

# **Experiencias con plataformas LEGO NXT y Arduino: Nuevas propuestas docentes con plataformas de bajo coste y alta funcionalidad**

**CEA-IFAC  
GT Educación en Automática**

*Eloy Irigoyen, Itziar Cabanes  
Dep. de Ingeniería de Sistemas y Automática  
UPV/EHU*

# Índice

- ¿Por qué pensar en pequeñas plataformas de desarrollo?
  - Aprendizaje de nuevos contenidos
  - Alumnos: capacidades frente a inquietudes
  - Accesibilidad de las nuevas plataformas
- ¿Qué nos ofrecen las pequeñas plataformas de desarrollo?
- ¿Qué se está haciendo desde la UPV/EHU?
  - NXT de LEGO
  - Arduino
  - Otras plataformas
- Conclusiones

# ¿Por qué pensar en pequeñas plataformas de desarrollo?

**Ampliar y mejorar el aprendizaje de contenidos relacionados con Automática e Ingeniería de Sistemas**

Capacidades e **inquietudes** de los alumnos

**Bajo coste** ...hasta para los alumnos

# Aprendizaje de contenidos

- Ampliar y mejorar el aprendizaje de contenidos relacionados con Automática e Ingeniería de Sistemas
  - Transición de planteamientos y desarrollos teóricos a aprendizaje mediante desarrollos reales.
    - Sin eliminar la parte analítica, robustecer el aprendizaje con aplicaciones reales.
  - "Nuevos" modelos de enseñanza.
    - Más allá del PC y las simulaciones.

# Alumnos: capacidades frente a inquietudes

- Hacer converger:  
**capacidades e inquietudes** de los alumnos
  - Las nuevas tecnologías son asimiladas por los alumnos más rápidamente que por los tutores o profesores.
  - La utilización de plataformas reales aumenta la posibilidad de aprendizaje teórico y práctico.

# Accesibilidad de las nuevas plataformas

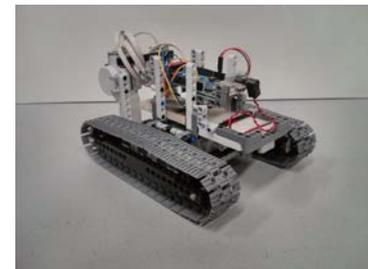
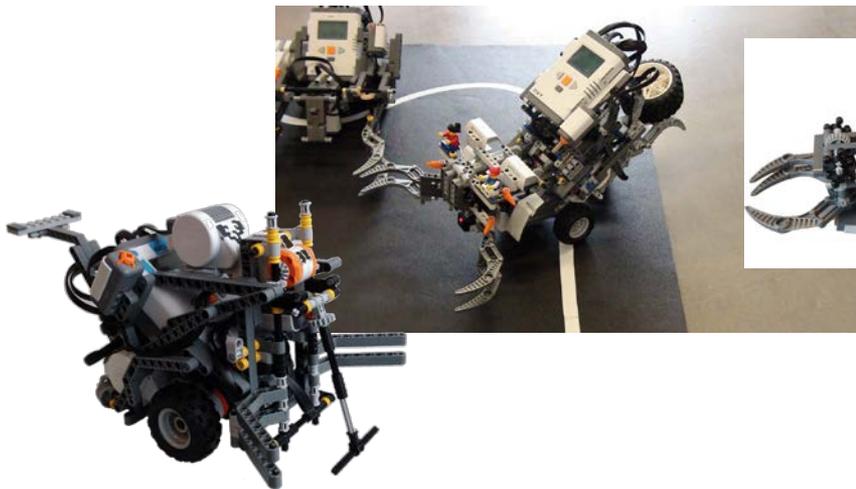
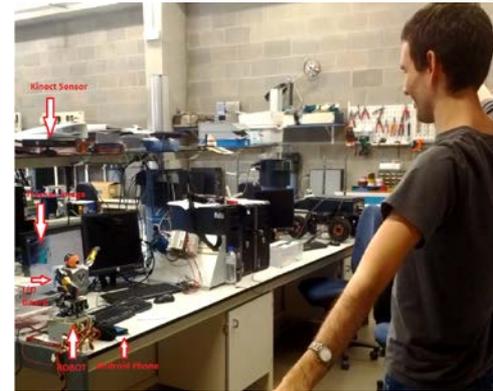
- Bajo coste
  - Inversión en laboratorios o aulas.
    - Complementando a las "clásicas" maquetas.
  - ... hasta para los alumnos.
    - Se da la circunstancia de que antes de proponer trabajos con estas plataformas ya existen alumnos que las han adquirido y desarrollan sus propias aplicaciones "caseras".

# ¿Qué nos ofrecen las pequeñas plataformas de desarrollo?

- Una herramienta para formar a los alumnos en:
  - **Programación**
    - Sencilla
    - Información disponible: Técnicas/comunidades/API
    - Desarrollo/utilización drivers
  - **Procesamiento**
    - Algoritmos de control
  - **Periferia**
    - Dimensionamiento: Desarrollar soluciones con elementos reales
    - Configuración: Parametrización de datos reales
    - Comunicación: Análisis/resolución de problemas reales
- La posibilidad de plantear la adaptación de contenidos teórico/prácticos a nuevas formas de trabajo, mostrando soluciones inmediatas, in situ.

# ¿Qué se está haciendo desde la UPV/EHU?

- Docencia + Investigación



# NXT de LEGO

## ● Presentación del equipo

### ○ LEGO® MINDSTORMS® NXT 2.0

- Microcontrolador ARM7 de 32 bits
- 256 Kb de memoria Flash
- 64 Kb de RAM
- Puertos: 4 I + 3 O
- Comunicación: USB + Bluetooth
- Firmware
- Ejecución: autónoma + remota

### ○ Sensorización especializada

- HiTechnic

### ○ Nuevas unidades EV3



# NXT de LEGO

- Posibilidad de programación

- SW LEGO



- RobotC



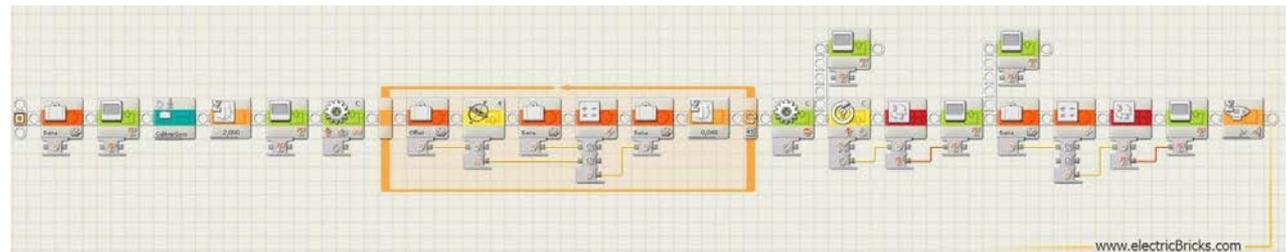
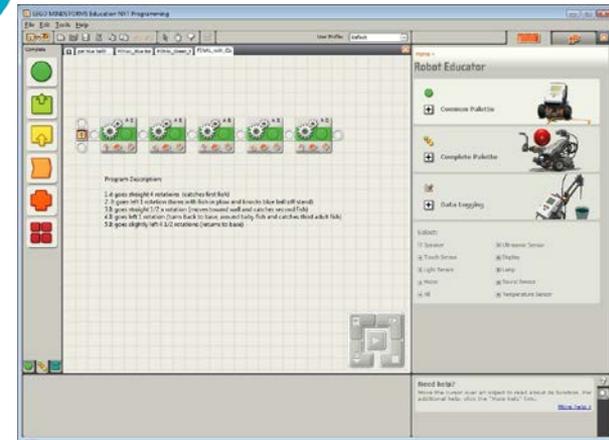
- MATLAB - Toolbox

- NI LabVIEW

- ADA

- JAVA

- ROS



# NXT de LEGO

## ● Posibilidad de programación

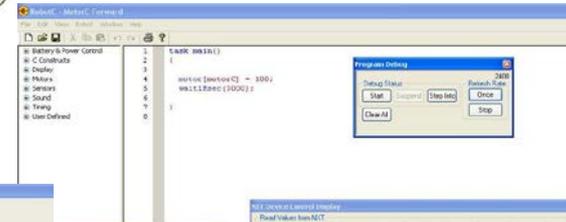
- SW LEGO



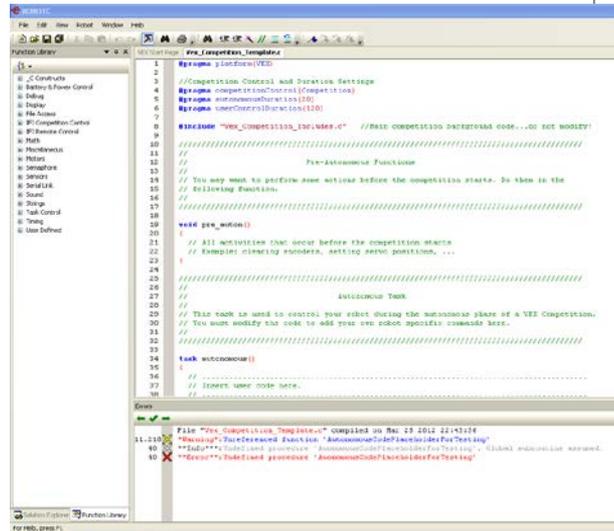
- RobotC



- MATLAB - Toolbox



- NI LabVIEW



- ADA

- JAVA

- ROS

Motor/Speed	PIC Mode	Read/Write	Units	Value	Range	Units	Value
A	E	OFF	rpm	0	0	0	0
B	C	OFF	rpm	0	0	0	0
C	E	OFF	rpm	0	0	0	0

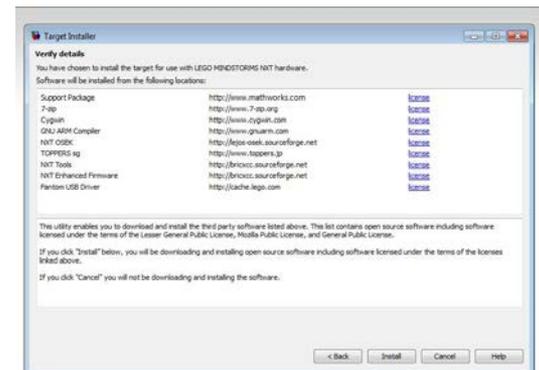
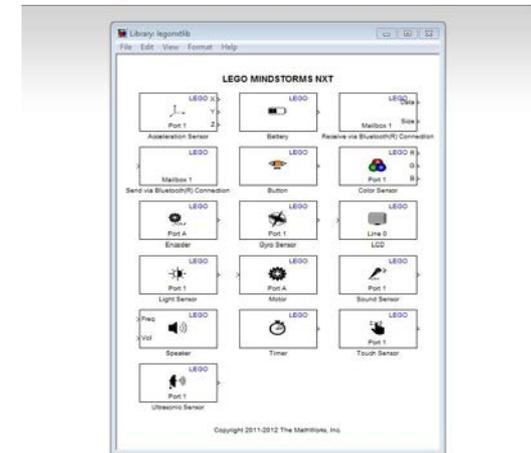
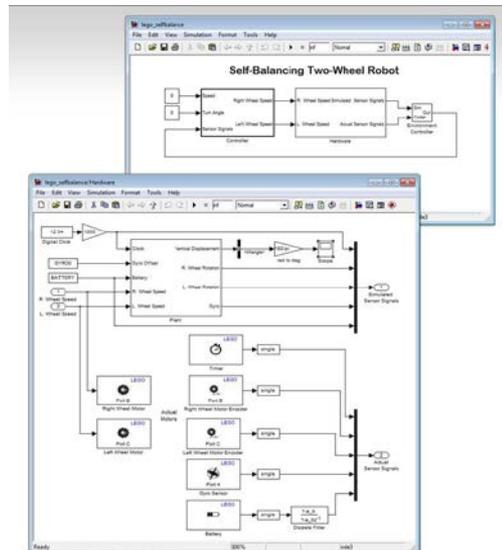
  

Device	Type	Units	Value	Range	Units	Value
11	Raw Value	motor	0	0	rpm	0
12	Raw Value	motor	1000	1000	rpm	0
13	Raw Value	motor	1000	1000	rpm	0
14	Raw Value	motor	1000	1000	rpm	0

# NXT de LEGO

## ● Posibilidad de programación

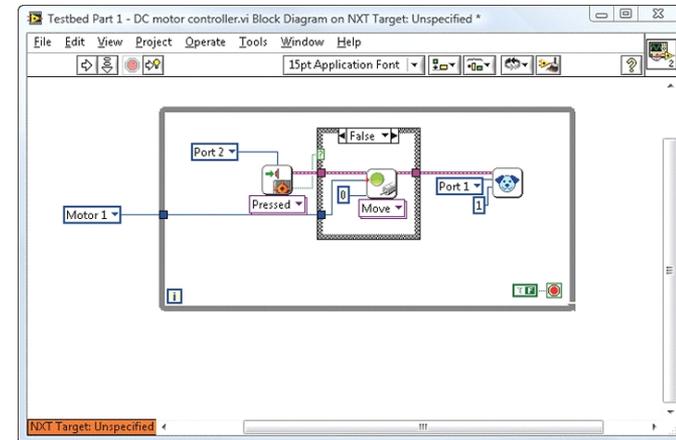
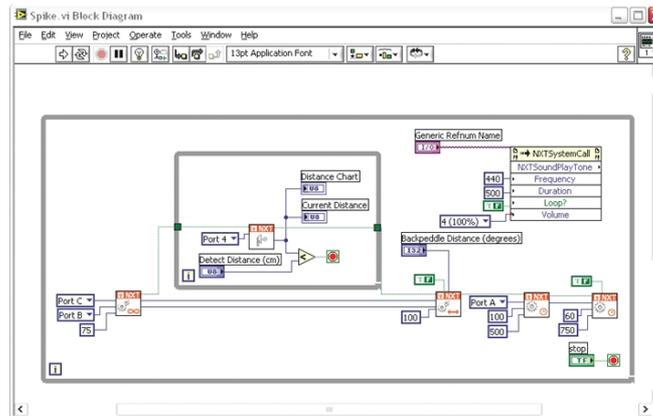
- SW LEGO
- RobotC
- MATLAB/SIMULINK - Toolbox
- NI LabVIEW
- ADA
- JAVA
- ROS



# NXT de LEGO

- Posibilidad de programación

- SW LEGO
- RobotC
- MATLAB - Toolbox
- NI LabVIEW
- ADA
- JAVA
- ROS



# NXT de LEGO

## ● Posibilidad de programación

- SW LEGO
- RobotC
- MATLAB - Toolbox
- NI LabVIEW
- ADA
- JAVA
- ROS



### Light Sensors

```
1 with NXT.Display; use NXT.Display;
2 with NXT.AVR;
3 with NXT.Light_Sensors; use NXT.Light_Sensors;
4 with NXT.Light_Sensors.Ctors; use NXT.Light_Sensors.Ctors;
5
6 procedure light_test is
7   use NXT;
8   Light : Integer;
9   Light_Sensor_1 : Light_Sensor := make(Sensor_1, True);
10 begin
11   loop
12     Clear_Screen_Noupdate;
13     Light := NXT.Light_Sensors.Light_Value (Light_Sensor_1);
14     Put_Noupdate(Light);
15     Screen_Update;
16   end loop;
17 end light_test;
```

### Ultrasonic Sensor

```
with Ada.Real_Time; use Ada.Real_Time;
with NXT.Display; use NXT.Display;
with NXT.AVR;
with NXT.Ultrasonic_Sensors; use NXT.Ultrasonic_Sensors;
with NXT.Ultrasonic_Sensors.Ctors;

procedure Ultrasonic_Test is
  use NXT;
  Result : Button_Id;
  Cur_Sensor : NXT.Ultrasonic_Sensors.Ultrasonic_Sensor := NXT.Ultrasonic_Sensors.Ctors.Ultrasonic_Sensor;
  Distance : Natural range 0..255;
begin
  NXT.AVR.Await_Data_Available;
  Put_Line ("Ultrasonic Test: Ping!");
  loop
    Ping(Cur_Sensor);
    Get_Distance(Cur_Sensor, Distance);
    if (Distance = 255) then
      Clear_Screen_Noupdate;
      Put_Noupdate("Nothing in sight!");
      Screen_Update;
    else
      Clear_Screen_Noupdate;
      Put_Noupdate("Hi Ken! Something is ");
      Put_Noupdate(Distance);
      Put_Noupdate(" cm away!");
      Screen_Update;
    end if;
  end loop;
end Ultrasonic_Test;
```

# NXT de LEGO

- Posibilidad de programación

- SW LEGO

- RobotC

- MATLAB - Toolbox

- NI LabVIEW

- ADA

- JAVA

- ROS



# NXT de LEGO

- Posibilidad de programación

- SW LEGO

- RobotC

- MATLAB - Toolbox

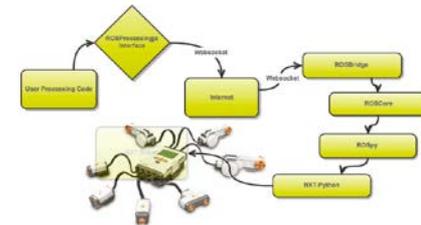
- NI LabVIEW

- ADA

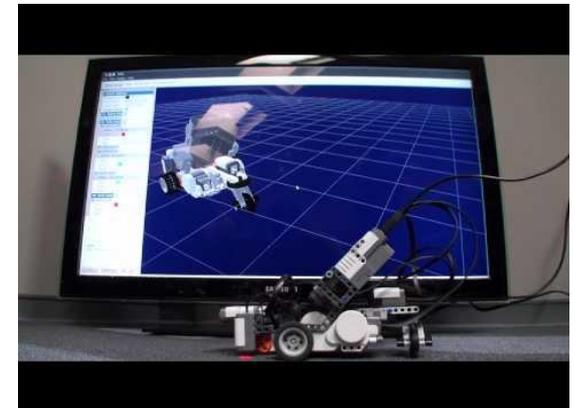
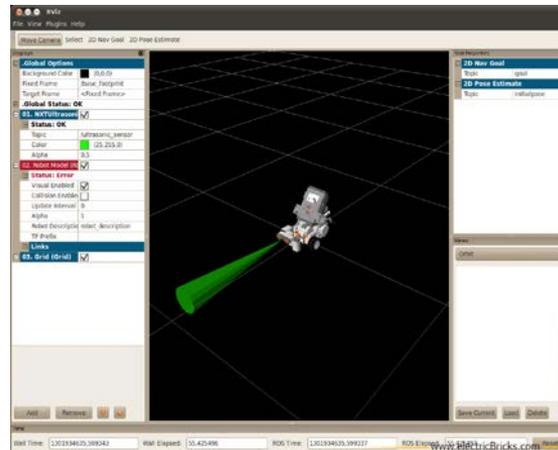
- JAVA

- ROS

ROS.org



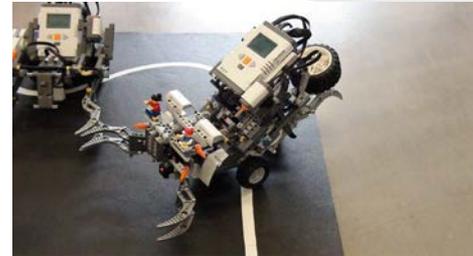
ROSProcessingNXT



# NXT de LEGO

- **Desarrollos ejemplo**

- **Asignaturas:**
  - **Tiempo Real**
  - **Sistemas de Percepción**



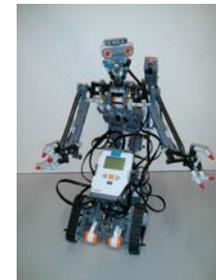
[http://www.youtube.com/watch?v=TIcvtBaVg\\_c](http://www.youtube.com/watch?v=TIcvtBaVg_c)

- **PFC**
- **Concurso GT CEA de CI**



[http://www.youtube.com/watch?v=3W6MwTS\\_syc](http://www.youtube.com/watch?v=3W6MwTS_syc)

- **Semana de la ciencia - Jornadas Ingenierías:**
  - **Segway + Johnny 5**



# Arduino

## ● Presentación del equipo

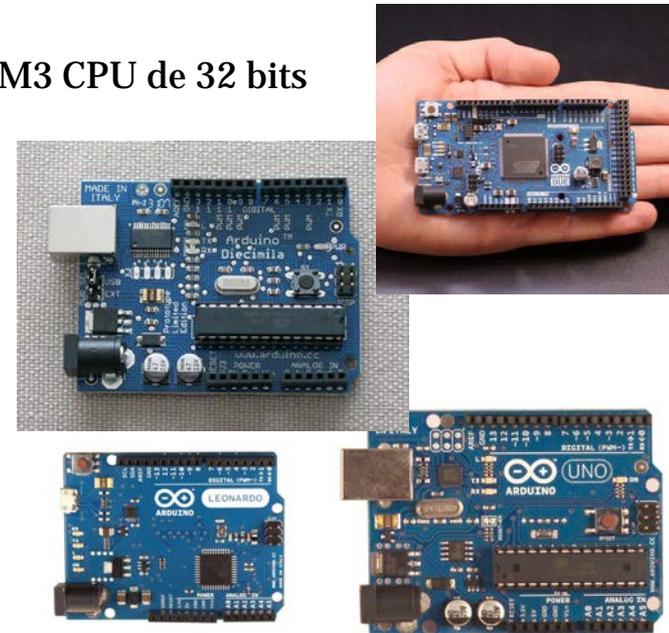
### ○ Arduino Due

- Microcontrolador Atmel SAM3X8E ARM Cortex-M3 CPU de 32 bits
- 256 a 512 Kb de memoria Flash
- 32 a 100 Kb de SRAM
- 54 pins de I/O digital (12 para PWM-O)
- Clock: 84 MHz
- Comunicación: USB 2.0
- Ejecución: autónoma + remota

### ○ Diversa sensorización

- Temperatura, ultrasonidos, acelerómetro,...

### ○ Extensa familia de placas base y dedicadas



# Arduino

- Desarrollos ejemplo
  - Estudio de sensor ultrasónico

<http://www.youtube.com/watch?v=b5Jfze6lthQ>

- Integración con diferentes elementos

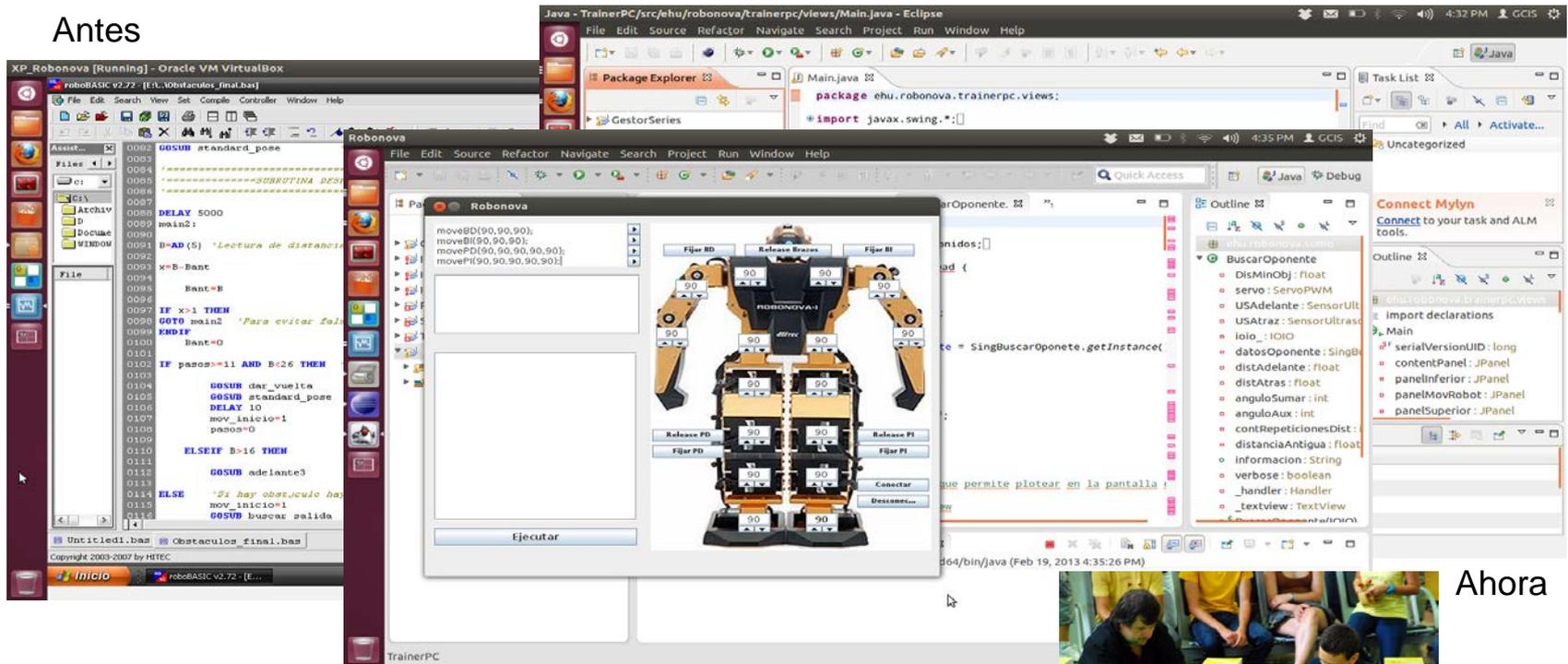


<http://www.youtube.com/watch?v=33V9n-U6UwQ>

# Arduino

- Desarrollos ejemplo
  - Nikita (Robonova): nuevo sistema de programación

Antes



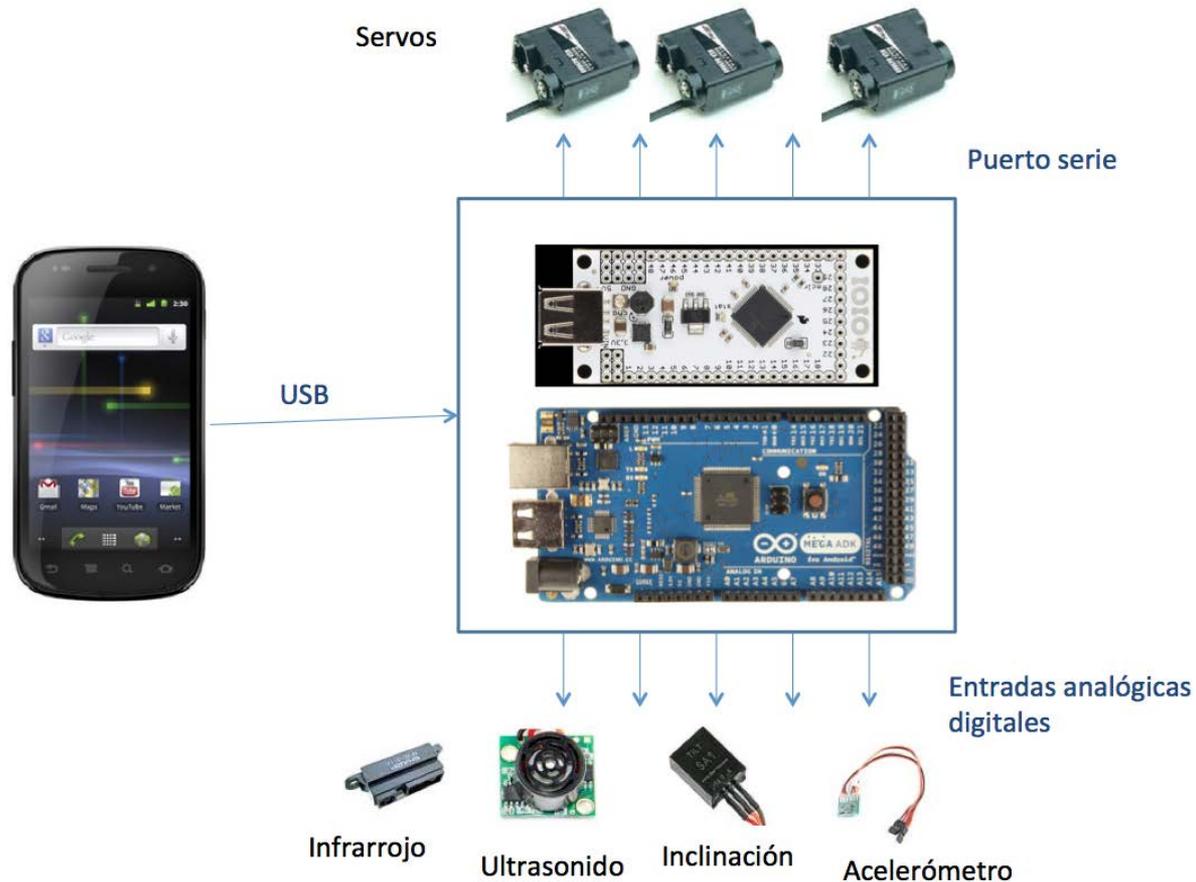
Ahora

- Concurso CEABOT del grupo Robótica de CEA



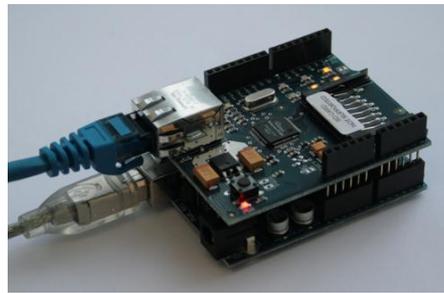
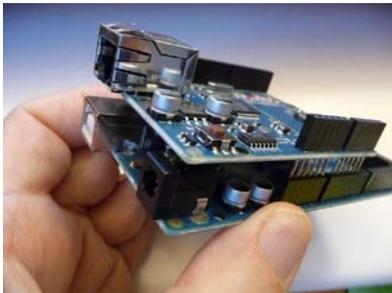
# Arduino

- **Desarrollos ejemplo**
  - **Nikita (Robonova): nuevo sistema de programación**



# Arduino

- **Desarrollos en camino**
  - **Utilización en aplicaciones tipo PLC**
    - **Arduino Ethernet Shield**

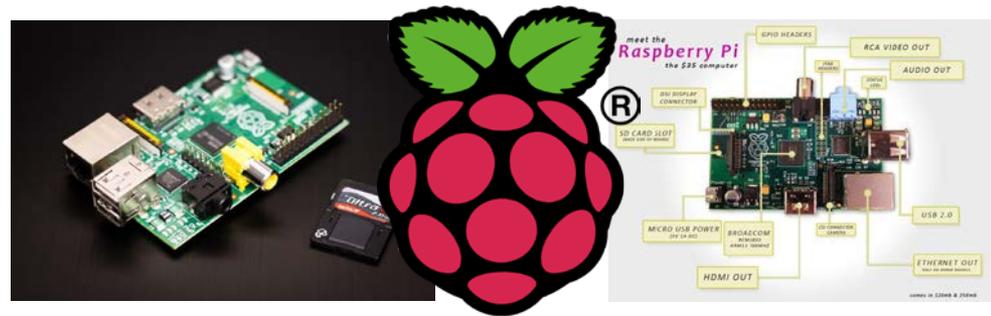


- **Domotica X10**
  - **Librería para protocolo de domótica X10**

# Otras plataformas

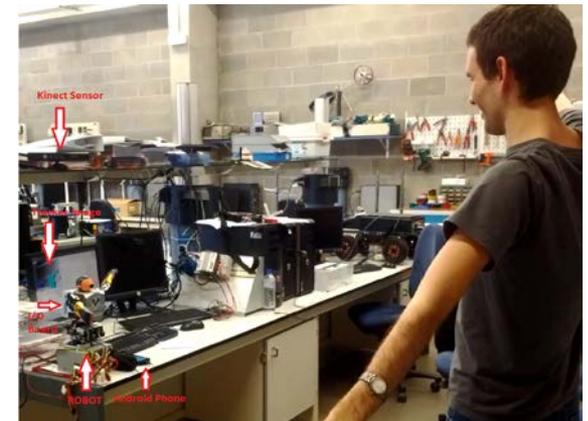
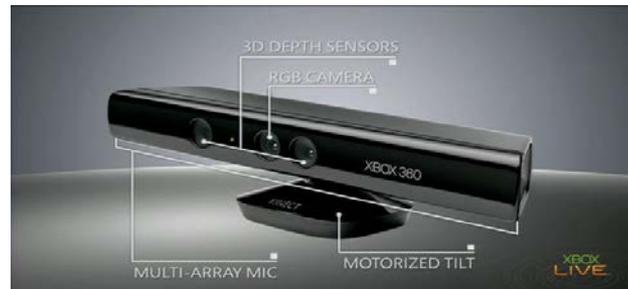
- Raspberry pi (<http://www.raspberrypi.org/>)

- Procesador ARM1176JZF-S a 700 MHz
- Clock de hasta 1 GHz
- 512 Mb de RAM
- Tarjeta SD



- Microsoft Kinect

**KINECT™**



<http://www.youtube.com/watch?v=UrYv1IArDJs>

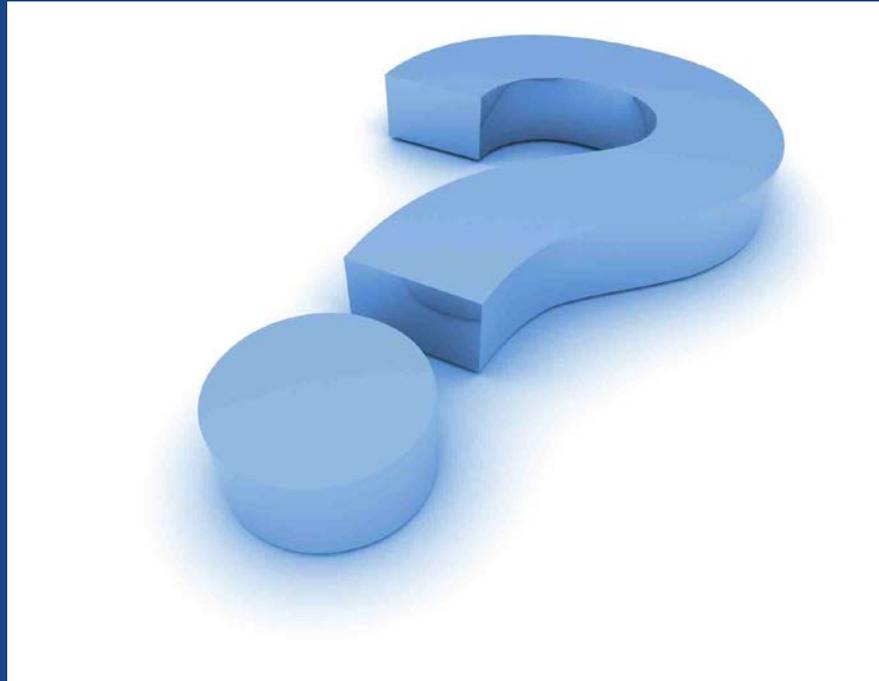
# Foros, comunidades y asociaciones

- **Generales**
  - Páginas WEB. Código abierto. Soluciones particulares.
  - Junior FIRST LEGO League (6-9 años), FIRST LEGO League (10-16 años) y FIRST Tech Challenge (17-23 años)
- **Educativos**
  - Preuniversitarios
  - Universitarios
- **Investigación**
  - Grupos de trabajo
  - Entidades: Asociaciones. Empresas (Spin-off)

# Conclusiones

- Rediseño y/o complementación de materias/asignaturas mediante la introducción de plataformas de desarrollo que ofrezcan la posibilidad de mejorar los modelos de enseñanza actuales.
- Sencillez en el manejo y bajo coste.
- Aprovechar las capacidades e inquietudes de los alumnos para mejorar las vías de aprendizaje.
- Formación del alumno en nuevas metodologías de programación.

¡Muchas gracias por vuestra atención!



*eloy.irigoyen@ehu.es*