Object Oriented Design in RTOS

John C. Shovic

LAN System Block Diagram (Rev 1)

Software Simulators

- Cycle/Cycle simulators preferred
- Special Hardware must be simulated
- Often, validation is not good. When run on real system, more failures
Software Debuggers

- Indispensable
- With HW support, very very useful
  - Very useful in Module Level Testing
  - Less Useful in System Testing

Most Important:

When removing code for debugging, don't use conditional branching. **Use conditional compilation instead. Branching invokes the Heisenberg uncertainty principle!**

The Software Heisenberg Uncertainty Principle

- The principle restated:

  \[
  \Delta r \Delta s = H
  \]

  Where:
  - \( H \) is some constant
  - \( r \) is the uncertainty of the code,
  - \( s \) is the uncertainty of the test specifications
Heisenberg might have been here...

- The more closely a system is examined, the more likely the examination process will affect the system being tested.
  - Software reliability is also affected by the code added to measure the correctness of a system!

Interrupts are Very Sensitive

- Single Stepping many times will not find interrupt problems
- Adding sensing instructions will change timing
- Non-intrusive methods are best in this case
  - Monitor system bus with LA, for example

OO Design

- Standard RTOS's are well suited to OO Design
- "Tasks" are like objects
- Communication paths need to be constrained for true OO.
Global Variables

- Few if any global variables
- Global variables should be protected by Semaphores for writers and readers
- Why?
  - Many Readers - One Writer
  - Many Readers - Many Writers
  - One Reader - Many Writers

OOD

- Isolation of function is important for good RTOS design
- Too high level of interaction is a bad design - too hard to get right
- Good cohesion allows for recovery from problems

Homework Assignment

You need to be working on your papers!

Vote early and often!