

## 《匠人手记》之三

# 一阶滤波算法之深入研究

作者：程序匠人

出处：《匠人的百宝箱》

### 1. 前言

关于一阶滤波的软件算法，匠人原来已经发表过一次，文件名称叫“《匠人手记》之三《一阶滤波方法》”。但当时限于时间仓促，只是简单地给出了两个流程图，没有做深入描述。

这次，匠人将根据历年来对该算法应用的亲身体会，重新进行整理，并改正原文档中的错误。修改后的手记更名为“《匠人手记》之三《一阶滤波算法之深入研究》”。

另外：匠人将对以往的手记进行整理和完善，不断添加新的心得体会。逐渐将《匠人手记》系列文章打造成网上的精品手记。

### 2. 一阶滤波算法的原理

一阶滤波，又叫一阶惯性滤波，或一阶低通滤波。是使用软件编程实现普通硬件 RC 低通滤波器的功能。

一阶低通滤波的算法公式为： $Y(n) = \alpha X(n) + (1 - \alpha)Y(n-1)$

式中： $\alpha$  = 滤波系数； $X(n)$  = 本次采样值； $Y(n-1)$  = 上次滤波输出值； $Y(n)$  = 本次滤波输出值。

一阶低通滤波法采用本次采样值与上次滤波输出值进行加权，得到有效滤波值，使得输出对输入有反馈作用。

### 3. 一阶滤波算法的公式

#### ● 公式原型：

本次滤波结果 = 新采样值 × 滤波系数 ÷ 256 + 上次滤波结果 × (256 - 滤波系数) ÷ 256

#### ● 公式优化：

在上面的公式中，一共需要进行 4 次乘/除法运算。

对于一些没有乘/除法指令的单片机来说，需要用循环加/减法来实现乘/除法运算。过多次乘/除法运算会降低系统的效率。因此，为了提高单片机的运算速度我们需要将公式进行运算优化。

经过下面的方法后，我们只需要进行 2 次乘/除法运算即可完成，效率提高了一倍。

#### ■ 当新采样值 < 上次滤波结果时：

滤波结果 = 上次滤波结果 - (上次滤波结果 - 新采样值) × 一阶滤波系数 ÷ 256

#### ■ 当新采样值 > 上次滤波结果时：

滤波结果 = 上次滤波结果 + (新采样值 - 上次滤波结果) × 一阶滤波系数 ÷ 256

#### ● 说明：

滤波系数 = 0 ~ 255；该系数决定新采样值在本次滤波结果中所占的权重。

一阶滤波系数可以是固定的，也可以按一定算法在程序中自动计算。

#### 4. 一阶滤波算法的程序

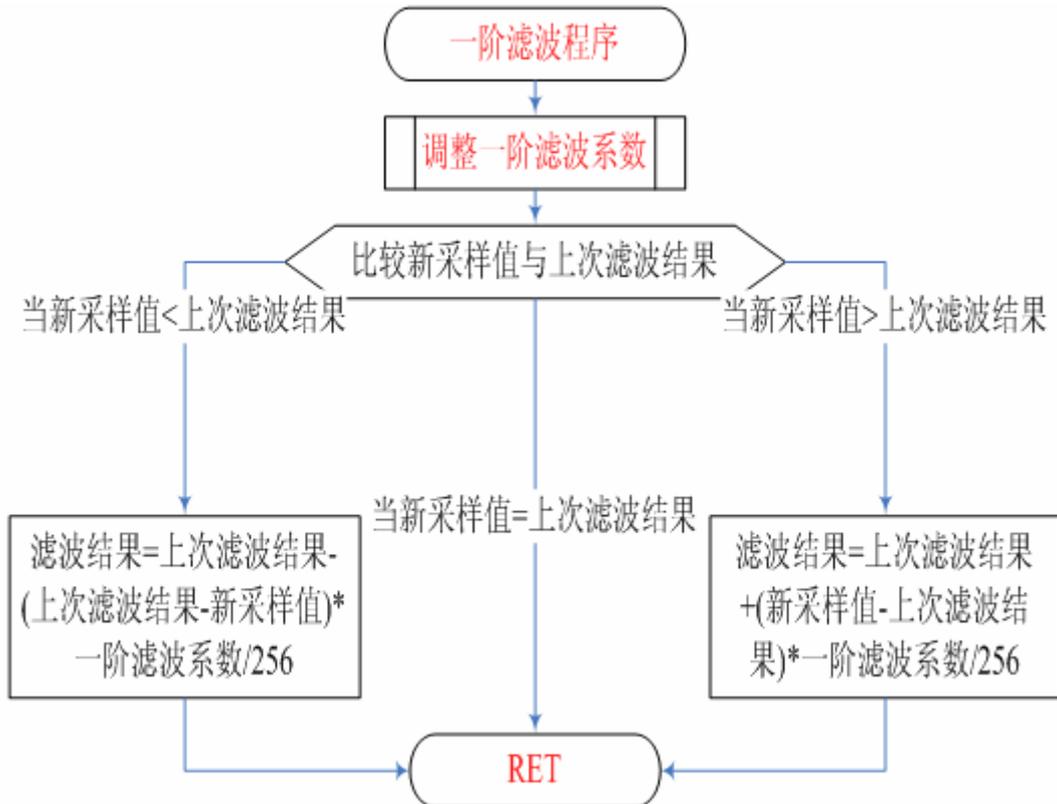
现在，我们给出一个滤波程序的基本流程图。

入口： FILTER\_NEW=新采样值

FILTER\_OLD=上次滤波结果

FILTER\_1ST\_CONST=滤波系数(0~255)(代表新采样值在滤波结果中占的权重)

出口： FILTER\_OLD=本次滤波结果



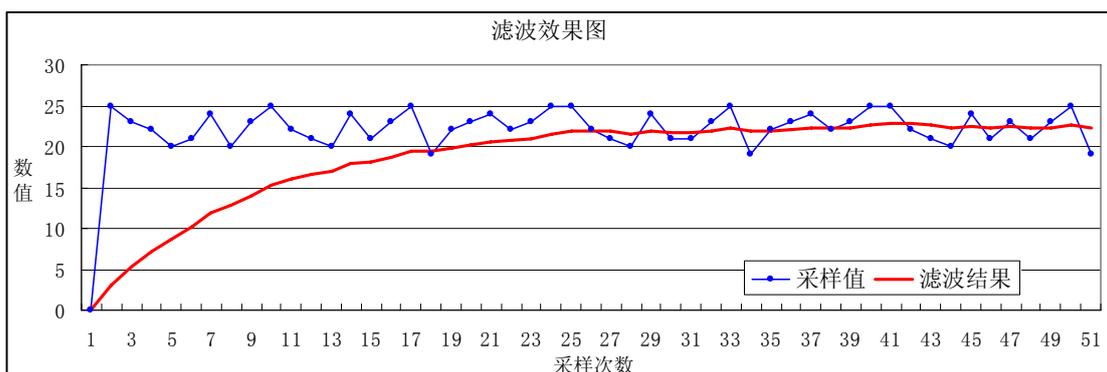
大家也许会注意到这个流程图开始的地方调用了“调整一阶滤波系数”子程序。这是一个更深入的问题，后面会有深入的探讨。在这里，让我们先忽略它吧。

#### 5. 一阶滤波算法的效果

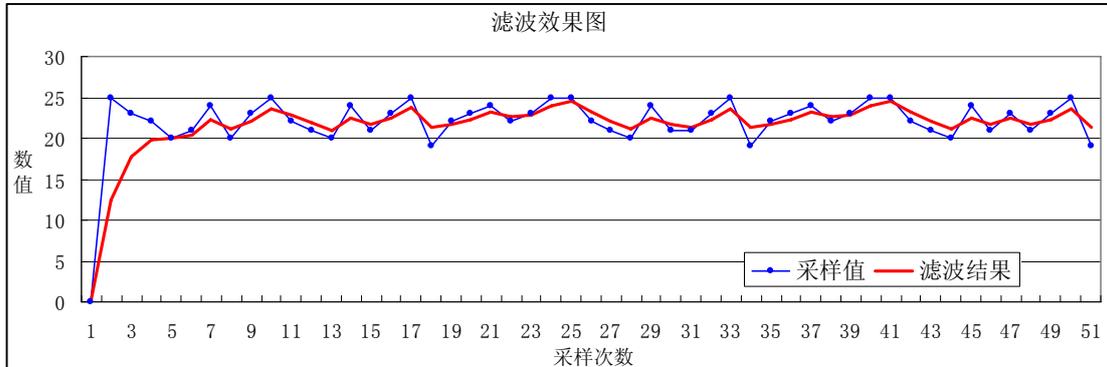
下面，我们用计算机软件来模拟一阶滤波的效果。

在下面的3个图中，兰色线代表采样数据，红色线代表滤波后的数据。

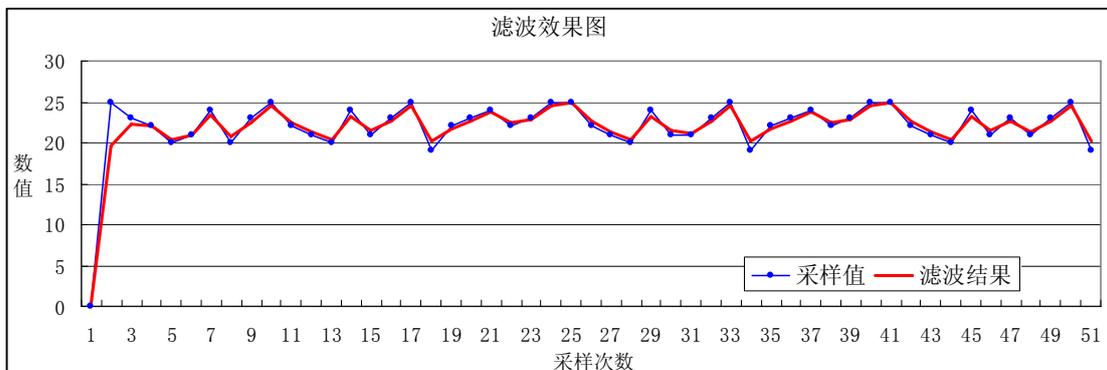
- 当滤波系数=30:



● 当滤波系数=128:



● 当滤波系数=200:



通过三个图的对比可以看出：滤波系数越小，滤波结果越平稳，但是灵敏度越低；滤波系数越大，灵敏度越高，但是滤波结果越不稳定。

由此可见，灵敏度和平稳度似乎是一对矛盾。二者无法完全兼顾。

写到这里，我们已经将一阶滤波的算法讲述清楚。接下来就是如何进一步的玩转它了。

## 6. 一阶滤波算法的不足

● 关于灵敏度和平稳度的矛盾

前面已经讲到了，一阶滤波无法完美地兼顾灵敏度和平稳度。有时，我们只能寻找一个平衡，在可接受的灵敏度范围内取得尽可能好的平稳度。这也许就是程序中折射出来的生活哲理吧。

而在一些场合，我们希望拥有这样一种接近理想状态的滤波算法，即：当数据快速变化时，滤波结果能及时跟进（灵敏度优先）；而当数据趋于稳定，在一个固定的点上下振荡时，滤波结果能趋于平稳（平稳度优先）。

● 关于小数舍弃带来的误差

一阶滤波算法有一个鲜为人知的问题：小数舍弃带来的误差。

比如：

本次采样值=25，上次滤波结果=24，滤波系数=10，

根据滤波算法，本次滤波结果= $(25*10+24*(256-10))/256=24.0390625$

但是，我们在单片机运算中，很少采用浮点数。因此运算后的小数部分要么舍弃，要么进行四舍五入运算。这样一来，本例中的结果 24.0390625 就变成了 24。假如每次采样值都=25，那么滤波结果永远=24。也就是说滤波结果和实际数据一直存在无法消除的误差。这个

误差就是因小数部分的舍弃带来的。

改善误差的办法有两种：

其一，是将滤波系数改大些，当滤波系数 $>128$ 时，可以消除这个问题。但是付出的代价就是降低了平稳度。

其二，就是扩展数据的有效位数。相当于把小数位也参与计算，待最后采信滤波结果时在把小数位消除掉。但是这样做的代价就是让 CPU 背负沉重的运算压力。

## 7. 一阶滤波算法的改进

提出问题的目的是为了分析问题，分析问题的目的是为了解决问题。

虽然这句废话听起来有点像绕口令，但是，既然我们已经知道了一阶滤波算法的种种不足，那么我们就应该尝试着想法去解决。

也许我们可以设计一种算法，去动态地调整一阶滤波的系数。

### ● 动态调整一阶滤波系数的算法应该实现以下功能：

- 当数据快速变化时，滤波结果能及时跟进（灵敏度优先）；并且数据变化越快，灵敏度应该越高。
- 当数据趋于稳定，并在一个固定的点上下振荡时，滤波结果能趋于平稳（平稳度优先）；
- 当数据稳定后，滤波结果能逼近并最终等于采样数据。（消除因小数舍弃带来的误差）。

### ● 在进行调整之前，我们需要先进行以下判断：

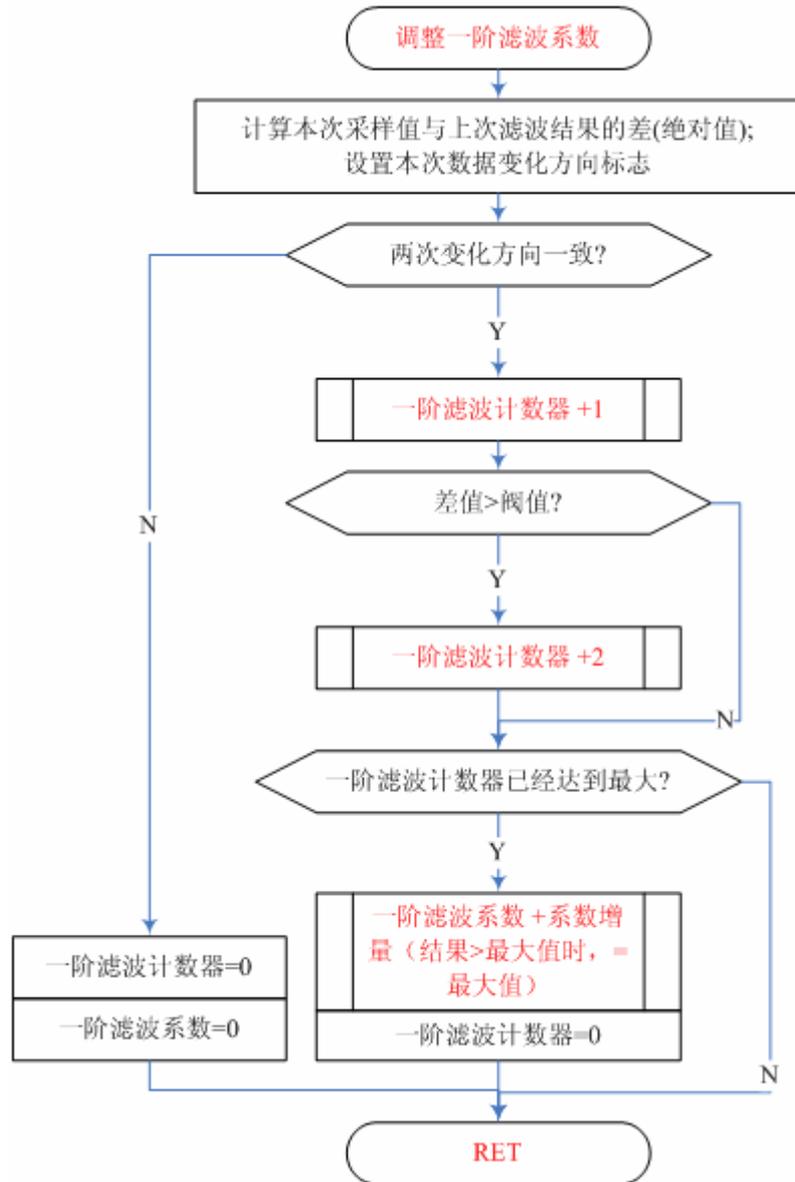
- 数据变化是否朝向同一个方向（比如，当连续两次的采样值都比其上次滤波结果大，视为变化方向一致，否则视为不一致）；
- 数据变化是否较快（主要是判断采样值和上次滤波结果之间的差值）。

### ● 调整的原理如下：

- 当两次数据变化方向不一致时，说明有抖动，将滤波系数清零，忽略本次新采样值；
- 当数据持续向一个方向变化时，逐渐提高滤波系数，提过本次新采样值的权；
- 当数据变化较快(差值 $>$ 消抖计数加速反应阈值)时，要加速提高滤波系数。

### ● 现在，我们给出一个动态调整一阶滤波系数的流程图。

入口：	FILTER_NEW	=新采样值
	FILTER_OLD	=上次滤波结果
	FILTER_1ST_CONST	=上次滤波系数(0~255)
	FILTER_NUM_CH_T	=上次数据变化方向标志(0=减少,1=增加)
	FILTER_ADP_COUNTER	=消抖计数器
出口：	FILTER_1ST_CONST	=本次滤波系数(0~255)
	FILTER_NUM_CH_T	=本次数据变化方向标志(0=减少,1=增加)
	FILTER_ADP_COUNTER	=消抖计数器
常量：	FILTER_ADP_V	=消抖计数加速反应阈值
	FILTER_ADP_MAX	=消抖计数最大值(此常量影响滤波的灵敏度)
	FILTER_1ST_INCREMENT	=滤波系数增量(此常量影响滤波的灵敏度)
	FILTER_1ST_MAX	=滤波系数最大值(此常量影响滤波的灵敏度)



## 8. 相关地址连接

- 《匠人的百宝箱》主站：<http://cxjr.21ic.org>
- 《匠人的百宝箱》副站：<http://cxjr.mcublog.com>
- 《匠人手记》系列原创文章：  
<http://www.mcublog.com/blog/blog2006/cxjr/archives/2006/19158.html>
- 《匠人夜话》系列原创文章：  
<http://www.mcublog.com/blog/blog2006/cxjr/archives/2006/19156.html>



更多精彩文章，尽在《匠人的百宝箱》。网址：<http://cxjr.21ic.org>

## 9. 附录：《匠人手记》系列原创文章列表

访问地址：<http://www.mcublog.com/blog/blog2006/cxjr/archives/2006/19158.html>

- 《匠人手记》01 《分段线性插值法》
- 《匠人手记》02 《用普通 IO 口检测 AD 值的方法（采集温度）》
- 《匠人手记》03 《一阶滤波算法之深入研究》
- 《匠人手记》04 《镍镉\_NiCd\_ 镍氢\_NiMh\_ 电池的充放电特性》
- 《匠人手记》05 《手机锂电池充电特性要求》
- 《匠人手记》06 《几种充电电池性能比较》
- 《匠人手记》07 《巧用移位法进行多字节乘法计算》
- 《匠人手记》08 《巧用移位法进行数制转换》
- 《匠人手记》09 《移相控制与过零检测》
- 《匠人手记》10 《复位电路集锦》
- 《匠人手记》11 《按键漫谈》
- 《匠人手记》12 《单个按键多次击键的检测方法》
- 《匠人手记》13 《解读 TL494 制作的降压电路》
- 《匠人手记》14 《串口七日之创世纪篇》
- 《匠人手记》15 《MC68HC908 应用手记》
- 《匠人手记》16 《天梯——MSP430 之学习札记\_第 1 部\_》
- 《匠人手记》17 《天梯——MSP430 之学习札记\_第 2 部\_》
- 《匠人手记》18 《天梯——MSP430 之学习札记\_第 3 部\_》
- 《匠人手记》19 《天梯——MSP430 之学习札记\_第 4 部\_》
- 《匠人手记》20 《天梯——MSP430 之学习札记\_第 5 部\_》
- 《匠人手记》21 《多种击键类型的处理流程图》
- 《匠人手记》22 《梦幻时钟摇摇棒大揭秘》
- 《匠人手记》23 《单片机入门知识与基本概念》
- 《匠人手记》24 《用单片机 IO 口实现单线单工通讯》



更多精彩内容，尽在《匠人的百宝箱》。网址：<http://cxjr.21ic.org>

---

## 《匠人手记》版权声明

《匠人手记》系列文章都为匠人原创或精心整理，其中耗费了匠人的诸多心血。这些文章推出以来，一直受到网友的欢迎。也有许多网站给予转载和推荐，匠人对此由衷感谢。

但是，最近我们发现，有些网站故意将《匠人手记》的内容拆开来发表，不但隐去了作者和出处等信息，而且还贴上他们自己网站原创的标签。匠人觉得这是一种不尊重原作者的行为。

在此，匠人特声明如下：《匠人手记》的正式发布版本均为 PDF 格式。匠人欢迎各位收藏或转载。但是，匠人要求转载者必须也以 PDF 格式提供。并且，转载者不得对 PDF 格式的《匠人手记》文件内容进行修改。转载者不得利用《匠人手记》谋取经济利益。

大伙能到《匠人的百宝箱》来做客，都是匠人的朋友。希望不要做伤害朋友的事情了。