CEC450 – Real-Time Systems

Course Description: The course introduces the concepts of real-time systems from the user and designer viewpoint. The requirements, design, implementation, and basic properties of real-time application software are described with an overview of system software. Related topics such as interrupts, concurrent task synchronization, sharing resources, and software reliability are discussed. A team project on a real-time prototype application may be incorporated in the course.

Goals: The purpose of the course is to provide an understanding of the particular challenges posed by real-time systems, particularly those for embedded and pervasive applications. Topics include the hardware and software co-development methodology, scheduling, tasks, semaphores, message queues, kernel objects, RTOS services, handling exceptions and interrupts, timer services, I/O concepts, memory management, and inter-task synchronization and communication. This course meets the expectations for alumni working in industry.

Textbook: Real-Time Embedded Components and Systems with Linux and RTOS, 2nd Edition, Sam Siewert and John Pratt, October 2015, 978-1942270041 (Mercury Learning, Amazon)

Weekly Syllabus [Please Consult Canvas for all Assignment Due Dates]

Week-1: RT Fundamentals

Read: RTECS, Chapter 1

Lecture: Introductions, Predictable Response; RTOS Project Overview; Cross-Development and Native Development

Demonstration: Final Creative and Standard Projects

Start Assignment #1

Week-2: RT Architectures – RTOS and Cyclic Executive

Read: RTECS, Chapter 2

Lecture: Real-Time System Challenges; Best Effort, HRT, SRT; Software Task Scheduling and Taxonomy, Utility Curves

Demonstration: Jetson and/or Raspberry Pi Embedded Linux

Week-3: RT Services High Level Design

Read: RTECS, Chapter 3

Lecture: Service Request Timeline; Resource Space – CPU, I/O, Memory, Power; Software Task Scheduling and Taxonomy, Utility Curves

Start Assignment #2

Week-4: Timing Diagrams and Safe Utilization Bounds
Read: Liu and Layland Paper on RM Least Upper Bound

**Lecture:** Interference (Preemption); Rate Monotonic Bound; Design Examples for Rate Monotonic, EDF, LLF

**Demonstration:** Real-time services simulation and worst-case analysis with Cheddar

**QUIZ - Part-1 on Foundations, Discuss Quiz Solutions**

**Week-5:** RT Software Validation and Verification

Read: RTECS, Chapter 4

**Lecture:** Rate Monotonic Feasibility Test (LUB Mathematical Derivation); Deadline Monotonic Feasibility Test, N&S Tests, Scheduling Point, Completion Test

**Discussion:** What are the Limitations of RM Theory?

Start Assignment #3

**Week-6:** Service Synchronization and Shared Resources (Real-Time Considerations)

Read: RTECS, Chapter 5

**Lecture:** ISRs and Canonical Services (Tasks) for AMP; Issues with Blocking and Unbounded Priority Inversion Scenario

**Discussion:** Is Unbounded Priority Inversion a Problem for Fair Schedulers like Windows and Linux?

**Week-7:** Additional Challenges with Real-Time Systems

Read: RTECS, Chapter 6

**Lecture:** WCET, Period Jitter, Overhead, Deadlock/Livelock, Unbounded Inversion; Methods to Handle Challenges, Priority Ceiling and Inheritance Protocols; Mars Pathfinder Story

**EXAM #1 REVIEW**

**QUIZ - Part-2 on RT Theory**

FALL BREAK

**Week 8 – Exam #1 – Chapters 1-6**

**EXAM-1 – Knowledge & Concepts, Analysis & Architecture**

Go over exam solutions
Week-9: **Real-Time System Device I/O Interfaces**

- **Return Exam #1 and Review Solutions in Class**
- **Read: RTECS, Chapter 7**
- **Lecture:** Bus I/O, Memory Mapped I/O and Single-Board Computers used in Class
- **Start Assignment #4**

Week-10: **RT Device Driver Interfaces**

- **Read: RTECS, Chapter 12**
- **Lecture:** Application Interface to Services for Devices and Hardware Interface

Week-11: **Working Memory and Persistent Memory Interfaces, Error Detection and Correction**

- **Read: RTECS, Chapter 8**
- **Lecture:** ECC Memory, XOR Parity Review, Hamming Code Review, and Principles of Flash storage and file systems
- **Demonstration:** SECDED (Single Error Correction, Double Error Detection) encoding and decoding (Excel Spreadsheet, Example Code)
- **Start Assignment #5**

Week-12: **Performance Tuning and Timing Verification**

- **Read: RTECS, Chapter 9 & 10**
- **Lecture:** Methods to Profile, Trace and Reduce WCET for Real-Time Systems
- **Demonstration:** Syslog and Custom profile/trace vs. F-trace & Kernelshark

Week-13: **High Availability and Reliability Differences**

- **Read: RECS, Chapter 11**
- **Discussion:** What are some key differences between Reliability and Availability?
- **Lecture:** Definition of High Availability vs. Reliability
- **Start Assignment #6**

**HOLIDAY – Thanksgiving!**
Week-14: RT Project Discussion and Review

REVIEW for EXAM #2, Quiz and Solutions in Class

EXAM-2 – RT Design and Implementation

Week-15: Final Project Lab Time

Debug Session #1 (15 minutes each team) – Meet in King 122

Debug Session #1 (15 minutes each team) – Meet in King 122

Discussion: Final Exam Presentation Format and Required Block Diagrams

Additional Debug Sessions at office hours until 12/10 – See Top Errors

Week 16 – FINAL EXAM - Final Exam and Design Presentation by Team (20+ Minutes Each)

Final Exam in Building 61

Access To Learning: ERAU is committed to the success of all students. It is University policy to provide reasonable accommodations to students with disabilities who qualify for services. If you would like to discuss and/or request accommodations, please contact Disability Support Services located in Hazy Library (first floor, end of hall), or call 928/777-6750 or 928/777-6749, or email the director at: marcee.keller@erau.edu

Civil Rights Equity and Title IX

ERAU seeks to provide an environment that is free of bias, discrimination, and harassment. If you have been the victim of harassment, discrimination or sexual misconduct, we encourage you to report this. If you inform me of an issue of harassment, discrimination, or sexual misconduct I will keep the information as private as I can, but I am required to bring it to the attention of the institution’s Title IX Coordinator. If you would like to talk to the Title IX Coordinator (Dr. Liz Higgins Frost) directly, she can be reached at Building 49, Dean of Students Office, 928-777-3747, froste@erau.edu. For more information, please refer to the Nondiscrimination/Title IX webpage at https://prescott.erau.edu/title-ix/

Learning Outcomes:

1. Knowledge of the issues, the decision and design process, and the available tools and methods for designing real-time, embedded systems (6)
   o Week-#3-6, Assignment #3

2. Understand the particular challenges presented by real-time, embedded systems (1)
   o Week-#1-6, Assignment #1-3
3. Understand the common design problems and their solutions (1)
   - Week-#1-2, Assignment #1

4. Understand the interaction of the cross-compiler and linker, and how that interaction is controlled during the development of the real-time, embedded system. Understand concepts of time-critical computing and identify real-time systems (6)
   - Week-#3-6, Assignment #2

5. Understand host-target development environment for time-critical systems (6)
   - Week-#7-14, Assignment #4-6

6. Write multitasking computer programs with inter-task communication and synchronization (1)
   - Week-#5-6, Assignment #3

7. Apply concepts of inter-task communication and synchronization via shared memory, message queues, signals, semaphores (1)
   - Week-#6-14, Assignment #4-6

8. Understand real-time kernels and task scheduling (1)
   - Week-#6-14, Assignment #4-6

9. Have basic understanding of the interfacing hardware (2)
   - Week-#6-14, Assignment #4-6

10. Understand concepts of reliability in relation to real-time software (2)
    - Week-#9, Assignment #5

11. Have basic understanding of Safety Critical Real-Time software (4)
    - Week-#6, Assignment #3

12. Relate the course experience to aviation applications (7)
    - Week-#6-14, Assignment #6

**ABET Student Outcomes** (Expectations of student knowledge and skills attained by graduation)

1. An ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics
2. An ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors
3. An ability to communicate effectively with a range of audiences
4. An ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts
5. An ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives
6. An ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions
7. An ability to acquire and apply new knowledge as needed, using appropriate learning strategies

**ABET Program Educational Objectives** (Expectations of our alumni working within industry)
The objectives of the Computer, Electrical, and Software Engineering Programs are to produce engineers who:

- Demonstrate achievements in their chosen profession
- Contribute to the development of the profession
- Engage in professional growth
- Contribute to the welfare of society through service

Examples of activities which demonstrate each of these objectives include:

**Demonstrate achievements in their chosen profession**
- Project Leadership
- Technical Leadership
- Received professional recognition
- Strong early-career performance

**Contribute to the development of the profession**
- Educational improvement within profession
- Product development or improvement
- Presentation, publications, patents

**Engage in professional growth**
- Membership in technical societies
- Obtaining additional education
- Achieved career advancement

**Contribute to the welfare of society through service**
- Participation in activities outside the professional area