EET-2242: C AND ASM PROGRAMMING WITH EMBEDDED APPLICATIONS

Cuyahoga Community College

Viewing: EET-2242 : C and ASM Programming with Embedded Applications

Board of Trustees:
June 2019

Academic Term:
Fall 2019

Subject Code
EET - Electrical/Electronic Engineer

Course Number:
2242

Title:
C and ASM Programming with Embedded Applications

Catalog Description:
Introduces microprocessor and microcontroller internal and external hardware components. Assembly language (ASM) programming is introduced to illustrate the internal working of a microcontroller. The C programming language is taught in a regular (C++) and embedded environment.

Credit Hour(s):
3

Lecture Hour(s):
2

Lab Hour(s):
2

Requisites

Prerequisite and Corequisite
EET-1241 Digital Fundamentals, or departmental approval.

I. ACADEMIC CREDIT

Academic Credit According to the Ohio Department of Higher Education, one (1) semester hour of college credit will be awarded for each lecture hour. Students will be expected to work on out-of-class assignments on a regular basis which, over the length of the course, would normally average two hours of out-of-class study for each hour of formal class activity. For laboratory hours, one (1) credit shall be awarded for a minimum of three laboratory hours in a standard week for which little or no out-of-class study is required since three hours will be in the lab (i.e. Laboratory 03 hours). Whereas, one (1) credit shall be awarded for a minimum of two laboratory hours in a standard week, if supplemented by out-of-class assignments which would normally average one hour of out-of-class study preparing for or following up the laboratory experience (i.e. Laboratory 02 hours). Credit is also awarded for other hours such as directed practice, practicum, cooperative work experience, and field experience. The number of hours required to receive credit is listed under Other Hours on the syllabus. The number of credit hours for lecture, lab and other hours are listed at the beginning of the syllabus. Make sure you can prioritize your time accordingly. Proper planning, prioritization and dedication will enhance your success in this course.

The standard expectation for an online course is that you will spend 3 hours per week for each credit hour.

II. ACCESSIBILITY STATEMENT

If you need any special course adaptations or accommodations because of a documented disability, please notify your instructor within a reasonable length of time, preferably the first week of the term with formal notice of that need (i.e. an official letter from the Student Accessibility Services (SAS) office). Accommodations will not be made retroactively.
For specific information pertaining to ADA accommodation, please contact your campus SAS office or visit online at https://www.tri-c.edu/student-accessibility-services/. Blackboard accessibility information is available at http://access.blackboard.com.

Eastern (216) 987-2052 - Voice. (216) 987-2423 - Fax
Metropolitan (216) 987-4344 – Voice.
(216) 987-3257 - Fax.
Western (216) 987-5079 – Voice. (216) 987-5118 - Fax.
Westshore (216) 987-3900 – Voice. (216) 987-5294 - Fax.
Brunswick (216) 987-5079 – Voice. (216) 987-5118 - Fax.
Off-Site (216) 987-5079 – Voice

### III. ATTENDANCE TRACKING

Regular class attendance is expected. Tri-C is required by law to verify the enrollment of students who participate in federal Title IV student aid programs and/or who receive educational benefits through other funding sources. Eligibility for federal student financial aid is based in part on enrollment status.

Students who do not attend classes for the entire term are required to withdraw from the course(s). Additionally, students who withdraw from a course or stop attending class without officially withdrawing may be required to return all or a portion of their financial aid based on the date of last attendance. Students who do not attend the full session are responsible for withdrawing from the course(s).

Tri-C is responsible for identifying students who have not attended a course before financial aid funds can be applied to students’ accounts.

Therefore, attendance is recorded in the following ways:

- For in-person and blended-learning courses, students are required to attend the course by the 15th day of the semester (or equivalent for terms shorter than five weeks) to be considered attending. Students who have not met all attendance requirements for in-person and blended courses, as described herein, within the first two weeks or equivalent, will be considered not attending.
- For online courses, students are required to login at least two times per week and submit one assignment per week for the first two weeks of the semester, or equivalent to the 15th day of the term. Students who have not met all attendance requirements for online courses, as described herein, within the first two weeks or equivalent, will be considered not attending.

At the conclusion of the first two weeks of a semester or equivalent, instructors report any registered students who have “Never Attended” a course. Those students will be administratively withdrawn from that course. However, after the time period in the previous paragraphs, if a student stops attending a class or wants or needs to withdraw, for any reason, it is the student’s responsibility to take action to withdraw from the course. Students must complete and submit the appropriate Tri-C form by the established withdrawal deadline.

Tri-C is required to ensure that students receive financial aid only for courses that they attend and complete. Students reported for not attending at least one of their registered courses will have all financial aid funds held until confirmation of attendance in registered courses has been verified. Students who fail to complete at least one course may be required to repay all or a portion of their federal financial aid funds and may be ineligible to receive future federal financial aid awards. Students who withdraw from classes prior to completing more than 60 percent of their enrolled class time may be subject to the required federal refund policy.

If illness or emergency should necessitate a brief absence from class, students should confer with instructors upon their return. Students having problems with coursework due to a prolonged absence should confer with the instructor or a counselor.

### IV. LEARNING OUTCOMES ASSESSMENT

Occasionally, in addition to submitting assignments to their instructors for evaluation and a grade, students will also be asked to submit completed assignments, called ‘artifacts,’ for assessment of course and program outcomes and the College’s Essential Learning Outcomes (ELOs). The artifacts will be submitted in Blackboard or a similar technology. The level of mastery of the outcome demonstrated by the artifact DOES NOT affect the student’s grade or academic record in any way. However, some instructors require that students submit their artifact before receiving their final grade. Some artifacts will be randomly selected for assessment, which will help determine improvements and support needed to further student success. If you have any questions, please feel free to speak with your instructor or contact the Learning Outcomes Assessment office.

### V. CONCEALED CARRY STATEMENT

College policy prohibits the possession of weapons on college property by students, faculty and staff, unless specifically approved in advance as a job-related requirement (i.e., Tri-C campus police officers) or, in accordance with Ohio law, secured in a parked vehicle in a designated parking area only by an individual in possession of a valid conceal carry permit.

As a Tri-C student, your behavior on campus must comply with the student code of conduct which is available on page 29 within the Tri-C student handbook, available at http://www.tri-c.edu/student-resources/documents/studenthandbook.pdf You must also comply with the College’s Zero Tolerance for Violence on College Property available at http://www.tri-c.edu/policies-and-procedures/documents/3354-1-20-10-zero-tolerance-for-violence-policy.pdf
VI. CORONAVIRUS/COVID-19 STATEMENT

Students are responsible for adhering to all College health and safety guidance, including that which relates to the COVID-19 pandemic. Public health requirements and standards are changing rapidly, and the College is adapting its guidance accordingly. Please check your Tri-C email and visit tri-c.edu/coronavirus regularly for updates. All students must adhere to the following general guidelines, until further notice:

• Remain at home. Do not attend any in-person class or gathering.
• Notify your instructor(s) if you are ill, have tested positive for COVID-19, or have been exposed to an individual who has tested positive for COVID-19.
• Wear a mask or face covering at all times, including, but not limited to: upon entering and exiting any Tri-C facility, in class, and in all common areas.
• Maintain a distance of at least six feet between yourself and others at all times.
• Provide the College with relevant information about your current health status and participate in any required on-site checks (e.g., temperature checks).
• Use only designated areas of Tri-C facilities, including entrances and exits. Sign in and out of Tri-C facilities as directed.

The general guidelines listed above do not encompass all coronavirus-related guidance. These guidelines are subject to change at the discretion of the College and under the direction of public health authorities. Students who fail to adhere to this guidance may be subject to disciplinary action under the College’s Student Code of Conduct and the Student Judicial Code.

Outcomes

Course Outcome(s):
Identify and explain the principal components in a microcontroller’s core.

Essential Learning Outcome Mapping:
Not Applicable: No Essential Learning Outcomes mapped. This course does not require application-level assignments that demonstrate mastery in any of the Essential Learning Outcomes.

Objective(s):
1. Determine the program address space of the microcontroller trainer.
2. Determine the writable address space of the microcontroller trainer (usually referred to as Random Access Memory, RAM).
3. Explain memory-mapped Input/Output and show its address space.
4. List the special functions registers and explain what they do.
5. List several arithmetic instructions.
6. List several logic instructions.
7. Explain what ‘flags’ are, how they are set/cleared/not-affected and why they are a necessity for a program.

Course Outcome(s):
Demonstrate the three principal components of a programming language, in this case the C programming language.

Essential Learning Outcome Mapping:
Not Applicable: No Essential Learning Outcomes mapped. This course does not require application-level assignments that demonstrate mastery in any of the Essential Learning Outcomes.

Objective(s):
1. Write and test programs using selector statements that includes if, if-else and case (switch) statements.
2. Write and test program that use loop statements that in while, do-while and for statements.
3. Use process statements and explain how their execution is controlled by selector and loop statements.

Course Outcome(s):
Demonstrate the Integrated Development Environment (IDE or a similar tool) using various components.

Essential Learning Outcome Mapping:
Not Applicable: No Essential Learning Outcomes mapped. This course does not require application-level assignments that demonstrate mastery in any of the Essential Learning Outcomes.

Objective(s):
1. In the IDE (Independent Development Environment), create a new project and specify the family and part that the project will use.
2. Create a source program in C and add it to the project.
3. Set the IDE’s communication system to transfer data to/from the target microcontroller.
4. Compile a program.
5. Correct syntax errors that the compiler identifies.
6. Link and download a program to the target microcontroller.
7. Demonstrate and use the IDE’s debugger functions that include breakpoints, single step, register contents, variable contents, etc.
8. Find and correct run time errors, if present.

**Course Outcome(s):**
Write a program that demonstrate interrupts.

**Essential Learning Outcome Mapping:**
Not Applicable: No Essential Learning Outcomes mapped. This course does not require application-level assignments that demonstrate mastery in any of the Essential Learning Outcomes.

**Objective(s):**
1. Set up the interrupt control registers and write an Interrupt Service Routine (ISR) for a timer interrupt or port change interrupt and using breakpoints or other identification methods, demonstrate the ISR is entered.
2. Display the physical starting address of an ISR, find the corresponding address in the vector space and show that the ISR’s address in in the proper vector space.

**Course Outcome(s):**
Write simple programs, view and explain key parts of the ASM (Assembly Language) code produced by a C compiler.

**Essential Learning Outcome Mapping:**
Not Applicable: No Essential Learning Outcomes mapped. This course does not require application-level assignments that demonstrate mastery in any of the Essential Learning Outcomes.

**Objective(s):**
1. Write a small C program with loop, selector and process statements.
2. View the ASM output file from the compiler and identify the comments C statements and the ASM statements that result.
3. The find condition statements in a C compiler’s ASM output file.

**Course Outcome(s):**
Using embedded C, demonstrate the use of GPIO, read data from ports and write data to ports.

**Essential Learning Outcome Mapping:**
Not Applicable: No Essential Learning Outcomes mapped. This course does not require application-level assignments that demonstrate mastery in any of the Essential Learning Outcomes.

**Objective(s):**
1. Configure GPIO (General Purpose Input/Output) ports for input and demonstrate how data is transferred to a variable.
2. Specify the minimum and maximum permissible voltages for a logic 0 and a logic 1.
3. Configure GPIO ports for output and transfer data from a variable to an output port and prove that the correct data is on the port.
4. Using the microcontroller’s data specification, determine the maximum source and sink currents.

**Course Outcome(s):**
Create design documents.

**Essential Learning Outcome Mapping:**
Not Applicable: No Essential Learning Outcomes mapped. This course does not require application-level assignments that demonstrate mastery in any of the Essential Learning Outcomes.

**Objective(s):**
1. Create a flow chart for a program in the design stage.
2. Create a state table for a program in the design stage.
3. Create a pseudo code for a program in the design stage.
Course Outcome(s):
Demonstrate C programming in a non-embedded application.

**Essential Learning Outcome Mapping:**
Not Applicable: No Essential Learning Outcomes mapped. This course does not require application-level assignments that demonstrate mastery in any of the Essential Learning Outcomes.

**Objective(s):**
1. Write various programs using selector, looping and control statement and generate a printout and verify that the problem statement has been solved.
2. Read and write data from a program to a disc file.
3. Read data from the keyboard for different types of variables and perform an operation on the data that will solve a problem statement.
4. Solve numerical methods problems that associate with electronics.

---

Course Outcome(s):
Using embedded C, demonstrate analog-to-digital conversion (ADC) and digital-to-analog conversion (DAC).

**Essential Learning Outcome Mapping:**
Not Applicable: No Essential Learning Outcomes mapped. This course does not require application-level assignments that demonstrate mastery in any of the Essential Learning Outcomes.

**Objective(s):**
1. Configure the ADC's special function register and connect a voltage of known value and write the code to display the result in base 10.
2. Read the ADC's output and write the code and convert to base 16.
3. Read the ADC's output and write the code to convert to base 2.

---

Course Outcome(s):
Using embedded C, demonstrate communications using the Universal Asynchronous Receiver Transmitter (UART) by using the associated special functions register.

**Essential Learning Outcome Mapping:**
Not Applicable: No Essential Learning Outcomes mapped. This course does not require application-level assignments that demonstrate mastery in any of the Essential Learning Outcomes.

**Objective(s):**
1. Configure the micro controller's special and set up a UART and write a program that sends data, loops back the data at the microcontroller's internal loop back, and then verify proper data.
2. Remove the internal loop back and physically loop back the transmit and receive pin and verify the data.

---

Course Outcome(s):
Using embedded C, demonstrate the use of counters and timers.

**Essential Learning Outcome Mapping:**
Not Applicable: No Essential Learning Outcomes mapped. This course does not require application-level assignments that demonstrate mastery in any of the Essential Learning Outcomes.

**Objective(s):**
1. Configure a timer control register to timeout and generate a specific periodic signal.
2. Configure a timer control register to count transitions on a port pin.

---

Course Outcome(s):
Using embedded C, demonstrate communications using Inter integrated Circuit (I2C) or Serial Peripheral interconnect (SPI).

**Essential Learning Outcome Mapping:**
Not Applicable: No Essential Learning Outcomes mapped. This course does not require application-level assignments that demonstrate mastery in any of the Essential Learning Outcomes.
Objective(s):
1. Using the microcontroller’s special functions register, write a program to send/receive data to/from an I2C or SPI peripheral and verify that the data is received at whatever is the receiving device (microcontroller or peripheral).
2. Use a scope to display and explain the clock and data waveforms.

Methods of Evaluation:
1. Examinations
2. Quizzes
3. Lab reports
4. Homework assignments

Course Content Outline:
A. Accumulator and register type microcontrollers
B. Code address space
C. Data address space
D. Memory-mapped I/O
E. Special function registers
F. Arithmetic and logic unit
G. Flags
H. Independent Development Environment
I. ASM (Assembly Language) and C source files
J. Assembling and compiling
K. Debugging
L. Writing programs
M. Components of a computer language
N. ASM output from C compiler
O. Design documents
P. GPIO
Q. Timers
R. Counters
S. I2C
T. SPI
U. UART
V. ADC
W. DAC
X. ISR
Y. Disk files
Z. Using standard I/O in C

Resources


Resources Other
1. Lab packet Provided in electronic form
2. Handouts including instructor authored notes and topic summaries, current journal articles, and manufacturer's literature.

Top of page
Key: 1665