

# Gli approfondimenti di

# L@BOROBOTICA

## ESP 32 Bluetooth

L'ESP32 WROOM è un modulo molto potente che include, oltre alle funzionalità Wi-Fi anche quelle Bluetooth, BLE. Integra inoltre una notevole quantità di periferiche, che vanno dai sensori capacitivi ai sensori di Hall, senza dimenticare l'SPI ad alta velocità e l'I2C.

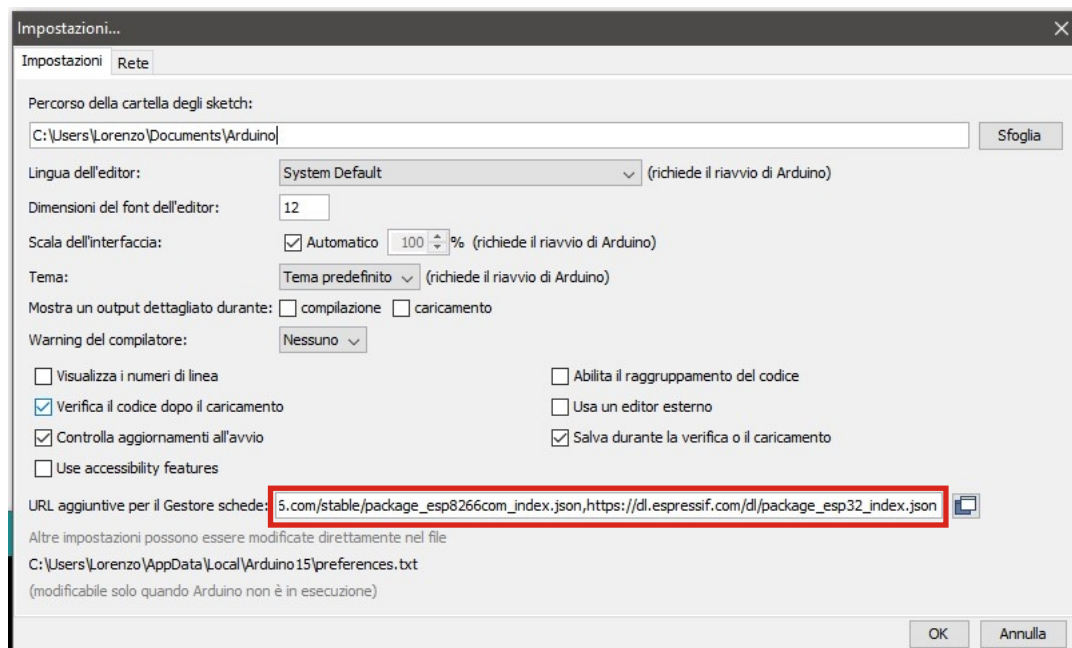
In queste schede proveremo ad utilizzarlo come modulo Bluetooth, al posto del famoso HC05, ampiamente descritto in Laborobotica Vol.2.

ESP32 ha due CPU che possono essere controllate individualmente, compreso un coprocessore che può essere utilizzato in applicazioni low-power (risparmio d'energia).

Proviamo, utilizzando uno sketch degli esempi, a impostare un collegamento bidirezionale tra la scheda ESP32 e un cellulare sul quale è installata un'apposita applicazione. Quella consigliata è Blue Serial Beta, che si può scaricare da Google Play.

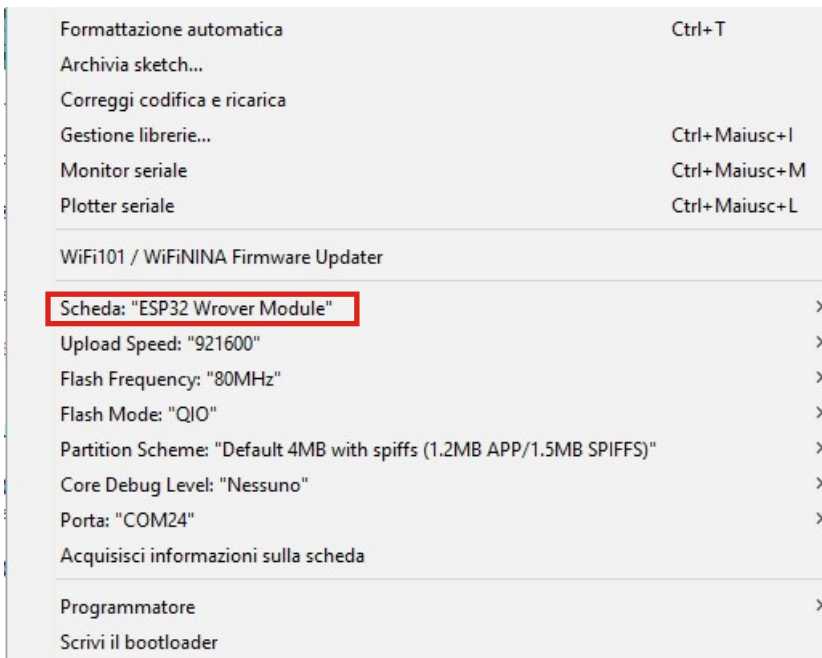


Per cominciare dovremo accertarci che nell'IDE di Arduino sia installata la scheda ESP32 WROOM.



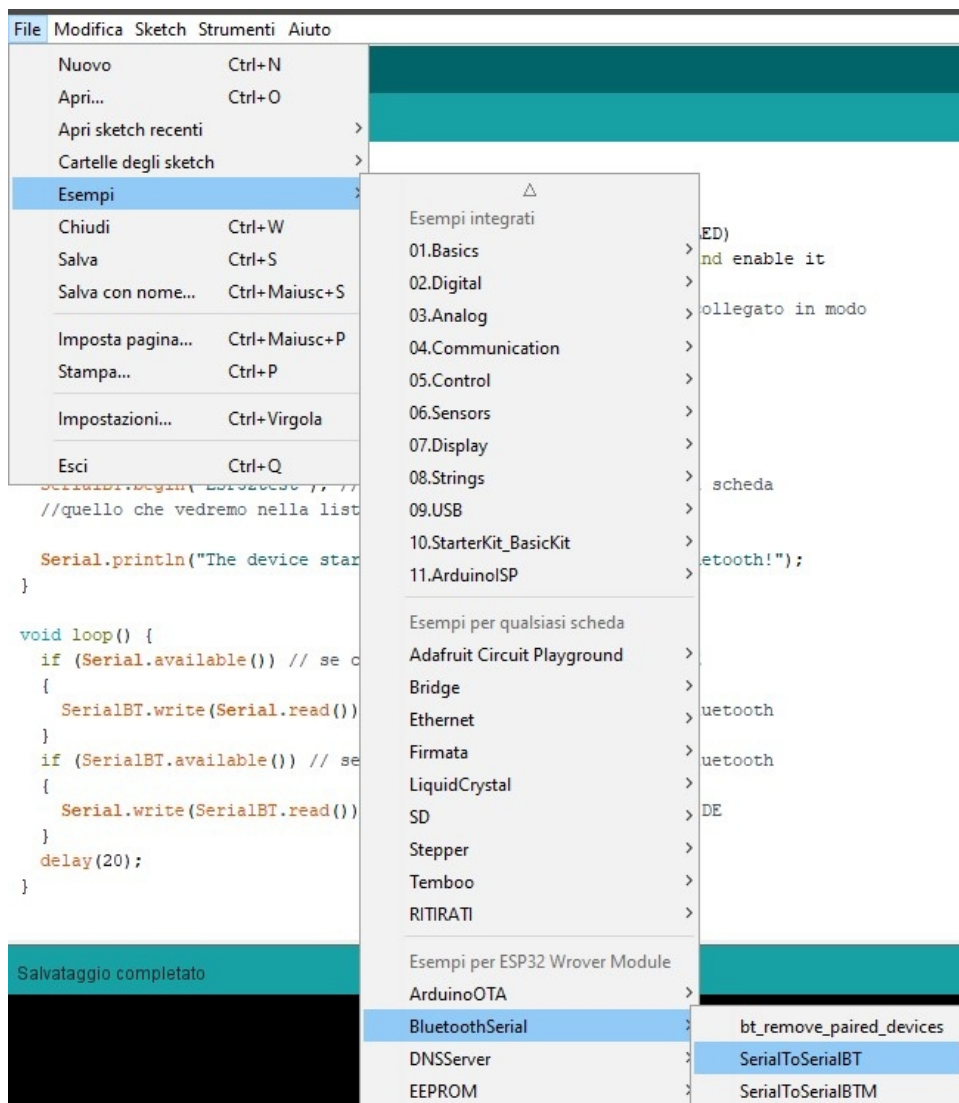
In File> Impostazioni dare l'indirizzo per poter caricare la scheda che ci interessa.

[https://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](https://arduino.esp8266.com/stable/package_esp8266com_index.json),[https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json)



Dopo aver installato la scheda (in gestore schede come descritto in Laborobotica Vol.2) la selezioniamo per il caricamento dello sketch.

A questo punto, dagli esempi relativi alla scheda ESP32 scegliamo BluetoothSerial > Serial to SerialBT.

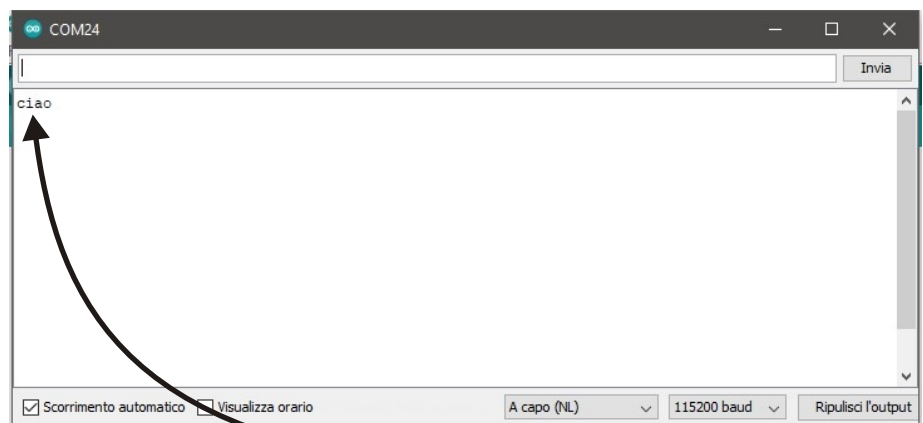
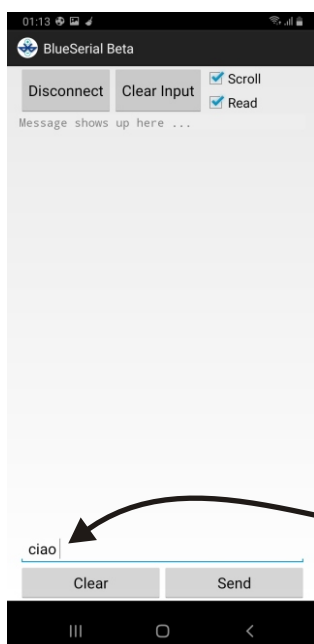


## COMUNICAZIONE\_BIDIREZIONALE

```
1
2 #include "BluetoothSerial.h" //include la libreria
3
4 #if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
5 #error Bluetooth is not enabled! Please run `make menuconfig` to and enable it
6 #endif
7 //queste tre linee diprogramma si accertano che il bluetooth sia collegato in modo
8 //corretto
9 BluetoothSerial SerialBT; //crea un'istanza denominata SerialBT
10
11 void setup() {
12   Serial.begin(115200); //stabilisce la velocità di comunicazione
13
14   SerialBT.begin("ESP32test"); //dà il nome di riconoscimento alla scheda
15   //quello che vedremo nella listadei dispositivi bluetooth
16
17   Serial.println("The device started, now you can pair it with bluetooth!");
18 }
19
20 void loop() {
21   if (Serial.available()) // se c'è un dato sulla seriale dell'IDE
22   {
23     SerialBT.write(Serial.read()); // scrivilo sulla seriale del bluetooth
24   }
25   if (SerialBT.available()) // se c'è un dato sulla seriale del bluetooth
26   {
27     Serial.write(SerialBT.read()); // scrivilo sulla seriale dell'IDE
28   }
29   delay(20);
30 }
```

Questo è lo sketch commentato. Carichiamolo sulla scheda avendo cura di tenere premuto il pulsante "BOOT". Lanciamo l'applicazione, facciamo il pairing, scegliendo nella lista dei dispositivi Bluetooth presenti "ESP32test" (il nome assegnatogli nello sketch).

Vedremo che scrivendo dei caratteri nella riga di comando del monitor dell'IDE di Arduino, li ritroveremo sul monitor dell'applicazione; scrivendoli sul monitor dell'applicazione li vedremo nell'IDE di Arduino.



Li ritroveremo sul monitor dell'IDE

Scrivendo dei caratteri sull'app