

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

RoboYún, robot esploratore con webcam e controllo Wi-Fi

Posted By *Adriano Gandolfo* On 16 settembre 2015 @ 1:00 In [Arduino, Robotica & Droni](#) | [14 Comments](#)

Sul sito *Elettronica Open Source*, ho già presentato alcuni modelli di robot comandati in remoto, come [LittleBot](#)^[1] che è comandato tramite Bluetooth, o un [altro modello](#)^[2] che era comandato tramite un telecomando ad infrarossi

L'opportunità data dal contest [20 Arduino Motor Shield di Infineon gratis per voi](#)^[3], ha portato all'idea di realizzare un nuovo modello di rover comandato in remoto utilizzando il sistema Wi-Fi. Per il controllo si è utilizzata una scheda Arduino Yún che è la combinazione di un classico Arduino Leonardo (basato sul processore ATMEGA32U4) con un sistema one-chip Wi-Fi in cui viene eseguito Linino (un MIPS GNU / Linux basata su OpenWRT). Altre particolarità sono date dal fatto che sulla scheda sono presenti alcuni connettori tra i quali una porta USB Host a cui è possibile collegare una web camera, che gestita dalla sezione Linux, permette di trasmettere in streaming le immagini riprese attraverso la rete Wi-Fi.

Descrizione dei componenti base che formano il rover.

Cominciamo con la descrizione dei principali elementi che formano il Rover:

Base robotica con telaio in alluminio e 2 motoriduttori;

Scheda con controllo Arduino Yún;

Motor shield di Infineon (ricevuto tramite il contest [20 Arduino motor shield di Infineon gratis per voi](#)^[3])

Dispositivo Pan & Tilt per movimentazione webcam;

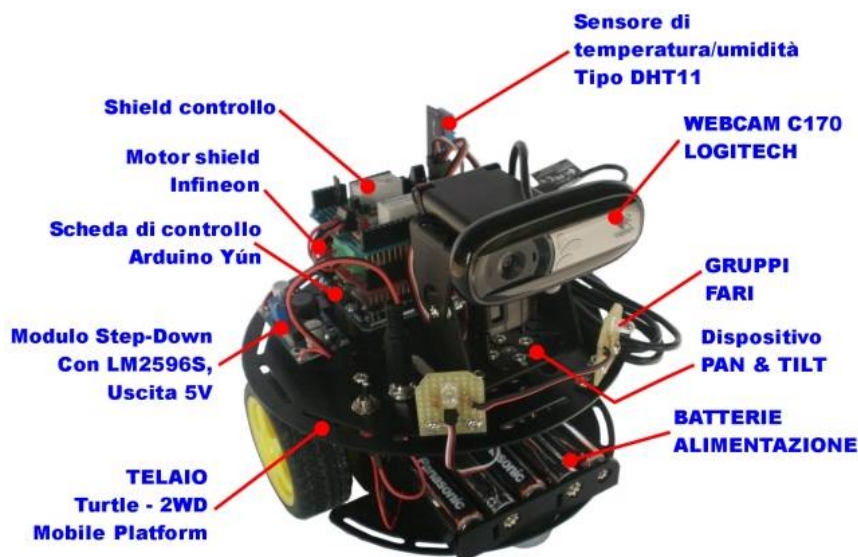
Shield con circuito gestione motori, dispositivo Pan & Tilt, sensori;

Scheda alimentazione;

Webcam.

Queste sono solo alcune delle caratteristiche del robot RoboYún:

Telaio	Il telaio in alluminio di tipo commerciale prodotto dalla DFRobot: Turtle - 2WD Mobile Platform ^[4]
Scheda di comando	Scheda Arduino Yún ^[5] , consigliato l'uso di una memoria SD esterna per ospitare i propri files
Modulo comando motori	DC Motor Control Shield with BTN8982TA for Arduino ^[6]
Shield controllo	Comando relè per inversione senso di rotazione motore Comando fari a led per illuminazione frontale Uscite per azionamento servo Pan & Tilt
Sensore	Temperatura/umidità – Sensore DHT11
Webcam	Webcam modello C170 della Logitech ^[7] - Altezza 60,5 mm, Larghezza 70,3 mm, Profondità 71 mm, Peso 75 g
Dispositivo Pan & Tilt	Dispositivo Pan & Tilt SparkFun codice ROB-10335 ^[8]
Alimentazione	5 batterie tipo AA per una tensione di 7,5V Modulo di tipo Step-Down basato sul circuito integrato LM2596S , per l'uscita di 5V alimentazione parte elettronica.
Motoriduttori	In plastica con rapporto riduzione 1:120, alimentazione 7,5V, assorbimento 170 mA
Ruote	Ruote di plastica e gomma, Diametro esterno: 65mm, Larghezza ruota: 26 millimetri, Hub: 6mm
Misure robot	Diametro 170 mm altezza massima 170 mm
Peso robot	750 g, compreso batterie



[9]

Posizione delle varie parti che costituiscono il robot

Passiamo ora a una breve descrizione dei singoli componenti:

Base robotica

Il telaio utilizzato per la realizzazione del Rover è prodotto dalla [DFRobot](#)s ^[10] e il suo nome commerciale è **Turtle - 2WD Mobile Platform** ^[11], si tratta di un kit costituito da un corpo in alluminio, due motoriduttori, due ruote Tamiya racing (diametro 65 mm), un ballcaster in metallo, un portabatteria, un deviatore, una presa di alimentazione, le viti e i dadi.



[12]

Il montaggio è molto semplice, è sufficiente seguire le indicazioni riportate sul manuale cartaceo ([o in versione pdf](#) ^[13]) presente all'interno della confezione.

Motoriduttori

All'interno del kit della base robotica sono presenti due motoriduttori (SKU: FIT0016 ^[14]) con motore a corrente continua con un'uscita di 160 rpm se alimentati a 6V.



[15]

Caratteristiche Elettriche:

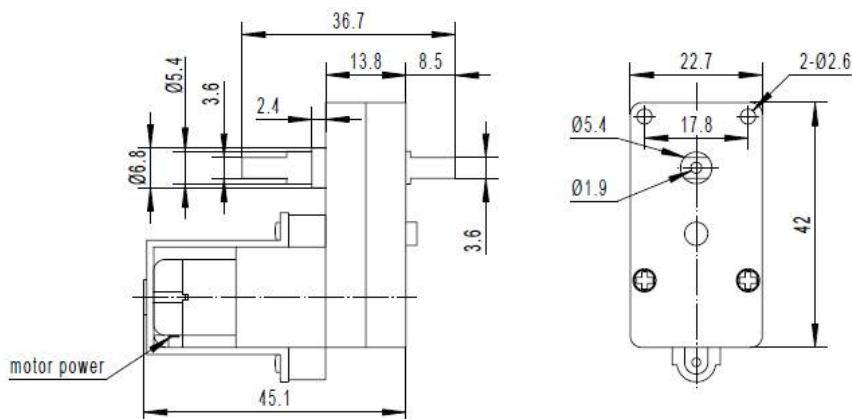
Tensione di funzionamento: 3 ~ 7,5V (6V nominale);
 Max. A vuoto Corrente (3V): 140 mA (3V) - 170 mA (6V);
 Velocità a vuoto (3V): 90 rpm (3V) - 160 rpm (6V);
 Max. Coppia di uscita: 0,8 kgf.cm;
 Max. corrente di Stallo: 2,8 A.

Condizioni Operative:

Carico stimato: 0,2 kgf.cm;
 Temperatura di funzionamento: -10 ~ + 60 °C;
 Tipo motore: 130.

Caratteristiche meccaniche:

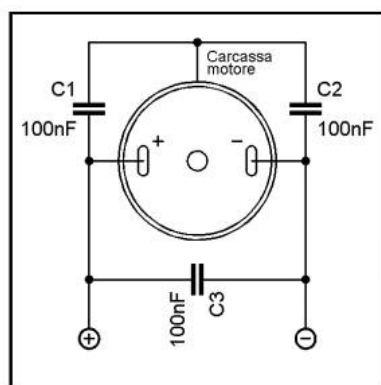
Modalità uscita: 2 lati;
 Rapporto di trasmissione: 1:120;
 Max. Diametro Asse di uscita: 5,4 millimetri;
 Peso netto: 45 g



[16]

Dimensioni del motoriduttore utilizzato

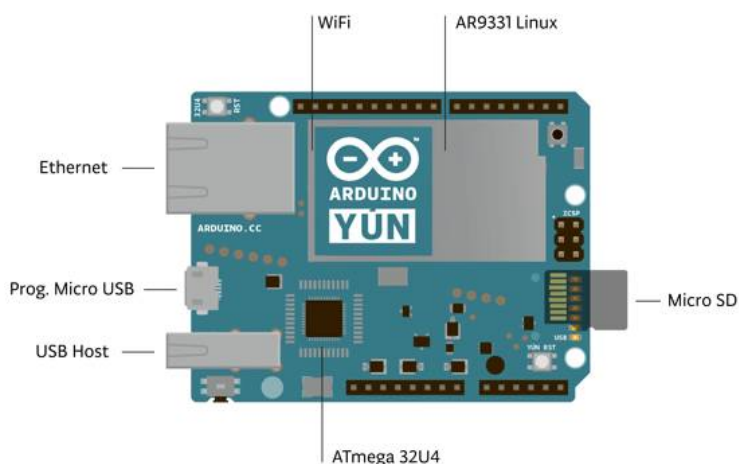
NOTA: Per ridurre le interferenze con la circuiteria elettronica causate dallo scintillio dovuto allo sfregamento delle spazzole del motore elettrico sul collettore, è consigliabile saldare tre condensatori ceramici (o meglio multistrato) da **100nF** uno tra positivo e carcassa, uno tra negativo e carcassa e uno tra positivo e negativo. Nella Figura è visibile lo schema di collegamento dei condensatori utilizzati come filtri antidisturbo.



[17]

Scheda di controllo Arduino Yún

La scheda di controllo utilizzata è il modello **Arduino Yún** [5], in cui sono presenti due microcontrollori: il primo è un **Atheros AR9331** [18] che supporta una distribuzione Linux basata su **OpenWRT** [19] chiamata **Linino** [20], il secondo è un **ATmega32u4** [21], lo stesso che utilizza la scheda **Arduino Leonardo** [22] o la scheda **Arduino Esplora** [23]. La scheda offre così un potente computer connesso con il microcontrollore di Arduino, è dotata di connettore per rete Ethernet, Wi-Fi, slot MicroSD e di 20 pin di I/O, dei quali 7 PWM e 12 ingressi analogici, le sue dimensioni sono 73x53mm per un peso di 32g.



[24]

Caratteristiche del microcontroller AVR Arduino

Microcontroller: ATMEGA32U4
 Tensione di funzionamento: 5V
 Tensione in ingresso: 5V
 Pin di I/O Digitale: 20
 Canali PWM: 7

Canali di ingresso analogici	12
Corrente per Pin I/O:	40 mA
Corrente sul Pin 3.3V:	50 mA
Flash Memory:	32 KB (di cui 4 KB utilizzato da bootloader)
SRAM:	2,5 KB
EEPROM:	1 KB
Velocità di clock:	16 MHz

Caratteristiche microprocessore Linux

Processore:	Atheros AR9331
Architettura:	MIPS @ 400MHz
Tensione di funzionamento:	3.3V
Ethernet:	IEEE 802.3 10 / 100Mbit / s
WiFi:	IEEE 802.11b / g / n
USB tipo A:	Host USB 2.0
Lettore di schede:	Micro-SD
RAM:	64 MB DDR2
Memoria flash:	16 MB

Alimentazione

L'alimentazione della scheda avviene attraverso la connessione micro-USB con una tensione di 5VDC. Se si vuole utilizzare il pin Vin, è necessario fornire una tensione di 5V DC stabilizzata, in quanto sulla scheda non è presente come nelle normali schede Arduino, l'apposito integrato regolatore. Fornire una tensione superiore può danneggiarla.

Memoria

Il processore ATMEGA32U4 ha 32 KB (con 4 KB per il bootloader). Ha anche 2,5 KB di SRAM e 1 KB di EEPROM (che può essere letta e scritta con la libreria EEPROM).

La memoria dell' AR9331 non è incorporata all'interno del processore. La RAM e la memoria di archiviazione sono collegate esternamente. Yún ha 64 MB di DDR2 RAM e 16 MB di memoria flash.

Pin di Ingresso e uscita

Si dispone in totale di 20 pin di input/output digitali, di cui 7 da utilizzare come output PWM e 12 come input analogici.

Ciascuno di questi pin può essere pilotato in maniera indipendente dal software.

Alcuni di questi pin hanno comportamenti specifici:

I pin 0 (Rx) e 1 (Tx) possono essere utilizzati per la comunicazione seriale.

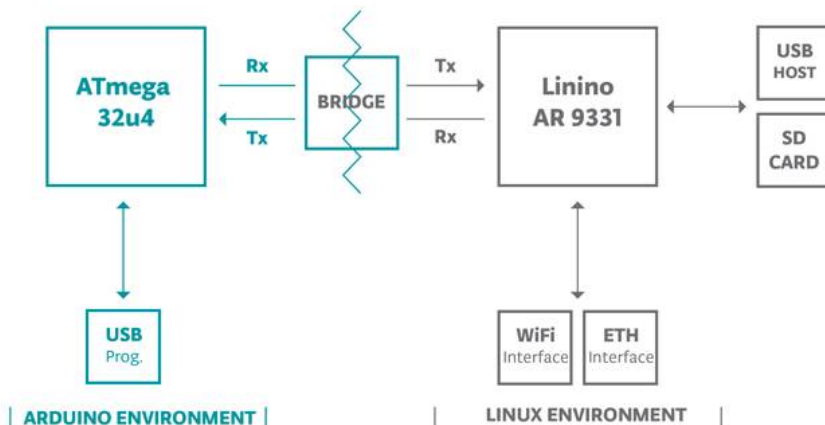
I pin 0, 1, 2, 3 e 7 possono essere configurati come trigger per eventi esterni

I pin 3, 5, 6, 9, 10, 11 e 13 possono essere configurati per generare segnali PWM con risoluzione di 8 bit

I pin A0-A5 e A6-A11 (sui pin digitali 4, 6, 8, 9, 10 e 12) sono gli ingressi analogici. Ognuno di questi pin può essere utilizzato come I/O digitale. Ogni ingresso analogico fornisce 10 bit di risoluzione (cioè 1024 valori differenti). Di default, gli ingressi analogici hanno un range che va da massa a 5V, anche se è possibile cambiare l'estremità superiore con il pin AREF.

Bridge

Come abbiamo visto Arduino Yún integra il microcontrollore **ATmega32u4** e l'**Atheros AR9331** in cui "gira" **Linino** [20], una distribuzione di Linux. Il "Bridge" è **una libreria** [25] che mette in comunicazione questi due componenti.



[26]

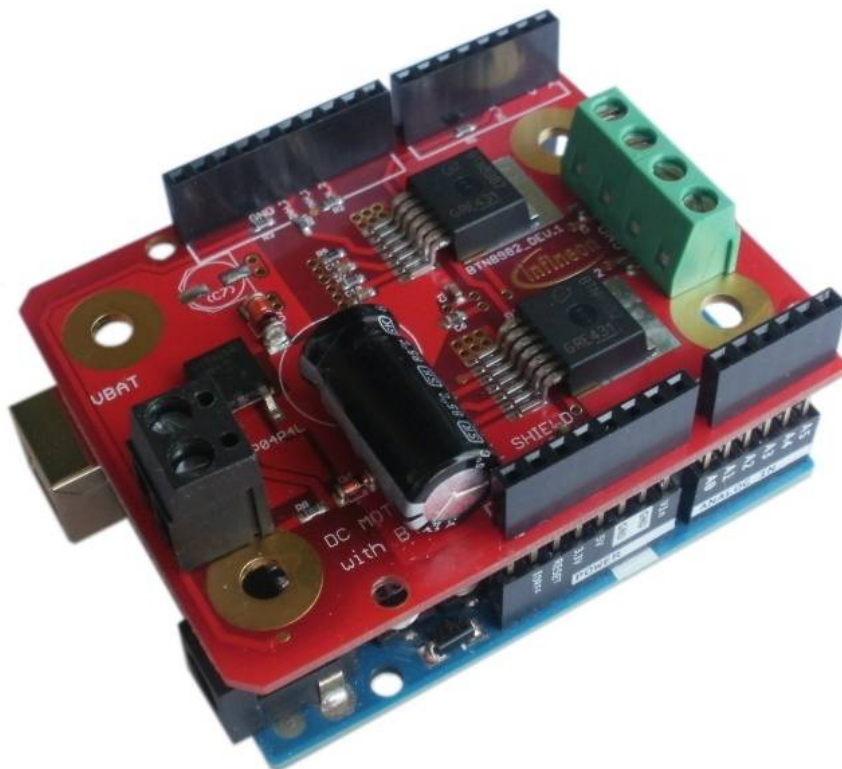
Dalla figura si nota come la microSD, l'interfaccia Wi-Fi ed Ethernet non siano direttamente connesse al microcontrollore ATmega; quest'ultimo ha accesso tramite il microprocessore AR 9331. Il Bridge consente quindi ad Arduino di interfacciarsi con queste periferiche.

Motor shield della Infineon

Il controllo dei motori è realizzato tramite il **motor Shield della Infineon** [27] che permette di controllare, tramite alcune porte I/O di Arduino,

un motore DC nei due sensi di marcia, oppure due motori DC nello stesso di marcia.

Lo schema della scheda è molto semplice, essendo formato da un circuito per la protezione contro l'inversione di polarità e da due circuiti ponte circondati da pochissimi altri componenti passivi.

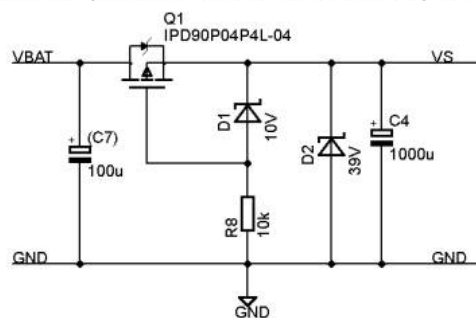


[28]

Circuito per la protezione contro l'inversione di polarità

Il circuito di protezione contro l'inversione della polarità è formato dal mosfet **Q1** tipo **IPD90P04P4L** [29], prodotto dalla stessa **Infineon**, questi è un mosfet di potenza canale P, la cui polarizzazione è realizzata dalla resistenza **R8** e dallo zener **D1**, mentre la tensione di alimentazione è limitata nel suo valore massimo tramite lo zener **D2** da 39 V, il condensatore elettrolitico **C4** da 1000 μ F è necessario per mantenere basso il ripple della tensione **VS** durante la commutazione.

Circuito di protezione contro l'inversione di polarità

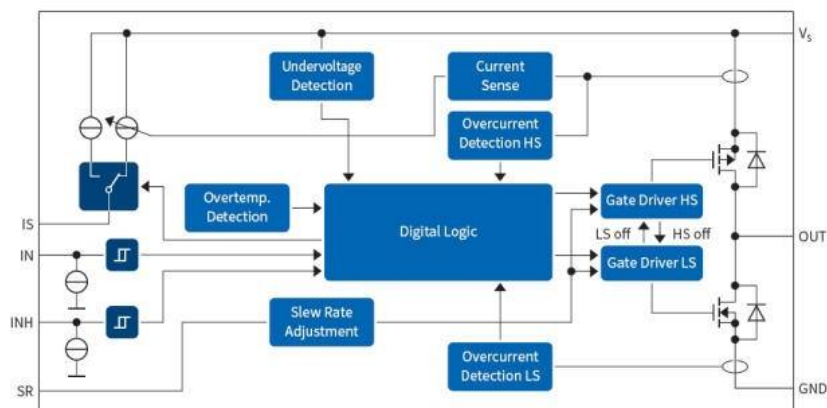


[30]

Circuito di potenza

Il cuore della scheda è rappresentato da due integrati tipo **BT8982TA** [31]: si tratta di un integrato semiponte H, il circuito interno è formato da un MOSFET canale P e uno a canale N più il relativo circuito driver.

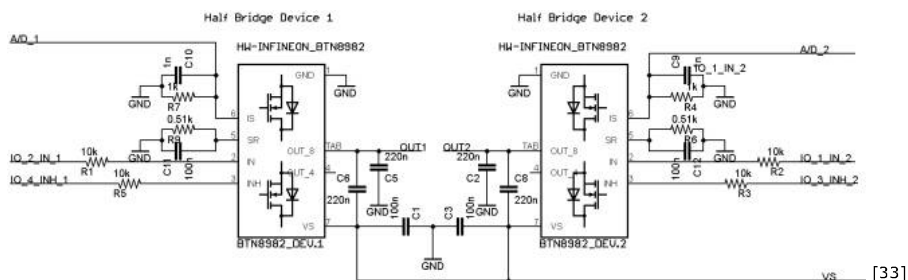
Il circuito è facile da controllare, poiché è sufficiente applicare segnali di livello logico al pin **IN** e **INH**.



[32]

Quando si applica un segnale **PWM** (acronimo inglese di **Pulse-Width Modulation** o modulazione di larghezza di impulso) al pin IN, la corrente fornita al motore, può essere controllata tramite il duty cycle del PWM.

Con un resistore collegato esternamente tra il pin SR e GND è possibile regolare la velocità di risposta, o slew rate.



Le principali caratteristiche dell'integrato sono:

Tensione d'ingresso nominale 8-18 V (max. 6 - 40 V);

Possibilità di controllo motori Brushed DC fino a 250 W con carico continuo

Unità per il controllo sia di un motore DC bidirezionale con spazzole o di due motori a corrente continua unidirezionali;

Corrente media pari a 30 A (BTN8982 ha una limitazione in corrente @55 A), il limite è dato dalla dissipazione del PCB;

Alta frequenza PWM, per esempio 30 kHz;

Slew Rate modificabile per EMI ottimizzato tramite resistori esterni;

Circuito del driver con ingressi a livello logico;

Protezione da sovraccarichi o temperature eccessive;

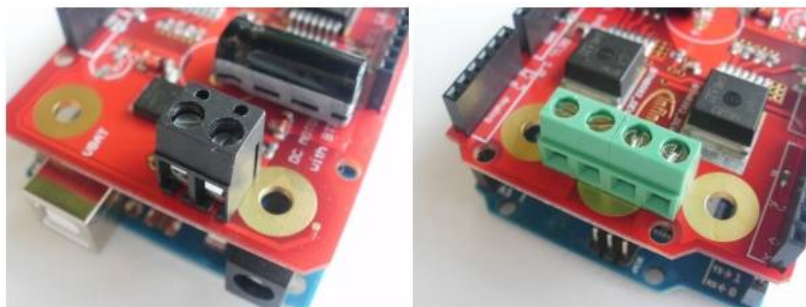
Diagnosi con current sense, protezione da sovratemperatura e sovracorrente;

Lo shield presenta già i collegamenti con i pin di controllo,

Modifiche effettuate sullo shield originale.

Allo shield originale sono state apportate alcune modifiche per adattarlo al progetto.

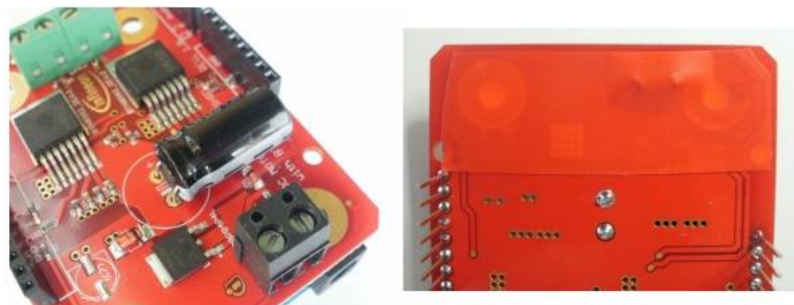
Inserimento morsettiere a vite: la scheda in origine prevede l'utilizzo di boccole per spinotto a banana con diametro 4 mm, si è invece preferito inserire le più comode morsettiere a vite. Per questo motivo si è dovuto forare il circuito ed effettuare qualche saldatura.



[34]

Modifica orientamento condensatore: il condensatore di livellamento C4 che era montato in verticale, per ridurre l'ingombro è stato montato orizzontalmente.

Isolamento pin alimentazione: esiste la possibilità che il pad dell'alimentazione vada a contatto con la presa usb (nel caso della scheda Arduino UNO) o con la presa ethernet (nel caso della scheda Arduino Yún) per questo motivo la scheda è stata isolata con del nastro adesivo.



[35]

Sensore di temperatura e umidità

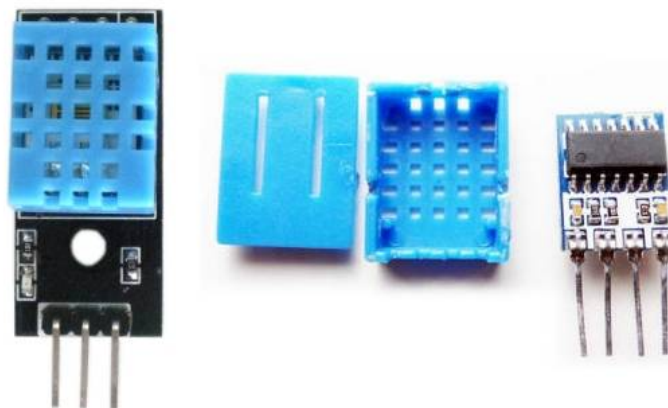
Il sensore di temperatura e umidità è il modello **DHT11** che è un sensore con uscita dati in formato digitale. I suoi elementi sensibili sono connessi ad un processore 8-bit single-chip.

Ogni sensore è compensato in temperatura e calibrato in un'apposita camera di calibrazione che determina in modo preciso il coefficiente che viene poi salvato all'interno della memoria OTP.

Modello	DHT11
Alimentazione	3-5,5V DC
Segnale di uscita	digitale tramite single-bus

Elemento sensibile	resistenza in Polimero
Campo di misura umidità	20-90% di umidità relativa, temperatura di 0-50 gradi Celsius
Precisione	umidità $\pm 4\%$ RH (Max $\pm 5\%$ di umidità relativa), temperatura $\pm 2,0$ Celsius
Risoluzione o la sensibilità	umidità 1% di umidità relativa, temperatura 0.1 Celsius
Ripetibilità umidità	$\pm 1\%$ di umidità relativa temperatura ± 1 Celsius
Umidità isteresi	$\pm 1\%$ RH
Stabilità a lungo termine	$\pm 0,5\%$ UR / anno
Tempo di rilevazione medio:	2s
Dimensioni	12x 15,5 x5,5 millimetri

Il modello utilizzato venduto da [Banggood](#) [36] è già montato su una basetta, insieme alla resistenza tra il pin di alimentazione e il pin dati, è inoltre presente un led che segnala la presenza dell'alimentazione. Il sensore è montato sullo speciale shield previsto sul robot ed è connesso al pin **D7**; per la sua gestione si utilizza una speciale [libreria DHT.h](#) [37].



[38]

Dispositivo PAN & TILT

Il dispositivo Pan & Tilt è analogo a quello già mostrato nell'articolo [Gestione di un dispositivo Pan & Tilt con la scheda Arduino Esplora](#) [39], questo è venduto dalla [SparkFun codice ROB-10335](#) [8].

Nel kit sono presenti i componenti meccanici cioè due staffe e tutto l'hardware necessario per il fissaggio, i servomotori devono essere presi a parte, quelli consigliati sono dei servo compatibili con il modello Hitec HS-55 codice [Sparkfun ROB-09065](#) [40] le cui caratteristiche sono:

Tensione: 4,8-6,0 V;
 Coppia: 1,6/ 2 Kg/cm. (4,8 / 6.0V);
 Velocità: 0,15 / 0,10 sec / 60 ° (4,8 / 6.0V);
 Rotazione: ~ 160 °;
 Ingranaggi in Nylon;
 3-Pole ferrite motore;
 Dimensioni: 31,8 x 11,7 x 29 mm;
 Filo Lunghezza: 160 mm;
 Peso: 9g.



[41]

I due servo sono collegati allo speciale shield realizzato per il robot e risultano collegati alle porte **D5** (servo Pan) e **D6** (servo Tilt). Per la gestione è utilizzata la [libreria servo.h](#) [42].

WebCam

RoboYún è dotato di una webcam, la scelta del modello, è fatta anche in funzione della possibilità di essere riconosciuta dalla sistema operativo della scheda Arduino Yún, quindi si è scelto il modello [C170 della Logitech](#) [7].



[43]

Questa WebCam è un modello poco costoso dalle buone caratteristiche, che sono:

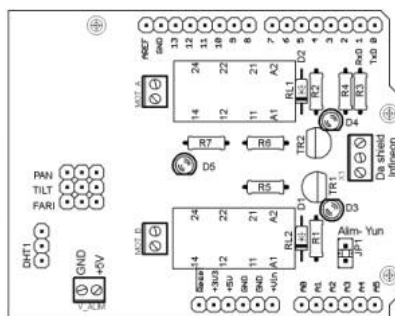
- Acquisizione video: Fino a 1024 x 768 pixel;
- Tecnologia Logitech Fluid Crystal;
- Foto: Fino a 5 megapixel (con interpolazione software);
- Microfono incorporato con funzionalità di riduzione del rumore;
- Certificazione Hi-Speed USB 2.0;
- Clip universale per laptop e monitor LCD o CRT.

Shield con circuito gestione motori, dispositivo Pan & Tilt, sensori

Come abbiamo visto il motor Shield della Infineon, presenta un solo ponte H, per cui può gestire un solo motore DC nei due sensi di marcia, oppure due motori DC nello stesso senso di marcia.

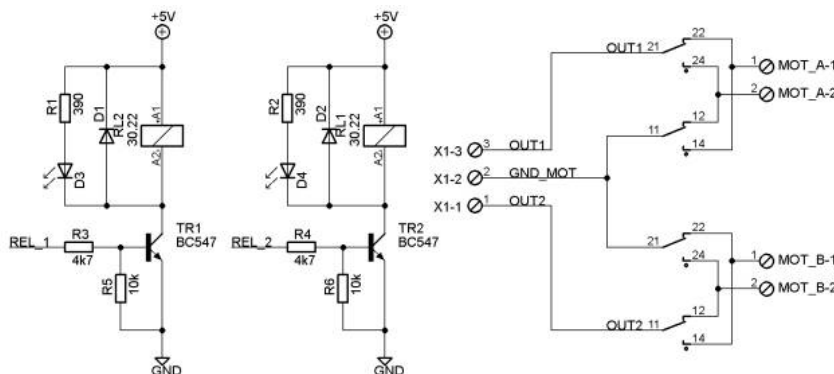
Per questo motivo nell'azionamento dei motori sono utilizzati dei motor shield con integrati tipo **L293** [44], **SN754410** [45], **L298** [46], che hanno al loro interno **4 ponti H**, il che permette di gestire due motori nei due sensi di marcia.

Per aggirare il problema si è realizzato un apposito shield ([Schema formato pdf](#) [47]) su cui sono montati due relè a doppio contatto, che permettono di invertire il senso di rotazione dei motori, indipendentemente dal senso dato dal modulo Infineon, che in questo caso viene utilizzato solo come regolatore della velocità di rotazione tramite la gestione PWM.



[48]

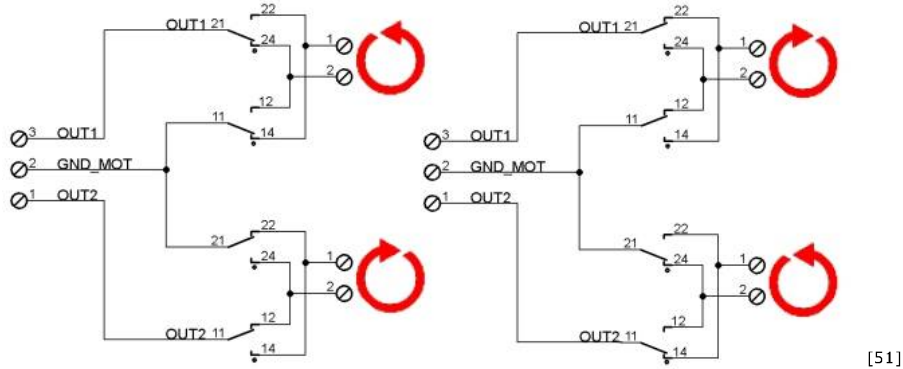
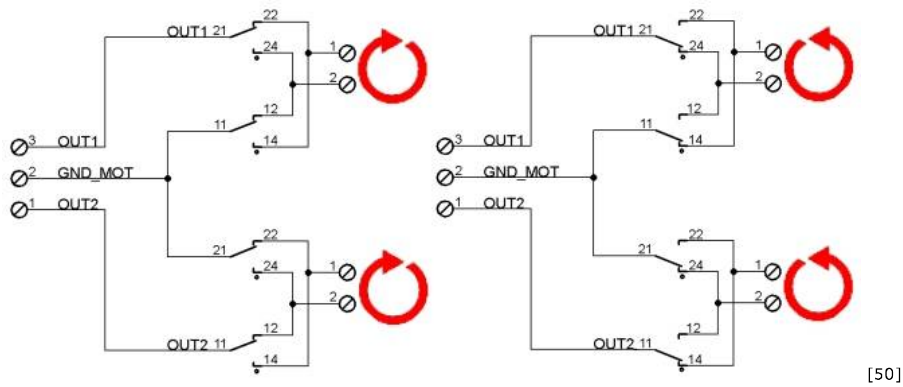
Il segnale proveniente dallo shield Infineon è applicato alla **morsettiera X1**, dove arriva ai due relè a doppio scambio controllati tramite i due transistor **TR1** e **TR2** (tipo **BC547**) pilotati dalle porte **D2** e **D4**.



[49]

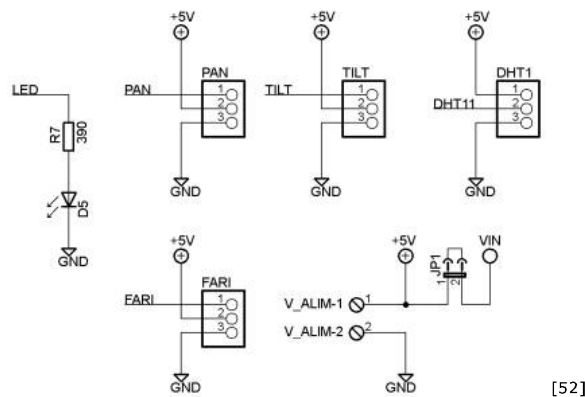
Dettaglio della sezione gestione dei relè controllo rotazione motori.

L'attivazione dei relè è segnalato anche tramite i due Led **D3** e **D4**, nel circuito sono poi presenti i diodi **D1** e **D2** che hanno il compito di scaricare rapidamente tutta l'energia accumulata nella bobina mantenendo la tensione inversa a valori accettabili e non dannosi per gli altri componenti.



Schema di attivazione relè e rotazione motori

Sulla scheda sono inoltre presenti i connettori per il collegamento dei due servo del dispositivo di orientamento della webcam, il connettore per il sensore di temperatura/umidità e uno per il pilotaggio del circuito dei fari.

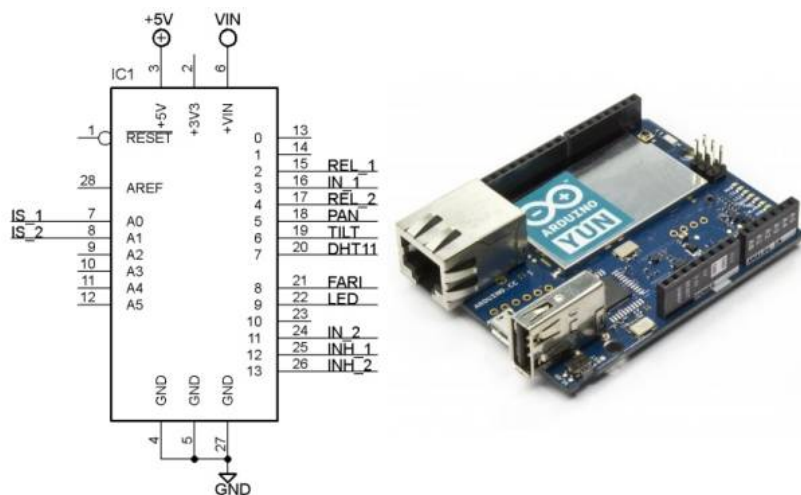


[52]

Sullo shield è poi montato un led di colore blu **D5** controllato dalla porta **D9** che è utilizzato per segnalare quando il robot è pronto ad accettare i comandi.

L'alimentazione +5V proveniente dal modulo regolatore è fornita alla morsettiera **V_ALIM**, dove tramite il ponticello **JP1**, viene trasferita al pin **VIN** per alimentare la scheda Arduino Yún. La funzione dei pin ingresso/uscita è riassunta in questa tabella:

Pin	IN/OUT	Funzione
D2	OUT	Comando Relè 1
D3	OUT	IN_1 (Infineon)
D4	OUT	Comando Relè 2
D5	OUT	Comando Servo TILT
D6	OUT	Comando Servo PAN
D7	IN	Sensore DHT11
D8	OUT	Gestione Faro
D9	OUT	Led blu
D11	OUT	IN_2 (Infineon)
D12	OUT	INH_1 (Infineon)
D13	OUT	INH_2 (Infineon)
A0	IN	IS_1 (Infineon)
A1	IN	IS_2 (Infineon)

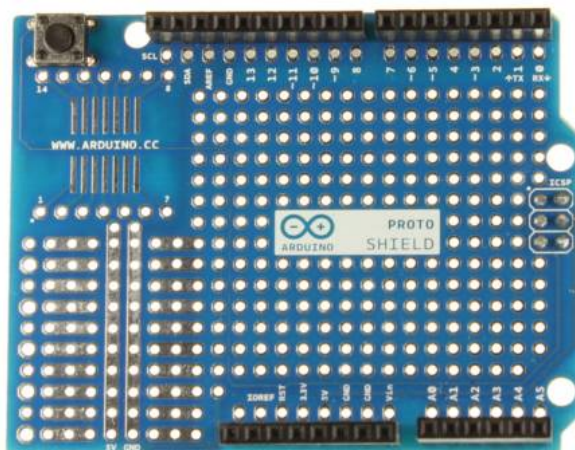


[53]

L'elenco dei componenti per la sua costruzione è:

Sigla	Componente
D1, D2	Diodo 1N4148
D3, D4	Diodo led ø 3mm Rosso
D5	Diodo led ø 3mm Blu
R1,R2, R7	Resistenza 390 Ω ¼ W
R3,R4	Resistenza 4,7 kΩ ¼ W
R5, R6	Resistenza 10 kΩ ¼ W
RL1, RL2	Relè FINDER tipo 30.22.7.005.0010, 5VDC
TR1, TR2	Transistor tipo BC547
X1	Morsettiera 3 poli passo 2,5 mm
MOT_A, MOT_B, V_ALIM	Morsettiera 2 poli passo 2,5 mm
FARI, PAN, TILT	Pinhead 3 poli maschio
DHT1	Pinhead 3 poli femmina
JP1	Pinhead 2 poli mascho con jumper

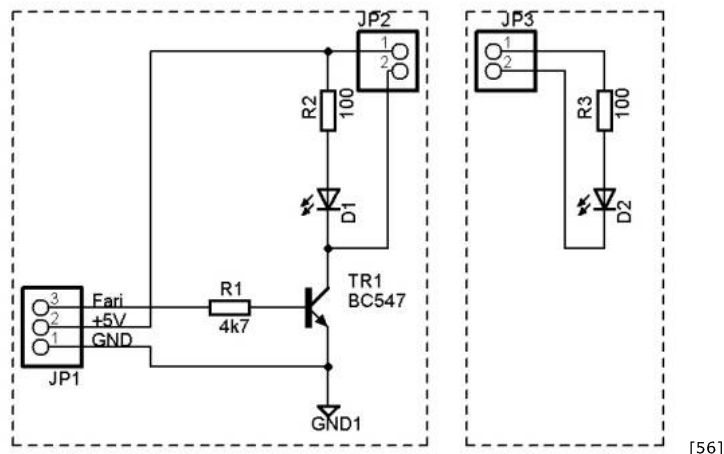
I componenti sono montati su una scheda prototipo tipo **Proto shield R3** [54].



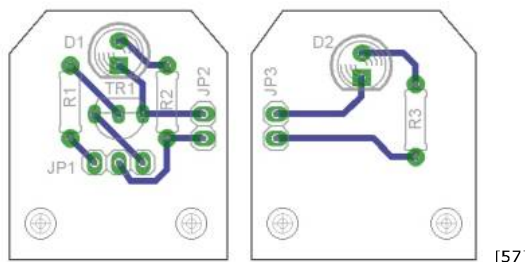
[55]

Circuito fari

Per l'illuminazione della zona sorvegliata dal robot si utilizzano due led bianchi ad alta luminosità, montati su due piccoli circuiti stampati fissati al telaio. Per il pilotaggio dei led si è utilizzato un transistor in modo da non sovraccaricare la porta di Arduino.



[56]



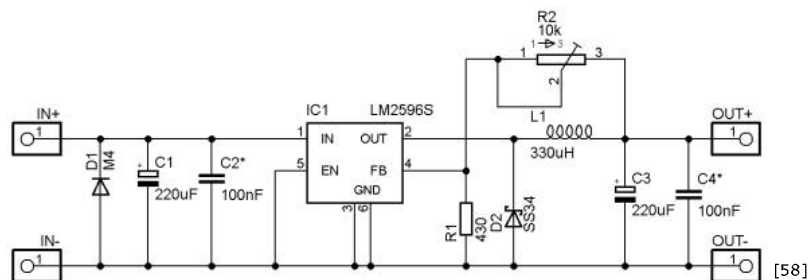
[57]

L'elenco dei componenti:

Sigla	Componente
D1, D2	Diodo led bianco ad alta luminosità ø 5mm
R1	Resistenza 4,7 kΩ ¼ W
R2,R3	Resistenza 100 Ω ¼ W
TR1	Transistor tipo BC547
JP1	Pinhead 3 poli maschio
JP2, JP3	Pinhead 2 poli maschio

Alimentazione del robot

L'alimentazione del robot è fornita dal pacco batteria composto da 5 batterie da 1,5V per un totale di 7,5V.



[58]

Il modulo utilizzato per la regolazione della tensione di alimentazione è basato sul circuito integrato **LM2596S** [59]: questa serie di regolatori monolitici fornisce tutte le funzioni attive, per realizzare un regolatore di commutazione di tipo Step-Down (buck), in grado di erogare sino a 3A con ottima regolazione del carico.

Questi dispositivi sono disponibili sia con tensioni fisse di uscita 3.3V, 5V, 12V oppure nella versione con uscita regolabile.

Per realizzare il circuito è richiesto un numero minimo di componenti esterni, questi regolatori sono semplici da usare avendo un circuito interno di compensazione della frequenza e un oscillatore a frequenza fissa.

La serie LM2596 opera a una frequenza di commutazione di 150 kHz permettendo l'uso di componenti del filtro di dimensioni più piccole di quanto sarebbe necessario con regolatori a commutazione operanti a frequenza inferiore.

Altre caratteristiche includono una tolleranza del $\pm 4\%$ sulla tensione di uscita e $\pm 15\%$ della frequenza dell'oscillatore.

E' possibile la **funzione di Shutdown** esterno, con una corrente di standby di circa 80 μ A.

Sono inoltre presenti funzioni di protezione che includono un limitatore della corrente d'uscita e l'arresto in caso di surriscaldamento o di guasto.



[60]

Nel nostro caso è stato utilizzato un piccolo modulo commerciale dalle ridotte dimensioni (con uscita regolabile tramite un trimmer multigiro) le cui caratteristiche sono:

Tensione di ingresso: da 3 V a 40 V;

Tensione di uscita regolabile: da 1,5 V a 35 V (la tensione d'uscita deve essere almeno 1,5V più elevata dell'ingresso);

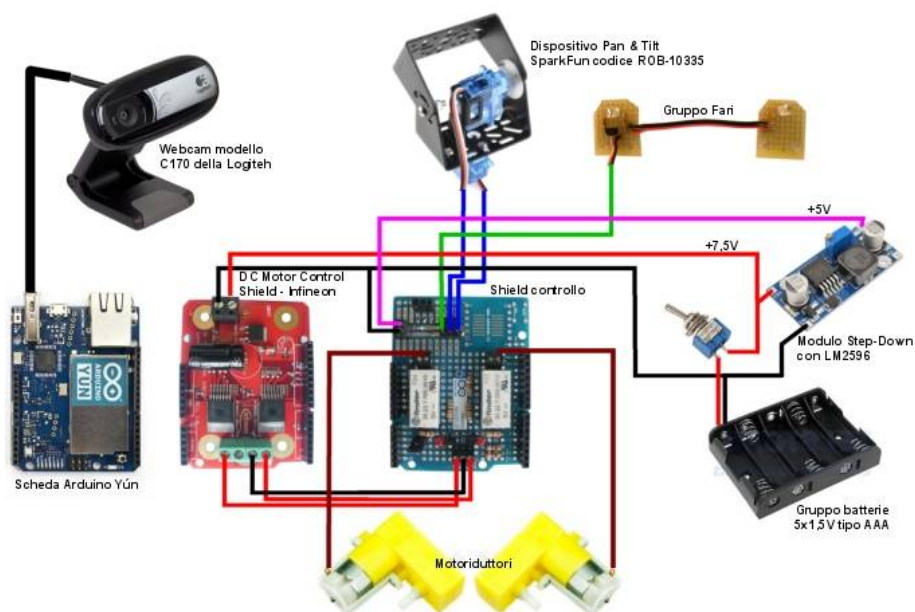
Corrente massima: 3 A;

Dimensioni: 48,35 x 23,35 x 14 mm.

Montaggio del robot

Le operazioni di montaggio sono molto semplici poiché il telaio è già dotato di tutte le forature necessarie.

Le schede Arduino Yún, il motor Shield della Infineon e lo shield di gestione si montano una sull'altra, un po' di manualità sarà necessaria per effettuare i vari cablaggi, come visibile nello schema sotto riportato.



[61]

Programmi di gestione.

Per la gestione del robot tramite la rete Wi-Fi è previsto l'utilizzo di un'interfaccia che potrà essere caricata su un PC o su un Tablet.

L'interfaccia è al momento in via di completamento: è composta da una normale pagina web che viene caricata all'interno della memoria della scheda Arduino Yún.

Questa pagina potrà essere richiamata direttamente tramite l'indirizzo della scheda se questa è collegata senza access point ip di default tramite:

<http://192.168.240.1/sd/robot/index.html>

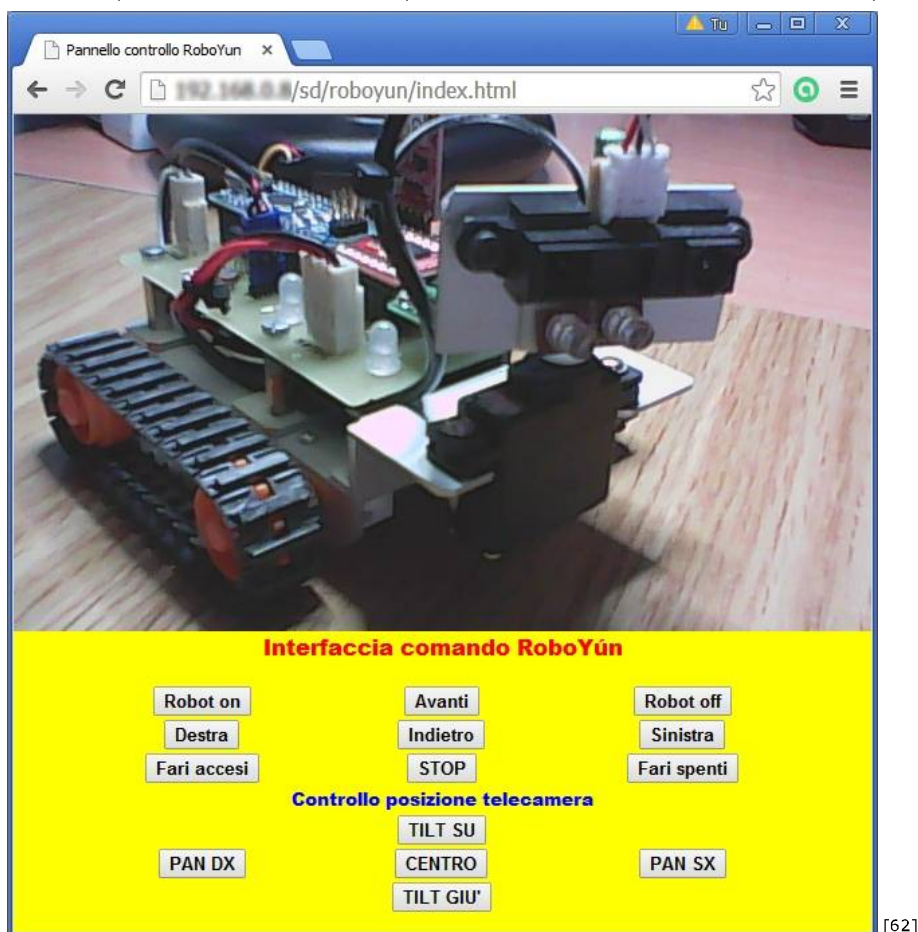
oppure se la scheda è connessa con access point con il comando

<http://arduino.local/sd/robot/index.html>

Il video stream potrà essere visto all'indirizzo:

<http://arduino.local:8080/stream.html>

Il Pannello di controllo è formato da due frame in cui il quello superiore mostrerà le immagini riprese dalla webcam, mentre in quello inferiore è presente l'interfaccia di comando.



[62]

Per la gestione dei comandi è utilizzata una libreria javascript Open Source chiamata **Zepto** [63], questa è rilasciata con licenza MIT ed è largamente compatibile con jQuery e destinata ai browser moderni e in particolare a quelli dei dispositivi mobili e proprio per questo motivo è piccolissima (5-10 kb) anche rispetto a jQuery.

Zepto utilizza le stesse API di jQuery in modo che gli sviluppatori possano utilizzarla senza dover imparare nuove istruzioni, per questo motivo insieme alle pagine HTML sarà necessario inserire anche il file: "zepto.min.js".

Il listato del programma di gestione sarà inserito in un futuro articolo, in questa fase si procederà solamente a testare le varie parti e a collegare e installare i driver per la gestione della webcam. Per quanto riguarda la programmazione di Arduino Yún utilizzeremo il relativo IDE, che in questo momento è disponibile nella **versione 1.6.6** [64].

```

RoboYun_test | Arduino 1.6.6 Hourly Build 2015/07/08 06:44
File Modifica Sketch Strumenti Aiuto

RoboYun_test

#include <Servo.h> //Libreria gestione servomotori
#include "DHT.h" //Libreria gestione sensore
#include <Console.h> //Libreria

#define DHITYPE DHT11 // Definisce il tipo di sensore DHT 11
#define DHTPIN 7 // Definisce il pin a cui è collegato
DHT dht(DHTPIN, DHITYPE); // Inizializza il sensore DHT11

Servo panservo;
Servo tiltservo;
const int rele_1 = 2;
const int rele_2 = 4;
const int fari = 8;
const int led_1 = 9;
const int vel = 120;

// Assegnazione dei pin per il primo half-bridge

Compilazione completata

Lo sketch usa 15.756 byte (54%) dello spazio disponibile per i programmi.
Le variabili globali usano 950 byte (37%) di memoria dinamica, lasciando
libera 1.650 byte (53%) di memoria dinamica per le variabili locali.

16 Arduino Yún on

```

[65]

Una volta installato, basterà selezionare **Arduino Yún** dal menu Strumenti> **Board**.

Per i dettagli, vedere la [pagina reference](#) [66] e [tutorials](#) [67].

L'ATmega32U4 su Arduino Yún è fornito con un bootloader che permette di caricare un nuovo codice senza l'uso di un programmatore hardware esterno. È anche possibile programmarlo direttamente utilizzando il collegamento Wi-Fi.

Programma di test

Per verificare il funzionamento del programma potremmo caricare **RoboYun_test** [68], che permette di testare tutte le varie parti.

Le varie fasi dell'esecuzione del programma possono essere seguite tramite il collegamento WiFi, in quanto i messaggi del programma sono inviati al computer con il comando "[console](#) [69]", questi utilizzando il bridge, che consente di inviare informazioni da Yún ad un computer proprio come si farebbe con il monitor seriale, ma in modalità wireless.

Il programma testerà in sequenza le varie parti del robot:

Accensione Fari

Accensione led blu

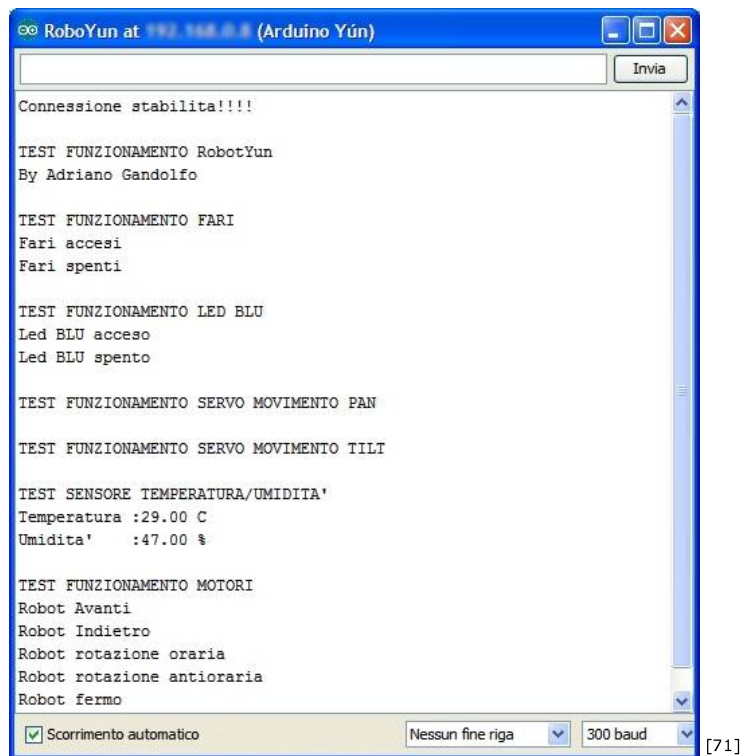
Test servo Pan (effettuare il test senza la webcam)

Test servo Tilt

Test sensore temperatura/umidità

Test motori

Una volta caricato il programma, si dovrà attivare il monitor seriale, che mostrerà le varie fasi di esecuzione del programma.



Listato del programma di test

```

#include <Servo.h> //Libreria gestione servomotori
#include "DHT.h" //Libreria gestione sensore
#include <Console.h> //Libreria

#define DHTTYPE DHT11 // Definisce il tipo di sensore DHT 11
#define DHTPIN 7 // Definisce il pin a cui è collegato
DHT dht(DHTPIN, DHTTYPE); // Inizializza il sensore DHT11

Servo panservo;
Servo tiltservo;
const int rele_1 = 2;
const int rele_2 = 4;
  
```

```
const int fari = 8;
const int led_1 = 9;
const int vel = 120;

// Assegnazione dei pin per il primo half-bridge
const int IS_1 = A0; // current sense and diagnostics
const int IN_1 = 3; // input (PWM)
const int INH_1 = 12; // inhibit (low = sleep)

// Assegnazione dei pin per il secondo half-bridge
const int IS_2 = A1; // current sense and diagnostics
const int IN_2 = 11; // input (PWM)
const int INH_2 = 13; // inhibit (low = sleep)

void setup() {

  // inicializa la comunicazione seriale
  Bridge.begin();
  Console.begin();
  // Gestione servo
  tilt servo.attach(5); // Collega il servo Tilt alla porta D5
  panservo.attach(6); //Collega il servo Pan alla porta D6
  tilt servo.write(90); //tilt servo al centro
  panservo.write(90); // panservo al centro

  // Gestione relè
  pinMode(rele_1, OUTPUT);
  digitalWrite(rele_1, LOW);
  pinMode(rele_2, OUTPUT);
  digitalWrite(rele_2, LOW);

  // Impostazione stato input/output dei pin
  pinMode(IS_1, INPUT);
  pinMode(IN_1, OUTPUT);
  pinMode(INH_1, OUTPUT);
  pinMode(IS_2, INPUT);
  pinMode(IN_2, OUTPUT);
  pinMode(INH_2, OUTPUT);
  pinMode(fari, OUTPUT);
  pinMode(led_1, OUTPUT);

  // Impostazione pin gestione motori
  analogWrite(IN_1, 0); // set motor speed to 0
  digitalWrite(INH_1, HIGH); // enable OUT1
  analogWrite(IN_2, 0); // set motor speed to 0
  digitalWrite(INH_2, HIGH); // enable OUT2
  digitalWrite(led_1, HIGH); // robot operativo

  while (!Console) {
    ; // Attesa per la connessione con la Console.
  }
  Console.println("Connessione stabilita!!!");
  digitalWrite(led_1, HIGH);
}

void loop() {
  //Messaggio iniziale
  Console.println("");
  Console.println("TEST FUNZIONAMENTO RoboYun");
  Console.println("By Adriano Gandolfo");
  Console.println("");

  //Test funzionamento fari
  Console.println("TEST FUNZIONAMENTO FARI");
  Console.println("Fari accesi");
  digitalWrite(fari, HIGH);
```

```
delay (1000);
Console.println("Fari spenti");
digitalWrite(fari, LOW);
delay (1000);
Console.println("");

//Test funzionamento led blu
Console.println("TEST FUNZIONAMENTO LED BLU");
Console.println("Led BLU acceso");
digitalWrite(led_1, HIGH);
delay (1000);
Console.println("Led BLU spento");
digitalWrite(led_1, LOW);
delay (1000);

// Test del servo Pan
Console.println("");
Console.println("TEST FUNZIONAMENTO SERVO MOVIMENTO PAN");
panservo.write(45);
delay(1000);
panservo.write(0);
delay(1000);
panservo.write(90);
delay(1000);
panservo.write(135);
delay(1000);
panservo.write(160);
delay(1000);
panservo.write(90);
delay(1000);

// Test del servo TILT
Console.println("");
Console.println("TEST FUNZIONAMENTO SERVO MOVIMENTO TILT");
tiltservo.write(45);
delay(1000);
tiltservo.write(0);
delay(1000);
tiltservo.write(90);
delay(1000);
tiltservo.write(135);
delay(1000);
tiltservo.write(160);
delay(1000);
tiltservo.write(90);
delay(1000);

// Test sensore
float t = dht.readTemperature();
float h = dht.readHumidity();
Console.println("");
Console.println ("TEST SENSORE TEMPERATURA/UMIDITA'");
Console.print ("Temperatura :");
Console.print (t);
Console.println (" C");
Console.print ("Umidita'   :");
Console.print (h);
Console.println (" %");

// Test funzionamento motori
// Robot AVANTI
Console.println("");
Console.println ("TEST FUNZIONAMENTO MOTORI");
Console.println("Robot Avanti");
avanti(vel);
delay(2000);
```



```
// Robot INDIETRO
Console.println("Robot Indietro");
indietro(vel);
delay(2000);

// Robot a DESTRA
Console.println("Robot rotazione oraria");
destra(vel);
delay(2000);

// Robot a Sinistra
Console.println("Robot rotazione antioraria");
sinistra(vel);
delay(2000);

//Robot fermo
Console.println("Robot fermo");
fermo();
delay(5000);
}

//Routine

void avanti(int motorSpeed) {
  digitalWrite(rele_1, HIGH);
  digitalWrite(rele_2, HIGH);
  delay (100);
  analogWrite(IN_1, motorSpeed);
  analogWrite(IN_2, motorSpeed);
}

void indietro(int motorSpeed) {
  digitalWrite(rele_1, LOW);
  digitalWrite(rele_2, LOW);
  delay (100);
  analogWrite(IN_1, motorSpeed);
  analogWrite(IN_2, motorSpeed);
}

void destra(int motorSpeed) {
  digitalWrite(rele_1, HIGH);
  digitalWrite(rele_2, LOW);
  delay (100);
  analogWrite(IN_1, motorSpeed);
  analogWrite(IN_2, motorSpeed);
}

void sinistra (int motorSpeed) {
  digitalWrite(rele_1, LOW);
  digitalWrite(rele_2, HIGH);
  delay (100);
  analogWrite(IN_1, motorSpeed);
  analogWrite(IN_2, motorSpeed);
}

void fermo() {
  digitalWrite(rele_1, LOW);
  digitalWrite(rele_2, LOW);
  analogWrite(IN_1, 0);
  analogWrite(IN_2, 0);
}
```

Collegamento webcam

Vediamo ora come collegare la webcam C170 alla scheda Arduino Yún.

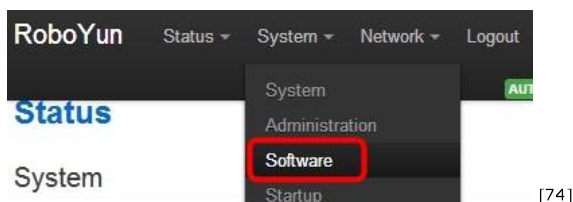
La distribuzione Linino in esecuzione su Arduino Yún è in grado di gestire le webcam supportate dal **driver UVC** [72].

Prima di collegare la webcam, dobbiamo però installare qualche pacchetto software, per questo motivo dovremmo necessariamente collegare Arduino Yún in rete con la possibilità di navigare in internet.

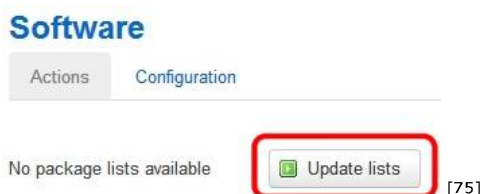
Una volta collegati con un browser (Internet Explorer, Google Chrome o quello che preferite) dovrete accedere all'interfaccia grafica di Arduino e, dopo aver inserito la password, poi "Configure" e "Advanced configuration panel":



Ora dal menu selezioniamo **System**, e poi **Software**:



Premiamo il pulsante **Update lists** per aggiornare la lista dei software disponibili:



Nel campo filtro digitiamo il nome del pacchetto **kmod-video-uvc** [76], spostiamoci nel tab **Available packages** e installiamolo:



Status

Installed packages (kmod-video-uvc) **Available packages (kmod-video-uvc)**

	Package name	Version	Description
Install	kmod-video-uvc	3.8.3-1	Kernel modules for supporting USB Video Class (UVC) devices.

Facciamo lo stesso per il pacchetto **fswebcam** [79]:



Status

Installed packages (fswebcam) **Available packages (fswebcam)**

	Package name	Version	Description
Install	fswebcam	20110717-1	fswebcam is a neat and simple webcam app. It captures images from a V4L1/V4L2 compatible device or file, averages them to reduce noise and draws a caption using the GD Graphics Library which also handles compressing the image to PNG or JPEG. The resulting image is saved to a file or sent to stdio where it can be piped to something like nctpput or scp.

Collegiamo ora la webcam alla porta USB Host di Arduino Yún . Per essere sicuri che Linino l'abbia riconosciuta, visualizziamo il **System log** di

sistema:



Tra le informazioni mostrate dovreste vedere quelle che riportano le informazioni della webcam, segno che è stata correttamente configurata:



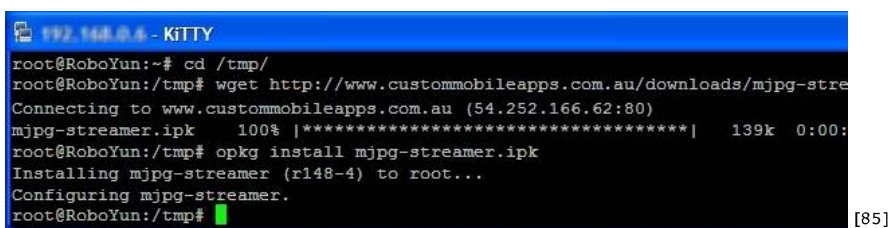
Dato che il nostro progetto prevede la trasmissione di uno streaming video, è necessario caricare un'ulteriore applicativo, realizzato sempre per Linux che è denominato **mjpg-streamer** [84]. Questo applicativo è in grado di catturare il video dalla webcam e renderlo disponibile tramite interfaccia web.

Per l'installazione dovremo spostarci nella cartella **/tmp** e scaricare il pacchetto tramite **wget** (che è un gestore di download per Linux)

wget http://www.custommobileapps.com.au/downloads/mjpg-streamer.ipk

Installiamo il pacchetto con **opkg**:

opkg install mjpg-streamer.ipk



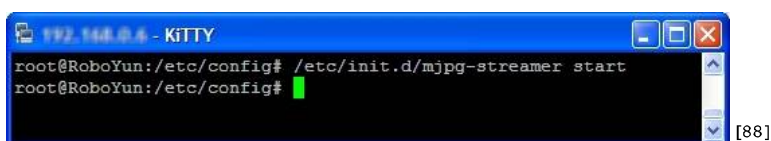
Procederemo a questo punto a modificare il file **/etc/config/mjpg-streamer** mettendo a 1 il parametro **enabled**.

Sempre nello stesso file potremo cambiare risoluzione, frame rate...,ecc.

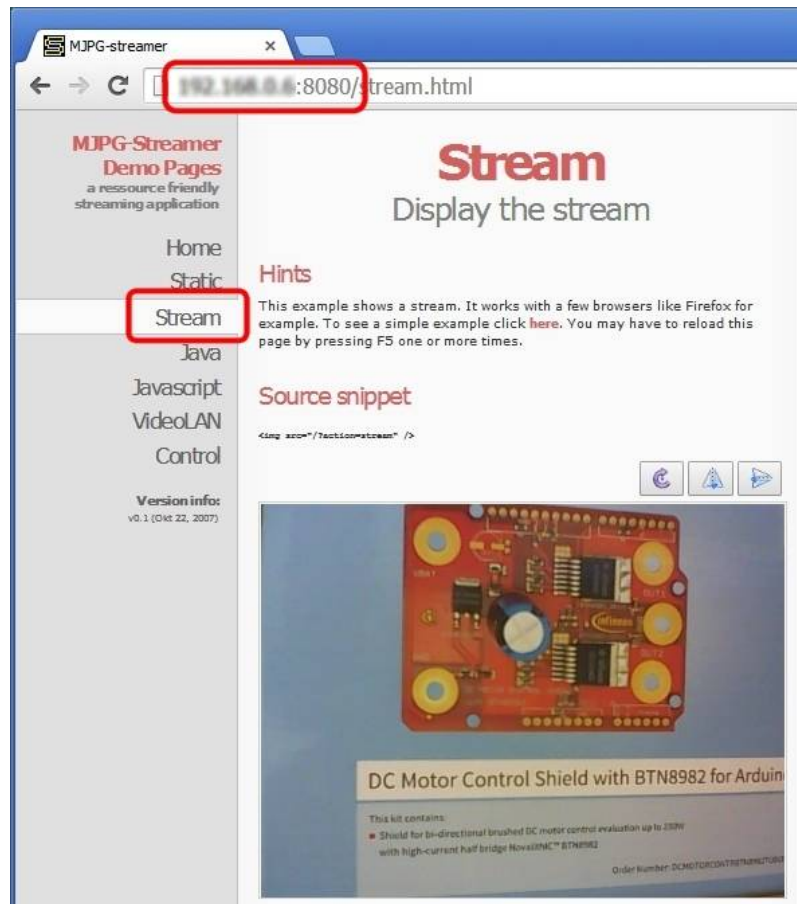
Per fare questo è stato utilizzato **WinSCP** [86] che è un client grafico open source per Windows per SFTP e FTP. La sua funzione principale è quella di copiare in modo sicuro file tra un computer locale e uno remoto, ma permette tra le altre cose di muoversi agevolmente nelle varie cartelle di Arduino Yún



Eseguiamo ora il programma:



Collegandoci con il browser all'indirizzo di Yún e alla porta **8080** possiamo visualizzare l'interfaccia di **mjpg-streamer**, e se selezioniamo la voce Stream potremo vedere il video proveniente dalla webcam. A questo punto saremo sicuri che tutto il processo d'installazione ha avuto successo.



[89]

Filmato illustrativo

E' possibile vedere l'esecuzione del programma visionando il filmato:

RoboYun Un robot a comando Wi Fi



Conclusioni

Termina così quest'articolo in cui si sono analizzate le parti che costituiscono il nostro robot a comando Wi-Fi, lo shield di Infineon e la possibilità di utilizzarlo per la gestione di una coppia di motori, aggirando la limitazione dovuta alla presenza di un solo ponte H.

Abbiamo anche visto come utilizzare una semplice webcam in accoppiamento con una scheda Arduino Yún per permettere la trasmissione di filmati in streaming.

Nel prossimo articolo vedremo l'interfaccia di comando e lo doteremo del programma completo.

Chi vorrà potrà comunque partire da questa base per realizzare una propria versione di robot, inserendo magari altri sensori, permettendo al robot di muoversi autonomamente: per esempio inserendo dei sensori radar in modo da controllare la presenza di eventuali ostacoli, mentre l'inserimento di un sensore di luce permetterebbe la gestione in automatico dell'accensione dei fari.

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

URL to article: <https://it.emcelettronica.com/roboyun-robot-esploratore-con-webcam-e-controllo-wi-fi>

URLs in this post:

- [1] LittleBot : <https://it.emcelettronica.com/presentazione-del-robot-littlebot>
- [2] altro modello: <https://it.emcelettronica.com/telecomando-tv-comandare-robot-cingolato>
- [3] 20 Arduino Motor Shield di Infineon gratis per voi: <https://it.emcelettronica.com/20-arduino-motor-shield-di-infineon-gratis-per-voi>
- [4] Turtle - 2WD Mobile Platform: http://www.dfrobot.com/index.php?route=product/product&product_id=65
- [5] Scheda Arduino Yún: <https://www.arduino.cc/en/Main/ArduinoBoardYun?from=Products.ArduinoYUN>
- [6] DC Motor Control Shield with BTN8982TA for Arduino: <http://www.infineon.com/cms/en/product/microcontroller/32-bit-industrial-microcontroller-based-on-arm-registered-cortex-tm-m/xmc-development-tools-kits-and-boards/boards-and-shields-for-arduino/channel.html?channel=5546d4614b0b239c014ba1e6c4a73098>
- [7] Webcam modello C170 della Logitech: <http://www.logitech.com/it-it/product/webcam-c170>
- [8] Dispositivo Pan & Tilt SparkFun codice ROB-10335: <https://www.sparkfun.com/products/10335>
- [9] Image: https://it.emcelettronica.com/wp-content/uploads/2015/07/descrizione_robotyun.jpg
- [10] DFRobots: <http://www.dfrobot.com/>
- [11] Turtle - 2WD Mobile Platform: http://www.dfrobot.com/index.php?route=product/product&product_id=65#.VZDfjxvtlHw
- [12] Image: https://it.emcelettronica.com/wp-content/uploads/2015/07/Turtle_2wd.gif
- [13] o in versione pdf: <http://www.dfrobot.com/image/data/ROB0005/3PA%20InstructionManual%20V1.1.pdf>
- [14] SKU: FIT0016: http://www.dfrobot.com/index.php?route=product/product&path=47&product_id=100#.VayLSKtIHw
- [15] Image: https://it.emcelettronica.com/wp-content/uploads/2015/07/motore_robot.jpg
- [16] Image: https://it.emcelettronica.com/wp-content/uploads/2015/07/dimensioni_riduttore.jpg
- [17] Image: https://it.emcelettronica.com/wp-content/uploads/2015/07/condensatori_motore.jpg
- [18] Atheros AR9331: https://wikidevi.com/wiki/Atheros_AR9331
- [19] OpenWRT: <https://openwrt.org/>
- [20] Linino: <http://www.linino.org/>
- [21] ATmega32u4: <http://www.atmel.com/devices/atmega32u4.aspx>
- [22] Arduino Leonardo: <https://www.arduino.cc/en/Main/ArduinoBoardLeonardo>
- [23] Arduino Esplora: <https://it.emcelettronica.com/scopriamo-nuova-scheda-arduino-esplora>
- [24] Image: <https://it.emcelettronica.com/wp-content/uploads/2015/07/YunParts.png>
- [25] una libreria: <https://www.arduino.cc/en/Reference/YunBridgeLibrary>
- [26] Image: https://it.emcelettronica.com/wp-content/uploads/2015/07/Bridge_arduino_yun.png
- [27] motor Shield della Infineon: <http://www.infineon.com/cms/en/product/promopages/Boards-for-Arduino/>
- [28] Image: https://it.emcelettronica.com/wp-content/uploads/2015/07/Motor_shield_Infineon_adrirobot.jpg
- [29] IPD90P04P4L: <http://www.infineon.com/cms/en/product/power/mosfet/automotive-mosfet/single-p-channel-40v-mosfet/IPD90P04P4L-04/productType.html?productType=db3a30442f69f6f4012f777c4ac00dd8>
- [30] Image: https://it.emcelettronica.com/wp-content/uploads/2015/07/protezione_alimentazione.jpg
- [31] BTN8982TA: <http://www.infineon.com/cms/en/product/power/motor-driver/integrated-half-bridge-driver/BTN8982TA/productType.html?productType=db3a30443ef951e3013f0f6c88742068>
- [32] Image: https://it.emcelettronica.com/wp-content/uploads/2015/07/infineon_motor_BTN8982TA_diagram.jpg
- [33] Image: https://it.emcelettronica.com/wp-content/uploads/2015/07/schema_ponte.jpg
- [34] Image: <https://it.emcelettronica.com/wp-content/uploads/2015/07/morsettiere.jpg>
- [35] Image: <https://it.emcelettronica.com/wp-content/uploads/2015/07/modifiche.jpg>
- [36] Banggood: <http://www.banggood.com/DHT11-Temperature-Relative-Humidity-Sensor-Module-For-Arduino-p-933153.html>
- [37] libreria DHT.h: <https://github.com/adafruit/DHT-sensor-library>
- [38] Image: https://it.emcelettronica.com/wp-content/uploads/2015/07/sensore_dht11.jpg
- [39] Gestione di un dispositivo Pan & Tilt con la scheda Arduino Esplora: <https://it.emcelettronica.com/gestione-di-un-dispositivo-pan-tilt-con-la-scheda-arduino-esplora>
- [40] Sparkfun ROB-09065: <https://www.sparkfun.com/products/9065>
- [41] Image: https://it.emcelettronica.com/wp-content/uploads/2015/07/kit-pan_tilt.jpg
- [42] libreria servo.h: <https://www.arduino.cc/en/reference/servo>
- [43] Image: <https://it.emcelettronica.com/wp-content/uploads/2015/07/c170.jpg>
- [44] L293: <http://www.ti.com/product/L293>
- [45] SN754410: <http://www.ti.com/product/SN754410>
- [46] L298: http://www.st.com/web/en/catalog/sense_power/FM142/CL851/SC1790/SS1555/PF63147
- [47] Schema formato pdf: https://it.emcelettronica.com/wp-content/uploads/2015/07/Shield_Roboyun.pdf
- [48] Image: https://it.emcelettronica.com/wp-content/uploads/2015/07/shield_yun.jpg
- [49] Image: https://it.emcelettronica.com/wp-content/uploads/2015/07/shield_yun_rele.jpg
- [50] Image: https://it.emcelettronica.com/wp-content/uploads/2015/07/rotazione_1.jpg
- [51] Image: https://it.emcelettronica.com/wp-content/uploads/2015/07/rotazione_2.jpg
- [52] Image: https://it.emcelettronica.com/wp-content/uploads/2015/07/shield_yun_output.jpg
- [53] Image: https://it.emcelettronica.com/wp-content/uploads/2015/07/shield_yun_pin.jpg
- [54] Proto shield R3: <https://www.arduino.cc/en/Main/ArduinoProtoShield>
- [55] Image: <https://it.emcelettronica.com/wp-content/uploads/2015/07/ArduinoProtoShield.jpg>
- [56] Image: https://it.emcelettronica.com/wp-content/uploads/2015/07/schema_fari_robot.jpg
- [57] Image: https://it.emcelettronica.com/wp-content/uploads/2015/07/STAMPATO_FARI.jpg
- [58] Image: https://it.emcelettronica.com/wp-content/uploads/2015/07/LM2596_DC-DC_Step_Down-Circuit.jpg

- [59] LM2596S: <http://www.ti.com/product/lm2596>
- [60] Image: https://it.emcelettronica.com/wp-content/uploads/2015/07/LM2596_DC-DC_Step_Down-foto.jpg
- [61] Image: https://it.emcelettronica.com/wp-content/uploads/2015/07/schema_collegamento.jpg
- [62] Image: https://it.emcelettronica.com/wp-content/uploads/2015/07/pannello_controllo_robot.jpg
- [63] Zepto : <http://zeptojs.com/>
- [64] versione 1.6.6: <https://www.arduino.cc/en/Main/Software>
- [65] Image: https://it.emcelettronica.com/wp-content/uploads/2015/07/IDE_arduino_ver-1.6.6.jpg
- [66] pagina reference: <https://www.arduino.cc/en/Reference/HomePage>
- [67] tutorials : <https://www.arduino.cc/en/Tutorial/HomePage>
- [68] RoboYun_test: https://it.emcelettronica.com/wp-content/uploads/2015/07/RoboYun_test.zip
- [69] console: <https://www.arduino.cc/en/Guide/ArduinoYun#toc18>
- [70] Image: https://it.emcelettronica.com/wp-content/uploads/2015/07/attivazione_programma.jpg
- [71] Image: https://it.emcelettronica.com/wp-content/uploads/2015/07/out_test_robot.jpg
- [72] driver UVC: <http://www.ideasonboard.org/uvc/>
- [73] Image: https://it.emcelettronica.com/wp-content/uploads/2015/07/cam_1.jpg
- [74] Image: https://it.emcelettronica.com/wp-content/uploads/2015/07/cam_2.jpg
- [75] Image: https://it.emcelettronica.com/wp-content/uploads/2015/07/cam_3.jpg
- [76] kmod-video-uvc: <http://wiki.openwrt.org/doc/howto/usb.video>
- [77] Image: https://it.emcelettronica.com/wp-content/uploads/2015/07/cam_4.jpg
- [78] Image: https://it.emcelettronica.com/wp-content/uploads/2015/07/cam_5.jpg
- [79] fswebcam: <http://manpages.ubuntu.com/manpages/lucid/man1/fswebcam.1.html>
- [80] Image: https://it.emcelettronica.com/wp-content/uploads/2015/07/cam_6.jpg
- [81] Image: https://it.emcelettronica.com/wp-content/uploads/2015/07/cam_7.jpg
- [82] Image: https://it.emcelettronica.com/wp-content/uploads/2015/07/cam_8.jpg
- [83] Image: https://it.emcelettronica.com/wp-content/uploads/2015/07/cam_9.jpg
- [84] mjpg-streamer: <http://sourceforge.net/projects/mjpg-streamer/>
- [85] Image: <https://it.emcelettronica.com/wp-content/uploads/2015/07/cam-10.jpg>
- [86] WinSCP: <https://winscp.net/eng/download.php>
- [87] Image: https://it.emcelettronica.com/wp-content/uploads/2015/07/cam_11.jpg
- [88] Image: <https://it.emcelettronica.com/wp-content/uploads/2015/07/cam-12.jpg>
- [89] Image: https://it.emcelettronica.com/wp-content/uploads/2015/07/cam_13.jpg

Copyright © 2017 Elettronica Open Source. All rights reserved.