

AT Command



NOTE: To get a 'better' Bluetooth experience on BLE 4.1 modules, we highly recommend you to update the newest firmware (https://www.dfrobot.com/wiki/index.php/DFRobot_BLE4.1_Module_SKU:_TEL0120#Firmware_Upgrade) first.

AT is the abbreviation of attention. AT command is the communication command used to configure Bluetooth parameters. It begins with AT, end with <CR+LF> (carriage return and line feed).



NOTE: <CR+LF> is not a part of AT Command.

AT Command List

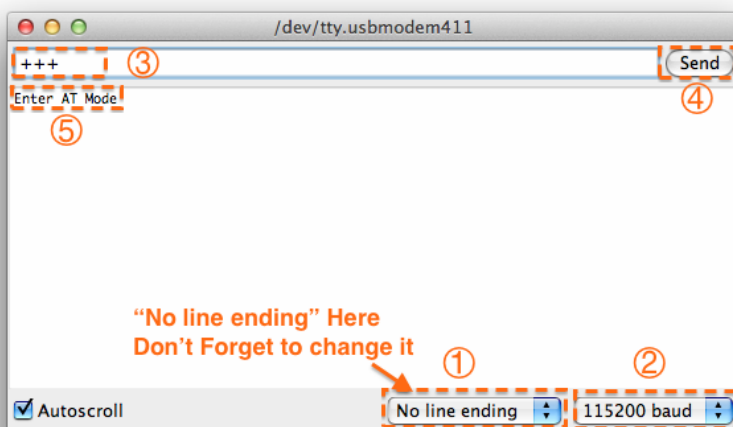
AT Command	Function	Parameters	Description
AT+ROLE= X	Set Bluetooth mode	"?" Query Bluetooth working mode "ROLE_CENTRAL" Set as the central device "ROLE_PERIPHERAL" Set as the peripheral device	Return Current Mode Return "OK" Return "OK"
AT+NETWORK=	Set network mode	"?" Query the network mode "P2P" Set to peer connection "STAR" Set to star network	Return Current Network Mode Return "OK" Return "OK"
AT+MIN_INTERVAL=	Set the minimum communication interval	"?" Query the current minimum communication interval (10~99999ms)	Return Minimum Communication Interval Return "OK"
AT+MAX_INTERVAL=	Set the maximum communication interval	"?" Query the current maximum communication interval (10~99999ms)	Return Maximum Communication Interval Return "OK"
AT+BIND=	Set binding Bluetooth address	"?" Query the current binding MAC address e.g.: 0x0017ea9397e1 Set Bluetooth MAC address in need	Return 0x000000000000 (Default) Return "OK"
AT+CMODE=	Set BLE connection mode	"?" Query the current connection mode "UNIQUE" Specify the connection mode "ANYONE" Set to free connection	Return BLE connection mode Return "OK" Return "OK"
AT+MAC=	Query the Bluetooth address	"?" Query the current Bluetooth MAC address	Return the current Bluetooth MAC address
AT+NAME=	Set the name	"?" Query the Bluetooth device name "DFBLEduino2"	Return Bluetooth device name Return "OK"
AT+SETTING=	Modify/Reset to default mode	"?" Query the current setting "DEFAULT" Reset to initial setting "DEFPERIPHERAL" Reset to initial slave setting "DEFCENTRAL" Reset to initial host setting	Return the device Current Status Return "OK" Return "OK" Return "OK"
AT+USB_MONITOR=	Turn the USB monitor ON/OFF	"?" Query the current USB monitor "ON" Turn on the USB monitor "OFF" Turn off the USB monitor	Return the current USB monitor status Return "OK" Return "OK"
AT+HELP=	Get help from AT command	"ALL" Query all AT commands help information	Return AT Command help information
AT+NET_INF=	Query device information in the current network	"?" Query device network information	Return the current network information

AT Command	Function	Parameters	Description
AT+VERSION=	Query the firmware version	? Query firmware version information	Return firmware version information

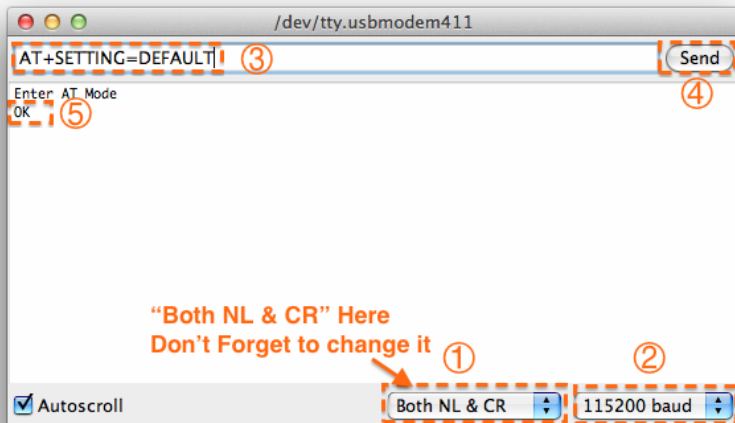
AT+RSSI=	Query the RSSI value	"?" Query the RSSI value	Return the current RSSI value (Return -000 when where is no connection)
AT+P=	Query & Set P(n, n=0,1,2,3...) status	"?" Set IO pins to input mode and query the voltage "1" Set IO pins to output mode and output HIGH voltage "0" Set IO pins to output mode and output LOW voltage	Eg:AT P47=? (Return 0 or 1) Eg:AT P47=1 (Return "OK") Eg:AT P47=0 (Return "OK")
AT+WL=	Query/Declare the White List	"?" Query the White List "NONE" Declare the White List	Return White List MAC address list Return "OK"
AT+LOWPOWER=	Query & Set the low-power mode	"?" Query the low-power mode "ON" Turn on the low-power mode "OFF" Turn off the low-power mode	Return current status Return "OK" Return "OK"
AT+EXIT	Exit the AT command mode	Exit the AT command mode	Return "OK"

Configure the BLE through AT command

1. Open the Arduino IDE.
2. Select the correct serial port in Menu->Tool->Serial port
3. Open the Serial monitor (on the upper right of the IDE windows)
4. Select the "No line ending" (①) and 115200 baud (②) in the pull-down menu
5. Type "+++" (③) like this and press send button (④)
6. If the AT Command Mode is successfully access , you will receive "Enter AT Mode" (⑤) from it.



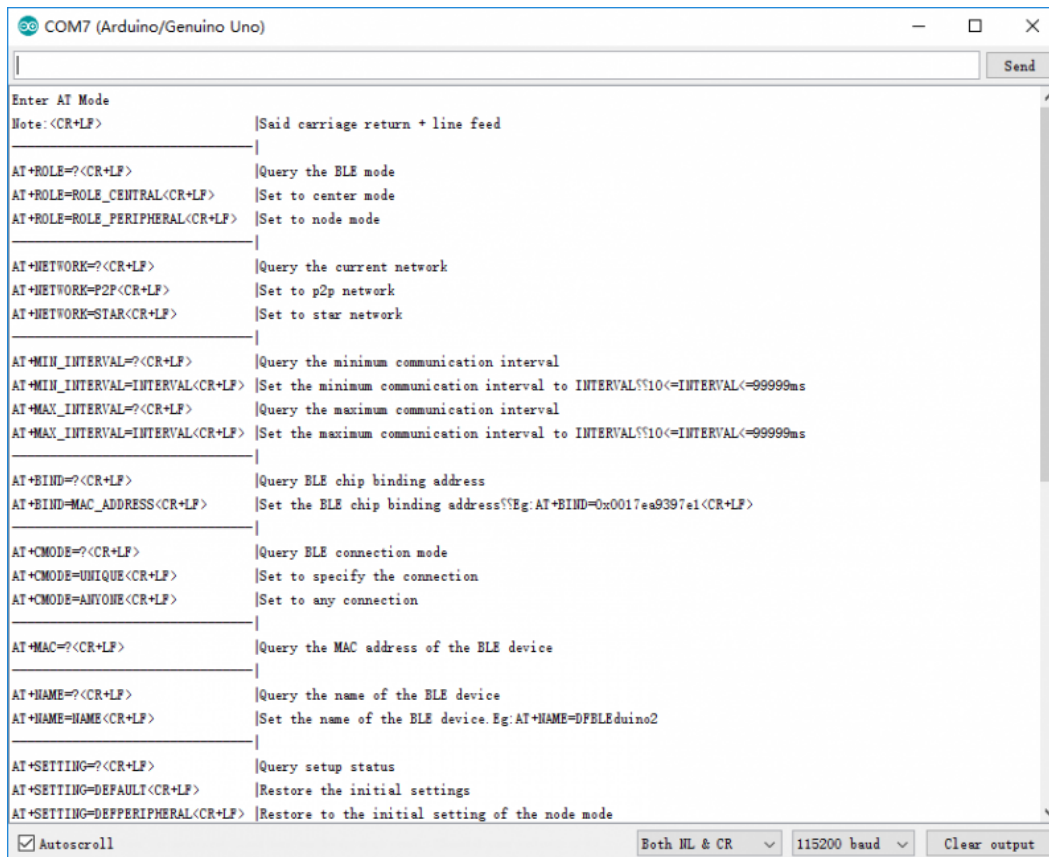
7. Select the "Both NL & CR" (①) and 115200 baud (②) in the pull-down menu
8. Type or copy the AT command in the dialog (③) like this and press send button (④)
9. If the BLE is successfully configured , you will receive "OK" (⑤) from it.



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

11. USE "AT+EXIT" to exit AT Mode

12. You can input AT+HELP=ALL to query all AT Command information



Bluetooth Connection

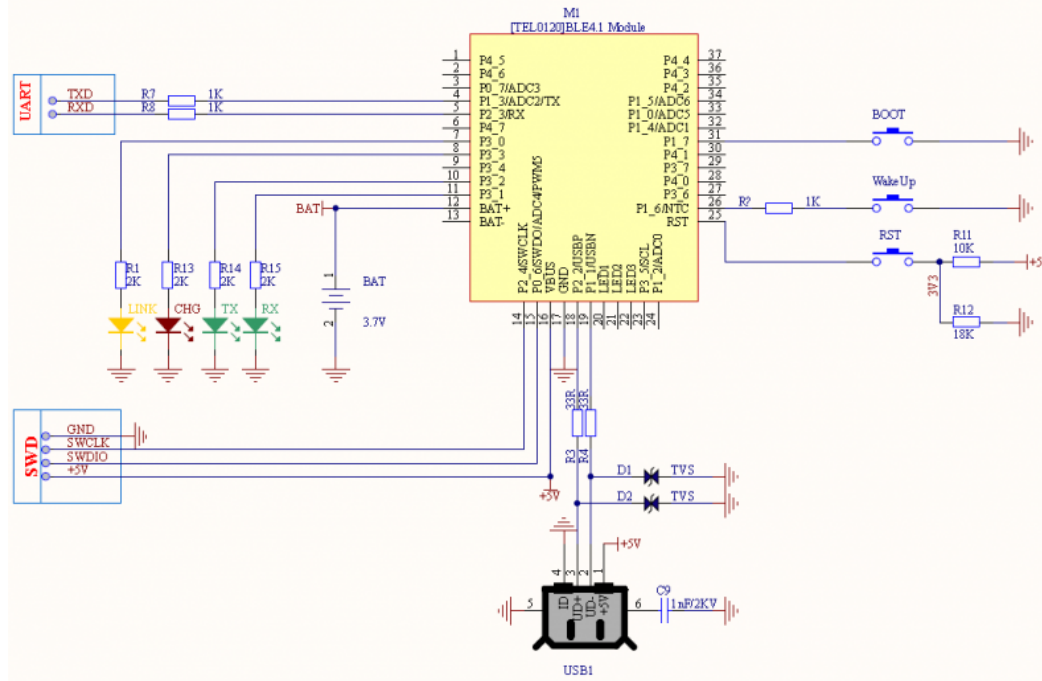
Before using BLE4.1 series devices, it is better to know some key buttons first.

- **BOOT:** Update Bluetooth firmware/Enable the approach connection (<10cm).
 - Firmware update**: Press and hold the "BOOT" button until you plug USB cable, entering firmware update mode. For any enquiry, please refers to

the part: Firmware Update. (You can release your hand after you plugged the USB cable.)

- Approach connection**: Press and hold the **host BOOT** button, approach the slave Bluetooth device to connect.
- **WAKEUP**: the button can wake up the device from low-power consumption mode (you can also connect this pin to MCU to realize the same function)
- **RST**: BLE4.1 reset button

Below is the BLE4.1 minimum system circuit diagram for your reference: Fig1: the BLE4.1 minimum system circuit diagram

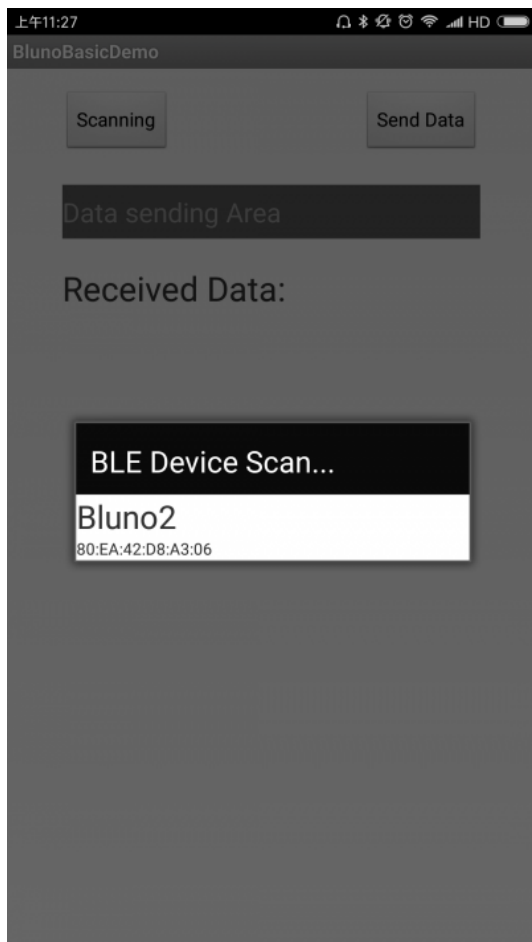


Smartphone Bluetooth Connection

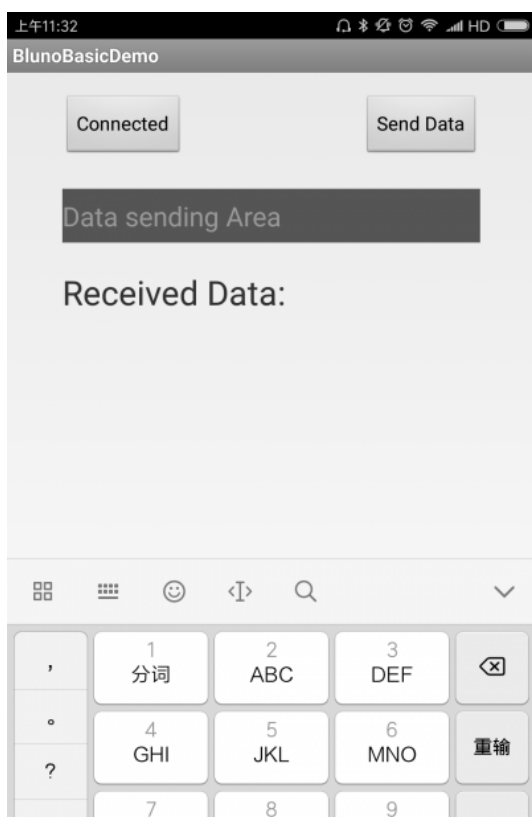
- Android Terminal (<https://raw.githubusercontent.com/DFRobot/BlunoAccessoryShieldDemo/master/Android/BLUNO.apk>) and IOS Terminal (<https://itunes.apple.com/us/app/wan-zhuan-bu-lu-nuo/id737793451?ls=1&mt=8>)
- BlunoBasicDemo Android (<https://raw.githubusercontent.com/DFRobot/BlunoBasicDemo/master/Android/BlunoBasicDemo.apk>), Android source code (DFRobot) (<https://github.com/DFRobot/BlunoBasicDemo/tree/master/Android>)
- LightBlue IOS (<https://itunes.apple.com/us/app/lightblue-explorer-bluetooth/id557428110?mt=8>)
- Blynk IOS (<https://itunes.apple.com/us/app/blynk-control-arduino-raspberry/id808760481?ls=1&mt=8>)

Different from the traditional Bluetooth (such as Bluetooth headset), BLE4.1 need specified eigenvalue and device service ID to connect smartphone. The direct connection between BLE and smartphone device manager will lead to communication problems. Therefore, it needs a the third party APP, such as **BunoBasicDemo** (published by DFRobot), **BLE Device Monitor** from TI .etc. In this section, we'll take BunoBasicDemo published by DFRobot as an example.

- Set BLE4.1 as peripheral device with command **AT+ROLE=ROLE_PERIPHERAL**. Set connection mode as **P2P** by **AT+NETWORK=P2P** command and restart BLE4.1 (power off and restart).
- Open **BunoBasicDemo** app, click **SCAN**, then we'll see the scanned BLE4.1 device .

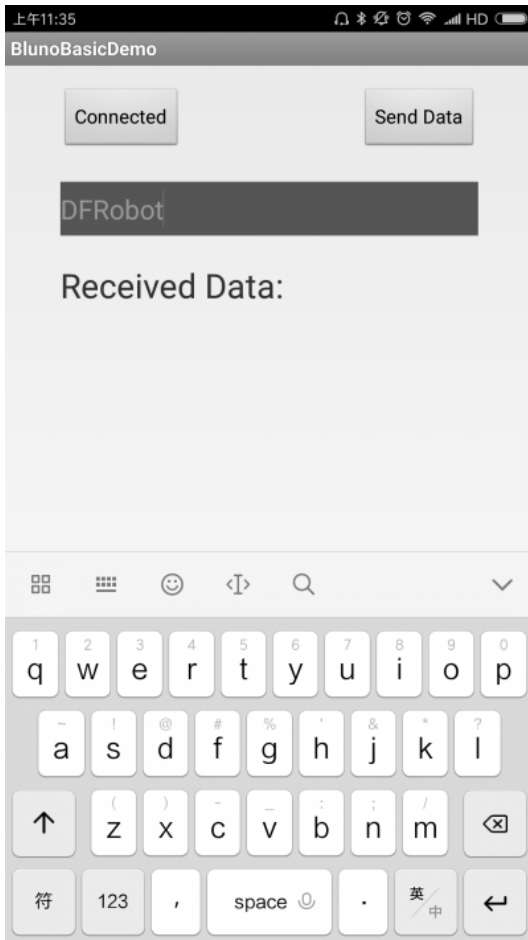


- Click the device to connect. Once it connected successfully, it will show Connected and LINK will light for 3s and blink for every 3s.

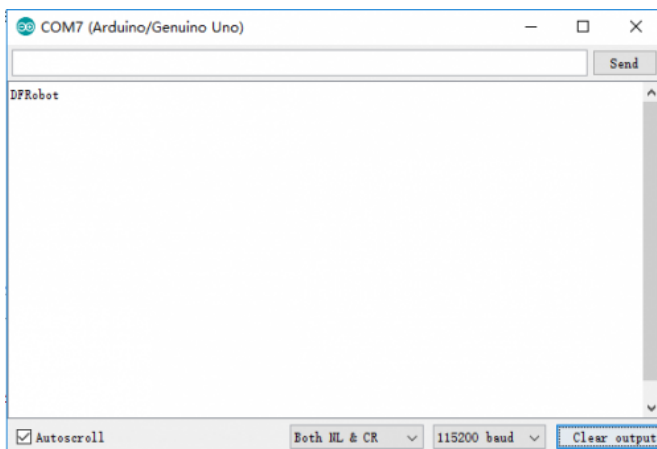




- Send data from the **Data sending Area**



- Click **Send Data** to send data.



PC Bluetooth Connection

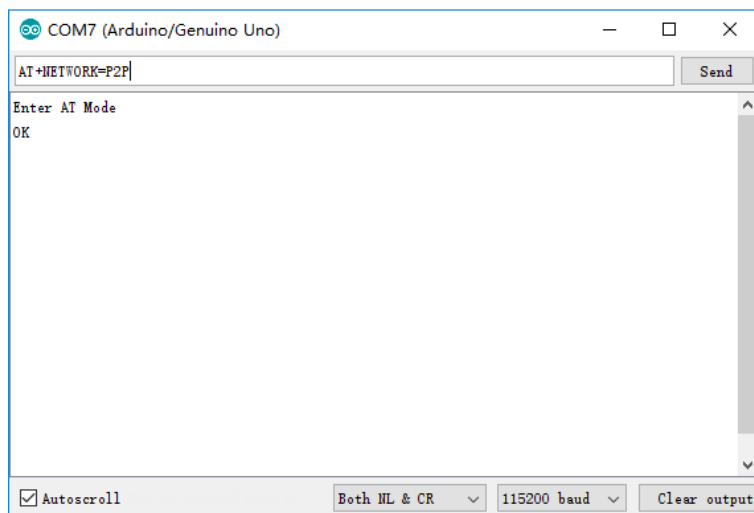
The direct connection between PC and BLE4.1 built-in Bluetooth is not available for temporarily (similar to smartphone, PC also need another software to realize the connection. The development is still in planning)

Bluetooth P2P Communication

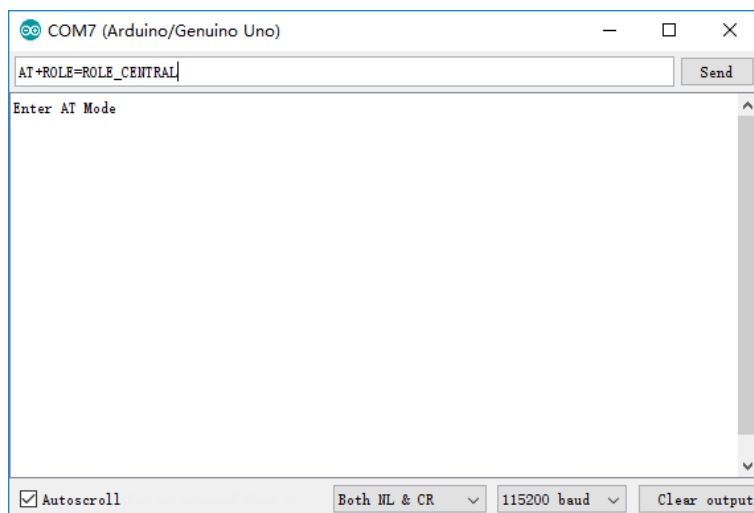
The procedure of BLE4.1 P2P connection is similar to BLE4.0, and the only difference is the P2P mode command.

There are two devices in all, one should be set as the master device(ROLE CENTRAL) and the other one should be set as the slave

device(ROLE_PERIPHERAL). Config two BLE4.1 devices as P2P connection mode with **AT+NETWORK=P2P** (Default: P2P mode).



- AT Command: **AT+ROLE=ROLE_CENTRAL**, **AT+ROLE=ROLE_PERIPHERAL**,



Restart BLE4.1 device and begin the approach connection.

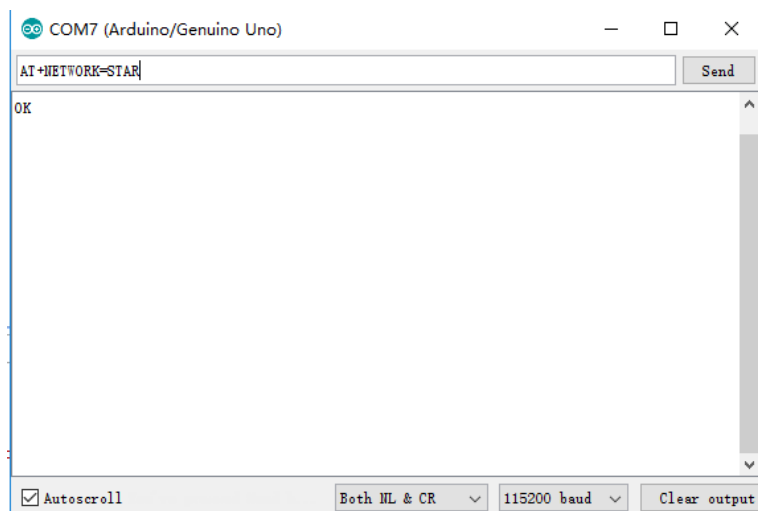
- **Approach Connection:** press and hold host **BOOT** button and move it close to the target slave device (<10cm) until a successful connection, then the **LINK** will light on each 3 seconds

NOTE:

1. The first time you press and hold the host **BOOT** button is only for add the new devices to the white list. It doesn't need the same operation for the second connection.
2. BLE 4.1 device is also compatible with BLE4.0 devices (Bluno 1st generation). And BLE 4.0 only supports P2P connection, so it doesn't need to set the connection mode for 4.0, just BLE 4.1 only.

Bluetooth Star Network Connection

The star network connection mode is similar to the P2P connection mode, the difference between them in setting is that the star network uses the command **AT+NETWORK=STAR**. The light **LINK** suggests the node device has been added to the white list of the central device and then it will connect the listed device automatically.



NOTE:Star network connection is only for BLE4.1 device, not applicable to Bluno 1st generation (BLE4.0) and other BLE brands device.

In the star connection, Bluetooth device adopt special data packet compression method. You can use Arduino library to realize network data transmit. Click the link to download Arduino library files (https://github.com/DFRobot/DFRobot_Bluno2) and Arduinojson library files (<https://github.com/bblanchon/ArduinoJson.git>). During the connection, the slave device which first connected to the host will get ID: 1 and hereby superimposed. The sample code shows how to get network ID of the device and ID of the data source. You'd better to power the device in turn when build star network.


```

#include <DFRobot_Bluno2.h>
#include <ArduinoJson.h>

DFRobot_Bluno2 blunoNet;
int blunoID=0;
void setup()
{
  Serial.begin(9600);
  blunoNet.begin(Serial);//获取id
}

void loop()
{
  uint8_t event=blunoNet.getEvent();//queue
  switch(event)
  {
    case EVENT_NETINFO:
    {
      eventNode e = blunoNet.popEvent();

      StaticJsonBuffer<200> jsonBuffer;
      JsonObject& root = jsonBuffer.parseObject((const char *)e.payload);
      if (!root.success()) {
        return;
      }
      int s = root["r"].size();
      for(int i = 0; i < s; i++){
        if(root["r"][i]["s"].as<int>()){
          blunoID = root["r"][i]["i"].as<int>();
          break;
        }
      }
      break;
    }
    case EVENT_DATA:
    {
      eventNode e = blunoNet.popEvent();
      blunoNet.sendPacket(e.src, blunoID, "hello12345678901234567890",26);
      break;
    }
    default: //no event
    {
      break;
    }
  }
  blunoNet.loop();
  if(Serial.available()){
    char message=Serial.read();
    blunoNet.sendPacket(!blunoID, blunoID, &message,1);
  }
}

```

Bluetooth Low Power Consumption Mode

After Bluetooth enter low power consumption mode, the BLE 4.1 consumes lower than 10uA, and it can still broadcast and keep data interaction. With command **AT+LOWPOWER=ON**, BLE4.1 device will enable low power consumption mode after you reboot the device. BLE4.1 device will enter this mode in 10s. You can also use **AT+BOOT** to make a software reboot. Here is something you should notice: The MCU should **WAKEUP** the BLE Chip first, or it will be messy code. In the low power consumption mode, if there is no extra operation, Bluetooth will enter this mode after 10s. You can awake Bluetooth by interruption pins; awake ATmega328P by Bluetooth:

- BLE4.1 wakes up the Arduino controller (ATmega328P) with P4_2 (D2_Arduino): high voltage wakeup
- Arduino controller (ATmega328P) wakes up BLE4.1 by WAKEUP (D3_Arduino): low voltage wakeup

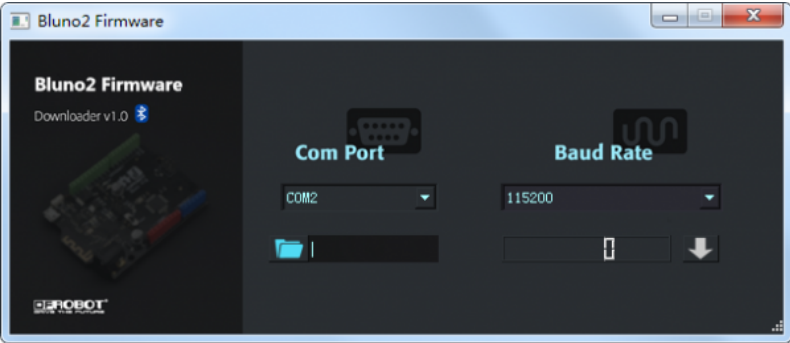
Wireless Programming

Bluno 2 wireless programming is completely compatible with Bluno1 (BLE4.0). And it is only available under P2P connection mode: the host (the central device) to the slave side.

- Similar to P2P connection, set one as host and plug to the PC; set the other one as slave. Pairing the host and the slave, download is available.

Firmware Upgrade

DFRobot BLE4.1 has the firmware method to BLE 4.0, click to download BLE4.1 firmware update tool (https://github.com/DFRobot/Bluno2_Firmware), BLE4.1 firmware address Firmware Download (https://github.com/DFRobot/Bluno2_Firmware).



Update steps are as below:

- Hold BOOT
- Connect BLE4.1 to PC and loose BOOT, CHG and LINK flashes alternatively.
- Open Bluno2 Firmware and select corresponding serial number.
- Choose firmware to update, the firmware format is .img
- Click to download

FAQ

Q&A	Some general Arduino Problems/FAQ/Tips
Q	How to deal with the situation cannot download?
A	There are many reasons for this problem:
1. Check basics, such as power supply, board type and COM port.	
2. Bluetooth will occupy the serial port which will hamper uploading. Please disconnect the current Bluetooth connection, and try again.	
3. The device serial port may be occupied. Please disconnect all peripherals then recompile and download again.	
4. Arduino IDE may crashed. Please disconnect all Bluetooth (include smartphone), close all Arduino IDE software, pull USB line, reopen and test.	
5. Bootloader in the FireBeetle 328P may get lost. Please try re-uploading Bootloader of ATmega328p.	
6. BLE 4.1 firmware may wrong. Please re-upload firmware and check the former chapter for reference.	
Q:	How to choose board when the device manager shows Arduino UNO COM x after connected to PC?
A	The COM name in default is Arduino UNO. Please choose the board according to bootloader, e.g. choose Arduino Pro Mini 3.3V@8MHz for FireBeetle BLE4.1.
Q: How to deal with the Bluetooth ID number varies with connection?	
A: The current Bluetooth ID number is defined with the connection order, ID1, ID2, ID3, ID4 in order and unchangeable. Please connect the host (the central device ID0) in turn.	
Q: How to deal with the Bluetooth ID number varies with connection?	

Q: Why the Bluetooth data been received is gibberish in the baud rate 115200? Q&A	Some general Arduino Problems/FAQ/Tips
--	--

A: The bootloader that FireBeetle BLE4.1 uses is 3.3V Pro Mini of 8MHz oscillator. The Arduino Pro Mini 3.3V@8MHz may runs wrong with high baud rate. The lower baud rate is recommend, e.g. 9600bps	
Q: Does BLE4.1 supports HID, ibeacon and tree network?	
A: The functions HID, ibeacon and tree network are under development.	

For any questions, advice or cool ideas to share, please visit the **DFRobot Forum** (<https://www.dfrobot.com/forum/>).

Turn to the Top