SIEMENS西门子 中国葫芦岛市智能化工控设备代理商

产品名称	SIEMENS西门子 中国葫芦岛市智能化工控设备代理商
公司名称	浔之漫智控技术(上海)有限公司
价格	.00/件
规格参数	西门子:代理经销商 模块:全新原装 假一罚十 德国:正品现货 实体经营
公司地址	上海市松江区石湖荡镇塔汇路755弄29号1幢一层A区213室
联系电话	15801815554 15801815554

产品详情

使用 PID_Temp 的多区域控制简介在多区域控制系统中，可同时控制工厂的多个部分（即所谓的多个区域），使其达到不同的温度。多区域控制系统的特点为各个温度区域会由于热耦合而相互影响，例如，某个区域的过程值会因热耦合而影响其它区域的过程值。这种影响的作用强度取决于工厂的结构和这些区域所选的工作点。示例：例如，塑料加工行业的挤压厂。必须对通过挤压机的混合物进行控制，使其达到不同的温度，以实现最优处理。例如，可能要求挤压机填料口的温度不同于排料口。各个温度区域会由于热耦合而相互影响。在多区域控制系统中使用 PID_Temp 时，每个温度区域都由单独的 PID_Temp 实例进行控制。在多区域控制系统中使用 PID_Temp 时，请遵照下列说明。分别进行加热和制冷预调节通常，对工厂进行初始调试时首先会执行预调节，以便对 PID 参数进行初始设置并对工作点进行控制。对多区域控制系统进行预调节时通常可对所有区域同时执行预调节。对于已激活制冷过程且将 PID 参数切换作为加热/制冷方法（Config.ActivateCooling = TRUE，Config.AdvancedCooling = TRUE）的控制器，PID_Temp 可在一个步骤中实现加热和制冷的预调节（Mode = 1，Heat.EnableTuning = TRUE，Cool.EnableTuning = TRUE）。但是，建议不要使用这种调节对多区域控制系统中的多个 PID_Temp 实例同时进行预调节。而应首先分别执行加热预调节（Mode = 1，Heat.EnableTuning = TRUE，Cool.EnableTuning =FALSE）和制冷预调节（Mode = 1，Heat.EnableTuning = FALSE，Cool.EnableTuning =TRUE）。只有当所有区域都完成加热预调节且达到工作点时，才能启动制冷预调节。这会降低调节过程中各区域间由于热耦合而产生的相互影响。157使用 PID_Temp7.5 使用 PID_Temp 的多区域控制PID 控制功能手册, 11/2022, A5E35300232-AF调整延迟时间如果应用 PID_Temp 的多区域控制系统的各区域间存在较强的热耦合，应quebaotongguoPIDSelfTune.SUT.AdaptDelayTime = 0 禁止调整预调节延迟时间。否则，如果在调整延迟时间期间（此阶段加热被禁用），某个区域的制冷因其它区域的热效应而无法进行，则确定延迟时间时可能出错。暂时禁用制冷对于已激活制冷 (Config.ActivateCooling = TRUE) 的控制器，通过设置 DisableCooling =TRUE，PID_Temp 可在自动模式下暂时禁用制冷。这可以确保当其它区域的控制器尚未完成加热调节时，此控制器在调试过程中不会以自动模式制冷。否则，调节可能会因各区域间的热耦合而受到负面影响。步骤对存在热耦合的多区域控制

系统进行调试时，可按以下步骤进行操作：1. 对于所有已激活制冷的控制器，设置 DisableCooling = TRUE。2. 对于所有控制器，设置 PIDSelfTune.SUT.AdaptDelayTime = 0。3. 指定所需设定值（Setpoint 参数）并对所有控制器同时启动加热预调节（Mode = 1，Heat.EnableTuning = TRUE，Cool.EnableTuning = FALSE）。4. 耐心等待，直到所有控制器均完成加热预调节。5. 对于所有已激活制冷的控制器，设置 DisableCooling = FALSE。6. 耐心等待，直到所有区域的过程值均达到稳定状态，且接近相应的设定值。如果对于某个区域，经过很长时间都无法达到设定值，则说明加热或制冷执行器的作用太弱。7. 对于所有已激活制冷的控制器，启动制冷预调节（Mode = 1，Heat.EnableTuning =FALSE，Cool.EnableTuning = TRUE）。说明过程值超出限值如果在自动模式下通过 DisableCooling = TRUE 禁用了制冷，则可能导致当 DisableCooling =TRUE 时，过程值超出设定值或过程值限值。使用 DisableCooling 时请注意观察过程值，在适用的情况下可以进行干预。说明多区域控制系统对于多区域控制系统，各区域间的热耦合在调试或运行期间可能导致过调次数增加、暂时或长时间超出限值或出现暂时或长时间的控制偏差。请注意观察过程值并准备好进行干预。根据系统不同，操作步骤可能会与上述步骤有所不同。替代设定值为了指定设定值，除 Setpoint 参数外，PID_Temp 会通过 SubstituteSetpoint 变量提供替代设定值。此替代设定值可通过设置 SubstituteSetpointOn = TRUE 或在调试编辑器中选中相应的复选框来激活。通过替代设定值，可在调试或调节等过程中直接在从控制器暂时指定设定值。这种情况下，不必在程序中对主控制器输出值与从控制器设定值的互连（级联控制系统正常运行所必需的）进行更改。为使主控制器对该过程产生影响或执行调节，必须禁用所有下游从控制器的替代设定值。可以对当前有效的设定值进行监视，因为该设定值以 CurrentSetpoint 变量的形式被 PID 算法使用参与计算。工作模式和故障响应PID_Temp 实例的主控制器或从控制器不会更改此 PID_Temp 实例的工作模式。如果其中一个从控制器发生故障，主控制器仍然保持当前工作模式。如果主控制器发生故障，从控制器仍然保持当前工作模式。 但是，由于将主控制器的输出值用作从控制器的设定值，之后从控制器的进一步操作将取决于主控制器的故障和组态的故障响应： 如果对主控制器组态了 ActivateRecoverMode = TRUE，且故障不会阻止 OutputHeat 的计算过程，则故障不会对从控制器产生任何影响。 如果对主控制器组态了 ActivateRecoverMode = TRUE，且故障会阻止 OutputHeat 的计算过程，则主控制器会输出上一次的输出值或已组态的替代输出值 SubstituteOutput，具体取决于 SetSubstituteOutput。 然后，从控制器会将其用作设定值。由于已对 PID_Temp 进行预组态，在此情况下会输出替代输出值0.0（ActivateRecoverMode = TRUE、SetSubstituteOutput = TRUE、SubstituteOutput =0.0）。 为应用组态合适的替代输出值，或启用上一个有效 PID 输出值(SetSubstituteOutput = FALSE)。 如果对主控制器组态了 ActivateRecoverMode = FALSE，则当发生故障或输出 OutputHeat= 0.0 时，主控制器会切换到“ 未激活 ”模式。 然后，从控制器会使用 0.0 作为设定值。故障响应位于组态编辑器的输出设置中。同步多个jingque调节过程如果在自动模式下启动jingque调节且 PIDSelfTune.TIR.RunIn = FALSE，则 PID_Temp 会尝试通过PID 控制和当前 PID 参数达到设定值。达到设定值后，才会启动实际调节过程。对于多区域控制系统，各个区域达到设定值所需的时间可能各不相同。如果要对多个区域同时执行jingque调节，PID_Temp 可以在达到设定值后，等待进一步的调节步骤，从而同步这些过程。步骤这可以确保当实际调节步骤启动时，所有控制器都已达到设定值。这会降低调节过程中各区域间由于热耦合而产生的相互影响。对于相应区域要同时执行jingque调节的各控制器，请执行以下步骤：1. 对于所有控制器，设置 PIDSelfTune.TIR.WaitForControlIn = TRUE。这些控制器必须处于自动模式，且 PIDSelfTune.TIR.RunIn = FALSE。2. 指定所需设定值（Setpoint 参数）并对所有控制器启动jingque调节。3. 耐心等待，直到所有控制器的 PIDSelfTune.TIR.ControlInReady = TRUE。4. 对于所有控制器，设置 PIDSelfTune.TIR.FinishControlIn = TRUE。然后，所有控制器会同时启动实际调节过程。

使用 PID_Temp 进行超驰控制超驰控制超驰控制时，两个或多个控制器共享一个执行器。只有一个控制器可以随时访问执行器并影响过程。由逻辑运算决定可以访问执行器的控制器。通常根据所有控制器的输出值比较结果做出此决定（例如，进行最大选择时），具有最大输出值的控制器将获得对执行器的访问权限。基于输出值的选择要求所有控制器均在自动模式下工作。对不影响执行器的控制器进行更新。为防止饱和效应及其对控制响应和控制器之间的切换产生负面影响，这很有必要。自版本 V1.1 起，PID_Temp 通过提供一个用于更新未激活控制器的简单过程，支持超驰控制：通过使用变量 OverwriteInitialOutputValue 和 PIDCtrl.PIDInit，可以预分配自动模式下控制器的积分作用，好像在上一周期中 PID 算法已为 PID

输出值计算 PidOutputSum =OverwriteInititalOutputValue。为此，OverwriteInitialOutputValue 与当前可以访问执行器的控制器的输出值互连。通过设置位 PIDCtrl.PIDInit，触发积分作用的预分配以及控制器循环和PWM 周期的重启。根据预分配的（并针对所有控制器同步的）积分作用，以及当前控制偏差的比例作用与积分作用，在当前循环中进行输出值的后续计算。通过 PIDCtrl.PIDInit = TRUE 调用期间，微分作用未激活，因此对输出值不起作用。此过程可以确保仅根据当前的过程状态和 PI 参数对当前输出值进行计算，并从而决定可以访问执行器的控制器。可防止未激活控制器的饱和效应，并因此防止切换逻辑的错误决定。159使用 PID_Temp7.6 使用 PID_Temp 进行超驰控制PID 控制功能手册, 11/2022, A5E35300232-AF要求 只有在激活了积分作用时（变量 Retain.CtrlParams.Heat.Ti 和 Retain.CtrlParams.Cool.Ti >0.0），PIDCtrl.PIDInit 才有效。 您必须在用户程序中自行分配 PIDCtrl.PIDInit 和 OverwriteInitialOutputValue（请参见下面的示例）。PID_Temp 不会自动更改这些变量。 仅当 PID_Temp 处于自动模式（参数 State = 3）时，PIDCtrl.PIDInit 才有效。 如果可能，请选择 PID 算法的采样时间（ Retain.CtrlParams.Heat.Cycle 和Retain.CtrlParams.Cool.Cycle 变量）以使所有控制器的采样时间均相同，并在同一个循环中断 OB 中调用所有控制器。这样，可以确保在一个控制器循环或 PWM 周期内不发生切换。说明不断调整输出值限制也可以通过在其它控制器系统中不断调整输出值限制实现这一操作，而不是如此处所述对没有执行器访问权的控制器进行主动更新。无法使用 PID_Temp 实现这一操作，因为在自动模式下不支持更改输出值限制。示例：大型锅炉的控制PID_Temp 用于控制大型锅炉。主要目标是控制温度 Input1。为此使用控制器 PID_Temp_1。此外，通过限制控制器PID_Temp_2 使温度 Input2 保持在附加测量点的上限值以下。这两个温度仅受一个加热器的影响。控制器的输出值对应于加热功率。通过编写程序变量 ActuatorInput 并借助 PID_Temp 的脉宽调制输出值（参数OutputHeat_PWM）对该加热器进行控制。在参数 PID_Temp_1.Setpoint 处指定温度 Input1的设定值。在参数 PID_Temp_2.Setpoint 处将附加测量点的温度上限值指定为设定值。两个控制器必须共享一个加热器作为共享的执行器。在这种情况下，通过 PID 输出值（采用实数格式，参数 PidOutputSum）的最小选择实现逻辑，该逻辑决定哪个控制器获得执行器的访问权。由于 PID 输出值对应于加热功率，因此需要较低加热功率的控制器将获得控制权。设备正常运行时，主受控变量的过程值对应于设定值。主控制器 PID_Temp_1 已稳定在固定的PID 输出值 PID_Temp_1.PidOutputSum。正常操作过程中，限制控制器 Input2 的过程值显著低于指定为 PID_Temp_2 设定值的上限。因此，限制控制器要增大加热功率以增大其过程值，即，它将计算一个大于主控制器 PID_Temp_1.PidOutputSum 输出值的 PID 输出值PID_Temp_2.PidOutputSum。切换逻辑的最小选择从而使得主控制器 PID_Temp_1 可以继续访问执行器。此外，quebaotongguo赋值 。