Design and Development of a Low-cost Embedded System Laboratory using TI MSP430 LaunchPad

Naveen Kumar Uttarkar, Raghavendra Rao Kanchi*

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

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

Abstract: In this paper, we present a series of experiments starting from Hello World!, up to project-based experiments using Texas Instrument (TI) MSP430 LaunchPad, which is an inexpensive kit from TI, USA. Two dedicated boards are designed and developed to perform experiments described herein, which are an outcome of the author's experience in teaching computer architecture and embedded systems in theory. These experiments can be introduced to the under-graduate students of engineering: electrical, electronics and communication, computer science, and post-graduate science students specialized in electronics. The salient feature of experiments presented in different modules is that, each experiment is explained by its hardware description, software development which includes the dumping of program on to the microcontroller's flash memory, leading to a hands-on experience. 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. Setting up a laboratory of this type is easy with the material described in this paper. A small alteration is suggested on MSP430 LaunchPad.

Keywords: Embedded Lab, MSP430 Launchpad, Stand-Alone Systems, Project-Based Experiments

1. Introduction

Electronic field such as automated homes, home appliances, intelligent automobiles, smart phones, robotics, wireless communication, internet, pervasive and ubiquitous computing are the fruits of the tree which has branches of embedded systems, embedded applications and embedded market [1, 2]. The embedded computer market uses various processors from different vendors. Certain of them are AVR, Atmel, PIC, Texas Instruments, Free scale, Coldfire, NXP (previously Philips), ARM etc. The embedded-computing market is dominated by 32-bit ARM processors. Applications of such processors include: mobile phones, photocopiers, cameras / camcorders, house hold appliances etc. Around 3 billion ARM microcontrollers are manufactured and shipped every year by companies like: Analog Devices, Texas Instruments, Atmel Corporation, NXP, Free Scale and others. ARM itself doesn't manufacture the chips; they simply license their intellectual property to other third party manufacturers [3].

Computer science students are less exposed to hardware and related software, little electronic design expertise, and are exposed to program a wide variety of systems. In spite of all these disadvantages, a computer science student can also step into the field of embedded systems by (learn-while doing) doing certain simple in-house experiments. Though it takes time to learn, it is always advisable to choose a bottom-up approach which includes low-level hardware and software. Further, it is being put as modular approach in computer science terms; it is advisable to take a step-by-step modular approach in learning embedded systems. A hands-on approach is always an added advantage besides the class room teaching [4-12].

Students from electrical, electronics and computer science disciplines are the ones who are suitable for such kind of training. Among these branches electronics and electrical engineering students have an edge compared to computer science students. This is because embedded systems are ones that are fully integrated into the physical structure of products and machine which have no keyboard, mouse or software tools unlike a desktop/laptop. Thus embedded system is characterized by:

- Systems with no boot process (Starts working as and when the power is applied).
- Systems which works in real time.
- Systems which does a specified (applied) task.
- Systems which are less power hungry compared to this counterparts.

There are several microcontrollers that are available on the market supplied by different vendors. Those vendors who provide a training platform, free software (with limited memory programming), without resource base and pre-requisite knowledge are limited. One such company that provides training platform and make their low-power advanced processors available at a cheaper and affordable price is Texas Instruments. This gives an opportunity for students and hobbyists to get training in the field of embedded systems at an affordable cost. Further, some of the key features of present generation processors put on the market by Texas Instruments are:

- Low to Ultra-low power consumption.
- Several programmable power saving modes.
- On-chip/On-board emulation and debugging features.

These features keep the student/hobbyist/developer in a comfortable room.

Thus, one of the best ways of entering into embedded system learning and development platform is using Texas Instrument's microcontroller: MSP430 Launch Pad, which was introduced in July 2010. Further, the newly introduced (January 2012) Energia open-source electronics prototyping platform creates an Aurdino like environment to the Texas instrument's MSP430 based Launch Pad [13].

2. Software and Hardware Description

In spite of the fact that recently introduced Energia prototyping platform is arduino like environment, we chose IAR workbench as the software development platform. This is for the reason that IAR workbench gives more insight into the hardware architecture of MSP430 series mixed signal controllers. Detailed discussion on the architecture description of MSP430 series controllers and programming procedure using IAR Work Bench are already explained by us [14]. Hence, limited and necessary description related to the present work is given here. Thus, IAR Workbench is used throughout for software development.

A new MSP430 LaunchPad kit was purchased from NSK Electronics, Bangalore [15]. The box contains the following components:

- MSP-EXP430G2 board which includes emulation hardware
- MSP430G2553-1 quantiy
- MSP430G2452-1 quantity
- 32.768 kHz watch crystal-1 quantity
- Micro USB-USB cable-1 quantity

- Female berg sticks (10 pin)-2 quantity

Conventional usage of this board is to insert carefully the target IC in to the base of MSP-EXP430G2 board, and program the IC by connecting the micro USB connector to the LaunchPad, and the USB to laptop/desktop, dump the hex file and verify the result. That is, after programming, hardware connections are to be made from the on-board male berg stick connectors for interfacing with the outside world. Further certain I/O pins of the MSP-EXP430G2 board are connected permanently to LED and push buttons. Hence these pins cannot be used for general purpose interfacing. This precludes the exclusive use of certain of I/O pins of the target IC. Another problem with the LaunchPad is that the target IC has to be inserted and removed from the 20-pin IC base provided on the MSP-EXP430G2 board, time and again. This will not only spoil the IC base but also the pins of the IC may not sit properly on to the base.

We have solved this problem by soldering a ZIF socket in the place of IC base. By this technique, the programmed IC can be removed easily and can be used in dedicated application hardware. Even without soldering, the ZIF socket fits tightly in to the IC base of the LaunchPad board. Figure 1 shows the MSP430 LaunchPad board in which a 20-pin ZIF socket is placed on the IC base.

The pin configurations of ICs [16, 17] accompanying the LaunchPad are shown in figure 2(a) and (b). The general architecture of MSP430 series mixed signal controller is shown in figure 3.
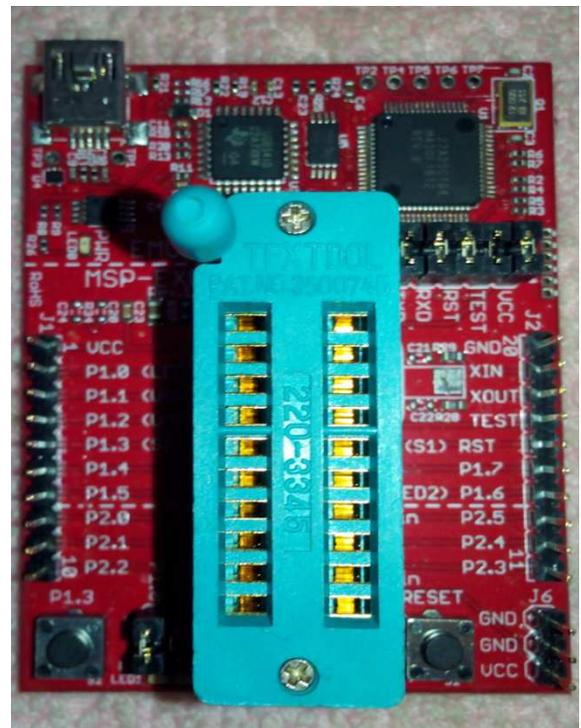
Fig 1. *Photograph of the MSP-EXP430G2 board with ZIF Socket mounted to IC base.*
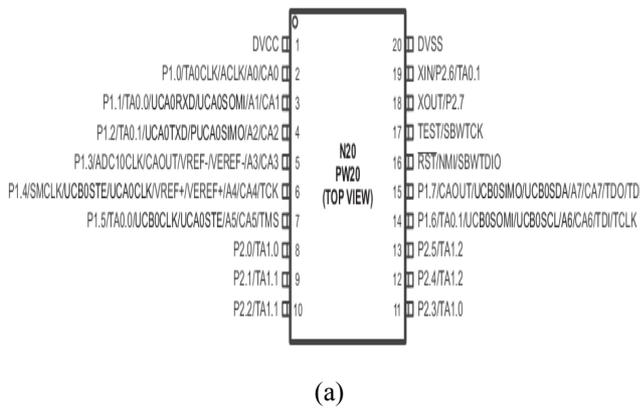
(a)
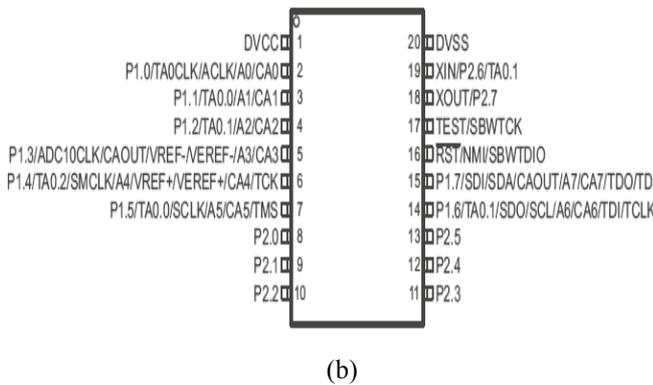
(b)

Fig 2(a). *Pin configuration of MSP430G2553 (Texas Instrument's, USA)*
(b). Pin configuration of MSP430G2452 (Texas Instrument's, USA)
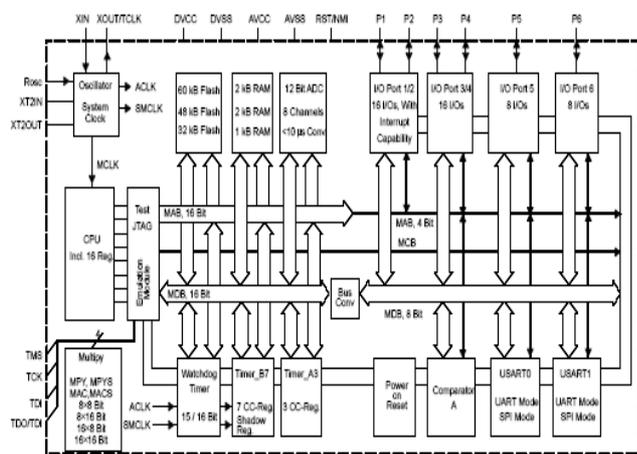
Fig 3. *General architecture of MSP430 series mixed signal microcontroller (Texas Instrument's, USA)*

A couple of dedicated boards have been designed and developed in the laboratory to perform experiments related to the I/O ports and exploitation of the on-chip peripherals presented by the IC MSP430G2553. Photographs of these boards are shown in figure 4. Schematic diagrams of these boards are shown in figure 5.
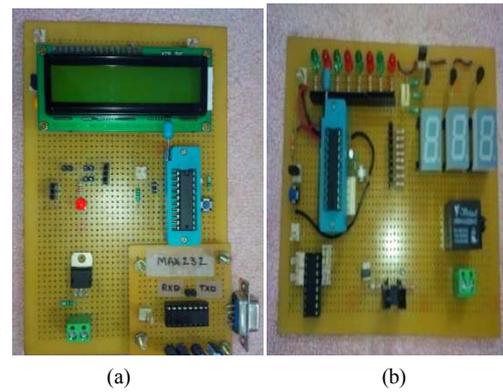
(a)   (b)

Fig 4(a). *Photograph of The on-chip peripheral exploitation board (Peripheral board) (b). The I/O programming board (I/O board)*
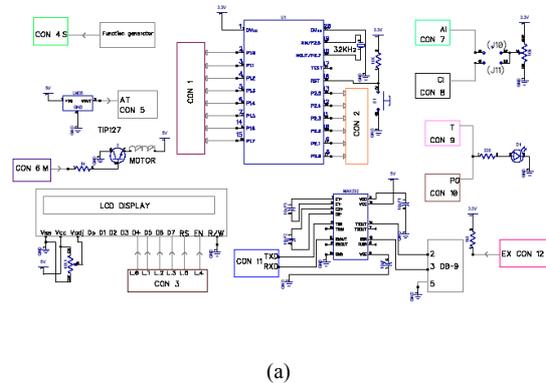
(a)
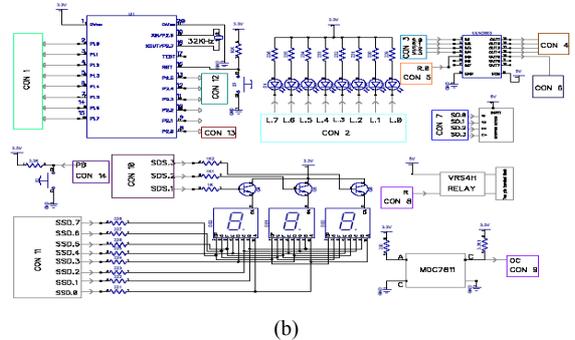
(b)

Fig 5. *(a). Schematic diagram of on-chip peripheral exploitation board (b). Schematic diagram of I/O programming board*

The I/O board designed and developed in the lab contains the following components:
- Reset switch
- Push button
- Relay
- Optocoupler
- Array of eight LEDs
- 3-Digit seven segment display
- ULN2003 (Darlington Pair Array)
- CON1, CON2, Jumpers (Male Berg Pins)
- The peripheral board contains the following components:
- Reset switch

- LCD module
- MAX232 with DB-9 connector
- TIP127 (Power transistor)
- Jumpers (Male Berg Pins)

3. Details of Individual Experiments

All the experiments designed and developed in the present work are divided into three modules: A, B and C. Module A describes the software and related hardware connections (I/O board) related to the I/O ports of MSP430G2553. Experiments related to the on-chip peripheral programming are presented in Module B. Project related experiments, which are designed and developed in our laboratory are explained in Module C. In all these experiments, MSP430 LaunchPad board is connected to the laptop's USB port and downloaded the hex file related to the developed program into the flash memory of MSP430G2553. Then the IC is removed from the ZIF socket and placed on the ZIF socket of the target board. The target board is powered up with 3.3V/5.0V dc supply.

Module A: This module contains interfacing experiments, relating to I/O programming.

3.1. Ex.A.1: Led Blinking (Hello World!)

This is a starting experiment in embedded systems. Software to make one of the LEDs connected to port P1 (P1.0) is made to toggle between logic 0 and logic 1, with a delay in between. Software is based on the following algorithm:

1. Start
2. Stop watchdog timer
3. Make P1.0 pin as output
4. Make P1.0 high logic
5. Delay
6. Make P1.0 low logic
7. Keep P1.0 toggling in infinite loop

The hex file of the program following the above algorithm is dumped on to MSP430G2553. The IC is removed and placed on the I/O board. Jumper is connected between P1.0 of CON 1 and L.0 of CON 2. The board is powered up to observe the LED blink. Figure A.1 shows the photograph of the LED blink working setup.

Fig A 1. *Photograph of LED blinking.*

3.2. Ex.A.2: Push Button Count Display on Three Digit Seven Segment Array (Including Software Debounce)

In this experiment, program is developed to count the number of pushes of the on-board push button (PB1). As the push button is a mechanical switch, it has to be debounce to get a true count of the pushes made. The process of debouncing is accomplished through software, and the true count of the pushes is displayed on the on-board 3-digit seven segment displays of the I/O board. Jumpers are connected between CON1 and CON2. The software is developed using following algorithm:

1. Start
2. Stop watchdog timer
3. Make port P1 as output port and P2.0 as input
4. Input the PB1 status
5. Increment the counter and debounce the PB1
6. Display the count on 3-digit 7_segment display

Fig A 2. *Photograph of the number of pushes displayed on three digits seven segment displays.*

3.3. Ex.A.3: Interfacing Stepper Motor

Here the software is developed to rotate the stepper motor in clock wise continuously. In the software port pins P1.0-P1.3 are used to energize the coil and rotate the stepper motor in clock wise direction. The software is developed using IAR Workbench, based on the following algorithm:

1. Start
2. Stop watchdog timer
3. Make P1 as output port
4. Keep 0x66 data right shift and send through port P1 with delay in infinite loop.

The hex created is dumped on to MSP430G2553. The IC is removed and placed on the I/O board. The jumper is used between connector 4 (CON 4) and connector 7 (CON 7) for stepper motor connections. The board is powered up to observe the continuous clock wise rotation of stepper motor. Figure A.3 shows the photograph of the interfacing stepper motor setup.
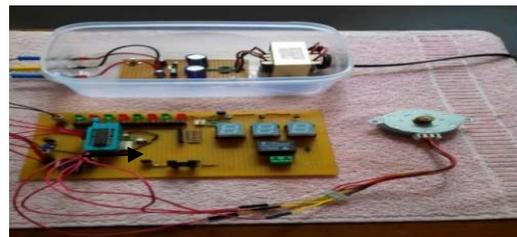
Fig A 3. *Photograph of the Interfacing stepper motor.*

3.4. Ex.A.4: Interfacing Relay

High power devices are connected to microcontrollers using relay. In this experiment, a relay (VRS4H) [18] is controlled by the following algorithm. An LED is connected as load to the relay. Relay is connected to P1.0 via ULN2003 [19] using a jumper. Energizing the relay turns the LED 'ON' and de-energizing the relay switches the LED to' OFF'.

- Stop watchdog timer
- Make P1.0 pin as output
- Make P1.0 high logic
- Delay
- Make P1.0 low logic
- Keep P1.0 toggling in infinite loop

The hex file of the program is dumped on to MSP430G2553. The IC is removed and placed on the I/O board. The jumper between pin P1.0 of connector 1 (CON 1) and pin R.0 of connector 5 (CON 5) is used to drive VRS4H relay via ULN2003 [19]. Figure A.4 shows the photograph of the driving LED via relay.
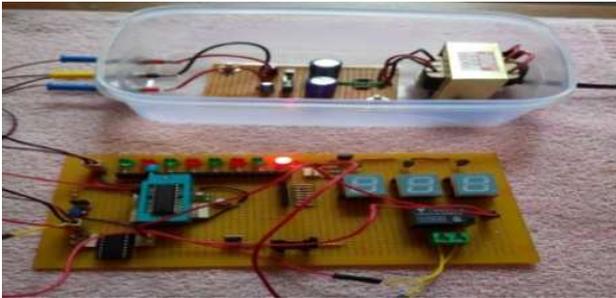
Fig A 4. Photograph of relay interfacing.

3.5. Ex.A.5: Interfacing Opto-Coupler (MOC7811)

An optocoupler is a sensor used in different applications. For example, in printers, if the printer is out of paper, it will be detected by an optocoupler, activating a buzzer or turning on a LED. In the present set up, software is developed to detect an obstacle (paper), and realization of it through toggling LED state. In the software pin P1.4 is used to read the obstacle presence from MOC7811 [20] and LED connected to P1.0 is used for toggle. The software is developed based on the following algorithm:

1. Start
2. Stop watchdog timer
3. Make P1.0 as output pin
4. Make P1.4 as input pin
5. Enable rising edge interrupt for P1.4 pin

The hex file of the program is dumped on to MSP430G2553. The IC is removed and placed on the I/O board. The jumper between pin P1.4 of connector 1 (CON 1) and pin OC of connector 9 (CON 9) is used to read the state of the optocoupler. The jumper between pin P1.0 of connector (CON1) and pin L.0 of connector 2 (CON 2) is used. Figure A.5 shows the photograph, LED ON indicating an obstacle.
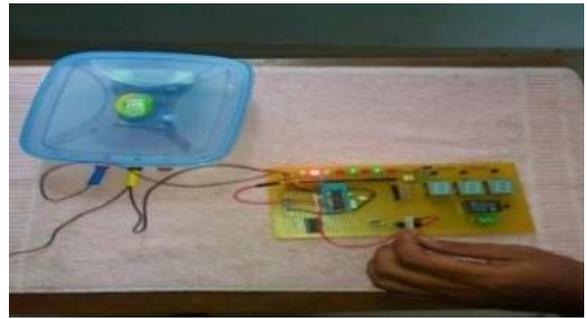

Fig A 5. Photograph of the detecting obstacle detection using opto-coupler.

Module B: In this module, we explain the use of on-chip peripherals such as: comparator, timers (Timer A, Timer B), 10-bit ADC, UART, temperature sensor.

3.6. Ex.B.1: Verification of On-Chip Comparator Functionality

MSP430G2553 contains an on-chip comparator (8-channel). The functionality of the comparator is tested by connecting its non-inverting input to a variable voltage source (variable terminal of 10kΩ/10-turn pot.). The inverting input of the comparator is internally fed with 0.25*Vcc=0.25*3.3V=0.825V. In the software P1.1 is selected as comparator input pin and the state of the comparator is realized using LED connected to P1.0. Such a comparator is useful in realization of ADC using on-chip DAC.

Software is developed based on following algorithm:
1. Start
2. Stop watch dog timer
3. Set P1.0 as output pin and P1.1 as input
4. Select the reference voltage (0.25*Vcc)
5. Poll P1.1 and set the LED status accordingly
6. Repeat the above steps

The hex file of the program is dumped on to MSP430G2553. The IC is removed and placed on the Peripheral board. A jumper between pin P1.1 of connector 1(CON 1) and connector 8 (CON 8) is used to feed external input voltage to pin P1.1. Figure B.1 shows the photograph.
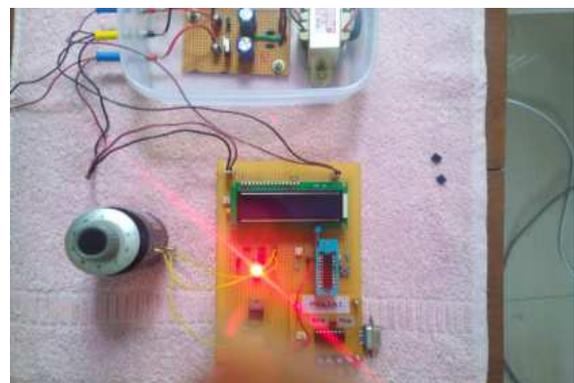

Fig B 1. Photograph for realization of on-chip comparator functionality LED indicates the state of the comparator (Analog input > Reference voltage).

3.7. Ex.B.2: One Second LED Toggling Using TIMER A

One of the important applications of timer is that generation of different time delays or generation of pulses of different widths. Here, TIMER A of the MSP430G2553 is used to the generate one second delay between LED toggle states. For MSP430G2553, watch crystal (32.768 kHz) is used as external clock source. TIMER A is set in counter mode and whenever the over flow of register occurs, TIMER A generates an interrupt. Interrupt service routine contains program to toggle pin P1.0. The software is developed based on the following algorithm:

1. Start
2. Stop watchdog timer
3. Make P1.0 as output pin
4. Load TA0CCR0 register with count 32000
5. Enable counter interrupt
6. Set timer clock source and count up mode

The hex file of the program is dumped on to MSP430G2553. The IC is removed and placed on the Peripheral board. Jumper is connected between pin P1.0 of connector 1 (CON 1) and L.0 of connector 2 (CON 2). LED toggling for every one second is observed. Figure B.2 shows the photograph of the LED toggling for every one second (using timer).

*Fig B 2. Photograph of LED toggling for every one second.*

3.8. Ex.B.3: Realization of Pulse Width Modulation (Using TIMER A)

On chip peripheral TIMER A can be configured for generation of pulse width modulation. Pulse width modulation is an important concept in control systems. TIMER A capture-compare register is used to set the duty cycle of the pulse and watch dog timer is used for 32ms interval timer. The software for generation of pulse width modulation signal is developed basing on the following algorithm:

1. Start
2. Set watch dog timer to 32ms interval interrupt generator
3. Set pwm direction variable with 1
4. Enable watch dog timer interrupt
5. Enable pulse width modulation output pin P1.6
6. Set pulse width modulation period
7. Initialize the TA0CCR1 with 1
8. Enable output mode-7(reset/set mode)
9. Set 1MHz internal clock source as timer clock source and set count up mode

10. For every 32ms the TA0CCR1 increment by 20 so it increases duty cycle and therefore LED and DC motor brightness/ speed   also increases gradually

The hex file of the program is dumped on to MSP430G2553. The IC is removed and placed on the Peripheral Board. The jumper between pin P1.6 of connector 1 (CON 1) and connector 10 (CON 10) is used to realize pulse width modulation on LED. The connector 6 (CON 6) is used to realize PWM on dc motor, connected via TIP127 [21]. Figure B.3 shows the photograph of realization PWM by using LED and dc motor.

Fig B 3. Photograph of PWM realization using LED and DC Motor

3.9. Ex.B.4: Frequency Meter (Using TIMER A)

In the earlier sections, we discussed about different application modes of TIMER A. In this section we configure the TIMER A to measure frequency of external input signal, and UART is used for sending the frequency of signal displayed on the hyper terminal of the laptop. The software is developed based on the following algorithm:

1. Start
2. Stop watch dog timer
3. Enable ACLK (32kHz) and   DCO clock source
4. As 16MHz
5. Initialize the UART for a baud   rate of 9600
6. Keep on frequency measurement and sending to laptop via UART in an infinite loop

The hex file of the program is dumped on to MSP430G2553. The IC is removed and placed on the Peripheral Board. The jumper between pin P1.0 of connector 1(CON 1) and connector 4 (CON 4) is used to feed input signal to P1.0. Figure B.3 shows the photograph of frequency meter.

Fig B 4. Photograph of frequency meter (Frequency of the input signal is displayed on the hyper terminal of the laptop).

3.10. Ex.B.5: UART Echo Experiment

MSP430G2553 contains one on-chip UART. The UART A0 registers UCA0BR0 and UCA0BR1 are used for selecting the baudrate. Here commonly used baudrate (9600 BPS) for serial communication is used for echo program. As the character typed on the keyboard is displayed on the hyper terminal, this is known as echo program. The software is developed based on following algorithm:

1. Start
2. Stop watch dog timer
3. Select P1.1, P1.2 as RXD and TXD
4. Set SMCLK as UART clock frequency
5. Load UCA0BR0 =104 and UCA0BR1=0 (setting UART to 9600 baudrate)
6. Enable UART receive interrupt
7. For every receive of data to microcontroller, microcontroller generates echo same data to laptop

The hex file of the program is dumped on to MSP430G2553. The IC is removed and placed on the Peripheral board. The jumper between pins P1.1-P1.2 of connector 1 (CON 1) and pins RXD TXD of connector 11 (CON 11) are used to send/receive the data to laptop. Figure B.5 shows the photograph of UART echo experiment.

Fig B 5. Photograph of USART echo experiment

3.11. Ex.B.6: Digital Voltmeter (Using ADC10)

MSP430G2553 contains 10- bit ADC which is useful for sensor interfacing. The resolution of ADC is 3.22mV. A0 to A.7 channels of the ADC10 are multiplexed with port P1 pins P1.0 to P1.7. The analog input (centre tap of 10kΩ/10turn pot) is fed to P1.0 (channel A0) is converted to digital data and stored in ADC10MEM register. The software is developed based on following algorithm:

1. Start
2. Stop watch dog timer
3. Initialize LCD
4. Select channel 0 and ADC10CLK as ADC10 clock source
5. Start ADC10 and enable interrupt
6. Keep analog voltage read, conversion and display into infinite loop
7. For every sample time the interrupt is generated then analog voltage is measured and stored into the ADC10MEM register

The hex file of the program is dumped on to MSP430G2553. The IC is removed and placed on the Peripheral board. The jumper is used between connector 2 (CON 2) and connector 3 (CON 3) for 4-bit mode LCD connections. The jumper between P1.0 of connector 1 (CON 1) and connector 7 (CON 7) is used to read analog input voltage. Figure B.6 shows the photograph of realization of digital voltmeter.

Fig B 6. Photograph of the digital voltmeter

3.12. Ex.B.7: Temperature Measurement Using Internal Temperature Sensor

Texas Instrument provides internal temperature sensor as on-chip peripheral for MSP430 core controllers. The calibration formula for the temperature reading is given in the MSP430G2553 datasheet [22]. This equation gives relation between voltage and temperature; with this equation we can measure the temperature of MSP430G2553 surroundings. The software is developed based on following algorithm:

1. Start
2. Stop watch dog timer
3. Initialize LCD in 4-bit mode
4. Select channel 10 (internal temperature sensor channel) and ADC10CLK as ADC10 clock source
5. Start ADC10 and enable interrupt
6. Keep internal temperature reading, conversion and display into infinite loop
7. For every sample time the interrupt is generated then analog voltage is measured and stored into the ADC10MEM register

The hex file of the program is dumped on to MSP430G2553. The IC is removed and placed on the Peripheral board. The jumper is used between connector 2 (CON 2) and connector 3 (CON 3) are used for LCD connections. Figure B.7 shows the photograph of temperature display on LCD.

Fig B 7. Photograph of the temperature measurement on LCD (using internal temperature sensor)

Module C: In this module, we explain the applications of MSP430 LaunchPad (MSP430G2553) with two projects.

They are:
1. Interfacing external temperature sensor (LM35) and displaying the temperature on LCD module
2. Interfacing the soil moisture measurement electrodes and displaying the soil moisture in percentage on LCD module.

3.13. Ex.C.1: Interfacing LM35 Temperature Sensor

LM 35 [23] is self signal conditioned temperature sensor and it has less self heating effect. The LM35 sensor outputs 10mv for every one degree increment. The output of LM 35 sensor is connected to A.0 channel of 10-bit ADC. The software is developed based on following algorithm:
1. Start
2. Stop watch dog timer
3. Initialize LCD
4. Select channel 0 (internal temperature sensor channel) , set SREF_2 (external reference voltage to ADC10), ADC10CLK as ADC10 clock source
5. Start ADC10 and enable interrupt
6. Keep internal temperature reading, conversion and display into infinite loop
7. For every sample time the interrupt is generated then analog voltage is measured and stored into the ADC10MEM register

The hex file of the program is dumped on to MSP430G2553. The IC is removed and placed on the Peripheral board. The jumper between pin P1.0 of connector 1 (CON 1) and connector 5 (CON 5) for LM35 connections and the jumper between connector 2 (CON 2) and connector 3 (CON 3) are used for 4 bit mode LCD connections. Figure C.1 shows the photograph of temperature displayed on LCD (using LM35).
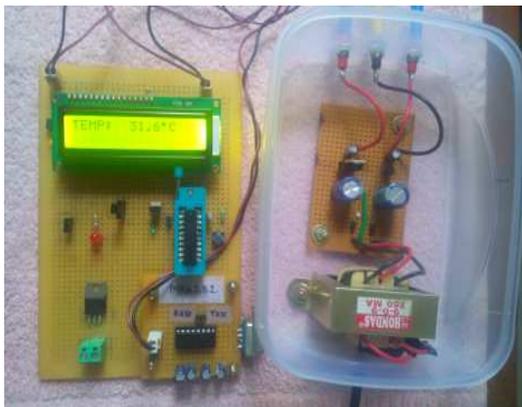
Fig C 1. *Photograph of temperature measurement (using LM35 temperature sensor)*

3.14. Ex.C.2: Soil Moisture Measurement System

Soil moisture sensor [24] is bought from NSK electronics [15]. It gives an output that is proportional to the content of moisture in soil. The operating voltage of this sensor is 12 volts and output is in the span of 0-12V. To convert this voltage into 0-3.3V range, we used 10 kΩ and 3.9 kΩ divider circuit followed by a voltage follower using LM324.

The output of voltage follower is connected to A.0 channel of on-chip10-bit ADC. The software is developed based on the following algorithm:
1. Start
2. Stop watch dog timer
3. Initialize LCD
4. Select channel 0 (internal temperature sensor channel), ADC10CLK as ADC10 clock source
5. Start ADC10 and enable interrupt
6. Keep soil moisture reading, conversion and display into infinite loop
7. For every sample time the interrupt is generated then analog voltage is measured and stored into the ADC10MEM register

The hex file of the program is dumped on to MSP430G2553. The IC is removed and placed on the developed board. Figure C.2 shows the photograph of soil moisture measurement system.
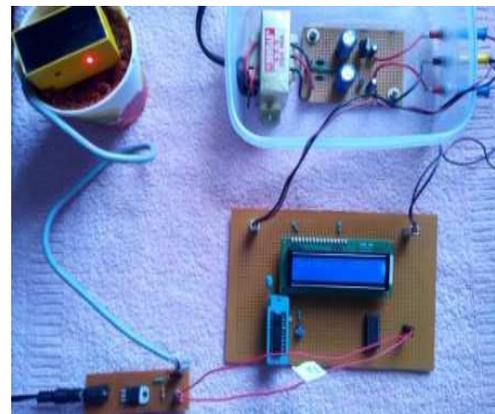
Fig C 2. *Photograph of the soil moisture measurement system*

4. Conclusion

A bunch of experiments are designed and developed using MSP430 LaunchPad. Two dedicated boards are designed and developed in the author's laboratory. A small change in the TI's MSP430 LaunchPad is suggested. The experiments with project-based ones are useful for one third semester laboratory for an under-graduate ECE/CSE laboratory. The hands-on experience a student gets by performing these experiments definitely gives confidence to think and develop a complex embedded system, which is very much essential in the present scenario. The work proposal in this paper can be extended in designing ultra low-power embedded systems, which is very much useful in the research field of embedded system design. Bill of the developed boards is given in below table.

S.No.	Name of the board	Price
1.	Peripheral Board	$5
2.	I/O Board	$2

Acknowledgement

References

[1] POST "Pervasive Computing" Post-Note", MAY 2006 Number 263, The Parliamentary Office of Science and Technology, 7 Millbank, London, SW1P3JA, 2006.

[2] T Basten, M Geilen Ambient Intelligence: Impact on Embedded System Design, Kluwer Academic Publisher, 2004.

[3] B. Benson, A. Arfaee, C. Kim, R. Kastner, R. Gupta, "Integrating Embedded Computing Systems Into High School and Early Undergraduate Education", IEEE Transactions On Education, 2010.

[4] Xiaolin Hu, Minghao Wang, Yuanfeng Xu, Kai Qian, "Modular Design and Adoption of Embedded Systems Courseware with Portable Labs in a Box", Proceedings of the World Congress on Engineering and Computer Science (WCECS 2012) Vol.I, October 24-26, 2012.

[5] www.cs.virginia.edu/sigbed/archives/2009-01/j-3-wese-jour nal-p12-huang.pdf .

[6] Kai Qian, Xiaolin Hu, Liang Hung, "Exterience on Teaching Multiple CS Courses with Portable Embedded System Labware in a Box", Proceedings of the World Congress on Engineering and Computer Science (WCECS 2011), Vol.I, San Fransisco, USA, Oct. 19-21, 2011.

[7] Zalewski,J., "A Comprehensive Embedded Systems Lab for Teaching Web-Based Remote Software Development", 23rd IEEE Conference On Software Engineering Education and Training, Pittsburgh, PA, March 9-12, 2010.

[8] Hamblen, J.O., "An Embedded Systems Laboratory to Support Rapid Prototyping of Robotics and The Internet of Things", IEEE Transaction On Education, Feb. 2013.

[9] Berndt, Don, "A Software Design Method for Embedded Systems", IEEE Real-Time Embedded Technology and Applications Symosium, Nov. 2005.

[10] Fowler, Kim, "Introduction To Developing Embedded Systems", Instrument & Measurement Technology Conference, Jan. 2005.

[11] Berndt, Donald, "A Software Design Methodology For Real-Time Embedded Systems PartII", Developed Exclusively For IEEE Expert Now, April 2010.

[12] Hans Henrik Lovengreen, Anders P.Ravn and Hans Rischel, "Design Of Embedded, Real-Time Systems: Developing A Method For Practical Software Engineering", IEEE Explore, Nov. 2009.

[13] http://www.energia.nu/.

[14] Naveen Kumar Uttarkar, Raghvendra Rao Kanchi, "Design and Development of a Low-Cost Embedded Systems Laboratory Using TI MSP430F149", International Conference on Communication and Signal Processing (ICCSP' 13), April 3-5,2013, India, IEEE Xplore, 2013.

[15] http://www.nskelectronics.com/msp_launchpad_.html.

[16] http://www.ti.com/lit/ds/symlink/msp430g2553.pdf.

[17] http://www.ti.com/lit/ds/symlink/msp430g2112.pdf

[18] http://www.hktdc.com/fair/proddetail/electronicasia-en/1X6 C3BQD/1X15MGTV/electronicAsia/General-Purpose-Rela y.htm

[19] http://www.ti.com/lit/ds/symlink/uln2003a.pdf

[20] http://www.datasheetarchive.com/MOC7811-datasheet.html

[21] http://www.rigelcorp.com/8051/TIP127.pdf.

[22] http://www.ti.com/lit/ug/slau144j/slau144j.pdf

[23] LM35 Precision Centigrade Temperature Sensor, Literature Number: SNIS 159B,Texas Instruments, November 2000.

[24] http://www.nskelectronics.com/soil_moisture_sensor.html