

RoboSusa

La Squadra:

Adrian Vasile, (caposquadra), programmatore senza esperienze passate.

Luigi Vair, addetto alla reperibilità della componentistica del robot senza esperienze passate.

Fulvio Tournoud, (co-caposquadra), addetto alla costruzione dell'hub motori e montaggio sensori senza esperienze passate.

Paolo Rocci, addetto alla costruzione del corpo movimento robot (cingoli) e montaggio sensori senza esperienze passate.

Marco Di Bella, addetto alla costruzione braccio meccanico (benna) e motori riguardanti essa senza esperienze passate.

L'obiettivo primario alla partecipazione della RoboCup è accrescere il nostro bagaglio culturale attraverso l'esperienza di una gara di gruppo di robotica la quale auspichiamo possa accrescere le nostre conoscenze nell'ambito della robotica.

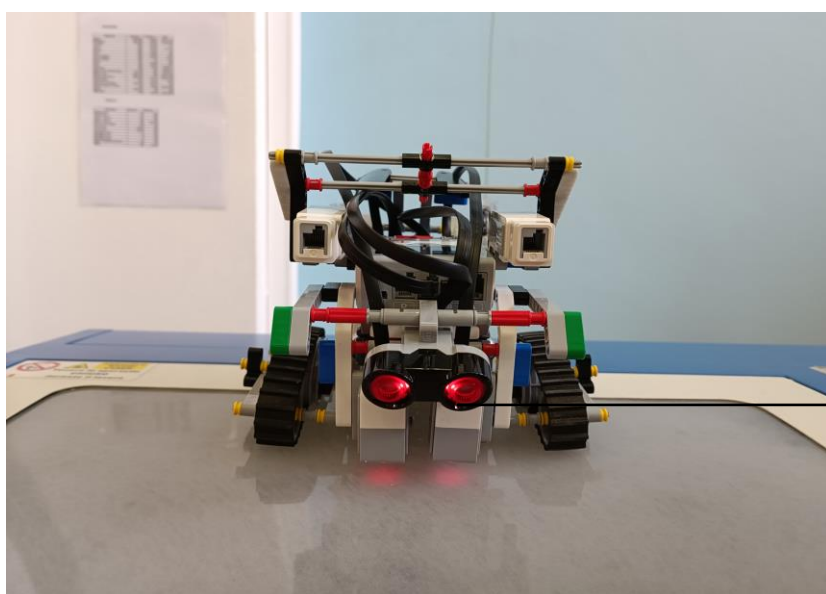
Per questa competizione il gruppo ha deciso di creare un robot con un movimento cingolato. I cingoli in questione sono composti di gomma, la struttura del cingolo è di forma triangolare grazie alla presenza di 3 cerchioni. Il movimento cingolato viene generato da 2 motori collegati alla parte anteriore dei cingoli.

Per il movimento del robot sono stati utilizzati 4 sensori, uno di distanza a ultrasuoni che permette il riconoscimento di eventuali ostacoli sul percorso e 3 sensori di colore che hanno rispettivamente 2 funzioni. Le funzioni dei sensori di colore sono: 2 sensori per l'inseguimento della linea nera e il riconoscimento delle intersezioni e 1 sensore per il riconoscimento delle vittime (nere se sono morte, argento se sono vive) per poterle trasportare sulla benna. Attaccata al centro di comando del robot vi è una benna per il trasporto delle vittime essa è collegata a 2 motori che le permettono di compiere due movimenti differenti grazie a dei riduttori. Il primo movimento serve per serrare la vittima portando la benna verso l'interno del robot. Il secondo movimento consiste nel

poter abbassare e alzare la benna da terra. Nella parte centrale del robot sono stati posti 3 sfere in metallo per bilanciare il suo peso e in aggiunta vi è un quarto peso nella parte posteriore. Nella parte superiore del robot collegata al brick centrale vi è una maniglia per poter trasportare il robot.

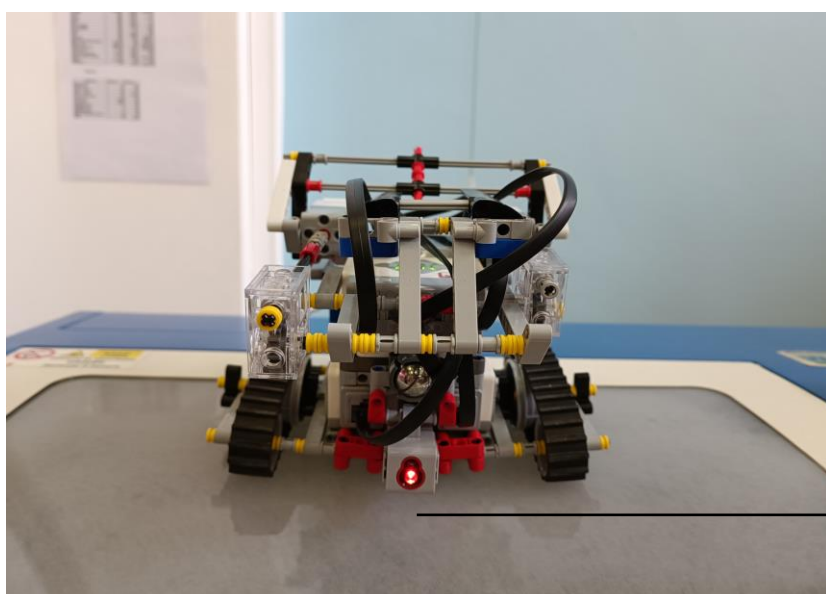
- Con la costruzione di questo robot e la sua programmazione siamo riusciti a far inseguire dal robot una linea nera equivalente a quella del percorso della RoboCup Junior. Il robot in questione riesce a superare anche pezzi di linea tratteggiata. Il robot riesce a fare curve e a superare incroci di 90° contrassegnati dal colore verde. Il robot riesce a superare con successo anche rampe con una pendenza non superiore a 30° grazie al movimento cingolato. Il robot riesce a superare anche ostacoli come altalene.
- Prima di creare la versione finale del robot, il nostro gruppo ha dovuto sperimentare e creare varie versioni. In primo luogo un problema è stato quello di trovare una tipologia di movimento da installare sul robot. La prima forma di locomozione installata su di esso era formata da 4 ruote. 2 di esse erano collegate a 2 motori indipendenti, creando un robot a trazione anteriore. Il problema riscontrato nella sperimentazione diretta del robot sul percorso però, era l'incapacità di esso di compiere curve molto strette non riuscendo ad avere un movimento sterzante sul robot. La seconda forma di locomozione installata sul robot allora fu quella cingolata a forma triangolare con cingoli in plastica; l'idea di partenza era quella di poter compiere un movimento sterzante su sé stesso facendo agire i motori in direzioni opposte, come gli escavatori. Successivamente tuttavia il robot non era in grado di compiere delle salite a causa dello scarso attrito generato dal contatto della plastica sulle mattonelle del percorso. La terza e ultima forma di locomozione sperimentata fu quella cingolata a forma triangolare con cingoli in gomma. I cingoli in gomma a differenza dei precedenti avevano una consistenza tale da creare un attrito sufficiente per poter superare senza difficoltà salite e rampe presenti nel percorso.
- I membri del nostro gruppo non avendo esperienze nell'ambito della robotica non ha potuto prendere spunto da idee passate. Ciò nonostante visto le esperienze nella nostra scuola di altri gruppi che avevano partecipato a gare di robotica, abbiamo potuto osservare e comprendere i loro modelli utilizzati in passato.
- Il sistema che abbiamo creato è formato da 4 sensori. Vi sono 2 sensori che servono per leggere la linea nera da percorrere. I sensori sono posti al di fuori della linea nera e correggono lo spostamento del robot ogni volta che puntano sulla riga, infatti i sensori leggono il colore bianco. Una volta che i sensori leggono il colore mandano il segnale al brick dove è stato messo un programma che è capace di far muovere i motori di conseguenza. Questo funzionamento dei sensori di colore è equivalente per tutti gli altri sensori di colore presenti nel robot. Di fatto i sensori leggono il colore, mandano il segnale al brick, il brick elabora una risposta in base al programma e manda il movimento ai rispettivi motori. Sul robot è presente un sensore di distanza a infrarossi il quale legge la presenza di

ostacoli sul percorso. Il sensore di distanza, lavora come i sensori precedentemente descritti; esso quando incontra un ostacolo sul percorso (grazie agli infrarossi), manda un segnale al brick il quale elabora come risposta o l'aggiramento dell'ostacolo o indica al robot di tornare indietro. Sul robot è presente anche un braccio con collegata una benna. Il movimento della benna è regolato grazie a due motori, questi ultimi hanno in uscita dei riduttori che servono per rallentare i movimenti dei motori e allo stesso tempo servono da unione per i vari pezzi. Al di sotto della benna vi è un sensore di colore che serve al riconoscimento delle vittime. Il sensore una volta riconosciuto il tipo di vittima manda il segnale al brick che aziona la benna. Dopo aver incastrato la vittima tra due pale, la benna si solleva in posizione orizzontale e poi si chiude su se stessa impedendo alla vittima di cadere. La programmazione della benna non è stata effettuata per motivi di tempo.

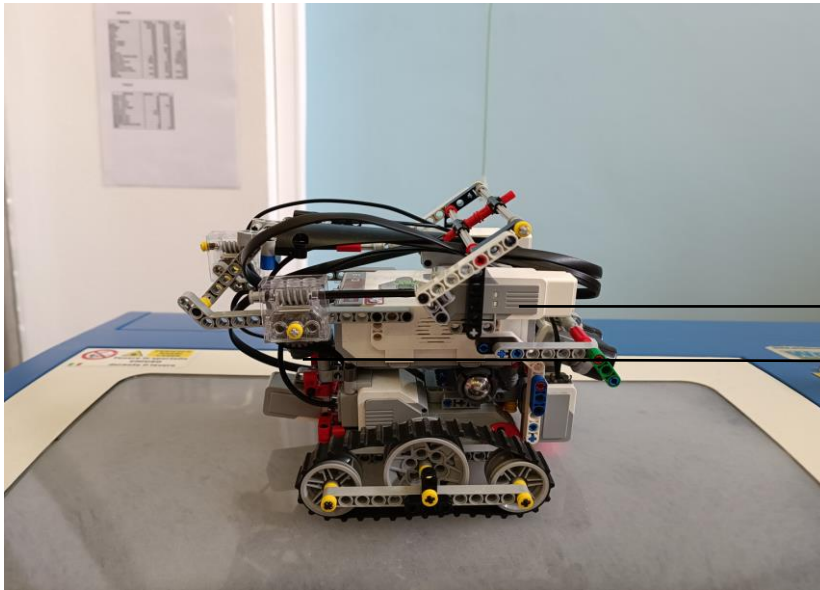


→ Sensore di distanza a infrarossi per la lettura della linea

→ Sensori di colore per la lettura della linea



→ Sensore di colore per il riconoscimento delle vittime



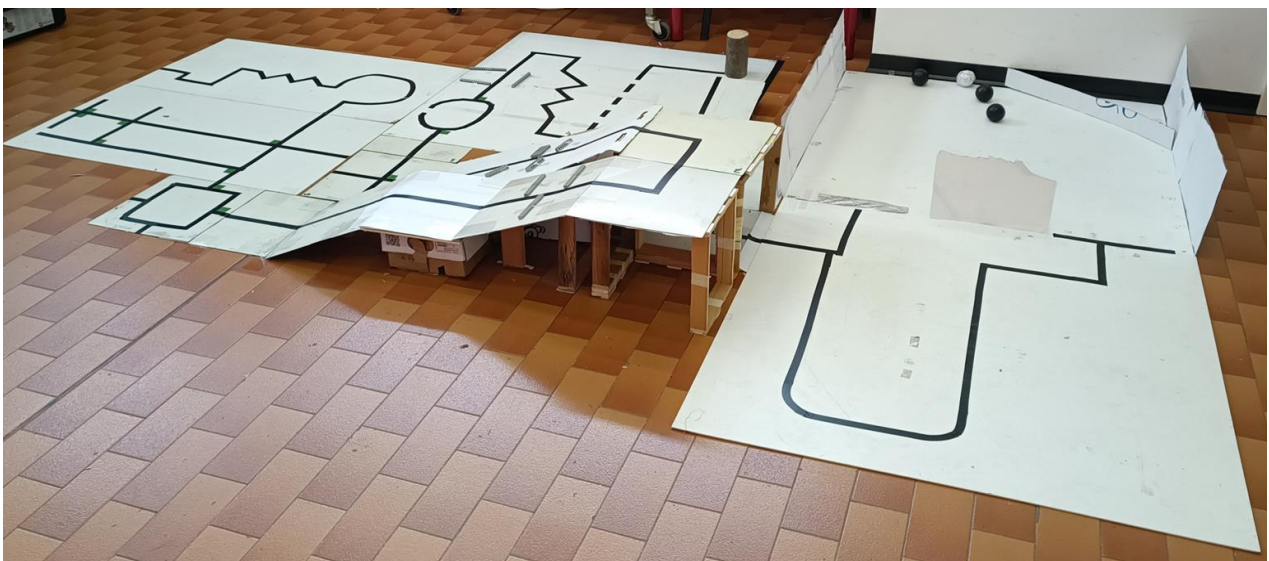
→ Motori benna

→ Riduttori

- Il robot che abbiamo deciso di ideare è composto da due cingoli in gomma di forma triangolare con il cerchione centrale di dimensioni maggiori rispetto a quelli esterni. I motori sono collegati alla parte anteriore dei cerchioni. Nella parte centrale del robot sotto al brick vi sono 3 pesi per bilanciarlo installati su un telaio di dimensioni più piccole. Nella parte posteriore del robot vi è un quarto peso sopra il sensore di colore che serve per leggere lo stato della vittima. Nella parte anteriore del robot vi è un attacco a "L" al quale sono attaccati i sensori di colore. Dai sensori di colore, parte un attacco al quale è agganciato il sensore di distanza a infrarossi. Sul brick sono installati 2 motori per il movimento della benna. Collegati ai due motori vi è una maniglia per poter raccogliere il robot sul percorso. Nella parte posteriore del robot, sul brick sono stati installati due pezzi che attaccati a un pezzo a "L", vanno a comporre l'aggancio per la benna.



- Il robot in generale riesce a compiere il percorso in maniera sufficientemente precisa, riesce a superare altalene, incroci e salite. Il robot non riesce però a recuperare le vittime e portarle nella zona di rescue, compiendo di fatto metà del lavoro richiesto. Tuttavia avendo lavorato solo 2 mesi e mezzo sulla costruzione e il collaudo, il robot è di buona qualità.
- Per verificare il funzionameto del robot, il gruppo ha creato un perorso simile a quello impiegato nella Robocup, inserendo rampe, incroci, altalene, ostacoli e curve. La prima parte del robot che siamo andati a verificare era la sua capacità di compiere curve. Per fare ciò abbiamo creato curve di 90° con la presenza di nastro verde in uno degli angoli della curva per far capire al robot dove curvare. In seguito abbiamo creato percorsi tratteggiati per fare in modo che il robot imparasse a seguire una linea nera anche solo segnata a tratti. Successivamente abbiamo creato rampe con una pendenza massima di 30° per testare il robot sulle salite. Le rampe che abbiamo costruito erano sopraelevate con la possibilità di far passare il robot sotto di esse.



- Avendo ricreato un percorso il più fedele possibile a un ipotetico tracciato della robocup, abbiamo potuto provare il robot sotto vari aspetti, i risultati ottenuti nelle varie prove all'inizio sono stati deludenti ma dopo aver perfezionato al meglio il programma, i risultati sono andati migliorando. Dopo un lungo periodo il robot è riuscito a superare tutte le prove che gli sono state presentate, eccezzion fatta per la rescue.
- Per creare il nostro robot sono stati neccessari 2 mesi e mezzo di fatica e lavoro che però hanno portato i suoi frutti. Sicuramente è stato creato un gruppo molto unito che è riuscito a raggiungere importanti traguardi, dalla realizzazione di un circuito per testare il robot alla creazione di un robot funzionante e di un secondo robot di riserva. Certamente resta il rammarico di non essere riusciti a completare

la programmazione del robot per compiere la rescue. Detto ciò il nostro gruppo spera che il robot sia capace di farsi valere.

Per permettere la massima personalizzazione delle azioni del robot è stato utilizzato micropython per ev3, in questo modo è possibile programmare utilizzando python ma avendo l'aspetto elettronico soddisfatto già dai componenti Lego.

L'ambiente di lavoro utilizza ev3dev su Visual Studio Code.

L'idea di funzionamento è relativamente semplice, seguire una linea in fondo non necessita di una particolarmente complessa programmazione, il problema nasce quando oltre al dover seguire la linea è necessario controllare altre variabili contemporaneamente e in maniera modulare, così permettendo la facile calibrazione del robot.

Il problema principale lo abbiamo nell'affidabilità delle letture date dai vari sensori, infatti il problema principale della programmazione in questo caso è validare tutti gli input, in modo da evitare falsi positivi, che se non gestiti correttamente potrebbero causare vari problemi.

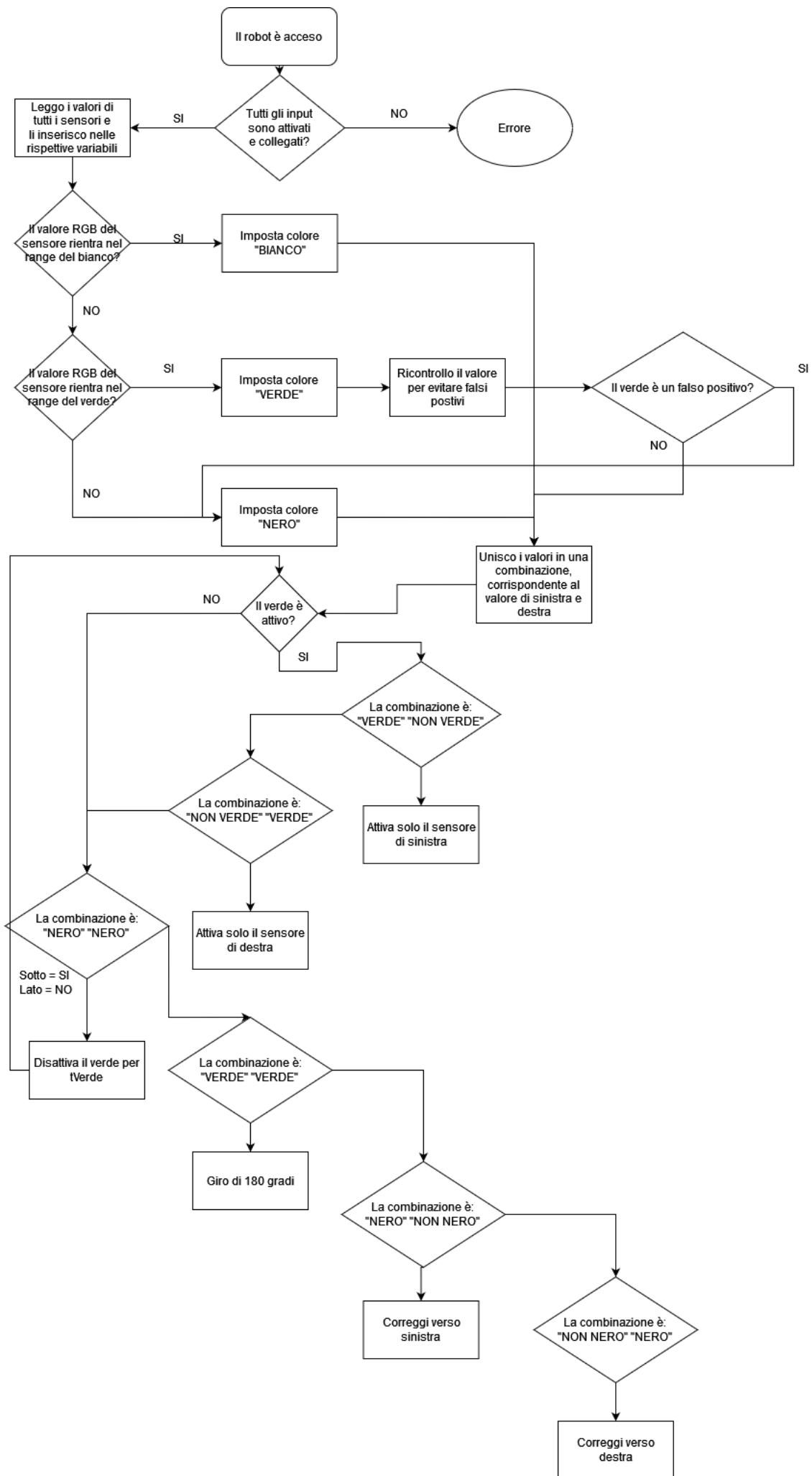
Il problema principale lo abbiamo nelle intersezioni, infatti nonostante il pensiero: "vedo verde a sinistra, giro a sinistra" funzioni in teoria non è così facile integrarlo in maniera affidabile, dato che i sensori (in questo caso) vengono utilizzati attraverso le letture rgb che essi danno e molto probabile che nelle transizioni tra bianco e nero o viceversa il valore corrispondente al verde spunti fuori, il nostro lavoro sta nel verificare che questo sia un falso positivo.

Il modo in cui è stato risolto spiegato in maniera molto semplice consiste nel controllare che nel caso il verde venga visto dal sensore attiviamo una procedura che rallenta il robot nel mentre che vede il verde, se il sensore verde il verde oltre a un certo numero di volte (Threshold del verde) allora possiamo essere abbastanza sicuri che il sensore abbia letto il valore correttamente e possiamo attivare il sottoprogramma che si occupa dell'intersezione.

In questo caso il programma disattiva il sensore opposto al verde e utilizza un solo sensore, assicurando che il sensore non esca dalla linea, e dopo un certo tempo possiamo riattivare entrambi i sensori.

Ecco un "codice" che mostra il processo sotto linea di testo:

```
CONTROLLA SENSORI(Sx, Dx)
SE ( UNO DEI SENSORI VEDE VERDE)
    RALLENTO()
    CONTO(VERDE)
    SE (VERDE > THRESHOLD)
        ATTIVO(FUNZIONE VERDE)
```

Ovviamente se andiamo nei vari intrighi avremo un flowchart decisamente più lungo, ma per capire le funzioni principali è perfetto.

Inoltre il codice per ora non implementa la “stanza” della rescue, questo è a cause di varie problematiche riscontrate durante la fase di programmazione.

Sotto un elenco delle peculiarità del codice:

1. L'utilizzo di micropython per una programmazione più efficace e compatta
2. L'utilizzo diretto dei valori RGB dati dai sensori di colore, difatti essi normalmente possono dare o il valore del colore che vedono oppure l'indice di riflesso del materiale nel loro raggio di azione, il vantaggio di utilizzare i valori RGB vuol dire che possiamo personalmente inserire i valori RGB corrispondenti al verde, ovviamente in un certo Min e Max.
3. Per la prevenzione di stallo il robot utilizza i valori dell'angolo dati dal motore e li confronta a quelli precedenti, se l'angolo non è aumentato di abbastanza significa che il robot non ha compiuto progressi e di conseguenza avvierà una procedura di “anti – stallo”

Il progetto rappresenta una vera e propria sfida, il senso di competitività dato dalla gara in questione ha dato una motivazione molto forte al progetto, inoltre nella creazione di questo robot sono state imparate numerose lezioni che non sarebbero state possibili da spiegare normalmente, la possibilità di progettare il robot e la sua realizzazione sono di se una gigantesca fonte di conoscenza e esperienza, infatti in futuro ci sono molte cose che verrebbero di sicuro fatte diversamente per evitare inutili grattacapo come invece è successo ora, ma tutto sommato per essere la nostra prima competizione di robotica pensiamo di aver fatto un lavoro soddisfacente.

Ringraziamo della possibilità di partecipare a questa competizione.

- *RoboSusa*