# How to use Grove Serial Bluetooth v3.0

What is Grove Serial Bluetooth v3.0

Grove - Serial Bluetooth is an easy-to-use module compatible with the existing Grove Base Shield, and designed for transparent wireless serial connection setup. In this tutorial, we will be using two Grove Serial Bluetooth modules and two Arduino to perform a wireless communication.
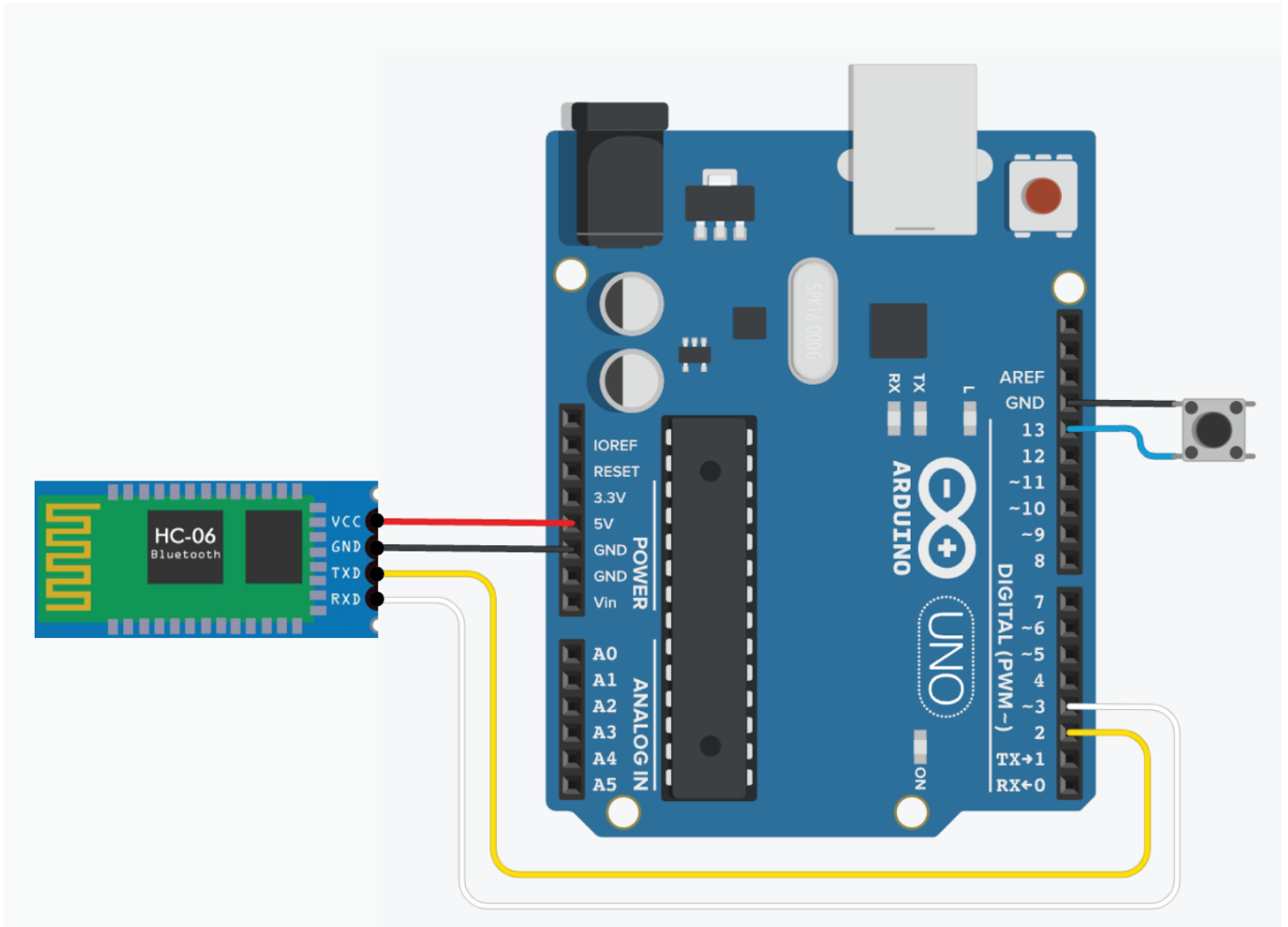
You can read more about this component **here**.
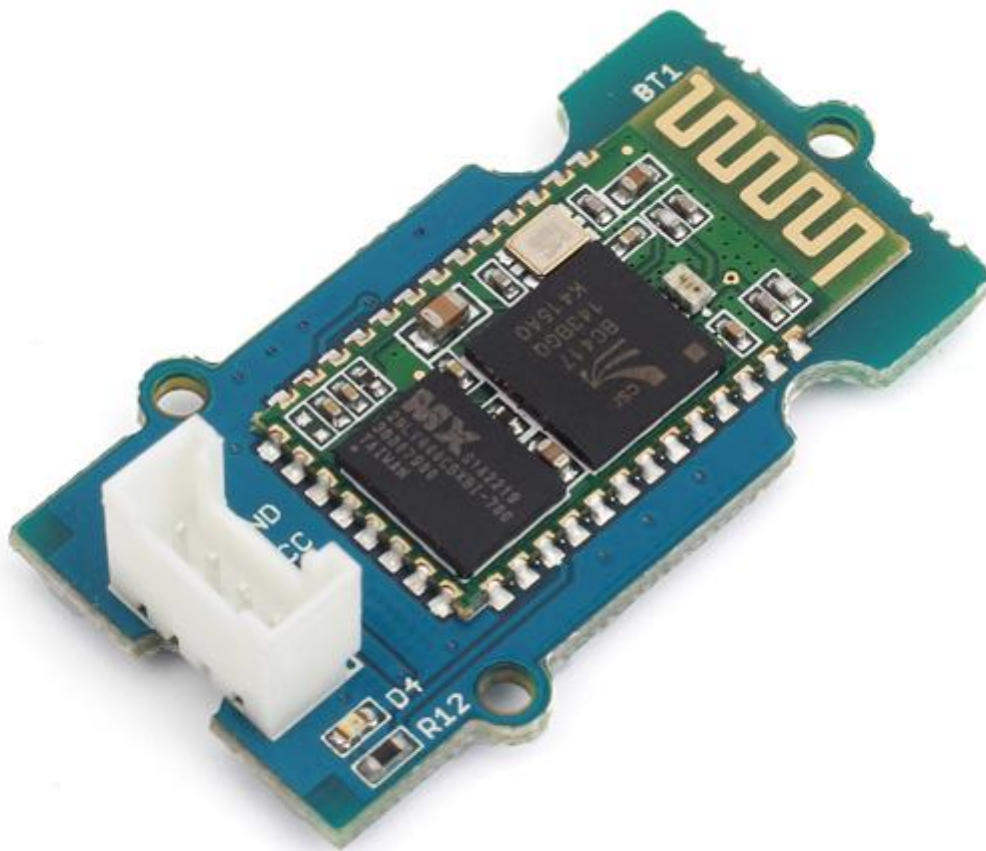
Wiring (Master - Sending data)

> **Interrupt Pins for RX/TX**
> In this tutorial, I am using an UNO which has pin 2 & 3 as the interrupts pins. Check the model you are using and change the pins accordingly.

1. VCC (Red) to 5V
2. GND (Black) to GND
3. RX (White) to pin 3 (Arduino TX)
4. TX (Yellow) to pin 2 (Arduino RX)
5. Button to GND
6. Button to pin 13

Wiring (Slave - Receving data)

1. VCC (Red) to 5V
2. GND (Black) to GND
3. RX (White) to pin 3 (Arduino TX)
4. TX (Yellow) to pin 2 (Arduino RX)
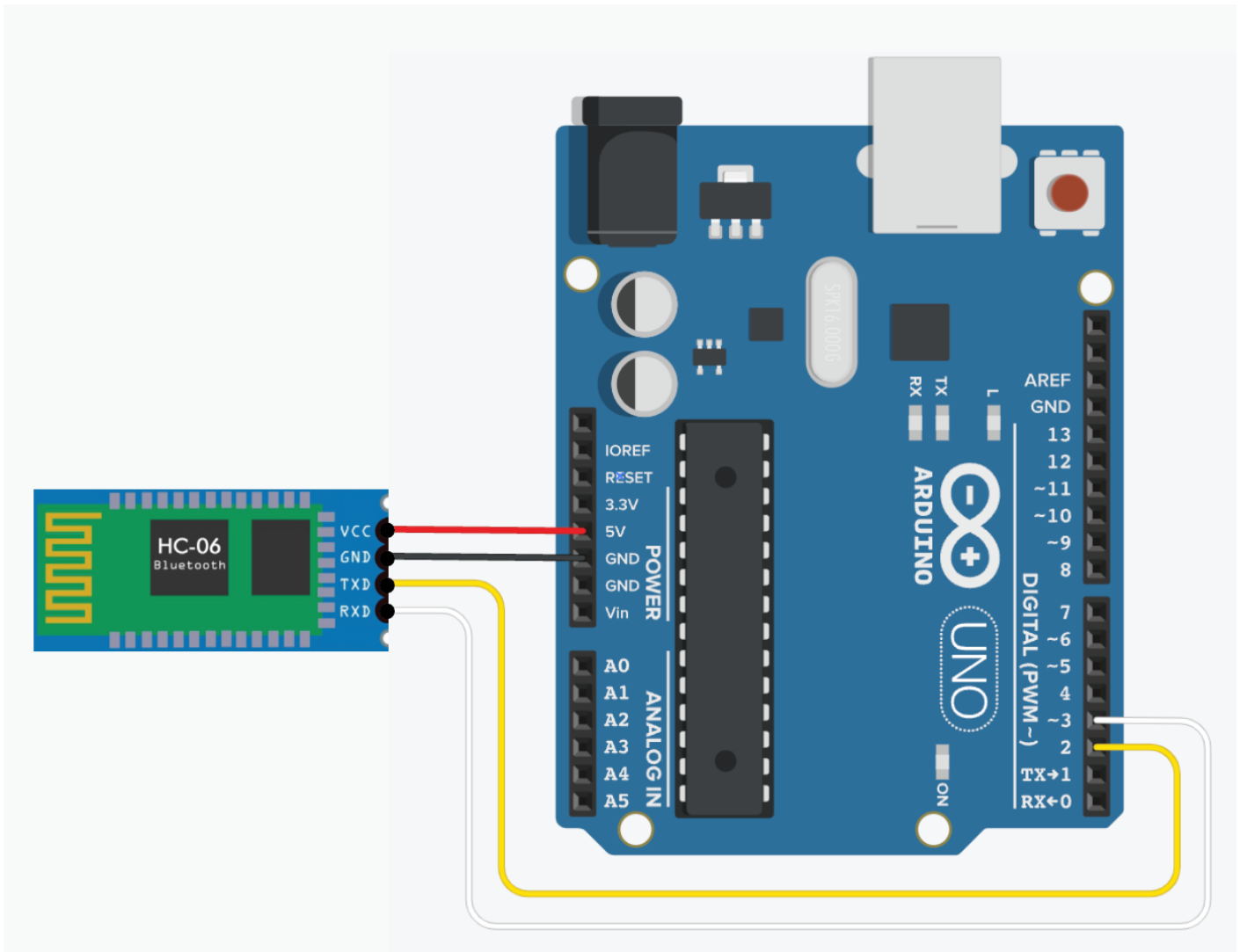
Code - Master

This code reads the signal from the button and sends it to the Slave Arduino.

```
/*
 * FM.h
 * A library for SeeedStudio Grove FM
 *
 * Copyright (c) 2012 seeed technology inc.
 * Website    : www.seeed.cc
 * Author     : Steve Chang
 * Create Time: JULY 2014
 * Change Log : Modified by loovee 2013-10-29 ,   Modified by jacob yan 2014-7-29
 *
 * The MIT License (MIT)
 *
```

```
 * Permission is hereby granted, free of charge, to any person obtaining a copy
 * of this software and associated documentation files (the "Software"), to deal
 * in the Software without restriction, including without limitation the rights
 * to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
 * copies of the Software, and to permit persons to whom the Software is
 * furnished to do so, subject to the following conditions:
 *
 * The above copyright notice and this permission notice shall be included in
 * all copies or substantial portions of the Software.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
 * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
 * AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
 * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
 * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
 * THE SOFTWARE.
 */

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

#define RxD        2
#define TxD        3

#define PINBUTTON   13                   // pin of button

#define DEBUG_ENABLED  1


SoftwareSerial blueToothSerial(RxD,TxD);

void setup()
{
   Serial.begin(9600);
   pinMode(RxD, INPUT);
   pinMode(TxD, OUTPUT);
   pinMode(PINBUTTON, INPUT_PULLUP);

   setupBlueToothConnection();
```

```
    //wait 1s and flush the serial buffer
    delay(1000);
    Serial.flush();
    blueToothSerial.flush();
}


void loop()
{

    static unsigned char state = 1;          // led off
    //Serial.println(digitalRead(PINBUTTON));


    if(digitalRead(PINBUTTON))
    {
        //state = 1-state;

        Serial.println("button on");

        blueToothSerial.print(state);

        delay(10);
        while(digitalRead(PINBUTTON))      // until button release
        {
            delay(10);
        }

        Serial.println("button off");
    }
}


/***************************************************************************
 * Function Name: setupBlueToothConnection
 * Description:  initilizing bluetooth connction
 * Parameters:
 * Return:
 ***************************************************************************/
void setupBlueToothConnection()
{

    
```

```
    blueToothSerial.begin(9600);

blueToothSerial.print("AT");

delay(400);

blueToothSerial.print("AT+DEFAULT");          // Restore all setup value to factory setup

delay(2000);

blueToothSerial.print("AT+NAMESeeedMaster");    // set the bluetooth name as "SeeedMaster" ,the length of
bluetooth name must less than 12 characters.

delay(400);

blueToothSerial.print("AT+ROLEM");           // set the bluetooth work in slave mode

delay(400);

blueToothSerial.print("AT+AUTH1");

    delay(400);

blueToothSerial.print("AT+CLEAR");          // Clear connected device mac address

    delay(400);

    blueToothSerial.flush();

}
```

Code - Slave

```
/*
 * FM.h
 * A library for SeeedStudio Grove FM
 *
 * Copyright (c) 2012 seeed technology inc.
 * Website    : www.seeed.cc
 * Author     : Steve Chang
 * Create Time: JULY 2014
 * Change Log : Modified by loovee 2013-10-29  ,  Modified by jacob yan 2014-7-29
```

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

#define RxD        2
#define TxD        3

#define PINLED     13

#define LEDON()    digitalWrite(PINLED, HIGH)
#define LEDOFF()   digitalWrite(PINLED, LOW)

#define DEBUG_ENABLED  1

SoftwareSerial blueToothSerial(RxD,TxD);

void setup()
{
   Serial.begin(9600);
```

```
    pinMode(RxD, INPUT);
    pinMode(TxD, OUTPUT);
    pinMode(PINLED, OUTPUT);
    LEDOFF();

    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(recvChar == '1')
            {
                LEDON();
            }
            else if(recvChar == '0')
            {
                LEDOFF();
            }
        }
    }
}




/****************************************************************************
 * Function Name: setupBlueToothConnection
 * Description:  initilizing bluetooth connction
 * Parameters:
 * Return:
 ***************************************************************************/
```

```
void setupBlueToothConnection()

{

    blueToothSerial.begin(9600);

    blueToothSerial.print("AT");

    delay(400);


    blueToothSerial.print("AT+DEFAULT");           // Restore all setup value to factory setup

    delay(2000);

    blueToothSerial.print("AT+NAMESeeedBTSlave");    // set the bluetooth name as "SeeedBTSlave" ,the length of

    bluetooth name must less than 12 characters.

    delay(400);


    blueToothSerial.print("AT+PIN0000");           // set the pair code to connect

    delay(400);


    blueToothSerial.print("AT+AUTH1");             //

    delay(400);


    blueToothSerial.flush();


}
```

Connection

After uploading both codes to both Arduinos, reset them simultaneously. The LEDs on the modules will be flashing and wait until they stay on, then they are connected.

You may need to repeat a couple of times to get them connected, it's all about patience.