

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

Dotiamo l'Arduino Esplora dell'interfaccia Bluetooth

Posted By *Adriano Gandolfo* On 20 dicembre 2013 @ 10:00 In Arduino | [5 Comments](#)



Prosegue la nostra analisi della scheda Arduino Esplora ^[1] in quest'articolo in cui vedremo come dotarla di un modulo Bluetooth, che imposteremo come Master. Nell'esempio proposto comanderemo un secondo modulo Bluetooth, impostato come Slave, collegato a una scheda Arduino UNO. Piloteremo un semplice led, ma l'esempio sarà utile per comandare, in futuro, dispositivi più complessi.

La scheda Arduino Esplora non dispone di interfacce di comando remoto, vedremo come dotarla di un modulo bluetooth.

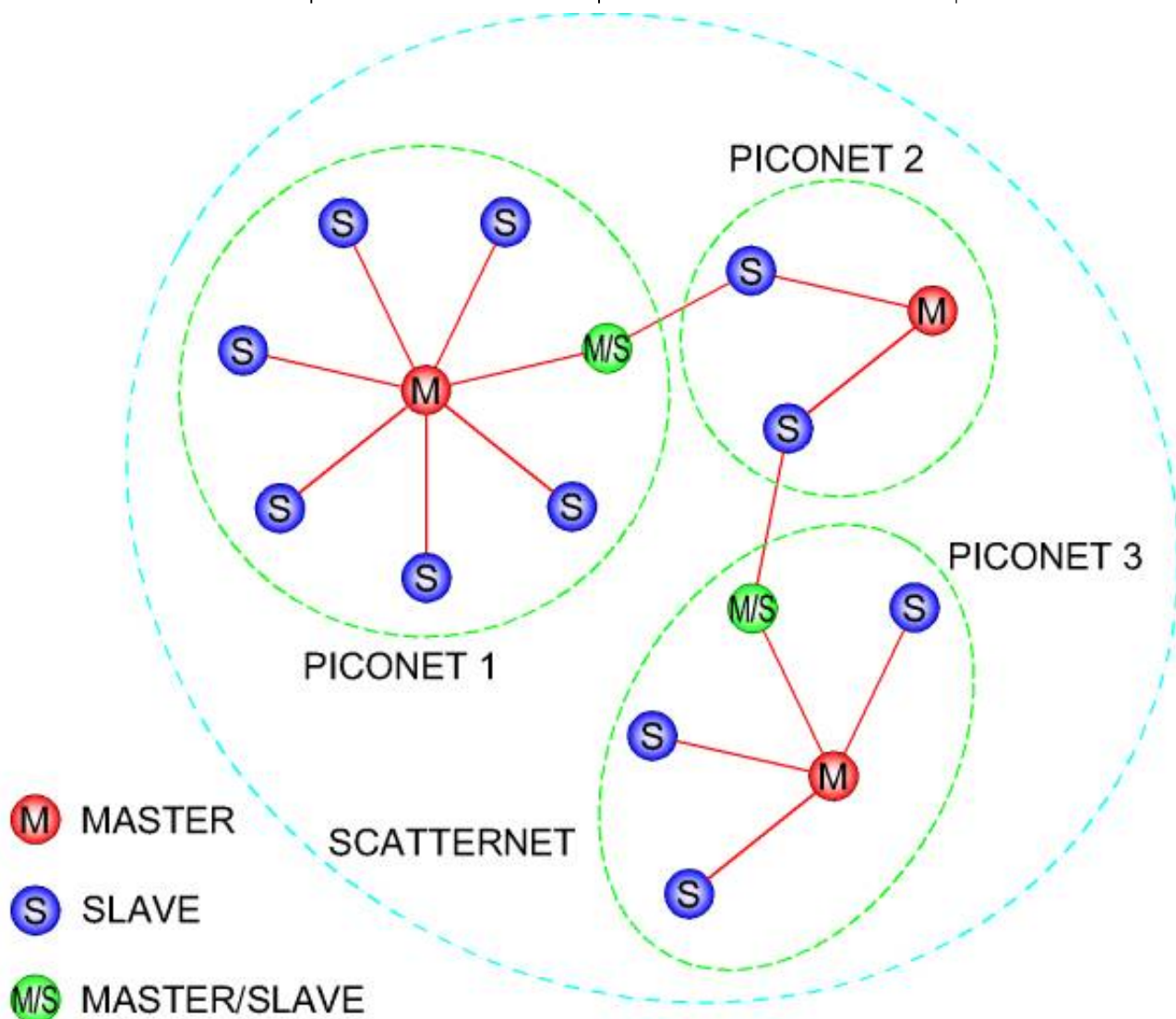
Breve descrizione della tecnologia Bluetooth

Il Bluetooth è una delle tecnologie che ha sostituito la necessità di collegamento via cavo, permettendo lo scambio di dati su brevi distanze.

Molti sono i dispositivi che ne sono attualmente dotati, a partire dai telefoni cellulari, alle stampanti e molti altri ancora.

La trasmissione utilizza la porzione inferiore dello spettro radio EM con frequenza di 2,45 GHz. Il sistema può connettere un'unità master con un massimo di altre sette unità, denominate slave, formando un **PAN** (**P**ersonal **A**rea **N**etwork) chiamato **Piconet**.

Otto di questi gruppi possono essere collegati insieme, sempre per mezzo della tecnologia Bluetooth, per formare un network ancora più grande chiamato **Scatternet**.



In teoria, non vi è un limite al numero di unità che possono essere collegate in questo modo via Bluetooth.

Ogni dispositivo Bluetooth è dotato di un indirizzo univoco a livello mondiale, assegnato dal costruttore.

Equivalente del **MAC address** nelle reti ethernet, è rappresentato da **48 bit**, il che permette di avere fino a $2,84 \times 10^{14}$ dispositivi.

Ad ogni dispositivo può anche essere assegnato un nome che può essere una stringa di caratteri unicode, definito dall'utente che non è univoco, ma che facilita il riconoscimento del dispositivo.

Natura master-slave della comunicazione

Come abbiamo visto, il collegamento bluetooth prevede un'unità **Master** e un'unità **Slave**, che devono seguire alcune regole:

Chi inizia una conversazione funge da Master

La sequenza dei canali utilizzati è derivata dall'indirizzo MAC del Master

Il master invia, durante uno slot, una richiesta ad uno slave, che gli risponde nello slot successivo (sul prossimo canale della sequenza pseudocasuale)

Possibilità di avere fino a 7 slave attivi

Master e slave devono mantenere il clock allineato

Paging e Inquiry

Un dispositivo che intende iniziare l'invio di dati, deve inviare dei pacchetti di richiesta riportanti l'indirizzo del destinatario.

Se il destinatario è in ascolto, può sincronizzarsi al master e accettare la comunicazione. L'invio di richieste si chiama "**paging**", la disponibilità del destinatario a ricevere richieste di connessione si chiama "**page scan**". La connessione richiede la conoscenza dell'indirizzo MAC del destinatario.

In alcuni casi per scoprire i possibili destinatari, una stazione può inviare pacchetti di "**Inquiry**": chi riceve tale pacchetto, risponde indicando il proprio indirizzo MAC, il proprio clock ed un intero indicante la tipologia del dispositivo (Class Of Device).

Per ricevere richieste di paging e di inquiry, mittente e destinatario devono utilizzare lo stesso canale radio.

Per ulteriori informazioni vedere l'articolo [Bluetooth come funziona](#) ^[2]

Componenti del progetto

Dopo la teoria sulla comunicazione bluetooth, possiamo ora ad analizzare i principali componenti che compongono il progetto:

[Scheda Arduino Esplora](#) ^[1]

Due Moduli Bluetooth BlueSMiRF Silver

[Arduino UNO](#) ^[3]

Per i componenti del circuito di alimentazione, fate riferimento all'articolo [Rendiamo autonoma la scheda Arduino Esplora](#) ^[4]

Scheda Arduino Esplora

Per una descrizione accurata della scheda si può fare riferimento al precedente articolo

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

Mentre per la sua programmazione potete leggere l'articolo [Programmiamo la scheda Arduino Esplora](#) ^[5]

Le principali caratteristiche della scheda sono:

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)

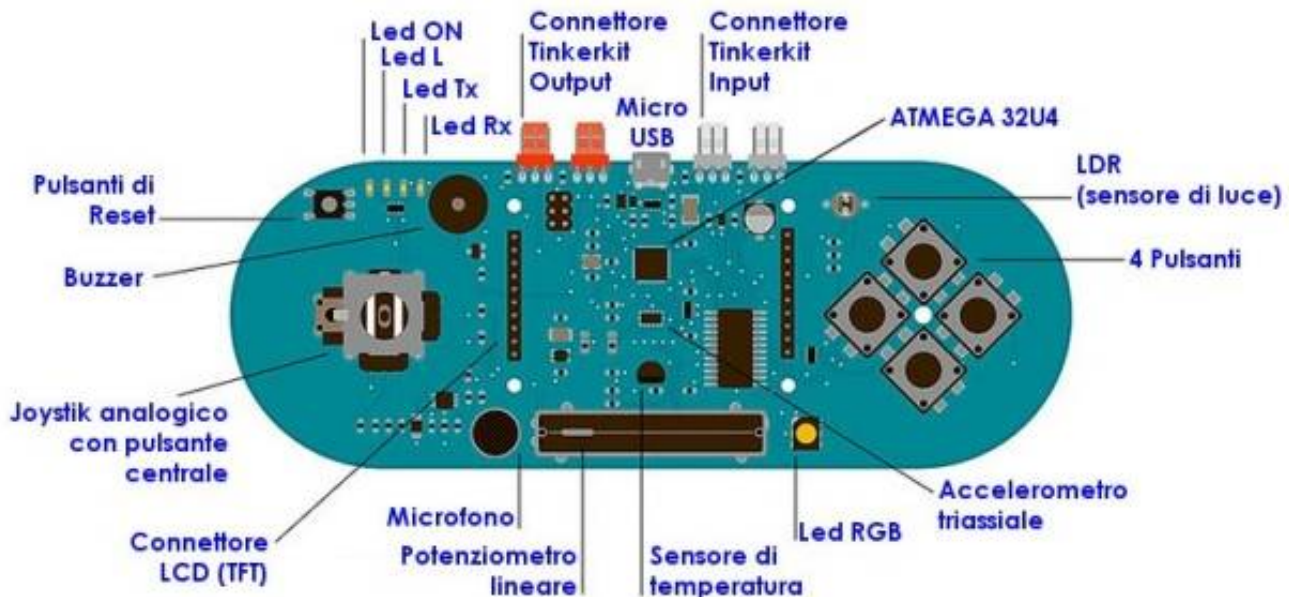
Un buzzer per produrre suoni

Un led luminoso a LED tipo RGB con elementi Rosso, Verde, 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



Arduino UNO

La **scheda Arduino UNO** [3] è basata sul processore **ATMEGA328** con bus a 8 bit prodotto dalla Atmel con architettura di tipo RISC, le sue principali caratteristiche sono:

Memoria flash da 32KB ISP con possibilità di lettura/scrittura

Memoria EEPROM da 1KB

Memoria SRAM da 2KB.

23 porte general purpose di I/O

32 registri di lavoro general purpose

Tre flessibili timer / contatori con modalità di confronto

Possibilità d'interrupts esterni e interni

Una seriale USART programmabile

Un'interfaccia seriale a 2 fili compatibile I2C

Una porta SPI seriale

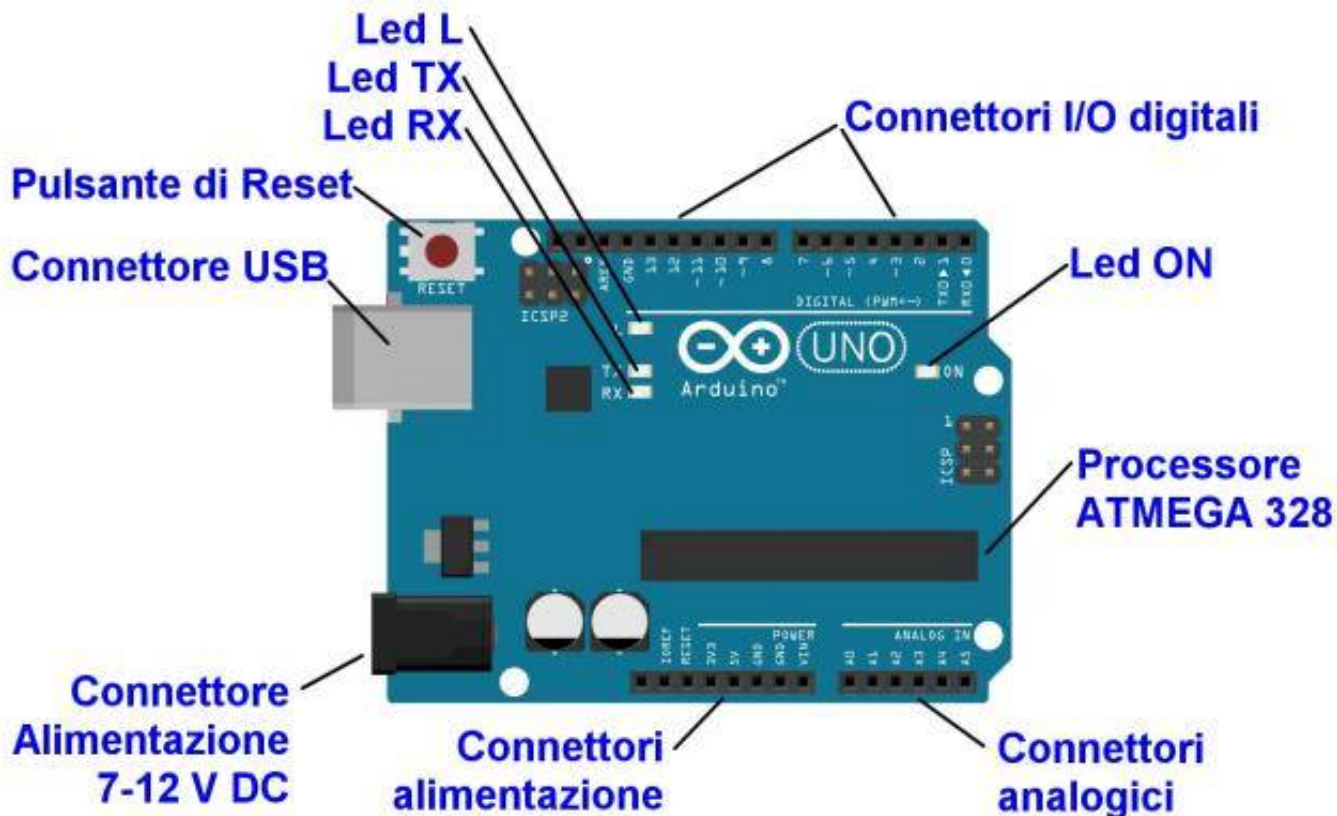
6 convertitori A/D a 10-bit

Timer watchdog programmabile con oscillatore interno

5 modalità di risparmio energetico selezionabili via software

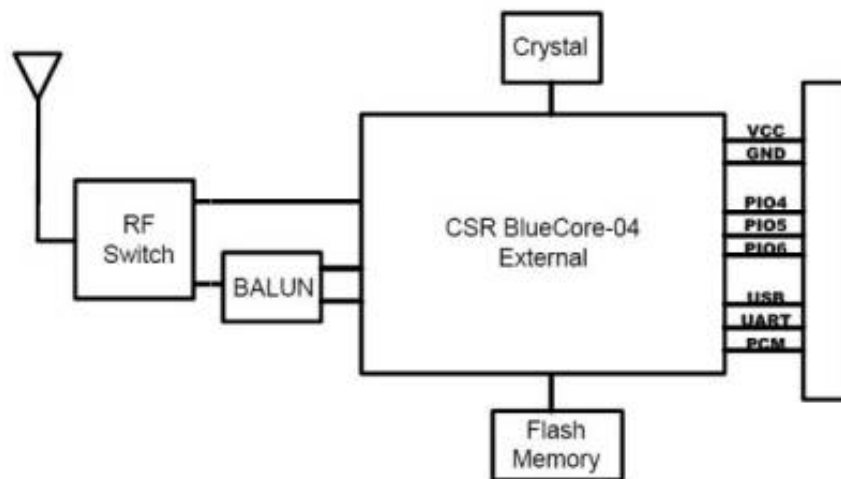
Tensione di funzionamento compresa tra 1,8-5,5 volt.

La scheda presenta dei pin strip, dove sono riportate l'alimentazione e le porte di I/O.

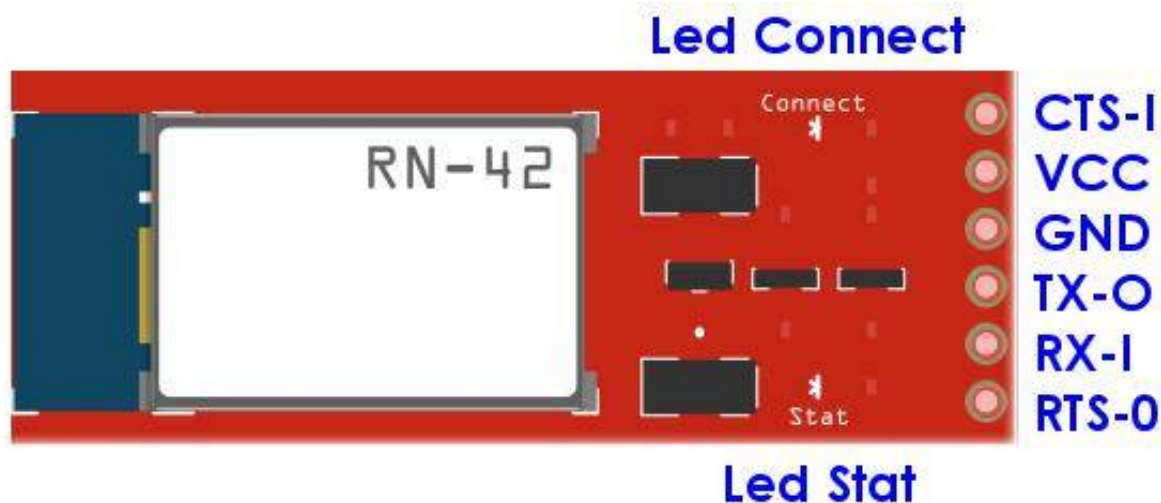


Modulo Bluetooth BlueSMiRF Silver

Il modulo **BlueSMiRF Silver** [6] è un modem wireless Bluetooth prodotto dalla **SparkFun** [7], basato sull'integrato **RN-42** [8] prodotto dalla **ROVING Networks** acquisita recentemente dalla **Microchip**.



Schema a blocchi del modulo RN-42



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 (18m)

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

Questo modem funziona come una porta seriale (RX/TX), potendo trasmettere un flusso seriale da 2400 a 115200 bps.

L'unità può essere alimentata con una tensione i cui valori possono essere compresi tra i 3,3 V fino a 6V, in quanto è dotato di un regolatore di tensione tipo **MIC5205** che fornisce in uscita i 3.3V per il funzionamento dell'integrato.

Tutti i pin del segnale sono 3V-6V tolleranti, in quanto sulla scheda sono già presenti i circuiti per adattare i livelli di tensione formati da MOSFET (N-Channel) di tipo SMD modello **BSS138**.

Il modulo presenta una pin strip a 6 pin in cui sono presenti i segnali: **RX, TX, CTS, RTS, VCC e GND**.

Sono inoltre, presenti due led indicatori: il led di colore rosso (**Stat**) segnala che il modulo è alimentato, ma non connesso; il led di colore verde (**Connect**) segnala che è stata stabilita una connessione.

Collegamenti Stazione di controllo

Nello schema seguente ([vedere allegato](#) ^[9] realizzato con il [programma Fritzing](#) ^[10]), sono indicati i collegamenti da compiere. Questi comportano una modifica di quelli previsti

nel precedente articolo **[Rendiamo autonoma la scheda Arduino Esplora](#)** ^[4].

Per cui occorre montare, oltre ai componenti descritti nel precedente articolo, il modulo bluetooth.

Ecco l'elenco completo dei componenti necessari:

Scheda [Arduino Esplora](#) ^[11]

[Display LCD TFT](#) ^[12]

Una Mini Breadboard 17x10

Una batteria di tipo Li-Ion

Un modulo caricabatteria [SparkFun codice PRT-10161](#) ^[13]

Un modulo convertitore DC/DC del tipo Step Up dalla [Pololu con codice 791](#) ^[14]

Modulo Bluetooth BlueSMiRF Silver [SparkFun codice WRL-10269](#) ^[6]

Un interruttore

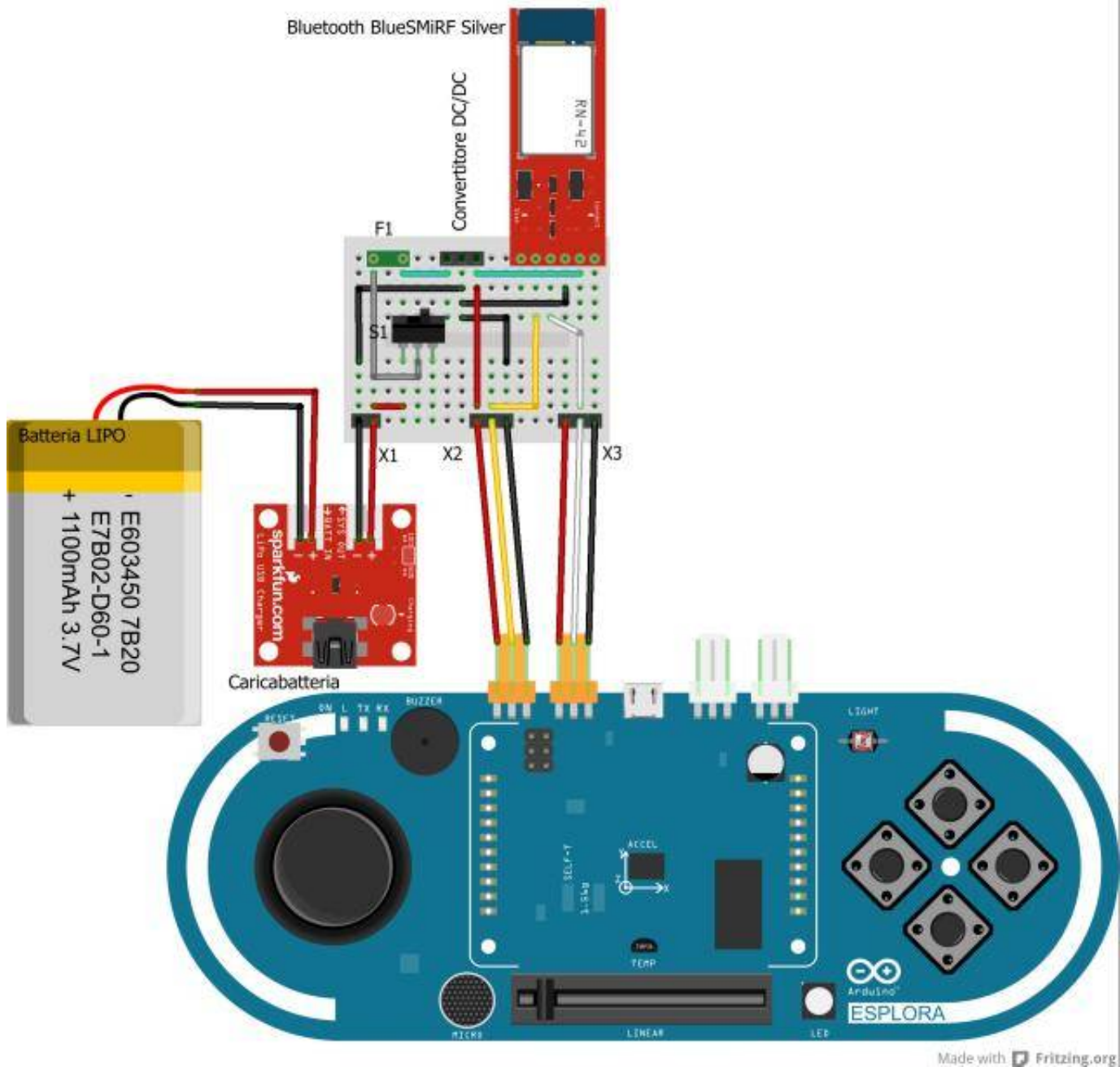
Un fusibile resettabile da 500 mA

Due cavi realizzati utilizzando dei connettori a tre pin femmina passo 2,54 mm tipo Molex

22-1-3037 ([codice RS 679-5375P](#) ^[15])

Cavi colorati per breadboard

Il collegamento viene effettuato utilizzando i due connettori presenti sulla scheda collegati ai pin digitali D3 e D11.



Schema di collegamento

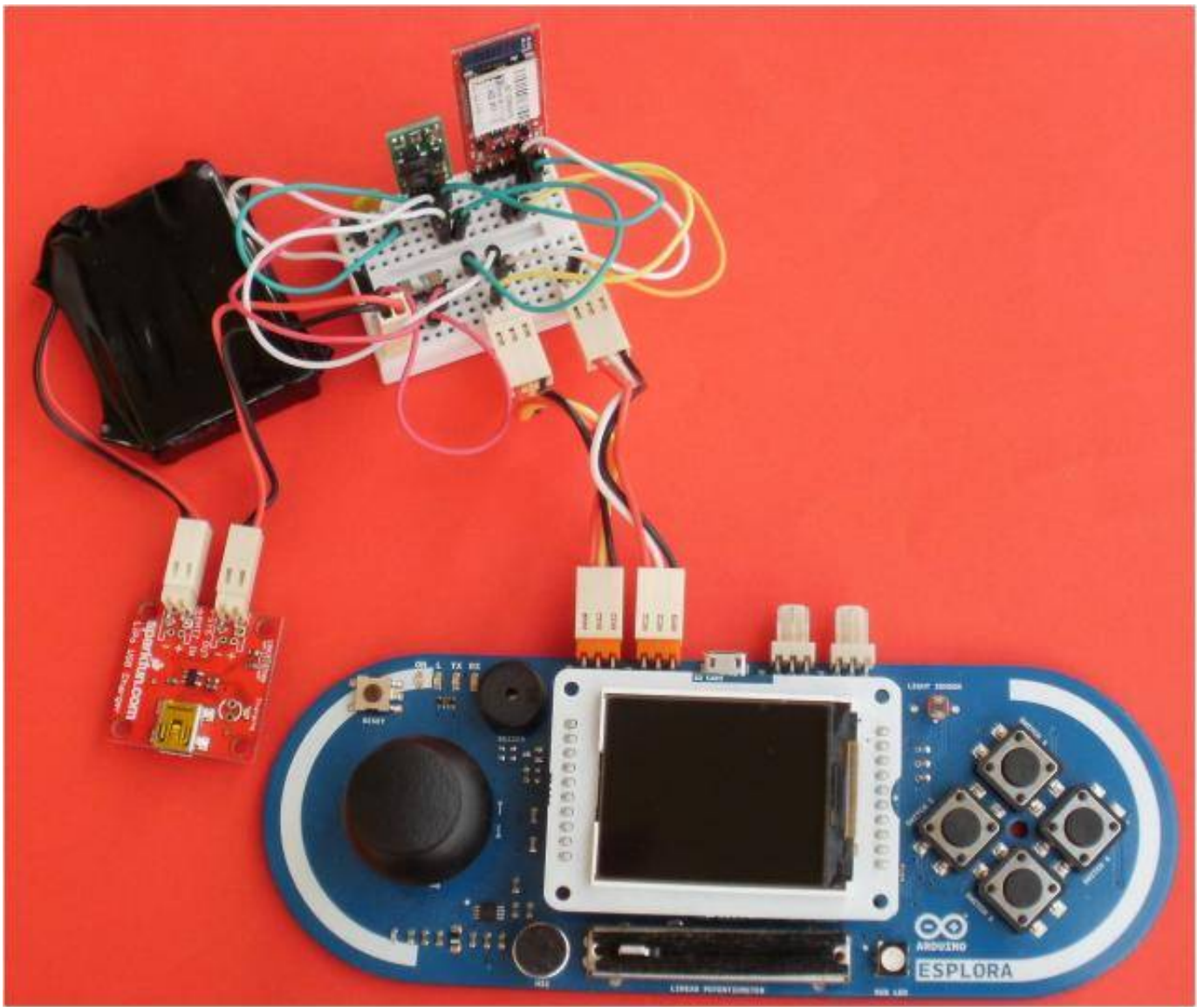
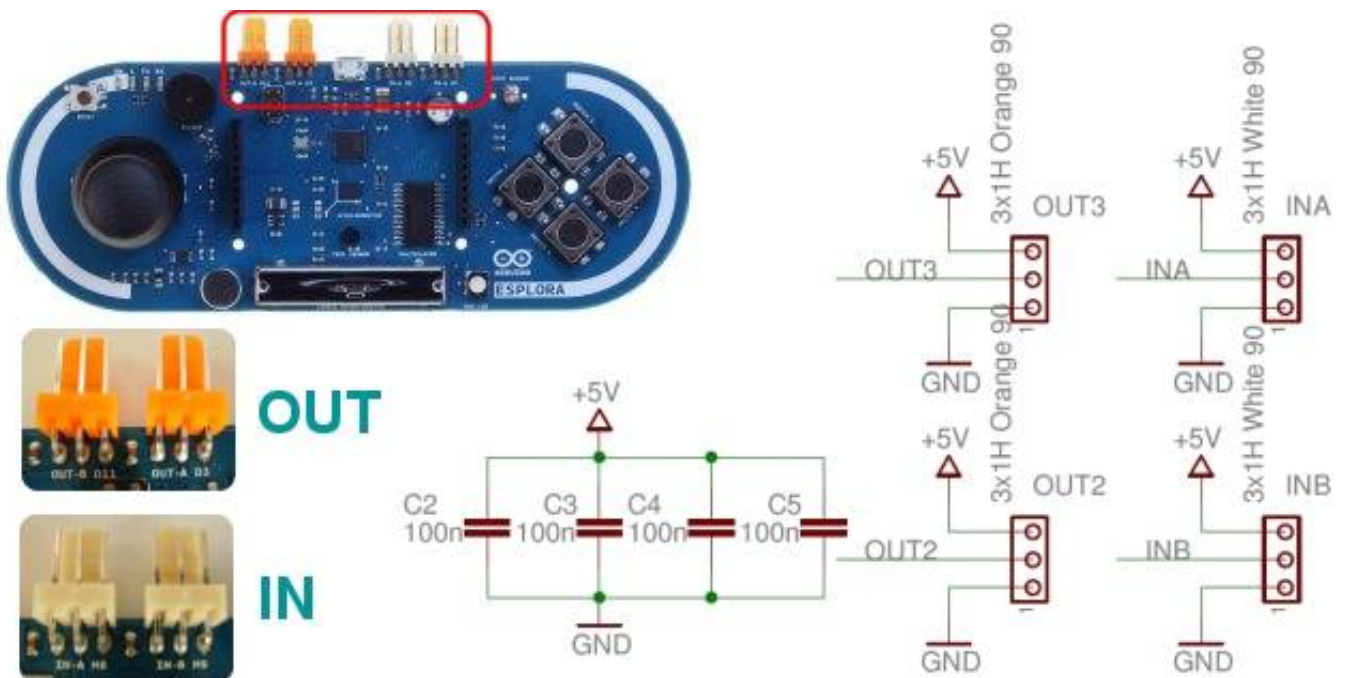
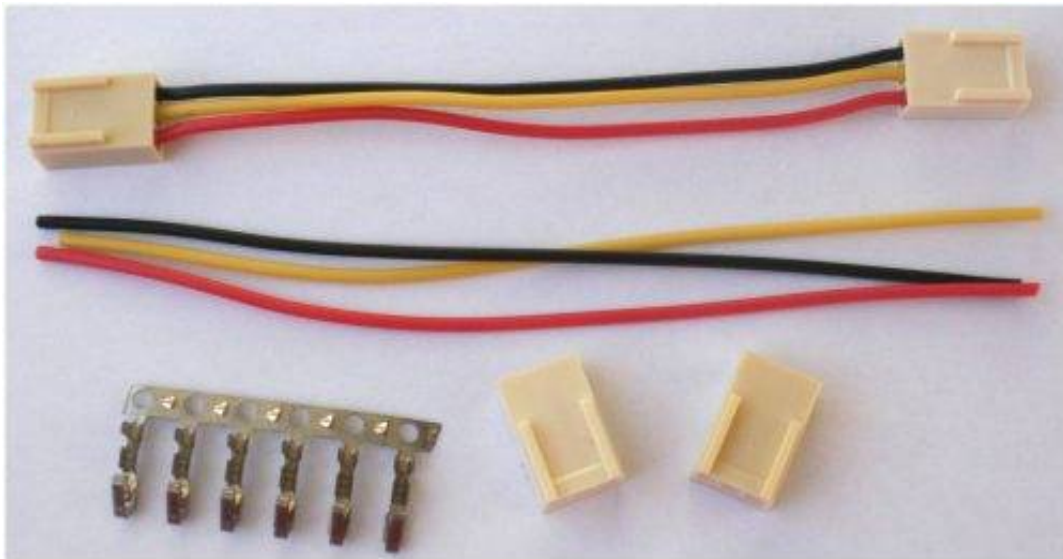


Foto del collegamento realizzato



Connettori di IN/OUT della scheda Esplora



Cavo di collegamento con scheda Arduino Esplora

Collegamenti stazione ricevente.

Nello schema seguente (vedere allegato ^[16]), sono indicati i collegamenti da effettuare per il test della stazione ricevente.

Ecco l'elenco completo dei componenti necessari:

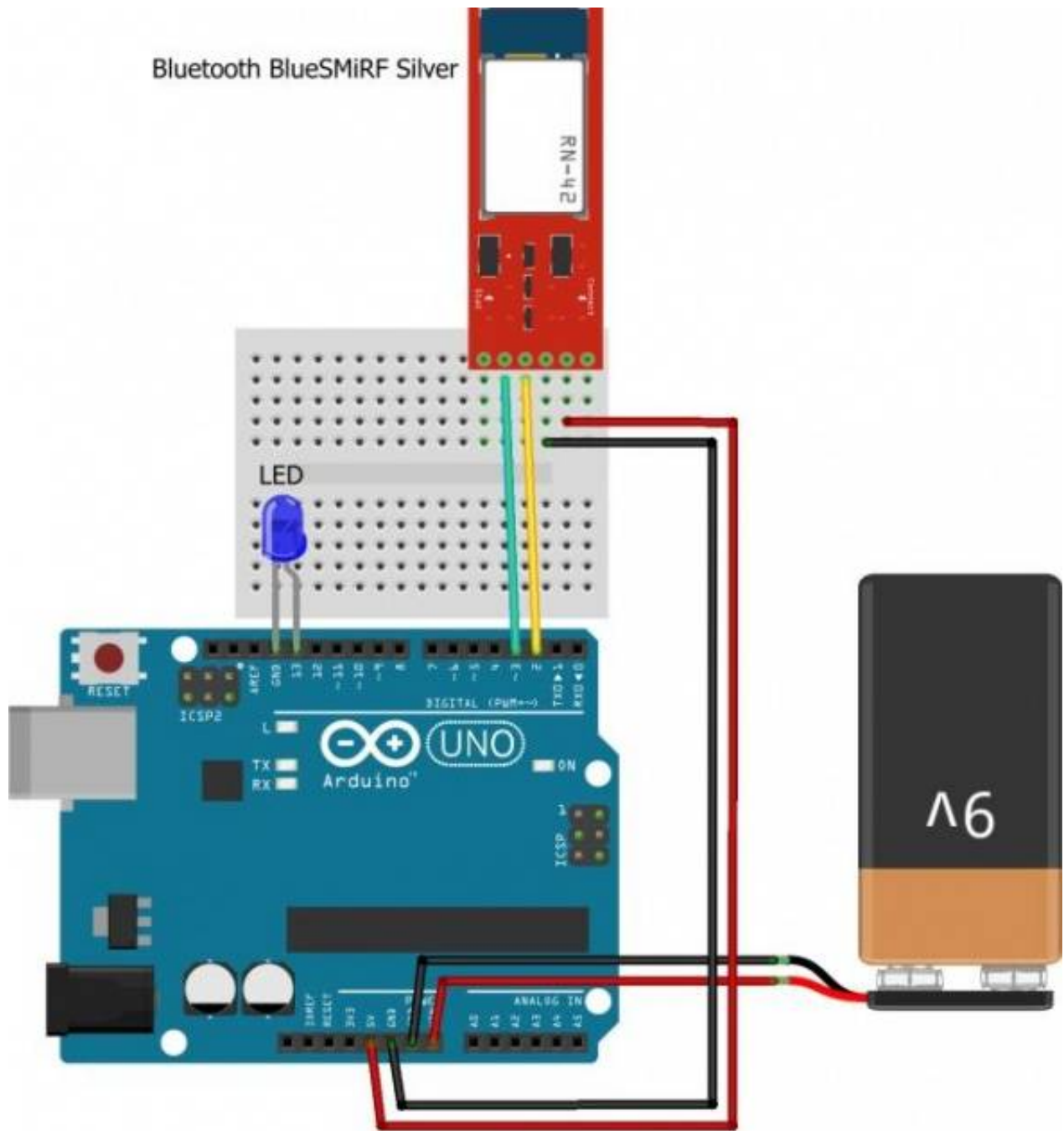
Scheda Arduino UNO ^[17]

Una Mini Breadboard 17x10

Modulo Bluetooth BlueSMiRF Silver SparkFun codice WRL-10269 ^[18]

un led

Cavi colorati per breadboard



Made with  Fritzing.org

Schema di collegamento

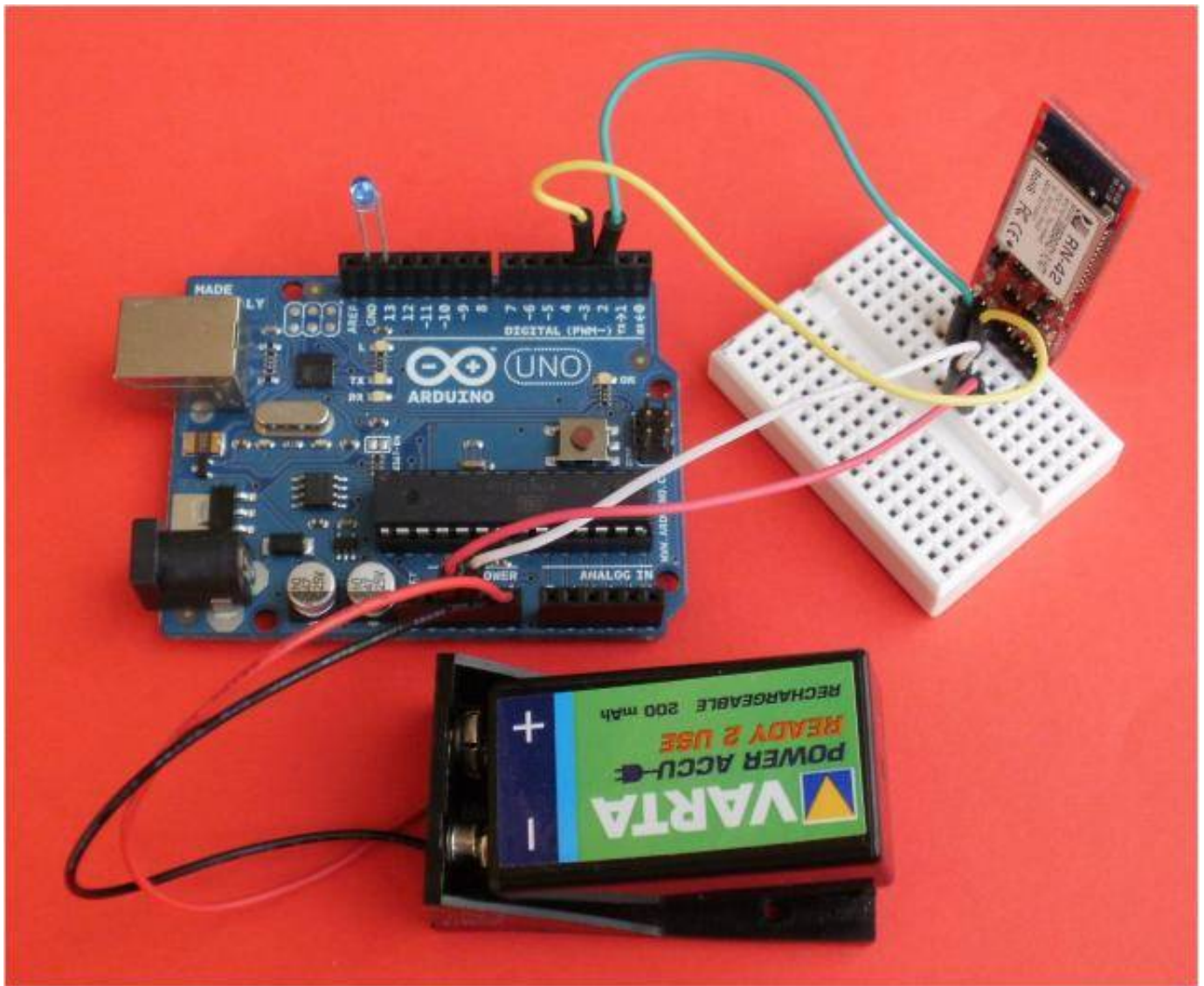


Foto del collegamento realizzato

Programma Master

Passiamo ora ad analizzare il programma da inserire sulla Scheda **Arduino Esplora** che eseguirà le seguenti funzioni:

Caricamento delle librerie necessarie: Software Serial, SPI, Esplora TFT

Definizione dei pin collegati alla porta seriale (nel nostro caso i pin 11 e 3)

Impostazione del pin di collegamento al led presente sulla scheda (il pin 13)

Nel dettaglio nel ciclo di **setup**, abbiamo l'impostazione del modulo bluetooth come **Master**.

Per fare questo, dopo aver impostato la velocità di trasmissione della seriale a **115200 bbs** (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.

Si 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.



Dopo una nuova pausa, trasmettiamo il comando `println("---")`, che permette l'uscita dal **Command Mode**.

Sotto le righe di codice corrispondenti:

```
mySerial.begin(115200);
mySerial.print("$$$");
delay(100);
mySerial.println("SM,1");
delay(100);
//inserire il codice MAC del modulo Slave
mySerial.println("C,000666464321");
delay(100);
mySerial.println("---");
```

Segue, una parte di programma per l'impostazione del **display TFT**.

Si passa quindi al programma vero e proprio presente nel **ciclo LOOP**.

Questo legge ciclicamente i tasti impostati (**2 e 4**) e a seconda di quello premuto, accede o spegne, il led presente sulla scheda (connesso al pin I/O 13), inviando poi tramite la seriale (utilizzando il pin I/O 3) il carattere "a" o "s", in contemporanea invia un Feedback al **display TFT**.

Segue il **listato completo** ^[19] che è presente negli allegati.

```
/*
Programma di test per interfaccia SERIALE
Sezione MASTER - Arduino esplora
* RX è il pin digitale 11 (da collegarsi al TX del modulo BlueSMiRF)
* TX è il pin digitale 3 (da collegarsi al RX del modulo BlueSMiRF)
```


Questo codice di esempio è di dominio pubblico.

```

*/
#include <SoftwareSerial.h>
#include <SPI.h>
#include <Esplora.h>
#include <TFT.h>
SoftwareSerial mySerial(11, 3); // RX, TX
int led = 13; // il Pin 13 è collegato al led L sulla scheda esplora
void setup()
{
  //Imposta il modulo BlueSMiRF SILVER come MASTER
  //e setta il numera MAC del ricevente slave
  mySerial.begin(115200);
  mySerial.print("$$$");
  delay(100);
  mySerial.println("SM,1");
  delay(100);
  //inserire il codice MAC del modulo Slave
  mySerial.println("C,000666464321");
  delay(100);
  mySerial.println("---");
  //-----
  pinMode(led, OUTPUT); // initialize il pin digitale come output
  EsploraTFT.begin();
  EsploraTFT.background(0,0,0); //sfondo nero
  EsploraTFT.setTextSize(2);
  EsploraTFT.stroke(0,255,0); //colore del testo verde
  EsploraTFT.text("  Test uso",0,0);
  EsploraTFT.text("  Modulo",0,16);
  EsploraTFT.text("  Bluetooth",0,32);
  EsploraTFT.setTextSize(1);
  EsploraTFT.stroke(255,255,255); //colore del testo bianco
  EsploraTFT.text(" 2-Accende - 4 Spegne",0,60);
  delay(500);
}
void loop() // LOOP di lettura tasti e invio carattere
{
  int button_2 = Esplora.readButton(SWITCH_2);
  int button_4 = Esplora.readButton(SWITCH_4);
  if (button_2==LOW) {
    digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
    mySerial.print ("a"); //Accende led
  }
}

```

```

cancellazione ();
EsploraTFT.stroke(255,0,0); //colore del testo rosso
EsploraTFT.text("LED ACCESO",30,80);
EsploraTFT.stroke(255,0,0);
EsploraTFT.fill(255,0,0);
EsploraTFT.circle(105,83,10);
}
if (button_4==LOW){
  digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
  mySerial.print ("s"); //Spegne led
  cancellazione ();
  EsploraTFT.stroke(255,255,255); //colore del testo bianco
  EsploraTFT.text("LED SPENTO",30,80);
  EsploraTFT.stroke(255,0,0); // Colore rosso
  EsploraTFT.circle(105,83,10); // Disegno cerchio
}
delay (100);
}
void cancellazione ()
{
  EsploraTFT.stroke(0,0,0);
  EsploraTFT.fill(0,0,0);
  EsploraTFT.rect(30,80, 80, 8); //Cancellazione valore
}

```

Programma Slave

Il programma da inserire sulla **Scheda ARDUINO UNO** esegue le seguenti funzioni:

Caricamento della libreria necessaria: Software Serial

Definizione dei pin collegati alla porta seriale (nel nostro caso i pin 2 e 3)

Impostazione del pin di collegamento al led presente sulla scheda (il pin 13)

Nel ciclo di setup abbiamo: prima di tutto l'impostazione del modulo bluetooth come

SLAVE, la procedura è analoga a quella precedentemente descritta, l'unica variazione è che il comando in questo caso sarà:

```
println("SM,o");
```

che imposta appunto il modulo come **Slave**.

Si passa quindi al programma vero e proprio presente nel **ciclo LOOP**: questo legge ciclicamente la porta seriale connessa al modulo bluetooth e quando riceve un carattere, lo analizza; se il carattere è **"a"** **accende il led**, mentre se il carattere è **"s"** lo **spegne**.

Segue il **listato completo** ^[20] che è presente negli allegati.

```

/*Programma di test per interfaccia SERIALE
Riceve dalla seriale hardware, invia alla seriale del software.
Riceve daL software seriale , e invia al sesiale hardware.
Il circuito:
* TX è il pin digitale 2 (da collegarsi al RX del modulo BlueSMiRF)
* RX è il pin digitale 3 (da collegarsi al TX del modulo BlueSMiRF)
Questo codice di esempio è di dominio pubblico.
*/
#include <SoftwareSerial.h>
SoftwareSerial mySerial(3, 2); // RX, TX
const int ledPin = 13; // the number of the LED pin
void setup()
{
  //Imposta il modulo BlueSMiRF SILVER come SLAVE
  mySerial.begin(115200); //imposta la porta di comunicazione con il modulo
BlueSMiRF
  mySerial.print("$$$");
  delay(100);
  mySerial.println("SM,0");
  delay(100);
  mySerial.println("---");
  delay (1000);
  //-----
  Serial.begin(115200); //imposta la porta di comunicazione seriale
  pinMode(ledPin, OUTPUT); // initialize the LED pin as an output:
}
void loop()
{
  while (mySerial.available() < 1) {
  } // Attesa sino a quando riceve un carattere
  char val = mySerial.read(); // Legge il carattere dal modulo Bluetooth
  Serial.println (val); //per test
  switch(val) // Eseguie i comandi in base al carattere ricevuto
  {
  case 'a': // Se il carattere ricevuto è 'a' accende il led
    // turn LED on:
    digitalWrite(ledPin, HIGH);
    break;
  case 's': //Se il carattere ricevuto è 's' spegen il led
    // turn LED off:
    digitalWrite(ledPin, LOW);
    break;
  }
}

```

```
}  
}
```

Collaudo del dispositivo

Una volta caricati i relativi programmi nelle due schede, si potrà passare al collaudo del nostro collegamento wireless.

In questo caso, è necessario alimentare la scheda Arduino UNO (modulo slave) prima della scheda Arduino Esplora (modulo Master) con un anticipo di circa 5 secondi.

Se tutto funzionerà, dopo un lampeggio dei led rossi (Stat), si avrà l'accensione dei led verdi (Connect) che segnerà l'avvenuta connessione dei due moduli.

A questo punto, premendo il tasto "SWITCH 2", si dovrà accendere il led presente sulla scheda Arduino UNO, mentre il tasto "SWITCH 4" lo farà spegnere.



Tasti utilizzati dal programma

Le istruzioni d'uso e lo stato del led sono riportate sullo schermo TFT della scheda Arduino Esplora.



Display Arduino Esplora

Filmato illustrativo

Conclusioni

In quest'articolo, abbiamo visto com'è possibile creare un collegamento wireless tra la scheda Arduino Esplora e un'altra scheda Arduino.

Dalla prossima puntata vedremo come costruire un piccolo robot che potremo comandare in modo wireless.

Nulla vieta comunque, di utilizzare questo sistema per ricevere da sensori esterni collegati alla scheda Arduino UNO , mostrando per esempio le informazioni sul display della Arduino Esplora.

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

URL to article: <https://it.emcelettronica.com/dotiamo-larduino-esplora-dellinterfaccia-bluetooth>

URLs in this post:

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

[2] **Bluetooth come funziona:** <https://it.emcelettronica.com/bluetooth-come-funziona>

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

[4] **Rendiamo autonoma la scheda Arduino Esplora:** <https://it.emcelettronica.com/rendiamo-autonoma-scheda-arduino-esplora>

[5] Programmiamo la scheda Arduino Esplora: <https://it.emcelettronica.com/programmiamo-scheda-arduino-esplora>

- [6] **BlueSMiRF Silver**: <https://www.sparkfun.com/products/10269>
- [7] **SparkFun**: <https://www.sparkfun.com/>
- [8] **RN-42** : <http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en558330>
- [9] vedere allegato:
https://it.emcelettronica.com/files/Schema_Esplora_bluetooth.zip
- [10] **programma Fritzing**: <http://http://fritzing.org/projects/>
- [11] Arduino Esplora: <http://arduino.cc/en/Main/ArduinoBoardEsplora#>
- [12] Display LCD TFT : <http://arduino.cc/en/Main/GTFT>
- [13] SparkFun codice PRT-10161: <https://www.sparkfun.com/products/10161>
- [14] Pololu con codice 791: <http://www.pololu.com/product/791>
- [15] codice RS 679-5375P: <http://it.rs-online.com/web/p/custodie-per-connettori-da-pcb/6795375P/>
- [16] vedere allegato:
https://it.emcelettronica.com/files/Schema_UNO_bluetooth.zip
- [17] Arduino UNO: <http://arduino.cc/en/Main/ArduinoBoardUno>
- [18] SparkFun codice WRL-10269: <https://www.sparkfun.com/products/10269>
- [19] listato completo: https://it.emcelettronica.com/files/test_master_code.zip
- [20] **listato completo**: https://it.emcelettronica.com/files/test_slave_code.zip

Copyright © 2017 Elettronica Open Source. All rights reserved.