

BLE-Link (SKU:TEL0073)

Introduction

If you have Arduino UNO,MEGA Controller, or Expansion board with xbee socket,you just add DFRobot latest BLE-Blink module,you can use to the function of bluetooth 4.0.

Imagine using Arduino or IOS combined with Arduino develop a wearable mobile device, such as smart phones, smart pedometer bracelet and so on.These wearable devices can communicate with your mobile phone via bluetooth 4.0.By linking into star network of low power consumption, low-power bluetooth 4.0 achieves rapid real-time communication.Many software or hardware engineers are hoping to have such a platform taht can realize above functions.

BLE - the LINK is the communication module based on bluetooth 4.0,it uses the XBEE package which is compact and compatible with the XBEE base.it also can run on 3.3V MCU.

With our Android and IOS app, you can quickly build the communication between phone and arduinos.

Two BLE4.0 modules can realize the point-to-point wireless transparent transmission, master-slave machine setting, wireless program burning, HID connection with PC.

At the same time, we provide developers more freedom and support. User not only can debug BLUNO through the AT command, but also update BLE chip program via USB.

BLE - LINK bluetooth 4.0 module can be plug on any Arduino MCU with XBEE base, so as to realize the bluetooth wireless control.

Specifications

- * bluetooth chip:TI CC2540
 - * Frequency: 2.4GHz
 - * Transfer rate: \leq 1Mbps
 - * Modulation: GFSK, bluetooth low power, V4.0
 - * Power consumption: working:10.6mA average ,ready mode:8.7mA
 - * sensitivity: -93dB
 - * Input Voltage: +3.3 DC
 - * Operating temperature: -40 °C ~ +85 °C
 - * Transmission distance: 60m in free space
 - * size:32mm * 32mm
-
- * support by the AT command to debug the BLE

- * support the master-slave machine switch
- * support transparent transmission serial port (only support UNO and Arduino Mega)
- * support bluetooth remote update the Arduino program (only support UNO and arduino mega)
- * support usb update BLE chip program
- * support bluetooth HID
- * convenient to upgrade firmware BLE
- * support Android and IOS applications, open source code, suitable for secondary development by the user

Android devices Support List

Nexus 4+

Samsung Galaxy s4

Samsung Galaxy note 3

Xiaomo 2s

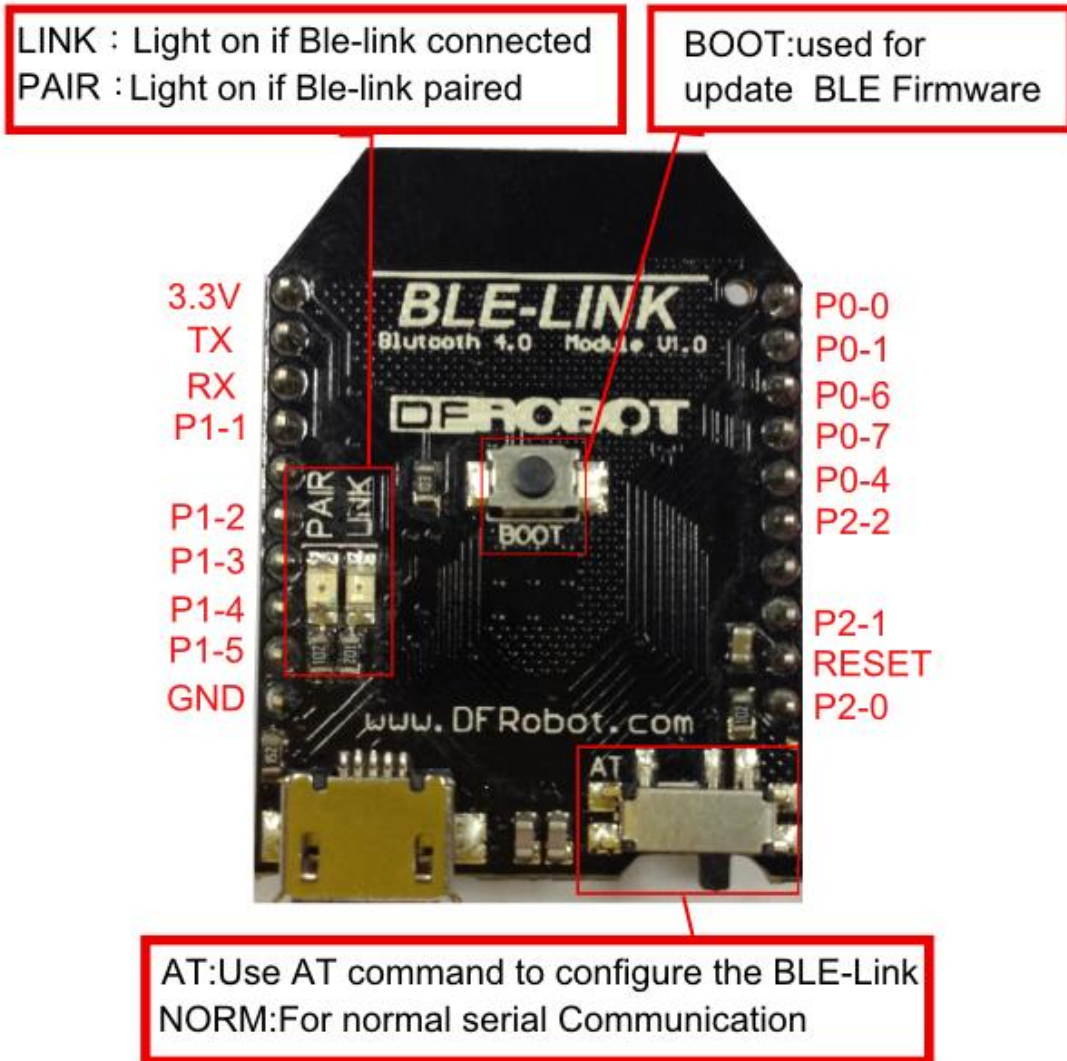
Android devices are various, it is not a completely list. The devices carry the Bluetooth 4.0 (BLE) and use the Android 4.3+ original firmware with BLE drivers would be compatible.

Apple devices Support List

IOS 7+ Device:

- * iPhone 4S+
- * iPad 3+
- * iPad Mini
- * iPod 5th Gen
- * MacBook Air Mid 2011+
- * Mac Mini Mid 2011+
- * MacBook Pro Early 2012+
- * iMac Late 2012+

Pinout



Get start with the BLUNO

In this section, you can use the BLUNO to connect with the Android phone or iPhone

Tools required

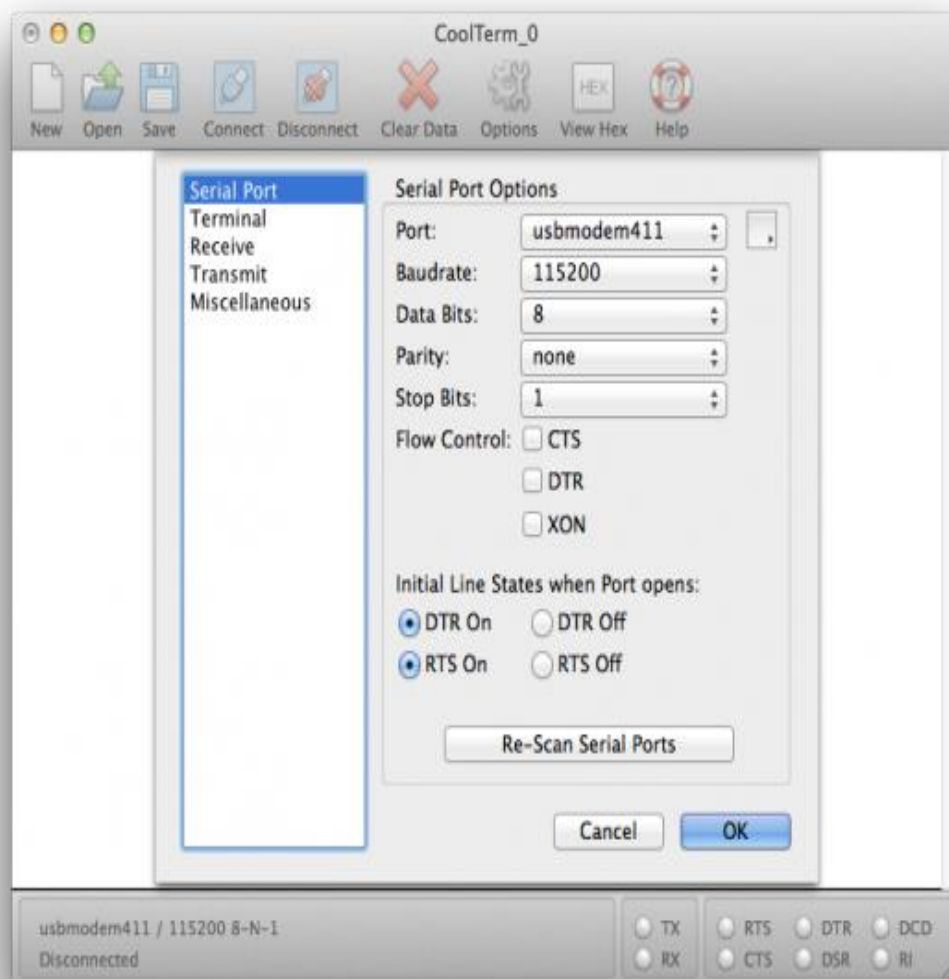
BLE-LinK x1

Android 4.3 + Devices with BLE or IOS 7.0 + Devices

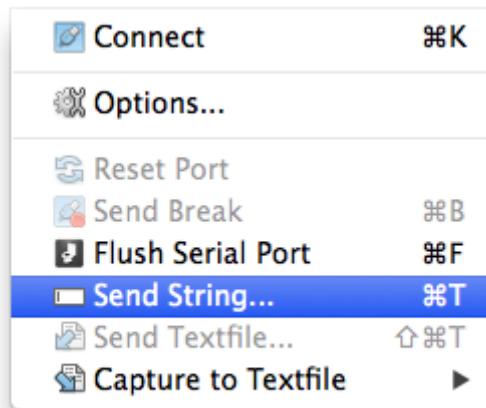
Micro USB cable x1

Configure the BLE through AT command

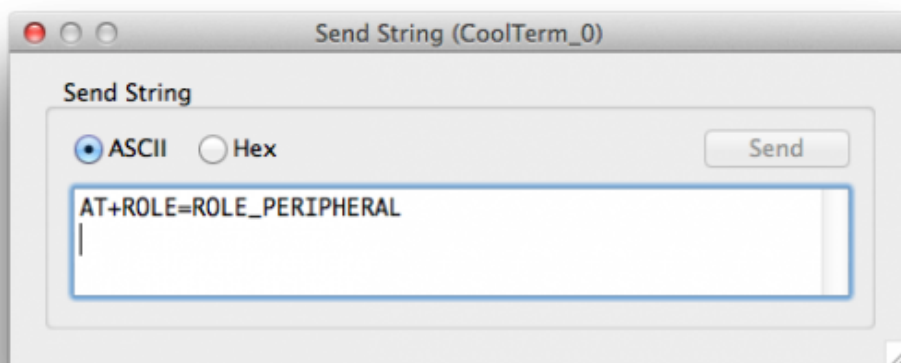
1. Turn the switch to "AT" and AT command mode will be entered.
2. We need a serial monitor for configuring the BLE in this part. There're lots of good tools like putty, CoolTerm and Arduino serial monitor. In this case, we choose the Coolterm, which is compatible with both Windows and Mac.
3. Click the "Options" button, set the baud rate to 115200 as following and choose the proper serial port :



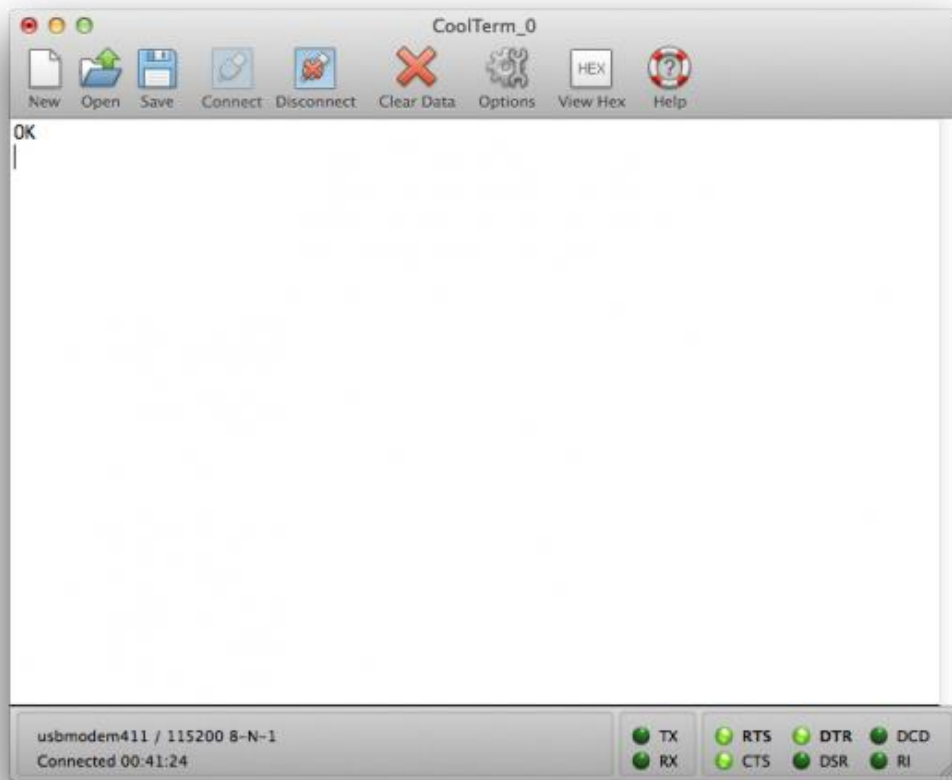
4. Click the "Connect" button to open the com port.
5. Click the "Connection" in the menu and select "Send String..."



6. Type or copy the AT command in the dialog like this and press send button. **The command is terminated by the <CR+LF>, so don't forget to press Enter(Return) key**



7. If the BLE is successfully configured , you will receive "OK" from it.



8. If received "ERROR CMD" instead, try sending it again or you should check whether the command is correct or not.

9. Sending the following command as mentioned above :

For IOS Device :

Input :AT+ROLE=ROLE_PERIPHERAL<CR+LF> Answer(Return):OK

Input :AT+FSM=FSM_TRANS_USB_COM_BLE<CR+LF> Answer(Return):OK

Input :AT+MIN_INTERVAL=20<CR+LF> Answer(Return):OK

Input :AT+MAX_INTERVAL=40<CR+LF> Answer(Return):OK

For Android Device :

Input :AT+ROLE=ROLE_PERIPHERAL<CR+LF> Answer(Return):OK

Input :AT+FSM=FSM_TRANS_USB_COM_BLE<CR+LF> Answer(Return):OK

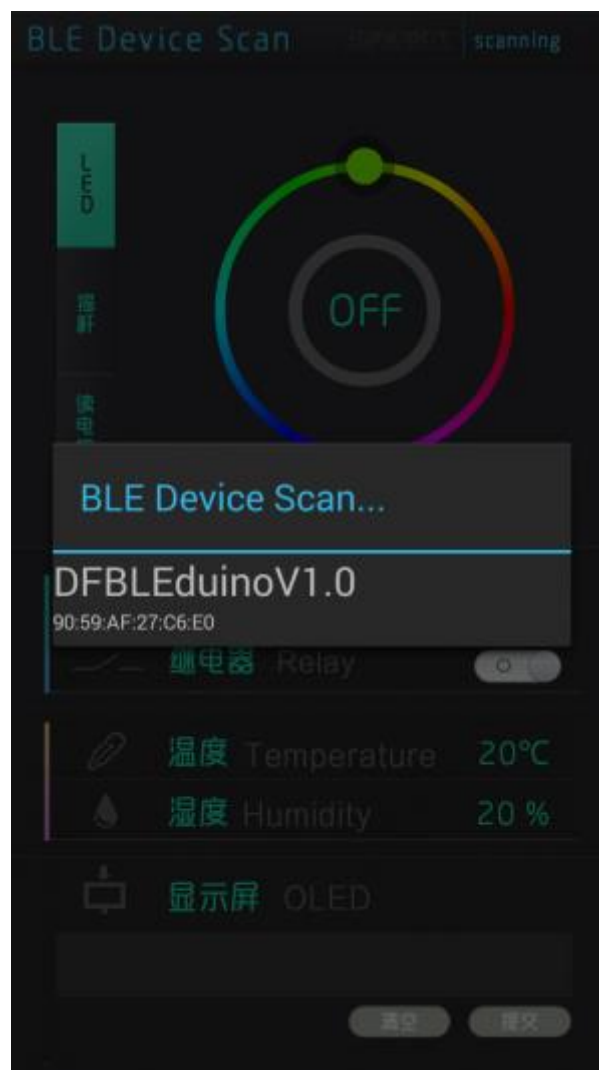
Input :AT+MIN_INTERVAL=10<CR+LF> Answer(Return):OK

Input :AT+MAX_INTERVAL=10<CR+LF> Answer(Return):OK

10. Turn the switch to "NORM" entering transparent communication mode.

Install the software and upload the sketch

11. Disconnect the USB, plug the shield onto the BLE-Link, and Reconnect it.
12. Put the PlainProtocol Library into your Arduino library folder.
13. Upload the BLE-Link sketch into the board. (If Bluetooth is connected, release the connection first)
14. Unplug and plug the USB .
15. ON Android, Install the BLE-Link application into phone. The source code is in the Document section below.
- ON IOS, download here. For developer, you can directly download the source code in the Document section below.
16. Run the application and select the BLE-Link board to connect with



17. After entering you can control the RGB LED, OLED, Relay, Buzzer and can get the temperature, humidity, and etc.



18.The top bar shows the BLE connection state. You can touch it to call the scanning dialog for connecting other devices.(Android only)

Note: On IOS, if there is something wrong with the bluetooth connection , please do the 1-10 steps again and restart the IOS device first.

Wireless Programming via BLE

In this section, we will learn how to Upload the sketch on air via BLE. It is really amazing that you can even use your mobile phone to do uploading process.

Tools required

- * **Ble-link x2**
- * **Micro USB cable x2**
- * **XBee/Bluetooth Bee Adapter x2**

Configure the BLE through AT command

1.There are two different roles of BLE devices , CENTRAL and PERIPHERAL. So if we want establish transparent communication, one device should be configured to CENTRAL, while the other should be configured to PERIPHERAL,and connected to the XBee/Bluetooth Bee adapter board.

2.Turn the switches to "AT" to enable the AT command mode.

3.Connect them with computer.

4.For the CENTRAL device, sending the following command as mentioned in the previous section :

Input : AT+ROLE=ROLE_CENTRAL<CR+LF> Answer(Return):OK

Input : AT+FSM=FSM_TRANS_USB_COM_BLE<CR+LF> Answer(Return):OK

5.For the PERIPHERAL one, sending the following command as mentioned in the previous section :

Input : AT+ROLE=ROLE_PERIPHERAL<CR+LF> Answer(Return):OK

Input : AT+FSM=FSM_TRANS_USB_COM_BLE<CR+LF> Answer(Return):OK

6.Turn the switches to "NORM", entering normal communication mode.

Transparent communication Test

7.Unplug and replug the power of the two and in seconds the LINK LEDs are on, which means they have connected.

8.Use Coolterm to connect the CENTRAL device via USB to test the connection.

9.Type the keyboard on the computer and the TX/RX LED on PERIPHERAL one will Blink , and thus you can get the data from Serial.

Transparent communication

10.Disconnect the Serial connection in Coolterm.

11.Click Upload in Arduino to upload the sketch to the PERIPHERAL device.

Update BLE Firmware on BLUNO

This method is compatible with Windows XP, and Windows 7. Some Windows 8 versions might not work with this method.

1.Press and hold the Boot button on BLUNO and connect computer through USB. After power on, release the button.

2.Computer will recognize the BLUNO as an USB flash drive.

3.Open the disk and delete the "*.bin" file. After that USB flash drive will automatically reboot and be remounted.

4.Copy the new "*.bin" file (see the Document section "BLE Firmware") into it.

5.After it automatically reboots, the BLE Firmware will be successfully updated.

AT Command List

Turn the switch to "AT" to enable the AT command mode. After BLE reboot, the setting will be in effect.

<CR+LF> means Carriage-Return and Line-Feed

1. "AT+FSM" change the working mode

AT+FSM=FSM_TRANS_USB_COM_BLE<CR+LF> USB-UART BLE
transparent mode

AT+FSM=FSM_HID_USB_COM_BLE_AT<CR+LF> USB-UART BLE HID
mode

AT+FSM=?<CR+LF> Request the working mode (default:
FSM_TRANS_USB_COM_BLE)

2."AT+ROLE" change the CENTRAL-PERIPHERAL configuration

AT+ROLE=ROLE_CENTRAL<CR+LF> BLE CENTRAL mode

AT+ROLE=ROLE_PERIPHERAL<CR+LF> BLE PERIPHERAL mode

AT+ROLE=?<CR+LF> Request the CENTRAL-PERIPHERAL configuration
(default: ROLE_PERIPHERAL)

3."AT+MIN_INTERVAL" change the minimum connection interval

AT+MIN_INTERVAL=10<CR+LF> Recommended minimum connection interval
(10ms) for PC and Android

AT+MIN_INTERVAL=20<CR+LF> Recommended minimum connection interval
(20ms) for IOS

AT+MIN_INTERVAL=?<CR+LF> Request the minimum connection interval
(default: 10)

4. "AT+MAX_INTERVAL" change the maximum connection interval

AT+MAX_INTERVAL=10<CR+LF> Recommended maximum connection interval
(10ms) for PC and Android

AT+MAX_INTERVAL=40<CR+LF> Recommended maximum connection interval
(40ms) for IOS

AT+MAX_INTERVAL=?<CR+LF> Request the maximum connection interval
(default: 10)

5. "AT+UART" change the baud rate of UART

AT+UART=115200<CR+LF> Set the baud rate to 115200

AT+UART=?<CR+LF> Request the baud rate of UART (default: 115200)

6."AT+BIND" bind another BLE chip

AT+BIND=0x0017ea9397e1<CR+LF> Set the BLE binding address to
0x0017ea9397e1

AT+BIND=?<CR+LF> Request the binding address (default: 0x000000000000)

7. "AT+CMODE" set whether the connection of BLE is binding or arbitrary

AT+CMODE=UNIQUE<CR+LF> BLE can only connect to the BLE chip with binding address (see "AT+BIND" command)

AT+CMODE=ANYONE<CR+LF> BLE can connect to any other BLE chips

AT+CMODE=?<CR+LF> Request the binding connection mode(default:ANYONE)

8."AT+MAC" Request MAC address

AT+MAC=?<CR+LF> Request MAC address of the BLE

9."AT+NAME" Set the name

AT+NAME=DFBLEduinoV1.0<CR+LF> Set the name of BLE to "DFBLEduinoV1.0".The length is limited to 13 Bytes or below

AT+NAME=?<CR+LF> Request the name of the BLE (default:DFBLEduinoV1.0)

10."AT+RESTART" restart the BLE

AT+RESTART<CR+LF> Restart the BLE chip

Documents

- * PlainProtocol library
- * Arduino Sketch
- * BLUNO Android source code
- * BLUNO IOS source code
- * BLUNO Android APK Package (Android 4.3+)
- * BLE_ENC_V1.6 Firmware
- * BLE-Link schematic
- * BLUNO schematic
- * [BLUNO IOS APP STORE](#)