Research Article
Vijayakumar D, IJPRET, 2013; Volume 2 (5): 68-80

ISSN: 2319-507X
IJPRET

INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

DESIGN AND IMPLEMENTATION OF ONLINE PATIENT MONITORING SYSTEM USING CORTEX-M3 CORE

D.VIJAYAKUMAR[1], K.SRINIVAS[2]

1. M.TECH (ES), Dept of E.C.E, Aurora's Technological & Research Institute, Hyderabad, India
2. Assistant Professor, Dept of E.C.E, Aurora's Technological & Research Institute, Hyderabad, India.

Accepted Date: 25/12/2013 ; Published Date: 01/01/2014

Abstract: Computer communication systems and especially the Internet are playing an important role in the daily life. Using this knowledge many applications are imaginable. Home automation, utility meters, security systems can be easily monitored using either special front-end software or a standard internet browser client from anywhere around the world[2]. Web access functionality is embedded in a device to enable low cost widely accessible and enhance user interface functions for the device[1]. A web server in the device provides access to the user interface functions for the device through a device web page[9].

Keywords: Critical care, Vital signs, Ethernet, web page, TCP/IP protocols.

Corresponding Author: Mr. VIJAYAKUMAR D.

Access Online On:

www.ijpret.com

How to Cite This Article:

Vijayakumar D, IJPRET, 2013; Volume 2 (5): 68-80

PAPER-QR CODE

1. INTRODUCTION

Online data monitoring system is one of the promising trends in the era of computing in today's system automation industry. The proposed project is one such attempt of designing online patient condition monitoring system using Cortex-M3 core[5]. In this project we will develop Ethernet device drivers for Cortex-M3 core to transmit the monitored sensor data (patient condition) to internet. The System can complete the remote monitoring and maintenance operations of equipment through the network using Web browser. By introducing Internet into control network, it is possible to break through the spatial temporal restriction of traditional control network and effectively achieve remote sensing, monitoring and real-time controlling of equipments.

The main essence of this project is to design and implement online data monitoring system using ARM CORTEX M3 CORE and TCP/IP Ethernet connection for data monitoring applications[2]. The real time analog voltages are converted into corresponding digital values using the ADC pins inbuilt in LPC 1768 Cortex M-3 and transfer them to the internet through Ethernet. In this project, if the patient condition is not good then a message will be send to the doctor through GSM[11].

2. Overview Of The Project

2.1 Objectives of the project

- Understanding the architecture of ARM Cortex M3 Core.
- Design and develop an ADC conversion driver for external input signal to ADC Pin using UART.
- Create HTML webpage and linkup HTML page with the ADC output values.
- Interface GSM module with cortex-M3 using UART.
- Developing UART drivers to send SMS through GSM
- Develop an Ethernet driver for packet transfer.

2.2 The Block diagram of the project

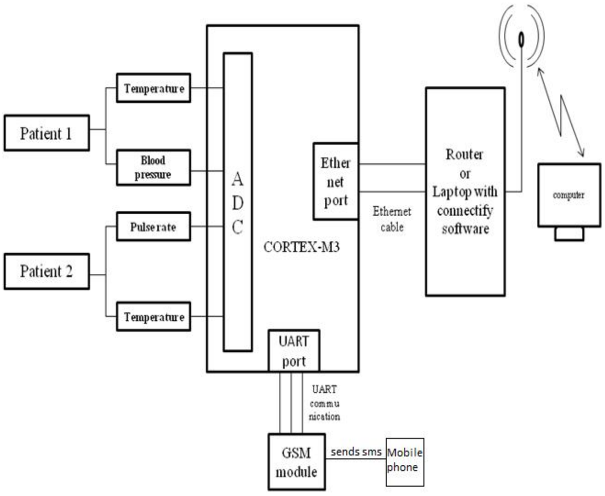

Fig: 2.2.1 Block diagram

The hardware part of project includes the following components:

- LPC 1768 H-Plus Ex Header Board
- Sensors– Heartbeat,Temperature,blood pressure
- Ethernet Cable
- GSM module
- USB-Power Supply Adapter

The software tools used in the project are as follows:

- Keil IDE
- Flash Magic
- Hotspot Connectif

3. Project Contents

3.1 Cotex m3 core processor

The LPC1768 are ARM Cortex-M3 based microcontrollers for embedded applications featuring a high level of integration and low power consumption. The ARM Cortex-M3 is a next generation core that offers system enhancements such as enhanced debug features and a higher level of support block integration. The LPC1768 operate at CPU frequencies of up to 100 MHz[5]. The LPC1769 operates at CPU frequencies of up to 120 MHz. The ARM Cortex-M3 CPU incorporates a 3-stage pipeline and uses Harvard architecture with separate local Instruction and data buses as well as a third bus for peripherals. The ARM Cortex-M3 CPU also includes an internal prefetch unit that supports speculative branching. The peripheral complement of the LPC1768 includes up to 512 kB of flash memory, up to 64 kB of data memory, Ethernet MAC, USB Device/Host/OTG interface, 8-channel general purpose DMA controller, 4 UARTs, 2 CAN channels, 2 SSP controllers, SPI interface, 3 I2C-bus interfaces, 2-input plus 2-output I2S-bus interface, 8-channel 12-bit ADC, 10-bit DAC, motor control PWM, Quadrature Encoder interface, four general purpose timers, 6-output general purpose PWM, ultra-low power Real-Time Clock (RTC) with separate battery supply, and up to 70 general purpose I/O pins. The LPC1768 are pin-compatible to the 100-pin LPC236x ARM7-based microcontroller[5].

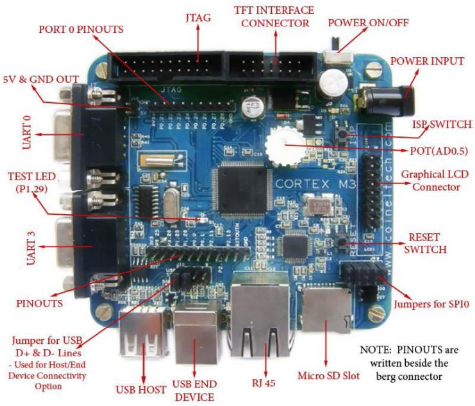

Fig 3.1.1: LPC1768 H-plus ex board

3.2 GSM Technology

GSM is a global system for mobile communication GSM is an international digital cellular telecommunication[11]. The GSM standard was released by ETSI (European Standard Telecommunication Standard) back in 1989. The first commercial services were launched in 1991 and after its early introduction in Europe; the standard went global in 1992. Since then, GSM has become the most widely adopted and fastest-growing digital cellular standard, and it is positioned to become the world’s dominant cellular standard. Today’s second-generation

GSM networks deliver high quality and secure mobile voice and data services (such as SMS/ Text Messaging) with full roaming capabilities across the world.

3.2.1 GSM Services

GSM services follow ISDN guidelines and classified as either tele services or data services. Tele services may be divided into three major categories:

➢ Telephone services, include emergency calling and facsimile. GSM also supports Videotex and Teletex, through they are not integral parts of the GSM standard.

➢ Bearer services or Data services, which are limited to layers 1, 2 and 3 of the OSI reference model. Data may be transmitted using either a transparent mode or nontransparent mode.

➢ Supplementary ISDN services, are digital in nature, and include call diversion, closed user group, and caller identification. Supplementary services also include the short message service (SMS).

3.2.2 Short Message Service

SMS stands for Short Message Service. It is a technology that enables the sending and receiving of message between mobile phones. SMS first appeared in Europe in 1992. It was included in the GSM (Global System for Mobile Communication) standards right at the beginning[11]. Later it was ported to wireless technologies like CDMA and TDMA. The GSM and SMS standards were originally developed by ETSI. ETSI is the abbreviation for European Telecommunication Standard Institute. Now the 3GPP (Third Generation Partnership Project) is responsible for the development and maintenance of the GSM and SMS standards[11].

One SMS message can contain at most 140 bytes (1120 bits) of data, so one SMS message can contain up to:

➢ 160 characters if 7-bit character encoding is used. (7-bit character encoding is suitable for encoding Latin characters like English alphabets).

➢ 70 characters if 16-bit Unicode UCS2 character encoding is used. (SMS text messages containing non-Latin characters like Chinese character should use 16-bit character encoding.)

Once the message is sent the message is received by SMSC, which must then get it to the appropriate mobile device. To do this the SMSC sends a SMS request to Home Location Register

(HLR) to find the roaming customer. Once HLR receives the request, it responds to the SMSC with the subscriber's status:

1 Inactive or active

2 Where subscriber is roaming.

If the response is "inactive", then the SMSC will hold onto the message for a period of time. When the subscriber access his device, the HLR sends a SMS notification to the SMSC and the SMSC will attempt delivery.

The SMSC transfer the message in a Short Message Delivery Point to Point format to the serving system. The system pages the device, and if it responds, the message gets delivered. The SMSC receives verification that the message was received by the end user, then categorizes the message as "sent" and will not attempt to send again. SMS provides a mechanism for transmitting short message to and from wireless devices. The service makes use of an SMSC, which acts as a store and forward system for short messages. One major advantage of SMS is that it is supported by 100% GSM mobile phones. Almost all subscription plans provided by wireless carriers include inexpensive SMS messaging service

3.3 Ethernet

My main application module regarding our project is to transfer the Digital Values which are obtained from the output of in built ADC (AD 0.5) to a web browser with the help of Ethernet and TCP/IP Protocol and display those values for another person in another location with the help of Wi-Fi. In this, I used two protocols: CSMA/CD Protocol for Ethernet Wire Transfer and TCP/IP Protocol for uploading these values to a Webpage and to display those in other PC's.In this report, first I would like to give the readers a brief idea about the Ethernet and its protocol followed by a clear description about how packets can be transferred in Ethernet and how TCP/IP protocol helps our project in data acquisition.

3.3.1 Ethernet

Ethernet is a family of computer networking technologies for local area networks (LANs). Ethernet was commercially introduced in 1980 and standardized in 1985 as IEEE 802.3. Ethernet has largely replaced competing wired LAN technologies. The Ethernet standards comprise several wiring and signaling variants of the OSI physical layer in use with Ethernet[2].

Ethernet is commonly mistaken to be a synonym for Internet. It is true that the technologies that power the Internet power Ethernet as well but there is a major distinction between the two– while the Internet is global in its nature, Ethernet is a local area network, which generally covers only a single building or premises that are close to each other. Modern technologies made it possible for Ethernet networks to span tens of kilometers but this doesn't change the local nature of Ethernet. Ethernet allows many computers to connect to one another into a network. This is done with the help of special Ethernet hardware and Ethernet protocols[2]. Ethernet is a local area technology, with networks traditionally operating within a single building, connecting devices in close proximity. At most, Ethernet devices could have only a few hundred meters of cable between them, making it impractical to connect geographically dispersed locations .Modern advancements have increased these distances considerably, allowing Ethernet networks to span tens of kilometers.

In networking, the term protocol refers to a set of rules that govern communications. Protocols are to computers what language is to humans. Since this article is in English, to understand it you must be able to read English. Similarly, for two devices on a network to successfully communicate, they must both understand the same protocols[2].

3.4 procedure of the project

3.4.1 Algorithm of project

1. Start
2. Include header files
3. Declare and initialize variables
4. Read sensor values in the form of analog values
5. Convert these analog values into digital form using in-built ADC
6. Derive formulae for different sensors of each patient
7. Send these sensor readings to HTML page
8. Send patient – 1 condition through GSM
9. Send patient-2 condition through GSM
10. Send all sensor values to Ethernet port

11. Repeat steps 4 to 10.

3.4.2 Flow chart of the project

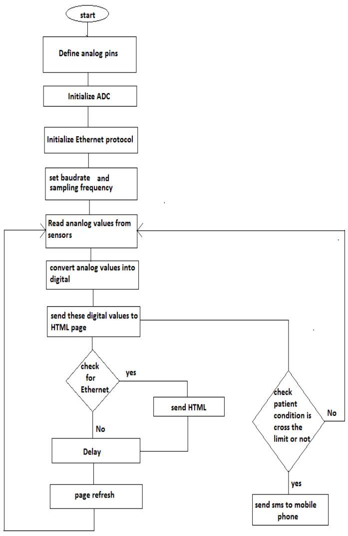

Fig 3.4.2.1: flowchart of project

3.4.3 Procedure of the project

Our project main motto is to display the ADC values on the Server's Webpage via Ethernet cable interfacing and further can be transferred over TCP network layer by creating a HTTP Server and display those values on client's Webpage and if the patient conditions are cross the limitations then a message will be sent to doctor by using GSM technology. In this, first the sensors of patient1& patient2 take the patients readings and then these analog values are given to the ADC of LPC1768 HPLUS Ex board. The ADC will convert these analog values into digital values .so for this the ADC API was developed and next I need to develop the Ethernet API. The step by step execution of our Ethernet_Adc source code is described in this section:

For this project to implement successfully, I need to design source codes for Ethernet packet driver for use with LAN controller, implementing the TCP/IP stack and providing a simple Application Interface to the user and finally implementing a dynamic HTTP Server by using the easyweb –API.I started with designing of source code for Ethernet packet driver (EMAC.c). The algorithm of this source code is described below:

I am using here the DP83848 Ethernet card. First, we declared some functions to write and read from the Physical device and wait until each operation is being completed. Then, add function to initialize the Rx Descriptors and set EMAC Receive Descriptor Registers. In the same fashion, write a function to initialize the Tx Descriptors and EMAC Transmit Descriptor Registers.

The remaining part describes about configuring port-pins for use with LAN-controller, reset it and send the configuration-sequence. Next function is initializing the EMAC and a modified function to access EMAC. First step is initializing the EMAC Ethernet controller. In this initialization, I need to power up the Ethernet, Enable P1 Ethernet pins. If mac module id is same as old mac module id, then P1.6 should be set otherwise no need to set. Reset all emac internal modules and add a delay after reset. Next step is initializing mac control registers. Put the DP83848C in reset mode. Here I need to do some conditions for the following: Check if this is a DP83848C PHY and Configure the PHY device. Configure Full/Half Duplex mode. Configure 100MBit/10MBit mode. Next, Set the Ethernet MAC Address registers.

At the start I had declared functions for Tx and Rx descriptors. Those should be initialized here. Next, Receive Broadcast and Perfect Match Packets. Enable EMAC interrupts. Reset all interrupts. Enable receive and transmit mode of MAC Ethernet core. It reads words in an order from Rx Buffer and Rx Frame port and copies them to the MCU Memory. Reads the length of the received Ethernet frame and checks if the destination address is a broadcast message or not. It requests space in EMAC memory for storing an outgoing frame and check if Ethernet controller is ready to accept the frame we want to send. Copies bytes from MCU-memory to frame port.

Next module is to write an easy web-API function using TCP/IP. For this, initialize the LAN-controller, reset flags, starts timer-ISR. Listen on 'My IP:TCP Local Port' for an incoming connection and try to establish a connection between 'My IP:TCP Local Port' and 'Remote IP:TCP Remote Port' then releases the receive-buffer and allows easy web to store new data. Rx-buffer must be released periodically, else the other TCP get no ACKs for the data it sent. Transmit data stored in 'TCP_TX_BUF'. Number of bytes to transmit must have been written to 'TCPTx Data Count' and data-count MUST NOT exceed' MAX_TCP_TX_DATA_SIZE'. Easy web's 'main()'-function must be called from user program periodically (the often - the better) and it handles network, TCP/IP-stack and user events.

Most important part is to check for the IP type (IPv4), get IP's frame length, get source IP, get destination IP, etc. and this tcpip.c links the broadcasted message received from the MAC address to the given IP address via tcp/ip protocol i.e., it links Ethernet and the HTTP server webpage. Coming to the main part of the program, I need to implement a dynamic HTTP-server by using the easy web-API. Here I initialized the page count and the ADC values to 100 and 0. For giving a timing interval of 10 ms for refreshing the webpage, I used Systick Timer Interrupt. It is enabled in the NVIC using the appropriate Interrupt Set Enable register. Systick timer is intended to generate a fixed 10ms interrupt for use by the OS. This timer is assigned to GPIO2 -> 2.0 pin with XOR operation to toggle led 0.

Next, coming to the main part, I started with setup of core clock and generated interrupt every 10ms. For the ADC input, I connected Potentiometer to AD0.5 pin i.e,. to P1.31, set clock frequency to 12 MHz and enable this pin. Clear HTTP-server's flag register and this HTTP server function is described later. Set port we want to listen to (Local port= HTTP port). Here, I created an infinite while loop and in it I write a command to listen for incoming TCP-connection, to handle network and easy web stack and finally implement a HTTP Server.

The project's final module is to design a HTTP server. This function implements a very simple dynamic HTTP-server. It waits until connected, and then sends a HTTP-header and the HTML-code stored in memory. Before sending, it replaces some special strings with dynamic values (A/D converter results). It establishes a server connection and checks if somebody had connected to our IP, checks if remote TCP sends data and throw it away. In this way bytes of information can be transferred from Ethernet to the HTTP server webpage. This whole content can be only viewed in a HTML webpage. For this, I designed a html webpage. The ADC values are displayed in the analog voltage scale on this webpage and at the bottom of the webpage, you can see a page count.

Finally, with the help of Hotspot Connectify software, I created a server and if any PC connected to this private network, they are able to see this webpage by typing the particular IP address in their Internet browser.

4. Results And Outputs

4.1 Data Monitoring System displayed on a webpage

Fig: 4.1 Data Monitoring System displayed on a webpage.

The real time analog voltages are converted into corresponding digital values using the ADC pins inbuilt in LPC 1768 Cortex M-3 and transfer them to the internet through Ethernet and by using tcp/ip protocols these values are stored in specified address. When we enter that address in the browser then those values are display on the browser. This webpage has analog readings ranging from 0V to 3.3V. For every 2 seconds, the webpage can be refreshed automatically and Page Count can be displayed below the analog output. The Data Monitoring System displayed on a webpage is shown in above fig 4.1.

4.2 Receiving message through GSM

4.2.1 Receiving message through GSM

The analog voltages are converted into corresponding digital values using the ADC pins inbuilt in LPC 1768 Cortex M-3.By using these values derive formulae for different sensors of each patient and based on this values the message will be send to mobile through GSM. If the patient1 temperature is high then a message is send to mobile through GSM immediately that is shown in the above figure 4.2.1.

5. CONCLUSION & FUTURESCOPE

The project "design and implementation of online patient monitoring system using cortex-m3 core" has been successfully designed and tested. It has been developed by integrating features of all the hardware components and software used. Presence of every module has been reasoned out and placed carefully thus contributing to the best working of the unit. Secondly, using highly advanced lpc1768 board and with the help of growing technology the project has been successfully implemented.

The Porting of Free RTOS Kernel onto the Cortex M3 Core Processor for Multitasking .So by this Free RTOS we will be getting the multitasking operation in our project and we will be controlling patient condition when the vital signs are exceed their ranges by giving some treatment with robotics. These are the future scopes of this project.

6. REFERENCES

1. Jean J Labrosse, "Embedded Systems Building Blocks" R&D Books, pp 61-100.
2. An Axelson "Embedded Ethernet and Internet Complete" Lakeview Research LLC, 2003.
3. Josephy Yiu "The Definitive Guide to ARM Cortex M3" Newness,©2007.
4. Pont,Michael J "Embedded C",Pearson Education ©2007.
5. "LPC 17xx User Manual", NXP Semiconductors.
6. "Web Convert User Manual", Precidia Technologies, ©2002.
7. HTML Coding from w3schools.com
8. Description about entire Lpc1768 and Ethernet form the Data sheet of Lpc17xx manual.
9. Creating HTML page from www.artima.com

10. Arul Prabahar A, Brahmanandha Prabhu, **"**Development of a Distributed Data Collection System based on Embedded Ethernet" 978-1-4244-9799-71111$26.00 ©2011 IEEE.

11. "GSM basics History" on Tutorial point, Wikipedia and private line.