Guida libreria LeJOS NXT

Sommario

[1 Classi di Wait 3](#_Toc502833)

[1.1 WaitMotor 3](#_Toc502834)

[1.2 WaitTime 3](#_Toc502835)

[1.3 WaitTouchSensor 3](#_Toc502836)

[1.4 WaitUltrasonicSensor 4](#_Toc502837)

[1.5 WaitSoundSensor 4](#_Toc502838)

[1.6 WaitLightSensor 5](#_Toc502839)

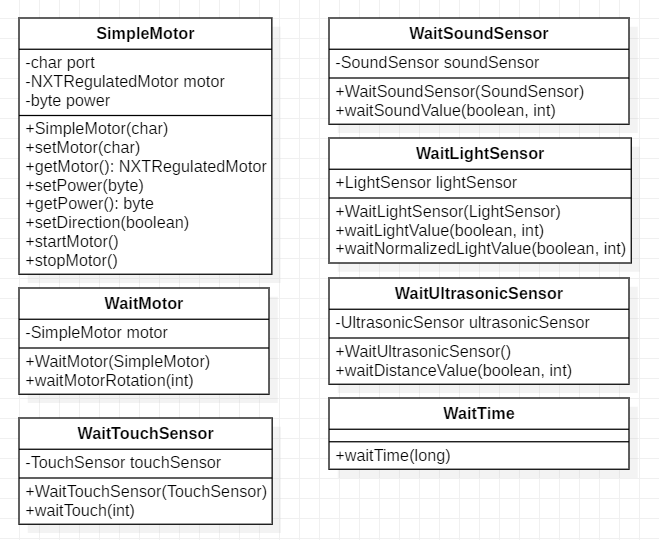
[2 Classi di Test 6](#_Toc502840)

[2.1 SimpleMotor 6](#_Toc502841)

# Classi di Wait

## WaitMotor

La classe WaitMotor permette di aspettare un determinato numero di rotazioni del motore prima di eseguire altre operazioni.

La classe è strutturata nel seguente modo:

Attributi:

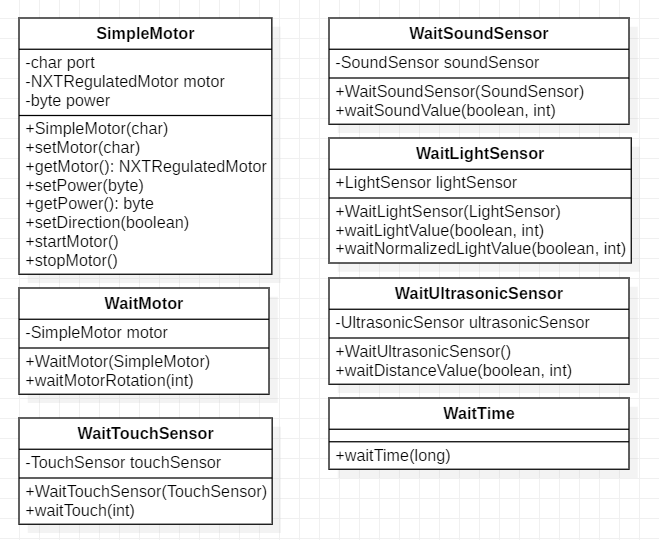
* SimpleMotor motor
  + Contiene il riferimento all’oggetto di tipo SimpleMotor

Metodi:

* public WaitMotor (SimpleMotor motor)
  + Costruttore, permette di istanziare un oggetto WaitMotor, per istanziarlo è necessario un oggetto SimpleMotor.
* Public void waitMotorRotation(int rotation)
  + Questo metodo permette di aspettare finché non sono stati effettuati un determinato numero di rotazioni del motore.  
    Il metodo ricontrolla ogni 500 millisecondi lo stato delle rotazioni

## WaitTime

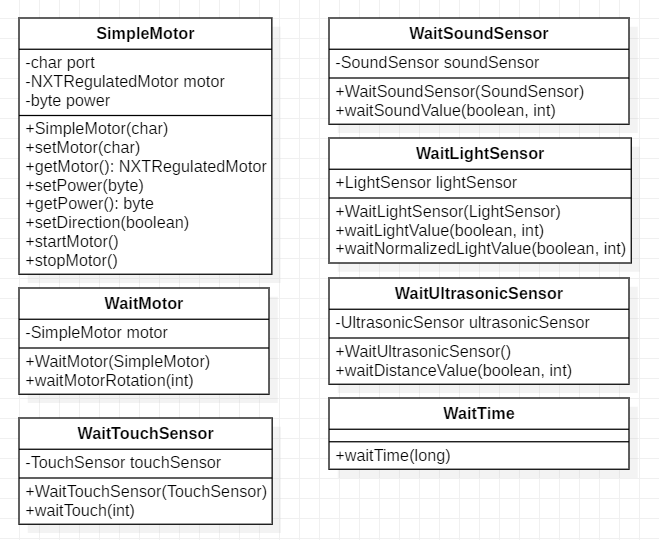
La classe WaitTime permette di aspettare un determinato lasso di tempo prima di eseguire altre operazioni.

La classe è strutturata nel seguente modo:

Metodi:

* waitTime (long milliseconds)
  + Questo metodo permette di far aspettare al programma un determinato lasso di tempo.

## WaitTouchSensor

La classe WaitTouchSensor permette di aspettare che il sensore di tatto sia stato premuto o rilasciato.   
La classe è strutturata nel seguente modo:

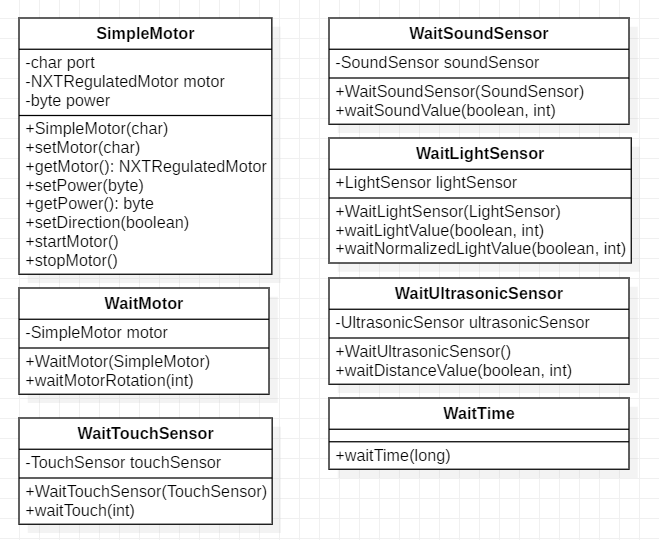
Attributi:

* TouchSensor touchSensor
  + Contiene il riferimento a un oggetto di tipo TouchSensor

Metodi:

* WaitTouchSensor (TouchSensor touchSensor)
  + Costruttore, permette di istanziare un oggetto WaitTouchSensor, per istanziarlo è necessario un oggetto TouchSensor.
* waitTouch (int mode)
  + Questo metodo permette di aspettare finche il sensore di tatto non venga premuto o rilasciato, l’azione viene scelta dal parametro mode (0 🡪 Premuto, 1🡪 Rilasciato)

## WaitUltrasonicSensor

La classe WaitUltrasonicSensor permette di aspettare che il sensore a ultrasuoni vede una certa distanza prima di effettuare altre operazioni.  
La classe è strutturata nel seguente modo:

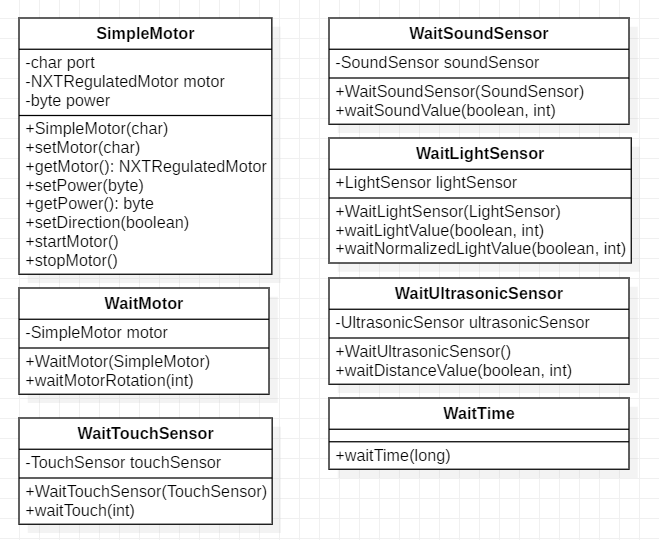
Attributi:

* UltrasonicSensor ultrasonicSensor
  + Contiene il riferimento a un oggetto di tipo UltrasonicSensor

Metodi:

* WaitUltrasonicSensor (UltrasonicSensor ultrasonicSensor)
  + Costruttore, permette di istanziare un oggetto WaitUltrasonicSensor, per istanziarlo è necessario un oggetto UltrasonicSensor.
* waitDistanceValue (Boolean sign, int value)
  + Questo metodo permette di aspettare una determinata distanza prima di effettuare altre operazioni, il parametro sign determina se la soglia deve essere maggiore o minore rispetto al valore value.

## WaitSoundSensor

La classe WaitSoundSensor permette di aspettare che il sensore di suono riceva una certa soglia di rumore prima di effettuare altre operazioni.  
La classe è strutturata nel seguente modo:

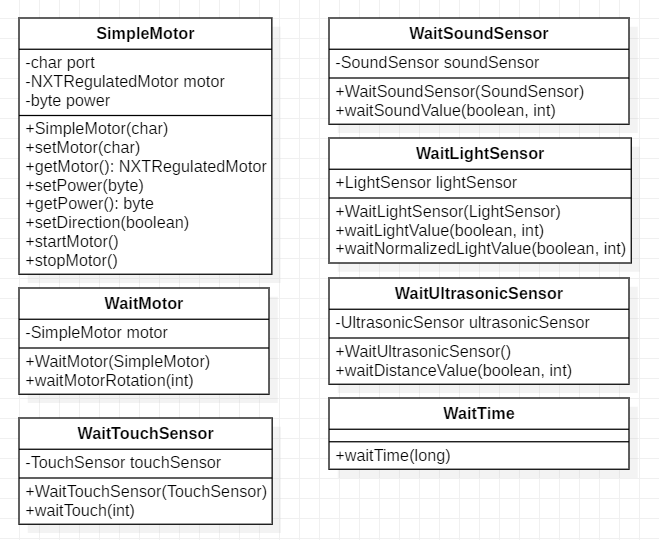
Attributi:

* SoundSensor soundSensor
  + Contiene il riferimento a un oggetto di tipo SoundSensor

Metodi:

* WaitSoundSensor (SoundSensor soundSensor)
  + Costruttore, permette di istanziare un oggetto WaitSoundSensor, per istanziarlo è necessario un oggetto SoundSensor.
* waitSoundValue (Boolean sign, int value)
  + Questo metodo permette di aspettare un determinato valore del sensore di suono prima di effettuare altre operazioni, il parametro sign determina se la soglia deve essere maggiore o minore rispetto al valore value.

## WaitLightSensor

La classe WaitLightSensor permette di aspettare che il sensore di luce vede una certa soglia di luce prima di effettuare altre operazioni.  
La classe è strutturata nel seguente modo:

Attributi:

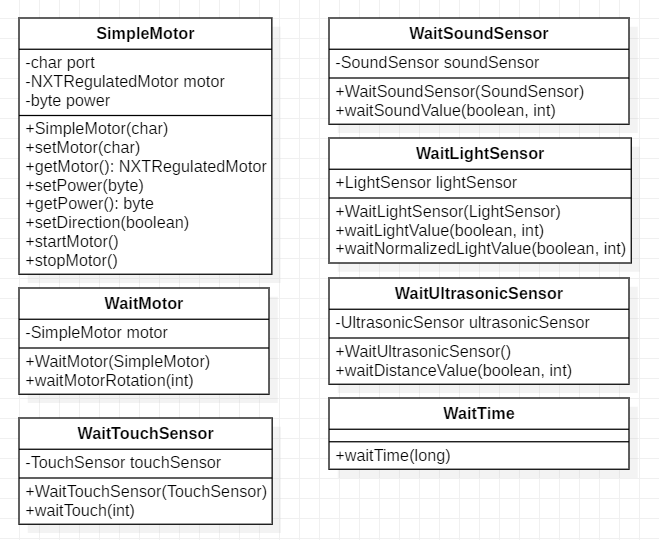
* LightSensor lightSensor
  + Contiene il riferimento a un oggetto di tipo LightSensor.

Metodi:

* WaitLightSensor (LightSensor lightSensor)
  + Costruttore, permette di istanziare un oggetto WaitLightSensor, per istanziarlo è necessario un oggetto LightSensor.
* waitLightValue(boolean sign, int value)
  + Questo metodo permette di aspettare un determinato valore del sensore di luce prima di effettuare altre operazioni, il parametro sign determina se la soglia deve essere maggiore o minore rispetto al valore value.
* waitNormalizedLightValue(boolean sign, int value)
  + Questo metodo permette di aspettare un determinato valore del sensore di luce, rispetto al metodo precedente aspetta il valore normalizzato (un valore da 0 a 1023), il parametro sign determina se la soglia deve essere maggiore o minore rispetto al valore value.

# Classi di Test

## SimpleMotor

La classe SimpleMotor è una classe che permette di gestire un singolo motore.  
La classe è strutturata nel modo seguente:

Attributi:

* Char port
  + La porta a cui è stato attaccato il motore
* NXTRegulatedMotor motor
  + Riferimento al Motore della classe NXTRegulatedMotor
* Byte power
  + La potenza del motore

Metodi:

* SimpleMotor (char port)
  + Costruttore, permette di istanziare un oggetto di tipo SimpleMotor, per istanziarlo è necessario indicare la porta a cui è collegato
* setMotor(char port)
  + Setta il motore in base alla porta indicata
* getMotor()
  + Ritorna il motore della classe NXTRegulatedMotor
* setPower(byte power)
  + Setta il valore della potenza del motore
* getPower()
  + Ritorna la potenza del motore
* setDirection(boolean direction)
  + Setta la direzione (forward o backward, avanti o indietro) in base al valore passato
* startMotor()
  + Permette di far partire il motore.
* stopMotor()
  + Permette di fermare il motore.