

NEWEL 3



SPEICHERPROGRAMMIERBARE- STEUERUNG FUNKTIONEN (SPS)

HILFE

Digitel behält sich das Recht vor, die in diesem Dokument enthaltenen Informationen ohne vorherige Ankündigung zu ändern.

Nichtvertragliches Dokument

Digitel SA

Alle Rechte vorbehalten

Inhaltsverzeichnis

1.	BESCHREIBUNG DER PLC-FUNKTIONEN (PLC)	4
2.	KOMPATIBLE HARD- UND SOFTWARE	4
3.	ANWENDUNGSBEISPIEL	4
4.	EINGANGS- UND AUSGANGSWERT	6
4.1.	EINGABEWERTE	6
4.2.	AUSGANGSWERTE	6
5.	WIE MAN EINE SPS-FUNKTION ERSTELLT	7
5.1.	ÄNDERUNG DER PARAMETER DURCH EINEN BENUTZER	15
6.	REFERENZEN DER PROGRAMMIERSPRACHE FÜR PLC-FUNKTIONEN	17
6.1.	STRUKTUREN	17
6.1.1.	Kommentare	17
6.1.2.	Zuweisung von Variablen	17
6.1.3.	Die Operatoren	17
6.1.4.	IF..THEN..ELSE..	19
6.2.	VORDEFINIERTE FUNKTIONEN	20
6.2.1.	digAlarmSet	20
6.2.2.	digAlarmGetState	21
6.2.3.	digMessageSend	21
6.2.4.	digTrace	22
6.2.1.	digSetpointShift	22
6.2.2.	digSetpointSetTR	23
6.2.3.	digSetpointSetTR_MP	23
6.3.	SYSTEMVARIABLEN, DIE IN SPS-FUNKTIONEN VERWENDET WERDEN KÖNNEN.	24
6.4.	SYSTEMKONSTANTE VERWENDBAR IN SPS-FUNKTIONEN	25
6.4.1.	CONTROLLER_OUTPUT	25
6.4.1.	CONTROLLER_SETPOINT	26

7.	HILFE KNOPF	27
8.	IMPORTIERUNG ET DUPLIZIERUNG VON FUNKTIONEN	28
9.	EINE FUNKTION SPEICHERN UND WIEDERHERSTELLEN	29
10.	BEISPIELE	31
10.1.	MASCHINENRAUMBELÜFTUNG	31
10.2.	Thermostat	32
10.3.	ZYKLISCHER TIMER	33
10.4.	VERWENDUNG VON VORDEFINIERTEN TIMERN IN DER ZENTRALEINHEIT	34

1. BESCHREIBUNG DER PLC-FUNKTIONEN (PLC)

Mit den Funktionen "Speicherprogrammierbare Steuerung" können bestimmte Aufgaben ähnlich wie bei einer SPS (Speicherprogrammierbare Steuerung) automatisiert werden. Diese Funktionen sind in TelesWin mit einer leicht verständlichen und schreibbaren Sprache programmiert und werden in der DC58-Zentraleinheit gespeichert, die sie alle zwei Sekunden ausgeführt. Mit diesen Funktionen ist es einfach, das System an Ihre Bedürfnisse anzupassen.

Die Digital-Lösung ermöglicht es, alle an das Netzwerk angeschlossenen Ein- und Ausgänge zu nutzen, unabhängig von ihrer Nutzung (Regelung von Kühlstationen, Kompressoren usw.). Im Vergleich dazu können herkömmliche SPSen nur auf ihre eigenen Ein- und Ausgänge und nicht auf die anderer Netzwerkelemente zugreifen.

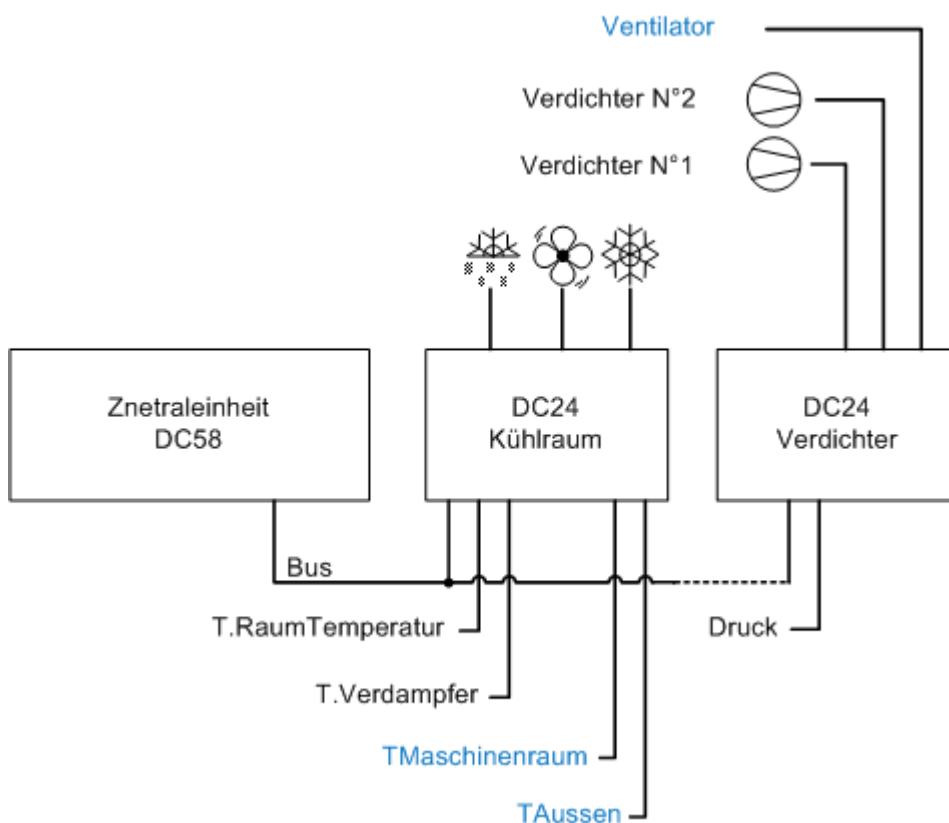
SPS-Funktionen ermöglichen es, auch nach der Inbetriebnahme und im laufenden Betrieb der Anlage neue Funktionalitäten zu schaffen oder bestehende zu erweitern, ohne Infrastrukturen wie die Verkabelung zu verändern. Eine umfangreiche Bibliothek von vorprogrammierten Funktionen ermöglicht es Ihnen, verschiedene und komplexe Aufgaben auszuführen.

2. KOMPATIBLE HARD- UND SOFTWARE

- TelesWin mit der Softwareversion 22.73-19.45-1 oder höher
- DC58 Zentraleinheit mit Softwareversion 19451 oder höher.
- DC24 D/DE/E/EE Regler mit Softwareversion 19451 oder höher.

3. ANWENDUNGSBEISPIEL

- Steuerung der Maschinenraumlüftung über 2 freie Kühlraumregler-Eingänge und einen freien Verdichterregler-Ausgang.



Konfiguration der SPS Funktionen

Liste der Variablen

Variablenname	Beschreibung	Typ	Wert
TMaschinenraum	Maschinenraumtemperatur	E/A eines Moduls	Ch. Froide
TAussen	Aussentemperatur	E/A eines Moduls	Ch. Froide
Sollwert	Sollwert	Parameter	25.0
Delta	Delta	Parameter	1.0
Ventilator	Ventilatosteuerung	E/A eines Moduls	Compresseurs

Code

```

IF  {{TMaschinenraum}} > ( {{Sollwert}}+{{Delta}} )  AND  {{TAussen}} < {{TMaschinenraum}}  THEN
    {{Ventilator}} = 1
ELSE
    IF  {{TMaschinenraum}} < {{Sollwert}}  OR  {{TAussen}} > {{TMaschinenraum}}  THEN
        {{Ventilator}} = 0
    END
END

```

Abbrechen OK Aide ?

4. EINGANGS- UND AUSGANGSWERT

- SPS-Funktionen ermöglichen es, Aktionen an den Ausgängen (S) entsprechend den Eingangswerten (E) durchzuführen. Es können alle Netzwerk-I/Os verwendet werden.

4.1. EINGABEWERTE

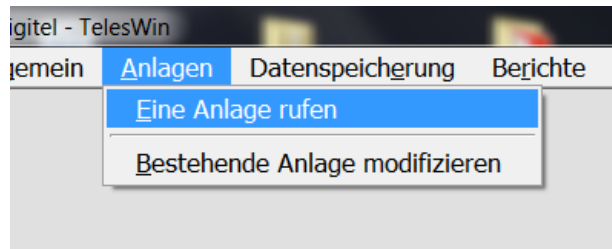
- Messung physikalischer Parameter (Temperatur, Druck, Feuchtigkeit, Helligkeit).
- In der Zentraleinheit definierte Timer.
- Feste Parameter, die von den Programmierern festgelegt werden.
- Einstellungen, die vom Benutzer geändert werden können.

4.2. AUSGANGSWERTE

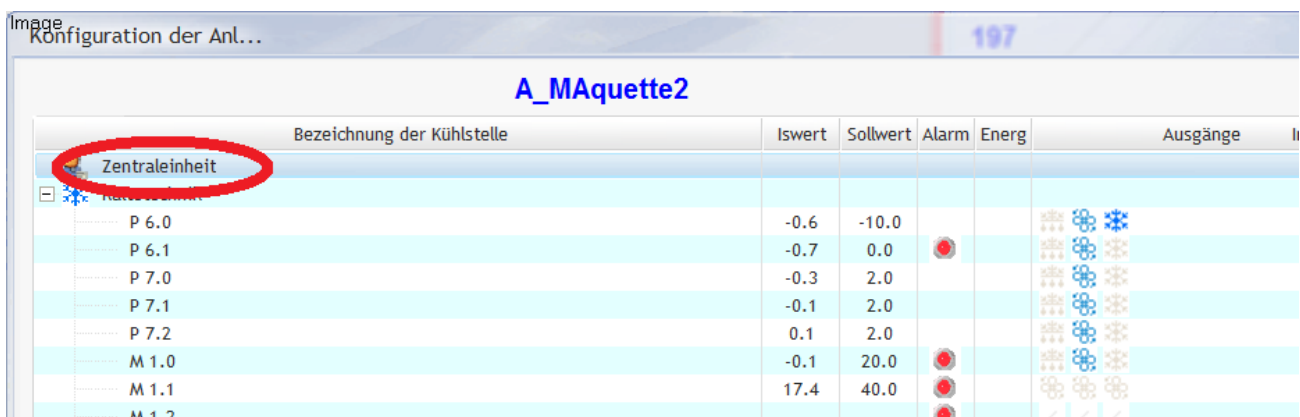
- Aktion an den digitalen Ausgängen von Satelliten (Ein-/Ausschalten von Ventilatoren, Kompressoren oder Relais).
- Steuerung von Analogausgängen.
- Aktivierung von in der Zentraleinheit definierten Alarmen.
- Senden von SMS.
- Senden von E-Mails.
- Anzeige von Meldungen und Werten in einer Konsole auf TelesWin.

5. WIE MAN EINE SPS-FUNKTION ERSTELLT

- Nur die Person mit der Berechtigung "Konfiguration" für die Installation (siehe Zentraleinheit "Zutrittskontrolle") kann SPS-Funktionen anlegen und ändern. Außerdem muss ein Dongle mit der Option "Systemkonfiguration" aktiviert sein.
- Öffnen Sie das Fenster **Konfiguration der Anlage** (Menü **Anlage / Eine Anlage rufen**).



- Doppelklicken Sie auf **Zentraleinheit**.

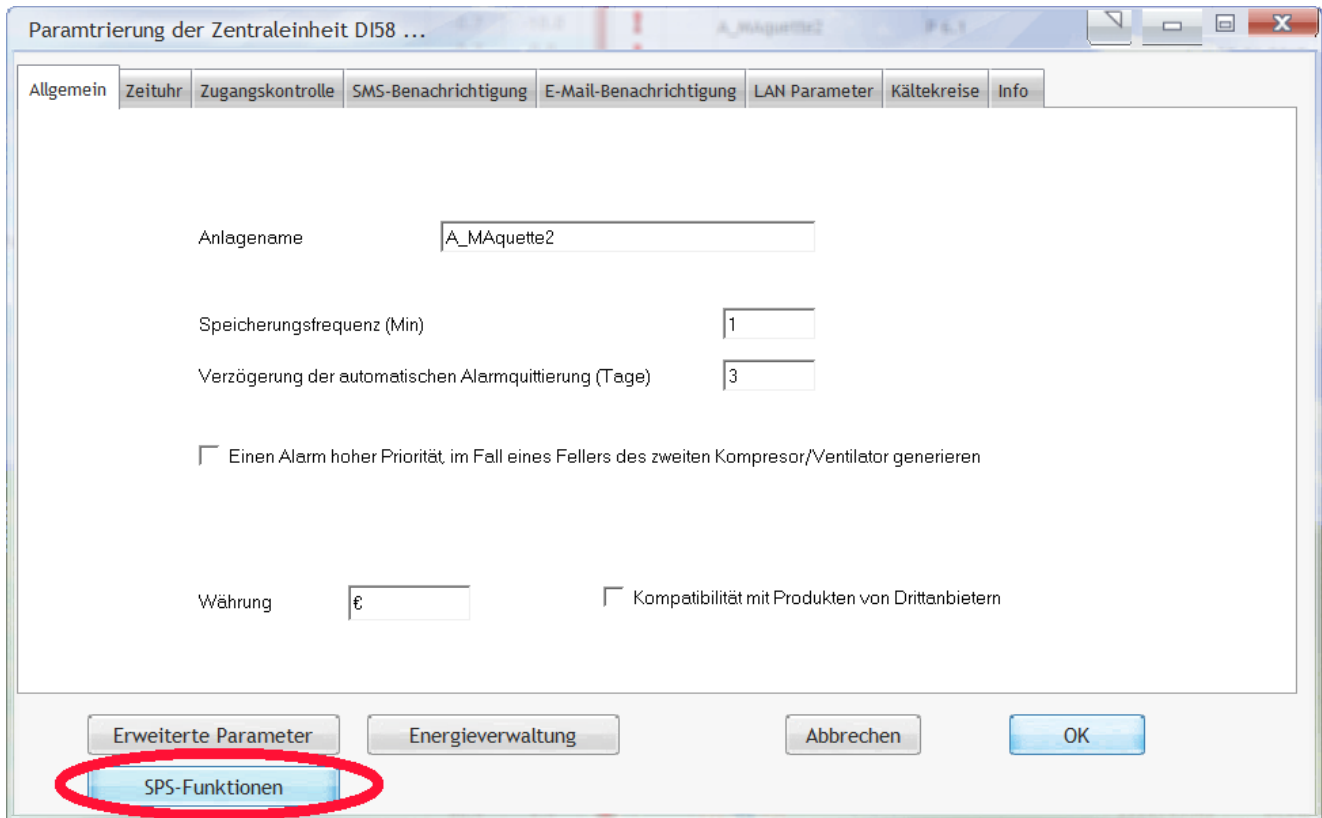


Konfiguration der Anlage

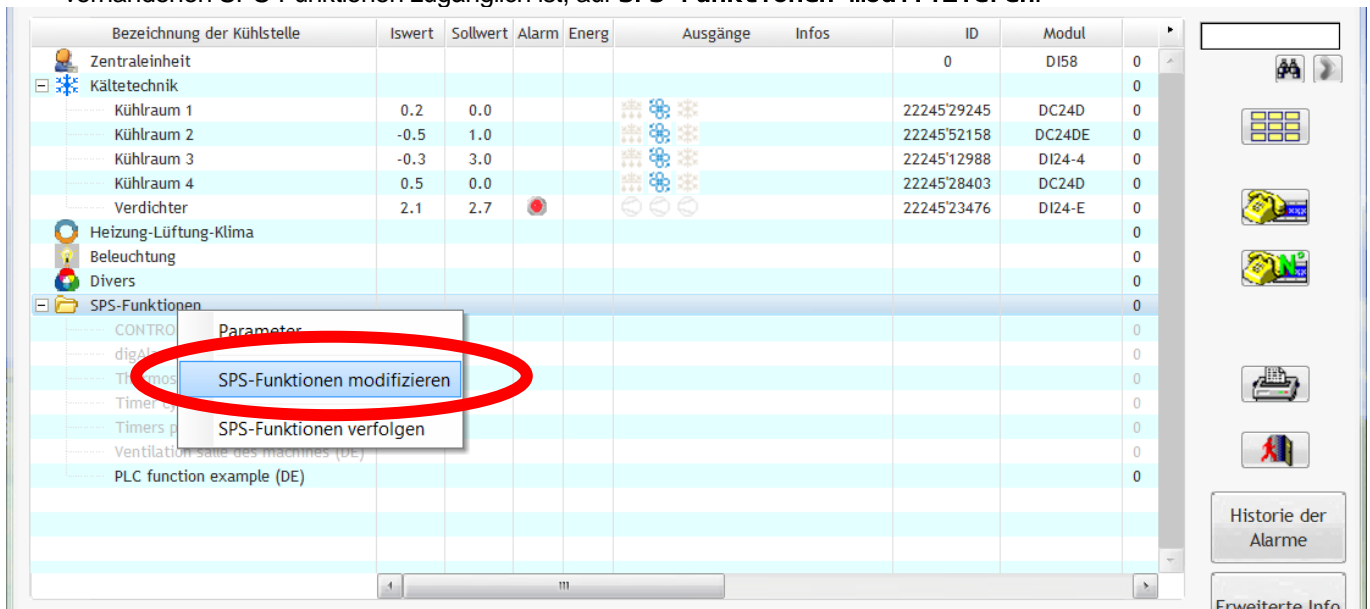
A_Maquette2

Bezeichnung der Kühlstelle	Iswert	Sollwert	Alarm	Energ	Ausgänge
Zentraleinheit					
P 6.0	-0.6	-10.0			
P 6.1	-0.7	0.0	●		
P 7.0	-0.3	2.0			
P 7.1	-0.1	2.0			
P 7.2	0.1	2.0			
M 1.0	-0.1	20.0	●		
M 1.1	17.4	40.0	●		
M 1.2			●		

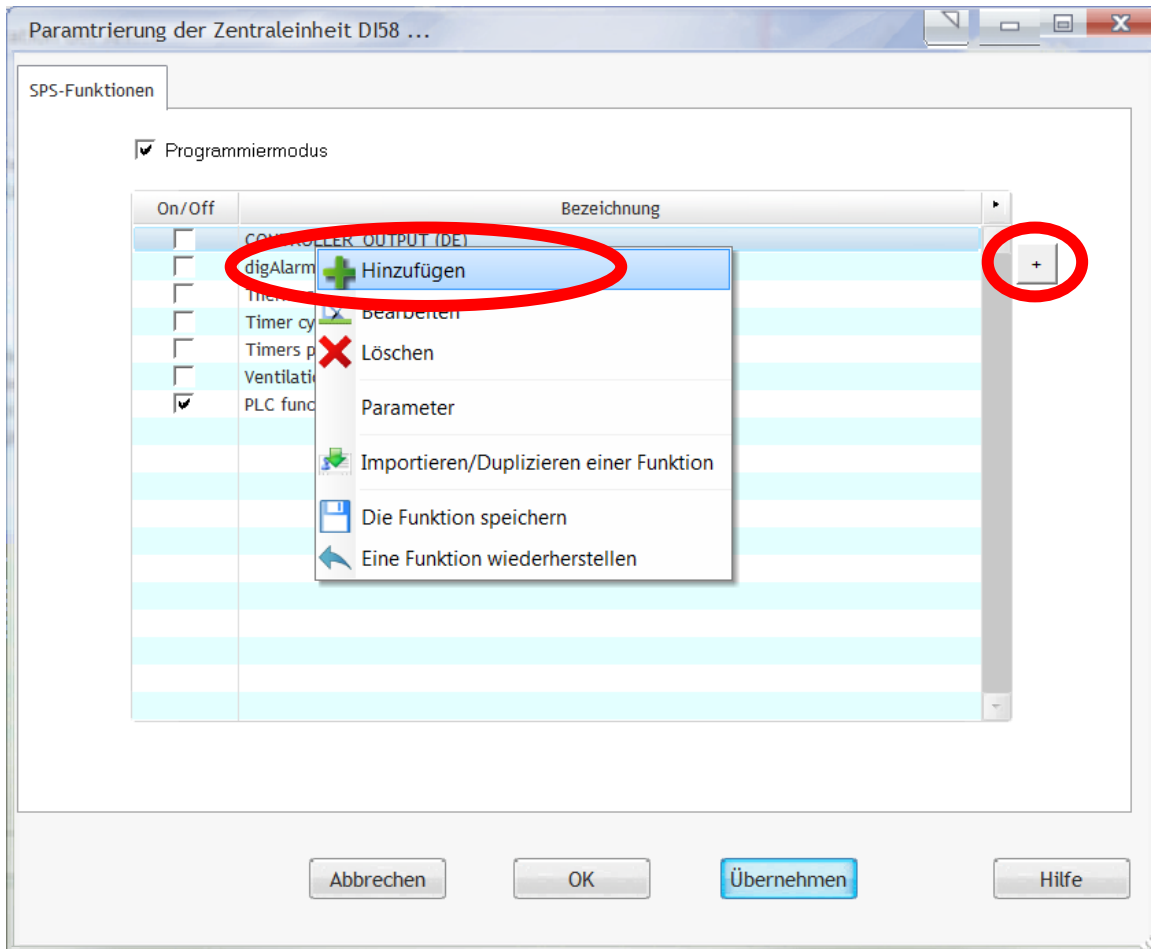
- Das Fenster **Paramtrierung der Zentraleinheit** öffnet sich.
- Klicken Sie auf den Knopf **SPS-Funktionen**.



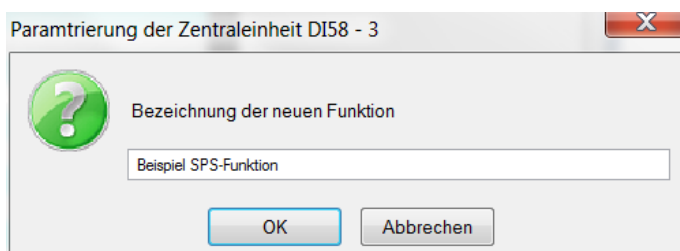
- Oder klicken Sie im Kontextmenü, das durch einen Rechtsklick auf **SPS-Funktionen** oder auf eine der vorhandenen SPS-Funktionen zugänglich ist, auf **SPS-Funktionen modifizieren**.



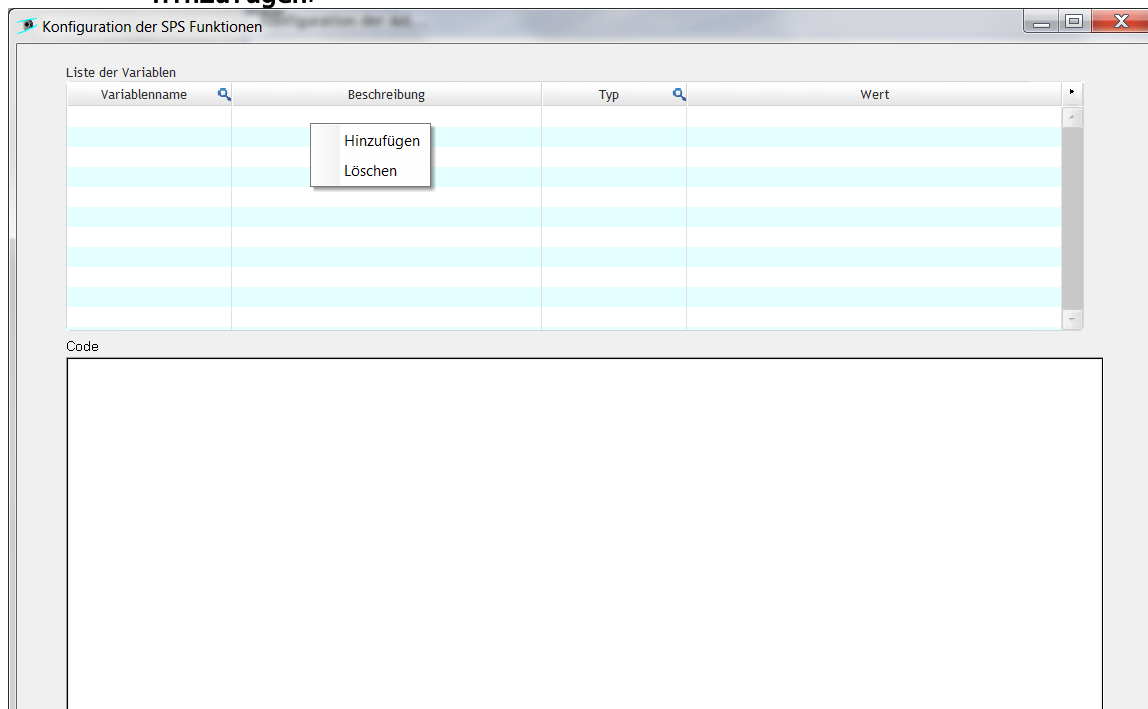
- Das Fenster **Paramtrierung der Zentraleinheit DI58** öffnet sich.
- - Klicken Sie auf die Schaltfläche +, um eine Funktion hinzuzufügen. Es ist auch möglich, die Funktion **Hinzufügen** des Kontextmenüs zu verwenden, das durch einen Rechtsklick in der Tabelle zugänglich ist.



- Es öffnet sich ein Popup-Fenster.
- Geben Sie den Namen der neuen SPS-Funktion ein. Zum Beispiel **Beispiel SPS-Funktion**.
- Klicken Sie auf OK.



- Das Fenster **Konfiguration der SPS Funktionen** öffnet sich.
- In diesem Fenster können Sie eine SPS-Funktion erstellen.
- Die Tabelle **Liste der Variablen** ermöglicht es Ihnen, die Variablen zu deklarieren, die von der SPS-Funktion verwendet werden.
- Das Feld **Code** wird zur Eingabe des Codes verwendet.
- Klicken Sie mit der rechten Maustaste in die Tabelle **Liste der Variablen** und klicken Sie auf **Hinzufügen**.



- Füllen Sie die Tabelle **Liste der Variablen** so aus, dass sie dem untenstehenden Bild entspricht.
- Um das Feld **Wert** auszufüllen, ist es notwendig, auf die Schaltfläche rechts neben dem Feld zu klicken (siehe Bild unten).
- Füllen Sie auch das Feld **Code** aus. Sie können das folgende Beispiel kopieren und einfügen.

```
// Demonstrationsprogramm, das den Wert eines Parameters anzeigt,  
// der von allen Benutzern geändert werden kann.
```

```
digTrace ( {{ "PARAMETER 1 = " }} + {{ Parameter1 }} )
```

PLC function example (DE)

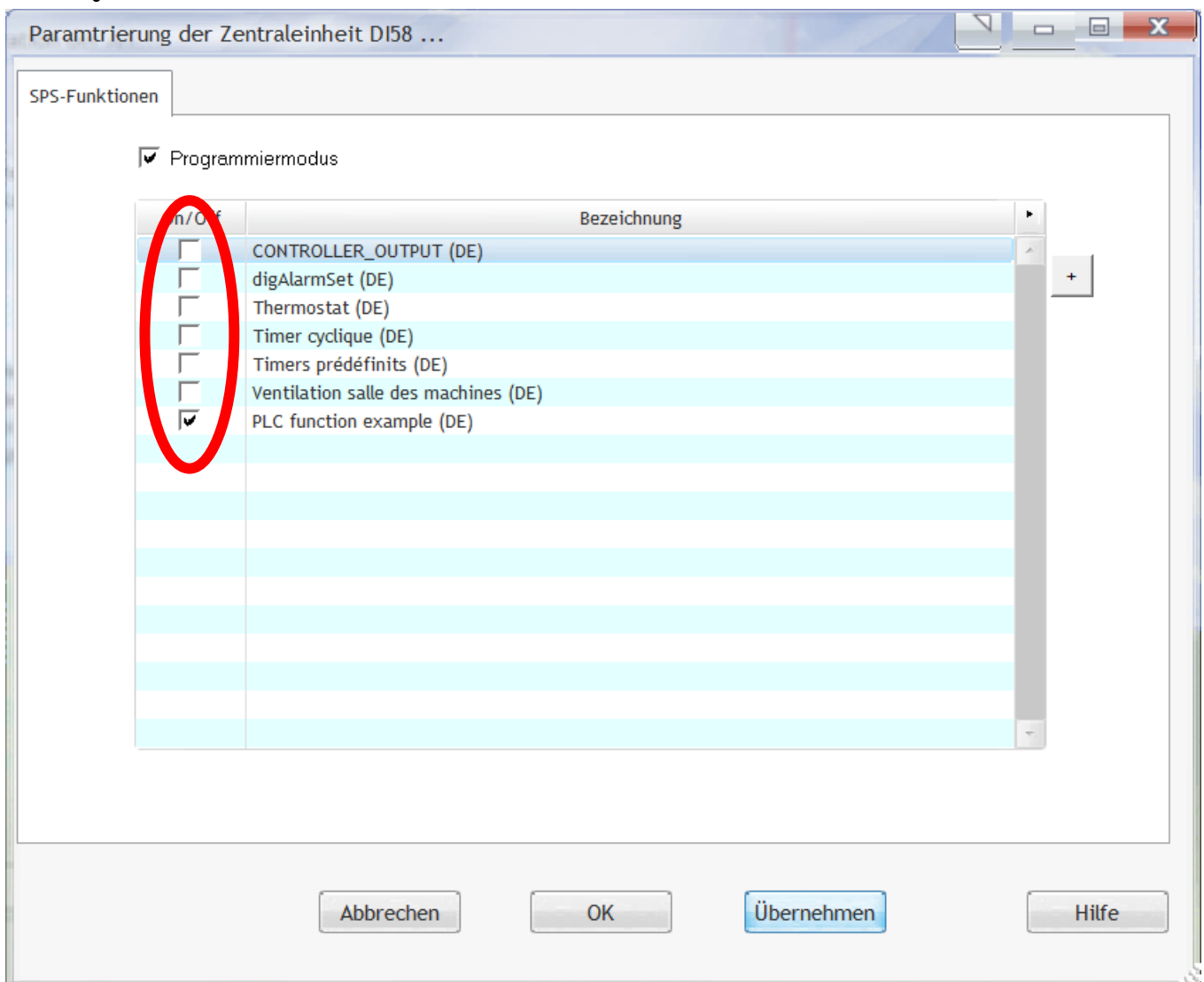
Variablenname	Beschreibung	Typ	Wert
Parameter1	Parameter der von allen geändert werden kann	Parameter	123456

Code

```
// Demonstrationsprogramm, das den Wert eines Parameters anzeigt,  
// der von allen Benutzern geändert werden kann.  
  
digTrace ( {{ "PARAMETER 1 = " }} + {{ Parameter1 }} )
```

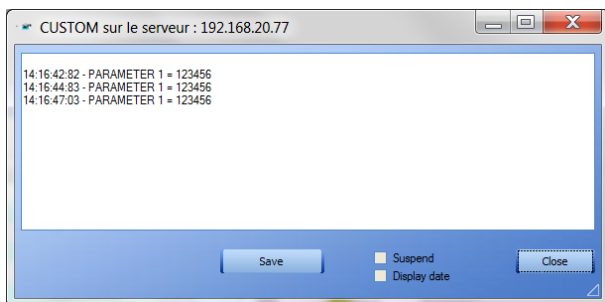
Abbrechen OK Übernehmen Hilfe

- Nachdem Sie die Parameter und den Code eingegeben haben, klicken Sie auf OK.
- Klicken Sie auf das Kontrollkästchen **ON/OFF** links neben der gerade erstellten Funktion, um sie auszuführen. Die Ausführungszeit beträgt ca. zwei Sekunden.



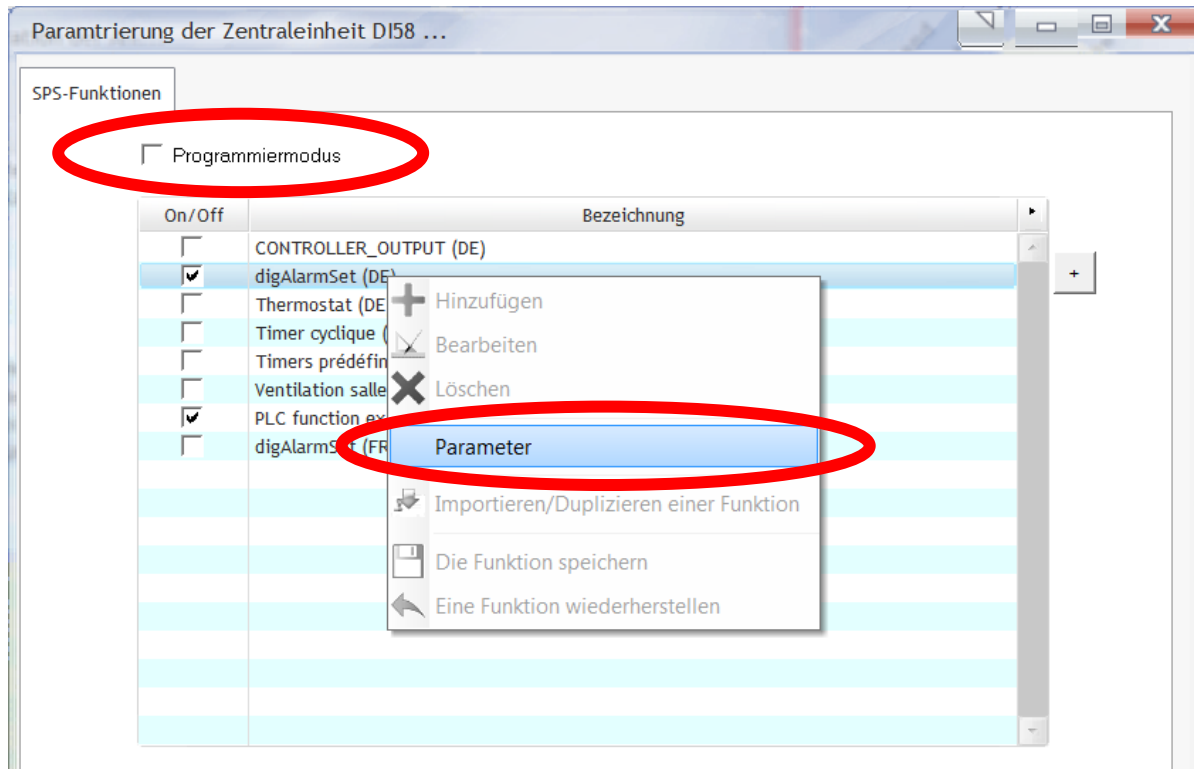
- Um den Fortschritt der Programmausführung zu verfolgen, wählen Sie im Kontextmenü der SPS-Funktionen "SPS-Funktionen verfolgen". Ein Fenster "Trace" öffnet sich und zeigt den in der digTrace-Funktion eingegebenen Text an.
- Wir sehen, dass die SPS-Funktion Texte anzeigt, die sich im Laufe der Zeit im Moment nicht ändern.
- Wir werden dann weitere interessante Beispiele sehen, mit Messwerten am Eingang und Relais am Ausgang.

Bezeichnung der Kühlstelle	Iswert	Sollwert	Alarm	Energ	Ausgänge	Infos	ID	Modul	
Zentraleinheit							0	DI58	0
Kältetechnik									0
Kühlraum 1	0.2	0.0			⊗ ⊗ ⊗ ⊗		22245'29245	DC24D	0
Kühlraum 2	-0.5	1.0			⊗ ⊗ ⊗ ⊗		22245'52158	DC24DE	0
Kühlraum 3	-0.2	3.0			⊗ ⊗ ⊗ ⊗		22245'12988	DI24-4	0
Kühlraum 4	0.5	0.0			⊗ ⊗ ⊗ ⊗		22245'28403	DC24D	0
Verdichter	2.1	2.7	●		⊗ ⊗ ⊗ ⊗		22245'23476	DI24-E	0
Heizung-Lüftung-Klima									0
Beleuchtung									0
Divers									0
SPS-Funktionen									0
CONTROLLER_OUTPUT (DE)									0
digAlarmSet (0
Thermostat (0
Timer cycliqu									0
Timers prédé									0
Ventilatio									0
PLC functio									0

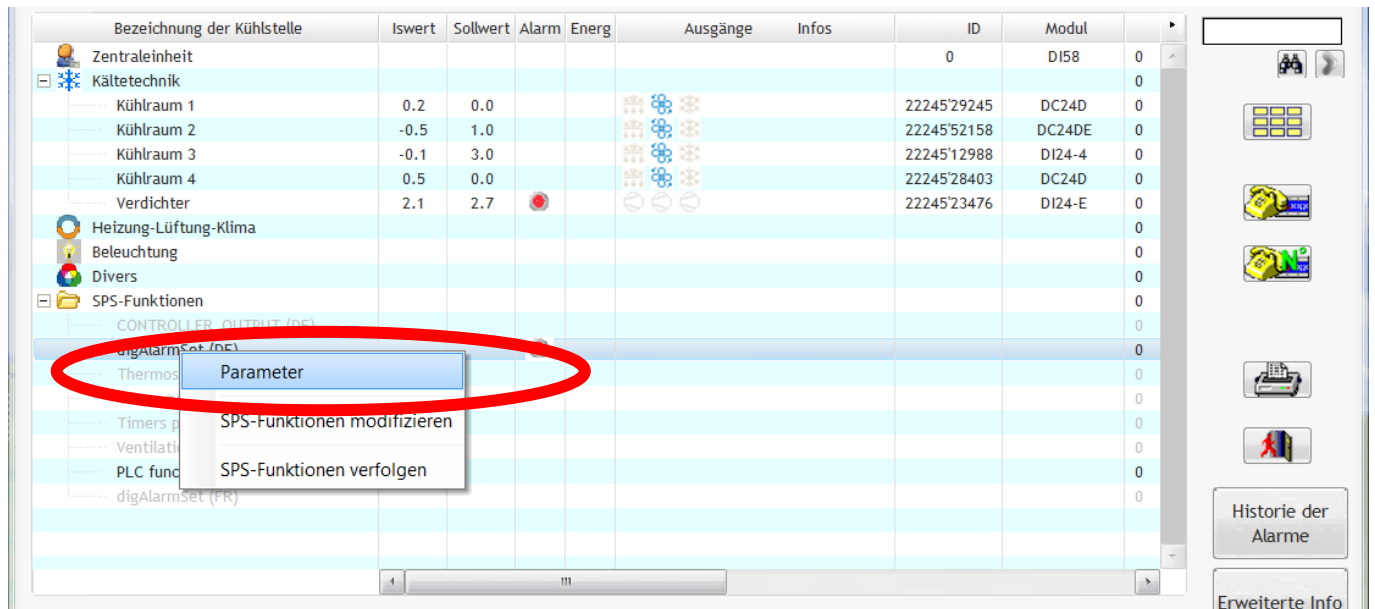


5.1. ÄNDERUNG DER PARAMETER DURCH EINEN BENUTZER

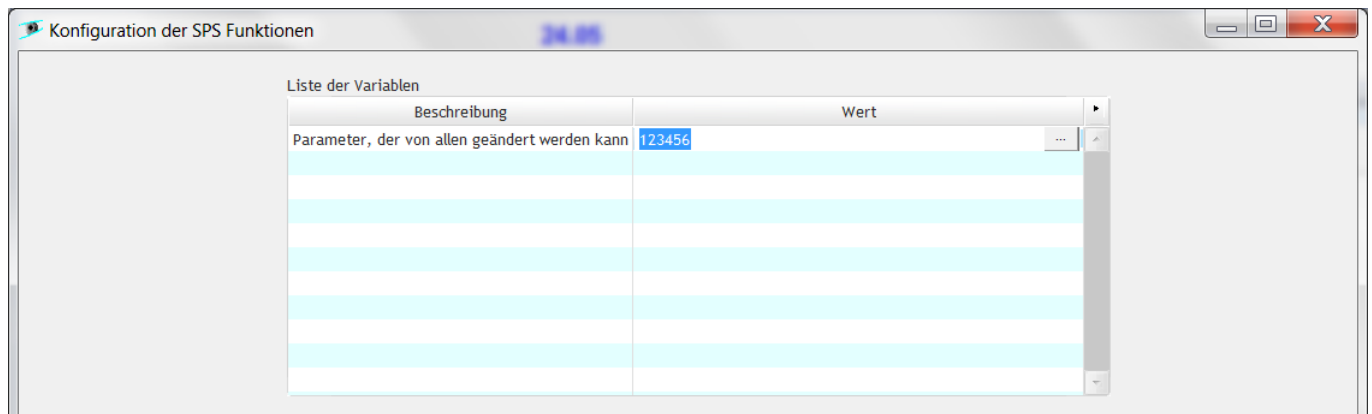
Die mit dem Parametertyp **Parameter** deklarierten Variablen können von einem Benutzer ohne Programmierrechte geändert werden.



Es ist auch möglich, die Parameter aus der globalen Sicht der Installation zu ändern. Klicken Sie einfach auf die Option Einstellungen des Kontextmenüs oder doppelklicken Sie auf die Funktion, deren Einstellungen Sie ändern möchten.



Bei Benutzern ohne Programmierrechte ist das Kontrollkästchen **Programmiermodus** immer deaktiviert. Wenn sie mit der rechten Maustaste klicken, um das Kontextmenü anzuzeigen, haben sie nur Zugriff auf die Option **Parameter**, mit der sie die Einstellungen ändern können.



6. REFERENZEN DER PROGRAMMIERSPRACHE FÜR PLC-FUNKTIONEN

Die von uns verwendete Programmiersprache basiert auf der von PC Soft entwickelten Sprache Windev Mobile. Dieses Handbuch beschreibt nur die grundlegenden Operatoren, die am häufigsten verwendet werden. Sie reichen aus, um einfache Funktionen zu erstellen. Um erweiterte Funktionen zu erstellen, empfehlen wir Ihnen, das Handbuch Windev Mobile zu lesen, das eine detaillierte Beschreibung aller Bediener und Funktionen dieser Sprache enthält.

Die Namen der Variablen der SPS-Funktionen können groß- und kleingeschrieben werden, d. h. bei der Verwendung einer Variablen muss Groß- und Kleinschreibung beachtet werden.

Variablennamen sollten nur Buchstaben, Zahlen und das unterstrichene Zeichen (_) enthalten. Sie dürfen keine Leerzeichen enthalten.

6.1. STRUKTUREN

6.1.1. Kommentare

Kommentare können dem Code hinzugefügt werden. Sie beginnen mit einem doppelten Schrägstrich `//` und enden am Ende der Zeile. Kommentare sind nützlich für die Dokumentation des Codes und werden nicht ausgeführt.

6.1.2. Zuweisung von Variablen

Variablen müssen in der Tabelle **Liste der Variablen** deklariert werden. Sie müssen von zwei Klammern `{...}` umgeben sein, wenn sie im Code verwendet werden, sei es beim Lesen oder Schreiben. Die möglichen Arten von Variablen sind:

Modul-I/O: Zuweisung des Wertes eines Ein- oder Ausgangs (I/O) an eine Variable im SPS-Code.

Interne Variable: Deklaration einer neuen Variablen, die im SPS-Code definiert und zugänglich ist.

Parameter: Um einen Parameterwert zuzuweisen, der von einem Benutzer geändert werden kann, der keine Programmierrechte hat.

Alarm: Um einen Alarm zu erzeugen.

Timer: So verwenden Sie den Rückgabewert eines bereits in der Zentraleinheit erstellten Timers (**Paramtrierung der Zentraleinheit / Erweiterte Parameter / Schaltuhren**)

6.1.3. Die Operatoren

VERKETTUNG VON ZEICHENKETTEN

Es ist möglich, Zeichenketten zwischen ihnen zu verketteten und auch Zeichenketten und Variablen zu verketteten. Das Ergebnis kann dann zur Anzeige von informativen Nachrichten verwendet werden. Der Kettenverkettungsoperator ist das `+` Zeichen.

BEISPIEL

Der folgende Code zeigt im Trace-Fenster die Zeichenkette `Temp = xx°C` an, wobei xx der Temperaturwert ist, z.B. `Temp = -18°C`. Die `Temp1`-Sonde muss in der Liste der Variablen definiert sein.

Konstante Zeichenketten müssen von doppelten geschweiften Klammern und Anführungszeichen umgeben sein `{{"Temp = "}}`, während Variablen nur geschweifte Klammern benötigen, nicht aber Anführungszeichen `{{Temp1}}`. Die Klammern können optional durch ein Leerzeichen gefolgt und vorangestellt werden, können aber nicht durch ein Leerzeichen getrennt werden.

Die Funktionsparameter sind von Klammern umgeben, die optional mit einem Leerzeichen vorangestellt und gefolgt werden können.

```
digTrace({{"Temp = "}} + {{Temp1}} + {"°C"})
```

ADDITIONEN

Die Addition verwendet auch das + Zeichen, wie z.B. die Verkettung von Ketten. Wenn Sie Addition und Verkettung in derselben Funktion verwenden wollen, müssen Sie die Operationen auf zwei getrennten Zeilen durchführen.

```
{{Val}} = {{Temp1}} + 1.5 // Das + Zeichen macht eine Addition.  
digTrace ( {"Temp1 = "}} + { Temp1 } ) // Das + Zeichen führt eine Verkettung durch.  
digTrace ( {"Val = "}} + { Val } ) // Das + Zeichen führt eine Verkettung durch.
```

SUBTRAKTION

```
{{Val}} = {{Temp1}} - 1.5
```

MULTIPLIKATION

```
{{Val}} = {{Temp1}} * 1.5
```

DIVISION

```
{{Val}} = {{Temp1}} / 1.5
```

ABSOLUTER WERT

```
{{Val}} = Abs ( {{Temp1}} )
```

VERGLEICHOPERATOREN

Vergleichsoperatoren geben boolesche Werte zurück, d.h. True oder False.

```
{{Val}} = {{Temp1}} > -20 // Größer  
{{Val}} = {{Temp1}} >= -20 // Größer oder gleichwertig  
  
{{Val}} = {{Temp1}} < -20 // Kleiner  
{{Val}} = {{Temp1}} <= -20 // Kleiner oder gleichwertig  
  
{{Val}} = {{Temp1}} = -20 // Gleichwertig  
{{Val}} = {{Temp1}} <> -20 // Anders
```

LOGISCHE OPERATOREN

```
{{Val}} = ( {{Temp1}} > -20 ) AND ( {{Temp2}} > -20 ) // UND logisch  
{{Val}} = ( {{Temp1}} > -20 ) OR ( {{Temp2}} > -20 ) // ODER logisch  
{{Val}} = NOT ( {{Temp2}} > -20 ) // NEGATION logisch
```

LOGISCHE WERTE

```
{{Val}} = True // Alle Zahlenwerte abweichend von 0  
// werden auch als wahr bewertet.  
{{Val}} = False // Der Zahlenwert 0 wird ebenfalls als False ausgewertet.
```

MODULO

Das Modulo ist der Rest der gesamten Division. Zum Beispiel $7 \text{ modulo } 3 = 1$, weil $7 / 3 = 2$ bleibt 1, oder mit anderen Worten $3 \times 2 + 1 = 7$. Mit dem Modulo können auf einfache Weise getaktete Signale erzeugt werden. Zum Beispiel, wenn wir ein Signal mit einer 10-minütigen Periode erzeugen wollen, das bei 1 für 2 Minuten und damit bei 0 für 8 Minuten liegt:

```
{{Val}} = modulo (minuteFrom2000, 10) < 2
```

6.1.4. IF..THEN..ELSE..

Die bedingte Anweisung **IF** ermöglicht es Ihnen, eine Aktion entsprechend einer Bedingung auszuführen.

EXEMPLE

```
IF {{Temp1}} > 0 THEN
  // Aktion 1
ELSE IF {{Temp1}} > -20 THEN
  // Aktion 2
ELSE
  // Aktion 3
END
```

6.2. VORDEFINIERTE FUNKTIONEN

6.2.1. digAlarmSet

Löst einen Alarm aus, wenn eine Bedingung für eine bestimmte Zeit erfüllt ist. Die Alarmmeldung muss im Feld **Beschreibung** der Variablenliste konfiguriert werden.

SYNTAX

```
digAlarmSet(Name, Bedingung, Verzögerung)
```

<Name> : Zeichenkette

Alarmreferenz (Name der Variablen in der Tabelle **Liste der Variablen**).

< Bedingung > : Boolean

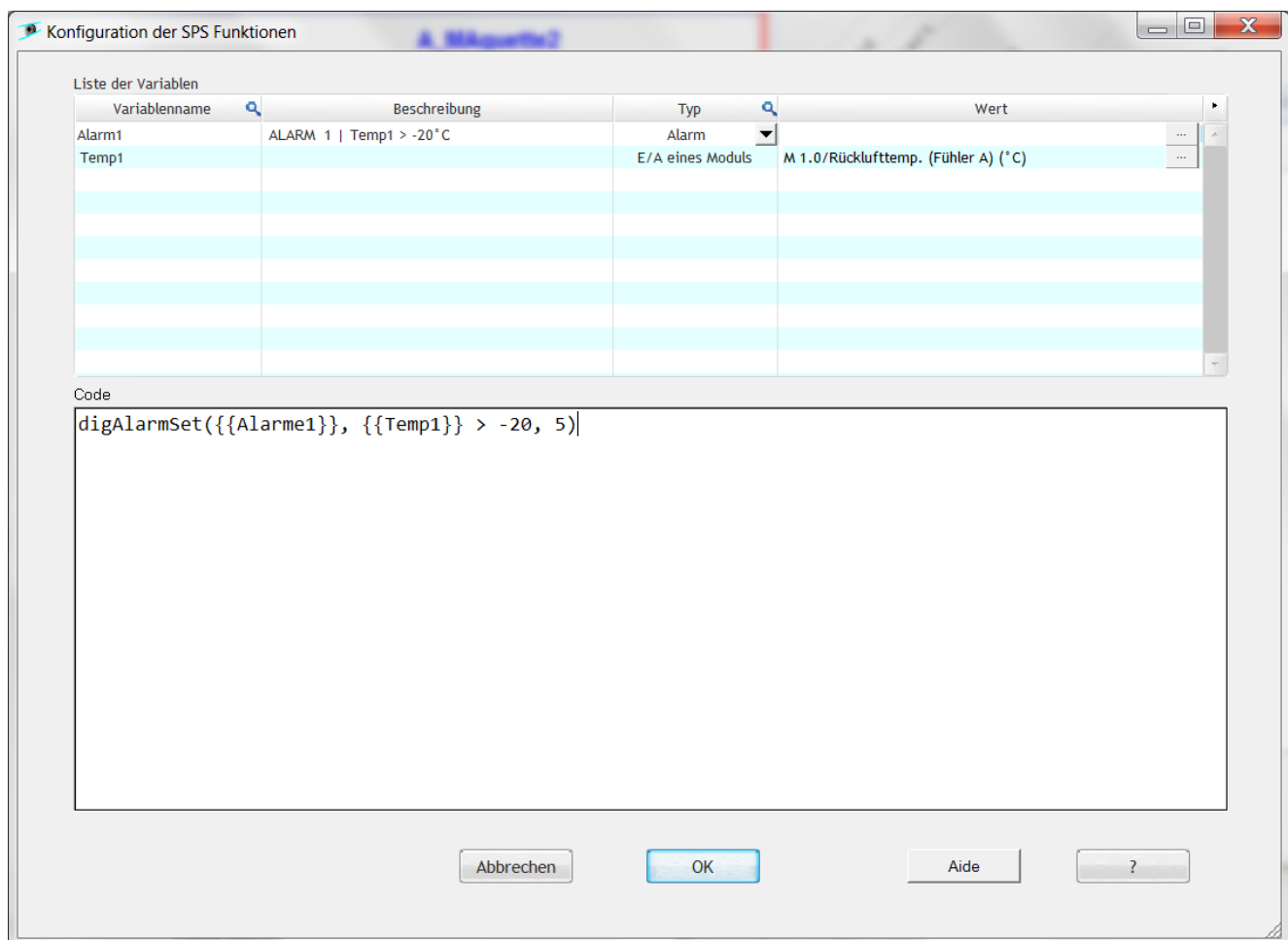
Bedingung, die den Wert False (Wert gleich 0) oder True (Wert ungleich 0) annehmen kann. Diese Bedingung wird typischerweise als Test geschrieben, z.B.: `{{Temp1}} > -20`.

< Verzögerung > : Ganz oder reelle Zahl

Verzögerung in Minuten, bevor der Alarm ausgelöst wird.

EXEMPLE

```
digAlarmSet({{Alarme1}}, {{Temp1}} > -20, 5)
```



6.2.2. digAlarmGetState

Liefert den aktuellen Alarmstatus eines bestimmten Alarms, 0 = kein Alarm, 1 = Alarm an.

SYNTAX

```
digAlarmGetState(Name)
```

<Name> : Zeichenkette

Alarmreferenz (Name der Variablen in der Tabelle **Liste der Variablen**).

BEISPIEL

```
IF digAlarmGetState({{Alarm1}}) THEN
    // Durchzuführende Aktion
END
```

6.2.3. digMessageSend

Sendet eine Nachricht per E-Mail oder SMS.

Achtung! Die Einstellungen "E-Mail-Benachrichtigung" in der Zentraleinheit müssen korrekt konfiguriert sein, damit das Senden von E-Mails funktioniert. Ebenso müssen die Einstellungen "SMS Benachrichtigung" konfiguriert und das GSM-Modem angeschlossen sein, damit der SMS-Versand funktioniert.

SYNTAX

```
digMessageSend(Ziel, Nachricht, Typ)
```

<Ziel> : Zeichenkette

E-Mail-Adresse des Empfängers bei `Typ = "EMail"` oder Telefonnummer bei `Typ = "SMS"`.

<Nachricht> : Zeichenkette

zu sendende Nachricht

<Typ> : Zeichenkette

Nachrichtentyp. `Typ = "EMail"` oder `"SMS"`.

EXEMPLE

```
{{Nachricht1}} = {{ " Zu sendende Nachricht " }}
digMessageSend({"test@example.com"}, {{ Nachricht1}}, {"EMail"})
digMessageSend({"+411234567"}, {{ Nachricht1}}, {"SMS"})
```

6.2.4. digTrace

Sendet eine Zeichenkette an das **Trace**-Fenster.

SYNTAX

```
digTrace(Nachricht)
```

< Nachricht > : Zeichenkette

Die Meldung, die im **Trace**-Fenster angezeigt werden soll..

EXEMPLE

```
digTrace({"Temperatur = "} + {{Temp}} + {" °C. Heizung = "} + {{Heizung}})
```

6.2.1. digSetpointShift

Kann mit Regler-Firmware-Versionen gleich oder größer als 21011 und mit Software-Versionen DC58 21011 oder höher verwendet werden. Kann mit den Reglern DC24D, DC24DE, DC24E, DC24EE verwendet werden, die in den Modi 0, 1 oder 2 arbeiten.

Verschiebt den Sollwert von dem in den Reglerparametern programmierten Sollwert. Der Sollwert wird für 10 Minuten seit dem letzten `digSetpointShift()`-Befehl verschoben. Nach dieser Zeit wird der in den Reglerparametern programmierte Sollwert automatisch wiederhergestellt. Der ursprüngliche Sollwert kann sofort wiederhergestellt werden, indem der gleiche Befehl mit der Konstante `CONTROLLER_SETPOINT` als Parameter `rShift` gesendet wird.

SYNTAXE

```
digSetpointShift(unitID, rShift)
```

<unitID> : Ganz oder reelle Zahl

Kennung des Reglers, die in der Spalte `unitID` der Tabelle "Systemkonfiguration" sichtbar ist

<rShift> : Ganz oder reelle Zahl

Offset (positiv oder negativ) in Bezug auf den im Controller programmierten Sollwert.

EXEMPLE

```
digSetpointShift(7, 5)
```

Siehe Kapitel 6.4.1 für ein weiteres Beispiel.

6.2.2. digSetpointSetTR

Kann mit Regler-Firmware-Versionen gleich oder höher als 21011 und mit Software-Versionen DC58 21011 oder höher verwendet werden. Kann nur mit Regler vom Typ DC24TR verwendet werden, die in den Modi 0 oder 1 arbeiten.

Ändert den Sollwert des Reglers auf den Sollwert, der im Parameter `rNewSetpoint` ist. Der Sollwert wird für 10 Minuten seit dem letzten `digSetpointShiftTR()`-Befehl verschoben. Nach dieser Zeit wird der in den Reglerparametern programmierte Sollwert automatisch wiederhergestellt. Der ursprüngliche Sollwert kann sofort wiederhergestellt werden, indem der gleiche Befehl mit der Konstante `CONTROLLER_SETPOINT` als Parameter `rNewSetpoint` gesendet wird.

SYNTAXE

```
digSetpointSetTR(unitID, rNewSetpoint)
```

`<unitID>` : Ganz oder reelle Zahl

Kennung des Reglers, die in der Spalte `unitID` der Tabelle "Systemkonfiguration" sichtbar ist

`<rNewSetpoint>` : Ganz oder reelle Zahl

Neuer Reglersollwert (Gaskühlersollwert in Modus 0 oder HD-Sollwert in Modus 1).

EXEMPLE

```
IF {{TempWarmwasser}} > {{ Grenzwert}} THEN
    digSetpointSetTR(7, 100)
ELSE
    digSetpointSetTR(7, CONTROLLER_SETPOINT)
END
```

6.2.3. digSetpointSetTR_MP

Kann mit Regler-Firmware-Versionen gleich oder höher als 21011 und mit Software-Versionen DC58 21011 oder höher verwendet werden. Kann nur mit Regler vom Typ DC24TR verwendet werden, die in den Modi 0 arbeiten.

Ändert den Sollwert des Reglers auf den Mitteldrucksollwert (MP), der im Parameter `rNewSetpoint` ist. Der Sollwert wird für 10 Minuten seit dem letzten `digSetpointShiftTR()`-Befehl verschoben. Nach dieser Zeit wird der in den Reglerparametern programmierte Sollwert automatisch wiederhergestellt. Der ursprüngliche Sollwert kann sofort wiederhergestellt werden, indem der gleiche Befehl mit der Konstante `CONTROLLER_SETPOINT` als Parameter `rNewSetpoint` gesendet wird.

SYNTAXE

```
digSetpointSetTR_MP(unitID, rNewSetpoint)
```

`<unitID>` : Ganz oder reelle Zahl

Kennung des Reglers, die in der Spalte `unitID` der Tabelle "Systemkonfiguration" sichtbar ist

`<rNewSetpoint>` : Ganz oder reelle Zahl

Neuer Reglersollwert (MP-Sollwert im Modus 1).

EXEMPLE

```
digSetpointSetTR_MP(7, 37)
```

6.3. SYSTEMVARIABLEN, DIE IN SPS-FUNKTIONEN VERWENDET WERDEN KÖNNEN.

`minuteFrom2000`

Minuten seit 1.01.2000 00:00

`secondFrom2000`

Sekunden seit 1.01.2000 00:00

`hourFrom2000`

Stunden seit 1.01.2000 00:00

`minuteOfDay`

Minuten seit Mitternacht (00:00:00)

`secondOfDay`

Sekunden seit Mitternacht (00:00:00)

`hourOfDay`

Stunden seit Mitternacht (00:00:00)

6.4.1. CONTROLLER_OUTPUT

Die Systemkonstante CONTROLLER_OUTPUT ermöglicht es Ihnen, die Hand zu einem Modul zurückzukehren, nachdem Sie den Ausgang auf 1 oder 0 für digitale Ausgänge oder zwischen 0 und 100% für analoge Ausgänge gezwungen haben.

Anders ausgedrückt, können einem Ausgang eines Moduls 3 verschiedene Werte zugeordnet werden:

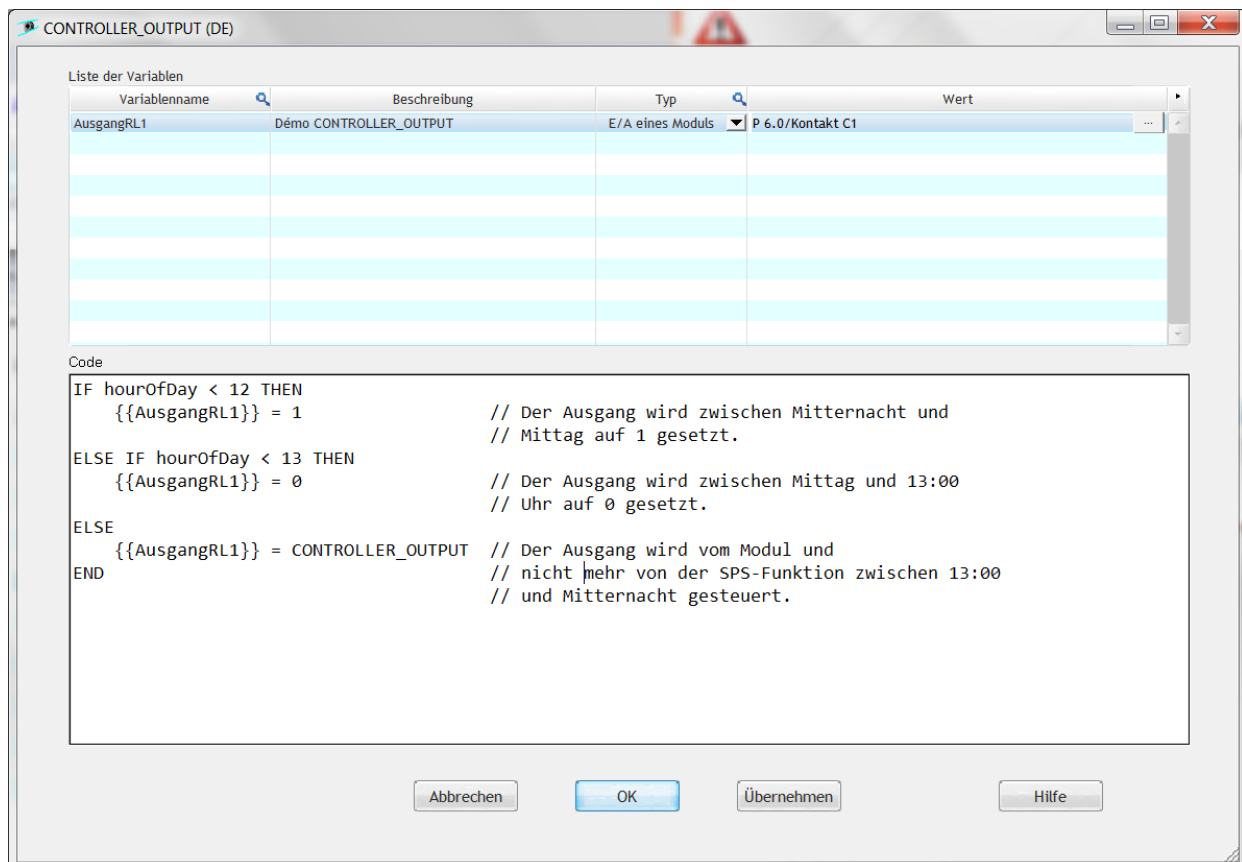
- **1**: Der Ausgang wird auf 1 gesetzt.
- **0**: Der Ausgang wird auf 0 gesetzt.
- **CONTROLLER_OUTPUT**: Der Ausgang wird vom Modul und nicht mehr von der SPS-Funktion gesteuert.

BEISPIEL

```

IF hourOfDay < 12 THEN
    {{AusgangRL1}} = 1                // Der Ausgang wird zwischen Mitternacht und
                                     // Mittag auf 1 gesetzt.
ELSE IF hourOfDay < 13 THEN
    {{AusgangRL1}} = 0                // Der Ausgang wird zwischen Mittag und 13:00
                                     // Uhr auf 0 gesetzt.
ELSE
    {{AusgangRL1}} = CONTROLLER_OUTPUT // Der Ausgang wird vom Modul und
END                                  // nicht mehr von der SPS-Funktion zwischen 13:00
                                     // und Mitternacht gesteuert.

```



6.4.1. CONTROLLER_SETPOINT

Die Systemkonstante `CONTROLLER_SETPOINT` wird verwendet, um nach dem Erzwingen einer Sollwertverschiebung (`digSetpointShift()`) oder eines neuen Sollwerts (`digSetpointSetTR()` bzw. `digSetpointSetTR_MP()`) die Hand an ein Modul zurückzugeben.

BEISPIEL

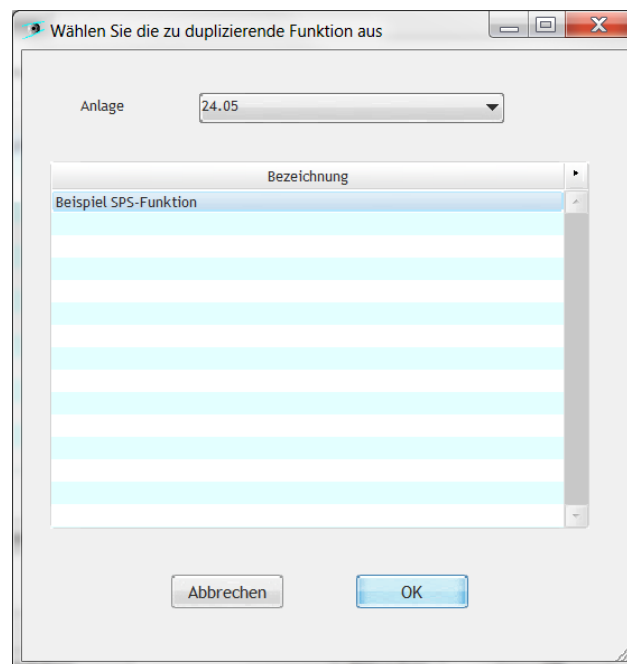
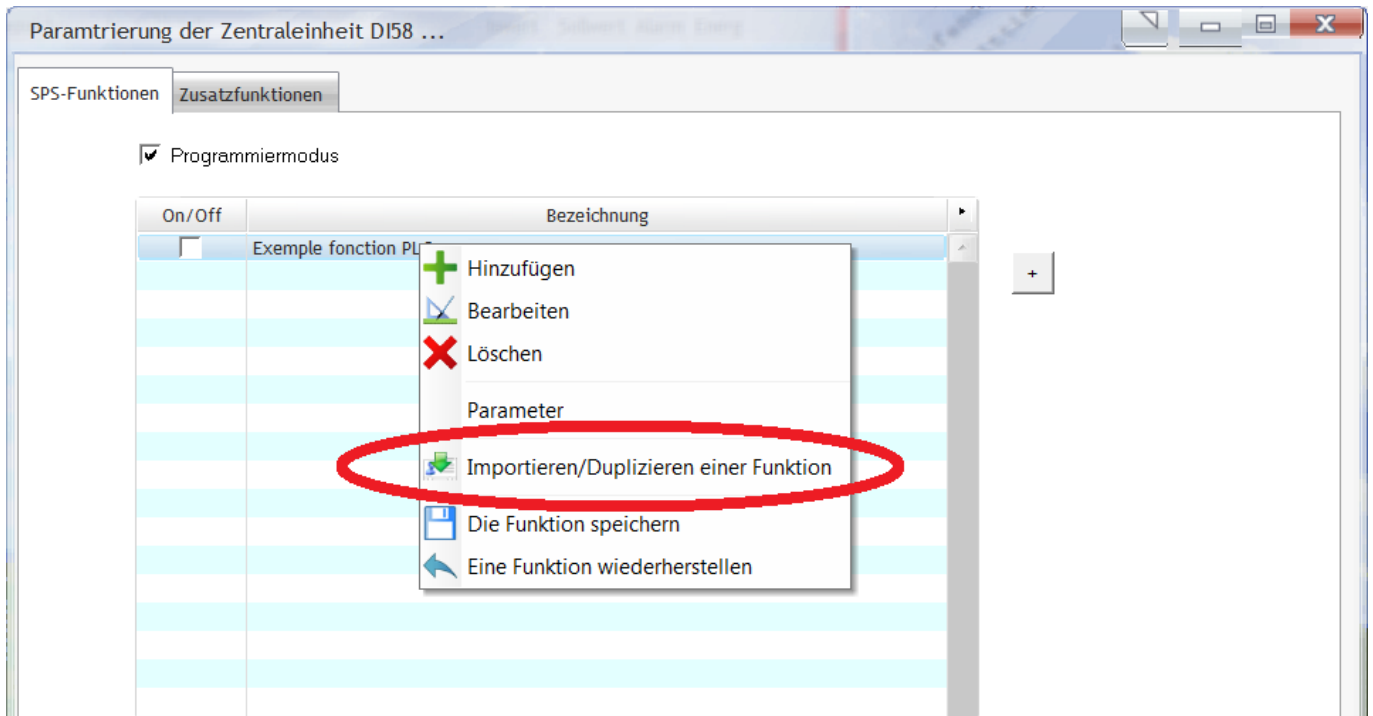
```
IF hourOfDay > 18 OR hourOfDay < 8 THEN
  digSetpointShift(10, 5)
  // Der Sollwert wird zwischen 18 Uhr und 8 Uhr um +5° verschoben.
ELSE
  digSetpointShift(10, CONTROLLER_SETPOINT)
  // Der Ausgang wird durch das Modul gesteuert.
END
```


8. IMPORTIERUNG ET DUPLIZIERUNG VON FUNKTIONEN

Es ist möglich, Funktionen aus einer anderen Installation zu importieren und zu duplizieren. Öffnen Sie dazu das Kontextmenü mit der rechten Maustaste und wählen Sie die Funktion **Importieren/Duplizieren einer Funktion**. Im folgenden Fenster können Sie die Anlage auswählen, auf der sich die Funktion befindet. Wenn sich diese Anlage von derjenigen unterscheidet, mit der Sie verbunden sind, wird die Funktion importiert. Wenn es sich dagegen um die gleiche Installation handelt, wird die Funktion dupliziert.

Die Verknüpfungen von Variablen vom Typ "I/O eines Moduls" mit den Ein- und Ausgängen der Module ("Wert" Spalte) werden nicht kopiert, da die neue Funktion im Allgemeinen andere Ein- und Ausgänge verwendet. Diese Links müssen manuell durch Anklicken der Schaltfläche "..." in der Spalte "Wert" ausgewählt werden.

Es ist möglich, die Funktionen durch Doppelklick auf ihren Namen umzubenennen.



10. BEISPIELE

10.1. MASCHINENRAUMBELÜFTUNG

```
// Maschinenraumbelüftung
IF {{TMaschinenraum}} > ( {{Sollwert}}+{{Delta}} ) AND {{TAussen}} < {{TMaschinenraum}} THEN
    {{Ventilator}} = 1
ELSE
    IF {{TMaschinenraum}} < {{Sollwert}} OR {{TAussen}} >= {{TMaschinenraum}} THEN
        {{Ventilator}} = 0
    END
END
digTrace({"Ventilator = "} + {{ Ventilator }})
```

Konfiguration der SPS Funktionen

Variablenname	Beschreibung	Typ	Wert
TMaschinenraum	Temperatur im Maschinenraum	E/A eines Moduls	M 1.1/Temperatur Fühler F (°C)
TAussen	Aussentemperatur	E/A eines Moduls	M 4.2/Fühler E (°C)
Sollwert	Sollwert	Parameter	25.0
Delta	Delta	Parameter	1.0
Ventilator	Ventilatorsteuerung	E/A eines Moduls	M 1.2/Ausgangskontakt RL3

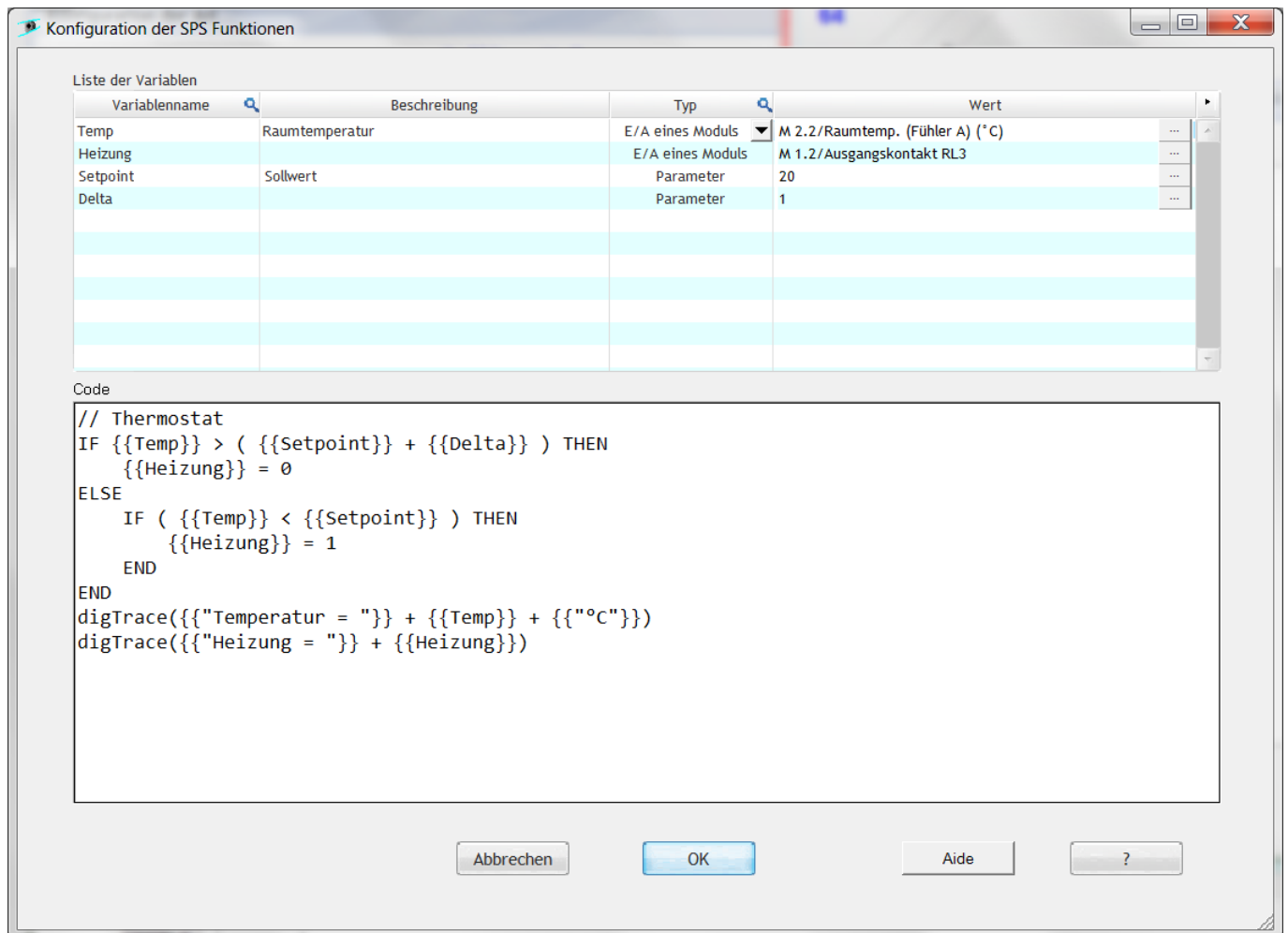
Code

```
// Maschinenraumbelüftung
IF {{TMaschinenraum}} > ( {{Sollwert}}+{{Delta}} ) AND {{TAussen}} < {{TMaschinenraum}} THEN
    {{Ventilator}} = 1
ELSE
    IF {{TMaschinenraum}} < {{Sollwert}} OR {{TAussen}} >= {{TMaschinenraum}} THEN
        {{Ventilator}} = 0
    END
END
digTrace({"Ventilator = "} + {{ Ventilator }})
```

Abbrechen OK Aide ?

10.2. THERMOSTAT

```
// Thermostat
IF {{Temp}} > ( {{Setpoint}} + {{Delta}} ) THEN
    {{Heizung}} = 0
ELSE
    IF ( {{Temp}} < {{Setpoint}} ) THEN
        {{Heizung}} = 1
    END
END
digTrace({"Temperatur = "} + {{Temp}} + {"°C"})
digTrace({"Heizung = "} + {{Heizung}})
```



The screenshot shows a software window titled "Konfiguration der SPS Funktionen". It contains a table of variables and a code editor.

Variablenname	Beschreibung	Typ	Wert
Temp	Raumtemperatur	E/A eines Moduls	M 2.2/Raumtemp. (Fühler A) (°C)
Heizung		E/A eines Moduls	M 1.2/Ausgangskontakt RL3
Setpoint	Sollwert	Parameter	20
Delta		Parameter	1

Code

```
// Thermostat
IF {{Temp}} > ( {{Setpoint}} + {{Delta}} ) THEN
    {{Heizung}} = 0
ELSE
    IF ( {{Temp}} < {{Setpoint}} ) THEN
        {{Heizung}} = 1
    END
END
digTrace({"Temperatur = "} + {{Temp}} + {"°C"})
digTrace({"Heizung = "} + {{Heizung}})
```

Buttons: Abbrechen, OK, Aide, ?

10.3. ZYKLISCHER TIMER

```
// Aktiviert den Ausgang für 2 Minuten,  
// Deaktiviert es dann 8 Minuten lang.  
// für einen Zyklus mit einer Gesamtdauer von 10 Minuten.
```

```
IF modulo (minuteFrom2000, 10) < 2 THEN  
    {{Output}} = 1  
ELSE  
    {{Output}} = 0  
END  
digTrace({"Output = "} + {{ Output }})
```

Konfiguration der SPS Funktionen

Variablenname	Beschreibung	Typ	Wert
Output	Heizung	E/A eines Moduls	M 1.2/Ausgangskontakt RL3

Code

```
// Aktiviert den Ausgang für 2 Minuten,  
// Deaktiviert es dann 8 Minuten lang.  
// für einen Zyklus mit einer Gesamtdauer von 10 Minuten.  
  
IF modulo (minuteFrom2000, 10) < 2 THEN  
    {{Output}} = 1  
ELSE  
    {{Output}} = 0  
END  
digTrace({"Output = "} + {{ Output }})
```

Abbrechen OK Aide ?

10.4. VERWENDUNG VON VORDEFINIERTEN TIMERN IN DER ZENTRALEINHEIT

```
// Für die Konfiguration des Timers siehe Handbuch  
// "Newel 3 - Complet - DE.pdf".  
// im Kapitel 10.12.13
```

```
IF {{Timer_1}} THEN  
    digTrace({"Timer = 1"})  
ELSE  
    digTrace({"Timer = 0"})  
END
```

