

Handbuch | DE

TS4110

TwinCAT 2 | PLC Temperature Controller

Supplement | Control



Inhaltsverzeichnis

1	Vorwort	5
1.1	Hinweise zur Dokumentation	5
1.2	Sicherheitshinweise	6
1.3	Hinweise zur Informationssicherheit	7
2	Übersicht	8
3	Prinzipschaltbild	9
4	Sollwertaufbereitung	10
5	Stellgrößenaufbereitung	12
6	Regelalgorithmus	13
7	Alarming	14
8	Selftuning	15
9	Schrittweise Inbetriebnahme des Reglers	16
10	Beispiel	19
11	PLC-API	20
11.1	FB_CTRL_TempController	20
11.2	Definition der Strukturen	22
11.3	Vorherige Implementierung	30
11.3.1	FB_TempController	33
11.3.2	Definition der Strukturen	37

1 Vorwort

1.1 Hinweise zur Dokumentation

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs- und Automatisierungstechnik, das mit den geltenden nationalen Normen vertraut ist.

Zur Installation und Inbetriebnahme der Komponenten ist die Beachtung der Dokumentation und der nachfolgenden Hinweise und Erklärungen unbedingt notwendig.

Das Fachpersonal ist verpflichtet, für jede Installation und Inbetriebnahme die zu dem betreffenden Zeitpunkt veröffentlichte Dokumentation zu verwenden.

Das Fachpersonal hat sicherzustellen, dass die Anwendung bzw. der Einsatz der beschriebenen Produkte alle Sicherheitsanforderungen, einschließlich sämtlicher anwendbaren Gesetze, Vorschriften, Bestimmungen und Normen erfüllt.

Disclaimer

Diese Dokumentation wurde sorgfältig erstellt. Die beschriebenen Produkte werden jedoch ständig weiter entwickelt.

Wir behalten uns das Recht vor, die Dokumentation jederzeit und ohne Ankündigung zu überarbeiten und zu ändern.

Aus den Angaben, Abbildungen und Beschreibungen in dieser Dokumentation können keine Ansprüche auf Änderung bereits gelieferter Produkte geltend gemacht werden.

Marken

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® und XPlanar® sind eingetragene und lizenzierte Marken der Beckhoff Automation GmbH.

Die Verwendung anderer in dieser Dokumentation enthaltenen Marken oder Kennzeichen durch Dritte kann zu einer Verletzung von Rechten der Inhaber der entsprechenden Bezeichnungen führen.

Patente

Die EtherCAT-Technologie ist patentrechtlich geschützt, insbesondere durch folgende Anmeldungen und Patente:

EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702

mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.

EtherCAT®

EtherCAT® ist eine eingetragene Marke und patentierte Technologie lizenziert durch die Beckhoff Automation GmbH, Deutschland

Copyright

© Beckhoff Automation GmbH & Co. KG, Deutschland.

Weitergabe sowie Vervielfältigung dieses Dokuments, Verwertung und Mitteilung seines Inhalts sind verboten, soweit nicht ausdrücklich gestattet.

Zu widerhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patent-, Gebrauchsmuster- oder Geschmacksmustereintragung vorbehalten.

1.2 Sicherheitshinweise

Sicherheitsbestimmungen

Beachten Sie die folgenden Sicherheitshinweise und Erklärungen!
Produktspezifische Sicherheitshinweise finden Sie auf den folgenden Seiten oder in den Bereichen Montage, Verdrahtung, Inbetriebnahme usw.

Haftungsausschluss

Die gesamten Komponenten werden je nach Anwendungsbestimmungen in bestimmten Hard- und Software-Konfigurationen ausgeliefert. Änderungen der Hard- oder Software-Konfiguration, die über die dokumentierten Möglichkeiten hinausgehen, sind unzulässig und bewirken den Haftungsausschluss der Beckhoff Automation GmbH & Co. KG.

Qualifikation des Personals

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs-, Automatisierungs- und Antriebstechnik, das mit den geltenden Normen vertraut ist.

Erklärung der Symbole

In der vorliegenden Dokumentation werden die folgenden Symbole mit einem nebenstehenden Sicherheitshinweis oder Hinweistext verwendet. Die Sicherheitshinweise sind aufmerksam zu lesen und unbedingt zu befolgen!

GEFAHR

Akute Verletzungsgefahr!

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, besteht unmittelbare Gefahr für Leben und Gesundheit von Personen!

WARNUNG

Verletzungsgefahr!

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, besteht Gefahr für Leben und Gesundheit von Personen!

VORSICHT

Schädigung von Personen!

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, können Personen geschädigt werden!

HINWEIS

Schädigung von Umwelt oder Geräten

Wenn der Hinweis neben diesem Symbol nicht beachtet wird, können Umwelt oder Geräte geschädigt werden.

Tipp oder Fingerzeig



Dieses Symbol kennzeichnet Informationen, die zum besseren Verständnis beitragen.

1.3 Hinweise zur Informationssicherheit

Die Produkte der Beckhoff Automation GmbH & Co. KG (Beckhoff) sind, sofern sie online zu erreichen sind, mit Security-Funktionen ausgestattet, die den sicheren Betrieb von Anlagen, Systemen, Maschinen und Netzwerken unterstützen. Trotz der Security-Funktionen sind die Erstellung, Implementierung und ständige Aktualisierung eines ganzheitlichen Security-Konzepts für den Betrieb notwendig, um die jeweilige Anlage, das System, die Maschine und die Netzwerke gegen Cyber-Bedrohungen zu schützen. Die von Beckhoff verkauften Produkte bilden dabei nur einen Teil des gesamtheitlichen Security-Konzepts. Der Kunde ist dafür verantwortlich, dass unbefugte Zugriffe durch Dritte auf seine Anlagen, Systeme, Maschinen und Netzwerke verhindert werden. Letztere sollten nur mit dem Unternehmensnetzwerk oder dem Internet verbunden werden, wenn entsprechende Schutzmaßnahmen eingerichtet wurden.

Zusätzlich sollten die Empfehlungen von Beckhoff zu entsprechenden Schutzmaßnahmen beachtet werden. Weiterführende Informationen über Informationssicherheit und Industrial Security finden Sie in unserem <https://www.beckhoff.de/secguide>.

Die Produkte und Lösungen von Beckhoff werden ständig weiterentwickelt. Dies betrifft auch die Security-Funktionen. Aufgrund der stetigen Weiterentwicklung empfiehlt Beckhoff ausdrücklich, die Produkte ständig auf dem aktuellen Stand zu halten und nach Bereitstellung von Updates diese auf die Produkte aufzuspielen. Die Verwendung veralteter oder nicht mehr unterstützter Produktversionen kann das Risiko von Cyber-Bedrohungen erhöhen.

Um stets über Hinweise zur Informationssicherheit zu Produkten von Beckhoff informiert zu sein, abonnieren Sie den RSS Feed unter <https://www.beckhoff.de/secinfo>.

2 Übersicht

Der TwinCAT Temperature Controller ist ein universell einsetzbarer SPS-Baustein zum Überwachen und Regeln von verschiedensten temperaturabhängigen Prozessen. Der Regler lässt sich betreiben im

- Automatik (Closed loop) und
- Handbetrieb (Open loop).

Die Stellgröße kann analog und digital abgegriffen werden. Der digitale Stellgröße ist pulsweitenmoduliert (PWM). Ein Zweipunkt- bzw. Dreipunktausgang steht auch zur Verfügung. Die Stellgröße ist begrenzt auf die zulässigen Minimal- bzw. Maximalwerte.

Die Sollgröße ist auch auf zulässige Minimal- bzw. Maximalwerte begrenzt und kann weiterhin gerampt werden. Zur einfachen Umschaltung von Sollwert zu Standby-Sollwert steht ein Bit im Interface des Bausteins zur Verfügung. Zur Unterstützung des sogenannten Heater-Bakings kann ein Soft-Start parametrierbar werden. Hierzu wird der Sollwert (optional gerampt) auf einen niedrigen Sollwert hochgefahren, dort für eine gewisse Zeit gehalten und dann zum eigentlichen Sollwert (optional gerampt) hochgefahren.

Die Istgröße kann digital gefiltert werden.

Der Regelalgorithmus ist PID basiert. Zur Verringerung des Überschwingens kann zusätzlich ein Vorregler eingesetzt werden.

Der Regler besitzt diverse parametrierbare Überwachungsfunktionen. Es gibt eine

- Toleranzbandüberwachung (zwei unterschiedliche Toleranzbänder),
- eine Absolutwertüberwachung,
- eine Geberüberwachung (offen, Backvoltage, reverse) und
- eine Heizstromüberwachung (offen, Kurzschluss, Leckstrom).

Um die Inbetriebnahme des Reglers wesentlich zu vereinfachen, gibt es einen Algorithmus zur Ermittlung der optimalen Reglerparameter. Dieser Algorithmus wertet einen Sprung aus und ermittelt über die Wendetangentenmethode die maximale Geschwindigkeit und die Verzugszeit der Strecke. Mit diesen Angaben kann ein Regler nach den Regeln von Chien, Hrones und Reswick entworfen werden. Hier werden auch die Parameter des Vorreglers ermittelt. Sollten schon die Reglerparameter schon bekannt sein, so kann der Regler auch mit diesen externen Parametern betrieben werden. Die Ermittlung der Reglerparameter kann für Heiz- und Kühlstrecke getrennt ermittelt werden. Ein entsprechender Ablauf des Tuning Vorgangs kann vorgewählt werden. Sollte für das Kühlen kein getrennter Parametersatz ermittelt werden, dann ist es auch möglich den Heizparametersatz mittels eines freiwählbaren Skalierungsfaktors auch für die Kühlstrecke zu verwenden.

[Schrittweise Inbetriebnahme des Reglers](#) [[▶ 16](#)]

Dokumentation des [Funktionsblocks](#) [[▶ 20](#)] und der notwendigen [Strukturen](#) [[▶ 22](#)].



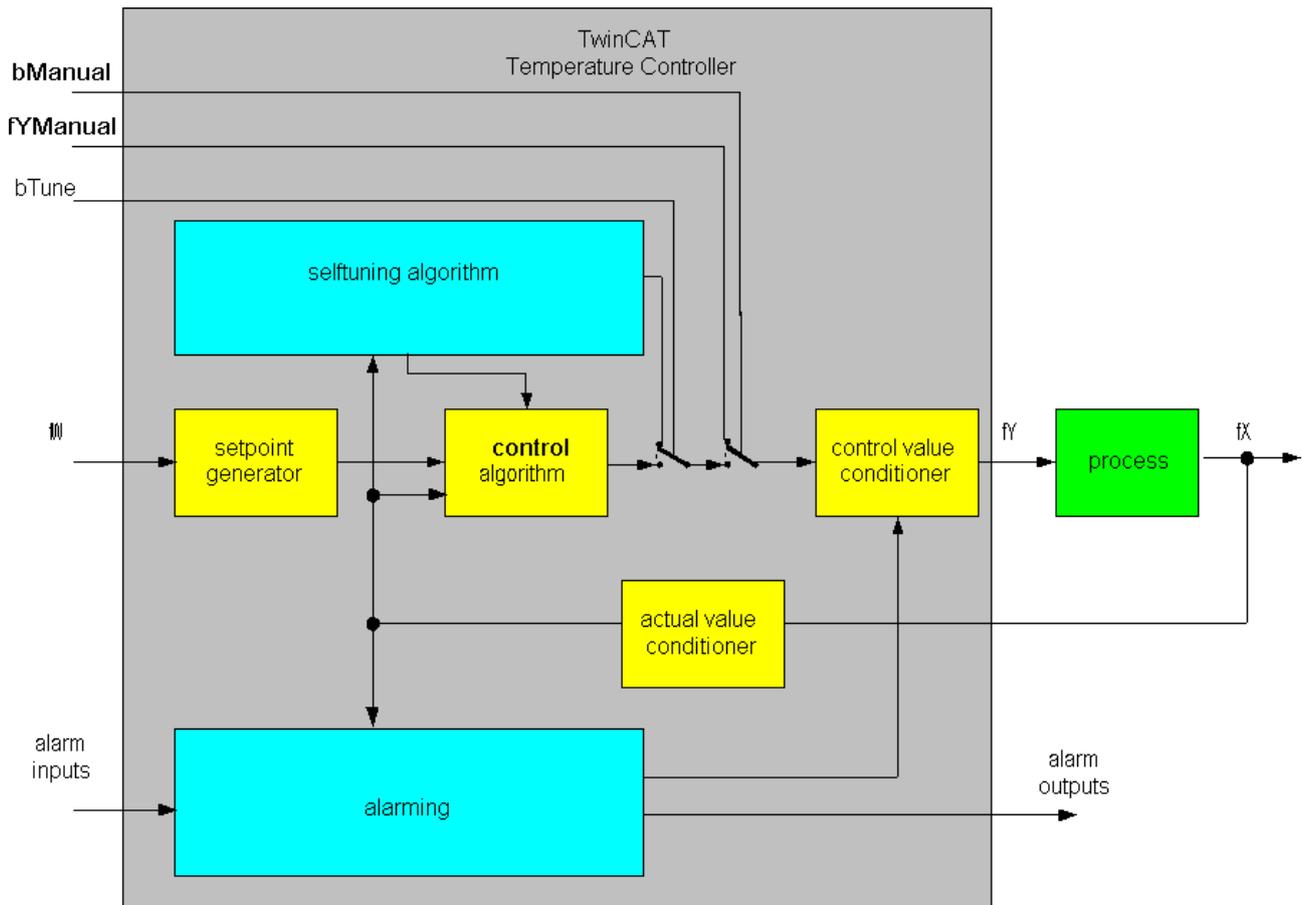
Aus Kompatibilitätsgründen ist die Beschreibung des bisherigen [Temperatur Reglers](#) [[▶ 30](#)] und der Strukturen in dieser Dokumentation noch enthalten.

3 Prinzipschaltbild

Der TwinCAT Temperature Controller besteht aus einer Reihe von Funktionsbausteinen. Es gibt folgende Funktionsbausteine:

- Selbstuning Algorithmus (FB_Selftuner)
- Regelalgorithmus (FB_ControlAlgorithm)
- Sollwertgenerator (FB_SetpointConditioner)
- Stellgrößengenerator (FB_ControlValueConditioner)
- Alarming (FB_Alarming)

Diese Funktionsbausteine rufen wiederum eine Reihe weiterer untergeordneter Funktionsbausteine auf.

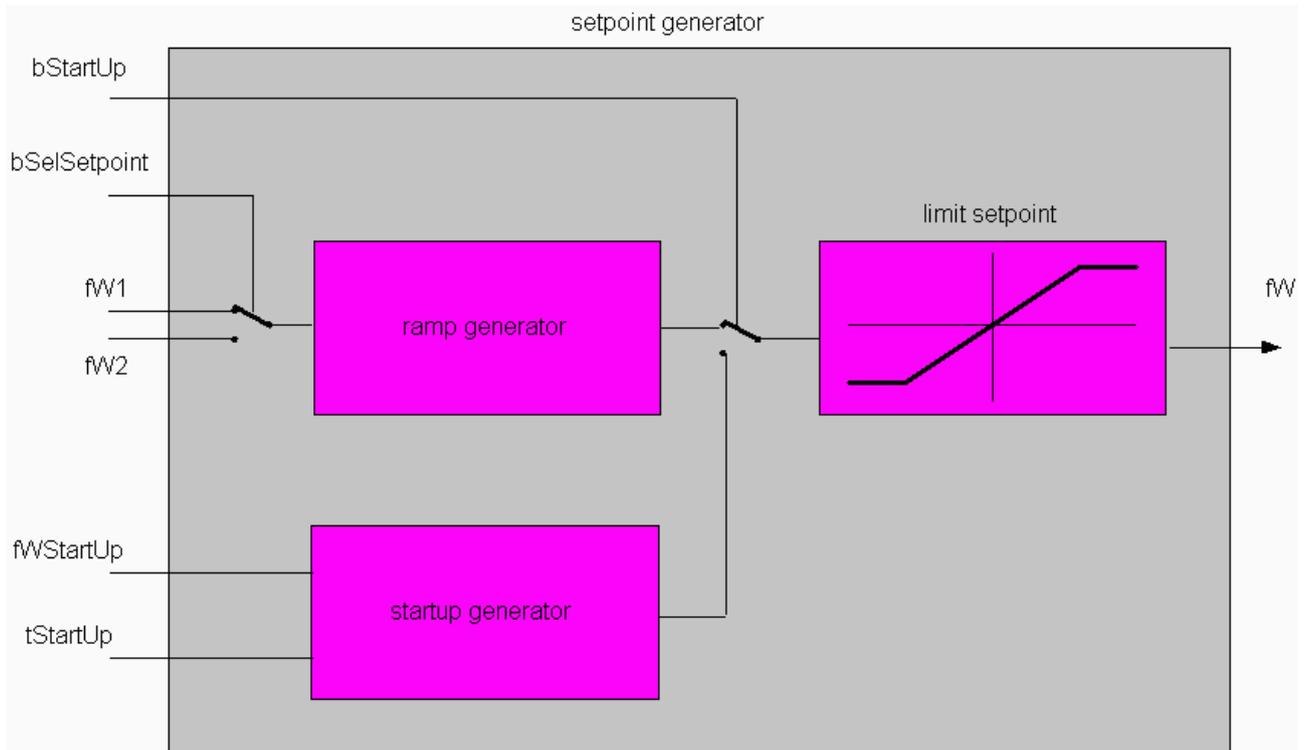


Das Bild zeigt die einzelnen Funktionsbausteine.

4 Sollwertaufbereitung

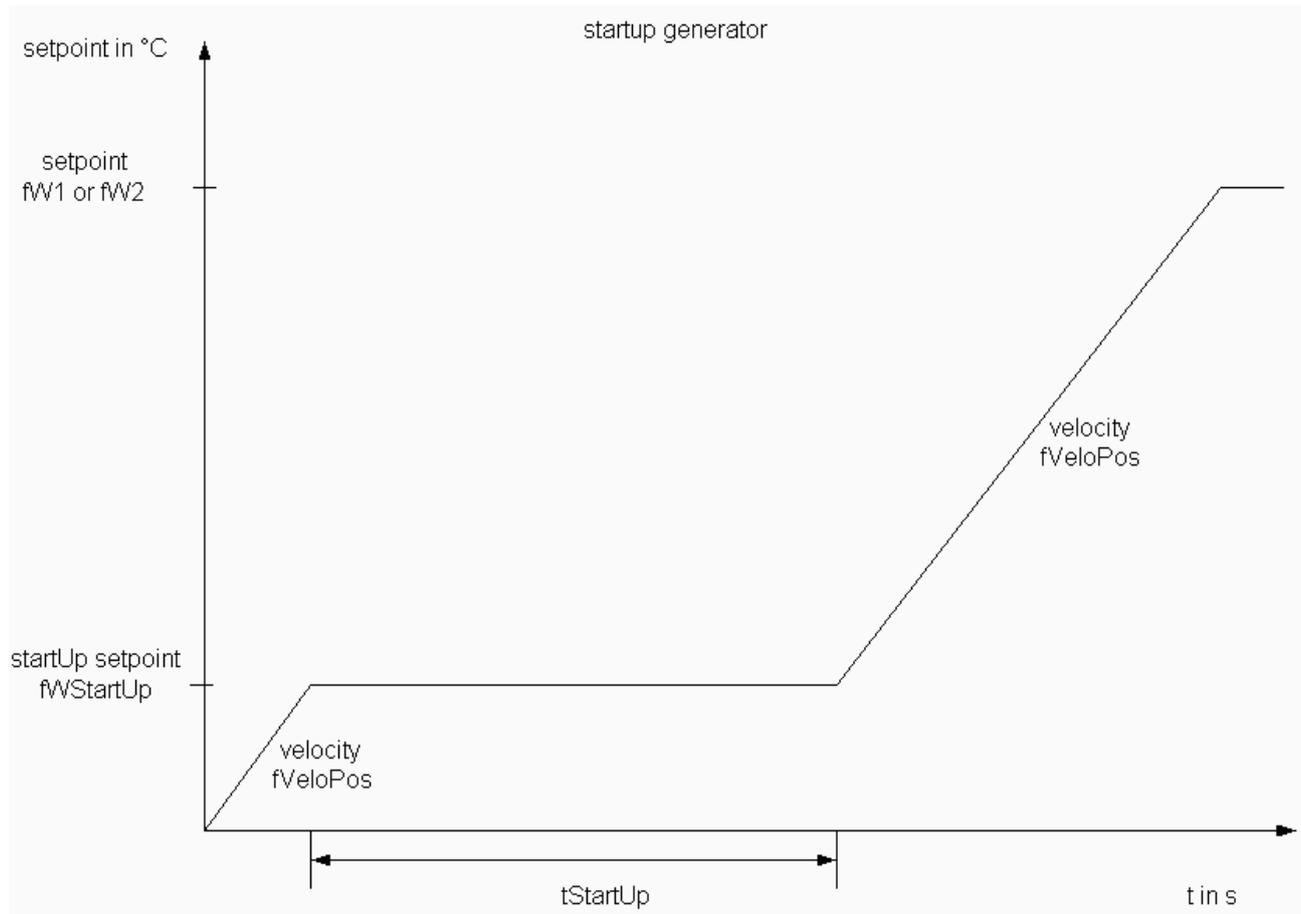
Die Sollwerte können über ein Bit umgeschaltet werden. Neben dem eigentlichen Sollwert gibt es noch einen Standby-Sollwert. Mit dem Standby-Sollwert kann die Temperatur in Pausen auf einen niedrigeren Wert zur Energieeinsparung reduziert werden. Bei Bedarf können die Sollwertgrößen sprünge verrampelt werden. Im Parametersatz für die Sollwerte stehen ein Anstiegsgeschwindigkeit und eine Abfallgeschwindigkeit zur Verfügung.

Die Sollwerte werden auf ihre Grenzen limitiert.



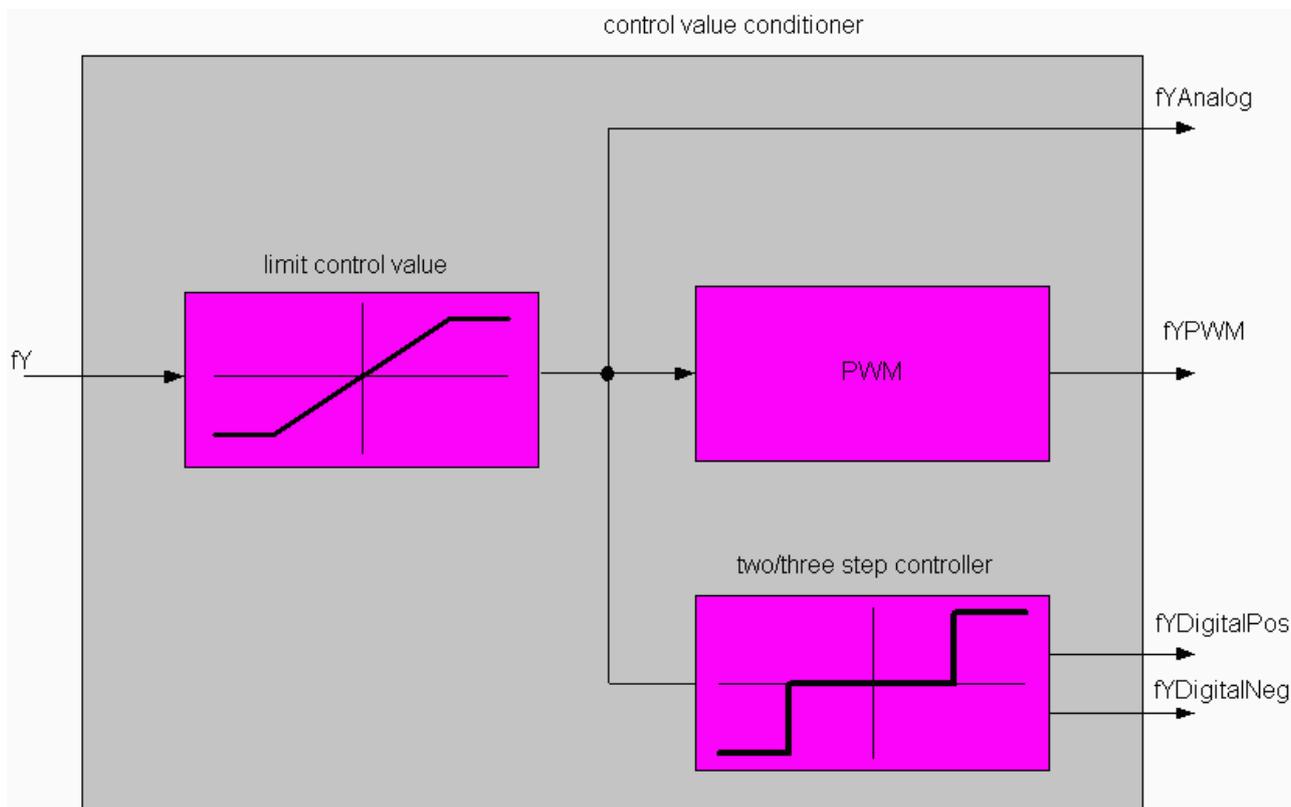
Um das sogenannte Heater-Baking zu ermöglichen, kann ein Softstart parametrierbar werden. Hier wird von Umgebungstemperatur zuerst auf einen niedrigen Sollwert ($fW_{StartUp}$) gerampelt. Die Temperatur wird dann für eine Zeit ($t_{StartUp}$) gehalten und erst danach wird auf den eigentlichen Sollwert hochgerampelt.

Start-Up

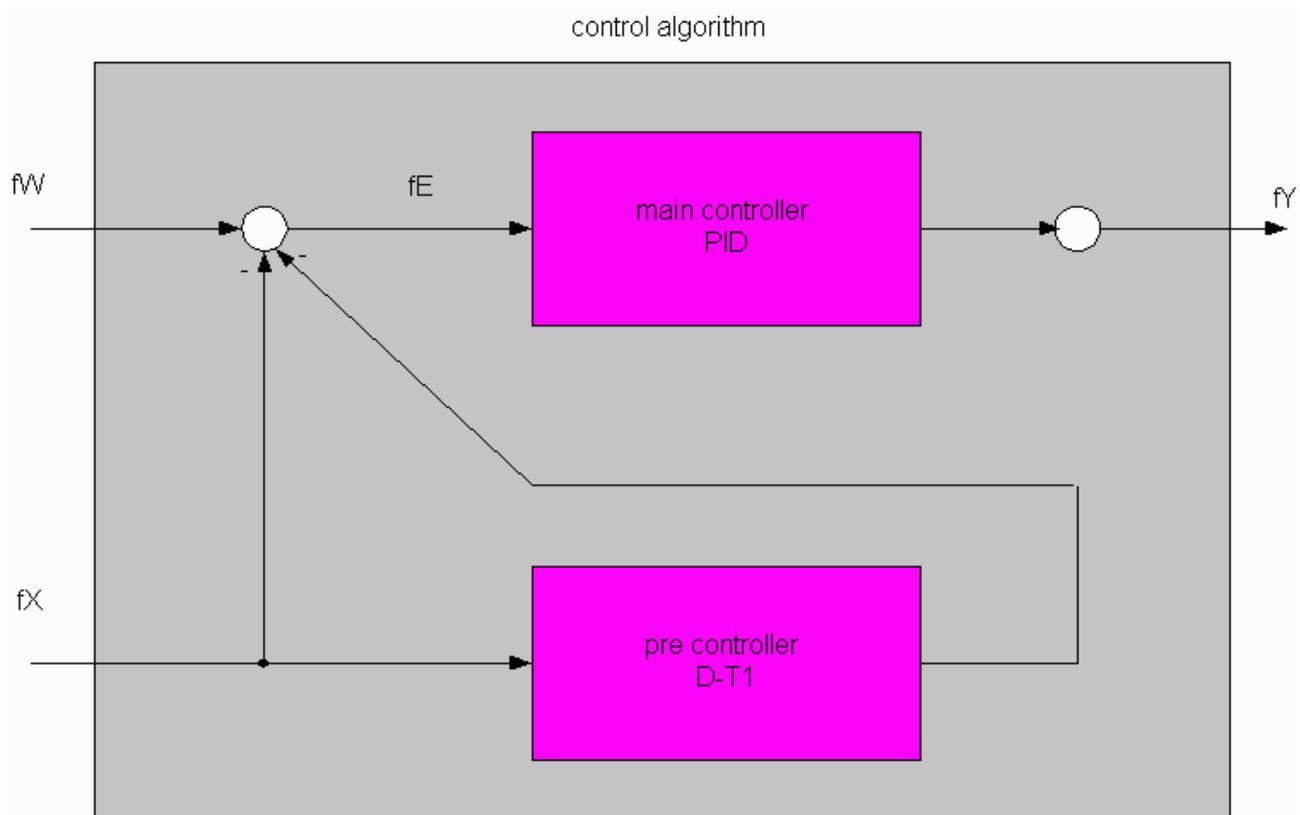


5 Stellgrößenaufbereitung

Die vom Regler berechnete Stellgröße **Control Value (CV)** wird zunächst auf gültige Werte limitiert. Die Begrenzungswerte werden über die Stellgrößenstruktur an den Reglerbaustein übergeben. Es stehen drei verschiedene Möglichkeiten der Stellgröße zur Verfügung. Die Stellgröße kann einmal analog abgegriffen werden. Die gebräuchlichere Ausgabe dürfte allerdings die digitale Ausgabe als pulsweiten moduliertes Signal sein. Die für die Pulsweitenmodulation nötige Zykluszeit wird über die Stellgrößenstruktur an den Regler übergeben. Daneben kann auch ein Zweipunktregler-Ausgang (für Heizen oder Kühlen) und ein Dreipunktregler-Ausgang (für Heizen und Kühlen) abgegriffen werden.

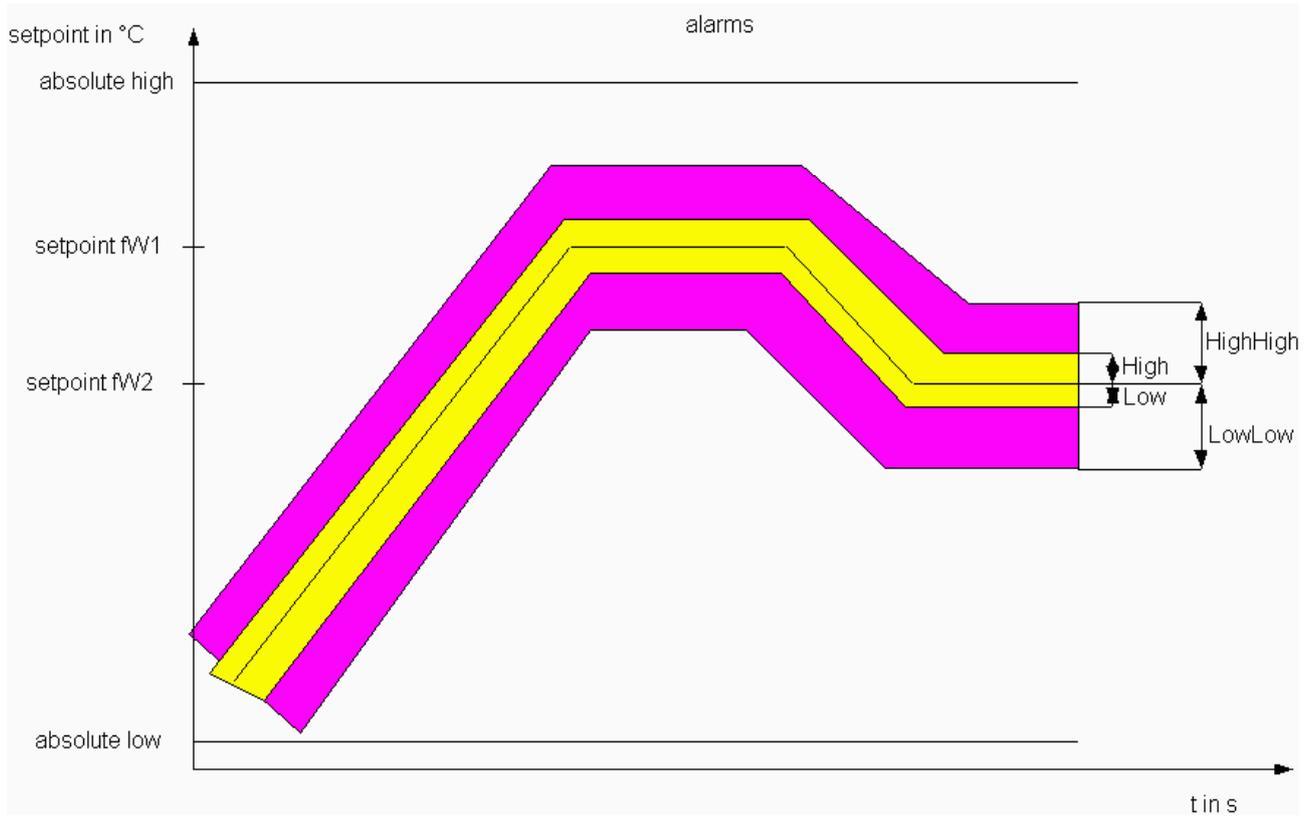


6 Regelalgorithmus



Der TwinCAT Temperature Controller setzt im Kern auf einem Standard-PID Regler auf. Dieser Reglerkern unterstützt auch Anti-Reset-Windup Maßnahmen zur Begrenzung des I-Anteils, wenn die Stellgröße in die Begrenzung geht. Da der Regler durch das Einstellverfahren von Chien, Hrones und Reswick auf Störgrößenminimierung ausgelegt wurde, kann es bei Sollwertänderungen zum Überschwingen kommen. Um dieses Überschwingen zu reduzieren, kann bei Sollwertänderungen ein Vorregler vorgeschaltet werden. Dieser Vorregler hat ein D-T1 Verhalten und verringert das Überschwingen des Gesamtreglers. Da der Vorregler durch den D-Anteil eine Aufräuhung der Stellgröße zur Folge hat, ist der Einsatz des Vorreglers abzuwägen. Der Vorregler wird abgeschaltet, wenn die Istgröße für eine gewisse Zeit in einem Band um den Sollwert eingelaufen ist. Das Abschalten des Vorreglers erfolgt gerampt über einen längeren Zeitraum. Zur Minimierung der Stellgrößenoszillation kann optional ein Filter hinter dem Hauptregler aufgeschaltet werden. Als Filter stehen P-T1 und Moving Average Filter zur Verfügung.

7 Alarming



Im Temperaturregler werden folgende Alarmer ständig überwacht:

- Absolute Temperaturen (high und low)
- Relative Temperaturen (in zwei Bändern um den Sollwert)

Außerdem können folgende sensorseitigen Hardware-Alarmer mit dem Temperaturregler verbunden werden:

- Open Thermocouple: Leitung zum Temperaturfühler ist unterbrochen.
- Back Voltage: am Temperaturfühler liegt eine unzulässig hohe Spannung an.
- Reverse Thermocouple: Temperaturfühler ist mit falscher Polarität angeschlossen.

Ist ein Stromsensor angeschlossen, so können folgende Signale mit dem Temperaturregler verbunden werden:

- Kurzschluss
- offener Stromkreis
- Leckstrom

8 Selftuning

Der Selftuningalgorithmus basiert auf der klassischen Wendetangenten-Methode. Diese Methode stammt in den Anfängen von Ziegler und Nichols. Basis dieses Verfahrens ist die Voraussetzung einer linearen P-T1-Strecke mit Totzeit. Während eines Sprungversuches wird die maximale Änderungsgeschwindigkeit ermittelt. Dies geschieht über eine Differenzenbildung über mehrere Abtastzyklen. Zum Zeitpunkt der maximalen Änderungsgeschwindigkeit wird eine Wendetangente angelegt und der Schnittpunkt mit der Zeitachse bestimmt. Die Verzugszeit T_u ist die Zeit vom Beginn der Messung bis zum Schnittpunkt der Wendetangente mit der Zeitachse. Mit den Angaben von T_u und V_{max} können über die Formeln von Chien, Hrones und Reswick die Reglerparameter für eine Störungsunterdrückung mit 20% Überschwingen ermittelt werden. Die Parameter des Vorreglers lassen sich mit heuristischen Formeln einfach aus den Parametern des Hauptreglers ableiten. Mit diesen Parametern wird nach dem Selftuning automatisch auf den Closed-Loop-Betrieb umgestellt.

9 Schrittweise Inbetriebnahme des Reglers

Folgende Schritte müssen durchgeführt werden:

- **Die Regler-Bibliothek ist über den Bibliotheksmanager in das Projekt einzufügen.**

Im Bibliotheksmanager ist die TcTempCtrl.lib einzufügen.

- **Mindestens eine Instanz des Reglers ist zu programmieren.**

Eine Instanz des Reglerbausteins `FB_TempController` ist anzulegen. Außerdem muss eine Instanz der Struktur `ST_ControllerParameter` angelegt werden.

- **Die notwendige äußere Beschaltung durchführen.**

Name		Beschreibung
<code>eCtrlMode</code>	Beschaltung notwendig	Schaltet den Regler in eine Betriebsart (aktiv, passiv, tuning)
<code>bSelSetpoint</code>	Beschaltung optional	Wählt einen von zwei möglichen Sollwerten aus. Mit FALSE wird der normale Sollwert gewählt, mit TRUE der Standby-Sollwert.
<code>fW1</code>	Beschaltung notwendig	Sollwert.
<code>fW2</code>	Beschaltung optional	Standby-Sollwert im Regelfall kleiner als <code>fW1</code> . Mit <code>fSelSetpoint</code> kann zwischen <code>fW1</code> und <code>fW2</code> umgeschaltet werden.
<code>fX</code>	Beschaltung notwendig	Istwert. Dieser Wert muss auf LREAL konvertiert werden.
<code>fYManual</code>	Beschaltung optional	Stellgröße im Handbetrieb.
<code>bOpenThermocouple</code>	Beschaltung optional	Mit TRUE ist das Thermoelement offen. Muss von der Hardware gemeldet werden (z.B. KLxxxx).
<code>bReverseThermocouple</code>	Beschaltung optional	Mit TRUE wird falsche Polarität des angeschlossenen Thermoelements gemeldet. Muss von der Hardware gemeldet werden.
<code>bBackVoltage</code>	Beschaltung optional	Bei TRUE wird eine zu hohe Eingangsspannung am Thermoelement angezeigt. Muss von der Hardware gemeldet werden.
<code>bLeakage</code>	Beschaltung optional	Mit TRUE wird ein Leckstrom zu Heizelement detektiert. Muss von der Hardware gemeldet werden.
<code>bShortCircuit</code>	Beschaltung optional	Mit TRUE wird ein Kurzschluss im Heizelement detektiert. Muss von der Hardware gemeldet werden.
<code>bOpenCircuit</code>	Beschaltung optional	Mit TRUE wird ein offener Stromkreis im Heizelement detektiert. Muss von der Hardware gemeldet werden.
<code>sControllerParameter</code>	Beschaltung notwendig	In dieser Struktur werden generelle Parameter (Abtastzeiten usw.) an den Baustein übergeben.
<code>sParaControllerExternal</code>	Beschaltung optional	In dieser Struktur wird ein externer Regler-Parametersatz an den Baustein übergeben.

- **Die notwendige Parametrierung des Reglers über die Struktur durchführen.**

Die Parameter können über Initialwerte oder über Zuweisung festgelegt werden. Wenn die Zuweisung der Parameter über Initialwerte erfolgt, dann könnte es beispielhaft so aussehen:

```
(* parameters *)
sControllerParameter : ST_CTRL_TempCtrlParameter :=
(
(* base *)
```

```

tCtrlCycleTime := t#1000ms,
tTaskCycleTime := t#10ms,

fYMin := -100,
fYMax := 100,
tPWMCycleTime := t#100ms,
fYManual := 20,
bFilter := FALSE,
tFilter := t#100ms,
bDeadband := FALSE,
fEDeadband := 1.0, (* deadband *)
fWMin := 15,
fWMax := 60,
fWStartUp := 20.0,
tStartUp := t#160s,
fWVeloPos := 0.01,
fWVeloNeg := 0.01,
bStartUpRamping := FALSE,
fWStartUpVeloPos := 0.1,
fWStartUpVeloNeg := 0.1,
iMode := eCTRL_ControlMode_HEATING,
dwAlarmSupp := 16#FF_FF_FF_FF,
bSelCtrlParameterSet := FALSE,

(* tuning *)
iTuningMode := eCTRL_TuneMode_heating,
fYTuneHeating := 100.0,
fYTuneCooling := -100.0,
fEndTunePercentHeating := 80.0, (* switch to closed loop control when X > 0.8*W *)
fEndTunePercentCooling := -70.0, (* switch to closed loop control when X < 0.2*W *)

iReactionOnFailure := eCTRL_ReactionOnFailure_StopController,
TempLow := -50.0,
TempLowLow := -100.0,
TempHigh := 100.0,
TempHighHigh := 155.0,
TempAbsoluteHigh := 150.0,
TempAbsoluteLow := -95.0,
bEnablePreController := FALSE,
bEnableZones := FALSE,
bEnableCVFilter := FALSE,
iFilterType := eCTRL_FilterType_AVERAGE,
iControllerType := eCTRL_ControllerType_PID
);

```

Die Zuweisung im Code kann in ST folgendermaßen aussehen:

```

sControllerParameter.tPWMCycleTime :=
t#100ms;

```

- **Festlegung der Reglerabtastzeit, der Taskzykluszeit und der PWM Zykluszeit**

Die Abtastzeit des Reglers muss der Strecke angepasst werden. Sie sollte kleiner oder gleich einem Zehntel der dominierenden Streckenzeitkonstanten gewählt werden. Die Taskzykluszeit ist durch die SPS Task festgelegt in der der Reglerbaustein aufgerufen wird. Dieser Wert kann aus der Taskkonfiguration (Plc Control: Ressourcen Taskkonfiguration) abgelesen werden. Die PWM Zykluszeit ist normalerweise gleich der Reglerzykluszeit. Wenn die Taskzykluszeit 10ms und die PWM Zykluszeit (=Reglerabtastzeit) zu 100ms gewählt wird, so stehen insgesamt 10 Stufen (PWM Zykluszeit/Taskzykluszeit) zur Verfügung.

- **TwinCAT Scope parametrieren**

Zur Kontrolle der Ergebnisse sollte auf jeden Fall vom Tuning Vorgang und vom Closed Loop Regelverhalten eine Scope-Aufnahme gemacht werden. Dazu ist das TwinCAT Scope View zu starten und zu parametrieren. Folgende Kanäle sollten aufgezeichnet werden: Sollwert (fW1 oder fW2), Istwert (fX) und analoge Stellgröße (fYAnalog).

- **Abschalten der Alarme während der Inbetriebnahmephase**

Während der Inbetriebnahmephase können die Alarme zeitweilig abgeschaltet werden. Dazu ist in dem Dword dwAlarmSupp eine entsprechende Bitmaske zu setzen. Ist in diesem Dword ein Bit gesetzt, so wird der entsprechende Alarm disabled. Die Belegung der einzelnen Alarme ist [hier \[► 30\]](#) beschrieben.



Nach der Erstinbetriebnahme sollten alle notwendigen Alarme wieder eingeschaltet werden!

- **Starten des Reglers mit Tuning**

Sollen die Reglerparameter mit Hilfe des Tunings ermittelt werden, so muss der Control-Mode auf `eCTRL_MODE_TUNE` eingestellt sein. Es läuft zunächst eine festeingestellte Wartezeit von 20s ab. In dieser Wartezeit wird geprüft, ob die Strecke innerhalb eines $\pm 1^\circ\text{C}$ Bandes bleibt. Sollte das Band verlassen werden, so wird die Wartezeit erneut gestartet. Dann erfolgt eine sprungförmige Anregung der Strecke mit einer Stellgröße von `fYTune`. Darauf reagiert die Strecke mit der Sprungantwort. Solange nicht 80% der Sollgröße erreicht sind, werden die Parameter der Strecke über die Wendetangenten-Methode ermittelt. Aus Sicherheitsgründen wird nach Erreichen von 80% vom Sollwert auf die Regelung im geschlossenen Regelkreis umzuschalten. Sollte die Temperatur zu schnell (ohne ausgeprägten Wendepunkt) die 80% Marke erreichen, so ist der Wert `fYTune` zu reduzieren. Die ermittelten Parameter werden für den PID-Regler benutzt und stehen in einer Struktur am Ausgang des Reglers zur Verfügung.

HINWEIS

Zum Tuning muss die Strecke mindestens einen Sprung von 40°C durchführen. Kleinere Sprünge können zur Ermittlung von falschen Parametern führen!

HINWEIS

Nach dem das Tuning erfolgreich durchgeführt wurde, wird der `eCtrlState` auf `eCTRL_STATE_TUNED` gesetzt. Der Regler geht jetzt in Wartestellung. Nur durch Setzen des Control-Modus auf `eCTRL_MODE_ACTIVE` wird der Closed-Loop-Betrieb mit den geschätzten Parametern aktiviert.

- **Verknüpfen der internen Reglerparameter mit externer Beschaltung**

Die durch das Tuning ermittelten Parameter des Reglers können wieder als externe Parameter auf den Regler aufgeschaltet werden. Das kann notwendig sein, wenn das Tuning nur ein einziges Mal (z.B. nur in der Inbetriebnahmephase) durchgeführt werden soll. Dazu ist die Struktur `sParaControllerInternal` auf den Eingang des Reglers `sParaControllerExternal` zurückzuführen und das Flag `bSelCtrlParameterSet` auf `TRUE` zu setzen.

- **Finetuning**

Die beim Tuning ermittelten Reglerparameter sind auf schnelles Einschwingen mit etwa 10% Überschwingen ausgelegt. Ist kein oder nur sehr geringes Überschwingen erlaubt, so kann mit den folgenden Parametern aus der `ST_ControllerParameter` Struktur ein fine tuning durchgeführt werden. Diese Werte sind als Anhaltswerte anzusehen.

Verhalten	fTuneKp	fTuneTn	fTuneTv	fTuneTd
Schnelles Einschwingen mit einem Überschwingen von 10%-20%	1.2	2.0	0.42	0.25
Langsameres Einschwingen mit geringerem Überschwingen	1.0	2.5	0.42	0.25
Nahezu asymptotische Einschwingen mit sehr geringen Überschwingen	0.5	3.0	1.0	0.25

10 Beispiel

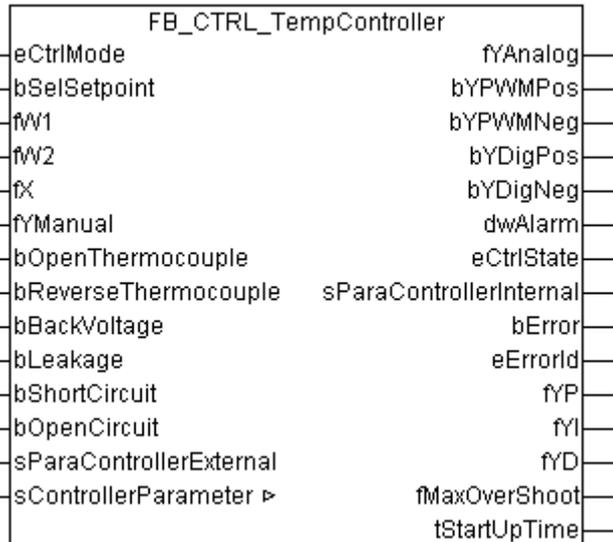
Das Beispielprogramm <https://infosys.beckhoff.com/content/1031/tcplctempcontrol/Resources/11292577291.zip> enthält die Einbindung in ein `MAIN` Programm. Die Regelstrecke wird durch ein PT2 Glied simuliert. Mit dem TwinCAT ScopeView kann eine Aufzeichnung der Werte gemacht werden.

11 PLC-API

11.1 FB_CTRL_TempController

Der Temperaturreglerbaustein FB_CTRL_TempController hat verschiedene Ein- und Ausgänge die im folgenden beschrieben werden. Alle Parameter des Reglers werden über Strukturen an den Regler übergeben. Die Definition der Strukturen und Enums ist [hier \[▶ 37\]](#) zu finden.

Funktionsbaustein



Interface

```

VAR_INPUT
  eCtrlMode : E_CTRL_MODE; (* controller mode: passive, activ,check *)
  bSelSetpoint : BOOL; (* FALSE = setpoint 1, TRUE = setpoint 2*)
  fW1          : FLOAT; (* first setpoint *)
  fW2          : FLOAT; (* second setpoint *)
  fX           : FLOAT; (* actual value *)
  fYManual     : FLOAT; (* manual control value *)
(* alarming inputs *)
  bOpenThermocouple : BOOL; (* thermocouple *)
  bReverseThermocouple : BOOL;
  bBackVoltage      : BOOL;
  bLeakage          : BOOL; (* heating system *)
  bShortCircuit     : BOOL;
  bOpenCircuit      : BOOL;
  sParaControllerExternal : ST_CTRL_ParaController; (* external controller parameter set *)
END_VAR
VAR_IN_OUT
  sControllerParameter : ST_CTRL_TempCtrlParameter; (* controller
parameter set *)
END_VAR
VAR_OUTPUT
  (* control output *)
  fYAnalog : FLOAT; (* control value analog *)
  bYPWMPos : BOOL; (* control value PWM *)
  bYPWMNeg : BOOL; (* control value PWM *)
  bYDigPos : BOOL; (* 3-Point: control value digital positive*)
  bYDigNeg : BOOL; (* 3-Point: control value digital negative*)
  (* alarming *)
  dwAlarm : DWORD; (* max. 32 alarms *)
  (*quality of control*)
  fMaxOverShoot : FLOAT;
  tStartUpTime : TIME;
  (* state *)
  eCtrlState : E_CTRL_STATE := eCTRL_STATE_IDLE;
  (* controller parameter *)
  sParaControllerInternal : ST_CTRL_ParaController;

```

```
(* general errors *)
bError : BOOL;
eErrorId : E_CTRL_ErrorCodes;
END_VAR
```

Eingänge

Name	Einheit	Wertebereich	Beschreibung
eControlMode	1	E_CTRL_MODE	Modeumschaltung.
bSelSetpoint	1	[TRUE,FALSE]	Wählt einen von zwei möglichen Sollwerten aus. Mit FALSE wird der normale Sollwert gewählt, mit TRUE der Standby-Sollwert.
fW1	°C	LREAL	Sollwert.
fW2	°C	LREAL	Standby-Sollwert im Regelfall kleiner als fW1. Mit fSelSetpoint kann zwischen fW1 und fW2 umgeschaltet werden.
fX	°C	LREAL	Istwert. Dieser Wert muss auf LREAL konvertiert werden.
fYManual	-100% - +100%	LREAL	Stellgröße im Handbetrieb.
bOpenThermocouple	1	[TRUE,FALSE]	Mit TRUE ist das Thermoelement offen. Muss von der Hardware gemeldet werden (z.B. KLxxxx).
bReverseThermocouple	1	[TRUE,FALSE]	Mit TRUE wird falsche Polarität des angeschlossenen Thermoelements gemeldet. Muss von der Hardware gemeldet werden.
bBackVoltage	1	[TRUE,FALSE]	Bei TRUE wird eine zu hohe Eingangsspannung am Thermoelement angezeigt. Muss von der Hardware gemeldet werden.
bLeakage	1	[TRUE,FALSE]	Mit TRUE wird ein Leckstrom zu Heizelement detektiert. Muss von der Hardware gemeldet werden.
bShortCircuit	1	[TRUE,FALSE]	Mit TRUE wird ein Kurzschluss im Heizelement detektiert. Muss von der Hardware gemeldet werden.
bOpenCircuit	1	[TRUE,FALSE]	Mit TRUE wird ein offener Stromkreis im Heizelement detektiert. Muss von der Hardware gemeldet werden.
sControllerParameter	keine	Struktur	In dieser Struktur werden generelle Parameter (Abtastzeiten usw.) an den Baustein übergeben.
sParaControllerExternal	keine	Struktur	In dieser Struktur wird ein externer Regler-Parametersatz an den Baustein übergeben.

Ausgänge

Name	Einheit	Wertebereich	Beschreibung
fYAnalog	keine	LREAL	Analoger Stellwert.
bYPWMPos	keine	[TRUE,FALSE]	Boolescher Ausgang pulsweitenmoduliert. Positiv/ Heizbetrieb

Name	Einheit	Wertebereich	Beschreibung
bYPWMNeg	keine	[TRUE,FALSE]	Boolescher Ausgang pulsweitenmoduliert. Negativ/ Kühlobetrieb
bYDigPos	keine	[TRUE,FALSE]	Boolescher Ausgang eines Dreipunktreglers (TRUE Stellgröße 100%, FALSE Stellgröße aus)
bYDigNeg	keine	[TRUE,FALSE]	Boolescher Ausgang eines Dreipunktreglers (TRUE Stellgröße -100%, FALSE Stellgröße aus)
dwAlarm	keine	DWORD	Alarmmeldungen (s. ENUM ...)
fMaxOverShoot	°C	LREAL	max. Überschwingen in °C über/unter Sollwert.
tStartUpTime	TIME	-	Anschwingzeit bis zum erstmaligen Erreichen des Sollwert.
eCtrlState	keine	E_CTRL_STATE	momentaner Status des Reglers (s. ENUM ...)
sParaControllerInternal	keine	Struktur	In dieser Struktur wird der interne (durch das Tuning ermittelte) Regler-Parametersatz zur Verfügung gestellt.
bError	keine	[TRUE,FALSE]	Liegt ein Fehler vor, so ist bError gleich TRUE.
iErrorId	keine	INT	Ist bError gleich TRUE, so zeigt iErrorId einen Fehlercode an (s. ENUM ...)

11.2 Definition der Strukturen

ST_ControllerParameter

```
TYPE ST_CTRL_TempCtrlParameter:
STRUCT
```

```
(*****)

(* general parameters *)
iMode : E_CTRL_ControlMode; (* 1=heating, 2=cooling,3=heating&cooling *)
iReactionOnFailure : E_CTRL_ReactionOnFailure; (* 0=controller
off, 1>manual op, 2=yMin, 3=yMax *)
bSelCtrlParameterSet : BOOL; (* FALSE = internal set, TRUE = external set *)
dwAlarmSupp : DWORD; (* alarm suppression *)
tCtrlCycleTime : TIME; (* controller cycle time *)
tTaskCycleTime : TIME; (* plc task cycle time *)

(*****)

(* tuning parameter *)
iTuningMode : E_CTRL_TuneMode; (* only heating, only cooling,
first heating then cooling or vice versa *)
tTuneStabilisation : TIME := t#20s; (* wait for a stable system*)
fEndTunePercentHeating : FLOAT := 80.0; (* switch to closed loop control when X > 0.8*W *)
fYTuneHeating : FLOAT; (* step change while tuning operation*)
fYStableHeating : FLOAT; (* tuning operation *)
fEndTunePercentCooling : FLOAT := 20.0; (* switch to closed loop control when X < 0.2*W *)
fYTuneCooling : FLOAT; (* step change while tuning operation*)
fYStableCooling : FLOAT; (* tuning operation *)
fScalingFactor : FLOAT := 1.0; (* Scaling factor for KP heating/cooling *)

(*****)

(* setpoint parameters *)
fWMin : FLOAT; (* lower limit *)
fWMax : FLOAT; (* upper limit *)
(* start up *)
bEnableSoftStart : BOOL; (* FALSE = no soft start, TRUE = soft start *)
bEnableRamping : BOOL; (* FALSE = no ramping, TRUE = ramping*)
```

```
fWStartUp : FLOAT; (* soft start plateau setpoint *)
tStartUp : TIME; (* soft start waiting time*)
bStartUpRamping : BOOL; (* enable ramping while start up phase*)
fWStartUpVeloPos : FLOAT; (* max gradient for increasing setpoint in start up phase*)
fWStartUpVeloNeg : FLOAT; (* max gradient for decreasing setpoint in start up phase *)
fWVeloPos : FLOAT; (* max gradient for increasing setpoint*)
fWVeloNeg : FLOAT; (* max gradient for decreasing setpoint*)

(*****)

(* actual value parameters *)
bFilter : BOOL;
tFilter : TIME;

(*****)

(* deadband parameters *)
bDeadband : BOOL;
fEDeadband : FLOAT; (* deadband *)

(*****)

(* control value parameters *)
fYMin : FLOAT; (* lower limit *)
fYMax : FLOAT; (* upper limit *)
fYManual : FLOAT; (* manual operation*)
fYOnFailure : FLOAT; (* control value on failure *)
tPWMCycleTime : TIME; (* PWM: period *)
tPWMMinOffTime : TIME; (* PWM: min off time *)
tPWMMinOnTime : TIME; (* PWM: min on time *)
tPWMMinWaitingTime : TIME; (* PWM: min waiting time *) (* not yet implemented !!!!*)
fYThresholdOff : FLOAT; (* 3-Point: Off threshold *)
fYThresholdOn : FLOAT; (* 3-Point: On threshold *)
nCyclesForSwitchOver : INT := 100;

(*****)

(* controller settings *)
bEnablePreController: BOOL; (* enable precontroller *)
bEnableZones : BOOL; (* enable zone around setpoint with open loop control *)
bEnableCVFilter : BOOL; (* enable filter for CV (type see iFilterType) *)
iFilterType : E_CTRL_FilterType; (* filtertype of CV filter*)
iControllerType : E_CTRL_ControllerType; (* used controller normally PID *)

(*****)

(* min max temperatures *)
TempLow : FLOAT;
TempLowLow : FLOAT;
TempHigh : FLOAT;
TempHighHigh : FLOAT;
TempAbsoluteHigh : FLOAT;
TempAbsoluteLow : FLOAT;

(*****)

(* internal tuning parameters *)
fTuneKp : FLOAT := 1.2;
fTuneTn : FLOAT := 2.0;
fTuneTv : FLOAT := 0.42;
fTuneTd : FLOAT := 0.25;
END_STRUCT
END_TYPE
```

Generelle Parameter

Name	Einheit	Wertebereich	Beschreibung
iMode	keine	INT	Reglerbetriebsmode (1 = heating, 2 = cooling, 3 = heating&cooling) (s.u.) [▶ 29]
iReactionOnFailure	keine	INT	Parametrierbare Reaktion auf Fehler (s.u.) [▶ 28]

Name	Einheit	Wertebereich	Beschreibung
bSelCtrlParameterSet	keine	BOOL	TRUE = externer Parametersatz, FALSE = interner Parametersatz (durch Tuning ermittelt)
dwAlarmSupp	keine	DWORD	Maskiert die Alarmer aus (s.u.) [► 30]
tCtrlCycleTime	s	TIME	Abtastzeit des Reglers. Der Regler errechnet im Takt dieser Abtastzeit neue Werte für die Stellgröße.
tTaskCycleTime	s	TIME	Zykluszeit der Task. Der FB wird im Takt der Task aufgerufen.

Tuning Parameter

Name	Einheit	Wertebereich	Beschreibung
iTuningMode	K	E_CTRL_TuneMode	Festlegung der Tuning Reihenfolge (s.u.)
tTuneStabilisation	s	TIME	Wartezeit bis Strecke stabil für Tuning Vorgang.
fEndTunePercentHeating	%	(L)REAL	Prozentwert von Sollgröße ab der auf Closed Loop Control umgeschaltet wird.
fYTuneCooling	K	(L)REAL	Stellgrößensprung beim Tuning.
fYStableCooling	K	(L)REAL	Stellgröße beim Umschalten auf Tuning beim Kühlen.
fScalingFactor	keine	(L)REAL	Skalierungsfaktor für Parameterumschaltung, wenn kein Tuning für das Kühlen durchgeführt wird.

Sollwert Parameter

Name	Einheit	Wertebereich	Beschreibung
fWMin	K	(L)REAL	Minimale Sollgröße.
fWMax	K	(L)REAL	Maximale Sollgröße.
bEnableSoftStart	keine	BOOL	FALSE = kein soft start, TRUE = soft start
bEnableRamping	keine	BOOL	FALSE = keine Verrampung, TRUE = Verrampung
fWStartUp	K	(L)REAL	Sollgröße beim Anfahren.
tStartUp	s	TIME	Zeit mit Sollwert fWStartUp.
bStartUpRamping	keine	[TRUE,FALSE]	Schaltet Verrampung während der StartUp-Phase ein.
fWStartUpVeloPos	K/s	(L)REAL	Anstiegsgeschwindigkeit (Rampe) während der StartUp-Phase.
fWStartUpVeloNeg	K/s	(L)REAL	Abfallgeschwindigkeit (Rampe) während der StartUp-Phase
fWVeloPos	K/s	(L)REAL	Anstiegsgeschwindigkeit (Rampe).
fWVeloNeg	K/s	(L)REAL	Abfallgeschwindigkeit (Rampe).

Istgrößen Parameter

Name	Einheit	Wertebereich	Beschreibung
tFilter	s	TIME	Filterzeitkonstante des Istwertfilters (P-T1 Filter 1. Ordnung)
bFilter	keine	[TRUE,FALSE]	Mit TRUE wird der Istwertfilter aktiviert.

Totband Parameter

Name	Einheit	Wertebereich	Beschreibung
bDeadband	keine	[TRUE,FALSE]	TRUE = Totband ein, FALSE = Totband aus
fEDeadband	K	(L)REAL	Größe des Totbandes in Grad.

Stellgrößen Parameter

Name	Einheit	Wertebereich	Beschreibung
fYMin	keine	(L)REAL	Minimalwert der Stellgröße.
fYMax	keine	(L)REAL	Maximalwert der Stellgröße.
fYManual	keine	(L)REAL	Stellgröße im Handbetrieb.
fYOnFailure	keine	(L)REAL	Stellgröße im Fehlerfall (parametrierbar).
tPWMCycleTime	s	TIME	Zykluszeit des PWM Signals.
tPWMMinOffTime	s	TIME	PWM: minimale Ausschaltzeit
tPWMMinOnTime	s	TIME	PWM: minimale Einschaltzeit
tPWMWaitingTime	s	TIME	PWM: Wartezeit beim Umschalten von Heizen nach Kühlen
fYThresholdOff	%	(L)REAL	3-Punkt: Ausschaltschwelle
fYThresholdOn	%	(L)REAL	3-Punkt: Einschaltschwelle
nCyclesForSwitchOver	keine	INT	Anzahl Zyklen für Übergang von einem Parametersatz zum anderen

Regler Parameter

Name	Einheit	Wertebereich	Beschreibung
bEnablePreController	keine	[TRUE,FALSE]	Schaltet Vorregler ein.
bEnableZones	keine	[TRUE,FALSE]	Schaltet OpenLoop Verhalten bis nahe Sollwert ein.
bEnableCVFilter	keine	[TRUE,FALSE]	Schaltet Stellgrößenfilter hinter Hauptregler ein.
iFilterType	keine	ENUM	Wahl eines Filtertypes für den Stellgrößenfilter hinter dem Hauptregler (s.u.) [▶ 29].
iControllerType	keine	ENUM	Wahl eines Regelalgorithmus (s.u.) [▶ 29]

Alarming Parameter

Name	Einheit	Wertebereich	Beschreibung
TempLow	K	(L)REAL	Relativer unterer Temperaturgrenzwert im ersten Band.
TempLowLow	K	(L)REAL	Relativer unterer Temperaturgrenzwert im zweiten Band.
TempHigh	K	(L)REAL	Relativer oberer Temperaturgrenzwert im ersten Band.
TempHighHigh	K	(L)REAL	Relativer oberer Temperaturgrenzwert im zweiten Band.
TempAbsoluteHigh	K	(L)REAL	Absoluter oberer Temperaturgrenzwert.
TempAbsoluteLow	K	(L)REAL	Absoluter unterer Temperaturgrenzwert.

Experten Parameter

Name	Einheit	Wertebereich	Beschreibung
fTuneKp	keine	(L)REAL	FineTuning Parameter für den PID Regler (nur für Experten)
fTuneTn	keine	(L)REAL	FineTuning Parameter für den PID Regler (nur für Experten)
fTuneTv	keine	(L)REAL	FineTuning Parameter für den PID Regler (nur für Experten)
fTuneTd	keine	(L)REAL	FineTuning Parameter für den PID Regler (nur für Experten)

Beschreibung

ST_CTRL_ParaController

```

TYPE ST_CTRL_ParaController :
STRUCT
  (* Controller parameter set - heating *)
  KpHeat : FLOAT;
  TnHeat : TIME;
  TvHeat : TIME;
  TdHeat : TIME;
  (* Controller parameter set - cooling *)
  KpCool : FLOAT;
  TnCool : TIME;
  TvCool : TIME;
  TdCool : TIME;
END_STRUCT
END_TYPE
    
```

Beschreibung

Name	Einheit	Wertebereich	Beschreibung
KpHeat	keine	(L)REAL	Verstärkungsfaktor für Hauptregler.
TnHeat	s	TIME	Nachstellzeit für Hauptregler (I-Anteil).
TvHeat	s	TIME	Vorhaltzeit für Hauptregler (D-Anteil).
TdHeat	s	TIME	Dämpfungszeit für Hauptregler.
KpCool	keine	(L)REAL	Verstärkungsfaktor für Hauptregler.
TnCool	s	TIME	Nachstellzeit für Hauptregler (I-Anteil).
TvCool	s	TIME	Vorhaltzeit für Hauptregler (D-Anteil).
TdCool	s	TIME	Dämpfungszeit für Hauptregler.

E_CTRL_ERRORCODES:

```

TYPE E_CTRL_ERRORCODES :
(
  eCTRL_ERROR_NOERROR := 0, (* no error *)
  eCTRL_ERROR_INVALIDTASKCYCLETIME := 1, (* invalid task cycle time *)
  eCTRL_ERROR_INVALIDCTRLCYCLETIME := 2, (* invalid ctrl cycle time *)
  eCTRL_ERROR_INVALIDPARAM := 3, (* invalid parameter *)
  eCTRL_ERROR_INVALIDPARAM_Tv := 4, (* invalid parameter *)
  eCTRL_ERROR_INVALIDPARAM_Td := 5, (* invalid parameter *)
  eCTRL_ERROR_INVALIDPARAM_Tn := 6, (* invalid parameter *)
  eCTRL_ERROR_INVALIDPARAM_Ti := 7, (* invalid parameter *)
  eCTRL_ERROR_INVALIDPARAM_fHystereisisRange := 8, (* invalid parameter *)
  eCTRL_ERROR_INVALIDPARAM_fPosOutOn_Off := 9, (* invalid parameter *)
  eCTRL_ERROR_INVALIDPARAM_fNegOutOn_Off := 10, (* invalid parameter *)
  eCTRL_ERROR_INVALIDPARAM_TableDescription := 11, (* invalid parameter *)
  eCTRL_ERROR_INVALIDPARAM_TableData := 12, (* invalid parameter *)
  eCTRL_ERROR_INVALIDPARAM_DataTableADR := 13, (* invalid parameter *)
  eCTRL_ERROR_INVALIDPARAM_T0 := 14, (* invalid parameter *)
  eCTRL_ERROR_INVALIDPARAM_T1 := 15, (* invalid parameter *)
  eCTRL_ERROR_INVALIDPARAM_T2 := 16, (* invalid parameter *)
  eCTRL_ERROR_INVALIDPARAM_T3 := 17, (* invalid parameter *)
  eCTRL_ERROR_INVALIDPARAM_Theta := 18, (* invalid parameter *)
  eCTRL_ERROR_INVALIDPARAM_nOrder := 19, (* invalid parameter *)
)
    
```

```

eCTRL_ERROR_INVALIDPARAM_Tt := 20, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_Tu := 21, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_Tg := 22, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_infinite_slope := 23, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_fMaxIsLessThanfMin := 24, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_fOutMaxLimitIsLessThanfOutMinLimit := 25, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_fOuterWindow := 26, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_fInnerWindow := 27, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_fOuterWindowIsLessThanfInnerWindow := 28, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_fDeadBandInput := 29, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_fDeadBandOutput := 30, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_PWM_Cycletime := 31, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_no_Parameterset := 32, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_fOutOn := 33, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_fOutOff := 34, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_fGain := 35, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_fOffset := 36, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_MODE_NOT_SUPPORTED := 37, (* invalid mode: mode not supported *)
eCTRL_ERROR_INVALIDPARAM_Tv_heating := 38, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_Td_heating := 39, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_Tn_heating := 40, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_Tv_cooling := 41, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_Td_cooling := 42, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_Tn_cooling := 43, (* invalid parameter *)
eCTRL_ERROR_RANGE_NOT_SUPPORTED := 44, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_nParameterChangeCycleTicks := 45, (* invalid parameter *)
eCTRL_ERROR_ParameterEstimationFailed := 46, (* invalid parameter *)
eCTRL_ERROR_NoiseLevelToHigh := 47, (* invalid parameter *)
eCTRL_ERROR_INTERNAL_ERROR_0 := 48, (* internal error *)
eCTRL_ERROR_INTERNAL_ERROR_1 := 49, (* internal error *)
eCTRL_ERROR_INTERNAL_ERROR_2 := 50, (* internal error *)
eCTRL_ERROR_INTERNAL_ERROR_3 := 51, (* internal error *)
eCTRL_ERROR_INTERNAL_ERROR_4 := 52, (* internal error *)
eCTRL_ERROR_INTERNAL_ERROR_5 := 53, (* internal error *)
eCTRL_ERROR_INTERNAL_ERROR_6 := 54, (* internal error *)
eCTRL_ERROR_INTERNAL_ERROR_7 := 55, (* internal error *)
eCTRL_ERROR_INTERNAL_ERROR_8 := 56, (* internal error *)
eCTRL_ERROR_INTERNAL_ERROR_9 := 57, (* internal error *)
eCTRL_ERROR_INVALIDPARAM_WorkArrayADR := 58, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_tOnTiime := 59, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_tOffTiime := 60, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_nMaxMovingPulses := 61, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_nAdditionalPulsesAtLimits := 62, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_fCtrlOutMax_Min := 63, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_fDeltaMax := 64, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_tMovingTime := 65, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_tDeadTime := 66, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_tAdditionalMoveTimeAtLimits := 67, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_fThreshold := 68, (* invalid parameter *)
eCTRL_ERROR_MEMCPY := 69, (* MEMCPY failed *)
eCTRL_ERROR_MEMSET := 70, (* MEMSET failed *)
eCTRL_ERROR_INVALIDPARAM_nNumberOfColumns := 71, (* invalid parameter *)
eCTRL_ERROR_FileClose := 72, (* File Close failed *)
eCTRL_ERROR_FileOpen := 73, (* File Open failed *)
eCTRL_ERROR_FileWrite := 74, (* File Write failed *)
eCTRL_ERROR_INVALIDPARAM_fVeloNeg := 75, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_fVeloPos := 76, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_DeadBandInput := 77, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_DeadBandOutput := 78, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_CycleDuration := 79, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_tStart := 80, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_StepHeigthTuningToLow := 81, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_fMinLimitIsLessThanZero := 82, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_fMaxLimitIsGreaterThan100 := 83, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_fStepSize := 84, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_fOkRangeIsLessOrEqualZero := 85, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_fForceRangeIsLessOrEqualfOkRange := 86, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_PWMPeriod := 87, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_tMinimumPulseTime := 88, (* invalid parameter *)
eCTRL_ERROR_FileDelete := 89, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_nNumberOfPwmOutputs := 90, (* File Delete failed *)
eCTRL_ERROR_INVALIDPARAM_nPwmInputArray_SIZEOF := 91, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_nPwmOutputArray_SIZEOF := 92, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_nPwmWaitTimesConfig_SIZEOF := 93, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_nPwmInternalData_SIZEOF := 94, (* invalid parameter *)
eCTRL_ERROR_SIZEOF := 95, (* SIZEOF failed *)
eCTRL_ERROR_INVALIDPARAM_nOrderOfTheTransferfunction := 96, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_nNumeratorArray_SIZEOF := 97, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_nDenominatorArray_SIZEOF := 98, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_a_n_IsZero := 99, (* invalid parameter *)

```

```
eCTRL_ERROR_INVALIDPARAM_WorkArraySIZEOF := 100, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_MOVINGRANGE_MIN_MAX := 101, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_MOVINGTIME := 102, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_DEADTIME := 103, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_fMinLimitIsGreaterThanfMaxLimit := 104, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_DataTableSIZEOF := 105, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_OutputVectorDescription := 106, (* invalid parameter *)
eCTRL_ERROR_TaskCycleTimeChanged := 107, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_nMinMovingPulses := 108, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_fAcceleration := 109, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_fDeceleration := 110, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_StartAndTargetPos := 111, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_fVelocity := 112, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_fTargetPos := 113, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_fStartPos := 114, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_fMovingLength := 115, (* invalid parameter *)
eCTRL_ERROR_NT_GetTime := 116, (* internal error NT_GetTime *)
eCTRL_ERROR_INVALIDPARAM_No3PhaseSolutionPossible := 117, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_fStartVelo := 118, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_fTargetVelo := 119, (* invalid parameter *)
eCTRL_ERROR_INVALID NEW PARAMETER TYPE := 120 (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_fBaseTime := 121, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_nOrderOfTheTransferfunction_SIZEOF := 122, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_nFilterOrder := 124, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_nCoefficientsArray_a_SIZEOF := 125, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_nCoefficientsArray_b_SIZEOF := 126, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_nDigitalFiterData_SIZEOF := 127, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_nLogBuffer_SIZEOF := 128, (* invalid parameter *)
eCTRL_ERROR_LogBufferOverflow := 129, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_nLogBuffer_ADR := 130, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_nCoefficientsArray_a_ADR := 131, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_nCoefficientsArray_b_ADR := 132, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_nPwmInputArray_ADR := 133, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_nPwmOutputArray_ADR := 134, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_nPwmWaitTimesConfig_ADR := 135, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_nPwmInternalData_ADR := 136, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_nDigitalFiterData_ADR := 137, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_nNumeratorArray_ADR := 138, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_nDenominatorArray_ADR := 139, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_nTransferfunction1Data_ADR := 140, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_nTransferfunction2Data_ADR := 141, (* invalid parameter *)
eCTRL_ERROR_FileSeek := 142, (* internal error FB_FileSeek *)
eCTRL_ERROR_INVALIDPARAM_AmbientTempMaxIsLessThanAmbientTempMin := 143, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_ForerunTempMaxIsLessThanForerunTempMin := 144, (* invalid parameter *)
eCTRL_ERROR_INVALIDLOGCYCLETIME := 145, (* invalid parameter *)
eCTRL_ERROR_INVALIDVERSION_TcControllerToolbox := 146,
eCTRL_ERROR_INVALIDPARAM_Bandwidth := 147, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_NotchFrequency := 148, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_DampingCoefficient := 149, (* invalid parameter *)
eCTRL_ERROR_INVALIDPARAM_fKpIsLessThanZero := 150 (* invalid parameter *)
);
END_TYPE
```

ENUM: E_CTRL_ReactionOnFailure

Name	Beschreibung
eCTRL_ReactionOnFailure_NoFailure	Kein Fehler.
eCTRL_ReactionOnFailure_StopController	Wenn Fehler (Alarm), dann stoppe Regler.
eCTRL_ReactionOnFailure_SetManMode	Wenn Fehler (Alarm), dann schalte Regler in Handbetrieb.
eCTRL_ReactionOnFailure_SetYMax	Wenn Fehler (Alarm), dann setze Stellgröße auf maximalen Wert.
eCTRL_ReactionOnFailure_SetYMin	Wenn Fehler (Alarm), dann setze Stellgröße auf minimalen Wert.
eCTRL_ReactionOnFailure_SetYMean	Wenn Fehler (Alarm), dann setze Stellgröße auf Mittelwert (in Planung).

ENUM: E_CTRL_ControllerStateInternal

Name	Beschreibung
E_CTRL_ControllerStateInternalHeating	intern

Name	Beschreibung
E_CTRL_ControllerStateInternalCooling	intern

ENUM: E_CTRL_ControlMode

Name	Beschreibung
eCTRL_ControlMode_HEATING	Nur Heizen.
eCTRL_ControlMode_COOLING	Nur Kühlen.
eCTRL_ControlMode_HEATING_COOLING	Heizen und Kühlen.

E_CTRL_STATE :

```

TYPE E_CTRL_STATE :
(
  eCTRL_STATE_IDLE           := 0, (* state idle *)
  eCTRL_STATE_PASSIVE        := 1, (* state passive *)
  eCTRL_STATE_ACTIVE         := 2, (* state active *)
  eCTRL_STATE_RESET          := 3, (* state reset *)
  eCTRL_STATE_MANUAL         := 4, (* state manual *)
  eCTRL_STATE_TUNING         := 5, (* state tuning *)
  eCTRL_STATE_TUNED          := 6, (* state tuning ready - tuned *)
  eCTRL_STATE_SELFTEST       := 7, (* state selftest *)
  eCTRL_STATE_ERROR          := 8, (* state error *)
  eCTRL_STATE_SYNC_MOVEMENT := 9  (* state synchronizing movement *)
);
END_TYPE
    
```

ENUM: E_CTRL_STATE_TUNIG

Name	Beschreibung
eCTRL_STATE_TUNING_INIT	Tuning: Initialisierung
eCTRL_STATE_TUNING_IDLE	Tuning: Warten auf stabiles Eingangssignal (Regelgröße)
eCTRL_STATE_TUNING_PULSE	Tuning: Anregung durch kurzen Puls (in Planung)
eCTRL_STATE_TUNING_STEP	Tuning: Anregung durch Sprung (Wendetangentenmethode)
eCTRL_STATE_TUNING_READY	Tuning: Ermittlung der Parameter, Finalisierung
eCTRL_STATE_TUNING_ERROR	Tuning: Fehler beim Tuning aufgetreten

ENUM: E_CTRL_TuneMode

Name	Beschreibung
eCTRL_TuneMode_HEATING	Tuning: nur Heizen
eCTRL_TuneMode_COOLING	Tuning: nur Kühlen
eCTRL_TuneMode_HEATING_COOLING	Tuning: erst Heizen, dann Kühlen
eCTRL_TuneMode_COOLING_HEATING	Tuning: erst Kühlen, dann Heizen
eCTRL_TuneMode_OSCILLATION	Tuning: on-the-fly Parameterschätzung durch Schwingungsanregung (in Planung)

ENUM: E_CTRL_FilterType

Name	Beschreibung
eCTRL_FilterType_FIRSTORDER	Filter erster Ordnung.
eCTRL_FilterType_AVERAGE	Mittelwertfilter.

ENUM: E_CTRL_ControllerType

Name	Beschreibung
eCTRL_ControllerType_PID	Standard-PID Regelalgorithmus.

Name	Beschreibung
eCTRL_ControllerType_PI	Standard-PI Regelalgorithmus.
eCTRL_ControllerType_PID_Pre	Standard-PID Regelalgorithmus mit Vorregler (in Planung).
eCTRL_ControllerType_PID2	serieller PID Regelalgorithmus (in Planung)

Bit-Masken für Alarme

Name	Maske	Beschreibung
nAlarmOpenThermocouple	2#0000_0000_0000_0000_0000_0000_0000_0001	Hardware: offener Temperaturfühler
nAlarmReverseThermocouple	2#0000_0000_0000_0000_0000_0000_0000_0010	Hardware: verpolter Temperaturfühler
nAlarmBackVoltage	2#0000_0000_0000_0000_0000_0000_0000_0100	Hardware: zu hohe Spannung am Temperaturfühler
nAlarmLeakageCurrent	2#0000_0000_0000_0000_0000_0000_0000_1000	Hardware: Leckstrom gemessen
nAlarmShortCircuit	2#0000_0000_0000_0000_0000_0000_0000_0001	Hardware: Kurzschluss
nAlarmOpenCircuit	2#0000_0000_0000_0000_0000_0000_0000_0010	Hardware: kein Strom
nAlarmLimitLow	2#0000_0000_0000_0000_0000_0000_0001_0000	Software: untere erste relative Temperatur unterschritten
nAlarmLimitLowLow	2#0000_0000_0000_0000_0000_0000_0010_0000	Software: untere zweite relative Temperatur unterschritten
nAlarmLimitHigh	2#0000_0000_0000_0000_0000_0000_0100_0000	Software: obere erste relative Temperatur überschritten
nAlarmLimitHighHigh	2#0000_0000_0000_0000_0000_0000_0000_1000	Software: obere zweite relative Temperatur überschritten
nAlarmAbsoluteHigh	2#0000_0000_0000_0000_0000_0001_0000_0000	Software: obere absolute Temperatur überschritten
nAlarmAbsoluteLow	2#0000_0000_0000_0000_0000_0010_0000_0000	Software: untere absolute Temperatur unterschritten

11.3 Vorherige Implementierung

Folgende Schritte müssen durchgeführt werden:

- Die Regler-Bibliothek ist über den Bibliotheksmanager in das Projekt einzufügen.

Im Bibliotheksmanager ist die TcTempCtrl.lib einzufügen.

- Mindestens eine Instanz des Reglers ist zu programmieren.

Eine Instanz des Reglerbausteins FB_TempController ist anzulegen. Ausserdem wird eine Instanz der Struktur ST_ControllerParameter anzulegen.

- Die notwendige äußere Beschaltung durchführen.

Name		Beschreibung
bOn	Beschaltung notwendig	Schaltet den Regler mit TRUE ein. Kann fest auf TRUE gelegt werden, wenn der Regler immer eingeschaltet sein soll.
bInit	Beschaltung notwendig	Initialisierungsflag, das genau für den ersten Zyklus des Regleraufrufes anstehen muss (TRUE).

Name		Beschreibung
bTune	Beschaltung notwendig, wenn Tuning eingesetzt werden soll	Schaltet mit der steigenden Flanke das Selftuning ein. Wird während des Selftuning-Vorgangs auf FALSE geschaltet, so wird das Selftuning abgebrochen und mit den alten Parametern (sofern vorhanden) weitergemacht.
bManual	Beschaltung optional	Schaltet mit TRUE auf Handbetrieb um. Wird da Signal wieder FALSE, so wird in den Automatikmode zurückgeschaltet.
bSelSetpoint	Beschaltung optional	Wählt einen von zwei möglichen Sollwerten aus. Mit FALSE wird der normale Sollwert gewählt, mit TRUE der Standby-Sollwert.
bSelCtrlParameterSet	Beschaltung optional	Wählt einen von zwei Parametersätzen aus. Mit FALSE wird der interne (ermittelte) Parametersatz benutzt, mit TRUE ein von aussen aufgeschalteter.
bEnableSoftStart	Beschaltung optional	Mit TRUE wird mit der Anfahrtschaltung (Softstart) hochgefahren.
bEnableRamping	Beschaltung optional	Mit TRUE wird jeder Sollwertsprung gerampt ausgeführt.
fW1	Beschaltung notwendig	Sollwert.
fW2	Beschaltung optional	Standby-Sollwert im Regelfall kleiner als fW1. Mit fSelSetpoint kann zwischen fW1 und fW2 umgeschaltet werden.
fX	Beschaltung notwendig	Istwert. Dieser Wert muss auf LREAL konvertiert werden.
bOpenThermocouple	Beschaltung optional	Mit TRUE ist das Thermoelement offen. Muss von der Hardware gemeldet werden (z.B. KLxxxx).
bReverseThermocouple	Beschaltung optional	Mit TRUE wird falsche Polarität des angeschlossenen Thermoelements gemeldet. Muss von der Hardware gemeldet werden.
bBackVoltage	Beschaltung optional	Bei TRUE wird eine zu hohe Eingangsspannung am Thermoelement angezeigt. Muss von der Hardware gemeldet werden.
bLeakage	Beschaltung optional	Mit TRUE wird ein Leckstrom zu Heizelement detektiert. Muss von der Hardware gemeldet werden.
bShortCircuit	Beschaltung optional	Mit TRUE wird ein Kurzschluss im Heizelement detektiert. Muss von der Hardware gemeldet werden.
bOpenCircuit	Beschaltung optional	Mit TRUE wird ein offener Stromkreis im Heizelement detektiert. Muss von der Hardware gemeldet werden.
sControllerParameter	Beschaltung notwendig	In dieser Struktur werden generelle Parameter (Abtastzeiten usw.) an den Baustein übergeben.
sParaControllerExternal	Beschaltung optional	In dieser Struktur wird ein externer Regler-Parametersatz an den Baustein übergeben.
sLogData	Beschaltung optional	In dieser Struktur werden Parameter (Dateinamen usw.) für das Logging an den Baustein übergeben.

• **Die notwendige Parametrierung des Reglers über die Struktur durchführen.**

Die Parameter können über Initialwerte oder über Zuweisung festgelegt werden. Sollen Initialwerte verwendet werden, so sieht die Instanz der Struktur mit Initialwerten folgendermassen aus:

```
sControllerParameter : ST_ControllerParameter :=
(
(*****)
(* general parameters *)
iMode := CTRLMODE_HEATING, (* 1=heating, 2=cooling, 3=heating&cooling *)
iReactionOnFailure := TC_OnFailureStopController, (* controller off or manual op or yMin or yMax *)
fYTune := 100, (* step change while tuning operation *)
fYStable := 0.0, (* tuning operation *)
dwAlarmSupp := 16#ff_ff_ff_ff, (* alarm suppression *)
```

```

tCtrlCycleTime := t#100ms, (* controller cycle time *)
tTaskCycleTime := t#10ms, (* plc task cycle time *)
(*****)
(* setpoint parameters *)
fWMin := 0.0, (* lower limit *)
fWMax := 200, (* upper limit *)

(* start up optional *)
fWStartUp := 0.0, (* soft start plateau setpoint *)
tStartUp := t#0s, (* soft start waiting time*)
bStartUpRamping := FALSE, (* enable ramping while start up phase *)
fWStartUpVeloPos := 0.0, (* max gradient for increasing setpoint in start up phase*)
fWStartUpVeloNeg := 0.0, (* max gradient for decreasing setpoint in start up phase *)

fWVeloPos := 0.0, (* max gradient for increasing setpoint *)
fWVeloNeg := 0.0, (* max gradient for decreasing setpoint *)
(*****)
(* actual value parameters *)
bFilter := FALSE,
tFilter := t#0s,
(*****)
(* control value parameters *)
fYMin := 0, (* lower limit *)
fYMax := 100, (* upper limit *)
fYManual := 0.0, (* manual operation*)
fYOnFailure := 0.0, (* control value on failure *)
tPWMCycleTime := t#100ms, (* PWM *)
(*****)
(* controller settings *)
bEnablePreController := FALSE, (* enable precontroller *)
bEnableZones := FALSE, (* enable zone around setpoint with open loop control *)
bEnableCVFilter := FALSE, (* enable filter for CV (type see iFilterType) *)
iFilterType := E_FilterType_FIRSTORDER, (* filtertype of CV filter *)
iControllerType := E_ControllerType_PID, (* used controller normally PID *)
(*****)
(* min max temperatures *)
TempLow := 10,
TempLowLow := 20,
TempHigh := 10,
TempHighHigh := 20,
TempAbsoluteHigh := 200,
TempAbsoluteLow := 0,
(*****)
(* internal tuning parameters *)
fTuneKp := 1.2,
fTuneTn := 2.0,
fTuneTv := 0.42,
fTuneTd := 0.25
);

```

Die markierten Parameter sind optional und müssen nur im Bedarfsfall initialisiert werden.

Die Zuweisung im Code kann in ST folgendermassen aussehen:

```

sControllerParameter.tPWMCycleTime :=
t#100ms;

```

- **Festlegung der Reglerabtastzeit, der Taskzykluszeit und der PWM Zykluszeit**

Die Abtastzeit des Reglers muss der Strecke angepasst werden. Sie sollte kleiner oder gleich einem zehntel der dominierenden Streckenzeitkonstanten gewählt werden. Die Taskzykluszeit ist durch die SPS Task festgelegt in der der Reglerbaustein aufgerufen wird. Dieser Wert kann aus der Taskkonfiguration (Plc Control: Ressourcen Taskkonfiguration) abgelesen werden. Die PWM Zykluszeit ist normalerweise gleich der Reglerzykluszeit. Wenn die Taskzykluszeit 10ms und die PWM Zykluszeit (=Reglerabtastzeit) zu 100ms gewählt wird, so stehen insgesamt 10 Stufen (PWM Zykluszeit/Taskzykluszeit) zur Verfügung.

- **TwinCAT Scope parametrieren**

Zur Kontrolle der Ergebnisse sollte auf jeden Fall vom Tuning Vorgang und vom Closed Loop Regelverhalten eine Scope-Aufnahme gemacht werden. Dazu ist das TwinCAT Scope View zu starten und zu parametrieren. Folgende Kanäle sollten aufgezeichnet werden: Sollwert (fW1 oder fW2), Istwert (fX) und analoge Stellgröße (fYAnalog).

- **Abschalten der Alarme während der Inbetriebnahmephase**

Während der Inbetriebnahmephase können die Alarmer zeitweilig abgeschaltet werden. Dazu ist in dem Dword dwAlarmSupp eine entsprechende Bitmaske zu setzen. Ist in diesem Dword ein Bit gesetzt, so wird der entsprechende Alarm disabled. Die Belegung der einzelnen Alarmer ist [hier \[► 41\]](#) beschrieben.

i Nach der Erstinbetriebnahme sollten alle notwendigen Alarmer wieder eingeschaltet werden!

• **Starten des Reglers mit Tuning**

Sollen die Reglerparameter mit Hilfe des Tuning ermittelt werden, so müssen die Eingangs bOn und bTune auf TRUE sein. Es läuft zunächst eine festeingestellte Wartezeit von 20s ab. In dieser Wartezeit wird geprüft, ob die Strecke innerhalb eines $\pm 1^{\circ}\text{C}$ Bandes bleibt. Sollte das Band verlassen werden, so wird die Wartezeit erneut gestartet. Dann erfolgt eine sprungförmige Anregung der Strecke mit einer Stellgröße von fYTune. Darauf reagiert die Strecke mit der Sprungantwort. Solange nicht 80% der Sollgröße erreicht sind, werden die Parameter der Strecke über die Wendetangenten-Methode ermittelt. Aus Sicherheitsgründen wird nach Erreichen von 80% vom Sollwert auf die Regelung im geschlossenen Regelkreis umzuschalten. Sollte die Temperatur zu schnell (ohne ausgeprägten Wendepunkt) die 80% Marke erreichen, so ist der Wert fYTune zu reduzieren. Die ermittelten Parameter werden für den PID-Regler benutzt und stehen in einer Struktur am Ausgang des Reglers zur Verfügung.

i Zum Tuning muss die Strecke mindestens einen Sprung von 40°C durchführen. Kleinere Sprünge können zur Ermittlung von falschen Parametern führen!

• **Verknüpfen der internen Reglerparameter mit externer Beschaltung**

Die durch das Tuning ermittelten Parameter des Regler können wieder als externe Parameter auf den Regler aufgeschaltet werden. Das kann notwendig sein, wenn das Tuning nur ein einziges Mal (z.B. nur in der Inbetriebnahmephase) durchgeführt werden soll. Dazu ist die Struktur sParaControllerInternal auf den Eingang des Reglers sParaControllerExternal zurückzuführen und das Flag bSelCtrlParameterSet auf TRUE zu setzen.

• **Finetuning**

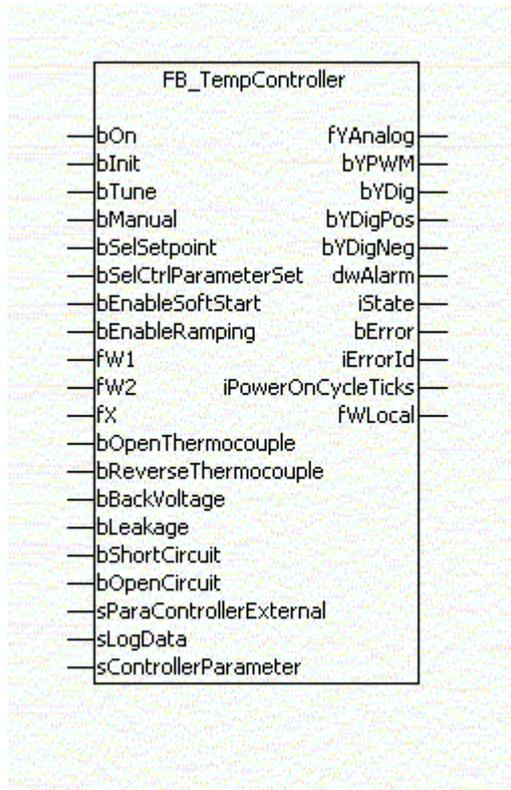
Die beim Tuning ermittelten Reglerparameter sind auf schnelles Einschwingen mit etwa 10% Überschwingen ausgelegt. Ist kein oder nur sehr geringes Überschwingen erlaubt, so kann mit den folgenden Parametern aus der ST_ControllerParameter Struktur ein fine tuning durchgeführt werden. Diese Werte sind als Anhaltswerte anzusehen.

Verhalten	fTuneKp	fTuneTn	fTuneTv	fTuneTd
Schnelles Einschwingen mit einem Überschwingen von 10%-20%	1.2	2.0	0.42	0.25
Langsameres Einschwingen mit geringerem Überschwingen	1.0	2.5	0.42	0.25
Nahezu asymptotische Einschwingen mit sehr geringen Überschwingen	0.5	3.0	1.0	0.25

11.3.1 FB_TempController

Der Temperaturreglerbaustein FB_TempController hat verschiedene Ein- und Ausgänge die im folgenden beschrieben werden. Alle Parameter des Reglers werden über Strukturen an den Regler übergeben. Die Definition der Strukturen und Enums ist [hier \[► 37\]](#) zu finden.

Funktionsbaustein



Interface

```

VAR_INPUT
  bOn : BOOL; (* start closed loop operating *)
  bInit : BOOL; (* init controller *)
  bTune : BOOL; (* start self tuning *)
  bManual : BOOL; (* manual operation *)
  bSelSetpoint : BOOL; (* FALSE = setpoint 1, TRUE = setpoint 2
*)
  bSelCtrlParameterSet : BOOL; (* FALSE = internal set, TRUE =
external set *)
  bEnableSoftStart : BOOL; (* FALSE = no soft start, TRUE = soft
start *)
  bEnableRamping : BOOL; (* FALSE = no ramping, TRUE = ramping
*)
  fW1 : LREAL; (* first setpoint *)
  fW2 : LREAL; (* second setpoint *)
  fX : LREAL; (* actual value *)
  (* alarming inputs *)
  bOpenThermocouple : BOOL; (* thermocouple *)
  bReverseThermocouple : BOOL;
  bBackVoltage : BOOL;
  bLeakage : BOOL; (* heating system *)
  bShortCircuit : BOOL;
  bOpenCircuit : BOOL;
  sParaControllerExternal : ST_ParaController; (* external
controller parameter set *)
  (* logging *)
  sLogData : ST_LogData := (bLog := FALSE, strLogFileName :=
'', strLogString := '' );
END_VAR
VAR_IN_OUT
  sControllerParameter : ST_ControllerParameter; (* parameters
*)
END_VAR
VAR_OUTPUT
  fYAnalog : LREAL; (* control value analog *)
  bYPWM : BOOL; (* control value PWM *)
  bYDig : BOOL; (* 2-Point *)
  bYDigPos : BOOL; (* 3-Point: control value digital positive
*)
  bYDigNeg : BOOL; (* 3-Point: control value digital
negative*)
  (* alarming *)

```

```

dwAlarm : DWORD; (* max. 32 alarms *)
(* state *)
iState : States := TC_STATE_IDLE;
(* controller parameter *)
sParaControllerInternal : ST_ParaController;
(* general errors *)
bError : BOOL;
iErrorId : ErrorCodes;
END_VAR
    
```

Eingänge

Name	Einheit	Wertebereich	Beschreibung
bOn	1	[TRUE,FALSE]	Schaltet den Regler mit TRUE ein.
bInit	1	[TRUE,FALSE]	Initialisierungsflag, das genau für den ersten Zyklus des Regleraufrufes anstehen muss (TRUE).
bTune	1	[TRUE,FALSE]	Schaltet mit der steigenden Flanke das Selftuning ein. Wird während des Selftuning-Vorgangs auf FALSE geschaltet, so wird das Selftuning abgebrochen und mit den alten Parametern (sofern vorhanden) weitergemacht.
bManual	1	[TRUE,FALSE]	Schaltet mit TRUE auf Handbetrieb um. Wird da Signal wieder FALSE, so wird in den Automatikmode zurückgeschaltet.
bSelSetpoint	1	[TRUE,FALSE]	Wählt einen von zwei möglichen Sollwerten aus. Mit FALSE wird der normale Sollwert gewählt, mit TRUE der Standby-Sollwert.
bSelCtrlParameterSet	1	[TRUE,FALSE]	Wählt einen von zwei Parametersätzen aus. Mit FALSE wird der interne (ermittelte) Parametersatz benutzt, mit TRUE ein von außen aufgeschalteter.
bEnableSoftStart	1	[TRUE,FALSE]	Mit TRUE wird mit der Anfahrerschaltung (Softstart) hochgefahren.
bEnableRamping	1	[TRUE,FALSE]	Mit TRUE wird jeder Sollwertsprung gerampt ausgeführt.
fW1	°C	LREAL	Sollwert.
fW2	°C	LREAL	Standby-Sollwert im Regelfall kleiner als fW1. Mit fSelSetpoint kann zwischen fW1 und fW2 umgeschaltet werden.

Name	Einheit	Wertebereich	Beschreibung
fX	°C	LREAL	Istwert. Dieser Wert muss auf LREAL konvertiert werden.
bOpenThermocouple	1	[TRUE, FALSE]	Mit TRUE ist das Thermoelement offen. Muss von der Hardware gemeldet werden (z.B. KLxxxx).
bReverseThermocouple	1	[TRUE, FALSE]	Mit TRUE wird falsche Polarität des angeschlossenen Thermoelements gemeldet. Muss von der Hardware gemeldet werden.
bBackVoltage	1	[TRUE, FALSE]	Bei TRUE wird eine zu hohe Eingangsspannung am Thermoelement angezeigt. Muss von der Hardware gemeldet werden.
bLeakage	1	[TRUE, FALSE]	Mit TRUE wird ein Leckstrom zu Heizelement detektiert. Muss von der Hardware gemeldet werden.
bShortCircuit	1	[TRUE, FALSE]	Mit TRUE wird ein Kurzschluss im Heizelement detektiert. Muss von der Hardware gemeldet werden.
bOpenCircuit	1	[TRUE, FALSE]	Mit TRUE wird ein offener Stromkreis im Heizelement detektiert. Muss von der Hardware gemeldet werden.
sControllerParameter	keine	Struktur	In dieser Struktur werden generelle Parameter (Abtastzeiten usw.) an den Baustein übergeben.
sParaControllerExternal	keine	Struktur	In dieser Struktur wird ein externer Regler-Parametersatz an den Baustein übergeben.
sLogData	keine	Struktur	In dieser Struktur werden Parameter (Dateinamen usw.) für das Logging an den Baustein übergeben.

Ausgänge

Name	Einheit	Wertebereich	Beschreibung
fYAnalog	keine	LREAL	Analoger Stellwert.
bYPWM	keine	[TRUE, FALSE]	Boolescher Ausgang pulsweitenmoduliert.

Name	Einheit	Wertebereich	Beschreibung
bYDig	keine	[TRUE,FALSE]	Boolescher Ausgang eines Zweipunktreglers (TRUE Stellgröße 100%, FALSE Stellgröße aus)
bYDigPos	keine	[TRUE,FALSE]	Boolescher Ausgang eines Dreipunktreglers (TRUE Stellgröße 100%, FALSE Stellgröße aus)
bYDigNeg	keine	[TRUE,FALSE]	Boolescher Ausgang eines Dreipunktreglers (TRUE Stellgröße -100%, FALSE Stellgröße aus)
dwAlarm	keine	DWORD	Alarmmeldungen (s. ENUM ...)
iState	keine	INT	momentaner Status des Reglers (s. ENUM ...)
sParaControllerInternal	keine	Struktur	In dieser Struktur wird der interne (durch das Tuning ermittelte) Regler-Parametersatz zur Verfügung gestellt.
bError	keine	[TRUE,FALSE]	Liegt ein Fehler vor, so ist bError gleich TRUE.
iErrorId	keine	INT	Ist bError gleich TRUE, so zeigt iErrorId einen Fehlercode an (s. ENUM ...)

11.3.2 Definition der Strukturen

ST_ControllerParameter

```

TYPE ST_ParaControlValue :
STRUCT
(*****
    (* general parameters *)
    iMode : E_ControlMode; (* 1=heating, 2=cooling,
3=heating&cooling *)
    iReactionOnFailure : E_ReactionOnFailure; (* 0=controller off,
1>manual op, 2=yMin, 3=yMax *)
    fYTune : LREAL; (* step change while tuning operation *)
    fYStable : LREAL; (* tuning operation *)
    dwAlarmSupp : DWORD; (* alarm suppression *)
    tCtrlCycleTime : TIME; (* controller cycle time *)
    tTaskCycleTime : TIME; (* plc task cycle time *)

(*****

    (* setpoint parameters *)
    fWMin : LREAL; (* lower limit *)
    fWMax : LREAL; (* upper limit *)
    (* start up *)
    fWStartUp : LREAL; (* soft start plateau setpoint *)
    tStartUp : TIME; (* soft start waiting time*)
    bStartUpRamping : BOOL; (* enable ramping while start up phase
*)
    fWStartUpVeloPos : LREAL; (* max gradient for increasing
setpoint in start up phase*)
    fWStartUpVeloNeg : LREAL; (* max gradient for decreasing
setpoint in start up phase *)
    fWVeloPos : LREAL; (* max gradient for increasing setpoint
*)
    fWVeloNeg : LREAL; (* max gradient for decreasing setpoint

```

```

*)
(*****)

(* actual value parameters *)
bFilter : BOOL;
tFilter : TIME;

(*****)

(* control value parameters *)
fYMin : LREAL; (* lower limit *)
fYMax : LREAL; (* upper limit *)
fYManual : LREAL; (* manual operation*)
fYOnFailure : LREAL; (* control value on failure *)
tPWMCycleTime : TIME; (* PWM *)

(*****)

(* controller settings *)
bEnablePreController : BOOL; (* enable precontroller *)
bEnableZones : BOOL; (* enable zone around setpoint with open
loop control *)
bEnableCVFilter : BOOL; (* enable filter for CV (type see
iFilterType) *)
iFilterType : E_FilterType; (* filtertype of CV filter *)
iControllerType : E_ControllerType; (* used controller normally
PID *)

(*****)

(* min max temperatures *)
TempLow : LREAL;
TempLowLow : LREAL;
TempHigh : LREAL;
TempHighHigh : LREAL;
TempAbsoluteHigh : LREAL;
TempAbsoluteLow : LREAL;

(*****)

(* internal tuning parameters *)
fTuneKp : LREAL := 1.2;
fTuneTn : LREAL := 2.0;
fTuneTv : LREAL := 0.42;
fTuneTd : LREAL := 0.25;
END_STRUCT
END_TYPE
    
```

Beschreibung

Name	Einheit	Wertebereich	Beschreibung
iMode	keine	INT	Reglerbetriebsmode (1 = heating, 2 = cooling, 3 = heating&cooling) (s.u.) [► 40]
iReactionOnFailure	keine	INT	Parametrierbare Reaktion auf Fehler (s.u.) [► 40]
fYTune	keine	LREAL	Stellgröße während der Selbsteinstellung (normalerweise 100%)
fYStable	keine	LREAL	Stellgröße während der Beruhigungsphase (normalerweise 0%)
dwAlarmSupp	keine	DWORD	Maskiert die Alarme aus (s.u.) [► 41]
tCtrlCycleTime	s	TIME	Abtastzeit des Reglers. Der Regler errechnet im Takt dieser Abtastzeit neue Werte für die Stellgröße.
tTaskCycleTime	s	TIME	Zykluszeit der Task. Der FB wird im Takt der Task aufgerufen.
fWMin	K	LREAL	Minimale Sollgröße.
fWMax	K	LREAL	Maximale Sollgröße.
fWVeloPos	K/s	LREAL	Anstiegsgeschwindigkeit (Rampe).
fWVeloNeg	K/s	LREAL	Abfallgeschwindigkeit (Rampe).

Name	Einheit	Wertebereich	Beschreibung
fWStartUp	K	LREAL	Sollgröße beim Anfahren.
tStartUp	s	TIME	Zeit mit Sollwert fWStartUp.
bStartUpRamping	keine	[TRUE,FALSE]	Schaltet Verrampung während der StartUp-Phase ein.
fWStartUpVeloPos	K/s	LREAL	Anstiegsgeschwindigkeit (Rampe) während der StartUp-Phase.
fWStartUpVeloNeg	K/s	LREAL	Abfallgeschwindigkeit (Rampe) während der StartUp-Phase
fYMin	keine	LREAL	Minimalwert der Stellgröße.
fYMax	keine	LREAL	Maximalwert der Stellgröße.
fYManual	keine	LREAL	Stellgröße im Handbetrieb.
fYOnFailure	keine	LREAL	Stellgröße im Fehlerfall (parametrierbar).
tPWMCycleTime	s	TIME	Zykluszeit des PWM Signals.
tFilter	s	TIME	Filterzeitkonstante des Istwertfilters (P-T1 Filter 1. Ordnung)
bFilter	keine	[TRUE,FALSE]	Mit TRUE wird der Istwertfilter aktiviert.
bEnablePreController	keine	[TRUE,FALSE]	Schaltet Vorregler ein.
bEnableZones	keine	[TRUE,FALSE]	Schaltet OpenLoop Verhalten bis nahe Sollwert ein.
bEnableCVFilter	keine	[TRUE,FALSE]	Schaltet Stellgrößenfilter hinter Hauptregler ein.
iFilterType	keine	ENUM	Wahl eines Filtertypes für den Stellgrößenfilter hinter dem Hauptregler (s.u.) [► 41].
iControllerType	keine	ENUM	Wahl eines Regelalgorithmus (s.u.) [► 41]
TempLow	K	LREAL	Relativer unterer Temperaturgrenzwert im ersten Band.
TempLowLow	K	LREAL	Relativer unterer Temperaturgrenzwert im zweiten Band.
TempHigh	K	LREAL	Relativer oberer Temperaturgrenzwert im ersten Band.
TempHighHigh	K	LREAL	Relativer oberer Temperaturgrenzwert im zweiten Band.
TempAbsoluteHigh	K	LREAL	Absoluter oberer Temperaturgrenzwert.
TempAbsoluteLow	K	LREAL	Absoluter unterer Temperaturgrenzwert.
fTuneKp	keine	LREAL	FineTuning Parameter für den PID Regler (nur für Experten)
fTuneTn	keine	LREAL	FineTuning Parameter für den PID Regler (nur für Experten)
fTuneTv	keine	LREAL	FineTuning Parameter für den PID Regler (nur für Experten)
fTuneTd	keine	LREAL	FineTuning Parameter für den PID Regler (nur für Experten)

Beschreibung

ST_ParaController

```

TYPE ST_ParaController :
STRUCT
    (* Main Controller parameter set *)
    KpMain : LREAL;
    
```

```
TnMain : LREAL;
TvMain : LREAL;
TdMain : LREAL;
(* Pre Controller parameter set *)
KpPre : LREAL;
TvPre : LREAL;
TdPre : LREAL;
END_STRUCT
END_TYPE
```

Beschreibung

Name	Einheit	Wertebereich	Beschreibung
KpMain	keine	LREAL	Verstärkungsfaktor für Hauptregler.
TnMain	s	TIME	Nachstellzeit für Hauptregler (I-Anteil).
TvMain	s	TIME	Vorhaltzeit für Hauptregler (D-Anteil).
TdMain	s	TIME	Dämpfungszeit für Hauptregler.
KpPre	keine	LREAL	Verstärkungsfaktor für Vorregler.
TvPre	s	TIME	Vorhaltzeit für Vorregler (D-Anteil).
TdPre	s	TIME	Dämpfungszeit für Vorregler.

ENUM: Errorcodes

Name	Beschreibung
TC_ERR_NOERROR	Kein Fehler.
TC_ERR_INVALIDPARAM	Ungültige Parameter.
TC_ERR_NO_INIT	Baustein Initialisierung fehlt.
TC_ERR_NO_INFLECTION_POINT	Bei der Selbsteinstellung wurde kein Wendepunkt gefunden. Es konnten keine Parameter ermittelt werden.
TC_ERR_INVALID_PARAM	Ungültige Parameter.
TC_ERR_INVALID_CYCLETIME	Ungültige Kombination von Zykluszeiten (Abtastzeiten und PWM Zykluszeiten).
TC_ERR_WRONG_TU	Es konnte durch eine fehlerhafte oder abgebrochene Selbsteinstellung kein gültiger Parameter Tu gefunden werden.

ENUM: ReactionOnFailure

Name	Beschreibung
TC_OnFailureNoFailure	Kein Fehler.
TC_OnFailureStopController	Wenn Fehler (Alarm), dann stoppe Regler.
TC_OnFailureSetManMode	Wenn Fehler (Alarm), dann schalte Regler in Handbetrieb.
TC_OnFailureSetYMax	Wenn Fehler (Alarm), dann setze Stellgröße auf maximalen Wert.
TC_OnFailureSetYMin	Wenn Fehler (Alarm), dann setze Stellgröße auf minimalen Wert.

ENUM: ST_ControlMode

Name	Beschreibung
CTRLMODE_HEATING	Nur Heizen.
CTRLMODE_COOLING	Nur Kühlen.
CTRLMODE_HEATING_COOLING	Heizen und Kühlen.

ENUM: states

Name	Beschreibung
TC_STATE_IDLE	Regler abgeschaltet.
TC_STATE_INIT	Regler wird initialisiert.
TC_STATE_OFF	Regler ausgeschaltet, war vorher eingeschaltet.
TC_STATE_TUNE	Regler im Tuning - Selbsteinstellungszustand.
TC_STATE_MANUAL_OPERATION	Regler im Handbetrieb.
TC_STATE_CLOSED_LOOP	Regler im Automatikbetrieb.
TC_STATE_TUNE_IDLE	Tuning gestartet, aber noch nicht angelaufen. Abwarten von Ruhezustand.
TC_STATE_TUNE_PULSE	Puls zum Ermitteln der Totzeit.
TC_STATE_TUNE_STEP	Sprung zum Ermitteln von Totzeit und max. Geschwindigkeit.
TC_STATE_TUNE_READY	Selbsteinstellung abgeschlossen.
TC_STATE_ERROR	Fehler (logischer Fehler).

ENUM: E_FilterType

Name	Beschreibung
E_FilterType_FIRSTORDER	Filter erster Ordnung.
E_FilterType_AVERAGE	Mittelwertfilter.

ENUM: E_ControllerType

Name	Beschreibung
E_ControllerType_PID	Standard-PID Regelalgorithmus.
E_ControllerType_PID2	geplanter serieller PID Regelalgorithmus.

Bit-Masken für Alarmer

Name	Maske	Beschreibung
nAlarmOpenThermocouple	2#0000_0000_0000_0000_0000_000_000_0000_0001	Hardware: offener Temperaturfühler
nAlarmReverseThermocouple	2#0000_0000_0000_0000_0000_000_000_0000_0010	Hardware: verpolter Temperaturfühler
nAlarmBackVoltage	2#0000_0000_0000_0000_0000_000_000_0000_0100	Hardware: zu hohe Spannung am Temperaturfühler
nAlarmLeakageCurrent	2#0000_0000_0000_0000_0000_000_000_0000_1000	Hardware: Leckstrom gemessen
nAlarmShortCircuit	2#0000_0000_0000_0000_0000_000_000_0001_0000	Hardware: Kurzschluss
nAlarmOpenCircuit	2#0000_0000_0000_0000_0000_000_000_0010_0000	Hardware: kein Strom
nAlarmLimitLow	2#0000_0000_0000_0000_0000_000_001_0000_0000	Software: untere erste relative Temperatur unterschritten
nAlarmLimitLowLow	2#0000_0000_0000_0000_0000_000_010_0000_0000	Software: untere zweite relative Temperatur unterschritten
nAlarmLimitHigh	2#0000_0000_0000_0000_0000_000_100_0000_0000	Software: obere erste relative Temperatur überschritten
nAlarmLimitHighHigh	2#0000_0000_0000_0000_0000_000_100_0000_0000	Software: obere zweite relative Temperatur überschritten
nAlarmAbsoluteHigh	2#0000_0000_0000_0000_0000_0001_000_0000_0000	Software: obere absolute Temperatur überschritten

Name	Maske	Beschreibung
nAlarmAbsoluteLow	2#0000_0000_0000_0000_0010_000_0000_0000	Software: untere absolute Temperatur unterschritten

Mehr Informationen:
www.beckhoff.de/ts4110

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Deutschland
Telefon: +49 5246 9630
info@beckhoff.de
www.beckhoff.de

