UNIVERSITA' POLITECNICA DELLE MARCHE

FACOLTA' DI INGEGNERIA INFORMATICA E DELL'AUTOMAZIONE

Corso di Laurea triennale in Ingegneria Informatica e dell'Automazione

STUDIO E PROGETTAZIONE DI KIT EDUCATIVI "PARLANTI"

STUDY AND DEVELOPMENT OF "TALKING" EDUCATIONAL TOOLS

Relatore: Chiar.mo/a

Prof. **David Scaradozzi**

Correlatori: **Laura Screpanti**, **Lorenzo Cesaretti**

Tesi di Laurea di:

Lorenzo Cichella

A.A. **2020/2021**

Ringraziamenti

Prima di procedere con la trattazione, vorrei dedicare qualche riga a tutti coloro che mi sono stati vicino in questo percorso di crescita personale e professionale.

In primis, un ringraziamento speciale al mio relatore Scaradozzi David, e ai miei correlatori Laura Screpanti e Lorenzo Cesaretti, per avermi guidato in questo progetto, per gli indispensabili consigli e per avermi trasmesso durante tutto il percorso preziose conoscenze che mi hanno fatto appassionare ancora di più a questa materia.

Un infinito ringraziamento va alla mia ragazza Elisa, che mi ha accompagnato in questi 3 anni, supportato nei momenti più difficili e sostenuto in tutte le mie scelte. Senza di lei non avrei mai fatto questo percorso di crescita che oggi mi ha portato fin qui.

Ringrazio di cuore i miei genitori e tutta la mia famiglia, per avermi appoggiato fin dalla scelta del mio percorso di studi, per non avermi mai fatto perdere la voglia di andare avanti, anche quando la salita sembrava troppo ripida e per il calore con cui mi hanno accompagnato sempre.

Un ringraziamento speciale ai miei amici, da quelli che porto nel mio cuore dall'infanzia a quelli che ho incontrato nel mio percorso e che hanno deciso di camminare al mio fianco e portarmi fin qui. Grazie per esserci stati sempre, per le risate nei momenti di sconforto e per i momenti di spensieratezza.

Indice

Introduzione

La finalità di questa tesi è lo sviluppo e l'implementazione di due sistemi di robotica educativa: il *LEGO MINDSTORMS Education EV3 [11]* e il Carro *M5StickC [2].* Il primo è stato motivo di applicazione pratica nell'articolo "*Utilizing data mining with Robotics for identification and assessment of educational experiences"* ([15] Scaradozzi, Cesaretti, Screpanti, & Mangina). Esso pone l'accento sulle applicazioni della robotica educativa in ambito scolastico e ne esalta le potenzialità attraverso una serie di lezioni in cui gli studenti vengono introdotti nel mondo della robotica e compiono i primi passi verso la comprensione di essa interagendo con un set base LEGO. Pertanto, il presente studio si è posto come obiettivo ideare strumenti che permettano agli studenti di interfacciarsi con un sistema che offre le stesse funzionalità, usando gli stessi meccanismi, seppur sulla base di componenti hardware e software diversi da quelli LEGO.

A tal fine è stato realizzato un kit robotico, il Carro *M5StickC [2],* che tenta di emulare, seppur con problematiche implementative, il comportamento del *LEGO MINDSTORMS Education EV3 [11],* e di offrire una libreria di comandi funzionanti su un dispositivo diverso. Il set creato è basato sul *BALA-C ESP32 Development Mini Self-Balancing Car [3]*, della *M5Stack [1]*, modificato strutturalmente (con la realizzazione di un manuale di istruzioni per il montaggio) e programmato attraverso *UIFlow [4].* Grazie a un'interessante estensione di quest'ultimo, *BlockMaker [8],* è stato possibile creare dei *TalkingBlocks*, ovvero blocchi *UIFlow [4]* "parlanti" capaci di inviare messaggi su un server *Discord [5]* offrendo la possibilità di osservare in tempo reale quali comandi vengono utilizzati, e in quale sequenza. Questa funzionalità, non presente nel set LEGO, esprime il suo potenziale se applicata in ambito scolastico, poiché attraverso di essa gli insegnanti possono monitorare il comportamento degli studenti in tempo reale.

L'elaborato consta di 5 capitoli; nel primo, lo Stato dell'arte, viene presentato l'argomento su cui è incardinata la tesi, utile al lettore per acquisire gli strumenti

terminologici e critici necessari per la comprensione dei contenuti. Nel secondo capitolo vengono descritti i materiali software e hardware utilizzati, mentre nel terzo viene descritta l'implementazione dei sistemi di robotica. Infine, nel quarto capitolo vengono illustrati i risultati ottenuti e nel quinto le limitazioni insorte a causa di alcune problematiche riscontrate ed eventuali sviluppi futuri volti all'applicazione di migliorie laddove possibile.

1 Stato dell’arte

La robotica è quella disciplina dell’ingegneria che si occupa dello studio e dello sviluppo di metodi e tecnologie che consentono ad un robot di eseguire specifici compiti, riproducendo in modo automatico, e il più fedelmente possibile, il lavoro umano. In altre parole, è una branca della meccatronica (una sotto – disciplina dell’ingegneria) nella quale confluiscono approcci, nozioni e concetti appartenenti a diverse discipline sia di natura umanistica, come la linguistica, sia di natura scientifica, come la matematica, la fisica, l’informatica, l’elettronica, e la meccanica etc. Essa trova applicazione odierna in innumerevoli campi, dalla vita quotidiana alle industrie, una tra queste prende il nome di robotica educativa, un innovativo approccio all'insegnamento basato sull'utilizzo dei robot a scuola e finalizzato a rendere più efficace e coinvolgente la didattica per bambini e ragazzi. In questo senso, quindi, la robotica diventa un metodo pedagogico che rende più semplice il lavoro degli insegnanti. Rappresenta, infatti, un valido strumento capace di trasformare lezioni che possono essere noiose in attività creative e divertenti. La seguente tesi si pone l’obiettivo di affrontare e risolvere la tematica trattata nell’abstract "*Identification and Assessment of Educational Experiences: Utilizing Data Mining With Robotics*" ([15] Scaradozzi, Cesaretti, Screpanti, & Mangina), ovvero l’implementazione di un kit robotico “parlante”, messo a confronto con quello utilizzato nell’articolo sopracitato, capace di effettuare i movimenti scelti dallo studente e programmabile attraverso un linguaggio a blocchi. Questi ultimi, soprannominati “Talking Blocks”, sono stati modificati affinché tenessero traccia in tempo reale delle scelte (in termini di blocchi) opzionate dagli alunni, e comunicassero con l’applicazione *Discord [5]* inviando messaggi su un server creato appositamente per la classe. I messaggi rivestono un ruolo importante in quanto permettono agli insegnanti di monitorare il comportamento degli studenti e il loro approccio quando vengono messi davanti a dei problemi da risolvere.

2 Materiali e metodi

Questo capitolo iniziale è incentrato sui materiali e i metodi impiegati per la realizzazione dei sistemi di robotica educativa. Gli studenti avranno bisogno di un set LEGO MINDSTORMS Education EV3 [11] e di un microcontrollore *M5StickC [2]*, fornito dalla *M5Stack [1]*, che verrà montato sul *Carro M5StickC [2]***,** una macchinina programmabile basata sul kit robotico *Bala.C [3]*, che verrà programmato attraverso l'IDE *UIFlow [4]*, una piattaforma di sviluppo che utilizza un linguaggio a blocchi. Gli insegnanti potranno monitorare i progressi degli studenti tramite l'applicazione *Discord [5]*, con un canale testuale riservato ad ogni classe, in cui verranno mostrati in maniera simultanea i blocchi utilizzati, grazie a dei messaggi dedicati.

2.1 M5StickC

M5StickC [2] ESP32 Development Board è una scheda di sviluppo estremamente versatile, programmabile in diverse modalità: in linguaggio Blockly e Micropython con l'apposito IDE *UIFlow [4]* e in linguaggio C con l'IDE *Arduino [6].* Il piccolo "mattoncino" ha dimensioni estremamente contenute 5×2,5cm, e presenta al suo interno diversi dispositivi:

- ESP32
- display da 0,96 pollici (risoluzione 160×80)
- microfono
- buzzer

- trasmettitore IR
- WiFi
- Bluetooth
- accelerometro
- giroscopio (6 gradi di libertà)
- LED di segnalazione integrato
- due pulsanti (A e B) programmabili
- batteria da 80 mAh
- memoria flash da 4MB
- modalità di programmazione via USB e WiFi

Una pressione di 2 secondi sul *Power Button*, sulla sinistra dell'*M5StickC [2]* provoca l'accensione del dispositivo, non appena compare il logo di *UIFlow [4]* premere il pulsante grande M5 per accedere al menù di Setup. Per spostarsi tra i menù disponibili utilizzare il pulsante in alto a destra di *M5StickC [2].* In questo modo può essere programmato via USB o in modalità WiFi, attraverso la sezione "Switch mode". Per riavviare il blocchetto basterà premere il Power Button sulla sinistra, mentre per spegnerlo occorre fare pressione per 6 secondi. Grazie all'interfaccia GROV e GPIO è possibile connettere una vasta gamma di dispositivi esterni, disponibili sulla pagina web di *M5Stack [1].*

Figura 1: M5StickC [2]

2.2 UIFlow

UIFlow [4] è una piattaforma di programmazione appositamente progettata per i dispositivi *M5Stack [1].* L'IDE di programmazione grafico a blocchi è basato su blockly, linguaggio grafico di programmazione ben conosciuto in campo didattico. *UIFlow [4]* consente inoltre di programmare qualsiasi oggetto *M5Stack [1]* in MicroPython, implementazione di Python 3 per microcontrollori e sistemi embedded. MicroPython inoltre è un linguaggio di programmazione snello ed efficiente, ideale per attività di Coding nelle classi e da sempre impiegato in campo scientifico. *UIFlow [4]* fornisce le funzionalità necessarie per la realizzazione, in maniera estremamente semplice, di progetti con forte interazione con il mondo reale, ideale quindi per chi si avvicina al mondo della prototipazione elettronica, dell'automazione e della programmazione. L'IDE *UIFlow [4]* può essere utilizzato on-line oppure localmente scaricandolo dal sito principale. Per ottenerlo collegarsi al sito https://m5stack.com/ e selezionare dal

menu principale: software > download.

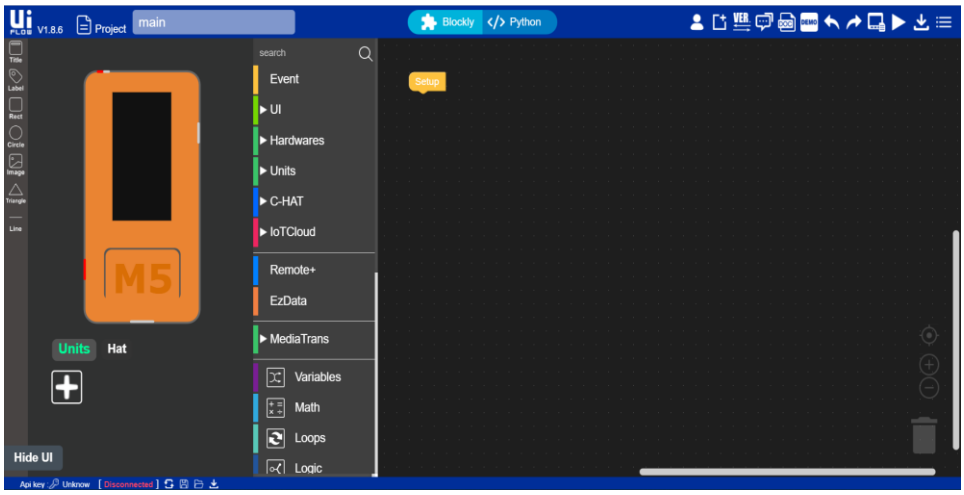


Figura 2: Interfaccia UIFlow [4] v1.8.6

In questo caso è stata usata per semplicità la versione online, in quanto non richiede l'installazione di alcun driver e l'aggiornamento dei firmware. Per collegare *l'M5StickC [2]* al pc, bisogna recarsi sulla pagina principale di *UIFlow [4]* (è stato utilizzato il browser *Chrome [7]*), in seguito seguire i seguenti passaggi:

1. Assicurarsi che il pc sia connesso ad una rete stabile e che abbia un nome senza spazi
2. Avviare il blocchetto M5 e nel menu Setup selezionare WiFi via-app
3. Sul browser cercare l'indirizzo 192.168.4.1
4. Subito dopo collegare il pc alla rete M5-7EE8, che sarà appena apparsa tra le reti disponibili

Nel browser verrà visualizzata una nuova pagina, in cui bisogna scegliere la rete alla quale si vuole collegare il blocchetto e inserire la relativa chiave d'accesso:

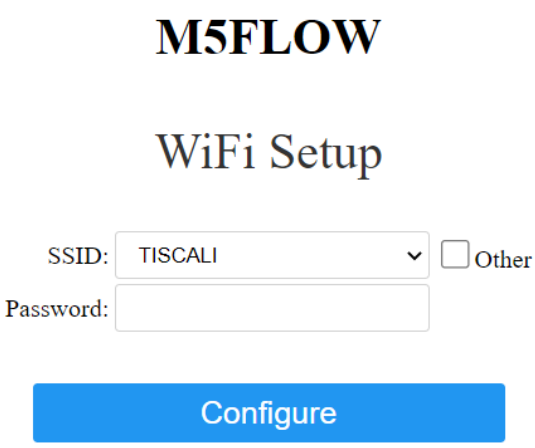


Figura 3: Interfaccia di connessione dell'M5StickC [2]

6. Dopo qualche istante l'M5 effettuerà la connessione:

Figura 4: avviso di connessione avvenuta correttamente

7. Infine, su *UIFlow [4]*, nella barra in basso a sinistra, cliccare sul refresh per connettere l'M5

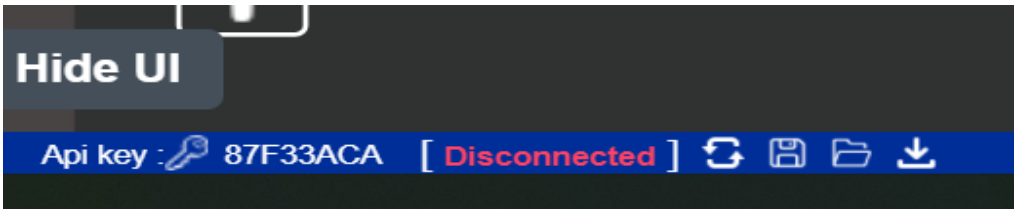


Figura 5: M5StickC [2] non connesso

Figura 6: M5StickC [2] connesso

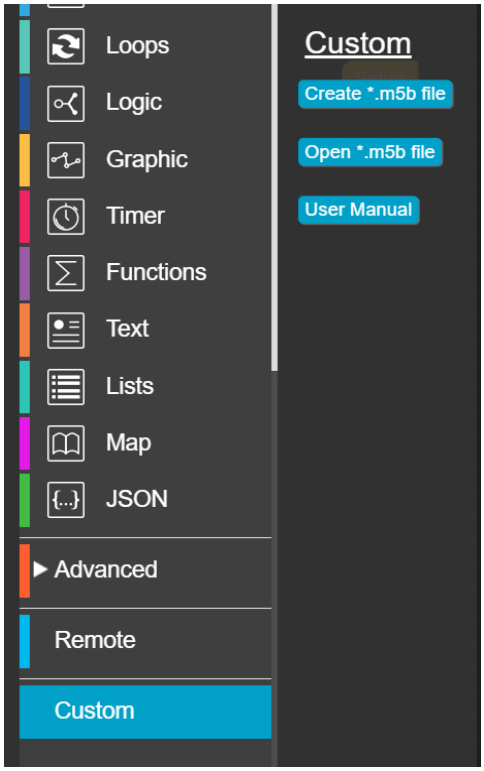


Figura 7: Libreria custom UIFlow [4]

A questo punto, si possono implementare programmi componendo i blocchi presenti nelle varie librerie, e agganciandoli al blocco predefinito *setup*. Inoltre, la piattaforma *UIFlow [4]* permette di creare dei blocchi personalizzati partendo da quelli già esistenti o ideandoli da zero, accedendo alla sezione *custom*. La sezione *Open*.m5b file* consente di importare librerie di blocchi customizzati, mentre cliccando su *Create*m5b file* si accede a

BlockMaker [8] (figura 8), una funzionalità di *UIFlow [4]* attraverso cui creare e personalizzare i blocchi. Per creare un nuovo blocco bisogna fornire:

1)*Group name*: il nome della libreria in cui è posizionato il blocco.

2)*Block color*: colore del blocco.

3)*Block label*: sono consentiti solo lettere, numeri e trattini bassi.

4)*Block type*: si possono definire blocchi di tipi *Value* (che restituiscono un valore) o blocchi di tipo *Execute* (svolgono un'azione).

5)*Number of blocks*: c'è la possibilità di aggiungere più blocchi contemporaneamente e salvarli in un file.m5b (cliccando su *download*).

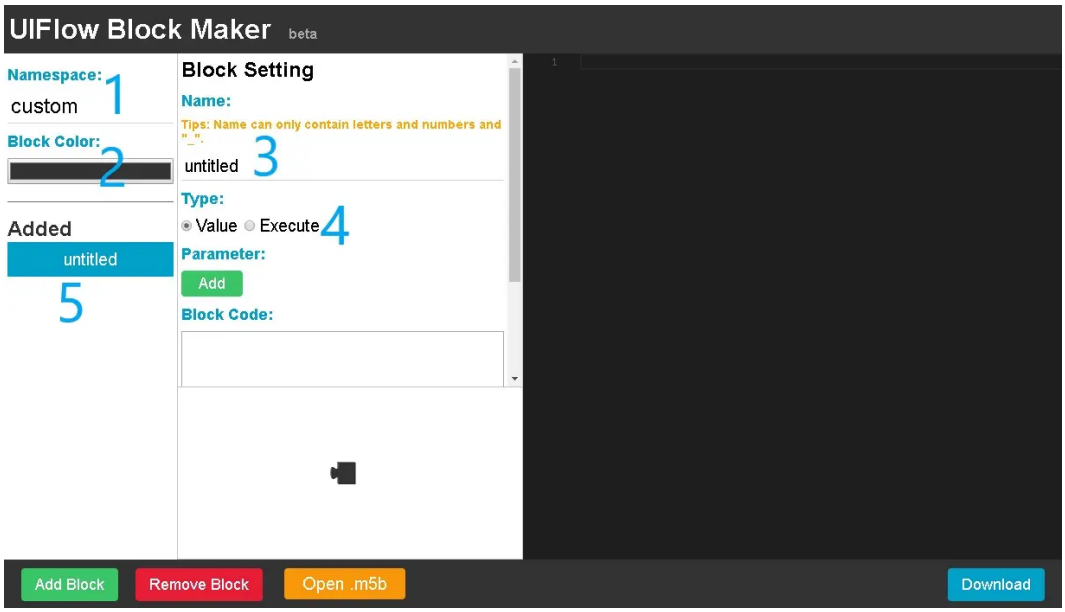


Figura 8: interfaccia BlockMaker [8]

Questa funzionalità offerta da *UIFlow [4]* è stata rilevante per la realizzazione del robot *M5StickC [2]* "parlante" (di cui si parlerà in seguito), ed ha consentito di aggiungere un protocollo per lo scambio di messaggi all'interno di ogni blocco (argomento trattato da *Rumeysa Gulesin [10]* e *Marco Cervigni [9]* nelle rispettive tesi), in modo tale da tenere traccia in tempo reale di tutti i blocchi che vengono utilizzati, attraverso un apposito *server Discord [5]*.

2.3 Discord

Discord [5] è una piattaforma di VoIP, dedicata alla messaggistica istantanea e alla distribuzione digitale di contenuti. Gli utenti hanno modo di comunicare con videochiamate, chiamate vocali, messaggi di testo, media o file, in chat private o chat condivise con altri. Presenta innumerevoli vantaggi grazie alla bassa latenza e un'alta stabilità, server gratuiti per gli utenti e infrastrutture per server dedicati. Le funzionalità da esso offerte sono state importanti per la realizzazione del sistema di messaggi implementato per monitorare i blocchi utilizzati su *UIFlow [4].*

Figura 9: Logo Discord [5]

2.4 M5Stack Bala.C Development Kit

Il *Bala.C [3]* è un kit robotico DIY a doppia ruota distribuito da *M5Stack [1]* e basato sul modulo *M5StickC [2]*. Utilizza un chip di serie STM32, due circuiti integrati per controllare i due motori Servo 9G ed una batteria *Li-Ion 16340* da 700 mAh sostituibile ricaricabile. Esso è programmabile tramite l'IDE *UIFlow [4], Micropython* e *Arduino [6]*. Il kit include al suo interno:

- 1x M5StickC [2]
- 1x base Bala.C [3]
- 2x ruote
- 2x connettori per ruote
- 2x servo 9G
- 2x elastici
- 2x viti
- 1x chiave esagonale
- 1x batteria 16340
- 1x cavo di tipo C da 10 cm

L'applicazione principale del *Bala.C [3]* è un programma di bilanciamento che mantiene il robot in equilibrio in posizione verticale, reso possibile grazie ad un chip IMU (MPU-6886). Inoltre, il kit presenta un design versatile compatibile con i LEGO. Quest'ultima peculiarità è stata fondamentale per effettuare modifiche strutturali al robot, in modo da emulare il *LEGO MINDSTORMS Education EV3 [11]* (di cui si parlerà in seguito) e creare il *Carro M5StickC [2]*, ovvero il prodotto finale di questo progetto.

Figura 10: Kit Bala.C [3]

2.5 LEGO MINDSTORMS Education EV3

Figura 11: Set LEGO MINDSTORMS Education EV3 [11]

Il *LEGO MINDSTORMS Education EV3 [11]* è un set base distribuito dalla LEGO che trova applicazione nel campo della robotica educativa. Esso consente agli utenti di creare, programmare e testare robot programmabili mediante l'uso di motori, sensori, ingranaggi, ruote, assi e altri componenti tecnici. Contiene un *Brick EV3 LEGO [13]*, un dispositivo piccolo e potente attraverso il quale è possibile controllare i motori, ricevere il feedback dai sensori e che mette a disposizione innumerevoli funzioni. Possiede 4 ingressi nella parte anteriore per dispositivi esterni, e 4 nella parte posteriore per collegare i motori. Il blocchetto

LEGO può essere collegato al pc (attraverso un software dedicato di cui si parlerà in seguito) tramite bluetooth, Wi-Fi o tramite cavo. Il set include:

- Tre servomotori interattivi
- Sensore di rotazione e sensore ad ultrasuoni incorporati
- Sensore di colore/luce, sensore giroscopico e due sensori di contatto
- Batteria ricaricabile
- Ruota a sfera
- Cavi di collegamento
- Istruzioni di montaggio
- Mattoncini per costruzioni LEGO® Technic per creare un'ampia varietà di modelli

Per la programmazione del robot educativo *LEGO MINDSTORMS Education EV3 [11]* è possibile usare un software proprietario oppure LabVIEW, JAVA, C/C++, Basic, Swift Playground e molti altri. In questo caso di studio è stato utilizzato il software apposito messo a disposizione dalla LEGO (*Software LEGO MINDSTORMS Education EV3 [12] v1.4.5*).

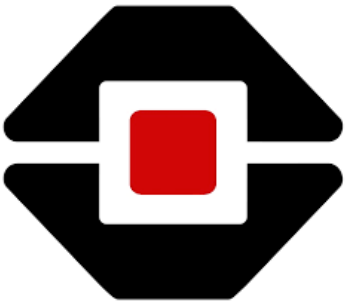


Figura 12: Logo Software LEGO MINDSTORMS Education EV3 [12]

Il software, che può essere scaricato gratuitamente, è ideato per la programmazione e il data-logging, di facile apprendimento e semplice da utilizzare, è basato sulla tecnologia National Instruments LabVIEW™ e comprende 48 tutorial dettagliati. È ottimizzato per l'uso all'interno delle classi in quanto presenta un linguaggio a blocchi molto intuitivo. In questo caso di studio è stato utilizzato il set LEGO sopra citato per realizzare una struttura motrice su due ruote, e programmarla attraverso il blocco *Move_Steering,* che verrà approfondito nei capitoli successivi, per poi emularne le funzionalità con il Carro *M5StickC [2].*

2.6 Dr. Explain

Dr. Explain [14] è un software per la creazione di manuali di linea, guide per l'utente, file guida e documentazione di supporto per prodotti e servizi software. È possibile creare istruzioni in formato HTML (per manuali online), CHM (file di aiuto MS Windows), RTF e PDF. Inoltre, offre strumenti utili ed efficaci per la semplificazione del lavoro, come una funzione per l'acquisizione delle immagini visualizzate su schermo, un ricco editor di contenuti messo a punto per la creazione della documentazione dei software e il monitoraggio della completezza del progetto tramite gli stati dell'argomento e il blocco. Nella

realizzazione di questo caso di studio il software *Dr. Explain [14]* ha svolto un ruolo determinante in quanto ha reso possibile realizzare un libretto di istruzioni per la costruzione del Carro *M5StickC [2]*.

Figura 13: Logo Dr. Explain [14]

3 Sviluppo e progettazione dei sistemi di robotica educativa

In questo capitolo verranno illustrati i due sistemi di robotica educativa che sono stati oggetto di studio, sia dal punto di vista hardware (come sono stati realizzati) che dal punto di vista software (come sono stati programmati e azionati). Sono stati utilizzati un kit *LEGO MINDSTORMS Education EV3 [11]* (con il relativo software) e un kit robotico *Bala.C [3]* (programmato attraverso *UIFlow [4]*). L'obiettivo principale è stato quello di riprodurre il robot LEGO e le sue funzionalità con un software completamente diverso, e osservare i risultati ottenuti, in vista di una eventuale applicazione pratica del *Carro M5StickC [2]* in ambito scolastico (creazione di lezioni simili a quelle realizzate nell'abstract ([15] Scaradozzi, Cesaretti, Screpanti, & Mangina) *Identification and Assessment of Educational Experiences: Utilizing Data Mining With Robotics.*

3.1 Costruzione e programmazione del robot LEGO

In questo caso di studio, è stata realizzata (a partire dal *set LEGO MINDSTORMS Education EV3 [11]*) una struttura motrice su due ruote anteriori, e un ruotino omnidirezionale posteriore per consentire movimenti in tutte le direzioni. Inoltre, è stato montato in posizione frontale un *sensore a ultrasuoni LEGO [16].* Quest'ultimo genera onde sonore e mediante la lettura dei loro echi rileva e misura la distanza dagli oggetti. Effettua misurazioni della precisione del centimetro e che vanno da 1 a 250 cm di distanza. Il robot è stato costruito seguendo un *manuale di istruzioni [21]* reperibile online o direttamente all'interno del kit.

Figura 14: Robot LEGO MINDSTORMS

Riguardo invece alla programmazione del robot, è stato utilizzato il *Software LEGO MINDSTORMS Education EV3 [12] v1.4.5.* Una volta aperta l'applicazione,

se si vuole creare un nuovo progetto bisogna selezionare FILE/ NEW PROJECT/ PROGRAM, in questo modo verrà aperto un nuovo progetto e visualizzata l'interfaccia di programmazione (figura 15). Quest'ultima avviene attraverso l'utilizzo di blocchi, che l'utente può trascinare con il cursore del mouse e collegare in sequenza per realizzare il programma che desidera. I blocchi, per essere abilitati devono essere agganciati al *blocco setup,* presente al centro della schermata. Le librerie sono presenti nella parte inferiore dell'interfaccia, e offrono varie funzionalità. In basso a destra l'utente può trovare un'estensione per collegare il *mattoncino EV3* al pc. L'associazione può essere effettuata tramite cavo USB, oppure wireless, via bluetooth o Wi-Fi, in questo caso è stata utilizzata la modalità bluetooth.

Connessione Bluetooth: attivare il bluetooth sul *Brick EV3 LEGO [13]* dalle impostazioni, se nell'interfaccia *LEGO MINDSTORMS* il blocchetto non è già presente, premere su *refresh* in basso a destra e accettare manualmente la richiesta sul mattoncino LEGO. Verrà chiesto un pass-key, che è preimpostato ed è 1234.

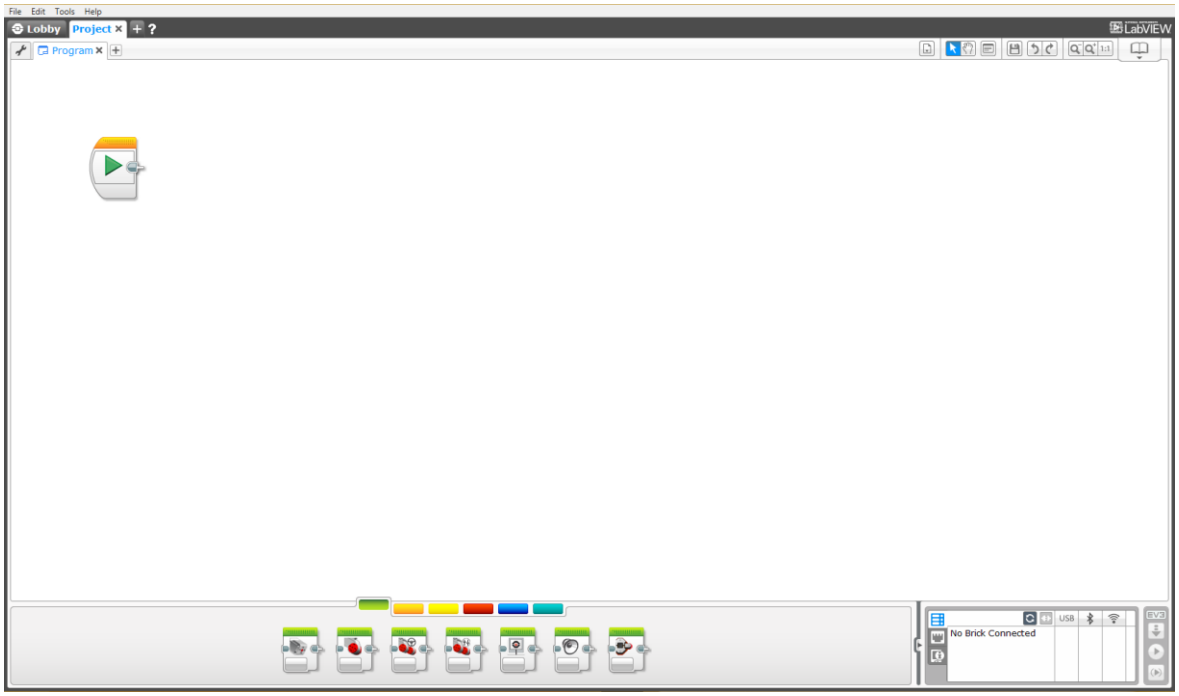


Figura 15: Interfaccia LEGO MINDSTORMS Education EV3 [12]

La libreria utilizzata è stata *Move_Steering*, questo blocco può far avanzare, retrocedere, girare o arrestare il robot, controllando contemporaneamente entrambi i motori. Lo sterzo può essere regolato per far andare dritto il robot, guidare in archi o fare curve strette. Il tutto viene racchiuso in un menù a tendina in cui l'utente sceglie tra diverse modalità, ognuna con le sue caratteristiche (figura 16).

1)Modalità *OFF*: spegne entrambi i motori, e viene utilizzata per arrestare il robot precedentemente avviato con un'altra modalità.

2)Modalità *ON*: accende entrambi i motori, quindi passa immediatamente al blocco successivo nel programma. È possibile controllare la velocità e la direzione del robot utilizzando gli ingressi *Power* (potenza) e *Steering* (sterzo). I motori

funzioneranno fino a quando non verranno arrestati o modificati da un altro blocco più avanti nel programma o fino al termine del programma.

3)Modalità *ON FOR SECONDS*: accende entrambi i motori e li mantiene accesi per il numero di secondi nell'ingresso *Seconds*, quindi li spegne. Il blocco attenderà che sia trascorso il tempo prima che il programma prosegua con il blocco successivo. Si può controllare la velocità e la direzione del tuo robot utilizzando gli ingressi *Power* e *Steering*.

4)Modalità *ON FOR ROTATIONS*: accende entrambi i motori, attende che le ruote abbiano raggiunto il numero di rotazioni scelte nell'ingresso *Rotations*, quindi spegne entrambi i motori. Questo può essere usato per far percorrere al tuo robot una distanza specifica o per far compiere alle ruote delle rotazioni. L'utente può anche scegliere la *potenza* e la *direzione dello sterzo*.

5)Modalità *ON FOR DEGREES*: accende entrambi i motori, attende che le ruote abbiano raggiunto i gradi scelti nell'ingresso *Degrees*, quindi spegne i motori. Viene utilizzato per far compiere al robot una distanza specifica o per farlo muovere fino ad una certa gradazione. 360 gradi corrispondo ad un giro completo dei motori, e quindi delle ruote. Questo blocco offre, come i blocchi precedenti, la possibilità di regolare *potenza* e *direzione dello sterzo*.

Da notare che nella parte superiore dei blocchi è possibile scegliere attraverso un menù a tendina fino a 4 lettere (A, B, C, D) in quanto il mattoncino EV3, come spiegato in precedenza, presenta 4 porte per collegare i motori. Quindi se si collegano i motori alle porte A e B, nel blocco in *LEGO MINDSTORMS* dovranno essere selezionate quelle lettere, altrimenti, se si selezionano porte diverse, il *Brick EV3 LEGO [13]* non manderà alcun impulso.

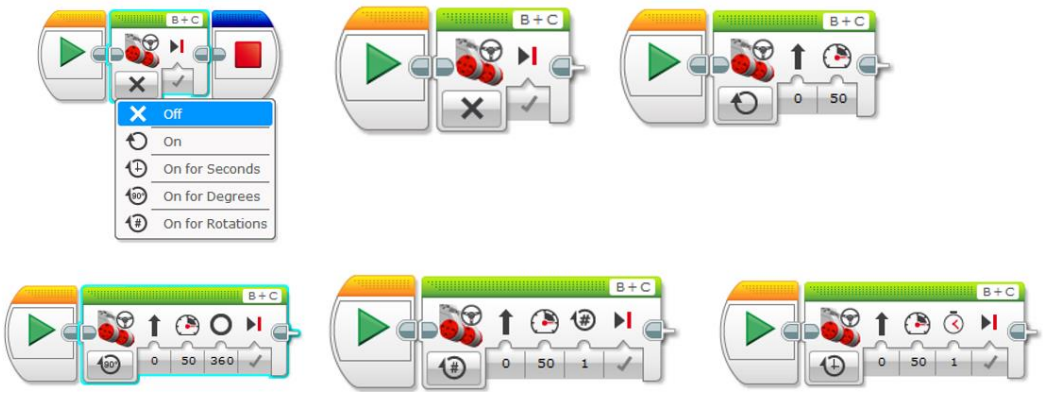


Figura 16: Blocchi libreria Move Steering

3.2 Implementazione dei Talking Blocks

Questa implementazione ha consentito di realizzare una feature all'interno del progetto del *Carro M5StickC [2]* che non è presente nel kit *LEGO MINDSTORMS EV3 [11]*. I *Talking Blocks* sono dei normali blocchi all'interno dell'IDE *UIFlow [4]*, con l'aggiunta di un'interessante funzionalità che consente di tenere traccia di tutte le istruzioni che vengono eseguite in tempo reale. Sono stati realizzati tramite *BlockMaker [8]*, aggiungendo nel *block-code* un protocollo per lo scambio di messaggi (figura 17):

```
wait_ms(500)

import urequests

try:
   req = urequests.request(method='POST', url='server_url',
   json={'content':'#ON;var1:${steering};var2:${power};end'},
   headers={'authorization':'                                                    '})
   print('Successo')
except:
   print('Fallimento')

wait_ms(500)
```

Figura 17: Esempio protocollo per lo scambio di messaggi

Per creare un “blocco parlante”, si copia il suo codice MicroPython (che si trova switchando dal linguaggio blockly su *UIFlow [4]*) nel *block-code*, e si aggiunge alla fine il protocollo riportato nella Figura 17. Per la ricezione dei messaggi è stata usata la piattaforma *Discord [5]*, in cui è stato creato un server con diversi canali testuali (uno per ogni studente) per riprodurre una classe. Per scegliere il canale in cui mandare il messaggio, bisogna eseguire il programma python “SwitchID.py”, che richiederà l’ID del canale testuale scelto (l’ID è reperibile su *Discord [5]* facendo tasto destro sul canale testuale e selezionando “copia l’ID”). Nella sezione ‘authorization’ invece va inserito un token, un codice associato ad ogni account *Discord [5].*

L’argomento dei *Talking Blocks* è stato meglio trattato e approfondito negli elaborati di *Rumeysa Gulesin [10]*(Rumeysa, 2020-2021) e *Marco Cervigni [9]* (Cervigni [9], 2020-2021).

3.3 Costruzione e programmazione del Robot M5StickC

Il *Carro M5StickC [2]* è stato costruito a partire dall’*M5Stack [1] Bala.C [3] Development Kit*, poi opportunamente modificato introducendo componenti aggiuntive per rendere la sua struttura simile al *LEGO MINDSTORMS Education EV3 [11]*.

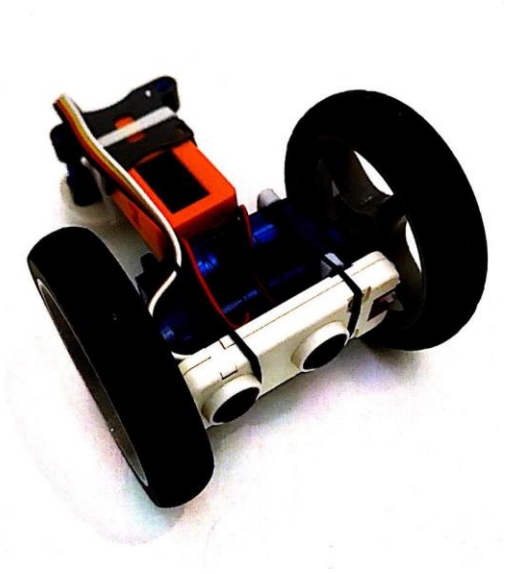


Figura 18: Carro M5StickC [2]

È stato realizzato un *manuale di istruzioni [22]* con il software *Dr. Explain [14]* in cui sono rappresentate le istruzioni di montaggio del robot. Esso presenta tutte le caratteristiche del *Bala.C [3]*, con l'aggiunta di un ruotino omnidirezionale scorrevole nella parte posteriore e di un *sensore a ultrasuoni [18]* fornito dalla *M5Stack [1]*. Quest'ultimo dispone di una sonda a ultrasuoni con una frequenza sonora di 40KHz, un angolo di direzione di ±20° e una precisione di 1 mm. Al suo interno è presente un chip di misurazione della distanza ad ultrasuoni RCWL-9600 e i risultati della misurazione possono essere ottenuti direttamente tramite l'interfaccia I2C (0x57). La distanza effettiva è di 30 - 150 cm. Il sensore è stato collegato con un cavo Grove alla porta dell'*M5StickC [2]* per dispositivi esterni.

Figura 19: Ultrasonic sensor [18]M5Stack

Il robot è stato programmato attraverso l'IDE *UIFlow [4]* . Visto che l'*M5StickC [2]* è stato montato su un modulo (in questo caso il *Bala.C [3]*) e vi è stato collegato un dispositivo esterno (*Ultrasonic sensor [18]*) è stato necessario aggiungere un'unità e un hat. In questo caso, l'hat *Bala.C [3]* non è stato potuto aggiungere poiché *UIFlow [4]* non dispone della libreria necessaria per programmarlo (essa infatti è disponibile sono in *Arduino [6]*); tuttavia è stato trovato un secondo hat, il *BugC [19]*, che presenta dei controlli compatibili con il *Bala.C [3]* e quindi è stato importato in *UIFlow [4].*

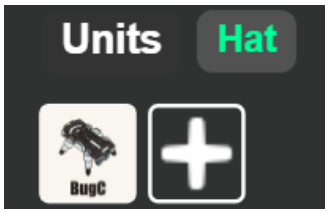


Figura 20: Importazione BugC [19]

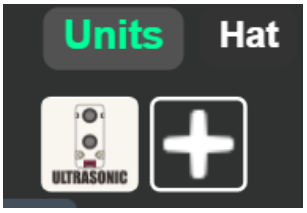


Figura 21: Importazione Ultrasonic Sensor [18]

Dopo l'importazione, nella libreria C-HAT sarà comparso il *BugC [19]* e in Units *l'Ultrasonic Sensor [18].* I blocchi relativi al sensore saranno richiesti in seguito, parlando di un'applicazione pratica di questo progetto in delle classi di studenti.

Per quanto riguarda invece i blocchi del *BugC [19]*, sono stato rilevanti per consentire al *Carro M5StickC [2]* di muoversi e sterzare come il LEGO MINDSTORMS. Il blocco *SetPulse* (Figura 22) è stato utilizzato per dare l'impulso (in microAmpere) ai motori SERVO del robot *M5StickC [2]*, l'utente può scegliere due valori: il primo, selezionabile da un menù a tendina che presenta interi da 0 a 3, serve per scegliere a quale motore dare l'impulso (sono presenti 4 scelte poiché il *BugC [19]* presenta 4 motori), il secondo invece per settare l'intensità dell'impulso (sempre attraverso un numero intero). Bisogna tenere presente che la potenza data ai motori rimane costante nel tempo, infatti, dato un impulso diverso da zero al robot, esso tenderà ad avanzare in una certa direzione senza sosta; quindi, se lo si vuole arrestare è necessario impostare l'impulso a zero. In questo caso, presentando il *Bala.C [3]* solo due motori, è stata eliminata l'opzione di scelta attraverso *BlockMaker [8]*, e sono stati impostati di default i SERVO 0 e 1.

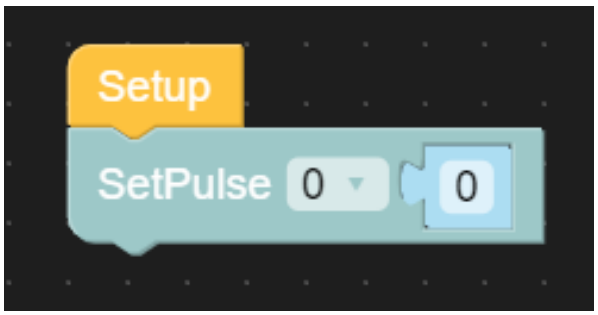


Figura 22: Blocco SetPulse

Il blocco *SetPulse* non è stato usato direttamente, ma è stato copiato il suo codice *MicroPython* da *UIFlow [4]* per creare i blocchi personalizzati della libreria *Move_Steering* in *BlockMaker [8].* La realizzazione di quest'ultima verrà trattata nel capitolo successivo, dove sono illustrati i risultati ottenuti, anche mediante dei tests sull'accuratezza.

4 Risultati ottenuti e applicazioni in un caso di studio

In questo capitolo vengono illustrati i risultati ottenuti dalla creazione della libreria *Move Steering*, attraverso un confronto diretto con la libreria di blocchi LEGO. In seguito, sarà effettuata una validazione sui dati finali per verificarne l'accuratezza, ed infine sarà presentata una possibile applicazione pratica degli strumenti sviluppati all'interno di classi di studenti.

4.1 Libreria Move Steering

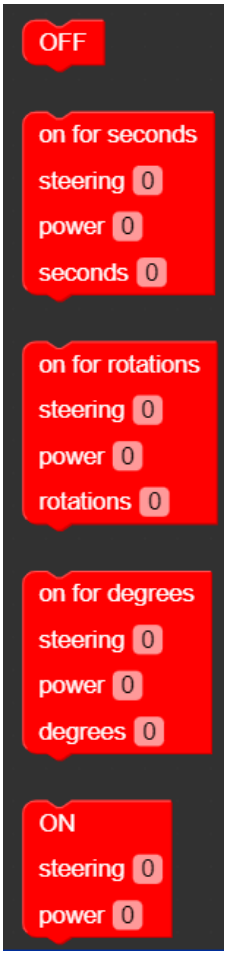


Figura 23: Libreria Move Steering UIFlow

La *libreria Move Steering* (Figura 23) è costituita da blocchi personalizzati creati attraverso BlockMaker [8]. In primo luogo, nel *Software LEGO MINDSTORMS Education EV3 [12]*, in un unico blocco venivano racchiuse tutte le funzionalità a disposizione dell'utente, che potevano essere selezionate attraverso un opportuno menù a tendina (Figura 16), funzionalità non riproducibile in *UIFlow [4];* quindi, l'idea è stata di creare blocchetti diversi che effettuassero le stesse operazioni di quelli LEGO, e collocarli in un'unica libreria. Ora verrà illustrata l'implementazione dei blocchi *in BlockMaker [8]:*

Block Setting

Name:

Tips: Name can only contain letters and numbers and "_".

off

Type:

○ Value ◉ Execute

Parameter:

name: off type: Label Remove

Add

Block Code:

```
import hat
hat_bugc_0 = hat.get(hat.BUGC)

hat_bugc_0.SetPulse(0, 0)
hat_bugc_0.SetPulse(1, 0)
wait_ms(500)
```

Figura 25: Block Code modalità OFF

-*Modalità OFF:* In questo caso, non ci sono parametri da scegliere, poiché quando viene eseguito il blocco, i motori vengono spenti. In *UIFlow [4]* questa opzione è stata realizzata settando l'impulso a zero in entrambi i motori del carro *M5StickC [2]*. Il blocco è di tipo *execute*, poiché bisogna eseguire un comando e non viene restituito dal programma nessun valore. L'unico parametro inserito è off, di tipo label, utilizzato per identificare il blocco.

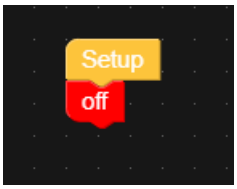


Figura 24: Blocco OFF UIFlow

Figura 26: Protocollo scambio di messaggi modalità OFF

```
wait_ms(500)

import urequests

try:

  req = urequests.request(method='POST',
url='https://discord.com/api/v9/channels/885981675580977202/messages', json={'content':'#Off;end'},
headers={'authorization':'ODcwNjEzNTM2MTI3MjA5NDgy.YQPU-A.2IqIEn2bmBLEP2bKTkEzK9wX3o4'})

  print('Successo')

except:

  print('Fallimento')

wait_ms(500)
```

Nel *Block Code* viene importata la libreria hat, e in seguito all'hat importato viene assegnato il *BugC [19].* Infine, due comandi di tipo *SetPulse* per impostare l'impulso dei due motori a 0, in modo che il robot si fermi se è in movimento, o rimanga fermo se è non è in movimento. Nella Figura 26 è presente il protocollo per lo scambio di messaggi, che consente di inviare un messaggio (sotto forma di stringa) su un server *Discord [5]* appositamente creato, in cui viene visualizzato il blocco appena utilizzato, e nella Figura 27 il messaggio visualizzato su *Discord [5].*

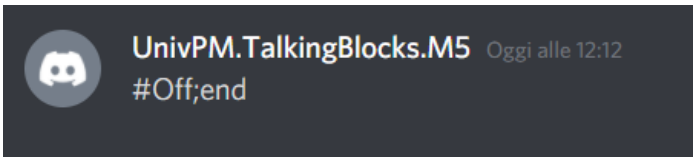


Figura 27: Messaggio Discord [5] blocco OFF

Figura 28: Blocco ON UIFlow

-Modalità ON: La modalità on accende entrambi i motori; quindi, passa immediatamente al blocco successivo nel programma. È possibile controllare la velocità e la direzione dei motori utilizzando i parametri *power* e *steering.* I motori funzioneranno fino a quando non verranno arrestati o modificati da un altro blocco più avanti nel programma o fino al termine del programma. Per riprodurre questo blocco in *BlockMaker [8]* sono state effettuate piccole modifiche che però sono riuscite a rendere il risultato molto vicino al blocco LEGO. Per prima cosa, data la mancanza di un sensore giroscopico sul carro *M5StickC [2]*, non è stato possibile riprodurre nella regolazione del parametro steering una scelta di angolazioni vasta come quella

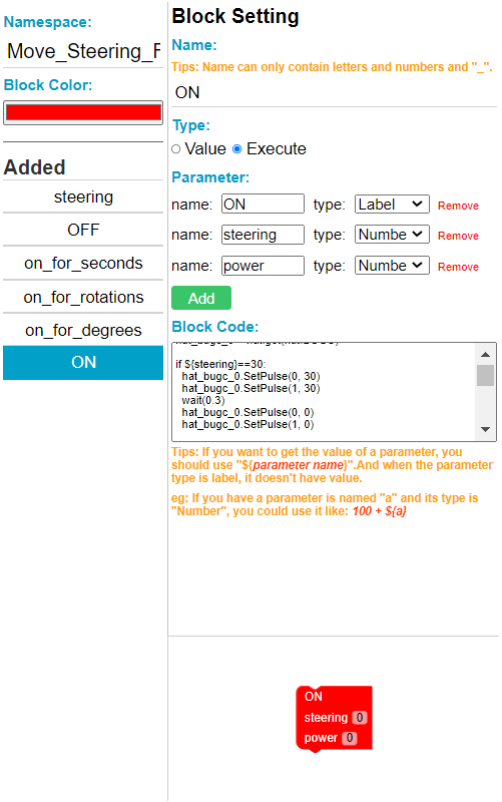


Figura 29: Block Code modalità ON

del blocco LEGO, quindi, attraverso test e misurazioni empiriche, sono stati inseriti come parametri disponibili gli angoli più comuni, ovvero 30°, 45°,90°,180° e 360°, sia in senso orario che antiorario. Similmente al blocco LEGO, l'utente può scegliere 2 parametri, *steering* che permette di settare l'angolazione, e *power*, attraverso cui si sceglie la potenza da dare ai motori. Una cosa importante da sottolineare è che il *LEGO MINDSTORMS EV3* riesce a curvare la traiettoria mentre avanza, nel carro *M5StickC [2]* questo non è stato possibile, poiché nel blocco *UIFlow [4]* è possibile settare solo l'angolazione iniziale, perciò il robot, all'avvio del programma, ruoterà secondo l'angolo scelto dall'utente, e poi avanzerà dritto. Per dare la possibilità di sterzare durante il movimento è stato creato un blocco *Steering*, attraverso cui si sceglie l'angolo di rotazione, e aggiungendolo su *UIFlow [4]* al blocco ON è possibile far sterzare il carro *M5StickC [2]* durante il suo avanzamento.

Per implementare gli angoli di rotazione sono stati realizzati una serie di IF, ognuno per ogni angolo, sia in senso orario che antiorario, in cui è stata data potenza ai motori per un certo timer. Il timer è diverso per ogni angolo, ed è il tempo che impiega il robot per compiere quella rotazione. Nel protocollo per lo scambio di messaggi (Figura 30) è stato modificato la stringa visualizzata su *Discord [5]* (#ON) e le variabili inserite dall'utente.

```
wait_ms(500)

import urequests

try:
   req = urequests.request(method='POST', url='server_url',
   json={'content':'#ON;var1:${steering};var2:${power};end'},
   headers={'authorization':'token'})
   print('Successo')
except:
   print('Fallimento')

wait_ms(500)
```

Figura 30: Protocollo scambio di messaggi blocco ON

Il messaggio su *Discord [5]* viene visualizzato correttamente, var1 è la variabile *steering,* e var2 è la variabile *power,* entrambe settate a 0.

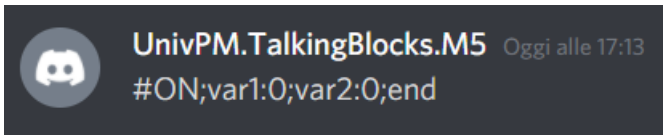


Figura 31: Messaggio Discord [5] blocco ON

-*Modalità ON FOR SECONDS*: La modalità ON FOR SECONDS presenta le stesse caratteristiche del blocco ON, ma fornisce un'opzione aggiuntiva, quella di far scegliere all'utente per quanti secondi mantenere accesi i motori, alla fine del timer il robot viene arrestato. Quindi i parametri scelti dall'utente sono *steering*, *power* e *seconds*. Nel protocollo per lo scambio di messaggi (Figura 33) è stata modificata la stringa visualizzata su *Discord [5]* e le variabili inseribili dall'utente.

Figura 32: Blocco ON FOR SECONDS UIFlow [4]

```
wait_ms(500)

import urequests

try:
   req = urequests.request(method='POST', url='server_url',
   json={'content':'#OnForSec;var1:${steering};var2:${power};var3:${seconds};end'},
   headers={'authorization':'Token'})
   print('Successo')
except:
   print('Fallimento')

wait_ms(500)
```

Figura 33: Protocollo scambio di messaggi blocco ON FOR SECONDS

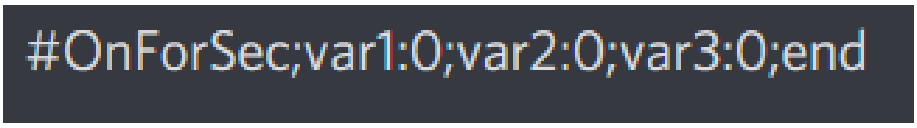


Figura 34: Messaggio Discord [5] blocco ON FOR SECONDS

La stringa visualizzata su *Discord [5]* (Figura 34) presenta 3 variabili: var1 per *steering*, var2 per *power* e var3 per *seconds*.

-*Modalità ON FOR ROTATIONS*: questa modalità riprende la modalità ON FOR SECONDS, ma fornisce, invece di seconds, il parametro *rotations*, ovvero dice al robot dopo quanti giri di ruota arrestarsi. Sono stati riscontrati problemi nella fase implementativa, soprattutto per ideare un sistema che facesse fermare le ruote dopo un certo numero di rotazioni scelte dall'utente, tutte le problematiche sono state trattate nel capitolo 5. Il blocco è stato creato ugualmente, mantenendo attive le opzioni di *steering* e *power*. Il parametro rotations è stato aggiunto ma nel Block Code non è presente il codice implementato. È stato aggiunto anche il protocollo per lo scambio di messaggi e i messaggi su *Discord [5]* vengono visualizzati correttamente.

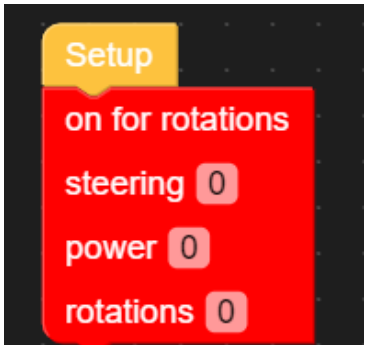


Figura 35: Blocco ON FOR ROTATIONS UIFlow [4]

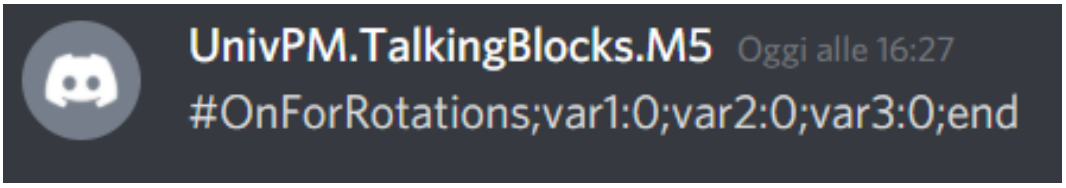


Figura 36: Messaggio Discord [5] blocco ON FOR ROTATIONS

-Modalità ON FOR DEGREES: anche nell'implementazione di questa modalità sono state riscontrate problematiche simili, che verranno ampliate in seguito. Anche in questo caso il blocco è stato creato conservando le funzioni di steering e power. Il protocollo di messaggi è stato implementato e funziona correttamente.

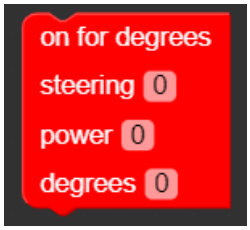


Figura 37: Blocco ON FOR DEGREES UIFlow [4]

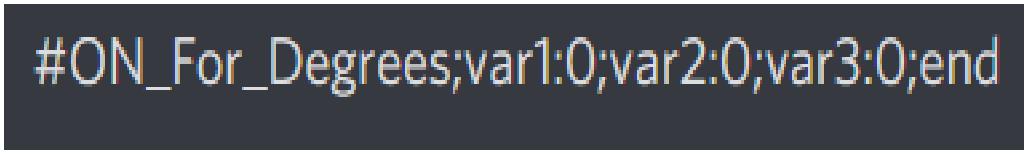


Figura 38: Messaggio Discord [5] blocco ON FOR DEGREES

4.2 Validazione e verifica

Per verificare l'efficacia e l'accuratezza del lavoro effettuato sono stati effettuati dei test, sia sul protocollo per lo scambio di messaggi che sul funzionamento dei blocchi su *UIFlow [4]*, mettendo a confronto i due kit, e i risultati sono stati schematizzati graficamente. Per prima cosa, per controllare l'efficacia dello scambio di messaggi, sono stati eseguiti tutti i blocchi presenti nella libreria *Move_Steering* per diverse volte, per vedere quanti messaggi arrivavano su *Discord [5],* e se eventualmente qualcuno veniva omesso. Per il blocco OFF è stata creata una sequenza su *UIFlow [4]* di 10 blocchi consecutivi, e mandati in esecuzione per 10 volte, mentre per gli altri blocchi, la memoria dell'M5 si saturava, dando in output un errore nell'allocazione della memoria, quindi, dopo delle prove si è stabilito che l'M5 riesce ad eseguirne fino a 6 insieme. Inoltre, dopo diverse esecuzioni (4/5) della pila di blocchi, la memoria dell'M5 si saturava ed è stato necessario riavviarlo per risolvere il problema.

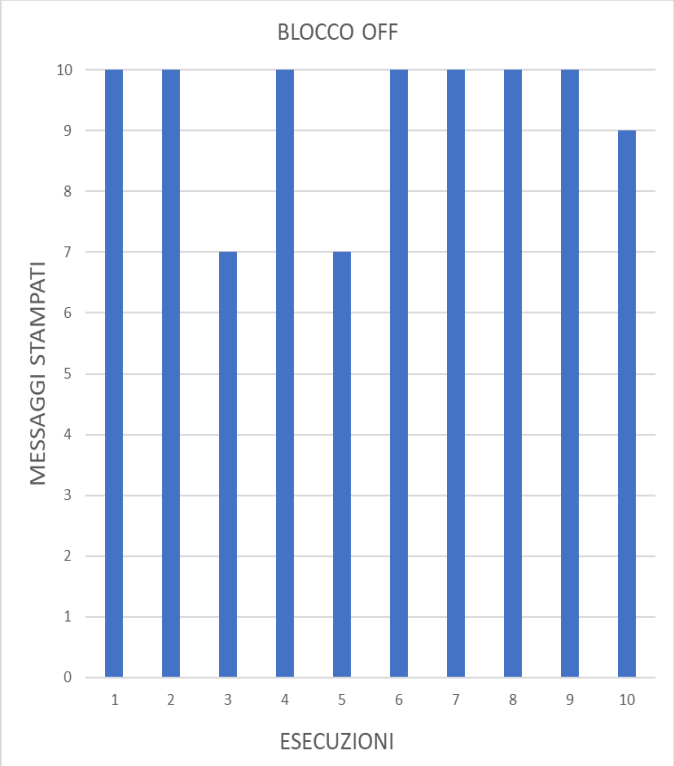


Figura 39: Test messaggi blocco OFF

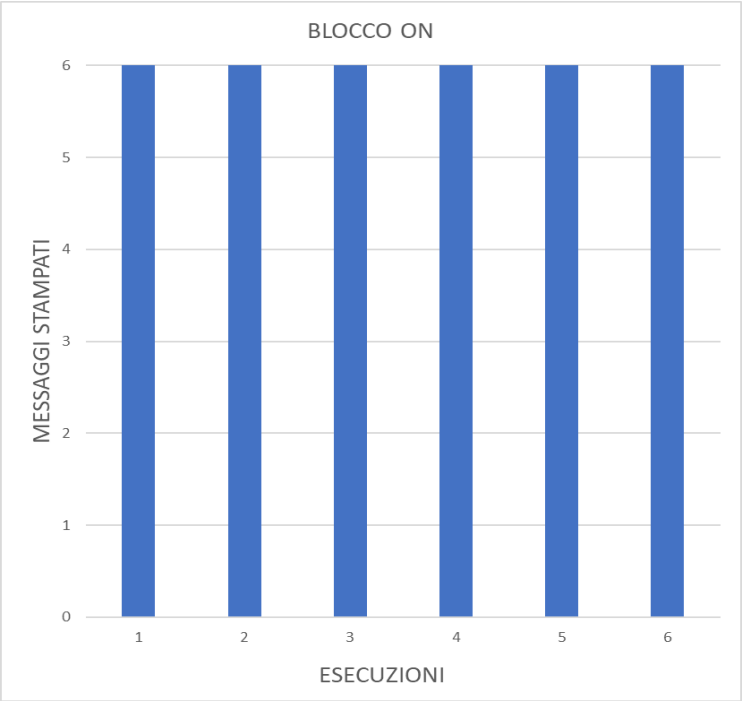


Figura 40: Test messaggi blocco ON

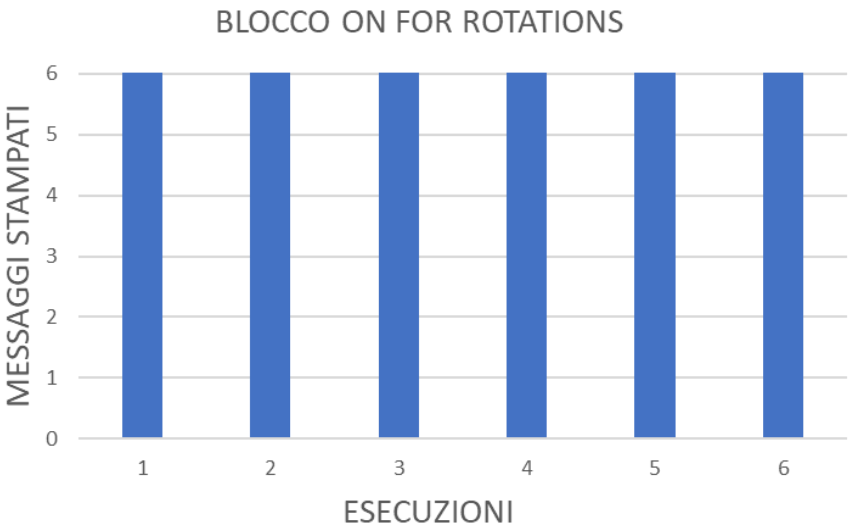


Figura 41: Test messaggi blocco ON FOR ROTATIONS

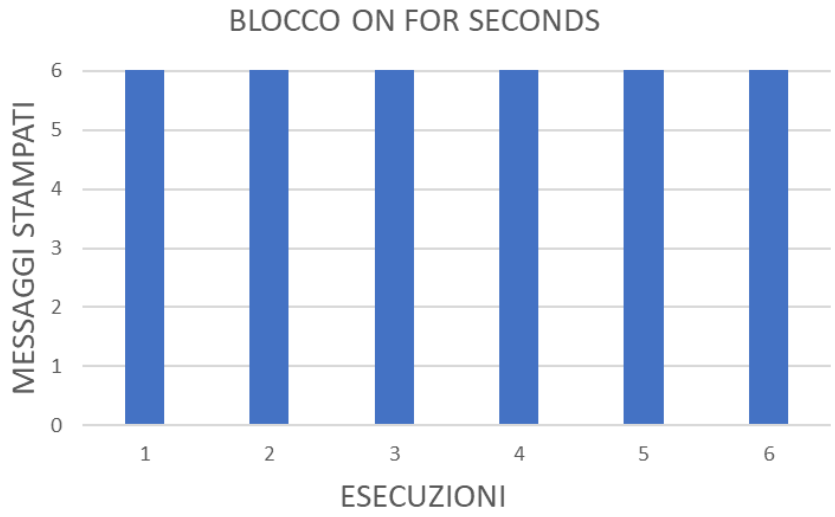


Figura 42: Test messaggi blocco ON FOR SECONDS

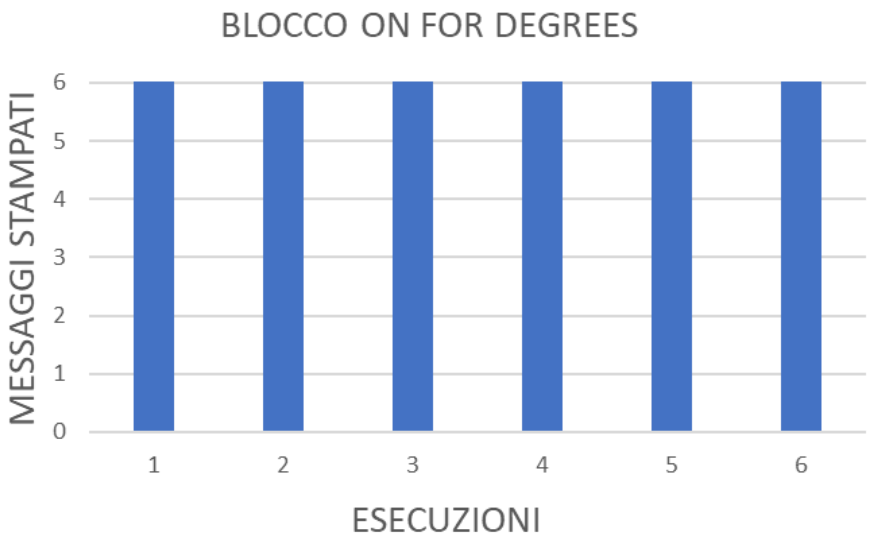


Figura 43: test messaggi blocco ON FOR DEGREES

In seguito, è stato effettuato un test pratico per confrontare i movimenti dei due robot e la fedeltà ottenuta nel Carro *M5StickC [2],* eseguendo due programmi (Figure 40 e 41) dalla struttura analoga sia sul software LEGO che su *UIFlow [4]*. I

risultati sono stati filmati in due clip, allegate nel *materiale multimediale online [22]*. Come si può notare, in entrambi i test il comportamento del carro *M5StickC [2]* ha emulato in modo corretto i movimenti del *LEGO MINDSTORMS*, un'unica differenza visibile è il ritardo della partenza del robot M5, che è dovuto alla latenza introdotta dal protocollo per lo scambio di messaggi. La latenza può essere eliminata per fornire al carro M5 maggiore reattività, a scapito di una minore efficienza del server *Discord [5]* (poiché ricordiamo che la latenza è stata introdotta per ridurre la percentuale di perdita dei messaggi).

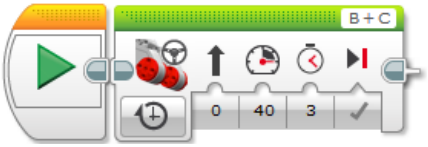


Figura 44: programma test LEGO

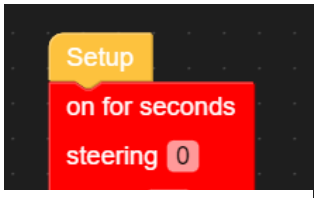


Figura 45: programma test UIFlow [4]

4.3 Lezioni sui sistemi di robotica implementati

I risultati ottenuti in questa tesi trovano un possibile riscontro pratico in delle lezioni da tenere in delle classi di studenti di scuola primaria, in cui gli insegnanti potranno applicare gli strumenti finora descritti in un progetto di robotica educativa. I docenti, attraverso delle slide in ppt, forniranno agli studenti alcune nozioni di base, essenziali per poi lavorare sul sistema di robotica educativa Carro *M5StickC [2]*. Le lezioni saranno articolate come segue:

- **Lezione 1**: **nozioni base di geometria**

  Agli studenti saranno illustrati dei concetti chiave di geometria, quali cerchio, raggio e circonferenza, che saranno indispensabili per l'applicazione pratica dei sistemi di robotica educativa.

- **Lezione 2: corso pregresso di robotica**

  Primo approccio al concetto di robotica, per la comprensione del funzionamento di un robot e della sua interazione con l'ambiente circostante, grazie a dispositivi come sensori e attuatori, coordinati a loro volta dall'intelligenza artificiale, che permette loro di mettersi in relazione con quello che percepiscono, risolvere problemi e agire verso un obiettivo specifico.

- **Lezione 3: costruzione del robot**

  Agli studenti verrà fornito il kit robotico di base Carro *M5StickC [2],* che con la supervisione dei docenti verrà assemblato seguendo un apposito *manuale di istruzioni [22].*

Lezione 4: programmazione del robot con esempi

Verranno proposti degli esercizi base per introdurre gli studenti alla programmazione del robot:

- **Esercizio A:** far percorrere al robot esattamente 1 metro. Gli studenti potranno utilizzare i blocchi forniti dalla libreria *Move Steering* per far muovere il robot, e per esempio calcolare, dopo aver appreso i concetti di raggio e circonferenza, dopo quanti secondi esso avrà percorso la distanza data. A quel punto potranno utilizzare un blocco ON FOR SECONDS per far avanzare il robot fino ad 1 metro, oppure utilizzare il blocco ON per tot secondi e il blocco OFF per far arrestare il robot una volta raggiunto l'obiettivo. Inoltre, dovranno prestare attenzione alla velocità scelta, poiché il timer cambierà per ogni velocità. La Figura 46 rappresenta un possibile svolgimento dell'esercizio.

Calcolando il raggio e la circonferenza, è possibile risalire al numero di giri che deve compiere il robot per percorrere una certa distanza, se poi si moltiplicano 60 secondi per i giri e si divide per la velocità si ottiene il tempo necessario per raggiungere l'obiettivo scelto. L'esercizio viene considerato accettabile se il robot si ferma a meno di 2.5 cm dalla distanza data.

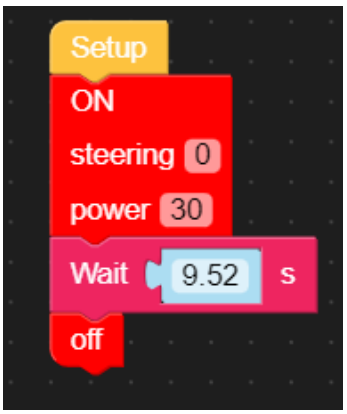
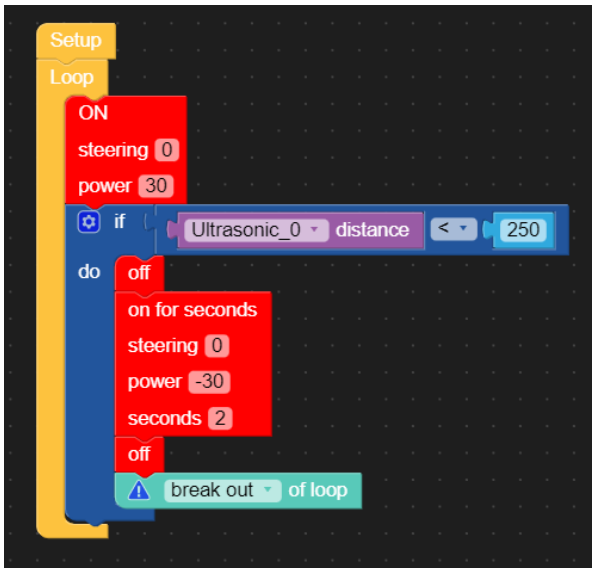


Figura 46: Esempio esercizio A

- **Esercizio B:** movimento con rilevamento di un ostacolo. Il Carro *M5StickC [2]* monta nella parte anteriore un *Ultrasonic sensor [18]* che rileva la distanza del robot dagli oggetti. Su *UIFlow [4]* è presente una libreria dedicata al sensore che viene sbloccata importandolo dalle unità. L'obiettivo è azionare il robot in avanti ed arrestarlo a 25 cm da un ostacolo. Un possibile esempio viene riportato nella figura 47. Viene inserito un *loop* che mantiene il carro in movimento finché la distanza rilevata dal sensore non è inferiore a 250 mm (25 cm), i motori vengono arrestati, il robot indietreggia per 2 secondi e si ferma.



L'ultimo comando, *break out of loop*, serve per uscire dal loop e interrompere la sequenza di istruzioni.

Figura 47: Esempio esercizio B

- **Lezione 5:** verifica finale. Attraverso una competizione a squadre, viene assegnato agli studenti il compito di assemblare il robot e risolvere 2 esercizi simili a quelli proposti precedentemente in un tempo limitato, concedendo loro 5 tentativi.

Quest'applicazione pratica prende spunto dal protocollo di *Educational Data Mining (EDM) and Learning Analytics (LA)* descritto nell'abstract "Utilizing data mining with Robotics for identification and assessment of educational experiences" *(Scaradozzi [15], Cesaretti, Screpanti, & Mangina).*

5 Limitazioni e sviluppi futuri

In questo capitolo verranno analizzate le problematiche riscontrate in fase realizzativa e implementativa di alcuni blocchi in *UIFlow [4],* e in seguito proposte delle alternative per migliorare il kit robotico Carro *M5StickC [2].*

Per quanto riguarda il blocco ON FOR ROTATIONS, non è stata trovata una soluzione per implementare le rotazioni, a causa soprattutto della mancanza di un encoder che contasse i giri, che è invece presente nel blocco LEGO EV3. Non montando questo dispositivo, l'unico modo per far fermare il carro *M5StickC [2]* dopo tot rotazioni è stato impostare un timer, rispondendo alla domanda ''in quanto tempo il robot compie tot giri?''. Per fare ciò sono state effettuate delle misurazioni per sapere in quanto tempo la ruota compiesse un giro a 30 di potenza (2.2 secondi), ed è stata ideata una proporzione del tipo: 1 giro: 2.2 sec = rotations : x , dove x è il tempo che impiega il robot per effettuare le rotazioni scelte attraverso il parametro rotations. In questo modo si riesce a far arrestare il Carro *M5StickC [2]* al momento giusto. Il problema subentra al variare della velocità (poiché cambia anche il timer), quindi bisognerebbe conoscere per ogni velocità in quanto tempo viene compiuto un giro. Un ulteriore problema si è verificato notando che l'impulso dato ai motori come potenza (misurato in uA) non è direttamente proporzionale ai giri, ovvero se il robot fa 1 giro in 2.2 secondi ad una potenza di 30, a potenza doppia (60) dovrebbe fare 2 giri in 2.2 secondi, invece non è cosi. Per validare questa teoria sono stati effettuati dei tests a 3 potenze diverse (Figura 48). Inoltre si è notato che il livello della batteria dell'*M5StickC [2]* influisce sulla precisione delle operazioni e dei movimenti del robot, infatti i timer del test della Figura 48 cambiano se il blocchetto M5 ha la batteria al minimo.

```
30 uA: 1 giro in 2.55 secondi
       2 giri in 4.85 secondi
       3 giri in 7.5 secondi

45 uA: 1 giro in 1.4 secondi
       2 giri in 2.8 secondi
       3 giri in 4.8 secondi

60 uA: 1 giro in 1.15 secondi
       2 giri in 2.30 secondi
       3 giri in 3.45 secondi
```

Figura 48: Test timers a differenti velocità

Per quanto riguarda invece la modalità ON FOR DEGREES le problematiche sono pressocchè simili a quelle della modalità ON FOR ROTATIONS. Per far avanzare il carro *M5StickC [2]* fino al raggiungimento dei gradi scelti

dall'utente, bisognava inserire un wait(t) dopo l'azionamento dei motori, in cui t è il tempo che impiega il robot per eseguire i degrees scelti. Tuttavia rimane il problema relativo alle velocità e all'impulso.

In sostanza, le migliorie più significative da applicare al kit potrebbero essere implementare un encoder, che risolverebbe il problema del contagiri, e di un sensore giroscopico, che consentirebbe al robot di avanzare per gradi ed effettuare svolte più precise. Mentre un'alternativa più performante vede l'utilizzo del kit *RoverC(W/O M5StickC [2]) - Omnidirectional Robot Base*, dotato di strumenti più avanzati, e che offrirebbe una molteplicità di funzioni aggiuntive.

Conclusione

Alla fine del presente lavoro si può affermare che gli obiettivi inizialmente preposti sono stati per la quasi totalità raggiunti. È stato infatti realizzato un kit robotico, il Carro *M5StickC [2]*, ottenuto basandosi sul *LEGO MINDSTORMS Education EV3 [11]*, utilizzato nell'abstract "*Utilizing data mining with Robotics for identification and assessment of educational experiences".* Durante il percorso sono sorte problematiche riscontrate soprattutto nell'implementazione di alcuni comandi a causa della mancanza di strumenti essenziali per le misurazioni. L'assenza di questi ultimi ha richiesto la ricerca di soluzioni alternative e l'utilizzo di proporzioni e formule matematiche funzionanti e al contempo comprensibili, tali da poter essere applicate in classi di studenti. È stato invece possibile tenere traccia dei blocchi utilizzati in *UIFlow [4]* grazie alla corretta realizzazione dei *Talking Blocks.*

In conclusione, l'implementazione del Carro *M5StickC [5]* potrà essere un valido strumento nel campo della robotica educativa, che grazie a miglioramenti in sviluppi futuri esprimerà ancora di più il suo potenziale. In questo modo sarà possibile introdurre classi di studenti nel futuro e vasto mondo della robotica.

Bibliografia e Sitografia

[1] M5Stack. (s.d.). Tratto da M5Stack: https://m5stack.com/

[2] M5StickC ESP32-PICO Mini IoT Development Kit. (s.d.). Tratto da M5Stack Store: https://shop.m5stack.com/products/stick-c?variant=17203451265114

[3] BALA-C ESP32 Development Mini Self-Balancing Car. (s.d.). Tratto da M5Stack Store: https://shop.m5stack.com/products/bala-c-esp32-development-mini-self-balancing-car?variant=32128285409370

[4] UIFlow. (s.d.). Tratto da M5Flow: https://flow.m5stack.com/

[5] Discord. (s.d.). Tratto da Discord: https://discord.com/

[6] Arduino. (s.d.). Tratto da Arduino: https://www.arduino.cc/

[7] Chrome DevTools. (s.d.). Tratto da Chrome DevTools - Chrome Developers: https://developer.chrome.com/docs/devtools/

[8] BlockMaker. (s.d.). Tratto da M5BlockMaker: http://block-maker.m5stack.com/

[9] Cervigni, M. (A.A 2020/2021). *Studio e Implementazione di Strumenti di Robotica Educativa per l'Identificazione e la Modellazione dell'Apprendimento.* Università Politecnica delle Marche.

[10] Gulesin, R. (A. (Cichella)A 2021/2022). *Studio e Implementazione di Strumenti di Robotica Educativa per l'Identificazione e la Modellazione dell'Apprendimento.* Università Politecnica delle Marche.

[11] LEGO MINDSTORMS Education EV3. (s.d.). Tratto da MINDSTORMS EV3: https://education.lego.com/en-us/products/lego-mindstorms-education-ev3-core-set/5003400#lego-mindstorms-education-ev3

[12] Software LEGO MINDSTORMS Education EV3 v1.4.5. (s.d.). Tratto da MINDSTORMS EV3 Downloads: https://education.lego.com/en-us/downloads/retiredproducts/mindstorms-ev3-lab/software

[13] Brick LEGO EV3. (s.d.). Tratto da Mattoncino Intelligente EV3 45500: https://www.lego.com/it-it/product/ev3-intelligent-brick-45500

[14] Dr. Explain. (s.d.). Tratto da Dr. Explain: https://www.drexplain.it/

[15] Scaradozzi, D., Cesaretti, L., Screpanti, L., & Mangina, E. (s.d.). Identification and Assessment of Educational Experiences: Utilizing Data Mining With Robotics. *IEEE Robotics & Automation Magazine*.

[16] Sensore a Ultrasuoni LEGO. (s.d.). Tratto da Sensore ad ultrasuoni EV3 45504: https://www.lego.com/it-it/product/ev3-ultrasonic-sensor-45504

[18] Ultrasonic Sensor M5Stack. (s.d.). Tratto da Ultrasonic Distance Unit (RCWL-9600): https://shop.m5stack.com/products/ultrasonic-distance-unit-rcwl-9600

[19] BugC (W/O M5StickC) Programmable Robot Base. (s.d.). Tratto da BugC (W/O M5StickC) Programmable Robot Base: https://shop.m5stack.com/products/bugc-w-o-m5stickc?variant=30885082497114

[20] RoverC (W/O M5StickC) - Omnidirectional Robot Base. (s.d.). Tratto da RoverC(W/O M5StickC) - Omnidirectional Robot Base: https://shop.m5stack.com/products/rovercw-o-m5stickc?variant=31185881792602

[21] Istruzioni di montaggio LEGO. (s.d.). Tratto da MINDSTORMS EV3 Support: https://education.lego.com/en-us/product-resources/mindstorms-ev3/downloads/building-instructions

[22] Cichella, L. (s.d.). *[22] materiale tesi.* Tratto da materiale tesi - Dropbox: https://www.dropbox.com/sh/s8syziljyr8umsw/AADs4uhHUtbFSSQ2TNT8r8Jpa?dl=0