	LEFT LEG	RIGHT LEG
- UPLOADER	J_0 1,15	J_0 1,5
- RESTART ETH	J_1 -1	J_1 -3,5
- DISCOVER		
- CHANGE IP		
- SPECIAL LOAD		
- RESTART		
- DISCOVER		
- UPDATE		
- JUMP TO UPDAER		
- SET DEF BOOT EUPD		
- NORM APPL		
-		

5

4

0

Login: icub / icubAdmin	icub-head / icub

Calibration if fine calibration + quite fine
(e.g. ankles turned, feet not flat)
put legjoints idle 0,25s. and back?
⇒ twoFeetStandingIdleAndCalib.sh (hold the robot. It will adjust position)

(M) red light blinking — lights
long press button — until green. wait motors stop blinking
shut down yarprobotinterface.
yarp clean.
start again.

---

see forces measured by robot with iCubGui.
Move base in iCubGui
yarp rpc /iCubGui/base:i
>> 0.0 0.0 0.0 0.0 0.0 1000.0 } This is to visualize vertical forces at the feet (when on ground)

Sometimes if robot falls / goes into hardware fault, some joints will reach their limits. When restarting yarprobotinterface, these joints will not fail being started / put into home position / controlled
+ yarpmotorgui: put joints in idle (the faulty joints)
move them back into their range
run joints
home the robot.

- UPLOADER
- RESTARTI ETH
- DISCOVER
- CHANGE IP
- SPECIAL LOAD
- RESTART
- DISCOVER
- UPDATE
- JUMP TO UPDAER
- SET DEF BOOT EUPD
- NORM APPL
-

	LEFT LEG	RIGHT LEG
J_0	1,15	1,5
J_1	-1	-3,5

5

4

0

Nancy

① Check if they updated the firmware
↳ If not, it has to be done.
Marco Accame will provide assistance (in vacation Mon-Tue)
Valentina Gaggero " " "

② Update Firmware

→ github.com/robotology/QA/issues/240 : How to update your robot to use 1.8.0
guide to migrate to latest icub release (applies to CAN and ETH robots)

②.1 On server computer : pull repositories {1- yarp /master
and compile {2- icub-main /master

On icub-head/pc104 : 1- pull yarp, icub-main, icub-firmware-shared,
" " -build, robots-configuration
2- compile yarp
3- compile icub-firmware shared : cd build-pc104 ...
4- " icub-main : " " " ...
5- Install robot-configuration files → instructions at ...ology/robots-config
⑥- Firmware update of ETH/CAN boards → instructions
icub-firmware-build/FirmwareUpdater.readme.quick.txt

Firmware update uses Firmware updater (from icub-main /master)

6.a Power on motors
b Firmware Updater GUI
Find firmware update.ini (should be installed with robots-configuration)
Select ETH device. Click discover to find available boards
→ A list of boards appear
c Select an EMS board. Can view (on side) board properties (running/idle etc)
Click ☑ to select all boards of the same type
d. Click Force ETH maintenance
e. Select all boards in maintenance.
Click Upload application → load the ems.hex file (in ~~robot-configuration~~ icub-firmware-build ?)
discover → f. Select board → check in Board Properties panel if correct Firmware version
g. Select all boards of type ☑
Click Force ETH application (send them in application state = re-start application
equivalent of power off. power on. motors)
h. Done. Quit.

③ Use FirmwareUpdater for debugging a board which is not responding with yarprobotinterface

a. Search board. See if it finds it
b. Ping board → if it replies → is yarprobotinterface crashed?
(not found) does not reply → is a cable loose? not well connected?
c. Click restart board (in magic state w/ more buttons) : FirmwareUpdater --admin
check console → debug info on silence time, timeout, etc.

ISSUE TO OPEN
ob [Note: robot-config does NOT install firmwareupdate.ini (check cmake list file)

4) UPDATE LOADERS + UPDATERS *Because this process changes IP address, each board is done one at a time *.

FirmwareUpdater -- main

Select ETH → Discover → upload eloader   /eth.../env†/ emsLoader.hex

It> b. Select ETH. Discover, appears ems 10.0.1.99 → select it
→ Force ETH maintainance
select again.
Change IP address : 10.0.1.1

Select ETH (unknown) → enlarge the window to see info
→ upload application   /ETH/ems/bin/env† / ems app... prog. update.
emsAppProgUpdater ("the special application")

Select ETH device   Restart ETH Boards
wait ~5 sec.

ETH → Discover → see eApp...
select Upload eUpdater   /.../emsUpdater

Discover.
Select ems 10.0.1.1 → check properties
updater on date (e.g. 2016.)
See bootstrap options
Click
Jump to eUpdater to verify that it works : it will change to eUpdater

ETH → Discover → now 10.0.1.1 (or .2 or ...)
Select → Set Def Boot eUpdater and reprogram

Upload application /.../application/ems.hex

Select 10.0.1.1 Force ETH application (~Restart)

REDO IT ON 10.0.1.2 Force eth maintenance. Upload eloader env/emsLoader
restart ETH boards. Subito Discover (to keep the eUpdater)
select device → Discover → becomes 10.0.1.99
change IP address → 10.0.1.2 (or .3 or ...)
Then 10.0.1.3   upload application ../env†/ emsAppProgUpdater
only ETH (not CAN!):   Discover restart ETH boards. Wait 5 seconds (should be changed to if not restart.)
upload eUpdater ../env†/ems updater
Discover Jump to eUpdater
Set Def boot eUpdater
This can also be done only once at the end to save time (takes about ~10 secs.) → [Upload application ../application/ ems.exe
Force ETH application]

[Select all boards (.6 → end)]
[Force ETH maintainance
→ chose a board. upload eloader. Restart ETH boards. change IP address
" the same"   emsAppProgUpdate. Restart ETH board. wait.
upload eUpdater...
Jump to eUpdater, set Def boot...
"   (Change IP address back)
→ choose next board.

→ Select all boards. Force ETH maintenance. Upload appli. ems.hex
"   Force ETH application

Random recommendations:

→ If the robot is not responding, don't turn it off already.
(can still press Fault button, though)
let it run for a while (some minutes)
maybe a loose ETH cable is causing very slow communications,
and errors would show up after a long time.

→ Old iCubs CPU in head: there was no hard disk. It was remotely mounted using NFS (from server)
this means the code was shared with the server computer
⇒ only the server computer needed to be updated

→ yarprobotinterface is a bunch of threads of equal priority.
But transmission tasks need to be run with high priority.
The new version in iCub-main/master fixes this
and offers better debug information w/ respect to it.

→ icub-main/master now handles R-T mode with high priority threads
for smoother transmission

→ iCub Nancy has "version 1" arms. Joints can be calibrated through similar files
as with green/purple, in robot-configuration
Calibration is done through matrix e 23 (e.g.) ⇒ motor encoder → joint
by setting the first 4 cols. = 1/4 ... and then ask Julien/Valentina.
(or changing delta if lazy)

N ISSUE
OCUMENTATION

(x) YARP → strain embobjinertial0 embobjMAIS... ON
inertial sensors: OFF

do w/
RAIN

.1
.3
.6/7
.8/9

FOC Boards don't keep update.

Boards with skin
, 2, 4 → Force ETH maintenance.
Select 4. Discover.
Select all skins (1 achile board)
Upload application → skin.hex
(x) if not all at same version! Turn off motors. Turn on. Wait.
ETH → Discover. When doing software updates in //,
casini may happen.
Better to do it one by one: upload application → skin.hex
(try selecting boards with non neighboring IP
BLACK MAGIC e.g.: 12 and .8, not .1 and .2)
for doing in //

} Repeat with 2, 11, (10)

.2 Left arm
.4 right arm
.11 right leg
.10 Left leg

already
st version
date

Move source files for PC104 (from icubsrv to icub01 computer)
(i.e share code directory btw PC104 / icub01)

(cmake cache ← clean. for paths which have changed)

① icub server laptop installation instructions

→ host file: in /etc/hosts → add line: 10.0.0.2 pc104

→ NFS server: icub01 hosts 2 directories. Exports them using NFS to be mounted by pc104 < icub01
/exports/code : (src YARP, iCub...)
exports/local_yarp : yarp config files - shared

(!) TYPO on wiki →
* Install NFS on icub01
* Create directories + set permissions
* configure nfs-kernel-server : in /etc/exports add lines + restart NFS

→ Configure icub01 to mount remote NFS shares : edit /etc/fstab
create mountpoints

→ Create a symbolic link to code export path : sudo ln -s /exports/code /mountpoint
→ Clone icub software into /exports/code : yarp, icub-main, icub-firmware - ← /shared/build)
→ Create a symbolic link to local yarp export path : mkdir -p /home/icub/.local/
sudo ln -s /exports/local_yarp /home/ " " " share yarp

(→ install ssh keys for PC104)
→ IP forwarding / NAT / IP / DNS / clock sync (laptop is clock of PC104 connect if / internet)

② pc104 installation instructions : should be done already.

③ Installation of software

for controllers
ini files for home position need to be copied from icubGenova04

F/T sensors through FirmwareUpdater

- Start up : FirmwareUpdater -- admin

ETH → Discover
- Select the desired board (10.0.1.1 → left arm joints 0 → 3
  10.0.1.3 → right " " " "
  10.0.1.6 → left leg joints 0 → 3
  10.0.1.7 → " " " 4 → 5
  10.0.1.8 → right leg joints 0 → 3
  10.0.1.9 → " " " 4 → 5

- Click Force ETH maintenance
- Select the board. Discover.
- Select the strain device.
- Calibrate STRAIN to launch the strain calibration GUI

[What to check:
1- Offsets should have a value between 600 and 900

2- Calibration matrix should have small values (below FFXX) (in hex)

Button: Automatic Offset Adjustment : does not seem to work.
Changes are NOT saved until closing the window (prompt asks for confirm)

~~10.0.1.1 → Ch:0 Offset : 904 → 821~~ still getting error saturation

Playing with Offsets will influence the current values of the channels
=> one can adjust the offsets in order to minimize the
(For example above 10 000 is too much)
sliding the scroll until achieving a satisfyingly low value is obtained for that channel.

Close. Accept to keep the changes. Close.

Force ETH Application. Close FirmwareUpdater.

Turn off motors. Turn motors back on.

left leg 0-3 : channels 0, 4

yarpscope --remote /icub/left_arm/analog:0

NOTES for Nancy people

* COPY ALL .LOCAL files to git robot-config° and commit.
* Will need to update firmware for board 10.0.1.10 left leg when new skin comes
* Open issue: to be able to type joint position in yarpmotorgui
* Provide document° on how to calibrate IMU: with robot on pole in home position (i.e. all 0)
  - → matlab2015b
  - → /codyco-superbuild/main/WBIToolboxControllers/utilities
    launch calibrateIMU.mdl
    run
    look at the IMU scope → [should be ~0 ? yellow ; ~0 ? blue ; ~9.8 red
  - → yarpmotorgui ⇒ move head joints to bring signals to 0
    keep note of the joint offsets. ⊛ joint positions
  - →... software/yarp/icubcontrib/robots/icubNancy01/calibrators
    open head-calib.xml
    add the offsets to the corresponding joints in "calibrationDelta"
    save. close.
  - → Restart the robot. check again with calibrateIMU.mdl
    Repeat previous steps as needed. (the IMU values should be closer to 0 but they may not yet be perfect)
  - ⇒ The head is CAN. not ETH. The above file will not be loaded properly.
    workaround: put the head in initial position required for the IMU, obtained at ⊛ as a workaround for now.

[on to run Yoga demo]
... software/yarp/icubcontrib/robots/icubNancy01/calibrators/ (same file)
"startupPosition" changed to ⊛ in head-calib.xml

1 [* HOME POSITION YOGA++: head home position also needs to be adjusted
yarpmotorgui -- from homePoseBalancing.ini
4 menu bar global joints commands → custom positions
→ move all parts to YogaAPP

workaround homePoseBalancing.ini is in .../codyco-superbuild/build/install/share/codyco/robots/icubNancy01
↳ it has been modified according to our measurements
⇒ $robot/head-position for joint 0 was changed! for homePoseYogaAPP
original value: 0
modified value: -1.5

2 [adjust F/T sensors measurements (adjust offsets) with
yarp rpc /wholeBodyDynamics/rpc
» calib all 300 (300 is a time delay in ms)

3 [makumba to release eventual stress in the legs:
» twoFeetStandingAndCalib.sh
(lightly hold the robot by the arms while running the script; it will move a bit)

matlab → .../codyco-superbuild/main/WBIToolboxControllers/controllers/BalancingTorque

- torqueBalancingR20155B6.mdl ← Simulation
- README ← documentation - outdated
- initTorqueBalancing.m ← contains a few more lines of doc.
  - → check robot name
  - → options to turn on/off scopes
- app/robots/iCubNancy01/gains.m ← controller gains, torque saturation value...
  - " /initStateMachine.m ← gains for yoga state machine *overwrites some variables from gains.m*
  - /initRefGen.m ← more parameters

+ changed hardware joint limits in PID file.

changes in robot-configurations motor control (PID gains from yarpmotorgui)

		kp	kbemf	ktau
torso →	yaw	450	0.0008	200
	roll	400	0.0015	200
	pitch	400	0.0015	200
l_arm →	shoulder + elbow joints		kbemf = 0	
r_arm →	"	"	"	"
l_leg →	knee ; switch signs kp, kz and kz = -100			
r_leg →	r_ankle_pitch : kp = -300		(was -200)	
	r_ankle_roll : kp = 0		(was -200)	⇒ PUT BACK TO -200*

↓ new change

BUG sm.com.threshold TWICE in initStateMachine.m