ECE 471 – Embedded Systems Lecture 22

Vince Weaver

http://www.eece.maine.edu/∼vweaver

vincent.weaver@maine.edu

21 November 2013

Announcements

- Project groups status report due after Thanksgiving.

Low-Level Linux/Project digressions

Wii Nunchuck

- Fairly easy to interface
- Put onto i2c bus. Device 0x52
- Send handshake to initialize. Use longer one (0xf0/0x55/0xfb/0x00) not the simpler one you might find(0x40/0x00). This works on generic nunchucks and possibly also disabled encryption of results.
- To get values, send 0x00 and read 6 bytes. This includes

joy-x, joy-x, accelerometer x/y/z and c and z button data. More info can be found online.

Linux and Keyboard

- Old ps/2 keyboard just a matrix of keys, controlled by a small processor. Communication via a serial bus. Returns ”keycodes” when keypress and release and a few others.

- Many modern keyboards are USB, which requires full USB stack. To get around needing this overhead (for BIOS etc) support bit-bang mode. OS usually has abstraction layer that supports USB keyboards same as old-style

- Linux assumes “CANONICAL” input mode, i.e. like a teletype. One line at a time, blocking input, wait until enter pressed.

- You can set non-CANONICAL mode, async input, and VMIN of 1 to get reasonable input for a game. Arrow keys are reported as escape sequences (ESCAPE-[-A for up, for example).

- Even lower-level you can access “RAW” mode which gives raw keycode events, etc.

- There are libraries like ncurses that abstract this a bit.

Also GUI and game libraries (SDL).

Other Digressions

- There was some talk of VMs (KVM on Linux)
- Also talk of Plan 9 Operating System

A History of Power Management on x86

Halt Instruction

- Oldest power-saving interface on x86
- Tells CPU to stay idle until an interrupt comes in
- 486-DX4 and later enters low-power mode
- Ring 0. The OS does this when idle
- Similar instruction available on 65c816
- ARM has `wfi` in ARMv7 and maybe `hlt` in ARMv8?

APM – Advanced Power Management

- For laptops
- Developed by Intel and Microsoft, 1992
- Made obsolete by ACPI
- Full On / APM Enabled / Standby / Suspend or Hibernate / Off
- Calls to BIOS. BIOS often buggy.

ACPI – Advanced Configuration and Power Interface

- http://www.acpi.info/presentations/ACPI_Overview.pdf
- Developed by Intel, Microsoft and Toshiba, 1996 Later HP and Phoenix
- Full ACPI interpreter needed.
- APM was a black box to Operating System. ACPI works with OS

- ACPI code in theory provided by Intel or similar, no need for each manufacturer to implement (like APM)
- OS-directed power management
- Hardware registers for interface
- BIOS provides tables, motherboard initialization

ACPI Sleep States

- G0/S0 – working
- G1 Sleeping
  - S1 – caches flushed, CPU stopped, CPU and RAM power maintained
  - S2 – CPU powered OFF
  - S3 – Standby, Sleep, Suspect to RAM. RAM still powered
  - S4 – Hibernate/Suspend to Disk – all memory stored to disk

- G2 (S5) – “Soft Off” – power off, but power still supplied to power switch and wake on lan, etc

- G3 – “Mechanical Off” – all power removed

ACPI C-States

- C0 – operating
- C1 – Halt – processor not executing, but can start nearly instantaneously
- C2 – Stop-Clock – all state is stored, but might take some time to get going again
- C3 – Sleep – Processor does not keep cache coherent, but otherwise holds state

ACPI P-States

- actual values can sometimes be configured via MSR access.
- Some V/F combinations unstable/unsafe so BIOS only exports known good combinations
- P0 – max power and frequency
- P1 – less than P0, DVFS
- P2 – less than P1, DVFS

- Pn – less than P(n-1), DVFS

ACPI T-States

- throttling
- Linear reduction in power, linear reduction in performance
- Does not save Energy! (halve the frequency, double the time)
- Mostly used for passive cooling

ACPI D-States

for devices such as modems, Cd-ROM, disk drive

CPU Scaling

- Intel SpeedStep
- Enhanced speed step. Change V and F at different points. Slower to change frequency if V not changed first. Bus clock keeps running even as PLL shut down 10ms transition
- AMD PowerNow! (laptop) Cool'n'Quiet (desktop)
- VIA PowerSaver/LongHaul – Fine grained DVFS

- p4-clockmod – mainly for thermal management, skip clocks, hurt performance without saving energy (throttling)

- IBM EnergyScale

- Transmeta LongRun – leakage varies due to process variation Longrun2 monitors performance/leakage and varies Vdd and Vt

Governors

- ondemand – dynamically increase frequency if at 95% of CPU load

  introduced in 2.6.9

- performance – run CPU at max frequency
- conservative – increase frequency if at 75% of load
- powersave – run CPU at minimum frequency
- userspace – let the user (or tool) decide

Governors – cont

- Various tunables under /sys/devices/system/cpu
- Can trigger based on ACPI events (power plug in, lid close)
- Laptop tools
- `cpufreq-info` and `cpufreq-set`
  Need to be root

User Governors

- typically can only update once per second
- ondemand people claim it reacts poorly to bursty behavior
- Powernowd – scale based on user and sys time
- cpufreqd
- Obsolete with introduction of "ondemand" governor?

Sources of Info for Governors

- System load
- performance counters
- input from user?

TurboBoost

- Nehalem/Ivy Bridge/Sandy Bridge (AMD has similar Turbo CORE)
- Some Core2 had similar “Intel Dynamic Acceleration”
- Kicks in at highest ACPI Pstate
- “Dynamic Overclocking”

TurboBoost – from HotChips 2011 Slides

- Monitors power, current, thermal limits, overclocks
- 100 uarch events, leakage function of temp and voltage
- P1: guaranteed stable state
  P0: turbo boost, maximum possible
- 12 temp sensors on each core
- PECI – an external microcontroller, used to control fans, package power

TurboBoost example

- From WikiPedia Intel_Turbo_Boost article
- Core i7-920XM
- Normal freq 2.0GHz
- 2/2/8/9 – number of 133MHz steps above with 4/3/2/1 cores active
- 2.26GHz, 3.06GHz, 3.20GHz

Non-x86 Power Saving

IBM EnergyScale

- Thermal reporting
- Static and Dynamic Power Save
- "Power Folding" – reduce the number of CPUs reported to the OS until they are all busy
- Power Capping (like RAPL)
- Fan Control – Avoid "over-cooling"

- Processor Nap – 2ms to wake up
- Processor Winkle (as in Rip Van) – 10-20ms to wake up, 95% of power

ARM Cortex A9 (Pandaboard)

- Cortex-A9 Technical Reference Manual, Chapter 2.4 Power Management

- Energy Efficient Features
  - Accurate branch prediction (reduce number of incorrect fetch)
  - Physically addressed caches (reducing number of cache flushes)
  - Use of micro TLBs

  - caches that use sequential access information? reduce accesses to tags
  - small instruction loops can operate without access icache

- Potentially separate power domains for CPU logic, MPE (multi-media NEON), and RAMs

- Full-run mode

- Run with MPE disabled

- Run with MPE powered off

- Standby – entered with `wfi` instruction. Processor mostly shutdown except part waiting for interrupt
- Dormant – caches still powered
- Shutdown

Pandaboard Power Stats

- Wattsuppro: 2.7W idle, seen up to 5W when busy
- http://ssvb.github.com/2012/04/10/cpuburn-arm-cortex-a9.html
- With Neon and CPU burn:

Idle system	550 mA	2.75W
cpuburn-neon	1130 mA	5.65W
cpuburn-1.4a (burnCortexA9.s)	1180 mA	5.90W
ssvb-cpuburn-a9.S	1640 mA	8.2W