

Lenguajes para programar robots

(P2: T7)

Yeray Méndez Romero
yeray.mendez@udc.es

Daniel Rivera López
d.rivera1@udc.es

ÍNDICE

- Introducción.
- Programación de robots:
 - Nivel gestual.
 - Nivel textual.
- Relación de los lenguajes de programación de robots y los habituales.
- Lenguajes en empresas.
- Lenguajes en educación.
- Conclusión
- Bibliografía.

INTRODUCCIÓN

- ¿ Qué es un robot ?



PROGRAMACIÓN DE ROBOTS

- Existen dos niveles de programación:

- Gestual



FIG.17

- Textual

```
1  ' Globales -----
2  Var Variable0:Booleano
3  Var Variable1:Cadena
4  ' Fin Globales -----
5  Proc Procedimiento ' <- Procedimiento sin retorno.
6    Var Variable2:Entero ' Locales
7    Var Variable3:Real
8
9    Si Variable0 = Falso Entonces ' Condición "If"
10     Contar Variable2 = 0 a 9 ' Bucle "For"
11     Variable1 = Variable1 + "1"
12     Seguir ' "End For"
13   FinSi ' "End If"
14
15   Variable3 = 5.13
16 FinProc
```

PROGRAMACIÓN DE ROBOTS

NIVEL GESTUAL

- Gestual:
 - Robot solo recuerda posiciones.
 - No es necesaria la experiencia.



PROGRAMACIÓN DE ROBOTS

NIVEL TEXTUAL

- Textual:
 - Necesaria experiencia en programación.
 - Tres tipos:
 - Nivel robot
 - Nivel objeto
 - Nivel tarea

PROGRAMACIÓN DE ROBOTS

NIVEL TEXTUAL

- Nivel robot:
 - Ejemplos:
 - RAPID(Robotics Application Programming Interactive Dialogue).



PROGRAMACIÓN DE ROBOTS

NIVEL TEXTUAL

- Nivel objeto:
 - Ejemplos:
 - RAPT y AUTOPASS utilizados en el ensamblaje de piezas.
 - Ejemplo de código AUTOPASS:

```
PLACE C1  
SUCH THAT C1 BOT CONTACTS C2TOP  
AND B1 A1 IS ALIGNED WITH C2A1  
AND B1 A2 IS ALIGNED WITH C2A2
```


PROGRAMACIÓN DE ROBOTS

NIVEL TEXTUAL

- Nivel tarea:
 - Ejemplos:
 - STRIPS: lenguaje formal utilizado por el planificador con el mismo nombre por el primer robot que podía razonar sobre sus acciones, Shakey.

```
Initial state: At(A), Level(low), BoxAt(C), BananasAt(B)
Goal state:   Have(bananas)
```

Actions:

```
// move from X to Y
_Move(X, Y)_
Preconditions: At(X), Level(low)
Postconditions: not At(X), At(Y)

// climb up on the box
_ClimbUp(Location)_
Preconditions: At(Location), BoxAt(Location), Level(low)
Postconditions: Level(high), not Level(low)

// climb down from the box
_ClimbDown(Location)_
Preconditions: At(Location), BoxAt(Location), Level(high)
Postconditions: Level(low), not Level(high)

// move monkey and box from X to Y
_MoveBox(X, Y)_
Preconditions: At(X), BoxAt(X), Level(low)
Postconditions: BoxAt(Y), not BoxAt(X), At(Y), not At(X)

// take the bananas
_TakeBananas(Location)_
Preconditions: At(Location), BananasAt(Location), Level(high)
Postconditions: Have(bananas)
```

RELACIÓN DE LOS LENGUAJES DE PROGRAMACIÓN DE ROBOTS Y LOS HABITUALES

- Dos tipos fundamentales:
 - Extensiones de los lenguajes habituales



- Lenguajes diseñados específicamente para programar robots.

RAPID

KAREL

V+

LENGUAJES EN EMPRESAS

- Ámbito empresarial: Muchos lenguajes.
- Nos basamos en la lista anual RBR50(Robotics Business Review) en la de este año encontramos a :
 - ABB con el lenguaje RAPID.
 - Fanuc con el lenguaje KAREL.
 - Adept con V+.

LENGUAJES EN EMPRESAS

- RAPID:
 - Lenguaje de programación textual de alto nivel.
 - Utilizado actualmente en la controladora IRC5 de ABB.

LENGUAJES EN EMPRESAS

- RAPID:

```
%%%  
VERSION:1  
LANGUAGE:ENGLISH  
%%%  
MODULE EJEMPLO  
  CONST robtarget A:=[[0,0,0],[0,0,0,0],[0,-1,0,0], [9E+09,...]];  
  
  CONST tooldata pinza:= [TRUE, [[0,0,0],[1,0,0,0]],  
                           [0,[0,0,0],[1,0,0,0],0,0,0]];  
  
  PROC cerrar_pinza()  
    Set spinza;  
  ENDPROC  
  
  PROC coger_pieza()  
    MoveJ B1,v100,z5,pinza;  
    MoveL B,v80,fine,pinza;  
    cerrar_pinza;  
  ENDPROC
```

```
  PROC main()  
    CONST dionum listo:=1;  
    abrir_pinza;  
    WHILE TRUE DO  
      MoveJ A,v100,fine,pinza;  
      WaitDI econtrol,listo;  
      coger_pieza;  
      MoveL B1,v80,z5,pinza;  
      MoveJ D,v100,z100,pinza;  
      MoveJ C1,v100,z5,pinza;  
      MoveL C,v80,fine,pinza;  
      abrir_pinza;  
      MoveL C1,v80,z5,pinza;  
    ENDWHILE  
  ENDPROC  
ENDMODULE
```

LENGUAJES EN EMPRESAS

- Ámbito empresarial: Muchos lenguajes.
- Problema:
 - Cada empresa solo tiene en cuenta sus necesidades en cuanto a software.
 - El código no es portable.

LENGUAJES EN EMPRESAS

- RESULTADO :
 - Usuario final se queda bloqueado en un determinado fabricante.
 - Pérdida de tiempo y económica.



LENGUAJES EN EMPRESAS

- ROS-Industrial
 - Miembros :

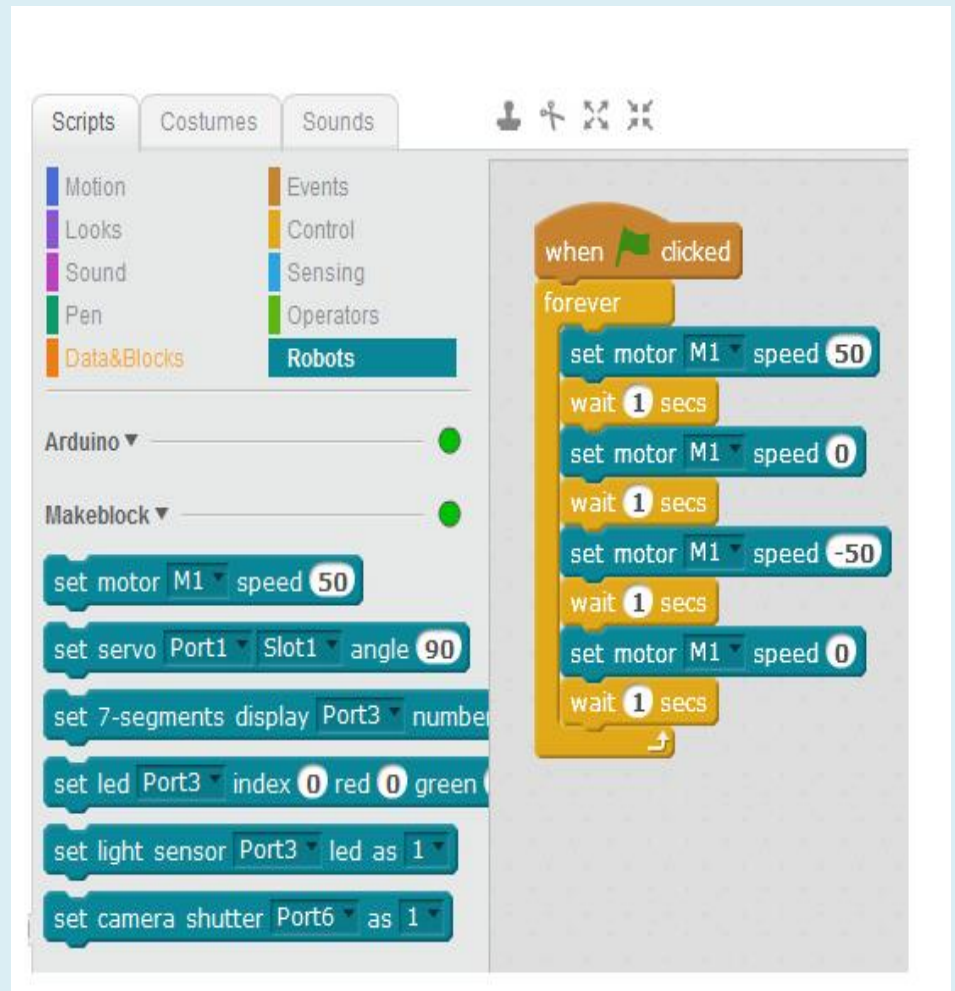


LENGUAJES EN EDUCACIÓN

- Ejemplos:
 - Scratch.
 - Arduino.
- Suelen emplearse lenguajes habituales como:
 - Matlab.
 - Python.
 - C/C++.
 - JavaScript.

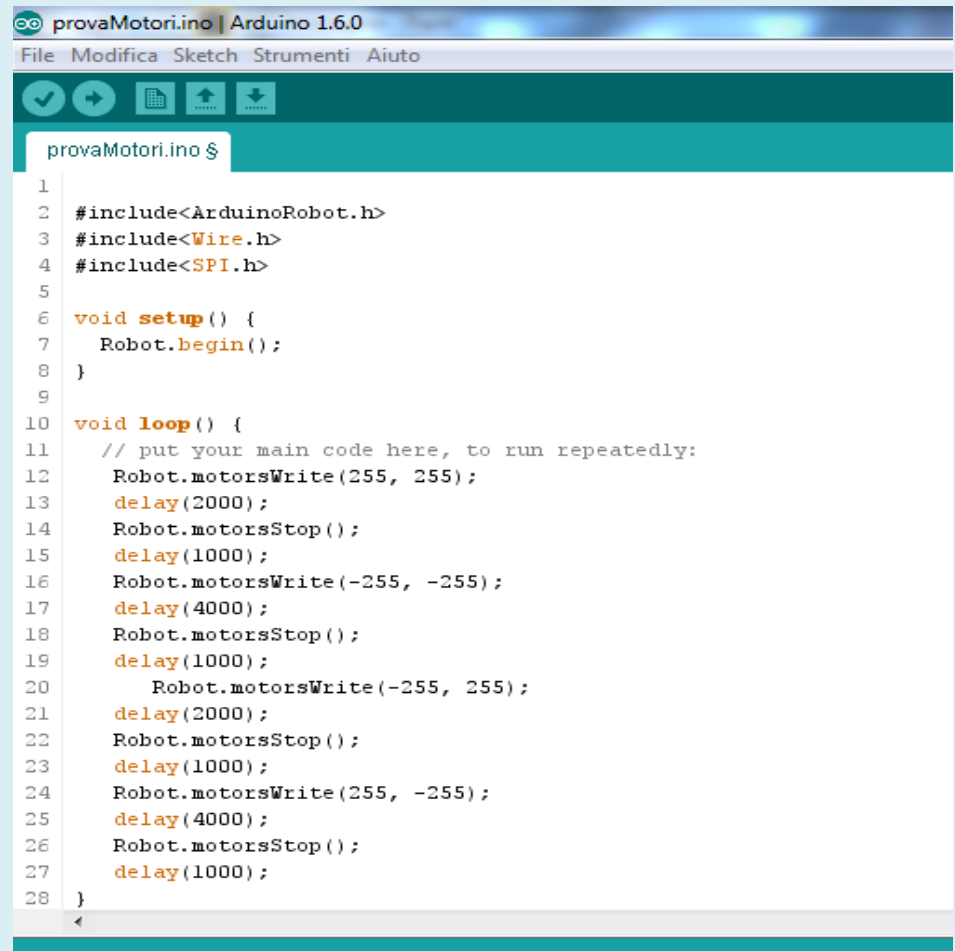
LENGUAJES EN EDUCACIÓN

- Scratch:
 - Lenguaje de programación visual y educativo.
 - Desarrollado por Lifelong Kindergarten Group en el MIT .
 - Paradigma imperativo.
 - Dirigido por eventos.
 - Primera versión en 2003.



LENGUAJES EN EDUCACIÓN

- Arduino:
 - Plataforma electrónica open-source basada en hardware y software sencilla de usar.
 - Es simplemente el lenguaje C/C++ con librerías añadidas.
 - Permite programación orientada a robots aunque normalmente se utilizan interfaces visuales, este es el caso de Scratch4Arduino y Mblock.



```
provaMotori.ino | Arduino 1.6.0
File Modifica Sketch Strumenti Aiuto

provaMotori.ino §
1
2 #include<ArduinoRobot.h>
3 #include<Wire.h>
4 #include<SPI.h>
5
6 void setup() {
7     Robot.begin();
8 }
9
10 void loop() {
11     // put your main code here, to run repeatedly:
12     Robot.motorsWrite(255, 255);
13     delay(2000);
14     Robot.motorsStop();
15     delay(1000);
16     Robot.motorsWrite(-255, -255);
17     delay(4000);
18     Robot.motorsStop();
19     delay(1000);
20     Robot.motorsWrite(-255, 255);
21     delay(2000);
22     Robot.motorsStop();
23     delay(1000);
24     Robot.motorsWrite(255, -255);
25     delay(4000);
26     Robot.motorsStop();
27     delay(1000);
28 }
```

CONCLUSION

- Similitudes:
 - No existe un cambio en la variedad de tareas a realizar en cada uno de los lenguajes mencionados.
- Diferencias:
 - Eficiencia del código y simplicidad(en los lenguajes de empresa).

BIBLIOGRAFÍA

- <https://es.wikipedia.org/wiki/Robot>
- <https://analisisyprogramacionoop.blogspot.com.es/2014/10/programacion-de-robots.html>
- http://platea.pntic.mec.es/vgonzale/cyr_0708/archivos/_15/Tema_5.6.htm
- <https://blog.robotiq.com/what-is-the-best-programming-language-for-robotics>
- https://en.wikipedia.org/wiki/Educational_robotics
- <http://petercorke.com/wordpress/toolboxes/robotics-toolbox>
- <https://www.makeblock.es/soporte/mblock/>
- <http://codigo21.educacion.navarra.es/autoaprendizaje/primeros-pasos-con-scratch-y-lego-wedo/>
- <http://www.onerobotics.com/posts/2013/introduction-to-karel-programming/>
- <http://www.monografias.com/trabajos3/progrob/progrob.shtml>
- <http://rosindustrial.org/>
- https://www.roboticsbusinessreview.com/companies/?companyType=rbr_50
- <https://en.wikipedia.org/wiki/RAPID>
- <http://hopl.info/showlanguage.prx?exp=7482>
- [https://wikivisually.com/lang-de/wiki/ARLA_\(Programmiersprache\)](https://wikivisually.com/lang-de/wiki/ARLA_(Programmiersprache))
- <http://www.robot-forum.com/robotforum/fanuc-robot-forum/karel-load-option-in-r30ib/>
- https://en.wikipedia.org/wiki/Adept_Technology
- [https://es.wikipedia.org/wiki/Scratch_\(lenguaje_de_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Scratch_(lenguaje_de_programaci%C3%B3n))
- https://wiki.scratch.mit.edu/wiki/Scratch#Notable_Information
- https://en.wikipedia.org/wiki/Educational_robotics
- <https://www.arduino.cc/en/Guide/Introduction>