

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 17/30 (2006.01)



[12] 发明专利申请公布说明书

[21] 申请号 200610170664.8

[43] 公开日 2008年7月2日

[11] 公开号 CN 101211353A

[22] 申请日 2006.12.27

[21] 申请号 200610170664.8

[71] 申请人 北京海尔集成电路设计有限公司

地址 100088 北京市海淀区花园路四号通恒大厦 205 室

[72] 发明人 史岩 张继达 敖钧

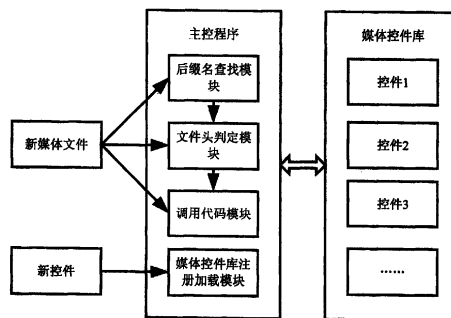
权利要求书 1 页 说明书 4 页 附图 2 页

[54] 发明名称

一种识别多格式媒体文件的方法

[57] 摘要

本发明提出一种准确、高效识别媒体文件是否能被播放的方法，同时，该方法还提出了防止用户误删媒体控件程序以及便于媒体控件版本升级的机制。包括新媒体文件、新控件、主控程序及媒体控件库；主控程序中包含后缀名查找模块、文件头判定模块、调用代码模块以及媒体控件库注册加载模块；媒体控件库中包含对应各种媒体文件的媒体控件。通过在主控程序中直接判断媒体文件的后缀名及媒体文件头，提高对媒体文件格式的识别率；通过媒体控件文件头格式的规定，使得调用代码模块的调用次数降低，提高媒体控件程序的效率；通过媒体控件库注册加载模块的使用，有效地对媒体控件进行管理，使媒体控件具有防误删除及方便地进行版本更新的功能。



-
- 1、一种识别多格式媒体文件的方法，包括主控程序及媒体控件库，其中，主控程序包含后缀名查找模块、文件头判定模块、调用代码模块以及媒体控件库注册加载模块，媒体控件库中包含对应各种媒体文件的媒体控件，其特征在于：所述主控程序发现新媒体文件和新控件后，由后缀名查找模块及文件头判定模块完成新媒体文件和新控件格式的识别；媒体控件库注册加载模块将媒体控件库的存储目录属性设置为“系统文件”和“隐藏文件”，以避免用户对媒体控件程序的无意删除。
 - 2、根据权利要求1所述的一种识别多格式媒体文件的方法，其特征在于：所述媒体控件库注册加载模块规定了媒体控件的格式，就是在获取的新控件头部即解码代码的前面增加了8个字节的媒体文件后缀名、4个字节的媒体控件版本号及长度可变的媒体文件头信息。

一种识别多格式媒体文件的方法

技术领域

本发明涉及多媒体播放技术，尤其是一种识别多格式媒体文件的方法及对这些媒体文件进行译码、播放的方法。

背景技术

随着多媒体技术、存储技术、传输技术的迅猛发展，媒体文件的种类层出不穷。多种图片格式、声音格式、图像格式、网络文档格式、动画格式等不断涌现，而且，对于特定的应用，会出现综合上述几种格式的媒体文件，又大大增加了媒体文件格式的种类。

传统的嵌入式设备一般都只是针对特定的媒体文件进行设计，一般只能播放特定格式的文件，导致下述三个问题：第一是系统的可扩展性不好，无法支持其他的媒体文件；第二是难以改正以前版本文件中的格式错误，通过升级虽然可以部分解决上述问题，但软件升级对于用户来讲操作复杂，且存在丢失程序的风险；第三是开发速度慢，如果要求支持的文件格式较多，需要在产品上市之前完成所有的代码，往往消耗很多时间。

传统的识别多格式媒体文件的方法是根据文件的后缀名判定该文件是否可播放，但对于具有相同后缀名而文件种类不同的情况，则无法处理，如果仅仅根据后缀名判断文件是否可以播放，会使用户在不能播放时产生误解，同时，调用播放程序需要一定的时间，影响用户的使用。

目前已公开的技术中，也有一些针对多格式媒体文件进行识别的方法，这些方法在判断文件后缀名的同时，还判断媒体文件的文件头，提高了对媒体文件格式的识别率，这些方法往往是通过建立译码程序库或模块解决上述问题，但存在的问题是：1、不能有效避免用户对译码程序的无意破坏；2、所建立的译码程序库或模块升级困难；3、调用译码程序库或模块影响媒体文件的播放效率。

如已公开的中国专利文献 CN1749953 中提出了一种多格式媒体文件识别方法，通过格式辨识模块判断媒体文件的后缀名或文件头，再通过译码程序调用模块调用相应的译码程序对媒体文件进行解码。该方法可以部分地解决对媒体文件识别率低下的问题，却不能完全解决上述现有技术中存在的所有问题。

发明内容

本发明的目的在于针对各种媒体文件格式纷繁且新媒体格式不断涌现的现状，提出一种准确、高效识别媒体文件是否能被播放的方法，同时，该方法还提出了防止用户误删媒体控

件程序以及便于媒体控件版本升级的机制。

本发明的技术方案如下：一种识别多格式媒体文件的方法，包括主控程序及媒体控件库，其中，主控程序包含后缀名查找模块、文件头判定模块、调用代码模块以及媒体控件库注册加载模块，媒体控件库中包含对应各种媒体文件的媒体控件，主控程序发现新媒体文件和新控件后，由后缀名查找模块及文件头判定模块完成新媒体文件和新控件格式的识别；媒体控件库注册加载模块将媒体控件库的存储目录属性设置为“系统文件”和“隐藏文件”，以避免用户对媒体控件程序的无意删除。所述媒体控件库注册加载模块规定了媒体控件的格式，就是在获取的新控件头部即解码代码的前面增加了 8 个字节的媒体文件后缀名、4 个字节的媒体控件版本号及长度可变的媒体文件头信息。

本发明的积极效果在于：通过在主控程序中直接判断媒体文件的后缀名及媒体文件头，提高对媒体文件格式的识别率，使得调用代码模块的使用次数降低，提高媒体控件程序的效率；通过媒体控件库注册加载模块的使用，有效地对媒体控件进行管理，使媒体控件具有防误删除及方便地进行版本更新的功能。

附图说明

- 图 1 为系统结构图；
- 图 2 为媒体文件识别流程图；
- 图 3 为控件程序存储目录结构图；
- 图 4 为播放控件格式示意图；
- 图 5 为媒体文件后缀名格式示意图；
- 图 6 为媒体文件头格式示意图；
- 图 7 为媒体文件头中存储数据结构图。

具体实施方式

本实施例所述的一种识别多格式媒体文件的方法，包含主控程序、媒体控件库，见图 1，其中主控程序中具有后缀名查找模块、文件头判定模块、调用代码模块及媒体控件库注册加载模块；媒体控件库中是已加载的各种媒体播放控件。主控程序发现新媒体文件和新控件后，由后缀名查找模块及文件头判定模块完成新媒体文件和新控件格式的识别；媒体控件库注册加载模块将媒体控件库的存储目录属性设置为“系统文件”和“隐藏文件”，以避免用户对媒体控件程序的无意删除。

如图 2 所示，当一个新媒体文件需要播放时，首先通过后缀名查找模块查找媒体控件库中是否有与未知的媒体文件后缀名一致的媒体控件，如果没有相应的媒体控件，则直接报告

无法播放信息并退出；如果媒体控件库中具有与后缀名对应的媒体控件，再判断媒体文件的文件头是否与经后缀名判断过的媒体控件中存储的文件头数据一致，如果不一致，则报告无法播放信息并退出；如果一致，说明媒体控件库中有后缀名及文件头与待播放的媒体文件的后缀名及文件头完全一致的媒体控件，将媒体控件中的代码对媒体文件进行译码，则可实现媒体文件的播放。

为防止用户在使用过程中误删媒体控件库中的文件，本实施例实现了对媒体控件的文件管理功能。如图 3 所示，在存放媒体控件的存储介质的根目录下建立一个标识属性为“系统文件”和“隐藏文件”的目录 System，这个目录在系统中是不可见的，用来存贮媒体控件，只有注册程序和加载程序才能访问这个目录，这个目录“系统文件”和“隐藏文件”的属性使得播放控件不会被用户无意破坏。当新的媒体控件出现后，用户首先需要对该媒体控件进行注册，将相应的媒体控件导入到 System 目录。媒体控件的后缀名为 hll，包含媒体文件的播放信息，它可以来源于任何外部设备或网络。

为提高对媒体文件识别的效率，同时为方便媒体控件的升级，本实施例中重新定义了媒体控件的格式，如图 4 所示，在解码代码的前面增加了 8 个字节的媒体文件后缀名、4 个字节的媒体控件版本号及长度可变的媒体文件头信息。

媒体文件后缀名固定为 8 个字节，以小端的 Unicode 码存贮，可以存贮最多 4 个字符。如果媒体的后缀名小于 4 个字符，不足的部分以 0x00、0x00 替代。如图 5 所示，如果媒体文件的后缀名为.hav，则存储为“.HAV”的 ASCII 代码：0x68、0x00、0x61、0x00、0x76、0x00、0x00、0x00。

版本号为 4 个字节，以 32 位整型小端方式存储，版本号用于在更新媒体控件时，判断媒体控件是否需要更新。在媒体控件注册时，在媒体控件中的后缀名与媒体文件的后缀名相同的情况下，如果发现新的媒体控件的版本号高于当前媒体控件的版本号，则注册程序会自动将新的媒体控件覆盖原有媒体控件。

媒体控件格式中的第三部分为媒体文件头信息，如图 6 所示，媒体文件头由三部分组成，前四个字节标示媒体文件头数据的长度，以 32 位整形小端方式存储，并以位长度而不是字节长度表示，当整个数值不是 8 的整数倍时，最后一个字节的剩余部分以 0 填充；第二部分为媒体文件头数据；第三部分为补 0。

媒体文件头信息所起的作用是判断媒体文件是否可以播放。如图 7 所示，如果一种媒体文件的文件头为 8 个字节：0x55、0xAA、XX、0x00、0x00、0x0X、0xFF、0xFF，其中 X 表示这个媒体文件的文件头在此位上是不确定的，用二进制展开后就是：01010101 10101010 XXXXXXXX 00000000 00000000 0000XXXX XXXXXXXX 11111111，对于这样的一种文件

头，我们用二进制 00 表示 0，二进制 11 表示 1，二进制 01 表示 X，则上面的文件头就变成了 0011001100110011 1100110011001100 0101010101010101 0000000000000000 0000000000000000 000000001010101 0101010101010101 1111111111111111，即为媒体文件头信息的存储数据。

媒体控件格式中的最后一部分是解码代码，为解码的核心部分。解码代码对于每种媒体格式均不相同，只是将媒体文件解码到特定的缓冲区，而由主程序完成解码后的媒体文件的音视频输出。

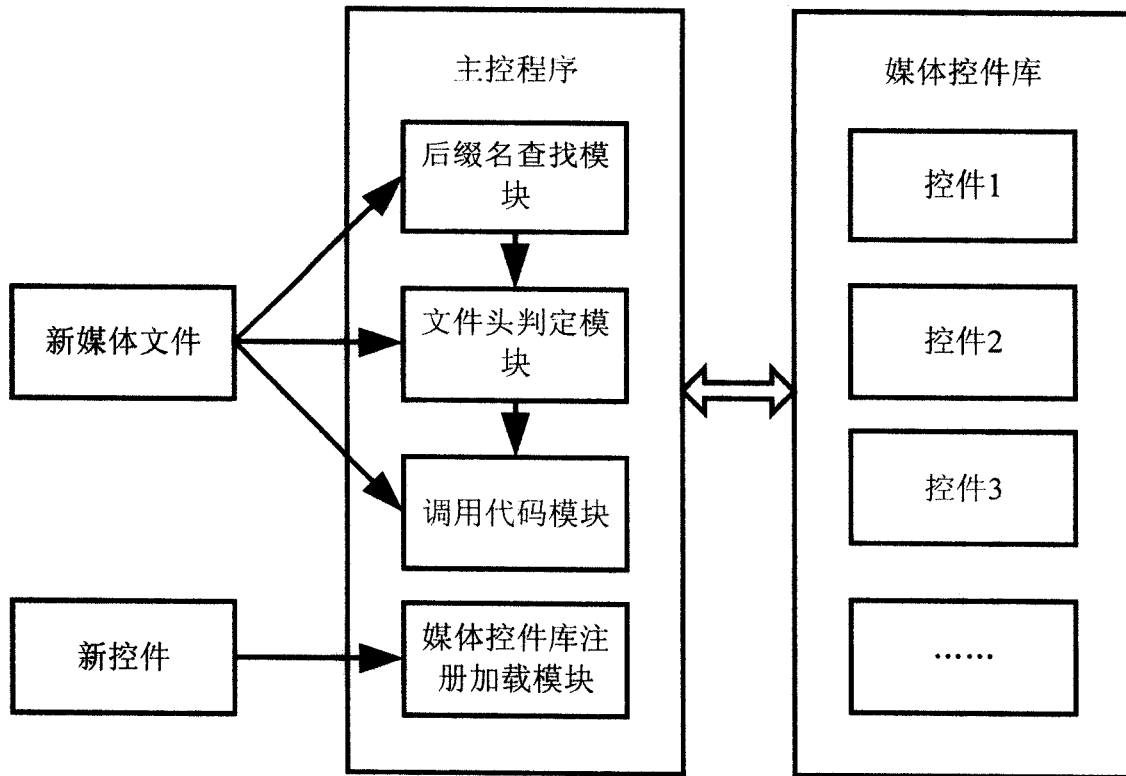


图 1

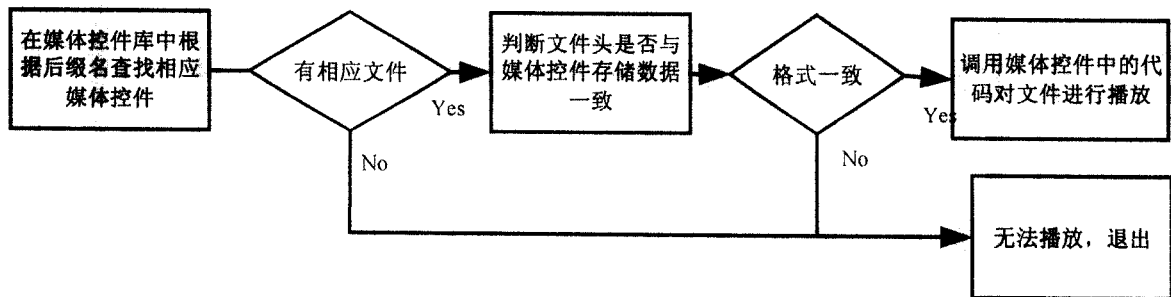


图 2

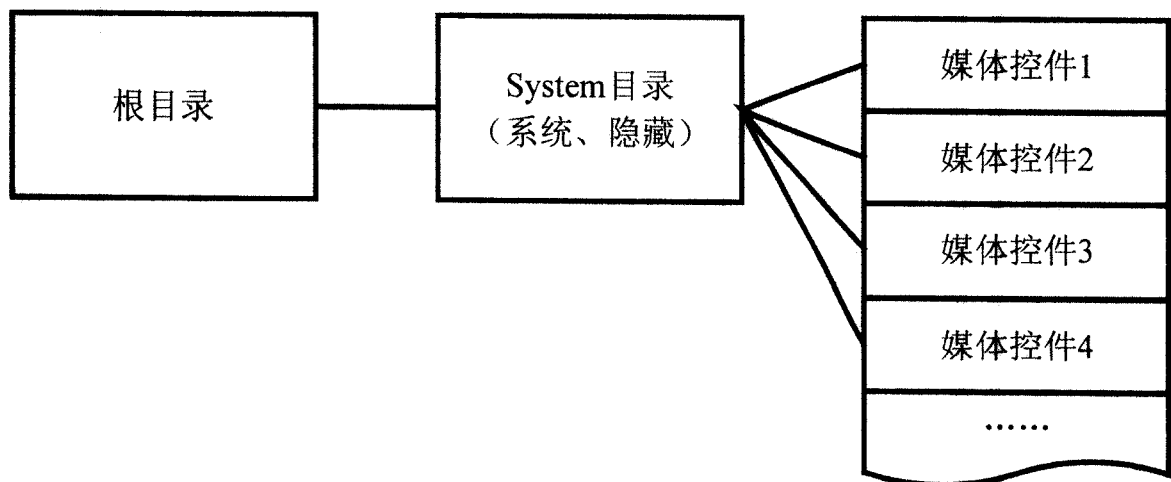


图 3

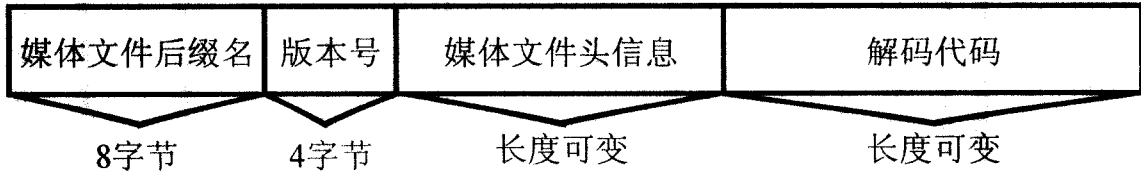


图 4

0x68	0x00	0x61	0x00	0x76	0x00	0x00	0x00
h		a		v		-	

图 5

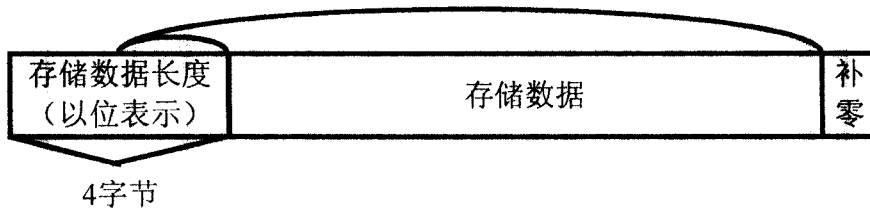


图 6

十六进制	0x55	0xAA	0xFF	0x00
二进制	0 1 0 1 0 1 0 1	1 0 1 0 1 0 1 0	X X X X X X X X	0 0 0 0 0 0 0 0
存储数据	00 11 00 11 00 11 00 11	11 00 11 00 11 00 11 00	01 01 01 01 01 01 01 01	00 00 00 00 00 00 00 00
十六进制	0x00	0x0X	0xFF	0xFF
二进制	0 0 0 0 0 0 0 0	0 0 0 0 X X X X	X X X X X X X X	1 1 1 1 1 1 1 1
存储数据	00 00 00 00 00 00 00 00	00 00 00 00 01 01 01 01	01 01 01 01 01 01 01 01	11 11 11 11 11 11 11 11

图 7