硕士学位论文

基于 TPM 的可信嵌入式平台的设计与实现

DESIGN AND IMPLEMENTATION OF TPM-BASED TRUSTED EMBEDDED PLATFORM

李然

哈尔滨工业大学

2012 年 7 月

中图分类号：TP315　　学校代码：10213
UDC：621.3　　密级：公开

工 学 硕 士 学 位 论 文

基于 TPM 的可信嵌入式平台的设计与实现

硕 士 研 究 生：李 然
导　　师：柏军副教授
申 请 学 位：工学硕士
学　　科：计算机科学与技术
所 在 单 位：计算机科学与技术学院
答 辩 日 期：2012 年 7 月
授予学位单位：哈尔滨工业大学

Classified Index: TP315
U.D.C: 621.3

Dissertation for the Master Degree in Engineering

Design and implementation of TPM-based trusted embedded platform

Candidate： Li Ran
Supervisor： Associate Professor Bai Jun
Academic Degree Applied for： Master of Engineering
Specialty： Computer Science and Technology
Affiliation： School of Computer Science and Technology
Date of Defence： July, 2012
Degree-Conferring-Institution： Harbin Institute of Technology

摘　要

当前是信息触手可及的时代，各种各样的嵌入式产品层出不穷。因此，对于专用领域的嵌入式系统来说，构建一个安全的终端极为重要。TCG 提出的可信计算跳出了传统解决信息安全问题的技术范畴，以基于硬件的 TPM 安全芯片和相应的软件为出发点，首先解决终端的安全问题，进而逐渐扩大安全可信的范围，最终构建整个系统的可信与安全。

本论文从实际入手，将可信计算与嵌入式系统结合起来，构造基于 TPM 芯片的嵌入式开发平台，为在嵌入式设备中应用 TPM 提供支持。利用该平台可开发基于可信计算的适应某领域的专用嵌入式产品或应用，这是未来嵌入式产品开发的主流，也是解决嵌入式系统所面临各种安全威胁的有力技术手段。本平台使用了主流嵌入式操作系统 Linux 以及嵌入式处理器 ARM9-S3C2410，TPM 芯片采用 Atmel 公司的 AT97SC3204T。硬件方面设计了一个基于 TPM 芯片的通用开发板，与传统开发平台相比加入了双口 RAM、CAN 总线、CPLD 等模块，使其功能性更强。软件方面编写了专用的 TPM 芯片底层驱动，并根据嵌入式系统的特点，重新设计了可信的引导程序 Bootloader，确保了引导程序的安全以及信任的传递。同时在嵌入式操作系统中移植了 TSS 软件协议栈，为使用 TPM 实现更多安全功能提供了 API 接口，使开发更方便。最后，利用该平台实现了基于 TPM 的安全身份认证模型，与传统的身份认证方法相比更具安全性和实用性。

目前，该平台软硬件已经设计开发完成，并通过了测试。与传统嵌入式开发平台相比，可信嵌入式平台功能性更强、安全性更高，利用 TPM 实现各种安全功能的同时又没有破坏嵌入式系统低功耗、低成本、实时性的特点。基于该平台可方便地模拟专用的可信模块或可信设备，同时也支持用户构建个性化的可信应用，支持灵活多变的开发模式。

关键词：可信计算；嵌入式；TPM

Abstract

Embedded systems are the hottest one of the most promising areas of IT applications currently. Therefore, build a secure Terminal in Embedded systems is extremely important. TCG's trusted computing jumped out of the traditional technical areas to solve information security problems. Based on the TPM security hardware device and the appropriate software to start, first of all address the security concerns of Terminal, and gradually expand the scope of security and credibility, ultimately building system-wide total confidence and security in the area.

This paper proceeds with the actual that combining trusted computing and embedded systems to structure TPM-Based embedded development platform. It provides support for TPM technology in embedded devices. Using this platform to develop adaptation based on trusted computing a dedicated embedded product or application area, this is the future of embedded product development mainstream. It is also a powerful address security threats faced by embedded systems technologies means. In this platform, the OS use mainstream embedded operating system Linux and TPM chip use Atmel Corporation AT97SC3204T. Embedded processor chip is ARM9-S3C2410. Hardware design of a Development Board based on TPM chips, compared with traditional development platform has joined modules such as dual-port RAM, CAN bus, CPLD. These modules make the platform more powerful. Software includes special TPM chip driver and redesign of the trusted bootloader that based on the characteristics of embedded systems.It ensure the security and trust of a boot pass. Simultaneous transplantation of TSS in the embedded operating system software protocol stacks achieving more security to use the TPM feature provides API interfaces. It makes development easier. Finally, It use the platform to realize TPM-based security authentication models. It compare with the traditional method of authentication more secure and practical.

At present, the platform software and hardware have been designed to develop complete and passed the test. Compared with the traditional embedded development platform, the trusted embedded platform is more powerful and stronger security. Using TPM implements a variety of security features without damage for embedded

systems features like low-power, low-cost, real-time and so on. Based on the platform can be easily simulated special module or a trusted device, while also supporting the trusted application for users to build personalized, supporting flexible development models.

Keywords: trusted computing, embedded system, TPM

目　录

第1章 绪 论

1.1 前言

当前是信息触手可及的时代，各种各样的嵌入式产品层出不穷，像我们平常见到的手机、PDA、电子字典、可视电话、VCD/DVD/MP3 Player、数码相机（DC）、数码摄像机(DV)、U-Disk、机顶盒(Set Top Box)、数字电视、汽车电子、机器人玩具、导航仪、核心交换机、数控设备机床、汽车电子、智能家电、X 光机等等都是典型的嵌入式产品[1]。

传统计算机系统的安全性保障主要利用底层 CPU 的内存管理功能，由操作系统完成，无可避免地要出现漏洞。增强计算机的安全性已经无法逃避，而计算机性能的提高也使安全性开销变得可以接受。目前比较有代表性的增强安全性的方法：一是在 CPU 内部增加更多的安全功能，例如增加安全级别，增加专有指令等方法，其代表性的应用就是虚拟机，IBM、Intel 和 AMD 都为虚拟机提供了专用的指令；二是使用专用的用于安全性的芯片或模块，这种方式的雏形或许来自于以往常见的软件狗，然而软件狗提供的安全性能支持是有限的。基于硬件来防护计算机的安全性已逐渐成为了共识，从守护神 Palladium 防护系统（Microsoft 于 2002 年提出，2003 年改名为次世代安全计算基 Next-Generation Secure Computing Base），Intel 的可信执行技术 TXT（Trusted Execution Technology，基于 TPM，以前的代号是 LaGrande，以 TPM1.2 版本为主，PC 相关芯片都会支持与内建 LT），TCG 的可信计算技术，到 Intel 的主动管理技术 AMT （Active Management Technology）等等，新近的发展都集中在以 TPM 芯片为核心的可信计算。

可信计算的核心是基于硬件加密技术的 TPM 安全芯片。TPM 芯片是一个含有片上系统用于计算机安全的芯片。TPM 芯片由微处理器、ROM、随机数发生器、密钥生成部件等硬件以及位于芯片内部的片上系统组成[2]。TPM 芯片符合可信赖平台模块的标准，具有完整性度量、身份认证与识别、数据加密等功能。TPM 是物理上对应主板上的一个与各种密钥运算相关的新型芯片，可对各种重要数据进行密钥运算，确保关键数据的可靠存储不被破解和泄密。与传统的软件保护信息安全相比，要破解加密数据，首先需要破解 TPM 芯片内存储的密钥，这是相当困难的，即使成功，代价也会相当高。基于 TPM 的可信计算技术从硬

件层次形成对计算机系统的全面防护，构建可信的安全的体系环境，将最终使得困扰计算机系统的病毒、木马等恶意软件得到有效屏蔽，无法攻击。

TPM 芯片中的密码运算部件是整个可信计算平台的核心，在独立的计算机系统中加入一个对外保密性很强的密码运算部件，未经授权或者非法的用户就无法获得密钥解密内部数据，从物理上保护了整个计算机系统的安全。可信平台模块提供软件配置鉴定和安全存储，TPM 具有报告平台状态、安全存储、安全身份标识等功能。TPM 与杀毒软件防火墙等软件的提取特征值的原理不同，不仅仅对数据进行物理上保护，还可以从被攻击篡改的数据值中恢复到原始数据，从而更加保证计算机系统的完整性。基于现有 PC 架构，配置上 TPM 及相应的软件(TSS)，实际上形成了跨平台软硬件系统的可信计算体系结构。

1.2 课题来源

本课题来源于山东省高等学校科技计划项目《基于 TPM 的可信嵌入式开发平台的研究与实现》。

1.3 背景及意义

2011 年底，CSDN 数据服务器被攻击引发了泄密门事件，为此，产生了连锁反应，许多 QQ、人人、甚至银行的用户由于设置了与 CSDN 相同的账号与密码而导致账号被盗，甚至是经济损失，信息安全的形势尤为严峻。计算机系统的底层硬件安全与可信的操作系统是整个系统安全的基石。加密解密、防火墙、身份认证等技术是解决信息安全问题的关键手段。随着 Web2.0 时代的到来，影响和威胁传统计算机与网络的信息安全问题，并没有因为各种软硬技术措施的采取而在过去的若干年内得到很好解决，相反双方愈来愈深陷入无休止的攻防之中，不仅如此，这些影响和威胁已开始逐渐向嵌入式系统蔓延，成为信息安全领域的一个更加棘手而难于解决的顽疾。而且，必须从整个信息系统的最底层开始，从下到上包括芯片、驱动、OS、网络信息流等方面综合采取措施，才能确保系统的安全，这一技术思想催生了可信计算。事实上，由于“空芯”缘故，我国信息领域面临严峻的安全威胁，信息安全“后门问题”一直存在。在信息安全领域，由于构成计算机的核心要素，CPU、内存、主板、BIOS、操作系统等长期以来一直被国外几个大厂商所垄断和把持，致使我国的“信息安全大门”一直没有关上，这其中，政府、军队、金融、通信等重要部门与关键行业，信息安全严峻形

势尤为突出。虽然高校和国内各学术与科研机构采取了若干技术措施，但关键部件尚未实现，关键技术尚未突破，仍会在较长时间内应用不可知和不可控的国外信息产品。因此，大力发展我国的信息安全产业尤为紧迫。可信计算的出现，对于我国既是机遇更是挑战。如果建立了自主的可信计算标准、规范，实现了自主可控的可信计算核心部件和产品，我国的信息安全又多了一道安全屏障，否则信息安全的威胁会更加恶化。计算机平台可能遭遇的攻击形式如图 1-1 所示。

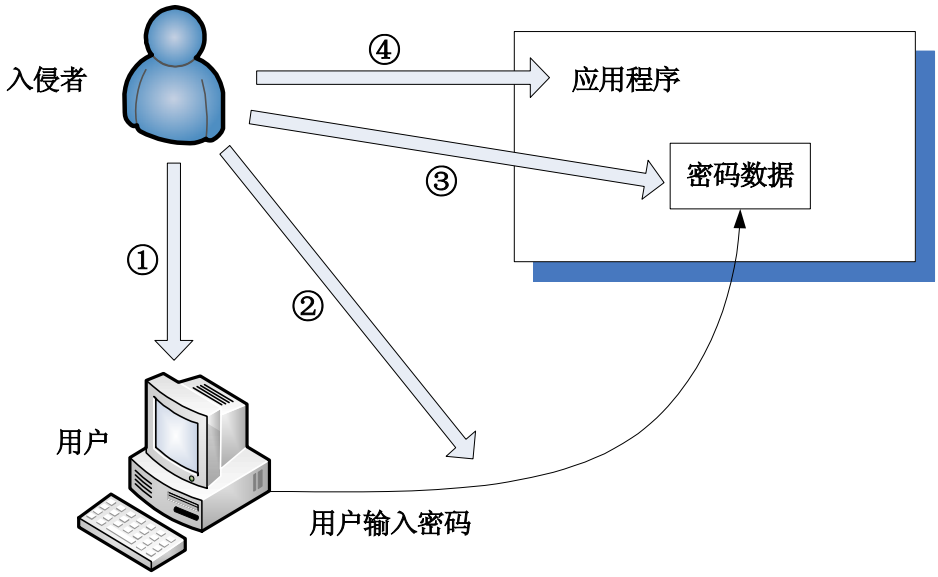


图 1-1 入侵者的几个攻击途径

在图 1-1 中，入侵者的攻击途径大致分为四种，其中①为入侵者伪造身份直接登录系统；②为入侵者利用木马等工具截获用户输入的密码或秘密邮件；③为入侵者直接在内存中读取密码数据；④为入侵者更改程序路径等手段欺骗应用程序而绕过密码登录等环节。对于专用领域嵌入式系统来说，信息安全问题十分严峻，一个安全的终端环境极其重要。当前时代是嵌入式的时代，各种嵌入式设备的处理计算能力得到了飞跃的发展，有些甚至超越了传统的 PC 机。随着 3G、wifi 等通信网络技术的迅速发展普及，针对嵌入式设备的上层应用得到了指数级的增长，比如手机上数据 GPS 导航、掌上淘宝、网上银行的电子支付服务等等。嵌入式系统易受攻击、安全性差，往往会造成了严重的损失，损失范围扩展到各种应用程序和工业范围内，除了直接的经济损失外还包括丢失各种盈利机会。由于嵌入式系统的经济利益得到了持续发展，而自身又不具备传统 PC 所具有的安全特性，所以嵌入式系统上的安全问题越来越引起人们的广泛关注，吸引了一大

批专家学者进行探索研究。到目前为止，大多数实现嵌入式安全的案例都集中在软件层次，包括操作系统防御和嵌入式杀毒软件等等。然而操作系统的定义是公开的，而且操作系统是非常复杂的软件系统，因此仅仅靠这些是不可能提供一个安全的环境。在某些关键领域，如军事上的保密通信、武器控制，银行的 ATM 终端，智能终端如手机的网上交易等等，嵌入式系统所需的安全性越来越高，哪怕一个微小的硬件故障或者黑客入侵都有可能造成灾难性的后果。

把可信计算的思想与传统嵌入式系统融合起来，开发基于可信计算的适应某领域的专用嵌入式产品，是可预见未来嵌入式产品开发的主流，是解决嵌入式系统所面临各种安全威胁的有力技术手段。

1.4 国内外研究现状分析

1983年，美国国防部制定了世界上第一个《可信计算机系统评价准则》（Trust Computer System Evaluation Criteria,TCSEC），在TCSEC中第一次提出可信计算机（Trusted Computer）和可信计算基（Trust Computer Base, TCB）的概念。1999年，IBM、HP、Intel、微软等著名IT企业发起成立了可信计算平台联盟（Trust Computing Platform Alliance, TCPA）。TCPA的成立，预示着可信计算进入了新的更高层次的发展空间。到2003年，TCPA改名为可信计算组织（Trust Computing Group, TCG），使可信计算真正成为一门系统的学科。2006年，欧洲启动了名为“开放式可信计算”的可信计算研究计划[3]。TCG可信计算组织提出了可信计算机平台的基本概念、系统结构及技术路线等等。同时，TCG还针对不同的平台，将技术路线具体化到个人电脑、掌上电脑、服务器和嵌入式平台等。

在国内，我国一直紧跟国外的脚步，政府对可信计算给予了极大的研究经费支持。中国国家密码管理委员会制定了可信计算模块的标准，中国科技部的863计划开展了关于可信计算技术领域的专题研究。国家自然基金委对“可信软件”展开了重大专项研究计划支持。

TPM 芯片是一个具有较高集成度的 SOC 产品，其生产和封装需要较高的技术和工艺。国外生产厂家主要有 Atmel（爱特美）、Infineon（英飞凌）、Broadcom 、ST（意法半导体）和 Winbond（华邦电子）等，国内主要有 Sinosun（兆日科技）（TPM 安全芯片 SSX35）及其解决方案、中兴、联想（恒智芯片）、浪潮、瑞达、同方（SSX0908 可信计算安全芯片）等。

目前，很多笔记本都集成了 TPM 安全芯片，用以实现各种密钥及相关运算和安全保护等基本功能。Lenovo/IBM、HP/Compaq、Dell、NEC 、Sony、Toshiba、

Samsung、Fujitsu、Acer、Gateway 等厂商均已在其生产的安全 PC 中引入了安全芯片，构筑安全环境。在北美，有超过 60%的计算机上安装有 TPM 芯片。服务器上（IBM、Gateway 的 PC Server）也逐渐开始应用 TPM 技术，但在嵌入式计算机中却是刚刚起步，手机、PDA 消费类产品尚未支持 TPM。

为了满足各个国家 IT 厂商，甚至是终端用户对安全性增强的迫切要求，可以预见，在接下来的几年内对可信计算机的需求会越来越急迫。目前很多计算机厂商生产的笔记本电脑和台式机主板上都安装有 TPM 芯片，基于可信芯片的计算机产品或许会像 3C 认证一样，成为厂商生产计算机和政府等部门采购的强制性标准。TPM 芯片及应用如图 1-2 所示。

(a) Infineon TPM 安全芯片

(b) 笔记本电脑主板上的 Infineon TPM 安全芯片

(c) 兆日公司生产的可信安全芯片

(d) 内置 TPM 芯片的 14.1 英寸笔记本电脑

图1-2 TPM芯片及应用

在安全措施上，金融领域采用较为高层的防范措施，物理上通过客户机上挂载基于USB接口的硬件来实施。个人用户为了确保数据安全可靠，采用U锁来进行防护，U锁无存储功能，只是基于USB接口的智能卡，用于实现开机身份识别、数据加密保密等功能。

中国工商银行采用一种客户证书U盾产品（移动数字证书），其中存放着用户的数字证书，并不可读取。U盾可以对网上交易的信息进行加解密和安全的数字签名，确保网上银行的真实性、机密性、完整性[4]。这些基于USB的设备，每次使用都需要插接上，容易丢失，依赖于操作系统，本质上还是基于软件的防护。U盾和U锁产品实物图如图1-3所示。

U锁

U盾

图1-3 U锁和U盾

相比之下，基于硬件的软硬结合的安全技术更加可靠，通过在主板上增设TPM安全芯片，配置以相应的软件协议栈后，整个计算机提供给用户的是一个安全可信的系统，并没有给用户额外的负担。

目前，在银行领域，TPM 除了在中心服务器上被使用以外，在各种终端，如ATM 机、税控机、酒店或超市 POS 机还未涉及，未来这些嵌入式设备将成为 TPM 应用的乐土。

基于 TPM 的各类密钥运算以及安全认证等服务，可为高度重视数据与信息安全的公司提供更为健全的安全解决方案。尤其在金融领域，为了安全采取了越来越多复杂的技术，但依然存在严重漏洞和安全隐患，基于 TPM 的安全技术有望成为理想的技术之选。

现在TPM作为一个安全芯片被独立的集成在主板上。随着TPM芯片价格的不断下降，应用领域和成熟度的不断扩大，以及用户的认可，TPM芯片可能会被作为计算机主板的常规组件而被集成到特定芯片组中，并最终被集成到CPU内而成为常态。

从上面可以看出，TPM 在嵌入式设备上的应用刚刚开始，但随着国际国内对安全性能的要求，在一些领域，如政府、国防军事、银行、电信、电子商务、金融证券等，TPM 很可能势在必行，甚至可能在法律上成为一种强制性要求，使得 TPM 安全芯片成为计算机系统构成的必要标配。

中国工程院院士沈昌祥指出，作为国家信息安全基础建设的重要组成部分，自主创新的可信计算平台和相关产品实质上也是国家主权的一部分[5]。目前，国内很多学者做了关于可信计算的相关研究工作，包括文献[6][7]中的可信PDA，可信中的信任链与度量模型等。张焕国教授等人做了大量关于可信计算的工作，

其中文献[7]针对可信计算平台信任链规范的信息流安全问题，用安全进程代数的方法对信任链系统接口进行形式化建模。郑宇等人在文献[8]中针对移动终端(ME)的特性,提出了结合USIM和TPM的可信移动平台(TMP)，但并没有上层软件的实现。冯登国教授在文献[9]中从构建可信终端的信任入手，建立了基于信任度的信任模型，给出了基于信息流的动态信任链构建方法，一定程度上解决了终端信任构建的实时性、安全性等问题。但是关于嵌入式可信方面的研究还比较少，并且不够具体，相对缺少包括硬件平台、可信Bootloader、可信操作系统等完整的实现。本课题就立足于已有研究成果之上，弥补嵌入式可信部分的相关缺失，构建出一个完整的可信嵌入式平台。

1.5 本论文的主要工作内容

可信嵌入式开发平台主要针对可信嵌入式应用系统的关键和共性问题开展基础性和应用性研究。可信嵌入式开发平台以“ARM9+嵌入式 Linux 操作系统”为核心，采用 ETPM(嵌入式 TPM 芯片)为安全硬件，在 TPM 的软件协议栈的配合下，从 Bootloader、各类驱动到内核进行全面可信性改造，构建典型的嵌入式可信开发平台，弥补针对嵌入式可信这部分的空缺，并在随后研制出具有市场前景的可信嵌入式产品。本文共五章，每章的具体内容如下：

第一章主要介绍了可信嵌入式开发平台的研究背景及其意义，以及可信计算在国内外研究和发展情况，并结合嵌入式的特点总结了本课题的技术可行性，最后概括了本文所做的主要工作内容。

第二章对可信嵌入式平台整体进行了架构设计，包括外围板和核心板的硬件设计，TPM 芯片的选取，并分析了其总体硬件结构，各个模块组成，以及在硬件之上的可信软件层次模型。

第三章详细介绍可信嵌入式平台硬件设计原理图，包括核心板和外围板的硬件设计，TPM 与核心板的连线，以及 CAN 总线、双口 RAM 等其他各模块的硬件连接。

第四章首先根据 S3C2410 中 IIC 的特性编写了 TPM 的驱动程序，编译成功后，以模块的形式添加到内核中供上层应用程序调用，然后改造了引导程序 U-boot，利用 TPM 的 SHA-1 算法功能加入了对引导程序的度量功能，实现了可信的启动过程，最后将 TSS 软件协议栈根据嵌入式系统的特点移植到嵌入式操作系统中。

第五章对可信嵌入式平台的功能和性能进行了测试，完成了基于 TPM 的安

全身份认证功能，并对可信嵌入式开发平台的性能做出评估。

最后对本文的研究开发工作进行了概括性总结，并提出了需要改进和完善的地方。

第2章 可信嵌入式平台总体设计

2.1 引言

可信嵌入式开发平台设计完成后可提供硬件模块、各类可信软件开发所需的API接口。基于该平台可方便地模拟专用的可信模块或可信设备，同时也支持用户构建个性化的可信应用，并在此基础之上开发符合自身需求的嵌入式可信应用软件和产品，支持灵活多变的开发模式。

2.2 平台构成

可信嵌入式平台包括：以嵌入式CPU、SDRAM、Flash和TPM芯片为核心的基础硬件和其他功能模块，以板级开发包、嵌入式可信Bootloader、嵌入式可信操作系统内核和 TSS 为核心的软件体系。可信嵌入式平台基本构成如表 2-1 所示。

表 2-1 可信嵌入式平台基本构成

可信嵌入式平台构成要素	说明	备注
嵌入式微处理器	Samsung 公司基于 ARM9 的 S3C2410	嵌入式 CPU
嵌入式 TPM	Atmel 公司的 AT97SC3204T	可信芯片
BootLoader	U-boot	可信改造
嵌入式操作系统	基于开源的嵌入式 Linux 操作系统	移植 API 接口

开发可信嵌入式开发平台，首先需要完成硬件电路板的设计与制作，该部分工作主要是在确定了物理硬件芯片和电路元器件之后，进行硬件相关连接设计，绘制板图，制板，并调试，搭建硬件基础平台。该平台采用核心板与外围板组合的设计，核心板包括嵌入式CPU 三星 S3C2410，两块现代 HY57V561620TP－H 内存颗粒组成 64M SDRAM，一块三星公司生产的 K9F1208 NAND Flash 存储器。外围板包括 TPM 芯片、以太网控制器 DM9000、电源、串口、双口 RAM、CPLD、CAN 总线、USB 接口、RJ-45 接口、音频接口、AD 转换等模块。硬件基本结构框图如图 2-1 所示。

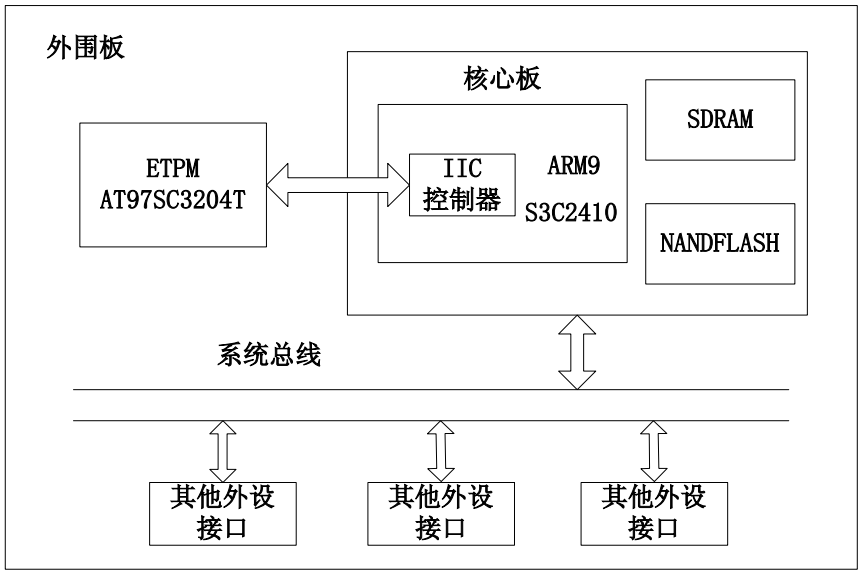


图 2-1 可信嵌入式开发平台基本结构框图

2.3 核心软硬件介绍

2.3.1 ARM9-S3C2410

S3C2410 处理器是韩国 SAMSUNG 公司生产的采用 0.18um 制造工艺的 32 位微处理器。该处理器内部结构如图 2-2 所示。

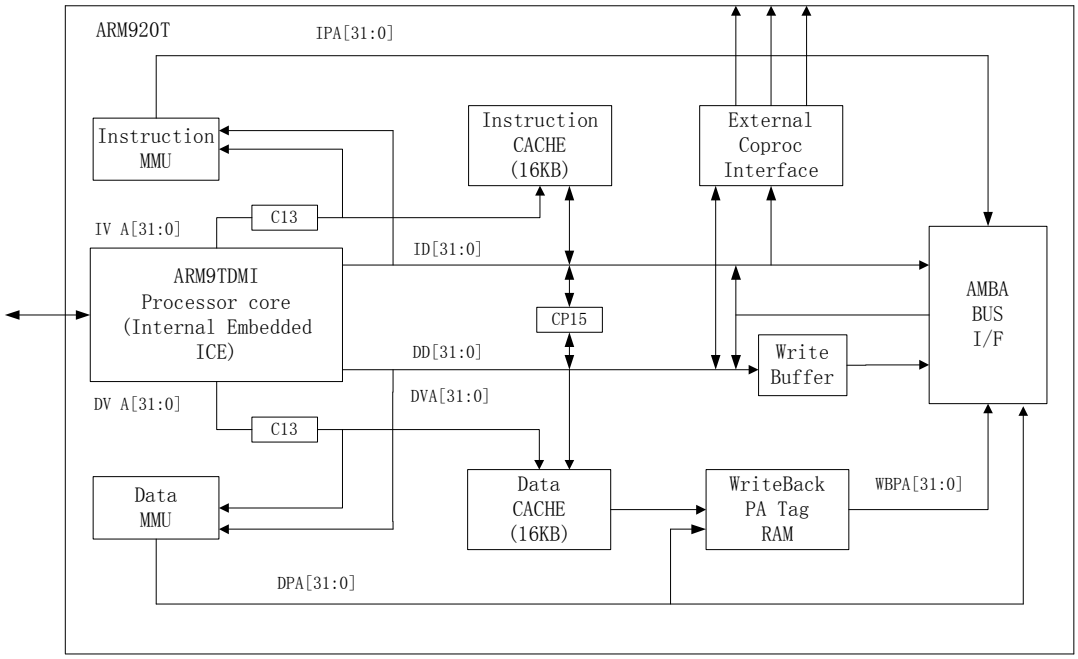


图 2-2 S3C2410 内部结构框图

2.3.2 TPM 芯片-AT97SC3204T

TPM 安全芯片是计算机系统“硬安全”的基础，通过配备必要的支撑软件，建立全面的安全解决方案已成为信息安全领域的一个重要解决之道。基于 TPM 的计算机安全性能之所以较高，是因为其安全性能是靠硬件支撑的。由传统应用软件单一策略来进行防护的措施到向基于 TPM 安全芯片而采用的软硬结合的综合技术措施，可为用户实现对计算机系统的真正防护[17]。

目前，在嵌入式领域，内置有 ETPM(嵌入式 TPM)芯片，并具备相配套的软件支撑环境的产品和开发平台很少见，但是随着手机、平板等各种设备带给人们信息触手可及的便捷，包括其他各类嵌入式产品和系统的广泛应用，基于 ETPM 产品和系统势必会普及开来。

Atmel 公司的 AT97SC3203S（遵循 TPM1. 2 规范）和 AT97SC3204T（遵循 TPM1. 2 规范）两款芯片都是 Atmel 公司针对嵌入式领域开发的 TPM 产品。其中，AT97SC3203S 是采用 SMBus 总线协议的 TPM 芯片。AT97SC3204T 采用 AVR RISC 微处理器，符合 TPM1. 2 规范，提供单芯片整体解决方案，硬件上包括不对称密码引擎、2048-bit RSA、真随机数发生器 RNG、内置 EEPROM 存储 RSA 密钥，采用兼容 IIC 的 TWI(Two-wire Serial Interface)总线接口与外部通信。AT97SC3204T 内部结构如图 2-3 所示。

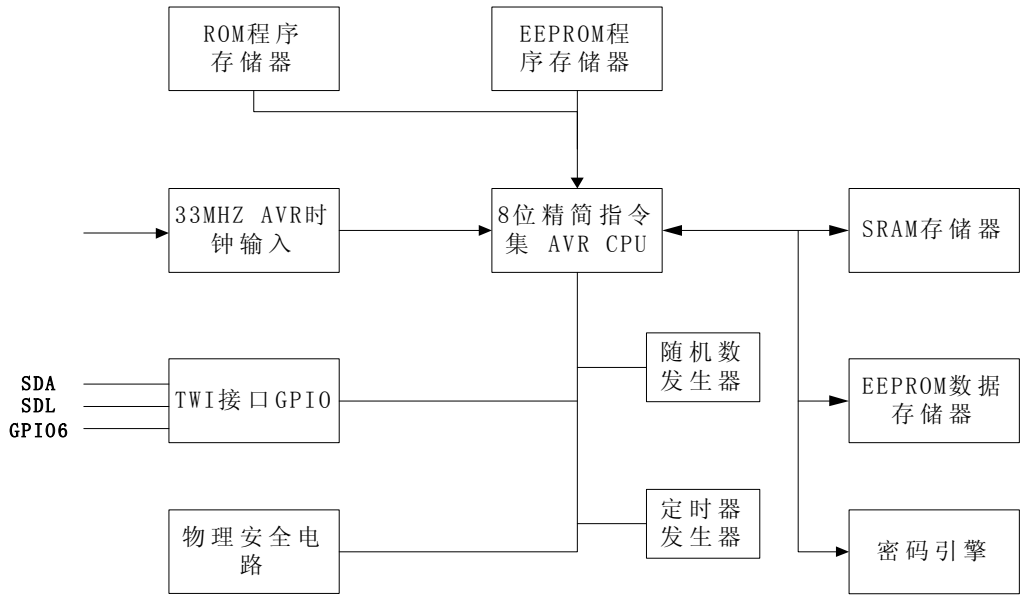


图 2-3 AT97SC3204T 内部结构框图

借助 ARM 嵌入式微处理器芯片上的 IIC 总线接口和 SMBus 总线接口（SMBus 总线其大部分基于 IIC 总线规范）来与 TPM 连接。IIC 是一种工作在主/从模式的二线制串行总线接口。AT97SC3204T 芯片采用 TWI (Two-wire Serial Interface)与外界进行通信，由于 IIC 与 TWI 信号与接口时序基本兼容，因此容易实现通过 ARM 芯片的 IIC 来进行连接 TPM 芯片。

2.3.3 引导程序 U-boot

U-boot 是开源的多平台支持的引导程序。U-boot 不仅仅支持嵌入式 Linux 系统的引导，它还支持 NetBSD， VxWorks， QNX，RTEMS，ARTOS，LynxOS 嵌入式操作系统。U-boot 的代码组织结构、编译过程等几乎与 Linux 内核一致。就目前来看，U-boot 对 PowerPC 系列处理器支持最为丰富，对 Linux 的支持最完善。

本平台之所以选取 U-Boot 作为可信 Bootloader 改造的蓝本，因为 U-boot 具有以下诸多优点：

（1）开放源代码，用户可以根据自己的需求进行修改，这是最重要的特点。

（2）支持本平台所使用的嵌入式操作系统内核 Linux，同时还支持其他嵌入式操作系统。

（3）支持本平台所使用的 ARM 架构处理器。

（4）稳定性与可靠性高。

（5）丰富的设备驱动支持。

2.3.4 TSS 软件协议栈

TSS(TCG Software Stack)是与 TPM 相配套的支撑软件，由多个软件协议层组成，包括 TCG 设备驱动库、TSS 核心服务层、TCG 服务提供者。TSS 对外提供给用户管控 TPM，对内操控 TPM，并与操作系统相结合[10]。嵌入式系统的一个基本特征是硬件系统与软件系统结合紧密，TPM 提供的各种功能，要通过其软件支撑体系 TSS(TCG Software Stack)来完成，上层应用通过 TSS 提供的 API 可以对 TPM 进行访问。为了适应嵌入式系统的要求，需要对 PC 下的 TSS 进行移植，使之能够在专用的嵌入式平台下运行。

嵌入式 TSS 的开发有两种方案：第一种是利用 BIOS、嵌入式操作系统平台和 TPM 驱动来直接进行开发实现。由于 TPM 管理的复杂性，这种开发方式难

度较大，对上层应用开发者提出的要求更高。第二种是对现有成熟的 TSS 实例进行改造和移植。由于嵌入式系统的特殊要求，需要对 PC 环境下的 TSS 进行改造，使之能够适应可信嵌入式开发平台的要求。

目前比较流行的基于 TSS 规范的实例是由 IBM 研发的 Trousers，其遵循 TPM 规范。Trousers 软件包主要包括两大功能部分：TPM Tools 部分和 TPM PKCS#11 命令，其中 TPM Tools 部分是一套用来管理 TPM 的程序。Trousers 可完成与 TPM 建立会话、密码计算与密钥管理、数据存储与维护等功能[11]。

在 Trousers 的移植过程中要保留其基本功能与关键功能，对于与可信嵌入式开发平台要求相关性较低部分进行简化，薄弱部分进行加固，确保满足要求。移植完成后，可信嵌入式平台整体软硬件架构如图 2-4 所示。

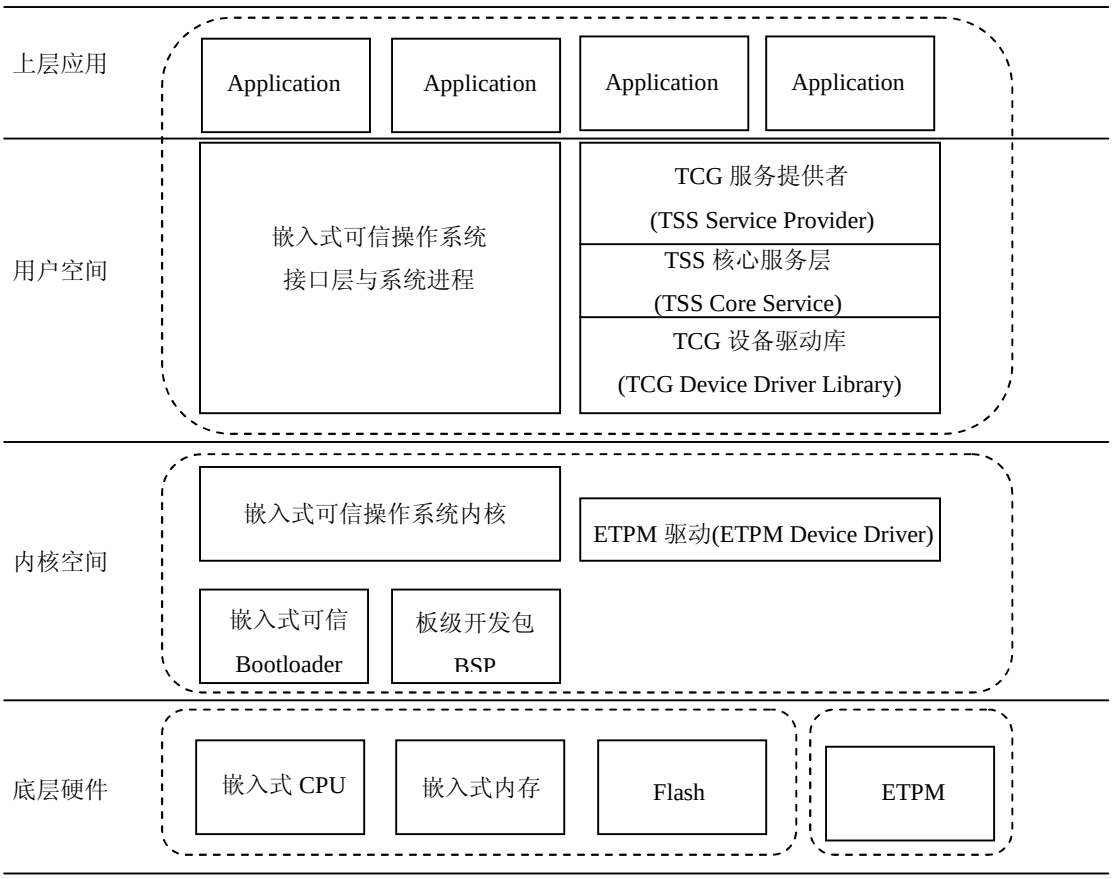


图 2-4 可信嵌入式平台整体软硬件架构

2.4 小结

传统计算机系统的安全性保障模式无可避免地要出现漏洞，以 TPM 芯片为

核心的可信计算技术为计算机安全提供了解决方案，目前 TPM 技术已经广泛地应用在服务器和笔记本电脑中。本课题旨在构造基于 TPM 的嵌入式系统开发平台，为在嵌入式设备中应用 TPM 提供技术支持。平台使用了主流嵌入式操作系统 Linux 以及嵌入式处理器 ARM-S3C2410，硬件方面设计了一个基于 TPM 芯片的通用的开发板，软件方面提供了通用的 TPM 芯片底层支持，并根据嵌入式系统的特点，对引导程序进行了可信改造以及移植了 TSS 协议栈。本课题的创新点有以下四点：

（1） 搭建了一个通用的基于 ARM9 和 TPM 芯片的硬件平台，完成了整体电路设计，利用它可方便地设计专用的可信模块或可信设备。

（2）实现了 TPM 芯片的底层驱动，并提供二次开发支持，为构造信任链提供支撑，同时也支持用户构建个性化的可信应用，支持灵活多变的开发模式。

（3）构造了适应嵌入式系统特点的信任链，从 Bootloader 到设备驱动、操作系统和应用层，充分保证了可信性的传递，使用该信任链能为嵌入式设备提供可信保障。

（4）移植了 TSS 协议栈并对其进行了相应改造，为上层应用开发提供了 API。

第3章　可信嵌入式平台硬件设计

3.1 引言

由于 PC 机自身软硬件设计的先天结构简化，系统很难区分执行状态，内部缺少越界保护等技术手段，再加上 PC 机上系统软件和应用软件的不断复杂化，在 PC 环境内建立可信环境异常困难。

PC、服务器是一个软硬件综合的复杂系统，其规模还在不断向前扩大。由于信任关系的复杂性，完全建立可信的 PC 与服务器系统是十分困难的。嵌入式系统由于其应用领域的专用性和系统对软硬资源的苛刻要求等，构建嵌入式系统领域的可信环境相对是比较容易的。

3.2 可信嵌入式平台主要硬件模块的设计

3.2.1 平台核心板

核心板主要器件包括：嵌入式 CPU 三星 S3C2410，2 片现代 HY57V561620D 作为 SDRAM，1 片三星公司 K9F1208 作为 NAND FLASH，XDL12M 晶振提供 CPU 主时钟。核心板采用 6 层板设计。核心板实物图如图 3-1 所示。

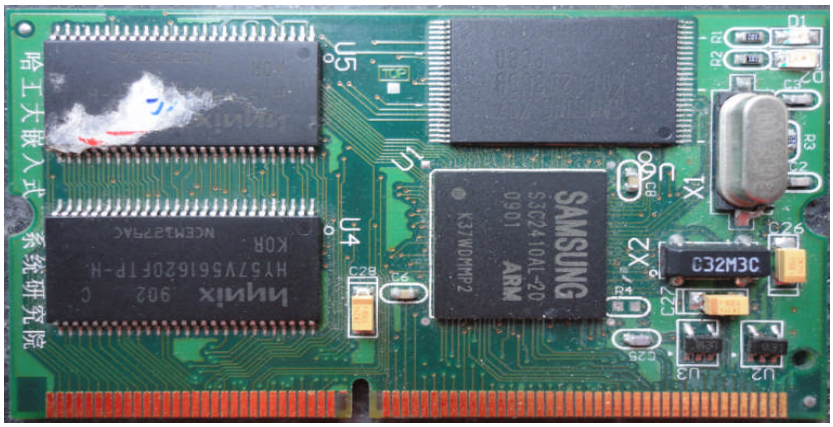


图 3-1 核心板实物图

核心板设计的主要功能是为了给外围板提供数据、地址和控制总线，通过扩展接口与外围板进行通信，形成了一个最小的计算机系统[14]。核心板体积小巧，

通过插槽与外围板相连，插槽连线包括系统总线、地址线、片选信号、中断和复用的 GPIO 等。核心板插槽硬件连接如图 3-2 所示。

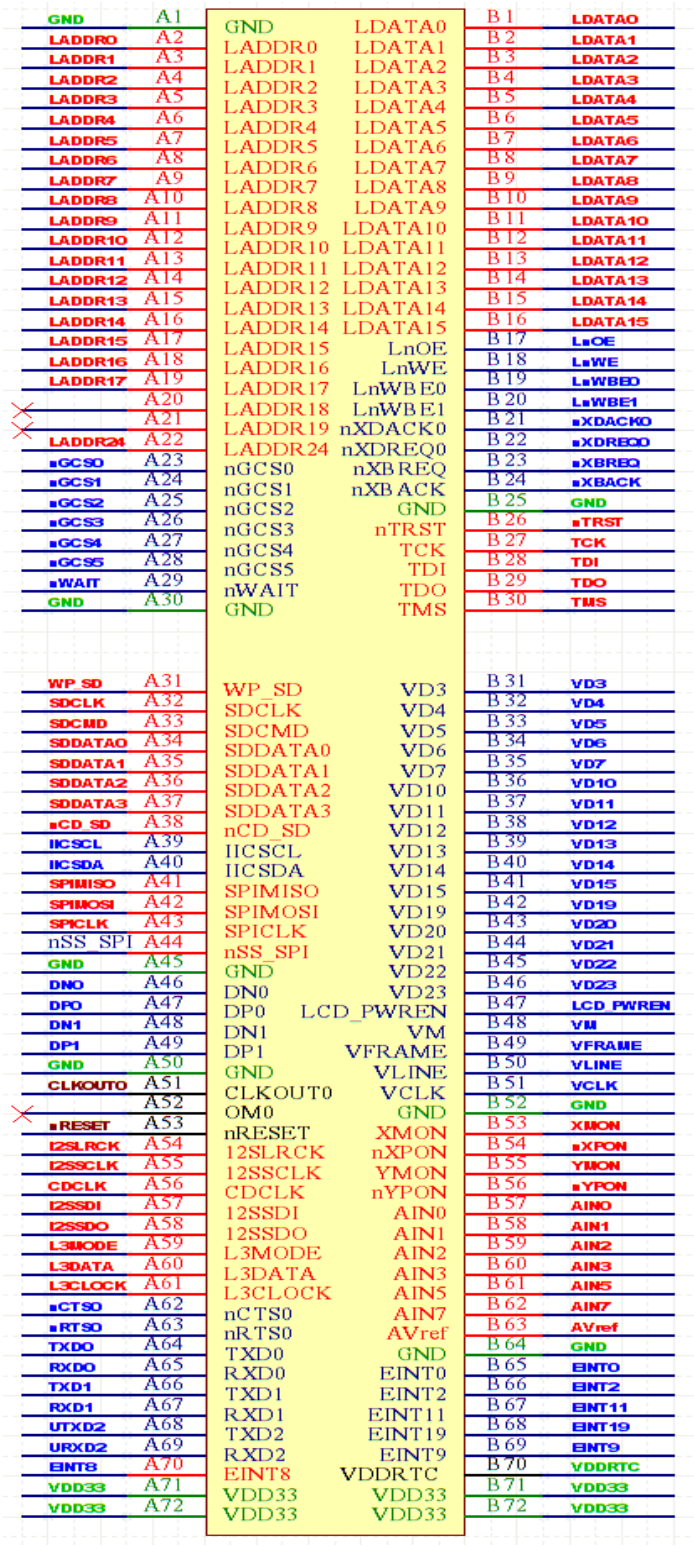


图 3-2 核心板插槽硬件连接图

3.2.2 网卡 DM9000

DM9000 是 DAVICOM 公司生产的百兆以太网控制器，由 1 个 25M 晶振提供时钟。DM9000 有 1 个一般处理接口，1 个 10/100M 自适应的 PHY 和 4K DWORD 值的 SRAM。DM9000 物理协议层接口完全支持使用 10MBps 下 3 类、4 类、5 类非屏蔽双绞线和 100MBps 下 5 类非屏蔽双绞线，符合 IEEE 802.3u 的规格，同时还支持 IEEE 802.3x 全双工流量控制。DM9000 通过 RJ45 口与主机或者互联网相连，能实现以太网的基本功能，包括 FTP、Telnet、TFTP、WEB 服务器等。 DM9000 硬件连接如图 3-3 所示。

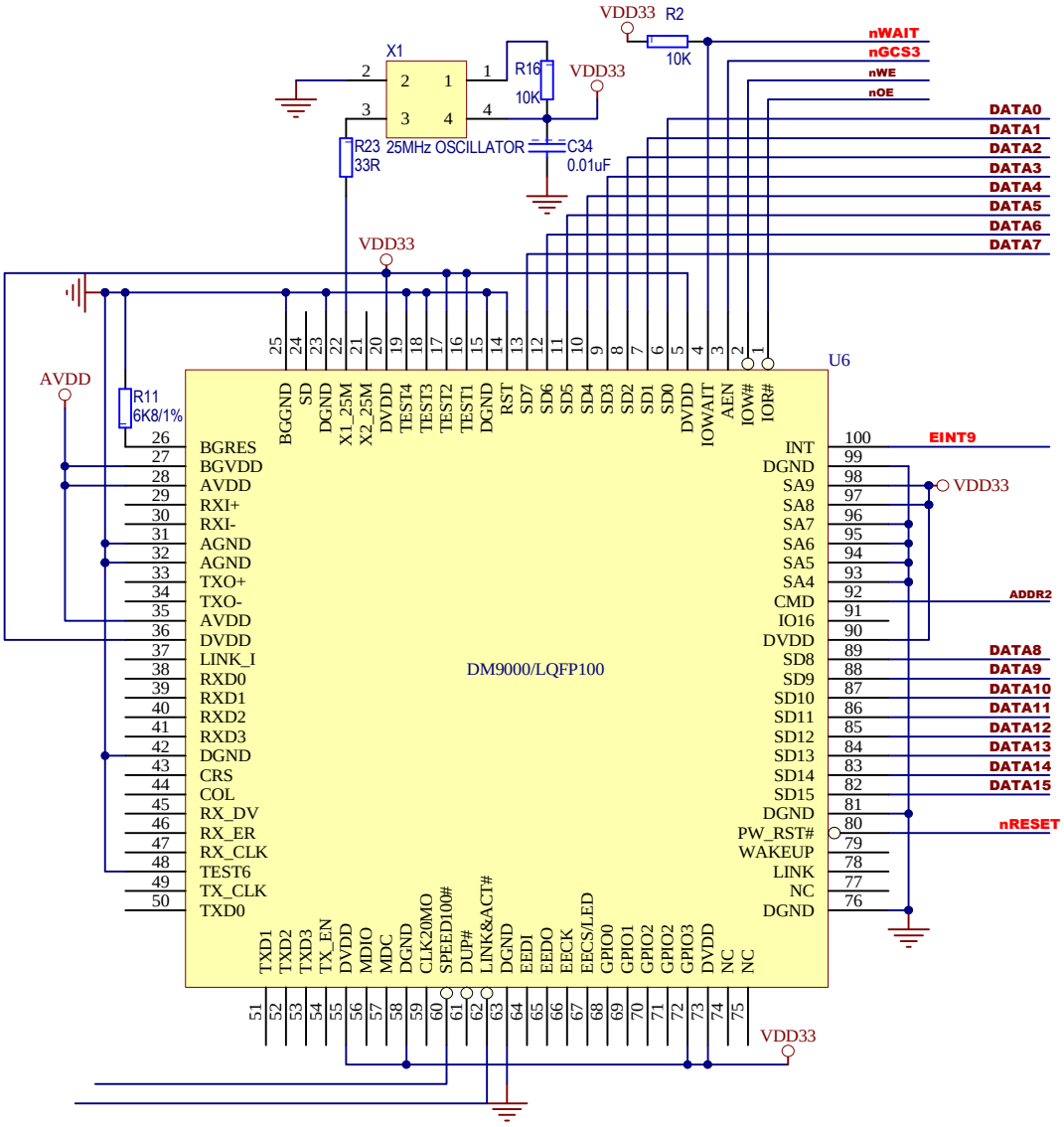


图 3-3 DM9000 硬件连接图

要实现以太网控制器DM9000的功能，第一部是对DM9000进行正确寻址。其中，AEN（地址允许)是输入引脚片选信号。SA4～SA9是地址总线4～9位，当AEN低且SA9和SA8高，而SA7、SA6、SA5、SA4为低时，则DM9000被选中。

DM9000 默认I/0 基地址为300H。DM9000的CMD引脚用于设置命令模式。其中，当CMD为高时，为数据端口。CMD为低时，为地址端口。数据端口和地址端口的地址码由下式决定：

DM9000地址端口=高位片选地址+300H+0H

DM9000数据端口=高位片选地址+300H+4H

DM9000高位片选的地址由核心板NGCS3产生，即为：0x100000000H，nWAIT为读写等待信号。由于在S3C2410中以太网卡的中断为9号中断，所以EINT9_ETHERNET为中断信号，RESET为网卡芯片重启信号，25MHz OSCILLATOR为芯片提供25MHz的工作频率，SD0～SD15数据总线与S3C2410的数据总线连接，最后将网卡驱动程序编译进U-boot和内核中，重新启动后就可以实现以太网的功能。在后期驱动程序和应用程序调试过程中往往要进行交叉编译，即用交叉编译工具生成可在ARM开发板上执行的程序，这时，可通过TFTP协议方便的进行可执行程序的传输。同样，U-boot也支持TFTP协议，可以用TFTP将内核镜像文件烧写到某地址。

3.2.3 双口 RAM

双口RAM即共享式多端口存储器。平台使用的是IDT70v07，是一款32k*8的带有左右两套读写控制逻辑的存储器。IDT70v07具有32k*8的数据存储阵列，选择8bit数据宽度具有很好的数据结构的兼容性，选择32k字节的存储比较适合流媒体的通信。IDT70v07具有旗语机制，旗语通信可以方便的实现通信的同步机制。对于需要同步的总线通信可以通过此方式合理的访问临界资源。例如，双端口RAM中的数据。IDT70v07具有中断机制，总线可以利用双端口RAM信箱通信机制方便的实现总线双向中断。

IDT70v07的外围连接可以分为4个部分：

（1）74LS138：产生不同的片选

（2）IDT70v07：双口 RAM

（3）CON4：与双口 RAM 相连，通过 40pin 连接线与另一开发平台的 CON5 相连

（4）CON5：通过左端口 40pin 输入口与 CON4 相连

IDT70v07的硬件连接如图3-4所示

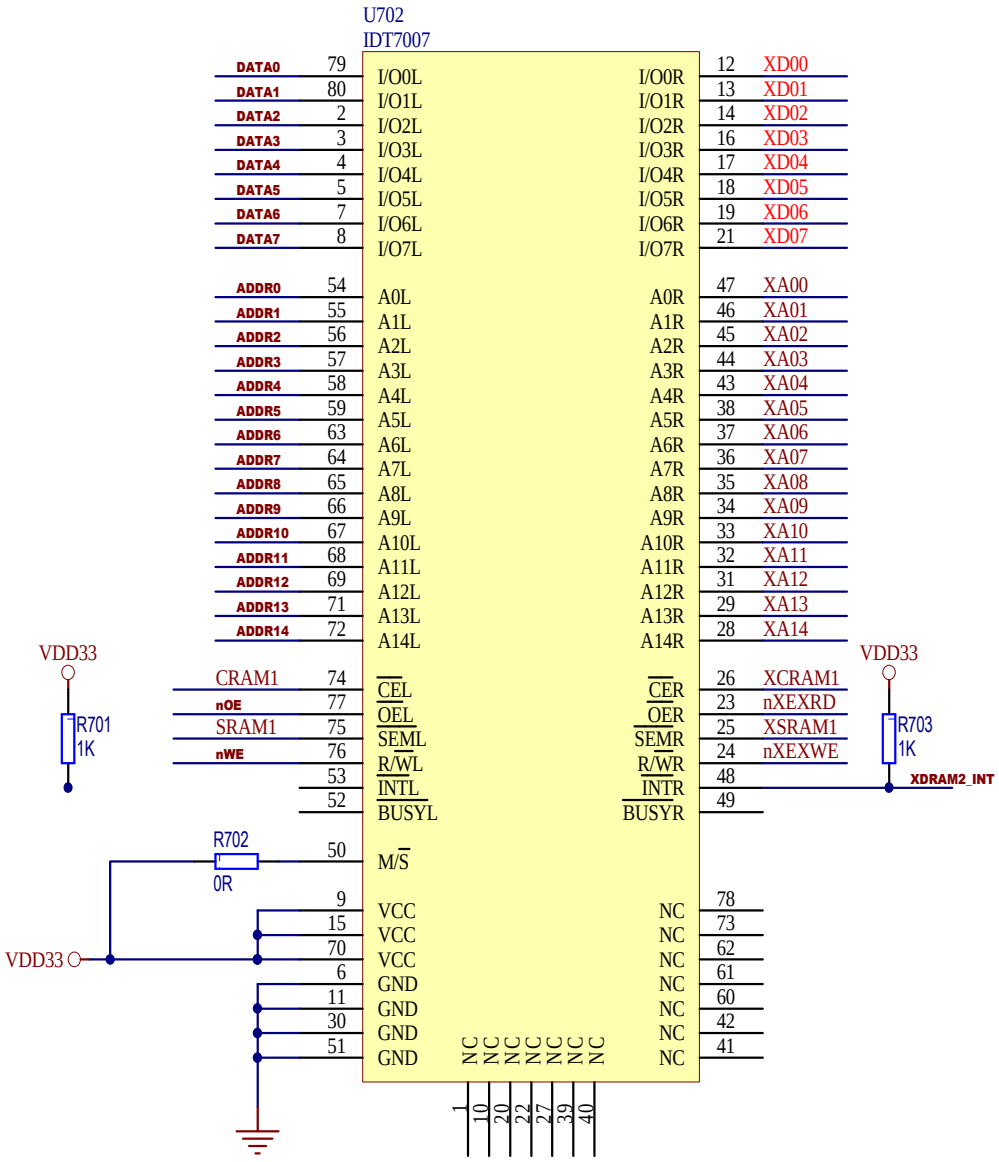


图 3-4 IDT70v07 硬件连接图

IDT70v07 中有一个 32k*8bit 的存储阵列，可以从左右两套数据线及地址线进行对双端口 RAM 数据的访问。这个大的功能模块又具体可以分为：本地数据读；本地数据写；远程数据读；远程数据写 4 个功能子模块。IDT70V07 具有 8 个旗语数据锁存器用来左右端口实现同步机制，旗语通信实际上可以归结为对 8 个锁存器的读写访问。这个功能模块又可以分为：本地旗语锁存器读；本地旗语锁存器写；远程旗语锁存器读；远程旗语锁存器写。

基于以上的设计总结出抽象的系统连接示意图，其中①，②，③，④，⑤，⑥分别表示不同的功能，这些功能被细分成6项，如图3-5所示。

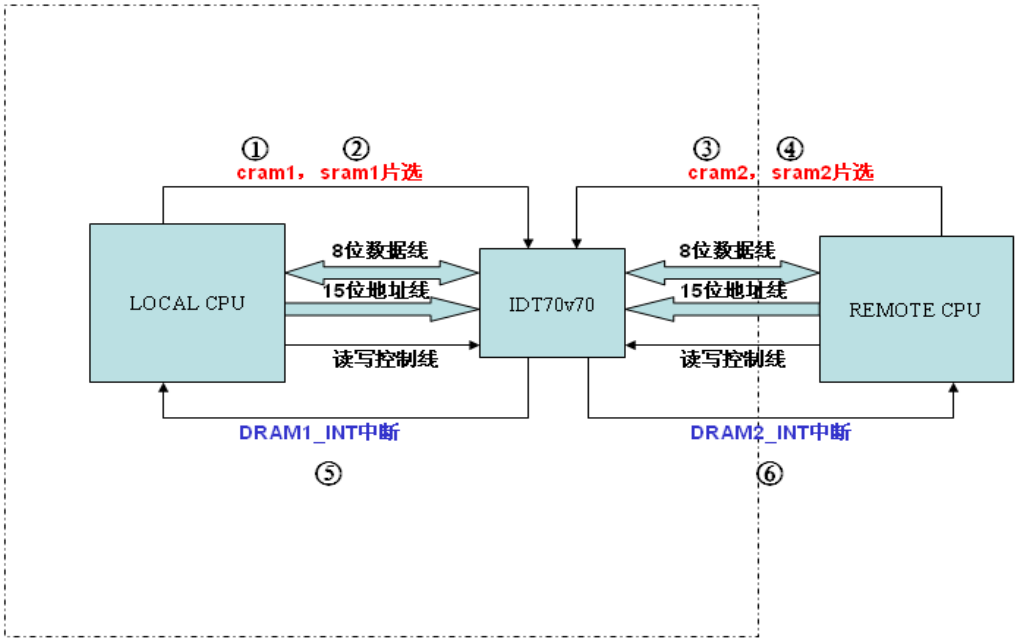


图3-5 IDT70v07系统连接示意图

对本地IDT70v07存储器的读写就是在74ls138译码器选通cram1的片选信号后对本地的IDT70v07的32k*8bit的存储空间进行读写。其中地址线、数据线、读写控制线信号来自本地CPU。对应功能①。

对本地 IDT70v07 旗语的读写就是在 74ls138 译码器选通 sram1 的片选信号后对本地的 IDT70v07 的 8bit（对应 8 个旗语标志）进行读写。其中地址线、数据线、读写控制线信号来自本地 CPU。对应功能②。

对远程 IDT70v07 存储器的读写就是在 74ls138 译码器上选通 cram2 后对远程的 IDT70v07 的 32k*8bit 的存储空间进行读写。其中地址线、数据线、读写控制线信号来自远程 CPU。对应功能③。

对远程 IDT70v07 旗语的读写就是在 74ls138 译码器选通 sram2 的片选信号后对远程的 IDT70v07 的 8bit（对应 8 个旗语标志）进行读写。其中地址线、数据线、读写控制线信号来自远程 CPU。对应功能④。

对 IDT70v07 传出中断的触发和清除是通过 0x7fff 的读写来实现的。对 0x7fff 的左端口进行写操作，实现触发 IDT70v07 的 DRAM2_INT，即将 DRAM2_INT 设置为低电平。对 0x7fff 的右端口进行读操作，实现清除 IDT70v07 的

DRAM2_INT，即将 DRAM1_INT 设置为高电平。可以这样理解 IDT70v07 的中断方式：0x7fff 看作是一个邮箱，本地 CPU 向邮箱里投信（即写 0x7fff 触发中断），远程 CPU 从邮箱里取信（即读 0x7fff 清除中断），而信的内容（0x7fff 中的内容）可以自己拟定。对另外一个中断可以同样予以分析。对应功能⑤。

对 IDT70v07 传入中断的触发和清除是通过 0x7ffe 的读写来实现的。对 0x7ffe 的右端口进行写操作，实现触发 IDT70v07 的 DRAM1_INT，即将 DRAM1_INT 设置为低电平。对 0x7ffe 的左端口进行读操作，实现清除 IDT70v07 的 DRAM1_INT，即将 DRAM1_INT 设置为高电平。对应功能⑥。

3.2.4 CAN 总线

CAN全称为控制器局域网络（Controller Area Network，CAN），是由德国Bosch公司为解决现代汽车中众多的控制和电子设备之间进行数据交换而开发的一种串行数据通信总线协议。CAN总线是一种多主总线，在当今自动控制领域占有重要的地位。CAN总线的特点是成本比较低、数据传输可靠性高、通信方式比较灵活以及抗干扰能力强。基于这些优点，CAN总线如今不单纯应用在汽车电子上，而是被广泛应用到网络化的控制系统中。

CAN总线拓扑结构如图3-6所示。

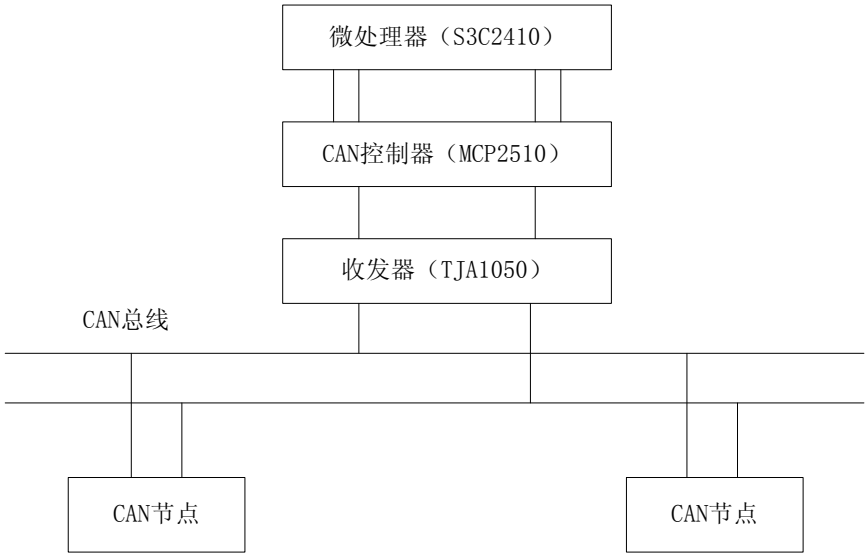


图3-6 CAN总线拓扑结构图

本平台基于嵌入式系统的 S3C2410 处理器，通过其 SPI 接口连接 CAN 控制

器 MCP2510 扩展 CAN 总线。MCP2510 硬件连接图和 SPI 硬件连接图如图 3-7，3-8 所示。

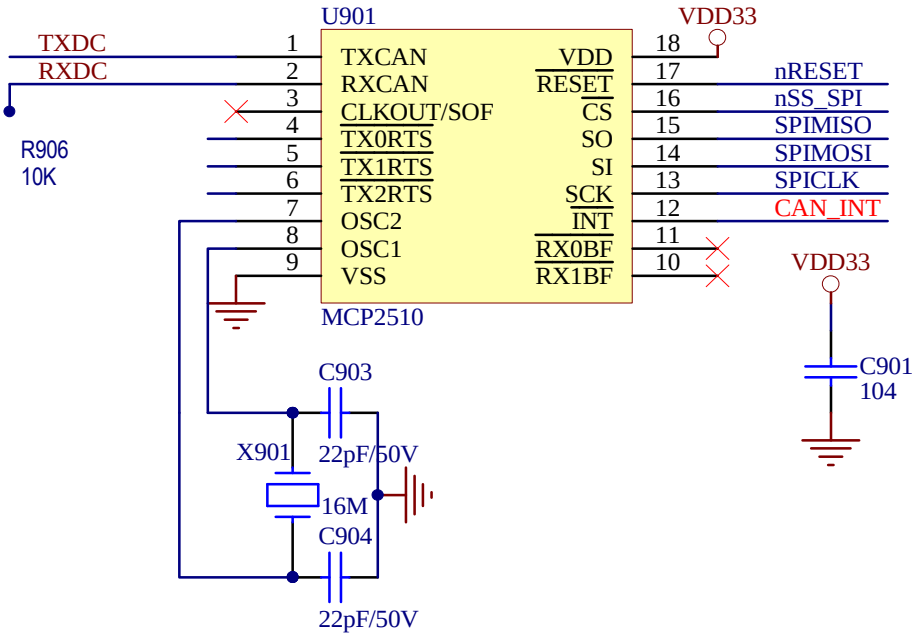


图3-7 MCP2510硬件连接图

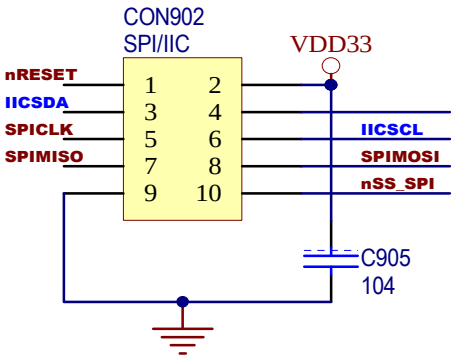


图3-8 SPI硬件连接图

3.3 TPM 硬件设计

整个平台的核心就是TPM芯片，本平台采用Atmel公司的AT97SC3204T。这款芯片遵循TPM1. 2规范，为嵌入式系统所专门设计。它能够在200ms完成2048位和40ms完成1024位RSA计算，20us完成64字节SHA-1计算，专门的生成机构可在400us内生成TCG密钥，其内部使用EEPROM来存储RSA密钥。AT97SC3204T通信方面兼容IIC的2线串行总线，有400kHz快速模式和100kHz标准模式，真随

机数发生器符合FIPS 140-2标准，1280字节的用户自定义存储空间(Flash)，电源采用3. 3v供电，28脚和40脚2种封装，工作在0-70℃温度范围，并且依赖于密钥的复杂度和大小，可以支持载入15到21个2048位RSA密钥。TPM芯片硬件连接如图3-9所示。

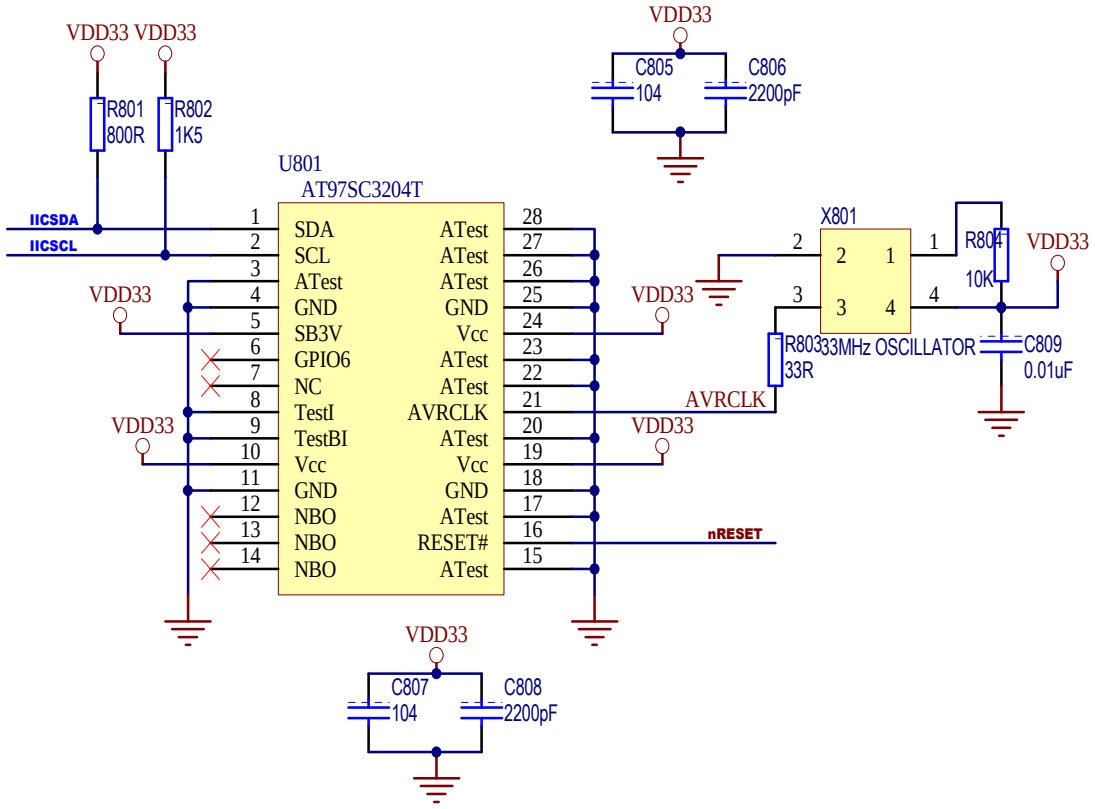


图3-9 TPM硬件连接图

在图3-9中，TPM有3个电源Vcc, 1个后备电源SB3V, 以及各自的GND。接3. 3v电源，去耦电容容值范围为2200pf-4700pf，应放在尽可能近的地方，小于5mm，连接相应的Vcc和GND。SB3V给内核供电，当Vcc掉电时，作为后备电源保持TPM的状态，如果不需要后备电源，可把它与Vcc直接相连。复位RESET#，低电平有效，最小复位脉冲宽度为2us，上电时，RESET#要保持为低，直到Vcc和AVRCLK稳定下来。时钟输入AVRCLK，33MHz的时钟用于驱动内部的AVR微处理器内核，时钟频率可以更低，范围是1-33M，节能后性能会线性降低，IIC也将不能有400k的性能。通用输入输出GPIO6内部带上拉电阻，缺省时为输入，不用时要浮空，这个引脚映射的NV索引号为TPM_NV_INDEX_GPIO_00，作为GPIO-Express-00使用。引脚SDA为IIC数据线，400khz工作时，上拉电阻应为

800Ω，如果小于400khz，需要进行测试以决定减少的电阻值。SCL为IIC的时钟引脚，400khz工作时，上拉电阻应为1. 5k，如果小于400khz，需要进行测试以决定减少的电阻值。 ATest、Testl、TestBl作为出厂测试用，直接或通过4. 7k电阻连GND或Vcc。NC、NBO引脚浮空。

3.4 可信嵌入式平台实物图

本平台所有硬件部分已经设计完毕并制作完成，其具体实物图如图3-10所示。

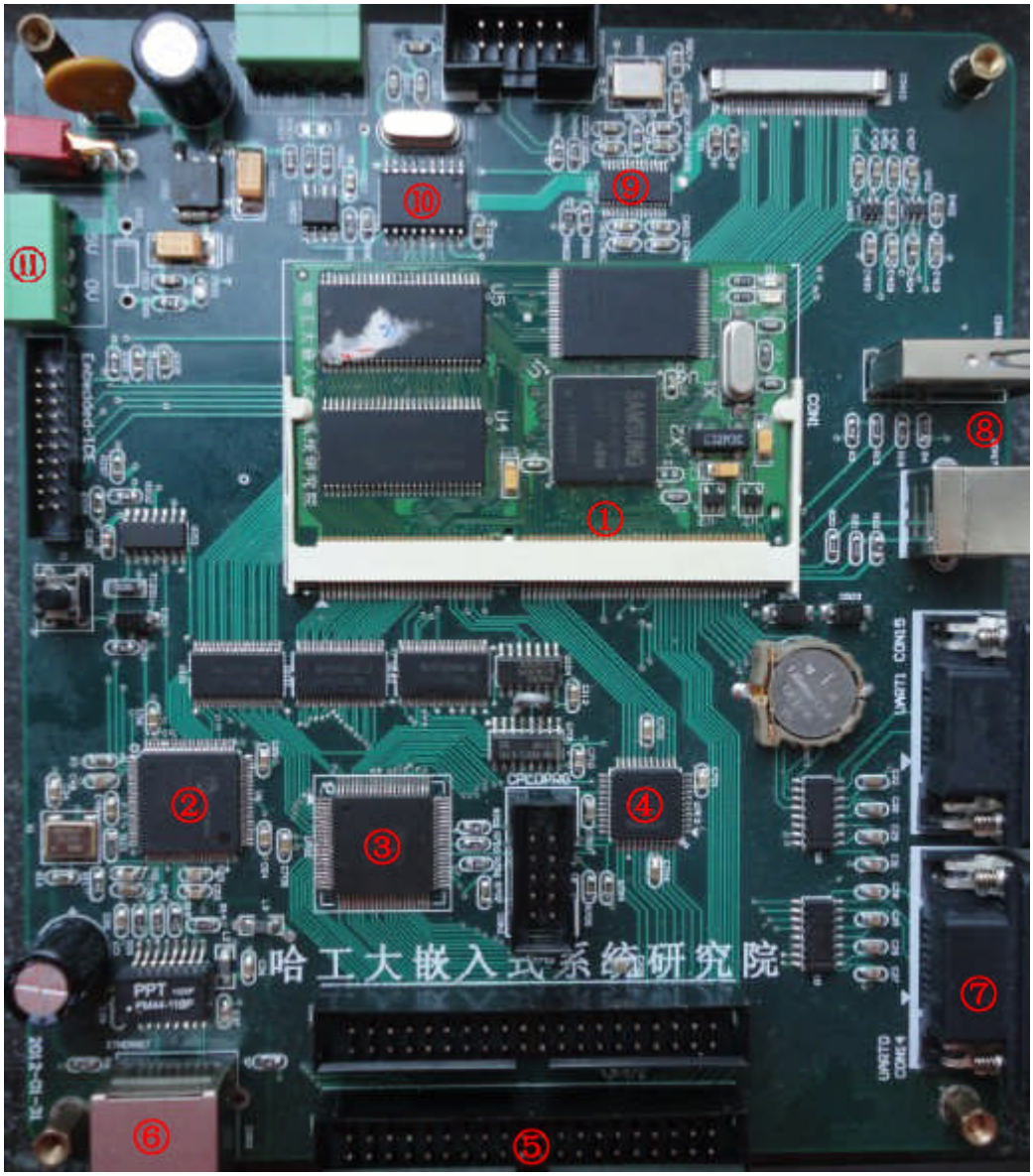


图3-10 可信嵌入式平台实物图

其中，①核心板，包括S3C2410、2片SDRAM、1片NAND FLASH。②网卡芯片DM9000。③双口RAM控制器IDT70v07。④ CPLD芯片EPM3032A。⑤LCD插槽。⑥RJ45接口。⑦串口。⑧USB Host与USB Slave。⑨TPM芯片AT97SC3204T。⑩CAN控制器MCP2510。

3.5 小结

本章完成了可信嵌入式平台的所有硬件设计，包括核心板和外围板上的功能模块。硬件电路开发板制作完成后，需要将引导程序、操作系统以及各个功能模块的驱动程序烧写到开发板上进行调试。在确保其他模块调试成功后将编写针对TPM的驱动程序。

第4章 可信嵌入式平台软件设计及移植

4.1 引言

可信嵌入式平台硬件部分设计完成后，还需要有相应的软件体系来支撑，这包括TPM底层的驱动程序、可信的引导程序、可信操作系统内核、支持TPM芯片并为应用程序提供API的TSS软件协议栈。软件部分设计完成后，将构成一个完整的可信嵌入式开发体系。

4.2 TPM 驱动程序编写与测试

4.2.1 IIC 总线简介

IIC总线是一种用于IIC器件之间连接的二线制总线，是由PHILIPS公司开发，用于连接微控制器、集成电路、功能模块及其他外围设备[16]。支持IIC的设备有单片机、ADC、LCD驱动器、键盘、存储器等等。本平台使用的TPM芯片支持IIC总线。S3C2410处理器内部IIC串行总线框图如图4-1所示。

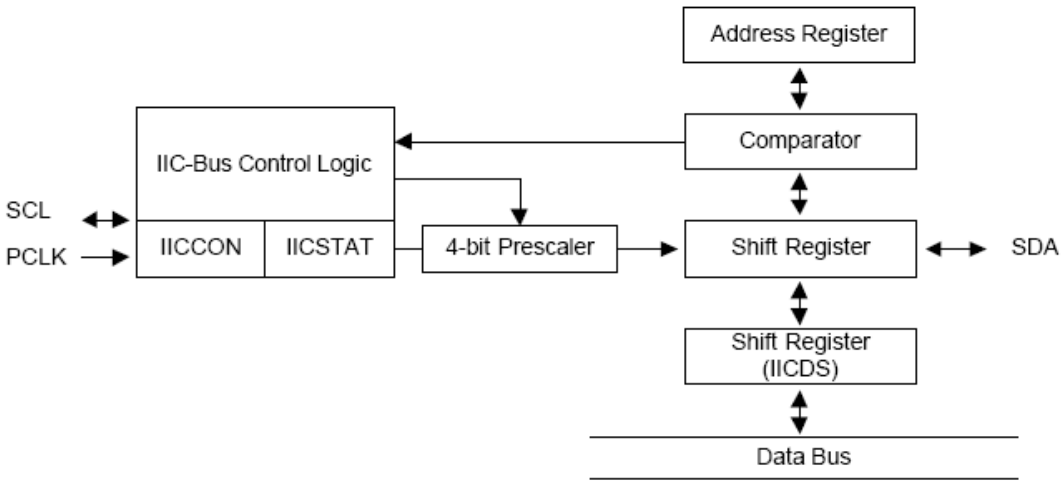


图4-1 三星S3C2410处理器IIC总线框图

在IIC总线上如果有1个设备的时钟线为低电平，则SCL线始终为低电平，只有所有设备的时钟线为高电平时，SCL为高电平[7]。IIC总线START与STOP执行

条件时序如图4-2所示。

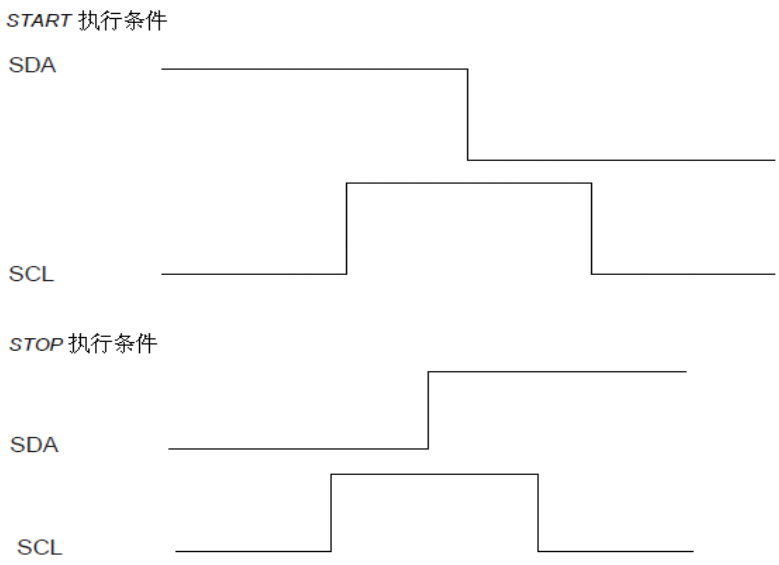


图4-2 START与STOP执行条件时序图

START的执行条件是SCL为高时，SDA由高变低，STOP的执行条件是SCL为高时，SDA由低变高[16]。在TPM空闲状态，接收到从主设备发出的START及随后的有效TPM从设备地址，TPM将进入一个已定义的内部状态序列，如TPM正处在接收命令或传送应答的过程中，主设备发出的STOP将中止这个过程，使TPM进入空闲状态。TPM与IIC总线电路连接如图4-3所示。

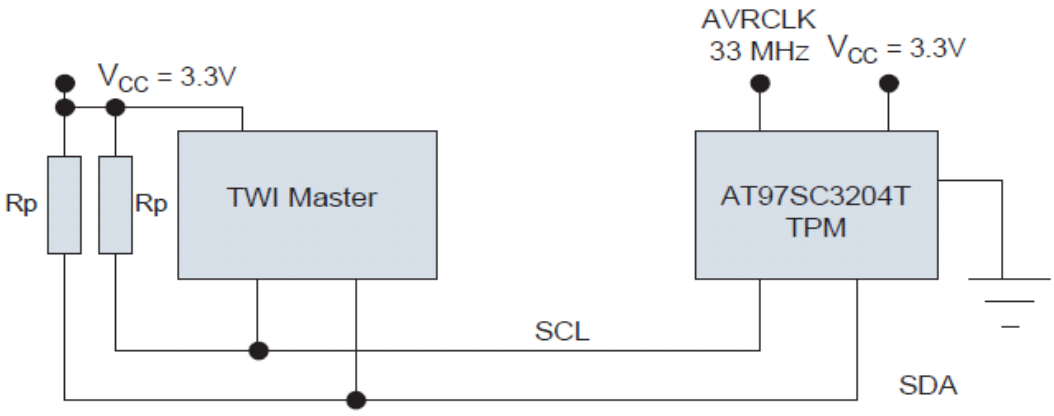


图4-3 TPM电路连接示意图

IIC 总线产生 START 信号后第一次发送的 8 位是选择从设备的地址，1-7 位

为从设备的地址，第 8 位为方向位[16]。当 START 信号开始后，IIC 总线上的各个设备将自己的地址与主设备送到总线上的地址进行匹配，如果一致，则从设备选定，然后由第 8 位判定是读还是写。

在 IIC 总线上可以传送多个字节数的数据，但是单次只能传送一个字节。每次传送完成的字节后跟一个认可位（第 9 位），即 ACK。每当设备传送完一个字节后，接着发出 SCL 线上的一个 ACK 认可位[16]。IIC 数据传送时序如图 4-4 所示。

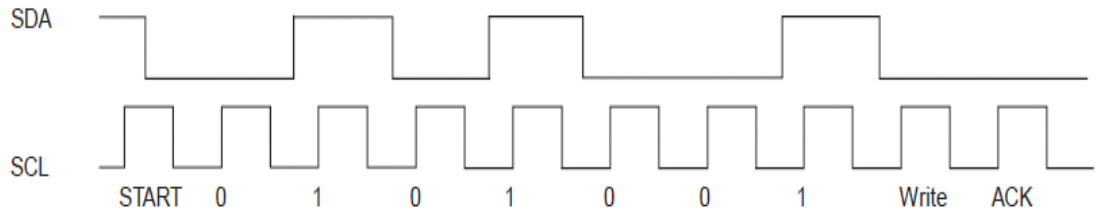


图4-4 IIC数据传送时序图

S3C2410内部有4个寄存器用于控制IIC总线数据传输，分别是IICCON、IICSTAT、IICADD、IICDS。其中IICCON为IIC总线控制寄存器。该寄存器信息及各位对应的定义如表4-1，表4-2所示。

表4-1 IIC总线控制寄存器

寄存器名称	地址	读/写状态	描述	复位值
IICCON	0x54000000	R/W	IIC总线控制寄存器	0000xxxx

表4-2 IIC总线控制寄存器各位定义

IICCON	位	描述	初始状态
ACK使能	[7]	0：禁止产生ACK信号；1：允许产生ACK信号	0
Tx时钟源选择	[6]	0：IICCLK=PCLK/16；1：IICCLK=PCLK/512	0
Tx/Rx中断使能	[5]	0：禁止Tx/Rx中断；1：使能Tx/Rx中断	0
中断清除标记	[4]	写0：清中断并开始写操作；读1：中断标志置位	0
发送时钟值	[3:0]	发送加载初始数据，决定频率	未定义

IICSTAT为IIC总线状态寄存器，用于配置IIC总线的各种状态等。该寄存器信息及各位的定义如表4-3，表4-4所示。

表4-3 IIC总线状态寄存器

寄存器名称	地址	读/写状态	描述	复位值
IICSTAT	0x54000004	R/W	IIC总线状态寄存器	00000000

表4-4 IIC总线状态寄存器各位定义

IICSTAT	位	描述	初始状态
模式选择	[7：6]	IIC总线主/从 Tx/Rx模式选择位 00：从接收模式；01：主接收模式； 10：从发送模式；11：主发送模式；	0
忙信号状态/启停条件	[5]	读0：IIC总线不忙；写0：产生IIC总线停止信号 读1：IIC总线忙； 写0：产生IIC总线启动信号	0
串行输出使能	[4]	0：禁止Tx/Rx传输；1：使能Tx/R传输	0
仲裁状态标记	[3]	0：总线仲裁成功；1：总线仲裁不成功	0
从设备状态标记	[2]	作为从设备时： 0：检测到启动或停止信号；1：接收到地址	0
零地址状态标记	[1]	作为从设备时： 0：检测到启动或停止信号；1：总线地址为0	0
接收到的最低数据位状态标记	[0]	0：接收到ACK应答信号；1：没有接收到ACK应答信号	0

IICADD为IIC总线地址寄存器。该寄存器及各位的定义如表4-5，表4-6所示。

表4-5 IIC总线地址寄存器

寄存器名称	地址	读/写状态	描述	复位值
IICADD	0x54000008	R/W	IIC总线地址寄存器	xxxxxxxx

表4-6 IIC总线地址寄存器

IICADD	位	描述	初始状态
从地址	[7：0]	从设备的设备地址和页面地址。位0为读写控制。 任何时候都可对该值进行读操作，当IICSTAT的 串行输出使能为0时，可对该位进行写操作	XXXXXXXX

IICDS为移位数据寄存器，主要用于产生移位数据。该寄存器及各位的定义如表4-7，表4-8所示。

表4-7 IIC总线地址寄存器

寄存器名称	地址	读/写状态	描述	复位值
IICDS	0x5400000C	R/W	IIC总线移位数据寄存器	xxxxxxxx

表4-8 IIC总线地址寄存器

IICDS	位	描述	初始状态
数据移位	[7：0]	IIC总线要移位发送/接收的数据。任何时候都可对该值进行读操作，当IICSTAT的串行输出使能为0时，可对该位进行写操作	xxxxxxxx

利用IIC对TPM进行读和写，通过驱动程序配置这四个寄存器来实现不同状态之间数据的传送。

4.2.2 TPM 驱动程序

TPM驱动程序主要功能是对应用程序开发者屏蔽硬件特性，接收上层应用的请求传递给TPM，并接收TPM的执行结果返回给上层应用，以及对TPM芯片进行其它管理等。采用ARM的IIC总线来扩展TPM芯片AT97SC3204T，由于AT97SC3204T采用与IIC兼容的TWI总线接口，因此驱动开发中可以利用IIC总线驱动来进行。TPM本身是一个字符型设备，通过填写file_operations等几个重要的数据结构可以像文件一样访问设备，应用程序使用标准系统调用打开、读取、写和关闭设备，对于调用者来说和设备看起来和普通文件一样。其中初始化、打开设备和关闭设备等操作与其他字符型设备相同，其他操作可以按照Linux字符型设备驱动程序来编写[20,24]，其读写操作实际是利用IIC进行读和写，通过配置IIC的几个寄存器，来实现TPM的读写操作。由于是利用IIC总线来扩展的TPM芯片，IIC中断在2410中属于内部中断，所以不需要单独的中断和片选信号， TPM驱动功能模块框图如图4-5所示。

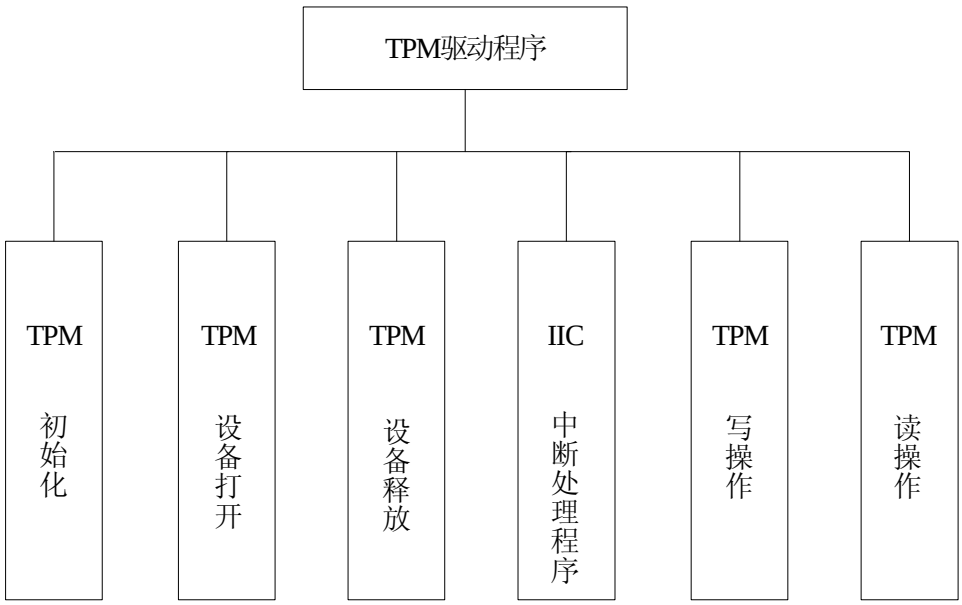


图4-5 TPM驱动程序功能模块框图

在图4-5中，TPM_init()完成TPM的初始化，主要是利用内核函数alloc_chrdev_region()动态分配主设备号，如果分配失败返回一个负值，在分配成功后，还需要在/dev下手动创建设备节点。接着对IIC进行初始化，查询S3C2410的芯片手册得知，2410有117个多功能的输入输出引脚，其中将端口E控制寄存器GPECON第31位和第29位置1，其余位置0，可实现IIC的SDA和SCL，即writel(readl(GPECON)|0xa0000000, GPECON)。然后，设置中断屏蔽寄存器INTMSK，使其不屏蔽IIC中断，配置IIC相关的四个寄存器，由于Linux不允许驱动程序直接操作硬件，否则会出现段错误，所以，要进行物理地址到虚拟地址的映射，Linux提供了一个内核函数ioremap()来实现这个映射。例如：

```
#define  S3C2410_IICCON   0x54000000//IIC  control
#define  S3C2410_IICSTAT  0x54000004 //IIC  status
#define  S3C2410_IICADD   0x54000008 //IIC  address
#define  S3C2410_IICDS    0x5400000c //IIC  data  shift
```

宏定义的是寄存器的物理地址，还需要映射成虚拟地址：

```
IICCON  =  ioremap(S3C2410_IICCON, 0x4);
```

```
IICSTAT = ioremap(S3C2410_IICSTAT, 0x4);
IICADD = ioremap(S3C2410_IICADD, 0x4);
IICDS = ioremap(S3C2410_IICDS, 0x4);
```

然后就可以利用writel()函数直接向寄存器中写入数据：

```
cmd=(1<<7) | (0<<6) | (1<<5) | (0xf);
writel(cmd, IICCON);
writel(0x10, IICADD);
writel(0x10, IICSTAT);
```

最后，利用内核函数request_irq（）将IIC中断服务程序注册到内核，如果返回0表示注册成功。

TPM_open()函数的功能是使用TPM前打开该设备，在使用完毕后TPM_release函数释放，与其他字符型设备一致。

IICHandle()为IIC中断处理程序，也是驱动程序的关键。发出START启动后，将中断源引脚寄存器SRCPND对应位置1，表示中断被触发，但此时有可能有多个中断被触发，由CPU中断仲裁后响应某个中断，则对应INTPND中的该位被置1，表示CPU正处理某个中断。然后，再根据响应中断之前的程序选择读数据还是写数据，响应中断完成后，还需要清除中断，回到原来程序断点继续执行。

TPM_write()是向TPM芯片写入数据，供上层应用来调用。TPM_write()实际上是调用IIC_write()，在上层应用中的数据通过IIC总线写入TPM。由于驱动程序工作在内核空间，应用程序工作在用户空间，所以，首先要在用户空间和内核空间分别申请一块大小相同缓冲区，然后利用copy_from_user()函数将指向用户空间的缓冲区指针传递给内核空间，并把用户空间缓冲区内的数据拷贝到内核空间。最后，调用IIC_write()逐次将内核缓冲区内的数据传送给TPM。有效TPM的7位从设备地址，被固定为0x29，接下来的第8位为读写位(读:1；写:0)，如果这个地址有效，第9位，SDA会被TPM拉低，这就是所谓的9位协议。先送0x52(写命令0x29+0)，ACK(或NACK)在第8个时钟的下降沿给出，在第9个时钟的高电平部分被识别。向TPM写入的数据是TCG规范的TPM命令，关于命令的格式在后续的章节会做详细说明，这里只说驱动程序的工作流程。TPM_write()工作流程图如图4-6所示。

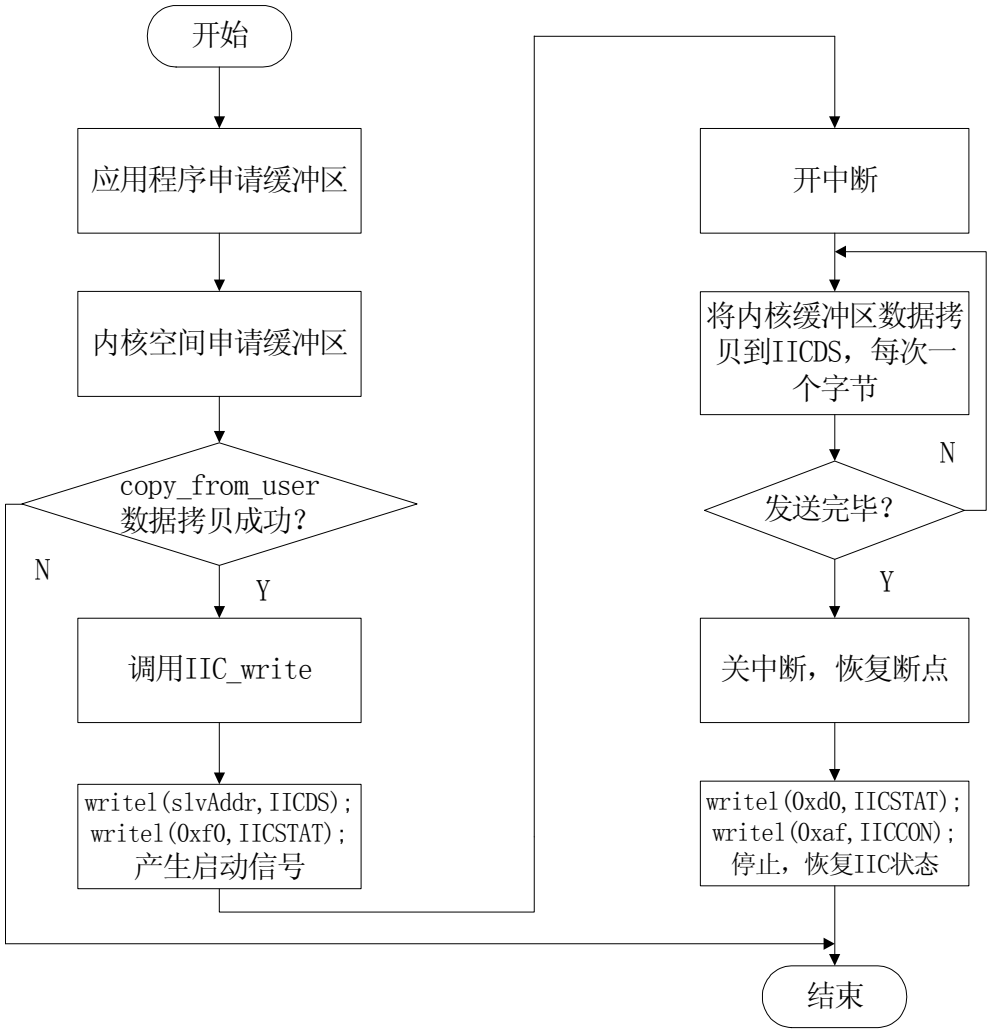


图4-6 TPM_write()工作流程图

同理，TPM_read()的工作流程与TPM_write()类似，TPM读是利用IIC总线从TPM设备读取数据，首先同样在内核空间和用户空间申请缓冲区，在产生启动信号触发中断，然后调用IIC_read从设备中逐次读取数据到内核缓冲区，读取完毕后，关中断，恢复程序断点，恢复IIC的寄存器状态，这些操作都与TPM_write()的中断处理一致。最后，再利用内核函数copy_to_user()将读取到的数据拷贝到用户缓冲区，返回给用户空间的应用程序。TPM_read()的工作流程图如图4-7所示。

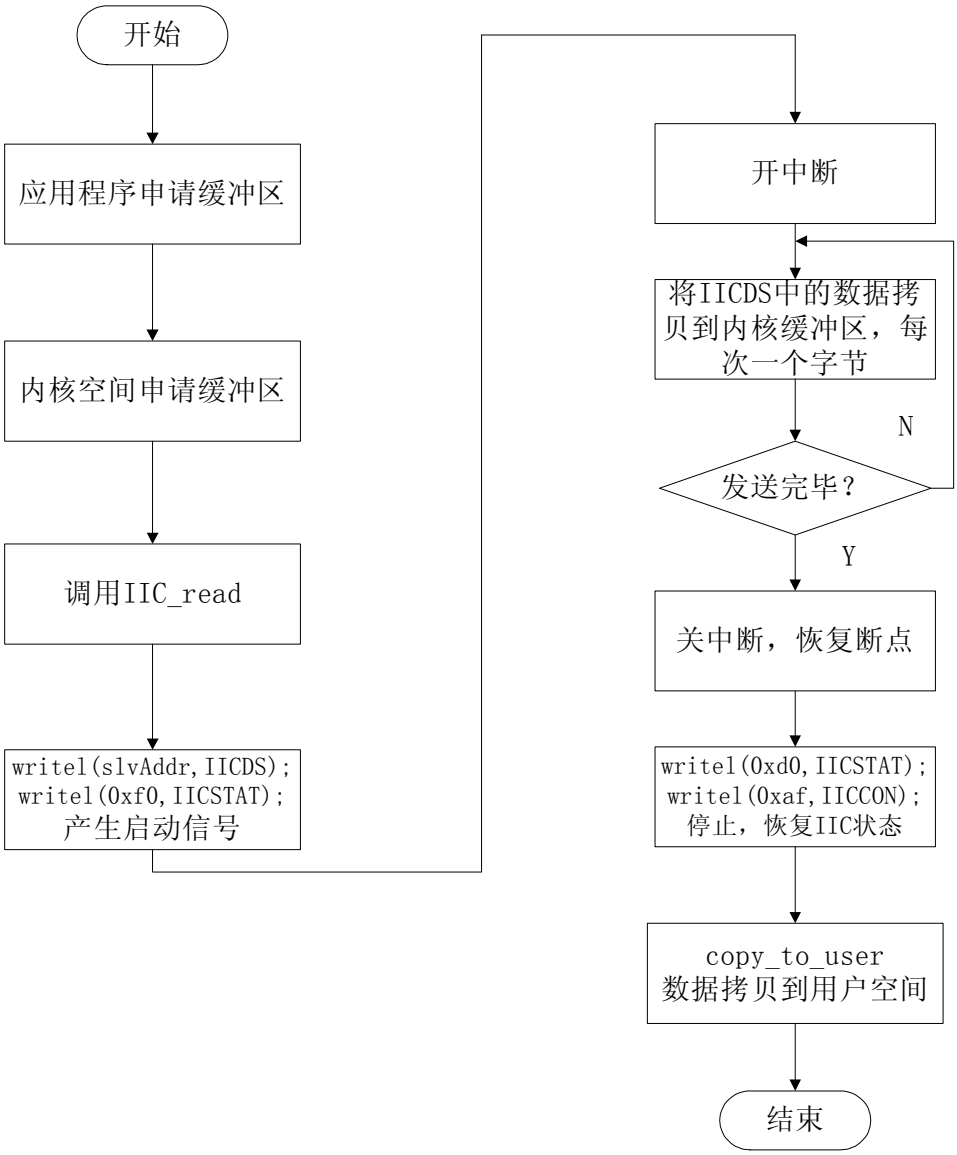


图4-7 TPM_read()工作流程图

至此，TPM驱动程序整体编写完成，然后加入：

```
module_exit(TPM_exit);
module_init(TPM_init);
MODULE_LICENSE("Dual BSD/GPL");
```

用作模块的装载与卸载，最后一行的意思是遵循GPL协议，否则编译时会产生警告信息。编写生成模块的makefile后，编译TPM驱动程序生成tpm. ko文件。配置

好minicom后，使用tftp将tpm. ko文件下载到平台上，如图4-8所示。

```
up-tech:~ #tftp -g 172.29.143.10 -r tpm.ko
up-tech:~ #ls
dram.ko          system           test_dram_write   xieqishun
dram3.ko         test_dram        test_eint5        zImage
eint5.ko         test_dram_double test_tpm
lost+found       test_dram_read   tpm.ko
```

图4-8 tftp传输tpm.ko

利用insmod命令将驱动模块加载到内核，然后用cat命令查看TPM已经加载到内核，主设备号是252[25]，如图4-9所示。

```
up-tech:~ #insmod tpm.ko
get device number
Success add TPM device!
up-tech:~ #cat /proc/devices
Character devices:
  1 mem
  2 pty
  3 ttyp
  4 /dev/vc/0
  4 tty
  4 ttyS
  5 /dev/tty
  5 /dev/console
  5 /dev/ptmx
  6 lp
  7 vcs
 10 misc
 13 input
 21 sg
 29 fb
 81 video4linux
 89 i2c
 90 mtd
 99 ppdev
128 ptm
136 pts
180 usb
189 usb_device
204 s3c2410_serial
252 TPM
253 usb_endpoint
254 rtc
```

图4-9 TPM加载到内核

利用mknod命令手动创建设备节点，其中c是表示这是一个字符型设备，252是主设备号，0为第一个设备，如图4-10所示。

```
up-tech:~ #mknod /dev/TPM c 252 0
up-tech:~ #ll /dev/TPM
crw-r--r--    1 root     root     252,   0 Jan  1 00:08 /dev/TPM
```

图4-10 创建设备节点

到此，TPM驱动已经编写完毕并以模块的形式加载到内核中。

4.2.3 TPM 命令传送与反馈

TCG组织规范定义了TPM命令协议，定义了一个10字节的命令协议，包括传送给TPM的命令和TPM的响应。10个字节中必须包含一个“paramSize”，说明输入命令和输出响应的数据字节数，paramSize+STOP或STOP定义输入输出序列的中断。如果TPM发出NACK，代表TPM忙，也能中断这个过程。如果主设备发出NACK，也能中断响应序列（如果不是最后一个字节的话）。响应序列完成，TPM进入空闲状态，等待下一个START+有效地址开始新的序列。AT97S3204T有1024个字节的输入缓冲区，还有100个字节缓冲区用于额外传输会话开销。10个字节的命令包含：2个字节的tag，说明命令的授权会话类型；4个字节的paramSize，包括tag和本身的全部输入字节数；4个字节的ordinal，TPM规范的命令原语。TPM命令的传送格式如图4-11所示。

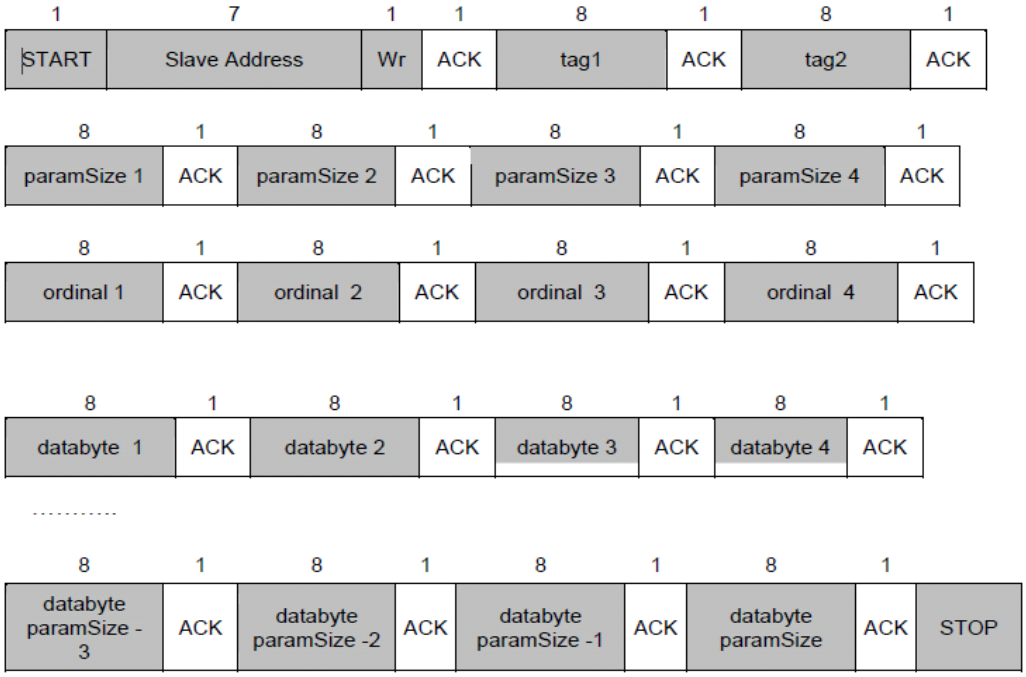


图4-11 TPM命令传送格式

收到TPM命令操作序列后，TPM将执行这些操作，因为许多操作的时间无法预计，此时TPM无法响应主设备。这时主设备使用TPM需要执行一个ACK查询程序判断TPM是否准备好，如果还在忙，则在第9位上响应NACK（逻辑0），否则响应ACK（逻辑1）。主设备对TPM进行判断的过程如图4-12所示。

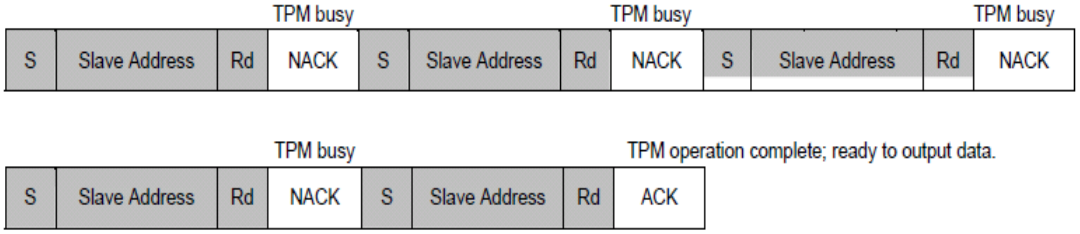


图4-12 ACK Polling

TPM在执行完命令操作后要返回命令数据，10个字节的数据命令包含：2个字节的tag，说明命令的授权会话类型；4个字节的paramSize，包括tag和本身的全部输出字节数；4个字节的returnCode，操作的返回码。传送的命令格式为：START+从设备地址+10字节序文+TPM响应数据。每个字节都需要主设备确认ACK（最后一个字节的ACK或NACK后接STOP，不是必须的），主设备也可通过NACK+STOP或NACK中断这个过程。除非复位或执行了另一个写命令，TPM的输出数据一直在输出缓冲区（1024个字节）中，主设备可重复多次读取。TPM返回命令的格式如图4-13所示。

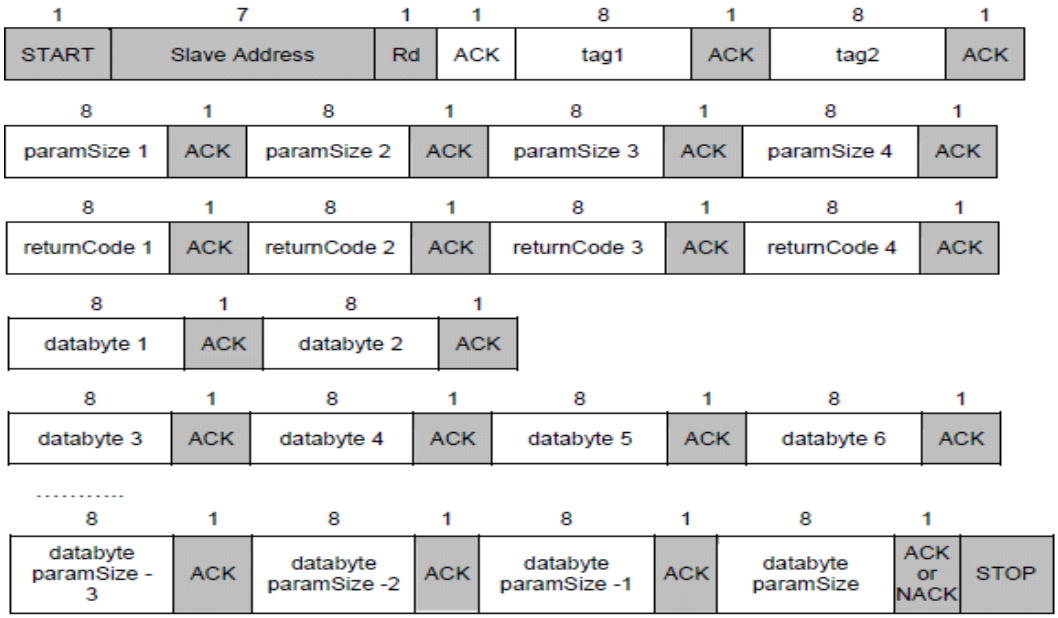


图4-13 TPM反馈命令格式

4.2.4 TPM 驱动测试

编写应用程序通过对TPM发送命令来实现对某一特定数据进行SHA-1签名，并通过程序输出结果验证TPM驱动是否成功运行。本文编写了一个上层应用程序实现了对某一数据实行散列算法并输出匹配结果，用到了TPM_ORD_GetRandom, TPM_ORD_SHA1Start, TPM_ORD_SHA1Complete三个命令，也可通过此程序验证TPM芯片的硬件连接及驱动是否正常工作。程序的流程图如图4-14所示。

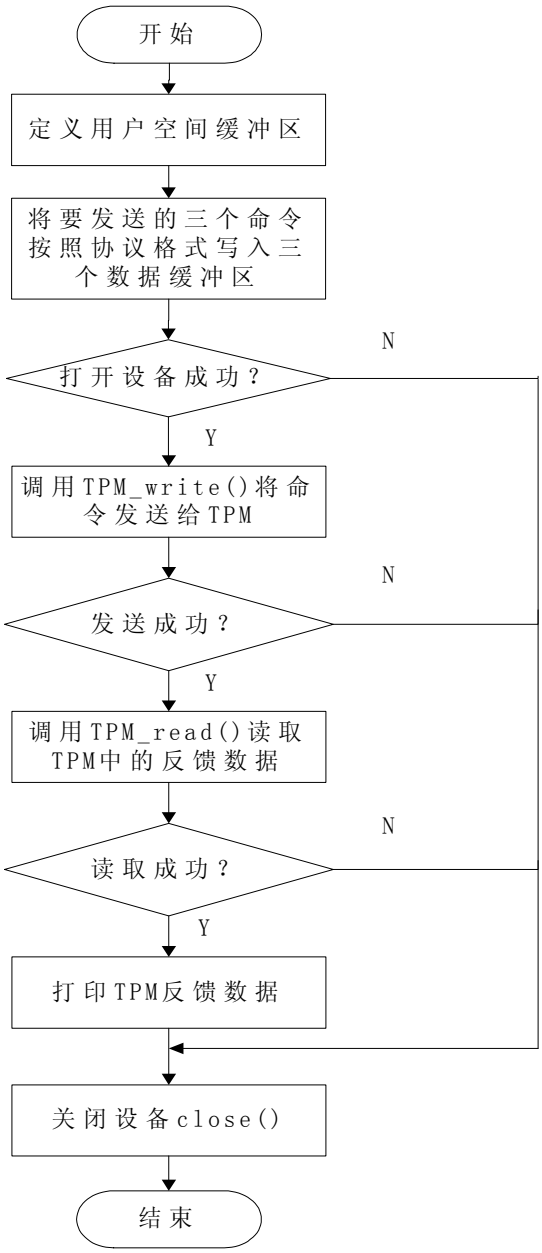


图4-14 SHA-1算法程序流程图

通过arm-linux-gcc进行编译，生成可执行文件通过TFTP传到开发平台上执行，执行结果如图4-15所示。

```
root@ubuntu:~# ./sha
sizeof(random_cmd):14
data in random_cmd:00c10000000e0000004600000008
ret of read random tpm0: 22
read tpm0 random data: 00 c4 00 00 00 16 00 00 00 00 00 00 00 08 b7 dc 29 56 2e
6b 5d 26
sizeof(tpm_sha1start): 10
data in tpm_sha1start: 00c10000000a000000a0
ret of read tpm0 after tpm_sha1start : 14
read tpm0 tpm_sha1start data: 00 c4 00 00 00 0e 00 00 00 00 00 00 08 00
sizeof(tpm_sha1complete): 142
data in tpm_sha1complete: 00c10000004e000000a2000000800102030405060708090a0b0c0d
0e0f101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f303132333435
363738393a3b3c3d3e3f404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d
5e5f606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f80
ret of read tpm0 after tpm_sha1complete : 30
read tpm0 data after tpm_sha1complete : 00 c4 00 00 00 1e 00 00 00 00 88 12 3d 8
d 81 28 b2 22 0f bc 68 51 84 10 81 92 d7 8f e9 52
```

图4-15 SHA-1算法执行结果

SHA-1算法称为安全HASH（或散列）算法，对于最大长度为2^{64}位的数据，SHA-1会生成一个20字节的基于此数据的摘要。通过对这个消息摘要进行比较，可以用来验证原始数据的完整性。SHA-1有如下特性：不能从消息摘要中复原信息；发生散列冲突的概率极低。SHA-1算法是完全单向性、抗碰撞性的安全HASH函数算法，所以能够保证TPM设备所度量数据的完整性，从而提高整个系统的安全性。

在嵌入式系统的实际应用中，可用SHA-1算法对嵌入式软件、文件或者某一段数据进行哈希运算做散列值，并由TPM进行签名，将散列值和签名保存到事先定义好的缓冲区或者寄存器中并与之前软件厂家或是自己在安全环境下形成的散列值和签名进行比较，结果相同说明软件或者数据未被篡改。

4.3 U-boot 的可信改造与移植

4.3.1 U-boot 工作过程

在核心基础软件方面，要努力裁剪出最小软件系统环境，确保该环境是完全自主可控的，在满足嵌入式系统性能要求下做到可信可控。

Bootloader 是衔接底层硬件平台与上层系统和应用软件的固件，可以对嵌入

式 CPU 和外围辅助硬件进行必要的初始化，建立基础运行环境，完成系统启动和映像文件的加载工作[23,27]。U-Boot、dBUG 和 Vivi 是常见的 Bootloader，但这些均没有较好的对可信与安全性的支持。因此在可信嵌入式开发平台的软件子系统构建中，首要工作是以嵌入式 Bootloader 即 U-boot 为基础，对其进行可信与安全性上的基础加固，保留传统 U-boot 可以屏蔽下层硬件平台的差异，方便上层软件开发与移植的同时，引入可信机制与安全措施，建立系统软件底层的第一层安全屏障，有效抗击对 U-boot 的非常攻击和篡改。在具体可信机制方面，建立并实现以信任链为主线的完整性度量和授权启动机制，确保可信启动。

U-boot 一般分为 2 部分 Stage1 和 Stage2。Stage1 部分由于靠近硬件常用汇编部分实现，用以执行简单的硬件初始化与配置，包括设置嵌入式 CPU 中若干控制寄存器、设置内存和堆栈、变量的初始化及执行方式等。Stage2 部分用 C 语言部分实现，负责复制内核映像数据和根文件系统映像数据到内存、设置启动参数、串口通信等功能。

U-boot 对 TPM 的支持主要包括三部分：一是在实现 U-boot 自身基本功能的同时，对 ETPM 设备进行必要的管控；二是对可信平台的身份认证以及授权控制；三是增加 U-boot 代码中对某些硬件的度量，建立信任链过程，后两部分也叫做 U-boot 的可信改造过程。

CRTM（Core Root of Trust Measurement）是可信度量根，是信任链的起点，是系统最先被执行的部分。在 PC 中 CRTM 通常与 BIOS 分开实现，而在嵌入式环境下，CRTM 在实现方式上，可采用与 U-boot 作为一个整体来实现。嵌入式系统中信任链的建立过程如图 4-16 所示。

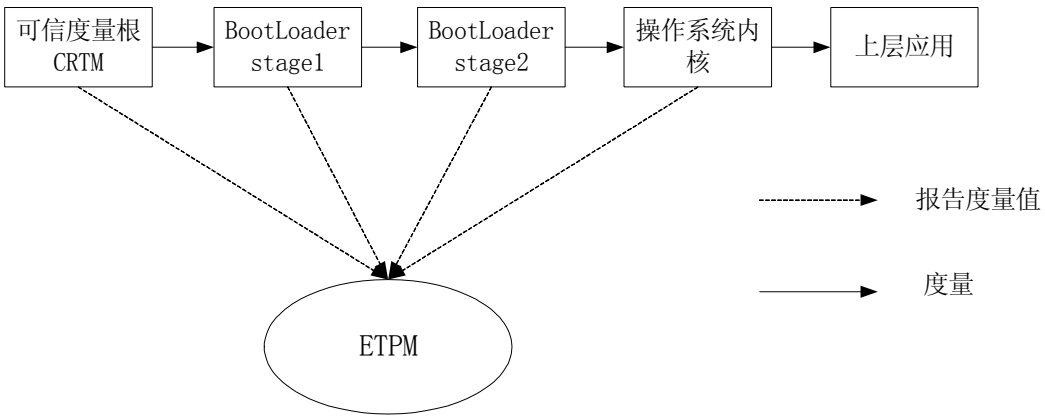


图 4-16 嵌入式系统信任链建立过程

4.3.2 U-boot 可信改造过程

首先介绍PCR平台配置寄存器。PCR是TPM上的存储区域。当系统第一次开机启动时，PCR中的值为0。然后，不同的度量操作会扩充PCR中的值，即将一个值与PCR中的值进行SHA-1散列运算，得出一个新值来替换当前PCR中的值，并且，这些值会被日志文件所记录。由于SHA-1散列运算本质上是不可逆的，所以利用PCR中的值来检验要计算的值是否被更改。一个TPM中至少要有16个PCR寄存器。

TPM驱动程序是在嵌入式操作系统被加载后才能与TPM进行通信的，也就是说，U-boot在引导操作系统的过程中，是没有任何驱动程序可用的。在这个阶段，为了要与TPM芯片进行通信，可以利用TCG给引导程序规定的一个最小中断集合，由TPM芯片厂商提供的一个简单接口。可信启动的度量流程图如图4-17所示。

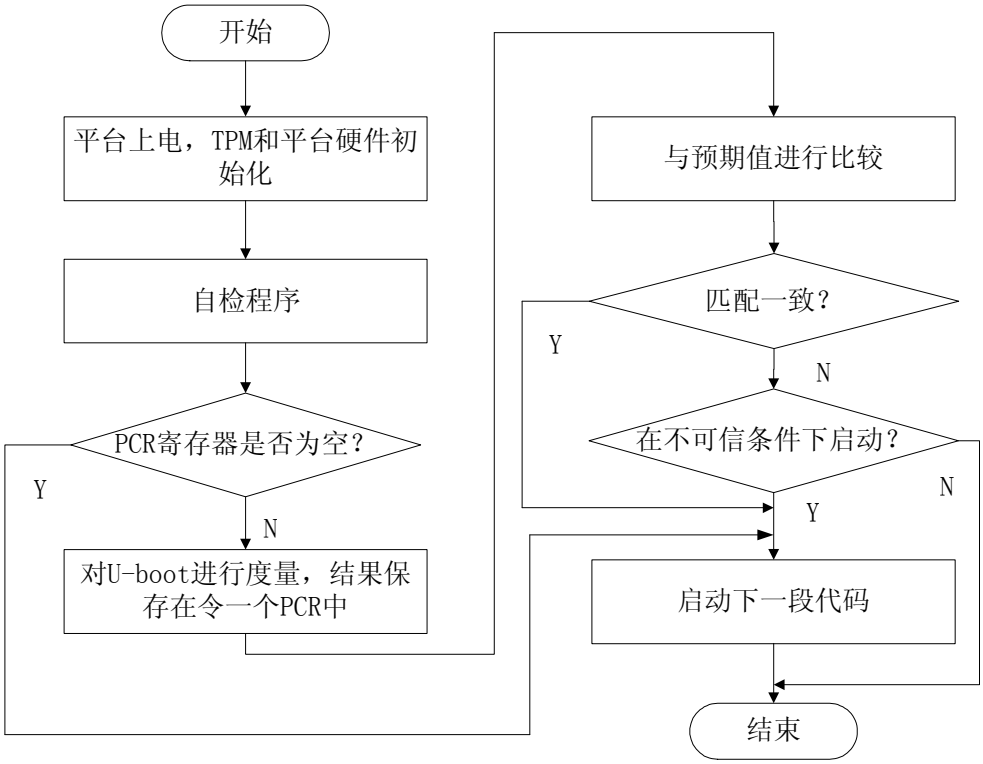


图4-17 可信启动流程图

在可信启动的初始阶段，所有信任都是在引导程序中一块不可变的可信代码开始的，在将控制权传递给下一部分前，首先要用SHA-1散列运算度量下一段代

码，并将结果扩展到某一PCR寄存器中，并与预期值匹配，如果与预期值相同，则向下执行。这样，每次在新代码要执行之前都会被上一部分度量，这样就建立起了一个信任链。如果某一部分代码被破坏，下次启动的时候，这段代码被度量的值就会与预期值比较，由于SHA-1算法本身的强壮性，恶意代码无法算出PCR中的预期值。

4.3.3 U-boot 可信改造实现

U-boot 是从 start. s 文件中开始执行的，这个文件位于/cpu/arm920t 目录下。开始的代码是处理器的异常处理向量表，接着是系统复位，然后是关闭看门狗，屏蔽中断，配置内部时钟频率，这些完成后，是代码重定向，将 U-boot 自身由 NAND FLASH 拷贝到 SRAM 中。本次改造是在该操作之前开始进行 U-boot 的度量[35]。用汇编语言将 IIC 的启动，停止，读写，应答等定义成子程序，然后在发送和接受命令时 call 调用。例如向 TPM 写入命令的部分代码为：

```
IIC_write:
LD          R14, #008h
AND         P2CON, #07fh
OR          P2CON, #040h              ;SDA 输出
IIC_writeLoop:
TM    R15, #080h
JR          Z,  IIC_write _1
OR          P2, #SDA    ;SDA = 1
JR          IIC_write _2
IIC_write _1:
AND         P2, #~SDA    ;SDA = 0
NOP
IIC_write _2:
OR          P1, #SCL    ;SCL = 1
NOP
AND         P1, #~SCL    ;SCL = 0
RL          R15
DEC         R14
```

```
JR          NZ, IIC_write Loop
CALL        GetAckIIC
JR          NC, IIC_writeError
RET
IIC_write Error:
CALL      IIC_stop
RET
```

TCG 规范了供引导程序中使用的 TPM 命令，向 TPM 发送 TPM_Startup 命令，TPM 收到命令后对 PCR 寄存器进行复位并启动 TPM，然后就可以向 TPM 发送散列运算命令来计算 PCR 值。在无内存时 TCG 规范了 4 个命令：TPM_SHA1Start, 帮助引导程序计算 SHA；TPM_SHA1Updata 当散列对象很大时，继续 SHA；TPM_SHA1Complete 完成 SHA；TPM_SHA1CompleteExtend 完成 SHA-1 并将值扩展到 PCR 中。通过对存储在指定地址的 U-boot 进行哈希计算后得到的返回值判定出 U-boot 代码是否已经被篡改，并且询问用户是否进行下一步操作。

这些工作完成后，将 U-boot 重新编译，生成镜像文件。利用 sjf2410. exe 在 XP 系统下的超级终端将 U-boot 烧写到平台开发板上。烧写成功后，为烧写操作系统内核做准备。为了方便使用，可以将 TPM 驱动编译进内核。执行 make menuconfig 命令，在内核配置文件中选中如下参数即可将 TPM 驱动编入内核：

```
Device Driver->
        Character Device->
                [*] TPM Hardware Support->
[*] TPM Interface Specification 1.2 Interface
```

然后执行 make 命令，会在内核/arch/arm/boot 下生成内核文件 zImage。使用 U-boot 生成工具 mkimage 生成 uImage 文件。uboot 源代码的/tools 目录下有 mkimage 工具(mkimage. c)，这个工具可以用来制作不压缩或者压缩的多种可启动映象文件。安装成功后执行./make_uImage，如图 4-18 所示。

```
root@ubuntu:~/kernel# ./make_uImage
Image Name:   Linux-2.6.24.4
Created:      Fri Mar 30 09:36:22 2012
Image Type:   ARM Linux Kernel Image (uncompressed)
Data Size:    1820852 Bytes = 1778.18 kB = 1.74 MB
Load Address: 0x30008000
Entry Point:  0x30008040
```

图4-18 制作uImage镜像

此时会在内核源代码根目录下生成 uImage 内核文件，如图 4-19 所示。

```
root@ubuntu:~/kernel# ls
2008-10-12.config  Kbuild                     REPORTING-BUGS
arch               kernel                     samples
block              lib                        scripts
COPYING            lyj_uptech_20081216.config security
CREDITS            MAINTAINERS                sound
crypto             Makefile                   System.map
Documentation      make_uImage                uImage
drivers            mkimage                    uptech2410cl-20100325.config
fs                 mm                         usb_gadget.patch
include            Module.symvers             usr
init               net                        vmlinux
ipc                README                     vmlinux.o
```

图4-19 生成uImage

烧写内核，拷贝生成的 uImage 文件到 TFTP 下载目录/tftpboot，然后进入 U-boot 控制台设置主机和平台板子上的 IP，然后用命令 tftp 0x30008000 uImage 固化到 NANDFLASH。擦除 NANDFLASH 空间 0x80000 和 0x200000，然后写入到这个地址。使用 bootm 引导内核，启动成功。这样，整个平台的基础软件部分的开发工作就完成了。

本文的改造只是提出一种可行的方法并简单的加以实现，与经过严格多平台测试的版本还有很大差距。引导程序是个大工程，需要后续严谨细致的开发调试。

4.4 TSS 软件栈在嵌入式操作系统中的移植

TSS 是 TCG 软件栈，是与 TPM 相配合的软件协议栈，为终端应用提供统一使用 TPM 的接口，处于 TPM 和上层应用之间，是操控 TPM 的中间接口层。TSS 以与厂商独立的形式为上层应用提供管理和使用 TPM 的所有功能。

按照由上到下的层次，TSS由TSP(TCG服务提供者)、TCS(TCG核心服务)和TDDL(TCG设备驱动程序库)构成。TDDL位于TPM驱动之上，用以屏蔽不同TPM驱动差异而设置的，在嵌入式开发中，由于其特殊专用性和资源节约要求，可以不设置TDDL。

目前比较流行的基于TSS规范的实例是由IBM研发的Trousers，其遵循TPM规范。Trousers软件包主要包括两大功能部分:TPM Tools部分和TPM PKCS#11命令，其中TPM Tools部分是一套用来管理TPM的程序。Trousers可完成与TPM建立会话、密码计算与密钥管理、数据存储与维护等功能。Trousers功能模块框图如图4-21所示

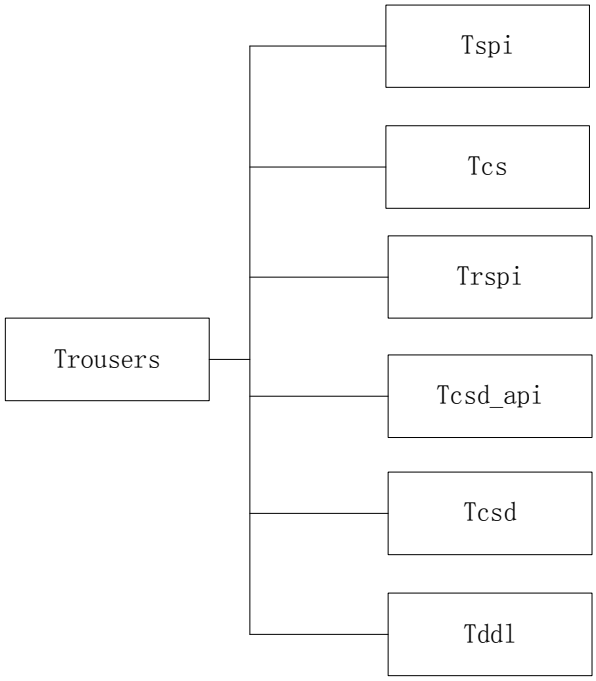


图4-21 Trousers功能模块框图

Tspi模块位于tsp层，主要为上层应用提供调用接口。Tcs模块位于tcs层，主要为调用Tcs层功能提供接口函数；Tcsd以daemon后台守护进程形式存在，侦听来自上层应用的访问请求，据此来调用相应模块执行，而Tcsd_api模块则为Tcsd后台守护进程提供有关函数接口。有关密码运算功能部分由Trspi模块中的crypt子模块调用OpenSSL库方式来提供实现。

在Trousers的移植与改造中要保留其基本功能与关键功能，对于与可信嵌入式开发平台要求相关性较低部分进行简化，确保满足本平台要求。本平台所选用

的 Trousers 版本为 0. 3. 7，首先的工作是交叉编译 OpenSSL 和 Trousers。具体做法是执行. /configure 命令时加入 arm-linux-gcc，并修改 makefile 里的编译器为 arm-linux-gcc，这样交叉编译后的 Trousers 能在 ARM 平台下执行。注意，交叉编译时 TSS 要指定 OpenSSL 的路径。编译成功后，就可以将编译好的 TSS 用文件系统制作工具 mkcramfs 生成镜像文件，并通过 tftp 拷贝到 ARM 开发平台上。启动 TSS 时，显示出 TCSD trousers 0.3.7:TCSD up and running，则移植 TSS 成功。

4.5 调用嵌入式 TSS 软件协议栈

由4. 4得知，对嵌入式应用程序来说，TDDL设备驱动程序库即本文所编写的TPM驱动程序。驱动程序提供了打开和关闭设备、读写数据、查询设备属性等小的API集合，提供了使用TPM最基本的功能。TCS层的功能是管理TPM的资源，授权会话和密钥上下文的交换，并提供一个TPM命令数据块发送器，能够把TCS提供的API转换成TPM能识别的字节流，还提供一个全局密钥存储设备，同步TSP层的应用数据访问。而TSP接口提供了TPM的所有功能，通过动态链接库或者共享对象的方式供应用程序调用[13]。本节通过一个应用程序调用TSS来创建一个密钥层次结构来使用TPM密钥。程序的流程图如图4-22所示。

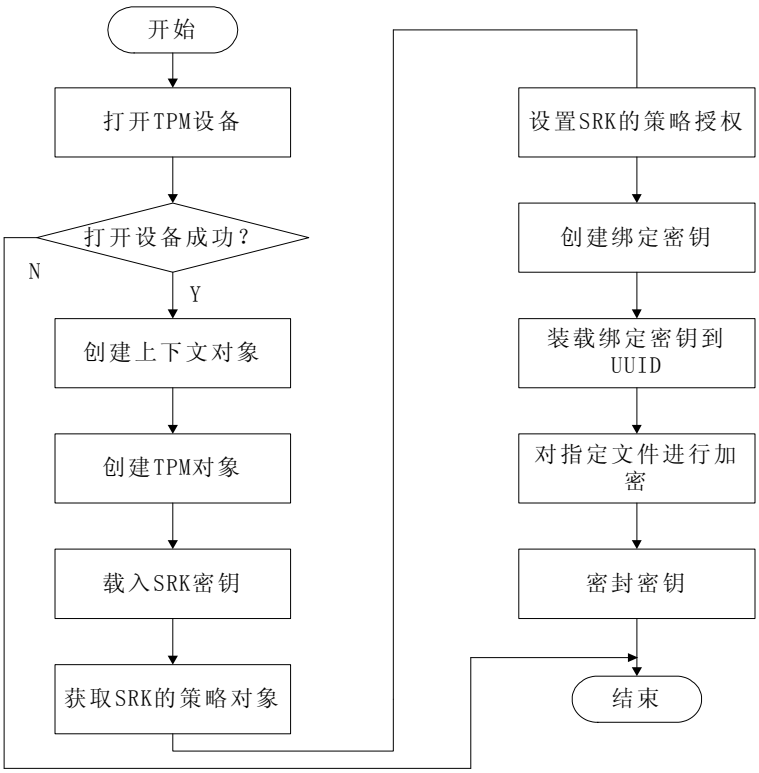


图4-22 创建TPM密钥

本程序调用了TSS目录下的头文件，通过Tspi_Context_Create()函数首先创建上下文对象，Tspi_Context_GetTpmObject()函数创建TPM对象，Tspi_Context_LoadKeyByUUID()函数载入SRK密钥，Tspi_GetPolicyObject()函数获取SRK的策略对象，Tspi_Policy_SetSecret()函数设置SRK的策略授权，Tspi_Context_CreateObject()创建绑定密钥，Tspi_Key_CreateKey()产生密钥，Tspi_Context_RegisterKey()装载密钥到UUID，Tspi_SealFile()文件加密。程序运行结果如图4-23所示

```
root@ubuntu:~# ./tss
创建上下文对象......
创建TPM对象......
载入SRK密钥......
获取SRK的密钥对象......
设置SRK的策略授权......
创建绑定密钥......
在TPM产生密钥前，设置填充类型......
产生密钥，该密钥不含PCR绑定
TSS Authentication Dialog
Enter PIN:
装载绑定密钥到UUID......
```

图4-23 使用TPM密钥程序执行结果

通过图4-23的运行结果可以得知TSS协议栈已经移植成功，至此所有软硬件已经设计完毕，形成了一个完整的可信软硬件平台。

4.6 小结

本章首先阐述了TPM的使用方法，即通过向TPM发送命令来实现各种功能，4.2.3的应用程序通过调用TPM驱动程序直接向TPM发送命令，通过程序执行结果得知TPM设备可以正常使用。4.5编写了一个应用程序调用TSS提供的API来实现TPM创建一个绑定密钥并对文件加密，这个程序验证了TSS软件协议栈向嵌入式操作系统中移植成功。

第5章 可信嵌入式平台功能实现与性能测试

5.1 引言

结合TCG的设计规范和TPM芯片AT97S3204T的功能，可信嵌入式平台的主要设计目标有两个：第一是提供一种可信的方法来度量和报告嵌入式平台的软硬件环境，即当前平台的状态，通过构建出一条完整的信任链来实现；第二是利用TPM芯片能够实现SHA-1算法和RSA算法等功能，对数据进行摘要、加密、安全签名、身份认证等。在无TPM芯片的嵌入式环境中，信息交互是通过明文或者经过软件加密后传输的，加密密钥存储在文件系统中，这种方法的安全隐患非常大，通过木马程序就可以截取到信息，即使信息不是明文，也可以通过攻击找出密钥与口令，再经过破解等手段就可以获得原始信息。通过软件的方法保护加密密钥不被完全泄露是不可能的。而在基于TPM的嵌入式环境中，加密密钥由TPM产生并保存在硬件中，签名在外部无法访问的硬件区域进行存储。本章实现了基于TPM的安全身份认证功能，通过TPM产生不可迁移密钥，在用户和服务器之间进行身份认证。最后，利用嵌入式性能分析工具来测试程序的性能。本章实例可适用于电子支付、应用程序登录、手机邮箱等应用的身份认证模块。

5.2 基于 TPM 的安全身份认证

5.2.1 传统的身份认证

在信息交互过程中确认交互双方身份是否合法的过程就是身份验证，这是当前一切信息交互的前提。例如QQ、人人网等通讯工具的用户登录，应用软件的CD-KEY, 网银、淘宝的电子支付都要用到身份认证。

在无TPM的环境中，进行身份认证的方法有静态密码、动态口令卡、USB KEY、手机短信、智能IC卡以及生物信息等。其中，静态密码属于利用你知道什么来判别。动态口令卡、USB KEY、手机短信、智能IC卡等属于你拥有什么来判别。生物信息如声音、指纹、视网膜识别等属于合法身份者特有的信息。传统信息交互中常用身份认证方法优缺点对比如表5-1所示。

表5-1 常用身份认证方法优缺点对比

身份认证方法	优点	缺点	适用领域
静态密码	简单，易于实现	容易被截获，安全性低	网络登录等
动态口令卡	基于硬件，安全性高	携带不方便，易丢失	网上银行等
USB KEY	软硬结合，安全性高	外围设备，不支持2次开发	网银、电子商务
手机短信	安全，普遍，易维护等	每次需收费，需要手机支持	网银、电子商务
智能IC卡	不可复制	传输过程易被监听	专用设备
生物技术	安全性比较高，且唯一	实现复杂，技术含量要求较高	门禁系统等

5.2.2 基于 TPM 的安全身份认证实现

基于TPM的嵌入式环境能以更安全的方式提供加密、身份认证等服务，因为TPM从硬件的最底层开始提供可以信任的基础[44]。利用可信嵌入式开发平台模拟一个嵌入式终端。嵌入式环境下基于TPM的安全身份认证实现过程如图5-1所示。

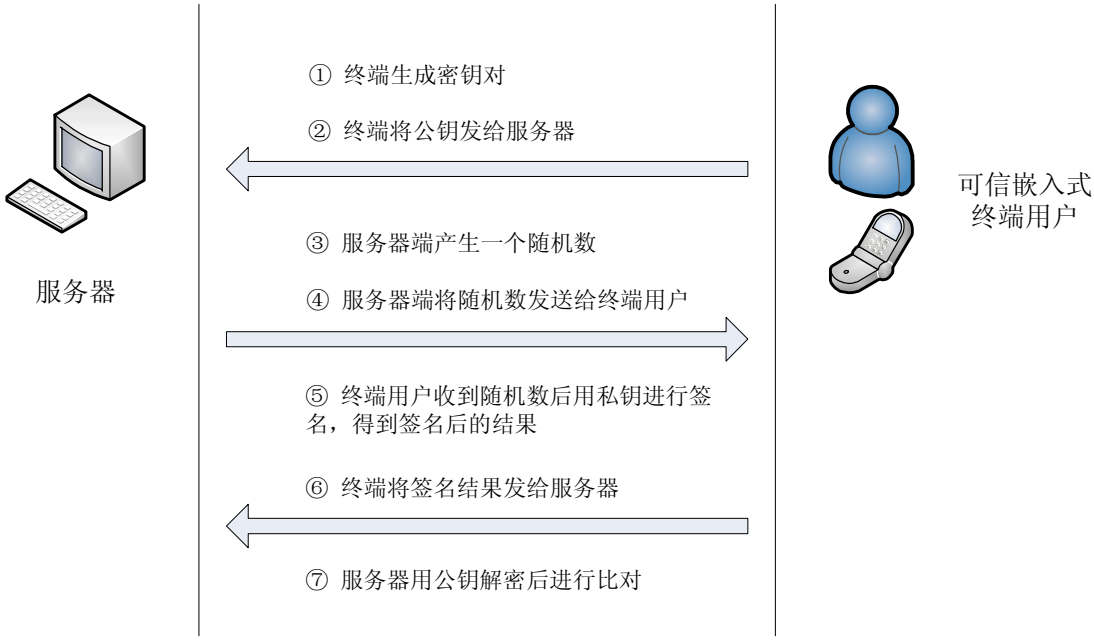


图5-1 基于TPM的身份认证执行过程

首先，嵌入式终端和服务器之间首先建立UDP协议。具体实现为在服务器端使用socket()创建一个套接字，初始化结构体sockaddr_in，其中端口号为19，IP地址为：172.39.143.25。使用bind()将IP地址和端口号与socket绑定，如果错误则

退出。成功后，调用函数listen()将该套接字转化为监听状态。此时，程序进入循环等待状态，等待终端发送数据。终端实现过程与服务器端基本一致，只是发送数据前要先进行与服务器的连接。虽然UDP协议是无连接的，不需要TCP的三次握手协议，但是connect()能帮助服务器端返回错误信息，防止进入死循环等待。调用socket.h文件中的sendto()和recvfrom()来发送和接受数据。

① 终端应用程序利用 TPM 和 TSS 提供的 Tspi_Key_CreatKey 生成 RSA 密钥对，其中公钥为 N，私钥为 P。

② 终端利用 sendto()将公钥 N 发送给服务器端。

③ 服务器端利用 stdlib.h 头文件提供的 srand()函数产生一个随机数序列，由于 rand()生成的随机数是伪随机数，每次执行程序所产生的序列是重复的。使用当前系统的时钟值作为随机数种子，语句为 srand(time(NULL))，可以获得不同的随机数序列，或者可以指定一个种子来生成不同的随机数。

④ 服务器将生成的随机数发送给终端。

⑤ 终端收到随机数后，用私钥进行签名 Tspi_Data_Bind(), 得到一个签名后的结果，并将结果拷贝到用户缓冲区。

⑥ 终端将签名结果发送给服务器端。

⑦ 服务器收到后用公钥进行解开签名结果 Tspi_Data_Unbind()，得到的明文与随机数进行比较，如果相同则身份验证成功。

通过以下三个方面证明基于 TPM 的身份认证的安全性和可靠性：

(1) 在 TPM 启动时，TPM 会利用内部的随机数发生器产生一个 2048 位的 RSA 密钥，即为 SRK 存储根密钥。存储根密钥为不可迁移密钥。嵌入式终端生成的公钥、私钥由 TPM 密钥层次结构中的父密钥 SRK 进行加密。在基于 TPM 的嵌入式终端中，加密密钥密封在其他应用程序无法访问的硬件中，即使受到暴力攻击，入侵者所花费的时间和代价都是极大的，而在攻击的过程中，TPM 会将外界试图猜测授权的行为告知操作系统。

(2) 在上述身份认证过程中，TPM 的密钥分层机制使用不可迁移的用户密钥。在 ATM 机或者电信商终端充值等终端固定的条件下具有很强的可靠性。

(3) 用户在终端使用密钥签名电子邮件，但又需要在家中及办公室的电脑使用电子邮件，这种情况通过使用 TPM 的可迁移密钥解决这一问题。TPM 可以识别密钥类型，会拒绝迁移一个不可迁移密钥。

本章实现的身份认证并非绝对意义上安全，只是在相对的条件的提出了终端的安全。在公钥传输和用私钥加密后的文件传输过程中有可能被截获，虽然入侵

者不会获得私钥，但是仍可以利用截获的数据进行欺骗行为。TPM 和 TSS 软件栈给出了许多命令和 API，可以利用这些功能实现安全性的扩展。例如在 TPM 启动时内部寄存器 PCR 值作为身份验证的一部分，在服务器和和终端之间建立某种同步机制，加强身份认证的安全性，为了使传输的明文不被破解，还可以先对明文进行哈希摘要，将摘要结果加密后传输。TPM 支持 CA（Certificate Authority）来加强对公钥和用户身份的管理。终端必须向服务器出示由 CA 发出的证书，使得服务器能准确获得该终端的公钥[44]。信息在网络传输的过程中，还可以引入时间戳来进行时间上的严格同步，即使被攻击窃取，信息也会因攻击要求较长时间而失效。

5.2.3 可信嵌入式平台性能分析

嵌入式系统是专用系统，实现相对较为单一的功能，与 PC 机中主板嵌入 TPM 芯片的安全解决方案相比，还需要考虑是嵌入式系统在加入 TPM 芯片后的性能问题。嵌入式系统往往要求低功耗，小尺寸，低成本的软硬件。对于嵌入式的实时系统来说，性能指标是至关重要的。嵌入式开发中必须对应用程序进行性能分析，以保证实时性要求，并且尽可能的提高应用程序的运行效率。本平台使用的 linux 操作系统虽然不是硬实时操作系统，但性能分析同样重要，在生成密钥，加密过程中通过考查函数执行时间和调用过程，判断 TPM 硬件和驱动程序的执行效率。本章测试工具选用 gprof。gprof 是 GNU profiler 工具。通过 gprof 能够打印身份认证过程中生成密钥、加密以及解密函数消耗的时间，可以就此分析出嵌入式平台在加入 TPM 芯片后对实时性的影响。在 arm-linux-gcc 交叉编译时加入-pg 参数，执行完终端程序后输入命令 gprof -b client gmon.out |less 会打印出该程序运行时各个函数的信息。本例只截取与 TPM 硬件相关的两个函数作为参考，函数运行信息如图 5-2 所示。

```
Flat profile:

Each sample counts as 0.01 seconds.
 no time accumulated

  %   cumulative   self              self     total
 time   seconds   seconds    calls  Ts/call  Ts/call  name
  0.00      0.00     0.00        1     0.00     0.00  Tspi_Data_Bind
  0.00      0.00     0.00        1     0.00     0.00  Tspi_Key_CreatKey
```

图 5-2 gprof 打印结果

在图 5-2 中，%time 表示本函数运行时间与整个程序运行时间的比值。Self seconds 为本函数自身所执行的时间。Calls 表示该函数在整个程序中被调用的次数。由于 gprof 只能显示到小数点后两位，密钥生成函数 Tspi_Key_CreatKey 和加密函数 Tspi_Data_Bind 的执行时间都在个位数毫秒以下，所以 TPM 芯片执行效率比较高，对整个平台的实时性基本没有影响。如果采用主频更高的嵌入式 CPU 以及更高性能的 TPM 芯片，整体性能还会提升。

5.3 可信嵌入式开发平台与传统嵌入式开发平台对比

通过以上章节的介绍，可信嵌入式平台是在传统的嵌入式开发平台基础上利用 IIC 总线扩展了 TPM 芯片，对引导程序进行了可信改造，并移植了为应用程序开发所使用的 TSS 软件栈，同时又没有破坏嵌入式平台低功耗、实时性的特点。可信嵌入式平台无论是功能模块还是在安全性上都有了很大的提升。具体对比如表 5-2 所示。

表5-2 可信嵌入式开发平台与传统嵌入式开发平台对比

	传统嵌入式开发平台	可信嵌入式开发平台
功能模块	网卡，USB 接口，音频接口，LCD，串口等	在传统平台的基础上，添加了双口 RAM、CAN 总线、CPLD、TPM 等模块，功能性更强
引导程序	U-boot 或 vivi	对 U-boot 进行了可信改造，在启动平台时对自身进行度量，并与存储在 PCR 中的值进行比对，防止了引导程序被篡改，提高了安全性
操作系统	嵌入式操作系统 Linux 或 winCE	移植了 TSS 软件栈到嵌入式操作系统中，为开发者提供了大量用于安全性开发的 API
安全性	操作系统完全裸露，无防护措施，安全性低	在 U-boot 可信改造后，还可以加入对操作系统及上层应用程序的度量，开发者利用 TPM 和 TSS 配合在应用程序中可以实现智能识别、登录口令验证、授权委托，进程度量等多种有效的安全措施，安全性高
实时性	高	通过对密钥生成函数和加密函数运行时间的分析，扩展 TPM 后对嵌入式系统的实时性基本无影响

5.4 小结

本章完成了一个基于 TPM 的安全身份认证，利用可信嵌入式开发平台模拟出一个嵌入式终端与服务器进行身份认证的模型。通过与传统的身份认证方法进行对比说明可信嵌入式平台的安全性与实时性。5.2 对加密密钥生成函数和加密函数运行时间的分析推断出扩展 TPM 不会对嵌入式系统的性能构成影响。5.3 中通过列表对比了传统嵌入式平台与可信嵌入式平台的特点。

结　论

本论文从实际入手，利用PC机和服务器中解决信息安全问题的技术范畴，将可信计算与嵌入式系统结合起来，构造了基于TPM芯片的嵌入式开发平台，为在嵌入式设备中应用TPM技术提供支持。本文提出的这种解决嵌入式终端安全问题的思想，弥补了嵌入式系统易受攻击、安全性差的缺点，从原理上实现了人们迫切需要的一个高安全性的嵌入式终端设备。基于TPM的嵌入式设备安全性之所以较高，是因为其安全性能是靠硬件支撑的，这种由传统应用软件单一策略来进行防护的措施到向基于TPM安全芯片而采用的软硬结合的综合技术措施，可为用户实现对嵌入式系统的真正防护。平台完成后，可供应用程序开发者，密码学爱好者等做进一步的开发，不仅仅使用TPM芯片，还可以利用本平台丰富的功能模块实现各种功能，如网卡、CPLD、双口RAM、CAN总线、USB接口等等。按照章节顺序，本文主要做了如下的工作：

(1) 本文结合可信嵌入式平台的开发背景和意义，首先对可信嵌入式开发平台的可行性进行了分析，依据硬件原理的不同和前期硬件制板的经验，选择了适合嵌入式平台上的 TPM 芯片，按照实验室前期对 ARM9-S3C2410 这款芯片积累的经验和资料重新对硬件平台进行了设计，利用 IIC 总线扩展了 TPM 芯片，并添加了其他流行的嵌入式开发的功能模块，如 CPLD、CAN 总线、双口 RAM 等。相比传统嵌入式开发平台功能性更强，安全性更高。

(2) 制板工作完成后，通过编写测试程序和示波器对硬件平台进行了调试，确保硬件开发板无连接的错误与设计缺陷。

(3) 读芯片手册掌握硬件的特性后，完成了 TPM 驱动的编写。首先编写的是 IIC 总线的驱动，TPM 驱动的读写操作直接调用 IIC 驱动的读和写来完成最核心的功能。通过编译成模块后添加到内核当中，供上层应用程序调用。

(4) 完成了嵌入式引导程序 U-boot 的可信改造，汇编语言添加了直接对 TPM 的操作，控制 TPM 开启和实现对引导程序的度量，并且返回度量报告给平台使用者，实现了信任的原点。然后根据嵌入式系统的特点，移植了 PC 机中使用的 TSS 软件协议栈 Trousers，为上层应用程序开发提供了 API。编写了上层应用程序调用 TPM 驱动程序直接向 TPM 发送命令，通过程序执行结果得知 TPM 设备可以正常使用。通过编写了一个应用程序调用 TSS 提供的 API 来实现 TPM 创建一个绑定密钥并对文件加密，这个程序验证了 TSS 软件协议栈向嵌入式操作系统中移植成功。到此，整个可信嵌入式开发平台的软硬件体系已经全部完成。

(5) 实现了基于TPM的安全身份认证，利用可信嵌入式开发平台模拟一个嵌

入式终端与服务器进行身份认证，并与传统的身份认证方法进行对比说明可信嵌入式平台的安全性与实时性。

由于时间的原因，TPM还有大量的功能没有被开发。通过TPM本身的特性和TSS提供的API还可以实现很多解决安全问题的功能，如智能识别、登录口令验证、授权委托、进程度量等等。基于该平台可方便地模拟专用的可信模块或可信设备，同时也支持用户构建个性化的可信应用，并在此基础之上开发符合自身需求的嵌入式可信应用软件和产品，支持灵活多变的开发模式。

参考文献

[1] 邓漫龄. ARM 嵌入式 Linux 系统的研究与实现[D]. 北京：北京邮电大学通信与信息系统硕士论文，2009：1-3.

[2] 张焕国，覃中平，刘毅，等. 一种新的可信平台模块[J]. 武汉：武汉大学学报信息科学版，2008（10）：1-2.

[3] SIEWIOREK Daniel P，杨孝宗. Industry Trends and Research in Dependable Computing[J]. 计算机学报，2008（10）：2-4.

[4] 胡文元. 中国工商银行深圳分行电子银行业务发展对策研究[D]. 湖南：中南大学硕士论文，2008：8-10.

[5] 沈昌祥. 坚持自主 创新加速发展可信计算[J]. 计算机安全，2006（6）：4-5.

[6] 赵波，张焕国，李晶，等. 可信 PDA 计算平台系统结构与安全机制[J]. 计算机学报，2011：2-4.

[7] 张焕国，赵恒，等. 可信计算平台信任链安全性分析[J]. 计算机学报，2010（1）：1-6.

[8] 郑宇，何大可，何明星. 基于可信计算的移动终端用户认证方案[J]. 计算机学报，2006（8）：1-3.

[9] 冯登国，秦宇，汪丹，等. 可信计算技术研究[J]. 计算机研究与发展，2011(8)：3-6.

[10] David Challener,Kent Yoder,Ryan Catherman，等著，赵波，严飞，俞发江，等译. 可信计算[M]. 北京：机械工业出版社，2009：18-220

[11] TrouSers.TheopensourceTCGSoftwareStatck.

[12] Lei HAN,Jiqiang LIU,Zhen HAN,Xueye WEI. Design and implementation of a portable TPM scheme for general-purpose trusted computing based on EFI[J]. Frontiers of Computer Science in China，2011(2)：1-5.

[13] SONG Cheng,LIU Bing,HU Zheng-ming,XIN Yang,YANG Yi-xian,YIN Han. Efficient ID-based TPM key loading scheme for trusted platform[J]. The Journal of China Universities of Posts and Telecommunications，2010(4)：2-4.

[14] 田泽. ARM9 嵌入式开发实验与实践[M]. 北京航空航天大学出版社，2006：202-208.

[15] 秦贵和，徐华中，王磊. ARM9 嵌入式技术及 Linux 高级实践教程[M]. 北京：北京航空航天大学出版社，2005：22-27.

[16] 周拥军，万里青，范治玉. IIC 总线在数字视频记录仪中的应用与实现[J]. 电

光与控制，2004(6)：3-5.

[17] 赵丽丽. ARM9 实验开发板设计[D]. 西南交通大学，2007：39-45.

[18] 韦荣.可信计算平台可信度量机制的应用与研究[D]. 西安电子科技大学硕士论文，2008：68-72.

[19] Sean W.Smith 著，冯登国，徐震，张立武，等译. 可信计算平台：设计与应用[M]. 北京：清华大学出版社，2006：1-5.

[20] 孙天泽，袁文菊，著. 嵌入式设计及 Linux 驱动开发指南[M]. 北京：电子工业出版社，2009：88-130.

[21] 王博，李波. 基于 TPM 的嵌入式可信计算平台设计[J]. 成都：单片机与嵌入式系统应用，2011(1)：3-4.

[22] 吴少刚，许华. 可信嵌入式龙芯启动加载程序 tPMON 的设计[J]. 计算机工程与设计，2008(1)：3-5.

[23] 王禹，王震宇，等. 嵌入式平台 TPM 扩展及可信引导设计与实现[J]. 计算机工程与设，2009(9)：1-3.

[24] (美)科波特，LINUX 设备驱动程序[M]. 中国电力出版社， 2006：23-178.

[25] DanielP，Bovet. 深入理解 LINUX 内核[M]. 中国电力出版社，2008：1-139.

[26] 方艳湘，黄涛. Linux 可信启动的设计与实现[J]. 计算机工程，2006(9)：4-6.

[27] 韩立毛，赵跃华，马祥顺. 嵌入式操作系统的内核安全研究与设计[J]. 计算机工程与设计，2010(14)：1-3.

[28] 张焕国,冯登国,等.信息安全综述[J].中国科学 E 辑信息科学,2007(2):13-140.

[29] 屈延文. 软件行为学[M]. 北京：电子工业出版社，2004：261-265.

[30] 李晓勇，左晓栋，沈昌祥. 基于系统行为的计算平台可信证明[J]. 电子学报，2007(7)：3-5.

[31] 卿斯汉. 国外可信计算的研究进展[J]. 信息安全与通信保密，2007(9)：22 -24.

[32] 彭彦. 基于 Java 智能卡的可信度量机制分析与实现[D]. 西安电子科技大学，2009：14-46.

[33] Chen Shu-Yi. Journal of Northeastern University(Natural Science)，2008： 96-99

[34] 孙勇，陈伟，杨义先. 嵌入式系统的可信计算[J]. 信息安全与通信保密，2006（9）：4-5.

[35] 胡庆武，崔贤玉. 基于 ARM 的嵌入式系统 Bootloader 的编译与启动分析[J]. 科学技术与工程，2007（14）：4-7.

[36] SailerR, Zhang XJaegerTl.Design and implementa-tion of a TCG-based integrity measurement architecture[C] //Proceedings of the 13thUsenix Security Symposium.California：Usenix，2004：223-238.

[37] OppligerR，Rytz R. Does trusted computing remedy computer security problem [J]. Security& Privacy Magazine(IEEE)，2005(2)：16-19.

[38] 张焕国，罗捷，金刚，等. 可信计算研究进展[J]. 武汉大学学报，2006（5）：2-5.

[39] 李佳著. ARM 系列处理器应用技术完全手册[M]. 人民邮电出版社，2006：22-45.

[40] 程卫芳，廖湘科，田斌，等. 可信授权技术的研究和实现[C]. 第一届中国可信计算和信息安全学术会议. 武汉大学学报(理学版)50 卷 l 期，2004：177-221.

[41] 朱强，杨义先. 一种可信软件栈的设计与实现[D]. 北京：北京邮电大学，2009：16-20.

[42] 李伟伟. 面向对象的安全嵌入式操作系统的研究[D]. 南京邮电大学，2003：5-10.

[43] 梁正平，毋国庆，张焕国. LSM 访问控制研究[J]. 计算机工程，2004（13）：1-4.

[44] 阮越，王成耀. 基于 LSM 的安全访问控制实现[J]. 计算机工程，2004（1）：7-8.

[45] WireX Communications.Linux Security Module.http://lsm.immunix.org/2001.

[46] 徐贤，龙宇，毛贤平. 基于 TPM 的强身份认证协议研究[J]. 计算机工程，2012（4）：3-5.

[47] T.Jaeger，R.Edwards，X.Zhang. Consistency analysis of authorization hook placement in the Linux security modules framework[C]. ACM TISSEC，2004(2)：115-325.

[48] X.Zhang,A.Edwards,T.Jaeger.Using CQUAL for static analysis of authorization hook placement.In 11thUSENIX Security，2002：15.

[49] TCG Software Stack(TSS)SpecificationVersion1.2.2007.

[50] 董玉娟，李健. 一种支持异构可信平台的可信计算软件栈研究[J]. 网络安全技术与应用，2008（10）：11

[51] Sean W.Smith. Trusted Computing Platforms:Design and Applications[M]. New York:Springer Press，2007：168-170.

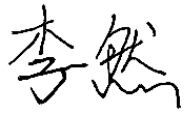
[52] David Challener,Kent Yoder，Ryan Catherman,David Safford,Leendert Van Doorn. A Practical Guide to Trusted Computing[M]. New York：IBM Press，2008：56-59.

[53] TCG infrastructure committee reference architecture for integrity information interoperability[Z]，2004：11-26.

哈尔滨工业大学学位论文原创性声明及使用授权说明

学位论文原创性声明

本人郑重声明：此处所提交的学位论文《基于 TPM 的可信嵌入式平台的设计与实现》，是本人在导师指导下，在哈尔滨工业大学攻读学位期间独立进行研究工作所取得的成果。据本人所知，论文中除已注明部分外不包含他人已发表或撰写过的研究成果。对本文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明。本声明的法律结果将完全由本人承担。

作者签名：  日期： 2012 年 7 月 7 日

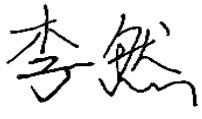
学位论文使用授权说明

本人完全了解哈尔滨工业大学关于保存、使用学位论文的规定，即：

（1）已获学位的研究生必须按学校规定提交学位论文；（2）学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文；（3）为教学和科研目的，学校可以将学位论文作为资料在图书馆及校园网上提供目录检索与阅览服务；（4）根据相关要求，向国家图书馆报送学位论文。

保密论文在解密后遵守此规定。

本人保证遵守上述规定。

作者签名：  日期：2012 年 7 月 7 日

导师签名：  日期：2012 年 7 月 7 日

致　谢

本课题是在柏军副教授的亲切关怀和指导下完成的，他治学严谨，学识渊博，对待工作严格认真，是位让人尊敬的长者，不仅仅从学习上开拓了研究视野，还在生活中给予了极大的关怀和鼓励。从本文开始的选题，到开发期间软硬件的设计再到最后论文的撰写，柏老师都做了大量悉心指导的工作，提出了许多宝贵的意见和方案，对整个课题的完成起了至关重要的重要。至今，柏老师的多次指导仍记忆犹新。至此，毕业论文的工作已接近尾声，向尊敬的柏老师致以最衷心的感谢。

特别感谢嵌入式实验室的张策、吕为工老师，他们为本课题提供了研究条件，张老师从可信计算的方向给我了很多指导，特别是在论文的撰写中给我了很大的帮助。吕老师在软硬件设计和调试过程中提出了许多可行的思路，帮助我解决了很多硬件设计上难题，在此，特别向两位老师表示感谢。

感谢论文中参考文献的作者们，正是你们的前期的工作和研究思想为我的工作铺平了道路，向你们表示感谢。

在求学期间，我的亲人、同学和朋友都对我给予了无微不至的关怀，对此，我也表示深深的感谢。