

嵌入式 web 服务及其在远程实验中的应用

刘青云, 肖铁军, 马学文, 沈 健

(江苏大学 计算机科学与通信工程学院, 江苏 镇江 212013)

摘 要: 分析低端嵌入式 web services 的研究现状和不足, 提出了一种在 STM32F407 平台上设计的适合于低端嵌入式系统的 web services 实现方案, 该方案的主要工作包括 XML 解析器、嵌入式 HTTP 和 SOAP 处理器的实现, 以及与具体服务绑定和应用. 该嵌入式 web services 具有实时性高、易移植和资源占用少等特点, 现已成功应用于远程教学实验系统, 实验结果证明该方案的实现运行稳定, 具有良好的可行性和推广价值.

关键词: 嵌入式; web services; 远程实验; SOAP; XML

中图分类号: TP393.0

文献标识码: A

文章编号: 1000-7180(2014)08-0085-05

Embedded Web Services and Application in Remote Experiment

LIU Qing-yun, XIAO Tie-jun, MA Xue-wen, SHEN Jian

(School of Computer Science and Telecommunication Engineering, Jiangsu University, Zhenjiang 212013, China)

Abstract: Analysis the research status of web services in low-end embedded systems(EWS) and the deficiencies. Present a kind of implementation designed for EWS on STM32F407, the main work includes realization of XML parser, embedded HTTP and SOAP processor, then bind this web services with specific application. The characteristics of this EWS include a high real time capability, easily-move and less resource etc. And this EWS has been successfully applied in remote experiment. The experimental results show that the application of this scheme is equipped with stable operation, good feasibility and replicability value.

Key words: embedded; Web services; remote experiment; SOAP; XML

1 引言

随着嵌入式和网络技术的发展, 嵌入式系统逐步摆脱了以往“信息孤岛”的局面, 相互之间形成了分布式的协作关系^[1]. Web services 具有接口统一、松散耦合、应用集成和可跨防火墙等特点^[2], 有利于实现嵌入式系统异构平台间的无缝集成, 可有效解决其通信中存在的平台异构、接口不统一和防火墙限制等问题.

针对嵌入式 web services(EWS)的实现方案, 多以应用为主, 采用性能较高的嵌入式处理器和较为成熟的 SOAP 工具. 针对低端 EWS 的研究, 文献[1]中以 ARM Cortex-M3 为核心, 实现了低端 EWS, 但文中设计的 HTTP 协议只实现 POST 方法, 也无法处理报文体错误以及客户端对 WSDL 的

请求; 文献[3]中设计的 EWS 以调用其他服务为主, 未涉及到 web 服务的绑定和向外提供服务调用接口; 文献[4]在 NiosII 平台上设计了 EWS, 其中 XML 拉式解析器 XPPES 占用内存少, 但未涉及 XML 生成方法.

本文在深刻理解目前低端 EWS 研究的基础上, 以 32 位微处理器 ARM Cortex-M4 为核心, 借助嵌入式实时操作系统和 TCP/IP 协议栈, 设计了 HTTP 引擎、XML 解析器以及 SOAP 处理器, 并在此基础上与具体应用服务绑定, 实现了嵌入式 web services, 并成功应用于远程教学实验系统.

2 远程实验系统的总体结构

本文研究的嵌入式 web services 以计算机组成原理课程实验为应用背景, 主要用于远程教学和远

程实验,从总体结构上来看,系统可分为三个部分(如图 1 所示):FPGA 计算平台、调试服务器和客户端.



图 1 远程教学实验系统的总体结构图

FPGA 计算平台是整个实验系统中实验数据计算的硬件平台,实验人员可以自行编写相应的硬件电路文件并通过 EthernetBlaster 下载到该实验平台中,再配合客户端程序就可以实现具体的远程实验操作. 调试服务器(本课题研究的嵌入式 web services 的载体)则是 FPGA 实验平台和客户端的中间设备,通过网络端口,调试服务器可以与远程客户端进行服务请求和响应,通过另一端的 JTAG 接口,调试服务器可以与 FPGA 实验平台进行数据和指令交流,并间接控制实验平台的运行. 客户端(服务的请求者)为 FPGA 计算平台的数据和信号窗口,通过客户端可向 FPGA 计算平台发送实验数据和命令,经过调试服务器的转化后,便可控制计算平台的运行并返回数据,实现远程实验操作.

3 嵌入式 web services 的技术实现

硬件部分,本文以 ST 公司的 STM32F407VGT6 芯片为核心,它提供最高 168 MHz 的工作频率、1 MB 片内 Flash 和 192 + 4 kB 的 SRAM,并外接 DP83848 以太网 PHY 控制器,完成底层网络通信,另外,开发板提供了 SPI 扩展接口,SPI 接口按照 JTAG 通信时序收发数据可与 FPGA 实验平台完成 JTAG 方式的数据通信. 软件部分,本文选择嵌入式实时操作系统 FreeRTOS 来完成实时和多任务控制,并选用轻量级 TCP/IP 协议 LwIP 来完成底层的 TCP/IP 通信.

3.1 HTTP 引擎设计

HTTP 在 TCP/IP 之上,是基于请求/响应模式的协议,为嵌入式 web services 底层通信支撑. 其通信过程分为 4 个步骤:建立连接、发送请求报文、接收响应报文和关闭连接. 在保证嵌入式 web services 可行的情况下,本文为了节省嵌入式系统资源,只实现 HTTP 八种常用方法中的 POST 和 GET 方法,POST 方法主要用于服务提供者和客户端间互

相传送 SOAP 消息,而 GET 方法则在客户端向服务提供者索取 WSDL 文档时使用.

图 2 显示了本文中嵌入式 web services 在一次服务请求中 HTTP 引擎的主要工作流程. 服务器建立 TCP 监听后,便可接收客户端 HTTP 请求,在收到 HTTP 报文数据后,先使用本文中设计的 HTTP 解析器分析 HTTP 报文语法是否正确,如不正确,返回错误码为 400(Bad Request)的回复报文;若报文语法正确,HTTP 引擎则根据报文的方法做进一步处理,来自客户端的 HTTP 报文有两种类型:GET 和 POST,若接收到的是 GET 报文,则向客户端返回 WSDL 文档(状态码 200);若接收的是 POST 报文,则组织相关实验数据打包成 SOAP 回复包(状态码 200),使用原 TCP 连接将该 SOAP 包回送给服务请求者,如果为其它方法,则返回错误 405(Method Not Allowed),并根据 TCP 连接的状态判断是否需要继续侦听 HTTP 报文.

3.2 XML 解析器设计

Web services 中服务提供者和请求者之间传递的 SOAP 消息以 XML 格式作为消息数据编码规范,XML 解析器可从 SOAP 消息中获取所包含的请求方法和参数数据等信息. 目前有两种常用类型的 XML 解析器:DOM 解析器和 SAX 解析器. DOM 解析器需在内存中建立和 XML 文件对应的树形结构,对内存消耗大. 本文设计一款基于事件流驱动的 SAX 解析器,SAX 解析器具有文档解析和文档生成两部分功能.

(1) XML 文件解析

本文中的 SAX 解析器逐字节解析 SOAP 消息包中的 XML 文档,解析器每当遇到一个标记、指令或者文本等节点,便会调用事件处理器来处理该事件,本文定义如下的结构体用来指定事件处理器中处理相应事件的处理函数(清单一).

```
typedef struct{
    XML_StarDocumentHandler  startDocumentHandler;
    XML_EndDocumentHandler   endDocumentHandler;
    XML_StartElementHandler  startElementHandler;
    XML_EndElementHandler    endElementHandler;
    XML_CharacterDataHandler  characterDataHandler;
    XML_ParseErrorHandler    parserErrorHandler;
    XML_CommentHandler       commentHandler;
    XML_InstructionHandler    instructionHandler;
    XML_DeclareHandler        declareHandler;
}XML_Parser;
```

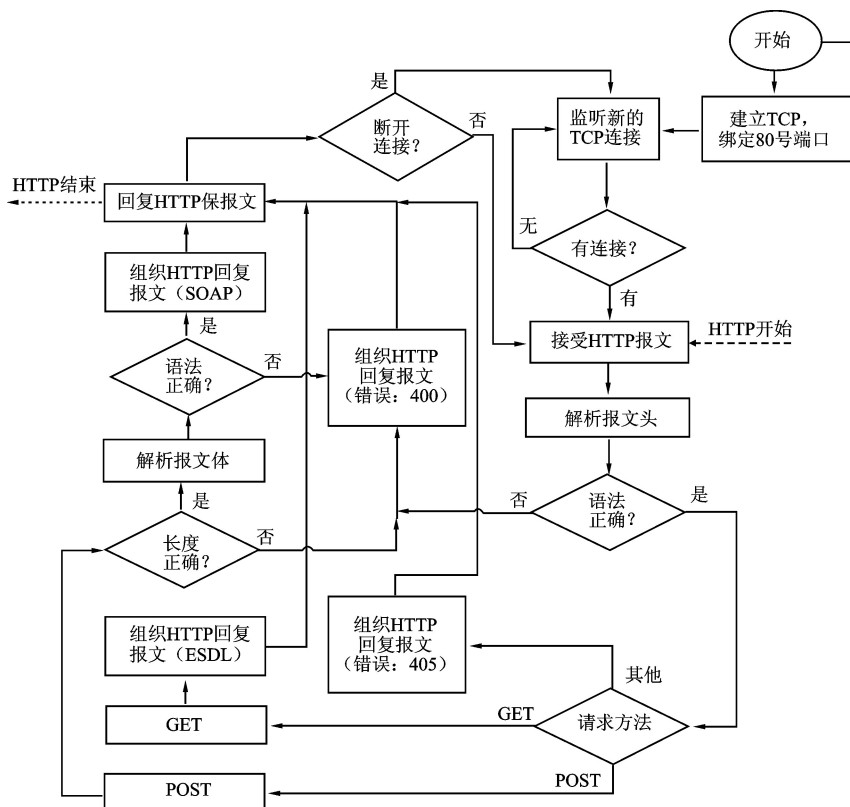


图 2 HTTP 引擎工作流程图

当上层程序使用 SAX 解析器解析 XML 文档之前,必须先初始化该事件处理器,指定在文档解析过程中相应事件的处理方法;本文中设计了 XML_doParse()方法用来解析 XML 文件,使用前必须传入 XML 文档数据源以及 XML 事件处理器,之后 SAX 解析器便开始逐字节解析 XML 文件,每当遇到某个事件(文档开始、标记开始、文本事件等)时,解析器便会调用事件处理器中相应的事件处理函数,并将此次事件所包含的名称、属性和节点位置(指在逻辑树中的位置)等信息以参数的方式传递给这些事件处理函数(回调函数),在回调函数中对相应的事件进行处理。解析器在某个 HTTP 连接完成后需要释放空间,以节省不必要的内存消耗。

(2) XML 文件生成

本文为了简化上层程序生成 XML 文件的工作,编写了 XML 树节点插入函数: int InsertXmlElement(const char * pos, const char * nam, const char * attri, const char * character)来简化该工作。该函数在已有 XML 文件中的指定位置(是指 XML 逻辑树中某个节点的逻辑位置,而非 XML 文档字符串中的位置)插入相应的节点(包含节点名称、属性和文本信息),需要在 XML 文件中寻找节点的具体插入位置(在字符串中的位置),本文设计

一款节点寻找算法来确定在 XML 文档字符串中需要插入节点字符串的精确位置。

例如,需要在图 3 中的简单 XML 文件中插入一个雇员节点,则需要找到“雇员></雇员列表”这两个尖括号中间的这个位置(在逻辑树中的位置为“11”,表示为根节点下第一个子节点后),算法中本文使用表 1(T1[N])来保存要插入的节点在 XML 逻辑树中的位置,表 2(T2[2][N])作为辅助表格,用来保存当前查找的位置,Flag 为游标。

--- XML文档示例 ---

```

<雇员列表>
  <雇员>
    <姓名> 张小四 </姓名>
    <性别> 男 </性别>
  </雇员>
</雇员列表>

```

表1 插入节点位置表格							
1	0	0	0	0	0	0	
Flag	表2 辅助查找表格						
	0	X	X	X	X	X	X
0	X	X	X	X	X	X	

图 3 XML 示例文档和辅助查找表格

算法中使用的 Flag(初值为 0)用来表示查找到的当前节点的层数,并逐字符查找节点,当查找到一个节点的结束位置时, T2[0][Flag++]++; T2[0][Flag] = 0; T2[1][Flag] = 0; 当查找到一个节点的开始位置时, Flag--; T2[1][Flag]++; 如果 Flag 的值等于表 1 中插入节点的层数,而且表 2 第一行的内容和表 1 的内容完全相同时(下标范围:

0—Flag), 查找结束, 即可在此位置上插入目标节点对应的字符串数据。

3.3 SOAP 消息处理

SOAP 是基于 HTTP、XML 的网络通信协议, 它用于在分布式环境中发送消息, 并执行远程过程调用。在嵌入式 Web services 的服务提供者和请求者之间以 SOAP 消息包传送数据, SOAP 可以简单地理解为: SOAP = RPC + HTTP + XML, RPC 为远程过程调用; HTTP 协议为底层的通信协议; SOAP 消息以 XML 为数据编码格式。

SOAP 消息处理以 HTTP 引擎和 SAX 解析器为基础, 其具体实现过程如为图 4 所示: web services 调用者根据服务提供者发布的 WSDL 文档打包基于 XML 格式并包含远程调用方法和参数数据的 SOAP 包, 使用 HTTP 协议中的 POST 或者 GET 方法将该 SOAP 包通过网络发送给服务提供者, 服务提供者接收到请求端的 SOAP 数据包后, 启动 HTTP 引擎处理 HTTP 报文头, 判断是否存在语法错误, 并使用 XML 解析器来解析 SOAP 包中函数的服务请求方法和参数数据, 若 SOAP 消息包正确则将相关数据进一步转化并通过 JTAG 接口传送给 FPGA 实验平台, 实验平台处理数据并返回, 服务提供者分析并组织此数据, 按照 WSDL 文档规定的格式打包成 SOAP 包, 再通过之前建立的 TCP 链路将 SOAP 包回送给服务请求者, 完成一次远程服务调用的整个过程。

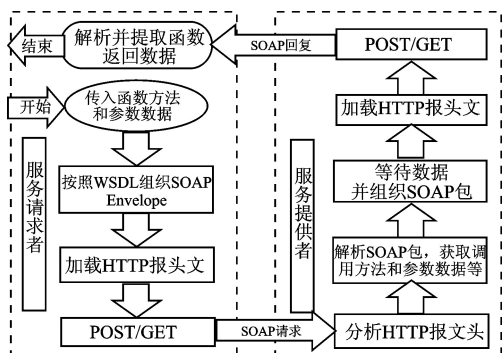


图 4 SOAP 远程调用流程图

4 实验系统远程服务实现

远程服务实现在上述嵌入式 Web services 技术的基础上, 结合实验系统的具体的实验要求, 实现实验系统中远程控制功能。本文中的远程实验系统客户端(服务请求者)请求服务后, 服务端(服务提供者)会在一定时间内绑定该客户, 即维持本次 TCP 连接, 能有效节约 TCP 连接反复建立和释放所消耗

的时间(符合 HTTP1.1 规范); 在 HTTP 实现中, 本文只实现其中的 POST 和 GET 方法, 在 HTTP 报文的 POST 方法中, 只包含了 SOAP 与 HTTP 绑定时必需的两个字段(清单二): Content-Type 和 Content-Length, 分别表示消息的 MIME 类型、字符编码(可选)和请求或响应主体的字节数。

```

POST /item HTTP/1.1
Content-Type: application/soap+xml; charset=utf-8
Content-Length: 1024

```

在 SOAP Envelope 的组织中, 该嵌入式 web service 根据实验系统远程方法调用方式简单固定的特点: 每次远程方法调用中, 都需要发送若干固定指令, 每条指定必须携带任意有限长度十六进制数据, 所以可以将远程调用方法打包成如下的 SOAP 消息包(清单三)。

```

POST /item HTTP/1.1
Content-Type: text/plain
Content-Length: 200
<? xml version="1.0" encoding="utf-8"? >
<soap: Envelope xmlns: soap = "http://schemas.xmlsoap.org/soap/envelope">
  <soap: Body>
    <PacketHead>0101215144...</PacketHead>
    <StatetoRTL>173278342...</StatetoRTL>
    <RTtoShiftDR>452345278...</RTtoShiftDR>
    <RtTltoShiftIR>4534530130...</RtTltoShiftIR>
    ...
    <PacketHead>45201045...</Packethead>
  </soap: Body>
</soap: Envelope>

```

服务端接收新连接、绑定客户端并等待服务请求, 当接收到客户端上述这种 SOAP 消息包之后, 会解析 SOAP 数据包, 从中获取服务请求者发送的服务请求中包含的参数和数据, 并进一步获取最终数据和打包 SOAP 消息, 再通过原连接返回 SOAP 包, 然后继续等待客户端下一个服务请求, 直到超时或者 TCP 连接断开, 之后再解除绑定释放内存。

5 嵌入式 Web services 应用及验证

本文中嵌入式 web services 在远程教学实验系统中得到验证和应用, 嵌入式 web services 定位于调试服务器之上, 由底层硬件和 FreeRTOS、LwIP 支撑, 并使用上层 HTTP 引擎、XML 解析器和 SOAP 处理器向外提供 web 服务, 并以 FPGA 计算平台为服务的最终数据来源。本文接下来通过远程实验中具体实例来验证该嵌入式 web services 的正确性和可行性。

实例 1:该实例中,客户端软件向嵌入式 web services 提供者(图 1 中的调试服务器)发送 SOAP 请求来获取服务端 FPGA 的硬件 ID 号,服务端收到请求后将获取 ID 号的指令和数据发送给 FPGA 计算平台,并通过 JTAG 接口获取 FPGA 的 ID 号,并将此 ID 号组织成 SOAP 包,回复给客户端软件.本文启动客户端软件开发平台的调试功能,捕获到该 SOAP 回复包,如图 5 所示,经对比分析,SOAP 包中的 ID 号与 FPGA 硬件 ID 号一致,该 SOAP 包的组织结构符合 SOAP 协议规范.

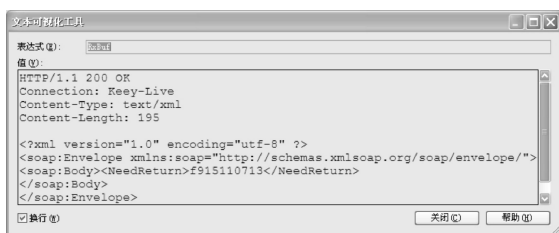


图 5 携带硬件 ID 号的 SOAP 回复包截图

实例 2:该实例中,客户端软件将每一步实验操作(包括指令和读写数据)打包成服务指定格式的 SOAP 包发送给服务提供者,服务提供者收到请求后进行数据解析、转发并获取 FPGA 计算平台的实时状态数据,再打包成 SOAP 包回送给客户端,完成一次实验操作服务请求/响应过程.图 6 中客户端软件界面上显示了请求得到回复后实验平台的实时状态,经对比,两者状态一致.

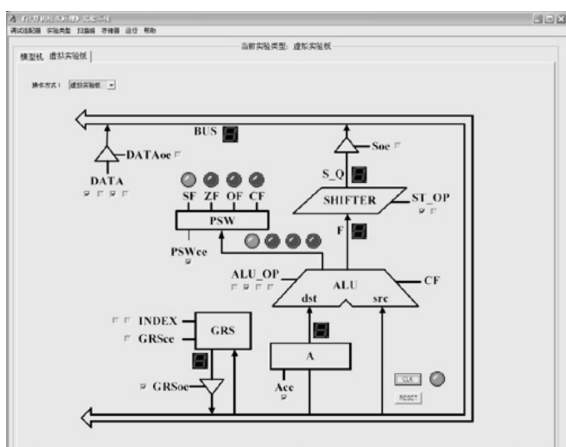


图 6 客户端软件界面显示实验状态

通过以上两个实验实例可以得出以下结论,该嵌入式 Web services 接口符合规范,在远程实验中的应用中数据传输正确,满足远程实验操作的应用要求,由此验证了该嵌入式 Web services 的正确性和可行性,也说明了在嵌入式异构平台间的数据交互中具有一定的推广价值.

6 结束语

本文选用 ST 公司 32 位微处理器 ARM Cortex-M4 系列芯片 STM32F407VGT6 和网络芯片 DP83848 为基础硬件,并以 FreeRTOS 和 LwIP 为软件基础,实现了不易实现的嵌入式 web services 技术,其中 HTTP 引擎的设计,能够满足嵌入式 web services 对 HTTP 的基本需求;SAX 类型的 XML 解析器基于事件流驱动,实时响应事件,与实验系统的具体应用要求相吻合,并设计了 XML 生成算法来简化 XML 生成工作;SOAP 处理器在 HTTP 引擎和 XML 解析器基础上,能正确处理 SOAP 请求^[5-6].该嵌入式 Web services 被成功应用于远程教学实验系统,实验结果也证明了该方案的应用运行稳定,具有良好的可行性和推广价值,方案采用纯 C 编码,可移植性好. Web services 具有的接口统一、松耦合和跨越防火墙等特点与嵌入式设备具有的专用性强、平台异构等特点相吻合,嵌入式 web services 的发展,能在未来嵌入式系统之间、嵌入式系统和其他异构平台的数据和信息集成中发挥重要作用.

参考文献:

- [1] 王海立,周杏鹏. 嵌入式环境下 Web Service 技术的实现[J]. 现代电子技术, 2011, 34(22): 61-64.
- [2] 陆云龙,雷志华. 嵌入式 Web Service 技术的设计及应用[J]. 微电子学与计算机, 2008, 25(1): 65-67.
- [3] 刘猛,刘杰,刘冰,等. 单片机系统中 Web Service 的调用方法研究[J]. 微计算机信息, 2008, 24(14): 79-81.
- [4] 黄丽君,肖铁军,刘芳. SOAP 及其在嵌入式 Web 服务中的实现[J]. 计算机工程与设计, 2008, 29(16): 4135-4137.
- [5] Shelby Z. Embedded web services[J]. Wireless Communications, IEEE, 2010, 17(6): 52-57.
- [6] Wahib M, Munawar A, Munetomo M, et al. A framework for cloud embedded web services utilized by cloud applications[C]// Services (SERVICES), 2011 IEEE World Congress on. USA: Wasington, IEEE, 2011: 265-271.

作者简介:

刘青云 男,(1988—),硕士研究生.研究方向为嵌入式系统;

肖铁军 男,(1963—),硕士生导师,教授.研究方向为嵌入式计算及系统.