

MICROCONTROLLORI Fibre ottiche SYSTEMVERILOG  
Arduino robot wireless TELEVISIONE 3D CONTROLLO CALDAIA  
LIBRERIA TRIGONOMETRICA PIC MikroBASIC CUBLOC PLC

# fe

fare elettronica

[www.farelettronica.com](http://www.farelettronica.com)

320 FEBBRAIO Mensile Anno XXVIII € 6,00

## ANTENNE WI-FI TUTTI I SEGRETI



VISIONE  
STEREOSCOPICA E TV 3D

APPLICAZIONI CON  
LE FIBRE OTTICHE

**Corso di mikroBASIC per PIC**

## CALCOLI TRIGONOMETRICI CON I PIC

## Controllo caldaia con cubloc

## Stazione di comando per ArduinoBOT



## Impara a progettare con EAGLE

**MOUSER  
ELECTRONICS**  
a M company

Distribuzione  
di semiconduttori  
e componenti per i  
progettisti elettronici



Distributore  
Autorizzato

**M** [mouser.com](http://mouser.com)

320 febbraio 2012

## Zoom in

**20** **LA TV 3D COME FUNZIONA?**  
Oggi si sente spesso parlare di televisione 3D e in commercio se ne trovano molteplici modelli. Ma come funziona un televisore 3D e quale principio sfrutta per ottenere la visione stereoscopica? Scopriamolo in questo articolo.

a cura di WWW.LATV3D.IT

## Progettare & costruire

**26** **CONTROLLO CALDAIA CON CUBLOC**

L'obiettivo di questo progetto è verificare la capacità del CUBLOC di implementare applicazioni di controllo industriale non complesse.

di Nicola Taraschi

**36** **ANTENNE WI-FI**

termini dB, dBi e dBm, pur facendo capo alla stessa unità di misura, non identificano la stessa quantità e non sono intercambiabili tra loro; dB e dBi rappresentano guadagni – o perdite, se negativi – che utilizzano un diverso termine di riferimento, mentre dBm è una misura di potenza. Cerchiamo di capire meglio l'argomento per evitare che qualcuno faccia tesoro della nostra confusione...

di Daniele Cappa



**46** **INTERFACCIA D'INGRESSO PER DATI A 16 BIT (PARTE QUARTA)**

Completiamo in questa puntata la rassegna di soluzioni alternative per consentire ai processori la lettura sincronizzata dei dati forniti dalle periferiche d'ingresso.

di Giorgio Ober

**54** **GUIDA AL PROGRAMMA EAGLE**

Chi si occupa di elettronica si sarà trovato nelle necessità di realizzare un circuito stampato. Esistono molti programmi che permettono, partendo dalla stesura del disegno del circuito di eseguire automaticamente la tracciatura delle piste da trasferire poi sulla piastra ramata. Uno di questi è il programma EAGLE, acronimo di Easily Applicable Graphical Layout Editor.

di Adriano Gandolfo

Rispondi e... **VINCI!** pag. **68**



DIVERTITI E METTI  
ALLA PROVA  
LE TUE CONOSCENZE CON  
**ELETTRO QUIZ**  
E VINCI OGNI MESE  
ESCLUSIVI PREMI!

**Elettroshop Pag. 32, 41**

Via Giotto, 7 - 20032 Cormano (MI)  
Tel. 02 66504755 - www.elettroshop.com

**Exposition Service Pag. 18**

V.le Dante Alighieri 54 - 48022 Lugo (RA)  
Tel. 0545 27548 - www.mondoelettronica.net

**Evr di Vignati Carlo Pag. 14**

Viale Kennedy, 96 - 20027 Rescaldina (MI)  
Tel. 0331-1815404 - www.evr-electronics.com

**Futura Elettronica Pag. 9**

Via Adige 11 - 21013 Gallarate (VA)  
Tel. 0331 799775 - www.futuranet.it

**Grifo Pag. 12**

Via dell'Artigiano 8/6 - 40016 San Giorgio Di Piano (BO)  
Tel. 051-892052 - www.grifo.it

**Italfiere Pag. 19**

Via Caduti di via Fani 65 - 47023 Cesena (FC)  
Tel. 0547 415674 - www.italfiere.net

**LeCroy Pag. 13**

via E. Mattei Valecenter 1/c/102a - 30020 Marcon (VE)  
Tel. 041 5997011 - www.lecroy.com

**MikroElektronika Pag. 6**

Visegradska, 1A - 11000 Belgrade  
Tel. +381 11 3628830 - www.mikroe.com

**Millennium Dataware Pag. 17**

Corso Repubblica 48 - 15057 Tortona (AL)  
Tel. 0131 860254 - www.mdsrl.it

**Mouser Electronics Pag. Icop, 3**

Centro Direzionale Milanofiori  
Strada 1 Palazzo E1 - 20090 Assago (MI)  
Tel. +39 02 57506571 - www.mouser.com

**PCB-Pool Pag. 15**

Bay 98-99 - Shannon Free Zone  
Shannon - County Clare  
Tel. 02 64672645 - www.pcb-pool.com

**RS Components Pag. IVcop**

Via M. V. De Vizzi 93/95 - 20092 Cinisello Balsamo (MI)  
Tel. 02 66058257 - rswww.it

**Tecnoimprese Pag. 79**

Via Console Flaminio, 19 - 20134 (MI)  
Tel. 02 210.111.1 - www.fortronic.it

**Teltools Pag. 16**

Via Della Martinella 9 - 20152 Milano (MI)  
www.teltools.it

**LUPUS IN FABULA**

**Avnet Abacus, 14 – Grifo, 15**  
**Linear Technology, 16 – Maxim, 16**  
**Microchip, 16 – Mikroelektronika, 14**  
**Mouser, 17 – Rigol, 14**  
**TE Connectivity, 15 – Toshiba, 17**

**70 FIBRE OTTICHE**

Le fibre ottiche sono filamenti di materiali vetrosi o polimerici, realizzati in modo da poter condurre al loro interno la luce, e che trovano importanti applicazioni in telecomunicazioni, diagnostica medica e illuminotecnica. Analizziamole in dettaglio.

di Maurizio Di Paolo Emilio

**80 CORSO MIKROBASIC**

(PARTE QUARTA)  
**LA LIBRERIA TRIGONOMETRICA**

In questo articolo si tratterà di funzioni trigonometriche di libreria del mikroBasic. Scopriremo le loro applicazioni principali e come usarle al meglio.

di Giovanni Di Maria

**Imparare & approfondire**

**74 CORSO SYSTEMVERILOG**

(PARTE NONA)  
**INTRODUZIONE ALLA LIBRERIA OVM: CREAZIONE DELLA SEQUENZA DI TEST**

Nella precedente puntata abbiamo visto come creare un ambiente per descrivere l'architettura di test e come connettere questo al DUT in un ambiente di simulazione basato sulla libreria OVM. Di seguito vedremo invece come descrivere le procedure di test per implementare il piano di verifica funzionale per il nostro sistema.

di Mariano Severi

**Robot Zone**

**94 ArduinoBOT (parte terza) LA STAZIONE DI CONTROLLO**

Siamo arrivati al termine del montaggio, passeremo ora alla realizzazione della stazione di controllo. Potremo successivamente caricare i programmi definitivi e passare all'utilizzo sul campo del nostro ArduinoBOT.

di Adriano Gandolfo

**rubriche**

- 7** Editoriale
- 10** Idee di progetto
- 14** News
- 18** Eventi
- 34** Info
- 68** Elettroquiz
- 108** IESHOP

**GLI ARTICOLI WWW ASSEGNATI COL SIMBOLO**

**sono già disponibili in formato PDF\* all'indirizzo [www.farelettronica.com/club](http://www.farelettronica.com/club)**

\*Puoi iscriverti al CLUB di Fare Elettronica versando una piccola quota annuale.



**ABBIAMO BISOGNO DEL TUO AIUTO!**

**FACCI CONOSCERE**

**LE TUE PREFERENZE VOTANDO**

**IL TUO ARTICOLO PREFERITO SUL SITO**

**WWW.FARELETRONICA.COM**



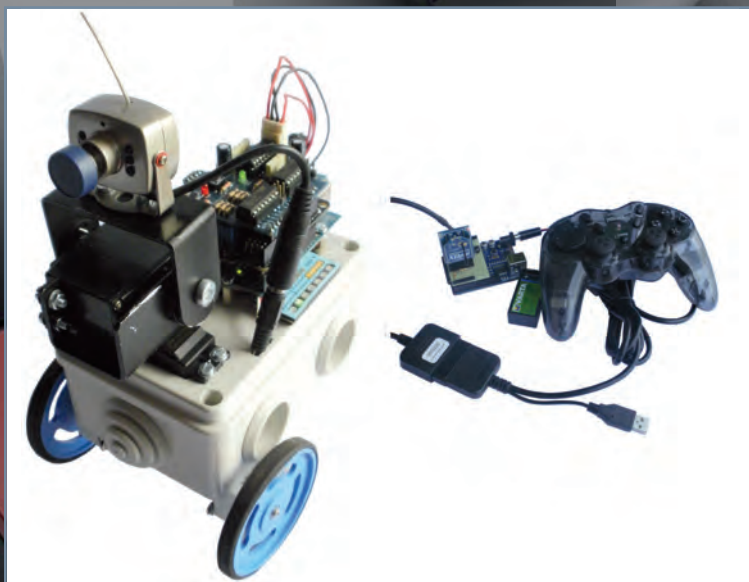
# ROBOTone

## 94 ArduinoBOT (parte quarta)

### LA STAZIONE DI CONTROLLO

*Siamo arrivati al termine del montaggio, passeremo ora alla realizzazione della stazione di controllo. Potremo successivamente caricare i programmi definitivi e passare all'utilizzo sul campo del nostro ArduinoBOT*

*di Adriano Gandolfo*



ArduinoBOT (parte quarta)

# LA STAZIONE *di* controllo

**Siamo arrivati al termine del montaggio, passeremo ora alla realizzazione della stazione di controllo.**

**Potremo successivamente caricare i programmi definitivi e passare all'utilizzo sul campo del nostro ArduinoBOT**

**N**ei precedenti articoli abbiamo analizzato le schede che formano la parte elettronica del robot, siamo poi passati alla costruzione vera e propria assemblando le varie parti. In quest'ultima puntata analizzeremo e costruiremo la stazione di controllo che ci permetterà, tramite l'utilizzo di un controller, di comandare i movimenti del robot, (figura 1) del dispositivo che movimenterà la telecamera e ci permetterà di vedere quello che viene ripreso.

Nel corso dell'articolo troverete i due programmi necessari, uno per la parte trasmittente (stazione di controllo) e una per la parte ricevente (robot).

## STAZIONE DI CONTROLLO

La stazione di controllo del robot utilizza una scheda. Su di essa è montato un piccolo shield (figura 2) che permette l'interfacciamento di un modulo Xbee e di un controller per console Playstation 2. In questo modo non è necessario essere direttamente connessi a un PC per il controllo del robot.

La stazione di controllo è formata dai seguenti elementi:

- scheda Arduino;
- shield d'interfaccia;
- scheda supporto Xbee - Simple Board con installato un modulo Xbee;
- cavo interfaccia tra shield e controller;
- controller PS2.

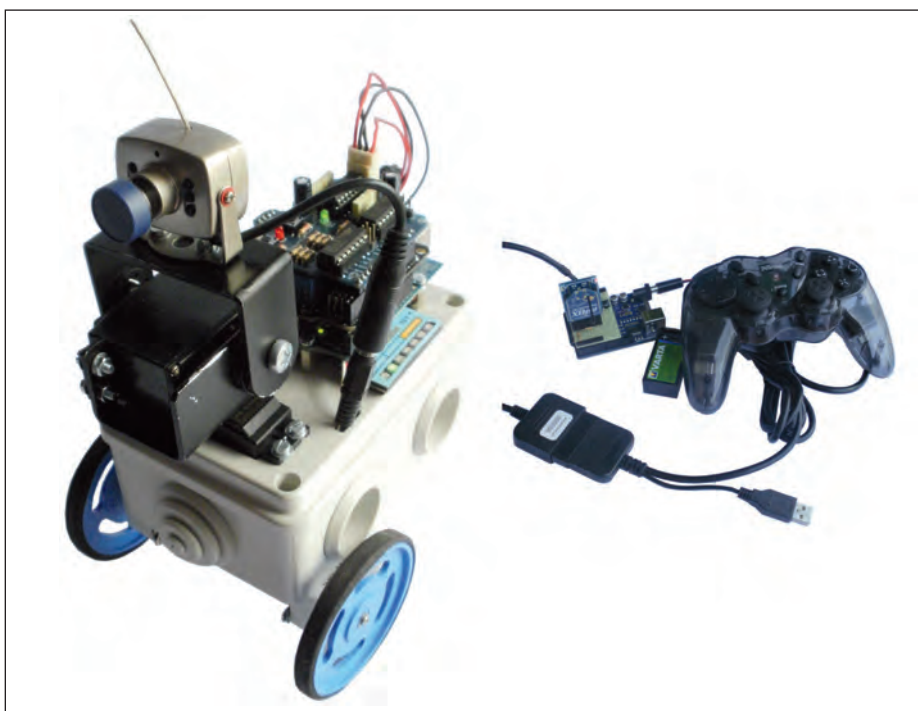


Figura 1: ArduinoBOT e la sua stazione di controllo.

### Scheda Arduino

La scheda madre della stazione di controllo è analoga a quella del robot e potrà essere una Arduino Duemilanove, Arduino UNO oppure Arduino Uno R3 (**figura 3**), per maggiori informazioni vedere il precedente articolo pubblicato sul numero 318 di *Fare Elettronica*.

### Shield d'interfaccia

Per interfacciare il modulo Xbee e il controller con la scheda Arduino è previsto l'utilizzo di un piccolo shield da auto-costruirsi (**figura 4**): su di esso, oltre ai vari connettori d'interfaccia, sono presenti un pulsante per il reset e alcuni componenti passivi (**tabella 1**).

Lo schema del circuito (**figura 5**) è molto semplice. Al connettore JP4 è collegato il controller, che utilizza un protocollo piuttosto sofisticato costruito da un'interfaccia seriale molto semplice e comune SPI (Serial Peripheral Interface). Questa interfaccia seriale sincrona utilizza quattro linee, il cui significato è riportato nella **tabella 2**.

Questa interfaccia è orientata al byte, quindi un trasferimento di base consiste in uno scambio di otto bit. Ci sono diversi parametri che devono essere concordati prima di stabilire una comunicazione: velocità, ordine dei dati, polarità del clock e fronte di clock attivo. Tuttavia, per la nostra applicazione, non dovremo occuparci di questi problemi poiché utilizzeremo un'apposita libreria per Arduino che si occuperà di convertire opportunamente i dati provenienti dal controller della PlayStation.

Le resistenze di pull-up R2 e R3 hanno la funzione di mantenere a livello alto le linee MISO e MOSI.

L'alimentazione della parte elettronica del controller in alcuni casi è 3,3 V, mentre quella in uscita dal connettore Arduino è 5 V.

Per risolvere la situazione è sufficiente inserire in serie due diodi (D1 e D2) la cui caduta di tensione pari a 0,6 V provoca la riduzione della tensione di 5 V a quella di 3,8 V ancora tollerata dai circuiti. È poi presente il connettore per il collegamento del modulo Xbee Simple Board (**figura 9**). Per quanto riguarda l'adattamento delle tensioni sulle linee TX e RX,

questo non si pone per il segnale in uscita dal modulo Xbee poiché Arduino riconosce il valore di 3,3 V ancora come valore alto. La tensione del segnale in ingresso in arrivo da Arduino pari a 5 V è troppo elevata e andrà ridotta. La soluzione utilizzata consiste nell'utilizzare un semplicissimo partitore di tensione fatto con due resistenze. Dai valori utilizzati si ricava che dal livello logico TTL alto (5 V), in uscita dal partitore, avremo una tensione pari a +3V (**figura 6**). Inoltre è pre-

sente un pulsante connesso all'apposito pin che permette di resettare il processore. Dalla scheda Arduino, tramite il connettore JP2, è prelevata la tensione di +5V e il contatto di massa.

### Costruzione della scheda

Vista la semplicità dello schema, il montaggio potrà essere realizzato utilizzando uno spezzone di basetta millefori dalle dimensioni di 35x53 mm (13x20 fori); per

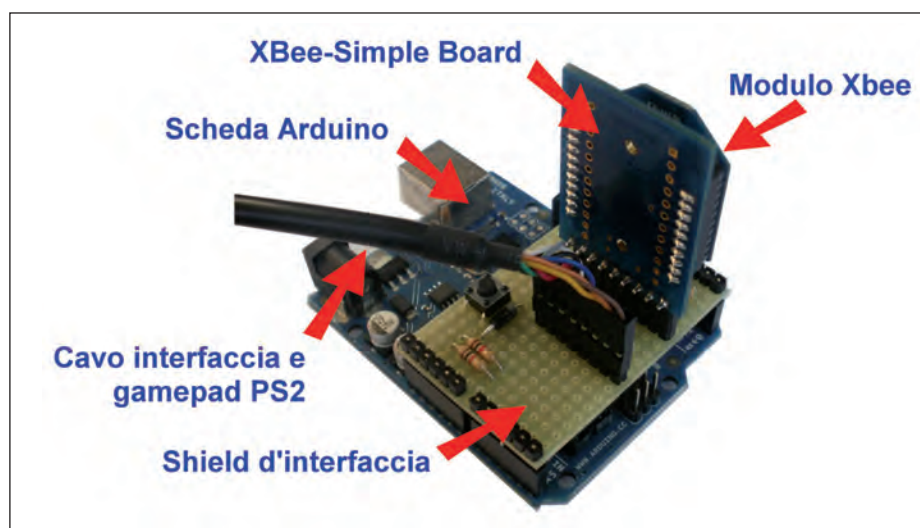


Figura 2: stazione di controllo.

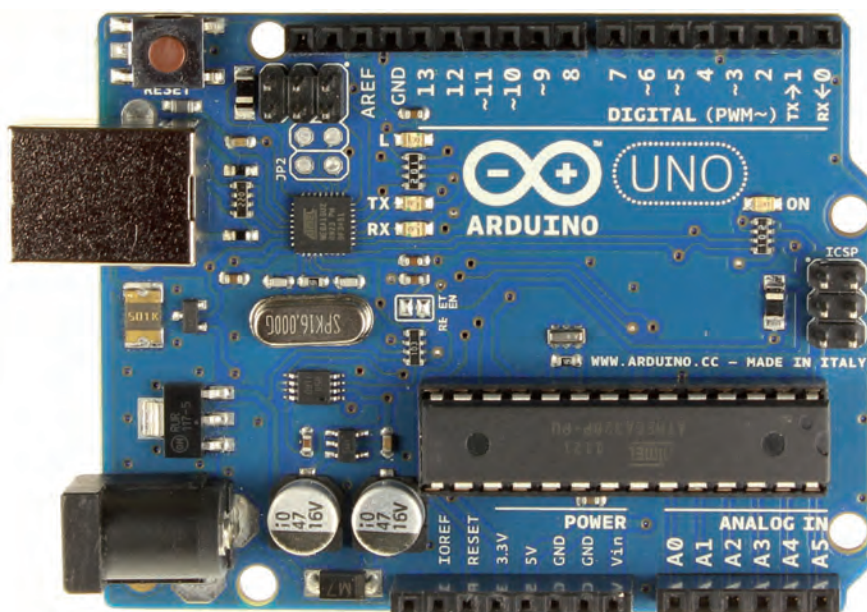


Figura 3: scheda Arduino UNO.

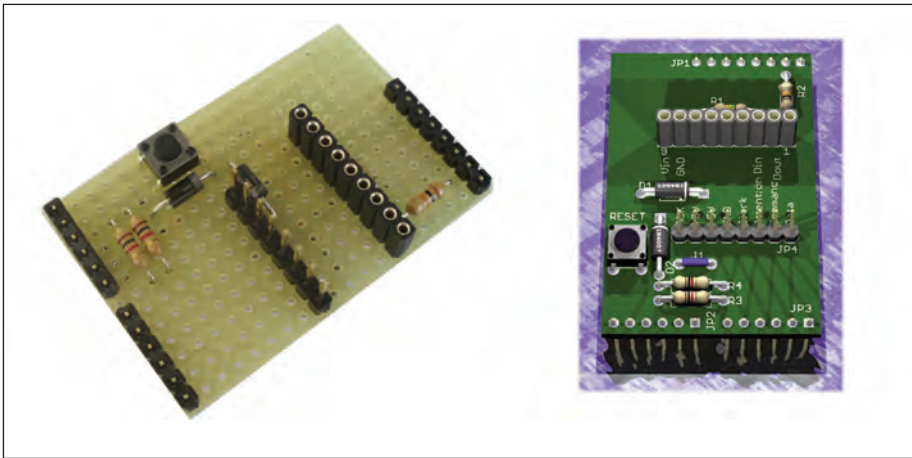


Figura 4: shield d'interfaccia Arduino-Controller.

la disposizione dei componenti ci si potrà rifare alla **figura 8**.

Se si optasse invece per la realizzazione di un'apposito circuito stampato si potrà scaricare, dal sito di *Fare Elettronica*, il PDF che riporta la traccia in scala 1:1. Per la sua realizzazione si utilizzerà una basetta in vetronite (monofaccia) di dimensioni 35x53mm circa (**figura 7**). Il metodo potrà essere quello della fotoincisione o del trasferimento termico utilizzando i cosiddetti fogli blu (PRESS-N-PELL).

Una volta inciso il rame, si verificherà in controluce, o mediante l'utilizzo di un multimetro, che non vi siano cortocircui-

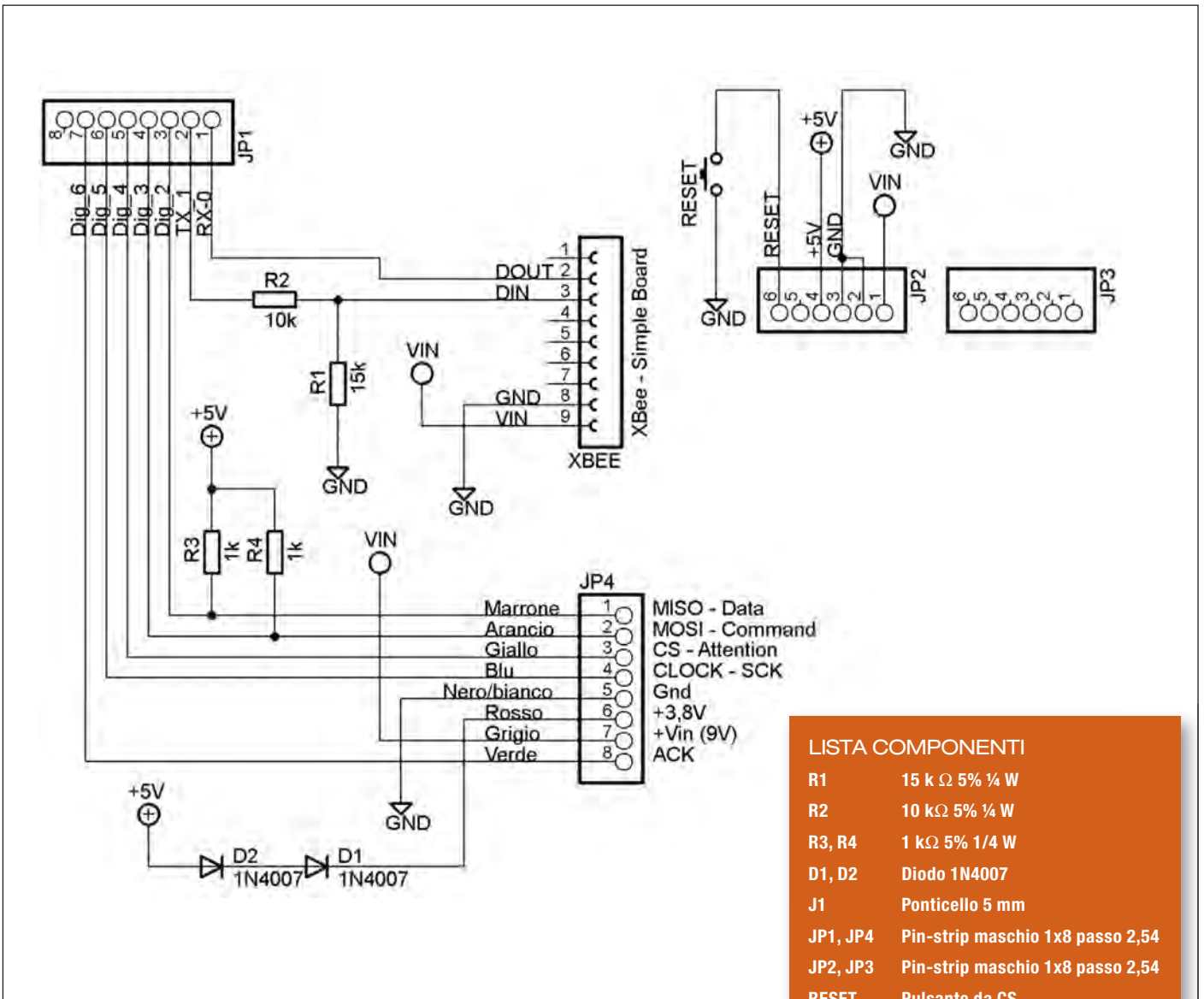


Figura 5: schema elettrico shield interfaccia.

**LISTA COMPONENTI**

R1	15 k $\Omega$ 5% 1/4 W
R2	10 k $\Omega$ 5% 1/4 W
R3, R4	1 k $\Omega$ 5% 1/4 W
D1, D2	Diode 1N4007
J1	Ponticello 5 mm
JP1, JP4	Pin-strip maschio 1x8 passo 2,54
JP2, JP3	Pin-strip maschio 1x8 passo 2,54
RESET	Pulsante da CS
XBEE	Pin-strip femmina 1x9 passo 2,54

$$V_{out} = V_{in} \cdot \frac{R_1}{R_1 + R_2} = 5 \cdot \frac{15}{10 + 15} = 3V$$

Figura 6: calcolo tensione in uscita dal partitore.

## TABELLA 2

<b>SCK:</b>	<b>Serial Clock:</b>	Emesso dal master, in questo caso Arduino
<b>MISO:</b>	<b>Master Input Slave Output</b>	Ingresso per il master e uscita per lo slave
<b>MOSI:</b>	<b>Master Output Slave Input</b>	Uscita dati dal master
<b>CS:</b>	<b>Chip Select</b>	Emesso dal master per scegliere con quale dispositivo slave vuole comunicare, la linea non è indispensabile in tutte le applicazioni

Tabella 2: protocollo SPI.

## PER approfondire...

- <http://www.arduino.cc/> - Sito del produttore della scheda Arduino, dove trovare informazioni sulla scheda e scaricare l'ambiente di sviluppo.
- <http://arduino.cc/playground/MotorControlShieldV3/0> - Informazioni sulla scheda controllo motori.
- <http://www.dfrobot.com/> - Sito del produttore della I/O Expansion Shield.
- <http://www.solarbotics.com/> - Sito del produttore dei motoriduttori.
- <http://www.lynxmotion.com> - Sito di Lynxmotion per il dispositivo Pan & Tilt e cavo interfaccia controller.
- <http://www.pololu.com/> - Sito del produttore del modulo DC/DC.
- <http://www.digi.com/> - Sito del produttore dei moduli XBee.
- <http://www.droids.it> - Sito per la scheda XBee - Simple Board.
- [http://pspunch.com/pd/article/arduino\\_lib\\_gpsx\\_en.html](http://pspunch.com/pd/article/arduino_lib_gpsx_en.html) - Sito originale libreria GPSX
- <http://www.arduino.cc/playground/Italiano/ArduinoPsx> - Descrizione in italiano sull'utilizzo della libreria GPSX.
- <http://arduino.cc/playground/Main/PSXLibrary> - Sito per libreria PSX.
- <http://store.curiousinventor.com/guides/PS2> - Informazioni sul connettore PSX.
- <http://www.nchsoftware.com/capture/index.html> - Programma di videocattura.
- <http://www.robot-italy.com/> - Sito per la scheda XBee - Simple Board e Xbee - USB Board.

ti soprattutto tra le piste più vicine. Si passerà quindi alla foratura della stessa, utilizzando principalmente una punta da 0,8-1 mm.

Quindi si posizioneranno e salderanno i componenti seguendo lo schema riportato nella **figura 8**. Per la saldatura si utilizzerà un piccolo saldatore a punta fine, della potenza di circa 25-30 W.

Si inizierà montando il ponticello, le resistenze, i diodi. Ricordando che questi ultimi sono polarizzati, occorre controllare la posizione della fascia sul loro corpo. Si proseguirà con il pulsante e si terminerà montando i vari connettori. Si è così concluso il montaggio della scheda.

### Scheda per modulo XBee

La scheda scelta per il montaggio del modulo XBee è denominata **XBee - Simple Board** ed è prodotta da Droids (**figura 9**). Come visibile sullo schema elettrico (**figura 11**), è presente un regolatore di tensione fisso tipo LM2937 che fornisce 3,3 V - 500 mA per l'alimentazione del modulo Xbee, e sono inoltre presenti due LED che consentono il costante monitoraggio dell'attività della scheda.

La scheda permette il collegamento di tutti i pin presenti sul modulo Xbee (**figura 10**). Altre informazioni sul modulo Xbee possono essere trovate sul precedente articolo pubblicato su *Fare Elettronica* n°318. Per impartire i comandi al nostro robot si utilizza un controller che di solito è utilizzato nella console per videogiochi tipo Playstation 2 di Sony (**figura 12**). Per pilotare ArduinoBOT può essere utilizzato qualunque controller compatibile con la versione originale. Su di esso sono presenti sedici pulsanti di cui quattro formano un pad direzionale e quattro tasti (identificati dai simboli triangolo, cerchio,

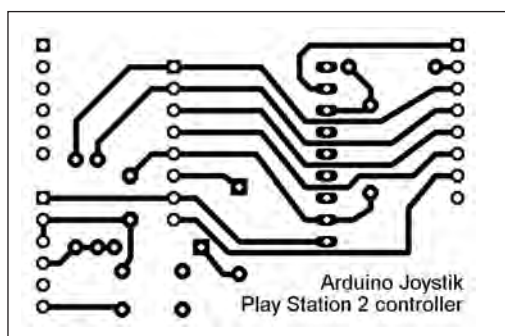


Figura 7: circuito stampato.

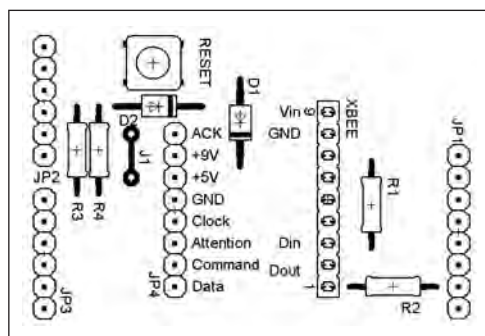


Figura 8: piano di montaggio dei componenti.





Figura 9: XBee - Simple Board.

croce e quadrato). Essi verranno utilizzati per il movimento della telecamera. Sono poi presenti due joystick analogici (figura 14).

### Controller per console per Playstation 2

Anche se ogni tasto può essere configurato per eseguire un'azione specifica, tutti funzionano con lo stesso principio. Ogni pulsante è un interruttore che completa un circuito quando è premuto; come già detto, sono presenti due joystick analogici realizzati con due potenziometri (resistenze variabili) posti ad angolo retto tra loro sotto il joystick (figura 13). In base alla posizione, il joystick varia la resistenza del potenziometro interno e, di conseguenza, anche la tensione in uscita. Leggendo questa tensione con un convertitore analogico è possibile determinare l'angolo in cui si trova il joystick, e determinare la risposta appropriata sulla base di tale angolo.

Un'altra particolarità del controller (se questo è della serie Dual Shock) è il cosiddetto *force feedback*, che fornisce una stimolazione tattile. Il force feedback è realizzato attraverso l'uso di un dispositivo molto comune: un semplice motore elettrico, inserito in ogni impugnatura. Sull'albero di ciascun motore è montato un peso sbilanciato. Quando l'alimentazione è fornita al motore, il peso inizia a girare. Poiché il peso è sbilanciato, il motore cerca di oscillare ma, essendo il motore bloccato all'interno del controller, l'oscillazione si traduce in una vibrazione del controller stesso. L'azionamento dei motori è controllato da un processore che li codifica e li tra-

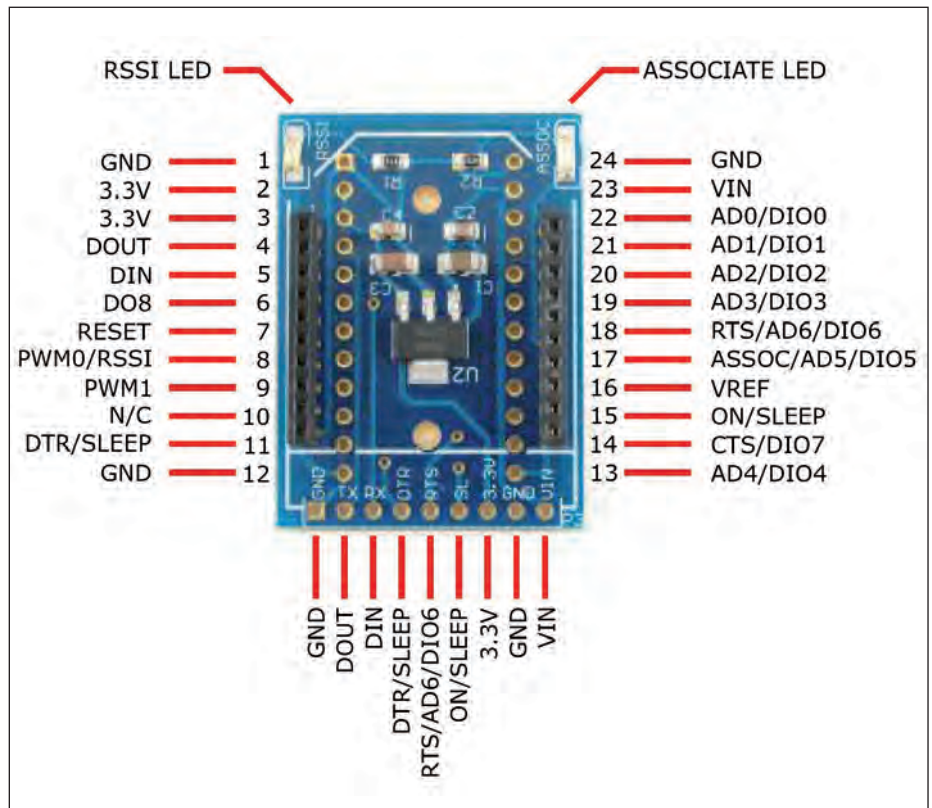


Figura 10: descrizione dei pin della scheda Simple Board.

TABELLA 3

PIN	NOME	DESCRIZIONE
1	DATA	Questo pin porta il segnale che il controllore invia ad Arduino ogni volta che si preme un tasto. Si tratta di una trasmissione seriale a 8-bit.
2	COMMAND	Questo pin è utilizzato da Arduino per inviare informazioni al controller. Tali informazioni possono azionare i motori in un controller Dual Shock al momento opportuno. Si utilizza una trasmissione seriale a 8 bit.
3	NON USATO	
4	MASSA	
5	ALIMENTAZIONE	Questo pin fornisce l'alimentazione che può variare da 3 a 5 V.
6	ATT	Questo pin è utilizzato da Arduino per avvertire il controller che sta per trasmettere dati. ATT è usato per ottenere l'attenzione del controller. Questo segnale andrà a livello basso per la durata di una trasmissione.
7	CLOCK	Usato per mantenere le unità in sincronia. Segnale inviato dalla scheda Arduino.
8	NON USATO	
9	ACKNOWLEDGE	Questo pin invia un segnale alla scheda Arduino dal controller dopo ogni comando che si riceve sul pin 2.

Tabella 3: significato dei vari pin del connettore.

## LISTATO 1

```
/*
Programma_controllo_ArduinoBOT.pde
Programma per stazione controllo
robot ArduinoBOT
    Vengono utilizzati i seguenti pin
Pin +5V      -> +5V
Pin GND      -> GND
Pin digitale 0 -> RX XBee
Pin digitale 1 -> TX XBee
Pin digitale 2 -> Pin Data Gamepad - ca-
vo marrone
Pin digitale 3 -> Pin Command Gamepad -
cavo arancio
Pin digitale 4 -> Pin Attention Gamepad -
cavo giallo
Pin digitale 5 -> Pin Clock Gamepad - ca-
vo blu
Pin digitale 6 -> Pin ACK Gamepad - cavo
verde
    Creato il 4/12/2011 da Adriano Gandolfo
*/

#if defined(ARDUINO) && ARDUINO >= 100
#include "Arduino.h"
#else
#include "WProgram.h"
#endif
#include <Psx.h> // Includes the Psx Li-
brary
#define dataPin 2
#define cmndPin 3
#define attPin 4
#define clockPin 5

Psx Psx; // Initializes the library
unsigned int data = 0; // data stores
the controller response
void setup()
{
    Psx.setupPins(dataPin, cmndPin, attPin,
clockPin, 10); // Defines what each pin is
used
    // (Data Pin #, Cmnd Pin #, Att Pin #, Clk
Pin #, Delay)
    // Delay measures how long the clock
remains at each state,
    // measured in microseconds.
    Serial.begin(9600);
}
void loop()
{
    data = Psx.read(); // Psx.read() initiates
the PSX controller and returns the button
data
    if (data & psxLeft){
        Serial.print ("a");
    }
    if (data & psxDown){
        Serial.print ("s");
    }
    if (data & psxRight){
        Serial.print ("d");
    }
    if (data & psxUp){
        Serial.print ("w");
    }
    if (data & psxSqu){
        Serial.print ("f");
    }
    if (data & psxX){
        Serial.print ("g");
    }
    if (data & psxO){
        Serial.print ("h");
    }
    if (data & psxTri){
        Serial.print ("t");
    }
    if (data & psxR1){
        Serial.print ("b");
    }
    if (data & psxL1){
        Serial.print ("x");
    }
    delay (500);
}
```

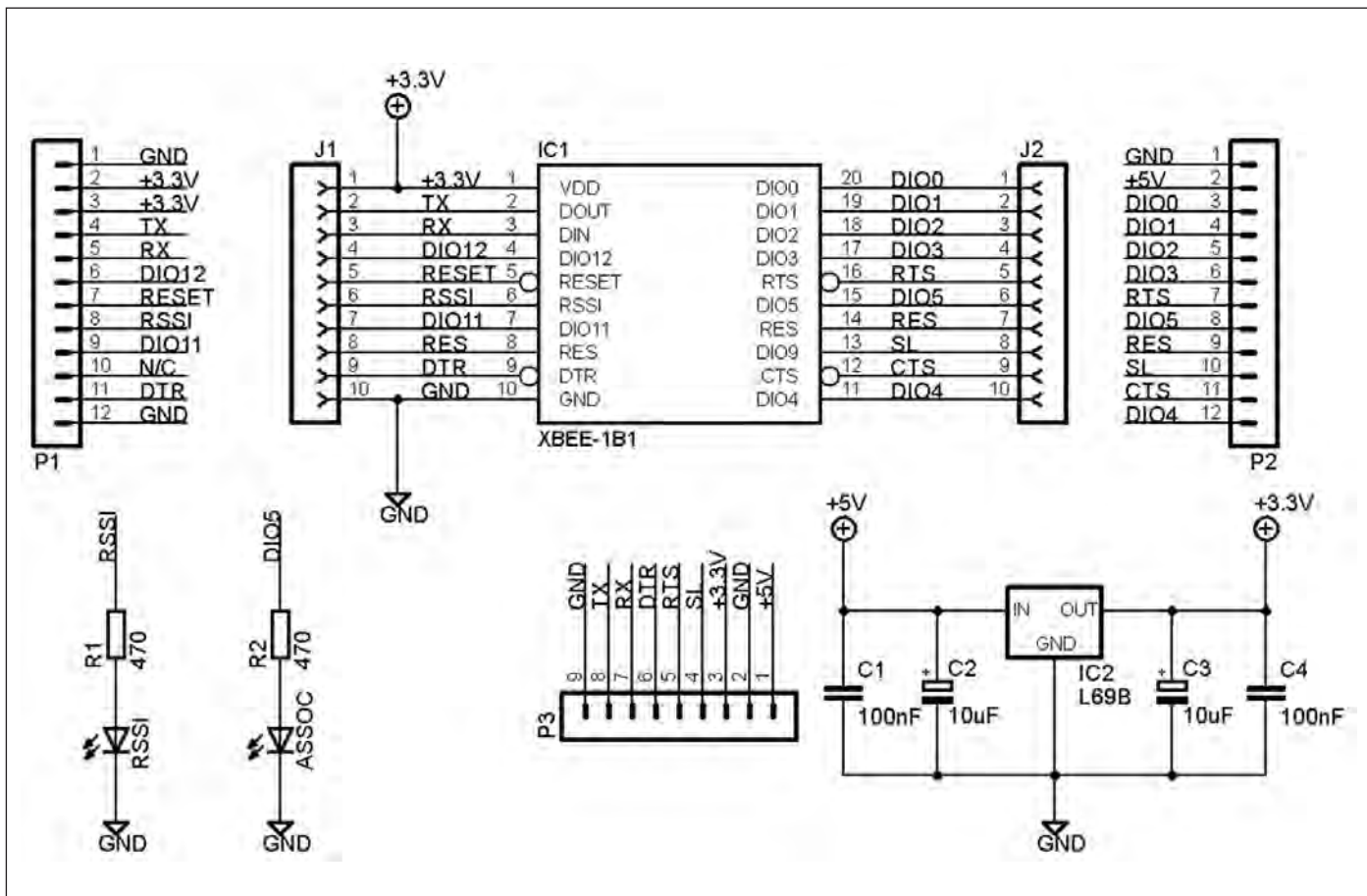


Figura 11: schema della scheda XBee - Simple Board.

smette tramite una linea seriale a 8 bit. L'uscita è rappresentata da uno speciale connettore a 9 pin (**figura 15**) le cui funzioni dei pin sono riportate nella **tabella 3**.

### CAVO INTERFACCIA CONTROLLER-SHIELD

Come abbiamo visto il controller ha uno speciale connettore per il collegamento alla console di gioco. Per evitare di dover tagliare il cavo è possibile utilizzare un apposito adattatore (**figura 16**) reperibile sul sito Lynxmotion con il codice PS2C-01. Il cavo presenta da un lato lo speciale connettore mentre all'altro capo i cavi sono raggruppati in quattro connettori secondo lo schema riportato nella **figura 17**.

### UTILIZZO DELLE LIBRERIE

Per l'interpretazione dei comandi forniti dal controller verso Arduino sono disponibili due tipologie di librerie, **GPSX** e **PSX**. La prima, denominata **GPSX**, è più completa poiché permette anche di monitorare

i joystick analogici e il controllo dei due motori che forniscono il force feedback. La seconda, **PSX**, è più semplice e fornisce solo l'indicazione dei tasti premuti.

#### Libreria GPSX

La libreria GPSX è in grado di gestire due joystick. I tasti che possono essere premuti sul joystick e definiti dalla libreria, sono:

- SELECT(padNo);
- LEFT(padNo);
- DOWN(padNo);
- RIGHT(padNo);
- UP(padNo);
- START(padNo);
- STICK\_RIGHT(padNo);
- STICK\_LEFT(padNo);
- SQUARE(padNo);
- CROSS(padNo);
- IRCLC(padNo);
- TRIANGLE(padNo);
- R1(padNo);
- L1(padNo);

segue a pag. 104

TABELLA 4

psxLeft	0x0001
psxDown	0x0002
psxRight	0x0004
psxUp	0x0008
psxStrt	0x0010
psxSlct	0x0080
psxSqu	0x0100
psxX	0x0200
psx0	0x0400
psxTri	0x0800
psxR1	0x1000
psxL1	0x2000
psxR2	0x4000
psxL2	0x8000

Tabella 6: tasti riconosciuti dalla libreria.

## LISTATO 2

```

/*
  ArduinoBOT_RX.pde
  Programma per gestione robot ArduinoBOT
  Vengono utilizzati i seguenti pin
  Pin +5V      -> +5V
  Pin GND      -> GND
  Pin Vin      -> +6V Batteria
  Pin digitale 0 -> RX XBee
  Pin digitale 1 -> TX Xbee
  Pin Digitale 3 -> Servo movimento Tilt (al-
to-basso)
  Pin Digitale 6 -> Servo movimento Pan
(destra-sinistra)
  Pin Digital 11 -> Abilita Motore SX
  Pin Digital 10 -> Abilita Motore DX
  Pin Digital 13 -> Direzione Motore SX
  Pin Digital 12 -> Direzione Motore DX

  Creato il 19/12/2010
  da Adriano Gandolfo <http://www.adrirobot.it>
  This example code is in the public domain.
  */
#include <Servo.h> // Libreria per controllo
servomotori
Servo PanServo; // Creazione dell'oggetto
PanServo
Servo TiltServo; // Creazione dell'oggetto
TiltServo
int tilt = 3; // Pin a cui è collegato
il servo Tilt
int pan = 6; // Pin a cui è collegato
il servo Pan
int E1 = 11; // Pin per abilitazione
Motore DX
int E2 = 10; // Abilita Motore SX
int M1 = 13; // Direzione Motore DX
int M2 = 12; // Direzione Motore SX
int Pan_neutro = 40; // Posizione iniziale
del servo Pan
int Tilt_neutro = 60; // Posizione iniziale
del servo Tilt
int AngoloPan = Pan_neutro; // Posizione
iniziale del servo Pan
int AngoloTilt = Tilt_neutro; // Posizione in-
iziale del servo Tilt

byte DeltaPan = 3; // movimento di
3 gradi
byte DeltaTilt = 3; // movimento di
3 gradi
void setup(void)
{
  Serial.begin(9600); //imposta la por-
ta di comunicazione con L'XBee
  pinMode(E1, OUTPUT); // Imposta il pin
come uscita
  pinMode(E2, OUTPUT);
  pinMode(M1, OUTPUT);
  pinMode(M2, OUTPUT);
  PanServo.attach(pan); // Collega al pin
relativo l'oggetto PanServo
  TiltServo.attach(tilt); // Collega al pin
relativo l'oggetto TiltServo
  PanServo.write(AngoloPan); //Invia al
PanServo gli impulsi necessari a portarlo al-
l'angolo indicato
  TiltServo.write(AngoloTilt); //Invia al
TiltServo gli impulsi necessari a portarlo al-
l'angolo indicato
}
void loop(void)
{
  while (Serial.available() < 1) {
  } // Attesa sino a
quando riceve un carattere
  char val = Serial.read(); // Legge il carattere
dal modulo XBee e lo salva nella variabile val
  switch(val) // Eseguie i co-
mandi in base al carattere ricevuto
  {
    case 'w': // Se il carattere ricevuto è 'w'
esegue la routine Robot avanti
      robot_avanti ();
      break;
    case 's': //Se il carattere ricevuto è 's' e-
segue la routine Robot indietro
      robot_indietro ();
      break;
    case 'a': //Se il carattere ricevuto è 'a' e-
segue la routine Robot sinistra
      robot_sinistra ();
  }
}

```

```

        break;
    case 'd'://Se il carattere ricevuto è 'd' e-
segue la routine Robot destra
        robot_destra ();
        break;
    case 'x'://Se il carattere ricevuto è 'a' e-
segue la routine Robot fermo
        stop();
        break;
    case 'f'://Se il carattere ricevuto è 'f'
ruota a destra il servo Pan
        AngoloPan += DeltaPan;
        pan_tilt();
        break;
    case 'h'://Se il carattere ricevuto è 'h'
ruota a sinistra il servo Pan
        AngoloPan -= DeltaPan;
        pan_tilt();
        break;
    case 'g'://Se il carattere ricevuto è 'g'
ruota in avanti il servo Tilt
        AngoloTilt += DeltaTilt;
        pan_tilt();
        break;
    case 't'://Se il carattere ricevuto è 't'
ruota in indietro il servo Tilt
        AngoloTilt -= DeltaTilt;
        pan_tilt();
        break;
    case 'b'://Se il carattere ricevuto è 'b'
centra la telecamera
    {
        AngoloPan=Pan_neutro;
        AngoloTilt=Tilt_neutro;
        pan_tilt();
        break;
    };
default:
    stop(); // Ferma il robot
    break;
}
}
void stop(void) //Robot fermo
{
    digitalWrite(E1,LOW); //Disabilita il co-
mando del motore
    digitalWrite(E2,LOW);
}
void robot_indietro()
{
    digitalWrite(E1,HIGH);
    digitalWrite(M1,LOW);
    digitalWrite(E2,HIGH);
    digitalWrite(M2,LOW);
}
void robot_avanti ()
{
    digitalWrite(E1,HIGH);
    digitalWrite(M1,HIGH);
    digitalWrite(E2,HIGH);
    digitalWrite(M2,HIGH);
}
void robot_sinistra ()
{
    digitalWrite(E1,HIGH);
    digitalWrite(M1,HIGH);
    digitalWrite(M2,HIGH);
    digitalWrite(M2,LOW);
}
void robot_destra ()
{
    digitalWrite(E1,HIGH);
    digitalWrite(M1,LOW);
    digitalWrite(M2,HIGH);
    digitalWrite(M2,HIGH);
}
void pan_tilt()
{
    // Controlla i limiti per il movimento Pan
    AngoloPan=(AngoloPan < 20)?20:AngoloPan;
    AngoloPan=(AngoloPan > 80)?80:AngoloPan;
    // Controlla i limiti per il movimento Tilt
    AngoloTilt=(AngoloTilt < 60)?60:AngoloTilt;
    AngoloTilt=(AngoloTilt > 100)?100:An-
goloTilt;
    // Invia ai servo i comandi necessari
    PanServo.write(AngoloPan);
    TiltServo.write(AngoloTilt);
}
}

```



Figura 12: controller per console per Playstation 2.



Figura 13: struttura del Joystick presente nel controller.



Figura 14: disposizione tasti sul controller.

segue da pag. 100

- R2(padNo);
- L2(padNo).

Per quanto riguarda i pulsanti, sono considerati tre casi:

- sono premuti, quindi l'istruzione sarà **PRESSED\_nome\_tasto(padNo)** e l'istruzione restituirà un 1 se è vera o 0 se è falsa;
- restano premuti, l'istruzione sarà **IS\_DOWN\_nome\_tasto(padNo)** e l'istruzione restituirà un 1 se è vera o uno 0 se è falsa.

Es. `while(IS_DOWN_SQUARE(PSX_PAD1) ==1){ //fai qualcosa} Arduino fa qualcosa dentro al loop while finché il tasto quadrato rimane premuto;`

- sono rilasciati, quindi l'istruzione sarà **RELEASED\_nome\_tasto(padNo)** e l'istruzione restituirà un 1 se è vera o uno 0 se è falsa. Inoltre possiamo leggere dati analogici dai due joystick sulla manopola lungo i due assi, i valori restituiti dalle seguenti istruzioni saranno compresi tra 0 e 255.

- `ANALOG_RIGHT_X(padNo);`
- `ANALOG_RIGHT_Y(padNo);`
- `ANALOG_LEFT_X(padNo);`
- `ANALOG_LEFT_Y(padNo);`

Nelle manopole della Playstation 2 ci sono due motori (Dual Shock) ed è possibile abilitarli e disabilitarli. Un motore fornisce una vibrazione più forte, mentre l'altro una più lieve.

- `PSX.motorEnable(padNo, Motor1Enable, Motor2Enable)`

Dove:

- `padNo = PSX_PAD1 | PSX_PAD2;`
- `Motor1Enable = MOTOR1_ENABLE | MOTOR1_DISABLE;`
- `Motor2Enable = MOTOR2_ENABLE | MOTOR2_DISABLE.`

La vibrazione è regolabile via software con una semplice istruzione:

- `PSX.motor(padNo, Motor1OnOff, uint8_t Motor2Speed)`

dove:

- `padNo = PSX_PAD1 | PSX_PAD2;`
- `Motor1OnOff = MOTOR1_ON | MOTOR1_OFF;`

- `Motor2Speed = uint8_t` che assumerà il valore zero se sarà uguale a 0x00 fino ad arrivare a 255 se posto uguale a 0xFF. Possiamo infine decidere, via software, se usare la manopola in modalità digitale, (quindi

TABELLA 5

**PIN SCHEDA ARDUINO**

**FUNZIONE NEL ROBOT ARDUINOBOT**

Pin +5V	+5V
Pin GND	GND
Pin Vin	+6V Batteria
Pin digitale 0	RX XBee
Pin digitale 1	TX XBee
Pin digitale 3	PWM pin 3 servo TILT (alto-basso)
Pin Digitale 6	PWM pin 10 servo PAN (destra-sinistra)
Pin Digitale 13	Direzione Motore DX (DIR B)
Pin Digitale 12	Direzione Motore SX (DIR A)
Pin Digitale 11	Attivazione Motore DX (SP B)
Pin Digitale 10	Attivazione Motore SX (SP A)

Tabella 7: pin utilizzati nella scheda Arduino.

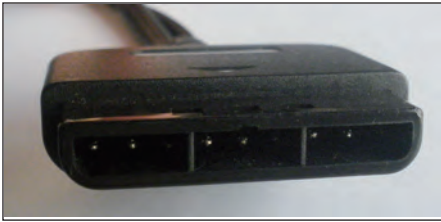


Figura 15: connettore del controller.

il joystick sinistro avrà lo stesso effetto di freccia su, freccia giù, sinistra-destra, e il joystick destro avrà lo stesso effetto di triangolo, cerchio quadrato e x), oppure in modalità analogica quindi i joystick, se mossi, restituiranno un valore analogico come visto nella funzione precedente **ANALOG\_xxx\_YYY\_asse(padNo)**:

• `PSX.mode(padNo, mode, lock)`

dove:

- `padNo` = `PSX_PAD1` | `PSX_PAD2`;
- `mode` = `MODE_DIGITAL` | `MODE_ANALOG`;
- `lock` = `MODE_UNLOCK` | `MODE_LOCK`.

### UTILIZZO DELLA LIBRERIA

Per utilizzare i comandi bisognerà prima di tutto scaricare la libreria da questo link: <http://pspunch.com/pd/files/library/GPSX.zip>

Il file andrà poi scompattato all'interno della cartella GPSX a sua volta posta nella cartella delle librerie Arduino. Dentro questa cartella troverete i seguenti file:

- `GPSX.c`
- `GPSX.h`
- `GPSXClass.cpp`
- `GPSXClass.h`
- `keywords.txt`
- `README_en.txt`
- `README_ja.txt`
- `Schematic.bmp`

È compresa anche una cartella denominata "Example" al cui interno troverete il programma "interface\_pad1.pde" per testare il controller.

Prima di utilizzare il programma occorrerà editare la libreria `GPSX.c` con un editor di testo (anche Notepad andrà bene), e una volta aperto il file sostituire nelle righe `#define` il numero del pin effettivamente utilizzato.

Per cui le linee:

```
#define DAT_PIN 7 // CTL1 Data
#define CMD_PIN 6 // CTL2 Command
```

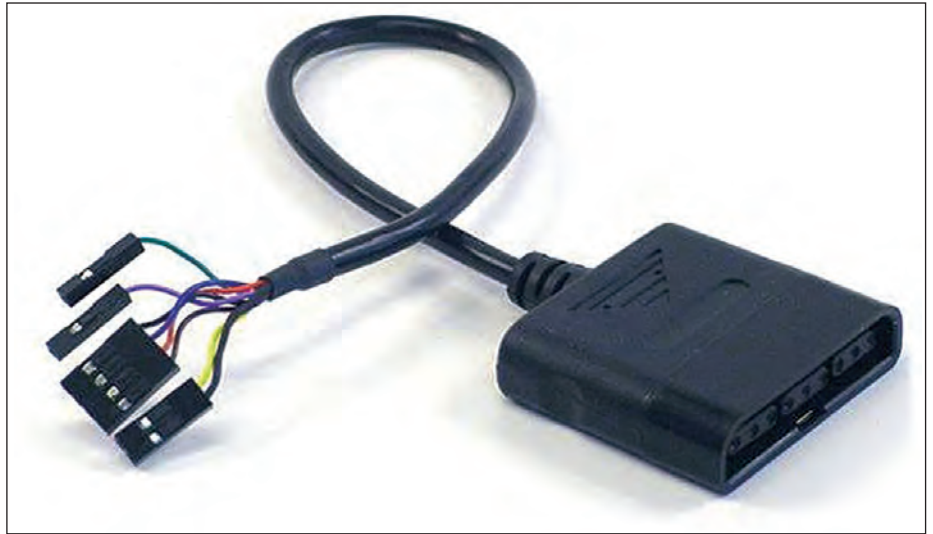


Figura 16: cavo d'interfaccia per controller.

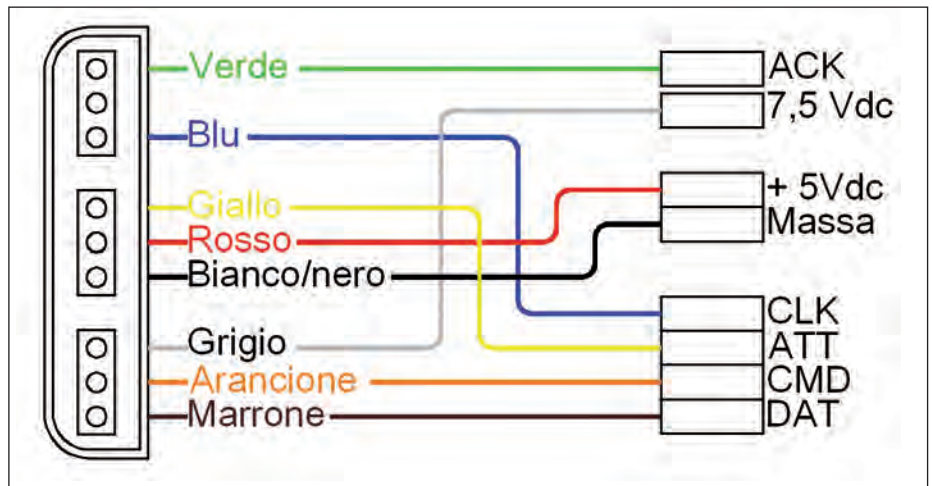


Figura 17: descrizione dei cavi.

TABELLA 6

TASTO CONTROLLER	CARATTERE INVIATO	FUNZIONE
Tasto direzione SU	W	Il robot si muove in avanti
Tasto direzione GIÙ	S	Il robot arretra
Tasto direzione SINISTRA	A	Il robot ruota a sinistra
Tasto direzione DESTRA	D	Il robot ruota a destra
Tasto L1	X	Il robot si ferma
Tasto TRIANGOLO	T	La telecamera si muove verso il basso
Tasto X	G	La telecamera si muove verso l'alto
Tasto QUADRATO	F	La telecamera ruota a sinistra
Tasto CERCHIO	H	La telecamera ruota a destra
Tasto R1	B	La telecamera si porta in posizione neutra

Tabella 8: tasti di comando del robot.

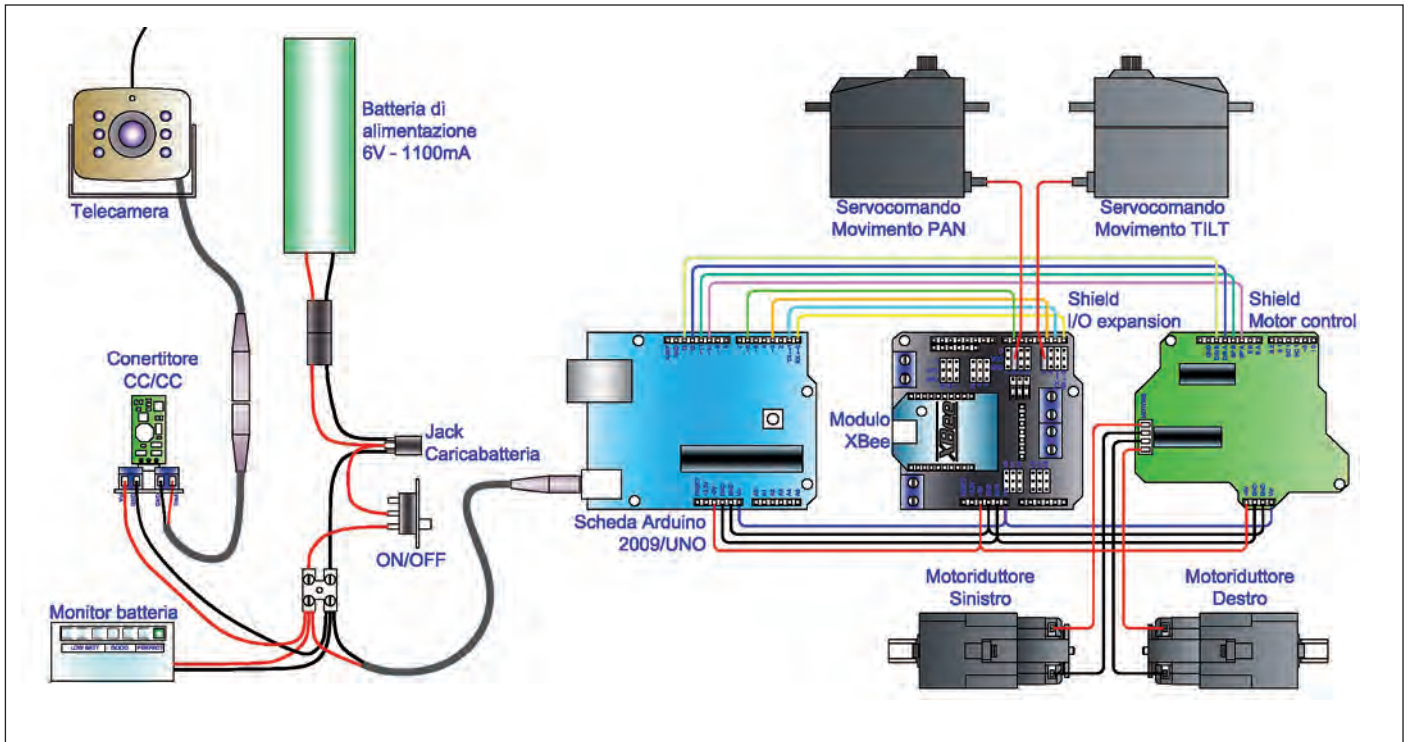


Figura 18: schema del robot ArduinoBOT.

TABELLA 7

Pan_neutro	Posizione iniziale del servo Pan
Tilt_neutro	Posizione iniziale del servo Tilt
AngoloPan	Posizione iniziale del servo
Angolo Tilt	Posizione iniziale del servo
DeltaPan	Gradi per il movimento del servo Pan
DeltaTilt	Gradi per il movimento del servo Tilt

Tabella 9: variabili di impostazione del dispositivo Pan & Tilt.

TABELLA 8

Movimento Robot		
	W	
A	X	D
	S	
Movimento Pan e Tilt		
	T	
F	B	H
	G	

Tabella 10: tasti per comando da PC.

```
// CTL3 9V for Motor (Optional)
// CTL4 GND
// CTL5 VCC
#define AT1_PIN 5 // CTL6 Attention (PAD1)
#define CLK_PIN 4 // CTL7 Clock
// CTL8 N/C
#define ACK_PIN 3 // CTL9 ACK
#define AT2_PIN 2 // CTL6 Attention (PAD2) per un eventuale secondo joystick
// CTL1 Data
#define CMD_PIN 3 // CTL2 Command

// CTL3 9V for Motor (Optional)
// CTL4 GND
// CTL5 VCC
#define AT1_PIN 4 // CTL6 Attention (PAD1)
#define CLK_PIN 5 // CTL7 Clock
// CTL8 N/C
#define ACK_PIN 6 // CTL9 ACK
#define AT2_PIN 7 // CTL6 Attention (PAD2)
Una volta fatta la modifica, salvate il file e assicuratevi che l'estensione sia sempre .c Ora potete aprire il programma Arduino e controllare se la libreria è stata installata correttamente.
```

### LibreriaPSX

Anche questa libreria permette ad Arduino di interfacciarsi con un controller digitale standard Playstation (o suoi cloni). Ciascun pulsante è definito nella libreria, i tasti riconosciuti sono: Left, Down, Right, Up, Strt, Slot, Squ, X, O, Tri, R1, L1, R2, L2.

### UTILIZZO DELLA LIBRERIA

Per l'installazione occorre scaricare la libreria da questo link: <http://arduino.cc/playground/uploads/Main/Psx.zip> e decomprimere il pacchetto





Figura 19: Xbee - USB Board.

nella directory Arduino \ librerie. Dentro questa cartella troverete i seguenti file:

- .DS\_Store;
- gpl.txt;
- keywords.txt;
- Psx.cpp;
- Psx.h.

È compresa una cartella denominata "Example" al cui interno troverete un programma "Psx.pde" per testare il controller. L'uso della biblioteca è semplice, richiedendo solo un'inizializzazione, un'assegnazione dei pin e un comando di lettura.

#### Comandi

- **Psx ()**: usato per inizializzare la libreria. Es.: Psx Psx;
- **setupPins (dataPin, cmndPin, attPin, clockPin, ritardo)**: definisce i pin di Arduino a cui è collegato il controller, così come il ritardo per cui il clock rimane alto o basso. Es.: Psx.setupPins (2, 3, 4, 5, 10).
- **read ()**: legge i dati pulsante dal controller Playstation. Es.: data = Psx.read (). I dati sono restituiti come numeri interi senza segno, e ogni bit rappresenta un pulsante specifico. Possono essere testati utilizzando un semplice if. Es.: If (dati & psxUp).

Ciascun pulsante è definito nella libreria, quindi non occorre usare codici esadecimali durante il test. I tasti riconosciuti sono riportati nella **tabella 4**.

## PROGRAMMI DI GESTIONE ROBOT

Per il controllo del robot (schema di **figura 18** e **tabella 5**) occorrono due programmi: uno per la stazione di controllo e uno per il robot. Il primo permette l'interfaciamento del controller che, secondo il tasto premuto, invia tramite il modulo Xbee un carattere verso il robot. Il secondo programma converte il carattere ricevuto in comandi che andranno ad attivare i motori del robot oppure i servo per muovere il dispositivo pan & tilt su cui è montata la telecamera. I programmi possono essere scaricati dal sito di *Fare Elettronica* e dovranno essere trasferiti sulla memoria del processore della scheda Arduino tramite l'apposito programma IDE che può essere scaricato gratuitamente al seguente link:

<http://arduino.cc/hu/Main/Software>.

NOTA: Per caricare i programmi è preferibile che sulle schede Arduino non siano montati gli Shield poiché i moduli Xbee comunicano con Arduino utilizzando i pin 0 e 1 (TX e RX) che sono gli stessi utilizzati per la programmazione la quale, a causa di interferenze, potrebbe non essere portata a compimento e fornire codice di errore.

#### Programma per la stazione di controllo

Il programma riportato nel **listato 1** utilizza per il funzionamento due librerie, una chiamata **psx.h** e una chiamata **Wpro-**

**gram.h** (oppure **Arduino.h** se utilizza il programma Arduino). Assicurarsi che queste siano presenti nella cartella delle librerie del programma. Nella **tabella 6** sono riportate le funzioni dei tasti e quale carattere viene inviato verso il robot.

#### Programma per il robot ArduinoBOT

Il programma riportato nel **listato 2** utilizza la libreria **servo.h**: per l'azionamento dei servo si noterà che il programma di controllo dei motori è diverso da quello utilizzato per le prove. In quest'ultimo la velocità dei motori è fissa, in quanto l'utilizzo della Libreria servo.h e in particolare la funzione **servo.attach()** inibisce la funzione **PWM** sui pin 9 e 10. Il pin 10 è utilizzato dalla scheda motore per l'abilitazione di uno dei motori e tale pin non può essere riconfigurato. Nel programma sono presenti alcune variabili che possono essere modificate per adattarsi al proprio montaggio, queste sono riportate nella **tabella 7**.


## CONCLUSIONI

Abbiamo così terminato la realizzazione del nostro robot ArduinoBOT: lascio al lettore la possibilità di migliorare o incrementare le capacità del robot. A tal proposito, vorrei far notare che una possibile modifica, può essere sostituire la base di comando con un modulo Xbee connesso direttamente al PC, utilizzando per esempio Xbee - USB Board prodotta da Droids (**figura 19**). Questa è stata progettata per consentire una connessione semplice e affidabile tra i moduli Xbee e il PC. La scheda supporta moduli Xbee e XbeePRO. Occorrerà realizzare un programma, magari utilizzando Visual Basic, il quale, in base ai tasti premuti sulla tastiera del PC, invii tramite la porta USB i relativi caratteri riportati nella **tabella 8**. Questa illustrata può essere una base per la realizzazione di robot più complessi. I giovani all'ultimo anno delle scuole superiori potranno certamente trarne idee per i prossimi esami di fine anno.

Potete vedere alcuni filmati preliminari del robot pubblicati su YouTube:

<http://youtu.be/wO20bwAAkK8>

<http://youtu.be/TgEljjUP9Ys>

<http://youtu.be/Ramy6iilLuhl> 

**Codice MIP 320107**