



KIT ROBOTICO EDUCATIVO PER L'APPRENDIMENTO DEL CODING NELLA SCUOLA DELL'INFANZIA

PENSIERO COMPUTAZIONALE E CODING



Seymour Papert

Pensiero Computazionale: i bambini attraverso l'attività di programmazione acquisiscono padronanza su una parte delle tecnologie moderne e costruiscono modelli di tipo intellettuale.

Coding: insieme di istruzioni atto a risolvere un problema.



Due bambini programmano Logo Turtle con la Button Box, 1974



Studenti alle prese con la programmazione di Lego TC Logo alla fine degli anni '80

INIZIATIVE DI CODING A SCUOLA



CodeWeek



Programma il futuro

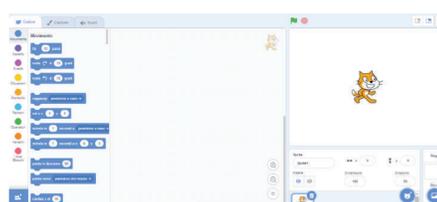


Code.org



codeMOOC

ESEMPI DI LINGUAGGI DI PROGRAMMAZIONE PER I BAMBINI



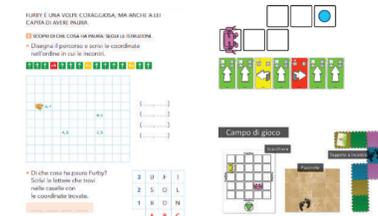
Scratch (8+ anni)



Scratch Jr (5-7 anni)



Coding Unplugged (attività che utilizzano strumenti non digitali)



PROGRAMMAZIONE TANGIBILE

La programmazione tangibile è un sistema che permette di imparare tramite la sperimentazione e l'esperienza diretta con l'ambiente e il mondo circostante. Questo sistema è una buona via di mezzo tra il Coding classico su dispositivi come pc e tablet e il Coding Unplugged.

TARGET

Bambini 3-6 anni

CODING: Codificazione e Decodificazione

CODIFICA



Carte codice da posizionare una dopo l'altra



Blocchi da inserire in una board



Blocchi da unire tra loro



Pulsanti

DECODIFICA



Utilizzo di sensori



Comunicazione USB



Sistema di identificazione ottica



Pulsanti

STATO DELL'ARTE



Kibo (2015)



Mochi (2019)



Code a Pillar (2016)



Bee Bot (2009)



Coding & Robotics (2018)



Cubetto (2017)



mTiny (2019)



Sphero Indi (2021)



MatataBot (2020)



Qobo (2019)



Coko (2018)



MojoBot (2019)

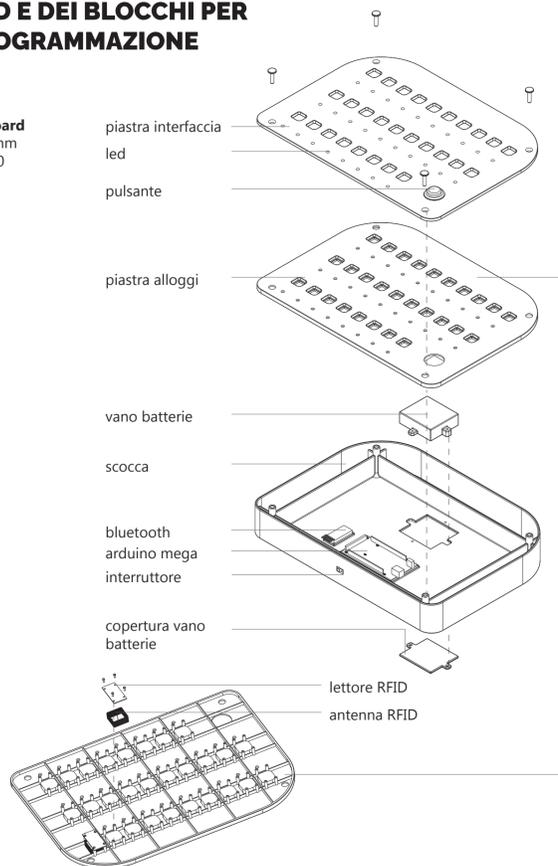
TABELLA COMPARATIVA DI VALUTAZIONE CRITICA DELLE MODALITÀ DI CODIFICAZIONE E DECODIFICAZIONE

Codifica	Decodifica	Esempio	Facilità di utilizzo singolo	Facilità di utilizzo in gruppo	Facilità di modifica della programmazione in caso di errore	Facilità di produzione	Costi di produzione	Precisione nella decodifica	Comandi supportati	Spazio utilizzato per la codifica
[Icon]	[Icon]	[Image]	●●●●●	●●●●○	●●●●●	●●●●○	●●●●○	●●●●●	●●●●○	●●●●○
[Icon]	[Icon]	[Image]	●●●●○	●●●●●	●●●●○	●●●●●	●●●●○	●●●●○	●●●●●	●●●●○
[Icon]	[Icon]	[Image]	●●●●○	●●●●●	●●●●○	●●●●○	●●●●○	●●●●○	●●●●●	●●●●○
[Icon]	[Icon]	[Image]	●●●●●	●●●●○	●●●●●	●●●●○	●●●●○	●●●●●	●●●●○	●●●●○
[Icon]	[Icon]	[Image]	●●●●○	●●●●○	●●●●○	●●●●○	●●●●○	●●●●○	●●●●○	●●●●○
[Icon]	[Icon]	[Image]	●●●●○	●●●●○	●●●●○	●●●●○	●●●●○	●●●●○	●●●●○	●●●●○
[Icon]	[Icon]	[Image]	●●●●○	●●●●○	●●●●○	●●●●○	●●●●○	●●●●○	●●●●○	●●●●○

IL PROGETTO DELLA BOARD E DEI BLOCCHI PER LA PROGRAMMAZIONE

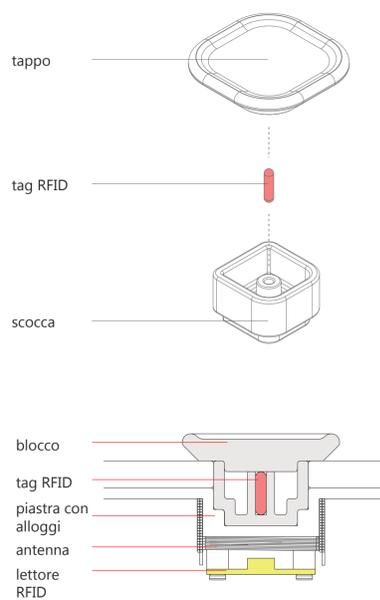
BOARD

Esploso board
Misure in mm
366x266x50
Scala 1:5

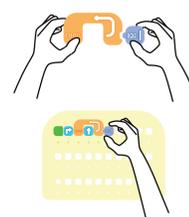


BLOCCO PROGRAMMAZIONE

Esploso blocco attuatore
Misure in mm
Scala 2:1



RFID (identificazione a radiofrequenza): sensori di riconoscimento univoco ed automatico, composti principalmente da: un tag, un antenna e un lettore. I tag permettono la realizzazione di comandi illimitati.



ATTUATORI



SENSORI



ACCENSIONE LED



ALLOGGIO BLOCCO

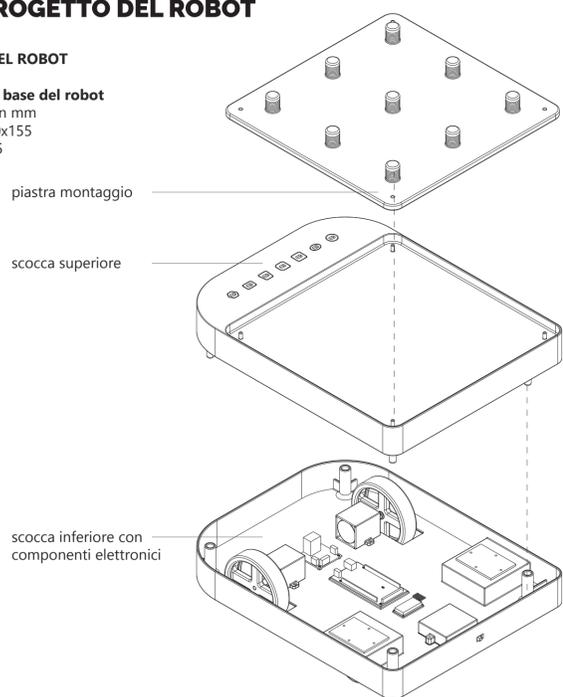
COMANDI DI CONTROLLO



IL PROGETTO DEL ROBOT

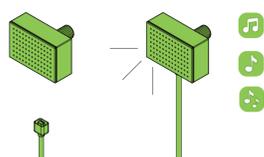
BASE DEL ROBOT

Esploso base del robot
Misure in mm
445x350x155
Scala 1:5

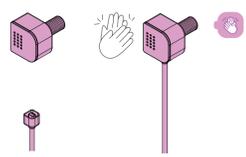


FUNZIONI COMPONENTI ELETTRONICI

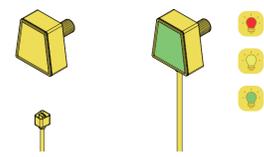
ATTUATORE RIPRODUTTORE AUDIO
riproduce fino a tre melodie differenti in base alla programmazione



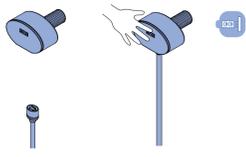
SENSORE SUONO
rileva un rumore come il battito delle mani



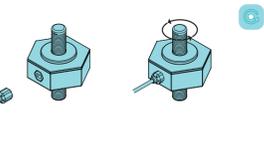
ATTUATORE LED
la luce può accendersi fino a tre colori differenti in base alla programmazione



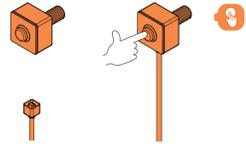
SENSORE PROSSIMITÀ
rileva la presenza di un ostacolo



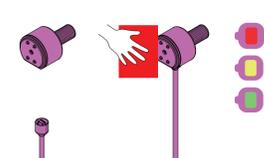
ATTUATORE SERVOMOTORE
permette una rotazione di 360° del componente a cui viene connesso



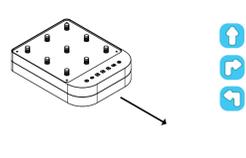
PULSANTE
rileva la pressione su di esso



SENSORE COLORE
rileva fino a tre colori differenti in base alla programmazione

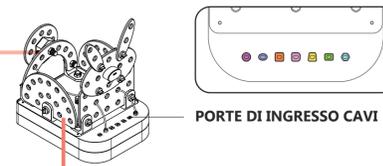


BASE ROBOT
permette il movimento del robot in tre direzioni differenti in base alla programmazione



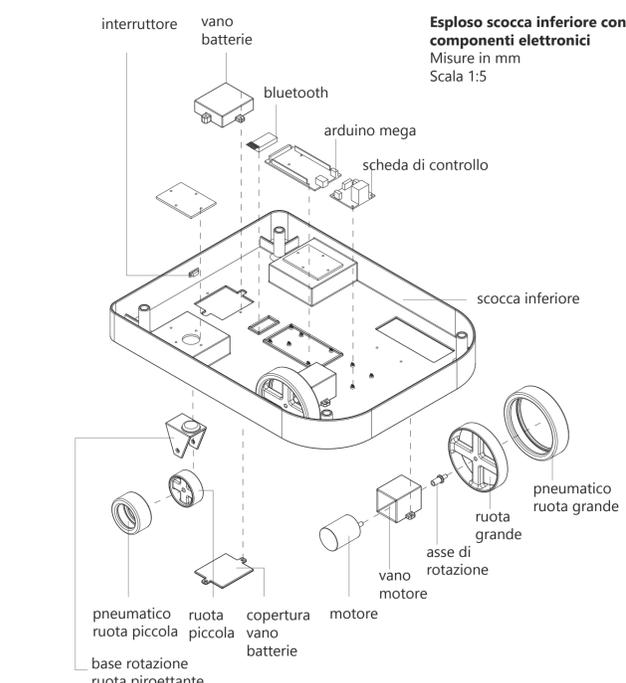
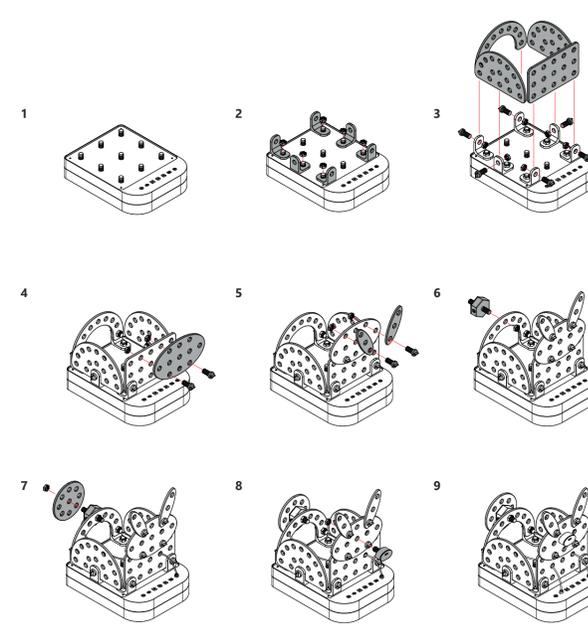
COMPONENTI ELETTRONICI

base robot	x1	sensori colore	x1	cavo attuatore audio	x1	cavo sensore suono	x1
attuatore riproduttore audio	x1	sensori suono	x1	cavo attuatore led	x1	cavo sensore prossimità	x1
attuatore Led	x1	sensori prossimità	x1	cavo attuatore servomotore	x1	cavo sensore pulsante	x1
attuatore servomotore	x1	sensori pulsante	x1	cavo sensore colore	x1	Lunghezza cavi: 1 metro	



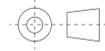
COMPONENTI COSTRUZIONE

cerchio grande	x4	ovale grande	x4	arco	x4
cerchio piccolo	x4	ovale piccolo	x4	barra	x4
rettangolo grande	x4	semicerchio grande	x4	connettore singolo	x12
rettangolo piccolo	x4	semicerchio piccolo	x4	connettore doppio	x6
quadrato grande	x4	triangolo grande	x4	vite	x50
quadrato piccolo	x4	triangolo piccolo	x4	bullone	x50

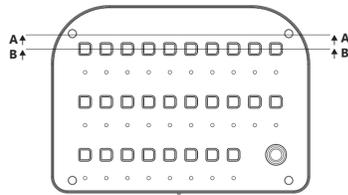
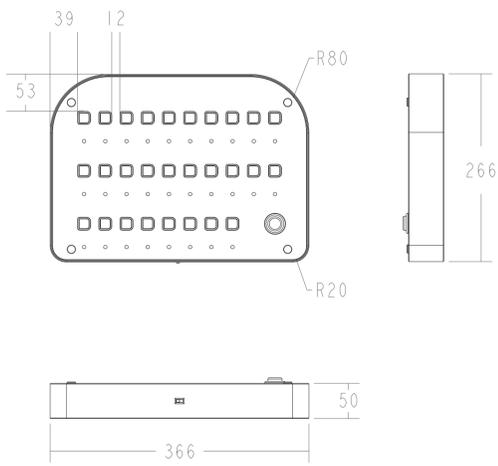


PROGETTO ESECUTIVO

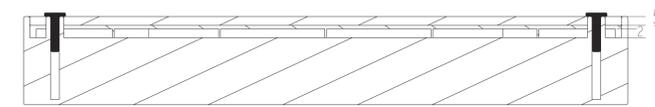
PROIEZIONI ORTOGONALI



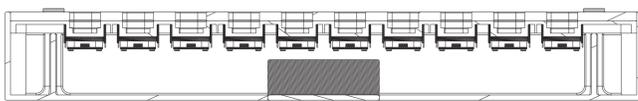
BOARD
Misure in mm
Scala 1:5



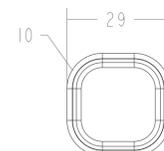
A-A'
Misure in mm
Scala 1:2



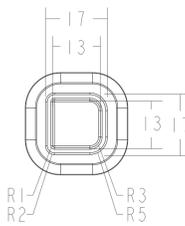
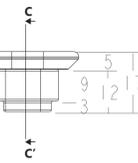
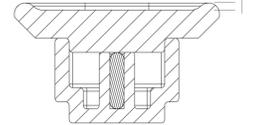
B-B'
Misure in mm
Scala 1:2



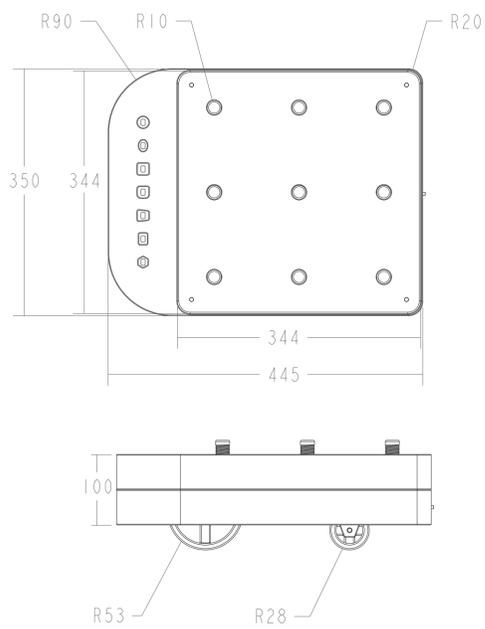
BLOCCHI:
Blocco attuatore
Misure in mm
Scala 1:1



C-C'
Misure in mm
scala 2:1

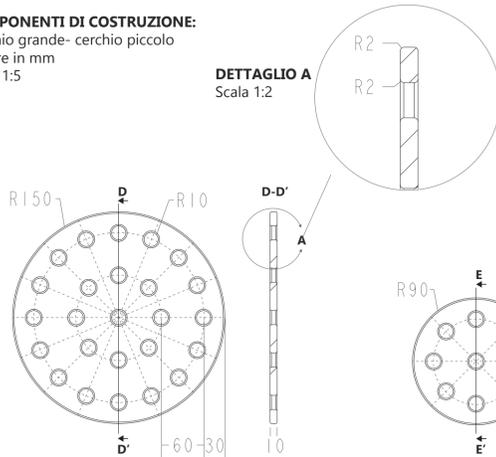


BASE DEL ROBOT
Misure in mm
Scala 1:5

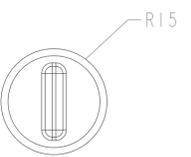
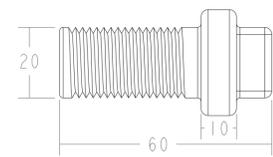


COMPONENTI DI COSTRUZIONE:
Cerchio grande- cerchio piccolo
Misure in mm
Scala 1:5

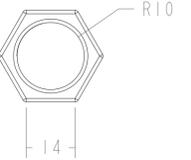
DETTAGLIO A
Scala 1:2



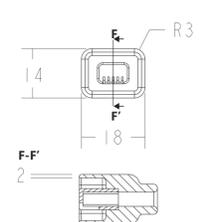
VITE
Misure in mm
Scala 1:1



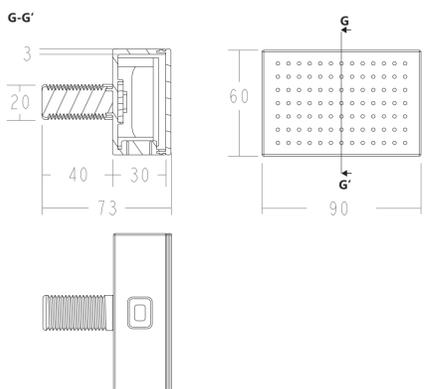
BULLONE
Misure in mm
Scala 1:1



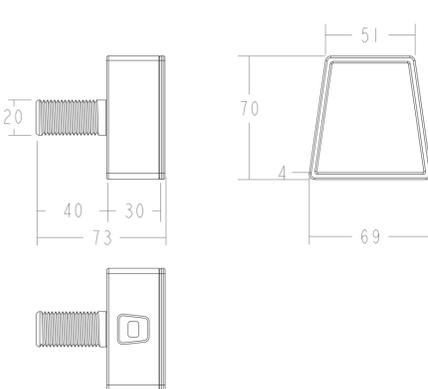
CONNETTORI ELETTRICI: RJ11
Connettore per riproduttore audio
Misure in mm
Scala 1:1



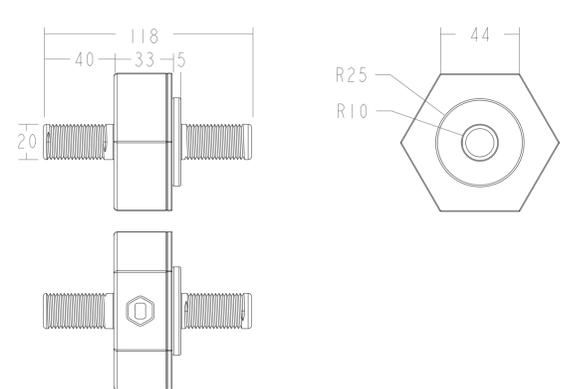
ATTUATORE RIPRODUTTORE AUDIO
Misure in mm
Scala 1:2



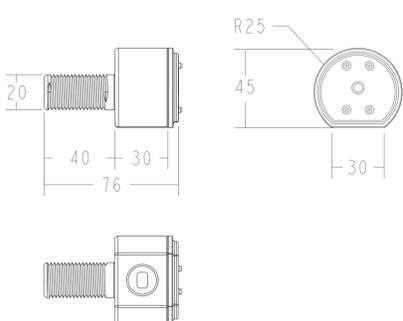
ATTUATORE LED
Misure in mm
Scala 1:2



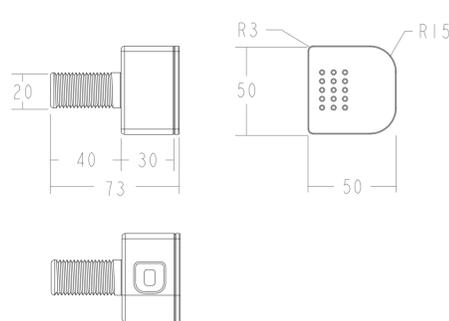
ATTUATORE SERVOMOTORE
Misure in mm
Scala 1:2



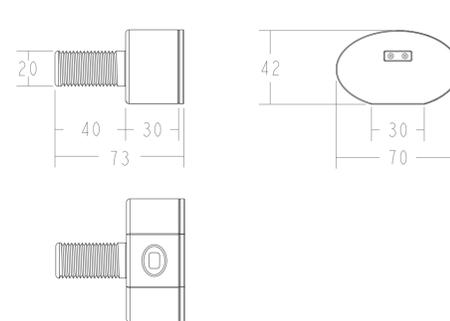
SENSORE COLORE
Misure in mm
Scala 1:2



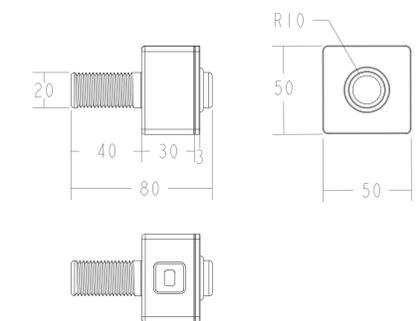
SENSORE SUONO
Misure in mm
Scala 1:2



SENSORE PROSSIMITÀ
Misure in mm
Scala 1:2



SENSORE PULSANTE
Misure in mm
Scala 1:2





COMBINAZIONE 1: CONIGLIO



COMBINAZIONE 3: CANE



COMBINAZIONE 2: LUCCIOLA

COMBINAZIONE 4: GIRAFFA



COMBINAZIONE 5: ORSO



BOARD E BLOCCHI PER LA PROGRAMMAZIONE

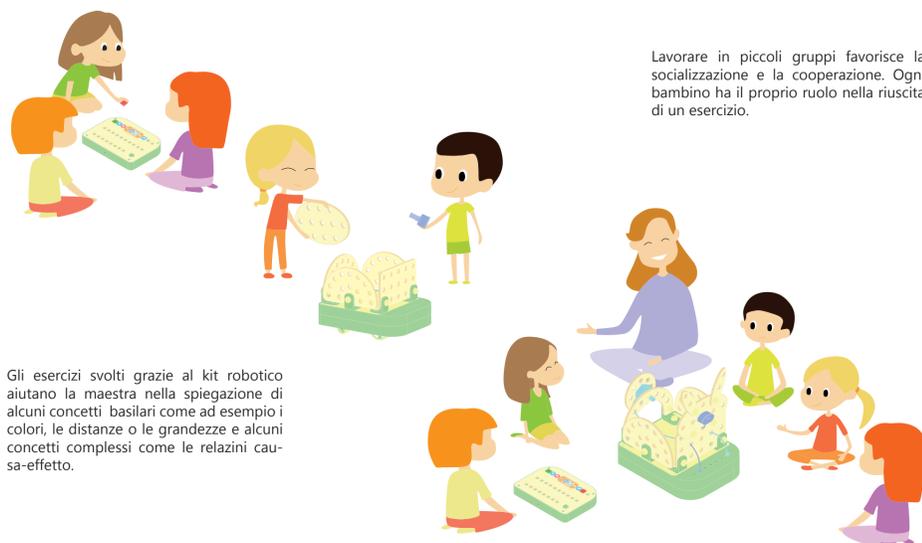
MATERIALI PRINCIPALI



ABS
(acrilonitrile butadiene stirene)



Compensato di betulla

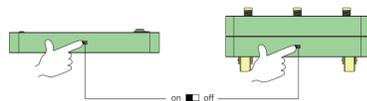


Lavorare in piccoli gruppi favorisce la socializzazione e la cooperazione. Ogni bambino ha il proprio ruolo nella riuscita di un esercizio.

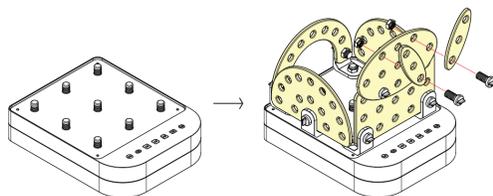
Gli esercizi svolti grazie al kit robotico aiutano la maestra nella spiegazione di alcuni concetti basilari come ad esempio i colori, le distanze o le grandezze e alcuni concetti complessi come le relazioni causa-effetto.

INTERAZIONE IN BREVE

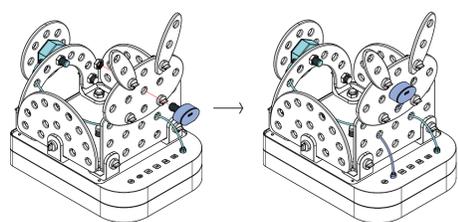
1. ACCENSIONE BOARD E ROBOT



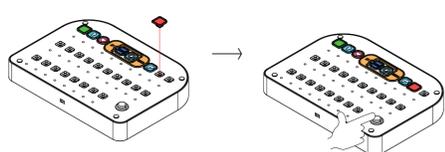
2. MONTAGGIO CON COMPONENTI DI COSTRUZIONE



3. MONTAGGIO SENSORI E ATTUATORI



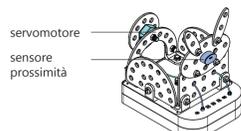
4. PROGRAMMAZIONE



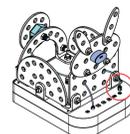
POSSIBILI ERRORI

ESEMPIO:
I bambini programmano il robot affinché si muova a destra, muova un componente, si muova in avanti ed infine al riconoscere di un ostacolo muova nuovamente un componente.

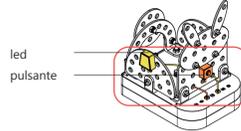
Programmazione e componenti elettronici CORRETTI



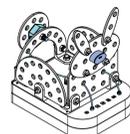
Inserimento del cavo di un componente elettronico ERRATO



Componenti elettronici ERRATI



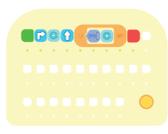
Inserimento dei blocchi di programmazione ERRATO



Programmazione ERRATA



Esempio di codice:
1. Inizio
2. Svolta a destra
3. Rotazione di un componente di 360°
4. Avanti
5. Condizione: se il robot rileverà un oggetto vicino allora un componente verrà ruotato di 360°
6. Fine

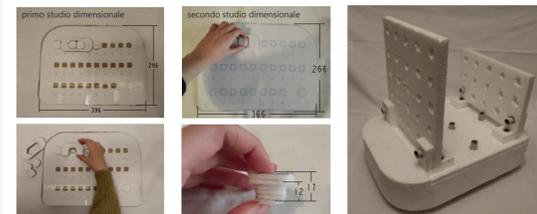


Esempio di codice:
1. Inizio
2. Svolta a destra
3. Rotazione di un componente di 360°
4. Avanti
5. Condizione: in questo caso la condizione non potrà essere risolta per mancanza di funzioni
6. Fine



Esempio di codice:
1. Inizio
2. Svolta a destra
3. Accensione luce gialla
5. Condizione: se verrà premuto il pulsante si accenderà la luce gialla
6. Fine

PROTOTIPI DIMENSIONALI



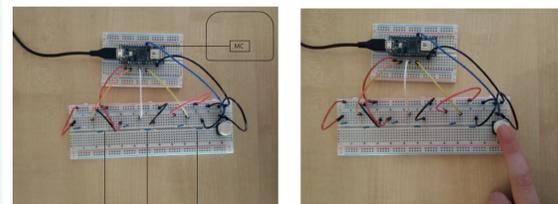
Studi dimensionali board e blocchi

Studio dimensionale robot

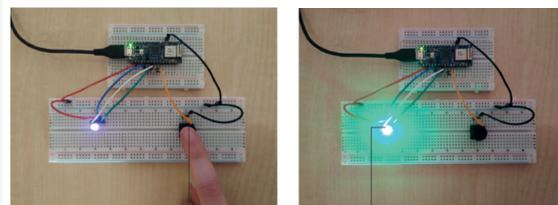
PROTOTIPO FUNZIONALE

Funzione: Condizione (se il pulsante verrà premuto si accenderà una luce verde)

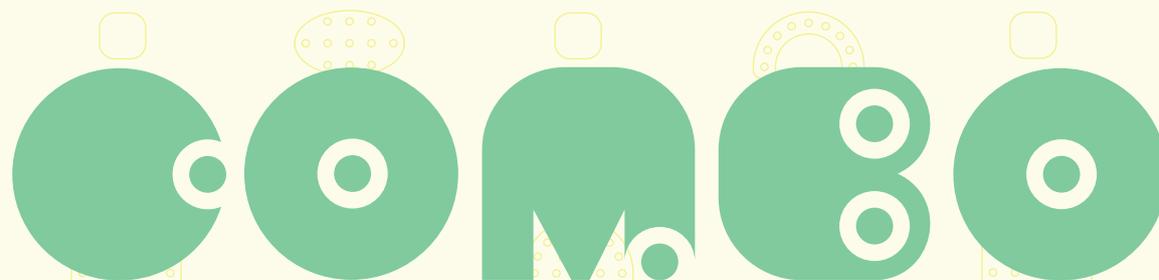
Codice:
1. Condizione: Resistenza 2.2K
2. Pulsante: Resistenza 3.3K
3. Led verde: Resistenza 10K



Si inseriscono i comandi affinché premendo il pulsante si accenda la luce verde. Una volta terminata la programmazione viene premuto il pulsante e i comandi vengono inviati alla scheda collegata all'attuatore e al sensore.



La condizione viene risolta se si preme il pulsante, in quel caso si accende la luce verde per 15 secondi. La condizione non viene risolta se non si preme il pulsante entro 30 secondi.



Kit robotico educativo per l'apprendimento del Coding nella scuola dell'infanzia

Scuola di Architettura e Design "E.Vittoria" dell'Università di Camerino (SAAD)
Magistrale in Design per l'Innovazione Digitale

Tesi di laurea Magistrale
A.A. 2021/2022

Laureanda:
Giulia D'Alessandro
Matricola 109457

Relatore:
Giuseppe Losco

Correlatore:
Davide Paciotti

COMBO

Kit robotico educativo per l'apprendimento del Coding nella scuola dell'infanzia

Scuola di Architettura e Design "E.Vittoria" dell'Università di Camerino (SAAD)
Magistrale in Design per l'Innovazione Digitale

Tesi di laurea Magistrale
A.A. 2021/2022

Laureanda:
Giulia D'Alessandro
Matricola 109457

Relatore:
Giuseppe Losco

Correlatore:
Davide Paciotti



Scuola di Ateneo
Architettura e Design "Eduardo Vittoria"
Università di Camerino

INDICE

ABSTRACT

CODING E PENSIERO COMPUTAZIONALE

1.1 Cos'è il Coding	8
1.2 Jean Piaget ed il Costruttivismo	8
1.3 Costruzionismo e pensiero computazionale	10
1.4 L'importanza del Coding a scuola nell'era dei nativi digitali	11
1.5 Iniziative di Coding a scuola	13

IL CODING NELLA SCUOLA DELL'INFANZIA

2.1 Scuola dell'infanzia	16
2.2 Campi di esperienza	16
2.3 Metodologie, strategie e mediatori didattici	17
2.4 Coding nella scuola dell'infanzia	19
2.5 Mitchel Resnick e le quattro P dell'apprendimento creativo	20

IL GIOCO COME ATTIVITÀ EDUCATIVA: dal giocattolo educativo all'utilizzo del coding e della robotica

3.1 Cos'è il giocattolo educativo	22
3.2 I primi giocattoli educativi	22
3.3 Il secolo del bambino	24
3.4 Sviluppo dell'elettronica	24
3.5 Coding e Robotica educativa	26
3.6 Programmazione tangibile	27
3.7 XXI secolo, Coding tra robotica educativa e realtà virtuale	30

LINGUAGGI DI PROGRAMMAZIONE PER BAMBINI

4.1 Cos'è un linguaggio di programmazione	35
4.2 Scratch	36
4.3 ScratchJr	42
4.4 Coding Unplugged	48

STATO DELL'ARTE

5.1 Kit robotici educativi in commercio programmati in modo tangibile	51
5.2 Kibo	52
5.3 Mochi	57
5.4 Code a Pillar	61
5.5 Bee Bot	64

5.6 Coding & Robotics	66
5.7 Cubetto	72
5.8 mTiny	75
5.9 Sphero Indi	79
5.10 MatataBot	82
5.11 Qobo	87
5.12 Coko	92
5.13 MojoBot	95
5.14 Tabelle	100

PROGETTO

6.1 Combo	110
6.2 Board e blocchi per la programmazione	110
6.3 Robot	129
6.4 Interazione	162
6.5 Render	166
6.6 Prototipo funzionale	168
6.7 Conclusioni	175

SITOGRAFIA E BIBLIOGRAFIA

Abstract

Il pensiero computazionale è un processo mentale sempre più necessario nel contesto in cui viviamo oggi dove la tecnologia sta diventando parte del nostro quotidiano. È dunque importante inserire il Coding in tutti i contesti scolastici sin dalla prima infanzia per aiutare i bambini ad avere un maggiore controllo sui cambiamenti attuali e futuri, facendo sì che la tecnologia venga utilizzata come uno strumento senza esserne dominati. Ad oggi il Coding viene spesso introdotto come insegnamento trasversale e sebbene per bambini e ragazzi della scuola elementare, media e superiore sia più semplice comprendere le dinamiche del pensiero computazionale, per i bambini della scuola dell'infanzia (3-6 anni) la questione diventa complessa. È per questo che l'obiettivo del Coding non deve essere quello di sviluppare processi di pensiero computazionale insegnando semplicemente un linguaggio di programmazione ma deve essere quello di dare la possibilità di esprimersi attraverso una nuova modalità. Durante la scuola dell'infanzia i bambini sono in una fase di inizio dell'apprendimento di sistemi simbolici come i numeri e l'alfabeto. È in questo contesto che l'insegnamento della codifica può introdursi come un ulteriore sistema simbolico da poter utilizzare per esprimere concetti ed idee.

Il Progetto sviluppato nel lavoro di tesi si basa sulla progettazione di un Kit Robotico educativo programmabile dai più piccoli che permetta di sviluppare oltre al pensiero computazionale anche la creatività, la socializzazione, il problem solving e lo storytelling. Per raggiungere l'obiettivo sono stati analizzati i linguaggi di programmazione e i prodotti pensati appositamente per aiutare i bambini dai 3 ai 6 anni nell'apprendimento del Coding. Il risultato che ne è emerso è quello di un Kit composto da due elementi principali: un robot personalizzabile nel suo aspetto con sensori e attuatori applicabili e una board adibita alla programmazione tangibile con blocchi per i comandi da inserire al suo interno. Con questo Kit i bambini possono dividersi i compiti lavorando in gruppo e collaborare

quando si presenta un errore nella programmazione o nell'assemblaggio mettendo in comune le diverse conoscenze ed abilità. Infine può essere uno strumento utile all'insegnante per introdurre concetti nuovi o approfondire quelli appresi in precedenza.

CODING E PENSIERO COMPUTAZIONALE

1.1 Cos'è il Coding

Il Coding è un termine inglese e significa "programmazione", considerata come la fase implementativa dell'intero ciclo di sviluppo del software e si serve di differenti linguaggi per poter scrivere un programma.

In informatica un programma è *"un insieme coordinato e strutturato di istruzioni, codificato secondo un opportuno linguaggio di programmazione, atto a risolvere un dato problema mediante un calcolatore"* (Treccani).

Il processo di Coding è strettamente legato al "Pensiero Computazionale" ovvero un processo mentale che mira prevalentemente alla risoluzione di problemi. L'attività di Coding è diventata più diffusa rispetto al passato, sia perché sono aumentati gli utenti che utilizzano i dispositivi elettronici e che hanno accesso ad Internet, sia perché sono aumentate le tipologie di dispositivi elettronici disponibili. Ad oggi sempre più spesso si sente parlare di Coding e di materie che fanno del pensiero computazionale una disciplina da inserire nel percorso scolastico. Nei prossimi paragrafi cercherò di comprendere da cosa nasce il Pensiero computazionale e come con il Coding entra nella realtà scolastica.

1.2 Jean Piaget ed il Costruttivismo

Quando si parla di Pensiero Computazionale inteso come un processo mentale non si può non far riferimento ai primi studi che fecero da precursori come quelli di Jean Piaget Figura 1. Piaget è stato uno psicologo, biologo, pedagogista e filosofo svizzero che spiegò nei suoi studi lo sviluppo mentale dei bambini. Fondò, verso la metà del XX secolo, l'epistemologia genetica, uno studio sperimentale delle strutture e dei processi cognitivi legati alla costruzione della conoscenza nel corso dello sviluppo. Con questa prospettiva psicologica intendeva spiegarci come la validità della conoscenza fosse collegata al modello della sua costruzione, per esempio la nostra esperienza

diretta della forza di gravità ha più validità della nostra esperienza indiretta con i buchi neri. Secondo Piaget, un essere umano sviluppa le sue abilità cognitive nel corso della sua vita, attraversando quattro stadi sequenziali di sviluppo, intesi come **fasi di sviluppo cognitivo**. Con questi studi dimostrò innanzitutto una differenza qualitativa tra le modalità di pensiero del bambino e quello dell'adulto. Le quattro fasi sono (1):

- **Fase senso-motoria** (dalla nascita ai 2 anni circa): questo periodo è definito senso-motorio perché i bambini utilizzano i sensi e le abilità motorie per esplorare la realtà che li circonda. Si sviluppa la "permanenza dell'oggetto", ovvero i bambini imparano che gli oggetti continuano ad esistere anche se non possono essere visti. Inoltre, iniziano a comprendere che le loro azioni possono causare delle reazioni nel mondo che li circonda, sperimentando un rapporto causa-effetto. In questo contesto il bambino opera per prove ed errori, organizzando le sue mosse in modelli comportamentali sempre più complessi.
- **Fase pre-operativa** (dai 2 ai 6 anni): i bambini iniziano a pensare in modo simbolico e imparano a usare parole e immagini per rappresentare oggetti, un esempio è il gioco creativo nel quale i bambini usano, per esempio, una scatola per rappresentare un tavolo. In questa fase il pensiero inizia a evolvere verso aspetti più astratti, ma è ancora guidato da un forte egocentrismo che limita i bambini in termini di flessibilità cognitiva e attenzione divisa. In altre parole, possono soffermarsi solo su un singolo aspetto per volta e il punto di vista personale sulla realtà è l'unica chiave di lettura possibile e credibile.
- **Fase operatoria-concreta** (dai 6 agli 11 anni): i bambini comprendono che i loro pensieri non vengono necessariamente condivisi da altri. Iniziano a pensare in modo logico ad eventi concreti, la capacità

di risolvere i problemi dipende dall'acquisizione di tre aspetti basilari del ragionamento come: l'identità (principio per cui un oggetto o una persona rimangono se stessi nel corso del tempo), compensazione (principio per cui i cambiamenti in una determinata dimensione possono essere controbilanciati da altri cambiamenti in un'altra dimensione) e reversibilità (l'abilità di pensare in una serie di passaggi e poi invertirla mentalmente per tornare al punto di partenza).

- **Fase delle operazioni formali** (dai 12 anni in poi): gli adolescenti iniziano a pensare di più a questioni morali, filosofiche, etiche, sociali e politiche che richiedono ragionamenti teorici e astratti. In questa fase finale avviene un aumento della logica, delle capacità di utilizzare il ragionamento deduttivo e comprensione delle idee astratte. A questo punto, le persone diventano capaci di vedere molteplici potenziali soluzioni ai problemi e pensare in modo più scientifico riguardo il mondo che li circonda.

L'intero processo di acquisizione di conoscenze subisce inoltre, nel corso della vita, innumerevoli e costanti processi di regolazione. I contenuti vengono organizzati in schemi concettuali che devono essere flessibili per potersi riorganizzare in vista di una richiesta ambientale. L'intelligenza è quindi strettamente legata anche alla capacità di sapersi adeguare all'ambiente sociale e fisico che ci circonda. È con queste premesse che Piaget (1971) ha teorizzato il ruolo di due meccanismi importati che caratterizzano l'adattamento: **l'assimilazione e l'accomodamento**.

- **Assimilazione:** consiste nella capacità di selezionare e incorporare nuove esperienze e informazioni agli schemi già in possesso.
- **Accomodamento:** consiste nel meccanismo opposto, ossia la modificazione dei comportamenti e degli schemi cognitivi preesistenti in relazione al contesto circostante.

La ricerca di un equilibrio tra assimilazione e accomodamento è alla base della produzione di ogni nuova conoscenza e dello sviluppo dell'intelligenza. Piaget con i suoi studi venne considerato come il precursore del **Costruttivismo**.



Figura 1- Jean Piaget, 1969

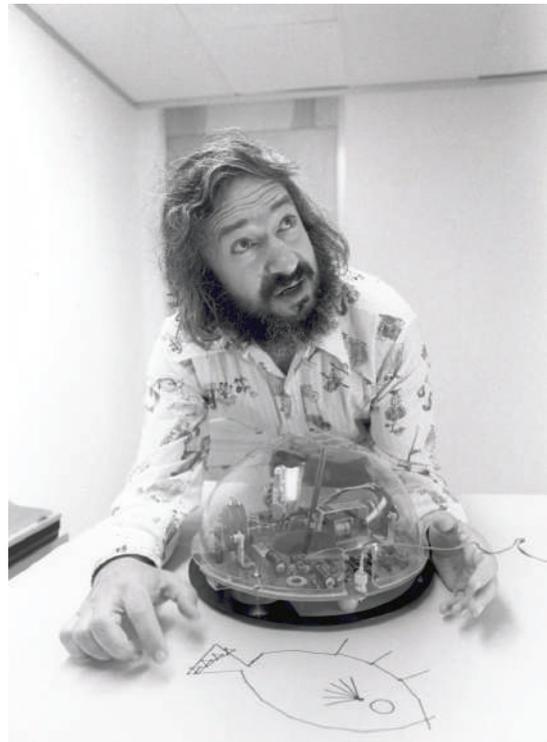


Figura 2- Seymour Papert con il primo prototipo di Logo Turtle

Secondo le posizioni costruttiviste, i sensi veicolano l'informazione proveniente dalla realtà, ma questa è troppo caotica per il nostro cervello. Pertanto, il nostro cervello deve strutturarla e, per farlo, categorizza le informazioni caotiche in concetti e interpretazioni.

1.3 Costruzionismo e pensiero computazionale

Il pensiero costruttivista portò a quello che venne definito successivamente Costruzionismo da **Seymour Papert** Figura 2, un filosofo, matematico, informatico e pedagogista. Tra il 1959 e il 1963, Papert, ha lavorato con Piaget nel "Centre International d'Epistémologie Génétique" dell'Università di Ginevra e dal 1964 è stato ricercatore al Massachusetts Institute of Technology (MIT) di Boston, dove ha fondato insieme a Marvin Minsky il Laboratorio di Intelligenza Artificiale. Lavorando con Piaget ha potuto approfondire lo studio del costruttivismo e della psicologia educativa, questo lo ha portato ad elaborare la propria teoria in cui fonde il Costruttivismo di Piaget con le nuove tecnologie e con gli strumenti informatici, dando vita al "**Costruzionismo**": il bambino impara facendo attraverso gli artefatti. Secondo questa teoria, i bambini e gli adulti apprendono più profondamente, se hanno la possibilità di esplorare, provare e manipolare. Si passa cioè dall'essere "consumatori" di informazioni a "produttori" di conoscenza. Nel modello di Papert l'esperienza diventa il tema centrale perché il bambino comprende qualcosa costruendolo "internamente" a partire dalla propria esperienza e da azioni preliminari. Nel 1980, Papert inserì le sue teorie nel libro "**Mindstorms: Children, Computers, and Powerful Ideas**". Nel libro non trattò solamente il Costruzionismo ma parlò per la prima volta della creazione di un linguaggio di programmazione adatto ai bambini denominato "**Logo**", creato dal suo laboratorio del MIT. Papert sosteneva che nel 1980 l'educazione era in un momento storico in cui era possibile un cambiamento radicale e la possibilità di quel cambiamento era direttamente legata all'impatto del computer e della programmazione. Il computer diventa in questo modo un mezzo di apprendimento, non più soltanto un mezzo tecnologico ma uno strumento per abbattere il muro tra cultura umanistica e scientifica.

Scrivo: *"In molte scuole oggi, l'espressione "istruzione assistita dal computer" significa fare in modo che il computer insegni al bambino. Si potrebbe dire che il computer viene utilizzato per programmare il bambino. Nella mia visione, il bambino programma il computer e, così facendo, acquisisce un senso di padronanza su una parte della tecnologia moderna, stabilendo un contatto con alcune delle idee più profonde della scienza, della matematica e dell'arte di costruire modelli*

intellettuali." ("Mindstorms: Children, Computers, and Powerful Ideas", 1980, pag.5)⁽²⁾

In questo libro è presente anche il tema dell' "**errore**" che prende parte al processo di "costruzione" della conoscenza e dell'apprendimento. Sbagliare dà la possibilità al bambino di riflettere su ciò che ha creato il problema, studiarne ogni aspetto e individuare quelle che secondo lui possono essere le soluzioni alternative. Secondo Papert, infatti, la distinzione giusto-sbagliato tende a bloccare il processo dell'apprendimento, poiché il bambino davanti a un errore tende a rimuoverlo e a dimenticarlo. La correzione degli errori fa parte del processo di comprensione e questi permettono di capire ciò che è accaduto e cosa non è andato bene. In questo avvicinarsi gradualmente all'obiettivo procedendo per tentativi e correzione degli errori, l'esperienza della programmazione (coding) si rivela uno strumento adeguato ed utile. È parlando del Coding e del linguaggio di programmazione Logo, che Papert usa per la prima volta l'espressione "**Pensiero computazionale**", ad oggi non esiste una vera e propria definizione ma si può affermare che il pensiero computazionale è un processo mentale che consente di risolvere problemi di varia natura seguendo metodi e strumenti specifici, pianificando una strategia e includendo una gamma di strumenti "mentali" che riflettono l'ampiezza della scienza informatica. Questo termine verrà ripreso dalla dottoressa Jeannette Wing, direttrice del Dipartimento di Informatica della Carnegie Mellon University, in un articolo del 2006 dal titolo "**Computational Thinking**".

Per J. Wing *"Pensare come un informatico significa molto di più di saper programmare un computer. Richiede piuttosto il saper pensare a più livelli di astrazione"*. (Wing, 2006) ⁽³⁾.

Nel suo articolo cerca di definire i punti che definiscono il pensiero computazionale:

- Riformulare un problema apparentemente difficile in uno che sappiamo come risolvere, magari semplicemente riducendolo, incorporandolo, trasformandolo o simulandolo.
- Pensare utilizzando i meccanismi di astrazione e decomposizione quando si ha ad avere a che fare con un compito o un sistema complesso e di grandi dimensioni. È separazione dei concetti. È saper scegliere il

metodo di rappresentazione più appropriato e modellare gli aspetti più importanti di un problema, in modo da renderlo più facilmente trattabile.

- Rappresenta un atteggiamento e una competenza universalmente applicabili da chiunque, non solo dagli informatici.
- Va dalla risoluzione di problemi e la progettazione di sistemi, a capire certi comportamenti umani, affidandosi ai concetti fondamentali e gli strumenti mentali propri dell'informatica.

Da ciò si può quindi comprendere come il Pensiero Computazionale altro non è che un'attitudine e un processo mentale applicabili da chiunque e in qualunque contesto da cui ogni aspetto della nostra vita può trarne beneficio.

1.4 L'importanza del Coding a scuola nell'era dei nativi digitali

L'espressione **nativi digitali** fa riferimento alla generazione di chi è nato e cresciuto in corrispondenza con la diffusione delle nuove tecnologie informatiche. E quindi si tratta, in genere, di persone che non hanno avuto alcuna difficoltà a imparare l'uso di queste tecnologie. Questa espressione si è diffusa in Italia a partire dal 2001 quando venne pubblicato il libro di Marc Prensky, "Digital Natives, Digital Immigrants".

Le infinite possibilità di comunicazione e di accesso all'informazione hanno portato ad un cambiamento economico e politico, giocando anche un ruolo importante nella formazione dell'identità sociale e personale.

Secondo Prensky, l'immersione totale nella tecnologia ha trasformato gli stili cognitivi e di apprendimento delle nuove generazioni in modo permanente e sempre più incisivo. Questa forma nuova di essere e di pensare rimane tuttavia lontana da chi, nato prima degli anni '80, si è invece dovuto adattare alle innovazioni tecnologiche, i cosiddetti "**immigrati digitali**". Nonostante queste distinzioni siano state utili agli inizi degli anni 2000, addentrandoci nel XXI secolo la distinzione fra nativi e immigrati digitali sarà sempre meno importante. Lo stesso Prensky, in un articolo del 2009, "H. Sapiens Digital: From Digital Immigrants and Digital Natives to Digital Wisdom"⁽⁴⁾, ha rivisto la sua posizione rispetto

a questo tema, parlando di "**saggezza digitale**" e "**stupidità digitale**". La prima è intesa come la capacità di usare in modo critico e responsabile le tecnologie digitali. La seconda, rappresenta il comportamento di chi fa un cattivo uso della tecnologia per sfuggire a situazioni sgradevoli o per danneggiare qualcuno piuttosto che per rafforzare la propria saggezza. La saggezza e la stupidità digitale sono slegate dai fattori generazionali e non significano abilità nel manipolare o meno la tecnologia ma capacità di prendere o no decisioni più sagge perché potenziate dalla tecnologia. Avere "saggezza digitale", ad oggi, viene considerata un obiettivo educativo e non più una condizione connaturata di essere nati o no nell'era digitale. Quello in cui viviamo è un mondo che richiede una quantità sempre maggiore di competenze e conoscenze, a partire da quelle digitali, ma non solo. La tecnologia è presente in quasi ogni aspetto della nostra quotidianità, dal tempo libero, all'istruzione, al mondo del lavoro, essere in possesso degli strumenti cognitivi per padroneggiarle è cruciale.

È nel contesto appena descritto che si inserisce l'importanza di introdurre nuove metodologie didattiche come le materie **STEM**, un approccio che cerca di valorizzare il contributo offerto dalle materie scientifiche nel riuscire a leggere e comprendere il funzionamento del mondo in cui viviamo. STEM è l'unione di Scienza, Tecnologia, Ingegneria e Matematica (in lingua inglese) o in alternativa, **STEAM** che sta per Scienza, Tecnologia, Ingegneria, Arti e Matematica. L'aggiunta di Arte a STEM per creare STEAM incorpora il pensiero creativo e le arti applicate in situazioni reali. L'arte riguarda la scoperta e la creazione di modi ingegnosi di risoluzione dei problemi. La componente chiave di STEM e STEAM è l'integrazione. Invece di insegnare discipline come materie indipendenti, le lezioni sono a tutto tondo, basate su progetti e indagini, con un focus sull'apprendimento interdisciplinare. Nonostante queste discipline siano importanti poiché consentono di insegnare agli studenti il pensiero computazionale concentrandosi sulle applicazioni del mondo reale, in Italia, tradizionalmente, si preferisce l'ambito umanistico (generalmente sentito come parte della cultura generale diffusa) a discapito delle discipline scientifiche (spesso considerate come argomento riservato agli specialisti). Le conseguenze di questa scelta, da quanto emerge dai dati del report dell'Osservatorio #conibambini (a cura di Con i Bambini e Openpolis)⁽⁵⁾, sono :

- **Una bassa percentuale di laureati nelle discipline Stem** rispetto alla media europea. A livello Ue, sono circa 21 ogni 1.000 i giovani laureati in materie come scienze, matematica, informatica, ingegneria. In Italia sono 16,4 ogni 1.000 persone tra 20 e 29 anni.
- **Un allargamento delle disparità di genere.** A causa di stereotipi sociali, le studentesse restano una minoranza nei percorsi scientifici. A fronte di una media Ue di circa 21 laureati Stem ogni 1.000 giovani tra 20 e 29 anni, le laureate sono solo 14,9. Il dato dei maschi è quasi doppio: 27,9.
- Il terzo effetto, che poi è alla radice di quelli appena citati, è **la scarsa diffusione nella popolazione giovanile di competenze Stem, abilità digitali, conoscenze applicative.** La percentuale di giovani 16-19 anni con competenze digitali base o superiori nel 2019 è del 64% con una media Ue dell'83%.

Per far fronte a questi problemi è importante introdurre sin dalla prima infanzia un approccio alle materie STEM. L'Italia, come tutti gli altri stati dell'Unione europea, in questi anni è stata impegnata nella transizione digitale, per ridurre i tanti gap tecnologici interni al paese.

Il 22 febbraio 2018, per far fronte a questi cambiamenti, MIUR ha presentato il documento "Indicazioni nazionali e nuovi scenari" ⁽⁶⁾, frutto del lavoro del Comitato scientifico per le Indicazioni nazionali della scuola dell'Infanzia e del primo ciclo di istruzione. Il documento propone alle scuole una rilettura delle Indicazioni nazionali emanate nel 2012 ed entrate in vigore dall'anno scolastico 2013/2014. In questo documento tra gli strumenti culturali per la cittadinanza appare il pensiero computazionale descritto come *"processo mentale che consente di risolvere problemi di varia natura seguendo metodi e strumenti specifici pianificando una strategia"*, è sempre più importante dunque allontanare la convinzione che il pensiero computazionale sia una competenza solo di chi ha fatto dell'informatica la propria professione. Nello stesso documento si parla anche di coding in questi termini:

"Nei contesti attuali, in cui la tecnologia dell'informazione è così pervasiva, la padronanza del coding e del pensiero computazionale possono aiutare le persone a governare

le macchine e a comprenderne meglio il funzionamento, senza esserne invece dominati e asserviti in modo acritico".

Quando si fa coding non vengono messi in atto solamente nuovi processi mentali ma anche nuovi modi di vedere il mondo e noi stessi. Karen Brennan e Mitchel Resnick, attraverso una ricerca intitolata **"Nuovi quadri per lo studio e la valutazione dello sviluppo del pensiero computazionale"** (2012)⁽⁷⁾, aiutano a comprendere come le attività di apprendimento basate sulla programmazione supportino lo sviluppo del pensiero computazionale. Da questa ricerca sono emerse "pratiche" e "prospettive" riscontrate nei bambini e ragazzi alle prese con la programmazione.

Pratiche:

- **Essere incrementali e iterativi:** con la programmazione è quasi impossibile o comunque molto difficile realizzare un progetto perfetto al primo tentativo, questa richiede controlli successivi che ne migliorino le funzionalità e ne aggiungano nuove.
- **Testing e debugging:** la pratica di testare e correggere, procedere per tentativi ed errori è fondamentale nel coding, in quanto per sapere se un programma sia corretto o meno è necessario provare, individuare i bug e cercare di risolvere i problemi.
- **Riuso e Remixing:** utilizzare lavori già scritti di altre persone, anziché creare un programma da zero, non è solo più conveniente in termini di tempo e fatica; costruire qualcosa di nuovo sulla base di qualcosa che è già stato creato consente lo sviluppo di capacità critiche di lettura del codice, per capire cosa è opportuno utilizzare, modificare o perfezionare dei progetti altrui e per arrivare a creare cose molto più complesse di quelle che si potrebbero fare da soli.
- **Astrazione e decomposizione:** pratiche importanti nella progettazione e nel problem solving, attività che consentono di costruire qualcosa concettualizzando il problema, per tradurlo in codice nel modo più essenziale possibile, tralasciando dettagli inutili e scomponendo e lavorando su singole parti più semplici da riunire in seguito per ottenere la soluzione completa.

Queste pratiche portano inevitabilmente al miglioramento di se stessi e delle relazioni con gli altri e il mondo. Per questo motivo nella ricerca sono state aggiunte anche "le prospettive" che rappresentano i cambiamenti osservati nei bambini e ragazzi.

Prospettive:

- **Esprimere se stessi:** fare uso dei media interattivi semplicemente come consumatori, permette di imparare ad usare le tecnologie, ma non promuove lo sviluppo di tutte quelle abilità legate al pensiero computazionale; questo modo di pensare consente di creare, di esprimere se stessi e le proprie idee attraverso un nuovo mezzo, con il quale ognuno può creare qualcosa di unico e personale.
- **Connettersi:** progettare insieme, cosa molto comune e proficua in questo tipo di attività, aiuta a comprendere come collaborare con gli altri e i benefici e le opportunità che questo offre. Collaborare consente di creare con gli altri, ma anche di creare per gli altri, quindi oltre alla possibilità di creare di più di quanto si potrebbe fare da soli, offre anche la possibilità di sperimentare il valore di avere un pubblico autentico, da intrattenere, coinvolgere, arricchire e educare.
- **Farsi domande:** per programmare un computer in modo da risolvere un problema che viene dalla realtà è necessario porsi domande sul mondo che ci circonda e sul suo funzionamento, interrogarsi su ciò che diamo per scontato e, attraverso la programmazione e la progettazione, cercare di trovare delle risposte.

Sulla base di queste parole è possibile comprendere come il Coding sia importante nel processo di istruzione di bambini e ragazzi, supportando un cambiamento educativo che sta avvenendo e continuerà ad avvenire.

1.5 Iniziative di Coding a scuola

Esistono ad oggi diverse iniziative che promuovono attività di Coding all'interno di contesti educativi. Qui di seguito alcuni esempi:

- **CodeWeek:** ⁽⁸⁾ (Figura 3) la settimana Europea della programmazione, è un'iniziativa gestita da volontari e sostenuta dalla Commissione europea. Ha lo scopo di portare la programmazione e l'alfabetizzazione digitale a tutti in modo divertente offrendo e condividendo esperienze. Chiunque (scuole, insegnanti, biblioteche, club di programmazione, imprese, autorità pubbliche) può organizzare un'attività #CodeWeek e aggiungerla alla mappa sul sito ufficiale. Per organizzare un'attività si ha bisogno di: personale che conosca la programmazione (insegnanti o formatori), un qualunque posto dove imparare, computer con connessione ad internet o materiali per il coding unplugged e una modalità di registrazione dei partecipanti. Il fine ultimo di CodeWeek è la condivisione della propria attività sulla mappa in modo tale da raggiungere più persone possibili che abbiano voglia di imparare. Questa iniziativa offre anche moduli di formazione per chi, insegnanti o studenti, è alle prime armi con la programmazione.
- **Code.org:** ⁽⁹⁾ (Figura 4) è un'organizzazione senza scopo di lucro e un sito web omonimo guidato dai fratelli Hadi e Ali Partovi, che mira ad incoraggiare tutte le persone ad avere un'istruzione informatica di qualità. Code.org è stato lanciato nel 2013 con un video che promuoveva l'informatica. Questo video divenne immediatamente virale e 15.000 scuole si unirono a loro per aiutarli, diventando in poco tempo una grande organizzazione a sostegno di un movimento mondiale. Ad oggi offre attività interattive strutturate in corsi graduali, adatte per ogni fascia di età, con obiettivi definiti e fornendo supporto per gli insegnanti. Tutti i piani di studio e le esercitazioni sono gratuite e tradotte per essere utilizzate a livello mondiale. Nel 2013, l'organizzazione, ha inoltre promosso l'**Ora del codice**, iniziata come un'ora di lezione per introdurre l'informatica. Da allora è diventata uno sforzo globale per diffondere l'informatica, iniziando con attività di programmazione della durata di un'ora, ed espandendosi grazie ad un incredibile

impegno collettivo. Ad oggi questa campagna viene supportata da più di 400 partner e 200.000 educatori in tutto il mondo svolgendosi ogni anno durante la Settimana di Educazione all'Informatica. L'Ora del codice propone lezioni di un'ora che funzionano su pc, tablet, smartphone, oltre a lezioni che non necessitano di alcun computer.

- **Programma il futuro:** ⁽¹⁰⁾ (Figura 5) è un progetto avviato nell'anno scolastico 2014-2015 dal Ministero dell'Istruzione in collaborazione con il CINI (Consorzio Interuniversitario Nazionale per l'Informatica). Ha preso ispirazione dal progetto Code.org ed ha l'obiettivo di fornire alle scuole una serie di strumenti semplici, divertenti e facilmente accessibili per formare gli studenti ai concetti di base dell'informatica e all'uso responsabile della tecnologia informatica. Il progetto è stato riconosciuto come iniziativa di eccellenza europea per l'istruzione digitale nell'ambito degli European Digital Skills Awards 2016. All'interno del sito sono disponibili lezioni tecnologiche ed attività unplugged, rivolte a studenti di ogni fascia di età, che non richiedono particolari competenze tecniche da parte degli insegnanti e degli stessi alunni. La modalità di partecipazione di base al progetto è definita dall'"Ora del Codice" (vedere Code.org), scegliendo tra lezioni tecnologiche (per le scuole dotate di postazioni informatiche e connessione ad internet) e attività tradizionali (che non necessitano di dotazioni tecnologiche), sono inoltre disponibili percorsi avanzati che gli alunni possono svolgere dopo aver partecipato all'"Ora del codice". Al termine delle attività svolte, gli alunni possono ricevere gli attestati di completamento personalizzati, messi a disposizione delle scuole da Programma il Futuro in formato pdf editabile.
- **CodeMOOC:** ⁽¹¹⁾ (Figura 6) è un MOOC (massive open online course) erogato gratuitamente dall'Università di Urbino sulla piattaforma europea MOOC Uniurb. E' un corso di formazione con l'obiettivo di aiutare gli insegnanti di qualsiasi scuola e di qualsiasi disciplina ad introdurre il pensiero computazionale in classe attraverso il coding, usando solo attività intuitive e divertenti da proporre direttamente agli alunni. Il corso consiste in video-lezioni in diretta e si serve di diversi canali quali la piattaforma, il canale YouTube, il blog e il canale Telegram.



Figura 3- interfaccia sito ufficiale codeweek.eu



Figura 4- interfaccia sito ufficiale Code.org

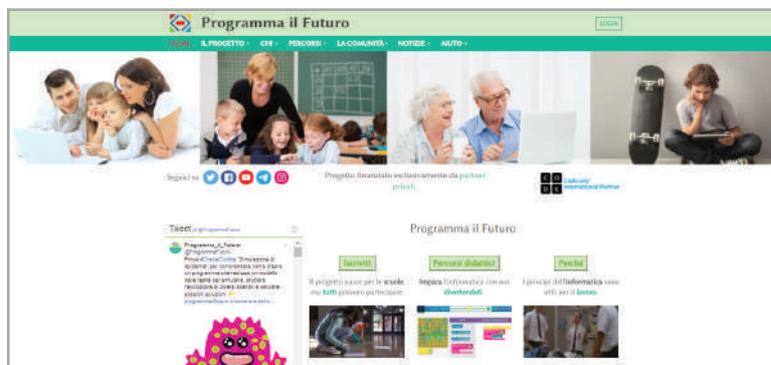


Figura 5- interfaccia sito ufficiale programmailfuturo.it



Figura 6- interfaccia sito ufficiale codemoo.org

IL CODING NELLA SCUOLA DELL'INFANZIA

2.1 Scuola dell'infanzia

La scuola dell'infanzia fa parte del Sistema integrato di educazione e di istruzione dalla nascita ai sei anni ed è il primo gradino del percorso di istruzione che precede la scuola primaria. La denominazione "scuola dell'infanzia" è stata introdotta dalla Riforma Moratti del 2003 in sostituzione della dicitura "scuola materna", inserendola, in questo modo, a pieno titolo, nel sistema educativo. La durata della scuola dell'infanzia è di 3 anni (dai 3 ai 6 anni di età) e non riveste carattere obbligatorio. La scuola dell'infanzia concorre all'educazione e allo sviluppo affettivo, psicomotorio, cognitivo, morale, religioso e sociale dei bambini promuovendone le potenzialità di relazione, autonomia, creatività, apprendimento e mira ad assicurare un'effettiva uguaglianza delle opportunità educative. Rappresenta da sempre un passaggio molto importante per il bambino da un punto di vista non solo cognitivo ma anche sociale ed emotivo. La fascia di età 3-6 anni è un importante momento di sviluppo per la definizione del carattere, i bambini si trovano ad affrontare un passaggio significativo verso l'autonomia e la socializzazione. Prima dell'ingresso nella scuola dell'infanzia, la socializzazione dei bambini avviene attraverso la relazione con gli adulti in un contesto familiare. Con l'ingresso a scuola, il bambino è posto di fronte a un ambiente sociale complesso, a un contesto formato da nuove relazioni extra familiari, nel quale è chiamato ad interagire con più persone contemporaneamente, pari e adulti e a mettere in atto interazioni sociali via via più articolate. La scuola dell'infanzia quindi si impegna a promuovere nei bambini lo sviluppo dell'identità, dell'autonomia, della competenza e li avvia alla cittadinanza⁽¹²⁾. Gli obiettivi principali sono:

- **Consolidare l'identità** significa vivere serenamente tutte le dimensioni del proprio io, stare bene e sentirsi sicuri in un ambiente sociale allargato, imparare a conoscersi e ad essere riconosciuti come persona unica e irripetibile.
- **Sviluppare l'autonomia** significa avere fiducia in sé e fidarsi degli altri; provare soddisfazione nel fare da sé e saper chiedere aiuto; esprimere sentimenti ed emozioni; partecipare alle decisioni esprimendo opinioni, imparando ad operare scelte e ad assumere comportamenti e atteggiamenti sempre più consapevoli.
- **Acquisire competenze** significa giocare, muoversi, manipolare, curiosare, domandare, imparare a riflettere sull'esperienza attraverso l'esplorazione, l'osservazione e il confronto tra proprietà, quantità, caratteristiche, fatti; significa ascoltare, e comprendere, narrazioni e discorsi, raccontare e rievocare azioni ed esperienze e tradurle in tracce personali e condivise; essere in grado di descrivere, rappresentare e immaginare, «ripetere», con simulazioni e giochi di ruolo, situazioni ed eventi con linguaggi diversi.
- **Vivere le prime esperienze di cittadinanza** significa scoprire l'altro da sé e attribuire progressiva importanza agli altri e ai loro bisogni.

Tali finalità sono perseguite attraverso l'organizzazione di un ambiente di vita, di relazioni e di apprendimento di qualità.

2.2 Campi di esperienza

"I bambini esplorano continuamente la realtà ed imparano a riflettere sulle proprie esperienze descrivendole, rappresentandole, riorganizzandole con criteri diversi. Pongono così le basi per la successiva elaborazione di concetti scientifici e matematici che verranno proposti nella scuola primaria", in questo modo il Miur scrive dei campi di esperienza nel documento sulle "Indicazioni nazionali per il curricolo della scuola dell'infanzia e del primo ciclo d'istruzione"⁽¹³⁾ (2012). Queste parole possono rappresentare una descrizione sintetica ma efficace del concetto che sta alla base di tutti i campi di esperienza.

Gli insegnanti accolgono, valorizzano ed estendono le curiosità, le esplorazioni, le proposte dei bambini e creano occasioni di apprendimento per favorire l'organizzazione di ciò che i bambini vanno scoprendo. L'esperienza diretta, il gioco, il procedere per tentativi ed errori, permettono al bambino, opportunamente guidato, di approfondire e sistematizzare gli apprendimenti. Ogni campo di esperienza offre un insieme di oggetti, situazioni, immagini e linguaggi, riferiti ai sistemi simbolici della nostra cultura, capaci di evocare, stimolare, accompagnare apprendimenti progressivamente più sicuri.

I campi di esperienza sono cinque:

- **Il sè e l'altro:** i bambini iniziano a formulare tanti "perchè", questo rappresenta la loro spinta a capire il significato della vita che li circonda e il valore morale delle loro azioni. Nel contesto scolastico hanno molte occasioni per prendere coscienza della propria identità percependo le proprie esigenze e i propri sentimenti, imparando ad esprimerli in modo sempre più adeguato.
- **Il corpo e il movimento:** i bambini iniziano a prendere coscienza del corpo come mezzo per scoprire se stessi ed il mondo. Il movimento viene considerato come il primo fattore di apprendimento: cercare, scoprire, giocare, saltare, correre a scuola è fonte di benessere e di equilibrio psico-fisico. La scuola dell'infanzia mira a sviluppare la capacità di esprimersi e di comunicare attraverso il corpo per giungere ad affinarne le capacità percettive e di conoscenza degli oggetti, la capacità di orientarsi nello spazio, di muoversi e di comunicare secondo immaginazione e creatività.
- **Immagini, suoni e colori:** i bambini imparano ad esprimere pensieri ed emozioni con immaginazione e creatività attraverso la drammatizzazione, il disegno, la pittura e le altre attività manipolative; utilizzano materiali e strumenti, tecniche espressive e creative. Seguono con curiosità e piacere spettacoli di vario tipo (teatrali, musicali, visivi e di animazione); sviluppano interesse per l'ascolto della musica e per la fruizione di opere d'arte.
- **I discorsi e le parole:** i bambini si presentano alla scuola dell'infanzia con un patrimonio linguistico significativo, ma con compe-

tenze differenziate, che vanno attentamente osservate e valorizzate. La lingua è uno strumento essenziale per comunicare e conoscere, per rendere via via più complesso e meglio definito, il proprio pensiero. I bambini imparano ad ascoltare storie e racconti, dialogano con adulti e compagni, giocano con la lingua che usano e si cimentano con l'esplorazione della lingua scritta. La scuola dell'infanzia ha la responsabilità di promuovere in tutti i bambini la padronanza della lingua italiana, rispettando l'uso della lingua di origine.

- **La conoscenza del mondo:** I bambini esplorano la realtà e imparano a riflettere sulle proprie esperienze. Iniziano ad esplorare oggetti, materiali e simboli, osservando la vita di piante ed animali, elaborando idee personali da confrontare con quelle dei compagni e degli insegnanti. Imparano a fare domande, a dare e a chiedere spiegazioni, a lasciarsi convincere dai punti di vista degli altri, a non scoraggiarsi se le loro idee non risultano appropriate. In questo campo di esperienza i bambini imparano a raggruppare e ordinare oggetti e materiali secondo criteri diversi, ne identificano alcune proprietà, confrontano e valutano la quantità (utilizzando simboli per registrare i risultati), iniziano a contare ed operare con i numeri.

2.3 Metodologie, strategie e mediatori didattici

Per facilitare l'apprendimento ed il raggiungimento degli obiettivi prefissati nei cinque campi di esperienza, gli insegnanti si servono di metodologie, strategie e strumenti specifici come i mediatori didattici.

Quando si parla di **metodologie didattiche** si fa riferimento a tutte le procedure per affrontare le unità didattiche. Prima di scegliere quale o quali adottare l'insegnante individua le abilità di base dei propri alunni e cerca di capire quali sono gli obiettivi da raggiungere. Esempi di metodologie didattiche possono essere: la didattica laboratoriale, cooperative learning, problem solving e peer education. Le **strategie didattiche** a differenza delle metodologie fanno riferimento ad una linea comportamentale che l'insegnante decide di stabilire per favorire il conseguimento degli obiettivi di apprendimento prefissati.

Le strategie didattiche sono essenzialmente di due tipi: **strategia espositiva e strategia euristica.** ⁽¹⁴⁾

- **Strategia espositiva:** viene considerato come un approccio trasmissivo all'istruzione (per esempio la classica lezione frontale) in cui l'insegnante preferisce soffermarsi sugli aspetti contenutistici dell'insegnamento. La trasmissione può avvenire anche in forma coinvolgente, non necessariamente di trasmissione passiva.
- **Strategia euristica:** in questo caso l'insegnante adopera le procedure tipiche di metodi operativi o di metodi euristici-partecipativi che svilupperà attraverso tecniche cooperative quali: il brainstorming, il team word webbing, il circle team, il group investigation, il jigsaw, il learning together, la complex instruction, lo structural approach e molte altre. La strategia euristica può essere considerata di tipo attivo poiché richiede un'attivazione senso-motoria e mentale dell'alunno, che potrebbe trovarsi a lavorare in modalità laboratoriale con i suoi compagni.

Con le differenti strategie didattiche è possibile utilizzare anche differenti mediatori. Quando si parla di mediatori didattici si fa riferimento ad ogni "dispositivo", reale o simbolico, catalizzatore del processo educativo, che facilita la relazione tra insegnamento ed apprendimento, potenziando la comunicazione e rendendola più efficiente. Elio Damiano nel suo libro "La mediazione didattica"⁽¹⁵⁾ (2013) individua quattro tipologie di mediatori didattici: **attivi, iconici, analogici e simbolici.**

- **Attivi:** ^{Figura 7} ricorrono all'esperienza diretta ed è il tipo di mediatore più vicino alla realtà (es. se l'insegnante vuole far capire ai bambini il significato della parola "bottiglia", farà loro vedere e toccare una bottiglia reale).



Figura 7 - Un mediatore didattico attivo potrebbe essere un esperimento scientifico.

- **Iconici:** ^{Figura 8} ovvero che utilizzano il linguaggio delle icone (grafico e spaziale) fatto di immagini, schemi, mappe concettuali (es. in questo caso l'insegnante mostrerà una bottiglia raffigurata su un cartello).



Figura 8 - Un mediatore didattico iconico potrebbe essere una carta geografica.

- **Analogici:** ^{Figura 9} potrebbero essere anche chiamati "ludici", poiché si basano sulle dinamiche del gioco e della simulazione (es. in questo caso l'insegnante dovrà fingere di bere).



Figura 9- Un mediatore didattico analogico potrebbe essere l'imitazione di un mestiere.

- **Simbolici:** ^{Figura 10} utilizzano codici di rappresentazione convenzionali e universali come lettere e numeri (es. in questo caso l'insegnante dovrà scrivere alla lavagna la parola bottiglia e spiegare a voce la definizione del termine).



Figura 10- Un mediatore didattico simbolico potrebbe essere la scrittura alla lavagna di operazioni matematiche.

2.4 Coding nella scuola dell'infanzia

Il Coding nella scuola dell'infanzia spesso viene proposto come attività trasversale per lo sviluppo del pensiero computazionale nei cinque campi di esperienza. Entro il 2022 il Coding sarà obbligatorio in tutte le scuole dell'infanzia e primaria (Mozione n. 1-00117 del 12 marzo 2019) per questo motivo garantire agli studenti una didattica digitale non sarà più un'opzione ma un dovere. Tanti sono i motivi per il quale è utile inserire il coding nel curriculum scolastico (vedere capitolo precedente paragrafo 1.4) ma è importante comprendere come questo possa avvenire in una fascia di età compresa dai 3 ai 6 anni. L'obiettivo dell'insegnamento del Coding non è solo quello di sviluppare processi di pensiero computazionale insegnando un linguaggio di programmazione ma è quello di dare la possibilità di esprimersi attraverso una nuova modalità. Nella scuola dell'infanzia la codifica presuppone la comprensione dell'esistenza di un linguaggio che rappresenta delle azioni e queste azioni possono essere combinate in modi differenti per creare progetti personali e sempre nuovi. Nel libro "Coding as a Playground" ⁽¹⁶⁾ (2020) Marina Umaschi Bers, mette in luce **sei fasi di apprendimento alla codifica** basate sull'osservazione del comportamento di bambini dai 4 ai 7 anni. Le fasi descrivono la progressione dell'apprendimento del Coding nei bambini piccoli. A volte questo processo è evolutivo e le progressioni sono viste come sequenziali ed ordinate. Altre volte lo sviluppo è molto più individualistico, è direzionale ma non del tutto lineare. Qui di seguito le sei fasi di apprendimento:

- **Fase 1 Emergente:**

- Il bambino sa come accendere e spegnere uno strumento.
- Sa che la tecnologia non è viva, ed è programmata da una persona.
- Sa che un comando (rappresentato da simboli) rappresenta un comportamento ed inizia ad esplorare solo alcuni comandi.

- **Fase 2 Codifica e Decodifica:**

- Il bambino sa creare programmi semplici con comandi di causa ed effetto.
- Si impegna nell'esplorazione di comandi orientati verso un obiettivo.
- Esegue il debug di problemi semplici tramite tentativi ed errori.

- **Fase 3 Scioltezza:**

- Il bambino sa creare programmi complessi utilizzando strutture di controllo come i cicli di ripetizione e le istruzioni "If / Then".
- Sa individuare e riparare errori nel codice
- Sta iniziando ad usare strategia nel correggere gli errori nel codice.

- **Fase 4 Nuova conoscenza:**

- Il bambino si impegna in un'esplorazione logica più orientata all'obiettivo.
- Crea programmi con cicli annidati (interazioni all'interno di altre interazioni) ed ottiene sequenze complesse.
- Usa strategia nel modo in cui esegue il Debug.

- **Fase 5 Prospettive multiple:**

- Il bambino sa creare programmi che coinvolgono l'input dell'utente.
- Sa creare più programmi che interagiscono tra loro.
- Sta cominciando ad analizzare, sintetizzare e tradurre concetti in codice.
- Sa eseguire il debug di più strutture di controllo.

- **Fase 6 Intenzionalità:**

- Il bambino è motivato a creare programmi complessi.
- È in grado di analizzare, sintetizzare e tradurre concetti astratti in codice.
- Sa tradurre concetti astratti in codice e di nuovo in concetti.
- Sa identificare più modi per tradurre concetti astratti in codice.

Queste fasi portano ad un pieno apprendimento del Coding e possono essere paragonate alle fasi di sviluppo dell'alfabetizzazione (proposte da Jeanne Chall nel 1983)⁽¹⁷⁾. Durante la prima infanzia i bambini iniziano a conoscere diversi sistemi simbolici come i numeri o l'alfabeto. Per questo motivo è importante introdurre la Codifica sin dalla scuola dell'infanzia come sistema simbolico da poter utilizzare per esprimere concetti e idee personali. Nelle sei fasi la progressione dell'apprendimento avviene in modo differente in ogni bambino e potrebbe avvenire che si acquisiscano abbastanza conoscenze da essere fluenti nella programmazione ma non essere stimolati al raggiungimento di obiettivi mirati. Per questo motivo oltre a comprendere come insegnare la codifica è bene comprendere anche come stimolare i bambini all'apprendimento creativo.

2.5 Mitchel Resnick e le quattro P dell'apprendimento creativo

Mitchel Resnick è docente in Learning Research al Media Lab del Massachusetts Institute of Technology (MIT) di Boston, dirige il gruppo di ricerca Lifelong Kindergarten ed è titolare della cattedra "LEGO". Nel suo libro **"Lifelong Kindergarten. Cultivating Creativity through Projects, Passion, Peers, and Play"** (18) (2017) parla per la prima volta della **"spirale dell'apprendimento creativo"** (Figura 11). Nel libro, Resnick porta come esempio un gruppo di bambini che giocano con delle costruzioni: qualcuno inizia a costruire e contemporaneamente qualcuno immagina una storia che si abbina alla costruzione. Quando però la costruzione crolla i bambini provano a renderla più stabile modificandola, e quindi riadattano la storia, o ne inventano una nuova. Questo processo di immaginazione e gioco viene reiterato, più e più volte, intervallato da fasi di riflessione su quanto svolto ("Perché la costruzione è crollata?" oppure "Come possiamo renderla migliore?") e di condivisione di idee fra i gruppi. Resnick immagina questo processo come "la spirale dell'apprendimento creativo" che si ripete all'infinito.

- **Immaginare:** nell'esempio i bambini iniziano ad immaginare il castello.
- **Creare:** i bambini traducono le loro idee in azioni costruendo un castello.
- **Giocare:** i bambini sperimentano giocando con le loro creazioni, cercando di costruire una torre più alta o aggiungendo dettagli.
- **Condividere:** i bambini collaborano alla costruzione del castello.
- **Riflettere:** quando il castello crolla, l'insegnante si avvicina ai bambini e li incoraggia a riflettere sul perché sia crollato e come potrebbero renderlo più stabile.

Resnick e il suo team di ricerca si sono interrogati su come sia possibile incoraggiare e sostenere un'esperienza di apprendimento creativo. Nascono così i quattro principi guida (le quattro P dell'apprendimento): **Projects, Passion, Peers e Play**. In breve, sostengono che il miglior modo per coltivare la creatività sia incentivare le persone nel lavorare a progetti basati sulle loro passioni, in collaborazione con i pari e in uno spirito giocoso.

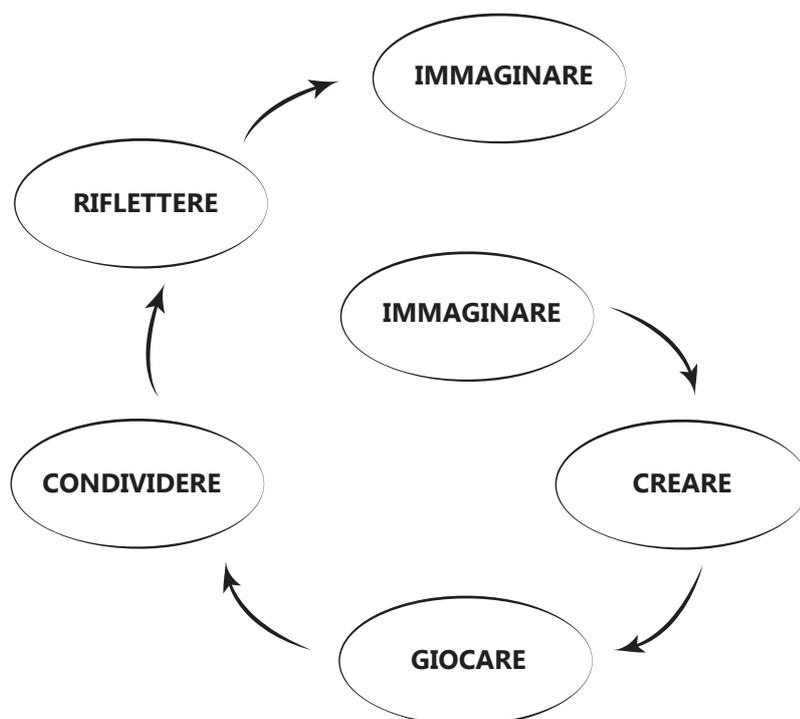


Figura 11- Spirale dell'apprendimento creativo

- **Projects (Progetti):** far lavorare i bambini in modo attivo su progetti, generando nuove idee e progettando prototipi.
- **Passion (Passione):** consentire ai bambini di basarsi sui loro interessi e di lavorare su progetti personalmente significativi.
- **Peers (Lavoro tra pari):** Supportare i bambini nella collaborazione e nella condivisione. L'apprendimento cresce se è un'attività sociale.
- **Play (Gioco):** Incoraggia i bambini a sperimentare, provare cose nuove, armeggiare con i materiali, testare i confini, correre rischi e ripetere.

In anni di ricerca Resnick e il suo team hanno notato come non sia stato sempre semplice mettere in pratica questi quattro principi. Le classi sono basate spesso su vecchi standard educativi e valutazioni e non è facile integrare approcci creativi basati sulla codifica ma fortunatamente, come visto in precedenza, questo sta cambiando. Nel prossimo capitolo andrò a fare una breve panoramica di quella che è la storia del giocattolo come strumento educativo per rendere l'apprendimento creativo.

IL GIOCO COME ATTIVITÀ EDUCATIVA:

dal giocattolo educativo all'utilizzo del
coding e della robotica

3.1 Cos'è il giocattolo educativo

I giocattoli educativi sono oggetti, generalmente progettati per i bambini, destinati a soddisfare uno scopo educativo come aiutare un bambino a sviluppare un'abilità o insegnarli un argomento particolare. Si potrebbe pensare che tutti i giochi possano insegnare qualcosa, tuttavia un gioco può essere definito veramente educativo solo nel caso in cui contenga dei contenuti educativi presentati in maniera esplicita e sia stato pensato per trasmetterli al fine di raggiungere un preciso obiettivo di crescita. In quanto strumento di crescita e formazione, il gioco educativo potrà avere obiettivi differenti, misurati in base alla fascia d'età dei bambini a cui si rivolge, e intenderà sviluppare determinate capacità ed abilità, siano esse di tipo motorio, come la coordinazione, o intellettuale, come la memoria e l'associazione di idee.

3.2 I primi giocattoli educativi

L'identificazione di giocattoli aventi uno scopo esplicitamente educativo risale alla fine del 1600, quando il filosofo britannico **John Locke** notò i figli di un amico giocare con blocchi di legno con incise lettere dell'alfabeto. Locke scoprì che i bambini in questo modo avevano imparato a sillabare. Incuriosito, descrisse i blocchi nel suo popolare libro del 1693, "**Some Thoughts Concerning Education**", affermando che "*l'utilizzo di blocchi con incise le lettere per insegnare ai bambini l'alfabeto, avrebbero reso l'apprendimento un'esperienza più piacevole*"⁽¹⁹⁾. J. Locke non ha scritto una propria filosofia dell'educazione ma ha approfondito le diverse teorie della riforma educativa inglese del XVII secolo, introducendovi alcune idee personali. "**I blocchi di Locke**"^(Figura 12), come vennero chiamati, furono i primi giocattoli progettati per far emergere le abilità e gli interessi dei bambini. Tuttavia, ci sono voluti altri 100 anni perché i blocchi di legno diventassero popolari e ciò avvenne grazie al contributo dell'educatore

tedesco **Friedrich Fröbel**. Nell'ultima parte della sua vita, Fröbel, si dedicò all'educazione pratica, concentrandosi sui bambini piccoli tanto che, nel 1837 aprì un istituto per bambini nel villaggio di Bad Blankenburg. Nell'istituto perfezionò le sue tecniche di educazione della prima infanzia, incentrate sui concetti di gioco guidato dall'attività per il processo di apprendimento e il suo pensiero divenne molto influente per i pedagogisti degli anni successivi. Per aiutare i bambini in questo processo, Fröbel progettò, nel 1850, una serie di giocattoli educativi che ha chiamato "**Gifts**" (in italiano "regali")^(Figura 13), una serie di materiali sempre più complessi progettati per stimolare la creatività tra i bambini. Nel 1880, i blocchi di legno progettati da Fröbel ispirarono altri giocattoli educativi auto-elaborati, gli "**Anchor Stone Blocks**" realizzati in Germania dai fratelli Lillenthal e i "**materiali sensoriali**" di **Maria Montessori**^(Figura 14). Quest'ultima fu un'educatrice, pedagoga, filosofa, medica, neuropsichiatra infantile e scienziata italiana, internazionalmente nota per il metodo educativo che prende il suo nome (Il metodo Montessori), adottato in migliaia di scuole dell'infanzia, elementari, medie e superiori in tutto il mondo. Con l'avvento della società industriale e con l'arrivo del consumo di massa, il mondo dei giocattoli cominciò a cambiare. Iniziarono ad essere commercializzati giocattoli prefabbricati che ebbero il risultato di espropriare il bambino dell'azione della manipolazione delle cose e della progettazione dell'intera struttura ludica. Con i materiali sensoriali, Maria Montessori, cercò di riportare il giocattolo agli elementi essenziali delle attività ludiche naturali dei bambini. Il kit era costituito da una varietà di oggetti in legno colorati di diverse finiture e forme geometriche. Inoltre, i materiali sensoriali erano progettati per avere "**controllo dell'errore**", vale a dire che il bambino poteva auto-correggersi, piuttosto che dover chiedere aiuto all'insegnante. L'idea era che il controllo dell'errore aiutasse il bambino a sviluppare l'indipendenza e le capacità di risoluzione dei problemi⁽²⁰⁾.



Figura 12- I blocchi di Locke



Figura 13- "Gifts" di Friedrich Fröbel



Figura 14- I "materiali sensoriali" di Maria Montessori

3.3 Il secolo del bambino

All'inizio del XX sec., ci fu un cambiamento importante per quanto riguarda l'educazione, questo secolo venne definito da Ellen Key "**secolo del bambino**" (1902) ⁽²¹⁾ e l'attenzione si focalizzò su scoperte nell'ambito della psicologia dello sviluppo riguardanti il linguaggio, il gioco, la motricità infantile ecc., che vennero recepite dapprima dalla borghesia colta. L'idea che il bambino avesse una propria personalità, della quale si doveva incoraggiare lo sviluppo, determinò un nuovo orientamento educativo. Il bambino, in quel periodo, sentiva il bisogno di ritrovare negli strumenti del gioco una dinamicità simile a quella che intuiva nella realtà adulta. La fine e l'inizio del secolo segnarono non solo un periodo di grandi mutamenti storici, sociali e culturali ma rappresentarono anche un momento di grande innovazione nella produzione del giocattolo. Nella prima parte del XX secolo, nacquero una varietà di nuovi materiali come la plastica e i processi di produzione divennero sempre più automatizzati. Ciò sostenne lo sviluppo di giocattoli educativi come i giocattoli da costruzione, poiché consentì la standardizzazione dei pezzi. **Meccano, Erector Sets, Tinkertoy e Lincoln Logs**

(Figura 15) vennero presentati tutti all'inizio del XX secolo e promossi per sviluppare capacità motorie, incoraggiare il gioco libero e la creatività e introdurre i bambini alle idee di ingegneria e costruzione. Gli educatori di quel periodo erano consapevoli dei cambiamenti sociali causati dall'industrialismo e speravano di interessare i giovani a possibili nuove carriere. A metà degli anni '50, apparvero giocattoli da costruzione sempre più a tema ingegneristico, tra cui travi, colonne e pannelli di plastica che potevano essere assemblati per creare vari modelli. È alla fine di quegli anni che nacque la linea di mattoncini assemblabili LEGO la cui produzione iniziò nel 1949, ma soltanto nel 1958 ^(Figura 16-17) essi assunsero la particolare forma che ne caratterizzerà l'assemblaggio ⁽²²⁾.

3.4 Sviluppo dell'elettronica

Nel 1948, attraverso la realizzazione del primo modello di transistor, per opera dello studio e della ricerca di tre scienziati americani: William Shockley, Walter Brattain e John Bardeen, si sviluppò l'elettronica. Questa nuova introduzione, rese possibile la realizzazione di dispositivi elettronici come "il com-



Figura 15- Pubblicità Meccano 1920



Figura 16- Bambini alle prese con Lego Mursten, 1958

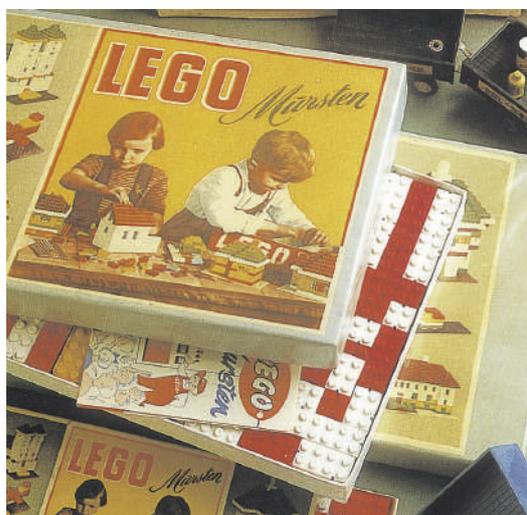


Figura 17- Lego Mursten, 1958

puter”, il quale esisteva già da prima dell’introduzione dei circuiti integrati e dei transistor, ma il suo funzionamento era dovuto a migliaia di valvole legate tra loro e a milioni di contatti saldati a mano, che ne richiedevano, per la costruzione e per l’utilizzo, l’impiego di enormi spazi. L’inizio della diffusione dei primi computer segnò profondamente anche il mondo dei giocattoli educativi. Come visto in precedenza, i giocattoli educativi erano pensati anche per spingere i bambini nell’apprendimento di attività che avrebbero portato a possibili nuove carriere, per questo motivo, i giocattoli di quel periodo non potevano che esporre i bambini all’apprendimento dei circuiti digitali e delle loro applicazioni. Qui di seguito alcuni esempi:

- **Geniac** (23) (Figura 18-19) era un giocattolo educativo venduto dal 1955 fino agli anni '60 per circa \$ 20, progettato e commercializzato da Edmund C. Berkeley, con Oliver Garfield. Il kit comprendeva interruttori rotanti, batterie a secco, lampadine e un manuale di istruzioni che spiegava in dettaglio come cablare ciascun circuito. Il suo funzionamento era semplice, i bambini dovevano ruotare manualmente gli interruttori per ottenere risposte a una sequenza di problemi. Circa un anno dopo uscì una versione simile chiamata “Brainiac”.

- Il kit **Lectron**, (24) (Figura 20) distribuito Raytheon nel 1967 negli Stati Uniti e in Europa. Il kit insegnava ai bambini a costruire un circuito elettrico con blocchi contenenti uno o più componenti elettronici o una parte di interconnessione. Il kit conteneva anche: un manuale di esperimenti e due schede di lavoro. I bambini potevano utilizzare il kit costruendo i circuiti proposti dal manuale, ad esempio: un termometro elettronico, un generatore di suoni o una radio (questi sono solo alcuni). Unire i blocchi era molto semplice, ogni blocco era dotata di piccoli magneti nei punti in cui non erano a contatto con un altro blocco. I lati e il fondo di ogni blocco erano realizzati in plastica trasparente in modo da poter vedere cosa c'era dentro e la parte superiore aveva il simbolo schematico del componente elettronico all’interno. Ogni blocco poteva essere fissata magneticamente a una piastra metallica che fungeva da superficie di lavoro, eliminando la necessità di saldature. Ciò aveva il vantaggio della sicurezza e la possibilità di riorganizzare i blocchi per

cambiare l'effetto sul circuito. Le tavole di lavoro su cui i bambini costruivano il circuito potevano essere appoggiate sui loro supporti incorporati o appese al muro in modo che un'intera classe potesse vedere la dimostrazione.



Figura 18- Geniac, 1955

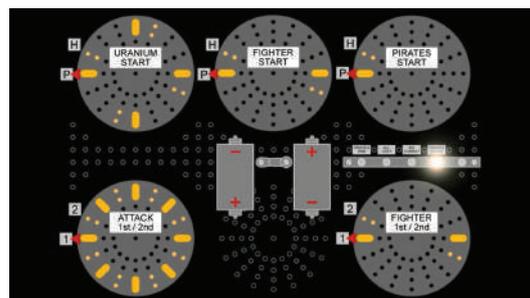


Figura 19- Esempio di esercizio con Geniac



Figura 20- Pubblicità Lectron 1967 (sinistra) e esempio di assemblaggio di un circuito (destra)

- Il **Sapientino**, ⁽²⁵⁾ (Figura 21) un gioco elettronico prodotto dalla Clementoni. Nato nel 1967, era costituito da una serie di schede, che presentavano dei fori in prossimità delle domande e delle risposte, riguardanti un'ampia varietà di argomenti, a cui, dal 1994, si è aggiunto l'inglese. Le schede andavano posizionate su un apposito tavolo, presente nella confezione. Con gli spinotti in dotazione era possibile selezionare la domanda e segnalare la risposta corrispondente. Se quest'ultima risultava esatta si chiudeva il circuito elettrico e veniva emesso un segnale acustico (o visivo). L'idea venne a Mario Clementoni l'industriale maceratese che lavorava in un'azienda di strumenti musicali a Pesaro e che decise di mettersi in proprio in una fase storica in cui l'industria italiana guardava con interesse soprattutto ai beni di prima necessità per colmare le difficoltà del periodo post bellico. L'idea fu per l'appunto quella di pensare ad un giocattolo che potesse essere di aiuto nel percorso di crescita dei bambini. La Clementoni con il Sapientino ha dato vita, in Italia, ad un nuovo mercato dei giocattoli che negli anni si sono evoluti in qualcosa di sempre più tecnologico.



Figura 21- Primo Sapientino , 1967

3.5 Coding e Robotica educativa

Durante gli anni '70, Seymour Papert iniziò a sperimentare la robotica con il linguaggio di programmazione Logo (vedi capitolo 1 paragrafo 1.3) creando il primo robot educativo, **Logo Turtle** (Figura 22-23-24). La funzione principale era quella di un movimento programmato dal computer. La tartaruga aveva la forma sferica di un caschetto ed era costituita da: due ruote che ne permettevano il movimento e una penna retrattile al centro che poteva essere utilizzata per disegnare con il comando "pen-

down". L'obiettivo principale di Papert era quello di dare agli studenti un mezzo che potesse aiutarli ad imparare la programmazione come si impara una lingua. Secondo lui il mondo che circondava i giovani, in quel momento, rendeva la scienza e la tecnologia aliene alla maggioranza delle persone, tutto ciò li portava a pensare che dispositivi come i robot appartenevano agli "altri". Questo ambiente, dissuadeva i giovani dal perseguire la ricerca in scienze, matematica e tecnologia, per questo motivo con il progetto LOGO Turtle si voleva rendere la robotica accessibile a tutti.



Figura 22- Primo prototipo di Logo Turtle



Figura 23- S. Papert e alcuni studenti alle prese con Logo Turtle

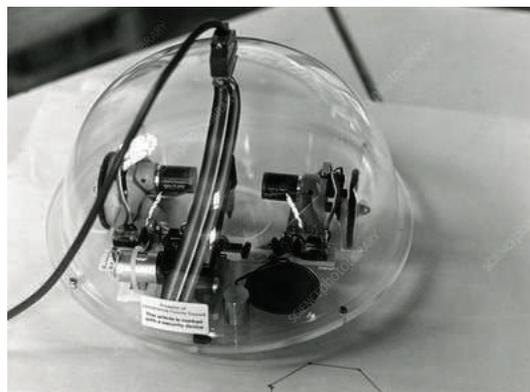


Figura 24- Logo Turtle

A metà degli anni '80 il gruppo di ricerca LOGO guidato da Papert al MIT di Boston iniziò una collaborazione con il gruppo LEGO, creando il sistema LEGO/LOGO, ovvero una combinazione dei prodotti Lego Technic (comprendente oltre ai classici mattoncini anche travi, ingranaggi e motori) e il linguaggio LOGO. Questo nuovo sistema ampliava di gran lunga le possibilità offerte da Logo Turtle; infatti non esisteva più una forma precostituita (la tartaruga) ma i bambini potevano ora costruire i loro robot limitati solo dalla fantasia e dai pezzi disponibili. Alla fine del 1980, il sistema LEGO/LOGO venne messo in commercio con il nome **"LEGO TC Logo"** ⁽²⁶⁾

(Figura 25) da parte del gruppo Lego. I componenti LEGO comunicano con il computer attraverso una scatola nera con 12 porte per collegare i cavi. Alcune delle porte sono porte di uscita, utilizzate per inviare comandi, come accensione o spegnimento ai motori o alle luci. Altre porte sono chiamate ingressi e sono utilizzate per ricevere messaggi o informazioni dai sensori (di due tipi: sensore tattile e sensore ottico). I robot costruiti avevano però dei limiti, dovevano essere collegati al computer con dei fili limitando di molto la loro mobilità e usabilità. Dal 1992 al 1996 venne creata una seconda generazione di "mattoncini" Lego programmabili utilizzando l'esperienza fatta sul campo con "LEGO TC Logo" e nel 1998 la Lego, rese disponibile un nuovo prodotto chiamato "LEGO Mindstorms Robotic Invention Kit" (Figura 26) composto da 717 pezzi tra mattoncini LEGO, motori, ingranaggi, sensori e il "Mattone RCX" contenente tre porte di ingresso e tre di uscita e collegato ad un micro controllore. Questo nuovo prodotto venne chiamato con un nome preso dal libro di Seymour Papert, questo dimostrò ancora una volta la fedeltà nei confronti della teoria Costruzionista. Il Kit "LEGO Mindstorms" venne sostituito da altre versioni. Quella del 2006 prevedeva un "mattoncino NXT" dotato di quattro pulsanti e un display LCD per una programmazione rapida. E quella del 2013, "mattoncino Ev3" in cui sono stati aggiunti una porta USB e uno slot SD.

3.6 Programmazione tangibile

Dopo l'introduzione di Logo nelle scuole, è diventato chiaro ai ricercatori che la maggior parte dei bambini non era pronta per iniziare a programmare in modo tradizionale (cioè, digitando il codice in un computer utilizzando

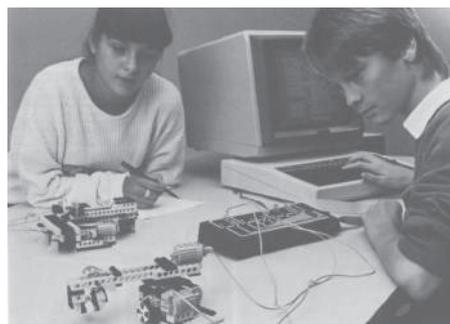


Figura 25- Studenti alle prese con Lego TC Logo



Figura 26- tre generazioni di Lego Mindstorms: Lego RCX (1998), Lego NXT (2006), Lego Ev3 (2013)

una tastiera) fino ad almeno i 10-14 anni. Radia Perlman, laureata presso il MIT Logo Lab a metà degli anni '70, riteneva che il maggiore ostacolo all'accesso dei bambini alla programmazione informatica non era solo il linguaggio, ma anche l'interfaccia utente.

Per questo motivo iniziò a progettare interfacce che consentissero anche ai bambini in età prescolare di imparare a programmare Logo Turtle ⁽²⁷⁾. Ne uscirono fuori due progetti importanti: **Button Box** (1974) ^(Figura 27) e la **Slot Machine** (1976) ^(Figura 28).

La Button Box è stata la sua prima creazione, era una scatola di plastica trasparente di 50x50x10 cm con quattro gruppi di pulsanti:

1. La casella dei numeri da 1 a 10
2. Il riquadro delle azioni: AVANTI, INDIETRO, DESTRA (gira in senso orario), SINISTRA (gira in senso antiorario), "accendi la luce", "spegni la luce" e "suona il clacson"
3. La scatola dei ricordi: "Inizia a ricordare", "Fallo" e "Lascia perdere"
4. Il riquadro delle quattro procedure: GIALLO, ROSSO, VERDE e BLU.

Perlman scoprì presto due grandi problemi con questo sistema, in primo luogo, l'utilizzo dei pulsanti era troppo astratto per i bambini più piccoli. In secondo luogo, il sistema non prevedeva in alcun modo di modificare un programma una volta registrato. Se il bambino trovava un errore, doveva registrare nuovamente l'intera sequenza dall'inizio.

Il sistema successivo di Pelman, la Slot Machine, ha affrontato entrambi i problemi. Invece di controllare la tartaruga usando i bottoni, usava schede di plastica che potevano essere inserite in una delle tre file colorate (rosso, giallo e blu). Sul lato sinistro di ogni fila c'era un pulsante "Esegui". Quando il bambino premeva il pulsante, la tartaruga eseguiva le azioni raffigurate sopra ogni carta, in sequenza da sinistra a destra. come un'azione veniva eseguita, una lampada sotto la carta si illuminava restituendo un feedback visivo. La Slot Machine offriva un grande vantaggio ovvero la capacità di modificare un programma direttamente, aggiungendo, riorganizzando e rimuovendo le carte. Per affrontare il problema della procedura con cui venivano eseguite le sequenze, Perlman introdusse carte speciali i cui colori corrispondevano ad una fila colorata.



Figura 27- Bambini alle prese con Logo Turtle controllata dalla Button Box

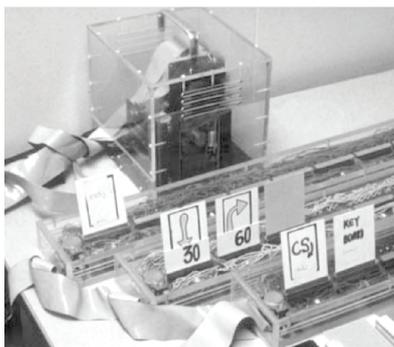


Figura 28- Slot Machine

Dall'esperienza di Radia Perlman si parlò sempre più di **"programmazione tangibile"**, una modalità di programmazione che utilizza un'interfaccia tangibile tra il modo digitale e quello reale. Il termine venne coniato da Hideyuki Suzuki e Hiroshi Kato sviluppatrici di **"AlgoBlock"** (1993)^(Figura 29).

Questo linguaggio di programmazione era simile a Logo ma utilizzava blocchi fisici che potevano essere collegati tra loro manualmente per formare un programma. Questi blocchi erano collegati ad un computer e gli studenti potevano visualizzare e muovere un sottomarino rappresentato graficamente sullo schermo ^(Figura 30). I blocchi erano grandi 15 cm e ad ognuno corrispondeva un comando, alcuni disponevano di interruttori fisici per il controllo dei parametri. AlgoBlock era pensato per essere utilizzato in un contesto scolastico per lavori di gruppo, inoltre la tangibilità consentiva di coordinare meglio i gesti: allungare la mano, indicare, guardare, volgersi verso, tenere, ecc. ⁽²⁸⁾

Negli anni successivi vennero sviluppati studi interessanti sulla programmazione tangibile, tra questi troviamo:

"Programmazione tangibile con i treni" ^(Figura 31) del 1998 di Fred Martin e Genee LynColobong ⁽²⁷⁾. Questo progetto dava la possibilità ai bambini di programmare facendo muovere un treno giocattolo. Il treno conteneva un piccolo microprocessore e un ricevitore a infrarossi e intorno alla pista c'erano una serie di segnali contenenti trasmettitori a infrarossi. I bambini controllavano il treno spostando i segni e i segnali lungo i binari. La raccolta di segnali includeva: stop, più veloce, più lento, luci accese, luci spente e clacson. I bambini trovarono divertente organizzare scenari realistici (es. far accendere le luci del treno quando entrava in un tunnel) e al tempo stesso erano impegnati in attività di programmazione.



Figura 29- Algo Block

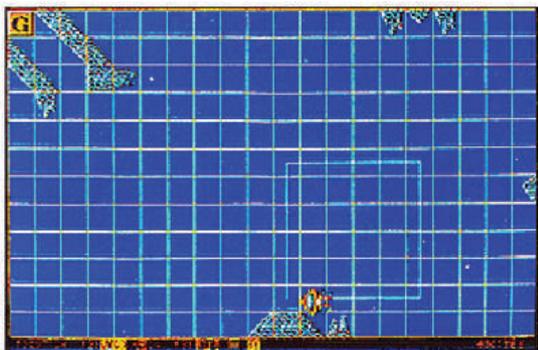


Figura 30- Gioco del sottomarino su computer

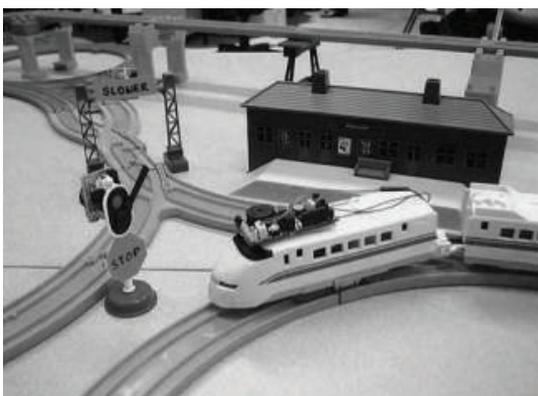


Figura 31- Prototipo del treno di F. Martin e G. LynColobong

Un altro esempio interessante di programmazione tangibile è:

Quetzal, (Figura 32-33) un progetto del 2007 di Michael Horn e Robert J.K. Jacob ⁽²⁹⁾. Quetzal era un linguaggio di programmazione per il controllo del Mattoncino LEGO Mindstorms RCX. Consisteva nell'incastro di piastre di plastica che collegate tra loro formavano un flusso di comandi. Il modo in cui venivano posizionate le piastre poteva generare un loop infinito ovvero il robot avrebbe eseguito il programma fino allo spegnimento. Il passaggio tra la programmazione tangibile e il pc avveniva tramite la scansione di ogni piastra con una fotocamera digitale standard collegata al pc.

Sulla base Quetzal, sempre nel 2007, venne sviluppato il progetto **"Tern"** (Figura 34) che prevedeva la programmazione con blocchi in legno. I bambini potevano collegare i blocchi per formare programmi per computer e le azioni includevano: loop e subroutine. Anche in questo caso venne utilizzata la scansione di ogni blocco tramite una fotocamera digitale.

Per molto tempo l'uso della programmazione tangibile è stato limitato quasi interamente ai

laboratori e alla ricerca. Solamente negli ultimi anni con l'introduzione del Coding e delle discipline STEM nella didattica della scuola primaria e dell'infanzia, si è potuto assistere allo sviluppo e alla commercializzazione di prodotti programmabili anche senza l'utilizzo di uno schermo.



Figura 32- Componenti del linguaggio di programmazione Quetzal



Figura 33- Esempio di programmazione Quetzal



Figura 34- Blocchi di legno utilizzati per la programmazione Tern

3.7 XXI secolo, Coding tra robotica educativa e realtà virtuale

Negli ultimi anni è stato possibile introdurre sempre più, a scuola e a casa, il Coding e la robotica educativa come mezzo di apprendimento, questo ha favorito per la didattica un sistema scolastico innovativo e completo rendendo anche accessibile a tutti l'insegnamento dell'informatica e della robotica.

I prodotti, messi in commercio e non, utilizzati per il coding e la robotica educativa si sono concentrati sempre più su un nuovo tipo di interfaccia di programmazione e su dispositivi programmabili che allargano lo spettro di quello che gli studenti possono programmare.

Qui di seguito alcuni esempi di prodotti utilizzati dal 2000 in poi:

- **Scratch**, (Figura 35) (vedere capitolo 4 paragrafo 4.2) è un software educativo sviluppato nel 2003 nell'ambito del Lifelong Kindergarten Group del Mit Media Lab. Scratch oltre ad essere un linguaggio di programmazione è un ambiente di sviluppo visuale dotato di icone grafiche e altri meccanismi che ne facilitano l'utilizzo. Il tipo di programmazione utilizzato in Scratch viene definito "a blocchi" e permette all'utente di ottenere dei risultati in tempi rapidi. I blocchi colorati sono creati per agganciarsi l'un l'altro e formare delle colonne. Ognuna di queste colonne di blocchi viene chiamata script e fanno parte del codice vero e proprio decodificato dalla macchina (il computer).

Con l'avvento di Scratch iniziarono a nascere altri linguaggi di programmazione simili come: SNAP! (2011), Blockly (2012), Swift Playground (2016) e Code Builder (2017). L'esempio di Scratch è fondamentale poiché ha permesso alla maggior parte dei robot programmabili presenti sul mercato di essere controllati da linguaggi di programmazione semplici ed adeguati per bambini e ragazzi. Scratch negli anni ha rinnovato la sua interfaccia rendendo sempre più semplice la programmazione e la condivisione dei progetti con altri utenti. Nel 2014 inoltre è stata lanciata la versione Scratch Jr. per bambini dai 5 ai 7 anni, questo ha permesso anche ai bambini più piccoli di relazionarsi con le basi della programmazione.

- **BeeBot**, (Figura 36) (vedere capitolo 5 paragrafo 5.5) è stato uno dei primi robot programmabili per bambini presente sul mercato,

messa in produzione nel 2009 dall'azienda TTS. BeeBot utilizza una programmazione tangibile e come precedentemente visto non era scontata la sua commercializzazione poiché la programmazione tangibile era ancora relegata nei laboratori di ricerca. È stata utilizzata nei primi laboratori di Coding nelle scuole e ad oggi è presente sul mercato nella sue più recenti versioni come "BlueBot" (2015) (Figura 37). BeeBot è un robot a forma di ape in grado di memorizzare una serie di comandi e muoversi di conseguenza. I comandi sono presenti sul dorso come pulsanti e sono: avanti, indietro, svolta a sinistra, svolta a destra, pausa, avvia e cancella il programma. Questa tipologia di robot è pensata per bambini della scuola dell'infanzia ma viene utilizzato anche nei laboratori di coding per la scuola primaria.

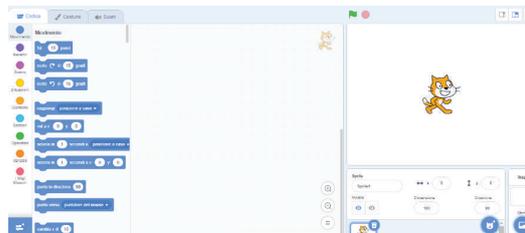


Figura 35- Interfaccia Scratch



Figura 36- BeeBot



Figura 37- BlueBot, permette ai bambini di vedere i circuiti interni

- Sempre nel 2009, venne lanciata la prima versione di **Legò WeDo** (Figura 38) per gli studenti delle scuole elementari. Questa prima versione includeva 158 pezzi Lego, due sensori (sensore di inclinazione e sensore di movimento), un motore e un hub USB responsabile del controllo e dell'alimentazione. Nella seconda versione **Legò WeDo 2.0** (Figura 39) (lanciata nel 2015) sono presenti 280 pezzi Lego, i due sensori, un motore e una Smart Hub programmabile che si collega a computer e tablet tramite bluetooth permettendo una maggiore libertà di movimento.

La prima versione poteva essere programmata o con Lego WeDo Software o con Scratch. La seconda versione può essere programmata con Scratch o con l'app aggiornata WeDo 2.0 LEGO Education.

Legò Mindstorm e Legò WeDo non sono gli unici kit educativi Lego, nel 2019 venne messo in commercio un ulteriore kit educativo, **Legò Spike Prime** (Figura 40). Questo prodotto si rivolge a ragazze e ragazzi dai 10 anni in su ed uno degli obiettivi è quello di ridurre il cosiddetto "gap di genere" ancora esistente in ambito scientifico tra ragazze e ragazzi. La scelta dei colori diventa significativa, non più colori scuri e troppo seri ma colori pastello. All'interno del kit troviamo come in quelli precedenti il cuore ovvero l'hub programmabile, diversi sensori (colore, distanza, forza/contatto), motori (grandi e medi) e 523 pezzi Lego. Nel 2021 è stata prodotta la versione **Spike Essential** (Figura 41) per bambini dai 6 anni in su. Il kit contiene un hub, motori piccoli, led, un sensore colore e 449 pezzi Lego di cui alcuni con forme predefinite (es. personaggi). Sono comprese inoltre, come in altri kit Lego, unità didattiche che gli insegnanti possono utilizzare per creare lezioni da 45 minuti. I kit educativi Lego⁽³⁰⁾ coprono tutti gli anni di scuola, dalla scuola primaria alla scuola superiore permettendo una continuità educativa.

- Oltre a questi kit robotici alle scuole sono stati proposti negli anni altri prodotti anche meno costosi è il caso di **mBot** (Figura 42) un robot da assemblare, presentato da MakeBlock nel 2015, pensato per ragazzi e ragazze dai 12 anni. Il kit base viene fornito completamente smontato poiché l'obiettivo è quello di dare delle basi di elettronica, o quantomeno sapere di quali parti è composto il robot così da comprenderne meglio il funzionamento.



Figura 38- Lego WeDo prima versione, 2009



Figura 39- Lego WeDo 2.0, 2015



Figura 40- Lego Spike Prime, 2019



Figura 41- Lego Spike Essential, 2021

Ad un componente principale (mCore) vengono collegati due motori e due ruote per lo spostamento, un sensore ultrasonico per rilevare gli ostacoli, un modulo bluetooth (o Wi-Fi in base alla versione) per la comunicazione senza fili, una batteria, un sensore chiamato "Me Line Follower" che è in pratica un sensore ottico che permette di muovere il robot su un percorso prestabilita tramite informazioni "visive". La programmazione avviene tramite mBlock, un software di programmazione molto simile a Scratch. MBot è un ottimo inizio per insegnare la programmazione e le basi della robotica. Inoltre, per stimolare la creatività e la collaborazione è stata creata una competizione chiamata MakeX⁽³¹⁾. Questa competizione si basa sull'utilizzo del kit mBot ma si può partecipare anche utilizzando altri kit.

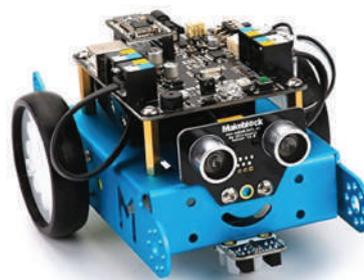


Figura 42- mBot

Ad oggi i giocattoli educativi stanno entrando in una nuova sfera tecnologica ovvero quella della realtà virtuale e della realtà aumentata, in questo modo i bambini e i ragazzi possono creare contenuti molto differenti da quelli proposti fino ad ora e in una modalità nuova, il lato negativo è che spesso si va a perdere la fisicità del giocattolo in favore del gioco totalmente virtuale. Qui di seguito due esempi utilizzati nella didattica e a casa:

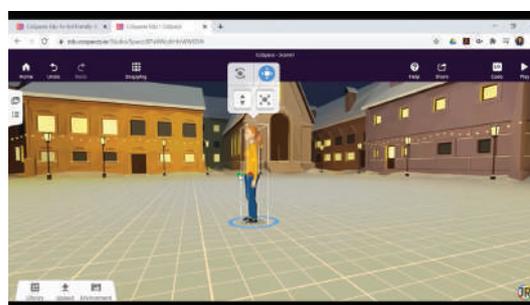


Figura 43- Spazio virtuale in CoSpaces EDU

- **CoSpaces EDU**,⁽³²⁾ (Figura 43) è una piattaforma rilasciata nel 2017 che permette agli studenti di creare, esplorare e condividere spazi di realtà virtuale e ai docenti di costruire ambienti virtuali di apprendimento, progettando lezioni per tutte le discipline. Con l'inizio del 2018, CoSpaces ha modificato la struttura della sua offerta, proponendo due 'license plans' distinti: uno 'BASIC' totalmente gratuito ed uno 'PRO' dietro il pagamento di un corrispettivo legato al numero di 'seat' ovvero account collegati. Non è solo una piattaforma per contenuti immersivi ma supporta attività di coding e storytelling. Le azioni dei personaggi possono essere programmate con un linguaggio di programmazione a blocchi (Figura 44-45), molto simile a Scratch. CoSpaces rende in questo modo la creazione in VR davvero facile, accessibile e divertente, in modo che chiunque può immergersi nelle creazione di contenuti 3D personalizzati. La visione dei contenuti inoltre può essere effettuata tramite visori in cartone per Smartphone, Gear VR, Google Daydream o semplicemente tramite PC ed App

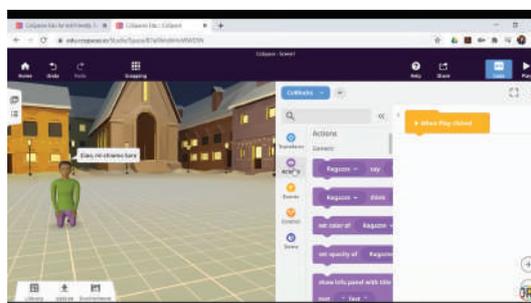


Figura 44- Programmazione delle azioni del personaggio in CoSpaces EDU

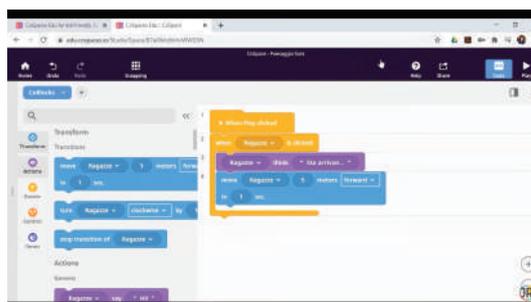


Figura 45- Interfaccia di programmazione in CoSpaces EDU

- **Botzees** ⁽³³⁾, un kit robotico prodotto nel 2019 da Pay Technology, per bambini dai 4 anni in su, che combina creatività, costruzione e programmazione. I bambini possono programmare e codificare 6 diversi Botzee ^(Figura 46) pre-progettati o crearne di propri. Ai Botzees può essere insegnato a muoversi, ballare, emettere suoni e illuminarsi utilizzando un'app gratuita. Nel kit sono presenti 130 pezzi facili da impugnare con bordi arrotondati, un componente principale programmabile, un servomotore e un sensore di prossimità. L'app aiuta i bambini nella costruzione dei robot ed al suo interno sono presenti diversi contenuti in realtà aumentata. I bambini possono utilizzare il robot in due modalità differenti:
 1. possono programmare il robot all'interno dell'app (programmazione a blocchi) e successivamente inquadrandolo possono dare vita ad avventure in realtà aumentata ^(Figura 47).
 2. possono utilizzare il robot in modalità "Normal Control", questa tipologia di controllo non richiede la programmazione ^(Figura 48).

(Figura 47).

2. possono utilizzare il robot in modalità "Normal Control", questa tipologia di controllo non richiede la programmazione

(Figura 48).

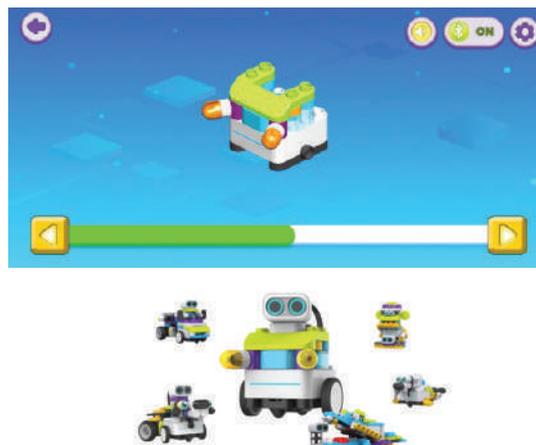


Figura 46- Istruzioni su come costruire Botzees sull'app "Botzees EDU", sei possibili configurazioni di Botzees



Figura 47- Programmazione sull'app "Botzees EDU"



Figura 48- Funzione "Normal Control" sull'app "Botzees EDU"

PRIMI GIOCATTOLI EDUCATIVI

GIOCATTOLI DA COSTRUZIONE

SVILUPPO DELL'ELETTRONICA

I blocchi di Locke



1693

1850

Gifts



I materiali sensoriali



1909

1920

Meccano



LEGO



1949

1955

Geniac



Kit Lector



1967



Sapientino

Logo e Logo Turtle

Nasce il linguaggio di programmazione Logo utilizzato per programmare Logo Turtle ed i robot nati successivamente

1969



1974

Button Box



Lego TC Logo



1980

1976

Slot Machine



1993

Algo Block



Lego Mindstorms



1998

1998

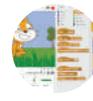
Programmazione tangibile con i treni



Scratch

Nasce il linguaggio di programmazione Scratch, molto più semplice di Logo ed ad oggi il più utilizzato per programmare i robot educativi

2003



2007

Quetzal e Tern



Lego WeDo



2009

BeeBot



Lego WeDo 2.0



2015

BlueBot



mBot



2017

CoSpaces Edu



Botzees



2019

Lego Spike Prime



2021

Lego Spike Essential



PROGRAMMAZIONE CON SOFTWARE E PRIMI ROBOT EDUCATIVI

PROGRAMMAZIONE TANGIBILE

SVILUPPO DEL PENSIERO COMPUTAZIONALE

CODING E ROBOTICA EDUCATIVA IN COMMERCIO

REALTÀ AUMENTATA E VIRTUALE PER CODING E ROBOTICA

LINGUAGGI DI PROGRAMMAZIONE PER BAMBINI

4.1 Cos'è un linguaggio di programmazione

Come detto in precedenza, fare Coding vuol dire programmare ovvero scrivere un programma che implica una sequenza di azioni che devono essere eseguite per raggiungere un determinato obiettivo. Chi programma comunica ad una macchina cosa fare e lo fa attraverso un codice che questa è in grado di comprendere. Questo tipo di comunicazione avviene tramite un microprocessore, un componente in grado di leggere, interpretare ed eseguire le istruzioni. Esistono linguaggi di programmazione di alto livello (esempio: Python) e di basso livello (esempio: Assembly): i primi permettono al programmatore di lavorare su vere e proprie operazioni logiche, lasciando a un programma, detto compilatore, il compito di tradurle in istruzioni per il processore scritte in linguaggio macchina, ossia in sequenze di codice binario comprensibili dal processore. I secondi sono più vicini al linguaggio macchina e le istruzioni scritte con tali linguaggi di programmazione non devono essere perciò sottoposte al lavoro compilatore.

Nonostante le differenze, formali e sostanziali, i linguaggi di programmazione si strutturano attorno a un certo insieme di concetti comuni. I linguaggi che saranno analizzati in questo capitolo sono linguaggi semplici, quindi fondamentali per l'utilizzo a scopo didattico. In una pubblicazione del 2012 **“Nuovi quadri per lo studio e la valutazione del sviluppo del pensiero computazionale”**⁽³⁴⁾, gli autori Karen Brennan e Mitchel Resnick hanno individuato sette concetti, comuni a molti linguaggi di programmazione (che verranno poi utilizzati in Scratch):

- **Sequenze:** un concetto chiave nella programmazione è quello che ogni attività è espressa come una serie di singoli passi o istruzioni eseguibili dal computer e ogni sequenza corrisponde ad una azione o ad un comportamento che deve essere

prodotto; l'ordine delle istruzioni determina il risultato finale.

- **Loop:** il meccanismo del loop consente l'esecuzione multipla di una sequenza, grazie all'utilizzo della ripetizione che consente alle istruzioni di essere eseguite più volte utilizzando il minor numero di comandi.
- **Eventi:** gli eventi sono situazioni che si verificano solamente se viene eseguita una determinata azione. (es. un pulsante di avvio che attiva l'inizio di un video musicale).
- **Parallelismo:** questa caratteristica permette l'esecuzione contemporanea di sequenze diverse. Questo può avvenire per istruzioni date ad oggetti diversi, ma anche sullo stesso oggetto.
- **Condizioni:** è il verificarsi di determinate situazioni solamente in base certe condizioni, che supportano l'espressione di esiti diversi.
- **Operatori:** consentono al programmatore la manipolazione numerica e delle stringhe, grazie al supporto per espressioni matematiche e logiche.
- **Variabili:** concetto che consente l'archiviazione, il recupero e l'aggiornamento di valori, in quanto una soluzione algoritmica ad un problema richiede la memorizzazione, l'analisi e la rappresentazione di informazioni.

Nei paragrafi successivi (4.2 e 4.3) analizzerò Scratch e Scratch Jr, due tra i linguaggi di programmazione più utilizzati dai bambini. Con questa analisi sarà più semplice comprendere le differenze tra un linguaggio adatto a bambini della scuola primaria (Scratch) e uno adatto a bambini della scuola dell'infanzia (Scratch Jr).

4.2 Scratch

Come precedentemente accennato nel capitolo 3, "Scratch" è un linguaggio di programmazione visuale per bambini da 8 anni in su, sviluppato nel 2003 dal gruppo Lifelong Kindergarten del MIT Media Lab guidato da Mitchel Resnick. L'idea alla base di Scratch è stata ispirata dal lavoro della Computer Clubhouse un'organizzazione americana senza scopo di lucro che gestisce un programma di apprendimento extrascolastico gratuito in cui i giovani provenienti da realtà svantaggiate lavorano con tutor per esplorare le proprie idee, sviluppare nuove competenze e costruire fiducia in se stessi attraverso l'uso della tecnologia. I bambini di questa organizzazione volevano creare storie interattive, giochi e animazioni ma non c'erano strumenti adatti a loro. La prima versione, Scratch 1.0, era molto simile a Scratch 1.4. A quel tempo il sito web era un piccolo blog ma, con la crescita di Scratch, è stato ampliato e sono stati rilasciati Scratch 1.1, Scratch 1.2, Scratch 1.3 e Scratch 1.4. A questo punto, Scratch aveva milioni di utenti, progetti e molte nuove funzionalità. Dopo il rilascio di Scratch 2.0 nel 2013, il sito Web e l'interfaccia utente sono stati modificati. La comunità di Scratch ha continuato a crescere, raggiungendo 30 milioni di utenti e progetti nel 2018. Negli ultimi anni è stato nuovamente aggiornato nella versione Scratch 3.0 (2019) e Scratch 4.0 (2020).⁽³⁵⁾

L'idea di Mitchel Resnick, era quella di creare un modo facile e divertente di imparare a programmare, accessibile liberamente a tutti. Nel TED Talk "Insegnamo ai bambini a programmare"⁽³⁶⁾, da lui tenuto nel Novembre del 2012, dice che quando si parla di programmazione è utile fare un'analogia con la lettura e la scrittura. Quando si impara a leggere e scrivere, si aprono nuove opportunità per imparare molte altre cose e quando si impara a leggere si potrà anche leggere per imparare. Queste cose valgono anche per la programmazione, se si impara a programmare, allora si potrà anche programmare per imparare e quando si impara a programmare, si apre la possibilità di imparare molte altre cose. Lo scopo di Scratch non è, quindi, quello di farci diventare tutti programmatori, piuttosto, è quello di insegnarci a pensare in modo creativo.

Con "Scratch" è possibile creare un vasto numero di contenuti diversi, per esempio cartoni animati, animazioni e storie interattive, quiz e puzzle, ma anche dei semplici giochi. Ad oggi la comunità di Scratch è vasta e solo in Italia è stato utilizzato nel 2020 da 500 mila utenti (dal report annuale pubblicato sul sito scratch.mit.edu)⁽³⁷⁾. L'ambiente di programmazione è disponibile sia via web, sul sito ufficiale "scratch.mit.edu", sia come software utilizzabile offline, chiamato "Scratch Desktop". Nonostante "Scratch" sia stato progettato per la fascia di età 8- 16 anni, è considerato un ottimo modo per introdurre il coding anche agli adulti.

Entrando nel dettaglio, Scratch utilizza una programmazione a "blocchi", questo vuol dire che non è necessario digitare alcun codice come avviene, invece, nella programmazione tradizionale facilitando di molto l'apprendimento. Per creare un programma basta trascinare dei blocchi di codice, prepostati e divisi per colore unendoli insieme in un ordine logico. L'interfaccia è suddivisa in quattro aree:

- **Stage:** un grande rettangolo bianco posto nell'angolo superiore destro dello schermo. Viene considerato come il palcoscenico, dove gli utenti possono vedere l'anteprima del progetto.
- **Sprite:** una finestrella più piccola sotto lo stage che mostra l'elenco degli sprite (il termine con cui vengono chiamate le immagini animate nello Scratch ovvero i personaggi utilizzati). Nel progetto di default tra gli sprite troveremo la mascotte di Scratch che ha le sembianze di un gatto arancione.
- **Elenco dei Blocchi:** si trova a sinistra dello schermo ed è una colonna che contiene tutti i blocchi di programmazione, suddivisi in categorie diverse.
- **Area degli Script:** un grande rettangolo grigio al centro ed è la principale area di lavoro. Viene considerato come il backstage, dove si trascinano i blocchetti e si ordinano per creare il codice.

Interfaccia Scratch

Blocchi da inserire nel codice

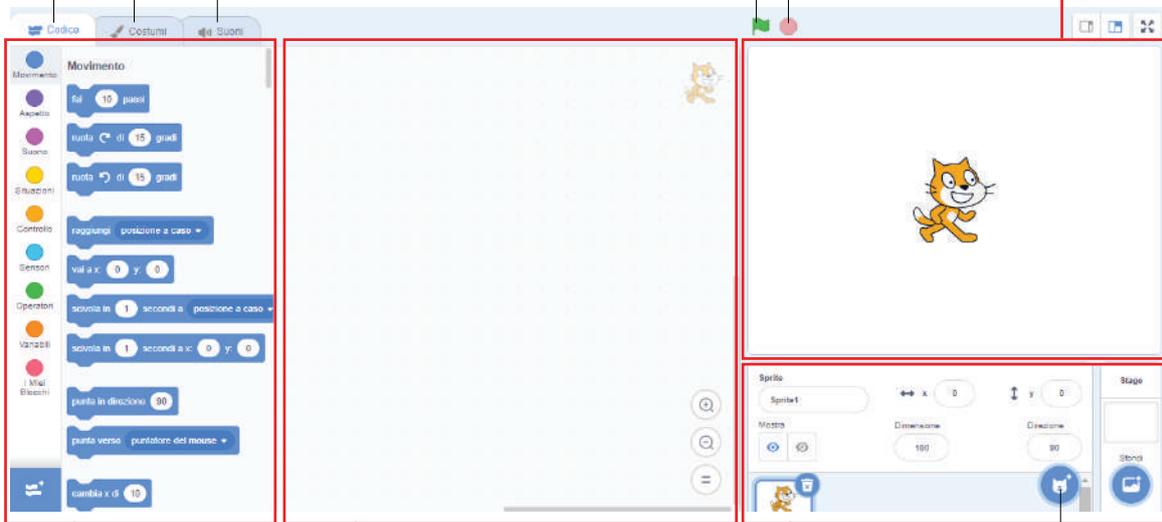
Costumi degli sprite

Permette di inserire suoni

Bandiera verde: fa partire l'animazione

Esagono rosso: ferma l'animazione

Area Stage



Area con elenco dei blocchi

Area degli Script

Area Sprite

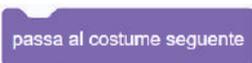
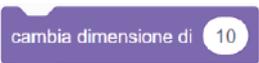
Questa opzione permette di scegliere un nuovo sprite



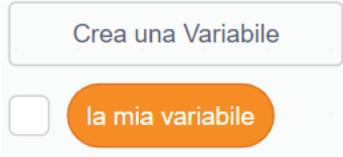
Scratch, grazie all'utilizzo di blocchi, da la possibilità di creare codici complessi in una modalità molto semplice.

I blocchi di Scratch sono organizzati in otto categorie diversificate per colore: **Movimento, Aspetto, Suono, Situazioni, Controllo, Sensori, Operatori e Variabili.** (Tabella 1)

Tabella 1- Le otto categorie di blocchi con alcuni esempi di comandi

BLOCCHI	COLORE	DESCRIZIONE	ESEMPI DI COMANDI
Movimento		Hanno lo scopo di modificare la posizione dello Sprite nello Stage.	avanti 
			ruota a destra 
			ruota a sinistra 
			raggiungi posizione (è possibile raggiungere una posizione casuale p dove si trova il puntatore del mouse) 
			vai a X:___ Y:___ (aggiungere coordinate) 
Aspetto		Hanno lo scopo di modificare lo Sprite: gli fanno dire qualcosa (appare un fumetto), gli cambiano costume, ne variano la dimensione e permettono di farlo sparire con una dissolvenza.	Dire "Ciao" (appare un fumetto) 
			Cambiare il costume dello Sprite 
			Cambiare la dimensione dello Sprite 
			Nascondere lo Sprite 

BLOCCHI	COLORE	DESCRIZIONE	ESEMPI DI COMANDI
Suono		Hanno lo scopo di produrre dei suoni (ad esempio "miao") o per agire sul volume di effetti sonori già in esecuzione.	Riprodurre un suono ("Miao" o una registrazione) 
			Cambiare il volume del suono 
Situazioni		Hanno lo scopo di creare degli eventi, cioè determinare quali accadimenti debbano scatenare l'avvio di qualcosa. Hanno una sagoma particolare in quanto presentano, nella parte superiore, una curvatura: questa forma sta ad indicare che tali blocchi devono trovarsi in cima alla sequenza dei comandi.	Quando si clicca sulla bandiera verde sullo schermo 
			Quando si preme un tasto della tastiera (spazio, freccia su, freccia giù, a, b ecc.) 
Controllo		Hanno lo scopo di controllare il flusso del programma con condizioni o parametri di ripetizione. Molti dei blocchi di questa categoria hanno una forma "strana" in quanto sono fatti a forma di "C" o "E" essendo predisposti ad ospitare altri blocchi all'interno della loro "pancia".	Ripetere un comando 
			Ripetere in Loop 
			Condizioni (da unire con blocchi esagonali relativi ai sensori) 
Sensori		Servono per gestire l'interazione degli Sprite tra loro, con lo Stage o con l'utente. Consentono di percepire gli eventi esterni (ad esempio il clic del mouse) e interni (interazione con altri oggetti del programma). Nella sezione dei sensori si trovano blocchi di ogni forma (esagonali, ovali o a puzzle)	Stai toccando.. (fa parte dei blocchi esagonali perfetti per le condizioni) 
			Variabile (I blocchi che hanno forma ovale possono essere inseriti all'interno di ogni blocco che necessita di un "valore" per portare a termine il proprio scopo.) 

BLOCCHI	COLORE	DESCRIZIONE	ESEMPI DI COMANDI
Operatori		I blocchi operatori sono di due tipi esagonali e ovali (queste forme si trovano anche nei sensori come visto in precedenza). I blocchi esagonali restituiscono una risposta vera o falsa (visibile sullo schermo) e i blocchi ovali restituiscono sempre un valore (visibile sullo schermo).	X > Y (inserire i valori) 
			X + Y (inserire i valori) 
Variabili		Le variabili, in programmazione, sono delle porzioni di memoria che vengono "riempite" con dei dati volatili, cioè che possono cambiare durante l'esecuzione del programma.	La mia variabile (permette di creare una variabile) 
I miei blocchi		Permette di creare blocchi personalizzati	

I comandi di esempio riportati nella tabella non sono tutti i comandi disponibili per ogni blocco ma una piccola parte. Qui di seguito un esempio di programmazione che utilizza i blocchi illustrati nella tabella (Figura 49).

1. Il primo blocco giallo appartiene alla categoria "Situazioni" e crea uno specifico evento (il click sulla bandiera verde).

2. Il secondo blocco blu appartiene alla categoria "Movimento" e fa muovere lo Sprite (il gatto) di 10 passi.

3. Il terzo blocco appartiene alla categoria "Aspetto" e fa dire "Ciao" allo Sprite per 2 secondi.

4. Il quarto blocco appartiene alla categoria "Controllo" e fa attendere lo Sprite 5 secondi.

5. Il quinto blocco appartiene alla categoria "Controllo" e rappresenta una condizione. Al suo interno troviamo il blocco esagonale appartenente alla categoria "Sensori" e il blocco puzzle appartenente alla categoria "Suono". Se l'utente preme il tasto spazio sulla tastiera allora viene riprodotto il suono "Miao".

6. Il sesto blocco appartiene alla categoria "Variabili" e al suo interno troviamo il blocco ovale appartenente alla categoria "Operatori". In questo modo l'utente assegna il valore 3 (dato dalla somma $2+1$) alla "mia variabile".

7. Il settimo blocco appartiene alla categoria "Controllo" e al suo interno troviamo il blocco ovale appartenente alla categoria "Variabili". In questo modo lo Sprite attende 3 secondi (il valore assegnato in precedenza alla "mia variabile").

8. L'ottavo blocco appartiene alla categoria "Movimento" e fa ruotare lo Sprite a destra di 15 gradi.

9. Il nono blocco appartiene alla categoria "Controllo" e chiude il programma.

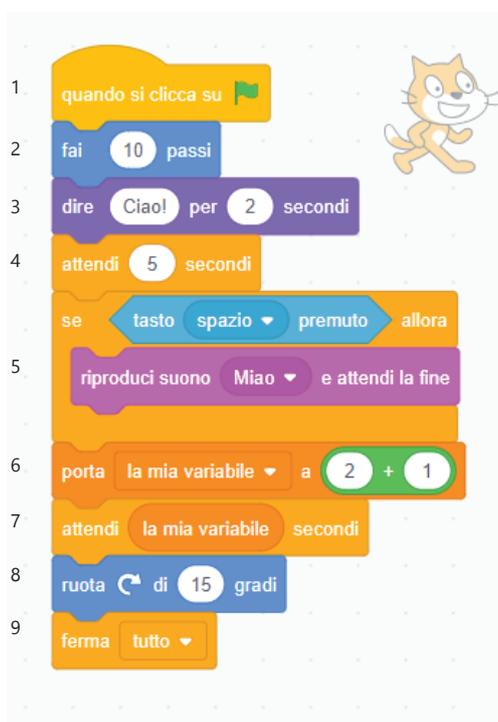


Figura 49- Esempio di programmazione in Scratch

4.3 ScratchJr

ScratchJr è un'applicazione di programmazione gratuita per bambini dai 5 ai 7 anni e nasce dalla collaborazione del DevTech Research Group della Tufts University, del Lifelong Kindergarten Group dei Media Lab del MIT e della Playful Invention Company. ScratchJr è stato lanciato per la prima volta nel 2014 come un programma gratuito scaricabile su iPad e da allora, negli anni successivi, è stato rilasciato per l'uso su diverse altre piattaforme, tra cui Tablet Android, tablet Amazon e Chromebook. Fino a febbraio 2020 è stato scaricato oltre 13 milioni di volte e viene utilizzato in 191 paesi in tutto il mondo ⁽³⁸⁾. L'obiettivo di ScratchJr è quello di fornire un linguaggio di programmazione gratuito a tutti i bambini per imparare a programmare, pensare in modi nuovi e usare la tecnologia per l'espressione personale.

Marina Umaschi Bers una delle figure impegnate nella progettazione di ScratchJr (professoressa presso il Dipartimento di Studi sull'Infanzia e Sviluppo Umano Eliot-Pearson con un Master of Science e PhD presso il MIT Media Laboratory in collaborazione con Seymour Papert) nel suo libro, "Coding as Playground: Programming and Computational Thinking in the Early Childhood Classroom" ⁽¹⁶⁾ (2020) ci spiega il processo di ricerca che ha portato alla nascita di ScratchJr ma anche di Kibo Robot (un kit robotico per l'infanzia). Dal racconto di Marina U. Bers è evidente che ScratchJr sia stato un lavoro sviluppato da un team molto grande e vario, hanno lavorato in collaborazione con i grafici in modo che l'interfaccia potesse trasmettere giocosità e le azioni programmabili potessero risultare divertenti. La cosa più difficile è stato comprendere quanto sia differente la fascia di età 5-7 anni rispetto ai bambini di 8-16 che utilizzano Scratch. Il team di ricerca ha trascorso molte ore negli asili e nelle scuole primarie per comprendere quali fossero i limiti di quella fascia di età durante l'utilizzo di Scratch. Notarono, ad esempio, sin dal principio come i bambini si sentissero persi avendo molti comandi di programmazione a disposizione così decisero di limitare la scelta a pochi e semplici comandi. Un altro cambiamento importante che apportarono rispetto a Scratch fu la direzionalità utilizzata nella programmazione con i blocchi. I bambini in quella fascia di età si stavano avvicinando per la prima volta alla scrittura da sinistra a destra per questo motivo ritennero fondamentale

mantenere questo verso anche nella programmazione (a differenza di Scratch che utilizza una programmazione dall'alto verso il basso per imitare linguaggi di programmazione più avanzati). Sulla base di questi risultati iniziarono la progettazione dei primi prototipi ScratchJr, Alpha e Beta, conducendo in ogni fase dei test sugli utenti (bambini, genitori e insegnanti).

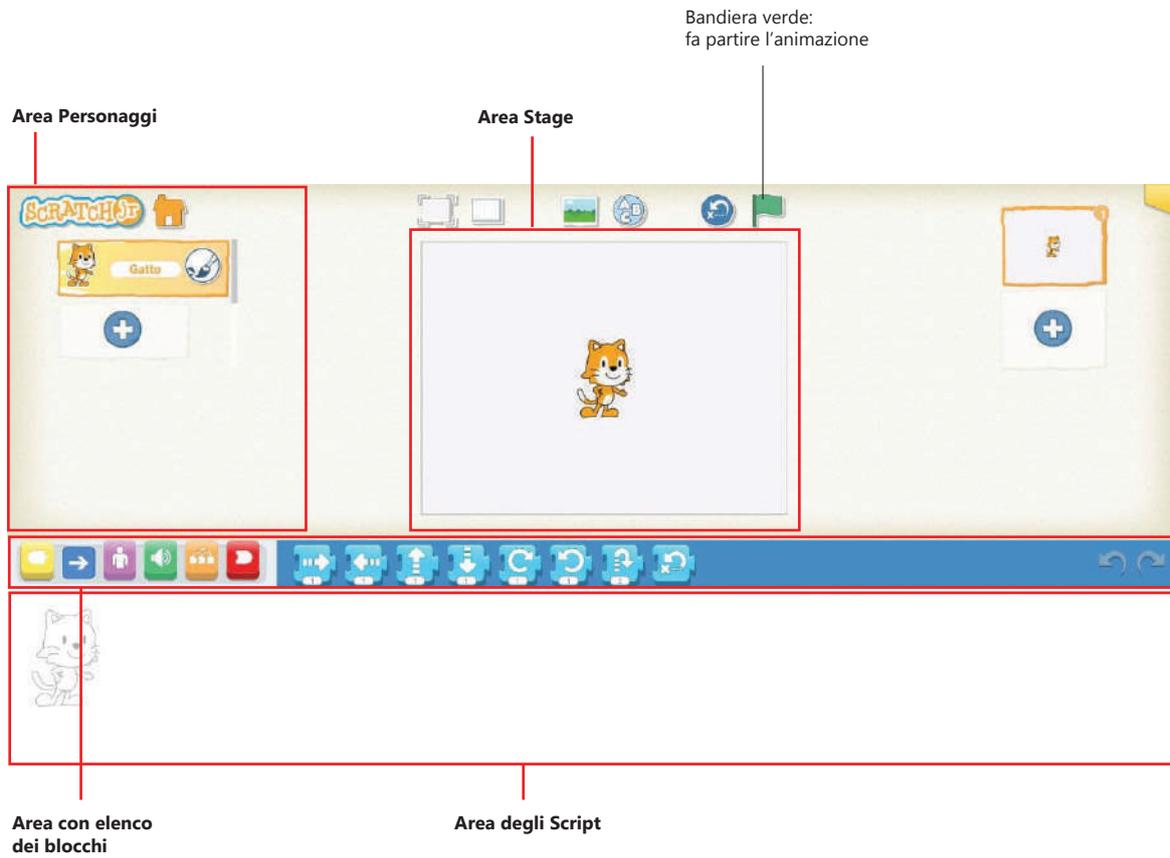
ScratchJr da gennaio 2016 a febbraio 2020, ha permesso di realizzare oltre 48,6 milioni di progetti e più di 64,8 milioni di progetti esistenti sono stati aperti di nuovo e modificati, indicando che gli utenti continuano a lavorare cercando di migliorare e gestire meglio i propri progetti personali.

La programmazione a blocchi di ScratchJr ha permesso agli insegnanti e ai genitori di avvicinare bambini molto piccoli alla programmazione dando la possibilità di esprimersi attraverso nuove modalità e rendendo l'apprendimento semplice e giocoso.

Entrando nel dettaglio ScratchJr ha un'interfaccia molto simile a Scratch con alcune differenze. L'interfaccia è suddivisa sempre in quattro aree, ciò che cambia a prima vista è la disposizione e il numero limitato di contenuti:

- **Stage:** il rettangolo bianco posto al centro dello schermo dove si svolgono le azioni dei personaggi.
- **Characters:** una finestra posta sul lato sinistro dello schermo, a differenza di Scratch non viene usata la terminologia "Sprite" ma semplicemente personaggi. Anche in questo caso nel progetto di default tra i personaggi troveremo la mascotte di Scratch che ha le sembianze di un gatto arancione.
- **Elenco dei Blocchi:** si trova al centro dello schermo sotto lo Stage ed è una riga che contiene tutti i blocchi di programmazione, suddivisi in categorie diverse.
- **Area degli Script:** un grande rettangolo posto sotto la riga che elenca i blocchi di programmazione. Questa è l'area dove i blocchi di programmazione vengono uniti per creare gli script che diranno al personaggio cosa fare.

Interfaccia ScratchJr



I blocchi utilizzati in ScratchJr, a differenza di quelli utilizzati in Scratch, sono divisi in sei categorie: **Situazioni, Movimento, Aspetto, Suono, Controllo e Blocchi di fine.** (Tabella 2)

Tabella 2- Le sei categorie di blocchi con alcuni esempi di comandi

BLOCCHI	COLORE	DESCRIZIONE	ESEMPI DI COMANDI
Situazioni		Hanno lo scopo di creare degli eventi, cioè determinare quali accadimenti debbano scatenare l'avvio di qualcosa. La forma arrotondata su un lato (in questo caso sinistro) sta ad indicare che tali blocchi devono trovarsi nella parte iniziale della sequenza dei comandi.	Avvia quando si tocca la Bandiera Verde 
			Avvia quando si tocca il personaggio 
			Avvia quando quando il personaggio viene toccato da un altro personaggio. 
Movimento		Hanno lo scopo di modificare la posizione del personaggio nello Stage.	Muovi verso l'alto 
			Muovi verso destra 
			Muovi verso sinistra 
			Ruota verso destra 
			Ruota verso sinistra 
Aspetto		Hanno lo scopo di modificare il personaggio, ad esempio: permettono di fargli dire qualcosa (appare un fumetto), ne variano la dimensione e permettono di farlo sparire con una dissolvenza.	Dire "Ciao" (appare un fumetto) 
			Ridurre la dimensione del personaggio 

BLOCCHI	COLORE	DESCRIZIONE	ESEMPI DI COMANDI
Suono		Hanno lo scopo di produrre dei suoni.	Riprodurre un suono "Pop" 
			Riprodurre un suono registrato 
Controllo		Hanno lo scopo di controllare il flusso del programma con ripetizioni, controllando la velocità, facendo delle pause o fermando il programma. Molti dei blocchi di questa categoria hanno una forma "strana" come una C capovolta, questo perchè sono predisposti ad ospitare altri blocchi all'interno della loro "pancia".	Attendere 
			Ferma 
			Imposta velocità 
			Ripeti 
Blocchi di fine		Indicano la fine dello Sript. Come i blocchi "Situazione" hanno la forma arrotondata su un lato (in questo caso destro) e sta ad indicare che tali blocchi devono trovarsi nella parte finale della sequenza dei comandi.	Fine 
			Vai alla pagina (passa alla pagina specificata del progetto) 
			Ripeti per sempre 

Come per l'analisi di Scratch del paragrafo precedente, i comandi di esempio riportati nella tabella non sono tutti i comandi disponibili per ogni blocco ma una parte.

Qui di seguito un esempio di programmazione che utilizza i blocchi illustrati nella tabella.
(Figura 50)

1. Il primo blocco giallo appartiene alla categoria "Situazioni" e crea uno specifico evento (il click sulla bandiera verde).

2. Il secondo blocco blu appartiene alla categoria "Movimento" e fa muovere il personaggio di 1 passo.

3. Il terzo blocco appartiene alla categoria "Aspetto" e fa aumentare la dimensione del personaggio di due volte.

4. Il quarto blocco appartiene alla categoria "Suono", il personaggio emette il suono "Pop".

5. Il quinto blocco appartiene alla categoria "Controllo" e rappresenta una ripetizione. Al suo interno troviamo due blocchi appartenenti alla categoria "Movimento". In questo caso la ripetizione è impostata a 4 volte, il personaggio va a destra e ruota verso destra per quattro volte.

6. Il sesto blocco appartiene alla categoria "Blocchi di fine" e chiude il programma.

Per facilitare l'insegnamento di ScratchJr ai bambini, il team di ricercatori ha sviluppato un curriculum per studenti **principianti, intermedi e avanzati**.

- Gli studenti principianti: esplorano gli algoritmi semplici con il sequenziamento lineare. Utilizzano blocchi situazioni, movimento, aspetto, suono e blocchi di fine.
- Gli studenti di livello intermedio: possono iniziare esplorare il concetto di ripetizione. Utilizzano i blocchi per principianti e in aggiunta sperimentano le funzionalità dei blocchi di controllo.
- Gli studenti avanzati: possono interagire con sequenze parallele di due o più programmi simultanei.

Questa divisione è molto interessante perchè permette di introdurre passo dopo passo i blocchi di Scratch permettendo ai bambini di selezionarli con facilità e comprenderne meglio tutte le funzioni. I bambini della scuola dell'infanzia o primaria hanno abilità differenti, i bambini di 3/4 anni hanno necessità differenti dai bambini di 5/6 anni. Questa divisione permette a tutti di apprendere nel giusto modo senza eccedere o limitare le informazioni. Inoltre i bambini più grandi potranno aiutare i bambini più piccoli lavorando in gruppo e imparando a socializzare.

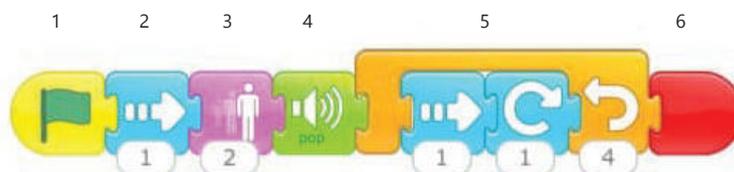
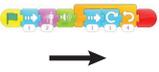


Figura 50- Esempio di programmazione in ScratchJr

Tabella 3- Differenze tra Scratch e Scratch Jr nella programmazione

	Scratch	ScratchJr
Target	8-16 anni	5-7 anni
Direzionalità della programmazione	Dall'alto verso il basso (come i linguaggi di programmazione avanzati) 	Da sinistra verso destra (come la scrittura) 
Blocchi	Tante possibilità di programmazione (10 categorie di blocchi)	Comandi limitati e semplificati (6 categorie di blocchi)
Tipologia di comunicazione	Utilizzo di testo Forme allungate e poco arrotondate 	Utilizzo di icone Forme grandi e molto arrotondate 
Distinzione delle categorie di blocchi	Colori differenti rappresentano categorie differenti di blocchi <ul style="list-style-type: none"> ● Movimento ● Aspetto ● Suono ● Situazioni ● Controllo ● Sensori ● Operatori ● Variabili ● I miei blocchi 	Colori differenti rappresentano categorie differenti di blocchi <ul style="list-style-type: none"> ● Movimento ● Aspetto ● Suono ● Situazioni ● Controllo ● Blocchi di fine

4.4 Coding Unplugged

Introdurre il Coding come insegnamento per bambini della scuola dell'infanzia non è sempre semplice e a volte non si dispone di dispositivi adatti. Per questo motivo si ricorre al Coding Unplugged, attività che utilizzano strumenti non digitali per introdurre i concetti fondamentali dell'informatica e alla logica della programmazione. L'approccio al Coding Unplugged compare nei progetti curriculari e nei libri di testo. Negli ultimi anni online si trovano varie attività da proporre a scuola anche grazie ad organizzazioni come Code.org e CodeMOOC (capitolo 1 paragrafo 1.5). Le attività possono strutturarsi in diversi modi, l'elemento comune che le caratterizza è il fatto di non usare computer, tablet né altri dispositivi digitali, in questo caso il mondo dell'analogico diventa fondamentale. Alcune attività possono essere realizzate con fogli di carta e matite colorate, altre come giochi da tavola ed altre ancora prevedono la realizzazione di attività su grandi scacchiere posizionate o disegnate a terra con il nastro adesivo di carta. Lo svantaggio del Coding Unplugged è che non permette di programmare altro se non il solo movimento. Qui di seguito alcuni esempi.

• **Esercizi di Coding Unplugged su libro di testo** (39) :

1. Nel primo esercizio si chiede al bambino di disegnare il percorso sulla griglia seguendo la riga di codice proposta. Successivamente si chiede di scrivere le coordinate nell'ordine in cui le si incontra nel percorso. (Figura 51)

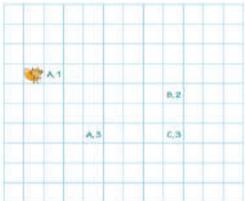
2. Nel secondo esercizio si chiede di scegliere quale riga di codice (tra le due proposte sulla destra) è stata utilizzata per creare il percorso sulla sinistra. (Figura 52)

FURBY È UNA VOLPE CORAGGIOSA, MA ANCHE A LEI CAPITA DI AVERE PAURA.

1 SCOPRI DI CHE COSA HA PAURA: SEGUI LE ISTRUZIONI.

- Disegna il percorso e scrivi le coordinate nell'ordine in cui le incontri.

↑ ↑ ↑ ↑ → ↑ ↑ ↑ ↑ ↘ ↑ ↑ ↑ ↑ ↙ ↑ ↑



• Di che cosa ha paura Furby? Scrivi le lettere che trovi nelle caselle con le coordinate trovate.

3	U	F	I
2	S	O	L
1	B	D	N
	A	B	C

Figura 51- Esercizio 1

2 OSSERVA IL PERCORSO DI FURBY: INDICA LA SEQUENZA CORRETTA.



Figura 52- Esercizio 2

- **Attività strutturata di Coding Unplugged svolta dai bambini della scuola dell'infanzia "Emanuele Serrilli" di San Marco in Lamis (FG):**

1. Lettura della storia (in questo caso "Storia dell'ape che non poteva più volare")
2. Inserimento dei personaggi e ostacoli nella griglia (Ape, alveare e sassi) (Figura 53)
3. Inizio dell'attività di coding (utilizzo delle carte: avanti, destra e sinistra) (Figura 54-56)
4. Movimento del personaggio sulla griglia (a volte i punti 3-4 vengono svolti in contemporanea) (Figura 55)



Figura 53- I bambini inseriscono i personaggi e gli ostacoli nella griglia definendo un punto di inizio e un punto di fine



Figura 54- I bambini iniziano l'attività di coding creando una linea di codice con le carte (avanti, destra e sinistra). A volte hanno la necessità di posizionare la carta sul personaggio per capirne la direzione



Figura 55- I bambini muovono il personaggio sulla griglia in base alle indicazioni del codice



Figura 56- Il codice può essere visualizzato anche come percorso

- **CodyRoby:**

un gioco per la programmazione Unplugged. Le istruzioni sono carte da gioco che contengono semplici simboli associati a tre azioni elementari: vai avanti, girati a sinistra, girati a destra (Figura 57). Ogni giocatore veste i panni di Cody e usa le carte per dare istruzioni a Roby, rappresentato da una pedina da muovere sulla scacchiera. CodyRoby è stato ideato da CodeMOOC (40) nel novembre del 2014, per offrire a tutti la possibilità di partecipare alla European Robotics Week anche all'ultimo momento e senza alcun costo. Lo starter kit di CodyRoby può essere scaricato, stampato e ritagliato per comporre il mazzo di carte, la scacchiera, le pedine e la scatola in cui tutto può essere riposto. Si può giocare su diversi campi di gioco: scacchiera in carta, scacchiera rappresentata da piastrelle o su un tappeto a incastro (Figura 58).

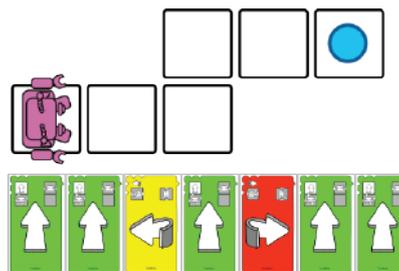


Figura 57- CodyRoby



Figura 58- Campi da gioco dove è possibile giocare a CodyRoby.

Esempio di gioco su tavolo con scacchiera in carta: **il Duello**

Le due squadre piazzano le proprie pedine in due caselle agli angoli opposti della scacchiera. Inoltre ogni squadra può collocare un blocco dove vuole per rendere inaccessibile una casella. Le 40 carte formano un mazzo che deve essere mescolato e posto sul tavolo tra i giocatori, con le carte rivolte verso il basso.

Ad ogni turno una squadra prende 5 carte dal mazzo senza mostrarle all'altra squadra. Se la squadra ha già carte in mano all'inizio del turno (perché non utilizzate nel turno precedente) prende dal mazzo solo le carte necessarie ad arrivare a 5. La squadra valuta le 5 carte a disposizione e ne usa una o più per muovere il proprio robot nella scacchiera (Figura 59). Le mosse si fanno calando ad una ad una le carte che la squadra intende utilizzare in questo turno. La squadra decide quante carte usare, ma deve usarne almeno una. Vince il duello la squadra che riesce a portare il proprio robot su una casella occupata dal robot dell'altra squadra (Figura 60).



Figura 59- Il giocatore utilizza le carte necessarie per muovere il proprio robot sulla scacchiera.

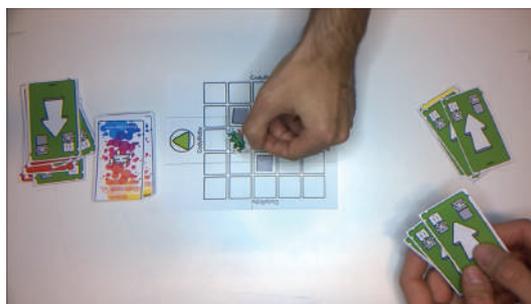


Figura 60- Il giocatore ha vinto il duello portando il proprio robot su una casella occupata dal robot avversario.

STATO DELL'ARTE

5.1 Kit robotici educativi in commercio programmati in modo tangibile

La programmazione tangibile è un sistema che permette di imparare tramite la sperimentazione e l'esperienza diretta con l'ambiente e il mondo circostante. Questo sistema è una buona via di mezzo tra il Coding classico su dispositivi come pc e tablet e il Coding Unplugged. Permette di programmare azioni di vario genere: Movimento, Suono, Aspetto ecc. (a differenza del Coding Unplugged che permette il solo Movimento) senza sottoporre i bambini alla visione di uno schermo (a differenza del Coding classico con applicazioni come Scratch). Le regole per quanto riguarda lo "screen time" (ovvero l'esposizione a uno schermo) contenute nelle nuove linee guida dell'Oms ⁽⁴¹⁾ (Organizzazione Mondiale della Sanità) dedicate al benessere dei bambini sotto i 5 anni di età ci dicono che mentre i bambini di età inferiore all'anno non dovrebbero avere alcun accesso agli schermi ("no screen time"), per i bambini tra i 2 e i 4 anni il "sedentary screen time" dovrebbe essere al massimo di 1 ora al giorno. Queste indicazioni sono in linea con quelle pubblicate nel 2016 dall'American Academy of Pediatrics⁽⁴²⁾: nessuno schermo consentito (tranne che per le video chat) per bambini al di sotto dei 18 mesi; "solo programmazione di elevata qualità" tra i 18 e i 24 mesi, con la presenza costante del genitore; infine, tra i 2 e i 5 anni, solo 1 ora al giorno di programmi "approvati". Oltre a non essere sottoposti ad uno schermo, con la programmazione tangibile, i bambini possono apprendere attraverso un'interazione diretta con il codice, questo rende l'apprendimento del linguaggio e della metodologia di programmazione più diretto, meno astratto, e maggiormente adatto ai bambini perché vicino al loro modo naturale di conoscere il mondo attraverso il gioco e l'esperienza sensoriale.

5.2 Kibo

DESCRIZIONE SINTETICA

KIBO Robot (Figura 61-62-63) è un kit robotico per l'apprendimento creato da KinderLab Robotics, dopo anni di ricerca di Marina Bers e del suo team. Il robot programmabile è progettato per bambini di 4-7 anni e promuove l'apprendimento del codice e il pensiero computazionale attraverso il gioco e la sperimentazione. I bambini programmano con blocchi di legno per creare sequenze e comprendere il processo di progettazione. KIBO può avere aspetti e comportamenti diversi, ogni volta che lo si programma. Con KIBO, i bambini di 4-7 anni creano, progettano, decorano e danno vita al proprio robot. (43)



Figura 61- Kibo



Figura 62- Kibo e altri elementi del kit

Azienda: KinderLab

Anno: 2015

Prezzo: KIBO 10 Kit - \$220,00
KIBO 15 Kit - \$335,00
KIBO 18 Kit - \$470,00
KIBO 21 Kit - \$590,00

Materiale: plastica e legno

Target: Bambini (3-6 anni)

Scenari d'uso: casa e scuola



Batterie: Sì

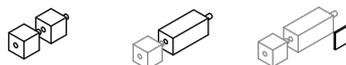
Compatibilità con altri elementi di costruzione: Sì



Possibile utilizzo di app: No

Connessione Wireless: No

Codifica: Blocchi in legno da unire tra loro e tessere da applicare sui blocchi.



Decodifica: Su ogni blocco c'è un codice a barre che viene letto da un sistema di identificazione ottica posizionato sulla parte frontale del robot



Tabella 4- Componenti Kibo

ID	COMPONENTI	KIBO 10	KIBO 15	KIBO 18	KIBO 21
1	Robot	1	1	1	1
2	Ruote	2	2	2	2
3	Blocchi di programmazione	10	15	18	21
4	Tessere parametri	4	4	12	12
5	Motori	2	2	3	3
6	Lampadina	/	1	1	1
7	Altoparlante/registratore	/	/	/	1
8	Sensore suono	/	1	1	1
9	Sensore intensità della luce	/	/	1	1
10	Sensore distanza	/	/	1	1
11	Piattaforma per arte	/	1	2	2
12	Moduli di estensione per Lego	4	5	5	5
13	Moduli di espressione	/	/	/	3



Figura 63- Esempio Kit Kibo 21

Tabella 5- Programmazione Kibo

PRESTAZIONI	COMANDI	QUANTITÀ	CODIFICA	DECODIFICA
Movimento	Avanti	1		
	Indietro	1		
	Svolta a destra	1		
	Svolta a sinistra	1		
	Tremare	1		
	Rotazione	1		
Suono	Melodia	1		
	Beep	1		
Suono/registrazione	Registrazione triangolo	1		
	Registrazione cerchio	1		
	Registrazione quadrato	1		
Colore luce	Luce rossa	1		
	Luce bianca	1		
	Luce blu	1		
Rilevare l'intensità della luce (da usare con i blocchi condizione)	Luce	1		
	Buio	1		
Rilevare un ostacolo (da usare con i blocchi condizione)	Vicino	1		
	Lontano	1		
Rilevare il suono (battito delle mani)	Attendere il battito delle mani	1		

PRESTAZIONI	COMANDI	QUANTITÀ	CODIFICA	DECODIFICA
Condizione	If	1		
	End If	1		
Ripetizione	Ripetizione	1		
	fine ripetizione	1		
PARAMETRI (Ripetizione)	2	1		
	3	1		
	4	1		
	Infinito	1		
ALTRO	Inizio	1		
	Fine	1		

Legenda



Attuatori



Sensori



Comandi di controllo



Codifica: blocchi in legno da unire tra loro



Codifica: tessera da applicare sui blocchi

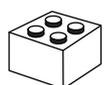
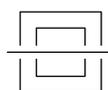
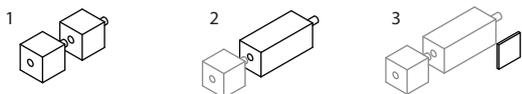


Codifica: blocchi in legno di forma allungata da unire tra loro



Decodifica: sistema di identificazione ottica

FUNZIONAMENTO



PROGRAMMAZIONE TANGIBILE CON BLOCCHI DA UNIRE TRA LORO:

I bambini creano una sequenza di istruzioni utilizzando i blocchi di legno e le tessere

1. Blocchi codifica semplici

2. Blocchi condizione (leggermente più lunghi dei blocchi semplici, presentano uno spazio bianco dove andrà posizionata la tessere parametro)

3. Tessere parametri da posizionare sui blocchi condizione

DECODIFICA CON SISTEMA DI IDENTIFICAZIONE OTTICA:

KIBO legge i codici a barre con uno scanner incorporato.



COMPATIBILITÀ CON ALTRI ELEMENTI DI COSTRUZIONE:

I bambini possono personalizzare Kibo utilizzando elementi esterni al kit come i mattoncini Lego o modelli in carta.

PRO

- Le azioni di Kibo possono essere personalizzate dal bambino.
- I bambini possono montarlo e smontarlo, questo è un esercizio utile per i bambini della scuola dell'infanzia.
- La programmazione a blocchi è intuitiva, i bambini possono toccare i blocchi in legno ed unirli con facilità.

CONTRO

- Per personalizzarne l'aspetto del robot i bambini devono utilizzare elementi esterni al kit.
- La scansione del codice a barre potrebbe non essere sempre facile da attuare per un bambino.
- La programmazione con i blocchi in legno, seppur intuitiva, potrebbe ostacolare le modifiche rapide al codice.

5.3 Mochi

DESCRIZIONE SINTETICA

L'obiettivo di Mochi (Figura 64-65) è promuovere le capacità creative e cognitive dei bambini. Mochi è stato originariamente sviluppato per favorire l'alfabetizzazione della prima infanzia e il pensiero creativo/computazionale nei bambini di età compresa tra 3 e 6 anni. L'esperienza è progettata in modo che i bambini di 3 anni possano impegnarsi in un gioco creativo e autodiretto. Mochi sfrutta l'amore dei bambini per le storie per introdurre concetti di programmazione astratti in un modo divertente e senza barriere. Dai blocchi di programmazione tattili alla narrazione uditiva, dal feedback visivo al gioco funzionale, tutto in Mochi è progettato per lasciare che la curiosità faccia da apripista per l'apprendimento. (44)



Figura 64- Mochi

Azienda: Learn with Mochi

Anno: 2019

Prezzo: Mochi Robotics Kit (1 libro) \$299,00

Mochi Robotics Kit (4 libri) \$329,00

Mochi Robotics Kit (6 libri) \$399,00

Mochi Robotics Kit (12 libri)\$610,00

Materiale: cotone (Peluche) e compensato di betulla (Robot, blocchi e board)

Target: Bambini (3-6 anni)

Scenari d'uso: casa



Batterie: Si

Compatibilità con altri elementi di costruzione: Si



Possibile utilizzo di app: No

Connessione Wireless: Si (Tra robot e board)



Codifica: Blocchi in legno da inserire nella board di programmazione



Decodifica: Sensori magnetici a effetto hall situati nei blocchi e lettori situati nella board



Tabella 6- Componenti Mochi

ID	COMPONENTI	Mochi (1 libro)	Mochi (4 libri)	Mochi (6 libri)	Mochi (12 libri)
1	Robot	1	1	1	1
2	Blocchi di programmazione	39	39	39	39
3	Libri di storie	1	4	6	12
4	Board	1	1	1	1
5	Mappa	1	1	1	1
6	Orsetto	1	1	1	1
7	Kit componenti in legno	1	1	1	1



Figura 65- Esempio Mochi Robotics Kit (1 libro)

Tabella 7- Programmazione Mochi

PRESTAZIONI	COMANDI	QUANTITÀ	CODIFICA	DECODIFICA
Movimento	Avanti	6		
	Indietro	3		
	Svolta a destra	3		
	Svolta a sinistra	3		
Suono	ninna nanna	2		
	melodia 1	2		
	melodia 2	2		
	melodia 3	2		
	melodia 4	2		
	domande	1		
	storia	1		
Funzione	Richiama la funzione	4		
Negazione	fa l'opposto	4		
LOOP	ricomincia da 1	2		

Legenda



Attuatori



Comandi di controllo

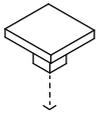


Codifica: blocchi da inserire nella board



Decodifica: utilizzo di sensori

FUNZIONAMENTO



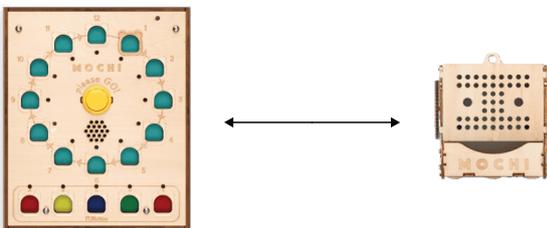
PROGRAMMAZIONE TANGIBILE CON BLOCCHI DA INSERIRE NELLA BOARD:

Il bambino programma Mochi inserendo i blocchetti nella board.



DECODIFICA CON L'UTILIZZO DI SENSORI :

La board legge i blocchi tramite sensori magnetici a effetto hall. Terminata la codifica, i bambini premono il pulsante centrale sulla board per avviare l'esecuzione del programma.



CONNESSIONE WIRELESS:

La board invia i comandi al robot tramite una connessione wireless.



COMPATIBILITÀ CON ALTRI ELEMENTI DI COSTRUZIONE:

I bambini possono personalizzare Mochi utilizzando elementi esterni al kit come i mattoncini Lego.

PRO

- Le azioni di Mochi possono essere personalizzate dal bambino.
- Utilizzo di storie.
- La programmazione a blocchi è intuitiva, i bambini possono toccare i blocchi di legno ed inserirli con facilità.

CONTRO

- I componenti forniti nel Kit base permettono una personalizzazione minima dell'aspetto del robot. Per personalizzarne l'aspetto i bambini devono utilizzare elementi esterni al kit .
- I comandi sono limitati poichè non sono presenti sensori.

5.4 Code a Pillar

DESCRIZIONE SINTETICA

Code a Pillar (Figura 66-67) è un giocattolo a forma di bruco che viene fornito con una testa motorizzata e otto moduli separati ognuno dei quali dedicato a una specifica azione (come andare avanti, girare, riprodurre un brano musicale ecc). Il bimbo aggancia i vari moduli uno all'altro realizzando di fatto una catena di istruzioni che il bruco esegue in sequenza una volta azionato il tasto start. Ad esempio, a seconda della sequenza che andiamo a impostare potremo far suonare una canzone dopo un certo tragitto, andare dritto o farlo girare a sinistra e a destra. Code-a-Pillar fa parte della serie di giocattoli Think & Learn, pensata per sviluppare le capacità di problem solving nei bambini in età prescolare.⁽⁴⁵⁾



Figura 66- Code a Pillar (2016)

Azienda: Fisher Price

Anno: 2016

Prezzo: € 89,99

Materiale: Plastica

Target: Bambini (3-6 anni)

Scenari d'uso: casa



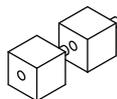
Batterie: Sì

Compatibilità con altri elementi di costruzione: No

Possibile utilizzo di app: No

Connessione Wireless: No

Codifica: Blocchi in plastica da unire tra loro



Decodifica: Ogni blocco è dotato di una porta USB da inserire nel blocco successivo

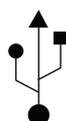


Tabella 8- Componenti Code a Pillar

ID	COMPONENTI	QUANTITÀ
1	Testa del Robot	1
2	Blocchi di programmazione	8
3	Disco inizio	1
4	Disco fine	1

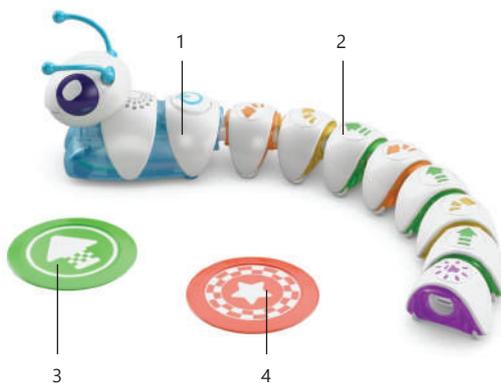


Figura 67- Code a Pillar Kit

Tabella 9- Programmazione Code a Pillar

PRESTAZIONI	COMANDI	QUANTITÀ	CODIFICA	DECODIFICA
Movimento	Avanti	3		
	Svolta a destra	2		
	Svolta a sinistra	2		
Suono	Melodia	1		

Legenda



Attuatori

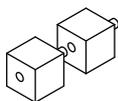


Codifica: blocchi da unire tra loro



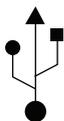
Decodifica: comunicazione USB

FUNZIONAMENTO



PROGRAMMAZIONE TANGIBILE CON BLOCCHI DA UNIRE TRA LORO:

I bambini creano una sequenza di istruzioni utilizzando blocchi in plastica collegabili tra loro



DECODIFICA CON COMUNICAZIONE USB:

I blocchi comunicano tra loro e con la testa del robot tramite porte USB presenti su ogni blocco



PRO

- Le azioni del robot possono essere personalizzate dal bambino.
- I bambini possono montarlo e smontarlo, questo è un esercizio utile per i bambini della scuola dell'infanzia.
- La programmazione a blocchi è intuitiva

CONTRO

- Il robot non può essere personalizzato
- I comandi sono limitati poiché non sono presenti sensori
- Per aggiungere comandi è necessario acquistare componenti aggiuntivi

5.5 Bee Bot

DESCRIZIONE SINTETICA

Bee Bot (Figura 68) è un robot a forma di ape in grado di memorizzare una serie di comandi base e muoversi di conseguenza su un percorso. Bee Bot ha tutti i comandi sul dorso: avanti, indietro, svolta a sinistra e a destra, pausa, avvia e cancella il programma. Utilizzando questi comandi base e combinandoli fra loro, è possibile impostare programmi che contano fino a 40 passaggi.

Gli obiettivi educativi sono: comprendere le basi del coding e i comandi sequenziali, sviluppare il ragionamento logico e le abilità motorie, stimolare la capacità di categorizzazione e problem solving.

Inoltre è un ottimo sussidio per attività di storytelling ed è adeguato per attività didattiche in piccoli gruppi. (46)



Figura 68- Bee Bot

Tabella 10- Componenti Bee Bot

ID	COMPONENTI	QUANTITÀ
1	Robot	1

Azienda: TTS

Anno: 2009

Prezzo: € 89

Materiale: Plastica

Target: Bambini (3-6 anni)

Scenari d'uso: casa e scuola



Batterie: Si

Compatibilità con altri elementi di costruzione: No

Possibile utilizzo di app: Si

Connessione Wireless: Non necessaria- possibilità di connettere la BeeBot con pc e tablet per programmarla tramite App.

Codifica e Decodifica: Pulsanti sul dorso del robot



Tabella 11- Programmazione Bee Bot

PRESTAZIONI	COMANDI	QUANTITÀ	CODIFICA	DECODIFICA
Movimento	Avanti	1		
	Svolta a destra	1		
	Svolta a sinistra	1		
	indietro	1		
ALTRO	Pausa 1 secondo	1		
	Avvia programma	1		
	Cancella programma	1		

Legenda



Attuatori



Comandi di controllo



Codifica e Decodifica: pulsanti

FUNZIONAMENTO



CODIFICA E DECODIFICA CON PULSANTI:

I bambini programmano il robot premendo i pulsanti sul dorso.



PRO

- Le azioni di Bee Bot possono essere personalizzate dal bambino.
- La programmazione è semplice

CONTRO

- Le funzioni possibili riguardano il solo cambio di direzione
- Il robot non può essere personalizzato nel suo aspetto

5.6 Coding & Robotics

DESCRIZIONE SINTETICA

Coding & Robotics (Figura 69-70) è un kit robotico. Il robot principale viene programmato semplicemente disponendo una sequenza di carte codice. Mentre il robot cammina sulle carte codice, uno scanner ottico ad ossido sul fondo del robot legge le carte una per una e carica il programma. Quindi, posizionando il robot su una griglia composta da carte mappa, il robot eseguirà il programma. È possibile programmare il robot per muoverlo in diverse direzioni, accendere il LED, riprodurre suoni e rispondere a diverse schede funzione. Possiede un servomotore integrato che consente di realizzare semplici robot con bracci o altre parti mobili che rispondono secondo le istruzioni del programma. (47)



Figura 69- Coding & Robotics

Azienda: Gigo

Anno: 2018

Prezzo: € 199,90

Materiale: Plastica

Target: Bambini (4+)

Scenari d'uso: casa



Batterie: Sì

Compatibilità con altri elementi di costruzione: No

Possibile utilizzo di app: No

Connessione Wireless: No

Codifica: carte codice posizionate una dopo l'altra



Decodifica: lettura carte codice tramite un sistema di identificazione ottica posizionato sulla parte inferiore del robot



Tabella 12- Componenti Coding & Robotics

ID	COMPONENTI	QUANTITÀ
1	Robot (con servomotori e luci)	1
2	Carte di programmazione	61
3	Pezzi per costruzione	49
4	Schede per mappa	38

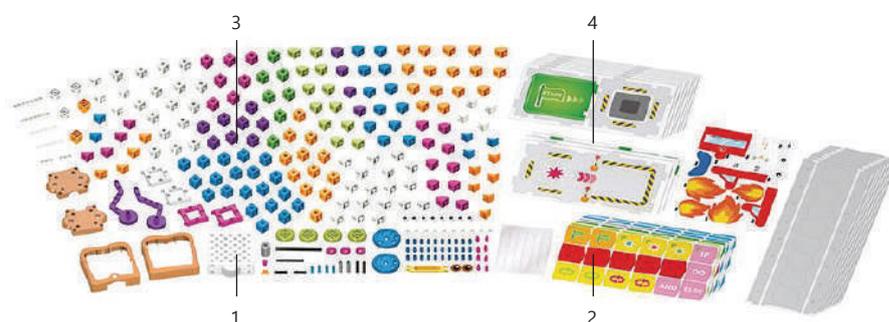


Figura 70- Coding & Robotics Kit

Tabella 13- Programmazione Coding & Robotics

PRESTAZIONI	COMANDI	QUANTITÀ	CODIFICA	DECODIFICA
Movimento	Avanti	24		
	Indietro	24		
	Svolta a destra	18		
	Svolta a sinistra	18		
	Gira le ruote in senso orario per 1 sec (usare solo con le funzioni)	5		
	Gira le ruote in senso antiorario per 1 sec (usare solo con le funzioni)	5		
	Pausa movimento	4		
	Pausa movimento ruote per 1 sec (usare solo con le funzioni)	4		

PRESTAZIONI	COMANDI	QUANTITÀ	CODIFICA	DECODIFICA
Luce	Blu	2		
	Viola	2		
	Rosa	2		
	Arancione	2		
	Giallo	2		
	Verde	2		
	Arcobaleno	2		
	Discoteca	2		
	Luce allarme	2		
	Stella cadente	2		
	Scintillante	2		
	Lucciola	2		
	Lampeggio lento	2		
	Lampeggio medio	2		
	Lampeggio veloce	2		
	Lampeggio accelerato	2		
Lampeggio rallentato	2			
Suono	Esclamazione: Hi	2		
	Esclamazione: Ahh	2		
	Esclamazione: Huh?	2		
	Esclamazione: Aargh	2		

PRESTAZIONI	COMANDI	QUANTITÀ	CODIFICA	DECODIFICA
Suono	Suono: Tifoseria	2		
	Suono: Fabbrica	2		
	Suono: Vigili del fuoco	2		
	Suono: Polizia	2		
	Verso animale: Topo	2		
	Verso animale: Pinguino	2		
EVENTO (da usare nelle condizioni, se il robot incontra il simbolo dell'evento durante il percorso allora deve essere soddisfatta l'esecuzione della funzione)	Evento 1	1		
	Evento 2	1		
	Evento 3	1		
	Evento 4	1		
PARAMETRI (ripetere un comando o un Loop)	1	2		
	2	2		
	3	2		
	4	2		
	5	2		
	6	2		
	7	2		
	8	2		
	9	2		
LOOP	Loop rosso	4		
	Loop verde	4		

PRESTAZIONI	COMANDI	QUANTITÀ	CODIFICA	DECODIFICA
FUNZIONE	Inizio funzione rossa	2		
	Inizio funzione verde	2		
	Inizio funzione blu	2		
CONDIZIONE	If	2		
	Do	2		
	Else	2		
	And	1		
	Or	1		
ALTRO	Inizio programmazione	4		
	Fine programmazione	10		

Legenda



Attuatori



Comandi di controllo

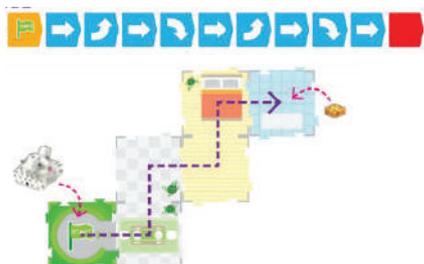
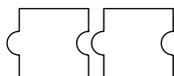


Codifica: carte codice da posizionare una dopo l'altra



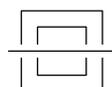
Decodifica: sistema di identificazione ottica

FUNZIONAMENTO



PROGRAMMAZIONE TANGIBILE CON CARTE CODICE POSIZIONATE UNA DOPO L'ALTRA:

Il bambino programma il robot disponendo una sequenza di carte codice.



DECODIFICA CON SISTEMA DI IDENTIFICAZIONE OTTICA:

Uno scanner ottico posto sul fondo del robot legge le carte codice una per una e salva il programma che verrà eseguito successivamente sul percorso.



PRO

- Le azioni del robot possono essere personalizzate dal bambino.
- I bambini possono montarlo e smontarlo, questo è un esercizio utile per i bambini della scuola dell'infanzia.
- Le carte sono intuitive e presentano diverse funzioni.

CONTRO

- Non è possibile collegare sensori e attuatori
- Il robot deve muoversi per leggere il codice.

5.7 Cubetto

DESCRIZIONE SINTETICA

Cubetto ^(Figura 71-72) è un robot programmabile per bambini dotato di ruote. Si controlla via wireless tramite una board in cui possono essere inseriti 16 blocchi di forme e colori diversi, a ciascuno dei quali corrisponde un comando. Il comportamento di Cubetto dipende dall'ordine in cui i blocchi vengono disposti sulla board. Ogni tassello colorato rappresenta un pezzo di codice: mettendoli in fila, uno dietro l'altro, si ottiene una sequenza di istruzioni che viene immediatamente trasmessa ed eseguita dal robot. Con Cubetto i bambini imparano la programmazione a blocchi senza bisogno di ricorrere a un computer, semplicemente usando le loro mani. Nella confezione si possono trovare mappe e storie educative da utilizzare per rendere l'esperienza di gioco ancora più stimolante. ⁽⁴⁸⁾



Figura 71- Cubetto (Robot, Board e blocchi per la programmazione)

Azienda: Primo Toys

Anno: 2017

Prezzo: € 178,18

Materiale: Plastica e legno

Target: Bambini (3-6)

Scenari d'uso: casa



Batterie: Sì

Compatibilità con altri elementi di costruzione: No

Possibile utilizzo di app: No

Connessione Wireless: Sì



Codifica: Blocchi in plastica da inserire nella board di programmazione



Decodifica: sensori magnetici a effetto hall situati nella board



Tabella 14- Componenti Cubetto

ID	COMPONENTI	QUANTITÀ
1	Robot	1
2	Blocchi di programmazione	16
3	Board	1
4	Mappa	1
5	Libro	1

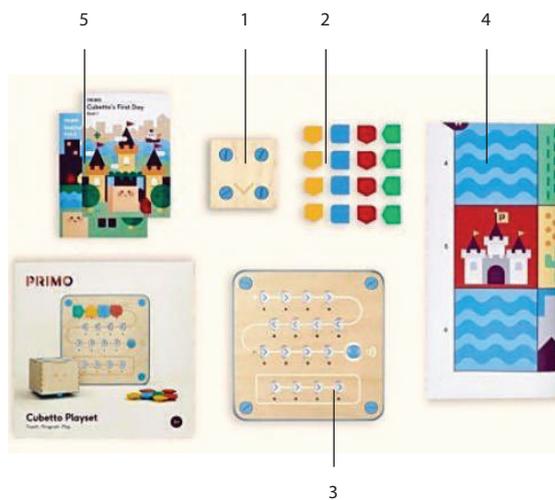


Figura 72- Componenti Cubetto Kit

Tabella 15- Programmazione Cubetto

PRESTAZIONI	COMANDI	QUANTITÀ	CODIFICA	DECODIFICA
Movimento	Avanti	4		
	Svolta a destra	4		
	Svolta a sinistra	4		
	indietro	4		
	Movimenti random	4		
Negazione	Fa l'opposto del blocco dopo	4		
Funzione	Funzione	4		

Legenda



Attuatori



Comandi di controllo

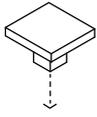


Codifica: blocchi da inserire nella board



Decodifica: utilizzo di sensori

FUNZIONAMENTO



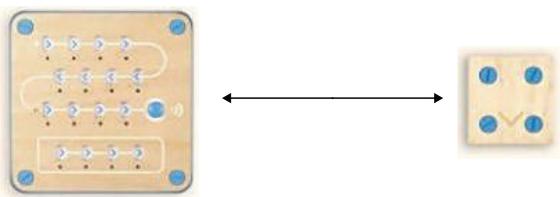
PROGRAMMAZIONE TANGIBILE CON BLOCCHI DA INSERIRE NELLA BOARD:

Il bambino programma Cubetto inserendo i blocchetti nella board.



DECODIFICA CON L'UTILIZZO DI SENSORI:

Terminata la codifica, i bambini, premono il pulsante blu sulla board per avviare l'esecuzione del programma. La board legge i blocchi tramite sensori magnetici a effetto hall.



CONNESSIONE WIRELESS:

La board invia i comandi al robot tramite una connessione wireless.

PRO

- Le missioni danno un obiettivo al bambino e grazie alle mappe tematiche riescono a sviluppare le capacità di storytelling.
- La programmazione è intuitiva

CONTRO

- Il robot non può essere personalizzato nel suo aspetto
- Le uniche funzioni sono quelle del movimento.

5.8 mTiny

DESCRIZIONE SINTETICA

mTiny (Figura 73-74) è un robot educativo per la prima infanzia. mTiny si muove su mappe tematiche coinvolgenti, viene programmato attraverso carte codice posizionate una dopo l'altra e la decodifica avviene grazie al passaggio di una penna speciale su ogni carta. I bambini ottengono un feedback immediato sul loro lavoro, sviluppando abilità di problem solving e incoraggiando al tempo stesso l'iniziativa e la creatività attraverso il coinvolgimento multidisciplinare su matematica, musica, arte e molto altro grazie ai tappeti a tema. (49)



Figura 73- mTiny (Robot, carte codice e penna per la programmazione)

Azienda: Makeblock

Anno: 2019

Prezzo: € 145,18

Materiale: Plastica

Target: Bambini (4+)

Scenari d'uso: casa



Batterie: Sì

Compatibilità con altri elementi di costruzione: No

Possibile utilizzo di app: No

Connessione Wireless: Sì



Codifica: Carte codice posizionate una dopo l'altra



Decodifica: lettura carte codice tramite un sistema di identificazione ottica posizionato sulla penna



Tabella 16- Componenti mTiny

ID	COMPONENTI	QUANTITÀ
1	Robot	1
2	Carte di programmazione	36
3	Penna	1
4	Pezzi mappa	24
5	Libro	1
6	Carte speciali	3
7	Cavo USB	1
8	Maschere	3
9	Asta bandiera	1
10	Bandiera	8

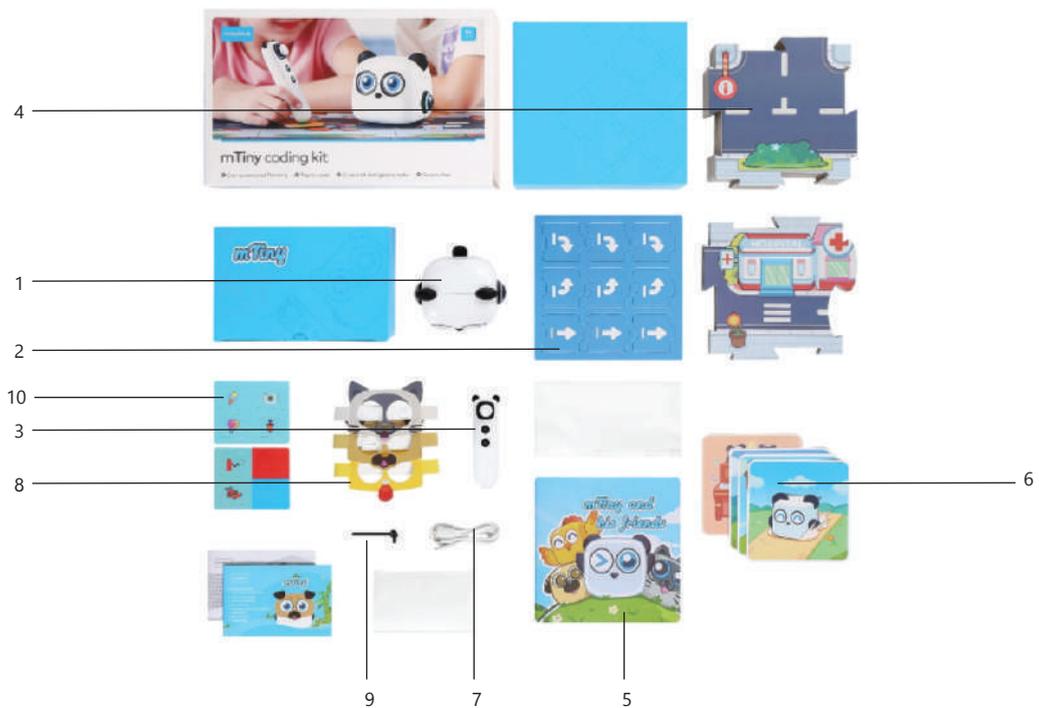


Figura 74- mTiny Kit

Tabella 17- Programmazione mTiny

PRESTAZIONI	COMANDI	QUANTITÀ	CODIFICA	DECODIFICA
Movimento	Avanti	4		
	Svolta a destra	4		
	Svolta a sinistra	4		
Espressioni facciali	Espressione 1	1		
	Espressione 2	1		
	Espressione 3	1		
	Espressione 4	1		
	Espressione 5	1		
	Espressione 6	1		
	Espressione 7	1		
Parametri (ripetizione)	2	2		
	3	2		
	4	2		
	5	2		
LOOP	Loop	1		
	Loop x 2 volte	1		
	Loop x 3 volte	1		
	Loop x 4 volte	1		
	Loop x 5 volte	1		
Funzione	(2		
)	2		

PRESTAZIONI	COMANDI	QUANTITÀ	CODIFICA	DECODIFICA
ALTRO	Inizio programma	1		
	Fine programma	1		

Legenda



Attuatori



Comandi di controllo

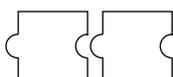


Codifica: carte codice da posizionare una dopo l'altra



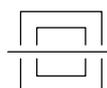
Decodifica: sistema di identificazione ottica

FUNZIONAMENTO



PROGRAMMAZIONE TANGIBILE CON CARTE CODICE POSIZIONATE UNA DOPO L'ALTRA:

Il bambino programma il robot disponendo una sequenza di carte codice.



DECODIFICA CON SISTEMA DI IDENTIFICAZIONE OTTICA:

Uno scanner ottico posto sulla penna legge le carte codice una per una e salva il programma che verrà eseguito successivamente sul percorso.



CONNESSIONE WIRELESS:

La penna invia i comandi al robot tramite una connessione wireless.

PRO

- Le azioni del robot possono essere personalizzate dal bambino.
- Le tessere di programmazione sono intuitive e presentano diverse funzioni.
- Le tessere per la creazione del percorso sono interattive.

CONTRO

- Non è possibile collegare sensori e attuatori
- Il robot non può essere personalizzato nel suo aspetto

5.9 Sphero Indi

DESCRIZIONE SINTETICA

Sphero Indi (Figura 75-76) è un robot di apprendimento indi progettato per introdurre gli studenti ai fondamenti del pensiero computazionale, di STEAM e dei principi dell'informatica, incoraggiando l'apprendimento fantasioso e aperto basato sul gioco. Gli studenti possono costruire labirinti personalizzati e risolvere enigmi. Indi offre un'esperienza di programmazione utilizzando un sensore di colore integrato nel robot.

Il sensore colore e le tessere colorate permettono ai bambini di costruire percorsi sempre nuovi e stimolare la loro creatività. (50)



Figura 75- Sphero Indi

Azienda: Sphero

Anno: 2021

Prezzo: Kit casa \$99,99 / Kit scuola \$124,99

Materiale: Plastica

Target: Bambini (4+)

Scenari d'uso: casa e scuola



Batterie: Si

Compatibilità con altri elementi di costruzione: No

Possibile utilizzo di app: Si

Connessione Wireless: No

Codifica: Carte codice posizionate una dopo l'altra



Decodifica: lettura carte codice tramite il sensore colore posizionato sulla parte inferiore del Robot



Tabella 18- Componenti Sphero Indi

ID	COMPONENTI	kit casa	kit scuola
1	Robot	1	1
2	Carte di programmazione	30 (materiale: carta)	20 (materiale: gomma)
3	Sticker	2	2
4	cavo USB	1	1
5	Carte sfida	/	15
6	Valigetta per studenti	/	1



Figura 76- Componenti Sphero Indi Kit

Tabella 19- Programmazione Sphero Indi

PRESTAZIONI	COMANDI	QUANTITÀ	CODIFICA	DECODIFICA
Movimento	Avanti (verde)	3		
	Rallenta (giallo)	2		
	Fermo (rosso)	2		
	Svolta a sinistra 90° (rosa)	3		
	Svolta a destra 90° (blu)	3		
	Svolta a destra 45° (verde acqua)	3		
	Svolta a sinistra 45° (arancione)	3		
	Rotazioni (viola)	1		

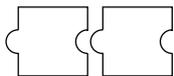
Legenda

Attuatori

Codifica: carte codice da posizionare una dopo l'altra

Decodifica: utilizzo di sensori

FUNZIONAMENTO



PROGRAMMAZIONE TANGIBILE CON CARTE CODICE POSIZIONATE UNA DOPO L'ALTRA:

Il bambino programma il robot disponendo una sequenza di carte colorate.



DECODIFICA CON L'UTILIZZO DI SENSORI:

Il sensore colore posto sulla parte inferiore del robot legge le carte colorate una per una ed esegue i comandi simultaneamente.



PRO

- Le azioni del robot possono essere personalizzate dal bambino.
- Le tessere di programmazione sono intuitive.
- Il programma viene eseguito in modo simultaneo.

CONTRO

- Non è possibile collegare sensori e attuatori
- Il robot non può essere personalizzato nel suo aspetto

5.10 MatataBot

DESCRIZIONE SINTETICA

MatataBot (Figura 77-78) è un robot che, oltre a muoversi nello spazio, può anche disegnare e cantare.

La misura del singolo passo è di 10 cm e, oltre a ruotare di 90° permette anche altri angoli di movimento.

Alla board di programmazione è unita la Torre di comando. Questa è simile a un faro ed è il centro di controllo dell'attività, in quanto può dire al MatataBot cosa fare grazie al lettore per il riconoscimento delle immagini. (51)



Figura 77- MatataBot (Robot e board di programmazione)

Azienda: Matatalab

Anno: 2020

Prezzo: \$149,99

Materiale: Plastica

Target: Bambini (4+)

Scenari d'uso: casa e scuola



Batterie: Si

Compatibilità con altri elementi di costruzione: Si



Possibile utilizzo di app: No

Connessione Wireless: Si



Codifica: Blocchi da inserire nella board di programmazione



Decodifica: lettura dei blocchi tramite un lettore di riconoscimento immagine posizionato sulla torre



Tabella 20- Componenti MatataBot

ID	COMPONENTI	QUANTITÀ
1	Robot	1
2	Torre di comando	1
3	Board	1
4	Blocchi di programmazione	46
5	Ostacoli	8
6	Bandiere	3
7	Mappa	2
8	Cavo USB	1



Figura 78- Componenti MatataBot Kit

Tabella 21- Programmazione MatataBot

PRESTAZIONI	COMANDI	QUANTITÀ	CODIFICA	DECODIFICA
Movimento	Avanti	4		
	Indietro	4		
	Svolta a sinistra	4		
	Svolta a destra	4		
	Danza	1		
	Casuale	1		
Suono	nota 1	1		
	nota 2	1		
	nota 3	1		
	nota 4	1		
	nota 5	1		
	nota 6	1		
	nota 7	1		
	nota 8	1		
	melodia	1		
LOOP	Inizio Loop	2		
	Fine Loop	2		
Parametri (gradi)	45°	1		
	60°	1		
	120°	1		
	135°	1		

PRESTAZIONI	COMANDI	QUANTITÀ	CODIFICA	DECODIFICA
Parametri (gradi)	144°	1		
Parametri (ripetizione)	2	2		
	3	2		
	4	2		
	5	2		
	Ripetizione casuale	2		

Legenda



Attuatori



Comandi di controllo

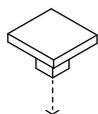


Codifica: blocchi da inserire nella board



Decodifica: sistema di identificazione ottica

FUNZIONAMENTO



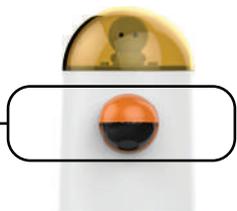
PROGRAMMAZIONE TANGIBILE CON BLOCCHI DA INSERIRE NELLA BOARD:

I bambini programmano il robot inserendo i blocchetti nella board.



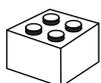
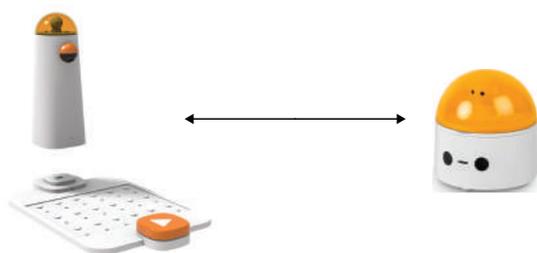
DECODIFICA CON SISTEMA DI IDENTIFICAZIONE OTTICA:

Terminata la codifica, i bambini, premono il pulsante sulla board per avviare l'esecuzione del programma. I blocchi vengono letti dalla telecamera posta sulla torre di comando.



CONNESSIONE WIRELESS:

La torre invia i comandi al robot tramite una connessione wireless.



COMPATIBILITÀ CON ALTRI ELEMENTI DI COSTRUZIONE:

I bambini possono personalizzare il robot utilizzando elementi esterni al kit come i mattoncini Lego.



PRO

- Le azioni di MatataBot possono essere personalizzate dal bambino.
- Utilizzo di storie.
- La programmazione a blocchi è intuitiva, i bambini possono toccare i blocchi ed inserirli con facilità.

CONTRO

- I componenti forniti nel Kit base permettono una personalizzazione minima dell'aspetto del robot. Per personalizzarne l'aspetto i bambini devono utilizzare elementi esterni al kit .

5.11 Qobo

DESCRIZIONE SINTETICA

Qobo (Figura 79-80) è una lumaca programmabile. I bambini costruiscono il percorso cercando di fare arrivare la lumaca al diamante finale. Può cantare, ballare, muoversi ed illuminarsi con luci diverse programmandola con le carte puzzle. Una carta può avere combinazione infinite e sviluppare la creatività del bambino. Il kit include una mappa di gioco e un libro di gioco con 16 lezioni basate su diversi livelli, dal più semplice al più avanzato. ⁽⁵²⁾



Figura 79- Qobo

Azienda: Roboblog

Anno: 2019

Prezzo: € 89,99

Materiale: Plastica

Target: Bambini (3-6 anni)

Scenari d'uso: casa



Batterie: Si

Compatibilità con altri elementi di costruzione: Si



Possibile utilizzo di app: Si

Connessione Wireless: Non necessaria- possibilità di connettere la BeeBot con pc e tablet per programmarla tramite App.

Codifica: Carte codice posizionate una dopo l'altra



Decodifica: lettura carte codice tramite un sistema di identificazione ottica posizionato sulla parte inferiore del robot



Tabella 22- Componenti Qobo

ID	COMPONENTI	QUANTITÀ
1	Robot	1
2	Carte di programmazione	1
3	Libro attività	1
4	Tessere mappa	46
5	Cavo USB	8



Figura 80- Componenti Qobo Kit

Tabella 23- Programmazione Qobo

PRESTAZIONI	COMANDI	QUANTITÀ	CODIFICA	DECODIFICA
Movimento	Avanti	1		
	Svolta a sinistra	2		
	Svolta a destra	2		
	banana o mela (in base alla missione richiesta in precedenza) banana: gira a sinistra mela : gira a destra	1		
	Pausa (aspetta di essere spostato)	1		
Luci	Rosso	1		
	Arancione	1		
	Giallo	1		
	Verde	1		
	Blu	1		
	Celeste	1		
	Viola	1		
Suono	Polizia	1		
	Treno	1		
	Melodia 1	1		
	Melodia 2	1		
Touch	Aspetta il tocco sulle labbra del robot	1		

PRESTAZIONI	COMANDI	QUANTITÀ	CODIFICA	DECODIFICA
LOOP	Loop x 3 volte	1		
	Loop x 6 volte	1		
	Fine Loop	1		
Funzione	Trova banana	1		
	Trova mela	1		
Funzione (random)	Trova banana o mela	1		
ALTRO	Ostacolo	1		
	Inizio	2		
	Fine	2		

Legenda



Attuatori



Sensori



Comandi di controllo

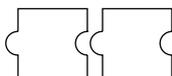


Codifica: carte codice da posizionare una dopo l'altra



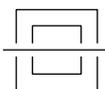
Decodifica: sistema di identificazione ottica

FUNZIONAMENTO



PROGRAMMAZIONE TANGIBILE CON CARTE CODICE POSIZIONATE UNA DOPO L'ALTRA:

Il bambino programma il robot disponendo una sequenza di carte codice.



DECODIFICA CON SISTEMA DI IDENTIFICAZIONE OTTICA:

Il lettore ottico posto sulla parte inferiore del robot legge le carte colorate una per una ed esegue i comandi o simultaneamente (nello stesso momento in cui legge la carta, il robot esegue il comando) o successivamente (spostando il robot manualmente su ogni carta, il lettore ottico legge prima tutte le carte e dopo ne esegue i comandi)



COMPATIBILITÀ CON ALTRI ELEMENTI DI COSTRUZIONE:

I bambini possono personalizzare il robot utilizzando elementi esterni al kit come i mattoncini Lego.

PRO

- Le azioni del robot possono essere personalizzate dal bambino.
- Le tessere del puzzle per la programmazione sono intuitive e presentano diverse funzioni.
- Il robot esegue i comandi sia simultaneamente sia successivamente

CONTRO

- Non è possibile collegare sensori e attuatori

5.12 Coko

DESCRIZIONE SINTETICA

Coko (Figura 81-82) è un cocodrillo programmabile, può dormire, nuotare e spaventarsi e i bambini possono programmarlo con 15 tessere da porre sulla sua schiena. Coko, inoltre, è in grado di riconoscere il "cibo" magnetico e reagisce quando gli si avvicina. Questo robot è pensato per bambini dai 3 anni in su e aiuterà il bambino a comprendere le relazioni causa-effetto della programmazione. (53)



Figura 81- Coko

Azienda: Clementoni

Anno: 2018

Prezzo: € 44,90

Materiale: Plastica

Target: Bambini (3-6 anni)

Scenari d'uso: casa



Batterie: Sì

Compatibilità con altri elementi di costruzione: No

Possibile utilizzo di app: No

Connessione Wireless: No

Codifica: Schede magnetiche da inserire nella parte superiore del robot



Decodifica: sensori magnetici a effetto hall situati nel robot



Tabella 24- Componenti Coko

ID	COMPONENTI	QUANTITÀ
1	Robot	1
2	Tessere di programmazione	15
3	Componente "Cibo"	1

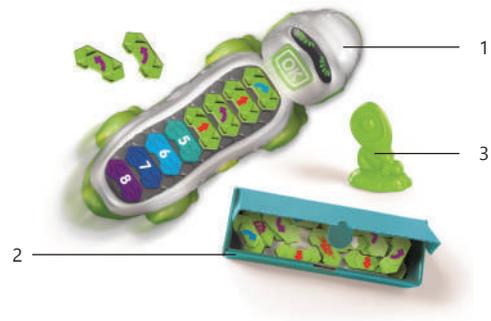


Figura 82- Componenti Coko Kit

Tabella 25- Programmazione Coko

PRESTAZIONI	COMANDI	QUANTITÀ	CODIFICA	DECODIFICA
Movimento	Avanti	6		
	Indietro	3		
	Svolta a destra	3		
	Svolta a sinistra	3		
	Rotazione	2		
Suono	Dormire	2		
	Pericolo	2		
	Nuotare	2		

Legenda



Attuatori

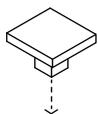


Codifica: blocchi da inserire nella board



Decodifica: utilizzo di sensori

FUNZIONAMENTO



PROGRAMMAZIONE TANGIBILE CON BLOCCHI DA INSERIRE NELLA BOARD:

Il bambino programma Coko inserendo i blocchetti sul dorso del robot.



DECODIFICA CON L'UTILIZZO DI SENSORI:

Terminata la codifica i bambini premono il pulsante "OK" sulla testa del robot per avviare l'esecuzione del programma. Il robot legge i blocchi tramite sensori magnetici a effetto hall.



PRO

- Le azioni di Coko possono essere personalizzate dal bambino.
- La programmazione è semplice.
- Interagisce con elementi esterni.

CONTRO

- Il robot non può essere personalizzato nel suo aspetto.

5.13 MojoBot

DESCRIZIONE SINTETICA

Mojobot (Figura 83-84-85) è un robot che rende facile per i bambini imparare i principi fondamentali della programmazione e della robotica. Mojobot è progettato pensando sia al divertimento che al valore educativo, è ricco di sensori, azioni e una piattaforma di codifica fisica estensibile. Il kit robotico Mojobot contiene: un robot Mojobot, una console di comando, una console di estensione, tag di codifica, mappa puzzle e dischetti che Mojobot può raccogliere e portare in giro. Per aiutare i bambini ad essere coinvolti nell'apprendimento della codifica è stato creato il libro Mojobot Adventures Stories e un gioco di carte Missione. (54)



Figura 83- MojoBot

Azienda: Project Lab

Anno: 2019

Prezzo: € 535,58

Materiale: Plastica

Target: Bambini (4+)

Scenari d'uso: casa



Batterie: Sì

Compatibilità con altri elementi di costruzione: No

Possibile utilizzo di app: No

Connessione Wireless: Sì



Codifica: Blocchi da inserire nella board di programmazione



Decodifica: sensori situati nella board di programmazione



Tabella 26- Componenti MojoBot

ID	COMPONENTI	MOJOBOT BASIC KIT	MOJOBOT EXTENSION KIT
1	Robot	1	/
2	Blocchi di programmazione	42	40
3	Board di programmazione	1	1
4	Mappa	1	/
5	Dischetto	14	/
6	Carte missione	33	22
7	Libro	1	1
8	Personaggi pedine MojoBot	4	/
9	Box ostacoli	/	2

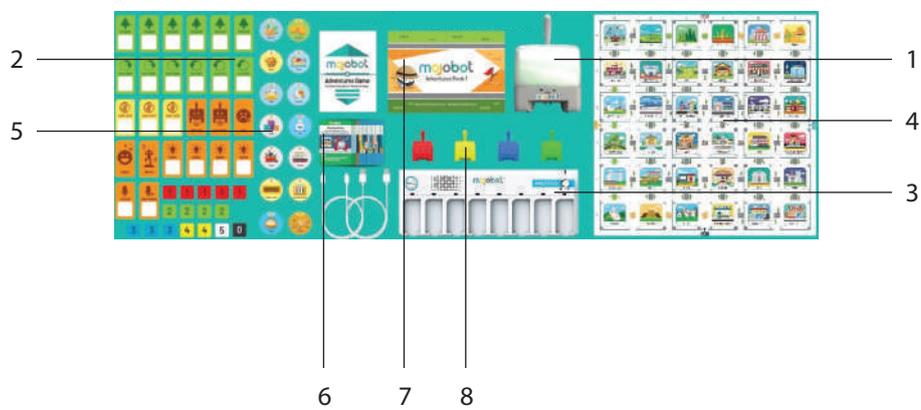


Figura 84- Componenti MojoBot basic Kit



Figura 85- Componenti MojoBot Extension Kit

Tabella 27- Programmazione MojoBot (basic set + estension set)

PRESTAZIONI	COMANDI	QUANTITÀ	CODIFICA	DECODIFICA
Movimento	Avanti	10		(((((
	Svolta a destra	5		(((((
	Svolta a sinistra	5		(((((
	Raccogli il dischetto	2		(((((
	Getta il dischetto	2		(((((
	Danza	1		(((((
Espressioni facciali (emozioni)	Felicità	1		(((((
	Tristezza	1		(((((
Suono/registrazione	Registrazione	2		(((((
	Riproduci registrazione	2		(((((
Luce	Accendere una luce	4		(((((
Rilevare l'intensità della luce	Luce	1		(((((
	Buio	1		(((((
Rilevare un ostacolo	Vicino	1		(((((
	Lontano	1		(((((
Rilevare un suono	Alto	1		(((((
	Basso	1		(((((
Rilevare la presenza del dischetto (a terra o all'interno del robot)	Prende il dischetto	1		(((((
	Lascia il dischetto	1		(((((

PRESTAZIONI	COMANDI	QUANTITÀ	CODIFICA	DECODIFICA
Attesa	Aspetta fino a	4		
Loop	Ripeti fino a	2		
	Fine ripetizione	2		
Condizione	If	2		
	End if	2		
Parametri (colore luce, ripetizioni e numero registrazione da riprodurre)	1 - rosso	6		
	2 - verde	5		
	3 - blu	4		
	4 - giallo	3		
	5 - bianco	2		
	0 - nero	1		
	infinito - grigio	1		
Funzione	Set A (programma una funzione)	4		
	Do A (Richiama la funzione in un'altra linea di codice)	1		

Legenda



Attuatori



Sensori



Comandi di controllo

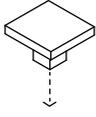


Codifica: blocchi da inserire nella board



Decodifica: utilizzo di sensori

FUNZIONAMENTO



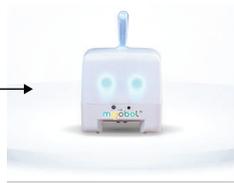
PROGRAMMAZIONE TANGIBILE CON BLOCCHI DA INSERIRE NELLA BOARD:

Il bambino programma MojoBot inserendo i blocchetti nella board. Può collegare un'estensione della board per programmare nuove linee di codice.



DECODIFICA CON L'UTILIZZO DI SENSORI :

La board legge i blocchi tramite sensori. Terminata la codifica i bambini premono il pulsante in alto a sinistra sulla board per avviare l'esecuzione del programma.



CONNESSIONE WIRELESS:

La board invia i comandi al robot tramite una connessione wireless.

PRO

- La programmazione a blocchi è intuitiva.
- Presenta una varietà di comandi possibili.

CONTRO

- Il robot non può essere personalizzato nel suo aspetto.
- Per programmare linee di codice è necessario utilizzare un'altra board.

5.14 Tabelle

Tabella 28- Tabella riassuntiva delle prestazioni di tutti i Kit dello Stato dell'arte

Prestazioni	Kibo		Mochi		Code a Pillar		Bee Bot		Coding & Robotics	
	Codifica	Decodifica	Codifica	Decodifica	Codifica	Decodifica	Codifica	Decodifica	Codifica	Decodifica
Movimento										
Suono										
Luce										
Registrazione										
Espressioni facciali										
Rilevare il suono										
Rilevare la luce										
Rilevare ostacoli										
Rilevare il tocco										
Negazione										
Parametri (ripetizione)										
Parametri (gradi)										
Condizioni										
Loop										
Funzione										
Cancela programma										
Inizio programma										
Fine programma										

	Cubetto		mTiny		Sphero Indi		MatataBot		Qobo	
Prestazioni	Codifica	Decodifica	Codifica	Decodifica	Codifica	Decodifica	Codifica	Decodifica	Codifica	Decodifica
Movimento										
Suono										
Luce										
Registrazione										
Espressioni facciali										
Rilevare il suono										
Rilevare la luce										
Rilevare ostacoli										
Rilevare il tocco										
Negazione										
Parametri (ripetizione)										
Parametri (gradi)										
Condizioni										
Loop										
Funzione										
Cancella programma										
Inizio programma										
Fine programma										

Prestazioni	Coko		MojoBot	
	Codifica	Decodifica	Codifica	Decodifica
Movimento				
Suono				
Luce				
Registrazione				
Espressioni facciali				
Rilevare il suono				
Rilevare la luce				
Rilevare ostacoli				
Rilevare il tocco				
Negazione				
Parametri (ripetizione)				
Parametri (gradi)				
Condizioni				
Loop				
Funzione				
Cancela programma				
Inizio programma				
Fine programma				

Legenda

-  Sensori
-  Attuatori
-  Comandi di controllo
-  Codifica: carte codice da posizionare una dopo l'altra
-  Codifica: blocchi da inserire nella board
-  Codifica: blocchi da unire tra loro
-  Codifica e Decodifica: pulsanti
-  Decodifica: utilizzo di sensori
-  Decodifica: comunicazione USB
-  Decodifica: sistema di identificazione ottica

Tabella 29- Tabella riassuntiva modalità di codifica e decodifica di tutti i Kit dello Stato dell'arte

	CODIFICA				DECODIFICA			
								
Kibo	●				●			
Mochi		●				●		
Code a Pillar	●						●	
Bee Bot			●				●	
Coding & Robotics				●	●			
Cubetto		●				●		
mTiny				●	●			
Sphero Indi				●		●		
MatataBot		●			●			
Qobo				●	●			
Coko		●				●		
MojoBot		●				●		

Legenda



Codifica



Decodifica



Codifica: carte codice da posizionare una dopo l'altra



Decodifica: utilizzo di sensori



Codifica: blocchi da inserire nella board



Decodifica: comunicazione USB



Codifica: blocchi da unire tra loro



Decodifica: sistema di identificazione ottica



Codifica e Decodifica: pulsanti

Tabella 30- Valutazione critica (Codifica e Decodifica)

Codifica	Decodifica	Esempio	Facilità di utilizzo singolo	Facilità di utilizzo in gruppo	Facilità di modifica della programmazione in caso di errore	Facilità di produzione
			● ● ● ● ● ●	● ● ● ● ○ ○	● ● ● ● ● ●	● ● ● ● ○ ○
			● ● ● ● ○ ○	● ● ● ● ● ●	● ● ○ ○ ○ ○	● ● ● ● ● ●
			● ● ● ● ○ ○	● ● ● ● ● ●	● ● ○ ○ ○ ○	● ● ● ● ● ○
			● ● ● ● ● ●	● ○ ○ ○ ○ ○	● ● ● ● ● ●	● ● ● ● ○ ○
			● ● ● ● ● ○	● ● ● ● ○ ○	● ● ● ● ● ○	● ○ ○ ○ ○ ○
			● ● ● ● ● ●	● ● ● ● ● ●	● ● ● ● ● ●	● ● ● ● ● ●
			● ● ● ● ● ●	● ● ● ● ○ ○	● ● ● ● ● ●	● ● ● ● ○ ○

Legenda Codifica e Decodifica

 Codifica: carte codice da posizionare una dopo l'altra

 Decodifica: utilizzo di sensori

 Codifica: blocchi da inserire nella board

 Decodifica: comunicazione USB

 Codifica: blocchi da unire tra loro

 Decodifica: sistema di identificazione ottica

 Codifica e Decodifica: pulsanti

Legenda valutazioni

- 0: molto difficile
- 1: difficile
- 2: abbastanza difficile
- 3: poco difficile
- 4: facile
- 5: molto facile

Facilità di utilizzo singolo: facilità con cui i bambini singolarmente riescono a codificare e decodificare.

Facilità di utilizzo in gruppo: facilità con cui la modalità di programmazione si presta ad essere utilizzata da più bambini contemporaneamente.

Facilità di modifica della programmazione caso di errore: facilità con cui è possibile modificare il codice e successivamente decodificarlo.

Facilità di produzione: facilità nel produrre gli elementi di codifica e decodifica.

Tabella 31- Valutazione critica (Codifica e Decodifica)

Codifica	Decodifica	Esempio	Costi di produzione	Precisione nella decodifica	Comandi supportati	Spazio utilizzato per la codifica
			● ● ● ● ○ ○	● ● ● ● ● ●	● ● ● ● ● ○	● ● ○ ○ ○ ○
			● ○ ○ ○ ○ ○	● ● ○ ○ ○ ○	● ● ● ● ● ●	● ● ● ● ● ○
			● ● ● ● ○ ○	● ● ○ ○ ○ ○	● ● ● ● ● ●	● ● ● ● ● ○
			● ● ● ○ ○ ○	● ● ● ● ● ●	● ○ ○ ○ ○ ○	● ○ ○ ○ ○ ○
			● ● ● ● ● ●	● ● ● ● ● ●	● ● ● ● ○ ○	● ● ● ● ● ○
			● ○ ○ ○ ○ ○	● ● ● ● ● ●	● ● ● ● ● ●	● ● ● ● ● ○
			● ● ● ● ○ ○	● ● ● ● ○ ○	● ● ● ● ● ○	● ● ● ○ ○ ○

Legenda Codifica e Decodifica

 Codifica: carte codice da posizionare una dopo l'altra

 Decodifica: utilizzo di sensori

 Codifica: blocchi da inserire nella board

 Decodifica: comunicazione USB

 Codifica: blocchi da unire tra loro

 Decodifica: sistema di identificazione ottica

 Codifica e Decodifica: pulsanti

Legenda valutazioni

- 0: nulla
- 1: poco
- 2: mediamente
- 3: abbastanza
- 4: molto
- 5: moltissimo

Costi di produzione: i costi dei componenti per la realizzazione.

Precisione nella decodifica: precisione nella lettura del codice.

Comandi supportati: possibilità di inserire nuovi comandi o ampliare la scelta di quelli esistenti.

Spazio utilizzato per la codifica: quantità di spazio all'interno di una stanza dedicato alla codifica. Il punteggio massimo (5) si riferisce all'utilizzo di uno spazio illimitato come potrebbe essere quello della realtà virtuale.

Tabella 32- Caratteristiche generali di tutti i Kit dello Stato dell'arte

	SCENARI D'USO	COMPATIBILITÀ CON LEGO	BATTERIA	POSSIBILE UTILIZZO DI APP	CONNESSIONE WIRELESS	ASSEMBLAGGIO-PROGRAMMAZIONE
Kibo	 	●	●			A-P
Mochi		●	●			A-P
Code a Pillar			●			A-P
Bee Bot	 		●	●	● (tra robot e pc/tablet)	P
Coding & Robotics			●			A-P
Cubetto			●		● (tra robot e board)	P
mTiny			●		● (tra robot e penna)	P
Sphero Indi	 		●	●	● (tra robot e pc/tablet)	P
MatataBot	 	●	●		● (tra robot e board)	A-P
Qobo		●	●	●	● (tra robot e pc/tablet)	A-P
Coko			●			P
MojoBot			●		● (tra robot e board)	P

Legenda



A: assemblaggio

P: programmazione

Tabella 33- Tabella riassuntiva dei componenti di tutti i Kit dello Stato dell'arte

		Kibo	Mochi	Code a Pillar	Bee Bot	Coding & Robotics	Cubetto
Sensori	Sensore suono	●					
	Sensore luce	●					
	Sensore prossimità	●					
Attuatori	Altoparlante/registratore	●	●	●	●	●	
	Luce led	●		●		●	
	Servomotore	●				●	
Elementi robot	Robot base	●	●	●	●	●	●
	Ruote	●					
Elementi programmazione	Blocchi di programmazione	●	●	●	●	●	●
	Elemento di lettura del codice						●
Elementi personalizzazione	Elementi di personalizzazione	●	●			●	
	Moduli estensione Lego	●	●				
Altro	Libro storie/ attività		●				●
	Mappa		●			●	●
	Cavo USB						
	Altro (ostacoli, bandiere ecc)			●			

Legenda

● elemento del kit ● esempio elementi del kit (supporto per Lego come elemento in aggiunta) ● esempio elemento presente nel robot (supporto per Lego presente sulla scocca del robot)

● elemento presente nel robot

		mTiny	Sphero Indi	MatataBot	Qobo	Coko	MojoBot
Sensori	Sensore suono						●
	Sensore luce						●
	Sensore prossimità						●
Attuatori	Altoparlante/registratore	●	●	●	●	●	●
	Luce led				●		●
	Servomotore						
Elementi robot	Robot base	●	●	●	●	●	●
	Ruote						
Elementi programmazione	Blocchi di programmazione	●	●	●	●	●	●
	Elemento di lettura del codice	●		●			●
Elementi personalizzazione	Elementi di personalizzazione	●	●				
	Moduli estensione Lego			●	●		
Altro	Libro storie/ attività	●			●		●
	Mappa	●			●		●
	Cavo USB	●	●	●	●		●
	Altro (ostacoli, bandiere ecc)	●	●	●		●	●

Legenda

● elemento del kit

● elemento presente nel robot

Tabella 34- Valutazione critica delle caratteristiche di tutti i kit dello Stato dell'arte

	Espressività	Spazio di utilizzo	Varietà dei comandi	Aiuto da parte di genitori/insegnanti
Kibo	● ● ● ● ○	● ● ● ● ○	● ● ● ● ●	● ● ● ● ●
Mochi	● ● ● ○ ○	● ● ○ ○ ○	● ● ○ ○ ○	● ● ○ ○ ○
Code a Pillar	● ○ ○ ○ ○	● ● ● ● ○	● ○ ○ ○ ○	● ● ○ ○ ○
Bee Bot	○ ○ ○ ○ ○	● ○ ○ ○ ○	● ○ ○ ○ ○	● ○ ○ ○ ○
Coding & Robotics	● ● ● ● ●	● ● ● ● ○	● ● ● ● ○	● ● ● ● ●
Cubetto	○ ○ ○ ○ ○	● ● ○ ○ ○	● ● ○ ○ ○	● ● ○ ○ ○
mTiny	● ● ○ ○ ○	● ● ● ● ○	● ● ● ● ○	● ● ● ● ●
Sphero Indi	○ ○ ○ ○ ○	● ● ● ● ○	● ○ ○ ○ ○	● ○ ○ ○ ○
MatataBot	● ● ● ○ ○	● ● ○ ○ ○	● ● ● ● ○	● ● ● ● ●
Qobo	● ● ● ● ○	● ● ● ● ○	● ● ● ● ○	● ● ● ● ●
Coko	○ ○ ○ ○ ○	● ● ● ● ○	● ○ ○ ○ ○	● ○ ○ ○ ○
MojoBot	○ ○ ○ ○ ○	● ● ○ ○ ○	● ● ● ● ○	● ● ● ● ○

Legenda valutazioni

- 0: per niente
- 1: poco
- 2: mediamente
- 3: abbastanza
- 4: molto
- 5: moltissimo

Espressività: le possibili combinazioni di costruzione o personalizzazione del kit.

Spazio di utilizzo: la quantità di spazio in una stanza dedicato alla programmazione ed al movimento del robot. Il punteggio massimo (5) si riferisce all'utilizzo di uno spazio illimitato come potrebbe essere quello della realtà virtuale. La valutazione si basa anche sulla presenza nel Kit di mappe che delimitano l'area di gioco.

Varietà dei comandi: la quantità di comandi differenti da utilizzare nella fase di programmazione.

Aiuto da parte di genitori/insegnanti: l'aiuto necessario per comprendere ed utilizzare il kit robotico.

Progetto

6.1 Combo

Combo è un kit robotico educativo per l'apprendimento del Coding nella scuola dell'infanzia e si rivolge a bambini dai 3 ai 6 anni. Come visto in precedenza, il Coding negli ultimi anni sta diventando fondamentale per introdurre il pensiero computazionale nelle scuole ed è bene proporlo come insegnamento sin dalla prima infanzia. Esistono diversi prodotti adibiti a questa funzione e dall'analisi dello stato dell'arte è emerso quali siano le soluzioni più adeguate e quali aspetti andrebbero migliorati. Con questo progetto di tesi si è cercato di realizzare un kit educativo programmabile che sia facile da utilizzare dando modo di apprendere concetti complessi della programmazione in modo semplice.

Gli obiettivi principali sono:

- **Permettere ai bambini di esprimersi in modo creativo tramite il Coding.**
- **Migliorare le capacità di problem solving e storytelling.**
- **Aiutare i bambini a sviluppare competenze trasversali come la collaborazione e la comunicazione.**
- **Aiutare i bambini a comprendere il ruolo positivo dell'errore.**
- **Essere un mediatore didattico per l'insegnante.**

Questo kit inoltre è stato pensato per essere utilizzato da più bambini contemporaneamente dando modo a tutti di avere un ruolo nell'attività. Il kit si presenta con due elementi principali: un robot personalizzabile nel suo aspetto con sensori e attuatori applicabili (applicando i componenti elettronici i bambini riescono a comprendere meglio alcuni meccanismi della robotica) e una board adibita alla programmazione tangibile con blocchi per i comandi da inserire al suo interno.

6.2 Board e blocchi per la programmazione

La modalità di programmazione utilizzata nel kit è di tipo tangibile ovvero non utilizza uno schermo per la codifica. Esistono diverse tipologie di programmazione tangibile (analizzate nello stato dell'arte, vedere capitolo 5) che vanno ad utilizzare tecnologie molto differenti tra loro. Tra tutte, quella che permette di: programmare e modificare il codice in modo semplice, ottenere maggiore precisione nella decodifica ed avere diverse possibilità di comandi, è l'utilizzo di sensori. Utilizzare sensori implica la realizzazione di un elemento di codifica ed uno di decodifica (che permetta la lettura dei singoli comandi). Per questo progetto sono stati realizzati blocchi da inserire in una board. I blocchi fungono da comandi per la codifica e la board funge da lettore per la decodifica.

Per la progettazione della programmazione sono stati presi in considerazione anche gli studi fatti per l'applicazione "ScratchJr" (vedere capitolo 4, paragrafo 4.3), questo permetterebbe una continuità educativa con il successivo utilizzo di "Scratch" che ad oggi è una delle applicazioni più utilizzate dai bambini della scuola elementare, media e superiore. La programmazione tangibile proposta in questo progetto rappresenta un'attività introduttiva di Coding, spesso queste attività vengono svolte in modo Unplugged (vedere capitolo 4, paragrafo 4.4) con la possibilità di programmare il solo movimento. Per coinvolgere il bambino in modo creativo è bene dare la possibilità di programmare diverse funzioni esplorando quelli che sono i concetti comuni della programmazione. Proporre funzioni limitate sarebbe come proporre un corso di scrittura insegnando solo la grammatica senza offrire la possibilità di creare e scrivere le proprie storie. Per questo motivo nel kit sono stati inseriti blocchi che rappresentano funzioni differenti e, seppur limitate per non confondere i bambini, permettono la programmazione di azioni diverse tra loro e sperimentabili in modo creativo.

Tabella 35- Tabella programmazione Combo

PRESTAZIONI	COMANDI	QUANTITÀ	CODIFICA	DECODIFICA
Movimento	Spostamento in avanti di 45 centimetri	6		
	Svolta a destra di 90°	3		
	Svolta a sinistra di 90°	3		
	Rotazione di un componente di 360°	1		
Suono	Melodia 1	1		
	Melodia 2	1		
	Melodia 3	1		
Colore luce	Accensione luce rossa	1		
	Accensione luce gialla	1		
	Accensione luce verde	1		
Rilevare un ostacolo	Rilevare un ostacolo a meno di 10 centimetri	1		
Rilevare un suono (battito delle mani)	Rilevare un rumore come il battito delle mani	1		
Rilevare un colore	Rilevare un oggetto di colore rosso	1		
	Rilevare un oggetto di colore giallo	1		
	Rilevare un oggetto di colore verde	1		
Pulsante	Rilevare la pressione sul pulsante	1		
Condizione	If - End if	3		
Loop	Loop	1		
	Loop: Ripeti fino a quando	3		
Pausa	Pausa 5 sec	3		
ALTRO	Inizio programma	1		

PRESTAZIONI	COMANDI	QUANTITÀ	CODIFICA	DECODIFICA
ALTRO	Fine programma	1		
	Blocco personale 1	1		
	Blocco personale 2	1		

Legenda



Attuatori



Sensori



Comandi di controllo



Codifica: blocchi da inserire nella board



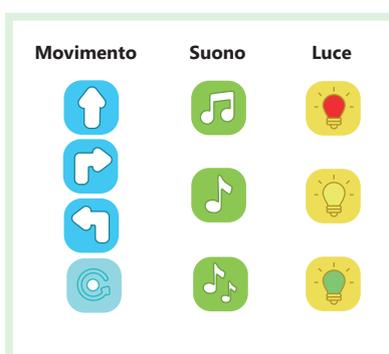
Decodifica: sistema di identificazione ottica

INTERFACCIA BOARD CON BLOCCHI

Per la progettazione dei blocchi e della board è stato preso in considerazione come i bambini non riescano a comprendere sin da subito il corretto ordine di inserimento dei comandi e spesso è difficile distinguerne le diverse funzioni. Per rispondere a queste esigenze sono state utilizzate forme differenti in base alle diverse categorie (attuatori, sensori e comandi di controllo) e colori differenti in base alle diverse funzioni. La programmazione va da sinistra verso destra (a differenza della programmazione classica che va dall'alto verso il basso), il bambino in questa fascia di età tende ad avvicinarsi per la prima volta alla scrittura e seguire lo stesso verso nella programmazione aiuta a non sbagliare.

ATTUATORI

Gli attuatori trasformano un comando in un'azione fisica. Il kit prevede la programmazione di tre tipologie di attuatori: Movimento (programma il movimento dei motori della base del robot e del servomotore), Suono (programma la riproduzione di tre melodie differenti), Luce (programma l'accensione di luce rossa, gialla e verde). Tutti i blocchi attuatori hanno una forma quadrata e ad ogni tipologia corrisponde un colore differente accompagnato da icone.



SENSORI

I sensori rilevano i cambiamenti nell'ambiente circostante convertendo un fenomeno fisico in una tensione analogica misurabile e rispondono mediante alcuni output grazie agli attuatori. Il kit prevede la programmazione di quattro tipologie di sensori: Prossimità, Suono, Colore e Pulsante. Tutti i blocchi sensori hanno una forma quadrata come gli attuatori ma, a differenza di questi, presentano una linguetta sulla sinistra che permette ai bambini di riconoscere facilmente i blocchi ai quali possono essere accostati.

ACCENSIONE LED

feedback visivo successivo all'inserimento di un blocco



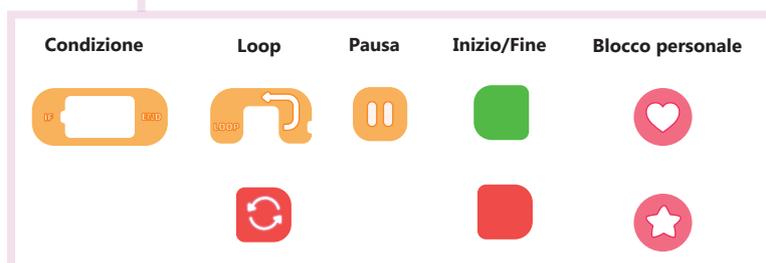
ALLOGGIO BLOCCO

presenta raccordi che aiutano a comprendere il verso in cui avviene la programmazione



COMANDI DI CONTROLLO

Sono comandi che controllano il flusso del codice di programmazione. Il kit prevede l'utilizzo di comandi generici come: Condizione, Loop, Pausa, Inizio/Fine, Blocco personale (richiama una funzione). Questi comandi aiutano i bambini ad ampliare le possibilità di creare codici sempre diversi e comprendere concetti complessi della programmazione.

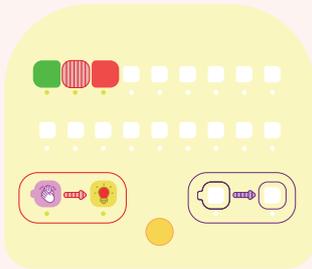


INTERFACCIA BOARD CON BLOCCHI

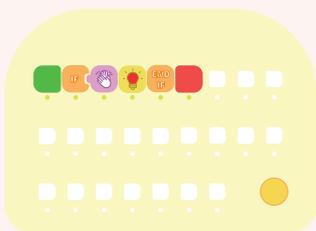
BLOCCO CONDIZIONE

Questa tipologia di blocco ha la forma a cornice e permette di racchiudere al suo interno un blocco sensore e uno attuatore (causa-effetto). Sul perimetro interno è presente un taglio a linguetta che aiuta a comprendere la corretta sequenza di inserimento dei blocchi ovvero prima un sensore e a seguire un attuatore (se viene rilevato un determinato evento allora avviene qualcosa). La forma a cornice ha inoltre la funzione di isolare la condizione dal resto del codice (se la condizione non si verificherà in un lasso di tempo di 30 secondi allora il programma continuerà saltando la condizione). Per la progettazione del blocco erano state prese in considerazione altre due soluzioni di interfaccia.

Prima soluzione: interfaccia aree separate e successivo richiamo della condizione con blocco singolo



Seconda soluzione: interfaccia a blocchi singoli

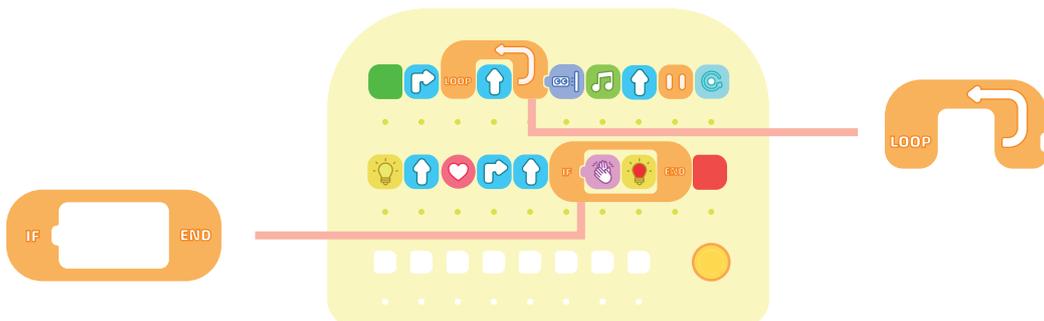
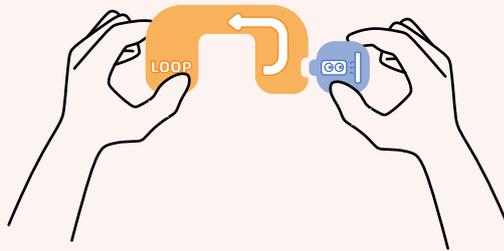
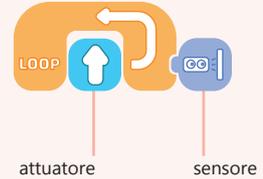
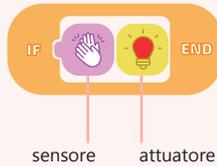


BLOCCO LOOP

Questa tipologia di blocco ha la forma di un arco e permette di ospitare al di sotto di esso il blocco attuatore che si vuole ripetere. Sul perimetro esterno è presente un taglio a linguetta che aiuta a comprendere la corretta sequenza di inserimento dei blocchi. In questo caso, a differenza del blocco condizione, avviene prima l'inserimento di un attuatore e poi di un sensore (ripeti l'azione fino a quando non viene rilevato un determinato evento). La forma ad arco permette di ospitare la freccia che sta ad indicare la ripetizione di un'azione, questo aiuta a comprendere meglio la funzione del blocco.

MODALITÀ DI UTILIZZO DEI BLOCCHI

I bambini possono accostare i diversi blocchi per comprenderne le differenti combinazioni e successivamente inserirli nel giusto ordine all'interno della board.



INTERFACCIA BOARD CON BLOCCHI

BLOCCO PERSONALE

Questa tipologia di blocco permette ai bambini di creare comandi personalizzati da inserire successivamente all'interno del codice.

Per registrare il programma da inserire nel blocco personalizzabile questo dovrà essere posto come primo blocco del codice.



----- Esempio di codice da inserire nel blocco personale:

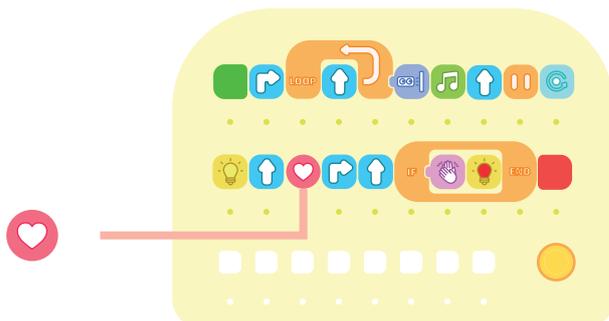
1. Blocco personale
2. Inizio
3. Condizione: se il robot rileva un oggetto vicino allora si accenderà la luce rossa
4. Avanti
5. Fine

Il blocco personalizzabile avrà all'interno le informazioni registrate in precedenza e potrà essere inserito all'interno del programma come un normale blocco.

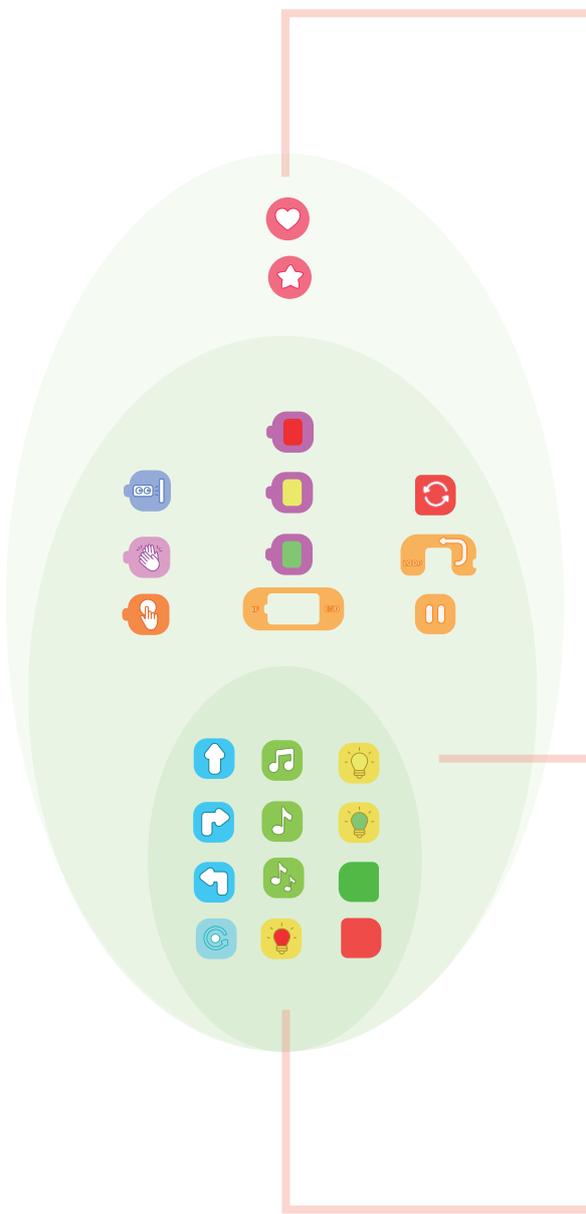


----- Esempio di codice:

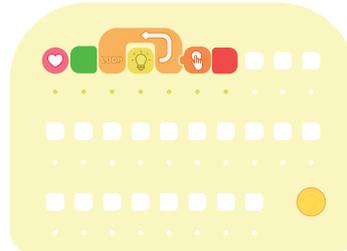
1. Inizio
2. Svolta a destra
3. Loop: vai avanti fino a quando non viene rilevato un oggetto vicino
4. Blocco personale (codice del blocco personale)
5. Fine



APPRENDIMENTO DEI BLOCCHI PER DIVERSE FASCE DI ETÀ



Avanzati (6 anni):
possono creare linee di codice aggiuntive.



Esempio di codice blocco personale:

1. Blocco personale
2. Inizio
3. Loop: la luce gialla resterà accesa fino a quando non verrà premuto il pulsante
4. Fine



Esempio di codice:

1. Inizio
2. Svolta a destra
3. Rotazione di un componente di 360°
4. Avanti
5. Condizione: se il robot rileverà un oggetto vicino allora un componente verrà ruotato di 360°
6. Blocco personale (codice del blocco personale)
7. Fine

Intermedi (5 anni):
possono iniziare ad esplorare i concetti di ripetizione e condizione con l'utilizzo dei sensori.



Esempio di codice:

1. Inizio
2. Svolta a destra
3. Rotazione di un componente di 360°
4. Avanti
5. Condizione: se il robot rileverà un oggetto vicino allora un componente verrà ruotato di 360°
6. Fine

Principianti (3-4 anni):
esplorano gli algoritmi semplici con il sequenziamento lineare.



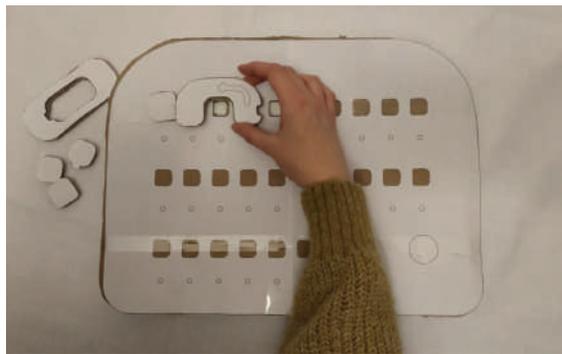
Esempio di codice:

1. Inizio
2. Svolta a destra
3. Rotazione di un componente di 360°
4. Avanti
5. Fine

**PRIMO STUDIO DIMENSIONALE
BOARD E BLOCCHI**



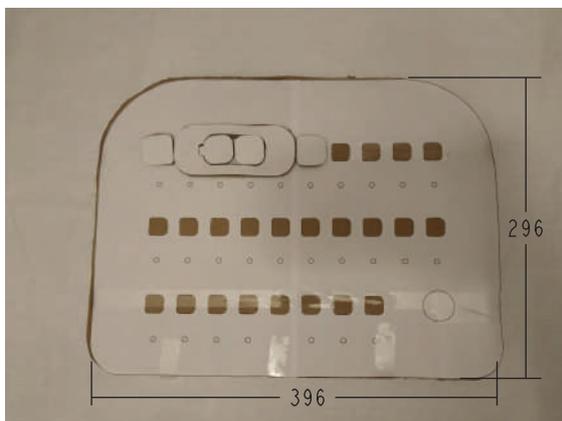
Per questo primo studio sono stati presi in considerazione i volumi della board e dei tappi di alcuni dei blocchi di programmazione



Sono state fatte delle verifiche di inserimento dei blocchi per verificare l'adeguatezza delle dimensioni rispetto alla board.

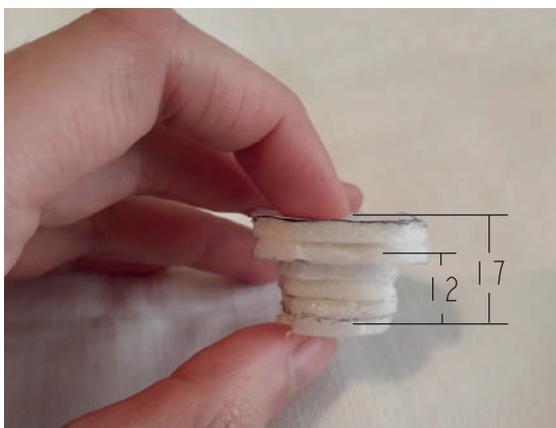
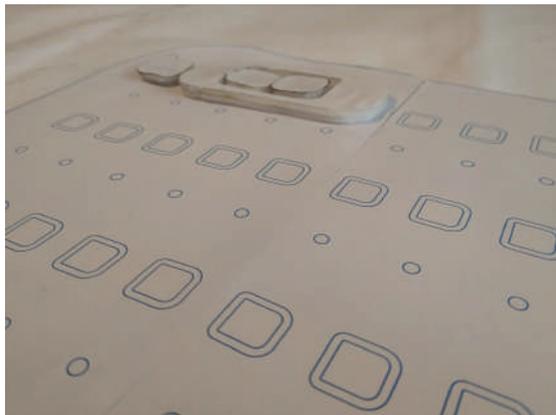


Grazie a questo primo studio si è deciso di procedere con la riduzione delle dimensioni della board e di alcune modifiche nella forma dei blocchi: Loop e Condizione



Esempio di programmazione con blocchi:
inizio, condizione, sensore, attuatore e fine

SECONDO STUDIO DIMENSIONALE BOARD E BLOCCHI



Per questo secondo studio le dimensioni della board sono state ridotte e la forma dei blocchi Condizione e Loop è stata modificata per facilitare l'estrazione dei blocchi da inserire all'interno.

Il blocco di programmazione è stato stratificato per comprendere le possibili altezze del tappo e della scocca.

COMUNICAZIONE BOARD E BLOCCHI: TECNOLOGIA RFID

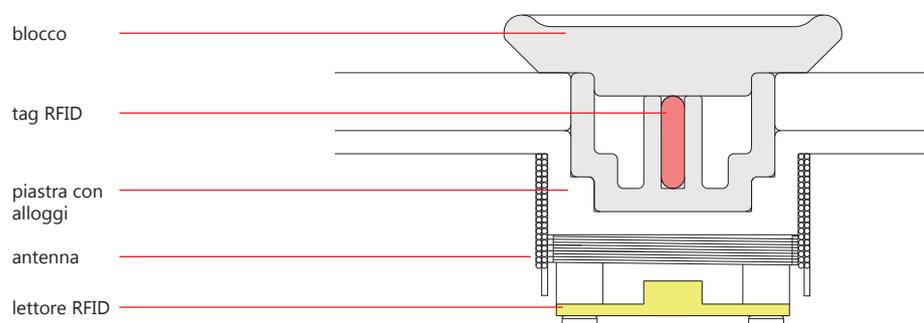
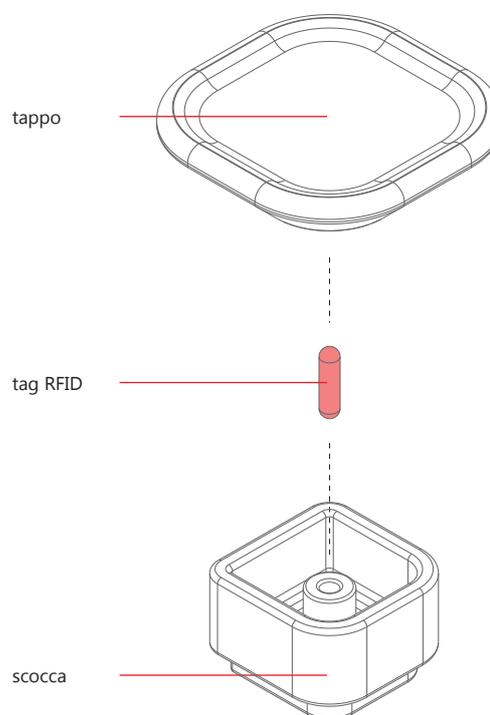
I sensori RFID (identificazione a radiofrequenza) sono un tipo di tecnologia di riconoscimento univoco ed automatico.

Questo tipo di tecnologia è composta principalmente da:

- **Tag:** un trasponder passivo a radiofrequenza che identifica univocamente un oggetto.
- **Antenna:** permette la comunicazione con il tag.
- **Il reader o lettore RFID:** un ricetrasmittitore controllato da un microprocessore che ha il compito di richiedere e ricevere informazioni in risposta da ciascun tag.

Questi elementi permettono di Codificare e Decodificare in modo semplice. Nel progetto è previsto l'inserimento di un tag RFID in ogni blocco per identificarlo in modo univoco. L'utilizzo di tag permette di realizzare una quantità di comandi illimitata poiché è possibile creare infiniti codici identificativi.

Per riconoscere ogni blocco sono stati inseriti antenne e lettori rfid, questi sono posti in corrispondenza di tutti gli alloggi.



DISTINTA COMPONENTI ELETTRONICI PER LA PROTOTIPAZIONE:

Tabella 36- Tabella componenti elettronici board e blocchi

COMPONENTI	DESCRIZIONE	POSIZIONAMENTO	IMMAGINE DI ESEMPIO
Tag RFID	I Tag RfId passivi sono privi di batteria e ricavano l'energia per il loro funzionamento dal segnale proveniente dal lettore	Blocco	
Antenna	Antenna a bobina spirale. Permette la comunicazione con il tag	Blocco	
Lettore RFID	Ricetrasmittitore controllato da un microprocessore che ha il compito di richiedere e ricevere informazioni in risposta da ciascun tag	Blocco	
Arduino mega	Funge da cervello della board e comunica con tutti i componenti elettronici	Board	
Bluetooth	Permette la comunicazione tra la board ed il robot	Board	
Interruttore di accensione a scorrimento	Permette l'accensione della board	Board	
Pulsante di avvio	Permette di avviare il programma	Board	
Led	Restituisce un feedback all'inserimento dei blocchi nella board	Board	
Batterie	Di tipo AA ricaricabili	Board	

DISTINTA COMPONENTI SEMILAVORATI:

Tabella 37- Tabella materiali dei componenti semilavorati di board e blocchi

ID	NOME	POSIZIONAMENTO	QUANTITÀ
PRT_1	scocca	board	1
PRT_2	piastra alloggi	board	1
PRT_3	piastra interfaccia	board	1
PRT_4	vano batteria	board	1
PRT_5	copertura vano batteria	board	1
PRT_6	scocca blocco	blocco	45
PRT_7	tappo blocco attuatore	blocco	22
PRT_8	tappo blocco sensore	blocco	6
PRT_9	tappo blocco inizio	blocco	1
PRT_10	tappo blocco fine- loop	blocco	2
PRT_11	tappo blocco condizione	blocco	3
PRT_12	tappo blocco loop: ripeti fino a quando	blocco	3
PRT_13	tappo blocco personale	blocco	2

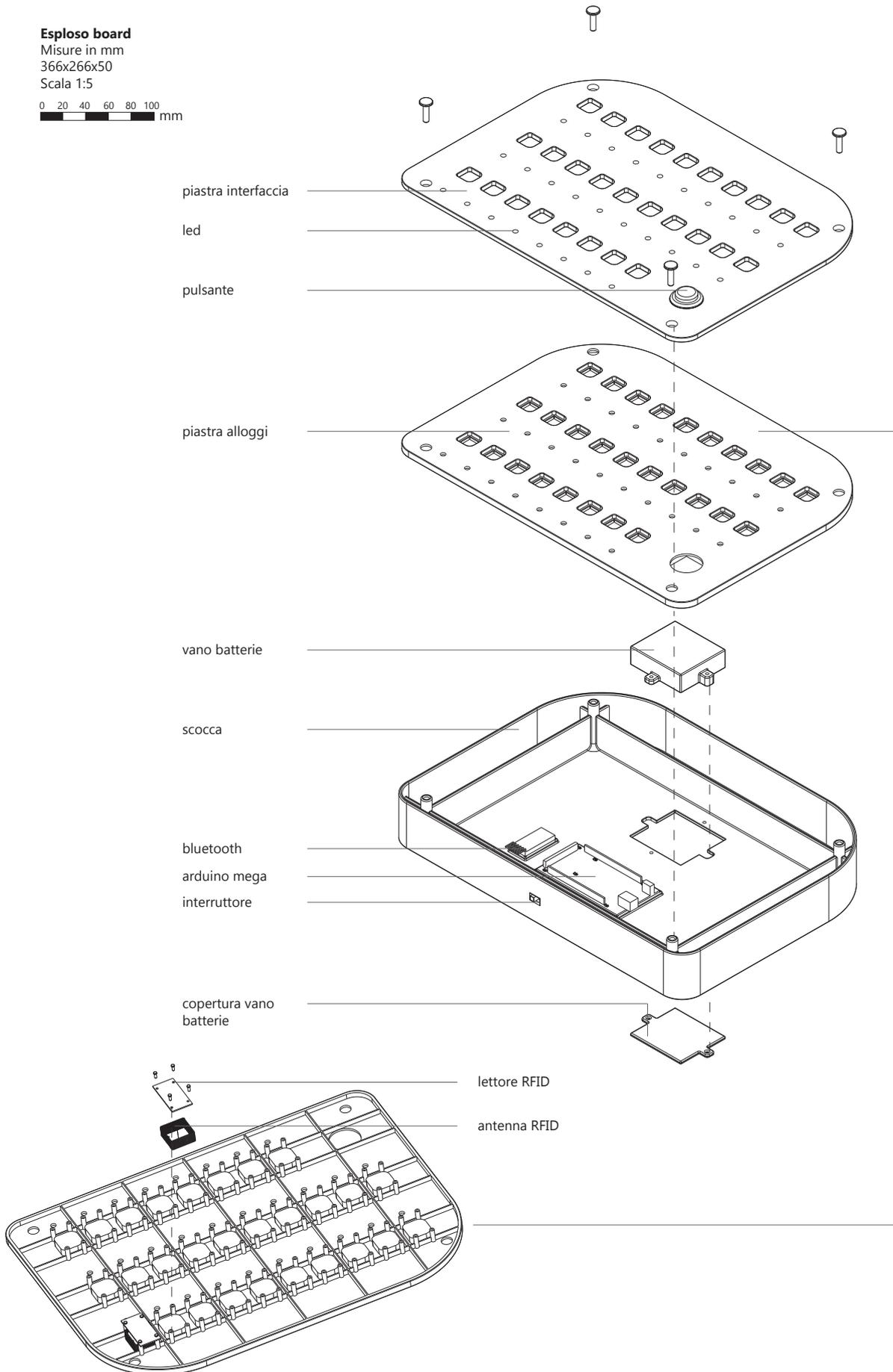
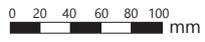
DISTINTA MATERIALI :

Tabella 38- Tabella materiali dei componenti di board e blocchi

MATERIALE	DESCRIZIONE	COMPONENTI
<p>ABS (acrilonitrile butadiene stirene)</p> 	<p>Il materiale ABS, acronimo di acrilonitrile butadiene stirene, è un materiale polimerico termoplastico che ha come caratteristiche primarie la leggerezza e la rigidità. Altra caratteristica importante è la possibilità di essere riscaldato ripetutamente che lo rende una delle migliori materie plastiche per il riciclaggio. Assieme al PVC, la ABS plastica è molto usata nello stampaggio ad iniezione ed è resistente a molti solventi e altri agenti aggressivi.</p> <p>È stato scelto questo materiale perchè è più adatto ad essere utilizzato dai bambini, non è tossico ed è resistente agli urti. I bambini spesso tendono a sottoporre i giocattoli a numerosi test di resistenza, avere un materiale resistente permette una durata maggiore del prodotto. Inoltre è un materiale facilmente riciclabile se non contaminato da altre tipologie di plastiche non compatibili.</p> <p>CARATTERISTICHE FISICHE:</p> <p>Densità: 1,04 g/cm³ Temperatura di fusione(°C): 200-280; consigliato 230 Temperatura di transizione vetrosa (°C): 105 Calore specifico (Kj/kg*K):1.4</p> <p>CARATTERISTICHE MECCANICHE:</p> <p>Resistenza alla trazione (MPa): 50 Dilatazione alla rottura (%) : 20 Resistenza alla flessione (MPa): 70 Modulo di elasticità (MPa): 2500 Resilienza (kj/m2): 15 Carico ammissibile a compressione (MPa):40</p> <p>CARATTERISTICHE TERMICHE:</p> <p>Coefficiente di dilatazione termica lineare (K -1): 0.7*10⁻⁴ Conducibilità termica(W/mk): 0.16</p>	<p>PRT_1, PRT_2, PRT_4, PRT_5, PRT_6, PRT_7, PRT_8, PRT_9, PRT_10, PRT_11, PRT_12, PRT:13</p>
<p>Compensato di betulla</p> 	<p>Il compensato è un semilavorato a strati di legno prodotti dal tronco dell'albero. Appartiene alla grande classe dei pannelli stratificati o multistrati, dai quali si distingue per l'orientamento incrociato dei vari strati. Il compensato di betulla è leggero ma resistente, tra i compensati più stabili dimensionalmente, anche se non è adatto agli utilizzi in esterni. Il legno di betulla è di colore chiaro e a grana fine, con una superficie molto liscia.</p> <p>È stato scelto questo materiale perchè leggero e semplice da lavorare. Inoltre il suo colore molto chiaro lo rende uno dei compensati più utilizzati per la produzione dei giocattoli.</p> <p>CARATTERISTICHE FISICHE:</p> <p>Densità: 0,7 g/cm³ Lavorazione: buona lavorabilità, giunzioni resistenti Impieghi: mobili, compensati</p> <p>CARATTERISTICHE MECCANICHE:</p> <p>Modulo di elasticità (N/mm2) : 13000 Carico di rottura a trazione (N/mm2): 95 Carico di rottura a compressione assiale (N/mm2): 60 Carico di rottura a flessione (N/mm2): 120 Carico di rottura a taglio(N/mm2): 6</p>	<p>PRT_3</p>

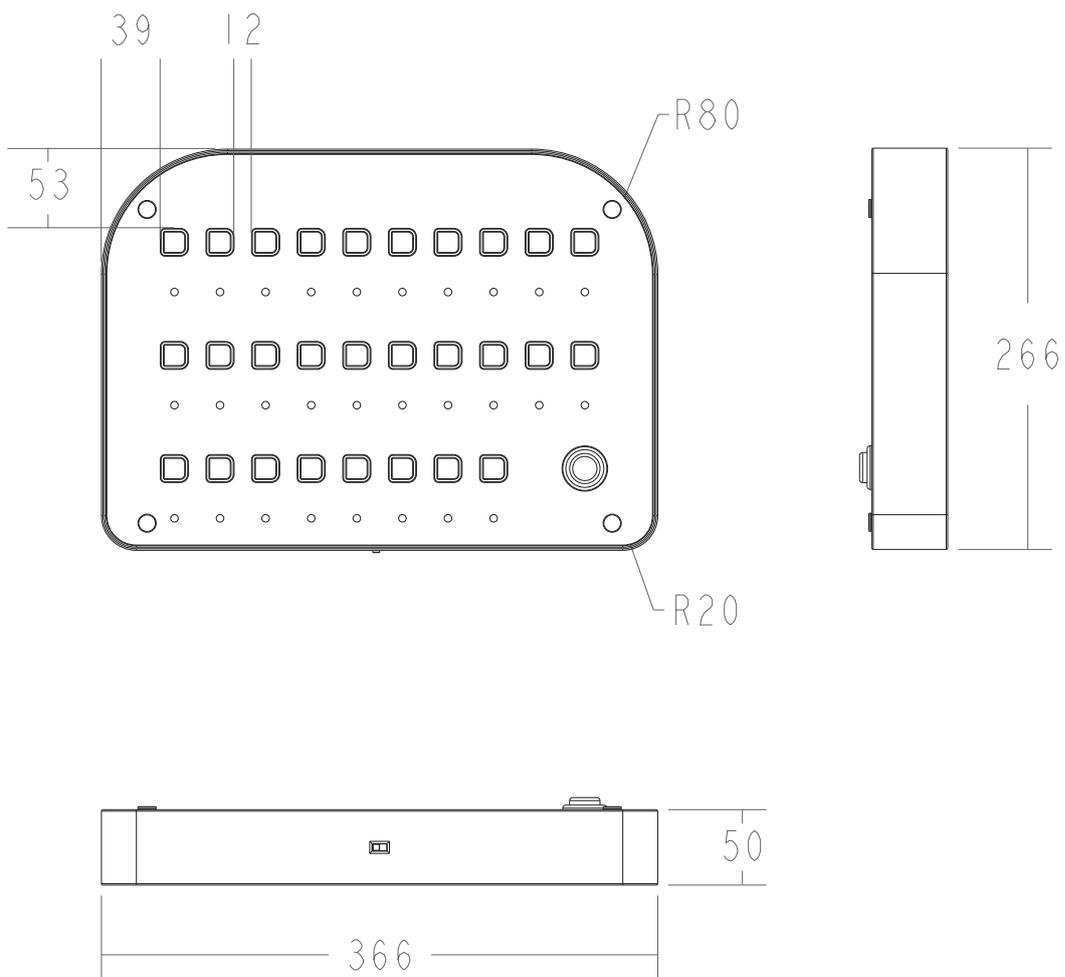
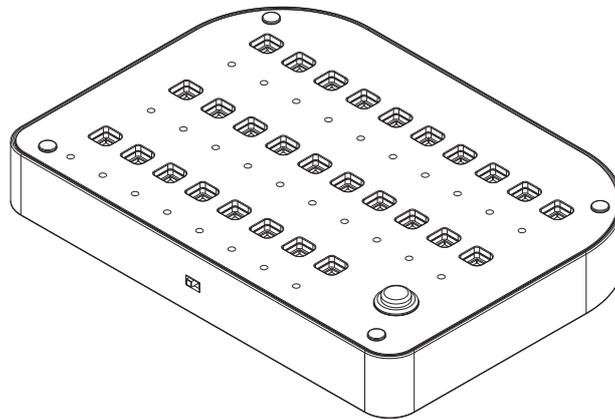
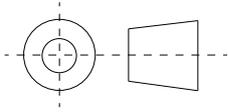
BOARD

Esploso board
Misure in mm
366x266x50
Scala 1:5



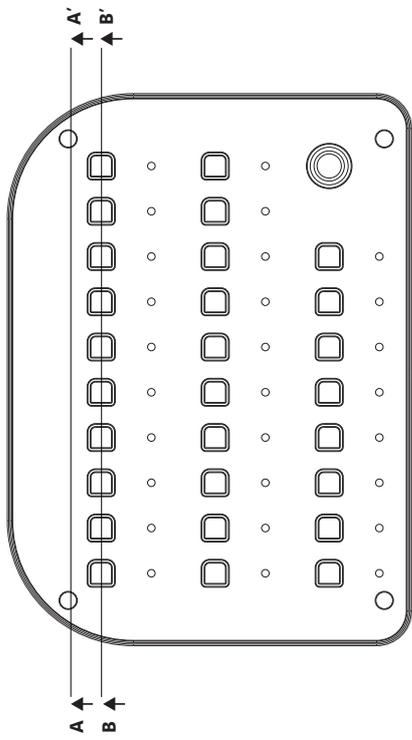
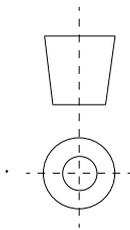
BOARD

Misure in mm
Scala 1:5



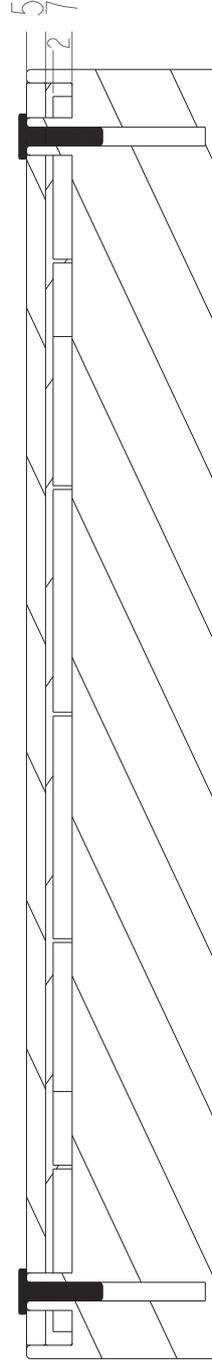
BOARD

Misure in mm
Scala 1:5



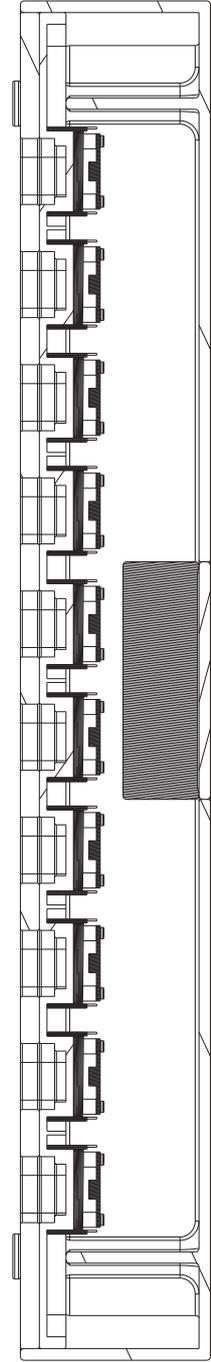
A-A'

Misure in mm
Scala 1:2



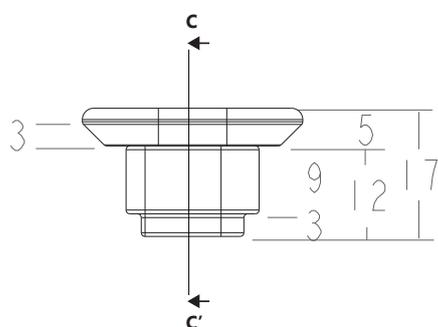
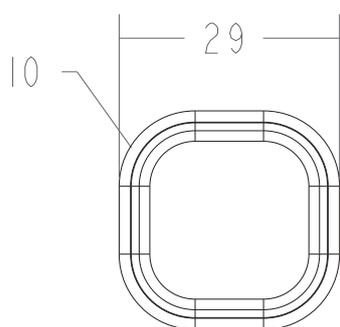
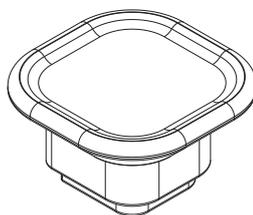
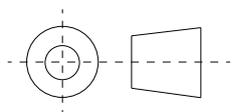
B-B'

Misure in mm
Scala 1:2



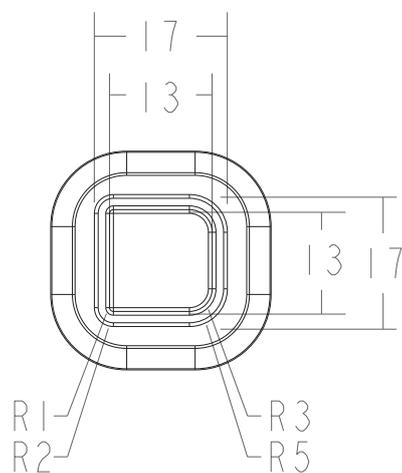
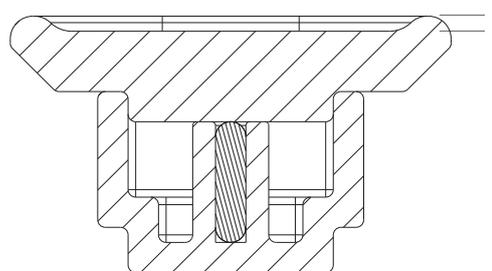
BLOCCO ATTUATORE

Misure in mm
Scala 1:1



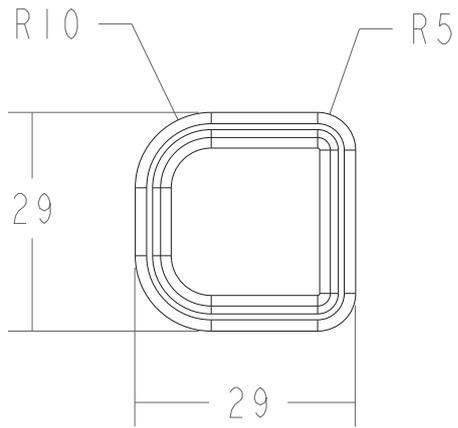
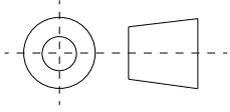
C-C'

Misure in mm
scala 2:1



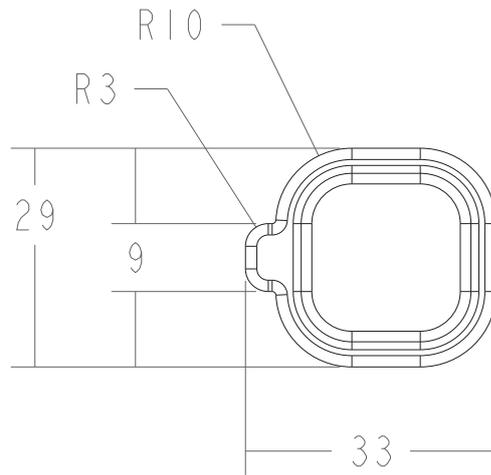
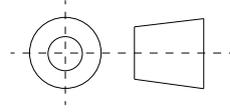
BLOCCO INIZIO-FINE

Misure in mm
Scala 2:1



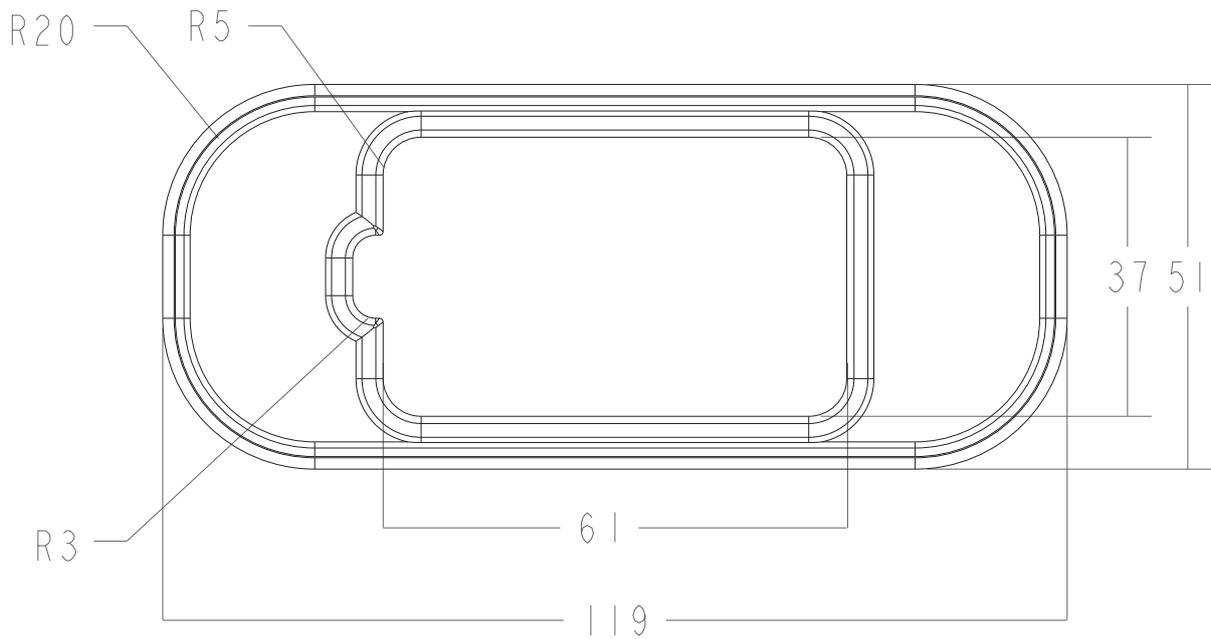
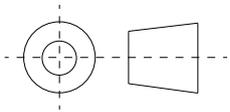
BLOCCO SENSORE

Misure in mm
Scala 2:1



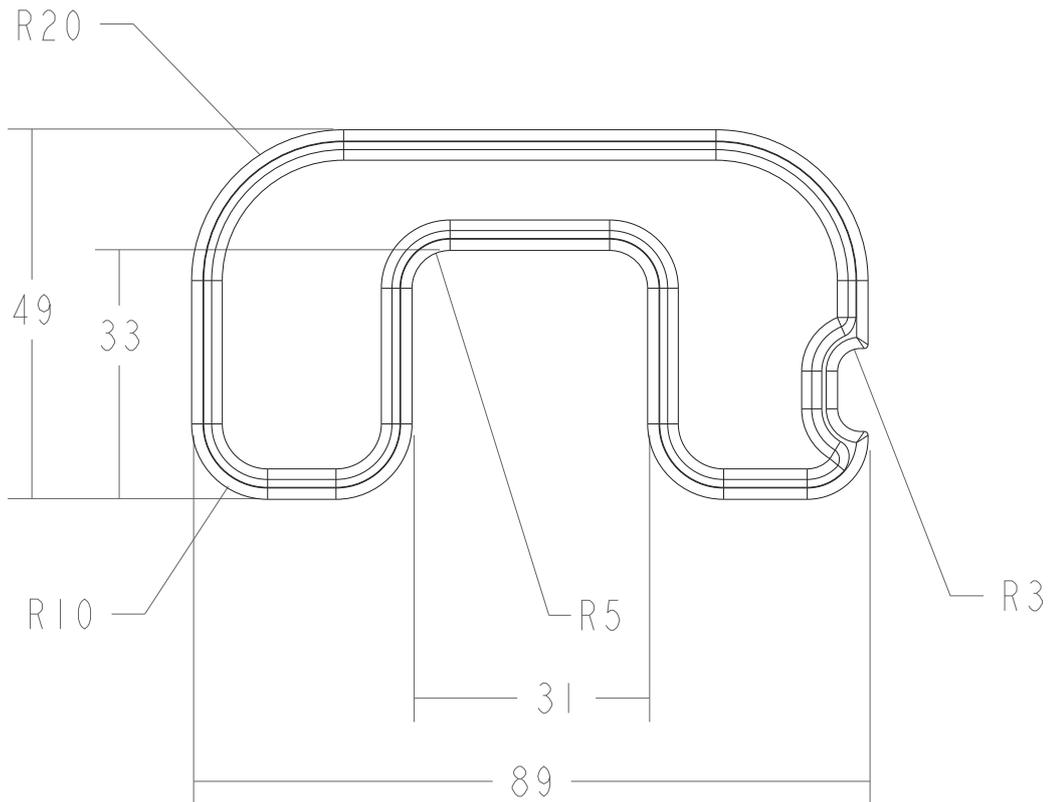
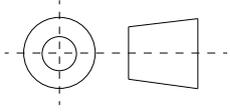
BLOCCO CONDIZIONE

Misure in mm
Scala 2:1



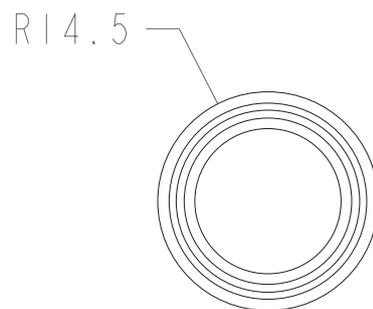
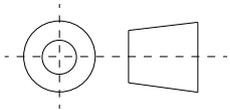
BLOCCO LOOP

Misure in mm
Scala 2:1



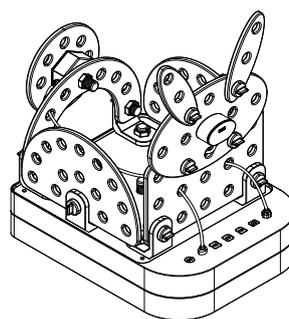
BLOCCO PERSONALE

Misure in mm
Scala 2:1

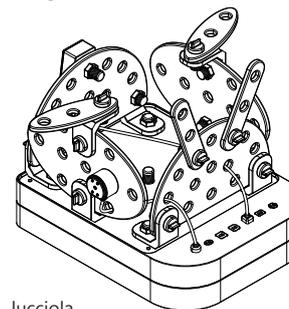


6.3 Robot

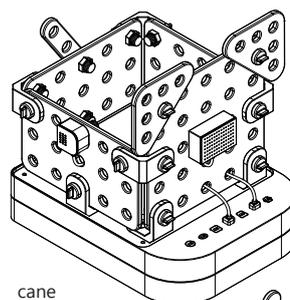
Il kit Combo prevede l'utilizzo di un Robot per avere un feedback immediato delle azioni programmate sulla board. Il robot non si presenta assemblato nel suo insieme ma saranno i bambini a comporlo in base alle diverse necessità sviluppando in questo modo non solo la creatività ma anche la collaborazione e il problem solving. Nel kit sono presenti sensori e attuatori da poter applicare durante la creazione del robot. Questi vengono programmati grazie ai comandi (blocchi) da inserire nella board. Ad ogni funzione dei blocchi di programmazione è associato un componente elettronico, i bambini possono comprendere facilmente l'associazione poichè ogni componente elettronico (sensore e attuttore) ha lo stesso colore del blocco di programmazione corrispondente (es. Il componente "Attuatore Led" è di colore giallo come i blocchi di programmazione corrispondenti alla funzione "Luce"). I componenti elettronici possono essere facilmente collegati alla base del robot tramite cavi. Per aiutare i bambini ad inserire i cavi nella presa corretta, ogni copriconnettore del cavo ha una forma differente corrispondente alla forma del sensore o attuttore a cui è associato. I bambini potranno assemblare i componenti elettronici del robot anche in base ai diversi gradi di apprendimento provando prima il solo movimento del robot e successivamente sperimentando le diverse funzioni dei sensori e degli attuatori, questo permette di creare un apprendimento continuo che accompagna i bambini nelle diverse fasce di età. L'aspetto del robot può essere personalizzato inoltre, anche grazie ad elementi in legno di dimensioni e forme geometriche differenti da unire attraverso connettori come viti e bulloni. Per introdurre i bambini alla costruzione nel kit sono state create configurazioni di esempio per la realizzazione di cinque animali diversi quali: coniglio, lucciola, cane, giraffa e orso.



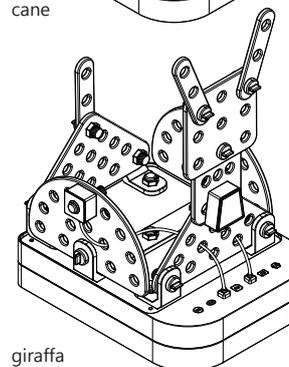
coniglio



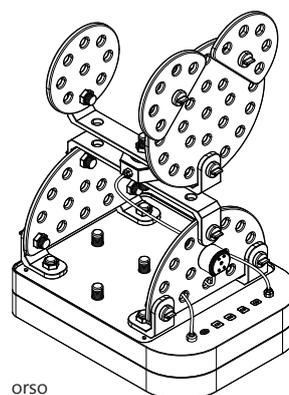
lucciola



cane



giraffa

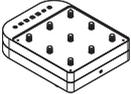


orso

COMPONENTI ROBOT

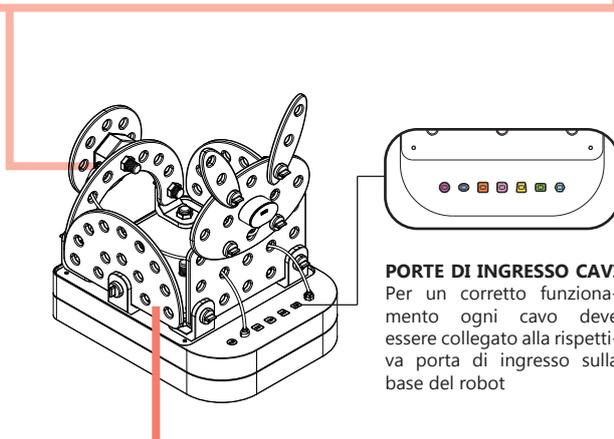
COMPONENTI ELETTRONICI

Il kit prevede l'utilizzo di diversi componenti elettronici programmabili quali: una base robot, sensori, attuatori e cavi di connessione. I sensori e gli attuatori presentano forme differenti per essere distinti dai bambini, i colori utilizzati sono quelli dei blocchi di programmazione corrispondenti alla loro funzione.

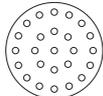
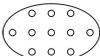
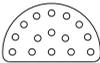
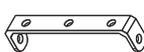
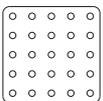
	base robot	x1		sensore colore	x1		cavo attuatore riproduttore audio	x1	Lunghezza: 1 m		cavo sensore suono	x1	Lunghezza: 1 m
	attuatore riproduttore audio	x1		sensore suono	x1		cavo attuatore led	x1	Lunghezza: 1 m		cavo sensore prossimità	x1	Lunghezza: 1 m
	attuatore Led	x1		sensore prossimità	x1		cavo attuatore servomotore	x1	Lunghezza: 1 m		cavo sensore pulsante	x1	Lunghezza: 1 m
	attuatore servomotore	x1		sensore pulsante	x1		cavo sensore colore	x1	Lunghezza: 1 m				

COMPONENTI COSTRUZIONE

Il kit prevede l'utilizzo di componenti per la personalizzazione della forma del robot. Tra questi troviamo principalmente componenti di costruzione in legno di forme differenti da unire tra loro grazie a connettori in plastica. È stata scelta questa tipologia di componenti poiché grazie alle diverse forme e dimensioni permettono di creare configurazioni differenti in modo semplice per bambini della scuola dell'infanzia. Inoltre, grazie ai fori presenti su di essi, i bambini possono nascondere con facilità, nella parte centrale del robot, i cavi per il collegamento dei componenti elettronici. Grazie ai fori per il passaggio, si evita di avere fili sporgenti che impediscano in qualche modo il corretto movimento del robot.



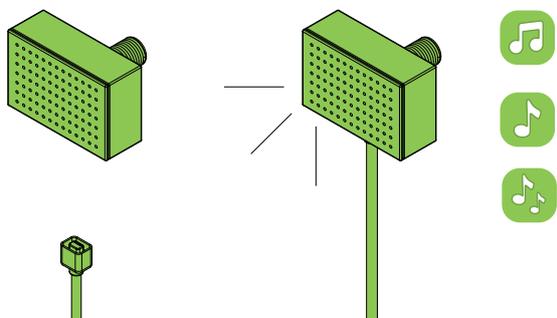
PORTE DI INGRESSO CAVI
Per un corretto funzionamento ogni cavo deve essere collegato alla rispettiva porta di ingresso sulla base del robot

	cerchio grande	x4		ovale grande	x4		arco	x4
	cerchio piccolo	x4		ovale piccolo	x4		barra	x4
	rettangolo grande	x4		semicerchio grande	x4		connettore singolo	x12
	rettangolo piccolo	x4		semicerchio piccolo	x4		connettore doppio	x6
	quadrato grande	x4		triangolo grande	x4		vite	x50
	quadrato piccolo	x4		triangolo piccolo	x4		bullone	x50

FUNZIONI COMPONENTI ELETTRONICI

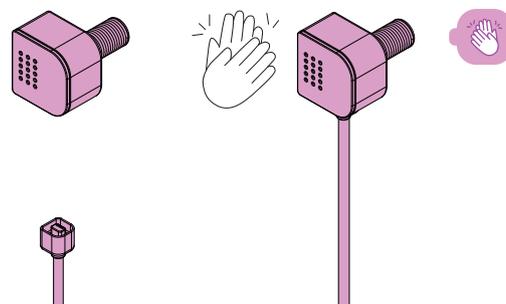
ATTUATORE RIPRODUTTORE AUDIO

riproduce fino a tre melodie differenti in base alla programmazione



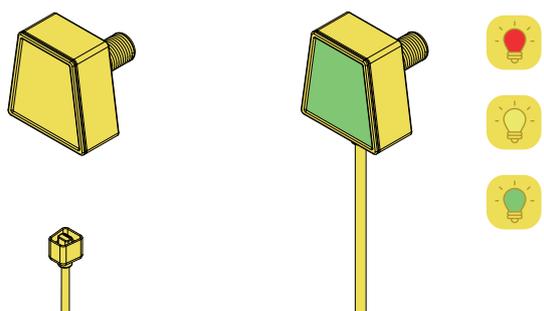
SENSORE SUONO

rileva un rumore come il battito delle mani



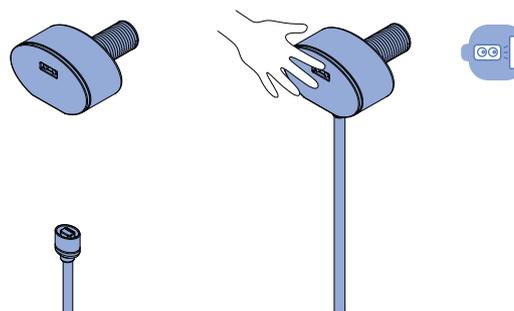
ATTUATORE LED

la luce può accendersi fino a tre colori differenti in base alla programmazione



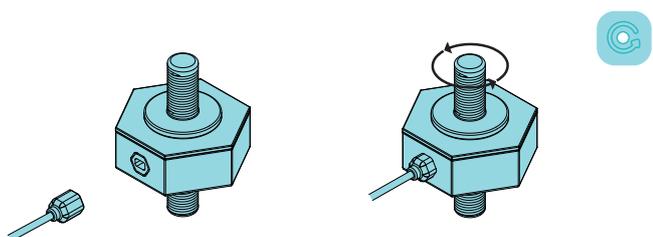
SENSORE PROSSIMITÀ

rileva la presenza di un ostacolo



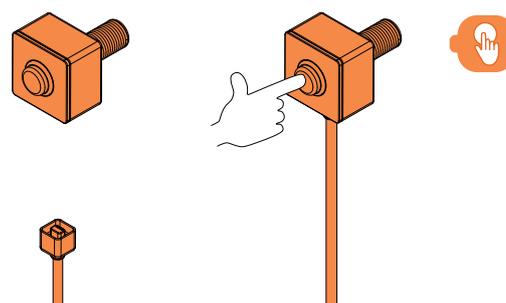
ATTUATORE SERVOMOTORE

permette una rotazione di 360° del componente a cui viene connesso



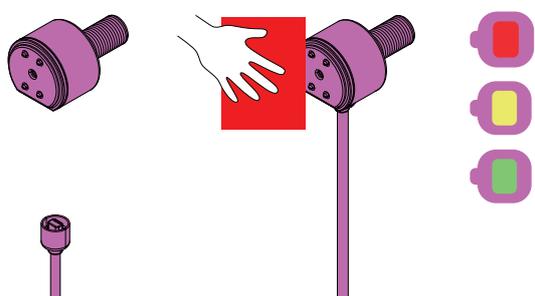
PULSANTE

rileva la pressione su di esso



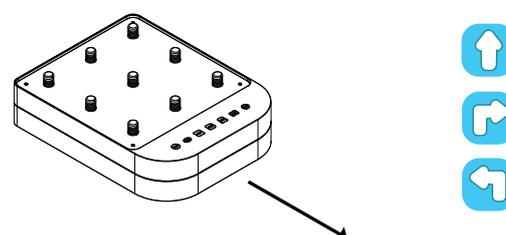
SENSORE COLORE

rileva fino a tre colori differenti in base alla programmazione



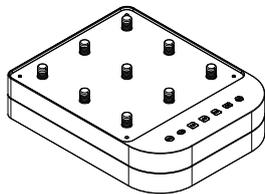
BASE ROBOT

permette il movimento del robot in tre direzioni differenti in base alla programmazione

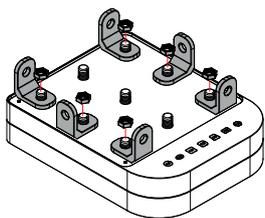


FASI COSTRUTTIVE DI ESEMPIO: Coniglio

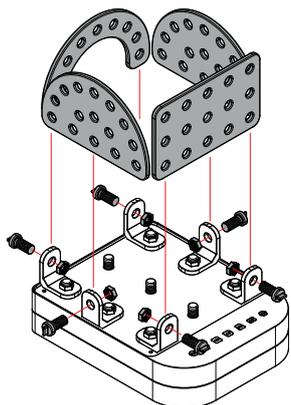
1. Base robot



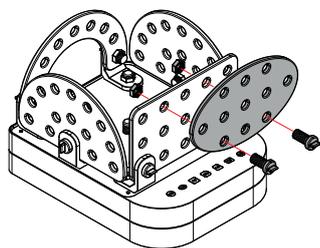
2. Assemblaggio connettori singoli



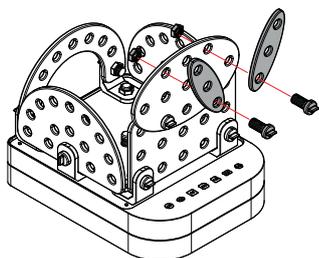
3. Assemblaggio corpo del coniglio



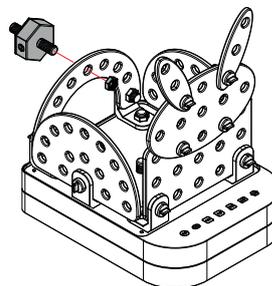
4. Assemblaggio testa del coniglio



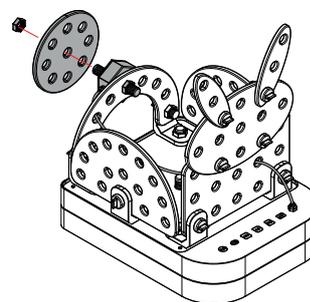
5. Assemblaggio orecchie del coniglio



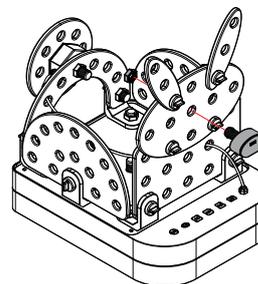
6. Assemblaggio servomotore



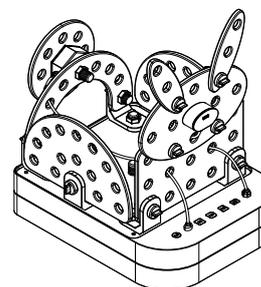
7. Assemblaggio coda del coniglio ed inserimento cavo servomotore



8. Assemblaggio sensore di prossimità



9. Forma definitiva

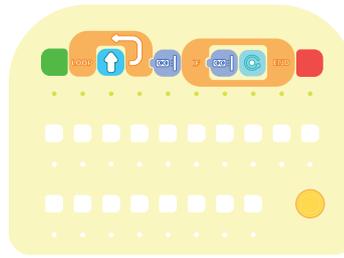
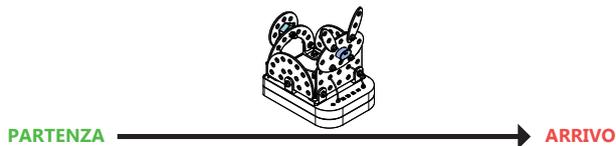


ESEMPIO DI ESERCIZIO

Si chiede ai bambini di aiutare il coniglio a raggiungere il punto B e una volta giunto all'arrivo se il coniglio incontrerà un ostacolo dovrà muovere la coda.

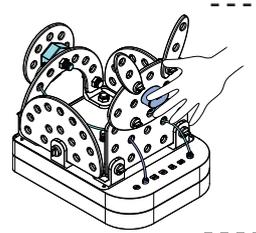
SOLUZIONE 1

Il coniglio si muoverà in avanti fino a quando non incontrerà un ostacolo e muoverà la coda quando incontrerà un altro ostacolo.

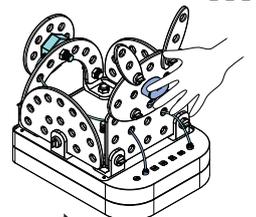


Codice:

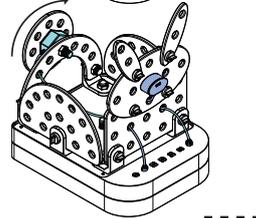
1. Inizio
2. Loop: il robot si muoverà in avanti fino a quando non rileverà un ostacolo
3. Condizione: se il robot rileverà un ostacolo allora un componente (in questo caso la coda) verrà ruotato di 360°
4. Fine



Loop: il robot si muoverà in avanti fino a quando non rileverà un ostacolo

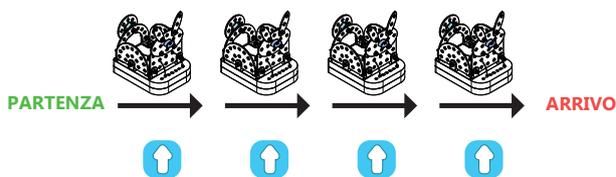


Condizione: se il robot rileverà un ostacolo allora un componente (in questo caso la coda) verrà ruotato di 360°



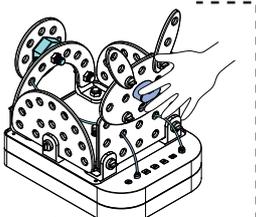
SOLUZIONE 2

Il coniglio si muoverà in avanti per 4 volte e muoverà la coda quando incontrerà un ostacolo.

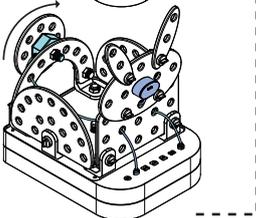


Codice:

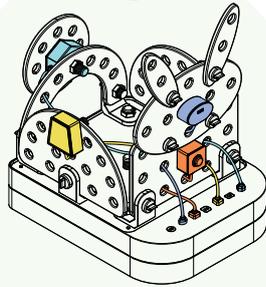
1. Inizio
2. Avanti x4
3. Condizione: se il robot rileverà un ostacolo allora un componente (in questo caso la coda) verrà ruotato di 360°
4. Fine



Condizione: se il robot rileverà un ostacolo allora un componente (in questo caso la coda) verrà ruotato di 360°



APPRENDIMENTO PER DIVERSE FASCE DI ETÀ

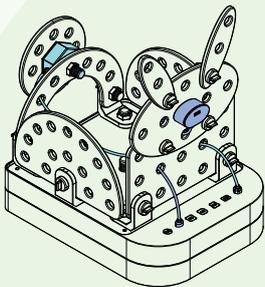


Avanzati (6 anni):
possono creare linee di codice aggiuntive.



Esempio di codice blocco personale:

1. Blocco personale
2. Inizio
3. Loop: la luce gialla resterà accesa fino a quando non verrà premuto il pulsante
4. Fine

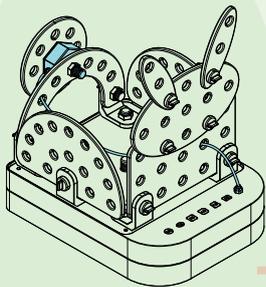


Intermedi (5 anni):
possono iniziare ad esplorare i concetti di ripetizione e condizione con l'utilizzo dei sensori.



Esempio di codice:

1. Inizio
2. Svolta a destra
3. Rotazione di un componente di 360°
4. Avanti
5. Condizione: se il robot rileverà un oggetto vicino allora un componente verrà ruotato di 360°
6. Blocco personale (codice del blocco personale)
7. Fine



Principianti (3-4 anni):
esplorano gli algoritmi semplici con il sequenziamento lineare.



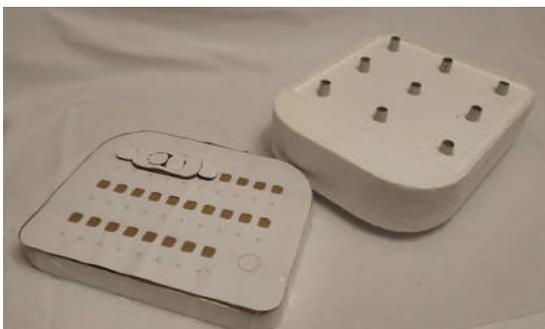
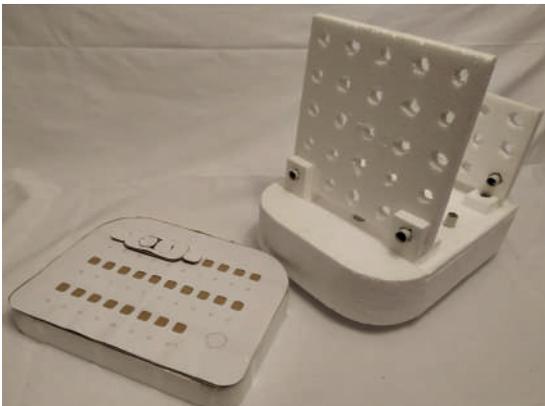
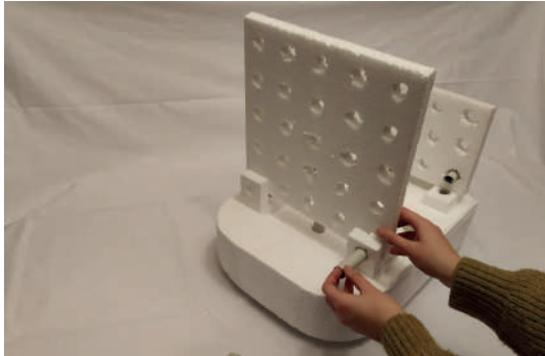
Esempio di codice:

1. Inizio
2. Svolta a destra
3. Rotazione di un componente di 360°
4. Avanti
5. Condizione: se il robot rileverà un oggetto vicino allora un componente verrà ruotato di 360°
6. Fine

Codice di esempio:

1. Inizio
2. Svolta a destra
3. Rotazione di un componente di 360°
4. Avanti
5. Fine

PRIMO STUDIO DIMENSIONALE ROBOT



Per questo modello sono stati presi in considerazione i volumi della base del robot e di alcuni dei componenti di costruzione. Da questo studio è scaturito che le dimensioni sono adatte per utilizzare il robot in gruppo e rendere la costruzione semplice.

Confronto dimensionale con il primo studio della board

Dimensioni primo studio della board: 396x296x50

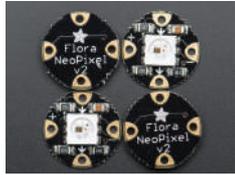
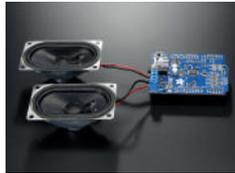
Dimensioni base del robot: 445x350x155

Successivamente a questo confronto si è scelto di procedere con un ridimensionamento della board.

DISTINTA COMPONENTI ELETTRONICI PER LA PROTOTIPAZIONE :

Tabella 39- Tabella componenti elettronici board e blocchi

COMPONENTI	DESCRIZIONE	POSIZIONAMENTO	IMMAGINE DI ESEMPIO
Arduino mega	Funge da cervello della board e comunica con tutti i componenti elettronici	Base robot	
Bluetooth	Permette la comunicazione tra la board ed il robot	Base robot	
Scheda di controllo	Controlla la velocità dei motori	Base robot	
Motori	Permettono il movimento del robot	Base robot	
Sensore colore	Rileva il colore di una superficie. I sensori emettono luce (rossa, verde, blu) sull'oggetto da rilevare, in base al raggio riflesso determinano le coordinate di cromaticità	Sensore colore	
Sensore prossimità	Sensore di prossimità di tipo infrarossi rileva la presenza di un ostacolo	Sensore prossimità	
Sensore suono	Rileva l'intensità del suono	Sensore suono	

COMPONENTI	DESCRIZIONE	POSIZIONAMENTO	IMMAGINE DI ESEMPIO
Sensore tattile pulsante	Rileva il tocco quando viene premuto	Sensore Pulsante	
Led RGB	Emettono luce di colori diversi. I led RGB combinano i colori rosso, blu e verde per creare diverse tonalità di luce	Attuatore Led	
Servomotore	Permette il movimento di un componente	Attuatore Servomotore	
Riproduttore audio	Emette diverse melodie	Attuatore Riproduttore audio	
Interruttore di accensione a scorrimento	Permette l'accensione del robot	Base robot	
Batterie	Di tipo AA ricaricabili.	Base robot	
Connettore elettrico: RJ11 femmina	Permette il collegamento tra la base del robot e i sensori e attuatori	Base robot, sensori (colore, prossimità, suono, pulsante) attuatori (led, servomotore, riproduttore audio)	
Connettore elettrico: RJ11 maschio	Permette il collegamento tra la base del robot e i sensori e attuatori	scocche RJ11	
Cavo elettrico	Permette il collegamento tra la base del robot e i sensori e attuatori. Lunghezza: 1 m	scocche RJ11	

DISTINTA COMPONENTI SEMILAVORATI:

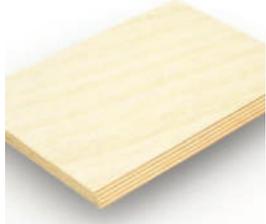
Tabella 40- Tabella dei componenti semilavorati del Robot

ID	NOME	POSIZIONAMENTO	QUANTITÀ
PRT_1	scocca inferiore	base robot	1
PRT_2	scocca superiore	base robot	1
PRT_3	alloggio motore	base robot	2
PRT_4	vano batteria	base robot	1
PRT_5	copertura vano batteria	base robot	1
PRT_6	ruota grande	base robot	2
PRT_7	pneumatico per ruota grande	base robot	2
PRT_8	ruota piccola	base robot	2
PRT_9	pneumatico per ruota piccola	base robot	2
PRT_10	asse rotazione ruota grande	base robot	2
PRT_11	base rotazione per ruota piccola (piroettante)	base robot	2
PRT_12	piastra montaggio	base robot	1
PRT_13	vite	vite	50
PRT_14	bullone	bullone	50
PRT_15	connettore singolo	connettore singolo	12
PRT_16	connettore doppio	connettore doppio	6
PRT_17	arco	arco	4
PRT_18	cerchio grande	cerchio grande	4
PRT_19	cerchio piccolo	cerchio piccolo	4
PRT_20	quadrato grande	quadrato grande	4
PRT_21	quadrato piccolo	quadrato piccolo	4
PRT_22	triangolo grande	triangolo grande	4
PRT_23	triangolo piccolo	triangolo piccolo	4
PRT_24	rettangolo grande	rettangolo grande	4
PRT_25	rettangolo piccolo	rettangolo piccolo	4
PRT_26	ovale grande	ovale grande	4

ID	NOME	POSIZIONAMENTO	QUANTITÀ
PRT_27	ovale piccolo	ovale piccolo	4
PRT_28	semicerchio grande	semicerchio grande	4
PRT_29	semicerchio piccolo	semicerchio piccolo	4
PRT_30	barra	barra	4
PRT_31	scocca attuatore led	led	1
PRT_32	tappo attuatore led	led	1
PRT_33	alloggio schede attuatore led	led	1
PRT_34	copertura trasparente attuatore led	led	1
PRT_35	scocca attuatore servomotore	servomotore	1
PRT_36	tappo attuatore servomotore	servomotore	1
PRT_37	scocca attuatore riproduttore audio	riproduttore audio	1
PRT_38	tappo attuatore riproduttore audio	riproduttore audio	1
PRT_39	scocca sensore colore	sensore colore	1
PRT_40	tappo sensore colore	sensore colore	1
PRT_41	scocca sensore prossimità	sensore prossimità	1
PRT_42	tappo sensore prossimità	sensore prossimità	1
PRT_43	scocca sensore suono	sensore suono	1
PRT_44	tappo sensore suono	sensore suono	1
PRT_45	scocca sensore pulsante	pulsante	1
PRT_46	tappo sensore pulsante	pulsante	1
PRT_47	scocca connettore RJ11 attuatore led	cavo attuatore led	2
PRT_48	scocca connettore RJ11 attuatore servomotore	cavo attuatore servomotore	2
PRT_49	scocca connettore RJ11 att. riproduttore audio	cavo attuatore riproduttore audio	2
PRT_50	scocca connettore RJ11 sensore colore	cavo sensore colore	2
PRT_51	scocca connettore RJ11 sensore prossimità	cavo sensore prossimità	2
PRT_52	scocca connettore RJ11 sensore suono	cavo sensore suono	2
PRT_53	scocca connettore RJ11 sensore tattile	cavo pulsante	2
PRT_54	connettore sensori e attuatori	scocche sensori e attuatori	7

DISTINTA MATERIALI :

Tabella 41- Tabella materiali dei componenti del Robot

MATERIALE	DESCRIZIONE	COMPONENTI
<p>ABS (acrilonitrile butadiene stirene)</p> 	<p>Il materiale ABS, acronimo di acrilonitrile butadiene stirene, è un materiale polimerico termoplastico che ha come caratteristiche primarie la leggerezza e la rigidità. Altra caratteristica importante è la possibilità di essere riscaldato ripetutamente che lo rende una delle migliori materie plastiche per il riciclaggio. Assieme al PVC, la ABS plastica è molto usata nello stampaggio ad iniezione ed è resistente a molti solventi e altri agenti aggressivi.</p> <p>È stato scelto questo materiale perchè è più adatto ad essere utilizzato dai bambini, non è tossico ed è resistente agli urti. I bambini spesso tendono a sottoporre i giocattoli a numerosi test di resistenza, avere un materiale resistente permette una durata maggiore del prodotto. Inoltre è un materiale facilmente riciclabile se non contaminato da altre tipologie di plastiche non compatibili.</p> <p>CARATTERISTICHE FISICHE:</p> <p>Densità: 1,04 g/cm³ Temperatura di fusione(°C): 200-280; consigliato 230 Temperatura di transizione vetrosa (°C): 105 Calore specifico (Kj/kg*K):1.4</p> <p>CARATTERISTICHE MECCANICHE:</p> <p>Resistenza alla trazione (MPa): 50 Dilatazione alla rottura (%) : 20 Resistenza alla flessione (MPa): 70 Modulo di elasticità (MPa): 2500 Resilienza (kj/m2): 15 Carico ammissibile a compressione (MPa):40</p> <p>CARATTERISTICHE TERMICHE:</p> <p>Coefficiente di dilatazione termica lineare (K -1): 0.7*10⁻⁴ Conducibilità termica(W/mk): 0.16</p>	<p>PRT_1, PRT_2, PRT_3, PRT_4, PRT_5, PRT_6, PRT_8, PRT_10, PRT_11, PRT_12, PRT_13, PRT_14, PRT_15, PRT_16, PRT_31, PRT_32, PRT_33, PRT_34, PRT_35, PRT_36, PRT_37, PRT_38, PRT_39, PRT_40, PRT_41, PRT_42, PRT_43, PRT_44, PRT_45, PRT_46, PRT_47, PRT_48, PRT_49, PRT_50, PRT_51, PRT_52, PRT_53, PRT_54</p>
<p>Compensato di betulla</p> 	<p>Il compensato è un semilavorato a strati di legno prodotti dal tronco dell'albero. Appartiene alla grande classe dei pannelli stratificati o multistrati, dai quali si distingue per l'orientamento incrociato dei vari strati. Il compensato di betulla è leggero ma resistente, tra i compensati più stabili dimensionalmente, anche se non è adatto agli utilizzi in esterni. Il legno di betulla è di colore chiaro e a grana fine, con una superficie molto liscia.</p> <p>È stato scelto questo materiale perchè leggero e semplice da lavorare. Inoltre il suo colore molto chiaro lo rende uno dei compensati più utilizzati per la produzione dei giocattoli.</p> <p>CARATTERISTICHE FISICHE:</p> <p>Densità: 0,7 g/cm³ Lavorazione: buona lavorabilità, giunzioni resistenti Impieghi: mobili, compensati</p> <p>CARATTERISTICHE MECCANICHE:</p> <p>Modulo di elasticità (N/mm2) : 13000 Carico di rottura a trazione (N/mm2): 95 Carico di rottura a compressione assiale (N/mm2): 60 Carico di rottura a flessione (N/mm2): 120 Carico di rottura a taglio(N/mm2): 6</p>	<p>PRT_17, PRT_18, PRT_19, PRT_20, PRT_21, PRT_22, PRT_23, PRT_24, PRT_25, PRT_26, PRT_27, PRT_28, PRT_29, PRT_30</p>

MATERIALE	DESCRIZIONE	COMPONENTI
<p>Gomma SBR</p> 	<p>Il copolimero stirene-butadiene (o gomma SBR, dall'inglese Styrene Butadiene Rubber) è un elastomero costituito da unità monomeriche di stirene e butadiene. Ha buona resistenza all'abrasione e stabilità all'invecchiamento. L'SBR è stabile nei confronti di sostanze quali olii minerali, grassi e idrocarburi, acidi e basi organiche e inorganiche. GRS e Buna S sono altri nomi commerciali di uso comune.</p> <p>È la gomma allo stirene industrialmente più importante, viene utilizzato principalmente per produrre pneumatici, tubi, tacchi e soles di scarpe, guarnizioni, piste di atletica leggera.</p> <p>È stato scelto questo materiale perchè resistente all'abrasione ed è uno dei principali materiali utilizzati per la produzione di pneumatici per giocattoli.</p> <p>CARATTERISTICHE:</p> <p>Densità: 1,23 g/cm³ Carico di rottura a trazione (MPa): 5-20 Conducibilità termica [W/(m·K)]: 0,17</p>	<p>PRT_7, PRT_9</p>

BASE DEL ROBOT

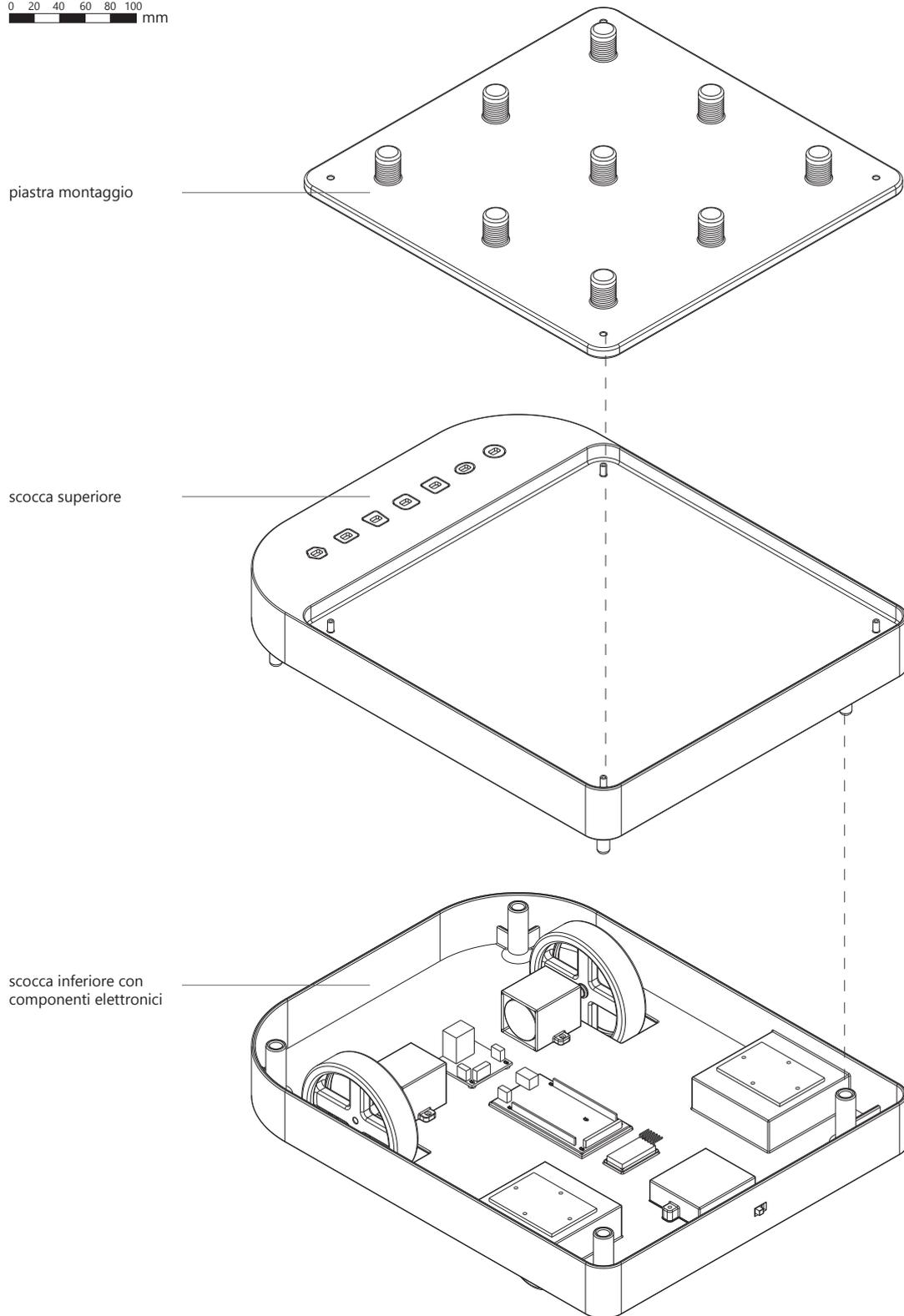
Esploso base del robot

Misure in mm

445x350x155

Scala 1:5

0 20 40 60 80 100 mm

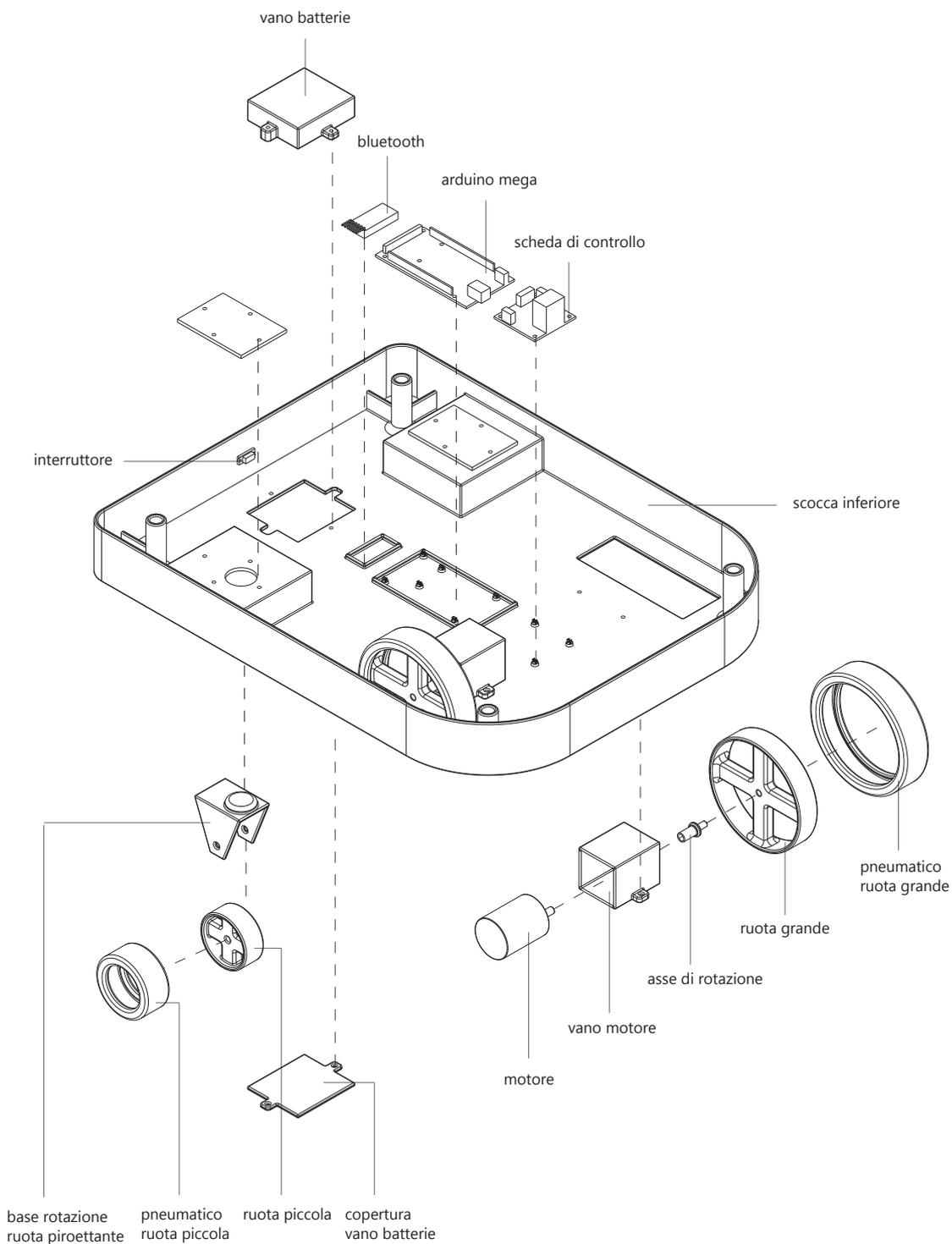


BASE DEL ROBOT

Esploso scocca inferiore con componenti elettronici

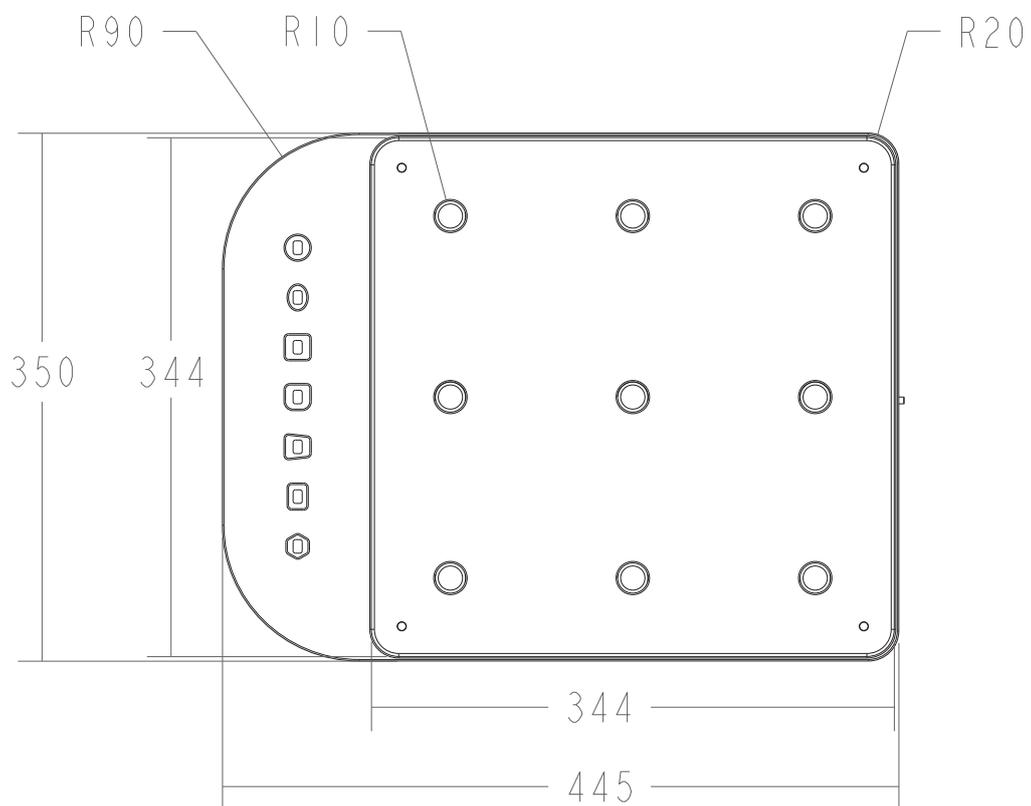
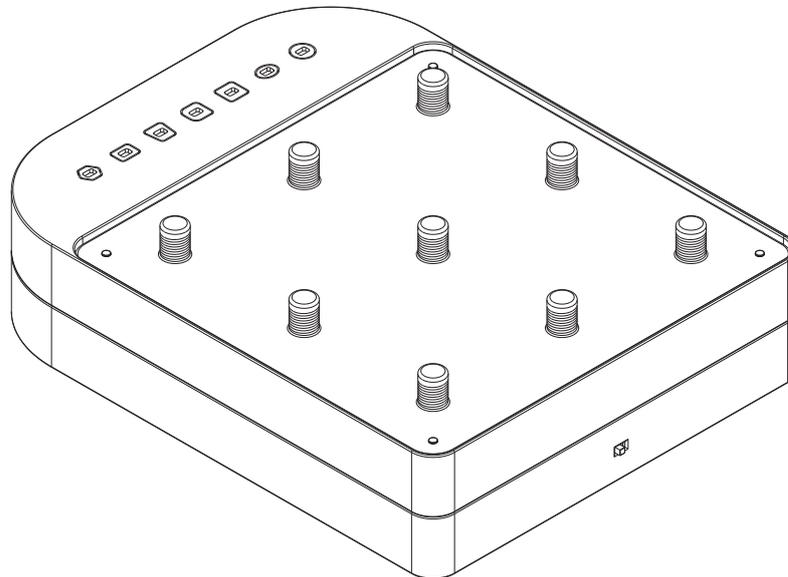
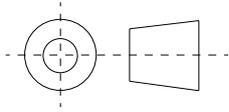
Misure in mm

Scala 1:5



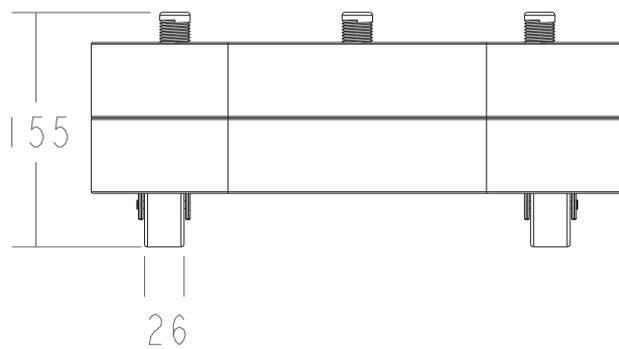
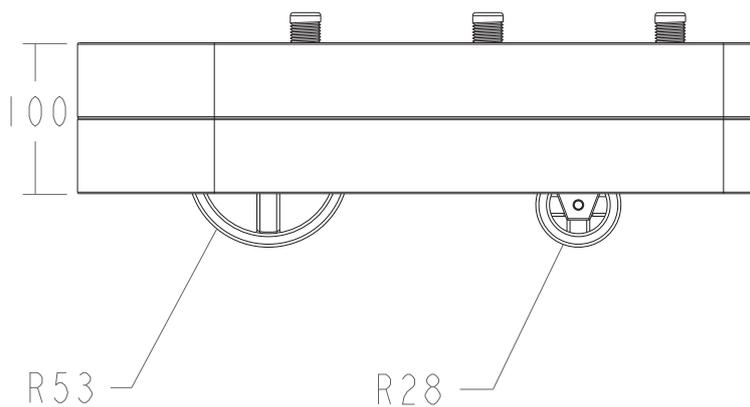
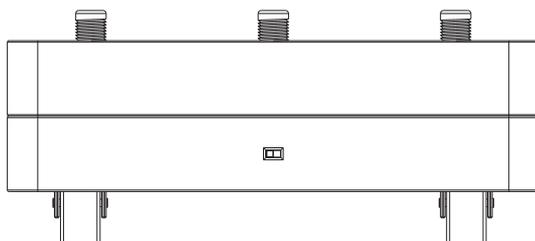
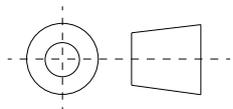
BASE DEL ROBOT

Misure in mm
Scala 1:5



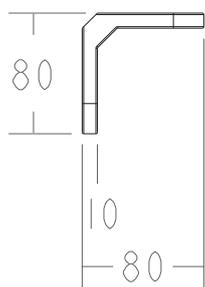
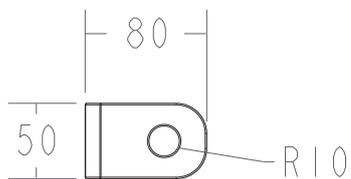
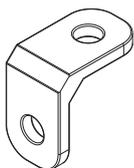
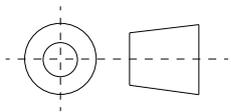
BASE DEL ROBOT

Misure in mm
Scala 1:5



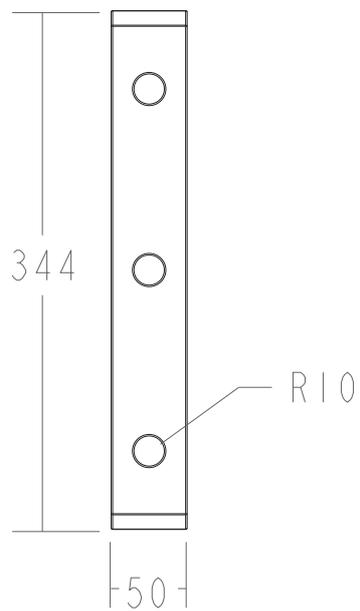
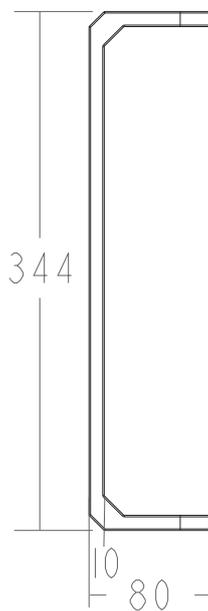
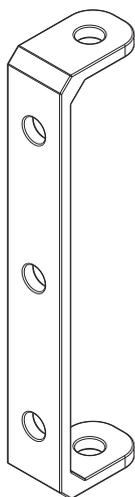
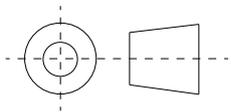
CONNETTORE SINGOLO

Misure in mm
Scala 1:5



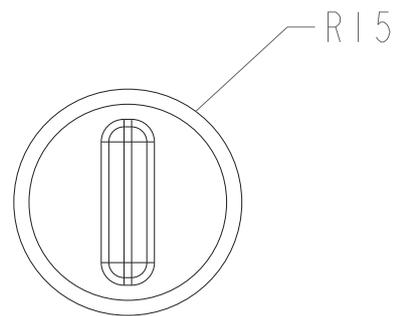
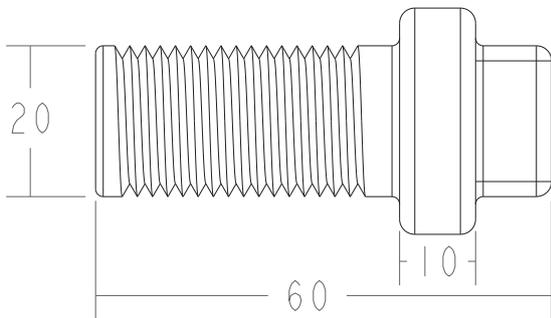
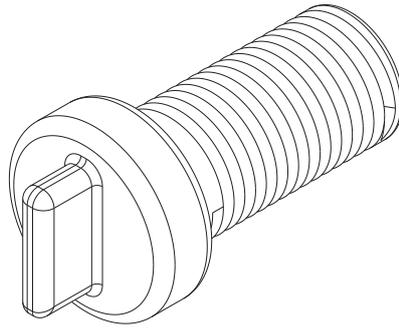
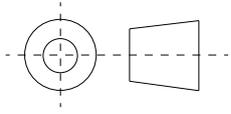
CONNETTORE DOPPIO

Misure in mm
Scala 1:5



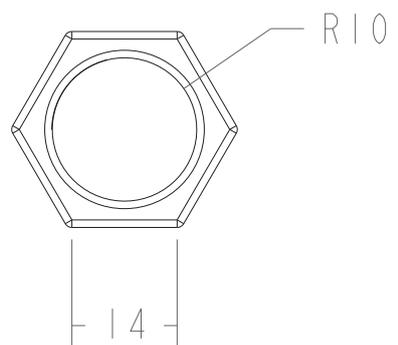
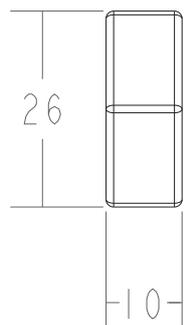
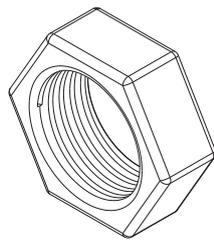
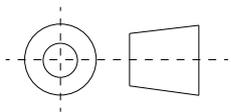
VITE

Misure in mm
Scala 1:1



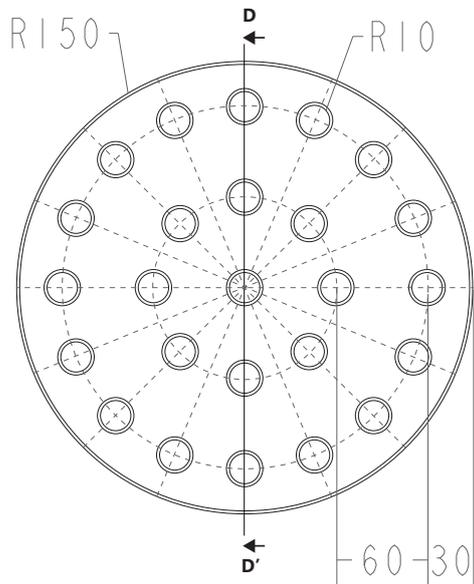
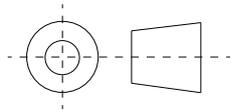
BULLONE

Misure in mm
Scala 1:1



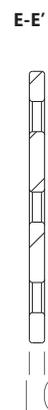
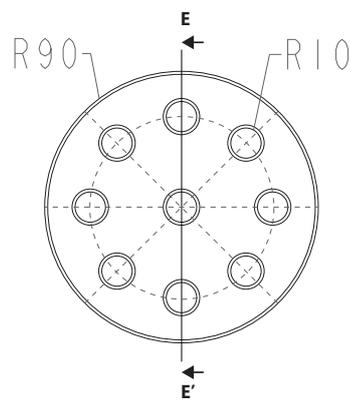
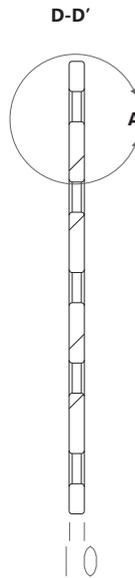
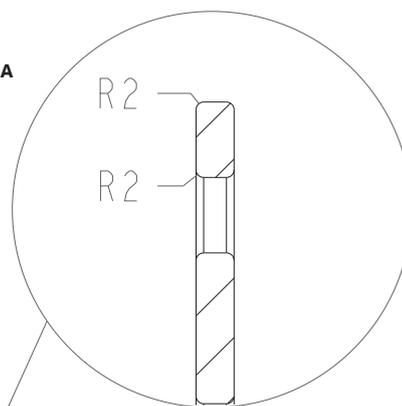
CERCHIO GRANDE -PICCOLO

Misure in mm
Scala 1:5



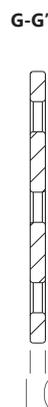
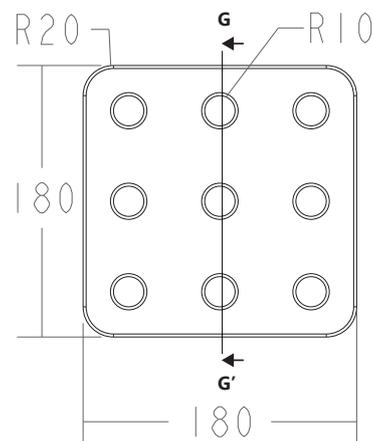
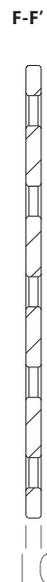
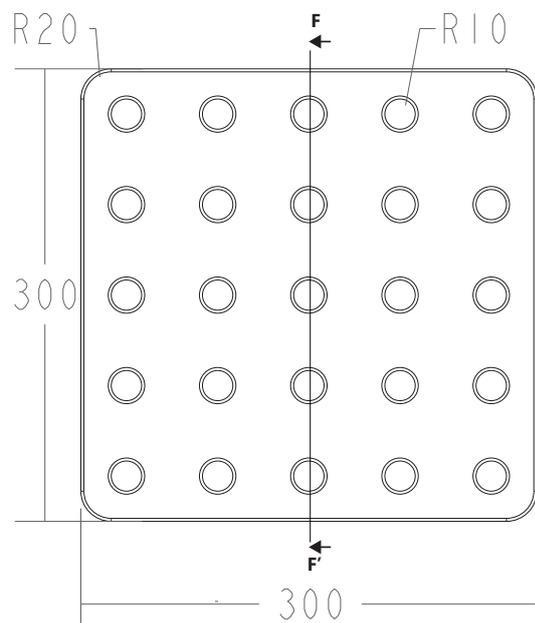
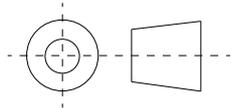
DETTAGLIO A

Scala 1:2



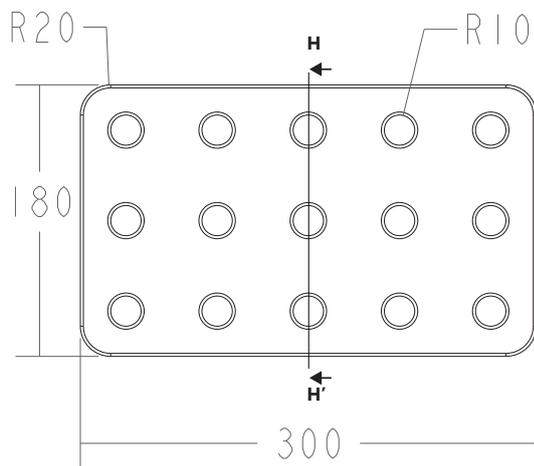
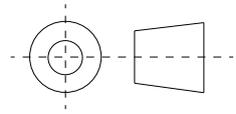
QUADRATO GRANDE -PICCOLO

Misure in mm
Scala 1:5

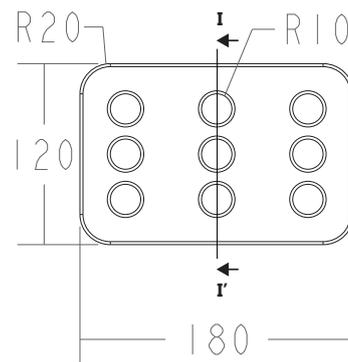


RETTANGOLO GRANDE -PICCOLO

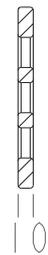
Misure in mm
Scala 1:5



H-H'

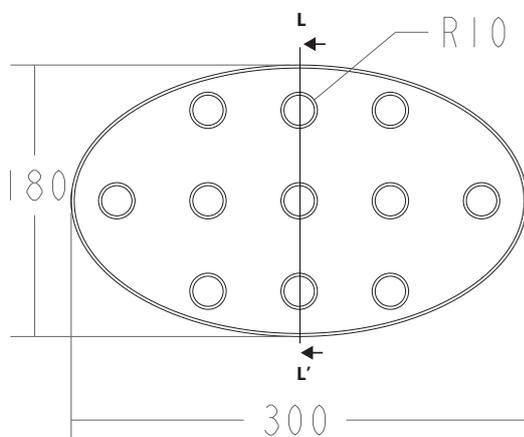
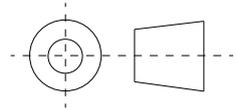


I-I'

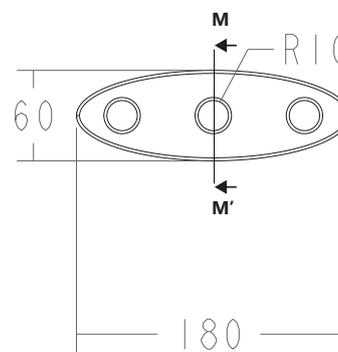
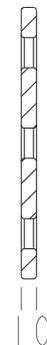


OVALE GRANDE -PICCOLO

Misure in mm
Scala 1:5



L-L'

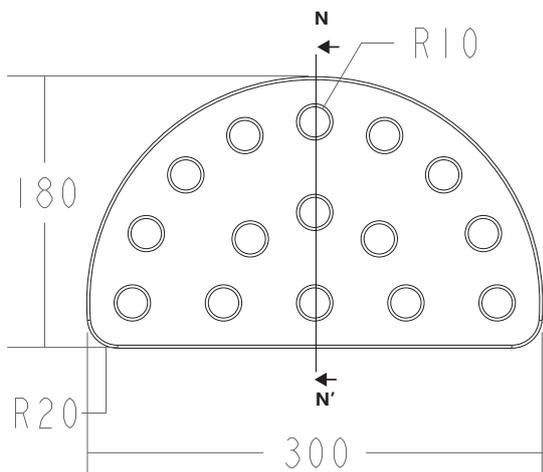
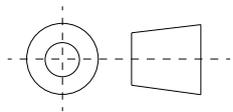


M-M'

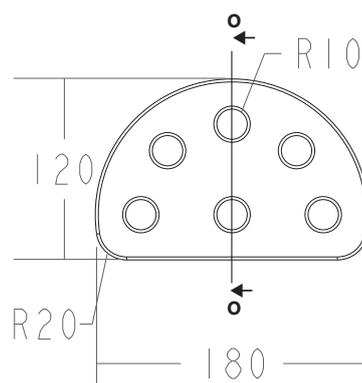
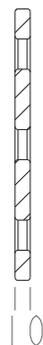


SEMICERCHIO GRANDE -PICCOLO

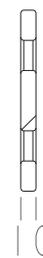
Misure in mm
Scala 1:5



N-N'

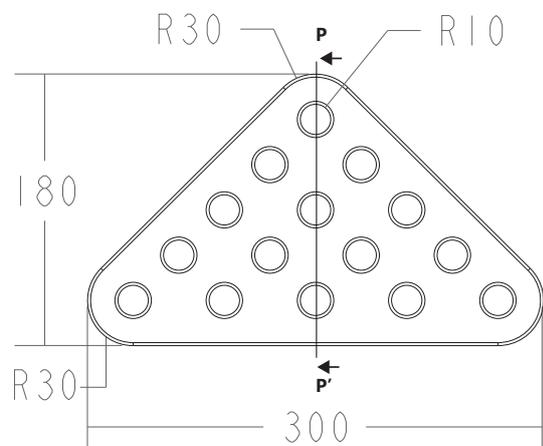
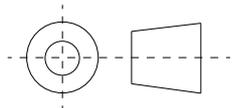


o-o'

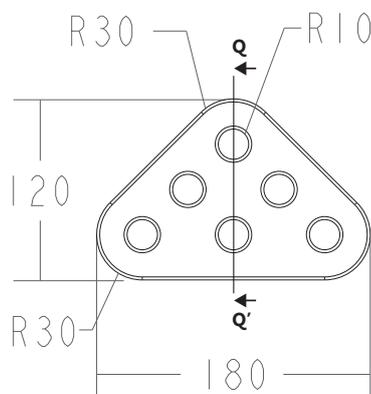
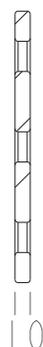


TRIANGOLO GRANDE -PICCOLO

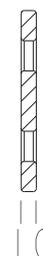
Misure in mm
Scala 1:5



P-P'

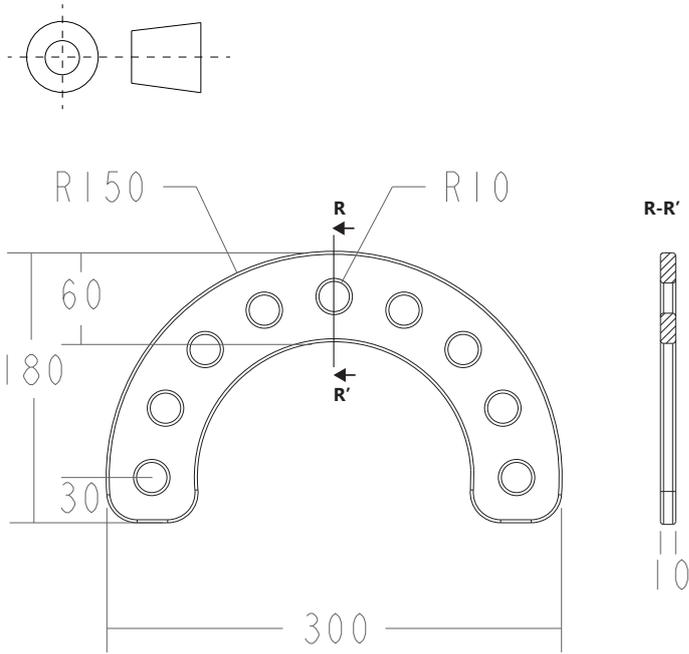


q-q'



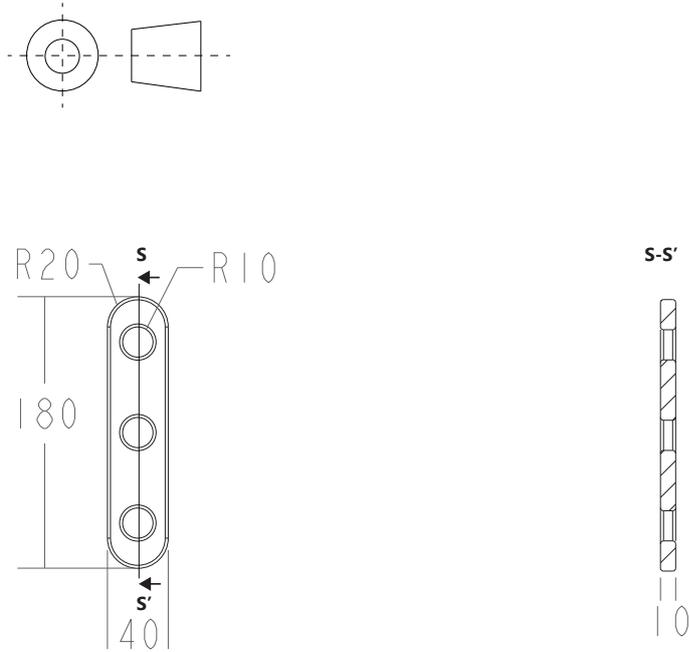
ARCO

Misure in mm
Scala 1:5



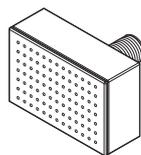
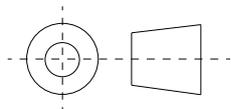
BARRA

Misure in mm
Scala 1:5

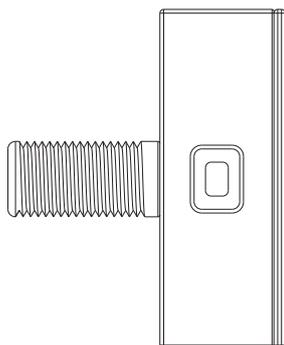
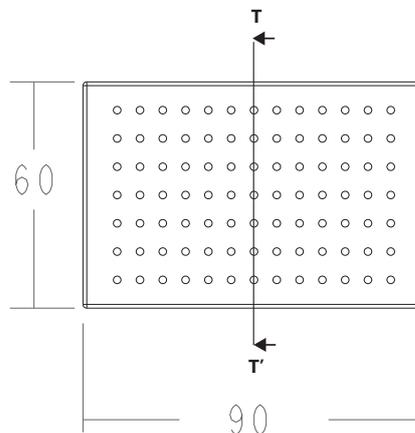
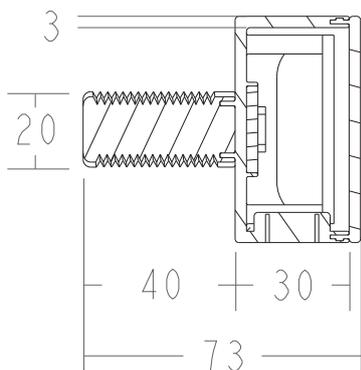


ATTUATORE RIPRODUTTORE AUDIO

Misure in mm
Scala 1:2

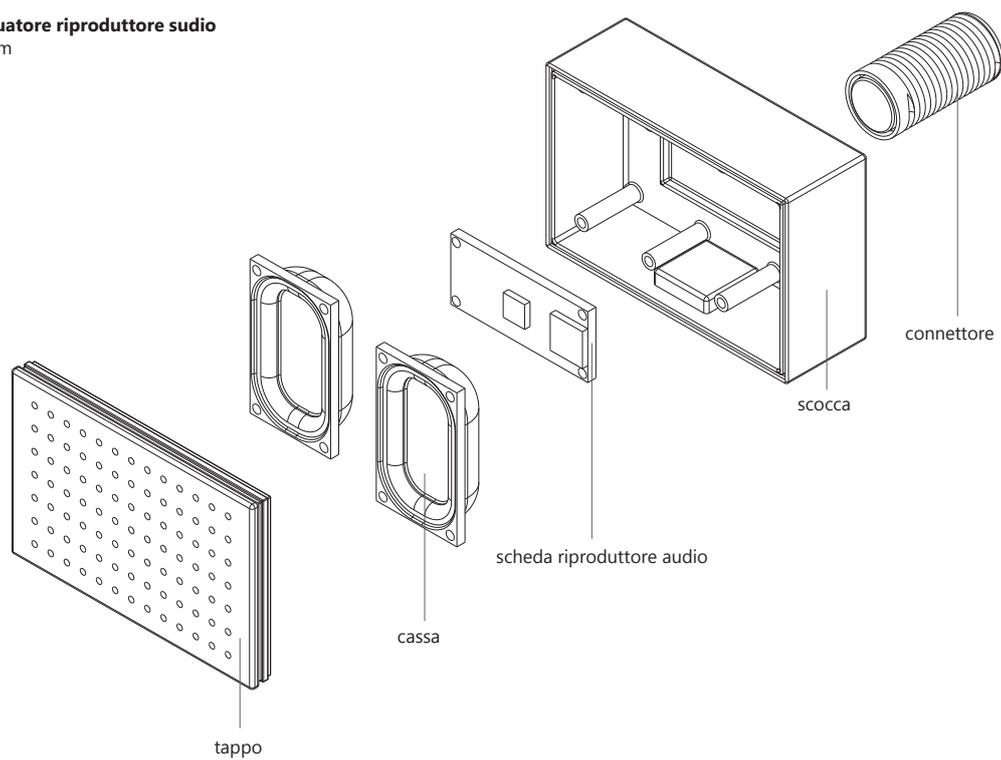


T-T'
Incastro scocca-tappo



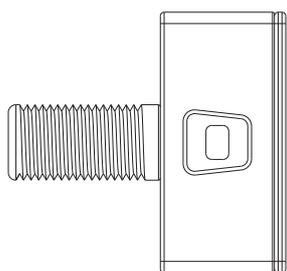
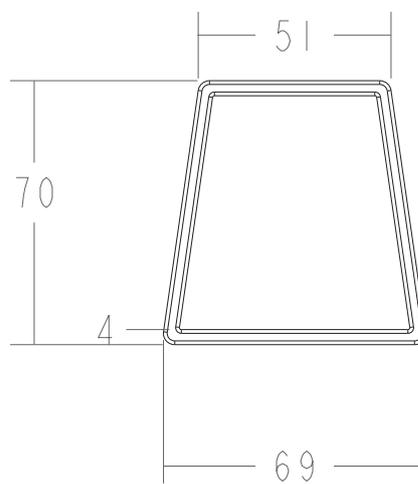
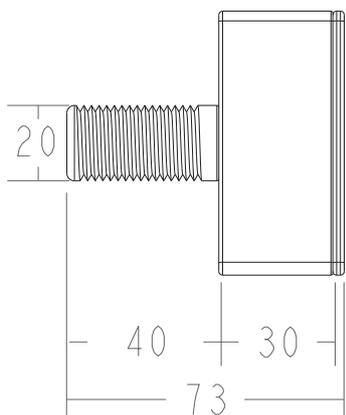
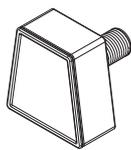
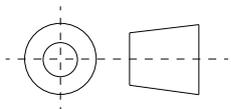
Esploso attuatore riproduttore audio

Misure in mm
Scala 1:2



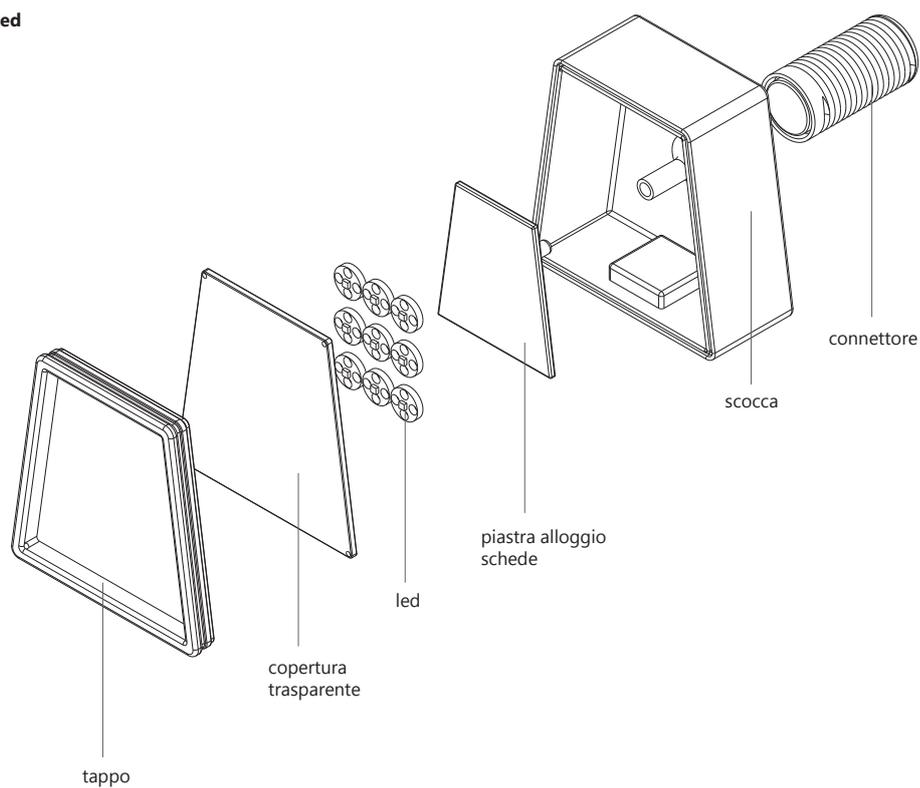
ATTUATORE LED

Misure in mm
Scala 1:2



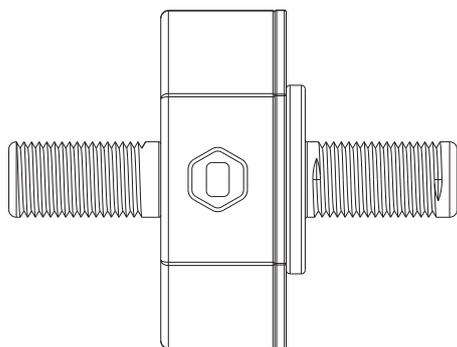
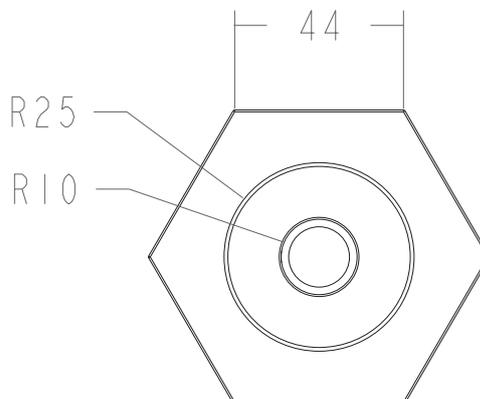
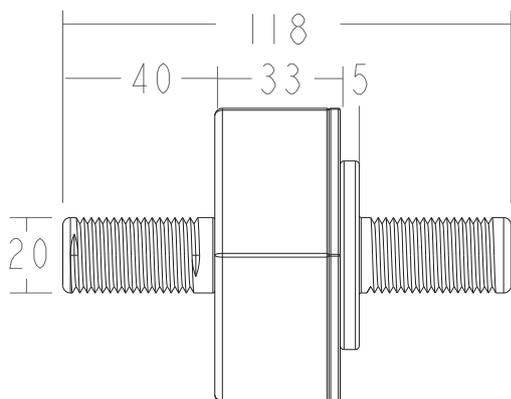
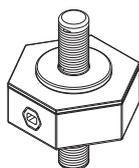
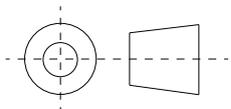
Esplso attuatore led

Misure in mm
Scala 1:2



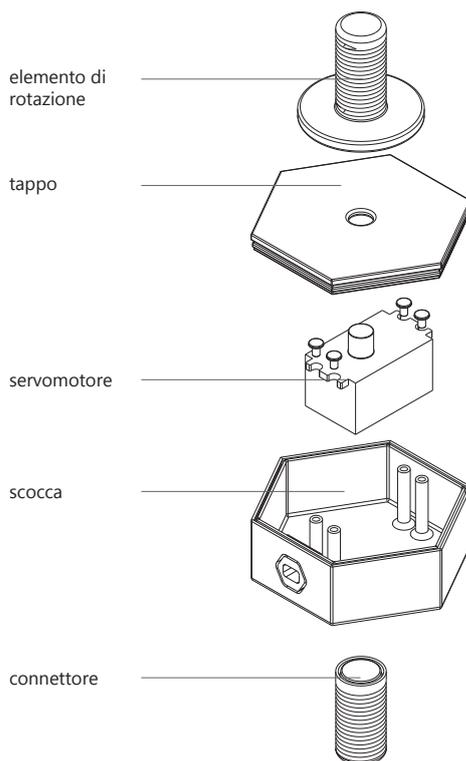
ATTUATORE SERVOMOTORE

Misure in mm
Scala 1:2



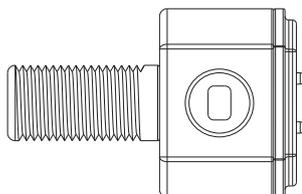
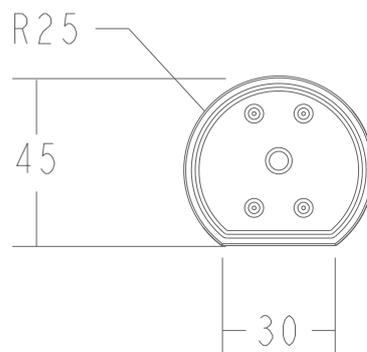
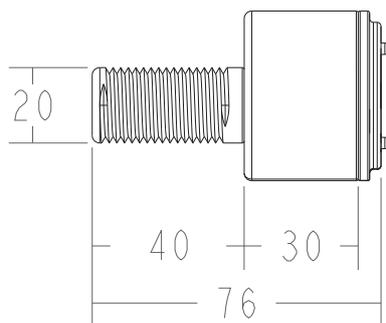
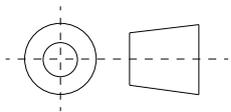
Esplso attuatore servomotore

Misure in mm
Scala 1:3



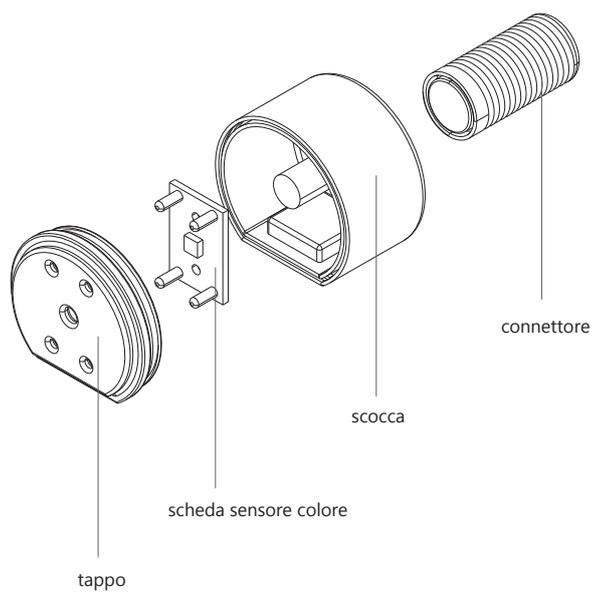
SENSORE COLORE

Misure in mm
Scala 1:2



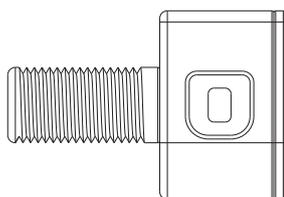
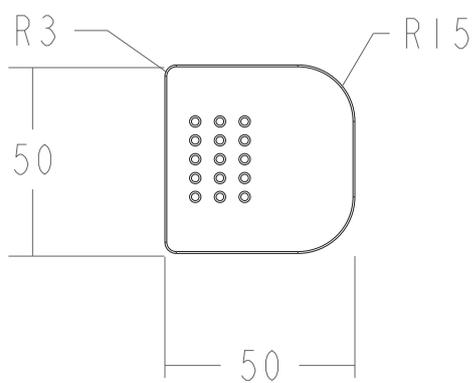
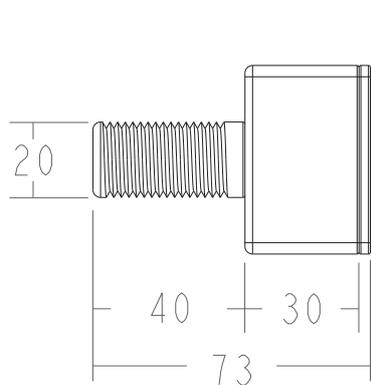
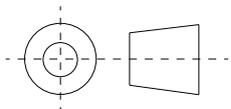
Esplso sensore colore

Misure in mm
Scala 1:2



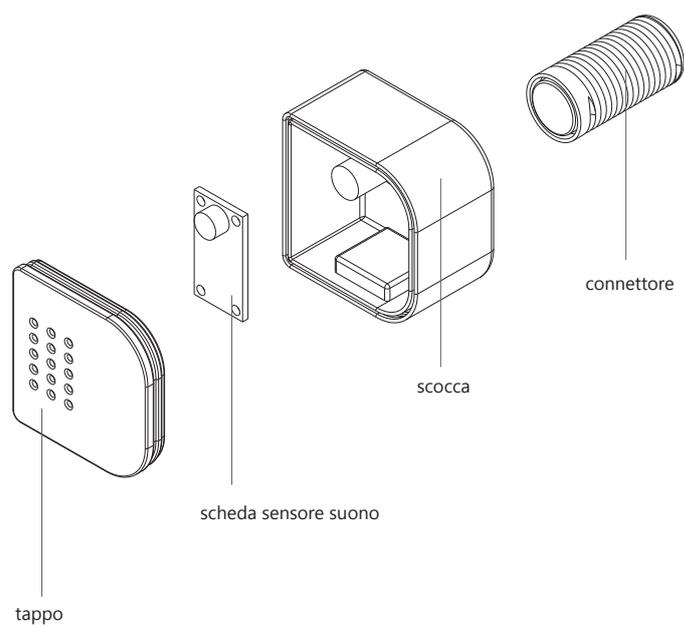
SENSORE SUONO

Misure in mm
Scala 1:2



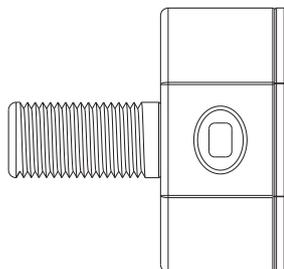
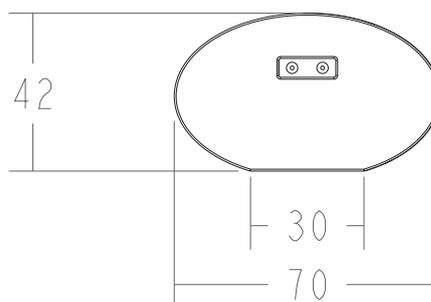
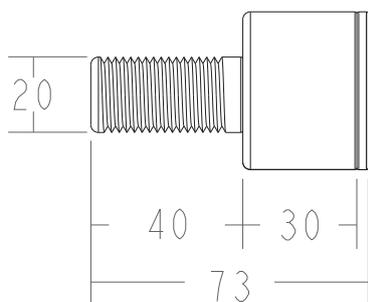
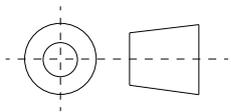
Esplso sensore suono

Misure in mm
Scala 1:2



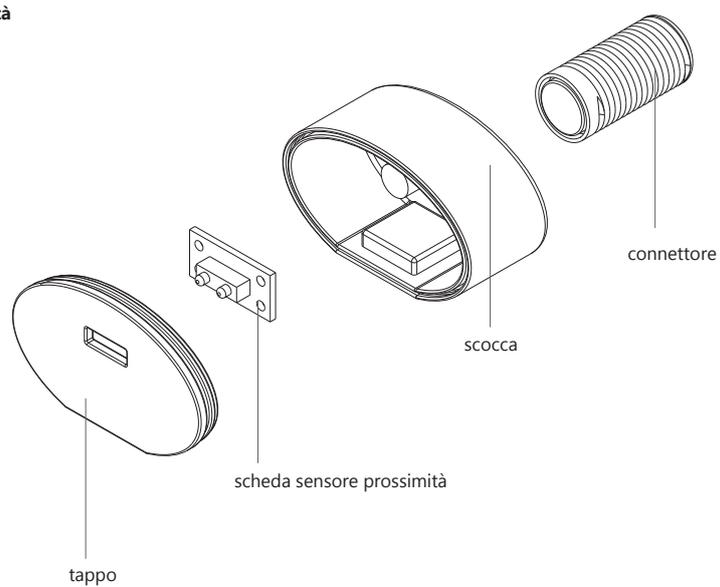
SENSORE PROSSIMITÀ

Misure in mm
Scala 1:2



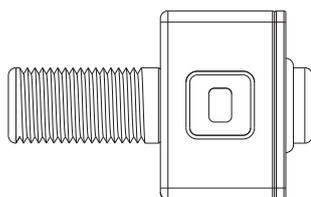
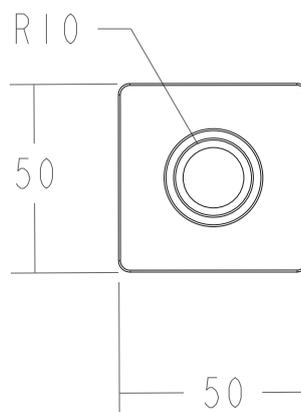
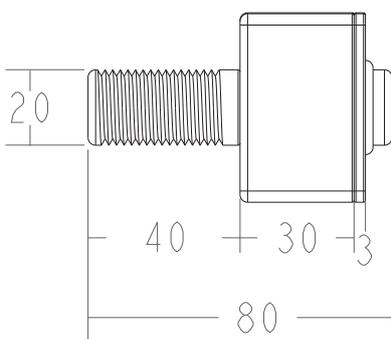
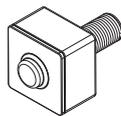
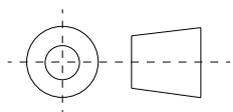
Esplso sensore prossimit 

Misure in mm
Scala 1:2



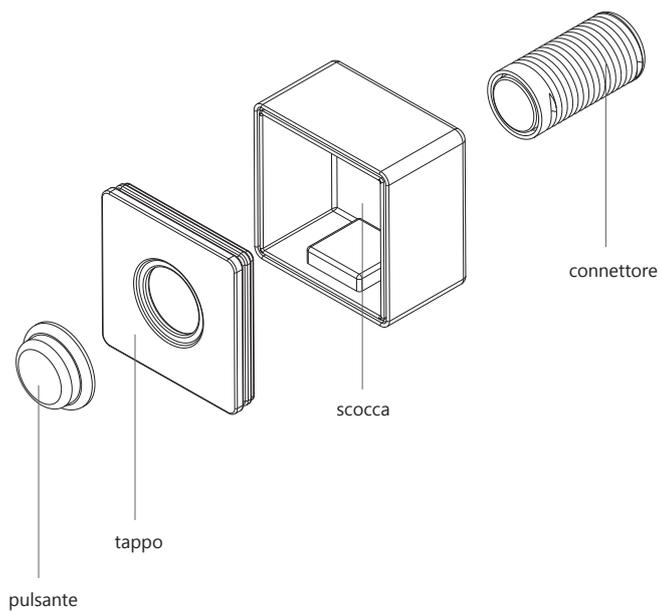
SENSORE PULSANTE

Misure in mm
Scala 1:2



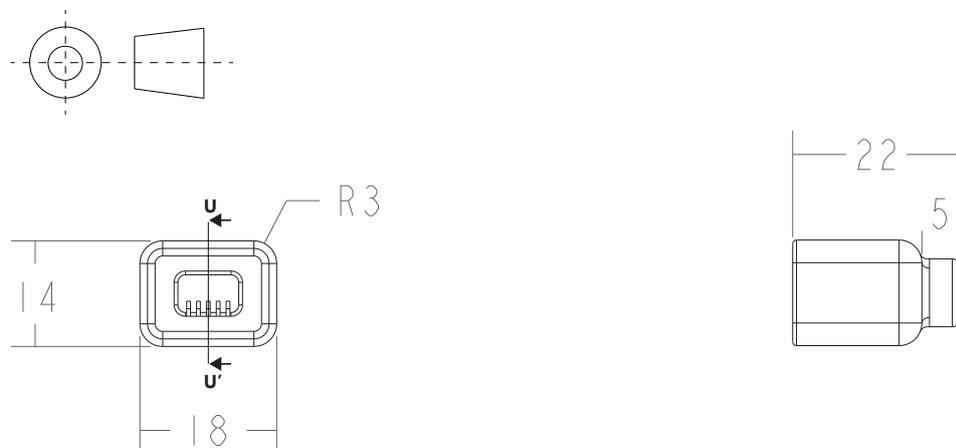
Esplso sensore pulsante

Misure in mm
Scala 1:2

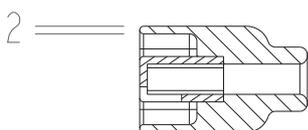


CONNETTORE ELETTRICO: RJ11 PER ATTUATORE RIPRODUTTORE AUDIO

Misure in mm
Scala 1:1

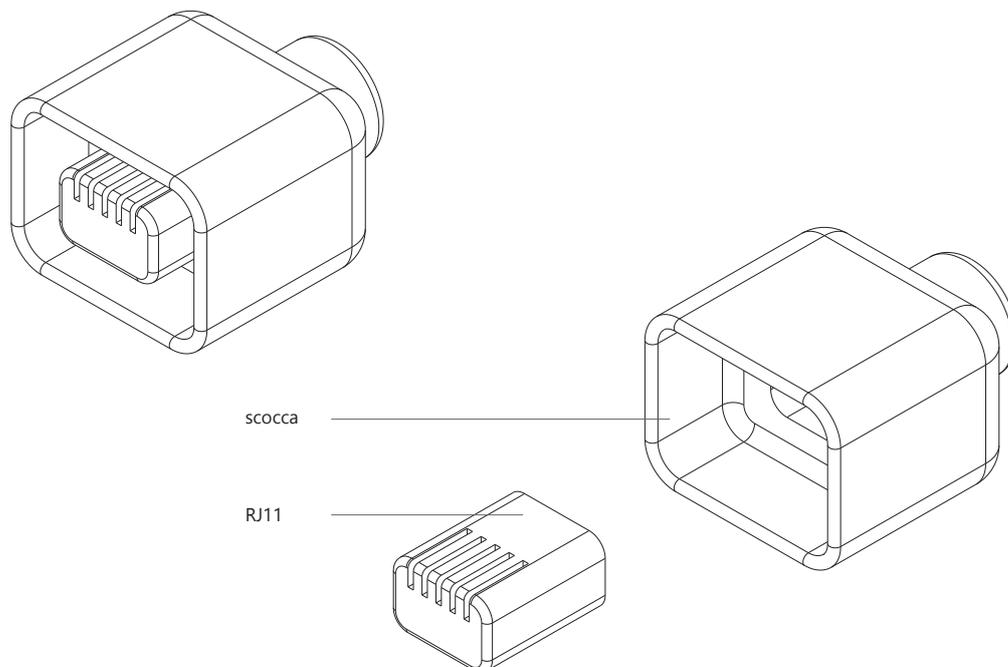


U-U'



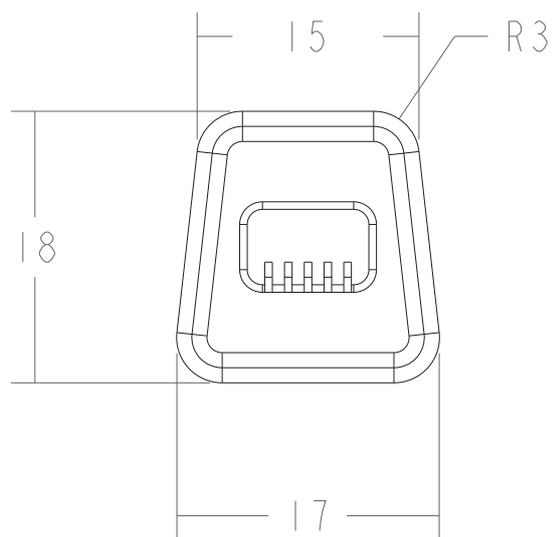
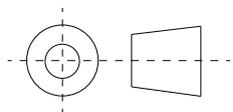
Esplso connettore elettrico per riproduttore audio

Misure in mm
Scala 2:1



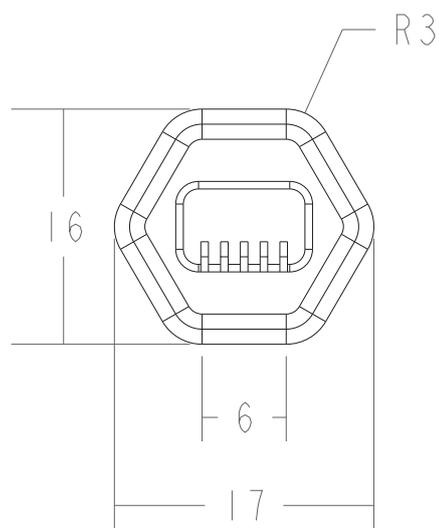
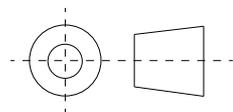
**CONNETTORE ELETTRICO: RJ11
PER ATTUATORE LED**

Misure in mm
Scala 2:1



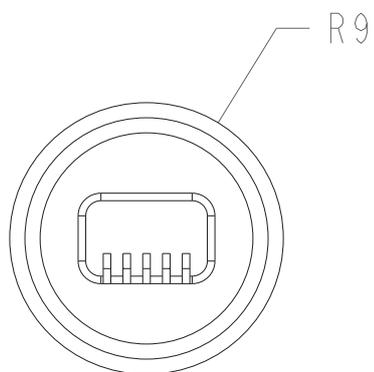
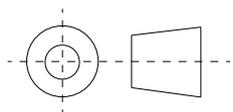
**CONNETTORE ELETTRICO: RJ11
PER ATTUATORE SERVOMOTORE**

Misure in mm
Scala 2:1



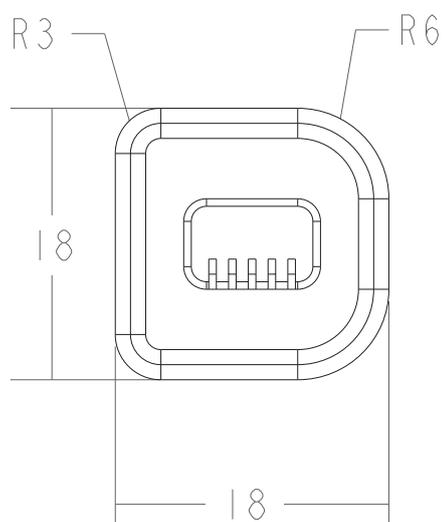
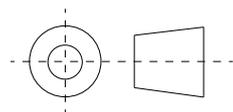
**CONNETTORE ELETTRICO: RJ11
PER SENSORE COLORE**

Misure in mm
Scala 2:1



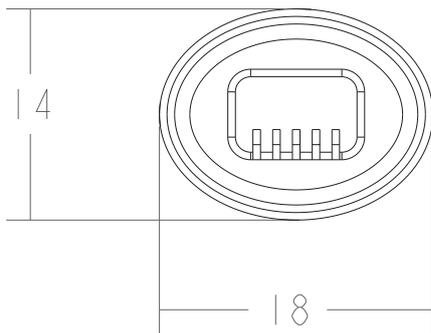
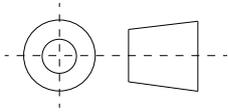
**CONNETTORE ELETTRICO: RJ11
PER SENSORE SUONO**

Misure in mm
Scala 2:1



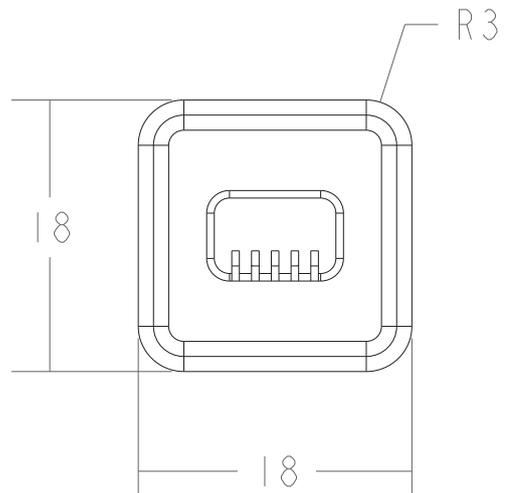
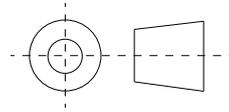
**CONNETTORE ELETTRICO: RJ11
PER SENSORE PROSSIMITÀ**

Misure in mm
Scala 2:1

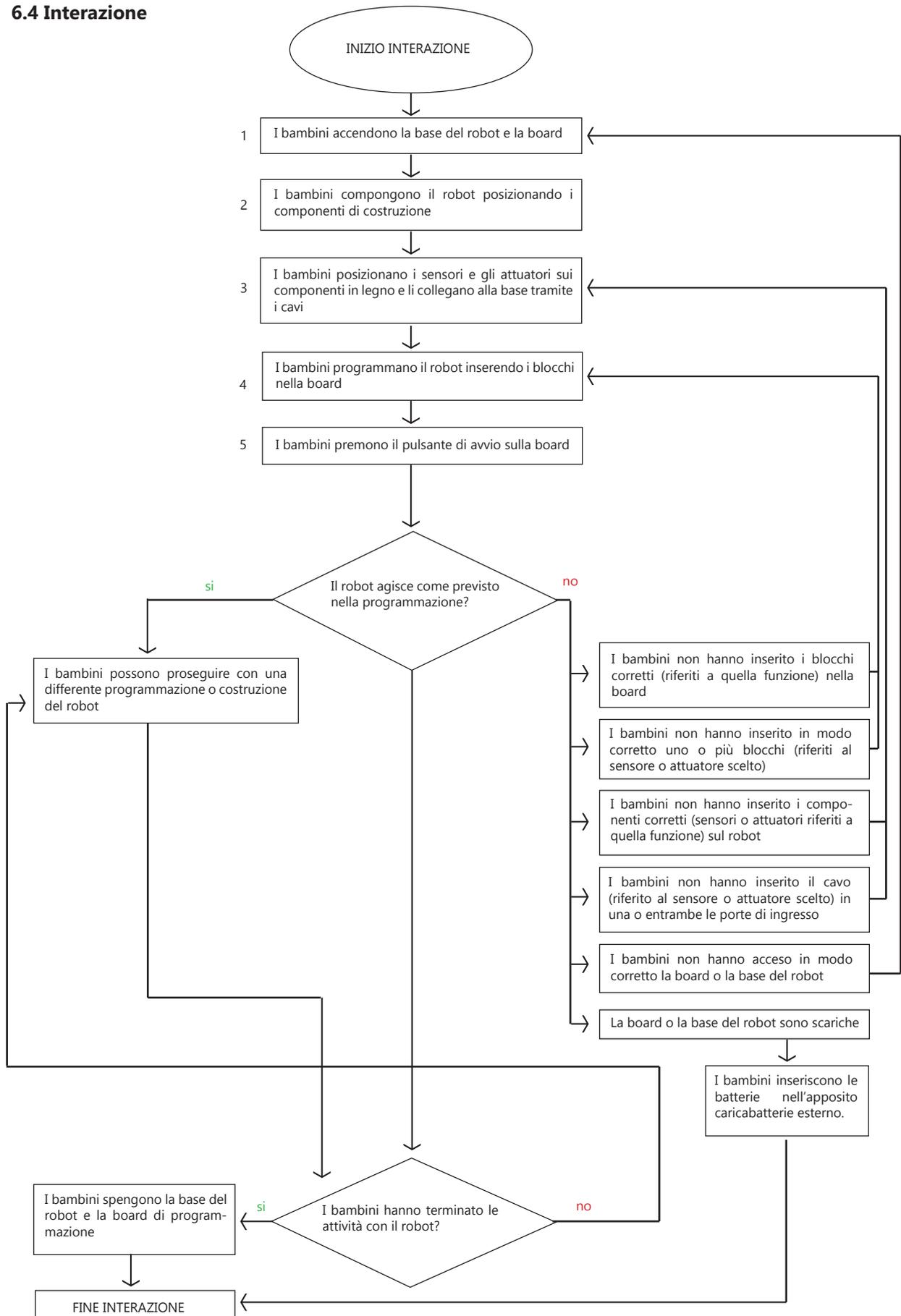


**CONNETTORE ELETTRICO: RJ11
PER SENSORE PULSANTE**

Misure in mm
Scala 2:1

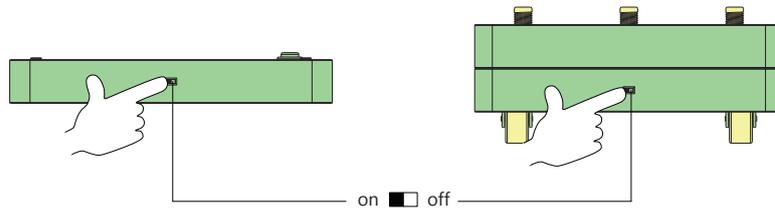


6.4 Interazione

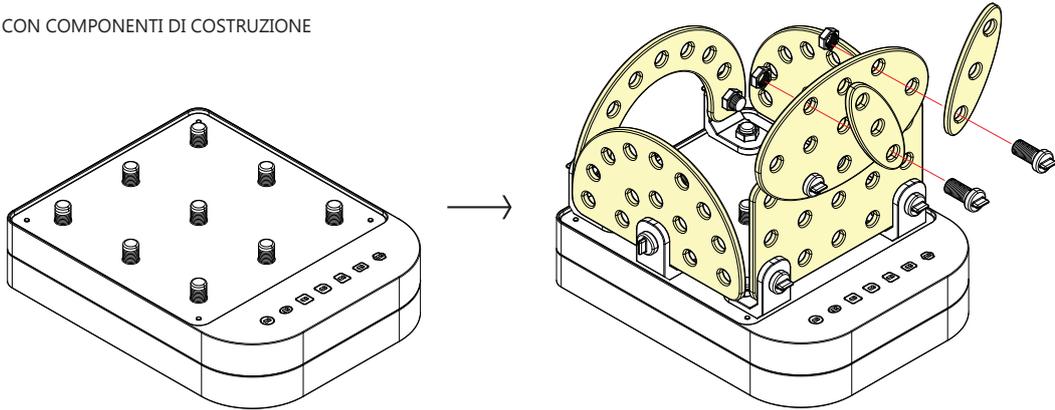


INTERAZIONE IN BREVE :

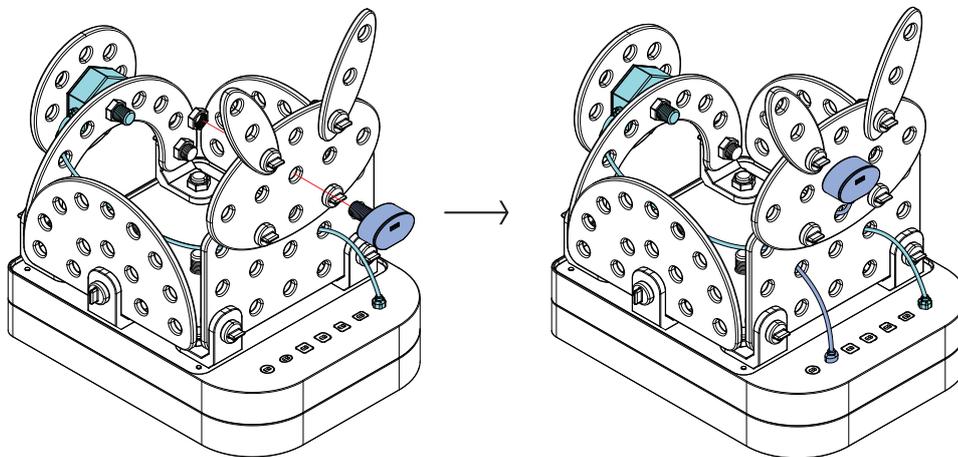
1. ACCENSIONE BOARD E ROBOT



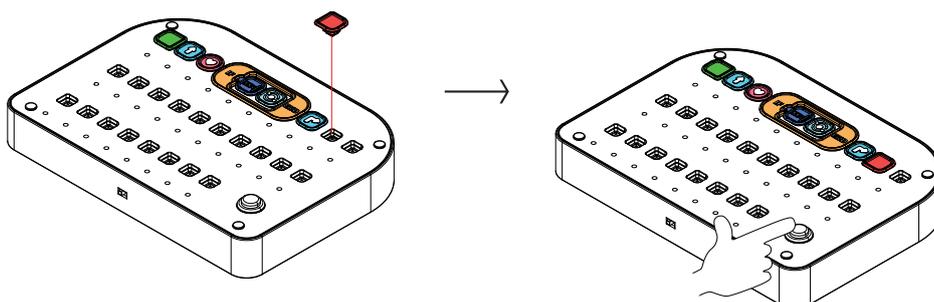
2. MONTAGGIO CON COMPONENTI DI COSTRUZIONE



3. MONTAGGIO SENSORI E ATTUATORI



4. PROGRAMMAZIONE



POSSIBILI ERRORI :

ESEMPIO:

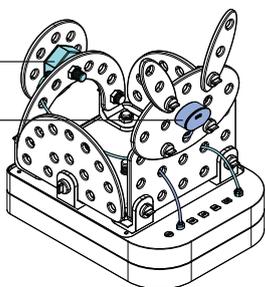
I bambini programmano il robot affinché si muova a destra, muova un componente, si muova in avanti ed infine al riconoscere di un ostacolo muova nuovamente un componente.

Programmazione e componenti elettronici

CORRETTI

servomotore

sensore
prossimità

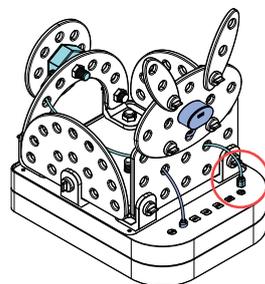


Esempio di codice:

1. Inizio
2. Svolta a destra
3. Rotazione di un componente di 360°
4. Avanti
5. Condizione: se il robot rileverà un oggetto vicino allora un componente verrà ruotato di 360°
6. Fine

Inserimento del cavo di un componente elettronico

ERRATO



Esempio di codice:

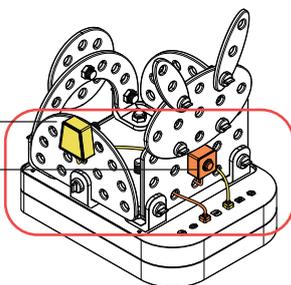
1. Inizio
2. Svolta a destra
3. Rotazione di un componente di 360°
4. Avanti
5. Condizione: se il robot rileverà un oggetto vicino allora un componente verrà ruotato di 360°
6. Fine

Componenti elettronici

ERRATI

led

pulsante

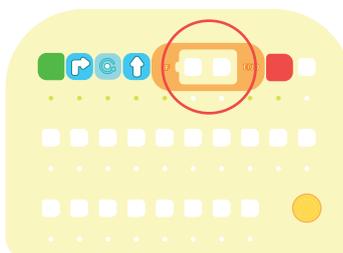
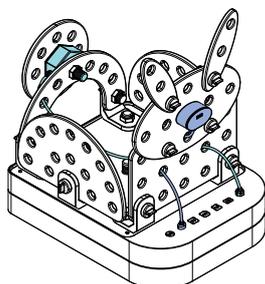


Esempio di codice:

1. Inizio
2. Svolta a destra
3. Rotazione di un componente di 360°
4. Avanti
5. Condizione: se il robot rileverà un oggetto vicino allora un componente verrà ruotato di 360°
6. Fine

Inserimento dei blocchi di programmazione

ERRATO

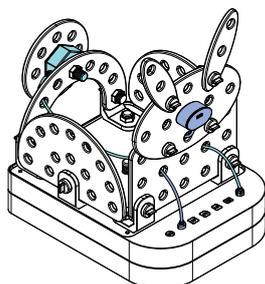


Esempio di codice:

1. Inizio
2. Svolta a destra
3. Rotazione di un componente di 360°
4. Avanti
5. Condizione: in questo caso la condizione non potrà essere risolta per mancanza di funzioni
6. Fine

Programmazione

ERRATA



Esempio di codice:

1. Inizio
2. Svolta a destra
3. Accensione luce gialla
5. Condizione: se verrà premuto il pulsante si accenderà la luce gialla
6. Fine



Lavorare in piccoli gruppi favorisce la socializzazione e la cooperazione. Ogni bambino ha il proprio ruolo nella riuscita di un esercizio.

Gli esercizi svolti grazie al kit robotico aiutano la maestra nella spiegazione di alcuni concetti basilari come ad esempio i colori, le distanze o le grandezze e alcuni concetti complessi come le relazioni causa-effetto.



6.5 Render

BOARD E BLOCCHI PER LA PROGRAMMAZIONE





COMBINAZIONE 1: CONIGLIO

COMBINAZIONE 2: LUCCIOLA



COMBINAZIONE 3: CANE

COMBINAZIONE 4: GIRAFFA



COMBINAZIONE 5: ORSO

6.6 Prototipo funzionale

Il prototipo funzionale (Figura 86) è stato programmato in linguaggio C++ e per la sua realizzazione sono state scelte interazioni di esempio per comprendere il funzionamento della programmazione tangibile e i tempi di azione. Le interazioni scelte sono:

1. Condizione: se il pulsante verrà premuto si accenderà una luce
2. Loop: la luce resterà accesa fino a quando non verrà premuto il pulsante
3. Accensione Luce verde, gialla e rossa

Per la riuscita delle diverse funzioni sono state utilizzate resistenze con valori differenti corrispondenti ai diversi blocchi (Loop, Condizione, Led, Pulsante) e una resistenza nota per creare un ripartitore di tensione e permettere al microcontrollore di leggere le resistenze inserite successivamente in ordine differente :

R1 (resistenza nota): 1K

R2 (Condizione): 2,2K

R3 (Pulsante): 3,3K

R4 (Led verde): 10K

R5 (Led giallo): 5.1K

R6 (Led rosso): 680

R7 (Loop): 6,8K

Tabella 42- Tabella componenti elettronici prototipo funzionale

COMPONENTI	QUANTITÀ	IMMAGINE
Pulsante	2	
Led RGB	1	
Resistenze	9	
Arduino Nano 33 iot	2	
Breadboard	4	
Cavo USB	2	

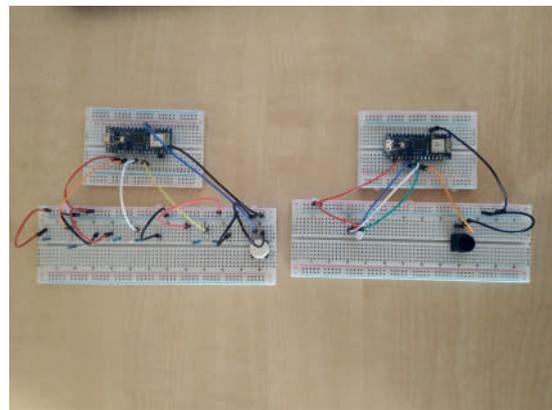
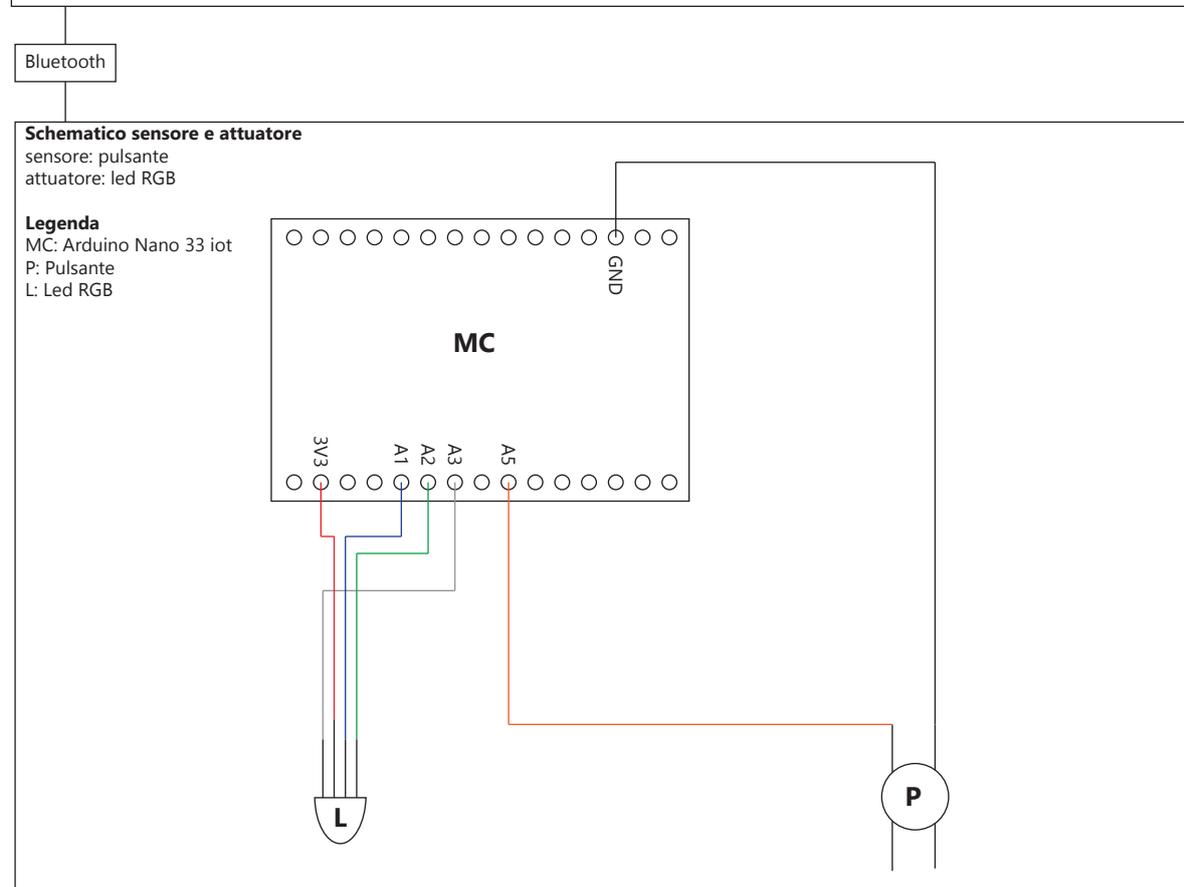
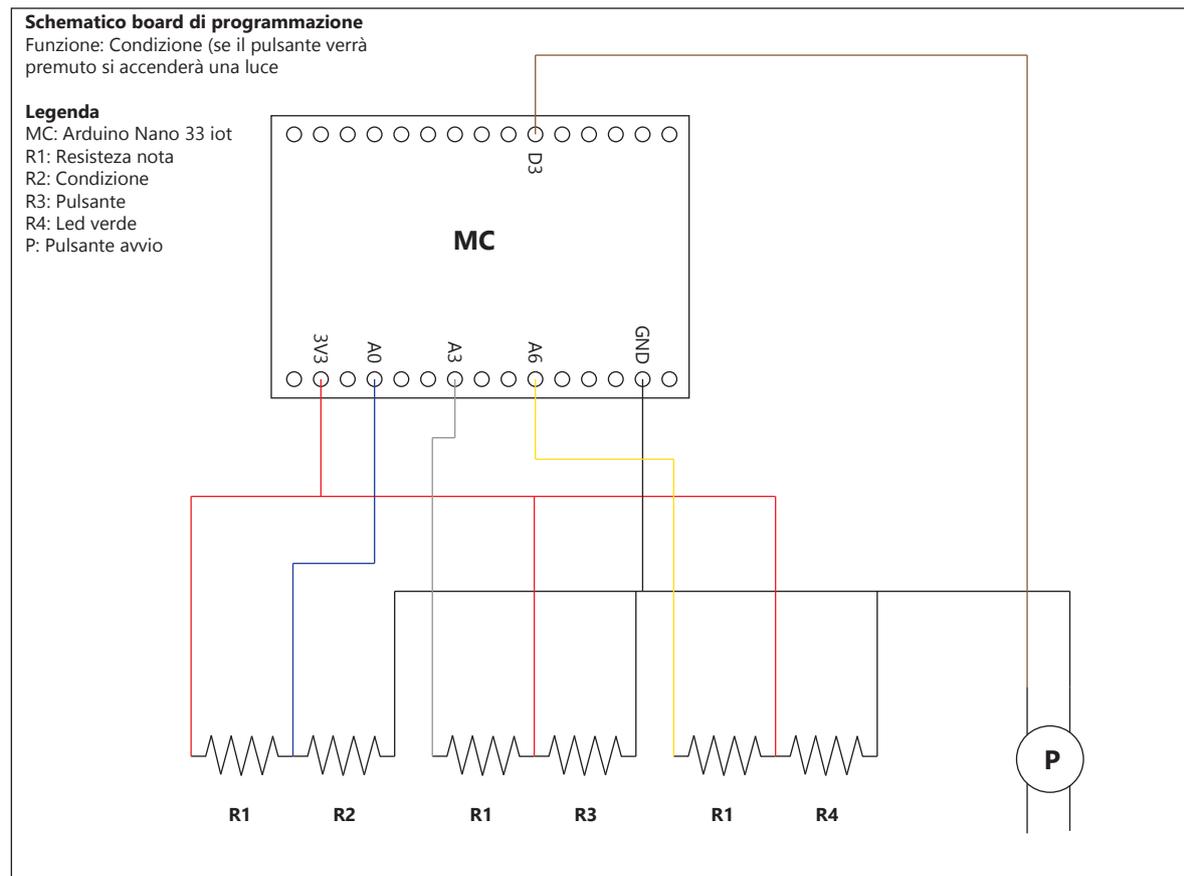
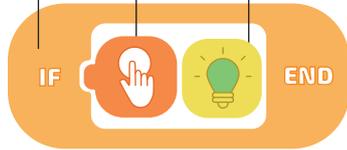
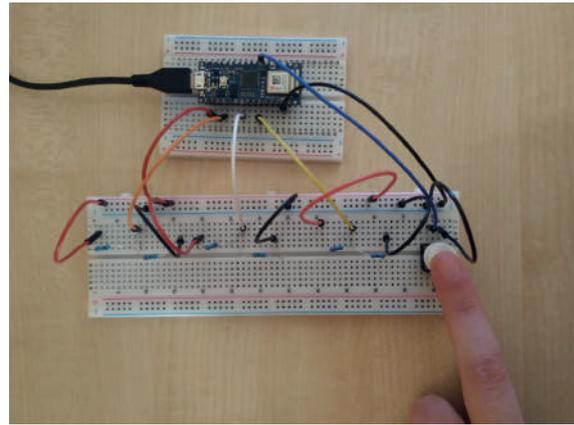
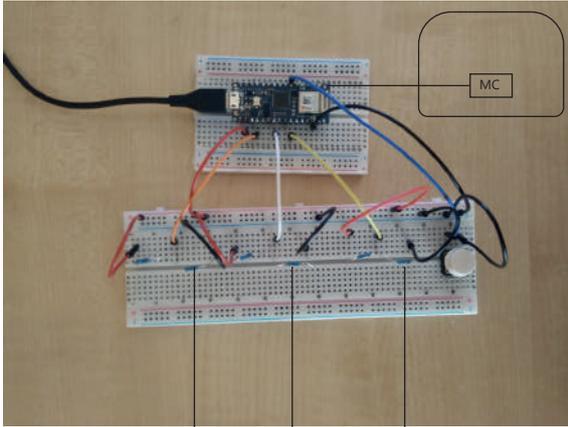


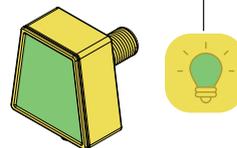
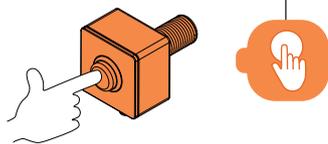
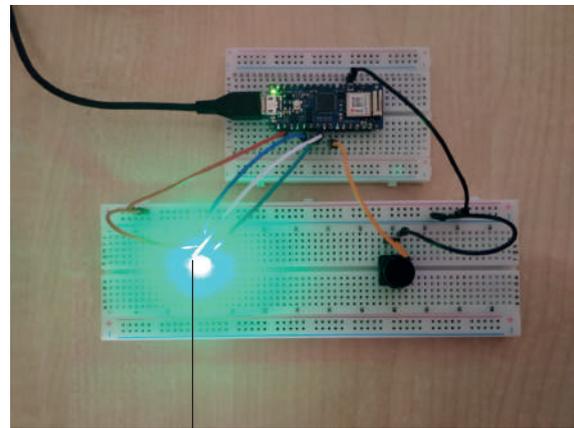
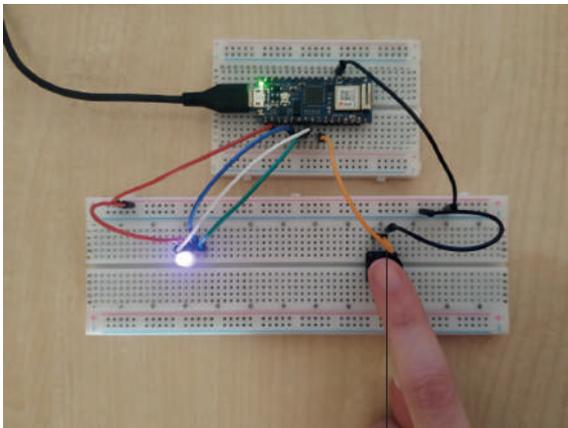
Figura 86- Prototipo funzionale assemblato

SCHEMATICO FUNZIONE 1:





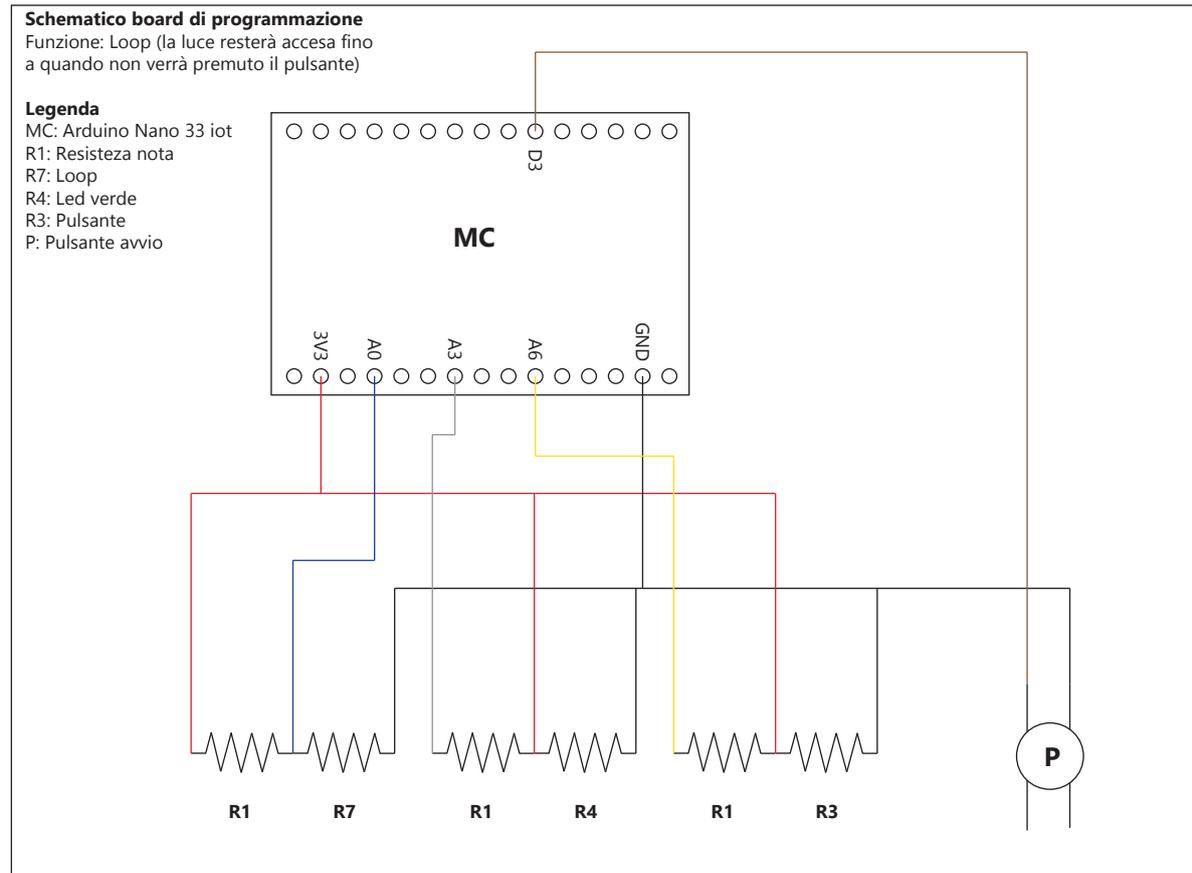
Si inseriscono i comandi affinché premendo il pulsante si accenda la luce verde. Una volta terminata la programmazione viene premuto il pulsante e i comandi vengono inviati al microcontrollore collegato all'attuatore e al sensore.



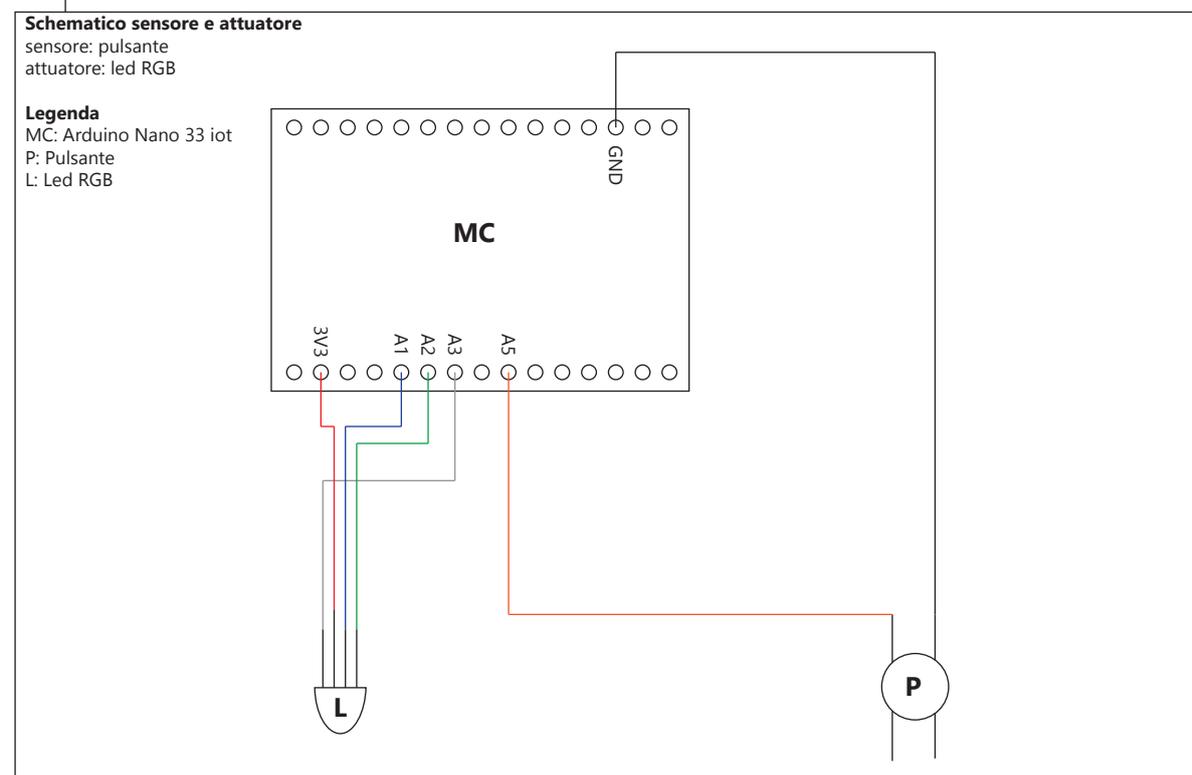
La condizione viene risolta se si preme il pulsante, in quel caso si accende la luce verde per 15 secondi.

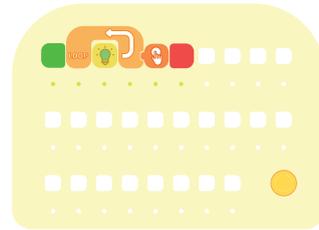
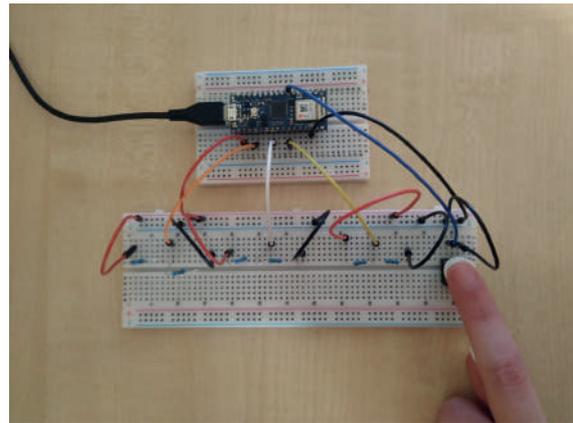
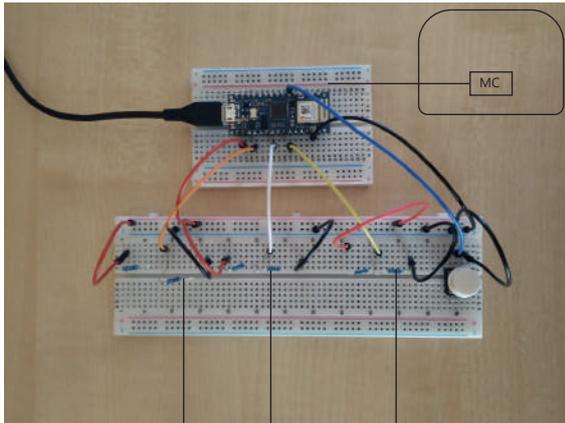
La condizione non viene risolta se non si preme il pulsante entro 30 secondi.

SCHEMATICO FUNZIONE 2:

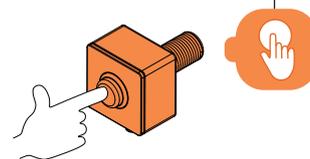
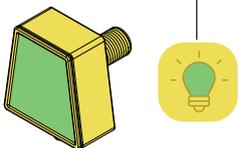
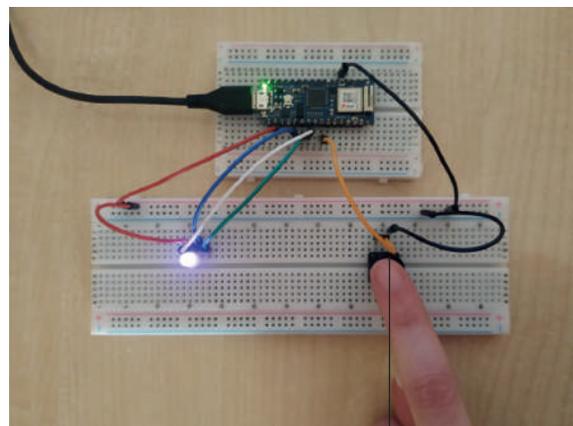
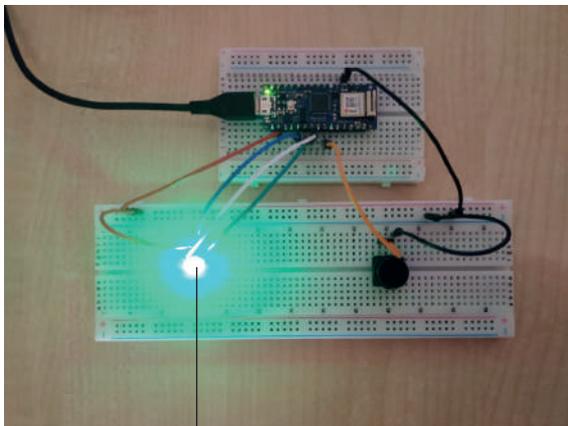


Bluetooth

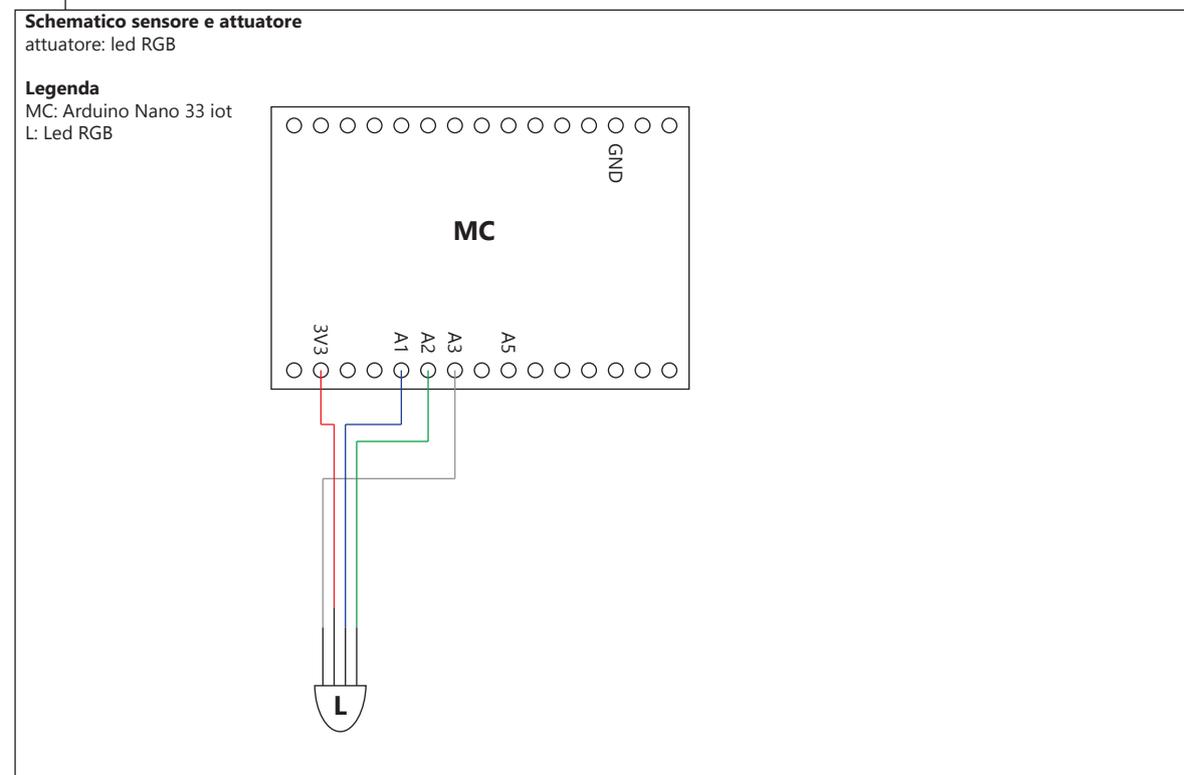
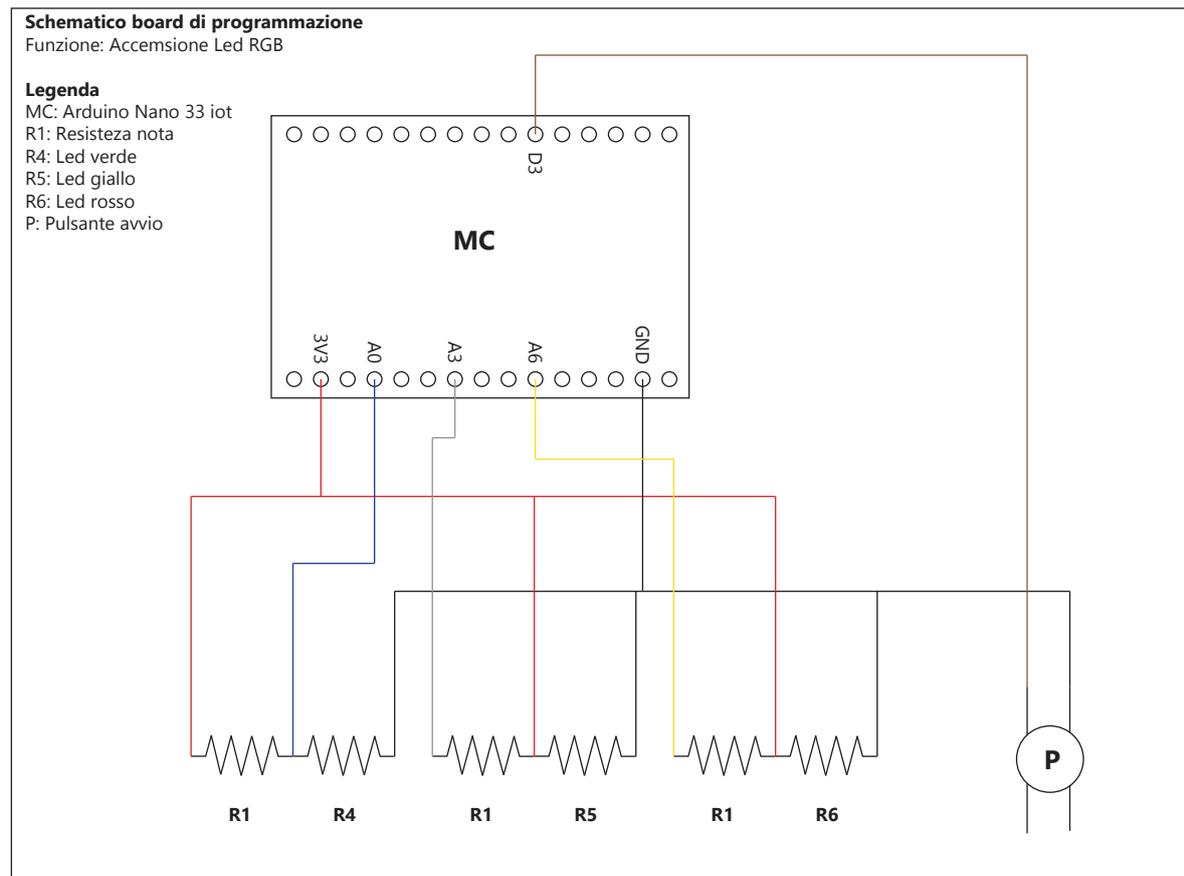


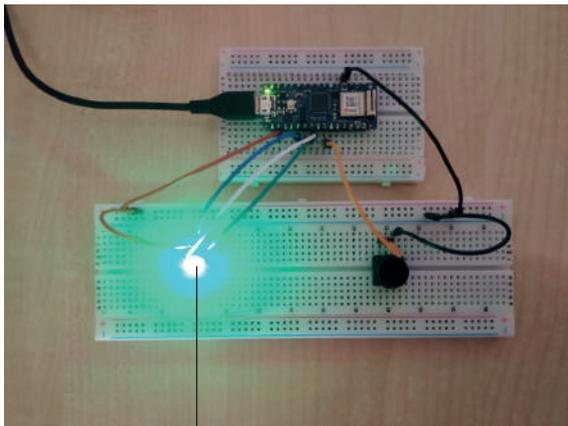
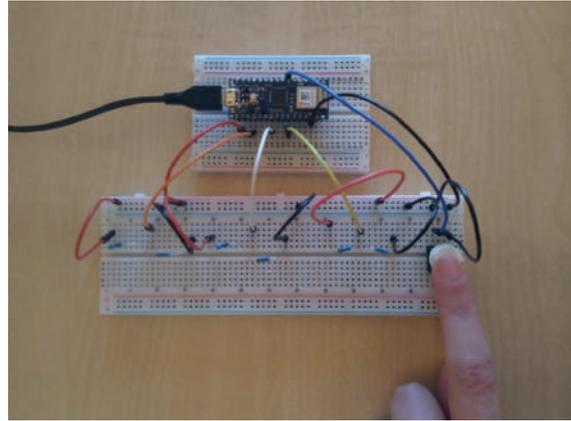
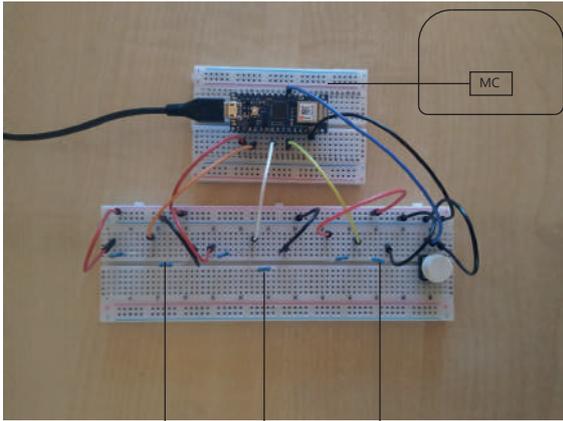


Si inseriscono i comandi affinché la luce verde resti acceso fino a quando non viene premuto il pulsante. Una volta terminata la programmazione viene premuto il pulsante e i comandi vengono inviati al microcontrollore collegato all'attuatore e al sensore.

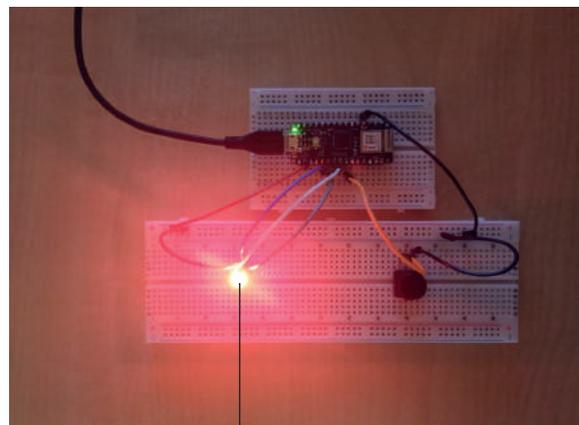
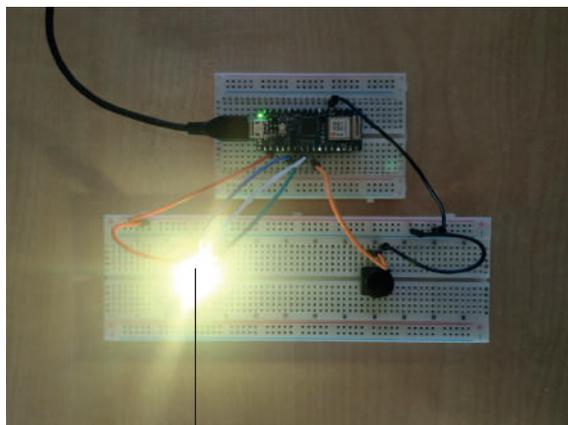
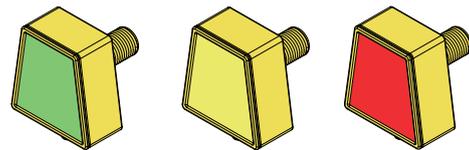


SCHEMATICO FUNZIONE 3:





Si inseriscono i comandi affinché si accenda prima la luce verde poi gialla ed infine rossa. Una volta terminata la programmazione viene premuto il pulsante e i comandi vengono inviati al microcontrollore collegato all'attuatore e al sensore. Ogni luce resta accesa per 15 secondi.



6.7 Conclusioni

Il kit robotico Combo promuove l'apprendimento del Coding nella scuola dell'infanzia e si pone come obiettivi:

- **Permettere ai bambini di esprimersi in modo creativo tramite il Coding.**
- **Migliorare le capacità di problem solving e storytelling.**
- **Aiutare i bambini a sviluppare competenze trasversali come la collaborazione e la comunicazione.**
- **Aiutare i bambini a comprendere il ruolo positivo dell'errore.**
- **Essere un mediatore didattico per l'insegnante.**

La sfida maggiore è stata quella di promuovere il pensiero computazionale con l'insegnamento del Coding in una fascia di età (3-6 anni) molto delicata per l'apprendimento poichè non si hanno abbastanza conoscenze preliminari necessarie alla comprensione dei concetti relativi alla programmazione. Per la progettazione di un kit robotico adeguato si è proceduto facendo uno studio preliminare di ricerca per comprendere meglio lo scenario della scuola dell'infanzia e quali fossero i linguaggi di programmazione per i bambini. Successivamente è stato fatto un approfondimento di quello che è ad oggi lo stato dell'arte dei kit robotici educativi che utilizzano la programmazione tangibile. In quest'ultima fase sono state individuate le diverse modalità di programmazione tangibile e i diversi componenti robotici utilizzati. Combo a differenza dei kit presenti nello stato dell'arte offre una differenziazione maggiore dei comandi di programmazione e dei componenti elettronici, permettendo ai bambini di raggiungere livelli differenti di abilità in base all'età e alle necessità di ogni bambino. Nella fase di progettazione del Kit sono state fatte delle verifiche dimensionali con la realizzazione di prototipi volumetrici e nella fase conclusiva si è proceduto con la prototipazione di alcune funzioni per comprendere l'effettivo funzionamento del kit.

Sicuramente ci sono numerosi aspetti che potrebbero essere approfonditi meglio in studi futuri, il kit si presta alla realizzazione di numerose espansioni sia per i comandi di programmazione sia per i componenti elettronici e di costruzione.

Le espansioni potrebbero ampliare i livelli di abilità facendo sì che questo progetto diventi sempre più un ponte tra la programmazione tangibile per i più piccoli e quella classica in cui è necessario l'utilizzo del computer, rendendo i bambini sempre più consapevoli sull'utilizzo e il funzionamento dei dispositivi elettronici.

SITOGRAFIA E BIBLIOGRAFIA

1. Diego Izzo, articolo "La genesi della conoscenza: il suolo dei processi di assimilazione e accomodamento nello sviluppo cognitivo di Piaget", pubblicato l'11 Settembre 2018, link:<https://www.ebookscuola.com/blog/la-genesi-della-conoscenza-il-ruolo-dei-processi-di-assimilazione-e-accomodamento-nello-sviluppo-cognitivo-di-piaget/>
2. Seymour Papert, libro "Mindstorms: Children, Computers, and Powerful Ideas", 1980
3. Jeannette Wing, articolo "Computational Thinking", pubblicato nel Marzo 2006, link: <https://www.cs.cmu.edu/afs/cs/Web/People/15110-s13/Wing06-ct.pdf>
4. Marc Prensky, "H. Sapiens Digital: From Digital Immigrants and Digital Natives to Digital Wisdom", pubblicato il 3 Gennaio 2009, link: <https://nsuworks.nova.edu/cgi/viewcontent.cgi?article=1020&context=innovate>
5. Report a cura di Openpolis e Osservatorio #conibambini, "Stem, una sfida per l'Italia", pubblicato a Gennaio 2022, link: https://www.openpolis.it/wp-content/uploads/2021/12/report_stem.pdf
6. Ministero dell'Istruzione, dell'Università e della Ricerca, "Iniziazioni Nazionali e Nuovi Scenari", pubblicato il 22 Febbraio 2018, link:<http://www.indicazioninazionali.it/wp-content/uploads/2018/08/Indicazioni-nazionali-e-nuovi-scenari.pdf>
7. Karen Brennan e Mitchel Resnick, ricerca "Nuovi quadri per lo studio e la valutazione del sviluppo del pensiero computazionale", pubblicata nel 2012, link:https://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf
8. CodeWeek, sito ufficiale link:codeweek.eu
9. Code.org, sito ufficiale link: Code.org
10. Programma il futuro, sito ufficiale link: programmailfuturo.it
11. CodeMOOC, sito ufficiale link: codemooc.org
12. Ministero dell'Istruzione Ministero dell'Università e della Ricerca, sezione "Scuola dell'infanzia". link: <https://www.miur.gov.it/scuola-dell-infanzia>
13. Ministero dell'Istruzione, dell'Università e della Ricerca, "Indicazioni Nazionali per il curriculum della scuola dell'infanzia e del primo ciclo d'istruzione", Roma, 2012.
14. Myriam Caratù, "Strategie didattiche: imitative, di ricerca, creative e attive", pubblicato il 15 Ottobre 2020, link: <https://www.orizzontescuola.it/strategie-didattiche-imitative-di-ricerca-creative-e-attive/>
15. Elio Damiano, "La mediazione didattica. Per una teoria dell'insegnamento", Franco Angeli, 2013
16. Marina Umaschi Bers, libro "Coding as a Playground: Programming and Computational Thinking in the Early Childhood Classroom", prima pubblicazione nel 2017 e seconda pubblicazione nel 2020.
17. Tiziana Mascia, libro "I percorsi del lettore: Teoria e buone pratiche per la formazione", pubblicato nel dicembre 2020, link:https://www.researchgate.net/profile/Tiziana-Mascia-2/publication/348715804_I-PERCORSI_DEL_LETTORE_Teorie_e_Buone_Pratiche_per_la_Formazione_MASCIA/links/600c57af299bf14088b8bf01/I-PERCORSI-DEL-LETTORE-Teorie-e-Buone-Pratiche-per-la-Formazione-MASCIA.pdf
18. Mitchel Resnick, libro "Lifelong Kindergarten: Cultivating Creativity through Projects, Passion, Peers, and Play", pubblicato nel 2017, link:https://www.google.it/books/edition/Come_i_bambini/h_hyDwAAQBAJ?hl=it&gbpv=1&dq=the+four+P+of+mitchel+Resnick&printsec=frontcover
19. Ernie Smith, articolo "The ABCs of Wooden Alphabet Blocks" pubblicato il 23 Maggio 2017, link: <https://www.atlasobscura.com/articles/history-alphabet-blocks>
20. Ilana Morelli, articolo "Metodo Montessori: i materiali sensoriali" pubblicato nel 2017, link: <https://www.uppa.it/metodo-montessori-i-materiali-didattici/>
21. Encyclopedia.com, "Century of the Child", link: <https://www.encyclopedia.com/children/encyclopedias-almanacs-transcripts-and-maps/century-child>
22. Lego.com, "About Us" link: <https://www.lego.com/en-us/aboutus/lego-group/the-lego-group-history/>
23. Megardi, articolo "GENIAC (Electric Brain)", pubblicato nel 2020, link:<https://www.instructables.com/GENIAC-Electric-Brain-Replica/>
24. Ilectron.info, museo online sul Ilectron, link: <https://Ilectron.info/>
25. Michele Romano, articolo "Sapientino compie 50 anni: la sfida vinta dei giochi educativi", pubblicato il 24 Ottobre 2017, link:https://www.ilsole24ore.com/art/sapientino-compie-50-anni-sfida-vinta-giochi-educativi-AEWGC6tC?refresh_ce=1

26. Manuale di utilizzo Lego TC Logo, link:https://archive.org/details/lego-tc-logo-teaching-the-turtle_202012/page/n13/mode/2up
27. Timothy S. McNERney, articolo "From turtles to Tangible Programming Bricks: explorations in physical language design", pubblicato nel 2004, link:<https://alumni.media.mit.edu/~mcnerney/personal-ubicomp.pdf>
28. Hideyuki Suzuki, ricerca "AlgoBlock: a Tangible Programming Language - a Tool for Collaborative Learning", link:https://www.researchgate.net/profile/Hideyuki-Suzuki/publication/242383829_AlgoBlock_a_tangible_programming_language_a_tool_for_collaborative_learning/links/575f5c8e08ae414b8e5496e3/AlgoBlock-a-tangible-programming-language-a-tool-for-collaborative-learning.pdf
29. Michael S. Horn e Robert J.K. Jacob, ricerca "Designing Tangible Programming Languages for Classroom Use", pubblicata nel 2007, link:<http://hci.cs.tufts.edu/tern/horn-jacob-tei07.pdf>
30. Kit educativi Lego, link:https://www.campustore.it/steam-lego-education?gclid=Cj0KCQiA3rQBhCNARIsACUEW_Y10zkjv8Pu5BJ3baj4hxhsvKAZTqsH9-JpWj55Y7mVcxfN64hTdEaApSIEALw_wcB
31. Competizione MakeX, link:<https://www.makex.it/>
32. CoSpaces EDU, link: <https://cospaces.io/edu/>
33. Botzees, link: <https://botzeestoys.com/>
34. Karen Brennan e Mitchel Resnick, ricerca "New frameworks for studying and assessing the development of computational thinking", pubblicata nel 2012, link:https://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf
35. Scratchwiki, "History", link: <https://en.scratch-wiki.info/wiki/Scratch>
36. TED Talk, "Mitch Resnick: Insegnamo ai bambini a programmare", novembre 2012, link: https://www.ted.com/talks/mitch_resnick_let_s_teach_kids_to_code?language=it#t-800760
37. 2020 Annual Report Scratch, link: <https://scratch.mit.edu/annual-report>
38. Marina Umaschi Bers, ricerca "Coding and Computational Thinking in Early Childhood: The Impact of ScratchJr in Europe", link:https://www.researchgate.net/publication/327489971_Coding_and_Computational_Thinking_in_Early_Childhood_The_Impact_of_ScratchJr_in_Europe
39. Libro di testo "Giocamici. Matematica, 1^a", pubblicato nel 2020, link: https://issuu.com/mondadori.education/docs/giocamici_matematica_vol1
40. CodyRoby, link:<http://codemooc.org/codyroby/>
41. OMS, "To grow up healthy, children need to sit less and play more", pubblicato il 24 Aprile 2019, link: <https://www.who.int/news/item/24-04-2019-to-grow-up-healthy-children-need-to-sit-less-and-play-more>
42. American Academy of Pediatrics, "Screen Time and Children", pubblicato nel 2016, link:https://www.aacap.org/AACAP/Families_and_Youth/Facts_for_Families/FFF-Guide/Children-And-Watching-TV-054.aspx#:~:text=Screen%20Time%20and%20Children&text=Children%20and%20adolescents%20spend%20a,spend%20up%20to%209%20hours.
43. Kibo, sito ufficiale link: <https://kinderlabrobotics.com/kibo/>
44. Mochi, sito ufficiale link:<https://learnwithmochi.com/>
45. Code a Pillar, video YouTube link: <https://www.youtube.com/watch?v=iYEKD1Befg8>
46. Bee Bot, sito ufficiale link: <https://www.tts-international.com/bee-bot-programmable-floor-robot/1015268.html>
47. Coding & Robotics, sito ufficiale link: <https://www.gigotoys.com/en/products/7072/>
48. Cubetto, sito ufficiale link: <https://primotoys.com/>
49. mTiny, sito ufficiale link: <https://store.makeblock.com/search?type=product&q=mTiny+>
50. Sphero Indi, sito ufficiale link: <https://sphero.com/pages/sphero-indi>
51. MatataBot, sito ufficiale link: <https://matatalab.com/en/coding-set>
52. Qobo, sito ufficiale link: <https://www.robobloq.com/product/Qobo>
53. Coko, video YouTube link: <https://www.youtube.com/watch?v=LUBGhdItwGo>
54. MojoBot, sito ufficiale link: <https://www.mojobot.io/>

