

Open-Source Ambassador

- Premise
 - The typical WRS customer is exploring Linux and/or Open-Source Software (OSS) in some way
 - The smart customers are going to figure out how to maximize their development dollars in this new world
 - WRS bottom-line and the customer are both served by helping the customer understand this new world

Welcome to an OSS Case-Study

- Today we will go through a design roundtable, featuring the engineering requirements of the Hypothetical Part-Stamping Machine.
- As we step through the design and implementation process, we will take breaks and demonstrate the concepts utilizing open source tools and software.

The Story (so far...)

- Monday morning 9:00am
 - Our hero arrives at work still thinking about his girlfriend and her ultimatum : more time with me or else...
 - On his desk is a note to see his boss, urgently
 - Behind closed doors, his boss explains the sorry state of the economy and what is coming down the pipe...

Our Story Continues...

- Monday morning in the boss' office
 - The company is losing too much money
 - Projects are being cut
 - Concentrating on core expertise
 - Layoffs are certain
 - Upper management has decided upon a contest

The Contest

- Monday morning in the boss' office
 - A new design is about to begin, taking the company in new directions
 - Each design team has one week to present their solutions to the next great Part-Stamping machine
 - The chosen design will be implemented, with all other teams receiving pink slips

The Dilemma

- Monday at Lunch
 - Our hero has much to think about, how can he save his relationship and his job in the same week?
 - It's a good thing our hero has just returned from a week at Comdex where his head has been filled with all the latest gizmos and gadgets

The Design

- Monday afternoon
 - Our hero sets about designing his solution to the Hypothetical Part-Stamping Machine.
 - While our hero is hard at work, what are some of your ideas on how to start this project?

A Plan of Action

- Understand Requirements
- Analyze Available Resources
- Decide What Can Be Accomplished
- Layout 1 Week Schedule
- Start working on the schedule milestones...

Understand Requirements

- Create a software architecture that...
 - Can handle machine control of N part-stamping machines
 - Provide UI through touch-screen and UART
 - Support timer interrupt

Analyze Available Resources

- No hardware to control
- No embedded control systems to play with
- No RTOS software systems to play with
- A networked desktop workstation, running RHL 7.3
- A plan starts to formulate...

Decide What Can be Accomplished

- Time and resource constraints prevent a working model from being considered
- This implies that anything delivered within 1 week period will consist of part-simulation, part-product
- A simple “perfect world” (see physics problems) simulation at the I/O level yields big dividends

1 Week Schedule

- Day 1 : Analyze Requirements and draft design document
- Day 2 : Start implementing design on selected hardware target
- Day 3 : Design running in Simulation mode
- Day 4 : Design running with I/O control
- Day 5 : Design ported to 2nd platform

PS Design Decisions

- Run PartStamper simulator on RHL7.3
- Build as much of core system as possible
- Remove UI from 1-week horizon
- Provide tool for system stimulus for test purposes
- Implement 5 control processes with a command/reply message interface
 - State Machine
 - Part Grabber, Part Pusher, Part Stamper
 - IO

Linux as a Target

- PS Control Processes
 - Heavyweight : process
 - Lightweight : thread
- Message Queue
 - IPC SYSV MSG
 - Pthreads
 - pipes

PS Implementation

- Heavyweight process
- SYSV IPC MSG
- SYSV IPC SHM
- Tcl for tool stimulus

Project Management 101

- What is the first thing our hero should do now that he is ready to get his hands dirty?
 - Start writing main.c ?
 - Draw some more diagrams ?
 - Fill in some more Pert Chart milestones ?
 - Build project infrastructure ?

Project Infrastructure

- Should be expandable
- Should support the chosen development tools
- Should support multiple users

Development Tools

- CVS : Source Code Control System
- make : Project Automation
- gcc : Host Tool Chain
- Tcl : Tool Command Language
- LTT : Linux Trace Toolkit
- Shell scripts

Recursive Project Make System

- Premise
 - A project structure can be implemented that is portable and expandable
 - Builds or ‘make’ can be performed anywhere in the development tree
 - Dependency information is auto-generated
 - Supports command-line tools

Next Steps

- The design is based on a tasking model, so in order to start implementing the software modules, we need basic tasking support
 - Ability to create processes
 - Ability to message between processes
- SYSV IPC mechanisms are first to be supported

Next Steps

- Replace fprintf with Log mechanism
- Add I/O monitor
 - turn_off
 - turn_on
 - value_of
 - wait_until
- Add test mechanism

Next Steps

- Personalize Tcl test process with application knowledge
 - Sendcmd

The C / Tcl Philosophy

- As espoused by John K. Ousterhout...
 - To make a Tcl application as flexible and powerful as possible, you should organize its C code as a set of new Tcl commands that provide a clean set of primitive operations... The purpose of the C code is to provide basic operations that make it easy to implement a wide variety of useful scripts.

Tcl and the Tk Toolkit
John K. Ousterhout, Addison-Wesley
Page 281

pstcl : sendcmd

- New Tcl command : sendcmd
- Connects pstcl to rest of PartStamper application
- Allows any command message to be generated and sent in the system
- With a message passing software architecture, this single test command allows virtually all aspects of the system to be tested. With a script-capable test harness, regression tests can be created from the engineer's interactive test tool.

Next Steps

- Build I/O simulator so state machine can be tested
- Expand sendcmd to include all PS command messages
- Code hardware control tasks

I/O Simulator

- By using the same technique that all High School and College Physics problems are based on, a very simple and effective solution is found
- When applying all of the forces for an equation, the Physics student is always cautioned to assume the force from friction is negligible and set it to 0
- In the Part Stamper Simulator, I/O activation times are negligible, as soon as a control line is toggled, the input values are adjusted.

Next Steps

- Finish coding and testing state machine
- Timing analysis
- Design Changes

Design Changes

- 30 Pitfalls for Real-Time Software Developers...16
 - Using message passing as primary inter-process communication
- I/O task has to disappear
 - I/O monitor calls have to atomize data writes
 - `turn_off`, `turn_on`, `value_of`, `wait_until`

Status update...

- Wednesday afternoon
 - Our hero's first design, with one fairly major adjustment, is up and running on RHL7.3
 - Actual I/O is simulated, all digital activations are instantaneous, with input values adjusted
 - Timing envelope of perfect machine is known
 - What's next, real I/O or new RTOS (VxWorks)?
 - What does the schedule say...

VxWorks Demo

- X86 Target Review
- Host / Target Diagram
- Host Setup
- X-Dev Environment