

# GUIDA

Guida per l'uso semplificato di leJOS con Lego NXT

## Sommario

Installazione dell'ambiente per l'uso della libreria .....	3
Installazione ambiente Java .....	3
Installazione Fantom Driver .....	6
Installazione leJOS .....	7
Classi .....	8
Navigation .....	8
WaitTime .....	9
WaitTouchSensor .....	9
WaitColorSensor .....	10
WaitUltrasonicSensor .....	10
WaitSoundSensor .....	11
WaitLightSensor .....	11

## Installazione dell'ambiente per l'uso della libreria

Per poter usare nel modo corretto questa libreria bisogna installare Java, il Fantom Driver e lejOS che ci permetteranno di far funzionare il tutto una volta messi assieme.

### Installazione ambiente Java

Java è l'ambiente su cui è basato lejOS, di conseguenza lo necessitiamo per fare funzionare i nostri programmi, l'installazione è relativamente semplice.

**Nota:** Bisogna scaricare Java solamente per pc Windows, su Mac è già installato di default.

Bisogna scaricare la JDK e la JRE che sono i due componenti principali di Java che troviamo nei seguenti posti:

JDK: <https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

JRE: <https://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html>

Per entrambi le procedure di installazione sono le stesse, bisogna andare sui due siti e cercare la versione corretta, nel nostro caso la **"Java SE Runtime Environment 8u201"**; Dovrebbe apparirci questa pagina:

Product / File Description	File Size	Download
Linux x86	68.1 MB	<a href="#">jre-8u201-linux-i586.rpm</a>
Linux x86	83.8 MB	<a href="#">jre-8u201-linux-i586.tar.gz</a>
Linux x64	64.91 MB	<a href="#">jre-8u201-linux-x64.rpm</a>
Linux x64	80.73 MB	<a href="#">jre-8u201-linux-x64.tar.gz</a>
Mac OS X x64	76.18 MB	<a href="#">jre-8u201-macosx-x64.dmg</a>
Mac OS X x64	67.77 MB	<a href="#">jre-8u201-macosx-x64.tar.gz</a>
Solaris SPARC 64-bit	46.27 MB	<a href="#">jre-8u201-solaris-sparcv9.tar.gz</a>
Solaris x64	50.14 MB	<a href="#">jre-8u201-solaris-x64.tar.gz</a>
Windows x86 Online	1.87 MB	<a href="#">jre-8u201-windows-i586-iftw.exe</a>
Windows x86 Offline	63.53 MB	<a href="#">jre-8u201-windows-i586.exe</a>
Windows x86	66.51 MB	<a href="#">jre-8u201-windows-i586.tar.gz</a>
Windows x64	71.44 MB	<a href="#">jre-8u201-windows-x64.exe</a>
Windows x64	71.29 MB	<a href="#">jre-8u201-windows-x64.tar.gz</a>

Successivamente per poter scaricare la versione dobbiamo accettare il contratto della licenza cliccando su **"Accept License Agreement"** in alto a sinistra. Successivamente la pagina dovrebbe essere la seguente:

Product / File Description	File Size	Download
Linux x86	68.1 MB	<a href="#">jre-8u201-linux-i586.rpm</a>
Linux x86	83.8 MB	<a href="#">jre-8u201-linux-i586.tar.gz</a>
Linux x64	64.91 MB	<a href="#">jre-8u201-linux-x64.rpm</a>
Linux x64	80.73 MB	<a href="#">jre-8u201-linux-x64.tar.gz</a>
Mac OS X x64	76.18 MB	<a href="#">jre-8u201-macosx-x64.dmg</a>
Mac OS X x64	67.77 MB	<a href="#">jre-8u201-macosx-x64.tar.gz</a>
Solaris SPARC 64-bit	46.27 MB	<a href="#">jre-8u201-solaris-sparcv9.tar.gz</a>
Solaris x64	50.14 MB	<a href="#">jre-8u201-solaris-x64.tar.gz</a>
Windows x86 Online	1.87 MB	<a href="#">jre-8u201-windows-i586-iftw.exe</a>
Windows x86 Offline	63.53 MB	<a href="#">jre-8u201-windows-i586.exe</a>
Windows x86	66.51 MB	<a href="#">jre-8u201-windows-i586.tar.gz</a>
Windows x64	71.44 MB	<a href="#">jre-8u201-windows-x64.exe</a>
Windows x64	71.29 MB	<a href="#">jre-8u201-windows-x64.tar.gz</a>

Come prossimo passo dobbiamo conoscere l'architettura del nostro sistema operativo, se il pc è un Windows basta andare nello start, scrivere "Questo PC" e cliccare il tasto destro andando su "Proprietà".

Appare una finestra con svariati dettagli sul proprio pc fra di cui l'architettura del sistema:

Edizione Windows

Windows 10 Pro

© 2016 Microsoft Corporation. Tutti i diritti sono riservati.

---

Sistema

Processore:

Intel(R) Core(TM) i7-4770HQ CPU @ 2.20GHz 2.19 GHz

Memoria installata (RAM):

8,00 GB

Tipo sistema:

Sistema operativo a 64 bit, processore basato su x64

Penna e tocco:

Nessun input penna o tocco disponibile per questo schermo

Per procedere al download basta cliccare sui link blu che stanno a destra della dimensione della versione e, conoscendo la propria versione del sistema operativo scegliere quella corretta.

Esempio per la versione di Windows 10 64-bit:

Java SE Runtime Environment 8u201		
You must accept the <a href="#">Oracle Binary Code License Agreement for Java SE</a> to download this software.		
Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.		
Product / File Description	File Size	Download
Linux x86	68.1 MB	<a href="#">jre-8u201-linux-i586.rpm</a>
Linux x86	83.8 MB	<a href="#">jre-8u201-linux-i586.tar.gz</a>
Linux x64	64.91 MB	<a href="#">jre-8u201-linux-x64.rpm</a>
Linux x64	80.73 MB	<a href="#">jre-8u201-linux-x64.tar.gz</a>
Mac OS X x64	76.18 MB	<a href="#">jre-8u201-macosx-x64.dmg</a>
Mac OS X x64	67.77 MB	<a href="#">jre-8u201-macosx-x64.tar.gz</a>
Solaris SPARC 64-bit	46.27 MB	<a href="#">jre-8u201-solaris-sparcv9.tar.gz</a>
Solaris x64	50.14 MB	<a href="#">jre-8u201-solaris-x64.tar.gz</a>
Windows x86 Online	1.87 MB	<a href="#">jre-8u201-windows-i586-iftw.exe</a>
Windows x86 Offline	63.53 MB	<a href="#">jre-8u201-windows-i586.exe</a>
Windows x86	66.51 MB	<a href="#">jre-8u201-windows-i586.tar.gz</a>
Windows x64	71.44 MB	<a href="#">jre-8u201-windows-x64.exe</a>
Windows x64	71.29 MB	<a href="#">jre-8u201-windows-x64.tar.gz</a>

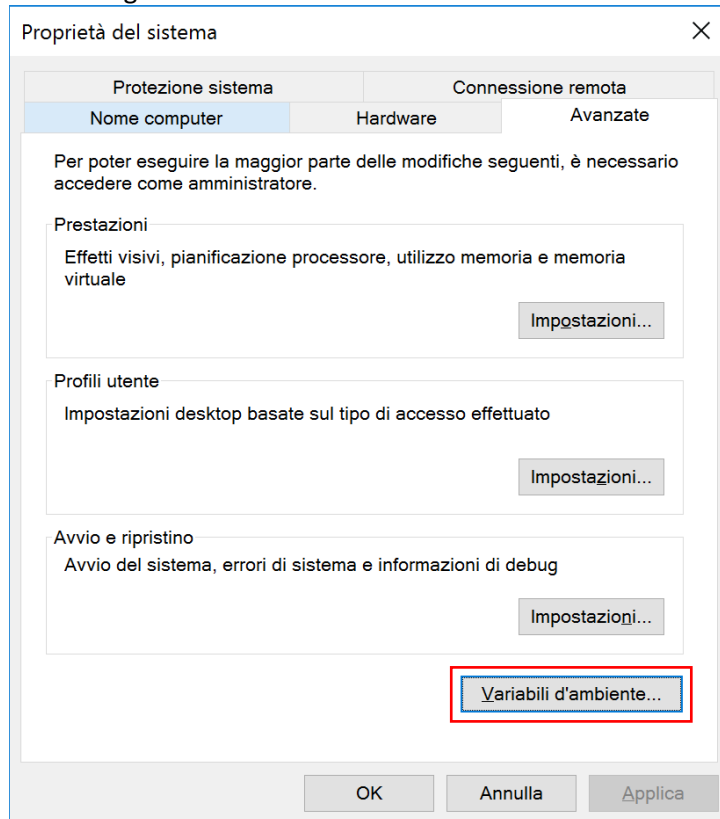
Fare questa procedura sia per la JDK che per la JRE.

Una volta scaricati entrambi, basta eseguirle i loro installer cliccando sempre su "Avanti".

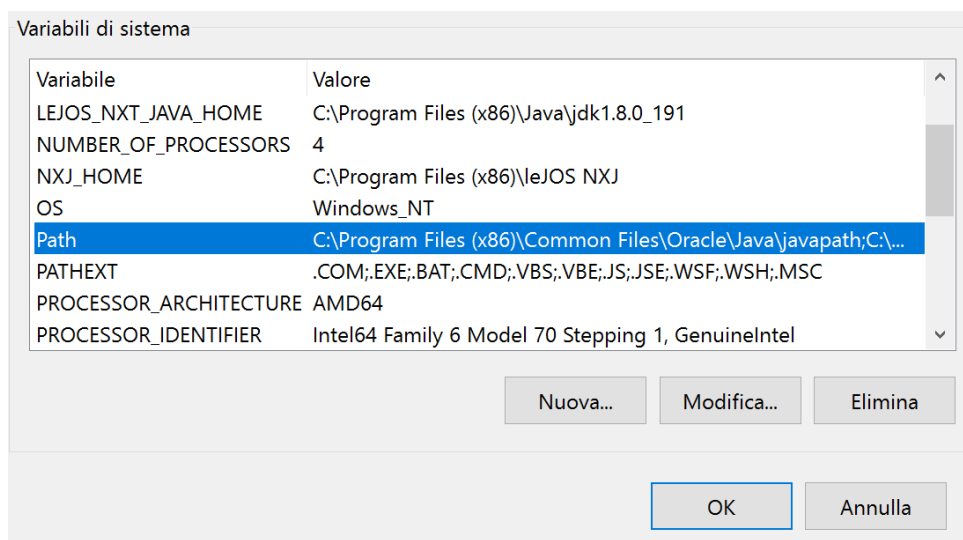


Come ultimo passo dobbiamo impostare la variabile d'ambiente del sistema facendo le seguenti operazioni:

Scriviamo “Modifica le variabili di ambiente del sistema” nello start e clicchiamo sull'icona che appare, successivamente bisogna andare su “**Variabili d'ambiente**”



Poi si deve andare sotto “Path” e cliccare “Modifica”



Infine basta cliccare il bottone “Nuovo” e immettere il percorso della cartella bin della JDK come ad esempio: **C:\Program Files\Java\jdk1.8.0\_191\bin**.

## Installazione Fantom Driver

Un'ulteriore tool che ci serve per far funzionare leJOS ed NXT nel modo corretto è un piccolo programmino che si chiama Fantom driver, questo lo si scarica sul sito della lego:  
<https://www.lego.com/en-us/mindstorms/downloads>

### NXT SOFTWARE DOWNLOAD (PC/MAC)

📄 NXT Software Download

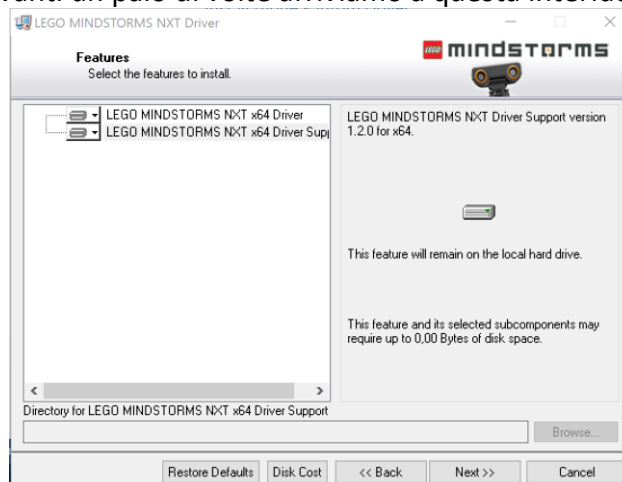
📄 NXT Firmware Download

📄 Download the NXT Fantom Driver

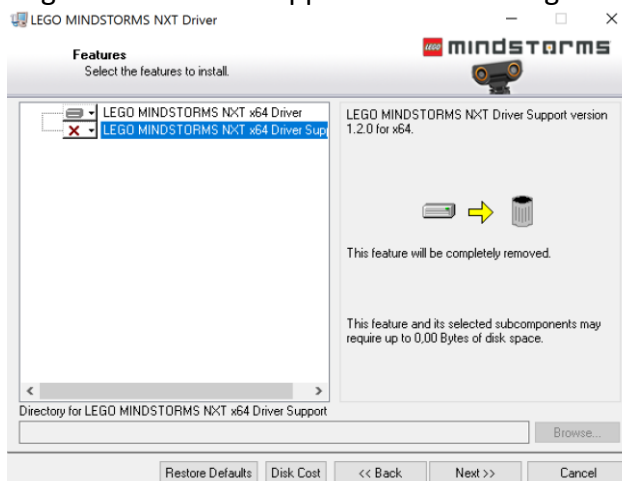


Basta cliccare sulla scritta con “Download the NXT...” e il driver verrà scaricato.

Una volta scaricato bisogna estrarre il contenuto della cartella .zip ed eseguire il setup.  
Dopo aver premuto avanti un paio di volte arriviamo a questa interfaccia:



Nella quale dobbiamo togliere il secondo supporto nel modo seguente:



Come ultima cosa basta cliccare “Next” e “Finish”.

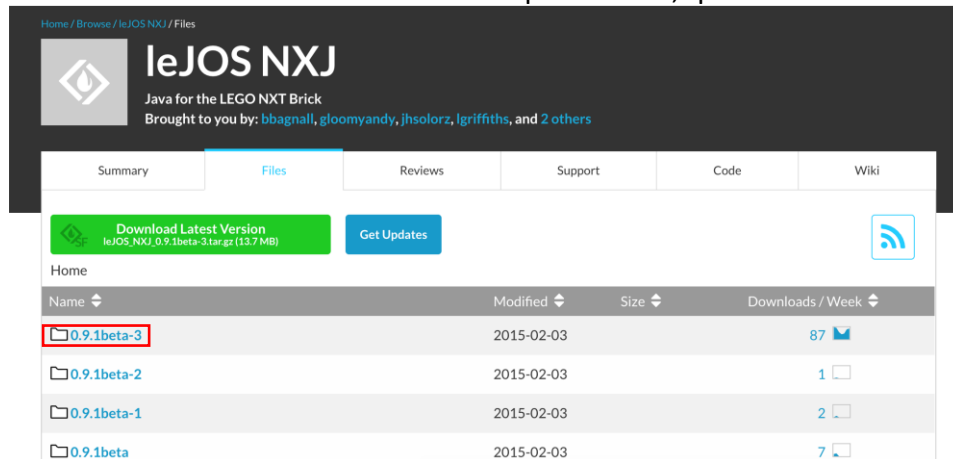
## Installazione leJOS

LeJOS è un insieme di classi scritte in Java che ci permette di controllare i robot di tipo NXT ed EV3(NXT nel nostro caso).

Per scaricarlo basta andare sul suo sito ufficiale di download:

<https://sourceforge.net/projects/nxt.lejos.p/files/>

Successivamente dobbiamo scaricare la versione più recente, quindi la 0.9.1-3.



Ed infine scaricare la versione .exe se siamo su Windows oppure la .zip se si è su Mac.

Una volta scaricato iniziamo ad eseguire l'installer nel quale dovremo solamente cliccare continuamente "Avanti".

## Classi

### Navigation

Navigation è la classe che si occupa di gestire l'uso dei motori con tutti i suoi rispettivi metodi.

Per implementare questa classe si usa:

```
Navigation navigator = new Navigation(LEFT_MOTOR_PORT, RIGHT_MOTOR_PORT);
```

I metodi che contiene sono i seguenti:

Navigation(char Left_Motor, char Right_Motor) ;	Costruttore della classe che riceve come parametri la porta sinistra e quella destra dei due motori. <i>Navigation('A', 'B');</i>
char getLeftMotorPort()	Ritorna la porta del motore sinistro.
char getRightMotorPort()	Ritorna la porta del motore destro.
void setMotorLeftPort(char port)	Controlla e assegna il valore "port" alla variabile leftMotor. Valori accettati: 'A', 'B' o 'C'. <i>setMotorLeftPort('A');</i>
void setMotorRightPort(char port)	Controlla e assegna il valore "port" alla variabile rightMotor. Valori accettati: 'A', 'B' o 'C'. <i>setMotorRightPort('A');</i>
int getMySpeed()	Ritorna la velocità dei due motori.
void setMySpeed(int speed)	Controlla ed assegna il valore "speed" alla variabile speed dei due motori. Valori accettati: Consigliato da 0-100 <i>setMySpeed(75);</i>
char getDirection()	Ritorna la direzione nella quale va il robot, 'F' o 'B'.
void setDirection(char direction)	Controlla ed assegna la direzione in cui va il robot. Valori accettati: 'F' per avanti, 'B' per indietro. <i>setDirection('F');</i>
void move()	Metodo che fa partire i motori e quindi muovere il motore.
void left(int howMuch)	Fa curvare il robot a sinistra per i gradi definiti fa "howMuch". Valori accettati: Consigliato da 0-180 <i>left(10);</i>
void right(int howMuch)	Fa curvare il robot a destra per i gradi definiti fa "howMuch". Valori accettati: Consigliato da 0-180 <i>right(10);</i>
void stop()	Ferma i motori e quindi il robot.



## WaitTime

WaitTime è una classe che permette di aspettare del tempo in millisecondi.

Per implementare questa classe si usa:

```
WaitTime wt = new WaitTime();
```

Questa classe contiene tre metodi che possono essere usati dal utente:

<code>void myWait(int time);</code>	Per usare questo metodo si usa: <code>wt.myWait(2000);</code> In questo caso il programma aspetta per 2 secondi e poi continua.
<code>void setStartTime();</code>	Questo metodo serve per il metodo <code>isFinished(int time)</code> .
<code>boolean isFinished(int time);</code>	Per usare questo metodo si usa: <code>setStartTime();</code> <code>while(wt.isFinished(2000)){</code> <i>//codice da eseguire per 2 secondi</i> <code>}</code> Questo metodo esegue il codice nel while per 2 secondi.

## WaitTouchSensor

WaitTouchSensor è una classe che permette di aspettare che un sensore di tatto venga premuto.

Per implementare questa classe si usa:

```
WaitTouchSensor wts = new WaitTouchSensor(new TouchSensor(SensorPort.S3));
```

In questa classe c'è int action che definisce l'azione per terminare l'attesa secondo questi criteri:

0 Premuto

1 Rilasciato

2 Cliccato(Premuto rilasciato)

Questa classe contiene due metodi che possono essere usati dal utente:

<code>void myWait(int action);</code>	Per usare questo metodo si usa: <code>wts.myWait(0);</code> In questo caso il programma aspetta che il pulsante venga premuto e poi continua.
<code>boolean isFinished(int action);</code>	Per usare questo metodo si usa: <code>while(wts.isFinished(0)){</code> <i>//codice da eseguire finché non premuto</i> <code>}</code> Questo metodo esegue il codice nel while finché non viene preuto il pulsante.

## WaitColorSensor

WaitColorSensor è una classe che permette di aspettare che un sensore di colore veda un certo colore.

Per implementare questa classe si usa:

```
WaitColorSensor wcs = new WaitColorSensor(new ColorSensor(SensorPort.S3));
```

Questa classe contiene due metodi che possono essere usati dall'utente:

<code>void myWait(int red, int green, int blue);</code>	Per usare questo metodo si usa: <code>wcs.myWait(255,0,0);</code> In questo caso il programma aspetta che il sensore di colore veda rosso.
<code>boolean isFinished(int red, int green, int blue);</code>	Per usare questo metodo si usa: <pre>while(wcs.isFinished(255,0,0)){     //codice da eseguire finché non vede rosso }</pre> Questo metodo esegue il codice nel while finché il sensore non vede rosso.

## WaitUltrasonicSensor

WaitUltrasonicSensor è una classe che permette di aspettare che il sensore di ultrasuoni non veda una certa distanza.

Per implementare questa classe si usa:

```
WaitUltrasonicSensor wus = new WaitUltrasonicSensor(new UltrasonicSensor(SensorPort.S3));
```

Questa classe contiene due metodi che possono essere usati dall'utente:

<code>void myWait(Boolean bigger, int value);</code>	Per usare questo metodo si usa: <code>wus.myWait(false, 40);</code> In questo caso il programma aspetta che il sensore di ultrasuoni è a 40 cm di distanza da un oggetto poi continua.
<code>boolean isFinished(Boolean bigger, int value);</code>	Per usare questo metodo si usa: <pre>while(wus.isFinished(false, 40)){     //codice da eseguire finché non vede 40cm }</pre> Questo metodo esegue il codice nel while finché il sensore di ultrasuoni non è a 40 cm di distanza da un oggetto.

## WaitSoundSensor

WaitSoundSensor è una classe che permette di aspettare che il sensore del suono non sente un rumore di una certa potenza/debolezza.

Per implementare questa classe si usa:

```
WaitSoundSensor wus = new WaitSoundSensor(new SoundSensor(SensorPort.S3));
```

Questa classe contiene due metodi che possono essere usati dall'utente:

<code>void myWait(Boolean bigger, int value);</code>	Per usare questo metodo si usa: <code>wss.myWait(false, 40);</code> In questo caso il programma aspetta che il sensore del suono sente un suono di potenza 40 poi continua.
<code>boolean isFinished(Boolean bigger, int value);</code>	Per usare questo metodo si usa: <code>while(wss.isFinished(false, 40)){</code> <i>//codice da eseguire finché non sente meno di 40</i> <code>}</code> Questo metodo esegue il codice nel while finché il sensore del suono sente un suono di potenza 40.

## WaitLightSensor

WaitLightSensor è una classe che permette di aspettare che il sensore di luce non vede una luminosità maggiore/minore di un valore.

Per implementare questa classe si usa:

```
WaitLightSensor wls = new WaitLightSensor(new LightSensor(SensorPort.S3));
```

Questa classe contiene due metodi che possono essere usati dall'utente:

<code>void myWait(Boolean bigger, int value);</code>	Per usare questo metodo si usa: <code>wls.myWait(false, 40);</code> In questo caso il programma aspetta che il sensore di luce non vede luminosità di 40 poi continua.
<code>boolean isFinished(Boolean bigger, int value);</code>	Per usare questo metodo si usa: <code>while(wls.isFinished(false, 40)){</code> <i>//codice da eseguire finché non vede 40</i> <code>}</code> Questo metodo esegue il codice nel while finché il sensore di luce non vede una luminosità di 40.