DESIGN AND IMPLEMENTATION OF ONLINE PATIENT MONITORING SYSTEM

Harsha G S
Department of Electronics & Communication
Channabasaveshwara Institute of Technology, Gubbi, 572216, India

ABSTRACT
Patient's condition is continuously monitored and web access functionality is embedded in a device to enable low cost widely accessible and enhance user interface functions for the device. A web server in the device provides access to the user interface functions for the end user through the device web page. This will be dynamically updated with the recent monitored patient values from the module. Upon which the end user in the network can avail live values on the webpage.

KEYWORDS *- Embedded; Ethernet; TCP/IP Protocols; LPC1768; Cortex M3.*

I. 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 signing online patient condition monitoring system using Cortex-M3 core [2]. In this project we will develop Ethernet device drivers or Cortex-M3 core to transmit the monitored sensor data (patient condition) to internet [1]. The system can complete the remote monitoring and maintenance operations of equipment through the network using Web browser [7]. 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 equipment. 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 [3].

II. ORGANIZATION OF THE PAPER

Introduction tells about the problem statement, existing and proposed systems, and disadvantages of old system as well as advantages of newly designed system is given. **Related work** explains the different terminology which are quite useful to understand Ethernet protocol which is the main communication protocol used in the system. **Requirement** lists the hardware and software specification for this project. It also describes the overall description of project, product perspective, user characteristics and specific requirements. Here different design constraints, interface and performance requirements explained. **System Definition** deals with the clear definition of the properties and characteristics of the embedded system prior to starting hardware and software development essential to achieve a final result that matches its target specifications. **System Design** deals with the advanced software engineering where the entire flow of the project is represented by professional data flow diagrams and sequence diagrams. **System Implementation** section explains coding guidelines and system maintenance for the project. **Results** explains in details about the outcome of the experiment and compares it with the result obtained in existing system. **Conclusion and Future work** section describe the summary of the related work and future enhancements of the proposed system. **Reference** section contains the papers and other documents referred in the research. **Author Profile** gives a brief introduction about the author.

III. RELATED WORK

Traditional method of deployment of Monitoring devices is based on the simple circuitry or low level controllers. This makes them inefficient to cope up with the speed they are needed to be processing the tasks and lacks functionalities like Ethernet support, SD card support, UART, Timers and Counters [6]. So they fail to communicate with the hospital personnel or the doctor directly and results in serious situations. But automating this process is one of the modern approaches for this problem. This is already implemented in many hospitals. Although the data is being monitored automatically in those systems, it is not capable of reaching the Doctor or concerned person or to bring it to their notice[8].

IV. REQUIREMENTS

These requirements are required in order to carry out the execution of project.

A. Hardware Requirements

Hardware requirements for this project are listed as follows:

- LPC 1768 H-Plus Ex Header Board: Cortex M3 Development board brought up by Coinel Technologies Ltd., Bangalore.
- Graphical LCD: 128*68 Pixels display with white backlit LED.
- Sensors: Pulse Rate (NS13), Temperature (LM35) and Gas sensor used to obtain the patient body conditions.
- Buzzer: Used as an actuator to produce alarm when the threshold is reached.
- GSM module – SIM300: To send SMS alert to the Doctor/concerned person.
- CooLinkEx J-Tag Debugger: To debug the program or to burn the program hex code into the processor.
- Ethernet Cable: To connect the network with the board.
- USB-Power Supply Adapter: To power up the LPC1768 HPlus ex board.
- D-Link Wi-Fi Router: To create wireless network.

B. Software Requirements

Software requirements for this projects are listed as follows:

- Keil IDE : Keil development environment to edit , compile and debug code. It can even burn the code to the processor using plugins required for J-Tag code burner.
- Flash Magic : To burn the Hex file into the microcontroller.
- Coocox CoLinkEX J-Tag Debugger : To debug code live on the hardware with step by step feature. We can also use software breakpoints.
- WebConverter V1.0 : To convert the webpage from HTML format into array of strings.
- MyPublicHotspot : To create WLAN using the Wi-Fi modem of the Laptop.

V. SYSTEM DEFINITION

Defining a system based on the hardware and software is an important stage. This contains four steps in defining a system with Ethernet connectivity for an embedded product. In these four stages, we consider the embedded system as a black box. Which has a clear definition of the properties and characteristics of the embedded system prior to starting hardware and software development is essential to achieving a final result that matches its target specifications.

A. Specifying Required Functionality

The first thing is to focus on the aim of our system being designed. This is related to the application and can encompass virtually anything that can be done using a MCU with considerably high speed. One such example can be seen from the figure 1.

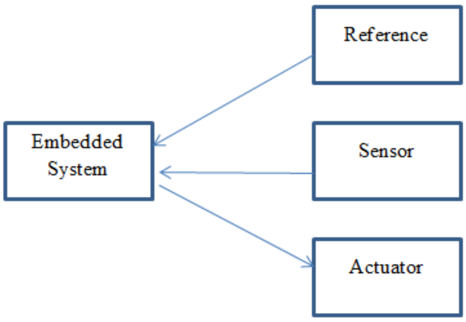

Figure 1. System Functionality Example.

B. Specifying Access Method

After we specify required functionality the next thing is to specify the accessing method of the requirement. That is how we can access the output being generated from the controller.

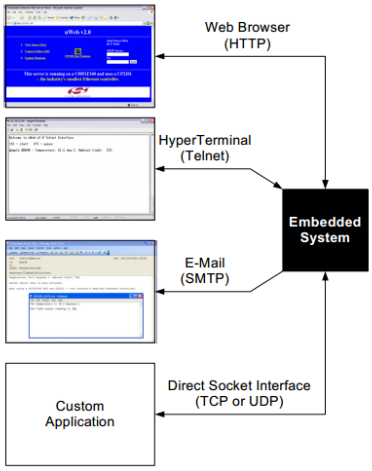

Figure 2. User Interface Options.

Figure 2 shows the black box representation of the embedded system and how the various access methods can be used to monitor and control the embedded system.

In our project we can go for one or more of the commonly used access methods used, they are:

- Using a web browser.
- Using HyperTerminal.
- Having the embedded system send e-mail.
- Using a custom application.

C. Specifying Configuration Method

Each and every device which is connected within a network will comprise of both a MAC address as well as IP address which will be unique for that system. These are used to identify, locate and communicate each other. Embedded systems with LPC1768 IC needs to obtain only the IP address because the MAC address of the same will be preprogrammed in the flash memory of the IC.
There are four common configuration methods available to choose from:

- Automatic Network Configuration.
- Automatic Network Configuration with Netfinder.
- Static Network Configuration.
- Static Network Configuration with Netfinder.

D. Specifying Field Re-Programmability Requirements

The final part of the system definition is determining the field re-programmability requirements of the embedded system.

The options for field re-programmability are:

- No support for field re-programmability.
- Re-programmability using a 3 or 5 pin header.
- Re-programmability using a 10-pin header.
- Re-programmability using a bootloader.

VI. SYSTEM DESIGN

System design is a transmission phase from a user oriented documented system to a purely programmatic oriented system for programmer's database personnel. The system design makes the high level decision about the overall architecture of the system. The system design phase provides the understanding and procedure details necessary for implementing the system recommended study. The target system is arranged into subsystems based on the analysis structure and the proposed architecture.

A. System Architecture

The figure 3 represents the functional architecture of the system. Here the pulse rate, temperature and gas sensors are the inputs for the system which will be used to obtain the parameters which are the input for the system. Using the analog input from these sensors the processor computes for the data required and handles the situation based on the requirements of our system.

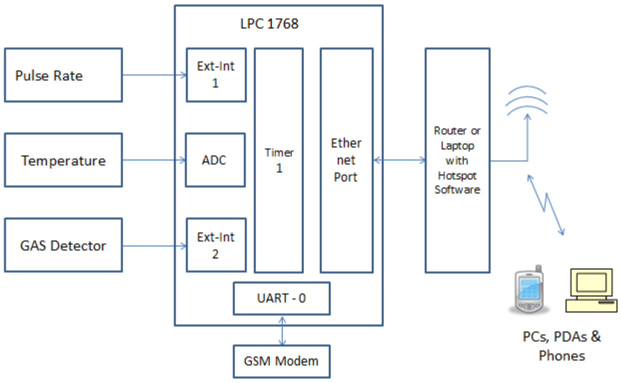

Figure 3. Block Diagram of the System.

After obtaining digital values for the sensor outputs, they are checked for the threshold levels based on the normal values expected and if the threshold is passed beyond then an alert is sent to doctor using GSM modem. Then the web page which is dynamically updated with the values shall be pushed to the network. This can then be accessed on a device within that network using its web browser pointed to the defined IP address.

B. System Flow Diagram

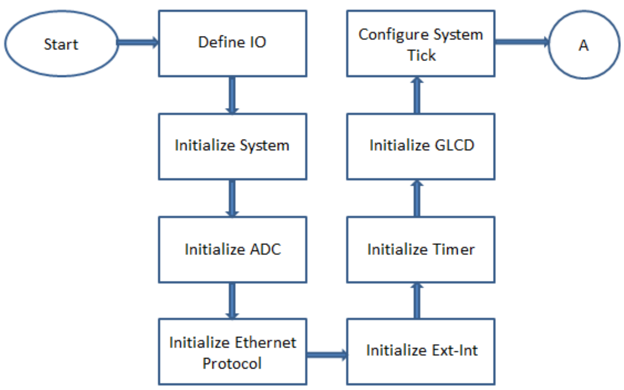

Figure 4. Flowchart of the Project.

The process starts with initialization of the input outputs and configuring system parameters like frequency. Then the peripheral devices are powered and configured for the purpose. Declaring and initialization of the variables are to be made along with the prototype inclusion. After initialization of the peripherals like external interrupt, timer interrupts, Ethernet, ADC etc. The sensor data is acquired in analog form and are converted into digital values as per the unit of the measured variable and are checked with threshold levels.

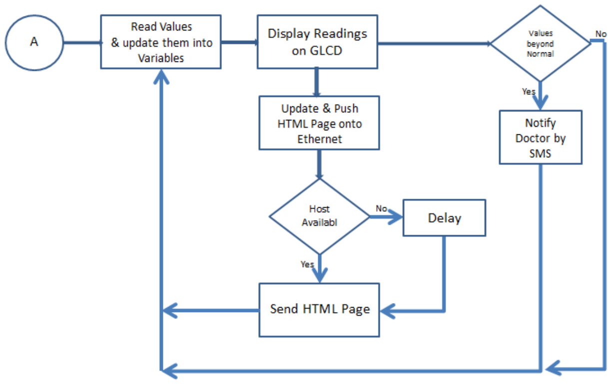

Figure 5. Flowchart of the Project (Cont..).

The values are updated on the local Display. If the threshold level is reached then the SMS alert is sent to the Doctor number using GSM modem then dynamically updated webpage with recent values is pushed to the Ethernet. This can be accessed from browsers of the devices in the same network.

VII. SYSTEM IMPLEMENTATION

Implementation is the realization of the application or execution of the plan given in the design. This section deals with detailed description of how the project goal is achieved.
This phase continues until the system is operating in production accordance with the defined user requirements. Successful implementation may not guaranteed improvement but it will prevent improper installation.
The whole Implementation part of this project is divided into three different levels, they are:

- Hardware Design
- Software Generation
- Application Development

A. Hardware Design

With a system definition in place, it is now time to start designing the hardware. The hardware design flow consists of 5 steps corresponding to the 5 sections of a schematic for an embedded system with Ethernet connectivity.
They are listed as follows:

- Custom application circuitry - sensors, indicators, and other application-specific circuitry.
- MCU - the main system controller.
- Ethernet Controller - provides the MCU with the capability to send and receive data over a network.
- Ethernet Connector - the RJ-45 connector, magnetics, and link/activity LEDs.
- Power circuit - provides the embedded system with regulated 3.3 V power.

B. Software Generation

In this step of the system design process, we will be generating the software that interacts with the LPC1768 Ethernet module to provide the embedded system with Ethernet connectivity. Using the TCP/IP Configuration Wizard, this step is one of the easiest steps in the entire system design process.

C. Application Development

The application code that will implement the required system functionality specified in the first question of the system definition must co-exist and share resources with the TCP/IP Library. To develop the code, we will need a good understanding of how the TCP/IP Stack operates.

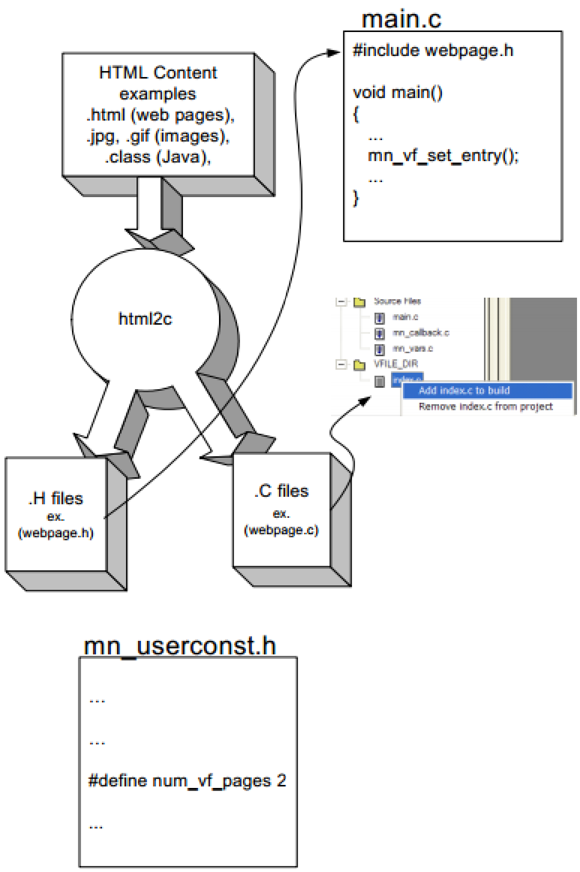

Figure 6. Adding Dynamic HTML Content.

The figure 6 shows how we can add the HTML code on to the loop and it also tells how we can update the values dynamically onto the webpage.

VIII. RESULTS

The output is seen on any of the devices within the same network just by accessing the IP address assigned for the webpage from any web browser. Snapshot of the result webpage can be seen in the figure 7.

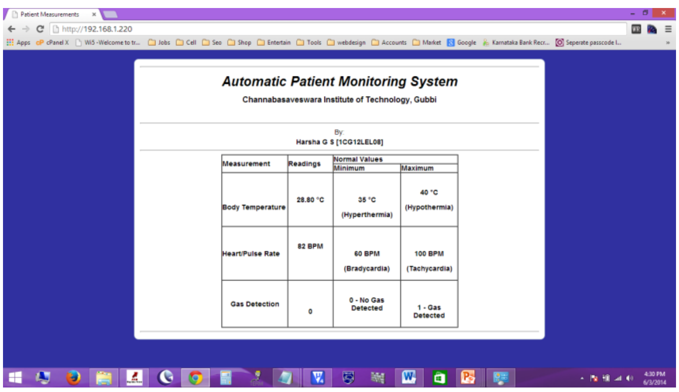

Figure 7. Adding Dynamic HTML Content.

IX. CONCLUSION & FUTURE WORK

A. Conclusion

- Three efficient sensors are interfaced with the controller & seen working fine.
- TCP-IP libraries are used to communicate through Ethernet.
- Cortex controller is proved to be highly flexible & efficient.
- Tasks are efficiently managed in the loop so as to perform all functions properly using available CPU time.
- Software sanity is obtained on interfacing with all the peripherals.
- WLAN is created and found to be working fine with all the clients in the network.
- Outputs are checked on GLCD, mobile phones, tablets, laptops and desktop pcs.

B. Future Work

- Actuators are automatically controlled based on the body conditions measured.
- Provisions to be made for the doctor to control these actuations from the webpage.

REFERENCE

[1] Dhruva R. Rinku, Mohd arshad , “Design and implementation of online data acquisition and controlling system using cortex m3core.” IJESAT - Volume-3, Issue-5,259-263 ,©2013.
[2] S.Navaneethakrishnan, T.Nivethitha, T.Boobalan, “Remote patient monitoring system for rural population using ultra low power embedded system”. ISSN: 2319-5967 ISO 9001:2008 Certified ,©2013.
[3] IEEE Standards for Local Area Networks Carrier Sense Multiple Access With Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications, ANSI/IEEE Std 802.3-1985, 1985.
[4] Jean J Labrosse, “Embedded Systems Building Blocks” R&D Books, pp 61-100.
[5] An Axelson “Embedded Ethernet and Internet Complete” Lakeview Research LLC, 2003.
[6] Josephy Yiu “The Definitive Guide to ARM Cortex M3” Newness,©2007.
[7] Pont, Michael J “Embedded C”, Pearson Education ©2007.
[8] “LPC 17xx User Manual”, NXP Semiconductors.
[9] 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.
[10] “GSM basics History” on Tutorial point, Wikipedia and private line.

AUTHOR PROFILE

Harsha G S (M’23) was born in Karnataka, India in 1990. He received the B.E. and M.Tech. Degrees from Visvesvaraya Technological University, Belgaum, India, in 2012 and 2014 respectively on Electronics. He is the founder of many technology blogs www.GoHarsh.com & www.TecLogs.com, and he also founded a webdesign company in the early 2012, www.WebsCrunch.com. His main areas of research interest are Embedded systems, VHDL, Image Processing and Web Design.