

Inhalt

Inhalt	1
Einleitung	3
Hinweis	3
Vorlage für ein Hintergrundprogramm	3
Allgemeine Aufgabenstellungen	5
H- / P- Programm löschen	5
Kennungsspezifisches Laden von H-Programmen	5
Register initialisieren.....	5
Boolschen (binären) Wert in einem Register toggeln (umschalten)	5
Registerinhalt als Meldung am Display ausgeben.....	6
Parallel - Error oder E-LAN - Error mit LED anzeigen	6
Nach Kennung im E-LAN fragen	6
Impuls erzeugen (Wischersignal)	6
Reaktionen auf Flanken und Pegel eines Signals	7
Auf ein Ereignis verzögert reagieren	7
Abfrage auf bestimmte Uhrzeit	7
Messwert nur nach größerer Änderung weiterverarbeiten	8
bedingte Sperre der Tastatur und bestimmter Eingänge.....	8
Tastendrucke simulieren.....	8
über Anwendermenü einstellbares REG-L -Register.....	8
Betriebsstundenzähler	9
Anzeige der Regelabweichung mit Leuchtdioden.....	9
Meldung, wenn der Schreiber-Speicher voll ist	9
Automatikbetrieb über Binären Eingang aktivieren.....	10
Stromeinfluß deaktivieren wenn Wirkleistung kleiner Null	10
Automatisches Erzeugen einer Gruppenliste	10
Master / Slave Funktion über Eingang aktivieren	11
Regelgröße selbst bestimmen und dem Regler zuführen	11
Wechseln zwischen den Regelgrößen U1 und U2	11
Ansprechen der einzelnen Relais einer BIN-D - Karte	12
blinkende Meldung.....	12
Einstellungen für 2. RelaisKarte im EEPROM.....	12
PowerOn-Reset nach jedem H-Prog-Download	12
Umschalten zwischen Sommer- und Winterzeit	12
Umschalten zwischen Sommer- und Winterzeit ausserhalb der MEZ.....	13
Zeitsynchronisierung über einen Minutenimpuls	13
Sammelmeldung für <U, >U und >I auf LED1 und Relais REL3	13
Spannungsabsenkung über binäre Eingänge.....	13
Toleranzband der Spannung überwachen.....	14
Quittierungsfunktion verändern.....	14
Auf bestimmte Ereignisse reagieren.....	14
Displaybeleuchtung über binären Eingang aktivieren.....	14
Eine oder mehrere Meldungen im Display anzeigen	15
Sollwert als BCD-Code auf BIN-D ausgeben	15
Berechneten Meßwert als Regelgröße verwenden	15
Wandlerdaten parametrieren	16
Überwachen auf Frequenzabweichung	16
E-LAN Error zwischen zwei Geräten auswerten.....	16
Leittechnikanbindung mit REG-P (Formatierungsbeispiele).....	16
Sollwert in % auf ganzzahlige Werte runden	17
Sollwerte wechseln	18
Auswahl von Sollwert 1..4 über Binäre Eingänge	18
einmaliges Umschalten auf Sollwert 2.....	18
4 Sollwerte über Anwendermenü auswählbar	19
Stufenstellung abfragen und ändern.....	20

Rückführung in eine Neutralstufe	20
Stufungen unterdrücken außerhalb der min. / max. Stufenstellung.....	20
Einstellbare Begrenzung der min. / max. Stufenstellung	21
Überwachung des Stufenschalter	21
Zählen der Stufungen pro Zeitraum; mit Reset.....	23
Stufengleichheit zwischen verschiedenen Reglern herstellen.....	24
Ausgabe der Stufenstellung als BCD-Code.....	25
Stufenstellung als BCD-Code auf Relais ausgeben	25

Analoge Ein- / Ausgänge 26

Analogausgang allgemein	26
Analogausgang mit geknickter Kennlinie.....	26
Analogeingang allgemein	27
Analogeingang für Stufenstellung	27
Analogeingang für Stufenstellung ab FW2.00	28
Analogeingang für Stufenstellung mit Anwendermenü.....	28
Analogeingang in Temperatur umrechnen und am Display anzeigen	29

Einleitung

Hinweis

Die in unserem Werkzeugkasten befindlichen Hintergrundprogramme sind Beispiele für die Lösung einfacher kundenspezifischer Aufgaben aus der alltäglichen Praxis. Diese Beispiele bieten zum einen den direkten Einstieg in die freie Programmierbarkeit der Komponenten des Spannungsregelsystems REGSysSM mit der Programmiersprache REG-L(anguage) und zeigen zum anderen typische Aufgabenstellungen, die aus der Umgebung des Transformators und der E-Spule stammen.

Bei Verwendung anliegender Beispiele bzw. bei der kundenseitigen Anpassung muß sichergestellt sein, daß die Standardfunktionen der Geräte nicht beeinträchtigt werden. Aus diesem Grund übernimmt die `a. eberle gmbh` für Schäden, die aus dem falschen Umgang mit dem REG-L Befehlssatz resultieren, keine Haftung.

Allerdings bieten wir Ihnen immer an, ein selbst erstelltes oder auch modifiziertes Hintergrundprogramm zwecks Überprüfung unseren Mitarbeitern der Projektierung zuzusenden (info@a-eberle.de oder norbert.wiedel@a-eberle.de), die Ihnen mit Rat und Tat zur Seite stehen werden.

Viel Erfolg und auch viel Spaß !

Vorlage für ein Hintergrundprogramm

So sollte ein Hintergrundprogramm aufgebaut sein.

```
#####  
# Dateiname:                xxx.xxx.xxx.rgl                Datum: xx.xx.xx  
# Version:                   xx  
# für Gerätetyp: REG-D/REG-DA/PAN-D  
# Kunde/Proj.:  
# erstellt:                  xx.xx.xx                A. Eberle Gmbh & Co. KG  
# geändert:  
#####  
# H-Programm stoppen  
hbreak+++  
# Prompt auf eigene Kennung setzen  
rem,aa::  
# alle H- und P- Programmzeilen löschen  
H 0..31=""  
P 0..31=""  
Q 0..31=""  
# Register löschen  
a *=0  
b *=0  
# Anwendermenü löschen  
menuappn *="" , menuapp *=-1  
  
H 0='# xxx.xxx.xxx.rgl (BV xxx)'  
  
# Sommer-/Winterzeitumschaltung  
H 31='SOWI, IF, ZEIT-, +, ZEIT=.'  
  
# H-Programm starten  
hbreak-
```

Parameter - Beschreibung #####

#

Allgemeine Aufgabenstellungen

H- / P- Programm löschen

Aufgabe:

alle H- und P- Programmzeilen sollen gelöscht werden (z.B. vor einer Neu-Programmierung)

Lösung:

```
H 0..31=' '  
P 0..31=' '
```

Kennungspezifisches Laden von H-Programmen

Aufgabe:

In mehrere Regler (z.B. Kennung "A", "B" und "C") sollen ähnliche H-Programme geladen werden, die alle in einer Download-Datei geschrieben wurden.

Lösung:

Prompt auf eigene Kennung setzen, damit alle Befehl auf dem eigenen Gerät ausgeführt werden
Vorsicht, lt. dem Programm "RegUpdate" darf der 1. Befehl keine Kennung enthalten

rem,aa::

z.B. die Zeile H 2 soll sich in den Reglern unterscheiden

Kenn, 1,==,if, H 2='meld "Ich bin Regler A"'

Kenn,11,==,if, H 2='meld "Ich bin Regler B"'

Kenn,21,==,if, H 2='meld "Ich bin Regler C"'

Register initialisieren

Aufgabe:

Alle Register sollen gelöscht werden und dann dem Register b12 ein Wert zugewiesen werden

Lösung:

alle Register löschen

H 1='a *=0, b *=0'

dem Register b12 den Wert 999 zuweisen

H 2='b12=999'

Boolschen (binären) Wert in einem Register toggeln (umschalten)

Aufgabe:

wenn a 1=1 => a 1=0

wenn a 1=0 => a 1=1

Lösung:

H 1='a 1,^,a 1='

Registerinhalt als Meldung am Display ausgeben

Aufgabe:

der aktuelle Wert des Registers b 0 soll am Display ausgegeben werden

Lösung:

```
H 30='b 0,meld "Wert B0 = %!"
```

Parallel - Error oder E-LAN - Error mit LED anzeigen

Aufgabe:

Sobald ein E-LAN-Error oder ein Parallel-Error festgestellt wird soll eine LED leuchten.

Lösung:

```
# Bei E-LAN-Error oder Parallel-Error leuchtet LED 1  
H 2='RegRelFW- 17,RegRelFW- 30,|,if,RegLED 1=1,else,RegLED 1=0'
```

Nach Kennung im E-LAN fragen

Aufgabe:

a1=1, wenn das Gerät mit der Kennung B1 im E-LAN vorhanden ist

Lösung:

```
H 1='B1:indir*,a 1=.'
```

Impuls erzeugen (Wischersignal)

Aufgabe:

Beim Auftreten eines BCD-Code-Fehlers soll ein 3 s Impuls am Relais 4 erzeugt werden

Lösung:

```
H 1='RegStufe-,99,==,iff,a 1=1'  
H 2='htd,a 1,if,a++ 63=.,a 63,3,>,if,a 1=0,endif,else,a 63=0'  
H 3='a 1,RegRel 4=.'  
oder  
H 1='RegStufe-,99,==,iff,zeit_*,a63=.,a 1=1'  
H 2='a1,if,3,zeit_*,a63,-,>,a1=.'  
H 3='a1,RegRel 4=.'
```

Variablen:

a 1: Merker für Stufenstellung 99
a63: Timer für 3s Impuls

Reaktionen auf Flanken und Pegel eines Signals

Aufgabe allgemein:

bei der 0=>1 Flanke eines Signals => Aktion1
bei der 1=>0 Flanke eines Signals => Aktion2
wenn das Signal =1 => Aktion3
wenn das Signal =0 => Aktion4

Lösung allgemein:

„**Signal**“,dup,iff, „**Aktion1**“,else, „**Aktion2**“,endif,if, „**Aktion3**“,else, „**Aktion4**“

Beispiel:

Es sollen zwei Parametersätze verwaltet werden. Über den Binäreingang E7 kann ein Parametersatz angewählt werden. Dies wird anhand des Parameters "zul. Regelabweichung" gezeigt.

Lösung:

- wenn E7 =0
=> Änderungen des 1.Satzes mitverfolgen (aktuelle Parameterwerte in den Wertesatz 1 kopieren)
- bei der 0=>1 Flanke von E7
=> Parametersatz 2 aktivieren (Parameter mit den Wertesatz 2 beschreiben)
- wenn E7 =1
=> Änderungen des 2.Satzes mitverfolgen (aktuelle Parameterwerte in den Wertesatz 2 kopieren)
- bei der 1=>0 Flanke von E7
=> Parametersatz 1 aktivieren (Parameter mit den Wertesatz 1 beschreiben)

H 1='RgE- 7,dup,iff,b2,RegABW =,else,b1,RegABW =,endif,if,RegABW-,b2 =,else,RegABW-,b1 =.'

Variablen:

b1: Wertesatz 1
b2: Wertesatz 2

Auf ein Ereignis verzögert reagieren

Aufgabe:

Die LED 1 soll einschalten, wenn die Regelspannung U1 länger als 60 s über 120V liegt

Lösung:

H 2='htd,RegU1-,120,>,if,b 0,^,if,a++ 31=.,a 31,60,>,b 0=.,endif,else,b 0=0,a 31=0'
H 3='b 0,if,RegLED 1=1,else,RegLED 1=0'
oder
H 1='RegU1-,120,>,dup,a1=.,iff,zeit_*,a31=.'
H 2='a1,if,60,zeit_*,a31,-,<,b0=.,else,b0=0'
H 3='b 0,if,RegLED 1=1,else,RegLED 1=0'

Parameter:

a1 =1, wenn Bedingung erfüllt
b 0 =1, wenn Bedingung erfüllt und die Verzögerung abgelaufen ist
a 31 Zähler für die Verzögerungszeit

Abfrage auf bestimmte Uhrzeit

Aufgabe:

Um 10:30 Uhr soll auf Sollwert 2 geschaltet werden

Lösung:

H 10='hh,10,==,mm,30,==,&,iff,RegSWI =2'

Messwert nur nach größerer Änderung weiterverarbeiten

Aufgabe:

Ein Messwert soll nur übernommen werden, wenn er sich um ein festgelegtes Delta geändert hat.
z.B.: RegU soll im Register a 0 gespeichert werden, wenn die Änderung (in b 0) mehr als 5 V beträgt.

Lösung:

```
H 0='1,iff,a 0=0,b 0=5'  
H 1='RegU-,dup,a 0,-,abs,b 0,>=,if,a 0=.'
```

bedingte Sperre der Tastatur und bestimmter Eingänge

Aufgabe:

Über binäre Eingänge soll die Tastatur und ausgewählte binäre Eingänge gesperrt werden.

Lösung:

```
# sperrt die Tastatur, solange E7=1  
H 1='RegE- 7,if,RegSperreTas =2,else,RegSperreTas =0'  
# sperrt die Eingänge E1...E6, solange E8=1 (sperrn über Bitmaske 0b00111111 = 63)  
H 2='RegE- 8,if,RegSperre_E =63,else,RegSperre_E =0'
```

Tastendrücke simulieren

Aufgabe:

Wenn die zulässige Regelabweichung überschritten wird, soll der Regler in den Reglermodus gehen und auf Grossanzeige wechseln. Wenn die Spannung wieder innerhalb der Regelabweichung ist, verschwindet die Grossanzeige wieder.

Lösung:

```
H 1='Regxw-,abs,1,>,iff,tt qqm1,RegBigDisp=1,else, RegBigDisp=0'
```

über Anwendermenü einstellbares REG-L -Register

Aufgabe:

Das Register a 2 soll im Anwendermenü verändert werden

Lösung:

```
# Defaultwert setzen  
H 1='1,iff,a2=55'  
# alle Menüs löschen  
H 2='1,iff,menuappn *=""',menuapp *=-1'  
# Anwendermenü Zeile 1  
H 3='1,iff,menuappn 1="Para a2",menuapp 1=103'  
# Menü definieren  
P 3='menuedit a 2 ="a2 [1..1000]" 1 1000'
```

Variablen:

a2: einzustellender Parameter

Betriebsstundenzähler

Aufgabe:

Über das Anwendermenü kann die Zeit abgefragt werden, wie lange der Regler schon am Netz angeschlossen ist.

Über „Master-Reset“ kann die Zeit zurückgesetzt werden, ansonsten wird der Wert über Batterie gepuffert.

Lösung:

Betriebszeit in Sekunden ermitteln und in Stunden umrechnen

```
H 1='1,iff,menuappn 1="Betriebsst.",menuapp 1=1'
```

```
P 1='zeit_-,3600,/ ,meld2 "Betriebsstunden" "%.2! h"'
```

Anzeige der Regelabweichung mit Leuchtdioden

Aufgabe:

Es soll über die LEDs angezeigt werden, wenn die Netzspannung aus dem Toleranzband läuft:

LED 1: Spannung hat das Toleranzband um das Doppelte überschritten

LED 2: Spannung hat das Toleranzband überschritten

LED 3: Spannung hat das Toleranzband unterschritten

LED 4: Spannung hat das Toleranzband um das Doppelte unterschritten

Lösung:

```
H 1='RegXw-,1,>, if, RegLED 2 = 1, else, RegLED 2 = 0'
```

```
H 2='RegXw-,1,<, if, RegLED 3 = 1, else, RegLED 3 = 0'
```

```
H 3='RegXw-,2,>, if, RegLED 1 = 1, else, RegLED 1 = 0'
```

```
H 4='RegXw-,2,<, if, RegLED 4 = 1, else, RegLED 4 = 0'
```

verkürzte Form:

```
H 1='RegXw-,1,>,RegLED 2 =.'
```

```
H 2='RegXw-,1,<,RegLED 3 =.'
```

```
H 3='RegXw-,2,>,RegLED 1 =.'
```

```
H 4='RegXw-,2,<,RegLED 4 =.'
```

zusätzlich werden hier die Meldungen bei Stillsetzung unterdrückt:

```
H 0='RegRelFW- 13,b00=.'
```

```
H 1='RegXw-,1,>,b00,^,&, RegLED 2 =.'
```

```
H 2='RegXw-,1,<,b00,^,&, RegLED 3 =.'
```

```
H 3='RegXw-,2,>,b00,^,&, RegLED 1 =.'
```

```
H 4='RegXw-,2,<,b00,^,&, RegLED 4 =.'
```

Meldung, wenn der Schreiber-Speicher voll ist

Aufgabe:

Relais 5 soll anziehen, wenn der Schreiber-Speicher über 80% voll ist

Lösung:

```
H 2='1,iff,RegRelFu 5=1'
```

```
H 3='FSize- rec p,80,>,RegRel 5=.'
```

Automatikbetrieb über Binären Eingang aktivieren

Aufgabe:

Mit dem Eingang E7 soll der AUTO-Betrieb erzwungen werden. Nach Deaktivierung des Eingangs soll auf HAND-Betrieb geschaltet werden, aber die Umschaltung HAND/AUTO über die Tastatur soll nun wieder möglich sein.

Lösung:

```
# dauerhaft auf Auto wechseln  
H 1='RegE- 7,if,RegAUTO=1'  
# einmalig auf Hand wechseln  
H 2='RegE- 7,^,iff,RegAUTO=0'
```

Stromeinfluß deaktivieren wenn Wirkleistung kleiner Null

Aufgabe:

Wird die Wirkleistung kleiner Null soll der Stromeinfluß (Stromprogramm) solange deaktiviert werden, bis die Wirkleistung wieder größer Null wird.

Lösung:

```
# Wenn Leistung  $P < 0$  dann soll Stromeinfluß deaktiviert werden  
H 1='RegP- 0,<,if,RegStromPR=0,else,RegStromPR=1'
```

Automatisches Erzeugen einer Gruppenliste

Aufgabe:

Bei allen Reglern (bis 10; über ELAN vernetzt) soll erkannt werden ob sie auf Schiene 1 (Eingang 7) oder auf Schiene 2 (Eingang 8) aufgeschaltet sind. Dementsprechend soll die Gruppenliste in den einzelnen Reglern gebildet werden. Wenn mindestens zwei Geräte auf einer Schiene arbeiten soll bei diesen der Parallelbetrieb aktiviert werden.

Lösung:

```
# Eingänge parametrieren  
H 1='RegEFu 7+8=1'  
# Zustand von E7 und E8 auf Merker legen  
H 2='RegE- 7,a1=.,RegE- 8,a2=.'  
# Schalterzustände auswerten  
H 3='a1,if,a2,if,a10=0,else,a10=1,eif,else,a2,if,a10=2,else,a10=0'  
# Gruppenliste erzeugen  
H 4='a11=1'  
H 5='a10,if,all r,a10,aa:a10,==,if,kenn,aa:,a11,RegParaGr .=. ,a++ 11,endif,nexta'  
H 6='a11,10,for,i,RegParaGr .="---",nexti'  
# Parallelbetrieb aktivieren  
H 7='a10,a11,2,>,&,if,RegParaPROGA=1,else,RegParaPROGA=0'
```

Variablen:

a01: Merker für Eingang E7 (Trenner für SS1)
a02: Merker für Eingang E8 (Trenner für SS2)
a10: Zeiger für Schalterkombinationen
0= Trafo hängt an keiner Schiene oder an beiden Schienen
1= Trafo hängt an Schiene 1
2= Trafo hängt an Schiene 2
a11: Zähler für laufende Nr. in der Gruppenliste

Master / Slave Funktion über Eingang aktivieren

Aufgabe:

Über den Eingang E8 soll einer der beiden Regler A: oder B: Master werden und der andere Slave

Lösung:

```
H 1='RegEFu 8=1,RegLEDFu 2=1'
```

```
H 3='A:RegE- 8,if,B:RegE- 8,if,a0=3,else,a0=1,endif,else,B:RegE- 8,if,a0=2,else,a0=0'
```

```
H 5='Kenn,1,==,if,a0,1,==,if,P 0,else,P 1'
```

```
H 7='Kenn,11,==,if,a0,2,==,if,P 0,else,P 1'
```

```
H 9='Kenn,1,==,Kenn,11,==,|,^,if,meld "falsche Kennung",P 1'
```

```
H 11='a0,3,==,if,meld2 "Signalzustand an" "E 8 ist falsch"
```

```
P 0='RegParaPr =6,RegParaGr 1= A,RegParaGr 2= B,RegParaProgA =1,RegLED 2=1'
```

```
P 1='RegParaPr =0,RegParaGr 1=---,RegParaGr 2=---,RegParaProgA =0,RegLED 2=0'
```

Variablen:

a0: Zustand der Eingänge E8 an Regler A: und B:

Regelgröße selbst bestimmen und dem Regler zuführen

Aufgabe:

Der Mittelwert der Spannungen U1 und U2 soll als Regelgröße verwendet werden.

wenn eine der beiden Spannungen unter 80% des Sollwertes abfällt soll nur noch mit der anderen Spannung weiter geregelt werden.

Lösung:

```
# Spannungen U1 und U2 auf Mapping per H-Programm parametrieren
```

```
H 1='RegMiMap 1=-1,RegMiMap 2=-1'
```

```
# Wert der Spannungen U1 und U2 auf Merker legen
```

```
H 2='RegMiVal*- 1,b11=.,RegMiVal*- 2,b12=.,'
```

```
#überprüfen ob eine Spannung kleiner 80% vom Sollwert.
```

```
H 3='RegSWI-,RegSW- .,b11,Pick 2,/,0.8,<,if,b12,b11=.
```

```
H 4='RegSWI-,RegSW- .,b12,Pick 2,/,0.8,<,if,b11,b12=.'
```

```
# Mittelwertbildung
```

```
H 5='b11,b12,+,2,/,RegMiVal 1=.'
```

Parameter:

b11 RegMiVal* 1, ungemappte Spannung U1

b12 RegMiVal* 2, ungemappte Spannung U2

Wechseln zwischen den Regelgrößen U1 und U2

Aufgabe:

Über den Eingang E7 soll auf die zweite Spannung geregelt werden, der zweite Sollwert verwendet werden und der Knu verändert werden.

Bemerkung:

Für diese Funktion ist das Merkmal "Dreiwickler" notwendig.

Lösung:

```
# wenn am Eingang E7 ein Pegel ansteht wird
```

```
# - auf U2 geregelt, ansonsten U1
```

```
# - der Sollwert SW2 verwendet, ansonsten SW1
```

```
# - der Knu von 300 verwendet (30KV), ansonsten Knu von 100 (10KV)
```

```
H 1='RegEFu 7=1'
```

```
H 2='RegE- 7,if,Reg3WSELU=2,RegSWI=2,RegKnu=300,else,Reg3WSELU=1,RegSWI=1,RegKnu=100'
```

Ansprechen der einzelnen Relais einer BIN-D - Karte

Aufgabe:

Relais 2 der BIN-D (an Hardware Adresse 0 einstellen) setzen

Lösung:

allgemein

```
DevRel <adresse> <RelaisNr>=<bool>
```

oder

```
DevRel* <adresse> =<byte> (wobei: bit0 => Relais1, bit1 => Relais2, ...)
```

```
H 1='DevRel 0 2=1'
```

oder

```
H 1='DevRel* 0=0b00000010'
```

blinkende Meldung

Aufgabe:

eine Meldung soll im Display blinkend erscheinen (2s Zykluszeit)

Lösung:

```
H1='ss,1,&&,meld2_ "blinkende" "Meldung" . '
```

Einstellungen für 2. RelaisKarte im EEPROM

Aufgabe

Notwendige Programmierung um eine Relais-Karte im Steckplatz 4 in der Firmware bekannt zu machen.

Lösung

```
sysopen__
```

```
syseep 4 0 = 0x2929
```

```
syseep 4 1 = 2
```

```
syseep 4 3 = 0
```

PowerOn-Reset nach jedem H-Prog-Download

Problem

Syseep-Einstellungen (z.B. für 2. RelaisKarte im REG-D) werden erst nach PowerOn-Reset wirksam

Lösung

```
H 30='1,iff,meld2 "Reset nach" "H-Prog laden",pause 3,sysreset=0'
```

Umschalten zwischen Sommer- und Winterzeit

Aufgabe

Sommer- Winterzeitumschaltung über internes Flag

Lösung

```
# summer-/winter-time
```

```
H 31='SOWI, IF, ZEIT-, +, ZEIT=.'
```

Umschalten zwischen Sommer- und Winterzeit auserhalb der MEZ

Aufgabe

Sommer- Winterzeitumschaltung an bestimmten Tag (15.April und 15.Oktober) mit einer bestimmten Zeitdifferenz (+2,5h, -2,5h) um 00:10Uhr umstellen

Lösung

```
# summer-/winter-time
H 30='If 15.4 0h10,Zeit-,9000,+,Zeit=.,eif, if 15.10 0h10,Zeit-,9000,-,Zeit=.,eif,'
```

Zeitsynchronisierung über einen Minutenimpuls

Aufgabe

Über einen Minutenimpuls auf den binären Eingang E16 am REG-D "A:" soll die Zeit synchronisiert werden.

Einmal täglich sollen alle anderen RegSys Geräte im ELAN-Verbund auf die Zeit von Regler "A:" synchronisiert werden.

Lösung

```
# time synchronization
H 29='kenn,1,==,if,RegE- 16,iff,ss-,30,<,if,zeit-,ss-, -,zeit=.,else,zeit-,60,ss-, -,+,zeit=!'
H 30='IF 12h00:00,all,a:ZEIT//-,ZEIT//=.,nexta'
```

Sammelmeldung für <U, >U und >I auf LED1 und Relais REL3

Aufgabe

Sammelmeldung für <U, >U und >I erzeugen und auf die LED1 und das Relais REL3 legen.

Lösung

```
# Sammelmeldung für <U, >U und >I
H 10='RegLedFu 1=1,RegRelFu 3=1'
H 11='RegRelFW- 5,RegRelFW- 6,|,if,RegLed 1=1,RegRel 3=1,else, RegLed 1=1,RegRel 3=1'
```

Spannungsabsenkung über binäre Eingänge

Aufgabe

Über zwei binäre Eingänge (E9, E10) soll der aktive Sollwert um je einen frei definierbaren (Anwendermenü) Wert abgesenkt werden, solange das Signal ansteht.

Lösung

```
#### Spannungsabsenkung über binäre Eingänge
# Eingänge parametrieren
H 1='1,iff,RegEFu 9+10=1,a1=5,a2=10'
# Eingänge auswerten
H 2='RegE- 9,^,if,RegE- 10,if,a10=2,else,a10=0,eif,else,RegE- 10,if,a10=0,else,a10=1'
# Anwendermenü
H 3='1,iff,menuappn 1="SW - %,E9",menuapp 1=101, menuappn 2="SW - %,E10",menuapp 2=102"
# Sollwert absenken
H 4='a10,iff,RegSWI-,RegSW- .,dup,a11=.,a10,a .,-,RegSWI-,RegSW .=.,else,a11,RegSWI-,RegSW .='
# Anwendermenü
P 1='muedit a 1=" SW-%,E9 0-20%" 0 20'
P 2='muedit a 2=" SW-%,E10 0-20%" 0 20'
```

Toleranzband der Spannung überwachen

Aufgabe

Ausgang "Regler gestört" (Relais REL3) setzen, wenn die Spannung während einer Zeitspanne von 15min nie in das Toleranzband geregelt werden konnte.

Lösung

```
# Relais parametrieren
H 1='RegRelFu 3=1'
# Toleranzband überwachen
H 2='htd,RegOFFTOL-,if,a++ 1=.,a1,900,>,RegRel 3=.,else,a1=0,RegRel 3=0'
```

Quittierungsfunktion verändern

Aufgabe

Der Regler soll über den binären Eingang E9 quittiert werden.

Lösung

```
# Quittierung festlegen
H 1='RegACKBL 1+4=1'
H 2='RegE- 9,iff,RegDoack=0'
```

Auf bestimmte Ereignisse reagieren

Aufgabe

Mit dem "Test" Taster am PAN-D soll das Relais REL14 gesetzt werden.

Lösung

```
# Relais parametrieren
H 5='RegRelFu 14=1'
# Auf Ereignis "ESC" länger 1s gedrückt, P1 aufrufen
H 6='EventApp 13=p1'
# Relais setzen
P 1='RegRel 3=1'
```

Displaybeleuchtung über binären Eingang aktivieren

Aufgabe

Mit einem Impuls am binären Eingang E7 soll die Displaybeleuchtung eingeschaltet werden. Diese schaltet sich nach 15min automatisch wieder aus.

Lösung

```
# Eingänge parametrieren
H 1='RegEFu 7=1'
# Display-Beleuchtung aktivieren
H 2='RegE- 7,iff,tt !'
```

Eine oder mehrere Meldungen im Display anzeigen

Aufgabe

Mit den binären Eingängen E9..E16 können im Display vordefinierte Meldungen einzeln oder auch mehrere gleichzeitig angezeigt werden.

Mit dem biären Eingang E8 kann die Meldung wieder zurückgenommen werden.

Lösung

```
# input assignments
```

```
H 1='1,iff,RegEFU- 8..16=1,b 11..18=41'
```

```
# meld
```

```
H 2='b1,b2,|,b3,|,b4,|,b5,|,b6,|,b7,|,b8,|,if,P1'
```

```
# reset
```

```
H 3='RegE- 8,iff,b 1..8=0'
```

```
# variable
```

```
H 10='RegE- 9 ,if,b1=5 ,eif,RegE- 10,if,b2=10,eif,RegE- 11,if,b3=15,eif,RegE- 12,if,b4=20,eif'
```

```
H 11='RegE- 13,if,b5=25,eif,RegE- 14,if,b6=30,eif,RegE- 15,if,b7=35,eif,RegE- 16,if,b8=40,eif'
```

```
# summer-/winter-time
```

```
H 31='SOWI, IF, ZEIT-, +, ZEIT=.'
```

```
# trips and alarms
```

```
P 0=' BucM BucT PRel TmpT BucG MotO Tmp1 Tmp2 '
```

```
P 1='b8,b7,b6,b5,b4,b3,b2,b1, meld2 "%-5.*p0%-5.*p0%-5.*p0%-5.*p0" "%-5.*p0%-5.*p0%-5.*p0%-5.*p0" 3'
```

Sollwert als BCD-Code auf BIN-D ausgeben

Aufgabe

Der aktuelle Sollwert zwischen 90..110V soll als BCD – Code (Wert 1..21) auf die BIN-D ausgegeben werden.

Lösung

```
# Sollwert als BCD-Code über BIN-D ausgeben, begrenzt auf 90..110 (BCD 1 ... 21)
```

```
H 1='RgSWI-,RgSW- .,89,-,a 00=.'
```

```
H 2='a 00,1,<,if,a 00=0'
```

```
H 3='a 00,21,>,if,a 00=0'
```

```
H 4='a 00,BIN2BCD,DevRel* 0=.'
```

Berechneten Meßwert als Regelgröße verwenden

Aufgabe

Wenn nur ein Wandler 10kV/100V für einen 5kV Trafo zur Verfügung steht soll der Regler aus den gemessenen 50V Nennspannung trotzdem 100V Regelspannung sehen.

Dadurch geht allerdings der Phasenbezug zwischen Strom und Spannung verloren und der Regler kann keine Phasenverschiebung messen und entsprechend auch keine Blindleistung etc. berechnen.

Lösung

```
# Meßwandlereingang auf REG-L umschalten und Werte vom Wandlereingang umrechnen
```

```
H 1='1,iff,RegMiMap 1=-1'
```

```
H 2='RegMiVal*- 1,2,*,RegMiVal 1=.'
```

Wandlerdaten parametrieren

Aufgabe

Über das Anwendermenü sollen die Wandlerdaten für KNU1, KNI1, KNU2 und KNI2 parametrierbar sein.

Lösung

Anwendermenü

H 1='1,iff,menuappn 1="knu 1",menuapp 1=101,a1=66 ,menuappn 2="kni 1",menuapp 2=102,a2=1'

H 2='1,iff,menuappn 3="knu 2",menuapp 3=103,a3=110,menuappn 4="kni 2",menuapp 4=104,a4=1'

Eingänge E3 und E4 auswerten

H 3='1,iff,RegEFu 3+4=1'

H 4='RegE- 3,RegE- 4,^,&,if,a1,RegKnu=.,a2,RegKni=.'

H 5='RegE- 4,RegE- 3,^,&,if,a3,RegKnu=.,a4,RegKni=.'

Anwendermenü

P 1='menuedit a 1 ="knu 1 [1..4000]" 1 4000'

P 2='menuedit a 2 ="kni 1 [1..10000]" 1 10000'

P 3='menuedit a 3 ="knu 2 [1..4000]" 1 4000'

P 4='menuedit a 4 ="kni 2 [1..10000]" 1 10000'

variable

a1: Merker für knu 1

a2: Merker für kni 1

a3: Merker für knu 2

a4: Merker für kni 2

Überwachen auf Frequenzabweichung

Aufgabe

Wenn die Frequenz um mehr als 0,2Hz von 50Hz abweicht soll der Regler auf HAND umschalten, erst wenn die Frequenz weniger als 0,1Hz abweicht dann soll der Regler wieder auf AUTO umschalten.

Lösung

Frequenzüberwachung

H 1='RegF-,50,-,abs,0.2,>,iff,RegAuto=0'

H 2='RegF-,50,-,abs,0.1,<,iff,RegAuto=1'

E-LAN Error zwischen zwei Geräten auswerten

Aufgabe

E-LAN-Error zwischen zwei Geräten (A: und B:) auswerten (Led1)und erst zurücksetzen wenn die Kommunikation zwischen den Geräten wieder funktioniert.

Lösung

E-LAN Error indication

H 24=a:RegE- 3, b:RegE-3'

H 25='errstat-,65536,&&,if,RegLED 3=1,else,RegLed 3=0'

Leittechnikanbindung mit REG-P (Formatierungsbeispiele)

Formatierungsbeispiele

rege+% "%+_.4w;" 2,a0,+ "%+_.4!;" liefert +1.0000E+00;5.0000E+00; (wenn E2=1 und A0=5)

Sollwert in % auf ganzzahlige Werte runden

Aufgabe

Sollwert in % auf ganzzahlige Werte runden

Hintergrund: Die %-Schrittweite des In-/Decrementieren vom SW bezieht sich auf Unenn=100V statt auf 100%. Dies führt zu ungeraden %-Sollwerten, wenn ein Sollwert ungleich von 100V als 100% definiert ist (die Sollwertabweichung != 100V).

Lösung

aktuellen Sollwert(a01) und SollwertAbweichung(b01) lesen

H 5='RegSWI-,RegSW- .,a01=., RegSWI-,RegSWA- .,b01=.'

Sollwert in % berechnen (a00); %-Sollwert ganzzahlig runden (b00)

H 6='a01,b01,/,100,*,a00=., a00,intr,b00=.'

wenn der Sollwert nicht dem gerundeten %-Wert entspricht, den Sollwert dem Wert anpassen

H 7='b00,100,/,b01,*,a01,-,abs,0.01,>,if, b00,100,/,b01,*,RegSWI-,RegSW- .='

Parameter - Beschreibung

a00: Sollwert in %

b00: gerundeter Sollwert in %

a01: Sollwert (normiert auf Unenn=100V)

b01: SollwertAbweichung (Wert, der als 100% definiert wurde)

Sollwerte wechseln

Auswahl von Sollwert 1..4 über Binäre Eingänge

Aufgabe:

Die 4 Sollwerte des Reglers sollen über Dauersignale an den Eingängen BE 1 ... BE 4 aktiviert werden. Liegt kein oder mehrere Signale an den Eingängen an, soll der Sollwert 1 aktiv sein.

Bemerkung:

Merkmal "4Sollwerte" muß aktiv sein

Lösung:

```
H 4='a 0=0,b 0=0'  
H 5='RegBE- 1,if,b 0=1, a++ 0'  
H 6='RegBE- 2,if,b 0=2, a++ 0'  
H 7='RegBE- 3,if,b 0=3, a++ 0'  
H 8='RegBE- 4,if,b 0=4, a++ 0'  
H 9='a 0,1,==,b 0,0,!,&,if,b 0,RegSWI=.,else,RegSWI=1'
```

Variablen:

a0: Zähler für Anzahl der Eingänge
b0: Merker für den Sollwertzeiger

einmaliges Umschalten auf Sollwert 2

Aufgabe:

nach dem Power-On und bei Umschaltung Hand->Auto soll als Standardsollwert SW 2 aktiv sein:

Lösung:

```
# Sollwert 2 ist aktiv bei PowerOn und Hand->Auto  
H 1='1,iff+,RgSWI=2'  
H 2='RgAuto-,iff,RgSWI=2'
```

Externe Sollwertverschiebung mit Begrenzung

Aufgabe:

Mit den externen Signalen auf E3 und E4 soll der aktuelle Sollwert verstellt werden, jedoch nur innerhalb bestimmter Grenzen:

Ein Pulse am E3 erhöht den Sollwert um 1V (begrenzt auf maximal 105 V)

Ein Pulse am E4 reduziert den Sollwert um 1V (begrenzt auf minimal 95 V)

Lösung:

```
# Externe Sollwertverschiebung mit Begrenzung  
H 2='RegE- 3,iff,RgSWI-,RgSW- .,105,<,if,RgSWI-,RgSW- .,1,+,RgSWI-,RgSW- .,='  
H 3='RegE- 4,iff,RgSWI-,RgSW- .,95,>,if,RgSWI-,RgSW- .,1,-,RgSWI-,RgSW- .,='
```

4 Sollwerte über Anwendermenü auswählbar

Aufgabe:

Im Anwendermenü soll der aktive Sollwert 1..4 anwählbar sein

Bemerkung:

Merkmal „4Sollwerte“ muß aktiviert sein

Lösung:

alle Menüs löschen

H 2='1,iff,menuappn *=""',menuapp *=-1'

Menüs schreiben

H 3='1,iff,menuappn 1="Sollwert1",menuapp 1=1'

H 4='1,iff,menuappn 2="Sollwert2",menuapp 2=2'

H 5='1,iff,menuappn 3="Sollwert3",menuapp 3=3'

H 6='1,iff,menuappn 4="Sollwert4",menuapp 4=4'

Sollwerte setzen

P 1='RegSWI=1'

P 2='RegSWI=2'

P 3='RegSWI=3'

P 4='RegSWI=4'

Stufenstellung abfragen und ändern

Rückführung in eine Neutralstufe

(automatisches Stufen in "Home"-Position)

Aufgabe: Wenn Binäreingang E4 = 1, soll der Regler in den HAND-Betrieb wechseln und automatisch in eine, über Applikationsmenü, festgelegte Stufe fahren.

Wird Binäreingang E4 = 0, soll der Regler in seinen alten Betriebszustand (HAND oder AUTO) zurückwechseln.

Lösung:

#Applikationsmenüs erstellen

H 2='1,iff,menuappn *=""',menuapp *=-1'

H 3='1,iff,menuappn 1="Homing Stufe",menuapp 1=101'

gewählte Homing Stufe auf Merker legen

P 1='menuedit a 10="1..19" 1 19'

#Regler in Homing-Stufe fahren, Stufe frei wählbar, default = 10

H 8='1, iff, a 10=10'

H 9='a 10,INTR,a 10=.'

H 10='RegE- 4,iff, RegAuto-, a 01=.,else, a 01, RegAuto=.'

H 11='RegE- 4, if, RegAuto = 0'

H 12='RegStufe-, a 10, <, b 02=., RegStufe-, a 10, >, b 03=.'

H 13='b 02, b 03, |, b 00=.'

H 14='RegE- 4, b 00, &, b 01=.'

H 15='b 01, iff, a 30=0, b 10=1, else,a 30=0'

H 16='htd, b 01, b 10, ^,&, if, a++ 30 =.,a 30, RegTLAUFL- ,1,+,>,iff,b 10=1,a 30=0'

H 17='b 01, b 10,&, if, b 10=0, b 02, if, RegHoeher* =1, endif, b 03, if, RegTiefer* =1'

Variablen:

a 01: alter Betriebszustand (0:Hand, 1:AUTO)

a 10: Neutralstufe, default = 10

a 30: Timer für Stellbefehle

b 00=1: Stufe zu tief oder zu hoch

b 01=1: autom. Stellvorgang aktiv

b 02=1: Stufe zu tief

b 03=1: Stufe zu hoch

b 10=1: Stellbefehl ausführen

Stufungen unterdrücken außerhalb der min. / max. Stufenstellung

Aufgabe:

Stellbefehle, die zur Überschreitung der Min-/ Max- Grenzen (z.B. 5 ... 20) führen würden, werden unterdrückt.

Wirkt nicht im „Hand“-Betrieb!

Lösung:

H 1='RegStufe-,5,<=,if,RegSperreT =3,else,RegSperreT =0'

H 2='RegStufe-,20,>=,if,RegSperreH =3,else,RegSperreH =0'

Einstellbare Begrenzung der min. / max. Stufenstellung

Aufgabe:

Stellbefehle, die zur Überschreitung der Min-/ Max- Grenzen, die im Anwendermenü einstellbar sind, führen würden, werden unterdrückt. Die Unterdrückung wirkt nicht im "Hand"-Betrieb!
Die Grenzen sind einstellbar im Bereich 1...19. Defaultwerte: Min=4, Max=15.

Lösung:

```
#Anwendermenü erstellen
H 1='1,iff,menuappn *=""',menuapp *=-1'
H 2='1,iff,menuappn 1="min. Stufe",menuapp 1=101'
H 3='1,iff,menuappn 2="max. Stufe",menuapp 2=102'
#Stufenbegrenzung, Stufe frei wählbar, default min.Stufe = 4, max.Stufe =15
H 6='1, iff, a 11=4, a 12=15'
H 7='RegStufe-,a 11,<=,if,RegSperreT =3,else,RegSperreT =0'
H 8='RegStufe-,a 12,>=,if,RegSperreH =3,else,RegSperreH =0'
#Applikationsmenü 1, min. Stufe einstellen
P 1='menuedit a 11="Stufe 1..19" 1 19, a 11,INTR,a 11=.'
#Applikationsmenü 2, max. Stufe einstellen
P 2='menuedit a 12="Stufe 1..19" 1 19, a 12,INTR,a 12=.'
# Variablen:
# a 11: Stufenbegrenzung unten, default = 4
# a 12: Stufenbegrenzung oben, default = 15
```

Überwachung des Stufenschalter

Aufgabe:

folgende Ereignisse sollen erkannt werden:

Allgemeines:

- die einzelnen Überwachungsfunktionen können als H-Programm-Module, einzeln und nach Bedarf, in den REG-D oder PAN-D geladen werden können.
- inverse Stufenschalter sollen berücksichtigt werden.

- Stufung ohne Lauflampe:

Wenn nach einen Stellbefehl vom Regler (im Auto- oder Handbetrieb) innerhalb von ca. 3 sec kein Lauflampensignal erkannt wird.

- Stufung ohne Stellbefehl:

Wenn ohne einen Stellbefehl vom Regler ein Lauflampensignal erkannt wird.

- Maximale Lauflampenzeit überschritten:

Wenn das Lauflampensignal länger als die parametrisierte Zeit aktiv ist.

- Stufung in falsche Richtung:

Wenn nach einem Höher- Stellbefehl der BCD-Code um1 kleiner wird.
Wenn nach einem Tiefer- Stellbefehl der BCD-Code um 1 größer wird.

- Falscher BCD-Code:

Wenn die Eingangsfunktionen "BCD10" oder "BCD20" verwendet werden und ein ungültiger BCD-Code erkannt wird.

- Stufung ohne BCD-Änderung:

Wenn sich nach einen Stellbefehl der BCD-Code nicht ändert.

Programmzeilen für REG-D geschrieben, für PAN-D müssen die entsprechenden

```
### Zeilen durch die mit "PAN-D" gekennzeichneten Zeilen ersetzt werden.
##### Stufungen ohne Lauflampe #####
# wenn Stufung läuft
H 2='RegRel- 1,RegRel- 2,|,iff,a48=1,a60=0'
# wenn Lauflampensignal nach 3s noch nicht ansteht -> FehlerFlag
H 3='htd,a48,if,a++ 60=.,RegLedFW- 38,if,a48=0,endif,a60,3,>,if,a51=1,a48=0,endif,else,a60=0'
# "PAN-D" #
#H 3='htd,a48,if,a++ 60=.,RegRelFW- 19,if,a48=0,endif,a60,3,>,if,a51=1,a48=0,endif,else,a60=0'
```

```
##### Stufungen ohne Stellbefehl #####
# wenn Stufung läuft
H 5='RegRel- 1,RegRel- 2,|,iff,a49=1,a61=0'
H 6='htd,a49,if,a++ 61=.,RegTLauf-,a61,<,if,a49=0'
# Wenn Lauflampensignal ausserhalb dieser Zeit im Auto-Modus kommt -> FehlerFlag
H 7='RegLedFW- 38,iff,RegAuto-,a49,^,&,if,a52=1'
# "PAN-D" #
#H 7='RegRelFW- 19,iff,RegAuto-,a49,^,&,if,a52=1'
```

```
##### Maximale Lauflampenzeit überschritten #####
# wenn Lauflampenzeit überschritten -> FehlerFlag
H 9='htd,RegLedFW- 38,a53,0,==,&,if,a++ 62=.,RegTLauf-,a62,<,if,a53=1,a62=0,endif,else,a62=0'
# "PAN-D" # "RegRelFW- 08=1 beinhaltet die gleiche Funktion"
```

```
##### Für diese Funktionen muß die Stufenstellung an den Regler zurückgeführt werden #####
```

```
##### allgemeiner Teil #####
# inverser Stufenschalter, oder normaler
H 10='merkmal- invers,if,a57=-1,else,a57=1'
#aktuelle Stufe erfassen
H 11='RegStufe-,b50=.'
# wenn HÖHER => erwartete Stufe = aktuelle + 1; "Stufung läuft" = 1
H 12='RegRel- 1,iff,b50,a57,+,b51=.,a57,a50=.'
# "PAN-D" #
#H 12='ar:RegRel- 1,iff,b50,a57,+,b51=.,a57,a50=.'
# wenn TIEFER => erwartete Stufe = aktuelle - 1; "Stufung läuft" = 1
H 13='RegRel- 2,iff,b50,a57,-,b51=.,a57,-1,*,a50=.'
# "PAN-D" #
#H 13='ar:RegRel- 2,iff,b50,a57,-,b51=.,a57,-1,*,a50=.'
# Stufendifferenz berechnen
H 14='b50,b51,-,a50,*,b52=.'
# wenn "Stufung läuft" && Differenz==0 => "Stufung läuft" beenden
H 15='a50,0,!=",b52,0,==,&,if,a50=0'
```

```
##### Stufungen in falsche Richtung #####
##### allgemeinen Teil einfügen #####
# wenn "Stufung läuft" && Differenz=-2 => "Stufung läuft" beenden und "Stufung in falsche Richtung" setzen
H 17='a50,0,!=",b52,-2,==,&,if,a50=0,a54=1'
```

```
##### Falscher BCD-Code #####
##### allgemeinen Teil einfügen #####
# wenn "Stufung läuft" && Differenz >0 | <-2 => "Stufung läuft" beenden und "falscher BCD-Code" setzen
H 19='a50,0,!=",b52,0,>,b52,-2,<,|,&,if,a50=0,a55=1'
# wenn "aktuelle Stufe" = 99 => "Stufung läuft" beenden und "falscher BCD-Code" setzen
H 20='b50,99,==,if,a50=0,a55=1'
```

```
##### Stufungen ohne BCD-Änderung #####
##### allgemeinen Teil einfügen #####
# wenn "Stufung läuft" => Timer inkrementieren else Timer =0
```

```
H 22='htd,a50,0,!=,if,a++ 63=.,else,a63=0'  
# wenn "Stufung läuft" && Differenz -1 && Timer > max.LaufLampenZeit=>"Stufung läuft"beenden;"Stufung  
ins Leere"setzen  
H 23='a50,0,!=,b52,-1,==,&,if,a63,RegTLaufl-,>,if,a50=0,a56=1'
```

```
##### Variablen #####  
# a48 Flag "Stufung läuft" Stufungen ohne Lauflampe  
# a49 Flag "Stufung läuft" Stufungen ohne Stellbefehl  
# a50 Flag "Stufung läuft" allgemeiner Teil  
# a51 FehlerFlag "Stufung ohne Lauflampe"  
# a52 FehlerFlag "Stufung ohne Stellbefehl"  
# a53 FehlerFlag "Lauflampenzeit überschritten"  
# a54 FehlerFlag "Stufung in falsche Richtung"  
# a55 FehlerFlag "falscher BCD-Code"  
# a56 FehlerFlag "Stufung ins Leere"  
# a60 Timer für 3 s  
# a61 Timer für eine Lauflampenperiode  
# a62 Timer für Lauflampe  
# a63 Timer für Stufung läuft  
# b50 aktuelle Stufe  
# b51 erwartete Stufe  
# b52 Stufendifferenz zwischen akt. und erwartet
```

Zählen der Stufungen pro Zeitraum; mit Reset

Aufgabe:

Im Anwendungsmenü soll abrufbar sein:

```
„Stufungen“ F1: "Anzahl der Stufungen" und "Zeitraum" seit letztem Reset  
„Reset“ F2: Setzt die Zähler für Stufungen und Zeitraum zurück
```

Lösung:

```
# Menüs löschen  
H 1='1,iff,menuappn *="",menuapp *=-1'  
#Menü erzeugen  
H 2='1,iff,menuappn 1=Stufungen,menuapp 1=1'  
H 3='1,iff,menuappn 2=Reset,menuapp 2=2'  
# Zähler rücksetzen beim Download  
H 4='1,iff,zeit-,a01=.,a00=0'  
# Stufungen hochzählen (a00)  
H 5='RegRel- 1,RegRel- 2,|,iff,1,a00++=.'  
# Meldung ausgeben (wird im Anwendermenü F1 ausgeführt)  
P 1='zeit-,a01,-,86400/,/dup,frac,86400,*,pick 2,int,a00,meld2 "Stufungen %!" "in %!Tagen %zz" 5'  
# die Zähler rücksetzen (wird im Anwendermenü F1 ausgeführt)  
P 2='zeit-,a01=.,a00=0'
```

Variablen:

```
a00: Zähler für Stufungen  
a01: Merker für Zeit nach Reset oder Download
```

Stufengleichheit zwischen verschiedenen Reglern herstellen

Aufgabe:

Um Zuschaltbedingung für den Parallelbetrieb herzustellen soll der Regler B: auf Stufengleichheit zu Regler A: gebracht werden, wenn am Regler A: an Eingang E3 ein Signal ansteht. Dazu werden beide Regler auf HAND gestellt.

Lösung: (H-Prog für Regler 'A')

```
### Wenn E3 --> dann...
# ... Merker für E3, Regler A: und B: auf Hand-Betrieb setzen
H 2='RegE- 3,iff,a1=1,RegAUTO=0,B:RegAUTO=0'
# ... Stufenstellung von Regler A: auf Merker
H 3='a1,iff,RegStufe-,a10=.'
# ... Stufe von Trafo B: an A: angleichen
H 4='B:RegStufe-,a10,<,b02=.,B:RegStufe-,a10,>,b03=.'
H 5='b 02, b 03, |, b 00=.'
H 6='a1,b00,&,b01=.'
H 7='b01,iff,a30=0,b10=1,else,a 30=0'
H 8='htd,b01,b10,^,&,if,a++ 30 =.,a 30,RegTLAUFL-,1,+,>,iff,b10=1,a30=0'
H 9='b01,b10,&,if,b10=0,b02,if,B:RegHoeher* =1,endif,b03,if,B:RegTiefer* =1'
# ... Merker für Stufengleichheit vorhanden
H 10='b00,^,if,a2=1,else,a2=0'
# Stufen angleichen: Abbruch nach 40 Sekunden
H 11='htd,b01,if,a++ 31=.,a31,40,>,if,a1=0,RegRel 5=1,eif,else,a31=0'
# Anwendermenü
H 12='1,iff,menuappn 1="Quittieren",menuapp 1=1'
# Anwendermenü ###
P 1='RegRel 5=0'
```

Variablen:

```
a01: Merker für Eingang E3
a02: Stufengleichheit vorhanden
a10: Merker für Stufenstellung von A: wenn E3 ansteht
a30: Timer für Stellbefehle
a31: Timer für Abbruch Stufengleichheit
b00: Stufe zu tief oder zu hoch
b01: autom. Stellvorgang aktiv
b02: Stufe zu tief
b03: Stufe zu hoch
b10: Stellbefehl ausführen
```

alternative (vereinfachte) Aufgabenstellung:

Der Regler stuft auf die Stufenstellung von Regler B, wenn am Eingang E3 ein Signal ansteht. (ohne Überwachungsfunktionen)

alternative Lösung:

```
#Stufenabgleich aktivieren, wenn E3=1
H 1='RegE- 3,b2=.'
#Stufendifferenz (a1 = Ist - Soll) berechnen
H 2='b2,if,RegStufe-,B:RegStufe-,-,a1=.'
# wenn (Stufenabgleich aktiv & keine Stufung läuft & Stufendifferenz != 0)
# Flag "Stufung läuft"=1 setzen
# wenn (Differenz > 0) Tiefer- sonst Höher stufen
H 3='b2,b1,^,&,a1,abs,0,!=,&,if,b1=1,a1,0,>,if,RegTiefer*=1,else,RegHoeher*=1'
# Flag "Stufung läuft" nach der Laufampenzeit+1 zurücknehmen
H 4='htd,b1,if,b++ 11=.,b11,RegTlauf-,1,+,>,if,b1=0,b11=0,eif,else,b11=0'
```



```
##### Parameter - Beschreibung #####  
#a1 Stufendifferenz Ist-Soll  
#b1 =1: Flag "Stufung läuft"  
#b2 =1: Stufenabgleich aktiv  
#b11 Timer für Stufung läuft
```

Ausgabe der Stufenstellung als BCD-Code

Aufgabe

Die Stufenstellung 0...+39 des Transformators soll über die binären Ausgänge des Reglers als BCD-Code ausgegeben werden. Größere Stellungen werden als ungültigen BCD-Code ausgegeben.

Lösung

```
# Relais und BA's parametrieren  
H 1='RegRelFu 3..5=1,RegBAFu 1..4=1'  
# Stufenstellung lesen  
H 2='RegStufe-,a0=.'  
# neg. Stufenstellungen verarbeiten (a11=1)  
H 3='a0,0,<,if,a11=1,a0,abs,a0=.,else,a11=0'  
# BCD-Code Vorzeichen setzen (Rel 3)  
H 4='a11,RegREL 3=.'  
# BCD-Code Ausgänge für 1,2,4,8, 10, 20 setzen (BA 1..4, Rel 4..5)  
H 5='a0,dup,20,>=,dup,RegRel 5=.,if,20,-,a0=.'  
H 6='a0,dup,10,>=,dup,RegRel 4=.,if,10,-,a0=.'  
H 7='a0,dup, 8,>=,dup,RegBA 4=.,if, 8,-,a0=.'  
H 8='a0,dup, 4,>=,dup,RegBA 3=.,if, 4,-,a0=.'  
H 9='a0,dup, 2,>=,dup,RegBA 2=.,if, 2,-,a0=.'  
H10='a0,dup, 1,>=,dup,RegBA 1=.,if, 1,-,a0=.'
```

Stufenstellung als BCD-Code auf Relais ausgeben

Aufgabe

Die aktuelle Stufenstellung soll über die Relaisausgänge als BCD-Code ausgegeben werden

Lösung

```
# Eingänge, Relais und LED's parametrieren  
H 1='RegRelFu 4..5=1,RegBAFu 1..4=1'  
# Ausgabe der Stufenstellung über Relais als BCD Code  
H 5='RegStufe-,bin2bcd,a1=.'  
H 6='a1,32,>=,if,RegBA 4=1,32,a-- 1=.,else,RegBA 4=0'  
H 7='a1,16,>=,if,RegBA 3=1,16,a-- 1=.,else,RegBA 3=0'  
H 8='a1,8,>=,if,RegBA 2=1,8,a-- 1=.,else,RegBA 2=0'  
H 9='a1,4,>=,if,RegBA 1=1,4,a-- 1=.,else,RegBA 1=0'  
H 10='a1,2,>=,if,RegRel 5=1,2,a-- 1=.,else,RegRel 5=0'  
H 11='a1,1,>=,if,RegRel 4=1,1,a-- 1=.,else,RegRel 4=0'
```

Analoge Ein- / Ausgänge

Analogausgang allgemein

Analogausgang konfigurieren:

AnaModSel =	AnaSSel =
0: -10 V ... 0 ... 10 V	0: + / -
1: -20 mA ... 0 ... 20 mA	1: +
2: -5 mA ... 0 ... 5 mA	2: -
3: S0 ??	
4: 4 mA ... 20 mA	

Analogausgang skalieren:

Ana = AnaN * AnaFactor + AnaOffset

wobei für AnaModSel=1: AnaN: -1 ... 1 => Ausgang: -20 mA ... 20 mA

Aufgabe 1:

Die normierte Regelspannung U1N soll als mA-Signal am Analogausgang Ana1 ausgegeben werden. Die Ausgabe soll folgendermaßen skaliert werden: U1N: 0 ... 120 V => Ana 1: 0 ... 20 mA

Lösung:

```
# konfigurieren von Ana 1
H 1='1,iff,AnaModSel 1 =1,AnaSSel 1 =1,AnaFACTOR 1 =20,AnaOFFSET 1 =0'
#skalieren von Ana 1
H 2='RegU1N-,6,/,Ana 1=.'
```

Aufgabe 2:

Die normierte Regelspannung U1N soll als mA-Signal am Analogausgang Ana1 ausgegeben werden. Die Ausgabe soll folgendermaßen skaliert werden: U1N: 0 ... 100 V => Ana 1: 4 ... 10 mA

Lösung:

```
# konfigurieren von AnaN 1 (0...100V => 4...10mA => 0,2...0,5AnaN)
H 1='1,iff,AnaModSel 1 =1,AnaSSel 1 =0 '
#skalieren von AnaN 1
H 2='RegU1N-,333.333,/,0.2,+,AnaN 1=.'
```

Analogausgang mit geknickter Kennlinie

Aufgabe:

Die normierte Regelspannung U1N soll als mA-Signal am Analogausgang Ana1 ausgegeben werden. Die Ausgabe soll folgendermaßen skaliert werden:

U1N = 0 ... 80 ... 120 V
Ana 1 = 0 ... 4 ... 20 mA

Lösung:

```
H 1='1,iff,AnaModSel 1 =1,AnaSSel 1 =0,AnaFACTOR 1 =20,AnaOFFSET 1 =0'
H 2='RegU1N-,80,<,if,RegU1N-,20,/,Ana 1=.,else,RegU1N-,2.5,/,28,-,Ana 1=.'
```

2. Beispiel mit universeller Skalierung:

z.B.:

UoN: 0...20 ...100 V => Ana1: 0...18 ...20 mA

allgemein:

UoN: 0...a00...a01 V => Ana1: 0...b00...b01 mA

Lösung (für Beispiel):

```
H 1='a00=20, a01=100, b00=18, b01=20'
H 2='b00,a00,/,a10='
H 3='b01,b00,-,a01,a00,-,/,a11='
H 4='a01,b00,*,a00,b01,*,-,a01,a00,-,/,b11='
H 5='RegU1N-, a00,<,if,RegU1N-, a10,*,Ana 1=,else,RegU1N-, a11,*,b11,+,Ana 1='
```

Variablen:

a00 Merker für Knick in der Eingangsspannung
 a01 Merker für Endwert der Eingangsspannung
 b00 Merker für Knick im Ausgangssignal
 b01 Merker für Endwert vom Ausgangssignal
 a10 Merker für Verhältnis am Knick
 a11 Merker für Verhältnis oberhalb vom Knick
 b11 Merker für berechneten Offset

Analogeingang allgemein

Analogeingang konfigurieren:

AnaModSel =	AnaSSel =
0: -10 V ... 0 ... 10 V	0: + / -
1: -20 mA ... 0 ... 20 mA	1: +
2: -5 mA ... 0 ... 5 mA	2: -
3: S0 ??	
4: 4 mA ... 20 mA	

Analogeingang skalieren:

Ana = AnaN * AnaFactor + AnaOffset

wobei für AnaModSel=1: AnaN: -1 ... 1 => Eingang: -20 mA ... 20 mA

Aufgabe:

Der Analogeingang Ana2 (0..20mA) soll benutzt werden um einen Meßwandlereingang (0..100V) zu simulieren.

Lösung:

```
# konfigurieren von Ana 2
H 1='1,iff,AnaModSel 2 =1,AnaSSel 2 =0,AnaFACTOR 2 =20,AnaOFFSET 2 =0'
# Meßwandlereingang für U1 wird ersetzt durch RegMIVAL 1
H 2='1,iff,RegMiMap 1=-1'
# skalieren von Ana 2 / Meßwandlereingang wird per RegMIVAL gesetzt
H 3='Ana- 2,5,*,RegMiVal 1='
# Anzeige des Analogeingangswertes im Display
H 4='Ana- 2,meld "AnaEin 2 = %! mA"'
```

Analogeingang für Stufenstellung

Aufgabe:

Der Analogeingang Ana1 (z.B. 0..20mA) soll benutzt werden um die Stufenstellung (z.B. 1..19) einzulesen.

Lösung:

```
# Ana1 0..20 mA => Stufenstellung 1..19
H 1='1,iff,AnaModSel 1 =1,AnaSSel 1 =1'
H 2='AnaN- 1,0.0555,/,1,+,INTR,RegStufe ='
```

oder

```
# Ana1 0..20 mA => Stufenstellung 1..19
H 1='1,iff,AnaModSel 1 =1,AnaSSel 1 =1, AnaFACTOR 1 =20,AnaOFFSET 1 =0'
H 2='Ana- 1,0.9,*,1,+,INTR,RegStufe =.'
```

Analogeingang für Stufenstellung ab FW2.00

Aufgabe:

Der Analogeingang Ana1 (z.B. 4..20mA, normiert 0.2 .. 1) soll benutzt werden um die Stufenstellung (z.B. 1..15) einzulesen.

Lösung:

```
# Grundeinstellungen für die Stufenstellungserfassung über analogen Eingang
H 1='1,iff,AnaFU 1=70,AnaUNIT 1=TAP,AnaParmSel 1=2, AnaLimSel 1=0'
# min. Stufe=1, bei 0.2(4mA), max. Stufe =15, bei 1(20mA)
H 2='1,iff, AnaSCALX0 1=1,AnaSCALY0 1=0.2,AnaSCALX2 1=15,AnaSCALY2 1=1,'
```

- Binäre Eingänge die mit BCD-Code oder BIN-Code parametrier sind sollten auf AUS oder eine andere gewünschte Funktion umparametriert werden.
- Stufenstellungsanzeige im Menü (SETUP5 -> Funktionen 1) freischalten.

Analogeingang für Stufenstellung mit Anwendermenü

Aufgabe

Über einen einstellbaren Offset und eine einstellbaren mA-Bandbreite pro Stufe soll über den Analogeingang AE1 die Stufenstellung berechnet werden.

Lösung

```
## application-menu
H 2='menuappn 2="Trans.Cali",menuapp 2=102'
H 3='menuappn 6="offset",menuapp 6=106'
H 4='menuappn 7="dl step",menuapp 7=107'
##### analog input to tap-position
## set default-values
H 24='1,iff,a0=0,a1=1'
##### x...y mA => tap-position #####
H 25='1,iff,AnaModSel 1 =1,AnaSSel 1 =0,AnaFACTOR 1 =20,AnaOFFSET 1 =0'
H 26='Ana- 1,a 0,-,a 1,/,INTR,RegStufe =.'
# summer-/winter-time
H 31='SOWI, IF, ZEIT-, +, ZEIT=.'
##### analog input to tap-position
P 2='tt >'
P 6='menuedit a 0 ="Off [-20..20]" -20 20'
P 7='menuedit a 1 ="dl [0,1..10]" 0.1 10'
# start the background-program
hbreak-
##### variable #####
#a0 offset: current in tap-position 1 (default = 0 mA)
#a1 dl: current per step (default = 1 mA)
```

Analogeingang in Temperatur umrechnen und am Display anzeigen

Aufgabe

Der Analogeingang AE1 4..20mA soll in eine Temperatur von 0..150°C umgerechnet werden.
Über das Anwendermenü soll dann die berechnete Temperatur für 10s ins Display eingeblendet werden.

Lösung

```
# applicationmenu
H 2='1,iff,menuappn 1="temp. 1",menuapp 1=1'
# analog input 1 (4..20mA --> 0..150°C)
H 6='1,iff,AnaModSel 1 =1,AnaSSel 1 =1,AnaFACTOR 1 =1,AnaOFFSET 1 =0'
H 7='AnaN- 1,0.2,-,187.5,*,intr,a11=.'
# applicationmenu
P 1='a11,meld "temp. 1: %! øC" 10'
##### variable #####
# a11: value for Temperature 1
```