RobotCub

Open Source hardware and software for robotics

Giorgio Metta

Italian Institute of Technology and University of Genoa

Cogsys
Cognitive Systems

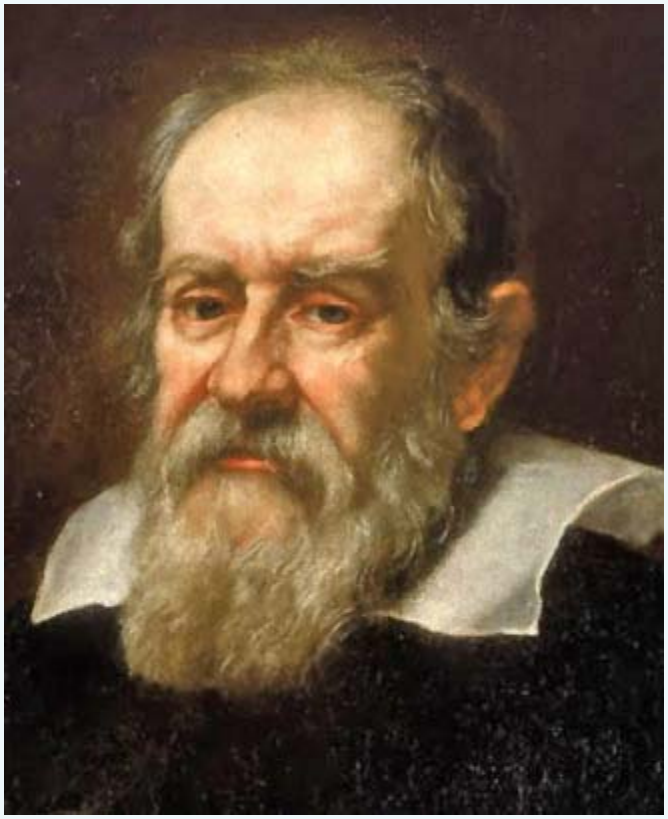
A collaborative project

U. of Genoa/IIT	Scuola S. Anna	U. of Zurich	U. of Uppsala	U. of Ferrara
Giulio Sandini Giorgio Metta David Vernon Lorenzo Natale Francesco Nori Paul Fitzpatrick Darwin Caldwell Nick Tsagarakis	Paolo Dario Cecilia Laschi	Rolf Pfeifer Harold Fernandez	Claes von Hofsten Kerstin Rosander	Luciano Fadiga Laila Craighero Andrey Oleyinik

U. Herthfordshire	IST - Lisbon	U. Sheffield	EPFL	Telerobot S.r.l
Kerstin Dautenhahn Chrystopher. Nehaniv	Jose' Santos-Victor Alexandre Bernardino Ricardo Beira Miguel Praça	John Gray	Aude Billard Auke Ijspeert Sarah Degallier Ludovic Righetti	Francesco Becchi David Corsini Giovanni Stellin

The Hardware

- Goals:
  - Quality design, industry level…
  - … with research-driven specifications
  - Consequences: dexterous manipulation, tendon driven robot, difficult to control, extensive sensorization



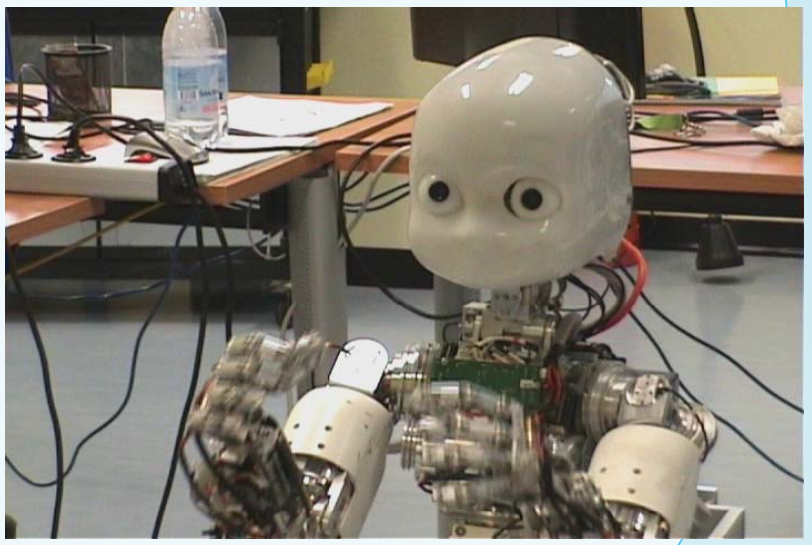
The iCub: quick summary

The **iCub** is the humanoid baby-robot designed as part of the **RobotCub** project

- The iCub is a full humanoid robot sized as a three and half year-old child.
- The total height is 104cm.
- It has 53 degrees of freedom, including articulated hands to be used for manipulation and gesturing.
- The robot will be able to crawl and sit and autonomously transition from crawling to sitting and vice-versa.
- The robot is GPL/FDL: software, hardware, drawings, documentation, etc.

Degrees of freedom

- Head: vergence, common tilt + 3 dof neck
- Arms: 7 dof each
  - Shoulder (3), elbow (1), wrist (3)
- Hands: 9 dof each ► 19 joints
  - 5 fingers ► underactuated
- Legs: 6 dof each
  - Hip (3), knee (1), ankle (2)
- Waist: 3 dof



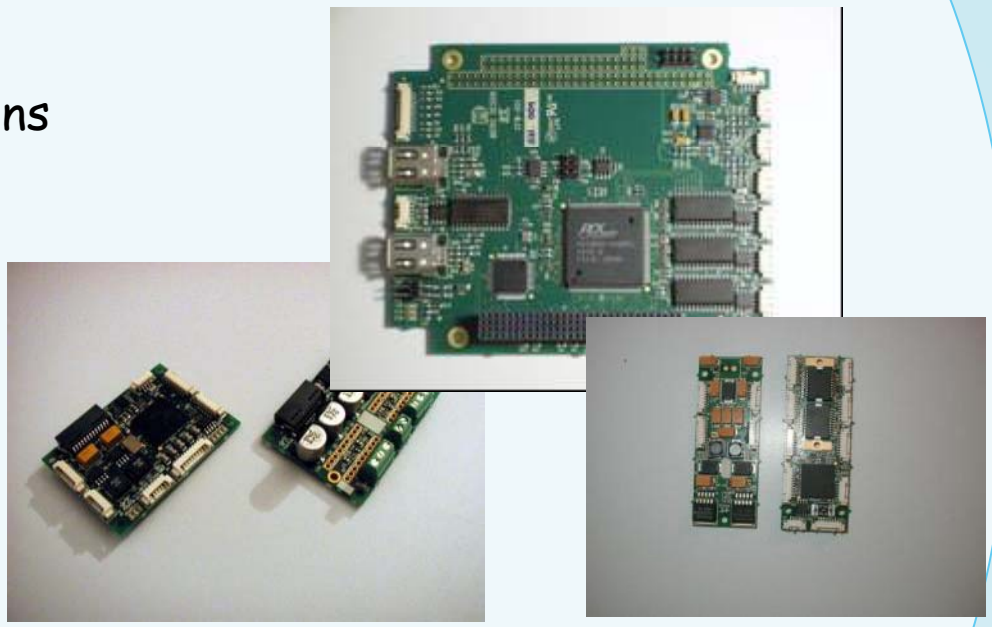
Σ = 53 dof (not counting the facial expressions)

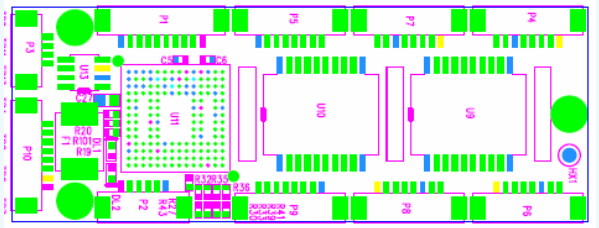
Sensorization

- Absolute position
  - On most joints, AMS magnetic encoder
- Cameras
  - Pointgrey Dragonfly firewire cameras
- Microphones, speaker
  - Standard condenser electrect miniature microphones
  - Pinnae
- Gyroscopes, linear accelerometers
  - Xsense: Mtx

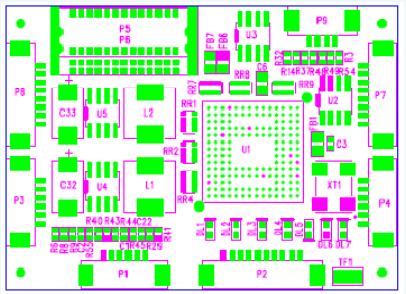
Custom electronics

- ADC card
  - Special connectors (40 pins < 1cm length)
  - 200µm stainless steel wires, coated in Teflon
- Motor control
  - C programmable DSP 40 MIPS
  - Up to 4A DC motor





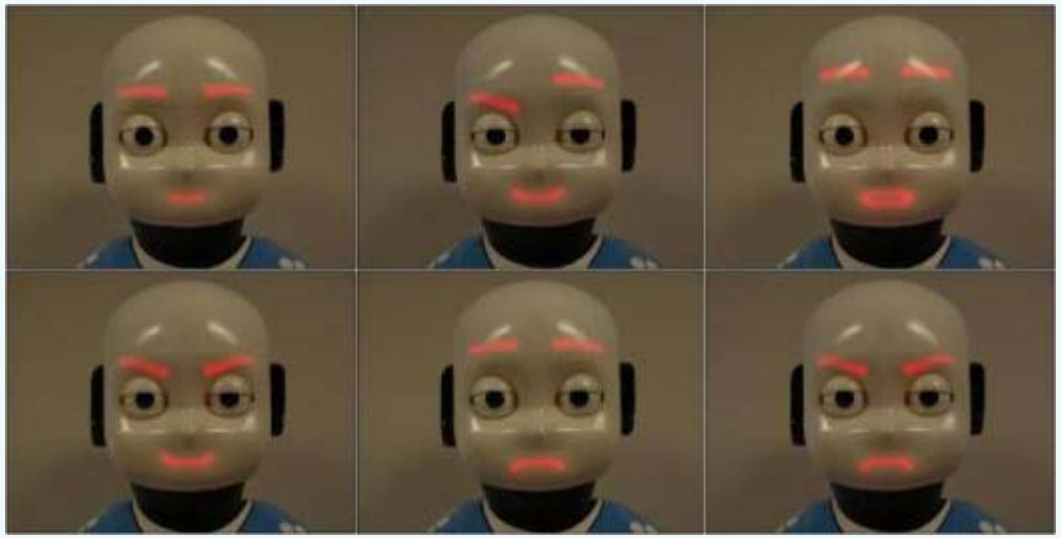
80x30mm

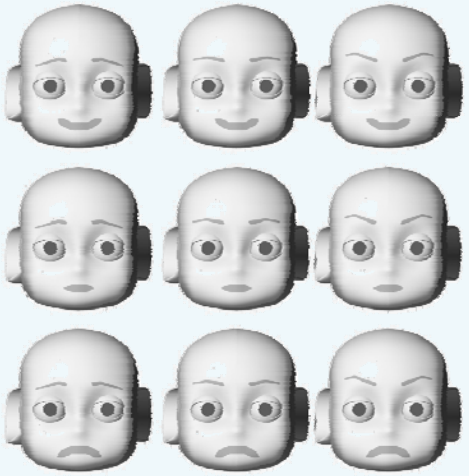


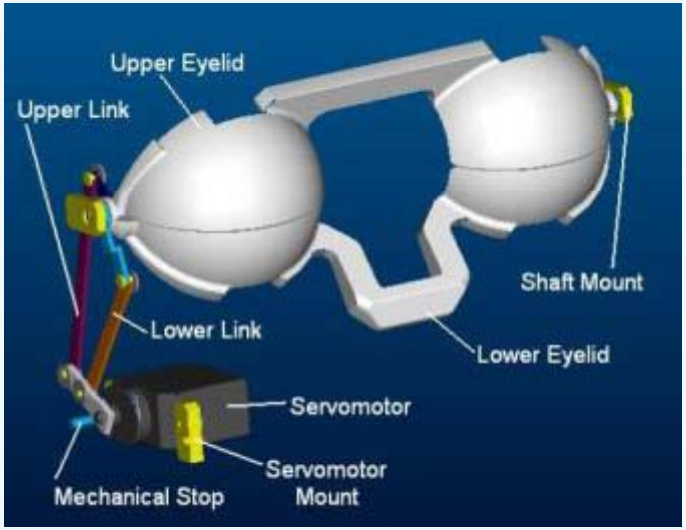
58x42mm

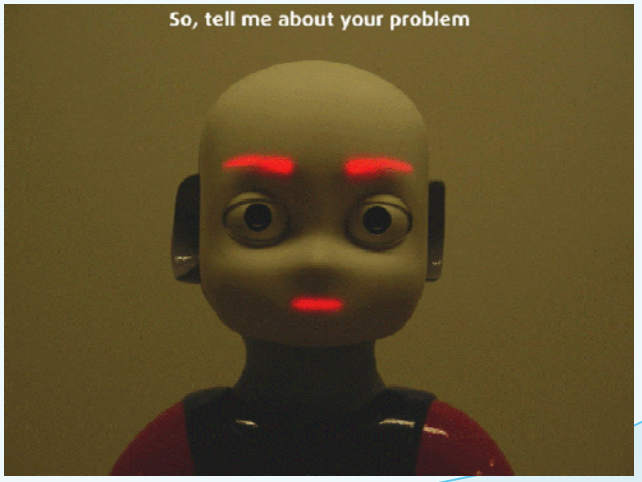
Motorola DSP56F807 (5680x family)
MAC instructions
PWM generation
ADC
Digital I/O
Can bus
C programmable

Facial expressions



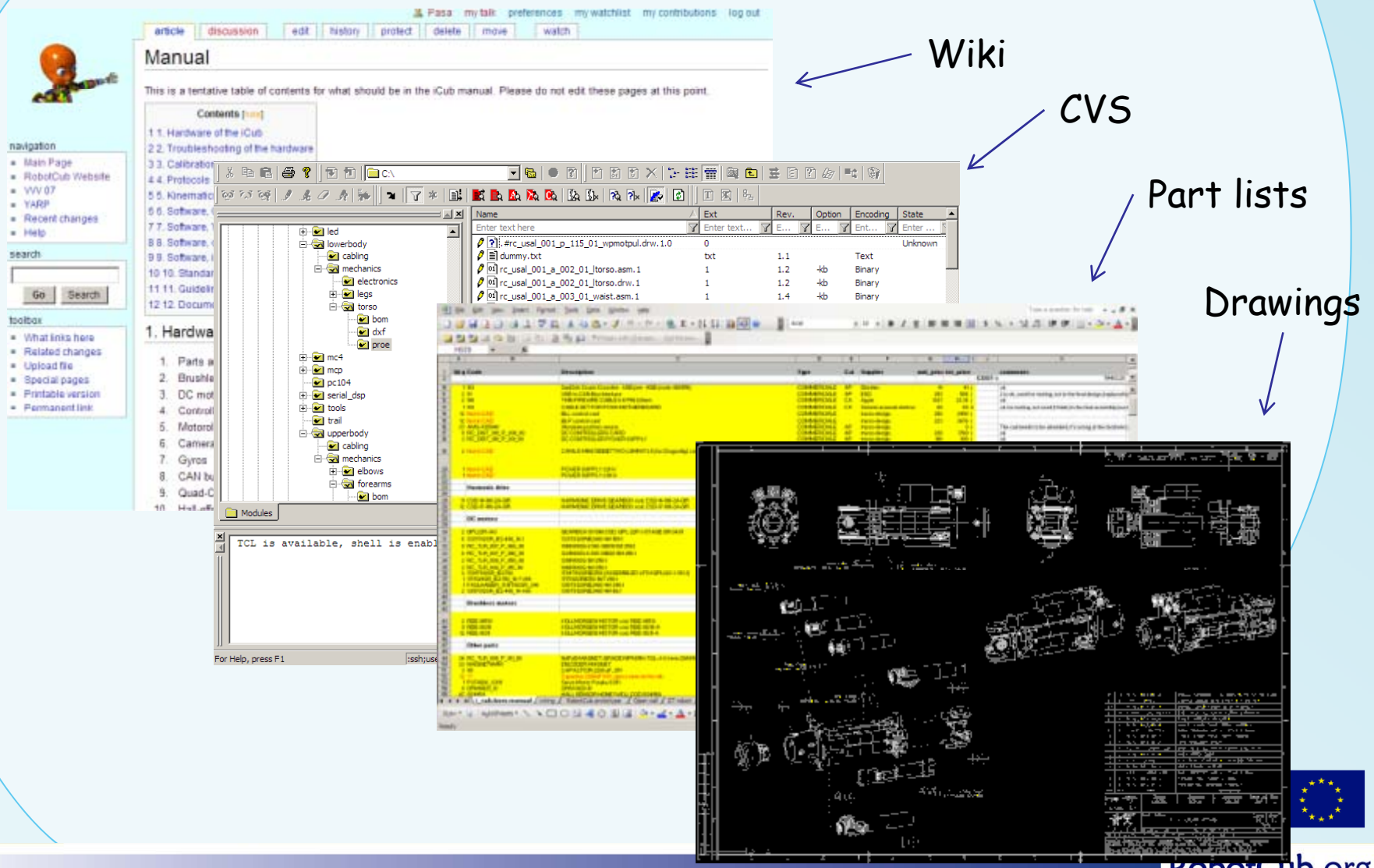






RobotCub

open source



RobotCub

At the end of Y5 we will have 18 working platforms

Cogsys
Cognitive Systems

Promoting the iCub

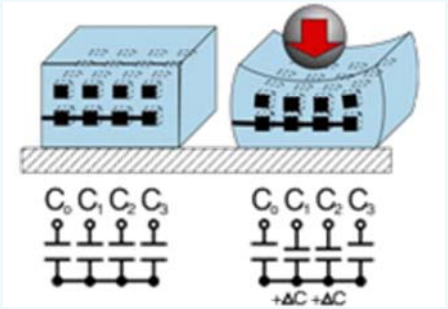
- RobotCub Open Call
  - 31 participants, 7 winners will receive a copy of the iCub free of charge
  - UPMC Paris, Imperial London, Inserm Lyon, TU Munich, METU Ankara, Pompeu Fabra Barcelona, Urbana-Champaign USA, IST Lisbon, EPFL Lausanne
- Further development…
  - EU project ITALK: 4 iCub's have been built
  - EU project ImClever: 3 iCub's will be built
  - EU project RoboSkin: a skin system compatible with iCub
  - EU project CHRIS: safety features for the iCub
- Collaborations
  - Univ. of Karlsruhe: new and longer legs
- Simulator:
  - Open Source simulator based on ODE and as a model in Webots

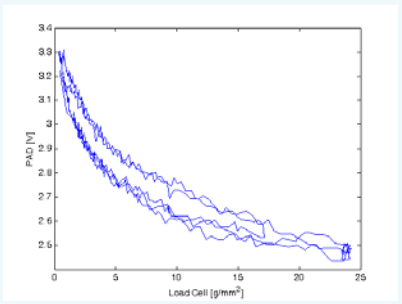
In the pipeline…

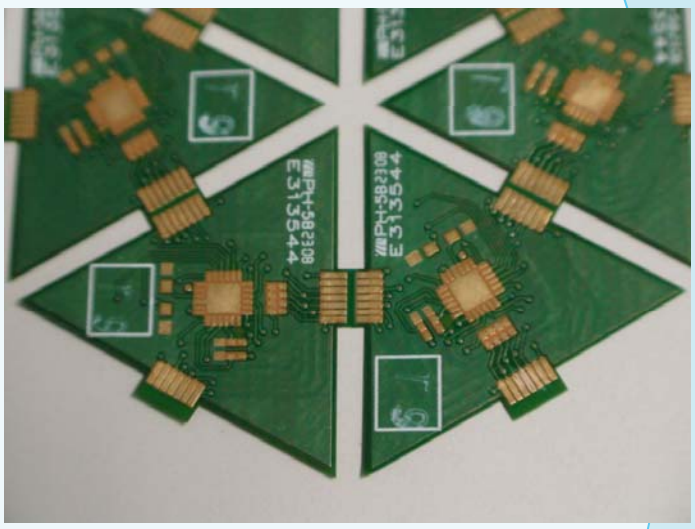
- Force control: joint level sensors, SEA or stain gauges based sensing
- Skin/tactile sensors: almost everywhere on the robot surface
- Robot general improvements: e.g. zero-backlash everywhere, better control electronics, higher resolution position sensors

The skin

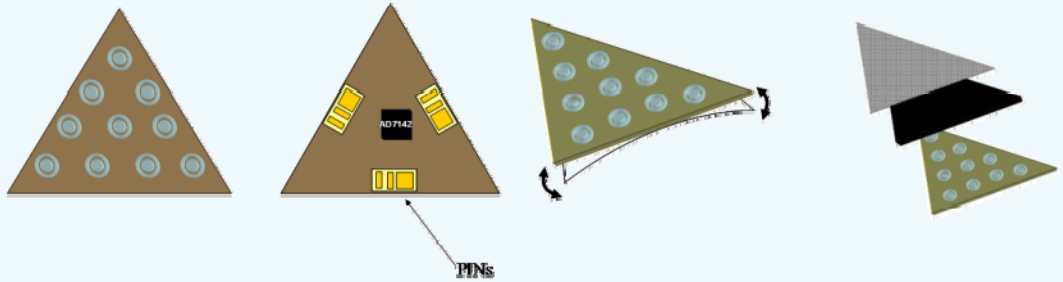
Principle

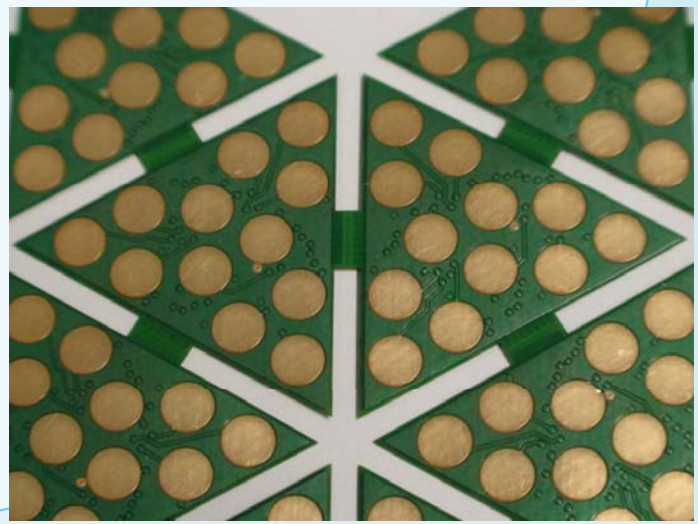




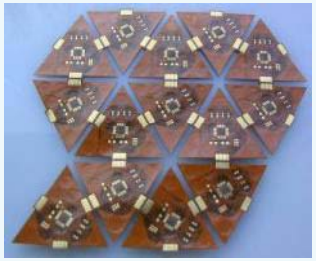


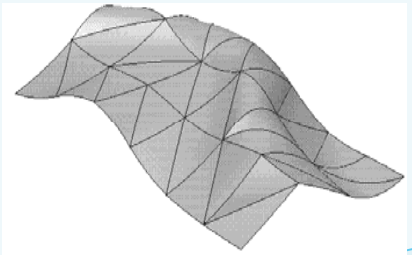
Lot of sensing points



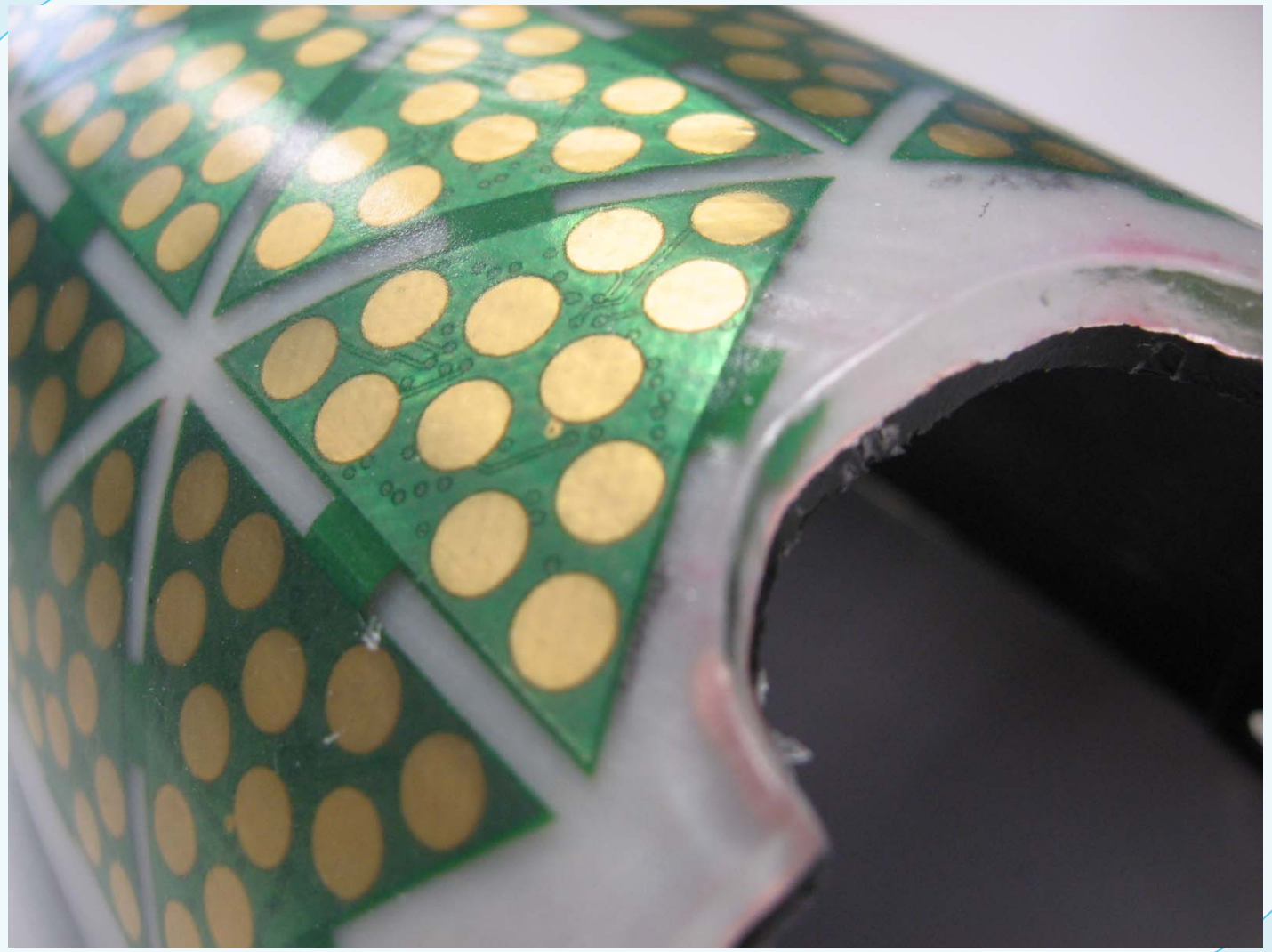


Structure of the skin





RobotCub

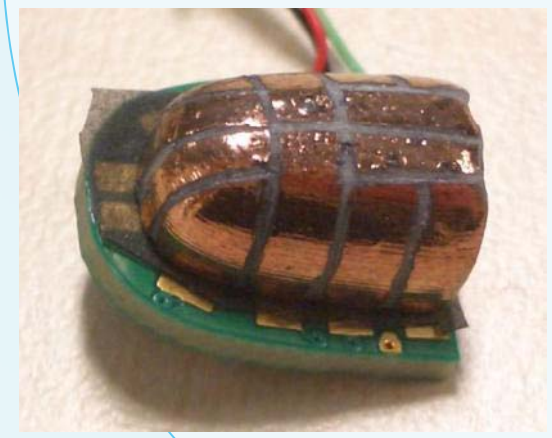


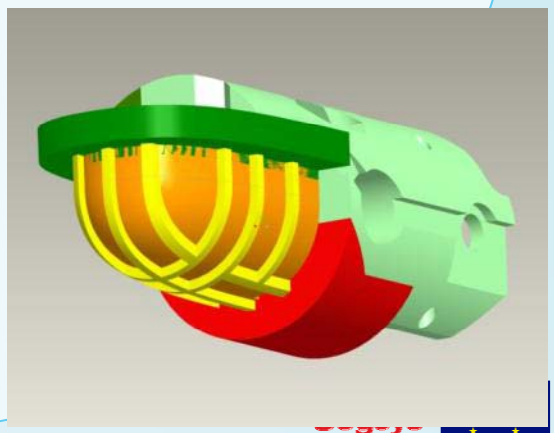
Cogsys
Cognitive Systems

RobotCub.org

Fingertips

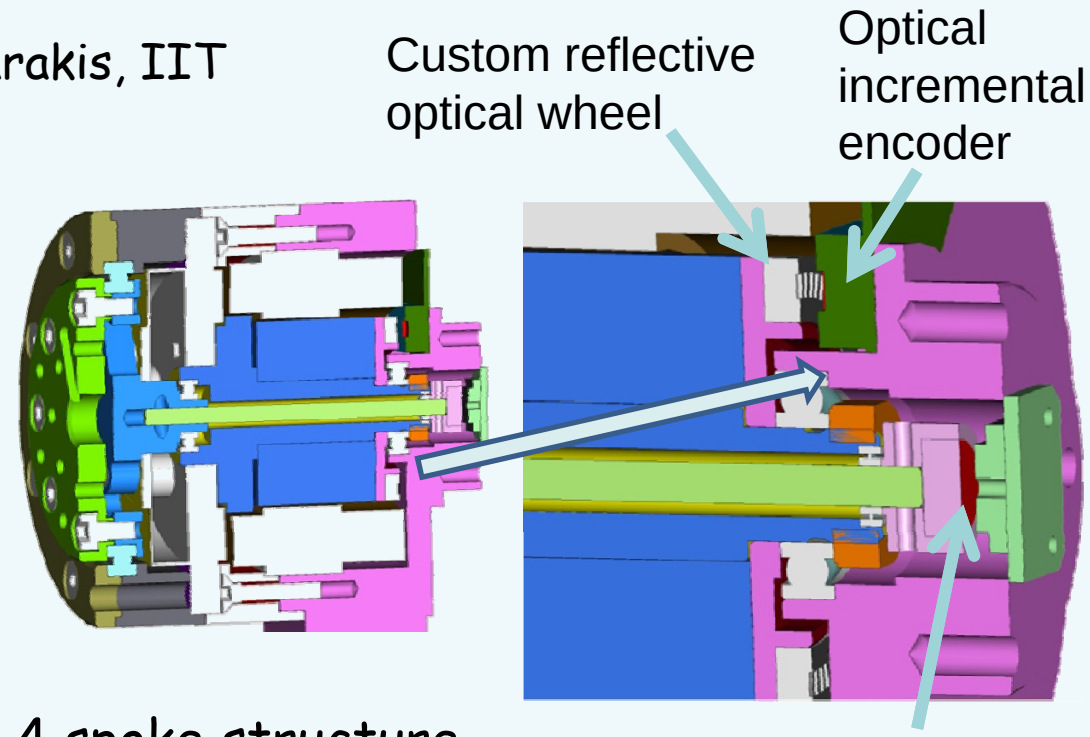
- Capacitive pressure sensor with 12 sensitive zones
- 14.5 mm long and 13 mm wide, sized for iCub
- Embedded electronics: twelve 16 bit measurements of capacitance
  - either all 12 taxels independently at 50 Hz or an average of the 12 taxels at about 500 Hz





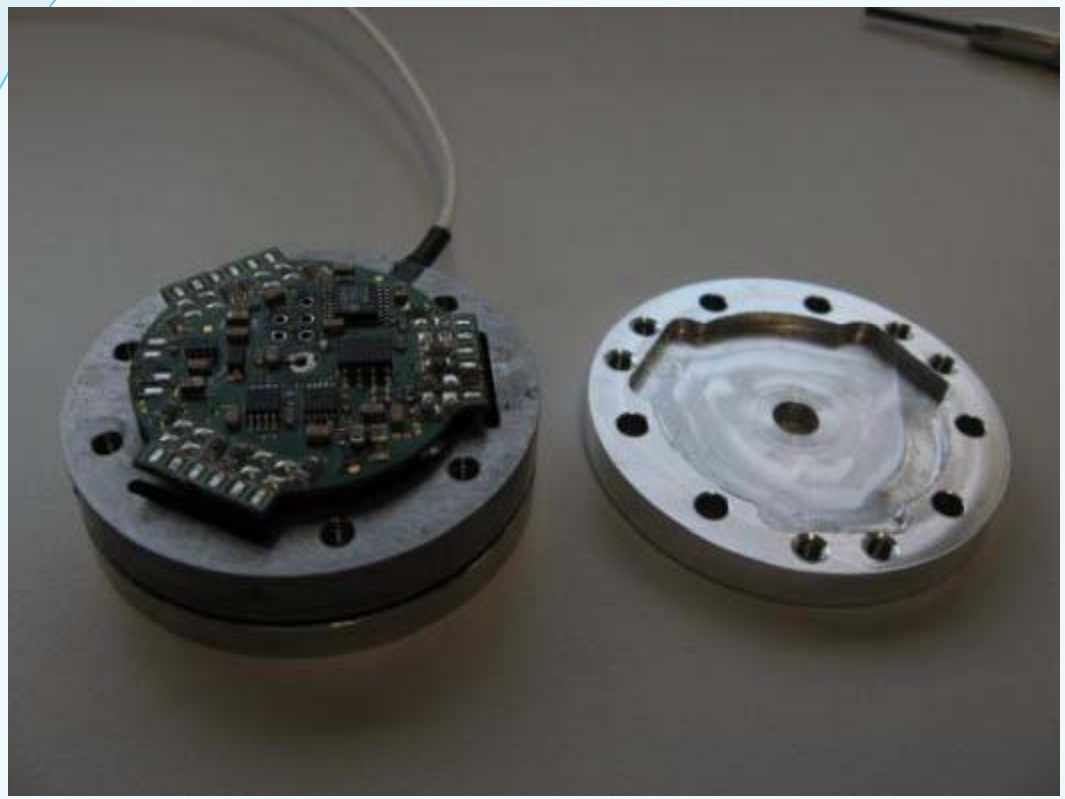
Joint-level torque sensing

Design: Nikos Tsagarakis, IIT



- Torque sensing - 4 spoke structure
- Position sensing – absolute (12bit) + incremental (19bit)
- Respect the original motor size
- Allow active compliance regulation

6-axial force/torque sensor



- Semiconductor strain gauges
- On board signal conditioning, sampling, and calibration
- Digital output: CAN bus

By Nikos Tsagarakis and Darwin Caldwell
Electronics: Claudio Lorini

RobotCub

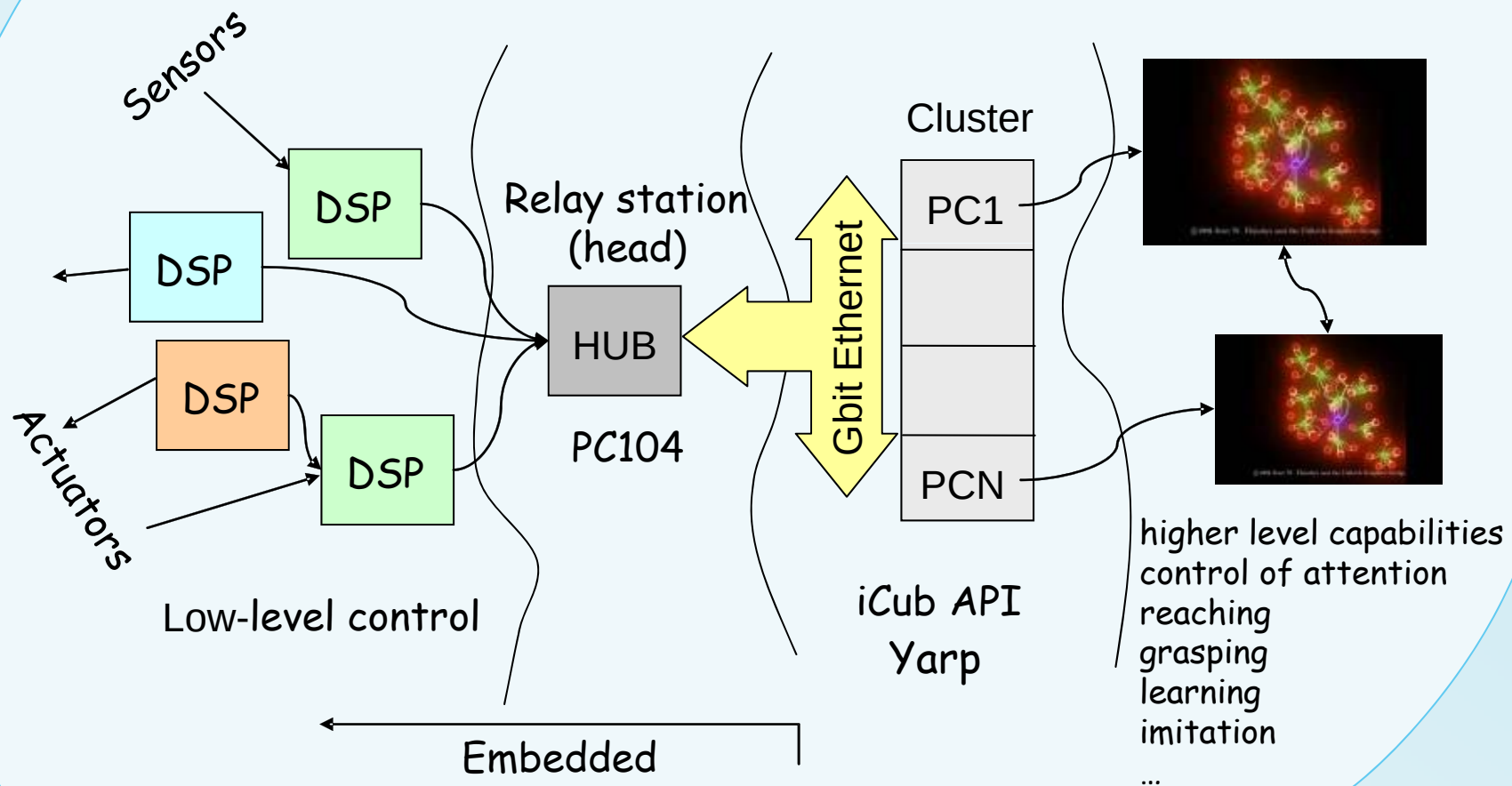
The Software

- Goals:
  - Foster collaboration in "space" and "time"…
  - … since we're a large Consortium and we don't want to re-invent the wheel too often
  - Manage the complexity of the hardware…
  - … since humanoid robots are complicated

  - Consequence: improved on existing Open Source libraries supporting a major overhaul of YARP to the iCub

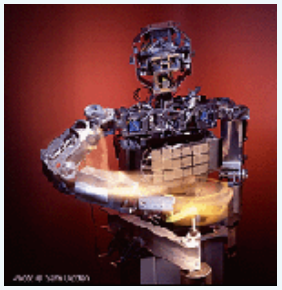
Cogsys
Cognitive Systems

Let's start from the hardware



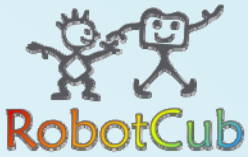
RobotCub

YARP



- YARP is an open-source middleware for humanoid robotics
- History
  - An MIT / Univ. of Genoa collaboration
  - Born on Kismet, grew on COG
  - With a major overhaul, now used by RobotCub consortium
  - Exists as an independent open source project
  - C++ source code
- In short: it's the plumbing

Cogsys
ve Systems

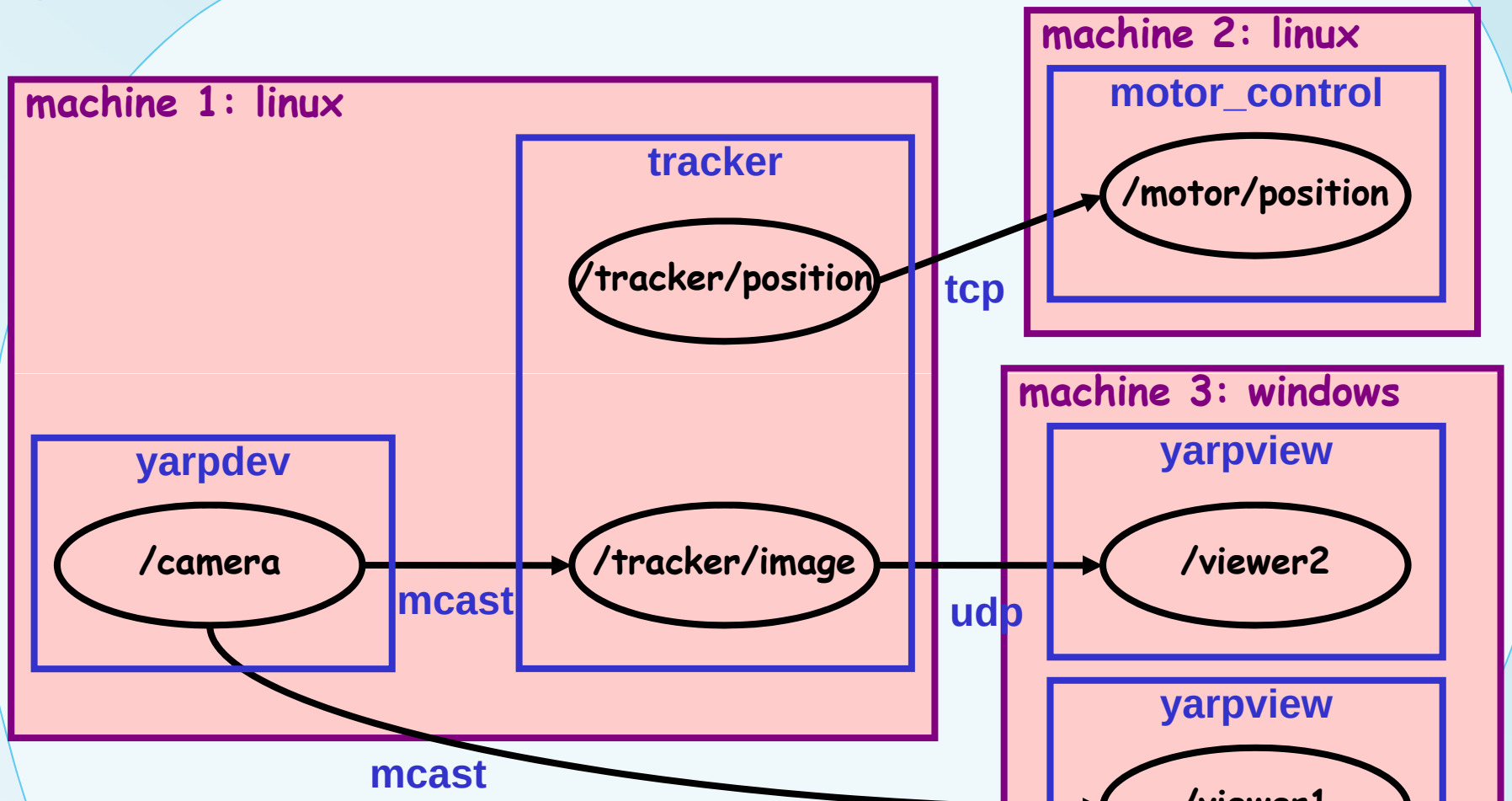


- It factors out:
  - the data flow: inter-process communication
    - it is often useful to keep algorithms away from the plumbing
  - the hardware: device drivers model
    - it is useful to avoid references to the hardware in the source code
- …while being portable:
  - across OS and development tools
  - across languages
    - libs in C++, bindings for many other languages

YARP Ports

- We follow the **Observer** design pattern.
- Special "Port" objects deliver data to:
  - Any number of observers (other "Port"s) ...
  - ... in any number of processes ...
  - ... distributed across any number of computers/OSes ...
  - using any of several underlying communication protocols with different technical advantages, streaming or RPC
- This is called the YARP Network

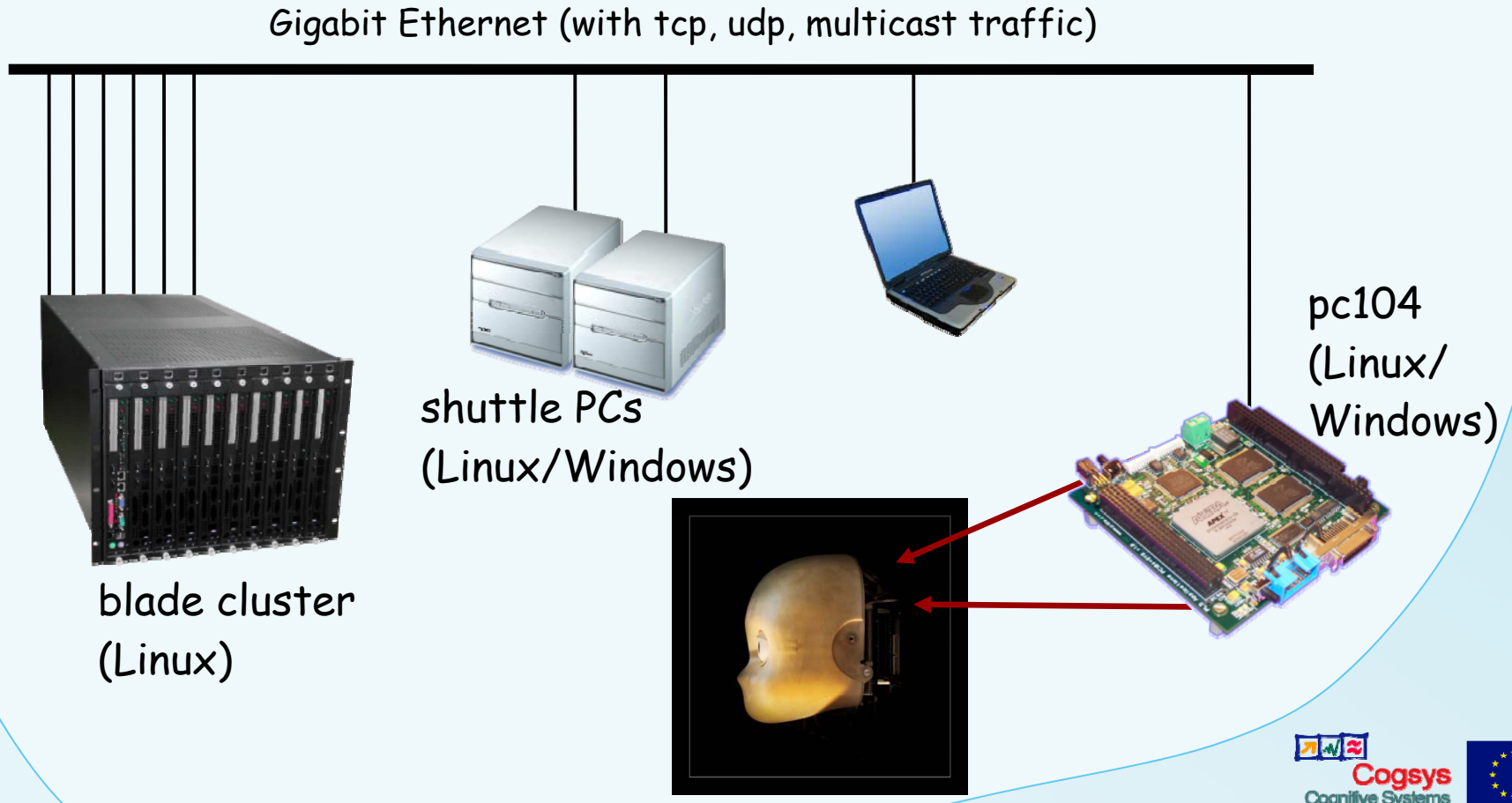
Typical YARP Network



- Connections can use different protocols
- Ports belong to processes
- Processes can be on different machines/OS

Physical Network

Example: iCub



RobotCub

Why is all this useful?

- We've separated out most of the plumbing
- We get to change it dynamically (handy)
- More importantly, we have better modularity
  - Programs can be moved around as load and OS/device/library dependencies dictate
  - Fundamental protocol for communication can be changed without affecting programs
  - Better chance that your code can be used by others (even just within your group)

Cogsys
Cognitive Systems

YARP Devices

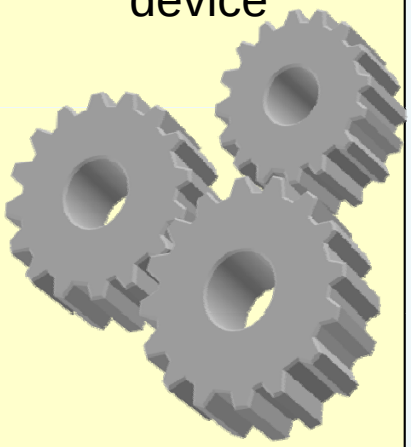
- Basic idea: if you view your devices **through well thought out interfaces**, the impact of device change can be minimized.
- There are three separate concerns related to devices in YARP:
  - Defining interfaces for **device families**
  - Implementing **specific drivers** for particular devices
  - Implementing **network wrappers** for interfaces

YARP Devices

- New devices come out all the time – needs to be easy to connect them to existing code
- YARP needs a minimal "wrapper" class to match vendor-supplied library with relevant interfaces that capture common capabilities
- YARP encourages separating configuration from source code
- Devices and communications remain distinct concerns

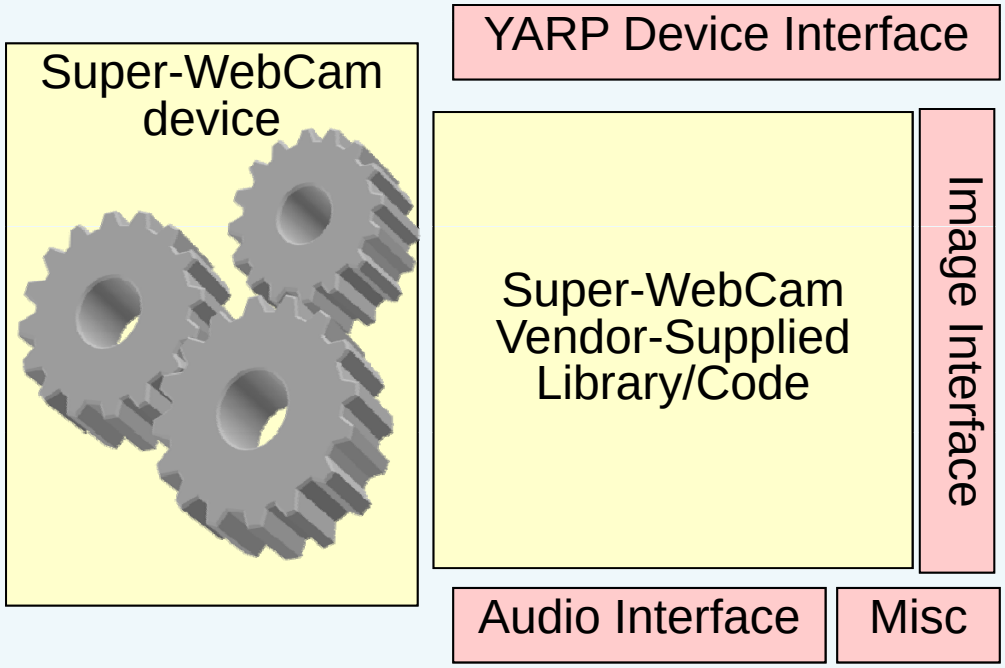
Example: New WebCam

Super-WebCam device

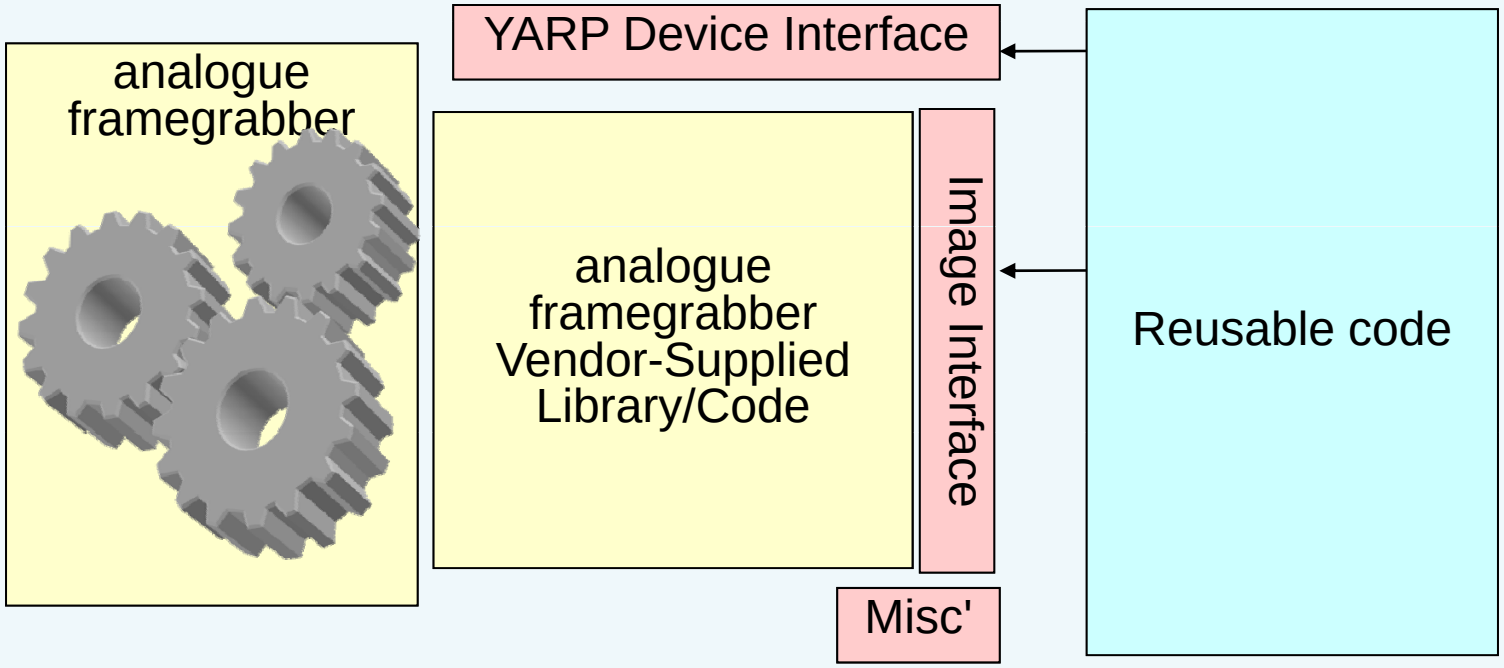


Super-WebCam Vendor-Supplied Library/Code

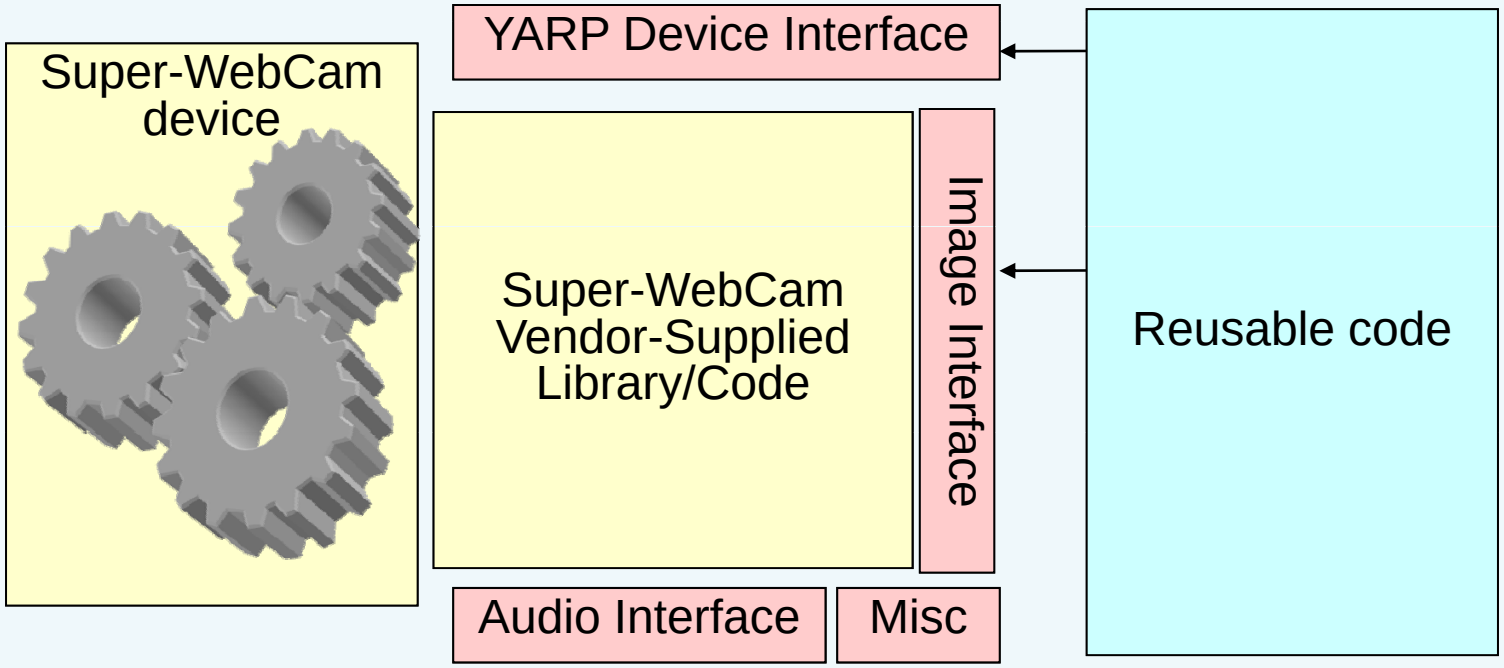
Example: New WebCam



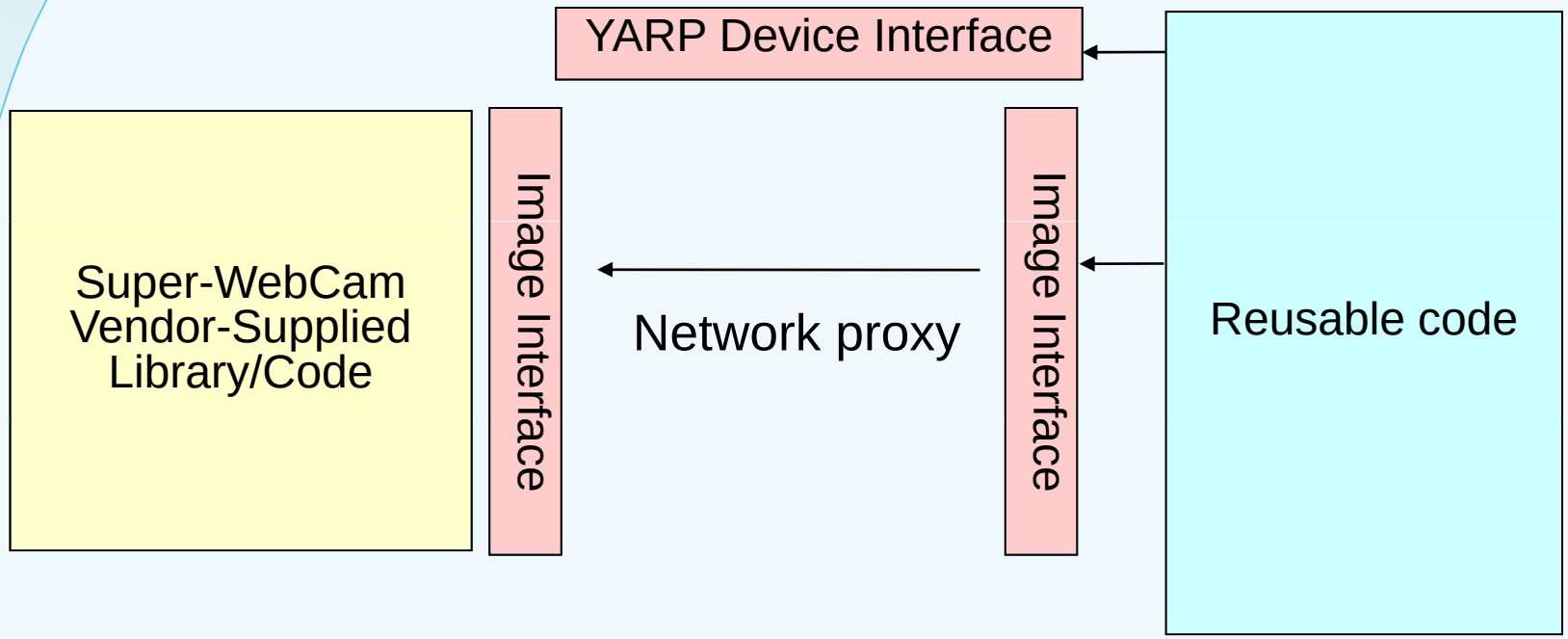
Example: Old Framegrabber



Example: New WebCam



Example: Networking



RobotCub

Why is this useful?

- Allows collaboration between groups whose robots have different devices
- Makes device changes less painful
- Devices and communications are orthogonal features
  - Can switch from remote use of device to local use and vice versa without pain
  - Local use can be very efficient, just an extra virtual method call

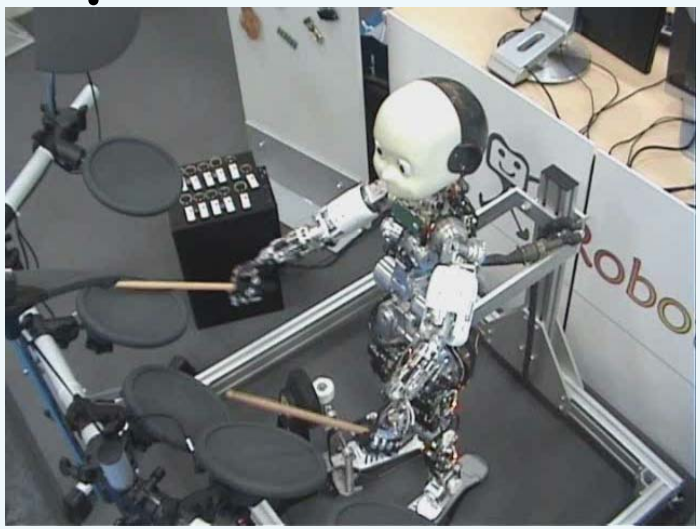
Final thoughts

Publication of software

- The literature of a research community both expresses its ideas, and aids in their evolution
  - Published ideas are read, evaluated, and built upon
  - Useful advances get published
- Publication of hardware and software can speed progress
  - Facilitates evaluating and comparing approaches
  - Brings new research topics into reach

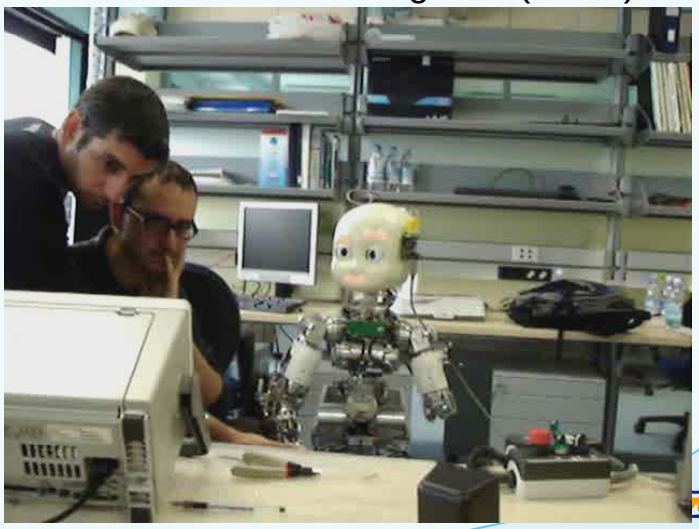
Examples…

With Peter Ford-Dominey (INSERM, Lyon)



With Auke Ijspeert, Ludovic Righetti, Sarah Degallier (EPFL)

With a lot of students
@ RobotCub summer school 2008



With VisLab (IST Lisbon)

Cogsys
Cognitive Systems