|  |  |  |
| --- | --- | --- |
| 作者： | 史汇源 | Logo_Beckhoff_Red上海市江场三路市北工业园区163号5楼（200436）TEL: 021-66312666FAX: 021-66315696 |
| 日期： |  |
| 版本： | V1.0 |
| E\_mail: | Shihuiyuan054342@hotmail.com |

|  |
| --- |
| **温度控制库使用说明与演示** |

|  |
| --- |
| 概 述 |

|  |
| --- |
| 此文档介绍了Beckhoff温度控制功能块的功能与参数，并以烤箱为被控对象演示功能块的实际使用方法与效果。 |

|  |
| --- |
| 文档中包含的文件 |

|  |  |
| --- | --- |
| 文件名称 | 文件说明 |
| Tempsys.pro | 测试所用的plc程序 |
| tempsys.tsm | 应用小实验组态文件 |

|  |
| --- |
| 备 注 |

|  |
| --- |
| 关键字：温度控制，pid，pwm输出 |

|  |
| --- |
| 免责声明 |

|  |
| --- |
| 我们已对本文档描述的内容做测试。但是差错在所难免，无法保证绝对正确并完全满足您的使用需求。本文档的内容可能随时更新，也欢迎您提出改进建议。*文档内容可能随时更新**如有改动，恕不事先通知* |

**温度库使用说明与实例**

一、实验准备

1.实验设备

Beckhoff CX1001-0122 嵌入式控制器

KL2012 数字量输出模块

KL3314热电偶模块

Kl2761 调光模块

KL9260 交流电供电模块

电烤箱

固态继电器

2.硬件连接

（1）将cx控制器与KL模块连接

（2）将烤箱热电偶接入到KL3314模块中，作为温度反馈，以形成闭环控制

（3）根据输出类型不同，有以下两种连接方式

 模拟量输出：

 首先将KL2761模块通过KL9260模块进行供电，然后将烤箱电热棒的电源线接入到KL2761的第一通道中，程序通过模拟量对KL2761的输出电流进行控制，从而影响电热棒的加热功率。

PWM输出：

 将交流电电源线L端与烤箱L端分别连接在固态继电器强电侧的两个接线柱上，然后电源线的N端与烤箱的N端相连。继电器的弱电侧接入到KL2012的第一输出通道中，程序通过Kl2012输出PWM信号，将电热棒电流进行斩波，从而控制电热棒的加热功率。

3.软件配置

（1）将程序编译后添加到PLC-Configuration中

（2）将Main\_Simu.fxValue于Kl3314的第一通道相连接

（3）按照输出方式不同，将Main\_Simu.byPWMPos与Kl2012的第一通道相链接或将Main\_Simu.fyValue与KL2761的第一通道链接

（4）激活配置并将systemmanager切换到run模式下

二、功能块与结构体简介

（1）功能块描述：



输入信号：

 eControlMode：控制模式

0： idle，1：passive，2：active，3：reset，4:manual 5:tune 6:selftest 7:synchronize

 bSelSetpoint：选择两个设定值中的一个。FALSE 为主设定值（加热值），true为辅助设定值(保温值)

 fW1：主设定值；

 fW2：辅助设定值，通常小于fW1，

 fSelSetpoint :用于选择切换fW1 和 fW2。；

 fX： 实际值，必须转化为LREAL格式；

 fYManual：手动控制；

 sParaControllerExternal：外部控制器参数输入；

 sControllerParameter：参数设置。

bOpenThermocouple：如果为 TRUE，表示热电偶开路。必须由硬件指示 (例如：KLxxxx)；

 bReverseThermocouple：TRUE 表示热电偶的连接极性错误，必须由硬件指示；

 bBackVoltage：TRUE 表示热电偶的输入电压太高，必须由硬件指示；

 bLeakage：TRUE 表示加热元件检测到峰值电流，必须由硬件指示；

 bShortCircuit：TRUE 表示加热元件检测到短路信号，必须由硬件指示；

 bOpenCircuit：TRUE 表示加热元件检测到开路信号，必须由硬件指示；

输出信号：

 fYAnalog：模拟量控制；

 bYPWMPos：PWM波输出，加热，Bool量；

 bYPWMNeg：PWM波输出，冷却，Bool量；

 bYDigPos：三点式输出，bool量，值为1时，控制值100%；

 bYDigNeg：三点式输出，bool量，值为1时，控制值-100%；

 dwAlarm：报警值；

 fMaxOverShoot：超调值；

 tStartUpTime：上升时间；

 eCtrlState：控制器实际的状态；

sParaControllerInternal：内部自整定参数；

bError：报警输出，有报警时，值置为1；

iErrorId：报警代码

（2） 控制器参数结构体简介

**ST\_ControllerParameter**

TYPE ST\_CTRL\_TempCtrlParameter:
STRUCT
    (\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*)
    (\* general parameters 一般参数\*)
    iMode : E\_CTRL\_ControlMode; (\* 1=加热, 2=冷却, 3=加热及冷却 \*)
    iReactionOnFailure : E\_CTRL\_ReactionOnFailure; (\* 0=控制器关闭, 1=手动操作, 2=yMin, 3=yMax \*)
    bSelCtrlParameterSet : BOOL; (\* FALSE = 内部pid参数设定, TRUE = 外部pid参数设定\*)

dwAlarmSupp : DWORD; (\* 报警限制值\*)
tCtrlCycleTime : TIME; (\*控制器周期时间\*)
tTaskCycleTime : TIME; (\* PLC任务周期时间\*)

(\* tuning parameter 整定参数\*)
    iTuningMode : E\_CTRL\_TuneMode; (\* 只加热, 只冷却, 先加热后冷却，先冷却后加热\*)
    tTuneStabilisation : TIME := t#20s; (\* 等待系统稳定时间\*)
    fEndTunePercentHeating : FLOAT := 80.0; (\* 大于80%时，切换成闭环控制\*)
    fYTuneHeating : FLOAT; (\* step change while tuning operation \*)

fYStableHeating : FLOAT; (\* 加热调整操作\*)
    fEndTunePercentCooling : FLOAT := 20.0; (\*小于20%时，切换成闭环控制\*)
    fYTuneCooling : FLOAT; (\* step change while tuning operation \*)
    fYStableCooling : FLOAT; (\* 冷却调整操作\*)
    fScalingFactor : FLOAT := 1.0; (\*加热和冷却KP比例系数 \*)

(\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*)
 (\* setpoint parameters  设定参数\*)
    fWMin : FLOAT; (\* 下限值\*)
    fWMax : FLOAT; (\* 上限值\*)

(\* start up 启动\*)
bEnableSoftStart : BOOL; (\*FALSE = 无软启动, TRUE = 软启动\*)
bEnableRamping : BOOL; (\*FALSE = 无斜波, TRUE = 有斜波\*)
fWStartUp : LREAL; (\* 软启动预设定值 \*)

tStartUp : TIME; (\* 软启动等待时间\*)

bStartUpRamping : BOOL; (\* 启动过程时允许斜波功能 \*)

fWStartUpVeloPos : LREAL; (\* 启动过程时增加给定的最大梯度值 \*)

fWStartUpVeloNeg : LREAL; (\* 启动过程时减小给定的最大梯度值 \*)

fWVeloPos : LREAL; (\* 增加给定的最大梯度值 \*)

fWVeloNeg : LREAL; (\* 减小给定的最大梯度值 \*)



(\* actual value parameters 实际值参数\*)

bFilter : BOOL;

tFilter : TIME;

(\* deadband parameters 死区参数\*)
    bDeadband : BOOL;
    fEDeadband : FLOAT; (\* 死区 \*)

(\* control value parameters    控制值参数 \*)
    fYMin : FLOAT; (\* 下限值\*)
    fYMax : FLOAT; (\* 上限值\*)
    fYManual : FLOAT;  (\* 手动操作值\*)
    fYOnFailure : FLOAT; (\* 故障时的控制值 \*)
    tPWMCycleTime : TIME; (\* PWM: 周期 \*)
    tPWMMinOffTime : TIME; (\* PWM: min off time \*)
    tPWMMinOnTime : TIME; (\* PWM: min on time \*)
    tPWMWaitingTime : TIME; (\* PWM: min waiting time \*) (\* not yet implemented !!!!\*)
    fYThresholdOff : FLOAT; (\* 3-Point: 关断极限\*)
    fYThresholdOn : FLOAT; (\* 3-Point: 开始极限\*)
    nCyclesForSwitchOver : INT := 100;

(\* controller settings 控制器设置 \*)

bEnablePreController : BOOL; (\* 允许预控制器 \*)

bEnableZones : BOOL; (\* 允许开环调节时的设定调节带 \*)

bEnableCVFilter : BOOL; (\*允许 CV 滤波 (类型参见FilterType) \*)

iFilterType : E\_FilterType; (\* CV 滤波器类型 \*)

iControllerType : E\_ControllerType; (\* 使用的控制器 一般为PID 类型 \*)

(\* min max temperatures 最小最大 温度 \*)

TempLow : LREAL;第一报警带下限

TempLowLow : LREAL;第二报警带下限

TempHigh : LREAL;第一报警带上限

TempHighHigh : LREAL; 第一报警带上限

TempAbsoluteHigh : LREAL;报警上限

TempAbsoluteLow : LREAL; 报警下限



(\* internal tuning parameters 内部整定参数\*)
    fTuneKp : FLOAT := 1.2; 比例
    fTuneTn : FLOAT := 2.0; 积分时间常数
    fTuneTv : FLOAT := 0.42; 微分时间常数
    fTuneTd : FLOAT := 0.25; 阻尼时间常数

END\_STRUCT
END\_TYPE

三、使用方法说明

首先要讲软件库TcTempCtrl.lib通过librarymanager添加到项目中。然后创 建 一 个 FB\_TempController 控 制 器 功 能 块 的 实 例 。同时也要创建一个ST\_ControllerParameter 控制器参数结构体实例。并根据需要配置参数。

1. 升温方式：

在本例子中，需要进行两段式加温，即先加热到一个温度并保温一段时间，然后再加热到第二个温度。此功能可通过将ST\_ControllerParameter结构体中的bEnableSoftStart置为true实现，恒温时间通过tStartUp进行设置。如需控制温度上升速度，可将bEnableRamping与bStartUpRamping也置为true，并通过fWStartUpVeloPos，fWStartUpVeloNeg ，fWVeloPos，fWVeloNeg设置斜率。需要注意的是，无论上升还是下降 的斜率参数都为正数，如将下降斜率设为负数，设定值发生器会报错导致无法进行正常运作。

1. 输出模式：

温度控制功能块提供了三种形式的输出，模拟量，PWM以及三点式输出。根据需要将变量填入到功能块相应的输出端即可。需要注意的是在设置输出范围时需考虑情况。如对Kl2761模块而言，其接受的控制值范围为0-32767。而对PWM输出而言其有效范围为0-100，100表示一直为高电平。因此在设置输出值范围，自整定输出fYTuneHeating等等参数时需要根据实际情况进行设置。

3.PID参数整定：

在第一次使用时，由于pid控制器参数为空，此时即使将功能块切换到eCTRL\_MODE\_ACTIVE模式下，也无法进行正常工作。此时需要对控制器的pid参数进行整定。整定方式有两种：功能块自整定和手动整定。

自整定方式比较方便，只需将控制器模式切换为eCTRL\_MODE\_Tune，并等待20秒，在此期间需维持传感器温度波动不超过正负一度，然后程序将以ST\_ControllerParameter中的fYTuneHeating值进行加热，直到达到设定温度的80%后加热停止。程序将根据温升曲线自动计算出PID控制器的参数。为了保证参数计算的准确性，必须保证升温在40度以上。此参数保存在ST\_CTRL\_ParaController结构体。为了避免重复整定参数，可将此结构体申明为PERSISTENT，并设置定时器或按扭来自动或手动的保存数据。下图为根据自动整定参数进行控制的烤箱温度曲线。首先烤箱到达了第一加热温度50度后，恒温了一定时间，然后达到最终温度100度。



自动整定的参数会产生大约10%的超调，如希望超调较小，可修改ST\_CTRL\_ParaController下列参数后再进行自整定：

手动整定：在手动整定时先要了解pid各个参数的作用。比例调节：加快系统响应速度，改善动态性能。但容易产生超调。积分调节：消除稳态误差，但Tn过大可能产生积分饱和现象。微分环节：根据变化率进行控制，预测变化趋势，可消除超调。但会使系统响应抗扰动性能降低，响应时间变慢。手动整定的过程为：

(1)确定比例系数Kp确定比例系数Kp时，首先去掉PID的积分项和微分项，可以令Ti=0、Td=0，使之成为纯比例调节。输入设定为系统允许输出最大值的60％～70％，比例系数Kp由0开始逐渐增大，直至系统出现振荡；再反过来，从此时的比例系数Kp逐渐减小，直至系统振荡消失。记录此时的比例系数Kp，设定PID的比例系数Kp为当前值的60％～70％。

(2)确定积分时间常数Ti比例系数Kp确定之后，设定一个较大的积分时间常数Ti，然后逐渐减小Ti，直至系统出现振荡，然后再反过来，逐渐增大Ti，直至系统振荡消失。记录此时的Ti，设定PID的积分时间常数Ti为当前值的150％～180％。

(3)确定微分时间常数Td微分时间常数Td一般不用设定，为0即可，此时PID调节转换为PI调节。如果需要设定，则与确定Kp的方法相同，取不振荡时其值的30％

下图为在自整定的基础上，为了实现较快的响应速度手动增加了比例因数Kp后的温度曲线：



4．在参数整定后将功能块切换到eCTRL\_MODE\_ACTIVE模式下，即可进行正常的闭环控制。如果遇到报错，可用eCTRL\_MODE\_RESET进行复位。此模式亦会将启动后修改的参数复位为原始状态（不包括pid控制器参数）。在需要停止加热时，可将功能块切换至eCTRL\_MODE\_IDLE模式下。