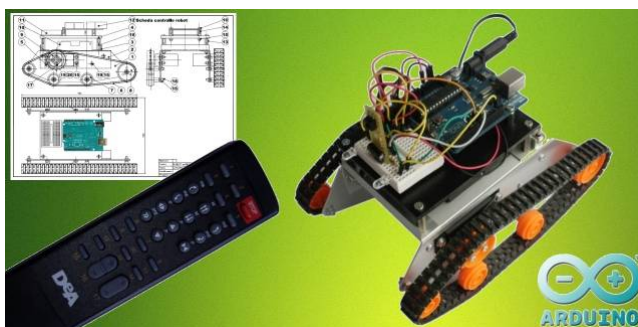


- Elettronica Open Source - <https://it.emcelettronica.com> -

Un telecomando TV per comandare un robot cingolato

Posted By *Adriano Gandolfo* On 26 agosto 2014 @ 5:00 In [Arduino](#), [Embedded](#), [Makers](#) & [Progetti Fai Da Te](#) | [4 Comments](#)



Ormai sono molti i sistemi per controllare un dispositivo a distanza senza fili, per esempio i moderni telefoni si connettono con la rete con il sistema WI-FI oppure tramite Bluetooth. [1]

Esiste, però, un modo ancora più economico ed è quello tramite raggi infrarossi. Quasi tutte le apparecchiature audio e video sono controllate in questo modo, per cui i componenti necessari sono abbastanza a buon mercato, ed il sistema è ideale per l'utilizzo nel campo della robotica o dell'automazione.

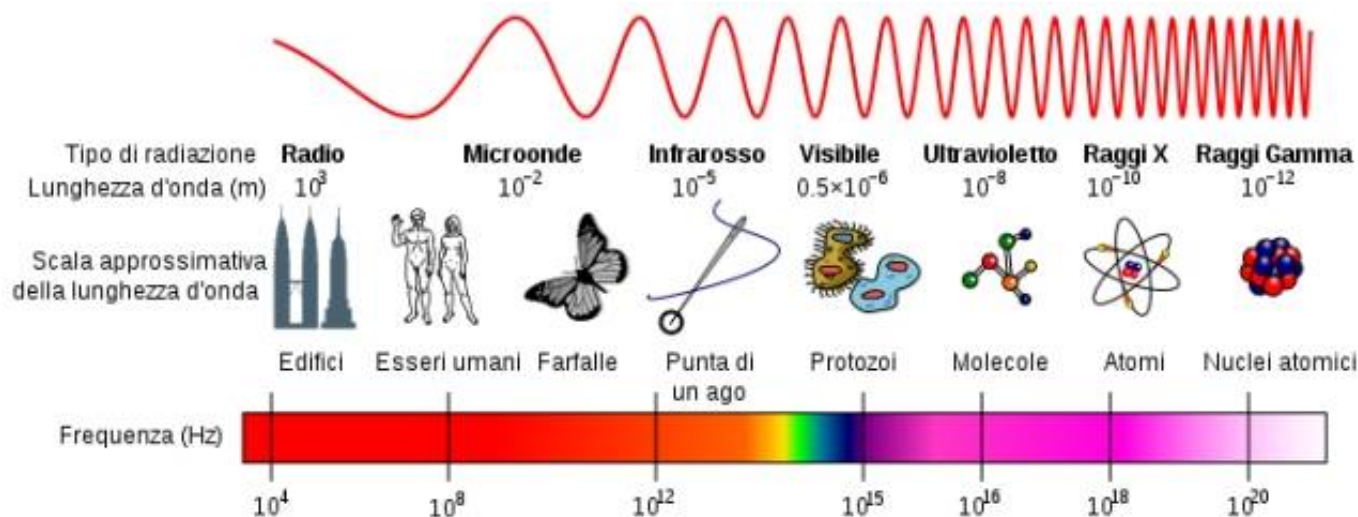
In quest'articolo analizzeremo come funziona un sistema di controllo a infrarossi, costruiremo un semplice robot cingolato e comanderemo l'azionamento dei servomotori di movimento tramite una scheda Arduino [2], utilizzando un sensore infrarosso e un telecomando.

La luce infrarossa

La luce infrarossa o radiazione infrarossa (IR) è una radiazione elettromagnetica con banda di frequenza dello spettro elettromagnetico inferiore a quella della luce visibile, con una lunghezza d'onda compresa tra 700 nm e 1 mm (banda infrarossa).

Il termine significa "sotto il rosso" (dal latino infra, "sotto"), perché il rosso è il colore visibile con la frequenza più bassa.

Per noi esseri umani, questa luce è invisibile perché la sua lunghezza d'onda è al di sotto dello spettro di sensibilità del nostro apparato di sensing.



Un metodo per "vedere" questa luce è quello di utilizzare una videocamera o fotocamera digitale (va bene anche quella di un telefonino).

Basta puntare il telecomando verso una macchina fotografica, premere un pulsante qualsiasi e sarà visibile la luce intermittente del LED.

Ci sono molte fonti di luce infrarossa, poiché tutto ciò che irradia calore, irradia anche luce infrarossa e questo potrebbe creare interferenze (rumore di fondo) tra il telecomando e il sensore.

Il sole è la fonte più brillante di tutte, ma ve ne sono altre come: lampadine, resistenze di riscaldamento, candele, impianto di riscaldamento centralizzato, anche il nostro corpo irradia luce infrarossa.

Perciò, occorre prendere alcune precauzioni per garantire che il nostro messaggio IR arrivi al ricevitore senza errori.

La modulazione

Per eliminare il problema del rumore di fondo, è necessario utilizzare la modulazione. Con la modulazione facciamo sì che la sorgente infrarossa emetta il segnale a una particolare frequenza.

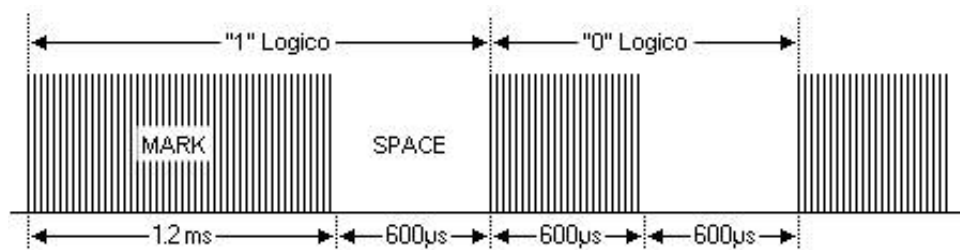
Basterà sintonizzare il ricevitore infrarosso su quella frequenza, in modo da ignorare tutto il resto.

Per analogia si può fare l'esempio del lampeggio delle luci gialle a lato dei cantieri che attirano l'attenzione anche in pieno giorno.

Nella comunicazione seriale di solito parla di «**MARK**» e «**SPACE**».

Il MARK è il segnale di default, che è lo stato di off nel caso del trasmettitore.

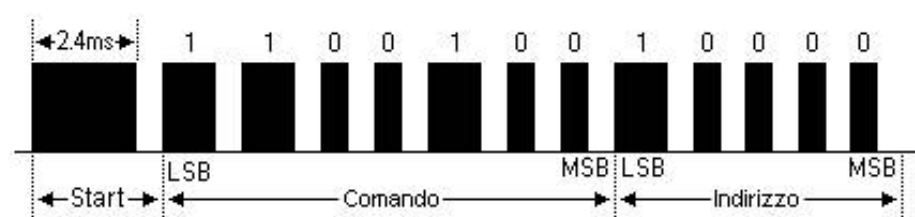
Nessuna luce è emessa durante lo stato di SPACE. Durante lo stato di MARK del segnale di luce IR è pulsata e si spegne a una particolare frequenza, di solito compresa tra 30 e i 60 kHz.



Sul lato ricevente un "SPACE" è rappresentato da un elevato livello di uscita del ricevitore. Un "MARK" è rappresentato automaticamente da un livello basso.

Si prega di notare che i "MARK" e gli "SPACE" non sono gli 1 e gli 0 che vogliamo trasmettere.

Il vero rapporto tra i MARK e gli SPACE e tra 1 e 0 dipende dal protocollo che viene utilizzato.



Nell'immagine sopra è riportato un tipico treno di impulsi del protocollo SIRC (SONY), con questo protocollo il **LSB** (Least Significant Bit - bit meno significativo) viene trasmesso per primo.

Il primo impulso è ampio 2.4ms, seguito da uno spazio standard di 0.6ms.

Oltre a segnalare l'inizio di un messaggio SIRC questo segnale di Start viene utilizzato anche per regolare il guadagno del ricevitore IR.

Viene trasmesso, quindi, il comando di 7-bit, seguito dall'indirizzo del dispositivo di 5-bit.

In questo caso sono trasmessi l'Indirizzo 1 e il Comando 19.

I comandi sono ripetuti ogni 45ms (misurato da Start a Start) per tutto il tempo per cui il tasto sul telecomando viene tenuto premuto.

Il trasmettitore

Il trasmettitore o telecomando è di solito portatile ed alimentato a batteria, deve consumare meno energia possibile, mentre deve fornire un segnale IR più forte possibile per ottenere una distanza di controllo accettabile.

Sul mercato si trovano chip che sono progettati per essere utilizzati come trasmettitori a infrarossi, di solito sono dedicati solo a uno dei molti protocolli che sono stati inventati. Ultimamente sono però utilizzati dei microcontrollori visto il loro basso prezzo e soprattutto la loro versatilità di utilizzo.

Nel funzionamento è previsto che, se non si preme alcun tasto, siano mantenuti in modalità a basso assorbimento e solo quando si preme un tasto, il processore viene risvegliato per l'invio del segnale IR appropriato.

Per il clock del processore non sono di solito utilizzati i quarzi, in quanto molto fragili e con

tendenza alla rottura in caso di cadute; in questo caso si utilizzano i risonatori ceramici, che sono leggermente meno precisi ma molto più resistenti agli urti.

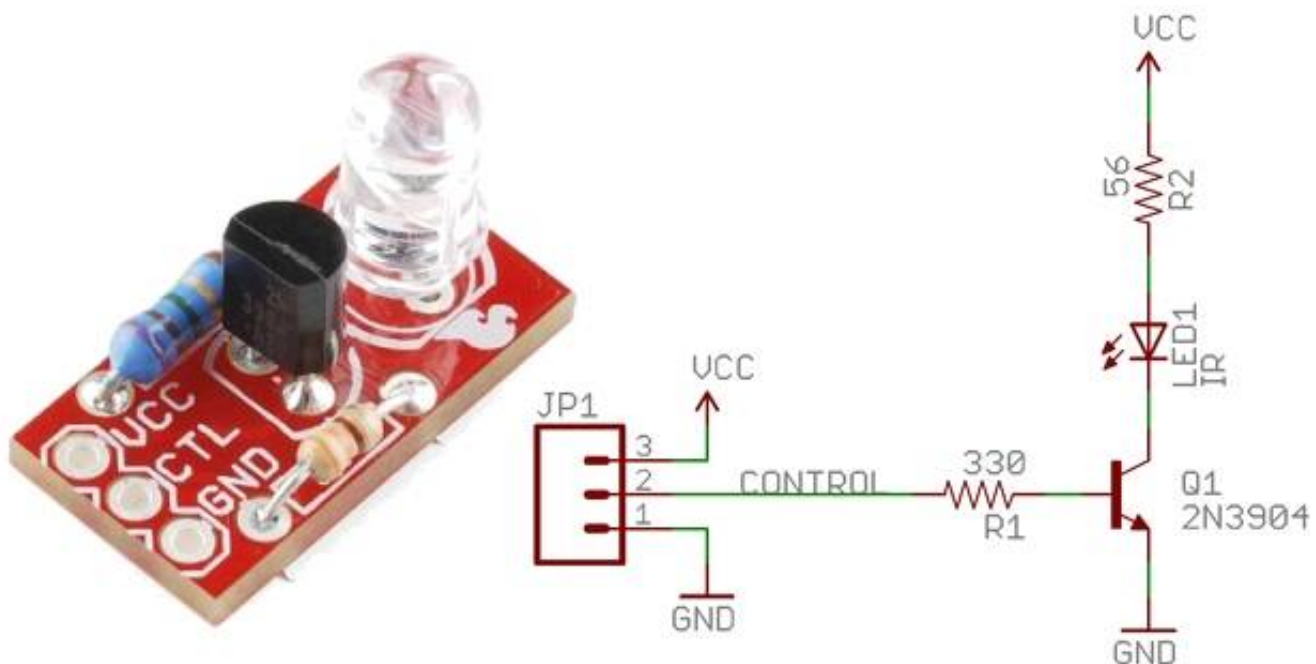
Per il pilotaggio del led, la corrente che lo attraversa, può variare da 50 mA a oltre 1A. Più è alta la corrente, maggiore sarà la distanza raggiunta dal segnale. Un buon compromesso può essere trovato tra i parametri dei LED, la durata della batteria e la distanza massima di controllo.

La corrente di alimentazione del LED può essere alta poiché gli impulsi di pilotaggio sono molto brevi, si consideri comunque che la dissipazione di potenza media del LED non deve superare il suo valore massimo.

È necessario verificare che la corrente massima di picco per il LED, dato che è riportato sulla scheda tecnica del led, non sia superata.

Un esempio di pilotaggio può essere un piccolo modulo fornito dalla **SparkFun**, il modulo denominato **Max Power IR LED Kit** [3]; è formato da uno diodo led emittente luce infrarossa con una lunghezza d'onda di 950nm, in serie è presente la resistenza R2 da 56Ω che è calcolata considerando i parametri di tensione di alimentazione, caduta di tensione sul led e corrente massima che deve scorrere al suo interno.

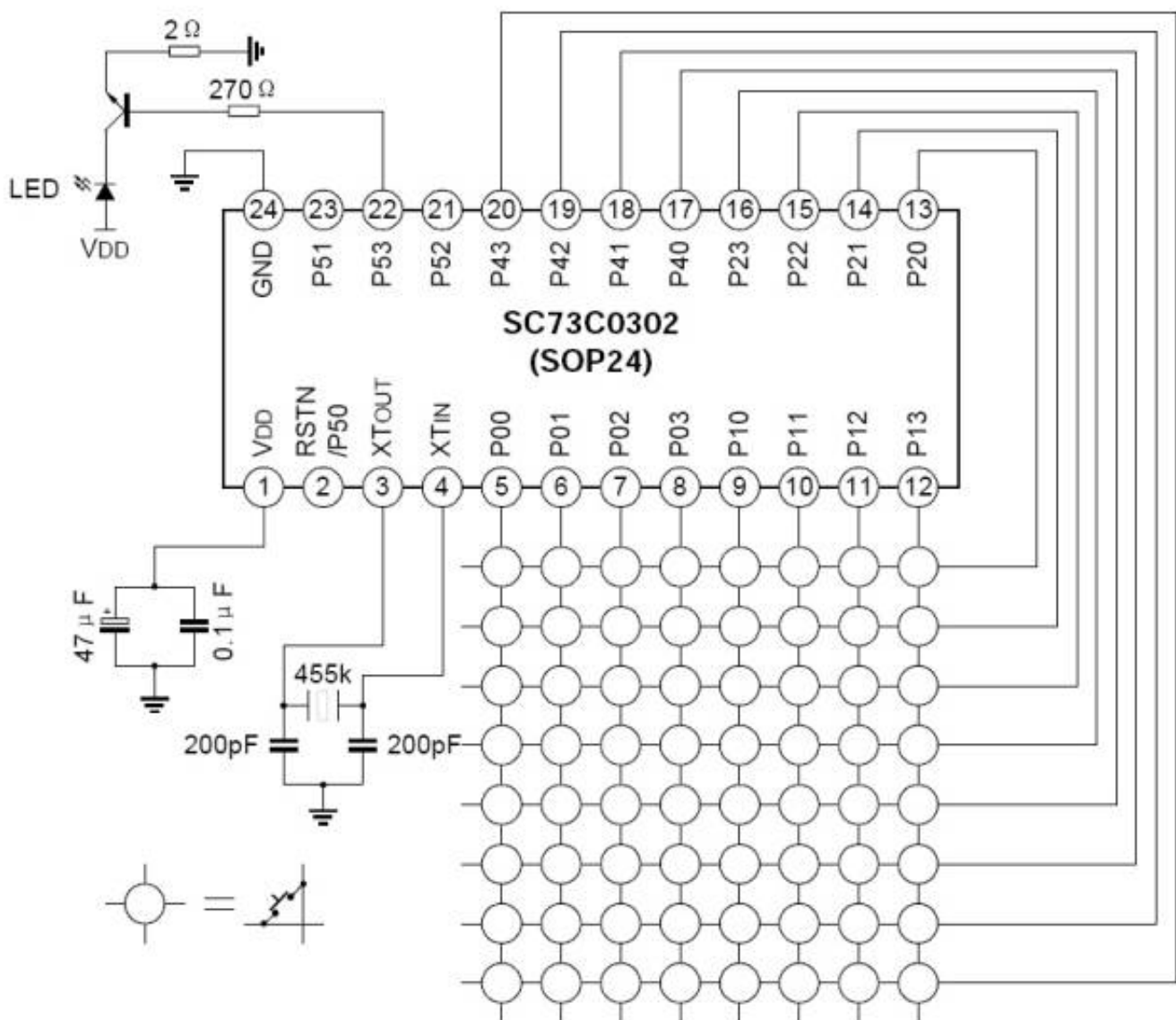
L'accensione è garantita da un transistor NPN tipo **2N3904** sulla cui base è presente una resistenza R1 da 330 Ω.



Nella nostra applicazione, sarà utilizzato un telecomando fornito a suo tempo per comandare il robot Panettone della DeAgostini.

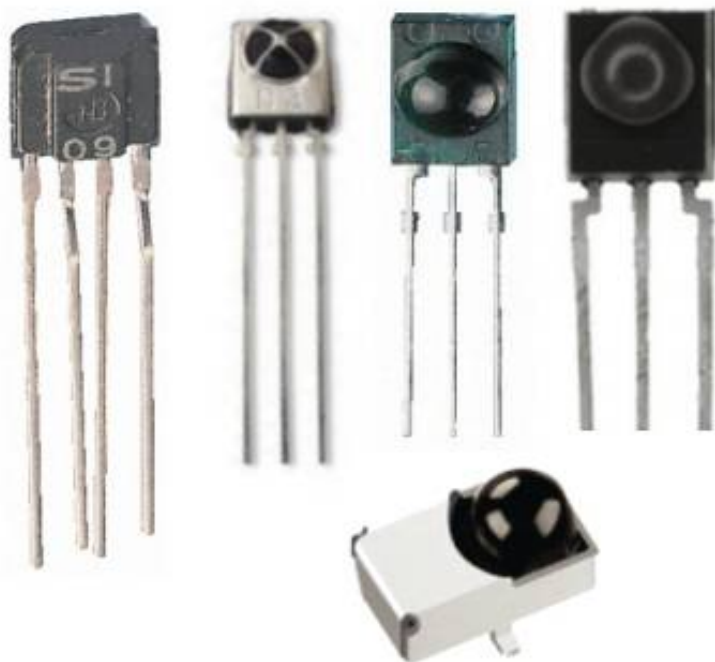


Cuore del circuito di comando è l'integrato **SC73C0302-008** di produzione cinese che codifica i segnali della tastiera con il protocollo Sony-compatibile e li invia tramite un led trasmettente ad infrarosso. L'alimentazione è fornita tramite 2 batterie 1,5V stilo. Il telecomando presenta 27 tasti a cui sono associate varie serigrafie.

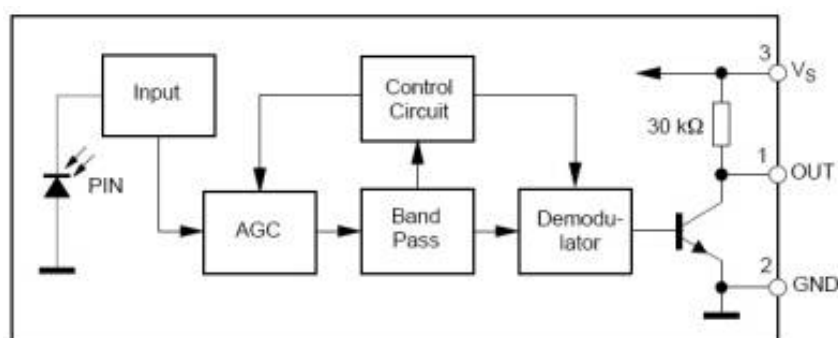


Il Ricevitore

Esistono diversi modelli di ricevitore sul mercato, il più importante criterio di selezione è la frequenza di modulazione utilizzata.



Analizziamo prima di tutto il loro funzionamento: nella foto sotto potete vedere lo schema a blocchi del ricevitore TSOP18XX.



Il segnale IR ricevuto è prelevato dal diodo di rilevazione a infrarossi PIN, questo segnale è amplificato e limitato nel primo blocco di Input.

Nel blocco AGC il segnale è trattato in modo da ottenere un livello di impulso costante, indipendentemente dalla distanza del telecomando.

Il segnale AC è poi inviato al filtro Passa Banda (Band Pass) che è sintonizzato sulla frequenza di modulazione del telecomando.

Queste frequenze possono andare, per esempio, nel modello TSOP18XX (da 30kHz a 40kHz).

Type	fo	Type	fo
TSOP1830	30 kHz	TSOP1833	33 kHz
TSOP1836	36 kHz	TSOP1837	36.7 kHz
TSOP1838	38 kHz	TSOP1840	40 kHz
TSOP1856	56 kHz		

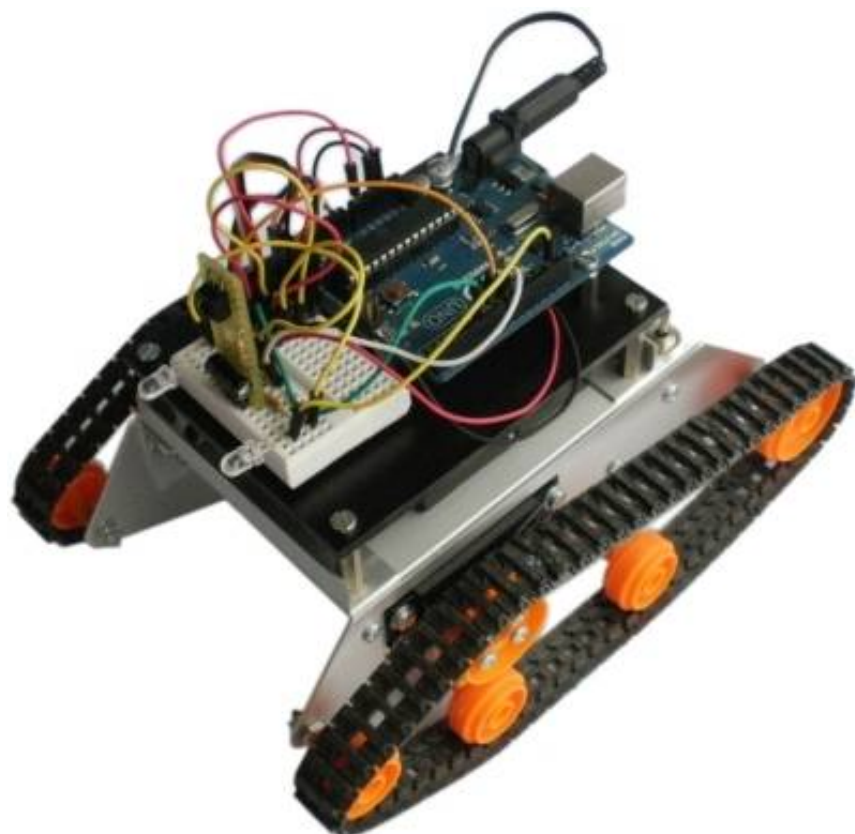
Nel blocco demodulatore avviene la rilevazione della presenza della frequenza di modulazione e, se questa è presente, l'uscita sarà attivata e si avrà il segnale sul pin OUT.

Esistono diversi produttori di ricevitori IR tra i quali: Siemens, Vishay e Telefunken.

Esempio di applicazione

Come esempio di utilizzazione, si è pensato di realizzare un semplice robot che sarà poi comandato tramite i comandi impartiti da un telecomando.

Il modulo ricevente potrà essere montato su una semplice bread board.

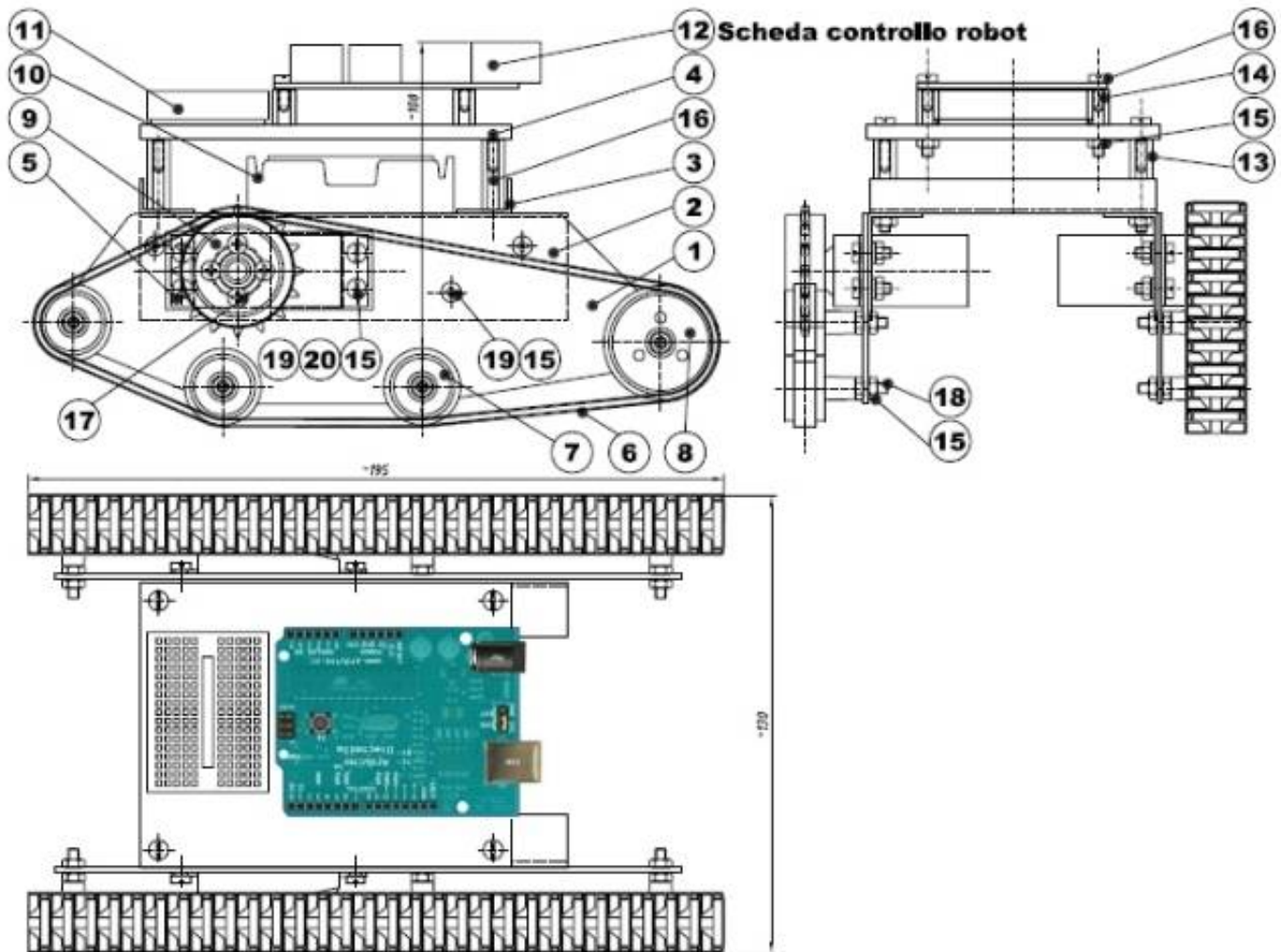


Queste sono solo alcune delle caratteristiche del robot proposto:

Processore scheda di comando	Arduino UNO - ATmega328
Uscite	Comando servomotori Comando led illuminazione
Sensori	Sensore decodificatore segnali IR tipo TSOP 1836 (36 KHz)
Alimentazione	6V tramite 4 batterie 1,5V tipo AA
Motorizzazione	2 servomotori a rotazione continua
Cingoli	In gomma, prodotti dalla Tamiya
Misure	195x130x108 mm
Peso	465 g

Montaggio del robot

Per la costruzione del robot inizieremo dalla lista delle parti necessarie, per la localizzazione sul robot fare riferimento al disegno sottostante.

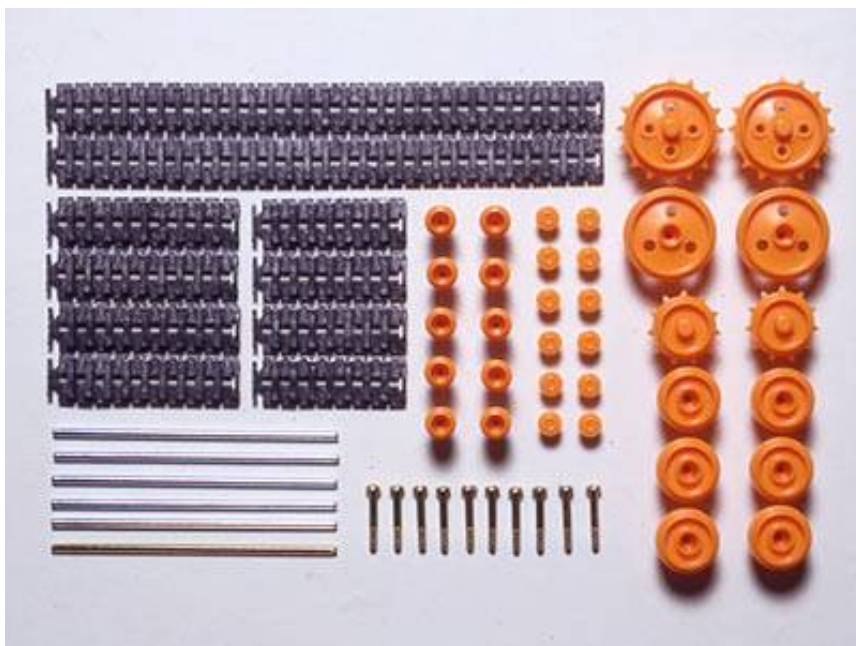


- [1] Piastre laterali - n° 2 pezzi;
- [2] Profilati fissaggio servomotori - n° 2 pezzi;
- [3] Traverse - n° 2 pezzi;
- [4] Piastra supporto scheda controllo - n° 1 Pezzo;
- [5] Servomotori a rotazione continua completi di squadrette - n° 2 pezzi;
- [6] Cingoli 66 maglie - n° 2 pezzi;
- [7] Ruote medie folli Ø21,5 - n° 6 pezzi;
- [8] Ruote grandi folli Ø32 - n° 2 pezzi;
- [9] Pignoni grandi - n° 2 pezzi;
- [10] Portabatteria per 4 batterie tipo AA - n° 1 pezzo;
- [11] Mini breadboard 17x10 - n° 1 pezzo;
- [12] Scheda di controllo utilizzata Arduino UNO;
- [13] Distanziali esagonali M3 L= 20mm - n° 4 pezzi;
- [14] Distanziali esagonali M3 L= 10mm - n° 4 pezzi;
- [15] Dadi M3 - n°38 pezzi;
- [16] Viti testa cilindrica con intaglio M3 L= 8 - n° 8 pezzi;
- [17] Viti testa croce M3 L= 25 - n° 8 pezzi;
- [18] Viti testa croce autofilettante M3 L= 8 - n° 8 pezzi;
- [19] Viti testa cilindrica con intaglio M3 L= 10 - n° 14 pezzi;
- [20] Rosette piane Ø3,2- - n° 8 pezzi;

- [21] Rondelle antisvitamento Ø3,2- - n° 8 pezzi;
- [22] Modulo sensore a infrarossi tipo TSOP1836;
- [23] Mammuto 2 poli;
- [24] Interruttore deviatore a Levetta sub miniatura;
- [25] Spina DC Dimensioni 2,1 x 5,5 x 9,5mm, terminali a saldare.

Cingoli

I particolari utilizzati sono all'interno della confezione **Tamiya 70100 Track and Wheel Set** [4]; sono presenti vari componenti come: spezzoni di cingoli in gomma di varie lunghezze, alberi, ruote e pignoni



La lunghezza di cingoli, pignoni e ruote di dimensioni diverse, consentono flessibilità nella progettazione del proprio robot.

Per il nostro robot si sono utilizzati:

tutti gli spezzoni per formare due cingoli (1 lungo+2 medi+2 piccoli) 66 maglie [6] ;
 n° 6 ruote piccoli folli Ø21,5 [7] ;
 n° 2 ruote grandi folli Ø32 [8] ;
 n° 2 pignoni grandi [9].

Servomotore

Per la motorizzazione del robot, si sono utilizzati due servomotori [5] di taglia normale, da questi è stato eliminato il blocco della rotazione.

Nel prototipo sono stati utilizzati **due servomotori Parallax** [5], ma in rete è possibile trovare vari riferimenti dove viene spiegato come effettuare la modifica su normali servomotori.

Alcune caratteristiche dei servo impiegati:

dimensioni: 40.5 × 20.0 × 38,0 millimetri;
 peso: 43 g;

alimentazione: 4,8-6V;
 velocità A 6V: 50 rpm;
 stallo A6V coppia: 3,0 kg/cm;
 lunghezza del cavo: 270 millimetri.



La batteria e i componenti per il cablaggio elettrico

L'alimentazione del robot è fornita da **4 batterie da 1,5V** tipo AA per un totale di 6V inserite all'interno di un portabatteria **[10]** opportunamente fissato al telaio del robot. È previsto un interruttore deviatore a levetta subminiatura **[24]** per l'accensione e un spina DC **[25]** dimensioni 2,1 x 5,5 x 9,5mm (terminali da saldare per il collegamento con la scheda Arduino).

Per il collegamento delle varie parti si utilizza un mammut 2 poli **[23]**.

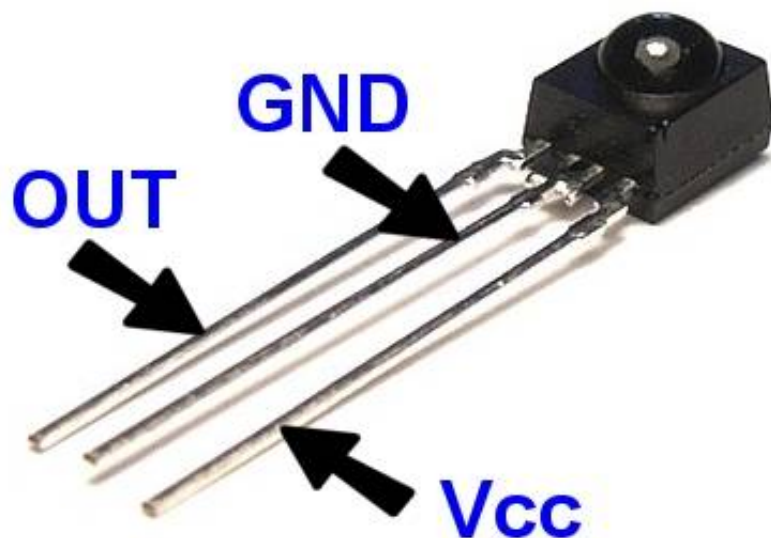


Modulo ricevitore infrarossi

Il sensore utilizzato nel progetto è il tipo **TSOP1836**, che fa parte della serie **TSOP18XX**

prodotto dalla Vishay, questo potrà essere sostituito con l'equivalente **TSOP 4836** ^[6] (da **Conrad codice 171107-62** ^[7]).

Questi sensori, sono ideali come ricevitore per sistemi di telecomando a infrarossi.



Come in precedenza descritto, all'interno del dispositivo, è presente il diodo PIN e il preamplificatore; sono montati su uno stampato inserito all'interno di un contenitore di resina epossidica che ha anche la funzione di filtro IR.

Il segnale di uscita può essere demodulato direttamente e decodificato da un microprocessore.

Il principale vantaggio è il funzionamento affidabile anche in ambienti disturbati e la protezione contro impulsi di uscita incontrollati.

Alcune caratteristiche:

rilevatore IR e preamplificatore in un unico contenitore;

filtro interno per la frequenza PCM;

compatibilità TTL e CMOS;

uscita attiva bassa;

schermatura contro i disturbi di natura elettrica;

maggiore immunità contro tutti i tipi disturbo della luce;

nessuna comparsa di disturbi sugli impulsi sull'uscita;

tempo di assestamento dopo l'accensione molto breve (<200s).

Il sensore è sensibile solamente ad infrarossi con lunghezza d'onda di 950 nm e con una frequenza di 36 kHz.

Questo previene le interferenze IR da sorgenti di infrarossi come la luce solare e l'illuminazione domestica.

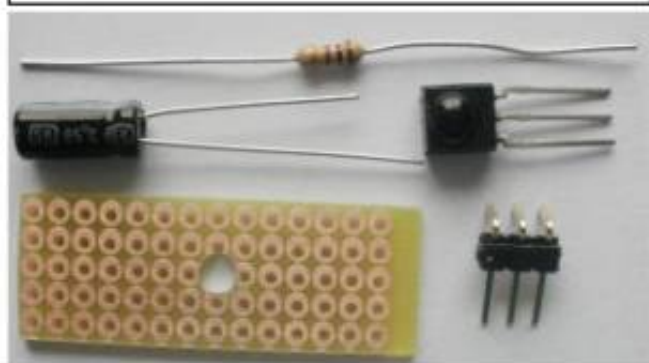
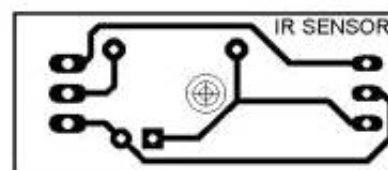
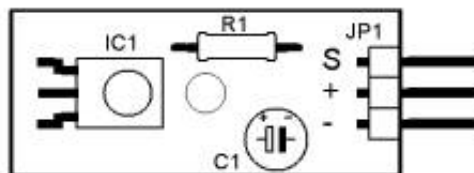
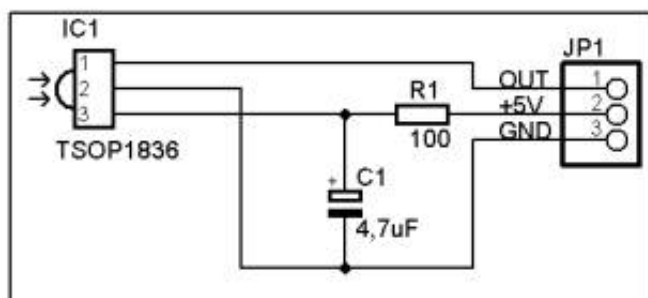
La luce solare è vista come un'interferenza in corrente continua (0 Hz), mentre l'illuminazione domestica è vista come un'interferenza a 100 Hz, poiché è al di fuori della banda passante del filtro elettronico; tale frequenza è completamente ignorata dal rivelatore IR.

Si può utilizzare il sensore così com'è, oppure si può acquistare un apposito modulo su cui, di solito, sono montati dei componenti passivi che prevedono dei pin di collegamento [8].



È possibile realizzarne uno su una millefori oppure con un proprio CS [9].

Dato che l'amplificatore interno all'integrato è impostato su un guadagno molto elevato, il sistema potrebbe iniziare ad oscillare molto facilmente; per questo motivo è presente la resistenza **R1** e il condensatore **C1** in modo da disaccoppiare la linee di alimentazione.



LED illuminazione

Due led ad alta luminosità completano il robot.

Questi sono attivati dal pin digitale 13 che è anche quello che attiva il led presente sulla scheda Arduino UNO [10].

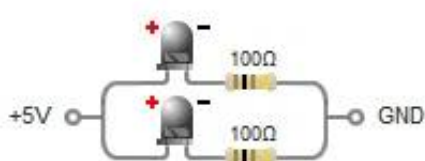
I led sono connessi in parallelo con la propria resistenza di limitazione del valore di 100 Ω ,

con tale valore la corrente assorbita dal pin è ancora nel range disponibile che è di 40 mA.

Per il dimensionamento si è utilizzata **questa risorsa della rete:** ^[11]

Tensione dell'energia (V):	<input type="text" value="5"/>	?
Caduta di tensione del LED (V):	<input type="text" value="3,2"/>	?
Rapporto di corrente del LED (mA):	<input type="text" value="19"/>	?
Numero di LED:	<input type="text" value="2"/>	?
Output:	<input checked="" type="radio"/> Schema elettrico <input type="radio"/> Schematico	
	<input type="button" value="Disegno Circuito"/>	

 [Stampa](#)



Le caratteristiche del led utilizzato tipo **L5WCN5** o equivalente sono:

dimensioni: 5mm;

colore + materiale emettitore: super bianco GaN/SiC;

lunghezza d'onda: 465nm;

intensità luminosa: 20000mcd ($I_f = 20\text{mA}$);

angolo di visione: 20 °;

tensione diretta: 3.2V.

Costruzione particolari del telaio

Il materiale base per la realizzazione del telaio del robot è rappresentato da profilati in alluminio di varie forme e dimensioni che potrà essere trovato presso una grossa ferramenta.

Occorre il seguente materiale:

angolare L in alluminio 30x15x1 mm;

angolare L in alluminio 15x10x1 mm;

piastra alluminio spessore 1,5 mm;

piastra PVC espanso spessore 3 mm.

Per le lavorazioni, sono necessari un seghetto per metalli e una lima da ferro.

Per le forature, occorre un trapano a colonna con le punte dei diametri indicati.

Si raccomanda di effettuare i tagli e le forature serrando i pezzi su una morsa interponendo eventualmente del materiale plastico per evitare di rovinare la superficie dei pezzi.

Le operazioni indicate producono dei trucioli di alluminio, quindi è fortemente consigliato l'utilizzo dei DPI di protezione: per esempio occhiali protezione in policarbonato e guanti per manutenzione, questi potranno essere trovati presso la stessa Ferramenta dove troverete i profilati.



Con questi attrezzi si costruiscono i seguenti particolari:

- [1] Piastre laterali - n° 2 pezzi;
- [2] Profilati fissaggio servomotori - n° 2 pezzi;
- [3] Traverse - n° 2 pezzi;
- [4] Piastra supporto scheda controllo - n° 1 Pezzo;

Per la loro costruzione si farà riferimento alle misure riportate sui **disegni presenti negli allegati.**

1 Piastra alluminio Sp. 1,5 mm 1 Pezzo

2 Profilato in alluminio L 30x15x1 L=120 2 Pezzi

3 Profilato in alluminio L 15x10x1 L=80 mm (1DX+1SX) 2 Pezzi

13 N° 4 pezzi

14 N° 4 pezzi

Riferimento:	1-2-3-13-14
Progetto:	ROBOT adriBOT - Ver. 1.0
Oggetto:	Particolari
Materiale grezzo:	
Quantita':	

Progetto realizzato da Adriano Gandolfo www.adriROBOT.it

5 Servomotore - 2 Pezzi

6 Cingolo 66 maglie (1 lungo [30] + 2 medi [10] + 2 corti [8]) - 2 Pezzi

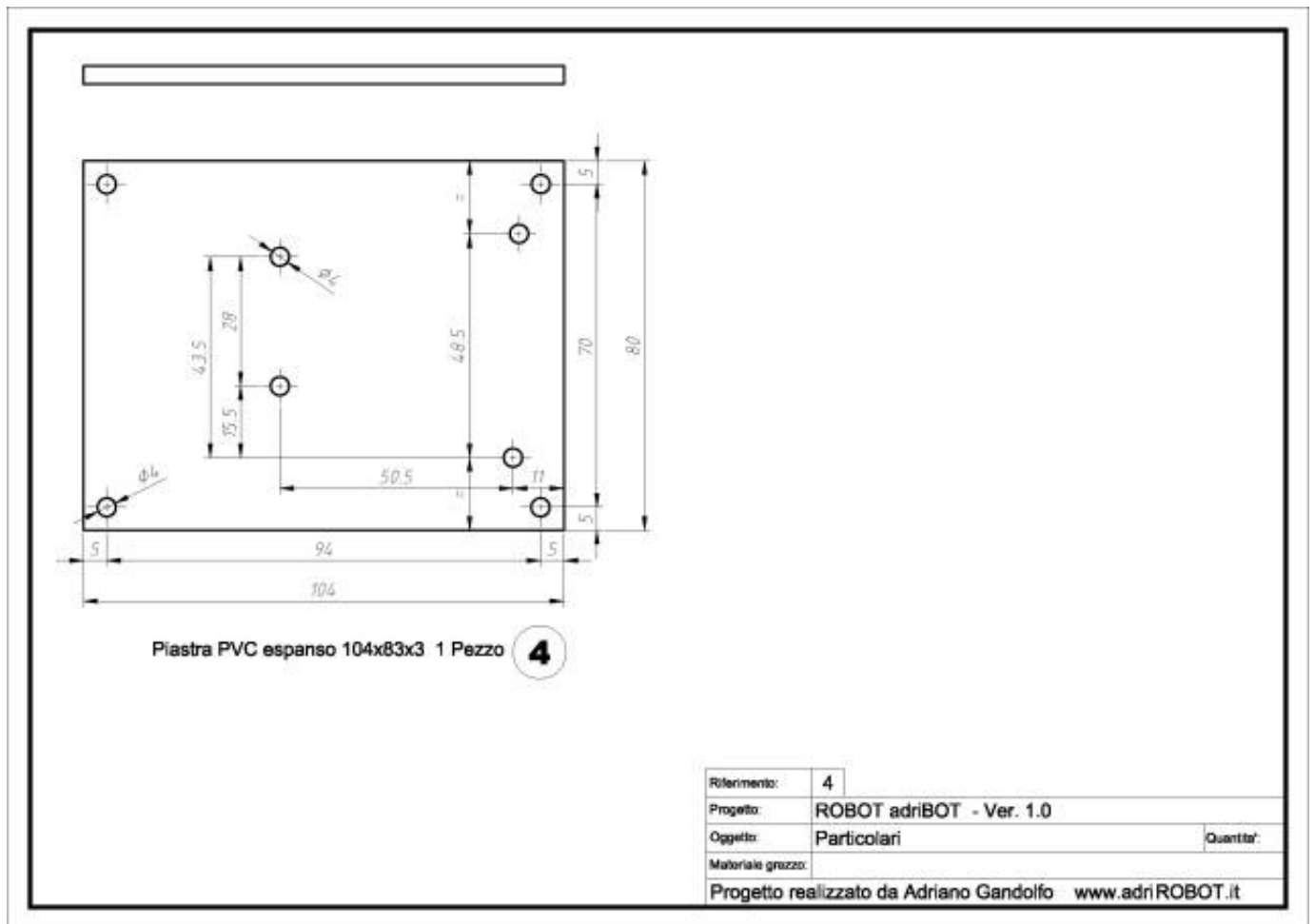
7 Ruota folle grande - 2 Pezzi

8 Ruota folle piccola - 6 Pezzi
Ruota motrice - 2 Pezzi

9 Ruota dentata grande - 2 Pezzi

Riferimento:	5-6-7-8-9
Progetto:	ROBOT adriBOT - Ver. 1.0
Oggetto:	Particolari
Materiale grezzo:	
Quantita':	

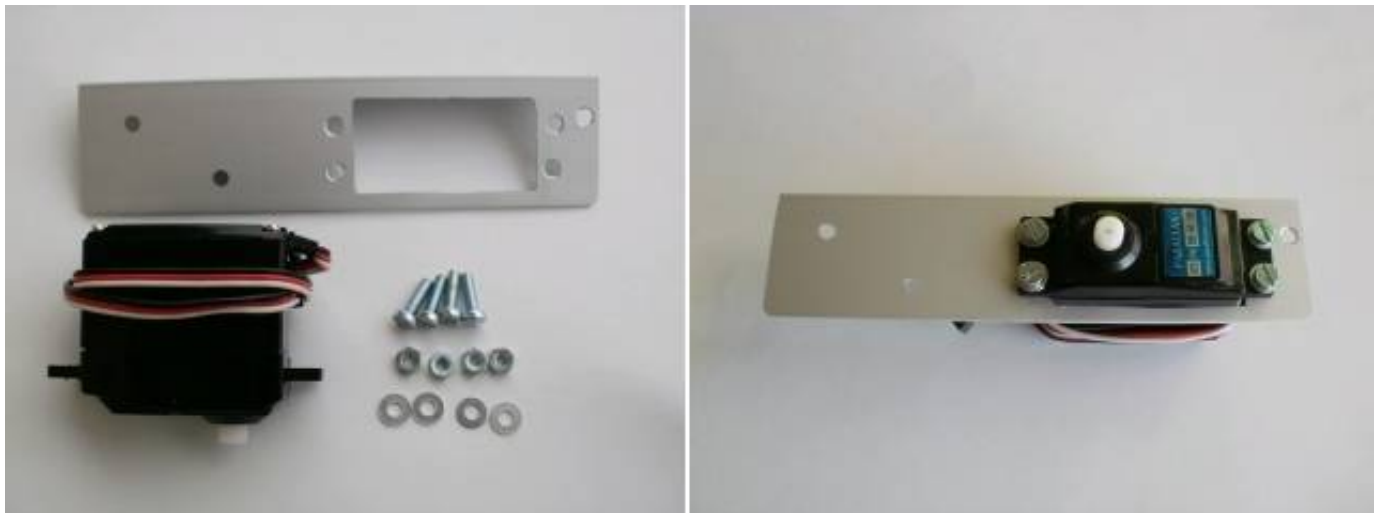
Progetto realizzato da Adriano Gandolfo www.adriROBOT.it



Assemblaggio dei particolari meccanici

Per il fissaggio dei vari particolari sono inoltre necessari:

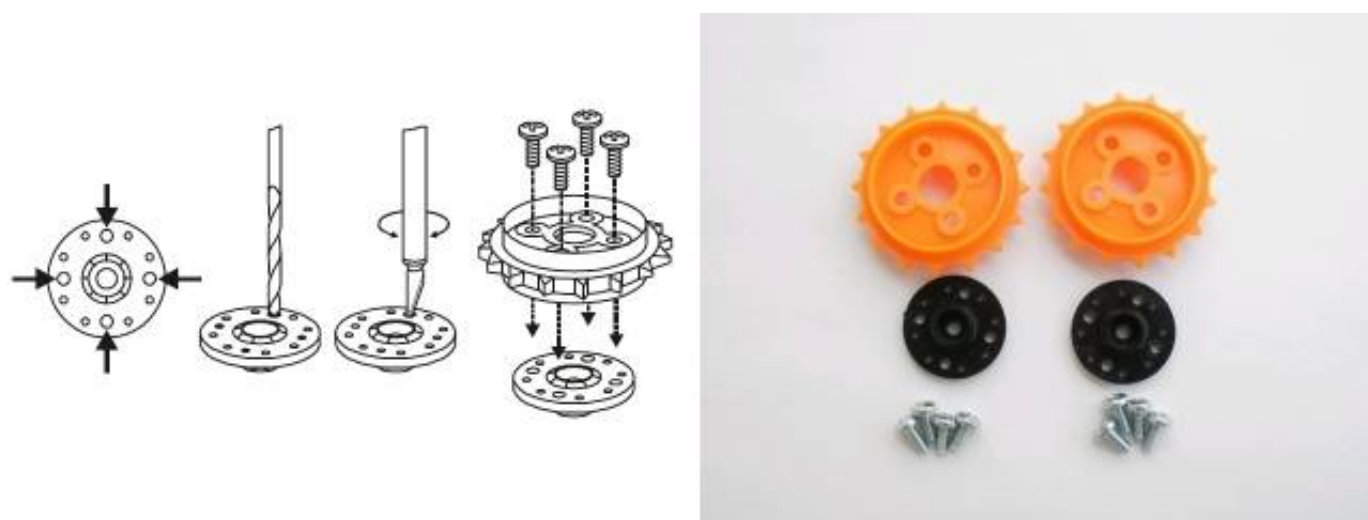
un cacciavite con la punta a taglio e uno con la punta a croce
 un paio di pinze



Il montaggio incomincia con le traverse laterali [2], i due servomotori a rotazione continua [5] ognuno mediante :n°4 viti M3x8 [16], n°4 dadi M3 [15] e n°4 rondelle piane $\phi 3,2$ [19].

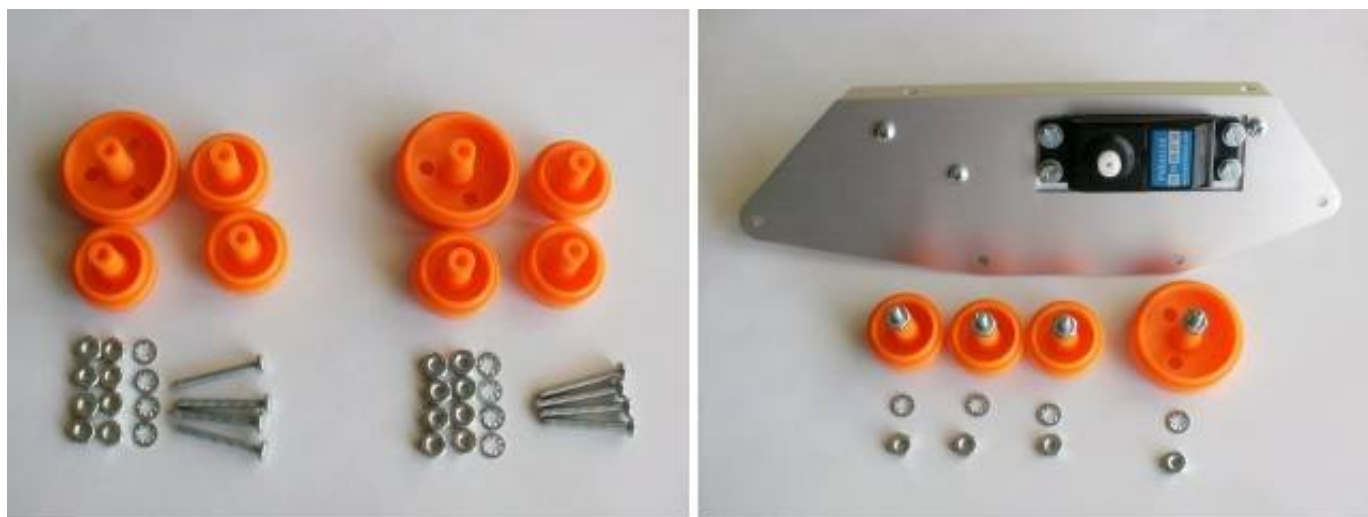


Si fisserà quindi al telaio laterale completo di servomotore la piastra di alluminio laterale **[1]** utilizzando 3 viti lunghe M3x10 **[19]** e 4 dadi M3 **[15]**.



Per il fissaggio delle ruote motrici **[9]** al servomotore, occorre allargare quattro fori presenti sulle squadrette che saranno state fornite con i servomotori **[5]**, i fori dovranno essere più piccoli della vite in modo che, quando saranno inserite le viti queste possano fare presa. È possibile eseguire questi fori con una punta di trapano, oppure utilizzando la lama di un cutter per ingrandire i fori come mostrato nella figura.

Se si utilizza il cutter, assicurarsi di non ingrandire eccessivamente i fori, magari mediante l'alesatura con il coltello sia sul lato superiore che inferiore, in questo modo il foro sarà più omogeneo.

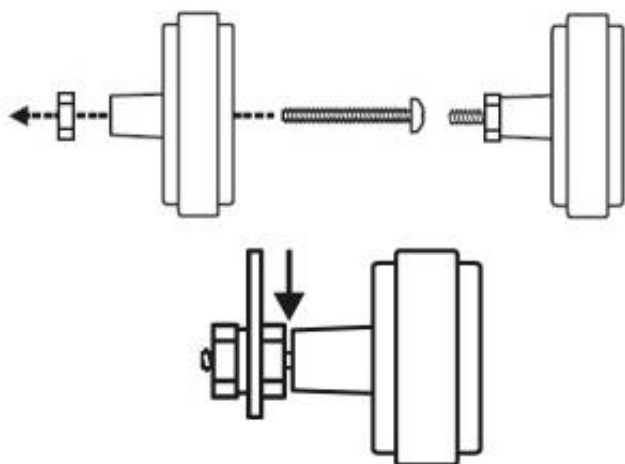


Si passerà poi al fissaggio delle ruote folli necessarie per la guida del cingolo per ogni lato del robot.

Sono necessarie una ruota grande **[8]**, 3 medie **[7]**, n°4 viti M3x25 **[17]** complete di dadi M3 **[15]** e rondelle antisvitamento **[21]**.



Si potranno unire le squadrette **[9]** al pignone mediante le viti autofilettanti **[18]**, quindi si fisseranno le varie ruote folli.



Per il fissaggio delle ruote si procederà serrando il dado sulla ruota poi allentandolo di 1/4 o

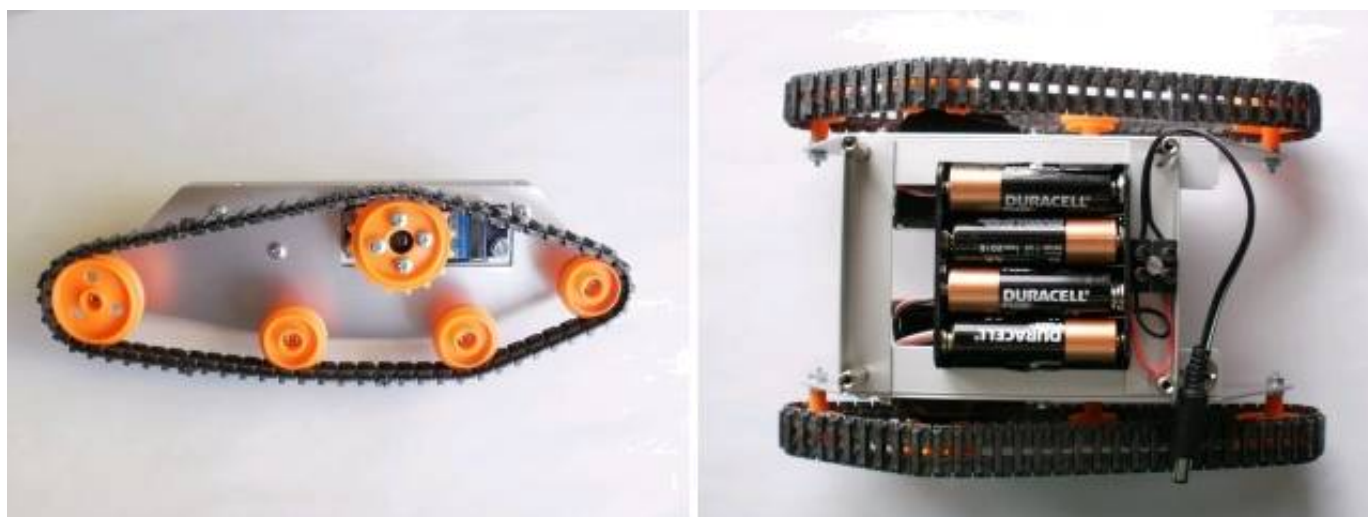
di 1/3 di giro.

Il gruppo ruota va collegato al telaio con un dado e una rondella di bloccaggio all'interno del telaio.

Quando i due dadi su ciascuna ruota sono serrati, controllare che ci sia un piccolo spazio tra la ruota e il dado esterno per consentire una facile rotazione, come mostrato nella figura.



Il montaggio dei due cingoli **[6]** richiede 66 maglie per cui sono necessari: n°1 pezzo da 30 maglie, n°2 pezzo da 10 maglie, n° 2 pezzo da 8 maglie.



A questo punto i due cingoli potranno essere installati sulle rispettive ruote, esercitando una leggera trazione sul cingolo stesso.

È necessario controllare che ogni ruota possa girare senza sforzo e senza incepparsi.

Se dovessero bloccarsi, allentando il dado e rimontando per mantenere il piccolo spazio come indicato precedentemente si risolverà il problema.

I due telai devono essere uniti con le traverse **[3]** fissando il tutto mediante i 4 distanziali esagonali L= 20 **[13]** e utilizzando 4 dadi M3 **[15]**.

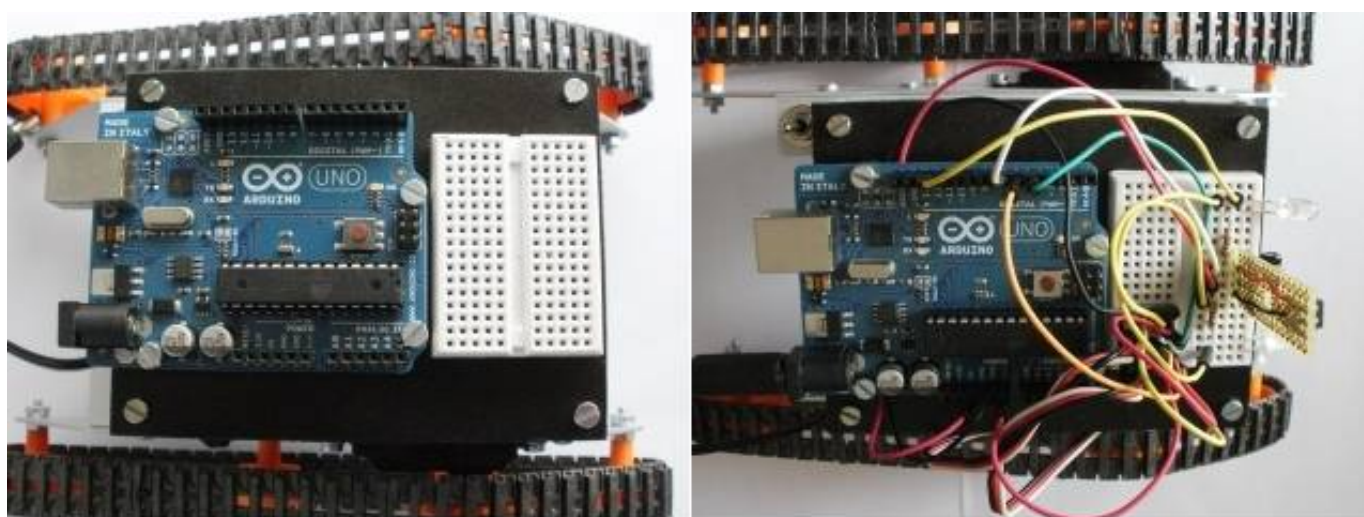
Si fisserà quindi il portabatteria **[10]** e l'interruttore **[24]** al telaio completando il cablaggio mediante la morsettiera a mammut **[23]**, il tutto terminerà con il connettore polarizzato

[25] che andrà poi inserito nella presa DC di Arduino.

Si noti che il cavo positivo della batteria andrà collegato al pin centrale del connettore polarizzato.

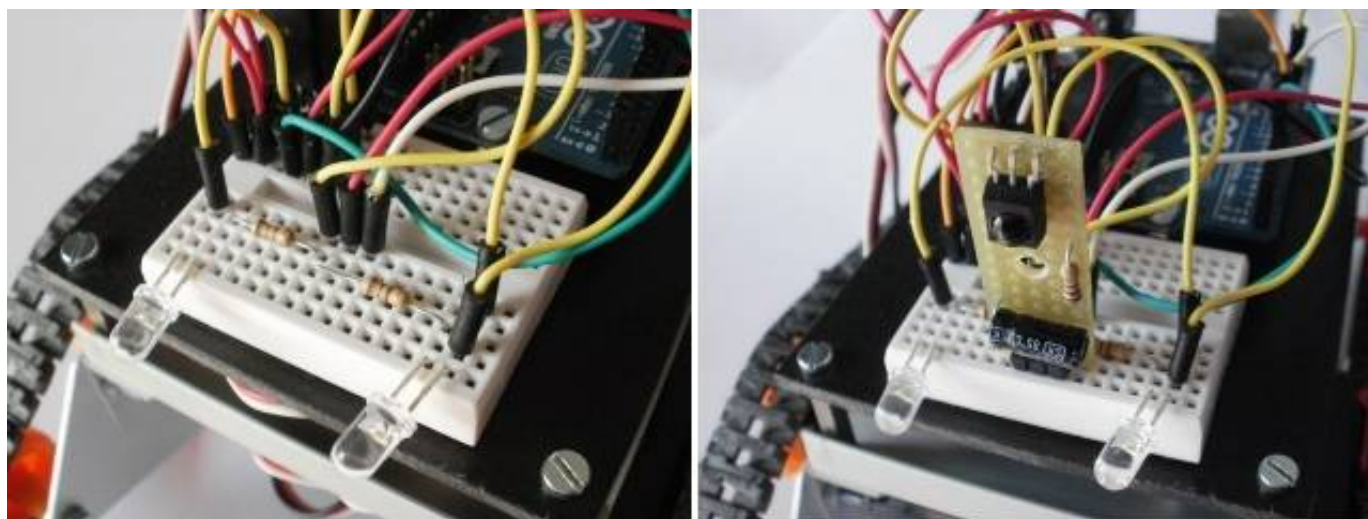


È possibile montare adesso la piastra supporto scheda **[4]** con n° 4 viti M3x8 **[16]** su cui si fisseranno 4 distanziali esagonali L=10 mm **[14]** fissati con 4 dadi M3 **[15]**. La scheda Arduino UNO **[12]** sarà fissata con n° 4 viti M3x8 **[16]**.

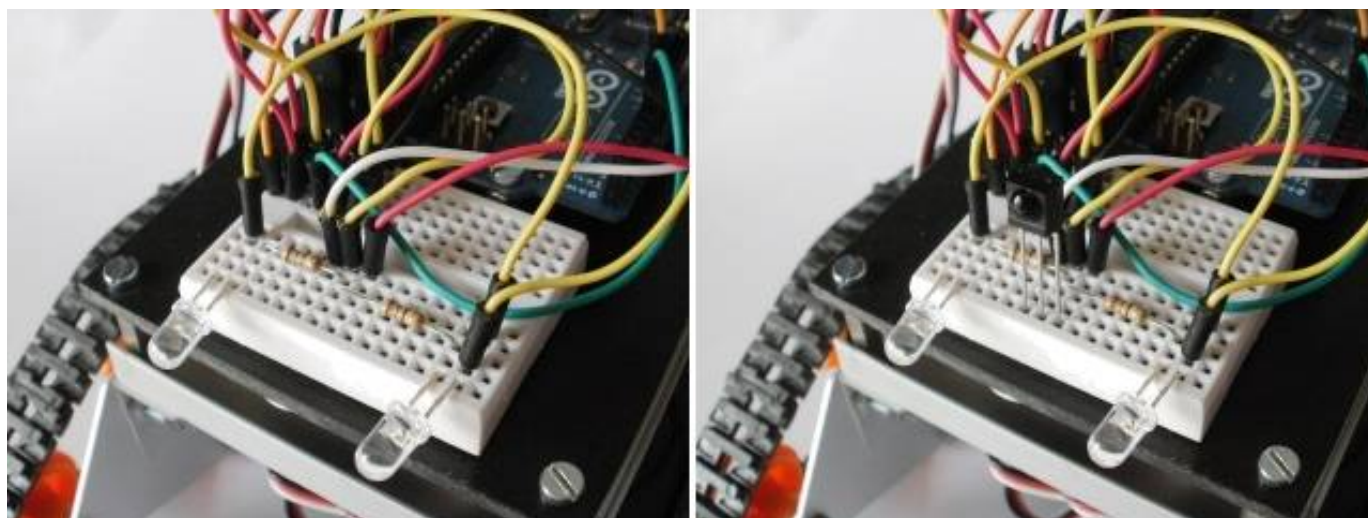


La mini breadboard **[11]** si fisserà alla piastra mediante la sua base autoadesiva.

Si passerà quindi al cablaggio seguendo lo schema riportato nel prossimo paragrafo.



Nella foto è riportato lo schema di collegamento del modulo sensore [22], che prevede il contatto di alimentazione centrale, quello di GND sulla sinistra e l'uscita del segnale sulla destra

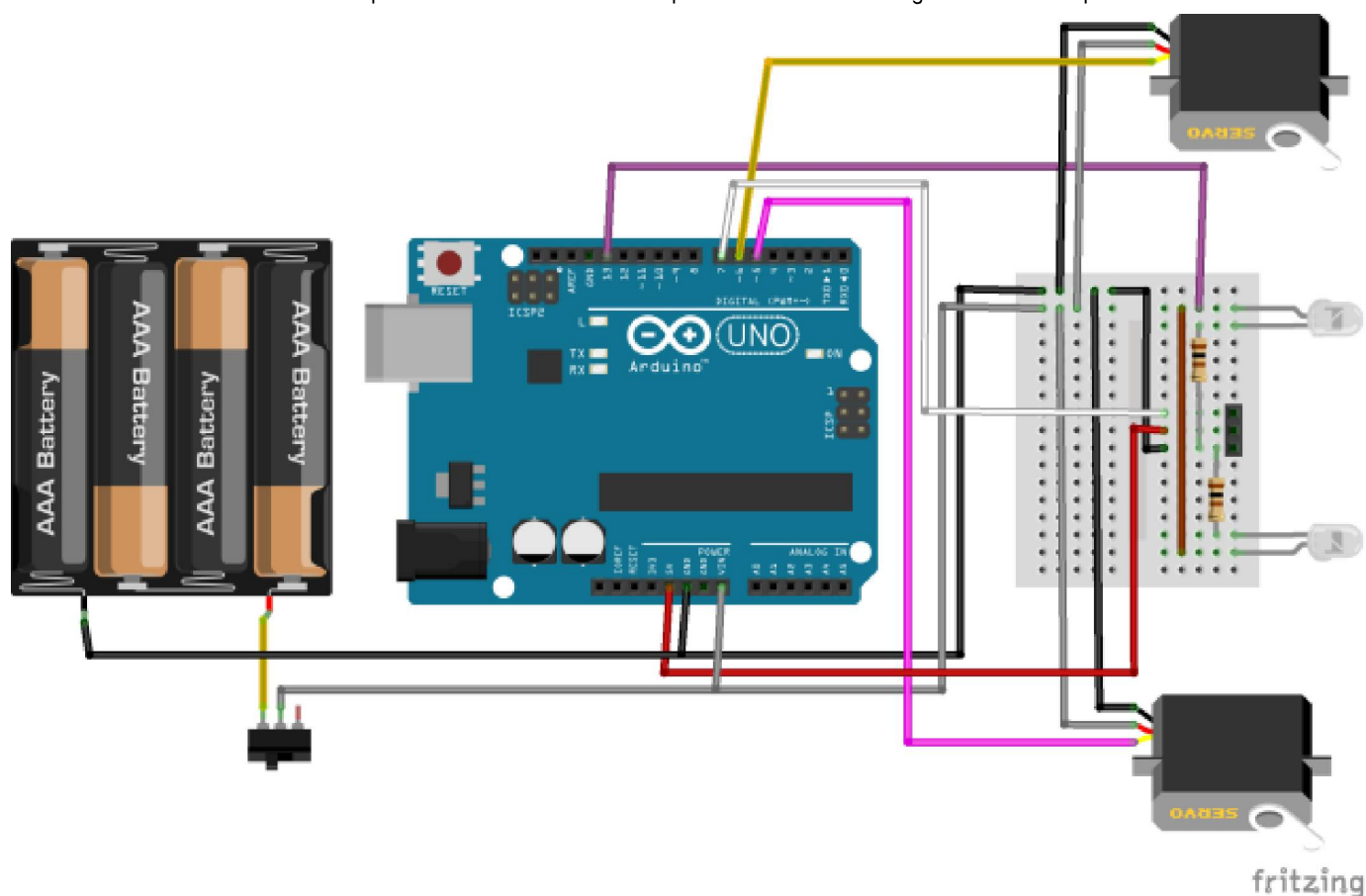


Nella foto è riportato invece lo schema di collegamento del semplice sensore, che possiede il contatto di alimentazione sulla destra, quello di GND al centro e l'uscita del segnale sulla sinistra.

Schema di collegamento

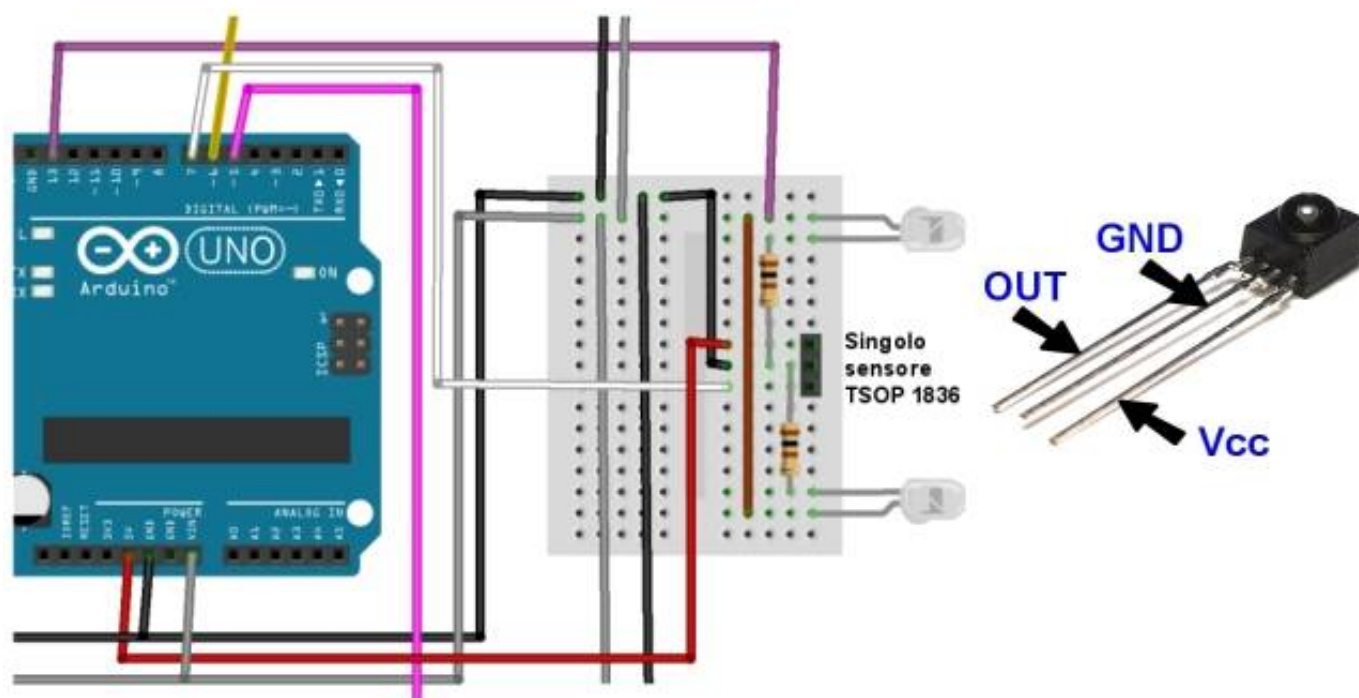
Nella figura sotto riportata sono presenti i collegamenti da eseguire: l'alimentazione dei servomotori è fornita dalle batterie derivate da Vin, mentre il sensore e il led sono alimentati a +5V.

I segnali sono forniti/letti dai pin digitali 5 e 6 per i servomotori, 7 per il sensore, 13 per i led.



fritzing

Nel caso si preferisse l'utilizzo del solo sensore **TSOP1836**, occorrerà modificare leggermente il cablaggio in quanto, rispetto al modulo, il pin di alimentazione è a sinistra, il pin di massa è al centro e l'uscita del segnale è a sinistra.



Libreria IRemote

Per la gestione del sensore IR occorre fare riferimento al lavoro fatto da **Ken Shirriff** [12] per cui occorrerà prima di tutto, scaricare **l'apposita libreria** [13].

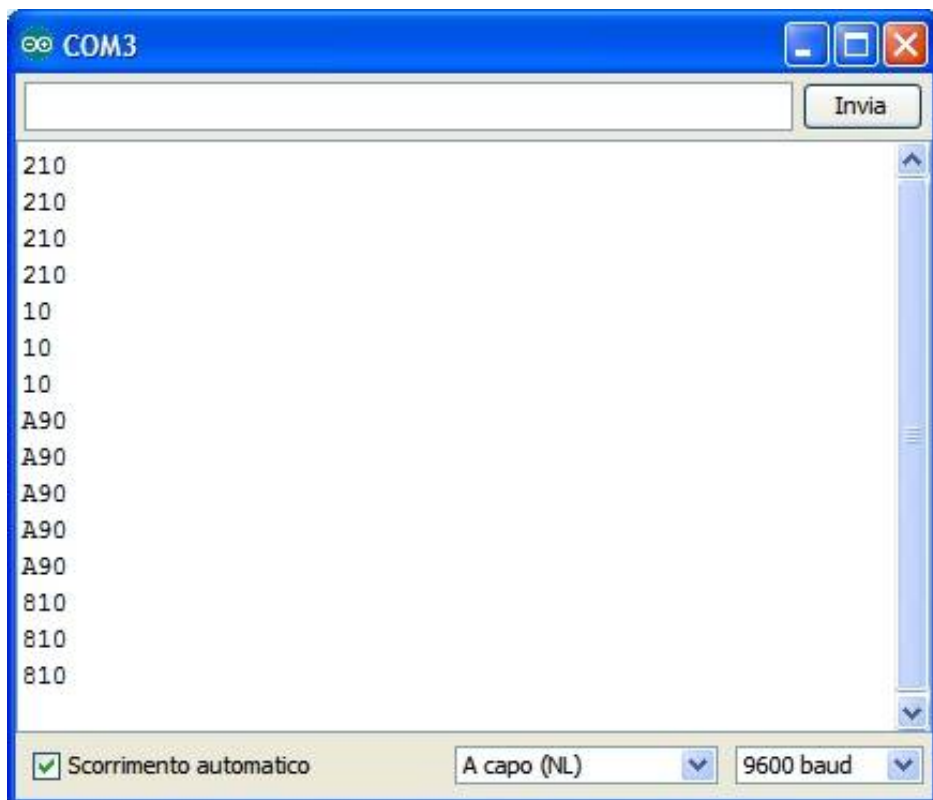
Il primo programma da utilizzare è quello necessario per la verifica del funzionamento del

sensore e per la determinazione del codice a cui sono associati i tasti del telecomando, quindi si caricherà lo sketch **IRrecvDemo** che si trova tra gli esempi della **libreria IRremote**.

Nel programma occorrerà modificare il numero della porta cui è collegati il sensore:

```
int RECV_PIN = 7; Pin a cui è collegato il sensore
```

a questo punto attivando il monitor seriale, si potrà vedere il numero di codice del comando inviato dal telecomando.



Resta inteso che la libreria può essere utilizzata sia per la trasmissione che per la ricezione e che mette a disposizione i seguenti comandi.

Comandi disponibili in Ricezione	Comandi disponibili in Trasmissione
IRrecv irrecv(receivePin) irrecv.enableIRIn() irrecv.decode(&results) irrecv.resume() irrecv.blink13(true)	IRsend irsend; irsend.sendNEC(IRcode, numBits); irsend.sendSony(IRcode, numBits); irsend.sendRC5(IRcode, numBits); irsend.sendRC6(IRcode, numBits); irsend.sendRaw(rawbuf, rawlen, frequency);

Nota: non tutti i tipi di telecomando sono riconosciuti dalla libreria, i modelli devono essere conformi alla codifica: NEC, Sony, RC5, RC6.

```
#include <IRremote.h>
int RECV_PIN = 7; Pin a cui è collegato il sensore
IRrecv irrecv(RECV_PIN);
decode_results results;
```



```

void setup()
{
  Serial.begin(9600);
  irrecv.enableIRIn(); // Start the receiver
}
void loop() {
  if (irrecv.decode(&results)) {
    Serial.println(results.value, HEX);
    irrecv.resume(); // Receive the next value
  }
}

```

È possibile scaricare il **programma già modificato**, presente negli allegati.

Il programma di gestione del robot.

Una volta determinato a quali codici corrispondono i tasti del proprio telecomando, si caricherà il programma di gestione che **potrete trovare negli allegati**.

Il programma prevede l'utilizzo, oltre che della **libreria IRemote**, anche la **libreria Servo** per la gestione dei servomotori, questa è già disponibile in quanto è tra le librerie standard dell'IDE.

Per i comandi della libreria servo si potrà far riferimento all'apposita **sezione presente sul sito Arduino** [14].

Il programma prevede prima di tutto la configurazione dei servomotori e l'inizializzazione del ricevitore IR.

Nel loop viene letto il codice decodificato dal sensore e in base al valore, è attivata la relativa routine di azionamento dei due servomotori.

Per le varie assegnazioni di codici sarà necessario modificare l'istruzione

```
if ( results.value == 0xA90 )
```

e sostituire il valore **A90**, che corrisponde alla pressione del tasto **STOP** per il telecomando utilizzato, quello del telecomando in vostro possesso.

È inoltre possibile inserire nuovi comandi semplicemente inserendo altre linee.

```

if ( results.value == 0XXXXX ){
  NUOVA ROUTINE COMANDO();
}

```

NOTA:

Per il suo funzionamento, la libreria IrRemote utilizza un Timer sia per la trasmissione che per la ricezione.

Se si desidera utilizzare un'altra libreria che richiede lo stesso Timer, è necessario modificare **IRremoteInt.h** per fare in modo che **IrRemote** utilizzi un timer differente.

Per questo motivo, per esempio, il comando **Tone()** ^[15] per la creazione di suoni tramite buzzer, senza le opportune modifiche, non funzionerà.

```
#include <Servo.h> // Caricamento libreria per la creazione degli
                //oggetti Servo che generano segnali PWM
Servo leftDrive; // Creazione per controllo servomotore sinistro
Servo rightDrive; // Creazione per controllo servomotore destro

#include <IRremote.h> // Caricamento libreria per gestione sensore IR
int receiver = 7; // Definizione del pin a cui è collegato il sensore IR
IRrecv irrecv(receiver); // Inizializzazione della libreria IRemote
decode_results results; // Decodifica del risultato che Arduino riceve dal
sensore
// per ottenere un valore numerico utilizzabile nel confronto.
#define ledpin 13 // Definizione del pin a cui sono collegati i led di
illuminazione

void setup()
{
  leftDrive.attach(5); // attribuisce il servo sul pin 5 all'oggetto servo
  rightDrive.attach(6); // attribuisce il servo sul pin 6 all'oggetto servo
  pinMode(ledpin, OUTPUT); //Imposta il pin dei led come Output
  irrecv.enableIRIn(); // si utilizza il metodo irrecv.enableIRIn(
  // della libreria IRemote perchè legga i valori provenienti dal sensore
  Serial.begin(9600); // Imposta la velocità della seriale per controllo
}

void loop()
{
  if (irrecv.decode(&results)) // È stato ricevuto un segnale IR?
  {
    Serial.println(results.value, HEX); // Mostra eventualemnte sul monitor
seriale
    // il valore esadecimale ricevuto
    irrecv.resume(); // prosegue mettendo il sensore nuovamente in modalità di
ascolto
  }
  if ( results.value == 0x810 ){
    driveForward();
  } // a seconda dei valori decodificati esegue una diversa routine

  if ( results.value == 0x210 ){
```

```
    driveBackward();
}
if ( results.value == 0xA90 ){
    STOP();
}
if ( results.value == 0x410 ){
    turnLeft();
}
if ( results.value == 0xA10){
    turnBackLeft();
}
if ( results.value == 0x10 ){
    turnRight();
}
if ( results.value == 0xC10 ){
    turnBackRight();
}
if ( results.value == 0xE10 ){
    digitalWrite(ledpin, HIGH);
} //Accende i led frontali
if ( results.value == 0x910 ){
    digitalWrite(ledpin, LOW);
} //Spegne i led frontali
}
//Routine azionamento
void turnRight() //Il robot gira a destra
{
    leftDrive.write(180);
    rightDrive.write(180);
}
void turnBackRight() //Il robot indietro a destra
{
    leftDrive.write(90);
    rightDrive.write(0);
}
void turnLeft() //Il robot gira a sinistra
{
    leftDrive.write(0);
    rightDrive.write(0);
}
void turnBackLeft() //Il robot gira indietro a sinistra
{
```

```
    leftDrive.write(180);
    rightDrive.write(90);
}
void driveForward() //Il Robot va avanti
{
    leftDrive.write(0);
    rightDrive.write(180);
}
void driveBackward() //Il robot va indietro
{
    leftDrive.write(180);
    rightDrive.write(0);
}
void STOP()// Il robot si ferma
{
    leftDrive.write(90);
    rightDrive.write(90);
}
```

Filmato illustrativo

Il filmato mostra l'utilizzo del robot.

Robot cingolato controllato da telecomando TV



Conclusioni

Abbiamo approfondito in quest'articolo l'uso di un sensore di raggi infrarossi per decodificare gli impulsi forniti da un telecomando.

Si è inoltre realizzato un robot, che potrà essere potenziato con altri sensori che potranno permettergli, per esempio, di muoversi autonomamente su terreno.

In rete potrete trovare innumerevoli esempi, magari con l'utilizzo di sensori sonar, per cui lasciamo al lettore eventualmente proporre modifiche e nuove implementazioni.

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

URL to article: <https://it.emcelettronica.com/telecomando-tv-comandare-robot-cingolato>

URLs in this post:

- [1] Bluetooth.: <https://it.emcelettronica.com/elettronica/etichette/bluetooth>
- [2] Arduino: <https://it.emcelettronica.com/arduino>
- [3] **Max Power IR LED Kit**: <https://www.sparkfun.com/products/10732>
- [4] Tamiya 70100 Track and Wheel Set: <https://www.sparkfun.com/products/321>
- [5] **due servomotori Parallax**: <http://www.parallax.com/product/900-00008>
- [6] TSOP 4836: https://it.emcelettronica.com/files/TSOP_4836.pdf
- [7] **da Conrad codice 171107-62**: <http://www.conrad.it/ce/it/product/171107/?insert=62&insertNoDeeplink&productname=Ricevitore-IR-Forma-speciale-assiali-36-kHz-950-nm-Vishay-TSOP-4836>
- [8] prevedono dei pin di collegamento:
<http://www.conrad.it/ce/it/product/181819/?insert=62&insertNoDeeplink&productname=Ricevitore-IR-36-kHz-950-nm-Conrad-IR-R-36>
- [9] **un proprio CS**: https://it.emcelettronica.com/files/modulo_infrarossi_rx.pdf
- [10] **scheda Arduino UNO**: <https://it.emcelettronica.com/7-progetti-da-fare-con-arduino>
- [11] **questa risorsa della rete**:: <http://ledcalculator.net/default.aspx?values=5,3.2,19,2,0&lang=it-IT>
- [12] Ken Shirriff: <http://www.righto.com/2009/08/multi-protocol-infrared-remote-library.html>
- [13] **l'apposita libreria**: <https://github.com/shirriff/Arduino-IRremote>
- [14] sezione presente sul sito Arduino: <http://arduino.cc/en/reference/servo>
- [15] **Tone ()**: <http://arduino.cc/en/reference/tone>

Copyright © 2017 Elettronica Open Source. All rights reserved.