American Journal of Embedded Systems and Applications
2013; 1(2): 46-53
Published online December 20, 2013 (http://www.sciencepublishinggroup.com/j/ajesa)
doi: 10.11648/j.ajesa.20130102.13

Design and development of a project-based embedded system laboratory using LPC1768

Aruna Kommu, Raghavendra Rao Kanchi*

VLSI and Embedded System Laboratory, Department of Physics, Sri Krishnadevaraya University, Ananatapuram, 515003, A.P., India

Email address:
kanchiraghavendrarao@gmail.com (R. R. Kanchi)

To cite this article:
Aruna Kommu, Raghavendra Rao Kanchi. Design and Development of a Project-Based Embedded System Laboratory Using LPC1768. *American Journal of Embedded Systems and Applications.* Vol. 1, No. 2, 2013, pp. 46-53. doi: 10.11648/j.ajesa.20130102.13

Abstract: In this paper, we propose project-based experiments useful in setting up an embedded system design laboratory. It is an outcome of the author's experience in teaching computer architecture and embedded systems in theory. The experiments and projects described herein are useful for the students to learn the building blocks of embedded system and can be implemented as one third semester laboratory course. Further, they are built around ARM based RISC processor-LPC1768 architecture, which supports modular programming. The components including the microcontroller with programmer are inexpensive. It gives a hands-on experience to the undergraduate student of Electronics and Communication Engineering (ECE) and Computer Science Engineering (CSE), or Post-Graduate students with electronics major. The salient feature of this module is that each experiment is explained by its hardware description, software algorithm which includes dumping the hex file of the program on to the microcontroller's flash memory. Our experience with the conduct of a 120 min-end of semester practical examination show that, with the chronological increase in hardware design, the students exhibited the confidence in designing new stand-alone systems with fairly complicated hardware and software. Setting up a training laboratory of this type is easy with the material described in this paper.

Keywords: Embedded System Laboratory, Low-Power Microcontroller, LPC1768microcoontroller, Stand-alone System, Hands-on Experiments

1. Introduction

The impact and presence of embedded systems is felt directly in our daily walk of life. Starting with cellular phones, digital cameras, home appliances, space applications, up to the ubiquitous networking and sensor networking, embedded systems are used [1]. Applications of embedded systems are increasing exponentially and is pervading in to the various branches of science and technology. Further, the development of low-power mixed signal controllers from various manufacturers particularly from NXP Semiconductors has paved path for the percolation of embedded systems in to mobile, industrial, automobile, space, agricultural and robotic environments.

In global scenario, embedded system courses are introduced in to the existing science and engineering curricula [2, 3]. Presently, in majority of Indian Universities, embedded system is included in the curriculum at the under-graduate level of Science and Engineering with electronics as major. In any of the engineering or science curriculum, a balance between theory course and practical training is always essential. In such a scenario, a relevant practical training is essential, particularly, while teaching a course on embedded systems in the disciplines: Computer Science, Electrical Engineering or an Applied Science Electronics course. Several papers have appeared in literature, stressing on such a need [4-6]. Further, the area of embedded systems and their interconnection is of paramount importance nowadays. This is because such systems are widely used in several fields, and the demand for trained personnel in this field is increasing fast. However, the teaching methods and laboratory training in embedded systems is lagging behind [7]. Existing problems in graduate teaching is well explained by Jiang Xiaoluo and Li Han [8]. Experiments on embedded system training are continuously monitored and improved basing on the changes taking place in the architecture of microcontrollers. Further, the industry and R&D department needs the new technological design but won't rely on old ones; since the newer controllers draw less power, occupy less space and provide advanced on-chip

features and better code optimization. Training the student at the entry level of embedded system course gives a double benefit besides the theoretical knowledge. It gives a scope to modern teaching practices, international project collaboration, and cooperative learning processes [9-11]. Also microcontroller based embedded system design with different controllers are available in the literature [12].

Keeping these facts in view point, we have developed more than a dozen of interfacing experiments using NXP Semiconductor low-power mixed signal controller LPC1768 (ARM Cortex M3). These experiments will cover one-third of semester. Of course, microcontrollers with advanced architecture are also available. Our aim is to start humbly, with a simple microcontroller and then move to complex architectures.

The experiments presented in this paper start with LED interfacing and goes up to the projects with interfacing of sensors and can be extended to sensor networking. The systematic approach of learn-while-doing, not only increases confidence in the student, but lights up the spark of innovation by thinking new ways of using the microcontroller. Further, because of the use of low-cost components for interfacing, student can afford to purchase them and hence it will be like a take-home experiments [13]. The set of experiments, described here, can also be useful for an electronics hobbyist having basic knowledge on electronics and to a fresh research student who wants to work in the direction of microcontroller instrumentation. In general, the embedded system design should qualify the following:

- The microcontroller/CPU should posses' high processing power and present sufficient on-chip memory.
- The total embedded system hardware should consume less power, facilitating battery powered operation.
- The total hardware should occupy less space and function properly in real-time, fault –tolerant and reliable modes

Further, all the above features are met with LPC1768 mixed signal controller. Performing the exercises described here, while learning the theory of LPC1768, the student acquires a partial skill in the embedded system design.

This paper is organized as: Section-II gives brief description of the architectural details of ARM Cortex M3 (LPC1768) mixed signal controller and programming procedure of the controller using KEIL µvision4. Section III gives the details of hardware and software, developed in the present study. Conclusions and scope for future work are included in Section IV.

2. Architecture and Programming Procedure using Keil µVision 4

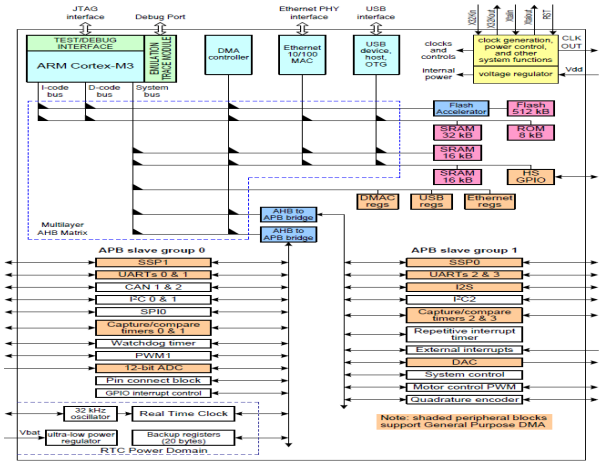
2.1. Architecture of LPC1768

The LPC1768 is a 32-bit mixed signal processor from NXP semiconductor, is most widely used in a number of embedded systems such as mobiles, automobiles, industrial control etc. LPC1768 Xplorer board was purchased from NGX Technologies, Bangalore, which cost about 35$ [14]. It consists of ARM Cortex M3 as its core with 512kB flash memory and 64kB data memory, which offers high level of integration and low power consumption. Other important features include:

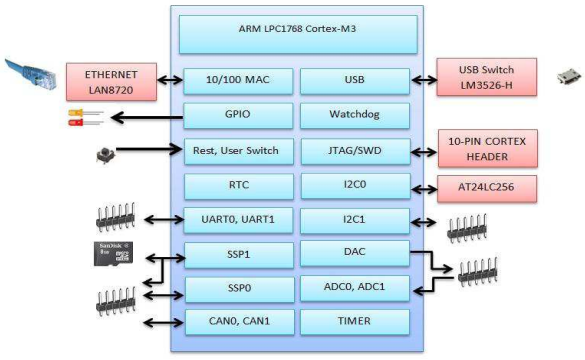
- Operates at a frequency up to 100MHz
- It incorporates a 3 stage Pipeline architecture of 0.91MIPS/MHz (fetching, decoding and running)
- Harvard architecture with separate instructions, local data buses and a third bus for Peripheral communication.

The on-chip peripheral components of the LPC1768 include: Ethernet MAC, a USB interface that can be configured as either Host, Device, or OTG, 8 channel general purpose DMA controller, 4 UARTs, 2 CAN channels, 2 SSP controllers, SPI interface, 3 I2C interfaces, 2-input plus 2-output I2S interface, 8 channel 12-bit ADC, 10-bit DAC, motor control PWM, Quadrature Encoder interface, 4 general purpose timers, 6-output general purpose PWM, ultra-low power RTC with separate battery supply, and up to 70 general purpose I/O pins [15].

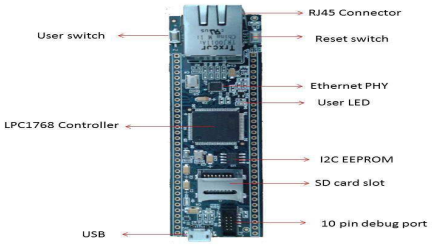
It supports the operating systems such as WINDOWS CE, LINUX, Palm OS and so on. The block diagram of ARM cortex M3 is shown in fig. 1(a), fig. 1(b) shows the internal architecture and fig. 1(c) shows the LPC1768 Xplorer board.



(a)



(b)



(c)

Fig 1. *(a). Block diagram of ARM CortexM3. (b). Internal architecture of ARM CortexM3. (c). LPC1768 Xplorer board*

2.2. Programming procedure using Keil µVision 4

ARM offers the Keil Microcontroller Development kit (MDK-ARM) for ARM powered microcontrollers. It features the industry standard compiler; the Kiel µvision IDE, sophisticated debugger and data trace capabilities. Keil µVision4 is open source software which provides best development tool and technical support [16]. Features such as free development tools and compilers plus the ability to integrate a low cost In Circuit Debugger (ICD) in to the ARM CPU module made the ARM an excellent choice for the new advances.

Programming steps involved in the development of a project are given below:

Step 1: To create a NEW project, go to PROJECT drop down menu and select "New Project". Input a file name and save it. The screen shot of Step 1 is shown in fig.2.

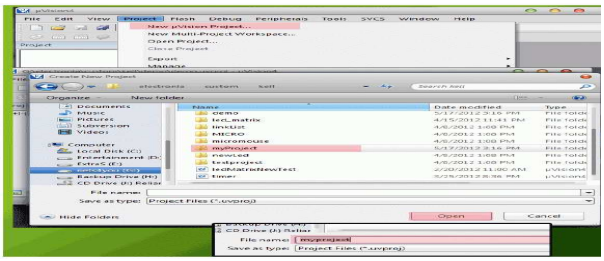


Fig 2. *Screen shot for creating a New Project.*

Step 2: In this step select the chip to be programmed. Here choose LPC1768 from NXP (Philips). After the selection of the chip, a message box will be displayed and asks for to load the start up code in to the project, click on OK. The screen shot is shown in fig. 3.

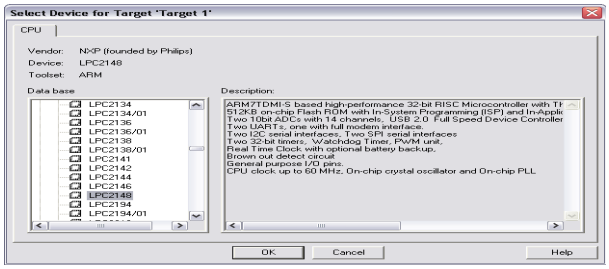


Fig 3. *Screen shot for Step 2*

Step 3: To start entering the code, select new file from file drop down menu, on right side Editor will be displayed where the code can be written. After typing C program, Save the file and configure the file with extension to be ".C". The command used here is "File→ Save as.... The screen shot is shown in fig. 4.

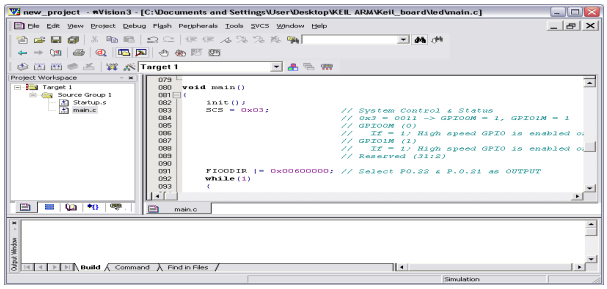


Fig 4. *Screen shot for step 3*

Step 4: In this step, add files in to project file using the commands "Project→ Components, Environment, Books", then select the desired Add file to add to project file. The screen shot of which is shown in fig. 5.

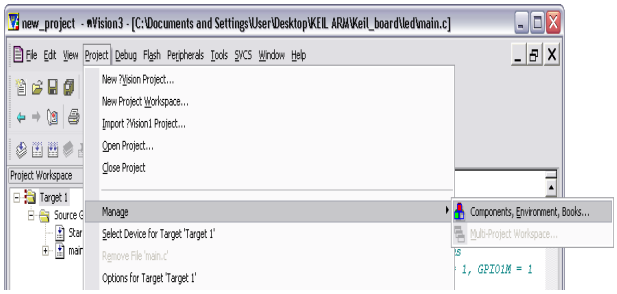


Fig 5. *Screen shot for step 4*

Step 5: After completion of Step 4, right click on target in project workspace, select "options for Target" and click on output to create "HEX FILE" and finally click on OK, the screen shot of which is shown in fig. 6.

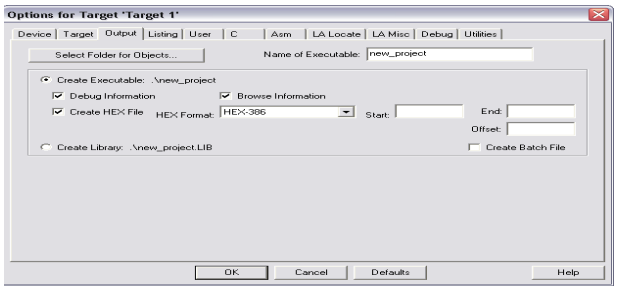


Fig 6. *Screen shot for step 5*

Step 6: In this step build the project using project drop down menu. Click on Build target.

Step 7: Upload the Hex File using the "Flash Magic". In this browse the hex file, click on START to upload to flash. The screen shot is shown in fig. 7.

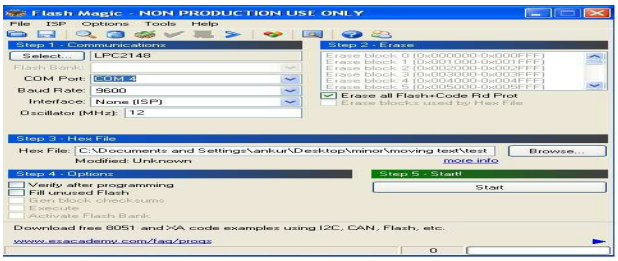


Fig7. *Screen shot for step 7*

3. Details of Individual Experiments

Before describing the projects in Module C, a few experiments are developed and described in modules: A and B. The first module explains the hardware and software details of the experiments specific to the exploitation of GPIO pins, which comes under the category of basic input/output interfacing. Module B pertains to the hardware and software description of experiments for exploitation of on-chip peripherals. A project based embedded application with sensor interfacing and their applications are described in Module C. In all these modules, the student is expected to use theoretical knowledge in assembling the hardware interface circuit and the practical aspect is fulfilled by the successful software development, and verifying the result by powering up the circuit.

Module A: Basic I/O Interfacing

This part will help the student in the first step of learning embedded system with simple interfacing such as LED, Seven segment display, Stepper Motor, Relays and LCD interfacing.

Ex .A 1: LED Blinking

The first experiment is LED toggling. This experiment is generally named as Hello Program in embedded systems. Light Emitting Diode (LED) is the most commonly used component, usually for displaying the digital status of I/O pins. At this juncture, the student is asked to refer to the user manual of LPC1768 [13]. This is for the reason that certain port pins are declared as open drain. To get acquainted with GPIO pins, LEDs are connected to port pins P0.4 to P0.7 with current limiting resistors (220Ω), mounted on a bread board. The embedded C program is developed using Keil µVision software.

The software is developed based on the following algorithm:

- Initialize the board
- Set the IODIR register to configure the GPIO pins as output pins.
- Manipulate the I/O pins using IOSET register and IOCLR register.
- Build the target
- Upload the resulting hex file using Philips Flash utility.
- Press the RESET button and
- Observe the shifting of LEDs in the form of nibbles connected to PORT pins

Figure 8 shows the photograph of the experiment.

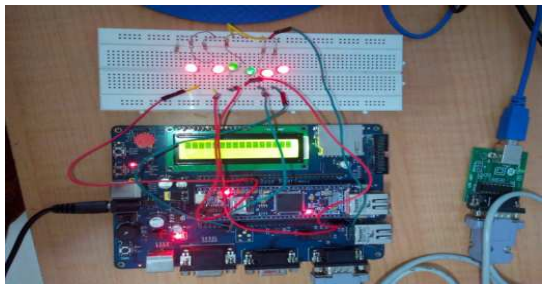


Fig 8. *LPC1768 interfaced with LEDs*

Ex A 2: Seven Segment Display

A seven segment display is the most basic electronic display device that can display the digits from 0-F (hexadecimal numbers). The seven segment pins (a,b,c,d,e,f,g) plus the decimal point of a common anode display are connected to port pins of LPC1768 via current limiting resistors (220Ω). The program is developed using Keil software to display hexadecimal numbers 0-F on the display. The software is developed based on following algorithm.

- Initialize the board
- Initialize the seven segment display with variables.
- Set the control direction register IODIR to configure the GPIO port pins as output pins.
- Display the hex numbers on seven segment display in up counter manner.

Figure A.2 shows the photograph of the experiment

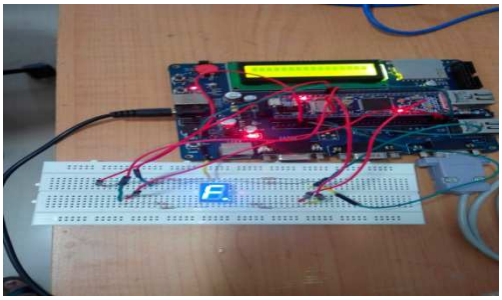


Fig 9. *LPC1768 interfaced with seven segment display, Hex value 'F' is displayed on the seven segment display*

A.3. Stepper Motor Interfacing Experiment

In this experiment the stepper motor is interfaced with LPC1768 using port pins P0.10, P0.11, P1.25, and P1.26. As the port outputs of the controller cannot drive the stepper motor directly, the power amplifier IC ULN2003 is used. It provides the necessary current to drive the motor. Collect a stepper motor from the junked floppy drive mechanism, IC ULN2003, RESET components, connecting wires and bread board. Assemble the circuit shown in figure 19, on the bread board. The stepper motor is rotated in steps continuously, by developing the software basing on the following algorithm:

- Initialize the board
- Initialize the variables to rotate the stepper motor
- Configure the GPIO pins as output pins, using IODIR register.

- Send the data sequence to rotate the stepper motor in clock wise direction by manipulating the port pins using IOSET and IOCLR register.
- Give a delay
- Stop the motor in low-power mode.

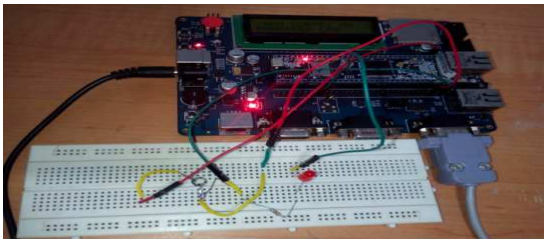
Ex.A.4. Push button interface, debouncing and Toggling LED

The main aim of the experiment is to debounce a mechanical switch by software while counting the number of times the button is pressed. The decimal number corresponding to the number of pushes after debouncing is observed using LED connected to P0.10. Push button is connected to P1.25 pin. Collect a mechanical switch, LEDs, RESET components, bread board and connecting wires. Assemble the circuit shown in figure 23.

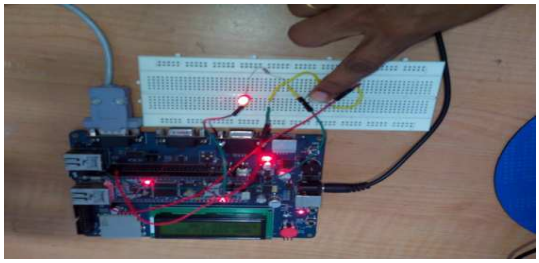
Program is developed to observe the number of toggles of LED after software debouncing, using the following algorithm:

- Initialize the board
- Configure pin P1.25 as input using IOPIN register.
- Configure P0.10 as output which is connected LED to observe the debouncing.

Figure 11 shows the photograph of the experiment.



(a)



(b)

Fig 10. *(a). LPC1768 interfaced with Push button. (b). Push button debouncing, LED toggling*

A.5. Relay Interfacing Experiment with LPC1768

In this experiment a Relay is interfaced with LPC1768 using port pins P0.10. As the port outputs of the controller cannot drive the Relay directly, the power amplifier IC ULN2003 is used. It provides the necessary current to drive the Relay. IC ULN2003, RESET components, connecting wires and bread board are the required components.

The software is developed based on the following algorithm:

- Initialize the board.
- Configure the GPIO pin P0.10 as input pin, using IODIR register.
- Give a delay
- Make P0.10 low using IOCLR and high using IOSET register
- Toggle the LED when the relay goes high state.

Figure 11shows the photograph of the experiment.

Fig 11. *LPC1768 interfaced with Relay*

With the description of the above experiments, Module A comes to an end. Thus Module A has focused more on basic input/output (I/O) programming. Performing these experiments, gives confidence in programming and working with the LPC1768 microcontroller, in standalone mode. The hardware circuit of all the I/O interfacing experiments described above is shown in figure 12.

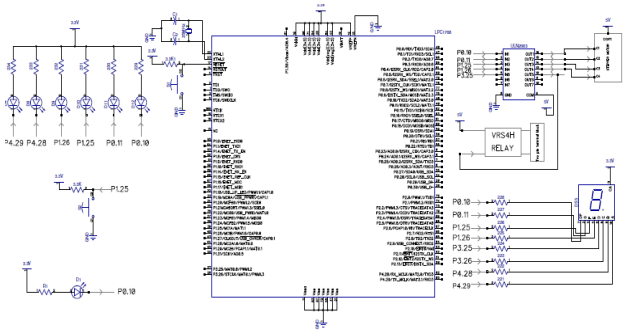


Fig 12. *Total hardware for experiments A.1 to A.5.*

Module B: Interfacing Experiments to Exploit the on-Chip Peripherals

One of the interesting features of LPC1768 (ARM CortexM3) is that each pin of LPC1768 can perform up to four different functions, which are selectable. This module gives the information about interfacing of the on-chip peripherals (Analog to digital converter (ADC), Pulse Width Modulation (PWM), Serial Communication using USART, Ethernet (EMAC), Timers, Watch Dog Timer (WDT)) with the outside world.

Ex B1: Measurement of Analog Voltage Using On-Chip ADC

The ADC present on LPC1768 is a 12-bit successive approximation converter, with a conversion rate of 200 kHz. The on-chip ADC0: Channel 5 (ADC0.5) is chosen in the present experiment to measure the analog input voltage. The software is developed based on the following algorithm:

- Initialize the board.
- Configure GPIO pin P1.31 as AD0.5 using PINSEL

register and power up ADC using PCONP register.

- Select AD0.5, CLKDIV, CLKS and PDN functions using ADCR0.
- Select ADDR5 register to read the digital equivalent of the analog input.

A linear 10kΩ / 10 turn (Bourns, USA), potentiometer is connected to AD0.5 as shown in the circuit diagram. Fig. 13 shows the photograph of the experiment.

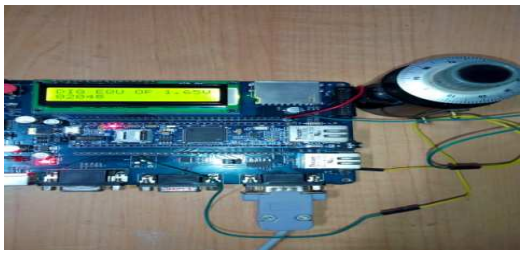


Fig 13. *Photograph of the Analog input voltage.*

Ex B2: Pulse Width Modulation (PWM)

The PWM modulator of LPC1768 is capable of producing six channels of single edge controlled PWM or three channels of dual edge controlled PWM.

The PWM module is used to generate a single channel of symmetrically modulated PWM signal. Software is developed based on the following algorithm:

- Initialize the board.
- Configure P0.7 as PWM2 output using PINSEL register and power up PWM using PCONP register.
- Load the Prescaler to 30 MHz
- Configure Match 0 register to reset the counter when a match event occurs.
- Configure all PWM related Match registers to toggle on match.
- Reset the counter to start the PWM program using PWMTCR register.

Figure 14 shows the photograph of the experiment.

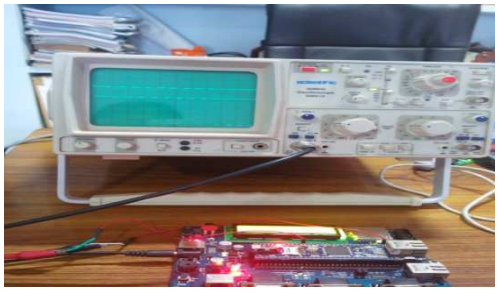


Fig 14. *Photograph of the PWM output*

Ex B3: Implementation of UART

Serial communication is easy and efficient protocal for tranferring data for long distances [15]. The LPC1768 has four on-chip USARTs with fractional baud rate generation, internal FIFO, IrDA, and DMA support. One UART has modem control I/O and RS-485/EIA-485 support.

The clock frequency for the USART is selected using software and baudrate 9600bps is set with USART Prescalar registers. Components required: MAX232, DB-9 connector, RESET components, capacitors and conecting wires. Program is developed on LPC1768 to send the characters to personal computer using the following algorithm:

1. Initialize the board.
2. Power up UART1 using PCONP register.
3. Configure peripheral clock for UART1 using PCLK SEL1 register.
4. Select P0.15 and P0.16 as TXD and RXD pins
5. Enable UART1 Transmit and Receive
6. Select 8 bit data length, no parity,1 stop bit using U0LCR register
7. Character "LPC1768 UART1 Done" is sent through LPC1768 to desktop hyper terminal, continuously.

Ex B4: Easy Web Application with on-Chip Ethernet

Ethernet block of LPC1768contains a full featured 10Mbps or 100Mbps Ethernet MAC (Media access controller) designed to provide optimized performance through the use of DMA hardware. It has many application areas [17]. LPC1768 Ethernet standard has the following features:

- Supports 10 or 100 Mbps PHY devices including 10 Base-T, 100 Base-TX, 100 Base-FX, and 100 Base-T4.
- Fully compliant with IEEE standard 802.3.
- TCP/IP Transmission Control Protocol / Internet Protocol

The software is developed based on the following algorithm:

- Initialize the system.
- Set the Ethernet power/clock control bit using PCONP register.
- Set the PINSEL2&3 register to configure Ethernet function.
- Set the MAC configuration register 1to pass all receive frames, set MAC register2 to append CRC and pad out frames.
- Enable Reduced MII interface
- Set the receive and transmit descriptor base address register
- Initialize TCP, clear HTTP server's flag register.
- Configure pin P0.10 as output pin using IODIR register.

The program developed is uploaded to the flash memory of LPC1768; browse the webpage using IP address, observe the LED on, off which is connected to P0.10.

Figure 15 shows the Photograph of easy web application.

Fig 15. *Photograph of the Easy web application.*

The hardware circuit of all the peripheral interfacing experiments described above and project based sensor interfacing experiments is shown in figure 16.

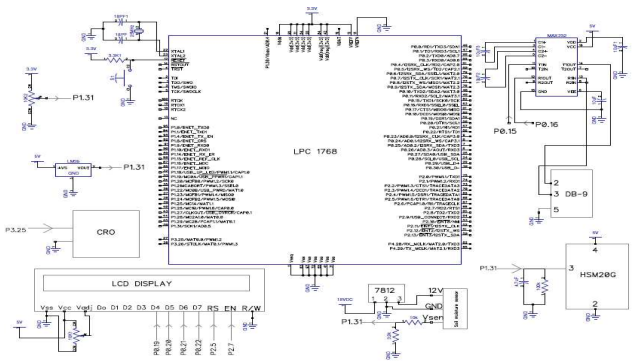


Fig 16. *Hardware Schematic of Peripheral interfacing and sensor Interfacing.*

Module: C: Project Based Embedded Application with Sensor Interfacing Experiments

Ex C1: LM35 Interfacing with LPC1768

The LM35 is internal signal conditioned temperature sensor [17]. Analog input channel of AD0.5 P1.31 is

selected for reading LM35. Components required: LM35, LCD, RESET components and connecting wires. The software is developed based on the following algorithm.

1. Start.
2. Initialize Analog channel AD0.5 using PINSEL function

and power up ADC using PCONP register.

3. Initialize port pin for LCD.
3. Display “Temperature in C” on first line.
4. Start the ADC conversion for reading analog input from LM35.
5. Convert the analog data into temperature and display on second line of LCD.

Ex C2: HSM-20G Interfacing with MSP430F149

The HSM 20G module is interfaced with LPC1768 analog input channel of (A5) ADC0.5 to measure the relative humidity present in air. Components required: HSM-20G, LCD, RESET components, and connecting wires. The software is developed based on the following algorithm:

1. Start.
2. Initialize Analog channel AD0.5 using PINSEL function

and power up ADC using PCONP register.

3. Initialize port pin for LCD.
4. Display “Humidity “on first line.
4. Start the ADC conversion for reading analog input from HSM 20G sensor.
6. Convert the analog data into humidity and display on LCD.

Ex C3: Soil Moisture Sensor interfacing with LPC1768

The analog input channel (A5) of ADC0.5 is selected for measurement of analog output voltage from soil moisture sensor. Software is developed based on the following algorithm

1. Start.
2. Initialize Analog channel AD0.5 using PINSEL function

and power up ADC using PCONP register.

3. Initialize port pin for LCD.
4. Display “Humidity “on first line.
4. Start the ADC conversion for reading analog input from HSM 20G sensor.
5. Convert the analog data into humidity and display on second line of LCD.

Figure 17 shows the photograph of the experiment

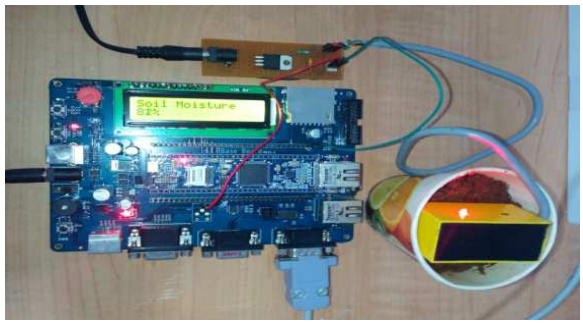


Fig 17. *Photograph of Soil moisture sensor interfaced with LPC1768.*

Ex C4: RFID Sensor Interfacing with LPC1768

In this experiment RFID reader is interfaced with LPC1768. The RFID proximity OEM Reader Module has a built in antenna in minimized form factor. It is designed to work with the frequency of 125 kHz [18, 19].

The software is developed based on the following algorithm.

1. Initialize the board.
2. Power up UART1 using PCONP register.
3. Configure peripheral clock for UART1 using PCLK SEL1 register.
4. Select P0.16 as RXD pins
5. Enable UART1 RXD and receive FIFO
6. Select 8 bit data length, no parity,1 stop bit using U0LCR register
7. Display the read data from RFID tag on the LCD display module.

Figure 18 shows the photograph of the experiment.

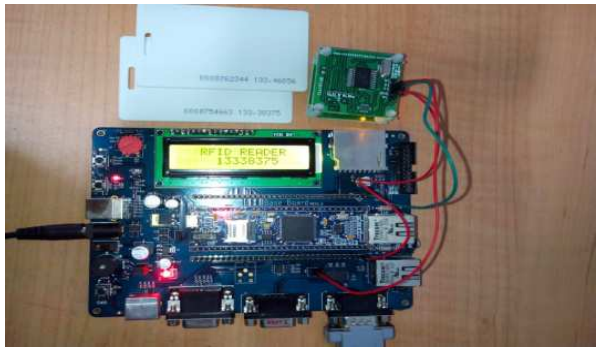


Fig 18. *Photograph of RFID (125 kHz) interfaced with LPC1768*

4. Conclusions

This paper presents pedagogical hands-on experiments for improving students learning in embedded systems and preparing students to be useful in industry. Majority of the components used are inexpensive. Our experience with the conduct of a 120 min-end of semester practical examination show that, with the chronological increase in hardware design, the student exhibited the confidence in designing new stand-alone systems with fairly complicated hardware and software.

Acknowledgment

K.Aruna is thankful to the University Grants Commission (UGC), New Delhi, for sanctioning Junior Research Fellowship (F.4-1/2006 (BSR)/11-23/2008(BSR JRF).

References

[1] Mitsui. H, Kambe. H , and Koizumi,. H " Use of Student Experiment for Teaching Embedded Software Development Including HW/SW Co_Design" , IEEE Transactions on Education, vol. 52, pp. 436-443, Aug. 2009.

[2] Crespo. A, Vila. J, Blanes. A and Ripoll . I " Real time education in a control engineering Curriculum", in 3rd IEEE Real time systems education workshop, 1998, pp. 112-116.

[3] K. G. Ricks, D. J. Jackson and W. A. Stapleton, " Incorporating Embedded programming skills in to an ECE Curriculum", SIGBED Rev., vol .4, no.1, pp .17-26, Jan. 2010.

[4] R. Bachnak, " Teaching Microcontrllers with hands-on hardware experiments", Journal of Computing Sciences in Colleges, 20(4):207-213, 2005.

[5] M.J. Callaghan, J.Harkin, C. Peters, T.M. Mc Ginnity, and L.P. Maguire, " A collaborative environment for remote experimentation", In Proceedings of the 2003 IEEE International Conference on Microelectronic Systems Education (MSE'03), 2003.

[6] D. Beetner, H. Pottinger, and K. Mitchell, " Laboratories teaching concepts in microcontroller and hardware-software co-design. In proceedings of the 30th Annual Franties in Education Conference, Vol.2, Pages SIC/1-SIC/5, Kasas city, Missouri, USA, october 2000.

[7] Jean-Samuel Chenard, Zeljko Zilic, Milos prokic, " A laboratory setup and Teaching Methodology for wireless and Mobile Embedded Systems", " IEEE transactions on Education", vol.51, No.3, August 2008, pp. 378-384.

[8] Jiang Xiaoluo and Li Han, " CDIO – Based Embedded systems training mode in graduate teaching", 2011 5th International conference on distance learning and education, vol. 12, pp. 78-82.

[9] D. Davcev, B. Stojkoska, S. Kalajdziski, and K. Trivodaliev, "Project Based learning of embedded systems", in Proceedings of the 2nd WSEAS Telecommunications, 2008.

[10] S. Nooshabadi and J. Garside, "Modernization of teaching in Embedded Systems design–An international collaborative Project", *IEEE* Transactions on Education, vol. 49, no. 2, pp. 254-262, May, 2006.

[11] J. W. Bruce, J. C. Harden, and R. B. Reese, "Cooperative and Progressive design experience for embedded systems", IEEE Transactions on Education, vol.47, no.1, pp.83-92, Feb. 2004.

[12] Silik. C, Nagvaara, P. Taskin. B, "A Microcontroller based Embedded system design course with PSoC3", " IEEE International conference on Microelectronic Systems Education (MSE), June. 2-3, 2013, pp.28-31.

[13] Wilfried Elmenreich, Christian Trodhandl and Bettina weiss, "Embedded system Home Experimentation", Proceddings of the 2003 IEEE International Conference on Microelectronic systems Education (MSE" 03).

[14] www.ngxtechnologies.com

[15] LPC1768 User Manual NXP Semiconductors-UM10360

[16] http://www.keil.com

[17] www.ti.com/lit/ds/symlink/lm35

[18] Limpraptono, F.Y, Sudibyo. H, Ratna. A.A.P, "The design of Embedded Web server for remote laboratories microcon troller system experiment", "IEEE Region Conference on TENCON 2011", pp.1198-1202, Nov. 21-24, 2011.

[19] www.nskelectronics.com/rfid_reader