

Real Time Embedded Homework 1

ECEN 5623

Zachary Vogel

January 19, 2016

Problem 1

A first example would be my senior capstone project. The goal was to use capacitive power transfer to charge the battery of an electric vehicle while it was moving. This meant the system had to respond to an interrupt detecting the vehicle, and apply the power transfer during the limited time frame in which the vehicle was over each individual plate. It is real-time because we had to respond within a millisecond or so so that power was being transferred for the majority of the time the vehicle was over the plate. It was embedded in that it had specific applications.

Another example would be a quadrotor. The system must measure it's current position and orientation constantly and adjust the motors to maintain stability. This is real-time because it must respond relatively quickly or risk crashing. Embedded in that it isn't a general purpose computer.

Problem 2

As far as I can tell, the reason we assume processes are periodic is because it makes the math more reasonable and often real time tasks are periodic. An example of this would be checking the position and orientation of a quadrotor. This is done periodically because the control engineers know their control algorithm works when they get that information at periodic deadlines. In a real application this might be a problem because many real time services are not periodic. For instance, a request for data from a server would not be periodic because the request come in randomly and take different amounts of time based on the amount of data needed. Pretty much anything that is respondent to a user input, or a random event would not be periodic. The reason this is okay is because you can largely account for your periodic processes and then use experience to determine what is a reasonable amount of overhead for your random non periodic services.

Problem 3

Hard real time services have specific deadlines that have to be met. If these deadlines are not met than the system will absolutely fail. These are "hard" in that your system will crash if these deadlines aren't met. An example would be the control systems that maintain the ball of plasma in a nuclear fusion reactor. If these systems fail maintain the correct magnetic field to keep the ball of plasma from touching the interior chamber of the system the interior chamber will melt and your system will be destroyed. Soft real time services have specific deadlines, but they don't always have to be met. That is to say, if you miss a few deadlines it probably isn't the end of the world and the whole system will not fail because of a few failures. Performance might degrade somewhat, but the system shouldn't fail. An example is a server farm providing music to customers. The servers have deadlines for how long they have to return the data for the music, but if they don't meet those deadlines the person listening on the end just has to wait a little bit.

Board Selection and Partner

I definitely want to use the Altera board to do embedded Linux. My partner is Chuyi Liu (Zoe).