

**Rete di scuole per la
ROBOCUP JR ITALIA**

ROBOCUP JR ITALIA 2011

Catania 14-16 aprile

**REPORT DI
DOCUMENTAZIONE**

Della squadra

SpikeBot

**Istituto di Istruzione
Superiore "Ettore Majorana" -
Piazza Armerina (EN)**

ROBOCUP JR ITALIA 2011 - Catania 14-16 aprile REPORT DI DOCUMENTAZIONE

La “Rete di scuole per la Robocup Jr ITALIA” è espressione dell’Autonomia scolastica regolata dal D.P.R. 275/99 (art. 7) che permette alle scuole statali di operare sinergicamente per obiettivi condivisi e ritenuti importanti per l’offerta formativa erogata all’utenza.

La Rete di scuole è nata sulla condivisione di una serie di principi EDUCATIVI e DIDATTICI riferiti alla realtà della scuola italiana. Questi principi e le conseguenti proposte operative erano stati riportati in un documento del maggio 2008 dal titolo: ***Manifesto per una RoboCupJr italiana - una proposta per la diffusione dell’utilizzo didattico della Robotica nelle scuole*** a cura di Andrea Bonarini, Augusto Chiocciariello e Giovanni Marciandò. Maggio 2008

L’obiettivo della Rete - organizzare l’edizione italiana della Robocup Jr - concretizza una spinta al confronto e alla collaborazione tra Istituti scolastici, elementi che motivano docenti e studenti all’impegno nell’innovazione, sia didattica che tecnologica, affrontando i problemi che costituiscono uno standard internazionale dal 2000, quando la Robocup (manifestazione riservata alle Università di tutto il mondo) ha proposto le tre “gare” per la scuola: Dance - Rescue - Soccer.

La Robocup Jr ITALIA è Una manifestazione nazionale fondata di tre punti forti:

1. una **struttura** che cura l’organizzazione e gestisce gli aspetti di organizzazione,



ROBOCUP JR ITALIA 2011 - Catania 14-16 aprile REPORT DI DOCUMENTAZIONE

promozione, svolgimento ai diversi livelli, regionali e nazionali;

2. un **contenuto** condiviso, ovvero regolamenti, formule di gara, supporto formativo e informativo ai partecipanti;

3. una **documentazione** delle proposte didattiche e del lavoro degli studenti che coinvolgono l'uso di kit o robot auto costruiti per la partecipazione agli eventi organizzati dalla Rete. Questo volumetto appartiene alla collana di documentazione.

Sul piano organizzativo e gestionale della Rete di scuole lo Statuto prevederà organismi ben distinti ma fortemente integrati:

COMITATO DI GESTIONE - formato dai Dirigenti scolastici degli Istituti fondatori o associati alla Rete. Si riunisce due volte l'anno in via ordinaria, e online per decisioni straordinarie.

ISTITUTO CAPOFILA - come previsto dal DPR 275/99 cura gli aspetti burocratici, amministrativi e contabili della Rete. Il Dirigente scolastico dell'Istituto capofila è il legale rappresentante della Rete e provvede a dare esecuzione alle delibere del Comitato di Gestione.

COMITATO TECNICO - formato dai docenti referenti degli Istituti fondatori o associati alla Rete, provvede a definire il Bando e i Regolamenti di gara per la manifestazione annuale nazionale, trasmettendoli al Comitato di gestione che li deve approvare.

RICERCATORI ASSOCIATI - portano nella Rete il loro contributo scientifico di alto livello. Partecipano di diritto al Gruppo Tecnico e un loro rappresentante al Comitato di Gestione (senza diritto di voto).



ROBOCUP JR ITALIA 2011 - Catania 14-16 aprile
REPORT DI DOCUMENTAZIONE

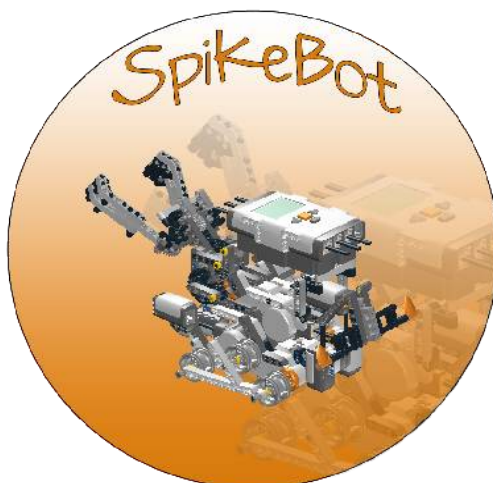
ISTITUTI PARTECIPANTI - iscrivendosi alle gare, beneficiano del supporto della Rete ma non partecipano alle decisioni gestionali o tecniche. La partecipazione alla gara nazionale li rende idonei per aderire alla Rete. Diversamente serve il parere del Comitato Tecnico.



ROBOCUP JR ITALIA 2011 - Catania 14-16 aprile
REPORT DI DOCUMENTAZIONE

autori

**Enrico Chiolo
Andrea Giarrizzo
Andrea Ogliaro
Emanuele Moschetti
Calogero Laneri**



PARTECIPA ALLA GARA DI RESCUE

**Istituto di Istruzione Superiore
"Ettore Majorana" - Piazza
Armerina**

Classi 3[^] e 4[^] Informatica



CAP. 1 - DATI GENERALI

Istituto: Istituto di Istruzione Superiore “Ettore Majorana” con annesso Istituto Tecnico Industriale

Docente Responsabile: Amorelli Giancarlo

Capitano Software: Chiolo Enrico

Capitano Hardware: Giarrizzo Andrea

Componenti: Ogliandolo Andrea, Laneri Calogero, Moschetti Emanuele.

Informazioni utili:

Nella formazione della squadra del Rescue i docenti hanno scelto tra di noi ragazzi, quelli che avevano già una conoscenza abbastanza approfondita del linguaggio di programmazione C, in modo da ridurre al minimo il periodo alfabetizzazione del linguaggio NXC, per concentrarci di più nella progettazione e nella elaborazione software di SpikeBot.



FIG. 1 - ANDREA GIARRIZZO DURANTE IL MONTAGGIO DEL PRIMO SPIKE

Per quanto riguarda la parte meccanica invece abbiamo utilizzato molto della piattaforma che l'anno precedente il nostro compagno Andrea Giarrizzo aveva realizzato e portato a Vicenza, ciò ci ha permesso di migliorare in poco tempo soprattutto il sistema di cattura, distribuire meglio il peso



ROBOCUP JR ITALIA 2011 - Catania 14-16 aprile
REPORT DI DOCUMENTAZIONE

importante per la salita della rampa, ottimizzare la disposizione dei sensori, anche in base all'esperienza dell'anno scorso.

Altro componente della squadra di Rescue che vogliamo citare ma che è rimasto a scuola, dato il numero limitato dei componenti a cinque è Di Silvestre Vincenzo che quest'anno si è classificato primo nel nostro istituto durante le olimpiadi di Informatica.

Enrico Chiolo, Andrea Ogialoro ed Emanuele Moschetti frequentano la classe 3[^] A, mentre Andrea Giarrizzo e Calogero Laneri frequentano rispettivamente le classi 4[^] sezz. A e B sempre della specializzazione Informatica "Abacus" dell'Istituto Tecnico Industriale.

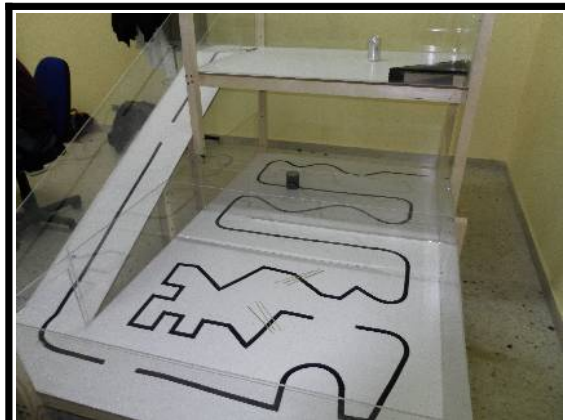


FIG. 1A - IL NOSTRO CAMPO DI GARA



CAP. 2 - DATI DI CONTESTO E MOTIVAZIONE

La partecipazione alla Robocup junior è decisamente una sfida per noi ragazzi. ci è stata lanciata dai nostri insegnanti di informatica Giancarlo Amorelli e Angelo Vallelunga, che puntualmente ogni anno coinvolgono i propri alunni proponendogli la sfida della robotica, facendo crescere tra di noi la passione per l'informatica. Sin da quando noi ragazzi della squadra frequentavamo il secondo anno di istituto superiore eravamo in continua ricerca dei professori del settore informatico per le nostre delucidazioni e la voglia di sapere. E' facile immaginare quindi il nostro entusiasmo nell'essere arrivati a questo punto, forti di aver fatto un buon lavoro di squadra. Un plauso inoltre va alle nostre famiglie che hanno contribuito notevolmente facendo dei viaggi per venirci a prendere il pomeriggio a scuola da paesi anche abbastanza lontani e permettendoci di frequentare il corso di robotica.



FIG. 2 - LO STUDIO DEI COMANDI DI NXC



ROBOCUP JR ITALIA 2011 - Catania 14-16 aprile REPORT DI DOCUMENTAZIONE

L'esperienza acquisita da noi ragazzi in occasione della competizione alla RoboCup Junior di certo è qualcosa che rimarrà indelebile nella nostra preparazione.

Grazie all'ausilio di professori tanto qualificati quanto preparati siamo riusciti a portare a termine un lavoro di certo per noi complicato essendo un'esperienza con cui mai prima d'ora abbiamo avuto a che fare. Il lavoro è stato suddiviso fin da subito in vere e proprie "squadre" come organismi di uno stesso sistema per arrivare alla sua realizzazione, un robot in grado di autogestirsi completamente in situazioni più o meno complesse.

I gruppi di lavoro avevano compiti diversi ad ogni incontro pomeridiano per il laboratorio di robotica, in modo tale da occuparci un po' per volta tutti di ogni campo relativo all'attività: progettazione, assemblaggio, programmazione dei robot.



CAP. 3 - NOME E STRUTTURA DEL ROBOT



FIG. 3 - ANDREA AL LAVORO SUL NUOVO SPIKE

Siamo molto orgogliosi di aver chiamato il nostro robot SpikeBot in onore del primo robot LEGO assemblato con la forma di scorpione e programmato nella nostra scuola un paio

di anni fa.

Come metodo di cattura per la vittima della terza stanza a SpikeBot sono state costruite delle pinze poco ingombranti con materiale Lego e con l'aggiunta di ritagli di gomma in modo da rendere la presa della vittima più sicura ed affidabile.



FIG. 4 - I PEZZI LEGO DIVENTATI SPIKEBOT



CAP. 4 - MECCANICA



FIG. 5 - IL "NOSTRO" MAGAZZINO

SpikeBot è attrezzato con tre motori, due dei quali fissati tra loro tramite supporti realizzati con componentistica LEGO®, studiate attentamente per realizzare la base del robot.

I due motori delegati al movimento si suddividono in motore destro e sinistro, e sono collegati ad una ruota ciascuno, che svolge la funzione di trazione per il cingolo in gomma morbida.

I supporti per i cingoli, per ottenere maggiore stabilità, sono fissati con un barretta lego da 13.

I cingoli in gomma morbida sono stati posti in trazione utilizzando tre supporti (ruote), per ogni cingolo, sistemati in maniera da creare un supporto triangolare che consente una maggiore fluidità nel moto.

La scelta di utilizzare i cingoli in gomma si è resa necessaria in seguito a numerosi problemi legati all'equilibrio durante il superamento di macerie piccole e grandi ed alla mancanza di grip durante la fase di salita lungo la rampa.



ROBOCUP JR ITALIA 2011 - Catania 14-16 aprile REPORT DI DOCUMENTAZIONE

Sulla base sono inoltre fissati diversi supporti per i sensori che utilizziamo.

Per quanto riguarda il terzo motore, esso è fissato alla base con i due motori delegati al movimento, tramite un' impalcatura realizzata con dei componenti LEGO® di tipo L.



FIG. 6 - PARTICOLARE COSTRUTTIVO DEL MECCANISMO DI CATTURA VITTIMA

Il robot si serve del terzo motore per gestire il funzionamento della chela. Essa è stata posizionata al centro del robot in modo da favorire una migliore presa dalla vittima, è composta da un blocco di ruote dentate che a loro volta sono fissate al

meccanismo rotante del motore. Queste fanno sì che se il motore ruota in avanti la chela si abbassa, posizionandosi al centro, e si apre quando la chela raggiunge la posizione orizzontale, mentre se gira all'indietro essa si chiude e subito dopo si alza. Per ottenere un attrito maggiore nella presa della vittima sono stati applicati diversi supporti in gomma.



FIG. 7 - SPIKEBOT NELLE PRIME VERSIONI



ROBOCUP JR ITALIA 2011 - Catania 14-16 aprile REPORT DI DOCUMENTAZIONE

L'intero schema di montaggio in 3D di SpikeBot, realizzato da noi con LEGO DIGITAL DESIGNER, costituisce l'allegato 1 al presente report e ne dettaglia tutte le fasi costruttive, per realizzare SpikeBot. In esso però i sensori non realizzati da LEGO non sono presenti ed LDD non prevede la possibilità di realizzarli, quindi sono stati sostituiti con altrettanti sensori LEGO aventi la stessa forma.

Infine per rendere SpikeBot, oltre che inquietante, anche un po' simpatico abbiamo aggiunto un omino LEGO® che guida il tutto, anche perché dietro ogni robot c'è sempre la mente umana!



CAP. 5 - UNITÀ DI CONTROLLO



FIG. 9 - IL BRICK LEGO

SpikeBot si avvale di un'unità centrale NXT 2.0, il brick, un micromattone-computer della azienda di giocattoli LEGO, dotato di processore Atmel ARM7 che permette di programmare in maniera versatile il robot nei linguaggi più disparati come ad esempio il linguaggio grafico NXT-G, Labview, o con i linguaggi NXC, NBC, leJOS NXJ e RobotC. I programmatori hanno quindi l'imbarazzo della scelta.

Il brick di NXT 2.0 consente di collegare un massimo di 3 motori e ricevere informazioni da 4 porte sensori.

Motori e sensori sono collegati al brick tramite cavi RJ12.



ROBOCUP JR ITALIA 2011 - Catania 14-16 aprile REPORT DI DOCUMENTAZIONE

Il display LCD 100x64 permette di visualizzare in maniera piuttosto chiara i dati necessari per capire la lettura dei sensori ad esempio durante le fasi di test, ed inoltre i 4 pulsanti posizionati subito sotto lo schermo permettono in modo veloce di selezionare le varie utilità di sistema tra cui anche selezionare ed avviare il programma creato da noi e caricato nel brick tramite un cavo USB a velocità (12 Mbit/s). E' inoltre presente un altoparlante che permette di riprodurre suoni ad una frequenza massima di 8kHz utili per capire quando e se si attivano le differenti parti del programma.

Il brick si collega al PC per mezzo di un connettore USB che consente il trasferimento del programma compilato per il processore ARM7.

Come già detto, esistono diversi ambienti di sviluppo per il Brick Lego, ma noi ci siamo indirizzati sul linguaggio NXC su indicazione dei professori, dato che molti di noi quest'anno hanno iniziato ad utilizzare il linguaggio C nella disciplina Informatica.

L'alimentazione a brick e sensori è fornita da un pacco batterie a ioni polimeri di litio da 7,2 Volt e 2100 mA. Durante la gara però potremmo utilizzare delle batterie alcaline da 1,5V ciascuna per un totale di 9V di alimentazione a brick e motori.



CAP. 6 - SENSORI

I sensori utilizzati, sono connessi al brick utilizzando un cavo con connettori a 6 pin che consente di gestire sensori che utilizzano comunicazioni sia di tipo analogica che digitale. L'interfaccia digitale consente di gestire comunicazioni seriali con bus I2C ed RS-485.

I sensori utilizzati da noi sono i seguenti:

Anteriori

- n. 2 Sensore colore Lego RGB
- Sensore di contatto

Posteriori

- Sensore Ultrasuoni Lego
- Sensore Hitechnic NXT Color
- Sensore Hitechnic accelerometro NXT

Laterale

- Sensore Hitechnic EOPD NXT

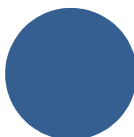
Sulla parte posteriore del terzo motore che si occupa del recupero vittima, troviamo fissato il sensore di CONTATTO LEGO®. Per gestire al meglio ogni possibilità di contatto è stata



FIG. 10 - I SENSORI LATERALI E POSTERIORI

realizzata attorno ad esso una particolare impalcatura larga quanto tutto il robot e molto sporgente.

Per quanto riguarda gli altri sensori, troviamo posizionati al di sotto della chela tre sensori ovvero il sensore ULTRASUONI LEGO®, il



ROBOCUP JR ITALIA 2011 - Catania 14-16 aprile REPORT DI DOCUMENTAZIONE

senso COLORE HiTechnic® ed il sensore ACCELEROMETRO HiTechnic®. I primi due sono stati posizionati in basso per favorire una migliore lettura di colori e distanze della vittima e della zona di salvataggio, mentre l'ultimo che serve a riconoscere la rampa, viene utilizzato anche nella terza stanza, per evitare false letture davanti alle pareti, dopo aver prelevato la vittima.

Sul lato sinistro troviamo a mezza altezza, il sensore EOPD HiTechnic® usato per la gestione



del movimento del robot rispetto alle pareti, mentre posizionati anteriormente e diretti verso il basso troviamo i due sensori RGB LEGO®, che ci permettono di ottenere una precisa ed accurata lettura della luce riflessa dalla superficie e quindi

far seguire, senza confondersi, la linea nera.

Per collegare tutti i sensori abbiamo utilizzato un componente elettronico, il MULTIPLEXER HiTechnic®, posizionato all'interno della struttura del robot, che ci permette di moltiplicare le porte a nostra disposizione per i sensori, consentendoci di collegare ben quattro sensori ad una sola porta di ingresso dei sensori.

Il Multiplexer è dotato di una alimentazione autonoma con batteria a transistor da 9V alloggiata in un contenitore in plastica abs nera posto anteriormente al robot.



Il Sensore colore Lego RGB



**FIG. 12 - SENSORE COLORE
LEGO RGB**

Il sensore colore Lego RGB, riconosce i colori ad una distanza di alcuni centimetri da esso, restituendo un numero che identifica il colore.

Può però essere configurato per emettere una sola delle componenti RGB e rilevare la percentuale di luce riflessa.

Abbiamo scelto questo tipo di configurazione per valutare anche le variazioni di colore bianco e nero, in modo da consentirci di sapere non solo se il sensore stava rilevando l'attraversamento della linea nera, ma anche se esso si trovava in pieno sopra la linea.

Dopo alcune prove, tra le luci Red, Green e Blue, abbiamo scelto di utilizzare la componente ROSSO, in quanto restituisce livelli maggiori di luce riflessa dalla superficie.

La configurazione dei sensori si effettua nel modo seguente:

```
// Sensore Colore NXT 2.0 Connesso alla porta 1
#define COLORE DESTRO  SENSOR_1
// Sensore Colore NXT 2.0 Connesso alla porta 2
#define COLORE SINISTRO SENSOR_2

// Imposta i sensori colore come sensori luce
SetSensorType(IN_1, SENSOR_TYPE_COLORRED);
SetSensorMode(IN_1, SENSOR_MODE_PERCENT);
ResetSensor(IN_1);
SetSensorType(IN_2, SENSOR_TYPE_COLORRED);
SetSensorMode(IN_2, SENSOR_MODE_PERCENT);
ResetSensor(IN_2);
```



ROBOCUP JR ITALIA 2011 - Catania 14-16 aprile
REPORT DI DOCUMENTAZIONE
mentre i valori vengono letti in tal modo:

```
int leggiDX, leggiSX;  
leggiDX= COLORE_DESTRO+OFFSETDX;  
leggiSX= COLORE_SINISTRO+OFFSETSX;
```

Al valore rilevato può essere aggiunto un offset poiché i sensori possono rilevare letture differenti quando riflettono sulla stessa superficie.

Il Sensore di contatto LEGO



**FIG. 13 - SENSORE DI
CONTATTO LEGO**

Il sensore di contatto è un sensore analogico (un semplice pulsante) interfacciato al bus I2C e restituisce al programma i livelli 0 se in assenza di contatto ed 1 se in presenza di contatto. La configurazione del sensore è la seguente:

```
// Sensore di contatto anteriore connesso alla porta 3  
#define CONTATTO Sensor(IN_3)  
  
// Imposta il sensore di contatto per rilevare gli ostacoli  
SetSensorType(IN_3, SENSOR_TYPE_TOUCH);  
SetSensorMode(IN_3, SENSOR_MODE_BOOL);  
ResetSensor(IN_3);
```

mentre il contatto viene rilevato così:

```
if (CONTATTO==true  
{
```



Il Sensore Ultrasuoni Lego



**FIG. 14 - SENSORE
ULTRASUONI LEGO**

L'obiettivo di questo sensore è di rilevare gli ostacoli. Il sensore è dotato di una capsula trasmittente ad ultrasuoni e una capsula ricevente. Il sensore ha un proprio microprocessore, di conseguenza, esso segnala la distanza in unità assolute piuttosto che su una scala relativa, a differenza di altri sensori. Il suo funzionamento è simile al sistema di un sonar. Trasmette un breve fascio di ultrasuoni a 40 kHz. Quindi misura il tempo necessario al fascio per arrivare ad un oggetto, riflettere e tornare indietro. I valori rilevati vanno da 5 a 255, quando il sensore comunica 255, il sensore potrebbe non rilevare alcun ostacolo o fornire una lettura errata per la vicinanza eccessiva di un ostacolo.

Per quanto riguarda la lettura, poiché il sensore ultrasuoni Lego è connesso al multiplexer, bisogna accedere al canale n. 1 ed acquisire il dato.

```
// Modalità per leggere in raw il sensore Ultrasuoni I2C digitale  
HTSMUXreadI2C(MUX, CH1, 0, 1, datiMUX);  
us=datiMUX[0];
```

Invece per quanto riguarda la configurazione, è sufficiente configurare il multiplexer.



Il Sensore Hitechnic NXT Color



**FIG. 15 - SENSORE
HITECHNIC NXT COLOR**

Il Sensore di colore HiTechnic a differenza del sensore colore Lego RGB, funziona utilizzando un singolo LED di colore bianco (emessa da un diodo) per illuminare l'obiettivo e analizzarne i componenti della luce riflessa dalla superficie dell'obiettivo e restituisce un numero identificativo del colore, è possibile inoltre leggere la percentuale di luce riflessa per ogni componente RGB.

Il Sensore Colore HiTechnic funziona meglio quando non è posizionato molto vicino alla superficie da leggere. Inoltre il sensore deve essere posizionato perpendicolarmente alla superficie in modo che l'angolazione dello stesso permetta di leggere correttamente i valori .

Per quanto riguarda la lettura, poiché il sensore HiTechnic NXT Color è connesso al multiplexer, bisogna accedere al canale n. 3, ed acquisire il dato.

```
// Modalità per leggere l'Hitechnic Colour Sensor I2C digitale  
HTSMUXreadI2C(MUX, CH3, 0, 4, datiMUX);  
col=datiMUX[0];
```

Le letture grezze dei canali rgb si effettuano leggendo datiMUX[1] per il canale RED, datiMUX[2] per il canale GREEN e datiMUX[3] per il canale BLUE.



ROBOCUP JR ITALIA 2011 - Catania 14-16 aprile
REPORT DI DOCUMENTAZIONE

Invece per quanto riguarda la configurazione, basta configurare il multiplexer.

Sensore Hitechnic accelerometro NXT



FIG. 16 - SENSORE
HITECHNIC
ACCELEROMETRO

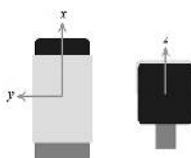


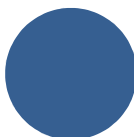
FIG. 17 - ASSI X, Y, E Z

Il sensore HiTechnic accelerometro si basa su tre assi x, y e z e ne misura l'accelerazione. L'accelerazione è misurata in un range di $-2g$ a $+2g$ con una scala approssimativa di 200 cicli per

g.

Il sensore si connette all'NXT tramite un cavo standard NXT che usa il protocollo di comunicazione digitale I2C. La misurazione dell'accelerazione per ogni asse è aggiornata approssimativamente 100 volte al secondo.

Il sensore di Accelerazione può inoltre essere usato per gestire l'inclinazione nei tre assi. Questo è possibile perché la gravità è percepita come accelerazione. Quando il sensore si trova in posizione perfettamente orizzontale gli assi x e y si avvicinano al valore 0, mentre l'asse z si avvicina al valore di 200, che rappresenta una g. Dal momento che la gravità è distribuita tra i tre



ROBOCUP JR ITALIA 2011 - Catania 14-16 aprile REPORT DI DOCUMENTAZIONE

vettori x, y e z , allora è possibile determinare l'inclinazione del sensore.

Se il sensore si inclina, il valore rilevato negli assi diminuisce e viceversa, restituendo valori compresi tra 0 e 255.

Per quanto riguarda la lettura, poiché il sensore HiTechnic Accelerometro è connesso al multiplexer, bisogna accedere al canale n. 4, ed acquisire il dato.

```
// Modalità per leggere l'HiTechnic Acceleration Sensor I2C digitale  
HTSMUXreadI2C(MUX, CH4, 0, 6, datiMUX);  
acc=datiMUX[0];
```

datiMUX[0] contiene la lettura rilevata sull'asse x ,
datiMUX[1] contiene la lettura rilevata sull'asse y
e datiMUX[2] contiene la lettura rilevata sull'asse z .

Sensore Hitechnic EOPD NXT



**FIG. 18 - SENSORE
HITECHNIC EOPD**

Il sensore EOPD dell' HiTechnic è un rilevatore elettro-ottico. Esso rileva la distanza degli oggetti utilizzando la luce modulata. Il sensore EOPD è simile al sensore di luce di LEGO rilevando la quantità di luce riflessa per mezzo del fotodiode rosso, solo che per eliminare le interferenze create dalla luce di fondo, modula la luce trasmessa. Tramite queste pulsazioni, la luminosità durante le pulsazioni appare poco



ROBOCUP JR ITALIA 2011 - Catania 14-16 aprile REPORT DI DOCUMENTAZIONE

luminosa. Il sensore misura la luminosità prima e durante le pulsazioni.

Il sensore EOPD dell' HiTechnic funziona in due modi di sensibilità, a corto raggio x1 ed a lungo raggio x4. Per quanto riguarda il lungo raggio x4 il sensore funziona correttamente fino ad un range di circa 15 cm. Il corto raggio invece x1 può essere invece utilizzato per riconoscere le parti bianche.

Il sensore EOPD campiona le informazioni 300 volte al secondo. Internamente, il sensore avvia tra i trecento e quattrocento campioni al secondo per assicurare che ci sia una lettura di almeno 300 campioni al secondo per l'NXT.

Ogni sensore EOPD funziona utilizzando ratei di campionamento leggermente diversi, in modo tale da minimizzare le interferenze tra sensori EOPD vicini tra loro.

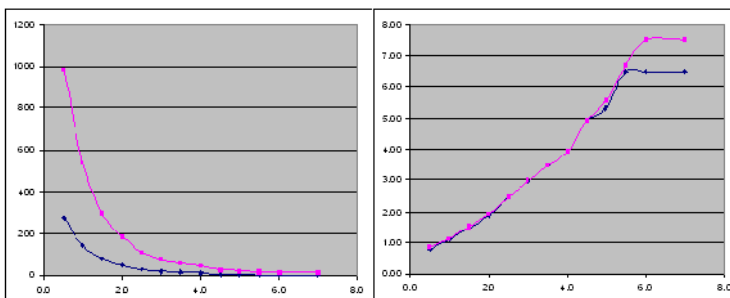


FIG. 19 - LETTURA GREZZA VS. LETTURA NORMALIZZATA

Nel primo grafico vediamo le letture del sensore (asse y) a confronto con il range catturato dal LED rosso espresso in pollici (asse x). La linea blu è l' X1 mentre la rosa l'X4. Questa è la modalità utilizzata da noi ed i valori decrescono a partire da 1023.



ROBOCUP JR ITALIA 2011 - Catania 14-16 aprile REPORT DI DOCUMENTAZIONE

Il secondo grafico invece la lettura normalizzata (linearizzata) in base all'operazione $\text{range} = \text{cal.factor} \sqrt{\text{raw.reading}}$. Il range è indicato sull'asse x mentre la y mostra i risultati dell'operazione.

Per quanto riguarda la lettura, poiché il sensore HiTechnic EOPD NXT è connesso al multiplexer, bisogna accedere al canale n. 2, ed acquisire il dato.

```
// utilizza il sensore EOPD per mantenerti alla stessa distanza dal muro  
eopd= 1024 - HTSMUXreadAD (MUX, CH2);
```

Il Multiplexer HiTechnic



FIG. 20 - MULTIPLEXER HITECHNIC

Il Multiplexer HiTechnic per sensori NXT, permette di collegare un'ampia gamma di sensori digitali ed analogici alle varie porte NXT, e quindi di estendere il numero di porte presenti nel brick.

Il multiplexer è connesso all'NXT tramite un cavo standard NXT che usa il protocollo di comunicazione digitale I2C. Esso è alimentato con una batteria da 9 V che consente di alimentare i sensori connessi alle porte del MUX.

Le porte di ingresso sono quattro, identificate rispettivamente CH1, CH2, CH3 e CH4.



ROBOCUP JR ITALIA 2011 - Catania 14-16 aprile REPORT DI DOCUMENTAZIONE

Purtroppo data la caratteristica dei multiplexer, è possibile leggere un solo sensore per volta. Questo ne limita un po' le modalità di utilizzo, ma la possibilità di ampliare la quantità di sensori collegati al brick ci ha convinti.

Di seguito la configurazione software del multiplexer. Per la lettura dei diversi canali, si rimanda ai singoli sensori collegati ai diversi canali.

```
// Multiplexer Hitechnic connesso alla porta 4
#define MUX                S4

// Dati per la gestione del MUX Hitechnic
#define HTSMUX_CMD_HALT 0x00
#define HTSMUX_CMD_AUTODETECT 0x01
#define HTSMUX_CMD_RUN 0x02
#define CH1 0 // ULTRASUONI POSTERIORE
#define CH2 1 // EOPD LATERALE SX
#define CH3 2 // HITECHNIC COLOR POSTERIORE
#define CH4 3 // ACCELEROMETRO/TILT SENSOR

// Attiva il MUX
SetSensorLowspeed(MUX);
Wait(200);
// Invia il comando halt al MUX
HTSMUXsendCmd(MUX, HTSMUX_CMD_HALT);
    Wait(100);
// Effettua la scansione dei sensori
HTSMUXsendCmd(MUX, HTSMUX_CMD_AUTODETECT);
    Wait(100);
// Attiva il MUX ponendolo in RUNNING
HTSMUXsendCmd(MUX, HTSMUX_CMD_RUN);
    Wait(200);
```

Il multiplexer è un dispositivo a bassa velocità e deve essere configurato prima di essere utilizzato.



CAP. 7 - ATTUATORI



FIG. 21 - MOTORE LEGO 9842

SpikeBot si avvale di motori elettrici interattivi LEGO 9842 ed il brick ne consente il collegamento contemporaneo di tre. Questi motori, attivati ad una certa velocità possono ruotare in tutti e due i sensi di rotazione, essi permettono inoltre di misurare l'esatta rotazione dei motori, comunicando direttamente con il brick.

Due di questi sono stati usati per far muovere il robot trasmettendo il moto ad un sistema di cingoli mosso da tre ruote di cui una è motrice. Uno invece è stato utilizzato per catturare la vittima.

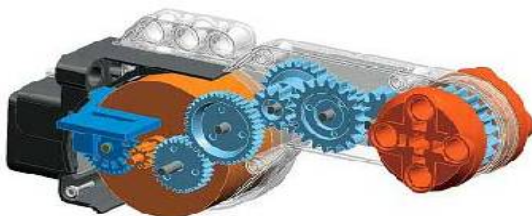


FIG. 22 - VISTA INTERNA DEL MOTORE LEGO

Il cavo NXT che collega il motore è composto da sei fili.



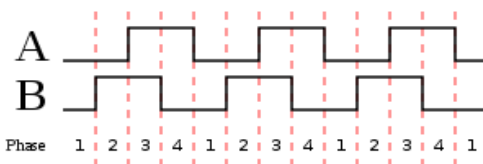
ROBOCUP JR ITALIA 2011 - Catania 14-16 aprile REPORT DI DOCUMENTAZIONE

I fili bianco e nero forniscono energia al motore. Con batterie alcaline la differenza di potenziale tra i due fili è di 9V mentre con batterie ricaricabili è di 7,2V.

Se il filo bianco è positivo ed il nero è negativo, il motore gira in una direzione. Se si inverte la polarità, il motore girerà al contrario. Il filo rosso è connesso alla base (GND). Il filo verde è connesso all'alimentazione +4,3 dell'NXT. I fili blu e giallo sono connessi all'encoder incrementale che rileva esattamente i giri del motore.

Colore Nome

| | |
|--------|-----------|
| Bianco | Motore 1 |
| Nero | Motore 2 |
| Rosso | GND |
| Verde | 4.3 Volts |
| Giallo | Tach00 |
| Blu | Tach01 |



**FIG. 23 - IMPULSI AD ONDA QUADRA RILEVATI
DAI FILI BLU E GIALLO**

I cavi blu e giallo restituiscono impulsi ad onda quadra sfasati di 90° tra loro. Se il fronte di salita su TACH00 precede il fronte di salita su TACH01 di 90° allora il motore sta ruotando in avanti. Se invece accade il contrario, il motore sta ruotando indietro.

Ogni forma d'onda quadra corrisponde a 2° di rotazione. La fig. 23 mostra che se A è l'impulso rilevato dal giallo e B è l'impulso rilevato dal blu, si rileva che il motore sta andando indietro poichè il fronte di salita di TACH01 precede quello di TACH00.

Misurando la frequenza dell'onda quadra, si può calcolare la velocità di rotazione, sapendo che un



ROBOCUP JR ITALIA 2011 - Catania 14-16 aprile REPORT DI DOCUMENTAZIONE

periodo (1Hz) corrisponde a 2° di rotazione del motore.

La risoluzione minima possibile è $\frac{1}{4}$ di periodo, ovvero $\frac{1}{2}^\circ$.

La velocità del motore viene controllata da un modulatore ad impulsi di codice, PWM, che fornisce al motore un segnale ad onda quadra con duty cycle variabile. Questo permette al motore di attivare o disattivare il motore velocemente. Più a lungo il motore è attivo e maggiore sarà la coppia che il motore potrà generare così il robot andrà più velocemente.

Efficienza dei motori in base alla tensione applicata.

Abbiamo effettuato una ricerca per controllare se la variazione di tensione applicata ai motori influisse durante l'utilizzo del motore a vuoto o sotto carico.

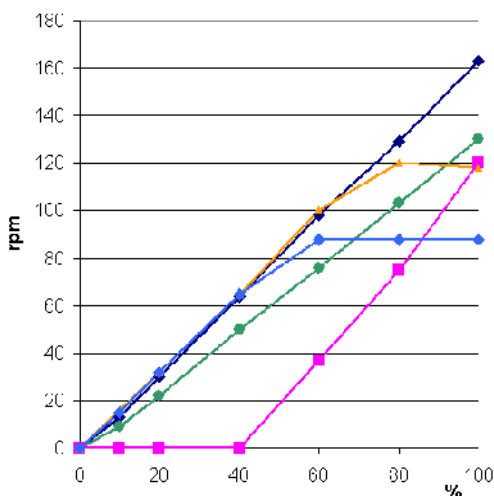


FIG. 24 - VELOCITÀ DI ROTAZIONE VS. LIVELLO DI POTENZA DEL MOTORE



ROBOCUP JR ITALIA 2011 - Catania 14-16 aprile REPORT DI DOCUMENTAZIONE

La figura 24 mette in relazione la velocità di rotazione del motore NXT (rpm) con il livello di potenza del motore (duty-cycle applicato). Le curve rappresentano un:

◆ Motore che gira a vuoto ed è alimentato a 9V. Si ottiene una relazione lineare tra livello di potenza e velocità del motore.

◆ Motore che gira a vuoto ma stavolta è alimentato con una tensione di 7.2V. La relazione è sempre lineare, ma la velocità di rotazione è inferiore in proporzione.

◆ Motore con applicato un carico di 11.5 Newton x cm, senza controllo di potenza ed alimentato con una tensione di 9V. Con livelli di potenza inferiori al 40% il motore si blocca (regione orizzontale della curva). Se si applica un livello di potenza superiore, la velocità di rotazione aumenta in modo proporzionale.

◆ Motore con applicato un carico di 11.5 Newton x cm, ed alimentato con una tensione di 9V. Questa curva mostra l'efficienza del controllo di potenza: la velocità di rotazione è la stessa di un motore senza carico fino al 70% di potenza. Per livelli superiori la curva si appiattisce ed il motore ruota al massimo.

◆ Motore con applicato un carico di 11.5 Newton x cm, ed alimentato con una tensione di 7.2V. Fino a livelli di potenza del 50% la velocità di rotazione è la stessa di un motore senza carico alimentato a 9V. L'appiattimento della curva è però evidente a valori superiori.



ROBOCUP JR ITALIA 2011 - Catania 14-16 aprile

REPORT DI DOCUMENTAZIONE

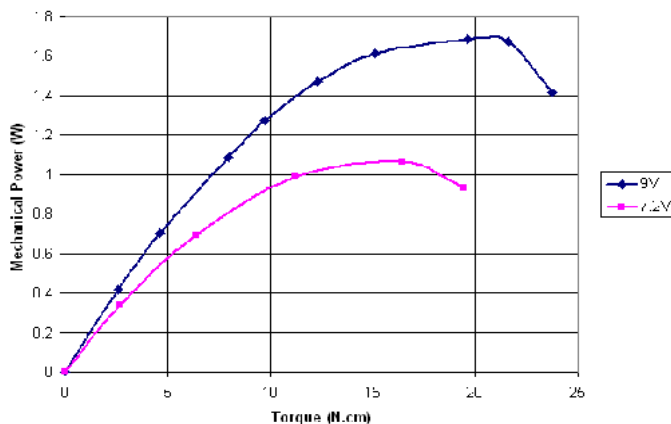


FIG. 25 - POTENZA DEL MOTORE VS. COPPIA

La figura 25 mostra la curva caratteristica dei motori Lego NXT con un carico applicato da 15 Newton x cm. La curva blu è relativa ad un motore alimentato con batterie alcaline a 9V, mentre la curva magenta è stata ottenuta con un motore alimentato a 7.2V. Entrambi i motori avevano il 100% di livello di potenza applicata. Confrontando le due curve risulta evidente che la potenza effettiva del motore è migliore se esso è alimentato con una tensione maggiore.

I motori possono essere attivati in diversi modi:

```
// Attiva i motori in avanti eventualmente imposta i motori a velocità differenti se serve
OnFwdSync(MOTORI,potenza,0);
```

Con OnFwdSync i motori si avviano ruotando avanti in sincronia tra loro. OnRevSync permette invece la rotazione indietro.



ROBOCUP JR ITALIA 2011 - Catania 14-16 aprile
REPORT DI DOCUMENTAZIONE

Lo spegnimento dei motori avviene con il comando Off(MOTORI);

Il comando RotateMotor consente invece la rotazione del motore e non restituisce il controllo al programma fino a quando tale rotazione non è avvenuta.

```
// Indietreggia con i motori un po'  
RotateMotor(MOTORI, -60, 300);
```

La rotazione in avanti o indietro è consentita utilizzando come secondo parametro della funzione un valore di potenza positivo o negativo. Il terzo parametro infine indica i gradi di rotazione che il motore deve eseguire.



CAP. 8 - AMBIENTE DI SVILUPPO

L'ambiente di sviluppo utilizzato è il BricxCC versione 3.3.8.8.

Bricx Command Center (BricxCC) è un programma comunemente noto come un ambiente di sviluppo integrato (IDE) per programmare l'NXT, programmabile con diversi linguaggi NQC, NXC, Pascal e Java utilizzando anche i firmware alternativi BrickOS, pbForth e Lejos.

Il linguaggio prescelto da noi è l'NXC versione 1.2.1r4, un linguaggio molto simile nei comandi ed in sintassi al Linguaggio C.

Abbiamo scelto quest'ambiente perché rispecchia la filosofia da noi adottata di utilizzare software e linguaggi open source o comunque freeware, inoltre così facendo evitiamo l'acquisto di onerose licenze che graverebbero sul bilancio scolastico.

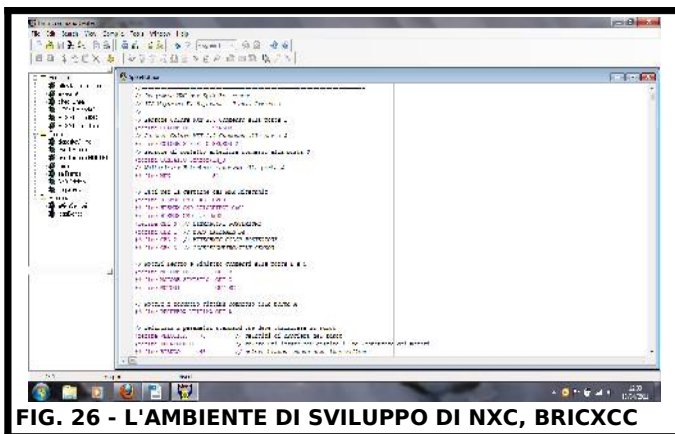


FIG. 26 - L'AMBIENTE DI SVILUPPO DI NXC, BRICXCC



CAP. 9 - IL PROGRAMMA SOFTWARE



FIG. 27 - TUTTI CONCENTRATI SULLA SCRITTURA DEL CODICE DEL PICCOLO SPIKE.....

Il programma è diviso in una serie di task e di procedure che si occupano di gestire i vari sensori ed i motori.

```
// Dichiarazione dei task
task evitaostacoloMOCTRL();
task evitaostacolo();
task seguiLinea();
task SaK Vittima();
task depositaVittima();
task salitaRampa();

// Dichiarazione delle subroutine
safecall int HTSMUXsendCmd(byte smux, byte cmd);
safecall int HTSMUXreadAD (byte smux, byte chan);
safecall int HTSMUXreadI2C (byte smux, byte chan, byte offset, byte length, byte &data[]);
safecall void leggiSensori();
safecall void attivaSensori();
safecall int checkLinea();
safecall int avvicinati(int potenza, int distanza, int precedente);
safecall int attivaMotoriControllo(int tipo, const byte &port1, const int &power1, const long &angolo1,
```

Le prime due stanze vengono gestite da tre task che si occupano rispettivamente di: seguire la linea nera, evitare eventuali ostacoli lungo il



ROBOCUP JR ITALIA 2011 - Catania 14-16 aprile REPORT DI DOCUMENTAZIONE

percorso e individuare che si ci trova in presenza della salita.

Il task che si occupa di seguire la linea nera riceve informazioni dai due sensori colore posti nella parte anteriore del robot. Il task richiama una funzione checklinea().

```
// Sottoprogramma che restituisce un valore correlato alla lettura
// dei due sensori di colore
safecall int checkLinea()
{
    int leggiDX, leggiSX;

    leggiDX= COLORE_DESTRO+OFFSETDX;
    leggiSX= COLORE_SINISTRO+OFFSETSX;

    if((leggiDX<(BIANCO-10)) && (leggiSX<(BIANCO-10))) // -7
    {
        // tutti e due i sensori sono sulla linea nera
        // però se uno dei due legge valori più piccoli, rileva dx o sx
        if (leggiDX<leggiSX) return 1;
        if (leggiSX<leggiDX) return 2;
        return -1;
    }
    if (leggiDX<(BIANCO-10)) // -7
    {
        // il sensore destro ha rilevato la linea nera
        return 1;
    }
    if (leggiSX<(BIANCO-10)) // -7
    {
        // il sensore sinistro ha rilevato la linea nera
        return 2;
    }
    // Linea nera non rilevata
    return 0;
}
```

In essa se entrambi i sensori rilevano di essere sulla linea nera restituendo valori equivalenti i motori vengono fermati un istante e ripartono con potenza differente in modo tale da sbilanciare un po' il robot. Se i due sensori rilevano di essere entrambi sulla linea nera ma



ROBOCUP JR ITALIA 2011 - Catania 14-16 aprile REPORT DI DOCUMENTAZIONE

con valori di riflessione differente il motore con il valore di riflessione più basso viene rallentato in maniera esponenziale e se necessario il motore viene ruotato al contrario, fino a quando entrambi i sensori rilevano di trovarsi sul bianco; in questo caso entrambi i motori procedono in maniera sincronizzata all' infinito a meno che uno dei due sensori non rilevi una variazione di riflessione, in questo caso si ripete quanto detto precedentemente.



FIG. 28 - ENRICO IL CAPITANO PROGRAMMATTORE DURANTE UNA FASE DI TEST SUL CAMPO

Nel caso in cui durante il tragitto il sensore contatto posto anteriormente rilevi un contatto con un ostacolo, viene attivato il task

evitaOstacolo().

Esso agisce sui motori facendo

tornare un po' indietro il robot e dopo averlo fatto girare di 45° procede in avanti finchè il sensore eopd non rileva d' aver superato l' ostacolo in questo caso quindi viene ruotato nuovamente il robot di 45° verso sinistra e si procede in avanti fino a quando il sensore EOPD non rileva la fine dell'ostacolo, a questo punto il robot ruota nuovamente di 45° verso sinistra, avanza un po' e poi ruota nuovamente di 45° a destra. Durante tutta la fase di sorpasso i due sensori di luminosità sono attivati per rilevare e seguire la linea nera, ma solo dopo che il sensore sinistro



ROBOCUP JR ITALIA 2011 - Catania 14-16 aprile REPORT DI DOCUMENTAZIONE

non supera la linea nera e fornisce il consenso al programma consentendo di seguire la linea rilevata anche dal sensore destro.

```
// Codice per la salita della rampa da attivare solo
//se non si sta superando un ostacolo
if (ostacoloAttivo==false && ostacoloAttivoSX==false)
{
  for (int i=1; i<=5; i++)
  {
    // Leggi l'inclinazione con il sensore accelerometri
    HTSMUXreadI2C(MUX, CH4, 0, 1, datiMUX);
    tilt=tilt+datiMUX[0];
    Wait(50);
  }
  mediaTilt=tilt/5;
  if (mediaTilt>225 && mediaTilt<247)
  {
    // Codice per la salita da eseguire 1 sola volta
    salita=true;
    // sospendi l'esecuzione di seguiLinea
    Off(MOTORI);
    PlayTone(440, 500);
    Wait(50);
    PlayTone(440, 500);
    // fai partire saliRampa
    Wait(2000);
  }
  mediaTilt=0;
  tilt=0;
}
```

Non appena il sensore accelerometro rileva una variazione di inclinazione di una certa durata, viene avviato il task saliRampa(). Esso si occupa di rallentare i motori aumentando quindi l'attrito tra cingoli in gomma e salita, e seguire l'eventuale presenza della linea nera durante la salita, combinando il funzionamento dei due sensori di luminosità anteriori con il sensore



ROBOCUP JR ITALIA 2011 - Catania 14-16 aprile REPORT DI DOCUMENTAZIONE

EOPD, mantenendo così una certa distanza di sicurezza dalla parete. I valori rilevati dal sensore EOPD vengono utilizzati per rallentare o velocizzare i motori se il robot non è correttamente allineato alla parete laterale.

```
// Attiva i motori in avanti eventualmente imposta i motori a velocità differenti se serve
OnFwdSync(MOTORI,potenza,0);
do
+
  // utilizza il sensore EOPD per mantenerti alla stessa distanza dal muro.
  eopd= 1924 - HISMUXreadAD (MUX, CH2);
  if (eopd>=1 && eopd<=3)
  {
    motoreDX=potenza - powerplus +*((12-eopd)*2));
    motoreSX=potenza + powerplus - 10; // -15
  }
  if (eopd>=4 && eopd<=14)
  +
    motoreDX=potenza;
    motoreSX=potenza;
  }
  if (eopd>=14)
  +
    if (eopd>=80) eopd=80;
    motoreDX=potenza - powerplus - 10; // -15
    motoreSX=potenza + powerplus +4*((eopd-13)*4)/11));
  }

// Controlla la lettura dei due sensori colore Lego
leggiLinea=checkLinea();
switch (leggiLinea)
+
  case 0 :
    // Nessuna linea rilevata, continua ad avanzare con i valori dati dall'EOPD
    if (motorePP!=0) motorePP=0;
    break;
  case 1 :
    // Il sensore destro ha rilevato la linea nera, accelera il motore SX in maniera incrementale
    motorePP=motorePP+0;
```

Non appena il sensore accelerometro ritorna a valori normali che indicano la fine della salita o se esso dovesse fallire nel rilevamento, il robot andrebbe ad urtare anteriormente sulla parete che delimita la piattaforma superiore della rampa. Si attiverebbe così il sensore contatto il robot in tal caso indietreggerebbe un po' ed effettuerebbe una rotazione di 90° a destra in modo da posizionarsi di fronte all'entrata della terza stanza e procedendo indietro fino a quando i due sensori colore posti anteriormente non rileverebbero la striscia di alluminio che indica che ci troviamo nella terza stanza.



ROBOCUP JR ITALIA 2011 - Catania 14-16 aprile REPORT DI DOCUMENTAZIONE

A questo punto il task saliRampa() viene disattivato, mentre viene contemporaneamente attivato il task SaRVittima(). Una volta entrati si procede in avanti leggendo i valori restituiti dal sensore colore per rilevare se la lattina si trova di fronte a noi. Si procede fino al centro della stanza ed il robot compie una rotazione di 360° alternativamente a destra ed a sinistra, leggendo i valori restituiti dal sensore ultrasuoni, quando viene rilevata una variazione di distanza individuando perciò un ostacolo si avvicina ad esso e controlla che sia la lattina, o un ostacolo o la parete previa verifica positiva con il sensore colore posteriore HiTechnic che restituisce anch'esso valori relativi alla luce riflessa dalla superficie.

```
// gira di 360° a destra o a sinistra controllando anche con il sensore USMax
if (DXLS2==1)
{
  controlla= attivaMotoriControllo(7, MOTORE_DESTRO, 0-(VELSTANZA), 360*MOTORE_SINISTRO, VELSTANZA, 360*4);
}
else
{
  controlla= attivaMotoriControllo(7, MOTORE_DESTRO, VELSTANZA, 360*MOTORE_SINISTRO, 0-(VELSTANZA), 360*4);
}

switch (controlla)
{
case 1:
  //sono troppo a contatto, indietro
  controlla = attivaMotoriControllo(6, MOTORE_DESTRO, VELSTANZA, 220, MOTORE_SINISTRO, VELSTANZA, 220);
  controlla= attivaMotoriControllo(7, MOTORE_DESTRO, 0-VELSTANZA, 20,MOTORE_SINISTRO, 0-VELSTANZA, 20);
  break;
case 2:
  // se trovi la vittima, esci dal ciclo
  break;
break;
}
```

Tali valori sono legati anche al consenso che deve essere fornito dal sensore ultrasuoni Lego che rileva la distanza. Se essa non è adeguata il robot si avvicina e ricomincia con la verifica del livello di luminosità riflessa, se si trova invece troppo vicino, si allontana e compie la stessa operazione descritta.



ROBOCUP JR ITALIA 2011 - Catania 14-16 aprile REPORT DI DOCUMENTAZIONE

Se la verifica ha esito positivo, si avvia la procedura per catturare la vittima altrimenti si procede come detto precedentemente sino a quando non viene individuata la vittima.

```
// Abbassa le chele per catturare la vittima
RotateMotor(RECUPERA_VITTIMA, -70, (360*5)-60);
// Indietreggia con i motori un po'
RotateMotor(MOTORI, -60, 300);

// Afferrala per trascinarla ( 1 FASE )
RotateMotor(RECUPERA_VITTIMA, 75, (360*2)+30);
// Avanza con i motori trascinando un pò la vittima
RotateMotor(MOTORI, 60, 360);
RotateMotor(MOTORI, -60, 60);
// Afferrala ed alzala al massimo ( 2 FASE )
RotateMotor(RECUPERA_VITTIMA, 85, (360*3)-180);

// ADESSO CERCA LA ZONA NERA
start depositaVittima;
```

Una volta afferrata la nostra vittima il task SaRVittima() attiva il task depositaVittima() che allinea il robot ad una delle pareti facendo ruotare il robot di 360° su se stesso e memorizzando il valore più vicino rilevato con il sensore ultrasuoni, e successivamente ruotando per ritornare in quella posizione. SpikeBot avanza fino a quando il sensore contatto non rileva l'urto con una eventuale parete o ostacolo, quindi indietreggia un po' e viene effettuato un giro di 180° per individuare attraverso il sensore colore se l'oggetto con il quale siamo andati a sbattere è la nostra base una parete o un ostacolo. Tale fase ci ha dato non pochi grattacapi poiché il sensore colore HiTechnic restituisce valori molto simili sia quando siamo davanti una parete nera,



ROBOCUP JR ITALIA 2011 - Catania 14-16 aprile REPORT DI DOCUMENTAZIONE

che quando è troppo distante dalla parete da rilevare.

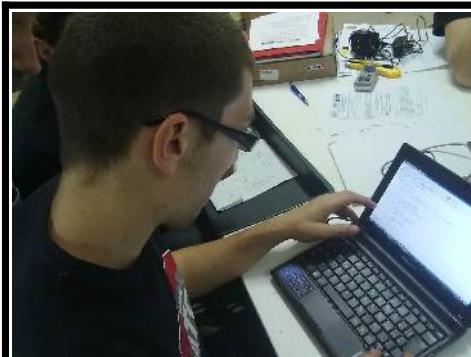


FIG. 29 - AL LAVORO SUL CODICE

Siamo riusciti a risolvere il problema effettuando la lettura del sensore colore e condizionandola al consenso della distanza rilevata dal sensore ultrasuoni Lego

e della corretta inclinazione del robot rilevata dal sensore accelerometro HiTechnic.

```
// leggi il colore se la distanza è <=0, ed accelerometro compreso tra 255 ed 1, è nero?
// esegui la lettura rgb sensore III color 1 volta sola
wait(100);
HTSMUX:=rotI2C(MUX, CH4, 0, 6, HaliMUX);
if (datiMUX[0]>=255 || datiMUX[3]<=1) noBuono=0; else noBuono=1;
ITSMUX:=readI2C(MUX, CH3, 0, 4, datiMUX);
if (datiMUX[0]==9 && datiMUX[1]<=NERO && datiMUX[2]<=NERO && datiMUX[3]<=NERO && guai<=8 && noBuono=1)
  controlla=7;
else
  attivaMotoriControllo(0, MOTORE_DESTRO, VEL3STANZA, 90, MOTORE_SINISTRO, VEL3STANZA, 90);
```

Altrimenti si effettua un ulteriore giro di 45° e si procede mantenendo una certa distanza dalla parete sino alla prossima attivazione del sensore di contatto, una volta individuata la zona di deposito della vittima viene abbassata la chela e rilasciata la lattina, per evitare che essa sia posta in posizione obliqua durante la cattura ed quindi anche durante il rilascio, SpikeBot indietreggia velocemente verso la zona di cattura e dopo aver ancor di più aperto le chele si allontana da essa per evitare di rimanere a contatto con la vittima.



CAP. 10 - SORGENTE DI ALIMENTAZIONE

Il robot può essere alimentato con batterie di diverso tipo. Abbiamo provato i diversi tipi di pacchi batterie, n. 6 batterie alcaline da 1,5V ciascuna, per un totale di 9 Volt, n. 6 ricaricabili al Ni-Mh da 1,2V 2800 mAh ciascuna per un totale di 7,2V e 2800 mAh ed i pacchi batterie ricaricabili Lego Li-Ion da 7,2V e 2100mAh e Ni-MH sempre da 7,2V e 1400 mAh.

Invece il multiplexer HiTechnic necessita di una sorgente di alimentazione autonoma fornita da una batteria a transistor da 9Volt alloggiata in un contenitore nero posto anteriormente al robot, vicino al Brick Lego.

Abbiamo effettuato quindi una ricerca per approfondire il differente comportamento dei motori quando le batterie tendono a scaricarsi fornendo una differenza di potenziale inferiore o quando essi sono alimentati con livelli di tensione diversi.

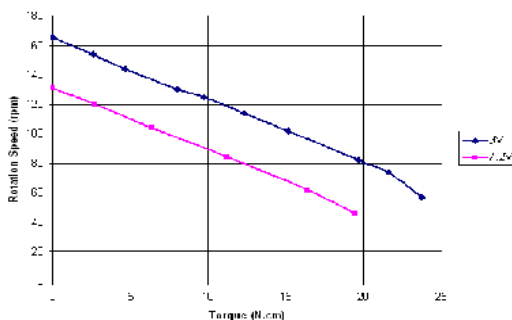


FIG. 30 - VELOCITÀ DI ROTAZIONE VS. COPPIA



ROBOCUP JR ITALIA 2011 - Catania 14-16 aprile REPORT DI DOCUMENTAZIONE

In fig. 30, con alimentazioni di tipo diverso, si nota la differente velocità di rotazione ed il miglioramento della coppia.

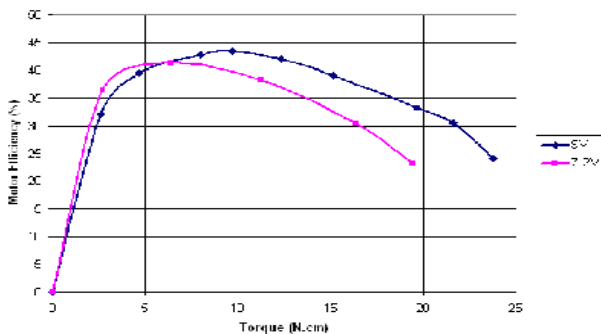


FIG. 31 - EFFICIENZA DEL MOTORE VS. COPPIA

Come possiamo notare nella figura 31, a parità di coppia anche l'efficienza del motore ne risente positivamente.

Il comportamento del robot è corretto con batterie cariche al 100%, ma i motori come visto si comportano meglio con le batterie alcaline a 9V.

Quindi durante la gara propenderemo per l'uso delle batterie alcaline rispetto a quelle ricaricabili.



APPENDICE 1 - SCHEMA DI MONTAGGIO IN 3D

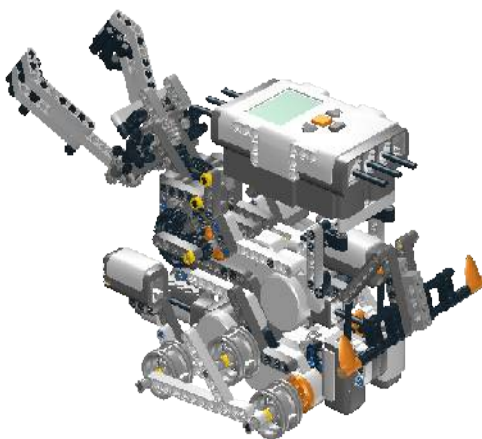


FIG. 32 - SPIKEBOT

Le pagine che seguono illustrano il montaggio di SpikeBot nei minimi dettagli.

Esso è stato realizzato con il software Lego Engineering Digital Design LDD, che consente la realizzazione in 3D dei robot LEGO. Dopo averlo disegnato lo abbiamo riportato in PDF con non poche difficoltà. Dovendo ridurre le pagine del report abbiamo condensato gli schemi applicandone più di uno per pagina.

Lego LDD non consente di inserire sensori o il multiplexer non LEGO, e quindi abbiamo sostituito i sensori non LEGO con altri LEGO aventi identica dimensione. Inoltre non sono presenti i cingoli in gomma morbida e pertanto essi non sono stati applicati.



INDICE

| | |
|--|--------------|
| CAP. 1 - DATI GENERALI | P. 6 |
| CAP. 2 - DATI DI CONTESTO E MOTIVAZIONE | P. 8 |
| CAP. 3 - NOME E STRUTTURA DEL ROBOT | P. 10 |
| CAP. 4 - MECCANICA | P. 11 |
| CAP. 5 - UNITÀ DI CONTROLLO | P. 14 |
| CAP. 6 - SENSORI | P. 16 |
| CAP. 7 - ATTUATORI | P. 27 |
| CAP. 8 - AMBIENTE DI SVILUPPO | P. 33 |
| CAP. 9 - IL PROGRAMMA SOFTWARE | P. 34 |
| CAP. 10 - SORGENTE DI ALIMENTAZIONE | P. 42 |
| APPENDICE 1 - SCHEMA MONTAGGIO 3D | P. 44 |

