Design and Development of ARM based Embedded System Laboratory

K.Navaneetha[1] R.Anil Kumar[2]
[1]M.Tech. Student [2]Assistant Professor
[1,2]Department of Electronics and Communication Engineering
[1,2]Jyothishmathi College of Engineering and Technology, India

Abstract— 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. Popular serial interfacing protocols include: USB, I2C, SPIISSP, CAN and UART for communication between integrated circuits for low/medium data transfer speed with on board peripherals. This paper presents a platform which deals with the implementation of certain of the above serial protocols presented by a low power 32-bit ARM RISC processor: LPC2148, with suitable examples, including hardware and software details. This platform is also useful for students of different disciplines to work with different serial protocols, which helps them in interfacing of sensors, memory ICs, analog subsystems and so on. It also aims to provide the students with hands on experience, practices in embedded systems and minimizing the prerequisite knowledge.

Key words: Embedded System; ARM7TDMl, 12C, SPl, RS232

I. INTRODUCTION

Embedded system is the combination of software and hardware design that can be combined with different areas such as electronics, computer science, and automatic control systems and so on. It has a strong extensive exercises, implementation, execution and application. The economic importance of embedded systems is growing exponentially, as electronic components are present in many different applications: right from small microcontroller devices to system on chip (SOC), network on chip (NOC) devices, embedded PCs, Robotics, signal processing application s etc [1]. It is required to develop embedded systems having good support with peripherals such as communication networking, sensor integration, GPS system, USB devices, DSP, ETHERNET etc.

Embedded system mainly uses serial communication to communicate with other peripherals. Certain examples are: medical devices, telecom equipment and other gadgets used in vehicular and home appliances. Therefore serial communication plays an important role in embedded system design. Further, if the microcontroller used in the designed embedded system has to communicate with externally connected peripherals such as memory sticks, sensors, application specific ICs (ASICs), knowledge on the implementation in the direction of hardware and software are required.

Further, most of the embedded systems have to be interfaced with the external devices to transmit data to PC/LAPTOP or workstation or to interact with another system for sharing data. For this reason, embedded systems are provided with communication interfaces for monitoring and control by a host system or a node on a network. Hence through these interfaces, an embedded system can also be accessed over the internet. Today we find two worldwide standards of communication protocols such as SPI, I2C. The two protocols coexist in modern digital electronic systems and also they probably will continue to compete in the future, as both I2C and SPI are actually quite complementary for some kind of communication [2, 3]. Jai Karan Singh [4] described the implementation of I2C and SPI on FPGA using VHDL, but it does not signify the importance of controller in embedded system design. Design flexibility of embedded system by interfacing SPI to I2C peripherals with low power techniques was developed by Kiran and Vinilanagaraj [5]. FPGA based implementation of I2C and SPI protocols and a comparative study was given by A.K. Oudjida et al [6] which is beneficial for the students to get the knowledge on SPI and I2C protocol implementation. None of these papers have stressed on practical implementation on these aspects using a microcontroller.

This paper describes a platform of communication protocols implementation using ARM7TDMI (LPC2148). The hardware and software described in this paper, helps the students to practice with serial standards. The developed system allows data exchange between any two systems. It is also useful for the students to gain knowledge on communication protocol implementation using learning based approach because a large number of peripheral devices and ICs use these standards. The platform can thus satisfy the need of beginners to experienced students, and also help them to to learn how to work with a complex development kit. Further the experiments described here can be implemented as a practical exercise in an educational environment. This approach differs from the previous embedded system design activity [7, 8].

The organization of the paper is as follows: Section-II briefly overviews the ARM architecture. Section-III gives detailed description of the platform with the hardware interface and software used to implement the SPI, 12C, UART standards. Conclusions are included in Section-IV.

II. ARCHITECTURE OF ARM7TDMI (LPC1768)

The LPC2148 board consists of ARM7TDMI as its core and it is designed by NSK electronics Bangalore [9]. This module is benefit for the students in such way that all the features of the microcontroller will be easily accessed by them. ARM7TDMI family has good performance in situations Bwhere the energy consumption is critical design goal. LPC2148 has ARM7TDMI as its core is called CPU core, the modules inside are connected by the CPU high performance bus called Advance High performance bus (ARB) and the peripherals are connected by VLSI peripheral bus (VPB). Block diagram of LPC2148 is shown in Fig. 1 (a), Fig. l(b) shows the LPC2148 microcontroller board.

A. *Memory:*

LPC 2148 has 2 types of memories:

- Flash memory of 512KB
- Static RAM of 40KB

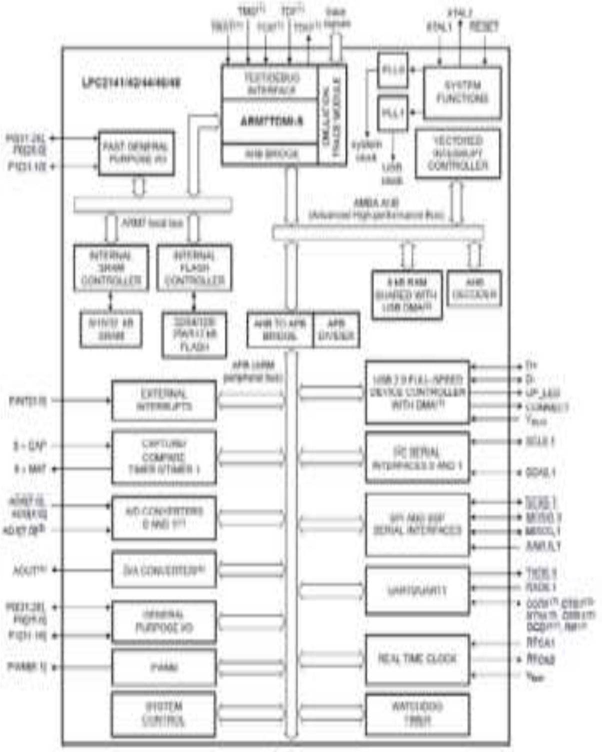

Fig. 1(a)

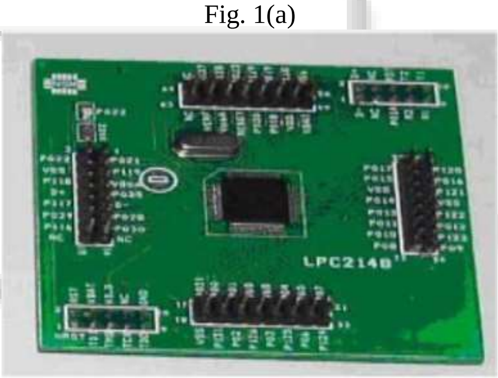

Fig. 1(b)

Fig. 1(a): Block Diagram of LPC2148 Architecture (b): LPC2148 Microcontroller Board

ARM7TDMI CPU works in the frequency range 12 MHz to 60 MHz and with an operating voltage of 3.3V. ARM7 CPU has sixteen 32-bit General purpose registers, 2 Special purpose registers - Current program status register (CPSR) and Saved program status register (SPSR). It also provides a wide range of peripherals like Two lO-bit ADCs with 14 channels, USB 2.0 full modem interface, Two 12c, Two SPI serial interfaces, Two 32 bit Timers, WDT, PWM unit, RTC with optional battery backup and 45 general purpose I/O pins. ARM offers the Keil microcontroller development kit (MDK-ARM) for ARM powered /lC. It features the industry standard compiler from ARM, the Kiel /lvision IDE and sophisticated debug and data trace capabilities. MDK-ARM offers good support for all Cortex, ARM7, and ARM9 processor based devices and is recommended for students to work on the standard ARM based MCU design [10].

III. DESCRIPTION OF IMPLEMENTING SERIAL STANDARDS

This section would help the students in the process of learning activity of implementing serial protocols using ARM7TDMI. The designed hardware and implemented software are explained below. This is from the view point that a large number of peripheral devices and ICs use these standards.

The need of data transfer between PC/Laptop/Desktop and microcontroller, vice versa and between two microcontrollers is a never ending process. This can be accomplished in two ways I. Parallel data transfer ii. Serial data transfer. Parallel data transfer needs "n" lines where n is the bit size of the microcontroller. This will increase the size of the hardware. An alternative to parallel is serial communication. Serial communication has challenges like synchronization, control etc. thus there are two ways of serial communication: i. Synchronous ii. Asynchronous. SPI and 12C come under the category of synchronous whereas communication using UART will be asynchronous. Fig. 2 shows the simplified structure of data transfer format with examples.

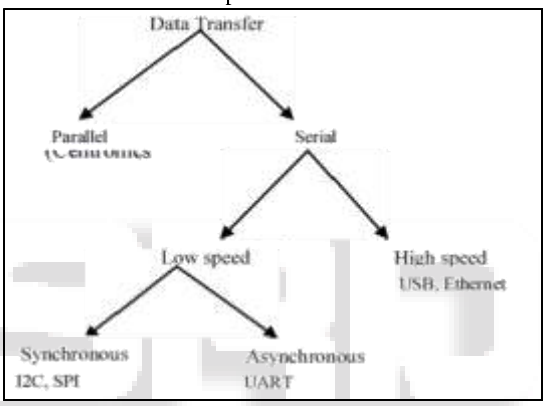

Fig. 2: simplified structure of data transfer format with examples

Serial peripheral interface (SPI) is the interface present in most of the embedded systems, supports full duplex communication with higher throughput, whereas Inter integrated circuit 12C is a two wired bus. In the case of SPI,more number of devices can be connected to the serial bus. Only the device whose chipselect signal is active will respond for communication. Conversely, for an 12C communication, address is essential to communicate with the slave device. Before going to the actual implementation process, a theoretical comparison that exists between these two protocols is given in Table I.

A. *Inter Integrated Circuit*

I2C stands for Inter Integrated circuit, it is a multi master serial bus developed by Philips. I2c is an ideal bus used for interfacing with low speed peripheral devices to embedded systems [11]. Some applications of the I2C interface include EEPROM, SDRAM memory module interfacing, DACs, low speed ADCS, fan control chips etc. I2C is based on two bidirectional lines, serial clock line (SCL) and serial data line (SDA) which are pulled high with pull up resisters (3.3kQ). These lines together are commonly known as I2C bus which is used for communicating among all I2C devices [12]. Key features of I2C data transfer are:

Data transfer up to 100 kbps in standard mode, and

- up to 400 kbps in fast mode

- Fixed data width of 8 bits
- Frame data transfer (unlimited number of bytes)
- Bidirectional data transfer, software programmable
- acknowledge bit
- Repeated data transfer and START detection
- Interrupt or polling mode transfer

In the present work LPC2148 is master device, A T24C04 [13] is slave which has address lines AO, AI, A2 are hardwired to different values. The master device initiates the I2C communication by sending START bit using register described in the data sheet [10]. I2C data transfer is shown in Fig. 3. The slave device sends ACK bit after each received bit. Data is transmitted between a Start and Stop condition as shown in Fig. 4. Fig. 5 shows the circuit diagram of interfacing LPC 148 with A T24C04.

	I2C	SPI
Originator	Philips (1982)	Motorola (1979)
Transfer type	Half Duplex	Full duplex
Transfer rate	Limited (100& 400KHz and 3.4MHz)	Free(nxMHz to 10nxMHZ)
Addressing	Software (7/10 bits)	Hardware chip select (slave select input pin nSS)
Protocol complexity	low	lower
Application	Multi master register access	Transfer of data s reams
Multi Transfer	yes	No
Multi slave	Yes	Yes
I/O constraints	Open drain with pull up resisters	No constraints
Synchronization	Serial clock with data acknowledge	Serial clock with programmable polarity and phase

Table 1: Comparison of I^2 and SPI Protocols

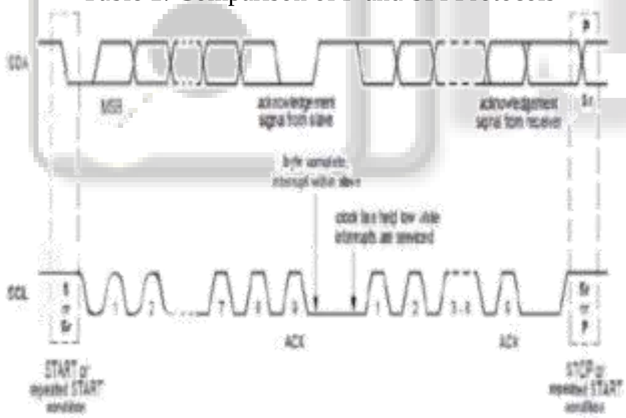

Fig. 3: I2C Data Transfer

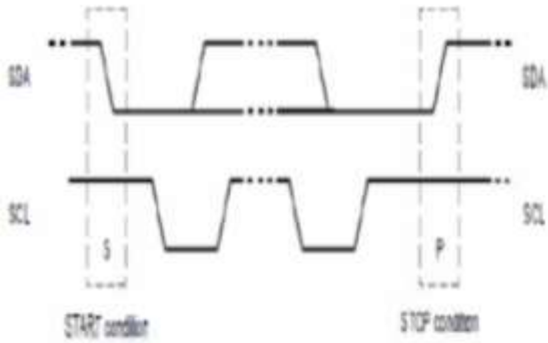

Fig. 4: I2C Start and Stop Condition

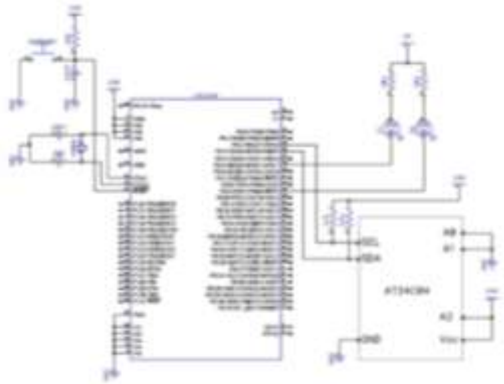

Fig. 5: Circuit Diagram for Interfacing LPC2148 with AT24C04

Two LEDs are used for visual indication of the communication status. The software is developed in embedded C and compiled using Kiell Vision IDE. The created hex file is dumped on to the flash memory using Philips Flash Utility. The flow chart is shown in Fig. 6 is for writing and verifying data bytes through I2C. The overall system was implemented and tested successfully. Fig.7 shows the implementation of I2C interfacing with A T24C04. Fig. 8 is the I2C simulation window.

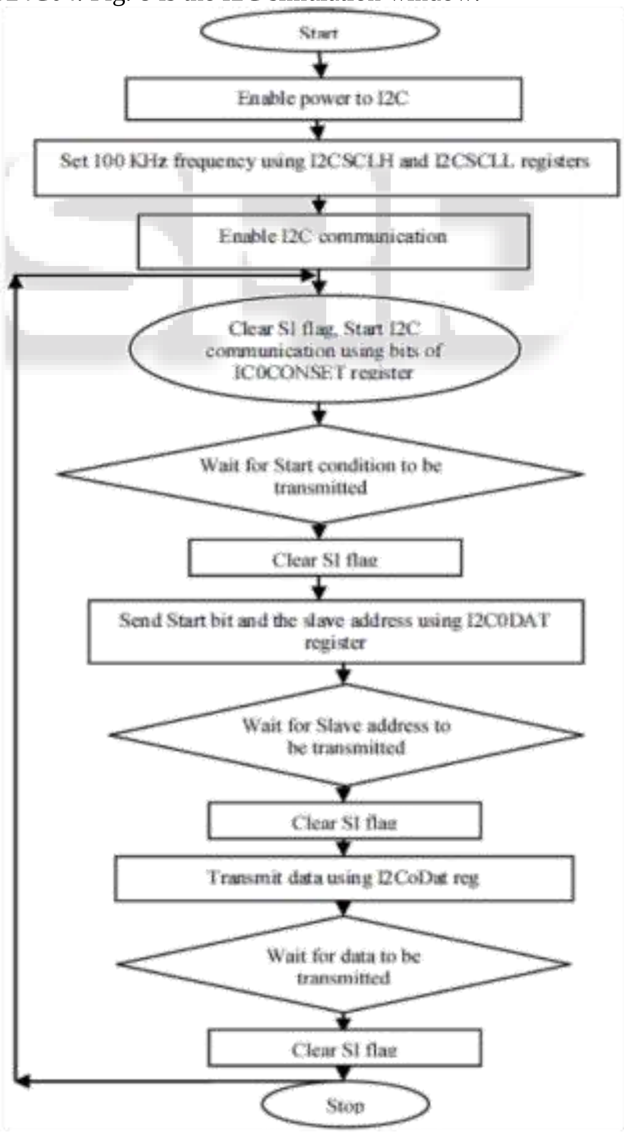

Fig. 6: Flow Chart of Writing Data Byte using I2C Communication

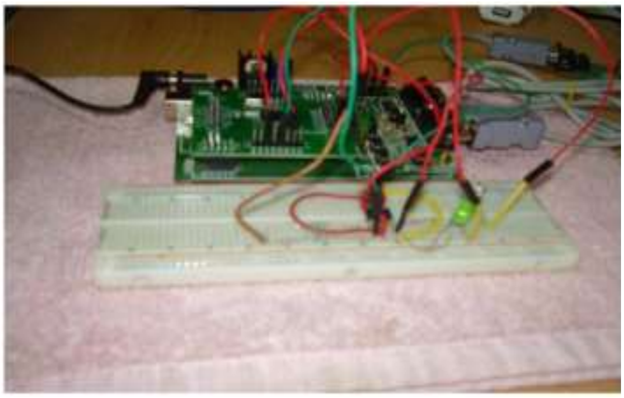

Fig. 7: Photograph of Interfacing I2C EEPROM (AT4C04)

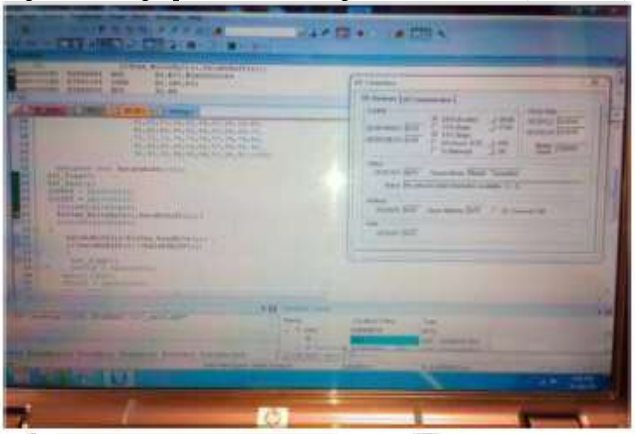

Fig. 8: Photograph of I2C Simulation Window using Keil Software

B. Serial Peripheral Interface (SPJ)

This is a four wire communication. In SPI protocol, serial communication is accomplished by means of two pins that are SI (Serial data In) and SO (Serial data out). The SCLK provides clock synchronization and CS is the chip select pin. This communication can be implemented between processors and peripheral that have SPI interface. Serial peripheral interface bus is synchronous serial data link standard at low speed. This is a full duplex mode of communication. The communication is performed in master/slave mode; master device initiates the data frame. The serial clock (SCLK) is generated by the master device, slave select SS should be in active low, for the device to communicate with the master. The SDO or serial data output signal is sent by the master on MOSI line. After receiving a clock pulse, the slave device responds back with SDI or serial data input signal (MISO). Key features of SPI data transfer is [14]:

– Low/medium data transfer rate depending on implementation issues Frame data transfer (unlimited number of words).
– Bidirectional data transfer
– Interrupt and polling mode transfer.
– Serial clock with programmable polarity and phase

In certain applications, internal ADC is not enough when we need more resolution or high speed conversions. The internal ADC of LPC2I48 is of IO-bit resolution. If we need more resolution than that of on-chip ADC, we need to interface an external ADC. The following exercise will help the students to develop hardware and software to interface an external ADC with ARM MCV, in data acquisition under SPI protocol. In the present work ARM is the Master and MCP3204 is a slave on the bus. SPI transfer is initiated by the MASTER by pulling the CS line low. SOSPCR control register is used to control SPI operation, in which SPI is enabled in 8-bit master mode and CPOL, CPHA are made to O. Data is sent using SOSPDR data register which is a bi directional register. Software is developed based on flow chart as shown in fig 9, Circuit diagram for SPI implementation is shown in fig 10. The program developed in the present study reads the I2-bit ADC of MCP3204 and displays it on the LCD module as shown in Fig. 11.

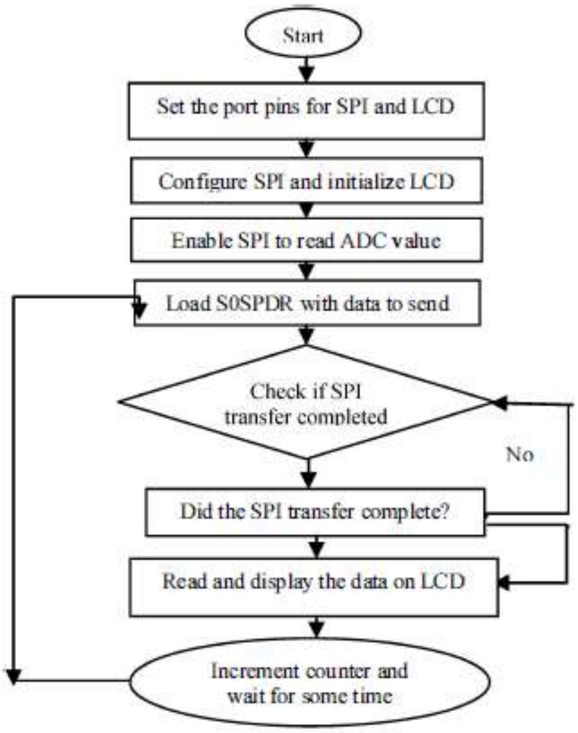

Fig. 9: Flow Chart for Reading Data through SPI Communication

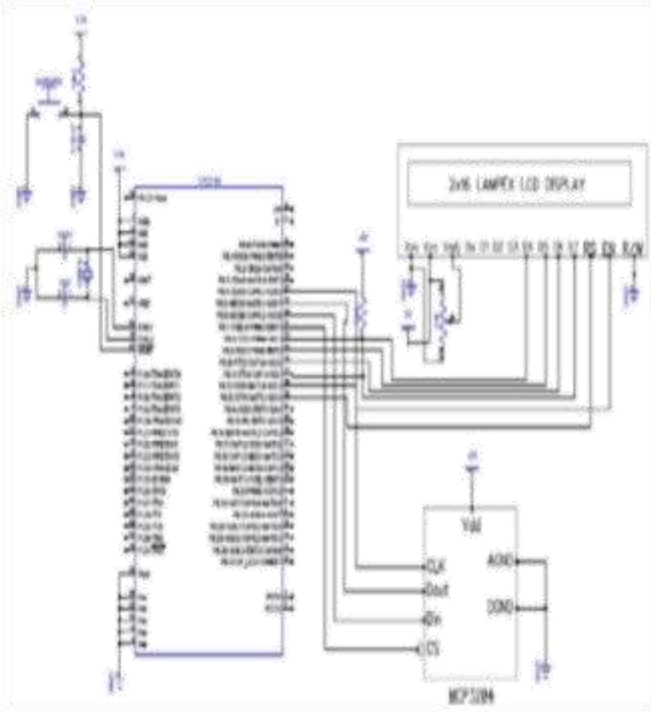

Fig. 10: Circuit Diagram of SPI Implementation

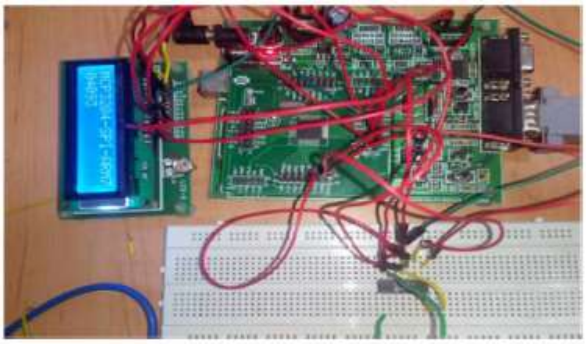

Fig. 11: Interfacing 12 Bit SPI ADC (MCP3204) with ARM LPC2148 Microcontroller

C. Universal Asynchronous Receiver/Transmitter (UART)

A Universal Asynchronous Receiver/Transmitter is usually an individual integrated circuit used for serial communications over a computer or peripheral devices' serial port. UARTs are mostly used in conjunction with communication standards such as RS232, RS-422 or RS-48S. In majority of present day microcontrollers, UARTs are integrated on the chip itself, for communication between an embedded system and an external device. Many modern ICs come with a UART that can also communicate synchronously; those devices are called as USARTs. Transmitting and receiving of data between UARTs or between UART and PC/laptop must be set for the same bit speed (baud rate), character length, parity and stop bits for proper operation, using the internal control register of the on-chip UART. The LPC2Ixx devices have two on chip UARTs. Both are identical to use except that U AR T l has an additional modem support. Both the peripherals have a built in baud rate generator and 16 byte transmit and receive FIFOs. In the present work, first pin select block is programmed to switch the processor pins from GPIO to the UART functions

(PINSEL =OxOOOOOOOS), next the UART Line Control Register is used to configure the format of the Transmitter Data Character. In our program the Character format is set to 8 bit, no parity and one stop bit,8-N-I (UOLCR = Ox00000083).

In the line control register, there is a bit called DLAB (Device Latch Access Bit), which is set to zero to protect the contents of the divisor register. LPC2148 has a baud rate generator of 16 bit preescaler which divides the Pclk to generate the UART clock which must run at 16 times the baud rate. . Baud rate is calculated using the formula: Baud rate = VPB clocklI6* divisor value This value needs to be entered in to the UODLL register (UODLL =Ox0000006I). The following algorithm configures UARTO to interface to a terminal program running on a host machine (PC's hyper terminal) at a baud rate of 9600, circuit diagram is shown in fig 12. The program code simply prints "LPC2148 working at 9600 baud rate" on the host terminal as shown in fig. 13, continuously, since the program code is included in a while loop. The software is developed based onthe following algorithm.

1) Initialize the board.
2) Power up UART l using PCONP register.
3) Configure peripheral clock for UART l using PCLK SEL l register.
4) Select PO.IS and PO. l6 as TXD and RXD pins
5) S. Enable UART l Transmit and Receive
6) Select 8 bit data length, no parity,I stop bit using UOLCR register
7) "LPC2148 working at 9600 baud rate" is sent to desktop/laptop's hyper terminal, continuously.

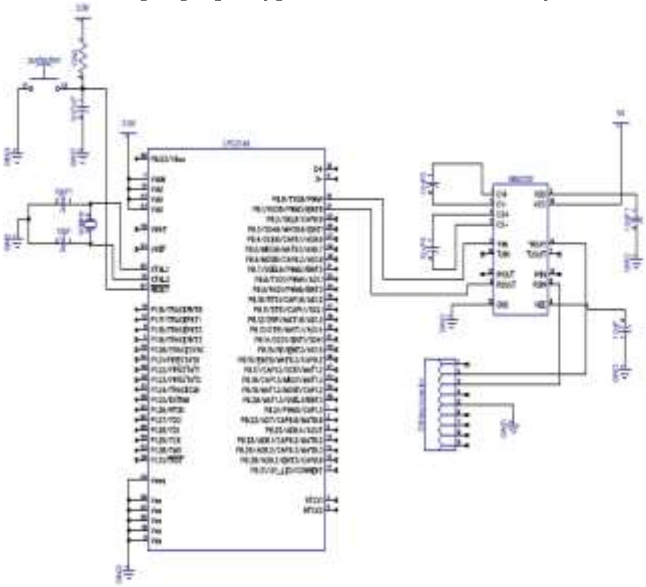

Fig. 12: Circuit diagram for USART implementation.

Fig. 13: Hyper terminal shows that "lpc2148 working at 9600 baud rate

IV. CONCLUSION

A platform based experimental method has been discussed which will be helpful to enhance students understsnding. This platform is also useful for both theoretical and experimental teaching; hence the embedded system teaching can be improved. From this activity students should be made aware of fundamental concepts of standard communication protocols using modern 32-bit ARM RISC processor architecture. Students are also encouraged to make themselves familiar with new development architectures for example, different microprocessors, cross compilers, tool chains, advanced communication protocols like USB, Blue tooth, Ethernet, CAN etc. The main focus is on the ARM based evolution boards that can be incorporated in the engineering design projects. The low cost and fully open structure of the ARM is considered to make a good embedded platform. With this platform engineering students will learn how challenging it must be for the designers to develop hardware design, hardware prototyping and sensor connectivity.

ACKNOWLEDGMENT

K. Aruna is thankful to the University Grants Commission (UGC), New Delhi, for sanctioning Junior Research Fellowship (UGC: No.F4-1I2006(BSR)/l1- 23/2008(BSR)).

REFERENCES

[1] Pau Marti, Manel velasco, Joseph M. Fuertes, Antonio Camacho, Giogio Buttazzo "Design of an Embedded control systems Laboratory Experiment" IEEE Transactions on industrial Electronics, Vol. 57, No. 10, October 2010.
[2] F. Leens, "An Introduction to 12C and SPI protocols ", IEEE instrumentation & measurement Magazine, pp. 8-13 feb. 2009.
[3] D. Davcev, B.Stojkoska, s. Kalajdziski, and k. Trivodaliev, "Project based learning of embedded system", in proceedings of the 2 nd WSEAS international Conference on circuits, systems, signal and Telecommunications, 2008.
[4] Jai Karan Singh et al " Design Implementation of 12C to SPI converter on FPGA using VHDL", "VSRD international Lournal of Electrical, Electronics and communication Engineering", voI.2(7), 201, 499-507.
[5] V.Kiran, Vinila Nagraj, "Design of SPI to 12C Protocol converter and implementation of Lo-power Techniques", "Internationl Journal of Advanced research in Computer and Communication Engineering", VoI.2,No.l 0,PP3770-3774, Oct ober,2013.
[6] A.K. Oudjida et al "FPGA Implementation of I2C & SPI protocols: a comparative study", 978-1-4244-5091, IEEE, 2009, pp. 507-510.
[7] P. Myers, "Interfacing Using Serial Protocoles: Using SPI and 12C".A vailable:http://intranet.daiict.ac. in/-ranjan/esp2005/paper/i2c _spi _341.pdf.
[8] L. Bacciarelli et ai, "Design, Testing and Prototyping of a Software Programmable 12C/SPI IP on AMBA Bus," Conference on Ph.D. Research in MicroElectronics and Electronics (PRIME'2006), pp. 373- 376, ISBN: 1-4244-0157-7, Ortanto, Italy, June 2006
[9] http://www.nskelectronics.com
[10] LPC2I48 user manual NXP semiconductors- UMIOI39
[11] I2C -bus specification and user manual UMI0204.
[12] [I2C- bus - specification-I995
[13] Two-wire Serial EEPROM AT24C02, AT24C04. Available on www. Atmel.com/lmages/docO 180.
[14] Motorola Inc., "SPI Block GuideV 03.06," February 2003.
[15] MCP3204 data sheet available on wW1.microchip.com

