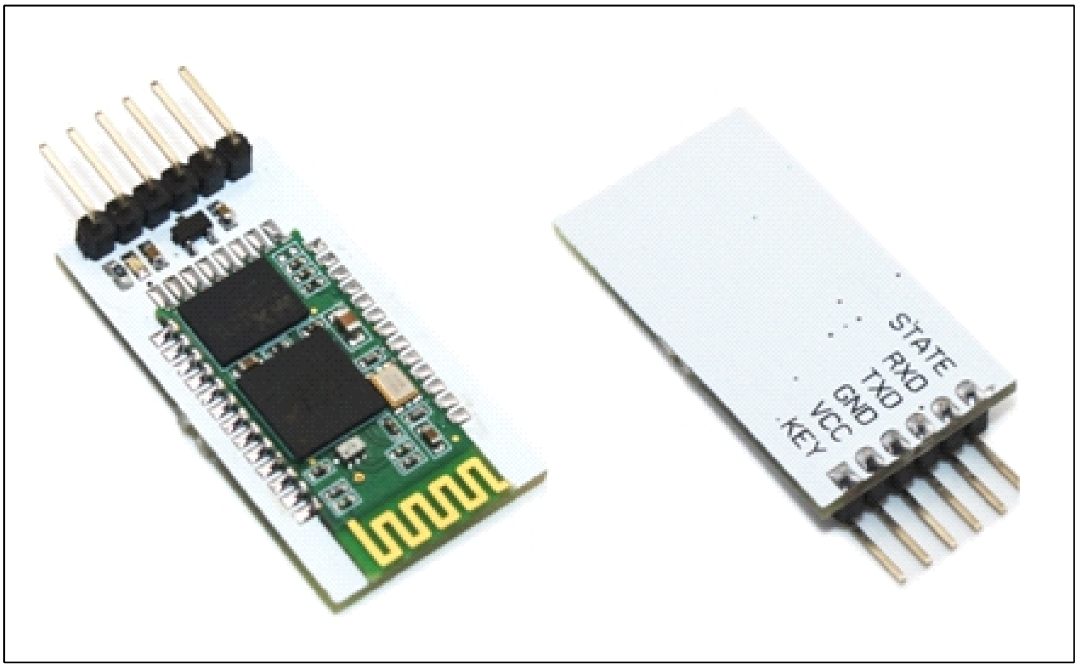
Bluetooth Modem User Guide

Bluetooth Modem Introduction

The **Bluetooth Modem** is the latest Bluetooth wireless serial cable! This version of the popular Bluetooth uses the HC-05/HC-06 module. These modems work as a serial (RX/TX) pipe. Any serial stream from 9600 to 115200bps can be passed seamlessly from your computer to your target.

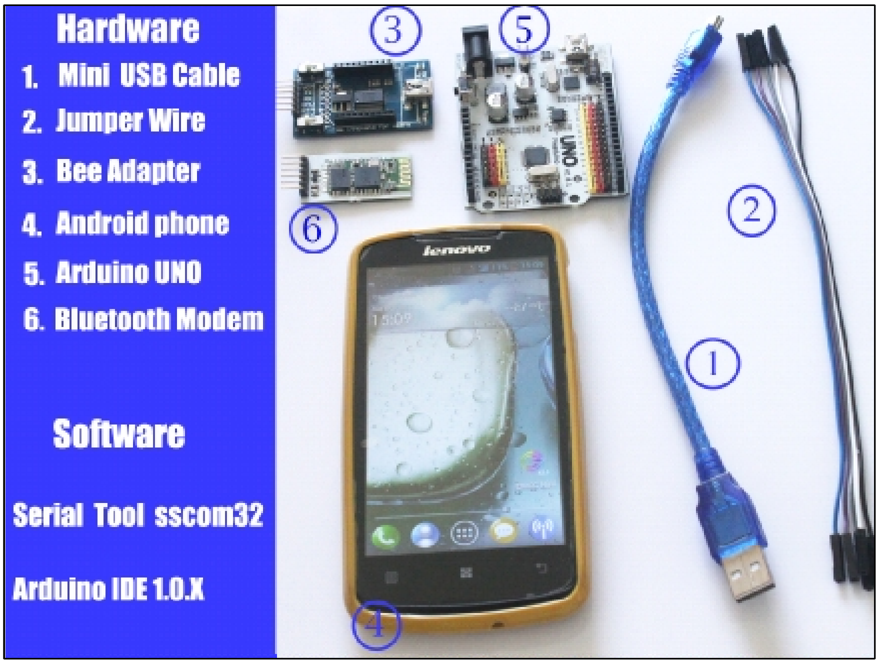
The remote unit can be powered from **3.3V up to 6V** for easy battery attachment. All signal pins on the remote unit are 3V-6V tolerant. No level shifting is required. **Do not** attach this device directly to a serial port. You will need an RS232 to TTL converter circuit if you need to attach this to a computer. You can either solder a 6-pin header or individual wires.



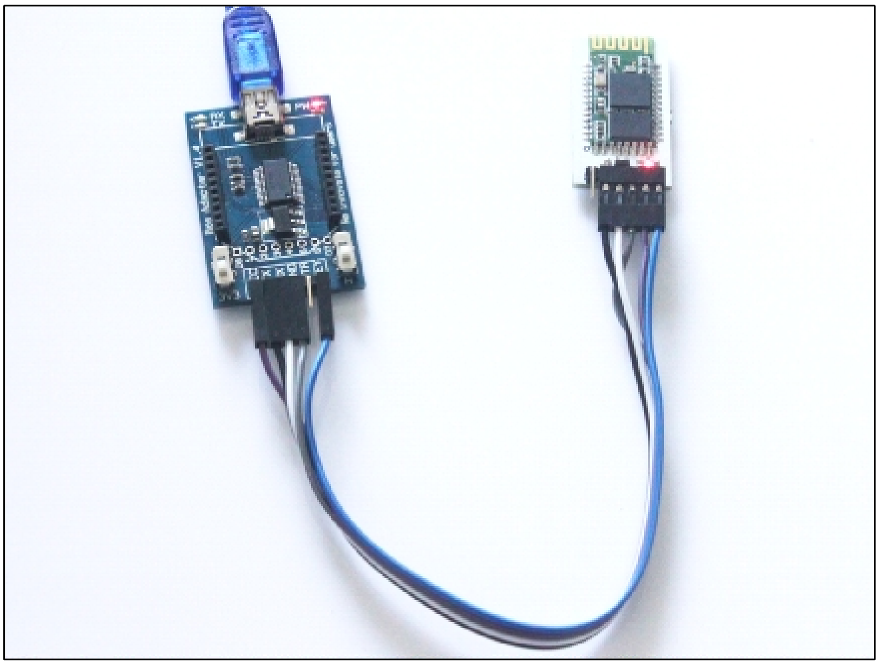
Power Supply Voltage : 3.3V-5V

IO Voltage : 3.3V/TTL

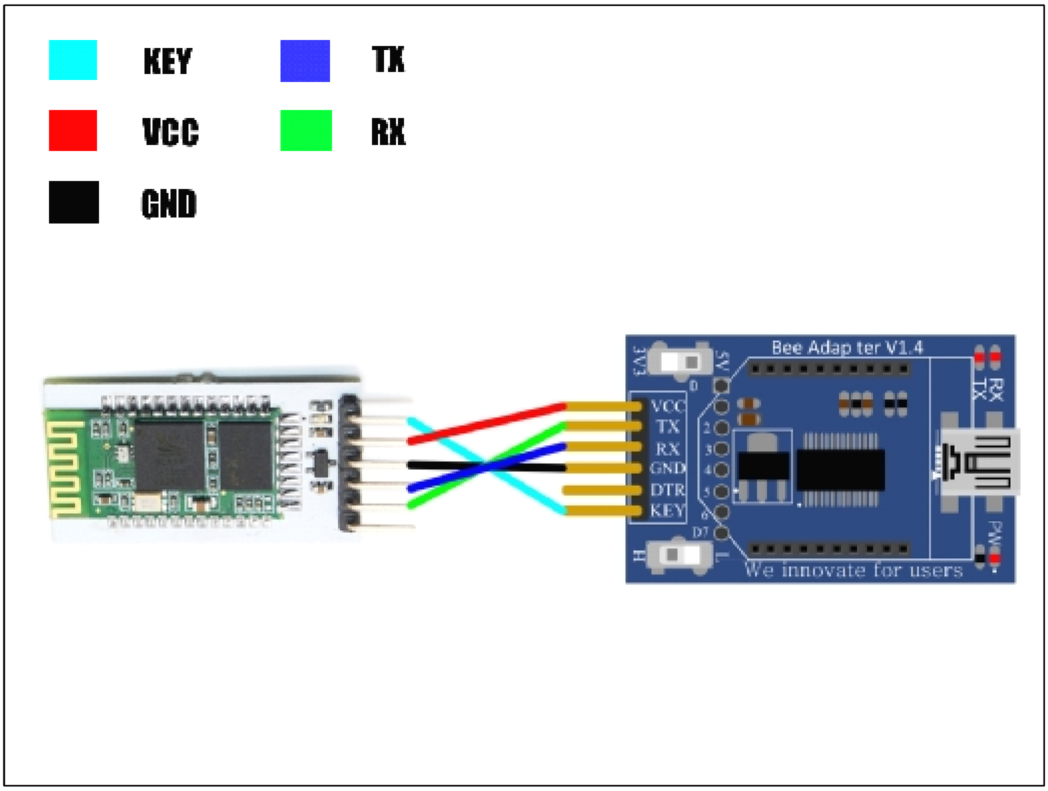
Hardware and Software Preparation



Part 1 Communication Between Bluetooth Modem and Bee Adapter



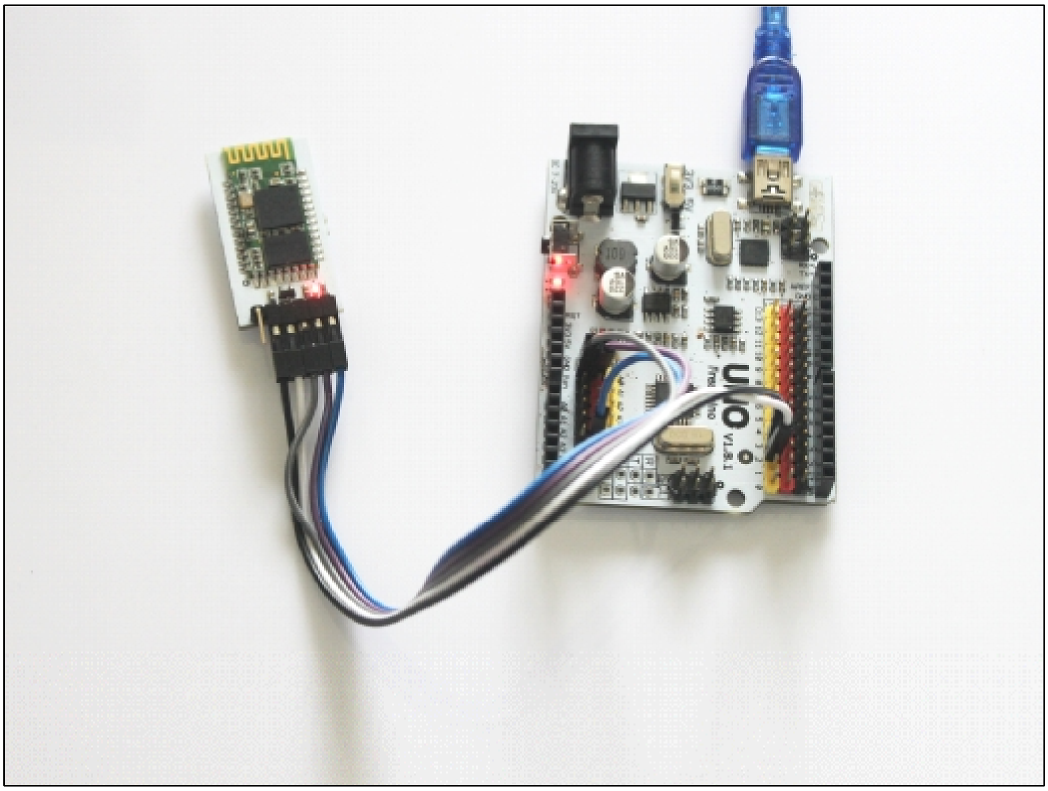
1. Set the Bee Adapter Switch in H Port to enter AT Mode
2. Set to IO Voltage 3V3
3. Assemble the Bee Adapter and Bluetooth Modem with Jumper Wire, and then connect to PC (make sure Bluetooth Modem LED blink slowly in AT Mode)



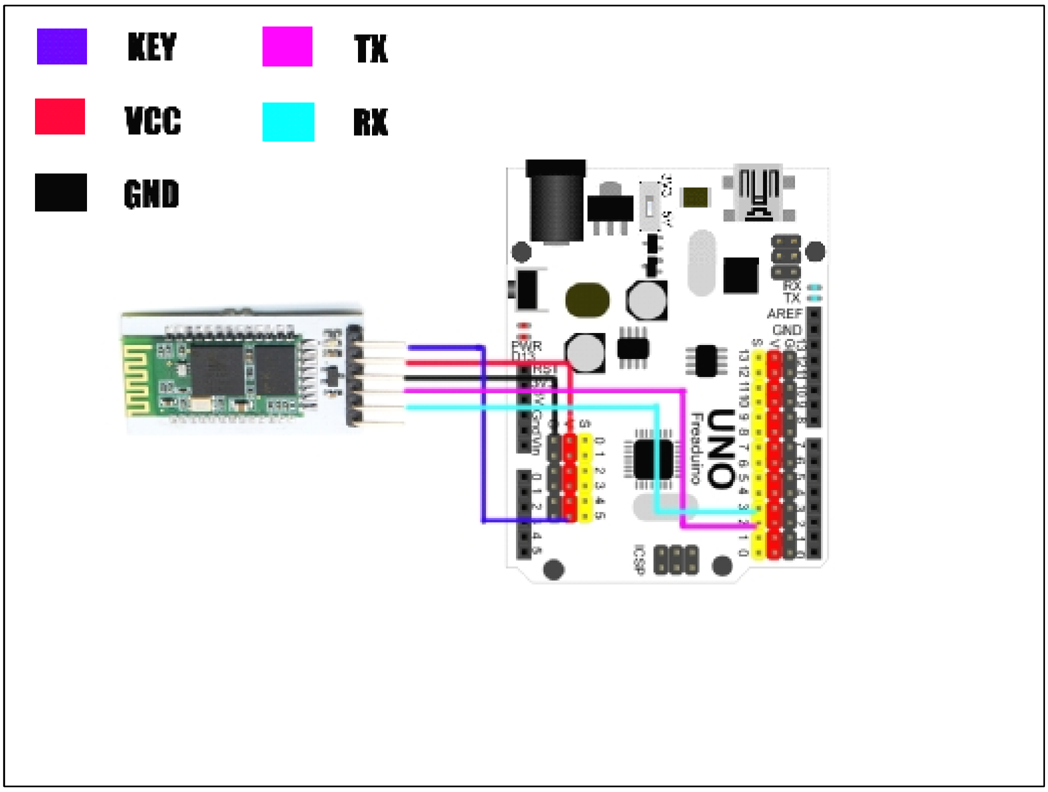
- KEY connect to KEY
- VCC connect to VCC
- GND connect to GND
- TX connect to RX
- RX connect to TX

4. Open Serial Tool sscom32 and set the BaudRate as 38400. Type AT, and it would respond OK. With Bee Adapter, you can achieve AT Command control not need of programming code.

Part 2 Communication Between Bluetooth Modem and Arduino

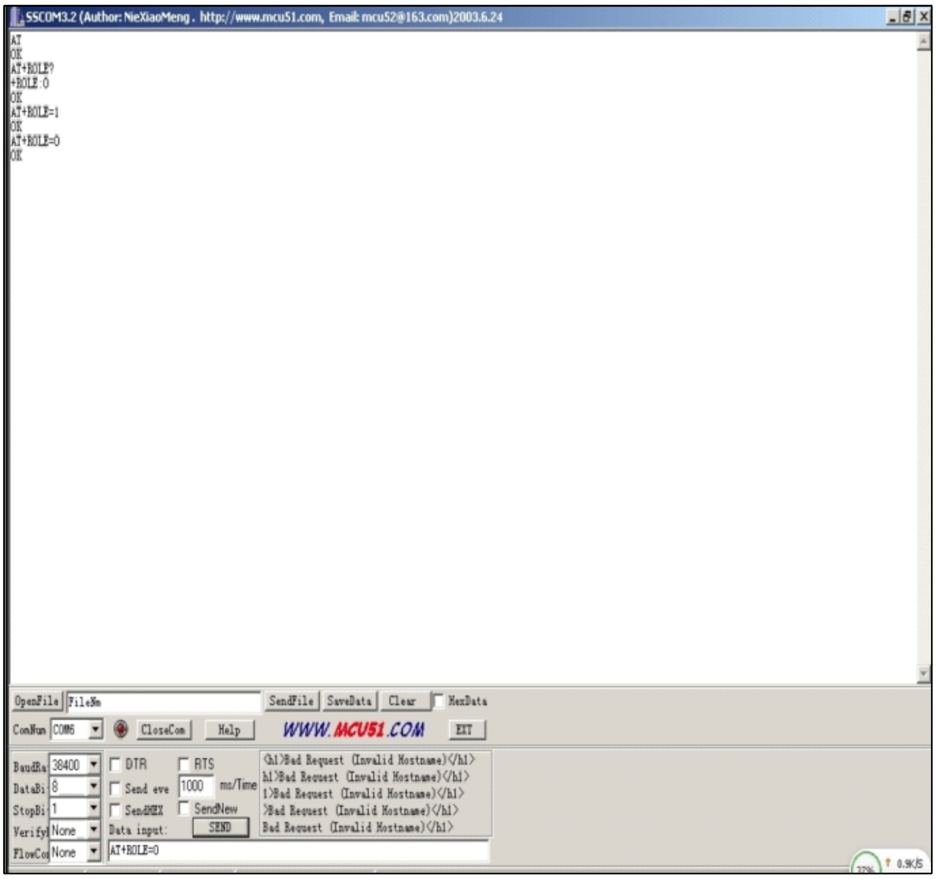


- Set to IO Voltage 3V3, and make sure Bluetooth Modem LED blink slowly in AT Mode.
- Connect Bluetooth and UNO with jumper wire as below, and then link to PC
- Bluetooth KEY link to UNO V(VCC), so as to enter AT Mode .
- Bluetooth VCC to UNO V(VCC)
- Bluetooth GND to UNO G(GND)
- Bluetooth TX to UNO D2
- Bluetooth RX to UNO D3



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 ArduinoIDE 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 sscom32 and set BaudRate as 38400
8.Manually input AT and it would respond OK
9. Input AT+ROLE? And it would respond +ROLE=0, therefore, We default the Bluetooth Modem 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



SSCOM3.2 (Author: NieXiaoMeng . http://www.mcu51.com, Email: mcu52@163.com)2003.6.24

```
AT
OK
AT+ROLE?
+ROLE:0
OK
AT+ROLE=1
OK
AT+ROLE=0
OK
```

ComNum: COM6

BaudRa: 38400

DataBi: 8

StopBi: 1

Verify: None

FlowCo: None

□ DTR □ RTS

□ Send eve 1000 ms/Time

□ SendHEX □ SendNew

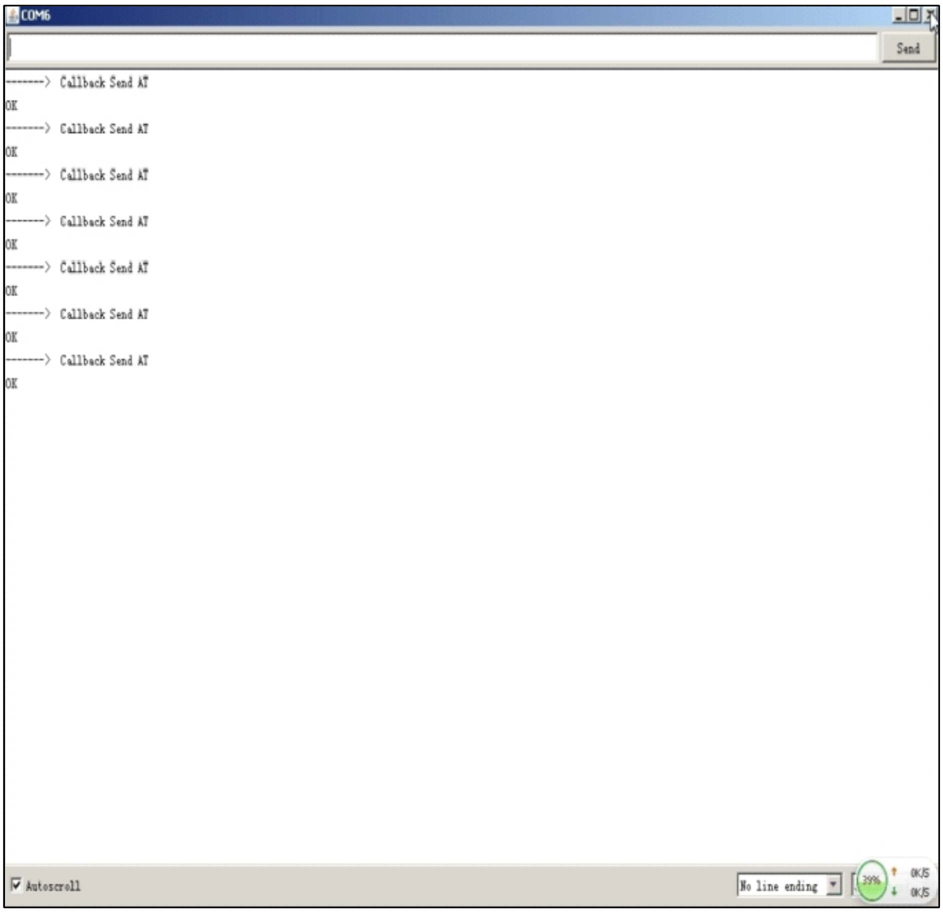
Data input: AT+ROLE=0

```
<h1>Bad Request (Invalid Hostname)</h1>
h1>Bad Request (Invalid Hostname)</h1>
1>Bad Request (Invalid Hostname)</h1>
>Bad Request (Invalid Hostname)</h1>
Bad Request (Invalid Hostname)</h1>
```

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. It can automatically respond OK for AT Command



```
-------> Callback Send AT
OK
-------> Callback Send AT
OK
-------> Callback Send AT
OK
-------> Callback Send AT
OK
-------> Callback Send AT
OK
-------> Callback Send AT
OK
-------> Callback Send AT
OK
```

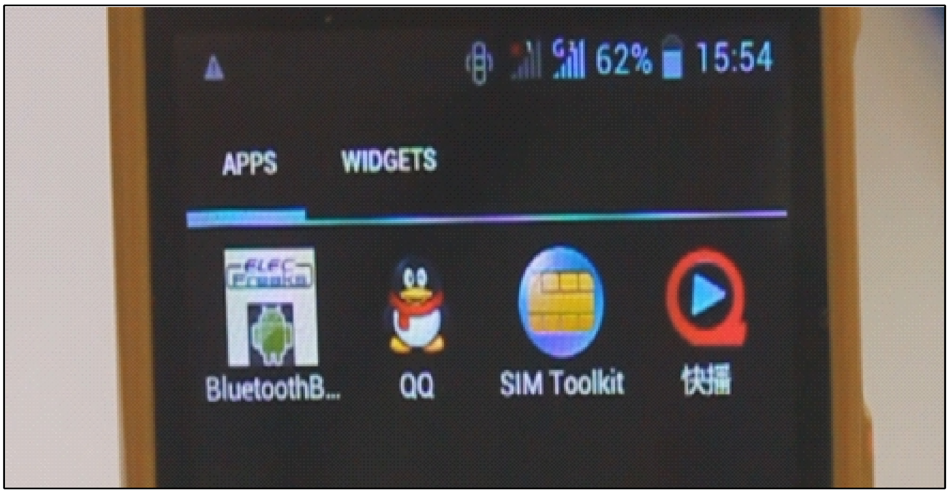
Part 3. Communication Between Android and Arduino with Bluetooth

- Remove the KEY PIN jumper wire to enter Communication Mode. In Communication Mode, make sure the Bluetooth Modem LED 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.

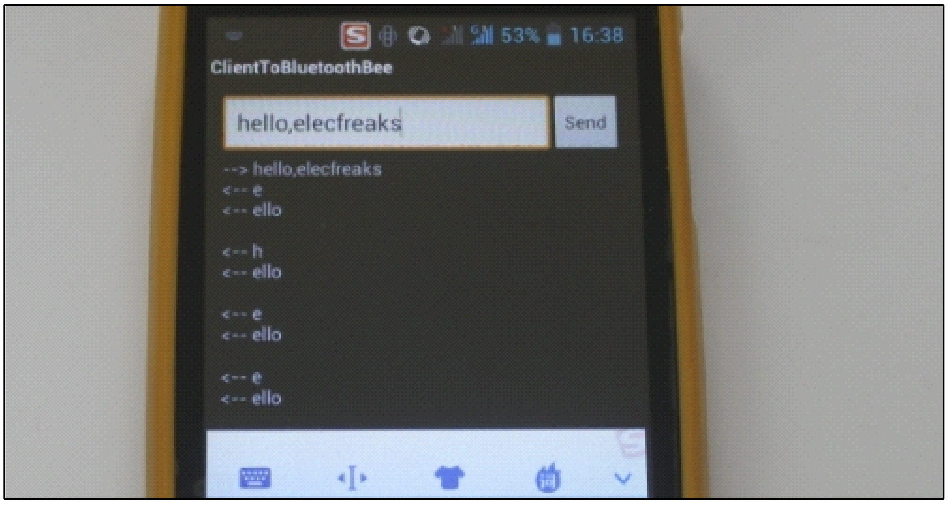
Step 1. Android Installation

1.Download APP for EF _BluetoothBeeV1.1 from
http://elecfreaks.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 PIN 1234.
5. Input hello,elecfreaks in mobile, and the PC would display hello,elecfreaks from mobile. (For sscom32, remember to set the BaudRate as 9600.)
6. Input hello from PC, mobile would also respond hello.



hello,elecfreaks

```
--> hello,elecfreaks
<-- e
<-- ello

<-- h
<-- ello

<-- e
<-- ello

<-- e
<-- ello
```

Q&A

If you use Arduino Mega1280/2560 or Freaduino ADK there are some difference. 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 means the library do not support D0-D7 as receiving pins, and you just could use pins:10, 11, 12, 13, 50, 51, 52, 53, 62, 63, 64, 65, 66, 67, 68, 69 for receiving. So there are two way to resolve it, but they all need external jumper wires.

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

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

Second method: Use the other Hardware Serial port because Mega has 4 hardware serial ports. Jumper wires connect to DOUT-RX1(D19) DIN-TX1(D18)

Note: This equipment should be installed and operated with minimum distance 20cm between the radiator and your body.

FCC Caution: Any changes or modifications not expressly
approved by the party responsible for compliance could void the user's authority to operate this equipment.
This device complies with Part 15 of the FCC Rules.
Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.
This device and its antenna(s) must not be co-located or operating in conjunction with any other antenna or transmitter.

15.105 Information to the user.
(b) For a Class B digital device or peripheral, the instructions furnished the user shall include the following or similar statement, placed in a prominent location in the text of the manual:
Note: This equipment has been tested and found to comply
with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:
—Reorient or relocate the receiving antenna.
—Increase the separation between the equipment and receiver.
—Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
—Consult the dealer or an experienced radio/TV technician for help.

Radiation Exposure Statement:
This equipment complies with FCC radiation exposure limits set forth for an uncontrolled environment.
This transmitter must not be co-located or operating in conjunction with any other antenna or transmitter.
The availability of some specific channels and/or operational frequency bands are country dependent and are firmware programmed at the factory to match the intended destination.
The firmware setting is not accessible by the end user.
The final end product must be labelled in a visible area with the following:
“Contains Transmitter Module 2AEN8HC06”