

# 不同协议栈间的手机软件移植

景建岗<sup>1</sup>, 刘高鹏<sup>2</sup>, 郑启龙<sup>3</sup>

(1. 中国科学技术大学软件学院, 合肥 230052; 2. 英飞凌科技(中国)有限公司, 上海 201203;  
3. 中国科学技术大学国家高性能计算中心, 合肥 230052)

**摘要:** 研究手机软件在不同协议栈之间的可移植性, 根据 AT 命令特性, 提出一个有效的协议栈适配层实现框架, 在几乎不改变手机软件结构的情况下, 通过 AT 命令实现上层通信层与芯片、GSM 模组或其他通信协议之间的通信, 使手机软件开发独立于特定的协议栈。基于该框架, 该文对短消息等相关功能进行了验证。

**关键词:** 手机软件移植; AT 命令; GSM 协议栈

## Mobile Software Porting Among Different Protocol Stacks

JING Jian-gang<sup>1</sup>, LIU Gao-peng<sup>2</sup>, ZHENG Qi-long<sup>3</sup>

(1. Department of Software, University of Science and Technology of China, Hefei 230052; 2. Infineon Technology(China) Co., Ltd., Shanghai 201203; 3. National High Performance Computing Center, University of Science and Technology of China, Hefei 230052)

**【Abstract】** This paper mainly researches the portability of mobile software among different protocol stacks, and it implements a valid adapter framework which is based on attention command protocol. At the condition of hardly changing the structure of mobile software, it implements the communication between the upper communication layer and other chipsets, other GSM modems or other communication protocol by AT commands, so the current mobile software development can be independent of the specific protocol stack. Meanwhile, the related functions of short message are implemented and verified on the framework.

**【Key words】** mobile software porting; AT command; GSM protocol stack

### 1 概述

蜂窝无线通信协议栈遵守 GSM 规范, 各个协议栈提供商根据 GSM 规范, 结合芯片具体功能设计出各自的协议。由于各种芯片包含不同子模块, 因此具有不同集成程度, 对上层中间件以及 MMI 提供不同的功能, 导致各大厂商协议栈之间存在极大差异。协议栈内部由于性能和数学逻辑的需要, 根据逻辑图自动生成其特有的数字式代码, 这些都导致了不同协议栈之间无法兼容。近来, 随着通信芯片越来越完善和成熟, 高性能多媒体芯片的极速发展, 多个芯片搭配起来完成通信和多媒体功能的应用越来越广泛。除了 GSM 芯片外, 具有更廉价通信能力的 WIFI 芯片也希望借助已有的通信基础来完成或者协助无线通信, 因此, 人们需要一种具有多媒体和 VOIP 功能的手机软件在不同架构的协议栈间进行平滑或者无缝移植。

在早期通信规范协议中, 为了方便测试以及人机交互的需要, 手机软件提供商协作推出了 AT 协议, 将协议栈中以紧密压缩数据结构表示的信号中间件和底层的交互部分, 以开发者能够理解的字符串的形式进行表达。为了实现双芯片通信的需要, 协议栈信号之间特有的信号交互模式因为难于移植而不被采用, 因此, 各大手机解决方案提供商经过协商提出一套标准的 AT 命令集来增强手机软件的灵活性, AT 本身不是一个成熟的完善系统, 其本身不是为了实现完整的通信系统而设计的, 但由于通信模块的功能基本齐备, 方案提供厂商都希望在此基础上扩展自己的指令, 以实现与其他芯片的兼容。

目前, 在国内外手机软件应用与开发领域中能将协议栈

部分独立移植出来的产品很少, 国内外只有少数几家拥有 GSM/GPRS 协议栈的公司才有可能做到。一般情况下, 客户需将手机软件交由协议栈提供商来对相应协议栈进行移植, 或者客户自己在开发手机软件的时候就必须针对某个特定协议栈来进行。这就凸显了手机软件在不同协议栈间进行无缝或平滑移植的重要性与迫切性。

### 2 GSM 手机软件架构

GSM 手机软件系统架构采用分层设计, 包括 MMI 层、中间件层、协议栈层和驱动层, 这样就使得各个模块相互独立、松散耦合, 便于手机软件在不同平台和应用上定制<sup>[1]</sup>。

#### 2.1 中间件层

中间件层主要对底层协议栈和驱动层进行抽象、隔离和封装, 对上层 MMI 隐藏协议栈的复杂性, 不影响中间件与协议栈和驱动层的接口, 对上层 MMI 提供 C 原型函数调用和事件回调函数, 提供访问网络通信服务、语音通话、增值服务、SIM 卡、点对点/广播短消息和 SIMToolKit 服务等。它隔离 MMI 的具体实现和特定的手机平台, 对用户后期进行上层 MMI 代码的修改和定制透明, 便于用户开发, 提高软件开发的灵活性和高度复用性。

#### 2.2 MMI 层

MMI 层位于中间件之上, 主要实现 MMI 相关的应用:

(1) 电话本应用程序;

**作者简介:** 景建岗(1981 - ), 男, 硕士研究生, 主研方向: 嵌入式系统软件设计; 刘高鹏, 工程师; 郑启龙, 副教授

**收稿日期:** 2007-11-25 **E-mail:** jgjin@mail.ustc.edu.cn

- (2)点对点短消息；
- (3)语音通话应用；
- (4)启动和关闭；
- (5)MMI 预告设置；
- (6)系统配置等；
- (7)GUI 组件(菜单、消息框、编辑框等)；
- (8)MMI 资源(字体、图标、位图、铃音等)。

每个应用都创建一个实时任务，它调用中间件访问协议栈相关服务，通过中间件的事件回调函数触发相应的消息，从实时操作系统的消息队列中读取消息。手机系统中每个应用模块和进程都有自己的消息循环队列与用户交互，集中解析所有系统级异步事件(例如 MT 电话/点对点短消息)，管理 MMI 聚集的切换，通过调用异步函数或异步信号来实现不同应用间通信，其中，函数调用包括同步函数调用和异步函数调用。

### 2.3 协议栈和驱动层

协议栈层和驱动层一般由硬件平台提供商提供，或者对于专门开发手机软件的公司，协议栈是由客户购买的。为了使手机软件可以灵活地移植到不同的协议栈，本文在手机软件(中间件层之下)与协议栈之间添加了一个适配层，它可以解析手机软件的信号与协议栈的标准AT命令<sup>[2]</sup>，与手机软件之间通过SDL信号通信，与协议栈ATC模块通过AT命令来适配通信；并且针对一些客户应用中没有标准AT命令与之对应的信号，则通过扩展标准的AT命令来实现，使得适配层对用户完全透明，从而无缝或平滑地移植用户的手机软件。GSM 手机的系统架构如图 1 所示。



图 1 GSM 手机软件框架

## 3 协议栈适配器

### 3.1 协议栈适配器在手机软件中的位置

如图 1 所示，为使手机软件可以独立于特定协议栈，只需在手机软件中间件与下层协议栈之间配置静态或者动态协议栈适配器即可，它可以通过GSM07.07<sup>[2]</sup>规范定义的标准来通信。适配器在手机软件中的位置如图 2 所示，适配器是可选的，如果下层协议栈是手机软件默认的协议栈，则通过编

译选项中的宏开关取消对适配器编译，否则通过宏开关激活适配器，例如，在目标板上编译适配器定义为 #if defined AT\_ADAPTER\_PRESENT，如果需要适配器来解析，则在编译器的编译选项中加入宏开关 AT\_ADAPTER\_PRESENT 即可，否则去掉。Windows 环境模拟器上编译适配器定义 #if defined AT\_ADAPTER\_PRESENT && defined WIN32 来激活适配器。

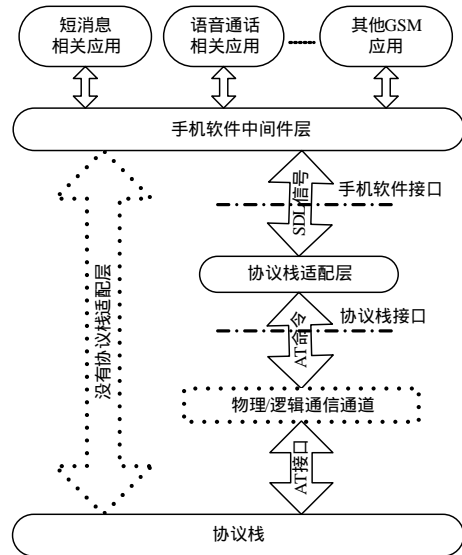


图 2 适配器在手机软件中的位置

### 3.2 适配器与上下层接口

#### 3.2.1 手机软件接口

适配器与手机软件的接口主要解析手机软件中间件发给协议栈的 SDL 通信信号，并将解析后的字符流发往适配器相应的 GSM 或者非 GSM 处理模块做编码和相应的处理；并用适配器编码的 SDL 信号向手机软件中间件提示协议栈的相关信息，手机软件中间件根据此信息调用相应事件回调函数通过消息通知 MMI 应用层对用户做相应的提示。

#### 3.2.2 协议栈接口

适配器与协议栈间的接口主要支持通过物理或者逻辑通信通道在协议栈和适配器之间的双向通信，适配器通过协议栈发送的 AT 命令启动相应的 GSM 或者非 GSM 模块处理程序做相应的处理，并且适配器支持 AT 命令的多通道协议<sup>[3-4]</sup>，其中通道是预先分配的，从物理连接来看，此通道一般是串口或者 SIP 口。从逻辑连接来看，此连接一般是一个内部任务消息通信机制。在缺省情况下，此连接为 UART0。适配器的功能和协议栈的接口可以通过在系统配置文件中添加不同通信介质的适配器进行扩展。

### 3.3 适配器

首先适配器提供的服务是线程安全的，它可以被不同的上下文实体调用，适配器对定时器的处理要求：适配器提供的服务在一定的时间片内会有明确的响应，并且当适配器运行出错时，它可以通过内部的错误处理机制通知手机软件中间件相应的错误信息，手机软件中间件再将相应的提示以消息方式发给 MMI 做相应的显示和预警处理。在 MMI 适配器中，对所有的 GSM 相关服务也是松耦合的模块化设计与实现，并且每个适配器内部的模块在系统编译的时候是可以被激活或者屏蔽的。并且下方物理或逻辑通信通道是可配置的，缺省是 UART0 来配置。

单一的 AT 命令和 SDL 信号无法一一对应，每一个独立的事件都是由某个序列 SDL 信号或某个序列的 AT 命令来完成。所以，适配器也需要保留一个事件的阶段状态来控制当前需要发出的 AT 命令或 SDL 信号，该状态在一个完整事件完成后即可回复原位。由于 AT 命令需要异步收发，因此规定每次协议栈通过发送最后一个命令，来控制发送 AT 命令的节奏。并且当前协议栈发给适配器的 AT 命令跟前一个适配器发给协议栈的命令相关。根据此特点，通过一个前置分析动态替换处理函数，同时将各个命令和对应的回复处理相关联，极大地简化状态机的复杂程度。

适配器启动后，首先从适配器工作队列中取得当前事件，对接收到的事件做预处理，如果此事件是与适配器 GSM 模块无关的，立刻将此事件处理且置处理标记为 TRUE，相应定时器对应的等待事件也将在此删除。如果此事件是 AT 命令，则适配器调用通用解码程序对此命令串进行解码，通用解码程序得到 AT 命令的串的命令 ID，根据此 ID 调用相关专用命令解码器提取命令的所有参数，为所有参数分配内存空间，返回此命令串的长度，再次调用通用解码程序解码剩余的串，直到所有串处理完，因为 AT 字符串可能由多条 AT 命令组成，其中一条是触发事件，其他的是这个事件相应的参数，他们的顺序不确定，所以，需要先解码所有参数的值，然后对触发事件进行信号转换。

适配器的核心是提供的 GSM 模块处理器。适配器 GSM 模块处理流程如图 3 所示。

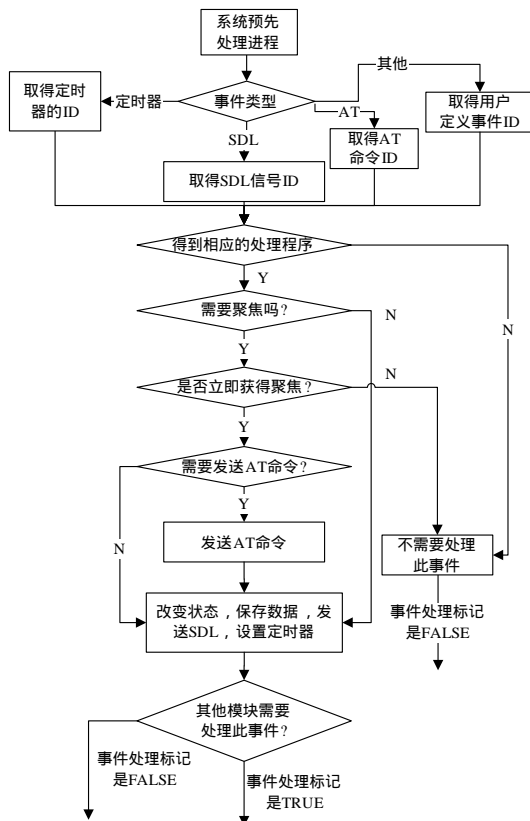


图 3 适配器 GSM 模块处理流程

如果系统预先处理进程不能设置此事件，则处理标记为 TRUE，专用模块接着处理此事件，无论适配器 GSM 模块的当前状态是激活且聚焦，激活非聚焦或空闲状态，处理流程都一样。如果处理模块没有得到事件的 ID，此事件将被忽略

并且处理标记不做任何修改。如果得到此事件 ID，则调用相关的回调函数去处理当前事件，此回调函数通过解码当前 SDL 信号或者从通用解码或专用解码创建的双向链表中搜索特定的结点，如果当前事件的处理模块不是聚焦状态但需要聚焦，并且此时系统没有激活且聚焦模块存在，当前模块将立即设置其状态为激活且聚焦状态，如果此时存在激活且聚焦状态模块，适配器将判断是否立即抢占聚焦，延迟抢占聚焦或不聚焦，当前适配器 GSM 模块需要改变其状态，保存数据(包括状态相关信息)，发送 SDL 信号，发送 AT 命令和设置定时器，并置此事件的处理标记设为 TRUE，否则设为 FALSE。

最后是适配器的收尾处理，当所有专用模块处理结束后，如果事件是 AT 或者定时器事件，它们将从工作队列中删除，如果事件是 SDL 或者用户定义的事件，并且事件处理标记是 FALSE，系统将设置等待标记暂不删除；如果处理标记是 TRUE，则下一个事件可能不是当前事件的下一个事件，因为适配器从工作队列头转去预先处理等待事件，这样可以满足等待事件的处理触发条件，同时等待事件能得到最快响应。如果在所有专用模块处理结束后，当前事件的处理标记为 FALSE，工作队列中的等待事件将不会处理。

#### 4 适配器短消息模块处理器实现

手机软件接口对接收到的 SDL 信号进行通用解码器解码，再调用适配器 GSM 模块判断当前的事件是否为短消息模块处理器，如图 4 所示，短消息模块处理器专用解码器对当前的事件解码，根据解码的结果判断需要协议栈提供何种服务，根据要提供的服务，短消息模块处理器向协议栈发送相应的 AT 命令，协议栈接口对接收到的 AT 命令根据所提供的服务做相应处理后，将 AT 命令串发给协议栈的 ATC 模块。如果是协议栈向手机软件发送的 AT 命令，则协议栈接口做简单处理(如拼接 AT 命令串)后，调用适配器短消息模块处理器，适配器调用通用解码器解码，再调用适配器 GSM 模块判断当前事件是否属于短消息模块处理器，然后此模块处理器对此 AT 命令进行专用解码，根据解码结果进行相应的 SDL 编码，向手机软件中间件发送相应 SDL 信号。

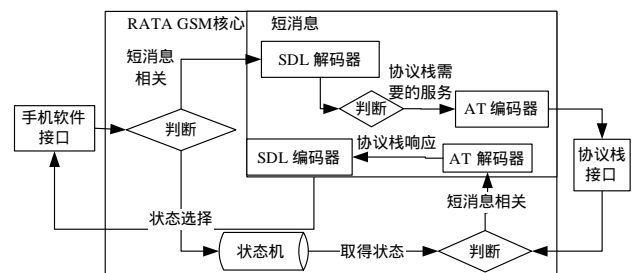


图 4 适配器短消息模块处理器

当有新短消息到达时，协议栈通过发送命令 AT+CNMI 通知适配器，适配器短消息处理模块对命令参数进行解析，然后发送信号 DR\_SM\_NEW\_MSG\_IND 通知手机软件中间件有新短消息到达，中间件立即通知 MMI 应用层做相应的 UI 处理。当手机软件中间件向适配器发送信号 DR\_SM\_READ\_REQ 请求读此短消息时，适配器短消息处理模块将解析此信号并向协议栈发送命令 AT+CMGR<sup>[3]</sup>进行读操作，当读操作成功，协议栈将返回 +CMGR:<paras>，所带的参数中包括些短消息所在存储介质上的标号和当前短消息的 (下转第 79 页)