The Communication System Based on Node.Js Using Embedded System

Madhuri Sali[1], Mrinal Shidore[2], Ajay Talele[3]
[1, 2, 3] Vishawakarma Institute of Technology

Abstract- *In order to realize the communication between the web applications and the embedded systems, and to improve the efficiency of PC program and embedded system communication, this paper puts forward a solution of building the hyper-text transfer protocol (HTTP) web server combined with transmission control protocol or user datagram protocol(TCP/UDP) socket program using Node.JS platform. This paper also introduces the method of building the webserver on the Express frame work cooperated with the MongoDB database.*

The communication progress and protocol between host computers and the embedded system are illustrated as well.

Keywords- Node.JS; Web server; Embedded System; HTTP;UDP; TCP

I. INTRODUCTION

In the process of many embedded systems, the communication with the host computer is necessary and essential. At present, the mainstream communication mode is universal asynchronous receiver/transmitter (UART) serial port interaction, network data interaction and so on. At the same time, most of the computer software development methods are using programming languages like JAVA, C#, C++ or Lab view to build up desktop applications. While the desktop applications based on the host computer have their larger

Drawbacks:

i)They need complex installation steps; even in the installation process, they often needs to cooperate with the installation of many complex dependent programs or plug-ins.

ii)The user experience of using remote virtual computer to operate the desktop application installed on a remote host computer is not decent. Remote users must create a system user firstly, and then operate the desktop application in an indirect way through the virtual hosts, which is complicated. At the same time, the latency of the whole process is relatively high. Based on the situation above, this article puts forward the concept of Web type host application of embedded systems and the solution of using Web server, based on Node.JS, as the host application to communicate with embedded systems. With the Web application type host, the remote host login to a virtual computer is no longer needed. It leads to the using of a browser to operate the embedded system directly and remotely, which improves the convenience of the whole remote configuration process of embedded systems.

II. WEB TYPE HOST SERVICES

A web hosting service is a type of Internet hosting service that allows individuals and organizations to make their website accessible via the World Wide Web. Web hosts are companies that provide space on a server owned or leased for use by clients, as well as providing Internet connectivity, typically in a data center. Web hosts can also provide data center space and connectivity to the Internet for other servers located in their data center, called colocation.

Web type host application consists of two parts:

i) The backend server program and
ii) the frontend client program.

The backend service program is mainly responsible for handling the user's commands, processing business data logic, communicating with embedded system through the Ethernet, and sending view pages and static resource to the front end clients. The frontend application running on the client device provides the graphic user interface and the simple business logic processing. The backend application is running on the server, while the frontend application is parsed and run on the browser or any other frontend client carriers.

Users can call uniform resource locator (URL) to send commands via the HTTP request, through the client Web application run on the browser, with a method called asynchronous JavaScript and XML (AJAX). After receiving the HTTP request from the frontend client, the backend server application will deal with this request, which operates the database or sends commands to the embedded system or

motivates the burst type data reading from the embedded system according to the business logic.

Several frontend clients can link to the server, at the same time, with many embedded systems linked. This topology enables multiple users to operate a number of embedded systems at the same time.

The connections between the frontend clients and the server can be built through the Ethernet via a lot of transmission methods like wired Ethernet, wireless Ethernet, LTE data, etc.

The convenience of Ethernet and the diversity of its Transmission media make the operation to the embedded systems happens anywhere and anytime possible.

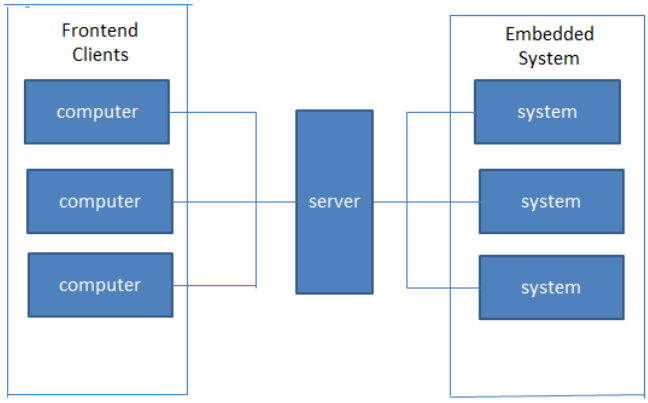

Figure 1. Topology of web type host application

III. THE PRINCIPLE AND IMPLEMENTATION OF THE WEBSERVER

Web server refers to server software, or hardware dedicated to running said software, that can serve contents to the World Wide Web. A web server processes incoming network requests over the HTTP protocol. The Web server application is developed on the Node.JSplatform. Node.JS is a development platform based on the Chrome V8 JavaScript engine. On this platform, with Node Package Manager (NPM), through the open source community and other channels of functional expansion, Node.JS has formed a very powerful and more comprehensive functional ecological group. Users can use the NPM to search for the required Node.JS functional modules for integration.

A. Web Server Application Framework

Embedded systems have traditionally been isolated, self-contained systems; at most, they might have communicated with other systems within a limited range on a local network. This is no longer the case: embedded systems — especially small, very deeply embedded devices — increasingly use the Internet as a way to communicate with each other and with the people managing them. Using the Internet to control an embedded system has a direct impact on the cost of building and maintaining the system. Physical controls like buttons mean additional components and cost as well as a design that must accommodate access to the controls. Even more importantly, embedded systems are increasingly being deployed in remote locations. In those applications, the cost isn't dominated by components, but rather by the people that must go out to maintain and operate the system. This makes remote access critical to the cost-effectiveness of the system. The Internet is the most straightforward way of getting to the target system because worldwide infrastructure already exists.

The design pattern for the Express framework based onNode.JS is the classic: MVC (model, view and controller)design pattern. The model module is responsible for associating the backend server program with the database and abstracting the data in the database as a data structure object that the program can directly operate. This enables the whole application to achieve a more efficient, convenient and safe data operation. The view module is responsible for integrating the data and functions provided by the controller module to provide a visual view and interface to the frontend client.The controller module is responsible for operating the dataor database through the model module, receiving the client's request and integrating it into the system business logic operation, communicating with the embedded system through the peripheral interface (such as serial port, network, etc.), and providing the required data to the frontend client.

B. Implementation of UDP Server and Embedded Device Interaction

This article uses the DGRAM package in the Node.J Secological chain to create the socket server object for the UDP server. Because only one UDP server is required in the backend server, the UDP server class object is constructed using the singleton schema, and the UDP socket is used globally for this object. Firstly, create the UDP socket, through the code:dgram.createSocket ('udp4'). After that, use the bind command to bind the corresponding port. Then the listening and the data sending on the UDP server port is available. Since this UDP server object needs to resolve the network commands of the embedded systems, reply to the data, and call the embedded device controller to record it into the

database, it is necessary to provide the Set and Get interfaces for the analyzed UDP packet results, so that the controller can use those interfaces to configure the remote UDP devices or record the UDP data stream history.

C. The Process from commands sent by the client to embedded system completing the required operation.

Since the request sent is an HTTP request sent by the client, the lifetime of the request is only valid in one round of the HTTP request and response. Therefore, in this life period, it is unreasonable and unstable to obtain or set the dynamic state of embedded devices directly.

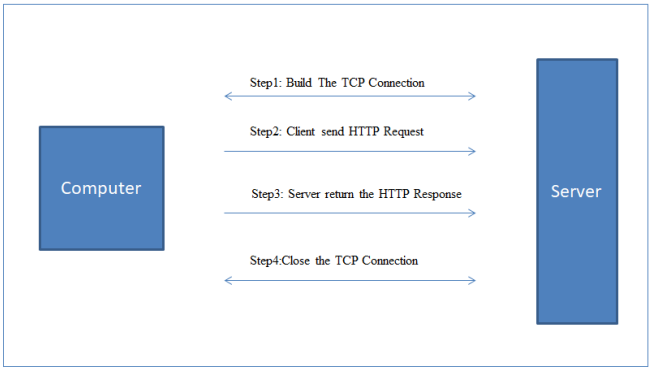

Figure 2. The progress of a HTTP request and response

IV. SERVER AND EMBEDDED DEVICE COMMUNICATIONPROTOCOL

Because UDP is an unreliable transmission protocol, the process of data transmission may produce errors. Therefore, this paper puts a protocol to make the data command has a higher error correction, and the embedded systems can resolve the commands package more easily and effectively. With this communication protocol, more complex commands with parameters can be transferred, accepted and executed. What's more, the commands themselves are not the only issues to be considered, but the requirement of security, leading to the user access controlling and checking, is also introduced into this protocol.

V. CONCLUSION

The remote operation center and a lighter weight local application on branch side host computer are becoming the trends. Most users want to get access to the control system in any place, at any time and even on any devices without installing any additional applications. The web application interface works with a backend server is a good choice for most users and using conditions.

So, at present, bulky desktop applications are gradually being replaced by more lightweight, flexible Web applications. The proposed approach to the HTTP Web application in this article, not only allows the user to use the browser to get easier method to view, access, change the embedded device, but also provides a cluster management system can be embedded.

As the Internet of things and wireless sensor network play a huge role in this era, which means a more convenient and effective embedded device management system, and a more comprehensive cluster of embedded node device management system, will have a huge economic value and social benefits and also being attractive. With the development of the public communication method like 4G long term evolution (LTE) and the 5G technology in the future, the embedded system can be controlled by the web application host computer easier. That is also the reason why the communication system put by this paper is promising.

REFERENCES

[1] L. Liang, L. Zhu, W. Shang, D. Feng and Z. Xiao, "Express supervision system based on NodeJS and MongoDB," 2017 IEEE/ACIS 16thInternational Conference on Computer and Information Science (ICIS),Wuhan, 2017, pp. 607-612.

[2] E. Stark, P. Bistak, S. Kozak and E. Kucera, "Virtual laboratory basedon Node.js technology," 2017 21st International Conference on ProcessControl (PC), StrbskePleso, 2017, pp. 386-391.

[3] B. Pfretzschner and L. b. Othmane, "Dependency-Based Attacks onNode.js," 2016 IEEE Cybersecurity Development (SecDev), Boston,MA, 2016, pp. 66-66.

[4] S. Farah, A. Benachenhou, G. Neveux, D. Barataud, G. Andrieu and T.Fredon, "Flexible and real-time remote laboratory architecture based onNode.js server," 2015 3rd Experiment International Conference(exp.at'15), Ponta Delgada, 2015, pp. 155-156.

[5] B. Carter, "HTML Educational Node.js System (HENS): An AppliedSystem for Web Development," 2014 Annual Global Online Conferenceon Information and Computer Technology, Louisville, KY, 2014, pp.27-31.

[6] T. Bosák and K. Žáková, "Node.js based remote control of thermoopticalplant," Proceedings of 2015 12th International Conference onRemote Engineering and Virtual Instrumentation (REV), Bangkok,2015, pp. 209-213.

[7] K. Lei, Y. Ma and Z. Tan, "Performance Comparison and Evaluation ofWeb Development Technologies in PHP, Python, and Node.js," 2014IEEE 17th International Conference on Computational Science andEngineering, Chengdu, 2014, pp. 661-668.

[8] M. F. Karagoez and C. Turgut, "Design and Implementation of RESTfulWireless Sensor Network Gateways Using Node.js Framework,"European Wireless 2014; 20th European Wireless Conference,Barcelona, Spain, 2014, pp. 1-6.
[9] S. Tilkov and S. Vinoski, "Node.js: Using JavaScript to Build High-Performance Network Programs," in IEEE Internet Computing, vol. 14,no. 6, pp. 80-83, Nov.-Dec. 2010.