

- Elettronica Open Source - <https://it.emcelettronica.com> -

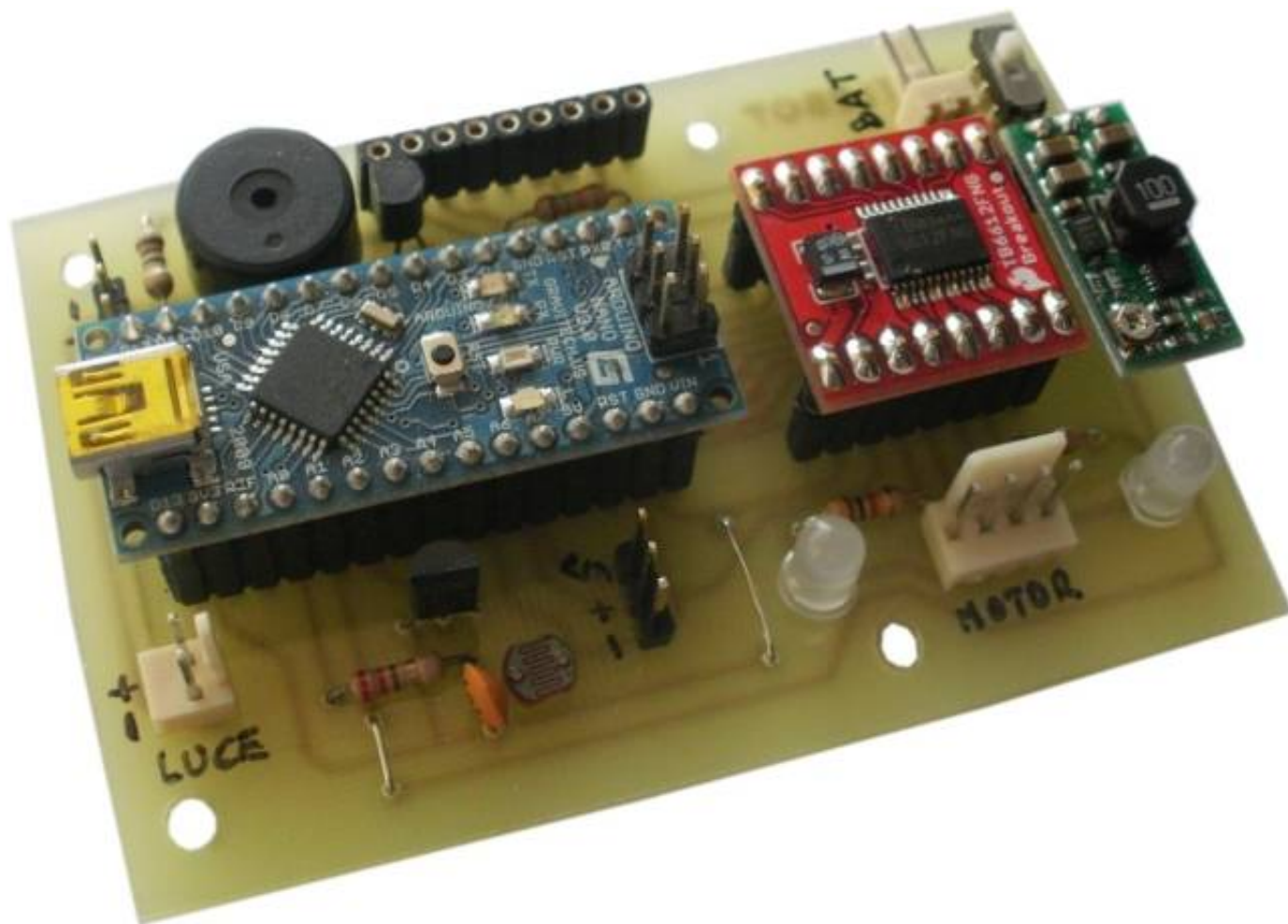
Costruzione del robot LittleBot – Scheda di controllo

Posted By *Adriano Gandolfo* On 4 febbraio 2014 @ 6:00 In [Arduino](#), [Robotica & Droni](#) | [17 Comments](#)

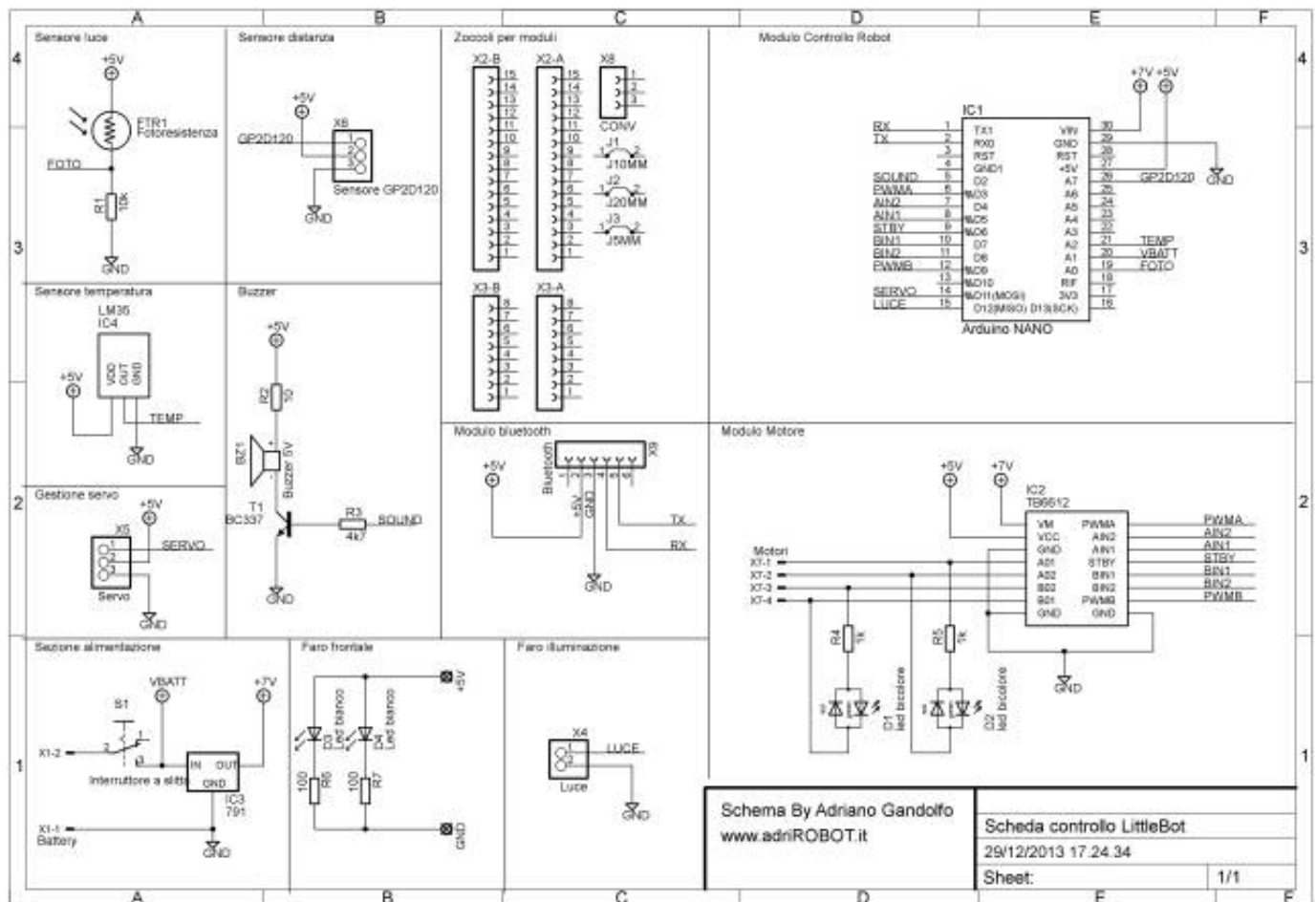


Nell'articolo precedente è stato presentato il robot LittleBot ^[1], si tratta di un progetto open source per la costruzione di un robot di piccole dimensioni equipaggiato con una scheda di controllo basata su Arduino Nano ^[2]. Il telaio è realizzato in alluminio, il gruppo motore è realizzato con due piccoli motoriduttori che muovono una coppia di cingoli. Il robot è dotato di alcuni sensori quali luminosità e temperatura, misuratore di distanza da un oggetto. Una volta montato il robot, potrà essere guidato tramite un collegamento wireless tipo bluetooth ^[3] effettuato con una scheda Arduino Esplora ^[4]. In quest'articolo, diviso in due parti, ne tratterò il montaggio: nella prima parte si analizzerà la scheda di controllo, mentre nella seconda passeremo all'assemblaggio vero e proprio del telaio, installazione della scheda ed eseguiremo un primo collaudo .

Il pezzo più importante del robot è rappresentato dalla sua scheda di controllo: a questa fa capo il pilotaggio dei motori, la lettura dei sensori e l'interfaccia per il pilotaggio in remoto del robot.



La scheda ha le dimensioni di 87x57mm e presenta quattro fori che ne permetteranno il fissaggio al telaio. Il suo schema, rappresentato nella figura sotto riportata, è anche presente negli allegati a quest'articolo, sia in formato PDF ^[5] che editabile con il programma EAGLE ^[6].



Come si può vedere, questo schema è suddiviso in varie sezioni distinte come: alimentazione, modulo comando, modulo motore, modulo bluetooth [7], sensore luce, sensore temperatura, sensore distanza, buzzer, faro illuminazione.

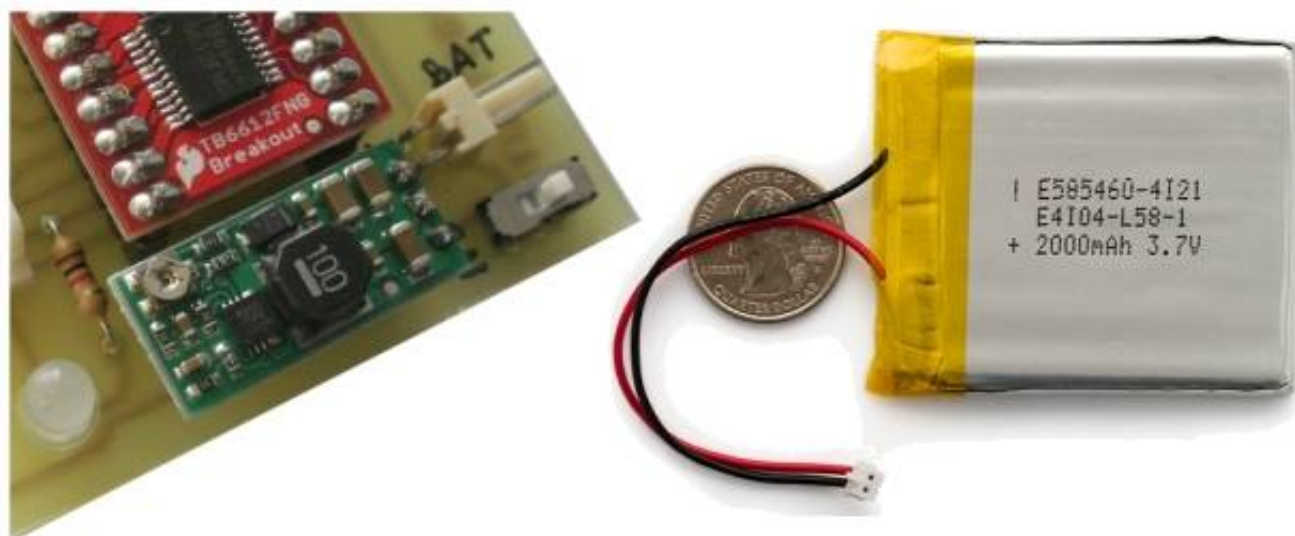
Analizziamo in dettaglio ogni gruppo.

Sezione alimentazione

L'alimentazione del robot è fornita da una batteria ricaricabile; la scelta è caduta su un accumulatore agli ioni di litio (a volte abbreviato in Li-Ion) analoga a quella utilizzata per il progetto Rendiamo autonoma la Arduino Esplora [8]; per maggiori informazioni su questa tipologia di batterie si può fare riferimento all'articolo Batterie al Litio: scegliamo le migliori [9].

Il modello utilizzato è una batteria Li-Ion con una tensione di 3,7V e una corrente di 1840 mA/h prodotta dalla ENIX Energies.

Si potrà comunque, utilizzare qualunque altra batteria con caratteristiche analoghe sia per tensioni/correnti sia per dimensioni, per fare in modo che possa essere posizionata sotto la scheda di controllo: per esempio una valida alternativa è la Polymer Lithium Ion Battery - 2000 mA/h [10] prodotta dalla **SparkFun** [11].



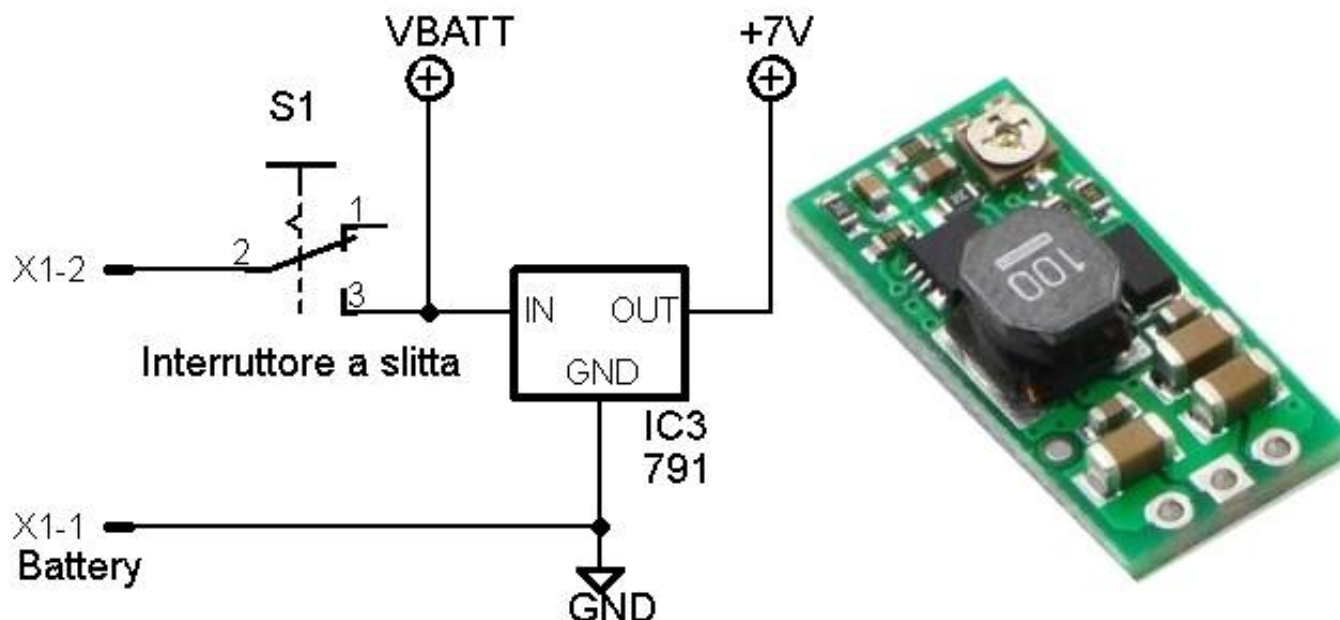
Tornando allo schema, vediamo che, dopo l'interruttore di accensione, abbiamo un primo stacco denominato **VBATT** che è connesso al **pin 20** del processore e che corrisponde all'ingresso della porta A/D **A1**.

In questo modo, il valore della tensione potrà essere monitorato e si potrà sapere in anticipo quando la batteria andrà ricaricata.

Dato che la tensione di 3,7V non è sufficiente per alimentare l'elettronica e i motori del robot, viene utilizzato un mini convertitore DC/DC siglato nello schema come **IC3**: il modulo è prodotto dalla **Pololu** ^[12] ed è analogo a quello già visto nell'articolo Rendiamo autonoma la Arduino Esplora ^[8] **codice 791** ^[13].

Ha dimensioni particolarmente ridotte (10,7 x 22,4 x 5,8 mm) ed è in grado di convertire una tensione continua compresa tra 1,5 e 16 volt in una tensione di uscita da 2,5 a 9,5 volt (regolabile tramite trimmer).

La massima corrente disponibile è di circa 800 mA.



La scheda si basa su un integrato **SC4501** ([datasheet](#) ^[14]) prodotto dalla **Semtech**, un regolatore switching di tipo step-up current-mode ad alta frequenza di commutazione con un transistor di potenza integrato da 2A.

La sua frequenza di commutazione elevata (programmabile fino a 2 MHz), consente l'utilizzo di piccoli componenti passivi esterni a montaggio superficiale.

Dispone di un Soft-start programmabile che elimina l'alta corrente di spunto in fase di avviamento.

La tensione d'uscita può essere regolata tramite un trimmer; in questo caso sarà regolato per un valore di circa **7-8V**.

Sulla scheda il modulo è montato in orizzontale utilizzando il connettore **X8**.

Sezione processore

La mente di tutto il robot è rappresentata da un **Arduino Nano 3.0** ([datasheet](#) ^[15]) che è una scheda Arduino prodotta dalla **GRAVITECH**, di piccolo formato basata sul Atmega328 ([datasheet](#) ^[16]).

Ha circa le stesse funzionalità della **Arduino UNO** ^[17], manca solo una presa di corrente continua e funziona con un cavo Mini-B USB al posto di uno standard.

Specifiche tecniche:

- Microcontrollore Atmel ATmega328;
- Tensione di funzionamento (livello logico) 5 V;
- Tensione di ingresso (consigliata) 7-12 V;
- Ingresso in tensione (limite) 6-20 V;

Digital I / O Pins 14 (di cui 6 forniscono PWM);
 8 Pin di ingresso analogico;
 Corrente fornita dai pin I/O: 40 mA;
 Flash 32 KB (di cui 2 KB utilizzati dal bootloader);
 SRAM 2 KB;
 EEPROM 1 KB;
 Velocità di clock 16 MHz;
 Dimensioni 18x43 mm.

Caratteristiche:

Ripristino automatico durante il download del programma;
 Led presenza tensione blu;
 Led Verde (TX), Led rosso (RX) e led arancione (L);
 Ingresso alimentazione rilevamento automatico/commutazione;
 Piccola mini-B USB per la programmazione e il monitor seriale;
 Connettore ICSP per la programmazione diretta;
 Spaziatura dei pin 2.54 tipo DIP;
 Pulsante di reset.

La sua alimentazione, oltre che tramite la connessione **Mini-B USB**, può essere fornita tramite il pin 30 in quanto sulla scheda è presente un regolatore **UA78M05** ([datasheet](#) [18]), che fornisce in uscita i 5V sul pin 27.

La fonte di alimentazione è selezionata automaticamente.

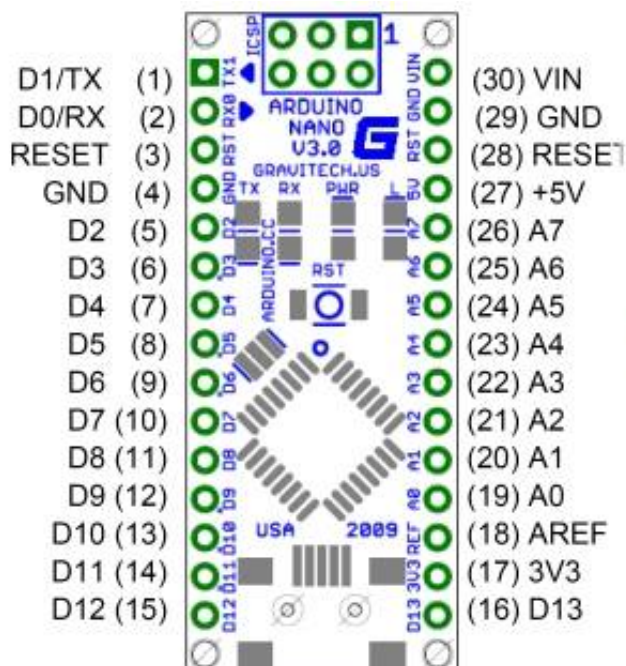
Sulla scheda di controllo, la tensione di alimentazione è derivata dal convertitore DC/DC e la tensione in uscita dal pin 27 viene poi utilizzata per alimentare l'elettronica dei sensori e del modulo motore. Alle porte I/O digitali e a quelle A/D di Arduino sono connessi i vari sensori e il modulo motore secondo questa tabella.

Pin Arduino	Porta	Tipo	Funzione
1	D1	Output Digitale	RX modulo Bluetooth
2	D0	Input Digitale	TX modulo Bluetooth
5	D2	Output Digitale	Comando buzzer
6	D3	Output Digitale	PWMA Modulo motore
7	D4	Output Digitale	AIN2 Modulo motore
8	D5	Output Digitale	AIN1 Modulo motore
9	D6	Output Digitale	STBY Modulo motore
10	D7	Output Digitale	BIN1 Modulo motore
11	D8	Output Digitale	BIN2 Modulo motore
12	D9	Output Digitale	PWMB Modulo motore
14	D11	Output Digitale	Comando Servo
15	D12	Output Digitale	Comando Luce

16	D13	Output Digitale	Led su Arduino Nano
19	A0	Input Analogico	Lettura fotoresistenza
20	A1	Input Analogico	Tensione batteria
21	A2	Input Analogico	Lettura temperatura
26	A7	Input Analogico	Lettura distanza

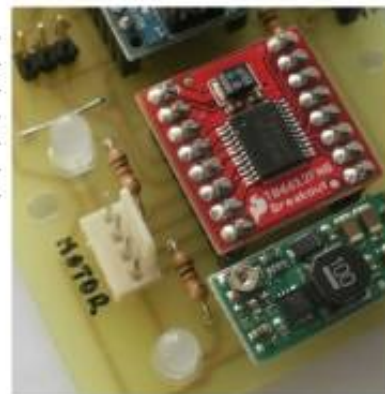
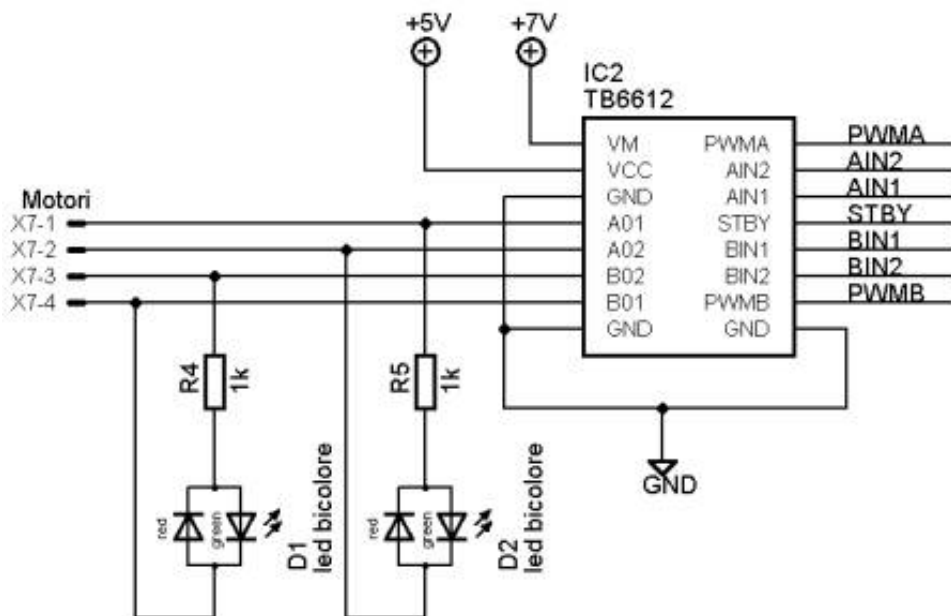
Per la programmazione del modulo, sarà utilizzata la Mini-B USB.

La presenza della tensione di alimentazione sarà segnalata dall'accensione del **led blu**, mentre il traffico tramite la linea seriale, sarà monitorato tramite i led TX e RX (colore **led verde e rosso**) presenti sulla scheda Arduino Nano.

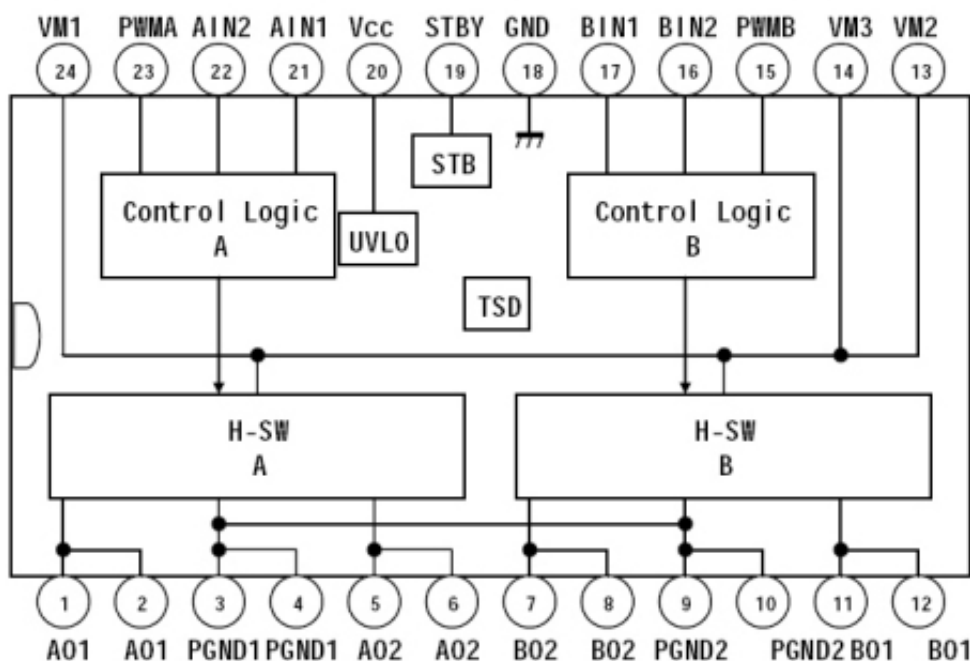


Sezione modulo motore

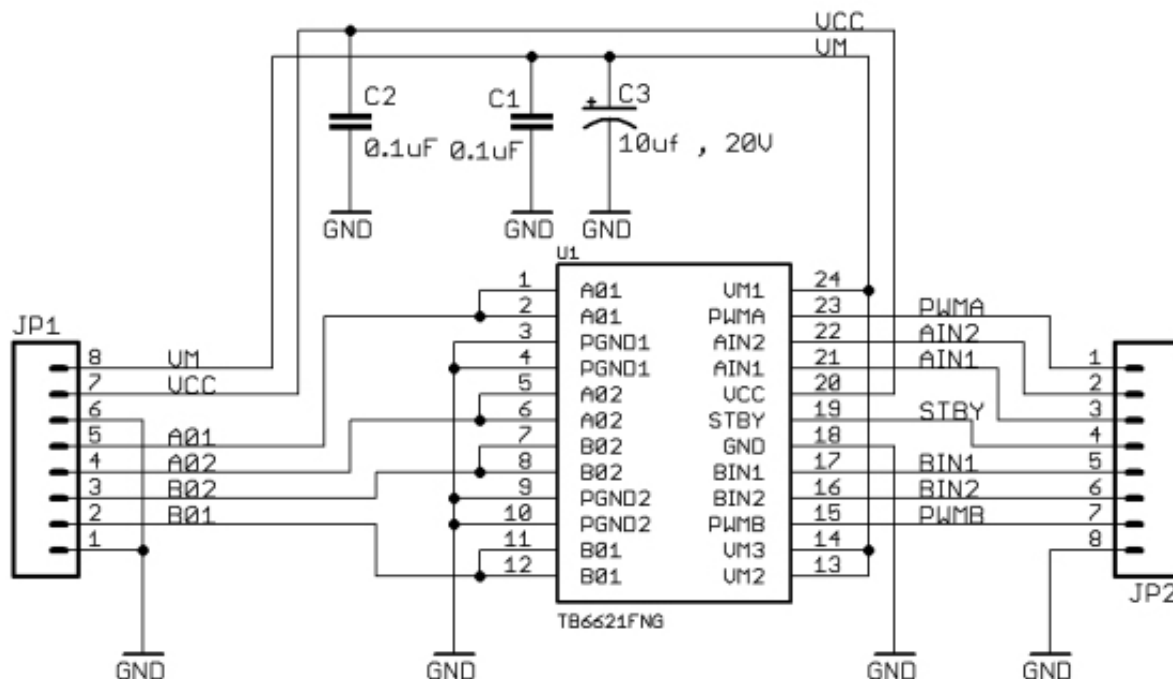
La sezione inerente al controllo dei motori è basata sull'integrato **TB6612FNG** ([datasheet](#) [19]) prodotto dalla **Toshiba** [20] ed è in grado di pilotare due motori CC con una corrente costante di 1.2A (3.2A di picco) interfacciandosi con un microcontrollore.



L'integrato è un H-bridge di tipo MOSFET molto più efficiente rispetto a un ponte H-bridge basato su BJT, il che consente di avere più corrente per alimentare i motori e meno da trarre dalla rete logica.



L'integrato è montato su una piccola breakout board ([schema elettrico](#) ^[21]) che misura solamente 20.3x20,3mm prodotta dalla **SparkFun** (**Motor Driver 1A dual TB6612FNG** ^[22]) il che permette l'accesso diretto a tutte le caratteristiche del TB6612FNG, sono inoltre presenti i condensatori di alimentazione e i condensatori di disaccoppiamento su entrambe le linee di alimentazione.



Ognuno dei due canali motore ha due pin di controllo direzione e un pin di controllo della velocità che accetta un ingresso PWM con una frequenza fino a 100 kHz.

Il pin STBY deve essere posto a livello alto per porre il driver in modalità standby.

Due segnali di ingresso (IN1 e IN2) possono essere utilizzati per comandare il motore in una delle quattro modalità di funzionamento - rotazione oraria CW, rotazione antioraria CCW, a freno rapido e stop.

La tensione di alimentazione della logica (VCC) può essere nel range di 2.7-5.5VDC, nella nostra scheda è derivata da quella fornita da Arduino Nano, mentre l'alimentazione del motore (VM) è limitata a una tensione massima di 15VDC, nel nostro caso +7V fornita dal modulo DC/DC. Ecco un riassunto delle caratteristiche del modulo utilizzato:

Tensione di alimentazione: 15 V max = VM, VCC = 2.7-5.5V;

Corrente di uscita: 1,2 A (medio) / 3.2A (picco);

Controllo standby per risparmiare energia;

CW / CCW / short freno / stop modalità di controllo motore;

Built-in circuito di shutdown termico e circuito di rilevazione bassa tensione;

Spaziatura dei pin 2,5 mm;

Condensatori di filtraggio su entrambe le linee di alimentazione;

Dimensioni: 20.3x20,3mm.

Per segnalare il senso di rotazione del motore sono presenti due led bicolore, ognuno con la propria resistenza di limitazione.

Il modulo andrà collegato sulle pin strip **X3A** e **X3B**, mentre i motori andranno collegati al connettore **X7**.

Per quanto riguarda i motoriduttori, per ridurre lo scintillio dovuto allo sfregamento delle spazzole sul collettore, che può causare interferenze con la circuiteria elettronica, sono stati

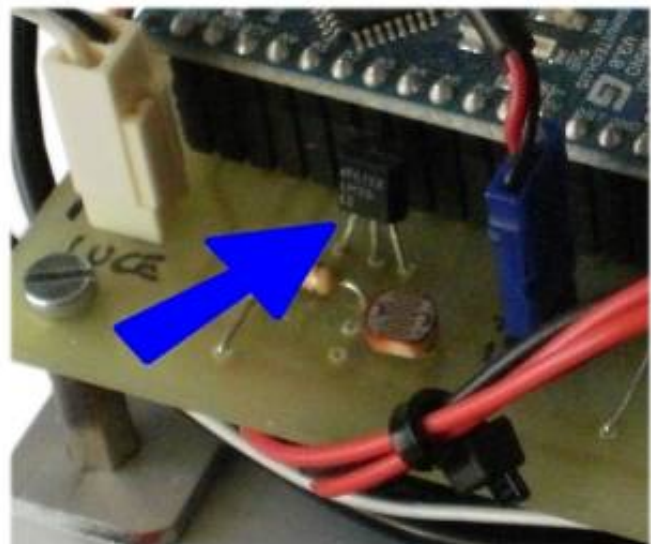
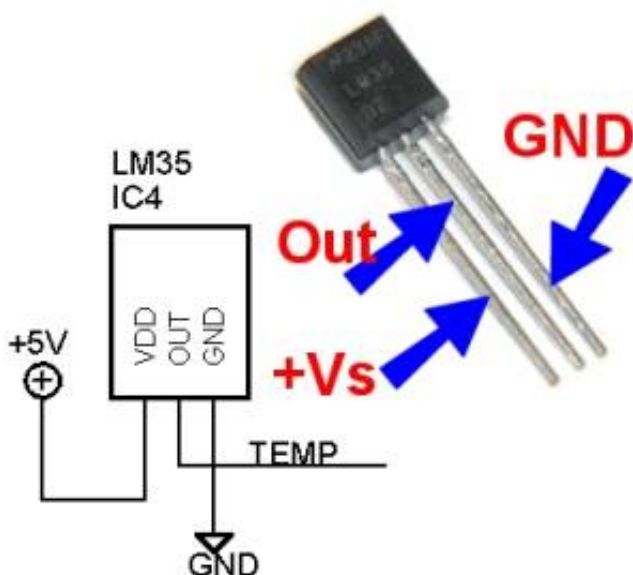
dotati di un condensatore da 100nF tra positivo e negativo dell'alimentazione.



Sensore di temperatura

Il sensore di temperatura è rappresentato dall'integrato **LM35** ([datasheet](#) [23]) prodotto dalla **National Semiconductor** [24], questo si presenta un package di tipo **T092** analogo a quello di un normale transistor e necessita solo del collegamento del pin d'uscita alla porta A/D **A2** di Arduino NANO per la lettura del valore di tensione fornito dal sensore.

La tensione in uscita, proporzionale alla temperatura rilevata, è pari a 10 mV per ogni grado centigrado.



Caratteristiche:

Precisione: $\pm 0.6^{\circ}\text{C}$;

Sensibilità: $10\text{mV}/^{\circ}\text{C}$;

Temperatura massima: 100°C ;

Temperatura minima: 0°C;

Tensione tipica di funzionamento: 4-30V;

Tipo uscita: Analogica.

Per altre informazioni vedere anche l'articolo: "[Arduino ed i sensori di temperatura](#) [25]"

Sensore di luce

Il sensore di luce è formato da una semplice fotoresistenza; questi dispositivi sono sensibili alle radiazioni luminose e sono costituiti da materiali semiconduttori leggermente drogati come Solfuro di Cadmio (CdS) , Solfuro di Piombo (PbS) Selenio (Se) e antimoniuro di indio (InSb).

Il loro principio di funzionamento è basato sull'effetto fotoconduttivo.

Quando la superficie sensibile del fotoresistore (di solito chiuso in involucri protettivi trasparenti) viene esposta alla luce, l'energia raggiante assorbita provoca la rottura dei legami covalenti e quindi l'aumento delle coppie lacune-elettroni rispetto a quelle generate per effetto termico.

Data la struttura fisica, si comprende come questi non siano quasi mai elementi di potenze elevate; valori caratteristici della massima potenza dissipabile sono sui 50mW per le più piccole, circa 1 W per le più grosse.

Le fotoresistenze sono caratterizzate dalla curva di sensibilità, cioè dal colore della luce al quale sono maggiormente sensibili e dai valori della resistenza al buio e alla luce forte, dette valore di buio e valore di luce; si ha indicativamente:

valore di buio: qualche Mohm;

valore di luce: intorno al kohm.

Il campo di variabilità è quindi molto elevato, la loro caratteristica di illuminazione è definita dall'equazione

$$R = A \cdot L^{-\alpha}$$

dove:

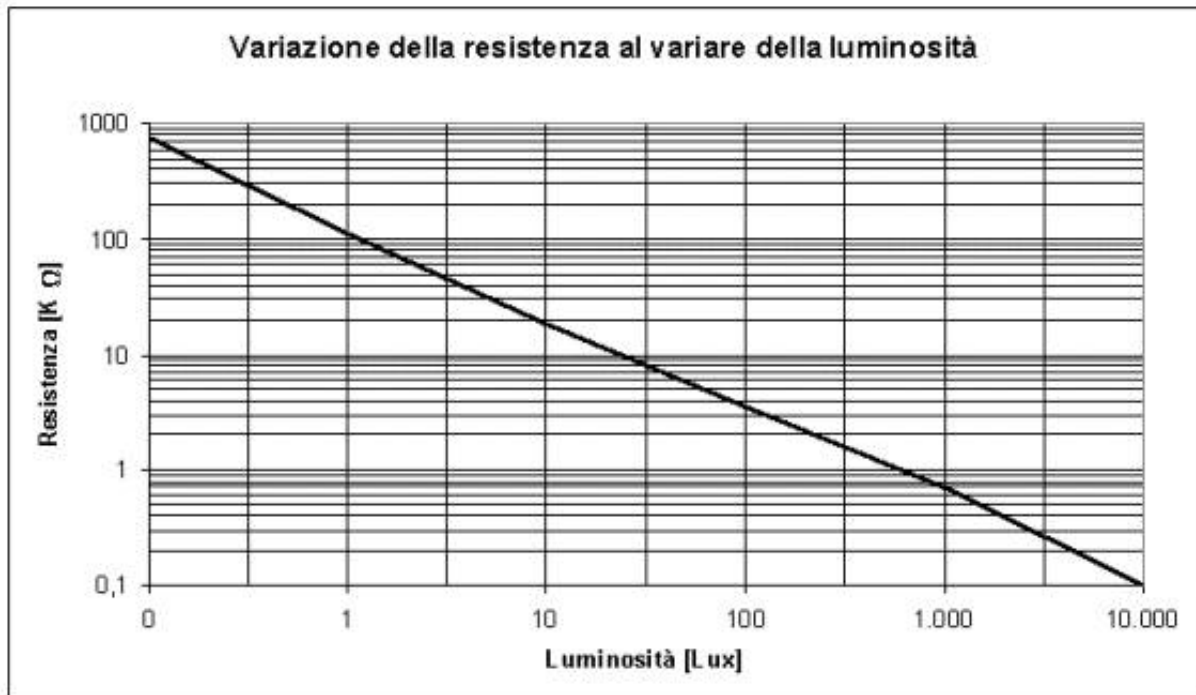
R è la resistenza del componente in ohm

A è la resistenza in condizioni di illuminamento unitario in ohm

L è l'intensità della radiazione luminosa espressa in lux

α è una costante che varia tra 0.7 e 0.9.

Quindi, date la resistenza di buio e di luce, si può tracciare una caratteristica rettilinea (in scala logaritmica) che approssima abbastanza bene quella reale (vedi grafico).



Bisogna però tener presente che questi elementi sono "lenti" (variazione di circa 200 Kohm/s) cioè se la luce varia rapidamente non è detto che il valore della resistenza la segua con la stessa legge.

In commercio si possono trovare diversi modelli di fotoresistenza, quella utilizzata in questo progetto ha queste caratteristiche:

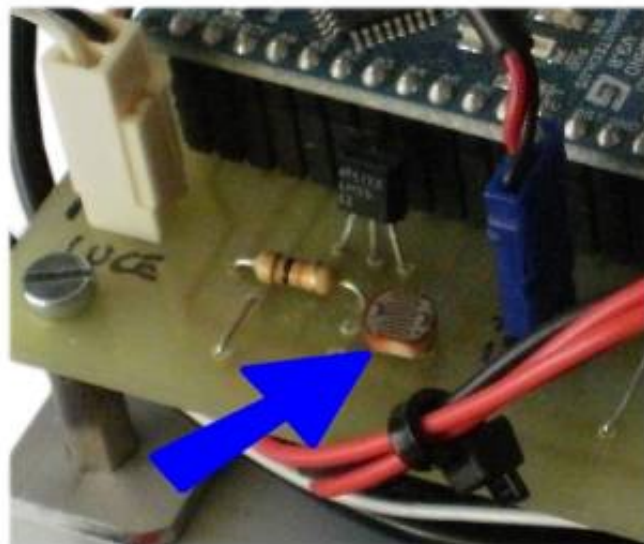
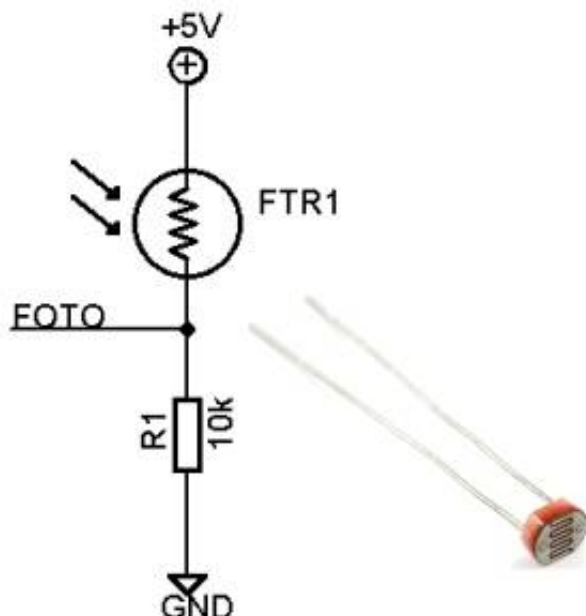
- Resistenza (sensore illuminato) 1 kohm;
- Resistenza (sensore non illuminato) 10 kohm;
- Massima tensione di lavoro 150 Vcc;
- Massima potenza dissipabile 100 mW;
- Dimensioni 2x4x5mm;
- Interasse Pins 4mm.

Nel circuito proposto, nel caso ideale, le due resistenze, di eguale valore, collegate come nello schema, dividerebbero in parti uguali la tensione, nel punto di giunzione e si misurerebbe una tensione che sarebbe la metà di quella di alimentazione.

Nel circuito reale uno dei due elementi è variabile e varia in funzione della luce che lo illumina.

Quando la fotoresistenza è colpita dalla massima luce, al punto di giunzione si troverà la massima tensione +5V; quando ci sarà buio, invece, si otterrà $V/2 = 5v / 2 = 2,5v$.

La tensione di uscita è misurata tramite una porta **A0** di Arduino Nano.



Sensore di distanza

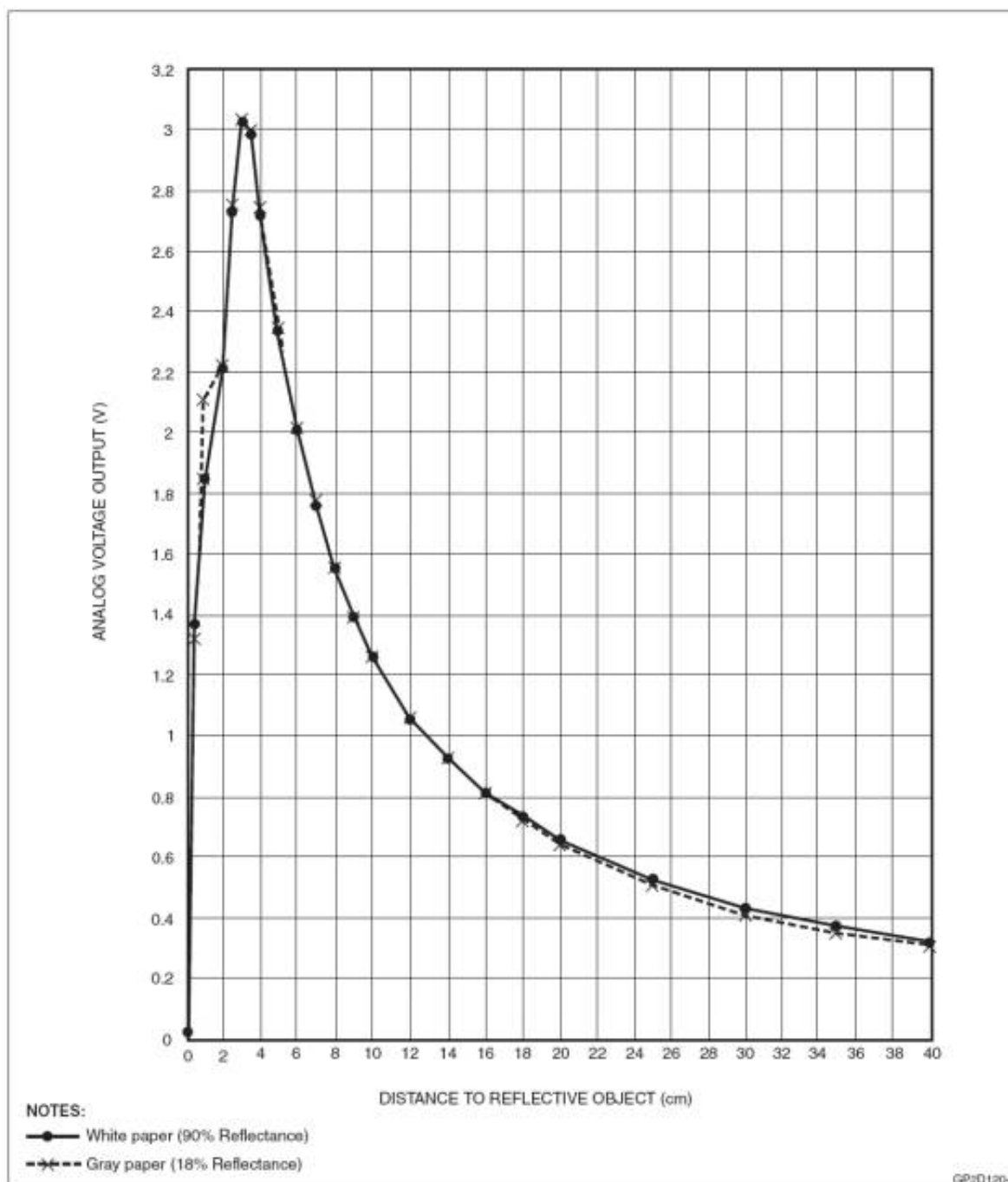
Il sensore frontale per la misurazione della distanza di un'eventuale ostacolo frontale è rappresentato da un sensore **GP2D120X** ([datasheet](#) ^[26]) prodotto dalla **Sharp** ^[27], questi appartiene ad una **categoria di sensori** ^[28] che hanno sensibilità diverse.

Questi sensori sono di tipo ad infrarosso, il prodotto non è stato concepito espressamente per la robotica ma viene utilizzato principalmente nelle fotocopiatrici per rilevare la presenza di carta.

È uno dei migliori sensori di distanza in circolazione, è abbastanza preciso, ha un costo abbordabile ed è facilmente interfacciabile, dispone infatti di soli 3 piedini.

Per collegarlo è sufficiente avere una porta con un convertitore analogico/digitale, in quanto in uscita abbiamo una tensione proporzionale (o quasi) alla distanza.

L'output del sensore non è lineare, ma la sua curva caratteristica è più simile ad un'iperbole.



Questi sensori hanno una buona immunità alla luce ambientale e possono per questo, possono essere utilizzati in condizioni di luce differenti ad eccezione di ambienti esterni molto luminosi.

Il sensore potrà essere utilizzato al posto di un sensore di contatto, il modello utilizzato nel progetto è in grado di rilevare un ostacolo ad una distanza compresa tra **4 e 30 cm**.

Come funziona il sensore

Il sensore **GP2D12**, contrariamente a quanto fanno i normali sensori ad infrarosso, non misura la quantità di infrarossi riflessi dalla superficie colpita, ma bensì funziona con il principio della triangolazione.

Un impulso di luce IR viene emessa dall'emettitore.

Questa luce viaggia nel campo visivo e colpisce un oggetto.

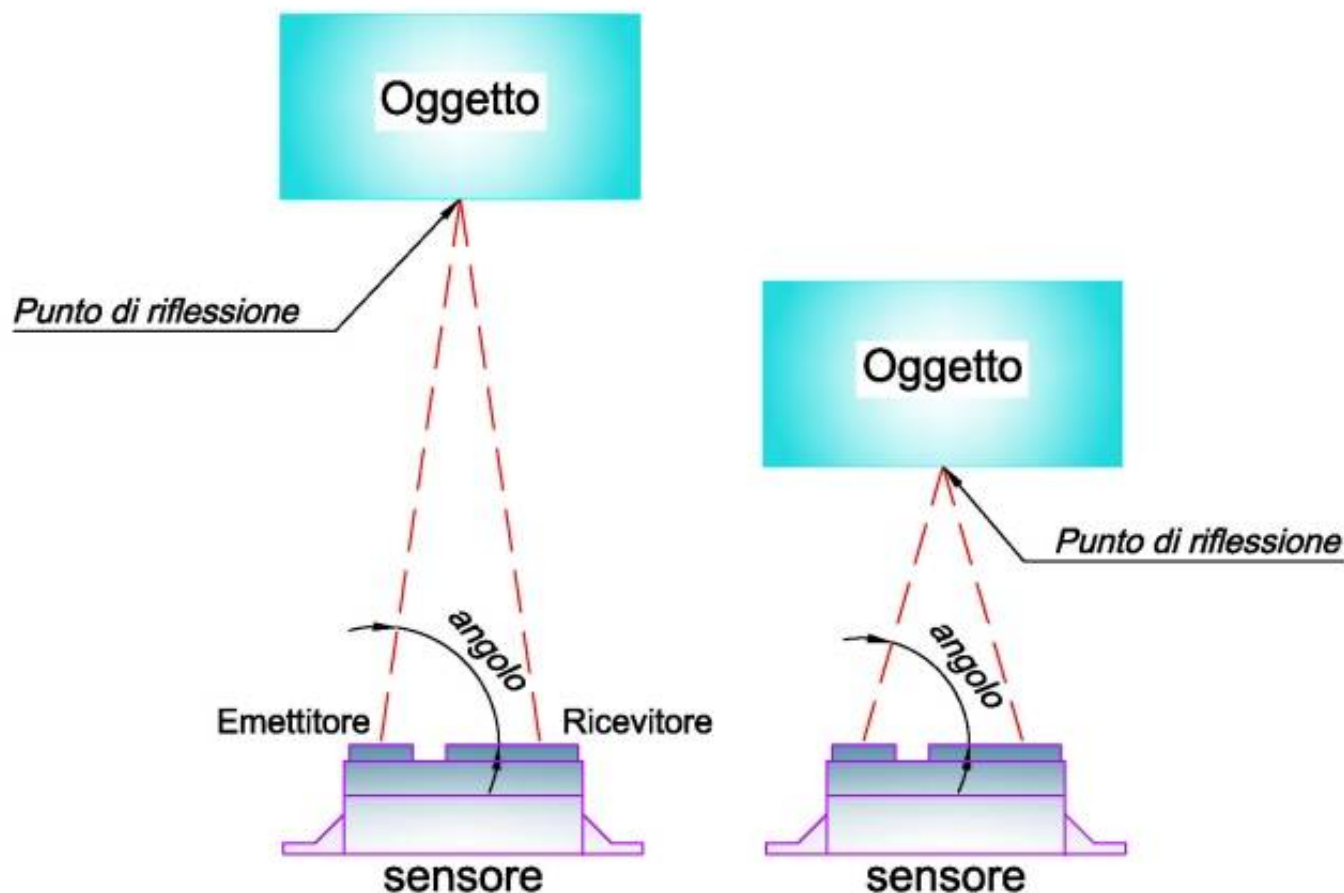
La luce riflessa ritorna al rivelatore e crea un triangolo tra il punto di riflessione, l'emettitore

e il rilevatore.

Gli angoli di questo triangolo variano in base alla distanza dall'oggetto.

Il ricevitore utilizza un obiettivo di precisione per trasmettere la luce riflessa su varie porzioni del CCD lineare in base all'angolo del triangolo descritto sopra.

La matrice CCD può quindi determinare l'angolo della luce riflessa di ritorno e quindi, si può calcolare la distanza dell'oggetto.



Caratteristiche:

Uscita: analogica

Gamma di rilevazione :4-30cm

Tensione di alimentazione: 4.5 ~ 5.5V

Corrente assorbita: 33mA

Dimensioni: 40x20x13.5 mm

Il sensore è connesso alla scheda di controllo tramite il connettore **X6** che è connesso, a sua volta, alla porta A7 (ingresso analogico) di Arduino Nano.

La libreria per la gestione del sensore da parte di Arduino è scaricabile al seguente [link](#) ^[29].

Nota: per migliorare il funzionamento del sensore è consigliabile collegare un condensatore elettrolitico da 10 uF tra GND e Vcc, direttamente sul sensore stesso, al fine di ridurre il rumore sull'alimentazione causata dalla corrente richiesta dal LED emettitore.

È invece, da evitare l'uso di un condensatore tra l'uscita del segnale e GND o Vcc, ciò può

ridurre drasticamente la dinamiche del sensore (filtro passa basso).



Buzzer

Sulla scheda di controllo è presente un buzzer per l'emissione di segnali acustici.

Il Buzzer (in italiano cicalino) è un dispositivo audio di segnalazione, che può essere di tipo meccanico, elettromeccanico o piezoelettrico.

Gli usi tipici dei cicalini sono i cercapersone, gli allarmi, i timer e la conferma di input dell'utente, come un clic del mouse o della tastiera.

Il cuore dei segnalatori acustici piezoelettrici è un semplice disco piezoelettrico, consistente di una lastra ceramica e di uno strato metallico applicato.

Quando il disco viene pilotato da un circuito oscillatorio, si parla di un trasduttore piezoelettrico, quando il circuito oscillatorio è incorporato nella cassa si parla di un buzzer piezoelettrico.

Il vantaggio di questa costruzione semplice è la robustezza e la produzione a basso costo.

I buzzer elettromagnetici, nel loro principio di funzionamento, corrispondono ai trasduttori elettromagnetici, dispongono comunque di un'elettronica incorporata.

Consistono in una membrana metallica, la quale viene messa in vibrazione per mezzo di un circuito elettromagnetico. Il circuito oscillatorio incorporato permette un comando tramite corrente diretta. Quello utilizzato nel progetto è del tipo piezoelettrico denominato **PK-12N40PAQ** ([datasheet](#) ^[30]) e presenta le seguenti caratteristiche:

Tensione di alimentazione (V): 3÷15;

Consumo di corrente (mA): 10 max;

Livello di pressione sonora (a 10cm): 83dB;

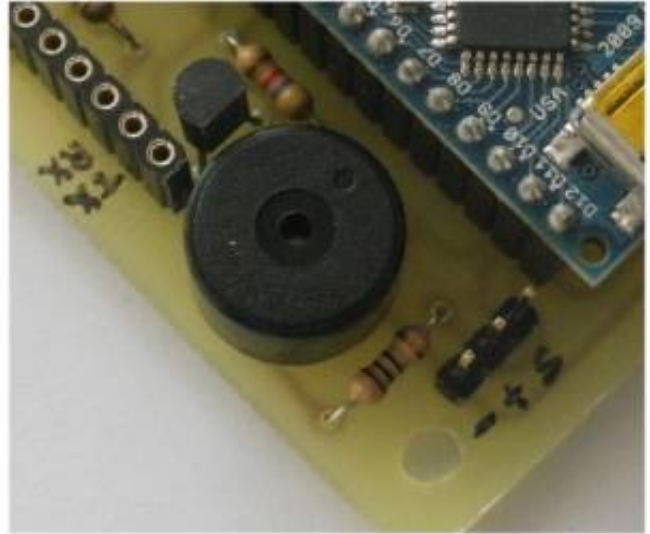
Frequenza di oscillazione: 4100 Hz;

Temperatura operativa (°C): -20~ + 90;

Peso: 0,8 g.

Per non sovraccaricare la porta dei processori, il buzzer è pilotato tramite un transistor **BC337** ([datasheet](#) ^[31]), la cui base è alimentata tramite una resistenza di limitazione R6 da

4,7 k Ω ; sulla linea di alimentazione del buzzer è presente un'ulteriore resistenza di limitazione R7 da 10 Ω .



Modulo luce (Faro)

Il modulo luci è formato da un circuito stampato a singola faccia in cui sono presenti 2 led bianchi luce ad alta intensità, mentre sul lato rame sono saldate 2 resistenze SMT per la limitazione della corrente.

Caratteristiche:

Tipo led: Led bianco ad altissima luminosità;

Contenitore: standard 5mm (T1 3/4);

Luminosità: 10.000 mcd;

Caduta tensione: 3,4V;

Corrente max: 20mA.

Per il dimensionamento delle resistenze abbiamo come dati:

Tensione alimentazione: 5V;

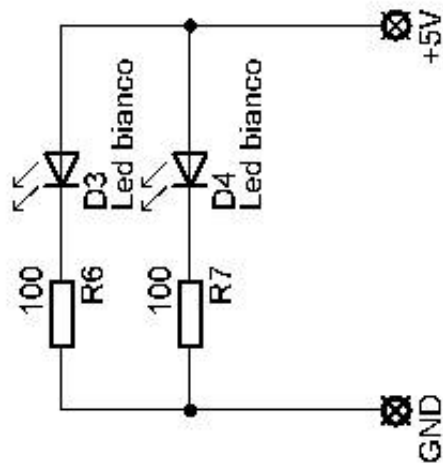
Caduta di tensione del led: 3,4V;

Corrente massima: 20 mA;

Numero led: 2.

Dal calcolo si ottiene un valore teorico di 82 ohm che può essere elevato al valore di 100 ohm. Per il calcolo consiglio questo [link](#) ^[32].

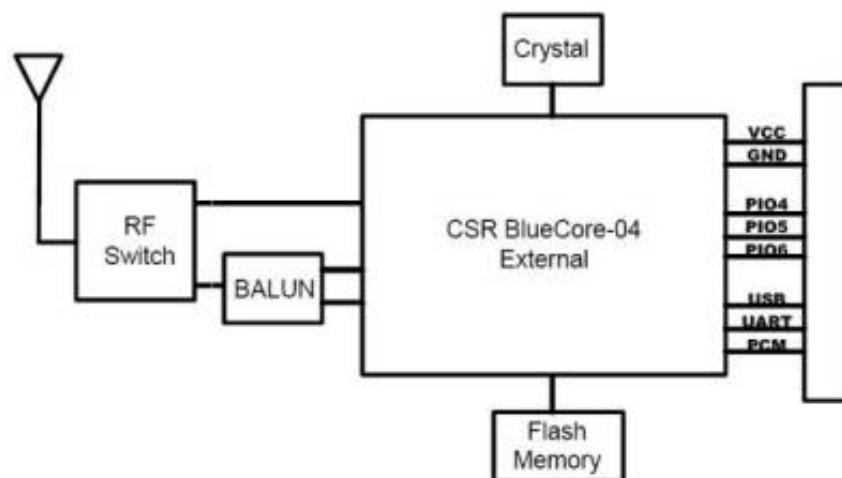
Nota: in realtà il modulo utilizzato è analogo a quello presente sul robot **I-Droid 01** ^[33].

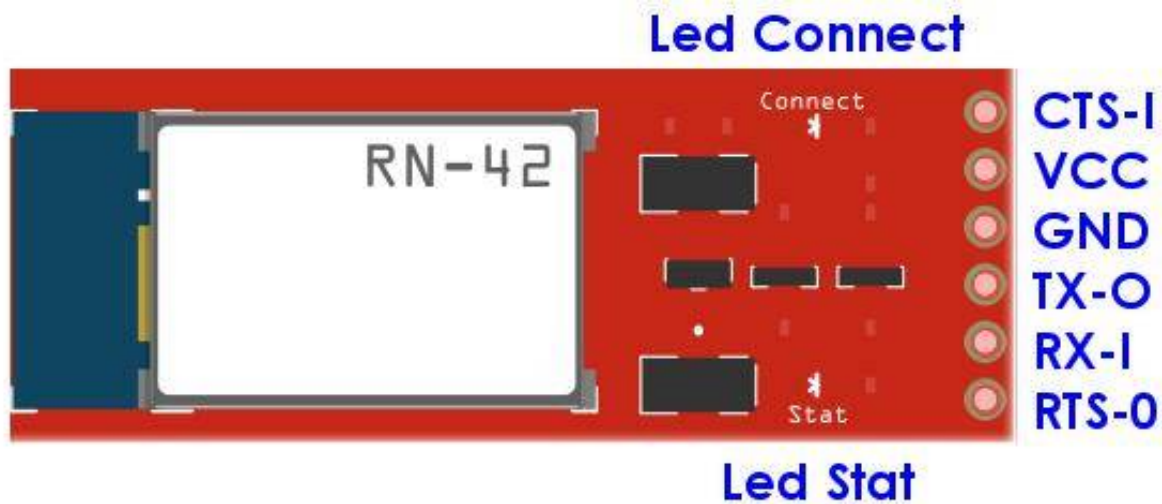


Modulo Bluetooth

Per il comando in remoto, sulla scheda di controllo è presente uno zoccolo per l'installazione di un modulo bluetooth tipo **BlueSMiRF Silver** ^[34] prodotto dalla **SparkFun** ^[11], e basato sull'integrato **RN-42** (datasheet ^[35]) prodotto dalla **ROVING Networks** acquisita recentemente dalla **Microchip** ^[36].

Il modulo è analogo a quello descritto nell'articolo: "**Dotiamo la Arduino Esplora dell'interfaccia Bluetooth** ^[37]".





Specifiche tecniche:

Approvazione FCC Classe 2 Bluetooth Modem Radio;

Dimensioni - 51.5x15.8x5.6mm;

Collegamento molto robusto sia in integrità sia come distanza di trasmissione (18m);

Hardy frequency hopping scheme - opera in ambienti RF come il Wi-Fi, 802.11g, e Zigbee;

Connessione crittografata;

Frequenza: 2.4 ~ 2,524 GHz;

Tensione di funzionamento: 3.3V-6V;

Comunicazione seriale: 2400-115200 bps;

Temperatura di funzionamento: -40 ~ +70 C;

Antenna integrata.

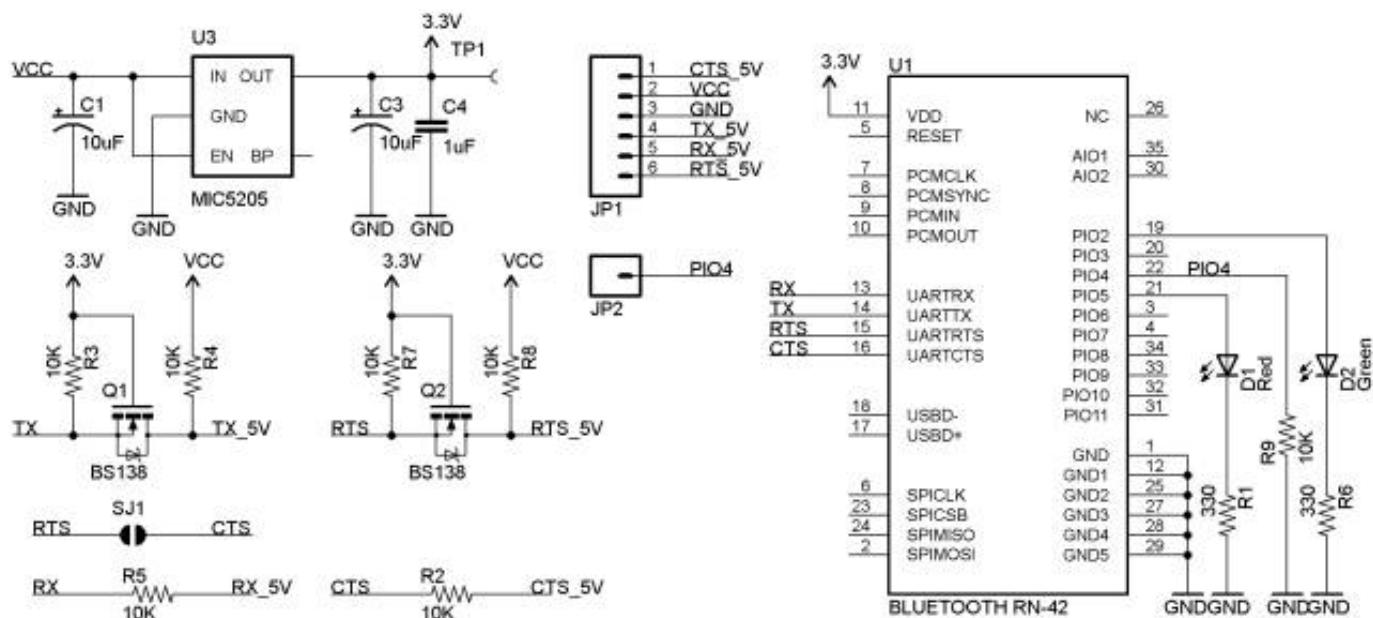
Questo modem funziona come una porta seriale (RX TX), potendo trasmettere un flusso seriale 2400 a 115200 bps.

Analizzando lo schema elettrico ^[38] del modulo, si può vedere che l'unità può essere alimentata con una tensione i cui valori possono essere compresi tra i 3,3 V fino a 6V, in quanto è dotato di un regolatore di tensione tipo **MIC5205** (datasheet ^[39]) che fornisce in uscita di 3.3V per il funzionamento dell'integrato.

Tutti i pin del segnale sono 3V-6V tolleranti, in quanto sulla scheda sono già presenti i circuiti per adattare i livelli di tensione formati da **MOSFET** (N-Channel) di tipo SMD modello **BSS138** (datasheet) ^[40].

Il modulo possiede una pin strip a 6 pin in cui sono presenti i segnali: RX, TX, CTS, RTS, VCC e GND.

Sono inoltre presenti due led indicatori: il led di **colore rosso (Stat)** segnala che il modulo è alimentato ma non connesso, l'accensione del led di **colore verde (Connect)** segnala che è stata stabilita una connessione.

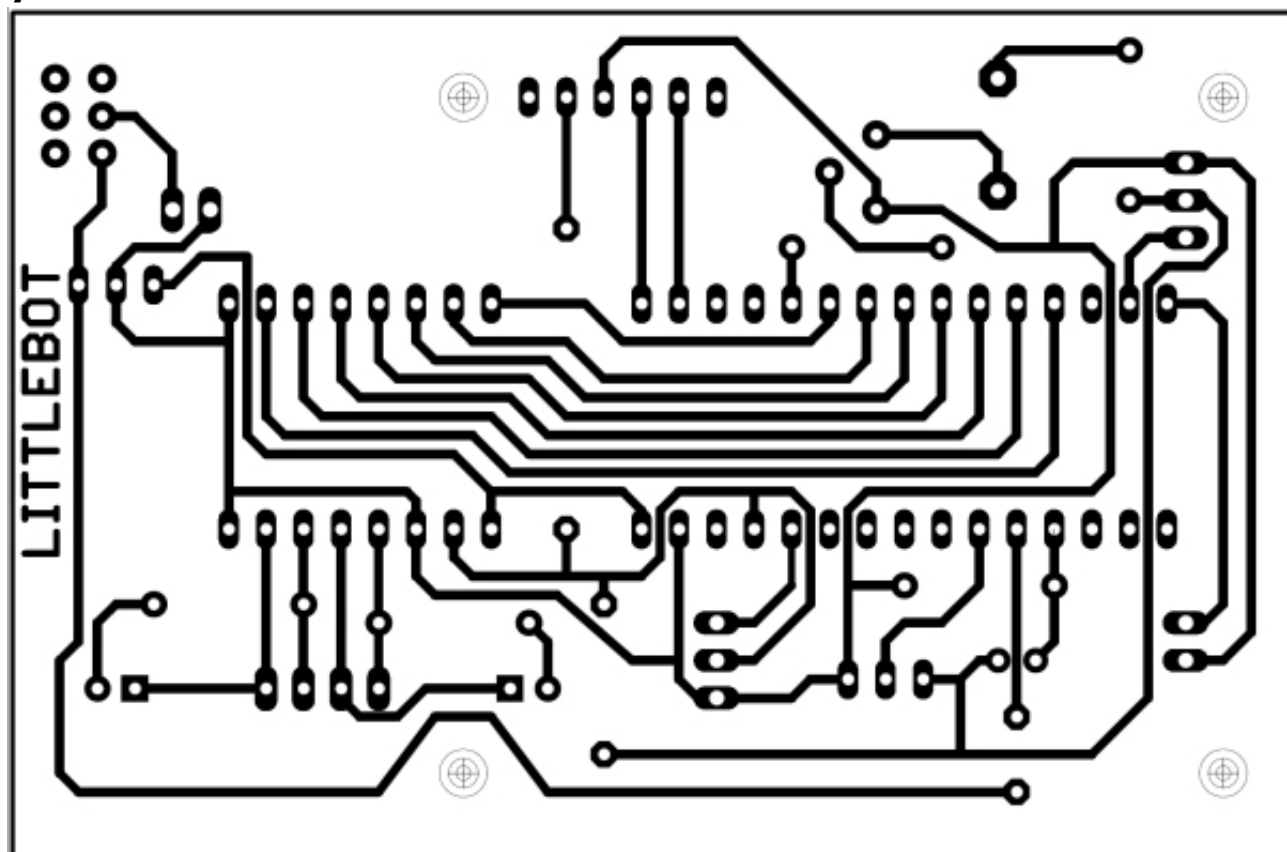


Il connettore di collegamento del modulo è X9.

Costruzione della scheda

Abbiamo così terminato l'analisi della scheda, possiamo ora passare alla sua costruzione.

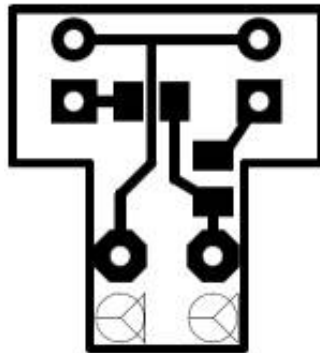
Layout



L'immagine sopra, riporta la traccia del circuito stampato che è del tipo monofaccia e sono previsti solamente tre ponticelli.

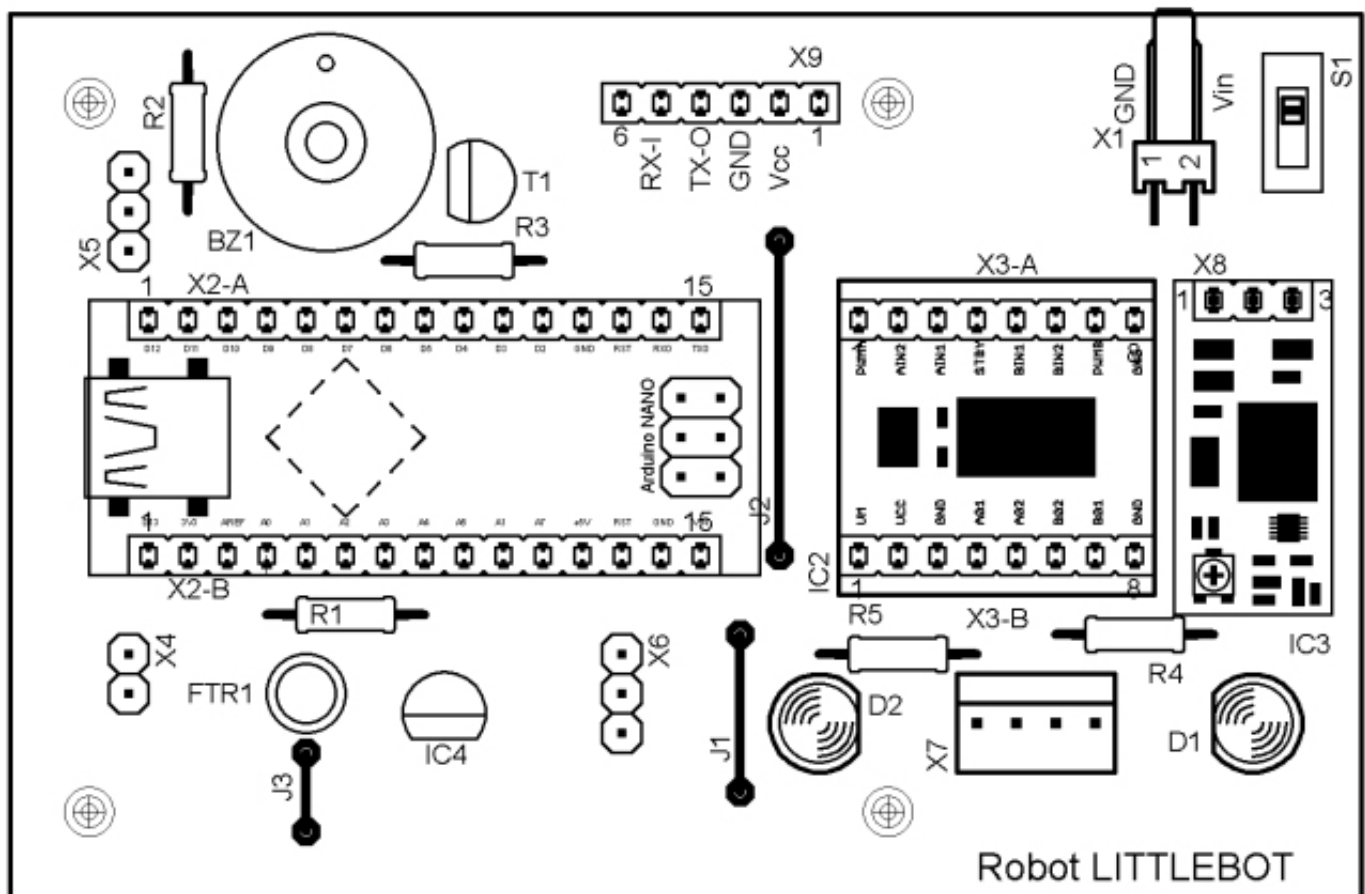
Negli allegati, oltre alla [traccia nel file in PDF](#) ^[41], sono anche presenti i [file in formato](#)

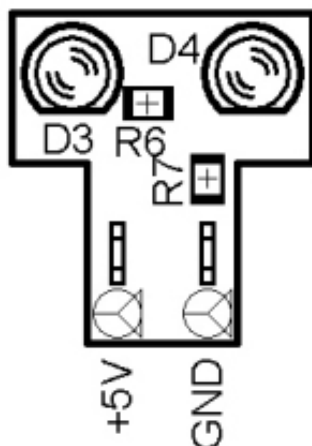
Gerber ^[42] che è lo standard de-facto utilizzato per la produzione di circuiti stampati (PCB) per tracciare le connessioni elettriche quali piste, vias e piazzole; in aggiunta, il file contiene informazioni per la foratura e la fresatura del circuito stampato.



Piano di montaggio

Di seguito invece, è visibile l'immagine del piano di montaggio.

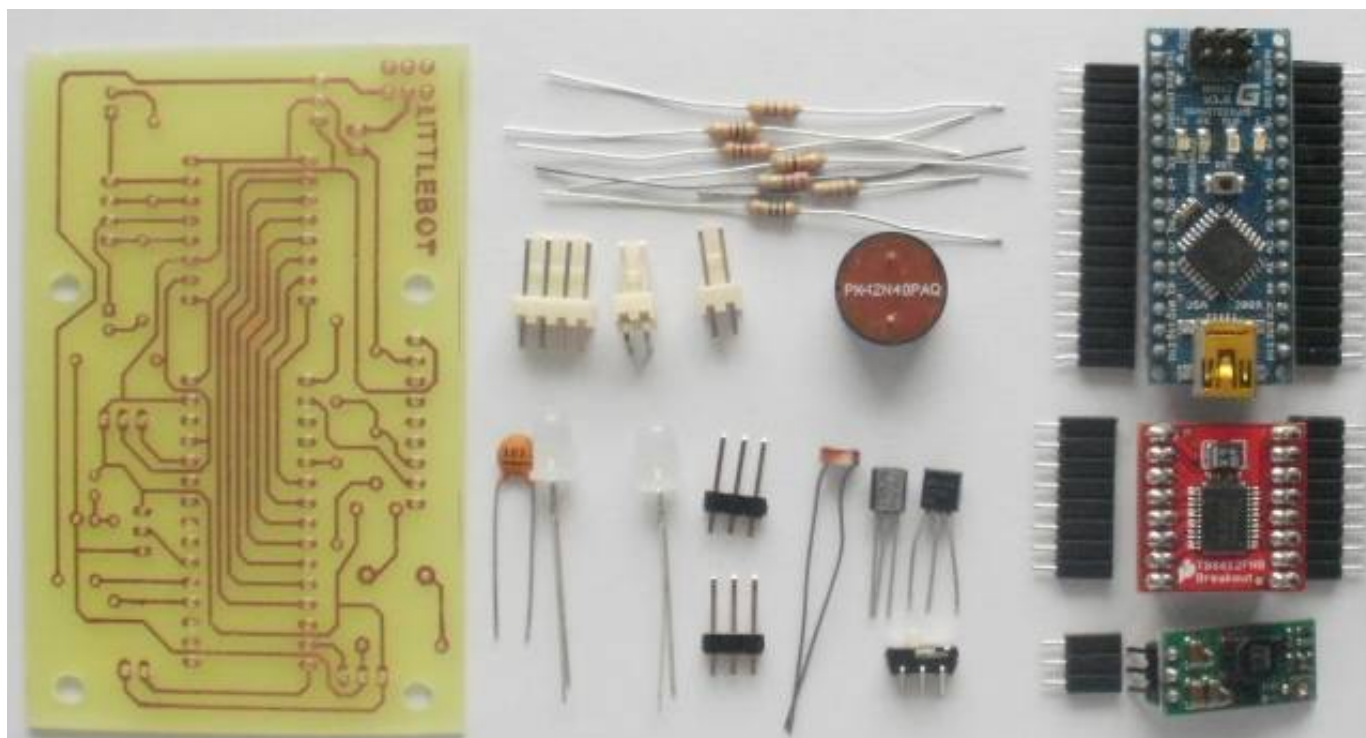




Lista Parti

n°	Q.tà	Riferimento	Descrizione	Particolare
1	1	IC1	Modulo	Arduino Nano
2	1	IC2	Modulo	Motor Driver 1A dual TB6612FNG
3	1	IC3	Modulo	Boot regolatore 791
4	1	IC4	Integrato	LM35
5	1	T1	Transistor	NPN tipo BC337
6	1	R1	Resistenza	¼W 10kΩ
7	1	R2	Resistenza	¼W 10'Ω
8	1	R3	Resistenza	¼W 4,7kΩ
9	2	R4,R5	Resistenza	¼W 1kΩ
10	2	R6,R7*	Resistenza	100'Ω smd M0805
11	2	D1,D2	Diodo bicolore	Led bicolore (verde/rosso) Ø 5mm
12	1	D3,D4*	Diodo led	Led bianco alta luminosità Ø 5mm
13	1	FTR	Fotoresistenza	fotoresistenza 1/10k 'Ω
14	1	BZ1	Buzzer	Buzzer PK-12N40PAQ
15	1	S1	Interruttore	Interruttore a slitta
16	1	X1	Connettore	Connettore 2 pin maschio 90° Molex
17	2	X2A, X2B	Pinstrip	Femmina 15x1
18	2	X3A, X3B	Pinstrip	Femmina 8x1
19	1	X4	Connettore	Connettore 2 pin maschio Molex
20	1	X5	Pinstrip	Maschio 3x1
21	1	X6	Pinstrip	Maschio 3x1 (vedi rif. N° 25)
22	1	X7	Connettore	Connettore 4 pin maschio Molex
23	1	X8	Pinstrip	Femmina 3x1
24	1	X9	Pinstrip	Femmina 6x1 (vedi rif N°26)
25	1		Sensore	GP2D120X
26	1		Modulo	Modulo bluetooth BlueSMiRF Silver

Nota: i componenti con * sono montati sul piccolo circuito indipendente.



Realizzazione pratica.

Per la costruzione della scheda, si procederà iniziando dalla realizzazione del circuito stampato, la traccia è scaricabile in scala 1:1 mediante il link , oppure utilizzando i file gerber.

Occorre ricordare che la stampa dovrà essere fatta deselezionando la funzione che adatta il foglio alla pagina. Per la sua realizzazione, si utilizzerà una basetta in vetronite (monofaccia) di dimensioni 87x57mm, il metodo potrà essere quello della fotoincisione o del trasferimento termico utilizzando i cosiddetti fogli blu (**PRESS-N-PELL** ^[43]), **in questo caso ricordo che l'immagine delle tracce del circuito dovrà essere speculare.**

Una volta inciso il rame, si verificherà in controluce o mediante l'utilizzo di un multimetro, che non vi siano cortocircuiti soprattutto tra le piste più vicine.

Si passerà quindi alla foratura della stessa, utilizzando principalmente una punta da 0,8 mm, mentre se ne utilizzerà una da 1 mm per le pin strip. In seguito, si potrà passare al posizionamento e alla saldatura dei componenti seguendo lo schema di montaggio visto prima.

Per la saldatura, si utilizzerà un piccolo saldatore a punta fine, della potenza di circa 25 – 30 W, oppure una stazione di saldatura come quella presentata nell'articolo: "**Come saldare i componenti elettronici con una stazione di saldatura digitale** ^[44]".

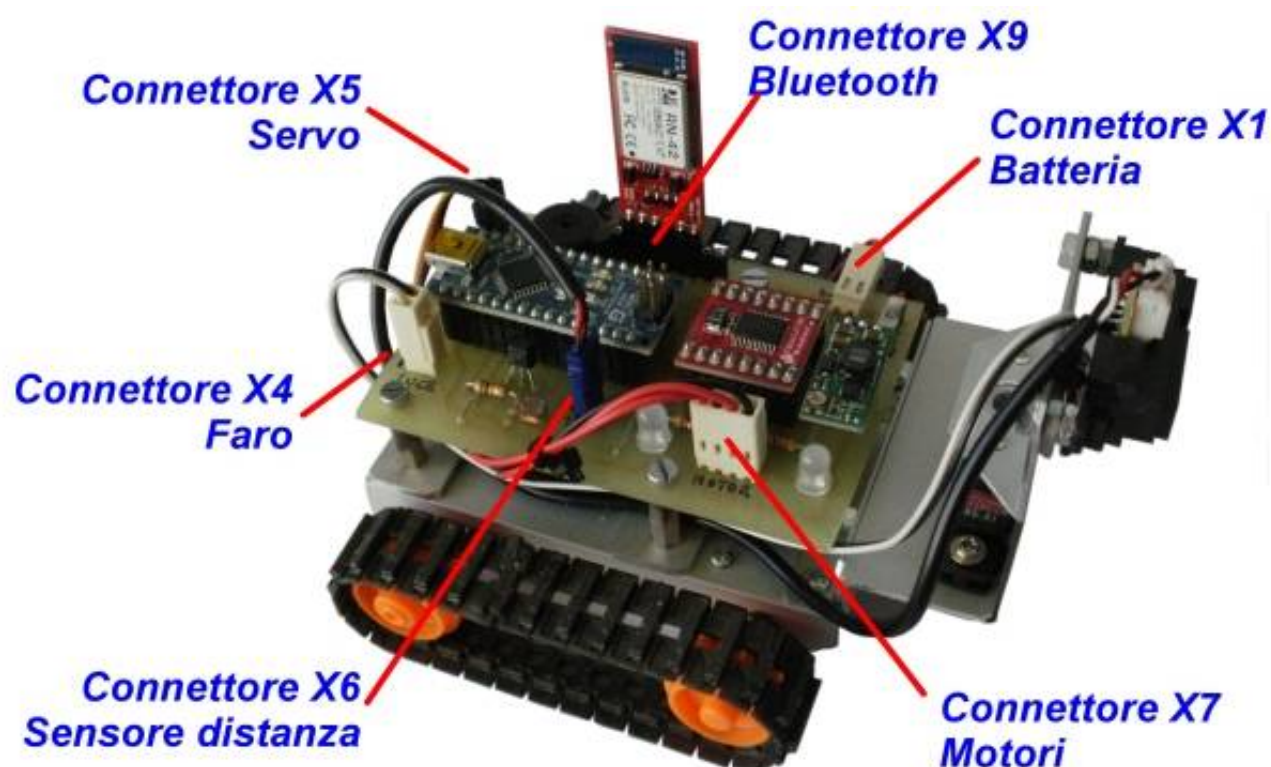
S'inizierà dai tre ponticelli, continuando con le resistenze, si potrà quindi, procedere con i diodi led, facendo attenzione alla tacca di riferimento. Si terminerà con l'interruttore a slitta, le varie pintrip e connettori.

Terminata la saldatura, si potranno inserire i vari moduli facendo riferimento al piano di

montaggio.

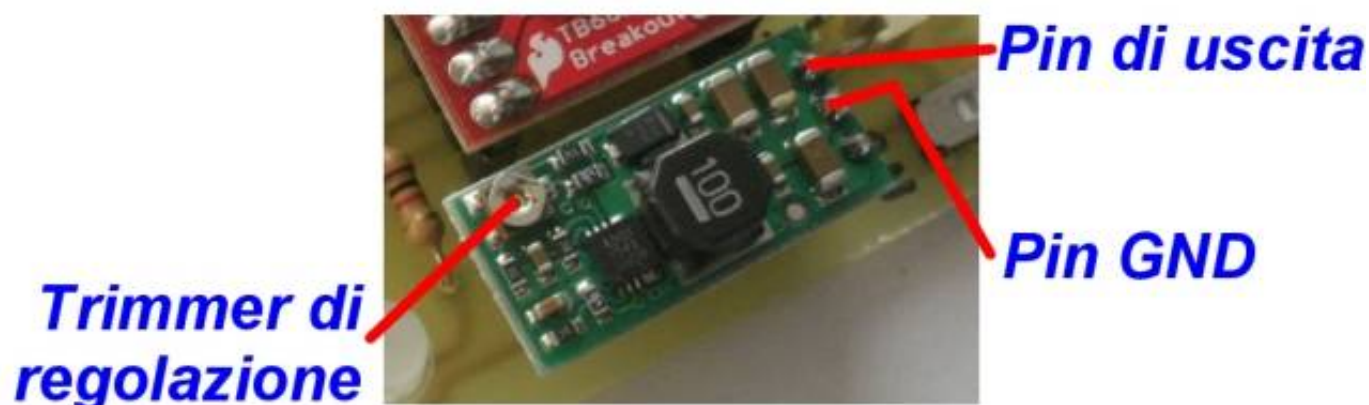
Posizione connettori

Per facilitare il collegamento nella foto sotto riportata sono indicate le posizioni dei vari connettori.



Taratura della tensione di alimentazione

Unica operazione di taratura da compiere è quella del trimmer presente sul modulo **IC3**: per fare questo, ruotare il trimmer sino a leggere tramite un multimetro digitale che tra il pin d'uscita e GND vi sia una tensione di 7-8V.



Programma di test

Per testare la scheda è possibile caricare nella memoria un semplice programma, che è presente negli allegati [45].

Il programma permetterà il controllo delle seguenti parti:

Lettura tensione batteria;
 Lettura temperatura da Sensore LM35;
 Lettura distanza da sensore GP2D120X;
 Lettura luminosità da fotoresistenza;
 Emissione di un Beep dal buzzer;
 Accensione/spegnimento faro;
 Accensione/spegnimento led sul Modulo Arduino Nano.

Per fare questo si caricherà il programma nella memoria di Arduino e poi, tramite il monitor seriale, si invieranno dei comandi e in risposta si riceveranno i dati da Arduino e si compiranno le azioni richieste.

I comandi disponibili sono:

b - Legge tensione della batteria

t - Legge temperatura

d - Legge distanza

l - Legge luminosità

s - Emette un BIP

k - Accende il faro

j - Spegne il faro

n - Accende led

m - Spegne led

Se si invia il comando **"h"**, si avrà l'elenco delle opzioni.



Nota: Il modulo Bluetooth comunica con Arduino Nano attraverso la porta hardware **UART**, la stessa porta è anche utilizzata dall'IDE Arduino per effettuare il download del firmware.

Esiste quindi un accesso contemporaneo alla risorsa e chiaramente le due cose non possono funzionare insieme.

Quindi, quando dovete programmare la scheda Arduino NANO, occorre rimuovere il modulo Bluetooth, effettuare il download, inserire nuovamente il modulo Bluetooth ed infine accendere la scheda.



```
/*
LittleBOT_test.ino
Il programma verifica il funzionamento
di alcuni sensori e dispositivi della scheda di controllo
```

Vengono utilizzati i seguenti pin

Pin Digitale 2 -> Buzzer

Pin Digitale 12 -> Faro

Pin Digitale 13 -> Led su Arduino Nano

Pin Analogico A0 -> Misurazione fotoresistenza

Pin Analogico A1 -> Misurazione tensione batteria

Pin Analogico A2 -> Misurazione temperatura

Pin Analogico A7 -> Misurazione distanza

Ultima modifica il 30/12/2012

da Adriano Gandolfo

Sito WEB <http://www.adrirobot.it>

Blog <https://it.emcelettronica.com/blog/26920>

This example code is in the public domain.

*/

```
#include <DistanceGP2Y0A41SK.h>
```

```
/* Pin Digitali */
```

```
#define buzzer 2
```

```
#define luce 12
```

```
#define ledpin 13
```

```
/*Pin analogici*/
```

```
#define vbat 1
#define foto 0
#define temp 2
#define sharp 7

/*Definizione variabili*/
DistanceGP2Y0A41SK Dist;
int distance;

void setup() {

    Serial.begin(9600); //imposta la porta di comunicazione con il modulo
    pinMode(luce, OUTPUT);
    pinMode(ledpin, OUTPUT);
    Dist.begin(sharp);
}

void loop() {

    while (Serial.available() < 1) {
    } // Attesa sino a quando riceve un carattere
    char val = Serial.read(); // Legge il carattere dal modulo Bluetooth e lo
salva nella variabile val
    switch(val) // Eseguie i comandi in base al carattere ricevuto
    {
    case 'k': //Se il carattere ricevuto è 'k' Accende faro frontale
        digitalWrite(luce, HIGH);
        break;
    case 'j': //Se il carattere ricevuto è 'j' Spegne il faro frontale
        digitalWrite(luce, LOW);
        break;
    case 'l': //Se il carattere ricevuto è 'l' Legge la fotocellula
        leggi_luce ();
        break;
    case 's': //Se il carattere ricevuto è 's' Emette un suono
        tone (buzzer,500,500);
        break;
    case 't': //Se il carattere ricevuto è 't' Legge temperatura tramite LM35
        temperatura ();
        break;
    case 'b': //Se il carattere ricevuto è 'b' Legge tensione della batteria
        batteria ();
    }
```

```

    break;
    case 'd': //Se il carattere ricevuto è 'd' Legge la distanza di un'oggetto dal
sensore
    distanza ();
    break;
    case 'n': //Se il carattere ricevuto è 'n' accende il led di segnalazione
    led_on ();
    break;
    case 'm': //Se il carattere ricevuto è 'm' spegne il led di segnalazione
    led_off ();
    break;
    case 'h': //Se il carattere ricevuto è 'h' esegue la routine HELP
    help();
    break;
    // default:
    // ();
    // break;
}
}
/* Accensione led segnalazione*/
void led_on()
{
    digitalWrite(ledpin, HIGH); // sets the LED on
}

/* Accensione led segnalazione*/
void led_off()
{
    digitalWrite(ledpin, LOW); // sets the LED off
}

/* Lettura della tensione della batteria*/
void batteria ()
{
    float vbatt = (analogRead(vbat)*((readVcc()/1000.0)/1024));
    Serial.print("\nTensione batteria: ");
    Serial.print (vbatt,1);
    Serial.print (" V");
}

/* Legge la temperatura */
void temperatura ()

```

```
{
  /* Lettura sensore */
  float tempC = analogRead(temp);
  tempC= (5.0 * tempC * 100.0)/1024.0; //conversione tensione-temperatura
  /*Trasmissione della temperatura */
  Serial.print("\nTemperatura: ");
  Serial.print(tempC,1);
  Serial.print(" gradi C");
}

/* Verifica funzionamento fotocellula */
void leggi_luce ()
{
  float lettura_foto = analogRead(foto); // lettura grezza dall'adc
  Serial.print("\nValore luce = ");
  Serial.print(lettura_foto,0); //stampiamo il valore
}

/* Lettura distanza*/
void distanza()
{
  distance = Dist.getDistanceCentimeter();
  Serial.print("\nDistanza in centimetri: ");
  Serial.print(distance);
}

/* Help*/
void help()
{
  Serial.print("\nComandi robot ");
  Serial.print("\nh -> Questo elenco");
  Serial.print("\nb -> Tensione della batteria");
  Serial.print("\nt -> Temperatura");
  Serial.print("\nd -> Legge distanza");
  Serial.print("\nl -> Legge luminosita'");
  Serial.print("\ns -> Emette un BIP");
  Serial.print("\nk -> Accende il faro");
  Serial.print("\nj -> Spegne il faro");
  Serial.print("\n -> Accende led");
  Serial.print("\nm -> Spegne led");
}
```

```
/* Routine per rendere la lettura ADC più accurata su Arduino tratta dal sito
http://provideyourown.com/2012/secret-arduino-voltmeter-measure-battery-voltage/
*/
long readVcc() {
    long result;
    // Read 1.1V reference against AVcc
    ADMUX = _BV(REFS0) | _BV(MUX3) | _BV(MUX2) | _BV(MUX1);
    delay(2); // Wait for Vref to settle
    ADCSRA |= _BV(ADSC); // Convert
    while (bit_is_set(ADCSRA,ADSC));
    result = ADCL;
    result |= ADCH<<8;
    result = 1126400L / result; // Back-calculate AVcc in mV
    return result;
}
```

Conclusioni

Abbiamo affrontato la realizzazione della scheda di controllo.

Già in questa fase, potremmo fare diverse prove, poiché sono già operativi i vari sensori.

Avendo a disposizione il file in formato EAGLE ^[46] ognuno potrà personalizzare la scheda, magari per adattarla a un proprio telaio diverso da quello previsto, poiché in vendita, esistono vari telai pronti da montare.

Potrà inoltre, utilizzare un altro modulo Arduino, per esempio un Arduino Micro ^[47].

Per quanto riguarda il programma, potrete modificarlo a vostro piacimento o integrarlo con altre funzioni. Nel prossimo articolo affronteremo la costruzione vera e propria del robot che collauderemo con un programma di test.

Nei commenti a quest'articolo potrete chiedere chiarimenti e approfondimenti inerenti alla costruzione della scheda.

Article printed from Elettronica Open Source: <https://it.emcelettronica.com>

URL to article: <https://it.emcelettronica.com/costruzione-del-robot-littlebot-scheda-di-controllo>

URLs in this post:

[1] robot LittleBot: <https://it.emcelettronica.com/presentazione-del-robot-littlebot>

- [2] Arduino Nano: <http://arduino.cc/en/Main/ArduinoBoardNano>
- [3] tipo bluetooth: <https://it.emcelettronica.com/elettronica/etichette/bluetooth>
- [4] scheda Arduino Esplora: <https://it.emcelettronica.com/scopriamo-nuova-scheda-arduino-esplora>
- [5] formato PDF: [https://it.emcelettronica.com/files/Schema elettrico Scheda controllo LittleBot.pdf](https://it.emcelettronica.com/files/Schema_elettrico_Scheda_controllo_LittleBot.pdf)
- [6] editabile con il programma EAGLE: [https://it.emcelettronica.com/files/Schema elettrico scheda LittleBot EAGLE.zip](https://it.emcelettronica.com/files/Schema_elettrico_scheda_LittleBot_EAGLE.zip)
- [7] bluetooth: <https://it.emcelettronica.com/bluetooth-guida-definitiva-introduzione-al-protocollo>
- [8] Rendiamo autonoma la Arduino Esplora: <https://it.emcelettronica.com/rendiamo-autonoma-scheda-arduino-esplora>
- [9] Batterie al Litio: scegliamo le migliori: <https://it.emcelettronica.com/batterie-al-litio-scegliamo-le-migliori>
- [10] Polymer Lithium Ion Battery - 2000 mA/h: <https://www.sparkfun.com/products/8483>
- [11] SparkFun: <https://www.sparkfun.com/>
- [12] Pololu: <http://www.pololu.com/>
- [13] codice 791: <http://www.pololu.com/product/791>
- [14] datasheet: <http://www.adrirobot.it/datasheet/integrati/pdf/SC4501%20Amp,%20MHz%20Step-Up%20Switching%20Regulator%20with%20Soft-Start.pdf>
- [15] datasheet: [https://it.emcelettronica.com/files/Datasheet Arduino Nano V3.pdf](https://it.emcelettronica.com/files/Datasheet_Arduino_Nano_V3.pdf)
- [16] datasheet: http://www.adrirobot.it/datasheet/processor/pdf/ATmega328_Processore.pdf
- [17] Arduino UNO: <https://it.emcelettronica.com/arduino>
- [18] datasheet: <http://www.adrirobot.it/datasheet/integrati/pdf/ua78m05.pdf>
- [19] datasheet: [https://it.emcelettronica.com/files/Datasheet TB6612FNG.pdf](https://it.emcelettronica.com/files/Datasheet_TB6612FNG.pdf)
- [20] Toshiba: <http://www.semicon.toshiba.co.jp/eng/#>
- [21] schema elettrico: [https://it.emcelettronica.com/files/Schema elettrico modulo motore.pdf](https://it.emcelettronica.com/files/Schema_elettrico_modulo_motore.pdf)
- [22] Motor Driver 1A dual TB6612FNG: <https://www.sparkfun.com/products/9457>
- [23] datasheet: <http://www.ti.com/lit/ds/symlink/lm35.pdf>
- [24] National Semiconductor: <http://www.ti.com/>
- [25] Arduino ed i sensori di temperatura: <https://it.emcelettronica.com/arduino-ed-i-sensori-di-temperatura>
- [26] datasheet: [https://it.emcelettronica.com/files/Datasheet GP2D120.pdf](https://it.emcelettronica.com/files/Datasheet_GP2D120.pdf)
- [27] Sharp: <http://www.sharpsma.com/>
- [28] categoria di sensori: <http://www.sharpsma.com/optoelectronics/sensors/distance-measuring-sensors/GP2D12>
- [29] link: <https://github.com/jeroendoggen/arduino-distance-sensor-library>
- [30] datasheet: [https://it.emcelettronica.com/files/Datasheet Buzzer PK-12N40PAQ.pdf](https://it.emcelettronica.com/files/Datasheet_Buzzer_PK-12N40PAQ.pdf)
- [31] datasheet: <http://www.adrirobot.it/datasheet/transistor/pdf/BC337-338.pdf>
- [32] link: <http://ledcalculator.net/default.aspx?lang=it-IT>
- [33] I-Droid 01: http://www.adrirobot.it/menu_new/index/index_idroid.htm
- [34] BlueSMiRF Silver: <https://www.sparkfun.com/products/10269>
- [35] datasheet: <http://www.adrirobot.it/datasheet/speciali/pdf/Bluetooth-RN-42-DS.pdf>
- [36] Microchip: <http://www.microchip.com/>
- [37] Dotiamo la Arduino Esplora dell'interfaccia Bluetooth:

<https://it.emcelettronica.com/dotiamo-l%E2%80%99arduino-esplora-dell%E2%80%99interfaccia-bluetooth>

[38] schema elettrico : **[https://it.emcelettronica.com/files/Schema elettrico BlueSMiRF.pdf](https://it.emcelettronica.com/files/Schema_elettrico_BlueSMiRF.pdf)**

[39] datasheet:

<http://www.adrirobot.it/datasheet/integrati/pdf/mic5205%20Regolatore%20di%20tensione.pdf>

[40] datasheet): **<http://www.adrirobot.it/datasheet/transistor/pdf/BSS138.pdf>**

[41] traccia nel file in PDF: **[https://it.emcelettronica.com/files/Stampato Scheda controllo LittleBot.pdf](https://it.emcelettronica.com/files/Stampato_Scheda_controllo_LittleBot.pdf)**

[42] file in formato Gerber : **[https://it.emcelettronica.com/files/File gerber_scheda LittleBot.zip](https://it.emcelettronica.com/files/File_gerber_scheda_LittleBot.zip)**

[43] **PRESS-N-PELL**: **http://www.futurashop.it/index.php?route=product/product&product_id=1427**

[44] **Come saldare i componenti elettronici con una stazione di saldatura digitale**: **<https://it.emcelettronica.com/come-saldare-i-componenti-elettronici-con-stazione-di-saldatura-digitale>**

[45] presente negli allegati:

https://it.emcelettronica.com/files/Programma_LittleBot_test.zip

[46] formato EAGLE: **<http://www.cadsoftusa.com/>**

[47] Arduino Micro: **<http://arduino.cc/en/Main/ArduinoBoardMicro>**

Copyright © 2017 Elettronica Open Source. All rights reserved.