Multi-Serial Interface Bridge

G. Kirubakaran[1], V. Natarajan[2]

[1,2]*Department of Electronics and Communication Engineering, SRM University*

Abstract— **In designing Embedded System, withstanding communication between several different interfacing protocols which are attached to a single main processor is one of the challenging tasks. Also in industries, systems are becoming very complex. Industries system needs to test the site equipment and also environmental conditions so that it can track the state of system in real time. This is more relevant in real time data acquisition and control systems. Also designing and implementing the online embedded web server is challenging part. Popular serial interfacing protocols include: USB, SPI/SSP, CAN, Ethernet, etc., for communication between integrated circuits for low/medium data transfer speed with on board peripherals. The main quality in the designed circuit is using Real Time Operating System (RTOS) on ARM series 32-bit processors, MCB 2300. Compared with the normal ways to control and data acquisition, the device based on the embedded system offers better ability to modify it easily, with an overall design for reliability, durability and ease of installation. This paper explicates the hardware architecture and real time multitask software process based on RL-ARM Library. There are enormous use for such system in control and data acquisition systems. Especially, in network interfaces with various protocol layers, it can be used as a clever gateway or router and so forth.**

Keywords— ***ARM7; communication; RL-ARM; RTOS, Web Server.***

I. INTRODUCTION

For expanding or connecting networks several devices such as repeaters, bridges, routers and gateways are used, due to the distances between networks or differences in their network layers. In connecting networks, one of the most significant factors is their protocols i.e. the way that they will interface with their devices. So based on the protocol used in each network, the kind of the interface device will vary. For example relating to two networks with the same protocol, routers are used but for connecting to two networks with different protocols in each layer of their network, nothing can be used other than gateways.

Based on the above mentioned a suitable circuit has been designed in a small box for connecting each kind of networks i.e., a gateway. There are many kinds of interface protocols such as RS232, RS485, CAN, USB, Ethernet, I2C, SPI/SSP etc., for communication between processors or controllers and PC, for low/medium data transfer speed with on board peripherals. Controlling different devices with such different protocols is not so easy and it may need a supervisor system to control all the devices safely. For controlling on-time and without power wasting, a simple Real Time Operating System (RTOS) has been used. The designed circuit has used an ARM7 series processor, LPC2478 as the main device of its circuit.

A. Problems and challenges in existing system

In embedded system design, managing communication among various bus interfaces and attaching multiple systems with different interfacing protocols to a main processor is one of the challenging tasks. There are so many projects (e.g. [1]-[3], [6]) existing which interface several serial devices to any microcontrollers but the problem is nothing were ported by RTOS. So the problem of existing system is, it doesn’t manage wastage of power in real time and also it can’t manage communication between various protocols.

B. Overcoming problems and challenges in existing system

This project’s main theme is to design a smart circuit to interface with some protocols such as RS232, USB, CAN, and the most important of all: Ethernet. The main feature in the proposed circuit is using Real Time Operating System (RTOS) which helps the system to overcome the problems in existing systems.

II. METHODOLOGY

A. Proposed Work

An embedded circuit based on ARM processors is presented for interfacing with different standard protocols such as RS232, USB, CAN, and LAN was designed. High precision data acquisition can be realized by the embedded system as well. For better management, working with different protocols and also, for power conservation a real time operating system, RL-ARM (RTOS from Keil) was used on MCB2300, 32- bit processor by Philips Company. The main algorithm is, the processor sends request read data command to lower computer at intervals; sending data to Ethernet and inputting IP address of the designed box; data of the lower computer will be measured and controlled real time. Since the embedded system has the capability to handle Multi-Tasks and can run real time operation systems, this can be achieved so easily.

B. General Block Diagram of Proposed Design

The general circuit for the designed circuit is given below: RS232, RS485, USB, CAN and Ethernet are used interfaces in this circuit. MAX 232 has been used to interface with RS232 protocol, since development board is used no need of MAX 232. MCP2551 has been used as a CAN bus driver; finally, for interfacing with LAN, DP83848 has been used. It is a physical chip.

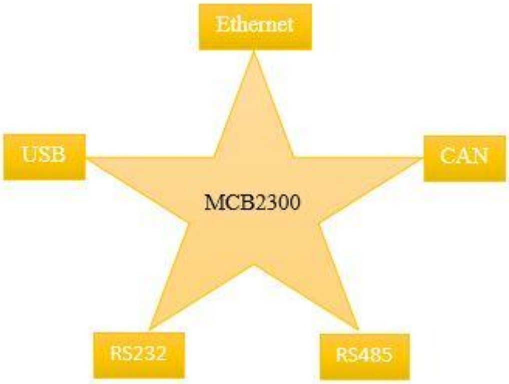


Fig. 1 General Block Diagram of Proposed Design

C. Overall model of proposed design

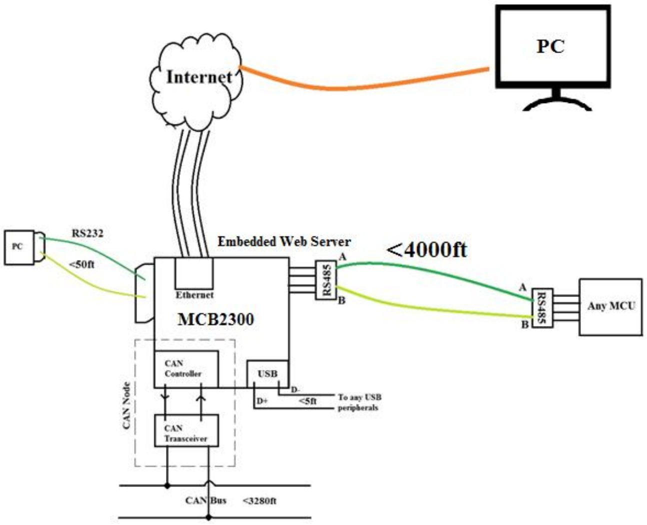


Fig. 2. Overall model of Proposed Design

Fig.2. represents the overall model of proposed system. The software, written with RL-ARM, in 'C', provides for connecting any of the serial port to another one. The configuration can be done using telnet client from any networked PC.

III. Hardware Part

Microcontroller going to be used is LPC23xx's variant and its manufacturer is Philips Company. Some of the main serial protocols going to be interfaced are RS232, RS485, USB, CAN and Ethernet.

Serial Interfaces:

The whole purpose of a serial interface is to provide a single path for data transmission wirelessly or over a cable. Parallel buses are still used in some applications. But with high-speed data so common today, a serial interface is the only practical option for communications over any distance greater than several feet. Serial interfaces can be used to provide standardized logic levels from transmitters to receivers, define the transmission medium and connectors, and specify timing and data rates. In some cases, they can perform serial-to-parallel and parallel to-serial conversion or specify a basic data protocol.

A. LPC23xx:

The NXP (founded by Philips) LPC2148 is an ARM7TDMI-S based high performance 32-bit RISC Microcontroller with Thumb extensions 512KB on-chip Flash ROM with In-System Programming (ISP) and In-Application Programming (IAP), 32KB RAM, Vectored Interrupt Controller, Two 10bit ADCs with 14 channels, USB 2.0 Full Speed Device Controller, Two UARTs, one with full modem interface. Two I2C serial interfaces, Two SPI serial interfaces Two 32-bit timers, Watchdog Timer, PWM unit, Real Time Clock with optional battery backup, Brown out detect circuit General purpose I/O pins. CPU clock up to 60 MHz, On-chip crystal oscillator and On-chip PLL.

B. RS232:

One of the oldest serial interfaces is generically called RS-232. It was originally established as a method of connecting data terminal equipment (DTE) such as electromechanical teletype writers to data communications equipment (DCE). Over the years its use has included connections to video terminals, computers, and modems. The first personal computers included an RS-232 called a serial port for connection to a printer or other peripheral device. Today, it is still widely used in embedded computer development systems, scientific instruments, and all sorts of industrial control equipment.

C. RS485:

RS485 is an electrical driver which defines not only a single device-to-device interface but also a communications bus that can be used to form simple networks of multiple devices. Its configuration and specifications also extend the range and data rate beyond the RS-232 interface capabilities. Maximum cable length is commonly defined as 1200 meters or about 4000 feet. The typical maximum data rate at 4000 feet is 100 kbits/s. This standard specifies a maximum of 32 drivers (transmitters) and 32 receivers. Line drivers are disconnected from the line when not transmitting. All receivers are fully connected, and the bus line is terminated in a load matching resistance.

D. USB:

USB, short for Universal Serial Bus, is an industry standard that defines the cables, connectors and communications protocols used in a bus for connection, communication, and power supply between computers and electronic devices. The USB is based on a so-called 'tiered star topology' in which there is a single host controller and up to 127 'slave' devices. The host controller is connected to a hub, integrated within the PC, which allows a number of attachment points (often loosely referred to as ports). A further hub may be plugged into each of these attachment points, and so on. However there are limitations on this expansion.

E. *CAN:*

A controller area network (CAN bus) is a vehicle bus standard designed to allow microcontrollers and devices to communicate with each other in applications without a host computer. It is a message-based protocol, designed originally for connecting modules within automobiles, but is also used in many other contexts. Also it is a serial communication standard that supports distributed real-time control with a very high level of security. CAN provides multi-master capabilities, and is particularly suited to networking “intelligent” devices as well as sensors and actuators within a system or sub-system.

CAN is a multi-master serial bus standard for connecting Electronic Control Units [ECUs] also known as nodes. Two or more nodes are required on the CAN network to communicate. The complexity of the node can range from a simple I/O device up to an embedded computer with a CAN interface and sophisticated software. The node may also be a gateway allowing a standard computer to communicate over a USB or Ethernet port to the devices on a CAN network. Each node requires a: CPU, a controller and a transceiver.

F. *Ethernet:*

Ethernet is the most widely installed local area network (LAN) technology. Ethernet is a *link layer* protocol in the TCP/IP stack, describing how networked devices can format data for transmission to other network devices on the same network segment, and how to put that data out on the network connection. It partly covers both Layer 1 (the physical layer) and Layer 2 (the data link layer) on the OSI network protocol model. Ethernet defines two units of transmission, packet and frame. The frame includes not just the "payload" of data being transmitted but also addressing information identifying the physical "Media Access Control" (MAC) addresses of both sender and receiver, VLAN tagging and quality of service information, and error-correction information to detect problems in transmission. Each frame is wrapped in a packet, which affixes several bytes of information used in establishing the connection and marking where the frame starts.

IV. SOFTWARE PART

Keil MDK is the complete software development environment for a wide range of ARM, Cortex-M, and Cortex-R based microcontroller devices.

A. *Real Time Operating System:*

Most operating systems appear to execute multiple tasks at the same time. This is called multi-tasking. In reality, each processor core can only be running a single thread of execution at any given point in time. A part of the operating system called the scheduler is responsible for deciding which program to run when, and provides the illusion of simultaneous execution by rapidly switching between each program.

The type of an operating system is defined by how the scheduler decides which program to run when. For example, the scheduler used in a multi user operating system (such as UNIX) will ensure each user gets a fair amount of the processing time. As another example, the scheduler in a desk top operating system (such as Windows) will try and ensure the computer remains responsive to its user. The scheduler in a Real Time Operating System (RTOS) is designed to provide a predictable (normally described as *deterministic*) execution pattern. This is particularly of interest to embedded systems as they often have real time requirements. A real time requirement is one that specifies that the system must respond to a certain event within a strictly defined time (the *deadline*). A guarantee to meet real time requirements can only be made if the behaviour of the operating systems scheduler can be predicted (and is therefore deterministic).

The RTOS used here is RTX/RL ARM Library by Keil. The Keil RTX is a royalty-free, deterministic Real-Time Operating System designed for ARM and Cortex-M devices. It allows you to create programs that simultaneously perform multiple functions and helps to create applications which are better structured and more easily maintained.

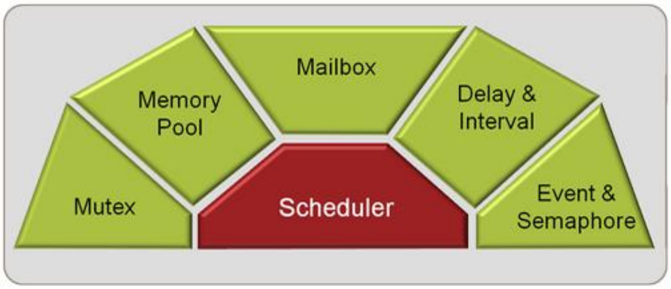


Fig. 3 Features of RL-ARM Library

IV. IMPLEMENTATION

A. *Interfacing RS232 serial driver with LPC23xx development board:*

Serial communication using RS232 protocol is the most common and widely used protocol in embedded systems. In this section how to perform serial communication and also how to communicate with PC using RS232 protocol is depicted. Connection between PC and microcontroller is shown in the fig.4

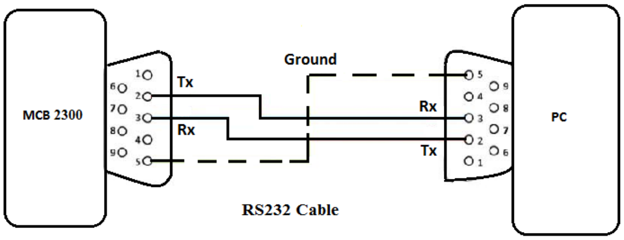


Fig. 4 Features of RL-ARM Library

Actually Microcontroller works on TTL or CMOS voltage logic levels. But PC operates on RS232 voltage logic level which is different from CMOS or TTL. So we need a voltage converter which can convert one logic level to another. MAX232 IC with four capacitors does this task. Since MCB2300 (LPC23xx variant) provides direct communication via its own COM port, from which it is possible to connect to the PC directly by RS232 cable (DB9 connector) alone.

MCB2300 has an inbuilt UART for carrying out serial communication. The serial communication is done in the asynchronous mode. A serial port, like other PC ports, is a physical interface to establish data transfer between computer and an external hardware or device. This transfer, through serial port, takes place bit by bit. An important parameter considered while interfacing serial port is the Baud rate which is the speed at which data is transmitted serially. It is defined as number of bits transmitted or received per second. It is generally expressed in bps (bits per second). Hyper Terminal, a Windows XP application, or any other virtual terminal can be used to receive or transmit serial data through RS232.

B. *Interfacing RS485 serial driver with LPC23xx development board:*

The RS485 is one of the serial communication protocol that is used to send and receive data bit-by-bit sequentially through a serial cable. The RS485 protocol allows multiple slave devices to communicate at a time with a master device (microcontroller or any other controller).

Since LPC23xx development board doesn’t have level translator in-built, MAX485 IC or RS485 module is needed. RS485 module is used because of its low cost and ease of implementation. Sometimes termination is required to reduce the loss of data.

Pins of RS485 and its configuration for connecting it to micro-controller is given in Table 1 and fig.5

Table.1

Pin Configuration for connecting RS485 module to micro-controller

RS485 pins	LPC 23xx pins	Description
DI	P0.8	TxD
DE	P0.10	Enables TxD mode
RE	P0.11	Enables RxD mode
RO	P0.9	RxD
A&B	To slaves	RS485 line pair
VCC	JTAG 2nd Pin	Power Supply
GND	JTAG 4th Pin	Ground

Programming RS485 serial communication is similar to RS232. But we need to consider the GPIO pins connected to Driver Enable (DE) and Receiver Enable (RE) pins according to the mode of communication i.e., transmission or reception. Since communication occurs between to microcontrollers, one microcontroller acts as master and the other acts as slave. So considering that programming is performed.

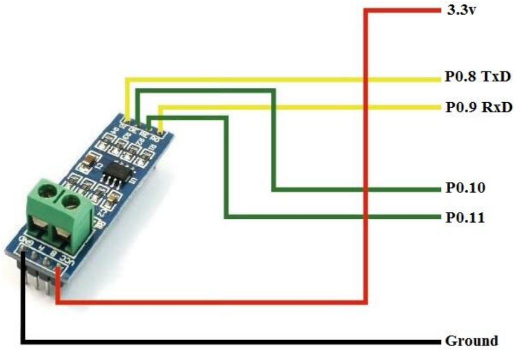


Fig.5 Connecting RS485 with Microcontroller

C. *Interfacing USB with lpc23xx development board:*

The USB is a 4 wire bus that supports communication between a host and a number (127 max.) of peripherals. In USB data cable Data+ and Data- signals are transmitted on a twisted pair. No termination needed.

There are several types of USB Connectors Type A, Type B, Mini A, Mini B, Micro A, Micro B, Standard A, Standard B and USB 3.0. MCB 2300 has Type B type of USB in-built.

USB pin out diagram and its configuration is given in fig.7 and table 2 respectively.

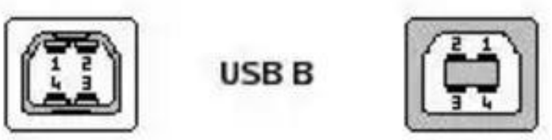


Fig.7 USB Pin-Out Diagram

Table.2

Pin Configuration for connecting USB to micro-controller

Pin	Name	LPC23xx pins
1	Vcc	P0.23 (Voltage)
2	Data-	D-
3	Data+	D+
4	GND	Ground

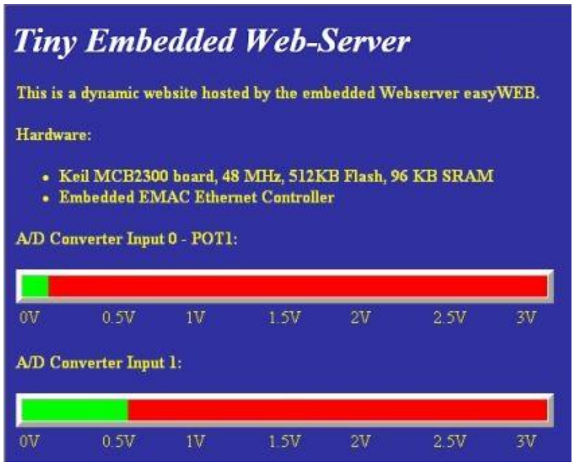
USB Kernel, USB Descriptors and USB Hardware Layer Module and also some other libraries required for our application is listed here:

- usbcfg.h
- usbcore.c
- ucbhw.c
- usbdesc.c
- mscuser.c

D. *Interfacing Ethernet with lpc23xx development board:*

Ethernet is the most widely-installed local area network (LAN) technology. An Ethernet LAN typically uses special grades of twisted pair wires. The most commonly installed Ethernet systems are called 10BASE-T and provide transmission speeds up to 10 Mbps. Devices are connected to the cable and compete for access using TCP/IP protocol.

Several applications can be built with the help of TCP/IP stack and Ethernet packet driver. Here tiny embedded web server is built with the help of simple html webpage and TCP/IP. After coding is done it must be loaded into the target and also it must be configured over the network. So that it can acts as embedded web server. For testing this embedded web server the IP address must be entered in any browser. So that the resulting web page shown in fig.8 will be displayed in the web browser.



Tiny Embedded Web-Server

This is a dynamic website hosted by the embedded Webserver easyWEB.

Hardware:

- Keil MCB2300 board, 48 MHz, 512KB Flash, 96 KB SRAM
- Embedded EMAC Ethernet Controller

A/D Converter Input 0 - POT1:

A/D Converter Input 1:

Fig.8 Sample of Tiny Embedded Web Server

V. CONCLUSION

In this paper, we present the design by interfacing various serial protocols such as RS232, RS485, CAN, USB and Ethernet. This circuit can be used for data acquisition and control system. Since this system uses various different protocols, the versatile ARM microcontroller has been used. So using RL-ARM RTOS, tasks are managed so easily and also power wastage can be minimized.

VI. FUTURE WORK

Future development of this project includes, incorporating MODBUS serial interface protocol with LPC23xx. Also for better management, working with different protocols and also, prohibition from power wasting a real time operating system, RL-ARM Library is going to be ported on LPC2148. Result of this proposed circuit without RTOS doesn't make the circuit so smart, since the circuit works on different protocols there won't be any management and also there will be power wasting. Still the development can be further extended by interfacing wireless serial drivers such as Bluetooth, Wi-Fi, etc. Therefore, High precision data acquisition can be realized by this proposed system.

ACKNOWLEDGMENT

I would like to express my sincere gratitude to my professor **V. Natarajan** for the continuous support of my project, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my M. Tech study.

REFERENCES

[1] Pau Marti, Manel velasco, Joseph M. Fuertes, Antonio Camacho, Giogio Buttzzo "Design of an Embedded control systems Laboratory Experiment" IEEE Transactions on industrial Electronics, Vol. 57, No. 10, October 2010.

[2] Gan Ping LEE, Ming LEE. "Networked Intelligent Controller Based on Embedded System", The 30th Annual Conference of the IEEE Industrial Electronics Society, busaan, korea, November2-6, 2004, pp.2942-2946

[3] Wang Junbo, Zou Jijun. "The Design of USB Bus and CAN Bus Protocol Adaptor". *JOURNAL OF EAST CHINA INSTITUTE OF TECHNOLOGY*, Wuhan, China. Vol.129 No13, Sep. 2006, pp.273-276.

[4] T.D.Liu W.Tang and H.Jiang, "The Design of Industrial Ethernet Adapter Based On Ethernet/IP," 4th International Conference on Computer Science & Education, pp.1239– 1241, 2009.

[5] Yonghong HU, Lu DING and Tong YAO, "Designing and Implementing of Serial Ports to Ethernet Gateway on Embedded System," Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference Volume 1, Issue , 20-22 Feb. 2006 pp.146-148

[6] Ligong Zhou. ARM embedded system experiment course [M].Beijing, Beijing University of Aeronautics and Astronautics publishing house, 2004.

[7] Theory on RS485, RS232 specs: http://rs485.com/rs485spec.html

[8] MCB2140's User Guide: http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.kui0055a/mcb2140_wp_devstudio.htm

[9] **"Implementation of Modbus Communication Protocol Based on ARM Coretx-M0"** published on 2014 IEEE International Conference on System Science and Engineering (JCSSE) by Kelong Wang, Daogang Peng, Lei Song, Hao Zhang.

[10] LPC2141/42/44/46/48 product Datasheet.

[11] RL ARM Guide: ARM Information Center… http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.kui0062a/rlarm_default.htm.