

Design and Implementation of Embedded Web Server Based On SOA

Zheng Liu^{1, 2, a}, Feng An^{1, b},

¹Nanometer Information Dept. of Suzhou Industrial Park Institute of Services Outsourcing,
Suzhou, 215123, China

²Suzhou High-tech Key Laboratory of Cloud Computing & Intelligent Information Processing,
Suzhou, 215123, China

^aemail:liuz@siso.edu.cn , ^bemail:anf@siso.edu.cn

Keywords: SOA; embedded web server; PHP

Abstract. SOA, which means Service-Oriented Architecture, is a component that could connect different terminals or devices without caring about the difference of each platform. Meanwhile, an embedded web server is a server within embedded systems. In this paper, we proposed a new idea to build the embedded web server: by means of SOA. The web server is transferred from embedded devices to PC, via PHP+HTML functionality, the embedded devices communicate with the server through the web request. Users can access the Web server to get information of the embedded devices, and can control devices, such as running through a browser. Take into account this idea, we liberated the operating pressure of embedded devices to improve the friendly interface, which has a high application value.

1. Introduction

Traditionally, there are mainly two methods to build an "embedded web server": directly based on the open source network protocol stack (for example :UIP(User Interface Process)), or Linux-based platform for lightweight server ((for example :Httpd). Whatever the way you choose, web server needs to be running within embedded devices, which also explain the reason for its name "embedded". However, with the web technology continuous development , the use of server increased rapidly, thus embedded web server is no longer composed of simple web send and CGI technologies, but needs to take on more complex functions, such as form updates, real-time streaming, large amount of data exchange/storage and complex logic behind management. However, the processing capabilities of embedded devices are vary widely from each other. Many devices could not afford a large operating pressure of the web server. Based on this case, we propose a new approach to build the server: transfer the server to a more powerful PC, through the sophisticated system ,which is based on PHP + HTML + JavaScript and other technologies to achieve powerful, elegant interface of the web server. Since server provides a feature named "web service", also known as SOA, with the client (embedded devices) to share informations. For users, direct access or control embedded devices through a browser, no sense of position changes within the Web server, so it can still be called "embedded web server". This strategy is comparable to the common website design in the foreground and background architecture, reached the minimum resources to maximize the effectiveness of operational efficiency.

Based on this idea, there is not necessary to worry about the limitations of hardware resource for Engineers, and in the practical application, select mainly Apache+PHP+HTML program to build the server, this schematic embedded web server as Fig.1 and Fig.2.

The more node equipments added to the applicaton system, or the more different development platform using in the application system (various mobile phones,equipment etc.), the more strengths can be reflect to this kind of embedded web server based on SOA.Through the Web service interface to communicate with a large number of devices, and devices through Web requests sending their collected data to the server, and then the server to collect data after logical analysis will be conducted in accordance with the PHP script specified function of these data stored or

discarded. Meanwhile, the server provides a complete Web page, so the user can access to the appropriate data through the browser, or some control over the device.

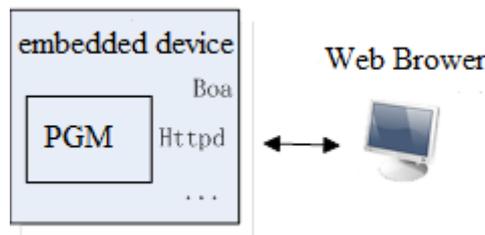


Fig.1. traditional embedded web server

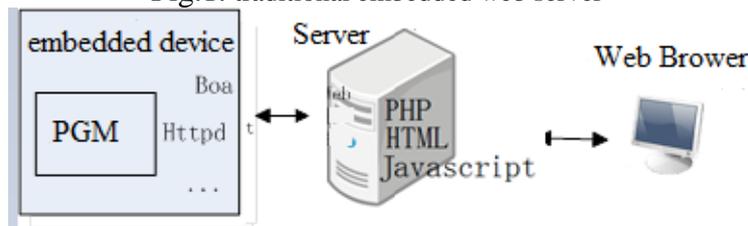


Fig.2. embedded web server based on SOA

To build this server, some works have to be done. First set up a web server in the traditional sense and implement service interfaces; the second is to write web request procedures in which embedded devices, constantly updating their status or send a message to the site among server. These elements will hereinafter be described in detail.

2. Web server set up

limited by the processing capability of embedded devices, previously only by means of lightweight Web server, such as BOA, HTTPD and so on. Now, we can take a more appropriate Web server. Apache, an open source HTTP server that can run on most computer operating systems, is one of the most popular Web server software due to its strengths of cross-platform and security. In this project, we can develop an efficient server, using a currently popular integrated installation package WAMP (or LAMP), which contains the Apache, PHP and Mysql and other functions.

The steps to build the server and development environment under Windows or Linux platform (ubuntu12.04) are as follows:

Setp1: LAMP installation

```
sudo apt-get install apache2 mysql-server mysql-client php5 php5-gd php5-mysql
```

Setp2: phpmyadmin installation:

```
sudo apt-get install phpmyadmin
```

Setp3: Apache configuration:

Enable mod_rewrite module : `sudo a2enmod rewrite`

Restart Apache server: `sudo / etc/init.d/apache2 restart`

WAMP installation is very simply, download the Windows platform integration WAMP installation package, the installation can be carried out accordingly.

3. Service Oriented Architecture

Service-Oriented Architecture (SOA) is a new technology that makes running different applications on different devices can exchange data or integrate data with each other, without the aid of additional, specialized third-party software or hardware. Web services communicate with the client via standard data interface, the client sends and receives data packets are encapsulated into JSON format, and that guarantees the scalability and maintainability of the system.

General by writing PHP scripts to implement many services Web services interfaces , in this project, we achieved up to 21 service interfaces, the client accessed through these interfaces, that we can achieve with the communication server. For example, the client interface receives the service information, returns the appropriate feedback information. Interfaces specified in the table below:

Table 1 web service interface specification

URL	http://localhost/lvsemoban/interface/getRFID.php		
Parameter type	POST		
Parameters	RFID	classroom	coursetime
Return type	JSON		
Return value	{"RESULT": "0"; "NUMBER": "1"}		
example	http://localhost/lvsemoban/interface/getRFID.php?RFID=10a0b231&&classroom=F205&&coursetime=1		
remark	Return 0: normal; 1: late; 2: wrong number; 3: repeat; 4: wrong classroom; 5: wrong id.		

As can be seen from Table 1, the server provides a service getRFID interface that accepts three parameters, and these parameters will be analyzed and judged. Client sends the appropriate information by visiting the URL http://localhost/lvsemoban/interface/getRFID.php, and the communication format using JSON format.

Server interface functions to achieve this, you need to write a PHP script, which corresponds to the project PHP script as follows:

```
<?php
    //fetch RFID, classroom and course time
    $RFID= $_REQUEST["RFID"];
    $classroom = $_REQUEST["classroom"];
    $courseIndex = $_REQUEST["coursetime"];
    ...
    echo JSON_encode($ret);
?>
```

Fig.3. code of server for request

Server side use the function `$_REQUEST` to obtain the information parameters sent by the client side. After a series of analysis, judgment and after treatment in server, The JSON format corresponding information will return to the client side. In this project, the client sends the collected information from RFID cards for students, classroom name and class time to the server, the server will determine the accuracy of the informations, and then store the processed information to the database. Of course, `JSON_encode` function will return the data encapsulated into JSON format before return.

For users, you do not come into contact with these complex background logics, simply enter the appropriate URL in the browser which can get the appropriate information. With HTML + Javascript, server provides a useful web interface. Meanwhile, the server back through the exchange of information with Web services interfaces for embedded devices, and real-time display on the front page.

4. Embedded Devices Program

We use the Linux + QT software development platform for software development of embedded devices. When the system is to collect information periodically (interval 3s) to access the Web service interface, QT provides us with a http Request program. Among the development of embedded devices, include packaging and sending data.

3.1 Data Package

On the QT platform, we borrowed JSON code to open the package and unpack the data. It provides two functions encode and decode, simply call these two interfaces can operate on the data. Unpack data to map out the container form, which we code the data in the container can be extracted for processing.

3.2 Data transmission and reception

In this project, embedded devices would send all the information to the server after collected, and the server would process and replay the client when receives, By QT signal - to integrate the

process of sending and receiving the data slot mechanism, the system starts at the beginning of the received signal with our QtNetwork handler connected so after receiving the return data to the server, the system can corresponding processing and display, system flow chart below:

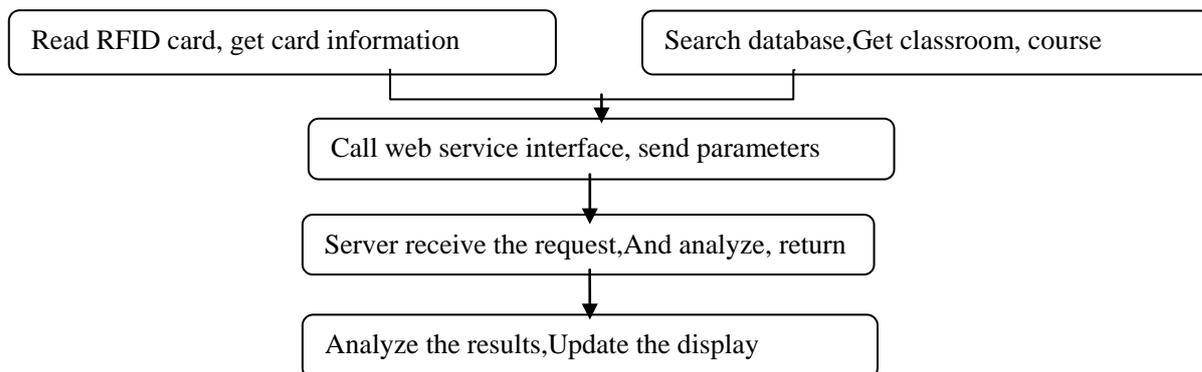


Fig.4. client data transmission and processing

5. Conclusion

This article discusses the implementation and application-based major web services technology embedded web server, so that developers can choose according to the actual resources of the target hardware platform, no longer because of limitations of hardware resources and reduce the performance of web server, while introduction of PHP scripts, making functionality and flexibility of web server greatly enhanced. More practical significance is: not only embedded devices, such as smart phones and many other devices can also communicate with the web server to achieve a multi-platform integration. Currently the program has been on a variety of practical applications open source platform to prove its feasibility in engineering practice.

Acknowledgement

In this paper, the research was sponsored by the Opening Project of Suzhou High-tech Key Laboratory of Cloud Computing & Intelligent Information Processing(Project No. SXZ201301) and the Jiangsu Province Department of Education Reform Project (No. 2013JSJG391).

Reference

- [1] <http://www.apache.org/>.
- [2]Zhang Yuantong, m, Zheng Zhiguo. Transplantation in MicroBlaze system in the PetaLinux operating system.Microcontroller and embedded systems, 2010,10 (7).
- [3] Wei Dongshan. Embedded Linux application development, fully manual. The posts and Telecommunications Press.
- [4] paper Quangang. Principle and application of embedded Linux operating system. Beijing::-Beihang University press,2002:155-169.
- [5] Qin Yunchuan. Building embedded Linux system (Second Edition). China Electric Power Press