ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645
Indian Journal of Science and Technology, *Vol 8(S2), 272–277, January 2015*
DOI:.10.17485/ijst/2015/v8iS2/72374

Embedded Web Server Application for Industrial Automation

Ajith Kumar P. Shetty*, K. Ketan and M. Shanmugasundaram

SENSE, VIT University, Vellore, Tamil Nadu - 63201, India; akpshetty5@gmail.comlm

Abstract

Web Server based monitoring has been an issue for industries as they make use of PC-based servers which consume large power and occupy large area. This limitation can be overcome by replacing the existing PC-based server by an Embedded Web Server using Raspberry Pi. The prime objective of this paper is to design a remote data acquisition system which is controlled by Linux portable ARM processor and web server application with General Packet Radio Service (GPRS) technology. This system focuses not only on device monitoring but also controlling it. The monitoring and data collection is accompanied by a Short Message Service (SMS), an email alert which is initiated in order to avoid the occurrence of a critical event. The system is capable enough to withstand power failure and capable of restarting from the point of failure.

Keywords: ARM, Embedded Web Server, GPRS, Raspberry Pi, Remote Data Acquisition, SMS

1. Introduction

The origin of Web server comes from the requirement of a Client trying to access data which is made through HTTP (Hypertext Transfer Protocol) so that the web server can process, store and send the data on the request of client. Although the vital role of Web servers is to provide data it can also in some instances accept data from clients. Traditional methods make use of Unix and Linux workstations[1], typically requiring, large database storage systems occupying large area and high setup cost[2]. The sole purpose of this paper is to overcome the area and cost constraints which can be cut down and the system can be made more efficient. The Embedded web server provides services with minimum computing resources. The embedded industry has hardly evolved in past years. 8-bit microcontrollers are the bread and butter for the industry but slowly now more and more devices are not only gaining popularity but these embedded systems is also getting smart enough to be able to connect them to a network[2]. The embedded web server should be relatively small in size and easily integration with many devices and Raspberry pi is fit for that. Although they have limited hardware and storage capabilities, these hurdles hardly matter and it is still capable enough to perform vital tasks with these limitations. Internet is starting to get into day to day life of everyone and has become an integral part of our life. Users all over the world, is it home or industry want to access their devices remotely using the internet technology. The expectation that embedded web server carry with it is that it should be able to replace the personal computers and give way to enhancements in all the parameters which will boost the overall efficiency of the system. These parameters which provide an upper hand over the traditional computers are listed in Table 1. The data which is available on the embedded web server should be secured in the sense that any unidentified person should not be allowed unless his authentication is verified. The information provided by the module is collected and this data can be displayed on web pages[3]. These pages are basically located in the memory. Here the need of Raspberry pi over microcontroller can be understood from the fact that whenever an IP address is entered by the user on the address bar user intends to access the data collected by server. The embedded server will provide dynamic data whenever requested by the client.

**Author for correspondence*

Table 1. Parameters of the simulated PV module

Parameters	Raspberry Pi based Embedded Web Server	Traditional PC Based Web Server
1. Size	Small	Large
2. Cost	Cost effective($55)	Costly(about $600)
3. Energy consumption	5W	>250W
4. Portability	Yes	No
5. Heat sink	Not required	Required
6. Performance	Good (uses 700MHz ARM-11 Processor)	Good

2. System Design

2.1 Embedded Web Server

The arm processor present in the Raspberry pi provides the platform for data acquisition, the control unit and the embedded web server. Figure1 depicts the working of embedded web server in a nutshell. The embedded web server is continuously monitoring the temperature values from DS1820 temperature sensor and placing them on the server. This task is accompanied by a control action on server side if the client also intends to do so. The Raspberry pi has to continuously serve the asynchronous interrupts[4]. The system is designed such that any particular increase in temperature over a predefined threshold will turn the control device off and this is accompanied by sending an e-mail and SMS to the user. The embedded web pages are written and designed in HTML. These pages are designed user friendly to avoid unnecessary complexities on client. The client on the other side can access a remote device using embedded web server, all the client has to do is to login to the page using a valid user name and password and within second he is able to access all the data.

2.2 Hardware Design

2.2.1 Raspberry Pi

Raspberry pi is a credit card sized computer developed in the UK. It is different from that of the regular computers because it's not only small in size but also has the ability to integrate itself with electronic components which is of vital importance when designing an embedded web server. It overpowers the traditional microcontrollers in the sense that it has high capacity of RAM and a powerful processor which makes it an ideal choice for handling embedded applications. The need to use Raspberry pi as an embedded web server can be understood from the fact that to control a device, microcontroller is good pick but to do the same remotely pi stands out due to its 512Mb capacity of RAM and to be able to provide a clock frequency of 700 MHz. There are multiple ways of using Raspberry pi right from controlling an LED to getting a basic understanding of operating system. It is the best way to experiment with the board and get an idea of the working from inside. It has in-built compilers for good number of languages and the best support is for the language python as pi in Raspberry pi means python. This module is also best in the sense that the price to afford this platform is low Like other computers Raspberry pi also needs an operating system and the OS which it uses is Raspbian. Digital and analog output is provided by HDMI port. The processor has some features which require special device drivers and that is not available in its Linux Distribution.

2.2.2 Digital Temperature Sensor

The sensor used is DS1820 a Digital Thermometer which provides a 9–bit to 12-bit temperature readings preferably controlled by user. The DS1820 can measure temperature over the range of –55°C to +125°C in 0.5°C (Resolution) increments. Information is sent from the DS1820 over a 1–Wire interface, so that only one wire needs to be connected to GPIO pin which avoids unnecessary wirings.

2.3 Software Design

2.3.1 SQLite and Apache

SQLite is one of the public domain software packages that provide database management system. SQLite has a unique ability of being lightweight when it is compared on platforms like complexity, administrative overhead involved, and amount of resource usage. SQLite's small code size and conservative resource use makes it well suited for embedded systems running limited operating systems.The Apache HTTP server software or a program runs in the background on an operating system. It provides user with multi-tasking and services to other applications that connect to it, such as client web browsers. The Apache Web server provides a full range of Web server features, including CGI, SSL, and virtual domains.

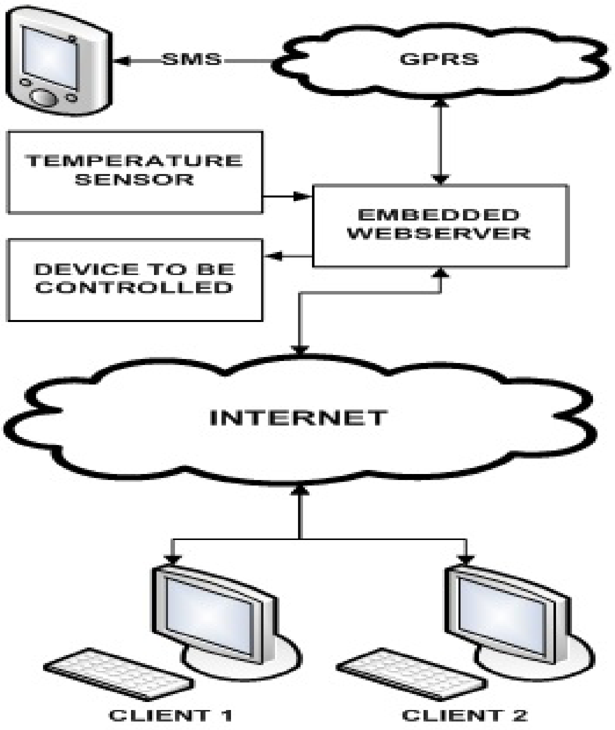


Figure 1. Block Diagram of Embedded Web Server.

2.3.2 Cron Tab

The software utility Cron is a time based job scheduler in UNIX like computer operating systems. It is used to schedule jobs (commands or shell scripts) to run periodically at fixed times, dates, or intervals. It has the capability to start the execution after recovering from a power failure.

3. Methodology

The system should be able to acquire data from remote areas, store and should be in a position to reproduce the data whenever demanded by the client at the other end. DS1820 is the sensor used for acquiring temperature. There is even a provision for controlling an electronic component from the client end which is demonstrated by controlling an LED.

The methodology is such that there are temperature sensors and LED in the remote area which are connected to the Raspberry pi module which acts as a Mini-computer in this case. This will be continuously monitoring the sensors and storing it in the database using SQL which is a light weight Database Management System. Since data is stored at a very high frequency, lot of unnecessary data is stored continuously in the Memory leading to filling up of memory space. To avoid this undesired event Cron Job is used which is basically a job scheduler in UNIX like operating System. This helps in scheduling and updation of data in database at a fixed time interval that can be decided by the user. So every five minutes (as defined by user) the Cron job automatically executes the program and stores temperature values in the database.

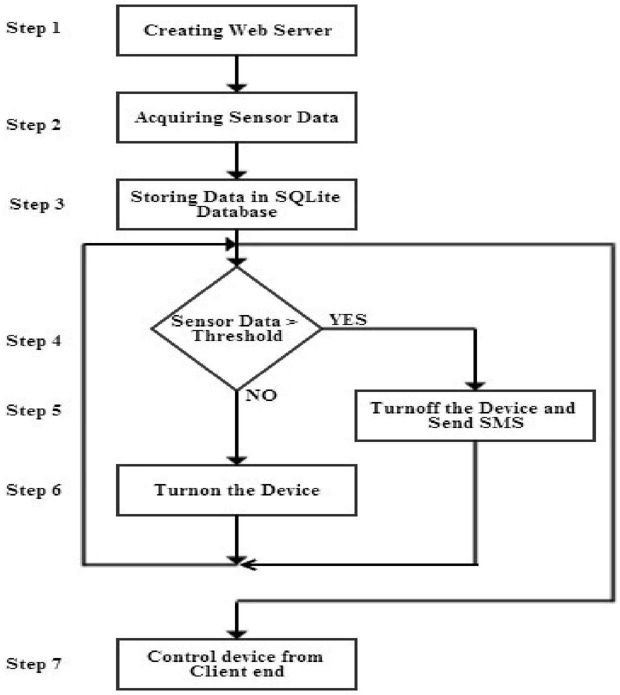
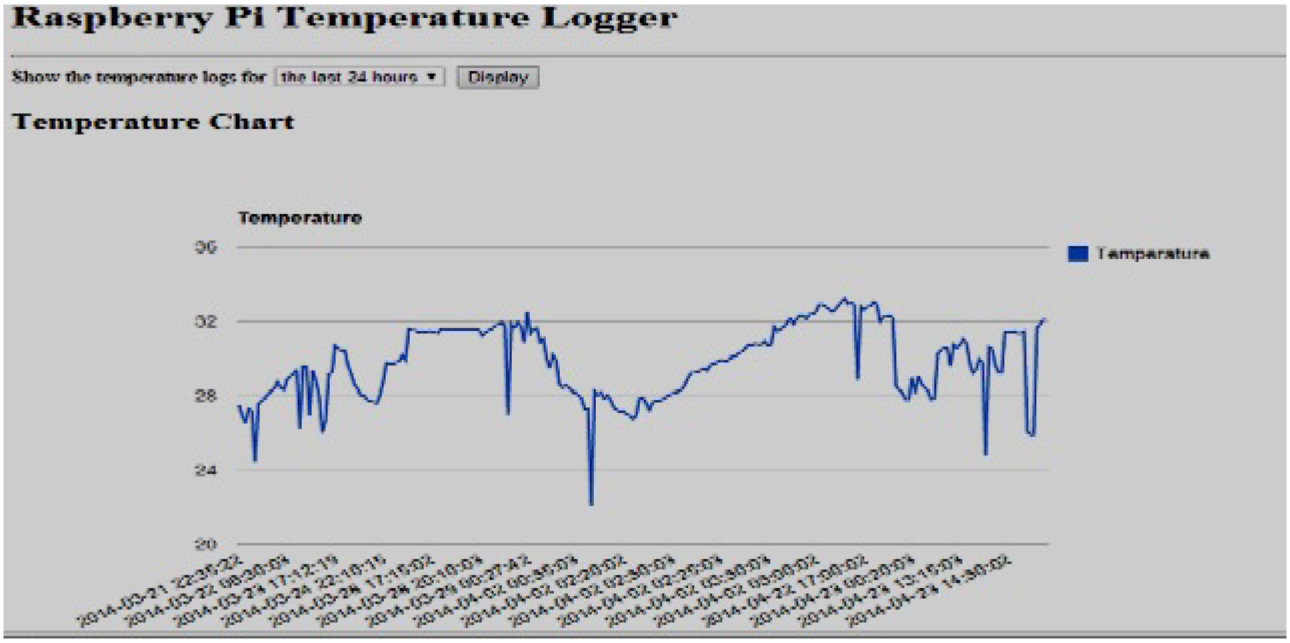


Figure 2. Flow chart of the working of Embedded Web Server.

The client on the other end is able to access the data using a Login page where authentication of his ID is checked using a Password if they do match client will not be allowed to access the data. After acquiring the temperature sensor values they are compared to that of threshold values and on the basis of comparison if they exceed the device is turned off and vice-versa. This simultaneously is accompanied by a control action to another device connected on the embedded web server which can be exercised by client. This is depicted in Figure 2.

This paper focuses mainly on device controlling task which is the upper hand when compared to the already existing systems wherein the communication takes place only in one direction and to make this system more user friendly Bidirectional connectivty is provided. Thus it provides user with multiple options of controlling a device from remote area which plays a vital role when considering that switchingoff a device can avoid a catastrophic event. The person at the client end can access the current as well as the previous data. To make the data Comprehensible, it is displayed as a graph for the ease of the user as shown in Figure 3.



Raspberry Pi Temperature Logger

Show the temperature logs for the last 24 hours ▾ Display

Temperature Chart

Figure 3. Temperature chart.

4. Experimental Results

Figure 4 depicts the continuous temperature values collected by Raspberry pi module at the inception which displays data at different intervals of time on the teminal screen. This data is to be managed by SQlite for storing the data in memory and it regularly flushes out the unnecessary data so that the memory does not overflow.

Figure 5 shows that their is a login page that is created to test the authenticity of the user. It dynamically checks for the credentials provided by the user so that an access to a web server's information can be provided to that user. As of now a single users login and password details are created which can extended to many users depending on the size of the RAM system has.

Figure 6 shows the control switches designed at the client end so that the client can control the device from a remote area.The control action can be, controlling a LED or a motor from the client end.

5. System Merits

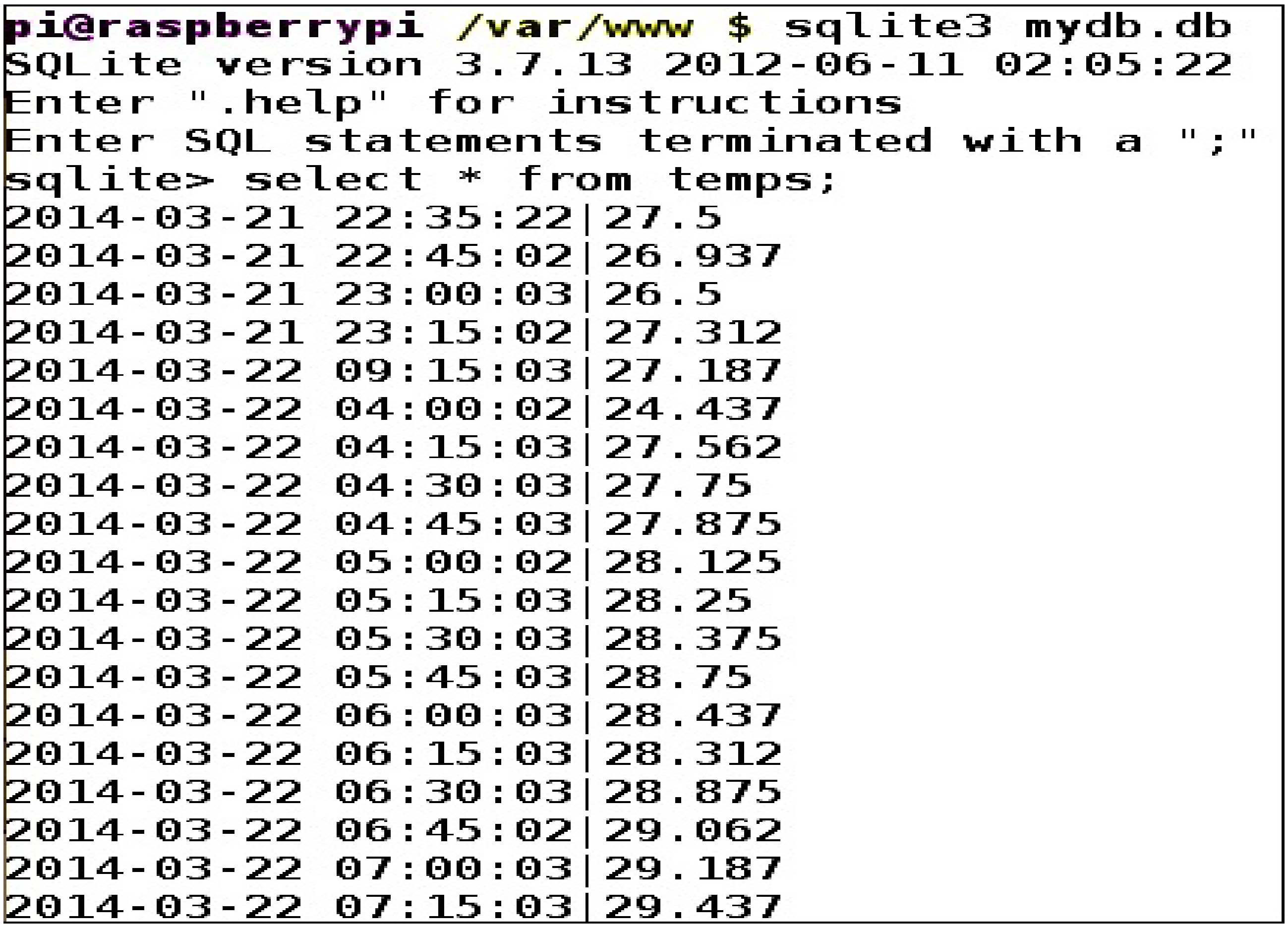
5.1 Existing Work

The method of using single chip data acquisition has a limitation in processing capability and also lags in producing reactive output.

Conventional web servers demand large amount of memories and area which lead also to an increase in cost. A comparison between existing and the proposed system is shown in Table 1.

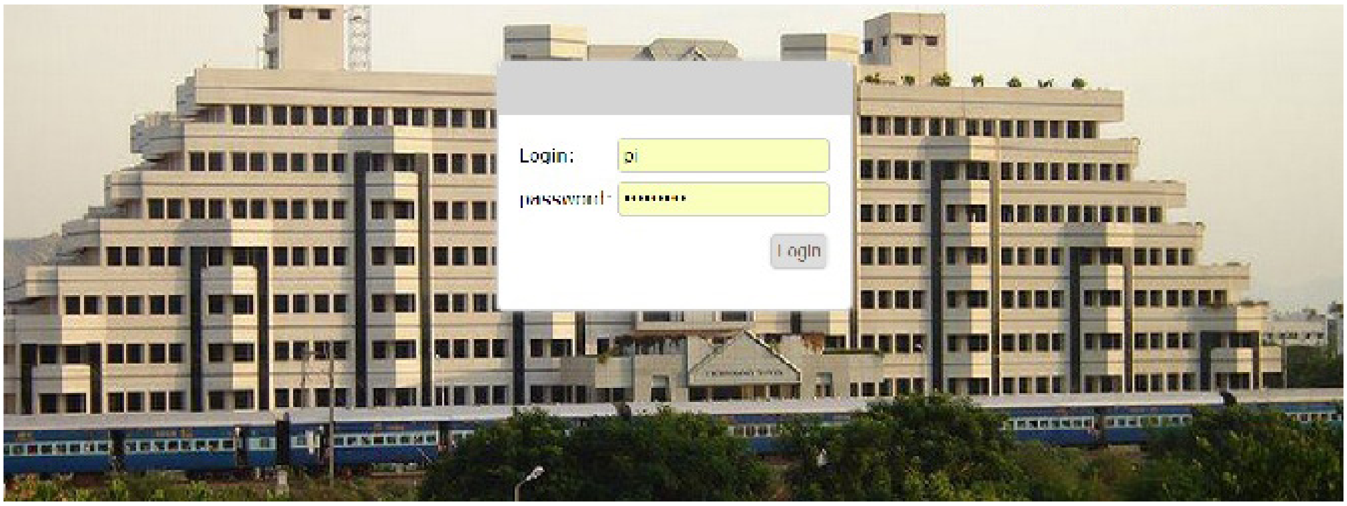
5.2 Proposed Work

The problem of Size, cost and power consumption are overcome by using the Raspberry Pi module as it does well in all the domains in which the conventional systems fail. Using Raspberry Pi as web server we are not only able to receive data from server but also able to control a device present in a remote area through proper authentication.



```
pi@raspberrypi /var/www $ sqlite3 mydb.db
SQLite version 3.7.13 2012-06-11 02:05:22
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
sqlite> select * from temps;
2014-03-21 22:35:22|27.5
2014-03-21 22:45:02|26.937
2014-03-21 23:00:03|26.5
2014-03-21 23:15:02|27.312
2014-03-22 09:15:03|27.187
2014-03-22 04:00:02|24.437
2014-03-22 04:15:03|27.562
2014-03-22 04:30:03|27.75
2014-03-22 04:45:03|27.875
2014-03-22 05:00:02|28.125
2014-03-22 05:15:03|28.25
2014-03-22 05:30:03|28.375
2014-03-22 05:45:03|28.75
2014-03-22 06:00:03|28.437
2014-03-22 06:15:03|28.312
2014-03-22 06:30:03|28.875
2014-03-22 06:45:02|29.062
2014-03-22 07:00:03|29.187
2014-03-22 07:15:03|29.437
```

Figure 4. Temperature readings.



Login: pi

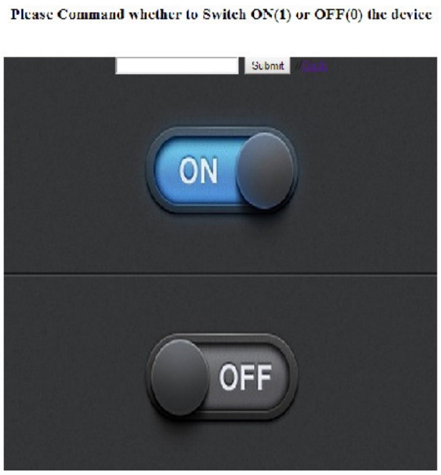
password: ••••••••

Figure 5. Login page.

6. Conclusion

The rapid development in industrial sector demands an efficient implementation of web server.

The Raspberry pi Embedded Web Server is an effective solution for acquiring the data and reproducing it in the form of a graph with current and previous values, this is done on clients demand which stands out in comparison to that of the traditional method of using PC-Based Unix servers. This system plays a vital role in cutting down the cost and area requirement. The module has an advantage that it can continue its operation even after a power interruption without human intervention.



Please Command whether to Switch ON(1) or OFF(0) the device

Figure 6. Control Switches.

7. References

1. Manivannan M, Kumaresan N. Embedded web server and gprs based advanced industrial automation using linux rtos. Int J Eng Sci Tech. 2010; 2(11):6074–81.

2. Choi M-J, Ju H-T. An efficient embedded web server for web-based network element management. Network Operations and Management Symposium, 2000. NOMS 2000. 2000 IEEE/IFIP; 2000 Apr; Honolulu. p. 187–200.

3. Lin T, Zhao H, et al. An embedded web server for equipments. 7th International Symposium on Parallel Architectures, Algorithms and Networks; 2004 May 10–12. p. 345–50. 4. Raskovic D, Revuri V, Giessel D. Embedded web server for wireless sensor networks. 41st Southeastern Symposium on System Theory; 2009 Mar; Tullahoma. p. 19–23.

5. Manivannan M, Kumaresan N. Design of on-line interactive data acquisition and control system for embedded real time applications. International Conference on Emerging trends in Electrical and Computer Technology; 2011 Mar; Tamilnadu. p. 551–6.

6. Patinge S, Suryawanshi Y, Kakde S. Design of ARM based data acquisition and control using GSM and TCP/IP Network. 2013 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC); 2013.

7. M Poongothai. ARM embedded web server based on DAC system. 2011 IEEE International Conference on Process Automation, Control and Computing (PACC); 2011.