iCub Courses

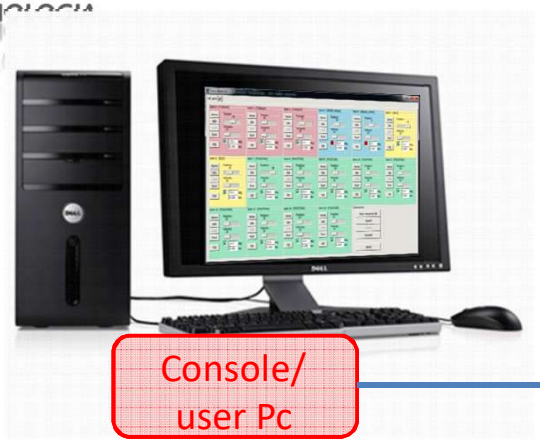
An introduction to iCub hardware and firmware

For CAN-based iCubs, with some concepts about Ethernet-based iCubs.

< marco.randazzo@iit.it >

ISTITUTO ITALIANO
DI TECNOLOGIA

Story of a command message

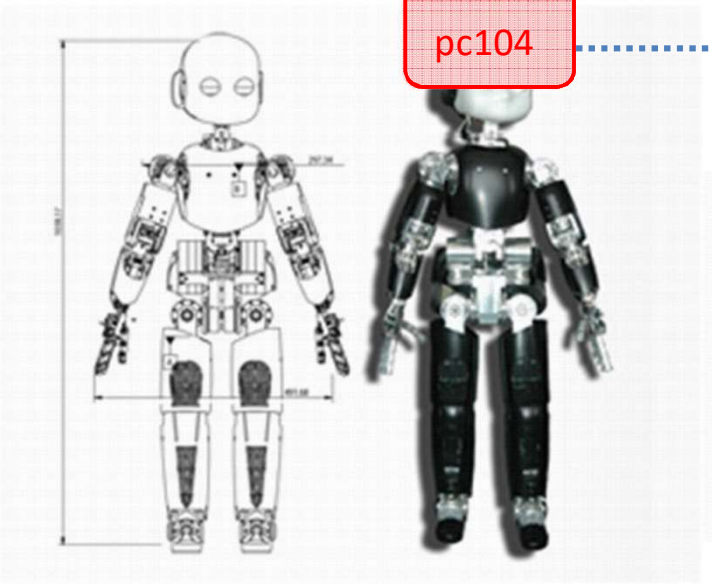


Console/
user Pc

Gb Ethernet

Blades/
server

pc104



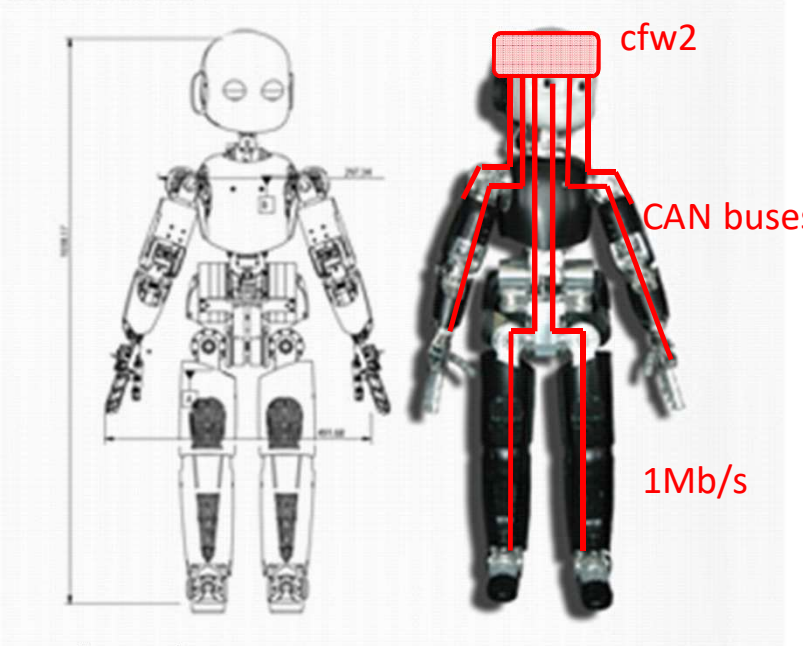
???

iCub
motors/sensors

ISTITUTO ITALIANO
DI TECNOLOGIA

iCub Hardware (CAN based)

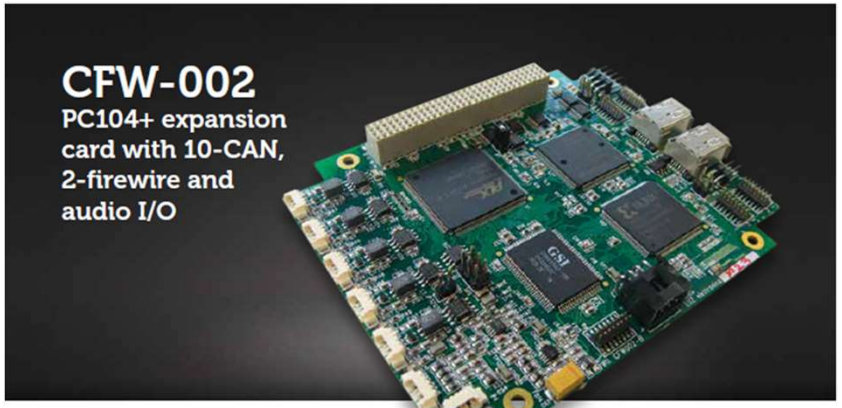
- Black iCub -



iCub CAN buses:

0 Head, Torso
1 Left arm
2 Right arm
3 Left leg
4 Right Leg
5 Left Hand
6 Right Hand
7-8-9 Skin (left arm, right arm, torso)

Each CAN bus can host up to 14 devices (motors boards/sensors). Address 0 is reserved to the cfw2.



The CFW-002 board is PC104+ standard expansion card directly connected to the PC104+ bus. The board has 10 CAN Bus 2.0B links, 2 IEEE 1394 (Firewire) high speed ports, one stereo microphone preamplifier and a single D-Class speaker output.
The CFW-002 is designed to operate with the PC104+ CPU (Linux based). The device drivers are available for Linux (2.6.x kernal) which enable high bandwidth, low latency data stream towards and from the PC104+ card.

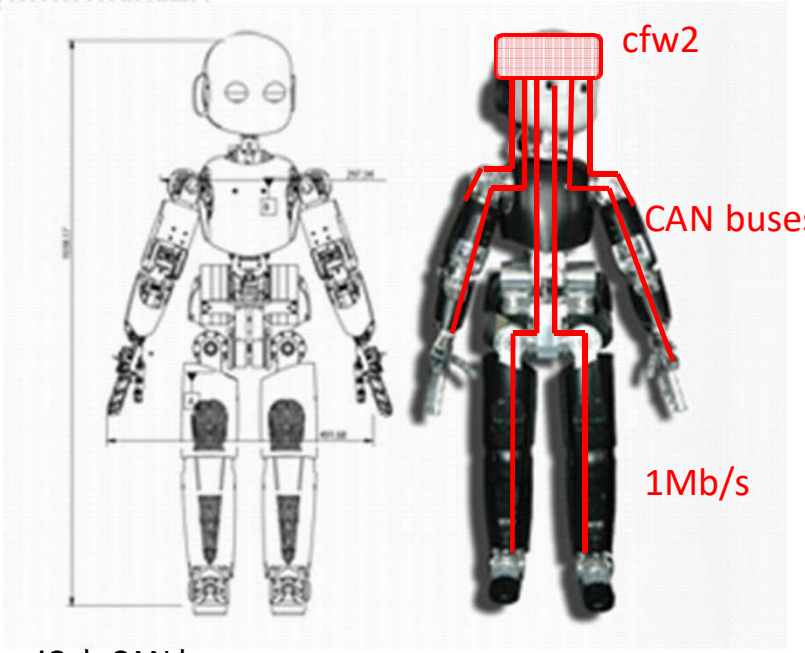
Specifications

Power supply	3.3V,5V and 12V
Communication	10x ESD-protected CAN Bus ports, 1Mbps. 2x Firewire 400mbps IEEE-1394 compliant ports 33MHz PCI
Audio PA maximum power	920 mWatt over 8 ohm
Audio LNA SNR	74dB
Audio PGA regulation	7-step software controlled
Onboard RAM for CAN packets FIFOs	128Kbits
Microcontrollers	2X Infineon XC2287/XC2287M, up to 80 Mhz clock
OS support	Linux 2.6.X custom device driver
PC side connection	Standard PC104+ 33Mhz PCI connector
Operating conditions	0 to 50°C, humidity <85% without condensation
Dimensions [LxWxH]	96x116x13
Weight	72 g

ISTITUTO ITALIANO
DI TECNOLOGIA

iCub Hardware (CAN based)

- Black iCub -

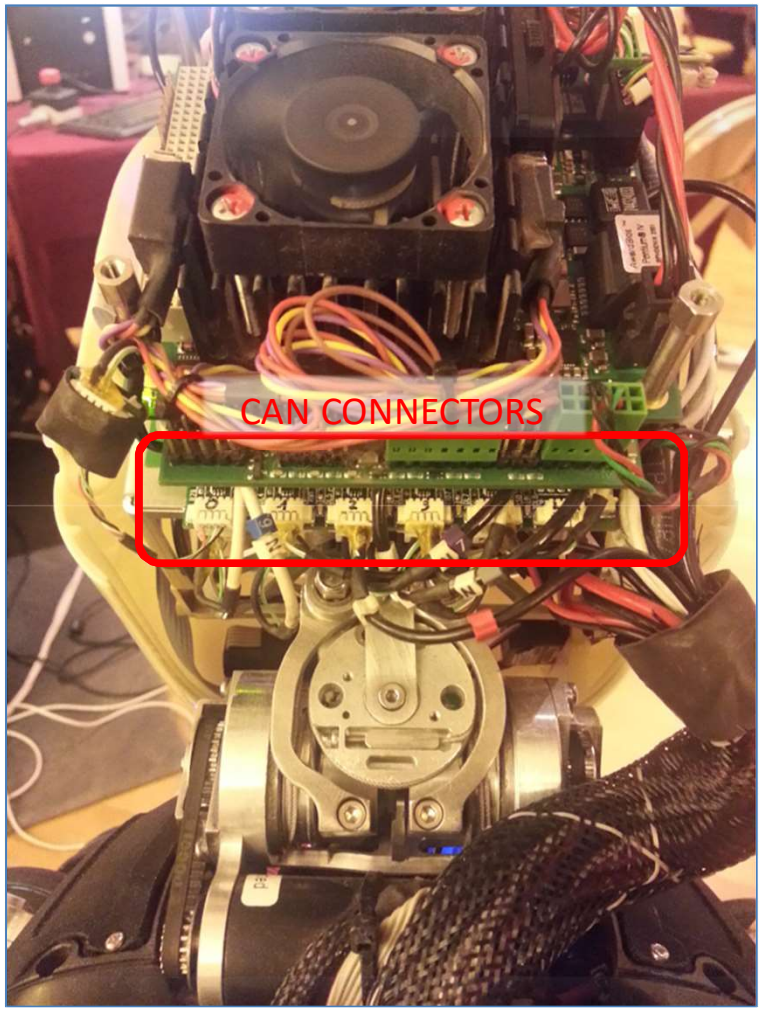


iCub CAN buses:

0 Head, Torso
1 Left arm
2 Right arm
3 Left leg
4 Right Leg
5 Left Hand
6 Right Hand
7-8-9 Skin (left arm, right arm, torso)

iCub is an assembly of 10 independent robots!

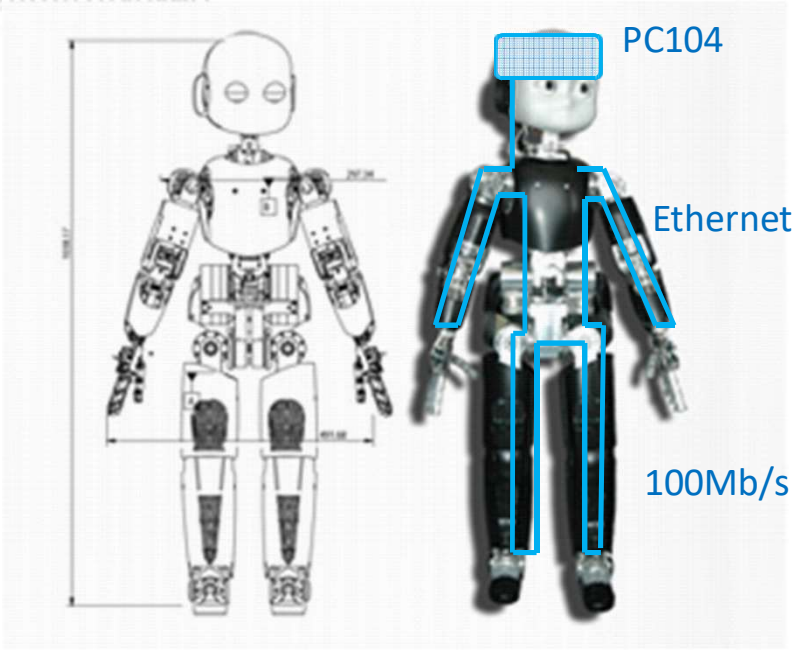
Each CAN bus can host up to 14 devices (motors boards/sensors). Address 0 is reserved to the cfw2.

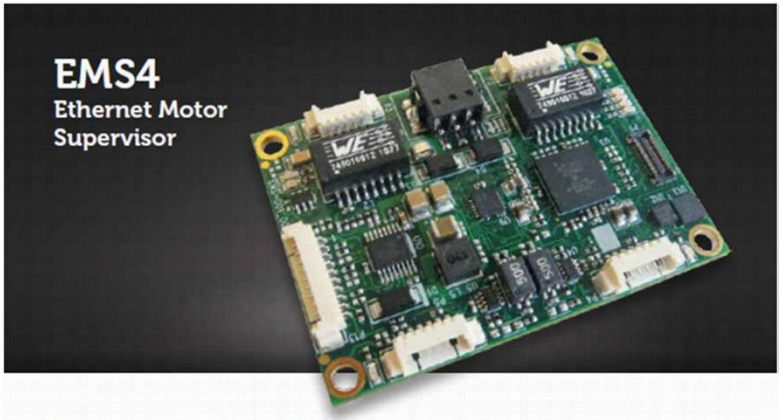


ISTITUTO ITALIANO
DI TECNOLOGIA

iCub Hardware (Ethernet based)

- Blue iCub -

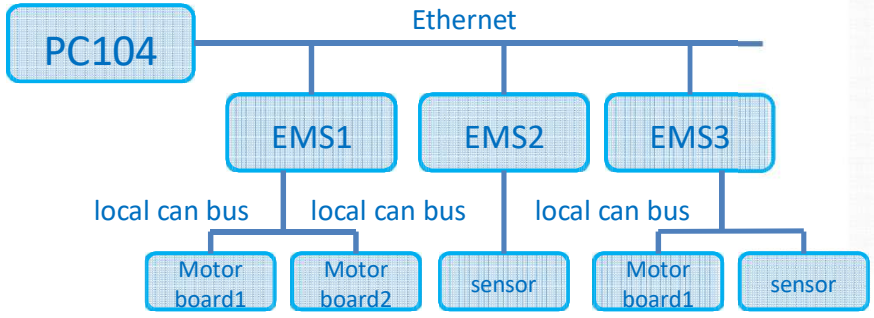


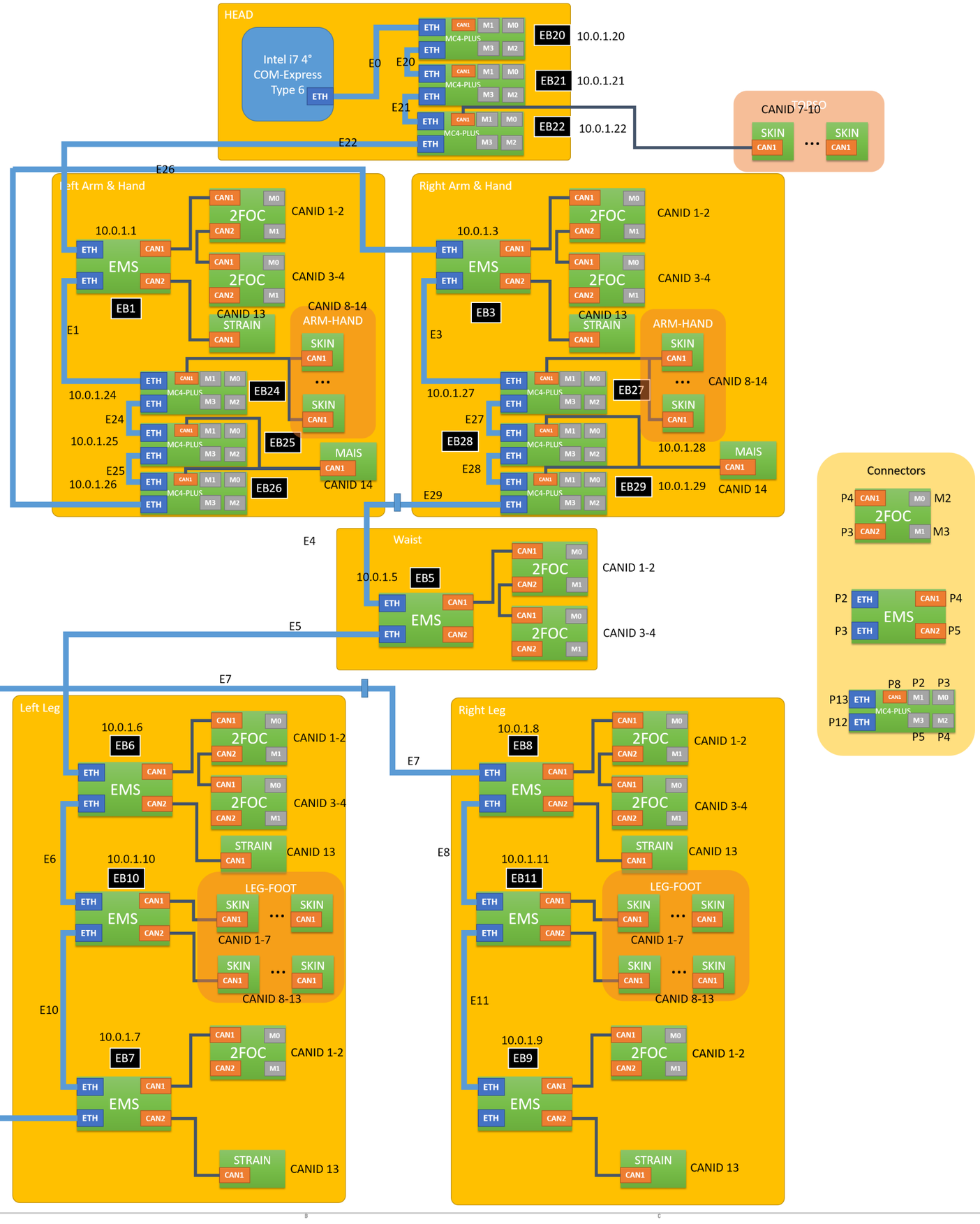


The EMS4 card is a 32-bit Arm Cortex-M4 embedded microcontroller based device designed for managing several communication channels in robotic applications. Its main function is to provide the bridges among two Ethernet 10/100 base-T high-speed link, two CAN-2.0B buses and six SPI-master buses. When connected to external motor-driver cards (i.e. 2FOC, BLL/BLP or MCP/MC4), the EMS4 embedded microcontroller provides enough computational power to directly manage up to four motor control-loops.

Specifications

Power supply	Operating: 9.5Vcc to 58Vcc
Microcontroller	STM32F407VGT6, ARM Cortex M4 168MHz clock
Memory	256KB Flash ROM, 64KB SRAM and 64KB non-volatile EEPROM
Communications	Ethernet 10/100Mbps MAC with IEEE1588 capability. On board managed switch with dual 10T/100TX PHY ports, with auto-MDI/ MDI-X protocol
Expansion buses	2 independent CAN 2.0B buses / 6 (3x2 multiplexed) SPI Master buses
Special functions	3 axes accelerometer: 16bit output, full scale 2g, 4g or 8g 3 axes gyroscope: 16bit output, full scales 250dps, 500dps or 2000dps
Expansion I/O	7 general purpose terminals configurable as digital I/O, incremental encoder inputs, PWM outputs, analog inputs (6 channels), and analog outputs (2 channel)
Generated power supply	5Vcc ±3%, max. 450mA (CAN and SPI buses) / 3.3Vcc ±3%, max 200mA (CAN and SPI buses)
Tools	Programming and debugging tools from ST, Kail, GNU, Raisonance, IAR and others. JTAG or USART download
Operating Conditions	0°C to 50°C, relative humidity <85% (operating)
Dimensions [LxWxH]	58x42x11 mm (2.28x1.65x0.43 in)
Weight	17g





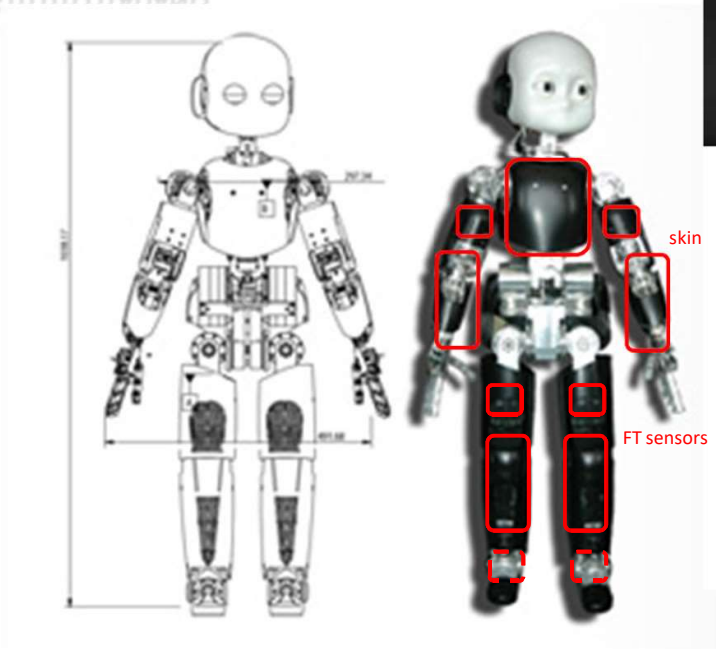
Revision History			
Revision	Date	Description	Short Description
3.3.0	26/10/2017		architecture
3.2.1	20/11/2016		architecture

IIT - Istituto Italiano di Tecnologia
Via Morego, 30
16163 - Genova

Project: iCub 2.5			
Drawn: marco maggiali	Created: marco maggiali	Approved:	
Document: 0_Architecture Diagram1			Revision: 3.3.0
Diagram Size: A1	Date: Thursday, October 26, 2017 3:28:34 PM		Sheet: 1 of 1

ISTITUTO ITALIANO
DI TECNOLOGIA

iCub Hardware (other CAN devices)



The **FTSens** is capable of measuring 3 forces and 3 torques (in a Cartesian reference system) using a CAN bus line to transmit data in digital form. The sensor is based on semiconductor strain-gauge technology; the mechanical assembly contains all the signal conditioning electronics and a microcontroller for communication.

Specifications

Power supply	5V±10%, current consumption max 100mA, provided from CAN Bus connector
Communication	CAN Bus 2.0B, 1Mbps
Channels	Six, 3 torques (Tx, Ty, Tz) and 3 forces (Fx, Fy, Fz)
Measure range	2000 N (Fx, Fy, Fz) 40 Nm (Tx, Ty) 30 Nm (Tz)
Resolution	0.25 N (Fx, Fy, Fz) 0.049 Nm (Tx, Ty) 0.037 Nm (Tz)
Output data	16 bit, 6 channels, up to 1K messages/sec
Microcontroller	dsPIC30F4013 16bit, 30MIPS, 48K Flash, 2K RAM, CAN, SPI
Alarms	CAN communication, memory, ADC and PGA
Digital filter	6 indipendent 5th order IIR
A/D Converter	16 bit, 250ksps
Gain settings	Fixed analog gain
Offset correction	digital offset correction
Utilities	In field reprogramming, device configuration, graphical data analysis
Operating conditions	0 to 50°C, humidity <85% without condensation
Dimensions [Φ,H]	45x18mm
Weight	122g

Large Area tactile sensor based on capacitive technology. Up to 16 triangles with 12 sensors each (192 taxels) interconnected on a flexible-pcb driven by a microcontroller board (MTB). The skin can be cut from the triangular panel and applied to any 3D surface.

Specifications

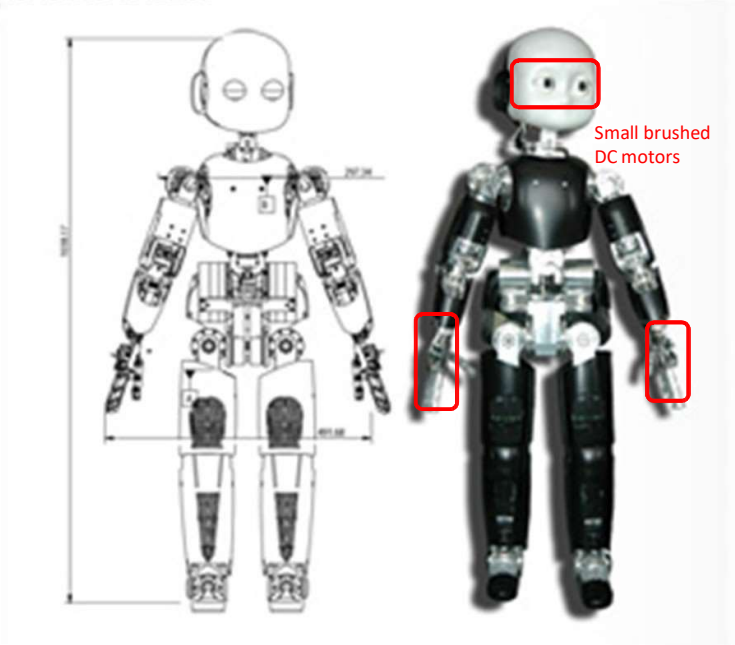
Power supply	5V±10%, current consumption max 70mA
Communication	CAN Bus 2.0B 1Mbps
Microcontroller	dsPIC30F4011 16bit, 40MIPS, 48K Flash, 2K RAM, CAN, SPI
Resolution	8 bit
Bandwidth	From 25Hz up to 250Hz
Utilities	In field reprogramming, drive configuration, graphical data analysis
Operating conditions	0 to 50°C, humidity <85% without condensation
Dimensions [LWH]	MTB 25.5x17.4x7mm / Triangle side of 30mm, etched on a 0.22 mm flexible PCB
Weight	MTB 2g / Triangle 0.3g

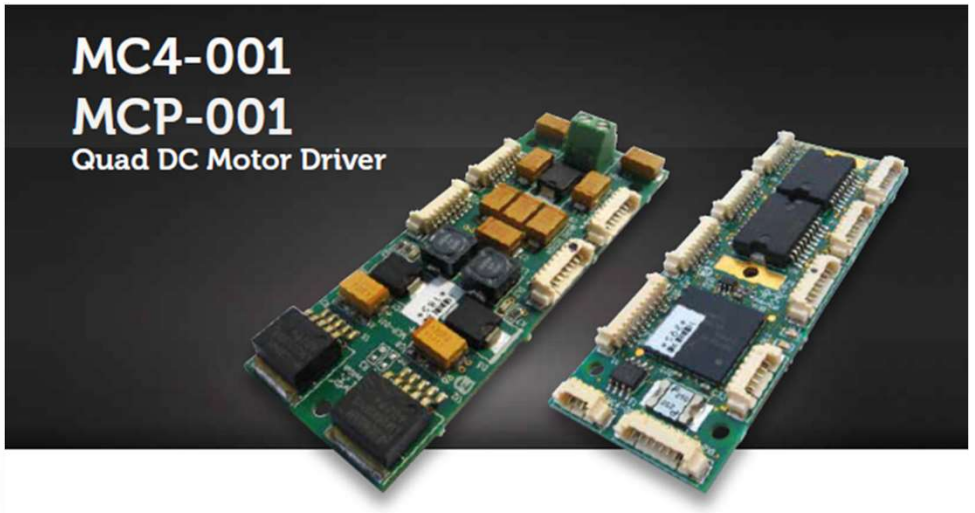
iCub CAN buses:

0 Head, Torso
1 Left arm
2 Right arm
3 Left leg
4 Right Leg
5 Left Hand
6 Right Hand
7-8-9 Skin (left arm, right arm, torso)

ISTITUTO ITALIANO
DI TECNOLOGIA

iCub Hardware (other CAN devices)





The **MC4** board is a small motor controller capable of driving up to four brushed DC motors, using CAN bus communication for commands and control data exchange. This is complemented by a small power supply board (MCP).

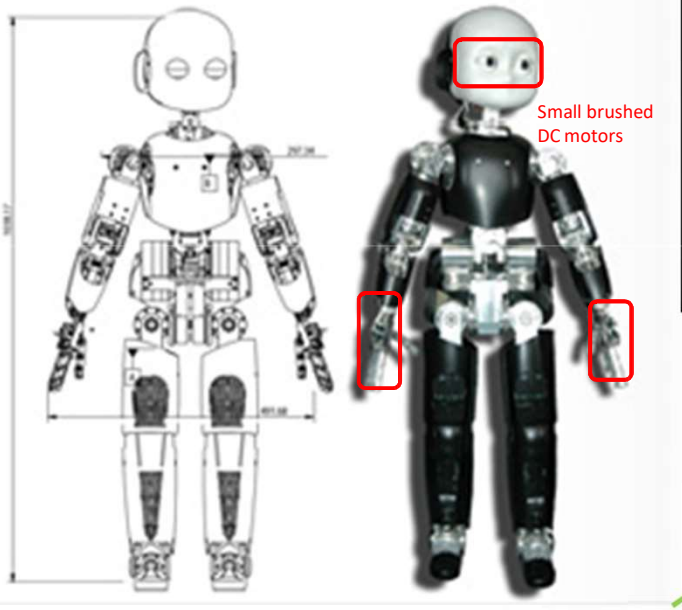
Specifications	
Power supply	12 to 24V
Communication	CAN Bus 2.0B 1Mbps
Motor number, type	4 Brushed DC motors
Output current	1A continuous, 2A overcurrent protection
Microcontrollers	Freescale DSP56F807, 80MHz, 144KB Flash, 8KB SRAM, 2KB EEPROM
Incremental encoders	Magnetic and optical, with an index up to 16.000cpr (see **LCORE**, **ROIE**, **ROIEL** info sheet)
Absolute encoders	Magnetic, SPI communication (see **AEA** and **AEA2** info sheet), Analog Hall effect sensors
Alarms	Overcurrent, I^2T, emergency button, sensor feedback, CAN communication, current sensors
Velocity loop speed	Up to 1KHz
Position loop speed	Up to 1KHz
Utilities	In field reprogramming, the board can be either stacked or connected trough a cable
Operating conditions	0 to 50°C, humidity <85% without condensation
Dimensions [LWH]	MC4 80x30x9mm / MCP 80x30x11mm
Weight	MC4 20g / MCP 20g

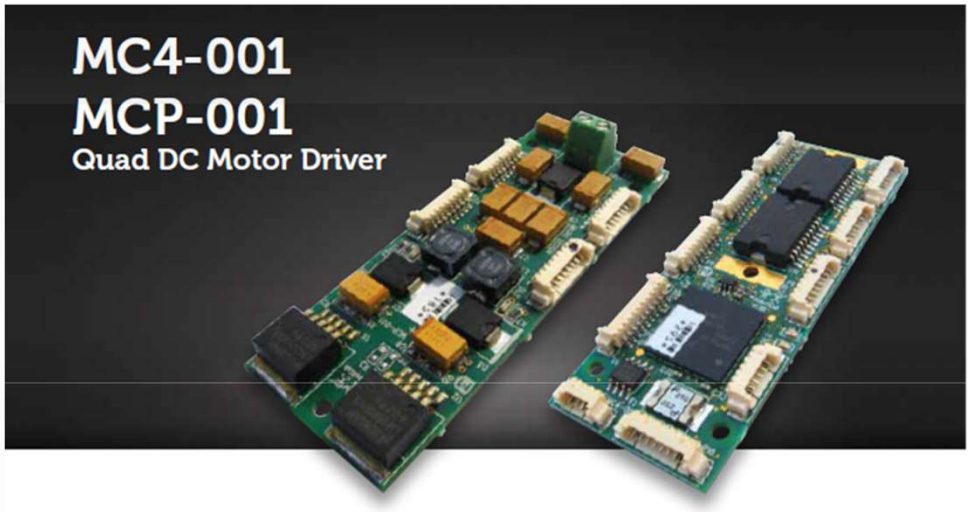
iCub CAN buses:

0 Head, Torso
1 Left arm
2 Right arm
3 Left leg
4 Right Leg
5 Left Hand
6 Right Hand
7-8-9 Skin (left arm, right arm, torso)

ISTITUTO ITALIANO
DI TECNOLOGIA

iCub Hardware (other CAN devices)

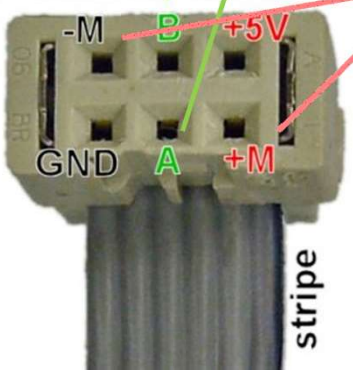


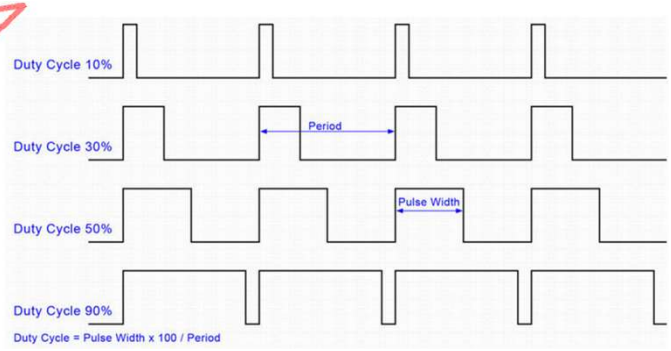


The MC4 board is a small motor controller capable of driving up to four brushed DC motors, using CAN bus communication for commands and control data exchange. This is complemented by a small power supply board (MCP).

Encoder

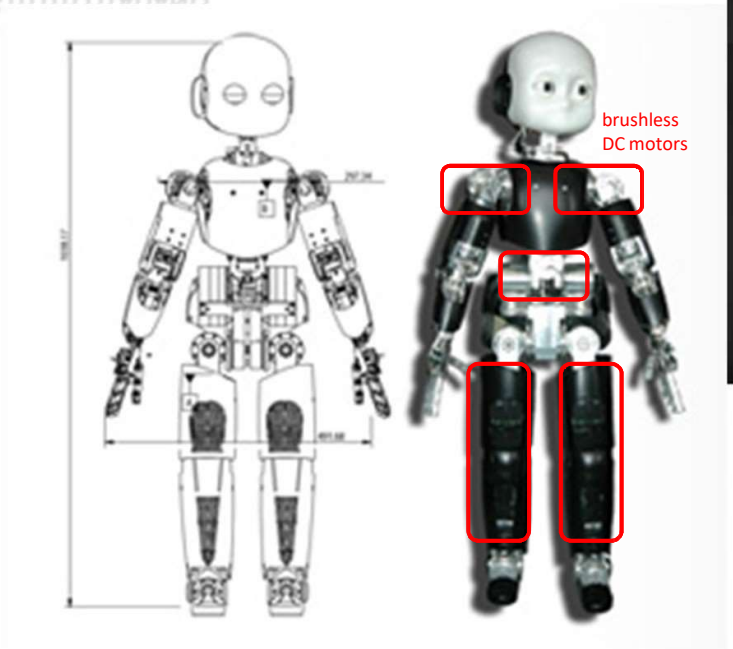
Motor PWM (pulse width modulation)





ISTITUTO ITALIANO
DI TECNOLOGIA

iCub Hardware (other CAN devices)



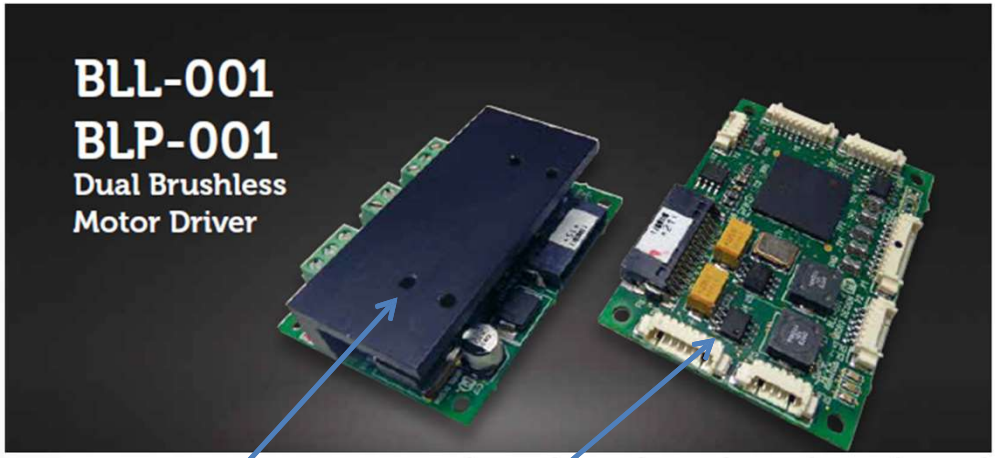
iCub CAN buses:
0 Head, Torso
1 Left arm
2 Right arm
3 Left leg
4 Right Leg
5 Left Hand
6 Right Hand
7-8-9 Skin (left arm, right arm, torso)

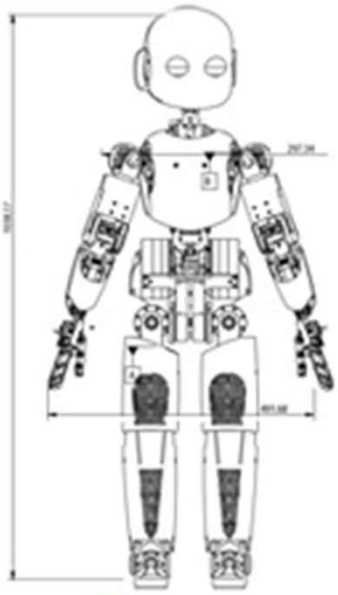
This motor controller is made of two separate boards. The BLL contains the logic circuits, while the BLP includes the power drivers. The controller is capable of driving up to two 250W Brushless DC motors using CAN bus communication for command and control data exchange.

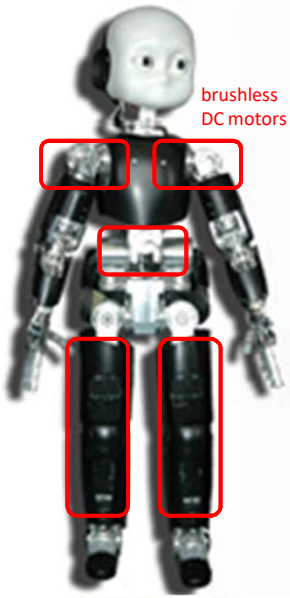
Specifications	
Power supply	18V to 48V
Communication	CAN Bus 2.0B 1Mbps
Motor number, type	Two Brushless DC
Output current	7A continuous, up to 25A I^2T limited
Microcontrollers	Freescale DSP56F807, 80MHz, 144KB Flash , 8KB SRAM, 2KB EEPROM
Incremental encoders	Magnetic and optical, with index up to 16.000cpr (see **LCORE**, **ROIE**, **ROIEL** info sheet)
Absolute encoders	Magnetic, SPI communication compatible with **AEA-001** and **AEA-002** (see info sheet), Hall effect sensors
Alarms	Overcurrent, overvoltage, undervoltage, I^2T, emergency button, sensor feedback, CAN communication, current sensors
Velocity loop speed	Up to 1KHz
Position loop speed	Up to 1KHz
Utilities	In field reprogramming, the board can be either stacked or connected trough a flat cable
Operating conditions	0 to 50°C, humidity <85% without condensation
Dimensions [LxWxH]	BLL 58x42x10mm / BLP 58x42x18mm
Weight	BLL 16g / BLP 47g

ISTITUTO ITALIANO
DI TECNOLOGIA

iCub Hardware (other CAN devices)



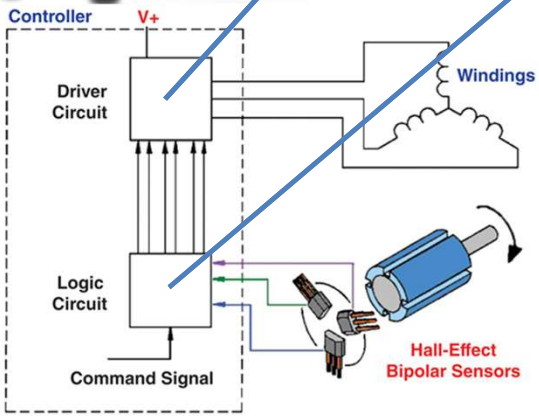


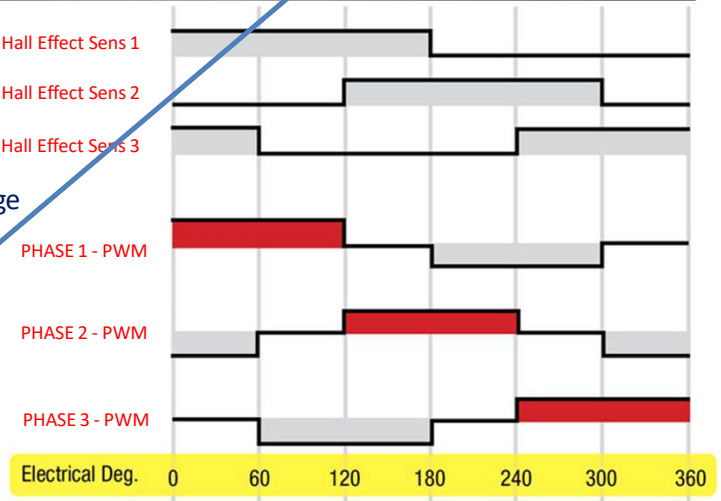


6 steps * 8 poles * 100 (harmonic drive reduction)
= 4800 ticks per revolution
hall effect sensors can be used as an encoder!

High voltage
(24-63V)

Low voltage
(5V)





Switch 1	1	1	1	0	0	0
Switch 2	0	0	1	1	1	0
Switch 3	1	0	0	0	1	1
Phase 1	+	+	0	-	-	0
Phase 2	-	0	+	+	0	-
Phase 3	0	-	-	0	+	+

ISTITUTO ITALIANO DI TECNOLOGIA

The Controller Area Network (CAN-bus)

- Typical for automotive applications. Microcontroller can exchange messages without an host computer
- 4 wires (V+ GND, CANH, CANL (twisted pair))
- 1 Mbps
- Robust to no noise
- 11 bits of address. On iCub address is splitted in three groups of bits (xxx yyyy zzzz)

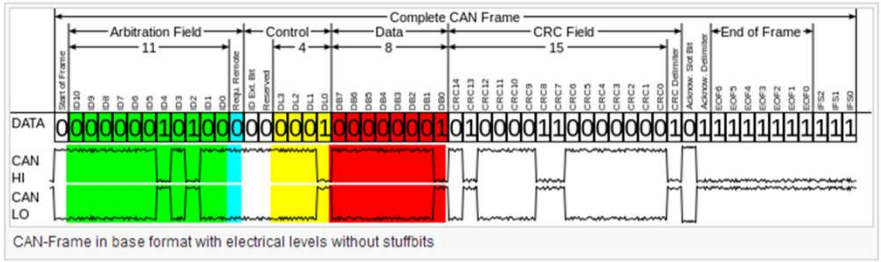
Data frame [edit]

The data frame is the only frame for actual data transmission. There are two message formats:

- Base frame format: with 11 identifier bits
- Extended frame format: with 29 identifier bits

The CAN standard requires the implementation must accept the base frame format and may accept the extended frame format, but must tolerate the extended frame format.

Base frame format [edit]



CAN-Frame in base format with electrical levels without stuffbits

The frame format is as follows:

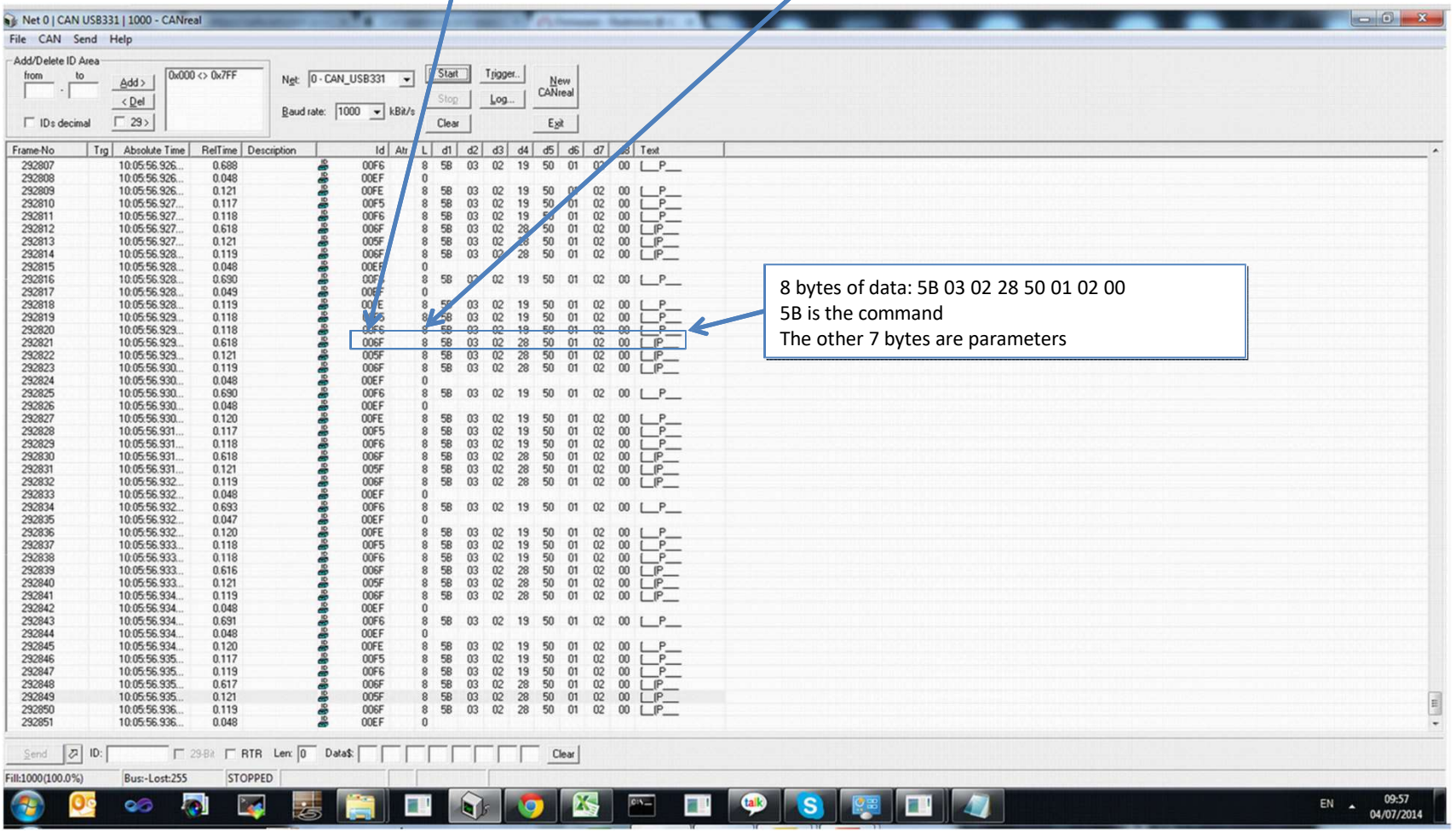
Field name	Length (bits)	Purpose
Start-of-frame	1	Denotes the start of frame transmission
Identifier (green)	11	A (unique) identifier for the data which also represents the message priority
Remote transmission request (RTR)	1	Dominant (0) (see Remote Frame below)
Identifier extension bit (IDE)	1	Declaring if 11 bit message ID or 29 bit message ID is used. Dominant (0) indicate 11 bit message ID while Recessive (1) indicate 29 bit message.
Reserved bit (r0)	1	Reserved bit (it must be set to dominant (0), but accepted as either dominant or recessive)
Data length code (DLC) (yellow)	4	Number of bytes of data (0–8 bytes)[a]
Data field (red)	0–64 (0-8 bytes)	Data to be transmitted (length in bytes dictated by DLC field)
CRC	15	Cyclic redundancy check
CRC delimiter	1	Must be recessive (1)
ACK slot	1	Transmitter sends recessive (1) and any receiver can assert a dominant (0)
ACK delimiter	1	Must be recessive (1)
End-of-frame (EOF)	7	Must be recessive (1)

The CAN bus

USB to CAN converter:
Can be attached to the CANbus wires to log data, debug commands, program the boards.

Id=006F
Class type: 0 (command) or 1 (status message)
From: board 6
To: board F

LEN=8



8 bytes of data: 5B 03 02 28 50 01 02 00
5B is the command
The other 7 bytes are parameters

The iCub CAN messages

Seven types of class messages:

- Class 0 (000 <XXXX> <YYYY>) command messages for motor control boards (e.g. position move) set/get messages (e.g. set pid)
- Class 1 (001 <XXXX> <TTTT> motor control messages messages broadcasted from a specific board to all the other devices (e.g. joint position, velocity, pwm)
- Class 2 (010 <XXXX> <YYYY>) command message for a sensor board (e.g. set period FT sensor)
- Class 3 (011 <XXXX> <TTTT>) broadcast message from a sensor board (e.g. force torque data)
- Class 7 (111 <XXXX> <yyyy>) canLoader message (e.g. firmware update)

<XXXX> source address (0-15)
<YYYY> destination address ID (0-15)
<TTTT> type of message (0-15)

Examples:
000 0000 0101 - a command from the user (0000) to board 5 (0101)
1 0101 0000 - the answer of command from board 5 (0100) to the user (0000)
2 0101 0100 - a broadcasted message of the current consumption 4 (0100) from board 5 (0101)

ISTITUTO ITALIANO
DI TECNOLOGIA

The iCub CAN messages

```
/*****************************************************************************
/
// command messages of class ICUBCANPROTO_CLASS_POLLING_MOTORCONTROL
/*****************************************************************************
/

#define ICUBCANPROTO_POL_MC_CMD__NO_MESSAGE                  0
#define ICUBCANPROTO_POL_MC_CMD__CONTROLLER_RUN              1
#define ICUBCANPROTO_POL_MC_CMD__CONTROLLER_IDLE             2
#define ICUBCANPROTO_POL_MC_CMD__TOGGLE_VERBOSE              3
#define ICUBCANPROTO_POL_MC_CMD__CALIBRATE_ENCODER           4
#define ICUBCANPROTO_POL_MC_CMD__ENABLE_PWM_PAD              5
#define ICUBCANPROTO_POL_MC_CMD__DISABLE_PWM_PAD             6
#define ICUBCANPROTO_POL_MC_CMD__GET_CONTROL_MODE            7
#define ICUBCANPROTO_POL_MC_CMD__MOTION_DONE                 8
#define ICUBCANPROTO_POL_MC_CMD__SET_CONTROL_MODE            9
#define ICUBCANPROTO_POL_MC_CMD__WRITE_FLASH_MEM             10
#define ICUBCANPROTO_POL_MC_CMD__READ_FLASH_MEM              11
#define ICUBCANPROTO_POL_MC_CMD__GET_ADDITIONAL_INFO         12
#define ICUBCANPROTO_POL_MC_CMD__SET_ADDITIONAL_INFO         13
#define ICUBCANPROTO_POL_MC_CMD__SET_SPEED_ESTIM_SHIFT       14
#define ICUBCANPROTO_POL_MC_CMD__SET_DEBUG_PARAM             18
#define ICUBCANPROTO_POL_MC_CMD__GET_DEBUG_PARAM             19
#define ICUBCANPROTO_POL_MC_CMD__GET_ENCODER_POSITION        20
#define ICUBCANPROTO_POL_MC_CMD__SET_DESIRED_POSITION        21
#define ICUBCANPROTO_POL_MC_CMD__GET_DESIRED_POSITION        22
#define ICUBCANPROTO_POL_MC_CMD__SET_DESIRED_VELOCITY        23
#define ICUBCANPROTO_POL_MC_CMD__GET_DESIRED_VELOCITY        24
#define ICUBCANPROTO_POL_MC_CMD__SET_DESIRED_ACCELER         25
#define ICUBCANPROTO_POL_MC_CMD__GET_DESIRED_ACCELER         26
#define ICUBCANPROTO_POL_MC_CMD__POSITION_MOVE               27
#define ICUBCANPROTO_POL_MC_CMD__VELOCITY_MOVE               28
#define ICUBCANPROTO_POL_MC_CMD__SET_ENCODER_POSITION        29
#define ICUBCANPROTO_POL_MC_CMD__SET_P_GAIN                  30
#define ICUBCANPROTO_POL_MC_CMD__GET_P_GAIN                  31
#define ICUBCANPROTO_POL_MC_CMD__SET_D_GAIN                  32
#define ICUBCANPROTO_POL_MC_CMD__GET_D_GAIN                  33
#define ICUBCANPROTO_POL_MC_CMD__SET_I_GAIN                  34
#define ICUBCANPROTO_POL_MC_CMD__GET_I_GAIN                  35
#define ICUBCANPROTO_POL_MC_CMD__SET_ILIM_GAIN               36
#define ICUBCANPROTO_POL_MC_CMD__GET_ILIM_GAIN               37
#define ICUBCANPROTO_POL_MC_CMD__SET_OFFSET                  38
#define ICUBCANPROTO_POL_MC_CMD__GET_OFFSET                  39
#define ICUBCANPROTO_POL_MC_CMD__SET_SCALE                   40
#define ICUBCANPROTO_POL_MC_CMD__GET_SCALE                   41
#define ICUBCANPROTO_POL_MC_CMD__SET_TLIM                    42
#define ICUBCANPROTO_POL_MC_CMD__GET_TLIM                    43
#define ICUBCANPROTO_POL_MC_CMD__SET_DESIRED_TORQUE          44
#define ICUBCANPROTO_POL_MC_CMD__GET_DESIRED_TORQUE          45
#define ICUBCANPROTO_POL_MC_CMD__STOP_TRAJECTORY             46
#define ICUBCANPROTO_POL_MC_CMD__SET_BOARD_ID                50
#define ICUBCANPROTO_POL_MC_CMD__GET_BOARD_ID                51
#define ICUBCANPROTO_POL_MC_CMD__SET_TORQUE_SOURCE           52
#define ICUBCANPROTO_POL_MC_CMD__GET_PID_ERROR               55
#define ICUBCANPROTO_POL_MC_CMD__GET_CALIBRATION_STATUS      56
#define ICUBCANPROTO_POL_MC_CMD__GET_ERROR_STATUS            60
#define ICUBCANPROTO_POL_MC_CMD__GET_ENCODER_VELOCITY        61
#define ICUBCANPROTO_POL_MC_CMD__SET_COMMAND_POSITION        62
#define ICUBCANPROTO_POL_MC_CMD__GET_PID_OUTPUT              63
#define ICUBCANPROTO_POL_MC_CMD__SET_MIN_POSITION            64
#define ICUBCANPROTO_POL_MC_CMD__GET_MIN_POSITION            65
#define ICUBCANPROTO_POL_MC_CMD__SET_MAX_POSITION            66
#define ICUBCANPROTO_POL_MC_CMD__GET_MAX_POSITION            67
#define ICUBCANPROTO_POL_MC_CMD__SET_MAX_VELOCITY            68
#define ICUBCANPROTO_POL_MC_CMD__GET_MAX_VELOCITY            69
#define ICUBCANPROTO_POL_MC_CMD__GET_ACTIVE_ENCODER_POSITION 70
#define ICUBCANPROTO_POL_MC_CMD__SET_ACTIVE_ENCODER_POSITION 71
#define ICUBCANPROTO_POL_MC_CMD__SET_CURRENT_LIMIT           72
#define ICUBCANPROTO_POL_MC_CMD__SET_BCAST_POLICY            73
#define ICUBCANPROTO_POL_MC_CMD__SET_VEL_SHIFT               74
#define ICUBCANPROTO_POL_MC_CMD__SET_OFFSET_ABS_ENCODER      75
#define ICUBCANPROTO_POL_MC_CMD__GET_OFFSET_ABS_ENCODER      76
#define ICUBCANPROTO_POL_MC_CMD__SET_SMOOTH_PID              77
#define ICUBCANPROTO_POL_MC_CMD__SET_TORQUE_PID              78
#define ICUBCANPROTO_POL_MC_CMD__GET_TORQUE_PID              79
#define ICUBCANPROTO_POL_MC_CMD__SET_TORQUE_PIDLIMITS        80
#define ICUBCANPROTO_POL_MC_CMD__GET_TORQUE_PIDLIMITS        81
#define ICUBCANPROTO_POL_MC_CMD__SET_POS_PID                 82
#define ICUBCANPROTO_POL_MC_CMD__GET_POS_PID                 83
#define ICUBCANPROTO_POL_MC_CMD__SET_POS_PIDLIMITS           84
#define ICUBCANPROTO_POL_MC_CMD__GET_POS_PIDLIMITS           85
#define ICUBCANPROTO_POL_MC_CMD__SET_VEL_TIMEOUT             86
#define ICUBCANPROTO_POL_MC_CMD__SET_IMPEDANCE_PARAMS        87
#define ICUBCANPROTO_POL_MC_CMD__GET_IMPEDANCE_PARAMS        88
#define ICUBCANPROTO_POL_MC_CMD__SET_IMPEDANCE_OFFSET        89
#define ICUBCANPROTO_POL_MC_CMD__GET_IMPEDANCE_OFFSET        90
#define ICUBCANPROTO_POL_MC_CMD__GET_FIRMWARE_VERSION        91
#define ICUBCANPROTO_POL_MC_CMD__SET_OPTICAL_ENC_RATIO       92
#define ICUBCANPROTO_POL_MC_CMD__SET_POS_STICTION_PARAMS     93
#define ICUBCANPROTO_POL_MC_CMD__GET_POS_STICTION_PARAMS     94
#define ICUBCANPROTO_POL_MC_CMD__SET_TORQUE_STICTION_PARAMS  95
#define ICUBCANPROTO_POL_MC_CMD__GET_TORQUE_STICTION_PARAMS  96
#define ICUBCANPROTO_POL_MC_CMD__SET_BACKEMF_PARAMS          97
#define ICUBCANPROTO_POL_MC_CMD__GET_BACKEMF_PARAMS          98
#define ICUBCANPROTO_POL_MC_CMD__SET_MODEL_PARAMS            99
#define ICUBCANPROTO_POL_MC_CMD__GET_MODEL_PARAMS            100
#define ICUBCANPROTO_POL_MC_CMD__SET_CURRENT_PID             101
#define ICUBCANPROTO_POL_MC_CMD__GET_CURRENT_PID             102
#define ICUBCANPROTO_POL_MC_CMD__SET_CURRENT_PIDLIMITS       103
#define ICUBCANPROTO_POL_MC_CMD__GET_CURRENT_PIDLIMITS       104
#define ICUBCANPROTO_POL_MC_CMD__SET_VELOCITY_PID            105
#define ICUBCANPROTO_POL_MC_CMD__GET_VELOCITY_PID            106
#define ICUBCANPROTO_POL_MC_CMD__SET_VELOCITY_PIDLIMITS      107
#define ICUBCANPROTO_POL_MC_CMD__GET_VELOCITY_PIDLIMITS      108
#define ICUBCANPROTO_POL_MC_CMD__SET_DESIRED_CURRENT         109
#define ICUBCANPROTO_POL_MC_CMD__GET_DESIRED_CURRENT         110
#define ICUBCANPROTO_POL_MC_CMD__SET_PERIODIC_MSG_CONTENTS   111
#define ICUBCANPROTO_POL_MC_CMD__SET_I2T_PARAMS              112
#define ICUBCANPROTO_POL_MC_CMD__GET_I2T_PARAMS              113
#define ICUBCANPROTO_POL_MC_CMD__SET_OPENLOOP_PARAMS         114
#define ICUBCANPROTO_POL_MC_CMD__GET_OPENLOOP_PARAMS         115
#define ICUBCANPROTO_POL_MC_CMD__SET_INTERACTION_MODE        116
#define ICUBCANPROTO_POL_MC_CMD__GET_INTERACTION_MODE        117
#define ICUBCANPROTO_POL_MC_CMD_MAXNUM
ICUBCANPROTO_POL_MC_CMD__GET_INTERACTION_MODE + 1
```

ISTITUTO ITALIANO
DI TECNOLOGIA

The iCub CAN messages

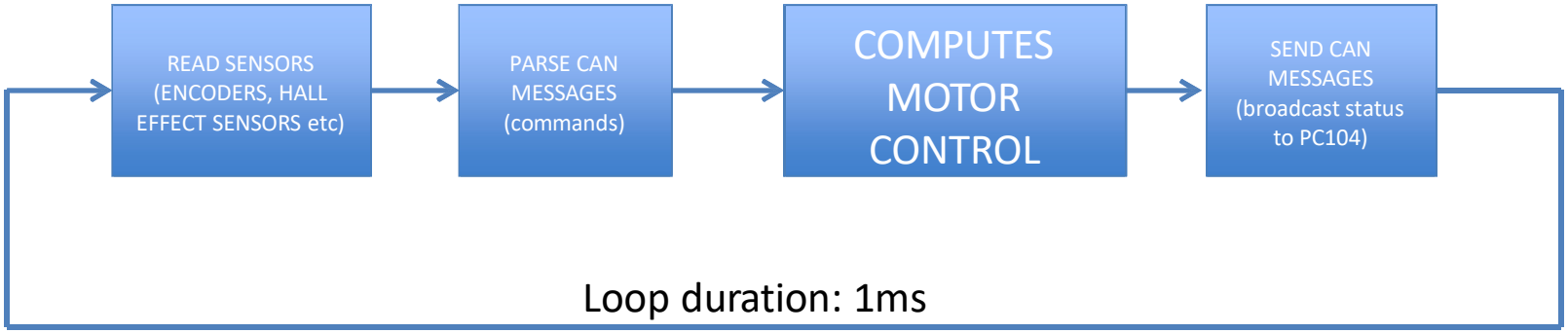
```
/*********************************************************************
/
// messages of class 1: ICUBCANPROTO_CLASS_PERIODIC_MOTORCONTROL
/*********************************************************************
#define ICUBCANPROTO_PER_MC_MSG__2FOC                    0
#define ICUBCANPROTO_PER_MC_MSG__POSITION                1
#define ICUBCANPROTO_PER_MC_MSG__PID_VAL                 2
#define ICUBCANPROTO_PER_MC_MSG__STATUS                  3
#define ICUBCANPROTO_PER_MC_MSG__CURRENT                 4
#define ICUBCANPROTO_PER_MC_MSG__OVERFLOW                5
#define ICUBCANPROTO_PER_MC_MSG__PRINT                   6
#define ICUBCANPROTO_PER_MC_MSG__VELOCITY                7
#define ICUBCANPROTO_PER_MC_MSG__PID_ERROR               8
#define ICUBCANPROTO_PER_MC_MSG__DEBUG                   9
#define ICUBCANPROTO_PER_MC_MSG__MOTOR_POSITION          10
#define ICUBCANPROTO_PER_MC_MSG__MOTOR_SPEED             11
#define ICUBCANPROTO_PER_MC_MSG__ADDITIONAL_STATUS       12
#define ICUBCANPROTO_PER_MC_MSG__EMSTO2FOC_DESIRED_CURRENT   15
```

```
/*********************************************************************/
// command messages of class 3: ICUBCANPROTO_CLASS_PERIODIC_ANALOGSENSOR
/*********************************************************************/
#define ICUBCANPROTO_PER_AS_MSG__FORCE_VECTOR    0xA // Transmit Torque values t1 t1 t2 t2 t3 t3
#define ICUBCANPROTO_PER_AS_MSG__TORQUE_VECTOR   0xB // Transmit Force  values f1 f1 f2 f2 f3 f3
#define ICUBCANPROTO_PER_AS_MSG__HES0TO6         0xC //hall effect sensors from 0  to 6   *8bits
#define ICUBCANPROTO_PER_AS_MSG__HES7TO14        0xD //hall effect sensors from 7  to 14  *8bits
```

```
/*********************************************************************
/
// command messages of class 2: ICUBCANPROTO_CLASS_POLLING_ANALOGSENSOR
/*********************************************************************
#define ICUBCANPROTO_POL_AS_CMD__NONE              0x0
#define ICUBCANPROTO_POL_AS_CMD__SET_IIR           0x1         // select IIR filters parameters
#define ICUBCANPROTO_POL_AS_CMD__SET_MATRIX_RC     0x3         // Set SG to TF trasformation matrix
#define ICUBCANPROTO_POL_AS_CMD__SET_CH_DAC        0x4         // set DAC for channel x
#define ICUBCANPROTO_POL_AS_CMD__SELECT_ACTIVE_CH  0x5         // set active channels (activation mask) only active channels are
#define ICUBCANPROTO_POL_AS_CMD__CALIBRATE_OFFSET  0x6         transmitted
#define ICUBCANPROTO_POL_AS_CMD__SET_TXMODE        0x7         // set the calibration offset
#define ICUBCANPROTO_POL_AS_CMD__SET_CANDATARATE   0x8         // set continuous/on demand transmission mode
#define ICUBCANPROTO_POL_AS_CMD__SAVE2EE           0x9         // set board CAN speed in milliseconds minimum, datarate 210ms
#define ICUBCANPROTO_POL_AS_CMD__GET_MATRIX_RC     0xA         // save Config to EE
#define ICUBCANPROTO_POL_AS_CMD__GET_CH_DAC        0xB         // Get TF trasformation matrix
#define ICUBCANPROTO_POL_AS_CMD__GET_CH_ADC        0xC         // Get DAC for channel x
#define ICUBCANPROTO_POL_AS_CMD__FILTER_EN         0xD         // Get ADC for channel x
#define ICUBCANPROTO_POL_AS_CMD__MUX_EN            0xE         // ENABLE/DISABLES FILTER
#define ICUBCANPROTO_POL_AS_CMD__MUX_NUM           0xF         // ENABLE/DISABLES MUX
#define ICUBCANPROTO_POL_AS_CMD__SET_RESOLUTION          0x10 //set data resolution
#define ICUBCANPROTO_POL_AS_CMD__SET_MATRIX_G            0x11 //set matrix gain
#define ICUBCANPROTO_POL_AS_CMD__GET_MATRIX_G            0x12 //get matrix gain
#define ICUBCANPROTO_POL_AS_CMD__SET_CALIB_TARE          0x13 //set/reset calibration
#define ICUBCANPROTO_POL_AS_CMD__GET_CALIB_TARE          0x14 //get calibration tare
#define ICUBCANPROTO_POL_AS_CMD__SET_CURR_TARE           0x15 //set/reset current tare
#define ICUBCANPROTO_POL_AS_CMD__GET_CURR_TARE           0x16 //get current tare
#define ICUBCANPROTO_POL_AS_CMD__SET_FULL_SCALES         0x17 //set/reset current tare
#define ICUBCANPROTO_POL_AS_CMD__GET_FULL_SCALES         0x18 //get current tare
#define ICUBCANPROTO_POL_AS_CMD__SET_SERIAL_NO           0x19 //set/reset current tare
#define ICUBCANPROTO_POL_AS_CMD__GET_SERIAL_NO           0x1A //get current tare
#define ICUBCANPROTO_POL_AS_CMD__GET_EEPROM_STATUS       0x1B //get eeprom status (saved/not saved)
#define ICUBCANPROTO_POL_AS_CMD__GET_FW_VERSION          0X1C //get fw version and check can protocol
#define ICUBCANPROTO_POL_AS_CMD__SET_BOARD_ADX           0x32 // Set board CAN address
```

Motor control firmware

Firmware main loop:



- Written in C
- Freescale Codewarrior crosscompiler for Motorola 56f807 microcontroller
- Source code available:

  •A library (Hardware Abstraction Layer):
  https://svn.code.sf.net/p/robotcub/code/trunk/iCub/firmware/libDsp56f807

  •The controller application:
  https://svn.code.sf.net/p/robotcub/code/trunk/iCub/firmware/motorControllerDsp56f807

iCub Firmware versions

ISTITUTO ITALIANO
DI TECNOLOGIA

CFW2CAN0: Head and Torso

Head

boardLabel		canDeviceNum	boardId	firmware
0B0	->	0 (cfw2can)	1	2BLLDC.1.62.out.S
0B1	->	0 (cfw2can)	2 (3)	4DC.2.15.out.S

Torso

boardLabel		canDeviceNum	boardId	firmware
0B3	->	0 (cfw2can)	6	2BLL.2.52.out.S
0B4	->	0 (cfw2can)	5	2BLL.2.54.out.S
6SG (optional)	->	0 (cfw2can)	14	6sg.hex

CFW2CAN1: Left arm

boardLabel		canDeviceNum	boardId	firmware
1B0	->	1 (cfw2can)	1	2BLL.2.50.out.S
1B1	->	1 (cfw2can)	2	2BLL.2.57.out.S
1B2	->	1 (cfw2)	3 (4)	4DC.2.19.out.S
virtual FT joint sensor (shoulder)	->	1 (cfw2can)	12	- - -
virtual FT joint sensor (wrist)	->	1 (cfw2can)	11	- - -
Strain	->	1 (cfw2can)	13	strain.hex
6SG Shoulder (optional)	->	1 (cfw2can)	9	6sg.hex

CFW2CAN5: Left hand

boardLabel		canDeviceNum	boardId	firmware
1B3	->	5 (cfw2can)	5 (6)	4DC.2.28.out.S
1B4	->	5 (cfw2can)	7 (8)	4DC.2.30.out.S
MAIS	->	5 (cfw2can)	14	mais.hex

CFW2CAN2: Right arm

boardLabel		canDeviceNum	boardId	firmware
2B0	->	2 (cfw2can)	1	2BLL.2.50.out.S
2B1	->	2 (cfw2can)	2	2BLL.2.57.out.S
2B2	->	2 (cfw2can)	3 (4)	4DC.2.19.out.S
virtual FT joint sensor (shoulder)	->	2 (cfw2can)	12	- - -
virtual FT joint sensor (wrist)	->	2 (cfw2can)	11	- - -
Strain	->	2 (cfw2can)	13	strain.hex
6SG Shoulder (optional)	->	1 (cfw2can)	9	6sg.hex

CFW2CAN6: Right hand

boardLabel		canDeviceNum	boardId	firmware
2B3	->	6 (cfw2can)	5 (6)	4DC.2.28.out.S
2B4	->	6 (cfw2can)	7 (8)	4DC.2.30.out.S
MAIS	->	6 (cfw2can)	14	mais.hex

CFW2CAN3: left leg

boardLabel		canDeviceNum	boardId	firmware
3B5	->	3 (cfw2can)	5	2BLL.2.51.out.S
3B6	->	3 (cfw2can)	6	2BLL.2.51.out.S
3B7	->	3 (cfw2can)	7	2BLL.2.51.out.S
virtual FT joint sensor	->	3 (cfw2can)	12	- - -
Strain	->	3 (cfw2can)	13	strain.hex
6SG Upper Leg (optional)	->	3 (cfw2can)	1	6sg.hex
6SG Lower Leg (optional)	->	3 (cfw2can)	2	6sg.hex

CFW2CAN4: right leg

boardLabel		canDeviceNum	boardId	firmware
3B8	->	4 (cfw2can)	8	2BLL.2.51.out.S
3B9	->	4 (cfw2can)	9	2BLL.2.51.out.S
3B10	->	4 (cfw2can)	10	2BLL.2.51.out.S
virtual FT joint sensor	->	4 (cfw2can)	11	- - -
Strain	->	4 (cfw2can)	14	strain.hex
6SG Upper Leg (optional)	->	4 (cfw2can)	1	6sg.hex
6SG Lower Leg (optional)	->	4 (cfw2can)	2	6sg.hex

CFW2CAN7: Skin Left Arm

boardLabel		canDeviceNum	boardId	firmware
1B7 (HAND)	->	7 (cfw2can)	14	skin.hex
1B8 (LEFT LOWER FOREARM)	->	7 (cfw2can)	12	skin.hex
1B9 (LEFT UPPER FOREARM)	->	7 (cfw2can)	13	skin.hex
1B10 (LEFT UPPER EXTERNAL)	->	7 (cfw2can)	10	skin.hex
1B11 (LEFT UPPER BOTTOM)	->	7 (cfw2can)	11	skin.hex
1B12 (LEFT UPPER INTERNAL)	->	7 (cfw2can)	9	skin.hex
1B13 (LEFT LOWER UPPERARM)	->	7 (cfw2can)	8	skin.hex

CFW2CAN8: Skin Right Arm

boardLabel		canDeviceNum	boardId	firmware
2B7 (HAND)	->	8 (cfw2can)	14	skin.hex
2B8 (LEFT LOWER FOREARM)	->	8 (cfw2can)	12	skin.hex
2B9 (LEFT UPPER FOREARM)	->	8 (cfw2can)	13	skin.hex
2B10 (RIGHT UPPER EXTERNAL)	->	8 (cfw2can)	10	skin.hex
2B11 (RIGHT UPPER BOTTOM)	->	8 (cfw2can)	11	skin.hex
2B12 (RIGHT UPPER INTERNAL)	->	8 (cfw2can)	9	skin.hex
2B13 (RIGHT LOWER UPPERARM)	->	8 (cfw2can)	8	skin.hex

CFW2CAN9: Skin Torso

boardLabel		canDeviceNum	boardId	firmware
0B7	->	9 (cfw2can)	7	skin.hex
0B8	->	9 (cfw2can)	8	skin.hex
0B9	->	9 (cfw2can)	9	skin.hex
0B10	->	9 (cfw2can)	10	skin.hex

More info:
http://wiki.icub.org/wiki/Can_addresses_and_associated_firmware

http://wiki.icub.org/wiki/Firmware

User has to update fw regularly from the repository!
https://svn.code.sf.net/p/robotcub/code/trunk/iCub/firmware/build

Robot Interface:
- asks for the fw version to the boards
- checks if the fw use the same CAN protocol
  - CAN protocol version is increased when new features are implemented
  - If CAN protocol version is not compatible (too old), a fw update is required (robot Interface will terminate)

ISTITUTO ITALIANO
DI TECNOLOGIA

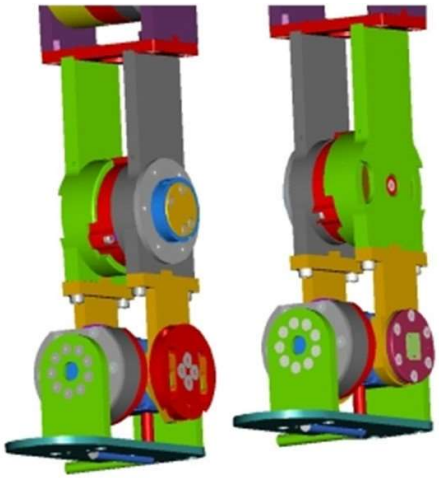
iCub Firmware versions

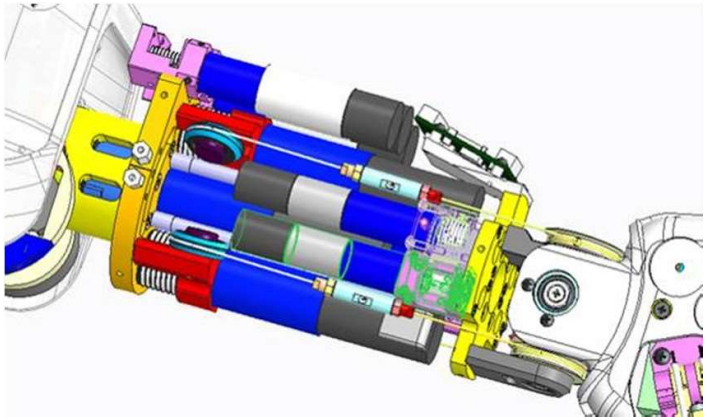
Why many firmware versions?

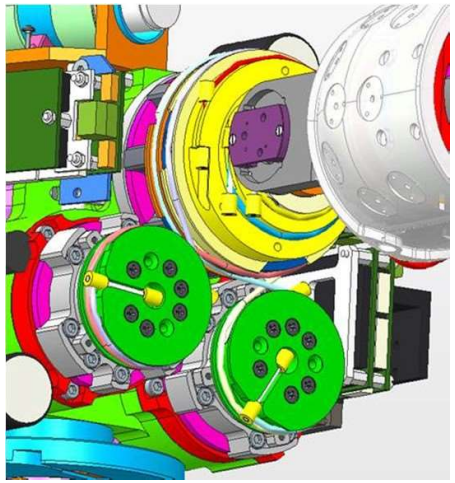
- Different type of motors
  - Brushless on major joints: arms, legs, torso
  - DC motors on small joints: eyes, fingers

- Different type of sensors (absolute magnetic/optical encoders, ability to perform force control using torque data)

- Joint couplings. E.g.
  - Firmware 1.51 (legs) implement a 1:1 coupling (one joint controlled by one motor)
  - Firmware 1.52 (waist) implements a 2:2 differential coupling
  - Firmware 1.50-1.57 (shoulder) implements a 3:3 differential coupling







Almost every joint has different, customized controller application -> many different firmware types

iCub configuration

Example: icub-main\app\robots\iCubGenova01\icub_all.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<robot name="icub">
  <!-- cartesian -->
  <devices file="cartesian/left_arm_cartesian.xml" />
  <devices file="cartesian/right_arm_cartesian.xml" />

  <!-- motor controllers -->
  <devices file="wrappers/motorControl/left_arm_mc_wrapper.xml" />
  <devices file="wrappers/motorControl/right_arm_mc_wrapper.xml" />
  <devices file="wrappers/motorControl/left_leg_mc_wrapper.xml" />
  <devices file="wrappers/motorControl/right_leg_mc_wrapper.xml" />
  <devices file="wrappers/motorControl/head_mc_wrapper.xml" />
  <devices file="wrappers/motorControl/torso_mc_wrapper.xml" />
  <devices file="hardware/motorControl/icub_left_arm.xml" />
  <devices file="hardware/motorControl/icub_left_hand.xml" />
  <devices file="hardware/motorControl/icub_right_arm.xml" />
  <devices file="hardware/motorControl/icub_right_hand.xml" />
  <devices file="hardware/motorControl/icub_left_leg.xml" />
  <devices file="hardware/motorControl/icub_right_leg.xml" />
  <devices file="hardware/motorControl/icub_head.xml" />
  <devices file="hardware/motorControl/icub_torso.xml" />

  <!-- VIRTUAL ANALOG SENSORS -->
  <devices file="wrappers/VFT/left_arm_VFT_wrapper.xml" />
  <devices file="wrappers/VFT/left_leg_VFT_wrapper.xml" />
  <devices file="wrappers/VFT/right_arm_VFT_wrapper.xml" />
  <devices file="wrappers/VFT/right_leg_VFT_wrapper.xml" />
  <devices file="wrappers/VFT/torso_VFT_wrapper.xml" />
  <devices file="hardware/VFT/left_arm_virtual_strain.xml" />
  <devices file="hardware/VFT/left_leg_virtual_strain.xml" />
  <devices file="hardware/VFT/right_arm_virtual_strain.xml" />
  <devices file="hardware/VFT/right_leg_virtual_strain.xml" />
  <devices file="hardware/VFT/torso_virtual_strain.xml" />

  <!-- REAL ANALOG SENSORS -->
  <devices file="wrappers/FT/left_arm_FT_wrapper.xml" />
  <devices file="wrappers/FT/left_leg_FT_wrapper.xml" />
  <devices file="wrappers/FT/left_foot_FT_wrapper.xml" />
  <devices file="wrappers/FT/right_arm_FT_wrapper.xml" />
  <devices file="wrappers/FT/right_leg_FT_wrapper.xml" />
  <devices file="wrappers/FT/right_foot_FT_wrapper.xml" />
  <devices file="wrappers/MAIS/left_hand_mais_wrapper.xml" />
  <devices file="wrappers/MAIS/right_hand_mais_wrapper.xml" />
  <devices file="hardware/FT/left_arm_strain.xml" />
  <devices file="hardware/FT/left_leg_strain.xml" />
  <devices file="hardware/FT/left_foot_strain.xml" />
  <devices file="hardware/FT/right_arm_strain.xml" />
  <devices file="hardware/FT/right_leg_strain.xml" />
  <devices file="hardware/FT/right_foot_strain.xml" />
  <devices file="hardware/MAIS/left_hand_mais.xml" />
  <devices file="hardware/MAIS/right_hand_mais.xml" />

  <!-- SKIN -->
  <devices file="wrappers/skin/left_arm_skin_wrapper.xml" />
  <devices file="wrappers/skin/right_arm_skin_wrapper.xml" />
  <devices file="wrappers/skin/torso_skin_wrapper.xml" />
  <devices file="hardware/skin/left_arm.xml" />
  <devices file="hardware/skin/right_arm.xml" />
  <devices file="hardware/skin/torso.xml" />

  <!-- CALIBRATORS -->
  <devices file="calibrators/head_calib.xml" />
  <devices file="calibrators/torso_calib.xml" />
  <devices file="calibrators/right_leg_calib.xml" />
  <devices file="calibrators/left_leg_calib.xml" />
  <devices file="calibrators/left_arm_calib.xml" />
  <devices file="calibrators/right_arm_calib.xml" />
  <devices file="calibrators/left_hand_calib.xml" />
  <devices file="calibrators/right_hand_calib.xml" />
</robot>
```

ISTITUTO ITALIANO
DI TECNOLOGIA

iCub configuration

Example: icub-main\app\robots\iCubGenova01\icub_all.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<robot name="icub">
  <!-- cartesian -->
  <devices file="cartesian/left_arm_cartesian.xml" />
  <devices file="cartesian/right_arm_cartesian.xml" />

  <!-- motor controllers -->
  <devices file="wrappers/motorControl/left_arm_mc_wrapper.xml" />
  <devices file="wrappers/motorControl/right_arm_mc_wrapper.xml" />
  <devices file="wrappers/motorControl/left_leg_mc_wrapper.xml" />
  <devices file="wrappers/motorControl/right_leg_mc_wrapper.xml" />
  <devices file="wrappers/motorControl/head_mc_wrapper.xml" />
  <devices file="wrappers/motorControl/torso_mc_wrapper.xml" />
CAN1 --> <devices file="hardware/motorControl/icub_left_arm.xml" />
CAN5 --> <devices file="hardware/motorControl/icub_left_hand.xml" />
CAN2 --> <devices file="hardware/motorControl/icub_right_arm.xml" />
CAN6 --> <devices file="hardware/motorControl/icub_right_hand.xml" />
CAN3 --> <devices file="hardware/motorControl/icub_left_leg.xml" />
CAN4 --> <devices file="hardware/motorControl/icub_right_leg.xml" />
CAN0 --> <devices file="hardware/motorControl/icub_head.xml" />
CAN0 --> <devices file="hardware/motorControl/icub_torso.xml" />

  <!-- VIRTUAL ANALOG SENSORS -->
  <devices file="wrappers/VFT/left_arm_VFT_wrapper.xml" />
  <devices file="wrappers/VFT/left_leg_VFT_wrapper.xml" />
  <devices file="wrappers/VFT/right_arm_VFT_wrapper.xml" />
  <devices file="wrappers/VFT/right_leg_VFT_wrapper.xml" />
  <devices file="wrappers/VFT/torso_VFT_wrapper.xml" />
CAN1 --> <devices file="hardware/VFT/left_arm_virtual_strain.xml" />
CAN3 --> <devices file="hardware/VFT/left_leg_virtual_strain.xml" />
CAN2 --> <devices file="hardware/VFT/right_arm_virtual_strain.xml" />
CAN4 --> <devices file="hardware/VFT/right_leg_virtual_strain.xml" />
CAN0 --> <devices file="hardware/VFT/torso_virtual_strain.xml" />

  <!-- REAL ANALOG SENSORS -->
  <devices file="wrappers/FT/left_arm_FT_wrapper.xml" />
  <devices file="wrappers/FT/left_leg_FT_wrapper.xml" />
  <devices file="wrappers/FT/left_foot_FT_wrapper.xml" />
  <devices file="wrappers/FT/right_arm_FT_wrapper.xml" />
  <devices file="wrappers/FT/right_leg_FT_wrapper.xml" />
  <devices file="wrappers/FT/right_foot_FT_wrapper.xml" />
  <devices file="wrappers/MAIS/left_hand_mais_wrapper.xml" />
  <devices file="wrappers/MAIS/right_hand_mais_wrapper.xml" />
  <devices file="hardware/FT/left_arm_strain.xml" />          <-- CAN1
  <devices file="hardware/FT/left_leg_strain.xml" />          <-- CAN3
  <devices file="hardware/FT/left_foot_strain.xml" />         <-- CAN3
  <devices file="hardware/FT/right_arm_strain.xml" />         <-- CAN2
  <devices file="hardware/FT/right_leg_strain.xml" />         <-- CAN4
  <devices file="hardware/FT/right_foot_strain.xml" />        <-- CAN4
  <devices file="hardware/MAIS/left_hand_mais.xml" />         <-- CAN5
  <devices file="hardware/MAIS/right_hand_mais.xml" />        <-- CAN6

  <!-- SKIN -->
  <devices file="wrappers/skin/left_arm_skin_wrapper.xml" />
  <devices file="wrappers/skin/right_arm_skin_wrapper.xml" />
  <devices file="wrappers/skin/torso_skin_wrapper.xml" />
  <devices file="hardware/skin/left_arm.xml" />
  <devices file="hardware/skin/right_arm.xml" />
  <devices file="hardware/skin/torso.xml" />

  <!-- CALIBRATORS -->
  <devices file="calibrators/head_calib.xml" />
  <devices file="calibrators/torso_calib.xml" />
  <devices file="calibrators/right_leg_calib.xml" />
  <devices file="calibrators/left_leg_calib.xml" />
  <devices file="calibrators/left_arm_calib.xml" />
  <devices file="calibrators/right_arm_calib.xml" />
  <devices file="calibrators/left_hand_calib.xml" />
  <devices file="calibrators/right_hand_calib.xml" />
</robot>
```

iCub configuration

Example: icub-main\app\robots\iCubGenova01\icub_all.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<robot name="icub">
  <!-- cartesian -->
  <devices file="cartesian/left_arm_cartesian.xml" />
  <devices file="cartesian/right_arm_cartesian.xml" />

  <!-- motor controllers -->
  <devices file="wrappers/motorControl/left_arm_mc_wrapper.xml" />
  <devices file="wrappers/motorControl/right_arm_mc_wrapper.xml" />
  <devices file="wrappers/motorControl/left_leg_mc_wrapper.xml" />
  <devices file="wrappers/motorControl/right_leg_mc_wrapper.xml" />
  <devices file="wrappers/motorControl/head_mc_wrapper.xml" />
  <devices file="wrappers/motorControl/torso_mc_wrapper.xml" />
  <devices file="hardware/motorControl/icub_left_arm.xml" />
  <devices file="hardware/motorControl/icub_left_hand.xml" />
  <devices file="hardware/motorControl/icub_right_arm.xml" />
  <devices file="hardware/motorControl/icub_right_hand.xml" />
  <devices file="hardware/motorControl/icub_left_leg.xml" />
  <devices file="hardware/motorControl/icub_right_leg.xml" />
  <devices file="hardware/motorControl/icub_head.xml" />
  <devices file="hardware/motorControl/icub_torso.xml" />

  <!-- VIRTUAL ANALOG SENSORS -->
  <devices file="wrappers/VFT/left_arm_VFT_wrapper.xml" />
  <devices file="wrappers/VFT/left_leg_VFT_wrapper.xml" />
  <devices file="wrappers/VFT/right_arm_VFT_wrapper.xml" />
  <devices file="wrappers/VFT/right_leg_VFT_wrapper.xml" />
  <devices file="wrappers/VFT/torso_VFT_wrapper.xml" />
  <devices file="hardware/VFT/left_arm_virtual_strain.xml" />
  <devices file="hardware/VFT/left_leg_virtual_strain.xml" />
  <devices file="hardware/VFT/right_arm_virtual_strain.xml" />
  <devices file="hardware/VFT/right_leg_virtual_strain.xml" />
  <devices file="hardware/VFT/torso_virtual_strain.xml" />

  <!-- REAL ANALOG SENSORS -->
  <devices file="wrappers/FT/left_arm_FT_wrapper.xml" />
  <devices file="wrappers/FT/left_leg_FT_wrapper.xml" />
  <devices file="wrappers/FT/left_foot_FT_wrapper.xml" />
  <devices file="wrappers/FT/right_arm_FT_wrapper.xml" />
  <devices file="wrappers/FT/right_leg_FT_wrapper.xml" />
  <devices file="wrappers/FT/right_foot_FT_wrapper.xml" />
  <devices file="wrappers/MAIS/left_hand_mais_wrapper.xml" />
  <devices file="wrappers/MAIS/right_hand_mais_wrapper.xml" />
  <devices file="hardware/FT/left_arm_strain.xml" />
  <devices file="hardware/FT/left_leg_strain.xml" />
  <devices file="hardware/FT/left_foot_strain.xml" />
  <devices file="hardware/FT/right_arm_strain.xml" />
  <devices file="hardware/FT/right_leg_strain.xml" />
  <devices file="hardware/FT/right_foot_strain.xml" />
  <devices file="hardware/MAIS/left_hand_mais.xml" />
  <devices file="hardware/MAIS/right_hand_mais.xml" />

  <!-- SKIN -->
  <devices file="wrappers/skin/left_arm_skin_wrapper.xml" />
  <devices file="wrappers/skin/right_arm_skin_wrapper.xml" />
  <devices file="wrappers/skin/torso_skin_wrapper.xml" />
  <devices file="hardware/skin/left_arm.xml" />
  <devices file="hardware/skin/right_arm.xml" />
  <devices file="hardware/skin/torso.xml" />

  <!-- CALIBRATORS -->
  <devices file="calibrators/head_calib.xml" />
  <devices file="calibrators/torso_calib.xml" />
  <devices file="calibrators/right_leg_calib.xml" />
  <devices file="calibrators/left_leg_calib.xml" />
  <devices file="calibrators/left_arm_calib.xml" />
  <devices file="calibrators/right_arm_calib.xml" />
  <devices file="calibrators/left_hand_calib.xml" />
  <devices file="calibrators/right_hand_calib.xml" />
</robot>
```

iCub configuration

Example: icub-main\app\robots\iCubGenova01\icub_all.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<robot name="icub">
  <!-- cartesian -->
  <devices file="cartesian/left_arm_cartesian.xml" />
  <devices file="cartesian/right_arm_cartesian.xml" />

  <!-- motor controllers -->
  <devices file="wrappers/motorControl/left_arm_mc_wrapper.xml" />
  <devices file="wrappers/motorControl/right_arm_mc_wrapper.xml" />
  <devices file="wrappers/motorControl/left_leg_mc_wrapper.xml" />
  <devices file="wrappers/motorControl/right_leg_mc_wrapper.xml" />
  <devices file="wrappers/motorControl/head_mc_wrapper.xml" />
  <devices file="wrappers/motorControl/torso_mc_wrapper.xml" />
  <devices file="hardware/motorControl/icub_left_arm.xml" />
  <devices file="hardware/motorControl/icub_left_hand.xml" />
  <devices file="hardware/motorControl/icub_right_arm.xml" />
  <devices file="hardware/motorControl/icub_right_hand.xml" />
  <devices file="hardware/motorControl/icub_left_leg.xml" />
  <devices file="hardware/motorControl/icub_right_leg.xml" />
  <devices file="hardware/motorControl/icub_head.xml" />
  <devices file="hardware/motorControl/icub_torso.xml" />

  <!-- VIRTUAL ANALOG SENSORS -->
  <devices file="wrappers/VFT/left_arm_VFT_wrapper.xml" />
  <devices file="wrappers/VFT/left_leg_VFT_wrapper.xml" />
  <devices file="wrappers/VFT/right_arm_VFT_wrapper.xml" />
  <devices file="wrappers/VFT/right_leg_VFT_wrapper.xml" />
  <devices file="wrappers/VFT/torso_VFT_wrapper.xml" />
  <devices file="hardware/VFT/left_arm_virtual_strain.xml" />
  <devices file="hardware/VFT/left_leg_virtual_strain.xml" />
  <devices file="hardware/VFT/right_arm_virtual_strain.xml" />
  <devices file="hardware/VFT/right_leg_virtual_strain.xml" />
  <devices file="hardware/VFT/torso_virtual_strain.xml" />

  <!-- REAL ANALOG SENSORS -->
  <devices file="wrappers/FT/left_arm_FT_wrapper.xml" />
  <devices file="wrappers/FT/left_leg_FT_wrapper.xml" />
  <devices file="wrappers/FT/left_foot_FT_wrapper.xml" />
  <devices file="wrappers/FT/right_arm_FT_wrapper.xml" />
  <devices file="wrappers/FT/right_leg_FT_wrapper.xml" />
  <devices file="wrappers/FT/right_foot_FT_wrapper.xml" />
  <devices file="wrappers/MAIS/left_hand_mais_wrapper.xml" />
  <devices file="wrappers/MAIS/right_hand_mais_wrapper.xml" />
  <devices file="hardware/FT/left_arm_strain.xml" />
  <devices file="hardware/FT/left_leg_strain.xml" />
  <devices file="hardware/FT/left_foot_strain.xml" />
  <devices file="hardware/FT/right_arm_strain.xml" />
  <devices file="hardware/FT/right_leg_strain.xml" />
  <devices file="hardware/FT/right_foot_strain.xml" />
  <devices file="hardware/MAIS/left_hand_mais.xml" />
  <devices file="hardware/MAIS/right_hand_mais.xml" />

  <!-- SKIN -->
  <devices file="wrappers/skin/left_arm_skin_wrapper.xml" />
  <devices file="wrappers/skin/right_arm_skin_wrapper.xml" />
  <devices file="wrappers/skin/torso_skin_wrapper.xml" />
  <devices file="hardware/skin/left_arm.xml" />
  <devices file="hardware/skin/right_arm.xml" />
  <devices file="hardware/skin/torso.xml" />

  <!-- CALIBRATORS -->
  <devices file="calibrators/head_calib.xml" />
  <devices file="calibrators/torso_calib.xml" />
  <devices file="calibrators/right_leg_calib.xml" />
  <devices file="calibrators/left_leg_calib.xml" />
  <devices file="calibrators/left_arm_calib.xml" />
  <devices file="calibrators/right_arm_calib.xml" />
  <devices file="calibrators/left_hand_calib.xml" />
  <devices file="calibrators/right_hand_calib.xml" />
</robot>
```

ISTITUTO ITALIANO
DI TECNOLOGIA

iCub configuration

Example: icub-main\app\robots\iCubGenova01\icub_all.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<robot name="icub">
  <!-- cartesian -->
  <devices file="cartesian/left_arm_cartesian.xml" />
  <devices file="cartesian/right_arm_cartesian.xml" />

  <!-- motor controllers -->
  <devices file="wrappers/motorControl/left_arm_mc_wrapper.xml" />
  <devices file="wrappers/motorControl/right_arm_mc_wrapper.xml" />
  <devices file="wrappers/motorControl/left_leg_mc_wrapper.xml" />
  <devices file="wrappers/motorControl/right_leg_mc_wrapper.xml" />
  <devices file="wrappers/motorControl/head_mc_wrapper.xml" />
  <devices file="wrappers/motorControl/torso_mc_wrapper.xml" />
  <devices file="hardware/motorControl/icub_left_arm.xml" />
  <devices file="hardware/motorControl/icub_left_hand.xml" />
  <devices file="hardware/motorControl/icub_right_arm.xml" />
  <devices file="hardware/motorControl/icub_right_hand.xml" />
  <devices file="hardware/motorControl/icub_left_leg.xml" />
  <devices file="hardware/motorControl/icub_right_leg.xml" />
  <devices file="hardware/motorControl/icub_head.xml" />
  <devices file="hardware/motorControl/icub_torso.xml" />

  <!-- VIRTUAL ANALOG SENSORS -->
  <devices file="wrappers/VFT/left_arm_VFT_wrapper.xml" />
  <devices file="wrappers/VFT/left_leg_VFT_wrapper.xml" />
  <devices file="wrappers/VFT/right_arm_VFT_wrapper.xml" />
  <devices file="wrappers/VFT/right_leg_VFT_wrapper.xml" />
  <devices file="wrappers/VFT/torso_VFT_wrapper.xml" />
  <devices file="hardware/VFT/left_arm_virtual_strain.xml" />
  <devices file="hardware/VFT/left_leg_virtual_strain.xml" />
  <devices file="hardware/VFT/right_arm_virtual_strain.xml" />
  <devices file="hardware/VFT/right_leg_virtual_strain.xml" />
  <devices file="hardware/VFT/torso_virtual_strain.xml" />

  <!-- REAL ANALOG SENSORS -->
  <devices file="wrappers/FT/left_arm_FT_wrapper.xml" />
  <devices file="wrappers/FT/left_leg_FT_wrapper.xml" />
  <devices file="wrappers/FT/left_foot_FT_wrapper.xml" />
  <devices file="wrappers/FT/right_arm_FT_wrapper.xml" />
  <devices file="wrappers/FT/right_leg_FT_wrapper.xml" />
  <devices file="wrappers/FT/right_foot_FT_wrapper.xml" />
  <devices file="wrappers/MAIS/left_hand_mais_wrapper.xml" />
  <devices file="wrappers/MAIS/right_hand_mais_wrapper.xml" />
  <devices file="hardware/FT/left_arm_strain.xml" />
  <devices file="hardware/FT/left_leg_strain.xml" />
  <devices file="hardware/FT/left_foot_strain.xml" />
  <devices file="hardware/FT/right_arm_strain.xml" />
  <devices file="hardware/FT/right_leg_strain.xml" />
  <devices file="hardware/FT/right_foot_strain.xml" />
  <devices file="hardware/MAIS/left_hand_mais.xml" />
  <devices file="hardware/MAIS/right_hand_mais.xml" />

  <!-- SKIN -->
  <devices file="wrappers/skin/left_arm_skin_wrapper.xml" />
  <devices file="wrappers/skin/right_arm_skin_wrapper.xml" />
  <devices file="wrappers/skin/torso_skin_wrapper.xml" />
  <devices file="hardware/skin/left_arm.xml" />
  <devices file="hardware/skin/right_arm.xml" />
  <devices file="hardware/skin/torso.xml" />

  <!-- CALIBRATORS -->
  <devices file="calibrators/head_calib.xml" />
  <devices file="calibrators/torso_calib.xml" />
  <devices file="calibrators/right_leg_calib.xml" />
  <devices file="calibrators/left_leg_calib.xml" />
  <devices file="calibrators/left_arm_calib.xml" />
  <devices file="calibrators/right_arm_calib.xml" />
  <devices file="calibrators/left_hand_calib.xml" />
  <devices file="calibrators/right_hand_calib.xml" />
</robot>
```

iCub configuration

Example: icub-main\app\robots\iCubGenova01\icub_all.xml

Disabling a part (left arm) …

```
<?xml version="1.0" encoding="UTF-8" ?>
<robot name="icub">
  <!-- cartesian -->
  <devices file="cartesian/left_arm_cartesian.xml" />
  <devices file="cartesian/right_arm_cartesian.xml" />

  <!-- motor controllers -->
  <devices file="wrappers/motorControl/left_arm_mc_wrapper.xml" />
  <devices file="wrappers/motorControl/right_arm_mc_wrapper.xml" />
  <devices file="wrappers/motorControl/left_leg_mc_wrapper.xml" />
  <devices file="wrappers/motorControl/right_leg_mc_wrapper.xml" />
  <devices file="wrappers/motorControl/head_mc_wrapper.xml" />
  <devices file="wrappers/motorControl/torso_mc_wrapper.xml" />
  <devices file="hardware/motorControl/icub_left_arm.xml" />
  <devices file="hardware/motorControl/icub_left_hand.xml" />
  <devices file="hardware/motorControl/icub_right_arm.xml" />
  <devices file="hardware/motorControl/icub_right_hand.xml" />
  <devices file="hardware/motorControl/icub_left_leg.xml" />
  <devices file="hardware/motorControl/icub_right_leg.xml" />
  <devices file="hardware/motorControl/icub_head.xml" />
  <devices file="hardware/motorControl/icub_torso.xml" />

  <!-- VIRTUAL ANALOG SENSORS -->
  <devices file="wrappers/VFT/left_arm_VFT_wrapper.xml" />
  <devices file="wrappers/VFT/left_leg_VFT_wrapper.xml" />
  <devices file="wrappers/VFT/right_arm_VFT_wrapper.xml" />
  <devices file="wrappers/VFT/right_leg_VFT_wrapper.xml" />
  <devices file="wrappers/VFT/torso_VFT_wrapper.xml" />
  <devices file="hardware/VFT/left_arm_virtual_strain.xml" />
  <devices file="hardware/VFT/left_leg_virtual_strain.xml" />
  <devices file="hardware/VFT/right_arm_virtual_strain.xml" />
  <devices file="hardware/VFT/right_leg_virtual_strain.xml" />
  <devices file="hardware/VFT/torso_virtual_strain.xml" />

  <!-- REAL ANALOG SENSORS -->
  <devices file="wrappers/FT/left_arm_FT_wrapper.xml" />
  <devices file="wrappers/FT/left_leg_FT_wrapper.xml" />
  <devices file="wrappers/FT/left_foot_FT_wrapper.xml" />
  <devices file="wrappers/FT/right_arm_FT_wrapper.xml" />
  <devices file="wrappers/FT/right_leg_FT_wrapper.xml" />
  <devices file="wrappers/FT/right_foot_FT_wrapper.xml" />
  <devices file="wrappers/MAIS/left_hand_mais_wrapper.xml" />
  <devices file="wrappers/MAIS/right_hand_mais_wrapper.xml" />
  <devices file="hardware/FT/left_arm_strain.xml" />
  <devices file="hardware/FT/left_leg_strain.xml" />
  <devices file="hardware/FT/left_foot_strain.xml" />
  <devices file="hardware/FT/right_arm_strain.xml" />
  <devices file="hardware/FT/right_leg_strain.xml" />
  <devices file="hardware/FT/right_foot_strain.xml" />
  <devices file="hardware/MAIS/left_hand_mais.xml" />
  <devices file="hardware/MAIS/right_hand_mais.xml" />

  <!-- SKIN -->
  <devices file="wrappers/skin/left_arm_skin_wrapper.xml" />
  <devices file="wrappers/skin/right_arm_skin_wrapper.xml" />
  <devices file="wrappers/skin/torso_skin_wrapper.xml" />
  <devices file="hardware/skin/left_arm.xml" />
  <devices file="hardware/skin/right_arm.xml" />
  <devices file="hardware/skin/torso.xml" />

  <!-- CALIBRATORS -->
  <devices file="calibrators/head_calib.xml" />
  <devices file="calibrators/torso_calib.xml" />
  <devices file="calibrators/right_leg_calib.xml" />
  <devices file="calibrators/left_leg_calib.xml" />
  <devices file="calibrators/left_arm_calib.xml" />
  <devices file="calibrators/right_arm_calib.xml" />
  <devices file="calibrators/left_hand_calib.xml" />
  <devices file="calibrators/right_hand_calib.xml" />
</robot>
```

ISTITUTO ITALIANO
DI TECNOLOGIA

iCub configuration

Example: icub-main\app\robots\iCubGenova01\wrappers\motorControl\left_arm_mc_wrapper.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<devices>
 <device name="left_arm_mc_wrapper" type="controlboardwrapper2">
  <param name="threadrate">  10  </param>
  <paramlist name="networks">
          <elem name="left_arm_joints">  0  7  0  7 </elem>
          <elem name="left_hand_joints"> 8 15  0  7 </elem>
  </paramlist>

  <param name="period"> 10                          </param>
  <param name="name">   icub/left_arm               </param>
  <param name="ports">  left_arm                    </param>
  <param name="joints"> 16                          </param>

  <action phase="startup" level="5" type="attach">
   <paramlist name="networks">
          <elem name="left_arm_joints">  left_arm_mc </elem>
          <elem name="left_hand_joints"> left_hand_mc </elem>
    </paramlist>
  </action>

  <action phase="shutdown" level="5" type="detach" />
        </device>
</devices>
```

- User typically does not need to edit files in the wrapper folder
- User may tune the configuration parameters of the robot editing files contained in the hardware folder

iCub configuration

Example: icub-main\app\robots\iCubGenova01\wrappers\motorControl\left_arm_mc_wrapper.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<devices>
 <device name="left_arm_mc_wrapper" type="controlboardwrapper2">
  <param name="threadrate">  10  </param>
  <paramlist name="networks">
          <elem name="left_arm_joints">  0  7  0  7 </elem>
          <elem name="left_hand_joints"> 8 15  0  7 </elem>
  </paramlist>

  <param name="period"> 10                          </param>
  <param name="name">   icub/left_arm               </param>
  <param name="ports">  left_arm                    </param>
  <param name="joints"> 16                          </param>

  <action phase="startup" level="5" type="attach">
   <paramlist name="networks">
            <elem name="left_arm_joints">  left_arm_mc </elem>
            <elem name="left_hand_joints"> left_hand_mc </elem>
    </paramlist>
  </action>

  <action phase="shutdown" level="5" type="detach" />
        </device>
</devices>
```

- User typically does not need to edit files in the wrapper folder (with the only exception of the period param)
- User may tune the configuration parameters of the robot editing files contained in the hardware folder

iCub configuration

Example: icub-main\app\robots\iCubGenova01\hardware\motorControl\left_arm_mc.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<devices>
  <device name="left_arm_mc" type="canmotioncontrol">
  <params file="general.xml" />

<group name="CAN">
<param name="canbusdevice"> sharedcan     </param>
<param name="physDevice">   cfw2can       </param>
<param name="CanAddresses"> 1 2 3 4       </param>
<param name="CanDeviceNum"> 1             </param>
<param name="NetworkId">    LEFT_ARM      </param>
<param name="CanMyAddress"> 0       10    </param>
<param name="CanPollingInterval">         </param>
<param name="CanTimeout">   500           </param>      1           1           1           1           1           1           1           </param>
<param name="CanTxTimeout"> 500           </param>
<param name="CanRxTimeout"> 500           </param>
<param name="broadcast_pos">              1
<param name="broadcast_pid">              1           1           1           1           0           0           0           0           </param>
<param name="broadcast_fault">            1           1           1           1           1           1           1           1           </param>
<param name="broadcast_current">          0           0           0           0           0           0           0           0           </param>
<param name="broadcast_canprint">         1           1           1           1           1           1           1           1           </param>
<param name="broadcast_vel_acc">          0           0           0           0           0           0           0           0           </param>
</group>

<group name="GENERAL">      8             </param>
<param name="Joints">

<param name="AxisMap">      0             1           2           3           4           5           6           7           </param>
<param name="Encoder">      -11.375       -11.375     -11.375     -11.375     -706.67     -978.46     -978.46     -2.83333    </param>
<param name="Zeros">        -185.5        -333        9.257819    -189        90          -24         -45         -554.118    </param>
<param name="TorqueId">     0x0C          0x0C        0x0C        0x0C        0x0C        0           0           0           </param>
<param name="TorqueChan">   0             1           2           3           4           0           0           0           </param>
<param name="TorqueMax">    8             8           8           8           2           2           2           2           </param>
</group>

<group name="VELOCITY">     8             8           8           8           8           8           8           8           </param>
<param name="Shifts">

<param name="Timeout">      100           100         100         100         100         100         100         100         </param>
</group>

<group name="LIMITS">       -95.5         0           -30         15          -90         -80         -20         0           </param>
<param name="Min">

<param name="Max">          10            160.8       75          106         90          25          25          60          </param>
<param name="Currents">     7000          7000        7000        7000        500         800         800         800         </param>
</group>

<group name="POS_PIDS">     32000         32000       10000       32000       200         100         100         200         </param>
<param name="kp">

<param name="kd">           50            50          0           20          1000        100         100         200         </param>
<param name="ki">           60            60          10          60          1           2           2           1           </param>
<param name="maxPwm">       800           800         800         800         1333        1333        1333        1333        </param>
<param name="maxInt">       800           800         800         800         1333        1333        1333        1333        </param>
<param name="shift">        13            13          13          13          6           6           6           4           </param>
<param name="ko">           0             0           0           0           0           0           0           0           </param>
<param name="stictionUp">   0             0           0           0           0           0           0           0           </param>
<param name="stictionDwn">  0             0           0           0           0           0           0           0           </param>
</group>

  </device>
</devices>
```

iCub configuration

Example: icub-main\app\robots\iCubGenova01\hardware\motorControl\left_arm_mc.xml

In the robot configuration files values, parameters are often expressed in hardware units (no floating point support in the firmware):

- Encoder conversion factor (from ticks to degrees):

```
<param name="Encoder">      -11.375      -11.375      -11.375      -11.375      -706.67      -978.46      -978.46      -2.83333      </param>
```

4096 (ticks per revolution) / 360 (degrees) = 11.375 (typical AEA encoder)
254400 (ticks per revolution) 360 (degrees) = 706,666 (optical encoder)
(sign depends if the motor rotates cw or ccw respect to the encoder)

- The PID gains:

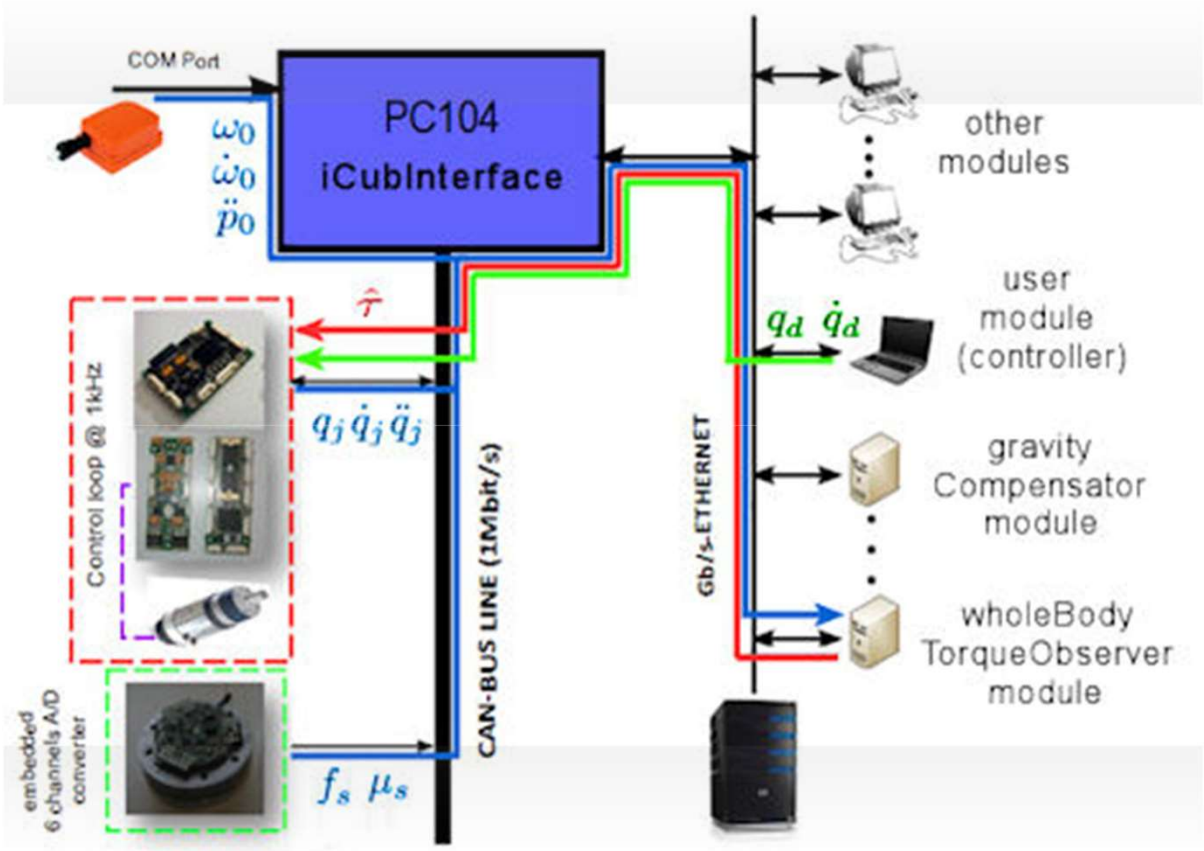
```
<group name="POS_PIDS">
<param name="kp">           32000        32000        10000        32000        200          100          100          200          </param>
<param name="kd">           50           50           0            20           1000         100          100          200          </param>
<param name="ki">           60           60           10           60           1            2            2            1            </param>
<param name="maxPwm">       800          800          800          800          1333         1333         1333         1333         </param>
<param name="maxInt">       800          800          800          800          1333         1333         1333         1333         </param>
<param name="shift">        13           13           13           13           6            6            6            4            </param>
<param name="ko">           0            0            0            0            0            0            0            0            </param>
<param name="stictionUp">   0            0            0            0            0            0            0            0            </param>
<param name="stictionDwn">  0            0            0            0            0            0            0            0            </param>
```

PWM = 1333 units correspond to the maximum supply voltage (typ: ~40v)

KP = 32000* 2^{-13} PWMs/encTICK = 0.256 PWMs/encTick
= 0.256*40/1333 V/encTick = 0.00768 V/encTick
= 0.00768*4096/360 V/deg = 0.0874 V/deg

ISTITUTO ITALIANO
DI TECNOLOGIA

iCub joint torque control: an example



1. Encoders and F/T data are broadcasted by motor control boards and sensors on the CAN bus.
2. robotInterfaces dispatches the data through the network of PCs. User modules can read data using YARP interfaces.
3. The software module WholeBodyDynamics estimates iCub joint torques (using the dynamic model of the robot) and send them back to robotInterface.
4. RobotInterfaces simulates a virtual analog sensor broadcasting on the CANbus force/torque data messages (SEE virtual analog sensor wrapper .xml file)
5. The motor control board receives the virtual analog sensor data. It can now close a torque control loop a t joint level.