Sistema didattico per Lego EV3/NTX con libreria e documentazione

1 Introduzione 3

1.1 Informazioni sul progetto 3

1.2 Abstract 3

1.3 Scopo 3

Analisi 4

1.4 Analisi del dominio 4

1.5 Analisi e specifica dei requisiti 4

1.6 Pianificazione 6

1.7 Analisi dei mezzi 6

1.7.1 Software 6

1.7.2 Hardware 6

2 Progettazione 6

2.1 Design dell’architettura del sistema 6

2.2 Design delle interfacce 6

2.3 Design procedurale 6

3 Implementazione 8

3.1 Classi Wait 8

3.1.1 WaitMotor 8

3.1.2 WaitTime 8

3.1.3 WaitLightSensor 8

3.1.4 WaitSoundSensor 8

3.1.5 WaitUltrasonicSensor 8

3.1.6 WaitTouchSensor 8

3.2 Classi di test 9

3.2.1 SimpleMotor 9

4 Test 9

4.1 Protocollo di test 9

4.2 Risultati test 9

4.3 Mancanze/limitazioni conosciute 9

5 Consuntivo 10

6 Conclusioni 10

6.1 Sviluppi futuri 10

6.2 Considerazioni personali 10

7 Bibliografia 10

7.1 Sitografia 10

8 Allegati 10

# Introduzione

## Informazioni sul progetto

Progetto: Sistema didattico per Lego EV3/NTX con libreria e documentazione

Docente responsabile: Francesco Mussi, Luca Muggiasca, Adriano Barchi, Massimo Sartori

Componenti del gruppo: Andrea Rauso, Peter Catania

Luogo di lavoro: Aula 417 Scuola arti e mestieri Trevano

Classe: I3AC

Materia: Modulo 306

Data di Inizio: 14/11/2018

Data di fine: 08/02/2019

## Abstract

*If you want to develop a program for a Lego NXT robot, mostly beginner’s programmers use the graphic Mindstorms tool. However, as the complexity of the programs increase, is observable that becomes less intuitive how arrange the graphics programming blocks. A possible solution of this problem is the embrace of a programming language, like Java or a more specific language for robots like RobotC. Introduce a programming language to beginner’s is a step too high, because of this we are offering a simpler way to embrace it. With our library, we offer a more understandable method of programming, having made available simple constructs integrated with the Java programming language.*

## Scopo

Andrea: Lo scopo di questo progetto è di creare una piccola libreria per i robot Lego NXT scritta con il linguaggio Java, questa libreria ha lo scopo di semplificare la programmazione del Robot, andando a eliminare i limiti dell’uso del linguaggio grafico creato dalla Lego.  
Questa libreria viene messa a disposizione degli allievi del secondo anno di informatica che nel corso del secondo semestre parteciperanno alla WRO

Peter: Lo scopo di questo progetto è di creare una semplice libreria per i robot Lego NXT, utilizzando il linguaggio di programmazione Java. In pratica offriamo dei costrutti simili ai blocchetti grafici presenti su Mindstorms per rendere l’esperienza di sviluppo più intuitiva e semplificata.  
Questa libreria viene messa a disposizione degli allievi del secondo anno di informatica che nel corso del secondo semestre parteciperanno alla WRO.

Lo scopo di questo progetto è offrire ai principianti della programmazione un modo semplice e funzionale di sviluppare programmi per i robot Lego NXT. Per ottenere un prodotto del genere, abbiamo creato una libreria nel linguaggio Java, che offre ideali costrutti per una programmazione intuitiva e semplifica per ogni possibile situazione.  
Nel nostro caso verrà messa a disposizione degli allievi del secondo anno di informatica che nel corso del secondo semestre utilizzeranno per partecipare alla WRO.

## Analisi

## Analisi del dominio

Oggigiorno molte persone che vogliono avvicinarsi alla programmazione utilizzando i Robot Lego NXT, utilizzano lo strumento grafico creato dalla Lego.  
Con questo metodo la maggior parte dei programmi possono essere sviluppati con quasi nessuna difficoltà, ma se si voule fare qualcosa di molto più complesso diventa molto meno intuitivo.

I principali utenti che si avvicinano alla programmazione con i robot della Lego sono principalmente i ragazzi che hanno iniziato i loro studi nell’ambito informatico.

## Analisi e specifica dei requisiti

Come obbiettivo supplementare ci è stato richiesto di testare la compatibilità del Robot EV3 su diversi sistemi operativi quali Windows 7, Windows 8.1, MacOS e Linux. Questo perché abbiamo riscontrato che il sistema operativo Windows 10 non è riuscito ad individuare e ad interfacciarsi con il blocchetto EV3 e quindi non è stato possibile caricare le classi del progetto.

|  |  |
| --- | --- |
| **ID: REQ-001** | |
| **Nome** | Creazione Libreria |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Si necessitano i permessi di root / Dipende dal requisito REQ-001 (Creazione DB) |
| **Sotto requisiti** | |
| **001** |  |

|  |  |
| --- | --- |
| **ID: REQ-002** | |
| **Nome** | Creazione Guide |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Si necessitano i permessi di root / Dipende dal requisito REQ-001 (Creazione DB) |
| **Sotto requisiti** | |
| **001** | Deve essere creata una guida all’installazione di LeJOS |
| **002** | Deve essere creata una guida per l’utilizzo della libreria |

|  |  |
| --- | --- |
| **ID: REQ-00** | |
| **Nome** | Compatibilita del sistema EV3 |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Compito supplementare che va a sostituire un progetto da creare con la libreria prodotta |
| **Sotto requisiti** | |
| **001** | Testare la compatibilità del Robot EV3 su diversi sistemi operativi |

## Pianificazione

## Analisi dei mezzi

Elencare e descrivere i mezzi disponibili per la realizzazione del progetto. Ricordarsi di sempre descrivere nel dettaglio le versioni e il modello di riferimento.

### Software

I software che sono stati usati sono i seguenti:

* Eclipse
* Plugin LeJOS per Eclipse
* StarUML
* LeJOS
* Microsoft Project
* Microsoft Word

### Hardware

L’hardware su cui è stato sviluppato questo sono i seguenti:

Portatile ASUS Vivobook Pro 15  
Portatile Apple MacBook Pro 15

Oltre al Robot Lego NXT sono stati utilizzati i seguenti sensori e attuatori: Sensore di tatto, Sensore di suono, Sensore Ultrasuoni, sensore di luce e il motore principale.

Questo progetto può girare su qualsiasi macchina che abbia almeno una porta USB per poter collegare il Robot Lego NXT, deve essere possibile anche installare l’IDE Eclipse grazie al quale, tramite il plugin dedicato, è possibile creare direttamente programmi per il Robot.

# Progettazione

## Design dell’architettura del sistema

# Implementazione

## Classi Wait

All’interno di questa sezione sono elencati tutte le classi di attesa, una per ogni sensore e attuatore più il tempo, ogni classe comprende sempre il metodo di attesa e eventualmente il costruttore.

### WaitMotor

La classe WaitMotor è una claase che permette all’utente di aspettare che il robot compia determinate azioni con i motori, all’interno della classe è contenuto il costruttore che riceve un oggetto di tipo SimpleMotor, contiene i metodi waitRotation, waitDegrees e waitTime che rispettivamente aspettano le rotazioni, i gradi o il tempo del motore.

### WaitTime

La classe WaitTime come dice il nome aspetta un determinato lasso di tempo, all’interno della classe è contenuto il metodo waitTime che riceve un valore long di attesa e ricontrolla ogni 10 millisecondi il tempo passato.

### WaitLightSensor

La classe WaitLightSensor è una classe che permette di aspettare un valore dal sensore di luce, all’interno della classe è contenuto un costruttore, il metodo getLightValue che riceve un valore int da trovare e un booleano per indicare se il valore deve essere maggiore o minore, contiene anche il metodo getNormalizedLightValue che rispetto al primo aspetta un valore normalizzato ( da 0 a 1023).

### WaitSoundSensor

La classe WaitSoundSensor è una classe che permette di aspettare un valore dal sensore di suono, all’interno della classe è contenuto un costruttore, il metodo getSoundValue che riceve un valore int da trovare e un booleano per indicare se il valore deve essere maggiore o minore.

### WaitUltrasonicSensor

La classe WaitUltrasonicSensor è una classe che permette di aspettare una distanza dal sensore a ultrasuoni, all’interno della classe è contenuto un costruttore, il metodo getDistance che riceve un valore int da trovare e un booleano per indicare se il valore deve essere maggiore o minore.

### WaitTouchSensor

La classe WaitTouchSensor è una classe che permette di aspettare un azione del sensore di tatto, all’interno della classe è contenuto il costruttore e il metodo waitTouch che riceve un int per iindicare l’azione che deve aspettare (0 🡪 premuto, 1 🡪 rilasciato, 2 🡪 cliccato).

## Classi di test

### SimpleMotor

La classe SimpleMotor ricrea la classe del motore contenuta nella libreria di LeJOS tenendo una versione semplificata

# Test

## Protocollo di test

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-001  REQ-012 | **Nome:** | Import a card with KIC, KID and KIK keys, but not shown with the GUI |
| **Descrizione:** | Import a card with KIC, KID and KIK keys with no obfuscation, but not shown with the GUI | | |
| **Prerequisiti:** | Store on local PC: Profile\_1.2.001.xml (appendix n\_n) and Cards\_1.2.001.txt (appendix n\_n).  PIN (OTA\_VIEW\_PIN\_PUK\_KEY) and ADM (OTA\_VIEW\_ADM\_KEY) user right not set. | | |
| **Procedura:** | 1. Go to “Cards manager” menu,  in main page click “Import Profiles” link, Select the “1.2.001.xml” file, Import the Profile 2. Go to “Cards manager” menu,  in main page click “Import Cards” link, Select the “1.2.001.txt” file, Delete the cards,  Select the “1.2.001.txt” file, Import the cards 3. Research the “41795924770” Card, Click the imsi card link Check the card details 4. Execute the SQL: SELECT imsi, dir, keyset, cntr, rawtohex(kickey), rawtohex(kidkey), rawtohex(kikkey), rawtohex(chv), rawtohex(dap)FROM otacardkey a where imsi='340041795924770' ORDER BY keyset; | | |
| **Risultati attesi:** | Keys visible in the DB (OtaCardKey) but not visible in the GUI (Card details) | | |

## Risultati test

## Mancanze/limitazioni conosciute

Riguardo alla compatibilità del Robot Lego EV3, non siamo riusciti a trovare un sistema operativo che riuscisse a identificare il Robot. Crediamo che il problema risieda nel driver RNDIS, che permette la comunicazione tramite USB, ma che non è stato possibile installarlo per incompatibilità.

# Consuntivo

(foto)

# Conclusioni

Grazie al nostro prodotto lo sviluppo di programmi per il Robot Lego NXT viene semplificato così da aiutare i giovani che andranno a partecipare alla competizione WRO.

## Sviluppi futuri

I metodi prodotti nella libreria son semplici

## Considerazioni personali

Da questo progetto si può dire che, nonostante i problemi avuti con il Robot Lego EV3, si può imparare quanto è importante la collaborazione in coppia.

Cosa ho imparato in questo progetto? ecc

# Bibliografia

## Sitografia

1. <http://www.lejos.org/> *LeJOS, Java for Lego MindStorms*
2. <http://www.lejos.org/ev3/docs/> *Overview (leJOS EV3 API Documentation)*
3. <https://lejos.sourceforge.io/forum/> *leJOS – Forum Index*

# Allegati

1. Diari di Progetto
2. Quaderno dei compiti
3. Guida installazione LeJOS
4. Guida Libreria
5. Prodotto
   1. Github: <https://github.com/andrearauso/Progetto-2>