

PA1688 Development and Application Training



- Development
 - Introduction of PA1688
 - Current product line and road map
 - The structure of software
 - Some special method for the third party.
- Application
 - Parameters configuration.
 - The way to capture data
 - Others

Introduction of PA1688

- Designed in 1999, produced in 2000, VoIP solution is ready in 2001, keep selling until now
- Dual CPU core, AC97, sdram, PIO, 8051 bus, LCD direct drive。
- 50MHz, suitable for the 1-channel VoIP equipment。
- 8051 MCU core, 8bit data bus, 64k bytes program space for each page。
- ADSP2181 DSP core, very small dm and pm space in DSP.

Introduction of PA1688

- Data type in 8051
 - Data, idata, xdata, code. **Data** and **idata** is the SRAM in 8051 core, 256 bytes totally; **xdata** is the SRAM in PA1688, 4.5k bytes totally, including the 512 bytes which could exchange data with SDRAM; **code** is in the flash space, to save some constant and table.
 - The main limitation in data memory is xdata and data space. For data space, all the varies used in the interrupt handle must be defined in data space. For the 3rd party, it will not be used in usual time. For xdata space, all the varies of program are defined here, so we must re-use xdata space. In fact, there are only 4k bytes for xdata, the other 512 bytes are Sdram_pDataCache and have special useness.

Introduction of PA1688

- Code Space of 8051
 - Each single page is 64k bytes. PA1688 supports page-switching, but can only switch to the start address of each page.
 - We use Keil 7.0 as the compiler. It support multi-page mode, no 64k bytes limitation.
 - Our current program structure combine the 2 ways above. For the programs in boot mode, we use single page compiling, page switching mode; in main program, we use multi-page mode

Introduction of PA1688

- PIO: LED, Interrupt detection, Pin control, Keypad
- 8051 bus: 8019AS, 16550 serial port, pixel LCD (for special display)
- LCD direct pin: connect LCD directly
- Codec: AC97 or UDA, we use AC97 interface now.

Product Line and Future development

- IP Phone: PA168S
- One port gateway: single FXS, single FXO, 1 FXS with PSTN line port, 1S1O
- NAT router? Not suitable for PA1688, has limitation on chip resource
- Chinese Display: In plan
- Multi-Language: Be in doing

Product Line and Future development



- Recent work
 - Different UI for different protocol
 - Echo cancellation, busy tone detection, T.38
 - Enhance the feature of one port gateway
 - Chinese display
 - IAX2, iLBC

The structure of Software

- File organization
- Data organization (sdram allocation)
- Code executive sequence
- API interface
- The rule to name functions and varies
- The way to read our code
- Compiling and upgrading

File organization

- All files are in C:\PalmH323, including SimIPPhone(Simulator), PalmTool(Debug and upgrade tool), all source and library running on hardware.
- Page0:the program execute from this page, there are some other testing code here.
- G723, g723misc, g729, g729misc:DSP code load to sdram here.
- Settings: the page for setting menu.
- Page7, page14, page15:For different flash, save the phone setting parameters.
- P_apps: function in application level
- Main: main code's folder, need 2-4 pages to run
- Lib: all libraries in this folder
- Inc: all header file in this folder

s dram index

- 8051 is 8 bits data bus. So it is fastest to use the char data. Sdram is the important space for data storage, 8051 always read/write sdram, so we must use char data to index the sdram to make it faster.
- We use two char data, high and low, to address a sdram block with 256 bytes. By this way, we can address $256*256*256=16\text{M}$ bytes sdram. We are using 2M bytes sdram now.
- 8051 use Sdram_pDataCache to read/write sdram, it is the pointer of those 512 bytes said before.

Data organization (sdram allocation)

- Sdram is a very important part in PA1688 VoIP system. All the DSP code is saved in sdram. We will load them into DSP's PM to execute when we need them. And we also save many message table , ring, music in sdram
- Allocation of Sdram, in `inc\sdram.h`
- Phone setting parameters definition , in `inc\option.h`, `pa1688.h`

Code executive sequence

- We have talked that PalmH323 use mixed mode to compile the program. For the boot pages, we use single page compiling and page switching; for the main page, we use multi-page0 compiling
- The code flow is different in different program flash type. The popular program flash we used are MXIC's mx29lv008t and Micron's mt28f016. Now let us analyze it.
- In main page, the code is executed in Loop-Interrupt mode. The while(1) loop in Main\main.c is the main loop, all kinds of handleinterrupt() functions, all Interrupt functions. And xxxRun() functions, are the real code to execute depends on the interrupt flags.

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