Bluetooth Bee

From Wiki

Contents

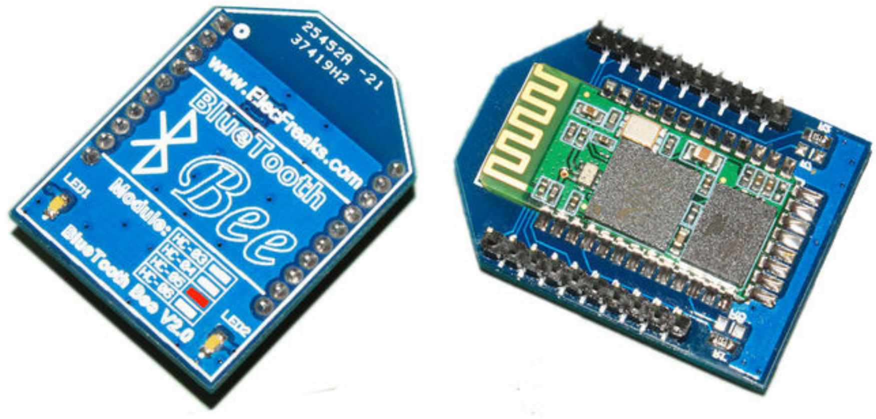
Introduction

Bluetooth Bee is a simple breakout board for the Bluetooth Module, which designed to DIP moudel compatible with existing **Xbee sockets**, designed for transparent wireless serial connection setup. Bluetooth Bee can be configured to Master, Slave two different modes(HC-06 just be Slave, coulud be customized), and it will connect to or be connected by other devices that support SPP protocol per configuration, and it is a suitable substitute for most applications. **Bluetooth Bee** is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup. This is a fully qualified Bluetooth **V2.0+EDR** (Enhanced Data Rate) 3Mbps modulation with complete 2.4GHz radio transceiver. It uses CSR **Bluecore4-external** single chip Bluetooth system with CMOS technology and with **AFH** (Adaptive Frequency Hopping).

The different of HC-03/HC-04/HC-05/HC-06:

- HC-03/HC-04 is Industrial Standard, HC-05/HC-06 is Commercial Standard.
- HC-03/HC-05 could be configured to Master/Slave by user.
- HC-04/HC-06 just be Master or Slave when the factory, cann't be configured by user.

Model: Bluetooth Bee - Base on HC-05 (http://www.elecfreaks.com/store/bluetooth-bee-masterslave-p-171.html)



Feature

Hardware

- Chipset CSR BC417143 (BlueCore4External)
- Bluetooth version V2.0+EDR
- Class 2 module with PCB antenna
- Flash 8Mbit
- Interface I2C UART PCM USB
- Support for 802.11 co-existence
- 3.3V operation
- SPP firmware
- Support for up to seven slaves
- ISM Band:2.40GHz~2.48GHz
- Master/Slave/Loopback working roles
- XBee compatible socket
- Rohs: Yes

Software

- Supported baud rate: 9600,19200,38400,57600,115200,230400,460800.
- Use CTS and RTS to control the data stream.
- Auto-reconnect in 30 min when disconnected as a result of beyond the range of connection.
- Auto-connect the last device on power as default.
- Permit matched device connect by default.
- Default Baud rate: **38400**, Data bits:8, Stop bit:1,Parity:No parity, Data control: has.
- Default PINCODE:”1234”.

Application Ideas

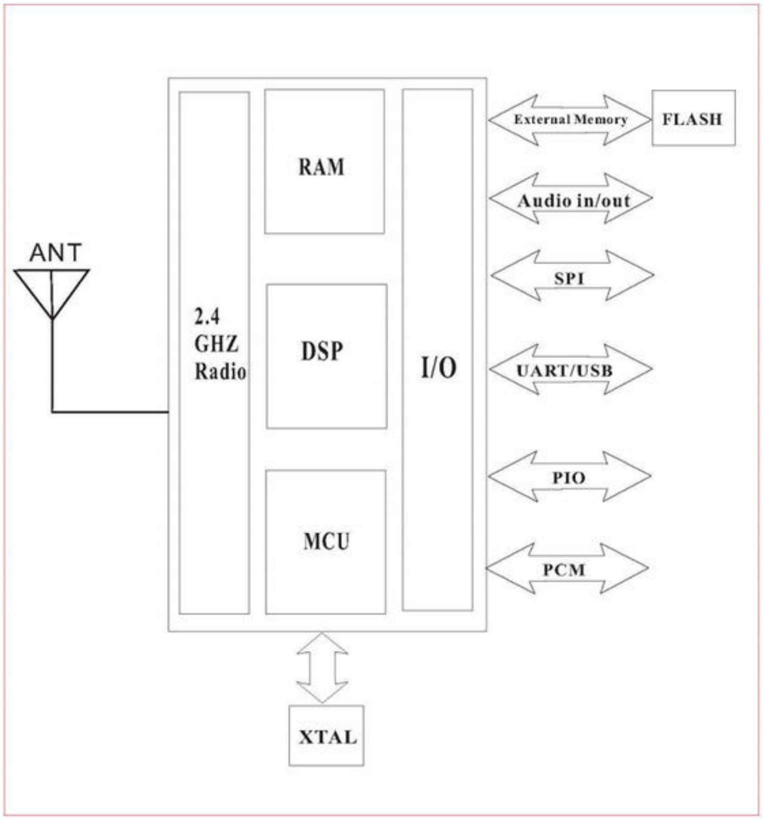
- Bluetooth Car Handsfree Device
- Bluetooth GPS
- Bluetooth PCMCIA , USB Dongle
- Bluetooth Data Transfer
- Bluetooth Android chat with Arduino (http://www.elecfreaks.com/829.html)

Cautions

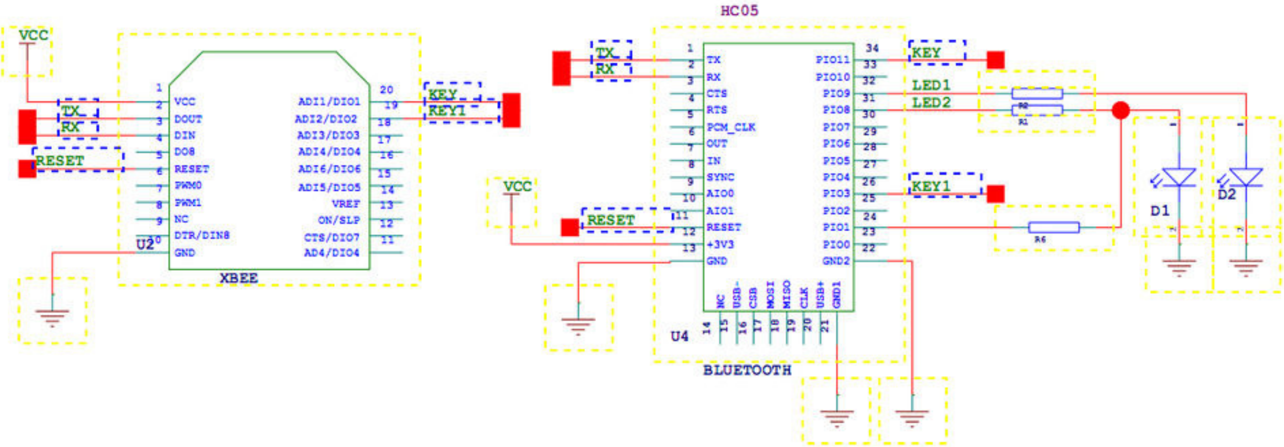
- **The operation voltage of Bluetooth Bee is 3.3V,do not exceed this range, otherwise it will be destroyed.**
- **Command to change baud rate is persistent even after reset.**
- **Bluetooth Bee need the other XBee shield (http://www.elecfreaks.com/store/xbee-shield-with-logic-level-converter-p-175.html) to connect with Arduino.**

Schematic

Block Diagram



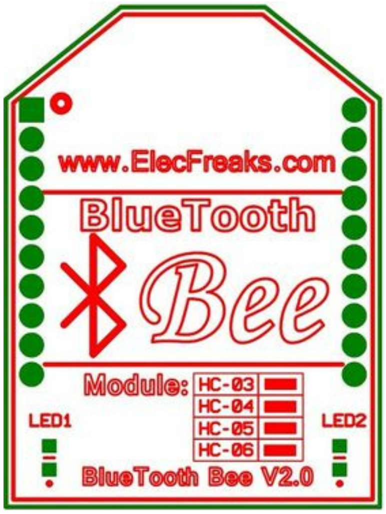
Schematic



Specification

Pin definition and Rating

Top-view:



Pin	#	Pad Type	Description
VCC	1	3.3V	Integrated 3.3V(+)supply with on-chip linear regulator output within 3.15-3.3V
DOUT	2	CMOS output,tri-stable with weak internal pull-up	UART Data output
DIN	3	CMOS input with weak internal pull-up	UART Data INput
RESET	5	CMOS input with weak internal pull-up	Reset if low,input must be low for >5ms to cause a reset
GND	10	VSS	Ground port
NG	11	Bi-Direction	No connection
NG	12	Bi-Direction	No connection
NG	4	Bi-Direction	No connection
NG	6	Bi-Direction	No connection
NG	7	Bi-Direction	No connection
NG	8	Bi-Direction	No connection
NG	9	Bi-Direction	No connection
NG	13	Bi-Direction	No connection
NG	14	Bi-Direction	No connection
NG	15	Bi-Direction	No connection
NG	16	Bi-Direction	No connection
NG	17	Bi-Direction	No connection
NG	18	Bi-Direction	No connection
KEY	19	CMOS input with weak internal pull-up	Programmable input/output line
KEY1	20	CMOS input with weak internal pull-up	Programmable input/output line

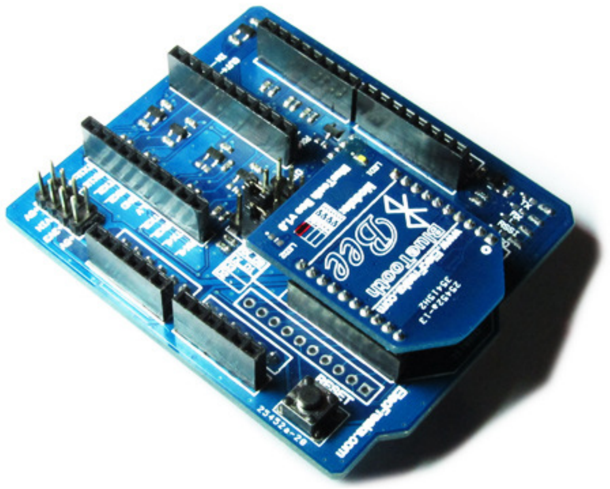
Mechanic Dimensions

PCB size : 33.7mm * 25.2mm * 1.6mm

Usage

Use a XBee Shield to connect Bluetooth Bee to Arduino/AVR.

Hardware Installation



Commands to change default configuration

Bluetooth Bee serial communication module (can be short for module) has two work modes: order-response work mode and automatic connection work mode. And there are three work roles (Master, Slave and Loopback) at the automatic connection work mode. When the module is at the automatic connection work mode, it will follow the default way set lastly to transmit the data automatically. When the module is at the order-response work mode, user can send the AT command to the module to set the control parameters and sent control order. The work mode of odule can be switched by controlling the module PIN (KEY) input level.

1. Set working MODE

\r\n+STWMOD=0\r\n	Set device working mode as client (slave). Save and Rest.
\r\n+STWMOD=1\r\n	Set device working mode as server (master). Save and Rest.

Note: \r\n is necessary for operation and the value of are **0x0D 0x0A** in Hex. **\r** and **\n** represent **carriage-return** and **line-feed**(or next line),

2.Set BAUDRATE

\r\n+STBD=115200\r\n	Set baudrate 115200. Save and Rest.

Supported baudrate: 9600, 19200,38400,57600,115200,230400,460800.

3. Set Device NAME

\r\n+STNA=abcdefg\r\n	Set device name as “abcdefg”. Save and Rest.

4. Auto-connect the last paired device on power

\r\n+STAUTO=0\r\n	Auto-connect forbidden. Save and Rest.
\r\n+STAUTO=1\r\n	Permit Auto-connect. Save and Rest.

5. Permit Paired device to connect me

\r\n+STOAUT=0\r\n	Forbidden. Save and Rest.
\r\n+STOAUT=1\r\n	Permit. Save and Rest.

6. Set PINCODE

\r\n +STPIN=2222\r\n	Set pincode “2222”, Save and Rest.

7. Delete PINCODE(input PINCODE by MCU)

\r\n+DLPIN\r\n	Delete pincode. Save and Rest.

8. Read local ADDRESS CODE

\r\n+RTADDR\r\n	Return address of the device.

9. Auto-reconnecting when master device is beyond the valid range (slave device will auto-reconnect in 30 min when it is beyond the valid range)

\r\n+LOSSRECONN=0\r\n	Forbid auto-reconnecting.
\r\n+LOSSRECONN=1\r\n	Permit auto-reconnecting.

Commands for Normal Operation:

1. Inquire

a) Master	
\r\n+INQ=0\r\n	Stop Inquiring
\r\n+INQ=1\r\n	Begin/Restart Inquiring
b) Slave	
\r\n+INQ=0\r\n	Disable been inquired
\r\n+INQ=1\r\n	Enable been inquired

When **+INQ=1** command is successful, the red and green LEDS blink alternatively.

2. Bluetooth module returns inquiring result

\r\n+RTINQ=aa,bb,cc,dd,ee,ff; name\r\n	Serial Bluetooth device with the address “aa,bb,cc,dd,ee,ff” and the name “name” is inquired

3. Connect device

\r\n+CONN=aa,bb,cc,dd,ee,ff\r\n	Connect to a device with address of "aa,bb,cc,dd,ee,ff”

4. Bluetooth module requests inputting PINCODE

\r\n+INPIN\r\n

5. Input PINCODE

\r\n+RTPIN=code\r\n
Example: RTPIN=0000 Input PINCODE which is four zero

6. Disconnect device Pulling PIO0 high will disconnect current working Bluetooth device.

7. Return status \r\n+BTSTA:**xx**\r\n
xx status:

- 0 - Initializing
- 1 - Ready
- 2 - Inquiring
- 3 - Connecting
- 4 - Connected

(**Note:** This is not a command, but the information returned from the module after every command)

Serial module PINs (HC-05)

1. PIO8 connects with LED. When the module is power on, LED will flicker. And the flicker style will indicate which work mode is in using since different mode has different flicker time interval.
2. PIO9 connects with LED. It indicates whether the connection is built or not. When the Bluetooth serial is paired, the LED will be turned on. It means the connection is built successfully.
3. KEY is the work mode switch. When this PIN port is input high level, the work mode will become order-response work mode. While this PIN port is input low level or suspended in air, the work mode will become automatic connection work mode.
4. The module can be reset if it is re-powered since there is a reset circuit at the module.

- **How to get to the AT mode:**

Way 1:
Step 1: Input low level to KEY.
Step 2: Supply power to the module.
Step 3: Input high level to the KEY. Then the module will enter to AT mode. The baud rate is as same as the communication time, such as 9600 etc.
Way 2:
Step 1: Connect KEY to the power supply PIN.
Step 2: Supply power to module (the KEY is also supplied with high level since the KEY is connected with power supply PIN). Then the module will enter to AT module. But at this time, the baud rate is 38400. In this way, user should change the baud rate at the AT mode, if they forget the communication baud rate.

- **How to get to the communication mode:**

Step 1: Input low level to KEY.
Step 2: Supply power to the module. Then the module will enter to communication mode. It can be used for pairing.

- **How to set this module be the master role:**

Step 1: Input high level to KEY.
Step 2: Supply power to the module. And the module will enter to the order-response work mode. Step 3: Set the parameters of the super terminal or the other serial tools (baud rate: 38400, data bit:8, stop bit:1, no parity bit, no Flow Control)
Step 4: Sent the characters "AT+ROLE=1\r\n" through serial, then receive the characters "OK\r\n". Here, "\r\n" is the CRLF.
Step 5: Input low level to PIO, and supply power to the module again. Then this module will become master role and search the other module (slave role) automatically to build the connection.

Notes

(1) Bluetooth Bee base on HC-03 and HC-05's command should end up with "\r\n". It means when you finish programming, you should add terminator ("ENTER" or "0x0d 0x0a") to the program. It's different from HC-04 and HC-06 (They don't need terminator).
(2) The most common commands for HC-03 and HC-05 are: AT+ROLE (set master –slave), AT+CMODE(set address pairing), AT+PSWD (set password). If you want the master module has the function of remembering slave module, the most simply way is: First, set AT+CMODE=1. Make the master module pair with the slave module. Second, set AT+CMODE=0. Then the master module just can make pair with that specified slave module.
(3) When KEY keeps high level, all commands can be used. Otherwise, only some of them can be used.

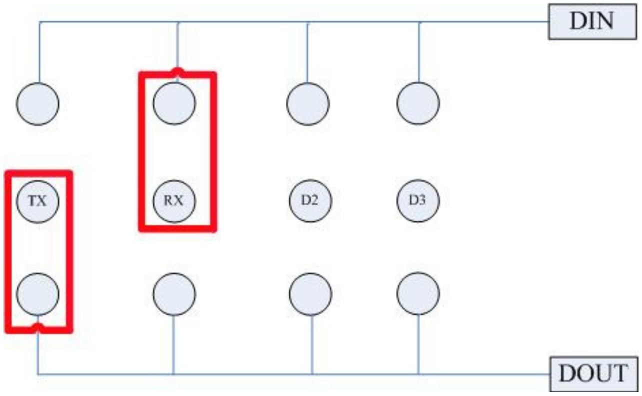
Software Instruction

The XBee shield (http://www.elecfreaks.com/store/xbee-shield-with-logic-level-converter-p-175.html) has 4 row jumpers. DIN corresponds with Bluetooth Bee's RX(DIN), DOUT corresponds with Bluetooth Bee's TX(DOUT), TX_H corresponds with Arduino's TX(D0), RX_H corresponds with Arduino's RX(D1), D2 corresponds with Arduino's D2, D3 corresponds with Arduino's D3. We select work mode by DIN and DOUT freedom match with TX_H, TX_H, D2, D3.

- Method One：

Bluetooth Bee direct use Arduino main board's UART(TX/RX) connect to PC. However this method must remove the MCU chip, and the Bluetooth Bee inserted on Arduino's main board with XBee shield. This means the Bluetooth Bee direct connect to PC Serial port. Setting the XBee shield's jumper connect as below:

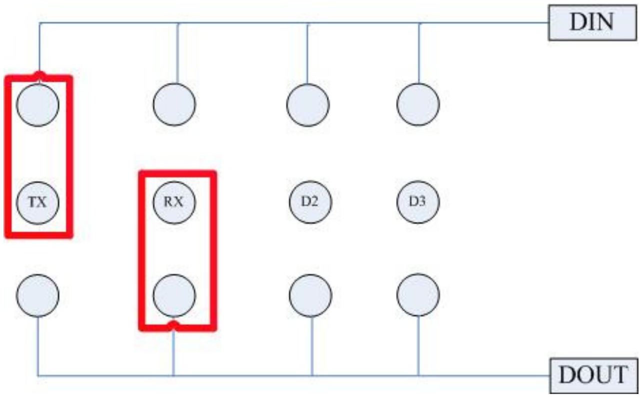
DIN to RX_H , DOUT to TX_H.



- Method Two：

The RX and TX pins of the microcontroller are still connected to the TX and RX pins (respectively) of the FTDI chip – data sent from the microcontroller will be transmitted to the computer via USB as well as being sent wirelessly by the Xbee module. The microcontroller, however, will only be able to receive data from the Xbee module, not over USB from the computer.
So this way just for you has perfect code and don't need monitor from serial port, because you cann't see anything from serial port tool. The second way, setting the XBee shield's jumper connect as below:

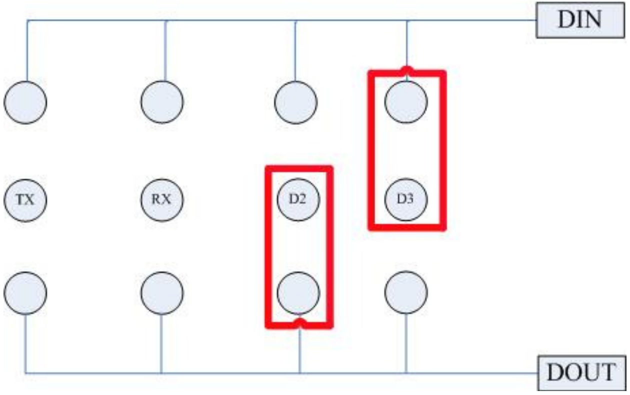
DIN to TX_H , DOUT to RX_H.



- Method Three：

Use Software Serial library to define D2 and D3 as a virtual serial port TX and RX. By this way, you can use Arduino’s Serial Monitor to Monitor operation of the code.

Setting the XBee shield’s jumper connect as below: **DIN to D3, DOUT to D2.**



- Please reference Communication between Android and Arduino with Bluetooth Bee (2) (http://www.elecfreaks.com/829.html)

Programming

Includes important code snippet. Demo code like :

```
#include <NewSoftSerial.h>   //Software Serial Port
#define RxD 2
#define TxD 3

NewSoftSerial blueToothSerial(RxD,TxD);

void setup()
{
    Serial.begin(9600);          //Serial port for debugging, Comment this line if not required
    pinMode(RxD, INPUT);
    pinMode(TxD, OUTPUT);
    setupBlueToothConnection();

}

void loop()
{

  if(blueToothSerial.read() == 'a')
  {
    blueToothSerial.println("You are connected to Bluetooth Bee");
    //You can write you BT communication logic here
  }
}

void setupBlueToothConnection()
{
    Serial.print("Setting up Bluetooth link");       //For debugging, Comment this line if not required
    blueToothSerial.begin(38400); //Set BluetoothBee BaudRate to default baud rate 38400
    delay(1000);
    sendBlueToothCommand("\r\n+STWMOD=0\r\n");
    sendBlueToothCommand("\r\n+STNA=modem\r\n");
    sendBlueToothCommand("\r\n+STAUTO=0\r\n");
    sendBlueToothCommand("\r\n+STOAUT=1\r\n");
    sendBlueToothCommand("\r\n+STPIN=0000\r\n");
    delay(2000); // This delay is required.
    blueToothSerial.print("\r\n+INQ=1\r\n");
    delay(2000); // This delay is required.
    Serial.print("Setup complete");

}

void sendBlueToothCommand(char command[])
{
    char a;
    blueToothSerial.print(command);
    Serial.print(command);                          //For debugging, Comment this line if not required
    delay(3000);

    while(blueToothSerial.available())              //For debugging, Comment this line if not required
    {                                               //For debugging, Comment this line if not required
       Serial.print(char(blueToothSerial.read()));  //For debugging, Comment this line if not required
    }                                               //For debugging, Comment this line if not required
}
```

AT command set(HC-06)

The way to the AT command mode: supply power to the module, it will enter to the AT mode if it needn’t pair. The interval of command is about 1 second. Default parameter: Baud rate:9600N81, ID: linvor, Password:1234

- **Test communication**

Send: AT (please send it every second)
Back: OK

- **Reset the Bluetooth serial baud rate**

Send: AT+BAUD1
Back: OK1200
Send: AT+BAUD2
Back: OK2400
……
1---------1200
2---------2400
3---------4800
4---------9600 (Default)
5---------19200
6---------38400
7---------57600
8---------115200
9---------230400
A---------460800
B---------921600
C---------1382400
PC can’t support the baud rate lager than 115200. The solution is: make the MCU have higher baud rate (lager than 115200) through programming, and reset the baud rate to low level through the AT command. The baud rate reset by the AT command can be kept for the next time even though the power is cut off.

- **Reset the Bluetooth name**

Send: AT+NAMEnam
Back: OKname
Parameter name: Name needed to be set (20 characters limited)
Example:
Send: AT+NAMEbill_gates
Back: OKname
Now, the Bluetooth name is reset to be “bill_gates” The parameter can be kept even though the power is cut off. User can see the new Bluetooth name in PDA refresh service. (Note: The name is limited in 20 characters.)

- **change the Bluetooth pair password**

Send: AT+PINxxxx
Back:OKsetpin
Parameter xxxx: The pair password needed to be set, is a 4-bits number. This command can be used in the master and slave module. At some occasions, the master module may be asked to enter the password when the master module tries to connect the slave module (adapter or cell-phone). Only if the password is entered, the successful connection can be built. At the other occasions, the pair can be finish automatically if the master module can search the proper slave module and the password is correct. Besides the paired slave module, the master can connect the other devices who have slave module, such as Bluetooth digital camera, Bluetooth GPS, Bluetooth serial printer etc.
Example:
Send: AT+PIN8888
Back: OKsetpin

Then the password is changed to be 8888, while the default is 1234. This parameter can be kept even though the power is cut off.

- **No parity check (The version, higher than V1.5, can use this command)**

Send: AT+PN (This is the default value)
Back: OK NONE

- **Set odd parity check (The version, higher than V1.5, can use this command)**

Send: AT+PO
Back: OK ODD

- **Set even parity check(The version, higher than V1.5, can use this command)**

Send: AT+PE
Back: OK EVEN

- **Get the AT version**

Send: AT+VERSIO
Back: LinvorV1.n

AT command set(HC-05)

Please reference Bluetooth Bee AT Commamd (http://elecfreaks.com/store/download/datasheet/Bluetooth/HC-0305%20serail%20module%20AT%20commamd%20set%20201104%20revised.pdf)

AT command error

Error_code (hex)	Explanation
0	AT command error
1	The result is default value
2	PSKEY write error
3	Device name is too long (mbytes)
4	Device name is 0 byte
5	Bluetooth address: NAP is too
6	Bluetooth address: UAP is too
7	Bluetooth address: LAP is too
8	PIO port mask length is 0
9	Invalid PIO port
A	Device class is 0 byte
B	Device class is too long
C	Inquire Access Code length is 0
D	Inquire Access Code is too long
E	Invalid Inquire Access Code
F	Pairing password is 0
10	Pairing password is too long (more than 16 bytes)
11	Role of module is invalid
12	Baud rate is invalid
13	Stop bit is invalid
14	Parity bit is invalid
15	No device in the pairing list
16	SPP is not initialized
17	SPP is repeatedly initialized
18	Invalid inquiry mode
19	Inquiry timeout
1A	Address is 0
1B	Invalid security mode
1C	Invalid encryption mode

Bill of Materials (BOM) /parts list

All the components used to produce the product.

FAQ

Support

If you have questions or other better design ideas, you can go to our wish (http://www.elecfreaks.com/store/index.php?main_page=login) to discuss.

Version Tracker

Revision	Descriptions	Release
v1.0b	Initial public release	date

Bug Tracker

Bug Tracker is the place you can publish any bugs you think you might have found during use. Please write down what you have to say, your answers will help us improve our

products.

Additional Idea

- Implement a wireless PS/2 Keyboard / Mouse.
- Wireless control a toy robot from PC.

Resources

- Video Tutorial (http://youtu.be/dPNtmoYnFpw)
- Bluetooth Bee User Guide (http://www.elecfreaks.com/store/download/product//Bluetooth/Bluetooth_Bee_User_Guide/Bluetooth_Bee_(HC-05_and_HC-06)User_Guide.pdf)
- Datasheet (http://elecfreaks.com/store/download/datasheet/Bluetooth/HC-06-Spec.pdf)
- Schematic (http://elecfreaks.com/store/download/datasheet/Bluetooth/Schematic.pdf)
- Bluetooth Bee AT Command (http://elecfreaks.com/store/download/datasheet/Bluetooth/HC-0305%20serail%20module%20AT%20commamd%20set%20201104%20revised.pdf)
- User Guide (http://elecfreaks.com/store/download/datasheet/Bluetooth/Manual(new).pdf)
- Serial Tool (http://elecfreaks.com/store/download/datasheet/Bluetooth/Serial_Port_Test.zip)
- HyperTerminal Tool (http://elecfreaks.com/store/download/datasheet/Bluetooth/HyperTerminal.zip)
- Android chat with Arduino by Bluetooth Bee（PART2）(http://www.elecfreaks.com/829.html)
- Android chat with Arduino by Bluetooth Bee（PART1）(http://www.elecfreaks.com/677.html)
- Android Bluetooth APIs Reference (http://developer.android.com/guide/topics/wireless/bluetooth.html)

How to buy

Click here to buy: http://www.elecfreaks.com/store/wireless-bluetooth-c-3_50.html

See Also

Other related products and resources.

Licensing

The Limited Inquiry Access Code(LIAC)is only intended to be used for limited time periods in scenarios where both sides have been explicitly caused to enter this state, usually by user action. For further explanation of the use of the LIAC, please refer to the Generic Access Profile. In contrast it is allowed to be continuously scanning for the General Inquiry Access Code (GIAC)and respond whenever inquired.

External Links

Links to external webpages which provide more application ideas, documents/datasheet or software libraries

Retrieved from "http://www.elecfreaks.com/wiki/index.php?title=Bluetooth_Bee"
Categories: Bluetooth Bee | Wireless | All Porducts

- This page was last modified on 8 August 2013, at 03:56.
- This page has been accessed 42,206 times.

- About Wiki
- Disclaimers