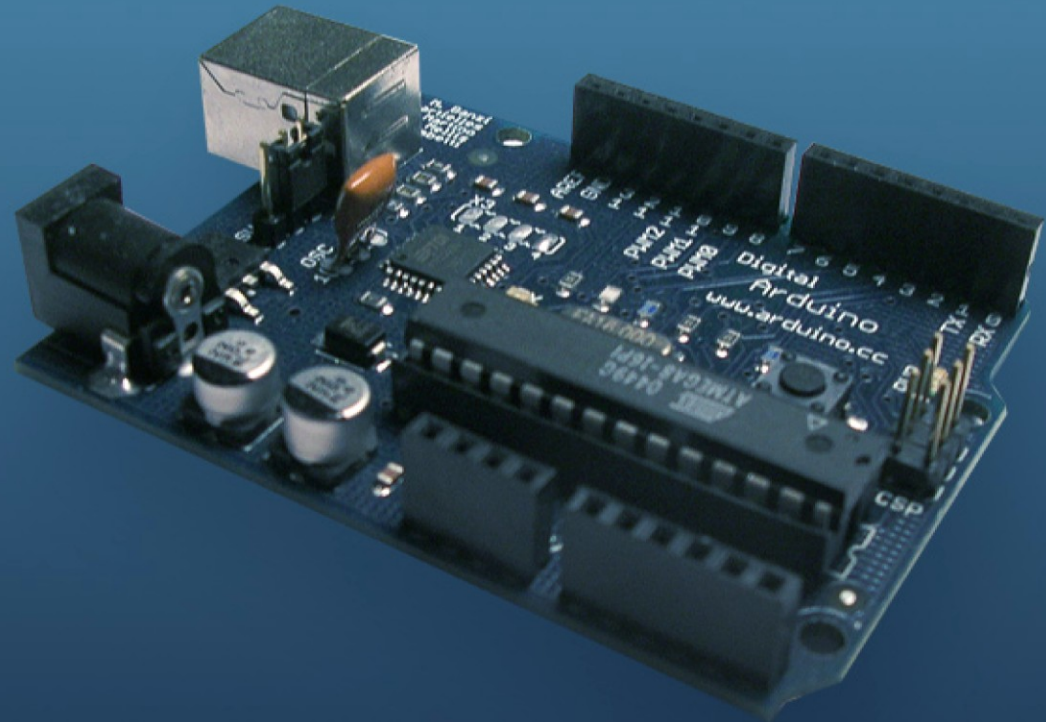


**Arduino**  
Physical Computing I/O board



17<sup>^</sup> parte : **Utilizzare il modulo Bluetooth JY-MCU**



Author: Ing. Sebastiano Giannitto (ITIS "M.BARTOLO" –PACHINO)



Premessa :

I piedini 0 ed 1 sono quelli collegati al convertitore Seriale <-> USB ... quindi, se si usa la USB per connetterti al PC o a uno smatphone è meglio **NON** usare i pin 0 e 1 in nessun caso, dato che, se poi si deve fare il debug, **NON** avremmo più la porta seriale, ossia rischiamo di bloccare ARDUINO e non poter effettuare più l'upload di qualsiasi altro sketch.

Il modulo bluetooth si chiama HC-06 oppure JY-MCU ( potete trovarne uno [cliccando qui](#) )



## Ricevere messaggi da uno smartphone Android

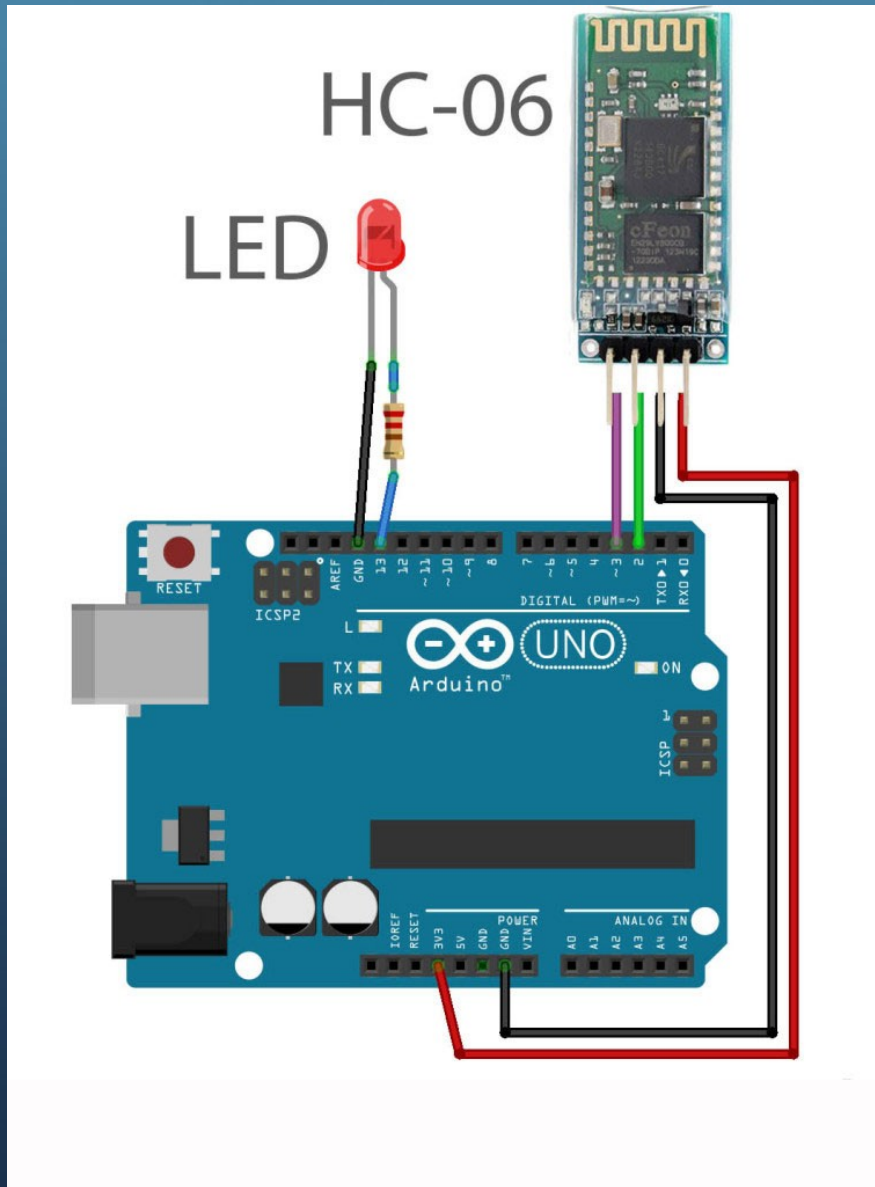
Con questo sketch visualizzerete sul monitor seriale di Arduino le stringhe di caratteri inviate al modulo bluetooth tramite uno Smartphone o un PC. Sul web esistono tantissime App in grado di inviare stringhe tramite bluetooth, per Android consiglio questa [App](#) (**Arduino Bluetooth Controller**). Mentre per iOS, purtroppo, il modulo HC-06 non viene riconosciuto.

```
#include <SoftwareSerial.h>
int rxPin = 2; // attenzione il TX dell'HC06 va collegato al pin che fa da ricevitore di Arduino
int txPin = 3;
SoftwareSerial bluetooth(rxPin, txPin);
String message; //string that stores the incoming message

void setup()
{ Serial.begin(9600); //set baud rate
  bluetooth.begin(9600); //set baud rate }

void loop()
{ while(bluetooth.available()) {
  message+=char(bluetooth.read()); }
  if (!bluetooth.available())
  { if (message!="")
    { //if data is available
      Serial.println(message); //show the data
      message=""; //clear the data }
    }
  delay(5000); //delay
}
```

## Accendiamo un led da uno smartphone Android



Il modulo bluetooth si chiama HC-06 oppure JY-MCU ( potete trovarne uno [cliccando qui](#))

Caricare questo semplice sketch, ricordandosi di tenere scollegato il modulo bluetooth durante il caricamento altrimenti l'IDE darà errore.

```
#include <SoftwareSerial.h>

int rxPin = 2;
int txPin = 3;
SoftwareSerial bluetooth(rxPin, txPin);

String message; //string that stores the incoming
message
const int Led = 13;

void setup()
{
  Serial.begin(9600); //set baud rate
  bluetooth.begin(9600); //set baud rate
  pinMode(Led,OUTPUT);
}

void loop()
{
  while (bluetooth.available())
  {
    message+=char(bluetooth.read());
  }

  if (!bluetooth.available())
  {
    if(message!="")
    { // if data is available
      if(message == "H")
      {
        digitalWrite(Led, HIGH);
        Serial.println("Led ON"); //show the data
        delay(20);
        message=""; //clear the data
      }
      else if (message == "L")
      {
        digitalWrite(Led, LOW);
        Serial.println("Led OFF"); //show the data
        delay(20);
        message=""; //clear the data
      }
    }
  }
}
```

Il funzionamento è semplice, quando il modulo bluetooth riceve il carattere **H** l'uscita 13 viene posta su HIGH, mentre se si riceve il carattere **L** l'uscita 13 viene posta su LOW.

Ora dobbiamo scaricare sul nostro smartphone un'applicazione che ci consente di inviare delle stringhe via bluetooth.

Possiamo usare "[blueserial](#)", oppure qualcosa di equivalente.

Alimentate il circuito, aprite l'app e collegatevi al modulo bluetooth (se vi chiede una passkey provate 1234).

Ora inviate H e il led si accenderà, con L lo spegnerete.

Naturalmente qui accendiamo un semplice led, ma con pochi componenti potremo ad esempio accendere una presa di corrente o aprire la porta del garage.

Se abbiamo la necessita di cambiare il Nome (SSID) del bluetooth e il Pincode è sufficiente caricare lo sketch seguente su Arduino (utilizzando gli stessi collegamenti visti in precedenza) e utilizzare il serial monitor di Arduino per configurare il vostro HC-06.

Si possono utilizzare i seguenti comandi:

**AT** -> Se la comunicazione funziona il modulo risponde OK

**AT+VERSION** -> Restituisce la versione del firmware

**AT+BAUDx** -> Imposta il Baudrate, al posto di x mettere 1 per 1200 bps, 2=2400, 3=4800, 4=9600, 5=19200, 6=38400, 7=57600, 8=115200, 9=230400, A=460800, B=921600, C=1382400

**AT+NAMEstring** -> Al posto di string mettere il nome che vuoi dare al modulo (massimo 20 caratteri)

**AT+PINxxx** -> Imposta il pincode del modulo bluetooth (es.1234)

```
#include <SoftwareSerial.h>
int rxPin = 2;
int txPin = 3;
SoftwareSerial bluetooth(rxPin, txPin);

String message; //string that stores the incoming message

void setup()
{
  Serial.begin(9600);
  bluetooth.begin(9600);
  Serial.println("Lista dei comandi HC-06:\n");
  Serial.println("AT          Se la comunicazione funziona il modulo risponde OK");
  Serial.println("AT+VERSION    Restituisce la versione del firmware");
  Serial.println("AT+BAUDx     Imposta il Baudrate, al posto di x mettere 1 per 1200 bps,
2=2400, 3=4800, 4=9600, 5=19200, 6=38400, 7=57600, 8=115200, 9=230400, A=460800,
B=921600, C=1382400");
  Serial.println("AT+NAMEstring Al posto di string mettere il nome che vuoi dare al modulo
(massimo 20 caratteri)");
  Serial.println("AT+PINxxxx   Imposta il pincode del modulo bluetooth (es.1234)");
}
```

```
void loop()
{
  if (bluetooth.available())
  {
    Serial.write(bluetooth.read());
  }
  if (Serial.available())
  {
    bluetooth.write(Serial.read());
  }
}
```

## ARDUINO MEGA

Come sappiamo Arduino Mega possiede **4 UARTs** (Serial, Serial1, Serial2, Serial3) quindi utilizzando il modulo bluetooth con la Mega board si può fare a meno della libreria ***SoftwareSerial.h***.

Collegate il modulo HC-06 ad Arduino Mega in questo modo:

HC-06		MEGA
VCC	->	+3.3V
GND	->	GND
TX	->	15
RX	->	14



Questo sketch ha la stessa funzione del primo, ossia quello di mandare messaggi da uno smartphone

```
String message; //string that stores the incoming message
void setup()
{
  Serial.begin(9600); //set baud rate
  Serial3.begin(9600); //set baud rate
}
void loop()
{
  while(Serial3.available()){
    message+=char(Serial3.read());
  }
  if(!Serial3.available())
  {
    if(message!="")
    {
      //if data is available
      Serial.println(message); //show the data
      message=""; //clear the data
    }
  }
  delay(5000); //delay
}
```

Lo sketch seguente ha lo scopo di far accendere e spegnere un led tramite bluetooth.

```
String message; //string that stores the incoming
message
const int Led = 13;

void setup()
{
  Serial.begin(9600); //set baud rate
  Serial3.begin(9600); //set baud rate
  pinMode(Led,OUTPUT);
}

void loop()
{
  while(Serial3.available()){
    message+=char(Serial3.read());
  }
  If (!Serial3.available())
  {
    if(message!="")
    {//if data is available
      if(message == "H"){
        digitalWrite(Led, HIGH);
        Serial.println("Led ON"); //show the data
        delay(20);
        message=""; //clear the data
      }
      else if(message == "L"){
        digitalWrite(Led, LOW);
        Serial.println("Led OFF"); //show the data
        delay(20);
        message=""; //clear the data
      }
    }
  }
}
```

Mentre questo è lo sketch per il cambio del PINCODE e del SSID:

```
String message; //string that stores the incoming message
```

```
void setup()
```

```
{  
  Serial.begin(9600);  
  Serial3.begin(9600);  
  Serial.println("Lista dei comandi HC-06:\n");  
  Serial.println("AT          Se la comunicazione funziona il modulo risponde OK");  
  Serial.println("AT+VERSION  Restituisce la versione del firmware");  
  Serial.println("AT+BAUDx    Imposta il Baudrate, al posto di x mettere 1 per  
1200 bps, 2=2400, 3=4800, 4=9600, 5=19200, 6=38400, 7=57600, 8=115200,  
9=230400, A=460800, B=921600, C=1382400");  
  Serial.println("AT+NAMEstring Al posto di string mettere il nome che vuoi dare  
al modulo (massimo 20 caratteri)");  
  Serial.println("AT+PINxxxx  Imposta il pincode del modulo bluetooth  
(es.1234)"); }  
}
```

```
void loop()
```

```
{  
  if (Serial3.available())  
  { Serial.write(Serial3.read()); }  
  if (Serial.available())  
  { Serial3.write(Serial.read()); }  
}
```