Facing the Complexity Challenge of Mobile Phones

Mobile phone makers have to develop a wide range of handsets for operators around the world, creating a complex hardware and software problem. Mike Chambers of Agere Systems discusses the development of a solution with a major handset maker

Mobile phone manufacturers are faced with a complexity problem. It's not just the complexity of the phones themselves that's the problem, but the complexity of the product ranges that are needed for all the different operators around the world.

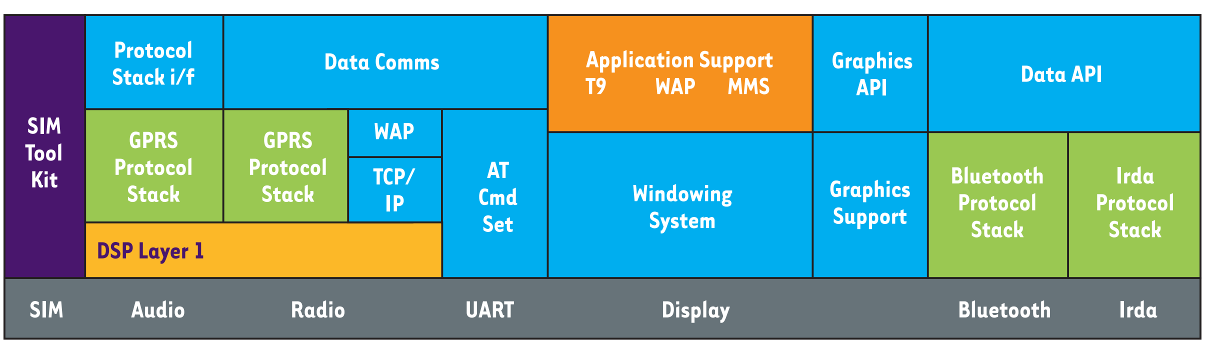
Operators demand a range of phones – from high-end smart phones through the mid-range camera phones to entry-level handsets – that also have to change regularly to keep their customers happy. In addition, there are different technologies needed, from the basic GSM technology through the enhanced EDGE technology and GPRS for data services, all the way to the new 3G services.

This diversification of product and geographical markets is vitally important to handset manufacturers. Agere supplies both GSM-based silicon and software to a number of top mobile phone developers, and reducing the overall handset development time and minimising the risks of the development is vitally necessary.

In developing the range of products, the hardware also varies significantly for phone designers. Smart phones are adding separate application processors to handle graphics and video, but these are not needed on the mid-range camera phones. Developing completely separate software environments for the different classes of phones is expensive in software development and in managing all the different software versions.

Currently it is very difficult to move extensions made in a single-processor design to the dual-processor version, so the changes have to be made all over again for each family of phones, adding to the development time, complexity, cost and risk.

The manufacturer then has to provide a wide range of handsets in different languages and with many different operator brandings on the man-machine interface (MMI). This is no small task, and often the menus and software features have to be optimised in different ways for different operators, and support multiple languages within a single, global operator.

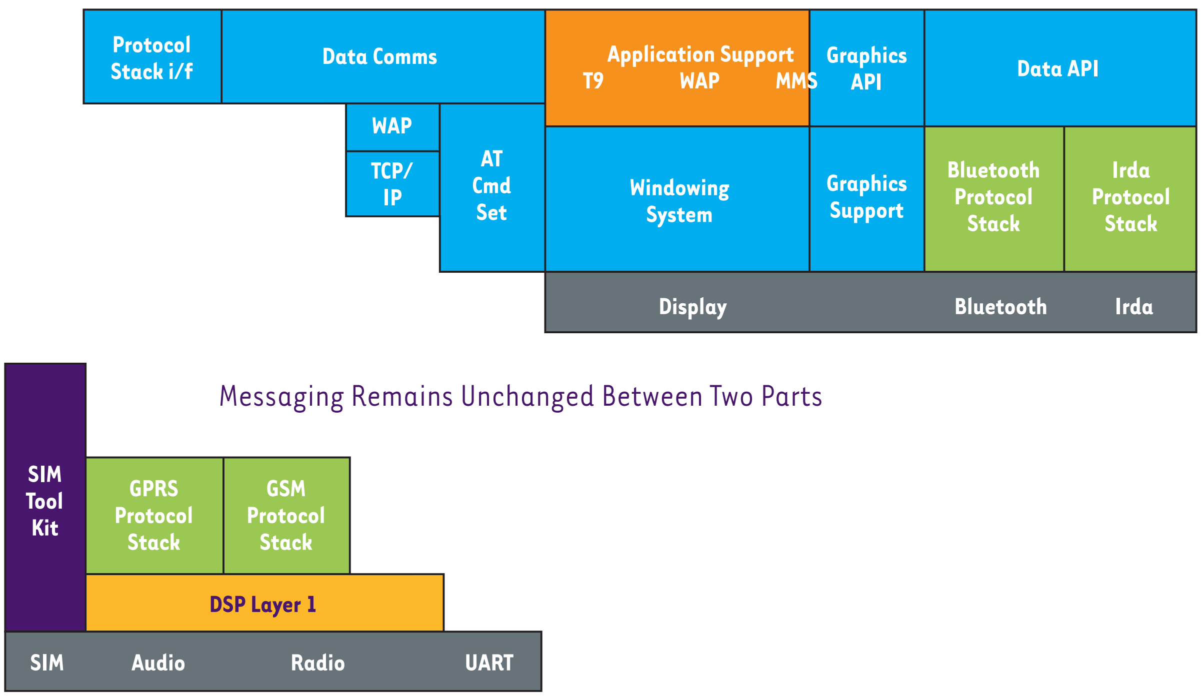


Current GPRS Engine & Agere Reference MMI running on single-core solutions

Increasingly, the handset manufacturers are looking to focus on what they do best – serving their operator customers through this customisation of appearance and function. New operating systems (OS) such as the SymbianOS® are making this easier than the days of proprietary systems that needed to be changed with each new generation of the handset, but integrating all these variants with the hardware can be a major headache.

These changes, while cosmetic, are vitally important to the look and feel of the phone and its ultimate success. But all these changes have to be made without affecting the underlying hardware. After all, this is only the MMI. But at the moment, changes to the MMI, especially the menu structures and features such as "shoot-and-send" to send photos with a single key press, require fundamental changes to the underlying code.

A tier one handset manufacturer approached Agere Systems to help them with this problem. They had already used the Agere Systems protocol stack in many successful GSM/GPRS handsets based on a single micro-processor, but now they wanted to use the same protocol stack in a dual processor handset running SymbianOS and Nokia Series 60 on the applications processor. They also wanted a scalable solution in software across single- and dual-processor designs that would also support 3G hardware when it was ready.



Software Split Between GPRS Engine & Apps Processor

The underlying problem is the AT command set that has served the handset industry so well for twenty years. These commands are well-established and simple, but the increasing complexity of the software meant that extending the AT command set and its interface is a problem.

To achieve the dual-processor solution, the GSM 07.10 multiplexing protocols and new AT commands would have to be implemented. Adding these functions into the software and then re-testing and re-qualifying all the software in the phone would have been a huge task, full of risks and potential delays.

The AT approach is also not easily scalable. Each time new functions are added that require interaction between Agere's communications engine and the applications, a new set of AT commands is required. These must be generated on the applications side and interpreted on the communications engine side. The whole system software must then be re-tested, so making changes and adding features can be time-consuming and expensive. This is especially true when dealing with the applications processor, given the task of re-testing and re-qualify the radio and data handling part of the phone.

All these problems were dramatically slowing down the development of new products with new features.

There are other approaches that aim to replace the AT commands with new structures, but these were found to be too complex as they had their own data standards and tools. This would create a problem if the data formats changed and the code had to be re-compiled and re-qualified. These approaches also do not provide a separation between the real time phone function in the communications engine and the applications. While the new extensions make changes easier and don't require re-compiling the interface, both the phone and the application code would still have to be re-compiled and re-tested.

Because the data formats were also relatively immature at the time the project was specified, there was also a worry with the ASN.1 compilers. Any potential problem in the final binary code can originate from either the source code itself or from a bug in the ASN.1 compiler. Without the detailed knowledge of the compiler technology, it would be difficult and time consuming to debug.

So Agere split the task. Instead of using and extending the existing AT commands, an Advanced Messaging Interface (AMI) was developed. This is a simple message-passing protocol that handles the interface between the real time communications processor section of the chip and the applications processor.

The real-time side of the design, the communications engine includes an ARM®7 controller and Agere's DSP16000 digital signal processor core, coupled with Agere's GSM/GPRS protocol stack remains constant, so any changes and additional features can be added to the applications code without changing the code for the phone hardware.

The two sides are physically connected via UARTs (Universal Asynchronous Receive Transmitter) that link to serial lines to transfer the data, although using shared memory is also possible.

This allows multiple chips to be used, and even supports the use of multiple cores in a single chip, all with the same software environment and without making any changes to the underlying software blocks.

The first interfaces for the AMI were hand-coded for the handset manufacturer, providing routines for packing and unpacking the messages. Each message has a simple structure, with a 4byte header that includes the ID and the length of the message so that space can be reserved in the interface FIFO buffer. Later, these interface routines will be generated automatically by a small, dedicated compiler, which means that changes to the messaging format could be easily added, if required.

All the interfaces are verified with Agere's established regression tests to ensure that the performance of the AMI is exactly the same as the existing software, minimising the risk of bugs from hand-coding or compilers.

Identifier	Length	Parameters	Checksum

Field Name	Length in Bytes	Description
Identifier	Fixed 2	A unique identifier for the signal taken from extprim.h
Length	Fixed 2	The length of the parameters field. This can be 0 if there are no parameters associated with the signal
Parameters	Variable	The parameters associated with the signal
Checksum	Fixed 2	A 16bit checksum to help detect corrupted signals

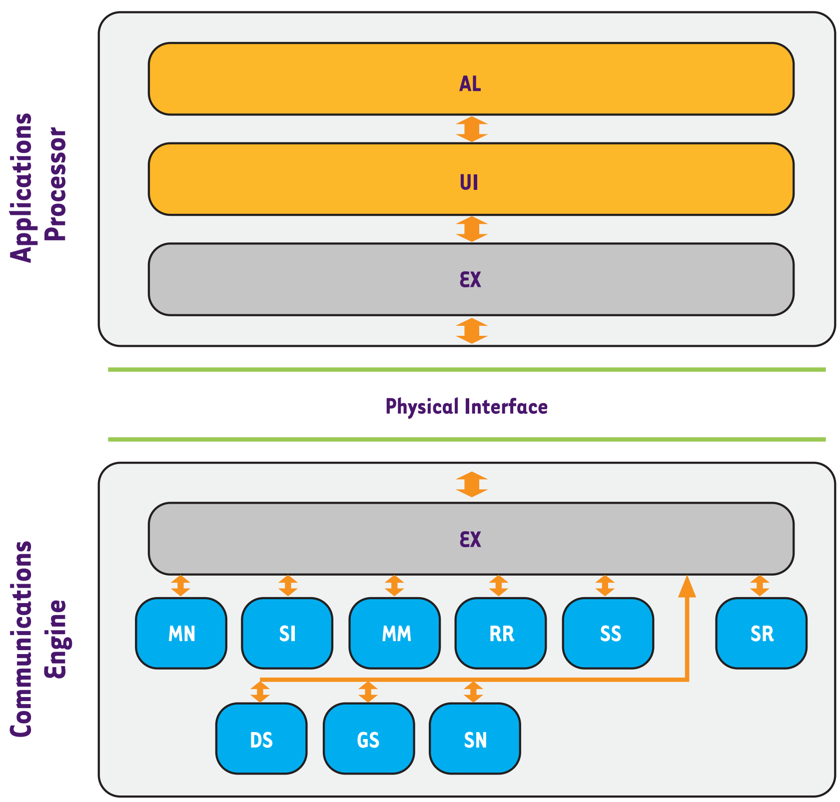
AMI Data Format

For the GPRS implementation there was no need to include priority data in the header; this information can be added for 3G designs, however, to ensure that the commands for video have priority across the interface and are not slowed down by other less time-critical applications. This is simple to implement by changing the messaging interfaces on each side. Again, the protocol stack code stays the same and unperturbed on the real-time communications engine.

AMI also has the advantage of controlling the protocol and both sides of the interface. New extensions can be easily added and the interface remains constant. With external protocols or interfaces, the specifications can change and tools can introduce bugs that are hard to track down. An open standard is of no practical use as the interface comes from the silicon supplier anyway, and this approach is no different from sourcing the silicon

But the AMI is not the end of the project by any means, as it has to be integrated with the operating system and with parts of the application code.

In this case, Symbian's operating system was used with Nokia's Series 60 application layer. To interface to this operating system, Agere created the Advanced Messaging Server (AMS), which includes software from the Agere reference MMI . The AMS Application Programming Interface (API) was defined to allow the applications to communicate with the communications engine via the Telephone System (TSY) and other SymbianOS plug-in modules. These include all standard GSM/GPRS functions such as starting a voice call, GPRS data transfer and supplementary services.

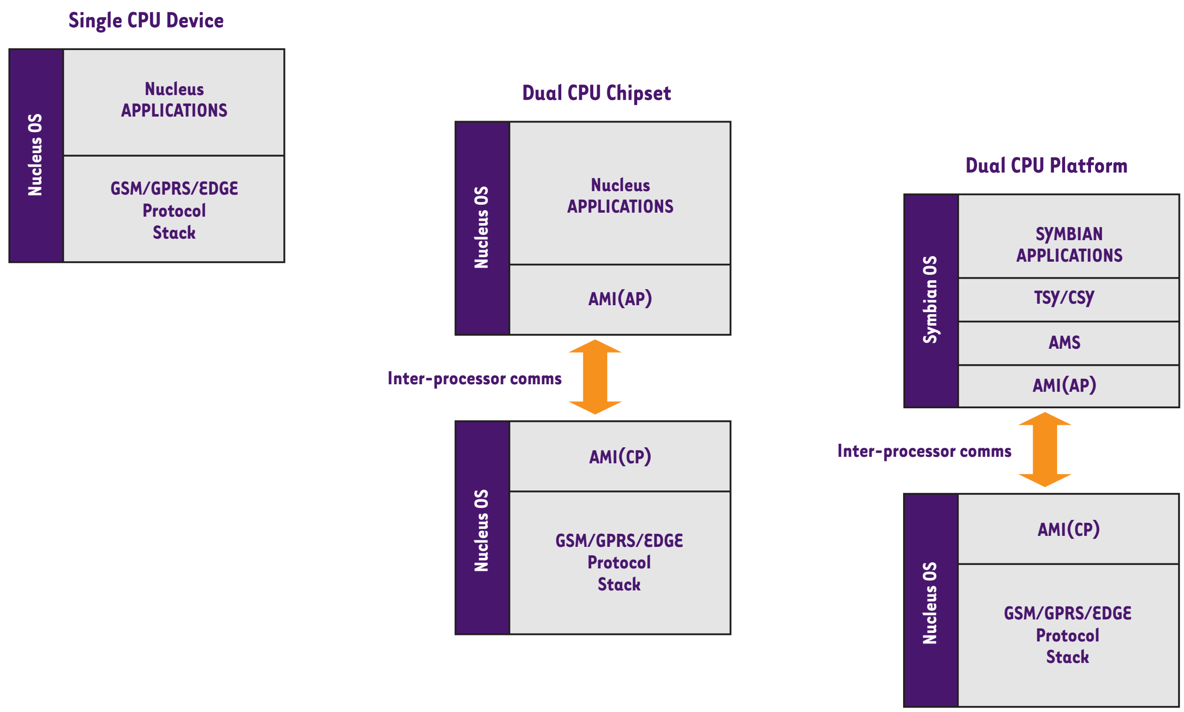


Module Separation (Agere MMI)

For example, to start a voice call, the AMS handles the negotiation with the protocol stack and the network, and provides a progress indicator and any error messages back to the TSY.

The main advantage of this approach is that new functions can be added to the applications processor without changing the code in the communications engine. This greatly speeds up the development of new features and dramatically simplifies the development of variants for different technologies, operators and countries.

For example, once the extensions for release 99 of the GPRS standard had been implemented in the communications engine and tested in a single processor environment, it was a simple to make these extensions available to an applications processor. The AMI packers and unpackers and the AMS API parts concerned with the changed data structures were extended, thus making the release 99 functions accessible to the applications. Using the old AT command approach would mean not just changing the data structures but also having them converted to AT command and AT response strings. This could lead to errors caused by misinterpretations of the data due to the change to string format.



Single to Dual CPU Environment

However, these SymbianOS plug-in modules have to be developed to use the AMS API. The TSY and network interface (NIF) modules come as templates from Symbian, and the handset manufacturer creates the modules to link them to the AMS API and the user interface.

Once these modules are built, they can be used across the whole range of phones, depending on the required features, again simplifying the complexity of managing the software. These modules can also be used for next generation Symbian handsets as well.

While the AMS is specific to the SymbianOS, support for other OS software may be requested in the future.

That led to Agere's Optiverse application framework, where the AMS can be replaced with one of the other operating systems such as Windows Mobile™, PalmOS®, or Linux. Having only the AMS server change for the operating system reduces the time, cost and risks of adding a new operating system to the chip platform. The code for the communications engine and the system remains unchanged, while only the extra OS-specific modules need to be added, simplifying the development of new applications across the range of handsets.

The development of the Advanced Messaging Interface and its supporting interfaces and system software have created a consistent platform for handset makers. The ability to use the same protocol stack, whether in a single or dual micro-processor configuration, has dramatically reduced the time taken to develop new products and customise them for different operators and different regions. This is especially important tothe increasing number of handset designers who supply the global operators with branded products.

Being able to change the MMI without having to change, re-test and re-qualify the underlying code for the real-time communication engine saves months of development time. That the modules developed for one model are usable without change on other models saves even more development and test time and helps build a library of software that can be used to mix and match different features in different phones at different price points for different consumer and geographical markets. This helps extend the product range and helps the handset maker serve its customer – the operator – more effectively.

For additional information, contact your Agere Systems Account Manager or the following:

INTERNET: **http://www.agere.com**
E-MAIL: **docmaster@agere.com**
N. AMERICA: Agere Systems Inc., Lehigh Valley Central Campus, Room 10A-301C, 1110 American Parkway NE, Allentown, PA 18109-9138
1-800-372-2447, FAX 610-712-4106 (In CANADA: **1-800-553-2448**, FAX 610-712-4106)
ASIA: Agere Systems Korea Ltd., 14 FL, Good Morning Tower, #23-2, Yoido-dong, Youngdeungpo-ku, Seoul 150-712, Korea
Tel. **(82) 2-767-1850**, FAX **(82) 2-767-1888**
CHINA: **(86) 21-54614688** (Shanghai), **(86) 755-25881122** (Shenzhen)
JAPAN: **(81) 3-5421-1600** (Tokyo), KOREA: **(82) 2-767-1850** (Seoul), SINGAPORE: **(65) 6778-8833**, TAIWAN: **(886) 2-2725-5858** (Taipei)
EUROPE: **Tel. (44) 1344 296 400**

January 2005
OT05-188MTD

射 频 和 天 线 设 计 培 训 课 程 推 荐

易迪拓培训(www.edatop.com)由数名来自于研发第一线的资深工程师发起成立，致力并专注于微波、射频、天线设计研发人才的培养；我们于 2006 年整合合并微波 EDA 网(www.mweda.com)，现已发展成为国内最大的微波射频和天线设计人才培养基地，成功推出多套微波射频以及天线设计经典培训课程和 ADS、HFSS 等专业软件使用培训课程，广受客户好评；并先后与人民邮电出版社、电子工业出版社合作出版了多本专业图书，帮助数万名工程师提升了专业技术能力。客户遍布中兴通讯、研通高频、埃威航电、国人通信等多家国内知名公司，以及台湾工业技术研究院、永业科技、全一电子等多家台湾地区企业。

易迪拓培训课程列表：http://www.edatop.com/peixun/rfe/129.html

射频工程师养成培训课程套装

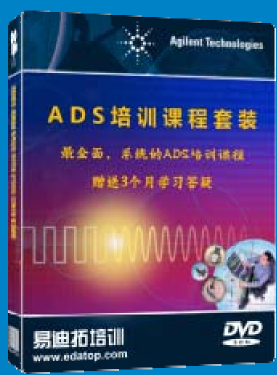
该套装精选了射频专业基础培训课程、射频仿真设计培训课程和射频电路测量培训课程三个类别共 30 门视频培训课程和 3 本图书教材；旨在引领学员全面学习一个射频工程师需要熟悉、理解和掌握的专业知识和研发设计能力。通过套装的学习，能够让学员完全达到和胜任一个合格的射频工程师的要求…

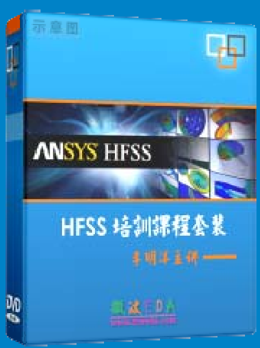
课程网址：http://www.edatop.com/peixun/rfe/110.html

ADS 学习培训课程套装

该套装是迄今国内最全面、最权威的 ADS 培训教程，共包含 10 门 ADS 学习培训课程。课程是由具有多年 ADS 使用经验的微波射频与通信系统设计领域资深专家讲解，并多结合设计实例，由浅入深、详细而又全面地讲解了 ADS 在微波射频电路设计、通信系统设计和电磁仿真设计方面的内容。能让您在最短的时间内学会使用 ADS，迅速提升个人技术能力，把 ADS 真正应用到实际研发工作中去，成为 ADS 设计专家...

课程网址： http://www.edatop.com/peixun/ads/13.html



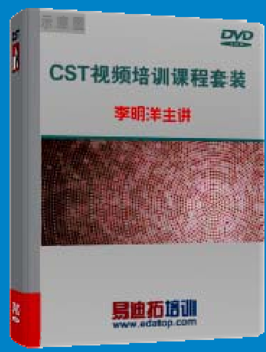


HFSS 学习培训课程套装

该套课程套装包含了本站全部 HFSS 培训课程，是迄今国内最全面、最专业的 HFSS 培训教程套装，可以帮助您从零开始，全面深入学习 HFSS 的各项功能和在多个方面的工程应用。购买套装，更可超值赠送 3 个月免费学习答疑，随时解答您学习过程中遇到的棘手问题，让您的 HFSS 学习更加轻松顺畅…

课程网址：http://www.edatop.com/peixun/hfss/11.html

CST 学习培训课程套装



该培训套装由易迪拓培训联合微波 EDA 网共同推出，是最全面、系统、专业的 CST 微波工作室培训课程套装，所有课程都由经验丰富的专家授课，视频教学，可以帮助您从零开始，全面系统地学习 CST 微波工作的各项功能及其在微波射频、天线设计等领域的设计应用。且购买该套装，还可超值赠送 3 个月免费学习答疑…

课程网址：http://www.edatop.com/peixun/cst/24.html

HFSS 天线设计培训课程套装

套装包含 6 门视频课程和 1 本图书，课程从基础讲起，内容由浅入深，理论介绍和实际操作讲解相结合，全面系统的讲解了 HFSS 天线设计的全过程。是国内最全面、最专业的 HFSS 天线设计课程，可以帮助您快速学习掌握如何使用 HFSS 设计天线，让天线设计不再难…

课程网址：http://www.edatop.com/peixun/hfss/122.html

13.56MHz NFC/RFID 线圈天线设计培训课程套装

套装包含 4 门视频培训课程，培训将 13.56MHz 线圈天线设计原理和仿真设计实践相结合，全面系统地讲解了 13.56MHz 线圈天线的工作原理、设计方法、设计考量以及使用 HFSS 和 CST 仿真分析线圈天线的具体操作，同时还介绍了 13.56MHz 线圈天线匹配电路的设计和调试。通过该套课程的学习，可以帮助您快速学习掌握 13.56MHz 线圈天线及其匹配电路的原理、设计和调试…

详情浏览：http://www.edatop.com/peixun/antenna/116.html

我们的课程优势：

※ 成立于 2004 年，10 多年丰富的行业经验，

※ 一直致力并专注于微波射频和天线设计工程师的培养，更了解该行业对人才的要求

※ 经验丰富的一线资深工程师讲授，结合实际工程案例，直观、实用、易学

联系我们：

※ 易迪拓培训官网：http://www.edatop.com

※ 微波 EDA 网：http://www.mweda.com

※ 官方淘宝店：http://shop36920890.taobao.com