Real-Time & Embedded Systems

**Agenda**
- Technology Development in RT&ES
What we have covered?

- Simple “Bare Metal” Configurations
  - Freescale 68HCS
  - MSP430
  - Other microcontrollers (Arduino)
- Athena/Athena II/Helios Architecture
  - Single processor
  - No additional cores
  - Utilization of RTOS
RTOS vs Bare Metal

• RTOS’ ease of use
  • Implementation of an RTOS can make life easier on embedded developer
    • Comes with caveats, of course!
      • Power – less likely to enter power saving mode
      • Storage – more space required for the RTOS
      • Memory – generally more needed to handled the threads, etc
  • Bare Metal is ”faster” for a given simple solution
    • Less ”cruff” to get in the way, such as the threads, IPC, drivers, dynamic loading, etc
    • Designed specifically for a need usually

(c) Copyright 2012 Dr. Phillip A. LaPlante
But wait there’s more...

- These are not revolutionary developments
  - There has been bare metal work since the dawn of uCs
  - RTOS has been around a *long* time (SE/CS/CE terms)
    - VxWorks
    - QNX
    - TI’s SYS/Bios
    - Freescale’s MQX
    - Others
  - RTOS’ that we have discussed are very proprietary / customized solutions
So what’s been happening?

- Multi-processor systems have made more in-roads into the development of RTOS’s
- Multicore solutions are becoming more prevalent/introduced
  - Homogenous Multiprocessor Systems
  - Heterogeneous Multiprocessor Systems
  - Hybrid Systems
Multiprocessor

- Multiprocessor arrangements add benefits
  - More available MIPS
    - If one processor is busy the other may be available
  - Separation of duties
    - Ex: one for HMI, one for Actuation
  - Security domain separation
    - Have “secure” processing on one versus the other

- Caveat
  - More space
  - More power
  - Possibly more support needed for IPC

(c) Copyright 2012 Dr. Phillip A. LaPlante
Multicore - Homogenous

- A system that uses identical processor “cores” to achieve one “chip”
- This has the same advantages of the Multiprocessor setups, though usually has additional benefits
  - Smaller footprint
  - Less power
  - Easier IPC, possibly with less components or even built in to the silicon

(c) Copyright 2012 Dr. Phillip A. LaPlante
Multicore – Heterogeneous

- Systems where the “cores” are not identical
  - Ex – Arm A9 for one core, Arm M9 on the other
- These can have several advantages over the Homogenous design
  - Use of one core only
  - Boot one core from the other
  - Run an RTOS on one only
  - Cost reduction
  - Space / power savings
Hybrid Systems

- These silicon solutions may have the form of one core of a uC and the addition of a FPGA core
- Interesting ideas, rather new to the RT&ES development environment
  - Intriguing possibilities – if one uC is always the same, you could implement a “soft core” on the FPGA to dynamically change the secondary core
SWAP and "Micronization"

- Concerns revolved around *Size, Weight and Power*
  - Everyone likes items smaller, take up less space
  - The need to have "more with less"
  - Utilize the current battery technologies more efficiently
    - Fuel cells
    - Alternative power
  - IoT

(c) Copyright 2012 Dr. Phillip A. LaPlante
Power Savings

- Obvious, and current, thoughts are putting uCs to “sleep” when not needed to save even mAs.
- Even to the point of “turning off” features of the uC (such as the ADC, DAC, the GPIO, the Timers, Counters, etc)
- Here there is efforts to “harvest” power using various techniques:
  - Motion
  - Solar
  - RF “re-harvesting” (BT, RF, WiFi, etc)
IoT – Internet of Things

- Does anyone think that Windows is going to run all the IoT devices connected to the Interwebs?
- What will those devices be?
- What will they run?
- What will they do?
- How will they be powered?
- Why are they connected, and how?
That was Hardware centric...what about the Software world???

- New languages / development methods to speed turn to the customer (faster product introduction)
- More tools to leverage work done, speed release
  - DTS in Linux is one (Device Tree – speed the ports to various boards, such as Beagle / Panda)
  - OOA / OOD techniques

(c) Copyright 2012 Dr. Phillip A. LaPlante
That was Hardware centric...what about the Software world???
(contd)

- Increase in Modeling techniques
  - Did someone say UML?!
  - Integration of suites to drive requirements from customer to development to test, with traceability! (IBM Rational Rhapsody suite)