

介绍 **dsp** 知识，为大家提供最新的 **dsp** 资讯，更多内容可以去南京研旭电气科技有限公司的官网 www.njyxdq.com www.f28335.com 或者官方论坛，嵌嵌 **dsp** 论坛 www.armdsp.net 进行交流学习

欢迎大家收听嵌嵌 **dsp** 论坛的官方微博

<http://t.qq.com/qianqiandsp>

目录

第 1 章 绪论.....	2
1 DSP 简介.....	2
第 2 章 总体方案的分析与设计.....	3
2.1 总体方案设计.....	3
2.2 正弦波信号发生器.....	3
第 3 章 硬件设计.....	4
3.1 硬件组成.....	4
3.2 控制器部分.....	5
3.4 人机接口部分.....	6
第 4 章 软件设计.....	7
4.1 流程图.....	7
4.2 正弦信号发生器程序清单.....	8
第 5 章 总结.....	13
参考文献.....	13

第 1 章 绪论

1 DSP 简介

数字信号处理(Digital Signal Processing, 简称 DSP)是一门涉及许多学科而又广泛应用于许多领域的新兴学科。20 世纪 60 年代以来,随着计算机和信息技术的飞速发展,数字信号处理技术应运而生并得到迅速的发展。数字信号处理是一种通过使用数学技巧执行转换或提取信息,来处理现实信号的方法,这些信号由数字序列表示。在过去的二十多年时间里,信号处理已经在通信等领域得到极为广泛的应用。

图一是数字信号处理系统的简化框图。此系统先将模拟信号转换为数字信号,经数字信号处理后,再转换成模拟信号输出。其中抗混叠滤波器的作用是将输入信号 $x(t)$ 中高于折叠频率的分量滤除,以防止信号频谱的混叠。随后,信号经采样和 A/D 转换后,变成数字信号 $x(n)$ 。数字信号处理器对 $x(n)$ 进行处理,得到输出数字信号 $y(n)$,经 D/A 转换器变成模拟信号。此信号经低通滤波器,滤除不必要的高频分量,最后输出平滑的模拟信号 $y(t)$ 。

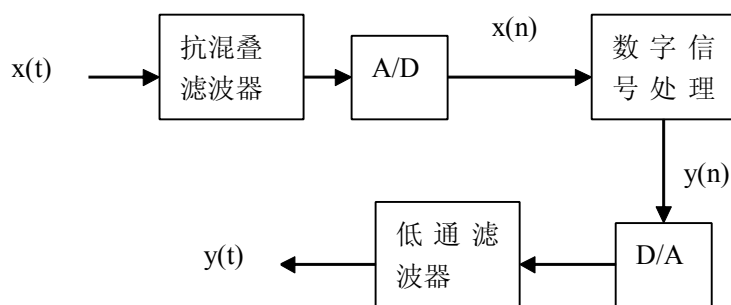


图 1.1 数字信号处理系统简化框图

数字信号处理是以众多学科为理论基础的,它所涉及的范围极其广泛。例如,在数学领域,微积分、概率统计、随机过程、数值分析等都是数字信号处理的基本工具,与网络理论、信号与系统、控制论、通信理论、故障诊断等也密切相关。近来新兴的一些学科,如人工智能、模式识别、神经网络等,都与数字信号处理

密不可分。可以说,数字信号处理是把许多经典的理论体系作为自己的理论基础,同时又使自己成为一系列新兴学科的理论基础。

第 2 章 总体方案的分析和设计

2.1 总体方案设计

- 1.基于 DSP 的特点,本设计采用 TMS320C54X 系列的 DSP 作为正弦信号发生器的核心控制芯片。
- 2.用泰勒级数展开法实现正弦波信号。
- 3.设置波形时域观察窗口,得到其滤波前后波形变化图;
- 4.设置频域观察窗口,得到其滤波前后频谱变化图。

2.2 正弦波信号发生器

正弦波信号发生器已被广泛地应用于通信、仪器仪表和工业控制等领域的信号处理系统中。

通常有两种方法可以产生正弦波,分别为查表法和泰勒级数展开法。查表法是通过查表的方式来实现正弦波,主要用于对精度要求不很高的场合。泰勒级数展开法是根据泰勒展开式进行计算来实现正弦信号,它能精确地计算出一个角度的正弦和余弦值,且只需要较小的存储空间。本次主要用泰勒级数展开法来实现正弦波信号。

产生正弦波的算法正弦函数和余弦函数可以展开成泰勒级数,其表达式:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \dots$$

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \dots$$

取泰勒级数的前 5 项,得近似计算式:

$$\begin{aligned} \sin(x) &= x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} \\ &= x \left(1 - \frac{x^2}{2 \times 3} \left(1 - \frac{x^2}{4 \times 5} \left(1 - \frac{x^2}{6 \times 7} \left(1 - \frac{x^2}{8 \times 9} \right) \right) \right) \right) \\ \cos(x) &= 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} \\ &= 1 - \frac{x^2}{2} \left(1 - \frac{x^2}{3 \times 4} \left(1 - \frac{x^2}{5 \times 6} \left(1 - \frac{x^2}{7 \times 8} \right) \right) \right) \end{aligned}$$

递推公式：

$$\sin(nx) = 2\cos(x)\sin[(n-1)x] - \sin[(n-2)x]$$

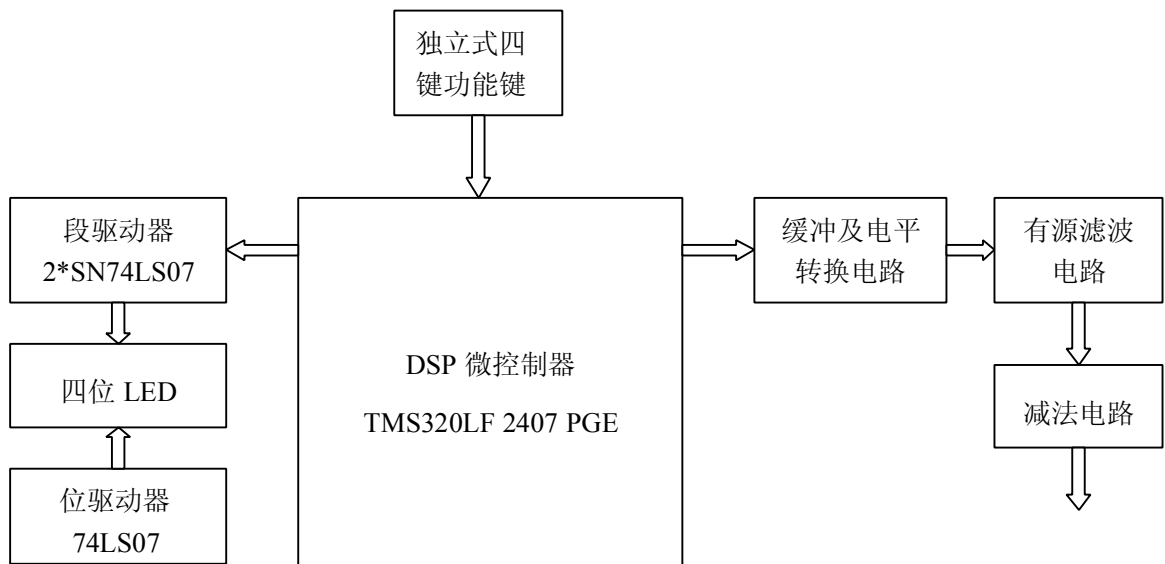
$$\cos(nx) = 2\cos(x)\cos[(n-1)x] - \cos[(n-2)x]$$

由递推公式可以看出，在计算正弦和余弦值时，需要已知 $\cos(x)$ 、 $\sin(n-1)x$ 、 $\sin(n-2)x$ 和 $\cos(n-2)x$ 。

第 3 章 硬件设计

3.1 硬件组成

基于 DSP 的信号发生器的硬件结构图如图 3.1 所示，它主要由 DSP 主控制器，输出 D/A 通道和人机界面等几个主要部分组成。



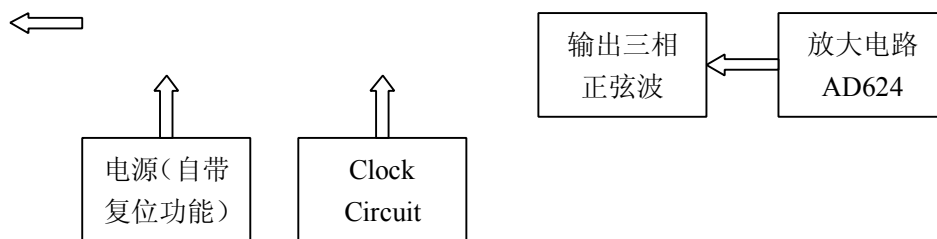


图 3.1 基于 DSP 的信号发生器系统框图

3.2 控制器部分

本系统采用 TI 公司的 TMS320LF2407 DSP 处理器，该器件具有外设集成度高，程序存储器容量大，A/D 转换精度高，运算速度快，I/O 口资源丰富等特点，芯片内部集成有 32KB 的 FLASH 程序存储器、2KB 的数据/程序 RAM，两个事件管理器模块（EVE 和 EVB）、16 通道 A/D 转换器、看门狗定时器模块、16 位的串行外设接口（SPI）模块、40 个可单独编程或复用的通用输入输出引脚（GPIO）以及 5 个外部中断和系统监视模块。

TMS320LF2407 芯片中的事件管理模块（EV）是一个非常重要的组成部分。SPWM 波形的产生和输出就是由这一部分完成的，它由两个完全相同的模块（EVA 和 EVB）组成，每个模块都含有 2 个通用定时器、3 个比较器、6 至 8 个 PWM 发生器、3 个捕获单元和 2 个正交脉冲编码电路（QEP）。由于 TMS320LF2407 有 544 字的双口 RAM（DARAM）和 2K 字的单口 RAM（SARAM）；而本系统的程序仅有几 KB，且所用 RAM 也不多，因此不用考虑存储器的扩展问题，而对于 TMS320LF2407 的 I/O 扩展问题，由于 TMS320LF2407 器件有多达 40 个通用、双向的数字 I/O（GPIO）引脚，且其中大多数的基本功能和一般 I/O 复用的引脚，而实际上，本系统只需要 17 路 I/O 信号，这样，就可以为系统剩余 50% 多的 I/O 资源，因此可以说，该方案既不算浪费系统资源，也为系统今后的升级留有余地。

3.3 微输出 D/A 通道部分

本系统的输出通道部分主要负责实现波形的输出，此通道的入口为 TMS320

LF2407 的 PWM8 口，可输出 SPWM 等幅脉冲波形，出口为系统的输出端，这样，经过一系列的中间环节，便可将 PWM 脉冲波转化为交流正弦波形，从而实现正弦波的输出，其原理框图如图 3.2 所示。

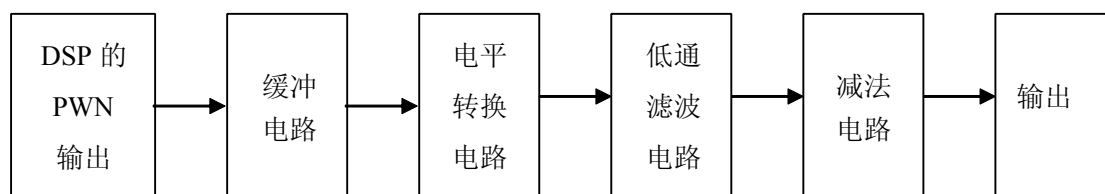


图 3.2 输出通道的原理结构

图 3.2 中的缓冲电路的作用是对 PWM 口输出的数字量进行缓冲，并将电压拉高到 5V 左右，以供后级模拟电路滤波使用。这一部分电路由两个芯片组成。一片用三态缓冲器，由于 PWM 口的输出为 3.3V 的 TTL 电平，这样，在设计时就应当选用输入具有 5V 的 TTL 输入，CMOS 输出电平的转换芯片（如 TI 公司的 74HCT04）；另一片则可选用 TOSHIBA 公司出品的光电耦合器 6N137；输出端连接的 5V 精密稳压电源可选用 BURR-BROWN 公司生产的 REF02 型精密稳压电源，以输出标准的 5V 电压。

系统中的减法电路的主要作用是把 0-10V 直流脉动信号的转换成 -5~+5V 的正弦交流信号，并使其电压增益为 1。设计使可利用差分式电路来实现其功能，为了简化电路，可以选用较为常用的 AD 公司的 AD524，并将 AD524 接成电压跟随器的形式，同时适当的选取电阻以满足要求，此外，为了使产生的正弦波信号具有 2-5mA 的驱动能力，可选用 AD624 来构成末级的信号放大电路。AD624 是高精度低噪声仪用放大器，若外接一只增益电阻，即可得到 1-1000 之间的任意增益值，其误差小于 1%。由于 AD624 的建立时间只有 15 μ s，所以它非常适宜在高速数据采集系统中使用。

3.4 人机接口部分

3.4.1 驱动器设计

位驱动器电路由两片集成电路组成，即由位驱动的 CMOS 芯片和将 TTL 电平转换成 CMOS 电平的电平转换芯片组成，电平转换芯片可以和输出通道的电平转换芯片共用一片 74HCT244（本部分使用 4 路，输出通道使用 3 路），其主要作用是对 DSP 输出的 3.3V TTL 电平与 5V CMOS 电平进行匹配，从而带动具有 CMOS 电平的位驱动器，根据动态扫描显示的要求，位驱动器需要选用每路输出吸收电流都要大于 200mA 的芯片，因此，本设计选用了 TI 公司的 74LS06 来做 LED 的大电流驱动器件。

3.4.2 键盘设计

本系统选用四个独立式按键，分别接入 PF3-PF6 口，并使用四个 220Ω 上拉电阻接 VCC。所谓独立式，就是将每一个独立键按一对一地直接接到 I/O 输入线上，而在读键值时，直接读 I/O 口，每一个键的状态通过读入键值的一位（二进制位）来反应，所以这种方式也称为一维直读方式，这种方式的查键软件比较简单，但占用 I/O 线较多，一般在键的数量较少时采用，不过，由于 DSP 芯片有足够的 I/O 接口可供使用，因而可大大方便设计，设计时可以充分利用这一特点来连接硬件，至于按键的消抖动措施，则可在软件中完成。

第 4 章 软件设计

4.1 流程图

本系统软件可以按照模块化设计思想来编写，包括主程序、常数计算程序、占空比计算程序和相应的一些功能子程序，主程序用于调用各功能子程序、初始化变量、查询键盘、判断显示数据是否需要刷新、同时判断一个脉冲是否完成发送等工作，具体方案见图 4.1 所示的流程图。

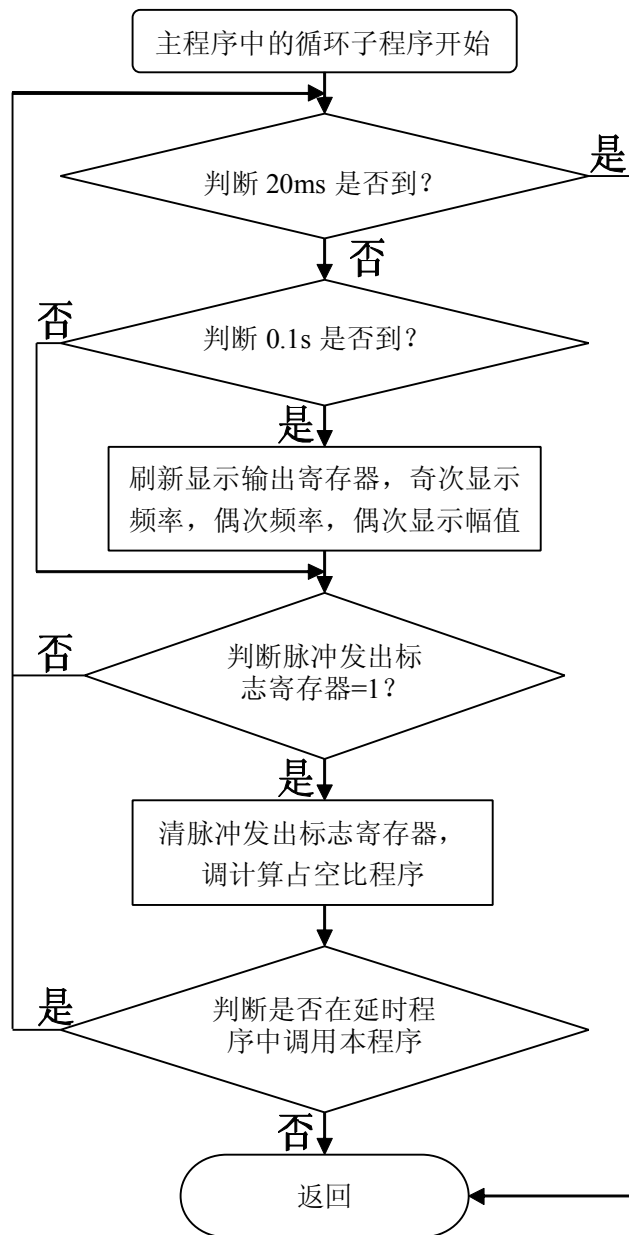


图 4.1 主程序流程图

在程序中，应在第 N-1 个脉冲周期里计算占空比，并在第 N 个脉冲周期里输出波形，这就要求在设计时要在一个脉冲周期内完成计算，如果选用 20MHz 的晶振，那么，在一倍频下，执行一条指令只需 50ns，若输出 400Hz 的正弦波，即每一个周期（即 2.5ms）要输出 200 个脉冲，这样，也就是说，一个脉冲需要 12.5 μ s（相当于 12500/50=250 条指令）。而执行一个占空比的计算程序只需要几十条指令，这种算法从软件开销上考虑是可以实现的。

4.2 正弦信号发生器程序清单

;This function generates the sine wave of angle using the Taylor series expansion

```
;sin(theta)=x(1-x^2/2*3(1-x^2/4*5(1-x^2/6*7(1-x^2/8*9))))
```

```
;cos(theta)=1-x^2/2*3(1-x^2/4*5(1-x^2/6*7(1-x^2/8*9)))
```

```
;sin(2*theta)=2*sin(theta)*cos(theta)
```

```
        .title    "sin.asm"
        .mmregs
        .def      _c_int00
        .ref      sinx,d_xs,d_sinx,cosx,d_xc,d_cosx
sin_x:  .usect   "sin_x",360
STACK:  .usect   "STACK",10
k_theta .set    286
PA0     .set    0
_c_int00

        .text
        STM      #STACK+10,SP
        STM      k_theta,AR0
        STM      0,AR1
        STM      #sin_x,AR6
        STM      #90,BRC
        RPTB    loop1-1
        LDM      AR1,A
        LD       #d_xs,DP
        STL      A,@d_xs
        STL      A,@d_xc
        CALL     sinx
        CALL     cosx
```

```

        LD      #d_sinx,DP
        LD      @d_sinx,16,A
        MPYA   @d_cosx
        STH    B,1,*AR6+
        MAR    *AR1+0
loop1:  STM    #sin_x+89,AR7
        STM    #88,BRC
        RPTB   loop2-1
        LD      *AR7-,A
        STL    A,*AR6+
loop2:  STM    #179,BRC
        STM    #sin_x,AR7
        RPTB   loop3-1
        LD      *AR7+,A
        NEG    A
        STL    A,*AR6+
loop3:  STM    #sin_x,AR6
        STM    #1,AR0
        STM    #360,bk
loop4:  PORTW  *AR6+0%,PA0
        B      loop4
sinx:
        .def   d_xs,d_sinx
        .data
table_s .word  01c7h
        .word  030bh
        .word  0666h
        .word  1556h
d_coef_s .usect "coef_s",4
d_xs     .usect "sin_vars",1
d_sQur_xs .usect "sin_vars",1

```

```
d_temp_s .usect "sin_vars",1
d_sinx .usect "sin_vars",1
c_1_s .usect "sin_vars",1
.text
SSBX FRCT
STM #d_coef_s,AR5
RPT #3
MVPD #table_s,*AR5+
STM #d_coef_s,AR3
STM #d_xs,AR2
STM #c_1_s,AR4
ST #7FFFh,c_1_s
SQR *AR2+,A
ST A,*AR2
||LD *AR4,B
MASR *AR2+,*AR3+,B,A
MPYA A
STH A,*AR2
MASR *AR2-,*AR3+,B,A
MPYA *AR2+
ST B,*AR2
||LD *AR4,B
MASR *AR2-,*AR3+,B,A
MPYA *AR2+
ST B,*AR2
||LD *AR4,B
MASR *AR2-,*AR3+,B,A
MPYA d_xs
STH B,d_sinx
RET
```

COSX:

```
.def      d_xc,d_cosx
d_coef_c .usect  "coef_c",4
        .data
table_c  .word   0249h
        .word   0444h
        .word   0aabh
        .word   4000h
d_xc     .usect  "cos_vars",1
d_sqr_xc .usect  "cos_vars",1
d_temp_c .usect  "cos_vars",1
d_cosx   .usect  "cos_vars",1
c_1_c    .usect  "cos_vars",1
        .text
SSBX     FRCT
STM      #d_coef_c,AR5
RPT      #3
MVPD     #table_c,*AR5+
STM      #d_coef_c,AR3
STM      #d_xc,AR2
STM      #c_1_c,AR4
ST       #7FFFh,c_1_c
SQR      *AR2+,A
ST       A,*AR2
        ||LD   *AR4,B
MASR     *AR2+,*AR3+,B,A
MPYA     A
STH      A,*AR2
MASR     *AR2-,*AR3+,B,A
MPYA     *AR2+
ST       B,*AR2
        ||LD   *AR4,B
```

```
MASR    *AR2-,*AR3+,B,A
SFTA    A,-1,A
NEG     A
MPYA    *AR2+
MAR     *AR2+
RETD
ADD     *AR4,16,B

STH     B,*AR2
RET
.end
```

MEMORY

```
{
PAGE 0:
    EPROM:  org=0E000h,  len=1000h
    VECS:   org=0FF80h,  len=0080h
PAGE 1:
    SPRAM:  org=0060h,   len=0020h
    DARAM1: org=0080h,   len=0010h
    DARAM2: org=0090h,   len=0010h
    DARAM3: org=0200h,   len=0200h
}
```

SECTIONS

```
{
.text    :>EPROM    PAGE 0
.data    :>EPROM    PAGE 0
STACK    :>SPRAM    PAGE 1
sin_vars :>DARAM1   PAGE 1
coef_s   :>DARAM1   PAGE 1
cos_vars :>DARAM2   PAGE 1
coef_c   :>DARAM2   PAGE 1
}
```

```
sin_x      : align(512){ } > DARAM3    PAGE 1
.vectors  :>VECS      PAGE 0
}
.title    "sin_v.asm"
.ref      _c_int00
.sect     ".vectors"
B         _c_int00
.end
```

第 5 章 总结

本次课程设计中遇到一些课堂中从未有过的问题，通过网络查找和同学交流，大大促进了设计进程。并在过程中进一步提高自身的创作、创新水平，扎实基础，扩展所学。并且此次课程设计，基于课程理论知识和网上资料，使我对数字信号处理课程有了更深一步的了解和掌握，对利用 CCS 软件编程的数字信号处理方法有了进一步的了解。在理论课的基础上进行实验实习，是对本门课程的深入学习和掌握，在以后的工作学习中，数字信号的处理都是采用计算机仿真的方法进行测试，因此，掌握基于计算机的数字信号处理方法对以后的工作和学习有很大的帮助。这样一个课程设计对我们的发展有着极大的帮助！

参考文献

- [1] 戴明帧.数字信号处理的硬件实现[M].北京：航空工业出版社，1988
 - [2] 张伟雄，陈亮，徐光辉.DSP 集成开发与应用实例[M]。北京：电子工业出版社
 - [3] 刘湘涛、江世民.单片机原理与应用[M].电子工业出版社，2006.
 - [4] 戴明桢，周建江.TMS320C54XDSP 结构，原理及应运[M]北京航空航天大学出版社
 - [5] 赵红怡.DSP 技术与应用实例.西安:电子工业出版社,2009
- 介绍 **dsp** 知识，为大家提供最新的 **dsp** 资讯，更多内容可以去南京研旭电气科技有限公司的官网 www.njyxdq.com www.f28335.com 或者官方论坛，嵌嵌 **dsp** 论坛 www.armdsp.net 进行交流学习
欢迎大家收听嵌嵌 **dsp** 论坛的官方微博 <http://t.qq.com/qianqiandsp>