Proceedings,
The 2013 ASEE North Midwest Section Conference,
October 17-18, 2013, Fargo, North Dakota.

Exploring Real-Time Applications in Hands-On Automation Courses

Nannan He, Ph.D, nannan.he@mnsu.edu
Gale Allen, Ph.D, gale.allen@mnsu.edu
Cameron Johnson, Senior Student, cameron.johnson@mnsu.edu
Minnesota State University Mankato

Abstract

Industry and legislative leaders press for improving the quantity and quality of the work force. For example, the need for capable, dedicated, and experienced automation engineers continues to increase. The training involves expensive laboratory equipment, small class size, and motivated faculty, but University budgets are decreasing and the emphasis on research and journal publications for tenure and promotion is increasing. At MSU Mankato state and industry support has come together with faculty interest for the past several years in the areas of automation engineering and manufacturing. Courses in industrial automation involving PLC's, sensors, and actuators have been taught since 2006 using hands-on active learning techniques. An effort is underway to increase the technical depth and broaden the training by exploring deterministic timing and modeling in complex real-time automation systems using traditional PLC and PC-based PLC equipment and future, large multicore computer designs.

1. Introduction

In Spring 2006, a laboratory and courses were first created for training our Electrical Engineering Technology and Computer Engineering Technology students in component-level industrial automation at Minnesota State University, Mankato. The financial support for building the laboratory came from several sources. The primary support was provided by the Minnesota Center for Excellence in Manufacturing & Engineering (MNCEME). Significant funding was provided by the College of Science, Engineering and Technology and by the Department of Electrical and Computer Engineering and Technology. Industry also provided strong support, for example, the equipment from Rockwell Automation, National Instruments, etc., was given through their education discount program. The courses take two semesters in sequence. They cover the details of automation components including programmable logic controllers (PLC's), actuators and robots, sensors, motors, drivers, and operator control displays. Students design, simulate, build, test and document automation systems for capstone projects to demonstrate their understanding of the subjects. The initial version of the courses and lab was described in a paper in late fall 2006 [1].

Since then, several different types of laboratory equipment have been added the course syllabus and material has been continuously improved. Active learning and hands-on learning are the basis for instruction in the courses. Several short projects have been added to the course. Teams of two-to-three students develop and complete capstone projects. More details about these improvements and projects have been presented in another paper [2].

Recently, from interactions with engineers and educators involved in automation, several members of our faculty have come to believe there is a need for engineers and technologists with more depth of knowledge of real-time control software and hardware. This includes specialized high-speed processor and communications technology. To address this need MnCEME provided funding to start building the resources and faculty needed to train engineering students and technology students in these areas. Alexandria Technical and Community College, thanks to Dr. Kennneth Ryan, helped start the project and has provided on-going support.

In support of this Beckhoff Automation provided automation equipment for the laboratory through donations and through their educational discount program. This consists of six stations of PLC's, motors, and drivers along with two PC-based stations. The laboratory space is being increased thanks to renovation funding provided by the University. In addition, 3S-Automation has provided technical support for students to gain experience in embedded applications using their development and runtime software. Rockwell Automation continues to support our automation efforts including work in real-time applications.

In this context, an effort is underway to increase the technical depth and broaden training by investigating deterministic timing and modeling techniques in complex real-time automation systems with traditional PLC, PC-based PLC equipment and future large multicore computer settings. This paper presents a new real-time embedded systems course, and the work of integrating real-time system design concepts, tools and applications into the hands-on automation courses.

2. Real-Time Embedded Systems Course description

2.1 Overview

This new real-time embedded systems (RTES) course targets learning real-time systems design and applications from the practitioner's point of view. This course is organized as 3h of lecture and 2h of laboratory per week. It has three objectives. The first is to improve students' awareness of real-time specifications in critical automation controllers and other embedded systems. The second is for engineering student to apply modern development tools and advanced techniques to designing and analyzing performance of small-scale real-time systems. The third is to enable student to develop real-time applications to solve problems with specific timing requirements. Moreover, in order to be compatible with the instruction approach of existing automation courses in which active learning and hands-on learning are the basis, this course has an experiential component that employs hands-on learning. It allows students to apply the advanced techniques learnt from this course to develop an understanding of their advantages and disadvantages in different applications.

The topics covered in this course include real-time scheduling approaches such as clock-driven scheduling and static and dynamic priority driven scheduling, resource handling, timing analysis, real-time operating systems (RTOS), hard and soft real-time systems, distributed real-time systems, concepts involved in the modeling, design, analysis and verification of real-time systems. Course materials were drawn from two text books [3, 4], FreeRTOS tutorial book and reference manual, CoDeSys Runtime User Manual, ARM Cortex-M microcontrollers' datasheets, websites, and other publications. Table 1 shows the classification of these topics.

1. Fundamentals (week 1)	Requirement Eng. for real-time systems
Basic concepts and misconceptions	(semi-) Formal methods in system specification
Multidisciplinary design challenges	**5. Performance analysis techniques (10th – 12th)**
2. Hardware for real-time systems (2nd-3rd)	Timing estimation of real-time system
Process architecture	Queue theory applications
Architectural advancements	Input/output performance
Peripheral interfacing	**6. Additional application issues (13th -14th)**
Distributed real-time architecture	Design for fault tolerance
3. Real-time operating systems (3rd - 6th)	Software verification & system integration
Multi-task scheduling	Performance optimization
System services for application programs	**7. Case studies: RTOS in practices**
RTOS selection issues	FreeRTOS (open source)
4. Requirement engineering (6th – 8th)	CoDeSys runtime system (industry)

Table 1. Course topics

2.2 Course Learning Outcomes

Our overall aim is to equip students with the knowledge of designing real-time systems and developing real-time applications to solve engineering problems in practice. In the development of this course, we identify course learning outcomes that stem from this aim and extend to learning activities. We developed twelve learning outcomes, classified into three core components as shown in Figure 1.

1. To demonstrate the ability of correctly defining real-time systems
   1.1 To identify problems as hard, firm or soft real-time system;
   1.2 To articulate and contrast different definitions in real-time systems.

2. To demonstrate the ability of real-time systems design
   2.1 To comprehend formal methods based specification approaches and utilize modeling tools;
   2.2 To understand the impact of hardware for real-time performance;
   2.3 To analyze the scheduling feasibility of a set of independent tasks and derive schedules;
   2.4 To understand resource policies and system services for inter tasks communication and synchronization;

   2.5 To understand the challenges and applications of performance analysis techniques;
   To understand real-time issues on advanced distributed control networks such as SCADA;

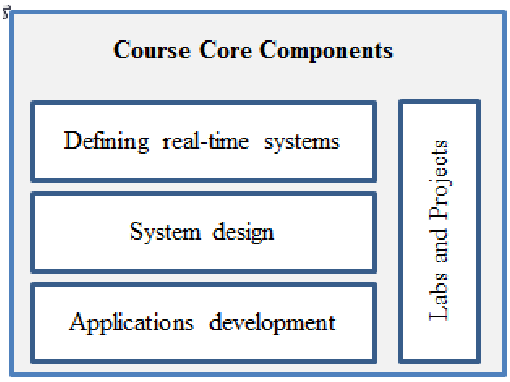

Figure 1. The course is designed based on three core components.

3. To demonstrate the ability of basic real-time application development
   3.1 To understand real-time software testing, verification and system integration.
   3.2 To be aware of performance optimization techniques.
   3.3 To utilize modern tools to simulate executions and critique different implementation choices.
   3.4 To comprehend the architecture, functions and applications of one or two existing RTOS.

2.3 Instruction approach

Similar to the existing automation courses, active learning and hands-on learning are also the fundamental teaching approaches applied to this real-time system design course. All of the classes are held in the laboratory. For this course, this setting eases the flexible adoption of a variety of teaching methods, depending on the characteristics of the different course topics in sequence. In Table 1, the time schedule of each top-level topic is given for this one semester course (16 weeks). Please note that the topic 7 - *Case studies* is not labeled with a specific schedule because its sub topics are provided in the combination with other topics throughout the semester. The teaching formats and material employed in this course are presented as the following.

The rationale for developing this course stems from our efforts to improve our undergraduate Computer Engineering program. At this time it is an elective for senior undergraduate and graduate computer and electrical engineering students. Undergraduate computer engineering students will have taken courses in software, computer architecture, and operating systems prior to taking this course. Electrical engineering students and other students with these skills are welcome to take the elective. Future improvement efforts may result in the course becoming required for CE students.

The course focuses on the practical applications as compared with graduate-level courses in universities which tend focus on the research and theoretical issues. Our laboratory experiences involve use of current real-time open-source software and supported microcontrollers.

At the beginning, to introduce students the topics on defining real-time systems, power point slides presentation and class discussion are used to in the lectures. As these topics are the basis for further learning, it is important to help students to set up a solid and comprehensive foundation. In class discussion questions are designed to enable student to reflect on key concepts in real-time systems, and to encourage active learning. Here are some examples: 1) Are real-time systems synonymous with '*fast or high performance*' systems?, 2) "In the statement 'All practical systems are ultimately real-time systems', what is your idea of the *degree of 'real-time*'?, 3) "Where a *response time* requirement of a system could come from?". As the discussions proceeds, students gradually deepen the comprehension and open their minds, at the same time are inspired to judiciously observe and analyze real-time applications. The initial homework exercises in this course are extracted from the practical application problems. One example question is: considering an automation system for car assembly, describe three different scenarios, classify the system as hard, firm or soft real-time under each of these scenarios. Such exercises are designed with open end questions. The goal is for students to think and give justification for their answers. The main purpose is to fortify student's understanding so as to be able to apply them in practice.

Existing real-time system courses are mostly offered to computer science or computer engineering major students for conducting the scientific research in this area, covering topics like reference models of real-time systems, algorithms for scheduling, resource access control, and RTOS. However, this course emphasizes engineering problems, from requirement engineering, hardware configuration, RTOS related issues and selection to timing performance evaluation for real-time system design. The primary experience of teaching requirement engineering and hardware for real-time system is presented in this subsection. The rest is introduced in the following subsection on lab assignment.

The main purpose of introducing formal requirement engineering techniques is for students develop an appreciation for the automated formal or semi-formal methods in rigorously specifying real-time system requirements. An example Sub-System Requirements Document for Fuel Management from AIRBUS is introduced as a case study. Students are convinced that formal specification like State chart is not just a scientific research issue, in fact has been widely used in requirement specification of safety-critical embedded systems in industry to avoid the ambiguity caused by conventional text-based specifications. Later on, they show great enthusiasm in learning the formal or semi-formal approaches that its relevant mathematical concepts are difficult for non-science major students. In the course evaluation, students made several comments that the requirement engineering offered in this course is one of the most beneficial aspects to them.

There is the increasing number of microcontrollers (MCUs) supporting real-time applications. In this course, the following MCU development boards and Integrated Development Environment (IDEs) are employed to be available for students.

- PIC24, dcPIC (Explorer 16 Development board from Microchip)
- ATmel SAM4S-EK (ARM Cortex-M4 microcontroller from Atmel)
- ATmel SAM4L-EK (ARM Cortex-M4 microcontroller for low power from Atmel)
- µVision IDE for ARM programming from Keil
- Atmel Studio with Atmel Software Framework (ASF) from Atmel
- MPLAB IDE from Microchip

The above MCUs are selected as they are all supported by the FreeRTOS software which is the main RTOS studied in this course. Another paper is dedicated to report our experience of teaching real-time embedded systems design on ARM-based MCUs.

The last part of this course offers students the topics served for the application development core component. As real-time systems are often applied in the ‘critical’ embedded applications with respect to reliability, safety and security, verification and validation (V&V) is an important issue in real-time application development. An on-going research on model-based V&V is incorporated in this course so that students could be exposed with the latest V&V advances. More details of this work could be found in another paper.

2.4 Lab Assignment and project design

Educators have explored hands-on RTOS development for enhancing student learning in real-time systems course [7,8]. The lab assignment based on the open source RTOS - FreeRTOS is the important experiential component of this course, which aims at gives student rich hands-on experience in building real-time systems. Figure 2 shows the core lab components.

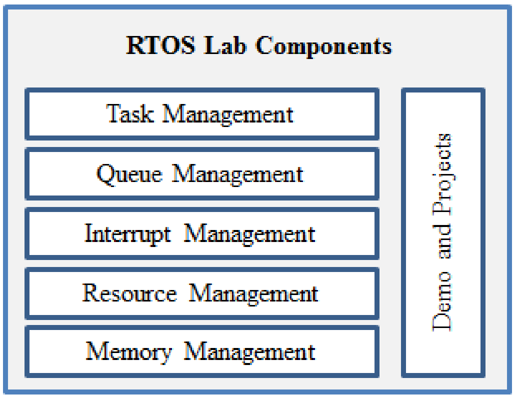

Figure 2. The RTOS lab assignment includes six core components.

Three things are prepared for tutoring each lab session before students start working on the lab assignment. First, a list of questions is designed for students to figure out the answers during each lab session. These questions are served as the guidelines to assist student's hands-on learning. Second, the concepts or algorithms related to each lab work, which has been introduced in lectures, are reviewed. Thirdly, as software programing is the main task in each lab, a set of relevant API functions are introduced to students for the efficient programming. These APIs are provided by either FreeRTOS or certain libraries included in a particular IDE. A standard demo project which incorporates MCU development board, simulator, logic analyzer and miniature RTOS with all basic features, is provided to students. They make use of this demo project as the basis to construct more application specific projects, which could achieve more efficient development compared with creating application from scratch. According to the course evaluation report, students gave the feedback that they benefit most from the hands-on programming experience.

In our automation courses, many capstone projects have been developed and completed by students. In the course evaluations, these projects have gained positive comments. Students said while working on the capstone projects, they learned most including learning about teamwork, persistence and technical skills. Details about some example capstone projects can be found in our previous paper [2]. Capstone projects created from real-time electro-mechanical applications will be the cornerstone of integrating real-time system design into automation courses. They can be explored from following three ways. First, some existing capstone projects can be modified to include real-time applications so as to achieve predictable and higher performance. Second, real-time applications based on Beckhoff Automation stations can also be developed and adopted as new capstone projects. Third, CoDeSys runtime system from 3S-Automation can be configured to be executed on specific MCUs so as to develop real-time automation systems. The created systems can be adapted to student capstone projects.

The Course Outcome, Course Goals, and related Student Outcomes are assessed using examinations and student work. Student work consists of laboratory reports, project presentations, and other documentation such as homework assignments. A numerical process is used to evaluate progress so as to provide a means of evaluating continuous improvement.

3. Conclusions

This paper presents an on-going work to achieve the ultimate education goal of increasing the technical depth and broaden training by investigating deterministic timing and modeling techniques in complex real-time automation systems. It mainly describes a new real-time system design course offered to EE and CE senior or graduate students. This new course is one of the important exploration steps towards the goal. At the same time, this goal is also an important guideline in the course preparation and teaching practices, as a result some special features of this course are formed compared with most existing real-time system design courses. The course objectives, topics being covered and learning outcomes of this course are first presented. Instruction approaches, laboratory equipment and assignment and projects employed in teaching are then described in detail.

Our courses in automation, senior/junior design, and microcontrollers are expected to benefit from this course. For example ur Industrial Automation courses EET 461 and EET 462 required for seniors our technology programs, EET and CET, will directly benefit by expanding the scope of these courses to cover some of the real-time coursework. Some senior design projects may focus on real-time automation as a result of students having taken this course.

In the future, we will continue adapting this new real-time system design course material to match the teaching approach, objectives and needs of the two semesters' automation courses. Capstone projects design is identified as the best way to integrated real-time system design and application development knowledge into the automation courses. The future work focuses on developing new capstone projects related to real-time automation systems in the three directions discussed above.

References

1. Gale Allen. "New Industrial Automation Laboratory & Courses, ECET Technology Program Advancement". ASEE IL/IN Section Conference, Indiana University Purdue University Indianapolis (IUPUI), Indianapolis Motor Speedway/Brickyard Crossing Hotel, Indianapolis, Indiana, March 30-31, 2007.
2. Gale Allen. "Hands-On Component-level Automation Courses for Technology and Engineering Students". *Proceedings of the ASEE North Midwest Sectional Conference*, Oct. 2008.
3. "Real-Time Systems" Jane W.S. Liu, Prentice Hall.
4. "Real-Time Systems Design and Analysis: Tools for the Practitioner" 4th edition, Philipp A. Laplante, Seppo J. Ovaska, Wiley Publisher.
5. "The FreeRTOS™ Reference Manual", FreeRTOS online.
6. "Using the FreeRTOS Real Time Kernel – A Practical Guide – Cortex-M3 Edition", FreeRTOS online.
7. Kumar, G.S.A.; Mercado, R.; Manimaran, G.; Rover, D.T., "Enhancing student learning with hands-on RTOS development in real-time systems course," *Frontiers in Education Conference, 2008. FIE 2008. 38th Annual*, vol., no., pp.S2H-11,S2H-16, 22-25 Oct. 2008.
8. Yu JX, Zhao YG, Li Y and Duang HY, "An Example of Course Project of Real-Time Multitask Programming", *I.J. Education and Management Engineering on MECS*, 2012.

Biographical information

Nannan He received the Ph.D. in computer engineering from Virginia Tech. From 2012 to present she is an Assistant Professor at the ECET department in Minnesota State University at Mankato. Her teaching and research interests are in safety-critical embedded software, real-time embedded systems, and software verification. She is an IEEE member and reviewers for many conferences and journals in EDA field.

Professor Gale Allen, Ph.D. held engineering and management positions in companies in the Minneapolis-St. Paul area for many years and in 2004 became a member of the faculty at Minnesota State University Mankato. His teaching and research interests are in electronics, communications, and industrial automation. He is a Senior Member of IEEE and is active in amateur radio.