The iCub repository
• Material from:
  http://eris.liralab.it/wiki/Better_Repository
Repository Organization

- **admin**: contains scripts for maintainance, of no interest for most developers
- **tutorials**: some tutorials on the software
- **pc104**: pc104 startup scripts
- **license**: keep information about licensing
- **app**: stores applications scripts and configuration files

Code is now contained in two different directories **main** and **contrib**.

- **contrib** stores modules that can be compiled individually. This is a good place to upload new modules that are still under test. No automatic tests are executed on these directories.
- **main** this is the directory that contains the main build. This structure contains code that respects iCub standards (dependencies, supported operating systems and compilers, etc.). To avoid causing troubles to other users place here only modules you are reasonably confident will compile correctly. To help you check this automatic tests run periodically on this build.
main

- *main/conf* contains cmake helper code, including Find* scripts
- *main/src* contains source code.
  - *examples*,
  - *libraries*,
  - *simulators*
  - *tools*
  - *modules*
  - *core*
Compiling the main build

• Dependencies: and environment variables
  – Section 6.1: prepare your system
• Compiling and getting YARP
• Getting the iCub software
  – compile the iCub software
  – remember the difference between ICUB_ROOT and ICUB_DIR
CMake Options

• Optional modules:
  – ENABLE_icubmod_*: optional devices mostly for controlling the hardware (pc104)

• CMAKE_INSTALL_PREFIX

• CMAKE_BUILD_TYPE ="Release“

• Dependencies:
  – ICUB_HAS_<package>: read only
  – ICUB_USE_<package>: this variable can be enables or disabled if we want to use a certain package or not
List of dependencies

- [http://eris.liralab.it/wiki/List_of_iCub_dependencies_as_package_names](http://eris.liralab.it/wiki/List_of_iCub_dependencies_as_package_names)

- Inside cmake:

  `<package>_INCLUDE_DIRS` include directories
  `<package>_LIBRARIES` libraries
  `<package>_FOUND` true/false (redundant with ICUB_HAS_<package>)
Check out our dashboard

• automatic tests
• iCub and YARP:
  – [http://dashboard.icub.org](http://dashboard.icub.org)
• Continuous and Nightly builds
  – YARP: compilation and regression tests
  – iCub: only compilation
Using the software

• The iCub package provides
  – modules
  – libraries
  – cmake files to find packages
Using the software: modules

• Modules are just executables that end up in $ICUB_DIR/bin, it is enough to add this directory to your path

• check the documentation:
Using the software: libraries

- Libraries are compiled within the build
- They also end up in $ICUB_DIR/lib
- But you need certain instructions for using them:
  - library to link
  - header files to add to the compiler
  - liker flags
Using the software: libraries

find_package(ICUB)

It defines:
ICUB_INCLUDE_DIRS: a list of directories that contain all include files
ICUB_LINK_FLAGS: linker flags (if any)
ICUB_LIBRARIES: all libraries exported in the build
ICUB_MODULE_PATH: cmake scripts (Find*)

individual libraries are also available for linking:
ctrlLib
iKin
iDyn
iCubVis
...
Find scripts

• scripts available in In YARP_DIR/conf (use YARP_MODULE_PATH)
  – FindAtlas
  – FindGtkPlus
  – FindGThread
  – FindACE
  – FindBoost
  – FindOpenCV
  – FindPortAudio

• scripts available in ICUB_DIR/conf (use ICUB_MODULE_PATH)
  – FindGtkMM
  – FindODE
  – FindIPP
  – FindIPOPT
  – FindESDCANAPI FindCFW2CANAPI FinPLXCANAPI
How are libraries organized

• We define C++ namespaces
  – iCub
  – iCub::contrib
  – iCub::mylib

• How header files are organized like namespaces:
  #include <iCub/mylib/header-file.h>
Full template:

• Modules:
  http://eris.liralab.it/wiki/Simple_template_for_modules_in_contrib

• Libraries:
  http://eris.liralab.it/wiki/Simple_template_for_libraries_in_contrib
See the tutorials:

• [http://eris.liralab.it/wiki/Manual](http://eris.liralab.it/wiki/Manual)  -->  Section 9.9

• or, directly: