

## Bluetooth Shield User Guide

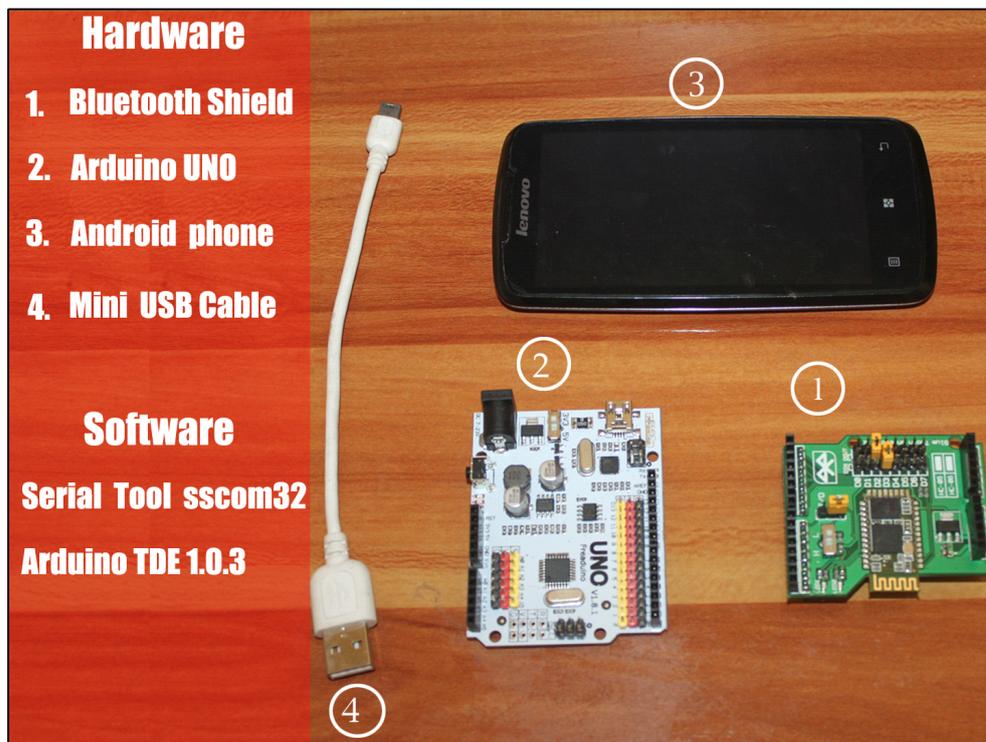
### Bluetooth Shield Introduction

Bluetooth Shield is a serial port Bluetooth module (Master/Slave) breakout board. You can choose ARDUINO's Digital PIN D0->D7 as Software UART to communicate with Bluetooth Shield(and D0,D1 is also Hardware UART).

There is a HC-05 on the board and it's an easy to use Bluetooth SPP module designed for transparent wireless serial connection setup.

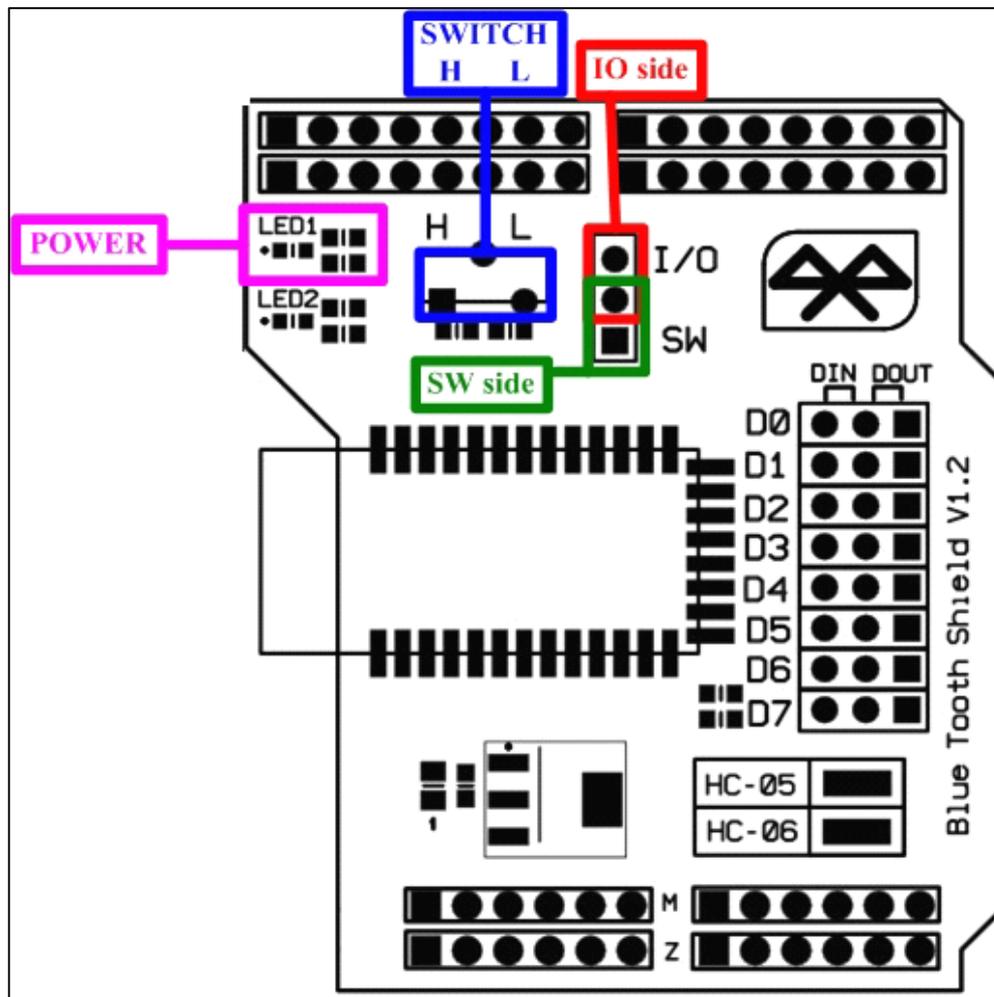
It can be configured to three different modes, Master Mode, Slave Mode or Loopback Mode respectively, and it will connect to or be connected by other devices that support SPP protocol per configuration, and it is a suitable substitute for most applications.

### Hardware and Software Preparation

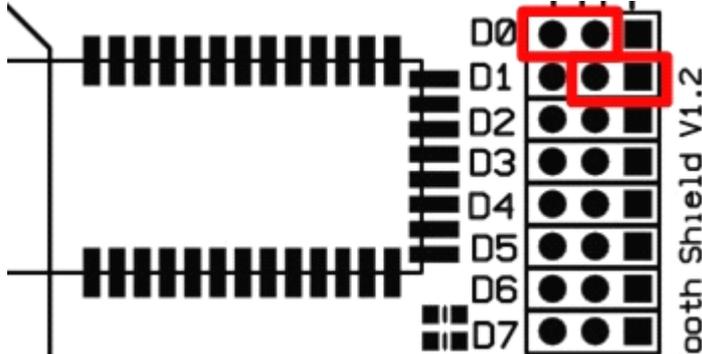


## Part 1 Communication Between Arduino and Bluetooth

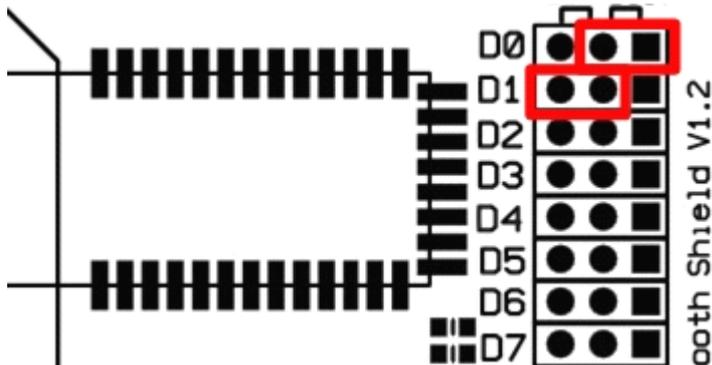
- Assemble the Arduino UNO and Bluetooth Shield, and then connect to PC.
- Set the switch in H port to enter AT mode, and the LED1 blink slowly in AT mode. In addition, we default the short caps on SW side.



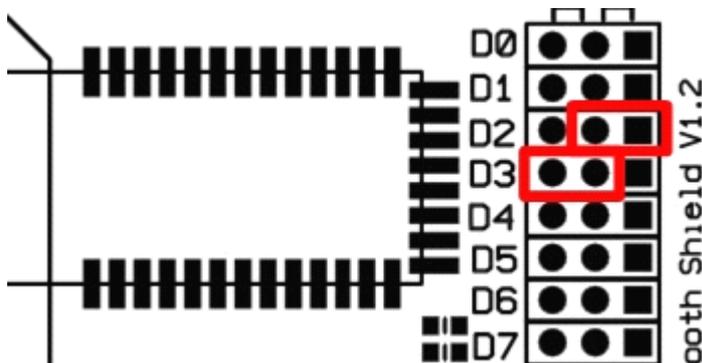
Case 1. Directly connect to Arduino main board Serial port, DIN to RX\_H(D0) , DOUT to TX\_H(D1) but it need you remove the MCU chip.



Case 2. Setting the Bluetooth shield's jumper connect: DIN to TX\_H(D1) , DOUT to RX\_H(D0). So this way just for you has perfect code and don't need monitor from serial port, because you can't see anything from serial port tool.



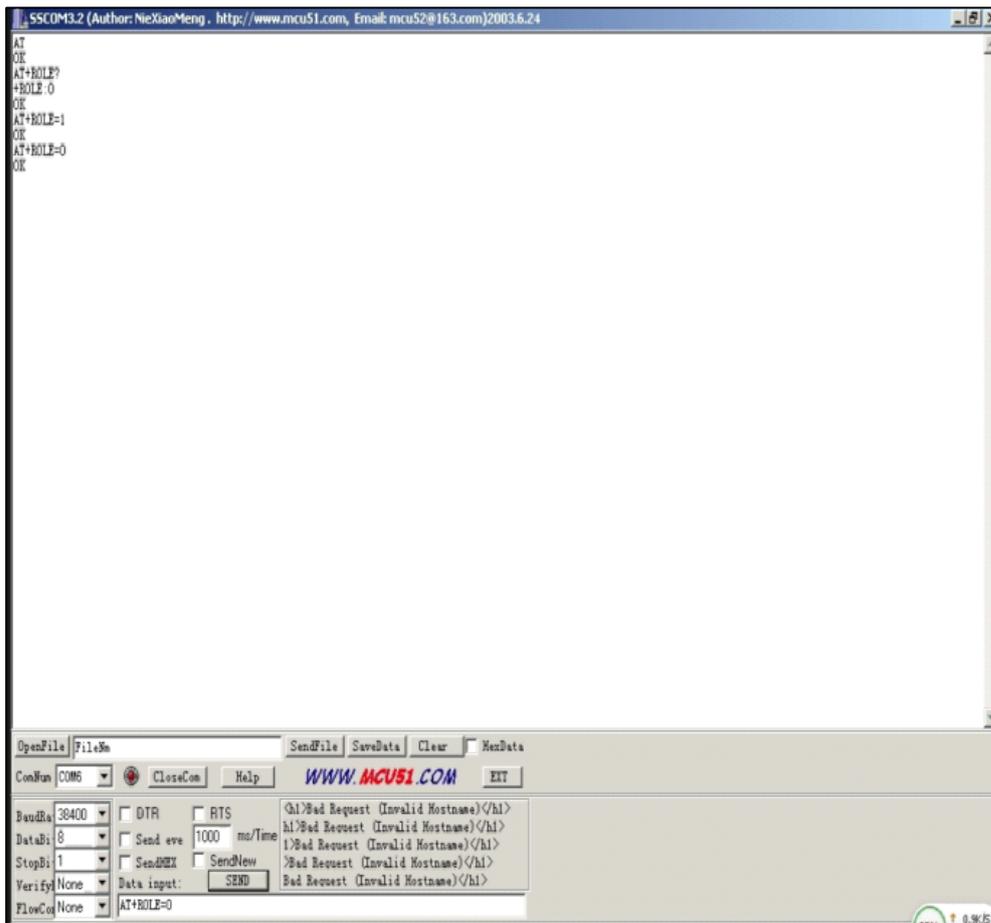
Case 3. Use Software Serial library to define D2 and D3(Of course others) as a virtual serial port TX and RX. By this way, you can use Arduino's Serial Monitor to Monitor operation of the code.



- We default the short caps on D2/D3 (Case 3), but if you use other pins, for example, D6/D7, remember to change code define to D6/D7.

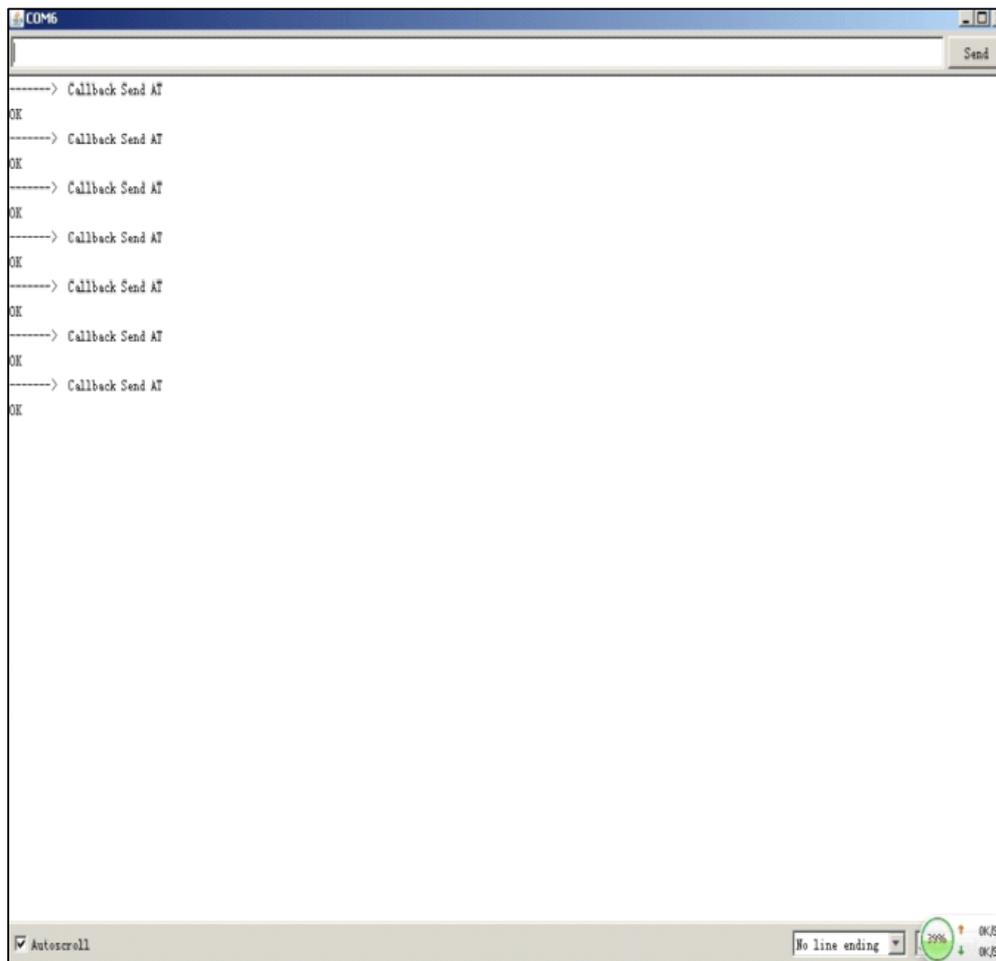
## Step 1. Achieve Manually AT Command Control

1. Open the Arduino IDE 1.0.X, and copy the code from ElecFreaks wiki to it.
2. Import library of SoftwareSerial, because library of NewSoftSerial is just suitable of edition before Arduino IDE 1.0, such as IDE0022, IDE0023.
3. Delete the library of NewSoftSerial and modify the NewSoftSerial into SoftwareSerial.
4. Download TimerOne from <http://code.google.com/p/arduino-timerone/downloads/list>
5. Compiling sketch until Done compiling appears
6. Upload code until Done uploading appears
7. Open sscm32 and set BaudRate as 38400
8. Manually input AT and the shield would respond OK
9. Input AT+ROLE? and the shield respond +ROLE=0, therefore, We default the Bluetooth Shield as SLAVE Mode
10. Input AT+ROLE=1, we can set it as Master Mode
11. Input AT+ROLE=0, we can set it back as SLAVE Mode



## Step 2. Achieve Automatically AT Command Control

1. Modify the code (Remove // before Timer1) and then upload it again in Arduino IDE 1.0.X
2. The shield can automatically respond OK for AT Command



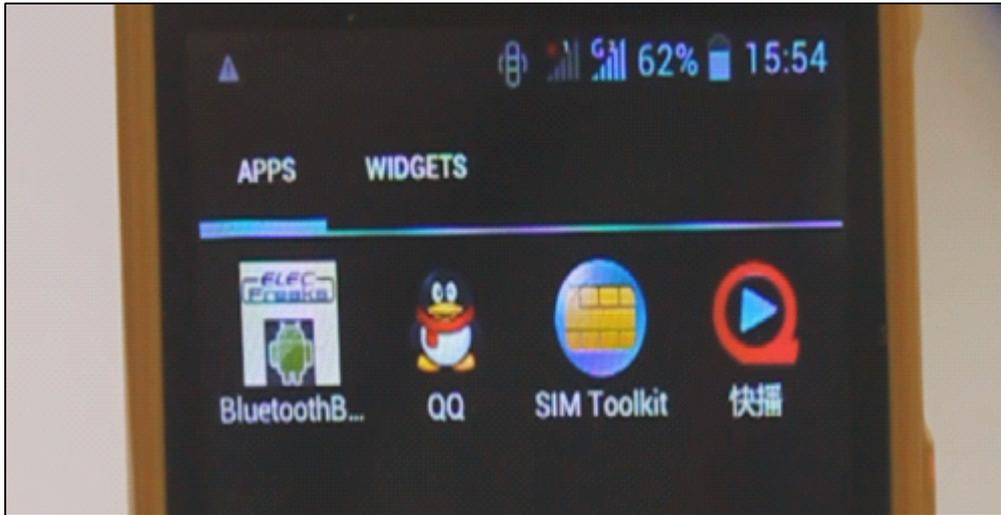
The screenshot shows a serial monitor window titled 'COM6'. The window contains a series of ten lines of text, each consisting of a right-pointing arrow followed by the text 'Callback Send AT'. Below each of these lines is the text 'OK'. The window has a 'Send' button in the top right corner. At the bottom of the window, there is a status bar with a checked 'Autoscroll' checkbox, a 'No line ending' dropdown menu, and a green circular progress indicator showing 20% with 'OK/E' labels.

## Part 2. Communication Between Android and Arduino with Bluetooth

- Set the Switch in L port to enter Communication Mode. In Communication Mode, make sure the LED1 blink quickly. If necessary, re-plug the USB Cable.
- Modify the code (add // before Timer1, especially replace the BaudRate from 38400 to 9600 ), and then upload the code again

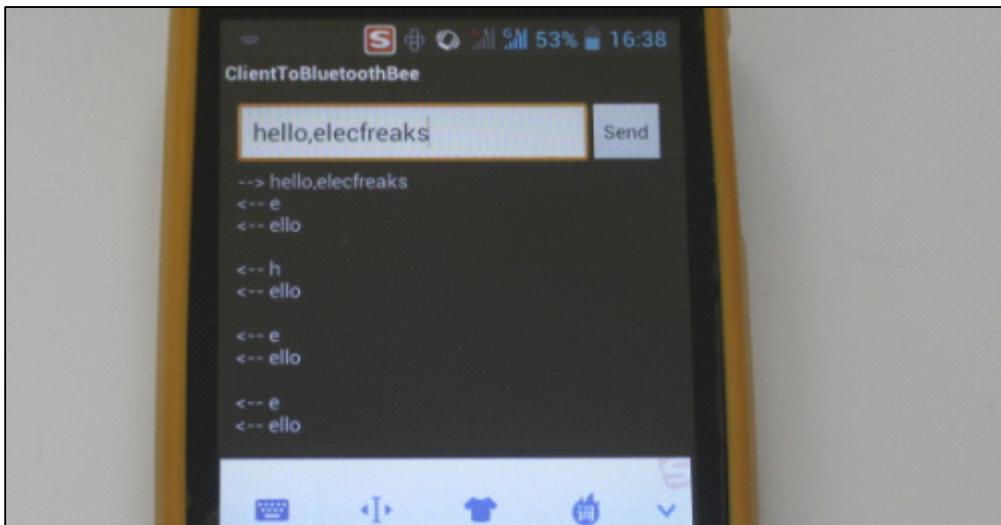
## Step 1. Android Installation

1. Download APP for EF \_BluetoothBeeV1.1 from [http://electfreaks.com/store/download/datasheet/Bluetooth/EF\\_BluetoothBeeV1.1.zip](http://electfreaks.com/store/download/datasheet/Bluetooth/EF_BluetoothBeeV1.1.zip)
2. Install the APK file to your Android phone.



## Step 2. Android Communication

1. Click the icon of ElecFreaks BluetoothBee as above and then open Bluetooth.
2. Click As Client and start scanning
3. When it scan out HC-05, click HC-05.
4. Type the Bluetooth pairing request 1234.
5. Input hello,electfreaks in mobile, and the PC would display hello,electfreaks from mobile. (For sscm32, remember to set the BaudRate as 9600.)
6. Input hello from PC, mobile would also respond hello.



## Q&A

---

If you use Arduino Mega1280/250 or Freaduino ADK there are some different. Please note the NewSoftSerial library about Mega2560 explanation in SoftwareSerial.cpp :

```
// Specifically for the Arduino Mega 2560 (or 1280 on the original Arduino Mega)
// majority of the pins are NOT PCINTs, SO BE WARNED (i.e. you cannot use them as receive pins)
// Only pins available for RECEIVE (TRANSMIT can be on any pin):
// (I've deliberately left out pin mapping to the Hardware USARTs - seems senseless to me)
// Pins: 10, 11, 12, 13, 50, 51, 52, 53, 62, 63, 64, 65, 66, 67, 68, 69
```

That's mean the library do not support D0-D7 as receive pins, just could use pins:10, 11, 12, 13, 50, 51, 52, 53, 62, 63, 64, 65, 66, 67, 68, 69 for receive. So there are two way to resolve it, but all need external jumper wires.

**First method:** Change the define of rxPin and txPin. Jumper wires connect DOUT-D10, DIN-D11.

```
#define rxPin 10
#define txPin 11
```

**Second method:** Use the others Hardware Serialport because of Mage there are 4 hardware serialport. Jumper wires connect DOUT-RX1(D19) DIN-TX1(D18)

