



Development of Http Server for Remote Data Monitoring and Recording system.

D.Joseph Sunder Singh¹, Mrs.L.Padmalatha²

¹M.Tech (Embedded Systems), Gudlavalleru Engineering College, Andhra Pradesh, India.

E-mail id: josephle401@gmail.com

²Associate Professor of ECE Department, Gudlavalleru Engineering College, Andhra Pradesh, India.

E-mail id: sruthivinay04@gmail.com

ABSTRACT

This paper focuses on development of remote data monitoring and recording system in industries. Now a days we are using many Networked embedded systems for monitoring and control the home or industrial devices. These low cost devices are capable of reporting and receiving information in just the same way that computers on a network. The advent of Internet communication standard, TCP/IP, offers significant potential in terms of remote monitoring and management of construction sites using embedded systems. In this paper the design method of low cost system of remote data monitoring, and recording is designed based on ARM. For this a small HTTP SERVER is built in LPC2148 and it is connected with the remote monitoring terminal through Ethernet. The data can be stored in the SD Card via SPI interface. At the same time, for the sake of the versatility, the FAT file system is built in the SD Card. So, an Ethernet-enabled remote data monitoring system with the ability of data recording is built. This design is having advantage of cost-effective, easily realized, stable and reliable transmission. It can be connected to the INTERNET or LAN through TCP/IP protocol. FreeRTOS is used as an operating system running on ARM processor, an industrial grade RTOS for hard time applications. By this design the data is sent without a PC and system favour's large scale data acquisition system.

Indexing terms/Keywords

Remote data monitoring and Recording, Control, ARM, FreeRTOS, FatFS, TCP/IP protocol Stack

Council for Innovative Research

Peer Review Research Publishing System

Journal: International Journal of Computers & Technology

Vol 11, No.4

editor@cirworld.com

www.cirworld.com, member.cirworld.com



1. INTRODUCTION

Industry analysts see embedded Internet systems as poised for rapid growth in the manufacturing sector in the next few years. To date, much of the research work pertaining to the Internet in construction, has focused on person to person, or computer to computer communication. Embedded Internet systems allow users to communicate with other items such as plant, machinery, and manufactured components delivered to site. This communication may range from passive remote monitoring to direct control or resetting. Embedded systems contain a computer or a computer-like device that is used to control the operation of plant, machinery and equipment. They have proven very useful and commercially successful in the manufacturing sector but have been little used in construction, partly because of the temporary and itinerant nature of the construction process. Embedded systems are found in engineering tools and process control systems. They are increasingly used in office and consumer products, such as cars, washing machines, photocopiers and bedside alarm clocks. The continued fall in cost of these devices, now justifies their use on disposable items.

With the increase of these electronic products, new problems about data transmission and equipment management arise. In the process of environment remote monitoring, the spot collecting areas are irregular, the communicating conditions are weak and the data obtaining and auto-collecting are difficult, so more effective monitoring, control and higher level equipment management are necessary. This paper proposes the solution for above remote application issues: the monitoring and recording terminal is designed and connected to the PC through Ethernet. At the same time, we can connect multiple monitoring terminals to the Ethernet, because the Ethernet is a kind of standard open network based on TCP / IP and it has the characters of commonality, easily networking, remote data transmitting. Figure 1 show that PC can monitor at real time and manage unitively the sporadic data collecting spot simultaneously. Moreover, these terminals are equipped with good ability of individual data recording.

DATA-ACQUISITION systems are in great demand in industry and consumer applications. In some applications, human beings have been replaced by unmanned devices that will acquire data and relay the data back to the base. There are data-acquisition and control devices that will be a substitute for a supervisor in a multisite job operation. A single person can monitor and even interact with the ongoing work from a single base station. In these applications, data are compiled in a central server and are then served to the clients via the Internet. The client framework is in a central server and has all the applications. A person that needs to access any data must first access the server. An indirect access to the data-acquisition unit makes the system unattractive for real-time control applications, where direct interaction with the system may be required.

2. SYSTEM ARCHITECTURE

The embedded system contains a microcontroller, which is the heart of the system. The microcontroller is connected to Ethernet controller which is responsible for the sending & receiving the messages through the internet. The RJ-45 jack connector connects the embedded system with the internet line. Different types of sensors/transducers may be incorporated in the system to read the physical parameters like temperature, pressure, humidity, or any type of signal which need to be read. The sensors/transducers may provide analog or digital outputs. The analog signals have to be converted in digital signal using Analog-to-Digital Converters which is interfaced with microcontroller. The microcontroller read-out these digital signals & compare with the user set values.

An embedded system is a computing system that is designed to solve a specific problem. These systems usually include one or more microprocessors, some I/O devices, and some memory—both RAM and ROM. As opposed to general-purpose computers, the software that embedded systems run is static, and it is sometimes referred to as firmware. The embedded system, including the firmware, must be carefully designed, because any mistake may require a recall. It is also important to minimize both the manufacturing and the operating costs of the system. This is achieved by minimizing several aspects of the design, such as the die area, the amount of memory, and the power consumption. These are the defining characteristics of an embedded system.

In this project, we will build one data monitoring and recording system based on LPC2148. The NXP (founded by Philips) LPC2148 is an ARM7TDMI-S based high-performance 32-bit RISC Microcontroller with Thumb extensions 512KB on-chip Flash ROM with In-System Programming (ISP) and In-Application programming (IAP), 32KB RAM, Vectored Interrupt Controller, Two 10bit ADCs with 14 channels, USB 2.0 Full Speed Device Controller, Two UART's, one with full modem interface. Two I2C serial interfaces, Two SPI serial interfaces Two 32-bit timers, Watchdog Timer, PWM unit, Real Time Clock with optional battery backup, Brown out detect circuit General purpose I/O pins, CPU clock up to 60 MHz, On-chip crystal oscillator and On-chip PLL.

2.1. Embedded HTTP Server Architecture

An embedded web server is an ARM processor that contains an internet software suite as well as application code for monitoring and controlling machines/systems. Embedded web servers are integral part of an embedded network. Figure 1 shows the proposed concept of DACS with embedded web server on a single chip module. This is a single hardware it contains RTOS portable ARM processor. ARM processor is the responsible part for measuring signals and controlling the devices remotely. ARM processor is the responsible for measuring signals and controlling the devices remotely. The system works in the DACS mode for measuring and embedded web server mode for sharing the data with the clients. The real time operating system manages all the tasks such as measuring signals, conversion of signals, data base up-dation, sending HTML pages and connecting/communicating with new users etc.

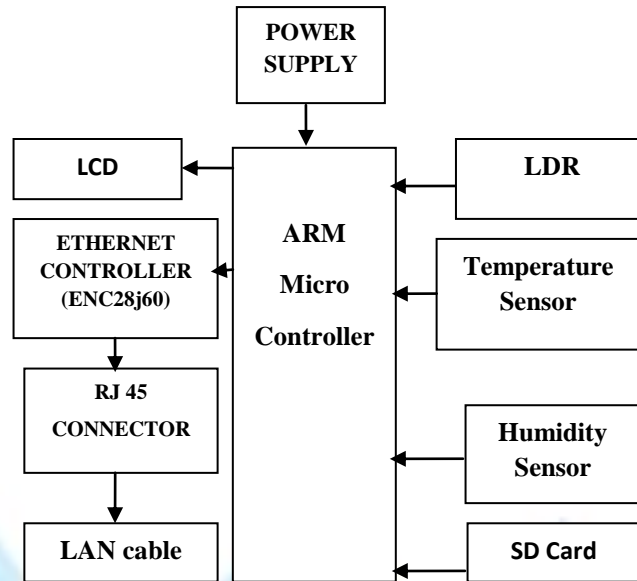


Figure 1: Data Acquisition System.

3. SYSTEM SOFTWARE DESIGN

The real time operating system manages all the tasks such as measuring signals, conversion of signals, data base up-dation, sending HTMLpages and connecting/communicating with new users etc., The RTOS manages all the required tasks in parallel and in small amounts of time. Embedded HTTP server has different requirements, such as low resource usage, high reliability, security, portability and controllability for which general web server technologies are unsuitable. The RTOS manages all the required tasks in parallel and in small amounts of time. RTOSes affect the real-time system development process in numerous ways. Some of the effects include hardware abstraction, multitasking, code size, learning curves, and the initial investment in the RTOS. None of these factors should be taken lightly.

FreeRTOS as the open-source light weight real-time operating system, achieves the management on the realtime dispatching, semaphore, alignment and storage and relieves from the authorization fees. To adopt the embedded real-time operating system (RTOS) can achieve the aim of multitask, which can take use of the CPU resource more reasonable and more effective. At the same time, it can simplify the design of the utility software and shorten the time of system exploit, which will ensure the real-time and reliability. The realization of FreeRTOS is constituted by documents of list.c, queue.c, croutine.c and tasks.c. Figure 2 shows list.c is the realization of the linked list supporting the dispatcher of the internal core; queue.c is the realization of the alignment, supporting the discontinued environment and semaphore control; croutine.c and task.c are the realization of the organization form of the task. As to croutine, all the task share one storage place, which reduce the demand of RAM and limit strictly on its application. Task is the traditional realization, all the task apply their own storage place, supporting the Preemptive dispatching.

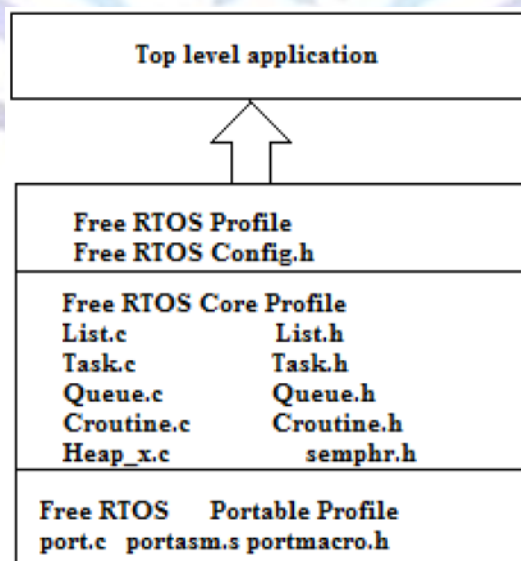


Fig.2: FreeRTOS configuration.

3.1 SYSTEM DESIGN FLOW

Figure 3 shows the basic block diagram of the embedded web server. The necessary requirement analysis is done on the RTOS, web server, TCP/IP and the target board. Based on the requirement analysis, the FreeRTOS is configured, as per the target board. The target is tested along with the RTOS, by booting it with the suitable RTOS. The application code for a specific RTOS is written and both the RTOS and web server are ported on the target. The web server application is tested for our desired function.

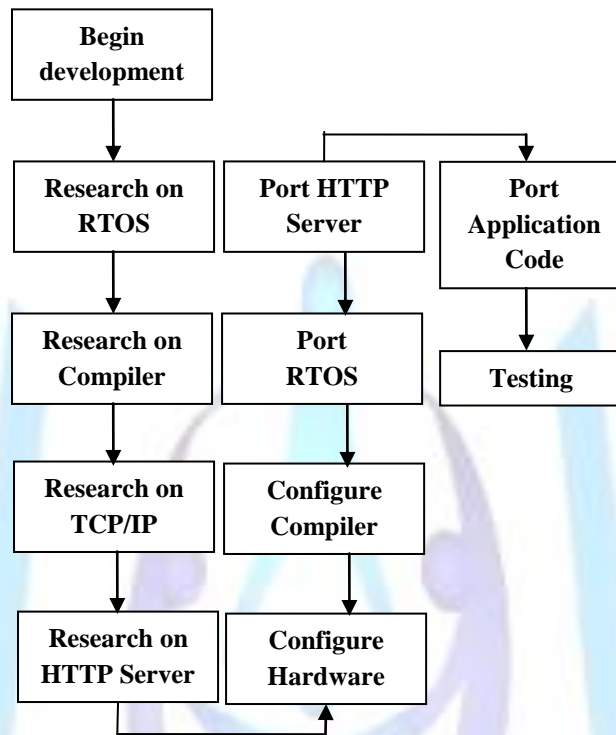


Fig. 3 Basic Design of the System

3.2 HTML PAGES

These are used for data communication between the client and the server. In the embedded web server, web pages are selected as the media of interaction. HTML is used in designing these web pages.

4. FAT DOCUMENT SYSTEM TRANSPLANT

Fat document system introduction:

After getting sensors values there is a need to store these values for maintaining a record. SDCard with its easy-to-be-carried character is chosen in this design. But document system should be built to realize the independence reading and writing on SDCard. The data will be stored according to the FAT document system format. This FAT document system which is free of charge on open source is especially used on embedded system exploit. It is written by standard ANSI C and separated from storage medium driver. The feature is that it is compatible with the FAT document system of Windows and has the individual hardware platform and easy to be transplanted. System structure is shown as Figure 4.

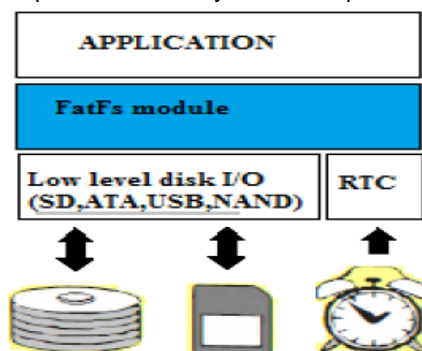


Fig.4. Fat document system structure



5. TESTING EMBEDDED WEB SERVER

Initially, the target is tested for the working of operating system by using WINARM compiler. The compilation of RTOS is shown in Figure 5. Now the embedded web server is responding to the clients, request is made to the server, embedded web server, by typing the IP address of the server in the client's browser. The user has to enter '10.1.1.27' IP to access the server. This request is taken by the operating system of the client and given to the LAN controller of the client system. The LAN controller sends the request to the router that processes and checks for the system connected to the network with the particular IP address. If the IP address entered is correct and matches to that of the server, a request is sent to the LAN controller of the server and a session is established and a TCP/IP connection is established and the server starts sending the web pages to the client. The client requested web page with sensors reading is shown in Figure 6.

```
C:\WINDOWS\system32\cmd.exe
g -D WINARM -D RUN_MODE=RUN_FROM_ROM -D GCC_ARM7 -D ...
t -D /test/FreeRTOS/include -mcpu=arm7tdmi -Tlpc2148-rom.ld -g -O3 -fomit-frame
arm-elf-ar -rc d:/test/common/common.a uart@ISR.o
-ld /test/FreeRTOS/include -mcpu=arm7tdmi -Tlpc2148-rom.ld -g -O3 -fomit-frame
-ld /test/FreeRTOS/include -mcpu=arm7tdmi -Tlpc2148-rom.ld -g -O3 -fomit-frame
make[1]: Leaving directory 'D:/test/uart/'
D:\test>
```

Fig.5.Compilation of FreeRTOS

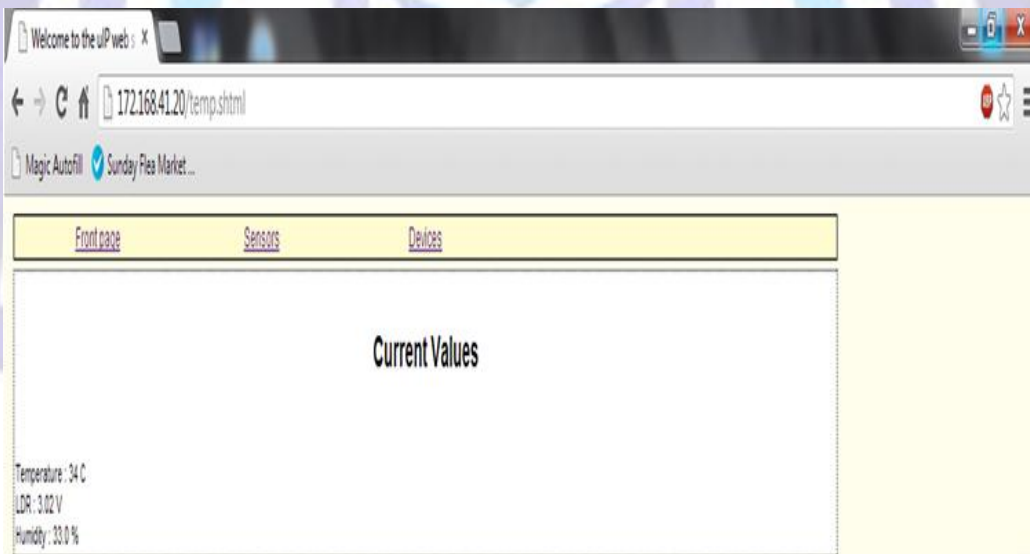


Fig.6.Client requested Web page

6. CONCLUSION

The system, whose design is based on Embedded c, ARM and has a building-block structure and intelligent function modules. In terms of software design, it supports any combination of system's function modules. So, there is no need to change the original hardware and software design, just add sensors to the system and conduct simple software upgrade/configuration.



REFERENCES

- [1] ZLG, etc., Cortex-M3 Development Guide · Based on LM3S8000 [M],Guangdong: Guangdong Zhiyuan Limited ,2008:87·
- [2] Xin-Feng Xiao, Song Qiang, Wang Lixin, etc .. Protocols and networkmanagement standard tutorial [M]. Beijing: Tsinghua University Press ,2007:150-161
- [3] The uIP Embedded TCP/IP Stack The uIP 1.0 Reference Manual [Z]. AdamDunkels, Swedish Institute of Computer Science, 2006:34-37
- [4] 347 Fred Halsall, Data communication, Computer Networks and OpenSystems [M],beijing: Machinery Industry Press, 2004, pp.572-580
- [5] Labrosse J ean IJCIOS-I I-Open source real-time embedded operating system system [M].beijing : China Electric Power Press , 200l.pp.l25-132
- [6] RTOS Evaluation Project, —What makes a good RTOS Dedicated SystemsExperts, 2001.
- [7] The ARM Processor website. [Online].
[Available: http://www.friendlyarm.com](http://www.friendlyarm.com)
- [8] RTOS website. <http://www.freertos.org>
- [9] ARM website. <http://www.arm.com>.

Author' biography with Photo



D. Joseph Sunder Singh¹ obtained his Bachelor's Degree from SRK Institute of Engineering & Technology, Vijayawada. His Areas of interest are Microprocessors, Microcontrollers, Embedded System Design, Digital systems and VLSI CAD tools. Presently he is doing M.Tech in embedded systems at Dept. of ECE, Gudlavalleru Engineering College, Gudlavalleru, Andhra Pradesh, India.



Mrs. L. Padmalatha² obtained her Bachelor of Engineering Degree in Electronics and Communication Engineering from MADURAI KAMARAJ UNIVERSITY & Master's degree in Digital Electronics and Communication Systems from JNTUH, Areas of interest are Signal and Image Processing, Microprocessors, Microcontrollers, Digital system Design, Embedded System Design. She published 05 technical papers in international journals/conferences and having 7 years of teaching experience. She is a professional member of IEEE and IETE. Presently working as Associate Professor of ECE Department, Gudlavalleru Engineering College, Gudlavalleru, Andhra Pradesh, India.