6.893 Pervasive Computing

Larry Rudolph

Course Overview

- Two sets of students -- MIT and SMA
- Two parts to class
  - Individual technologies (PS & Quiz)
  - Group projects
- Materials
  - iPaq, backpaq, cricket, ??
  - Slides, handouts, notes (raw)
  - TA's -- filter for your questions
    - Albert @ MIT, ???@ SMA

What is pervasive computing?

- Post PC -- PC not the center
- Digital devices all around us
- Ubiquitous Computing
  - Mark Weisner -- Calm Computing

Oxygen Vision

To bring an abundance of computation & communication
within easy reach of humans
through natural perceptual interfaces of speech and vision
so computation blends into peoples' lives
enabling them to easily do tasks they want to do:

- ***collaborate,***
- ***access knowledge,***
- ***automate routine tasks***

Pervasive, Human-Centric Computing

- What do these words mean?
- Computers are already pervasive
  - even in Boston and Singapore
- Computers are already human-centric
  - are they for the birds?
- It’s not really about computing
  - we already know how to do that

So, what do we mean?

- Pervasive
  - Should be where we need them
    - not have to go to them or set them up
- Human-centric
  - Computers should adapt to humans
    - computation enters our world/environment
- Computing
  - Computer-mediated function
    - digital media

Look back to see ahead

- Monolithic Programs & Hardware
- Decompose into interactive pieces
  - Compose to build large thing
- Continue decomposing into autonomous, interacting components

Finding and naming stuff

- Few items
  - Use list
- Many items
  - Use heirarchy
- Very many items
  - Use multi-index

Organization of material

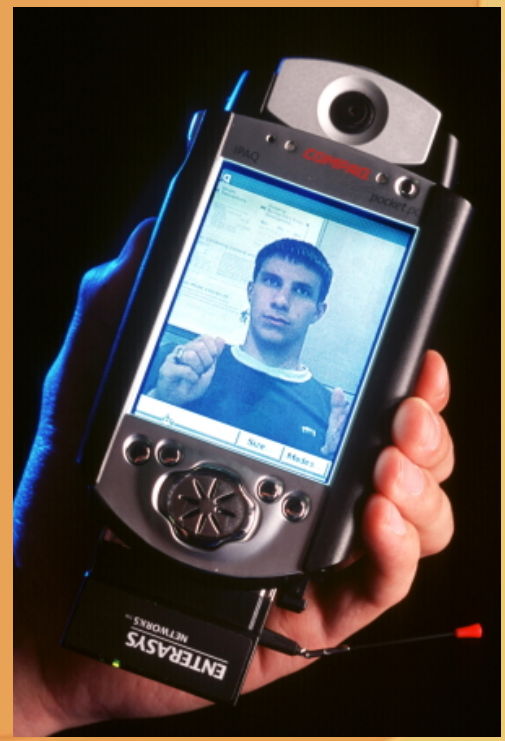
- Top-down
  - would be nice to start writing apps
  - but we are not there yet
- Bottom-up
  - Build on what is known
    - Keyboard, mouse, pen
    - Location, Speech, Multimodal
    - Integrative Technologies

H21 components

- Hardware
  - iPAQ
  - Backpaq
  - Wireless Communication
  - Location sensing

Software

- Linux
- Landcam
- Galaxy Audio Interface
- Cricket Location Reporting



iPAQ 3870

3870 iPAQ

- 206 MHz Strong Arm
- 64 Mbytes SDRAM
- 32 Mbytes flash storage
- Bluetooth
- SD/MMC card slot
- 16 bit color display

5500 iPAQ

- 400 MHz Xscale
- 128 Mbytes SDRAM
- 48 Mbytes flash storage
- Bluetooth & WiFi
- SD/MMC card slot
- 16 bit color display

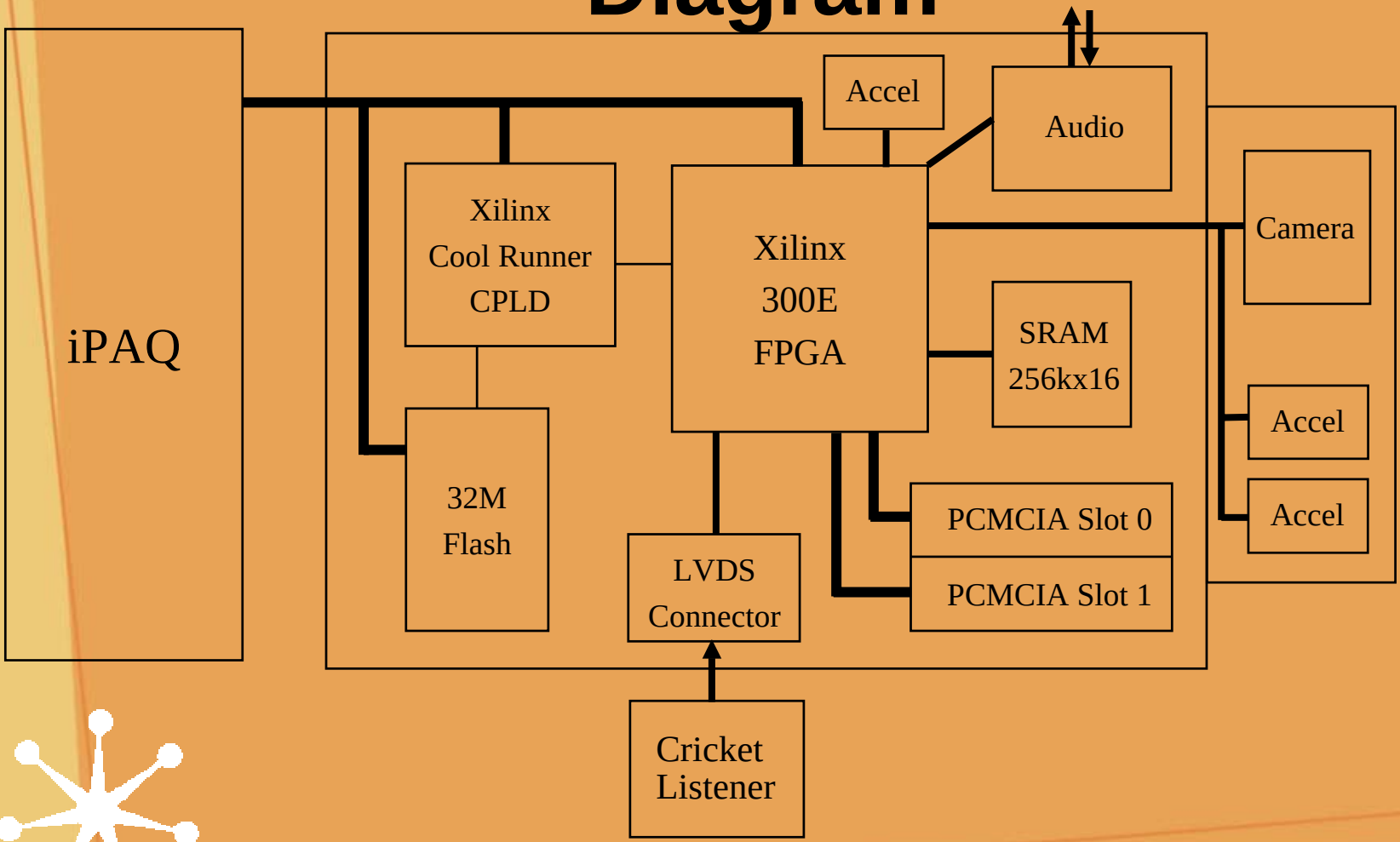
Linux on H21

- Why Linux?
  - Linux allows full access to all software
  - Common development with desktop
  - Can use open source code from many sources
- Porting Linux to a handheld device
  - More difficult than standard PC or Laptop
    - Non-standard interfaces (screen, control FPGAs, touch screen, …)
    - Requires rewritable Flash ROMs
    - For iPaq, port done by HP's Cambridge Research Lab

H21 Backpaq

- Redesigned BackPaq (Version 3)
  - Philips imager (640x480 CMOS color imager UPA1022)
  - Larger FPGA (Xilinx Virtex 300E)
  - 256k x 16 SRAM
  - Lower power
  - 3-axis accelerometer in camera housing
  - 2-axis accelerometer in Backpaq
  - Dual PCMCIA slots
  - Audio input/output codec and headset jac
  - 32 MBytes Flash in Backpaq
  - LVDS Connector from FPGA pins
  - Lion Battery

V3 Backpaq Block Diagram



Philips Imager

- Philips UPA1022 Imager
  - 640x480 CMOS
  - Improved image processing
  - 9 bits/pixel/color
  - Fixed focus optics

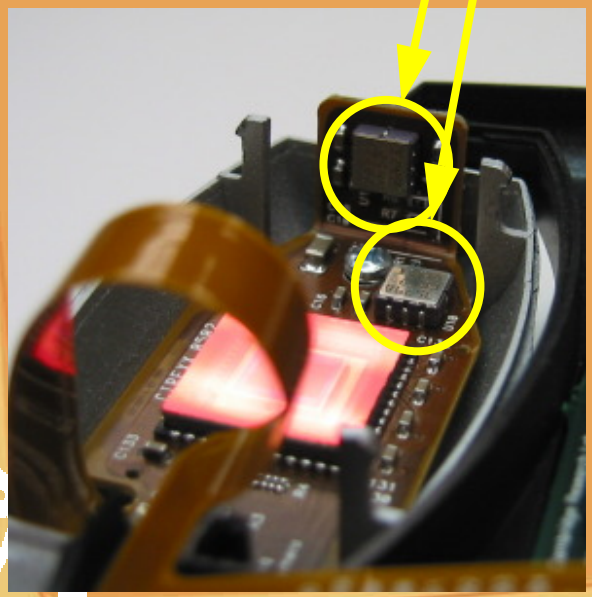
Additional Accelerometers

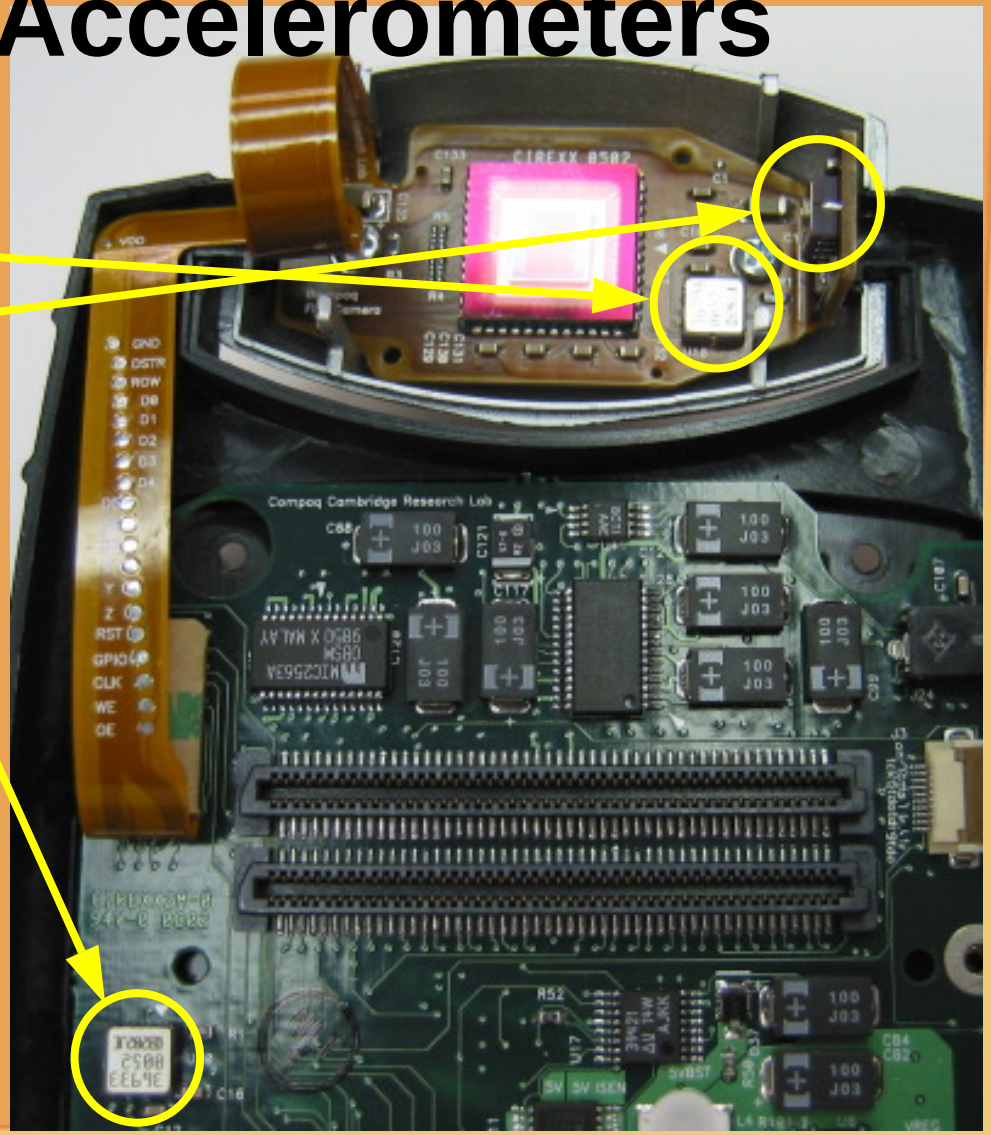
Three Accelerometers:

One in camera plane

One perpendicular

One on Backpaq PCB





Accelerometer Linux Devices

- 2-Axis accelerometer (on main PCB)
  - `/dev/backpaq/accel`
  - Each read returns X and Y acceleration values
- 2-Axis accelerometer (on camera PCB)
  - `/dev/backpaq/cam_accel`
  - Each read returns X and Y acceleration values
- 3-Axis accelerometer (in camera housing)
  - `/dev/backpaq/cam_accel_xyz`
  - Created from 2 perpendicular 2-axis accelerometers
  - Each read returns X, Y and Z acceleration values

Reading the Accelerometers

- Linux character device
- Open the device:
  - `fd = open("/dev/backpaq/accel",O_RDONLY | O_NOCTTY);`
- Read from the device
  - `struct h3600_backpaq_accel_data accel_buffer;`
  - `read(fd,&accel_buffer,sizeof(accel_buffer));`
- The structure returned:
  - `struct h3600_backpaq_accel_data {`
    - `short x_acceleration;`
    - `short y_acceleration;`
- From:
  - http://cvs.handhelds.org/cgi-bin/viewcvs.cgi/apps/backpaq/oneko/

Audio Headset Jack

- Mono-input / Mono-output
  - Same codec as iPAQ
  - Driver and FPGA code under development now
  - Cell phone headset connector (2.5mm)

SRAM

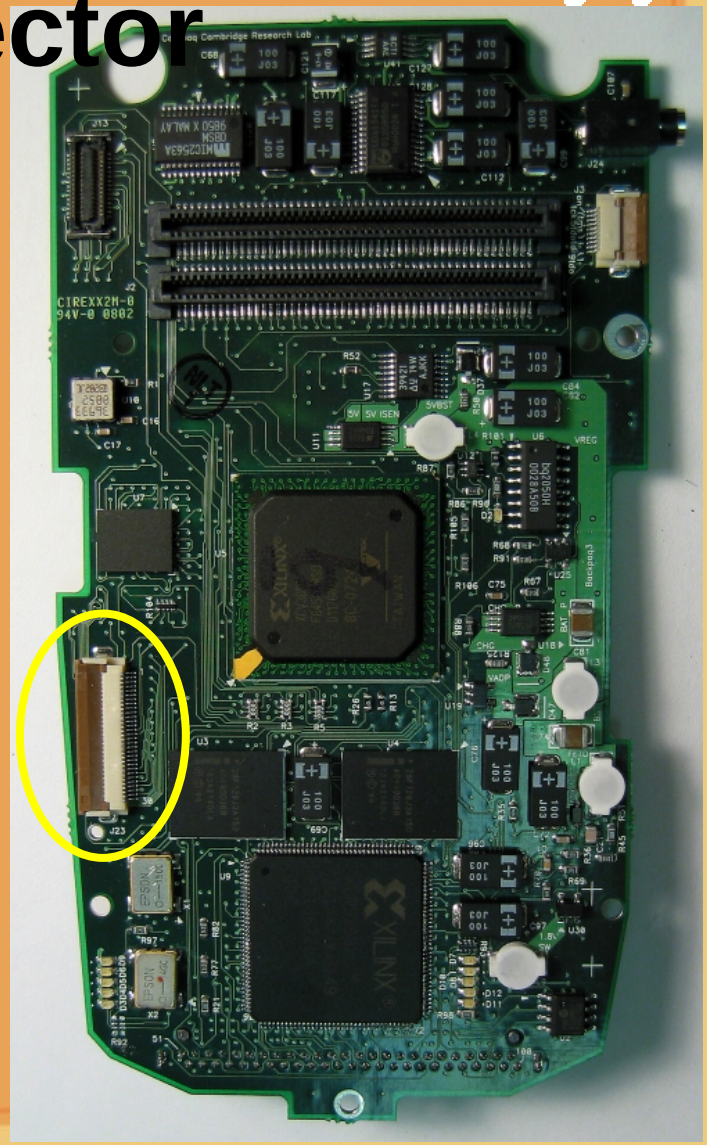
- 64K 16-bit words
  - Cypress CY7C1021BV-33-10BAC
  - 10ns Asynchronous SRAM
- Connected to FPGA
- Planned for use in image processing on FPGA

PCMCIA Slots

- Dual PCMCIA Slots
- 16-bit cards (Not CardBus) supported today
- Pins driven directly from the FPGA
  - Hardware supports CardBus cards
  - Could implement CardBus controller in FPGA

LVDS Connector

- Low Voltage Differential Signaling
  - 10 differential signal pairs
    - **or 20 single signals**
  - Driven from Xilinx 300E FPGA
  - 5 Power and 5 ground pins
- 30 pin flex cable ZIF connector



Backpaq FPGA

- Controls Backpaq functionality:
  - Camera
  - Accelerometers
  - PCMCIA Slots
  - Audio
  - LVDS Connector
  - SRAM
- Reprogramable
  - cat fpga.bin > /dev/backpaq/fpga
  - Automatically loaded from (/etc/fpga.bin) on each power cycle
  - Open source VHDL (8K lines)
  - http://cvs.handhelds.org/cgi-bin/viewcvs.cgi/mercury/backpaq/fpga_fifo_test/

Xilinx XCV300E

- Package XCV300E-FG456
  - Wire-bond Fine-pitch BGA (1.0 mm ball spacing)
  - 23mm x 23mm
  - 312 pins
  - 1.8V
  - 0.18 μm six layer metal process
- Logic
  - 412K System gates (20-30% used as RAM)
  - 32 x 48 CLB Array (3072 slices)
  - 6912 Cells (4-input LUT + Register)
  - 6144 Flip-flops
  - 128K ram bits in 32 blocks + max 96k distributed memory

References

- Oxygen Overview
  - http://oxygen.lcs.mit.edu
- Linux on handheld devices
  - http://handhelds.org
  - http://oxy.lcs.mit.edu
- Example Source Code
  - http://cvs.handhelds.org/cgibin/viewcvs.cgi/apps/backpaq/oneko/
- VHDL for FPGA
  - http://cvs.handhelds.org/cgi-bin/viewcvs.cgi/mercury/backpaq/fpga_fifo_test/

Oxygen Integration Scenario Demonstration

- Video Conference Migration

Location Aware Computing

- Computer:
-
- Knows where it is
- Uses the best available resources
- Automatically transitions between resources

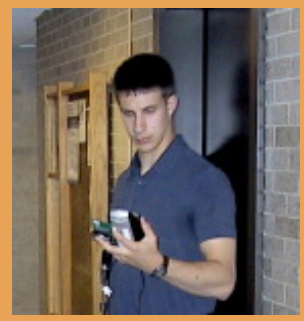
Mobile Video Conference

Best Available Resour

Outside

- Cell modem
- Audio conferencing on H21

Inside



- 802.11b wireless network
- Video conferencing with H21 camera
- Audio conferencing on H21

In E21

- Video projector
- High quality microphone
- Higher frame rate camera
- Wired network connection
- Higher quality compression using faster CPU

Location Awareness

- Goal: Automatically determine location
- Technology:
- Cricket Location System
  - Detect transition into the building
  - Detect transition into the E21
- Accelerometers in H21
  - Detect H21 being placed on the tab
- Future:
  - Network monitoring

Switch Between Network Resources

- Goal: Seamless network transitions
  - Cell modem outside
  - 802.11b wireless inside
- Technology:
- Migrate

Switch Between Physical Resources

- Goal: Transition to best available equipment automatically
  - iPAQ: camera, microphone, speaker, screen
  - E21: camera, microphone, speaker, projector

- Technology:
- MetaGlue

Location Aware Computing

- Computer:
-
- Knows where it is
- Uses the best available resources
- Automatically transitions between resources
- Move around in the real world

One More Time

Future Mobility Technology

- Goal: Self-mobile computing

Technology: Teleportation

Status: Testing on inanimate objects