

采用 DS80C400 建立网络多媒体应用

...无需任何硬件辅助的图像压缩，DS80C400 便可每秒传输 4 帧原始的黑白图像 (240 x 180)。

令人振奋的多媒体应用——包括有线广播 (PA) 系统、网络门、MP3 播放器以及安全摄像机等——可以采用一种低成本的网络型微处理器建立。本文将讨论如何在音视频范例系统中应用 DS80C400 网络型微控制器。

建立网络型 PA 系统

可以这样描述一个 PA 系统 (也称为“头顶传呼系统”)，接线员在上面广播一些诸如此类的消息“所有工作人员请注意，正在测试火警系统”，或“主管人员，请到化学品库报到”。这个装置采用独立的布线和基础结构，并常常使用一些在晶体管出现之前就已有的技术。试想一下，如果将这个系统搬到网络上情况会怎样。不仅可以省掉独立的音频布线，而且还可以使系统变得更加智能。例如，可以将传呼系统连接至建筑物的访问控制系统或某个网络服务器，它们更有可能知道某个雇员所处的位置。这种计算机化的 PA 系统还能够自动重复某个消息，使接线员能够解放出来去处理更多的呼叫。另外，系统也可以接入公司范围内的邮件系统，实现邮件至语音转换服务，或通过一个能够输入传呼请求的网站，发布消息，而无须人工干预。

怎样才能建立这样一种 PA 系统呢？首先，至少需要一台运行 web 界面、电子邮件网关并带有麦克风或相似设备的服务器。我们称这台服务器为“主控”。下一步，需要一些扬声器模块。这些模块是一些网络单元，具有能够驱动扬声器的数模转换器 (DAC)。这些扬声器单元的价格要求低廉，而且要求现场安装相当容易。

图 1 展示了两座建筑之内的网络草图，包括 7 个扬声器单元和一个主控服务器。建筑物之间的网络连接采用了一个路由器，而不是网桥 (这样做的主要好处在于软件方面，随后将在文中描述)。在本例中，DS80C400 网络微控制器驱动扬声器单元。虽然微控制器不具备最新 PC 系统

的处理能力和存储资源，不过，这类 PA 系统也没有很高的带宽和处理能力需求。采样率在 22.05kHz x 8 位的非压缩单声道音频带宽需求在 180kbps 以内，足以提供优异的话音质量。而且也不用为硬件解压缩支付成本。

图 2 说明了网络音频系统的相对带宽需求。即使在早期的 (半双工) 10Mb 网络上，相对于 5Mbps 的有效带宽，音频应用也仅使用了 4% 的容量。如今的大多数以太网至少为 100Mbps。

扬声器硬件

除 DS80C400 外，扬声器单元还需要一些储存器 (512kB 的 SRAM 就足够了)、网络 PHY、DAC、放大器和扬声器。完整的方案

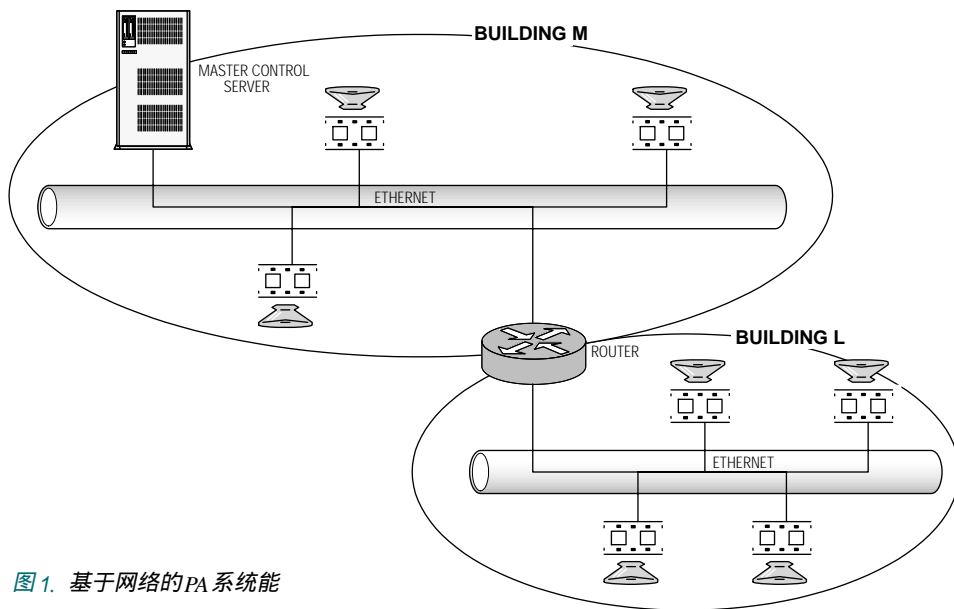


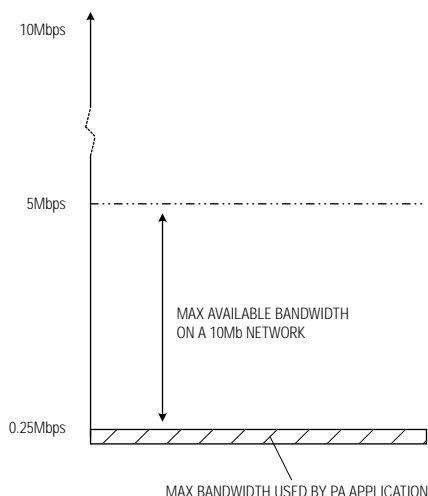
图 1. 基于网络的 PA 系统能够覆盖多座建筑。

描述参见 *Application Note 609: Internet Speaker with the DS80C400 Silicon Software* (可从 www.maxim-ic.com/appnoteindex 下载)。

下列技术简化了安装和代码的分发：

- DS80C400 NetBoot (参见 *DS80C400 网络应用程序的升级*)
- DHCP省去了IP配置，降低了安装和配置成本
- 以太网供电(见 *以太网供电*)简化了布线，降低了材料成本

上电后，DS80C400 ROM通过DHCP获取IP地址，接着查询网络，以获取最新版的应用程序。应用程序执行后，系统就准备接收音频数据。在现场找一个未使用的网口，连上电缆，就完成了新扬声器模块的安装。



DS80C400 网络应用程序的升级

DS80C400 配备了网络引导能力，可非常方便地装载代码到一个新设备中。“NetBoot”通过串行装载机中的N命令激活。支持SRAM和闪存。

网络引导方式优于人工装载一个新设备。在一个没有非易失存储器，如闪存的设计中，在经历一次断电之后，这是一种最简单的安装应用程序的方式。NetBoot还能够自动检查网络以升级和安装代码(如果网络不存在，则运行先前安装的“旧”代码)。NetBoot采用DHCP和TFTP网络协议获取IP地址和装载程序数据。网络配置也可以静态设置，并存储在1-Wire器件中。支持IPv6地址。

为TINI400评估模块/接口建立一个自动网络代码升级功能，要求以下步骤(详细情况，包括所支持的1-Wire器件，参见 *High-Speed Microcontroller User's Guide: DS80C400 Supplement* 于 www.maxim-ic.com/microcontrollers)：

- 1) 确认网络上存在DHCP服务器，或分配一个静态IP地址，并存储于1-Wire器件。
- 2) 安装一个TFTP服务器(如tftpd)，并或者通过DHCP公布其IP地址，或记录地址于1-Wire器件。
- 3) 以 *tbini2* 格式将代码上传至TFTP服务器，文件名 `TINI400`。
- 4) 设置TINI400上的NetBoot跳线器。
- 5) 此后每次系统复位时，将自动从TFTP服务器重装或更新代码。

图2. 与目前的以太网带宽相比，网络音频所要求的带宽极低。

扬声器软件

为简化硬件的安装，软件需要做一些额外的工作。由于两座建筑之间存在路由器(图1)，广播消息不能够直接从一座建筑到达另一座建筑。所以，不能采用简单的消息广播。一个新的扬声器必须发送组播消息(参见 *组播*)，直到主控确认了该扬声器的位置和参数。一个新的扬声器系统事先无法知道主控的位置，因而也需要发送组播消息来请求主控确认自己。如果考虑安全问题，则交换信息可以采用数字签名，以排除那些冒充服务器的系统。一旦采用传统的单播消息配置后，该扬声器就加入一个组播群，等待音频包。这些音频包由主控组播。网络

PA系统的范例程序采用C语言编写(见开发DS80C400)。下列代码能够接收通过组播套接字传送过来的网络音频数据。

```
#define MULTICAST_IP_MSB 239
#define MULTICAST_IP_2 192
#define MULTICAST_IP_3 0
#define MULTICAST_IP_LSB 22
#define MULTICAST_PORT 6789

int s; /* socket handle */
struct sockaddr address; /* IP address */
unsigned char xdata buffer[1600];

/* Step 1: Create the socket */
s = socket(PF_INET, SOCK_DGRAM, 0);
/* Step 2: Join the multicast group */
memset(address, '\0', sizeof(address));
address.sin_addr[12] = MULTICAST_IP_MSB;
address.sin_addr[13] = MULTICAST_IP_2;
address.sin_addr[14] = MULTICAST_IP_3;
address.sin_addr[15] = MULTICAST_IP_LSB;
address.sin_port = MULTICAST_PORT;
join(s, &address, sizeof(struct sockaddr));

/* Step 3: Listen for incoming packets */
memset(address, '\0', sizeof(address));
address.sin_port = MULTICAST_PORT;
bind(socket_handle, &address, sizeof(struct sockaddr));

/* Step 4: Now we're ready to receive data */
recvfrom(s, buffer, 1600, 0, &address, sizeof(struct sockaddr));
/* We actually received data! We could play it. */
printf("Data received:\r\n");
...
```

网络型 PA 系统可以连接至建筑物的访问控制系统或某个网络服务器，它们知道某个雇员所处的位置，可自动重发一条消息，或者通过一个能够输入传呼请求的网站发布消息，而无须人工干预。

除 `join()` 之外，这些步骤对于那些为 TCP/IP 网络编写代码的程序员来说，应该是相当熟悉的。该函数调用非常类似于 Posix 和 Winsock (注意：`xdata` 是 Keil C 的一个关键字，它告诉编译器在哪里分配数据)。在 DS80C400 中，`join()` 和 `leave()` 函数分别用来启动或终止组播群中的成员。在本例和下面的例程中，所有库函数调用——`bind()`、`recvfrom()` 等——都返回状态码。不同于本简化例程，检查这些返回码中的出错信息并作出相应的响应，是一种可取的做法。

以太网供电

连接设备到网络中有一个不利的方面——它要求给网络添加额外的电缆。幸好，电源可以被整合到以太网电缆中空余的导线上。解决这个问题有多种方法，最为常见的是 IEEE802.3af 标准，规定将 48V 电源连接至 8 引脚以太网连接器的 7、8 引脚(+)和 4、5 引脚(GND)。48V 电源普遍应用于电话系统，因此配线柜中通常已带了电源。

为了能够应用于微控制器，电源必须降压至某个合适的水平。参见 www.maxim-ic.com/appnoteindex，相关应用笔记描述了如何用 MAX5910 和 MAX5014 构建一个高效的 IEEE802.3af 兼容电路。

文本到语音

使用麦克风记录现场语音，或播放存储在网络服务器上的录音，是应用PA系统的方式之一。另外一种用法是利用接收自电子邮件、web页或蜂窝电话短消息业务的文本广播消息。

为系统添加语音合成非常容易。转换可以直接在主控服务器上进行，利用文本到语音转换引擎从输入文本产生音频波形。然后就可以将这个波形如同其它音频信号一样发送至扬声器，扬声器模块无须作任何改动。

文本至语音转换引擎已广泛应用，并且已经成为有些操作系统(例如 Mac OS X)的组成部分。在 freetts.sourceforge.net/上也可以找到一个免费的Java语音引擎。商用方案听起来更自然一些，在 www.nuance.com上可尝试一下“Vocalizer”演示版，带有美国、英国或澳大利亚口音。

娱乐级音频

如果音频采样以非压缩方式传输，则CD音频就会遇到问题。以44.1kHz x 16位采样、未经处理的立体声需要1.4Mbps的网络带宽(几乎占了10Mb网络的30%)，已经超出了许多网络可提供的带宽。

象MP3这样的压缩算法能够降低数据速率，由此大大提高网络负载能力，使系统切实可行。配合硬件解压缩芯片，DS80C400能够轻松完成任务。事实上，36MHz的时钟速率足以连续播放192kb的MP3。www.mp3elf.net网上的TINI MP3项目就是一个基于Dallas微控制器的网络MP3设计。原理图及全套的多媒体应用都可找到。

组播

DS80C400 ROM和TINI运行环境支持组播。不同于从一个源发往一个目标的单播数据包，组播允许多个目标主机接收相同的数据，从而消除了重复传输，节省了带宽。组播不同于简单的广播。采用IGMP协议，一个主机可注册为一个或多个组播群的成员，组播业务只被路由到那些至少存在一个接收方的网络部分。不同于要求网络桥接的广播消息，路由器能够传递组播消息。

在网络中，组播包采用一类特殊的目标IP地址，也称为组播群。从应用的角度来看，增加对于组播的支持轻而易举。调用 `join()` 函数便可将主机加入组播群，可以象其它UDP业务一样接收数据包。当不再需要接收消息时，好的习惯是调用 `leave()` 函数离开组播群(当加入至少一个组播群后，主机会定期发送成员报告。如果主机在离开群之前崩溃，其在群内的成员资格最终也会被取消)。当为自己的应用选择组播群时，一定要避免重复分配，遵从RFC 2365的指导方针。

成像

也可以将视频信号连接至DS80C400。这样的系统将十分有用，如安全照相机，采用一个廉价的照相机每秒获取一个快照，并传送到网上用于显示和存储。服务器端的后处理可完成运动监视并警示保安人员。

比较好的照相机选择是现代的蜂窝电话——它们不仅小巧、廉价而且随处可得。大多采用一种串行协议进行通信，但因制造商而异。在连接至某个特定相机之前，确认你已经得到了所有需要的技术信息。

网络相机系统实例：
一个网络相机系统范例、
原理图和代码、可从站点
[ftp://ftp.dalsemi.com/pub/tini/ds80c400/reference_designs/netcam/](http://ftp.dalsemi.com/pub/tini/ds80c400/reference_designs/netcam/)找到。

实验表明，无须任何硬件辅助的图像压缩，DS80C400便可每秒传输4帧原始的黑白图像(240 x 180)，而且还留有一定的余量可用于处理话音级的音频。下列代码样例打开了一个至服务器的TCP连接用于传送一页图片(尽管我们显示了连接的关闭，实际应用可保持连接开通以降低开销)。

```
int s; /* socket handle */
struct sockaddr address; /* IP address */
unsigned char xdata buffer[1600];

/* Step 1: Create the socket */
s = socket(PF_INET, SOCK_STREAM, 0);

/* Step 2: Fill in the target address 192.168.0.11 */
memset(address, '\0', sizeof(address));
address.sin_addr[12] = 192;
address.sin_addr[13] = 168;
address.sin_addr[14] = 0;
address.sin_addr[15] = 11;
address.sin_port = 8080; /* Target port */

/* Step 3: Connect to the server */
connect(s, &address, sizeof(address));

/* Step 4: Send the data in buffer */
send(s, buffer, sizeof(buffer), 0);

/* Step 5: Close the connection */
closesocket(s);
```

对于那些熟悉Unix网络编程的人来说，`closesocket()`就是`close()`。DS80C400版本的`close()`函数由文件系统使用。与Windows系统一样，DS80C400上的套接字句柄与文件句柄是不可互换的，必须使用独立的套接字函数。

相机系统提供给DS80C400的时钟频率为73.7MHz，接近于75MHz的极限值。73.7MHz的频率由基频模式的18.432MHz晶体和集成在DS80C400上的PLL经4倍频后得到。这种设计降低了整个系统的成本，同时还允许工作在接近微控制器最高频率的上限。此外，18.432MHz x 4对于异步串行通信来讲还是一个很好的波特率发生器。

网络门

将安防照相机与双向音频、按钮和蜂鸣器等结合在一起是很容易做到的。这样的系统允许我们建造一个网络门(图3)。应用是无止境的，尤其是与访问控制和安全登录等结合在一起时。

对于DS80C400来说，按钮和蜂鸣器仅仅是外部设备，可以直接连接到普通I/O上。在Keil C中，通过`sfr`和`sbit`，很容易定义I/O端口：

```
/* Define port 1 */
sfr p1 = 0x90;
/* Define P1.7 (port 1 is bit addressable) */
sbit p1_7 = p1^7;

/* Toggle P1.7 */
p1_7 = !p1_7;
```

采用 iButton 和内置于 DS80C400 的 1-Wire 主控制器接口，很容易在网络门上添加安全认证功能(某种程度上讲，该接口的编程更要复杂一些，因此 Dallas Semiconductor 提供了库文件，以便简化任务)。关于说明如何连接一个电动门的实例参见 www.ibutton.com/TINI/applications/lock/。

最后提示：象网络门这样的系统可能需要完成多种处理(或任务)，DS80C400 ROM 包含了一个任务调度器。下面实例说明了如何在 C 语言中使用它。还有，在工业级应用中，应该检查返回代码。

```
unsigned char pri, task;

/* Get the current task */
task = task_getcurrent();

/* The current task's priority */
pri = task_getpriority(0);

/* Decrease the priority */
task_setpriority(0, pri-1);

/* Sleep */
task_sleep(0, 0, 500);
```

样例程序还包含了一些有用的函数，如 `task_fork()`，通过复制当前任务，创建一个新的任务。`task_kill()` 取消一个任务，以及 `task_suspend()` 挂起一个任务。这些功能以及其他一些功能在网站上的下述资料中有所描述：*High-Speed Microcontroller User's Guide: DS80C400 Supplement*。

结论

小巧、价廉的网络微控制器可以作为诸多有趣实用的多媒体应用的强大内核。我们鼓励读者使用 DS80C400，应用或改进本文所描述的设计思想。寻求免费样品请访问 www.maxim-ic.com。

开发 DS80C400

可以采用多种方法开发 DS80C400 的应用软件。若想快速评估方案，可以考虑采用 Java 语言和 TINI 运行环境。对于那些要求考虑每个指令周期的应用，最好采用最优化的汇编语言。

在本文中，我们采用了 C 语言。Keil C 编译器(www.keil.com)支持 DS80C400 连续的 24 位寻址模式(参考 *Application Note 606: Configuring Keil PK51 Tools to Support 24-Bit Contiguous Addressing Mode*)，允许高达 16MB 的代码/数据存储空间。为了使用这种模式，要求采用扩展版本的编译器和连接器(CX51、LX51)。这些工具包含在专业开发者工具中(PK51)。

Dallas Semiconductor 提供接口至 DS80C400 内部网络栈的 C 库文件。在 Dallas Semiconductor 的 ftp 网站：<ftp://ftp.dalsemi.com/pub/tini/ds80c400/> 上，可以找到该库文件，以及一份使用 Keil 开发环境创建 DS80C400 项目的详细操作指南。这些库文件极大地简化了网络编程。例如，创建一个 TCP 连接已简化为全球皆知的 `socket()` 和 `connect()` 顺序调用。寻求技术支持，请加入 TINI 邮件讨论组 lists.dalsemi.com。

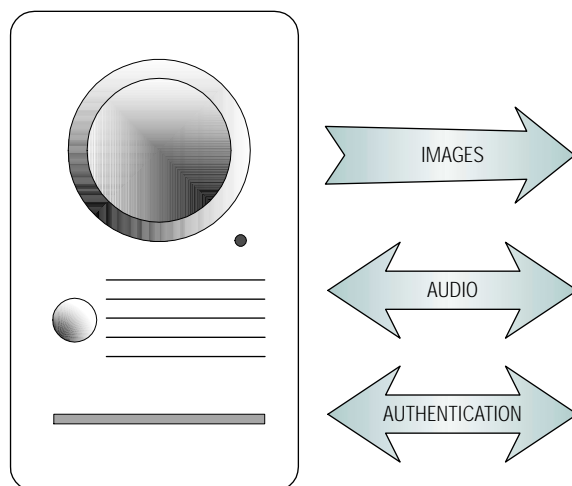


图3. 网络门接口包括了视频、音频和访问控制。