN HIGH:=1
dt_TIMER_10ms	MW22	BASIC: TIMER 10ms : Zeitdifferenz zwischen 2 CPU-Zyklen
dt_TIMER_100ms	MW24	BASIC: TIMER 100ms: Zeitdifferenz zwischen 2 CPU-Zyklen
BASIC_PRV0_1_BYTE	MB21	Für UP 'INIT_TIMER_ACK': 1 BYTE PRIVATE SPEICHER
BASIC_PRV1_2_BYTE	MW26	Für UP 'INIT_TIMER_ACK': 2 BYTE SPEICHER = t_ALT, 10ms TIMER
BASIC_PRV2_2_BYTE	MW28	Für UP 'INIT_TIMER_ACK': 2 BYTE SPEICHER = t_ALT, 100ms TIMER
BASIC_PRV3_1_BYTE	MB30	Für UP 'INIT_TIMER_ACK': 1 BYTE SPEICHER=TIMER Q_TAKT
BASIC_PRV4_1_BYTE	MB31	Für UP 'INIT_TIMER_ACK': 1 BYTE SPEICHER=TIMER tD_RUN
PERMANENT_TIMER_1ms	T64	BASIC: Timer 1ms: Im UP 'INIT_TIMER_ACK' permanent neu gestartet
PERMANENT_TIMER_10ms	T100	BASIC: Timer 10ms: Im UP 'INIT_TIMER_ACK' permanent neu gestartet
PERMANENT_TIMER_100ms	T255	BASIC: Timer 100ms: Im UP 'INIT_TIMER_ACK' permanent neu gestartet

UP = UNTERPROGRAMM

SUBROUTINE_BLOCK INIT_TIMER_ACK:SBR37

TITLE=UNTERPROGRAMM 'INIT_TIMER_ACK':

!!ACHTUNG, IN DIESEM UP werden die Sprungmarken 200,201,202,203 benutzt!

!!ES IST IN JEDEM CPU-ZYKLUS IM NETWERK 1, ALS ERSTES PROGRAMM ZU BEARBEITEN.

Kurzbeschreibung:

Im Unterprogramm 'INI_TIMER_ACK' werden die Permanent-Timer aktiviert, aus Hardware- oder OP-Quittierungen ein SPS-Interner Quittierpuls und ein Quittiertakt zur Ansteuerung eines Hardware-Ausganges gebildet. Nach FIRST_SCAN=SM0.1, wird eine Verzögerungszeit gestartet und nach Ablauf dieser Zeit eine Speicherstelle auf TRUE gesetzt.

Mit dieser Wartezeit können Peripheriezugriffsfehler verhindert werden, da nach Netz 'AUS-EIN', die dezentrale Peripherie oder Erweiterungseinheiten zeitverzögert an das Netz gehen.

SUBROUTINE_BLOCK DELTA_t:SBR38

TITLE=UNTERPROGRAMM: 'DELTA_t'

!!ACHTUNG, IN DIESEM UP werden KEINE Sprungmarken benutzt!

!!Dieses Unterprogramm ist nur gemeinsam mit dem UP 'INIT_TIMER_ACK' lauffähig.

Kurzbeschreibung:

Im UP 'DELTA_t' wird für den im UP 'INIT_TIMER_ACK' permanent neu gestarteten lms-TIMER - 'T64' - die ZEITDIFFERENZ $DELTA_t = [T(n+1) - T(n)]$ berechnet, die zwischen 2 Unterprogrammaufrufen 'n' und 'n+1' vergangen ist. 'DELTA_t' ist unterschiedlich, da die lms-TIMER im laufenden Zyklus aktualisiert werden. Gleichzeitig wird vom Unterprogramm der TIMER-Neuwert $T(n+1)$ in einem INT-Speicher, dessen Adresse an das Unterprogramm im 'AC3' übergeben wird, als TIMER-Altwert $T(n)$ abgespeichert.

!!An das Unterprogramm sind folgende Parameter zu übergeben

Mit 'MOVD &VBxxxx,AC3' wird an das Unterprogramm die Adresse eines Speicherbereiches übergeben.

!!Vom Unterprogramm werden folgende Werte zurückgeliefert:

Im AC0 gibt das Unterprogramm die Zeitdifferenz im 'DINT-FORMAT' zurück ('DELTA_t' liegt immer im Bereich von $0 \leq DELTA_t \leq \text{max ZYKLUSZEIT}$.)
und

schreibt in das als Adresse übergebene Speicherwort den aktuellen TIMERSTAND.

(Dieser TIMERSTAND ist dann für den nächsten Aufruf des UP der TIMER-ALTWERT)

!!Das Unterprogramm verändert nur 'AC0'!!

!!Ist SM0.1=1 -> FIRST_SCAN=HIGH: DELTA_t:=0!!

Beispiel: Die Speicherstelle für den TIMERISTWERT ist VW810. Mit

```
LD SM 0.0
MOVD &VW810,AC3
CALL DELTA_t
```

SUBROUTINE_BLOCK TIMER_tD_ON_OFF:SBR39

TITLE=UNTERPROGRAMM: 'TIMER_tD_ON_OFF'

!!ACHTUNG, IN DIESEM UP werden keine Sprungmarken benutzt!

!!Dieses Unterprogramm ist nur gemeinsam mit dem UP 'INIT_TIMER_ACK'

!!und dem UP 'DELTA_t' lauffähig.

Kurzbeschreibung:

Das UP 'IMER_tD_ON_OFF' stellt einen TIMER dar, der als EINSCHALTVERZÖGERUNGSZEIT ODER MINDESTAUSSCHALTZEIT UND AUSCHALTVERZÖGERUNGSZEIT ODER MINDESTEINSCHALTZEIT parametrisiert werden kann. Die Zeiten können in MINUTEN UND SEKUNDEN als Festkommazahl mit dem Faktor '0,1' eingegeben werden.

SUBROUTINE_BLOCK MUL_MINUS1_INT:SBR40

TITLE=UNTERPROGRAMM: 'MUL_MINUS1_INT'

!!ACHTUNG, IN DIESEM UP werden keine Sprungmarken benutzt!

Kurzbeschreibung:

Das UP 'MUL_MINUS1_INT' multipliziert einen INTEGERWERT mit "-1". Dabei wird das Produkt $(-1) \cdot (-32768)$ auf den WERT $=+32767$ gesetzt. Diese Multiplikation wird zur Invertierung von Signalen benötigt.

SUBROUTINE_BLOCK DATA_SWITCH_INT:SBR41

TITLE=UNTERPROGRAMM: 'DATA_SWITCH_INT'

!!ACHTUNG, IN DIESEM UP werden keine Sprungmarken benutzt!

Kurzbeschreibung:

Mit dem UP 'DATA_SWITCH_INT' ist es möglich, abhängig vom Zustand einer binären Speicherstelle einen von 2-Integerwerten auszuwählen.

SUBROUTINE_BLOCK AH_SWITCH_INT:SBR42

TITLE=UNTERPROGRAMM: 'AH_SWITCH_INT'

!!ACHTUNG, IN DIESEM UP werden keine Sprungmarken benutzt!

Kurzbeschreibung:

Mit dem UP 'AH_SWITCH_INT' kann vom HMI-System ein beliebiges INTEGER-Signal von AUTO auf HAND über eine Tastfunktion und einem UP-internen FLIP-FLOP umgeschaltet werden. Im Moment der Umschaltung ist das HAND- mit dem AUTOSIGNAL identisch und kann dann frei gewählt werden. Die Begrenzung des HANDSIGNALES auf einen zulässigen Wertebereich muß im HMI-System erfolgen.

In der Symboltabelle sind für alle AUTO-HAND-Umschalter 8BYTE lange Variablen-Strukturen festgelegt, die, wie nachstehend beschrieben, zum parametrieren der IN- und OUTPUTS des AUTO-HAND-Umschalters benutzt werden müssen. Die Adresse der Anfangsvariablen dieser Strukturen ist über den INPUT 'PRV8YTE' an das Unterprogramm zu übergeben. Alle mit '[W]' gekennzeichneten Variablen können wahlweise im OB1, im Datenbaustein oder das HMI-System parametrieren werden.

!!SIMATIC HMI = Human Machine Interfaces!!

SUBROUTINE_BLOCK MIN_MAX_LIMIT:SBR43

TITLE=UNTERPROGRAMM: 'MIN_MAX_LIMIT'

!!ACHTUNG, IN DIESEM UP werden keine Sprungmarken benutzt!

Kurzbeschreibung:

Das UP 'MIN_MAX_LIMIT' begrenzt ein INTEGERSIGNAL auf eine frei parametrierbare UNTER- UND OBERGRENZE.

SUBROUTINE_BLOCK MIN_MAX_LIMIT_TOTBAND:SBR44

TITLE=UNTERPROGRAMM: 'MIN_MAX_LIMIT_TOTBAND'

!!ACHTUNG, IN DIESEM UP werden keine Sprungmarken benutzt!

Kurzbeschreibung:

Das UP 'MIN_MAX_LIMIT_TOTBAND' 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 Eingangssignales nur dann, wenn sich das Eingangssignal um einen größeren Wert ändert, als der Betrag des TOTBANDES ist ODER das Eingangssignal die Unter-/Obergrenze erreicht hat.

SUBROUTINE_BLOCK RAMP_INT:SBR45

TITLE=UNTERPROGRAMM: 'RAMP_INT'

!!ACHTUNG, IN DIESEM UP wird die Sprungmarke 240 benutzt!

!!Dieses Unterprogramm ist nur gemeinsam mit dem UP 'INIT_TIMER_ACK'

!!und dem UP 'DELTA_t' lauffähig.

Kurzbeschreibung:

Das UP '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 Analogausgang für einen Frequenzumformer eines Antriebes abgebildet. Die Steilheit der Rampen für das Inkrementieren und Dekrementieren ist unterschiedlich parametrierbar.

SUBROUTINE_BLOCK DAMP_INT:SBR46

TITLE=UNTERPROGRAMM: 'DAMP_INT'

!!ACHTUNG, IN DIESEM UP wird keine Sprungmarke benutzt!

!!Dieses Unterprogramm ist nur gemeinsam mit dem UP 'INIT_TIMER_ACK'

!!und dem UP 'DELTA_t' lauffähig.

Kurzbeschreibung:

Das UP '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 Signales 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.

SUBROUTINE_BLOCK SWG_INT:SBR47

TITLE=UNTERPROGRAMM: 'SWG_INT'

!!ACHTUNG, IN DIESEM UP wird die Sprungmarke 241 benutzt!

Kurzbeschreibung:

Das UP 'SWG_INT' stellt eine lineare Zuordnung zwischen der Führungsgröße 'x' und dem Sollwert 'y' in Form von Geradengleichungen 'y=a*x+b' her. UP-intern wird für jeden Wert 'x' ein Punktpaar P_i UND P_{i+1} aus den $n\{1,2,\dots,10\}$ vorgegebenen Stützpunkten bestimmt, für das gilt ' $x_i \leq x \leq x_{i+1}$ '. 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.

In der Symboltabelle sind für alle Sollwertgeber 38BYTE lange Variablen-Strukturen festgelegt, die, wie nachstehend beschrieben, zum parametrieren der IN- und OUTPUTS benutzt werden müssen. Die Adresse der Anfangsvariablen dieser Strukturen ist über den INPUT 'PRV48BYTE' an das Unterprogramm zu übergeben. Alle mit '[W]' gekennzeichneten Variablen können wahlweise im OB1, im Datenbaustein oder das HMI-System parametrieren werden.

Es kann eine Sollwertführung durch eine aus bis aus 10 Geradenabschnitten bestehende Kurve (Ein typischer Einsatzfall ist die in Abhängigkeit von der Außentemperatur 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.

!!SIMATIC HMI = Human Machine Interfaces!!

SUBROUTINE_BLOCK SWG_TIME:SBR48

TITLE=UNTERPROGRAMM: 'SWG_TIME'

!!ACHTUNG, IN DIESEM UP werden die Sprungmarken 242 bis 248 benutzt!

!!Dieses Unterprogramm ist nur gemeinsam mit dem UP 'INIT_TIMER_ACK'

!!und dem UP 'DELTA_t' lauffähig.

Kurzbeschreibung:

Das UP 'SWG_TIME' stellt eine frei parametrierbare Sollwert-Zeitplanführung = 'ZPF' dar. In der Symboltabelle sind für alle 'ZPF' 200BYTE lange Variablen-Strukturen festgelegt, die, wie nachstehend beschrieben, zum parametrieren der IN- und OUTPUTS benutzt werden müssen. Die Adresse der Anfangsvariablen dieser Strukturen ist über den INPUT 'PRV200BYTE' an das Unterprogramm zu übergeben. Alle mit '[W]' gekennzeichneten Variablen können wahlweise im OB1, im Datenbaustein oder das HMI-System parametrieren werden.

Die 'ZPF' besteht aus maximal 10 Sollwertstützpunkten = 'SSP'. Zwischen 2 'SSP' wird der Sollwert zeitabhängig nach einer Geradengleichung $w = a \cdot t + b$ berechnet. Die 'Fahrkurve' des Sollwertes besteht also aus 10 Geradenabschnitten. Beim Umschalten von einem 'SSP' auf den nächsten oder am Ende der 'ZPF' werden für statistische Auswertungen, separat für jeden 'SSP', der Istwert der Regelgröße, die benötigte Zeit und ein Zeitstempel im Format 'TMMhmm' abgespeichert. Alle für die 'ZPF' benötigten Parameter werden im INTEGER-FORMAT eingegeben. Die Zeiteingaben sind im Festkommazahlen mit der Maßeinheit [0.01h] oder wahlweise [1.0s]. Damit ist die maximal eingebare Zeit für jeden 'SSP' = 327.67[h] ODER 32767[s]. 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 Sollwertführung solange eingefroren, bis die Abweichung $|x-w| \leq \text{MAX_DRIFT}$ ist.

Mit der Funktion 'FREEZE' kann die Sollwertführung über eine Tastfunktion vom HMI-System angehalten und wieder freigegeben werden. Ebenso sind die Funktionen 'START' und 'RESET' über Taster vom HMI-System bedienbar. Das UP 'SWG_TIME' hat ebenfalls die Inputs 'FREEZE', 'START' und 'RESET'. Damit kann auch vom Programm die Zeitplanführung gesteuert werden.

Die Zeitplanführung von Sollwerten ist für die Temperaturregelung von Brennöfen und biologischen Prozessen von großer Bedeutung.

!!SIMATIC HMI = Human Machine Interfaces!!

SUBROUTINE_BLOCK NORMSIGNAL_ANALOGOUTPUT:SBR49

TITLE=UNTERPROGRAMM: 'NORMSIGNAL_ANALOGOUTPUT'

!!ACHTUNG, IN DIESEM UP werden keine Sprungmarken benutzt!

Kurzbeschreibung:

Mit dem UP 'NORMSIGNAL_ANALOG_OUT' wird der Bereich $\{-1000, \dots, 0, \dots, +1000\}$, in dem sich der INTEGERWERT eines NORMSIGNALS bewegt, in das SPS-interne Analogausgangssignal $\{-32000, \dots, 0, \dots, +32000\}$ umgewandelt. Liegt das Normsignal nicht im o.g. Bereich, so wird es UP-intern begrenzt. Es besteht die Möglichkeit, eine 0-20mA Analogausgangsbaugruppe für ein 4-20mA Ausgangssignal zu parametrieren.

SUBROUTINE_BLOCK SCALE_LINEAR_LIMIT_INT:SBR50

TITLE=UNTERPROGRAMM: 'SCALE_LINEAR_LIMIT_INT', 'y=a*x+b'

!!ACHTUNG, IN DIESEM UP wird die Sprungmarke 204 benutzt!

Kurzbeschreibung:

Mit dem UP 'SCALE_LINEAR_LIMIT_INT' kann ein beliebiger Bereich einer im INTEGER-FORMAT vorliegenden Punktmenge {X} linear auf die Punktmenge {Y} abgebildet werden. Die lineare Zuordnung zwischen den Punktmenge wird in Form der Geradengleichung $y=a \cdot x+b$ hergestellt. Die Punktmenge {Y} wird im INTEGERFORMAT berechnet und kann auf einen MIN-/MAX-Wert begrenzt werden.

Das UP 'SCALE_LINEAR_LIMIT_INT' ist für die Skalierung beliebiger Analogeingangswerte in physikalische Größen vorgesehen. Es können auch beliebige Bereiche eines Integerwertes in ein Analogausgangssignal gewandelt werden.

SUBROUTINE_BLOCK AVERAGE_2_INT:SBR51

TITLE=UNTERPROGRAMM: 'AVERAGE_2_INT'

!!ACHTUNG, IN DIESEM UP werden keine Sprungmarken benutzt!

Kurzbeschreibung:

Das Unterprogramm 'AVERAGE_2' berechnet aus 2 beliebigen INTEGERVARIABLEN einen MITTELWERT im INTEGERFORMAT.

SUBROUTINE_BLOCK AVERAGE_2_8_INT:SBR52

TITLE=UNTERPROGRAMM: 'AVERAGE_2_8_INT'

!!ACHTUNG, IN DIESEM UP werden keine Sprungmarken benutzt!

Kurzbeschreibung:

Das UP 'AVERAGE_2_8_INT' 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.

SUBROUTINE_BLOCK MIN_MAX_SELECT_2_INT:SBR53

TITLE=UNTERPROGRAMM: 'MIN_MAX_SELECT_2_INT'

!!ACHTUNG, IN DIESEM UP werden keine Sprungmarken benutzt!

Kurzbeschreibung:

Das UP 'MIN_MAX_SELECT_2_INT' ermittelt aus 2 beliebigen INTEGERVARIABLEN das MINIMUM und MAXIMUM .

SUBROUTINE_BLOCK MIN_MAX_SELECT_2_4_INT:SBR54

TITLE=UNTERPROGRAMM: 'MIN_MAX_SELECT_2_4_INT'

!!ACHTUNG, IN DIESEM UP werden keine Sprungmarken benutzt!

Kurzbeschreibung:

Das UP 'MIN_MAX_SELECT_2_4_INT' ermittelt aus 4 beliebigen INTEGERVARIABLEN das MINIMUM und MAXIMUM . Die Anzahl der in die Ermittlung des MINIMUMS und MAXIMUMS eingehenden Variablen ist im Bereich {2,3,...,4} frei wählbar.

SUBROUTINE_BLOCK MIN_MAX_ALARM:SBR55

TITLE=UNTERPROGRAMM: 'MIN_MAX_ALARM'

!!ACHTUNG, IN DIESEM UP werden keine Sprungmarken benutzt!

Kurzbeschreibung:

Das UP 'MIN_MAX_ALARM' aktiviert jeweils einen MIN- ODER MAX-Alarm, wenn eine INTEGER-Variable frei parametrierbare MIN-/MAX-Grenzwerte erreicht bzw. unter- oder überschreitet.

SUBROUTINE_BLOCK HY_SWITCH_hw:SBR56

TITLE=UNTERPROGRAMM: 'HY_SWITCH_hw'

!!ACHTUNG, IN DIESEM UP werden die Sprungmarken 207 + 208 benutzt!

Kurzbeschreibung:

Das UP 'HYST_SWITCH_hw' stellt einen Hystereseschalter dar, dessen Output 'y' sich aus der Regelgröße 'x', dem Sollwert 'w' und der Hysterese 'h' berechnet. Mit den Inputs 'ENABLE' und 'FREEZE' können Sequenzverriegelungen programmiert werden. Die Wirkungsrichtung des Hystereseschalters, direkt- oder umgekehrt wirkend, ist konfigurierbar.

In der Symboltabelle sind für alle Hystereseschalter 9BYTE lange Variablen-Strukturen festgelegt, die, wie nachstehend beschrieben, zum parametrieren der IN- und OUTPUTS des Reglers benutzt werden müssen. Die Adresse der Anfangsvariablen dieser Strukturen ist über den INPUT 'PRV9BYTE' an das Unterprogramm zu übergeben. Alle mit '[W]' gekennzeichneten Variablen können wahlweise im OB1, im Datenbaustein oder das HMI-System parametrieren werden.

!!SIMATIC HMI = Human Machine Interfaces!!

SUBROUTINE_BLOCK HY_SWITCH_xON_OFF:SBR57

TITLE=UNTERPROGRAMM: 'HY_SWITCH_xON_xOFF'

!!ACHTUNG, IN DIESEM UP werden die Sprungmarken 209 + 210 benutzt!

Kurzbeschreibung:

Das UP 'HY_SWITCH_xON_xOFF' stellt einen Hystereseschalter dar, dessen Output 'y' sich aus der Regelgröße 'x', dem Einschaltpunkt 'xON' und dem Ausschalt- punkt 'xOFF' berechnet. Dabei wird die parametrierbare Wirkrichtung des Hyste- reseschalters, direkt- oder umgekehrt wirkend, beachtet.

Mit den Inputs 'ENABLE' und 'FREEZE' können Sequenzverriegelungen programmiert werden.

In der Symboltabelle sind für alle Hystereseschalter 9BYTE lange Variablen- Strukturen festgelegt, die, wie nachstehend beschrieben, zum parametrieren der IN- und OUTPUTS des Hystereseschalters benutzt werden müssen. Die Adresse der Anfangsvariablen dieser Strukturen ist über den INPUT 'PRV9BYTE' an das Unter- programm zu übergeben. Alle mit '[W]' gekennzeichneten Variablen können wahlweise im OB1, im Datenbaustein oder das HMI-System parametrieren werden.

!!SIMATIC HMI = Human Machine Interfaces!!

SUBROUTINE_BLOCK P_CONT:SBR58

TITLE=UNTERPROGRAMM: 'P_CONT'

!!ACHTUNG, IN DIESEM UP werden die Sprungmarken 211 + 212 benutzt!

Kurzbeschreibung:

Das UP 'P_CONT' stellt einen Proportionalregler dar, dessen Parameter im INTEGER- FORMAT eingegeben werden. Der UP-Input 'SH' ermöglicht Sollwertschiebungen für Kaskadenregelungen. Ober- und Untergrenze der Sollwertschiebung sind parame- trierbar.

Der Regler besitzt folgende 3 Norm-Ausgangssignale im Bereich $\{-1000, \dots, +1000\}$:

'y' $\{-1000, \dots, 0, \dots, +1000\}$, als direktwirkendes Signal, dessen Vorzeichen durch das Vorzeichen der Regelabweichung '(x-w)' bestimmt wird.

'yIN' $\{+1000, \dots, 0\}$ als umgekehrtwirkendes Signal für $(x-w) \leq 0 := |-y|$ ODER $:= 0$

'yDI' $\{0, \dots, +1000\}$ als direktwirkendes Signal für $(x-w) > 0 := +y$ ODER $:= 0$

Alle Parameter des Reglers sind Festkommazahlen. Aus diesem Grund wird statt der Proportionalverstärkung 'KP' das Proportionalband 'XP' als Parameter des Reglers verwendet. ($XP=1/KP!$)

In der Symboltabelle sind für alle Proportionalregler 22BYTE lange Variablen- Strukturen festgelegt, die, wie nachstehend beschrieben, zum parametrieren der IN- und OUTPUTS des Reglers benutzt werden müssen. Die Adresse Anfangsvariablen dieser Strukturen ist über den INPUT 'PRV22BYTE' an das Unterprogramm zu über- geben. Alle mit '[W]' gekennzeichneten Variablen können wahlweise im OB1, im Da- tenbaustein oder das HMI-System parametrieren werden.

!!SIMATIC HMI = Human Machine Interfaces!!

SUBROUTINE_BLOCK PID_CONT:SBR59

TITLE=UNTERPROGRAMM: 'PID_CONT'

!!ACHTUNG, IN DIESEM UP werden die Sprungmarken 213 bis 217 benutzt!

!!Dieses Unterprogramm ist nur gemeinsam mit dem UP 'INIT_TIMER_ACK'

!!und dem UP 'DELTA_t' lauffähig.

Kurzbeschreibung:

Das UP 'PID_CONT' stellt einen Proportional-, Integral- und Differentialregler dar, dessen Parameter im INTEGER-FORMAT eingegeben werden. Der UP-Input 'SH' ermöglicht Sollwertschiebungen für Kaskadenregelungen. Ober- und Untergrenze der Sollwertschiebung sind parametrierbar.

Der Regler besitzt folgende 3 Norm-Ausgangssignale im Bereich $\{-1000, \dots, +1000\}$:

'y' $\{-1000, \dots, 0, \dots, +1000\}$, als direktwirkendes Signal, dessen Vorzeichen durch das Vorzeichen der Regelabweichung '(x-w)' bestimmt wird.

'yIN' $\{+1000, \dots, 0\}$ als umgekehrtwirkendes Signal für '(x-w) <= 0 := |-y| ODER := 0'

'yDI' $\{0, \dots, +1000\}$ als direktwirkendes Signal für '(x-w) >= 0 := +y ODER := 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 Stellgröße - 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.

In der Symboltabelle sind für alle Proportionalregler 52BYTE lange Variablenstrukturen festgelegt, die, wie nachstehend beschrieben, zum parametrieren der IN- und OUTPUTS des Reglers benutzt werden müssen. Die Adresse der Anfangsvariablen dieser Strukturen ist über den INPUT 'PRV52BYTE' an das Unterprogramm zu übergeben. Alle mit '[W]' gekennzeichneten Variablen können wahlweise im OB1, im Datenbaustein oder das HMI-System parametriert werden.

!SIMATIC HMI = Human Machine Interfaces!!

SUBROUTINE_BLOCK ZUSATZSEQUENZ:SBR60

TITLE=UNTERPROGRAMM: 'ZUSATZSEQUENZ'

!!ACHTUNG, IN DIESEM UP werden keine Sprungmarken benutzt!!

Kurzbeschreibung:

Mit dem UP 'ZUSATZSEQUENZ' kann ein beliebiger Bereich einer im Integerformat vorliegenden Variablen in ein Normstellsignal $\{0, \dots, +1000\}$ gewandelt werden. Zusatzsequenzen werden häufig in der Klimatechnik benötigt, wo es z.B möglich ist, abhängig vom Außenluftzustand, eine energetisch günstigere Form der Luftaufbereitung zu wählen.

SUBROUTINE_BLOCK THREE_STEP_CONT:SBR61

TITLE=UNTERPROGRAMM: 'TREE_STEP_CONT'

!!ACHTUNG, IN DIESEM UP werden die Sprungmarken 218 + 219 benutzt!

Kurzbeschreibung:

Das UP 'TREE_STEP_CONT' stellt einen klassischen 3-PUNKT-REGLER dar.

In der Symboltabelle sind für alle 3-PUNKT-REGLER 9BYTE lange Variablenstrukturen festgelegt, die, wie nachstehend beschrieben, zum parametrieren der IN- und OUTPUTS des Reglers benutzt werden müssen. Die Adresse der Anfangsvariablen dieser Strukturen ist über den INPUT 'PRV9BYTE' an das Unterprogramm zu übergeben. Alle mit '[W]' gekennzeichneten Variablen können wahlweise im OB1, im Datenbaustein oder das HMI-System parametriert werden.

!SIMATIC HMI = Human Machine Interfaces!!

SUBROUTINE_BLOCK AD_TAKTGEBER:SBR62

TITLE=UNTERPROGRAMM: 'AD_TAKTGEBER'

!!ACHTUNG, IN DIESEM UP werden die Sprungmarken 220 bis 226 benutzt!

!!Dieses Unterprogramm ist nur gemeinsam mit dem UP 'INIT_TIMER_ACK'

!!und dem UP 'DELTA_t' lauffähig.

Kurzbeschreibung:

Mit dem UP 'AD_TAKTGEBER' kann ein analoges Normsignal $\{-1000, \dots, 0, \dots, +1000\}$ in die periodisch taktenden, impulsdauermodulierten digitalen Stellsignale 'yPLUS' und 'yMINUS' gewandelt werden.

Die Periodendauer, eine minimale Ein-/Ausschaltzeit und eine Totzeit bei Schaltungswechsel muß parametrisiert werden.

In der Symboltabelle sind für alle 'AD_TAKTGEBER' 18BYTE lange Variablen-Strukturen festgelegt, die, wie nachstehend beschrieben, zum parametrisieren der IN- und OUTPUTS des 'AD_TAKTGEBER' benutzt werden müssen.

Die Adresse der Anfangsvariablen dieser Strukturen ist über den INPUT 'PRV18BYTE' an das Unterprogramm zu übergeben. Alle mit '[W]' gekennzeichneten Variablen können wahlweise im OB1, im Datenbaustein oder das HMI-System parametrisiert werden.

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, 3-Punkt-Stellantriebe, Dosierpumpen, Thyristorschalter für Elektroheizungen usw. als Stellglieder in Regelkreisen einsetzen.
!!SIMATIC HMI = Human Machine Interfaces!!

SUBROUTINE_BLOCK AD_SM:SBR63

TITLE=UNTERPROGRAMM: 'AD_SM'

!!ACHTUNG, IN DIESEM UP werden die Sprungmarken (227 bis 239)benutzt!

!!Dieses Unterprogramm ist nur gemeinsam mit dem UP 'INIT_TIMER_ACK'

!!und dem UP 'DELTA_t' lauffähig.

Kurzbeschreibung:

Mit dem UP 'AD_SM' können digitale Stellantriebe von einem analogen Normsignal mit dem Wertebereich $\{0, \dots, 1000\}$ angesteuert werden. Aus dem absoluten Wert dieses Signales und dem Stellungs-Istwert $\{0, \dots, 1000\} * 0.1\%$ des Stellantriebes wird die Stellrichtung und die Zeitdauer des Stellbefehles 'AUF'/'ZU' berechnet. Ist die Zeitdauer kleiner als das parametrisierte 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.

In der Symboltabelle sind für alle ANALOG-DIGITAL-WANDLER FÜR STELLMOTOREN 'AUF/ZU' 38BYTE lange Variablenstrukturen festgelegt, die, wie nachstehend beschrieben, zum parametrisieren der IN- und OUTPUTS des UP's 'AD_SM' benutzt werden müssen.

Die Adresse der Anfangsvariablen dieser Strukturen ist über den INPUT 'PRV38BYTE' an das Unterprogramm zu übergeben. Alle mit '[W]' gekennzeichneten Variablen können wahlweise im OB1, im Datenbaustein oder das HMI-System parametrisiert werden.

Vom HMI-System kann der Stellantrieb von der Betriebsart 'AUTO' nach 'HAND' umgeschaltet und im Bereich $\{0, \dots, 1000\} * 0.1\%$ angesteuert werden.

Über den FC-Input 'FREEZE' ist eine Sequenzverriegelung mehrerer Stellantriebe möglich.

Das UP '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 Laufzeitüberwachung!), 'NEUWERT STÖRUNG' und 'HAND' jederzeit ermittelt werden.

!!SIMATIC HMI = Human Machine Interfaces!!