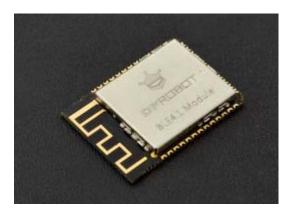# DFRobot BLE4.1 Module SKU: TEL0120

Introduction

BLE4.1 Module is DFRobot newly developed Bluetooth 4.1 transmission module, adhering to the same usage of Bluno, and increasing the star network function, KISS connection (Approach Connection). The new Bluetooth is designed based on BLE 4.1 technology. It has low power consumption and other special features. In the BLE transmission, the point-to-point output rate is 4Kbps, equivalent to the original Bluno twice the rate.

BLE4.1 Module uses DA14681 high-performance BLE chip, it integrates advanced power management system, coming with 3.7V lithium battery charging circuit. It obtains power directly from the USB port. Support 400mA maximum charge current,

Feature

- Support BLE 4.1 master and slave switch
- Support Bluetooth serial data transmission
- Support Bluetooth point-to-point wireless programming
- Support Bluetooth 4.1 multi-device cascade (up to 4)
- Support Arduino Bluetooth library (support for multi-Bluetooth connection)
- Support low power consumption Bluetooth (standby 25uA; interrupt 70uA)
- Two-way interrupt wake up:
- * BLE4.1 wake up controller via **P4_2** (high level wake up)
- * The controller wakes up BLE4.1 via **WAKEUP** (low level wake)
- Support KISS connection method (Approach Connection)
- Support Bluetooth firmware update
- Support lithium battery powered
- Support USB to charge the lithium battery
- LED status indicator

Specification

- Processor: DA14681 (ARM Cortex-M0 CPU)
- Operating Voltage: 3.3V
- USB input Voltage: 4.2V ~ 5.0V
- Lithium battery powered: 3.7V
- Support USB charging (chip comes with charging function)
- Charging current: 10mA ~ 400mA (max)
- Allow external lithium battery capacity: 20mAh ~ 500mAh (max)
- Supports low power consumption: only 30uA / MHz minimum 2uA
- Single I / O port maximum output current: 4.5mA
- Allows I / O port maximum output total current: 100mA
- Supports three network connection modes (P2P connection, star network, tree network [Developing])
- Maximum number of slave connections: 4 (simultaneous connection)
- Maximum transfer rate of 4Kbps
- Flash: 8Mbit
- 64KB OTP
- 128KB Data SRAM
- 16KB Cache SRAM
- 128KB ROM
- On-chip crystal: 16MHz, 32.768KHz
- Three power management modes: Extended sleep, Deep sleep, Hibernation
- 29 digital I / O ports
- 8 analog inputs
- 1 SPI
- 2 I2C (supports 100KHz and 400KHz two communication rates)
- 2 UARTs
- 6 PWM
- I2S / PCM master-slave interface
- IR interface (PWM)
- Support USB1.1 (computer-side simulation into a serial port)
- Built-in temperature sensor
- Transmit power consumption: 3.4mA (TX) 3.1mA (RX)
- Operating temperature: -40 ℃ ~ +85 ℃
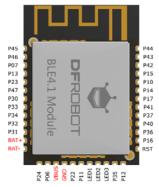- Dimension: 25 * 18 * 3 mm/ 0.98 * .70 * 0.11 inches

Layout

Fig1: DFRobot_BLE4.1_Module PinOUT

BLE 4.1 Module Pin Map

引脚	PWM	ADC	UART	USB	NTC	SWD	SPI	IIC
P06	PWM5	ADC4				SWDIO		
P07		ADC3						
P10		ADC5						
P11				USBN				
P12		ADC0						SDA
P13		ADC2	TX					
P14		ADC1						
P15	PWM0	ADC6						
P16	PWM1				NTC			
P22				USBP				
P23			RX					
P24		ADC7				SWCLK		
P35	PWM2							SCL
P36	PWM3							
P37	PWM4						MOSI	
P40							MISO	
P41							SCLK	

Fig2: DFRobot_BLE4.1_Module PinMap

    Note: VBUS and GND are USB Power port; BAT+ and BAT- are battery power po
rt.

⚠ **Note: There is a 0.1 ohm resistor between BAT- and GND. The battery can also be used directly between BAT + and GND when in use**
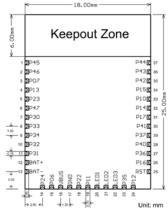
Dimension

- Dimension: 25 × 18 x 3 mm

Fig3: DFRobot_BLE4.1_Module Dimension

AT Command

> ⚠️ To get a 'better' Bluetooth experience on BLE 4.1 modules, we highly recommand you to update the newest firmware 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 <X>
AT+ROLE=<X>	Set Bluetooth mode	"?" Query Bluetooth working mode "ROLE_CENTRAL" Set as the central device "ROLE_PERIPHERAL" Set as the peripheral device
AT+NETWORK=<X>	Set network mode	"?" Query the network mode "P2P" Set to peer to peer connection "STAR" Set to star network
AT+MIN_INTERVAL=<X>	Set the minimum communication interval	"?" Query the current minimum communication interval (10~99999ms)
AT+MAX_INTERVAL=<X>	Set the maximum communication interval	"?" Query the current maximum communication interval (10~99999ms)
AT+BIND=<X>	Set binding Bluetooth address	"?" Query the current binding MAC address e.g.: 0x0017ea9397e1 Set Bluetooth MAC address in need
AT+CMODE=<X>	Set BLE connection mode	"?" Query the current connection mode "UNIQUE" Specify the connection mode "ANYONE" Set to free connection
AT+MAC=<X>	Query the Bluetooth address	"?" Query the current Bluetooth MAC address
AT+NAME=<X>	Set the name	"?" Query the Bluetooth device name "DFBLEduino2"
AT+SETTING=<X>	Modify/Reset to default <X> mode	"?" Query the current setting "DEFAULT" Reset to initial setting "DEFPERIPHERAL" Reset to initial slave setting "DEFCENTRAL" Reset to initial host setting
AT+USB_MONITOR=<X>	Turn the USB monitor ON/OFF	"?" Query the current USB monitor "ON" Turn on the USB monitor

		"OFF" Turn off the USB monitor
AT+HELP=\<X>	Get help from AT command	"ALL" Query all AT commands help information
AT+NET_INF=\<X>	Query device information in the current network	"?" Query device network information
AT+VERSION=\<X>	Query the firmware version	"?" Query firmware version information
AT+RSSI=\<X>	Query the RSSI value	"?" Query the RSSI value
AT+P\<n>=\<X>	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
AT+WL=\<X>	Query/Declare the White List	"?" Query the White List "NONE" Declare the White List
AT+LOWPOWER=\<X>	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
AT+EXIT	Exit the AT command mode	Exit the AT command mode

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)

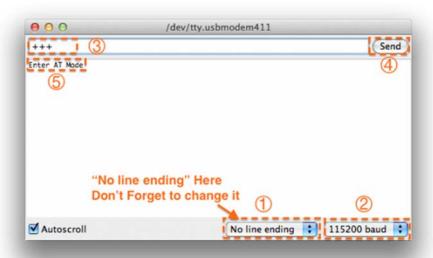6. If the AT Command Mode is successfully access , you will receive "Enter AT Mode"(⑤) from it.

Fig1: +++ enter the AT CMD Mode

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)

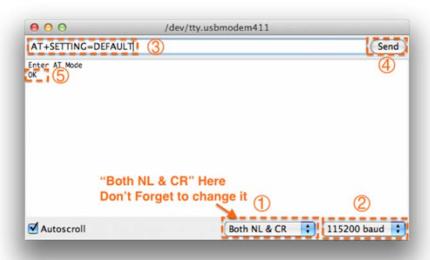9. If the BLE is successfully configured , you will receive "OK"(⑤) from it.

Fig1: enter the AT command, remember selecting the Both NL & CR

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
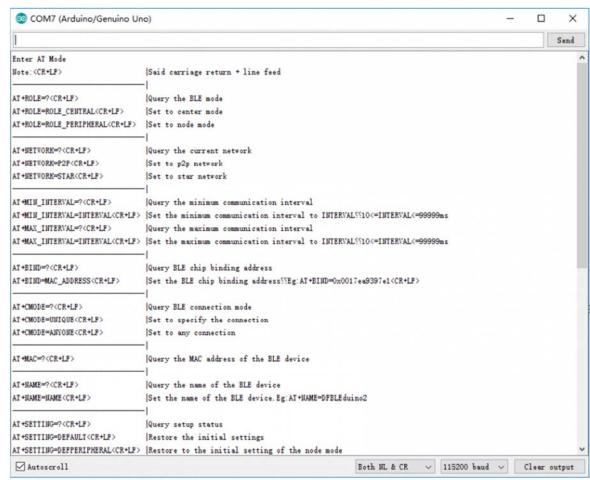
Fig1: Enter AT+HELP=? to get AT Command List

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 consum[ption 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
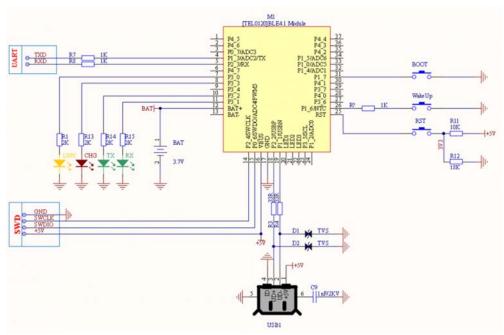
Fig1: BLE4.1 Minimum System Schematic

Smartphone Bluetooth Connection

- Android Terminal and IOS Terminal
- BlunoBasicDemo Android, Android source code (DFRobot)
- LightBlue IOS
- Blynk IOS

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).
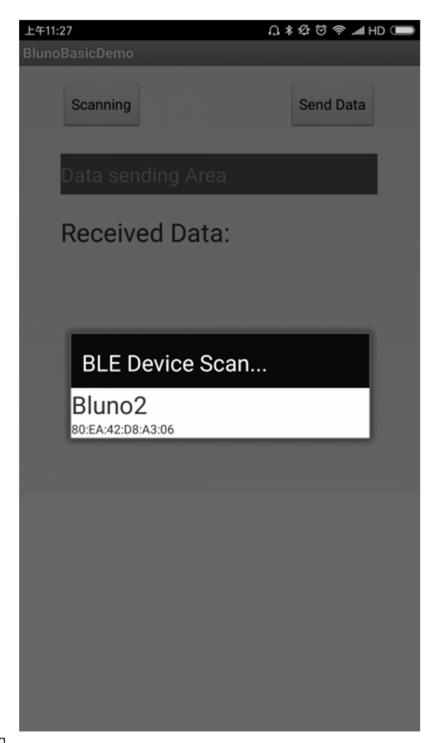- Open **BunoBasicDemo** app, click **SCAN**, then we'll see the scanned BLE4.1 device .

Fig1: Scanning

- Click the device to connect. Once it connected successfully, it will show Connected and LINK will light for 3s and blink for every 3s.
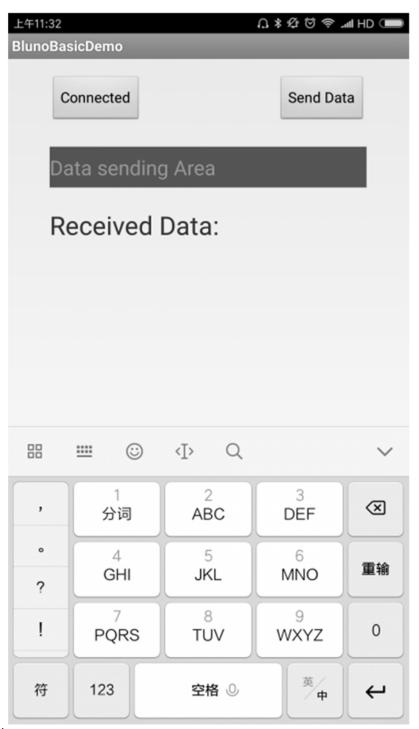
Fig2: Connected

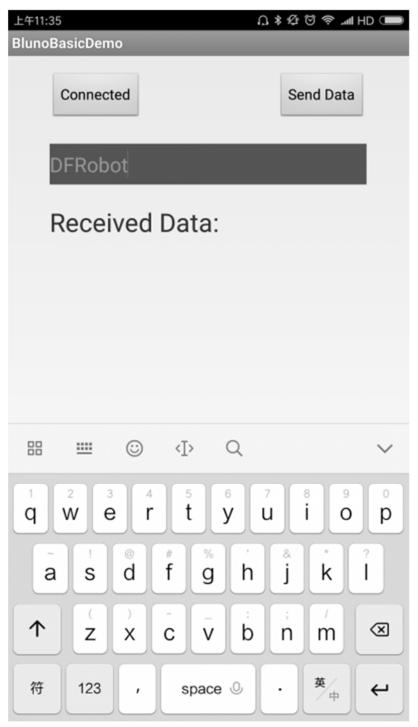- Send data frome the **Data sending Area**

Fig3: Send Data
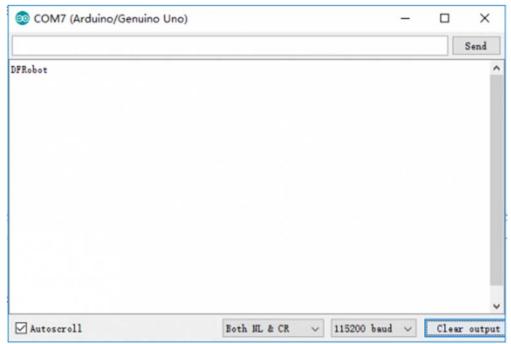
- Click **Send Data** to send data.

Fig4: Print what it received

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).
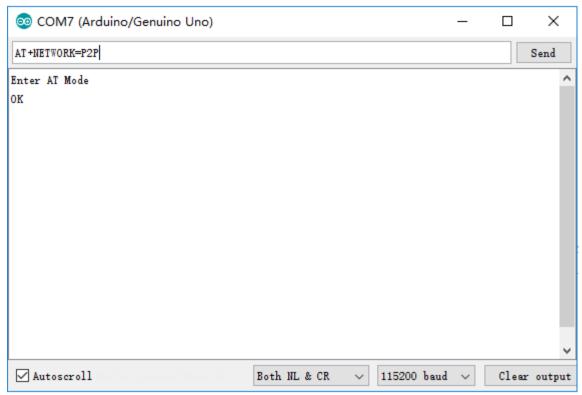
Fig1: set the connection mode

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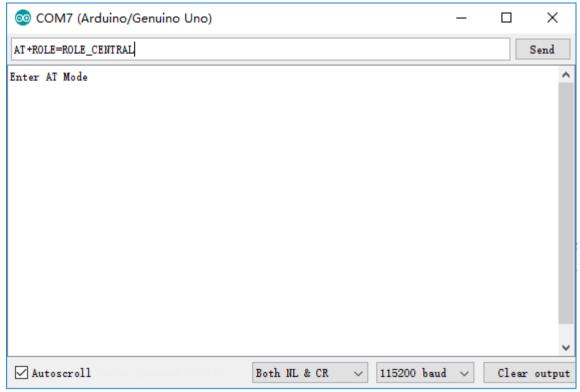

Fig2: set the device as central mode

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.
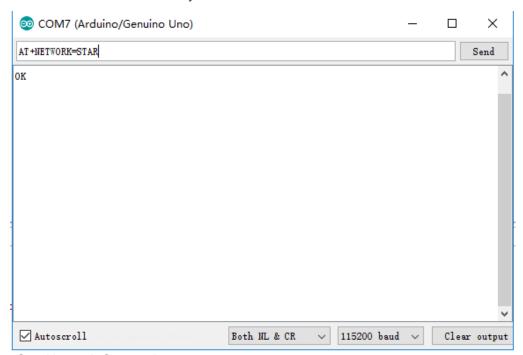

Fig1: Set Star Network Connection

⚠️ **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 and Arduinojson library files. 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.

```cpp
#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, BLE4.1 firmware address Firmware Download。
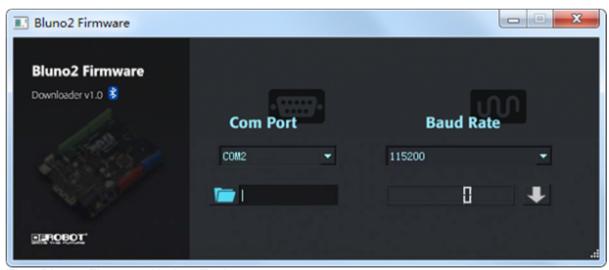

Fig1: Bluno2 Firmware Update Tool

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

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 ATmaga328p.
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, DI4 in order and unchangeable. Please connect the host (the central device ID0) in turn.

Q: Why the Bluetooth data been received is gibberish in the baud rate 115200?

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**.