Embedded Web Server Based Surveillance System using ARM

Dr. Shaik Meeravali (Head Of the Department),
S. Madhu (Asst. Professor),
Electronics & Communication Engineering,
RRS College of Engineering & Technology,
Muthangi (V), Patancheru (M), Medak (D), A.P,
India.

M. Sarojini (M. Tech, Embedded Systems),
Electronics & Communication Engineering,
RRS College of Engineering & Technology,
Muthangi (V), Patancheru (M), Medak (D), A.P,
India.

Abstract

This paper presents a networked embedded system for monitoring and controls the home devices. With the scalable networking solutions, the server enables web access to distributed measurement/ control systems and provides optimization for home automation. Here, the principles and design of a system for Internet based control in an automated home by using Advanced RISC machine i.e. ARM Processor are presented.

.

1. Introduction

Home appliances that need to be controlled or tracked from a far-flung place are connected to the embedded web server board developed using ARM 7 based processor. This board is also connected to the Internet or LAN. A web page on the remote browser is used to control and monitor the appliances provided that the browser must be connected to the Internet or the same LAN. The web server is given its unique IP address. This IP address is pinged at web browser. It shows a web page on the browser. This web page includes control buttons and indicators for controlling and tracking the appliances respectively.

In this paper along with the device tracking, physical quantity monitoring is also proposed. Here, temperature sensor is used to collect the temperature and is displayed on the remote browser. The appliances or tools need to be tracked are connected to the web server board. The present state of the appliances is displayed on the far-flung browser. State represents ON or OFF condition of the appliances. In a similar way, the appliances are controlled using switches on the web page. Control represents the turning ON or OFF the appliances.

Figure 1. shows the block diagram of the proposed system. The principal parts of the system are ARM based microcontroller and Ethernet controller. The micro controller is responsible to perform all necessary control functions. Ethernet controller performs data packets transmission and reception over the LAN or Internet. The design ensures a high speed data transfer of 10 Mbps with Collision Detection mechanism.

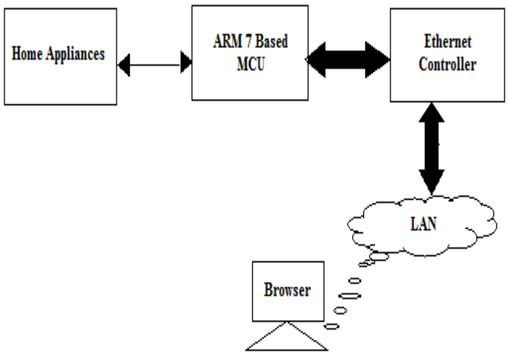

Figure 1. Block diagram

2. Overview of the System

This technique is to control and monitor the appliances or equipment's from the remote place through a web page. Here all the devices, which are to be controlled, are connected to the relays (act as switches) on the web server circuit board. The web-server circuit is connected to LAN or Internet. The client or a person on the PC is also connected to same LAN or Internet. By typing the IP-address of LAN on the web browser, the user gets a web page on screen; this page contains all the information about the status of the devices. The user can also control the devices interfaced to the web server by pressing a button provided in the web.

3. LM 35 Sensor

LM 35 is a precision integrated temperature sensor. Its output is linearly proportional to the Celsius (Centigrade) temperature.

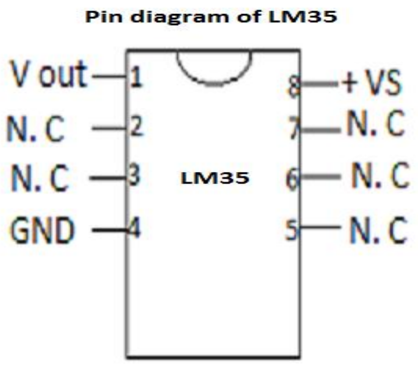

Figure 2. Pin diagram of temperature sensor

3.1. Testing of Temperature Sensor

In order to test the temperature sensor, various temperatures are applied to the LM35 sensor and these temperatures are measured using thermometer. The respective output voltages of LM35 sensor are measured using multimeter.

Table 1. Observations of temperature sensor

Temperature (in ◦C) measured using thermometer	Output Voltage (mV) measured at V out pin of LM35
25	252
27	275
30	301
32	323
35	350
38	381
40	404
43	433
44	441
45	455

3.1.1. Conclusion. Above readings indicate that output voltage of LM35 sensor rises by 10 mV /◦C rise in temperature approximately. Thus its output is linear with scale factor 10mV/◦C.

3.2. Interfacing LM35 with LPC 2148

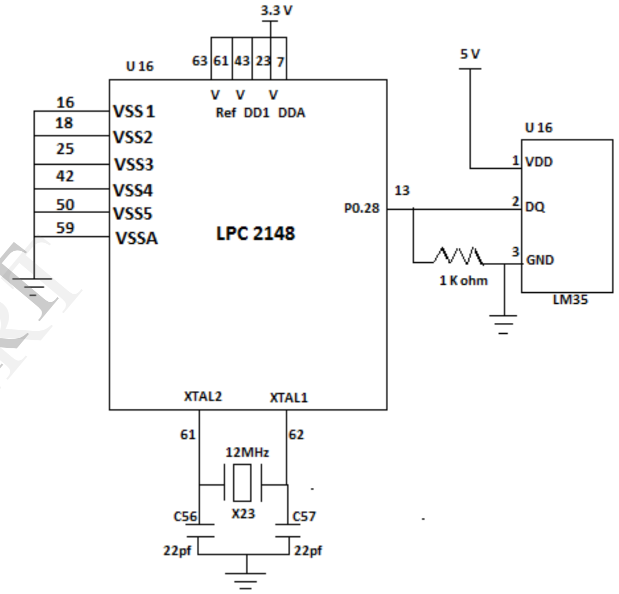

Figure 3. Interfacing temperature sensor with LPC 2148

3.3. Program to Read the Value of Temperature Sensor

```
unsigned long A2D(void)
{           unsigned char wait=20;
            AD0CR =           0x00250602;
            BURST=1 | CLKDIV = 0x06
            AD0CR |=          0x01000000;
            while(wait--);
            do
            {
               vals = AD0GDR
            }
            while ((vals & DONE) == 0);
            vals = ((vals >> 6) & 0x03FF);
            return(vals);
     }
```

4. Ethernet

Ethernet is a family of computer networking technologies for Local Area Networks (LANs).) It was commercially introduced in 1985 and standardized in 1985as IEEE 802.3. The Ethernet standards comprise several wiring and signaling variants of the OSI physical layer in use with Ethernet. Ethernet stations communicate by sending each other data packets; blocks of data individually sent and delivered.

Signals communicating over Ethernet divide a stream of data into shorter pieces called frames. A frame begins with preamble and start delimeter, followed by an Ethernet header featuring source and destination addresses. The middle section of the frame consists of payload data including any other headers for other protocols (e.g., IP) carried in the frame. The frame ends with a 32- bit cyclic redundancy check, which is used to detect corruption of data in transit.

Each Ethernet station is given a 48-bit MAC address. The MAC addresses are used to specify both the destination and source of each data packet. Original stations shared coaxial cable traversed a building or campus to every attached machine. Since all communications happen on the same wire, collisions happen when two stations attempt to transmit at the same time. They corrupt transmitted data and require stations to retransmit.

In a modern Ethernet, the stations do not all share the one channel through shared cable, instead, each station communicate with a switch, which in turn forwards that traffic to the destination station. In this topology, collisions are only possible if station and switch attempt to communicate with each other at the same time and collisions are limited to this link. The 10Base-T standard introduces a full-duplex mode of operation where switch and station can communicate with each other simultaneously, and therefore modern Ethernets are completely collision free.

5. Ethernet Controller ENC28J60

The ENC28J60 is a stand-alone Ethernet controller with an SPI interface that serves as a communication channel between LPC2148 microcontroller and the ENC28J60. It meets all of the IEEE 802.3 specifications. The ENC28J60 has the MAC (Medium Access Control) module that implements IEEE 802.3 compliant MAC logic and a PHY (Physical Layer) module that encodes and decodes the analog data that is present on the twisted pair interface.

It incorporates a number of packet filtering schemes to limit incoming packets. Communication with LPC2148 microcontroller is implemented via two interrupt pins and the SPI, with data rates of up to 10 Mb/s.

5.1. Interfacing ENC28J60 with LPC2148

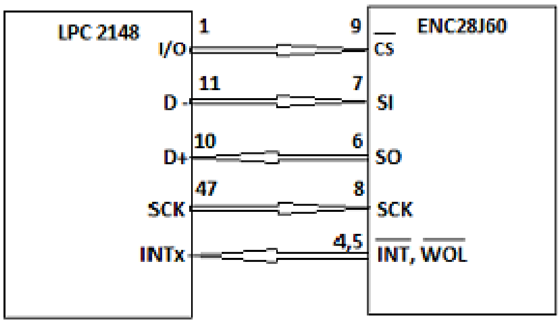

Figure 4. Interfacing Ethernet controller with LPC 2148

6. Transmission Control Protocol/ Internet Protocol (TCP/IP)

The Transmission Control Protocol (TCP) is one of the center protocols of the Internet protocol suite (IP). TCP provides consistent, prearranged, error-checked deliverance of a stream of octets between programs running on computers connected to a LAN, intranet or the public Internet. Web browsers use TCP when they connect to servers on the World Wide Web, and it is used to deliver email and transfer files from one location to another.

This protocol corresponds to the transport layer of TCP/IP suite. TCP provides a communication service at a transitional level between an application program and the Internet Protocol (IP). That is, when an application program desires to send a large amount of data across the Internet using IP, instead of breaking the data into IP-sized pieces and issuing a series of IP requests, the software can issue a single request to TCP and let TCP handle the IP details.

IP works by exchanging pieces of information called packets. A packet is a series of octets and consists of a header followed by a body. The header describes the packet's source, destination and control information. The body contains the data IP is transmitting.

Therefore, TCP is a reliable stream delivery service that guarantees that all bytes received will be identical with bytes sent and in the accurate order. Since packet transfer over many networks is not reliable, a technique known as positive acknowledgment with retransmission is used to guarantee consistency of packet transfers. This fundamental technique requires the receiver to respond with an acknowledgment message as it receives the data. The sender keeps a record of each packet it sends. The sender also maintains a timer from when the packet was sent, and retransmits a packet if the timer expires before the message has been acknowledged. The timer is needed in case a packet gets lost or corrupted.

TCP protocol operations may be alienated into three phases. Connections must be properly established in a multi-step handclasp process (connection establishment) before entering the data transfer stage. After data transmission is completed, the connection termination closes established virtual circuits and releases all allocated resources. Throughout the lifetime of a TCP connection, the local end-point undergoes a series of state changes.

6.1. Two-way Communication

6.1.1. Connection Establishment. To establish a connection, TCP uses a three-way handclasp. Before a client attempts to connect with a server, the server must first attach to and listen at a port to open it up for connections. This is called a passive open. Once the passive open is established, a client may initiate an active open. To establish a connection, a 3-step handclasp occurs.

1. **SYN**: The active open is performed by the client sending a SYN to the server. The client sets the segment's sequence number to a random value A.

2. **SYN-ACK**: In response, the server replies with a SYN-ACK. The acknowledgment number is set to one more than the received sequence number i.e. A+1, and the sequence number that the server chooses for the packet is another random number, B.

3. **ACK**: Finally, the client sends an ACK back to the server. The sequence number is set to the received acknowledgement value i.e. A+1, and the acknowledgement number is set to one more than the received sequence number i.e. B+1.

At this point, both the client and server have received an acknowledgment of the connection. The steps 1, 2 establish the connection factor for one direction and it is acknowledged. The steps 2, 3 establish the connection factor for the other direction and it is acknowledged. With these, a full-duplex communication is established.

6.1.2. Connection Termination. The connection termination uses a four-way handclasp, with each side of the connection terminating separately. When an endpoint desires to stop its half of the connection, it transmits a FIN packet, which the other end acknowledges with an ACK. Therefore, a typical break up requires a pair of FIN and ACK segments from each TCP endpoint. After both FIN/ACK interactions are concluded, the side which sent the first FIN before receiving one waits for a timeout before finally closing the connection, throughout which time the local port is engaged for new connections; this prevents confusion due to deferred packets being delivered during successive connections.

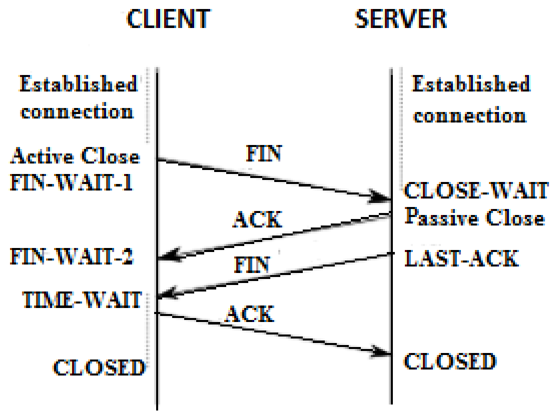

Figure 5. Connection termination

FIN-WAIT-1 represents waiting for a connection termination request from the isolated TCP, or an acceptance of the connection termination request formerly sent.

FIN-WAIT-2 represents waiting for a connection termination request from the isolated TCP.

CLOSE-WAIT represents waiting for a connection termination request from the confined user.

LAST-ACK represents waiting for an acknowledgment of the connection termination request previously sent to the remote TCP (which includes an acknowledgment of its connection termination request).

TIME-WAIT represents waiting for enough time to pass to be sure the remote TCP received the acknowledgment of its connection termination request.

CLOSED represents no connection state at all.

7. IP Address

An IP address is a commercial tag assigned to each machine participating in a computer network that uses the Internet Protocol for communication. It serves two prime functions: host or network interface recognition and position addressing. The designers of the Internet Protocol defined an IP address as a 32- bit number. This system is known as Internet Protocol Version 4(IPv4).

IP addresses are binary numbers, but they are usually stored in manuscript files and displayed in human legible notations such as 192.168.10.106. In IPv4, an address consists of 32 bits. It confines the address space to 4294967296(2^ 32) possible distinctive addresses. IPv4 assets some addresses for special purposes such as private networks. IPv4 addresses are represented in dot decimal notation, which consists of four decimal numbers, each ranging from 0 to 255, alienated by dots. Each part represents a cluster of 8-bits of the address e.g., the 32- bit hexadecimal address C0A80A6A is written as 192.168.10.106.

8. Hyper Text Transfer Protocol (HTTP)

HTTP defines the set of laws by which web browsers, web servers, proxies, and other web systems launch and preserve connections with each other. It uses only four protocol layers within a computer system. The lowest layer is the Network layer. The protocol layer above the Network layer is the Internet Protocol (IP) layer. Next is the TCP. The final protocol layer is HTTP, a TCP/IP application layer. There is a confined and physical communication between each protocol layer.

HTTP reads or writes data from/ to TCP. TCP interacts directly with IP. IP interacts with the protocol controlling Network layer (i.e., Ethernet). A protocol uses its header and name for the entity of data it sends and receives. Each protocol layer adds or removes its own explicit information. HTTP follows client/ server regulations where only a client initiates the communication. HTTP requires a TCP connection. After the connection is established, the client can send a HTTP request. Each HTTP request begins with a request line. This is a manuscript line which indicates the HTTP method, the resource, and the HTTP version. The request line is followed by optional message head, and optional message body. The HTTP response begins with a status line. This line starts with the HTTP version that the server supports. Then a status code follows within the status line. The status line is followed by optional message header, and the optional message body.

9. Flow Chart

A web server can merely be imagined like a special kind of a file server. The files are the resources (e.g., HTML files, GIF and JPEG pictures etc.). The web server delivers these resources with a very special communication protocol called HTTP. It needs a storage space for the resources. For this purpose, the registers are provided in an embedded web server. The communication link between the client and embedded appliance is established using an Ethernet Controller.

An embedded web server is an ARM processor that contains an internet software suite as well as application code for tracking and controlling machines/systems. These embedded web servers are an integral part of an embedded network. ARM processor is the responsible part for measuring signals and controlling the devices remotely. It manages all the tasks such as measuring the signals, data updating, sending HTML pages and linking/ communicating with users. To directly contact an embedded web server, the IP address of the embedded appliance should be made available to the consumer side. Whenever the consumer wants to access data, he/she sends the request to the server. This request is taken by the router, which is linked to the LAN or Internet. The web processes the request made and finally it connects to the desired web server, access the required data and send to the client.

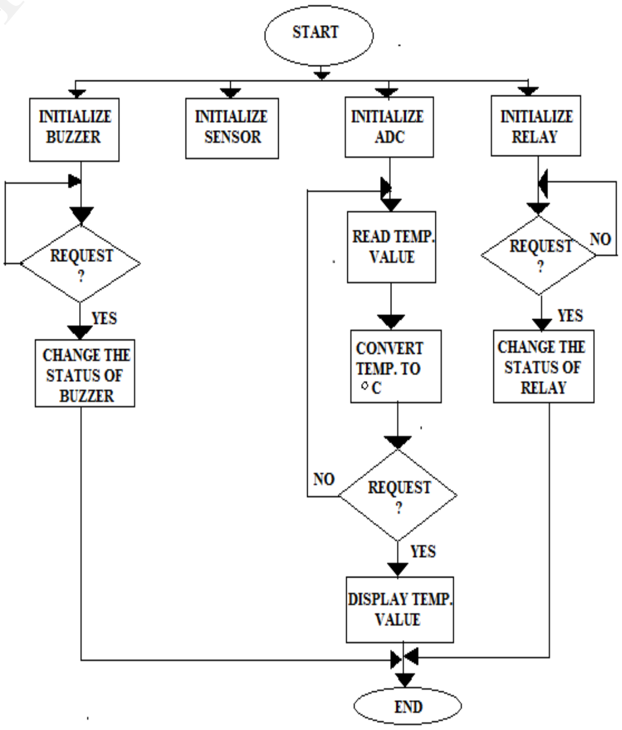

Figure 6. Flow chart

10. Conclusion

Embedded web servers are an integral part of an embedded network. Embedded Servers can be used to change the status of the various Gadgets connected to the kit by means of Internet. The embedded web server design includes a complete web server with TCP/IP support and Ethernet interface. It provides the software for automatic configuration of the web server in the network. The embedded web server reference design includes complete source code written in C-language.

References

[1] http://www.flashmagictool.com
[2] http://www.nxp.com
[3] http:// www.dilnetpc.com/WSforES1-2.pdf
[4] http://www.nxp.com
[5] http://www.embeddedethernet.com
[6] http://www.electronic-circuits-diagrams.com
[7] http://www.microchip.com
[8] http://www.elexol.com
[9] http://www.ti.com
[10] http://www.embedtranics.blogspot.in
[11]Robert L. Boylestad, Louis Nashelsky, Electronic Devices and Circuit theory
[12] Embedded Ethernet by Jan Axleson
[13] Black Book of HTML, by Steven Holzner
[14]www.atmel.com/dyn/resources/prod_documents/doc0336.pdf
[15] Ethernet Technologies. Cisco Systems
[16] Ethernet Controller Technical Reference Manual. Cirrus Logic Inc
[17] Transmission Control Protocol by Postel J
[18] Internet Protocol (IP) by Postel J

Authors' details

First Author

Dr. Shaik. Meeravali,
Professor and Head,
Department of Electronics and Communication Engg,
RRS College of Engineering and Technology,
Muthangi (V), Patancheru (M), Medak (D), A.P, India.

Second Author

Mr. S. Madhu,
Asst.Professor,
Department of Electronics and Communication Engg,
RRS College of Engineering and Technology,
Muthangi (V), Patancheru (M), Medak (D), A.P, India.

Third Author

M. Sarojini (M. Tech, Embedded Systems),
Department of Electronics and Communication Engg,
RRS College of Engineering and Technology,
Muthang i(V), Patancheru (M), Medak (D), A.P, India.