CEC 450
Real-Time Systems

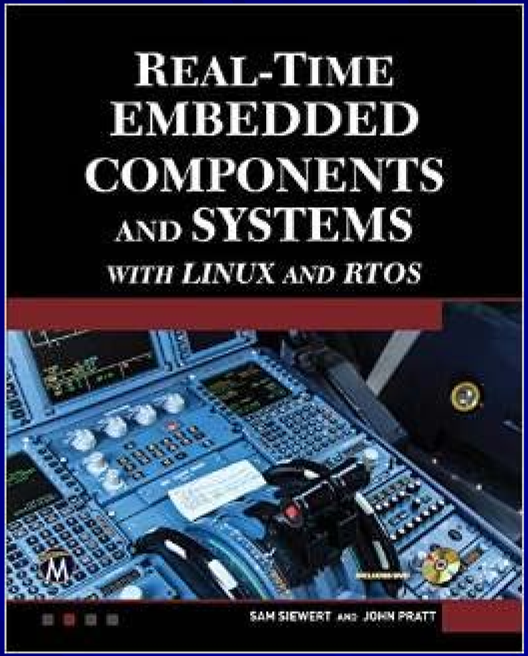
Lecture 1 - Introduction

Real-Time Background

- NASA Johnson Mission Control, SAIL
  - **1990-92 -** Software Avionics – Ascent / Entry Guidance
  - Flight Software Upgrades and MCC Upgrades
- Ph.D. Work – RT and Interactive Systems
  - 1994-2000
  - LTA UAS – a “Blimp”
  - Optical Navigation and Control Systems and Software
  - Extension to Rate Monotonic Theory (Statistical RMA)
  - Spitzer Multi-Epoch Scheduling and RMA
- RT Instrumentation and Machine Vision
  - Spitzer Space Telescope – 1997-2000
  - 5 Observing Modes (Mosaic, Total Power, Super-resolution, Spectral Energy Distribution, Raw)
- Network Processing – Non-RT (2000 – 2012)
  - Throughput, Low Latency (10G Ethernet, Fiber Channel, SAS/SATA, RAID)
- Cable Labs RT “Head End in a Box” (2006)
  - Broadcast UTC (GPS Time, NTP)
  - Soft Real-Time Encode, Transcode, Decode, Multiplex
  - QAM, IP, DVB-ASI
- Consulting (2012 – Present)
  - RT Graphics, Through-put Storage and Networking, RT UAS/UAV
- Current Research
  - RT Multi-spectral Object Detection, Fusion, 3D Mapping

Course Goals and Outline

- **Real-Time Embedded Components and Systems with Linux and RTOS, 2nd Edition, Sam Siewert and John Pratt, October 2015, 978-1942270041, [Mercury Learning], [Amazon], [Graduate Class at CU Boulder]**
- A balance of 1/3 Theory, 1/3 Practice, and 1/3 Development [Project – Time Lapse Video or Student Initiated]
- http://mercury.pr.erau.edu/~siewerts/cec450/
- Fall 2016 Syllabus



Supporting Materials On DVD and on our Class Website

Hard and Soft RT

What's Hard? Soft?

We'll Find out….

RT Embedded Systems - Lots of Jargon

- True of Computing in General – New, Specialized, Evolving
- More True for RT
- Systems Jargon – HW, FW, SW
- See Textbook Glossary

jar·gon[1]
/ˈjärgən/

noun
noun: **jargon**; plural noun: **jargons**

special words or expressions that are used by a particular profession or group and are difficult for others to understand.
"legal jargon"
synonyms: specialized language, slang, cant, idiom, argot, patter; More

- a form of language regarded as barbarous, debased, or hybrid.

Origin

OLD FRENCH

jargoun → jargon
twittering, chattering,
late Middle English

Use definitions provided in RTECS for class

Quiz on Key Terminology (Jargon)

Much Comes from Microprocessors, Computer Architecture, Operating Systems, Compilers

Embedded Linux and FreeRTOS

- Option #1 – Jetson Embedded Linux – POSIX RT
  - Linux Provides only Predictable Response (Not Deterministic) and Only with Use of POSIX RT Extensions
  - Some Labs will Require Use of Linux
  - Final Project can use Linux or FreeRTOS

- Option #2 – Use Texas Instruments Microprocessor and FreeRTOS
  - Labs Developed by Dr. Brian Davis
  - Labs Developed by Dr. Siewert, Ported from VxWorks to FreeRTOS
  - Some Labs will Require Use of FreeRTOS
  - Final Project can use FreeRTOS or Linux

Administrivia

- Introductions
  - Instructor (Office Hours) – Fall 2016 Schedule
  - Students (Introductions) – Please do Collaborate, but cite well!
  - Policies - http://mercury.pr.erau.edu/~siewerts/cec450/policies/
- ERAU ERNIE and Canvas
  - Primary Assignment Management Tool - https://erau.instructure.com/
  - Access via ERNIE - https://ernie.erau.edu
  - Mercury Website - http://mercury.pr.erau.edu/~siewerts/cec450/
- Course Information
  - Attendance & E-mail list (please sign up on sheet being passed around)
  - Lecture Notes at http://mercury.pr.erau.edu/~siewerts/cec450/documents/Lectures/
  - Assignments Posted on CANVAS
- You Will Each Have Access to an Embedded Linux Jetson System (Shared with CS 415, HCI)
- We have Texas Instruments Boards for Use with FreeRTOS

Huge Range of Embedded Systems…

- From High Performance Computing…

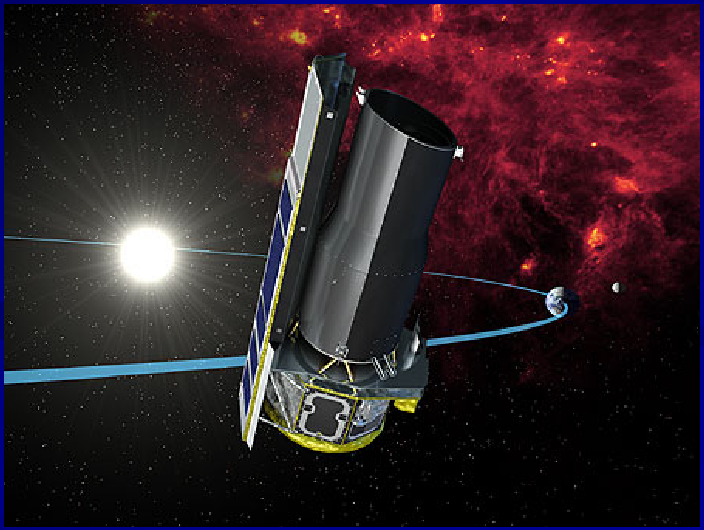
http://www-128.ibm.com/developerworks/views/power/libraryview.jsp?search_by=big+iron+lessons

- PowerPC: from Play-Station/X-Box to Blue Gene/L

http://www.llnl.gov/asci/platforms/bluegenel/photogallery.html

University of California, Lawrence Livermore National Laboratory, the Department of Energy and the Advanced Simulation and Computing

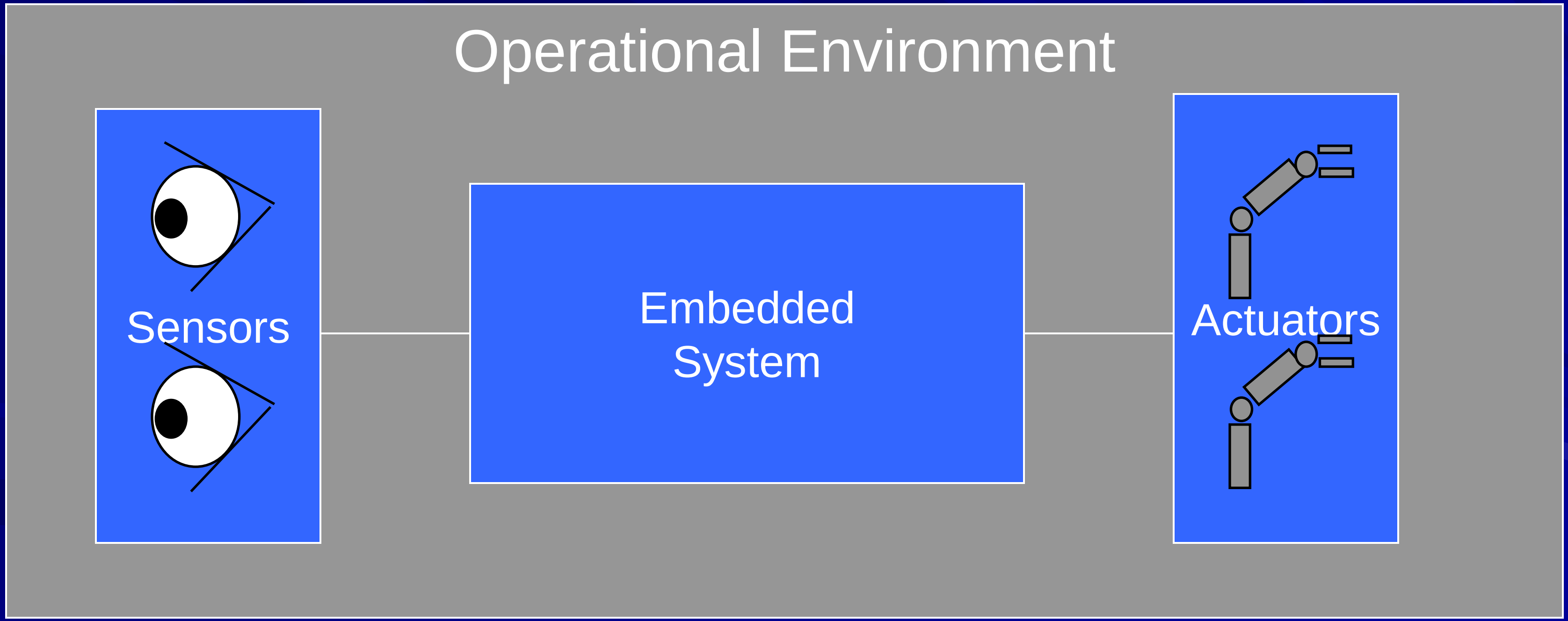
- Reconfigurable HPC - http://www.srccomp.com/



- To Cosmic Origins
- Embedded PowerPC Spitzer Telescope
  - http://www.spitzer.caltech.edu
  - NASA Jet Propulsion Lab and Cal Tech University

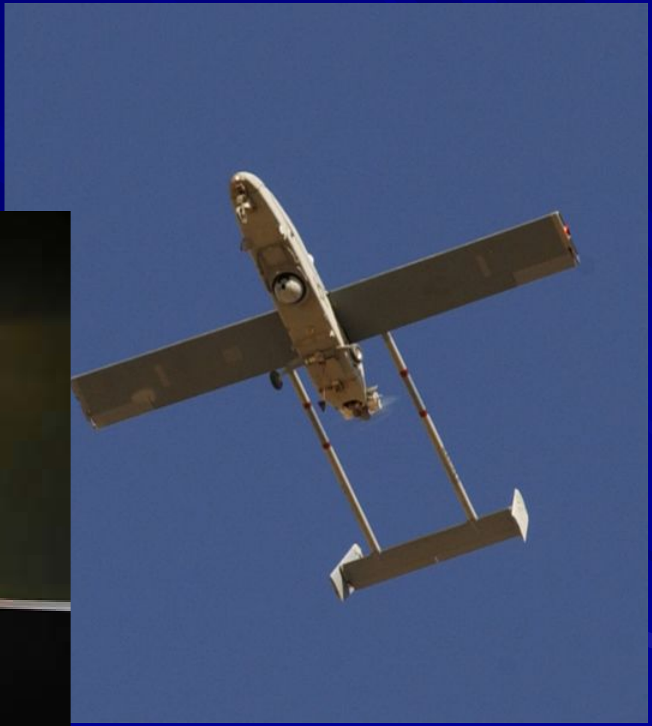
To Be Embedded

- A Compute Node That Provides Specific Services by Processing Inputs and Producing Required Responses
  - Provides Specific Services Rather than General Purpose Computing
  - Often No Direct Connection to User Input/Output
  - Contained within a Larger System as a Sub-system



Many Real-Time Embedded Systems

- Real Time – Must Respond to Requests for Service by a Deadline relative to request
  - Failure to Respond Prior to Deadline Results in a System Failure
  - Request Rate for Service Driven by Real-World Events
  - Controls Processes and Delivers Deadline Driven Services
- Anti-Lock Braking
- Streaming Media (Video and Audio)
- Process Control
- Aircraft Flight Control
- Robotic Systems



Why are RT Embedded Systems a Challenge?

- **Real-Time Services** – **<u>Correct Results on time – Deadlines!</u>**
  - Multi-service Concurrency Required, for Software, Multi-threaded
    - Multiple interfaces to service in addition to data processing
    - Multi-threaded compared to Main Loop + ISR Executive
    - Supports RT analysis and design (Rate Monotonic)
  - Function/Service Allocation – HW Service Off-load
  - Management of CPU, IO, and Memory Resources
- **CPU Resource**
  - Modern architecture – high throughput, less deterministic
  - pipelines, super-scalar, branch prediction, VM, split-transaction and burst transfer bus interfaces, multi-level caches
- **I/O Interface Resources**
  - Sensors / Actuators (Interaction with Real World)
  - Networks (Latency and QoS)
  - Off-load and Memory Devices (e.g. Flash, FPGAs, DSP)
- **Memory Hierarchy Resources**
  - Register file, L1/L2 cache, SRAM, dynamic RAM, Flash

How to Make RT Embedded Systems Easier!

- **RT Service and CPU Resource Management**
  - RT Theory, Practice, and Pitfalls (Theory -> System)
    - RMA and DMA Resource Theory
    - Prediction and Measurement of Performance
    - When to Allocate Services/Functions to HW, FW, or SW
  - Multi-threaded RTOS Systems
    - Design Methods (DFD, SDL, EFSM and MSC methods)
    - RTOS Mechanisms (e.g. message queue, signal, semaphore)
    - Analysis Tools (e.g. Windview)
    - HW/FW Debug Tools (e.g. ICE / JTAG)
- **I/O Device Interfaces and Drivers**
  - Abstracted SW-HW Interfaces
  - Interaction with Memory System (MMIO, DMAs, Plug-n-Play)
- **Memory Hierarchy Analysis and Abstraction**
  - Multi-level Cache Performance Models
  - Abstracted Non-Volatile Memory Filesystems

Final Project (Assignment #5 & #6)

- Groups of 2-3 Students (Individual by Approval)
  - Must demonstrate **2+ Deadlines in Requirements**
  - Can use FreeRTOS or Linux with POSIX RT Extensions
- Default RT Emphasis Project
  - Time Lapse Video of Physical Process
  - Harder than it Sounds to Do Well
  - Verify with External Clock
- Project of Your Own Design
  - Computer Vision Projects - Peak-Up, Optical-Nav, Video Compression / Transmission
  - Robotics (Proof-of-Concepts) – E.g. Tilt/Pan Servo Tracker

Example Projects – Time Lapse

- Sweep Second Hand
  - Observe Precision Analog Quartz Clock
  - Synchronize to Second Hand
  - Analyze Latency and Jitter
- Observe Physical Process
  - With Clock in View
  - Which Clock is Correct?
  - How Do We Know?
- Observation at
  - 1 Hz
  - 10 Hz
  - Max Hz
- Off by Second in 30 Minutes!
  - 230 milliseconds by analysis
  - Camera?
  - I/O?
  - Processing?

Frame N

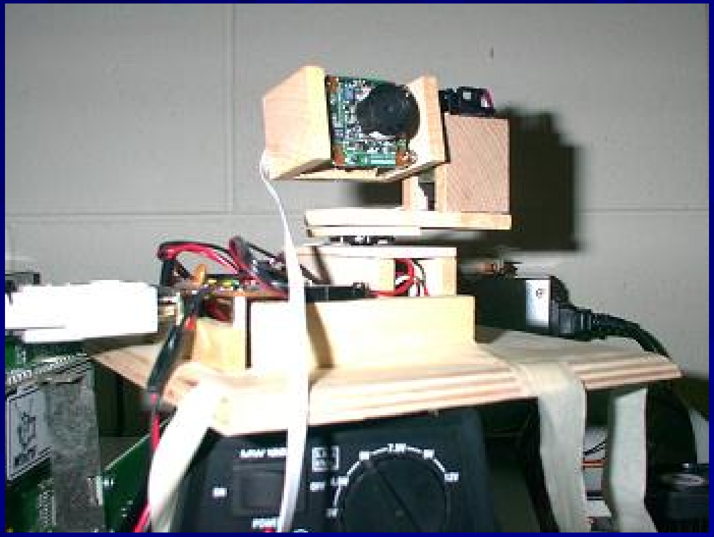
Frame N+1

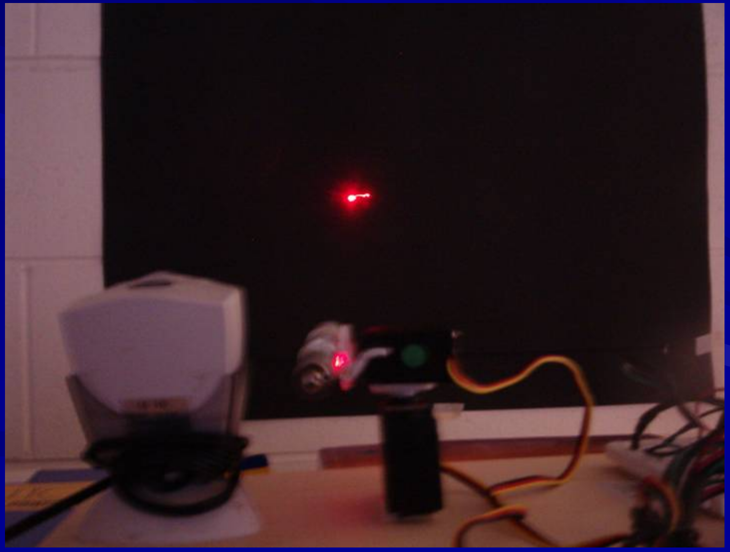
Frame 1

Frame 1800

Example Projects – Creative

- Camera Peak-Up
  - Camera tilts and pans to track object
- Target Tracker
  - fixed camera, laser pointer tilts and pans to track
- Stereo Vision Tracker
- Tracking Speed
- Intensive Image Processing
- Numerous Examples found Here
- Correct Time – NIST, NTP, RF Broadcast, F2 Atomic Clock, Time Standards, USNO Time





Embedded Linux Overview

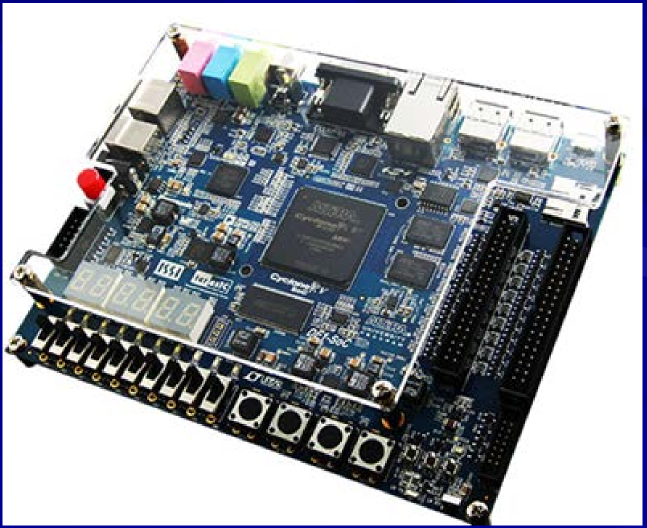
Introduction Session

EMBRY-RIDDLE
ARIZONA

Integrating Linux into RT System

- RTECS v2 – Chapter 11
- User Space POSIX Threads and RT API
- Kernel Modules
- Linux Kernel Patches to Improve Pre-emptibility
- Risky for HRT (Hard RT), Option for SRT (Soft RT)

Hybrid Solutions – FPGA + SoC



Altera DE1-SoC

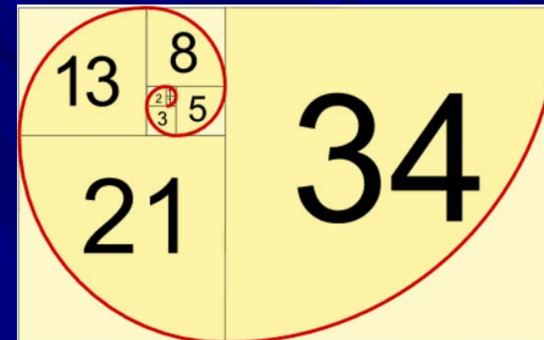
Embedded Linux

- Jetson Systems – Jetson eLinux, NVIDIA Embedded
- Must Use **root** privilege POSIX RT Threads
- Possible to Get Predictable Response
- Never as Deterministic as FPGA State-Machine or RTOS

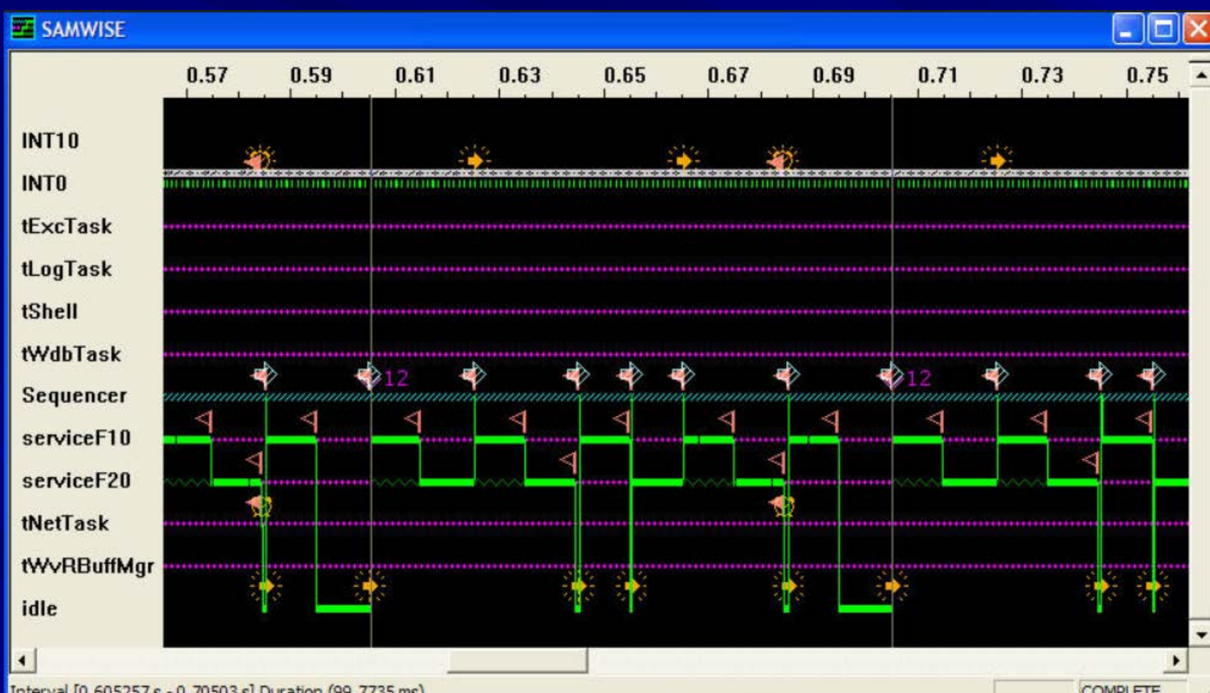
https://developer.nvidia.com/jetson-tk1
http://coloradoengineering.com/tk1-som-nvidia-tegra-k1-ultimate-iot-m2m-platform/

POSIX RT Task Example

- S_1 – Computes Fibonacci for 10 milliseconds and quits
- S_2 – Computes Fibonacci for 20 milliseconds and quits
- C_1=10, C_2=20 milliseconds
- T_1=D_1=20 & T_2=D_2=50 milliseconds



https://www.mathsisfun.com/numbers/fibonacci-sequence.html



CPU Loading = ____ %

Feasible Schedule?

Safe? Easy to Emulate?

Violates Rate Monotonic Feasibility Test?

$$U = \sum_{i=1}^{m} (Ci/Ti) \le m(2^{\frac{1}{m}} - 1)$$

What Does Theory Tell Us?

- Look over LCM of Periods (*J Lehoczky, L Sha, Y Ding*)
- If Schedule Feasible over LCM, Feasible for All Time

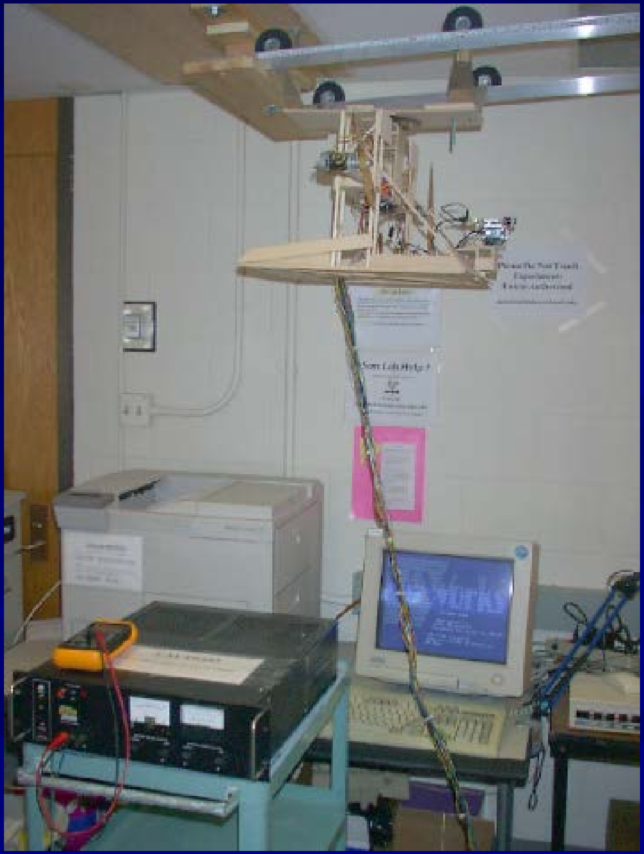
Example 5	T1	2	C1	1	U1	0.5	LCM =	10		
	T2	5	C2	2	U2	0.4				
	T3	10	C3	1	U3	0.1	Utot =	1		
RM Schedule	1	2	3	4	5	6	7	8	9	10
S1										
S2										
S3										
EDF Schedule										
S1										
S2										
S3										
TTD										
S1	2	X	2	X	2	X	2	X	2	X
S2	5	4	3	2	X	5	4	3	X	X
S3	10	9	8	7	6	5	4	3	2	1
LLF Schedule										
S1										
S2										
S3										
Laxity										
S1	1	X	1	X	1	X	1	X	1	X
S2	3	2	2	1	X	3	3	2	X	X
S3	9	8	7	6	5	4	3	2	1	0

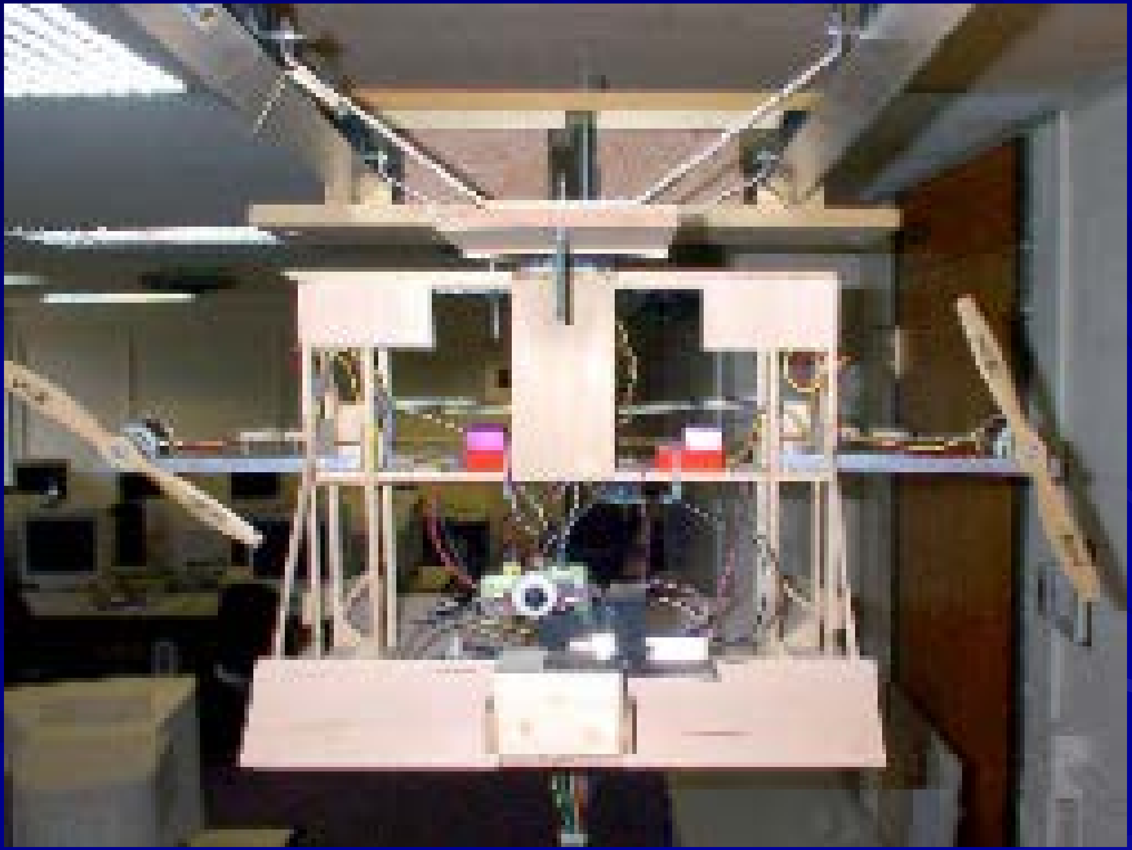
Next Time …

- What Does Theory Tell Us?
- Can We Emulate the Fibonacci Workload Accurately
  - In Linux?
  - In FreeRTOS?
  - With an ISR and Interrupts Asserted by a Programmable Interval Timer?
- What are Pitfalls?
- Why Would I want to Do This?

Complex Multi-Service Systems

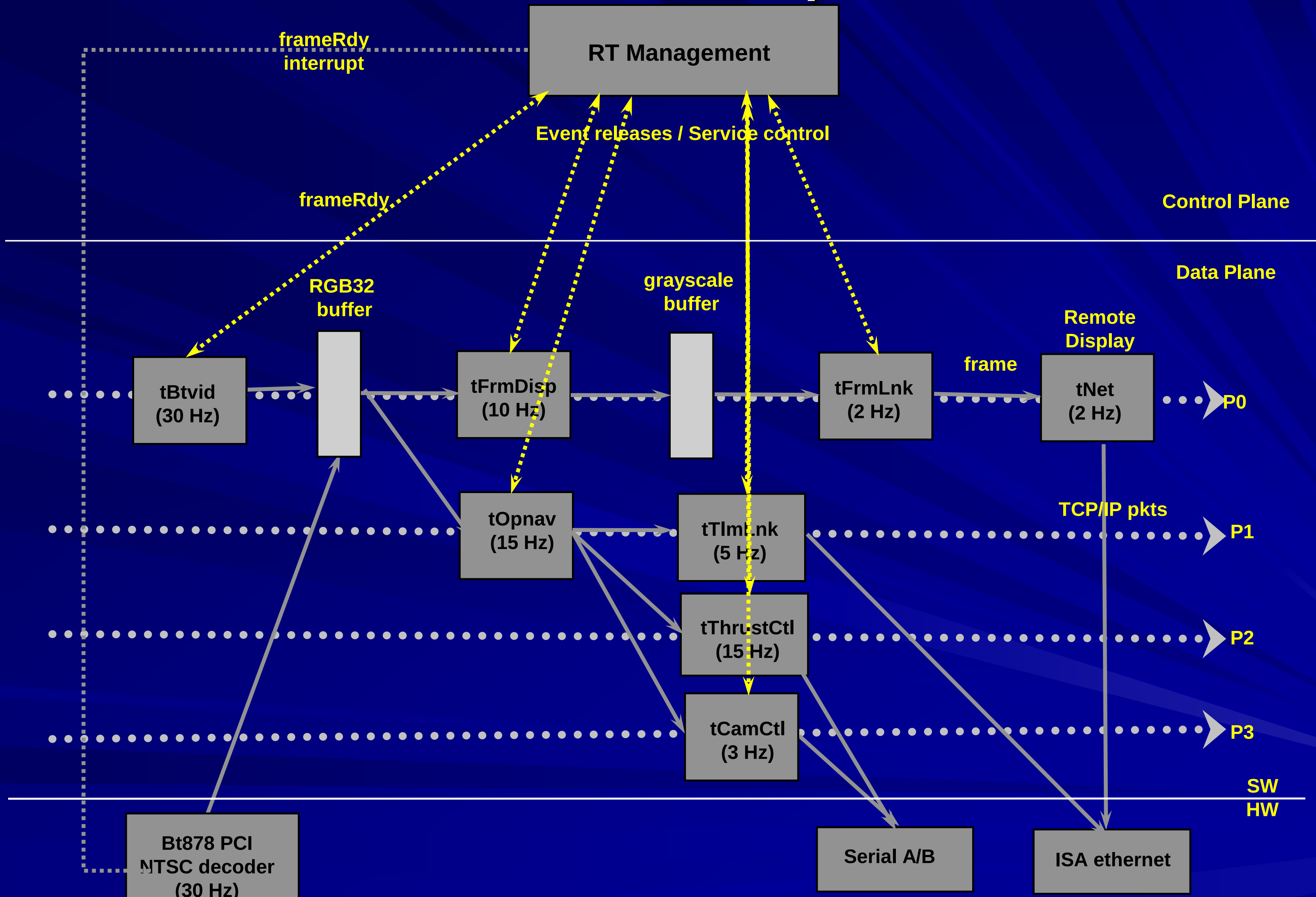
- Multiple Software Services (Dynamic Admission – Reconfigurable)
- Synchronization Between Services
- Communication Between Services
- Multiple Sensor Input and Actuator Output Interfaces
- Intermediate IO, Shared Memory, Messaging





[Ph.D. Dissertation, CU Boulder, January 2000]

Multi-Service Pipelines



So Why Use Software for HRT Systems?

- ASIC and FPGA State-Machine Solutions Offer Hardware Clocked Deterministic Solutions
- FPGAs Can be Updated with New Bit-streams (Synthesized HDL to Reconfigurable Logic Elements)
- Software (Firmware) Remains Simplest for Field Upgrade (Reconfigurable at Run-time)
- FPGAs Can be Costly Per Unit
- Cost of Software Engineering vs. Hardware Engineering