

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

Robot LittleBot – Stazione di controllo e programmi di gestione.

Posted By *Adriano Gandolfo* On 6 maggio 2014 @ 5:00 In Robotica & Droni | [6 Comments](#)



Nei precedenti articoli è stata affrontata la costruzione della scheda di controllo e l'assemblaggio del telaio del robot LittleBot, che è possibile comandare tramite l'utilizzo di uno smartphone su cui è stata caricata una speciale APP.

Questo articolo illustrerà come comandare il robot utilizzando la Scheda Arduino Esplora equipaggiandola con un modulo Bluetooth, dotandola di una batteria che la renderà autonoma e si farà riferimento a quanto appreso in alcuni articoli già pubblicati su Elettronica Open Source.

Per migliorare l'affidabilità della stazione di controllo, rispetto a quanto in precedenza pubblicato, il modulo di alimentazione è ora rappresentato da un'unica scheda che comprende il caricabatteria, l'alimentatore e un sistema per il controllo dello stato di carica.

Per quanto riguarda il programma di controllo, sarà possibile leggere sul display della scheda Arduino Esplora lo stato dei sensori installati sul robot.

Prima di iniziare l'assemblaggio della stazione di controllo, analizziamo i componenti necessari, che sono:

- Scheda Arduino Esplora;
- Display TFT;
- Batteria Li-ion;
- Scheda caricabatteria/alimentazione;
- Modulo Bluetooth;
- Cavi connettori per il collegamento delle varie parti.

Scheda Arduino Esplora

Nell'articolo **[Scopriamo la nuova scheda Arduino Esplora](#)** ^[1], potrete trovare la descrizione completa della Scheda Arduino Esplora.

Di seguito, ricordiamo le principali caratteristiche:

Processore **ATMEGA32U4** con bus a 8 bit prodotto dalla Atmel con architettura di tipo RISC, Velocità di clock 16 MHz, memoria Flash da 32 kB di cui 4 kB utilizzati dal bootloader, memoria EEPROM da 1kB, 20 porte Digital pin I / O, 12 Canali di ingresso analogici, 7 canali PWM, tensione di funzionamento 5V, 1 porta USB Full speed;

Un joystick analogico a due assi (X e Y) con pulsante centrale;

4 pulsanti disposti a rombo;

Un potenziometro lineare a cursore;

Un microfono per rilevare il rumore ambientale;

Un sensore di luce per la misurazione dell'intensità luminosa;

Un sensore per la misurazione della temperatura ambiente;

Un accelerometro triassiale (X, Y e Z);

Un buzzer per produrre suoni;

Un LED luminoso a LED tipo RGB con elementi Rosso Verde e Blu;

2 ingressi per collegare i moduli sensore della serie Tinkerkit;

2 uscite per collegare i moduli attuatori della serie Tinkerkit;

Un connettore per l'inserimento del display TFT a colori, dotato di uno slot per scheda SD.



Display LCD TFT per Arduino esplora

Per sfruttare le caratteristiche della scheda Arduino Esplora occorrerà dotarla del **display di tipo LCD TFT** [2].

Le caratteristiche principali del display sono: lo schermo che misura 40x44mm circa, la presenza di uno slot micro-SD nella parte posteriore (non utilizzata nella presente applicazione); è di tipo **TFT** (Thin Film Transistor), una tecnologia applicata ai display a cristalli liquidi (LCD) o OLED, identificati come display a matrice attiva. Il display TFT utilizzato è retroilluminato e misura 1,77" di diagonale, con risoluzione di **160 x 128 pixel**, per **262K colori**.

La gestione avviene tramite un'interfaccia SPI. I dati possono essere memorizzati all'interno della memoria RAM on-chip. La retroilluminazione del display è dimmerabile.



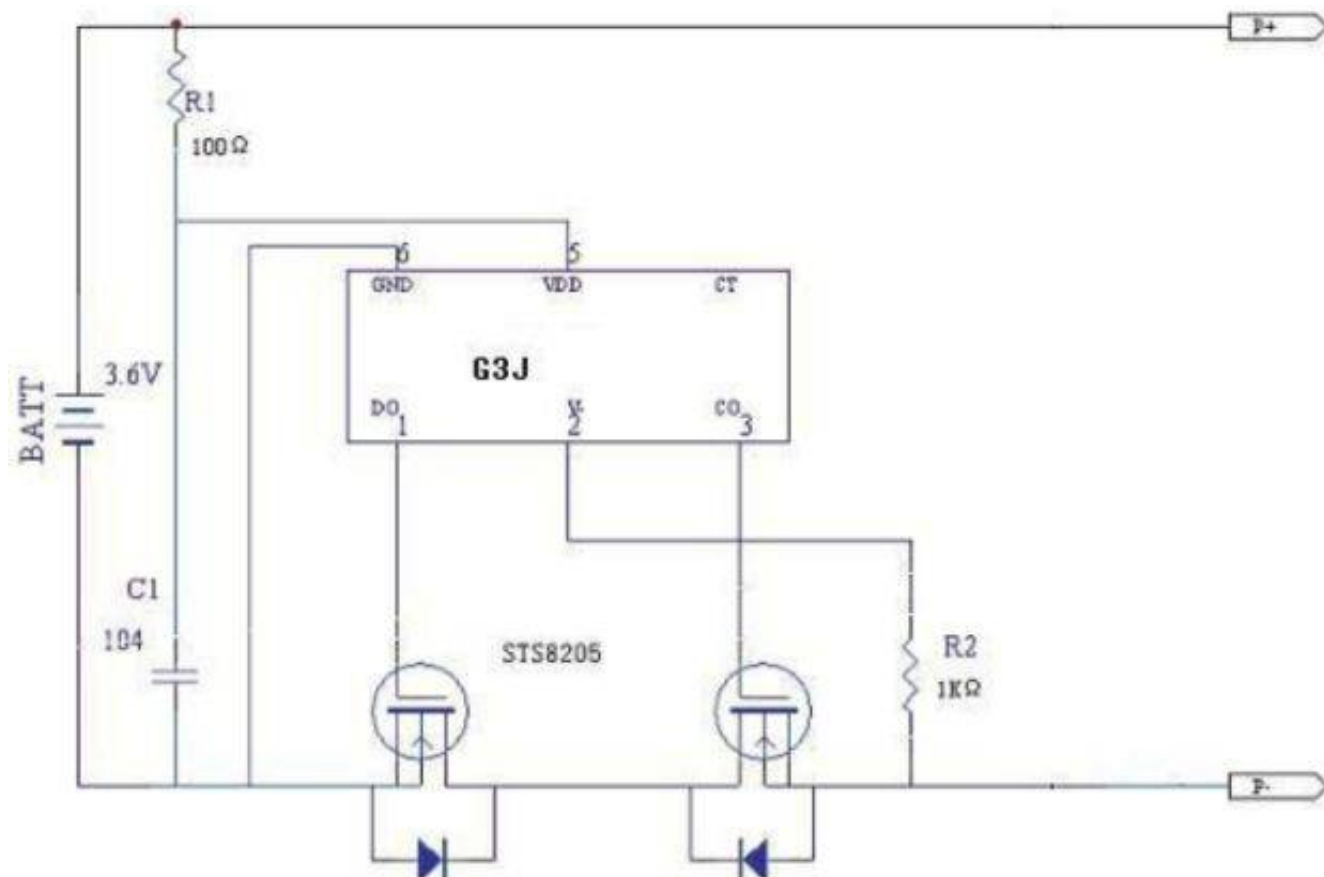
Batteria Li-ion

La batteria di tipo Li-ion potrà essere di **qualsunque modello** ^[3], rispetto a quella descritta nei precedenti articoli e anche nell'ottica di riduzione dei costi, la scelta questa volta è andata ad un modello commercializzato dalla **Seedstudio** ^[4], (**codice POW103C1P** ^[5]) si tratta di una batteria da **3,7V 2000 mA** (**datasheet** ^[6]).



La batteria ha le dimensioni di 68x48x5.7mm, un peso di circa 43g e dispone di una protezione integrata contro le sovratensioni, sovracorrente e minima tensione.

Il cuore del circuito di protezione è rappresentato dall'integrato tipo **S-8261** (**Datasheet** ^[7]) prodotto dalla **Seiko Instruments Inc** ^[8].



Schema del circuito di protezione della batteria

I cavi sono dotati di un connettore **JST 2-pin** analogo a quello presente sulla scheda di carica/alimentazione.

Scheda caricabatteria/alimentazione

Come anticipato, rispetto a quanto riportato nell'articolo **Rendiamo autonoma la scheda Arduino Esplora** ^[9], si è pensato di rendere più affidabile e professionale il sistema di alimentazione della scheda.

Infatti, in precedenza, erano utilizzati:

per la carica della batteria Li-Ion, un modulo prodotto dalla **SparkFun** ^[10] **codice PRT-10161** ^[11];

per elevare la tensione dal valore di 3,7V fornita dalla batteria a quello di 5V un convertitore DC/DC prodotto dalla Pololu con **codice 791** ^[12];

un interruttore e un fusibile.



Il tutto va poi assemblato e il circuito tarato.

Per semplificare il tutto e per aggiungere qualche funzione in più, si è scelta la scheda **Lipo Rider Pro** ^[13] prodotta dalla **Seed Studio** ^[14] con sede a Shenzhen in Cina, creatrice di molti prodotti elettronici.

Ecco alcune caratteristiche della scheda:

Corrente massima del carico **1A**;

Uscita stabile di **5V** dalla presa USB indipendentemente dalla fonte;

Algoritmi di ricarica batteria integrati in chip di controllo;

Possibilità di carica della batteria tramite USB o pannello solare;

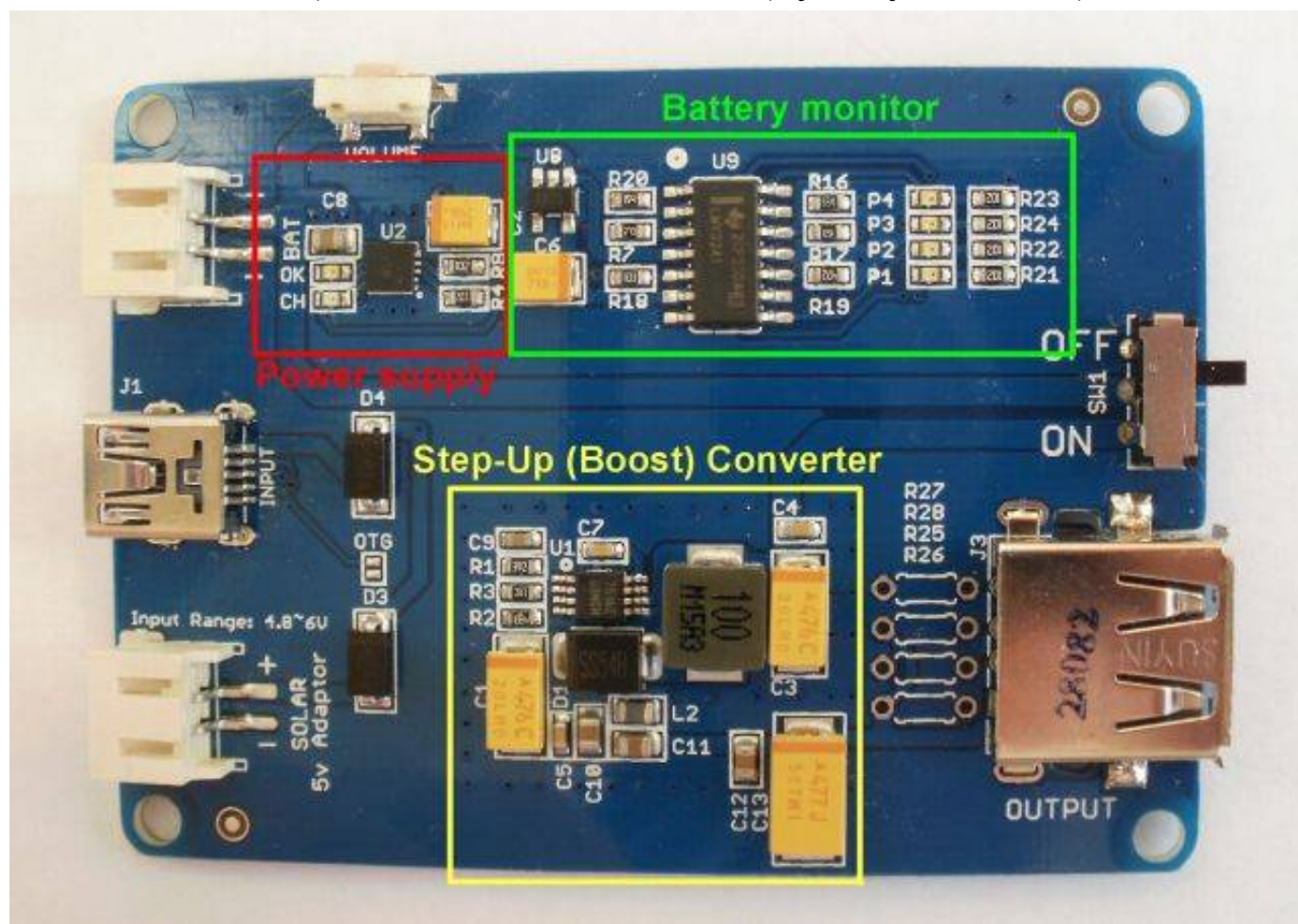
Tensione di alimentazione stabile sia attraverso o USB;

Indicazioni tramite LED della carica della batteria o del suo completamento;

Indicazione tramite 4 LED verdi dello stato di carica residua della batteria.

Alcuni parametri caratteristici del modulo:

Descrizione	Minimo	Normale	Massimo
V _{in} (pannello solare)	4.8V	5.0V	6.5V (10s)
I _{carica} (RISET = 3.9kΩ)	400mA	500mA	600mA
I _{carico}	0 mA		1000mA
V _{batteria} (R _x = 0Ω)	4.2V		
V _{source} USB	5.0V		
V _{uscita} USB	5.0V		



Come si può vedere nella foto, sulla scheda sono identificabili tre sezioni tra loro dipendenti:

Power Supply;

Step-Up (Boost) Converter;

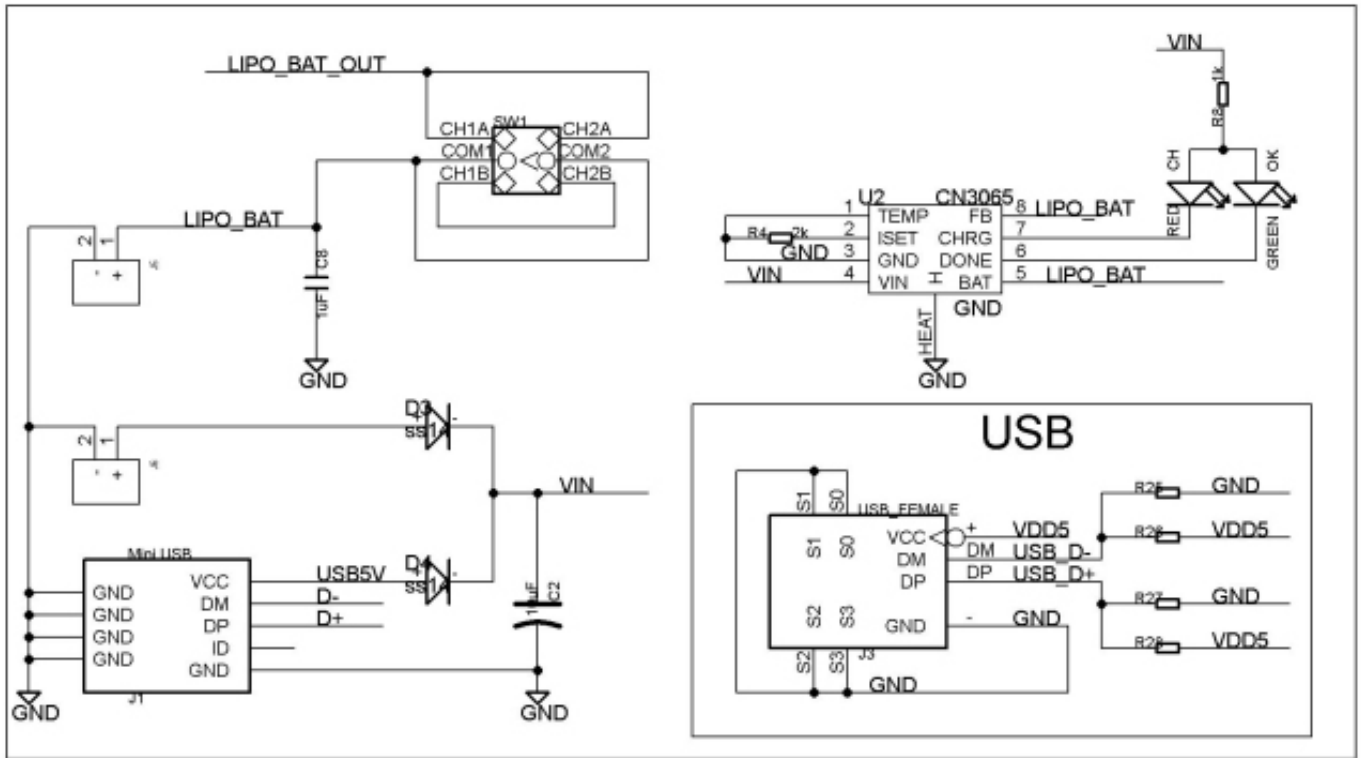
Battery Monitor.

Più alcuni connettori, un pulsante e un interruttore a levetta.

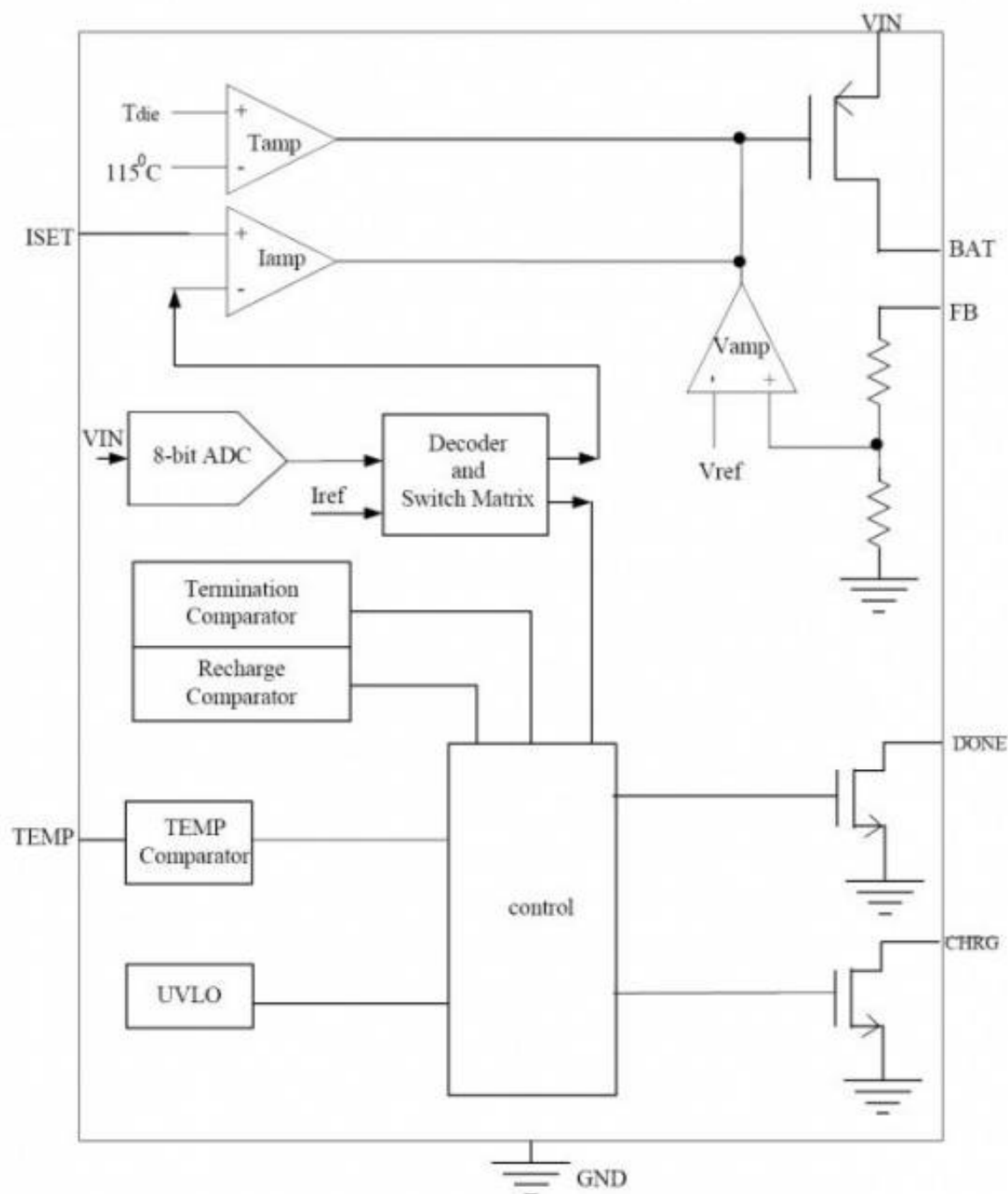
Power Supply

La sezione power supply compie la funzione di caricabatterie ed è realizzata utilizzando l'integrato **CN3065 (datasheet)** ^[15] prodotto dalla **Consonance** ^[16], in package DFN a 8 pin, che è un completo caricabatterie lineare a corrente costante/tensione costante per batterie ricaricabili a singola cella **Li-ion** e **Li Polymer**.

La tensione di carica della batteria connessa al connettore **BAT**, può essere fornita tramite la mini USB **J1** oppure tramite un pannello solare collegato al connettore **SOLAR**.



L'integrato **CN3065** contiene al suo interno un **MOSFET** di potenza on-chip ed elimina così la necessità di inserire all'esterno il current sense resistor e il diodo di blocco.



Schema a blocchi dell'integrato CN3065

Un **ADC** a 8 -bit on-chip è in grado di regolare la corrente di carica automaticamente in base alla capacità di uscita dell'alimentazione in ingresso.

Un feedback termico regola la corrente di carica per limitare la temperatura dell'integrato durante il funzionamento ad alta potenza o a temperatura ambiente elevata.

La tensione di regolazione è fissata internamente a 4.2V con 1 % di precisione.

La corrente di carica può essere programmata esternamente con una singola resistenza collegata al pin ISET.

Quando la tensione d'ingresso è rimossa, l'integrato entra automaticamente in modalità sleep a bassa potenza, riducendo la corrente di drain della batteria a meno di 3µA.

Altre caratteristiche includono il blocco di sottotensione, ricarica automatica, il rilevamento della temperatura della batteria e indicatore di carica/fine carica tramite i LED connessi ai pin

CHRG (LED rosso) e **DONE** (LED verde).

Di seguito il significato dello stato dei due LED:

Livello del pin CH (LED rosso)	livello pin OK (LED verde)	Significato
Livello basso (ON)	Livello Alto (OFF)	In carica
Livello alto (OFF)	Livello basso (ON)	Carica completa
Segnale a impulsi (flash)	Segnale di impulso (ON)	La batteria non è collegata
Livello alto (OFF)	Livello alto (OFF)	Due situazioni: - Tensione di ingresso inferiore tensione di gate - La tensione di ingresso inferiore alla tensione della batteria

Step-Up (Boost) Converter

Lo stadio di uscita è rappresentato dall'integrato **ISL97516** ([datasheet](#) ^[17]) con un package 8 Ld MSOP con un'altezza massima di 1,1 mm ed è prodotto dalla **INTERSIL** ^[18]. Si tratta di un regolatore di tensione ad alta frequenza, alta efficienza di tipo **step-up** che funziona a frequenza costante in modalità **PWM**.

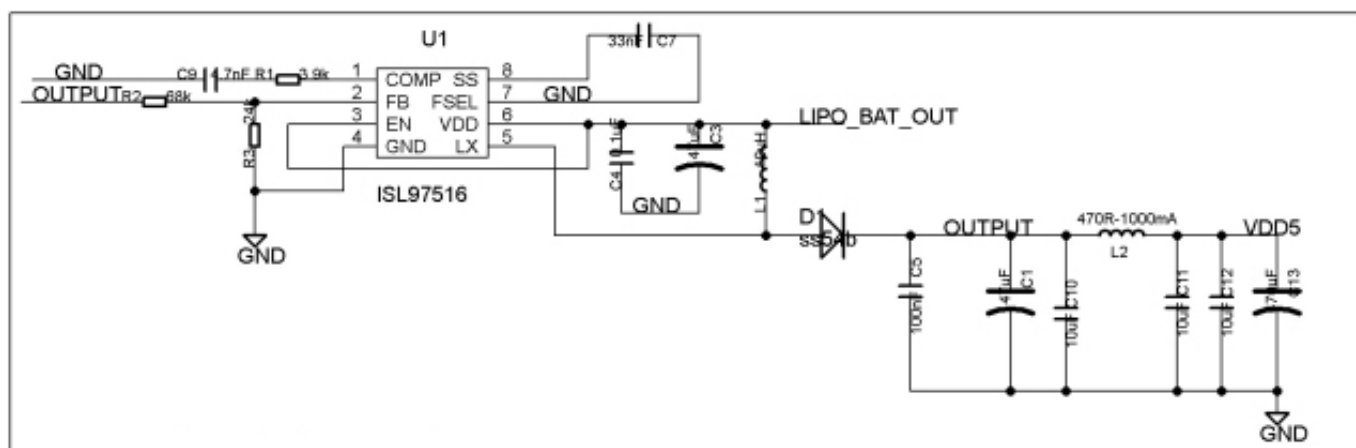
Con un'uscita di corrente tipica di 2.0A e 200mΩ il MOSFET, è in grado di fornire più del 90% di efficienza.

La frequenza selezionabile tra **600kHz** e **1,2 MHz** permette l'utilizzo d'induttori più piccoli e più veloci per la risposta ai transienti; nella nostra scheda, essendo il pin posto a livello basso, è prevista la frequenza di **600 kHz**.

Il valore della tensione d'uscita è impostato tramite il valore delle due resistenze poste sul pin FB, la formula è riportata nel datasheet.

$$V_{OUT} = V_{FB} \times \left(1 + \frac{R_1}{R_2} \right)$$

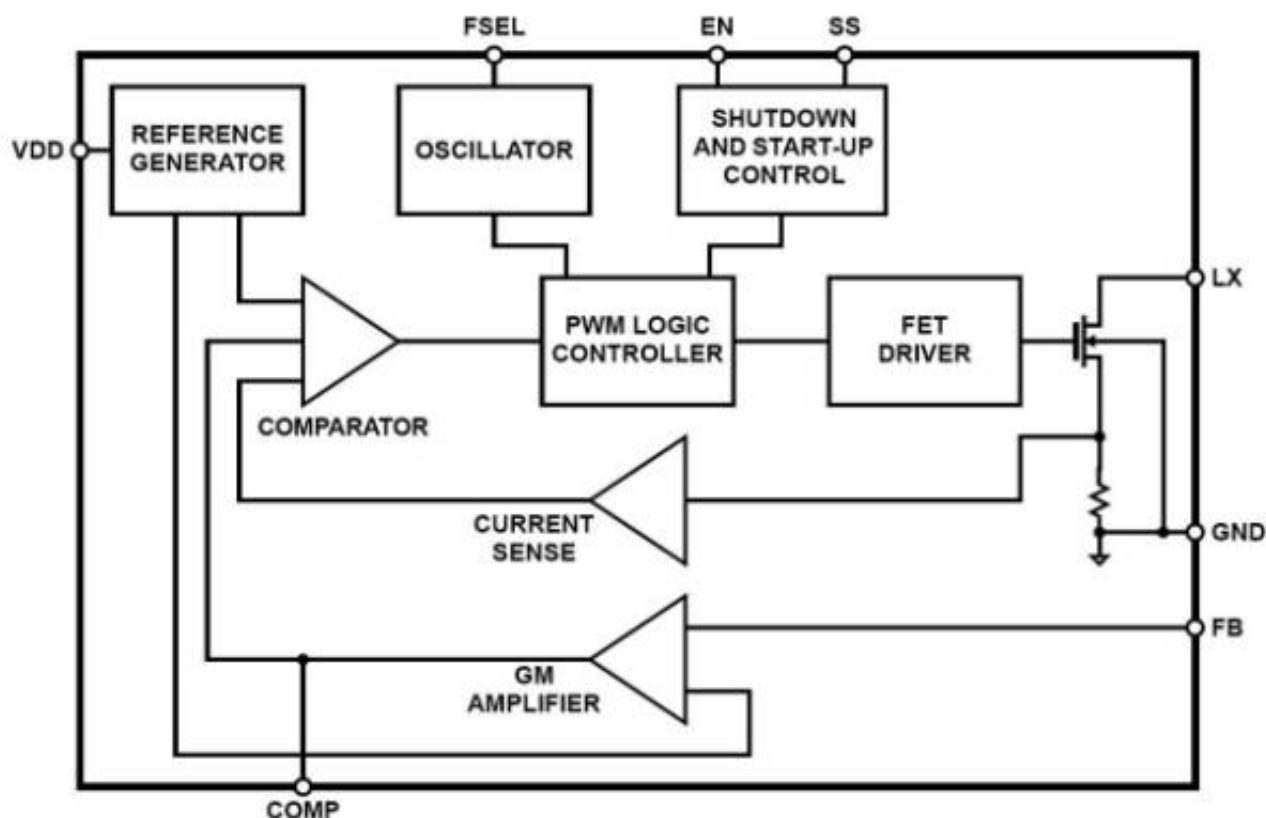
Dove V_{FB} vale 1.294V, considerato che nel circuito le due resistenze hanno il valore di 68K e 24K, si ottiene un valore teorico di **4.96V**.



Il pin di compensazione esterna **COMP** permette una maggiore flessibilità nella determinazione della compensazione in frequenza, permettendo l'uso di condensatori di uscita ceramici a basso ESR.

Quando disattivato, l'integrato assorbe una corrente inferiore a $1\mu\text{A}$, può operare con una tensione minima d'ingresso di 2.3V.

L'integrato può funzionare nella gamma di temperature comprese tra $-40\text{ }^\circ\text{C}$ a $+85\text{ }^\circ\text{C}$.



Schema a blocchi dell'integrato ISL97516

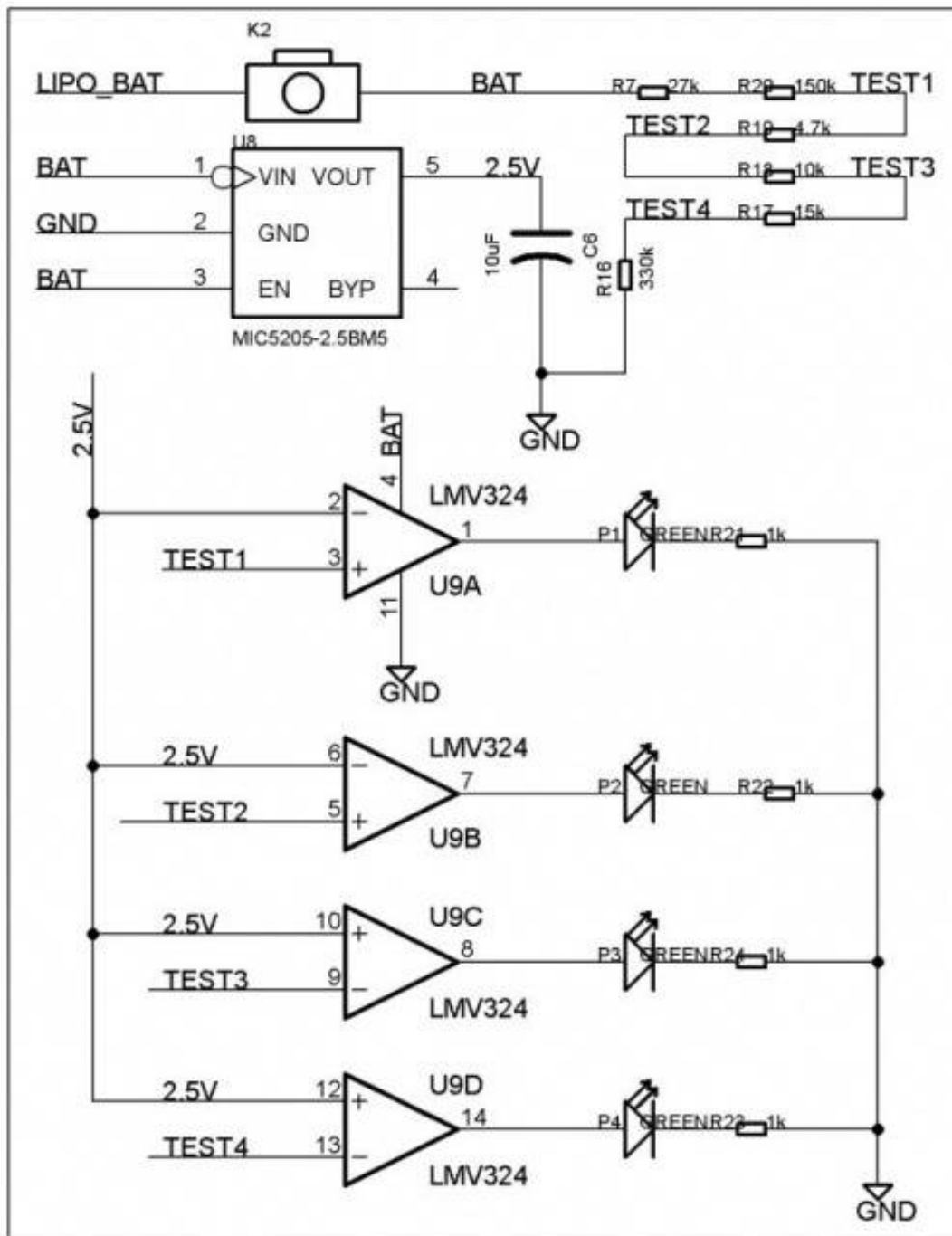
Battery Monitor

L'ultima sezione del modulo permette di verificare la carica residua della batteria.

Fanno parte del circuito due integrati: U8 è un **MIC 5205 -2.5** ([Datasheet](#) ^[19]) prodotto dalla **Micrel** e un regolatore lineare di tensione con uscita a bassissimo rumore e bassissima tensione di dropout.

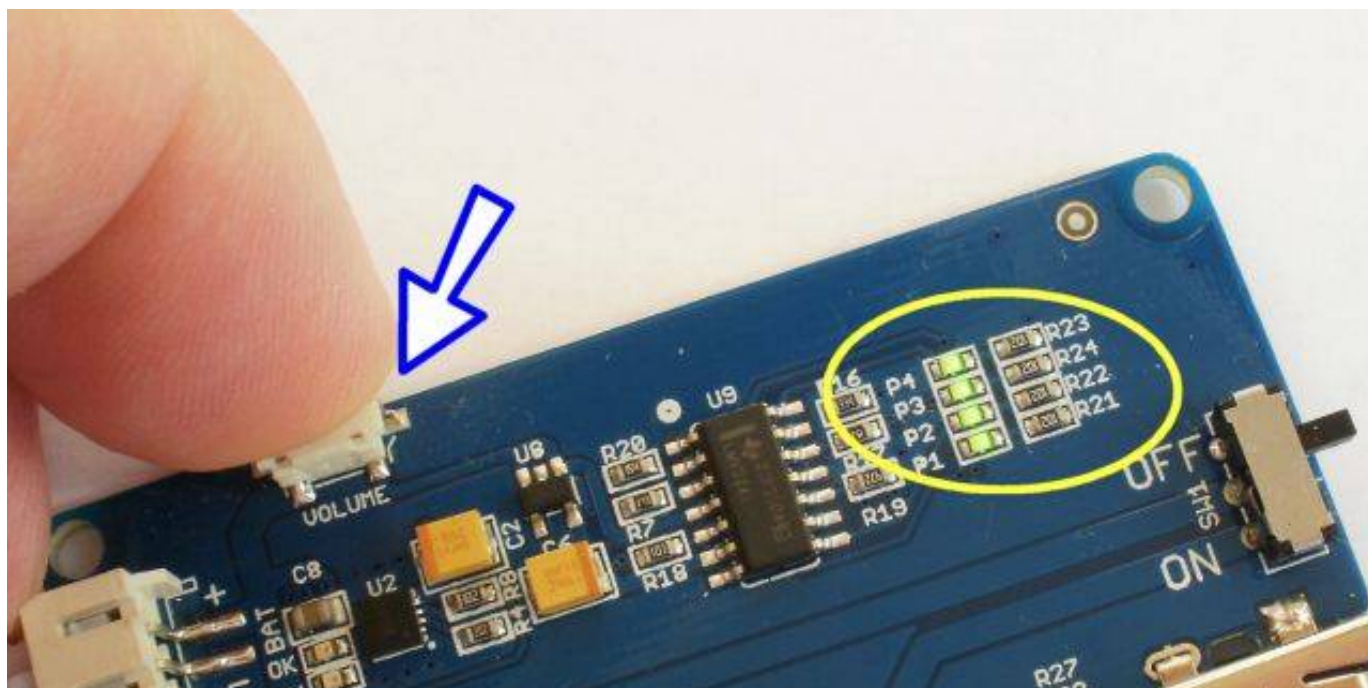
L'integrato alimentato tramite la batteria, fornisce in uscita una tensione di riferimento di 2.5V che viene poi ridotta tramite il partitore formato dalle resistenze R16, R17, R18, R19, R20, R7.

Queste resistenze, forniscono le tensioni di test utilizzate dai quattro amplificatori operazionali a bassa tensione con uscita Rail To Rail contenuto nell'integrato tipo **LMV 324** ([Datasheet](#) ^[20]) prodotto dalla **Texas Instruments** ^[21].



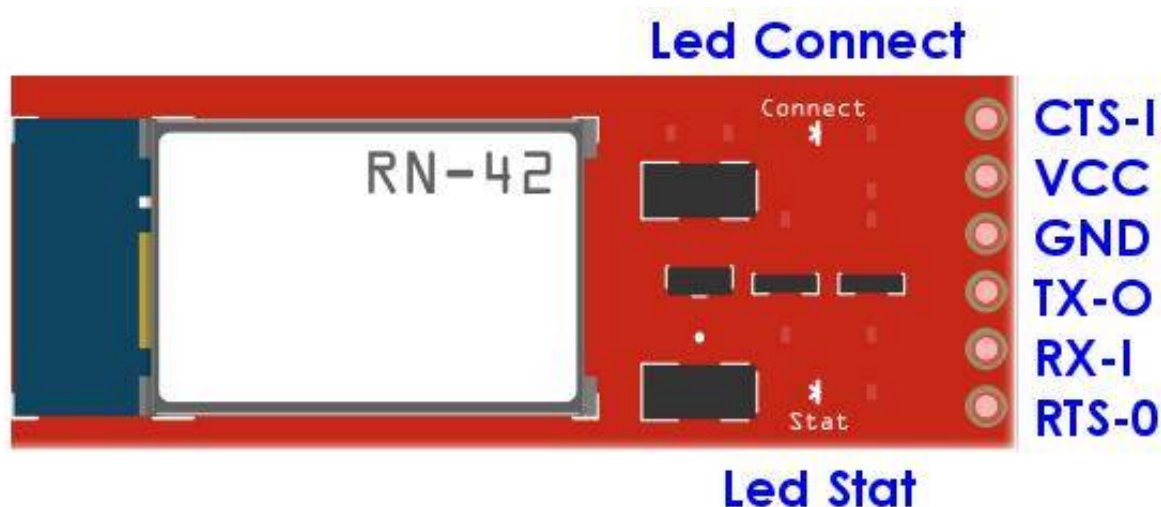
Quando si preme il tasto, che stranamente riporta la scritta **VOLUME**, si accendono i 4 LED verdi presenti sulla scheda con i valori di carica riportati nella tabella seguente.

Numero di LED illuminati	Quantità di energia residua
4	90 ~ 100%
3	60 ~ 90%
2	30 ~ 60%
1	10 ~ 30%
0	0 ~ 10%



Modulo Bluetooth BlueSMiRF Silver

Il modulo **BlueSMiRF Silver** è un modem wireless Bluetooth prodotto dalla SparkFun, basato sull'integrato **BlueSMiRF-42 RN** ([datasheet](#) ^[22]) prodotto dalla **ROVING Networks**, acquisita recentemente dalla **Microchip** ^[23].



Significato dei pin e dei LED del modulo Bluetooth

Ecco alcune specifiche tecniche:

Approvazione FCC Classe 2 Bluetooth Modem Radio;

Piccole dimensioni- 51.5x15.8x5.6mm;

Collegamento molto robusto sia in integrità che come distanza di trasmissione;

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

Connessione crittografata;

Frequenza: 2.4 ~ 2,524 GHz;

Tensione di funzionamento: 3.3V-6V;

Comunicazione seriale: 2400-115200 bps;

Temperatura di funzionamento: -40 ~ +70 C;

Antenna integrata.

Nota:

Se si cerca in rete il modulo Bluetooth presso la **SparkFun**, si può vedere che sono in vendita diverse tipologie di modulo :

BlueSMiRF Silver [24];

BlueSMiRF Gold [25];

Bluetooth Mate Gold [26];

Bluetooth Mate Silver [27];

BlueSMiRF HID [28].

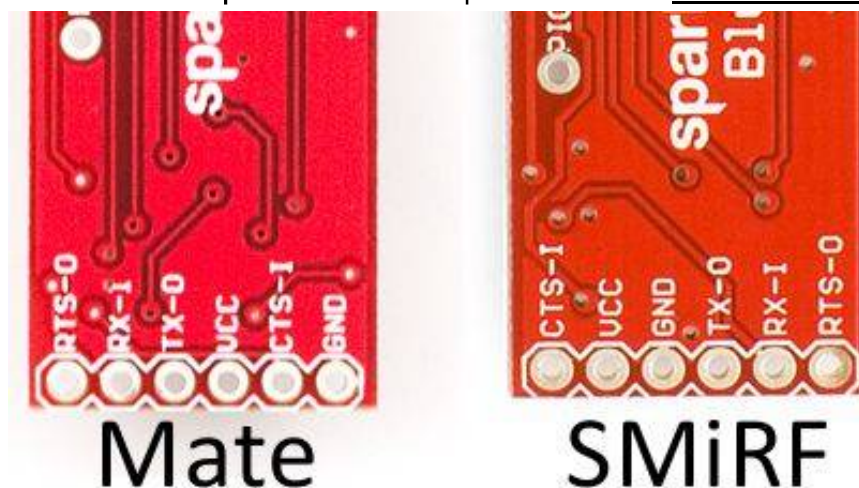
Quali sono le loro differenze:

I tipi **"Silver"** utilizzano un modulo tipo **RN-42 di classe 2**, che limita l'intervallo di trasmissione a 10 metri;

I tipi **"Gold"** utilizzano un modulo tipo **RN-41 di classe 1** che può comunicare fino a 100 metri, con un conseguente aumento dell'assorbimento da parte della batteria e anche il loro prezzo è maggiore;

Il tipo **HID** è essenzialmente la stessa scheda del BlueSMiRF Silver nel cui modulo **RN-42** è caricato un firmware **HID** o **"Human Interface Device"**, che è il protocollo di comunicazione utilizzato per periferiche come tastiere, mouse e joystick. Questo modulo può essere eventualmente riprogrammato per funzionare come un modulo BlueSMiRF Silver;

La differenza tra Mate e SMiRF è rappresentata dalla diversa posizione del pin-out dell'installazione a sei pin (riportata nella foto sottostante), in quanto il modulo **Mate** è stato creato specificatamente per le schede **Arduino Pro** [29] e **Arduino LilyPad** [30].



Ci sono due LED sul modem Bluetooth: uno rosso etichettato **"Stat"**, e uno verde con la scritta **"Connect"**. Questi aiutano per indicare lo stato del modulo

Il LED verde si accende quando si forma una connessione wireless.

Il LED "Stat" può indicare che il modulo è in uno dei tre stati, a seconda di quanto velocemente lampeggia:

Modalità	Velocità di lampeggio delLED Stat	Note
Configuration	10 al secondo	Il modulo è in modalità di configurazione.
Startup/Config Timer	2 al secondo	Il modulo non è in modalità di configurazione, ma il tempo di configurazione non è ancora esaurito.
Discoverable/Inquiring/Idle	1 al secondo	Il modulo non è in modalità di configurazione, ed il tempo di configurazione è esaurito.

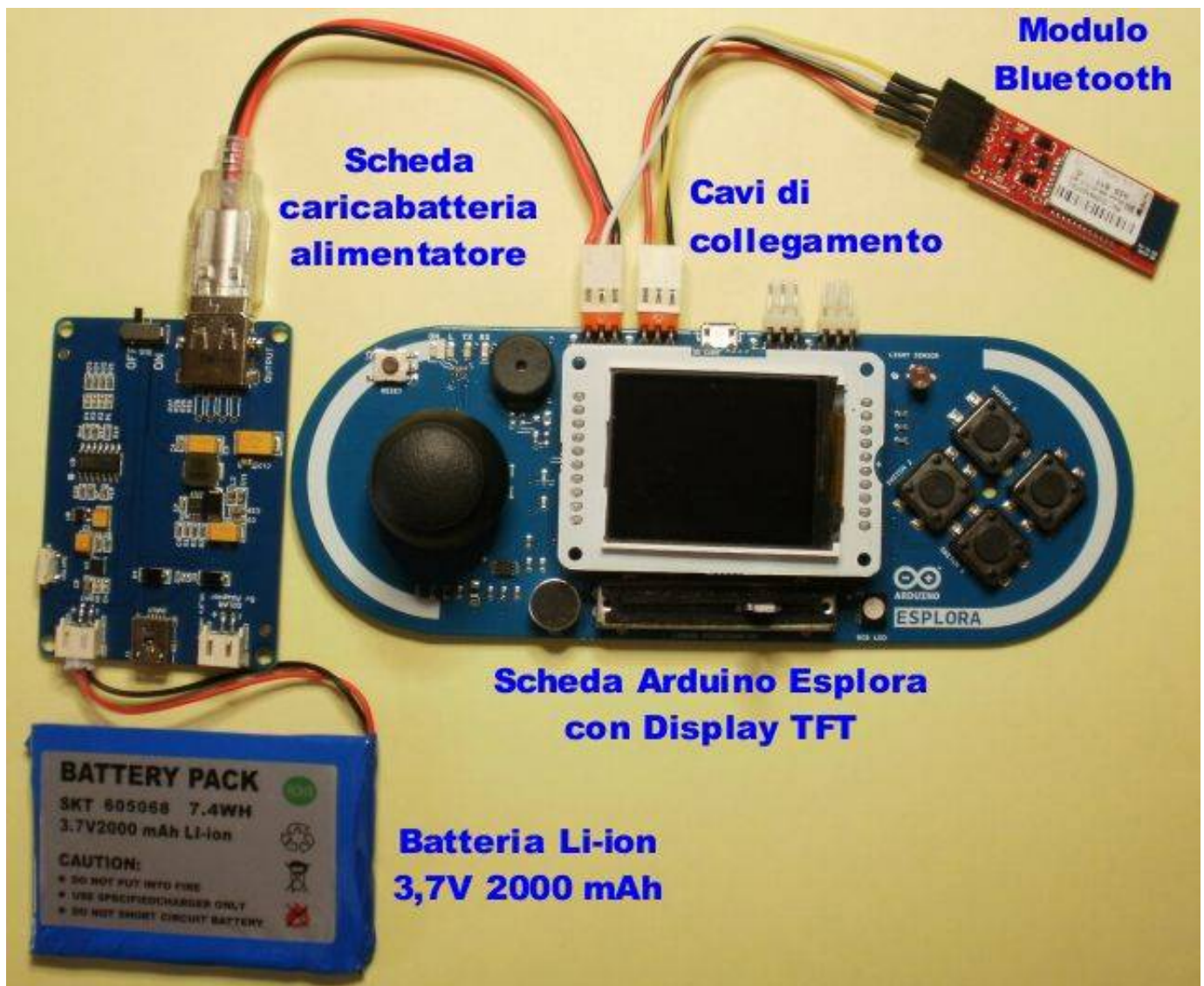
Montaggio della stazione di controllo

Dopo aver analizzato i vari componenti, vediamo come assemblare il tutto: per fare questo dovremo preparare il gruppo di cavi e connettori che permettono il collegamento delle varie parti che sono:

Scheda Arduino esplora con display TFT;

Scheda caricabatteria/alimentazione con Batteria Li-ion;

Modulo Bluetooth BlueSMiRF Silver.



Cavo di collegamento

Per la realizzazione del cavo di collegamento occorrono i seguenti componenti:

2 connettori tipo MOLEX - 22-01-1032- Femmina 3 pin (**Codice RS 687-7897** ^[31]);

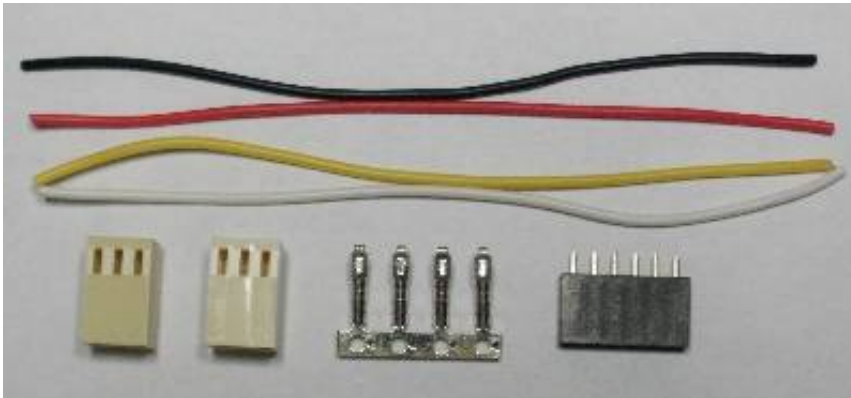
6 Terminali a crimpare, F, stagnato, 22-28awg (**Codice RS 720-6643** ^[32]);

1 Pin strip femmina 6x1;

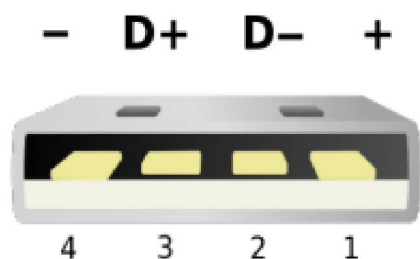
1 presa USB tipo A volante a saldare completa di guscio in plastica;

Cavi di collegamento di vario colore;

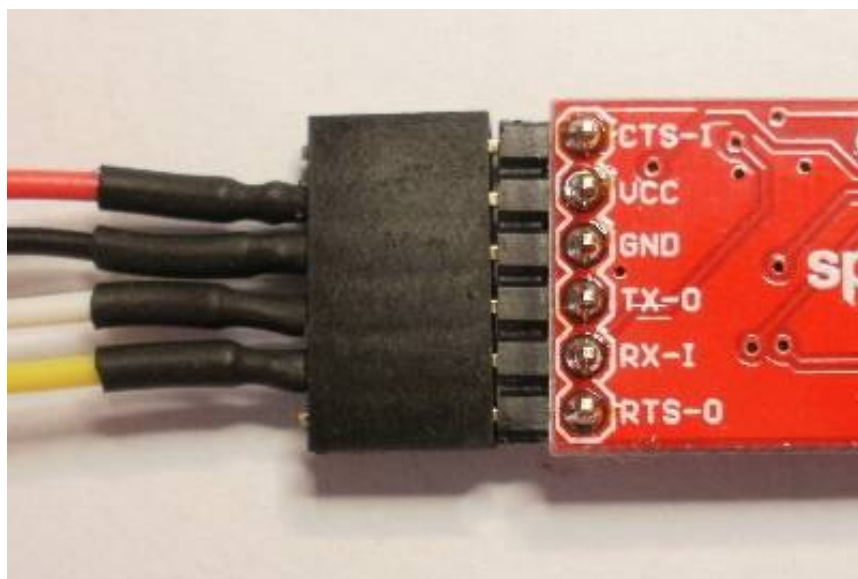
Del tubo termo restringente.



Con l'aiuto delle foto non dovrebbe essere difficile realizzare i collegamenti, per quanto riguarda la presa USB, il cavo rosso andrà saldato al pin 1 e il cavo nero al pin 4.



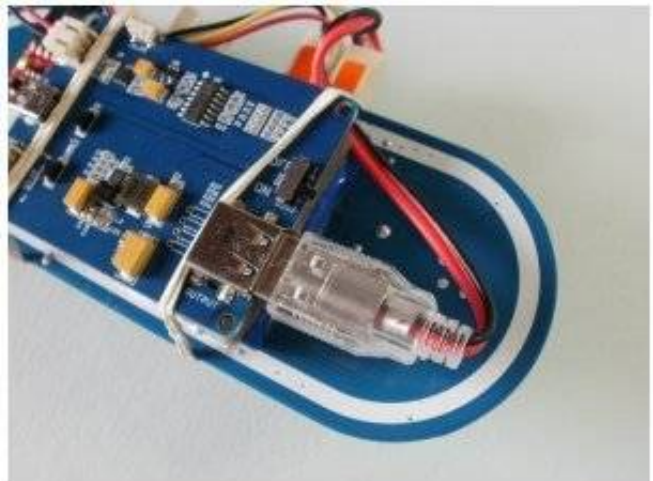
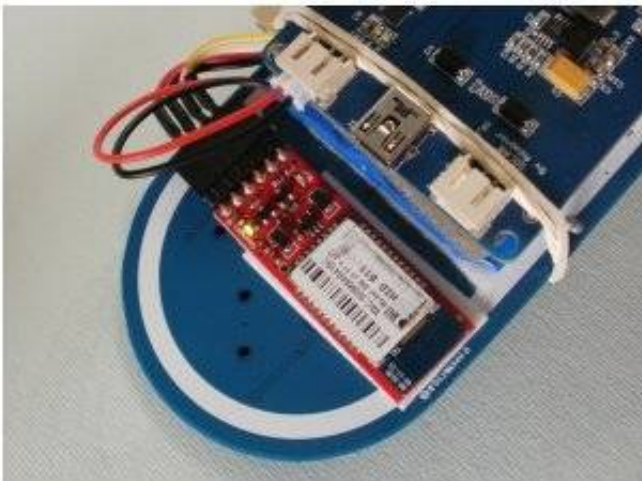
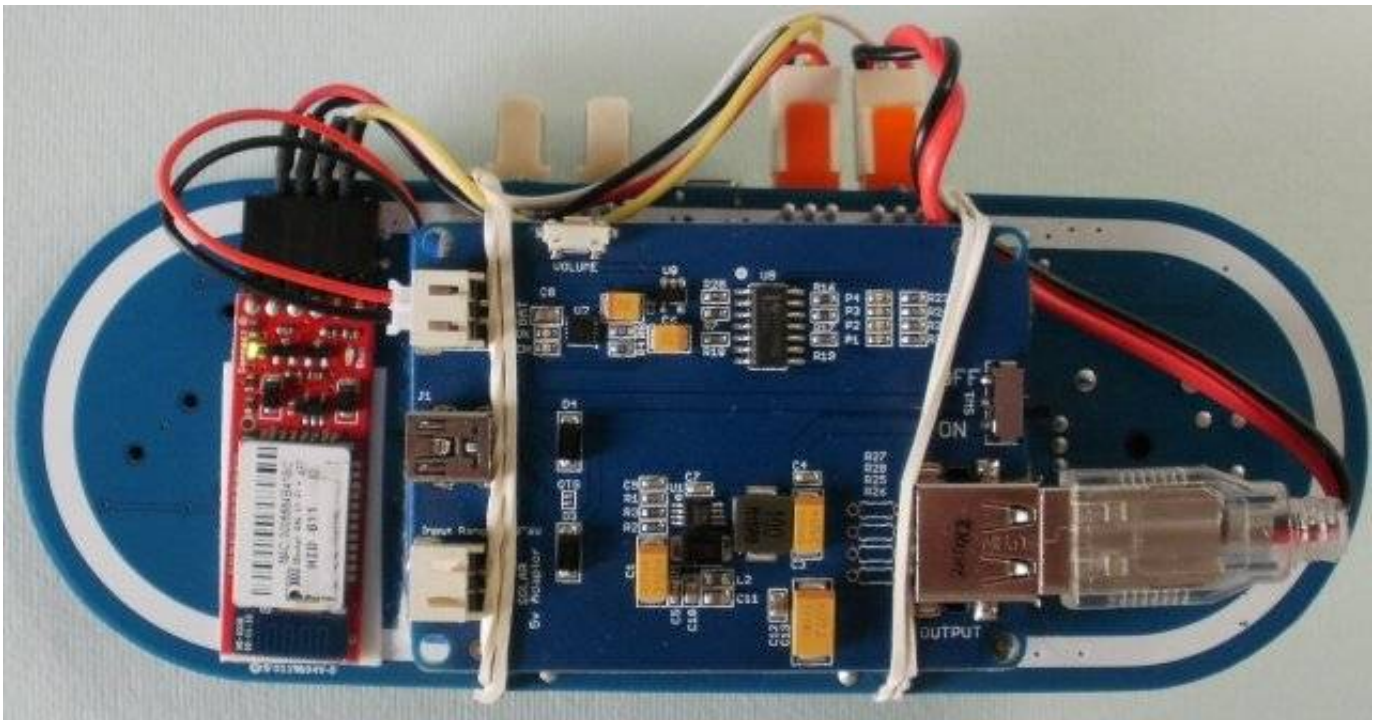
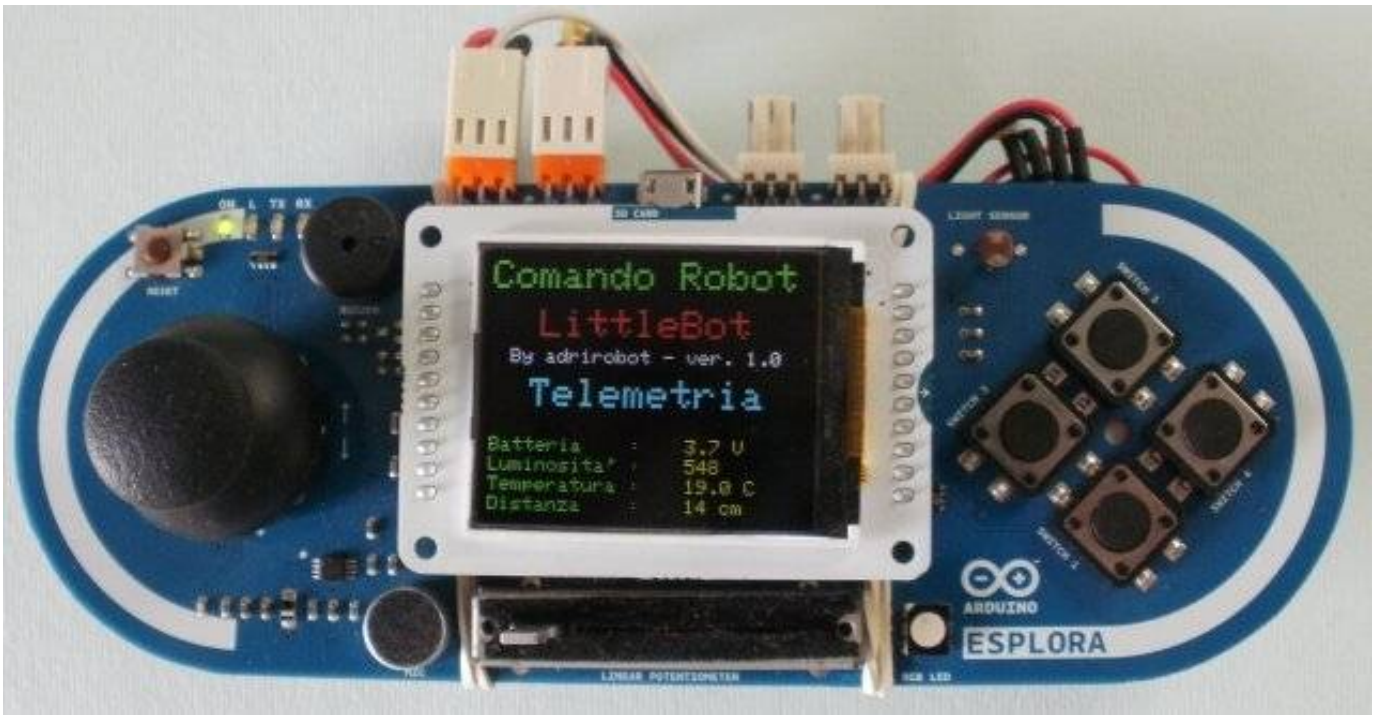
Per il collegamento del modulo Bluetooth, il **cavo giallo RX** andrà inserito nel pin centrale del connettore collegato a **OUT A**, mentre il **cavo bianco TX**, andrà inserito nel pin centrale del connettore collegato a **OUT B**.



Nelle foto sottostanti è visibile il risultato finale del montaggio.

Per il fissaggio alla scheda Arduino Esplora della batteria e della scheda Alimentazione/caricabatteria si sono utilizzati dei normali elastici interponendo tra le parti della gommapiuma.

Il modulo Bluetooth è fissato con del nastro biadesivo in schiuma di PE (per esempio tipo **TESA 4952** ^[33]).

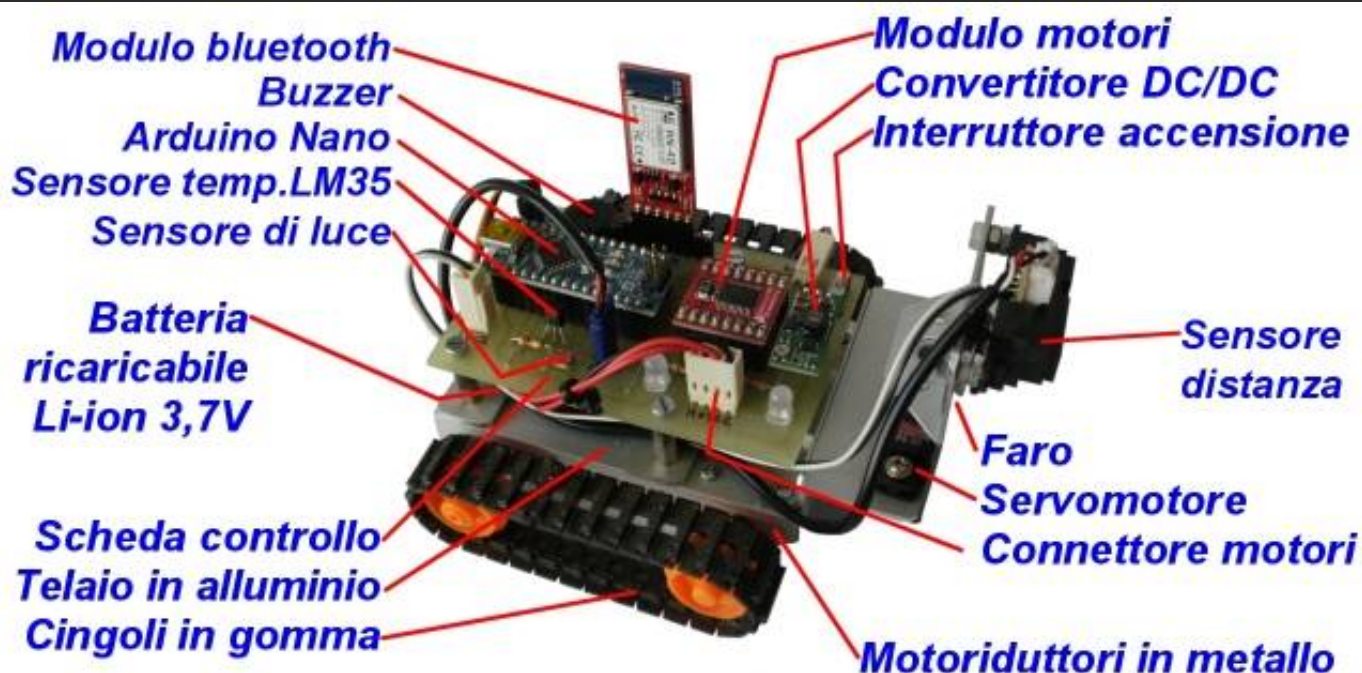


Il robot LittleBot

La costruzione del robot è stata oggetto di due precedenti articoli **Costruzione del robot LittleBot - Scheda di controllo** ^[34] e **Costruzione del robot LittleBot – Il telaio** ^[35].

Qui saranno riportate solo alcune delle caratteristiche, rimandando agli articoli gli eventuali approfondimenti.

Processore di comando	Arduino Nano
Modulo comando motori	Motor Driver 1A Dual TB6612FNG
Uscite	Segnalazione verso rotazione con LED bicolore
	Comando servo tipo HS81 della Hitec – coppia 2.6 kg/cm
	Comando faro a LED per illuminazione
	LED su scheda Arduino Nano
Sensori	Buzzer
	Temperatura LM35
	Luce Fotoresistenza
	Distanza IR GP2D120
	Controllo livello tensione tramite processore.
Alimentazione	Batteria Li-ion 3,7V 1840mA/h
	Convertitore DC/DC Pololu Adjustable Boost Regulator 2.5-9.5V
Trasmissione telemetria e controllo	Modulo Bluetooth tipo BlueSMiRF Silver della SparkFun - Classe 2, distanza di trasmissione massima di 10 metri;
Motoriduttori	In metallo rapporto di riduzione 30:1
Cingoli	In gomma prodotti dalla Tamiya
Misure	130x103x82
Peso	250 g



Problemi e soluzioni per la scrittura dei programmi.

Prima di passare alla descrizione dei programmi da installare, è opportuno soffermarsi su alcuni problemi incontrati durante la scrittura del programma per la stazione di controllo. Questi non si erano evidenziati nelle prime fasi poiché era previsto solamente un flusso di dati semplice, come l'invio di un solo carattere e con una direzione che era dalla Scheda Arduino Esplora verso la scheda di controllo del robot.

I problemi si sono verificati quando si è tentato di inviare dei dati, come la telemetria dei sensori, dal robot verso la scheda esplora.

Primo problema – Velocità di trasmissione

I moduli Bluetooth utilizzati, permettono una velocità massima di 115200 bps, che è tra l'altro il valore impostato di default.

Ora, se questo valore può andare bene per collegamenti semplici con la UART implementata in hardware sul chip, presente sulla scheda, i problemi si presentano quando si utilizza una seriale UART creata per via software, per cui un primo passo per evitare questo problema è stato ridurre a 9600 bps la velocità.

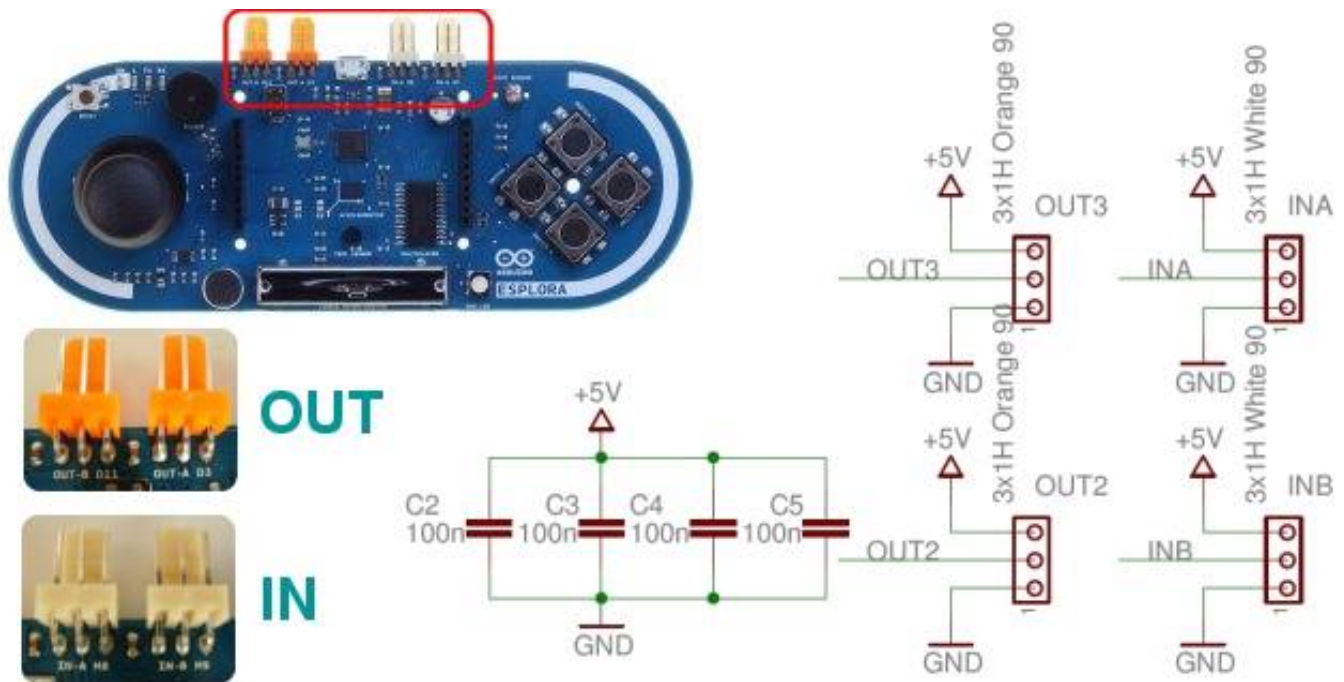
Secondo problema - Particolarità del processore della scheda Arduino Esplora

Come riportato nella descrizione della scheda, il processore della scheda Arduino Esplora è un **ATMEGA32U4** ([datasheet](#) ^[36]) che è lo stesso della scheda **Arduino Leonardo** ^[37], una delle sue particolarità è che ha già a bordo un'interfaccia USB.

Questo ha due implicazioni importanti: il costo più contenuto e che è in grado di operare in qualsiasi modalità USB.

Ciò significa che si possono creare delle **Human Interface Device (HID)**, come mouse, tastiere e stampanti.

Come abbiamo visto nell'articolo [Dotiamo l'Arduino Esplora dell'interfaccia Bluetooth](#) ^[38], per il collegamento del modulo con il processore, si sono utilizzati i due pin digitali disponibili connessi alle porte **D3** e **D11**.



Per la creazione della porta virtuale si sono utilizzati il pin 11 come RX e il pin 3 come TX.

```
SoftwareSerial mySerial(11, 3); // Definizione RX, TX per collegamento
```

Questa è la sola configurazione possibile in quanto, come leggiamo nella [pagina del sito Arduino](#) ^[39] che descrive i vari comandi della libreria **Software Serial**, sono riportate le seguenti limitazioni note.

Non tutti i pin della Leonardo supportano il cambio di interrupts, solo i seguenti possono essere usati come RX: 8, 9, 10, 11, 14 (MISO), 15 (SCK), 16 (MOSI).

Conversione dei valori ricevuti dal robot.

Altro problema è stato quello di inviare e decodificare i dati ricevuti dal robot.

La soluzione utilizzata ha previsto la creazione di una funzione che, dopo aver letto i vari sensori e convertito i dati a seconda del tipo, li invia alla seriale tramite dei comandi **Serial.println**; se il valore da inviare è decimale, questo viene prima moltiplicato per 10, in modo da inviare sempre un numero intero.

```
void telemetria()
{
  LED_on ();
  //Preparazione dei valori
  float vbatt = (analogRead(vbat) * ((readVcc() / 1000.0) / 1024));
  int batteria = vbatt * 10;

  int fotocellula = analogRead(foto); // lettura grezza dall'adc

  int reading = analogRead(temp);
  float voltage = (reading * 5.0) / 1024;
  int tempC = ((voltage - 0.5) * 100) * (-1);
```

```

int temperatura = tempC *10;

distance = Dist.getDistanceCentimeter();

//Invio valori verso stazione di comando
Serial.println (batteria);
Serial.println (fotocellula);
Serial.println (temperatura);
Serial.println (distance);
LED_off ();
}

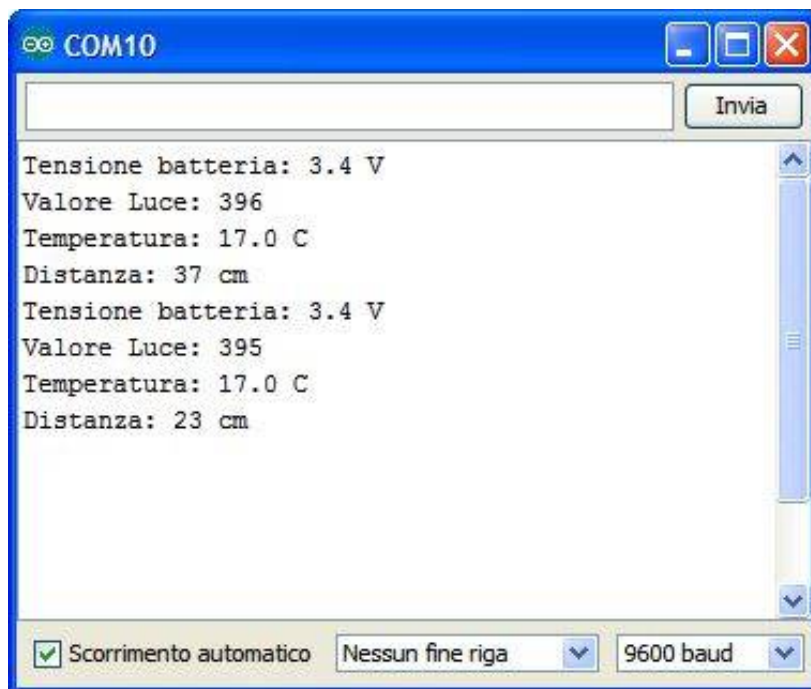
```

Sulla Stazione di controllo, il comando utilizzato per decodificare i valori ricevuti, utilizza la funzione **Serial.parseInt ()** che restituisce il primo numero intero dal buffer seriale, non considera il segno meno e i decimali e termina con il primo carattere non numerico. Per ricreare i valori decimali si è nuovamente diviso il valore per 10, successivamente si è utilizzato il comando **sprintf ()** che permette di creare una stringa con la giusta formattazione e con i valori calcolati, per esempio :

```
sprintf(tensione1, "%d.%d V", iBattVoltage, dBattVoltage);
```

Tensione1 è il nome della stringa in uscita.

"%d.%d V" è il formato della stringa formato dalla prima cifra (nel nostro caso il valore assunto da **iBattVoltage**), dal "punto", dalla seconda cifra (nel nostro caso il valore assunto da **dBattVoltage**).



Valori della telemetria visualizzati tramite il monitor seriale

Programma per stazione di controllo

Il programma per la stazione di controllo denominato **Robot Littlebot controller.ino**

[40], può essere riassunto nelle seguenti funzioni:

Caricamento delle librerie;

Definizione delle variabili e della porta seriale virtuale;

Assegnazione del carattere inviato al robot in base al tasto premuto;

Impostazione del modulo Bluetooth come Master con assegnazione del codice del modulo slave;

Impostazione del display;

Ciclo di lettura del joystick e dei tasti con invio al robot del codice corrispondente, nel caso il tasto premuto sia il numero 1 o il numero 3 viene riportato sul display il valore aggiornato della telemetria.

Il joystick ha la funzione di muovere il robot nelle quattro direzioni, mentre i tasti servono a:

1 - emissione suono;

3 - accensione/spengimento faro frontale;

2-4 rotazione servo frontale.



Come già spiegato in un precedente articolo (vedere [Dotiamo l'Arduino Esplora dell'interfaccia Bluetooth](#) ^[38]), il collegamento Bluetooth della scheda Arduino Esplora è effettuato impostando il modulo come Master.

Per fare questo, occorre inviare al modulo alcuni comandi che sono presenti nella sezione setup del programma.

Dopo aver impostato la velocità di trasmissione della seriale a **115200 bps** (che è quella di default del modulo), viene inviato il comando per entrare in command mode rappresentato da **print ("\$\$\$")** a cui deve seguire una breve pausa.

```
mySerial.begin(115200);
mySerial.print("$");
mySerial.print("$");
mySerial.print("$");
delay(100);
```

Quindi, s'invia il comando **println("SM,1")** che imposta il modulo come **Master**.

A questo punto indichiamo a quale modulo deve connettersi mediante il comando **println("C, <address>)** dove **<address>** deve essere sostituito dal codice MAC del vostro modulo slave, questo numero si trova scritto sull'esterno del modulo, ed è formato da 12 cifre.

```
mySerial.println("SM,1");
delay(100);
mySerial.println("C,000666464321");
delay(100);
```



Dopo una nuova pausa, impostiamo il valore della velocità di trasmissione a 9600 bps con il comando

```
mySerial.println("U,9600,N");
```

che cambia temporaneamente il baudrate a 9600, no parity, questo alla prossima riaccensione del modulo si riporterà al valore di 115200. Si trasmetterà quindi il comando

```
mySerial.println("---");
```

che permette l'uscita dal **Command Mode**, dopo un'ulteriore pausa con il comando

```
mySerial.begin(9600);
```

Si inizializza nuovamente la seriale di collegamento con il Bluetooth a 9600 bps.

Sotto le righe di codice corrispondenti:

```
mySerial.begin(115200);
mySerial.print("$");
mySerial.print("$");
mySerial.print("$");
delay(100);
mySerial.println("SM,1");
delay(100);
mySerial.println("C,000666644021");
```

```
delay(100);  
mySerial.println("U,9600,N"); // Cambia temporaneamente il baudrate a 9600,  
no parity  
mySerial.println("---"); //Uscita command mode  
delay (100);  
mySerial.begin(9600); // Reinizializza la seriale Bluetooth 9600  
delay (100);
```

Per il resto, il programma dovrebbe essere sufficientemente commentato da comprenderne le funzioni.

Nota sul funzionamento.

Una nota per chi utilizzerà il listato: nonostante tutte le prove e i vari tentativi di correzione, per un perfetto funzionamento del programma, occorre compiere **due reset consecutivi** della scheda Arduino Espora, tramite il tasto presente sulla scheda stessa, affinché funzioni correttamente.

Programma per il Robot

Il programma per il robot denominato **Robot_Littlebot_robot.ino** ^[41] è una versione più completa di quello già utilizzato per testare la scheda.

Oltre alle possibilità che erano disponibili:

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

Si aggiungono ora le seguenti funzioni:

- Servo frontale a destra;
- Servo frontale a sinistra;
- Servo frontale al centro;
- Robot avanti;
- Robot indietro;
- Robot a SX;
- Robot a DX;
- Robot fermo;
- Trasmissione telemetria.

I comandi disponibili sono accessibili inviando i seguenti caratteri alfanumerici:

b - Legge tensione della batteria

- t - Legge temperatura
- d - Legge distanza
- l - Legge luminosità
- s - Emette un "BIP"
- k - Accende il faro (se acceso lo spegne)
- n - Accende LED
- m - Spegne LED
- p - Servo a DX
- i - Servo a SX
- o - Servo al centro;
- 8 - Robot avanti;
- 2 - Robot indietro;
- 4 - Robot a SX;
- 6 - Robot a DX;
- 0 - Robot fermo;
- q - Trasmette telemetria;

Se s'invia il comando "h", si avrà l'elenco delle opzioni, la cui visione si potrà avere collegando il robot tramite il cavo al PC.



Alcune particolarità del programma

Il programma utilizza una libreria denominata **DistanceGP2Y0A21YK.h** scritta da Jeroen

Doggen che può essere scaricata al **seguente link** [42], questo semplifica l'utilizzo del sensore.

E' possibile variare la velocità del robot modificando il valore della variabile **vel** che, attualmente, è impostata a 100.

```
#define vel 100 //velocità dei motori
```

il valore può variare teoricamente tra 0 e 255, ma se è troppo basso, i motori non ruoteranno , perchè sono controllati in **PWM (Pulse Width Modulation)**.

Con questo sistema, la velocità del motore non viene regolata variando la tensione ma variando il tempo durante il quale, l'intera tensione di alimentazione viene applicata ai terminali del motore.

Per l'impostazione del modulo Bluetooth (che in questo caso sarà impostato come **SLAVE**), si segue una procedura simile a quella vista per la stazione di controllo, l'unica variazione è che il comando in questo caso sarà:

```
Serial.println("SM,0");
```

che imposta appunto il modulo come Slave.

Ecco in dettaglio la parte di programma:

```
Serial.begin(115200); //imposta la porta di comunicazione con il modulo
Serial.print("$");
Serial.print("$");
Serial.print("$");
delay(100);
Serial.println("SM,0"); // Imposta il modulo come slave
delay(100);
Serial.println("U,9600,N") // Cambia temporaneamente il baudrate
                          // a 9600, no parity
delay(100);
Serial.println("---"); //Uscita command mode
delay(100);
Serial.begin(9600); // Reinizializza la seriale Bluetooth 9600
```

Filmato illustrativo

Il filmato mostra sia la stazione di comando come realizzata sia il comando del robot.

Robot LittleBot - Collaudo finale



Conclusioni

Siamo così arrivati al termine di questa serie di articoli che hanno portato alla realizzazione del **Robot LittleBot** e della sua stazione di comando. Oltre all'articolo attuale, i precedenti:

[Presentazione del robot LittleBot](#) ^[43];

[Costruzione del robot LittleBot - Scheda di controllo](#) ^[34];

[Costruzione del robot LittleBot – Il telaio](#) ^[35].

Per approfondire la conoscenza della stazione di controllo basata sulla **scheda Arduino Esplora** si potrà fare riferimento ai precedenti articoli:

[Scopriamo la nuova scheda Arduino Esplora](#) ^[1];

[Programmiamo la scheda Arduino Esplora](#) ^[44];

[Rendiamo autonoma la scheda Arduino Esplora](#) ^[9];

[Dotiamo l'Arduino Esplora dell'interfaccia Bluetooth.](#) ^[38]

I programmi presentati possono essere un'ottima base di partenza per integrarli e possibilmente migliorarli.

In questa fase il robot è completamente pilotabile, ma nulla vieta di implementare delle funzioni di navigazione automatica, utilizzando il sensore frontale in accoppiamento al servo che potrebbe scandagliare un terreno frontalmente al robot e in base ai valori letti, muoversi opportunamente.

Altra modifica potrebbe essere quella che il robot, in base al valore di luminosità del sensore, accenda o meno in modo automatico il faro.

Oppure, al raggiungimento di un valore minimo di tensione della batteria, il robot emetta dei bip e si fermi automaticamente.

Anche sulla stazione ricevente si possono apportare delle modifiche: una potrebbe essere quella di utilizzare l'accelerometro in sostituzione del Joystick, per cui il movimento del robot sarebbe dato dall'inclinazione della scheda.

Utilizzando in modo diverso i tasti, si potrebbero utilizzare più funzioni del robot. I dati rappresentati sul display potrebbero comprendere anche quelli dei sensori presenti sulla scheda stessa.

In base al valore di luminosità rilevata dal robot, il LED RGB presente sulla scheda, variare il suo colore.

Si potrebbe poi salvare i dati rilevati dal robot sulla micro SD presente sul retro del display utilizzata come data logging e molto altro ancora.

Mi sembra che per chi ha voglia di cimentarsi nella programmazione, ci siano molte possibilità.

Sarebbe interessante poter vedere i programmi di altri costruttori del robot, magari inserendoli in allegato tra i messaggi.

Ci sono state molte richieste per un kit, al momento ne stiamo studiando la fattibilità e se le richieste saranno sufficienti, non è detto che non possa concretizzarsi.

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

URL to article: <https://it.emcelettronica.com/robot-littlebot-stazione-di-controllo-programmi-di-gestione>

URLs in this post:

- [1] **Scopriamo la nuova scheda Arduino Esplora:** <https://it.emcelettronica.com/scopriamo-nuova-scheda-arduino-esplora>
- [2] **display di tipo LCD TFT:** <http://arduino.cc/en/Main/GTFT>
- [3] **qualunque modello:** <https://it.emcelettronica.com/batterie-al-litio-scegliamo-le-migliori>
- [4] **Seedstudio:** <http://www.seeedstudio.com/>
- [5] **codice POW103C1P:** <http://www.seeedstudio.com/depot/Lithium-Ion-polymer-Battery-2A-p-603.html?cPath=73>
- [6] **datasheet:** http://www.adrirobot.it/datasheet/vari/pdf/605068P_2000mAh.pdf
- [7] **Datasheet:** <http://www.adrirobot.it/datasheet/speciali/pdf/S-8261%20Protezione%20batteria.pdf>
- [8] **Seiko Instruments Inc:** <http://www.sii.co.jp/en/>
- [9] **Rendiamo autonoma la scheda Arduino Esplora:** <https://it.emcelettronica.com/rendiamo-autonoma-scheda-arduino-esplora>
- [10] **SparkFun:** <https://www.sparkfun.com/>

- [11] codice PRT-10161: <https://www.sparkfun.com/products/10161>
- [12] codice 791: <http://www.pololu.com/product/791>
- [13] **Lipo Rider Pro** : <http://www.seedstudio.com/depot/LiPo-Rider-Pro-p-992.html>
- [14] Seeed Studio: <http://www.seedstudio.com>
- [15] **datasheet**): http://www.adrirobot.it/datasheet/speciali/pdf/DSE-CN3065_Lithium_Ion_Battery_Charger_for_Solar-Powered_Systems.pdf
- [16] **Consonance**: <http://www.consonance-elec.com/index-E.html>
- [17] **datasheet**:
http://www.adrirobot.it/datasheet/integrati/pdf/ISL97516_Step-up_Regulator.pdf
- [18] **INTERSIL**: <http://www.intersil.com/en.html>
- [19] **Datasheet**:
<http://www.adrirobot.it/datasheet/integrati/pdf/mic5205%20Regolatore%20di%20tensione.pdf>
- [20] **Datasheet**: <http://www.adrirobot.it/datasheet/integrati/pdf/LMV358.pdf>
- [21] **Texas Instruments**: <http://www.ti.com/>
- [22] **datasheet**: <http://www.adrirobot.it/datasheet/speciali/pdf/Bluetooth-RN-42-DS.pdf>
- [23] **Microchip**: <https://www.microchip.com/>
- [24] **BlueSMiRF Silver**: <https://www.sparkfun.com/products/12577>
- [25] **BlueSMiRF Gold**: <https://www.sparkfun.com/products/12582>
- [26] **Bluetooth Mate Gold**: <https://www.sparkfun.com/products/12580>
- [27] **Bluetooth Mate Silver**: <https://www.sparkfun.com/products/12576>
- [28] **BlueSMiRF HID**: <https://www.sparkfun.com/products/10938>
- [29] **Arduino Pro**: <http://arduino.cc/en/Main/ArduinoBoardPro>
- [30] **Arduino LilyPad**: <http://arduino.cc/en/Main/arduinoBoardLilyPad>
- [31] **Codice RS 687-7897**: <http://it.rs-online.com/web/p/custodie-per-connettori-da-pcb/6877897/>
- [32] **Codice RS 720-6643**: <http://it.rs-online.com/web/p/contatti-per-connettori-da-pcb/7206643/>
- [33] **TESA 4952**: http://www.tesaitalia.it/industry/products/tesa_4952.html
- [34] **Costruzione del robot LittleBot - Scheda di controllo**:
<https://it.emcelettronica.com/costruzione-del-robot-littlebot-scheda-di-controllo>
- [35] **Costruzione del robot LittleBot - Il telaio**:
<https://it.emcelettronica.com/costruzione-del-robot-littlebot-%E2%80%93-telaio>
- [36] **datasheet**:
<http://www.adrirobot.it/datasheet/processori/pdf/ATMega32U4.pdf>
- [37] **Arduino Leonardo**: <http://arduino.cc/en/Main/arduinoBoardLeonardo>
- [38] **Dotiamo l'Arduino Esplora dell'interfaccia Bluetooth**:
<https://it.emcelettronica.com/dotiamo-l%E2%80%99arduino-esplora-dell%E2%80%99interfaccia-bluetooth>
- [39] **pagina del sito Arduino**: <http://arduino.cc/en/Reference/SoftwareSerial>
- [40] **Robot_Littlebot_controller.ino**:
https://it.emcelettronica.com/files/Robot_Littlebot_controller.zip
- [41] **Robot_Littlebot_robot.ino**:
https://it.emcelettronica.com/files/Robot_Littlebot_robot.zip
- [42] **seguente link**: <https://github.com/jeroendoggen/arduino-distance-sensor-library>
- [43] **Presentazione del robot LittleBot**: <https://it.emcelettronica.com/presentazione-del-robot-littlebot>
- [44] **Programmiamo la scheda Arduino Esplora**:

<https://it.emcelettronica.com/programmiamo-scheda-arduino-esplora>

Copyright © 2017 Elettronica Open Source. All rights reserved.