CSE321 Realtime and Embedded Operating Systems Fall 2019

Goals:

To apply the embedded and real-time systems concepts that you will study in the course. These are:

1. Ability to design and construct a complex hardware and software system, component, or process to meet desired needs using relevant software engineering principles, within realistic constraints such as economic, environmental, social, political, ethical, health & safety, manufacturability, and sustainability.
2. Ability to identify, formulate, and solve hardware and software computer engineering problems using sound computer engineering principles.
3. Ability to effectively communicate technical information in speech, presentation, and in writing.
4. Continue the learning in the course to engage in lifelong learning.
5. Ability to understand contemporary issues in this area.

Objectives: We plan to address the goals stated above using a hands-on project that will involve original design, and implementation, demonstration and presentation of embedded/realtime system. This is a demonstration project that includes a presentation of your completed project. For this project you will work as individuals.

Problem Statement: Choose a concept/idea after discussion with your TA and your instructor. The idea has to be approved by either the TA or the instructor. The scope of the project should be implementable within the time and the resources available.

1. Choose/Identify an idea that has economic, environmental, social, ethical, health and safety, manufacturing and sustainability (say, in energy) impact.
2. Formulate the solution for the problem identified using best practices learned in the course.
3. Implement and test the solution. Prepare for the demo and presentation.

What to do?

You will design and develop the solution in stages. To guide you through the process the term project development is divided into three stages: Lab1, Lab2 and final presentation of project. More details will be provided

Phase 1: Lab1: Research for a topic and idea. Make sure you pay attention to all the goals stated earlier. Discuss the project with TA or the instructor and get it approved. Record the idea on a Google form provided. Prepare a document with these items: (i) a title (ii) 100 word abstract of what, why, how, impact etc. and (iii) one or more pictures explaining what you plan to do, (iv) functional and non-functional requirements, (v) hardware and software requirements along with tools needed and (vi) references. Submit the document for grading. Due date: 9/20

Phase 2: Lab 2: Design, and implementation of the prototype. Design the solution using standard methods that we will discuss in class. Procure the items needed for the development of the design. Implement a basic prototype and verify a baseline implementation. Demo the prototype to the TAs during recitation. Due date: 11/1

Finalize term project: Complete the implementation and prepare a detailed documentation of the project. Document all the measurements and observations. Present the completed project during lecture/recitation and possibly at UB or local event or competition. 11/18 onwards

Solution ideas: These are a few possibilities. You can always work with other platforms you are familiar with or interested in exploring.

How to choose a project idea? Think about the keyword "automation" and automating something really useful for you and your peers. Think about an innovative idea that can be presented at a "startup weekend" or at such competitions. Think about the keyword "smart".

Think of impactful domain: healthcare, automotive systems, assistive systems, IOTs and similar.

You can work with Adruino (www.arduino.cc) Uno hardware as the primary base hardware. You can use other similar small embedded system such as Rasberry PI (http://www.raspberrypi.org/), Maple (http://leaflabs.com/devices/maple/) or Parrot drone or Phantom drone. The discussion below is equally applicable to any of the platforms.

Arduino itself has many variations such as Arduino Zero, Mega, Arduino Duo etc. Arduino is a very popular hardware used in many modern/contemporary systems. Ford OpenXC is ardunio compatatible, for example. "The OpenXC kit includes a vehicle interface module based on the popular Arduino platform developers can use to read data from the vehicle's internal communications network. The hardware module provides real-time access to parameters like the vehicle sensors, GPS receiver and vehicle speed. The hardware module is connected to a smartphone or tablet on which apps can be written to consume and use these data. See http://openxcplatform.com/
How about Google's driverless car? Apple's? How about the drones for various purposes?