



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

# **ROBOCUP JR ITALIA 2013**

**PESCARA  
17-20 aprile**

**REPORT DI DOCUMENTAZIONE  
Della squadra  
PERSEO**

# **IIS Istituto “Italo Calvino”**

**liceo scientifico (*opzione scienze applicate*), indirizzo elettronico ed elettrotecnico e indirizzo informatico e telecomunicazioni**

via Borzoli 21 16153 Genova

tel. 010 6508778/6504672

fax 010 6504241

e-mail: [istituto@calvino.ge.it](mailto:istituto@calvino.ge.it)

ROBOCUP JR ITALIA 2013 – PESCARA 17-20 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 Marciànò. 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, 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 prevede 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.

**COMITATO LOCALE** - Cura l'edizione annuale della manifestazione, ed è formato a cura dell'Istituto fondatore o associato a cui il Comitato di Gestione ha assegnato la cura dell'evento.

**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.

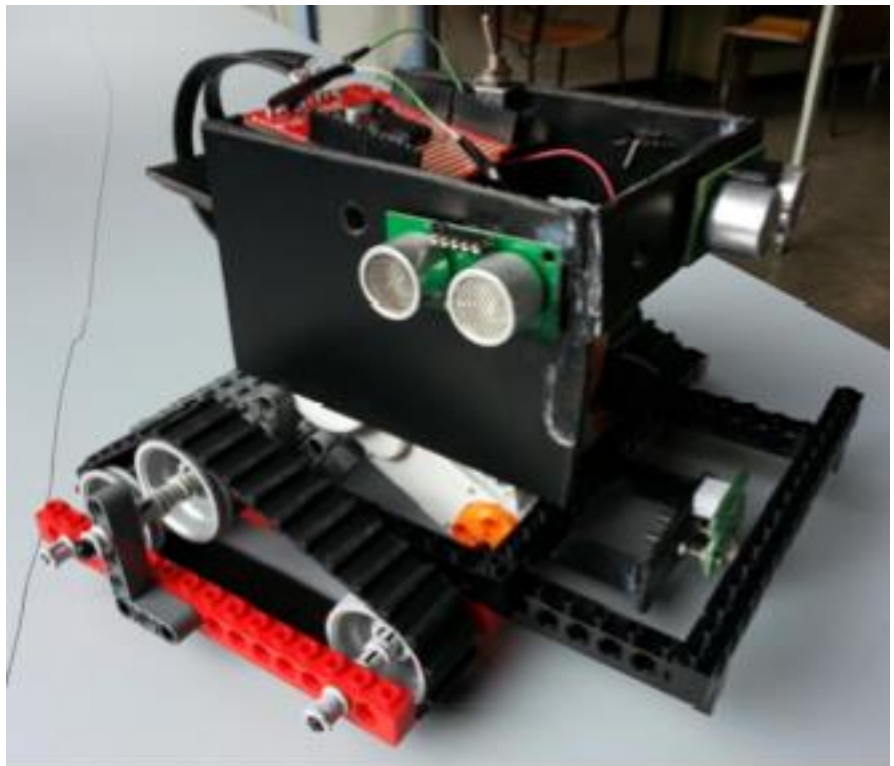
ROBOCUP JR ITALIA 2013 – PESCARA 17-20 aprile  
REPORT DI DOCUMENTAZIONE

**Locarno Marco**  
**Fenu Simone**  
**Prof. Drei Gloria**

# **PERSEO**

**PARTECIPA ALLA GARA DI:**  
**Rescue B**

**Istituto Tecnico Italo  
Calvino**  
**Classe: 4° Elettronica e  
Telecomunicazioni**



## CAP. 1 - DATI GENERALI

Istituto di provenienza:

Istituto Tecnico Italo Calvino di Genova

Componenti della squadra:

Locarno Marco, Fenu Simone

Docenti responsabili:

Drei Gloria, Ventura Pietro Franco

Durata del progetto:

Novembre 2012 - Aprile 2013

Categoria:

Rescue B





## CAP. 2 - DATI DI CONTESTO E MOTIVAZIONE

L'Istituto Italo Calvino, dal 2009, propone agli studenti del triennio il progetto di robotica, che si pone come obiettivi finali, tramite il lavoro di gruppo e l'impegno attivo, l'apprendimento di diverse applicazioni dell'elettronica e la partecipazione ad una gara nazionale.

Nello specifico, nell'a.s. 2009/10 gli studenti hanno partecipato e vinto la gara nazionale Romecup nella categoria Rescue A, con conseguente partecipazione alla gara mondiale tenutasi a Singapore.

L'anno seguente il progetto è continuato e si è concluso nel mese di Maggio con una sfida con l'Istituto Gaslini di Genova, vinta dai nostri studenti.

Nell'a. s. 2011/12 tre gruppi di studenti hanno partecipato alla Romecup nelle categorie Explorer e Rescue A, raggiungendo il secondo posto a livello nazionale in quest'ultima.

Quest'anno l'attività di robotica è stata estesa anche agli studenti del biennio, grazie alla gara First Lego League, e differenziatasi per quanto riguarda il triennio nelle gare di Rescue A (con partecipazione alla Romecup 2013) e Rescue B.

La proposta di partecipazione alla Robocup Jr 2013 di Pescara è nata dagli studenti stessi (Locarno Marco e Fenu Simone) ed appoggiata con forte spirito d'iniziativa dalla professoressa Drei Gloria, attiva sin dall'inizio

dell'attività nel 2009 e notevolmente motivata a diffondere una cultura nell'ambito della robotica a tutti gli studenti dell'istituto.

Il professore Ventura Pietro Franco ha anch'esso contribuito allo svolgimento del progetto, inserendosi questo stesso anno e fornendo utili consigli tecnici agli studenti.

La scuola ha sostenuto il progetto permettendo lo svolgimento delle attività nei laboratori utilizzati nelle ore pomeridiane; nonostante la mancanza di fondi, gli studenti, motivati dallo spirito competitivo e dal desiderio di cimentarsi in queste esperienze formative, hanno comunque finanziato i progetti tramite un contributo economico da parte delle famiglie. La scuola ha inoltre fornito strumentazioni e componenti elettronici necessari alla costruzione del robot.

La partecipazione per il primo anno alla categoria Rescue B risulta molto stimolante anche dal punto di vista didattico, e potrà rappresentare un ottimo spunto di analisi e apprendimento nelle ore di lezione in classe; allo stesso modo, l'utilizzo per la prima volta dell'hardware Arduino sarà sicuramente utile all'introduzione di questa piattaforma nelle materie tecniche e anzi ha già stimolato molti studenti a partecipare al corso offerto da Scuola di Robotica di Genova.

## CAP. 3 – NOME E STRUTTURA DEL ROBOT

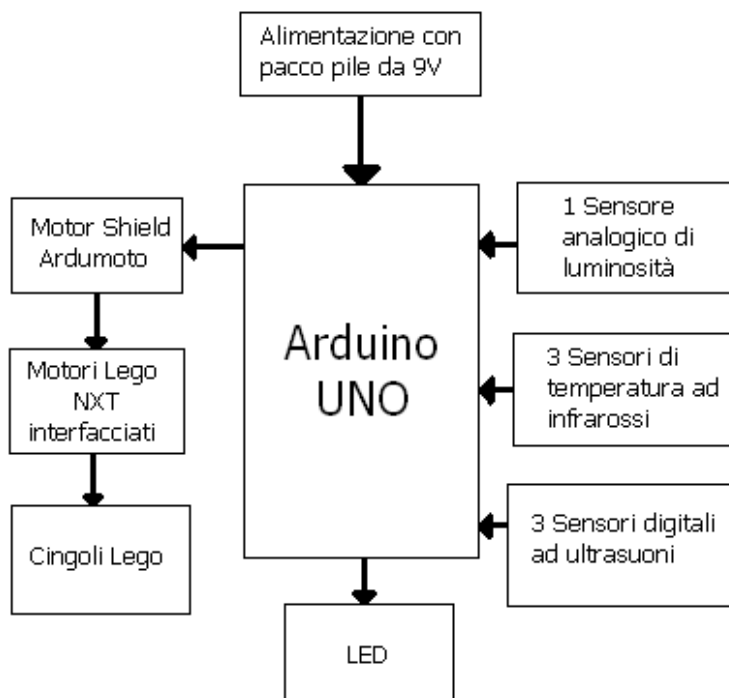
Perseo è un robot che prende il nome dal famoso eroe della mitologia greca; è stato scelto questo nome perché ci piace pensare che non esistono sfide impossibili (come dimostra il personaggio stesso con l'uccisione di Medusa), ma solo esperienze ed occasioni da cogliere al momento giusto.

Il nome è stato utilizzato anche nel 2012 per il robot partecipante alla categoria Rescue A della Romecup, grazie al quale siamo arrivati secondi; si è voluto mantenere anche quest'anno perché ancora di più iniziare un progetto del tutto innovativo e particolarmente impegnativo rappresenta una vera sfida.

Abbiamo progettato Perseo basandoci sulla scheda Arduino Uno e costruito la struttura del robot da zero, combinando pezzi Lego (per i motori e i cingoli particolarmente affidabili) e parti meccaniche e strutturali varie.

La parte hardware è stata prevalentemente curata da Fenu Simone, mentre quella software prevalentemente da Locarno Marco. La logica di funzionamento è stata studiata e progettata da entrambi gli studenti.

Lo schema a blocchi seguente rappresenta in modo generale l'architettura del robot:



ROBOCUP JR ITALIA 2013 – PESCARA 17-20 aprile  
REPORT DI DOCUMENTAZIONE

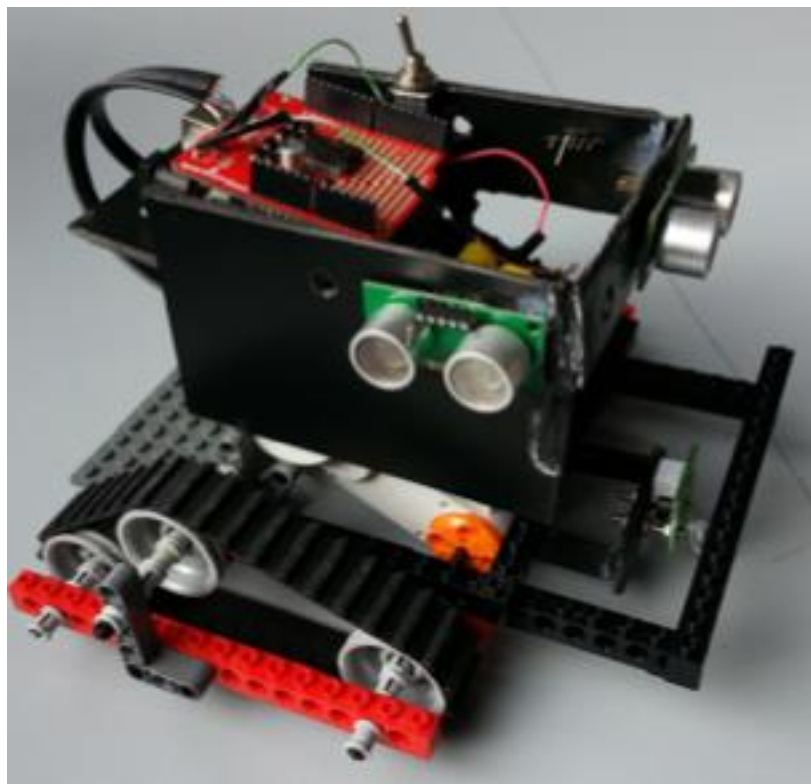
L'elenco dettagliato dei componenti principali è il seguente:

- Pacco pile da 9V
- Scheda Arduino Uno
- 2 Motori Lego NXT, collegati ai cingoli Lego
- 1 Motor Shield Ardumoto per il pilotaggio dei motori
- 1 LED (inserito in un piccolo circuito correttamente dimensionato) per la segnalazione delle vittime
- 1 Sensore analogico di luminosità RDI-202B
- 3 Sensori ad ultrasuoni SRF05, per la rilevazione delle pareti del labirinto
- 3 Sensori di temperatura ad infrarossi MLX90614 in impostazione SMBus
- Circuito stampato diviso in tre piani che va a sostituire molti collegamenti altrimenti impossibili da gestire

(Nota: il circuito stampato non è presente nelle foto in quanto momentaneamente non ancora disponibile)

Ogni parte è spiegata in dettaglio nei relativi capitoli.

Segue una foto esplicativa del robot nel suo insieme, risalente al 27/03/13.



## CAP. 4 – MECCANICA

Per quanto riguarda lo scheletro generale del robot, abbiamo realizzato una struttura in plastica PVC saldamente fissata alla base, a sua volta costituita dai cingoli Lego e dai motori Lego NXT. Di fatto, la parte superiore è incollata ai motori e sostiene il deviatore (utilizzato per l'accensione), i sensori ad ultrasuoni e quelli di temperatura.

E' presente una parte di sostegno in PVC per il sensore di luminosità fissata direttamente ai motori. Il pacco pile è legato ai motori con degli elastici, per rendere veloce il cambio di questo in caso di necessità.

I cingoli Lego sono stati costruiti in modo da lavorare con trazione anteriore e rendere stabile il robot durante la salita. Questa parte è composta interamente da pezzi Lego.

La scheda Arduino e la motor shield sono solo appoggiate alla struttura in quanto esiste la necessità di staccare spesso la scheda per caricare nuovi programmi e cambiare il pacco pile.





## CAP. 5 – UNITÀ DI CONTROLLO

L'unità di controllo utilizzata è la scheda Arduino Uno, facile da programmare, funzionale ed in generale adatta alle nostre esigenze.

La scheda è stata alimentata tramite Ardumoto, a sua volta collegata direttamente all'alimentazione. Abbiamo optato per questa scelta in quanto alimentando solo Arduino non si sarebbe potuta ottenere una corrente sufficiente ad alimentare i motori, mentre alimentando la motor shield viene alimentata automaticamente anche la scheda principale.



Seguono i dati tecnici della scheda (fonte

<http://www.futurashop.it/allegato/7300-ARDUINOUNOREV3.asp?L2=CORRELATI&L1=7200-5976&L3=&cd=7300-ARDUINOUNOREV3&nvt=&d=24,50> )

- Microcontrollore: ATmega328
- Tensione operativa: 5 V
- Alimentazione:
  - da 7 a 12 V (tramite plug)
  - 5 V (tramite porta USB)
- Tensione di alimentazione (limiti): 6-20V
- Ingressi/uscite Digitali: 14 (di cui 6 possono essere utilizzate come uscite PWM)
- Ingressi analogici: 6
- Corrente Dc per pin I/O: 40 mA
- Corrente DC per pin 3,3 V: 50 mA
- Memoria Flash: 32 kB (di cui 0,5 kB utilizzati dal bootloader)
- SRAM: 2 kB
- EEPROM: 1 kB
- Velocità di Clock : 16 MHz

L'interfacciamento con il PC è stato possibile tramite connettore USB, che funge anche da alimentazione a 5V. Ciò è stato necessario per il caricamento del programma sul microcontrollore e per la comunicazione seriale (utile ad una rilevazione veloce dei dati dei sensori).

## CAP. 6 – SENSORI

- Sensore analogico di luminosità RDI-202



Il sensore di luminosità utilizzato funziona secondo una logica molto semplice.

Quando alimentato il LED emette continuamente radiazioni infrarosse (appunto non percepibili dall'occhio umano) che sono riflesse, a seconda dell'assorbimento luminoso dovuto al colore della superficie, e misurate dal ricevitore, che dà in uscita una tensione proporzionale alla lettura.

Tale valore può essere letto mediante l'ingresso analogico di Arduino con l'apposita funzione.

Non è servito un sensore di precisione in quanto serve solo a distinguere la differenza tra bianco e nero.

- Sensore ad ultrasuoni SRF05



Il sensore ad ultrasuoni SRF05 è un trasduttore piuttosto comune ma molto potente, che permette misure molto affidabili fino a 3-4 metri di distanza con precisione di qualche cm. Utilizzato per il rilevamento delle pareti (quindi entro una decina di centimetri) è risultato estremamente affidabile.

Abbiamo utilizzato tale dispositivo nell'impostazione per cui ingresso e uscita del sensore sono presenti su due linee distinte, per rendere veloce e snello il programma. Inviando un impulso della durata di  $10\ \mu\text{s}$  in ingresso, viene fatto vibrare un quarzo che emette appunto una raffica di ultrasuoni.

Il segnale inviato viene riflesso quasi totalmente (in base al materiale colpito) e captato dal ricevitore. Il sensore dà in uscita un impulso che rimane a livello alto in proporzione al tempo di risposta (se non riceve nessun segnale dà la durata massima dell'impulso). Questa durata è facilmente misurabile tramite la funzione "pulseIn" e convertibile in cm.

ROBOCUP JR ITALIA 2013 – PESCARA 17-20 aprile  
REPORT DI DOCUMENTAZIONE

- Sensore IR di temperatura MLX90614



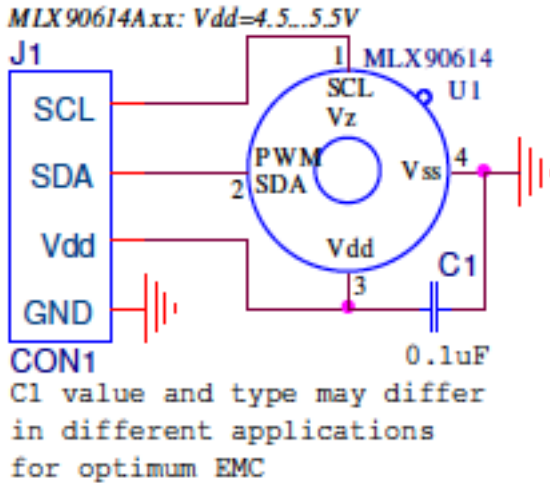
Il sensore MLX90614 risulta essere uno dei migliori trasduttori ad infrarossi di temperatura, utilizzato in molti campi, da quello medico a quello militare.

Ha un'altissima precisione (0,01 °C) e un range che copre l'intervallo da -40 a 125 °C.

Il sensore utilizzato è di tipo BAA, per cui funziona a 3,3V e contiene al suo interno un filtro.

Può essere utilizzato in impostazione SMBus (per cui lavora su due pin oltre a quelli dell'alimentazione) o in PWM. Abbiamo scelto il primo modo sia perché risulta essere estremamente preciso sia perché esisteva già una libreria apposita.

Il collegamento di un sensore è quello rappresentato nella figura seguente:



*(fonte: datasheet ufficiale della Melix)*

Dovendo lavorare con tre sensori (uno per ogni lato) abbiamo scelto di collegare assieme i pin SCL ed SDA (clock e data) in modo da utilizzare un'unica funzione per tutti e tre i sensori. Per usarne uno alla volta, quindi, abbiamo collegato l'alimentazione a tre pin di Arduino e abbiamo fatto in modo che uno solo fosse attivo mentre gli altri rimanevano spenti.

Dal momento che si tratta di tre sensori di tipo BAA (vedi datasheet) l'alimentazione è di 3,3V. Abbiamo per questo messo tre diodi in serie all'uscita i quali permettono una caduta di tensione fissa di circa 2,1V che porta quindi l'ingresso all'alimentazione del sensore a 2,9V (accettabile in quanto la soglia minima è di 2,4V).

## CAP. 7 – ATTUATORI

Gli unici due attuatori utilizzati sono i Motori Lego NXT, necessari per il movimento del robot; questi sono stati pilotati attraverso la motor shield Arduimoto che permette di gestire direzione e potenza dei due motori.



La motor shield è stata alimentata direttamente con 9V del pacco pile.

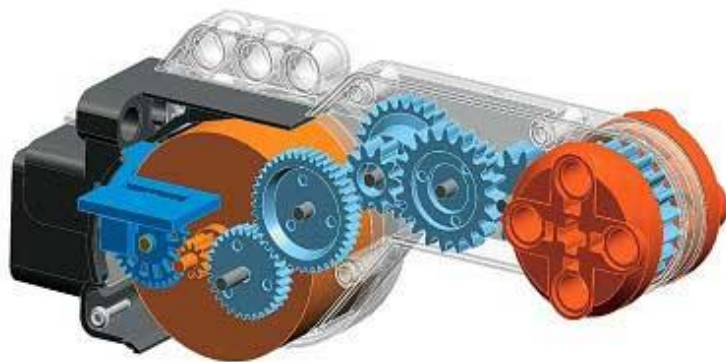
I motori Lego NXT di per sé sono destinati all'utilizzo tramite Brick Lego NXT, ma è stato piuttosto facile interfacciarli con Arduino. Abbiamo infatti tagliato il connettore finale e abbiamo studiato quali fossero i due fili principali (per il pilotaggio vero e proprio) che servivano a far funzionare il motore. Successivamente li

abbiamo utilizzati come dei normali motori tramite Arduino.

L'unico problema è che non è stato possibile usufruire delle diverse funzioni specifiche (ad esempio misurare e utilizzare le rotazioni in gradi del motore).

Per questo ultimo motivo il movimento è tutto basato sulla gestione dei tempi delle azioni (ad esempio: per fare una rotazione di  $90^\circ$  serve un ritardo di 1,5 secondi).

Segue la rappresentazione della sezione di un singolo motore.





## CAP. 8 – AMBIENTE DI SVILUPPO

Utilizzando Arduino, il linguaggio di programmazione e in generale l'ambiente di sviluppo è quello classico per questa piattaforma.

La programmazione è molto simile a quella del C/C++ e si basa su due funzioni principali (al posto del main): setup e loop. La prima contiene tutte quelle istruzioni che vanno eseguite solo una volta, ad esempio la dichiarazione dei pin come ingressi o uscite, mentre la seconda è un ciclo infinito che esegue continuamente tutte le istruzioni presenti al suo interno.

L'ambiente di sviluppo lavora a sketch, ovvero finestre che costituiscono tutte lo stesso programma, ma che rendono più facile la lettura e la gestione delle varie parti del programma.

La versione dell'ambiente di sviluppo è la 1.0.2



ROBOCUP JR ITALIA 2013 – PESCARA 17-20 aprile  
REPORT DI DOCUMENTAZIONE

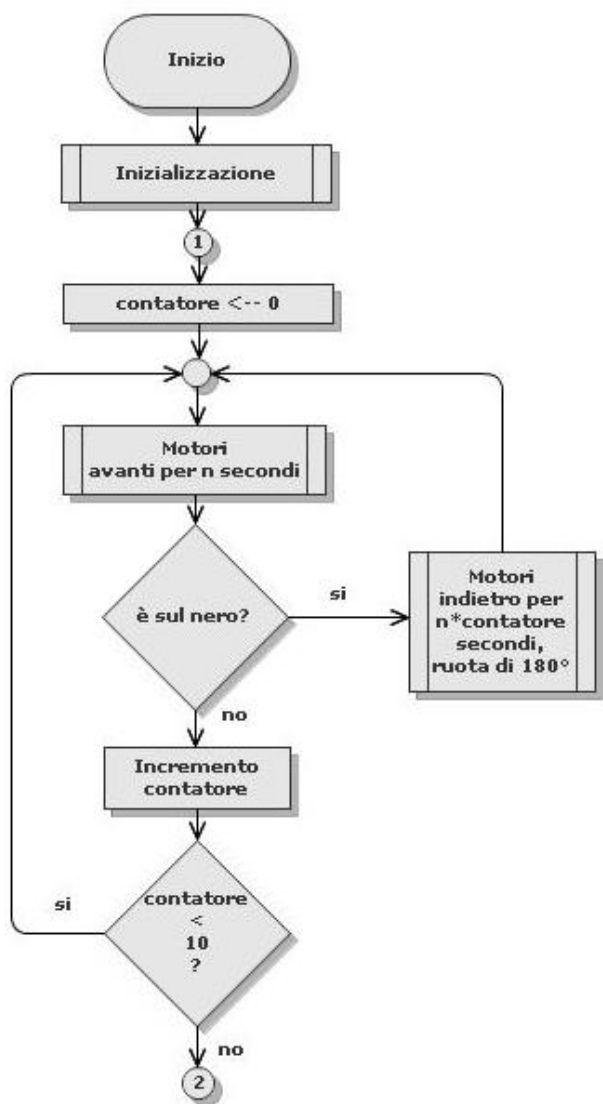
La struttura generale del programma si divide in tre parti:

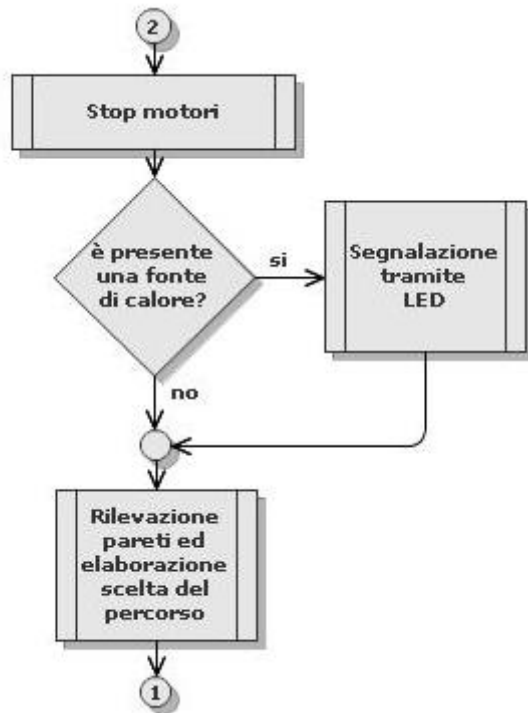
- setup (istruzioni eseguite una sola volta all'inizio del programma, inizializzazione pin e comunicazione)
- loop (ciclo infinito, corpo del programma)
- funzioni (divise in motori [a loro volta suddivise in avanti, indietro, destra, sinistra e stop] e nelle varie routine dei singoli sensori)

La strategia utilizzata per esplorare il labirinto consiste nel semplice metodo della mano destra, ovvero quello di dare una priorità alle possibili scelte (prima destra, poi avanti e infine, se unica possibilità, sinistra) per raggiungere la fine del labirinto (in questo caso esplorerà la maggior parte di esso).

Il vantaggio di questo metodo sta nella facile implementazione e nell'utilizzo di istruzioni singole piuttosto che l'introduzione di vettori o matrici (che probabilmente occuperebbero troppo spazio nella memoria del microcontrollore), mentre lo svantaggio è che, se presenti delle pareti non connesse ai muri principali, alcune zone potrebbero non essere visitate.

L'algoritmo estremamente semplificato del programma è il seguente:





Si rilevano dati interessanti per quanto riguarda i sensori. La lettura del sensore di luminosità infatti varia molto in funzione delle condizioni esterne, nonostante correttamente coperto, per cui di volta in volta sarà necessaria una calibrazione per ottenere valori accettabili.

Al contrario, il sensore ad infrarossi di temperatura fornisce dati di estrema precisione, per cui dalla lettura si possono apprezzare valori differenti di qualche

centesimo di grado. In realtà non è necessaria questa precisione, ma risulta una caratteristica utile in quanto rende evidente i pochi problemi tecnici che nasceranno da queste letture.

Di seguito vengono riportati due semplici frammenti di codice.

ROBOCUP JR ITALIA 2013 – PESCARA 17-20 aprile  
REPORT DI DOCUMENTAZIONE

- Funzione lettura del sensore US

```
int pinUSin=3; //Pin di input
int pinUSout=2; //Pin di output
int pinLED=12; //Pin collegato ad un LED
unsigned long temp=0; //Variabile per tempo
unsigned long dist=0; //Variabile per distanza

void US ()
{
  digitalWrite (pinUSout, HIGH); //Manda l'impulso
  delayMicroseconds (10);
  digitalWrite (pinUSout, LOW);
  temp= pulseIn (pinUSin, HIGH); //Riceve
                                   //l'impulso e
                                   //calcola il
                                   //tempo
  dist=temp/58; //Distanza in cm

  if (dist<50) //Se la distanza è minore di
               //50 cm
  {
    digitalWrite (pinLED, HIGH); //Accende il LED
    delay (1000); //Aspetta 1 secondo
    digitalWrite (pinLED, LOW); //Spegne il LED
  }
}
```

- Funzione Motori Avanti

```
#define PWR 100      //Potenza percentuale
                    //motori
int pinDirA=12;     //Pin Direzione Motore A
int pinDirB=13;     //Pin Direzione Motore B
int pinPwmA=3;      //Pin Pwm Motore A
int pinPwmB=11;     //Pin Pwm Motore B
int pwr=255/100*PWR;//Potenza convertita in
                    //scala 0-255

void ava ()        //Funzione Avanti
{
  digitalWrite (pinDirA, HIGH); //Direzione avanti
  digitalWrite (pinDirB, HIGH); //Direzione avanti
  analogWrite (pinPwmA, pwr); //Potenza motore A
  analogWrite (pinPwmB, pwr-26); //Potenza
                                //motore B
}
```



## CAP. 10 – SORGENTE DI ALIMENTAZIONE

L'alimentazione del robot durante la gara è completamente fornita dal pacco pile da 9V. Questo valore dipende leggermente dal tipo di pile utilizzate e da quanto sono cariche.

Durante il trasferimento dei programmi o per prove dei singoli sensori la sorgente di alimentazione è il PC tramite porta USB (che in questo caso fornisce 5V).

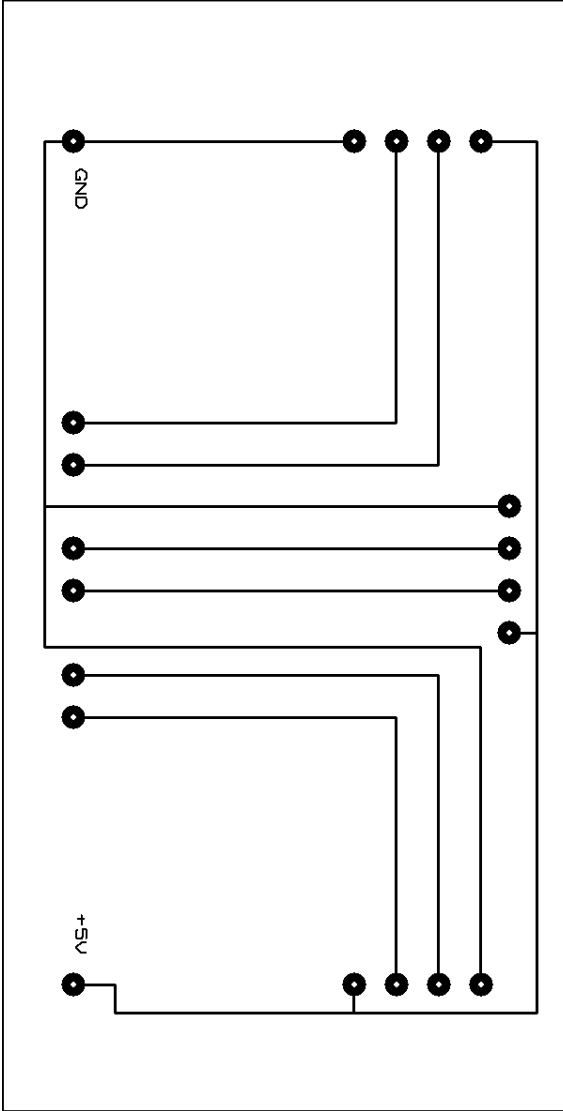


## APPENDICE - CIRCUITO STAMPATO

Come detto in precedenza, abbiamo fatto realizzare ed utilizzato i seguenti tre piani del circuito stampato per semplificare i collegamenti (più che altro quelli delle alimentazioni comuni).

Le tre immagini seguenti rappresentano (in ordine) i piani del circuito stampato.

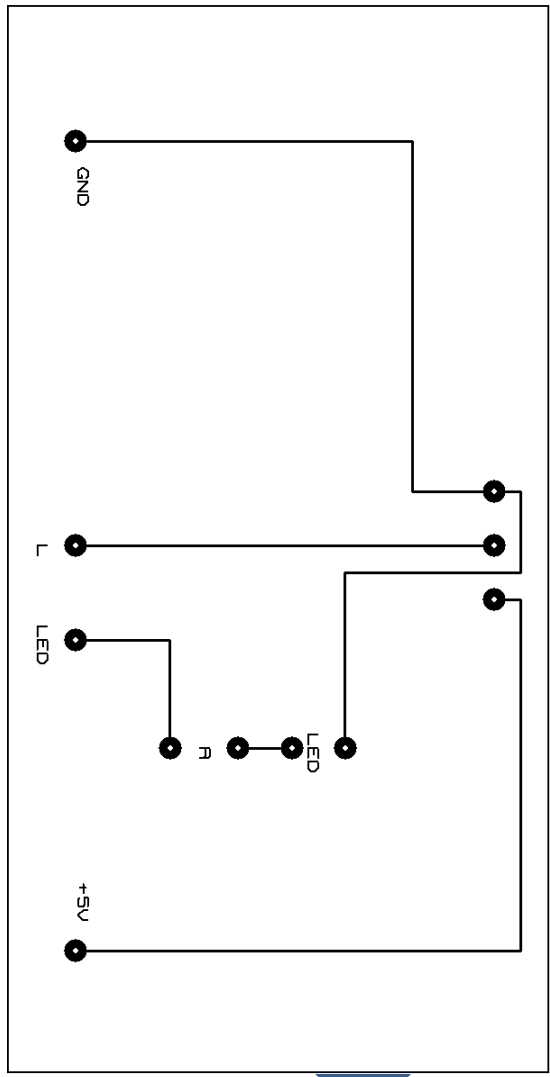
47002



3950E7

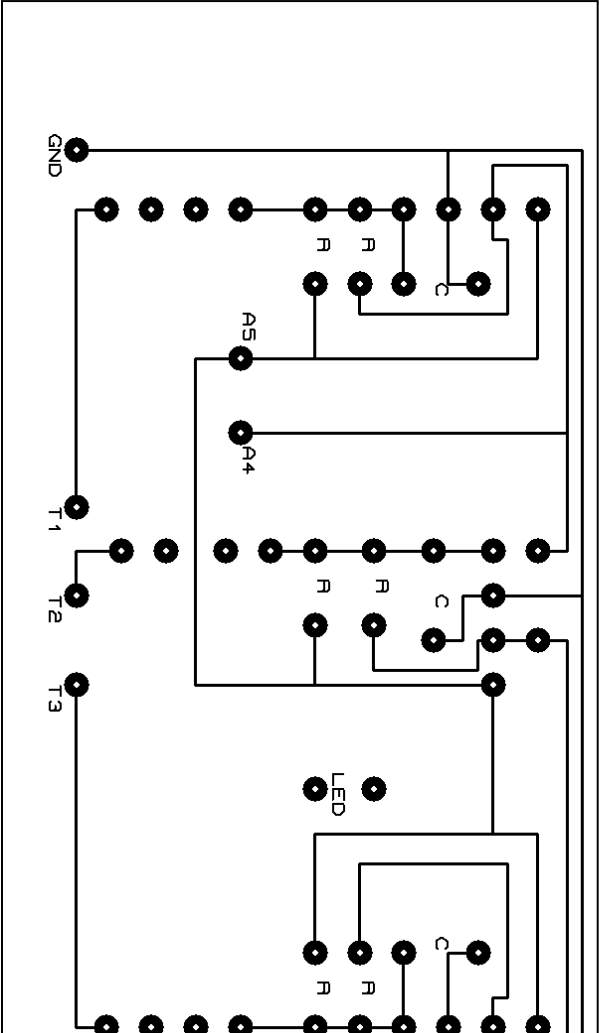
ROBOCUP JR ITALIA 2013 – PESCARA 17-20 aprile  
REPORT DI DOCUMENTAZIONE

← 49000 →



39500 →

← 2000FH →



↑ 3950FH

# INDICE

---

- P.7- CAP. 1 - DATI GENERALI
- P.9 - CAP. 2 - DATI DI CONTESTO E  
MOTIVAZIONE
- P.11- CAP. 3 – NOME E STRUTTURA DEL  
ROBOT
- P.15- CAP. 4 – MECCANICA
- P.17- CAP. 5 – UNITÀ DI CONTROLLO
- P.19- CAP. 6 – SENSORI
- P.23- CAP. 7 – ATTUATORI
- P.25- CAP. 8 – AMBIENTE DI SVILUPPO
- P.27- CAP. 9 – IL PROGRAMMA SOFTWARE
- P.33- CAP. 10 – SORGENTE DI  
ALIMENTAZIONE
- P.35- APPENDICE - CIRCUITO STAMPATO

