



温度控制系统的实现

V2.0

恒温控制，越来越多地出现在工业应用中。类似的还有压力控制等等。成熟的温度控制系统不在少数，采用的都是 PID 算法。但是用户如果没有相当的经验和精力，是无法完成的。本文试图给大家提供一个简单易行的温度控制系统方案。

本文将从硬件和软件 2 个方面进行阐述。

一：系统硬件构成：

1：温度模块 CF-6T-2DA 一个。

CF-6T-2DA 可以测量 6 路温度输入，并且可以输出 2 路电压。

温度测量使用 K 型热电偶，测量范围 0-600℃。测量到的温度值通过串口线自动传送到 PLC CF2n-20MT 的寄存器 D0-D5 里面。由于 CF-6T-2DA 本身有 2 个通信口，支持穿透功能，所以，并不占用 PLC 的通信口。

2：PLC CF2n-20MT 一个。CF2n20MT 有 12 点输入，8 点晶体管输出。PLC 的输出端 Y0-Y5



（注意，日本三菱的只可以在 Y0、Y1 可以输出 PWM）可以使用 PWM 指令控制固态继电器或者继电器，然后驱动加热棒。

3：简易文本 CF-160，CF-160 可以显示和修改 PLC 里面的寄存器和继电器。可以显示负数及小数点等。在这里用来设定目标温度，显示当前温度，以及设定加热的一些参数如加热速度和稳定



误差调整值。按键也可以控制系统的启停等。

整个系统的连接如上图。

二：准备知识

接下来的是一些准备知识：

1：加热棒。

加热棒用来给系统加热，加热棒的功率的选择要注意。加热棒的功率要大于系统的散热功率（速度）。否则，无论你怎么加热，都达不到目标温度。这个相信大家都懂。

但是，系统的散热功率（速度）不是恒定的。例如，进料的速度，初始温度，以及设备的环境温度，都会改变散热速度。千万不能做一台设备，夏天能够正常工作，冬天就不行。或者车间打开电风扇就不行了。所以，加热棒的加热功率要有足够的余量。

2：加热棒的驱动。

加热棒的驱动需要用继电器或者固态继电器。由于在接下来的算法中，继电器大约每 5 秒钟就动作一次，所以，这个继电器属于消耗品，需要使用座子，方便客户自行更换。

不建议属于固态继电器，主要是成本和更换的难易程度考虑。

也不建议使用 PLC 内部的继电器，主要是更换的难易程度考虑。

温度采集过来后，通过运算，得到一个控制值，用这个控制值去控制 PWM 的输出, PWM 输出口通过继电器接加热棒。

3：稳定误差。在有些应用上，可能会存在一个稳定误差。即温度控制很稳定，但始终比目标温度低。例如，设定的目标温度是 120℃，但是显示的当前温度始终是 115 摄氏度，而且很稳定。这个 5 摄氏度的误差，就称之为温度误差。这个是由于加热系统产生的热量，刚好和系统的散热达到平衡所导致的。所以要通过一个调整值来消除稳定误差。类似 PID 算法里面的 D。调整后，系统就没有稳定误差了。

但是稳定误差会随着工作环境和进料的速度，初始温度等变化，这个就比较麻烦。还好变化不大。而且，一般来说，稳定误差的变化，客户会通过调整目标温度的方法来应对，所以，大多数的应用都没有什么问题。这也是我们接下来要给大家介绍的这个算法的缺点。如果系统的控制要求很高，就需要用 PID 算法了。

为了调试的方便，也为了减少你以后下现场的机会，需要把稳定误差的值设置在文本或者触摸屏里面。最好设置在单独的页面里面，然后再加密码。防止被人修改。



调试的时候，先把稳定误差设为零，等你把系统运行起来了，自然就知道温度误差了，然后再设置。

5: 比例值。

比例值越大，加热功率就越大，但是就容易造成温度过冲。太小则达到目标温度的时间太长。为了调试的方便，加热速度也设置在文本或者触摸屏里面。

加热棒的功率更换后（电热棒损坏，买不到合适的），也需要调整加热速度。比如加热棒功率改小了，则需要相应地把加热速度调大。反之亦然。

6: 当前温度的显示

一般地，系统有触摸屏或者文本显示器来显示当前温度值，设置目标温度。


由于系统噪声的影响，采集到的当前温度值可能会不停地来回跳变，即使在取平均值后依然改善不大。在某些工作环境或者电网下，还比较剧烈。如果把采集到的当前温度，直接显示给客户，会影响客户对产品的信心。

所以，强烈建议不要用文本或者触摸屏直接显示当前温度值。

如下程序，当前温度值在 D191 里面，但是文本显示

的却是 D181 里面的值。每秒钟把当前温度从 D191 到 D181 刷新一次。这样的话，在一秒以内，虽然 D191 里面的值在不停地跳

变，D181 的值，也就是文本显示



的值最多只变一次。这样一来，显示的“稳定性”就好多了。

实际上，所有的数据采集系统，不论工业，还是民用，底层的处理都是如此。

三：温度控制系统的算法

大家最关心的莫过于温度控制系统的算法，即通过测量当前温度，和控制加热棒的功率，如何能把系统的温度稳定在目标值附近。这一点绝对是个技术活。如果你想搞定，下面的算法必须完全看懂。否则，你可能要天天下现场，去处理客户的投诉了。

温度控制系统的算法，最典型的莫过于 PID。不过，PID 算法过于高深，参数众多，又互相制约。没有一定的理解和经验，是无法达到好的应用效果的。在这里，我们不做讨论。

在此，提出一个不使用 PID 指令，而控制温度的算法。简单易行，调试方便。

算法的思路是，当前值距目标值越远，PWM 的占空比就越大，加热棒的功率也就越大。从而是当前值无限接近目标值。

考虑到稳定误差，需要把目标值抬高一点，那么就把目标值加个稳定误差就可以了。



调整后的目标值减去当前值，得到温度差，温度差*加热速度得到控制值。然后用控制值去控制 PWM。

加热速度越大，控制值就会变大，从而 PWM 的占空比变大。

温度差越大，控制值就会变大，从而 PWM 的占空比变大。

以下红色的部分是符合 C 语言格式的伪代码。

调整后的目标值=目标值+稳定误差； //注释：把目标值抬高一点，就可以抵消稳定误差

If (调整后的目标值>当前值) {

 温度差=调整后的目标值-当前值；

}

Else 温度差=0； //注释：当前温度已经大于目标温度了，当然不能加热了。

控制值=温度差*加热速度；

If (控制值>5000) { //注释： PWM 指令要求 S2>=S1，否则就会报错，停止运行的。

 控制值=5000；

}

然后用控制值去控制 PWM 即可。

程序是如此的简单，为什么我们还要花这么大的功夫去找一篇文章，主要就是很多用户编程时，没有对目标值和当前值进行大小判断，或者没有对控制值和 5000 进行大小判断，导致程序偶尔报错，导致下现场。

如果你觉得以上的东西太难懂了，没有关系，我们现在的产品已经把以上算法集成到底层程序里面了，你只要做简单的设置即可。如图，最多可以实现 8 个通道的恒温系统的控制。

通道	使能继电器	输出口	当前值	目标值	比例值	稳定误差	控制周期 (mS)
通道 0	M8080	Y0	D190	D200	D210	D220	D230
通道 1	M8081	Y1	D191	D201	D211	D221	D231
通道 2	M8082	Y2	D192	D202	D212	D222	D232
通道 3	M8083	Y3	D193	D203	D213	D223	D233
通道 4	M8084	Y4	D194	D204	D214	D224	D234
通道 5	M8085	Y5	D195	D205	D215	D225	D235
通道 6	M8086	Y6	D196	D206	D216	D226	D236
通道 7	M8087	Y7	D197	D207	D217	D227	D237



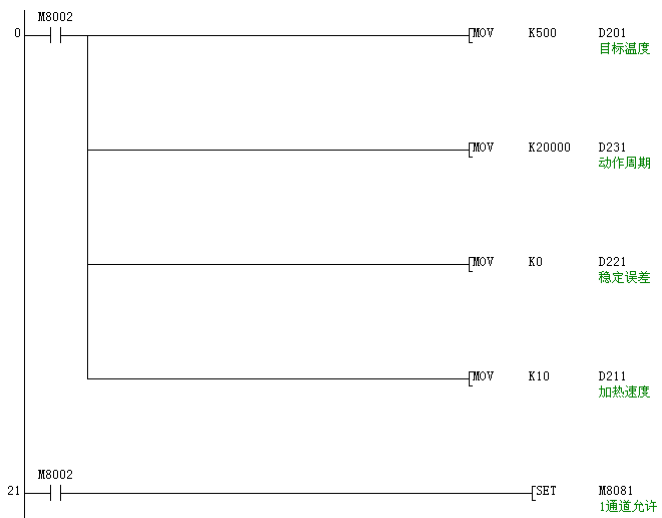
以下是三菱 PLC 的梯形图程序

程序如图。使用通道 1 进行恒温控制。

当前温度值即实际温度，在 D191 里面，不管你是用什么方法得到当前温度值，请把当前温度值送到 D191 里面即可。

D200-D511 是掉电保持寄存器。所以这些设定值得电后是不会丢失的。

在初次调整时，D221（稳定误差）的值设为 0，这样可以方便调整 D200 里面的加热速度。一般选 1-10 即可，在文本或者触摸屏里面要对范围进行限定。加热速度



刚开始的时候要小。以免稳定过冲对设备或者材料造成伤害。

加热速度调整好以后，如果观察不到稳定误差，可以令 D221 里面的值为 0。

由于加热速度和稳定误差调整值都通过文本来调整，不需要反复下载 PLC 程序，所以非常方便客户现场调试。甚至不需要工程师到现场，极大地降低了人工成本。

一旦设备调试稳定，就不可以再调整 D211 和 D221 的值了。所以建议在文本上面对加热速度和稳定误差调整值的修改页面设置密码。