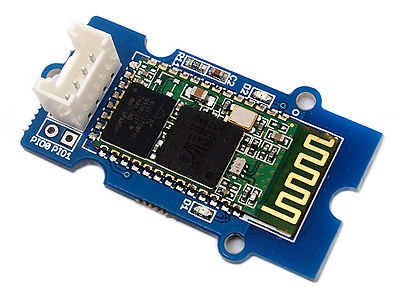
Grove - Serial Bluetooth

|  |
| --- |
| **Contents**   [[hide](javascript:toggleToc())]   * [1 Introduction](http://www.seeedstudio.com/wiki/Grove_-_Serial_Bluetooth#Introduction) * [2 Features](http://www.seeedstudio.com/wiki/Grove_-_Serial_Bluetooth#Features)   + [2.1 Hardware Features](http://www.seeedstudio.com/wiki/Grove_-_Serial_Bluetooth#Hardware_Features)   + [2.2 Software Features](http://www.seeedstudio.com/wiki/Grove_-_Serial_Bluetooth#Software_Features) * [3 Cautions](http://www.seeedstudio.com/wiki/Grove_-_Serial_Bluetooth#Cautions) * [4 Interface Function](http://www.seeedstudio.com/wiki/Grove_-_Serial_Bluetooth#Interface_Function) * [5 Software Instruction](http://www.seeedstudio.com/wiki/Grove_-_Serial_Bluetooth#Software_Instruction)   + [5.1 Working Sketch Map](http://www.seeedstudio.com/wiki/Grove_-_Serial_Bluetooth#Working_Sketch_Map)   + [5.2 Flowchat](http://www.seeedstudio.com/wiki/Grove_-_Serial_Bluetooth#Flowchat)   + [5.3 Commands to change default configuration](http://www.seeedstudio.com/wiki/Grove_-_Serial_Bluetooth#Commands_to_change_default_configuration)   + [5.4 Commands for Normal Operation:](http://www.seeedstudio.com/wiki/Grove_-_Serial_Bluetooth#Commands_for_Normal_Operation:) * [6 Usage](http://www.seeedstudio.com/wiki/Grove_-_Serial_Bluetooth#Usage)   + [6.1 Hardware Installation](http://www.seeedstudio.com/wiki/Grove_-_Serial_Bluetooth#Hardware_Installation)   + [6.2 Software Programming](http://www.seeedstudio.com/wiki/Grove_-_Serial_Bluetooth#Software_Programming) * [7 FAQ](http://www.seeedstudio.com/wiki/Grove_-_Serial_Bluetooth#FAQ) * [8 Additional Idea](http://www.seeedstudio.com/wiki/Grove_-_Serial_Bluetooth#Additional_Idea) * [9 Bill of Materials (BOM) /parts list](http://www.seeedstudio.com/wiki/Grove_-_Serial_Bluetooth#Bill_of_Materials_.28BOM.29_.2Fparts_list) * [10 Resources](http://www.seeedstudio.com/wiki/Grove_-_Serial_Bluetooth#Resources) * [11 How to buy](http://www.seeedstudio.com/wiki/Grove_-_Serial_Bluetooth#How_to_buy) * [12 See Also](http://www.seeedstudio.com/wiki/Grove_-_Serial_Bluetooth#See_Also) * [13 Licensing](http://www.seeedstudio.com/wiki/Grove_-_Serial_Bluetooth#Licensing) * [14 External Links](http://www.seeedstudio.com/wiki/Grove_-_Serial_Bluetooth#External_Links) |

Introduction

**Grove - Serial Bluetooth**is an easy to use module compatible with the existing Grove Base Shield, and designed for transparent wireless serial connection setup. The serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR(Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses **CSR Bluecore 04**-External single chip Bluetooth system with CMOS technology and with AFH(Adaptive Frequency Hopping Feature).It has the smallest footprint of **12.7mm x 27mm**. Hope it will simplify your overall design/development cycle.

**Model:**[**WLS31746P**](http://www.seeedstudio.com/depot/grove-serial-bluetooth-p-795.html?cPath=139_142)

[](http://www.seeedstudio.com/wiki/File:Twigbt00.jpg)

Features

**Hardware Features**

* Typical -80dBm sensitivity.
* Up to +4dBm RF transmit power.
* Fully Qualified Bluetooth V2.0+EDR 3Mbps Modulation.
* Low Power Operation.
* PIO control.
* UART interface with programmable baud rate.
* Integrated PCB antenna.

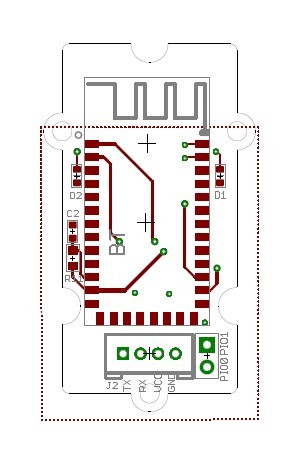
**Software Features**

* Default Baud rate: **38400**, Data bits:8, Stop bit:1,Parity:No parity.
* Supported baud rate: 9600,19200,38400,57600,115200,230400,460800.
* When a rising pulse is detected in PIO0, device will be disconnected.
* Status instruction port PIO1: low-disconnected, high-connected;
* PIO10 is connected to red led, PIO11 is connected to green led. When it fails or disconnects green led blinks 2 times/s .In pairing mode , red and green led blinks in interval , and green led blinks 1 time/2s after connections.
* Auto-connect the last device on power as default.
* Permit matched device connect by default.
* Default PINCODE:”0000”.
* Auto-reconnect in 30 min when disconnected as a result of beyond the range of connection.

Cautions

* While using with Seeeduino / Arduino, set the operation voltage to 5V. Else use a proper logic level converter.
* While using with UartSBee, set the operation voltage to 5V
* Command to change baud rate is persistent even after reset. Hence remember the baud rate for next use.

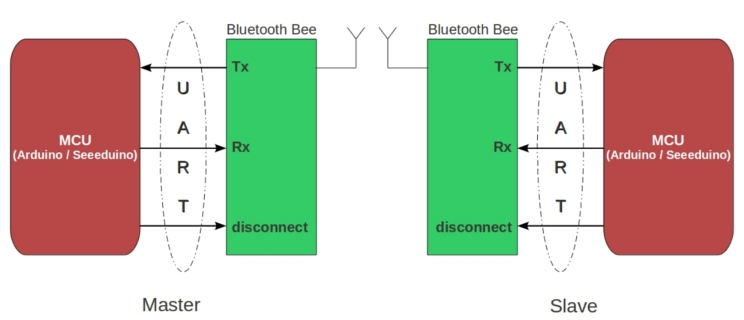
Interface Function

[](http://www.seeedstudio.com/wiki/File:SBT.jpg)

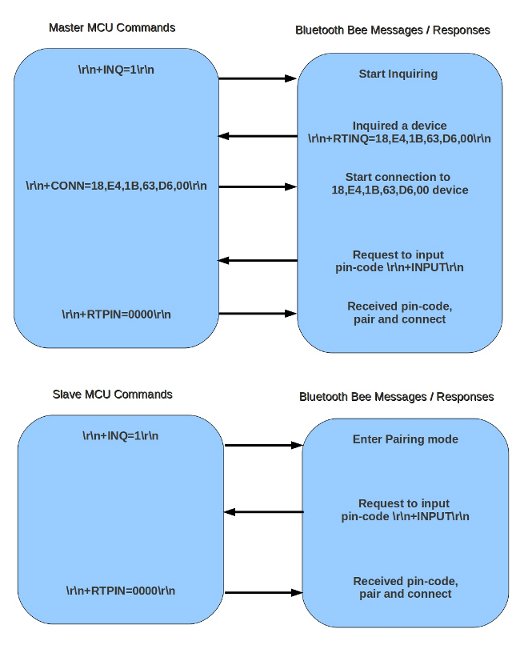
|  |  |
| --- | --- |
| **Pad Type** | **Description** |
| GND | Ground port |
| PIO1 | Status instruction port PIO1: low-disconnected, high-connected |
| PIO0 | When a rising pulse is detected in PIO0, device will be disconnected |
| RX | UART Data input |
| TX | UART Data output |
| VCC | Designed for 5V(+)supply using IC MIC5205\_3.3 for Bluetooth module power supply |

Software Instruction

**Working Sketch Map**

[](http://www.seeedstudio.com/wiki/File:Bluetooth-1.jpg)

**Flowchat**

[](http://www.seeedstudio.com/wiki/File:Bluetooth-2.jpg)

**Commands to change default configuration**

**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.

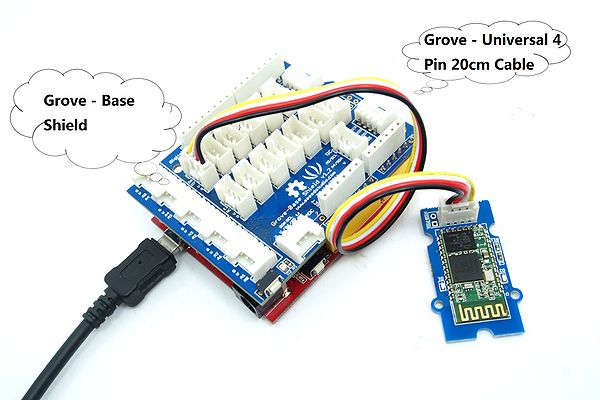
**7. Return status** \r\n+BTSTA:**xx**\r\n   
**xx** status:

* 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)

Usage

**Hardware Installation**

[](http://www.seeedstudio.com/wiki/File:Grove-Serial-Bluetooth.JPG)

**Software Programming**

Download the [demo code](http://www.seeedstudio.com/wiki/File:Bluetooth_demo_code.zip)，upload the skech to Arduino and reset it. When it at pairing status ,the green and red led blinks in interval ,and only the green led blinks 1 time/2s after connecions .When it fails or disconnects the green led blinks 2 times/s ,check if the hardware connection and baud rate is right.And please do remember the baud rate every time you set.

// This module work as Master

// Upload this sketch into Seeeduino and press reset

// Please note that this module can't be used to connected your Phone/PC as Master

#include <SoftwareSerial.h> //Software Serial Port

#define RxD 6

#define TxD 7

String retSymb = "+RTINQ=";//start symble when there's any return

String slaveName = ";SeeedBTSlave";//Set the Slave name ,caution that ';'must be included

**int** nameIndex = 0;

**int** addrIndex = 0;

String recvBuf;

String slaveAddr;

String connectCmd = "**\r\n**+CONN=";

**SoftwareSerial** blueToothSerial(RxD,TxD);

**void** **setup**()

{

**Serial**.begin(9600);

pinMode(RxD, **INPUT**);

pinMode(TxD, **OUTPUT**);

setupBlueToothConnection();

//wait 1s and flush the serial buffer

delay(1000);

**Serial**.flush();

blueToothSerial.flush();

}

**void** **loop**()

{

**char** recvChar;

**while**(1){

**if**(blueToothSerial.available()){//check if there's any data sent from the remote bluetooth shield

recvChar = blueToothSerial.read();

**Serial**.print(recvChar);

}

**if**(**Serial**.available()){//check if there's any data sent from the local serial terminal, you can add the other applications here

recvChar = **Serial**.read();

blueToothSerial.print(recvChar);

}

}

}

**void** setupBlueToothConnection()

{

blueToothSerial.begin(38400); //Set BluetoothBee BaudRate to default baud rate 38400

blueToothSerial.print("**\r\n**+STWMOD=1**\r\n**");//set the bluetooth work in master mode

blueToothSerial.print("**\r\n**+STNA=SeeedBTMaster**\r\n**");//set the bluetooth name as "SeeedBTMaster"

blueToothSerial.print("**\r\n**+STPIN=0000**\r\n**");//Set Master pincode"0000",it must be same as Slave pincode

blueToothSerial.print("**\r\n**+STAUTO=0**\r\n**");// Auto-connection is forbidden here

delay(2000); // This delay is required.

blueToothSerial.flush();

blueToothSerial.print("**\r\n**+INQ=1**\r\n**");//make the master inquire

**Serial**.println("Master is inquiring!");

delay(2000); // This delay is required.

//find the target slave

**char** recvChar;

**while**(1){

**if**(blueToothSerial.available()){

recvChar = blueToothSerial.read();

recvBuf += recvChar;

nameIndex = recvBuf.indexOf(slaveName);//get the position of slave name

//nameIndex -= 1;//decrease the ';' in front of the slave name, to get the position of the end of the slave address

**if** ( nameIndex != -1 ){

//Serial.print(recvBuf);

addrIndex = (recvBuf.indexOf(retSymb,(nameIndex - retSymb.length()- 18) ) + retSymb.length());//get the start position of slave address

slaveAddr = recvBuf.substring(addrIndex, nameIndex);//get the string of slave address

**break**;

}

}

}

//form the full connection command

connectCmd += slaveAddr;

connectCmd += "**\r\n**";

**int** connectOK = 0;

**Serial**.print("Connecting to slave:");

**Serial**.print(slaveAddr);

**Serial**.println(slaveName);

//connecting the slave till they are connected

**do**{

blueToothSerial.print(connectCmd);//send connection command

recvBuf = "";

**while**(1){

**if**(blueToothSerial.available()){

recvChar = blueToothSerial.read();

recvBuf += recvChar;

**if**(recvBuf.indexOf("CONNECT:OK") != -1){

connectOK = 1;

**Serial**.println("Connected!");

blueToothSerial.print("Connected!");

**break**;

}**else** **if**(recvBuf.indexOf("CONNECT:FAIL") != -1){

**Serial**.println("Connect again!");

**break**;

}

}

}

}**while**(0 == connectOK);

}

// This module work as Slave

// Upload this sketch into Seeeduino and press reset

#include <SoftwareSerial.h> //Software Serial Port

#define RxD 6

#define TxD 7

**SoftwareSerial** blueToothSerial(RxD,TxD);

**void** **setup**()

{

**Serial**.begin(9600);

pinMode(RxD, **INPUT**);

pinMode(TxD, **OUTPUT**);

setupBlueToothConnection();

}

**void** **loop**()

{

**char** recvChar;

**while**(1){

**if**(blueToothSerial.available()){//check if there's any data sent from the remote bluetooth shield

recvChar = blueToothSerial.read();

**Serial**.print(recvChar);

}

**if**(**Serial**.available()){//check if there's any data sent from the local serial terminal, you can add the other applications here

recvChar = **Serial**.read();

blueToothSerial.print(recvChar);

}

}

}

**void** setupBlueToothConnection()

{

blueToothSerial.begin(38400); //Set BluetoothBee BaudRate to default baud rate 38400

blueToothSerial.print("**\r\n**+STWMOD=0**\r\n**"); //set the bluetooth work in slave mode

blueToothSerial.print("**\r\n**+STNA=SeeedBTSlave**\r\n**"); //set the bluetooth name as "SeeedBTSlave"

blueToothSerial.print("**\r\n**+STPIN=0000**\r\n**");//Set SLAVE pincode"0000"

blueToothSerial.print("**\r\n**+STOAUT=1**\r\n**"); // Permit Paired device to connect me

blueToothSerial.print("**\r\n**+STAUTO=0**\r\n**"); // Auto-connection should be forbidden here

delay(2000); // This delay is required.

blueToothSerial.print("**\r\n**+INQ=1**\r\n**"); //make the slave bluetooth inquirable

**Serial**.println("The slave bluetooth is inquirable!");

delay(2000); // This delay is required.

blueToothSerial.flush();

}

FAQ

Here is the Grove - Serial Bluetooth FAQ, users can list the Frequently Asked Questions here, example as below :

* If it can be used for Arduino Mega ？

    Yes, it is . But you need to change the software serial port case not all pins on the Mega and Mega 2560 support change interrupts , so only the following can be used for software serial RX: 10, 11, 12, 13, 50, 51, 52, 53, 62, 63, 64, 65, 66, 67, 68, 69 . It is better to read more detail about [SoftwareSerial Library](http://arduino.cc/en/Reference/SoftwareSerial) .

* Bluetooth cann't work right with the demo code .

    Please check if you used the correct baud rate , if you have changged it , please do remember it because it cann't be setted to restore defaults.

Additional Idea

* [How to connect PC with Serial Bluetooth Grove](http://www.seeedstudio.com/forum/viewtopic.php?f=18&t=1436&p=5637#p5637)
* [Set up connections between two Bluetooth modules step by step](http://www.seeedstudio.com/forum/viewtopic.php?f=4&t=687)

Bill of Materials (BOM) /parts list

* [Serial port bluetooth module (Master/Slave)](http://www.seeedstudio.com/wiki/Serial_port_bluetooth_module_(Master/Slave))

Resources

* [Library for Arduino 1.0](http://www.seeedstudio.com/wiki/File:Bluetooth_demo_code.zip)
* [Schematic in PDF](http://www.seeedstudio.com/wiki/images/5/58/Serial_Bluetooth.pdf)
* [Schematic and Layout in Eagle format](http://www.seeedstudio.com/wiki/File:Grove_-_Serial_Bluetooth_eagle_file.zip)
* [Bluetooth Software instruction](http://www.seeedstudio.com/wiki/File:Bluetooth_Software_Instruction.pdf)
* [Bluetooth - module Datasheet](http://www.seeedstudio.com/wiki/File:Bluetooth_module.pdf)

How to buy

Click here to buy : [Grove - Serial Bluetooth](http://www.seeedstudio.com/depot/grove-serial-bluetooth-p-795.html?cPath=139_142)

See Also

* [Serial port bluetooth module (Master/Slave)](http://www.seeedstudio.com/depot/serial-port-bluetooth-module-masterslave-p-572.html?cPath=139_142)
* [Bluetooth Shield](http://www.seeedstudio.com/depot/bluetooth-shield-p-866.html?cPath=132_134)
* [Bluetooth Bee](http://www.seeedstudio.com/depot/bluetooth-bee-p-598.html?cPath=139_142)
* [Bluetooth Bee Standalone](http://www.seeedstudio.com/depot/bluetooth-bee-standalone-p-1157.html?cPath=139_142)