单片机编程经验

来源：电子产品世界

经验之一：用“软件陷阱+程序口令”对付 PC 指针的弹飞

当 CPU 受到外界干扰,有时 PC 指针会飞到另一段程序中,或跳到空白段去。其实,如果 PC 指针飞到空白段去，倒也好处理。只要在空白段设立软件陷阱(拦截指令),将程序拦截到初始化段或程序错误处理段。但是,如果 PC 指针飞到另一段程序中去了，系统如何办?小匠在这里推荐一种方法——程序口令，思路如下：

1、首先，程序必须模块化。每个模块（子程序）执行一个功能。每个模块只有一个出口（RET）。

2、设立一个模块（子程序）ID 寄存器。

3、为每个子程序配置一个唯一的 ID 号码。

4、每当子程序执行完毕，要返回（RET）之前，先将本子程序的 ID 号送入 ID 寄存器。

5、返回到上级程序后，先判断 ID 寄存器中的 ID 号。

如果正确，则继续执行；如果不正确，则表示 PC 指针有可能已经跳错了，子程序没有按预计的出口返回，这时将程序拦截到初始化段或程序错误处理段。

这种方法，如同在程序中设立了若干个岗哨，每次调用子程序返回后，都要对口令（ID 号），验明正身后再放行。再配合软件陷阱，基本上可以将大多数 PC 指针弹飞的现象检测到。到了程序错误处理段，要杀要剐（冷启动还是热启动）就由您了。

仅以一条代码来揭示程序飞跑的本质！750102H ；MOV 01H，#02H ，如当前 PC 不是指向 75H，而是指向 01H 或 02H，那么 51 内的指令译码器将把她们忠实地翻译成 AJMP X01H 或 LJMP XXXXH 而 XX01H XXXXH 又是什么呢？天知道!这样恶性飞跑下去那还不死定！改革一下：

```
CLR A ； 0C4H
INC A ； 04H
MOV R1， A ； 0F9H
INC A ； 04H
MOV @R1,A ； 86H
```

每一字节代码都不能在生成跳转和循环，且都是单字节指令！往那跑去？跑出去了都要自己回来！“在家”千日好！“跳出”事事难嘛！这样只要平时习惯了用累加器和寄存器把数

倒一倒，把那些危险代码都给倒掉，这样虽说给 PC 的“足”上多加了两字节的“包”可它不好“跑”啊！“足包”====跑！有朋友会问：要是 PC 抓做 02H--LJMP 又有抓做了老鼻子远的 XXH，再抓做隔壁的 YYH 不就没用了吗？提这样的问题只有 ZENYIN 这种钻牛角得才会提！PC 那一位最活跃啊？PC0 啊！要“扯拐”显然发生在她身上，至于那 PC15 同志啊，睡得更死猪一样，雷爆（强干扰）来了都打不醒？此外如果干扰都强到了 PC 高位都出错的地步！关电！关电！不干了！“不是我们不行而是敌人太强大”！反过来要是敌人在你的专政下，只是偶尔出来捣捣乱，但一出来就冲到屁西（PC）高层，就要问问是不是你的王国根基（硬件）有问题了？而非出在意识形态（软件）上！硬件为本！软件为标！标本兼治铸就坚强体魄，方能百毒不侵！

经验之二、不要轻信软件狗

关于软件狗的讨论，论坛上多矣。匠人也曾经查阅过许多关于软件狗的文章。有些大师确实提出了一些比较有技巧性的方法。但是，匠人的忠告是：不要轻信软件狗！其实，软件狗相当于软件的一种自律行为。一般的思路都是通过设立一个计数器，在计时中断中对其+1，在主程序的适当地方对其清零。如果程序失控了，清零指令未被执行，但中断造常发生，则计数器溢出（狗狗叫了）。但是这里有个问题：万一干扰导致中断被屏蔽了，那软件狗就永远不会叫了！——针对这种可能，有人提出在主程序中反复刷新中断使能标志，保证不让中断被屏蔽。——但万一程序飞到某个死循环中去了，不再执行“刷新中断使能标志”这一功能了，还是有可能把狗狗活活饿死。

所以，匠人的观点是：看门狗必须拥有独立的计数器。（即硬件看门狗）好在现在好多芯片都提供了内部 WDT。这种狗都是自带计数器的。即使干扰导致程序失控，WDT 还是会造常计数直到溢出。当然，匠人也没有要将软件狗一棍子全部打死的意思。毕竟不管是软狗还是硬狗，逮到耗子就是好狗嘛（狗拿耗子——多管闲事？）。如果哪位训狗专家确实养过一条能看门的好软件狗，请牵出来让大伙瞧瞧。

经验之三、话说 RAM 冗余技术

所谓的 RAM 冗余，就是：

1、将重要的数据信息备份 2 份（或以上）并存放在 RAM 中不同的区域（指地址不相连）。

2、当平时对这些数据进行修改时，同时也更新备份。

3、当干扰发生并被拦截到“程序错误处理段”中时，将数据与备份做比较，采用表决方式（少数服从多数）选出正确（或可能正确？）的那个。

4、备份越多，效果越好。（当然，你得有足够的存储空间）。

5、只备份最最原始的数据。中间变量（指那些可以从原始数据重新推导出来的数据）不必备份，

注：

1、这种思路的理论依据，据说是源于一种“概率论”，即一个人被老婆打肿脸的概率是很大的，但如果他捂着脸去上班却发现全公司每个已婚男人的脸都青了，这种概率是很小的。同理，一个 RAM 寄存器数据被冲毁的概率是很大的，但地址不相连的多个 RAM 同时被冲毁的概率是很小的。

2、前两年，小匠学徒时，用过一次这种方法，但效果不太理想。当时感觉可能是概率论在我这失效了？现在回想起来，可能是备份的时机选的不好。结果将已经冲毁的数据又备份进去了。这样以来，恢复出来的数据自然也就不对了。

经验之四、话说指令冗余技术

前面有个朋友问到指令冗余，按匠人的理解，指令冗余，就是动作冗余。举个例子，你要在某个输出口上输出一个高电平去驱动一个外部器件，你如果只送一次“1”，那么，当干扰来临时，这个“1”就有可能变成“0”了。正确的处理方式是，你定期刷新这个“1”。那么，即使偶然受了干扰，它也能恢复回来。除了 I/O 口动作的冗余，匠人强烈建议大家在下面各方面也采用这种方法：

1、LCD 的显示。有时，也许你会用一些 LCD 的专用驱动芯片（如 HT1621），这种芯片有个好处，即你只要将显示数据传送给它，它就会不断的自动扫描 LCD。但是，你千万不要以为这样就没你啥事了。正确的处理方式是，要记得定期刷新送显数据（即使显示内容没有改变）。对于 CPU 中自带 LCD DRIVER 的，也要定期刷新 LCD RAM。

2、中断使能标志的设置。不要以为你在程序初始化段将中断设置好就 OK 了。应该在主程序中适当的地方定期刷新一下，以免你的中断被挂起来。

3、其它一些标志字和参数寄存器（包括你自己定义的），也要记得常常刷新。

4、其它一些你认为有必要反复刷新的地方。

经验之五、10 种软件滤波方法

下面奉献——匠人呕心沥血搜肠刮肚冥思苦想东拼西凑整理出来的 10 种软件滤波方法：

1、限幅滤波法（又称程序判断滤波法）

A、方法：根据经验判断，确定两次采样允许的最大偏差值（设为 A），每次检测到新值时判断：如果本次值与上次值之差<=A,则本次值有效。如果本次值与上次值之差>A,则本次值无效,放弃本次值,用上次值代替本次值

B、优点：能有效克服因偶然因素引起的脉冲干扰。

C、缺点：无法抑制那种周期性的干扰，平滑度差。

2、中位值滤波法

A、方法：连续采样 N 次（N 取奇数），把 N 次采样值按大小排列，取中间值为本次有效值。

B、优点：能有效克服因偶然因素引起的波动干扰，对温度、液位的变化缓慢的被测参数有良好的滤波效果。

C、缺点：对流量、速度等快速变化的参数不宜。

3、算术平均滤波法

A、方法：连续取 N 个采样值进行算术平均运算。N 值较大时：信号平滑度较高，但灵敏度较低；N 值较小时：信号平滑度较低，但灵敏度较高。N 值的选取：一般流量，N=12；压力：N=4

B、优点：适用于对一般具有随机干扰的信号进行滤波，这样信号的特点是有一个平均值，信号在某一数值范围附近上下波动。

C、缺点：对于测量速度较慢或要求数据计算速度较快的实时控制不适用，比较浪费 RAM。

4、递推平均滤波法（又称滑动平均滤波法）

A、方法：把连续取 N 个采样值看成一个队列，队列的长度固定为 N，每次采样到一个新数据放入队尾,并扔掉原来队首的一次数据.(先进先出原则)，把队列中的 N 个数据进行算术平均运算,就可获得新的滤波结果。N 值的选取：流量，N=12；压力：N=4；液面，N=4~12；温度，N=1~4

B、优点：对周期性干扰有良好的抑制作用，平滑度高，适用于高频振荡的系统。

C、缺点：灵敏度低 ，对偶然出现的脉冲性干扰的抑制作用较差，不易消除由于脉冲干扰所引起的采样值偏差，不适用于脉冲干扰比较严重的场合，比较浪费 RAM

5、中位值平均滤波法（又称防脉冲干扰平均滤波法）

A、方法：相当于“中位值滤波法”+“算术平均滤波法”。连续采样 N 个数据，去掉一个最大值和一个最小值，然后计算 N-2 个数据的算术平均值。N 值的选取：3~14

B、优点：融合了两种滤波法的优点，对于偶然出现的脉冲性干扰，可消除由于脉冲干扰所引起的采样值偏差。

C、缺点：测量速度较慢，和算术平均滤波法一样，比较浪费 RAM。

6、限幅平均滤波法

A、方法：相当于“限幅滤波法”+“递推平均滤波法”，每次采样到的新数据先进行限幅处理，再送入队列进行递推平均滤波处理。

B、优点：融合了两种滤波法的优点，对于偶然出现的脉冲性干扰，可消除由于脉冲干扰所引起的采样值偏差。

C、缺点：比较浪费 RAM。

7、一阶滞后滤波法

A、方法：取 a=0~1，本次滤波结果=（1-a）*本次采样值+a*上次滤波结果。

B、优点：对周期性干扰具有良好的抑制作用，适用于波动频率较高的场合。

C、缺点： 相位滞后，灵敏度低，滞后程度取决于 a 值大小，不能消除滤波频率高于采样频率的 1/2 的干扰信号。

8、加权递推平均滤波法

A、方法：是对递推平均滤波法的改进，即不同时刻的数据加以不同的权。通常是，越接近现时刻的数据，权取得越大。给予新采样值的权系数越大，则灵敏度越高，但信号平滑度越低。

B、优点：适用于有较大纯滞后时间常数的对象和采样周期较短的系统。

C、缺点：对于纯滞后时间常数较小，采样周期较长，变化缓慢的信号不能迅速反应系统当前所受干扰的严重程度，滤波效果差。

9、消抖滤波法

A、方法：设置一个滤波计数器将每次采样值与当前有效值比较：如果采样值＝当前有效值，则计数器清零如果采样值<>当前有效值，则计数器+1，并判断计数器是否>=上限 N(溢出)，如果计数器溢出,则将本次值替换当前有效值,并清计数器 。

B、优点：对于变化缓慢的被测参数有较好的滤波效果，可避免在临界值附近控制器的反复开/关跳动或显示器上数值抖动。

C、缺点：对于快速变化的参数不宜，如果在计数器溢出的那一次采样到的值恰好是干扰值,则会将干扰值当作有效值导入系统。

10、限幅消抖滤波法

A、方法：相当于“限幅滤波法”+“消抖滤波法” 先限幅,后消抖。

B、优点： 继承了“限幅”和“消抖”的优点改进了“消抖滤波法”中的某些缺陷,避免将干扰值导入系统。

C、缺点：对于快速变化的参数不宜。

IIR 数字滤波器

A. 方法：确定信号带宽，滤之。Y(n) = a1*Y(n-1) + a2*Y(n-2) + . + ak*Y(n-k) + b0*X(n) + b1*X(n-1) + b2*X(n-2) + . + bk*X(n-k)。

B. 优点：高通，低通，带通，带阻任意。设计简单(用 matlab）

C. 缺点：运算量大。