

# Fundamentos en Robótica

Kits robóticos y robots paradigmáticos

# Temario

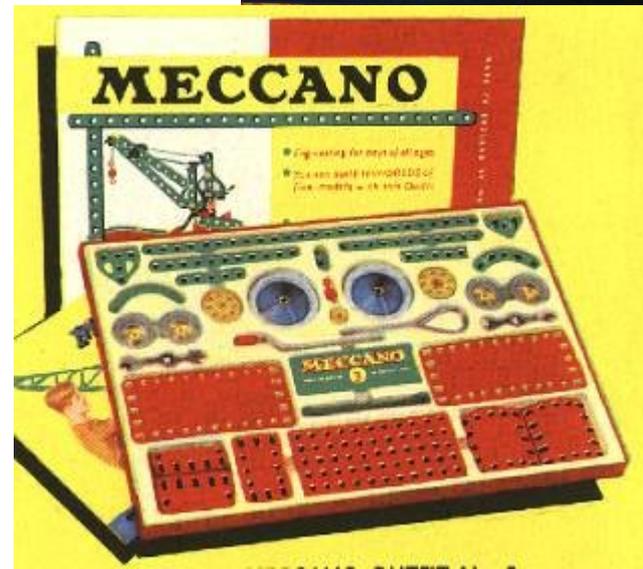
- Motivación
- Historia
- Mindstorms
- Butiá SAM (2.0)
- Robots paradigmáticos

# Motivación kits

- Estandarizan los componentes mecánicos y electrónicos para simplificar la construcción y programación de un robot.
- Utilizados fuertemente para estimular el interés por disciplinas científicas, las llamadas STEM: Science, Technology, Engineering and Mathematics.

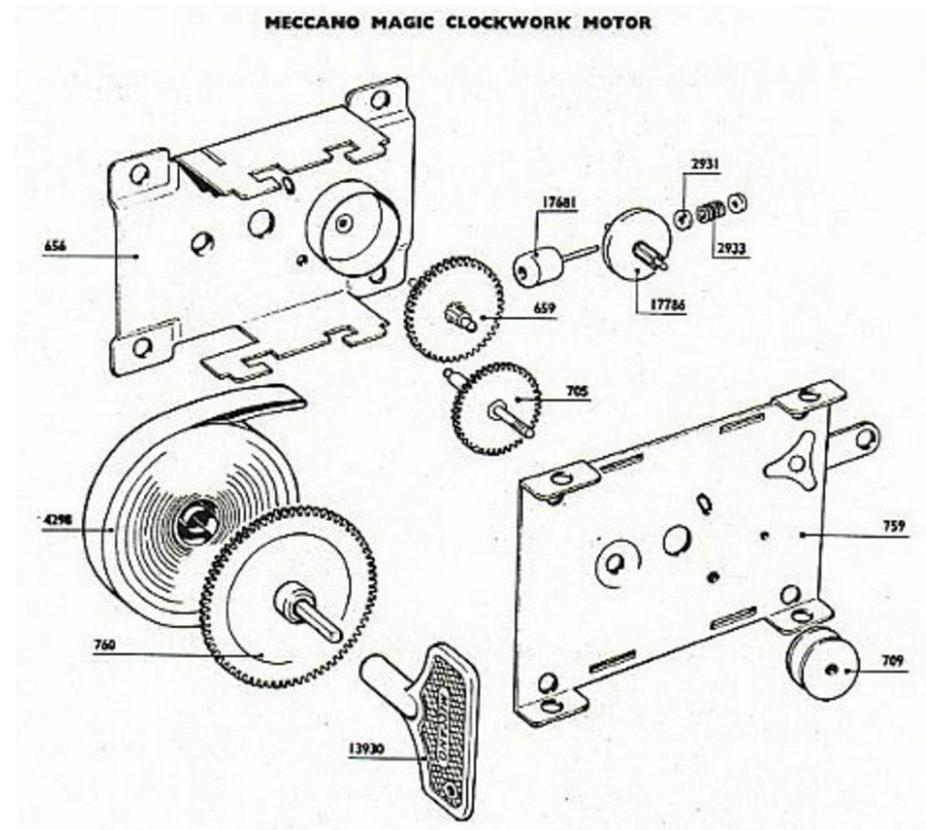
# Algunos ejemplos históricos <sup>(1/2)</sup>

- En 1901 Frank Hornby, inventa y patenta un nuevo juguete llamado "Mechanics Made Easy"



# Algunos ejemplos históricos<sup>(2/2)</sup>

- **Magic Clockwork motor M1** (introducido en 1935) implementando con un resorte que permite almacenar energía cuando se enrolla, y volverla a dar durante la expansión.



# Historia

LEGO clásico: 1949

Diseño actual: 1958

Plástico ABS desde 1963

LEGO Technic: 1982

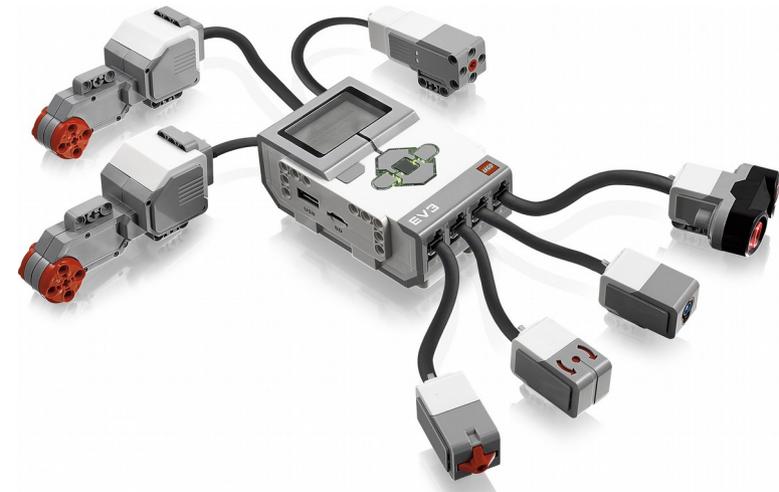
Nuevas piezas, motores,  
engranajes, luces...

Mindstorms RIS: 1998

Mindstorms NXT: 2006

Mindstorms NXT 2.0: 2009

Mindstorms EV3: 2013



# RIS, NXT & EV3

Ladrillos y piezas Technic

Controlador: RCX y NXT

Sensores

Comunicación

Motores

Software



# NXT: Controlador

Microcontrolador ARM7 32-bit, a 48MHz

- 256Kb FLASH, 64Kb RAM

Microcontrolador AVR 8-bit a 8MHz

- Control de los motores (modulación PWM)
- 4kB FLASH, 512B RAM

Bluetooth (SPP) + USB 2.0

3 Salidas

4 Entradas

LCD, botones, parlante

6 pilas AA



# NXT: Sensores<sup>(1/2)</sup>

Contacto (#2 en NXT 2.0)

- press/release/bump

LUZ (solo en NXT 1.0)

- Luz ambiente, reflejada
- monocromático

Sonido (solo en NXT 1.0)

- dB y dBA, en %.

Ultrasonido

- distancias 0-255cm+/- 3cm



# NXT: Sensores<sub>(2/2)</sub>

Color (solo en NXT 2.0)

Modo detección de color

Modo medición de luz ambiente, reflejada



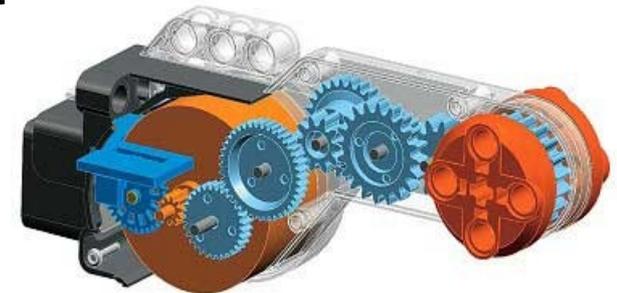
Existen muchos sensores más no incluidos en el kit básico

# NXT: Motores

Reducción integrada

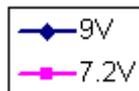
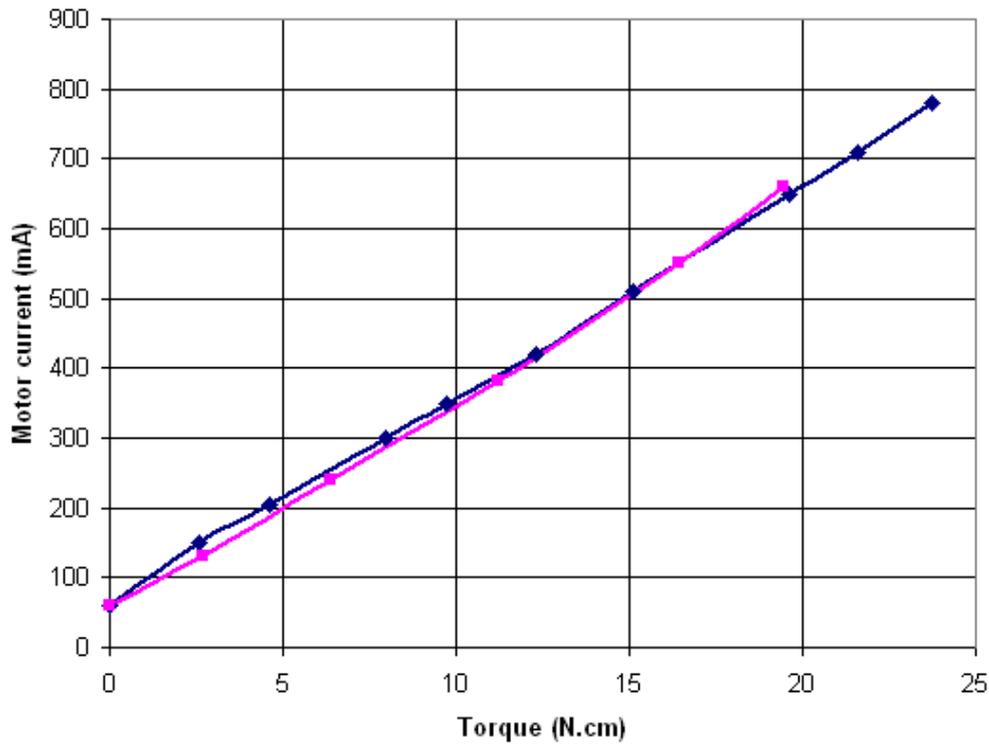
Servos

- Control:
  - Rotación continua
  - Angulo
- 0 a 165 RPM controlados de forma lineal @9V sin carga
- 0 a 130 RPM controlados de forma lineal @7.2(batería de NiMH) sin carga
- Medidor de rotación integrado (1° precisión)

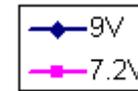
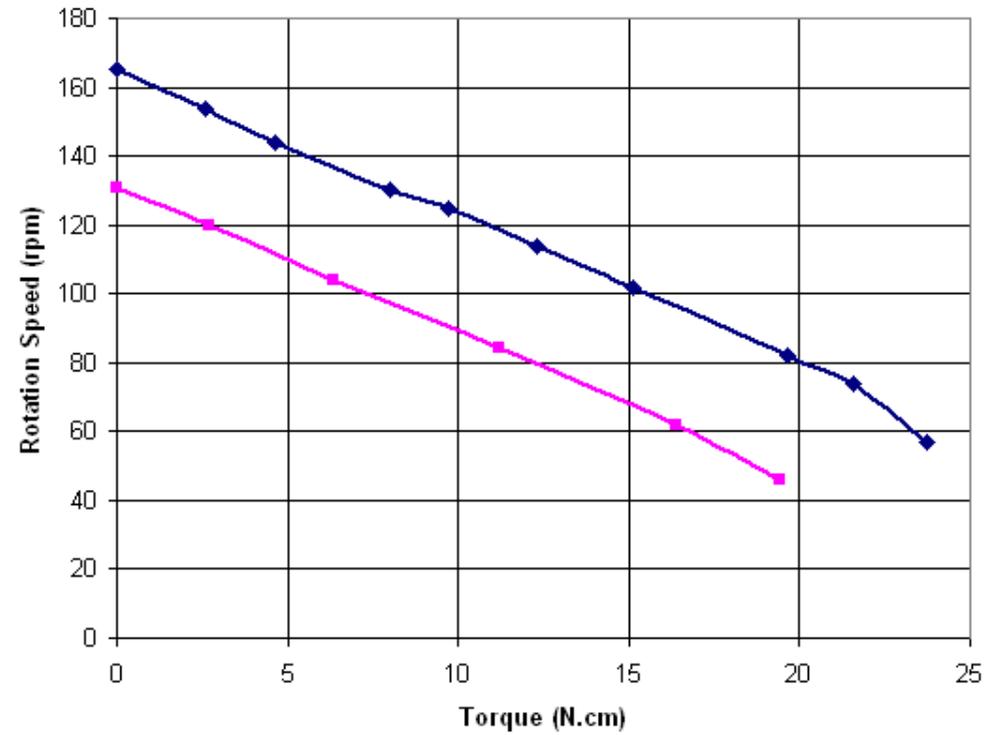


# NXT: Motores

Motor current vs. Torque



Rotation Speed vs. Torque



# Programación

## Onboard o programación embebida

- RobotC
- LeJOS
- NXC/NBC

## Control Remoto

- ICommand (Java)
- NXT\_Python
- TurtleBots

# Uso NXT\_Python<sub>(1/4)</sub>

Ejemplo de uso de los motores:

```
#!/usr/bin/env python

# importamos las funciones de la API
import nxt.locator
from nxt.motor import *

# esta función trata de conectarse al brick
b = nxt.locator.find_one_brick()

# movemos el motor conectado en el puerto B del brick b
motor_izquierdo = Motor(b, PORT_B)

# con potencia 100 y que de una vuelta completa (360º)
motor_izquierdo.turn(100, 360)
```

Advertencia: verificar que estén instaladas las reglas udev

# Uso NXT\_Python<sub>(2/4)</sub>

Ejemplo de uso del sensor de contacto:

```
#!/usr/bin/env python

# importamos las funciones de la API
import nxt.locator
from nxt.sensor import *

# esta función trata de conectarse al brick
b = nxt.locator.find_one_brick()

# usaremos el botón conectado en el puerto 1 del brick
boton = Touch(b, PORT_1)

#obtenemos el valor del sensor
valor_boton = boton.get_sample()

# imprimimos en pantalla el valor
print 'Botón: ', valor_boton
```

# Uso NXT\_Python<sub>(3/4)</sub>

Ejemplo de uso del sensor de distancia:

## Sensor Ultrasónico

```
#!/usr/bin/env python
```

```
# importamos las funciones de la API
```

```
import nxt.locator  
from nxt.sensor import *
```

```
# esta función trata de conectarse al brick
```

```
b = nxt.locator.find_one_brick()
```

```
# usaremos el sensor ultrasónico conectado en el puerto 4 del brick
```

```
ultrasonico = Ultrasonic(b, PORT_4)
```

```
#obtenemos el valor del sensor
```

```
valor_ultrasonico = ultrasonico.get_sample()
```

```
# imprimimos en pantalla el valor
```

```
print 'Ultrasónico: ', valor_ultrasonico
```

# Use NXT\_Python<sub>(4/4)</sub>

```
#!/usr/bin/env python
import nxt.locator
from nxt.sensor import *
from nxt.motor import *

brick = nxt.locator.find_one_brick()

motor_power = 127
button = Touch(brick, PORT_1)
motorB = Motor(brick, PORT_B)
motorC = Motor(brick, PORT_C)

syncmotors = SynchronizedMotors(motorB, motorC, 0)
syncmotors.run(motor_power)
button_data = button.getSample()

while (button_data!=1):
    button_data = boton.getSample()

motor_power = 0
syncmotors.run(motor_power)
```

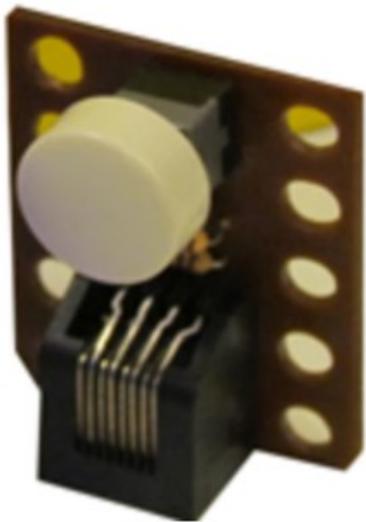
# Butiá SAM

- Plataforma que expande capacidad sensorial y de actuación de una computadora portátil
- Desarrollada en FING
- Licencia GNU/GPLv3



# Sensores

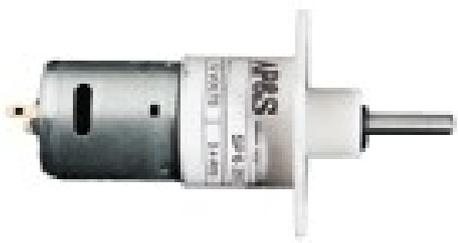
De izquierda a derecha: Contacto, Grises, Luz, Distancia



# Motores

## Motorreductores

Corriente continua, reducción.



## Dynamixel AX12

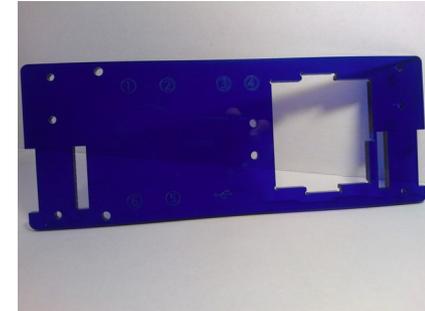
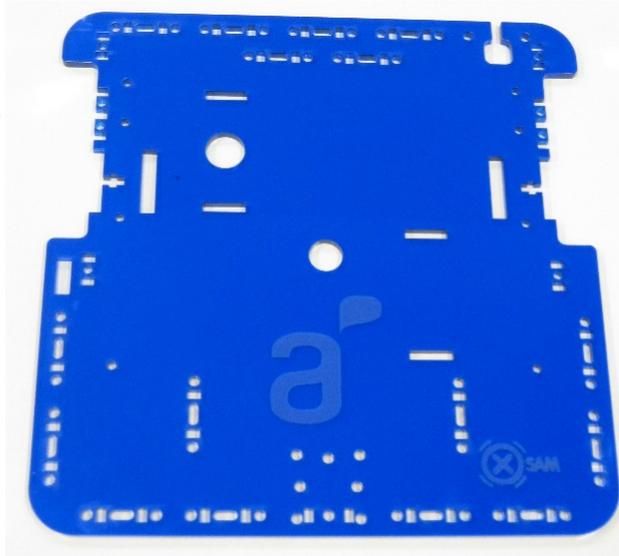


# Piezas

## Plataforma, chasis

Acrílico 6mm

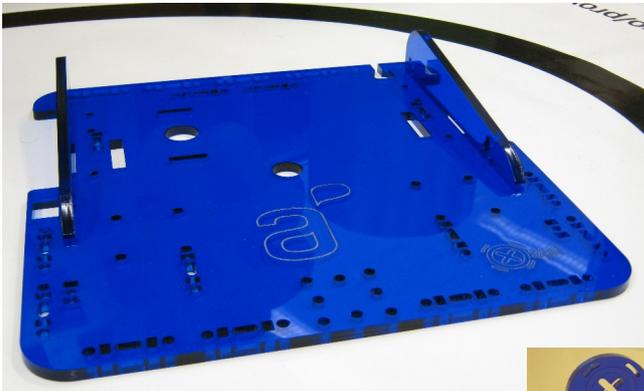
Otros: Madera, chapa, PVC



## Barandas

XO horizontal en la placa.

Otros más grandes sobre barandas.



# Ruedas

## Ruedas Motrices

Llanta de acrílico

Aro de goma

Tapas

Diámetro 82mm

Pisada 9mm



## Tamaño de la rueda

Diámetro - Velocidad

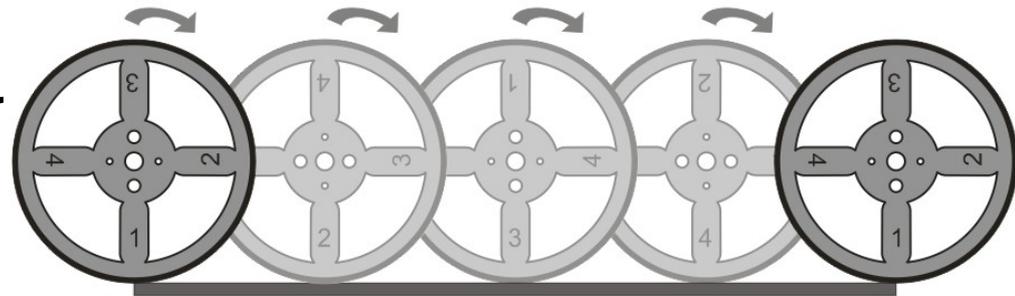
Diámetro: 82mm

Perímetro:  $2 \cdot \text{Pi} \cdot R$  (257,5mm)

Una vuelta recorre 0,26m

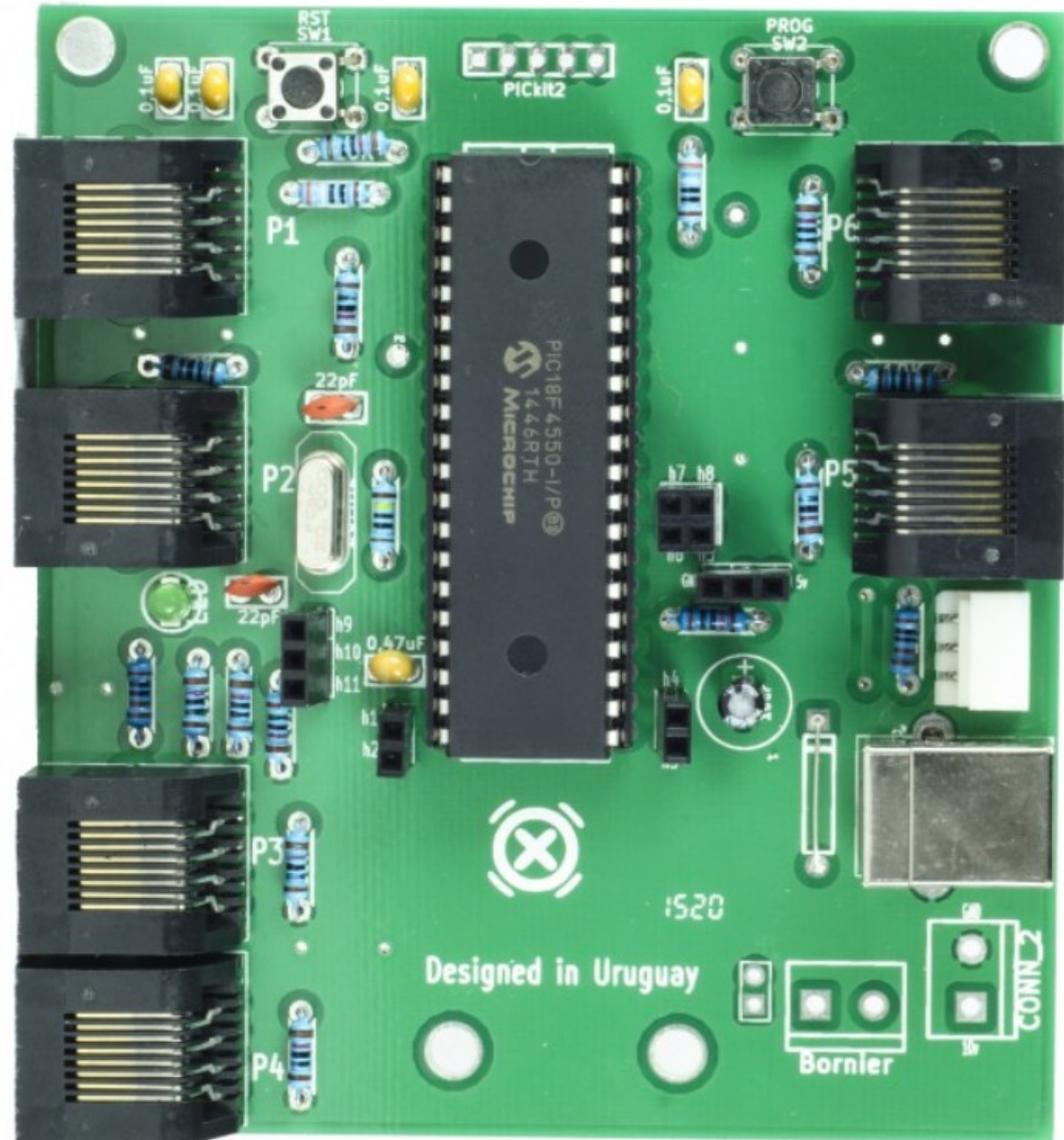
Velocidad del motor: 80RPM - Velocidad de Butiá: 20,8m/min

(0,35m/s)



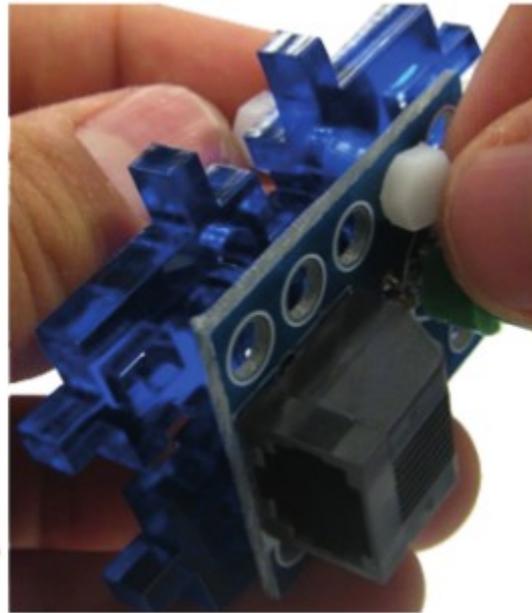
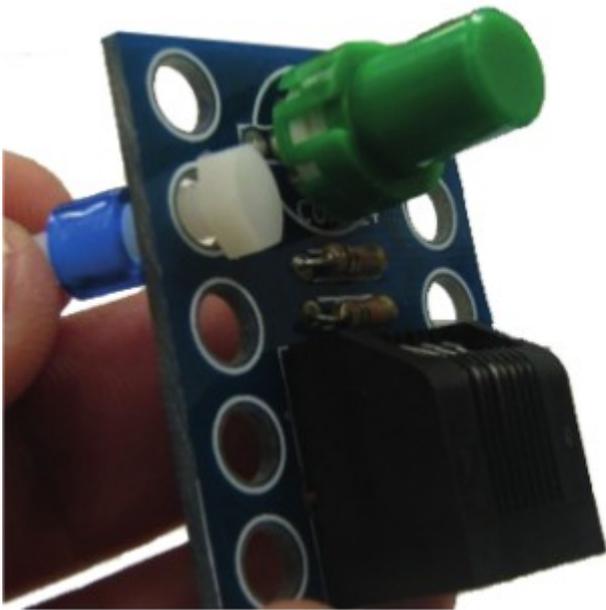
# USB4Butiá

- 6 puertos generales de E/S
- plug&play
- Expansión mediante hackpines
- Bus para controlar motores AX12
- Conexión USB



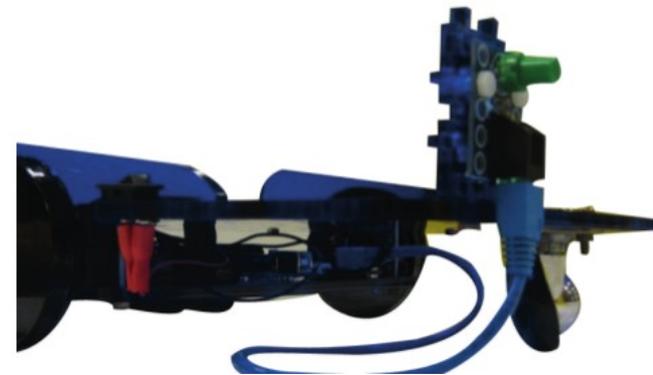
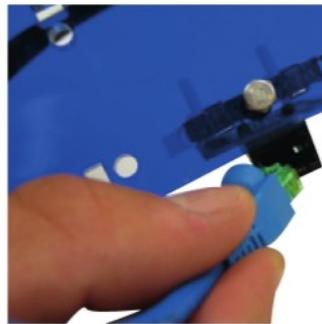
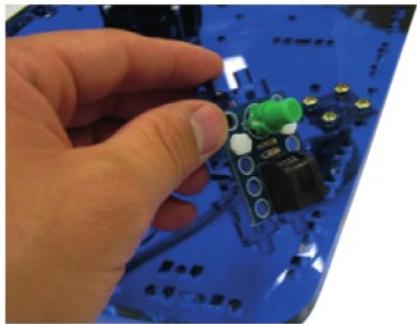
# Sistema de encastre <sub>(1/2)</sub>

Los sensores se montan sobre la ficha de encastre usando tornillos y tuercas



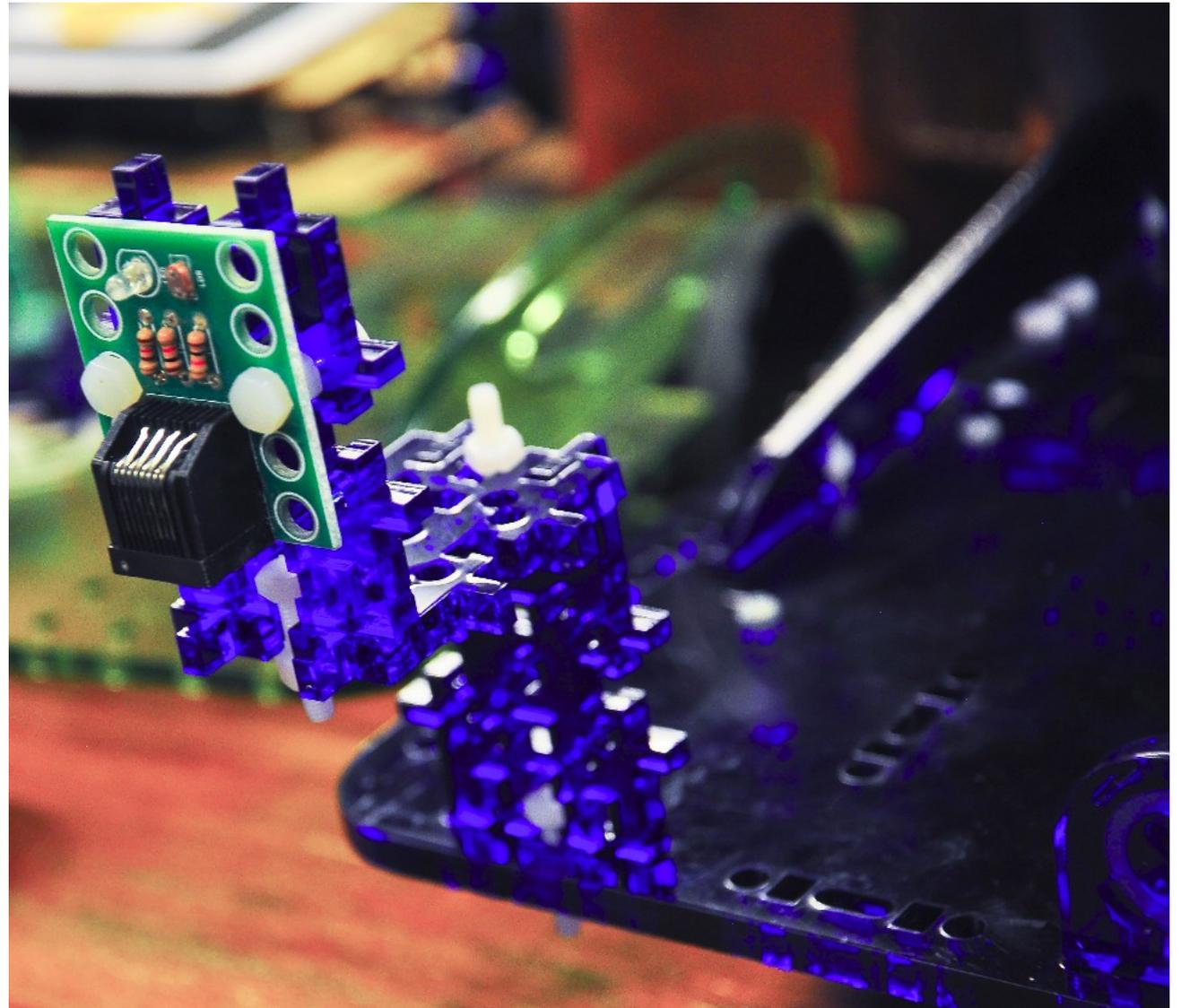
# Sistema de encastre (2/3)

Luego se monta sobre la plataforma, usando **solo un** tornillo y una tuerca, y se conecta el sensor a la placa USB4Butiá



# Sistema de encastre (3/3)

Es posible conectar varias fichas entre si, formando estructuras más complejas.



# Uso API USB4Butiá-pybot<sub>(1/3)</sub>

Ejemplo de uso de los motores:

---

```
#!/usr/bin/env python
#si no estoy en el directorio
import sys
sys.path.insert(0, '/home/olpc/Activities/TurtleBots.activity/plugins/butia')
#importamos las funciones de la API
from pybot import usb4butia
#genera una instancia de la placa USB4Butiá
robot = usb4butia.USB4Butia()

motor_power = 1023
#pone a girar el motor izquierdo (0), en sentido 1, a velocidad máxima
robot.setMotorSpeed(0, 1, motor_power)
```

# Uso API USB4Butiá-pybot<sup>(2/3)</sup>

Ejemplo de uso del sensor de distancia:

---

```
#!/usr/bin/env python
#si no estoy en el directorio
import sys
sys.path.insert(0, '/home/olpc/Activities/TurtleBots.activity/plugins/butia')
#importamos las funciones de la API
from pybot import usb4butia
#genera una instancia de la placa USB4Butiá
robot = usb4butia.USB4Butia()

#obtiene un valor de distancia del sensor conectado al puerto 1
distance = robot.getDistance(1)
```

# Uso API USB4Butiá-pybot<sup>(3/3)</sup>

```
#!/usr/bin/env python
#si no estoy en el directorio
import sys
sys.path.insert(0, '/home/olpc/Activities/TurtleBots.activity/plugins/butia')
from pybot import usb4butia
BUTTON_PORT = 1

robot = usb4butia.USB4Butia()
motor_power = 1023

robot.set2MotorSpeed(0, motor_power, 0, motor_power)
lectura_boton = robot.getButton(BUTTON_PORT)

while (lectura_boton!=1):
    lectura_boton = robot.getButton(BUTTON_PORT)

motor_power = 0
robot.set2MotorSpeed(0, motor_power, 0, motor_power)
```

# TurtleBots

- Pensada como herramienta inicial de programación
- Útil para prototipado rápido, permite validar ideas de software o mecánicas fácilmente.
- Comptible con Butiá y Lego NXT
- Disponible en: <https://www.fing.edu.uy/inco/proyectos/butia/files/package/>

# Husky UGV

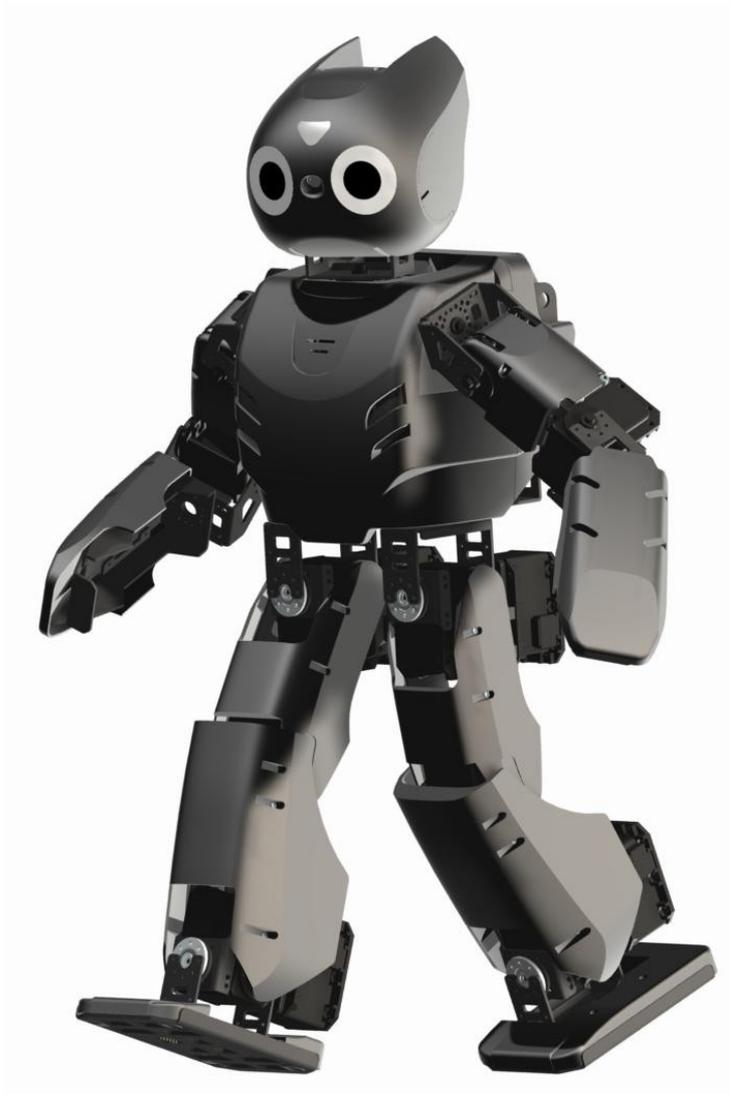
- Plataforma robótica de tamaño medio
- Permite que la carga útil del robot sea customizada
- Gran torque y agarre, útil para terrenos irregulares
- Útil como plataforma para proyectos de investigación



# Especificaciones técnicas

SIZE AND WEIGHT	
EXTERNAL DIMENSIONS (L x W x H)	990 x 670 x 390 mm (39 x 26.4 x 14.6 in)
INTERNAL DIMENSIONS	296 x 411 x 155 mm (11.7 x 16.2 x 6.1 in)
WEIGHT	50 kg (110 lb)
WHEELS	330 mm (13 in) Lug Tread
GROUND CLEARANCE	130 mm (5 in)
SPEED AND PERFORMANCE	
MAX. PAYLOAD	75 Kg (165 lb)
ALL-TERRAIN PAYLOAD	20 Kg (44 lb)
MAX SPEED	1.0 m/s (2.3 mph)
DRIVETRAIN / DRIVE POWER	4x4 Zero-Maintenance
MAX CLIMB GRADE	45° (100% Slope)
MAX TRAVERSAL GRADE	30° (58% Slope)
BATTERY AND POWER SYSTEM	
BATTERY CHEMISTRY	Sealed Lead Acid
CAPACITY	24 V, 20 Ah
RUNTIME - STANDBY	8 Hours
RUNTIME - NOMINAL USAGE	3 Hours
CHARGE TIME	4 Hours
USER POWER	5 V, 12 V, 24 V Fused at 5 A each. 192 W total available power (upgrade to 480 W optional).
INTERFACING AND COMMUNICATION	
CONTROL MODES	Direct voltage, wheel speed, and kinematic velocity.
FEEDBACK	Battery voltage, motor currents, wheel odometry, and control system output.
COMMUNICATION	RS232 @ 115200 baud
ENCODERS	Quadrature: 78,000 pulses/m
DRIVERS AND APIS	ROS, C++, and Python.
ENVIRONMENTAL	
OPERATING AMBIENT TEMPERATURE	-10 to 30 °C (14 to 86 °F)
STORAGE TEMPERATURE	-40 to 50 °C (-40 to 122 °F)
RATING	IP 44 (upgrade to IP 55 available)

# DARwin-OP



Altura: 45.5 cms

Peso: 2.9 kg

20 actuadores MX-28T (6 DOF leg × 2 + 3 DOF arm × 2 + 2 DOF neck) con engranajes metálicos

CPU : Intel Atom Processor N2600 (dual core, 1.6 GHz)

RAM : 4GB (DDR3 204-pin SO-DIMM module), user-replaceable

SSD : SSD / mSATA module (unlimited space), user-replaceable

OS : GNU/Linux y Windows

Programación: C++ embebido u otro lenguaje corriendo en un PC y usando API remota

Puertos accesibles por el usuario : 2 x USB2.0, mini HDMI, LAN (gigabit), mic/audio line in,out)

Sensores:

- \* Giroscopio de 3 ejes

- \* Accelerómetro de 3 ejes

- \* 2 Micrófonos

- \* 1 cámara USB

Alimentación: batería de Li-PO 11.1V 1000mAh. La batería tiene 30 minutos de autonomía.

Precio: 9600 USD

# DARwin-OP



# Aldebaran-Nao



Altura: 58 cm

Peso: 4.3 kg

Alimentación: Baterías de lithium, que proveen de 48.6 Wh

Autonomía: 90 minutos (uso activo)

Grados de libertad: 25

CPU: Intel Atom @ 1.6 GHz

Sistema operativo: NAOqi 2.0 (basado en GNU/Linux), permite instalar Windows o Mac

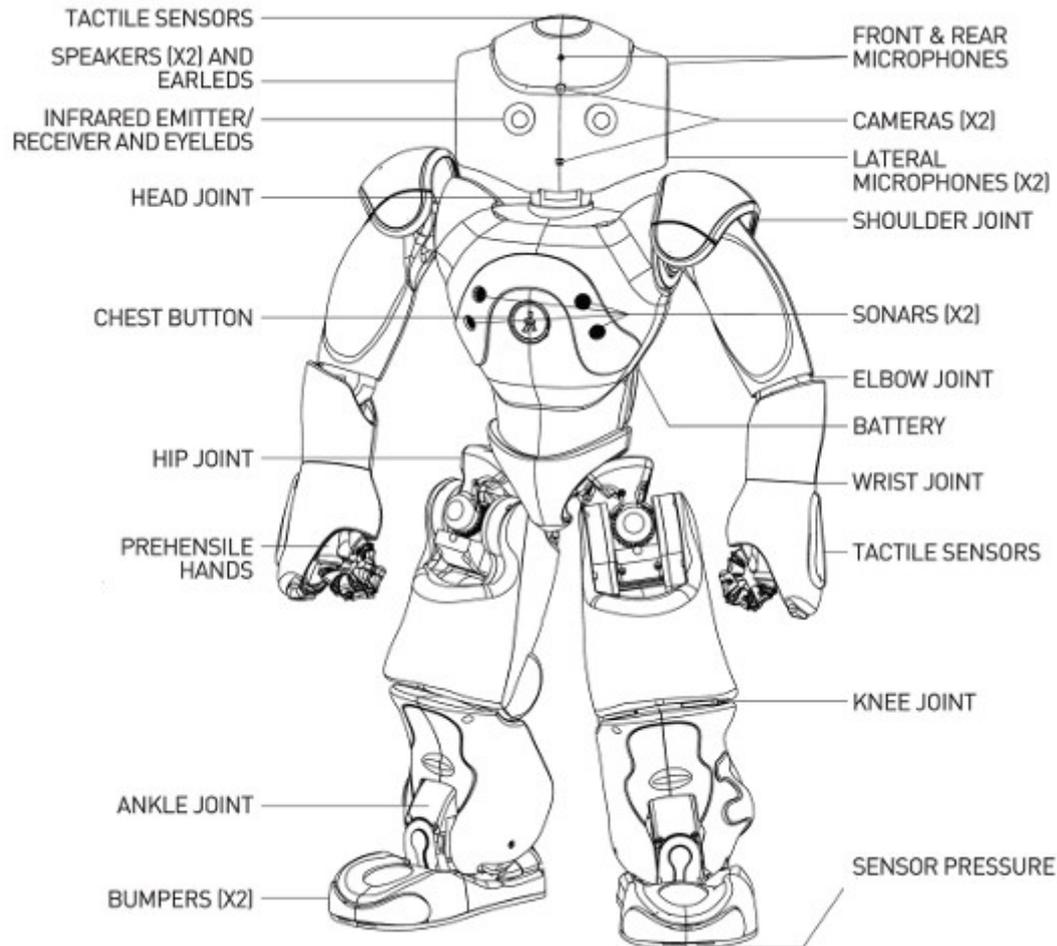
Lenguajes de Programación: C++, Python, Java, MATLAB, Urbi, C, Net

Sensors: Dos cámaras HD, 4 micrófonos, Sonar, dos emisores y receptores infrarrojos, inertial board, 9 sensores táctiles, 8 sensores de presión

Conectividad: Ethernet, Wi-Fi

Precio:: 10000 USD

# Aldebaran-Nao



# Khepera IV



Sistema Operativo: GNU/Linux

Processor: 800MHz ARM Cortex-A8 Processor

Microcontrolador adicional para el manejo de periféricos

RAM: 512 MB

Flash: 512 MB + 4GB para datos

Actuadores:

- 2 motores DC con caja de engranajes y encoders incrementales (147 pulso por mm recorrido por el robot)
- 3 LEDs programables RGB LED

Velocidad máxima: 1m/s

Sensores:

- 8 de proximidad infrarrojo rango hasta 25 cms,
- 4 infrarrojos apuntando al piso para (seguidor de obstáculos, evitar caídas)
- 5 distancia ultrasónico con rango 25cm a 2 metros,
- acelerómetro y giróscopo de 3 ejes
- 1 micrófono
- cámara integrada color (752×480 pixels, 30FPS)

Autonomía: ~7 hs.

Batería: 7.4V Lithium Polymer, 3400mAh

Ready for docking (Power input and I2C communication)

Conectividad: 1x USB 2.0 host (500mA), 1x USB 2.0 device, 802.11 b/g WiFi, Bluetooth 2.0 EDR

Diámetro: 14 cms

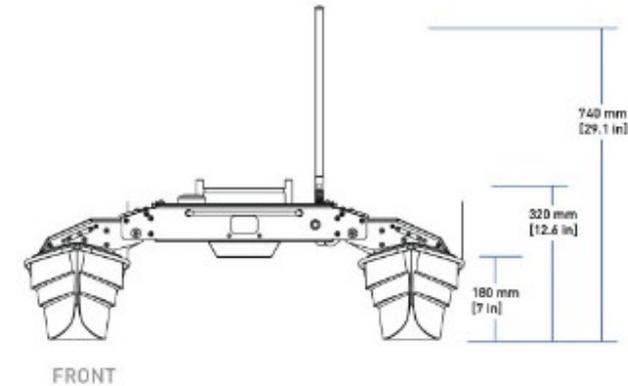
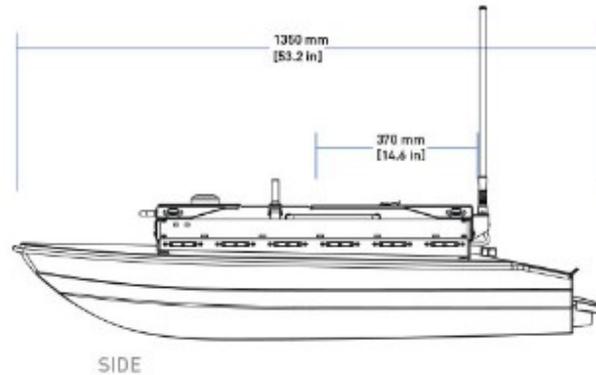
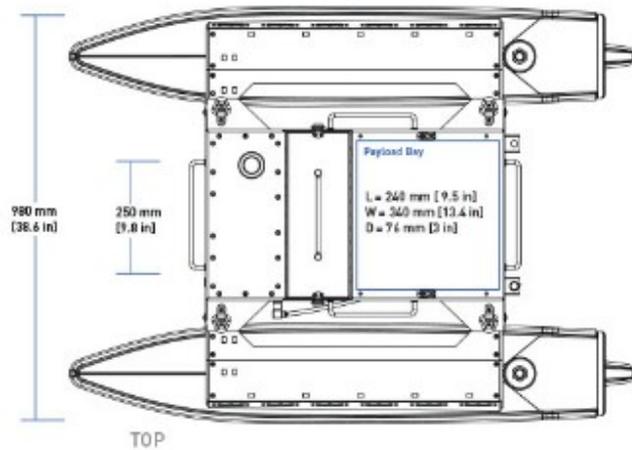
Altura: 5.8 cms

Peso: 540g

Carga máxima: ~ 2000 g

# Robot Heron

(UNMANNED SURFACE VESSEL)



<b>DIMENSIONS DEPLOYED</b> (L x W x H)	1350 x 980 x 320 mm (53.2 x 38.6 x 12.6 in)
<b>DIMENSIONS STORED</b> (L x W x H)	1350 x 560 x 330 mm (53.2 x 22 x 13 in)
<b>WEIGHT</b>	28 kg (62 lbs)
<b>RATED PAYLOAD</b>	10 kg (22 lbs)
<b>MAXIMUM SPEED</b>	1.7 m/s (3.3 kn)

<b>DRAUGHT</b>	120 mm (4.7 in)
<b>BATTERY PACK</b>	NiMH 14.4V 29Ah 2.5 hrs life
<b>DRIVE POWER</b>	70 W peak 40 N Thrust
<b>RECHARGE TIME</b>	8 hrs
<b>COMMUNICATION</b>	Ethernet, RS232

<b>PROPULSION</b>	Water Jet (Electric)
<b>ENVIRONMENTAL</b>	IP 65 - 10 / +30 °C
<b>SENSORS</b>	GPS, IMU
<b>DRIVERS/ APIs</b>	ROS, C++, MOOS-IvP
<b>CONTROL MODES</b>	TELE-OP. PROGRAMMABLE. GPS NAVIGATION

# Referencias

- NXT\_Python: <https://github.com/Eelviny/nxt-python>
- <http://www.philohome.com/nxtmotor/nxtmotor.htm>
- USB4Butiá: A truly free as in freedom input/output board  
[http://www.olpcnews.com/use\\_cases/technology/usb4butia\\_a\\_truly\\_free\\_as\\_in\\_freedom\\_input\\_output\\_board.html](http://www.olpcnews.com/use_cases/technology/usb4butia_a_truly_free_as_in_freedom_input_output_board.html)
- USB4All: Interfaz USB genérica para comunicación con dispositivos electrónicos, Aguirre, Fernández, Grossy  
<https://www.fing.edu.uy/inco/grupos/mina/pGrado/pgusb/Docs/tesis.pdf>
- Pybot:  
<https://www.fing.edu.uy/inco/proyectos/butia/mediawiki/index.php/PyBot>