

Peter Catania

Cannobio, 05.09.2018

Lavori svolti

Sito web con pagina di registrazione e verifica dati

Orario	Lavoro svolto
13:15 - 14:45	Oggi ci anno presentato il modulo 306 e cosa andremo fare come primo progetto. Abbiamo visto cosa è richiesto e le vari cose da trattare nella documentazione. E abbiamo fissato come deve essere strutturato il diario del progetto e cosa dobbiamo metterci dentro. Infine ho aggiunto il diario sul git mandando a tutti l'email.

Problemi riscontrati e soluzioni adottate

Punto della situazione rispetto alla pianificazione

Programma di massima per la prossima giornata di lavoro

Cannobio, 05.09.2018

Lavori svolti

Sito web con pagina di registrazione e verifica dati

Orario	Lavoro svolto
13:15 - 14:45	Oggi ci hanno presentato il modulo 306 e cosa andremo fare come primo progetto. Abbiamo visto cosa è richiesto e le varie cose da trattare nella documentazione. E abbiamo fissato come deve essere strutturato il diario del progetto e cosa dobbiamo metterci dentro. Infine ho aggiunto il diario sul git mandando a tutti l'email.
13:15 - 15:30	Spiegazione sulla documentazione, in particolare sull'introduzione, progettazione, implementazione, test, consuntivo, conclusioni, bibliografia e allegati.
15:30 - 16:15	Spiegazione del ciclo di vita del progetto: <ul style="list-style-type: none">- Quando nasce e quando muore- Il modello iterativo o incrementale- Il modello incrementale- Il modello evolutivo- Il modello cascata e le sue fasi- Analisi dei requisiti- Progettazione- Implementazione e test di modulo- Integrazione e test sistema- Manutenzione Spiegazione dell'analisi dei requisiti

Problemi riscontrati e soluzioni adottate

**Punto della situazione rispetto alla
pianificazione**

**Programma di massima per la prossima
giornata di lavoro**

Cannobio, 05.09.2018

Lavori svolti

Sito web con pagina di registrazione e verifica dati

Orario	Lavoro svolto
13:15 - 14:45	<p>Oggi ci hanno presentato il modulo 306 e cosa andremo fare come primo progetto.</p> <p>Abbiamo visto cosa è richiesto e le varie cose da trattare nella documentazione.</p> <p>E abbiamo fissato come deve essere strutturato il diario del progetto e cosa dobbiamo metterci dentro.</p> <p>Infine ho aggiunto il diario sul git mandando a tutti l'email.</p>
13:15 - 15:30	<p>Spiegazione sulla documentazione, in particolare sull'introduzione, progettazione, implementazione, test, consuntivo, conclusioni, bibliografia e allegati.</p>
15:30 - 16:15	<p>Spiegazione del ciclo di vita del progetto:</p> <ul style="list-style-type: none">- Quando nasce e quando muore- Il modello iterativo o incrementale- Il modello incrementale- Il modello evolutivo- Il modello cascata e le sue fasi- Analisi dei requisiti- Progettazione- Implementazione e test di modulo- Integrazione e test sistema- Manutenzione <p>Spiegazione dell'analisi dei requisiti</p>

13:15 - 14:45	Fatte domande sul quaderno dei compiti, con conseguenti risposte e chiarimenti. Iniziata analisi del progetto.
---------------	---

Problemi riscontrati e soluzioni adottate

Punto della situazione rispetto alla pianificazione

Programma di massima per la prossima giornata di lavoro

Peter Catania

Cannobio, 14.09.2018

Lavori svolti

Sito web con pagina di registrazione e verifica dati

Orario	Lavoro svolto
13:15 - 16:30	Oggi ho fatto l'analisi e specifica dei requisiti, dettagliando le varie fasi del progetto, e le priorità da compiere per prime.

Problemi riscontrati e soluzioni adottate

Punto della situazione rispetto alla pianificazione

Programma di massima per la prossima giornata di lavoro

Peter Catania

Cannobio, 19.09.2018

Lavori svolti

Sito web con pagina di registrazione e verifica dati

Orario	Lavoro svolto
13:15 - 14:45	Ho corretto le tabelle dei requisiti con l'aiuto del professore, e ho iniziato e completato il gantt del progetto.

Problemi riscontrati e soluzioni adottate

Punto della situazione rispetto alla pianificazione

Programma di massima per la prossima giornata di lavoro

Peter Catania

Cannobio, 21.09.2018

Lavori svolti

Sito web con pagina di registrazione e verifica dati

Orario	Lavoro svolto
13:15 - 14:30	Fatto la teoria sulla progettazione
14:30 - 14:45	Sono andato avanti con l'analisi.
15:00 - 16:30	Sono andato avanti con la documentazione.

Problemi riscontrati e soluzioni adottate

Punto della situazione rispetto alla pianificazione

Programma di massima per la prossima giornata di lavoro

Peter Catania

Cannobio, 26.09.2018

Lavori svolti

Sito web con pagina di registrazione e verifica dati

Orario	Lavoro svolto
13:15 - 14:45	Fatto Verifica

Problemi riscontrati e soluzioni adottate

Punto della situazione rispetto alla pianificazione

Programma di massima per la prossima giornata di lavoro

Peter Catania

Cannobio, 28.09.2018

Lavori svolti

Sito web con pagina di registrazione e verifica dati

Orario	Lavoro svolto
13:15 - 16:30	Andato avanti con la progettazione: Completato 2.1 Design dell'architettura del sistema Completato 2.2 Design dei dati Iniziato 2.3 Design delle interfacce

Problemi riscontrati e soluzioni adottate

Punto della situazione rispetto alla pianificazione

Programma di massima per la prossima giornata di lavoro

Peter Catania

Cannobio, 28.09.2018

Lavori svolti

Sito web con pagina di registrazione e verifica dati

Orario	Lavoro svolto
13:15 - 14:05	Assistito alla presentazione e spiegazione della progettazione, implementazione e test.
14:05 - 14:45	Andato avanti con la progettazione.

Problemi riscontrati e soluzioni adottate

Punto della situazione rispetto alla pianificazione

Programma di massima per la prossima giornata di lavoro

Peter Catania

Cannobio, 28.09.2018

Lavori svolti

Sito web con pagina di registrazione e verifica dati

Orario	Lavoro svolto
13:15 - 14:45	Fatto il primo protocollo di test
15:00 - 16:30	Andato avanti con la progettazione

Problemi riscontrati e soluzioni adottate

Punto della situazione rispetto alla pianificazione

Programma di massima per la prossima giornata di lavoro

Peter Catania

Cannobio, 10.10.2018

Lavori svolti

Sito web con pagina di registrazione e verifica dati

Orario	Lavoro svolto
13:15 - 14:45	Fatta progettazione, migliorato il gantt, sistemate imperfezioni parti di progettazione e di analisi.

Problemi riscontrati e soluzioni adottate

Punto della situazione rispetto alla pianificazione

Programma di massima per la prossima giornata di lavoro

Peter Catania

Cannobio, 10.12.2018

Lavori svolti

Sito web con pagina di registrazione e verifica dati

Orario	Lavoro svolto
13:15 - 14:45	Terminata progettazione. Acquisizione template per pagine web: FONTE: https://colorlib.com/wp/free-bootstrap-registration-forms/
15:00 - 16:30	Iniziata implementazione, fatta interfaccia registrazione e pagina benvenuto.

Problemi riscontrati e soluzioni adottate

--	--

Punto della situazione rispetto alla pianificazione

--	--

Programma di massima per la prossima giornata di lavoro

--	--

Peter Catania

Cannobio, 10.17.2018

Lavori svolti

Sito web con pagina di registrazione e verifica dati

Orario	Lavoro svolto
13:15 - 14:45	Continuato il form di registrazione, grafica, validazione, il bottone “Cancella” e “Registra”.

Problemi riscontrati e soluzioni adottate

--	--

Punto della situazione rispetto alla pianificazione

--	--

Programma di massima per la prossima giornata di lavoro

--	--

Peter Catania

Cannobio, 10.24.2018

Lavori svolti

Sito web con pagina di registrazione e verifica dati

Orario	Lavoro svolto
13:15 - 14:45	Concluso il form di registrazione

Problemi riscontrati e soluzioni adottate

--	--

Punto della situazione rispetto alla pianificazione

--	--

Programma di massima per la prossima giornata di lavoro

--	--

Peter Catania

Cannobio, 10.26.2018

Lavori svolti

Sito web con pagina di registrazione e verifica dati

Orario	Lavoro svolto
13:15 - 14:45	Fatta tabella che richiama dati e li invia.

Problemi riscontrati e soluzioni adottate

--	--

Punto della situazione rispetto alla pianificazione

--	--

Programma di massima per la prossima giornata di lavoro

--	--

Peter Catania

Cannobio, 07.11.2018

Lavori svolti

Sito web con pagina di registrazione e verifica dati

Orario	Lavoro svolto
13:15 - 14:45	Fatto quando va all'ultima pagina salva i dati nel csv e mostra tutte le registrazioni giornaliere, a partire dalla più nuova, Fatto che non si può tornare indietro con le freccette del browser, ma solo con i bottoni.

Problemi riscontrati e soluzioni adottate

--	--

Punto della situazione rispetto alla pianificazione

--	--

Programma di massima per la prossima giornata di lavoro

--	--

Peter Catania

Cannobio, 09.11.2018

Lavori svolti

Sito web con pagina di registrazione e verifica dati

Orario	Lavoro svolto
13:15 - 14:30	Fatta presentazione progetto. Sistemati commenti codice. Finita documentazione. Sistemato il codice e alcune imprecisioni.

Problemi riscontrati e soluzioni adottate

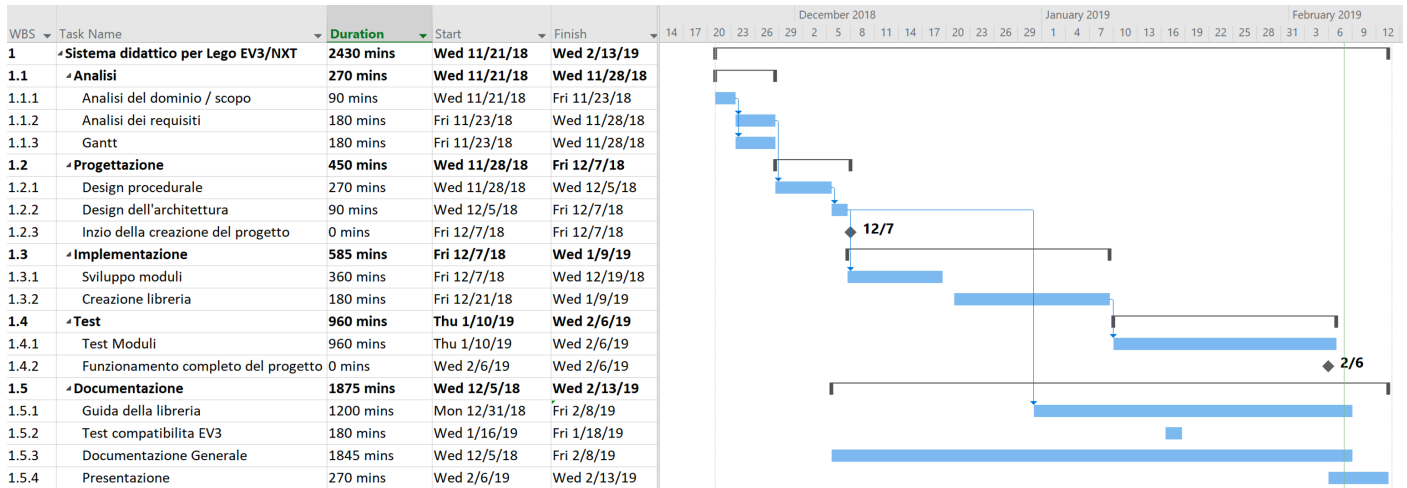
--	--

Punto della situazione rispetto alla pianificazione

--	--

Programma di massima per la prossima giornata di lavoro

--	--



Guida all'installazione di LeJOS

Sommario

Requisiti	3
Installazione LeJOS	3
Installazione Fantom Driver	5
Comandi per il caricamento dei file.....	5
Installazione plugin eclipse	6

Requisiti

Per l'installazione di LeJOS è necessario

- Robot Lego NXT
- Software di installazione di LeJOS
- JDK Java 32-bit
- Driver Fantom
- Cavo USB fornito con il robot NXT

Installazione LeJOS

L'installazione del firmware LeJOS sul blocchetto NXT è molto semplice.

Prima di tutto è necessario scaricare l'ultima versione del pacchetto LeJOS (in questo caso la versione 0.9.1) da questo indirizzo: <https://sourceforge.net/projects/nxt.lejos.p/files/>

SourceForge

ARMEDANGELS | Fair Fashion

leJOS NXJ

Java for the LEGO NXT Brick

Brought to you by: [ibagnail](#), [gloomyandy](#), [jhsolori](#), [lgriffiths](#), and 2 others

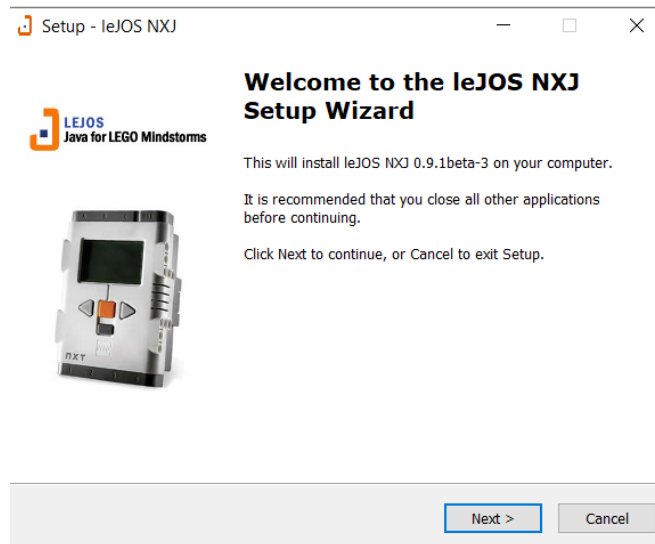
Download Latest Version
leJOS NXJ 0.9.1beta-3 win32_setup.exe (21.9 MB)

Get Updates

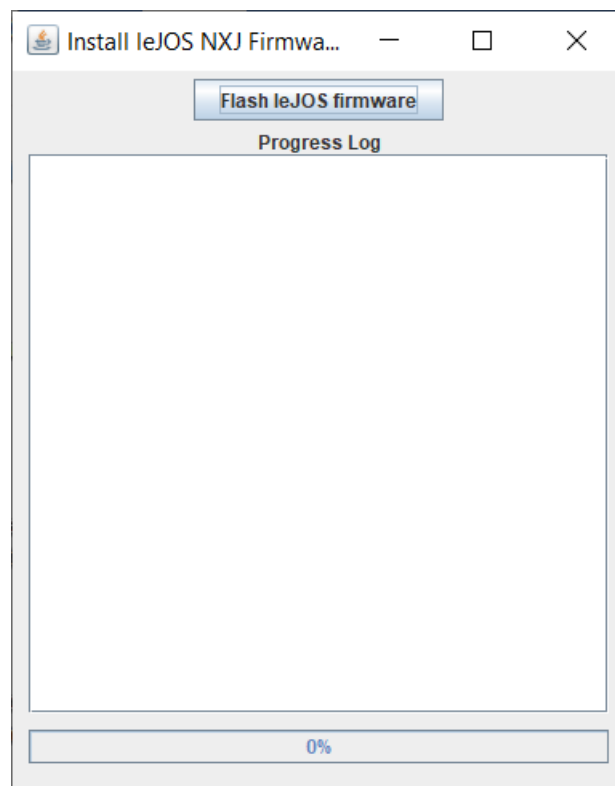
Name	Modified	Size	Downloads / Week
0.9.1beta-3	2015-02-03		90
0.9.1beta-2	2015-02-03		0
0.9.1beta-1	2015-02-03		0
0.9.1beta	2015-02-03		0

Scaricare anche la versione più aggiornata di Java nella versione a 32 bit, è consigliata la versione 1.7 ma è possibile utilizzare anche la versione 1.8.

Dopo aver scaricato il pacchetto avviate l'installazione di LeJOS sul vostro computer e accettate tutto quello che viene richiesto.



Appena viene completata l'installazione di LeJOS sul computer è possibile collegare tramite cavo USB il blocchetto NXT. Aprire l'applicativo NXJ Flash per immettere il firmware nel blocchetto, cliccare poi su "Flash LeJOS firmware" e attendere la fine dell'installazione.



Installazione Fantom Driver

Per poter usare il robot Lego NXT è necessario installare il driver Fantom (usato per la comunicazione tra il blocchetto e il computer), che normalmente viene installato con il software grafico Lego Mindstorms.

Il link per scaricare il Driver è il seguente: <https://www.lego.com/r/www/r/mindstorms/-/media/franchises/mindstorms%202014/downloads/firmware%20and%20software/nxt%20software/nxt%20fantom%20drivers%20v120.zip?l.r2=-964392510>

Verrà scaricato un archivio zip contenente gli installer per Windows e MacOS.

Comandi per il caricamento dei file

Per caricare i programmi sul blocchetto NXJ é necessario che questi vengano prima compilati con java e poi dal compilatore di LeJOS essendo che sfrutta librerie proprie per lo sviluppo del robot.

I comandi necessari sono:

- `Nxjc <java-files>` → compila il programma con il compilatore di Java
- `nxjlink -v -o HelloWorld.nxj HelloWorld` → viene creato un file nxj che fa il link a una determinata classe di Java. In questo viene fatto il link di una ipotetica classe HelloWorld
- `nxjupload <filename>.nxj` → permette l'upload del file creato sull'blocchetto
- `nxj <Nome classe java>` → permette di fare il link e l'upload di una classe di Java, corrisponde a fare nxjlink e nxjupload assieme

Installazione plugin eclipse

Gli sviluppatori di LeJOS hanno creato un plugin per l'IDE Eclipse per permettere la progettazione e lo sviluppo dei programmi per NXT.

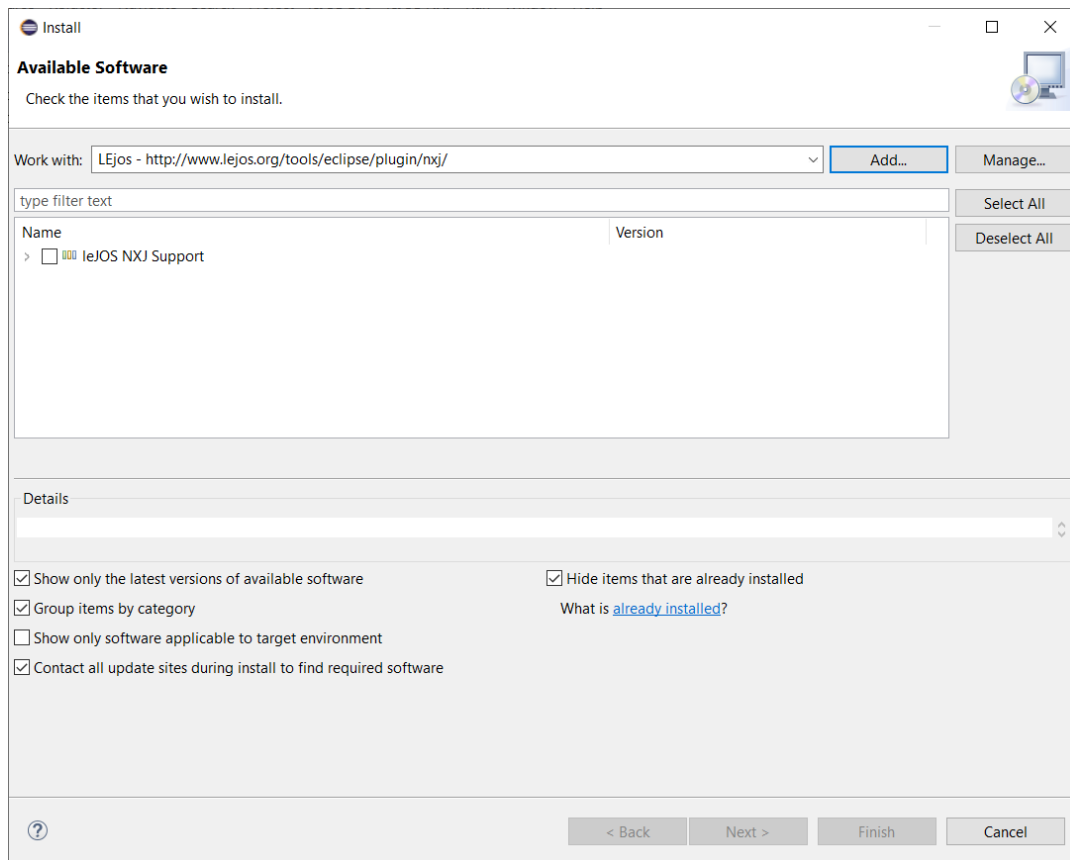
Per installarlo aprire l'IDE Eclipse, cliccare su "Help" e poi su Install New Software, si aprirà una finestra in cui è possibile installare dei software per Eclipse.


Per installare il plugin cliccare su "Add" e inserire il nome e l'URL del plugin.

Il link del plugin di LeJOS NXT è il seguente: <http://www.lejos.org/tools/eclipse/plugin/nxj/>

Il plugin ora appare nella lista, selezionatelo e poi cliccate "Next" per installarlo.

Ora è possibile, quando si crea un nuovo progetto, di includere e usare direttamente la libreria di LeJOS NXT.



	Scuola Arti e Mestieri Trevano
	Sezione informatica

Guida libreria LeJOS NXT

Titolo del progetto: Esempio di documentazione
Alunno/a: Andrea Rauso, Peter Catania
Classe: SAM I3AC
Anno scolastico: 2018/2019
Docente responsabile: Francesco Mussi, Luca Muggiasca, Adriano Barchi, Massimo Sartori

Sommario

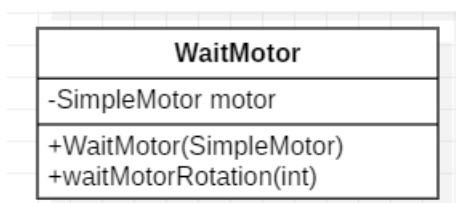
1	Classi di Wait.....	3
1.1	WaitMotor.....	3
1.2	WaitTime.....	3
1.3	WaitTouchSensor	3
1.4	WaitUltrasonicSensor	4
1.5	WaitSoundSensor.....	4
1.6	WaitLightSensor	5
2	Classi di Test.....	6
2.1	SimpleMotor	6

1 Classi di Wait

1.1 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

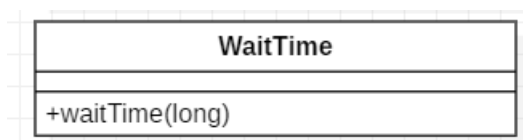
Metodi:

- 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

1.2 WaitTime

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

La classe è strutturata nel seguente modo:



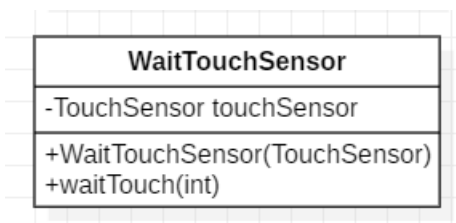
Metodi:

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

1.3 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

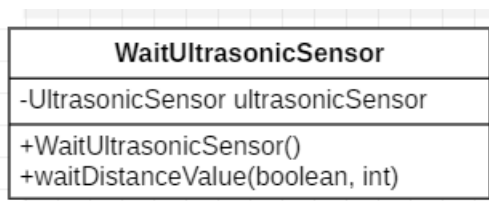
Metodi:

- 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)

1.4 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

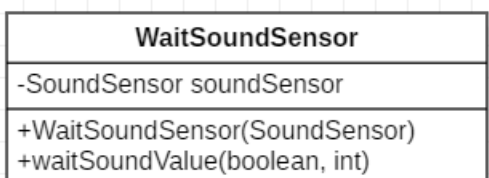
Metodi:

- 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.

1.5 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:

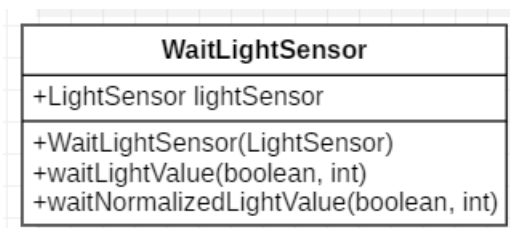
- WaitSoundSensor (SoundSensor soundSensor)

- 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.

1.6 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.

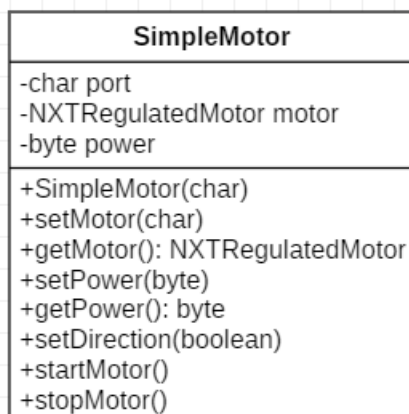
Metodi:

- 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.

2 Classi di Test

2.1 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.