AC 2011-2307: DESIGN PROJECTS FOR PROGRAMMABLE EMBEDDED SYSTEM-ON-CHIP COURSE

Prawat Nagvajara, Electrical and Computer Engineering Drexel University

Prawat Nagvajara received his Ph.D. degree from Boston University, Boston, Massachusetts, in 1989, MS and BS degrees from Northeastern University, Boston, Massachusetts, in 1979 and 1980, respectively. He joined the Department of Electrical and Computer Engineering at Drexel University in 1990 where he is currently an associate professor. His research focus is on application specific computing using custom hardware and commodity high-performance accelerators and embedded systems education.

Robin Kizirian, Electrical and Computer Engineering Drexel University

Robin Kizirian completed his M.S. degree in Computer Engineering at Drexel University in Philadelphia and his B.S. degree in Computer Engineering with a minor in Computer Science at the University of Hartford in Connecticut. He is currently serving as a research assistant at the Engineering Technology department at Drexel University. Robin has been involved in various projects funded by Pfizer, NASA, NSF and Department of Education. His areas of research include Embedded Systems, Mechatronics, Efficient Solar Energy Systems, Internet-based Quality Control and 3-D Online Education.

Design Projects for Programmable Embedded System-On-Chip Course

Abstract

The integration of microcontrollers with programmable electronic devices and design automation software tools has greatly improved today's embedded system design. Most recently, there has been a trend to integrate microcontrollers with programmable digital and analog peripherals on a single chip. This technology is known as programmable embedded system-on-chip. At the author's institution, programmable system-on-chip (PSoC) devices such as the Cypress PSoC boards have been adopted in embedded system design courses. This paper presents design projects for embedded system learning purposes. Programmable system-on-chip, with its design software, offers a hands-on approach that aids students in understanding the concepts behind each of the building blocks, providing them with the sufficient tools and experience to develop real-world applications. This paper summarizes three projects that have been developed and taught as the assignments in embedded systems course. Each of the projects involves real-time software concepts, e.g. interrupts, interfacing with external sensors and actuators, and digital and analog hardware theories. The project materials will be disseminated on the authors' institution website for future use by other institutions.

I. Introduction

The study of embedded system design with microcontrollers is relevant to today's automatically controlled electronic devices. A microcontroller is an integrated circuit comprises of a microprocessor, memory, and other input and output peripherals. It offers many features such as interrupts and General-Purpose Input/Output interfaces (GPIO). The interrupt feature provides the chip with real-time capabilities, including the capability of providing a proper response to a specific event. The GPIO feature allows microcontroller to interface with other electronic devices. It is common to incorporate microcontrollers together with programmable digital and analog peripherals on a single chip[1,2].

One new trend in embedded designs is to integrate microcontrollers with programmable peripherals on the chip[3,4]. This technology is known as programmable embedded system-on-chip (PSoC). The Cypress PSoC device is an example of a programmable system-on-chip device. It includes a microprocessor and mixed-signal arrays of configurable integrated analog and digital peripherals[5,6] which offers a practical embedded systems learning tool. The PSoC boards have been adopted in embedded system design courses at the authors' institution.

This paper presents a number of microcontroller-based design projects for embedded system learning purposes. These projects are implemented using the Cypress PSoC boards[11]. PSoC technology provides students with a hands-on approach in understanding the concepts behind each of the building blocks when developing real-world applications[7-9]. This paper presents three projects: Pulse Width Modulation (PWM) fan control, gravity measurement, and a wireless traffic light control system. The projects have been taught as part of an embedded systems course and their materials will be disseminated on the authors’ institution website for future use by other institutions. The projects cover real-time software concepts such as interrupts, interfacing with external sensors and actuators, and digital and analog hardware theories. These projects can better prepare students for the industry, emphasize the impact of software on the controller’s architecture, and introduce the use of networking as an engineering tool.

II. Programmable System-on-Chip Technology Overview

Programmable System-on-Chip (PSoC) is an integrated circuit whose architecture consists of a CPU core, configurable analog and digital blocks, and programmable routing and interconnects. The core includes multiple core options, choice of internal or external oscillator, flash memory, SRAM, sleep and watchdog timers, and multiple clock sources. The number of digital and analog blocks varies from one device to another, but is a major factor in differentiating the PSoC from a regular microcontroller. Having the ability to configure analog and digital circuitry on the same chip provides the engineer with the flexibility of integrating both logic resources with complex analog flows. At the heart of the analog blocks is a collection of switch capacitors, op-amps, comparators, ADC, DAC, and digital filter blocks. Re-routing between the I/O pins also offers the user more logic freedom and simplification as the global buses allow for signal multiplexing and logic operations[6].

The projects in this paper are based on the CY8C2xxxx family of PSoC1 devices. These devices can be programmed by using the PSoC Designer software that can be downloaded for free from the Cypress website[11]. The software simplifies the coding process by allowing the addition or removal of components by dragging and dropping the desired digital or analog peripherals. Each of the building blocks is fully explained and documented in separate PDF documents. After creating the necessary blocks, the software automatically builds the API libraries to access and control these blocks. The processor is programmed by using the API libraries and C Language. The boards that are used in this paper are shown in Figure 1. Figure 1(a) displays the PSoC1 Evaluation board while Figure 1(b) shows the wireless PSoC1 Low Power RF node. Both boards are Cypress Semiconductor products.

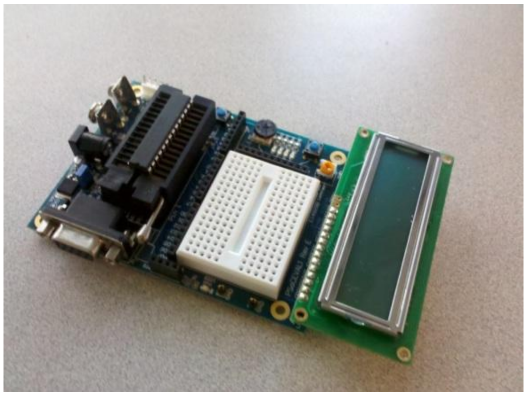


Figure 1(a): PSoC1 Evaluation Kit

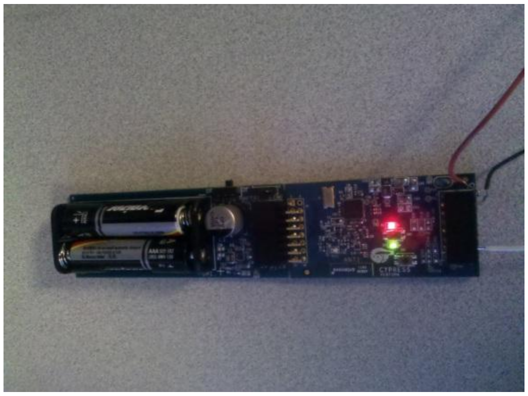


Figure 1(b): PSoC CYFI Low Power RF Node

III. Pulse Width Modulation Fan Control Project

An instructional application of embedded systems is designing an efficient fan-based temperature controller. Van Ess et. al presented this design project in their laboratory manual[5]. The relevance of the project can be justified by the sensitivity of electronic devices to temperature. Most of these devices contain fans which have to be controlled to keep the electronic environment at a stable temperature. This represents a practical application that can familiarize students with different engineering theories and concepts such as PWM and frequency measurement techniques.

The system design is shown in Figure 2(a). The fan is equipped with an integral tachometer which allows the monitoring and controlling of the fan's speed. The controller utilizes a simple pulse width modulator to change the width of the pulse provided to the fan and thus, allowing to change its speed. Although the fan's speed is calculated by measuring the frequency from the tachometer, this measurement is only possible when the fan is fully driven. For this purpose, a tachometer overriding state machine has been programmed into the PWM interrupt handler to generate an override signal and calculate the speed appropriately. In addition to this setup, an integrating speed controller has been added to capture the fan's speed and adjust the PWM duty cycle to drive the fan with a constant speed. This concept teaches students about closed feedback loop systems and shows them their significance in adapting to sensitive variations in a system.

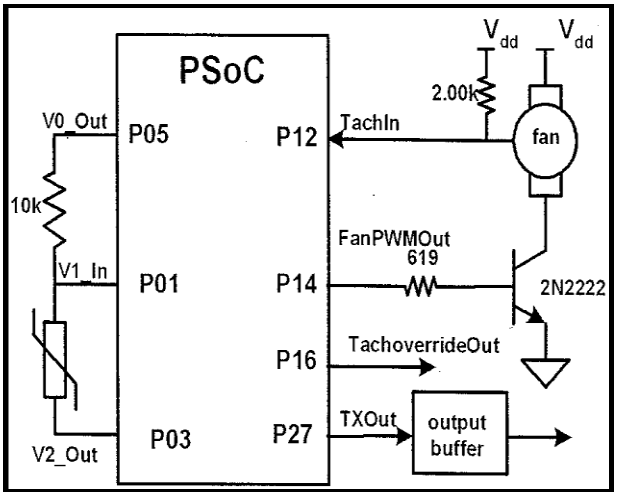


Figure 2(a): Schematic for Fan Project[5]

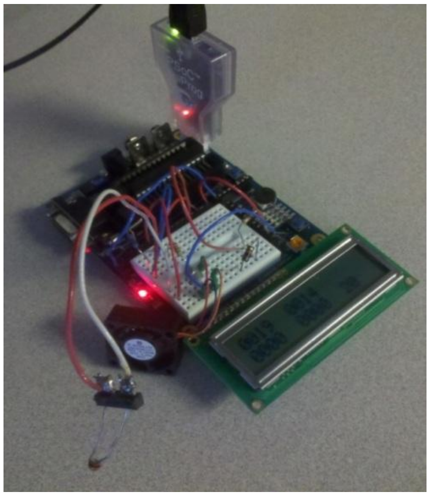


Figure 2(b): The Implemented Project

The temperature measurement is performed by using a thermistor. Thermistors provide resistance values as a function of the measured temperature. By calculating their resistance, an accurate measurement of the temperature can be obtained. This relationship between the resistance and the temperature is described by Equation 1: The Steinhart-Hart equation[5].

$$\frac{1}{T} = A + B\ln R + C\ln R^3 \tag{1}$$

where **T** is the temperature in Kelvin
R is the resistance in ohms
A, **B**, and **C** are the constants

By using the Steinhart-Hart equation, the necessary look-up thermistor table is created. As for the resistance calculation, it can be performed in Figure 2(a) by Equation 2.

$$\boldsymbol{R_{therm} = 10K \times \frac{V1\_In - V2\_Out}{V0\_Out - V1\_In}} \qquad (2)$$

Comparing the measured resistance to the look-up table, it is possible to find out the temperature. Once the temperature is determined, the controller can use the information to modify the fan's speed accordingly to keep the system cool. The project is implemented on the PSoC1 Evaluation Board and is displayed in Figure 2(b).

IV. Gravity Measurement Project

Real-time computing is an essential phenomenon in the study of embedded systems. A common technique to address this issue is by using interrupts. An interrupt-driven system allows an asynchronous signal to pause the processor, save its execution state, and then execute the code in the interrupt handler. Once the interrupt has been handled, the saved state of execution is recovered and the program pointer returns to where the processor had been stalled. The gravity measurement project serves as a useful tool to teach about the concept of requesting interrupts and handling them. In PSoC1, an interrupt has to be declared and coded in C language by using the #pragma interrupt_handler directive. This statement will order the compiler to save and recover the value of the registers, and return from the interrupt subroutine while the execution has ended. In the boot file, the Interrupt Vector Table has to also be edited so that it jumps to the required handler once the interrupt has been requested.

The project has been setup as in Figure 3(a). A number of Infrared digital sensors, as in Figure 3(b), have been placed in a vertical fashion on a wall. The sensors have been fixed equidistantly at one foot from each other. They are connected to the general purpose ports on the PSoC board. A round object is dropped at free fall from the top. Once the ball is dropped, each of the sensors detects the ball and signals the controller as a general purpose input/output interrupt request to keep track of the elapsed times at which the ball crossed the sensor. The elapsed times are gathered within the interrupt handler and later used for calculating the gravitational acceleration of the ball. The calculations are conducted by utilizing linear regression methodologies and finding the best linear fit of a Velocity v.s. Time graph.

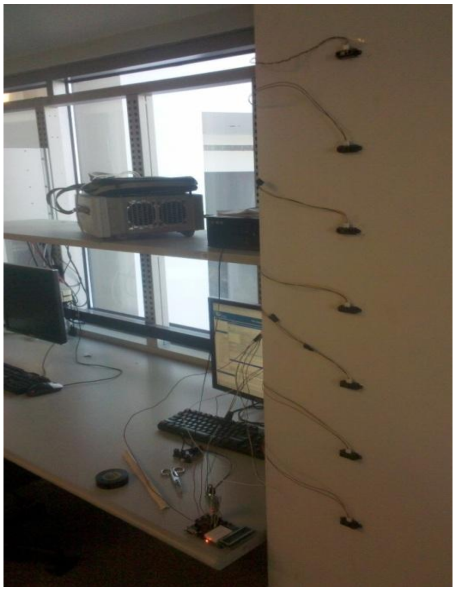


Figure 3(a): Gravity Measurement Project Setup

Figure 3(b): IR Digital Sensor

Figure 4 represents the pseudo code for the controller algorithm. The elapsed times are captured and calculated in the interrupt handling section. The elapsed times are stored in an array whose pointer is incremented at every signal from a sensor. Once all the sensors have triggered the interrupt, the Boolean that signifies the completion of the collection process becomes true. At this point, the post-data-collection processing takes place. The velocities corresponding to the elapsed times are calculated and the linear regression technique is applied. The calculated slope of the regression line is the value of the gravitation acceleration. The value for gravity is finally displayed on the LCD screen that is provided by the PSoC board.

```
//Initiation
Distance = Distance between sensors
Collection_Done = False
Start Timer

//GPIO Interrupt Handling and Processing
Read Timer Value
Store Timer Value in Array
Increment Pointer

If Pointer = # of Sensors
      Collection_Done = True

//Post-Data-Collection Processing
If Collection_Done = True
      Calculate Elapsed times from Array
      Calculate Velocities corresponding to Elapsed Times
      Apply Linear Regression Technique
      Gravity = Slope

Display Gravity Measurement on LCD Screen
```

Figure 4: Controller Algorithm to Calculate the Average Gravity

The linear regression methodology[10] used is described in Equations 3, 4, and 5.

$$\boldsymbol{m} = \frac{\boldsymbol{n}\Sigma(\boldsymbol{xy}) - \Sigma\boldsymbol{x} \cdot \Sigma\boldsymbol{y}}{\boldsymbol{n}\Sigma(\mathbf{x}^2) - (\Sigma\mathbf{x})^2} \tag{3}$$

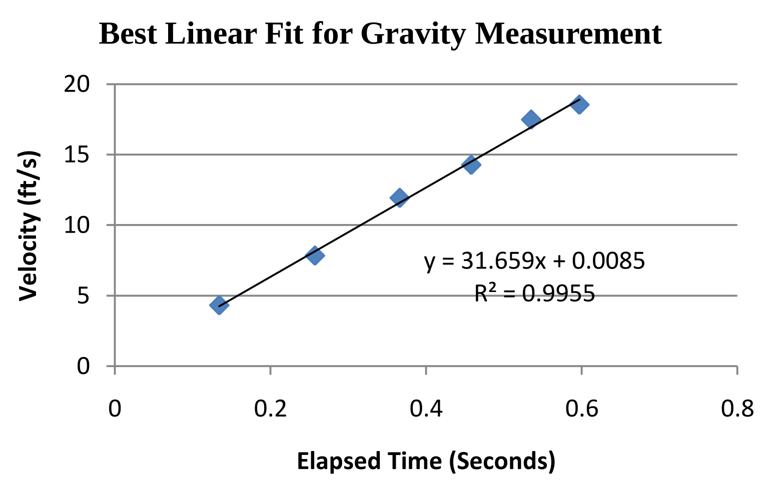
$$\boldsymbol{b} = \frac{\Sigma\boldsymbol{y} - \boldsymbol{m}\Sigma\boldsymbol{x}}{\boldsymbol{n}} \tag{4}$$

$$\boldsymbol{r} = \frac{\boldsymbol{n}\Sigma(\boldsymbol{xy}) - \Sigma\boldsymbol{x} \cdot \Sigma\boldsymbol{y}}{\sqrt{[\boldsymbol{n}\Sigma(\mathbf{x}^2) - (\Sigma\mathbf{x})^2] \cdot [\boldsymbol{n}\Sigma(\mathbf{y}^2) - (\Sigma\mathbf{y})^2]}} \tag{5}$$

where **x** and **y** are the variables
n is the number of values
m is the slope of the regression line
b is the intercept point of the regression line
r is the correlation coefficient

Table 1 displays sample gravity measurements data and the corresponding best linear-fit graph. As can be seen, the slope of the line is 31.659 ft/s^2 which is approximately equal to the average gravitational acceleration of 32.2 ft/s^2.

Table 1: Sample Gravity Calculation Data

Elapsed Time (seconds)	Velocity (ft/s)	
0.134	4.3148	
0.257	7.8345	
0.366	11.9256	
0.458	14.2693	
0.535	17.4754	
0.597	18.5344	

V. Wireless Traffic-Light Control Project

The main project in the embedded systems course involves designing a wireless traffic-light controller. This project presents the concepts of network protocols and topologies and their usage as engineering tools. It combines these networking tools with microcontroller technologies to develop an integrated wireless embedded design. Using PSoC devices for wireless embedded systems provides the advantage of fewer components and shorter design cycles. PSoC utilizes the CyFi Star Network Protocol[11] as the basis for its wireless system. The protocol is portrayed in Figure 5. In the star network topology, packets are transmitted back and forth between the nodes and the centralized hub. These communications are bidirectional and take place as from node to hub or hub to node. Up to 250 nodes can be connected to one hub. The nodes and the hub are implemented using wireless PSoC1 devices and are powered using two AAA batteries.

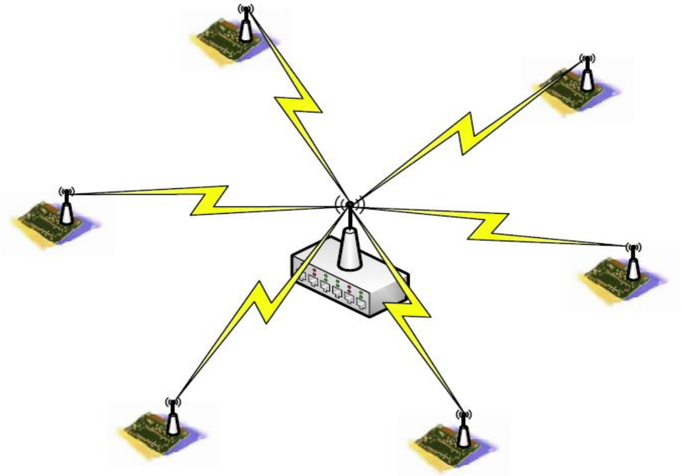


Figure 5: Wireless Star Network Topology

In each of the wireless components, the PSoC device executes the network protocol, runs the programmed application, and drives the CyFi transceiver. The CyFi transceiver features a 2.4 GHz RF (Radio Frequency) solution and is controlled by an active power management system for low power communications among the wireless components. Figure 6 displays the proposed system design for the wireless traffic light controller. The system consists of two nodes and a hub. On one side, the sensor node interfaces with analog proximity sensors, collects traffic information, and then transmits it to the hub. The hub correspondingly forwards the information to the traffic light controller node. The traffic light controller node sends a signal to a PSoC1 traffic light controller board to either maintain its state or change it accordingly.

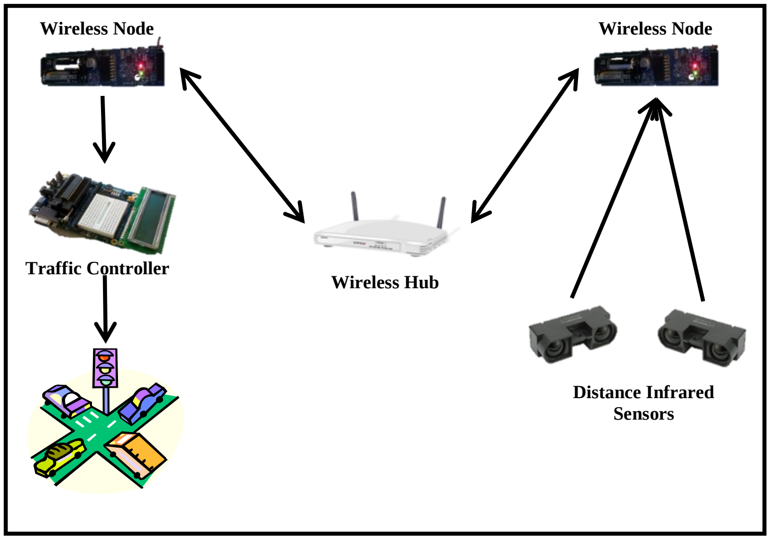


Figure 6: The Proposed System for Controlling Traffic Lights Wirelessly

The idea behind using a traffic controller board is to have enough power to drive a relay system to light up the traffic lights. The PSoC1 Evaluation board can satisfy the necessary 5V operating voltage for the relays whose coils are connected to a 12-volt power supply. For safety purposes, the installed traffic light bulbs operate at 12 volts and utilize a low current of approximately 500mA. Inside the controller, a state machine as in Figure 7 has been programmed to change the state of the lights according to the signal from controller node. At any of the states except for Green/Red, the state is maintained until an expired timer signal has been received. When the state is Green/Red, the state is not changed until an expired timer signal is received and traffic information indicates that a car is waiting.

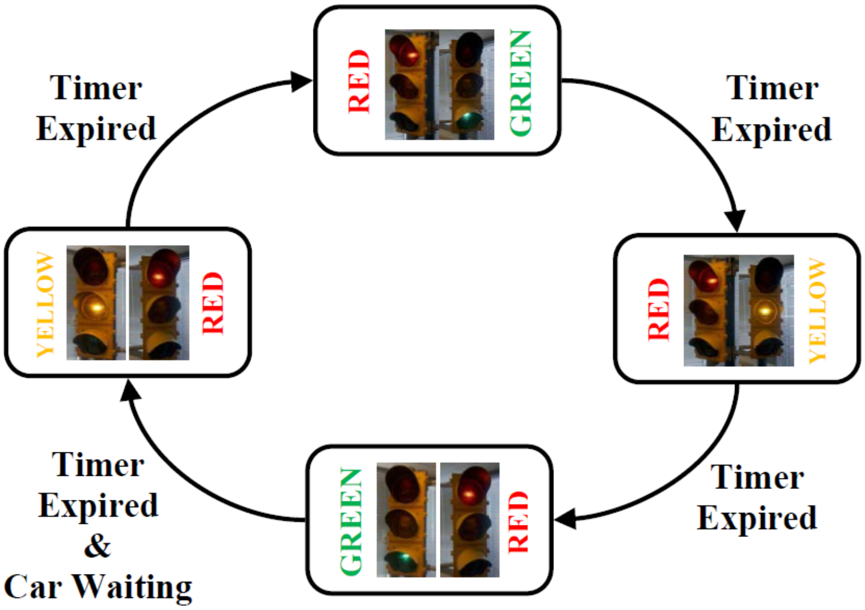


Figure 7: Traffic Light Controller State Machine

Cypress has included the Application Programming Interface (API) for its CyFi topology to simplify the control of its Star Network Protocol. Before packets can be transmitted among the nodes, each of the nodes has to be bound with the hub. The binding process is started by the hub when it initiates its listening process to any requesting nodes. The nodes then connect to the hub by requesting to be bound and registering their device IDs. This means that the hub stores all the nodes' device IDs; however, the nodes do not know each other's IDs. After the nodes are bound, the network has been established and data can be transferred. Figure 8 demonstrates the different stages through which a packet traverses when it is transmitted from the sensor node (Node 1) to the control node (Node 2). At the sensor node, the packet is programmed to include the control node's device ID in its payload. The packet also contains the traffic information data that the sensor node is planning to send to the control node. Since CyFi utilizes a star network protocol, the packet has to be sent to the hub initially before it is forwarded to its destination. Thus, the header information indicates that the sender is the sensor node and the receiver is the hub. When the packet arrives to the hub, the hub determines the packet's destination from its payload. It creates a new packet which includes the sender's device ID and a copy of the data in its payload. At this stage, the header information indicates that the sender is the hub and the receiver is the control node. Once the packet arrives at the control node, the node identifies that the sender is the sensor node and then analyzes the data that it has received.

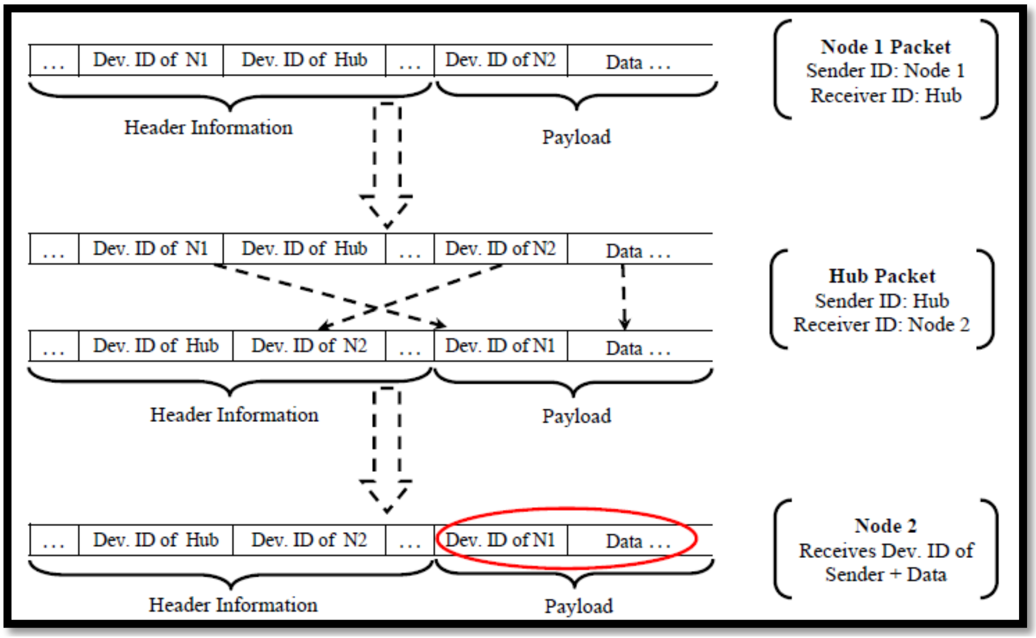


Figure 8: Packet Transmission from Node 1 to Node 2

The overall system has been setup in a laboratory environment at the author’s institution. Figure 9 displays the implemented wireless traffic light controller and highlights the different sections of the system.

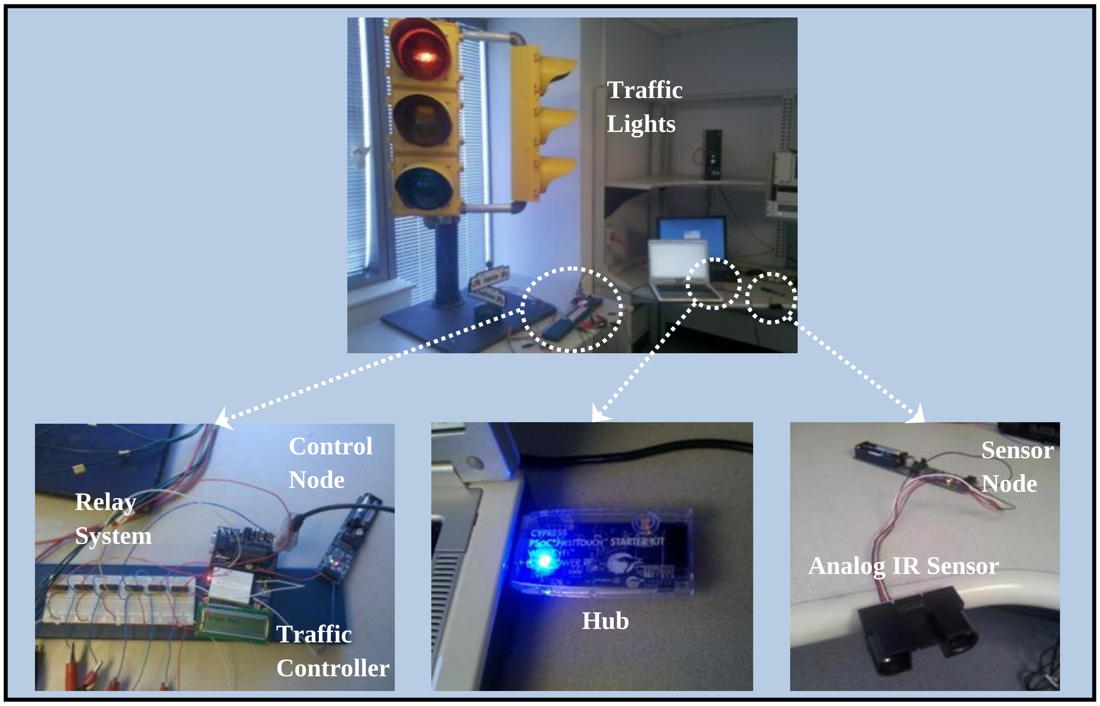


Figure 9: The Implemented Wireless Traffic Light Controller

VI. Evaluations

The students in the embedded systems course were presented with pre-course evaluations to determine their strengths and weaknesses prior to performing the laboratory experiments that accompany the course. After being taught the programmable system-on-chip technology and getting involved in the projects, the students were asked to complete the same course evaluation questionnaire.

The questionnaire targets the students' understanding of the course materials and measures the enhancement in the learning of embedded systems concepts and applications. The evaluation questionnaire is the same for pre- and post-evaluations and is displayed in Figure 10.

The pre- and post-course evaluation results are presented in Figures 11 and 12. The final results demonstrate a marked increase in the students' understanding of microcontroller-based technology and embedded systems. By using programmable system-on-chip devices, the students gained experience in designing systems composed of both hardware and software components. Their understanding about sensors and actuators in addition to their knowledge in computer architecture, interrupts, and mixed signal systems drastically improved.

Survey Question	**1** Least like me	**2** Not like me	**3** Neutral	**4** Like me	**5** Most Like me
1. I have experience working with software program interfacing with hardware.	**O**	**O**	**O**	**O**	**O**
2. I understand how embedded systems and microcontrollers are applied to engineering problems.	**O**	**O**	**O**	**O**	**O**
3. I have experience designing systems that have hardware and software components.	**O**	**O**	**O**	**O**	**O**
4. I have used design automation tool that integrates programmable hardware and software development.	**O**	**O**	**O**	**O**	**O**
5. I understand how software programs interface with sensors and actuators.	**O**	**O**	**O**	**O**	**O**
6. I have knowledge in computer architecture and organization, and assembly language.	**O**	**O**	**O**	**O**	**O**
7. I have used interrupts and interrupt service routines in real-time software.	**O**	**O**	**O**	**O**	**O**
8. I have worked with mixed-signal systems (digital and analog).	**O**	**O**	**O**	**O**	**O**
9. I design embedded systems with design stages and debugging plan.	**O**	**O**	**O**	**O**	**O**
10. I am prepared for career in embedded systems design.	**O**	**O**	**O**	**O**	**O**

Figure 10: Pre and Post Course Evaluation Questionnaires

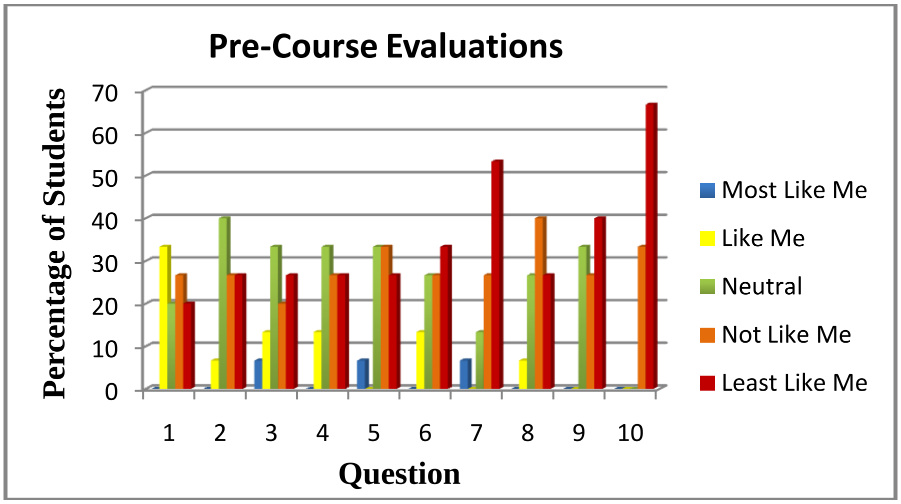


Figure 11: Results from the Pre-Course Evaluations

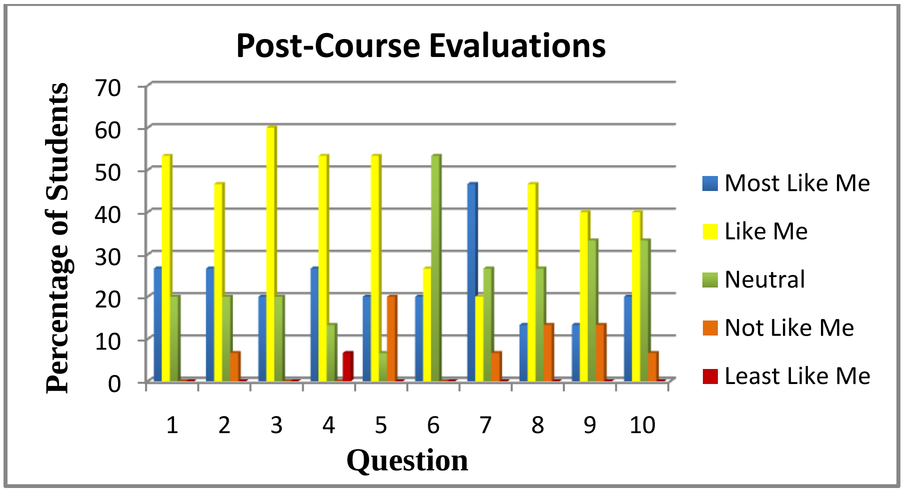


Figure 12: Results from the Post-Course Evaluations

VII. Conclusion

Enhancing microcontroller teaching techniques is important for insuring quality and up-to-date education for future generations of electrical and computer engineers. The paper proposes the integration of programmable system-on-chip technologies into microcontroller learning. Programmable system-on-chip devices, such as the Cypress PSoC boards, have been utilized in the authors' institution for the laboratory experiments of an embedded systems course. These devices offer programmable digital and analog blocks that introduce the students to the different digital and analog design criteria and provide them with a better hands-on industry-like experience. Three main projects were incorporated into the course, each of which targeted a specific field of embedded systems learning. Based on an evaluation questionnaire completed by each student at the beginning and end of the course, significant improvement in student learning (and understanding of microcontroller-based concepts and their integration into embedded systems) was noted. Integrating PSoC devices into microcontroller coursework appears to be of great benefit in electrical and computer engineering education.

References

[1] Mar, M., Sullam, B., and Blom, E., "An Architecture for a Configurable Mixed-Signal Device", *IEEE Journal of Solid-State Circuits,* Vol. 38, pp. 565-568, March 2003.

[2] Fang, W., Kedar, S., Owen, S., Wei, G., and Brooks, D., "System-on-Chip Architecture for Intelligent Sensor Networks," *Proceedings of the 2006 International Conference on Intelligent Information Hiding and Multimedia Signal Processing(IIH-MSP'06),* Pasadena, CA, December 18-20, 2006.

[3] Faura, J., Horton, C., Krah, B., Cabestany, J., Aguirre, M.A., and Insenser, J.M., "A New Field Programmable System-on-a-Chip for Mixed Signal Integration", *Proceedings of the European Design and Test Conference (ED&TC'97),* Paris, France, March 17-20, 1997.

[4] Wallner, S., "Design Methodology of a Configurable System-on-Chip Architecture", *Proceedings of the 12th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM'04),* Napa, CA, April 20-23, 2004.

[5] D. Van Ess, E. H. Currie, and A. Doboli, "Laboratory Manual for Introduction to Mixed-Signal, Embedded Design", ISBN 978-0-9814679-1-7

[6] Seguine, D., "Just Add Sensor – Integrating Analog and Digital Signal Conditioning in a Programmable System on Chip", *Proceedings of IEEE Sensors,* Vol. 1, pp. 665-668, November 2002.

[7] Hall, T., and Hamblen, J., "System-on-a-Programmable-Chip Development Platforms in the Classroom", *IEEE Transactions on Education,* VOL. 47, No. 4, November 2004.

[8] Nurmi, J., Madsen, J., Ofner, E., Isoaho, J., and Tenhunen, H., "The SoC-Mobinet Model in System-on-Chip Education", *Proceedings of the 2005 IEEE International Conference on Microelectronic Systems Education (MSE'05),* Anaheim, Ca, June 12-14, 2005.

[9] De Man, H., "System-on-Chip Design: Impact On Education and Research", *IEEE Desgin & Test of Computers,* Vol. 16, Issue 3, 1993.

[10] D. Montgomery, E. Peck, and G. Vining, "Introduction to Linear Regression Analysis", ISBN: 978-0471315650

[11] Cypress Semiconductor Website: http://www.cypress.com