ECE3849
D-Term 2021

Real Time Embedded Systems

Course Overview

Course delivery: Hybrid Model

- Lectures:
  - In-person: AK 219 MTRF 9:00 – 9:50.
  - Zoom for live streaming as well as recording.
- Exams:
  - Fully remote.
- Lab:
  - In-person AK113 and zoom.
    - Thursdays 10:00-12:50 and 2:00-4:50.
  - Zoom only for extra lab help time.
- Office Hours:
  - Zoom is recommended when needing to share written or electronic information.
  - In-person in AK214.
- Detailed schedules in canvas.

Course Organization

- Course is divided in to 5 Modules
  - Each module 4 – 5 Lectures long.
  - At the end of each module is an exam.
  - Each module has a corresponding module on Canvas.
    - Lecture slides are posted in advanced to the module.
  - Each Module has an Information Page that is updated daily with the following information.
    - Summary of topics.
    - Annotated lecture slides.
    - Link to zoom recording.
    - Related informational links.
  - Each Module has a Module Questions Document
    - After each lecture this document is updated with questions that you should be able to answer from that day.
    - There is no graded homework. This document is used in lieu of homework.
    - I will be using this document as inspiration for the Exam.
    - It is at your discretion on how to use it.

Exams

- 5 Exams, 1 for each Module
- Evenly weighted and 50% of grade.
- Exams
  - Exams will be fully remote during regularly schedule class time.
  - Exam documents will be posted to canvas 5 minutes prior to start of class @ 8:55 am in docx and pdf formats.
  - Exams are open-notes and open-book.
  - Exams are submitted back to canvas in pdf format @ 10:15. This gives 15 minutes to get your answers into pdf format.
  - If technical difficulties arise, do not panic. Contact Prof. Stander immediately.
- Absolutely no collaboration of any kind with others during exam periods.

Labs

- 4 Labs worth 50% your grade. See canvas for weights.
- Due to COVID NO lab partners. Each lab must be done by each individual.
- Lab Attendance
  - Lab attendance is not required at a specific time.
  - Labs must be signed-off by the staff to receive credit for the work submitted.
  - Credit will always be given for the work completed successfully.
    - You can sign-off multiple times if needed.
- Late Penalty
  - Lab sign-offs and reports will be accepted up to a week late.
  - 20% late penalty will be deducted from late submissions.
  - Exception: Last lab cannot be late as grades are due shortly after term closes.
- Sign-off procedure
  - Sign-off starts with the student giving the staff a walk through of their code.
    - What functions are implemented, where they are in the code, and a brief description of the implementation.
  - The student builds program and TA checks for avoidable warnings.
  - The student runs program on hardware.
  - The sign-off checklist is followed and points are given for each fully operational section.
  - Software project is archived and the resulting zip is posted to the lab's code assignment at the time of sign-off. Sign-off is not complete until file is posted to canvas.

Canvas Walk Though

- Everything is in Canvas
  - Syllabus page for schedule of assignments.
  - Modules section for lecture and lab information.
  - Front Page for weekly lab and office hour schedule.
- Enable Notifications in Canvas.
  - I will be using Canvas email and announcements for important or urgent communication.

What is this course about?

- Real-time software
  - Fundamentals of real-time design.
  - Interrupt-driven software, without an operating system.
  - Real-time operating systems (RTOS)
- Embedded system architecture
  - CPU architecture and performance
  - Input / Output interfaces
  - Memory subsystems

What should you already know?

- ECE2049 Strongly recommended
- C programming
- In-circuit debugging
- Direct access to I/O ports
- Periodic interrupts
- Analog I/O basics
- Helpful background
    - Operating Systems
    - Assembly language

Different types of embedded systems.

- Real-Time Systems
  - Example: Motor controller for a robot
  - Focuses on reliability
  - Lacks advanced features
    - All parts of the system must be well understood.
    - Simplicity is important
  - Runs either on bare hardware or an RTOS.
  - Runs on a microcontroller (usually from on chip memory).
- User interface and/or feature driven
  - Example: Smart phone – click an app and it runs.
  - Advanced features are important.
  - Runs on a traditional OS, such as Linux.
  - Runs on a system-on-chip with off-chip main memory storage.
  - Not well suited for real-time software.