CIS 541: Embedded and Cyber Physical Systems
Course Project (Draft)
Pacemaker Challenge

February 2, 2010

Goal

This project familiarizes the students with the basic procedures of model based software design, verification, and implementation. Critical issues with regards to life-critical embedded real-time systems will be addressed.

Problem Description

Embedded systems, when used in conjunction with sensors that interpret inputs from environment (patients) and actuators to control its environment, are building blocks of cyber physical systems. One such example is the pacemaker controller. A pacemaker is a medical device, which senses heartbeats regularly and sends electrical impulses to the heart in order to maintain the heart rhythm of a patient.

Timing constraints are of one important aspect of correctness criteria. In this project, you will mainly focus on the design, verification, and implementation of a pacemaker controller (working in the VVI mode and DDD mode) with timing constraints.

More information with the Pacemaker Challenge can be found at

- The specification document http://sqrl.mcmaster.ca/_SQRLDocuments/PACEMAKER.pdf.
- The website http://sqrl.mcmaster.ca/pacemaker.htm.
- The FAQ page http://www.cas.mcmaster.ca/wiki/index.php/Pacemaker_FAQ.
- The “Pacemaker Cartoon Book” Cardiac Pacemakers Step-by-Step: An Illustrated Guide http://www.wiley.com/WileyCDA/WileyTitle/productCd-1405116471.html.

In this project, you will design a simple pacemaker controller. However, instead of just coding, you will be instructed to finish the following milestones, which will be assigned in multiple homework assignments.

Milestone 1: Modeling

You are asked to design a higher-level model of the system. This model will serve as the blueprint of the subsequent work. Based on the model, you will verify properties of the model to make sure the design is (conceptually) correct. You will also use this model to serve as a guide for implementation. The modeling language UPPAAL (http://www.uppaal.com) will be introduced in class. It is able to create timed automata models and can verify properties specified in temporal logics. It is allowed to do this part with Matlab (Stateflow/Simulink) instead of UPPAAL.

Milestone 2: Verification

You are asked to assure the correctness properties of the model designed. This will involve reading the pacemaker specification document and identify correctness properties, specifying the properties in temporal logics, and employing verification tools (UPPAAL) or testing tools (Matlab) to verify the correctness of your design. You need use the notion of "assurance cases" to put together your claims and evidence.

Milestone 3: Implementation

You are to actually program the code with devices provided. The pacemaker controller will run on PIC18F4250 board (http://ww1.microchip.com/downloads/en/DeviceDoc/39631B.pdf). Sensed heart signals can be simulated by the programming environment for the board (MPLAB IDE, http://www.microchip.com/mplab). A/D converters that transform the sensed heart signal are also available. Experience with C programming in general is expected in this milestone. (There may be other platforms available for this part.)

Milestone 4: Validation

Due to the many timing constraints for the pacemaker, validation of the correctness of the program is an important step. In this milestone, you will employ methods of assurance cases to reason more correctness of our design, in addition to argument what the properties you've verified in Milestone 2 are also not violated. Necessary profiling and/or testing of your implementation to gather enough evidences for reasoning are expected.

Milestone 5: In-class Demo

Each group should finalize what they've done in the project as a report. Also, an in-class demonstration of the implementation is expected.

Collaboration Policy

The project can be done in groups of three or four. It is not advised that each member take one milestone and work alone. For one reason, the workloads of the milestones are different. For another, it would be beneficial for all members get to know the process of model based software design in an embedded and cyber physical setting.

Assessment

Final grade for the project is assessed based on the items below. The first six should appear in the project report.

1. The design model (15%)
2. What correctness properties are considered (15%)
3. How the model satisfies correctness properties, i.e., the verification result (10%)
4. Implementation code (25%)
5. Using assurance cases to argue the correctness of your design/implementation (25%)
6. Member contributions to the project (5%)
7. In class demonstration (5%)
8. Extra credit: additional features. (Pacemaker operation modes other than VVI or DDD; interfacing with A/D converters rather than just software simulation; etc.)