
Entorno docente con Arduino y Python para Educación Robótica en Secundaria

Jose M^a Cañas, Julio Vega, Francisco Pérez, Aitor Martínez

jmplaza@gsyc.es



eMadrid
www.emadridnet.org



*Seminario sobre "Pensamiento Computacional"
20 enero 2017*

Contenidos

1. Introducción
2. Diseño y plataforma de desarrollo
 - Plataforma hardware Arduino
 - Kit software Arduino IDE
3. Programa educativo
4. Conclusiones

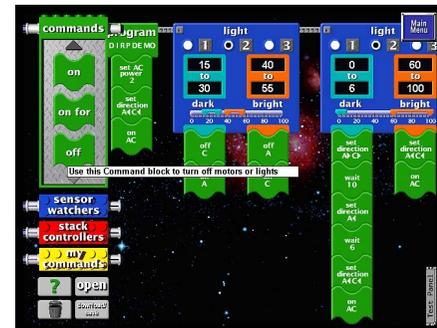
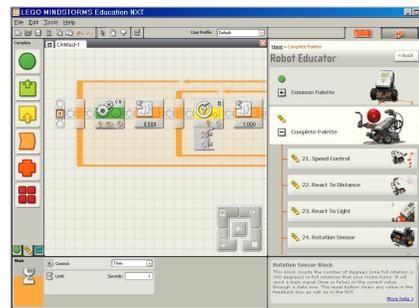
1. Introducción

- La **educación en robótica** en E.S.O. está cobrando mucha importancia
- La Com.Madrid ha introducido una asignatura en el currículum
- **Aplicaciones** en el mercado: aspiradoras, drones, coches autónomos



Creciente importancia de la programación

- **Plataformas:** Lego (RCX, NXT, Ev3, WeDo), mBot o Zowie
- Incluyen placa Arduino con sensores de bajo coste y servos
- Se enseñan **lenguajes** sencillos: RCX-code, Scratch o Blockly
- La funcionalidad reside fundamentalmente en la *programación*



Limitaciones de los kits educativos

- **Scratch o Lego** son ideales para *primeros cursos* de E.S.O.
Aprendizaje inicial casi inmediato: gran entusiasmo
Plataforma muy versátil en su construcción
- Para **cursos posteriores** se quedan cortas

2. Diseño y plataforma de desarrollo

- **Entorno JdeRobot-Niños:** placa Arduino + Python



- ¿Por qué **Arduino**?

Entorno **sencillo**, versátil, completo, potente

Barato, de **hardware libre**, gran compatibilidad

La interacción con hardware real es muy enriquecedora

- ¿Por qué **Python**?

Lenguaje Arduino es versión simplificada de C, sintáxis difícil de aprender por los estudiantes, compilado

Python: más intuitivo y potente, interpretado

Se emplea en la universidad y en muchos ámbitos

- Las prácticas fomentan **aprender haciendo** y motivan
- Robot = hardware + software (programas)
- Robot conectado al ordenador, donde se ejecuta el código

Ordenador Personal



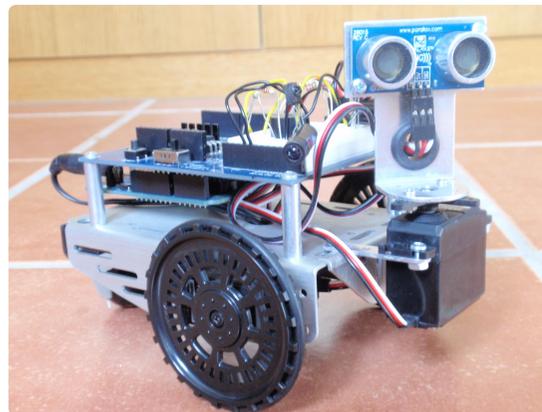
Robot



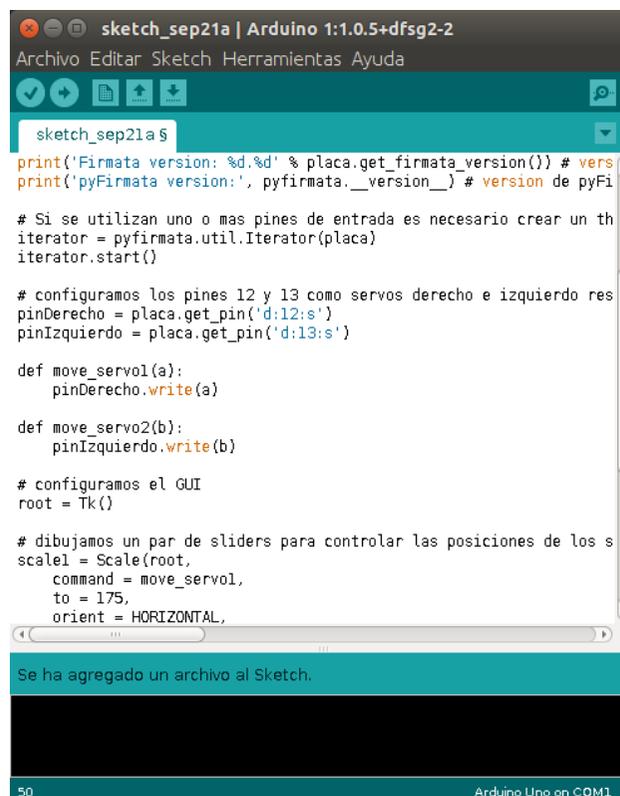
USB/Wifi/Bluetooth

2.1 Plataforma hardware Arduino

- **Procesador:** placa Arduino UNO, cerebro del robot
- **Sensores:** captan información del mundo que les rodea
- **Actuadores:** permiten actuar sobre el entorno



2.2 Kit software Arduino IDE



```
sketch_sep21a | Arduino 1:1.0.5+dfsg2-2
Archivo Editar Sketch Herramientas Ayuda
sketch_sep21a $
print('Firmata version: %d.%d' % placa.get_firmata_version()) # vers
print('pyFirmata version:', pyfirmata.__version__) # version de pyFi

# Si se utilizan uno o mas pines de entrada es necesario crear un th
iterator = pyfirmata.util.Iterator(placa)
iterator.start()

# configuramos los pines 12 y 13 como servos derecho e izquierdo res
pinDerecho = placa.get_pin('d:12:s')
pinIzquierdo = placa.get_pin('d:13:s')

def move_servo1(a):
    pinDerecho.write(a)

def move_servo2(b):
    pinIzquierdo.write(b)

# configuramos el GUI
root = Tk()

# dibujamos un par de sliders para controlar las posiciones de los s
scale1 = Scale(root,
    command = move_servo1,
    to = 175,
    orient = HORIZONTAL,
```

Se ha agregado un archivo al Sketch.

50 Arduino Uno on COM1

- Para usar con Python hacemos uso de la librería **pyFirmata**

3. Programa educativo

- Prácticas escalonadas en complejidad, de menos a más.
 1. Nociones básicas de **programación de ordenadores**, pseudocódigo
 2. Conceptos y claves teóricas del lenguaje **Python**
 3. Prácticas robóticas con **sensores y actuadores** de Arduino
 4. **Proyecto** robótico completo: comportamiento de un robot
- Se está utilizando ya (60 alumnos)
 - Colegio Nuestra Señora del Sagrado Corazón (4º de la E.S.O.)
 - Colegio Villa de Móstoles (extraescolar, toda ESO)

3.1 Nociones básicas de programación

- Comprender la forma de trabajar de un **ordenador** a nivel interno
- Entender la utilidad de **variables** o funciones
- Aterrizar en conceptos como **bucles** o condicionales
- Lenguaje de **Pseudocódigo**

Interiorizar estructura, organización y restricciones de un lenguaje

Nuevos **conceptos**: contador, uso de conjuntos o vectores

3.2 Lenguaje Python

- Conocer **palabras** clave del lenguaje en cuestión
- Matizar cuestiones **sintácticas** y propias de este lenguaje
- Realizar algunos **ejercicios clásicos** de iniciación a la programación:
Ej.: **programa** que imprima la suma de los 100 primeros números

3.3 Prácticas con Arduino: sensores y actuadores

- Prácticas más complejas y directamente relacionadas con la robótica
- Repasar conceptos elementales de electrónica
- **Manejo básico de sensores y actuadores desde programa.** Por ejemplo:
 - Uso de **LEDs y zumbador**
 - Lectura de **sensores** complejos: luz, infrarrojos, ultrasonidos
 - Control de **motores**: avance, retroceso y parar
 - Lectura de **botones y potenciómetro**
- <http://jderobot.org/Robotica-en-secundaria>

3.4 Prácticas con Arduino: comportamientos

- Elaborar un **proyecto completo, comportamiento autónomo**
- Programar al robot para que haga algo usando sus sensores y sus actuadores
- Por ejemplo:
 - un **robot que navegue** mientras esquiva obstáculos
- <http://jderobot.org/Robotica-en-secundaria>

4. Conclusiones

- La **robótica es una herramienta educativa** muy útil (STEM)
- La sociedad demanda cada vez más habilidades con robots
- Múltiples plataformas robóticas educativas de fácil iniciación, pero en breve dejan de entusiasmar a los jóvenes
- Entorno educativo **JdeRobot-niños**:
 - **Arduino**, sencillo y potente. Experiencia hardware
 - **Python**, lenguaje sencillo y versátil

Líneas futuras

- Posibilidad de simulador
- Correctores automáticos
- *Gamificación* como juegos competitivos (como RoboCampeones)

