Controlling Laser Ranging with RTAI-based Real-Time Linux

Evan Hoffman, HTSI
Randall Ricklefs, UT/CSR

Introduction - 1

- Current NASA laser systems use costly proprietary real-time, Unix-like LynxOS for controller systems
- Several real-time variants of Linux are available
- RTAI was chosen, as it is free (open source) and there is an active user community

Introduction - 2

- Advantage of Linux:
  - “Free”, stable, and secure
  - Development and many other tools are available with the distributions
  - Active user community and support
  - Many distributions to choose from
- Disadvantage
  - No vendor to blame if things go wrong :-)

RTAI
(Real Time Application Interface)

- Free, and supported
- Provides for soft and hard real time for interrupts and timers
- Requires modification of a standard kernel
- Does not require major rewrite of existing software
- Linux runs as guest of RTAI – not as bad as it sounds

Other Real Time Options

- RTLinux
  - Commercial and Free version
- Xenomi
  - Emphasis on extensibility and portability
  - RTAI focuses more on lowest possible latency and interrupts
- RTAI seems to be under the most active development of the Free options

Concerns

- RTAI is a number of kernel versions behind the state of the art: RTAI 3.8.1 support kernels up to 2.6.32.11 (2.6.38.2 latest)
- Documentation is less than coherent

Arch Linux

- Advantages
  - Rolling release (latest bug fixes and features during development)
  - Feature rich package manager
  - Extremely customizable, can be as lean as you want it to be
  - Designed for simple configuration structure (BSD style)
- Disadvantages
  - Rolling release (updates require care and review)
  - Requires knowledge about Linux/OpenBSD system management (not a beginner distribution)

CentOS

- Advantages
  - Long term release based on RHEL means security updates are available for years
    - Updating process allows omitting updates for packages such as the kernel
  - Many tools and packages immediately available
- Disadvantages
  - Not lean
  - Slow (re)boot compared to LynxOS
  - Older kernel (2.6.22.1)

Other Linux Options

- Ubuntu/Ubuntu Studio
  - Most popular distro, very user friendly
  - Some versions designated “Long Term Support”
  - Studio: soft real time, not hard real time
  - Bloat
- Debian
  - Reputation for stability
  - Very long release time

At the GGAO 48”...

Major Code Ovehaul using:

* the GTK+ toolkit (GUI)
* RTAI shared memory and FIFOs to communicate between RTAI and regular Linux

MLRS Conversion of existing code
Phase I – LynxOS to Linux

- Replaced a number of library calls
- Fixed some memory violations (not caught under LynxOS)
- Used existing Linux-based CAMAC PCI device driver
- Wrote CAMAC simulator device driver (for Austin)
- Installed patches for Blueheat serial driver into Linux serial driver module (kernel patch)
- Handled priority numbering differences from LynxOS (still in progress)

MLRS Conversion of existing code
Phase II – Linux to RTAI Linux

- Added RTAI kernel patches
- Added “helpers” to each RT thread to use RTAI scheduler in “user” mode
- May need to convert from POSIX to RTAI semaphores
- May require device driver modifications for real time

Current Status – GGAO 48”

- Conversion to RTAI is part of an overall hardware and software modernization effort
- Software being developed to run with PCI bus and without the CAMAC interface
- Currently some analog signal controls require hard real time

Current Status - MLRS

- Controller software suite can now run under RTAI Linux
  - Serial and CAMAC I/O and interrupts working properly
    - Met sensors, radar, handpaddle
    - clock, timers, laser, detector
  - Hardware-simulated ranging (100% returns) works well
  - Have tracked stars and satellites
  - Cals look OK, but no satellite data yet (1 pass attempted)
- Still have issues with hard real time

Conclusion

- Linux and RTAI appear to be a good match for SLR controller upgrades
- Updates so far have been only slightly painful for various reasons
- More at the next workshop!