

Manual de usuario.

versión 1.0

[Resumen](#)

[Introducción](#)

[¿Qué es el robot Butiá?](#)

[Antes de programar el robot](#)

[TurtleBots](#)

[¿Qué es TurtleBots?](#)

[Sugar](#)

[Debian](#)

[Otras](#)

[Programando con TurtleBots](#)

[El primer comportamiento](#)

[Preguntas Frecuentes](#)

1. Resumen

El presente documento describe a grandes rasgos de que se trata el proyecto Butiá y la plataforma robótica Butiá SAM. Además sigue los pasos necesarios para instalar el software necesario y empezar a trabajar con el robot.

2. Introducción

1. ¿Qué es el robot Butiá?

El robot Butiá es una plataforma robótica, constructiva, simple y económica que pone al alcance de los estudiantes las herramientas necesarias para interiorizarse en la programación de robots. La versión 1.0 comenzó a ser desarrollada en el año 2009 y, actualmente, la versión 2.0



representa la última versión estable.

En la versión 2.0 de la plataforma robótica se han realizado mejoras tanto a nivel de hardware, de software, como del diseño mecánico.

El robot consta de 8 partes principales: placa de control USB4Butiá, chasis acrílico, ruedas, motores, sensores, piezas de encastre, cables, baterías y cargador.

La placa USB4Butiá nos permite comunicar la parte sensora/actuadora del robot con la computadora. Dicha placa cuenta con 6 puertos de entrada/salida, un bus Dynamixel y hackpines en los cuales tiene conectado un shield para motores de corriente continua pudiendo desarrollarse shields para los más diversos usos.

Los motores del robot son motores de corriente continua, existiendo la posibilidad de cambiarlos por motores Dynamixel AX-12.

El chasis del robot es de acrílico, donde van alojados los motores, placa de control, baterías y la computadora. Una mejora en este chasis es que mediante las piezas de encastre es posible colocar sensores en diferentes posiciones y acoplar piezas de otros kits comerciales como Lego¹, FischerTechnics² y Meccano³ entre otros.

Los sensores incluidos en el kit básico son: escala de grises, intensidad de luz, distancia y botón. Además, pueden agregarse fácilmente sensores de resistencia y voltaje.

3. Antes de programar el robot

El robot Butiá, además del hardware necesita un programa para actuar. El robot, sin ningún programa, no tiene comportamiento y por ende no es capaz de realizar tarea alguna.

Lo que necesitamos entonces para completar el robot es conectarle una computadora, a través de un cable USB, para poder programar su comportamiento. El diseño del robot Butiá permite acoplar varios tipos de laptops como: XO 1.0, 1.5 y 1.75, Magallanes, Olidata o cualquier otra que posea un sistema operativo basado en GNU/Linux.

4. TurtleBots

1. ¿Qué es TurtleBots?

TurtleBots extiende la herramienta predecesora TurtleArts, presente en las laptops XO del proyecto OLPC (Plan Ceibal en Uruguay, agregando varios plugins que la adaptan


¹ <http://mindstorms.lego.com>

² <http://www.fischertechnik.de>

³ <http://www.meccano.com/>



para su uso con robots (Butiá, Lego NxT, Lego WeDo). Otra característica de este software es que permite su instalación en ambientes Sugar y también en otras distribuciones de GNU/Linux.

| | |
|--|--|
|  NOTA: | La última versión de TurtleBots la puede descargar desde: http://www.fing.edu.uy/inco/proyectos/butia/files/package/ |
|--|--|

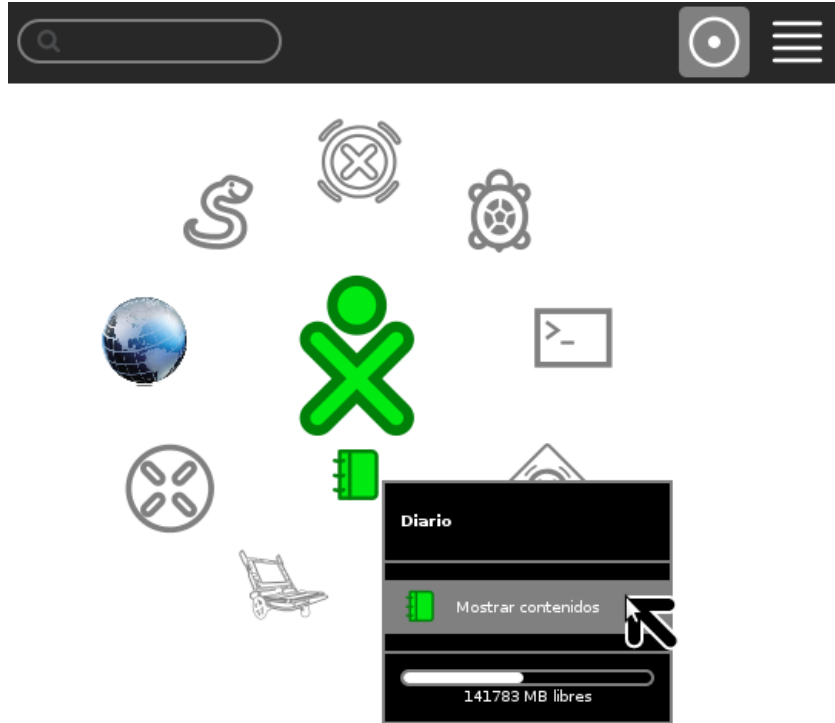
2. Instalación y configuración.

La instalación y configuración del programa en sus diferentes opciones:


1. Sugar

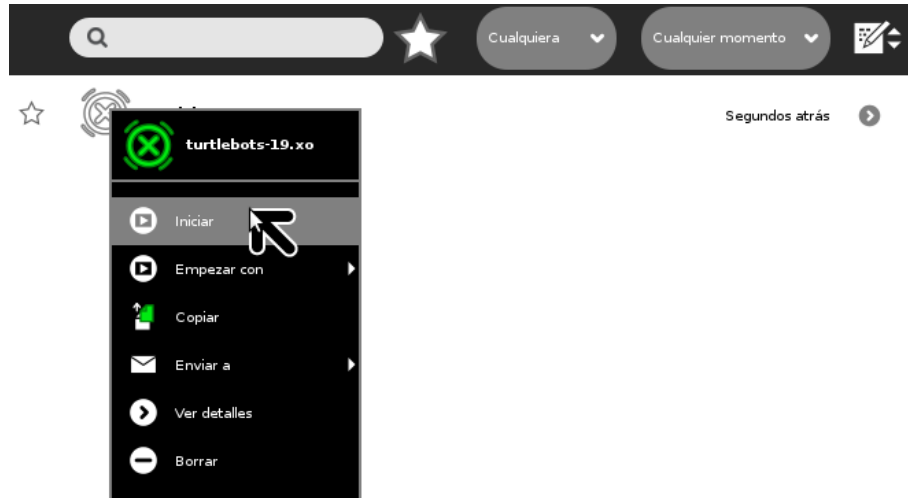
La instalación del TurtleBots se realiza como cualquier otra actividad Sugar: simplemente con un click. Basta con descargar el .xo desde la página. Las reglas necesarias para controlar el robot deberían estar presentes en el sistema, por lo cual, es necesario utilizar la nueva imagen de Ceibal: **1_0a041** o posterior. En el diario podemos encontrar el .xo descargado:





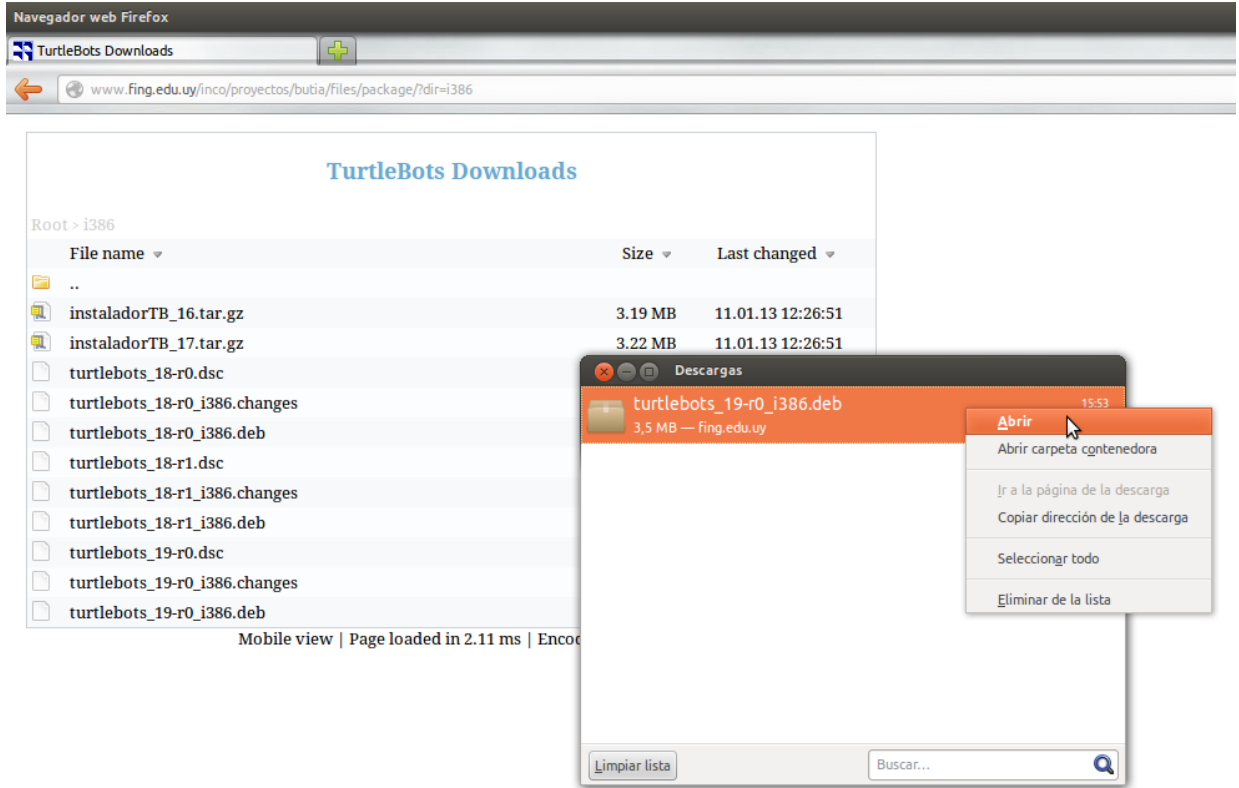
Simply by clicking on the .xo file, the activity will be installed.

| | |
|--|---|
|  NOTA: | La última imagen Sugar de Ceibal la puede descargar desde: http://www.ceibal.edu.uy |
|--|---|

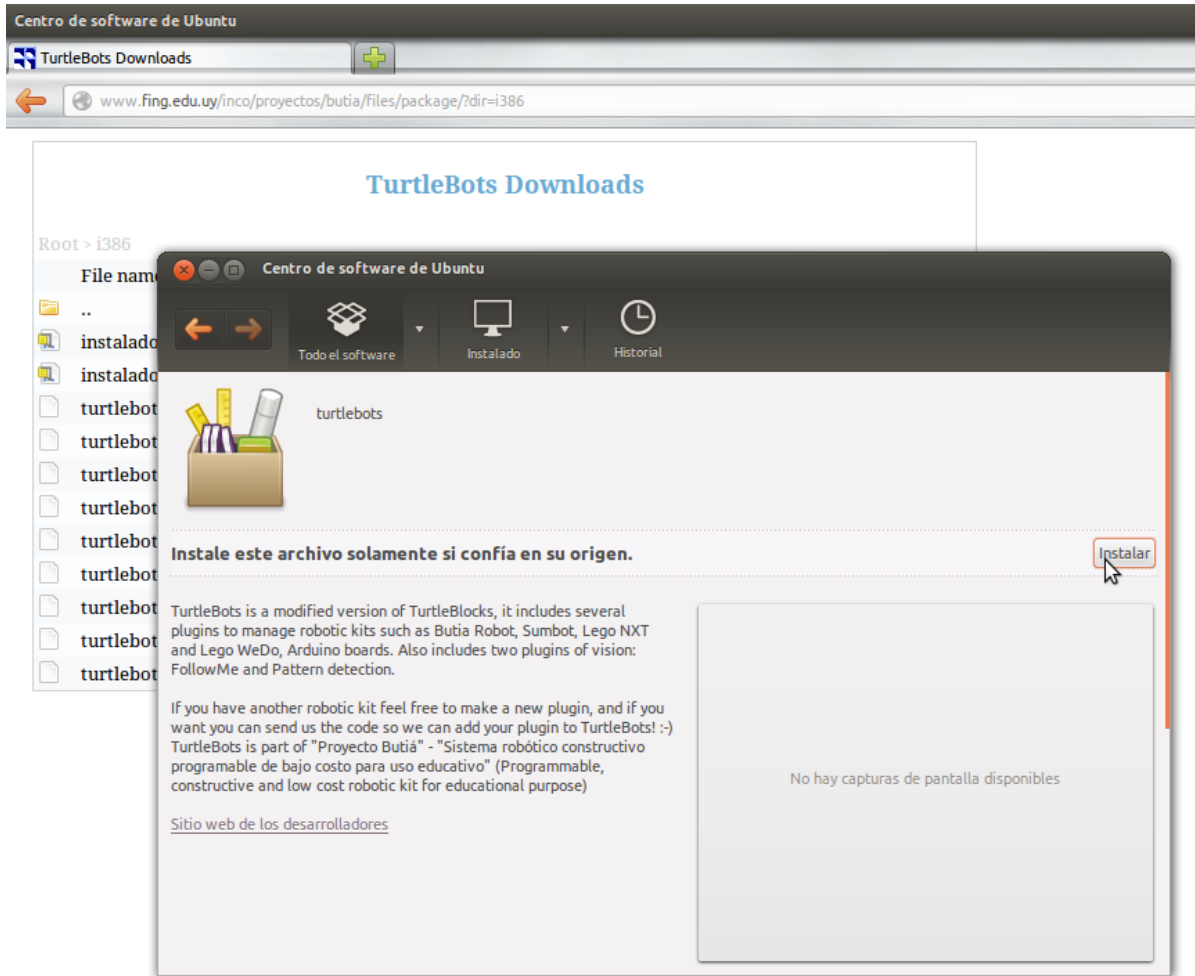


2. Debian

El paquete debian permite instalar TurtleBots en cualquier **GNU/Linux** derivado de Debian: Ubuntu, Edubuntu, Lubuntu, Xubuntu. Inclusive, el paquete permite que no se disponga de Sugar en el entorno (ya que TurtleBots funciona directamente en el escritorio GNOME/KDE sin necesidad de Sugar). Se descarga el paquete para la arquitectura del PC donde se desee instalar:



Se presiona el botón de **instalar** en el instalador que se despliega:



Cuando termine la instalación se puede acceder a la aplicación directamente desde el Inicio:



NOTA:

Error común: “Este paquete entra en conflicto con ‘turtleart’”, esto sucede cuando ya se tiene instalado la actividad *TurtleArt 98*.
Para solucionarlo: Abrir la Terminal y escribir:
`sudo apt-get remove turtleart sugar-turtleart-activity`
Este comando desinstala la actividad. Luego volver a ejecutar el paquete DEB.

3. Otras

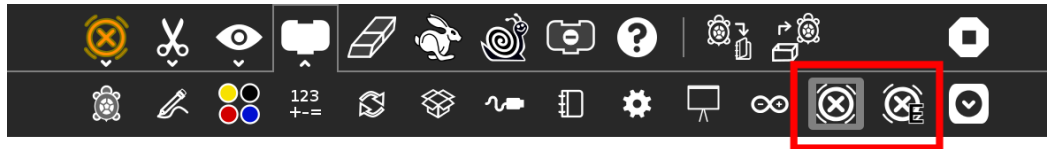
Véase wiki del proyecto Butiá

3. Programando con TurtleBots

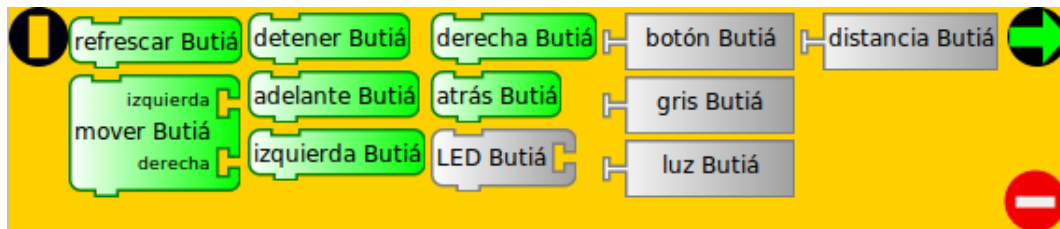
La primer pregunta que debemos hacernos es ¿Qué es TurtleBots?

TurtleBots es un ambiente de desarrollo para robótica, basado en Tortugarte, y que permite al usuario programar comportamientos para el robot mediante la utilización de bloques, facilitando la tarea al usuario que va a utilizar el robot. Con el mecanismo de programación constructivo, similar al armado de un puzzle, aún cuando el usuario no tenga conocimientos previos en el área de la programación, le permite comenzar a interactuar con el robot de manera sencilla y rápida.

Paletas y bloques Butiá



Dentro de lo que son las paletas en TurtleBots encontramos las paletas Butiá y Butiá Extras. La paleta Butiá contiene los bloques para controlar los movimientos y los sensores del robot Butiá.



Los bloques de la paleta Butiá se pueden agrupar en las siguientes tres categorías:
Información:

- refrescar Butiá: actualiza el estado de la paleta y de los bloques Butiá.


Actuación:


- mover Butiá: mueve los motores del robot a la velocidad especificada en las entradas previstas para motor derecho e izquierdo. Esta función permite hacer que una rueda gire mas rápido que otra y así el Butiá puede variar su radio de giro. Rango: 0-1023.
- detener Butiá: detiene el robot.
- adelante Butiá: mueve el robot hacia adelante.
- izquierda Butiá: el robot rota sobre su eje hacia la izquierda.
- derecha Butiá: el robot rota sobre su eje hacia la derecha.
- atrás Butiá: mueve el robot hacia atrás.
- LED Butiá: prende o apaga el LED; 0 es apagado, 1 es prendido. Si está presente, el bloque aparece de color verde, gris en otro caso.

Sensado:

- botón Butiá: en caso de estar presente, devuelve 1 cuando el botón está presionado y 0 en otro caso.
- gris Butiá: en caso de estar presente, devuelve el nivel de gris de un objeto. Rango: 0-65535.
- luz Butiá: devuelve el nivel de luz. Rango: 0-65535.

- *distancia Butiá*: devuelve la distancia al objeto en frente del sensor. Rango: 0-65535.

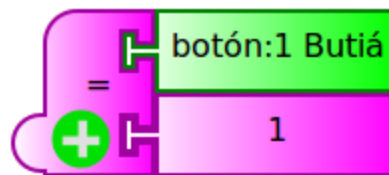
| | |
|--|---|
|  NOTA: | En los bloques de la paleta Butiá se vé el bloque gris:5Butiá de color verde, esto indica que hay un “Sensor de Grises” conectado en el puerto 5. Cada vez que se conecte se cambiará a verde el color del bloque correspondiente y se agregará el número de puerto al que fué conectado. En caso de que el sensor se desconecte su bloque quedará de color gris. |
|--|---|

| | |
|--|--|
|  NOTA: | Por información sobre el uso de la paleta Butiá Extras, referirse a la wiki Butiá. |
|--|--|

Ejemplos de uso

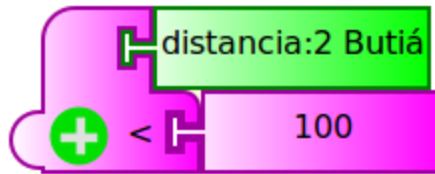
Los bloques de sensor representan tanto sensores digitales como analógicos. Los primeros nos devuelven un dígito binario (1 o 0), mientras los analógicos devuelven un valor entero dentro de un rango especificado.

Botón



Si queremos saber si el botón Butiá está presionado lo comparamos con 1. Esta construcción nos va a devolver Verdadero si el botón Butiá está presionado y Falso en otro caso.

Distancia

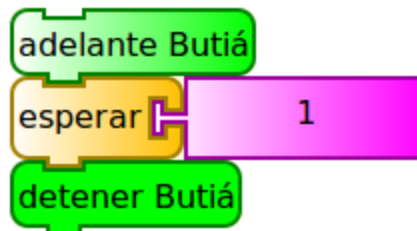


Los bloques de sensor que nos devuelven un rango de valores (en el ejemplo el sensor de Distancia) los comparamos con los bloques de menor o mayor ya que es muy difícil que nos devuelvan exactamente el mismo valor. En el ejemplo, nos va a devolver Verdadero si el valor retornado por el sensor de distancia es menor a 100, Falso en otro caso.

El primer comportamiento

El robot Butiá (o cualquier otro) no realiza tarea alguna si no cuenta con algún programa de software o algoritmo que le indique lo que debe hacer. Estos programas y algoritmos pueden ser muy sencillos o muy complejos, pero siempre, en definitiva, serán responsables del comportamiento que el robot exhibirá durante su ciclo de vida.

En este punto, a modo de primer acercamiento, se muestra un comportamiento simple.

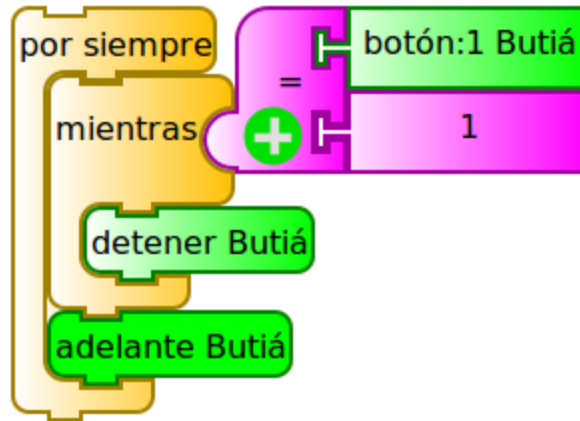


Este código hace que el butiá avance durante 1 segundo y luego se detenga. Los comportamientos del robot van a manejarse de la siguiente manera:


1. se indica una acción
2. se indica durante cuánto tiempo debe realizarse la misma

En el ejemplo, se le da una orden (“adelante butiá”), luego se indica que antes de ejecutar la siguiente acción, debe esperar x segundos (durante ese tiempo x , mientras el robot está avanzando no se ejecuta ninguna otra acción) y seguido a esto se ejecuta la siguiente acción (“detener”).

Comportamiento Avanzado



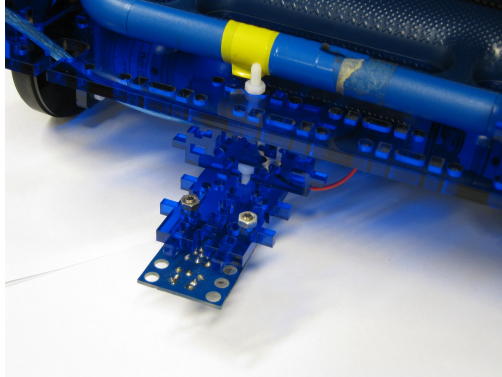
Un comportamiento más avanzado usando un bucle condicional se muestra en este ejemplo. La semántica es la siguiente: mientras el botón no está presionado, el butiá avanza, en cuanto se presiona el botón, el butiá se detiene.

| | |
|--|---|
|  NOTA: | El botón presionado nos devuelve el valor 1, en otro caso nos devuelve 0. |
|--|---|

Seguidor de línea simple

El siguiente código de ejemplo permite que el robot siga una línea negra (recta o curva) sobre un fondo claro. Para esto será necesario utilizar un sensor de escala de grises y programar el comportamiento deseado.

Primeramente, debe conectarse el sensor de escala de grises en algún puerto disponible del Butiá. Uno de los lugares recomendados para dicho sensor es situarlo delante del robot, apuntando hacia el suelo. Existen varias disposiciones de piezas que permitan colocar el sensor de escala de gris apuntando al piso, a modo de ejemplo pueden utilizarse dos piezas “Placa sensor” como se muestra en la figura.



El mismo nos devolverá un valor “pequeño” cuando detecte un color oscuro y un valor “grande” cuando detecte un color claro.

A los efectos de determinar el rango de valores que se corresponden con la zona blanca y con la línea negra deben realizarse algunas pruebas. En el programa, situamos el bloque escala de gris en el lienzo, el área clara en turtlebots donde se pueden arrastrar los bloques, ubicamos el sensor sobre la línea de color negro y hacemos un clic sobre el bloque. El valor devuelto por el sensor es el correspondiente al negro en la escala de grises. Repitiendo el mismo procedimiento ubicando el sensor sobre la superficie blanca, se obtiene el correspondiente para el color blanco. Una vez que se obtienen los valores de referencia es posible escribir el programa. A este procedimiento lo vamos a llamar calibración de sensores.

En la imagen se muestra la forma de usar el sensor para detectar negro, comparando el valor devuelto por el mismo con un valor de referencia, obtenido previamente mediante el procedimiento descrito.



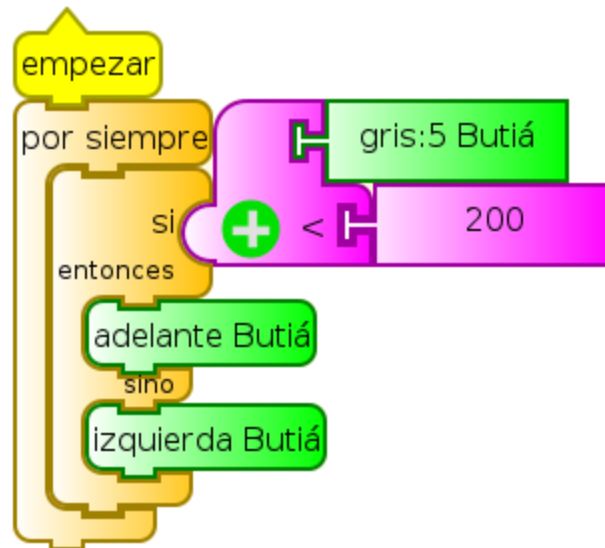
NOTA:

Para más información sobre cómo conectar un sensor, ver el Manual de Armado, que se entrega con este kit.

Por otra parte, se puede concebir el comportamiento de seguir línea considerando dos acciones básicas:

- avanzar: el robot avanza mientras está sobre la línea (detecta el color negro).
- girar: si el robot detecta un color claro (no cumple la condición de ser menor al valor de referencia), gira hacia un lado (izquierda o derecha).

El siguiente código muestra un programa completo que permite seguir líneas. Se puede ver que tiene un bucle principal que consta de un bloque “Por siempre”. Esto implica que una vez en funcionamiento el robot realizará la tarea de seguir línea hasta que se lo apague o se agoten sus baterías. Luego realiza una lectura del sensor y compara el resultado con el valor de referencia previamente establecido (valor correspondiente a “negro”). Si esta comparación es verdadera significa que está sensando el color negro, por lo tanto debe seguir avanzando. En cambio si la comparación resulta falsa, debe girar para encontrar nuevamente la línea negra.



NOTA:

Este seguidor de líneas sólo nos permite seguir la línea con curvas solo hacia un lado, para seguir líneas con curvas hacia los dos lados, mirar el ejemplo de seguidor de línea avanzado en wiki del proyecto Butiá.

5. Preguntas Frecuentes

A continuación se dejan algunas de las preguntas que suelen plantearse los usuarios al comenzar a utilizar *Turtlebots*. Podrá también encontrar esta lista actualizada en la página de preguntas frecuentes (FAQ) del sitio del Proyecto Butiá:

http://www.fing.edu.uy/inco/proyectos/butia/mediawiki/index.php/Preguntas_frecuentes

Enchufa el robot y no se colorean los bloques. ¿Qué pasa?

Pueden ser dos problemas, de corriente o de instalación.

Verificar que la llave del robot esté encendida, las pilas en su lugar. Si esto no soluciona su problema, puede fijarse en la instalación de reglas UDEV y seguir los pasos que se indican en la wiki Butiá.

Prendo el butiá, enchufo el USB, los bloques de los sensores que tengo conectados se colorean pero los bloques de mover no. ¿Qué pasa?

Seguramente sea problema de alimentación o de conexión del shield, revise las baterías y la conexión del shield a la placa USB4Butiá.

Cuando hago click arriba de un bloque de sensor me devuelve -1. ¿Qué significa esto?

El sensor no está conectado. Revise la conexión.



Glosario

Bus Dynamixel: Protocolo de comunicación usado por los motores dynamixel.

Hackpin/Hackpoint: Pines libres de la placa que nos permiten extender la funcionalidad de la placa. Utilizados también para conectar el Shield de Motores.

Shield: Placa que se conecta a los Hackpines y nos permite controlar dos motores de corriente continua.