(https://www.dfrobot.com/product-1646.html)

Introduction

DFRobot FireBeetle series are low power consumption controllers designed for Internet of Things (IoT) development. This Bluetooth controller combines Bluetooth Low Energy 4.1 with Arduino controller, offering functions similar to the former Bluno series but with lower power consumption.

FireBeetle BLE 4.1 is compatible with Bluno series (BLE 4.0) and performs much better. It equipped with more stable DIALOG Bluetooth 4.1 chip, supporting maximum 4Kbyte/s transmission rate and low power consumption (25uA POWER DOWN). It can be directly applied to low power consumption application such as wearable and intelligent tools and so on.

It also supports multiple Bluetooth cascade, KISS connection(approach connection) and compatible with Bluno series. Its connection can be realized by holding host BOOT and moving to the target peripheral device.

Features

- Support Arduino IDE program

- Support BLE 4.1 central and peripheral switch

- Support Bluetooth serial port data transmit

- Support Bluetooth peer to peer (P2P) wireless programming

- Support Bluetooth 4.1 multiple device cascade (maximum:4)

- Support Arduino Bluetooth library (support multiple Bluetooth connection)

- Support low power consumption Bluetooth (standby current: 25uA; enabled interrupt current: 70uA)

- Double interruption to wake up:
    - BLE4.1 awake Atmage328P by D2 (high level awake)
    - Atmage328P awake BLE4.1 by D3 (low level awake)

- Support KISS connection(approach connection)

- Support Bluetooth firmware update

- Support USB charge

- Support lithium battery power supply

- Equipped with status LEDs

Specification

- Processor: ATMEGA328P

- Bootloader: Arduino Pro or Pro Mini, ATmega328 (3.3V, 8MHz)

- Operating Current: 3.3V

- VCC I/O Voltage: 3.7V~5.5V

- Lithium Battery Voltage: 3.7V

- Support maximum 4Kbyte/s transmission rate (more device, lower rate)

- Support maximum 4Kbyte/s transmission rate (more device, lower rate)

- Support low power consumption: 25uA (POWER DOWN); 70uA (enabled interruption)

- Support maximum discharge current: 600mA (LDO-3.3V OUTPUT)

- Support maximum charge current: 400Ma

- Digital Interfaces: 14

- Analog Interfaces: 6

- SPI: 1

- I2C: 1

- UART: 1

- LED_BUILTIN: D13

- Interface Mode: 2.54mm (Pin/Female header), soldering free in default

- Dimension: 58 * 29 mm/ 2.28 * 1.14 inches

Board Overview
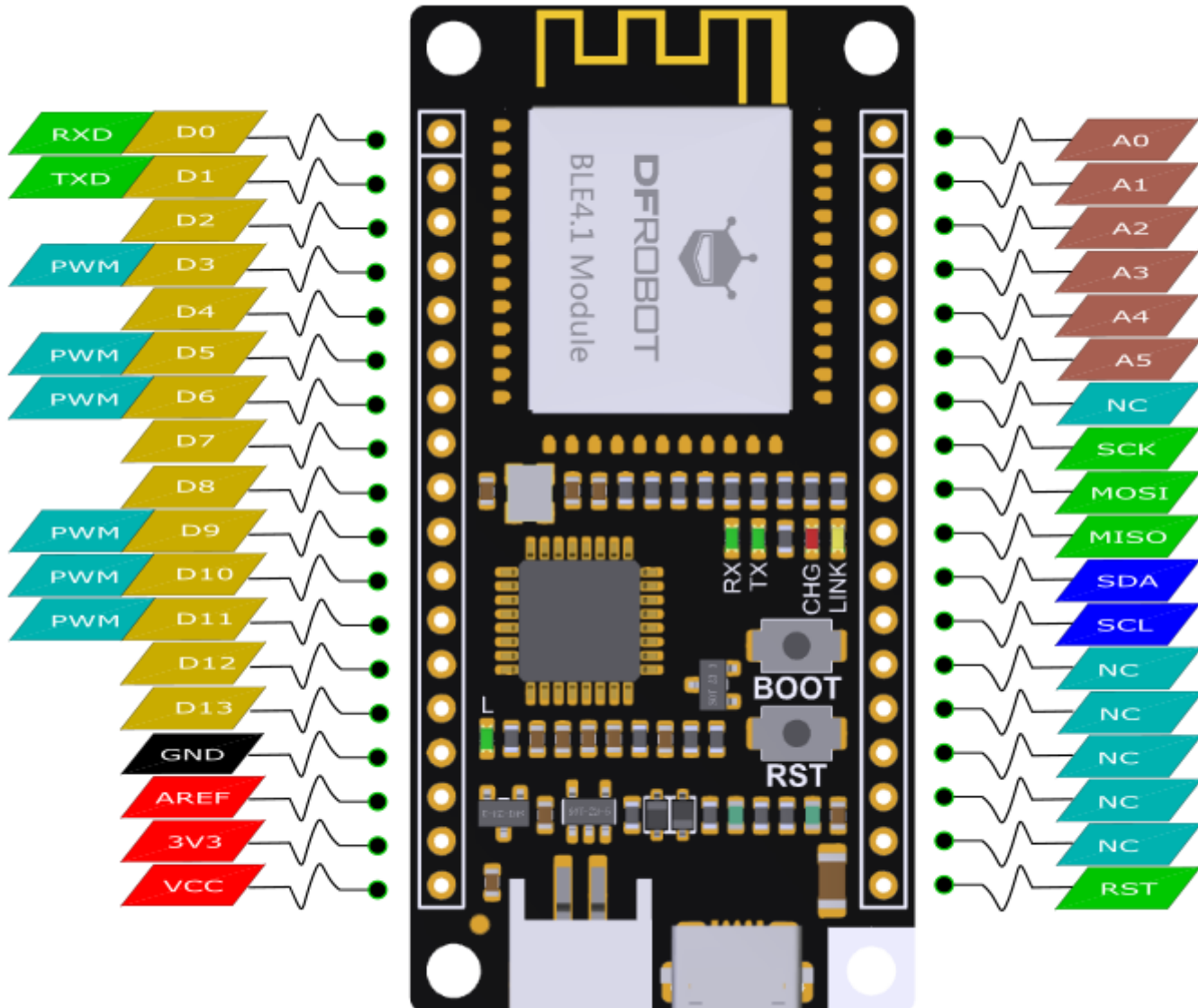
LED indicators

- RX: Serial data receiving indicator, light up when receive serial data
- TX: Serial data transmitting indicator, light up when send serial data
- CHG: Battery charge indicator: Flash without lithium battery; Light ON in charging; Light OFF when full charged
- LINK: a Bluetooth connect indicator; when Bluetooth connect to the device successfully, LINK will light for 3s and blink for every 3s, light OFF when disconnected.

Buttons

- **BOOT**: Bluetooth firmware update; enable KISS connection (approach connection).
    - Firmware update: please press and hold **BOOT** button and power-up (plug in USB), turning to the firmware update mode. For any enquiries, please refers to the part: Firmware Update.
    - Approach connection: please hold the central **BOOT** button when it is in the working mode, approaching the peripheral Bluetooth device and connect.
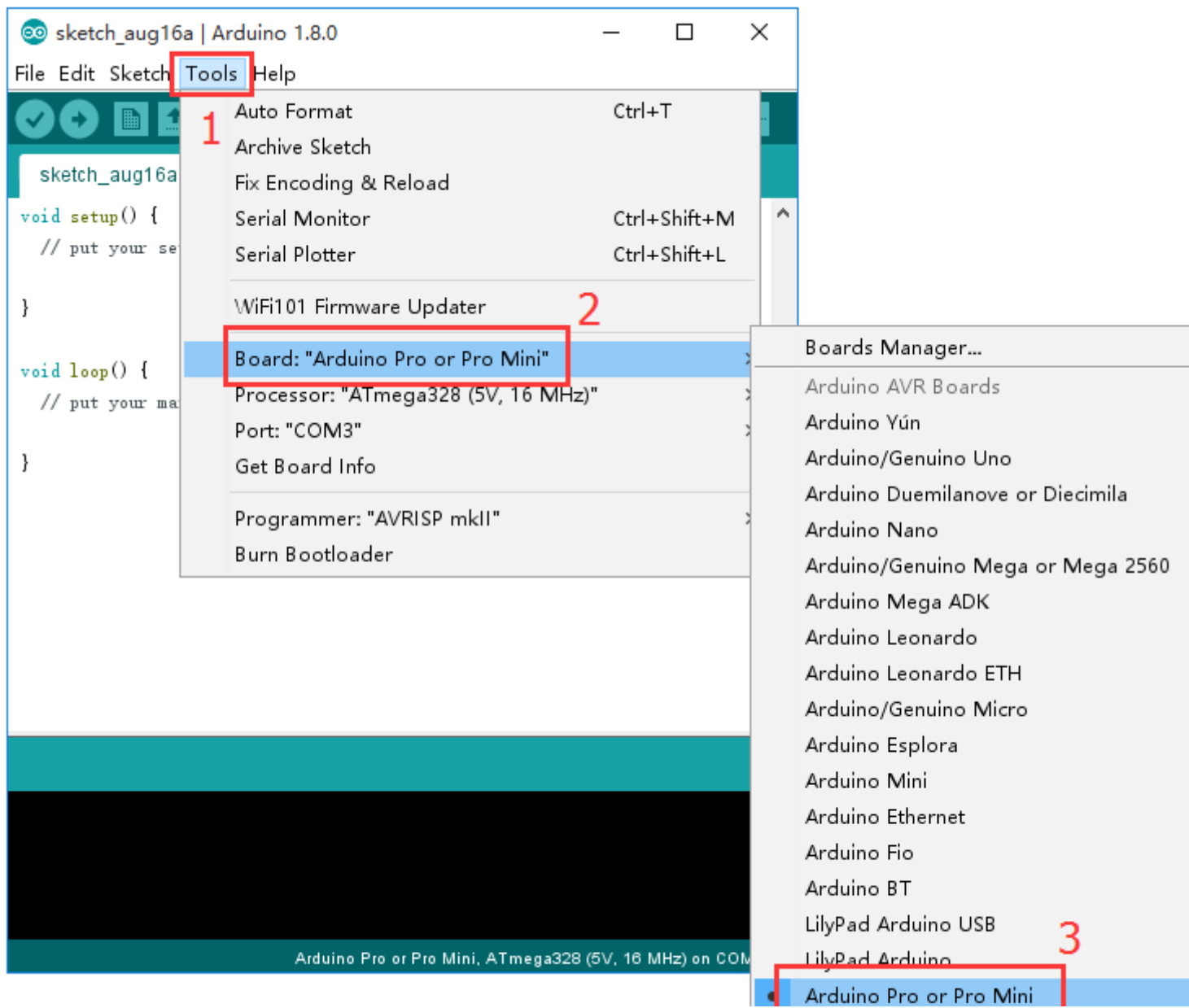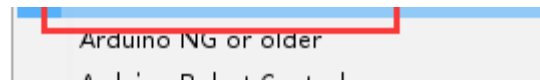- **RST** Arduino Reset button

> ⚠️ **NOTE**: VCC is the output voltage of the power supply (USB power supply: 5V, lithium battery power supply: 3.7V)

Arduino Tutorial

Choose a suitable Arduino board

Considering low power consumption, we select **Arduino Pro Mini @3.3V 8MHz** as Bootloader for FireBeetle BLE4.1 microcontrollers.

Arduino NG or older

> ⚠️ **SPECIAL NOTE**: Pro Mini uses 8MHz OCXO. Its processing speed is relatively slow for 115200-baud rate. The high baud rate may lead to unreadable code and packet loss etc. 9600-baud rate is recommended.

Low Power Consumption Mode

BLE 4.1 switch to low power consumption mode with AT command: **AT+LOWPOWER=ON**. The Bluetooth can still broadcast and connect in the low power mode. When the Bluetooth received data, it will awake automatically and send an interrupt signal to the chip: 328p. The interrupt pin here should be connected to D2 of 328p which is interrupt 0 in the code.

```
void wakeup(){
  sleep_disable();
  delay(2000);
}
void lowpower(){
 ADCSRA &= ~(1<<ADEN);/*turn off ADC*/
 TWCR &= ~(1<<TWEN);/* turn off TWI*/
 delay(10);
 set_sleep_mode(SLEEP_MODE_PWR_DOWN);/*set sleep mode to power down mode*/
 sleep_enable();/*enable sleep mode */
 MCUCR |= (1<<BODS|1<<BODSE);
 MCUCR = MCUCR & (~(1<<BODSE)) | (1<<BODS);
 sleep_cpu();/*enter sleep mode */
}
void setup() {
 attachInterrupt(0, wakeup, CHANGE);
 /* set interrupt method to trigger D2 to voltage change, callback function is wakeup( ), which means awake functions of tl
 lowpower();
}
```

```
void loop() {
 lowpower();
 delay(2000);
}
```

- In the low-power mode, the whole power consumption of the FireBeetle Board-328P with BLE4.1 board is about 70uA.
- Turn off the interrupt and other peripherals, the whole power consumption of the FireBeetle Board-328P with BLE4.1 board is about 25uA.

AT Commands

AT commands are communication instructions used to configure Bluetooth parameter. AT is the abbreviation of ATtention. Every command line starts with "AT" and ends with a carriage return character and a linefeed character<CR+LF>.

Note: <CR+LF> only stands for carriage return and linefeed character, not part of the AT command.
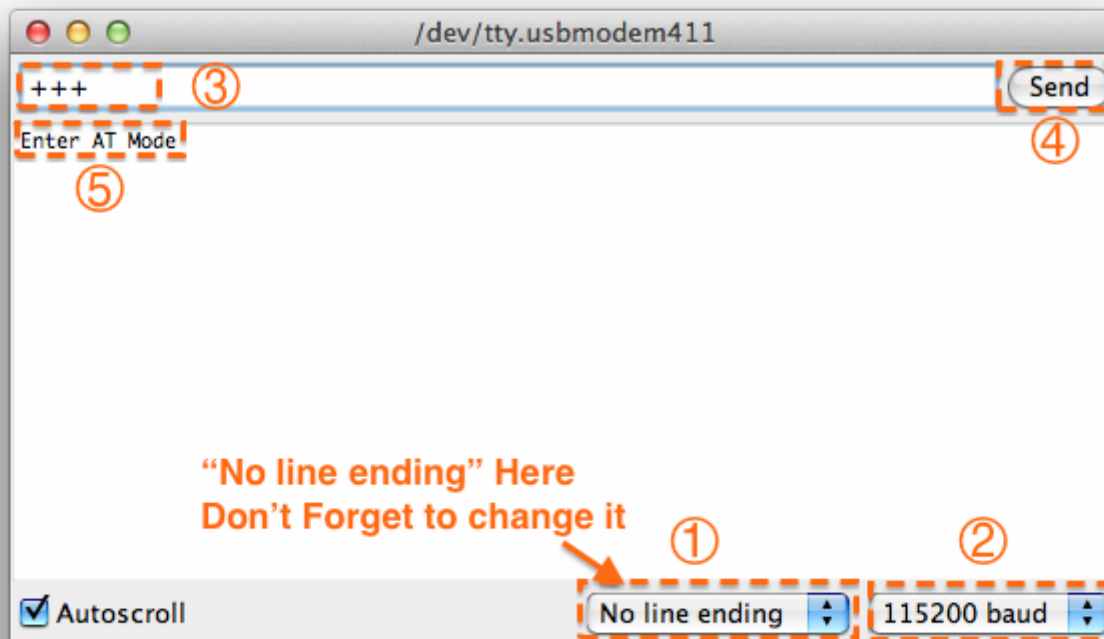
AT Commands Set

AT command	Functions	Parameter
AT+ROLE=	set Bluetooth mode	"?"query the current BLE mod;e "ROLE_CENTRAL"set to center mode; "ROLE_PERIPHERAL" set to node mode
AT+NETWORK=	set network mode	"?"query the current network; "P2P"set to p2p network; "STAR"set to star network
AT+MIN_INTERVAL=	set the minimum communication interval	"?"query the minimum communication interval(10~99999ms)
AT+MAX_INTERVAL=	set the maximum communication interval	"?" query the maximum communication interval (10~99999ms)
AT+BIND	set binding Bluetooth	"?"query the current binding MAC address. e.g.: 0x0017ea9397e1 Set Bluetooth

AT+BIND=	address	MAC address in need
AT+CMODE=	set BLE connection mode	"?" Query the current connection mode "UNIQUE"set to specify the connection; "ANYONE"set to any connection

AT command	Functions	Parameter
AT+MAC=	set the Bluetooth address	"?"query the MAC address of the BLE device
AT+NAME=	set the name	"?"query the name of the BLE device "DFBLEduino2"
AT+SETTING=	change/reset the defalut mode	"?"query the current setting; "DEFAULT" reset to the initial settings; "DEFPERIPHERAL" reset to the initial setting of the node mode; "DEFCENTRAL" reset to the initial setting of the center mode
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
AT+HELP=	get help from AT command help information	"ALL" query all AT commands help information
AT+NET_INF=	query the device information of the current network	"?"query the device information of the current network
AT+VERSION=	query the firmware version	"?"query the current firmware version
AT+RSSI=	query the RSSI value	"?"query the current RSSI value
AT+P=	query/set the P(n, n=0, 1, 2...)status	"?"set IO pins to inout mode and query its level; "1"set the pin to output mode and output HIGH"; "0" set the pin to output mode and output LOW
AT+WL=	query/clear white list	"?"query white list;"NONE"clear white list
AT+LOWPOWER=	query/set the low-power mode	"?"query the current low-power mode; "ON" turn on the low-power mode; "OFF" turn off the low-power mode

	mode	"OFF" turn on 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).

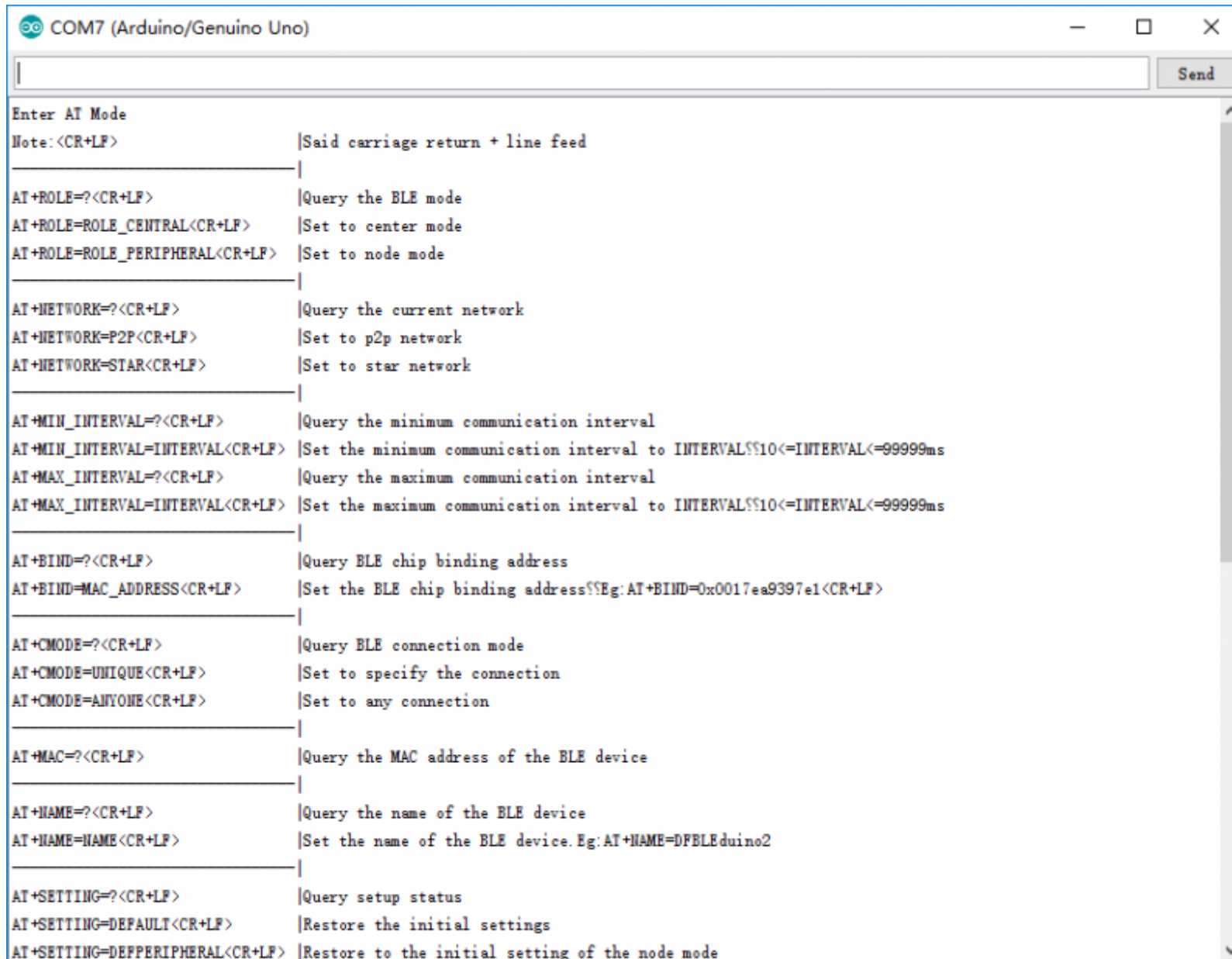6. If the AT Command Mode is successfully access, you will receive "Enter AT Mode"(⑤) from it.

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.

```
COM7 (Arduino/Genuino Uno)                                              —    □    ×

|                                                                          Send

Enter AT Mode
Note:<CR+LF>                    |Said carriage return + line feed
_____ |
AT+ROLE=?<CR+LF>                |Query the BLE mode
AT+ROLE=ROLE_CENTRAL<CR+LF>     |Set to center mode
AT+ROLE=ROLE_PERIPHERAL<CR+LF>  |Set to node mode
_____ |
AT+NETWORK=?<CR+LF>             |Query the current network
AT+NETWORK=P2P<CR+LF>           |Set to p2p network
AT+NETWORK=STAR<CR+LF>          |Set to star network
_____ |
AT+MIN_INTERVAL=?<CR+LF>        |Query the minimum communication interval
AT+MIN_INTERVAL=INTERVAL<CR+LF> |Set the minimum communication interval to INTERVAL§§10<=INTERVAL<=99999ms
AT+MAX_INTERVAL=?<CR+LF>        |Query the maximum communication interval
AT+MAX_INTERVAL=INTERVAL<CR+LF> |Set the maximum communication interval to INTERVAL§§10<=INTERVAL<=99999ms
_____ |
AT+BIND=?<CR+LF>                |Query BLE chip binding address
AT+BIND=MAC_ADDRESS<CR+LF>      |Set the BLE chip binding address§§Eg:AT+BIND=0x0017ea9397e1<CR+LF>
_____ |
AT+CMODE=?<CR+LF>               |Query BLE connection mode
AT+CMODE=UNIQUE<CR+LF>          |Set to specify the connection
AT+CMODE=ANYONE<CR+LF>          |Set to any connection
_____ |
AT+MAC=?<CR+LF>                 |Query the MAC address of the BLE device
_____ |
AT+NAME=?<CR+LF>                |Query the name of the BLE device
AT+NAME=NAME<CR+LF>             |Set the name of the BLE device.Eg:AT+NAME=DFBLEduino2
_____ |
AT+SETTING=?<CR+LF>             |Query setup status
AT+SETTING=DEFAULT<CR+LF>       |Restore the initial settings
AT+SETTING=DEFPERIPHERAL<CR+LF> |Restore to the initial setting of the node mode
```
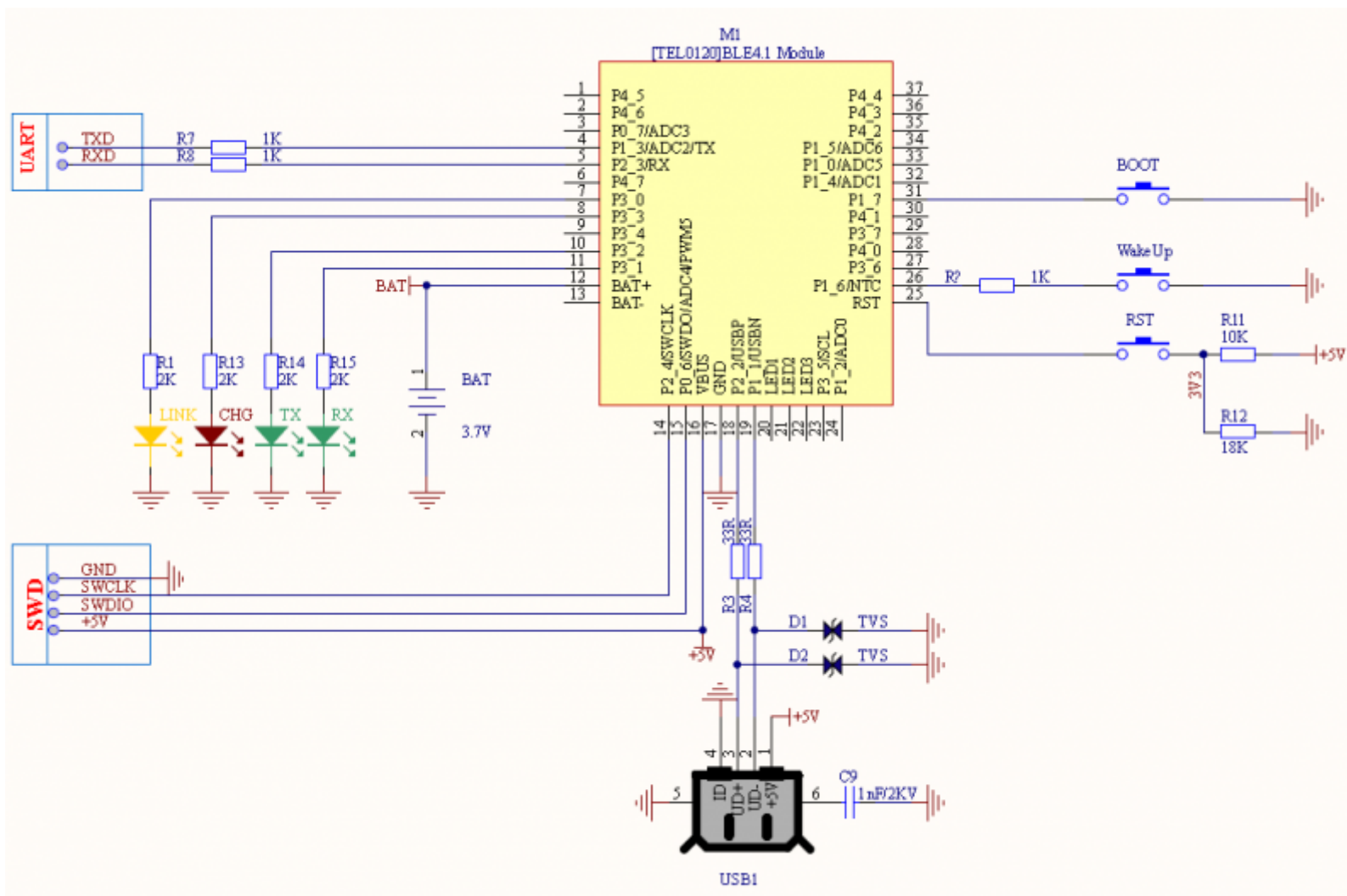
☑ Autoscroll   Both NL & CR ∨   115200 baud ∨   Clear output

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
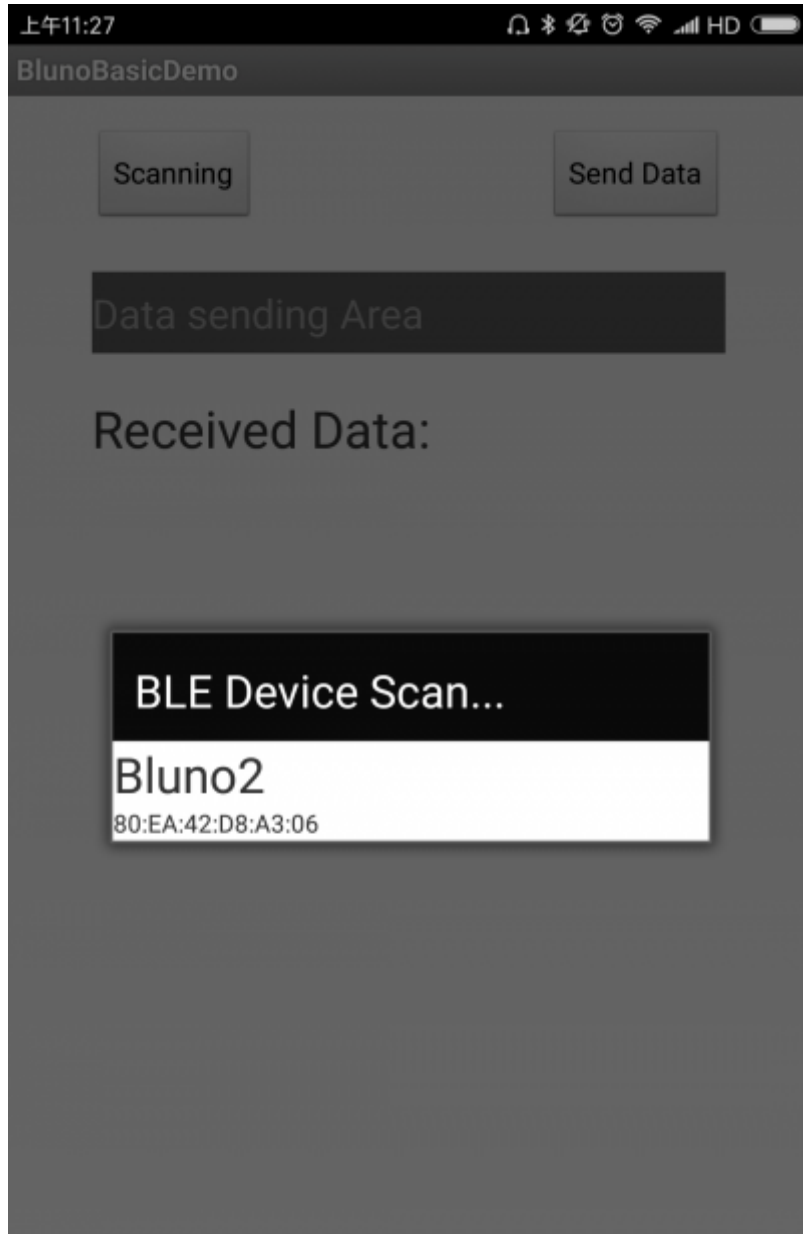
Smartphone Bluetooth Connection

- Android Terminal (https://dfimg.dfrobot.com/nobody/wiki/6d4122cad8b76ec9018dd9089b682a07.apk) and IOS Terminal

(https://itunes.apple.com/us/app/wan-zhuan-bu-lu-nuo/id737793451?ls=1&mt=8)

- BlunoBasicDemo Android (https://dfimg.dfrobot.com/nobody/wiki/b3c77e3cc84b6f19090753a477a887bb.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 .
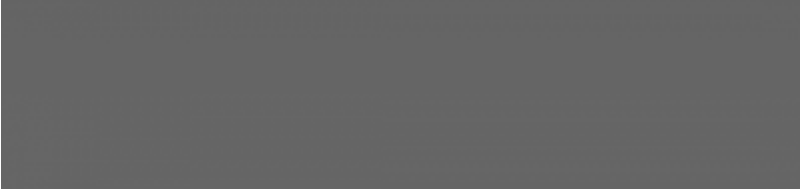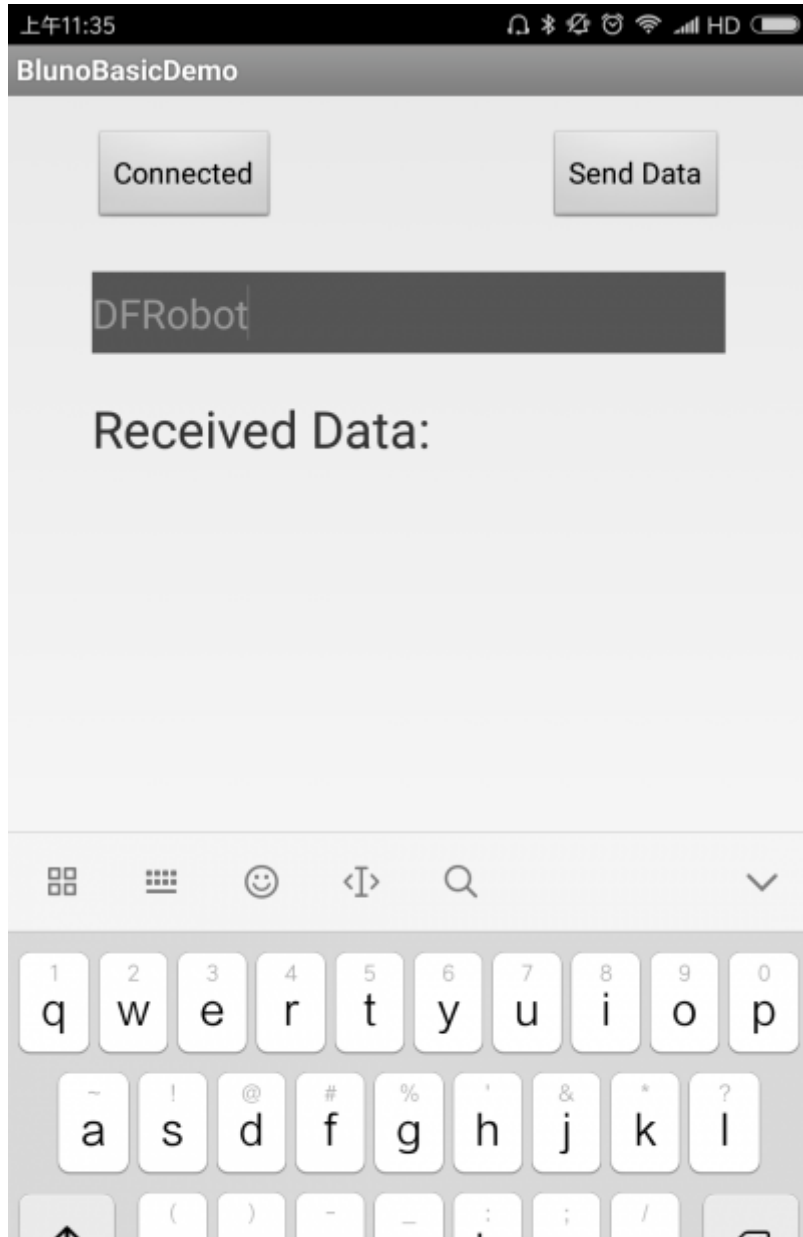
- Click the device to connect. Once it connected successfully, it will show Connected and LINK will light for 3s and blink for every 3s.
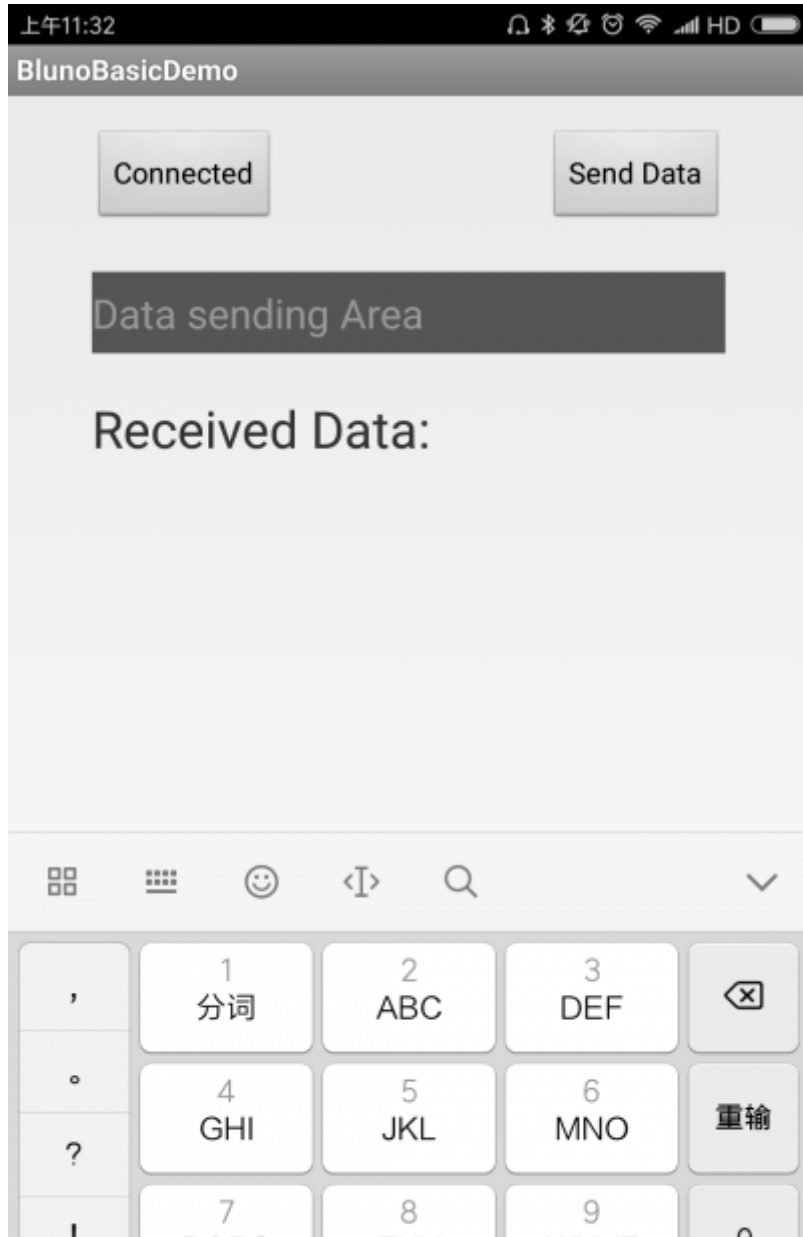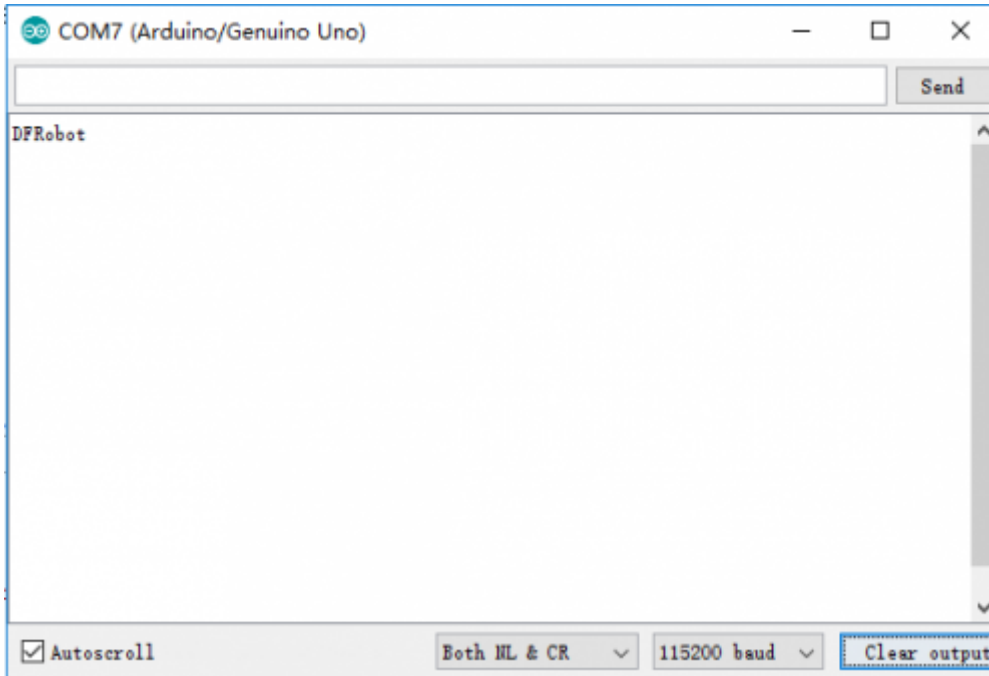
- Send data frome the **Data sending Area**

- 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)

Config two BLE4.1 devices as P2P connection mode with **AT+NETWORK=P2P** (Default: P2P mode).
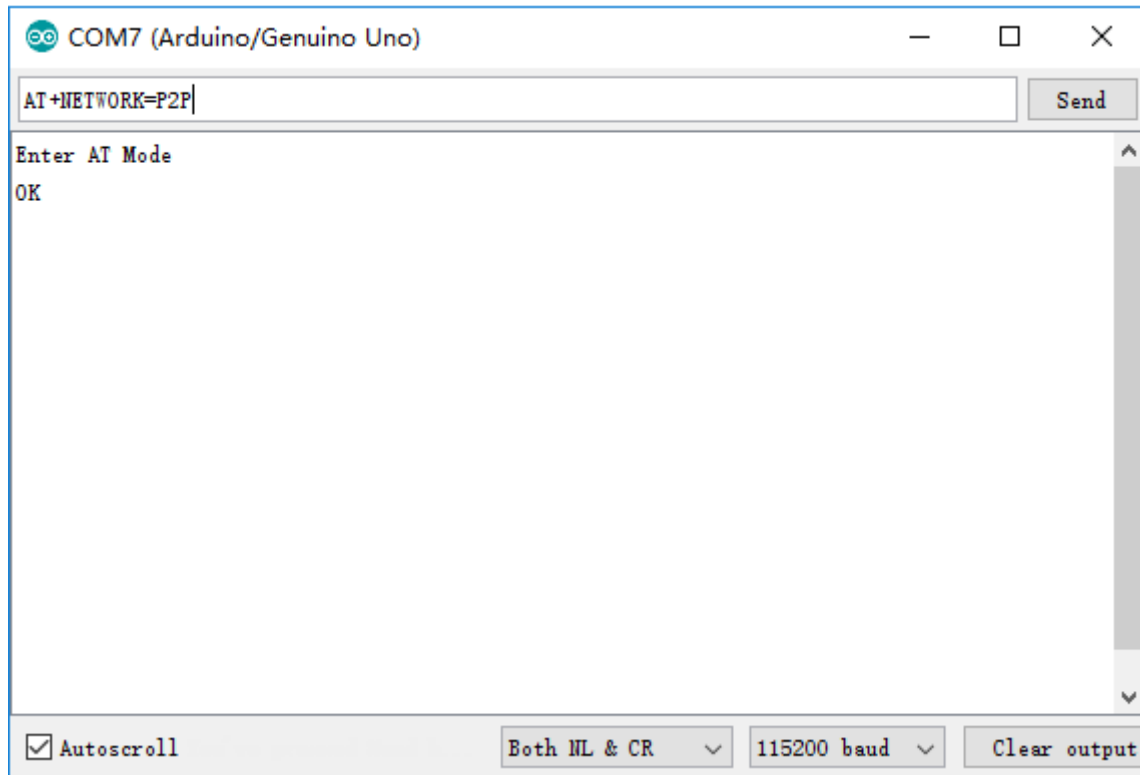
- AT Command: **AT+ROLE=ROLE_CENTRAL**, **AT+ROLE=ROLE_PERIPHERAL**,
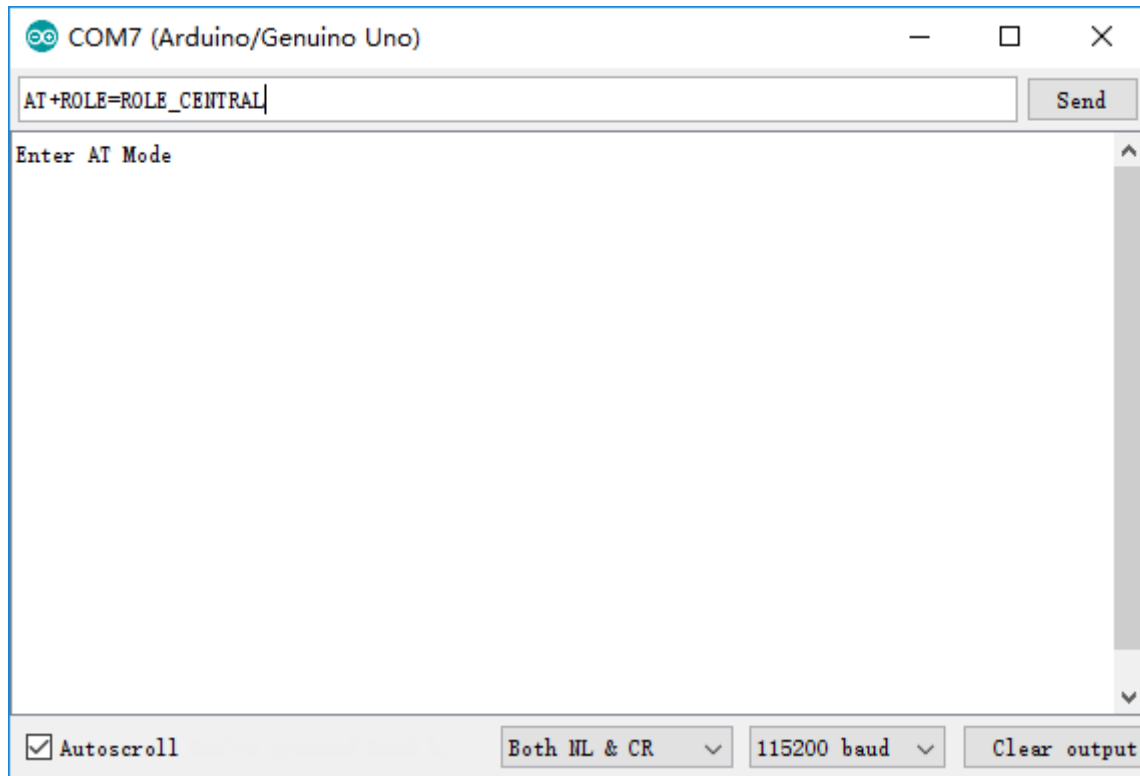
Restart BLE4.1 device and begin the approach connection.

- **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
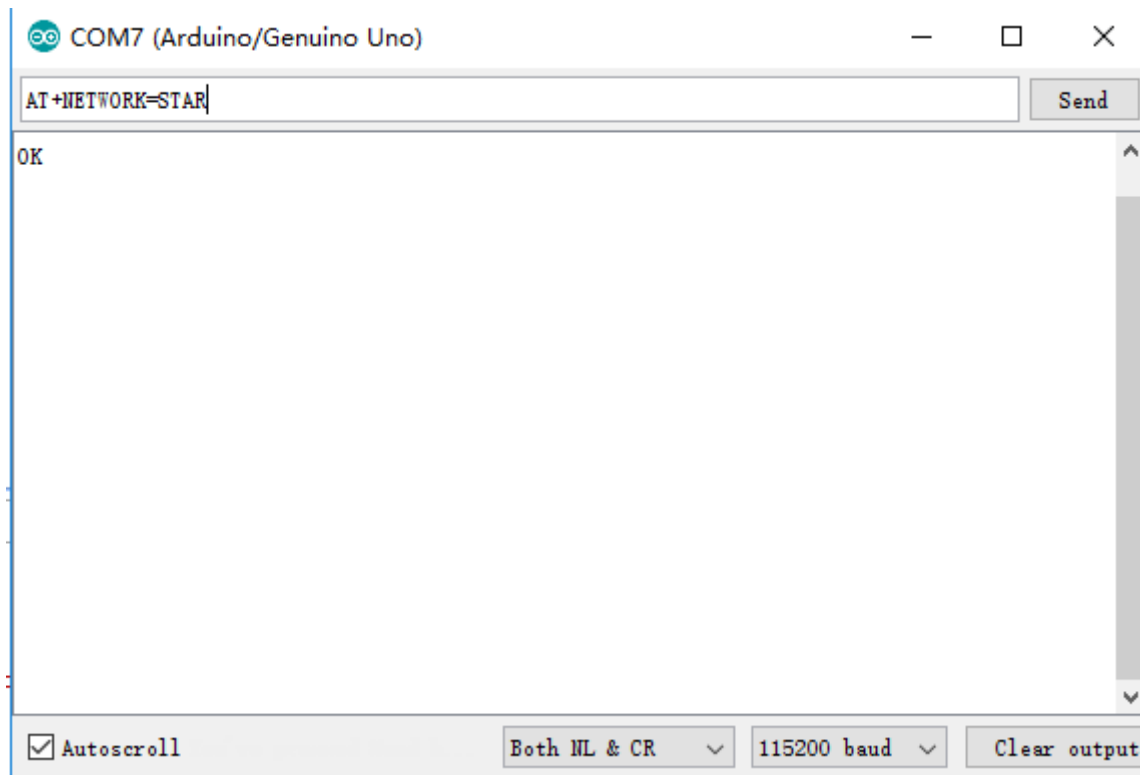
NOTE:

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

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);//get 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;
        }
```

```
            ʃ
            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).
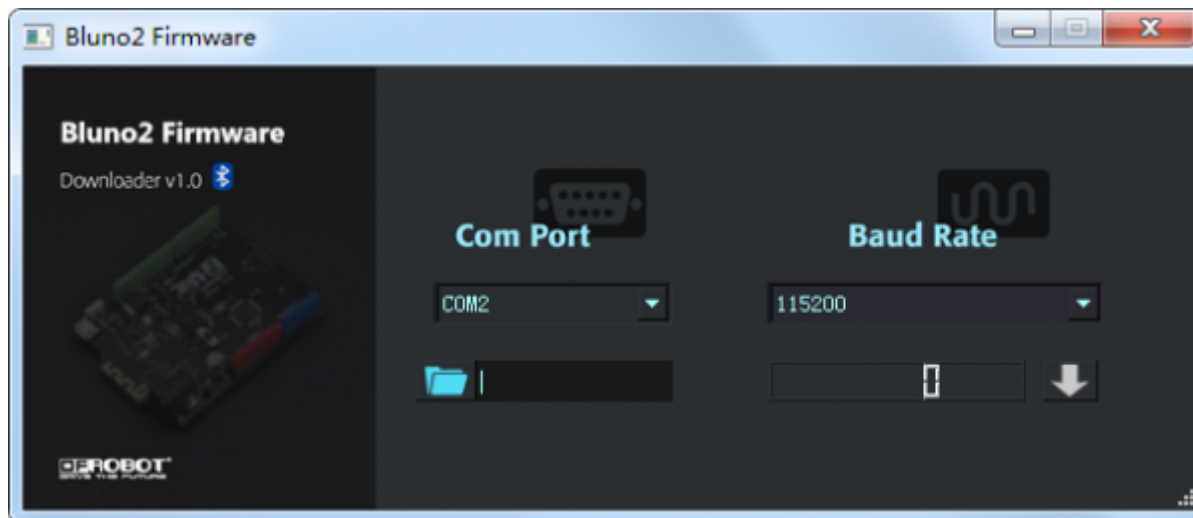
Update steps are shown as below:

Update steps are shown 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: 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 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 Mi
```

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.

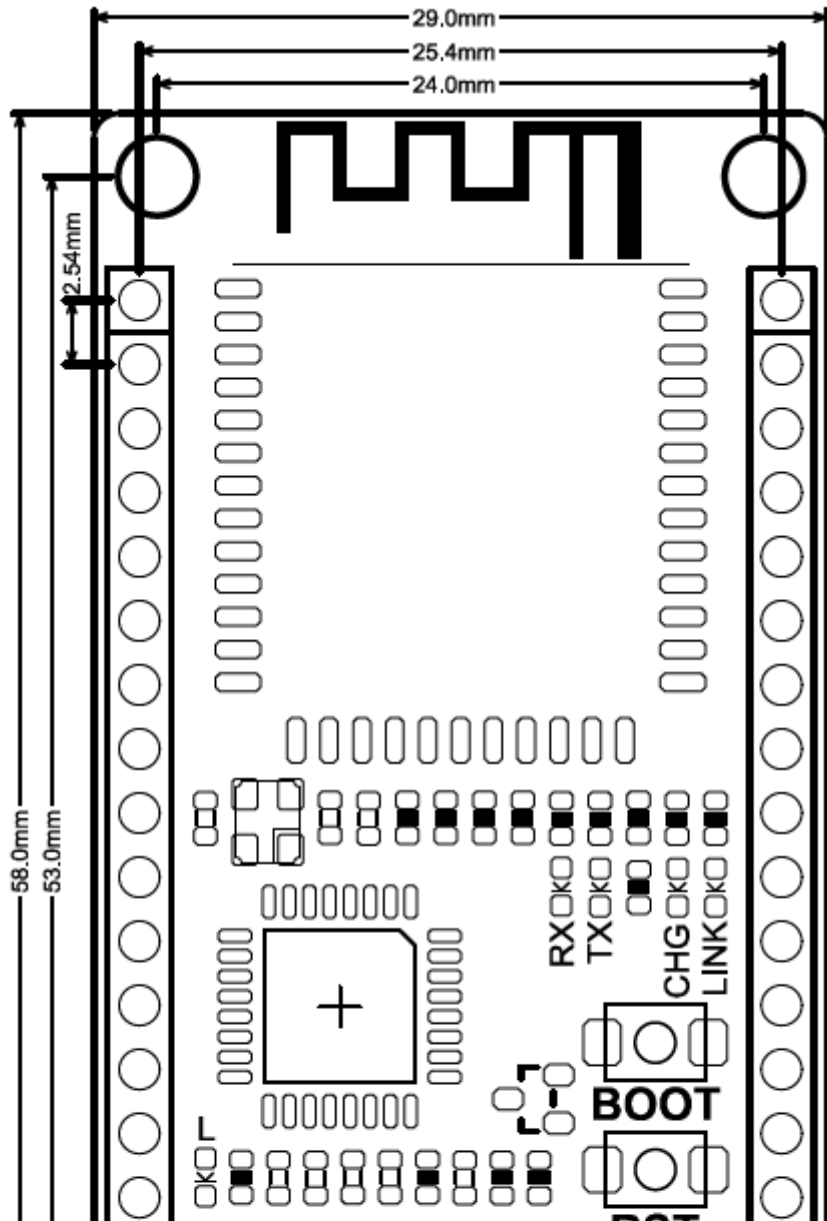**Q:** Why the Bluetooth data been received is gibberish in the baud rate 115200?

A: The bootloader that FireBeetle BLE4.1 uses is 3.3V Pro Mini of 8MHz oscillator. The Arduino Pro Mini 3.3V@8MHz may
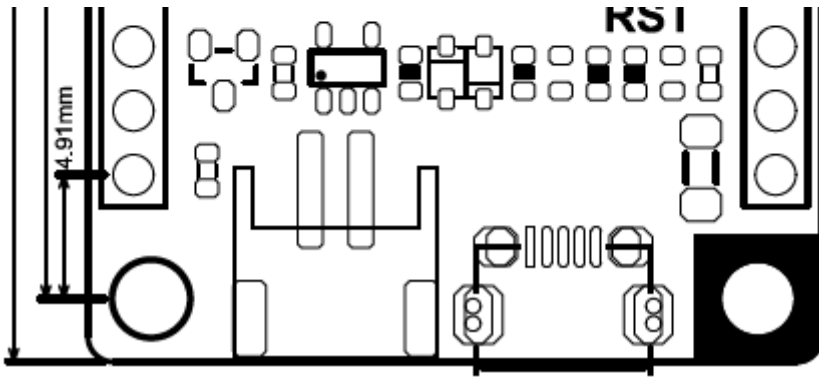
Q Does BLE4.1 supports HID, ibeacon and tree network?

A: The functions HID, ibeacon and tree network are under development.

Dimension Diagram

- Pin Pitch: 2.54mm
- Mounting Hole Space: 24mm/53mm
- Mounting Hole Size: 3.1mm
- Main-Board Size: 29.00mm×58.00mm
- Thickness: 1.6mm

More Documents

- DIALOG-DAL468 Datasheet (https://www.dfrobot.com.cn/images/upload/File/20170927153142zeu06u.pdf)

🛒 Get **FireBeetle Board-328P with BLE4.1_SKU_DFR0492** (https://www.dfrobot.com/product-1646.html) from DFRobot Store or **DFRobot Distributor.** (https://www.dfrobot.com/distributor)

Turn to the Top