

DOCUMENTACIÓN LEGO - JAM

Robot: Profesor Jirafales



Integrantes:

Kevin Esteban Gutierrez Gutierrez

Mateo Llano Avendaño

David Alejandro Marín Alzate

Deiry Sofía Navas Muriel

**UNIVERSIDAD DE ANTIOQUIA
INGENIERÍA DE SISTEMAS
ANÁLISIS Y DISEÑO DE SISTEMAS II
MEDELLÍN
2017**

Índice

Lego Jam	3
Objetivos	3
Lineamientos del proyecto	4
Conocimientos previos	4
¿Cómo funciona la Raspberry pi con Brickpi?	4
¿Cómo realizar la conexión inalámbrica de la raspberry para manejarlo?	6
¿Dónde instalar el sistema operativo?	7
Instalación del sistema operativo	7
Guía de funcionamiento	11
Configuración inicial para motores y sensores	11
Modo automático	12
Modo manual	12
Modo disparo	13
Ejecutar código en la terminal	14
Construcción del robot - Resultado	16
Links de interés	21

Lego Jam

Los estudiantes de Análisis y diseño de sistemas II y Estructuras físicas de la información participaron en una competencia de robots, contruidos en Lego y programados en Python.

Se busca dar a conocer los pasos necesarios para la debida construcción de un robot, el cual debe seguir una pista y en cierto momento disparar a un objetivo.

Se presenta los conocimientos previos para entender su funcionamiento, la configuración y ajustes iniciales, donde se tiene un Raspberry pi /Brickpi, que será el cerebro del robot, debido a que contendrá la SD con el sistema operativo; y la parte externa, que está construido con partes de lego.

Objetivos

Desarrollar destrezas para la construcción física y funcional de un robot, esto a través de la utilización de diversas herramientas como fichas LEGO para la parte estructural, un cerebro Brickpi-pi y Raspberry para el desarrollo de la parte lógica del robot. Además se tiene como objetivo ampliar los conocimientos básicos sobre el funcionamiento y lenguaje de programación Python.

La prueba consiste en la creación de un robot con la capacidad de seguir una línea sin salirse de ella. Para ello el robot deberá ser manejado de manera manual y automática. Al momento de encontrar una línea roja, el robot deberá buscar un objetivo (diana) al cual deberá disparar y sobre el cual se evaluará para calificar la puntería y correcto diseño del robot.

Para la construcción del robot se podrá hacer uso de los diferentes sensores que se disponen y se utilizará el raspberry con el controlador de motores y fichas LEGO.

Específicos

- Aprender e implementar el lenguaje de programación Python para la parte lógica del robot.
- Conocer, experimentar el uso de herramientas útiles tales como el raspberry pi
- Asumir y apropiarse del reto de crear un robot que pueda satisfacer los objetivos propuestos por los profesores

Lineamientos del proyecto

Para el correcto funcionamiento del robot y la superación de las pruebas se tienen los siguientes objetivos:

- Hacer uso del raspberry pi como “cerebro” del robot.
- Establecer una conexión vía Bluetooth o wi-fi que permita el manejo del robot a través de un dispositivo móvil.
- Lograr que el robot recorra de manera autónoma una pista formada por una línea negra sin salirse de ella.
- Lograr que el robot pueda seguir una pista de manera manual a través de un dispositivo móvil sin salirse de ella.
- Disparar hacia un objetivo (Diana) que se encuentra a una distancia de 30 cm cuando encuentre una línea roja en la pista. El acierto o no del disparo será medido para el momento de la competencia entre robots.
- Tener un conocimiento claro sobre la estructura de los robots creados para el momento del desarme y armado de ellos.

Conocimientos previos

Para el correcto funcionamiento de nuestro robot es necesario primeramente entender conceptos bases que nos permitirán tener un panorama más amplio de lo que vamos a realizar, por ello veremos a continuación algunos conceptos que son de gran ayuda antes de comenzar.

¿Cómo funciona la Raspberry pi con Brickpi?

Básicamente un Raspberry Pi es un computador de placa reducida, computador de placa única o computador de placa simple para promover la enseñanza básica de la informática. Por tanto este será nuestro “cerebro” sobre el cual trabajaremos y pondremos a correr nuestro código.



Imagen 1: Raspberry Pi

Mientras que el Brickpi es un complemento que utilizaremos, es un sistema que convierte la raspberry pi en un robot, debido a que está formado por una placa que se conecta a la Raspberry Pi y que hace de interfaz con los elementos de LEGO Mindstorms.



Imagen 2: Brick Pi

Es muy importante tener en cuenta que el Brickpi consta de unos puertos los cuales utilizaremos para conectar con nuestros motores y sensores, por tanto es necesario conocer la ubicación de ellos.

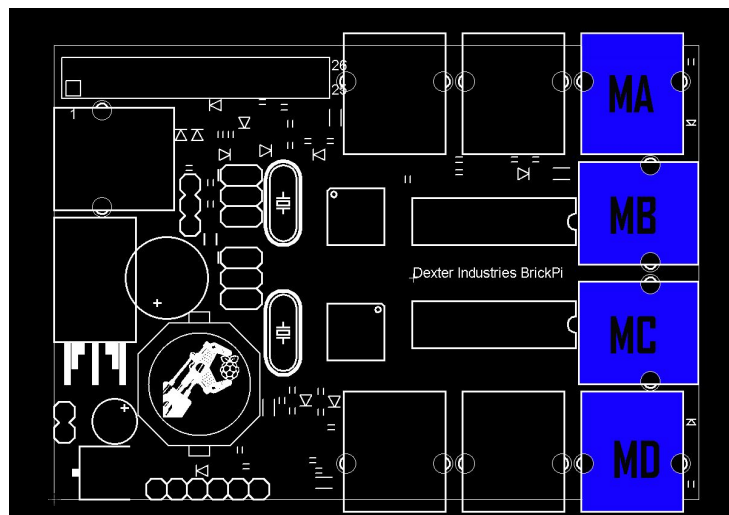


Imagen 3: motores BrickPi

El brickpi cuenta con 4 puertos para motores, por tanto esa es la cantidad de máxima de motores que podemos utilizar.

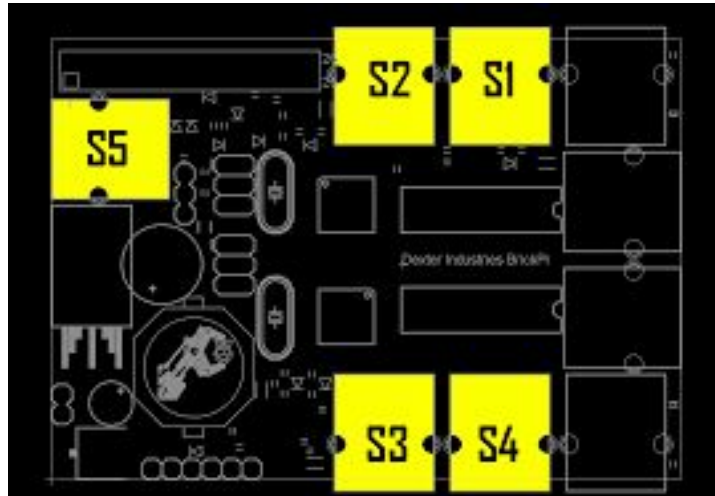


Imagen 4: Sensores BrickPi
 El brickpi cuenta con 5 puertos para sensores, por tanto esa es la cantidad de máxima de motores que podemos utilizar.

¿Cómo realizar la conexión inalámbrica de la raspberry para manejarlo?

Se realizó por medio de un adaptador Wifi, dispositivo que permite añadirle la funcionalidad a nuestra raspberry, puede ser utilizada para establecer la conexión del raspberry con un dispositivo móvil a través de una [conexión SSH](#) o a través de una [aplicación móvil](#).



Imagen 5: Adaptador wifi Dexter Industries

¿Dónde instalar el sistema operativo?

La tarjeta Secure Digital (SD) es un dispositivo en formato de tarjeta de memoria para dispositivos móviles, en nuestro caso en Raspberry.

La tarjeta SD será muy importante, ya que será la encargada de almacenar el sistema operativo de la raspberry, por ello tendrá que ser **mínimo de 8 gb** de almacenamiento. Recomendamos utilizar una propia para el proyecto debido a las pocas unidades disponibles.



Imagen 6: Tarjeta SD Dexter Industries

Instalación del sistema operativo

Para el uso del raspberry pi es necesario el uso de una tarjeta SD, ya que esta tendrá el sistema operativo que nos permitirá operar nuestra placa o computador. Por esta razón se recomienda utilizar una tarjeta SD de 8 gb de espacio de almacenamiento, sin embargo una de 4 gb será útil para almacenar el Sistema operativo.

En caso de que la tarjeta SD ya tenga el sistema operativo instalado no será necesario realizar los siguientes pasos, en caso contrario será de gran ayuda seguir estas recomendaciones.

Lo primero que debemos hacer es ingresar a la pagina www.dexterindustries.com y acceder a la opción “robots” y allí seleccionar “Brickpi”, una vez cargada la página ingresamos en la sección “Connect LEGO” y allí “Step by Step”



Looking for BrickPi instructions and projects? Click here.

BrickPi

If you like LEGO, you'll love BrickPi.

[Get It Now](#)

RASPBERRY PI

Replace LEGO NXT/EV3 with the Raspberry Pi-BrickPi duo for more functionality.

[What is Raspberry Pi?](#)

CONNECT LEGO

Integrate web services, network multiple robots together, and operate remotely through wi-fi.

[Step-by-step Tutorial](#)

MORE LANGUAGES

Build with LEGO sensors, motor & Technic parts but program in Python, Scratch, or Java.

[See the full list](#)

Imagen 7: página principal Dexter industries

Una vez allí, desplegamos la opción “Getting Started”, recomendamos leer la documentación que aparece para reforzar conceptos e ideas para la construcción del LEGO.

Seleccionamos la opción 2 “Set up SD card”

1. Assemble the Case

- Assemble the Basic Case: 2-sided BrickPi case (works with all versions of Raspberry Pi)
- Assemble the Upgraded Case: 6-sided BrickPi B+ case (works with Raspberry Pi B+ & 2)

2. Set up the SD Card

3. Power up

4. Set up your BrickPi

- Set up from Windows
- Set up from a Mac
- Set up Wi-Fi

Imagen 8 :Información Dexter industries

Allí se encuentra la documentación sobre la instalación paso a paso del sistema operativo sobre la SD card.

El sistema operativo se puede descargar a través del siguiente link <https://drive.google.com/file/d/oBoWChwP4CnLBOFNQlFobFcXWUU/view>

También recomendamos descargar el programa “Etcher” en nuestro computador para montar la imagen del sistema operativo en la SD. Se puede descargar a través de: <https://etcher.io/>

Una vez con este programa instalado crearemos una imagen del sistema operativo en la SD, seleccionaremos el archivo, luego el espacio de almacenamiento y daremos continuar hasta finalizar el proceso.

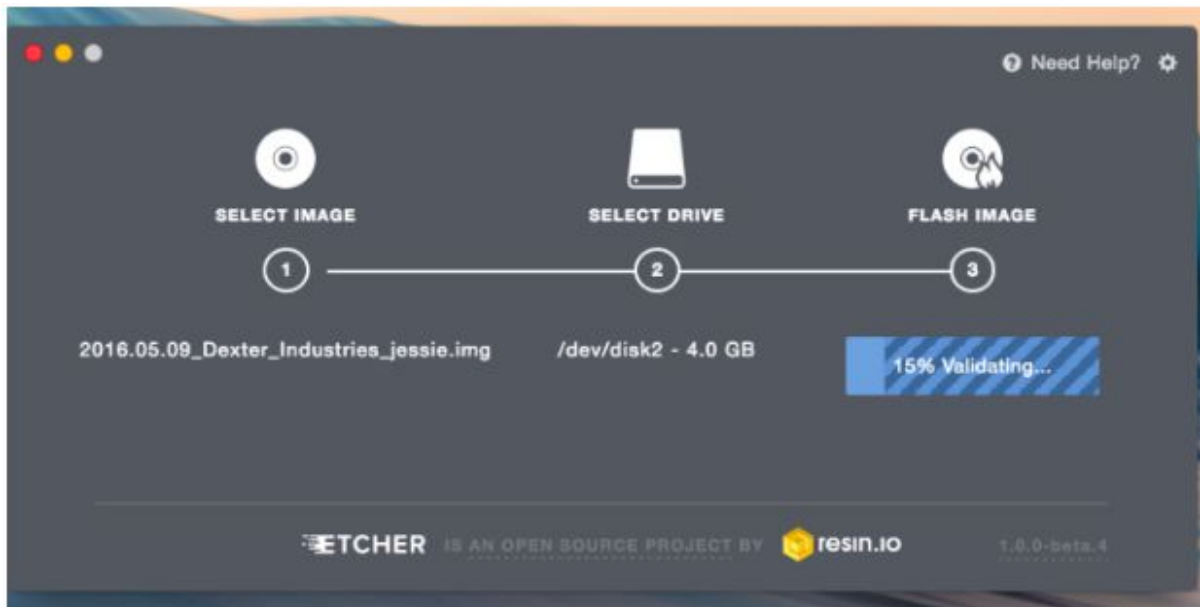


Imagen 9: Funcionamiento programa “Etcher”

Luego de haber realizado el proceso ingresamos la tarjeta SD a nuestro brickpi y lo conectaremos a través de una fuente de energía de máximo 12 voltios.

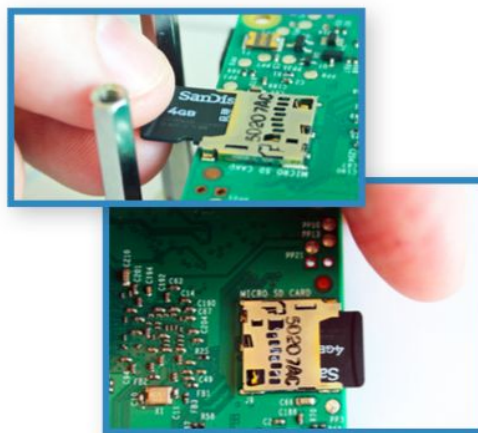


Imagen 10: Inserción SD card

Después de realizar los pasos anteriores, conectamos el raspberry a nuestra pc a través de un cable de red (LAN) y accedemos desde nuestro computador a Dex.local/

Guía de funcionamiento

Configuración inicial para motores y sensores

1. Importar la librería y Setup BrickPi

```
from BrickPi import *  
BrickPiSetup()  
BrickPiSetupSensors() // Setup para los sensores
```

2. Configurar los motores y el sensor, debemos identificar en qué puerto están conectados, como se explica previamente [aquí](#).

```
#B motor izquierdo  
B = PORT_B  
#D motor derecho  
D = PORT_D  
#C disparar  
C = PORT_C  
#s1 sensor  
S1 = PORT_1  
  
BrickPi.MotorEnable[D] = 1  
BrickPi.MotorEnable[B] = 1  
BrickPi.MotorEnable[C] = 1  
BrickPi.SensorType[S1] = TYPE_SENSOR_EV3_COLOR_M2
```

❖ Para tener en cuenta:

Hay que darle un tiempo a los motores para que vuelvan y actualicen sus valores. min 0.01 segundos

```
time.sleep(0.01)
```

Cada vez que se cambian los valores del motor y sensor hay que actualizarlos

```
BrickPiUpdateValues()
```

¿Cómo asignar valores al motor?

```
BrickPi.MotorSpeed[D] = 120  
BrickPi.MotorSpeed[B] = 120  
  
BrickPiUpdateValues()
```

¿Cómo obtener los valores del sensor?

```
color_sensor = BrickPi.Sensor[S1]
```

Modo automático

Para este modo se usa un solo sensor de color, el funcionamiento básico es que mientras detecte la línea negra los dos motores funcionan a la misma velocidad, cuando detecta un color que no sea negro, se detienen los dos motores, y el robot empieza a oscilar de izquierda a derecha invirtiendo el movimiento de los motores para que se mueva circularmente, aumentando gradualmente el radio de giro hasta que el sensor detecte la línea negra de nuevo, para continuar con el movimiento de los dos motores.

Modo manual

En este modo se maneja el robot desde los eventos del **teclado**, con la siguiente librería

```
import curses
```

Para obtener la tecla presionada

```
st = curses.initscr()
    curses.cbreak()
    st.keypad(1)
    key = st.getch()
    st.refresh()
```

Identificar que tecla fue presionada

Teclas de desplazamiento: `key == curses.KEY_LEFT`

Letra : `key == ord('d')`

Ejemplo:

```
#tecla flecha izquierda - el robot gira a la izquierda
if key == curses.KEY_LEFT :
    print "izq"
    BrickPi.MotorSpeed[D] = 80
    BrickPi.MotorSpeed[B] = -30

#tecla flecha derecha - el robot gira a la derecha
if key == curses.KEY_RIGHT :
    print "dere"
    BrickPi.MotorSpeed[D] = -30
    BrickPi.MotorSpeed[B] = 80

#tecla flecha arriba - el robot se mueve hacia adelante
if key == curses.KEY_UP :
    BrickPi.MotorSpeed[D] = 120
    BrickPi.MotorSpeed[B] = 120
```


Modo disparo

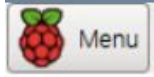
Este modo básicamente funciona de la misma manera que el modo manual, sólo con unos ligeras cambios, el más importante de ellos es que se les reduce la velocidad a los motores para así tener movimientos menos bruscos, para si poder acertar al objetivo con precisión, también se añade la funcionalidad de disparar y presionando la tecla “Q” podemos salir de este modo y así seguir en nuestro modo manual.

Código fuente:

El código fuente se puede encontrar [aquí](#).

Ejecutar código en la terminal

Para abrir la terminal es tan solo con oprimir en el icono de la terminal  que está en

la barra de tareas, otra manera de abrir la terminal es dar click en menú  luego

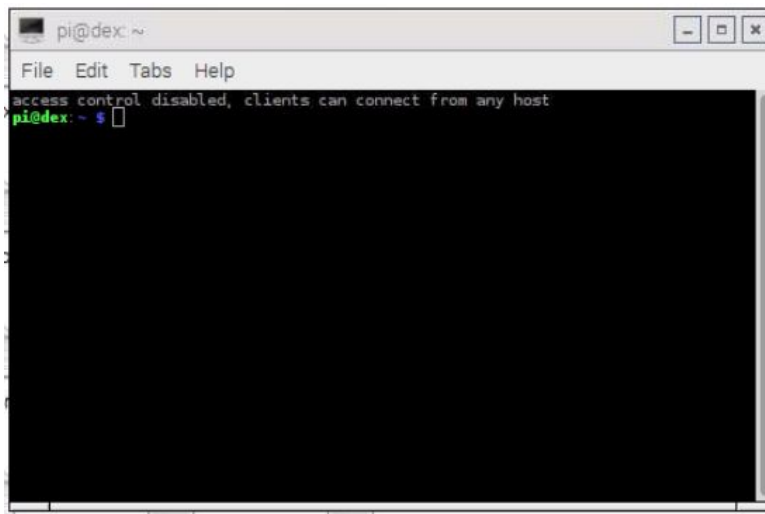
dar click en Other



luego buscamos en el menú LXTerminal



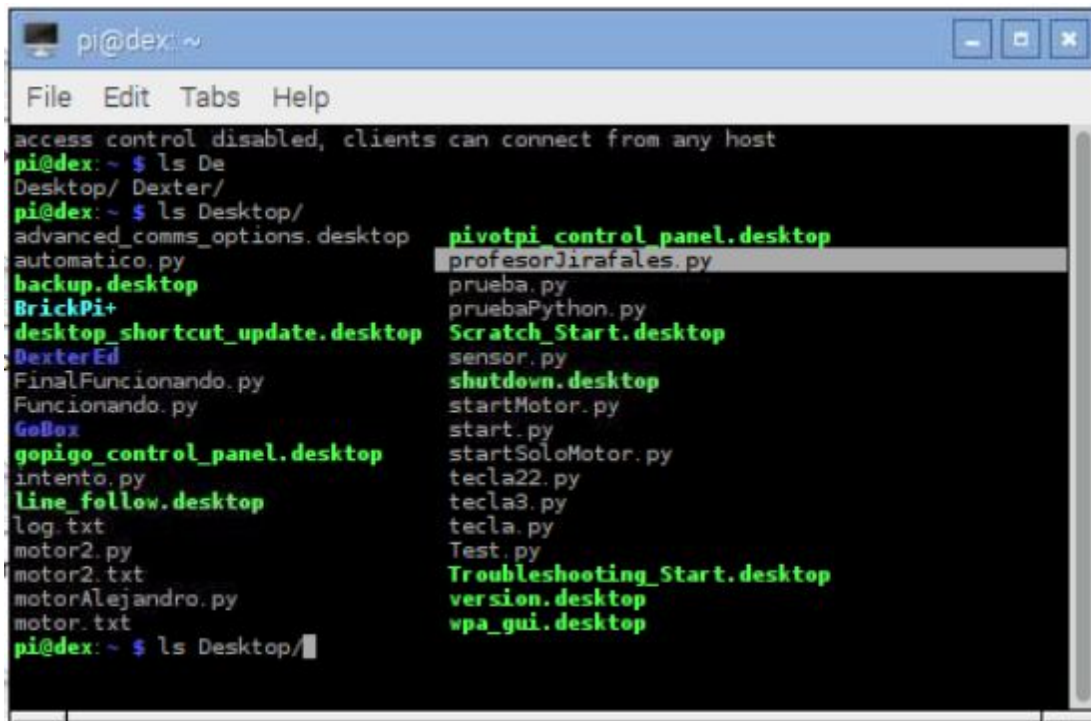
luego no abrirá la terminal.



Para ejecutar el código tenemos que saber donde lo tenemos alojado, lo normal es guardarlo en el escritorio para mirar si está en el escritorio tecleamos en la terminal.

```
pi@dex ~ $ ls Desktop/
```

Debe aparecer una lista con los archivos que hay en el escritorio ahí debe aparecer nuestro código en python llamado profesorJirafales.py



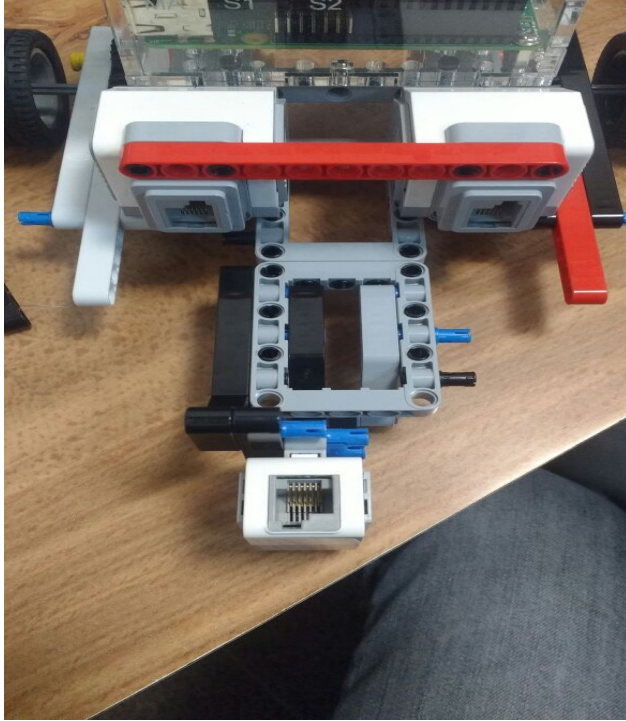
```
pi@dex: ~  
File Edit Tabs Help  
access control disabled, clients can connect from any host  
pi@dex: ~ $ ls De  
Desktop/ Dexter/  
pi@dex: ~ $ ls Desktop/  
advanced_comms_options.desktop pivotpi_control_panel.desktop  
automatico.py profesorJirafales.py  
backup.desktop prueba.py  
BrickPi+ pruebaPython.py  
desktop_shortcut_update.desktop Scratch_Start.desktop  
DexterEd sensor.py  
FinalFuncionando.py shutdown.desktop  
Funcionando.py startMotor.py  
GoBox start.py  
gopigo_control_panel.desktop startSoloMotor.py  
intento.py tecla22.py  
line_follow.desktop tecla3.py  
log.txt tecla.py  
motor2.py Test.py  
motor2.txt Troubleshooting_Start.desktop  
motorAlejandro.py version.desktop  
motor.txt wpa_gui.desktop  
pi@dex: ~ $ ls Desktop/
```

Para ejecutar el script de python en la terminal se escribe la siguiente línea y presionamos la tecla ENTER.

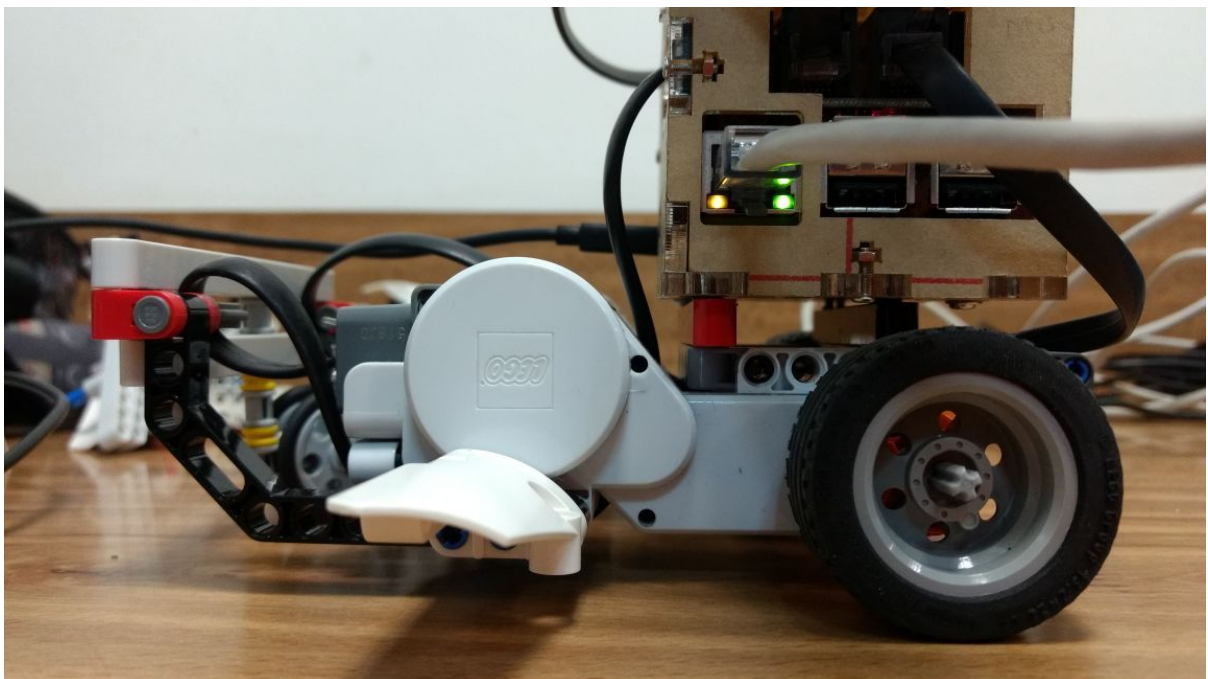
```
pi@dex ~ $ python Desktop/profesorJirafales.py
```

El script comenzará a ejecutarse. Para salir del script se puede presionar Ctrl + C o saliendo de la terminal.

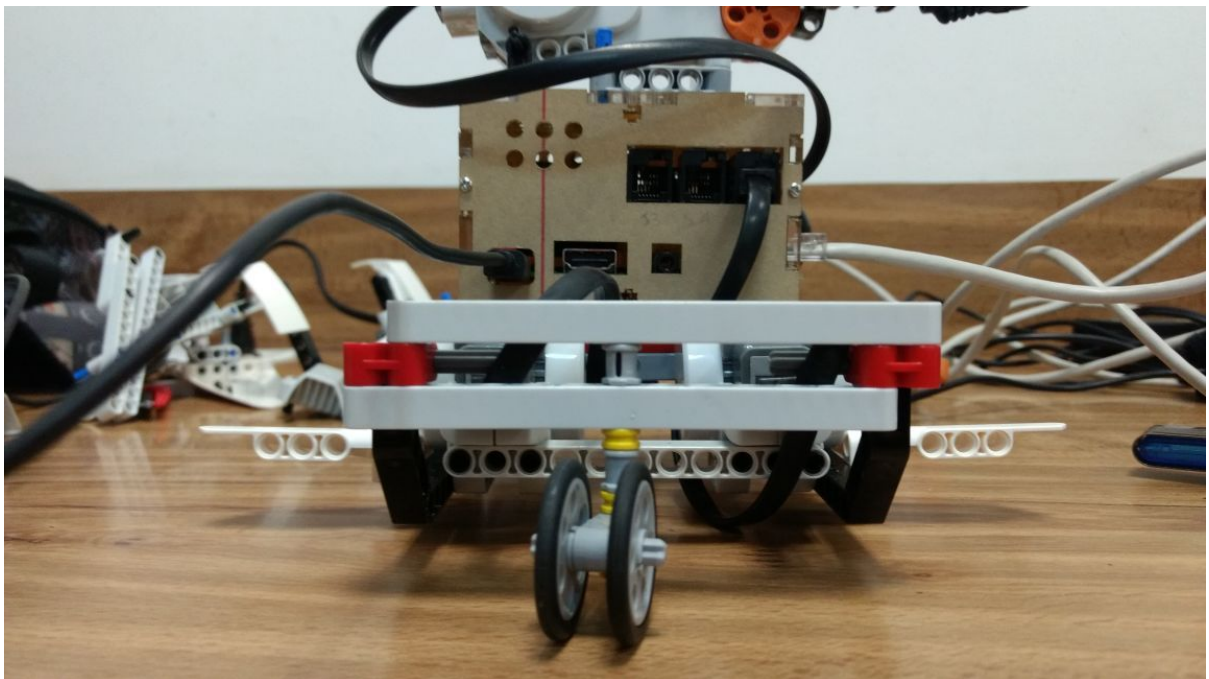
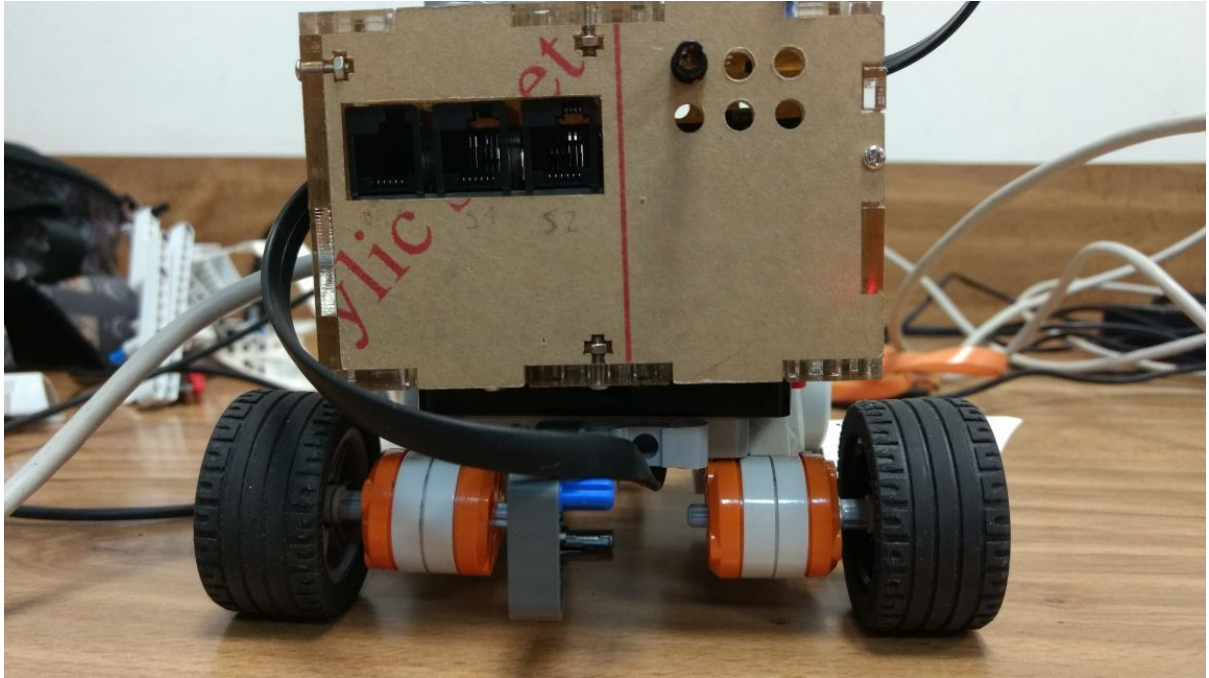
Construcción del robot - Resultado



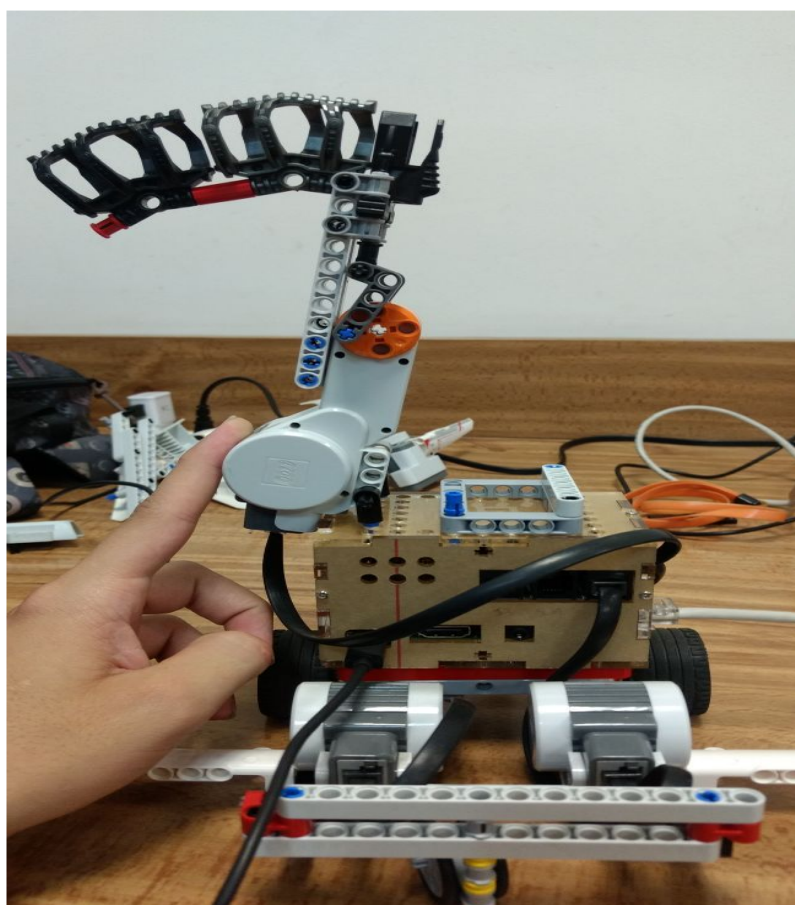
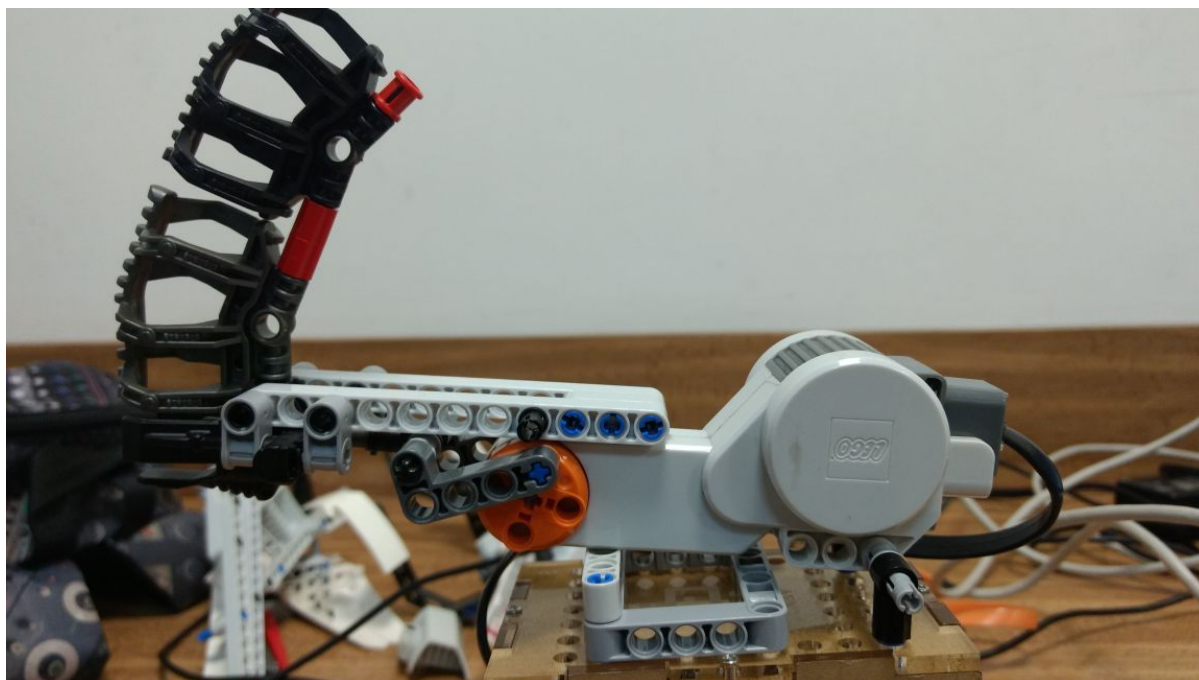
Adaptación del Raspberry pi /Brickpi



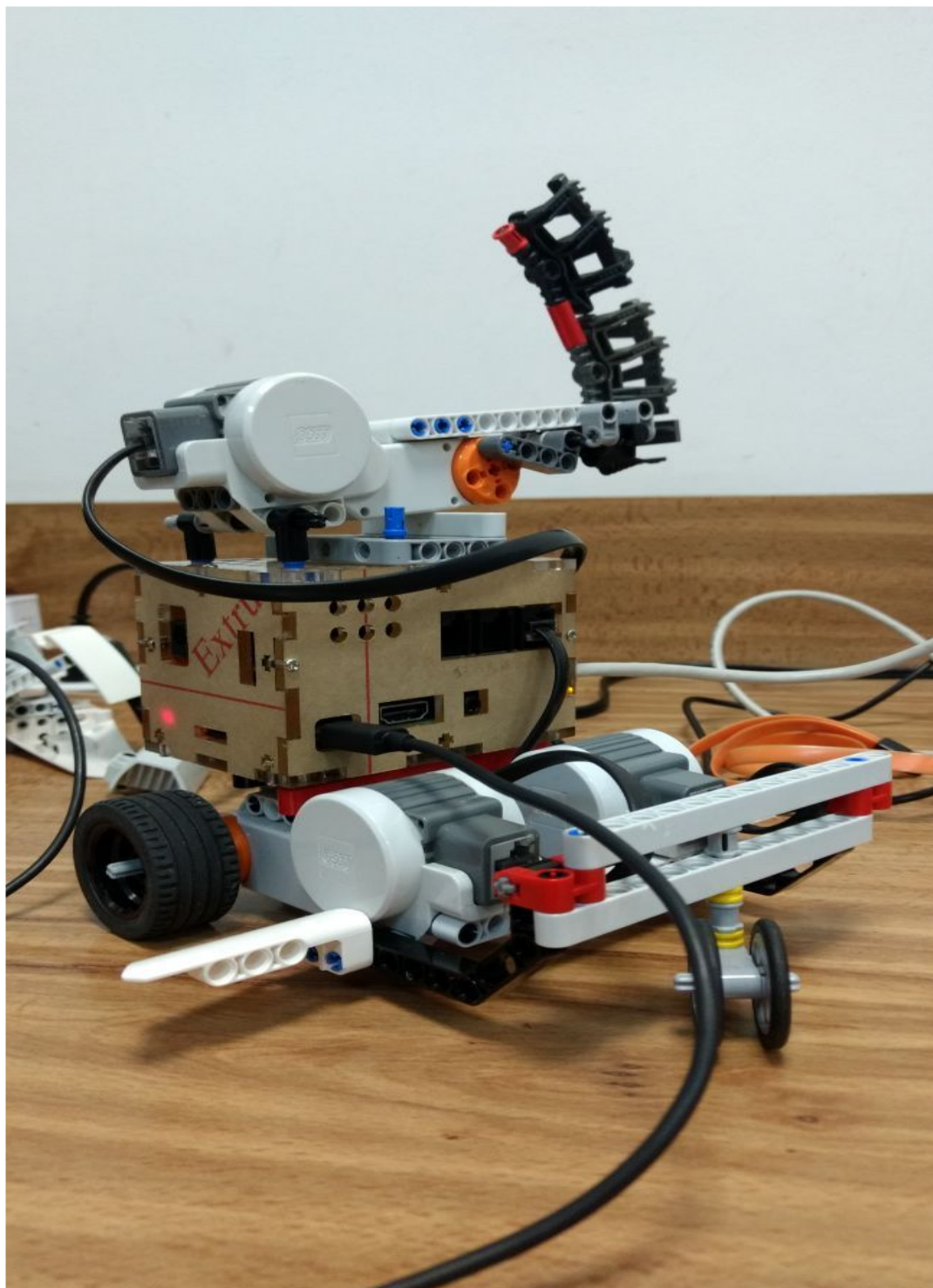
Partes traseras y delanteras del robot



Instalación del mecanismo para realizar el disparo



Así luce el robot terminado!!



Links de interés

Github ejemplos de python para el BrickPi Dexter Industries

https://github.com/DexterInd/BrickPi/tree/master/Software/BrickPi_Python

Github ejemplos de sen

https://github.com/DexterInd/BrickPi/blob/master/Software/BrickPi_Python/Sensor_Examples/EV3-ColorSensor.py

Home Page BrickPi Dexter Industries

<https://www.dexterindustries.com/brickpi/>

Resultado final

<https://goo.gl/photos/jt8eSvS7QyTdTgb1A>