

Aggiornamento

Interfacciamento di Arduino con uno smartphone

Il trasferimento di dati tra una scheda Arduino e uno smartphone (figura 1) può avvenire nelle due direzioni (smartphone-scheda e scheda-smartphone) grazie alla presenza di un modulo bluetooth (HC-05 o HC-06) che, interposto tra di essi, trasforma una porta seriale in una porta bluetooth.

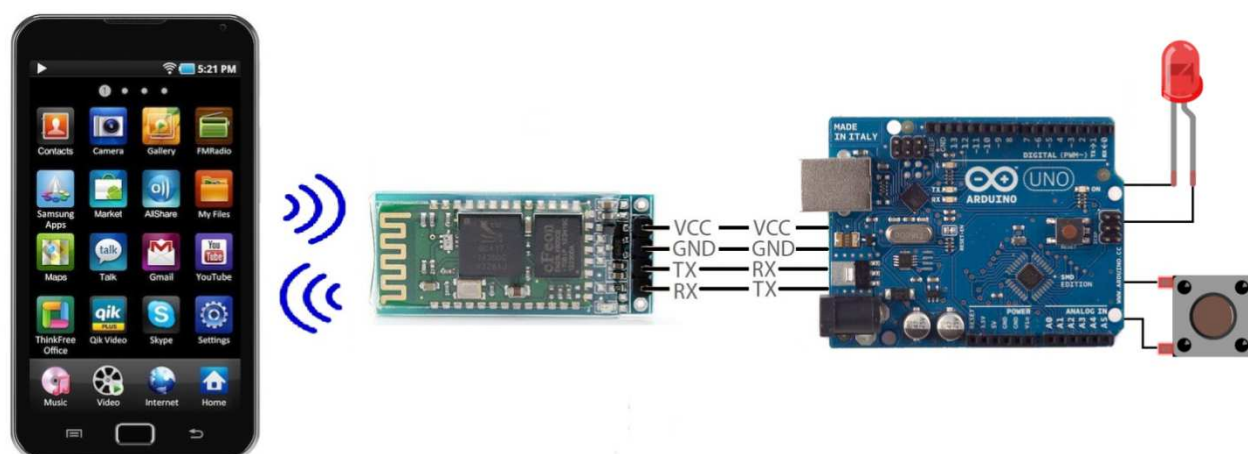


Figura 1 _ Trasferimento di dati tra una scheda Arduino e uno smartphone

Il modulo bluetooth viene utilizzato quando si vuole far comunicare un microprocessore (e di conseguenza anche una scheda Arduino) con dispositivi forniti di connessione bluetooth (computer, smartphone e tablet).

I quattro pin evidenziati in figura svolgono le seguenti funzioni:

- VCC e GND vengono utilizzati per l'alimentazione;
- RX e TX sono dedicati alla comunicazione seriale con Arduino, rispettivamente per la ricezione e per la trasmissione dei dati.

Il pin VCC deve essere collegato ai 5 V della scheda, il pin GND al GND della scheda.

Il pin di ricezione RX del modulo deve essere collegato al pin di trasmissione della scheda (il pin 1); viceversa il pin di trasmissione TX deve essere collegato al pin di ricezione della scheda (il pin 0).

Si precisa che durante la fase di caricamento del programma i pin 0 e 1 della scheda devono essere lasciati liberi; a caricamento completato devono essere collegati nel modo indicato in precedenza ai pin del modulo bluetooth.

Utilizzando la libreria *Software.Serial* si evita questo inconveniente.

L'istruzione *Software.Serial nome oggetto (rxPin,txPin)* consente infatti l'impiego di pin diversi da 0 e 1 per lo svolgimento delle funzioni di ricevitore e trasmettitore; un esempio di *nome oggetto* è *BTSerial*.

Per rendere possibile la trasmissione dei dati si deve scaricare sullo smartphone un'applicazione *open source* che consente di inviare delle stringhe via bluetooth; facendo riferimento a dispositivi che utilizzano il sistema operativo Android si può ad esempio utilizzare l'applicazione *Arduino bluetooth controller*.

Per poter procedere dal punto di vista operativo è in ogni caso necessario:

- alimentare il circuito;
- aprire l'applicazione;
- collegarsi al modulo bluetooth in una delle quattro modalità consentite (*joystick*, *interruttore*, *dimmer* o *terminale*).

In *modalità terminale* compare una tastiera che consente l'introduzione di caratteri alfanumerici.

Seguono alcuni esempi applicativi.

ESERCITAZIONE 1_

Scopo dell'esercitazione è l'invio da uno smartphone di un comando che, attraverso la scheda Arduino, determini l'accensione di un LED secondo le seguenti modalità:

- inviando il carattere a si accende il LED collegato al pin 13;
- inviando il carattere b si accende il LED collegato al pin 13;
- inviando il carattere c il LED collegato al pin 13 si accende a intermittenza ad intervalli di 1 s per 10 volte (ciclo for), di seguito si spegne;
- inviando il carattere d il LED collegato al pin 13 si accende a intermittenza ad intervalli di 1 s per 10 volte (ciclo for), di seguito rimane acceso.

Dopo aver effettuato il collegamento al modulo bluetooth in *modalità terminale* si verifica che, digitando sulla tastiera dello smartphone uno dei quattro caratteri, il LED si comporta di conseguenza nel modo desiderato; inoltre sullo schermo dello smartphone viene visualizzato lo stato del LED (osservabile anche sul monitor seriale).

Il listato del programma viene di seguito proposto.

La lettura dei caratteri digitati sulla tastiera dello smartphone avviene utilizzando l'istruzione *Serial.read*, la scrittura dello stato del LED sulla tastiera con *Serial.println*.

```
char dato;
int l;
void setup(){
  Serial.begin(9600); // impostazione velocità di comunicazione seriale
  pinMode(13,OUTPUT); // definizione pin 13 come uscita
}
void loop(){
  dato=Serial.read(); // lettura dato impostato tramite smartphone
  if(dato=='a'){ // digitando a
    digitalWrite(13,HIGH); // il LED si accende
    Serial.println("ON"); // scrivi sullo smartphone
  }
  if(dato=='b') { // digitando b
    digitalWrite(13,LOW); // il LED si spegne
    Serial.println("OFF"); // scrivi sullo smartphone
  }
  if(dato=='c'){
```

```

// digitando c il LED si accende a intermittenza per 10 cicli e poi si spegne
Serial.println("blink"); // scrivi sullo smartphone
for(l=1;l<=10;l++) {
digitalWrite(13,HIGH); // il LED si accende per 1 s
delay(1000);
digitalWrite(13,LOW); // il LED si spegne per 1 s
delay(1000);
}
Serial.println("OFF"); // scrivi sullo smartphone
}
if(dato=='d'){
// digitando d il LED si accende a intermittenza per 10 cicli e poi rimane acceso
Serial.println("blink"); // scrivi sullo smartphone
for(l=1;l<=10;l++) {
digitalWrite(13,HIGH); // il LED si accende per 1 s
delay(1000);
digitalWrite(13,LOW); // il LED si spegne per 1 s
delay(1000);
}
digitalWrite(13,HIGH); // il LED rimane acceso
Serial.println("ON"); // scrivi sullo smartphone
}
}
}

```

ESERCITAZIONE 2

Scopo dell'esercitazione è l'invio da uno smartphone di un comando che, attraverso la scheda Arduino, determini l'accensione di un LED secondo le seguenti modalità:

- inviando il carattere a si accende il LED collegato al pin 13;
- inviando il carattere s si accende il LED collegato al pin 13;
- inviando il carattere 1 il LED collegato al pin 13 si accende a intermittenza ad intervalli di 1 s;
- inviando il carattere 3 il LED collegato al pin 13 si accende a intermittenza ad intervalli di 3 s;
- inviando il carattere 5 il LED collegato al pin 13 si accende a intermittenza ad intervalli di 5 s.

Dopo aver effettuato il collegamento al modulo bluetooth in *modalità terminale* si verifica che, digitando uno dei cinque caratteri, il LED si comporta di conseguenza nel modo desiderato.

In questo caso viene utilizzata la libreria *SoftwareSerial* (per includere la libreria si utilizza l'istruzione `#include<SoftwareSerial.h>`).

In questo vengono utilizzati per la trasmissione e la ricezione dei caratteri i pin contrassegnati con 8 e 9.

Il listato del programma viene di seguito proposto.

La lettura dei caratteri digitati sulla tastiera dello smartphone avviene utilizzando l'istruzione *BTSerial.read*, con *BTSerial.available* si può verificare la disponibilità dei dati.

```

#include<SoftwareSerial.h> // si include la libreria
char dato;
SoftwareSerial BTSerial(8,9);
// viene definito l'oggetto BTSerial
// definizione pin 8 e 9 come RX e TX
// devono essere collegati rispettivamente ai pin TX e RX del modulo
void setup(){
BTSerial.begin(9600); // definizione velocità di trasmissione
pinMode(13,OUTPUT); // definizione pin 13 come uscita
}
void loop(){
if(BTSerial.available()){ // se ci sono dati disponibili
dato=BTSerial.read(); // viene letto il carattere digitato sullo smartphone
}
if(dato=='a'){ // se viene letto il carattere a
digitalWrite(13,HIGH); // il LED si accende
}
if(dato=='s'){ // se viene letto il carattere s
digitalWrite(13,LOW); // il LED si spegne
}
if(dato=='1'){ // se viene letto il carattere 1
digitalWrite(13,HIGH); // il LED si accende per 1 s
delay(1000);
digitalWrite(13,LOW); // il LED si spegne
delay(1000);
}
if(dato=='3'){ // se viene letto il carattere 3
digitalWrite(13,HIGH); // il LED si accende per 3 s
delay(3000);
digitalWrite(13,LOW); // il LED si spegne per 3 s
delay(3000);
}
if(dato=='5'){ // se viene letto il carattere 5
digitalWrite(13,HIGH); // il LED si accende per 5 s
delay(5000);
digitalWrite(13,LOW); // il LED si spegne per 5 s
delay(5000);
}
}
}

```

ESERCITAZIONE 3

Scopo dell'esercitazione è di contare il numero di impulsi che si verificano premendo il pulsante collegato al pin 2 della scheda; tra due impulsi successivi deve trascorrere più di 1/2 s; al decimo impulso il conteggio deve essere azzerato e resettato.

Dopo aver effettuato il collegamento al modulo bluetooth in *modalità terminale* il numero di impulsi deve essere visualizzato progressivamente sullo schermo di uno

smartphone; il conteggio viene indicato con numeri compresi tra 0 (primo impulso) e 9 (ultimo impulso).

Il LED del pin 13 si accende quando si verificano i primi cinque impulsi della sequenza; al sesto impulso si spegne (e rimane spento fino al decimo).

Il listato del programma viene di seguito proposto.

```
int statoPB,l=0;
void setup(){
  Serial.begin(9600); // velocità di comunicazione seriale
  pinMode(2,INPUT); // il pin 2 viene definito come ingresso per il pulsante
  pinMode(13,OUTPUT); // il pin 13 viene definito come uscita per il LED
}
void loop(){
  statoPB=digitalRead(2); // lettura stato pulsante
  if(statoPB==HIGH){ // se viene letto un livello alto
    Serial.println(l); // viene visualizzato sullo schermo il valore attuale
    l=l+1; // viene incrementato il valore della variabile contatore
    delay(500); // tempo d'attesa tra due impulsi
  }
  if((l-1)==0){ // quando il conteggio vale 0
    digitalWrite(13,HIGH); // si accende il LED del pin 13
  }
  if((l-1)==5){ // quando il conteggio vale 5
    digitalWrite(13,LOW); // si spegne il LED del pin 13
  }
  if(l==10){ // quando il contatore raggiunge il valore finale
    l=0; // avviene il reset
  }
}
```