International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 05 Issue: 05 | May - 2021 ISSN: 2582-3930

EMBEDDED WEB SERVER APPLICATION FOR INDUSTRIAL AUTOMATION USING RASPBERRY PI

Sachin Kushwaha / Ritul Gaikwad / Mrignaini Amblimath / Anjali Singh Project Guide / Prof. Jagdish Mandhare
Bharati Vidyapeeth College of Engineering, Navi Mumbai
Department Of Instrumentation / India

Abstract —In the present era, there is a great need of fully automated industrial system. The objective of all the researchers through their proposed project is to solemnly introduce a new and advance technology for the industries in the world of control & monitoring as well as data acquisition & computation & to take the industries a step forward in the era of industrial revolution 4.0. To achieve the desired goal, the idea is to replace all the traditional microcontrollers & their control panels with a single compact set of microcontroller & microprocessor known as Raspberry Pi. It acts as a complete set of compact CPU. It is converted into a private web server allowing the connection of the whole process industry with the digital world via internet. Thus, connecting the industries to the technologies such as IOT & IIOT. This system overcomes the problem of area, cost & power constraints by cutting down to a single compact microcontroller making the system more efficient & effective. The idea behind the present system was to facilitate the user for unmanned & remote access of the real time data. The said system is capable of measuring physical parameters of electrical device such as system temperature, level, liquid flow etc. In addition to this, the given real time data is also sent to pre-designed web page & is accessible on global scale. By using this project, it is possible to change the status (ON/OFF) of any appliance connected to ARM processor board via internet remotely.

Keywords — ***ARM, Embedded Web Server, Wireless Sensor Network, TCP/IP Protocol, Remote Data Acquisition, Raspberry Pi, Embedded System, Ethernet.***

I. 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 workstations with operating systems such as windows, Linux and UNIX. These workstations are capable of storing large amount of data occupying large area and setup cost. The sole purpose of this project is to overcome the area, cost and power constraints which can be cut down and the system can be made more efficient. The embedded web server can provide services with minimum computing resources. Embedded Industry has hardly evolved in past years. The embedded web server should be relatively small in size and easy integration with many devices and Raspberry pi is fit for that. This module is able to collect the data and display it on web pages. Whenever any user types IP address in the web browser address bar, he is intend to access the data collected by the server. The embedded server is able to provide dynamic data whenever requested by the client. The system is designed such that raspberry pi continuously monitors the level and if the level exceeds a predefined value then raspberry pi will turn off the control device. An alert mail can be sent to the user. The embedded web pages are designed by using HTML, CSS and PHP programming languages. The pages are designed in a simple user friendly fashion to avoid unnecessary complexities. However, the client can control the devices remotely by using embedded web server, the client has to login to the web page using login credentials and can access all the data.

II. BLOCK DIAGRAM

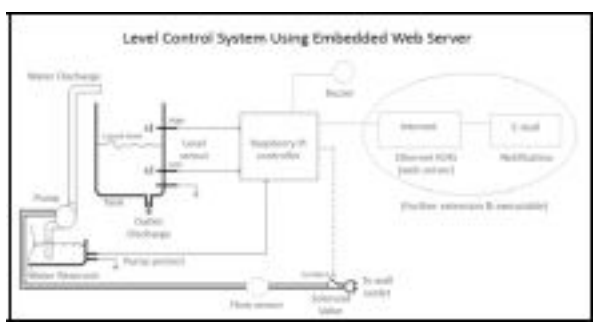

Fig. 1 Block Diagram

Figure 1. depicts the working of embedded web server in a nutshell. The embedded web server is continuously monitoring the level parameters through water level 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. The system is designed such that any particular increase in water level 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, the client has to login to the page using a valid user name and password and within second he is able to access all the data.

Parameters	Raspberry Pi based Embedded Web Server	Traditional PC based Web Server
1. Size	Small	Large
2. Cost	Cost Effective	Costly
3. Performance	Good(Uses 700 MHz ARM-V8 Processor)	Good
4. Portability	Yes	No
5. Energy Consumption	5W	>250W

Table 1. Parameters of Simulated PV Module

III. HARDWARE

The list of Hardware Components are given below:

SR NO.	HARDWARE	TYPES
1.	Raspberry Pi	Model 3B+
2.	Level Process Parameters	a. Solenoid Valve (12V DC)
		b. Water Flow Measurement Sensor (YF-S402)
		c. Water Pump (12V, 4A, 5.5 LPM)
		d. Ultrasonic Sensor (HC-SR04)

IV. SYSTEM DESIGN

➢ 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 level values using the ultrasonic 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 interrupts4. 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.

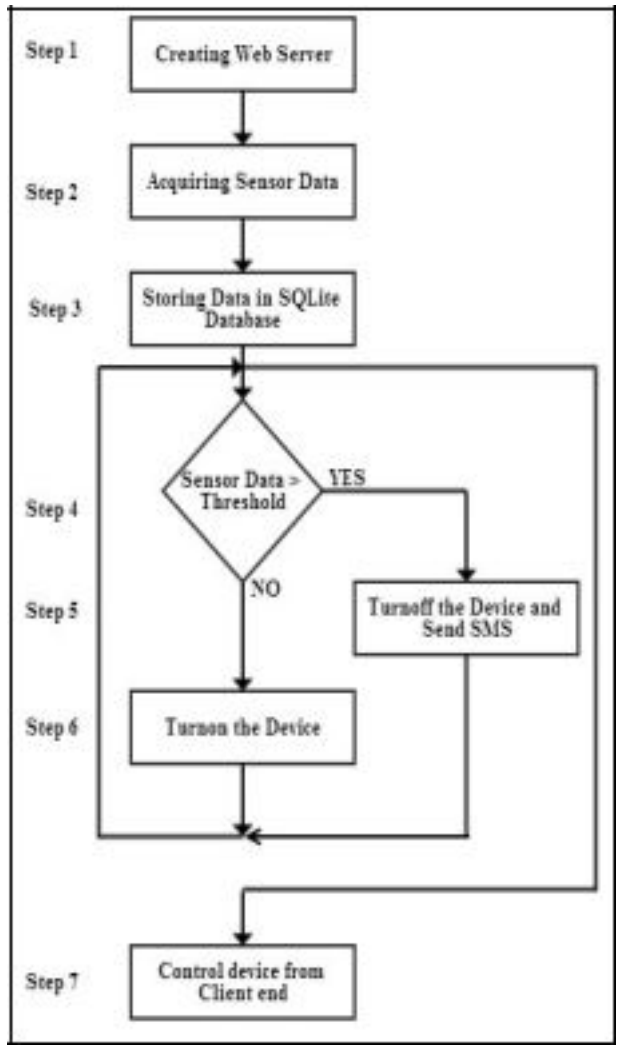

Fig. 2 Flow chart of Working of Embedded Web Server

V. HARDWARE DESIGN

1. RASPBERRY PI

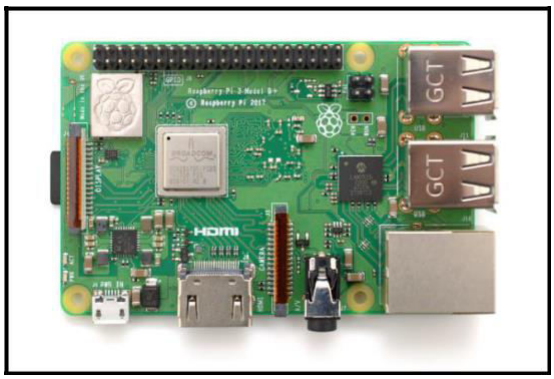

Fig. 3 Raspberry Pi Model 3B+

Raspberry pi is a credit card sized computer developed in 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 a good pick but to do the same remotely pi stands out due to its 1GB capacity of RAM and to be able to provide a clock frequency of 700 MHz.

2. ULTRASONIC SENSOR

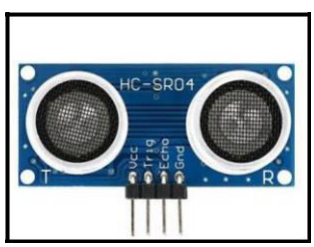

Fig. 4 Ultrasonic Sensor

Ultrasonic Sensors operate on the basic principle of using sound waves to determine fluid level. The frequency range for ultrasonic methods is ~20-200 kHz. A transducer directs the sound waves downward and it bursts onto the surface of the water. Echoes of these waves return to the transducer, which performs calculations to convert the distance of wave travel into a measure of height, and therefore water level.

Proper mounting is important to ensure that sound waves are reflected perpendicularly back to the sensor. Otherwise, even slight misalignment of the sensor in relation to the process material reduces the amount of sound wave detected by the transducer. Since the ultrasonic transducer is used both for transmitting and receiving the acoustic energy, it is subjected to a period of mechanical vibration known as "ringing". This vibration must attenuate (stop) before the echoed signal can be processed. The net result is a distance from the face of the transducer that is blind and cannot detect an object. It is known as the "blanking zone", typically 150mm – 1m, depending on the range of the transducer.

The requirement for electronic signal processing circuitry can be used to make the ultrasonic sensor an intelligent device. Ultrasonic sensors can be designed to provide point level control, continuous monitoring or both. Due to the presence of a microprocessor and relatively low power consumption, there is also capability for serial communication from/to other computing devices making this as a good technique for adjusting calibration and filtering of the sensor signal, remotely wireless monitoring or plant network communications.

VI. SOFTWARE DESIGN

1. RASPBERRY PI OS (RASPBIAN)

Raspbian

Raspberry Pi OS (formerly known as Raspbian) is a Debian-based operating system for Raspberry Pi. It has been officially provided by the Raspberry Pi Foundation as the primary operating system for the Raspberry Pi family of compact single-board computers. Raspberry Pi OS is highly optimized for the Raspberry Pi line of compact single-board computers with ARM CPUs. Raspberry Pi OS uses a modified LXDE as its desktop environment with the Open-box stacking window manager plus a new theme and few other changes. The distribution is shipped with a copy of the algebra program Wolfram Mathematica and a version of Minecraft called Minecraft Pi as well as a lightweight version of Chromium as of the latest version.

2. APACHE

The **Apache HTTP Server**, called as **Apache** is a free and open-source, cross-platform web server software. It runs in the background on an operating system. It provides user with multi-tasking 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.

3. SOCKET.IO

Socket.IO is a JavaScript library for real-time web applications. It enables real-time, bi-directional communication between web clients and servers. It has two parts: a client-side library that runs in the browser, and a server-side library for Node.js. Both components have a nearly identical API. Like Node.js, it is event-driven.
Socket.IO primarily uses the Web Socket protocol, while providing the same interface. Although it can be used simply as a wrapper for Web Socket, it provides many more features, including broadcasting to multiple sockets, storing data associated with each client, and asynchronous I/O. It can be installed with the NPM tool.

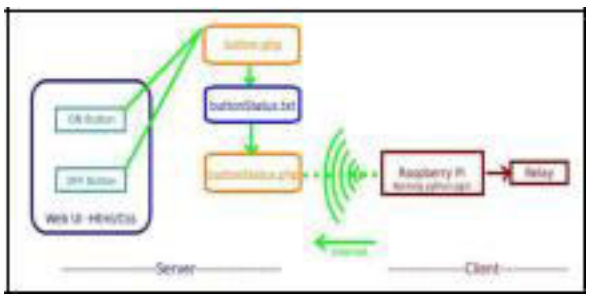

Fig. 5 Server Side & Client Side Communication

VII. 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. Ultrasonic Sensor is the sensor used for acquiring level. 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 level 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 a light weight Database Management System.

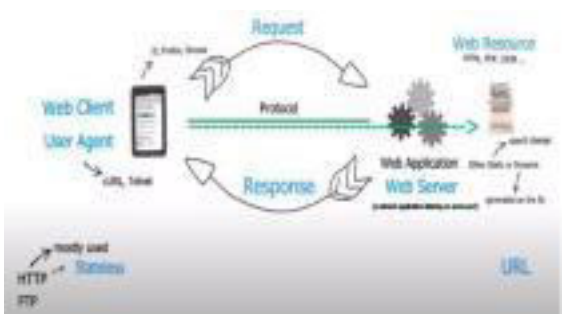

Fig. 6 Actual Communication of the System

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 level 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 6.

VIII. EXPERIMENTAL RESULTS

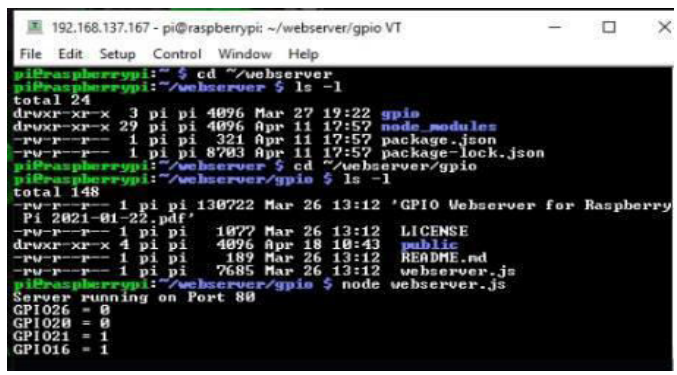

Fig. 7 Creation of Web Server

Figure 7. depicts the Web Server Creation for Raspberry pi module at the inception which displays data at different intervals of time on the terminal screen. This data is to be managed for storing the data in memory and it regularly flushes out the unnecessary data so that the memory does not overflow.

IX. PROPOSED WORK

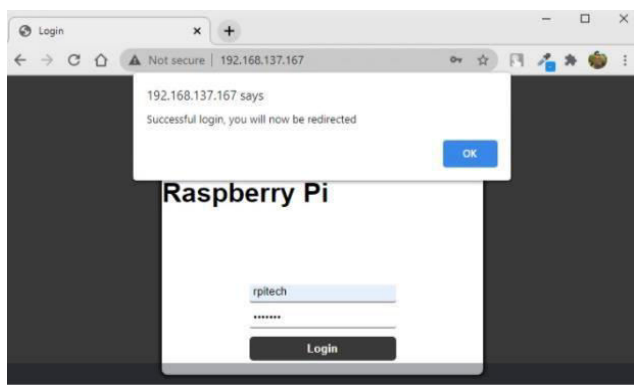

Fig.8 Login Page

Figure 8. shows, 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 too many users depending on the size of the RAM system has.

X. CONCLUSION

The Raspberry pi Embedded Web Server is an effective solution for acquiring data and reproducing it, 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.

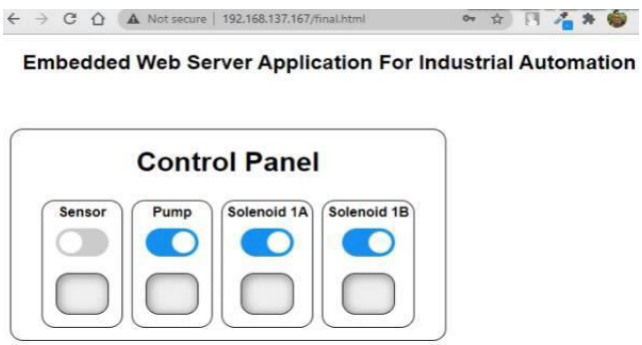

Fig. 9 Control Panel on Terminal Screen

Figure 9. 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 the desired parameter from the client end.

XI. REFERENCES

[1] Prof.S.S.Wagh, PradeepV.Kalge, Remote monitoring and control of industrial parameters Using Embedded web server and GSM, Indian Journal of applied research, vol. 3, pp. 214, Jul. 2013.

[2] SarikaChhatwani, Embedded web server, IJEST, vol. 3, no. 2, pp. 1233- 1238, Feb. 2011.

[3] Igor Klimchynski, Extensible Embedded web server architecture for internet based data Acquisition and control, IEEE Sensor Journal, vol. 6, no. 3, pp. 804-811, Jun. 2006.

[4] 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.

[5] Nagisetty Sasidhar and Monica P. Suresh. "ARM microcontroller based Wireless Industrial Automation System", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (An ISO 3297:2007 Certified Organization) Vol. 3, Special Issue 4, April 2014.

https://www.researchgate.net/publication/276167789_Embedded_Web_Server_Application_for_Industrial_Automation

https://www.raspberrypi-spy.co.uk/2014/09/infographic-raspberry-pi-how-to-get-started/

https://www.engineersgarage.com/article_page/level-sensors/

https://www.instructables.com/Simulated-analogdigital-converter-for-RaspberryPi/

https://github.com/Azure-Samples/raspberry-pi-web-simulator