

**Università Ca'Foscari di Venezia**

***Dipartimento di Scienze Ambientali, Informatica e Statistica***

**Corso di Ingegneria del Software**

**A.A. 2018-2019**

**Docente Responsabile : prof. Agostino Cortesi**

## **Documento di Progettazione**



**Gruppo LegoStoneng**

**864553 - Marco Barnà**

**862653 - Pasquale Soldano**


**865154 - Alvise Battistich**

**864881 - Riccardo Denticò**



# INDICE

---



1. Introduzione	3
1. Scopo del documento	3
2. Struttura del documento	3
2. Glossario	3
3. Modello di struttura del sistema	4
4. Modello di controllo	6
5. Diagrammi di stato	7
6. Interfaccia Grafica	8
1. Splash Screen	9
2. Home Page	10
3. Test	11
4. Advanced Actions	12
5. Connetti	13
6. Help	14
7. Faq	15
8. Manual	16
9. Contacts	17
10. Credits	18
11. Settings	19

# 1. Introduzione

---

## 1. Scopo del documento

L'obiettivo di questo documento è di fornire indicazioni riguardo la progettazione architettuale dell'applicazione del gruppo Lego StonEng. Ha lo scopo inoltre di specificare le modalità con le quali le funzionalità dell'applicazione verranno implementate.

## 2. Struttura del documento

- Glossario:** contiene una descrizione dettagliata dei termini utilizzati all'interno del documento. Questa sezione è predisposta per agevolare la lettura e la comprensione del documento anche per utenti non esperti.

- Modello e struttura del sistema:** contiene informazioni relative all'interazione fra le varie componenti (sottosistemi) presenti nel sistema. La buona collaborazione di queste componenti fra di loro permetteranno lo sviluppo più efficiente dell'applicazione.

- Modello di controllo:** contiene la descrizione relativa al tipo di controllo che serve a disciplinare le relazioni tra i sottosistemi presenti.

- Diagramma degli stati:** presenta tutti i vari diagrammi di stato utili a dare una descrizione delle relazioni del sistema o più in generale del sistema stesso.

- Interfaccia grafica:** sezione del documento contenente le principali schermate visibili da un primo prototipo di applicazione.

## 2. Glossario

---

- **Telegram:** è un servizio di messaggistica istantanea basato su cloud disponibile gratuitamente per diverse piattaforme.
- **Android:** è un sistema operativo per dispositivi mobili sviluppato da Google.
- **API**(Application Programming Interface):un insieme di procedure disponibili al programmatore.
- **Oreo:** è la versione 8.0/8.1 del sistema operativoAndroid.
- **IDE:** ambiente di sviluppo integrato che aiuta i programmatori nello sviluppo del codice sorgente dei programmi.
- **Android Studio:** Android Studio è un ambiente di sviluppo integrato (IDE) per lo sviluppo per la piattaformaAndroid.
- **Git:** è un software di controllo versione distribuito(VCS).
- **Github:** è un servizio di hosting per progetti software che utilizzano Git come sistema di controllo versione.
- **Commit:** è un insieme di modifiche che hanno portato l'applicazione ad un nuovo stato.
- **Bug:** errore nella scrittura del codice sorgente che comporta comportamenti anomali del software.
- **Alpha:** versione di un software in fase di sviluppo le cui funzionalità non sono ancora state implementare completamente, spesso queste versioni sono affette da bug.
- **Beta:** versione non definitiva di un software, già testata da esperti disponibile ad un numero maggiore di utenti.
- **Testing:** indica l'attività di verifica e collaudo del software.
- **Release:** è una specifica versione di un software resa disponibile ai suoi utenti finali. E' univocamente identificata da un numero in modo da distinguerla dalle release del software precedenti e future.

## 2. Glossario

---

- **App o Applicazione:** La parola app è una forma abbreviata di "applicazione". Di per sé il termine può essere utilizzato per indicare qualsiasi programma, indipendentemente dal supporto su cui è utilizzato. Nell'uso quotidiano, però, ci si riferisce con questa parola soprattutto alle app mobile, quindi a quelle per cellulari e tablet. Questi piccoli software condividono alcune caratteristiche particolari: sono in genere essenziali, leggeri ("pesano" pochi MB) e strutturati per garantire un'esperienza di utilizzo il più semplice e intuitiva possibile.
- **Software:** informazioni utilizzate dai sistemi informatici e memorizzate su supporti informatici. Tali informazioni sono, dunque, rappresentabili da programmi, dati, oppure da una combinazione delle due componenti.
- **Material design:** È un design sviluppato da Google. Le regole di progettazione del Material Design si concentrano su un maggiore uso di layout basati su una griglia, animazioni e transizioni ed effetti di profondità come l'illuminazione e le ombre.
- **Smartphone:** telefono cellulare con capacità di calcolo, di memoria e di connessione dati molto più avanzate rispetto ai normali telefoni cellulari, basato su un sistema operativo per dispositivi mobili.
- **Java:** un linguaggio di programmazione orientato agli oggetti a tipizzazione statica, specificatamente progettato per essere il più possibile indipendente dalla piattaforma di esecuzione.
- **End user:** utente finale che utilizza l'applicazione.
- **XML:** È un metalinguaggio per la definizione di linguaggi di markup, ovvero un linguaggio marcatore basato su un meccanismo sintattico che consente di definire e controllare il significato degli elementi contenuti in un documento o in un testo.
- **UML:** è un linguaggio di modellazione che serve a specificare, costruire, visualizzare e documentare gli artefatti di un sistema.
- **Crash:** quando un'applicazione va in crash significa che si interrompe improvvisamente.

### 3. Modello di struttura del sistema

L'applicazione è strutturata secondo un modello client-server. Nello specifico Three-tier è un'architettura client-server in cui l'interfaccia utente, i processi logico funzionali ("regole aziendali"), l'archiviazione informatica dei dati e l'accesso ai dati sono sviluppate e mantenute come moduli indipendenti; la maggior parte delle volte su piattaforme separate.

Il three-tier è un modello di architettura software e allo stesso tempo uno schema di progettazione software.

Oltre ai vantaggi abituali di software modulare con interfacce ben definite, l'architettura three-tier è destinata a consentire a qualsiasi dei tre livelli di essere aggiornati o sostituiti indipendentemente dal cambiamento di requisiti o tecnologia. Ad esempio, un cambiamento di sistema operativo nel livello di presentazione interesserebbe solo il codice di interfaccia utente.

In genere, l'interfaccia utente viene eseguita su un desktop PC o workstation e utilizza un'interfaccia utente grafica standard. La logica di processo funzionale può essere costituita da uno o più moduli separati in esecuzione su una workstation o applicazioni server, e un RDBMS in un database server o mainframe contiene i dati di archiviazione logica del computer. Il livello intermedio può essere anche multi-tier (in questo caso l'architettura complessiva si chiama "n-tier architecture").

(fonte: [https://it.wikipedia.org/wiki/Architettura\\_multi-tier](https://it.wikipedia.org/wiki/Architettura_multi-tier) )

Three-tier architecture ha i seguenti tre livelli:

- **Livello di presentazione**

Questo è il livello più alto dell'applicazione. Il livello di presentazione mostra le informazioni relative a servizi come distanza percorsa dal robot, velocità ecc. Comunica con altri livelli attraverso i risultati di output al client e tutti gli altri livelli

- **Livello applicazione**

La logica di primo livello viene tirato fuori dal livello di presentazione e, come suo proprio livello, controlla la funzionalità di un'applicazione eseguendo elaborazioni dettagliate.

- **Livello dati**

Questo livello è costituito da una serie di librerie che permettono la comunicazione e la gestione di alcuni servizi che permettono al robot di funzionare. Qui le informazioni vengono memorizzate e recuperate.

Nel nostro caso l'applicazione è così suddivisa:

- **Livello di presentazione:** App (la nostra applicazione grafica)
- **Livello applicazione:** Algoritmi
- **Livello Dati:** Uso delle librerie per la comunicazione con l'EV3

## 4. Modello di controllo

---

Il modello di controllo che intendiamo utilizzare è orientato ad un sistema basato su un controllo centralizzato (dove un sottosistema ha il controllo globale e dà inizio e fine agli altri sottosistemi).

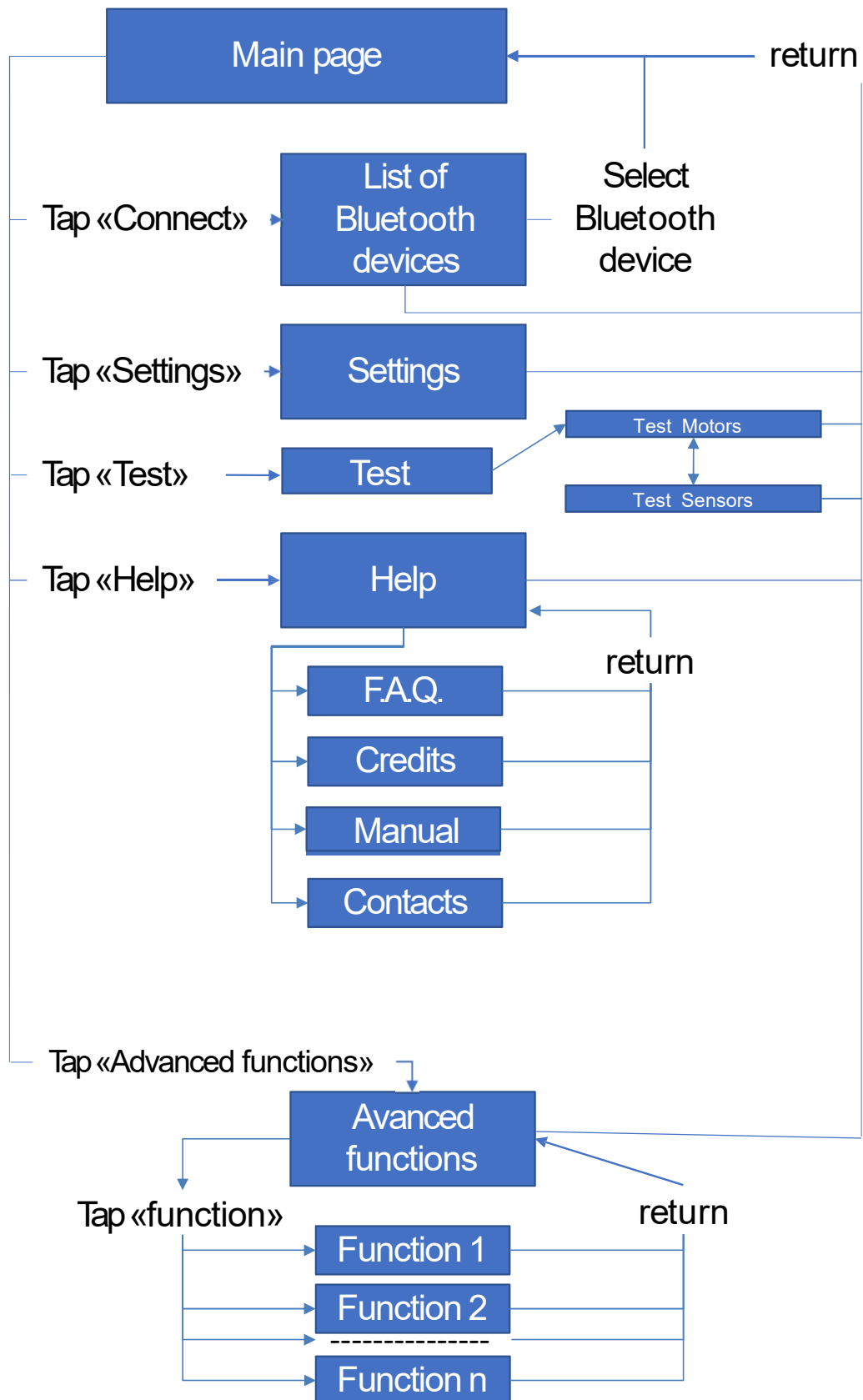
Nello specifico il modello usato è «**Manager**», dove una componente del sistema controlla l'interruzione, l'inizio e il coordinamento degli altri processi.

Nel nostro caso è molto importante perché dobbiamo verificare costantemente che tutti i componenti del robot funzionino indipendentemente gli uni dagli altri.

# 5. Diagrammi di stato

In questa sezione è indicato il diagramma di stato dell'applicazione.

## 5.1 Accesso standard





## 6. Interfaccia Grafica

---

### 6.1 Spash Screen

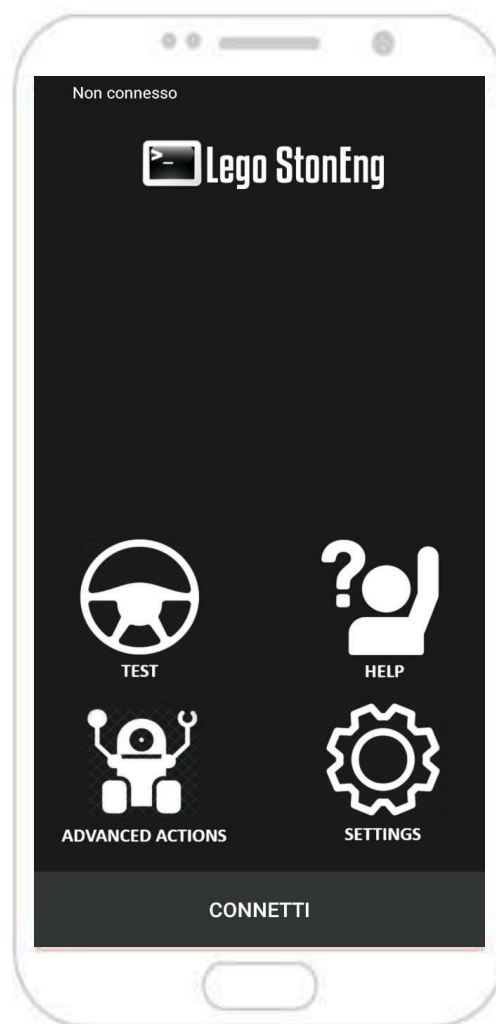
Questa schermata si presenta al caricamento dell'app-activity e mostra il nome del gruppo che l'ha sviluppata e un eventuale logo dell'app. Al termine dei 4 secondi, la schermata scompare e viene mostrata la schermata della home page.



## 6. Interfaccia Grafica

### 6.2 Home Page

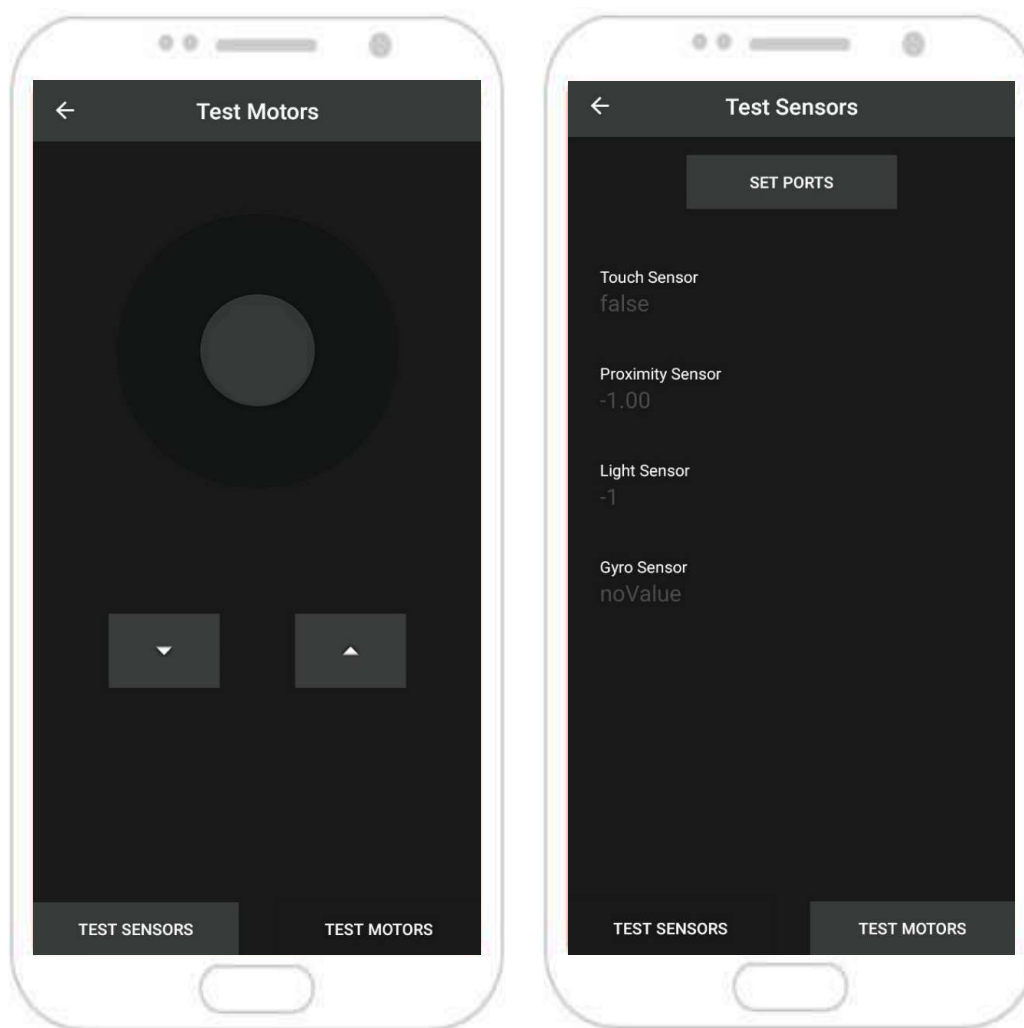
Questa schermata è la schermata principale dell'app. La Home Page mostra le sezioni principali (Test, Settings, Advanced Actions, Help) e il tasto connetti/disconnetti per la connessione al dispositivo. L'app ha più livelli di profondità, quindi si potrà tornare indietro di un livello in qualsiasi momento premendo il tasto indietro. Se si preme il tasto indietro dalla Home Page si uscirà dall'applicazione. Inoltre nella Home Page vengono mostrate in alto delle informazioni sullo stato della connessione al robot.



## 6. Interfaccia Grafica

### 6.3 Test

Questa schermata permette di testare il robot lego mindstorm, in particolare tutte le versioni che hanno i motori con le ruote. Dai pulsanti in fondo si può andare a cambiare tra Test Sensors e Test Motors. In Test Motors è possibile, attraverso un JoyPad mostrato sullo schermo, andare in tutte le direzioni e, con i due pulsanti, muovere i meccanismi del carrello. Invece in Test Sensors si possono settare le porte dell'EV3 attraverso il pulsante «Set Ports» ed avere una panoramica dello stato attuale dei sensori. Il tasto indietro permetterà di ritornare alla Home Page.



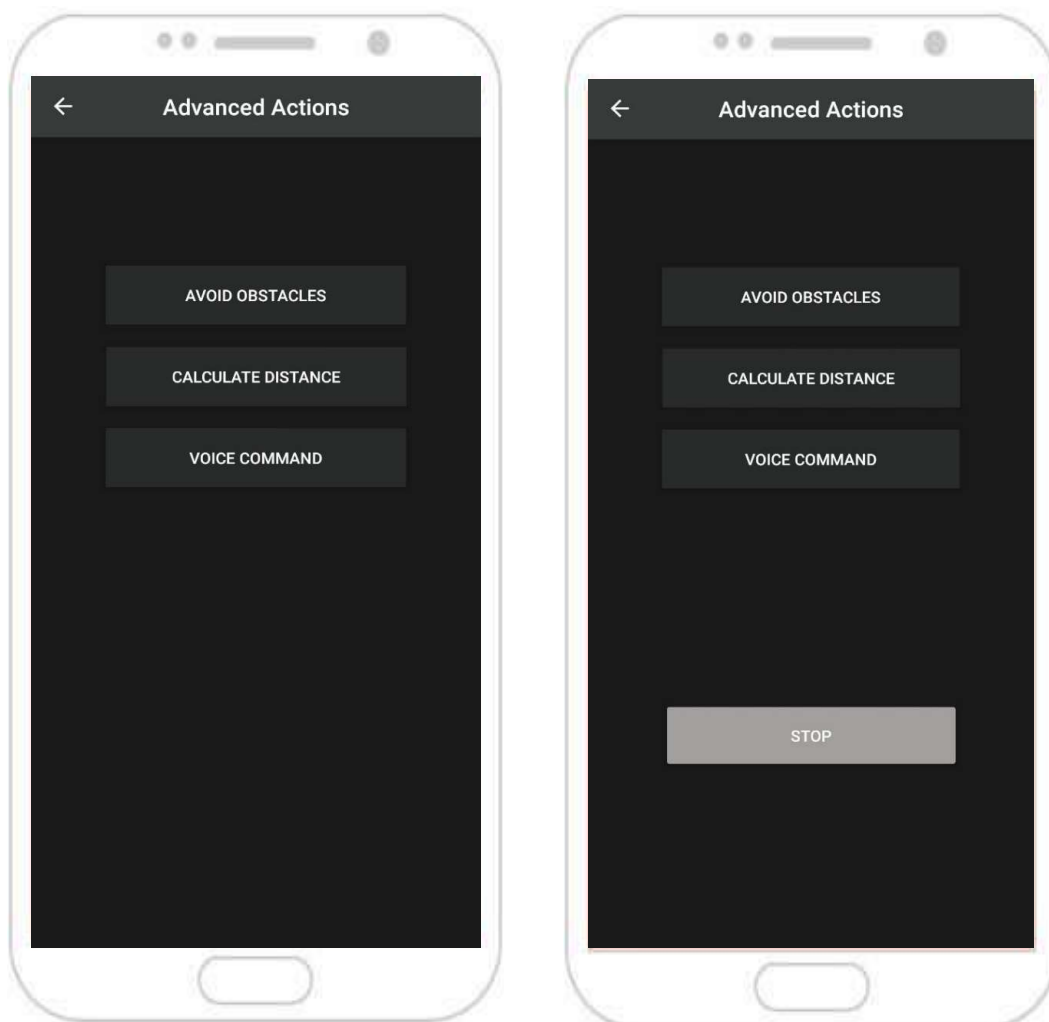
## 6. Interfaccia Grafica

### 6.4 Advanced Actions

Questa schermata mostra una lista di azioni programmate per il robot lego minstorm ev3. Per informazioni sulle azioni verrà tutto spiegato nella sezione help dell'app. Una volta scelta l'azione da fare al robot verranno mostrati dei risultati in basso sullo schermo. I risultati rimarranno fino al click della prossima azione. Le azioni sono degli algoritmi scritti in linguaggio Java che usano i sensori del robot (ogni azione utilizza almeno 3 sensori) per ritornare il risultato all'utente.

Dopo aver avviato un'azione comparirà un pulsante «STOP» per interromperne l'esecuzione.

Il tasto indietro permetterà di ritornare alla Home Page.

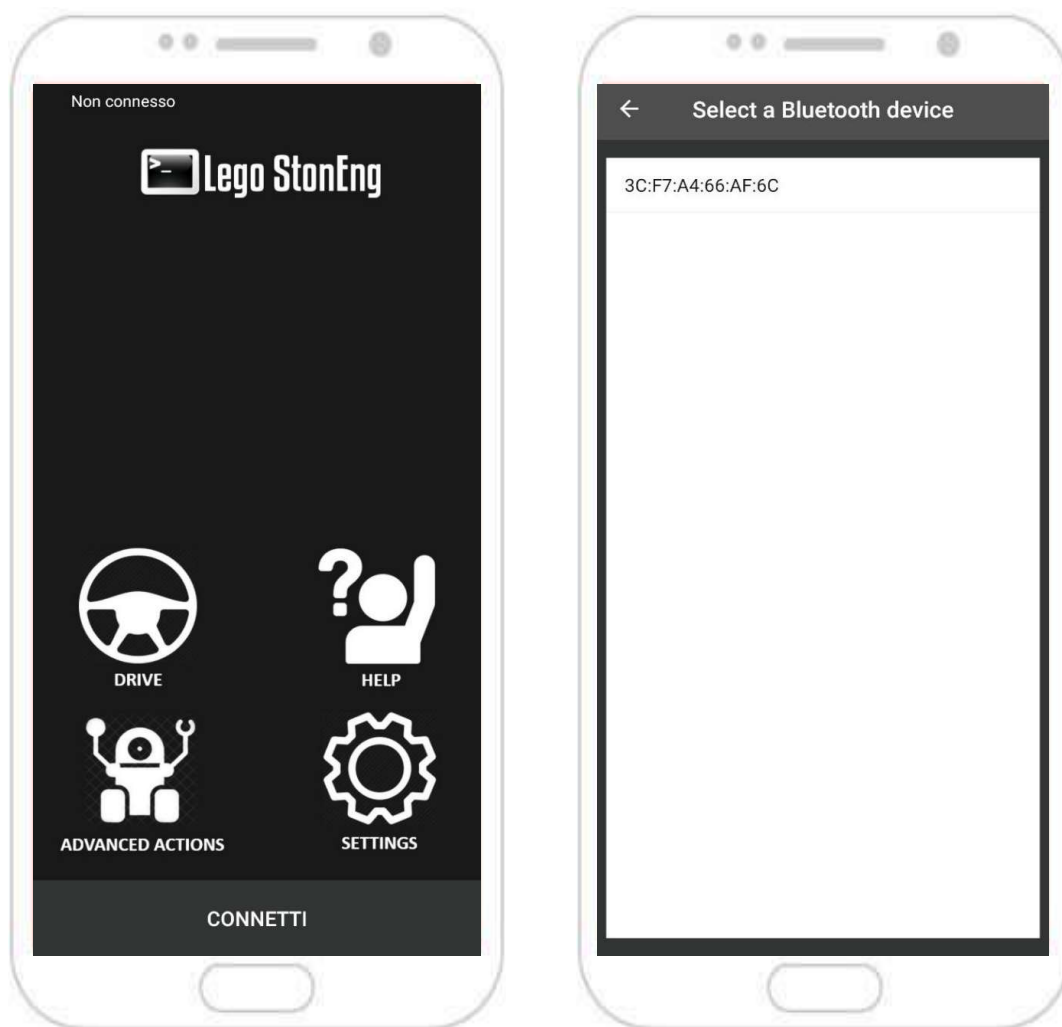


## 6. Interfaccia Grafica

### 6.5 Connetti

Questa schermata permette di connettere lo smartphone al robot lego mindstorm ev3 attraverso la connessione Bluetooth. Verrà mostrata una lista di dispositivi precedentemente associati al telefono. L'utente selezionerà il dispositivo con il nome del robot. Per riconoscere il robot basterà andare nelle impostazioni del robot e vederne il nome. Una volta stabilita la connessione si ritornerà alla schermata home dove verrà mostrata la scritta «connesso» sullo stato del robot, se la connessione non è andata a buon fine verrà mostrata la scritta «non connesso». Per stabilire una connessione è necessario attivare il Bluetooth sia del dispositivo che del robot.

Il tasto indietro permetterà di ritornare alla Home Page, annullando così la connessione al robot.

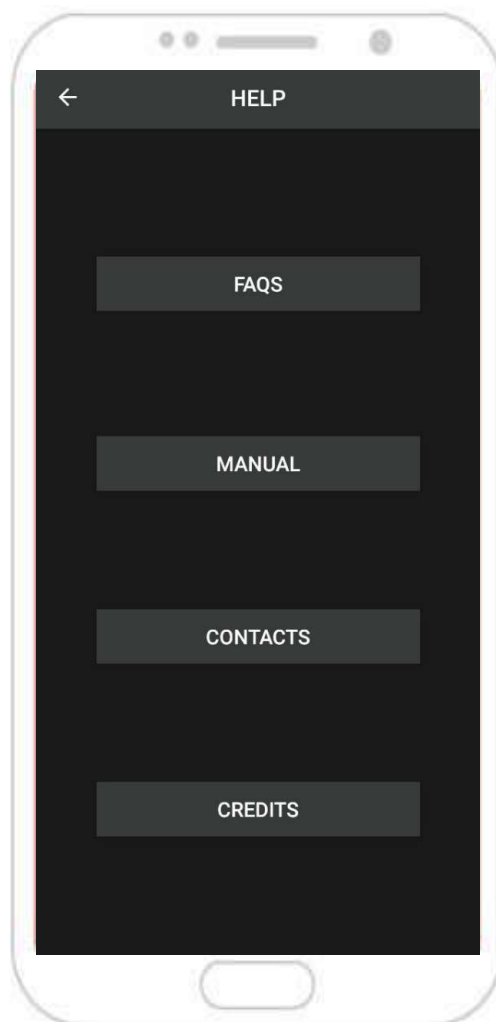


## 6. Interfaccia Grafica

### 6.6 Help

Questa schermata aiuta l'utente nell'uso dell'applicazione. Verrà mostrata una lista di categorie che comprende: "FAQs", "MANUAL", "CONTACTS" e "CREDITS".

L'utente, selezionando ciascuna di queste, avrà accesso alla relativa pagina contenente le informazioni richieste. Il tasto indietro permetterà di ritornare alla Home Page.

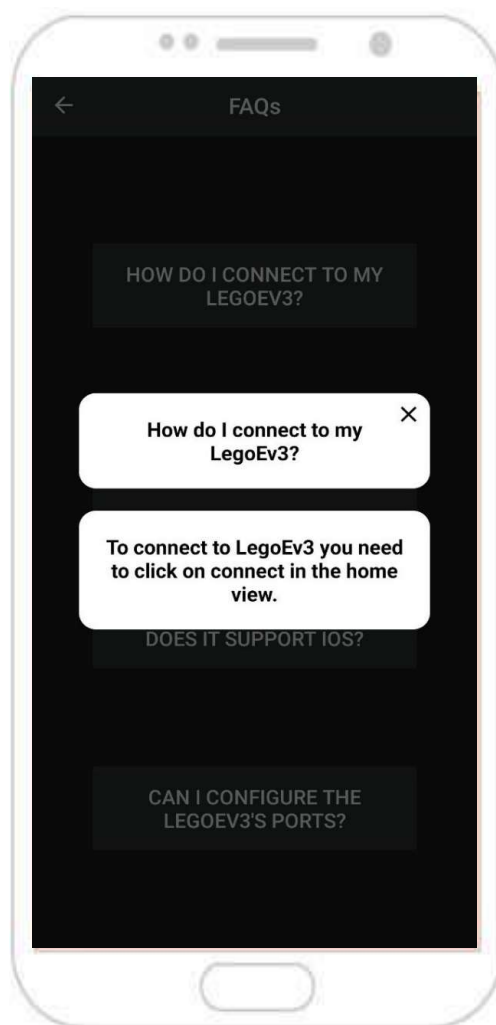


## 6. Interfaccia Grafica

### 6.7 FAQs

Questa schermata risponde ad eventuali domande da parte dell'utente riguardo l'applicazione. Qui verranno visualizzate le domande più frequenti. Cliccando su una domanda comparirà un pop-up con la domanda e la relativa risposta.

Il tasto indietro permetterà di ritornare alla sezione Help.



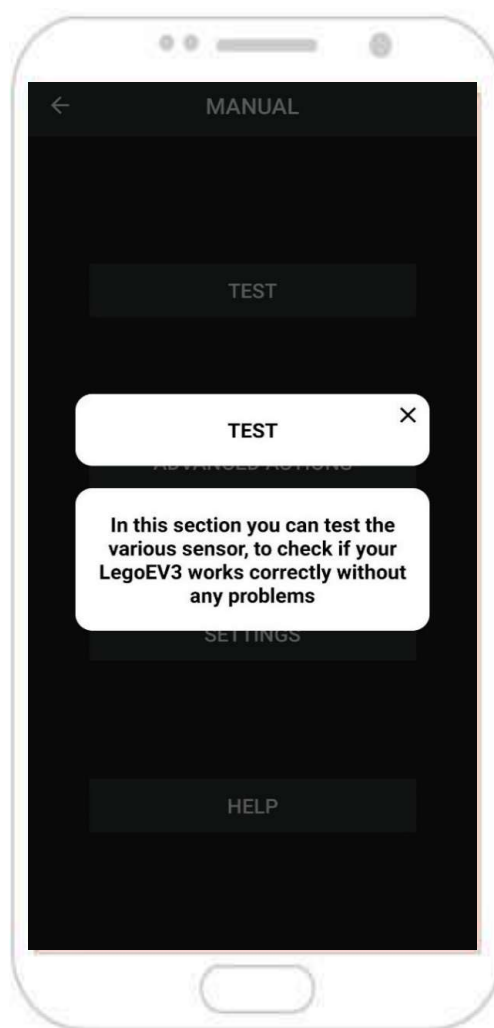
## 6. Interfaccia Grafica

### 6.8 Manual

Questa pagina mostra delle informazioni sulle modalità di guida presenti nell'app, sulle funzioni avanzate e sui comandi per il lego EV3.

Cliccando su un pulsante, comparirà un pop-up con la relativa informazione.

Il tasto indietro permetterà di ritornare alla sezione Help.

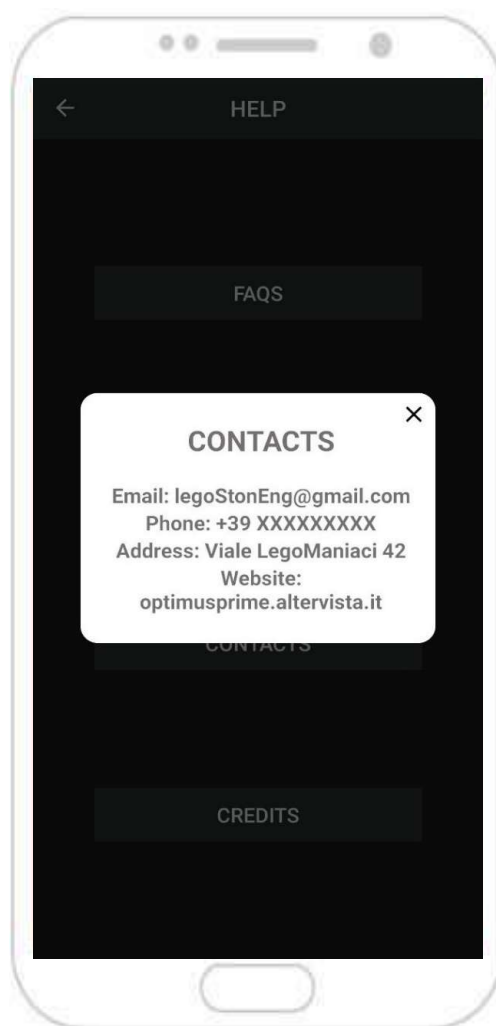




## 6. Interfaccia Grafica

### 6.9 Contacts

Cliccando sul relativo pulsante, compariranno tramite un pop-up, le relative descrittive e i contatti, per eventuali segnalazioni di ogni membro del gruppo LegoStonEng. E' presente il tasto indietro per tornare nella Home page.

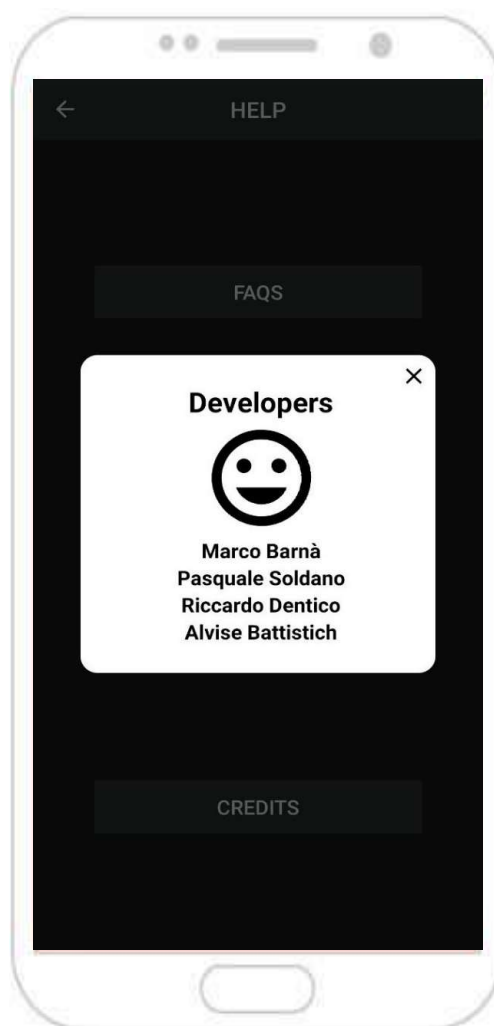


## 6. Interfaccia Grafica

### 6.10 Credits

Cliccando sul relativo pulsante, compariranno tramite un pop-up, le relative informazioni sui componenti del gruppo LegoStoneng che si sono resi partecipi di questo progetto.

E' presente il tasto indietro per tornare nella Home page.



## 6. Interfaccia Grafica

### 6.11 Settings

Questa pagina mostra le seguenti voci: “Set Ports”, “Bluetooth Connection”, “Review App” e Select Language. Selezionato il primo pulsante, apparirà un popup dove l’utente specificherà i collegamenti effettuati sul robot EV3. Qui potranno essere impostate le corrispondenze fra porta e sensore ad esso collegata. La seconda voce permette di attivare il bluetooth sul dispositivo su cui è installata l’applicazione, La terza voce di cambiare la lingua tra italiano / English. Infine “Review app” aprirà tramite browser o applicazione appostita il link alla pagina dell’applicazione nel market nella quale è stata pubblicata. E’ presente il tasto indietro per tornare nella home page.

