ERAU–Daytona Beach

Department of Electrical, Computer, Software, and Systems Engineering

CEC 320: Microprocessor Systems

General Information

Fall 2021 (Rev. 1.0, 8/07/2021)

# General Info

Class time: MWF 11:15 am to 12:05 pm

Room: LB 269

Instructor: Dr. Jianhua Liu, Associate Professor of Electrical and Computer Engineering

Office: LB 349

E-mail: [jianhua.liu@erau.edu](mailto:jianhua.liu@erau.edu)

Phone: (386) 226–7713 (Work)

Office hours: MWF 3:00 pm to 4:30 pm; TBD on Zoom. There are also tutoring hours from the TAs of the labs.

# Prerequisites and Corequisites

Prerequisite: CEC 220. Corequisite: CEC 322.

# Required Textbook

Yifeng Zhu, *Embedded Systems with ARM Cortex-M Microcontrollers in Assembly Language and C*, Third Edition*,* E-Man Press LLC, 2017.

# Reference Books

* ARM, *Cortex-M4 Devices---Generic User Guide*, 2011. Download available at <http://infocenter.arm.com/help/topic/com.arm.doc.dui0553a/DUI0553A_cortex_m4_dgug.pdf>
* Joseph Yiu, *The Definitive Guide to ARM Cortex-M3 and Cortex-M4 Processors*, Third Edition*, Newnes*, 2014.

# Suggested Reading:

* John Sonmez, *The Complete Software Developer’s Career Guide: How to Learn Programming Languages Quickly, Ace Your Programming Interview, and Land Your Software Developer Dream Job*, 2017.

# Required Hardware

* **Same as CEC 322**



**Catalog Course Description (CCD):**

(A) Study of digital computer organizations. (B) Introduction to microcomputer systems using a current microprocessor. (C) Assembly language programming techniques for microcomputers will be used to study digital computer operation. (D) Input and output techniques, memory devices, RS 232, and other interfacing techniques will be studied. (E) Hardware and software relationships will also be discussed.

**Learning Outcomes (LO) as Connected to CCD**

After taking this course, students will be able to do the following:

1. (A) Summarize the basics of the ARM Cortex-M processor architecture, including memory-mapped peripheral registers.
2. (B) Use C to program an MCU for logic operations and run the code in the simulator of Keil.
3. (D) Choose to use the different modes of GPIO and program them using C and assembly languages.
4. (B) Use integer expressions and fixed-point real number expressions.
5. (B) Determine overflow or carry flags for integer operations using ARM convention.
6. (C) Use assembly instructions for arithmetic operations and relate them to corresponding C statements.
7. (C) Use assembly instructions for load and store and relate them to memory access using pointers in C.
8. (C) Use assembly instructions for flow control based on comparisons.
9. (C) Use structured programming in assembly.
10. (E) Use timers to trigger periodic events.
11. (E) Write ISRs to handle interrupts from GPIO and timers.
12. (D) Program peripherals such as USART and I2C.
13. (B) Represent real numbers using IEEE 754 standard for floating-point numbers.

**Modules of the Course**

The contents of the course are organized into modules, each lasting about week. We have the following modules for this semester.

|  |  |  |
| --- | --- | --- |
| Module/Week | **Contents** | **Learning Outcomes** |
| Module 1 | The architecture of ARM Cortex-M processors and an MCU | LO 1 |
| Module 2 | Intro to embedded C programming | LO 2 |
| Module 3 | General-purpose input-output (GPIO) principle and operations using C | LO 3 |
| Module 4 | Data representations and operations | LOs 4, 5 |
| Module 5 | ARM instruction set architecture and mixed C and assembly programming | LO 3 |
| Module 6 | Shift, arithmetic, and logic operations | LO 6 |
| Module 7 | Load/store instructions and C pointers | LO 7 |
| Module 8 | Condition code based on data operations and conditional executions | LO 8 |
| Module 9 | Program flow control instructions based on both signed and unsigned numbers | LO 8 |
| Module 10 | Structured programming and subroutines in assembly | LO 9 |
| Module 11 | Applications of subroutines in assembly | LOs 5 to 9 |
| Module 12 | Interrupt and applications | LOs 10, 11 |
| Module 13 | USART and I2C communications | LO 12 |
| Module 14 | Representation of real numbers using floating-point expressions | LO 13 |

**Content Delivery**

Course content comes to you in several forms. The textbook is the *primary* source of material for this course. The student will learn material more effectively if note contents for each class meeting is reviewed *prior* to the meeting. In-class activities consist of a mix of *mini-lectures* by the instructor and *active learning* activities by the student. Further practice in learning the material comes through *practice problems* pertinent to class meeting learning outcomes.

**Assessment of Learning Outcomes**

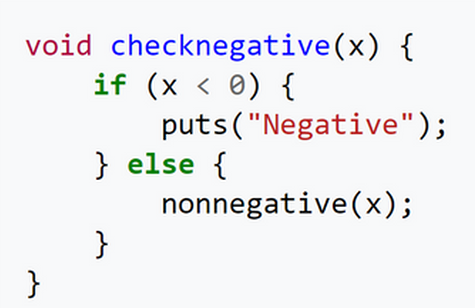
Assessment of students’ mastery of the material is based on a comprehensive approach, including weekly in-class quizzes and homework, as well as programming workshops, which are usually given and collected on Wednesdays. There are TWO tests and an optional file exam.

On Wednesdays, we usually have the following activities in the order of the listing:

* Collect the HW due for that day.
* Answer common questions regarding the HW problems.
* Give QZ. QZ will include three parts: new material not covered in HW yet; material just practiced in HW, and old material covered in previous HW and QZ.
* Perform workshop of the week.

For all the submissions, students are expected to act in professional manner, including:

* Submitting work on time. (Assignments will be given enough time to finish. No late submission is accepted.)
* Working independently when specifically instructed or collaboratively.
* Writing solutions neatly.
* Typing program code neatly following the 1TBS (one true brace style) format (with the 4-space indentation), illustrated below.



**Grading Weighting and Letter Grade Assignment**

There are TWO different ways to calculate the class grade. If you chose the second, a passing grade in the final exam is required to pass the course.

1)

|  |  |  |  |
| --- | --- | --- | --- |
| Test 1 | 30 points | Tests 2 | 30 points |
| Quizzes | 12 points total | Workshops | 18 points total |
| Homework | 12 points total |  |  |

Total: 102

2)

|  |  |  |  |
| --- | --- | --- | --- |
| Test 1 | 16 points | Tests 2 | 16 points |
| Quizzes | 12 points | Workshops | 18 points total |
| Homework | 12 points total | Final exam | 30 points |

Total: 104

**Grading Scale:** 90 + = A, 80 -- 89 = B, 70 -- 79 = C, 60 -- 69 = D, 0 -- 59 = F.

**Class Policies:**

* Active, collaborative learning is proven effective for improving learning. Students are highly encouraged to form teams of 2 or 3 students to do the exercises and workshops in class or online, observing social distancing. Yet, each student should turn in his/her own solutions to all problems, including the workshop problems.
* In the class, no electronic distractions, such as texting or web browsing, is allowed. Calculations can only be performed by stand-alone scientific calculators (**no cell phones**). Talking that is not related to the subject of the practices is not allowed in the classroom. Repeated violations can result in the dropping of the student from the class.
* Students are encouraged to visit the instructor during the virtual office hours if there is anything unclear about the lecture notes, videos, or examples or grading of the problems.
* All work turned in for grading must be done neatly and professionally on Letter-sized “papers”, leaving **enough margins** for making “copy” and “binding”. Please submit pdf only except otherwise specified. Use correct approach and show all necessary intermediate steps. Draw neat, clearly‑labeled sketches as necessary; all the variables used must be clearly defined.

**ACADEMIC INTEGRITY:**

Embry‑Riddle Aeronautical University is committed to intellectual integrity and considers academic dishonesty a very serious offense. Such offenses include cheating (accepting or giving unauthorized assistance in preparing assignments), or plagiarism (taking the ideas, writings, words, and/or work of another and representing them as one's own without appropriate acknowledgment). A student who cheats or commits any form of academic fraud will be subject to a failing grade for the course. In addition, the incident will be reported to the Dean of Students. If any other academic integrity violations have been documented, the student will be recommended for dismissal.