1.- Introduction

1.1.- Goals

Many developers around the world choose leJOS, Java for Lego Mindstorm, as the main platform to develop robots with NXT Lego Mindstorm. I consider that this eBook will help leJOS community, Lego Mindstorm community, Robot's developers and Java fans to develop better software.

Robotics will be very important for the humanity in the next 10 years and this eBook is an effort to help in this way.

Many people spend several hours in their robotics projects with problems with wires & electronics, protocols and problems with programming languages, Lego Mindstorm is easy and Java/leJOS is an excellent platform to demonstrate your software engineering skills to develop better robots. NXT Brick is the easiest way to enter in the robotics world and leJOS the best platform in the moment to use software engineering ideas.

Enjoy, Learn, Contact with me to improve the eBook and share your ideas.

Juan Antonio Breña Moral.
www.juanantonio.info

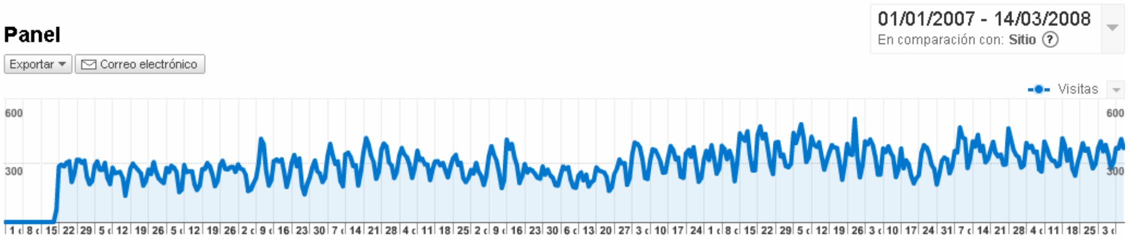
1.2.- LeJOS Project

LeJOS is Sourceforge project created to develop a technological infrastructure to develop software into Lego Mindstorm Products using Java technology.

Currently leJOS has opened the following research lines:

1. NXT Technology
    a. NXJ
    b. iCommand
2. RCX Technology
    a. leJOS for RCX

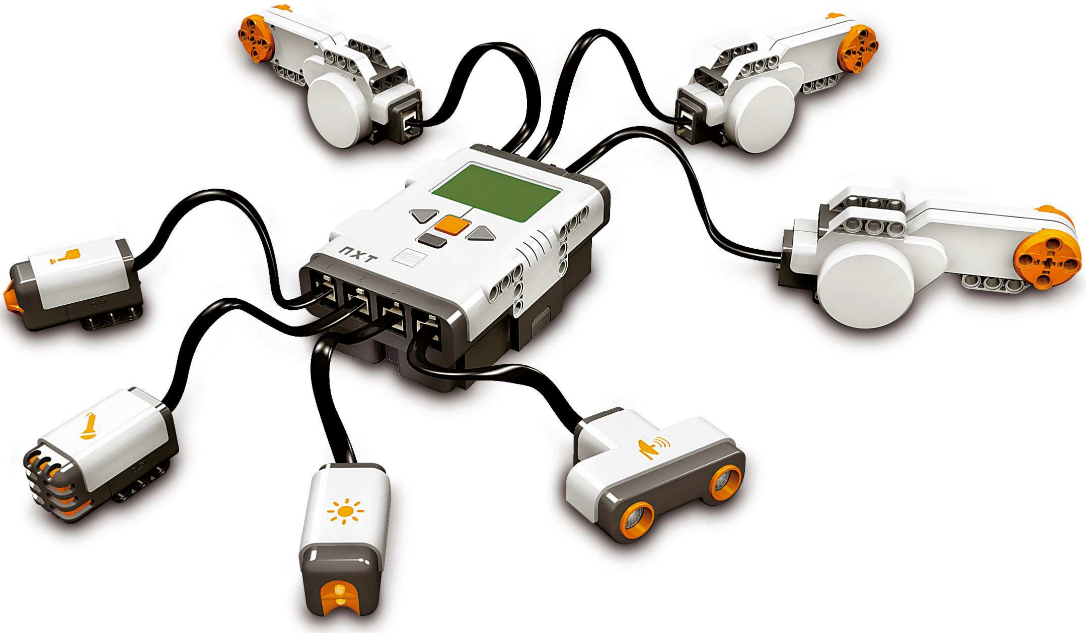
LeJOS project's audience has increased. Currently more than 500 people visit the website every day.



This eBook will focus in NXT technology with NXJ using a Windows Environment to develop software.

1.3.- NXT Brick

The NXT is the brain of a MINDSTORMS robot. It's an intelligent, computer-controlled LEGO brick that lets a MINDSTORMS robot come alive and perform different operations.



Motor ports
The NXT has three output ports for attaching motors - Ports A, B and C

Sensor ports
The NXT has four input ports for attaching sensors - Ports 1, 2, 3 and 4.

USB port
Connect a USB cable to the USB port and download programs from your computer to the NXT (or upload data from the robot to your computer). You can also use the wireless Bluetooth connection for uploading and downloading.

Loudspeaker
Make a program with real sounds and listen to them when you run the program

NXT Buttons
Orange button: On/Enter /Run
Light grey arrows: Used for moving left and right in the NXT menu
Dark grey button: Clear/Go back

NXT Display
Your NXT comes with many display features - see the MINDSTORMS NXT Users Guide that comes with your NXT kit for specific information on display icons and options

Technical specifications

- 32-bit ARM7 microcontroller
- 256 Kbytes FLASH, 64 Kbytes RAM
- 8-bit AVR microcontroller

- 4 Kbytes FLASH, 512 Byte RAM
- Bluetooth wireless communication (Bluetooth Class II V2.0 compliant)
- USB full speed port
- 4 input ports, 6-wire cable digital platform (One port includes a IEC 61158 Type 4/EN 50 170 compliant expansion port for future use)
- 3 output ports, 6-wire cable digital platform
- 100 x 64 pixel LCD graphical display
- Loudspeaker - 8 kHz sound quality. Sound channel with 8-bit resolution and 2-16 KHz sample rate.
- Power source: 6 AA batteries

1.3.1.- NXT Sensors used in the eBook

NXT Sensors used in the document are the following:

- NXT Motor
- Ultrasonic Sensor
- Compass Sensor
- NXTCam
- Tilt Sensor
- NXTCam
- RFID Sensor

NXT Motor

Ultrasonic Sensor

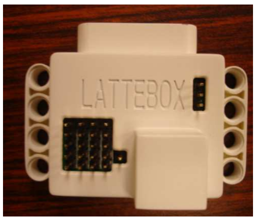
Compass Sensor

Tilt Sensor

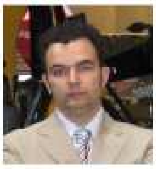
NXTCam

RFID Sensor

Lattebox NXTe



1.4.- About the author



Juan Antonio Breña Moral collaborates in leJOS Research team since 2006. He works in Europe leading Marketing, Engineering and IT projects for middle and large customers in several markets as Defence, Telecommunications, Pharmaceutics, Energy, Automobile, Construction, Insurance and Internet.

Further information:
www.juanantonio.info
www.esmeta.es

2.- Managing RC Servos with leJOS

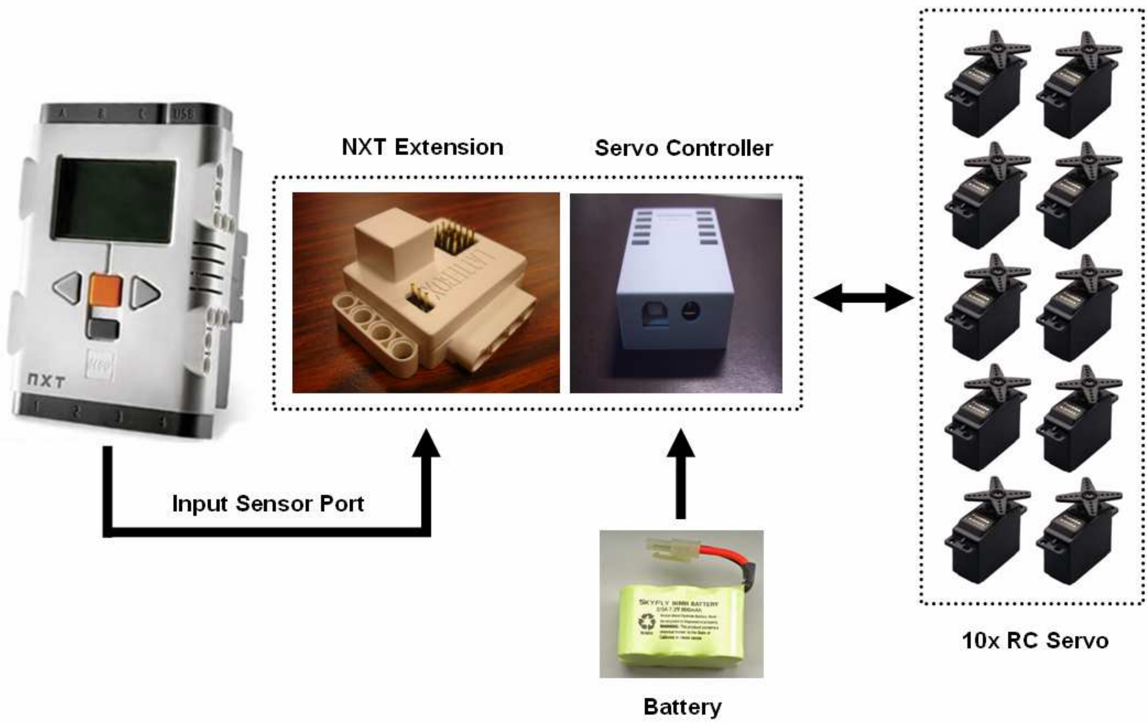
2.1.- Introduction

In 2008, Lattebox a hi-tech company located in Taiwan, launch a new kind of NXT device, NXTe. NXTe allows controlling RC servos easily. NXT brick has 4 sensor port inputs to control NXT sensors as Ultrasonic Sensors, Compass Sensors, NXTCam Sensors, etc…

If you connect Lattebox NXTe in any free input sensor port, you could manage until 10 RC Servos with your NXT brick with an unique NXTe kit.

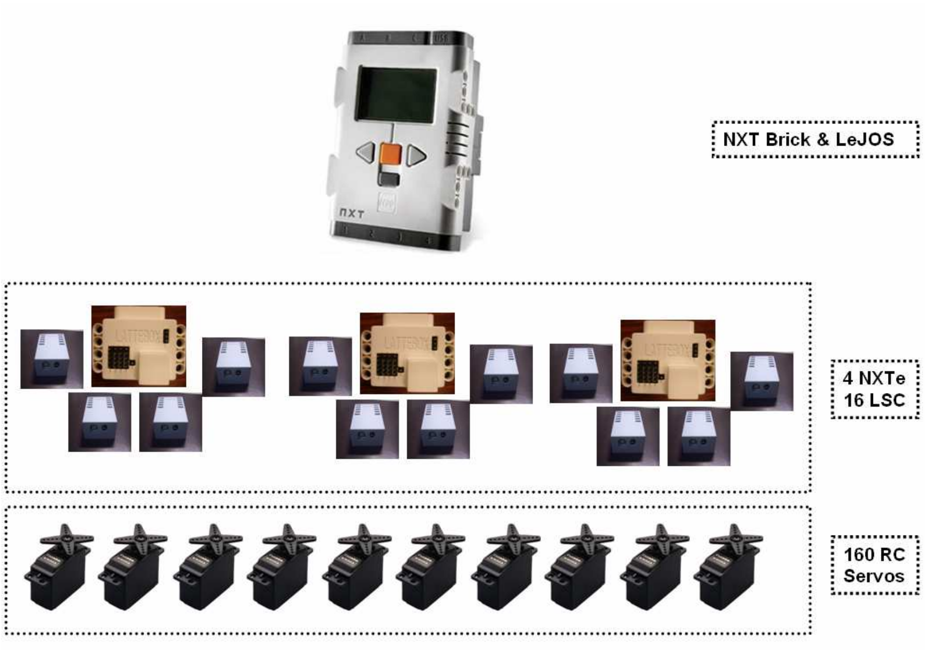
2.2.- Lattebox NXTe architecture

The NXTe architecture is the following:



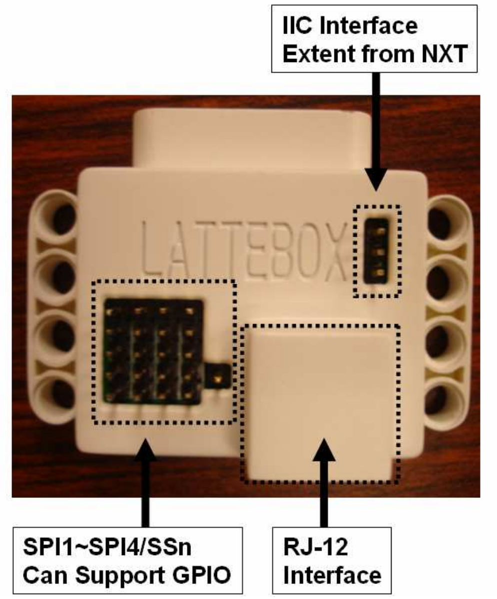
If you notice, with NXTe technology you can control:

4x NXTe * 4x LSC * 10x RC Servos = 160 Servos



2.2.1.- NXT Extension, NXTe

NXT Extension is new device developed by Lattebox to stablish a bridge between the world of RC Servo and NXT Technology. NXTe uses the energy from NXT Brick. This device is able to manage until 4 Servo Controller.

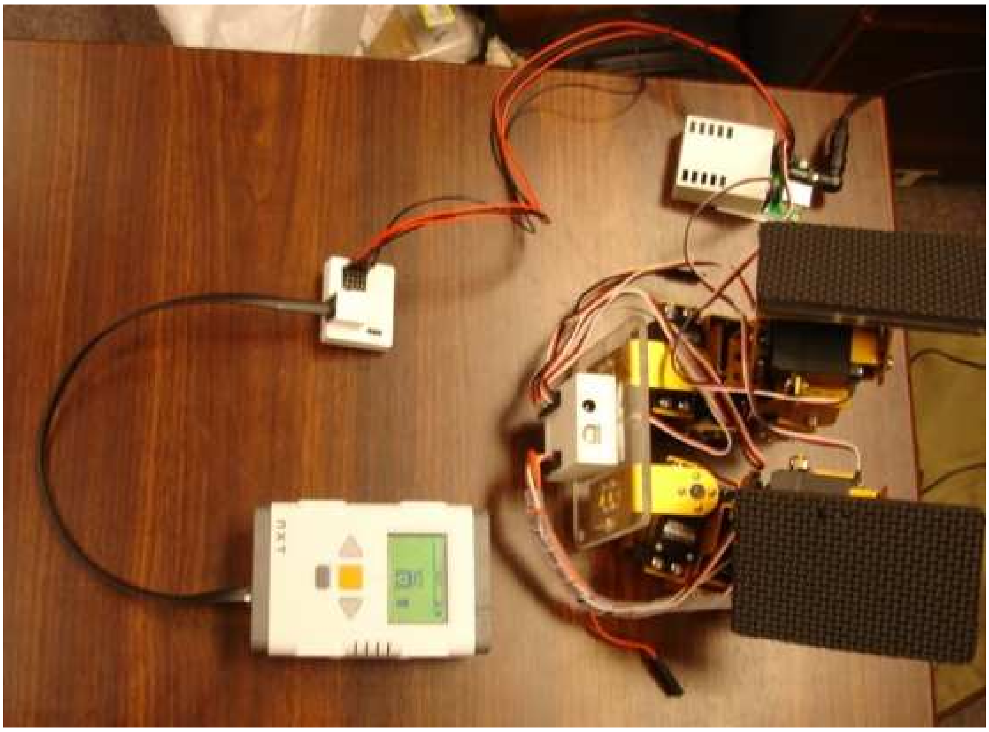


2.2.2.- Servo Controller

Servo Controller Device, is connected with NXTe to manage until 10 RC Servos.

This device needs an external energy, 6.8V/4000mAh source to runs.

If you have 1 Servo Controller, connect this one in SPI 1 in NXTe:



When you connect a any servo into LSC, you have to know pin details:

- Pin 1: Signal
- Pin 2: Positivie
- Pin 3: Negative

2.3.- Servos

A Servo is a small device that has an output shaft. This shaft can be positioned to specific angular positions by sending the servo a coded signal. As long as the coded signal exists on the input line, the servo will maintain the angular position of the shaft. As the coded signal changes, the angular position of the shaft changes. Most servo motors can rotate about 90 to 180 degrees. Some rotate through a full 360 degrees.

[redacted]

2.3.1.- How does a servo work?

The servo motor has some control circuits and a potentiometer (a variable resistor, aka pot) that is connected to the output shaft. If the shaft is at the correct angle, then the motor shuts off. If the circuit finds that the angle is not correct, it will turn the motor the correct direction until the angle is correct. The output shaft of the servo is capable of travelling somewhere around 180 degrees. Usually, its somewhere in the 210 degree range, but it varies by manufacturer. A normal servo is used to control an angular motion of between 0 and 180 degrees. A normal servo is mechanically not capable of turning any farther due to a mechanical stop built on to the main output gear.

The amount of power applied to the motor is proportional to the distance it needs to travel. So, if the shaft needs to turn a large distance, the motor will run at full speed. If it needs to turn only a small amount, the motor will run at a slower speed. This is called proportional control.

2.4.- LeJOS and Lattebox

2.4.1.- Lattebox Support in LeJOS

Currently leJOS support NXTe but it is neccesary to improve:

- Load a unique servo
- Unload all servos in a LSC
- Unload a unique servo
- Test NXTe with DC Motors
- Control Servo 2-10 (Currently only manage 1 Servo)
- Manage speed in Servos

2.4.2.- Example using lattebox classes with leJOS

If you use leJOS, currently exist in Beta Phase a set of classes to manage NXTe, LSC and Servos easily.

```
import lejos.nxt.*;

public class LatteboxTest{
      private static NXTe NXTeObj;
      private static DebugMonitor dm;
      private static int angle;
      private static int motion;

      //Main
      public static void main(String[] args) throws Exception{
            dm = new DebugMonitor();
            dm.setLCDLines(6);
            dm.echo("Testing NXTe");

            try{

                  NXTeObj = new NXTe(SensorPort.S1);//NXTe Controller
pluged in Port1
                  NXTeObj.addLSC(0);
                  dm.echo("Calibrating LSC");
                  NXTeObj.LSC(0).addServo(0,"SAVOX, Digital SC-0352");
                  NXTeObj.LSC(0).addServo(1,"SAVOX, Digital SC-0352");
                  //NXTeObj.LSC(0).addServo(2,"HITEC, HS-785HB");
                  NXTeObj.LSC(0).calibrate();
                  dm.echo("Load all servos");
                  NXTeObj.LSC(0).loadAllServos();
                  dm.echo(NXTeObj.LSC(0).Servo(0).getName());
                  NXTeObj.LSC(0).Servo(0).setMinAngle(200);
                  NXTeObj.LSC(0).Servo(0).setMaxAngle(1700);

                  NXTeObj.LSC(0).Servo(0).setDelay(1, 2);

                  while(!Button.ESCAPE.isPressed()){

                        if (Button.LEFT.isPressed()){
                              //NXTeObj.LSC(0).Servo(0).setAngle(1,
400);
                              NXTeObj.LSC(0).Servo(0).goToMinAngle();

                              while(NXTeObj.LSC(0).Servo(0).isMoving()
== true){

      //dm.echo(NXTeObj.LSC(0).Servo(0).readMotion());
                              }
                              angle =
NXTeObj.LSC(0).Servo(0).getAngle(1);

                              dm.echo("Goto Min");
                              dm.echo(angle);
                        }

                        if (Button.ENTER.isPressed()){

      NXTeObj.LSC(0).Servo(0).goToMiddleAngle();
```

```
                              while(NXTeObj.LSC(0).Servo(0).isMoving()
== true){

      //dm.echo(NXTeObj.LSC(0).Servo(0).readMotion());
                              }
                              angle =
NXTeObj.LSC(0).Servo(0).getAngle(1);

                              dm.echo("Goto Middle");
                              dm.echo(angle);
                        }

                        if (Button.RIGHT.isPressed()){
                              //NXTeObj.LSC(0).Servo(0).setAngle(1,
2000);
                              NXTeObj.LSC(0).Servo(0).goToMaxAngle();
                              while(NXTeObj.LSC(0).Servo(0).isMoving()
== true){

      //dm.echo(NXTeObj.LSC(0).Servo(0).readMotion());
                              }
                              angle =
NXTeObj.LSC(0).Servo(0).getAngle(1);

                              dm.echo("Goto Middle");
                              dm.echo(angle);
                        }
                  }

            }catch(Exception e){
                  dm.echo(e.getMessage());
            }

            dm.echo("Test finished");
      }
}
```