Design Telemetry Monitoring System Based on ARM

S.B.Bhosale, PG Student
Sinhgad College Of Engineering, Vadagaon (Bk), Pune
bhosalebsharvari@gmail.com

Abstract- Network controlled device has its own demand and application with fast development in technology. The system of remote data monitoring is designed based on the core processor of LPC2378 in the paper. The proposed paper is design to add security issue to any general premises. In this paper, telemetry monitoring based on embedded platform with Ethernet port support is designed. Method of embedded system interface based on Ethernet and ARM7LPC2378 processor. The platform has Ethernet interface on it DP83848H can directly connect with the Ethernet MAC, to develop system. A data can be transmitted transparently through Ethernet interface unit to the remote end desktop computer. This design has the advantage of cost-effective, easily realized, stable and reliable transmission and so on. The simulation of the system is carried on network analyzer and it is seen that the no. of bytes on wire matched to captured one.

Keywords: *Embedded Ethernet; ARM; PIR sensor.*

I. INTRODUCTION

Embedded system is a special computer system that bases on computer technology, focuses on application, software and hardware customizable, suitable for the strict requirement of application system on function, reliability, cost, volume and power. It is widely used in military, civil electronics, household appliances and consumption electronic products in recent years. But most of the embedded systems are used independently at current stage, CAN, RS-232 and RS-485 are the most commonly used technologies to deal with the communication between multiple microprocessors in industrial control fields and has the disadvantages such as low transmission rate, limited coverage, relatively less communication protocols, etc. which cause it very difficult to perform flexible remote access and management [1]. The Ethernet (IEEE 802.3) is the most mature and widely used LAN technology, connecting the embedded device to the network device such as Hub, switch thus realizing a flexible real time control and monitoring has already become an inevitable development trend of embedded technology [4]. In this paper, the design and implementation process of a monitoring system based on ARM7. ARM 7 LPC2378 development kit with DP83848H module, are connected using a crossover Ethernet cable RJ 45. Data is transmitted from DP83848H device to desktop computer. On the transmission side, user-entered data is compiled into an IEEE 802.3 frame; on the reception side, data is extracted from the frame and displayed through the network analyzer.

II. SYSTEM DESIGN

Having 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. As Ethernet is a one of the most popular network protocol so it is needed to understand network protocol stack concepts prior implementation [2]. A computer's network protocol stack consists of the modules involved with networking [5]. Figure 1 shows an example of a network protocol stack for a computer that connects to an Ethernet network and supports common Internet protocols. At the bottom of the stack is the hardware interface to the network cable. At the top of the stack is a module or modules that provide data to send on the network and use the data received from the network. In the middle there may be one or more modules involved with addressing, error-checking, and providing and using status and control information. In transmitting, a message travels down the stack from the application layer that initiates the message to the network interface that places the message on the network. In receiving, the message travels up the stack from the network interface to the application layer that uses the data in the received message. The number of layers a message passes through can vary. For some messages that travel only within a local network, the application layer can communicate directly with the Ethernet driver. Messages that travel on the Internet must use the Internet Protocol. Messages that use the Internet Protocol can also use the User Datagram Protocol or the Transmission Control Protocol to add error checking or flow-control capabilities. For this system UDP protocol is used. A UDP datagram has a header, followed by a data portion that contains the application data.

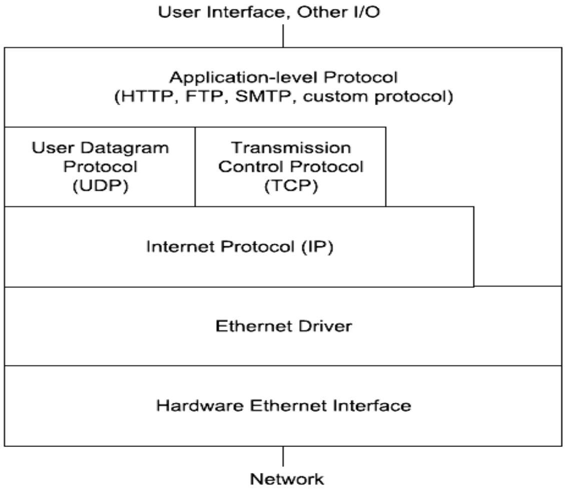

Figure 1. Network Protocol Stack

UDP includes fields for specifying ports and optional error-checking, but no support for flow control. Windows and many development kits for embedded systems include support for UDP[5]. The LPC2378 microcontroller board uses UDP protocol packet to transmit data to a remote computer. UDP protocol is one of protocols in the TCP/IP protocol suite that is used in the place of TCP when a reliable delivery is not required. There is less processing of UDP packets than the one of TCP. UDP protocol is widely used for streaming audio and video, voice over IP (VoIP) and video conferences, because there is no time to retransmit erroneous or dropped packets. The application provides data to send on the network and uses data received from the network. An application often has a user interface that enables users to request data from a computer on the network or provide data to send on the network. The data that the application sends and receives may be anything: a single byte; a line of text; a request for a Web page; the contents of a Web page; a file containing text, an image, binary data, or program code; or anything that a computer wants to send to another computer in the network. The data sent by an application follows a protocol, or set of rules, that enables the application at the receiving computer to understand what to do with the received data. An application may use a standard protocol such as the hypertext transfer protocol (HTTP) for requesting and sending Web pages, the file transfer protocol (FTP) for transferring files, or the simple mail transfer protocol (SMTP) or Post Office Protocol (POP3) for e-mail messages[4]. Here FTP is chosen. In Ethernet networks, a unique hardware address identifies each interface on the network. IP addresses are more flexible because they aren't specific to a network type. A message that uses IP can travel through different types of networks, including Ethernet, token-ring, and wireless networks, as long as all of the networks support IP. A communication in a local network that doesn't use TCP or UDP may not require IP[6]. Instead, the application layer may communicate directly with a lower layer such as the Ethernet driver. In Hardware Ethernet Interface layer LPC2378 board with RJ 45 connector is present.

III. THE HARDWARE DESIGN

The system is mainly composed of Processor module is the core part of the design, in which the ARM chip LPC2378 used. LPC2378 is an ARM-based microcontroller for applications requiring serial communications for a variety of purposes. These microcontrollers incorporate a 10/100 Ethernet MAC(DP83848H), USB 2.0 Full Speed interface, four UARTs, two CAN channels, an SPI interface, two Synchronous Serial Ports (SSP), three I2C interfaces, an I2S interface, and a Minibus (LPC2378 only: 8-bit data/16-bit address parallel bus)[8].

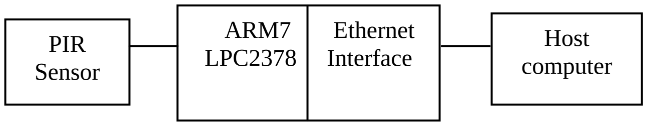

Figure 2. System Block Diagram

For the purpose of testing, the microcontroller is used Passive Infra Red (PIR) sensor for motion detection. The PIR (Passive Infra-Red) Sensor is a pyroelectric device that detects motion by measuring changes in the infra-red levels emitted by surrounding objects. This motion can be detected by checking for a high signal on a single I/O pin, in the current development [11]. Here #555-28027 PIR sensor manufactured by Parallax used. In the LPC2378 board have in built Ethernet interface module is provided and the collected data by #555-28027 PIR sensor is uploaded to a PC via Ethernet interface as shown in Figure 2.In the Ethernet interface module, Ethernet controller DP83848H (PHYTER mini extended temperature qualified to AEC-Q100 grade 2) are divided into two layers according to its functions .One is media access controller (MAC) layer and the other is network physical (PHY) layer. They correspond to Layer 2 and Layer 1 in ISO model. MAC layer provides the treatment on data sending and receiving. It also provides an interface to PHY through an internal medium independent interface [10] as shown in Figure 3.

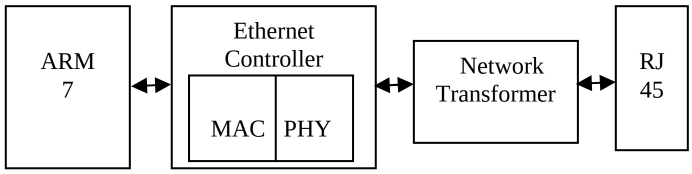

Figure3.Ethernet controller and Ethernet interface

In this design, LPC2378 supports Ethernet interface and network transformer HR601680 which is connected between Microcontroller Lpc2368 and connector RJ45 is used to improve the signal anti-interference capability. HR601680 is a 1:1 transformer with a smaller package and it supports 10M/100M Ethernet.

IV. ETHERNET COMMUNICATION

In the design of network communication, we should handle Ethernet frame accepting and Ethernet frame transmitting, According to the received packet type, choose a different approach; the length of frame cannot exceed the IEEE802. 3 standards (1 514 bytes) in accepting, when transmit Ethernet frame, we should add 14 bit Ethernet header logo on the transmission data, namely, increase the packet length. The typical Ethernet frame is shown in Table 1.

Table 1. The typical Ethernet Frame

Preamble	Start of frame delimiter	Destination address	Source address	Length	Data	CRC
7 bytes	1 bytes	6 bytes	6 bytes	2 bytes	46 to 1500bytes	4 bytes

Ethernet Header	IP Header	UDP Header	Data (Ethernet stream Structure)	Ethernet Trailer

Enabling and disabling of reception & transmission of data is performed using various registers provided by LPC2378.Using those registers how can transmit or receive data is described below,

A) Enabling and disabling reception

After reset, the receive function of the Ethernet block is disabled [8]. The receive function can be enabled by the device driver setting the RxEnable bit in the Command register and the “RECEIVE ENABLE’ bit in the MAC1 configuration register (in that order). The status of the receive datapath can be monitored by the device driver by reading the RxStatus bit of the Status register. Figure illustrates the state machine for the generation of the RxStatus bit. After a reset, the state machine is in the INACTIVE state. As soon as the RxEnable bit is set in the Command register, the state machine transitions to the ACTIVE state. As soon as the RxEnable bit is cleared, the state machine returns to the INACTIVE state. If the receive datapath is busy receiving a packet while the receive datapath gets disabled, the packet will be received completely, stored to memory along with its status before returning to the INACTIVE state. Also if they Receive descriptor array is full, the state machine will return to the INACTIVE state.

B) Enabling and disabling transmission

After reset, the transmit function of the Ethernet block is disabled. The Tx transmit datapath can be enabled by the device driver setting the TxEnable bit in the Command register to 1. The status of the transmit datapaths can be monitored by the device driver reading the TxStatus bit of the Status register. After reset, the state machine is in the INACTIVE state. As soon as the TxEnable bit is set in the Command register and the Produce and Consume indices are not equal, the state machine transitions to the ACTIVE state. As soon as the TxEnable bit is cleared and the transmit datapath has completed all pending transmissions, including committing the transmission status to memory, the state machine returns to the INACTIVE state. The state machine will also return to the INACTIVE state if the Produce and Consume indices are equal again i.e. all frames have been transmitted [8].

V. SYSTEM SOFTWARE DESIGN

Data transmission over the Ethernet medium can be segmented into three main steps:

1. Initialization of the EMAC, PHY and the CPU:
   (a) Initialize EMAC Control Registers.
   (b) Initialize PHY registers.
   (c) Enable interrupts after setting interrupt handlers for the EMAC interrupts.
2. Transmission of frame:
   (a) Generate fields within the frame.
   (b) Generate an Ethernet packet with the help of the frame.
   (c) Create a descriptor table for the Ethernet packet.
   (d) Transfer control of the buffer to EMAC for Transmission.
   (e) Once the packet is transmitted interrupt occurs and the program is looped back to initial starting point.
3. Reception of frame:
   (a) Check the status of reception.
   (b) Copy data into memory location specified by the user.
   (c) Update the EMAC Interrupt Status .Register and loop to initial starting point.

In the Ethernet transmission task, in order to receive the data from Ethernet in the system, the local IP address and subnet mask must be set firstly, and the appropriate UDP port is opened to monitor whether there are data in UDP port[3]. After transmitting data it is very important to verify that data is received or not on network itself. In troubleshooting network problems, it's often helpful to be able to view the network traffic to find out exactly what is (and isn't) transmitting. A network analyzer makes this possible and can be extremely helpful in tracking down problems. A network analyzer may be software only or it may be a hardware device that runs analyzer software.

VI. SYSTEM TESTING

After the software and hardware designs have been completed, the ultimate generated codes are compiled and downloaded to the target system for testing. Here SPJ's LPC2378 Target board and SCARM compiler used. **SCARM** is SPJ Systems' C Compiler for ARM. It also includes an IDE and other tools like Visual Code Generator (VCG) and terminal emulation utility (SPJTerm) [9].

Wireshark is a network packet analyzer. A network packet analyzer will try to capture network packets and tries to display that packet data as detailed as possible. Wireshark to view the contents of Ethernet frames along with timing and other information Wireshark from any PC in the network you're monitoring. To ensure that the frames want to see are visible, the PC running Wireshark should be in the same collision domain as the computer whose traffic monitoring.

VII. RESULT ANALYSIS

Figure. 4 indicate the monitoring interface of captured data using Wireshark Network Protocol analyzer; from it we know the system can fulfill the design requirements. The send data are control commands sent from monitor computer to equipment while the receive data are state feedback data received from equipment. The number of bytes in wire (RJ45) matched to captured one. Here windows showing data captured from sample frame transmitted through RJ 45.

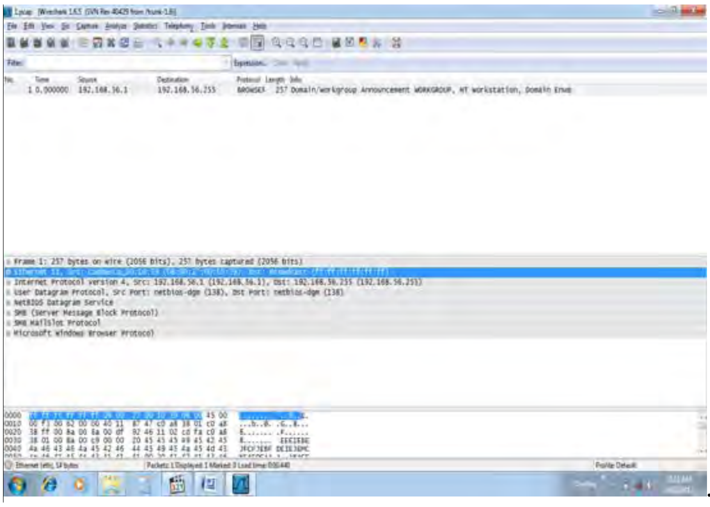

Figure 4. The Wireshark Network Protocol Analyzer decodes and displays Ethernet traffic.

VIII. CONCLUSION

The system developed will be very useful to increase security against the theft & general premises resources. While the Ethernet is most mature and widely used data link layer technology for Internet, the scheme proposed in this paper not only can be used in connecting the embedded device to LAN, but also as a data link and physical layer support for embedded web system. This system can be used widely in the domain of data-acquisition and remote data transportation, access control, security, attendance system and industrial on-site network monitoring etc.

ACKNOWLEDGEMENT

The authors are grateful to our principal, Dr.S.D.Lokhande, for providing necessary facilities towards carrying out this work. We acknowledge the diligent efforts of our Head of the Department Dr.A.D.Jadhav in guide us towards implementation of this idea.

REFERENCES

[1] Wang Jing,He Huiming, ARM-based Embedded Video Monitoring System Research,2010 IEEE, 978-1-4244-5540
[2] BoQu, Daowei Fan, Design of remote data monitoring and recording system Based on ARM, 2010 2nd International Conference on Industrial and Information Systems.
[3] V L Varaprasad Nagula, M.Venkateswara Rao, T.Raghavendra Vishnu, Embedded Ethernet Monitor And Controlling Using Web Browser, International Journal Of Engineering Science & Advanced Technology Volume - 2, Special Issue - 1, 1 – 5
[4] Fred Halsall, Data communication, Computer Networks and Open Systems [M].beijing: Machinery Industry Press, 2004, pp.572-580
[5] Wayne Wolf. Computers as Components Principles of Embedded Computing System Design [M), Beijing: Publishing House of Machinery Industry. 2002, 10-70.
[6] 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.
[7] YU Cheng-bo, LIU Jie, and TAO Hong-yan, "Reseach on remote monitoring technology of equipment," Information and Control, Magn. China, vol.31 (3), pp.236-240, June 2002.
[8] LPC2378 Technical manual & user manual
[9] SPJ's user's manual for MiniARM 23xx
[10] http://www.nxp.com
[11] www.parallax.com