TEACHING TOUCH SENSING TECHNOLOGIES USING ARM CORTEX-M4 MICROCONTROLLERS

[1]Nannan He, [1]Han-Way Huang, [2]Qijie Cai
[1]Electrical and Computer Engineering Technology Department
[2]Academic Technology Services
Minnesota State University at Mankato

Abstract

This paper presents our experiences of introducing in a senior level microprocessor course the latest touch sensing technologies, especially programming capacitive touch sensing devices and touchscreens. The emphasis is on the teaching practice details, including the enhanced course contents, outcomes and lecture and lab organization. By utilizing the software package provided by Atmel, students are taught to efficiently build MCU-based embedded applications which control various touch sensing devices. Different from the previous work on teaching simple capacitive touch only 8-bit Arduino boards, this work makes use of the 32-bit ARM Cortex-M4 microprocessor to control complex touch sensing devices (i.e., touch keys, touch slider and touchscreen). The Atmel SAM 4S-EK2 board is chosen as the main development board employed for practicing the touch devices programming. Multiple capstone projects have been developed, for example, adaptive touch-based servo motor control, and calculator and games on the touchscreen. Our primary experiences indicate that the project-based learning approach with the utilization of the selected microcontroller board and software package is efficient and practical for teaching advanced touch sensing techniques. Students have shown great interest and capability in adopting touch devices into their senior design projects to improve human-machine interfaces.

Introduction

Modern touch sensing devices allow rapid, accurate and direct interaction with the user, which conventional keyboard and mouse systems cannot offer. Many touch sensing technologies are currently available. Among them, the analog resistive, surface capacitive, projected capacitive, infrared grid, optical imaging and surface acoustic wave are the most important ones. The touchscreen, an electronic visual display controlled by touching the screen, is popular in many information appliances like tablet computers and smartphones. Semiconductor manufacturers have also recognized the trend of using the touch device as a highly desirable user interface component and started to integrate the touch-sensing technology into their products. To keep up with this industry technology change, we prepared a series of lectures and lab projects on touch sensing technologies for an advanced level microprocessor course provided to electrical and computer engineering students.

This new course extension covered the advanced touch sensing relevant topics, such as the history of the touchscreen, the fundamental electronic principles underlying each of the major touch sensing technologies, and the application issues for developing microcontroller-based embedded systems which utilize touch devices. Resistive touch and capacitive touch (i.e. surface and projected) were taught in detail. To deepen students' understanding of the advantages and disadvantages of these touch technologies, the comparisons among these technologies and their applications in real-world electronic devices were discussed. Among these topics, this course focused on teaching students to program various touch sensing devices through a series of project examples, from the simple ones which use only one kind of touch device to more complex ones which involve multiple touch devices. Project-based learning was employed as the main teaching approach.

The Atmel SAM 4S-EK2 board was the main development board employed for practicing touch device programming. The microprocessor employed on this board is the 32-bit ARM Cortex-M4. This board includes rich touch sensing devices, for instance, five touch keys and a touch slider which utilize capacitive touch sensing technology, and a resistive touchscreen on a color LCD display. Furthermore, Atmel provides a royalty free and open-source software library called Atmel Software Framework which includes a complete set of library functions for developing various microcontroller applications in C. This framework includes the QTouch library which provides Application Programming Interfaces (APIs) for programming Touch elements in particular. Students learned to utilize this framework so as to provide touch sensing capability in their projects. Students have shown great interest in this new topic and are capable of incorporating touch devices to improve the human machine interface of their senior design projects.

This paper mainly presents our primary experience of teaching touch sensing technologies to engineering and technology students, with the emphasis of how we designed the lab projects with the utilization of the Atmel development board and software packages. The description of this course is first given, including learning outcomes, course contents and project organization. Then, we present the lab projects and capstone projects developed by students using the selected Atmel development board integrated with touch devices and the software package to program these devices. Finally, the paper is ended with conclusions and future work

Course Design Principles

Many scholars argue that more practical components and hands-on experience should be brought into the engineering curriculum so as to prepare students who can develop out-of-the-box yet pragmatic solutions to engineering challenges [1]. As a result, project-based learning models have been increasingly integrated into engineering courses in recently years, especially in the USA [2]. Project-based learning (PBL) reflects the Theory of Constructivism. The central theme of the theory is that learning should be an active process in which learners construct new ideas or concepts by getting involved in the process of learning and experience [3]. Applying this principle, PBL offers authentic learning experiences and requires students to go through an extended, “hands-on” process of inquiry to a design challenge [1, 3].

According to current research, PBL has been identified as an effective and promising pedagogical model in engineering and science education [4]. When incorporated in the course, PBL can help students develop engineering thinking and intuition and increase their motivation. It also allows students to control their own learning, and therefore, students can develop more responsibility and ownership of their study [5]. Additionally, through working on real industry projects, students gain real-life experiences, as well as the knowledge and skills and interpersonal abilities that are required by the professional world [6].

Our course design has several distinct characteristics that reflect the principles of PBL:

- We designed four projects for students to complete. Each project presents a realistic, not school-like challenge, which requires students to focus on questions or problems that directly relate to the central concepts of the subject [7].

- Students are given clear guidelines on how to approach the problem and are directed to the necessary knowledge that is essential to identifying solutions to the problem [8]. For example, we developed the coding templates to help the students get started and the step-by-step, scaffolding instructions to make tasks more manageable.

- As students are gradually empowered to solve the tasks, we designed the last capstone challenge, which presents a more complicated and challenging task. It encourages students to apply the knowledge and skills they gained from the previous project experience to develop solutions to new problems.

- Last but not least, throughout the whole PBL process, students are instructed to run the code they developed to see the results of their design solution. Therefore, students received instant feedback that motivated them to use learning and metacognitive skills to diagnose problems and evaluate overall outcomes [9].

The overall goal of this new teaching practice is to equip students with the knowledge of advanced touch sensing technologies and developing microcontroller-based applications involving various touch sensing devices to solve engineering problems in practice. To achieve this goal, we used the problem-based learning model to design the course content and activities.

Course Learning Outcomes

We taught the added course materials in 5 weeks, two hours of lecture time and three hours of laboratory per week. It has three major objectives.

- To improve students' awareness of common and different features among major touch sensing technologies.

- To introduce students to existing popular touch sensing devices including touchscreen and relevant software development packages.

- To enable students to develop microcontroller-based systems to program touch sensing devices.

In the design of this course, we identify course learning outcomes that stem from the overall teaching goal discussed before and extend them to learning activities. We developed seven learning outcomes, grouped into three top outcomes.

1. To demonstrate the basic knowledge of advanced touch sensing technologies
   1.1 To understand the working mechanism of important touch technologies (i.e. resistive and capacitive touch)
   1.2 To articulate different features of various touch technologies and their main application fields.

2. To demonstrate the capability of designing touch sensing applications
   2.1 To comprehend the composition of basic touch sensing devices (i.e., QTouch slider and keys from Atmel)
   2.2 To get familiar with the software tools and packages (i.e., Atmel QTouch Library) for programming touch devices.

3. To demonstrate the capability of developing microcontroller-based touch sensing applications
   3.1 To utilize software packages for developing simple touch sensing applications on ARM microcontrollers
   3.2 To utilize software packages for developing touchscreen applications on ARM microcontrollers
   3.3 To utilize touch sensing devices to improve the human-machine interface experiences of microcontroller-based embedded systems.

Course Contents

The topics covered in this course extension include origin of the touchscreen, fundamental electronic principles, construction, features and benefits of touch sensing technologies such as analog resistive touch, surface capacitive touch and projected capacitive touch, comparison among popular touch sensing technologies, Atmel QTouch keys and slider, resistive touchscreen on a color LCD display, Atmel QTouch library, and the design and development of touch applications on ARM Cortex-M microcontrollers. Course materials were drawn from white papers, user manuals, example projects, technical reports and presentations [10-12] from major touch device vendors (i.e. Elo TouchSystems and 3M) and semiconductor companies (i.e., Atmel, Texas Instruments) which have provided touch solutions. To the best of our knowledge, there is no textbook which introduces Touch sensing technologies and microcontroller-based development from the practitioner's point of view. Table 1 shows the classification of these topics. In this table, the time schedule of each topic is given within the five week timeframe.

Students started preparing their final project from week 3, including designing the project and completing the project proposal report. During week 4, five basic touch application lab projects were introduced to students, from the project creation, library modules integration, code editing and debugging, project analysis. These projects played the important role in our project-based learning. They inspired the students and demonstrated the capability and applications of touch devices. Furthermore, these examples laid the programming foundations for students to develop their final projects which are more complex projects targeted at solving practical engineering problems.

Table 1. Course topics.

1. Fundamentals (week 1)
Evolution of touchscreen
Analog resistive touch sensing (Four/five/eight wire resistive touchscreen)
Surface capacitive technology
Projected capacitive technology (Self-capacitance and mutual-capacitance)
Infrared grid & optical imaging touch sensing
Comparison of touch technologies
2. Touch device programming (Week 2)
Atmel capacitive touch acquisition methods (QTouch and QMatrix)
Atmel Touch devices
Resistive touchscreen
Atmel Software Framework (ASF)
Atmel QTouch library
3. Microcontroller-based touch application development (week 3, 4)
Touch devices (i.e., key, slider) combined with UART development
Touch devices (i.e., key, slider) combined with GPIO development
Touch devices (i.e., key, slider) combined with PWM development
Basic touchscreen and graphical LCD development
Touchscreen combined with touch devices development
4. Final projects development (Week 3, 4, 5)

Use of Atmel Touch Devices and ARM Cortex-M Microcontrollers on Projects

In this section, we present our experience of teaching engineering and technology students the knowledge of touch technologies from two aspects: development board selection, and lab projects and capstone projects

Microcontroller Board Selection

As the computing capability of MCUs (for example, ARM Cortex-A and Cortex-M series) increased dramatically, major microcontroller vendors started providing their solutions to develop touch applications. Such technical solutions typically consist of the following components: 1) touch devices or touchscreen either embedded in some microcontroller boards or as individual chips (i.e., a Booster Pack from TI); 2) software library to program the touch devices; 3) facilities to assist developing touch-based applications, such as simulation tools to measure and adjust the sensitivity of touch sensors. Nowadays, many low-cost development boards are equipped with touch elements that can be used for education purposes. For example, the STM32L1 discovery board from ST Microelectronics includes a linear touch slider and four touch keys Freescale's FRDM-KL25Z development platform provides a capacitive touch slider; Texas Instrument offers low power touch devices for keys, sliders, wheels and proximity applications. Atmel offers two types of patented capacitive touch acquisition methods, QTouch and QMatrix. QTouch is based on the self-capacitance sensing technology. Placing a finger on the touch surface introduces external capacitance that affects the flow of electron charge at the touch point. It supports the detection of proximity of a finger rather than absolute touch and is easy to tune the sensitivity. In comparison, QMatrix detects a touch by a scanned passive matrix of electrode sets, which belongs to the mutual-capacitance sensing technology category. It is more resilient to moisture & environment changes.

In this course, the Atmel SAM4S-EK2 evaluation kit, which is equipped with a 32-bit ARM Cortex-M4 microprocessor, was chosen as the development board for conducting lab projects due to three main reasons. Firstly, this kit includes rich touch sensing elements: a 2.8" color graphical LCD display with resistive touchscreen, five touch keys (UP, DOWN, RIGHT, LEFT and VALID) and a touch slider. Both touch keys and sliders utilize the QTouch acquisition method. The touch keys use five pairs of IO pins of the MCU to detect their states (i.e., pressed or released); the slider uses three pairs of IO pins to detect a linear finger displacement on a sensitive area. Secondly, Atmel offers a QTouch Library, a free software library for simplifying the development of capacitive touch sensing applications on AVR and ARM microcontrollers [12]. It provides multiple library files for each touch device such as key, slider and wheel. Using the library files, the host application can easily make touch measurements and determine the status of touch sensing elements. Moreover, The Atmel Software Framework (ASF) provides a rich set of software modules classified as *board*s, *driver*s, *component*s and *service*s to program microcontrollers. In fact, the QTouch library is included in ASF as a *service* module. And the software tool ASF wizard facilitates importing the QTouch Library and linking together other driver or service modules of an application project. Thirdly, Atmel provides around one hundred example projects for developing various applications on the SAM4S-EK2 kit. Two examples are mostly related to Touch elements development. One is sensing the status of touch keys and slider on board; another is calibrating the resistive touchscreen, and displaying the touching position on the terminal after successful calibration. In this course, we developed four more projects for this evaluation kit to demonstrate the application of QTouch elements and touchscreen in ARM-based embedded systems. Students could quickly learn programming microcontrollers and touch devices from these example projects. Moreover, they could use them as the basis to develop their own applications. More details of these projects and their usage in this course are presented in the following.

Lab Projects and Capstone Projects

The lab projects that we prepared for this course serve different teaching purposes. We organized them into two groups according to the complexity of the projects and applied them to the course in two phases. Table 2 gives an overview of four basic lab projects that were introduced to students in the first phase. We first introduced to students four basic lab projects which utilize only one kind of touch element. The lab manual for these projects provides the step-by-step instructions about creating an ASF board project for the SAM 4S-EK board, adding into the project the ASF software library modules, building and debugging the projects. The major ASF software library modules we used include standard serial I/O driver, Atmel QTouch services, and components for touchscreen programming (e.g., backlight controller, LCD controller, resistive touch service). Through the practices in these phase I lab projects, students could have a better understanding of the usage of common API functions in these library modules.

Phase II projects are summarized in Table 3. These projects were used as examples to show students how to develop common MCU projects with the integration of touch elements. Students could also use them as the basis to design and implement their final course projects.

Table 2. Phase I Lab Projects.

Project Name	**Function description**	**Learning purposes**
Getting_started	- Blink on-board LED at fixed rate. - Display messages on a terminal application via serial UART function	- Create, build and debug projects for the 4S-EK2 board in Atmel Studio; - Get familiar with ASF and ASF wizard - Program core peripherals of ARM Cortex-M4 microprocessor.
QTouch	- Detect the status (i.e., pressed or released) of five Qtouch keys and slider on board. - Output Qtouch values when touching via serial UART IO.	Study QTouch library. - Use the C structure that holds the status of touch keys, and linear positions of the slider. - Use API functions for configuring and monitoring the status of QTouch elements.
Graphic_LCD	- Configure the graphical LCD controller on-board. - Draw text, image, basic shapes (i.e., line, rectangle, circle) on LCD.	Study library for graphical LCD - Use API functions to initialize LCD and turn on the backlight. - Use API functions to choose foreground color. - Use API functions to draw basic shapes on LCD.
Touchscreen	- Configure and initialize the touchscreen controller as well as the underlying graphical LCD. - Calibrate the touchscreen, then output the pen position on the touchscreen at the terminal.	Study library for resistive touchscreen - Use APIs for configuring touchscreen. - Use APIs for the touchscreen calibration. - Use the event_handler() as the callback when the pen is pressed on the touchscreen.

Table 3. Phase II Lab Projects.

Project Name	Function description	Learning purposes
QTouch_PWM	- Use one QTouch key to start and stop the PWM generation. - Use QTouch slider to control the duty cycle of generated PWM waveform. (Left and right poles of the slider represents 0 and 100% duty cycle respectively.)	- Study how to use the status of Touch sensing elements to configure the application of timer and PMW functions. - Prepare to develop complex projects, such as using a touch device to smoothly control servo motor speed.
touchscreen_GPIO	- Draw two circles on the touchscreen with different blue/green colors. - Detect the touch on two circles: Touching on the blue/green circle toggles the on-board blue/green RGB led respectively.	- Study how to integrate touchscreen into the conventional GPIO applications of MCU.
touchscreen_LCD	- Adjust the four corner positions of the calibration area. - After calibration, draw rectangles /circles from the center of touching position with fixed size (width and height) / radius.	- Study how to use the touchscreen in the basic graphical applications of MCU.
touchscreen with QTouch	- Draw a circle at the center of LCD. - Use four QTouch keys to move the circle towards up, down, left and right at a fixed distance.	- Study how to develop applications which combine both QTouch features with touchscreen features.

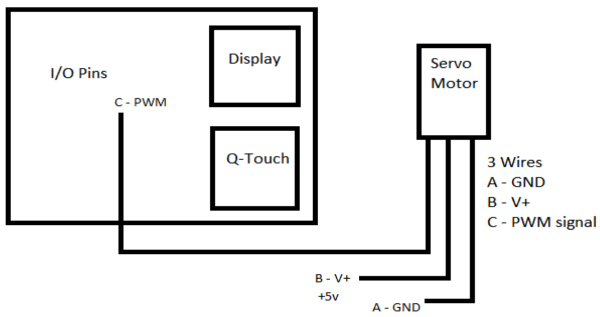
We briefly introduce three capstone projects developed by students in the following. Through these projects, students have shown the capabilities of programming touch devices and applying them in complex MCU projects.

Cyber Adaptive Spinny Thing

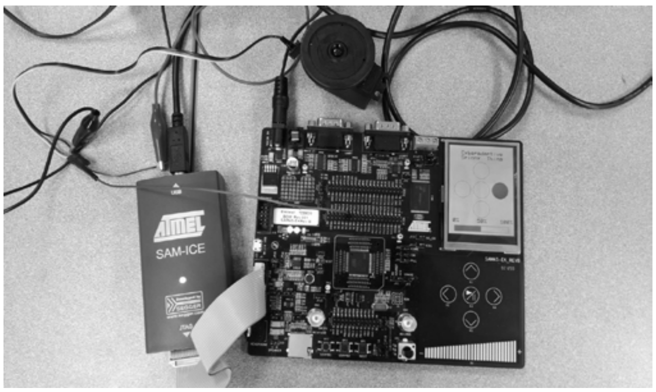
This project presented an MCU application to drive a Hi-Tec HS-311 servomotor. Users could control the speed of the servomotor by the QTouch slider on the SAM4S-EK2 board and the rotation direction by the QTouch keys (RIGHT, LEFT and VALID) on board. The RIGHT key controls rotating counter-clockwise, the LEFT key controls the motor's rotation is clockwise, and the VALID key in middle starts or stops the rotation. The graphic LCD screen of the microcontroller is also applied to display the status of QTouch elements. The slider is represented as a rectangle at the bottom of the screen whose filled area corresponds to the position of the slider and also the speed of the servomotor. Figure 1(a) shows the wiring diagram, (b) displays the test setup and (c) the GUI. Students showed high interest in using QTouch to flexibly and smoothly control the operation of the motor.

Touchable Calculator

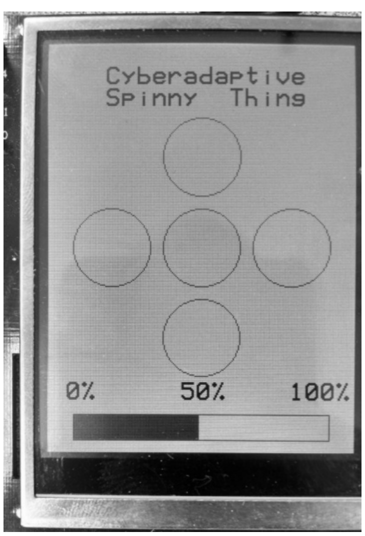
This project is about creating a basic touchable calculator on the resistive touchscreen on the SAM 4S-EK2 board. Ten numbers 0-9, four basic arithmetic operators "/", "x", "-" and "+" and two control symbols "AC" and "=" are drawn within sixteen squares on the screen. To enter an operand, a user touch the corresponding square with the stylus, for example, the square labeled with "5". The entered number is then displayed on the top bar as shown in Figure 2. The program then reads the operator and the second operand from the user in the same manner. After the user touches the "=" symbol, the result of the operation is displayed on the top. The program cleans the memory and starts reading new entries from the users by pressing the square labeled with "AC".



(a)



(b)



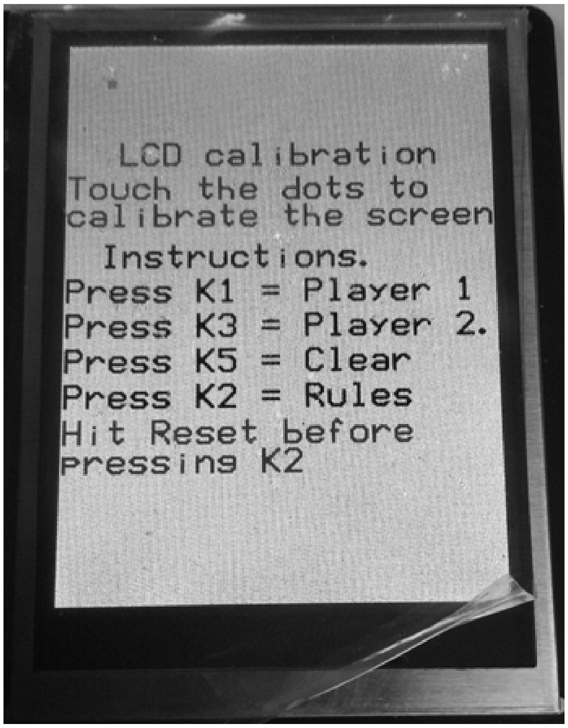
(c)

Figure 1: Capstone Project 1
(a) Design diagram, (b) Test setup, (c) GUI.

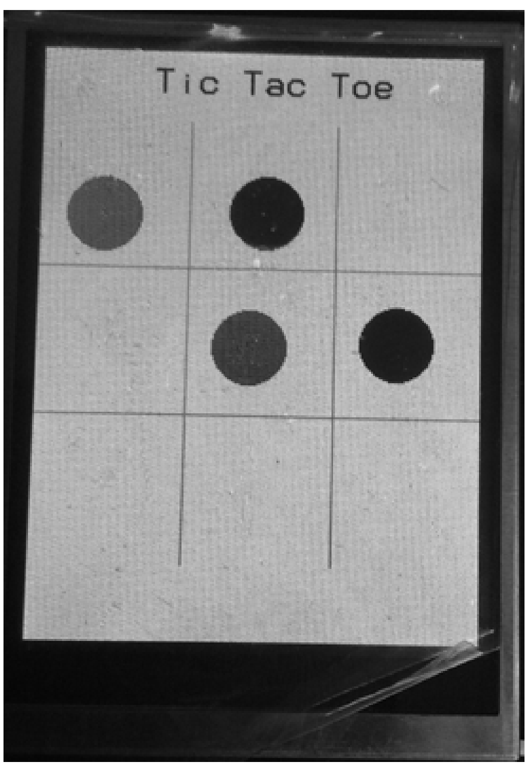
Figure 2: Capstone Project 2.

Miniature TIC-TAC-TOE Game

This project aims at creating a simple version of the Tic Tac Toe game with two players using the touchscreen and QTouch keys. Once the board is powered on, a user menu is shown on the screen as in Figure 3 (a). Then, the user calibrates the touchscreen by precisely pressing red dots at each corner using the stylus. After the successful calibration, the game starts off with a grid consisting of two vertical lines and two horizon lines. User 1 could first press the QTouch key K1 (TOP), afterward, when a cell of the touchscreen is touched, the program draws a circle in blue within that cell. Then, User 2 could start his/her move by pressing the QTouch key K3 (MIDDLE), a circle in black is drawn within the area of the cell that has been touched. In this project, Qtouch keys are used for switching between two players, displaying game rules and clearing the touchscreen. Figure 3 (b) shows the results after two movements from players.



(a)



(b)

Figure 3: Capstone project 3
(a) User menu, (b) Game panel.

Conclusions

This paper presents our experiences in teaching engineering and technology students advanced Touch sensing technologies. An increasing number of low-cost MCU development kits embedded with Touch elements have become available for teaching Touch technologies. The Atmel SAM4S-EK2 board was employed in the advanced Microprocessor course to teach students Touch device programming. This evaluation board is equipped with rich resistive and capacitive Touch sensing elements. By utilizing the Atmel QTouch library and Atmel Software Framework, students are able to develop MCU-based capstone projects using the touch sensing devices. Some enhanced the user interface of a traditional motor control project; some are touchscreen centered projects for multi-media applications. In the future, we will encourage students to develop more complex MCU projects with Touch devices.

References

1. K. J. Chua, W. M. Yang, and H. L. Leo. "Enhanced and conventional project-based learning in an engineering design module." *International Journal of Technology and Design Education, 24*, 437-458, 2014.

2. J. A. Simoes, C. Relvas, and R. Moreira. "Project-based teaching-learning computer-aided engineering tools". *European Journal of Engineering Education, 29*(1), 147-161, 2004.

3. N. Jumaat, and Z. Tasir. "Integrating project based learning environment into the design and development of mobile apps for learning 2D-animation". *Procedia Social and Behavioral Sciences, 103*, 526-533, 2013.

4. M.R. Yasin and S. Rahman. "Problem oriented project based learning (POPBL) in promoting education for sustainable

development". *Procedia Social and Behavioral Sciences, 15,* 289–293, 2011.

5. M. Frank, I. Lavy, and D. Elata. "Implementing the project-based learning approach in an academic engineering course". *Intl. Journal of Technology and Design Education, 13*(3), 273–288, 2003.

6. Y. I. Chang and C. L. Miller. "PLM curriculum development: Using an industry-sponsored project to teach manufacturing simulation in a multidisciplinary environment". *Journal of Manufacturing Systems, 24*, 171-177, 2005.

7. J. W. Thomas. "A review of research on project-based learning". Retrieved from http://www.bie.org/index.php/site/RE/pbl_research/29, 2000.

8. T. Markham, J. Larmer, and J. Ravitz. "Project-based learning handbook: A guide to standards focused project-based learning for middle and high school teachers". *Novato, CA: Buck Institute for Education*, 2003.

9. P.C. Blummenfeld, E. Soloway, R.W.Marx, J. S. Krajcik, M. Guzdial, and A. Palincsar. "Motivating project-based learning: Sustaining the doing, supporting the learning". *Educational Psychologist, 26*(3/4), 369-398, 1991.

10. White paper – "Comparison All Resistive Touch Technologies", 2014. http://www.elotouch.com/Technologies/compare_resist.asp

11. Geoff Walker, "Fundamentals of Touch Technologies", 2013. www.walkermobile.com/Touch_Technologies_Tutorial_Latest_Version.pdf

12. Atmel QTouch Library: http://www.atmel.com/products/TouchSolutions/touchsoftware/default.aspx

Biographical Information

Nannan He is an Assistant Professor of the ECET Department in Minnesota State University at Mankato, Minnesota, USA. She received the Ph.D. in computer engineering from Virginia Tech. Her teaching and research interests are in embedded software, real-time systems, and software verification.

Han-Way Huang is a Professor of the ECET Department of Minnesota State University at Mankato, Minnesota, USA. He received the Ph.D. in computer engineering from Iowa State University.

Qijie Cai is an Instructional Designer and Evaluation Specialist for Academic Technology Services at Minnesota State University, Mankato. She holds a Ph.D. in School Administration and Supervision from the University of Virginia. Her research interests include school leadership, educational organization, and instructional design and evaluation.