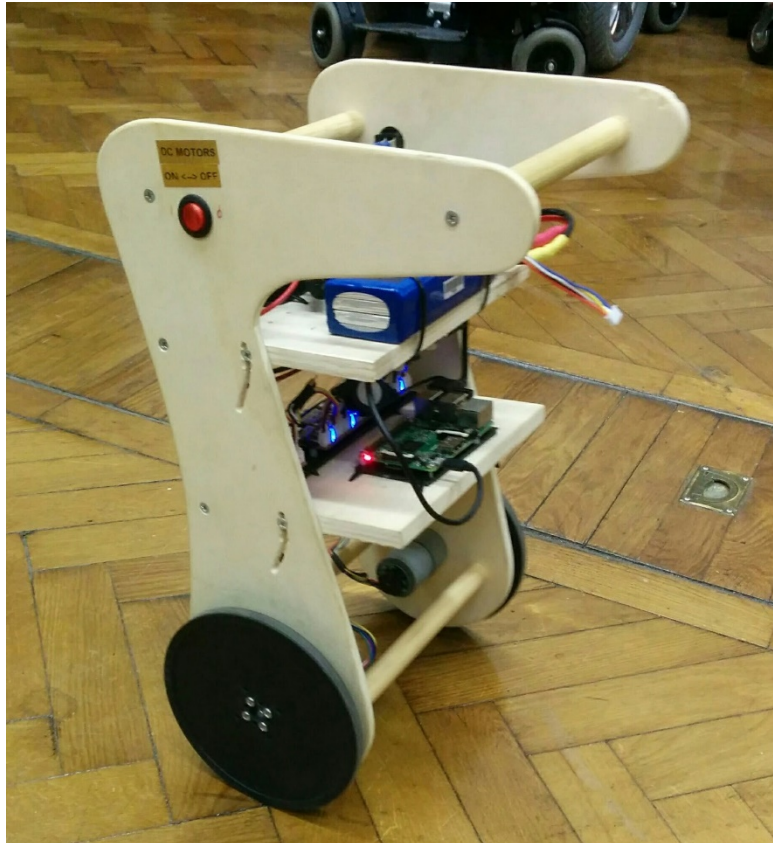


# ROBOT BILANCIANTE “PINGUIGLIO”

Manuale d’uso a cura di:

Lattarulo Marco [marco.lattarulo@mail.polimi.it](mailto:marco.lattarulo@mail.polimi.it)

Persiani Michele [michele.persiani@mail.polimi.it](mailto:michele.persiani@mail.polimi.it)



## 1. Descrizione del robot

Il robot bilanciante Pinguiglio è un piccolo robot mobile teleoperato e dotato di due ruote, sulle quali rimane in equilibrio. Il Pinguiglio è pilotabile mediante un joystick, da una distanza massima di 10/15 metri. E' inoltre possibile visualizzare su uno schermo remoto il video catturato da un ASUS xTion montato sulla testa del Pinguiglio.

## 2. Struttura del robot e configurazione hardware

- Struttura

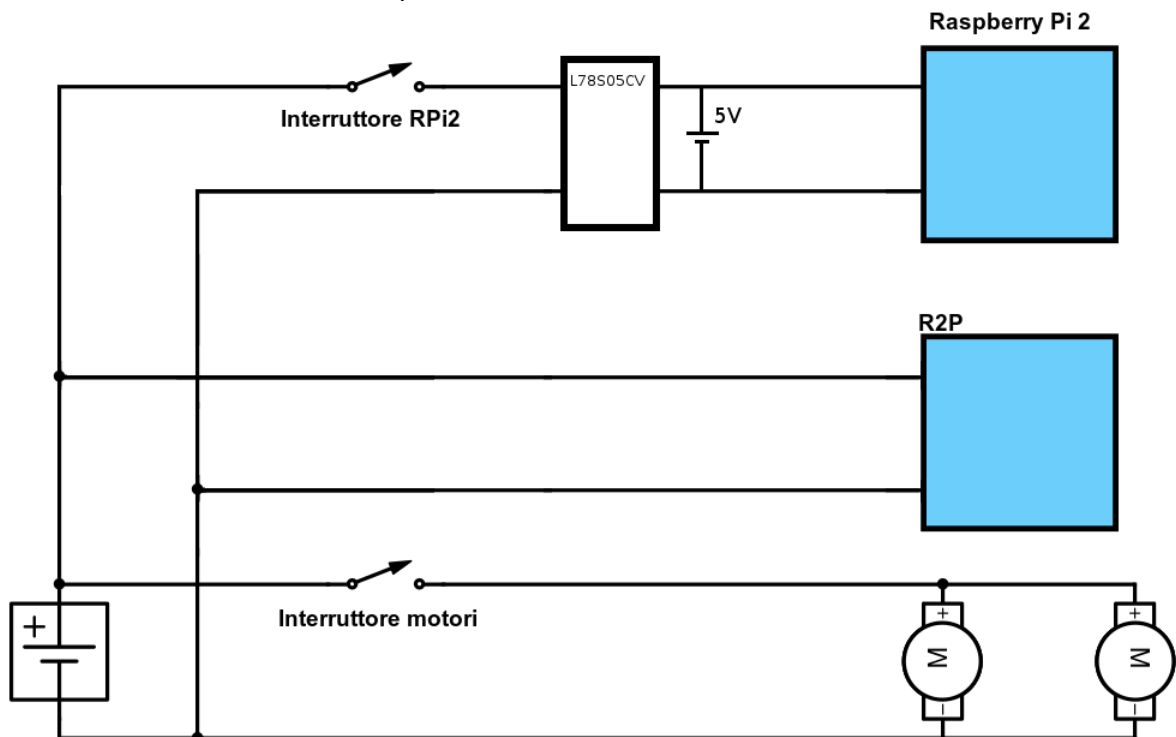
La struttura del robot è costituita da uno scheletro in legno con due piani orizzontali sovrapposti per l'alloggiamento dell'elettronica e dei sensori. I due motori sono direttamente collegati alle ruote nella parte inferiore dello scheletro. Visto di profilo, il

robot assume la forma di un pinguiglio (ovvero, in base al senso di marcia, sembra un pinguino o un coniglio), con i due interruttori rossi come occhi.

- Circuito elettrico

Ci sono tre diversi circuiti di alimentazione, due dei quali attivabili da interruttori dedicati: un interruttore accende e spegne i motori, l'altro interruttore accende e spegne il Raspberry Pi 2, mentre i moduli R2P (collegati l'uno con l'altro in serie) sono direttamente collegati all'alimentazione. E' presente un regolatore di tensione L78S05CV per fornire al Raspberry Pi 2 i 5V e 2A di cui ha bisogno

Schematica del circuito di potenza:



**N.B.:** Il circuito di alimentazione dei motori passa per un apposito bridge sui due moduli R2P dedicati ai motori.

- Alimentazione

L'alimentazione elettrica dell'intero robot è fornita da una batteria LiPo da 4500mAh (14,8V), mentre lo xTion è alimentato tramite una delle porte USB del Raspberry Pi 2.

- Elaborazione

L'elaborazione è affidata ad un Raspberry Pi 2 sul quale sono in esecuzione tutti i nodi ROS. Nello slot di memoria del Raspberry Pi 2 è presente una microSD da 16GB. Sono presenti inoltre 5 moduli R2P.

- Sensori

Oltre all'unità di misura inerziale presente su uno dei moduli R2P, è presente un ASUS xTion Pro.

- Motori  
Sono presenti due motori DC.
- Periferiche esterne  
E' presente un adattatore Wi-Fi USB per il Raspberry Pi 2 e un Joystick Logitech con ricevitore Bluetooth collegato anch'esso al Raspberry Pi

### 3. R2P (aka Nova)

Il cuore del Pinguiglio sono una serie moduli R2P, che implementando un controllore PID gli permettono di stare in equilibrio e muoversi, inseguendo un riferimento di velocità.

I moduli impiegati sono:

- 2x Modulo motore DC
- 1x Modulo IMU
- 1x Modulo sensori
- 1x Modulo Gateway

R2P comunica con ROS mediante il modulo gateway, che una volta connesso via USB al Raspberry Pi 2, può scambiare messaggi tramite una versione modificata (per R2P) del nodo ROS `rosserial_python`, pubblicando i topic che permettono il tuning dei moduli e la lettura dei sensori della IMU. In particolare è possibile modificare i parametri del controllore PID che tiene in equilibrio il robot, pubblicando opportuni messaggi sui due topic ROS dedicati: `/balcfg` e `/velcfg` (i.e.: `rostopic pub -r 1 /velcfg r2p_msgs/PidParameters 0 0 0` oppure `rostopic pub -r 1 /balcfg r2p_msgs/PidParameters 250 0 0`). Questi topic accettano messaggi contenenti i tre valori  $K_p$ ,  $K_i$ , e  $K_d$ .

Non dovrebbe comunque essere necessario modificare questi valori.

E' sempre possibile verificare il tipo di messaggi pubblicati sui topic mediante il comando **`rostopic info $topic_name`**

### 4. Configurazione Raspberry Pi 2

Il sistema operativo in esecuzione nel Raspberry Pi 2 è una versione di Ubuntu 14.04.2 Trusty opportunamente adattata al processore ARM v7 presente, disponibile a questo link: <https://wiki.ubuntu.com/ARM/RaspberryPi> (con download e istruzioni dettagliate per una corretta installazione). Il sistema operativo viene avviato in modalità grafica con il desktop environment Xubuntu.

Il Raspberry Pi 2 è configurato in modo da collegarsi automaticamente alla rete AIRLab2, permettendo l'accesso tramite SSH con username: *ubuntu* e password: *ubuntu*.

E' necessario trovare l'indirizzo IP del Raspberry usando un qualsiasi tool di IP scanning dal PC (i.e. Angry IP Scanner). L'hostname del Raspberry Pi 2 è *ubuntu.local* (o *ubuntu.lan*).

In alternativa alla Wi-Fi è possibile collegare Raspberry e PC alla stessa rete mediante cavi Ethernet.

Al momento non è possibile usare il Raspberry Pi 2 tramite desktop remoto. Supponiamo che ciò sia dovuto a mancanze nella versione di Ubuntu per ARM v7 utilizzata.

## 5. Nodi ROS

Il framework di sviluppo utilizzato per la programmazione del robot è ROS (Robot Operating System), e l'intera logica del robot è distribuita tra i seguenti nodi:

- **rosserial\_python serial\_node.py:** (versione modificata da Martino Migliavacca per R2P - aggiunge solo la classe astratta r2pHardware al nodo standard) E' usato per comunicare con i moduli R2P.
- **joy joy\_node:** E' usato come driver del joystick. Link al wiki ufficiale (compreso di tutorial): <http://wiki.ros.org/joy>
- **pinguiglio pinguiglio\_node:** E' un nodo creato da noi, che si sottoscrive al topic `/joy` per filtrare e ripubblicare i comandi joystick sotto forma di messaggio `geometry_msgs/Twist` leggibile da R2P
- **openni2\_camera camera\_node:** Questo nodo contiene launch files per devices compatibili con le librerie OpenNI. In questo caso è stato usato per leggere il topic `\rgb\image\compressed` contenente le immagini provenienti dallo xTion. Link al wiki ufficiale: [http://wiki.ros.org/openni2\\_camera](http://wiki.ros.org/openni2_camera)

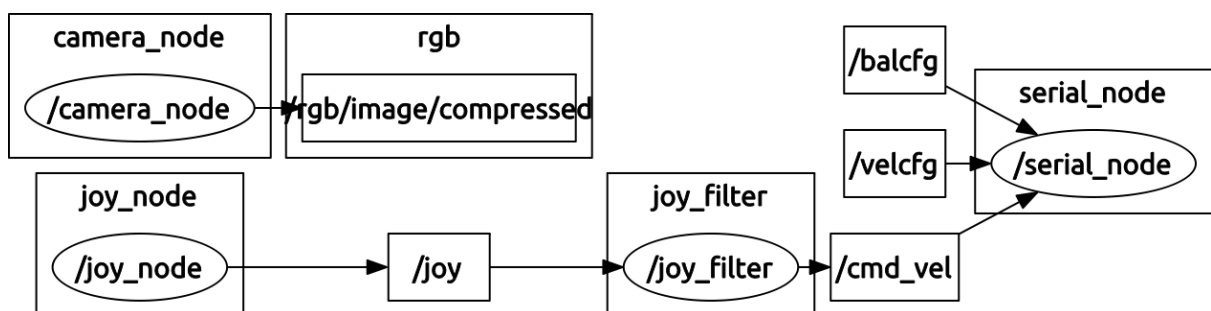
### Installazione

La versione di ROS presente è la Indigo, configurata secondo la guida ufficiale disponibile a questo link: <http://wiki.ros.org/indigo/Installation/UbuntuARM>

I packages **joy** e **openni2\_camera** sono quelli ufficiali di ROS e per installarli è sufficiente dare il comando **sudo apt-get install ros-indigo-openni2-camera ros-indigo-joy**

I packages **rosserial\_python** e **pinguiglio** sono due pacchetti personalizzati e quindi si trovano nel workspace catkin alla directory `/home/ubuntu/catkin_ws/src`. Nel caso fosse necessario reinstallare quest'ultimi il comando da dare è **cd ~/catkin\_ws && catkin\_make**

La seguente immagine catturata da `rqt_graph` mostra l'architettura dei nodi ROS sul Pinguiglio:



## 6. Istruzioni d'uso

### a) Avvio

Prima di collegare la batteria all' elettronica, è necessario assicurarsi che l'interruttore che alimenta i motori sia impostato su OFF, altrimenti questi non ricevendo un input coerente, inizieranno a girare alla massima velocità.

Per inizializzare correttamente il Pinguiglio, è necessario tenerlo in posizione di equilibrio mentre la batteria viene collegata, mantenendolo in tale posizione per 2-3 secondi e permettendo all'algoritmo di calibrazione automatica presente nel codice di controllo R2P di arrivare a completamento. Una volta finita la calibrazione è possibile accendere i motori: a questo punto il robot rimarrà automaticamente sulla posizione di equilibrio.

**N.B. E' di fondamentale importanza il corretto posizionamento dei pesi del robot, in particolare il posizionamento della batteria nell'apposito alloggiamento.**

L'accensione del Raspberry Pi 2 è possibile mediante l'apposito interruttore in qualsiasi momento successivo al collegamento della batteria.

Una volta effettuato l'accesso ad Ubuntu (vedi punto 4) è sufficiente dare da terminale il comando:

**roslaunch pinguiglio pinguiglio.launch** il quale si occuperà di lanciare tutti i nodi ROS necessari. Da questo momento in poi il sistema è pienamente funzionante.

### b) Controllo tramite joystick

Eseguita la sequenza d'avvio, il Pinguiglio è subito pilotabile mediante la levetta analogica destra del joystick Logitech. Per accendere il joystick è sufficiente premere il tasto Start. Gli altri tasti del joystick sono per ora inutilizzati, ma correttamente riconosciuti da ROS. NB: prima di sollevare il robot per qualsiasi motivo, assicurarsi che i motori siano scollegati

### c) Acquisizione video dallo xTion

E' possibile visualizzare in diretta su PC il video catturato dallo xTion del Pinguiglio, previa installazione e configurazione di ROS. Per fare ciò basta lanciare il nodo ROS rqt (o rviz) dal proprio PC e sottoscrivere al topic /rgb/image/compressed.

Questa operazione è possibile solo se PC e Pinguiglio sono sulla stessa rete WiFi. Inoltre prima di lanciare rqt sarà necessario dare i seguenti comandi sul PC utilizzato:

```
export ROS_MASTER_URI=http://IP Raspberry:11311  
export ROS_IP=IP_PC
```

### d) Reset

La sequenza corretta per il reset del robot è:

- I. Scollegare i motori mediante l'apposito interruttore
- II. Spegnerne il Raspberry Pi 2 mediante l'apposito interruttore
- III. Scollegare la batteria
- IV. Seguire le istruzioni alla voce: Avvio

## 7. Appendice

Topic attualmente in uso:

/ba1c1fg: Utilizzato per modificare i parametri del PID (control loop interno)

/velcfg: Utilizzato per modificare i parametri del PID (control loop esterno)

/cmd\_vel: Utilizzato per comunicare a R2P i riferimenti di velocità

/imu: Pubblica le velocità angolari e le accelerazioni lineari misurate dalla IMU, dopo essere state filtrate (topic /imu\_raw per le misurazioni raw)

/joy/rgb/image/compressed: Utilizzato per visualizzare l'immagine catturata dallo xTion

Altri topic:

**openni2\_camera camera\_node:**

/camera\_node/parameter\_descriptions

/camera\_node/parameter\_updates

/depth/camera\_info

/depth/image

/depth/image/compressed

/depth/image/compressed/parameter\_descriptions

/depth/image/compressed/parameter\_updates

/depth/image/compressedDepth

/depth/image/compressedDepth/parameter\_descriptions

/depth/image/compressedDepth/parameter\_updates

/depth/image/theora

/depth/image/theora/parameter\_descriptions

/depth/image/theora/parameter\_updates

/depth/image\_raw

/depth/image\_raw/compressed

/depth/image\_raw/compressed/parameter\_descriptions

/depth/image\_raw/compressed/parameter\_updates

/depth/image\_raw/compressedDepth

/depth/image\_raw/compressedDepth/parameter\_descriptions

/depth/image\_raw/compressedDepth/parameter\_updates

/depth/image\_raw/theora

/depth/image\_raw/theora/parameter\_descriptions

/depth/image\_raw/theora/parameter\_updates

/ir/camera\_info

/ir/image

/ir/image/compressed

/ir/image/compressed/parameter\_descriptions

/ir/image/compressed/parameter\_updates

/ir/image/compressedDepth

/ir/image/compressedDepth/parameter\_descriptions

/ir/image/compressedDepth/parameter\_updates

/ir/image/theora

/ir/image/theora/parameter\_descriptions

/ir/image/theora/parameter\_updates

/rgb/camera\_info

```
/rgb/image  
/rgb/image/compressed/parameter_descriptions  
/rgb/image/compressed/parameter_updates  
/rgb/image/compressedDepth  
/rgb/image/compressedDepth/parameter_descriptions  
/rgb/image/compressedDepth/parameter_updates  
/rgb/image/theora  
/rgb/image/theora/parameter_descriptions  
/rgb/image/theora/parameter_updates
```

**joy joy\_node:**

```
/diagnostics
```

**rosserial\_python serial\_node.py:**

```
/imu_raw
```

```
/odom
```

### Repository

Questo repository di GitHub contiene l'intera directory catkin\_ws, in modo tale da dare veloce accesso al codice del robot per tracciabilità e futuri aggiornamenti:

<https://github.com/marcowplm/pinguiglio.git>