

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

Rendiamo autonoma la scheda Arduino Esplora.

Posted By *Adriano Gandolfo* On 21 ottobre 2013 @ 5:00 In [Arduino](#) | [5 Comments](#)



Dopo aver presentato la scheda Arduino Esplora, ^[1] aver visto come istallarla e programmarla, ^[2] è arrivato il momento di utilizzare la nostra scheda. Vedremo in quest'articolo come alimentare la scheda con una batteria permettendoci di scollegarla dal PC e di renderla autonoma.

La scheda Arduino Esplora, per il suo funzionamento deve essere alimentata tramite la porta USB.

Una batteria ricaricabile di tipo Li-Ion

Un modulo caricabatteria

Un modulo convertitore DC/DC del tipo Step Up

Un deviatore a slitta unipolare

Un fusibile resettabile da 500 mA

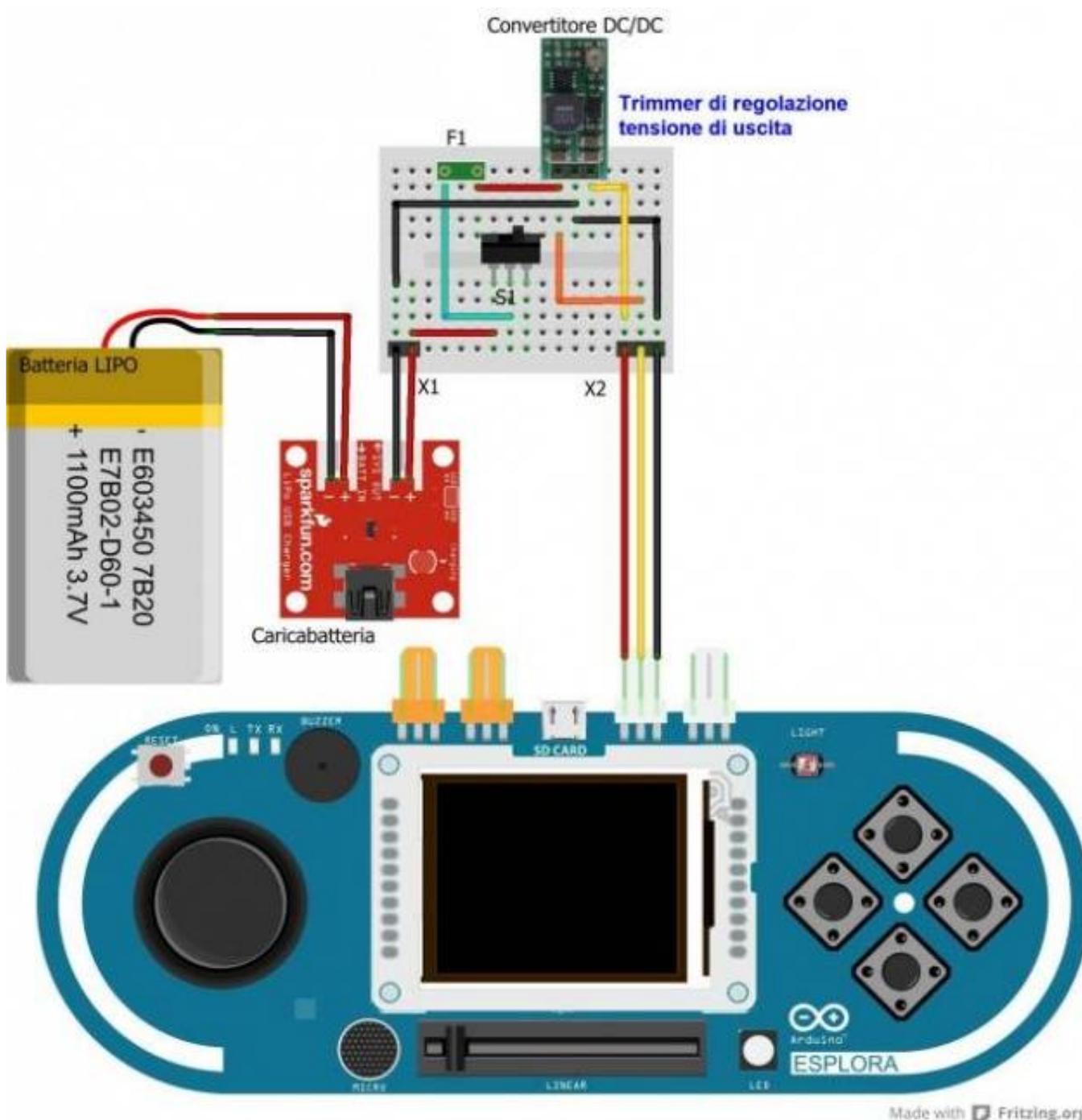
Una mini breadboard

Cavetti assortiti maschio/maschio per cablaggi su breadboard

Il circuito è stato realizzato su breadboard, poiché la versione definitiva sarà poi integrata da una parte TX/RX basata su un modulo bluetooth che renderanno la scheda **Arduino Esplora** una stazione di comando completamente autonoma e completamente wireless.

Schema del circuito

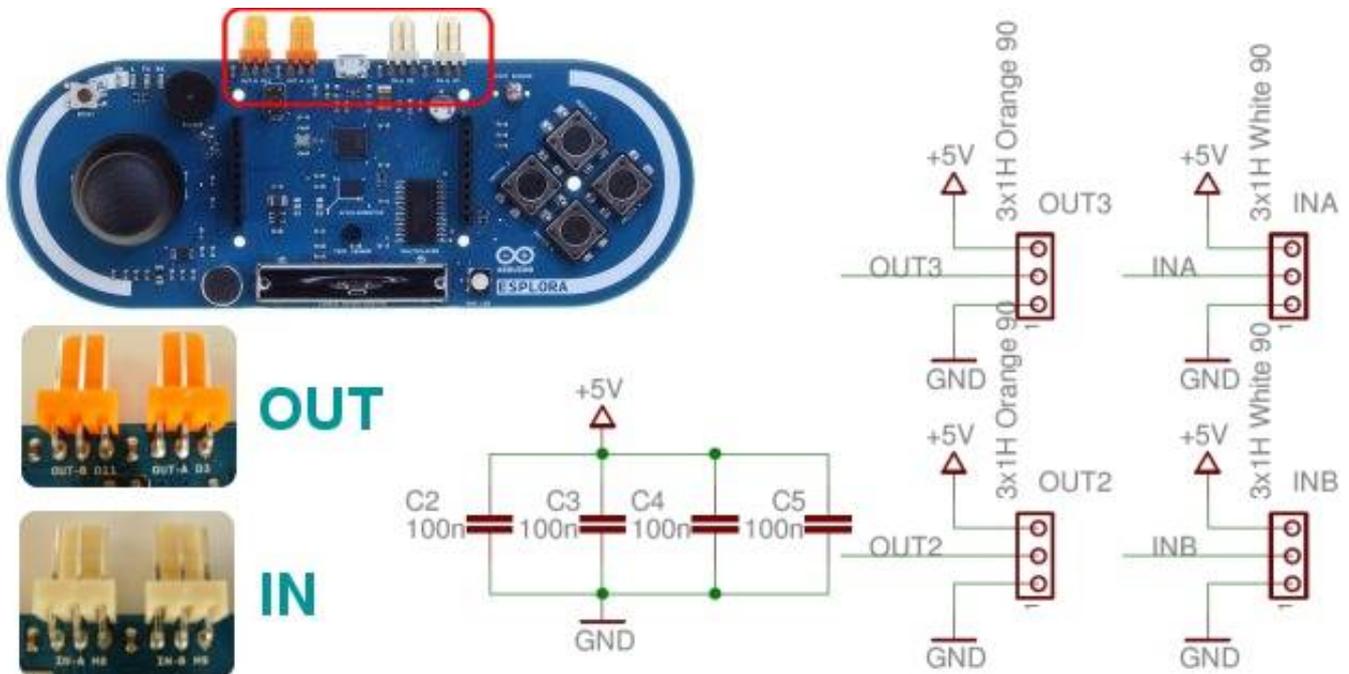
Lo schema del circuito, realizzato con il [programma Fritzing](#) ^[3] è riportato sotto.



La batteria è connessa al caricabatteria e da qui la tensione tramite l'interruttore e il fusibile resettabile arriva al convertitore DC/DC che eleva il valore della tensione da 3,7V a 5V.

Per alimentare la scheda utilizzeremo uno dei connettori **Tinkerkit**, l'ingresso INA, i cui pin 5V e GND sono connessi al resto del circuito.

La scheda ESPLORA ha quattro connettori compatibili con il sistema Tinkerkit: due per gli ingressi di colore bianco e due di uscita di colore arancio.



Utilizziamo anche il pin d'ingresso che in questo caso monitora il valore della tensione della batteria.

Passiamo adesso all'analisi in dettaglio dei componenti principali.

La Batteria

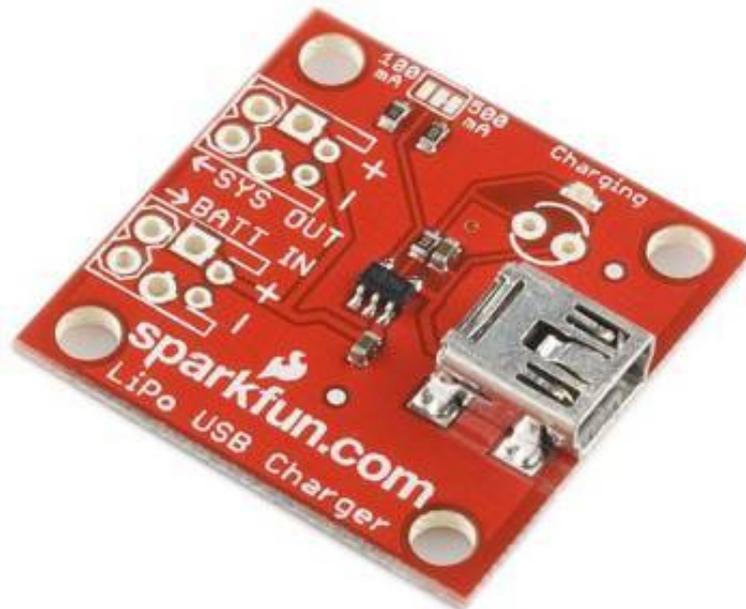
La batteria ricaricabile scelta è un accumulatore agli ioni di litio (a volte abbreviato in Li-Ion).

Per maggiori informazioni si può fare riferimento all'articolo [Batterie al Litio: scegliamo le migliori.](#) [4]

Per questa applicazione viene utilizzata una [batteria Li-Ion con una tensione di 3,7 e una corrente di 184 ma/h prodotta dalla ENIX Energies.](#) [5]

ambientali. Questa regolazione termica consente di ottimizzare il tempo del ciclo di carica pur mantenendo l'affidabilità del dispositivo.

Sono disponibili diverse opzioni, per impostarle si può fare riferimento al data sheet.



La scheda proposta dalla **SparkFun** ha un LED di stato, ed un jumper a saldare per selezionare il valore della corrente di carica a scelta tra 500mA o 100 mA.

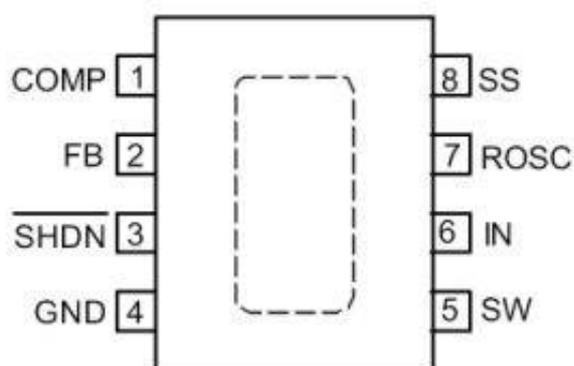
Dispone poi, della presa USB per l'alimentazione e di due connettori; una per collegare la batteria "**BAT IN**" e una di un'uscita "**SYS OUT**", che consente di collegare il circuito di carica direttamente al progetto in modo che non è necessario scollegare il caricabatterie ogni volta che si desidera utilizzarlo.

Convertitore DC/DC

Per elevare la tensione dal valore di 3,7V fornita dalla batteria a quello di 5V necessario ad alimentare la scheda **Arduino Esplora**, la scelta è andata ad un piccolo convertitore DC/DC prodotto dalla Pololu con codice 791 ^[8].

Ha dimensioni particolarmente ridotte (10,7 x 22,4 x 5,8 mm), ed è in grado di convertire una tensione continua compresa tra 1,5 e 16 volt in una tensione di uscita da 2,5 a 9,5 volt.

La massima corrente disponibile è di circa 1600 mA.



La scheda si basa su un integrato SC4501 prodotto dalla Semtech,^[9] un regolatore switching di tipo step-up current-mode ad alta frequenza di commutazione con un transistor di potenza integrato da 2A. La sua frequenza di commutazione elevata (programmabile fino a 2 MHz) consente l'utilizzo di piccoli componenti passivi esterni a montaggio superficiale. Dispone di un Soft-start programmabile che elimina l'alta corrente di spunto in fase di avviamento.

La tensione di uscita può essere regolata ruotando il piccolo trimmer presente sulla scheda.

Fusibile autoripristinante

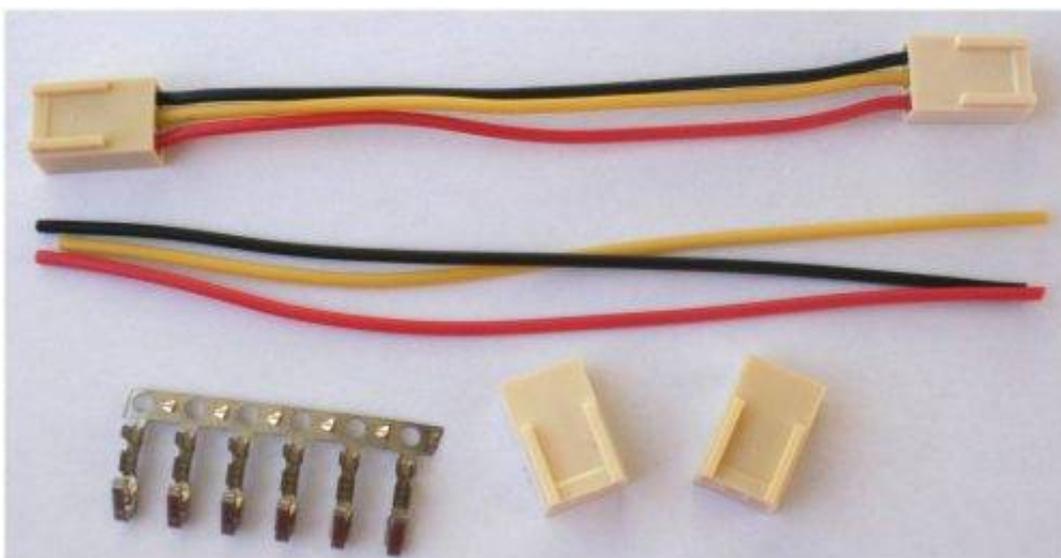
Per la protezione del circuito si è utilizzato un **fusibile autoripristinante** chiamato PTC (polyfuse o polyswitch): si tratta di termistori non-lineari che nel caso del loro surriscaldamento (dovuto ad un elevato assorbimento) diventano non conduttori, ritornando ad esserlo dopo la rimozione del problema e consentendo al circuito di funzionare di nuovo senza aprire il telaio o sostituire nulla.

Assemblaggio

Come già detto, attualmente il circuito è stato assemblato su una basetta bread-board, ma in futuro è prevista la costruzione di un apposito circuito stampato.

Per il cablaggio occorre riferirsi allo schema riportato precedentemente nell'articolo.

Per il collegamento con la scheda occorre realizzare un apposito cavo utilizzando due connettori a tre pin femmina passo 2,54 mm tipo Molex 22-1-3037 (codice RS 679-5375P [10]).



Taratura e collaudo.

L'unica regolazione da compiere è quella della tensione di uscita del modulo switching, per cui con la batteria collegata e l'interruttore su ON, si collegherà un multimetro digitale all'uscita e si regolerà tramite un piccolo cacciavite il trimmer presente sul modulo elevatore sino a leggere il valore di 5V.

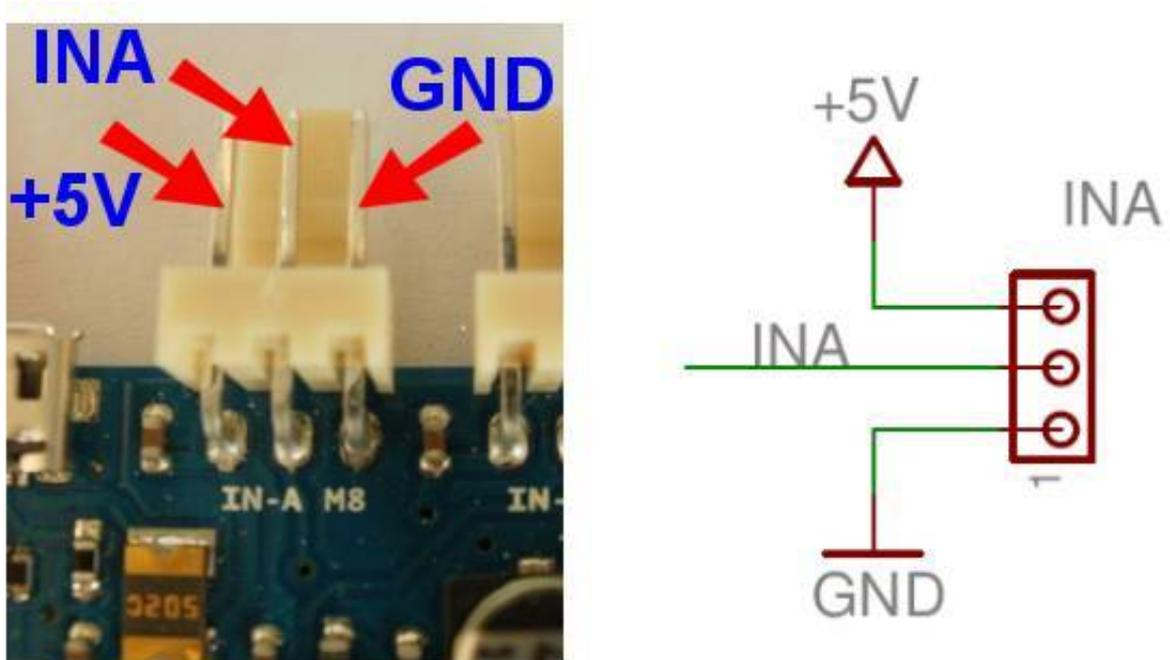
Si potrà a questo punto, considerare conclusa la taratura del circuito.

Programma di test.

Per verificare il funzionamento del circuito, è presente un piccolo programma che mostrerà sul display TTF il valore fornito dalla batteria.

Nel programma, la sezione inerente la lettura del pin Analogico è derivata da un programma scritto da Mike Barela che potrete trovare a [questo link](#) ^[11].

L'operazione preliminare sarà quella di eseguire un collegamento, in modo da portare la tensione tramite un cavo al pin centrale del connettore **IN-A**.



Per caricare il programma, che è presente in [allegato](#) ^[12], utilizzeremo la scheda **Arduino Esplora** collegata normalmente tramite il solo cavo USB alla porta del PC.

Se non avete ancora configurato la scheda potrete leggere l'articolo [Programmiamo la scheda Arduino Esplora.](#) ^[2]

```
// Include le librerie necessarie
#include <Esplora.h>
#include <TFT.h>    // Libreria per LCD ArduinoLCD library
#include <SPI.h>

//Definizione variabile di servizio
int BatteryValue = 0;
char tensione1 [50];
String BatteryValue1;
char batteryPrintout[5];
char valuePrintout[5];
float fBattVoltage = 0;
int iBattVoltage = 0;

//Programma derivato dal sito di Mike Barela per la lettura delle porte analogiche
// http://21stdigitalhome.blogspot.it/2013/03/reading-arduino-esplora-tinkerkit-inputs.html

const byte CH_TINKERKIT_INA = 8; // Add values missing from Esplora.
const byte CH_TINKERKIT_INB = 9;
const byte INPUT_A      = 0;
const byte INPUT_B      = 1;
unsigned int readTinkerkitInput(byte whichInput) { // return 0-1023 from Tinkerkit Input A or B
```

```

return readChannel(whichInput+CH_TINKERKIT_INA);
} // as defined above
unsigned int readChannel(byte channel) { // as Esplora.readChannel is a private function

digitalWrite(A0, (channel & 1) ? HIGH : LOW); // we declare our own as a hack
digitalWrite(A1, (channel & 2) ? HIGH : LOW); //
digitalWrite(A2, (channel & 4) ? HIGH : LOW); // digitalWrite sets address lines for the input
digitalWrite(A3, (channel & 8) ? HIGH : LOW);
return analogRead(A4); // analogRead gets value from MUX chip
}
void setup() {
  Serial.begin(9600);
  // Inizializzazione del display grafico
  EsploraTFT.begin();

  // cancella lo schermo con uno sfondo nero
  EsploraTFT.background(0,0,0);
  // Imposta il colore del testo a viola
  EsploraTFT.stroke(255,0,255);
  // Imposta altezza testo a 2
  EsploraTFT.setTextSize(2);
  // Scrittura del testo nella parte superiore sinistra dello schermo
  // Questo testo rimarra' statico
  EsploraTFT.text("Tensione: \n ",0,0);

  // Imposta altezza testo per il loop a 5
  EsploraTFT.setTextSize(5);
}

void loop() {
  // lettura valore della tensione e inserimeto valore in
  // una variabile
  unsigned int BatteryValue= readTinkerkitInput(INPUT_A);

  // Calcolo del valore della tensione della batteria
  float fBattVoltage = ((BatteryValue*5.0)/1023)*1000.0;

  iBattVoltage = (int)fBattVoltage; // Conversione del valore float in integer
  char battVoltageStr[50]; //Assegnazione a char
  sprintf(battVoltageStr,"%d",iBattVoltage); // Copia intero char come stringa
  // Utilizzando gli argomenti indicizzati viene messo ogni numero al posto giusto
  // nota il punto decimale e' inserito manualmente.

```

```
printf(tensione1,"%c.%c%cV",battVoltageStr[0],battVoltageStr[1],battVoltageStr[2]);

// Conversione valore letto da porta analogica in stringa
BatteryValue1 = String (BatteryValue);
BatteryValue1.toCharArray (valuePrintout, 6);

// imposta il colore del testo a rosso
EsploraTFT.stroke(255,0,0);
// stampa su LCD valore letto e di quello in volt
EsploraTFT.text(tensione1, 0, 80);
// imposta il colore del testo a verde
EsploraTFT.stroke(0,255,0);
// stampa su LCD valore letto dall' ADC
EsploraTFT.text(valuePrintout, 0, 30);

//Pausa di un secondo per la prossima lettura
delay(1000);

// cancella il testo per la prossima lettura
EsploraTFT.stroke(0,0,0);
EsploraTFT.text(tensione1, 0, 80);
EsploraTFT.text(valuePrintout, 0, 30);
}
```

Filmato illustrativo

Conclusioni

Nella prossima puntata vedremo come collegare un modulo bluetooth alla scheda e con

questo comanderemo un dispositivo connesso ad una scheda **Arduino UNO** ^[13],

Per cui, non perdetevi la prossima puntata.

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

URL to article: <https://it.emcelettronica.com/rendiamo-autonoma-scheda-arduino-esplora>

URLs in this post:

[1] scheda Arduino Esplora,: <https://it.emcelettronica.com/scopriamo-nuova-scheda-arduino-esplora>

[2] installarla e programmarla,: <https://it.emcelettronica.com/programmiamo-scheda-arduino-esplora>

[3] programma Fritzing : <http://fritzing.org/>

[4] Batterie al Litio: scegliamo le migliori.: <https://it.emcelettronica.com/batterie-al-litio-scegliamo-le-migliori>

[5] batteria Li-Ion con una tensione di 3,7 e una corrente di 184 ma/h prodotta dalla ENIX Energies.: <http://it.rs-online.com/web/p/pacchi-batteria-al-litio-ricaricabili/5306325/>

[6] SparkFun codice PRT-10161: <https://www.sparkfun.com/products/10161>

[7] Microchip tipo MCP73831/2:

<http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en026250>

[8] Pololu con codice 791: <http://www.pololu.com/catalog/product/791>

[9] SC4501 prodotto dalla Semtech,: <http://www.semtech.com/power-management/switching-regulators/sc4501/>

[10] RS 679-5375P: <http://it.rs-online.com/web/p/custodie-per-connettori-da-pcb/6795375P/>

[11] questo link: <http://21stdigitalhome.blogspot.it/2013/03/reading-arduino-esplora-tinkerkit-inputs.html>

[12] allegato: https://it.emcelettronica.com/files/Esplora_lettura_batteria.zip

[13] **Arduino UNO**: <https://it.emcelettronica.com/arduino>

Copyright © 2017 Elettronica Open Source. All rights reserved.