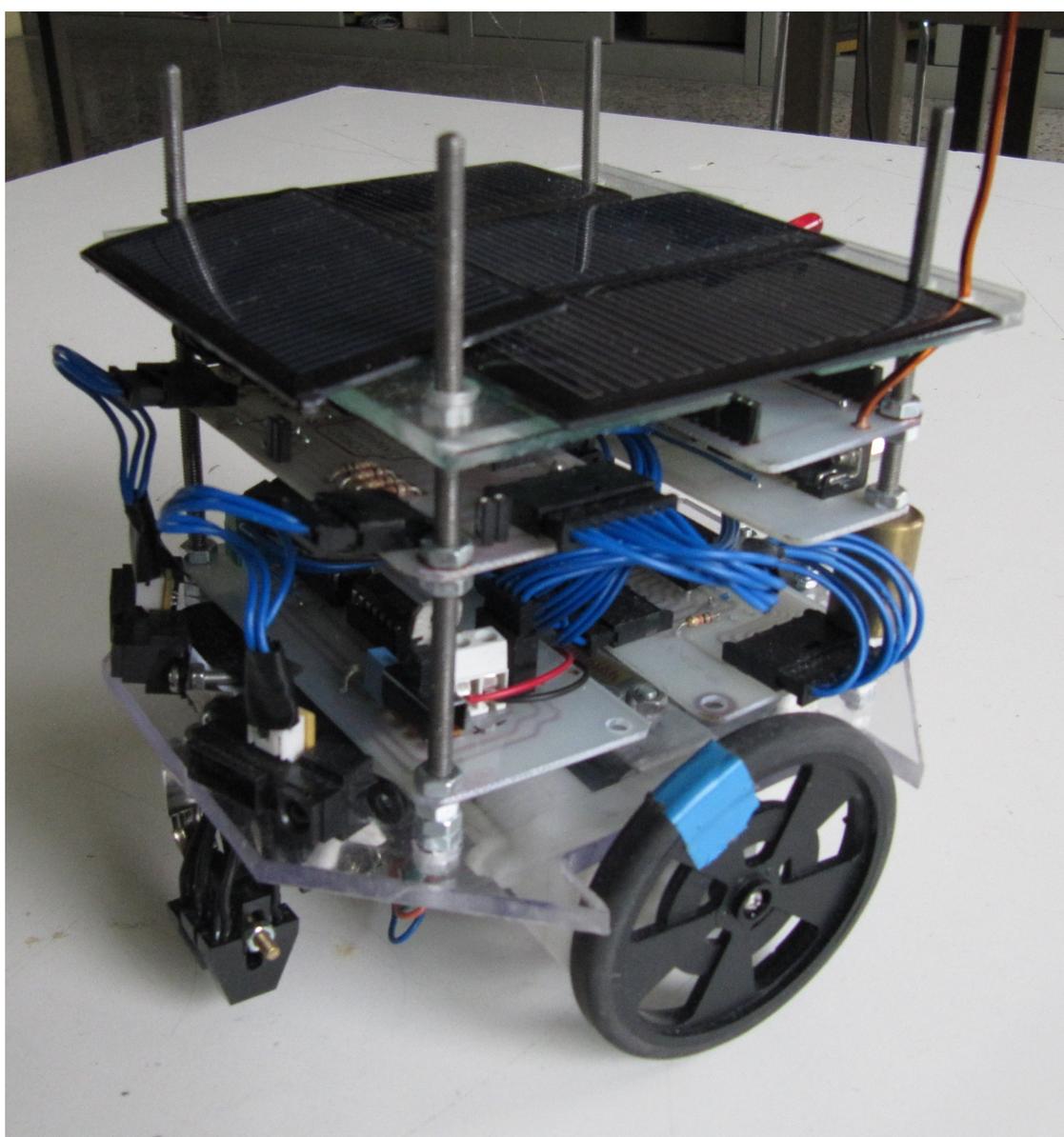




A.M.P.E.R.

Autonomous Multifunction Programmable Ecologic Robot

Un progetto di Coradi Silvio, Kalinskyy Volodymyr e Znacchi Zani Hari



	<p style="text-align: center;">ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA</p>	
	<p style="text-align: center;">RELAZIONE TECNICA – PROGETTO DI FINE ANNO</p>	<p style="text-align: right;">Pagina 2 di 115</p>

Indice generale

Obiettivi	5
Obiettivi futuri.....	6
Applicazioni pratiche del robot.....	7
Elementi costituenti il robot.....	9
Elenco.....	9
Elementi meccanici.....	12
Motoriduttori GM2.....	12
Ruote	12
Ball Caster.....	13
Telaio.....	13
Elementi sensoriali (sensori).....	15
FAIRCHILD GP2D120.....	15
FAIRCHILD QRB1134.....	17
Fotoresistenze.....	20
Elementi energetici.....	22
Pacco batterie.....	22
Celle solari.....	23
Elementi elettronici (integrati).....	25
Microcontrollore.....	25
Gestione del movimento.....	29
Caratteristiche dei sistemi di movimento.....	29
Strutture.....	29
Centro istantaneo di rotazione	30
Gradi di movimento.....	31
Grado di sterzabilità.....	32
Sistema a carro armato (dual drive).....	33
Introduzione.....	33
Calcolo dei parametri.....	34
Pulse width modulation (PWM).....	38
Driver motori.....	40
Locked anti-phase (LAP).....	41
Sign magnitude (SM).....	42
Microprocessore.....	44
Programma.....	44
Comunicazione robot-PC.....	55
Introduzione.....	55
EIA RS-232.....	55
Storia dello standard	55
Caratteristiche elettriche.....	56
Definizione.....	56

	ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA	
	RELAZIONE TECNICA – PROGETTO DI FINE ANNO	Pagina 3 di 115

Le unità di misurazione.....	56
Descrizione del segnale.....	57
MAX232.....	59
Comunicazione wireless.....	60
Comunicazione – Lato microprocessore.....	63
Parte hardware.....	63
Parte software.....	64
Comunicazione – Lato software.....	68
Finestre principali.....	69
Finestra index – Robot.....	69
Finestra del grafico dei segnali – Grafici	71
Finestra di trasmissione – Monitor di trasmissione.....	72
Finestre secondarie.....	73
Finestra di comunicazione – Comunicazione.....	73
Finestra di luce e pannelli solari – Luce.....	73
Finestra audio - Altoparlante.....	74
Finestra ausiliarie.....	75
Finestra versioni – Note di versione.....	75
Finestra informazioni – About	76
Finestra di caricamento – Splash Screen.....	77
Bibliografia.....	78
Sitografia.....	78
Fornitura.....	78
Licenza e copyright.....	79
Licenza di distribuzione Creative Common 2.5.....	79
Allegati.....	80
Allegato 01 – Schematico scheda controllo motori.....	81
Allegato 02 – PCB scheda controllo motori.....	82
Allegato 03 – Schematico scheda PIC sheet 1.....	83
Allegato 04 – Schematico scheda PIC sheet 2.....	84
Allegato 05 – PCB scheda PIC.....	85
Allegato 06 – Schematico scheda sensori.....	86
Allegato 07 – PCB scheda sensori.....	87
Allegato 08 – Schematico scheda seriale.....	88
Allegato 09 – PCB scheda seriale.....	89
Allegato 10 – Schematico scheda Scheda1.....	90
Allegato 11 – PCB scheda Scheda1.....	91
Allegato 12 – Schema scheda trasmettitore wireless.....	92
Allegato 13 – PCB scheda trasmettitore wireless.....	93
Allegato 14 – Listato funzioni manuale e esplorazione.....	94
Allegato 15 – Listato funzione luce.....	96
Allegato 16 – Listato funzione inseguì.....	99

	<p style="text-align: center;">ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA</p>	
	<p style="text-align: center;">RELAZIONE TECNICA – PROGETTO DI FINE ANNO</p>	<p style="text-align: right;">Pagina 4 di 115</p>

Allegato 17 – Listato funzione line-following.....	102
Allegato 18 – Listato funzione minisumo.....	104
Allegato 19 – Listato funzione suona.....	106
Allegato 20 – Robot 3D v1.....	108
Allegato 21 – Robot 3D v2.....	109
Allegato 22 – Robot 3D retro.....	110
Allegato 23 – Robot 3D v3.....	111
Allegato 24 – Scheda EasyPic 4.....	112
Allegato 25 – Datasheet TIP120.....	113

	ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA	
	RELAZIONE TECNICA – PROGETTO DI FINE ANNO	Pagina 5 di 115

Obiettivi

L'obiettivo del progetto è realizzare un dispositivo semovente in grado di svolgere determinate funzioni e di comunicare con un PC, il quale provvede a fornire istruzioni.

Gli obiettivi possono essere divisi in **primari**, **secondari**, e **terziari**; i primi sono le funzioni basilari del robot senza le quali non ha senso parlare di funzioni avanzate (es. movimento), le seconde sono appunto le funzioni avanzate necessarie a rendere il progetto più completo (es. inseguimento linea), le terze sono le funzioni non necessarie aggiunte unicamente per didattica o sperimentazione.

Il robot è stato pensato per essere completamente espandibile oltre le funzioni primarie; per questo l'obiettivo minimo è il completamento di questa categoria di funzioni mentre l'obiettivo massimo raggiungibile dipende unicamente dal tempo disponibile e dall'inventiva del gruppo. In questo modo abbiamo la sicurezza di ottenere un progetto completo indipendentemente dal tempo concessoci.

Segue un elenco dettagliato delle funzioni prefissate.

Obiettivi **primari** :

- **alimentazione** – La funzione più importante di tutte ovvero fornire al dispositivo l'energia necessaria a svolgere le proprie funzioni. Essendo un robot semovente l'energia deve essere disponibile a bordo sotto forma di batterie; è compito dell'alimentazione adattare i voltaggi alle varie parti del circuito.
- **meccanica** – Il dispositivo deve poggiare su una base rigida e adeguata in grado di resistere alle sollecitazioni richieste.
- **movimento** – Questa funzione si occupa della gestione del robot nello spazio, esso deve essere in grado di avanzare, indietreggiare e ruotarsi sia a destra che a sinistra variando la propria velocità in base alle esigenze. A discrezione può compiere curve più o meno accentuate.

Obiettivi **secondari** :

- **comunicazione** – Il robot deve essere in grado di comunicare efficacemente con un PC. Il robot invia al computer una serie di pacchetti contenenti i dati rilevati (es. sensori, velocità motori) e il computer provvede ad elaborare questi risultati e a inviare al robot le istruzioni necessarie.
- **esplorazione** – Il robot deve essere in grado di aggirarsi in un ambiente sconosciuto senza urtare oggetti o persone. Per questo motivo esso deve essere dotato di sensori in grado di rilevare questi ostacoli.
- **Inseguimento di linea** – Il robot deve essere in grado di riconoscere e seguire una linea scura su fondo chiaro, questi robot sono detti *line-follower* e sfruttano dei sensori di superficie in grado di riconoscere il colore del campo.

	<p style="text-align: center;">ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA</p>	
	<p style="text-align: center;">RELAZIONE TECNICA – PROGETTO DI FINE ANNO</p>	<p style="text-align: right;">Pagina 6 di 115</p>

- **Inseguitore solare** – Il robot deve essere in grado di riconoscere la direzione dalla quale proviene la sorgente luminosa più intensa e seguirla. Questa funzione è spesso associata a robot dotati di **pannelli solari** in grado di ricaricare le batterie e aumentare quindi l'autonomia generale.
- **minisumo** – Questa funzione si ispira al Sumo tradizionale, il robot deve allontanare da un'area delimitata tutti gli oggetti presenti spingendoli al di fuori del cerchio.

Obiettivi **terziari** :

- **altoparlante** – Il robot è in grado di generare dei suoni convenzionali e delle note. Questa funzione, oltre a essere ludica, serve per attirare l'attenzione sul robot o per segnalare errori.

Obiettivi futuri

Il progetto è, come già detto, completamente espandibile e quindi già predisposto ad aggiornamenti e modifiche future.

Le prossime modifiche saranno:

- **Potenziamento dei motoriduttori** – I motori verranno sostituiti per generare più coppia ed essere più veloci; a causa di questo sarà necessario adattare il pacco batterie e il controllo motori alle mutate esigenze.
- **Potenziamento meccanica** – Al robot verrà aggiunta una protezione in Lexan 5mm per aumentarne la resistenza.
- **Potenziamento comunicazione wireless** – La comunicazione wireless verrà affidata a potenti moduli ricetrasmittenti sulla banda di 2,4 Ghz per eliminare disturbi e falsi positivi.
- **Braccio robotico** – È prevista l'aggiunta di un braccio robotico controllato per poter interagire l'ambiente.

	<p style="text-align: center;">ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA</p>	
	<p style="text-align: center;">RELAZIONE TECNICA – PROGETTO DI FINE ANNO</p>	<p style="text-align: right;">Pagina 7 di 115</p>

Applicazioni pratiche del robot

Il robot può, grazie alla parte hardware, essere impiegato in molteplici applicazioni sia civili, sia ludiche, sia (potenziando meccanica ed elettronica) militari.

Segue una lista di applicazioni possibili nella disposizione attuale o compiendo piccole modifiche:

- **Robot aspirapolvere** – Il robot si aggira in un ambiente sconosciuto aspirando sporco e polvere; non è previsto un percorso predefinito ed il robot si aggira in maniera casuale cercando di coprire tutta l'area nel tempo previsto. Hardware necessario: sensori di distanza. Funzioni necessarie: esplorazione.
- **Robot tagliaerba** – Il robot si aggira in un ambiente sconosciuto tagliando l'erba tramite un filo attaccato ad un motore posto nella parte inferiore; non è previsto un percorso predefinito ed il robot si aggira in maniera casuale cercando di coprire tutta l'area nel tempo previsto. Hardware necessario: sensori di distanza. Funzioni necessarie: esplorazione.
- **Robot estintore** – Il robot attende nella stanza fino a quando non rileva una temperatura troppo elevata, da questo momento si aggira per l'ambiente in cerca della fonte di calore per segnalarla tramite impulsi sonori o, se dotato di estintore, per estinguerla. Hardware necessario: sensori di distanza, NTC, altoparlante/estintore. Funzioni necessarie: esplorazione, inseguì oggetto, ricerca di luce (sostituendo le fotoresistenze con delle NTC questa funzione diventa un inseguitore di calore).
- **Robot da competizione – Minisumo** – Il robot può partecipare a competizioni di Minisumo. Hardware necessario: sensori di distanza, sensori di superficie. Funzioni necessarie: minisumo.
- **Robot da competizione – Line-follower** – Il robot può partecipare a competizioni di inseguimento di linea. Hardware necessario: sensori di distanza, sensori di superficie. Funzioni necessarie: line-follower.
- **Photovore** – Il robot è in grado di riconoscere la sorgente luminoso più forte nel suo ambiente ed avvicinarsi, i pannelli solari forniranno così più corrente. Hardware necessario: fotoresistenze, celle solari. Funzioni necessarie: inseguitore solare.
- **Robot treno** – I robot (sono richieste più unità) sono in grado di seguire un robot leader in modo affine a come avviene in una coda. Hardware necessario: sensori di distanza. Funzioni necessarie: inseguitore di oggetto.
- **Robot trasportatore** – Il robot può trasportare oggetti da un punto all'altro seguendo un percorso prestabilito tracciato da una linea nera. Hardware necessario: sensori di distanza, sensori di superficie. Funzioni necessarie: line-follower.
- **Robot rilevatore di passaggio / Robot allarme** – Il robot è in grado, se posizionato correttamente, di rilevare il passaggio di alcuni oggetti; inoltre può calcolarne la velocità ed emettere segnali acustici di allarme. Hardware necessario: sensori di distanza, altoparlante.

	<p style="text-align: center;">ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA</p>	
	<p style="text-align: center;">RELAZIONE TECNICA – PROGETTO DI FINE ANNO</p>	<p style="text-align: right;">Pagina 8 di 115</p>

Funzioni necessarie: nessuna.

- **Robot rilevatore crepuscolare** – Il robot, dotato di una sorgente di luce, può variare questa luce in base alla luce rilevata. È importante un controllo che evita di rilevare la luce emessa. Hardware necessario: fotoresistenze, sorgente di luce. Funzioni necessarie: funzione apposita realizzata ex-novo.
- **Robot rover** – Il robot è in grado di aggirarsi in un ambiente esterno, evitando ostacoli e buche, per fornire informazioni ad una base grazie ad una telecamera. Hardware necessario: sensori di distanza, sensori di superficie. Funzioni necessarie: esplorazione.
- **Robot antimina** – Il robot può essere comandato a distanza e seguito da una telecamera per rilevare ed eventualmente far detonare delle mine inesplose. Al giorno d'oggi questa funzione è di vitale importanza poiché, in caso di esplosione accidentale, la perdita consiste solo in un robot invece di vite umane. Hardware necessario: sensori di distanza, sensori di superficie. Funzioni necessarie: comando a distanza.
- **Robot spia** – Il robot può essere comandato a distanza per spiare ambienti nemici; può essere usato sia in ambito militare che in ambito civile con le stesse specifiche. Hardware necessario: sensori di distanza, sensori di superficie. Funzioni necessarie: comando a distanza.

	<p style="text-align: center;">ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA</p>	
	<p style="text-align: center;">RELAZIONE TECNICA – PROGETTO DI FINE ANNO</p>	<p style="text-align: right;">Pagina 9 di 115</p>

Elementi costituenti il robot

Elenco

Componenti utilizzati :

- Componenti meccanici
 - 2 Motoriduttori GM2
 - 2 Ruote plastiche Ø 70 mm
 - 1 Ball caster Tamiya
 - 1 Telaio 15cm x 15cm in Lexan® spessore 5mm
- Componenti sensoriali (sensori)
 - 2 Sensori di distanza GP2D120
 - 3 Sensori di superficie QRB113
 - 3 Fotorisistenze
 - 3 NTC
- Componenti energetici
 - 4 Celle solari 7V 60mA 0.42W
 - 6 Batterie SKB 1.2V 1100mAh
- Componenti elettronici
 - 1 Microcontrollore 16F886 28 pin
 - 1 74HC04N – NOT
 - 1 L293D – Driver motor
 - 1 7805 – Voltage regulator
 - 2 XTR 903 – Aurel RTX 433 Mhz
 - 1 Darlington TIP120
 - 1 LED
 - 1 MAX232
 - 1 Dip-switch 41
 - 1 Pulsante

	<p style="text-align: center;">ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA</p>	
	<p style="text-align: center;">RELAZIONE TECNICA – PROGETTO DI FINE ANNO</p>	<p style="text-align: right;">Pagina 10 di 115</p>

- Componenti passivi
 - Resistenze
 - 03 220Ω
 - 01 4,7 KΩ
 - 11 10KΩ
 - Condensatori
 - 2 100uF
 - 1 330nF
 - 4 100nF
 - 1 10uF
 - 4 1uF
- Connettori
 - 1 Connettore 2 poli
 - 2 Connettore 3 poli
 - 2 Connettore 4 poli
 - 2 Connettore 5 poli
 - 3 Connettore 6 poli
 - 2 Connettore 8 poli
 - 2 Connettore 10 poli
 - 3 Morsetti
 - 1 Presa seriale femmina

	<p style="text-align: center;">ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA</p>	
	<p style="text-align: center;">RELAZIONE TECNICA – PROGETTO DI FINE ANNO</p>	<p style="text-align: right;">Pagina 11 di 115</p>

Strumenti utilizzati :

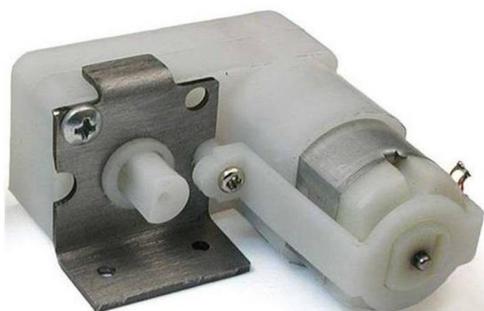
- Oscilloscopi
 - Oscilloscopio digitale Tektronix TDS 220 100MHz 1GS/s
 - Oscilloscopio analogico Eurotek DLIN 7025 20MHz
- Analizzatore di spettro
- Generatore di funzioni
 - TCELLNER TCE7404
- Alimentatori
 - GW Instek GPS-3303 3 CH
- Sistema di sviluppo
 - Programmatore PICmicro in-line
 - EasyPic 5
- Multimetro
 - GBC KDM-350CTF

Software utilizzati :

- Borland Delphi 5 – IDE
- Borland Delphi 7 – IDE
- Microcap 9 – Simulazione
- MPLAB 8.43 – IDE microcontrollore
- Eagle 4.16.2 – Schematici e PCB
- ICProg – Programmazione microcontrollore
- EasyPic Software – Programmazione microcontrollore

Elementi meccanici

Motoriduttori GM2



Dati tecnici	
Rapporto di riduzione	224:1
Coppia nominale	3 Kg*cm
Coppia frizione	3.5 Kg*cm
Velocità di rotazione	38 rpm
Consumo nominale	52 mA
Consumo in stallo	600 mA

I motoriduttori sono essenzialmente dei motori con delle opportune riduzioni in grado di ridurre la velocità ed

aumentare la coppia.

Ogni riduzione è caratterizzata da un rapporto di riduzione (n o i) il quale determina di quanto coppia e velocità vengono variate, essi vengono usati per permettere ad un motore di spostare carichi più elevati di quanto potrebbe fare senza riduzioni.

I motoriduttori usati (visibili in figura) sono realizzati completamente in plastica per ridurre peso e costo, sono stati fissati al telaio tramite delle staffe metalliche realizzate su misura.

La coppia, in questi motoriduttori, può essere prelevata su due mozzi posizionati sullo stesso asse.

Un'altra caratteristica di questi particolari motoriduttori è la possibilità di sostituire il motore con altri con diverse caratteristiche; è stato provato un motore più potente e veloce il quale però assorbiva troppo per il driver motor (412mA).

Ruote



Dati tecnici	
Diametro ruota	63mm
Spessore gommatura	3,5mm
Diametro totale	70mm
Raggio totale	35mm

Per poter trasferire la coppia dei motoriduttori a terra sono necessarie due ruote realizzate in plastica del diametro di 63 mm (70 mm con gomma), queste ruote sono gommate per garantire una migliore aderenza ed evitare slittamenti.

Sono pensate per potersi adattare agevolmente ai motoriduttori usati e possono essere fissate all'asse con una sola vite.

	<p style="text-align: center;">ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA</p>	
	<p style="text-align: center;">RELAZIONE TECNICA – PROGETTO DI FINE ANNO</p>	<p style="text-align: right;">Pagina 13 di 115</p>

Ball Caster



Dati tecnici

Altezza massima	35mm
Altezza minima	25mm
Diametro sfera	12mm
Materiale sfera	acciaio

Poiché due punti di appoggio non sono sufficienti a mantenere in equilibrio la base è necessario aggiungerne un terzo situato fuori dall'asse dei motori.

Il problema di trovare un punto di appoggio il quale abbia idealmente attrito nullo viene risolto con l'inserimento di un ball-caster, questo dispositivo incorpora una sfera metallica in grado di ruotare liberamente riducendo

l'attrito a valori quasi trascurabili.

Questa soluzione è stata preferita, rispetto alla classica rotellina pivotante, per motivi di attrito e di disponibilità.

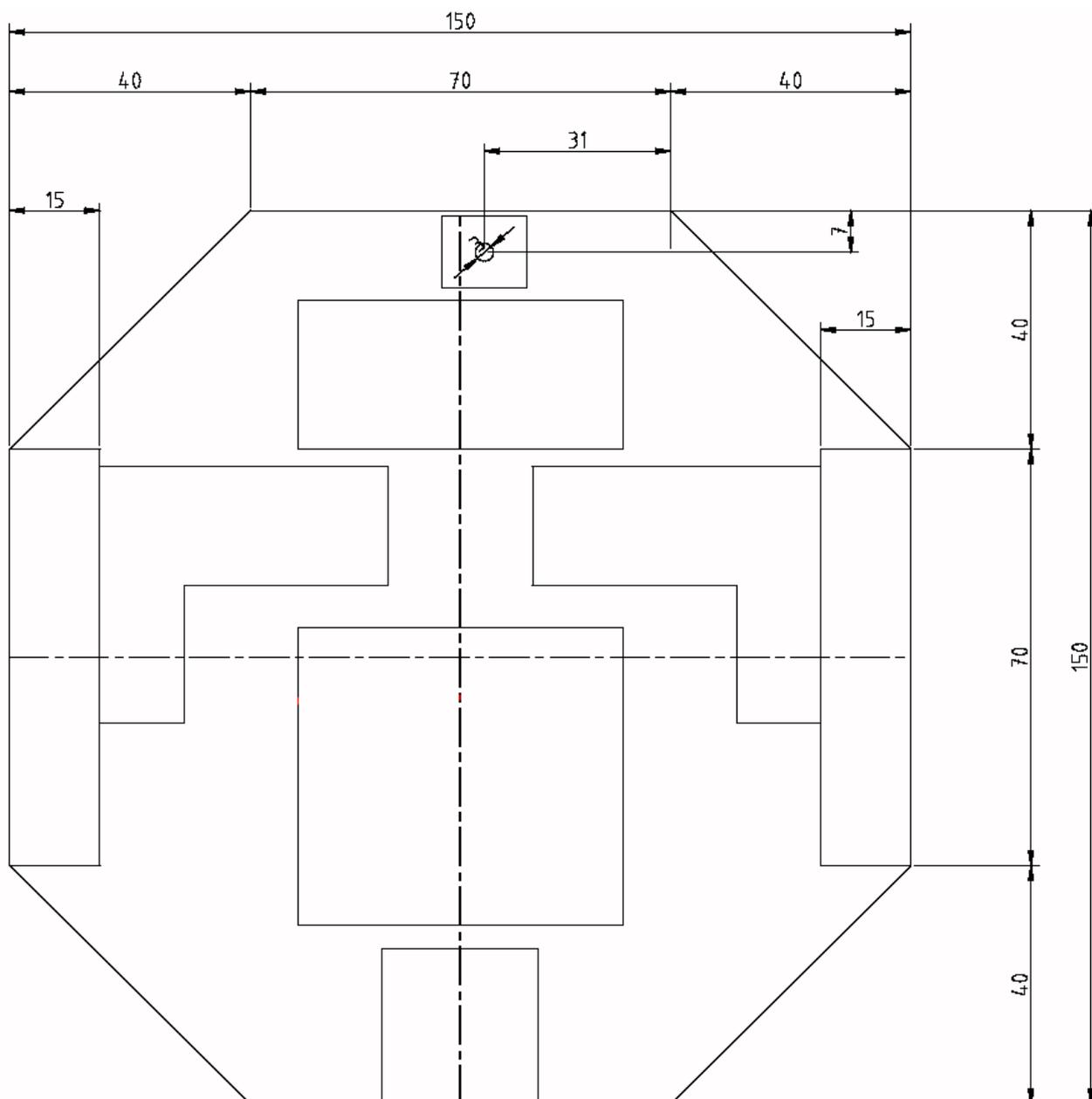
Telaio

Il telaio, essendo parte fondamentale della meccanica, è stato più volte modificato per adattarsi alle mutate necessità del robot.

Il primo prototipo è stato realizzato in polivinilcloruro (PVC) espanso di spessore 3 mm. Su questo telaio sono stati spostati svariate volte tutti i componenti principali, come motori e sensori, per trovare la configurazione migliore.

Con l'avanzare del progetto, e quindi del suo peso, il PVC cominciò a mostrare segni di cedimento in corrispondenza dell'asse dei motori, si rese quindi necessario sostituire il telaio con uno con un materiale più resistente.

Il secondo prototipo venne quindi realizzato in Lexan trasparente dallo spessore di 5 mm, questo materiale è in grado, con ampi margini, di sopportare le forze generate dai motori e dal peso del robot stesso.



Elementi sensoriali (sensori)

FAIRCHILD GP2D120



Illustrazione 2: FAIRCHILD GP2D120

(Ta=25°C, Vcc=5V)

Parameter	Symbol	Rating	Unit
Supply voltage	V _{cc}	-0.3 to +7	V
Output terminal voltage	V _o	-0.3 to V _{cc} +0.3	V
Operating temperature	T _{opr}	-10 to +60	°C
Storage temperature	T _{stg}	-40 to +70	°C

Illustrazione 3: Absolute maximum ratings

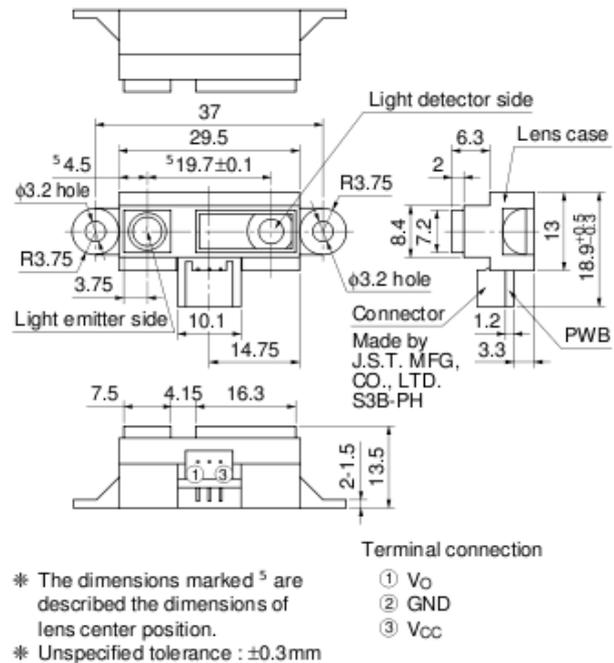


Illustrazione 1: Progetto del package

I sensori FAIRCHILD GP2D120 incorporano, in un unico package, un emettitore ed un ricevitore di segnali infrarossi oltre ad un circuito in grado di gestire questi due dispositivi.

Per effettuare una misurazione, il sensore invia un segnale infrarosso codificato (spot) il quale, in presenza di ostacoli, verrà poi ricevuto dal ricevitore; il circuito calcola il tempo trascorso tra invio e ricezione e fornisce in uscita un valore seguendo la curva di Fig.4 in Illustrazione 4.

Il sensore fornisce quindi un'uscita la quale varia da un massimo di circa 3V (con un ostacolo di circa 4cm) a un minimo di circa 0.1V (senza ostacoli).

L'uscita è inversamente proporzionale alla distanza e raggiunge il valore massimo a circa 4cm, la zona compresa tra 0 e 4cm non dovrebbe essere utilizzata poiché, come si può vedere nel grafico di Fig.4 in Illustrazione 4 (Analog Output Voltage vs.Distance to Reflective Object), non segue l'andamento generale del grafico.

Questo problema è stato risolto montando i sensori circa 2cm all'interno del robot per evitare letture inferiori a questo valore.

Fig.3 Analog Output Voltage vs. Surface Illuminance of Reflective Object

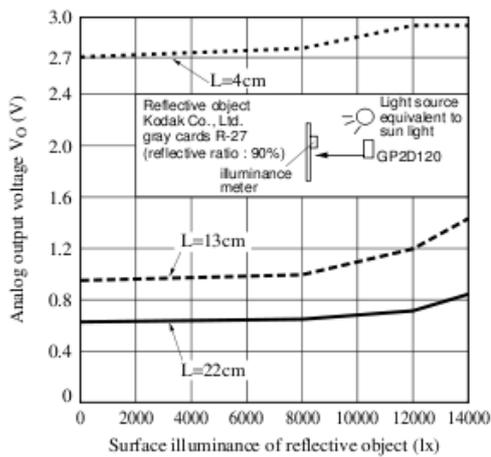


Fig.4 Analog Output Voltage vs.Distance to Reflective Object

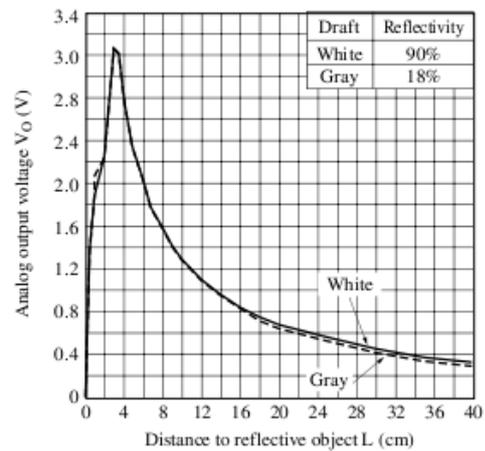


Fig.5 Analog Output Voltage vs.Ambient Temperature

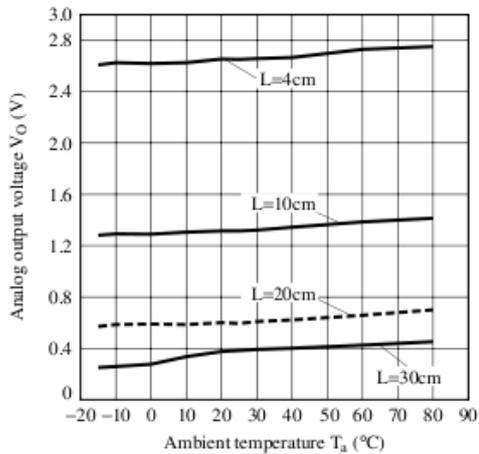


Fig.6 Analog Output Voltage vs.Detection Distance

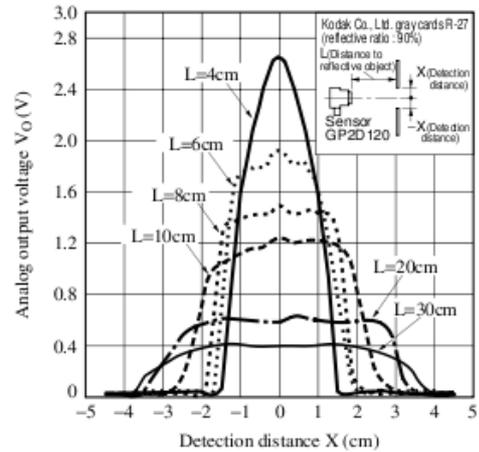


Illustrazione 4: Parametri di funzionamento

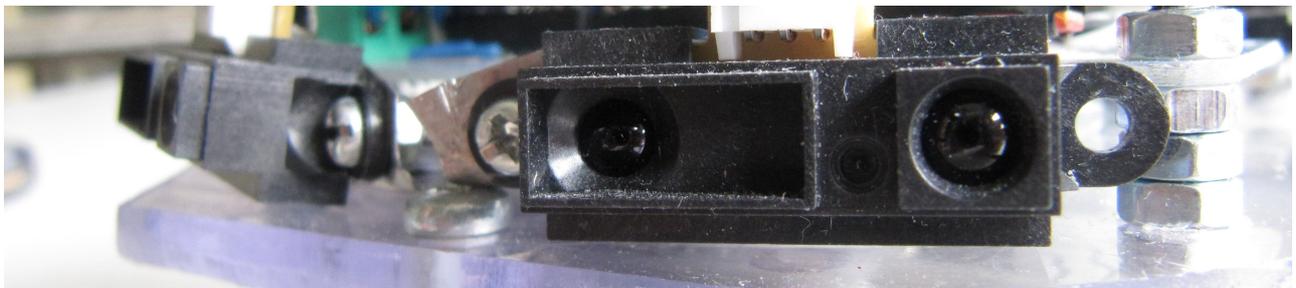


Illustrazione 5: Una macro dei sensori



FAIRCHILD QRB1134

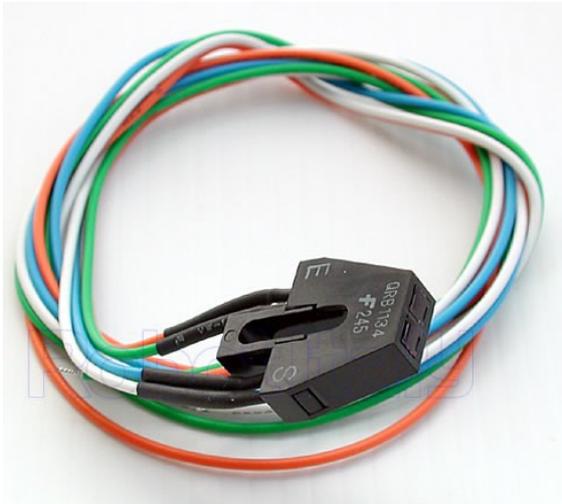
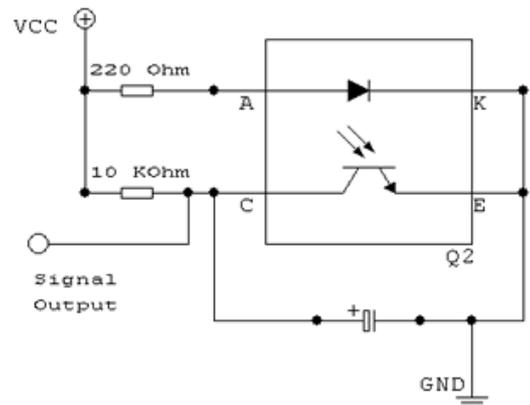


Illustrazione 6: FAIRCHILD QRB1134



A (Anode) = Orange C (Collector) = White
K (Cathode) = Green E (Emitter) = Blue

Schema 1: Collegamento del sensore

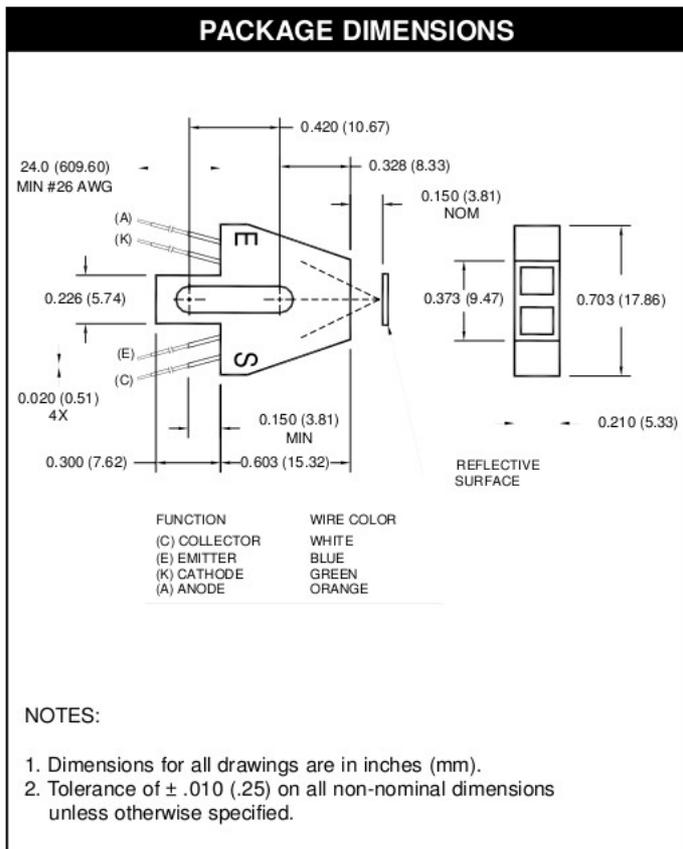
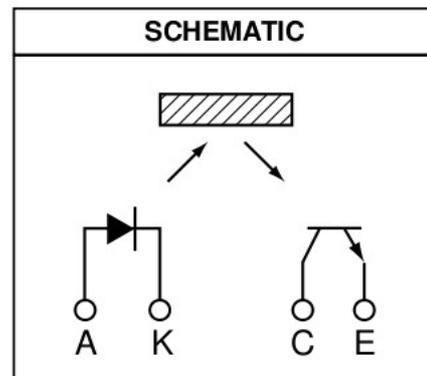
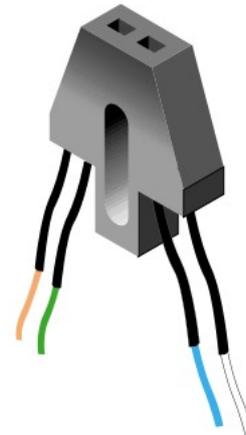


Illustrazione 7: Package e piedinatura





ABSOLUTE MAXIMUM RATINGS ($T_A = 25^\circ\text{C}$ unless otherwise specified)			
Parameter	Symbol	Rating	Units
Operating Temperature	T_{OPR}	-40 to +85	$^\circ\text{C}$
Storage Temperature	T_{STG}	-40 to +85	$^\circ\text{C}$
Soldering Temperature (Iron) ^(2,3,4)	T_{SOL-I}	240 for 5 sec	$^\circ\text{C}$
Soldering Temperature (Flow) ^(2,3)	T_{SOL-F}	260 for 10 sec	$^\circ\text{C}$
EMITTER			
Continuous Forward Current	I_F	50	mA
Reverse Voltage	V_R	5	V
Power Dissipation ⁽¹⁾	P_D	100	mW
SENSOR			
Collector-Emitter Voltage	V_{CEO}	30	V
Emitter-Collector Voltage	V_{ECO}	50	V
Collector Current	I_C	20	mA
Power Dissipation ⁽¹⁾	P_D	100	mW

Fig. 1 Forward Voltage vs. Forward Current

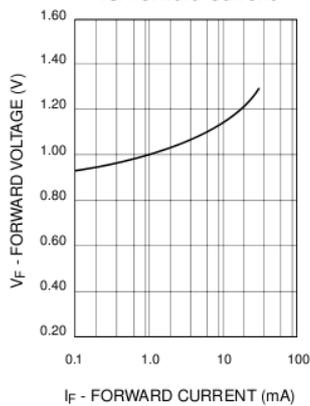


Fig. 2 Normalized Collector Current vs. Forward Current

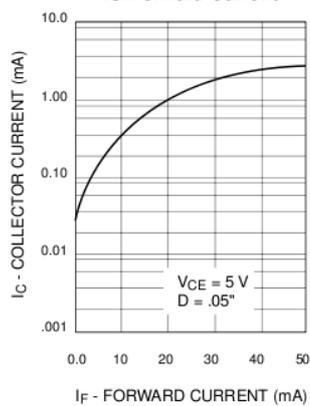


Fig. 3 Normalized Collector Current vs. Temperature

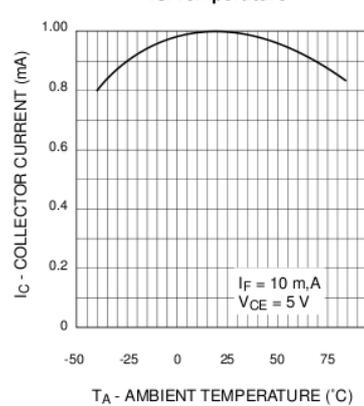


Fig. 4 Normalized Collector Dark Current vs. Temperature

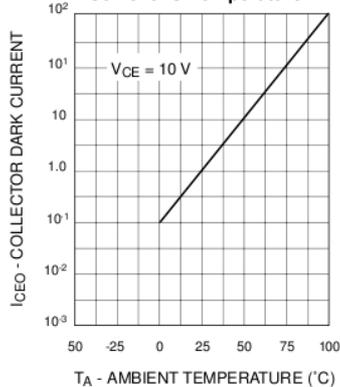
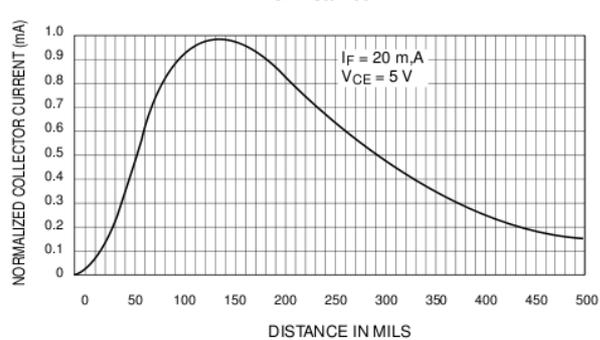


Fig. 5 Normalized Collector Current vs. Distance



	<p style="text-align: center;">ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA</p>	
	<p style="text-align: center;">RELAZIONE TECNICA – PROGETTO DI FINE ANNO</p>	<p style="text-align: right;">Pagina 19 di 115</p>

Questi sensori, come i precedenti, incorporano in un unico package un emettitore (fotodiode) e un ricevitore di infrarossi (fototransistor).

A differenza dei precedenti, questi sensori non si occupano di misurare una distanza bensì di discriminare il tipo di superficie; notoriamente una superficie nera assorbe la luce mentre una bianca la riflette.

Seguendo lo Schema 1 si può notare come, quando la superficie è bianca e quindi molta della luce viene riflessa, il fototransistor va in saturazione e il valore di uscita viene collegato a massa (livello logico basso); viceversa quando la superficie è scura il fototransistor rimane interdetto lasciando l'uscita collegata a Vcc (livello logico alto).

Questi sensori vengono usati, nel nostro progetto, per rilevare una linea di nastro nero isolante posta su una superficie chiara; ne vengono usati tre per rilevare la linea correttamente e sono utilizzati nelle funzioni *line-follower* e *minisumo*.

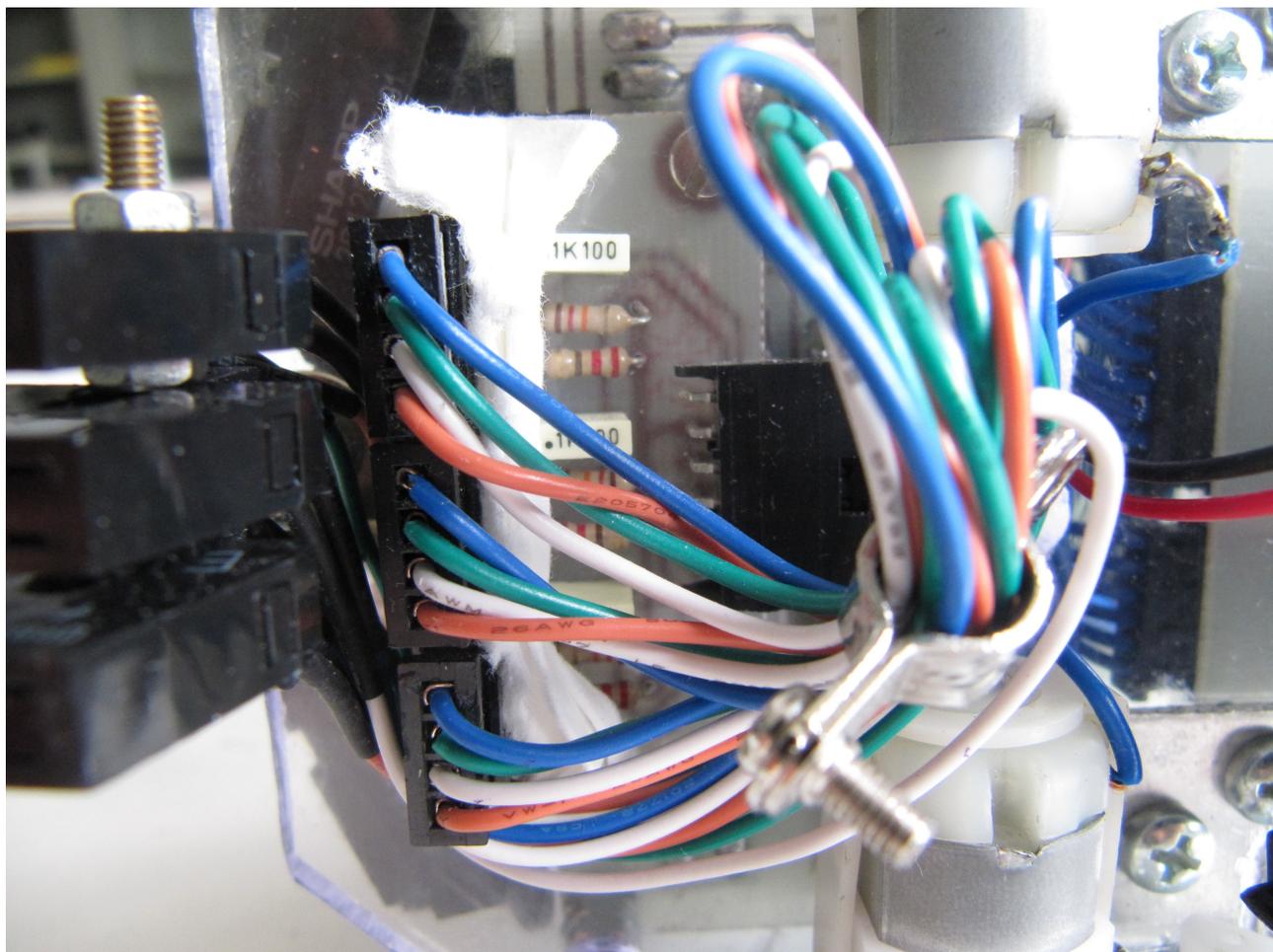


Illustrazione 8: Sensori di superficie con relativa sceda di gestione

Fotoresistenze

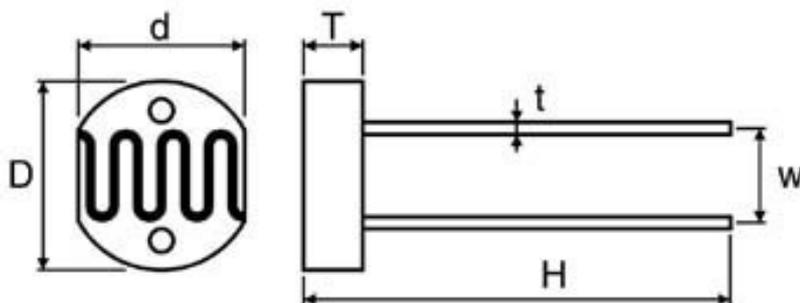
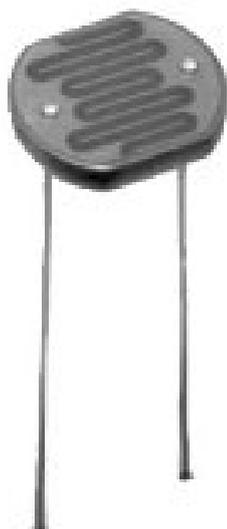


Illustrazione 10: Parametri meccanici di una fotoresistenza

Illustrazione 9: Una fotoresistenza

Tipica caratteristica di una fotoresistenza

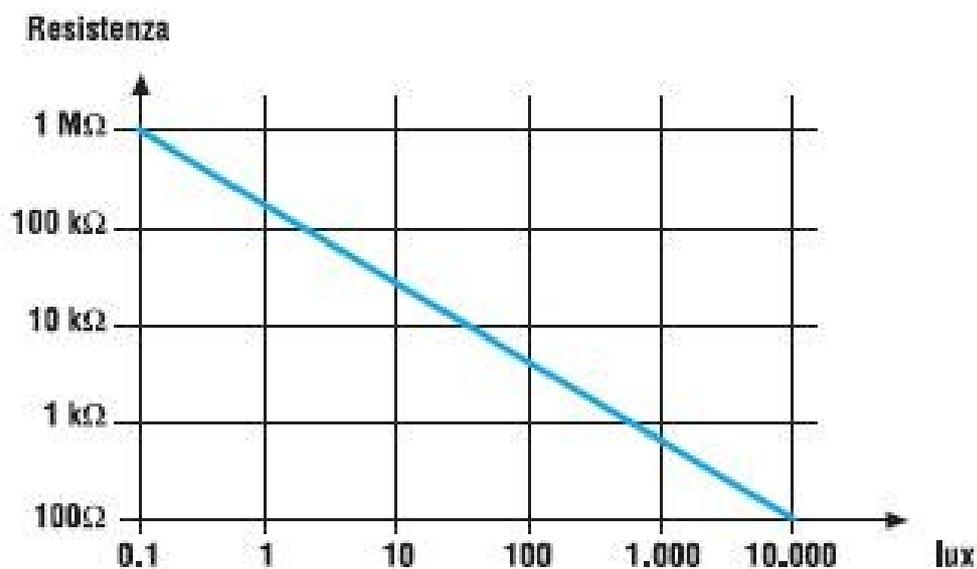


Grafico 1: Caratteristica di una fotoresistenza

	ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA	
	RELAZIONE TECNICA – PROGETTO DI FINE ANNO	Pagina 21 di 115

Una fotoresistenza è un sensore in grado di variare la resistenza ai propri capi al variare della luce incidente. È usata per rilevare in modo approssimativo la presenza o l'assenza di luce e in quale quantità; si trova spesso in rivelatori crepuscolari o luci intelligenti.

All'aumentare della luce diminuisce questa resistenza la quale, in presenza di forte luce, rientra nell'ordine delle centinaia di ohm, viceversa, in assenza di luce, raggiunge circa 1 MΩ.

Questa resistenza è esprimibile tramite la seguente formula (empirica e molto approssimata) :

$$R_{PHOTO}[\Omega] = 1000000 - 100 Lux$$

Questa formula è stata ricavata sia dal Grafico 1 sia da un campionamento della tensione di partitore al variare della luce incidente.

Poiché le fotoresistenze sono inserite in un partitore conosciuto, conoscendo la tensione ai loro capi possiamo conoscere anche la resistenza della fotoresistenza e, grazie alla formula precedente, ricavare il valore dei Lux; questa funzionalità è stata inserita sul programma di controllo e gestione ma non è stata approfondita per l'elevata imprecisione intrinseca.



I circuiti realizzati per la gestione delle fotoresistenze possono essere adattati a tutti i trasduttori i quali forniscono informazioni attraverso la variazione della propria resistenza.

Possono essere quindi usate indipendentemente fotoresistenze, NTC, PTC e qualunque trasduttore resistivo.

Illustrazione 11: Una NTC

	<p style="text-align: center;">ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA</p>	
	<p style="text-align: center;">RELAZIONE TECNICA – PROGETTO DI FINE ANNO</p>	<p style="text-align: right;">Pagina 22 di 115</p>

Elementi energetici

Pacco batterie



Illustrazione 12: Batteria usata

Il robot necessita di due valori di tensione per funzionare correttamente: una tensione per alimentare la logica compreso il PIC e gli integrati TTL e una tensione per alimentare i motori e i rispettivi driver.

Si è visto che, se microcontrollore e motori vengono alimentati con la stessa sorgente, il PIC risente dei disturbi e non funziona correttamente.

La logica TTL necessita di 5V e per ottenerli abbiamo sfruttato il diffusissimo 7805 in configurazione stabilizzatore di tensione.

Contando una V_{drop} tipica dello stabilizzatore di 2V il valore minimo di V_{in} diventa 7V.

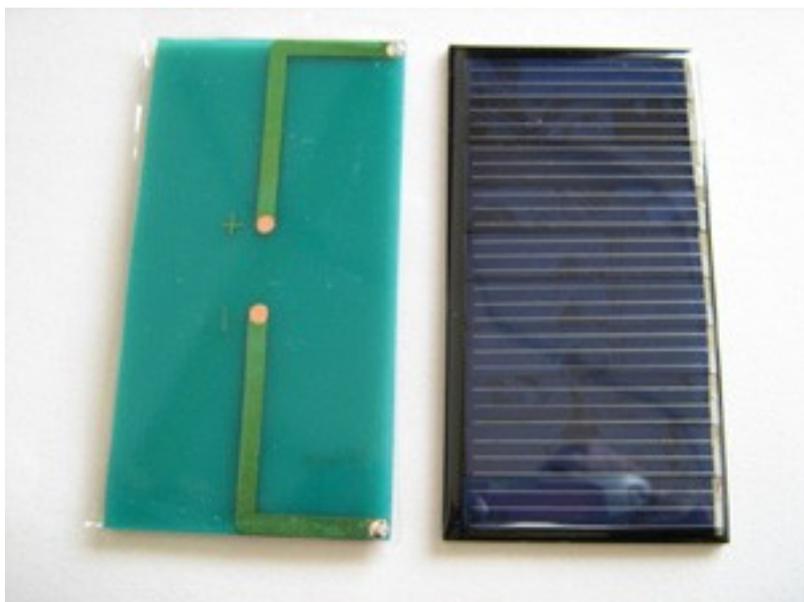
Per quanto riguarda la corrente assorbita la maggior parte dipende dai motori i quali (in coppia) consumano da 104 a 1200 mA con un consumo medio di 300 mA; volendo garantire un'autonomia di almeno 60 minuti abbiamo utilizzato delle batterie da 1100 mAh.

Nello specifico il pacco batterie è formato da 6 batterie ricaricabili Ni-Mh della SKB, la tensione nominale è di 1.2 V anche se dopo una ricarica presentano una tensione a vuoto di circa 1.35 V.



Illustrazione 13: Il pacco batterie

Celle solari



Ognuna delle celle usate fornisce al massimo 7 V e 60 mA per una potenza massima di 0.42 W.

Sono state usate quattro celle messe prima in serie e poi in parallelo a coppie per fornire in uscita 14 V e 120 mA.

In realtà se la luce incidente non è uniforme su tutti i pannelli alcuni di questi possono comportarsi anche da carico riducendo drasticamente la corrente di uscita, il valore massimo ricavato è stato di 13.56 V e 103 mA per un'efficienza media del 91%. La

potenza delle celle è quindi di circa 1.7 W nel punto di massima potenza.

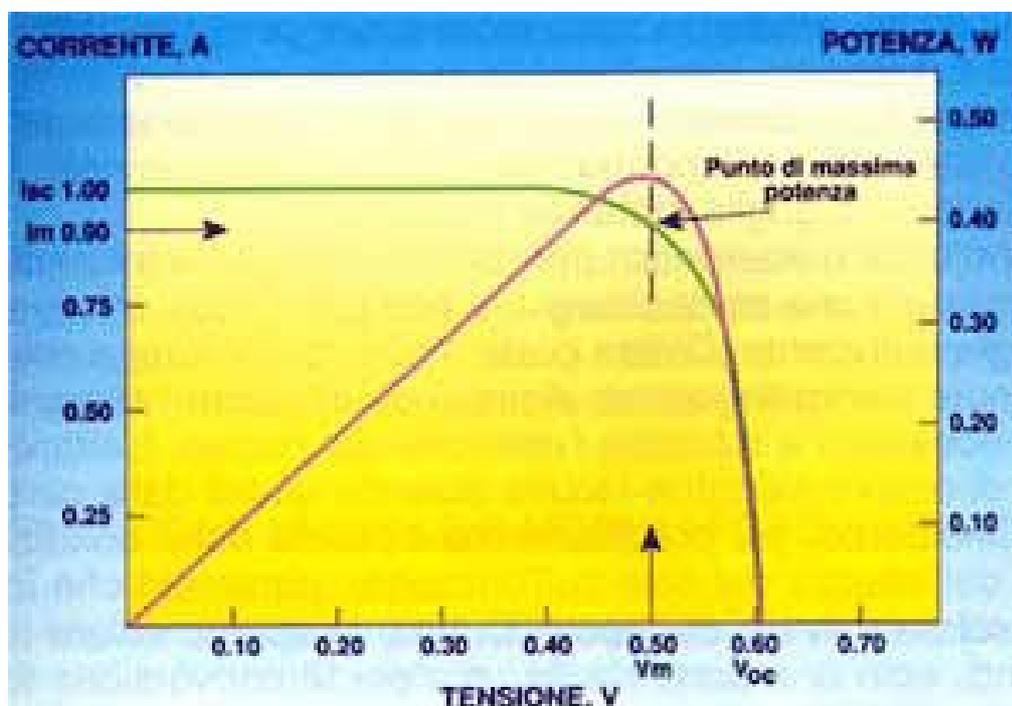


Illustrazione 15: Caratteristica elettrica di una cella solare

	<p style="text-align: center;">ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA</p>	
	<p style="text-align: center;">RELAZIONE TECNICA – PROGETTO DI FINE ANNO</p>	<p style="text-align: right;">Pagina 24 di 115</p>

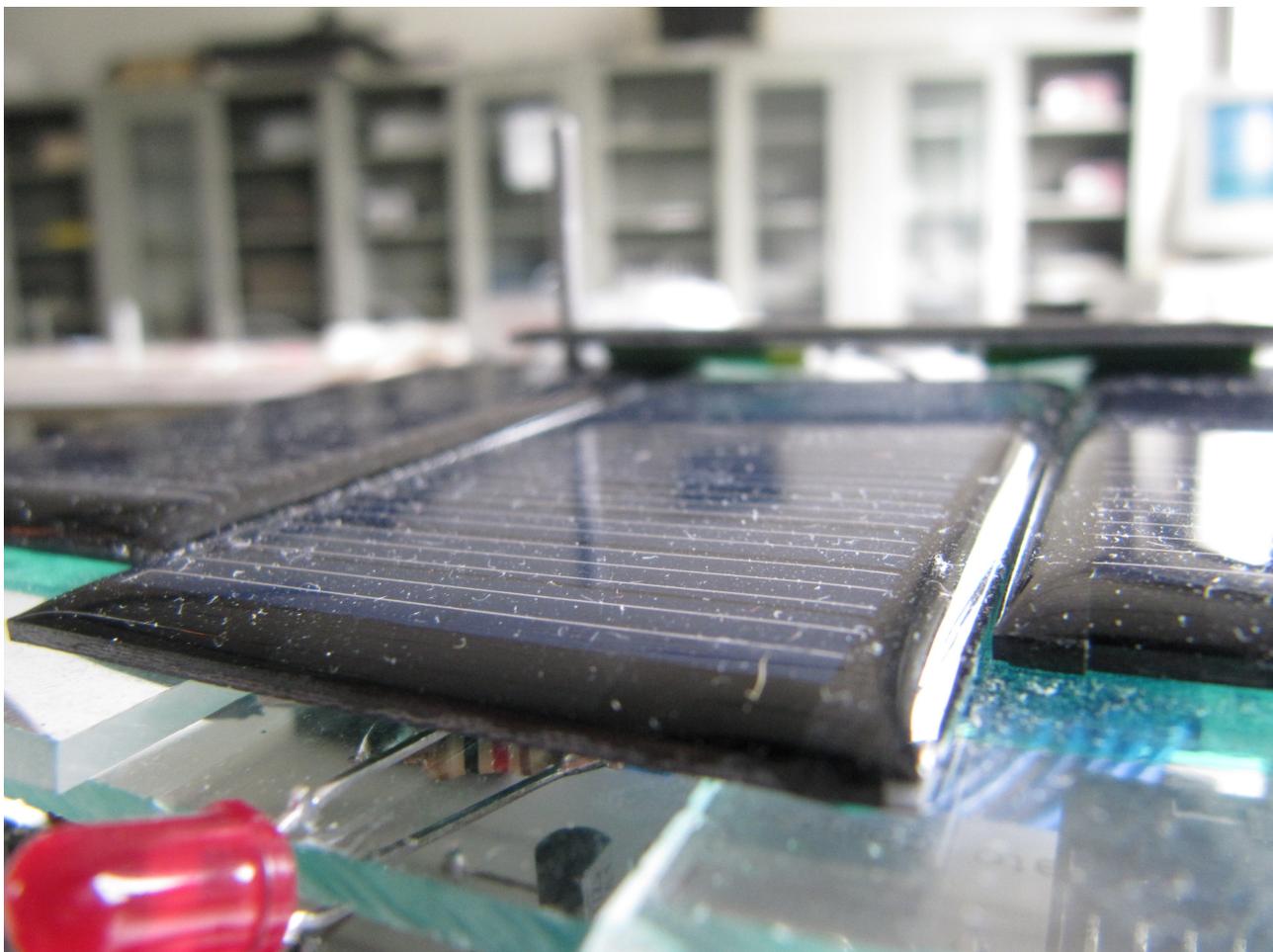


Illustrazione 16: Un ingrandimento sulle celle solari utilizzate

Queste celle non alimentano completamente il robot bensì forniscono una corrente in grado di ricaricare parzialmente il pacco batterie, il consumo medio del robot si aggira intorno ai 300 mA mentre i pannelli ne forniscono circa 120 mA ovvero il 40%.

Per limitare la corrente di carica è stato necessario utilizzare un LM317 in configurazione limitatore di corrente settato su 120mA con un diodo in serie per evitare che le batterie si scarichino sulle celle.

Elementi elettronici (integrati)

Microcontrollore

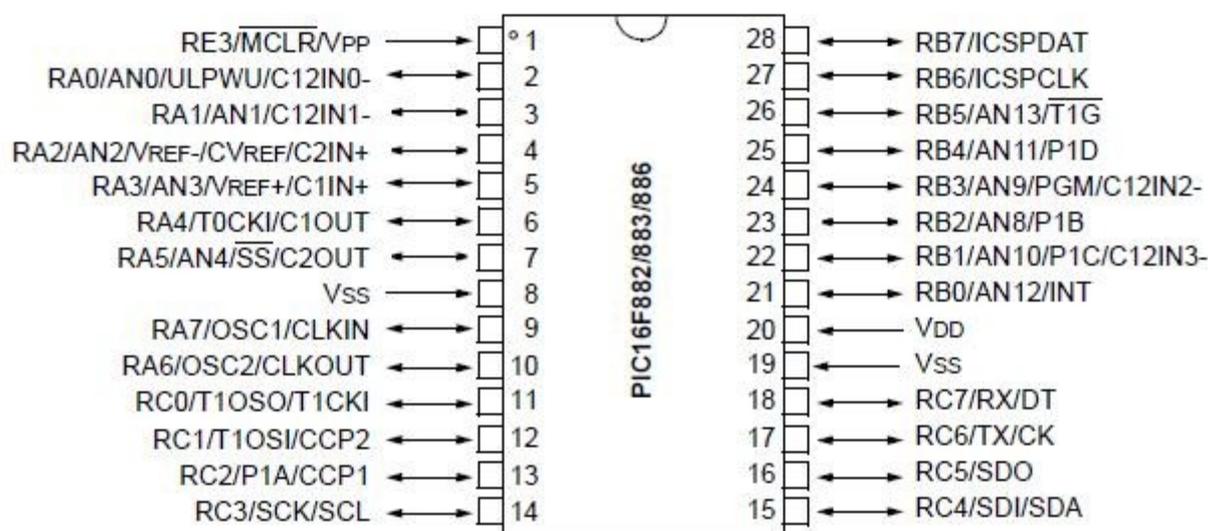


Illustrazione 17: Piedinatura 16F886

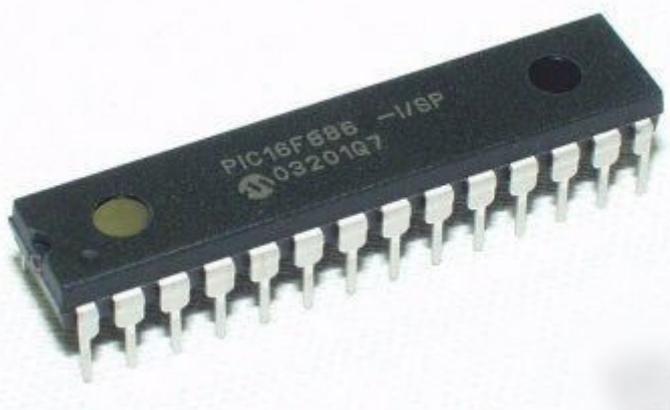


Illustrazione 18: Il microcontrollore usato

e ADC (analogic to digital converter); per realizzare lo stesso progetto con le stesse funzioni con l'elettronica tradizionale sarebbero stati necessari talmente tanti circuiti da non poter più utilizzare la medesima base a causa della sua dimensione.

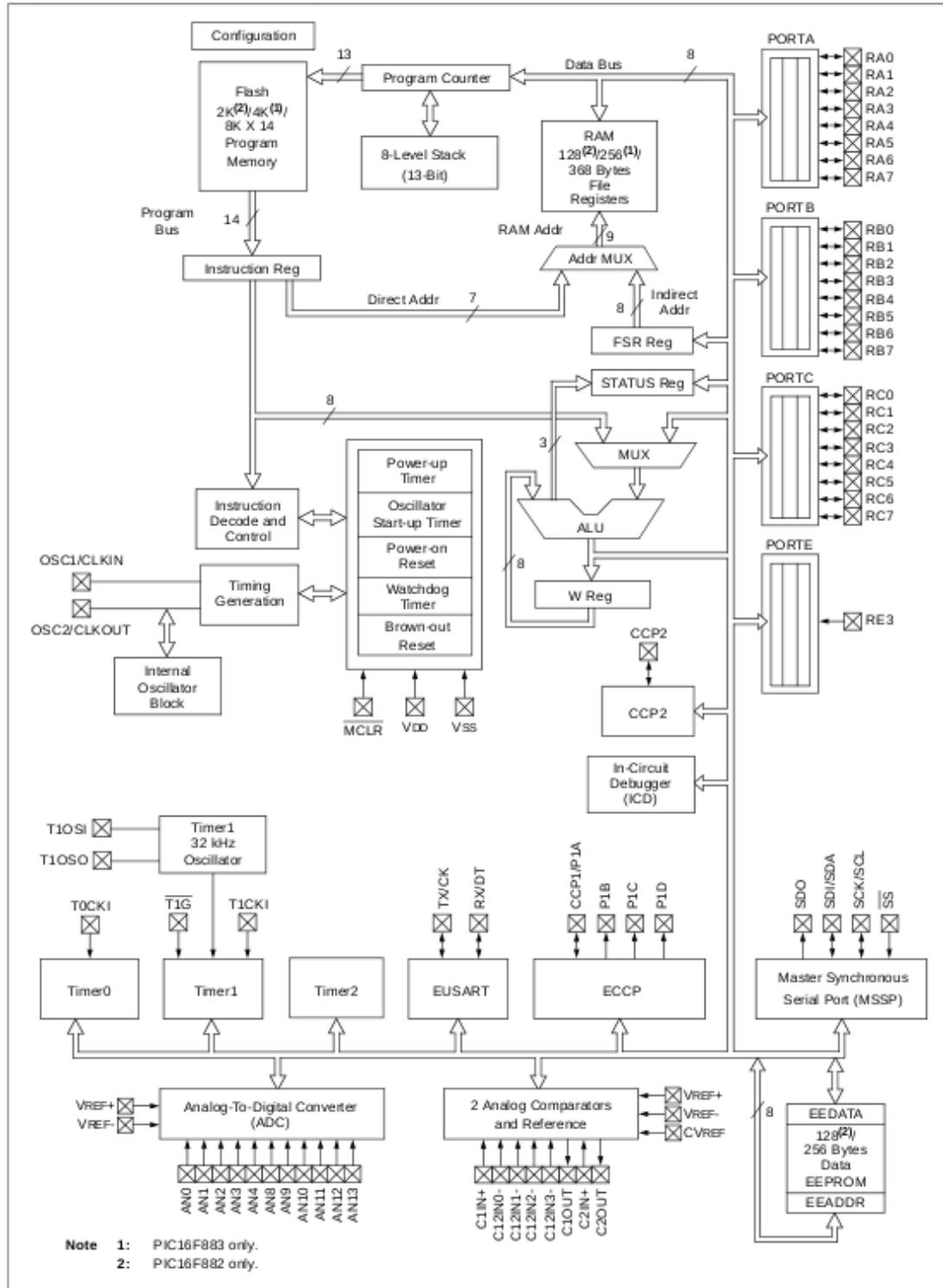
Seguono delle illustrazioni tratte dal datasheet della Microchip in grado di introdurre il PIC come schema a blocchi generale, funzioni dei pin ed elenco dei registri.

Il microcontrollore è sicuramente l'elemento più importante del progetto: si occupa di gestire ed interpretare tutto il traffico di dati provenienti dai sensori, dai motori e dal ricevitore wireless.

Grazie a questo elemento programmabile il progetto può essere espanso senza modificare la parte hardware ma semplicemente riprogrammando il PICmicro.

Il PIC incorpora al suo interno funzioni avanzate come USART, PWM generator

FIGURE 1-1: PIC16F882/883/886 BLOCK DIAGRAM



	ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA	
	RELAZIONE TECNICA – PROGETTO DI FINE ANNO	Pagina 27 di 115

TABLE 1: PIC16F882/883/886 28-PIN SUMMARY (PDIP, SOIC, SSOP)

I/O	Pin	Analog	Comparators	Timers	ECCP	EUSART	MSSP	Interrupt	Pull-up	Basic
RA0	2	AN0/ULPWU	C12IN0-	—	—	—	—	—	—	—
RA1	3	AN1	C12IN1-	—	—	—	—	—	—	—
RA2	4	AN2	C2IN+	—	—	—	—	—	—	VREF-/CVREF
RA3	5	AN3	C1IN+	—	—	—	—	—	—	VREF+
RA4	6	—	C1OUT	T0CKI	—	—	—	—	—	—
RA5	7	AN4	C2OUT	—	—	—	SS	—	—	—
RA6	10	—	—	—	—	—	—	—	—	OSC2/CLKOUT
RA7	9	—	—	—	—	—	—	—	—	OSC1/CLKIN
RB0	21	AN12	—	—	—	—	—	IOC/INT	Y	—
RB1	22	AN10	C12IN3-	—	P1C	—	—	IOC	Y	—
RB2	23	AN8	—	—	P1B	—	—	IOC	Y	—
RB3	24	AN9	C12IN2-	—	—	—	—	IOC	Y	PGM
RB4	25	AN11	—	—	P1D	—	—	IOC	Y	—
RB5	26	AN13	—	T1G	—	—	—	IOC	Y	—
RB6	27	—	—	—	—	—	—	IOC	Y	ICSPCLK
RB7	28	—	—	—	—	—	—	IOC	Y	ICSPDAT
RC0	11	—	—	T1OSO/T1CKI	—	—	—	—	—	—
RC1	12	—	—	T1OSI	CCP2	—	—	—	—	—
RC2	13	—	—	—	CCP1/P1A	—	—	—	—	—
RC3	14	—	—	—	—	—	SCK/SCL	—	—	—
RC4	15	—	—	—	—	—	SDI/SDA	—	—	—
RC5	16	—	—	—	—	—	SDO	—	—	—
RC6	17	—	—	—	—	TX/CK	—	—	—	—
RC7	18	—	—	—	—	RX/DT	—	—	—	—
RE3	1	—	—	—	—	—	—	—	Y ⁽¹⁾	MCLR/VPP
—	20	—	—	—	—	—	—	—	—	VDD
—	8	—	—	—	—	—	—	—	—	VSS
—	19	—	—	—	—	—	—	—	—	VSS

Note 1: Pull-up activated only with external MCLR configuration.



FIGURE 2-4: PIC16F882 SPECIAL FUNCTION REGISTERS

File Address	File Address	File Address	File Address
Indirect addr. ⁽¹⁾ 00h	Indirect addr. ⁽¹⁾ 80h	Indirect addr. ⁽¹⁾ 100h	Indirect addr. ⁽¹⁾ 180h
TMR0 01h	OPTION_REG 81h	TMR0 101h	OPTION_REG 181h
PCL 02h	PCL 82h	PCL 102h	PCL 182h
STATUS 03h	STATUS 83h	STATUS 103h	STATUS 183h
FSR 04h	FSR 84h	FSR 104h	FSR 184h
PORTA 05h	TRISA 85h	WDTCON 105h	SRCON 185h
PORTB 06h	TRISB 86h	PORTB 106h	TRISB 186h
PORTC 07h	TRISC 87h	CMLCON0 107h	BAUDCTL 187h
		CM2CON0 108h	ANSEL 188h
PORTE 09h	TRISE 89h	CM2CON1 109h	ANSELH 189h
PCLATH 0Ah	PCLATH 8Ah	PCLATH 10Ah	PCLATH 18Ah
INTCON 0Bh	INTCON 8Bh	INTCON 10Bh	INTCON 18Bh
PIR1 0Ch	PIE1 8Ch	EEDAT 10Ch	EECON1 18Ch
PIR2 0Dh	PIE2 8Dh	EEADR 10Dh	EECON2 ⁽¹⁾ 18Dh
TMR1L 0Eh	PCON 8Eh	EEDATH 10Eh	Reserved 18Eh
TMR1H 0Fh	OSCCON 8Fh	EEADRH 10Fh	Reserved 18Fh
T1CON 10h	OSCTUNE 90h		
TMR2 11h	SSPCON2 91h		
T2CON 12h	PR2 92h		
SSPBUF 13h	SSPADD 93h		
SSPCON 14h	SSPSTAT 94h		
CCPR1L 15h	WPUB 95h		
CCPR1H 16h	IOCB 96h		
CCP1CON 17h	VRCON 97h		
RCSTA 18h	TXSTA 98h		
TXREG 19h	SPBRG 99h		
RCREG 1Ah	SPBRGH 9Ah		
CCPR2L 1Bh	PWM1CON 9Bh		
CCPR2H 1Ch	ECCPAS 9Ch		
CCP2CON 1Dh	PSTRCON 9Dh		
ADRESH 1Eh	ADRESL 9Eh		
ADCON0 1Fh	ADCON1 9Fh		
	General Purpose Registers A0h		
	32 Bytes BFh		
	96 Bytes C0h		
	EFh		
	accesses F0h		
	70h-7Fh FFh		
Bank 0 7Fh	Bank 1	Bank 2 16Fh	Bank 3 1EFh
		accesses 170h	accesses 1F0h
		70h-7Fh 17Fh	70h-7Fh 1FFh

■ Unimplemented data memory locations, read as '0'.

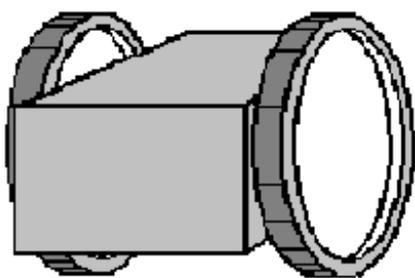
Note 1: Not a physical register.

	<p>ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA</p>	
	<p>RELAZIONE TECNICA – PROGETTO DI FINE ANNO</p>	<p>Pagina 29 di 115</p>

Gestione del movimento

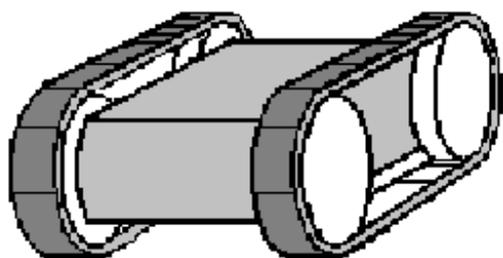
Caratteristiche dei sistemi di movimento

Strutture



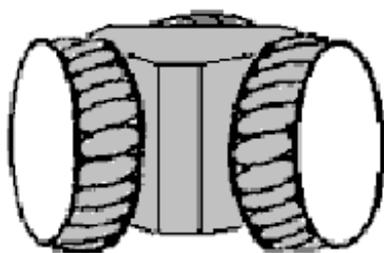
A due ruote

- Facile da modellizzare
- Rischi di scivolamento
- Sensibile al tipo di terreno



A cingoli

- Alta aderenza a terreni impervi
- Facile attraversamento di buche
- Odometria molto imprecisa



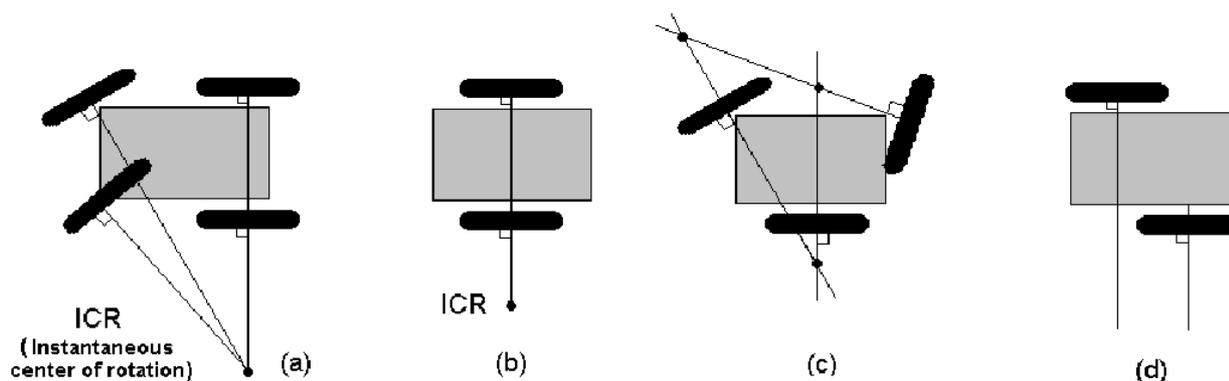
Omnidirezionale

- Massima libertà di movimento
- Struttura molto complessa

	ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA	
	RELAZIONE TECNICA – PROGETTO DI FINE ANNO	Pagina 30 di 115

Centro istantaneo di rotazione

Il centro istantaneo di rotazione (CIR) corrisponde al centro di rotazione in un determinato momento e si trova con l'intersezione di tutti gli assi delle ruote.



(a) Sistema a quattro ruote fisse, questo sistema è una semplificazione istantanea di un sistema a quattro ruote di cui due sterzanti e dipendenti; poiché in ogni istante le ruote saranno inclinate in un determinato punto tutto il sistema può essere ricondotto al caso (a). In questo sistema il CIR è quindi unico poiché vi è solo un'intersezione di tutti e quattro gli assi.

(b) Sistema a due ruote non sterzanti posizionate sullo stesso asse; poiché l'intersezione di due rette identiche è la retta stessa il numero di CIR è infinito e corrisponde appunto all'asse delle ruote. Questo sistema può effettuare curve di un qualsiasi angolo dalla rotazione sul posto all'avanzamento.

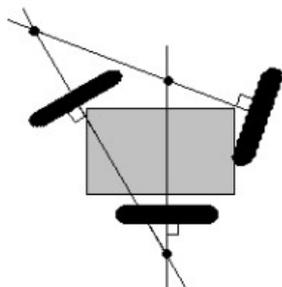
(c) Sistema a tre ruote fisse non sterzanti; gli assi di queste tre ruote non si incontrano in un solo punto perciò il CIR non esiste e non è possibile ricavarlo.

(d) Sistema a due ruote non sterzanti posizionate su assi differenti; come per il sistema precedente il punto di CIR non esiste e non è possibile ricavarlo.

	<p style="text-align: center;">ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA</p>	
	<p style="text-align: center;">RELAZIONE TECNICA – PROGETTO DI FINE ANNO</p>	<p style="text-align: right;">Pagina 31 di 115</p>

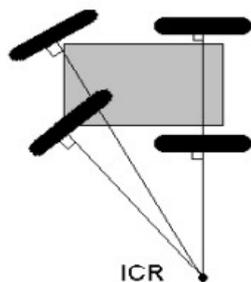
Gradi di movimento

Il grado di movimento di un sistema definisce quanto questo sistema possa muoversi nello spazio e con che capacità; più è alto il grado maggiore sarà questa capacità.



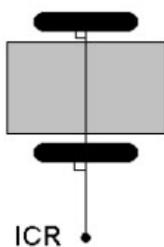
Il robot non può muoversi a causa dell'assenza di CIR

Gradi di mobilità : 0



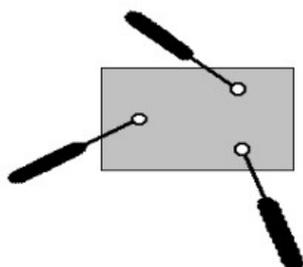
Arco fisso di movimento (1 CIR)

Gradi di mobilità : 1



Arco variabile di movimento (CIR distribuiti sulla linea)

Gradi di mobilità : 2

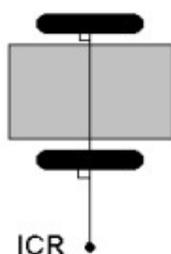
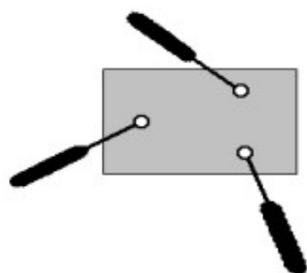


Movimento libero (CIR in qualsiasi punto)

Gradi di mobilità : 3

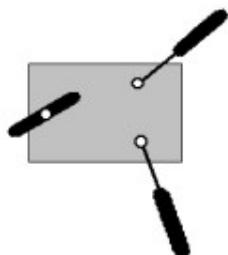
Grado di sterzabilità

Questo grado equivale al numero di ruote centrate orientabili che si muovono indipendentemente per sterzare il robot. È, insieme al grado di mobilità, un importante indice della capacità del robot di spostarsi nello spazio.



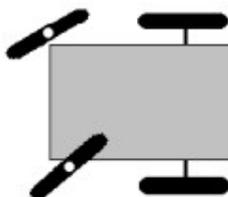
Non ci sono ruote in grado di sterzare ergo tutte le ruote sono fisse.

Gradi di sterzabilità : 0



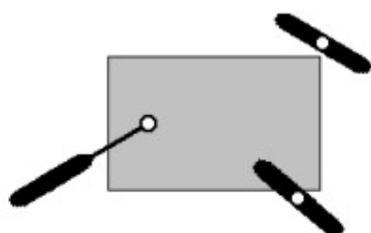
Una ruota centrata e orientabile.

Gradi di sterzabilità : 1



Due ruote dipendenti centrate e orientabili.

Gradi di sterzabilità : 1



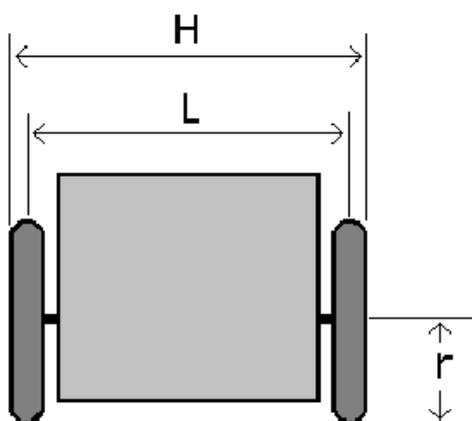
Due ruote indipendenti centrate e orientabili.

Gradi di sterzabilità : 2

	ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA	
	RELAZIONE TECNICA – PROGETTO DI FINE ANNO	Pagina 33 di 115

Sistema a carro armato (dual drive)

Introduzione



tipo (2,0)

Illustrazione 19: Sistema a due ruote fisse coassiali o dual-drive

Il movimento a carro armato, o dual-drive, è il sistema più utilizzato per muovere robot di piccole e medie dimensioni; questo sistema necessita di due ruote e di un punto di appoggio e permette, a differenza dei sistemi a ruote sterzanti, la rotazione sul posto.

Il robot si può muovere avanti e indietro e può compiere curve di qualsiasi grandezza fino alla rotazione sul posto; l'importanza di questo tipo di rotazione è da ricercarsi in ambienti angusti nei quali il robot non può permettersi nessun tipo di manovra.

Con questo sistemi si riducono così peso, costo, energia richiesta senza nessuno svantaggio rilevante.

I motori, posti sullo stesso asse, possono variare in direzione (avanti o indietro) e in velocità (da 0 a 100%); tutte le variazioni di questi due parametri permettono al robot di compiere moltissimi movimenti in due dimensioni.

Questo movimento può essere ricondotto ad un sistema di vettori posizionati ad una certa distanza; i vettori dei motori saranno caratterizzati da modulo (velocità del motore), verso (avanti o indietro) e direzione (fissa e uguale per entrambi); questi vettori saranno posizionati sullo stesso asse ad una distanza pari ad L (come visibile in Illustrazione 19).

Il baricentro del sistema sarà posizionato sul punto medio del segmento che unisce il mozzo delle due ruote e quindi la distanza tra ruota e baricentro sarà pari ad $L/2$.

	ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA	
	RELAZIONE TECNICA – PROGETTO DI FINE ANNO	Pagina 34 di 115

Calcolo dei parametri

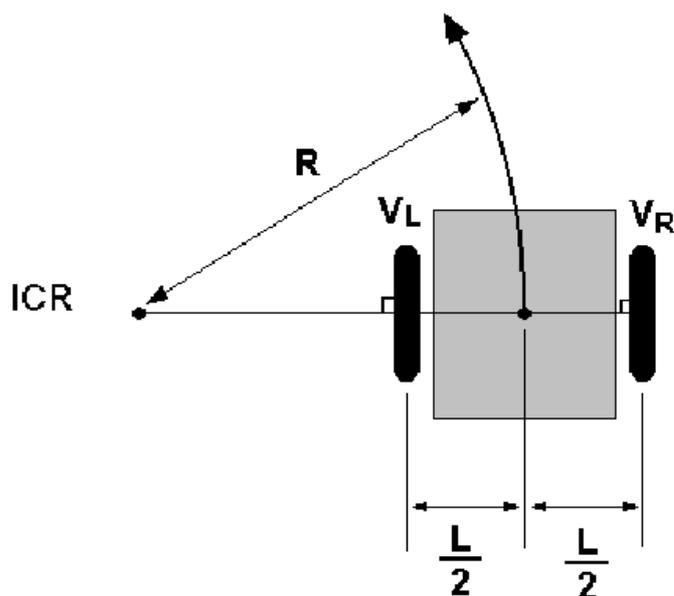


Illustrazione 20: Esempio di movimento

Chiamando R , V_R e V_L rispettivamente il raggio di curvatura, la velocità del motore destro e la velocità del motore sinistro possiamo ricavare le formule che legano questi parametri in modo da prevedere il movimento del robot conoscendo V_R e V_L .

Si suppone che il robot stia curvando a sinistra; nel caso opposto tutte le formule rimangono valide, semplicemente si inverte il segno per $L/2$.

La velocità di ogni ruota corrisponde alla velocità angolare moltiplicata per la lunghezza del raggio e perciò:

$$V_R = \omega \left(R + \frac{L}{2} \right)$$

$$V_L = \omega \left(R - \frac{L}{2} \right)$$

	ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA	
	RELAZIONE TECNICA – PROGETTO DI FINE ANNO	Pagina 35 di 115

$$\omega = \frac{V_R - V_L}{L} = \frac{V_R}{R + \frac{L}{2}}$$

$$v = \frac{V_R + V_L}{2}$$

La seguente formula ci permette, conoscendo le velocità dei motori, di calcolare il raggio di curvatura e di conseguenza le omega delle ruote.

$$R = \frac{L}{2} \frac{V_R + V_L}{V_R - V_L}$$

Le velocità dei motori possono essere espresse in percentuale e in qualsiasi unità di misura ad essa lineare; essa può variare da -100 a 100 poiché valori negativi corrispondono a verso opposto.

CASO 1) AVANZAMENTO

IPOTESI : $V_L = V_R = V_{MAX} = 100\%$

TESI : Il robot avanza in linea dritta

DIMOSTRAZIONE :

$$v = \frac{V_R + V_L}{2} = \frac{100 + 100}{2} = \frac{200}{2} = 100$$

$$R = \frac{L}{2} \frac{V_R + V_L}{V_R - V_L} = \frac{L}{2} \frac{100 + 100}{100 - 100} = \frac{L}{2} \frac{200}{0} = \infty$$

Si ha quindi un avanzamento alla massima velocità possibile con un raggio infinito; il raggio infinito può essere assimilato ad una retta e quindi il robot avanza in linea retta.



CASO 2) INDIETREGGIAMENTO

IPOTESI : $V_L = V_R = V_{\text{MIN}} = -100\%$

TESI : Il robot indietreggia in linea dritta

DIMOSTRAZIONE :

$$v = \frac{V_R + V_L}{2} = \frac{(-100) + (-100)}{2} = \frac{-200}{2} = -100$$

$$R = \frac{L}{2} \frac{V_R + V_L}{V_R - V_L} = \frac{L}{2} \frac{(-100) + (-100)}{(-100) - (-100)} = \frac{-L}{2} \frac{200}{0} = \infty$$

Si ha quindi un indietreggiamento alla massima velocità possibile con un raggio infinito; il raggio infinito può essere assimilato ad una retta e quindi il robot indietreggia in linea retta.

CASO 3) CURVA SULLA RUOTA

IPOTESI : $V_L = 0\%$ e $V_R = V_{\text{MAX}} = 100\%$

TESI : Il robot curva con un raggio pari a $L/2$

DIMOSTRAZIONE :

$$v = \frac{V_R + V_L}{2} = \frac{100 + 0}{2} = \frac{100}{2} = 50$$

$$R = \frac{L}{2} \frac{V_R + V_L}{V_R - V_L} = \frac{L}{2} \frac{100 + 0}{100 - 0} = \frac{L}{2} \frac{100}{100} = \frac{L}{2}$$

Il CIR è quindi spostato su una ruota (quella sinistra in questo caso) la quale rimane ferma mentre la destra ci gira intorno. Il raggio di curva per la ruota sinistra vale 0 mentre per quella destra vale L .

	<p style="text-align: center;">ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA</p>	
	<p style="text-align: center;">RELAZIONE TECNICA – PROGETTO DI FINE ANNO</p>	<p style="text-align: right;">Pagina 37 di 115</p>

CASO 4) CURVA SUL POSTO

IPOTESI : $V_L = V_{MIN} = -100\%$ e $V_R = V_{MAX} = 100\%$

TESI : Il robot ruota sul posto

DIMOSTRAZIONE :

$$v = \frac{V_R + V_L}{2} = \frac{100 + (-100)}{2} = \frac{0}{2} = 0$$

$$R = \frac{L}{2} \frac{V_R + V_L}{V_R - V_L} = \frac{L}{2} \frac{100 + (-100)}{100 - (-100)} = \frac{L}{2} \frac{0}{200} = 0$$

Il CIR coincide con il baricentro del robot e il robot ruota su se stesso senza spostarsi in nessuna direzione

	ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA	
	RELAZIONE TECNICA – PROGETTO DI FINE ANNO	Pagina 38 di 115

Pulse width modulation (PWM)

Prima di introdurre il concetto di comando di motori con PWM è necessario aprire una panoramica su questo argomento.

La modulazione ad ampiezza di impulso è un tipo di modulazione nella quale l'informazione è codificata sotto forma di durata del periodo positivo.

Il PWM è caratterizzato da:

- Frequenza costante
- Duty cycle variabile
- 2 possibili valori di ampiezza (nel nostro caso 0 e +5V)

i parametri del PWM sono quindi:

- Duty cycle
- Risoluzione

Il duty cycle è il rapporto percentuale tra il periodo nel quale il segnale è a V_{CC} e il periodo totale:

$$T = \frac{1}{f} = T_{ON} + T_{OFF}$$

$$\sigma = \frac{T_{ON}}{T} 100 = \frac{T_{ON}}{T_{ON} + T_{OFF}} 100$$

Un σ del 100% corrisponde ad tempo di ON pari al periodo (segnale continuo pari a V_{CC}) mentre un σ dello 0% corrisponde ad un tempo di ON pari a 0 (segnale continuo pari a 0). Tra tutti i valori intermedi è rilevante un σ del 50% poiché il tempo di ON e quello di OFF si equivalgono.

Idealmente σ può assumere qualsiasi valore mentre nella realtà si parla di risoluzione del PWM espressa in numero di bit; più è alta questa risoluzione maggiori saranno i valori di PWM possibili da raggiungere.

$$\text{Risoluzione} = \log_2(\text{valori possibili del PWM})$$

Nel nostro caso il PWM può assumere 24 valori diversi e la risoluzione necessaria risulta (per eccesso) 5 bit.

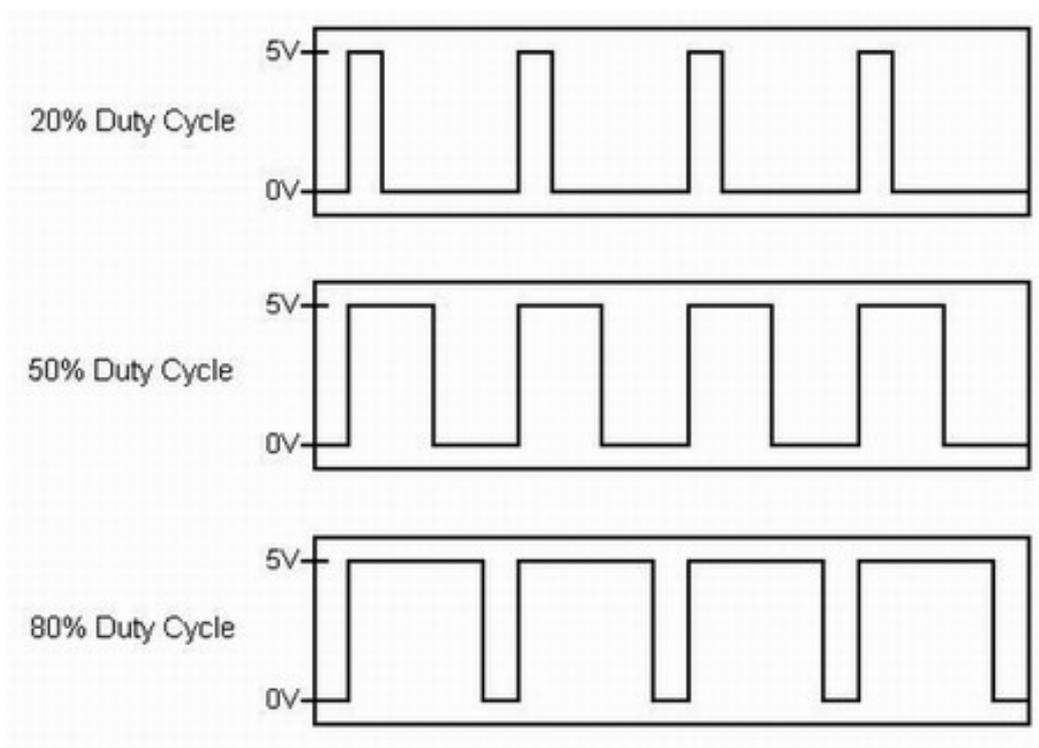


Illustrazione 21: Un esempio di tre modulazioni PWM

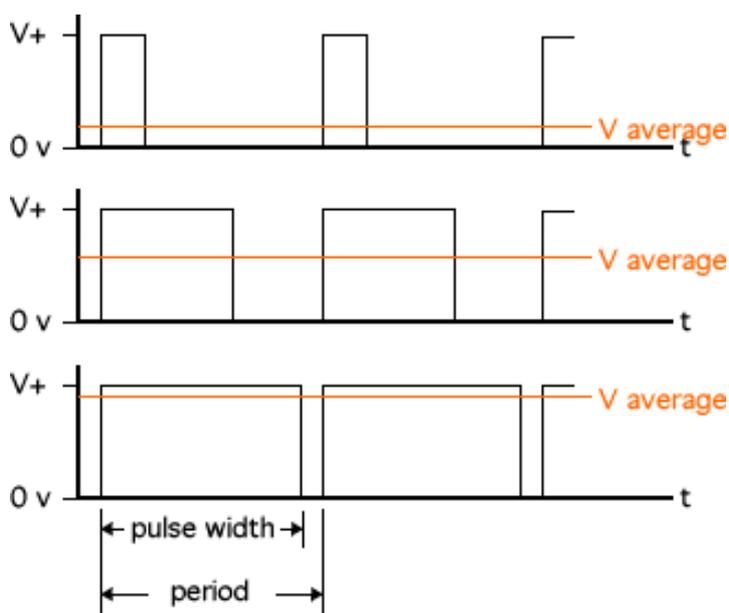


Illustrazione 22: V media in un segnale PWM

Le immagini presenti in questa pagina (Illustrazione 21 e 22) mostrano i tre segnali PWM maggiormente usati ovvero un segnale al 50% e due segnali mediati tra questo valore e gli estremi.

È molto importante la seconda illustrazione per comprendere il funzionamento del pilotaggio motori con la sign magnitude (SM).

Driver motori

Il microprocessore può erogare pochi mA per canale mentre i motori assorbono minimo 52mA; per ovviare a questo problema si deve inserire un driver in grado di pilotare un motore in base ai segnali ricevuti dal PIC.

I driver motori possono essere realizzati in due modi: con componenti discreti e in forma integrata; il primo metodo si usa quando le potenze in gioco sono tali da non permettere una corretta dissipazione con l'integrato; poiché i motori usati consumano relativamente poco la scelta è ricaduta su un diffusissimo driver: l'L293D.

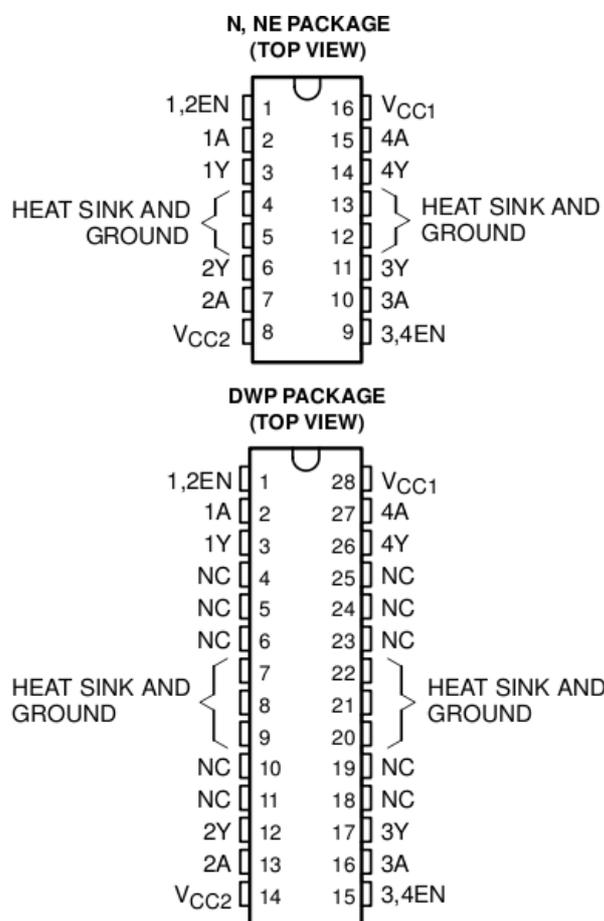


Illustrazione 23: L293D nelle due versioni commerciali

Il motore ovvero il motore può trovarsi solamente in tre stati: a tutta velocità in verso orario, a tutta velocità in verso antiorario o fermo. Per modificare la velocità si sfruttano i seguenti metodi (i quali utilizzano il segnale PWM precedentemente trattato) : locked anti-phase e sign magnitude.

Questo integrato contiene al suo interno due ponti ad H per comandare due carichi induttivi; ogni ponte ad H può essere inteso come un buffer di amplificazione per il motore.

Ogni buffer supporta fino a 600mA ovvero la corrente di stallo del motore e perciò è adeguato alle nostre richieste.

Per comandare ogni buffer sono necessari due segnali in contofase; questo sfasamento si può realizzare con una NOT poiché i comandi sono di tipo digitale. Se ai due ingressi vengono applicati gli stessi segnali (ovvero o 1-1 o 0-0) il motore non ruoterà e si danneggerà il driver.

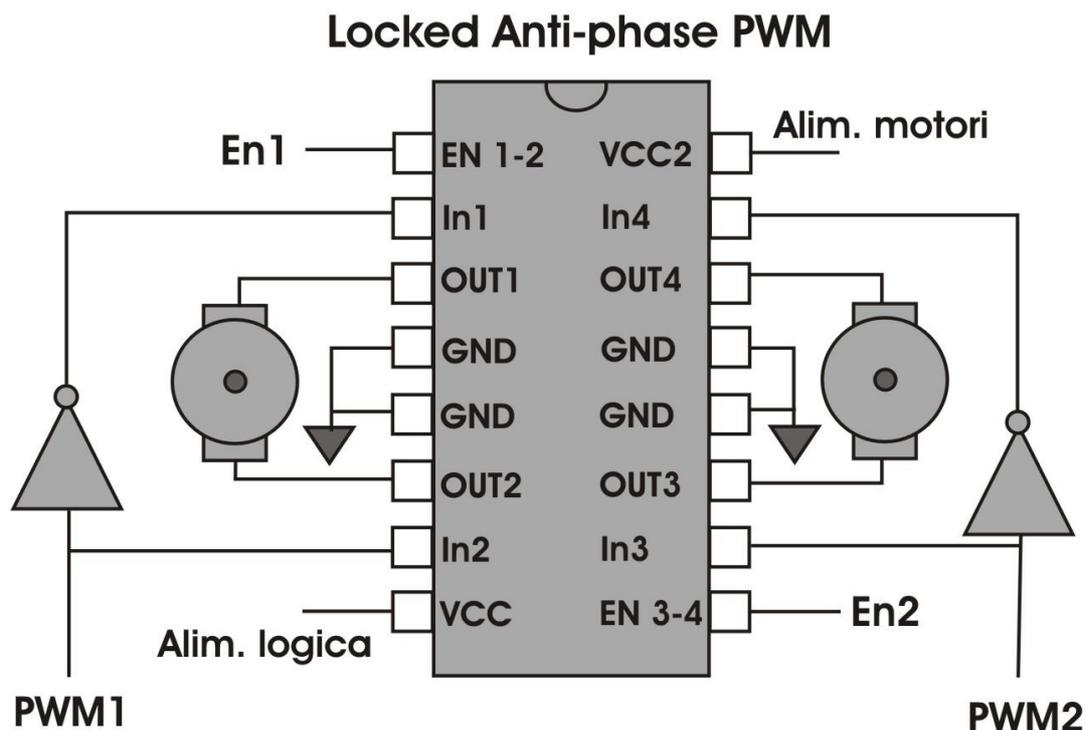
Ai due ingressi V_{cc1} e V_{cc2} vengono applicate rispettivamente la tensione logica e la tensione con la quale vengono pilotati i motori.

I piedini di enable, se attivati, permettono il corretto funzionamento del ponte e, viceversa, disabilitano le uscite.

Generalmente questo driver viene utilizzato con ingressi tradizionali i quali controllano verso e abilitazione del

	ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA	
	RELAZIONE TECNICA – PROGETTO DI FINE ANNO	Pagina 41 di 115

Locked anti-phase (LAP)



Schema 2: L293D in configurazione LAP

In questa configurazione le uscite sono sempre abilitate poiché il piedino di enable è posto a livello logico alto. Tutte le possibili varianti di movimento vengono fatte dal segnale PWM entrante nella NOT.

Come già detto, il segnale PWM è composto da due livelli (in questo caso 0 e +5V) i quali si alternano in base al duty cycle.

Applicando un PWM con duty cycle del 50% (ovvero un onda con un tempo di ON pari a quello di OFF) il motore gira in un verso per tutto il periodo di ON mentre gira nell'altro per tutto il periodo di OFF; poiché la frequenza del PWM è molto alta (intorno ai 10KHz) il motore continua a ruotare, per lo stesso tempo, da una parte e dall'altra con il risultato che il movimento apprezzabile è nullo.

Applicando un segnale PWM con duty cycle maggiore del 50%, per lo stesso motivo già spiegato, il motore ruoterà in un verso con velocità maggiore all'aumentare del PWM raggiungendo la velocità massima al 100%.

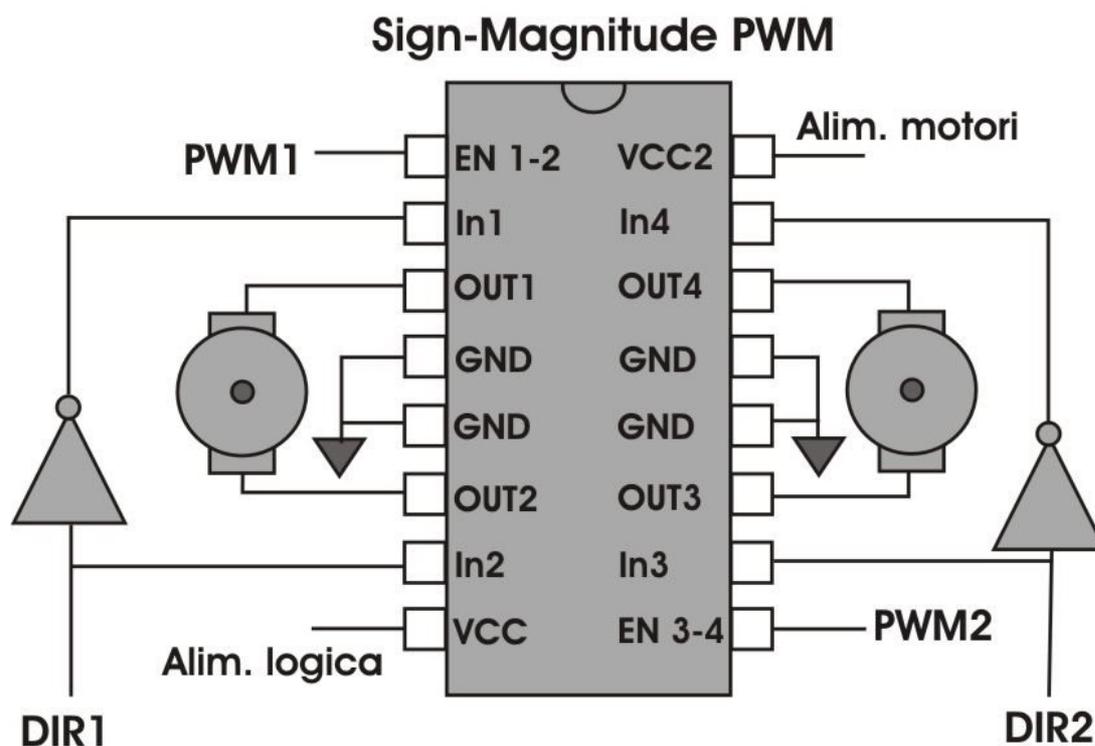
Applicando un segnale PWM con duty cycle minore del 50% il motore ruoterà nel verso opposto

con velocità maggiore all'aumentare del PWM raggiungendo la velocità massima al 0%.

Questo controllo, molto utilizzato, presenta le seguenti caratteristiche:

- È necessario solo un segnale per poter variare la velocità e la direzione.
- L'arresto del motore è immediato.
- La partenza del motore è più reattiva.
- Consuma anche da fermo.
- La potenza del motore viene assorbita dalle commutazioni

Sign magnitude (SM)



Schema 3: L293D in configurazione SM

In questa configurazione gli ingressi del motore sono comandati da un segnale tradizionale (ovvero continuo a commutazioni molto rare) mentre il segnale PWM è applicato sui piedini di enable.

Tutto ciò significa che il motore ruota liberamente solo nel tempo di ON del PWM e si ferma nel

	<p style="text-align: center;">ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA</p>	
	<p style="text-align: center;">RELAZIONE TECNICA – PROGETTO DI FINE ANNO</p>	<p style="text-align: right;">Pagina 43 di 115</p>

tempo di OFF; in realtà non è un vero e proprio arresto bensì è un girare per inerzia senza ulteriori spinte.

Se il PWM ha un duty cycle del 100% questo si traduce nella massima velocità possibile nella direzione scelta.

Se il PWM ha un duty cycle dello 0% il motore non si muove a prescindere dalla direzione applicatagli.

Tutti i valori intermedi corrispondono ad una variazione di velocità dal minimo (0%) al massimo (100%).

Questo sistema è leggermente in disuso e presenta le seguenti caratteristiche:

- Sono necessari due segnali (tradizionale e PWM) per comandare ogni motore.
- Il motore è libero quindi partenza e arresto sono più lenti della configurazione LAP.
- La variazione di velocità è più precisa poiché la stessa risoluzione di PWM si applica solo ad un verso mentre nella LAP si applica a due versi.
- Il motore non consuma da fermo.
- Le commutazioni non assorbono potenza (poiché non vi sono commutazioni causate dal PWM).

	<p style="text-align: center;">ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA</p>	
	<p style="text-align: center;">RELAZIONE TECNICA – PROGETTO DI FINE ANNO</p>	<p style="text-align: right;">Pagina 44 di 115</p>

Microprocessore

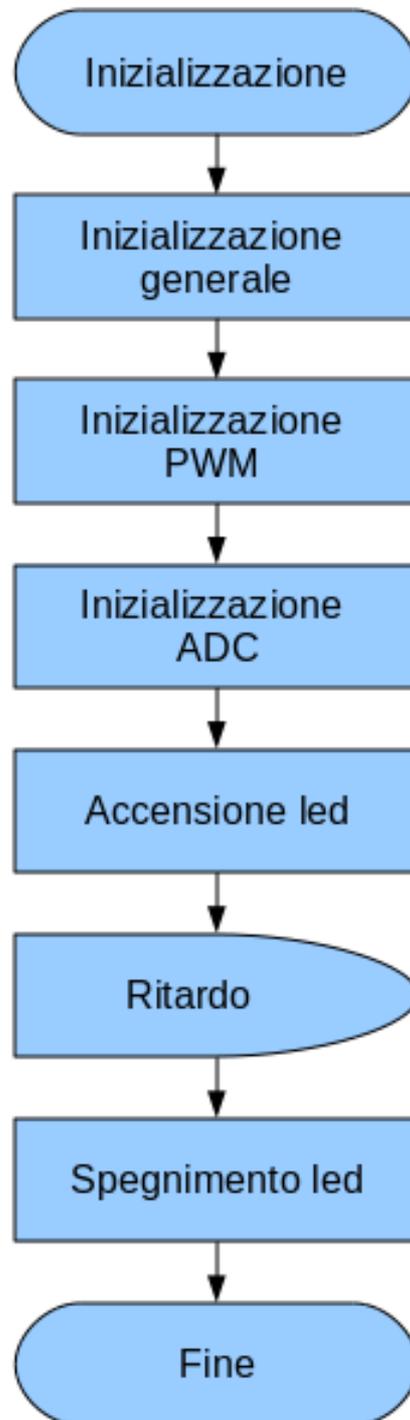
Programma

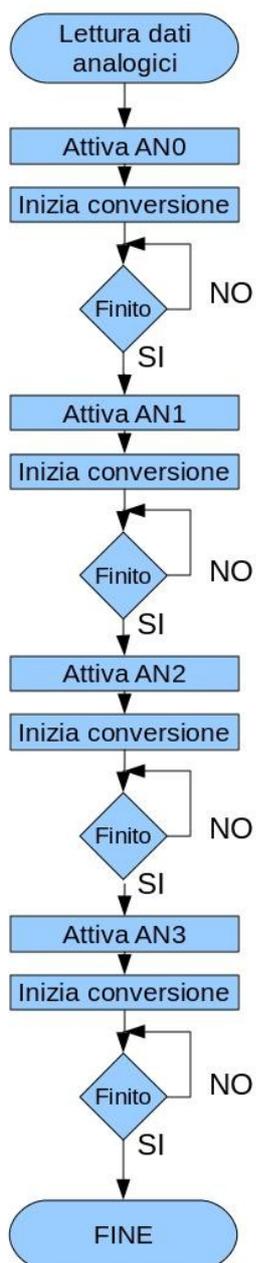
Il programma del microprocessore (Allegato 1 a questa relazione) è stato scritto in Assembly ed è il risultato di 19 versioni (attualmente l'ultima versione stabile è la 1.9 mentre la prima versione è stata la 0.1), conta all'incirca 1500 righe di codice .

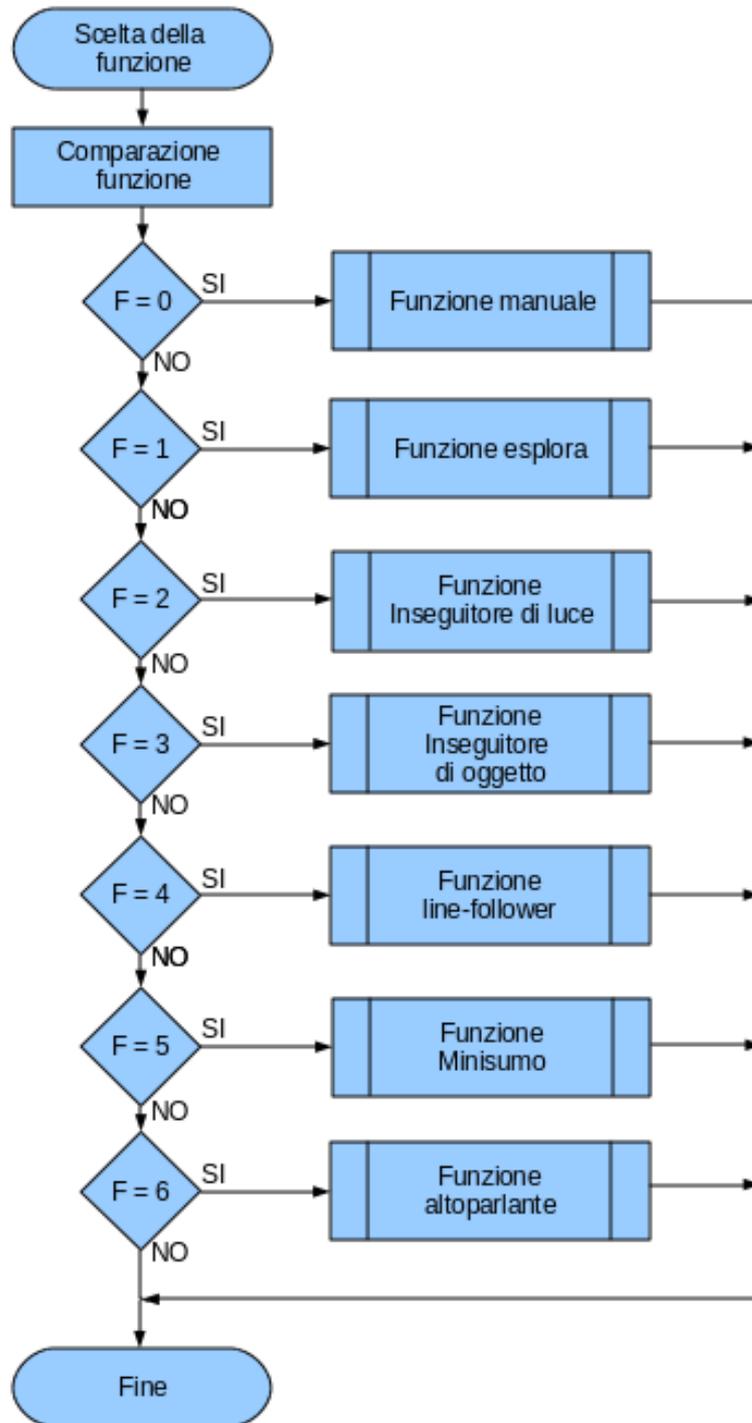
Invece di analizzare il programma passo per passo è stato preferito inserire i flow chart delle funzioni limitando così lo spazio occupato dal programma su questo testo.

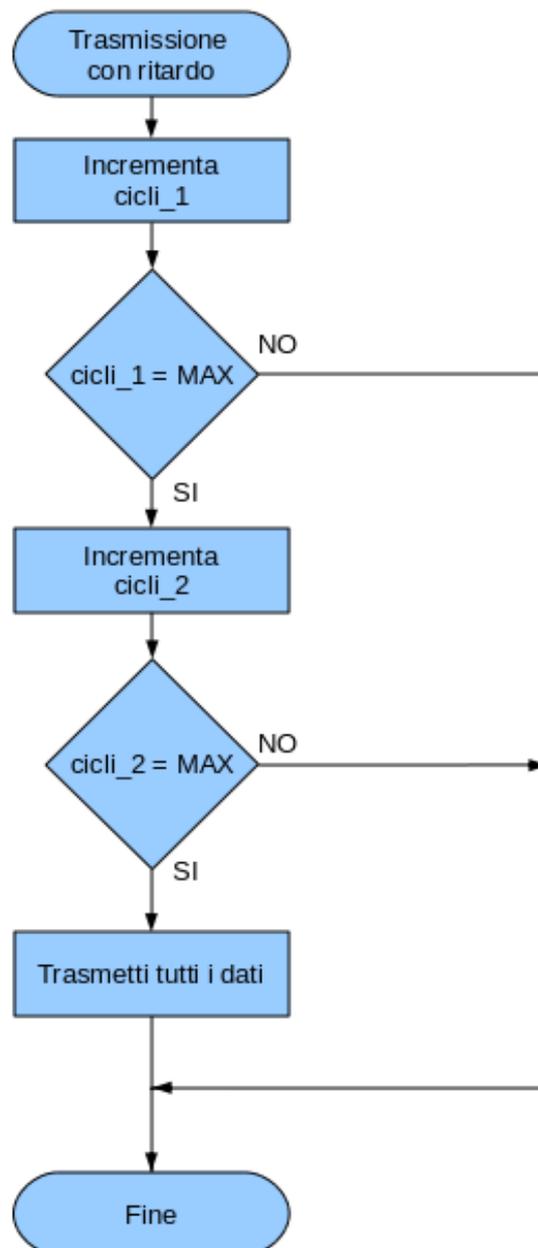
Per poter rendere la lettura dei flow chart più sintetica ed efficace, il diagramma principale (main) è stato diviso in sottosezioni divise per funzioni.

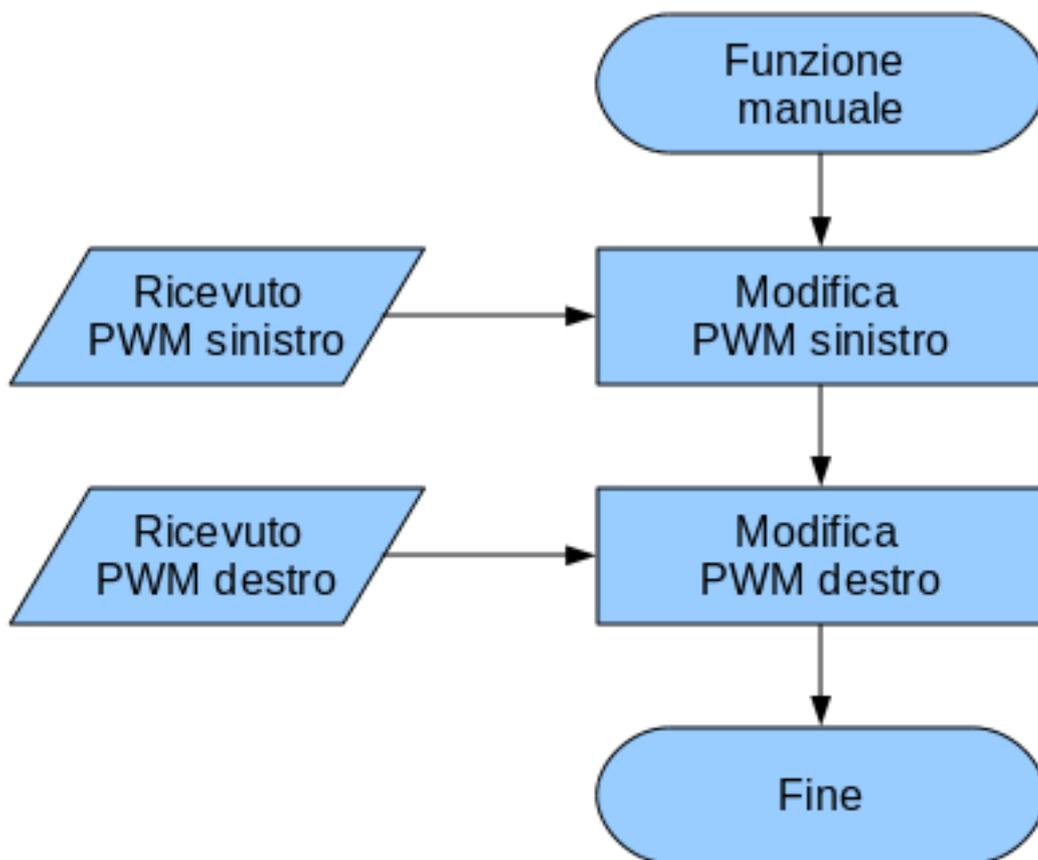


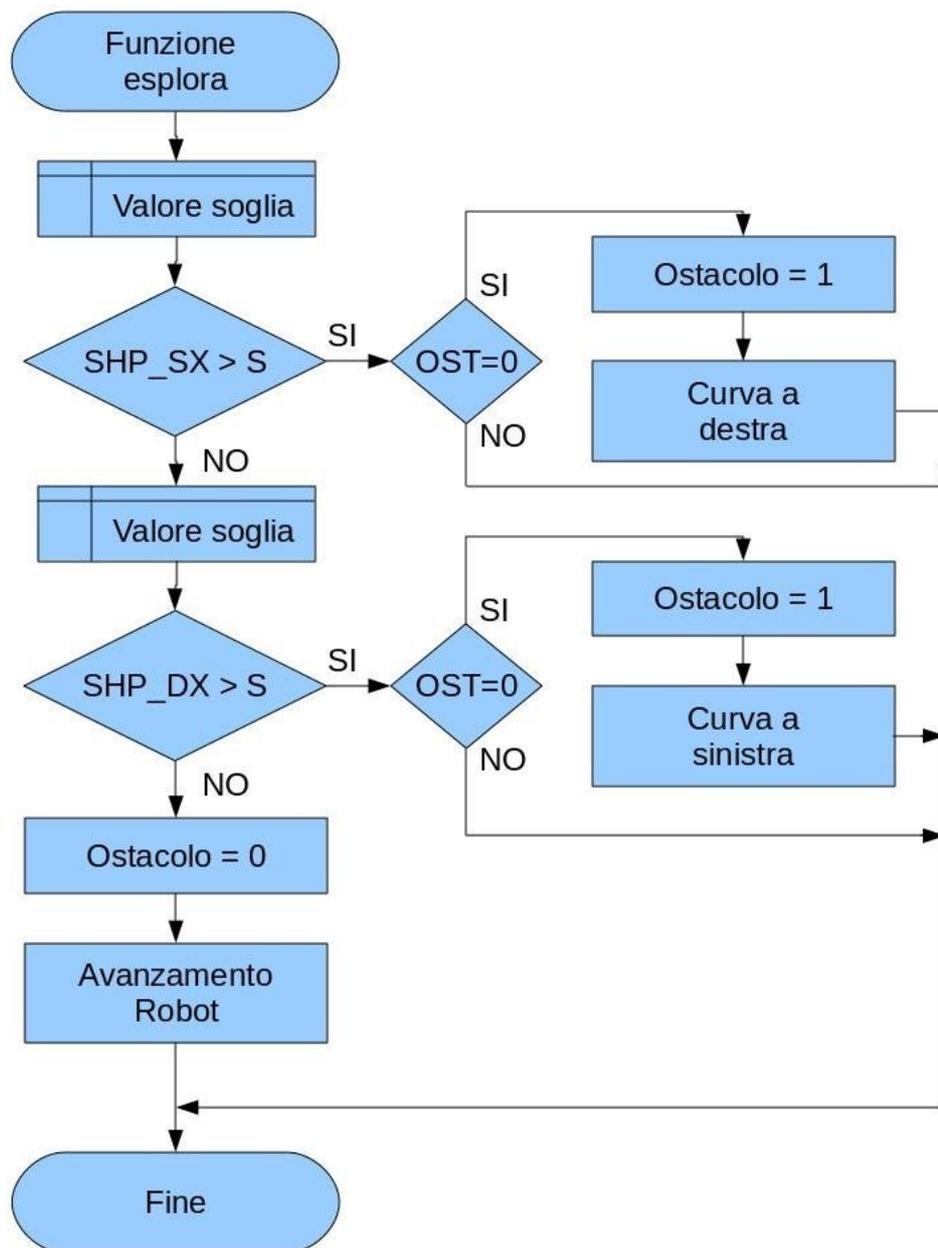


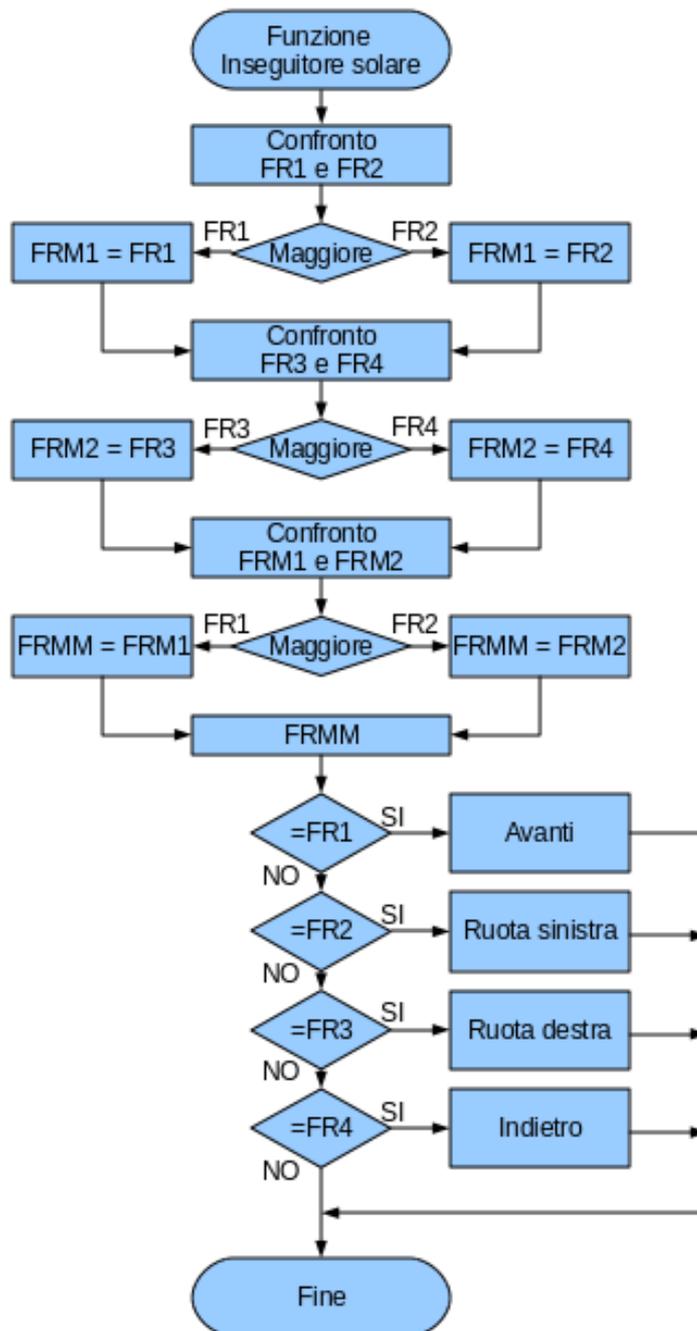


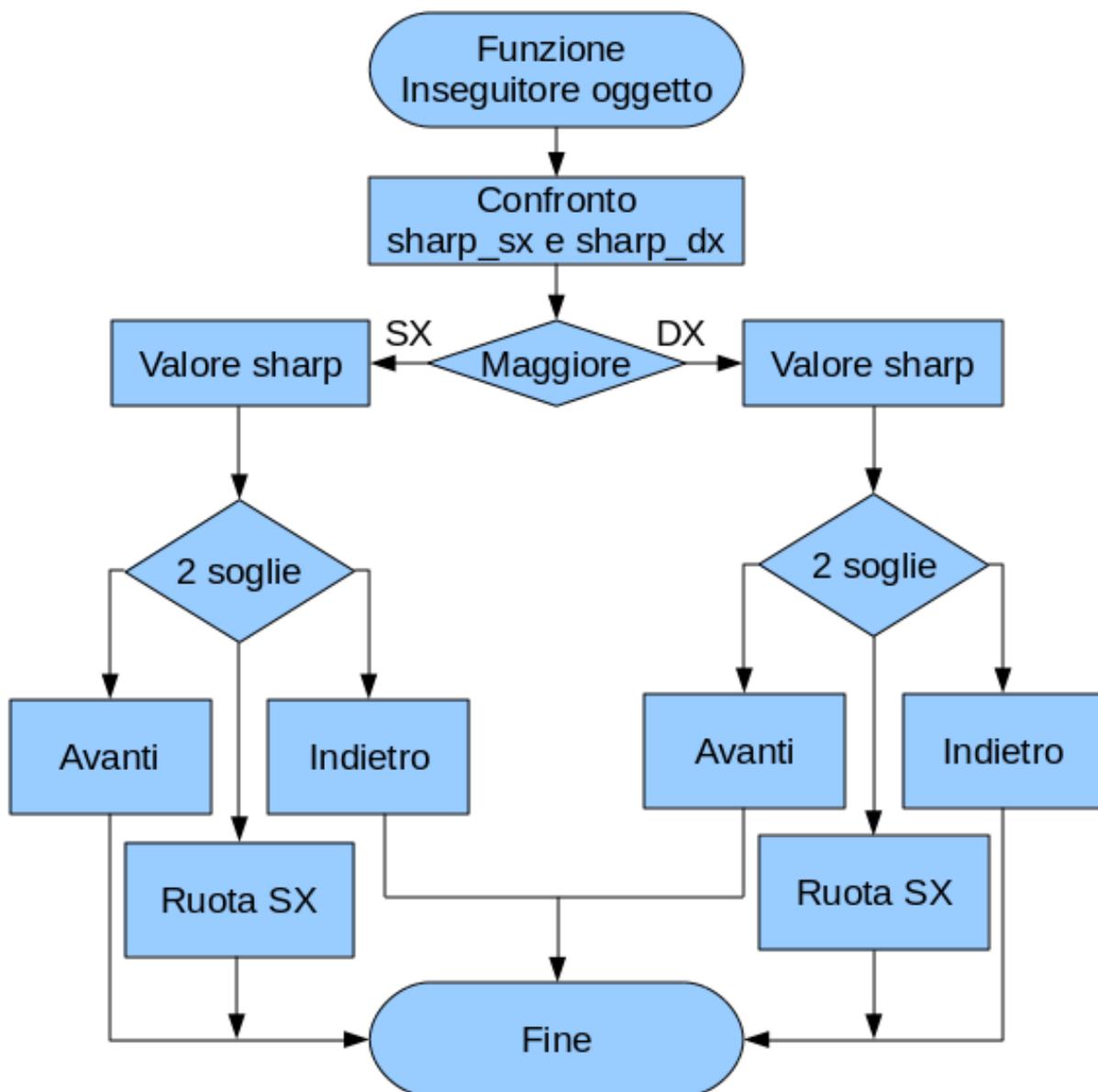


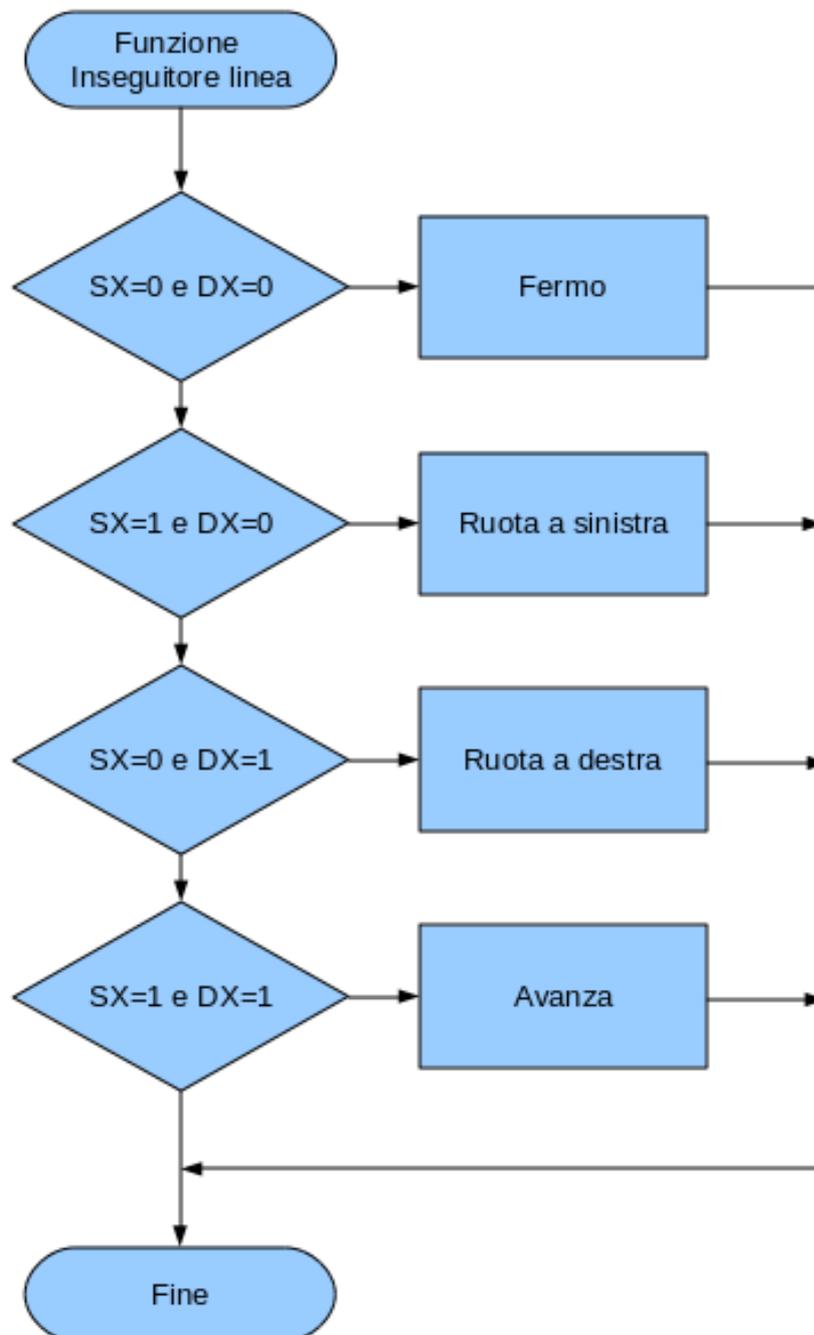


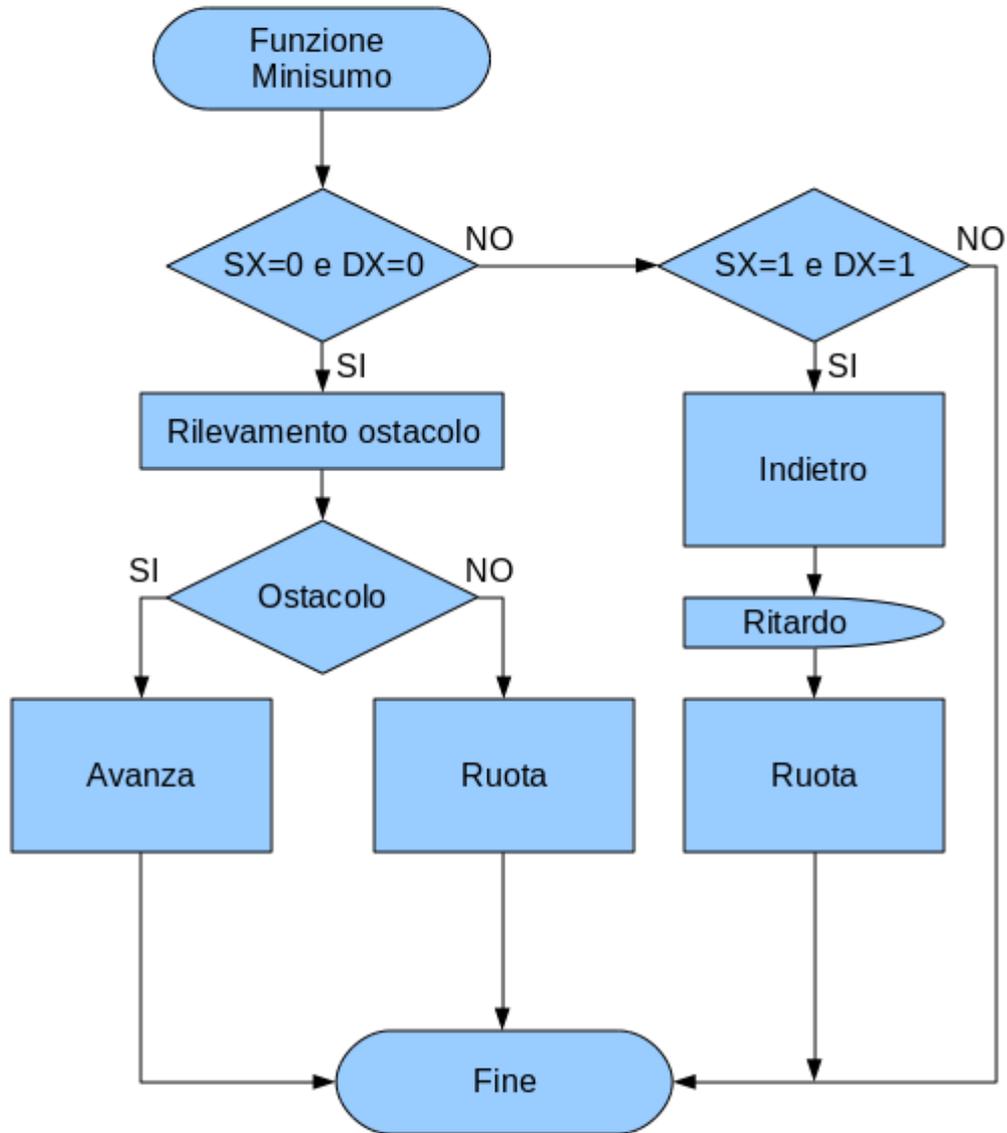












	<p style="text-align: center;">ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA</p>	
	<p style="text-align: center;">RELAZIONE TECNICA – PROGETTO DI FINE ANNO</p>	<p style="text-align: right;">Pagina 55 di 115</p>

Comunicazione robot-PC

Introduzione

Dopo la parte meccanica e la gestione del movimento, la comunicazione robot-pc è sicuramente la parte più importante.

In questa comunicazione, il compito del robot è quello di seguire le istruzioni provenienti dal PC e di inviare tutti i dati raccolti dai sensori; dall'altra parte il PC si occupa di inviare le istruzioni desiderate e di ricevere e gestire il flusso di dati (periodici e scanditi, gli impulsi provengono dal PIC ad una determinata frequenza).

Il sistema si compone così di tre elementi principali: il robot (e quindi il sistema che permette al robot di comunicare), il mezzo trasmissivo (via cavo o wireless) e il PC (e in particolare il programma scritto in Delphi 7 in esecuzione sul computer).

EIA RS-232

La comunicazione è di tipo seriale e sfrutta il protocollo Electronic Industries Alliance Recommended Standard 232, ovvero il più conosciuto EIA RS-232 (equivalente allo standard europeo CCITT V21/V24) il quale definisce un'interfaccia seriale a bassa velocità per lo scambio di dati tra dispositivi digitali.

Storia dello standard

L'interfaccia seriale EIA RS-232 è uno standard costituito da una serie di protocolli meccanici, elettrici ed informatici che rendono possibile lo scambio di informazioni a bassa velocità tra dispositivi digitali. Esso includeva le caratteristiche elettriche dei segnali, la struttura e le temporizzazioni dei dati seriali, la definizione dei segnali e dei protocolli per il controllo del flusso di dati seriali su un canale telefonico, il connettore e la disposizione dei suoi pin ed infine il tipo e la lunghezza massima dei possibili cavi di collegamento.

Nel corso di oltre 40 anni lo standard si è evoluto, pur mantenendosi in larga parte invariato. L'evoluzione è riconoscibile dalla sigla, leggendo l'ultima lettera; l'ultima revisione è del 1997 ed è indicata come EIA RS-232f. Probabilmente la versione più diffusa è la RS232c, del 1969, che corrisponde alle specifiche europee CCITT, raccomandazione V.24. Pur essendo un protocollo piuttosto vecchio, attualmente la EIA RS-232 è ancora largamente utilizzata per la comunicazione a bassa velocità tra microcontrollori, dispositivi industriali ed altri circuiti relativamente semplici, che non necessitano di particolare velocità; è invece praticamente scomparsa in ambito "desktop", nel quale lo standard è nato per la comunicazione tra un computer ed un modem.

	ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA	
	RELAZIONE TECNICA – PROGETTO DI FINE ANNO	Pagina 56 di 115

Caratteristiche elettriche

Le caratteristiche di natura elettrica previste dallo standard consentono di acquisire le poche informazioni necessarie a progettare dispositivi elettronici che comunicano con un PC attraverso questa porta. I segnali che vengono supportati dal dispositivo sono di tre tipi:

- MARK: Compresi tra -3 e -15 V;
- SPACE: Compresi tra +3 e +15 V;
- INCERTEZZA: Compresi tra -3 e +3 V;

Questi tipi di segnali hanno la stessa logica per quasi tutti i circuiti del dispositivo ma posso avere logica inversa per alcuni tipi di circuito.

Definizione

L'interfaccia EIA RS-232 ridotta (ovvero solo asincrona) utilizza un protocollo di trasmissione seriale di tipo asincrono.

Seriale significa che i bit che costituiscono l'informazione sono trasmessi uno alla volta su di un solo "filo". Questo termine è in genere contrapposto a "parallelo": in questo caso i dati sono trasmessi contemporaneamente su più fili, per esempio 8, 16 o 32.

Parlando astrattamente si potrebbe pensare che la trasmissione seriale sia intrinsecamente più lenta di quella parallela (su di un filo possono passare meno informazioni che su 16). In realtà questo non è vero in assoluto, soprattutto a causa della difficoltà di controllare lo skew (disallineamento temporale tra i vari segnali) dei molti trasmettitori in un bus parallelo, e dipende dalle tecnologie adottate.

Asincrono significa, in questo contesto, che i dati sono trasmessi, byte per byte, in modo anche non consecutivo e senza l'aggiunta di un segnale di clock, cioè di un segnale comune che permette di sincronizzare la trasmissione con la ricezione; ovviamente sia il trasmettitore che il ricevitore devono comunque essere dotati di un clock locale per poter interpretare i dati. La sincronizzazione dei due clock è necessaria ed è fatta in corrispondenza della prima transizione sulla linea dei dati.

Le unità di misurazione

Le unità di misura della velocità di trasmissione sono essenzialmente due: il baud (simboli al secondo) ed il bit per secondo (bps o, più raramente, b/s), spesso trattate erroneamente come sinonimi.

La baud rate indica il numero di transizioni al secondo che avvengono sulla linea; il bps indica, come dice il nome, quanti bit al secondo sono trasmessi lungo la linea.

Nel caso di trasmissione binaria (cioè in cui è presente un livello alto ed uno basso) le due cose ovviamente coincidono numericamente, da cui la parziale equivalenza dei due termini. Nel caso di trasmissioni a più livelli, invece, è possibile trasmettere con una sola transizione più bit: se per

	<p style="text-align: center;">ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA</p>	
	<p style="text-align: center;">RELAZIONE TECNICA – PROGETTO DI FINE ANNO</p>	<p style="text-align: right;">Pagina 57 di 115</p>

esempio posso trasmettere otto diversi valori di tensione tra 0 e 7 V, con un solo valore di tensione invio tre bit (0 V = 000, 1 V = 001, 2 V = 010...) ed in questo caso una trasmissione a 1000 baud equivale ad una a 3000 bps.

Nel caso dello standard EIA RS-232 i livelli utilizzati sono due, quindi il baud rate coincide numericamente con il bps.

Descrizione del segnale

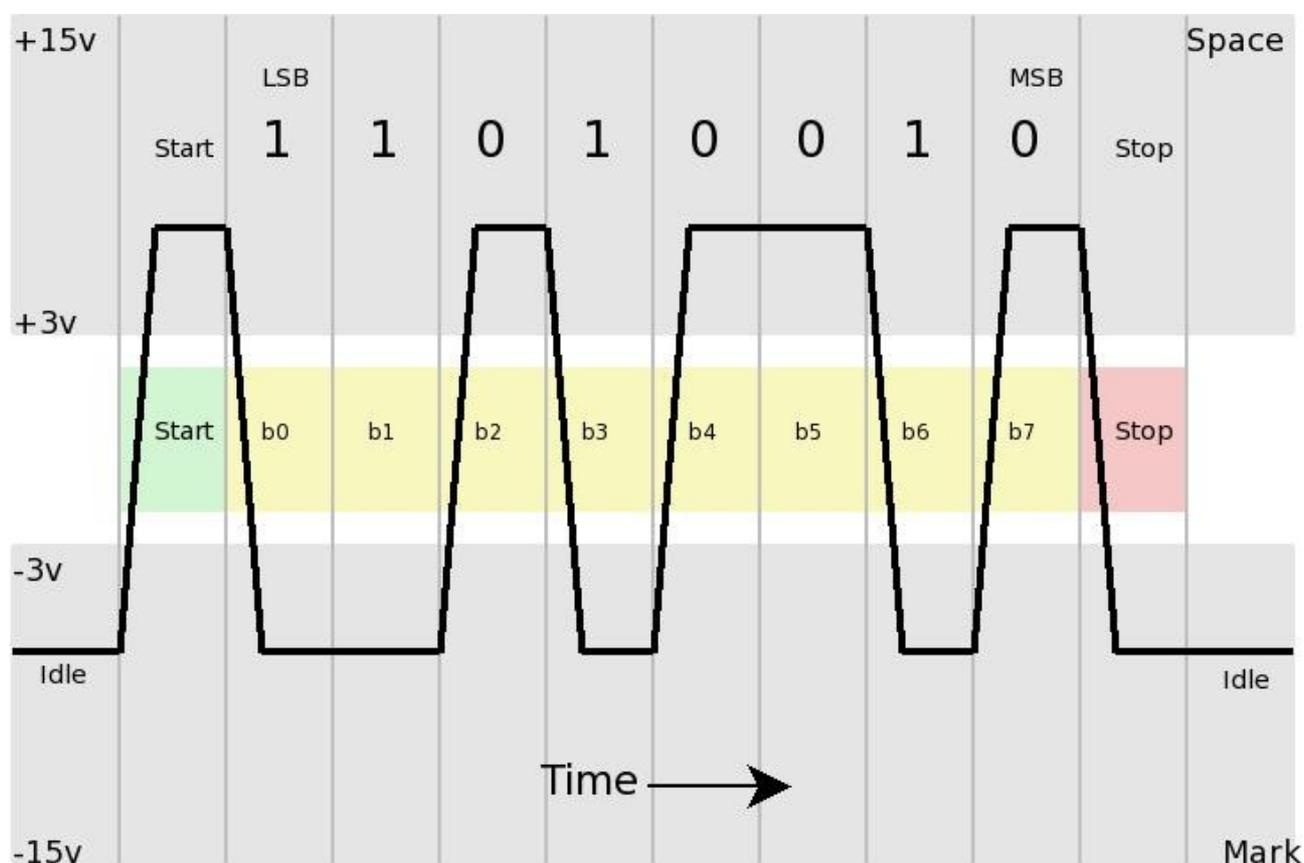


Illustrazione 24: Esempio di pacchetto in trasmissione RS-232

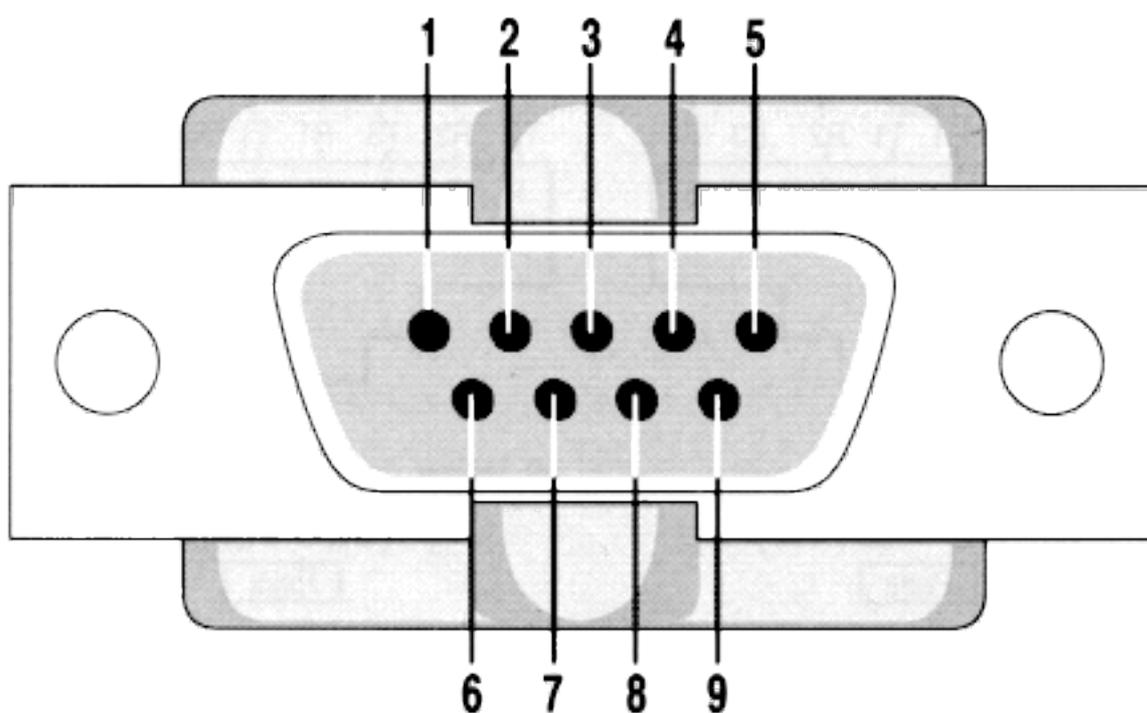
Tutte le transizioni appaiono in corrispondenza di multipli di 104 μ s (pari ad 1/9600 s, cioè ciascun bit dura esattamente l'inverso del baud rate).

La linea si trova inizialmente nello stato di riposo, alta (nessun dato in transito); la prima transizione da alto a basso indica l'inizio della trasmissione (inizia il "bit di start", lungo esattamente 104 μ s), a cui segue il bit meno significativo (LSB), dopo altri 104 μ s, e così via, per otto volte, fino al bit più significativo (MSB).

Da notare che il byte è trasmesso "al contrario", cioè va letto da destra verso sinistra. Segue infine un periodo di riposo della linea di almeno 104 μ s, cioè un bit di stop e quindi (eventualmente) inizia un nuovo pacchetto di bit.

Il tipo di trasmissione viene espresso da una sigla: la prima cifra indica quanti bit di dati sono trasmessi (nel nostro caso 8) ; la prima lettera indica il tipo di parità (rispettivamente nessuna ed even-parity, cioè parità pari, nel nostro caso nessuna quindi n); la seconda cifra indica il numero di bit di stop (solitamente 1,2 o 1.5; nel nostro caso 1).

La nostra trasmissione può essere espressa con questa sigla: 8n1.



Pin	Signal	Pin	Signal
1	Data Carrier Detect	6	Data Set Ready
2	Received Data	7	Request to Send
3	Transmitted Data	8	Clear to Send
4	Data Terminal Ready	9	Ring Indicator
5	Signal Ground		

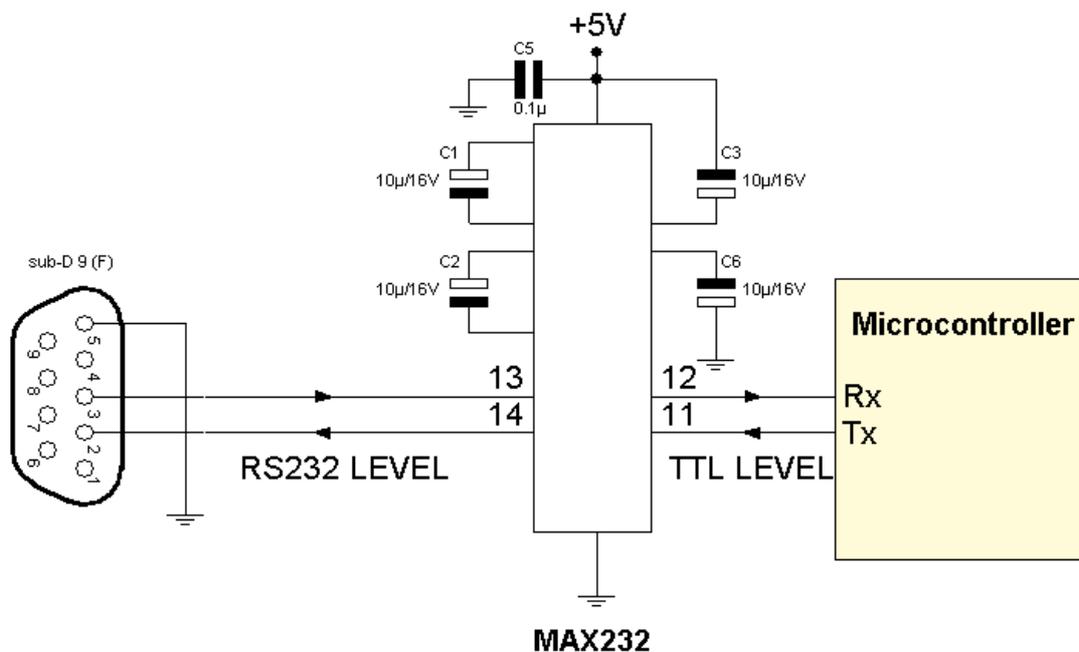
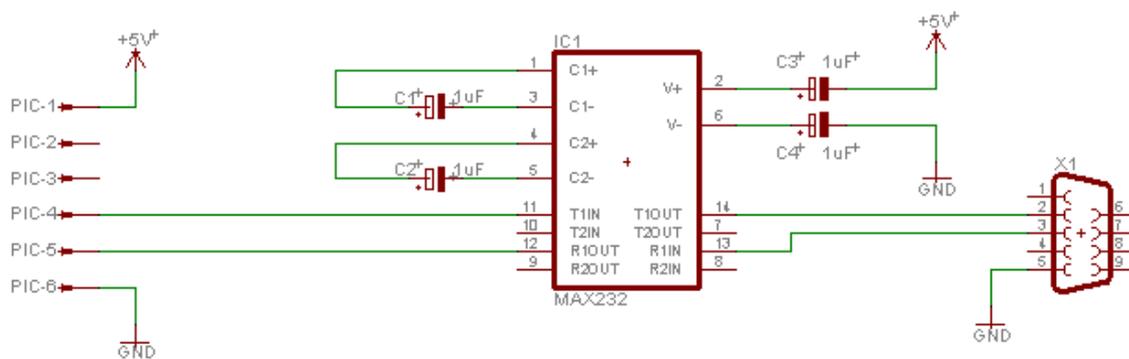
Illustrazione 25: La piedinatura di un connettore femmina

MAX232

Il MAX232 è un integrato prodotto dalla Maxim altamente diffuso in molte comunicazioni seriali.

Il MAX funge da adattatore di livelli tra i livelli della porta seriale del computer ($\pm 15V$) e i classici livelli TTL (0-5V) usati da moltissimi integrati e microprocessori.

Il primo schema (vedere Allegato 8) è lo schematico realizzato sul robot mentre il secondo schema mostra il tipico collegamento PC-microcontrollore.



Comunicazione wireless



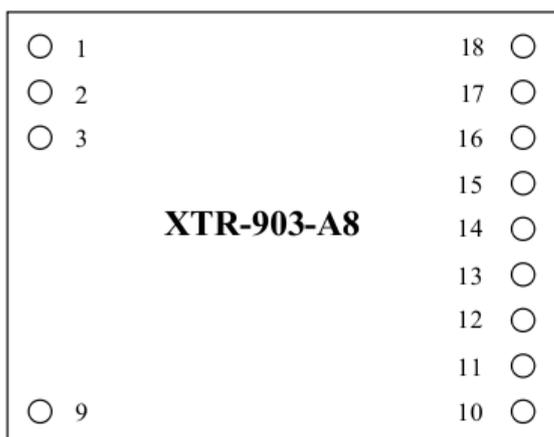
Illustrazione 26: Modulo XTR 903

contro gli 868MHz del modello A8).

Questi moduli incorporano tutta l'elettronica necessaria per controllare, inviare e ricevere mentre per trasmettere è necessario applicare il segnale al piedino corretto.

Per permettere al robot di svolgere le proprie funzioni senza l'ingombro del cavo seriale è stata implementata una comunicazione wireless (ovvero senza fili) su portante analogica a frequenza 433 MHz.

Per semplificare la gestione della trasmissione sono stati utilizzati i moduli ricetrasmittenti della AUREL modello XTR-903 A4 (A4 sta per portante a 433MHz



Pin No.	Nome Pin
Pin 1-3	RF GND
Pin 2	ANT
Pin 9-10-18	GND
Pin 11	SP1
Pin 12	RSRX
Pin 13	485EN
Pin 14	RSTX
Pin 15	SP2
Pin 16	PWRDN
Pin 17	Vcc

Illustrazione 27: Piedinatura del modulo

	ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA	
	RELAZIONE TECNICA – PROGETTO DI FINE ANNO	Pagina 61 di 115

	Min.	Tip.	Max.	Units
DC values				
Supply Voltage	2.7	3	3.3	V
(rx mode) Current		26		mA
(tx mode @ -8 dBm) Current		20		mA
(tx mode @ 10 dBm) Current		31		mA
(stand-by mode) Current		8	10	μA
Input/output Logical 1 Level	0.7xVcc		Vcc	V
Input/output Logical 0 Level	0		0.3xVcc	V
RF				
Modulation		FSK		
Rx Sensitivity		-105		dBm
Tx Max Power Output	-8		10	dBm
Performance				
Input Bit Rate ¹	9600, 19200 e 38400			bps
Outdoor range		200		m
Available Channels			7	
Timing				
PWRDN → RX		7		ms
PWRDN → TX		5		ms
TX → RX		3		ms
RX → TX		3		ms
CH #X → CH #Y		3		ms
Default Values (NO programming)²				
Channel frequency		868.88		MHz
Tx Output Power		4		dBm

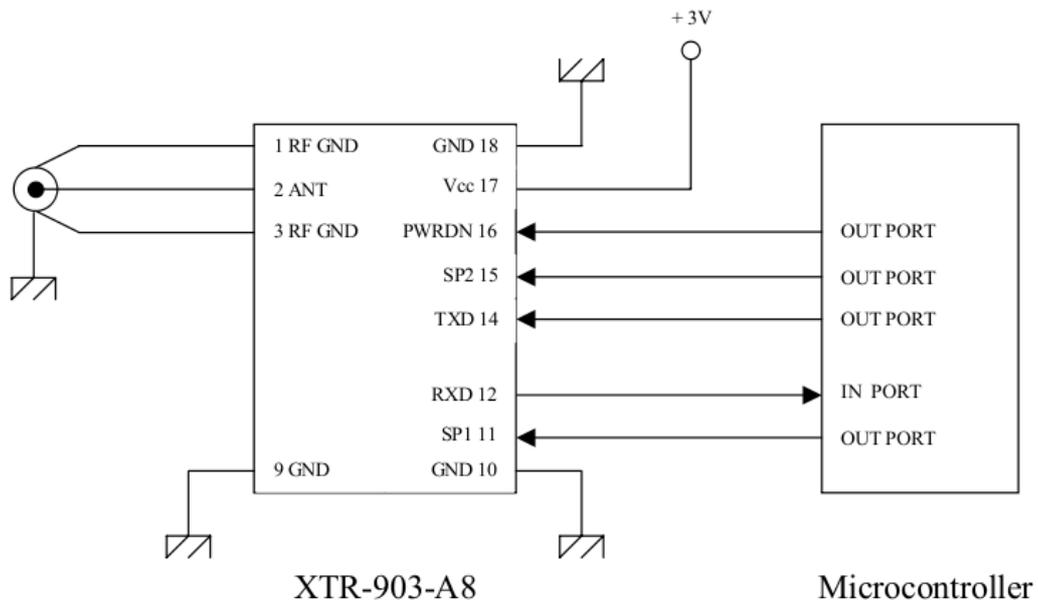


Fig.3 – Example of connection between XTR-903-A8 and microcontroller.

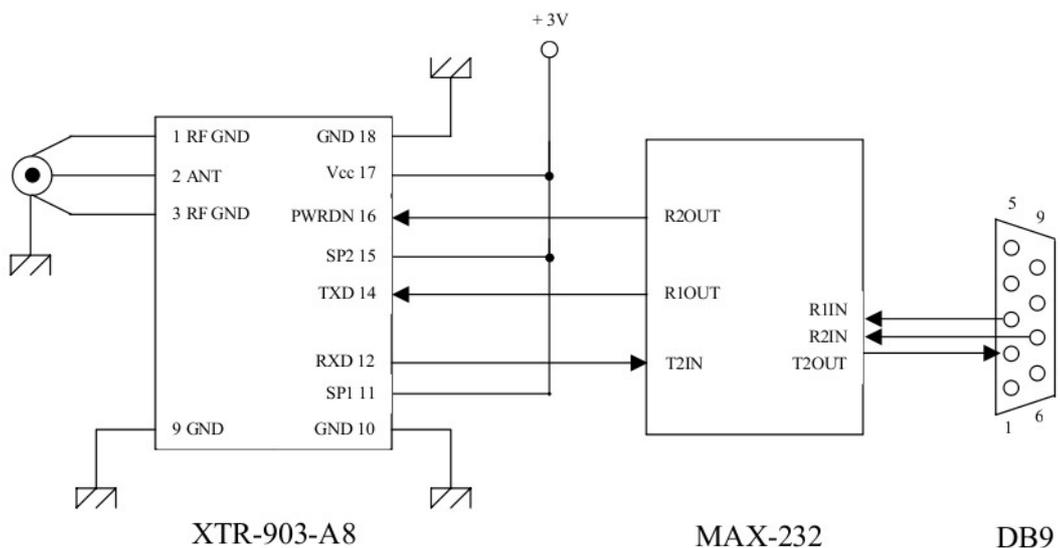


Fig.4 – Example of connection XTR-903-A8 and RS-232 serial port at 19200 bps.

Comunicazione – Lato microprocessore

Parte hardware

La comunicazione avviene attraverso un USART (Universal Synchronous-Asynchronous Receiver-Transmitter) ovvero un dispositivo hardware dedicato in grado di convertire flussi di dati seriali in formato parallelo (ricezione) o viceversa (trasmissione).

Questo dispositivo è molto complesso ed è completamente integrato nel microprocessore; le uniche opzioni di modifica riguardano la sua configurazione e la scansione del funzionamento.

I seguenti schemi a blocchi delle due parti che compongono l'USART (ricezione e trasmissione) aiutano a comprenderne il funzionamento.

FIGURE 12-2: EUSART RECEIVE BLOCK DIAGRAM

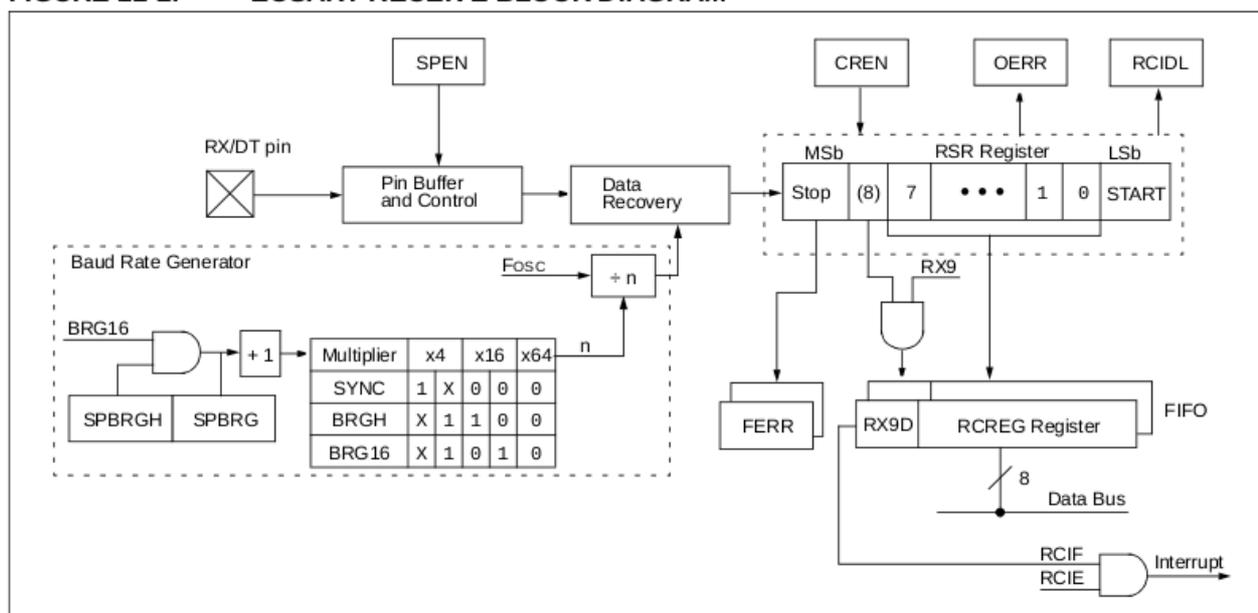
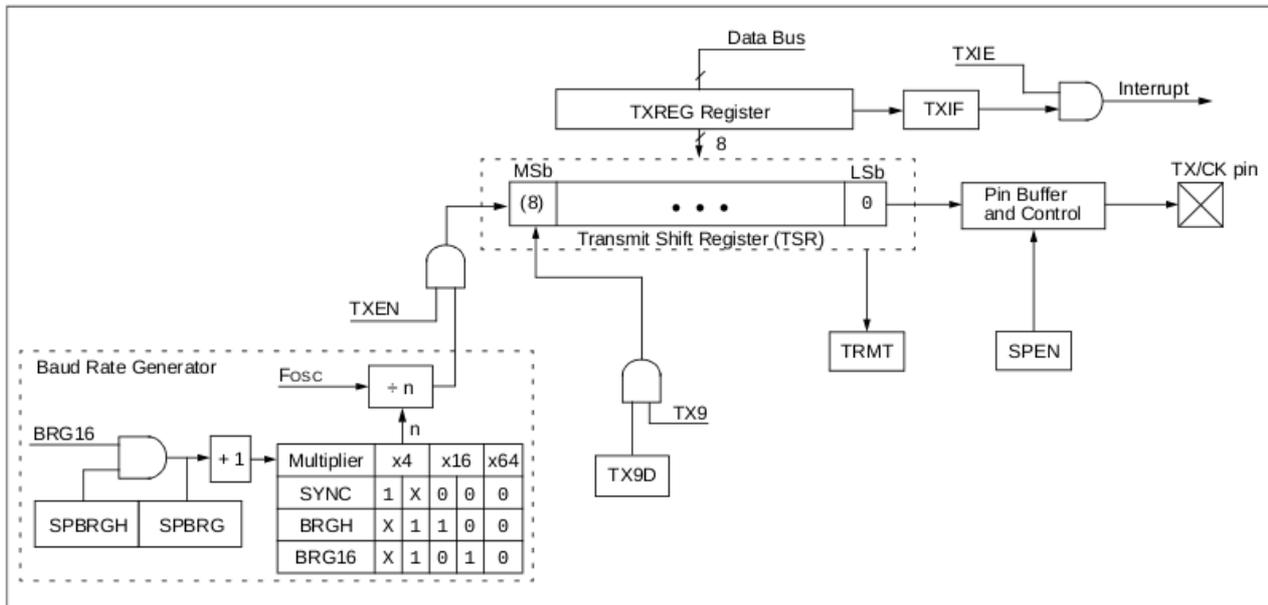


FIGURE 12-1: EUSART TRANSMIT BLOCK DIAGRAM



Parte software

La parte software, oltre ad occuparsi della configurazione iniziale, si occupa della tempistica e della gestione dei dati ricevuti/inviati.

```

;-----
;          FUNZIONI DI INTERRUPT
;-----

irq_exec
    movwf    save_w           ; Salva W
    swapf   STATUS,W
    clrf    STATUS
    movwf   save_stat       ; Salva lo STATUS
    movfw   FSR
    movwf   save_fsr       ; Salva lo STATUS
    movfw   PCLATH
    movwf   save_pclath    ; Salva l'indirizzo di pagina
    clrf    PCLATH
    clrwdt

    banksel RAM_BANK0
    
```

Quando l'USART riceve un dato, si attiva un interrupt. Un interrupt è, come dice il nome stesso, un'interruzione del normale svolgimento del programma, finita l'esecuzione del programma dell'interrupt il programma torna al punto che stava eseguendo prima dell'interruzione. La prima

	ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA	
	RELAZIONE TECNICA – PROGETTO DI FINE ANNO	Pagina 65 di 115

parte serve quindi a salvare la posizione, i dati, le variabili di stato ed ogni informazione necessaria per tornare, a fine interrupt, al programma principale.

```

irq_rx
    btfsc    PIR1,RCIF
    goto    irq_rx_data    ; IRQ RX

irq_tmr
    btfsc    INTCON,T0IF
    goto    irq_time      ; IRQ TMR0
    goto    irq_ret

;-----
irq_rx_data
    bcf     PIR1,RCIF      ; Clear flag RX-IRQ
    movfw   p_buff_rx
    movwf   FSR
    movfw   RCREG         ; Riceve

irq_rx1  ;movwf   INDF
  
```

Poiché un interrupt può essere chiamata da diverse funzioni è necessario verificare quale di queste ha realmente interrotto il programma; la verifica viene fatta su dei particolari bit chiamati flag i quali si attivano prima della chiamata.

Dopo essere entrato nella sezione di ricezione salva il byte ricevuto nel registro W e attende che quest'ultimo venga gestito.

```

;-----
;          INIZIO SALVATAGGIO DATI RICEVUTI
;-----

    ;movwf   TXREG        ; Reinvia il dato ricevuto
    ;movwf   PORTB       ; Mostra il dato sul PORTB
    movwf   buffer_rx
    movfw   buffer_index_rx
    movwf   index_rx

    ; Ogni blocco verifica a che byte è arrivato e lo salva
    ;nell'apposita variabile

    movlw   B'00000000'
    bcf    STATUS,Z
    xorwf  index_rx,W
    btfsc  STATUS,Z
    goto   primo_byte
  
```



```
    movfw    buffer_index_rx
    movwf    index_rx

    movlw    B'00000001'
    bcf      STATUS,Z
    xorwf    index_rx,W
    btfsc    STATUS,Z
    goto     secondo_byte

    movfw    buffer_index_rx
    movwf    index_rx

    movlw    B'00000010'
    bcf      STATUS,Z
    xorwf    index_rx,W
    btfsc    STATUS,Z
    goto     terzo_byte

    movfw    buffer_index_rx
    movwf    index_rx

    movlw    B'00000011'
    bcf      STATUS,Z
    xorwf    index_rx,W
    btfsc    STATUS,Z
    goto     quarto_byte

primo_byte
    movfw    buffer_rx
    movwf    pwm_sx_rx
    goto     end_buffer_rx

secondo_byte
    movfw    buffer_rx
    movwf    pwm_dx_rx
    goto     end_buffer_rx

terzo_byte
    movfw    buffer_rx
    movwf    funzione_rx
    goto     end_buffer_rx

quarto_byte
    movfw    buffer_rx
    movwf    note_rx
    goto     end_buffer_rx

end_buffer_rx
```

	ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA	
	RELAZIONE TECNICA – PROGETTO DI FINE ANNO	Pagina 67 di 115

```

        incf      buffer_index_rx
;-----
;      FINE SALVATAGGIO DATI RICEVUTI
;-----
        goto    irq_tmr

```

Ogni pacchetto ricevuto è formato da quattro byte i quali dovranno essere salvati nella corretta posizione ovvero nella variabile desiderata.

Il programma si affida ad un indice incrementale; il primo byte ricevuto viene salvato nella prima variabile poiché l'indice vale uno; al secondo byte l'indice sarà stato incrementato di una posizione per salvare il dato nella posizione corretta e così fino al quarto bit.

Dopo ogni pacchetto l'indice viene azzerato per poter ricominciare la ricezione.

Comunicazione – Lato software

Il programma per controllare il PC da computer è stato realizzato nell'IDE (Integrated development environment) Borland Delphi7 con il linguaggio di programmazione Delphi, omonimo dell'ambiente di sviluppo.

Il programma è il risultato di 18 versioni (l'ultima versione stabile è la 1.8 mentre la prima versione è stata la 0.1) e conta 1000-2000 righe di codice e diverse form.

Come per il programma del microprocessore, non è possibile inserirlo totalmente nel testo ma è stato allegato in forma integrale a questo documento (Documento allegato esterno 2).

Del programma, più della struttura, interessa il funzionamento e perciò si è preferito analizzare form per form invece dei diagrammi di flusso; ogni form è però commentata in ogni sua funzione, a differenza del programma per microprocessore.

L'Illustrazione 28 mostra le principali finestre del programma durante il funzionamento del robot.

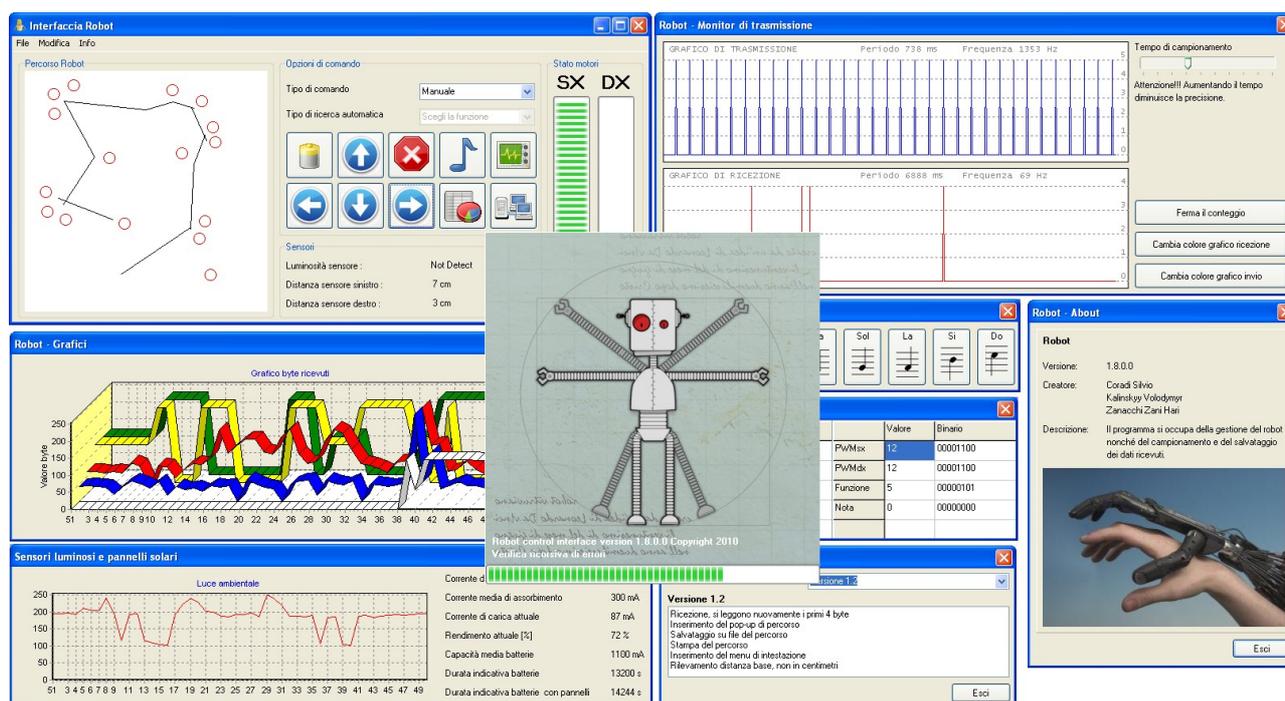
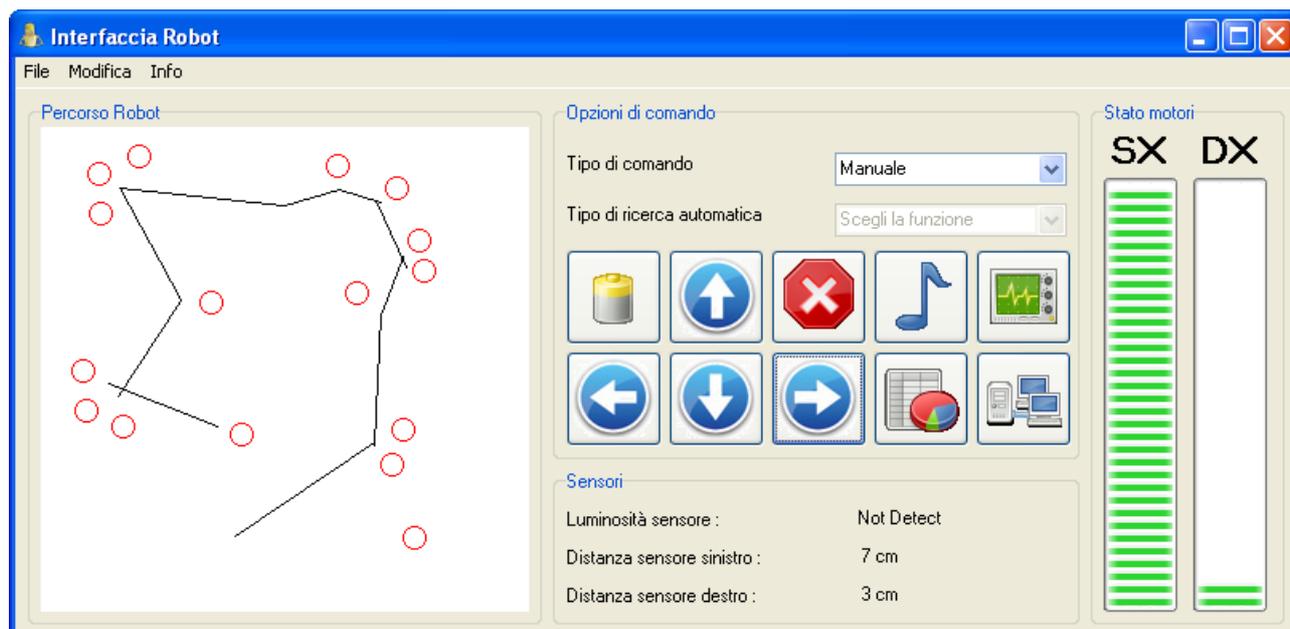


Illustrazione 28: Schermata omnicomprensiva

	ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA	
	RELAZIONE TECNICA – PROGETTO DI FINE ANNO	Pagina 69 di 115

Finestre principali

Finestra index – Robot



La finestra principale è, ovviamente, la finestra più importante del programma; in essa vi sono contenuti tutti i collegamenti necessari e sono riassunte tutte le informazioni più importanti.

Partendo da sinistra si nota subito l'immagine del percorso probabile: questo percorso tenta di ricostruire il percorso del robot partendo dalla velocità dei motori e dal loro tempo di accensione; questo grafico è abbastanza approssimativo anche a causa dell'odometria imprecisa dei robot dual-drive. I cerchi rossi sono delle modellizzazioni degli ostacoli inserite per ricostruire l'ambiente nel quale si muove il robot.

Al centro, nella parte alta, si trovano le comboBox di comando; da qui si possono selezionare tutte le funzioni desiderate (selezionando il tipo di comando automatico i comandi di direzione vengono disabilitati) e il comando manuale visibile più in basso.

Nella parte centrale ci sono delle icone le quali aprono le rispettive funzioni; ogni icona ricorda la funzione stessa per creare un feedback ed aumentare la velocità di scelta: in ordine di importanza ci sono la finestra dei grafici (icona grafico a torta), la finestra di comunicazione (icona oscilloscopio), la finestra di monitoraggio (icone computers), la finestra dell'altoparlante (icona nota) e la finestra della luce e della ricarica tramite pannelli solari (icona della pila in ricarica).

Sempre sulla parte centrale, in basso, si possono vedere i dati riassunti dei sensori i quali, grazie a formule di approssimazione, mostrano il Lux nella stanza e le distanze rilevate dai due sensori.

	ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA	
	RELAZIONE TECNICA – PROGETTO DI FINE ANNO	Pagina 70 di 115

Infine, a destra, ci sono due barre di caricamento disposte per mostrare graficamente il PWM dei motori (nell'immagine il robot sta curvando a sinistra in modo molto marcato).

Ogniquale volta viene cambiata la funzione del robot tramite le due comboBox, il programma si occupa di inviare al robot i dati necessari. Viene ora analizzata l'invio di una tipica istruzione, la scritta *{{DA QUESTO SPAZIO È STATO RIMOSSO DEL CODICE NON FONDAMENTALE}}* indica al lettore che in quel punto è stata rimossa una parte non necessaria nella spiegazione.

```

procedure TfrmRobot.cmbFunzioneChange(Sender: TObject);
begin
  TX1 := 12;
  TX2 := 12;
  TX3 := cmbComando.ItemIndex * (cmbFunzione.ItemIndex + 1);
  TX4 := 0;
  index_TX := 1;
  if TX3 > 2 Then TX3 := TX3 + 2;
  tmrTX.Enabled := True;
end;

procedure TfrmRobot.tmrTXTimer(Sender: TObject);
begin
  // Ad intervalli regolari invia tutti i byte del pacchetto
  case index_TX of
    1 : begin
      VaComm1.WriteChar(char(TX1));
      inc(index_TX);
      end;
    2 : begin
      VaComm1.WriteChar(char(TX2));
      inc(index_TX);
      end;
    3 : begin
      VaComm1.WriteChar(char(TX3));
      inc(index_TX);
      end;
    4 : begin
      VaComm1.WriteChar(char(TX4));
      tmrTX.Enabled := False;
      index_TX := 1;

      {{DA QUESTO SPAZIO È STATO RIMOSSO DEL CODICE NON FONDAMENTALE}}
      end;
  end;

  {{DA QUESTO SPAZIO È STATO RIMOSSO DEL CODICE NON FONDAMENTALE}}
end;

```

Il codice precedente serve ad aggiornare il robot sulla funzione scelta ed è il codice generale

	ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA	
	RELAZIONE TECNICA – PROGETTO DI FINE ANNO	Pagina 71 di 115

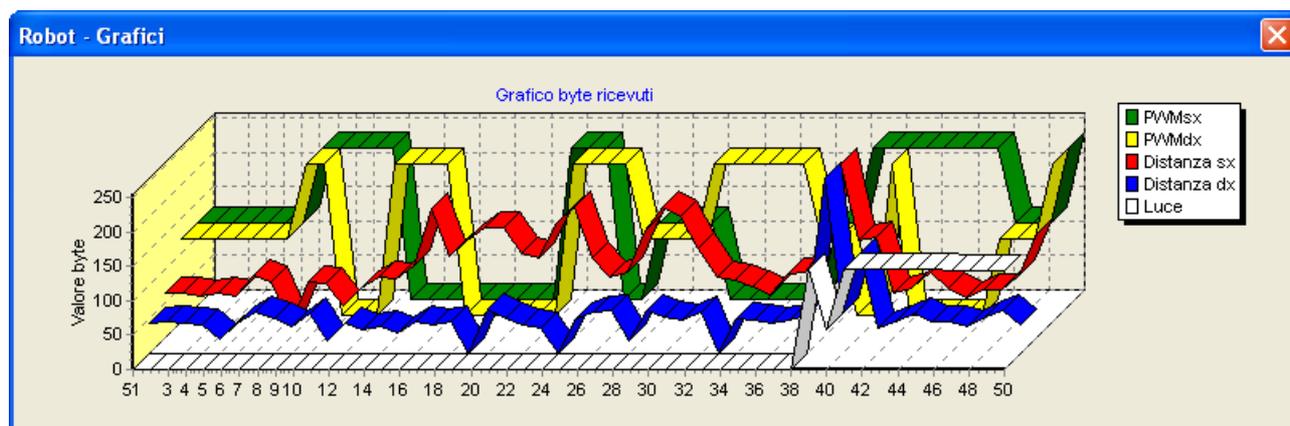
dell'invio di un pacchetto di dati.

La prima procedura (cmbFunzioneChange) prepara l'invio dei vari byte del pacchetto (TX1, TX2, TX3, TX4) e gli assegna i corretti valori; in questo caso si possono vedere i primi due byte corrispondenti ai due segnali PWM posti a 12 (ovvero il 50%) e quindi motori fermi, il terzo byte invia il numero della funzione scelta dalla comboBox e il quarto non è utilizzato (in questo momento).

Infine la procedura pone a uno la variabile index_TX ed attiva il timer di trasmissione; il timer di trasmissione, regolato ad intervalli di 20ms, invia un carattere ad ogni intervallo incrementando la variabile index_TX; finita la trasmissione il timer si disabilita e rimane in attesa di altre trasmissioni.

La trasmissione vera e propria avviene grazie al componente VaComm il quale interfaccia Delphi alla seriale permettendo di inviare e ricevere caratteri in modo molto semplice.

Finestra del grafico dei segnali – Grafici



La finestra grafici è stata pensata per mostrare l'andamento dei dati ricevuti nel tempo; la legenda a destra permette di trovar il dato scelto nel grafico.

Il valore di questi dati varia da 0 a 255; questi valori dipendono dal fatto che i pacchetti contengono un byte ovvero 8 bit e possono assumere valori da 0 a 255: $2^8=256$.

N.B. : I grafici dei due PWM (sx e dx) sono stati volutamente amplificati per essere rilevanti nel grafico; nello specifico questi valori variano da 0 a 24 e sono amplificati di 10 volte.

I dati ricevuti dal computer sono quindi:

- **PWMsx** : ovvero il valore del PWM sul motore sinistro.
- **PWMdx** : ovvero il valore del PWM sul motore destro; questi due dati permettono di ricostruire il percorso probabile del robot ed hanno commutazioni nette e veloci inoltre, grazie ad un sistema software di riconoscimento di errori, i segnali PWM vengono sempre

	ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA	
	RELAZIONE TECNICA – PROGETTO DI FINE ANNO	Pagina 72 di 115

approssimati al livello più vicino.

- **Distanza sx** : ovvero il valore (convertito da un ADC) digitale della distanza misurata dal sensore sinistro.
- **Distanza dx** : ovvero il valore (convertito da un ADC) digitale della distanza misurata dal sensore destro; con questi valori si possono ricavare le distanze e le posizioni dei cerchi nel percorso probabile (vedi finestra principale).
- **Luce** : ovvero il valore (convertito da un ADC) digitale della luce massima rilevata dal robot. Questo dato si attiva solamente quando viene attivato l'inseguitore di luce (infatti si può vedere per la prima parte del grafico il valore luce spento).

Finestra di trasmissione – Monitor di trasmissione



Questa finestra mostra il traffico dei pacchetti attraverso il mezzo trasmissivo; ogni impulso nel grafico corrisponde ad un pacchetto (ricevuti dal robot per quelli in basso mentre inviati dal robot per quelli in alto); l'altezza di un impulso indica quanti byte contiene quel pacchetto, nel nostro caso la comunicazione è sempre composta da pacchetti di quattro byte.

Inoltre, come si può notare dal grafico, la trasmissione di dati dal robot al PC è continua e periodica per tenere il software aggiornato e non farlo rimanere per troppo tempo senza aggiornamenti.

La ricezione, invece, è sporadica e non continuativa poiché basta anche una singola istruzione (come per esempio l'invio del comando esplora) per tenere il robot impegnato per il tempo necessario.

Per ogni grafico è mostrato il tempo che intercorre tra i due impulsi e la loro relativa frequenza; inoltre si può variare il tempo di campionamento del sistema per visualizzare più o meno impulsi

per pagina.

Finestre secondarie

Finestra di comunicazione – Comunicazione

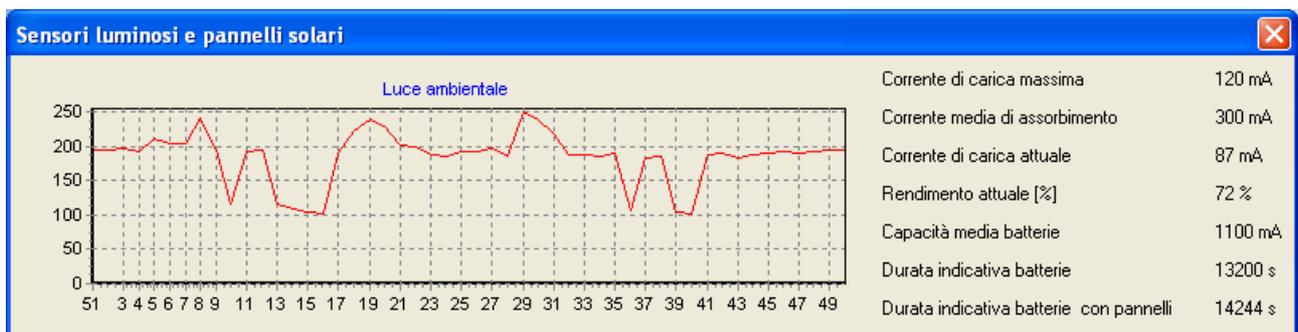
Robot - Comunicazione dati					
	Valore	Binario		Valore	Binario
PwM _{sx}	6	00000110	PwM _{sx}	12	00001100
PwM _{dx}	18	00010010	PwM _{dx}	12	00001100
Distanza dx	66	01000010	Funzione	5	00000101
Distanza sx	39	00100111	Nota	0	00000000
Luce	0	00000000			

La finestra di comunicazione mostra il valore dei dati inviati e di quelli ricevuti istante per istante; a differenza della finestra Grafici, qui non importa l'andamento generale bensì quello istantaneo.

Tutti i valori sono inoltre convertiti in binario in real time per velocizzare alcuni calcoli nei prototipi e nei successivi sviluppi.

La conversione binaria avviene in real-time ad intervalli scanditi da un timer; grazie alle funzioni MOD e DIV le quali forniscono, rispettivamente, il resto e il quoziente di una divisione è possibile dividere la cifra decimale per i valori binari (128, 64, 32, 16, 8, 4, 2, 1) ed ottenere il valore binario.

Finestra di luce e pannelli solari – Luce

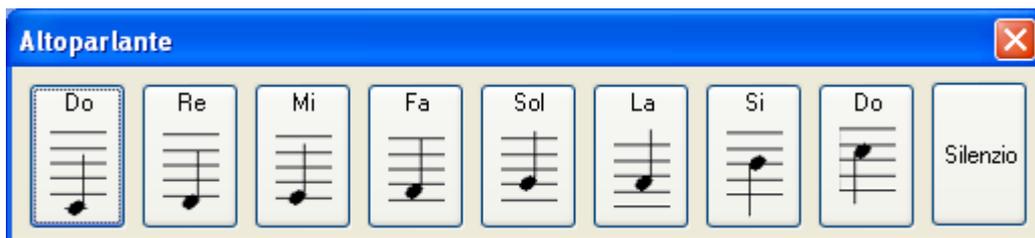


Questa finestra si attiva solamente quando viene attivata la funzione di inseguitore solare; essa mostra l'andamento della luce massima rilevata nel tempo e calcola alcuni parametri relativi alle celle solari.

Conoscendo dati come la corrente di carica massima, la capacità delle batterie e la corrente media di

assorbimento e rilevandone altri come la corrente di carica attuale si possono calcolare il rendimento del sistema e la durata indicativa delle batterie.

Finestra audio - Altoparlante



Questa finestra si occupa della gestione della funzione altoparlante; è composta da nove pulsanti corrispondenti ad un'ottava di note più un DO alto e un tasto per interrompere l'esecuzione della musica.

```

procedure TfrmAltoparlante.cmdDOClick(Sender: TObject);
begin
  // Nota Do
  TX1 := 12;
  TX2 := 12;
  TX3 := 8;
  TX4 := 61;
  index_TX := 1;
  frmRobot.tmrTX.Enabled := True;
end;

```

Questo codice, affine al codice di trasmissione già visto, si occupa di istruire il robot su quale nota suonare; nello specifico viene suonata la nota DO (grave) la quale corrisponde al valore 61.

In sintesi il codice richiede l'arresto dei motori, informa la funzione scelta (funzione 8 ovvero altoparlante) e in TX4 definisce la frequenza di questa nota; segue una tabella dei valori di TX4 corrispondenti alle note scelte:

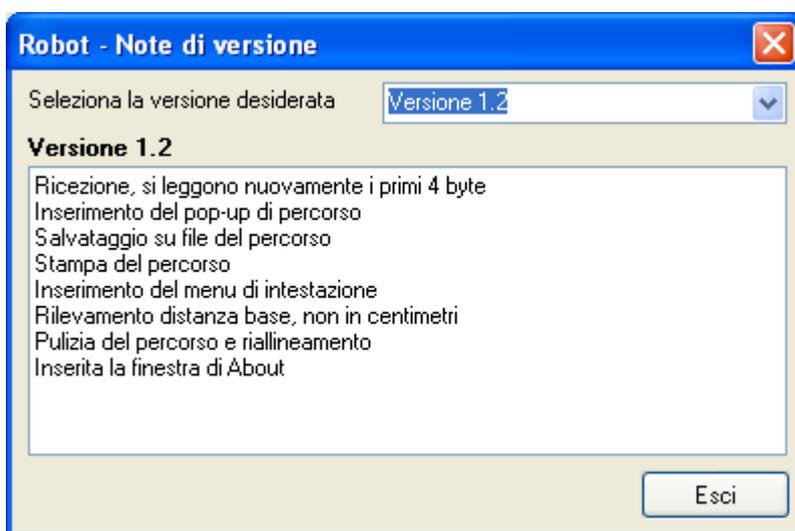
Nota	Frequenza [Hz]	TX4
DO	262	61
RE	394	54
MI	330	48
FA	349	45
SOL	342	41
LA	440	36
SI	494	31

	ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA	
	RELAZIONE TECNICA – PROGETTO DI FINE ANNO	Pagina 75 di 115

DO	523	27
Silenzio	0	0

Finestra ausiliarie

Finestra versioni – Note di versione



Questa finestra si occupa di mostrare tutte le modifiche effettuate (note di versione) per ogni versione del programma; tramite una comboBox si possono vedere tutte le versioni disponibili e anche le modifiche effettuate.

```

procedure TfrmNoteVer.cmbVersioneChange(Sender: TObject);
begin
  // Gestione note di versione
  lblVersione.Caption :=
  'V'+cmbVersione.Items.ValueFromIndex[cmbVersione.ItemIndex];
  Case cmbVersione.ItemIndex of
    0 : begin
      memFeature.Clear;
      memFeature.Lines.Add('F1');
      memFeature.Lines.Add('F2');
      memFeature.Lines.Add('F3');
      memFeature.Lines.Add('F4');
      memFeature.Lines.Add('F5');
      memFeature.Lines.Add('F6');
    end;
  {{DA QUESTO SPAZIO È STATO RIMOSSO DEL CODICE NON FONDAMENTALE}}

```

Il codice mostra un esempio di commutazione di versione dove Fn sono le note aggiunte per la

	<p style="text-align: center;">ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA</p>	
	<p style="text-align: center;">RELAZIONE TECNICA – PROGETTO DI FINE ANNO</p>	<p style="text-align: right;">Pagina 76 di 115</p>

versione selezionata; per ogni nuova versione è necessario inserire un nuovo codice, identico a quello mostrato, collegato alla nuova funzione ed inserirgli le modifiche effettuate.

È molto importante gestire correttamente ogni versione poiché nel caso un componente non dovesse funzionare basterebbe verificare quando è stata effettuata l'ultima sua modifica per trovare il codice corretto.

Le versioni di entrambi i programmi non sono state gestite da un cosiddetto Version Control poiché quando il tempo (e quindi il numero di versioni) non è così elevato questo non è necessario; quando verranno implementate le nuove funzioni si potrà pensare di riscrivere il programma sotto Version Control seguendo anche i principi generali dell'ingegneria software.

Finestra informazioni – About



La finestra di About, o di informazione in italiano, raccoglie le informazioni essenziali riguardo il programma stesso. Questa finestra è molto importante poiché fornisce informazioni essenziali ed inquadra il programma nel suo tempo di sviluppo.

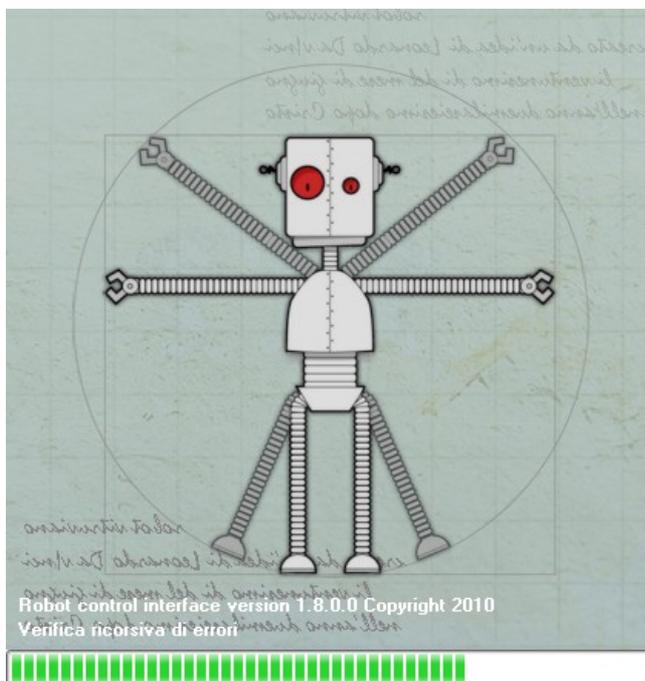
Informazioni come versione, creatori, descrizione sommaria del programma ed eventualmente sito web degli sviluppatori sono incluse in questa pagina.

È stata inserita anche un'immagine per rendere la finestra più estetica.

Questa finestra è solamente estetica e quindi l'unico codice contenuto consente di chiudere la finestra ogniqualvolta venga premuto il bottone 'Esci'.

	<p style="text-align: center;">ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA</p>	
	<p style="text-align: center;">RELAZIONE TECNICA – PROGETTO DI FINE ANNO</p>	<p style="text-align: right;">Pagina 77 di 115</p>

Finestra di caricamento – Splash Screen



La finestra di caricamento si attiva all'apertura del programma e serve a coprire un tempo tipicamente morto nel quale il programma può caricare dati dai database o elaborare grosse quantità di dati.

Molto spesso la Splash Screen è usata anche come pubblicità per mostrare versione, creatore e azienda produttrice del software.

Nel nostro caso queste informazioni sono mostrate sopra la barra di caricamento e condizionano tutte le informazioni del programma; per esempio, ad ogni cambio di versione non è necessario cercare in ogni form il riferimento alla versione attuale poiché tutte sono collegate alla versione della

Splash Screen: modificando questa variabile tutto il programma ne viene influenzato.

La Splash Screen consiste anche in una simpatica immagine ritraente un robot nella classica posa (inscritto in un cerchio ed in un quadrato) dell'uomo di Vitruvio.

	ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA	
	RELAZIONE TECNICA – PROGETTO DI FINE ANNO	Pagina 78 di 115

Bibliografia

Sitografia¹

Wikipedia, EIA RS-232, http://it.wikipedia.org/wiki/EIA_RS-232

Datasheetcatalog, <http://www.datasheetcatalog.com/>

Digilander, L293D, http://digilander.libero.it/beamweb/driver_motori.htm#293

Digilander, PWM, http://digilander.libero.it/beamweb/CONTROLLO_PWM.htm

Fornitura

Robot-Italy, <http://www.robot-italy.com/>

Ebay, <http://www.ebay.it>

1 Sitografia aggiornata in data 6 giugno 2010

	<p style="text-align: center;">ISTITUTO TECNICO INDUSTRIALE STATALE LICEO SCIENTIFICO TECNOLOGICO "Benedetto Castelli" BRESCIA</p>	
	<p style="text-align: center;">RELAZIONE TECNICA – PROGETTO DI FINE ANNO</p>	<p style="text-align: right;">Pagina 79 di 115</p>

Licenza e copyright

Licenza di distribuzione Creative Common 2.5²

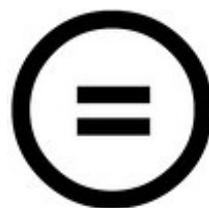
Tu sei libero di riprodurre, distribuire, comunicare al pubblico, esporre in pubblico, rappresentare, eseguire e recitare quest'opera alle seguenti condizioni :



Attribuzione — Devi attribuire la paternità dell'opera nei modi indicati dall'autore o da chi ti ha dato l'opera in licenza e in modo tale da non suggerire che essi avallino te o il modo in cui tu usi l'opera.



Non commerciale — Non puoi usare quest'opera per fini commerciali.



Non opere derivate — Non puoi alterare o trasformare quest'opera, ne' usarla per crearne un'altra.

Prendendo atto che :

Rinuncia — E' possibile rinunciare a qualunque delle condizioni sopra descritte se ottieni l'autorizzazione dal detentore dei diritti.

Pubblico Dominio — Nel caso in cui l'opera o qualunque delle sue componenti siano nel pubblico dominio secondo la legge vigente, tale condizione non è in alcun modo modificata dalla licenza.

Altri Diritti — La licenza non ha effetto in nessun modo sui seguenti diritti:

- Le eccezioni, libere utilizzazioni e le altre utilizzazioni consentite dalla legge sul diritto d'autore;
- I diritti morali dell'autore;
- Diritti che altre persone possono avere sia sull'opera stessa che su come l'opera viene utilizzata, come il diritto all'immagine o alla tutela dei dati personali.

Nota — Ogni volta che usi o distribuisce quest'opera, devi farlo secondo i termini di questa licenza, che va comunicata con chiarezza.

² La licenza completa è disponibile all'indirizzo <http://creativecommons.org/licenses/by-nc-nd/2.5/it/legalcode>

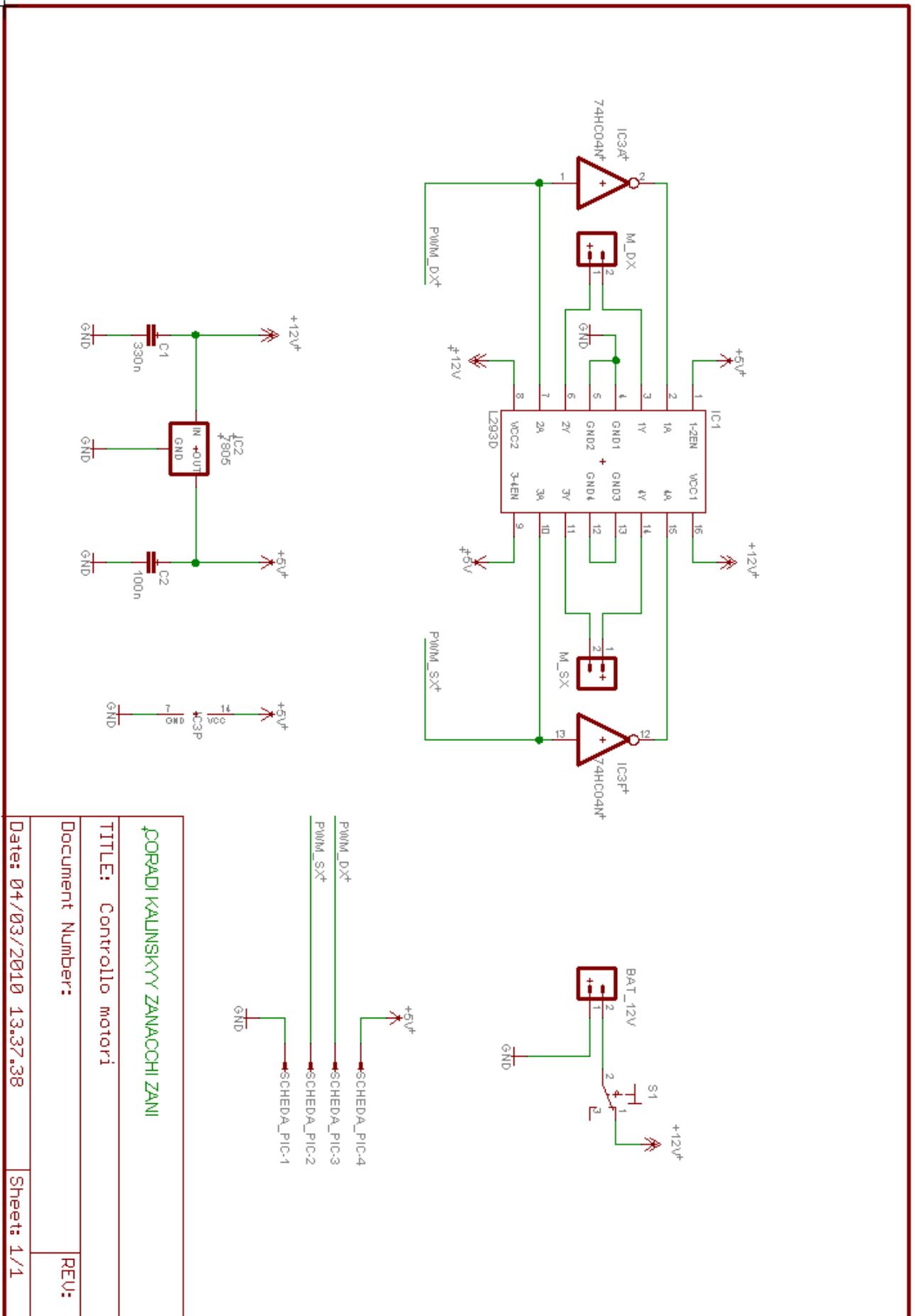


Allegati

Indice generale degli allegati

Allegati.....	80
Allegato 01 – Schematico scheda controllo motori.....	81
Allegato 02 – PCB scheda controllo motori.....	82
Allegato 03 – Schematico scheda PIC sheet 1.....	83
Allegato 04 – Schematico scheda PIC sheet 2.....	84
Allegato 05 – PCB scheda PIC.....	85
Allegato 06 – Schematico scheda sensori.....	86
Allegato 07 – PCB scheda sensori.....	87
Allegato 08 – Schematico scheda seriale.....	88
Allegato 09 – PCB scheda seriale.....	89
Allegato 10 – Schematico scheda Scheda1.....	90
Allegato 11 – PCB scheda Scheda1.....	91
Allegato 12 – Schema scheda trasmettitore wireless.....	92
Allegato 13 – PCB scheda trasmettitore wireless.....	93
Allegato 14 – Listato funzioni manuale e esplorazione.....	94
Allegato 15 – Listato funzione luce.....	96
Allegato 16 – Listato funzione inseguì.....	99
Allegato 17 – Listato funzione line-following.....	102
Allegato 18 – Listato funzione minisumo.....	104
Allegato 19 – Listato funzione suona.....	106
Allegato 20 – Robot 3D v1.....	108
Allegato 21 – Robot 3D v2.....	109
Allegato 22 – Robot 3D retro.....	110
Allegato 23 – Robot 3D v3.....	111
Allegato 24 – Scheda EasyPic 4.....	112
Allegato 25 – Datasheet TIP120.....	113
Documenti allegati esterni	
Programma completo PIC	
Programma completo Delphi	

Allegato 01 – Schematico scheda controllo motori



GORADI KALINISKYY ZANACCHI ZANI

TITLE: Controllo motori

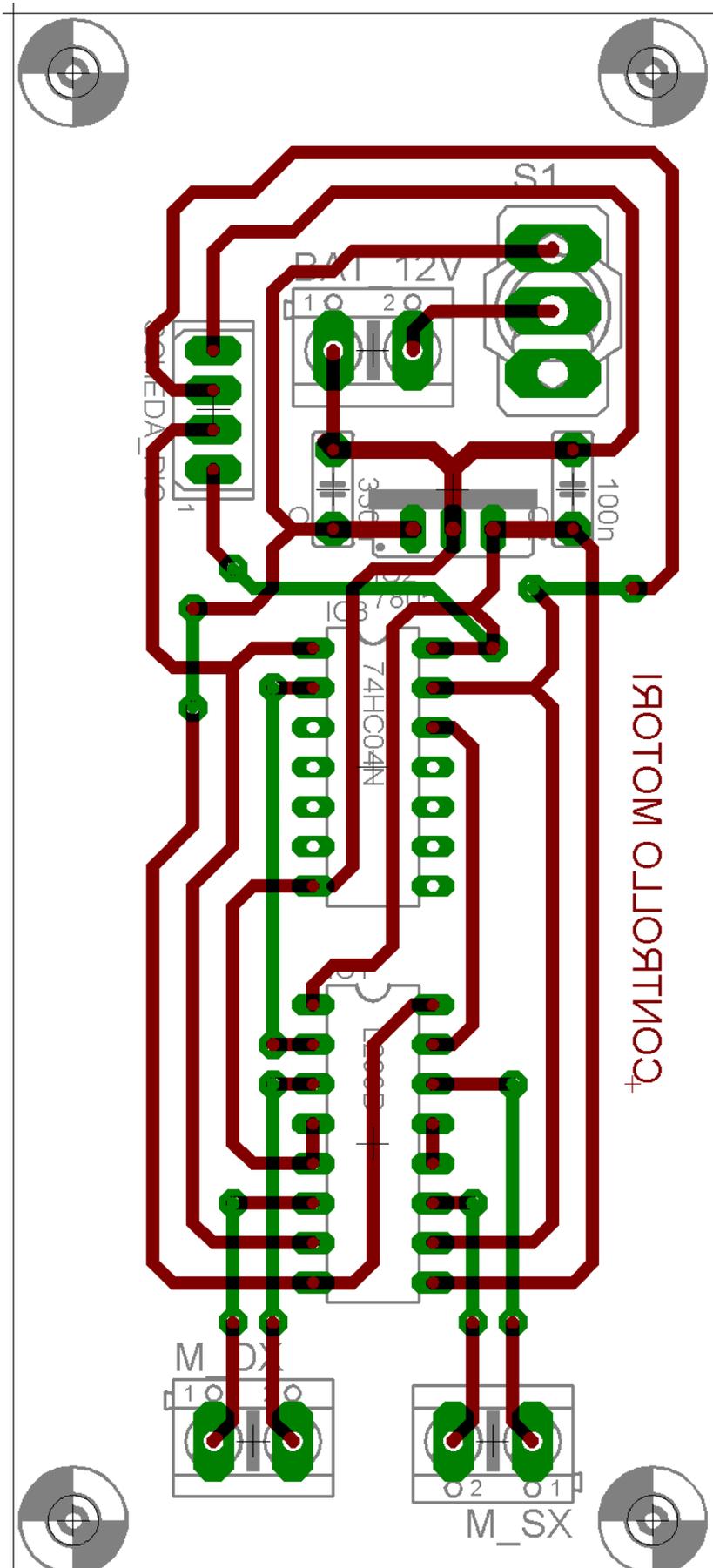
Document Number:

REV:

Date: 04/03/2010 13.37.38

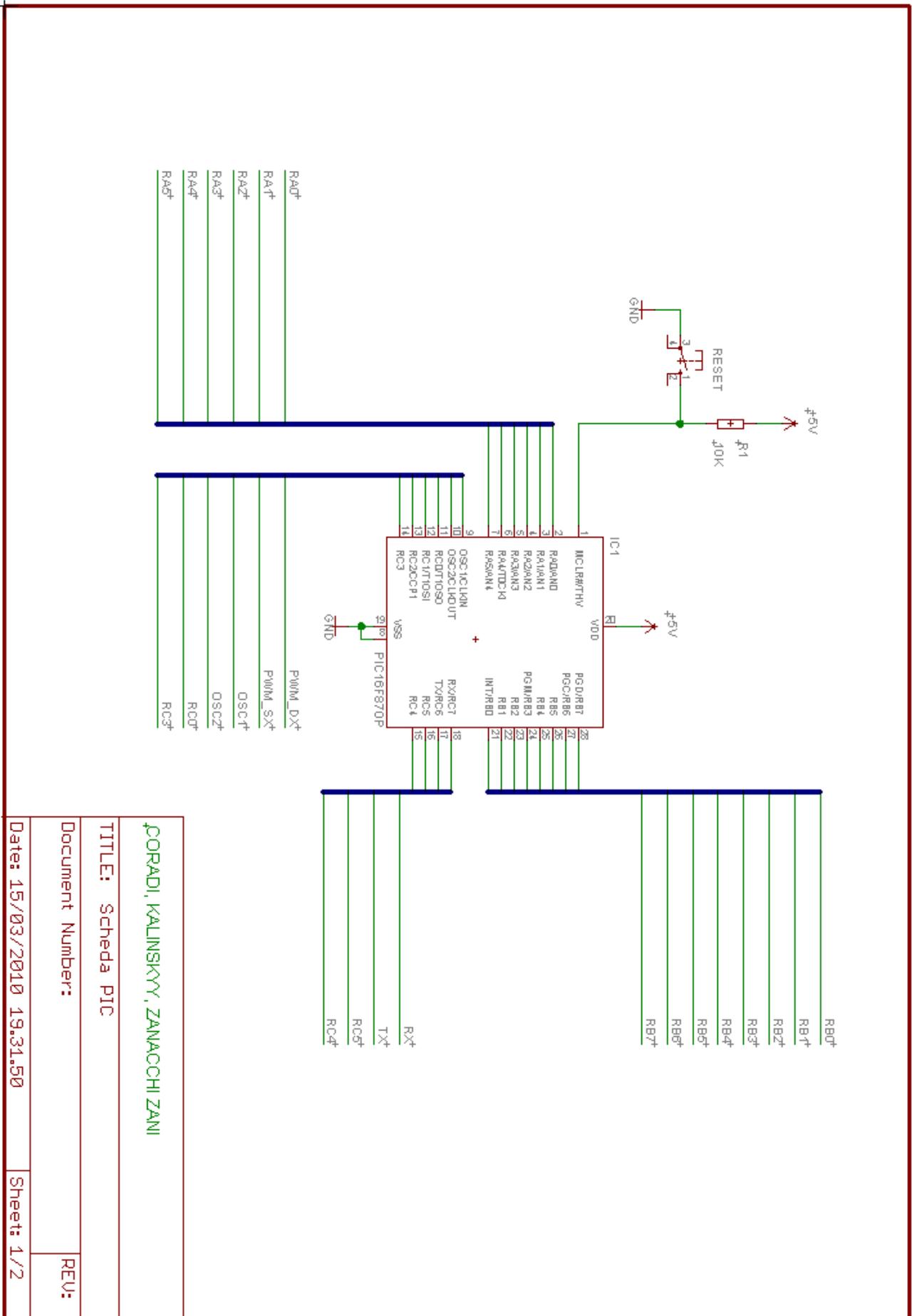
Sheet: 1/1

Allegato 02 – PCB scheda controllo motori





Allegato 03 – Schematico scheda PIC sheet 1



CORADI, KALINSKY, ZANACCHI ZANI

TITLE: Scheda PIC

Document Number:

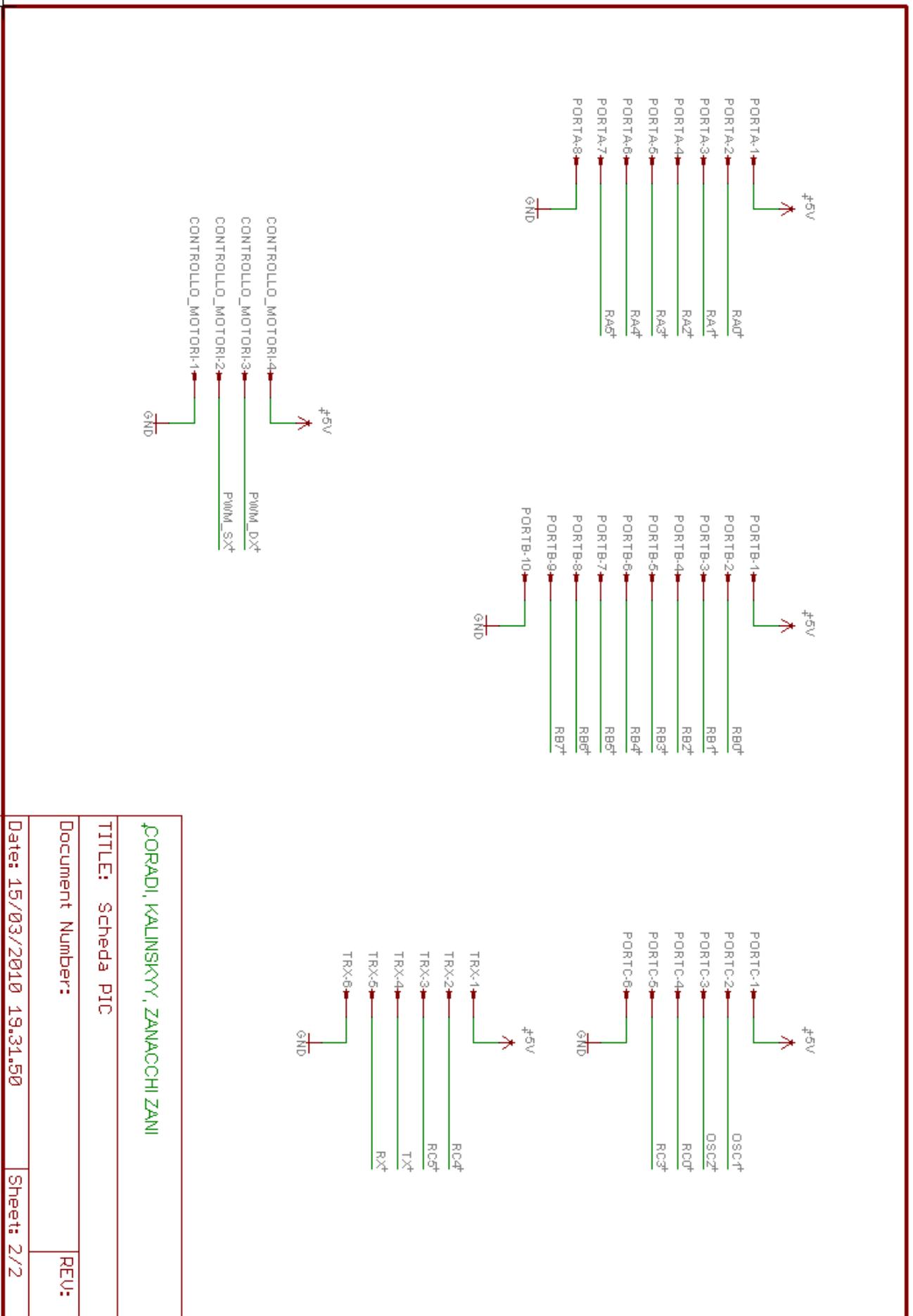
REV:

Date: 15/03/2010 19.31.50

Sheet: 1/2



Allegato 04 – Schematico scheda PIC sheet 2



PORADI, KALINSKY, ZANACCHI ZANI

TITLE: Scheda PIC

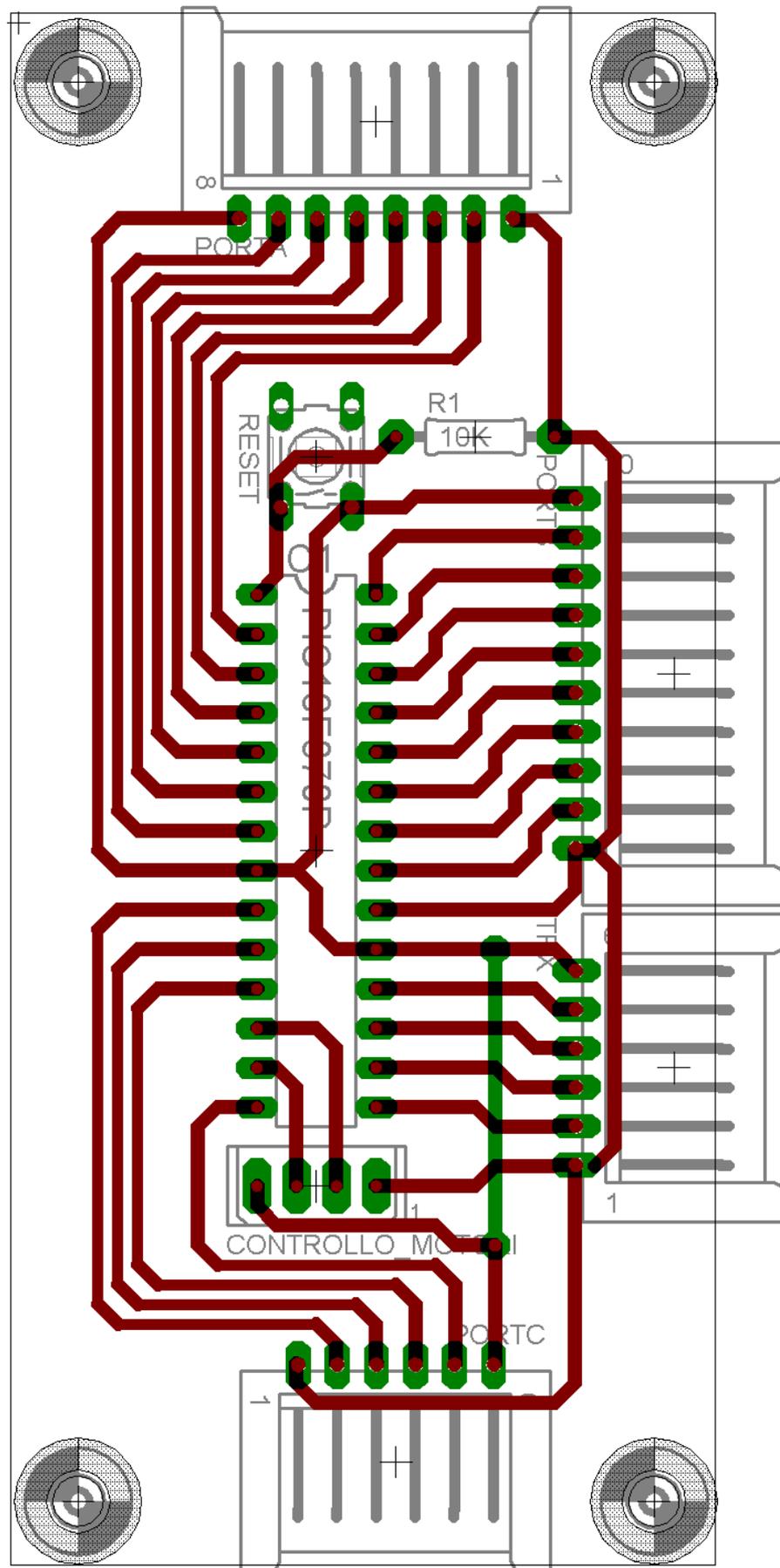
Document Number:

REV:

Date: 15/03/2010 19.31.50

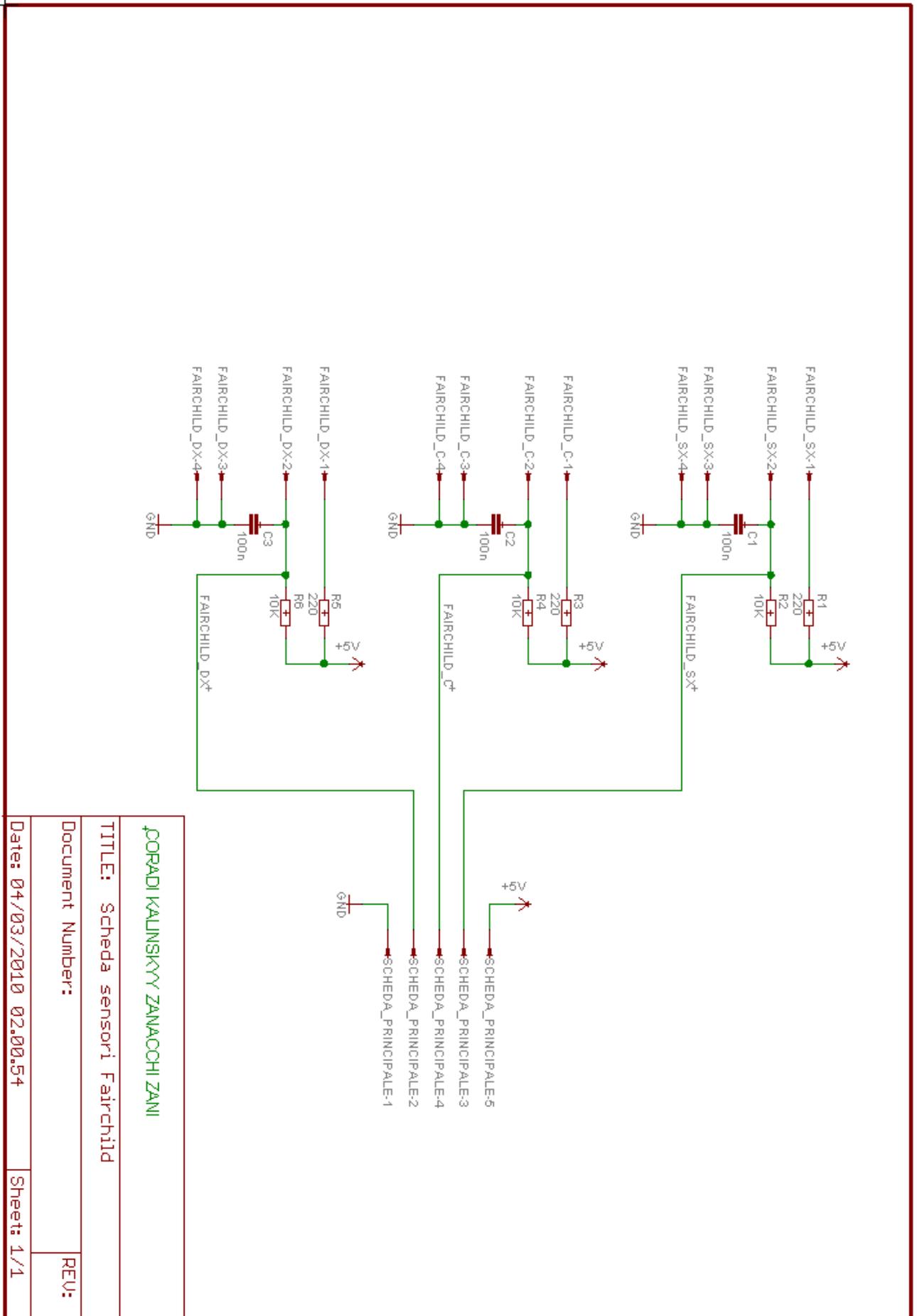
Sheet: 2/2

Allegato 05 – PCB scheda PIC

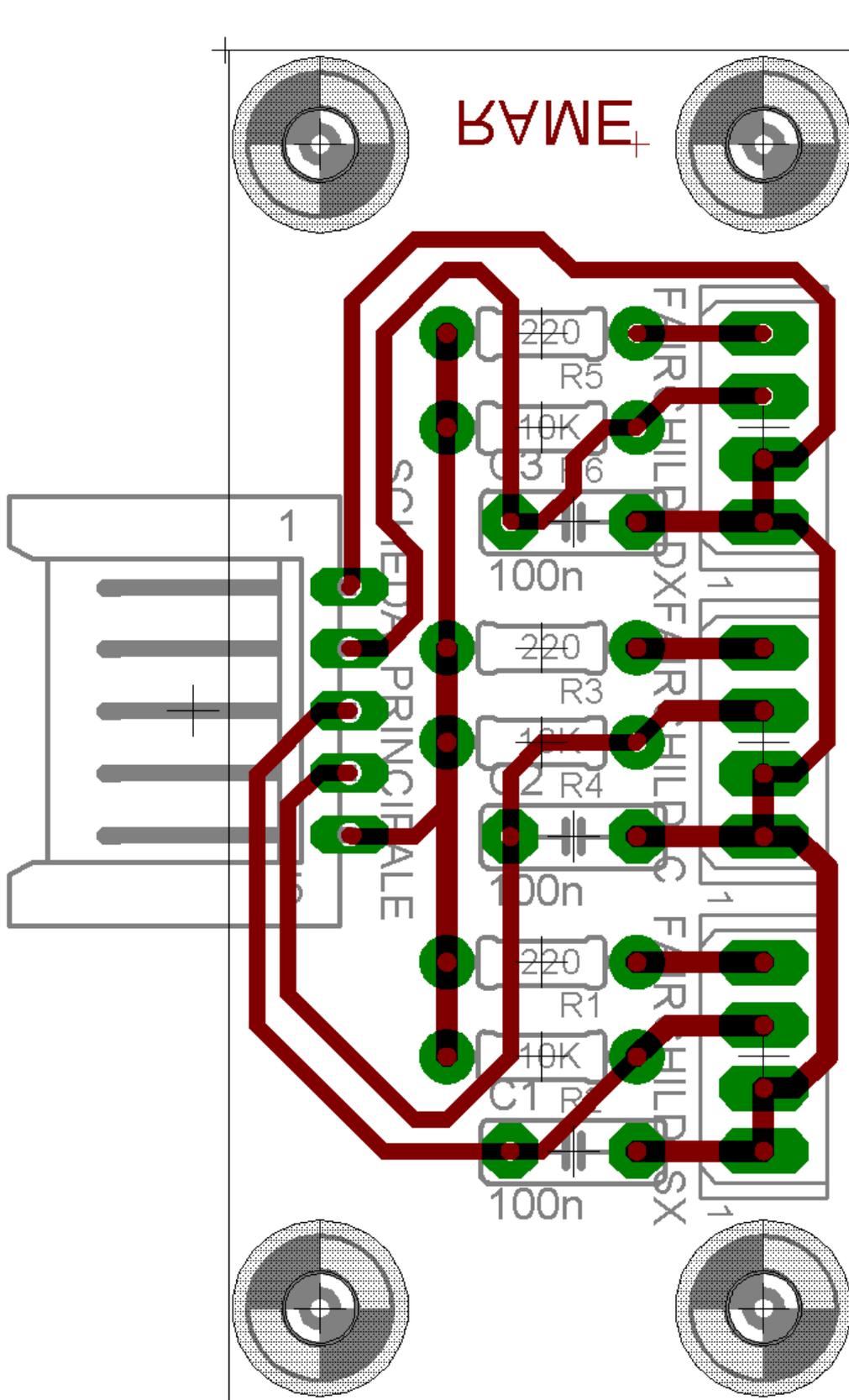




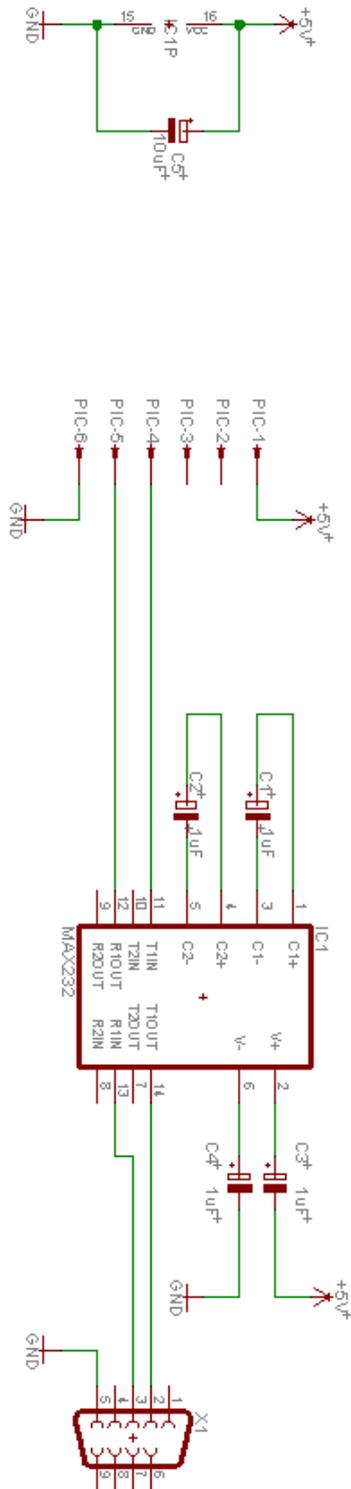
Allegato 06 – Schematico scheda sensori



Allegato 07 – PCB scheda sensori



Allegato 08 – Schematico scheda seriale



FORADI, KALINSKIYY, ZANACCHI ZANI

TITLE: RS232

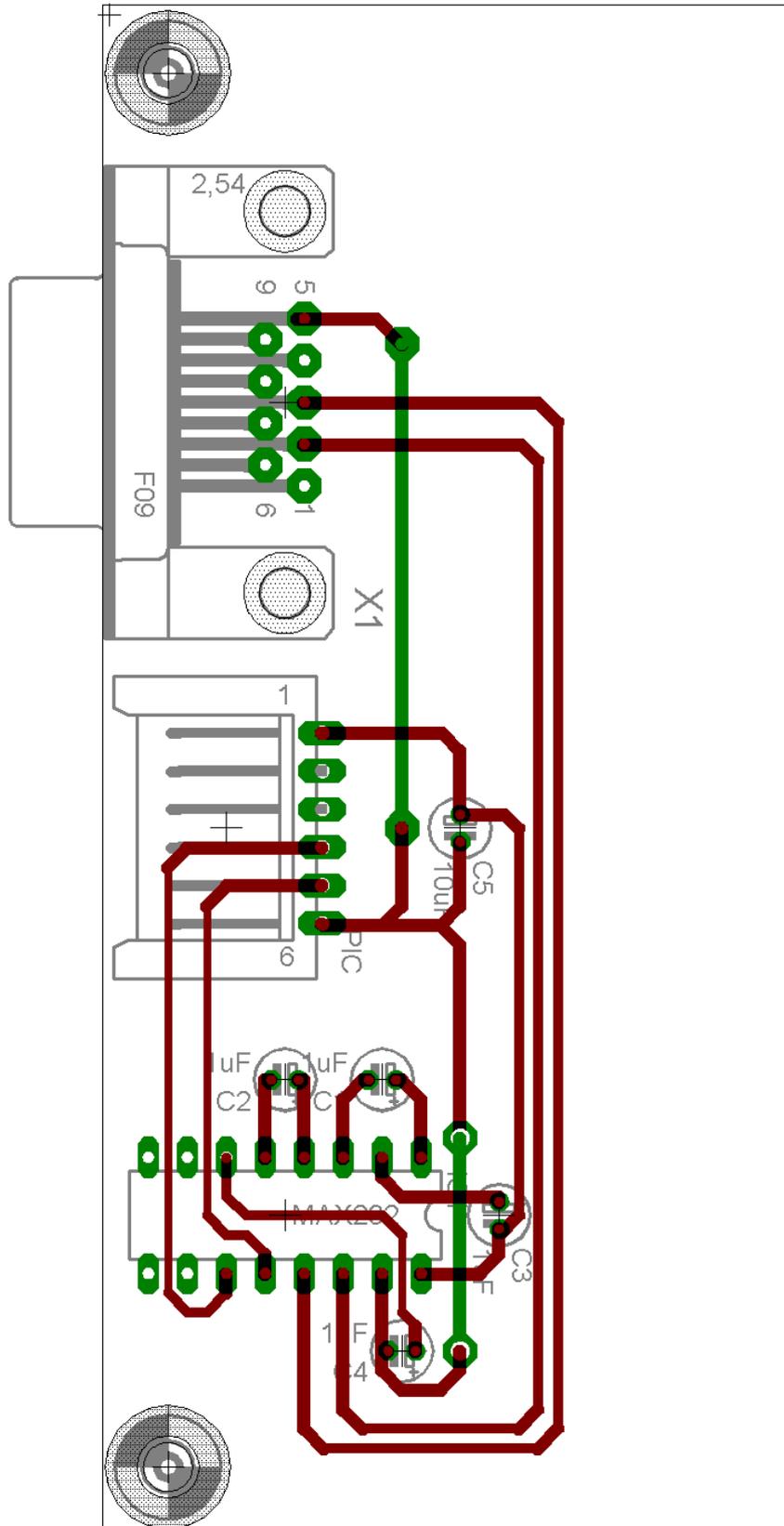
Document Number:

Date: 14/04/2010 18.52.01

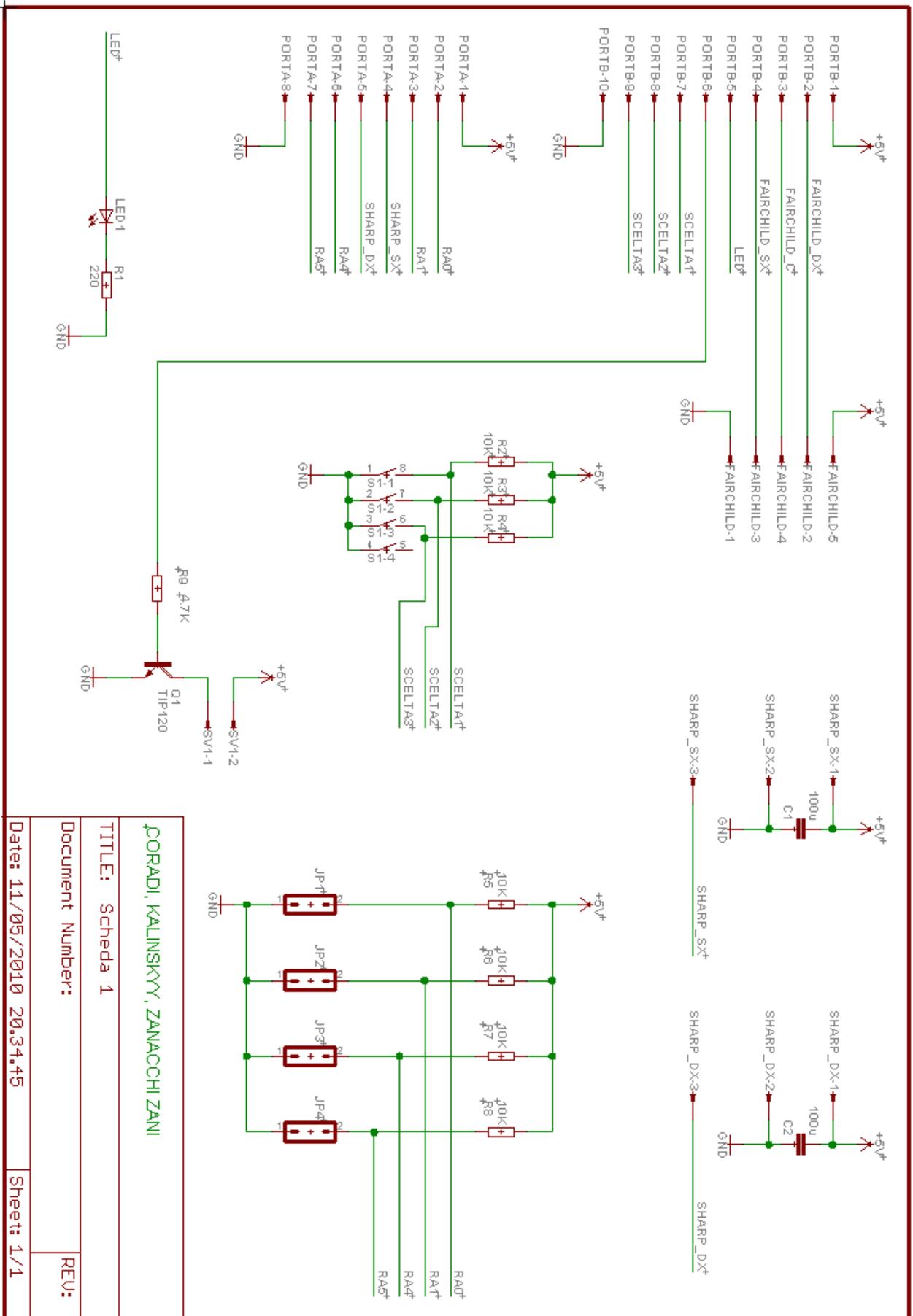
Sheet: 1/1

REV:

Allegato 09 – PCB scheda seriale

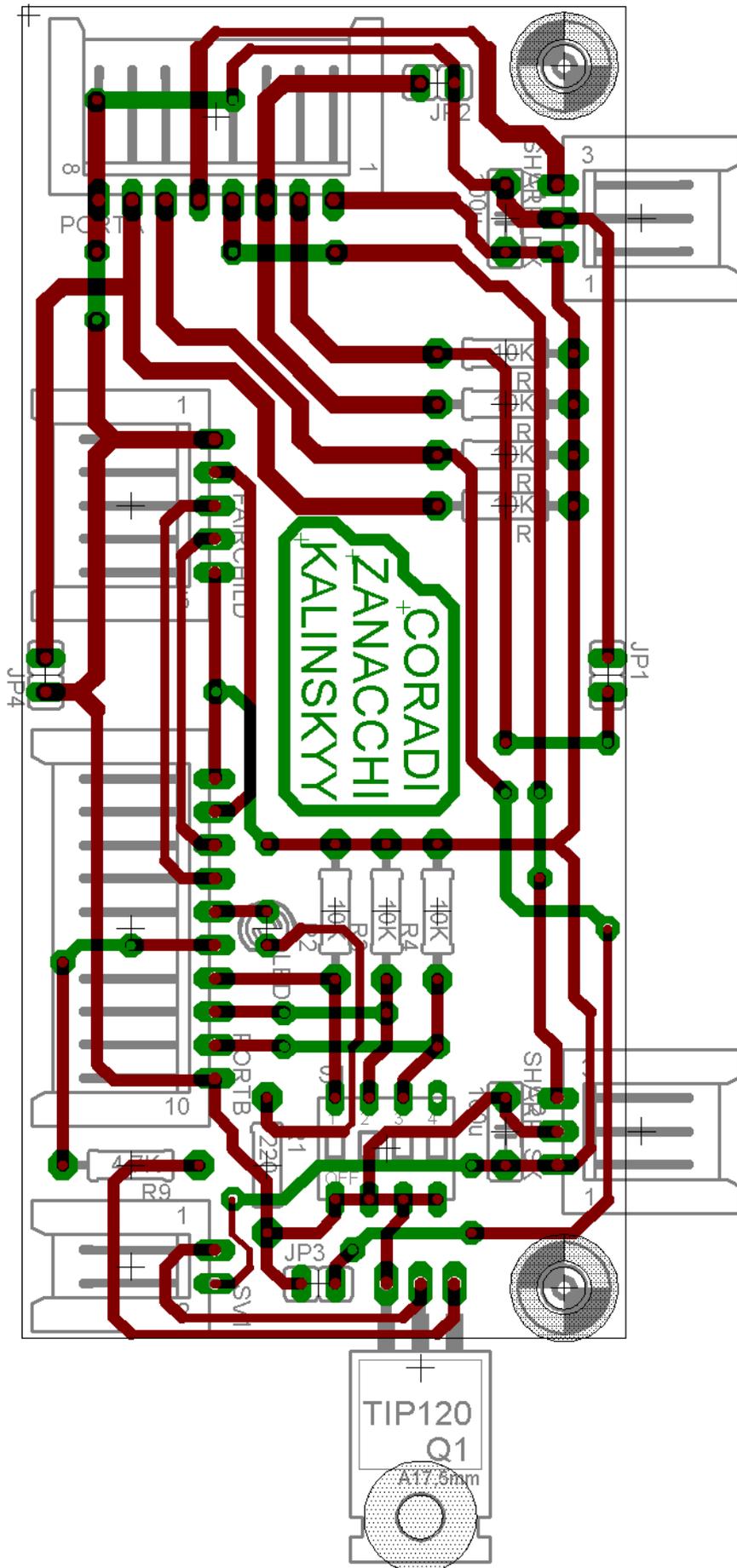


Allegato 10 – Schematico scheda Scheda1

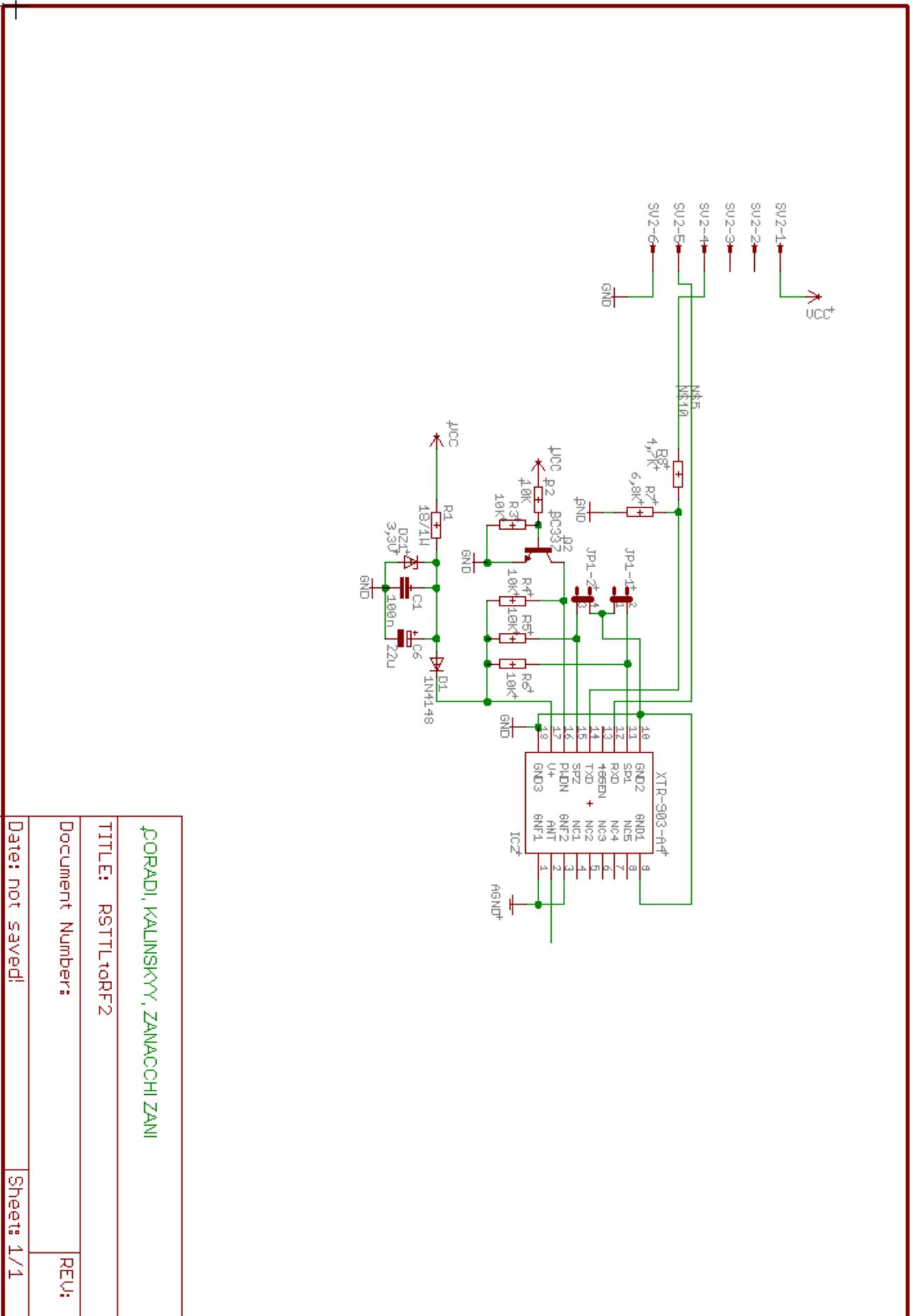


<p>PORADI, KALINSKIYY, ZANACCHI ZANI</p>	
<p>TITLE: Scheda 1</p>	<p>REV:</p>
<p>Document Number:</p>	<p>REV:</p>
<p>Date: 11/05/2010 20.34.45</p>	<p>Sheet: 1/1</p>

Allegato 11 – PCB scheda Scheda1



Allegato 12 – Schema scheda trasmettitore wireless



✪CORADI, KALINSKY, ZANACCHI ZANI

TITLE: RSTLTtoRF2

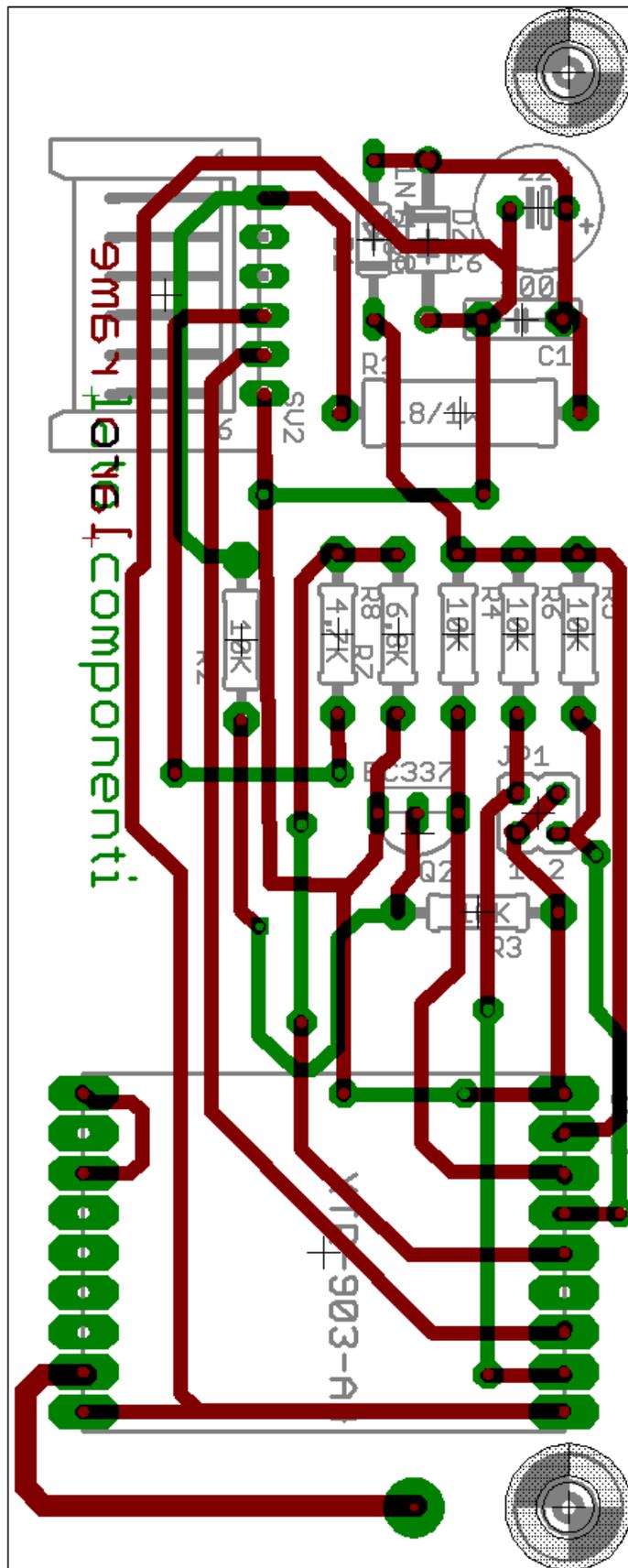
Document Number:

REV:

Date: not saved!

Sheet: 1/1

Allegato 13 – PCB scheda trasmettitore wireless





Allegato 14 – Listato funzioni manuale e esplorazione

```
-----  
;  
;  
COMANDO MANUALE  
-----  
;  
;  
funz_manuale  
    movfw    buffer_pwm_sx_rx    ; Motore SX  
    banksel  CCPR1L  
    movwf    CCPR1L  
    movfw    buffer_pwm_dx_rx    ; Motore DX  
    banksel  CCPR2L  
    movwf    CCPR2L  
  
    goto     end_funzione  
  
-----  
;  
;  
FUNZIONE ROVER - ESPLORAZIONE  
-----  
;  
;  
funz_esplora  
    movfw    sharp_sx  
  
    sublw    B'01100000'    ; 00100000 x esterni  
    btfsc   STATUS,C  
    goto    OSTACOLO2  
  
OSTACOLO1  
    movlw    B'00000000'  
    bcf     STATUS,Z  
    xorwf   ostacolo,W  
    btfss   STATUS,Z  
    goto    end_funzione  
  
    movlw   .1  
    movwf   ostacolo  
    movlw   .1  
    movwf   buffer_pwm_sx_rx  
    banksel CCPR1L  
    movwf   CCPR1L  
    movlw   .23  
    movwf   buffer_pwm_dx_rx  
    banksel CCPR2L  
    movwf   CCPR2L  
  
    call    ritardo  
    banksel PORTB  
    goto    end_funzione  
  
OSTACOLO2
```



```
movfw      sharp_dx

sublw      B'01100000'      ; Sogliabassa 000010011
btfsc     STATUS,C
goto      OSTACOLO3

movlw     B'00000000'
bcf      STATUS,Z
xorwf    ostacolo,W
btfss   STATUS,Z
goto     end_funzione

movlw     .1
movwf    ostacolo

movlw     .23
movwf    buffer_pwm_sx_rx
banksel  CCPR1L
movwf    CCPR1L
movlw     .1
movwf    buffer_pwm_dx_rx
banksel  CCPR2L
movwf    CCPR2L

call     ritardo

banksel  PORTB

goto     end_funzione
```

OSTACOLO3

```
movlw     .0
movwf    ostacolo

movlw     .23
movwf    buffer_pwm_sx_rx
banksel  CCPR1L
movwf    CCPR1L
movlw     .23
movwf    buffer_pwm_dx_rx
banksel  CCPR2L
movwf    CCPR2L

banksel  PORTB

goto     end_funzione
```



Allegato 15 – Listato funzione luce

```
-----  
;                                     LUCE                                     ;  
;                                     ;                                     ;  
;                                     ;                                     ;  
-----  
funz_luce  
    movfw    fr1    ; Salvo i valori acquisiti nei corrispettivi buffer  
    movwf    buf_fr1  
    movfw    fr2  
    movwf    buf_fr2  
    movfw    fr3  
    movwf    buf_fr3  
    movlw    B'11111111'  
    movwf    buf_fr4  
    movfw    fr1  
    subwf    fr2  
    btfsc    STATUS,C  
    goto     fr1_mag  
fr2_mag  
    movfw    buf_fr2  
    movwf    frM1  
    goto     confronto2  
fr1_mag  
    movfw    buf_fr1  
    movwf    frM1  
confronto2  
    movfw    fr3  
    subwf    fr4  
    btfsc    STATUS,C  
    goto     fr3_mag  
fr4_mag  
    movfw    buf_fr4  
    movwf    frM2  
    goto     confronto3  
fr3_mag  
    movfw    buf_fr3  
    movwf    frM2  
confronto3  
    movfw    frM1  
    movwf    buf_frM1  
    movfw    frM2  
    movwf    buf_frM2  
  
    movfw    frM1  
    subwf    frM2
```



```
frM2_mag      btfsc      STATUS,C
               goto      frM1_mag

frM2_mag      movfw     buf_frM2
               movwf     frMM
               goto      confronto4

frM1_mag      movfw     buf_frM1
               movwf     frMM

confronto4    movfw     buf_fr1           ; Ricarica i dati reali
               movwf     fr1
               movfw     buf_fr2
               movwf     fr2
               movfw     buf_fr3
               movwf     fr3
               movfw     buf_fr4
               movwf     fr4

               movfw     frMM
               bcf       STATUS,Z
               xorwf     buf_fr1,W
               btfsc    STATUS,Z
               goto      fr1_magX

               movfw     frMM
               bcf       STATUS,Z
               xorwf     buf_fr2,W
               btfsc    STATUS,Z
               goto      fr2_magX

               movfw     frMM
               bcf       STATUS,Z
               xorwf     buf_fr3,W
               btfsc    STATUS,Z
               goto      fr3_magX

               movfw     frMM
               bcf       STATUS,Z
               xorwf     buf_fr4,W
               btfsc    STATUS,Z
               goto      fr4_magX

fr1_magX     movlw     .18
               movwf     buffer_pwm_sx_rx
```



```
banksel      CCPR1L
movwf        CCPR1L
movlw        .18
movwf        buffer_pwm_dx_rx
banksel      CCPR2L
movwf        CCPR2L
banksel      PORTB
goto         end_mov

fr2_magX

movlw        .6
movwf        buffer_pwm_sx_rx
banksel      CCPR1L
movwf        CCPR1L
movlw        .18
movwf        buffer_pwm_dx_rx
banksel      CCPR2L
movwf        CCPR2L
banksel      PORTB
goto         end_mov

fr3_magX

movlw        .18
movwf        buffer_pwm_sx_rx
banksel      CCPR1L
movwf        CCPR1L
movlw        .6
movwf        buffer_pwm_dx_rx
banksel      CCPR2L
movwf        CCPR2L
banksel      PORTB
goto         end_mov

fr4_magX

movlw        .6
movwf        buffer_pwm_sx_rx
banksel      CCPR1L
movwf        CCPR1L
movlw        .6
movwf        buffer_pwm_dx_rx
banksel      CCPR2L
movwf        CCPR2L

banksel      PORTB
goto         end_mov

end_mov

goto         end_funzione
```



Allegato 16 – Listato funzione insegui

```
-----  
;                                     INSEGUI                                     ;  
;                                     -----  
;                                     ;  
funz_insegui  
    movfw    sharp_dx  
    movwf    sharp_dx_buf  
    movfw    sharp_sx  
    subwf    sharp_dx_buf  
    btfsc    STATUS,C  
    goto     mag_dx  
  
mag_sx  
    movfw    sharp_sx  
    sublw    B'01100000'  
    btfss    STATUS,C          ; Se sharp > x  
    goto     s1_sx  
  
    movfw    sharp_sx  
    sublw    B'00110000'  
    btfss    STATUS,C          ; Se sharp > x  
    goto     s2_sx  
  
    movlw    .23                ; Motore SX fermo  
    movwf    buffer_pwm_sx_rx  
    banksel CCPR1L  
    movwf    CCPR1L  
    movlw    .23                ; Motore DX fermo  
    movwf    buffer_pwm_dx_rx  
    banksel CCPR2L  
    movwf    CCPR2L  
    goto     end_sx  
  
s2_sx  
    movlw    .20                ; Motore SX fermo  
    movwf    buffer_pwm_sx_rx  
    banksel CCPR1L  
    movwf    CCPR1L  
    movlw    .4                 ; Motore DX fermo  
    movwf    buffer_pwm_dx_rx  
    banksel CCPR2L  
    movwf    CCPR2L  
    goto     end_sx  
  
s1_sx  
    movlw    .1                 ; Motore SX fermo
```



```

    movwf    buffer_pwm_sx_rx
    banksel
    movwf    CCPR1L
    movlw    .1                                ; Motore DX fermo
    movwf    buffer_pwm_dx_rx
    banksel
    movwf    CCPR2L
    movwf    CCPR2L
end_sx

    goto    end_funzione

mag_dx

    movfw    sharp_dx
    sublw    B'01100000'
    btfss   STATUS,C                            ; Se sharp > x
    goto    s1_dx

    movfw    sharp_dx
    sublw    B'00110000'
    btfss   STATUS,C                            ; Se sharp > x
    goto    s2_dx

    movlw    .23                                ; Motore SX fermo
    movwf    buffer_pwm_sx_rx
    banksel
    movwf    CCPR1L
    movlw    .23                                ; Motore DX fermo
    movwf    buffer_pwm_dx_rx
    banksel
    movwf    CCPR2L
    goto    end_dx

s2_dx

    movlw    .4                                ; Motore SX fermo
    movwf    buffer_pwm_sx_rx
    banksel
    movwf    CCPR1L
    movlw    .20                                ; Motore DX fermo
    movwf    buffer_pwm_dx_rx
    banksel
    movwf    CCPR2L
    goto    end_dx

s1_dx

    movlw    .1                                ; Motore SX fermo
```



```
movwf    buffer_pwm_sx_rx
banksel  CCPR1L
movwf    CCPR1L
movlw    .1                                ; Motore DX fermo
movwf    buffer_pwm_dx_rx
banksel  CCPR2L
movwf    CCPR2L

end_dx

goto     end_funzione
```



Allegato 17 – Listato funzione line-following

```
-----  
; LINE-FOLLOWER  
-----  
funz_line  
    movfw    sharp_dx  
    movlw    B'00000000'        ; MODALITA BINARIO  
    sublw    B'00100000'        ; Sogliabassa 000010011  
    btfss    STATUS,C  
    goto     FERMO  
    btfsc    PORTB,0            ; Se SX = 0 e DX = 0  
    goto     ELSELF1  
    btfsc    PORTB,2  
    goto     ELSELF1  
  
FERMO  
; ISTRUZIONI IFLF1 -----  
    movlw    .12                ; Motore SX fermo  
    movwf    buffer_pwm_sx_rx  
    banksel CCPR1L  
    movwf    CCPR1L  
    movlw    .12                ; Motore DX fermo  
    movwf    buffer_pwm_dx_rx  
    banksel CCPR2L  
    movwf    CCPR2L  
; -----  
  
GOTO     ENDIFLF1  
  
ELSELF1  
; ISTRUZIONI ELSELF1 -----  
    btfsc    PORTB,0            ; Se SX = 1 e DX = 0  
    goto     ELSELF2  
    btfss    PORTB,2  
    goto     ELSELF2  
  
; ISTRUZIONI IF2 -----  
    movlw    .8                ; Motore SX indietro  
    movwf    buffer_pwm_sx_rx  
    banksel CCPR1L  
    movwf    CCPR1L  
    movlw    .23                ; Motore DX avanti  
    movwf    buffer_pwm_dx_rx  
    banksel CCPR2L  
    movwf    CCPR2L  
; -----
```




Allegato 18 – Listato funzione minisumo

```
-----  
;  
;  
; MINISUMO  
;  
-----  
funz_minisumo  
    btfsc    PORTB,0          ; Se SX = 0 e DX = 0  
    goto    ENDIFMS1        ; Il robot è all'interno del  
cerchio  
    btfsc    PORTB,2  
    goto    ENDIFMS1  
  
    movfw   sharp_sx        ; Dato corretto  
  
    sublw   B'00001111'     ; Soglia bassa  
  
    btfsc    STATUS,C  
    goto    OSTACOLO_NO  
  
OSTACOLO_SI  
    ; ISTRUZIONI IFMS1 -----  
    movlw   .23              ; Motore SX avanti veloce  
    movwf   buffer_pwm_sx_rx  
    banksel CCPR1L  
    movwf   CCPR1L  
    movlw   .23              ; Motore DX avanti veloce  
    movwf   buffer_pwm_dx_rx  
    banksel CCPR2L  
    movwf   CCPR2L  
    ; -----  
  
    goto    ENDIFMS1  
  
OSTACOLO_NO  
    ; ISTRUZIONI IFMS1 -----  
    movlw   .23              ; Motore SX avanti veloce  
    movwf   buffer_pwm_sx_rx  
    banksel CCPR1L  
    movwf   CCPR1L  
    movlw   .1              ; Motore DX indietro veloce  
    movwf   buffer_pwm_dx_rx  
    banksel CCPR2L  
    movwf   CCPR2L  
    ; -----  
  
ENDIFMS1  
    btfss   PORTB,0          ; Se SX = 1 e DX = 1
```



```
goto      ENDIFMS2           ; Il robot ha trovato il bordo
btfss    PORTB,2
goto      ENDIFMS2

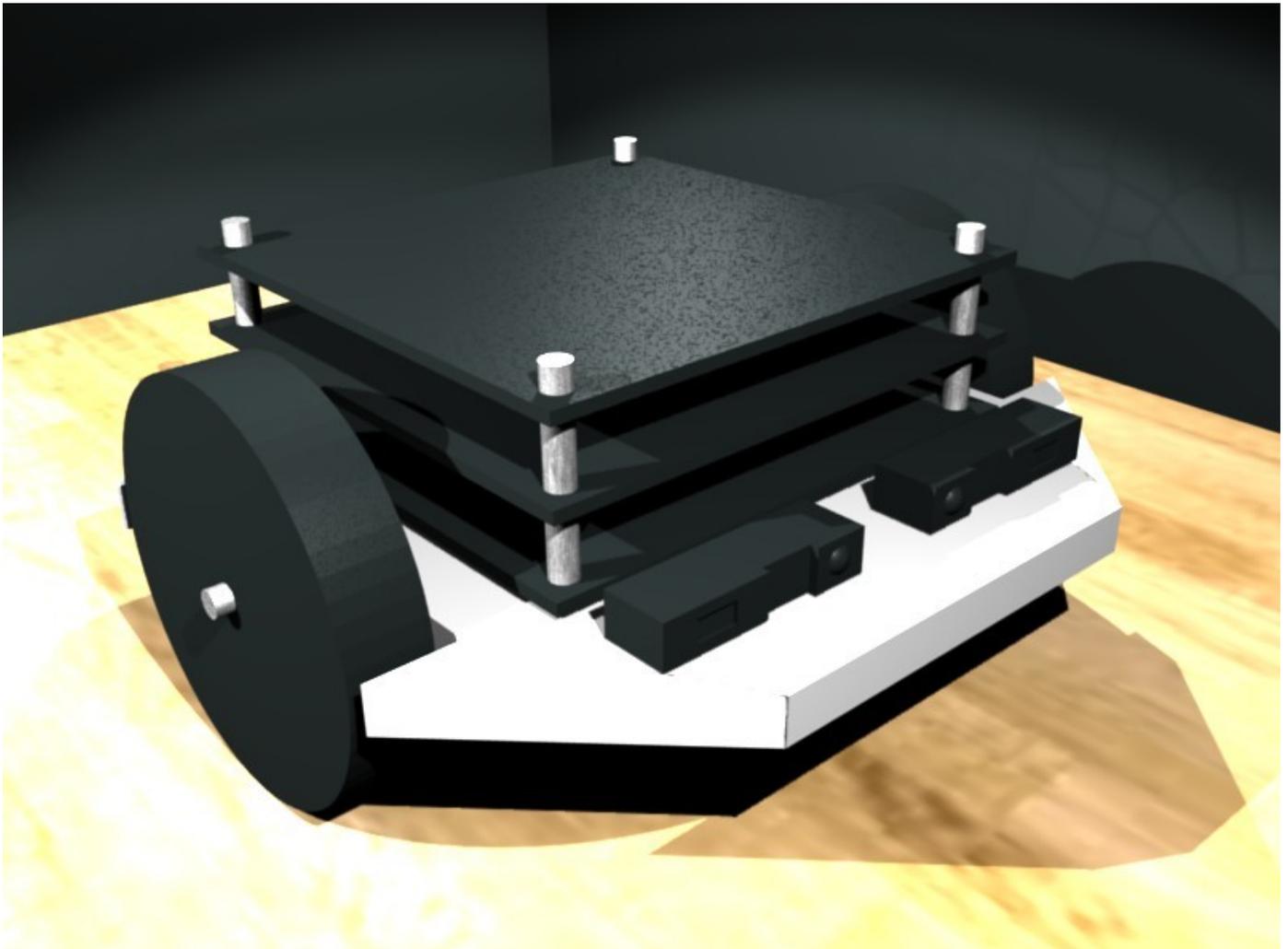
; ISTRUZIONI IFMS2 -----
movlw    .1                   ; Motore SX indietro veloce
movwf    buffer_pwm_sx_rx
banksel  CCPR1L
movwf    CCPR1L
movlw    .1                   ; Motore DX indietro veloce
movwf    buffer_pwm_dx_rx
banksel  CCPR2L
movwf    CCPR2L

call     ritardo
call     ritardo

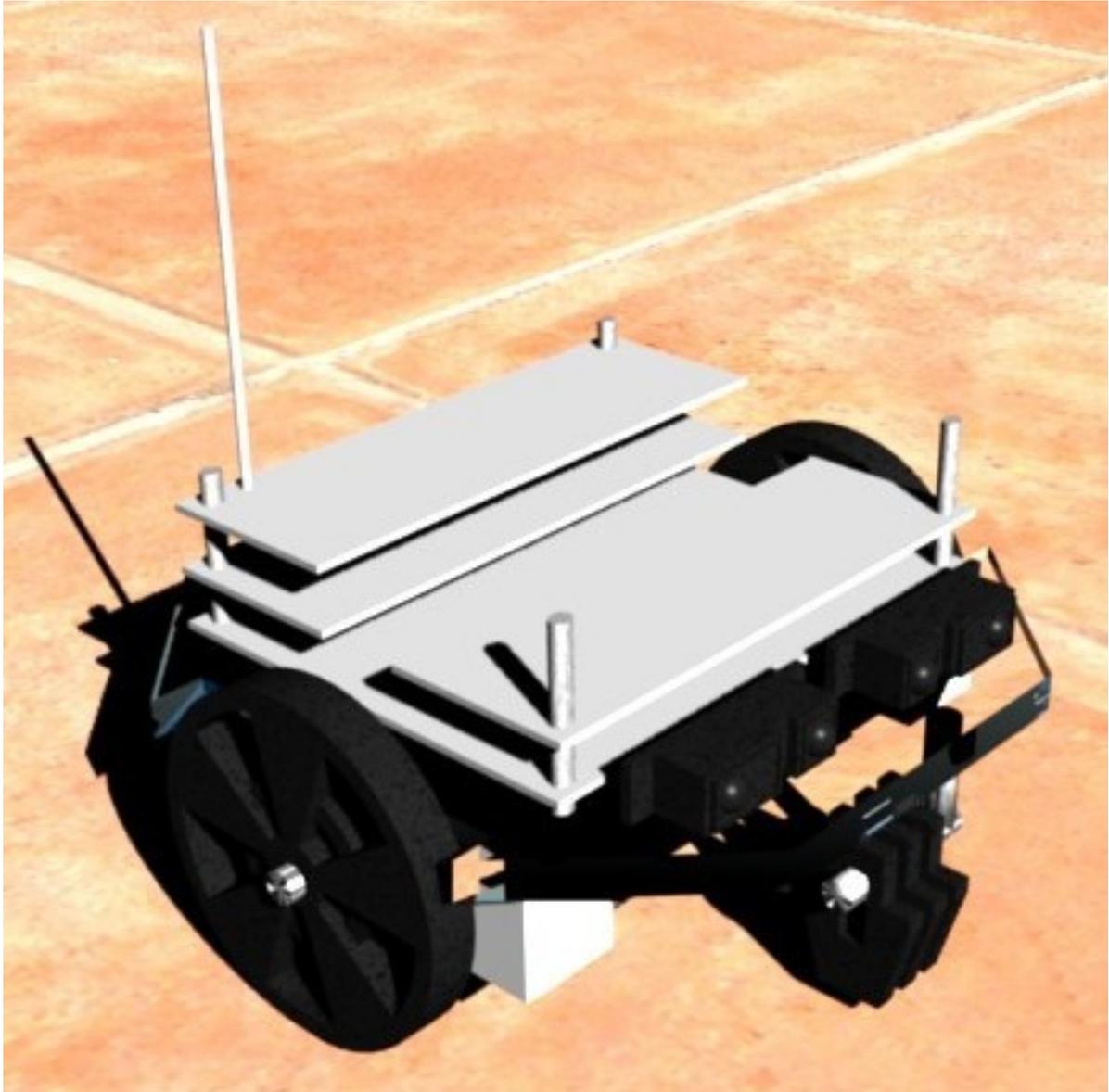
movlw    .23                  ; Motore SX avanti veloce
movwf    buffer_pwm_sx_rx
banksel  CCPR1L
movwf    CCPR1L
movlw    .1                   ; Motore DX indietro veloce
movwf    buffer_pwm_dx_rx
banksel  CCPR2L
movwf    CCPR2L

call     ritardo
call     ritardo
; -----
ENDIFMS2
goto     end_funzione
```

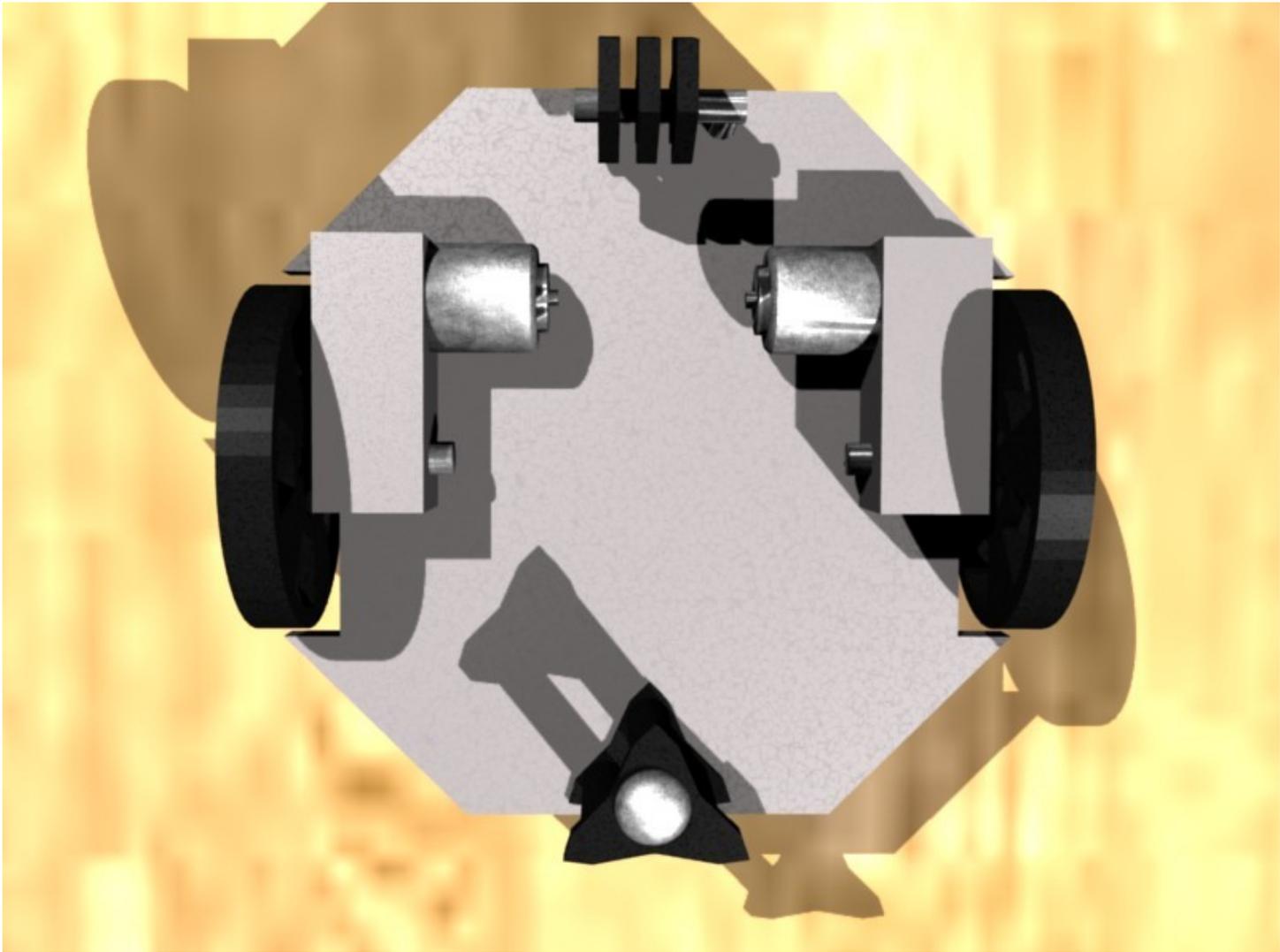

Allegato 20 – Robot 3D v1



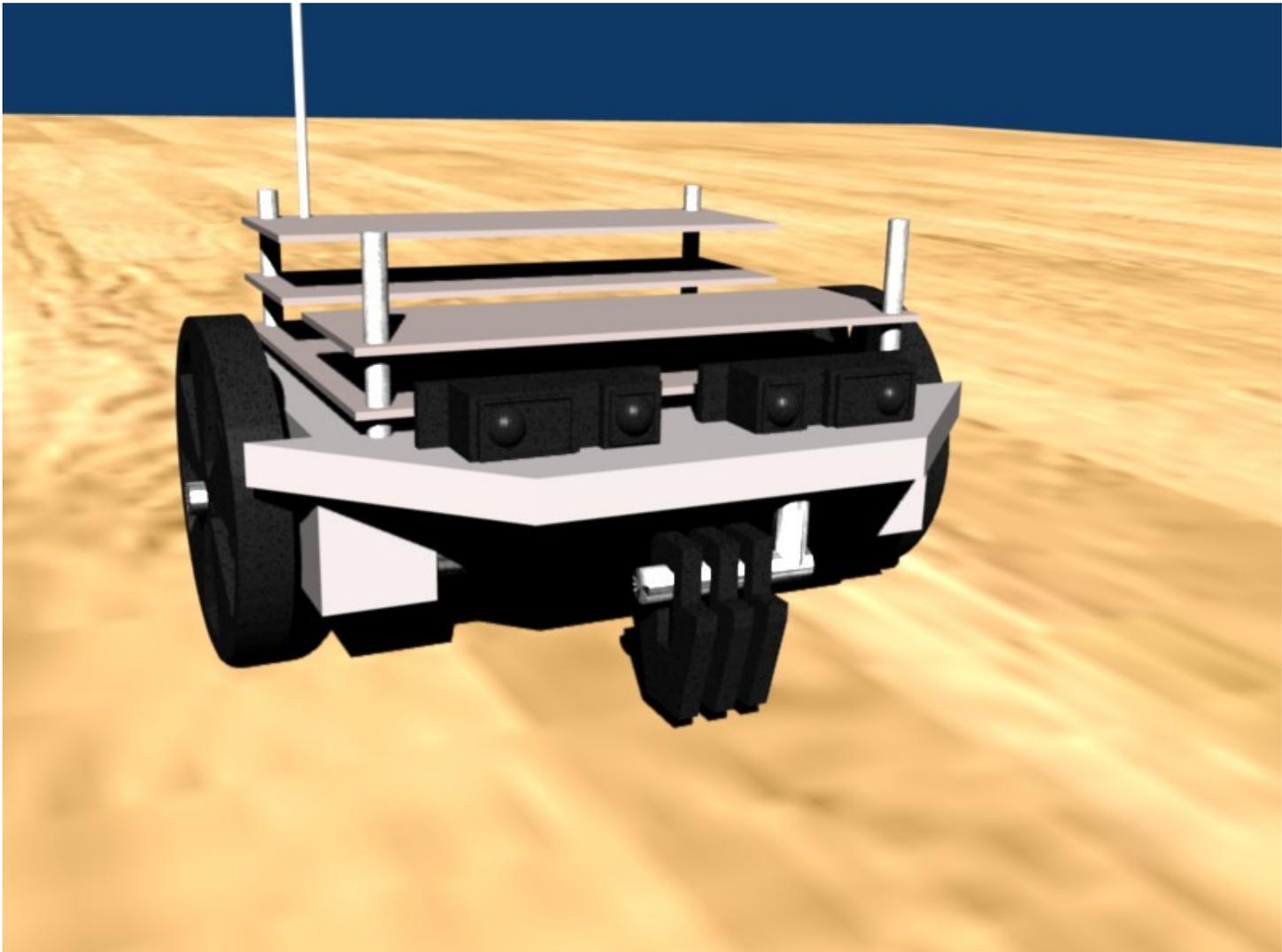
Allegato 21 – Robot 3D v2



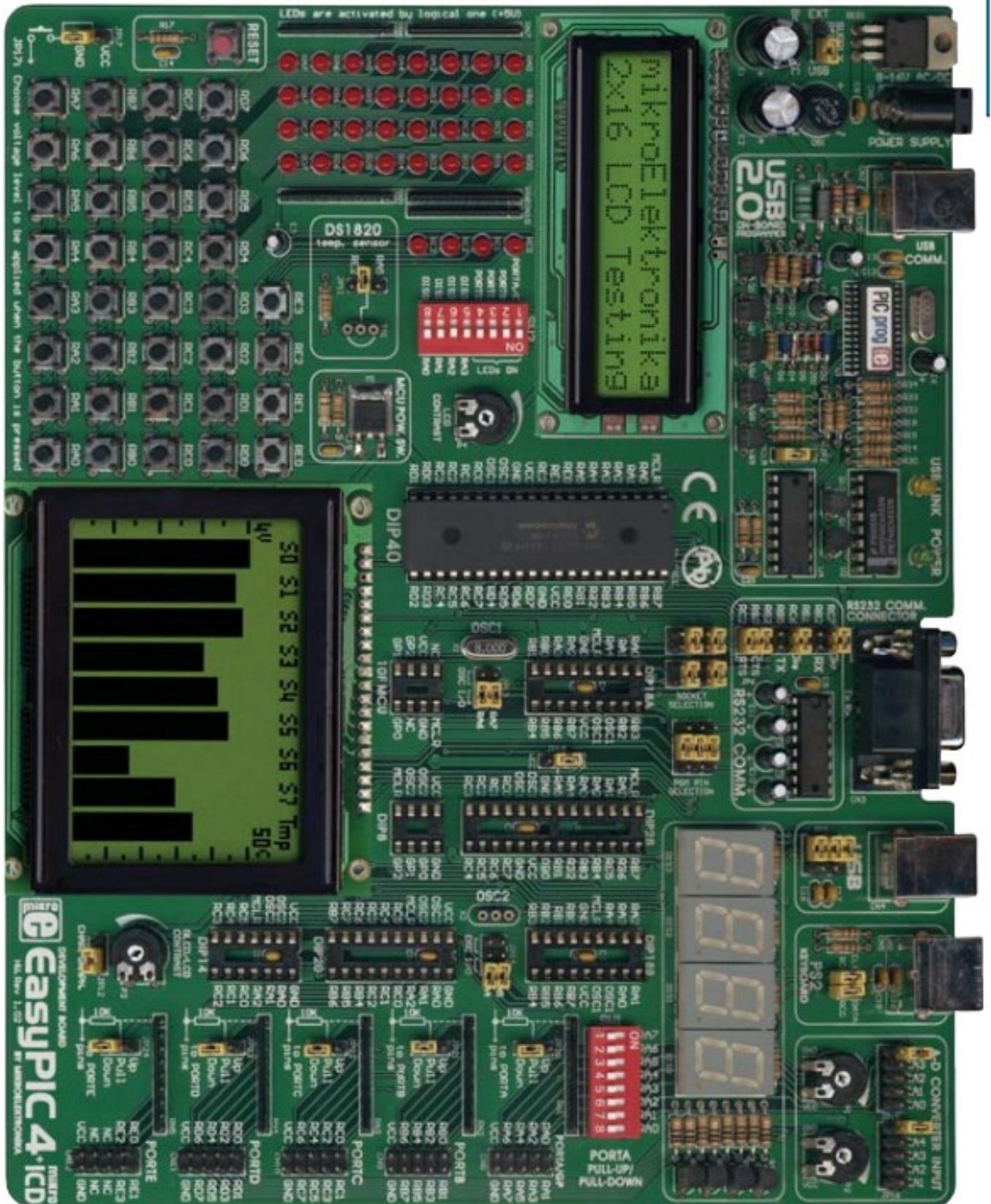
Allegato 22 – Robot 3D retro



Allegato 23 – Robot 3D v3



Allegato 24 – Scheda EasyPic 4



Allegato 25 – Datasheet TIP120

AMOSPEC

PLASTIC MEDIUM-POWER COMPLEMENTARY SILICON TRANSISTORS

...designed for general-purpose amplifier and low speed switching applications

FEATURES:

- * Collector-Emitter Sustaining Voltage-
 $V_{CE(sus)}$ = 60 V (Min) - TIP120, TIP125
 = 80 V (Min) - TIP121, TIP126
 = 100 V (Min) - TIP122, TIP127
- * Collector-Emitter Saturation Voltage
 $V_{CE(sat)}$ = 2.0 V (Max.) @ $I_C = 3.0$ A
- * Monolithic Construction with Built-in Base-Emitter Shunt Resistor

NPN	PNP
TIP120	TIP125
TIP121	TIP126
TIP122	TIP127

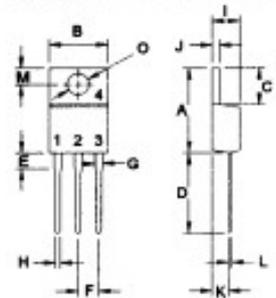
**5.0 AMPERE
 DARLINGTON
 COMPLEMENTARY SILICON
 POWER TRANSISTORS
 60-100 VOLTS
 65 WATTS**

MAXIMUM RATINGS

Characteristic	Symbol	TIP120 TIP125	TIP121 TIP126	TIP122 TIP127	Unit
Collector-Emitter Voltage	V_{CEO}	60	80	100	V
Collector-Base Voltage	V_{CBO}	60	80	100	V
Emitter-Base Voltage	V_{EBO}	5.0			V
Collector Current-Continuous -Peak	I_C I_{CM}	5.0 8.0			A
Base Current	I_B	120			mA
Total Power Dissipation @ $T_C = 25^\circ\text{C}$ Derate above 25°C	P_D	65 0.52			W W/ $^\circ\text{C}$
Operating and Storage Junction Temperature Range	T_J, T_{STG}	- 65 to +150			$^\circ\text{C}$



TO-220

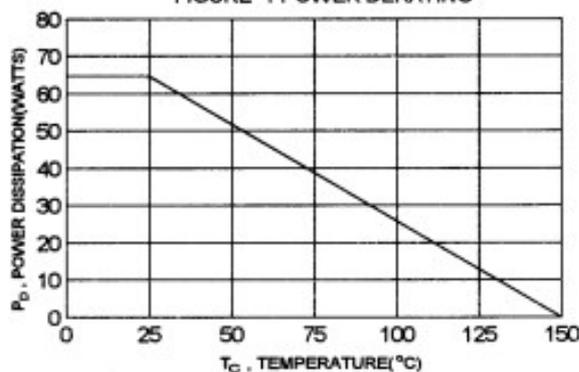


PIN 1.BASE
 2.COLLECTOR
 3.EMITTER
 4.COLLECTOR(CASE)

THERMAL CHARACTERISTICS

Characteristic	Symbol	Max	Unit
Thermal Resistance Junction to Case	$R_{\theta jc}$	1.92	$^\circ\text{C/W}$

FIGURE -1 POWER DERATING



DIM	MILLIMETERS	
	MIN	MAX
A	14.68	15.31
B	9.78	10.42
C	5.01	6.52
D	13.06	14.62
E	3.57	4.07
F	2.42	3.66
G	1.12	1.36
H	0.72	0.96
I	4.22	4.98
J	1.14	1.38
K	2.20	2.97
L	0.33	0.55
M	2.48	2.98
O	3.70	3.90

TIP120, TIP121, TIP122 NPN / TIP125, TIP126, TIP127 PNP

ELECTRICAL CHARACTERISTICS ($T_C = 25^\circ\text{C}$ unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit
----------------	--------	-----	-----	------

OFF CHARACTERISTICS

Collector - Emitter Sustaining Voltage (1) ($I_C = 30\text{ mA}$, $I_B = 0$)	TIP120, TIP125 TIP121, TIP126 TIP122, TIP127	$V_{CE(sus)}$	60 80 100	V
Collector Cutoff Current ($V_{CE} = 30\text{ V}$, $I_B = 0$) ($V_{CE} = 40\text{ V}$, $I_B = 0$) ($V_{CE} = 50\text{ V}$, $I_B = 0$)	TIP120, TIP125 TIP121, TIP126 TIP122, TIP127	I_{CEO}	0.5 0.5 0.5	mA
Collector Cutoff Current ($V_{CB} = 60\text{ V}$, $I_E = 0$) ($V_{CB} = 80\text{ V}$, $I_E = 0$) ($V_{CB} = 100\text{ V}$, $I_E = 0$)	TIP120, TIP125 TIP121, TIP126 TIP122, TIP127	I_{CBO}	0.2 0.2 0.2	mA
Emitter Cutoff Current ($V_{EB} = 5.0\text{ V}$, $I_C = 0$)		I_{EBO}	2.0	mA

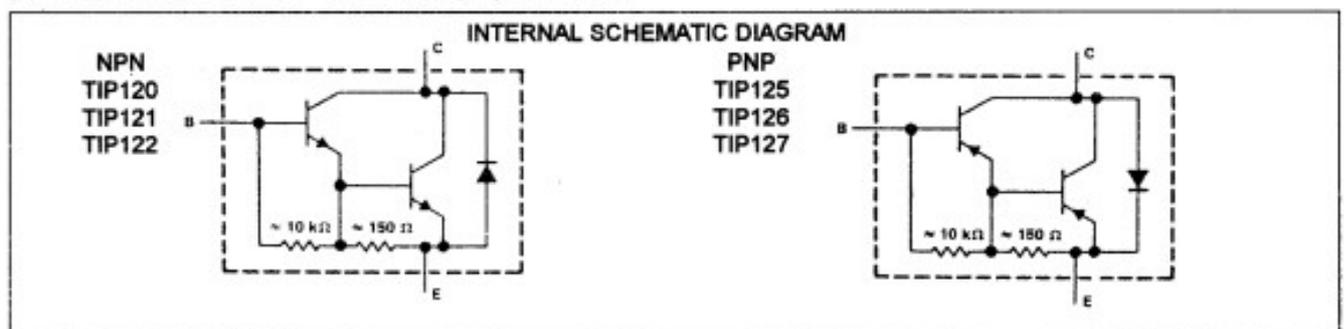
ON CHARACTERISTICS (1)

DC Current Gain ($I_C = 0.5\text{ A}$, $V_{CE} = 3.0\text{ V}$) ($I_C = 3.0\text{ A}$, $V_{CE} = 3.0\text{ V}$)		h_{FE}	1000 1000	
Collector-Emitter Saturation Voltage ($I_C = 3.0\text{ A}$, $I_B = 12\text{ mA}$) ($I_C = 5.0\text{ A}$, $I_B = 20\text{ mA}$)		$V_{CE(sat)}$	2.0 4.0	V
Base-Emitter On Voltage ($I_C = 3.0\text{ A}$, $V_{CE} = 3.0\text{ V}$)		$V_{BE(on)}$	2.5	V

DYNAMIC CHARACTERISTICS

Small-Signal Current Gain ($I_C = 3.0\text{ A}$, $V_{CE} = 4.0\text{ V}$, $f = 1.0\text{ MHz}$)		h_{fe}	4.0	
Output Capacitance ($V_{CB} = 10\text{ V}$, $I_E = 0$, $f = 0.1\text{ MHz}$)	TIP120, TIP121, TIP122 TIP125, TIP126, TIP127	C_{ob}	300 250	pF

(1) Pulse Test: Pulse width = 300 μs , Duty Cycle $\leq 2.0\%$





TIP120, TIP121, TIP122 NPN / TIP125, TIP126, TIP127 PNP

FIG-2 SWITCHING TIME

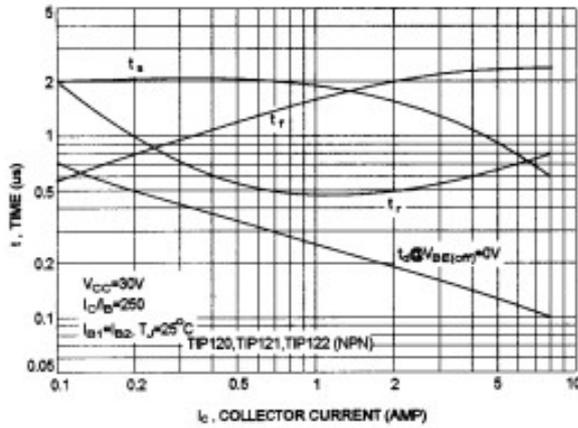


FIG-3 SWITCHING TIME

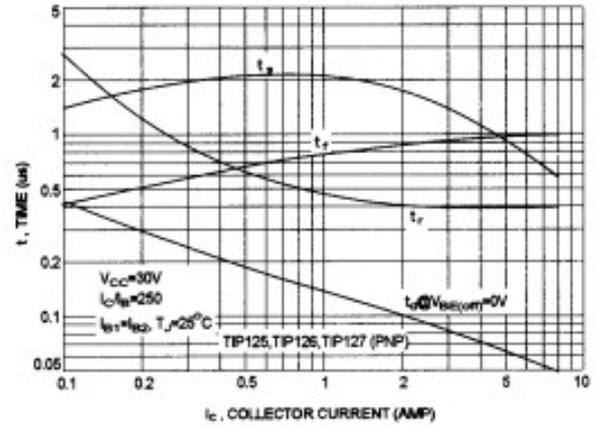


FIG-4 SMALL-SIGNAL CURRENT GAIN

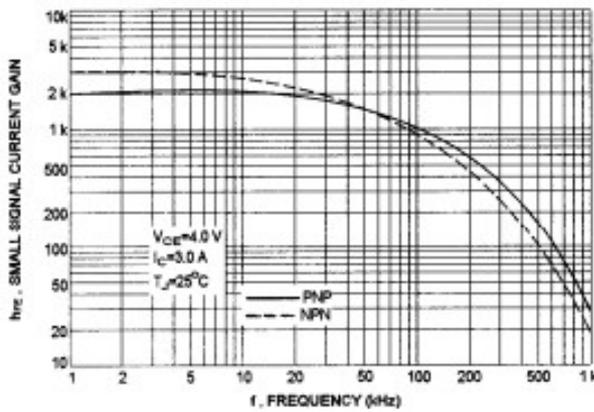


FIG-5 CAPACITANCES

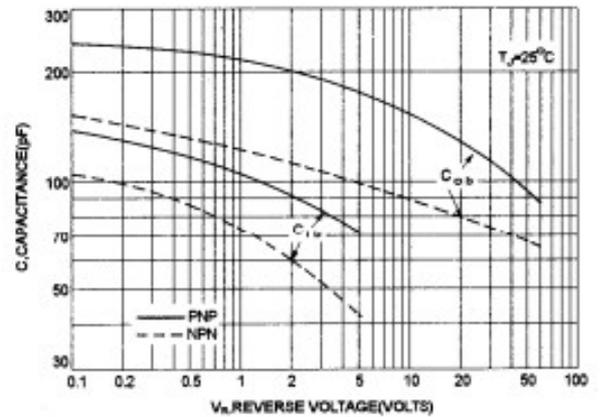
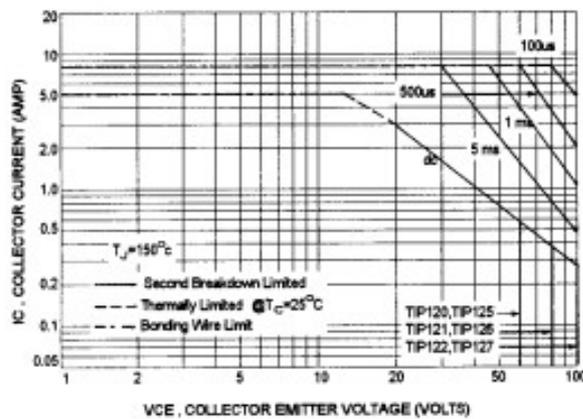


FIG-6 ACTIVE REGION SAFE OPERATING AREA



There are two limitation on the power handling ability of a transistor: average junction temperature and second breakdown. safe operating area curves indicate I_c - V_{CE} limits of the transistor that must be observed for reliable operation i.e., the transistor must not be subjected to greater dissipation than curves indicate.

The data of FIG-6 is base on $T_{jPK}=150^\circ C$; T_c is variable depending on power level. second breakdown pulse limits are valid for duty cycles to 10% provided $T_{jPK} \leq 150^\circ C$. At high case temperatures, thermal limitation will reduce the power that can be handled to values less than the limitations imposed by second breakdown.