

# An Embedded System Laboratory Training Using dsPIC30F6014A

Swapna Chintakunta and Raghavendra Rao Kanchi\*

**Abstract**— In this paper, we present a series of experiments useful for an embedded system laboratory. 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. They are built around the dsPIC30F6014A low power RISC Digital Signal Controller, which suggest the MPLAB IDE programming. The salient feature of dsPIC30F6014A chip module is that each experiment is explained by its hardware description, software development, which includes the dumping of program on to the microcontroller flash memory. The series of experiments described in the present paper will be a point of take-off to start a journey on working in the area of embedded system. It gives an opportunity to think individually and to develop creativity towards the fast growth of technology about an embedded system.

**Index Terms**— dsPIC30F6014A, on-chip peripherals, low power microcontroller, embedded system laboratory, MPLAB IDE, sensors.

## 1 Introduction

Nowadays, the applications of embedded systems are increasing in diversified fields of electronics, especially in the fields of robotics. This statement is further augmented by the prediction that the application of robotics in industry will under go exponential growth, becoming a 66 billion dollar industry world wide by 2025 [1],[2]. This growth is due to the development in the fields of hardware (VLSI design), microelectromechanical (MEM) and nanoelectromechanical (NEM) sensors, artificial intelligence (AI) and so on, which leads from the miniaturization and development of embedded devices. Further, embedded devices are already accounted for over 98% of the world's processors in 2003 itself [3]. Thus, in future everything of material significance will be outfitted with an embed microcontroller. This demand calls for a trained and skilled work force in the form of budding engineers and scientists in the fields of computer science (CS), electrical engineering (EE) and computer engineering (CmpE). To accomplish this, curriculum framing, teaching and learning programs and laboratory facilities have to be improved at undergraduate and graduate level embedded system courses.

In present scenario, the designer must project the software required by the hardware, allowing it to work as expected [4], [5]. Subjects like operating systems, real time systems, fixed and floating point arithmetic, digital signal processing and programming languages such as assembly, C/C++ or java are of major relevance to the development of embedded systems [6]. Majority of Universities in India offers courses on microcontrollers like PIC, ATMEL, INTEL, MOTOROLA etc. There are several papers dealing with the development of laboratory experiments using different platforms and different microcontrollers [7], [8].

In global scenario, embedded system courses are introduced into existing science and engineering curricula [9], [10]. Experiments on embedded system training are continuously monitored and improved basing on the changes taking place in the architecture of microcontrollers. Papers dealing with development of laboratory exercises using PIC based microcontroller are very less and development of technology with high performance digital signal controller like dsPIC and on software side the operating systems windows 7 can run on dsPIC digital signal controller has been developed. Training the student at the entry level of embedded system course gives a scope to modern teaching practices, international project collaboration, and cooperative learning process [11], [12] and [13].

Keeping these facts in view point, we have developed certain interfacing experiments using the dsPIC30F6014A low power digital signal controller manufactured by microchip. It starts with LED BLINKY experiment and goes up to sensor interfacing. The salient feature of the work is that we have explained all the experiments in stand-alone mode, using digital signal controller trainer board consisting of all interfacing components integrated on the board itself. The set of experiments described in this paper, can also be useful for electronics and to a fresh research student who wants to work in the direction of microcontroller instrumentation.

## 2 ARCHITECTURE DESCRIPTION OF DIGITAL SIGNAL CONTROLLER

The dsPIC30F6014A is a 16-bit digital signal controller with modified Harvard Architecture design for low power applications. This development board was purchased from Robo Electronic, Mumbai [14]. It is a high performance modified RISC processor with the following salient features:

- High performance modified RISC CPU
- Choice of selecting clock frequency
- On chip peripherals: Timers, Analog to Digital Converter, Flash Memory, Digital I/O pins, Watch Dog Timer, 16-bit Capture input functions, 16-bit Compare /PWM output functions, Data Converter Interface

\* Swapna Chintakunta is currently a Research Student working in VLSI and Embedded System Laboratory for Ph.D. degree in Physics, Sri Krishnadevaraya University, Anantapuramu, India, Ph-09491986311.  
E-mail: [swapna.reddy311@gmail.com](mailto:swapna.reddy311@gmail.com)  
• Raghavendra Rao Kanchi, is currently working as Professor of Physics and Chairman, BOS in Physics, VLSI and Embedded System Laboratory, Department of Physics, Sri Krishnadevaraya University, Anantapuramu, Ph-09440204722, E-mail: [kanchiraghavendrarao@gmail.com](mailto:kanchiraghavendrarao@gmail.com), Corresponding Author

(DCI), 3-wire SPI module, I2C module, two addressable UART modules and two Can bus modules.

- The hardware multiplier is useful for digital signal processing.

The internal architecture diagram of dsPIC30F6014A is shown in figure 1, and the 80-pin configuration is shown in figure 2 respectively.

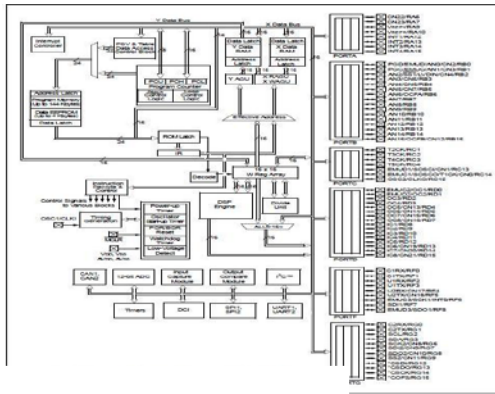


Fig. 1 Architecture of dsPIC30F6014A

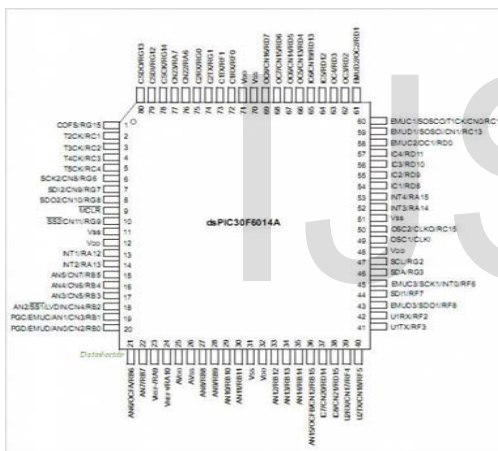


Fig. 2 Pin diagram of dsPIC30F6014A

**2.1 Features of proposed experiments with dsPIC30F6014A**

The following features a series of proposed experiment which is implemented on dsPIC30F6014A.

- Digital I/O Experiments.
- On chip peripherals exploitation experiments namely on chip ADC, UART, SPI, I2C and timers.
- Interfacing experiments like LM 35: analog temperature sensor, DS18B20: digital temperature sensor and Humidity measurement.

**2.2 Programming procedure of dsPIC30F6014A**

In the present work dsPIC30F6014A is programmed using MPLAB C30 compiler. It uses PIC kit-3 programmer to program dsPIC30F6014A's flash memory. MPLAB IDE V8.90 was

used for the software development of dsPIC30F6014A. The compiler for this version is freely available from the Microchip's website [15]. All the experiments described here fits within this limit. Brief description of the programming procedure is explained below, in the form of steps. When the program written for am specific purpose is free from bugs, then the generated hex file is loaded into the flash memory of dsPIC30F6013A using PIC kit-3 programmer. Now the module works on the standalone mode.

The following steps describe the program development and execution process for led blink program as an example. After installation MPLAB IDE software and before following the steps. Click on MPLAB IDE ICON from desktop.

Step 1: Choose configure on popup menu and select the device dialogue to select the dsPIC30F6014A. To select File, click on new file and to type the source code, given the file name with an extension of c, e.g., LED\_BLINK.c, and save it on any drive. The screen shot of this is shown in figure 3.

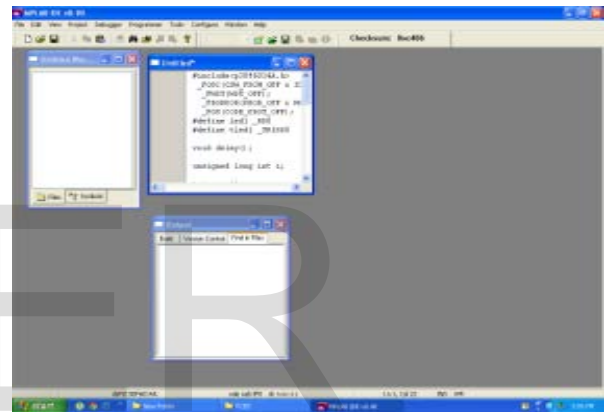


Fig. 3 Saving the source program named as LED\_BLINK.c

Step 2: In this step, to create new project, select the project in the popup menu and select project wizard. To select the language, click the NEXT to continue. Figure 4 shows the new project window dialogue box.

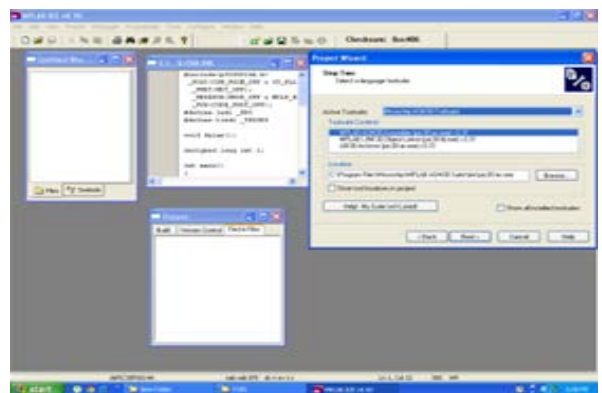


Fig. 4 Creating a new project window

Step 3: In this step, given name of the project, click the button: browse, prescribe the project route the catalogue then click the NEXT button to continue. The screen shot of this is shown in figure 5.

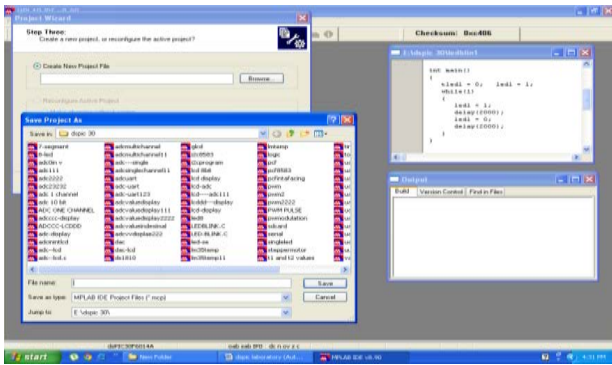


Fig. 5 Screen shot of Step 3

Step 4: In this step, the project wizard allows selecting files for the project and choosing the file name; pressing ADD, adds it into the list file used in the project and also add the link description file p30f6014A.gld; then click the finish. The following figure shows this process.

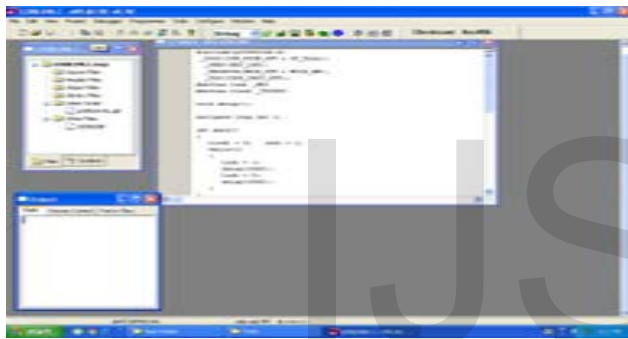


Fig. 6 Screenshot of Step 4

Step 5: In this step, choose the project and click on Build All, compile the source file, check if there is any grammar or writing mistake in the source file; then edit and re-compile by Build all after edit the source file till it succeeds and produces target file (here will produce the ledblink.hex). The final screen shot is shown in figure 7.

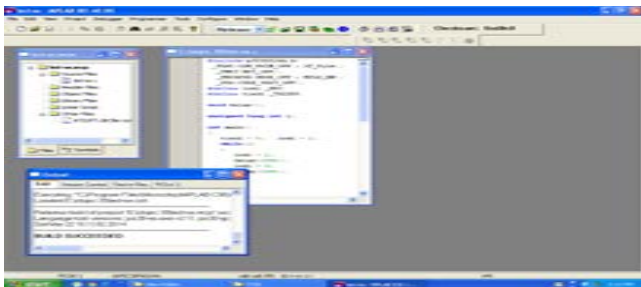


Fig.7 Compiling the project

### 3 DETAILS AND LIST OF EXPERIMENTS

The total experiments developed and described in this paper are divided into three modules: A, B and C. The first module explains the input/output interfacing. Module B corresponds to the experiments for exploitation of the on-chip peripherals.

Sensor interfacing with dsPIC30F6014A is described in Module C. The student uses his theoretical knowledge in assembling the hardware interface circuit and the practical aspect fulfilled by the successful software development and verifying the result by powering up the circuit.

#### Module A: Input/Output Interfacing

##### Ex.A.1: LED blink program

This experiment is generally named as hello program in embedded systems. Components needed for this experiment include: LED, push button, resistors, and connecting wires. In this experiment RD0 pin is used as output pin. The circuit shown in figure 8 is assembled on a bread board.

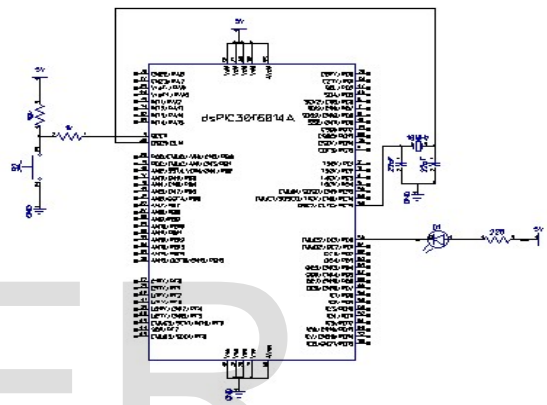


Fig. 8 Schematic Diagram of LED blink

The software is developed basing on the following algorithm:

1. Start
2. Stop the watch dog timer
3. Make RD0 pin as data out direction.
4. Make RD0 high and low with delay.
5. Keeps LED toggling in continuous loop.

The program describing the above algorithm is written in embedded C and the corresponding hex file is loaded on to the flash memory of the dsPIC30F6014A. The photograph of the working hardware assembly is shown in figure 9.

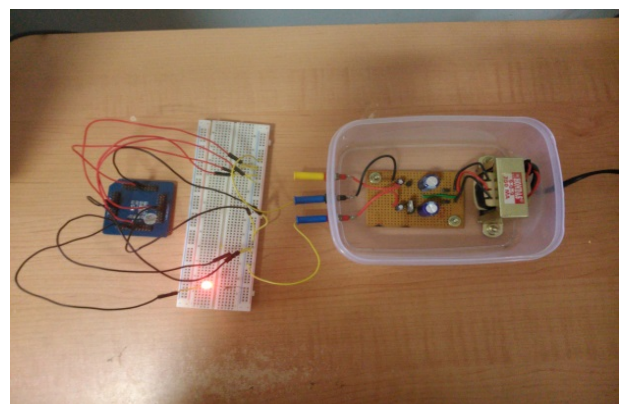


Fig. 9 Photograph of LED blink

Ex.A.2: LED shift

This experiment is useful for the student to understand how logical shift can be done in embedded C using D-register. The components required are LEDs, push button, resistors and connecting wires. Port pins RD0 to RD7 are used to realize the logical shift using LEDs. The circuit is assembled on a bread board as shown in figure 10.

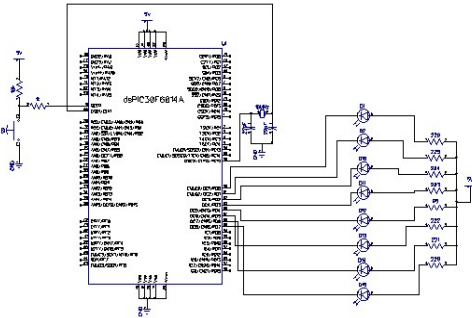


Fig. 10 Schematic Diagram of LED blinks shift

The software is developed basing on the following algorithm:

1. Start.
2. Stop the watch dog timer.
3. Make port pins RD0 to RD7 as data output.
4. Initialize the D-register with 0x00FEh
5. LED blink with shifting can be realized by making the bits RD0 to RD7 of D-register logical shifting. This program is kept in continuous loop.

The program describing the above algorithm is written in embedded C and corresponding hex file is loaded on to the flash memory of the dsPIC30F6014A. The photograph of hardware is shown in figure 11.

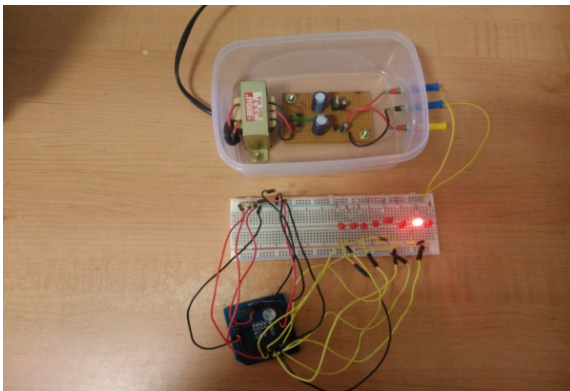


Fig. 11 Photograph of LED blink shift

Ex.A.3: Seven-segment Display

In this experiment the student get better understanding on how the codes of 0-F is stored in the form of an array and sent to 7-segment display connected to RD0-RD7. For this experiment one common anode type display, resistors, push button

and connecting wires. The circuit is assembled on a bread board as shown in figure 12.

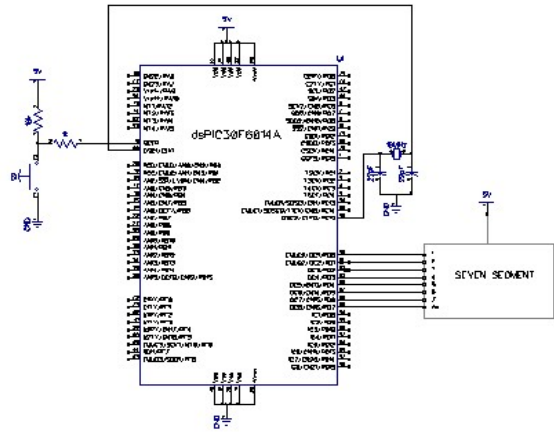


Fig. 12 Schematic Diagram of 7-segment display

The software is developed basing on the following algorithm:

1. Start.
2. Stop watch dog timer.
3. Make port pins RD0 to RD7 as data out.
4. Store the hexadecimal data (Look-Up Table, LUT) to display 0-F on 7-segment display in the form of array.
5. Output the data on port pins RD0 to RD7 continuously.

The program describing the above algorithm is written in embedded C and the corresponding hex file is loaded on the flash memory of dsPIC30F6014A. The photograph of the hardware is shown in figure 13.

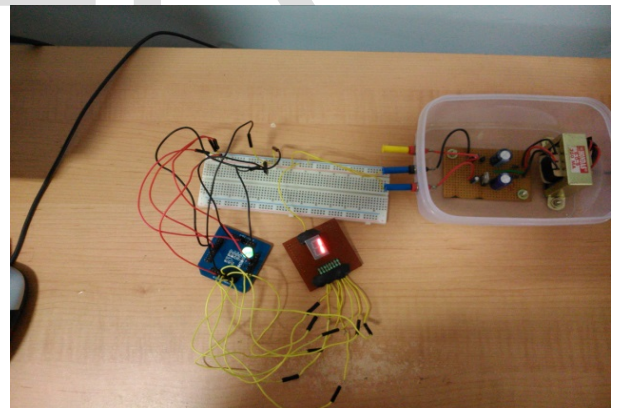


Fig. 13 Photograph of 0-F display on 7-segment display

Ex.A.4: Stepper Motor Interfacing

This experiment gives better understanding on how a mechanical device like a stepper motor can be controlled by an I/O port using Darlington current driver IC ULN 2003. For this experiment, components needed are: stepper motor, IC ULN 2003 and connecting wires. In this experiment RD0 to RD3 port pins are used for interfacing stepper motor via the driver. The circuit is assembled on a bread board and tested as shown in figure 14.

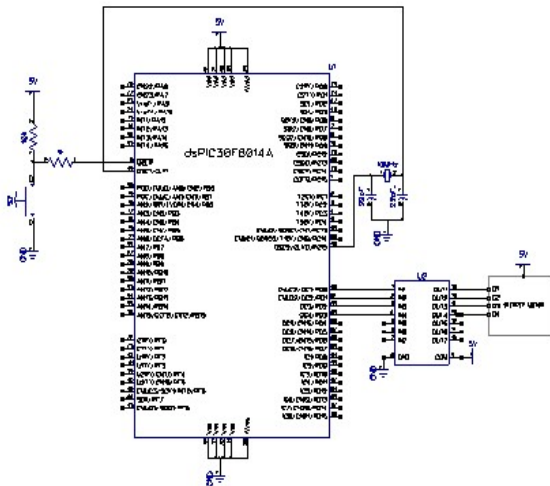


Fig. 14 Schematic Diagram of Stepper Motor Interfacing

The software is developed basing on the following algorithm:

1. Start.
2. Stop the watch dog timer.
3. Define port pins RD0 to RD3 as data out direction.
4. Send data through port pins RD0 to RD3 with delay.
5. Keep on sending the data to rotate the motor shaft by sending appropriate data through port pins RD0 to RD3.
6. Keep the above steps in continuous loop.

Program describing the above algorithm is written in embedded C and corresponding hex file is loaded on to the flash memory of dsPIC30F6014A. The photograph of the hardware tested in the laboratory is shown in figure 15.

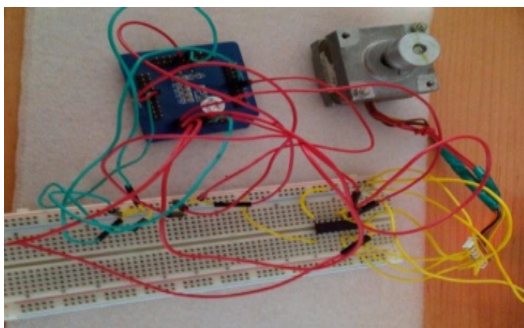


Fig. 15 Photograph of Stepper Motor Interfacing

### Ex.A.5: LCD Interfacing

Several electronic testing and measuring instruments use liquid crystal display (LCD) module for displaying the result. This experiment gives an overall view of interfacing an LCD module with the microcontroller in 8-bit mode. For this experiment components needed are: LCD module, 10 kΩ potentiometer, resistors and connecting wires. The circuit connections in the form of schematic diagram are shown in figure 16.

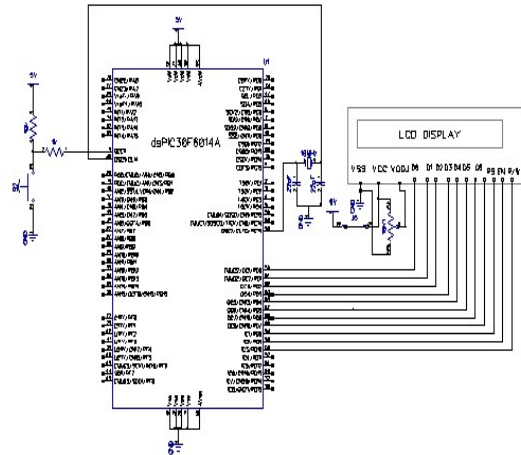


Fig. 16 Schematic Diagram of LCD Interfacing

- The software is developed basing on the following algorithm:
1. Start.
  2. Stop the watch dog timer.
  3. Initialize RD0 to RD7 for data and RB4 to RB6 for control communication with LCD.
  4. Initialize LCD.
  5. Keep on sending the string "hello world" through port pins RD0 to RD7.

The program describing the above algorithm is written in embedded C and the corresponding hex file is loaded on to the flash memory of dsPIC30F6014A. The photograph of the LCD interfacing experiment is shown in figure 17.

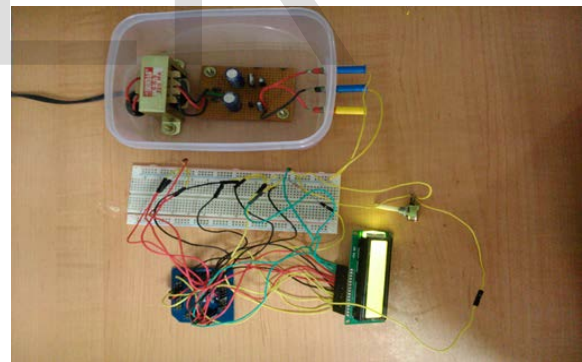


Fig.17 Photograph of LCD Interfacing Module B: Interfacing with PIC18F4550

### Ex.B.1: UART Implementation

Serial communication is one of the easy and efficient protocol for transferring data between two electronic devices. USART is used to accomplish this task. dsPIC has two on-chip UARTs, the clock frequency for the UART is selected using software and baudrate of 9600 is set with the UART's baud rate control registers. Components required: IC MAX 232, DB-9 connector, capacitors and connecting wires. The circuit is assembled as shown in figure 18.

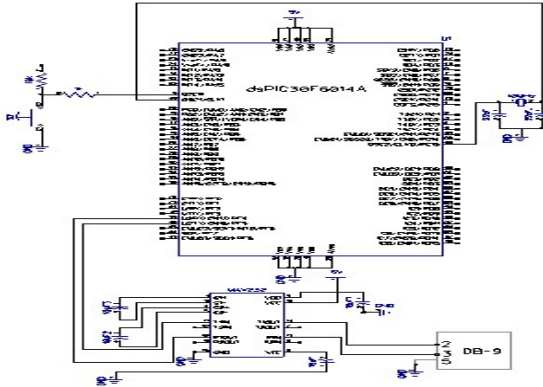


Fig. 18 Schematic Diagram of serial Interfacing using MAX232

Program is developed on dsPIC30F6014A to send the string "hello world" to desktop monitor, using the following algorithm:

1. Start.
2. Stop the watch dog timer.
3. Select RF4 and RF5 as TX2 and RX2 (UART2) pins.
4. Enable the UART2 transmitter and receiver.
5. Select 8-bit data length.
6. Select UCLK as clock source for UART2.
7. Load UART2 baud rate control registers.
8. Send the string "hello world" from dsPIC30F6014A to desktop monitor continuously.

The program describing the above algorithm is written in embedded C and the corresponding hex file is loaded on to the flash memory of dsPIC30F6014A. The photograph of the hardware and string display is shown in figure 19.

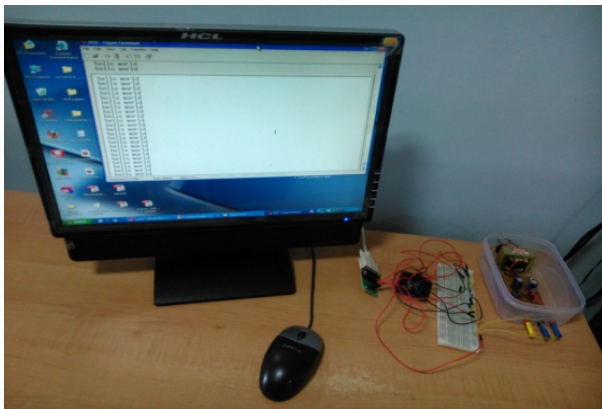


Fig.19 Photograph of "hello world" display on desktop

Ex.B.2: Analog to Digital Converter

As mentioned earlier, dsPIC30F6014A has a 12-bit, 16-channel on-chip ADC. This experiment highlights the measurement of analog voltage using ADC12 of dsPIC30F6014A and displaying its digital equivalent on the hyper terminal. Components needed are: potentiometer, IC MAX 232, capacitors, DB-9 connector and connecting wires. Figure 20 shows the circuit.

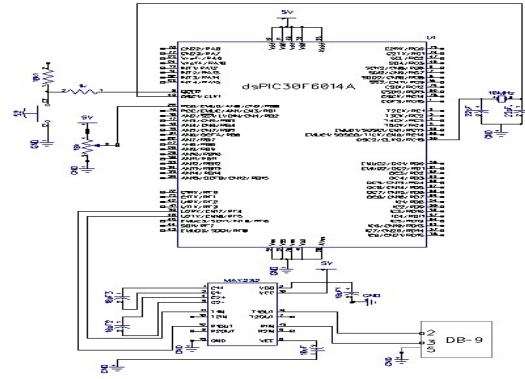


Fig. 20 Schematic Diagram to interface the on-chip ADC12

The software is developed basing on the following algorithm:

1. Start.
2. Stop the watch dog timer.
3. Initialize the port pins RF4 and RF5 for serial communication.
4. Initialize the ADC12 and select the channel 0 for reading external analog input from 10KΩ potentiometer.
5. ADC 12 interrupt is enabled to scan the voltage at the analog input channel and to store it in the buffer, and displaying the same on desktop monitor.

The above program is written embedded C and the corresponding hex file is loaded on dsPIC30F6014A. The photograph of the hardware is shown in figure 21.

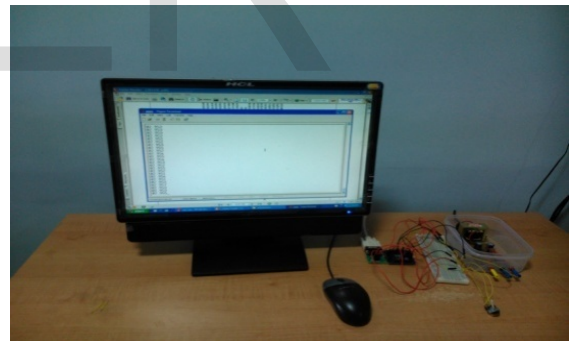


Fig. 21 Photograph of displayed digital output using ADC12

Ex.B.3: Serial Peripheral Interfacing

Serial Peripheral Interfacing (SPI) is a serial bus standard established by Motorola and supported in silicon products from various manufactures. It is a synchronous serial data link that operates in full duplex mode. It is an interface bus commonly used to send data between microcontroller and small peripherals such as converters, sensors, SD cards etc. It uses separate clock (SCK), and single bit Input and single bit output data lines, along with a select line (CS) to choose one among the several devices connected to the common I/O bus. dsPIC30F6014A has three wire SPI modules. This proposed activity will guide the students how to interface an external SD card with dsPIC30F6014A Digital Signal Controller. SD

card uses SPI serial bus and it requires 3.3 volts for its operation. Four lines are required to interface SD card with dsPIC30F6014A Digital Signal Controller using SPI, this four lines are SDIx (Serial data input), SDOx (Serial data output), SCKx(shift clock input or output), SSx (active low slave select). SPI is always controlled by the MASTER. Here dsPIC30F6014A is the master and SD card is a slave on the bus. SPI transfer is initiated by the MASTER pulling the CS line low. The SPIxCON control register is used to control SPI operation, in which SPI is enabled in 8-bit or 16-bit master mode. The 8-bit data or 16-bit data is sent using SPIxBUF register. The program reads the data and stores it on the SD card and displays it on LCD. The circuit assembled as shown in figure 22.

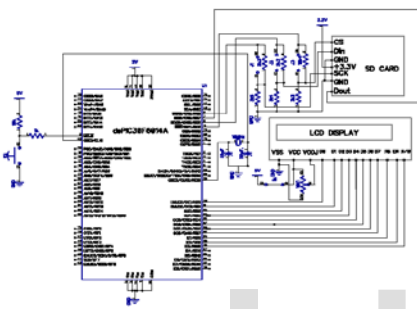


Fig. 22 Schematic Diagram of Serial Peripheral Interfacing

The software is developed based on the following algorithm:

1. Start.
2. Stop the watch dog timer.
3. Initialize the SPI2.
4. Initialize RD0 to RD7 for data and RB4 to RB6 for control communication with LCD.
5. Initialize LCD.
6. Reset SD card.
7. Write block of data to SD card through SPI.
8. Send the data through port pins of RD0 to RD7 for displaying the LCD.

The program describing the above algorithm is written in embedded C and the corresponding hex file is loaded on to the flash memory of dsPIC30F6014A. The photograph of the hardware is shown in figure 23.

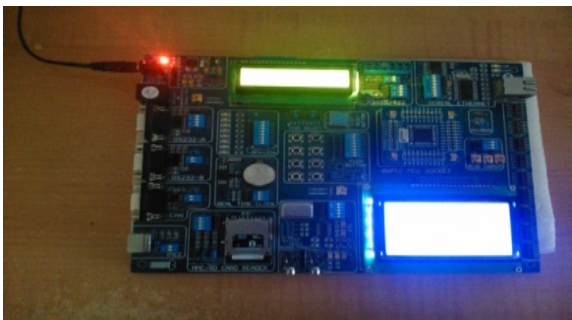


Fig. 23. Photograph of SD card Interfacing

#### Ex.B.4: I<sup>2</sup>C Interfacing

The name I<sup>2</sup>C is stands for Inter Integrated Circuit. It provides good support for communication with various slow, on-board peripheral devices that are accessed intermittently. It is a simple, low-bandwidth, short-distance protocol. I<sup>2</sup>C is easy to use to link multiple devices together since it has a built-in addressing scheme. dsPIC30F6014A consists of one on-chip I<sup>2</sup>C.

The strategy adopted to design the learning platform of I<sup>2</sup>C communication is accomplished by considering the communication between dsPIC30F6014A and IC PCF8583 Real Time Clock [16]. I<sup>2</sup>C is a two line interface (SCL and SDA). The SCL function presented by dsPIC30F6014A is available on RG2 is connected to SCL pin of PCF8583. The SDA line is pulled high using 10KΩ pull up resistors. SDA is available on RG3 of dsPIC30F6014A. This line is connected to SDA line of PCF8583 real time clock. The circuit assembled as shown in figure 24.

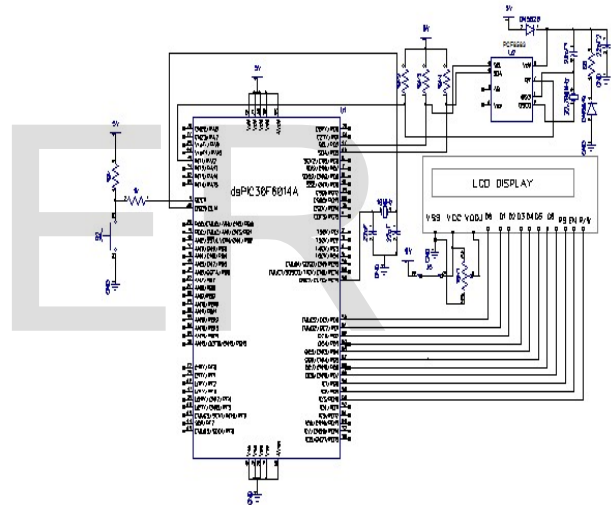


Fig. 24. Schematic Diagram of PCF8583 Interfacing

The software is developed based on the following algorithm:

1. Start.
2. Stop the watch dog timer.
3. Initialize the I<sup>2</sup>C.
4. Initialize RD0 to RD7 for data and RB4 to RB6 for control communication with LCD.
5. Initialize LCD.
6. Read data from PCF8583 through I<sup>2</sup>C.
7. Send the data through port pins of RD0 to RD7 for displaying the LCD.

The program describing the above algorithm is written in embedded C and the corresponding hex file is loaded on to the flash memory of dsPIC30F6014A. The photograph of the hardware is shown in figure 25.



Fig. 25 Photograph of PCF8583 Interfacing

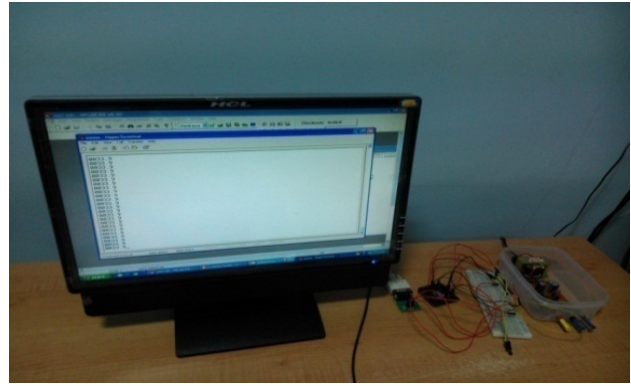


Fig. 27 Photograph of LM 35 Interfacing

Module C: Interfacing with Sensor

Ex.C.1: LM 35 interfacing

This experiment explains the programming method for reading LM 35 Temperature sensor. The components required are LM 35 temperature sensor [17], capacitors, IC MAX 232, DB-9 connector and connecting wires. The circuit is as shown in figure 26.

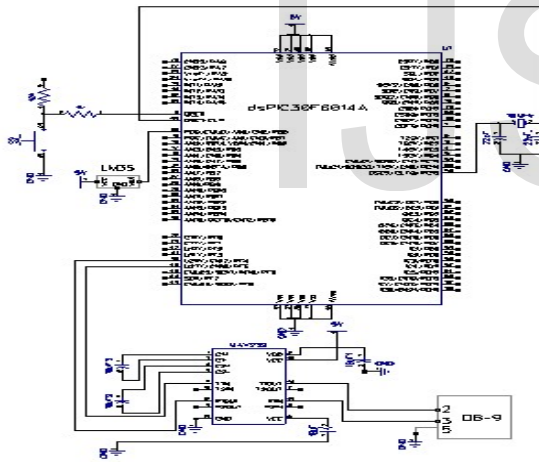


Fig. 26 Hardware schematic for LM 35 Interfacing

The software is developed basing on the following algorithm:

1. Start.
2. Stop watch dog timer.
3. Initialize the RF4 and RF5 for serial communication.
4. Initilize the ADC12 register for reading analog input from LM 35.
5. Convert the analog data into temperature and display on desktop monitor.

The program describing the above algorithm is written in embedded C language and the corresponding hex file is loaded on to the flash memory of the dsPIC30F6014A. The photograph of the hardware is shown in figure 27.

Ex.C.2: Humadity Measurement System

This experiement explains the programming method for reading HSM-20G sensor [18]. The components required for this experiment are HSM-20G, resistor, capacitors, IC MAX232, DB-9 connector and connecting wires. The circuit is as shown in figure 28.

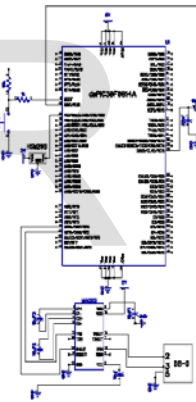


Fig. 28 Hardware schematic for HSM-20G interfacing

The software is developed basing on the following algorithm.

1. Start.
2. Stop the watch dog timer.
3. Initialize the RF4 and RF5 port pins for
4. Initialize the ADC12 register for reading analog input from HSM-20G.
5. Convert the analog input data into humadity and display on HyperTdrminal window.

The describing the above algorithm is written in embedded C language and the corresponding hex file is loaded on to the flash memory of dsPIC30F6014A.

Ex.C.3: Interfacing Digital Thermometer

This experiment explains the programming method for reading digital temperature sensor. The components required DS18B20 digital thermometer [19], capacitors, IC MAX 232,



DB-9 connector and connecting wires. The circuit is shown in figure 29.

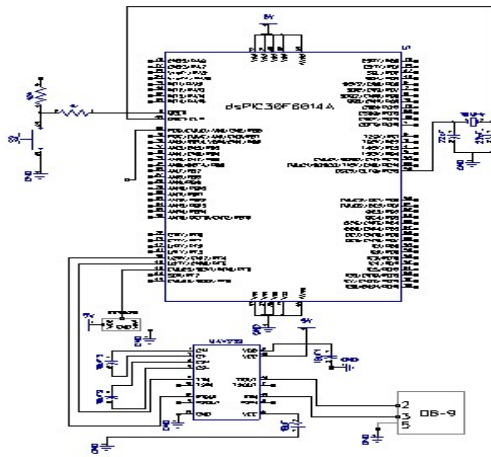


Fig. 29 Schematic Diagram of Digital Thermometer

The software is developed on the following algorithm

1. Start.
2. Stop the watch dog timer.
3. Initialize the RF4 and RF5 for serial communication.
4. Initialize the RF6 port pin as data out direction.
5. Digital temperature data display on desktop monitor.

The describing the above algorithm is written in embedded C and the corresponding hex file is loaded on to the flash memory of dsPIC30F6014A daughter board. The photograph of the hardware is shown in figure 30.

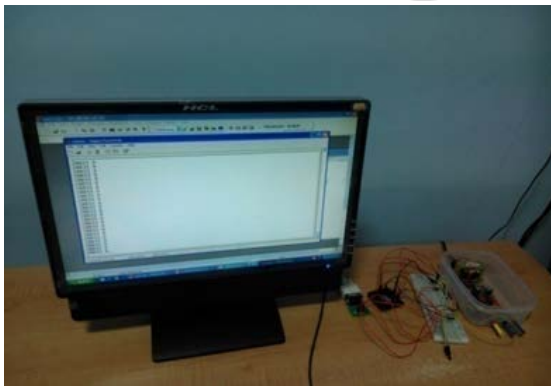


Fig. 30 Photograph of the DS18B20 Interfacing

#### 4 CONCLUSION

In this paper, we have described a bunch of experiments with dsPIC30F6014A Digital Signal Controller. These experiments are grouped into three, namely the first group describing the input/output interfacing, the second group describing the on-chip peripherals and the third group explaining the sensor interfacing. These series of experiments are useful for undergraduate ECE/CSE laboratory. The in-house development of hardware interfaces provide an inexpensive, easy to design

interfacing. The hands-on experience the student gets will make him to design and develop complex embedded systems using dsPIC30F6014A Digital Signal Controller.

#### ACKNOWLEDGMENT

C. Swapna is thankful to University Grants Commission (U.G.C.), New Delhi, for sanctioning Junior Research Fellowship (B.S.R.)

#### REFERENCES

- [1] <http://nextbigfuture.com/2011/10/robotics-industry-is-now-on-exponential.html>
- [2] Japan Robotics Association, Tokyo, Japan, "The State of the industry worldwide robotics market growth" June 2011.
- [3] J. Turley, "The two percent solution", embedded system Program, Vol-16, no.1, p.29, Jan 2003.
- [4] Aruna Kommu, Raghavendra Rao Kanchi, "Design and Development of a Low-Cost Embedded System Laboratory Using LPC1768", American Journal of Embedded Systems and Applications, Vol.1, no.2, pp. 46-53, 2013.
- [5] Naveen Kumar Uttarkar, Raghavendra Rao Kanchi, "Design and Development of a Low-Cost Embedded System Laboratory Using TI MSP430 Launchpad", American Journal of Embedded Systems and Applications, Vol.1, no.2, pp. 37-45, 2013.
- [6] Tiago Goncalves, A. Espirito-Santo, B.J.F. Ribeiro, P.D.Gasper, "Design and Learning Environment for Embedded Systems", Proceedings of the World Congress on Engineering 2010, Vol.1, WCE 2010, June 30 - July 2, 2010, London, U.K.
- [7] Yao Li "Teaching Embedded Systems Using a Modular Approach Microcontroller Training Kit", World Transactions on Engineering and Technology Education, Vol. 6, Vol.6, pp.135-138, 2013.
- [8] Feisel. L and Rosa.A, "The Role of the Laboratory in Under Graduate Engineering Education", Journal of Engineering Education 94, 2005.
- [9] Crespo.A, Vila.J, Blanes.A and Rippoll.I, "Real Time Education in a Control Engineering Curriculum", in 3<sup>rd</sup> IEEE Real Time System Education Workshop, pp.112-116, 1998.
- [10] 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.
- [11] Davcec, B. Stojkosa, S. Kalajdziski and K. Trivodaliev, "Project based learning of Embedded Systems", in proceedinf of the 2<sup>nd</sup> WSEAS Telecommunication, 2008.
- [12] S. Nooshabadi and J. Garside, "Modernization of teaching in Embedded Systems Design. An international Collaborative project", IEEE Transactions on Education, Vol.47, no.1, pp.83-92, Feb., 2004.
- [13] J.W.Bruce, J.C. Harden and R.B. Reese, "Cooperative and progressive design experience for embedded systems", IEEE Transactions Education, Vol.47, No.1, PP.83-91, Feb 2004.
- [14] [www.vegarobokit.com](http://www.vegarobokit.com)
- [15] [www.microchip.com](http://www.microchip.com)
- [16] [pdf.datasheetcatalogue.com/datasheet/NXP\\_semiconductor/PCF8583.pdf](http://pdf.datasheetcatalogue.com/datasheet/NXP_semiconductor/PCF8583.pdf).
- [17] [www.ti.com/lit/ds/symlink/lm35.pdf](http://www.ti.com/lit/ds/symlink/lm35.pdf).
- [18] [www.cytron.com.my/ds/sensor/Humidity\\_userManualy1.pdf](http://www.cytron.com.my/ds/sensor/Humidity_userManualy1.pdf).
- [19] <http://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>