Embedded Web Server Based on DAC System Using ARM

S.A.N.Sandeep
M.Tech Student
PKSK College
ECE Dept

P.Malyadri
Assoc Prof
PKSK College
ECE Dept

Abstract:
Now a days we are using many Networked embedded systems for monitoring and control the home or industrial devices.With the scalable networking solution The server enables Web access to distributed measurement/control systems and provides optimization for educational laboratories, instrumentation, Industrial and home automation. In this paper , we present the principles and to design a system for Internet-based data-acquisition system and control by using Advanced RISC Machine i.e ARM processor and in-build web server application with General Packet Radio Service (GPRS) technology. The main core of the system is an embedded hardware running on a NUT OS, a industrial grade RTOS for hard time applications. The embedded device communicates through General Packet Radio Service (GPRS),and also help us to communicate with the GSM as most of us were communicating with GSM which makes it accessible from anywhere in the world through a web server built into the embedded device.The proposed system eliminates the need for server software and maintenance. The proposed system minimizes the operational costs while operating with a large amount of data.
Key Terms: GPRS, GSM, Real Time Processing, Dataacquisition**.**

I. INTRODUCTION

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 In some applications, human beings have been replaced by unmanned devices that will acquire data and relay the data back to the base [1]. 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. An acquisition unit designed to collect data in their simplest form is detailed in [2], which is based on Linux [3], which is a popular choice for embedded PC systems. A similar system in [4] provides data acquisition with no concern for remote access. 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. The need to maintain an additional server will also increase the setup costs and the costs to maintain the acquisition systems, such as regular maintenance costs, system updates, etc. Therefore, the central server has to be eliminated for a real time system. The closest to this idea is published in [5]. In this system, a reliable bidirectional Point-to-Point Protocol (PPP) link for real-time control and surveillance via a GSM network is formed. However, there is still no effort to minimize the operational costs. In addition, this system is based on an industrial PC, thus making it an expensive solution. Interaction with the embedded unit is also an important issue. In [6], an embedded PC card placed on the Internet allows limited interaction through commands sent through Transmission Control Protocol/IP (TCP/IP) and User Datagram Protocol. In this paper, we propose a GPRS based portable low-cost data-acquisition system, which can establish a reliable bidirectional connection for data acquisition. The proposed system uniquely reduces the Costs occurring from frequently requested data and eliminates the need for a well-established server. The system uses a dummy server for static information, thus optimizing the transfer of large data. The user can directly log in and interact with the embedded device in real time without the need to maintain an additional server. The system is modularly built, allowing different modules to be added. In addition, it is flexible to accommodate a wide range of measurement devices with appropriate interfaces.

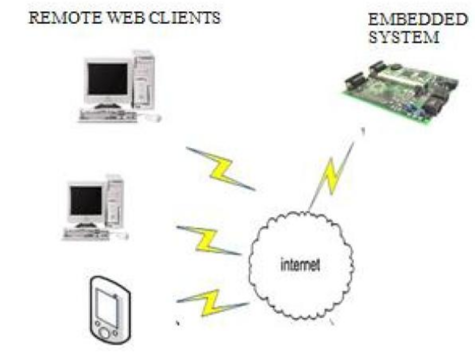

Fig 1. System architecture

In Section II, we will introduce the details of the aforementioned system. In Section III will describe the Testing of Embedded Web Server. In Section IV data acquisition sensors are connected to the embedded system to demonstrate its operation and the results are presented. Section V presents the conclusion.

II. INTERACTIVE DATA-ACQUISITION SYSTEM

Interactive Internet-based systems provide a way to monitor and adjust using standard web browsers and a PC. The target systems can be monitored and controlled independent from the location and the platform since standard web browsers can be used on the client side. A typical data-acquisition system is connected to web clients via the Internet, as shown in Fig. 1. The data acquisition system needs to relay the acquired information to the requesting clients. Digitally acquired data are stored in web server's data base. Whenever the client wants to access data, it sends the request to server; this request is taken by the router, which is connected to the internet. The web processes the request made and finally connects to the desired web server, access the requested data and sends the data to the client.

Embedded Web 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 [7]. Fig 2. 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. Measurements can be done by DACS mode and the data are shared with clients through embedded web server by embedded web server mode. The real time operating system manages all the tasks such as measuring signals, conversion of signals, data base up-dation, sending HTMLpagesandconnecting/communicating with new users etc.,

The RTOS manages all the required tasks in parallel and in small amounts of time. Web based management user interfaces using embedded web server have many advantages: ubiquity, user-friendly, low-development cost and high maintainability. Embedded web server has different requirements, such as low resource usage, high reliability, security, portability and controllability for which general web server technologies are unsuitable.

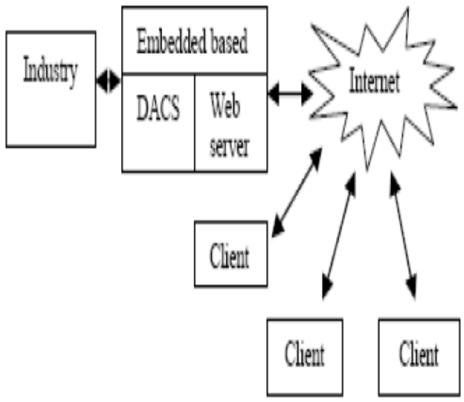

Fig. 2. Embedded Web-Server architecture

A. Establishing a Direct Communication Link between the Client and the Embedded Device:

The use of GPRS is well known to everybody and almost all service providers on GSM are giving this service. GSM and GPRS [8] are developed for cellular mobile communication. Hence it is very much easy to be getting connected to the Internet world. Once a GPRS connection has been established queried data can be relayed to the client via a central server[8]. Using a central server to relay the acquired data has some disadvantages. First, a central server needs a client interface framework. An additional data transfer corresponds to time delays before the data are made available to the client. In addition, since the server acts as a relay, no direct bidirectional communication between the client and the embedded system can be established. This makes the system unsuitable for real-time control applications.

The basic idea behind real-time processing is that the embedded system is expected to respond to the queries in time. Real time should be fast enough in the context in which the system is operating and reliable as well. Real-time system correctness depends not only on the correctness of the logical result of the computation but also on the result delivery time [9]. This method also increases the data transfer cost as the number of clients increases due to the access amount of data transfers via GPRS. Direct communication, on the other hand, enables access to only relevant information in the embedded system by pre processing the data. The embedded system should also handle the web services. This eliminates the need for a central server and reduces the amount of data sent from the remote unit since only the queried data will be transferred. In the proposed system, the GPRS architecture and protocols are compliant with [8]. This system is configured to be virtually online at all times in a GSM network. An admin script is executed after the boot of the operating system, initiating the GPRS connection software module. A PPP connection is established by a GPRS modem that works at 900/1800/1900 MHz operating frequencies. A PPP daemon (PPPD) is used to manage the PPP network connections between the client and the embedded module.

The PPPD is responsible for setting up the GPRS parameters, such as the connection speed and compression. To directly access an embedded system, the IP address of the embedded device should be made available to the client side. There are two choices available. A static (hard-coded) IP could be used, or the remote device should initiate a connection by reporting its IP. This choice is quite straightforward and simple. Although the usage cost remains unchanged, it requires a static IP setup by the service provider and involves monthly recurring costs. The static IP is preferred for its simplicity in designing a system; however, its overhead may be impractical. The other choice is to use a dynamic IP assigned through a Dynamic Host Configuration Protocol (DHCP) server of the GSM provider for every connection established. However, this IP needs to be known by any client requesting an access to the embedded server. One solution is to broadcast this IP to a dummy FTP server. The FTP server is a dummy server and does not require regular software updates or maintenance. The folder structure of the FTP server is shown in Fig. 3. A script on the embedded device is configured to update its IP address on the FTP server in Hypertext Meta-Language as an index.htm file, under a folder uniquely named by its hostname. This script simply parses the current IP for that embedded device and sends an html file with the IP information of the embedded device to the FTP server.

Once this file is in place, a direct connection can be established with the desired embedded device by a simple query. An example embedded system, named DAS (Fig.3), can be queried from the FTP server by a simple command. The web browser processes the (index.htm) file in the specified folder as default; therefore, a file name is not needed for referencing.

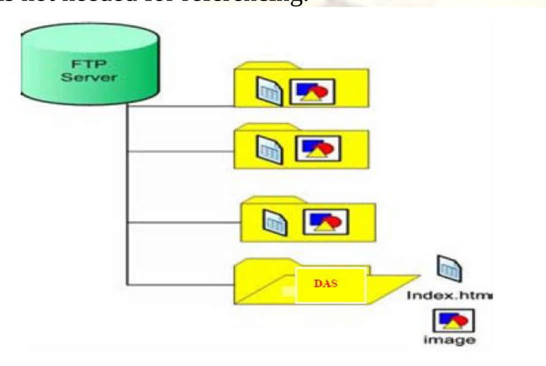

Fig. 3.A script on the embedded device

This process is illustrated in Fig. 4. The DHCP approach is more flexible and works better compared with the static approach as a cost-effective solution, despite the necessity for a script running on the embedded server, one-time broadcasting its IP to the FTP server. The hypertext file placed on the FTP server by the embedded system and queried by the client is shown in Fig. 5. With this mechanism in place, the embedded system updates it IP information on the FTP server upon every reboot, which causes an IP refresh from the GSM service- provider

B. Data Management in the System

The Internet server is used to decrease the management costs by sending all the pictures (logo, picture, bar graphics, etc.) to the client through a server on the Internet. Text data such as coordinates, temperature, and altitude are served from the embedded system. If bulky data are going to be sent, the embedded module is set to send the image only once via GPRS and placed on an FTP server. This approach eliminates the transfer of large data through GPRS more than once, thus reducing the transfer costs, particularly if more than one client is involved or multiple requests to the same data are needed, as shown in Fig. 6.

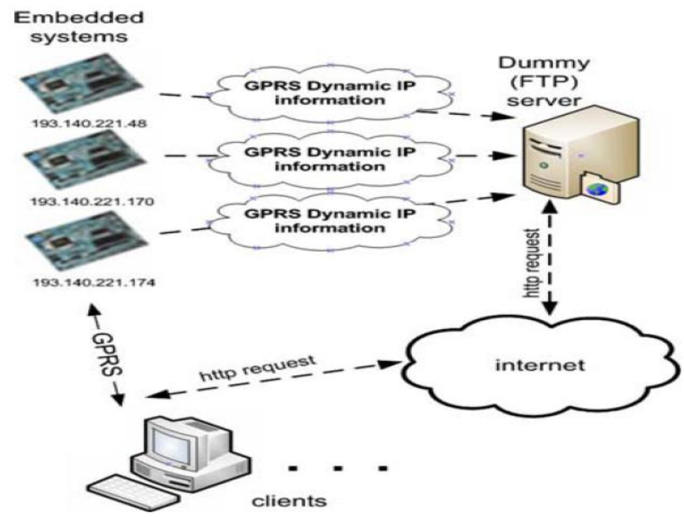

Fig. 4. IP address lookup for the stored IP.

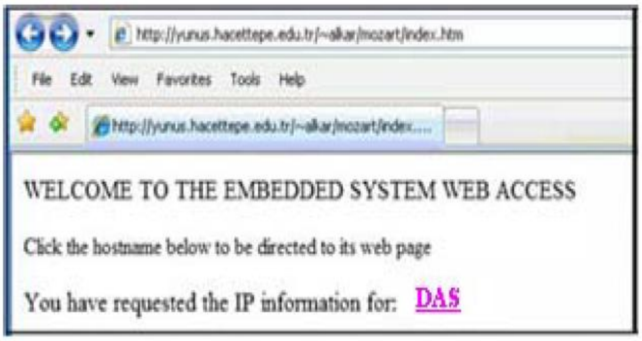

Fig. 5. Index.htm on the FTP server as viewed by the client.

C. Hardware

The general hardware structure of the remote I/O data acquisition and control system based on ARM processor is shown in Fig 7. The remote I/O data acquisition and control system based on embedded ARM platform has high universality, each acquisition and control device equipped with 24-way acquisition/control channels and isolated from each other. Each I/O channel can select a variety of electrical and non electrical signals like current, voltage, resistance etc., Digital acquisition are done by special ADC. The measured data are stored in external memory in which the memory is act as a data base during web server mode. The ARM processor directly supports the Ethernet service and RS485 communication. Hence the data has been stored and controlled by some other PCs or network via RS485 and Ethernet. ARM processor has internal I2C module. So it has the ability to communicate with any other peripherals. A Data acquisition module, and a GPRS are integrated into the embedded board to form a sample application, as shown in Fig. 8. **DAS**

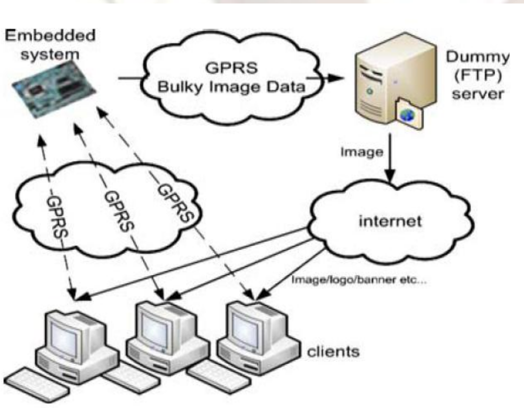

Fig. 6. Data management in the proposed system

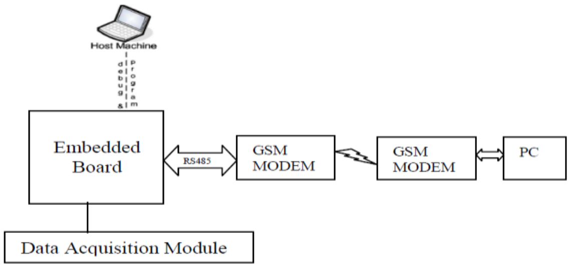

Fig. 7. Block diagram of the embedded system with sample devices attached.

D.Software and Operating System Choice

Nut/OS [10] is an embedded open-source operating system for microcontrollers, licensed by egnite software [11] under the BSD license. It is allowed to modify the source code and chip binary software, as long as the original copyright notice is included. Nut/OS offers many features to the programmer like dynamic memory allocation, threading, formatted output and many more. But to ease portability of the stack itself as few Nut/OS specific features as possible were used. The operating system also offers a modular design, so that only needed parts are linked together and no unneeded parts are loaded into the limited program space of the host. Another advantage is the support for cooperative multi threading. The application itself runs in a separate thread and there is no need for the application programmer to cope with events, dispatching and other confusing techniques. Multiple threads can wait on a queue and block until an event is posted to this queue. Then either one or all threads wake up. Nut/OS also supports dynamic memory management to allocate and free memory on the heap, although this is not used to ease portability. Timer support and stream I/O functions are also supported. In the Nut/OS part serial device drivers and a PPP stack is included. Additionally Nut/OS features a Nut/Net part, including a fully functional TCP/IP stack providing ARP, IP, UDP, ICMP and TCP protocol, automatic configuration via DHCP, a HTTP server with CGI functions and TCP and UDP socket API for the application programmer.

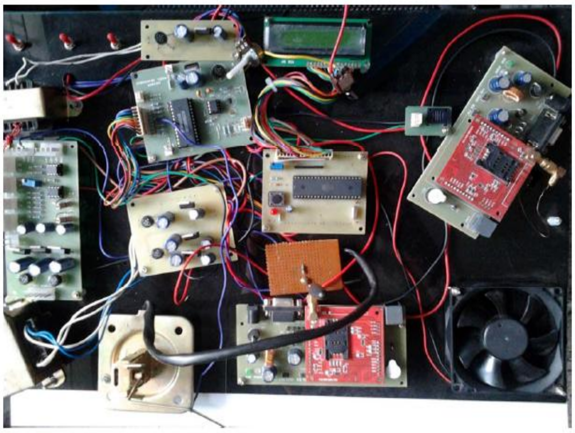

Fig. 8. Components of the embedded system.

III.TESTING EMBEDDED WEB SERVER

Initially, the target is tested for the working of operating system. This is done by booting the target using the hyper terminal. After the target is successfully booted with Nut/OS, it is tested over the network using ping command. 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 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 establishes and the server starts sending the web pages to the client.

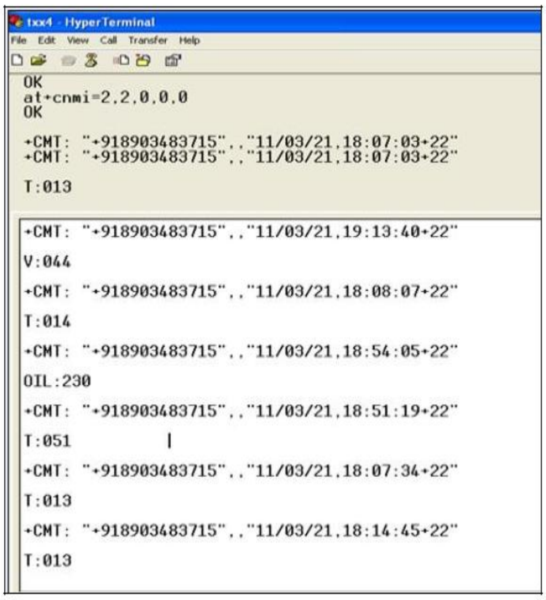

Fig. 9. Raw Data Acquisition Module data sample using hyper terminal

III. RESULTS AND DISCUSSIONS

Fig 10 and 11 shows execution results of ARM Embedded web server based on DACS system.

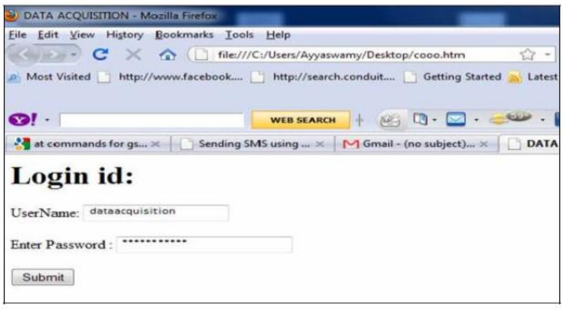

Fig 10. On-line processing web page

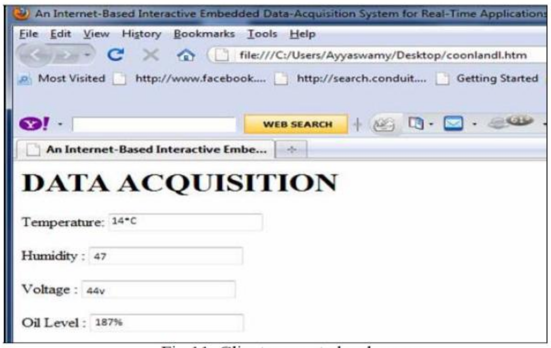

Fig 11. Client requested web page
(Issued by ARM web server)

Above showed web pages are requested by the client and served by the embedded web server which is ported on ARM9 processor. Client can interact with the machine through its own browser via these embedded web pages. Every client's control has been executed in industry via the embedded web server.

IV. CONCLUSION

In this application, a low-cost, Internet-based data acquisition and control system has been designed and implemented that should find interest from researchers. The application possibilities are virtually unlimited by attaching modules with appropriate interfaces, although the usage of the system is demonstrated with only a few sample devices. Compared with other applications, this system has advantages in terms of allowing direct bidirectional communication and reducing overhead, which can be vitally important for some real-time applications. The operational costs have been reduced by relinquishing the storage of large data to an FTP server on the Internet. The system is designed to support both static and dynamic IPs. A method to distribute the IP information has been developed. This cost-minimization effort is a big concern for mobile systems using wireless communication methods and has not been discussed before. The overall cost advantage of the system in terms of the components used makes it an attractive choice for data-acquisition applications. The power demand of the device is still in the process of being improved by putting the attached devices into sleep mode at times when they are not in use to conserve power.

REFERENCES

[1] C. E. Lin, C.-W. Hsu, Y.-S. Lee, and C.C.Li, "Verification of unmanned air vehicle flight control and surveillance using mobile communication,"J. Aerosp. Comput. Inf. Commun., vol. 1, no. 4, pp. 189–197, Apr. 2004.W.-K. Chen, Linear Networks and Systems (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.

[2] K. Jacker and J. Mckinney, "TkDAS—A data acquisition system using RTLinux, COMEDI, and Tcl/Tk," in Proc. Third Real-Time Linux workshop 2001. [Online]. Available: The Real Time Linux Foundations: 2001/papers.html

[3] E. Siever, A. Weber, S. Figgins, and R. Love, Linux in a Nutshell. Sebastopol, CA: O'Reilly, 2005.

[4] Q. Zhou, W. Wu, and Y. Ma, "The embedded data acquisition system for Mössbauer spectrum," in Proc. Third Real-Time Linux Workshop Embedded Linux Expo Conf. Real-Time Embedded Comput. Conf., Milan, Italy, Nov. 2001, pp. 26–29.

[5] C. E. Lin, C.-C. Li, A.-S. Hou, and C.-C.Wu, "A real-time remote control architecture using mobile communication," IEEE Trans. Instrum. Meas.,vol. 52, no. 4, pp. 997–1003, Aug. 2003.

[6] T. Motylewski, "The industrial data-acquisition system with embedded Rt-Linux and network server technology," in Proc. Third Real-Time Linux Workshop, 2001. [Online]. Available: The Real Time Linux Foundation: http://www.realtimelinuxfoundation.org/events/rtlws-2001/papers.html

[7] Klimchynski, "Extensible embedded Web server for internet-based data acquisition and control," in Proc. 3rd IEEE Int. Conf. Sensors,Vienna, Austria, Oct. 24–27, 2004, vol. 1, pp. 52–55.

[8] C. Bettstetter, H.-J. Vögel, and J. Eberspächer, "GSM phase 2+ General Packet Radio Service GPRS: Architecture, protocols, and air interface," IEEE Commun. Surveys Tuts., vol. 2, no. 4, pp. 2–14, Third Quarter 1999.

[9] RTOS Evaluation Project, "What makes a good RTOS," Dedicated Systems Experts, 2001. [Online]. Available:http://www.dedicatedsystems.com

[10] www.ethernut.de/en/firmware/nutos.html

[11] www.egnite.de/en/software/nutos.html