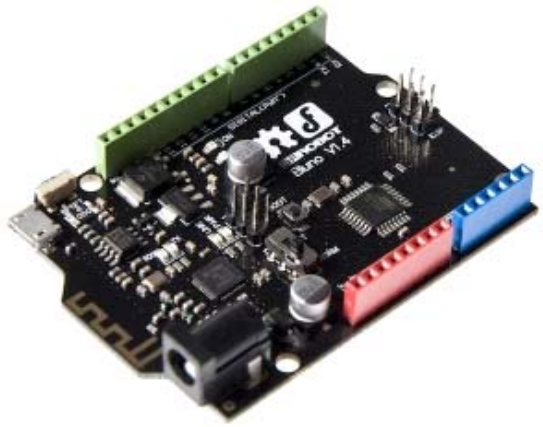
Bluno SKU:DFR0267



Contents

Introduction

It's time to get **Bluetooth 4.0** into your project, together with your phone! For aficionados of smart devices and wearables, now you can go further than hacking things bought in the market to building your own prototype out of garage. The Bluno board is the first Arduino board intergrating BT 4.0(BLE) module, making it an ideal prototyping platform for both software and hardware developers to go wireless. You will be able to develope your own smart bracelet , smart pedometer and so on. Through the low- power Bluetooth 4.0 technology, real-time low energy communication can be made

really easily. What's more, we also developed the App for the Bluno (both Android and IOS), and they are completely **opensource**, so that you can modify and develope your own BLE-hardware platform.

For the demo application and Arduino code, we integrated dfrobot wireless libraries for the beginners. The idea is to offer a simple way for you to use wireless modules without learning the sophisticated wireless comunication protocol. However, for the developer, recommend to custom or choose the protocol according to the product features or the application.

By the way, except of many other BLE products, we also developed the Accessory Shield for Bluno for you to help you build your idea faster and easier, enjoy your wireless journey!

Keep Your Bluno Updated

We have released a new version bootloader which is much more stable than the last version. Especially that it can resist in insufficient power supply, motor magnetic-field interferenceit, etc. Visit forum to find how, How to upgrade DF BLE device bootloader to 2.0?

- **To avoid getting your BLE card defective (bootloader lost)**, please read Common Arduino Operation Notes, NO.1

Specification

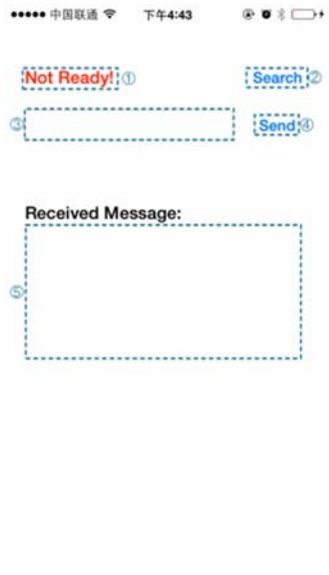
Basic	Feature
On-board BLE chip: TI CC2540 Baud Rate: 9600 - 115200 bps Transmission range(Open Space): more than 70m Transmission range(Office): about 20m Power Supply:USB Powered or External 7V~12V DC Output Current (I/O pin): 40mA Output Current (Power pin): 200mA Microcontroller: Atmega328 Bootloader: Arduino Uno Compatible with the Arduino Uno pin mapping Size: 60mm * 53mm Weight: 30g	Support Wireless Programming Via BLE Support Bluetooth HID Support IBeacons Support AT command to config the BLE Transparent communication through Serial BLE firmware updating **Supported:** **Android System 4.3+ with BLE4.0 module inside with original firmware with BLE driver**. e.g. Nexus 4+, Xaiomi 2s, Samsung Galaxy s4, Samsung Galaxy note 3 etc. **iOS 7.0+ devices**: iPhone 4s+, iPad 3+, iPad Mini, iPod 5th Gen (iPhone 4s is not 100% supported.) **Not compatible with:** Other brand BLE modules/devices since different firmware using in CC2540 e.g. FAQ > Q12 Bluetooth 2.0 or other types of bluetooth modules/devices

NOTE: If you want to use a **computer** to communicate with Bluno, a BLE link or USB BLE link is needed, the BLE on your computer is not compatible!

Bluno Basic Demo

This section is focused on the basic function of Bluno. You can easily develop your own Android application based on this Demo.

Bluno Basic Demo with library for Android



Bluno Basic Demo with library for IOS

Tools needed

- Bluno x1
- Android 4.3+ Devices Or IOS Devices x1
- Micro USB cable x1

Steps

1. If you stuck in any of the following process, Go to Trouble Shooting section for help.

2. Install the Arduino IDE.

3. Copy the source code below and paste it into Arduino IDE. (This Sketch only do one thing that replies the same message received from Serial port)

```
void setup() {
  Serial.begin(115200);  //initial the
Serial
}


void loop() {
  if (Serial.available())  {
    Serial.write(Serial.read());//send
what has been received
    Serial.println();   //print line f
eed character
  }
}
```

4. Select the "Arduino Uno" as the target board in Menu->Tools->Board and its serial port.

5. Upload the Sketch.

NOTE: If the Bluno was connected, i.e. the LINK led is ON, then it will fail to upload any sketch since the Serial port was occupied by the BLE. Please unlink the connection.

For Android

6. Install the APK file into your Android phone.

7. Run the application.

8. Click the "Scan" button（①） for scanning and select the device.

9. After connected, Click the "Send Data" button（②） to send the message in the text view(③) to Bluno.

10. The Bluno will reply with the same data. Thus you can see the same "received data" （④）.

11. Notice that the on-board RX and TX led will blink when sending and receiving the message.

For IOS

6. Compile the source code into your apple device.

7. Run the application.

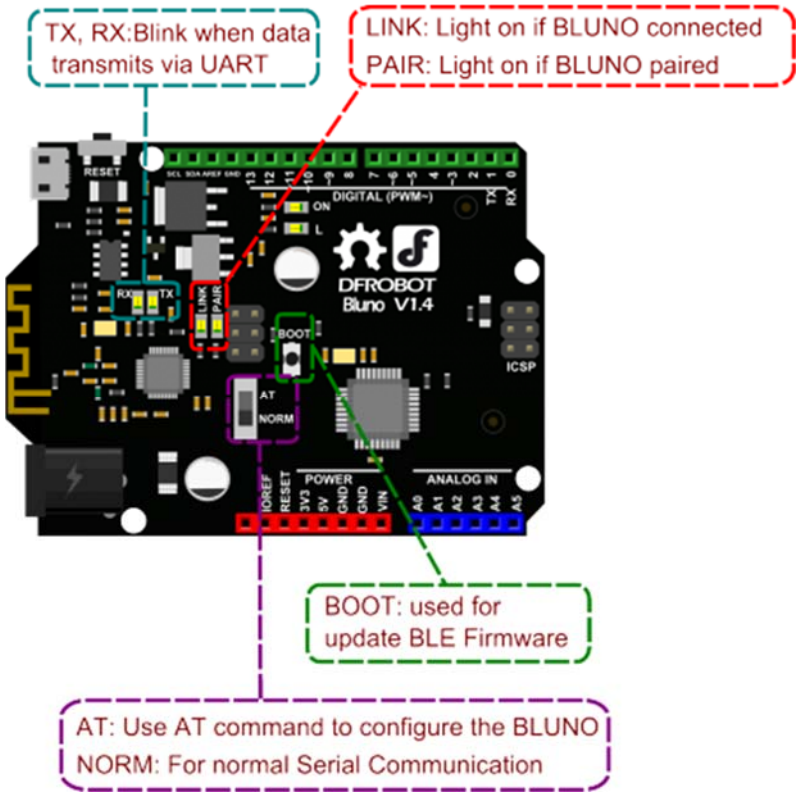
8. Click the "Search" button (②) for searching and select the device.

9. After the Connection state change from "Not Ready!" to "Ready!" (①), Click the "Send" button（④） to send the message in the text view(③) to Bluno.

10.The Bluno will reply with the same data. Thus you can see the same "received data" （⑤）.

11. Notice that the on-board RX and TX led will blink when sending and receiving the message.

Board Overview



Wireless Programming via BLE

In this section, we will learn how to Upload the sketch on air via BLE.

Tools required

- BLUNO/Bluno Nano/BLE link x2
- Micro USB cable x2

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.

2.Turn the switches to "AT"(Before V1.7)/Input "+++" (After 1.8) in the Serial port to enable the AT command mode.

3.Connect them with computer.

4.For the CENTRAL device, sending the following AT command :

Input : AT+SETTING=DEFCENTRAL<CR+LF>	Answer(Return):OK
Input : AT+BLUNODEBUG=OFF<CR+LF>	Answer(Return):OK

5.For the PERIPHERAL one, sending the following AT command :

Input : AT+SETTING=DEFPERIPHERAL<CR+LF>	Answer(Return):OK
Input : AT+BLUNODEBUG=OFF<CR+LF>	Answer(Return):OK

6. "AT+BLUNODEBUG=OFF" will make Wireless Programming more stable. However in this mode, you can not monitor the Serial port through USB on PC.

7. Turn the switches to "NORM"(Before V1.7)/ Input "AT+EXIT"(After V1.8) to exit AT communication mode.

8. Unplug the USB connection of the PERIPHERAL one, and use other external power supply like battery.

9. In seconds, the Link LED will be on, which means they have connected.

10. Click Upload, and sketch will be successfully uploaded to the PERIPHERAL device.

NOTE:

1. You have to choose the board of receiver in the setting "Tools > Board", i.e. if the sender is Bluno and the receiver board is **Bluno Mega2560**, then you should choose **Mega2560**.
2. The wireless programming can be done in bi-way between the CENTRAL and PERIPHERAL BLE devices, but the speed is different, i.e.

- CENTRAL -> PERIPHERAL 2KB/s
- PERIPHERAL -> CENTRAL 4KB/s

Update BLE Firmware on Bluno（AT+VERSION to check the version）

> **NOTE**: The setting will be **reserved** after the update. So if you want to apply Factory Settings, just use the AT command **AT+SETTING=DEFAULT**.

Version 1.7 or before

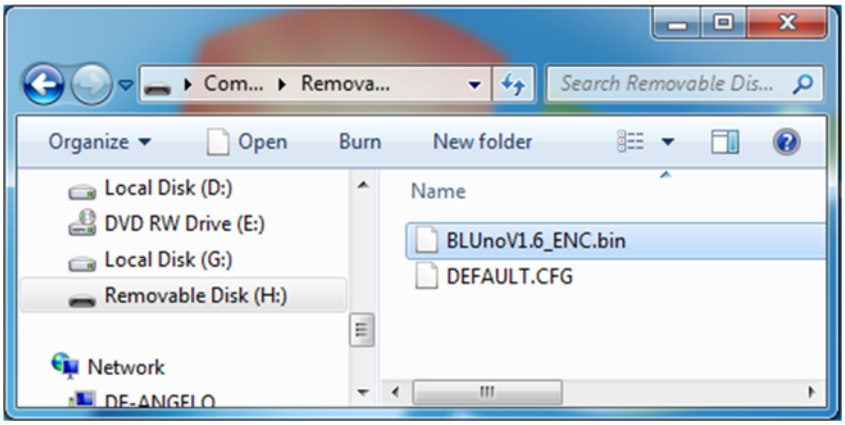
This step is only for the early versions of this board 1.6 and previous firmware. For version 1.7 or later. Skip to next section **Version 1.8 or later**.

> **NOTE**: This method is only compatible with Windows XP, and Windows 7. Windows 8 and 10 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.



Bluno Firmware update

4.Unpack UBL2SBL.bin (this link) patch into the Bluno flash drive.

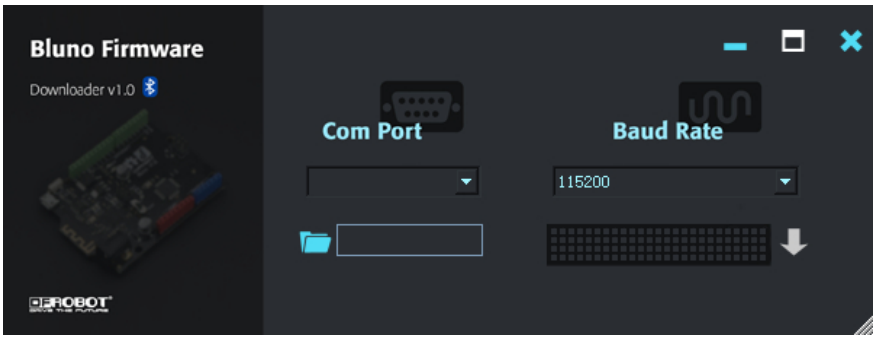
5.After it automatically reboots, and we can proceed with step 2.

Version 1.8 or later

On Version 1.7 (and later), firmware can be uploaded using the following software.

Click on the revelant link for your Operating system.

- DFRobot blunoFWDownloader
- DFRobot blunoFWDownloader for MAC



Bluno Firmware update

The software is very simple to use.

1. **Press and hold the boot button down**
2. Connect the USB
3. Two LED flash alternately
4. Download and decompression the Firmware in the Document section (This is the Link)
5. Load the firmware file (For Bluno or Bluno Nano, please use "SBL_BlunoV*.bin". For BLE-Link, please use "SBL_BLE-LinkV*.bin")
6. Click the upload button and wait 2 minutes.

Configure the BLE through AT command

Version 1.8 or later

The AT Mode Switch becomes useless at Firmware version V1.8 or later.

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 two pull-down menu

5. Type "+++"（③） like this and press send button（④）

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

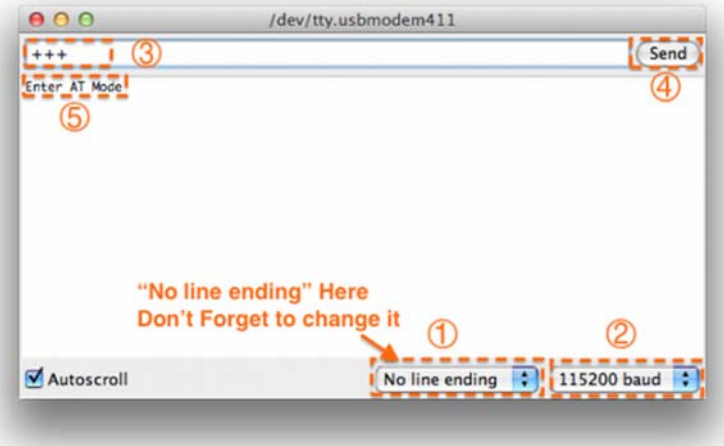


Fig1: +++ enter the AT CMD Mode

7. Select the "Both NL & CR"(①) and 115200 baud（②） in the two 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.

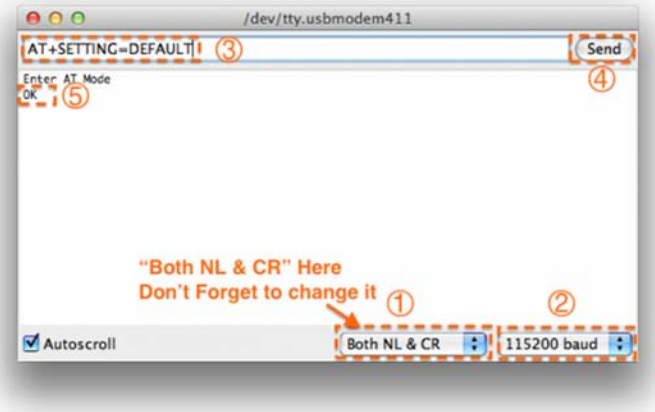


Fig1: enter the AT command, remember selectting the Both NL & CR

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**

Version 1.7 or before

1. Turn the on-board 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 Arduino Serial monitor, which is easy to use.

3. Select the correct serial port in Menu->Tool->Serial port

4. Open the Serial monitor (on the upper right of the IDE windows)

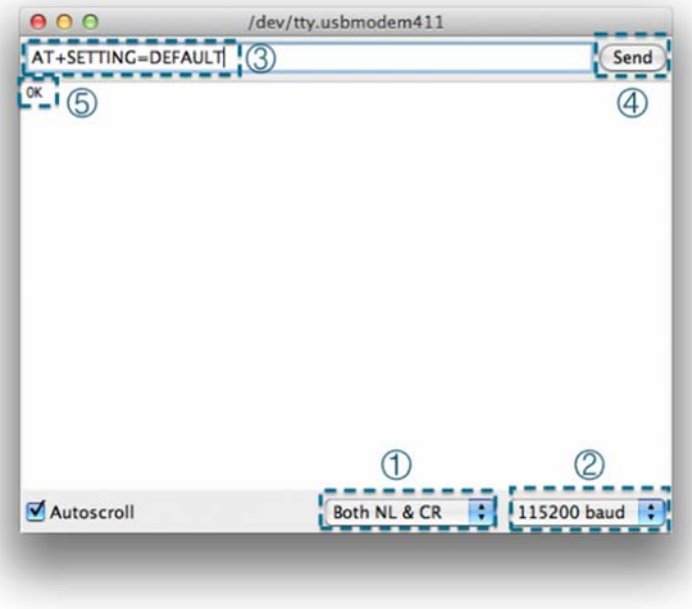


Fig1: Coolterm setting

5. Select the "Both NL & CR"(①) and 115200 baud（②） in the two pull-down menu

6. Type or copy the AT command in the dialog（③） like this and press send button（④）

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

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

9. **Turn the switch to "NORM", entering the normal mode**

AT Command List

NOTE: **<CR+LF>** means Carriage-Return and Line-Feed, which is the same meaning with "Both NL & CR" in Arduino Serial monitor, do not include the string "<CR+LF>" itself.

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)

1*. "AT+KEY" to simulate pressing some buttons in HID mode, read FAQ 16 below for **How to use HID mode**.

AT+KEY=keyValue0<CR+LF>	To simulate one button was pressed
AT+KEY=keyValue0+keyValue1<CR+LF>	To simulate two buttons was pressed
AT+KEY=keyValue0+keyValue1+keyValue2<CR+LF>	To simulate three buttons was pressed

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: 115200,8,N,1)

6. "AT+BIND" bind another BLE chip. BLE can only connect to the BLE chip with this MAC address

AT+BIND=0x0017ea9397e1<CR+LF>	Set the BLE binding (destination) MAC address to 0x0017ea9397e1
AT+BIND=?<CR+LF>	Request the binding (destination) MAC address (default: 0x8A6D3B8A6D3B)

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(destination) MAC 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

11. "AT+SETTING" change the default setting (new in BLE firmware 1.6)

AT+SETTING=DEFAULT<CR+LF>	Restore the default settings, same as PERIPHERAL mode
AT+SETTING=DEFPERIPHERAL<CR+LF>	Restore the default settings for PERIPHERAL mode
AT+SETTING=DEFCENTRAL<CR+LF>	Restore the default settings for CENTRAL mode
AT+SETTING=?<CR+LF>	Request the setting mode (default: DEFPERIPHERAL). If the settings are changed by AT command, "UNKNOWN" will be replied.

12. "AT+BLUNODEBUG" When Bluetooth is connected and BLE chip(CC2540) received the UART message from MCU(ATMEGA328), send the UART message not only to the Bluetooth, but also to the USB port. So that when Bluetooth is connected, we can use the serial monitor to get the UART message. (new in BLE firmware 1.6)

AT+BLUNODEBUG=ON<CR+LF>	Turn on the BLUNO DEBUG so that when Bluetooth is connected, we can use the serial monitor to get the UART message.

AT+BLUNODEBUG=OFF<CR+LF>	Turn off the BLUNO DEBUG so that wireless programming will be more stable.
AT+BLUNODEBUG=?<CR+LF>	Request the BLUNO DEBUG state (default: ON)

13. "AT+USBDEBUG" When Bluetooth is connected and BLE chip(CC2540) received the Bluetooth message from IOS or Android device, send the data not only to the UART, but also to the USB port. So that when Bluetooth is connected, we can use the serial monitor to directly get the Bluetooth message. (new in BLE firmware 1.6)

AT+USBDEBUG=ON<CR+LF>	Turn on the BLUNO DEBUG So that when Bluetooth is connected, we can use the serial monitor to directly get the Bluetooth message from IOS or Android device.
AT+USBDEBUG=OFF<CR+LF>	Turn off the USB DEBUG so that wireless programming will be more stable.
AT+USBDEBUG=?<CR+LF>	Request the USB DEBUG state (default: OFF)

14. "AT+TXPOWER" Change the Transmitted Power which will change the signal range. (new in BLE firmware 1.6)

AT+TXPOWER=0<CR+LF>	Change the Transmitted Power to fit the iBeacon calibration. (4, 0, -6 -23 is acceptable)
AT+TXPOWER=?<CR+LF>	Request the Transmitted Power (default: 0)

15. "AT+IBEACONS" Enable the iBeacons feature(new in BLE firmware 1.6)

AT+IBEACONS=ON<CR+LF>	Enable the iBeacons feature.

AT+IBEACONS=OFF<CR+LF>	Disable the iBeacons feature.
AT+IBEACONS=?<CR+LF>	Request whether the iBeacons feature is enabled. (default: ON)

16. "AT+VERSION" the version of the firmware(new in BLE firmware 1.6)

AT+VERSION=?<CR+LF>	Request the version of the firmware.

17. "AT+RSSI" Request the RSSI of the BLE (new in BLE firmware 1.6)

AT+RSSI=?<CR+LF>	Request the RSSI of the BLE(if there is no connection, "-000" will be returned)

18. "AT+MAJOR" Set the major number of the iBeacons (new in BLE firmware 1.6)

AT+MAJOR=0<CR+LF>	Set the major number of the iBeacons to "0". (0 to 65535 is acceptable)
AT+MAJOR=?<CR+LF>	Request the major number of the iBeacons.(default "0")

19. "AT+MINOR" Set the minor number of the iBeacons (new in BLE firmware 1.6)

AT+MINOR=0<CR+LF>	Set the minor number of the iBeacons to "0". (0 to 65535 is acceptable)

AT+MINOR=?<CR+LF>	Request the minor number of the iBeacons.(default "0")

20. "AT+EXIT" Exit the AT Command Mode (new in BLE firmware 1.8),

AT+EXIT<CR+LF>	Exit the AT Command Mode.