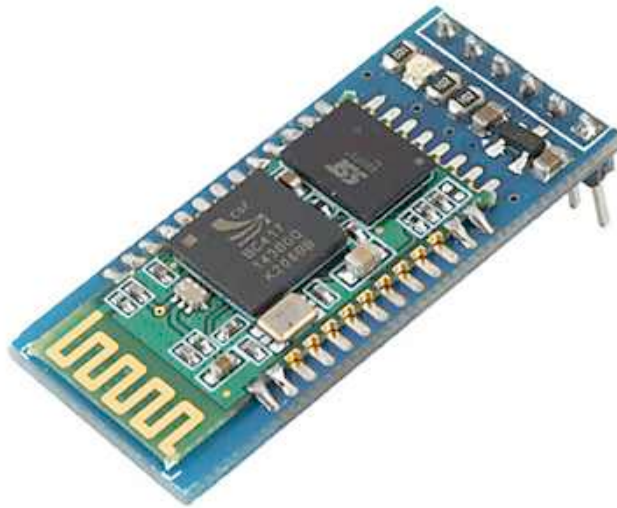


Serial Bluetooth Module with Arduino



For use of Serial Bluetooth Module with Arduino (Slave Mode)

- 1) Upload the Arduino sketch below first and press reset
- 2) Connect bluetooth to Arduino as follows:

Bluetooth TX -----> Arduino Uno RX (Pin 6)

Bluetooth RX -----> Arduino Uno TX (Pin 7) via the voltage divider!

Bluetooth GND -----> Arduino GND pin

Bluetooth Vcc (3.3V) -----> Arduino 3.3 V pin

```
// This module work as Slave
```

```
// Upload this sketch into Arduino 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=SeedBTSlave\r\n"); //set
the bluetooth name as "SeedBTSlave"
}

```

```

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();
}

```

For use of Serial Bluetooth Module with Arduino (Master Mode)

- 1) Upload the Arduino sketch below first and press reset
- 2)Connect bluetooth to Arduino as follows:

Bluetooth TX -----> Arduino Uno RX (Pin 6)
 Bluetooth RX -----> Arduino Uno TX (Pin 7) via the voltage divider!
 Bluetooth GND -----> Arduino GND pin
 Bluetooth Vcc (3.3V) -----> Arduino 3.3 V pin

```

/ This module work as Master
// Upload this sketch into Arduino 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 = ";SeedBTSslave";//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=SeedBTMaster\r\n");//set the bluetooth
name as "SeedBTMaster"
  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);
}

```

For use of Serial Bluetooth Module with Arduino Mega

For use with Arduino Mega 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](#) .