

# PA1688 开发与应用培训

- 开发部分
  - PA1688芯片的简单介绍
  - 现有的产品线和未来的开发方向
  - 软件的整体结构
  - 第三方开发的内容与方法
- 应用部分
  - 参数配置
  - 抓包工具与方法
  - 其他

# PA1688芯片的介绍

- 1999年设计，2000年投片，2001年Voip方案成型，销售至今
- 双CPU的结构，0.25微米工艺，AC97, sdram, PIO, 8051 bus, LCD direct drive。
- 50MHz, 并行处理，从处理能力上，只适合单路的终端设备。
- 8051的MCU core, 8bit 数据总线, 64k bytes的单页面程序空间。
- ADSP2181 DSP core, 但dm和pm的容量都小很多

# PA1688芯片的介绍

- 8051的数据类型
  - Data, idata, xdata, code. Data和idata是core内自带的sram，共256 byte，xdata是PA1688内的sram，共4.5k byte。其中有0.5k byte可以和sdram交换数据。Code是在外挂的flash中，存储数据的区域，只能存储const和table。
  - 数据空间的主要限制是xdata和data。对于data段，凡是中断处理程序中要用到的变量，都要定义在data段。第三方开发的客户，一般不会涉及到data段。对于xdata段，程序中所有的变量，都要定义在这里，所以我们要充分复用xdata。事实上，可用的xdata只有4k，另外0.5k专门用于和sdram交换数据(Sdram\_pDataCache)

# PA1688芯片的介绍

- 8051的程序空间
  - 单页程序空间只有64k，PA1688支持页面切换，但只能切换到某个页面的起始地址。
  - 我们使用keil7.0的编译器，可以支持多页面的编译模式，就没有了64k地址空间的限制。
  - 现在的程序，是采用上面两种方式相结合的程序空间结构。对于初始化过程中的引导程序，采用单页编译，页面切换的方式，到主程序中，采用多页混合编译的模式。

# PA1688芯片的介绍

- 8051的程序空间
  - 单页程序空间只有64k，PA1688支持页面切换，但只能切换到某个页面的起始地址。
  - 我们使用keil7.0的编译器，可以支持多页面的编译模式，就没有了64k地址空间的限制。
  - 现在的程序，是采用上面两种方式相结合的程序空间结构。对于初始化过程中的引导程序，采用单页编译，页面切换的方式，到主程序中，采用多页混合编译的模式。

# PA1688芯片的介绍

- PIO的使用: LED, 中断检测, 控制管脚, 键盘
- 8051 bus: 挂外设的接口, 8019AS, 16550 串口, 点阵型LCD (中文显示)
- LCD direct pin: 直接挂LCD, 字符型
- Codec: AC97 or UDA, we use AC97 interface now.

# 现有的产品线和未来的开发方向

- 电话终端 PA168S
- 单口网关：单S口，带电话线接口，单O口，1S1O
- Router功能？不适合在PA1688上作，资源和功能的限制。
- 中文显示：有计划，硬件快出来了
- 多语言支持：已经作了一部分，可以看version.h的注释

# 现有的产品线和未来的开发方向

- 近期的工作目标
  - 分协议的UI
  - 回声抵消，忙音检测，T.38
  - 单口网关产品的基本功能
  - 中文显示
  - IAX2协议的支持，iLBC算法的支持

# 软件的整体结构

- 文件的组织结构
- 数据的组织结构(sdram分配)
- 程序的执行顺序
- 库文件和应用层文件的分隔
- 函数，变量的命名规则
- 阅读代码的切入点
- 程序的编译，更新

# 文件的组织结构

- 所有的文件都在C:\PalmH323目录下，包括仿真程序SimIPPhone，调试更新工具PalmTool，以及运行在硬件上的源文件和库文件。
- Page0:硬件上电后，从该页开始执行。还有部分测试程序也在该页。
- G723, g723misc, g729, g729misc: DSP程序在这几页中被导入到sdr
- Settings: 设置菜单的页面，从page0可以切换到这个页面，单独的，不能在主程序中直接作设置。
- Page7, page14, page15:对于不同的flash，存储话机的参数设置的页面
- P\_apps: 应用层协议和一些应用函数
- Main: 主程序的目录，一般占用2-4个页面
- Lib: 库文件，包括所有库的库文件
- Inc: 头文件
- 其他的目录:

# *sdram*的寻址

- 8051是8位数据线，对char型的数据访问，速度最快。而sdram是重要的数据存储区，8051要经常访问sdram，因此我们要使用char型数据，来对sdram寻址，以提高速度。
- 现在是用两个char，分别是high和low来定位一个256 bytes的sdram块。这样寻址范围就是 $256*256*256=16\text{M}$  byte。我们现在使用的是2M byte的sdram。
- 8051利用Sdram\_pDataCache访问sdram，这是前面说的512字节sram的头指针。

# 数据的组织结构(sdram分配)

- 在PA1688的VoIP系统中，sdram是很重要的一个方面。DSP的程序，都是存储在sdram中的，在需要的时候，才导入到DSP的内存并执行。由于MCU的sram很少，我们也要把很多固定的消息表存储到sdram中。各种铃声，音乐，也是从sdram中得到的。
- Sdram的分配，在inc\sdram.h中，看代码具体分析。
- 话机参数的位置定义和执行过程，在inc\option.h, pa1688.h中，看代码具体分析

# 程序的执行顺序

- 前面已经说过，PalmH323是采用单页面编译，执行完再切换，和多页面混合编译的混合模式，除了main，其他页面都是单页面编译，执行后用switchpage切换
- 各种flash的执行顺序是不同的，我们现在的主流flash是mx29lv008t，我们正在推广的flash是micron的mt28f016。看代码分析这两种flash的执行顺序。
- 在主程序中，程序的执行，就是按照循环-中断模式进行的。Main\main.c中有while(1)循环，就是程序的主循环，各种handleinterrupt程序，就是中断处理程序。xxxRun函数，就是根据中断处理程序的标记，执行的具体代码。

# 库文件和应用层文件的分隔

- 由于我们没有使用操作系统，为了便于第三方的开发，我们把整个程序分成了应用层文件和库文件。对于一些底层的操作函数，比如TCP/IP，PA1688的基本操作，通信协议等内容，我们都做成了库文件，应用层直接调用，减少了第三方开发者需要了解的内容。
- 库文件包括开放代码和不开放代码的，原则是，对于复杂的协议处理，如TCP/IP，DSP；我们的核心代码，如通信协议部分，还有为客户定制的代码，只提供库文件。对于基本的PA1688操作，或者基本的应用层操作，我们可以提供库的源代码。
- 库文件是应用层对底层的调用接口，同样，我们还提供底层对应用层的回调函数接口，在main\v\_task.c中，看代码分析。

# 函数，变量的命名规则

- 函数的命名规则
  - 只在本文件内部使用的函数：以下划线作前缀
  - 会在其他文件中被调用的函数：正常的函数命名
- 变量的命名规则
  - 基于标准匈牙利命名法，i前缀表示数，p前缀表示数组和指针，b前缀表示布尔量
  - 一个函数内部定义的局部变量：正常命名
  - 只在本文件内使用的全局变量：前面加下划线
  - 会在其他文件中使用的全局变量：在下划线前，再加表示变量模块位置的名字，如Ac97\_iBufNum, Dsp\_bEncode等等

# 阅读代码的切入点

- 从page0开始，跟踪程序的执行顺序，了解如何进入安全模式和设置菜单的。
- 可以利用BootIndication()函数，跟踪程序执行，直到main。
- 主程序里面，main\main.c的while(1)循环，是整个程序的主循环，里面会调用xxxRun()这样的函数，处理各个模块的工作。xxxHanleInterrupt()函数，是硬件的中断处理程序。我们的做法，就是在HandleInterrupt中根据中断，设置对应的标记；在xxxRun中，作实际的处理工作。
- 阅读代码，不要泛泛的，看一个文件里面的所有函数，而应该按你所要了解的内容，从xxxRun和HandleInterrup函数开始，按程序的调用关系，一层层的看下去。

# 程序的编译, 更新

- 从1.39正式版开始, 不再同时支持多协议了, 而是改为一个binary文件, 对应一个协议。因此话机设置中的协议选项, 也取消了。
- 我们使用Keil7.0作为程序的编译器, 将其安装在C的根目录下。
- 将API的目录PalmH323放到C的根目录下, 去掉inc\version.h的只读属性。在cmd模式的c:\PalmH323下, 运行setenv, 设置环境变量。然后运行make type protocol language > 1.bak。
- Type是硬件类型, 对应version.h中的VER\_XXX, protocol是通信协议类型, 对应version.h中的CALL\_XXX, language是界面语音, 对应version.h中的RES\_XXX。这样就可以生成正确的版本, 并将编译的信息输出到1.bak中, 便于检查。
- 举例make pa168s h323 cn > 1.bak, make pa168p sip us > 1.bak
- 这样就直接生成了更新用的bin文件, 用PalmTool更新。

# 第三方开发的内容与方法

- 第三方开发的三个层次
  - 用户界面级
  - 应用层网络协议级
  - PIO驱动及控制级
- 用户界面级的开发方法
  - 用户界面的开发，主要集中在键盘和显示。因此这部分开发首先要理清现有代码中按键事件的传递和处理函数，显示往往是被动的基于按键触发的，也可以从按键处理的程序跟踪下去（看程序）

# 第三方开发的内容与方法

- 应用层网络协议的开发
  - 没有操作系统，我们的TCPIP协议栈自行开发，一般用户作的协议都是基于TCP或者UDP的。方法是，首先要明白的掌握你要作的协议，其次，参考我们的p\_apps下的应用层协议代码，建立自己的处理函数。  
(用程序说)
- PIO驱动及控制
  - 涉及到这部分的开发，一般都是用户新加了一些器件，需要中断检测和控制。这是最难的情况，因为涉及到中断，不方便调试。一般不建议用户自行作这方面的开发，如果需要的话，请将具体想法和我们交流，订出合理的方案。

# 第三方开发的内容与方法

- 新增变量或者文件后，如何编译得到正确的firmware
  - 新增变量：如果新增变量涉及到库文件，比如p\_util下的文件，则首先编译库，再编译程序。
  - 新增文件：一般新增文件，都在main或p\_apps目录下，修改main\make.bat和linkfile.txt，按已有格式把新增文件的编译命令加进去

# 第三方开发的内容与方法

- 程序的调试

- 编译时的错误：将编译输出的结果重定向到文件中，察看具体的错误指示信息。一般来说，有两种错误，一是C语言不过关，有语法错误，另一种是使用的xdata, data太多，造成溢出。对于前者，请自己检查；对于后者，我们一般的建议，就是谨慎使用全局变量和大的数组，尽量用局部变量，参数传递，和sdram存储数组。
- 运行时的错误：一般而言，这种错误，都是写程序时考虑不清造成的，可以用OutputVal或OutputVals输出相关信息，在PalmTool Debug窗口，看输出结果。从而作为修改程序的依据。

# 第三方开发的内容与方法

- 程序的调试

- 网络协议的编程：一般不直接在话机上编程，而是使用SimIPPhone。这是一个PC上的仿真软件，模拟了PA1688的开发环境，和PA1688有公用的代码。Demo一下

# 第三方开发的内容与方法

- 程序的调试

- 网络协议的编程：一般不直接在话机上编程，而是使用SimIPPhone。这是一个PC上的仿真软件，模拟了PA1688的开发环境，和PA1688有公用的代码。Demo一下