

DOI:10.3969/j.issn.1001-4551.2020.02.016

基于同步异构 DSP 的 CLA 模块 的脉冲均分算法研究*

梁磊磊,何高清*

(合肥工业大学 机械工程学院,安徽 合肥 230009)

摘要:针对数控系统中精插补器输出脉冲分布不均匀的问题,对精插补器的运行机制、硬件结构与软件算法进行了研究,对精插补器的传统解决方案与输出脉冲分布不均匀的原因进行了归纳,提出了一种基于同步异构数字信号处理器(DSP)的控制律加速器(CLA)模块的脉冲均分算法。以使用相同时钟信号、数据总线与指令总线的 DSP 与 CLA 为硬件构架,采用并行处理任务、硬件触发 CLA 程序的模式,使用改进的双脉冲周期插补算法,构建了能够输出分布均匀的脉冲的精插补器;使用 TMS320F28377S 为硬件平台,利用其在线调试功能,对该脉冲均分算法进行了测试。实验结果表明:在数控系统的循环调度周期中,该设计方案输出脉冲的时间可以达到 100%,脉冲周期最短可以达到 2 μs ;该脉冲均分算法可以实现输出脉冲分布的均匀化。

关键词:数字信号处理器;控制律加速器;脉冲均分算法;精插补器

中图分类号:TH-39;TP368.2

文献标识码:A

文章编号:1001-4551(2020)02-0191-05

Pulse uniformization algorithm of CLA module based on synchronous heterogeneous DSP

LIANG Lei-lei, HE Gao-qing

(School of Mechanical Engineering, Hefei University of Technology, Hefei 230009, China)

Abstract: Aiming at the problem of uneven distribution of output pulse by the fine interpolation in the numerical control system, experimental study on the operation mechanism, hardware structure and software algorithm of the fine interpolator were summarized, the traditional solution of the fine interpolator and the reason about uneven distribution of output pulse were comparatively analyzed. A method of pulse uniformization algorithm based on the control law accelerator (CLA) module of synchronous digital signal processor (DSP) was proposed. A fine interpolator that could output distributed pulse evenly was constructed based on the hardware architecture with DSP and CLA in same clock signal, data bus and instruction, the running mode of processing task in parallel with running program in CLA by hardware trigger and the improved algorithm of double pulse period interpolation. A method was presented to test the pulse uniformization algorithm by hardware platform with TMS320F28377S and its online debugging function. The experimental results indicate that the time of output pulse occupy 100% of loop scheduling period in the CNC system by the design, and the pulse period can be as short as 2 μs , achieve the uniformization of distribution about output pulse.

Key words: digital signal processor (DSP); control law accelerator (CLA); pulse uniformization algorithm; fine interpolation

收稿日期:2019-06-16

基金项目:合肥市远大轴承锻造有限公司项目(W2016JSKF0039)

作者简介:梁磊磊(1994-),男,硕士研究生,主要从事数控技术与嵌入式系统方面的研究。E-mail:278195005@qq.com

通信联系人:何高清,男,副教授,硕士生导师。E-mail:hegq2008@163.com

0 引 言

插补算法的质量直接影响数控系统的精度和速度,是数控机床的重要技术指标。插补器是数控系统用来完成运动轨迹拟合的程序或硬件。数控系统的插补算法大致分为:脉冲当量法与数字积分法^[1]。(1)脉冲当量法适用对速度与精度要求不高,以步进电机为主要控制对象的开环数控系统^[2];而现在的数控系统大都采用数字积分法插补,相较于脉冲当量插补,数字积分法拥有更高的插补速度与精度。(2)数字积分法包括粗插补与精插补,其中粗插补是将给定的轨迹分割为较大的段,完成插补的预处理;精插补是将粗插补算出的直线段再细分,进行插补运算,并形成最后的脉冲输出,从而精确地控制电机的运动。精插补的关键在于实现输出脉冲的均匀化,若只采用 CPU 完成精插补,则会消耗 CPU 大量的时间资源,影响其他通讯与控制程序的执行,降低系统的实时性。为了提高系统的实时性,同时完成输出脉冲均匀化的运动控制,当前主流的解决方案是由主 CPU 完成粗插补,再由 CPLD、FPGA 或运动控制卡等模块,独立完成精插补程序^[3-8]。但是,这种结构增加了控制系统的硬件与软件的复杂度,开发难度大,不利于在空间要求高的紧凑型数控系统中使用。

本研究提出一种基于同步异构 DSP 的 CLA 模块的脉冲均分插补器(CLA 是 TMS320F28377 芯片中可以独立于 CPU 运行的内嵌模块,通过主控制器与 CLA 模块并行处理任务,即主控制器发送指令触发 CLA 软中断,CLA 模块上运行脉冲均分算法执行精插补器的任务,输出脉冲;这种结构无需外接其他控制组件,可降低系统的复杂度,提高系统的可靠性与实时性^[9])。

1 CLA 模块原理与特性

内置于同步异构 DSP 的模块,通过引入并行处理的功能,扩展了主 CPU 的能力,用于对时间要求严格的控制,实现了更快的系统响应和更高频率的控制回路;通过将 CLA 用于对时间敏感的关键任务,可以释放主 CPU 去执行其他系统任务和通信,从而提高了系统的实时性。

目前,为了实现精插补过程中,保持输出脉冲的均匀化,同时不降低系统的实时性,往往采用 CPU 外接运动控制模块来实现。相较于传统的解决方案,采用同步异构数字信号处理器的 CLA 模块来实现脉冲均分算法,具有以下优势:

(1)CLA 模块与 CPU 共用系统时钟源,即拥有相同的时钟速率,无需为外置模块配置独立的时钟信号源;

(2)CLA 模块建立于独立的构架,其任务由 CPU 软件触发或由硬件中断触发,无需配置复杂的数据与控制总线;

(3)CLA 模块拥有 GPIO 等外设的操作权限,这些权限由 CPU 分配给 CLA,因此 CLA 可以绕过 CPU 直接控制 GPIO,不必通过 CPU 来控制运动控制模块输出脉冲,减少了消息传递的跨度,提高了系统的可靠性与实时性;

(4)CLA 模块与 CPU 的程序都可以采用 C 语言编写,且可以在相同的开发环境下进行程序的编写、调试与下载,降低了开发难度,提高了开发效率与系统的可维护性。而如 DSP + FPGA、ARM + CPLD 等方案,需要针对每个主芯片,采用不同的开发方式与环境^[10],难度较大的 VHDL 等硬件描述语言,更是增加了控制系统的研发成本与时间。

2 精插补脉冲均匀化算法的设计

精插补脉冲均匀化算法是服务于数字积分法的精插补部分。由于计算机系统是离散的,在精插补的过程中,直线段细分的结果可能得不到整型数,导致计算机无法处理。在计算机中,最直接的方法是对结果进行取整,但这样会导致在每个插补周期脉冲周期偏小,即脉冲周期内会有一段时间没有脉冲产生,即为空行程,从而导致了脉冲不均匀,会对电机运行的稳定性产生消极的影响。

以插补周期为 4 ms,GPIO 输出脉冲最小翻转时间为 1 us 为例,则有:

$$N_{\max} = \left\lfloor \frac{T_1}{2T_{\min}} \right\rfloor = 2\ 000 \quad (1)$$

式中: N_{\max} —插补周期内的最大脉冲数; T_1 —插补周期时间; T_{\min} —输出脉冲最小翻转时间。

插补周期内的脉冲周期为:

$$T_p = 2 \times \left\lfloor \frac{T_1}{2N} \right\rfloor, N \in [0, 2\ 000] \quad (2)$$

式中: N —插补周期内输出的脉冲数量。

脉冲输出的时间占插补周期的比值可以量化输出脉冲均匀化的水平,其值愈小,则输出脉冲愈不均匀,其数学表述为:

$$\mu = 1 - \delta = 1 - \frac{T_p \times N}{T_1} = 1 - \frac{N}{2\ 000} \left\lfloor \frac{2\ 000}{N} \right\rfloor \quad (3)$$

式中: μ —脉冲不均匀水平; δ —脉冲均匀化水平。

通过式(3),可得脉冲不均匀水平 μ 与脉冲数 N 的

关系,如图 1 所示。

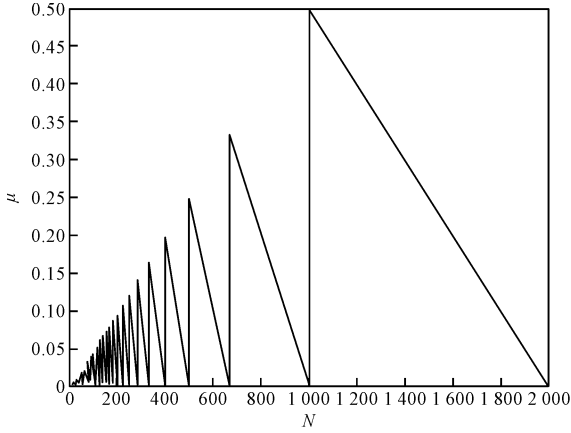


图 1 脉冲不均匀水平 μ 与脉冲数 N 的关系

由图 1 可知:在脉冲数 $N = 1\ 001$ 时,脉冲不均匀水平 μ 可达到最高值 0.499 5;且在 N 的多数取值中都出现了脉冲不均匀现象,故有必要采取算法消除脉冲不均匀的现象。

通常情况下,若输出脉冲的周期变化小于驱动器允许脉冲输入的最小周期时,就可以认为输出脉冲周期是均匀的。以此为依据,在插补周期内,设置 2 个脉冲周期,两者之间相差 1 个输出脉冲最小周期,即驱动器允许脉冲输入的最高频率为 500 kHz,则有方程组:

$$\begin{cases} N_1 + N_2 = N \\ T_{P1} \times N_1 + T_{P2} \times N_2 = T_1 \\ T_{P2} = T_{P1} + T_B \\ T_{P1} = 2 \times \left[\frac{T_1}{2N} \right] \end{cases} \quad (4)$$

解得:

$$\begin{cases} T_{P1} = 2 \times \left[\frac{T_1}{2N} \right] \\ T_{P2} = T_{P1} + T_B \\ N_1 = \frac{1}{T_B} (N \times T_{P2} - T_1) \\ N_2 = \frac{1}{T_B} (T_1 - N \times T_{P1}) \end{cases} \quad (5)$$

式中: T_{P1} —脉冲周期 1; N_1 —周期为 T_{P1} 的脉冲数; T_{P2} —插补周期 2; N_2 —周期为 T_{P2} 的脉冲数。

将 $T_B = 2\ \mu\text{s}$ 与 $T_1 = 4\ \text{ms}$ 代入式(5),可得:

$$\begin{cases} T_{P1} = 2 \times \left[\frac{2\ 000}{N} \right] \\ T_{P2} = T_{P1} + 2 \\ N_1 = \frac{N \times T_{P2}}{2} - 2\ 000 \\ N_2 = 2\ 000 - \frac{N \times T_{P1}}{2} \end{cases} \quad (6)$$

本文通过设置这 2 个插补周期 T_{P1} 与 T_{P2} ,进行精插补运算,可以在算法层面彻底消除脉冲输出脉冲不均匀的现象。

3 基于 CLA 模块的脉冲均分算法实现

基于同步异构 DSP 的 CLA 模块,本文构建的双脉冲周期精插补脉冲均匀化算法的硬件系统结构图如图 2 所示。

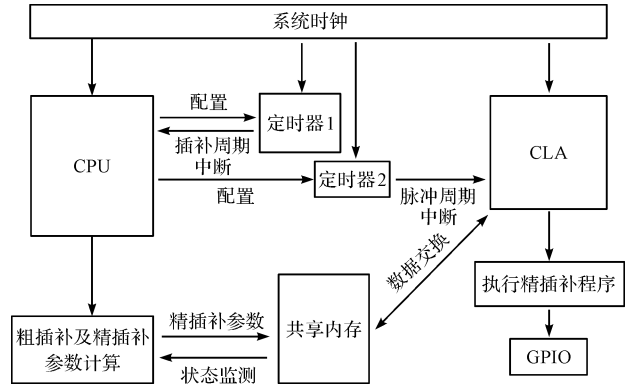


图 2 硬件系统结构

由图 2 可知:控制系统同时为 CPU 与 CLA 提供时钟源与存储空间,通过主 CPU 配置定时器 2 的定时中断,为 CLA 模块提供插补的基准频率,触发 CLA 模块来执行精插补运算。同时,控制系统分配给 CLA 模块控制 GPIO 的权限,用来输出插补的脉冲。

TMS320F28377 的定时器 1 的定时中断周期设置为 4 ms,用于提供插补周期,由主 CPU 来执行该中断服务程序;定时器 2 的定时中断周期为 1 μs ,为最小脉冲的电平翻转时间,即基准频率,由此得出精插补器输出的最小脉冲周期为 2 μs 。

CLA 的插补任务由定时中断来触发,在定时器 2 的每个定时中断服务程序中,CLA 可以选择脉冲的发送使能或停止,从而控制输出脉冲的周期;执行精插补程序变量存储在 CPU 与 CLA 的共享内存区域,CPU 与 CLA 都可以直接读取与修改这些变量,以加快数据传输的速度。

在每个插补周期中,精插补中 2 个插补周期的长度与脉冲数,由主 CPU 计算得出,再传递给 CLA,这部分的 C 语言源程序如下:

```
PulseA = 2 000 / N;
PulseB = PA + 1;
CountA = (PulseA + 1) * N - 2000;
CountB = 2 000 - PulseA * N;
```

其中: N —插补周期内需要输出的脉冲数; $PulseA$ —脉冲的电平翻转时间 1; $PulseB$ —脉冲的电平

翻转时间 2; $CountA$ —翻转时间为 $PulseA$ 的脉冲计数; $CountB$ —翻转时间为 $PulseB$ 的脉冲计数。

使用 CLA 模块,实现脉冲均分算法的流程简图如图 3 所示。

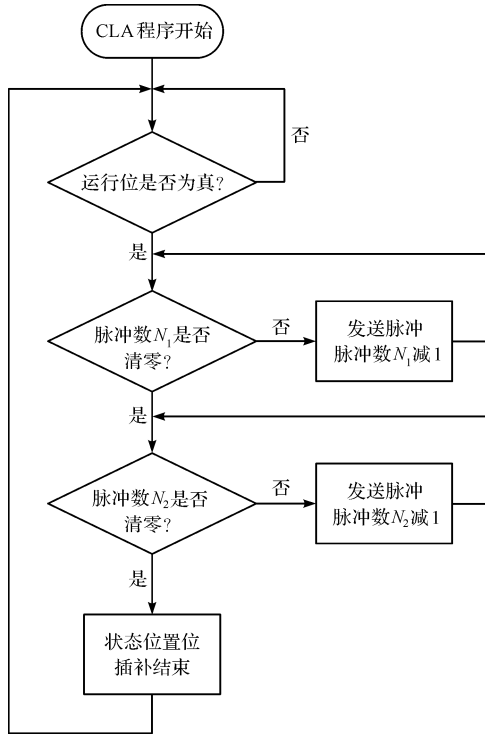


图 3 脉冲均分算法的流程简图

由图 3 可知:CLA 通过接受主 CPU 通过共享内存传递过来的参数,来执行精插补。

当一个插补周期开始时,设置运行位为真,CLA 便以定时器 2 的定时中断为基准,开始执行 2 个插补周期与脉冲数的插补指令,插补完成以后,以状态位置位来通知主 CPU 当前的插补周期任务已经完成,CLA 接着开始等待下一个插补周期的任务。

4 仿真与实验

本文使用 CCS(code composer studio)软件的在线调试功能,以 TMS320F28377S 为硬件实验平台,在插补周期为,GPIO 输出脉冲的最小周期为的条件下,选取插补周期内输出的脉冲数 $N = 1\ 001$ 与 667 这两个会导致脉冲不均匀水平较大的取值,进行单脉冲周期精插补算法,并与基于 CLA 的双脉冲周期精插补算法进行对比实验。

当 $N = 1\ 001$ 时,使用单脉冲周期精插补算法后,脉冲在插补周期内的分布情况如图 4 所示(这种情况下脉冲分布的不均匀程度最严重,空行程时间占据了脉冲周期的 49.95%)。

由图 4 可知:在整个插补周期内,脉冲只分布在 $0 \sim 2\ ms$ 这一时间段,在该时间段内已经输出了 1 001

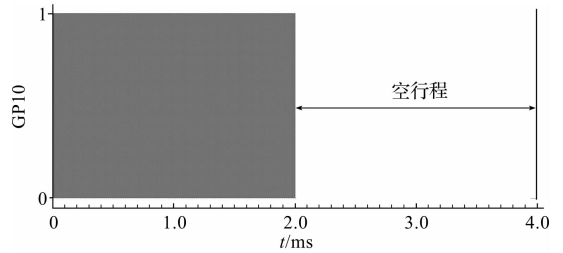


图 4 当 $N = 1\ 001$ 时使用单脉冲周期精插补算法后脉冲在插补周期内的分布情况

个脉冲;在 $2\ ms \sim 4\ ms$ 时间段内没有脉冲输出,即空行程时间占脉冲周期的比例约为 50%。由此可见,在误差允许的范围内,该结果与理论结果(空行程时间占据了脉冲周期的 49.95%)相符合。

当 $N = 667$ 时,使用单脉冲周期精插补算法后,脉冲在插补周期内的分布情况如图 5 所示。

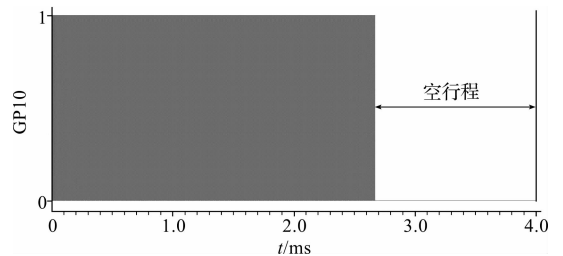


图 5 当 $N = 667$ 时使用单脉冲周期精插补算法后脉冲在插补周期内的分布情况

由图 5 可知:在整个插补周期内,闲置时间占据了脉冲周期的比例约为 $1.33/4 = 33.25\%$ 。由此可见,在误差允许的范围内,该结果与由式(3)得出的理论值 33.33% 相符合。

这些实验结果证明,采用单脉冲周期精插补算法会出现输出脉冲分布不均匀的情况。

理论上,采用改进的双脉冲周期精插补脉冲均匀化算法后,在的取值范围内,脉冲输出时间都会占据整个插补周期的 100%,从而可以消除输出脉冲不均匀的现象。

由式(6)可得,当 $N = 1\ 001$ 、667 时,脉冲均分算法的运行参数如表 1 所示。

表 1 当 $N = 1\ 001$ 、667 时脉冲均分算法的运行参数

N	N_1	N_2	$T_{p1}/\mu s$	$T_{p2}/\mu s$
1 001	2	999	2	4
667	1	666	4	6

当 $N = 1\ 001$ 时,使用双脉冲周期精插补算法后,脉冲在插补周期内的分布情况如图 6 所示。

当 $N = 667$ 时,使用双脉冲周期精插补算法后,脉冲在插补周期内的分布情况如图 7 所示。

从图(6,7)可以看出:在 $N = 1\ 001$ 与 667 的情况下,输出脉冲占据了插补周期的全部时间,没有空行

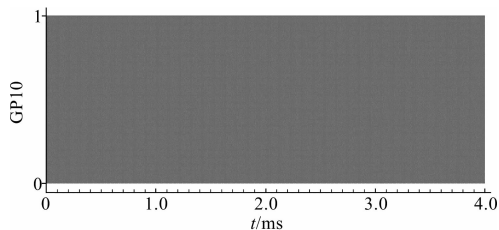


图6 当 $N=1001$ 时使用双脉冲周期精插补算法后脉冲在插补周期内的分布情况

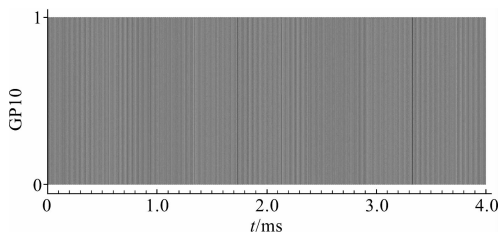


图7 当 $N=667$ 时使用双脉冲周期精插补算法后脉冲在插补周期内的分布情况

程,达到了输出脉冲均匀化的效果。

当 $N=1001$,使用双脉冲周期精插补算法时,脉冲在插补周期起始时间段的输出波形如图8所示。

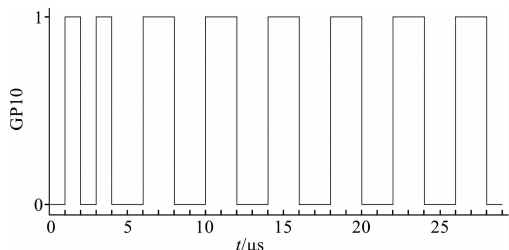


图8 $N=1001$ 时使用双脉冲周期精插补算法时脉冲在插补周期起始时间段的输出波形

由图8可知:当 $N=1001$ 时,输出脉冲由起始2个周期为 $2\mu\text{s}$ 的脉冲与后续周期为 $4\mu\text{s}$ 的脉冲组成。

当 $N=667$,使用双脉冲周期精插补算法时,脉冲在插补周期起始时间段的输出波形如图9所示。

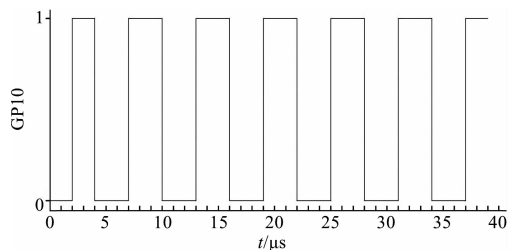


图9 $N=667$ 时使用双脉冲周期精插补算法时脉冲在插补周期起始时间段的输出波形

由图9可知:当 $N=667$,输出脉冲由起始1个周期为 $4\mu\text{s}$ 的脉冲与后续周期为 $6\mu\text{s}$ 的脉冲组成。

将以上结果与表1中的数据相比较可知,脉冲均分算法的实际输出结果与理论的一致,说明该算法在实际硬件中能够正确运行,且达到了预期的效果。同时,实验结果说明了能够基于同步异构 DSP 的 CLA 模块的硬件结构,实现双脉冲周期精插补脉冲均分算法,使插补周期中没有闲置的空行程,达到了输出脉冲均匀化的目的。

5 结束语

本研究通过脉冲均分算法,借助同步异构 DSP 的 CPU 与 CLA 协同处理任务的能力,设计了数控系统中用于运动控制的精插补器,实现了插补过程中脉冲的均匀化输出。

该系统硬件与软件结构简单、稳定可靠,可以在不借助外部逻辑电路的情况下(如 CPLD、FPGA 等),达到与之相同的效率与实时性。

笔者在研究过程中发现该系统还有不足之处,即只在 CLA 上实现了精插补,粗插补还是由主 CPU 来完成。因此,在今后的研究中,可以将粗插补器也移植到 CLA 中,以进一步提高硬件的使用效率与系统的实时性,达到更好的效果。

参考文献(References):

- [1] 廖效果. 数控技术[M]. 湖北:湖北科学技术出版社,2000.
- [2] 王广炎,张润孝,王小椿. 脉冲均匀化插补方法[J]. 制造技术与机床,2000(3):37-38.
- [3] 施群,王小椿. 基于高速均匀脉冲分配算法的数控系统精插补方法研究[J]. 工业仪表与自动化装置,2004(2):69-72.
- [4] 郑云华. 基于 SOPC 的嵌入式数控系统控制算法的研究与实现[D]. 长沙:中南大学计算机学院,2011.
- [5] 张卿杰,徐友. 手把手教你学 DSP—基于 TMS320F28335[M]. 北京:北京航空航天大学出版社,2015.
- [6] 朱春光,龙佑喜,盛德军. 基于 DSP 和 CPLD 的数控插补技术研究[J]. 机电一体化,2002,8(4):24-26.
- [7] 陈宏臣. 基于 FPGA 技术的数控插补器算法改进研究[D]. 成都:西南交通大学机械工程学院,2010.
- [8] 王汉霆. 基于 FPGA 的运动插补控制芯片的研究与设计[D]. 武汉:华中科技大学机械科学与工程学院,2011.
- [9] 寇玉泉,程树康. 交流伺服电机及其控制[M]. 北京:机械工业出版社,2008.
- [10] 石岛胜. 小型交流伺服电机控制电路设计[M]. 北京:科学出版社,2012.

[编辑:程浩]

本文引用格式:

梁磊磊,何高清. 基于同步异构 DSP 的 CLA 模块的脉冲均分算法研究[J]. 机电工程,2020,37(2):191-195.

LIANG Lei-lei, HE Gao-qing. Pulse uniformization algorithm of CLA module based on synchronous heterogeneous DSP[J]. Journal of Mechanical & Electrical Engineering, 2020,37(2):191-195.

《机电工程》杂志: <http://www.meem.com.cn>