

PC104 with embedded Linux

## Trips and Ticks

S. Herb, MSC, Desy

Tine Workshop, Sept 2007

# TINE Servers: Desktop vs. Embedded

**History:** Main Line of Development was with Desktop PCs

DOS Win3.1 WfW WinNT WinXP ... Wista?

*(but also significant Unix, vxWorks ...)*

## Desktop Advantages:

- Same environment on Servers, Consoles
- Inexpensive hardware, software,
- Powerful GUI with associated tools
- .....

# TINE Servers: Desktop vs. Embedded

## Desktop Disadvantages:

- Powerful GUI (complexity, bloat and vulnerabilities)
- Complicated Update management (esp. remote)
- Tied to rapid consumer OS/hardware cycles

*Example – oscilloscope with Win2000 as virus infection source*

## My solution: back to basics with 'PC104'

- x86 Architecture, ISA, PCI bus
- Stable industrial standard
- Lots of CPU, I/O cards available

# (Our) PC104 configuration

## Doesn't have:

- GUI, Browser, email, RealPlayer, Excel, JVM, compiler
- Case Fan, CPU Fan, Hard disk

## Does have:

- x86 CPU, 133 MHz (*new version 400 MHz, ~5 Watts*)
- 32 MB IDE Flash Disk, 32 MB DRAM (*new version 256 / 256*)
- Serial Ports, USB (*new version has USB Stick boot option*)
- Hardware watchdog
- 2 Channel CANbus add-on card
- Embedded Linux OS (2.4, 2.6, Sysgo ELINOS distribution)  
(needs about **6 MB** of space on Flash Disk)
- Root partition mounted 'Read-Only'
- TINE Server, autoproc, remote (**< 1 MB**),
- ssh with tcp wrappers (*remote access*)
- ~ 2 years operation experience ( 7 devices in FLASH)

# PC104: some Issues

- How do I mount it?

PC104 box, 1U rack box, DIN rail, crate? *No really great solution*

- Where do I get 5V?

*depends on above (POE ?? but it's 50V)*

**TRIP:** be wary of accepting free voltage from strangers

*( your server should be **On** even when the hardware is **Off**)*

**TICK:** your server should be powered during machine shutdowns  
*(and as an aside, if the hardware is non-functional, you should write a **hardware simulator** so that the control system team can make the most of the valuable shutdown time)*

# PC104: some Issues

- Is it bulletproof (and does it need to be?)

- Embedded systems often have **more hw/sw bugs** than desktop systems -- because they have fewer users !!
- Lab tests usually miss aspects of real operation

**TRIP:** Log files showed that most of the operational PC104 servers were rebooting ~ once/2 weeks. This was traced to the ethernet interface, probably under heavy loads. It (*fortunately!*) disappeared when the interface was constrained to 10Mb ½ duplex.

**TRIP:** Watchdog hangs when card voltage falls to 3V and returns to 5V (not probable, but what if you had to drive 50 km to push *reset* ?)

- ☑ **TICK:** If you need high reliability, test for it! Don't assume that 'Industrial Quality' will automatically solve your problems.

# PC104: some Issues

- **Linux Configuration:** dozens of possibilities!  
**Roll your own?** Possible, but what about maintenance?  
Are you dependent on a single linux kernel expert in your group?
- ☑ **TICK:** Commercial distributions provide tools which help in structuring the project, and are suited to group involvement
- **Cross-Development Tool Chain?** (compilation ...)  
Useful: develop on desktop Linux machine without need to synchronize desktop OS with version of embedded system OS  
*[ but it may only be possible to drive hardware from the PC104 ]*
- ? With new PC104 cards an alternative could be to couple them to a hard disk (or NFS disk), for development and debugging on the PC104 itself *(but not with heavyweight tools like Eclipse!)*

# PC104: some Issues

- **Linux Drivers for PC104 I/O cards**

- Many companies are still somewhat nervous about open source
- Linux versioning requires driver recompile for each version

**TICK:** Get some experience with the drivers and support from the manufacturer before you commit to a particular solution

- **Our Solution**

- CANbus cards from 'Peak Systems' include driver source with compatible interfaces for ISA (PC104) and USB (desktop)

=> We can perform development **and testing** on a Linux **desktop** using **the same source code** (for TINE as well as for the driver) as on the **PC104**. The only difference is a few lines in the PC104 Makefile which point to the cross-compile directories.

**TICK:** Try to find hw and sw combinations which give you an efficient and comfortable development environment

# PC104: Conclusions, comments

- I find this alternative for front end servers very attractive. But it takes some effort to put together a system which is both stable and convenient !!
- Once a stable system is in place, it requires very little maintenance or other attention.
- Remote operation and maintenance has worked well, with almost no local access required. TINE has lots of features which help with remote testing and debugging.

**TICK:** Google is your friend:

pcan.o: pcan.o: unresolved symbol register\_chrdev\_R2e77096d

pcan.o: pcan.o: unresolved symbol add\_wait\_queue\_R24d317ed

pcan.o: pcan.o: unresolved symbol \_\_pollwait\_R87156f10

+ ~available Friday (or whenever) for consultation ( x3049, Building 30/520)