LinuxCNC Real-Time Evaluation

Stephen Decker, Jeremy Tang, Ethan Glassman

LinuxCNC: Background

- “LinuxCNC (the Enhanced Machine Control) is a software system for computer control of machine tools such as milling machines and lathes.”
  - -- linuxcnc.org
- Open source (GPL, LGPL)
- Runs on Linux with real-time extensions
  - Distribution provided by the LinuxCNC site is a patched Debian Wheezy

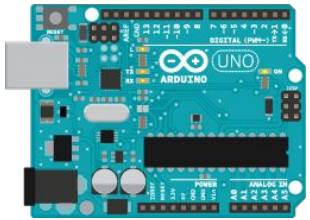
Project Proposal

- We intend to evaluate real-time performance of the LinuxCNC operating system environment on a standard desktop PC.
- We intend to use a Raspberry Pi/Arduino as a connected peripheral that will perform calculations on a data set and return the result of this calculation as a response.
- In this way we can test the real-time performance of the kernel while performing I/O operations without needing a large, complex, and dangerous real load.

Debian Linux Distribution With Real-Time Kernel

Sends directions to workload

Receives results from workload



Microcontroller

Possible Data Sets

- GCode (Includes next cartesian coordinate, speed)
  - Calculate time - to - position given desired machine speed, return true after delay when coordinate is “reached”
- Historic stock trading data
  - Pi/Arduino as real-time data source, desktop to calculate sell/buy
- Basic mathematical workload
  - Calculate primes & return or similar
- Video game server/client simulation
  - Pi/Arduino as game clients with varying “connection quality”, desktop as game server.
  - Keep track of player positions at all times.

Stretch Objectives

- Real-time performance comparison of different interfaces (Ethernet, USB)
- Measure scheduling latency of the Linux environment that LinuxCNC uses (Hourglass, Cyclictest)
- Determining specific circumstances when LinuxCNC does not maintain real time scheduling
- Create recommended steps for how LinuxCNC can be modified to maintain real time performance

Timeline/Goals

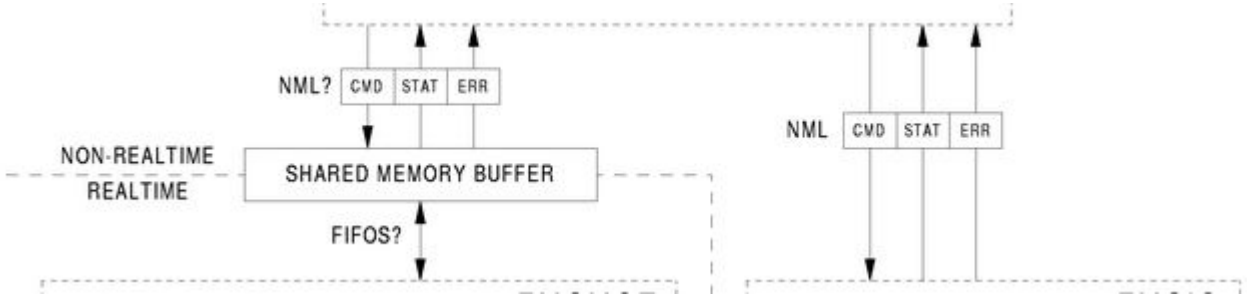
- Pre Planning: Selecting a good data set and simulated load calculation. Complete by 9/25
- Hardware Selection and Setup: We have not yet selected between the Raspberry Pi and an Arduino. We need to finalize our hardware setup and purchase cabling, install LinuxCNC, etc. Complete by 10/2
- Initial Implementation: Code simulated load calculation on Raspberry Pi/Arduino. Setup desktop with data set and I/O operations. Complete by 10/9
- Real-time evaluation setup: Code desktop with software to time evaluation of I/O performance using high accuracy timing APIs (e.g. rtdsc), and begin to collect data on timing parameters. Complete by 10/16
- Prepare demo 1: Presentation of progress so far. Presentation on 10/21

Timeline/Goals (cont.)

- Real-time evaluation: Continue to collect data and refine desktop code. Perform ~1 million tests to determine if schedule is ever missed. If schedule is missed, determine a best guess of what is causing the scheduling to be missed and during what scenarios this might occur. Complete by 11/6
- Prepare demo 2: Presentation of progress so far. Presentation on 11/9 or 11/11
- Finalize real-time evaluation: Complete any lingering tests, write up results. Complete by 11/27
- Prepare final demo: Presentation of progress throughout class. Presentation on 11/30

Challenges

- Real-time performance of the workload
  - We want to test the performance of the LinuxCNC operating system
  - NOT the workload
- Picking a realistic data set and calculation
  - We want to realistically emulate the kinds of workloads that LinuxCNC is used for
  - NOT just crunch numbers arbitrarily
- Work off the provided system architecture diagrams to ensure we are working in the real-time components of the system



Questions?