



GRADO EN INGENIERÍA EN TECNOLOGÍA DE LAS TELECOMUNICACIONES

Curso Académico 2021/2022

Trabajo Fin de Grado

ROBÓTICA EDUCATIVA CON PYTHON Y MBOT

Autor : Eva García Domingo
Tutor : Jose María Cañas Plaza

Índice general

1. Introducción	1
2. Objetivos	3
3. Infraestructura	5
3.1. Entorno	5
3.2. Hardware	6
3.2.1. Placa Arduino	6
3.2.2. MBot	6
3.2.2.1. Sensores	8
3.2.2.2. Actuadores	9
3.3. Software	11
3.3.1. Scratch y mBlock	11
3.3.2. Arduino	14
3.3.3. Python	14
4. Plataforma PyBoKids	15
4.1. Diseño	15
4.2. Programa residente	15
4.3. Resultado: programa PC y programa residente	15
5. Aplicación educativa	17
6. Conclusiones	19
Bibliografía	21

Capítulo 1

Introducción

Esto es un ejemplo. Ver en [1] [2]

Capítulo 2

Objetivos

Esto es un ejemplo. Ver en [1] [2]

Capítulo 3

Infraestructura

En este capítulo describiremos la infraestructura utilizada, tanto software como hardware, detallando los pasos a seguir si se desea emular. En caso de que algún componente haya sido elegido entre otros de igual aplicación, expondremos las razones de la elección.

3.1. Entorno

Teniendo en cuenta el carácter educativo de este Trabajo, y su pretendida aplicación en estudiantes de Educación Primaria, se ha elegido el sistema operativo Windows, un entorno conocido, amigable y fácilmente accesible, para instalar y utilizar las diferentes herramientas software.

Tradicionalmente, cuando se hablaba de programación y especialmente de programación robótica, siempre se ha utilizado el sistema operativo Linux. Éste daba la posibilidad de instalar todos los paquetes de lenguajes de programación y entornos, mientras que Windows era especialmente cerrado en cuanto a lenguajes no nativos (fuera de *bash* o de *visual basic*, se hacía complicado utilizar un lenguaje de programación sin acabar recurriendo a virtualizar una máquina Linux), y los entornos software (aplicaciones) de terceros diseñados para programar habitualmente no tenían una versión instalable para Windows.

Sin embargo, los últimos años el Sistema Operativo se ha abierto a esta operativa, ya que la política popular demandaba poder utilizarlo, siendo el sistema operativo más utilizado por usuarios, como herramienta de desarrollo también a nivel usuario. La nueva línea de comandos de Windows, **Powershell** (también lenguaje de scripting), añadió a la original *Bash* características nativas de Linux, siendo una herramienta de programación además de administración. De hecho, en la última versión, está disponible para el propio SO de Linux (en algunas distribuciones).

Al principio de este Trabajo, se consideró a Linux un entorno menos amigable para usuarios sin experiencia en programación, y de la edad comentada anteriormente, por ser considerado "no de usuario": en el poco probable caso de que un estudiante conociera el sistema, lo consideraba algo muy complejo y no accesible para su nivel. Por tanto, elegir Windows eliminaba ese prejuicio y contaba con la ventaja de predisponer positivamente al alumno y de facilitarle el acceso al entorno.

3.2. Hardware

En esta sección describiremos los componentes hardware utilizados. La razón de la elección de éstos responde a la misma filosofía que en la sección anterior, la facilidad de acceso a los componentes y la facilidad con que se complementa en el entorno.

3.2.1. Placa Arduino

Las placas Arduino son las más extendidas en cuanto a robótica. Arduino nació como una solución barata con el principal objetivo de utilizarlo en Educación. Además, al ser un proyecto liberado al público, su uso está extendido a toda una comunidad, que amplía y comparte sus propios desarrollos.

Estas placas son hardware libre (uno de las principales razones de su bajo coste), y contienen un procesador re-programable y una serie de pines hembra, donde se conectarán los periféricos de entrada/salida necesarios para controlar un robot. Hay diferentes modelos de placas Arduinos, cada una fabricada con un propósito diferente; en este caso, hemos utilizado el modelo mCore, basado en [Arduino Uno](#), ya que es la que lleva por defecto el robot educativo Mbot (que comentaremos en la sección 3.2.2). Además de los pines hembra, o puertos, contiene una serie de actuadores y sensores integrados en la placa.

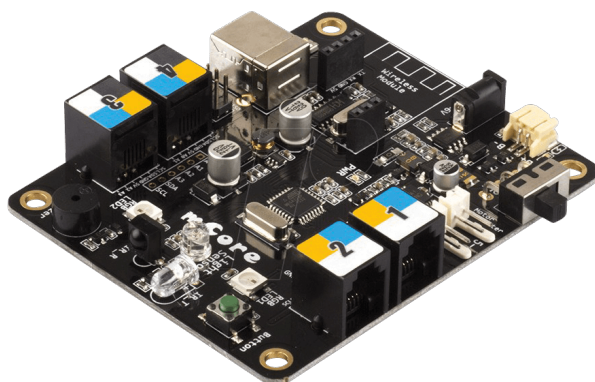


Figura 3.1: Placa mCore

3.2.2. MBot

Actualmente hay una gran variedad de robots educativos, orientados principalmente a los alumnos más jóvenes. El objetivo de la robótica educativa es ofrecer un entorno amigable y divertido, además de *más realista* que la programación tradicional. Es más fácil para un niño o niña sentirse interesado por algo que tiene una reacción *visible*, en algo que puede tocar y manejar, que en la programación normal que, aunque produce una reacción, es mucho más abstracta. Esto convierte la robótica en la herramienta perfecta para enseñar lenguajes, conceptos básicos de programación como funciones, uso de variables, y avanzar hasta la algoritmia se hace un proceso natural al que los propios alumnos llegan ellos mismos.

El principal problema al que nos enfrentamos para enseñar programación a los más jóvenes son los lenguajes de programación. Son poco intuitivos y legibles, sumado a la dificultad de mantener la atención de un alumno o alumna tan joven escribiendo en un ordenador en estos lenguajes. Era necesaria, por tanto, una forma de enseñar estos conceptos de programación

comentados anteriormente sin tener que utilizar lenguajes clásicos de programación, al menos para los alumnos más jóvenes. La solución fue la programación por bloques de *pseudocódigo*. El pseudocódigo es un *framework*, una carcasa, que recubre el verdadero lenguaje de programación de la placa del robot, y lo hace legible y entendible. Además, siendo esto una de las grandes ventajas añadidas, la mayoría de los pseudocódigos están en casi todos los idiomas, permitiendo a los alumnos programar en su propio idioma.

Esto nos lleva al robot Mbot. Está basado en una placa Arduino Uno, y pensado especialmente para la enseñanza: los diferentes sensores y actuadores (los que vienen en el paquete básico, aunque hay muchos más posibles del mismo fabricante) están pensados para ofrecer prácticas entretenidas y, lo más importante, con diferentes niveles de dificultad, por lo que se puede utilizar con alumnos de diferentes edades y, durante un mismo curso escolar, crear prácticas con las que evolucionar en los conceptos.



Figura 3.2: Modelo Mbot utilizado

Este robot Mbot se programa utilizando un lenguaje llamado *Scratch* (3.3.1), un pseudo código muy gráfico y sencillo pero potente, que "recubre" la placa base. Sin embargo, al ser una placa Arduino y por lo tanto, reprogramable, siempre se puede programar en el propio lenguaje Arduino. Esto se comentará en profundidad más adelante, en el punto 4.

Como se puede observar en la imagen, el robot cuenta con dos motores conectados a la placa; contiene además cuatro puertos a los que poder conectar diferentes periféricos con los que trabajar (numerados del uno al cuatro), sin contar con los motores. El modelo mostrado es el básico, sin embargo se pueden cambiar los componentes, añadiendo y/o cambiando la estructura base, y crear "otro" robot (por ejemplo, en las figuras siguientes ??). Aunque en este Trabajo de Fin de Grado solo trabajaremos con la versión básica del Mbot, esta posibilidad de añadirle componentes o cambiarlos es muy interesante para los alumnos, ya que trabajan la mecánica y pueden utilizar más tipos de periféricos.



Figura 3.3: Posibles cambios en el Mbot

3.2.2.1. Sensores

En robótica un *sensor* es un periférico de entrada que, conectado a una placa base, recoge información del medio (cantidad de ruido, temperatura, distancia frontal, etc) y la envía a la placa, dejándola disponible para toma de decisiones. La cantidad de información que se pueda recibir sólo depende de la cantidad de sensores que se pueda conectar a la vez a la placa (cuatro, en este caso, si solo se trabajara con sensores y ningún actuador). En el robot básico (de la imagen ??) los sensores son:

Sensor de ultrasonidos Está colocado en el frente, y recoge información de **distancia** hasta un objeto, en *cm* (el valor máximo es 400)

Sensor infrarrojo Sigue Líneas Está ubicado de tal forma que lea, del "suelo", si el sensor está tapado o no, es decir: está sobre blanco o negro (de fondo, es un sensor binario). Se compone en realidad de dos sensores, izquierdo y derecho, por lo que habrá cuatro posibilidades, cada una codificada con un valor numérico (el valor que devuelve el sensor a la placa Arduino):

- Ningún sensor tapado: 0
- Sensor izquierdo tapado y derecho no: 1
- Sensor derecho tapado e izquierdo no: 2
- Los dos sensores tapados: 3

Sensor de Luz En este caso, está integrado en la placa, por lo que no es necesario utilizar un puerto para él. Nos da un valor de cantidad de luz en el ambiente, pudiéndolo utilizar para saber si hay más o menos luminosidad de la deseada. Por supuesto, para este valor "deseado" será necesario obtener un valor inicial de la habitación en la que nos encontremos, para poder establecer ese valor barrera (*threshold*)



(a) Ultrasonidos: sensor de distancia



(b) Infrarrojos: sensor siguelíneas



(c) sensor de luz integrado

Figura 3.4: Sensores

3.2.2.2. Actuadores

La definición de *actuador* es un periférico de salida al que la placa envía datos con los que éste realiza una acción de una forma u otra. Por ejemplo, teniendo una velocidad v_0 , este valor es enviado a los motores, que se moverán a esa velocidad y no a otra.

Los actuadores, los que no están integrados en la placa directamente, se conectan a la placa a través de los mismos puertos que los sensores. Para poder usarlos, habrá que especificarle a la placa en qué puerto están conectados (igual que con los sensores). En nuestro robot básico, tenemos los siguientes actuadores:

Leds . Están integrados en la placa, en la parte superior, y compuesto por dos led individuales que poder combinar dependiendo de qué valor codificado se envíe a la placa:

- Los dos led: 0
- Sólo led derecho: 1
- Sólo led izquierdo: 2

Los led son RGB (*[red, green, blue]*), y el color de los dos led se codifica con un valor entre 0 y 255 para cada uno de los colores rojo, verde y azul. Así, por ejemplo, el rojo completo sería [0,255,0], el morado sería [255,0,255], el negro [0,0,0] o el blanco [255,255,255]. Estos led están codificados con valores decimales, en vez de hexadecimales como sería una codificación RGB tradicional, para facilitar la programación a los alumnos, que no conocerían el sistema hexadecimal.

americana. La equivalencia, que los alumnos necesitan conocer, es la siguiente:

Europea	Americana
Do	C
RE	D
MI	E
FA	F
SOL	G
LA	A
SI	B

Para utilizar diferentes notas más agudas o más graves, se utilizan números a continuación de las letras: **C0** es más grave que **C5**. Internamente, el zumbador entiende valores enteros, correspondientes en frecuencia con cada nota.



Figura 3.7: Zumbador integrado

3.3. Software

En esta sección describiremos el diferente software utilizado y el propósito de éste en el marco de este Trabajo Fin de Grado.

3.3.1. Scratch y mBlock

Como se ha comentado anteriormente, el robot Mbot es programable con Scratch, un lenguaje de **programación por bloques**. Un *bloque de código* consiste en codificar en un "paquete" una sentencia completa de lenguaje (del lenguaje correspondiente, Arduino en este caso). Así, el estudiante que utilice Scratch, será capaz de utilizar una sentencia *if..else* o un bucle *for* de forma muy fácil y entendible, sin necesitar aprenderse todas las reglas de sintaxis del lenguaje real. A continuación, se muestran algunos ejemplos de bloques en Scratch correspondientes a los conceptos de programación más utilizados:



Figura 3.8: Algunos ejemplos de bloques en Scratch

Como puede observarse, las sentencias que se quieran repetir, o las variables de condición, tienen un lugar muy intuitivo donde colocarse. Además, pueden crearse variables, en las que guardar los valores de los sensores y poder utilizar como entrada de actuadores, o como valores de referencia.

Ciertamente, y aunque el lenguaje Scratch puede utilizarse como lenguaje de programación "tradicional", utilizando un escenario virtual con un personaje para observar el resultado del programa (sin robot físico), es mucho más completo al añadirle el módulo del robot deseado. Este módulo contiene bloques para recoger valores de los sensores o enviar valores a los actuadores, ya sean "on board" (integrados en la placa), o teniendo que especificar el puerto al que están conectados:

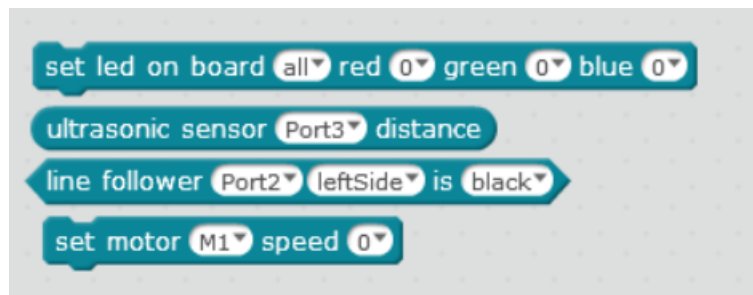


Figura 3.9: Ejemplos de bloques de Mbot en Scratch

Para utilizar Scratch y programar el robot, es necesario instalar un programa, que es el que contiene el compilador y el que permite conectar el robot al PC para enviarle el programa, llamado *mBlock*. El ejecutable se descarga de la página oficial del fabricante, [Makeblock](#), disponible para PC (también existe una versión simplificada para dispositivos móviles).

Una vez instalado el software, el proceso para poder empezar a usar el mBot es muy simple:

1. Conectar el robot al pc con el cable USB, y conectarlo con el programa (el robot debe estar encendido).

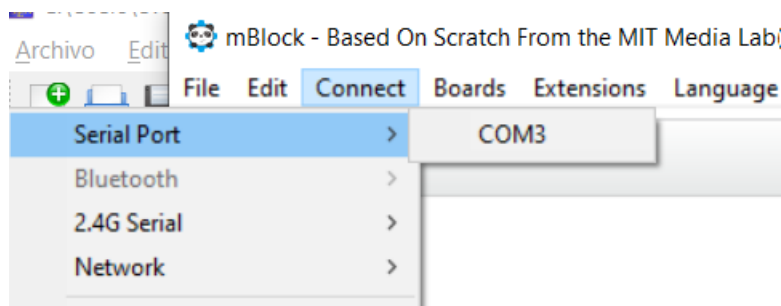


Figura 3.10: Conectar el mBot

2. Asegurarse de que la placa corresponde con el robot que tenemos

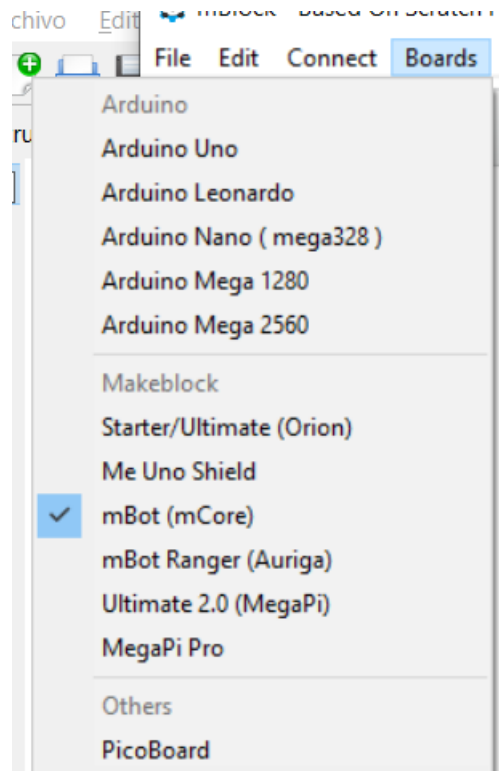


Figura 3.11: Placa mBot

3. Actualizar el programa Arduino subido a la placa base, para que funcione con Scratch (en caso de haber utilizado el robot con otro programa)

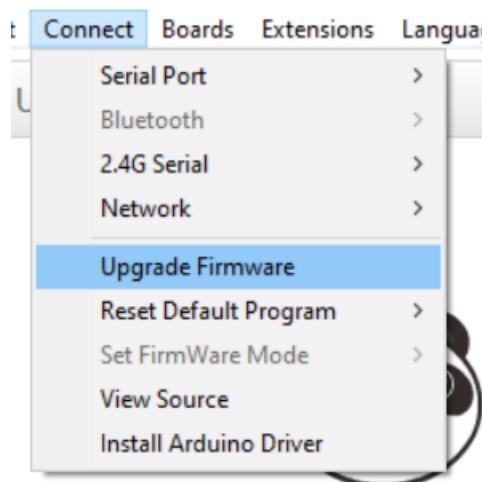


Figura 3.12: Actualizar firmware de la placa

4. Una vez programado algo de código, para subirlo al robot: click derecho sobre el código, 'upload to arduino', para compilar el programa, y otra vez a 'upload to arduino' cuando el compilador aparezca en la parte derecha de la pantalla, para subirlo a la placa

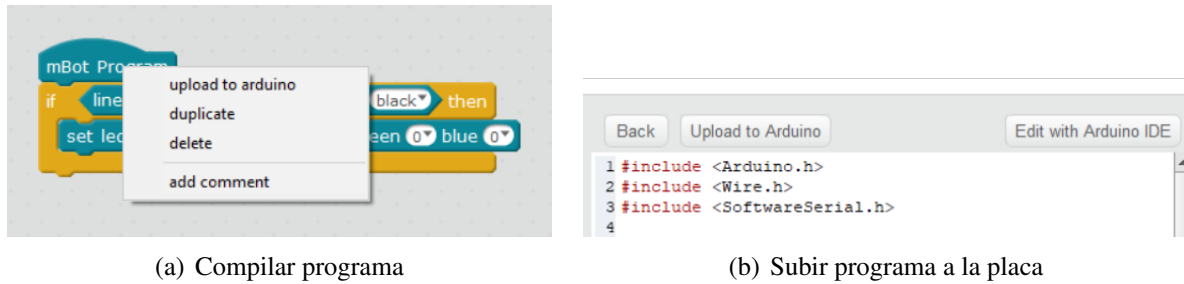


Figura 3.13: Subir programa a la placa del mBot

5. Una vez subido el programa, el robot comenzará a funcionar tal y como lo hayamos programado.

Este lenguaje, y la interfaz gráfica, es una perfecta primera aproximación a

3.3.2. Arduino

3.3.3. Python

Capítulo 4

Plataforma PyBoKids

Esto es un ejemplo. Ver en [1] [2]

4.1. Diseño

4.2. Programa residente

4.3. Resultado: programa PC y programa residente

Capítulo 5

Aplicación educativa

Esto es un ejemplo. Ver en [1] [2]

Capítulo 6

Conclusiones

Esto es un ejemplo. Ver en [1] [2]

Bibliografía

- [1] DARWIN, C. *El origen de las especies por medio de la selección natural*. Editorial CSIC-CSIC Press, 2009.
- [2] DEL VALLE-INCLÁN, R. *Luces de bohemia*. Dirección General de Música y Teatro, 1984.