EMARO

UNIVERSITÀ DEGLI STUDI DI GENOVA

TheEngineRoom

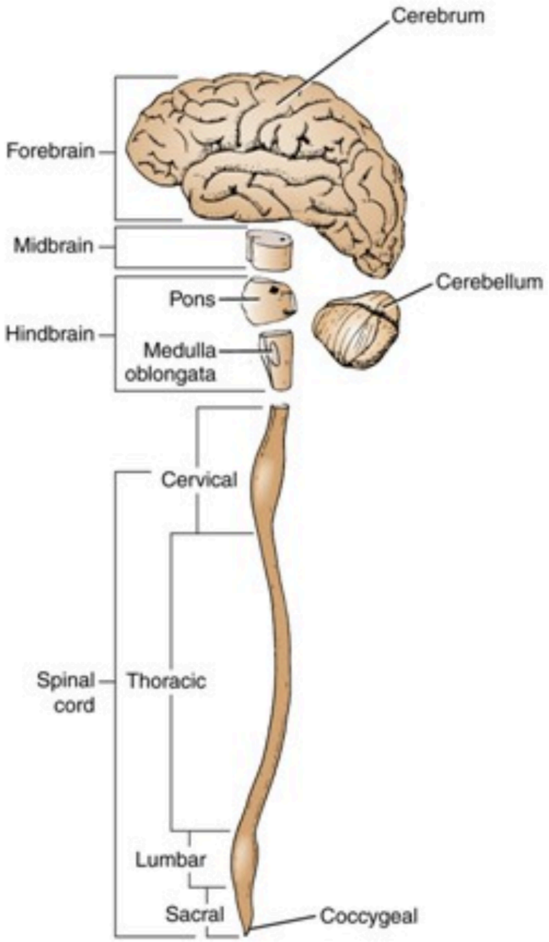
MiRo

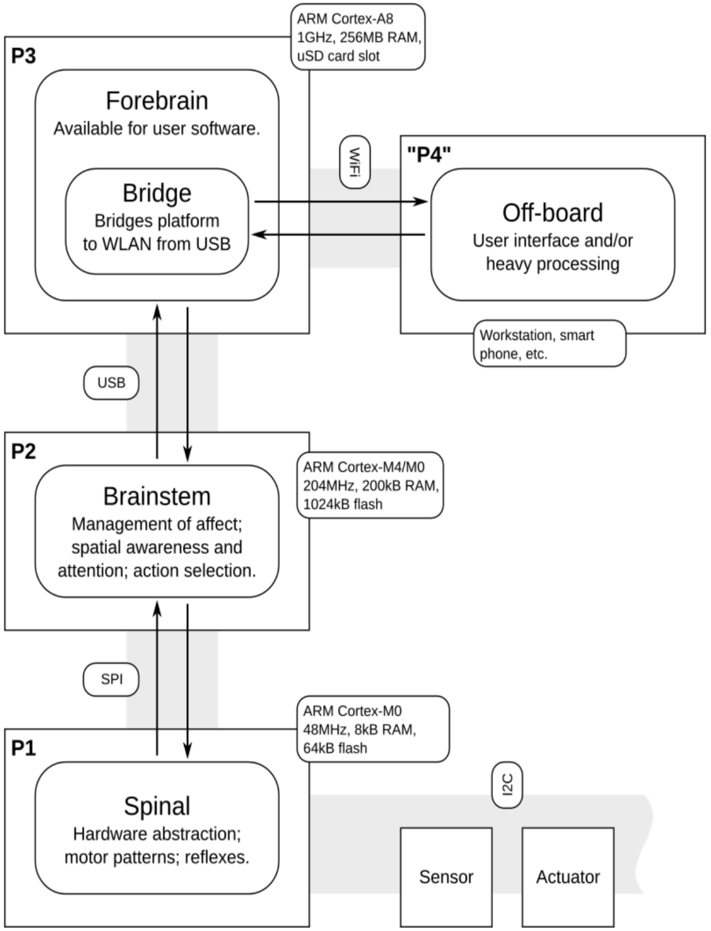
bioMimetic Robot

MiRo

- Companion robot
- Human-Robot interaction (HRI)
- Robot assisted therapy
- Biomimetic and brain-based robotics

Note





ARM Cortex-A8
1GHz, 256MB RAM,
uSD card slot

P3

Forebrain
Available for user software.

Bridge
Bridges platform
to WLAN from USB

WiFi

"P4"

Off-board
User interface and/or
heavy processing

Workstation, smart
phone, etc.

USB

P2

Brainstem
Management of affect;
spatial awareness and
attention; action selection.

ARM Cortex-M4/M0
204MHz, 200kB RAM,
1024kB flash

SPI

ARM Cortex-M0
48MHz, 8kB RAM,
64kB flash

P1

Spinal
Hardware abstraction;
motor patterns; reflexes.

I2C

Sensor

Actuator

Setup for MiRo

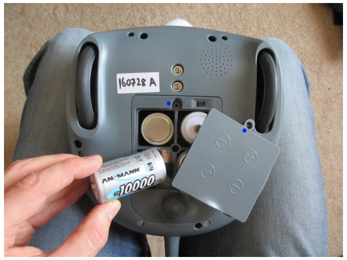
About MiRo: http://consequentialrobotics.com/miro/

Note

- The following slides are a quick summary of what is described in detail here: https://consequential.bitbucket.io/
- Specifically the ‘developer manual’: https://consequential.bitbucket.io/Developer.html

Critical points ⚠

HARDWARE



- While removing or inserting batteries. Place MiRo as can be seen in the image on the left.
- Quick look at this page (related to basic safety): https://consequential.bitbucket.io/Demonstrator_Commissioning_Before_You_Start.html
- Quick look at this page (if any technical doubts): https://consequential.bitbucket.io/Technical.html

SOFTWARE

- If you wish to use MIROsim (http://consequentialrobotics.com/mirosim/). It is better to have ubuntu as dualboot on your workstation rather than having ubuntu in virtual machine.
- Nonetheless, if you are using virtual machine and MIROsim does not work properly, try again by disabling 3D acceleration (In VirtualMachine settings)

Ros installation

Copy and paste in terminal, one by one, everything after the ‘$’ sign.

```
$ sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" >
/etc/apt/sources.list.d/ros-latest.list’
$ sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net:80 --recv-key
421C3665BD9FF1F717815A3895523BAEEB01FA116
$ sudo apt-get update
$ sudo apt-get install ros-kinetic-desktop-full
$ sudo rosdep init
$ rosdep update
$ echo "source /opt/ros/kinetic/setup.bash" >> ~/.bashrc
$ source ~/.bashrc
$ sudo apt-get install python-rosinstall python-rosinstall-generator python-wstool build-essential
$ rosversion ros
```

MiRo devkit and android app

- Download from: http://labs.consequentialrobotics.com/miro/mdk/
  - The MDK (Complete Edition) onto your workstation
  - The MIROapp onto your Android mobile device directly
- Then, in your workstation:
  - Make a new directory called ‘lib’ in your ‘home’ directory
  - Extract the recently downloaded MDK (Complete Edition) in ‘lib’
  - In the terminal enter into the mdk directory and execute the following command:

```
$ ln -s ~/lib/mdk-180509 ~/mdk
```

Configuring MiRo

- In the smartphone: open the MIROapp and check its IP address
- In your workstation: open a new terminal window and type in:

  ```
  $ ssh root@<Put_Here_IP_Of_MIRO_Found_In_Android_App>
  ```
- MIRO's SSH entry password is:

  ```
  $ MIROOpen1
  ```
- Once inside MIRO's Terminal, type in:

  ```
  $ sudo nano ~/.profile
  ```
- Once inside .profile
  - Change ROS_MASTER_IP as the IP address of your workstation. To check the IP address of your workstation open a new terminal window and type in:

    ```
    $ ifconfig
    ```

Testing sample codes

(MIRO is going to move, be careful if it is on the table) ⚠

- To run sample python code
  - Proceed as shown in the link:

    https://consequential.bitbucket.io/Developer_Examples_Python_Command-line_Client.html
- To run sample C++ code
  - Proceed as shown in the link:

    https://consequential.bitbucket.io/Developer_Examples_C++_Command-line_Client.html
- To run sample python code (GUI for sensors and actuators of MIRO):
  - Proceed as shown in the link:

    https://consequential.bitbucket.io/Developer_Examples_Python_GUI_Client.html

LAST BUT NOT THE LEAST

Take a look at Application Domains for Inspiration.

http://consequentialrobotics.com/domains/

CONFIGURING THE WORKSTATION: TO HAVE STATIC IP

Replace '<instruction_stated_inside>' entirely with the instruction stated inside.

In your workstation:

- **In (Network Preferences/TCP-IP), change (configure IPV4) to 'Manual' from 'DHCP'**
- **Change IPV4 Address to 130.251.13.<select_any_number_between_87_and_94>**
- **In (Network Preferences/DNS), add 130.251.1.4 and 8.8.8.8**
- **If using a virtual machine with ubuntu, all previous steps must be done on the main OS and after which you must go into the network settings of virtual machine and select "attached to" as 'Bridged Adapter'**

CONTINUED…

Copy and paste in terminal, everything after the ‘$’ sign.

Replace ‘<instruction_stated_inside>’ entirely with the instruction stated inside.

- **In your workstation (ubuntu):**

```
$ ifconfig
```

- **Check the IP address next to inet addr for enp___**
- **Copy that IP address (It is now the static IP of your workstation)**

```
$ sudo nano ~/.bashrc
```

- **Copy and paste in .bashrc the lines below and modify '<>' parts**

```
# cofiguration
export MIRO_PATH_MDK=~/mdk
export ROS_IP=<Put_Here_StaticIP_Of_WorkStation>
export ROS_MASTER_URI=http://localhost:11311

# make our custom messages available to ROS/python
export ROS_PACKAGE_PATH=$MIRO_PATH_MDK/share:$ROS_PACKAGE_PATH
export PYTHONPATH=$MIRO_PATH_MDK/share:$PYTHONPATH

# usual Gazebo setup
source /usr/share/gazebo/setup.sh

# announce MIRO resources to Gazebo
export GAZEBO_RESOURCE_PATH=$MIRO_PATH_MDK/share:${GAZEBO_RESOURCE_PATH}
```

- **Finally**

```
$ roscore
```