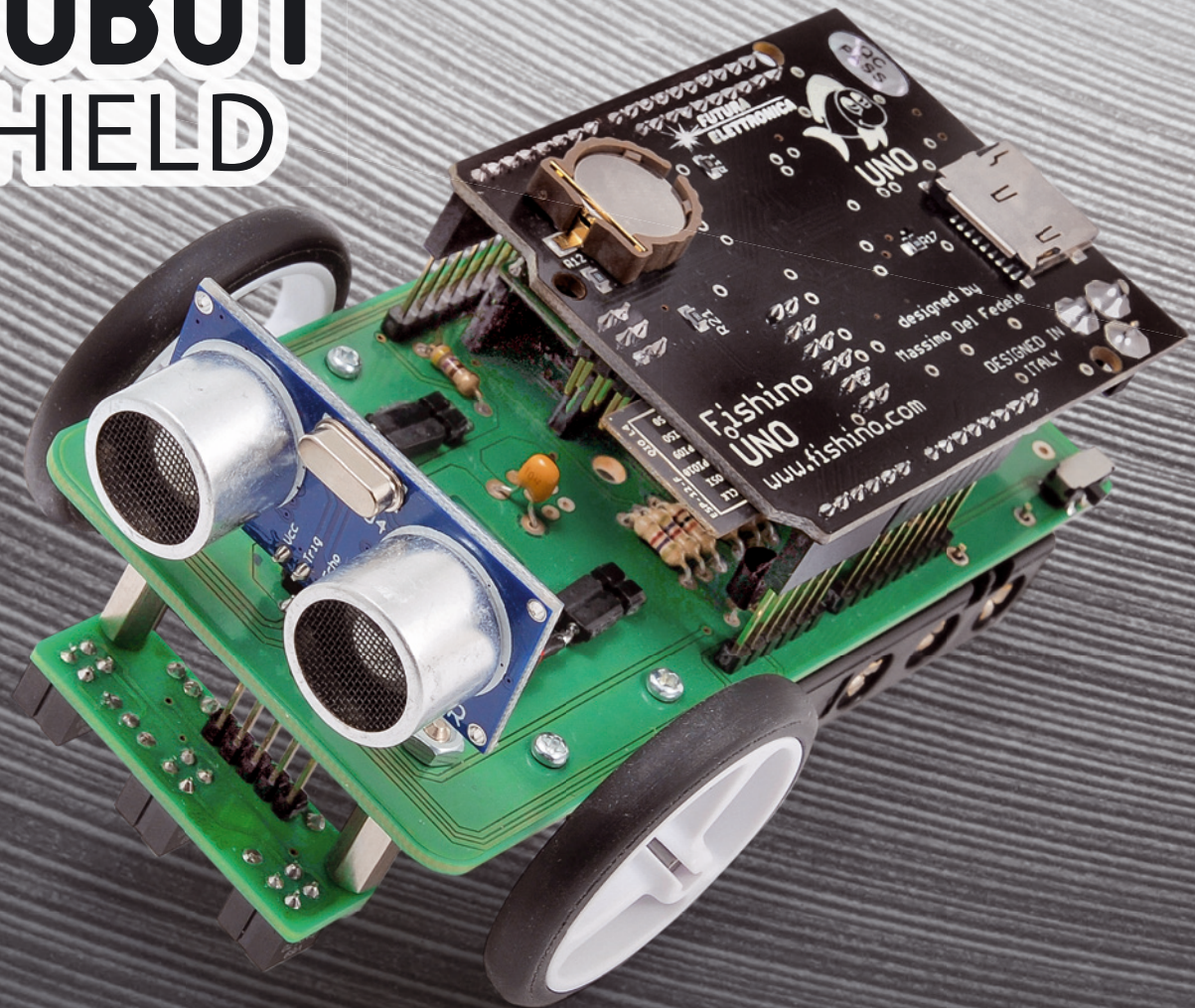


EASY ROBOT SHIELD

Piattaforma per sviluppare un piccolo robot su ruote controllabile tramite uno smartphone collegato in WiFi.

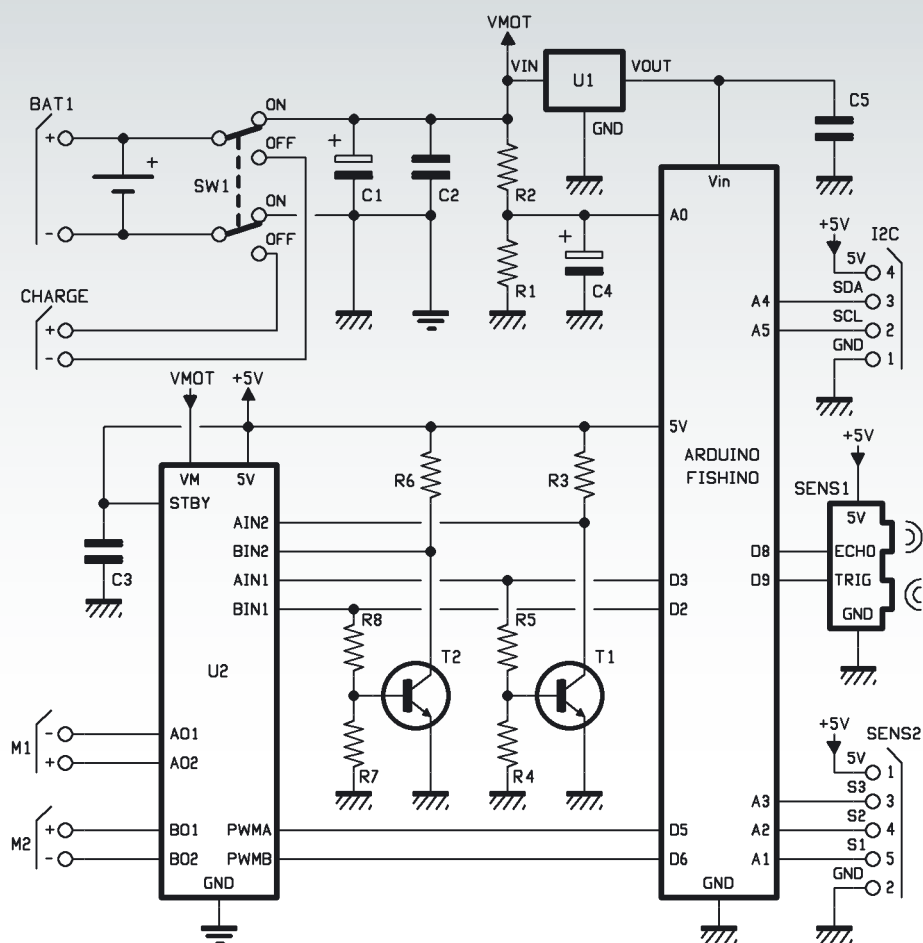


dell'Ing. MIRCO SEGATELLO

Non è la prima volta che presentiamo un robot su ruote basato su Arduino, ma questa volta, oltre a proporvi una nuova variante di facile utilizzo (non a caso utilizziamo il termine Easy Robot...) più che un robot vogliamo introdurre uno shield molto particolare, perché, come vedremo nel corso dell'articolo, costituirà sia la parte elettrica che meccanica: uno shield in verità anomalo, perché ha una particolare sagoma e dimensioni adatte a fare da supporto elettrico e meccanico all'intero robot, compresa una scheda Arduino, che sarà il

cervello. Questo modo di procedere consente di abbattere i costi di produzione e semplificare notevolmente la sua realizzazione, permettendo anche ai meno esperti di cimentarsi con la robotica. Il valore aggiunto in questo progetto è la piena compatibilità con le schede Arduino, ma anche con schede di terze parti come le nostre Fishino. Su quest'ultima si concentrerà il nostro lavoro, giacché proporremo un'applicazione pratica proprio con questo tipo di scheda, che, lo ricordiamo, oltre a svolgere le normali funzioni di una sche-

[schema elettrico **SHIELD**]



da Arduino è equipaggiata con un modulo WiFi (più un RTC e SD Card) quindi può interagire direttamente in modalità wireless con altri dispositivi, come ad esempio uno smartphone o un tablet.

SCHEMA ELETTRICO

Analizziamo ora lo schema elettrico, per capire come funzionano i diversi blocchi funzionali e i moduli componenti utilizzati. Premettiamo che per semplificare lo schema ci siamo affidati, per alcune sezioni, a moduli già pronti reperibili in commercio come il radar a ultrasuoni. L'insieme consta di due schede, ossia lo shield che fa da supporto a tutti i componenti, i moduli e la scheda Fishino, più il circuito del

motore a riflessione a tre stadi che fornisce al robot la capacità di inseguimento delle linee.

L'alimentazione dell'intero robot è ottenuta grazie all'utilizzo di un pacco di quattro pile tipo ministilo da 1,5 volt, in grado di fornire una tensione compresa tra 4,5 e 6,5 V, a seconda del livello di carica. Nulla vieta (anzi è caldamente consigliabile) di utilizzare delle batterie ricaricabili NiMH che, pur fornendo una tensione leggermente più bassa, consentono di essere riutilizzate previa la loro ricarica.

L'alimentazione a batterie è la scelta elettiva per un robot che debba muoversi in libertà. La tensione fornita dalle quattro batterie è ottimale per alimenta-

re, senza alcuno stress, i piccoli motoriduttori ma risulta insufficiente per alimentare correttamente una scheda Arduino o Fishino, pertanto utilizziamo un boost converter per innalzare la tensione ad un valore di circa 7V, indipendentemente dal livello di carica delle batterie. Si tratta di un piccolissimo convertitore DC/DC di tipo step-up, in grado di elevare una tensione continua in ingresso, compresa tra 1,8 e 8V, in una tensione regolabile di 9V, impostando una tensione in uscita del valore di 9V, saremo sicuri di alimentare correttamente la scheda Arduino, rispettando il vincolo imposto dalle caratteristiche del modulo.

Per la lettura del livello di tensione delle batterie è stato previsto un partitore di tensione, in grado di fornire a Fishino una tensione dimezzata rispetto a quella

Tabella 1 - Disposizione pin dello shield.

pin	nome	funzione
D0	RX	RX Arduino
D1	TX	TX Arduino
D2	DIRB	Direzione motore B
D3	DIRA	Direzione motore A
D4	---	usato da Fishino
D5	PWMA	PWM motore A
D6	PWMB	PWM motore B
D7	---	usato da Fishino
D8	ECHO_PIN	Sens. ultrasuoni echo pin
D9	TRIG_PIN	Sens. ultrasuono Trigger pin
D10	---	usato da Fishino
D11	---	usato da Fishino
D12	---	usato da Fishino
D13	---	usato da Fishino
A0	VBAT	Misura tensione batteria
A1	S1	Sensore di linea 1
A2	S2	Sensore di linea 2
A3	S3	Sensore di linea 3
A4	---	A disposizione per ampliamenti
A5	---	A disposizione per ampliamenti

fornita dalle batterie; a livello software convertiremo il livello di tensione in un valore numerico in percentuale, utile a rilevare lo stato di carica della batteria. Il driver impiegato per controllare i motori è fornito dalla Futura Elettronica e identificato dal codice 7300-TB6612FNG; tale driver è in grado di gestire sino a due motori contemporaneamente, con correnti sino a 1A e tensione di alimentazione da 4,5 a 13,5 Vcc. Il driver e quindi i motori, sono alimentati direttamente dalla batterie, così da svincolarli dalla limitazione della massima corrente di uscita fornita dal convertitore di tensione, che invece alimenterà solo la scheda di controllo.

Per gestire il modulo driver servono sei linee di controllo, che ridurremo a quattro grazie all'utilizzo di due BJT utilizzati per invertire il segnale su due linee; in questo modo, per ciascun motore disporremo della linea per il PWM, che controllerà la potenza sul motore, ed una linea digitale per impostare il verso di rotazione.

I SENSORI

Per quanto riguarda la sensoristica abbiamo previsto un sensore a ultrasuoni per identificare la distanza alla quale si trovano eventuali ostacoli di fronte al

Tabella 2 - Elenco materiale.

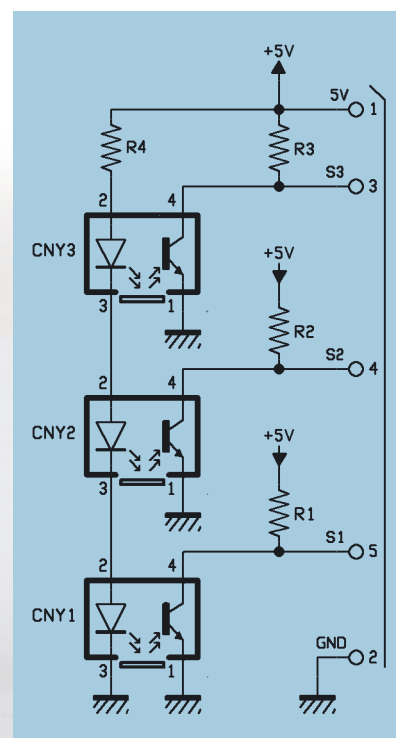
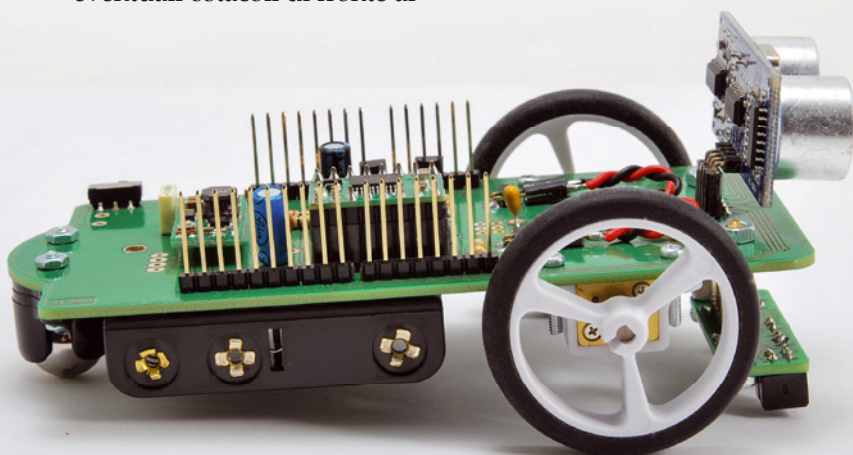
Componente	Codice
Micro motoriduttore in metallo 85 RPM	1686-MMG150
Coppia di supporti per motoriduttori MMG298	7300-MMGSUPP
Coppia di ruote pololu 40x7mm per motoriduttore mmg	8218-RP407
Ball caster con sfera in metallo da 1,2 cm	7300-PBCASTER12
Convertitore DC/DC uscita regolabile da 2,5 a 9,5V	7300-ABR2595
Sensore ultrasuoni 2=450 cm	2846-MISDIST04
Portabatteria 4 ministilo (AAA) con terminali a saldare (L x L x A) 52 x 49 x 13 mm	BH441P

robot. Si tratta di un economico modulo SONAR (radar a ultrasuoni codice 2846-MISDIST04) che è in grado di eseguire misure di distanza comprese tra 2 e 450 cm; per il suo controllo sono necessarie due linee digitali, una per il segnale di trigger ed una per il segnale echo, la cui durata è proporzionale alla distanza dell'oggetto; a tale riguardo va detto che l'uscita (echo) fornisce un impulso quando al sensore viene richiesto attraverso un impulso logico a livello alto della durata di 10 μ s sul trigger: il sensore trasmette otto impulsi a 40 kHz, che riflessi tornano alla capsula ricevente e sul pin di uscita echo viene fornito un impulso la cui durata rappresenta il ritardo e, di riflesso, la distanza. Ad esempio un impulso da 150 μ s corrisponde a 2 cm e uno da 25 ms a 4,5 m. Se non rileva alcun ostacolo entro i 4,5 m, il SONAR produce un impulso di durata di

circa 38 ms. In generale la distanza è pari alla durata dell'impulso in μ s diviso 58.

Il rilevamento è valido frontalmente o con un'angolazione di $\pm 15^\circ$ rispetto alla perpendicolare al sensore.

Oltre al SONAR abbiamo previsto tre sensori di linea ad infrarossi basati sull'affermato sensore a riflessione CNY-70 (codice 7300-CNY70) che, nella configurazione da noi usata, forniscono in uscita una tensione inversamente proporzionale al livello di riflessione del suolo in cui si trova ad operare il robot: sfondi chiari forniranno tensioni basse e viceversa sfondi scuri forniranno valori alti. Questa è



[schema elettrico **SENSORE IR**]

Elenco Componenti:

R1, R2, R4, R7: 10 kohm

R3, R5, R6: 4,7 kohm

R8: 4,7 kohm

C1: 47 μ F 63 VL elettrolitico

C2, C3: 100 nF ceramico

C4: 1 μ F 63 VL elettrolitico

C5: 100 nF ceramico

T1, T2: BC547

U1: Modulo Boost DC Converter

U2: Driver motori (TB6612FNG)

SW1: Deviatore slitta 2 vie

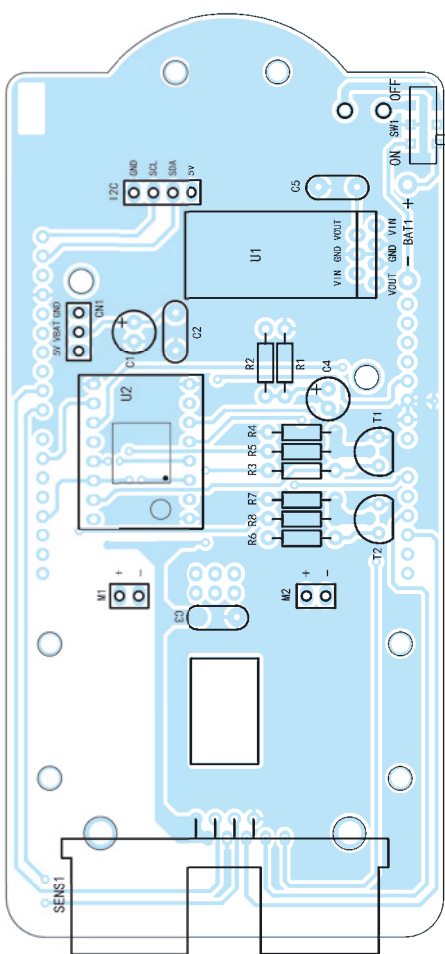
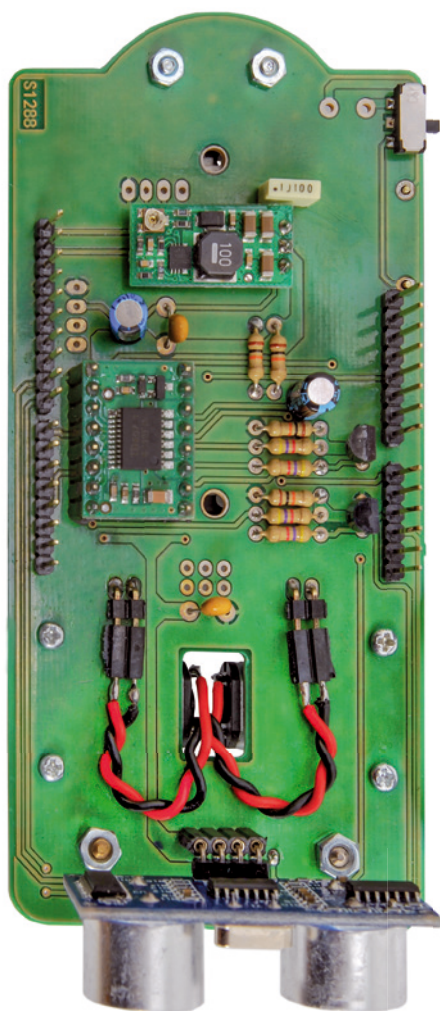
SENS1: Sensore ultrasuoni (MISDIST04)

SENS2: Modulo CNY70 (FT1289)

M1, M2: Micro Motoriduttore 6V (MMG150)

Varie:

- Strip maschio 2 vie 90° (2 pz.)
- Strip maschio 4 vie (2 pz.)
- Strip femmina 2 vie (2 pz.)
- Strip femmina 4 vie 90° (3 pz.)
- Strip femmina 4 vie (2 pz.)
- Strip maschio 6 vie 18 mm
- Strip maschio 8 vie 18 mm (2 pz.)
- Strip maschio 10 vie 18 mm (2 pz.)
- Porta Batteria 4xAAA da CS
- Ball Caster (PBCASTER12)
- Ruota 40x7 mm (2 pz.)
- Distanziale M/F 15mm (2 pz.)
- Vite 10 mm 3 MA (2 pz.)
- Dado 3 MA (4 pz.)
- Circuito stampato S1288 (58 x 123 mm)



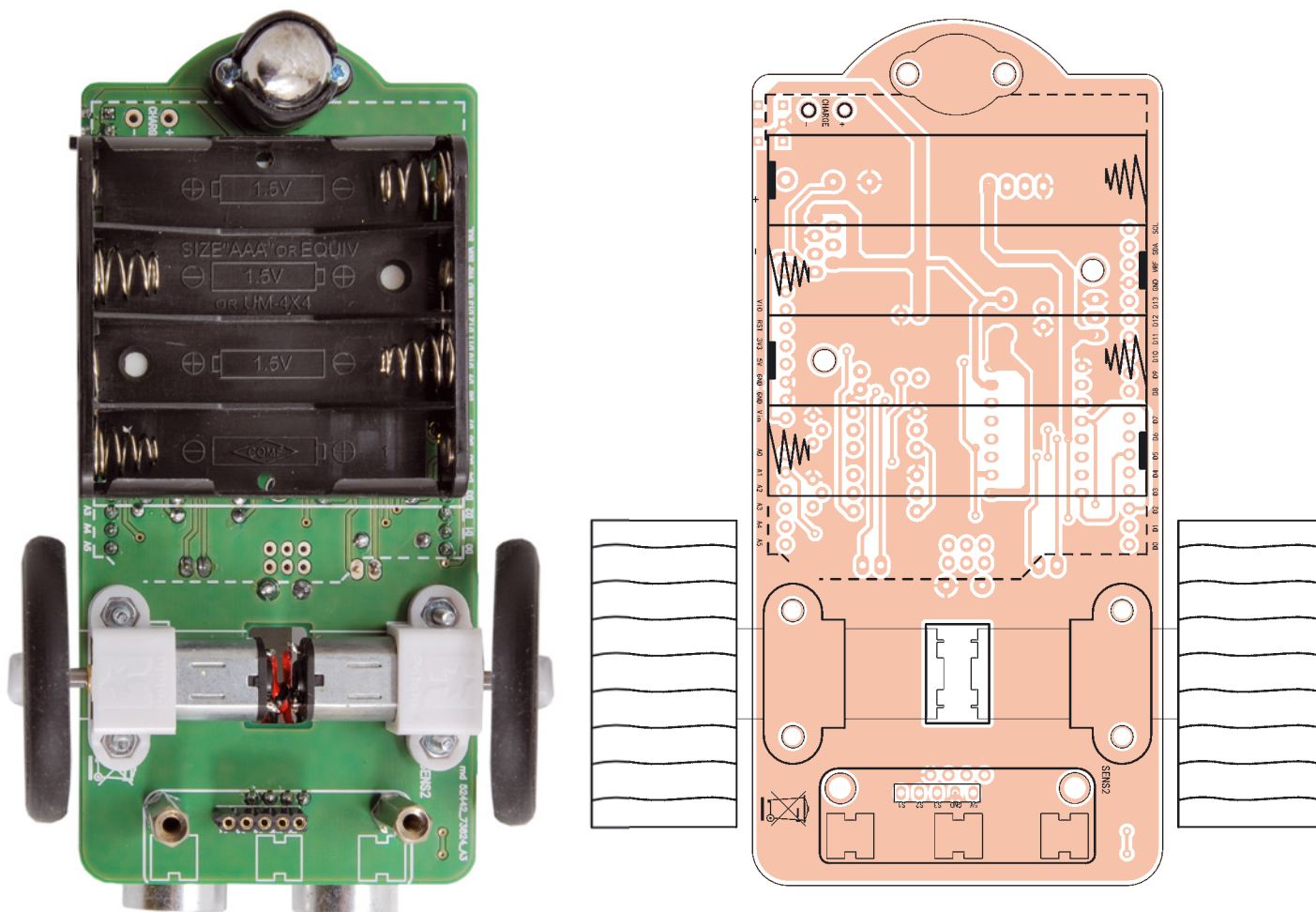
la classica funzionalità richiesta per realizzare robot inseguitori di linea o anche semplicemente per identificare il tipo di pavimento in cui si trova a operare il robot. Lo shield è stato pensato per essere utilizzato in abbinamento ad una scheda Fishino, pertanto i pin a disposizione per il nostro shield sono solo quelli non utilizzati dalla scheda di controllo. Alla fine solo i pin A4 e A5 non vengono usati e lasciati disponibili in un opportuno connettore, a disposizione per essere usati come generici pin digitali, come ingressi analogici oppure per comunicazioni I²C. Per completezza nella **Tabella 1**

riportiamo l'elenco dei pin utilizzati e la loro funzione.

REALIZZAZIONE PRATICA

La realizzazione del robot non comporta difficoltà, perché i componenti da saldare sono pochi, mentre i componenti SMD sono tutti forniti sotto forma di breakout board per renderne facile l'utilizzo. Una volta ottenuti i PCB (quello grande è lo shield e il piccolo il triplo sensore di linea) iniziate la saldatura dai componenti a più basso profilo proseguendo mano a mano con quelli più alti, aiutandovi con le immagini riportate in queste pagine e i piani di montaggio. Per

ultimi fissate il convertitore di tensione e il driver per i motori, direttamente oppure dotandoli di connettori strip, a seconda dell'altezza finale che vorrete ottenere; ricordate che sopra lo shield sarà posta la scheda Arduino rovesciata, ragion per cui è probabile che sia necessario montare gli elettrolitici sdraiati. Provvedete a tagliare con cura i reofori dei componenti sulla parte inferiore che sarà utilizzata per fissare il porta batteria. Terminata la saldatura di tutti i componenti, potete passare all'installazione dei componenti meccanici, infatti questo shield svolge anche la funzione di



chassis rendendo la realizzazione molto più semplice. Saldate due piccoli filetti sui contatti dei motori, meglio se terminati con un piccolo connettore ricavato da uno strip femmina; durante le prime prove sarà semplice invertire la polarità dei motori se non dovessero girare nel modo opportuno. Fissate i motori con il loro supporto negli appositi alloggiamenti avendo cura di allineare i mozzi delle ruote allo stampato. Fissate il porta batteria con due piccole viti e prevedete il collegamento elettrico, filo positivo e negativo di alimentazione. Il piccolo stampato sarà elettricamente e meccanicamente connesso

allo shield tramite una coppia di connettori strip maschio sullo stampato dei sensori e femmina sullo shield, per quest'ultimi esistono due altezze commerciali, 12 e 18mm; se usate quelli da 18 mm avrete un maggior spazio per i componenti saldati sullo shield. Completate il secondo stampato (quello più piccolo) con i tre sensori CNY70 solo dopo aver saldato le resistenze, fate molta attenzione all'orientamento dei componenti. I motori vanno fissati allo shield tramite due piccoli supporti (codice 7300-MMGSUPP). Installate le ruote sul mozzo dei motori e fissate il ball caster con

gli adeguati spessori in modo da garantire che il robot rimanga orizzontale, a questo punto avrete la precisa altezza da terra dello shield e quindi potete provvedere all'installazione dello stampato dei sensori di linea. La distanza

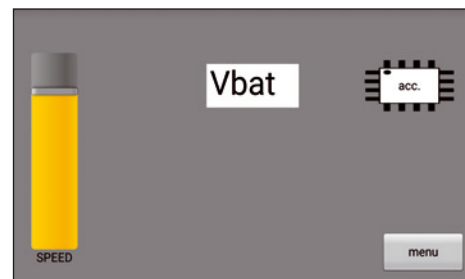


Fig. 1 - Creazione dell'interfaccia tramite l'app RoboRemo.

Listato 1

```
// Ricezione pacchetto dati dallo smartphone ed estrapolazione valori numerici
if (packetSize)
{
  int len = Udp.read(cmd, 255);
  if (len > 0) cmd[len] = 0;

  isConnected=1;

  if(cmd[0]=='s' && cmd[1]=='p' && cmd[2]=='e' && cmd[3]=='e' && cmd[4]=='d' && cmd[5]==' ') {
    speed = 0;
    if (cmd[6]=='-')
      {
        for(int i=7; cmd[i]!='\n'; i++) {
          speed = speed*10 + cmd[i]-'0';
        }
        speed = -speed;
      }
    else
      {
        for(int i=6; cmd[i]!='\n'; i++) {
          speed = speed*10 + cmd[i]-'0';
        }
      }
    //Serial.println(speed);
  }

  // Parse incoming steer data
  if(cmd[0]=='s' && cmd[1]=='t' && cmd[2]=='e' && cmd[3]=='e' && cmd[4]=='r' && cmd[5]==' ') {
    steer = 0;
    if (cmd[6]=='-')
      {
        for(int i=7; cmd[i]!='\n'; i++) {
          steer = steer*10 + cmd[i]-'0';
        }
        steer = -steer;
      }
    else
      {
        for(int i=6; cmd[i]!='\n'; i++) {
          steer = steer*10 + cmd[i]-'0';
        }
      }
  }

  // do command
  turn(speed, steer);
}
```

za finale dovrà essere di 2 o 3 millimetri dal suolo, a tale scopo serviranno due distanziatori da 15mm in metallo o plastica secondo disponibilità.

Il sensore ad ultrasuoni può essere saldato direttamente sullo stampato garantendo un profilo finale molto basso, oppure potete utilizzare un connettore strip femmina, così da garantire il recupero del componente all'occorrenza.

Il robot si presta a diversi utilizzi sia didattici che hobbistici, dal classico inseguire linea sino ad esercitazioni di coding, la compatibilità con la scheda Fishino ci ha però permesso di sperimentare nuove applicazioni, proprio grazie alla connettività

a disposizione. Il modulo WiFi interno a Fishino permette di connettersi ad una rete WiFi esistente e scambiare dati con essa,

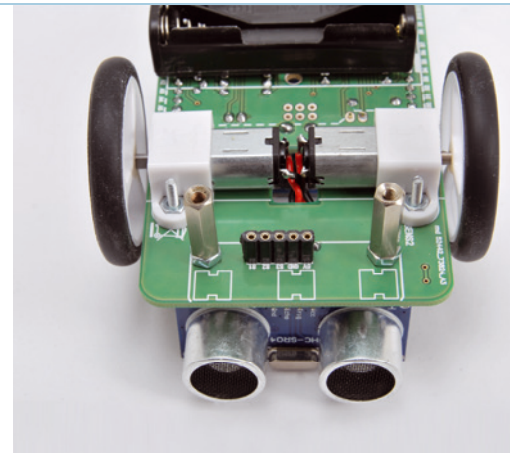
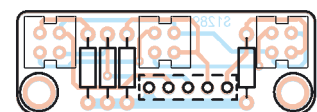
[piano di montaggio **SENSORE IR**]

Elenco Componenti:

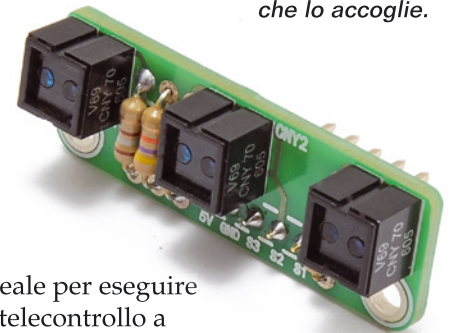
R1 ÷ R3: 22 kohm
R4: 180 ohm
CNY1: CNY70
CNY2: CNY70
CNY3: CNY70

Varie:

- Strip maschio 5 vie 18mm
- Circuito stampato S1289 (39 x 14 mm)



Il circuito stampato del triplo sensore a riflessione va montato dotandolo di pin-strip e fissato tramite colonnine esagonali al lato superiore dello shield, dove trova posto lo strip femmina che lo accoglie.



l'ideale per eseguire un telecomando a distanza di ultima generazione. Abbiamo già apprezzato in passato l'utilizzo dell'app RoboRemo (descritta nell'articolo Smart Home System con Fishino pubblicato nel fascicolo n° 211) per scambiare dati con sistemi connessi in rete WiFi (e non solo) e ci è sembrato interessante riuti-

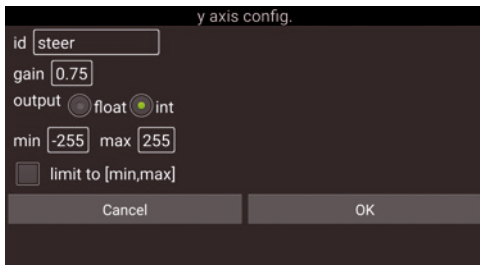


Fig. 2 - Impostazione widget accelerometer.

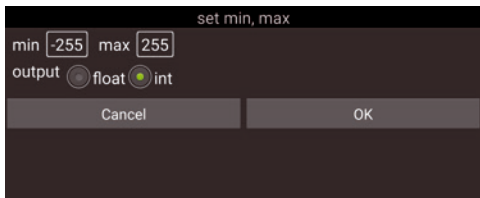


Fig. 3 - Impostazione widget slider.

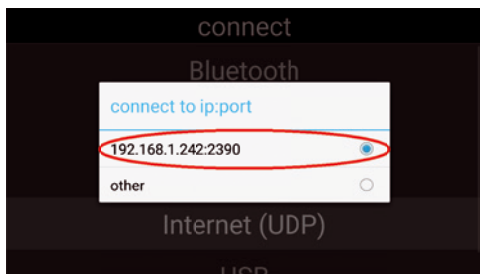


Fig. 4 - Connessione a EasyRobot.

lizzarla anche in questo progetto, vediamo come. L'idea è di controllare a distanza il nostro robot utilizzando il nostro smartphone come telecomando costruendo un'interfaccia personalizzata e personalizzabile proprio con l'app RoboRemo. Tramite uno slider potremmo impostare la velocità di avanzamento del robot, mentre inclinando lo smartphone imposteremo la sterzata, il robot può anche girare su se stesso semplicemente impostando a zero la velocità ed agendo solo sullo sterzo. Per prima cosa programmeremo Fishino affinché possa comunicare tramite pacchetti UDP con dispositivi connessi alla stessa rete. In assenza di una struttura di rete sarà sufficiente programmare Fishino in modalità SoftAP (software enabled access point) al quale il vostro smartphone si

Tabella 3 - Elenco delle funzioni del file *funzioni.ino*.

Funzione	Parametri	Descrizione
void turn(int power, int turn)	power da -255 a +255 turn da -255 a +255	controllo del movimento del robot (avanti e indietro) sterzando nella misura indicata da turn
void forward(int power)	power da 0 a 255	fa avanzare il robot con potenza indicata dal parametro power
void reverse(int power)	power da 0 a 255	fa indietreggiare il robot con potenza indicata dal parametro power
void stop()		ferma il robot
int distance()		visualizza la distanza misurata con il sensore ad ultrasuoni in cm
float VBat()		Visualizza la tensione della batteria in volt

Tabella 4 - Impostazioni parametri interfaccia.

Widget	Parametro	Valore
Slider	set label	SPEED
	set id	speed
	min, max	-255 e +255
	send when moved	on
	set autoreturn	on
Accelerometer	set autoreturn value	mid
	configurare asse Y	smartphone orizzontale
	set id	steer
	gain	0.75
	output	int
	min, 255	-255 e +255
Text Filed	set repat periodo	200msec
	set id	VBAT
	set text	Vbat

connetterà indipendentemente dall'esistenza oppure no di una rete WiFi. La versione free di RoboRemo non permette di utilizzare molti widget, ma a noi basta il control-

lo della velocità e della direzione; quest'ultimo sarà implementato utilizzando l'accelerometro interno dello smartphone, cosicché sarà sufficiente tenere tra le mani lo smartphone (in posizione oriz-

Listato 2

```
// Verifica dello stato della connessione ed invio del valore di carica della batteria
if (millis() > lasttime+1000)
{
    //Verifica ogni secondo se manca segnale
    if (isConnect==0)
        turn(0,0);
    isConnect=0;

    //send battery level
    char tmp[4];
    int BatLevel=100*(VBat()-VbatMIN)/(VbatMAX-VbatMIN);
    BatLevel = constrain(BatLevel, 0, 100);
    sprintf(tmp,"%d",BatLevel);
    String VbatString(tmp);
    sendData= "VBAT " + VbatString + "%\n";

    Udp.beginPacket(Udp.remoteIP(), localPort);
    Udp.print(sendData);
    Udp.endPacket();
    lasttime=millis();
}
```

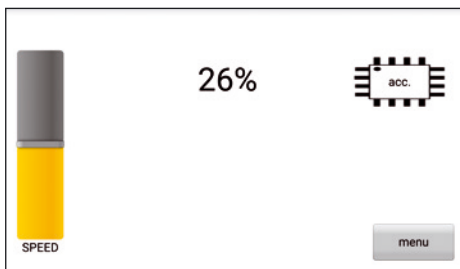


Fig. 5 - Interfaccia di comando conclusa.

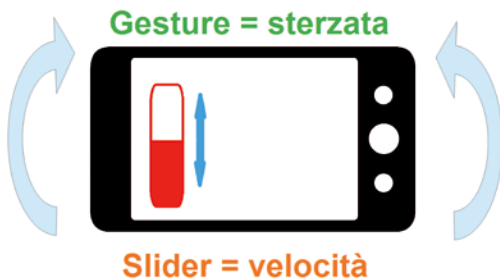


Fig. 6 - Controlli del robot.



Fig. 7 - Easyrobot in funzione in una piccola pista approntata per testarne le funzionalità.

zontale) ed inclinandolo comandare la sterzata del robot. Dallo smartphone saranno inviate essenzialmente due stringhe contenenti, rispettivamente, il valore della velocità (sia in avanti che retromarcia) che quello corrispondente alla sterzata richiesta; dal lato Fishino sarà sufficiente interpretare le due stringhe ricevute e comandare di conseguenza il robot. Non riportiamo la parte relativa alla connessione alla rete WiFi, perché già ampiamente descritta in articoli passati; riportiamo invece, nel **Listato 1**, la porzione di codice che si occupa di estrapolare i dati dalle stringhe inviate da RoboRemo. I valori di velocità e sterzata sono poi utilizzati dalla funzione *turn* per comandare correttamente i motori affinché eseguano il comando richiesto. Pur non avendo previsto una specifica libreria, abbiamo realizzato un file chiamato *function.ino*, che contiene tutte le funzioni necessarie al funzionamento del robot, in modo che il listato principale rimanga pulito e snello e non vi dobbiate preoccupare di

scrivere complicate istruzioni. In questo file sono contenute le istruzioni a basso livello necessarie ad attivare tutte le funzionalità del robot.

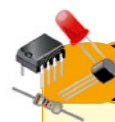
Per l'utilizzo dei sensori di linea non abbiamo previsto alcuna funzione in quanto l'istruzione *analogRead()* è sufficiente per leggere correttamente i valori, eventualmente eseguite alcune misure su diverse superfici per tarare adeguatamente la loro lettura. Una seconda parte del programma si occupa, ad intervalli di un secondo, di inviare allo smartphone lo stato della batteria e di controllare lo stato della connessione; a riguardo considerate che in mancanza di essa per più di un secondo, il robot viene fermato (vedere il **Listato 2**). Prima di caricare il firmware su Fishino dovrete configurare i parametri di accesso alla vostra rete modificando le righe `#define MY_SSID` e `#define MY_PASS` in:

```
#define MY_SSID "MY_SSID"
#define MY_PASS "MY_PASS"
```

Se volete realizzare delle sfide

con i vostri amici o semplicemente volete usare più robot contemporaneamente, è necessario che ciascuno disponga di un indirizzo IP differente; attenti a non andare in conflitto con altri dispositivi connessi alla rete. La riga per definire l'IP address è del tipo:

```
#define IPADDR 192, 168, 1, 242
```



per il MATERIALE

Tutto il materiale utilizzato in questo progetto è disponibile presso Futura Elettronica. Lo shield Easyrobot per Arduino-Fishino (cod. EASYROBOT) è in vendita a Euro 74,00, il sensore IR per robotica (cod. FT1289K) è disponibile a Euro 7,50. Il misuratore di distanza ad ultrasuoni (cod. MI-SDIST04) costa Euro 6,00 e la board Fishino UNO (cod. FISHINOUNO) è disponibile a Euro 36,00. Tutti i prezzi si intendono IVA compresa.

Il materiale va richiesto a:
Futura Elettronica, Via Adige 11,
21013 Gallarate (VA) - Tel: 0331-799775
<http://www.futurashop.it>



Vuoi vedere
il nostro
EasyRobot in
azione? Nulla
di più semplice:
collegati alla
pagina

<https://youtu.be/lxJm-Z5cSeY>.
Dal nostro canale YouTube
www.youtube.com/user/ElettronicaIN
potrai anche vedere i video
di moltissimi altri nostri progetti
proposti in passato.



Altri due parametri possono essere impostati, relativamente ai livelli minimo e massimo della tensione ai capi della batteria, questi saranno usati per ricavare il livello di carica della batteria in valore percentuale; si tratta di:

```
const float VbatMIN = 4.0;  
//tensione a batteria scarica pari a 0%  
const float VbatMAX = 6.5;  
//tensione a batteria carica pari a 100%
```

Il passo successivo consiste nell'installare sul vostro smartphone l'applicazione RoboRemo disponibile gratuitamente nel Play Store. Una volta avviata l'app entrate nella modalità *edit ui* ed inserite uno *slider*, un *accelerometer* ed un *text field*, dopodiché settate i loro parametri come indicato nella **Tabella 4** e mostrato nelle **Fig. 2**, **Fig. 3** e **Fig. 4**. Impostate quindi la proprietà *autorotate* su *lock* per evitare che l'interfaccia ruoti quando ruotate lo smartphone per far sterzare il robot.

Per far funzionare il tutto è necessario accendere il robot ed attendere che si connetta alla

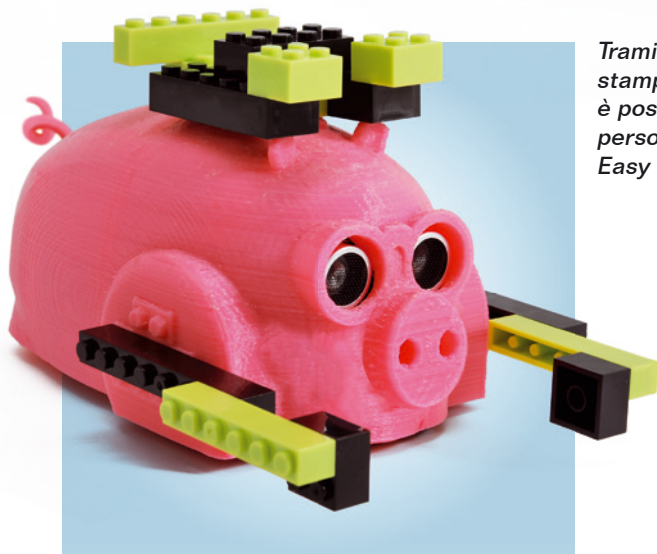
rete: di solito sono sufficienti circa 5 secondi, ma per sicurezza potete lasciare connessa la scheda Fishino al Personal Computer ed attendere i messaggi di debug per verificare se le operazioni vanno a buon fine.

Una volta programmata l'app RoboRemo potete connettervi al robot, con la funzionalità *Connetti*, specificando l'indirizzo IP e la porta indicati nel firmware. L'interfaccia di comando è illustrata nella **Fig. 5** e comprende i controlli visibili nella **Fig. 6**.

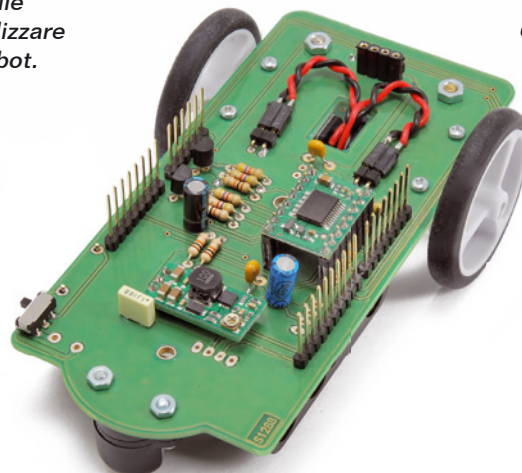
CONCLUSIONI

Con la componentistica proposta in questo articolo, il robot ha un movimento molto fluido e preciso, adattissimo ad esempio per un uso didattico; inoltre l'assorbimento di corrente e il consumo di potenza sono assai modesti, il che permette qualche ora di funzionamento anche utilizzando delle batterie stilo. Nulla vieta di sostituire i motori con altri dello stesso formato, ma con diverso numero di giri, per ottenere delle prestazioni più spinte, fermo restando che l'autonomia di

esercizio si ridurrà e che all'occorrenza occorrerà impiegare una batteria con maggiore corrente di spunto. Per darvi un'idea delle capacità del nostro Easy Robot, potete guardare la **Fig. 7**, che vi mostra il nostro piccolo robot in azione su un tracciato che l'autore ha realizzato appositamente per testarlo. ■



Tramite una stampante 3D è possibile personalizzare Easy Robot.



Il sensore che rileva la distanza è dotato di un pin-strip a quattro contatti (+, - dell'alimentazione, oltre a trigger ed echo) che permette di innestarlo nello strip-femmina previsto sul lato superiore dello shield.

