2023 Annual Conference & Exposition
Baltimore Convention Center, MD | June 25 - 28, 2023
The Harbor of Engineering Education for 130 Years
Paper ID #39381

Implementing Laboratory and Project-Based Embedded Control Sequence Courses in Electrical and Computer Engineering

Dr. Maryam Nasri, SUNY Buffalo State University

Maryam Nasri is an Assistant Professor of Electrical in the Department of Engineering Technology at SUNY Buffalo State University, Buffalo, NY. She was previously an Associate Professor at the department of Electrical Engineering Technology at SUNY, Alfred State College of Technology, Alfred, NY. She received her Ph.D. in Electrical Engineering from Simon Fraser University, Surrey, BC, Canada. She completed her master's degree in Software Engineering from the University of British Columbia, Vancouver, BC, Canada.

©American Society for Engineering Education, 2023

Implementing Laboratory and Project-Based Embedded Control Sequence Courses in Electrical and Computer Engineering

Abstract

This paper will demonstrate a successful method of teaching a two-semester embedded course that will target the development of an autonomous firefighting robot at the end of the second course. These two embedded courses have been designed, implemented, and improved gradually within ten years. They have become one of the core courses of the two following Accreditation Board of Engineering and Technology (ABET) -accredited Bachelor of Science (BS) programs: Electrical and Computer Engineering Technology. Also, students in Mechatronics Technology are required to take the first Embedded Control course. In these two courses, students learn about AVR microcontroller architecture and interfacing them with embedded system components, such as sensors and actuators. In addition, these two courses include weekly lab experiments where students practice hardware setup and Assembly or C Programming by reinforcing lecture materials. While half of the course materials and fundamental lab experiments are included in the first course, most advanced topics and the course project are covered in the second course, which is an upper-division one. This paper will explain the primary materials covered in each of the two embedded courses and list most of the embedded lab experiments within these courses that led to the ability of the students to design and develop an autonomous Firefighting robot as their course project. As part of their coursework, students had to present and demonstrate their project at the end of the semester and attend the college's annual Robot competition. A sample lab experiment and a student's project results are included in the paper. Also, the results of student evaluation and course assessments according to the assigned ABET outcomes have been discussed at the end of this paper.

1. Introduction

A detailed Market Research Report published in 2020 suggested that the global market for Embedded Systems will grow 6.1% from 2020 to 2025 and reach USD 116.2 billion [1]. However, by searching the engineering school curricula, we can find a limited number of ECE departments that offer a comprehensive and hands-on course in Embedded control systems. Consequently, although the educational aspect of teaching Embedded Control Systems course have been reviewed in some of the article specialty in ASEE annual conferences [2], [3], higher education needs to train many more engineers familiar with the embedded system to fulfill the needs of a fast-growing market. Also, to educate the engineers who are familiar with real-work problems and ready to work on design and fabrication projects, Electrical and Computer Engineering programs need to add applied learning courses to their curriculum. Furthermore, by exposing students to the embedded control systems, they also will become familiar with the foundation of robotics, microcontroller, and autonomous system.

In these embedded courses, students deploy their knowledge and utilize devices from pre-sequel courses, such as Intro to Electrical Engineering and Digital Logics. Subsequently, the design and implementation of the autonomous robot project will help students learn a complete cycle of design and development of an embedded control system. This experience has benefited them in implementing projects in the other courses, such as Feedback Control Systems and Advanced Power Systems. In the latter one, embedded systems have been used in controlling power systems parameters such as voltage regulations [4], [5]. So, the embedded control system sequences efficiently integrate the programs' lower and upper-level classes. This procedure has been displayed as a flowchart in Figure 1. The elective courses and processes are shown with dashes, and the required courses are demonstrated with solid lines.

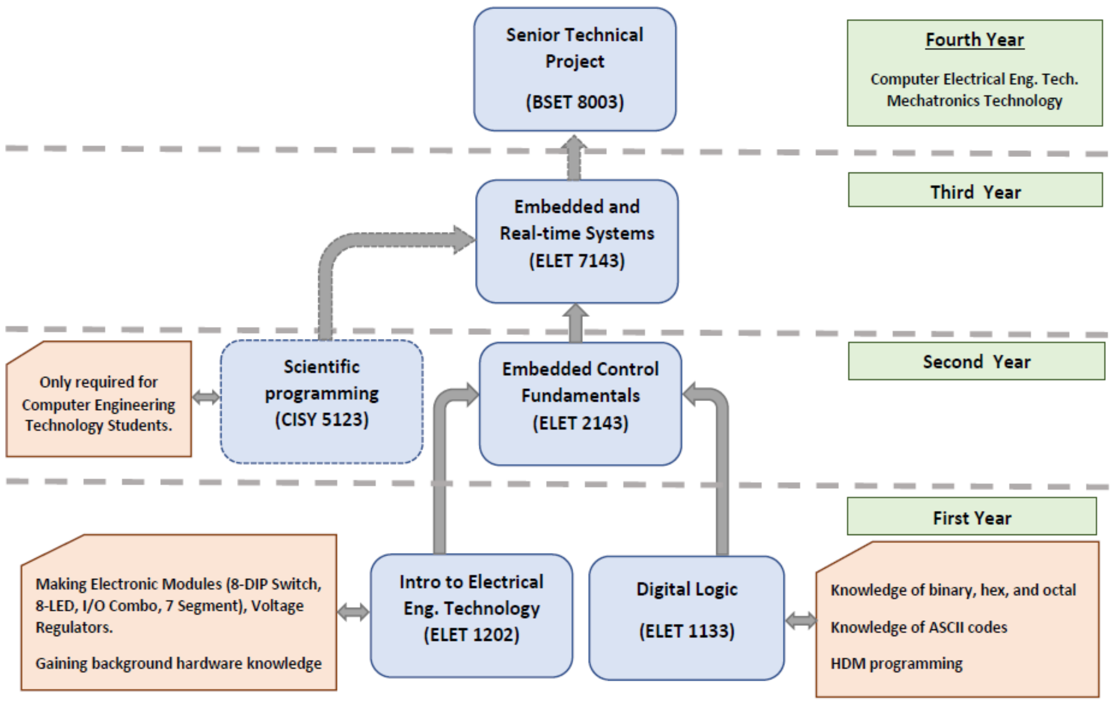


Figure 1. The location of Embedded courses in the curriculum

2. Course Description

The courses focus on studying and applying AVR, a widely used microcontroller, in embedded control systems. Arduino UNO is an inexpensive board built on the AVR ATMega328P microcontroller used in lab experiments and projects. This board's key features include onboard PWM, ADC, and UART. Furthermore, a software application can be developed by C and Assembly programming on Microchip Studio IDE or Arduino programming on Arduino IDE.

These courses have been in Electrical and Computer Engineering Technology curricula for over ten years. Initially, the two embedded courses were lab-focused, and lectures were designed to prepare the students for the lab experiments. In 2016, the courses were redesigned by adding two textbooks, digging deeply into the theory to create a strong foundation in AVR microcontroller architecture and programming. In addition, many lab experiments were added to help students understand the topics, deeply deploy their learnings and build up the knowledge required to design and implement their course project. These changes have been made based on the feedback from the Industrial Advisory Board and collaboration with other faculty members in the Department of Mechanical and Electrical Engineering Technology. Also, the graduates' comments on their exit survey have been considered in applying the changes. Some lab experiments that have been developed in these courses include:

- Coding PWM programs in AVR to control and operate a DC motor direction and speed.
- Programming ADC and DAC in AVR

The complete list has been included in Table 3. The applied learning funds have been used to equip an Embedded laboratory where Electrical Engineering devices such as oscilloscopes and power generators were installed. In addition, the students would be provided the AVR microcontroller educational kits and small lab parts like sensors and actuators for one semester. Also, A robot maze was installed in the lab to let students work on their projects during the semester.

The first embedded course is a lower-division three credits, including two lectures and three lab hours per week focusing on the microcontroller's hardware and software fundamentals. The second embedded course is upper-level four credits, including three lecture hours and three lab hours per week which concentrates on applications of embedded controllers. Some materials taught include interfacing digital and analog devices such as LEDs, distance sensors, and DC motors, and analog-to-digital and digital-to-analog data conversion. A significant focus of the second course will include the team design, construction, and testing of an embedded controller project. A typical project is a Mobile Autonomous Robotic System (MARS) that will participate in the school's annual robotic competition. The Assembly, C, and AVR programming languages are taught and used, although basic knowledge of C programming will help students in labs and course projects. The following sections describe the details of the two courses, one lab experiment, and a sample project.

3. First Course: Embedded Control Fundamentals

This course starts with the basics of Assembly programming language and introduces the architecture of AVR, a widely used 8 bits microcontroller. AVR is a RISC (reduced instruction set computer) microcontroller with an in-system programmer to create an engineering development system. This course focuses on the fundamentals of both hardware and software aspects of the AVR ATmega328P microcontroller. Lab experiments are designed to clarify the concepts of each topic and allow students to learn by doing.

Typical devices used in the lab are switches as digital controllers, light-emitting diodes (LEDs), and seven-segment displays as digital outputs. Students learn to interface these input/output modules by writing code in assembly language, assembling the program, and uploading their codes to an AVR Atmega328P microcontroller. In addition, they learn to control different parts of microcontrollers using registry programming. Structured programming using flow charts, well-written and organized coding, and step-by-step software and hardware troubleshooting are emphasized. This sophomore college-level course requires prerequisites, including Digital Logic and Introduction to Electrical Engineering Technology. Table 1 includes the topics of course material and their assigned lecture and lab hours.

Table 1: Topics covered in Embedded Control Fundamentals (First embedded course)

Topic	Lecture Hours	Lab Hours
Introduction to computing	1	
The AVR Microcontrollers: architecture, instruction, and Features	2	3
AVR Architecture and the fundamental concept of assembly language programming	2	3
Application of a text editor and an assembler	1	3
Application of Branch, Call, and Time Delay Loop in Assembly Language Programming	2	3
Flowcharting, structured programming, and top/down design	1	3
Addressing modes, conditional instructions, logic instructions, bit manipulation instructions	2	3
I/O port programming in AVR	2	6
AVR Advanced Assembly Programming Language including: Assembler directives Register and direct addressing modes Application of X, Y, and Z registers indirect addressing modes Lookup Table and Table processing using LPM (Load Program Memory) Instruction	8	6
AVR Hardware connection, hex files	1	3
Programming to interface devices such as switches, Light Emitting Diodes (LEDs), seven-segment displays, and pushbuttons.	2	6
Evaluation including quizzes, midterms	2	3
Total	28 (14 weeks)	42 (14 weeks)

A comprehensive textbook called “AVR Microcontroller and Embedded Systems: Using Assembly and C by Mazidi” was opted as the primary textbook [6]. The book is entirely covered through the two courses. While Assembly Programming and interfacing digital electronic devices are discussed in the first embedded course, the second course mainly focuses on C programming and interfacing analog electronic instruments. Students must purchase only an Arduino UNO R3 board for the lab experiments. In addition, an ATATMEL-ICE-BASIC - Debugger / Programmer Kit, including the debugger, USB 2.0 cable, and IDC flat cable, will be lent to them for the semester. The devices are shown in Figure 2.

Students also use the electronic lab modules they made in their Introduction to Engineering Technology course, a prerequisite for this course. For example, the modules in Figure 3 include 8 LED Output, 8 Position DIP Switch, Combination Input/Output, and a 7-segment Display [7]. They control the digital input and display the digital outputs by interfacing with the Arduino

Uno. The main benefit of using these electronic modules is avoiding extra wiring for the lab experiment and decreasing the chance of making errors.

Figure 2: Essential lab requirements for Embedded Control courses

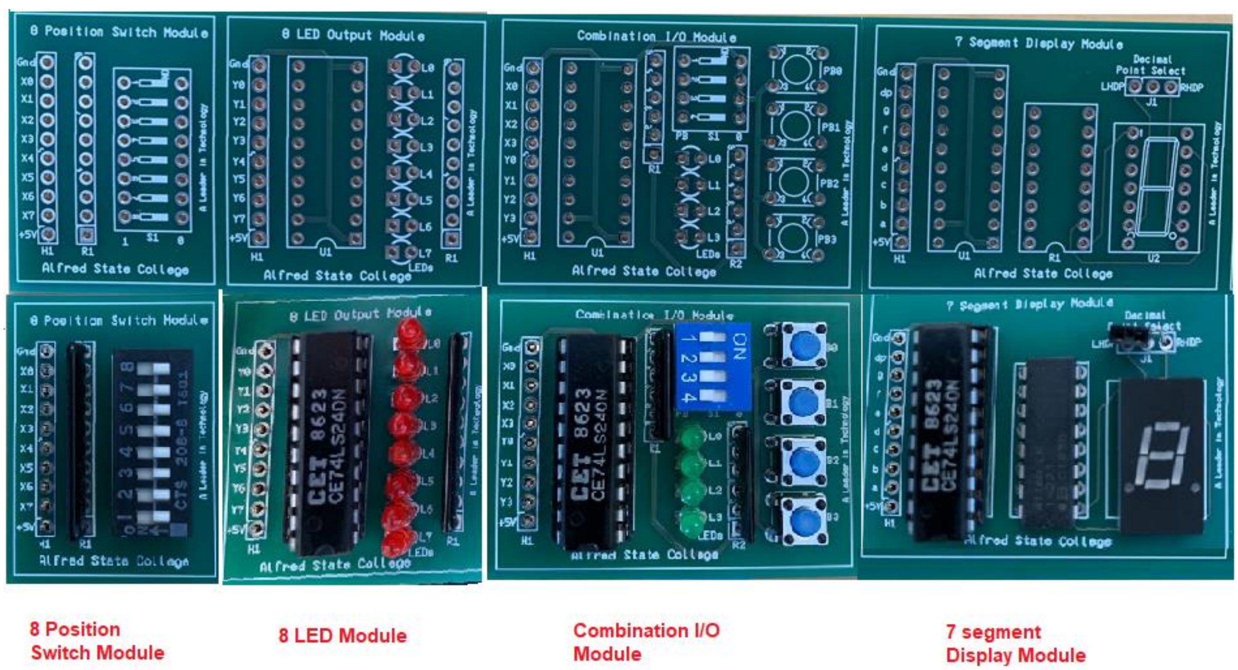


Figure 3. Electronic module to control the digital input and display the digital output on the AVR microcontroller before and after the soldering.

4. A Sample Lecture and Related Lab Taught in the Course

In advanced Assembly Programming, students learn the direct or indirect register addressing modes to access data stored in the data memory. Programming of register indirect addressing mode in Assembly is like programming Pointers in C Programming, which is challenging for some beginners. In the AVR, three registers are used for this purpose: X, Y, and Z. These are

16-bit registers allowing access to the entire 65,536 bytes of data memory space. Indirect addressing register mode is used to access all the data memory locations in AVR. So, the program can save the data in each one of the memory locations and restore the data by calling any one of the X, Y, and Z registers, which keeps the address to the assigned memory location.

To explain these topics to the student, a couple of labs are designed. One of the labs focuses on interfacing and programming a 7 X 5 Matrix LED Module, shown in Figure 4. This is a customized display module designed and developed in the fabrication lab based on the technology used in one cell of a regular LCD module[7]. There are 35 light-emitting diodes (LEDs) in the TIL305P display electronic, while there are not 35 I/O pins available in Arduino UNO, an eight bits microcontroller to interface each LED. To solve the problem, an old algorithm called raster scan which was used in the television displays, has been deployed in this program. The code switches on/off each row of LEDs briefly until all rows have been reached. Considering the frequency of the microcontroller, which is 16 Megahertz, the period of running each line of the code is a few nanoseconds that are invisible to the human eyes.

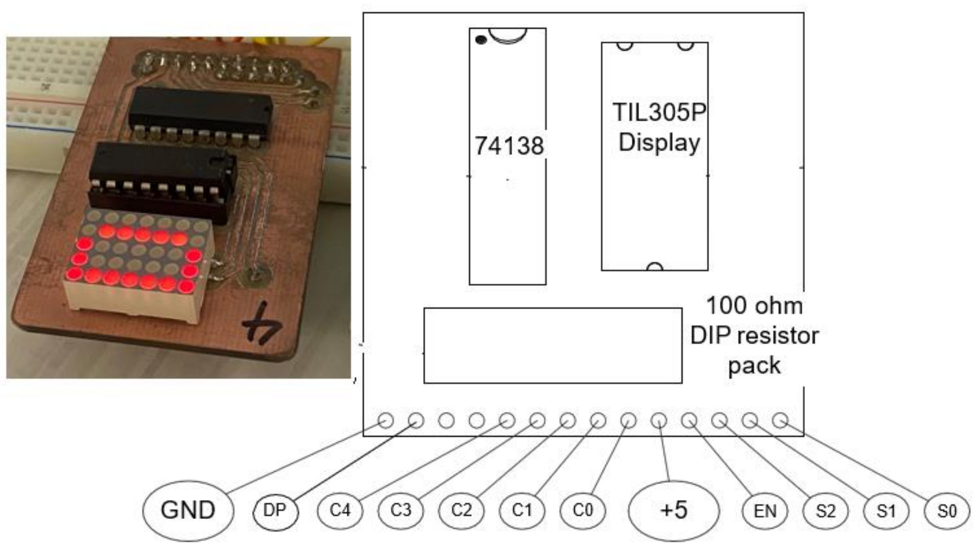


Figure 4. 7x5 Display Matrix designed and developed in ASC

The process repeats continuously for as long as desired to "display" the composite character. For each character to be displayed, we need to create a lookup table as below and make the letter using binary digits of 0 and 1 to show them off and on status in each LED. The last three bits (s0, s1, and s2) display row numbers, while the first 5 bits (c0-c4) will use for displaying characters. For example, in Figure 5, the letter M is made 0bc0c1c2c3c4s2s1s0 in a Matrix created in Excel; also, part of the Assembly code to canning LED using the indirect addressing and Z Register is shown.

		Column4	Column3	Column2	Column1	Column0			
		c4	c3	c2	c1	c0	s2	s1	0
Row0	0b	1	0	0	0	1	0	0	0,
Row1	0b	1	1	0	1	1	0	0	0,
Row2	0b	1	0	1	0	1	0	1	0,
Row3	0b	1	0	0	0	1	0	1	0,
Row4	0b	1	0	0	0	1	1	0	0,
Row5	0b	1	0	0	0	1	1	0	0,
Row6	0b	1	0	0	0	1	1	1	0

```
        ldi     ZH, high(first_character)           ; point to information to be displayed
        ldi     ZL, low(first character)
        lsl     ZL                                  ; shift [ZL] left for use with LPM instruction
        rol     ZH                                  ; rotate [ZH] left for use with LPM instruction

        ldi     row_counter, 7                      ;number of rows for each character

next_row:
        lpm     character_row, Z+                   ; get a character row and get ready for the next
        out     PORTD, character_row                ;display the character row
        (time delay – adjust for visual comfort)    ; one to ten millisecond scan delay
        dec     row_counter                         ; all character rows read from table?
        breq    read_pushbuttons                    ; if so, back to reading pushbuttons
        jmp     next_row                            ; otherwise, get next character row
```

Figure 5. Design and Assembly coding of the lookup table to scan the LED cells.

Figure 6 displays a setup in the lab for this experiment. Every student should create an Assembly code to receive control from the input push buttons and send a pattern to the 7x5 matrix display as explained in the following:

- Pressing PB1 will display your first initial. There is one millisecond time delay.
- Pressing PB2 will display your second initial (use X if you have no middle initial). There is one millisecond time delay.
- Pressing PB3 will display your third initial. There is one millisecond time delay.
- Pressing PB4 will display a cool character of your choice. Again, there is one millisecond time delay.

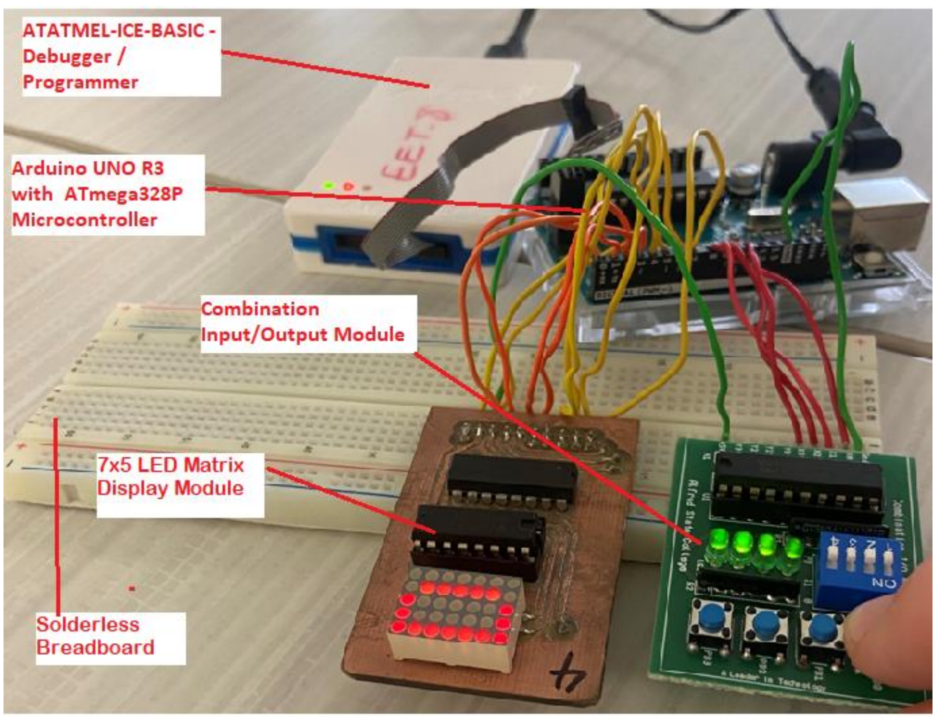


Figure 6. Hardware setup and displaying output after uploading code on the microcontroller.

5. Second Course: Application of Embedded Systems

This upper-level four-credit course, which includes three hours of lectures and three hours of lab per week, focuses on interfacing analog sensors and actuators to the AVR microcontroller using registry programming. Some of the student's learning outcomes are:

- Learning how to program in AVR programming and C.
- Communication with serial and parallel devices.
- AVR Programming to interface sensors and actuators such as switches, light-emitting diodes, seven-segment displays, Liquid Crystal Displays (LCDs), and D.C. motors.
- Interrupt-driven programming on data collected from analog sensors and converted to digital using ADC (analog to digital converter).
- An interrupt-driven program using pulse width modulation to control a D.C. motor.
- A typical project is a Mobile Autonomous Robotic System (MARS).

The main topics discussed in the course are shown in Table 2.

Table 2: Topics covered in Real-time and Embedded Systems (Second embedded course)

Lecture Topics	Lecture Hours	Lab Hours
1. AVR Programming in C	2	
2. AVR Timer Programming in Assembly and C	3	3
3. AVR Interrupt Programming in Assembly and C	2	3
4. AVR Serial Port Programming in Assembly and C	2	1.5
5. LCD Interfacing	1	3
6. ADC, DAC, and Sensor Interfacing	2	1.5
7. Relay, Optoisolator, and DC Motor Interfacing with AVR Input Capture and Wave Generation in AVR	2	3
8. PWM Programming and DC Motor Control in AVR	4	3
9. Techno Bot Control; Robot competition rules	1	
10. C++ Programming of Boarduino/Arduino	8	
11. Interfacing devices such as switches, light-emitting diodes, and seven segments display liquid crystal, D.C. motors, and Infrared Distance Sensors.	4	3
12. Fire Fighting Robot Construction and Control	2	21
13. Evaluation including presentation, test	3	
Total	42 (14 weeks)	42 (14 weeks)

Each week students have a related lab experiment to learn how to use their knowledge in practice and gradually learn to program the primary electronic devices used in their robot projects, such as distance sensors, DC motors, servo motors, and temperature sensors. By completing the first and the second-course labs halfway through the second course, students are ready to work in a group of three or four and assemble their knowledge to design and implement their course project. A list of the selected lab experiments has been added to Table 3. For each lab report, there is an assignment in the LMS that students must answer the targeted questions, such as which part of the microcontroller and which new programming instructions have been used in this lab. They must also upload a video of their working circuits and original codes.

Table 3: List of the selected lab experiments in two embedded courses

	Lab Experimental in both Embedded Courses	Part of the Microcontroller Programmed	Programming Language, Software	Main Hardware Used
1	Display an arbitrary data byte on LEDs connected to Port D	- General Purpose Registers (GPRs) in CPU - I/O Registers (Arithmetic Logic Unit (ALU) in CPU - Program Counter - Flash Memory - SRAM - X, Y, Z Registers for Indirect Addressing - Assembler Directives	- Assembly Language - Atmel AVR Assembler Microchip Studio for AVR® 7.0	- ATmega 328P installed on Arduino AVR - 8 LED Output Display Module - 7-Segment Output Display Module - Input/Output - Combination Module - Speakers
2	Reading data from switches connected to Port B and display on LEDs connected to Port D			
3	Counter with the display on 8-LED to Port D.			
4	Interfacing 7-Segment Module as a Counter			
5	Branches and Loops in Assembly			
6	Using software delay to send a pattern to the 7 Segment Display Module			
7	Interfacing 7X5 Matrix LED Module and Combination Module			
8	Generating music tones with software delay and Indirect addressing			
9	Creating Software Delay using Polling w/o interrupt	- All the upper parts are also used. - Timer/Counter Registers (both 8 and 16 digits) - Timer Interrupt Register - A/D Control and Status Registers	- C Programming Language - Atmel AVR GCC - Microchip Studio for AVR® 7.0 - Arduino programming language - Arduino IDE 2.0.4 - Tinkercad [8] Arduino Simulator - UnoArduSim [9]	- All the upper devices are also used. - Potentiometer - Liquid Crystal Display (LCD) - Servo Motors - DC motors - Infrared Distance Sensors - Ultrasonic Distance Sensor - TMP36 Temperature Sensor
10	Creating Software Delay using interrupt			
11	Generating Square Wave using AVR Timer Normal and CTC modes			
12	Analog-to-Digital Converter (ADC) and Digital Analog Converter (DAC)			
13	Pulse Width Modulation (PWM) using Timer/Counter Registers and PWM modes			
14	Serial Data Transfer by interfacing UART Register			
	Total			

6. Course Project: Design, Development, and Demonstration of Autonomous Firefighting Robot

The project is an autonomous firefighting robot designed based on Trinity College International Robot competition handbook. Before the event's cancelation due to the pandemic, the ASC IEEE and Robotic Club arranged the attendance of some groups to compete in the annual robot contest

[10]. In addition, at the end of each spring semester, the Colleges Robotics Club holds a local robotic competition that all the project groups must attend as part of their coursework. The robots must navigate a standard maze with four rooms in the shortest time, locate a candle, extinguish the candle, and pass obstacles such as a carpet and a stuffed animal. The winning groups receive a certificate and a cash prize.

The project has been broken into different activities, each having a portion of the grade. For example, in the final presentation, students explain their approach to the software and hardware design and receive feedback from the audience. A rubric for the project demonstration and presentation is shown in Table 4.

Table 4: A rubric for the second project presentation

Criteria	**Exemplary**	**Satisfactory**	**Needs Improvement**
Design procedure	30 points Flowcharts	20 points Explanation	10 points
Choosing programming application and software design method	30 points	20 points	10 points
Troubleshooting process	20 points A complete explanation of the troubleshooting process for software and hardware	15 points A full description of the troubleshooting process for part	10 points A general explanation
Time Scheduling (Gantt Chart)	20 points	15 points	10 points
Introduction, Conclusion, or smooth transition between slides	20 points	15 points	10 points
Teamwork	15 points	10 points	5 points
Informative slides	15 points	10 points	5 points
Clarity of presentation	15 points	10 points	5 points
Preparation in advance	15 points	10 points	5 points

Table 5: Summary of Operational Mode based on Trinity College Handbook

	Level 1		Level 2		Level 3	
Options (Below)/Divisions	Jr	Walk/HS/Sr	Jr	Walk/HS/Sr	Jr	Walk/HS/Sr
Sound Activation	N/A	Required	N/A	Required	N/A	Required
Furniture	Optional	Optional	Optional	Optional	Required	Required
Return Trip	Optional	Optional	Optional	Optional	Required	Required
Arbitrary Start	Optional	Optional	Optional	Optional	Required	Required
Non-Air	Optional	Optional	Optional	Optional	Required	Required
Candle Location	Optional	Optional	Required	Required	Required	Required

All the groups must complete level two of the operational mode in Table 5. As you can see, the robot must start arbitrarily and sound activated at a particular frequency. Most groups designed and implemented the sound activation PCB board in the fabrication lab, while some used 3D printers to create and print some parts of their robots. Part of the presentation from a group project is included in Figure 7.

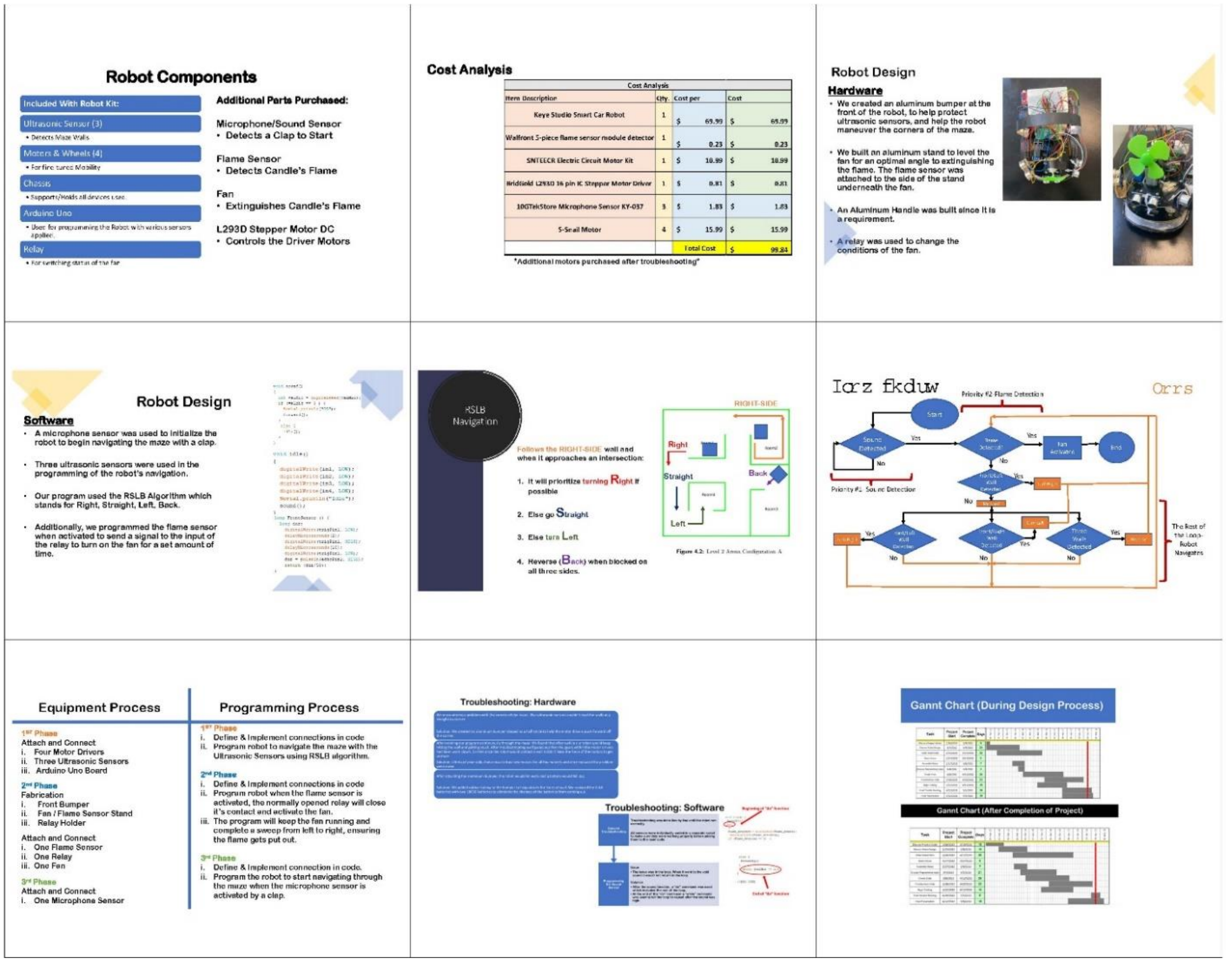


Figure 7. Part of a student's presentation

Some of the samples of students' work are shown in Figure 8. The different groups use different designs and programming languages. To increase the group's efficiency, each group has been made with a combination of Electrical, Computer, and Mechatronics Engineering students. So, the students work together in hardware and software parts and create stronger groups.

In addition, the student's annual robotic contest results have been occasionally published in the local media, which effectively promotes the Electrical and Computer Engineering programs. For example, in the following, you can see a competition report by Olean Times Herald staff on May 7th, 2021, called " College students participate in Firefighting Robot Competition." Figure 9 shows part of the article in the magazine [11]. Also, some groups have demonstrated their projects in the ASC student research showcase and SUNY Research Undergraduate Conference (SURC) through the years.

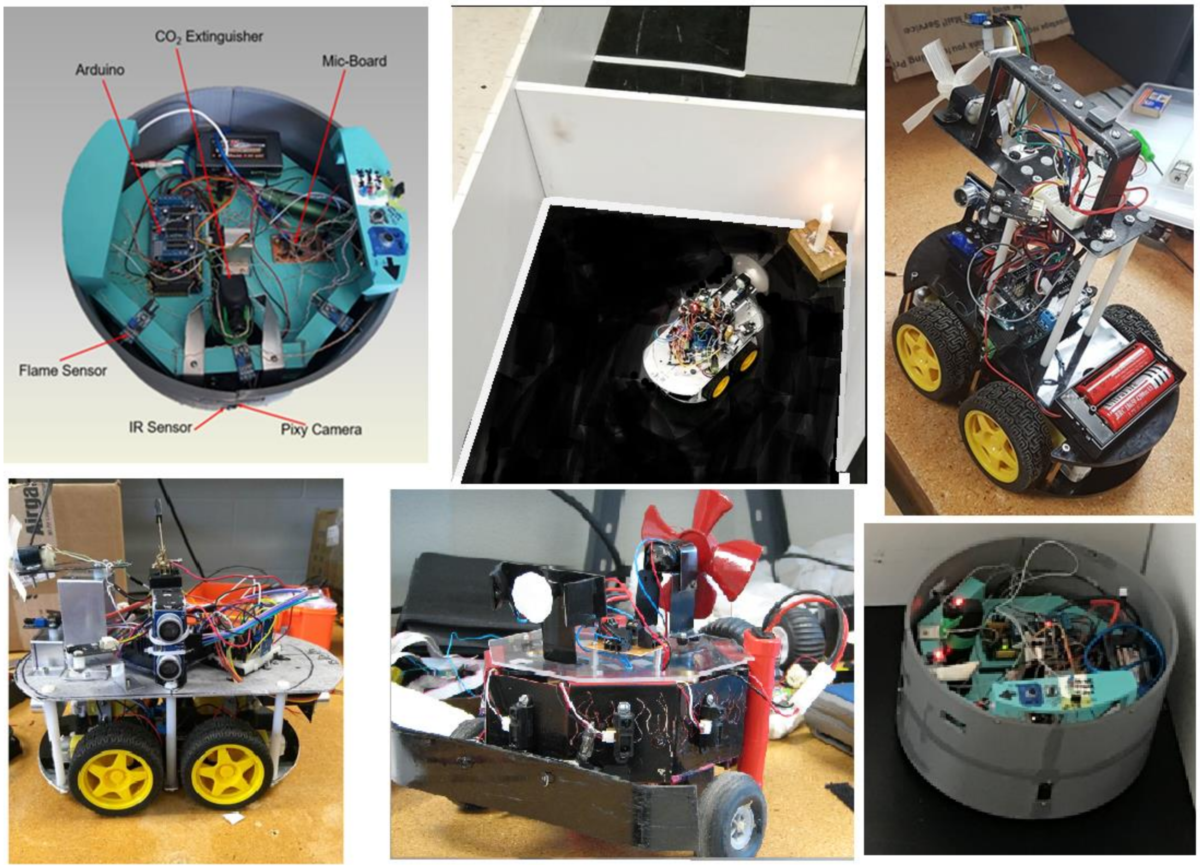


Figure 8. Some samples of students' projects were designed and developed as part of the course project.

Figure 9. Students compete in the college's annual Firefighting Robot Competition

7. Course Student Evaluation and Assessment for ABET

The embedded courses have become one of the critical technical aspects of Electrical and Computer Engineering Technology programs. The course assessment has been split into many areas and will help students learn the topics through theory, experiments, and projects. Also, all students with different strengths and weaknesses can get higher grades in their expertise. Figure 10 shows the grading policy for two courses.

Grading Policy

1. Homework, Quizzes 25%
2. Laboratory 25%
3. Midterms (Two) 30%
4. Final 20%

Grading Policy

1. Homework, Quizzes 15%
2. Laboratory Assignment and Reports (Group of two) 15%
3. Project Presentations and Demonstration, Poster (Group of 4)........................ 25%
4. Midterms (two) 25%
5. Final 20%

Figure 10. Grading Policy for Embedded Control Fundamentals (ELET 2143) and Embedded and Real-time Systems (ELET 7404)

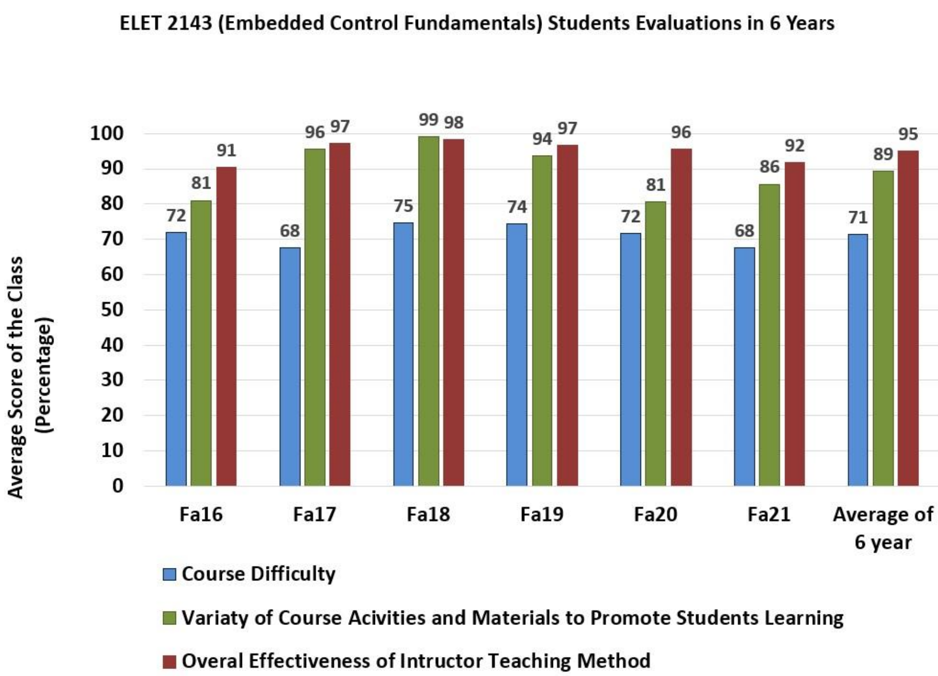


Figure 11. Student's evaluation of instructor and courses through the years for Embedded Control Fundaments (ELET 2143)

7.1. Student Evaluation of Embedded Courses

Students of both courses were anonymously surveyed on course materials and the instructor's teaching methods at the end of every semester. Figures 11 and 12 display three survey questions asking about the course level of difficulty, the variety of course activities and materials used to promote students' learning, and the effectiveness of instructor teaching methods. In the surveys, students regularly showed positive feedback with comments like "This course was one of their favorite classes in college."

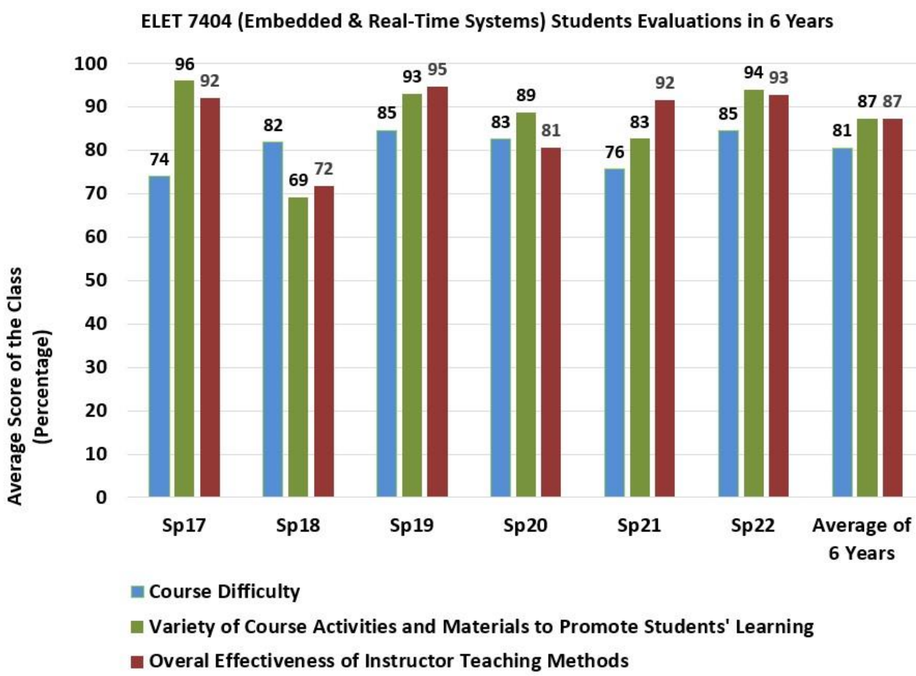


Figure 12. Student's evaluation of instructors and courses through the years for Embedded and Real-time Systems (ELET 7404).

Benchmark Scoring Guidelines	Score
Less than 60% of students score 75% or higher	0
60% of students score 75% or higher	1
70% of students score 75% or higher	2
80% of students score 75% or higher	3
90% of students score 75% or higher	4

Direct assessments that do not include student grading are assigned an achievement score by the MEET ABET committee

Figure 13. Benchmark scoring guidelines for the ABET evaluation.

7.1. Course Assessment based on the ABET Outcomes

On the other hand, regarding their wide range of course activities and topics, these two courses have been part of the program assessment every semester. Each ABET accreditation assessment for three programs of Electrical Engineering Technology, Mechanical Engineering Technology, and Mechatronics Engineering Technology included these courses in their maps. One of the assignments will be evaluated and scored to track the success rate of that program's outcome. Figure 13 shows the benchmark scoring guideline for each outcome to assess teaching success. For example, in Cycle 8 of the Computer Engineering Technology BS program, a programmatic assessment goal of an average score of "2 out of 4" was defined as the acceptance rate.
A summary of the following steps that have been taken with the program coordinators to assess the course outcomes regularly:

1. The program learning outcome map is created for six years of ABET visits. Later the whole period split into two three years cycles to cover all the program outcome criteria. They are displayed in Figure 14.

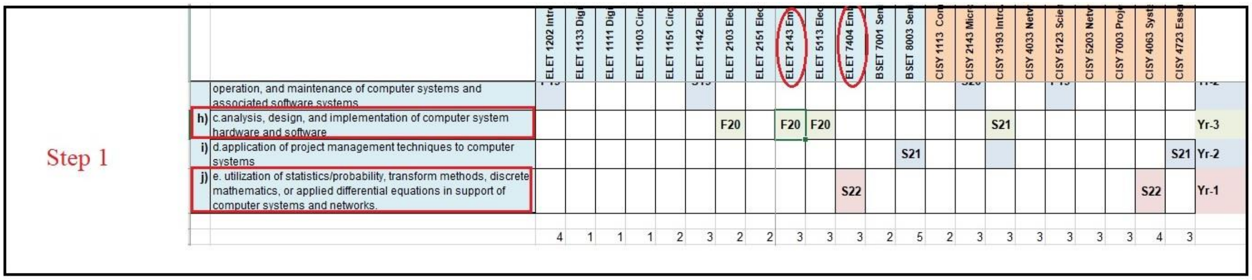


Figure 14. A screenshot of the Program Learning Map for Computer Engineering Technology in Cycle 8.

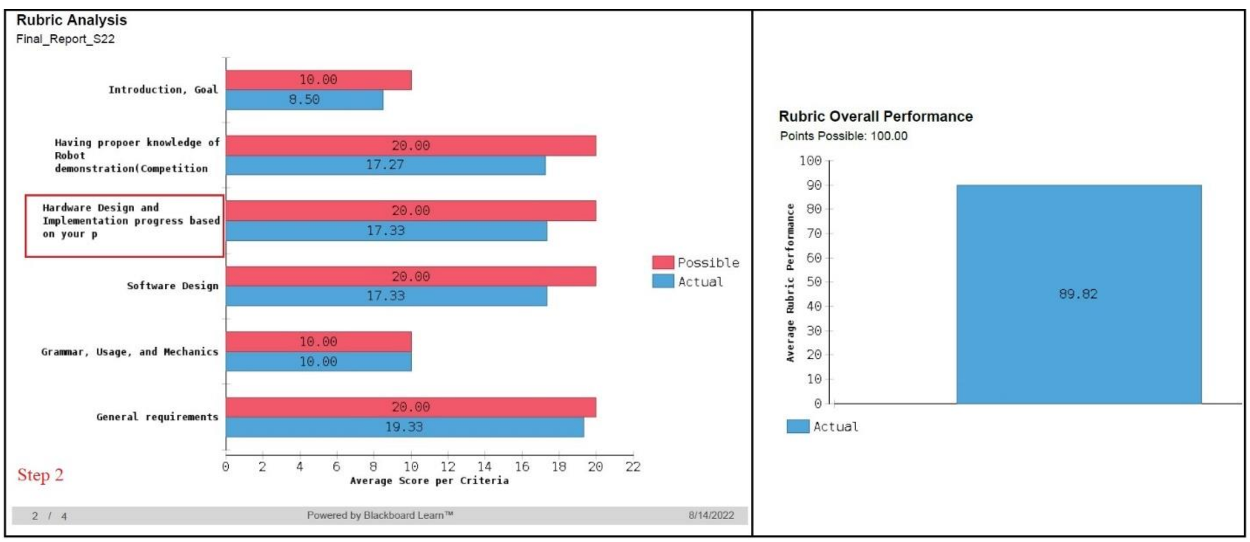


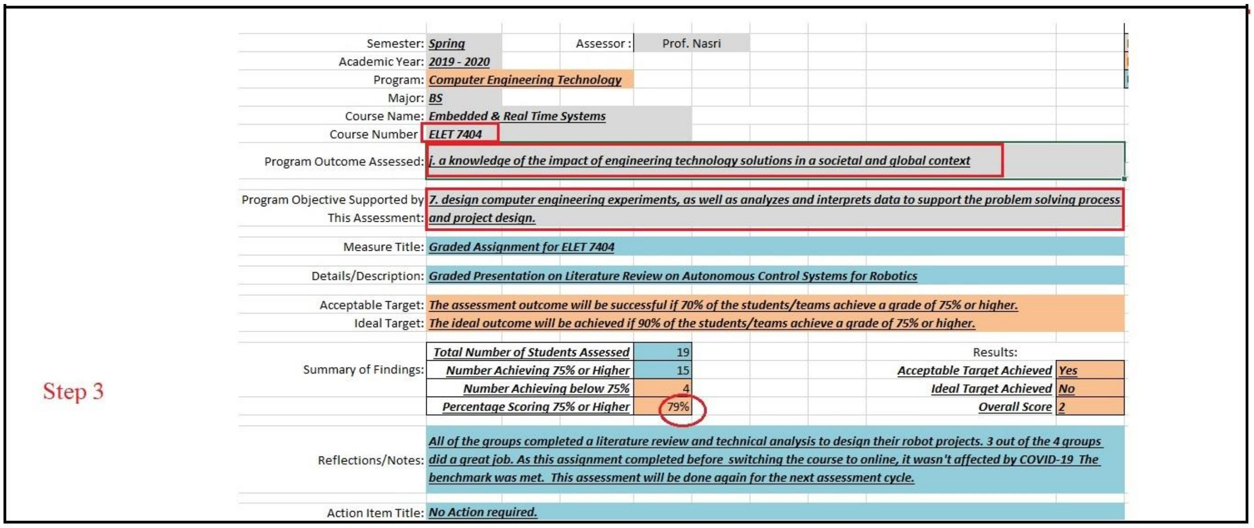
Figure 15. Rubric analysis of the final report for the Embedded Course Project in Spring 2022.

2. Program coordinators and course instructors meet at the beginning of each cycle to discuss the specific assignment in the course that will be used as a measurement. For example, part of the final project report assessed outcome j, which requires program graduates to have "a knowledge of the impact of engineering technology solutions in a

societal and global context." Also, the program Objective Outcome of 7, which indicates graduates should be able to "design computer engineering experiments, as well as analyzes and interprets data to support the problem-solving process and project design," has been supported by this assignment. Figure 15 shows a screenshot of the Rubric Statistic Report from Blackboard. Students have completed this assignment, including hardware and software designs and fabrications, by an average of 89.82% for the class.

3. The forms will be sent to the instructors, and they need to complete them at the end of the semester and return them, along with the assignment and three student sample works. A screenshot of the Excel form is provided in Step 3 of Figure 16.

4. The program coordinators will collect and analyze the results and enter the Watermark system to keep the electronic track of assessments.



Semester:	Spring — Assessor: Prof. Nasri
Academic Year:	2019 - 2020
Program:	Computer Engineering Technology
Major:	BS
Course Name:	Embedded & Real Time Systems
Course Number	ELET 7404
Program Outcome Assessed:	j. a knowledge of the impact of engineering technology solutions in a societal and global context
Program Objective Supported by This Assessment:	7. design computer engineering experiments, as well as analyzes and interprets data to support the problem solving process and project design.
Measure Title:	Graded Assignment for ELET 7404
Details/Description:	Graded Presentation on Literature Review on Autonomous Control Systems for Robotics
Acceptable Target:	The assessment outcome will be successful if 70% of the students/teams achieve a grade of 75% or higher.
Ideal Target:	The ideal outcome will be achieved if 90% of the students/teams achieve a grade of 75% or higher.

Summary of Findings:

Total Number of Students Assessed	19	Results:	
Number Achieving 75% or Higher	15	Acceptable Target Achieved	Yes
Number Achieving below 75%	4	Ideal Target Achieved	No
Percentage Scoring 75% or Higher	79%	Overall Score	2

Reflections/Notes: All of the groups completed a literature review and technical analysis to design their robot projects. 3 out of the 4 groups did a great job. As this assignment completed before switching the course to online, it wasn't affected by COVID-19 The benchmark was met. This assessment will be done again for the next assessment cycle.

Action Item Title: No Action required.

Figure 16. A screenshot of the course evaluation form completed by the instructor for Computer Engineering Technology in Cycle 8 of the assessment.

As shown in Figure 17, the assessment analysis results from different assignments used as measurements for the three Engineering Technology Programs at the BS level, and there is a consistent level of success. Figure 18 summarizes the analysis results of direct assessment for three Engineering Technology BS programs: Electrical Engineering, Computer Engineering, and Mechatronics Systems, in 2021-2022. The graphs show that all three programs have an average above 75%, an acceptable target for ABET accreditation.

In addition, in the student's senior projects survey, more than 75% of the graduates in EET and CpET programs mentioned these two courses among the few courses that helped them to design and implement their senior projects.

The three Associate programs in Engineering technology also use the first embedded course as a solid evaluation for their accreditation goals. However, this article has only reviewed three Bachelor's degrees mentioned above.

Direct Assessments for BS Degrees (Electrical Eng Tech:EET, Computer Eng Tech: CpET, Mechatronics Eng Tech: McET)									
Program/Course Assess	ABET Outcome	PEO #	Assessment Period	ABET Cycle #	Semester Assessed	Number of Students Assessed	Percentage Achieving 75% or Higher (Accepted Target)	Acceptable Target Achieved	Assessment Assignment
EET BS / ELET 2143	c *	2 ^	2018-2024	8	Fa21	15	76%	Yes	Graded Lab Report for a Group Assignment on 7-Segment Displays Programmed by Assembly
EET BS/ ELET 7404	j **	7 ^^	2018-2024	8	Sp22	12	82%	Yes	Graded Exam Question on PWM Programming
CpET BS/ ELET 2143	h ***	2 ^^^	2018-2024	7	Fa21	11	89%	Yes	Graded Lab Group Assignment on Interfacing a 7x5 Matrix and Push-buttons as output/input
CpET BS/ ELET 7404	j ****	7 ^^^^	2018-2024	7	Sp22	9	79%	Yes	Graded Final Report on Literature Review, Design, and Fabrication of Autonomous Firefighting Robot
McET BS/ ELET 2143	g *****	3 ^^^^^	2021-2024	1	Fa22	6	92%	Yes	Graded Lab Group Assignment on Programming a speaker to play music tunes using Advanced Assembly

* c) (3) an ability to apply written, oral, and graphical communication in broadly defined technical and non-technical environments; and an ability to identify and use appropriate technical literature.
** j) (e) the ability to utilize differential and integral calculus, as a minimum, to characterize the performance of electrical/electronic systems.
*** h) (c) an understanding of the need for and an ability to engage in self-directed continuing professional development
**** j) (e) a knowledge of the impact of engineering technology solutions in a societal and global context
***** g) (b) application of circuit analysis, analog and digital electronics, basic instrumentation, associated software, and computers to aid in the characterization, analysis, and troubleshooting of electromechanical systems
^ 2. Produce graduates who function professionally with effective communication and with ethical responsibility as individuals and as members of a multidisciplinary team
^^ 7. produce graduates who design electrical engineering experiments, as well as analyze and interpret data to support the problem solving process and project design
^^^ 2. can function effectively as technicians in the Computer or related field of engineering technology
^^^^ 7. design computer engineering experiments, as well as analyzes and interprets data to support the problem solving process and project design
^^^^^ 3. Function professionally, with effective communication, as an individual and as a member and /or leader within a multidisciplinary team.

EET: Electrical Engineering Technology Program — ELET 2143: Embedded Controller Fundamentals
CpET: ComputerEngineering Technology Program — ELET 7404: Embedded & Real-Time Systems
McET: Mechatronics Engineering Technology Program

Figure 17. Direct Assessment assignments and results for three Engineering Technology BS programs: Electrical, Computer, and Mechatronics in 2021-2022

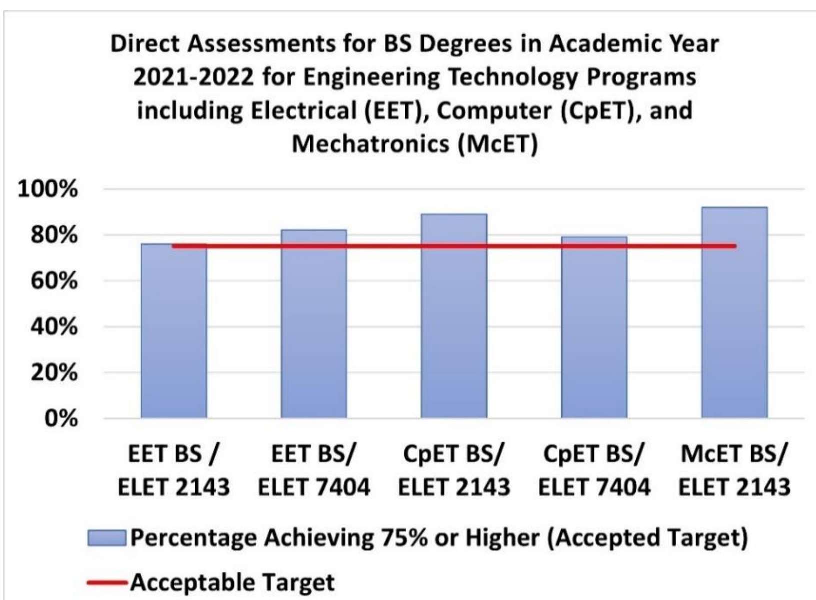


Figure 18. Analysis of Direct Assessment assignments results for three Engineering Technology BS programs: Electrical, Computer, and Mechatronics in 2021-2022

8. Conclusion

In this paper, a successful method of teaching a two-semester embedded course has been presented. The sequence courses are required and cover seven credits of the Electrical Engineering Technology and Computer Engineering Technology students receiving a BS degree. The first embedded course helps students learn the AVR microcontroller architecture and Assembly programming through theory and lab experiments. Students in this course deploy the knowledge and the product of the devices they have made in lower-level course labs such as Digital Logic and Intro to Electrical Engineering.

In the second embedded course, students learn more advanced topics in AVR architecture and Arduino and C programming. Students design and develop an autonomous firefighting robot as their course project, along with the lab experiments. The project helps students be exposed to real-world problems and allows them to integrate what they have learned in both courses by designing and implementing an embedded project. Also, doing the course project prepares for their upper-division courses, such as the Senior Technical Project. So, the embedded system courses create a bridge between lower and upper-level classes. The paper has presented the course topics, a sample lab, and a sample student's project.

In addition, the Accreditation Board of Engineering and Technology (ABET) guidelines in accrediting engineering technology programs have been followed in both courses. As described thoroughly in the paper, different course assignments have been assessed regularly to evaluate the program outcomes and have mainly achieved the accepted target defined by the ABET committee. The student evaluation results also have been discussed in this article. Despite the high difficulty level, student feedback confirmed the presented method's effectiveness in enhancing students' understanding and ability to apply embedded systems design concepts to solve real-world engineering problems.

References:

[1] Dublin, March 26, 2020 (GLOBE NEWSWIRE) -- The "Embedded System Market by Hardware (MPU, MCU, Application-specific Integrated Circuits, DSP, FPGA, and Memories), Software (Middleware, Operating Systems), System Size, Functionality, Application, Region - Global Forecast to 2025". Available: ResearchAndMarkets.com.

[2] Y.M. Roshan and M. Moallem, "Developing a course and laboratory for embedded control of mechatronic systems," Proceedings of ASEE Annual Conference, 2011.

[3] Sababha, B. H., Alqudah, Y. A., Abualbasal, A., & AlQaralleh, E. A. (2016). Project-based learning to enhance teaching embedded systems. Eurasia Journal of Mathematics, Science & Technology Education, 12(9), 2575-2585.

[4] Agrawal, Jai P., Omer Farook, and Abdulhaleem Abdullah Alowaisheer. "Embedded System Design for Sun-tracking Solar Panels: A Class Project Template." In 2012 ASEE Annual Conference & Exposition, pp. 25-515. 2012.

[5] Nasri, Maryam, Herbert L. Ginn, III, and Mehrdad Moallem. 2021. "Agent-Based Coordinated Control of Power Electronic Converters in a Microgrid" Electronics 10, no. 9: 1031.

[6] AVR Microcontroller and Embedded Systems: Using Assembly and C; Mazidi, Naimi & Naimi, ©2017, ISBN-13: 978-0997925968

[7] Designed and developed by David Hunt at Alfred State College for Introduction to Electrical Eng. Technology Course

[8] Tinker CAD was developed by Autodesk in 2011. Available: https://www.tinkercad.com/

[9] UnoArduSim was developed by Professor Simmons of Queen's University, Ontario, Canada.

[10] Trinity College Fire-Fighting Home Robot Contest 2020 Rules, Last Updated: November 7, 2019. Available: https://trinityrobotcontest.org/

[11] By Olean Times Herald staff, " Alfred State students participate in Firefighting Robot Competition," Olean Times Herald, May 7, 2021. Available: https://www.oleantimesherald.com/news/allegany_county/alfred-state-students-participate-in-firefighting-robot-competition/article_4df1877e-76dd-5ec7-a79d-75ee7c8bff24.html