

Introducción a la programación con Lego Mindstorms

Profesores coordinadores: María Luisa Porto Llamas y Iago

Autores del trabajo: Alejandro Rico y Alexander Mychlo de 1º Bachillerato

I.E.S Severo Ochoa

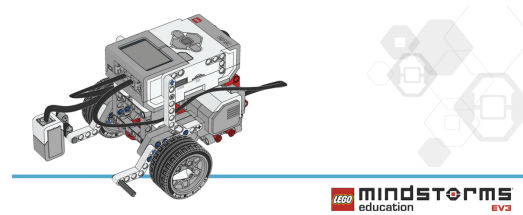
2021-2022

Índice:

1. Resumen introductorio (abstract y palabras claves)
2. Introducción
3. Finalidad
4. Planificación
5. Cuerpo del trabajo (material y métodos)
 - 5.1. Utilización de Git y Github para la creación de conocimiento para futuros proyectos.
 - 5.2. Diferencias entre bloques y python.
 - 5.3. Primeros pasos con el robot educador.
 - 5.4. Principios de Python aplicados al robot.
 - 5.4.1. Librerías de python y teoría aplicada a la programación de mindstorms.
 - 5.4.1.1. Declaración de utilización de librerías y que contiene cada una de ellas.
 - 5.4.1.2. Utilización de variables y relación con el álgebra.
 - 5.4.1.3. Utilización de estructuras y bucles de control o condicionales.
 - 5.4.1.4. Utilización de funciones.
 - 5.4.2. Selección de puertos del Bloque Inteligente EV3.
 - 5.4.3. Inicialización del bloque y movimientos básicos.
 - 5.4.4. Inicialización de los sensores
 - 5.4.5. Explicación de programa complejo elaborado con python y Ev3 mindstorms.
 - 5.5. Pruebas con distintos sensores.
 - 5.5.1. Sensor de contacto
 - 5.5.2. Sensor ultrasónico
 - 5.5.3. Sensor giroscópico
 - 5.5.4. Sensor de colores
 - 5.6. LeoCad e ingeniería del modelo.
 - 5.6.1. Diseño 3D del modelo educador.
 - 5.6.2. Diseño 3D del modelo educador con sensor.
 - 5.6.3. Diseño 3D de soluciones para el robot de SUMO.
6. Realización de un proyecto global con lo aprendido.
7. Conclusión
8. Valoración personal
9. Agradecimientos
10. Bibliografía

Introducción

Durante el valor investigativo de nuestro trabajo hemos descubierto y documentado los materiales necesarios para empezar este proyecto:



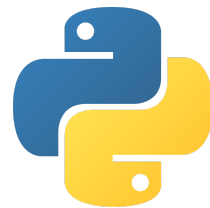
Físicos:

- Lego Mindstorms EV3 Set Básico.
- Cargador.
- MicroSDHC (Tarjeta MicroSD de capacidad máxima de 32GB).
- Ordenador o portátil con acceso a Internet.



Digitales:

- Microsoft Visual Code.
- Documentación.
- Software oficial de Lego Mindstorms EV3 para programar con bloques.
- Última versión de Python.
- Archivo .iso de MicroPython para la tarjeta MicroSD.
- Procedimiento



Visual Studio Code

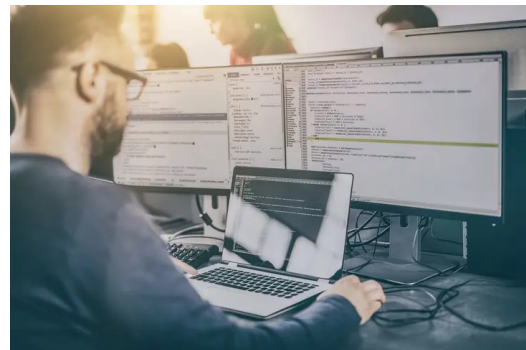
Finalidad

La finalidad de este proyecto es abrir un campo de investigación sobre las posibilidades que tiene el kit lego mindstorms EV3.

En este trabajo estamos buscando poner un precedente para las futuras aplicaciones de este set en el instituto. Además de exponer las razones por las que debemos utilizar un sistema por delante del otro, con sus pros y contras.

Después de valorar este paso nos adentraremos en las funciones básicas del kit para que en un futuro las bases estén completadas para poder hacer trabajos más complejos en este ámbito. En este proyecto además de lo mencionado anteriormente queremos hacer soluciones de ingeniería a los problemas que nos han surgido en el paso del proyecto. Esto no solo incluye la parte del software sino también a la de construcción del robot, como veremos en el cuerpo del proyecto.

Para concluir nuestro proyecto también tiene la finalidad de difundir lenguajes en auge como python, además de el mundo del software en general , y de sus amplias posibilidades, incluso profesionales.



Cuerpo de trabajo

Utilización de Git y Github para la creación de conocimiento para futuros proyectos.

Git es una herramienta utilizada en el ámbito profesional y científico de proyectos tecnológicos. Sirve para organizar, guardar y documentar el proceso que se haya llevado a cabo entre un grupo de personas, ya que permite que cada una pueda hacer sus cambios por sí solo y luego “empujarlos” al servidor donde esté guardado el proyecto.

Esos cambios que hace un programador se llaman “commits” y al enviarlo al servidor principal se puede ver quién ha realizado esos cambios y cuando lo han hecho. Además el servidor guarda el estado del proyecto en cada commit, y por lo tanto permite documentar el proceso que se ha llevado a cabo.

La plataforma Github es una página web que sirve como un portal a los servidores donde se guardarán los proyectos. Es semejante al almacenamiento cloud de Google Drive, pero en la perspectiva de Git.

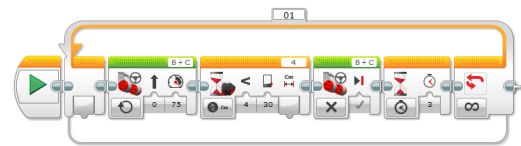
Github resulta ser muy útil para el ámbito del trabajo ya que varias compañías como Microsoft y Netflix lo utilizan para crear sus programas y aplicaciones. También es útil para la creación, mejoramiento y descubrimiento de programas de código abierto, que son programas donde cualquier persona puede cambiar el código, siempre y cuando el

o los propietarios o encargados de ese proyecto hayan verificado los cambios, similar a lo que ocurre en Wikipedia. Resulta ser muy importante para nuestro proyecto ya que el programa que se instala en el microSDHC se llama ev3dev y es justo un programa de código abierto que ha sido comprobado por Lego.

<https://github.com/ev3dev/ev3dev>

Diferencias entre programación de bloques y python.

Al empezar con Lego Mindstorms EV3 y su documentación nos encontramos con el programa oficial de Lego Mindstorms que nos enseñara a programar con bloques.



Por eso empezamos a mirar la posibilidad de buscar la forma de programar con un entorno con el cual ya estamos ya acostumbrados, la programación por código. El entorno recomendado para trabajar con código y Lego Mindstorms es python. Aquí podemos observar cómo solucionamos el problema y en este código el robot avanzara 1000 mm y girara 360 grados hacia la derecha y volverá a su posición inicial tras hacer otro giro con el mismo ángulo.

```
35 robot.straight(1000)
36 robot.turn(360)
37 robot.turn(-360)
```

Python es un lenguaje de programación muy sencillo que nos puede ayudar a adentrarnos en lenguajes más complejos como C++, Java... Python tiene la parte positiva de que es muy sencillo pero a su vez tiene su parte negativa, que es que tiene un tiempo de respuesta mucho más lento que los lenguajes mencionados anteriormente. Python tiene otra cara por lo que es muy conocido, ya que se puede utilizar para el mundo del hacking, ya sea ético o no (el hacking ético es el cual su finalidad no es dañar a usuarios).

Igualmente la programación por bloques es una forma muy intuitiva, fácil y visual de empezar a tener recursos para poder más adelante programar con código.

Desde nuestro punto de vista la programación con código debería ser la forma de hacer este proyecto, ya que puede darte unas experiencias que pueden incluso ser válidas para un futuro laboral o profesional.

La programación con bloques puede ser una buena idea, desde nuestro punto de vista, para personas de primero de la ESO hasta tercero de este mismo ciclo. La programación en código puede ser interesante a partir de tercero o cuarto de la educación secundaria obligatoria.

Primeros pasos con el robot educador

El primero de los pasos es completar el set básico de Lego Mindstorms EV3 siguiendo los pasos dentro del manual para crear al robot básico.

Después de haber construido el robot básico y asegurarse de que el programa incluido dentro de él funcionase, podemos continuar utilizando el programa oficial de Lego Mindstorms EV3 para programar con los bloques y completar los ejercicios dentro de este. A continuación, para aventurarse más dentro de lo posible que se puede hacer con este proyecto, se instala el .iso de MicroPython en el MicroSD y se inserta dentro del slot del robot (tiene que estar apagado). Al encenderlo se verá que el bloque principal se iniciará en una partición de Linux llamada “ev3dev” que permite transmitir archivos de Python y utilizarlos. Si se desea volver al sistema básico anterior, simplemente con quitar la tarjeta MicroSD cuando el robot esté apagado reiniciaría dentro de él.

A continuación, para aventurarse más Utilizando una extensión de Microsoft Visual Code llamada “LEGO MINDSTORMS EV3 MicroPython” se puede conectar el bloque principal al ordenador y transmitir archivos de Python para ejecutar dentro de un simple programa.

```
Movimiento > movimiento_basico.py @ robot
1  #!/usr/bin/env pybricks-micropython
2
3  """
4  Example LEGO MINDSTORMS EV3 Robot Educator Driving Base Program
5  .....
6
7  This program requires LEGO MINDSTORMS EV3 MicroPython v2.0.
8  Download: https://education.lego.com/en-us/support/mindstorms-ev3/python-for-ev3
9
10 Building instructions can be found at:
11 https://education.lego.com/en-us/support/mindstorms-ev3/building-instructions#robot
12 """
13
14 from pybricks.hubs import EV3Brick
15 from pybricks.ev3devices import Motor
16 from pybricks.parameters import Port
17 from pybricks.robotics import DriveBase
18
19 # Initialize the EV3 Brick.
20 ev3 = EV3Brick()
21
22 # Initialize the motors.
23 left_motor = Motor(Port.A)
24 right_motor = Motor(Port.B)
25
26 # Initialize the drive base.
27 robot = DriveBase(left_motor, right_motor, wheel_diameter=55.5, axle_track=184)
28
29 # Go forward and backwards for one meter.
30 robot.straight(1000)
31 ev3.speaker.beep()
32
33 robot.straight(-1000)
34 ev3.speaker.beep()
35
36 # Turn clockwise by 360 degrees and back again.
```

Principios de Python aplicados al robot.

Para poder continuar con el proyecto necesitamos unas nociones muy básicas del lenguaje de programación que estamos utilizando, python. Este lenguaje, como mencionado anteriormente es de los más

sencillos que puedes encontrar ahora mismo. Este apartado solo tratará la teoría necesaria para entender los programas para el robot. Hay mucha documentación sobre este tema y muy interesante sobre el tema así que os animamos a informarnos.

Librerías de python y teoría aplicada a la programación de mindstorms.

Declaración de utilización de librerías y que contiene cada una de ellas.

Python en sí mismo es un lenguaje muy limitado, pero hay autores que añaden más características a este programa. Estos grupos de características son llamados librerías. Podemos encontrar desde librerías sencillas para crear programas para descargar videos de youtube, pasando por otras que quitan el fondo automáticamente de las imágenes, hasta otras que ayudan a crear programas de reconocimiento facial. Estas librerías ocupan todos los campos imaginables, incluso, como mencionado anteriormente, el mundo del hacking.

Como ya ha sido mencionado en el anterior apartado al instalar la extensión de Visual Studio Code “LEGO

MINDSTORMS EV3 MicroPython” se instalan también las librerías que contienen lo que nos va a ayudar a hacer crear los programas para el robot.

A continuación veremos que hacen cada una de estas librerías, para poder observar que se puede hacer con el kit. En total hay nueve librerías pero solo hablaremos

Programmable Hubs

```
from pybricks.hubs
```

Esta librería hace todo lo relacionado con el Bloque Inteligente EV3. La función más importante que cumple esta librería es la inicialización del bloque, que hablaremos más adelante. Además de inicializar el Bloque Inteligente EV3 esta librería también se encarga de otras tareas muy interesantes como, la utilización de los botones de la base,



por ejemplo que al pulsar el botón central el bloque haga algo. En la foto anterior podemos observar que el bloque tiene una luz (brick status light), también se puede controlar con esta librería poniendo cualquier color. Otra función muy interesante de esta librería

es el altavoz integrado en el bloque (speaker). Nosotros hemos utilizado mucho esta función para saber qué punto del programa estaba fallando, ya que si sonaba el “beep” hasta el punto programado sabíamos que hasta ahí estaba todo bien.

```
30 robot.straight(1000)
31 ev3.speaker.beep()
32
33 robot.straight(-1000)
34 ev3.speaker.beep()
35
```

Finalmente también puede ser utilizada para poner imágenes en la pantalla, que nosotros la hemos utilizado para el sensor de color que mencionaremos en el apartado 5.5.4. También se puede comprobar el estado de la batería y su carga actual.

EV3 Devices

Con esta librería pasamos a utilizar los motores y sensores del kit de Lego Mindstorm EV3. Esta librería será fundamental para el movimiento y el funcionamiento de los sensores (los cuales detallaremos en el apartado 5.5.4. Para hablar de esta librería necesitaremos saber qué unidades maneja el robot:

- Tiempo: milisegundos (ms)
- Ángulo: grados (deg)
- Velocidad angular: grados por segundos (deg/s)
- Distancia y dimensiones (radio de las ruedas): milímetros (mm)
- Distancia relativa (sensores): porcentaje (%)

-Frecuencia(altavoz): Herzios (Hz)
-Corriente(batería): miliamperios (mA)
Sabiendo esto, esta librería tiene dos grandes partes, sensores y motores. Los motores igualmente se dividirán en tres partes: Medida (Measuring): que puede medir la velocidad del motor y su velocidad angular. Parando (Stopping). Hace que el movimiento se reduzca de dos maneras, desacelerando poco a poco o repentinamente. Y finalmente Acción (Action) que hace que el motor haga acciones. Estas acciones como por ejemplo avanzar, girar etc...

```
30 robot.straight(1000)
31 ev3.speaker.beep()
32
33 robot.straight(-1000)
34 ev3.speaker.beep()
35
36 robot.turn(360)
37 ev3.speaker.beep()
38
39 robot.turn(-360)
40 ev3.speaker.beep()
41
```

El apartado de sensores lo proseguiremos en el apartado e, ya que es más complejo.

Robotics

Esta librería también es muy importante ya que con esto le daremos la información del robot que hemos construido alrededor del Bloque Inteligente Ev3. La función más utilizada es la DriveBase(left motor, right motor, wheel diameter, axle track). Esta función se desarrollará más en el apartado d.2.

```
robot = DriveBase(left_motor, right_motor, wheel_diameter=56, axle_track=192)
```

Parameters and Constants

En parámetros y constantes encontraremos los parámetros que

necesitamos introducir a nuestro programa para que lo entienda. En este apartado le diremos al Bloque Inteligente donde se encuentran los diferentes “devices” para que ejecute las órdenes dadas a través de las librerías de EV3 devices. Aquí es importante nombrar donde están ubicados los motores y los sensores (que hablaremos más en profundidad en el bloque de d.2)

```
23 left_motor = Motor(Port.B)
24 right_motor = Motor(Port.C)
25
18 front_line_sensor = ColorSensor(Port.S1)
```

tools

En la librería de tools encontraremos las herramientas para poder trabajar con el tiempo. Esto es muy útil para en vez de utilizar las medidas de mm en las acciones de los motores, utilizar el tiempo. También puede medir cuánto tiempo tarda el programa en realizar acciones.

media

Esta librería hace de complemento para *Programmable Hubs*, y su apartado de la pantalla. Te da la posibilidad de proyectar iconos ya hechos y de cambiar la fuente de texto. También hace uso de archivos de sonido que son muy útiles al trabajar con sensores y no puedes estar mirando constantemente a la pantalla.

Declaración de las librerías

El primer paso de todo proyecto de python es declarar las librerías que vamos a utilizar. Hay dos formas de hacer este paso, declarando toda la librería activa, o utilizando la parte de la librería que te interesa.

```
from pybricks.hubs import EV3Brick
from pybricks.ev3devices import Motor
from pybricks.parameters import Port
from pybricks.robotics import DriveBase
```

En este caso estamos declarando a la librería *Programmable Hubs* que importamos solo el Bloque Inteligente. De la librería *EV3 Devices* hemos importado el motor para hacer que el robot se mueva, si hubiéramos requerido también algún sensor también lo hubiéramos puesto aquí. De parámetros hemos importado solo los motores y finalmente damos la información del robot con la librería *Robotics*.

Aquí vemos como podemos importar más de una parte de la librería.

```
from pybricks.ev3devices import Motor, ColorSensor
from pybricks.parameters import Port, Color
```

Hemos importado el sensor de color y los parámetros de color para que el robot sepa que color esta viendo cuando recibe los datos del sensor.

De esta otra manera vemos como cargamos toda la librería. El * se puede sustituir por un “todo”.

```
from pybricks.ev3devices import *
from pybricks.parameters import *
from pybricks.robotics import *
```

Cuando requerimos de una librería en el código escribimos la función de esta que queremos utilizar después del objeto de esa acción.

```
35 robot.straight(1000)
36 robot.turn(360)
37 robot.turn(-360)
```

Aquí hacemos que el robot utilice la librería *EV3 Devices* para mandar la instrucción de girar y de ir recto. Recordamos que la librería *EV3 Devices* es la encargada de los motores y de los sensores.

Utilización de variables y relación con el álgebra

Las variables no son únicamente utilizadas en python, sino en toda la programación. Las variables son palabras, letras o números que contienen información dentro de ella.

```
14 left_motor = Motor(Port.A)
15 right_motor = Motor(Port.B)
```

En esta imagen vemos que hemos denominado a “left_motor” a un motor cualquiera que está en el puerto A del bloque inteligente. Al utilizar esto ya no necesitamos estar todo rato nombrando al motor que está en el puerto A. Esto es lo mismo que utilizar un sistema de ecuaciones, resolviendo con el método de sustitución, pero al revés ya que partimos de la $x = 5 + 3y$ para luego utilizar la x , mientras que en el sistema de ecuaciones como el siguiente se omite la x para poner lo que equivale. Esto se hace para evitar errores al utilizar muchas palabras.

$$2(5 + 3y) + y = 3$$

$$10 + 6y + y = 3$$

$$10 + 7y = 3$$

$$y = \frac{3 - 10}{7}$$

$$y = \frac{-7}{7} = -1$$

```
robot = DriveBase(left_motor, right_motor, wheel_diameter=56,
```

Aquí continuando con el ejemplo anterior se sustituye left_motor por Motor(Port.A) e igualmente right_motor por Motor(Port.B)

Utilización de estructuras y bucles de control o condicionales.

Las estructuras de control permiten modificar el flujo de ejecución de las instrucciones de un programa. Esto quiere decir que dependiendo que son una función que hará si se ejecuta una parte del código.

Python te da varios recursos para modificar el flujo. Vamos a ver los tres que más hemos utilizado en este proyecto.

```
if front_line_sensor.color() == Color.BLACK:
    ev3.speaker.beep()
    robot.straight(-10)
    robot.turn
```

En este ejemplo podemos observar el uso de la estructura de control “if”, que se puede traducir al español por “si”. una variable (explicado en el punto 5.4.1.2) que en este caso es el sensor anterior de nuestro robot, detecta que si el color que esta observando es negro de retroceder 10mm, y posteriormente girar.

El siguiente metodo es