



- [Home](#)
- [Tutorials](#)
- [Forum](#)
- [About](#)
- [Contact](#)

0 Product(s) in cart |

Total \$0.00 |

[Shopping cart](#)

Arduino with XBee (ZigBee) Tutorial

Tutorial for XBee Module and XBee Shield with Arduino



Overview

The XBee (ZigBee) Module provides an alternative way to transfer data without the use of wires. Here's how it's done:

The XBee (ZigBee) uses a wireless 2.4GHz transceiver to communicate with another XBee (ZigBee) module. Furthermore, XBee (ZigBee) modules are capable of communicating with more than one XBee (ZigBee) module. Thus, it means you can create a network of XBee modules all over the place! As long as they are in range, of course.

Now for the technical details, the XBee module features:

802.15.4 Protocol created by the IEEE foundation.

Data rate of 250KBps (Kilobits per second).

Can be used indoors and outdoors.

Range is from 100ft-300 for standard XBee modules and 300ft-1 Mile for XBee Pro Modules (depending on where it's used and the line of sight from one XBee to the next XBee).

The standard XBee has a 1mW transmit power and the XBee Pro has a 60mW transmit power.

Serial communication for interfacing.

No configuration is required out of the box.

Default baud rate is 9600bps. Although, you can change the configuration of how fast you want to transmit but for this tutorial we will just leave the baud rate at default.

In this tutorial, we will provide how to use the XBee module using the XBee shield in combination with an Arduino Development board out-of-the-box. The standard XBee module will be used; however, you can use the XBee Pro Module as well.

Tutorial

Tools Required:

2 - [Arduino Development Boards.](#)

2 – [XBee Shields.](#)

2 – [XBee Modules.](#)

1 or 2 – [Breadboards.](#)

3 – [LEDs \(Light Emitting Diodes\)](#)

2 – 1K Ohm Resistors

[Hookup Wire](#)

A Power Source for the Arduino Development Board. (USB or Power Supply)

Thinking Cap

*Please note that you can purchase a [XBee Kit here](#)

Learn

The XBee module is an awesome accessory to have especially if you're trying to communicate or control devices without the use of wires.

So how does wireless communication work?

Well in a nutshell, wired and wireless communication use a transmitter and a receiver to communicate.

Unlike wired communication, wireless communication uses electromagnetic waves tuned to a certain frequency in which both a transmitter and receiver can understand the tuned frequency from both sides (See the illustration below).

Now you may think “wow that’s cool!” Yes it is cool, however, there’s a slight challenge involved when using wireless communication. When transmission of a signal (the data) gets involved with the real world that we live in, sometimes data can become disrupted to known and unknown factors. These factors can be considered as noise or interference which could comprise another radio frequency signal, conduction of the signal or a combination of other factors such as weather and

what's blocking the signal line of sight communication. However, engineers have understood the problems that have arisen from these factors and implemented schemes to avoid interference.

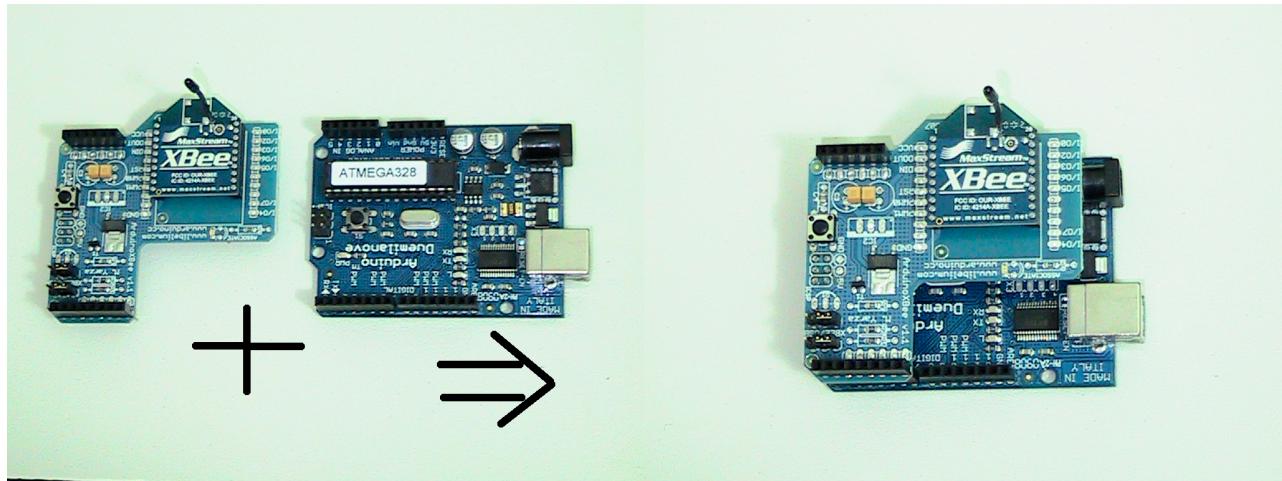
In this case, the XBee module uses a particular wireless protocol to talk to another XBee module known as DSSS or Direct Sequence Spread Spectrum. We won't get into too much detail about this but in simple terms the technology used in the XBee uses a transmission pattern of multiple frequencies (frequency hopping) to avoid interference whereby the receiving XBee understands the pattern. Since the both devices understand the frequency pattern, then it is assumed that data can be received successfully with minimal interference.

Enjoy

Now that you have learned and hopefully we didn't bore you with the concept of wireless communication, let's get started!

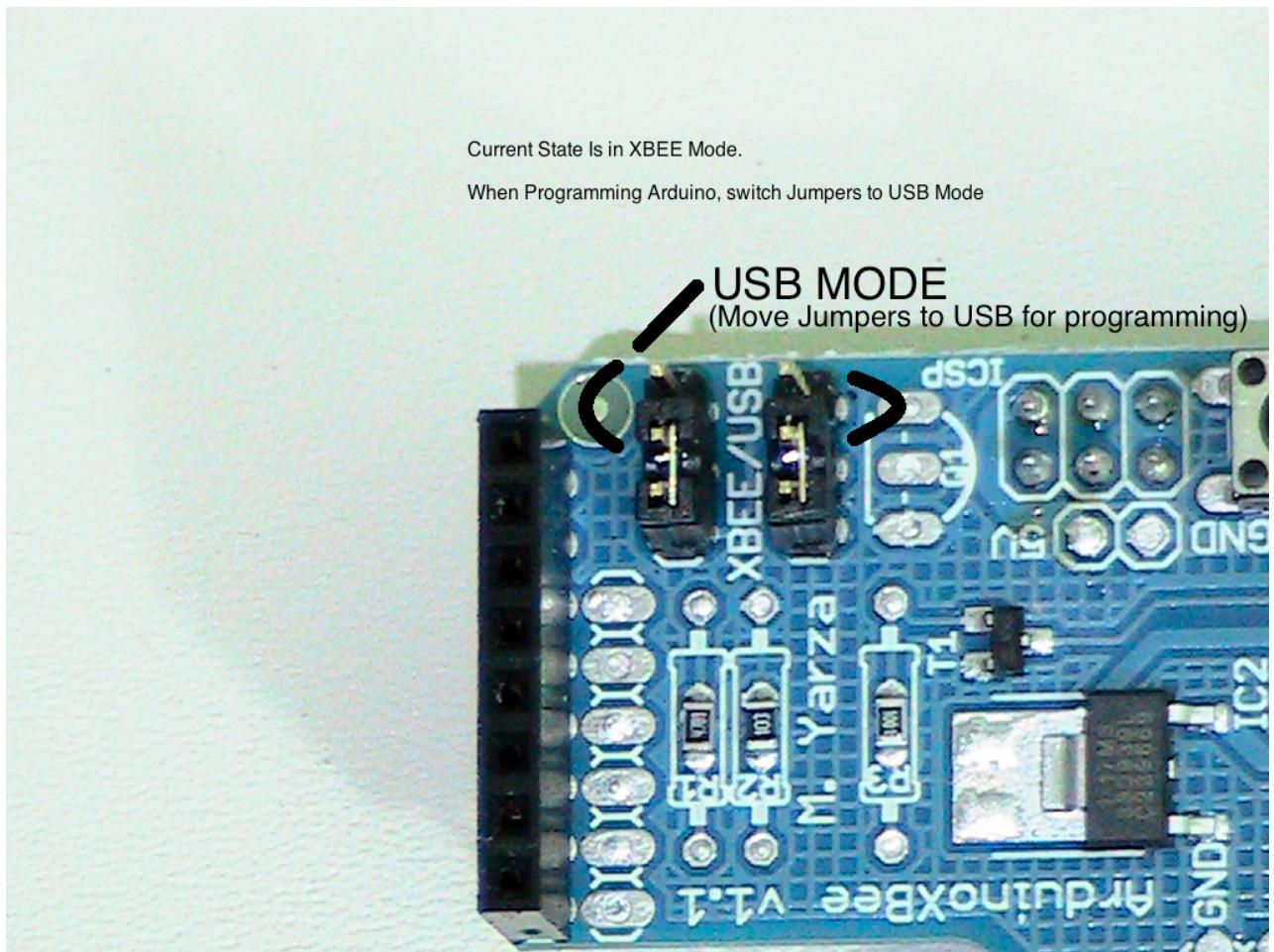
Part 1: One Transmitter, One Receiver plus LEDs

Preliminary Step: Connect the XBee Shield and XBee module together. Then attach XBee shield to the Arduino board.



Please make sure that both Jumpers of the XBee shield are placed on 'USB' side or else you will not be able to program the Arduino (See Diagram Below). Then plug in power to start programming.

PIN DIAGRAM



We are going to need a program that allows you to transmit a signal from one XBee module to another XBee module. Thus, one Arduino development board will be the transmitter.

So here's the program below for the transmitter:

```

void setup()
void setup() {
    Serial.begin(9600);           //Arduino initial setup
                                //set baud rate on Arduino to 9600
}
void loop() {                  //loop program
    Serial.print("L");          //Send the character L
    delay(2000);                //wait 2 seconds
    Serial.print("H");          //Send the character H
    delay(2000);                //wait 2 seconds
    Serial.print ("X");         //Send the character X
    delay(2000);                //wait 2 seconds
}

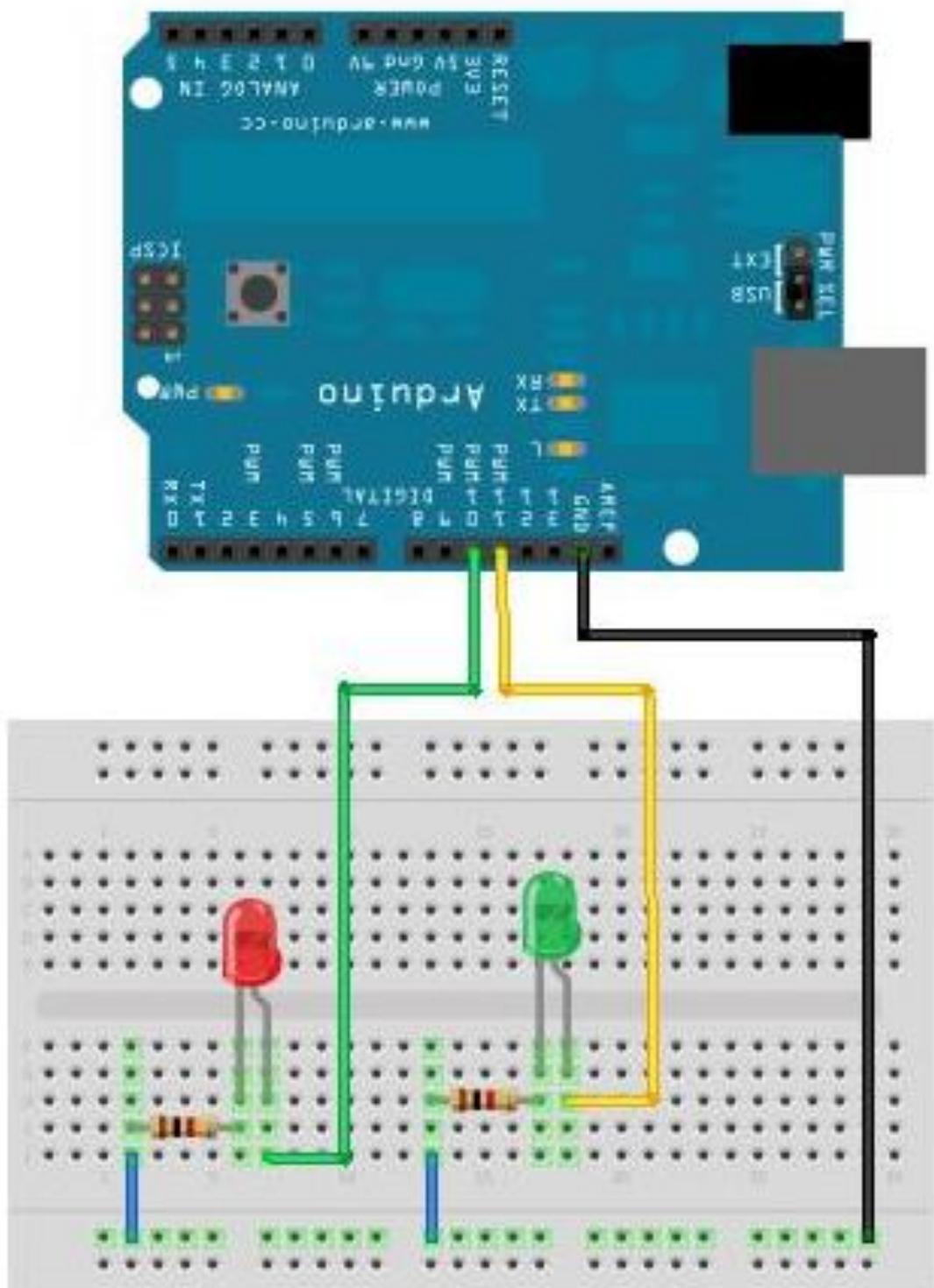
```

Once you have this programmed without any problems, unplug the power from the Arduino.

Then set both Jumpers of the XBee shield on the 'XBEE' side or else you will not be able to transmit your data through the XBee module (See Pin Diagram Above).

The XBee-Arduino receiver is going to need some way of knowing it is receiving correct data and this is where LEDs (Light Emitting Diodes) come in place. Using the diagram below we are going to setup each of the LEDs so that it can emit a color when a certain data has been received.

Using a breadboard, connect a 1K Ohm resistor to ground and connect the Green LED in series to the resistor and connect that in series to Digital pin number 10. Then connect a 1K Ohm resistor to ground and connect the Red LED in series with the resistor and connect that in series to Digital pin number 11. For a clearer understanding see the picture below.



Then, we are going to need a program that allows you to receive a signal from one XBee module to another XBee module.

Thus, the second Arduino development board will be the receiver.

(Once again, please make sure to unplug power from the Arduino set the jumper settings on the shield to 'USB' side. After that bring power back to the Arduino for programming)

So here's the program below for the receiver:

```

int LEDredPin = 10;           //Assign Pin 10 for Red LED
int LEDgreenPin = 11;         // Assign Pin 11 for Green LED
void setup() {                //Arduino initial setup

    Serial.begin(9600);        //Set baud rate for Arduino to 9600
    pinMode(LEDredPin, OUTPUT); //Make Digital Pin 10 as Output Pin
    pinMode(LEDgreenPin, OUTPUT); // Make Digital Pin 11 as Output Pin

}

void loop() {                  // loop program
    if (Serial.available()) {  //If there is data in the Serial Line
        int dataByte = Serial.read(); //save data into integer variable dataByte
        Serial.print(dataByte, BYTE); //Print to screen the variable received

        if(dataByte == 'H'){       //if the variable is the character H
            digitalWrite(LEDredPin, HIGH); //then Red LED pin is ON
            digitalWrite(LEDgreenPin, LOW); // then Green LED pin is OFF

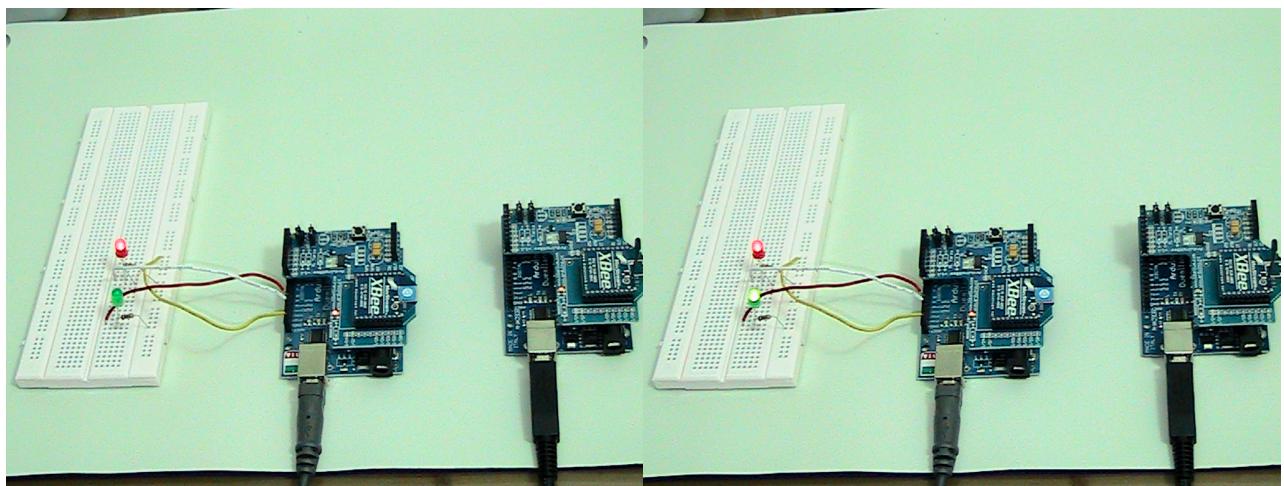
        }

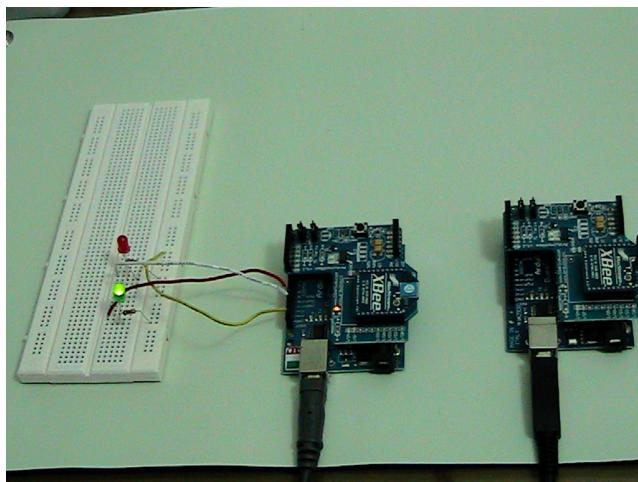
        else if(dataByte == 'L') { // if the variable is the character L
            digitalWrite(LEDgreenPin, HIGH); // then Green LED pin is ON
            digitalWrite(LEDredPin, LOW); // then Red LED pin is OFF
        }

        Else                      //otherwise
        { digitalWrite(LEDredPin, HIGH); // Green LED pin is ON
            digitalWrite(LEDgreenPin, HIGH); // Red LED pin is ON
        }
    }
}

```

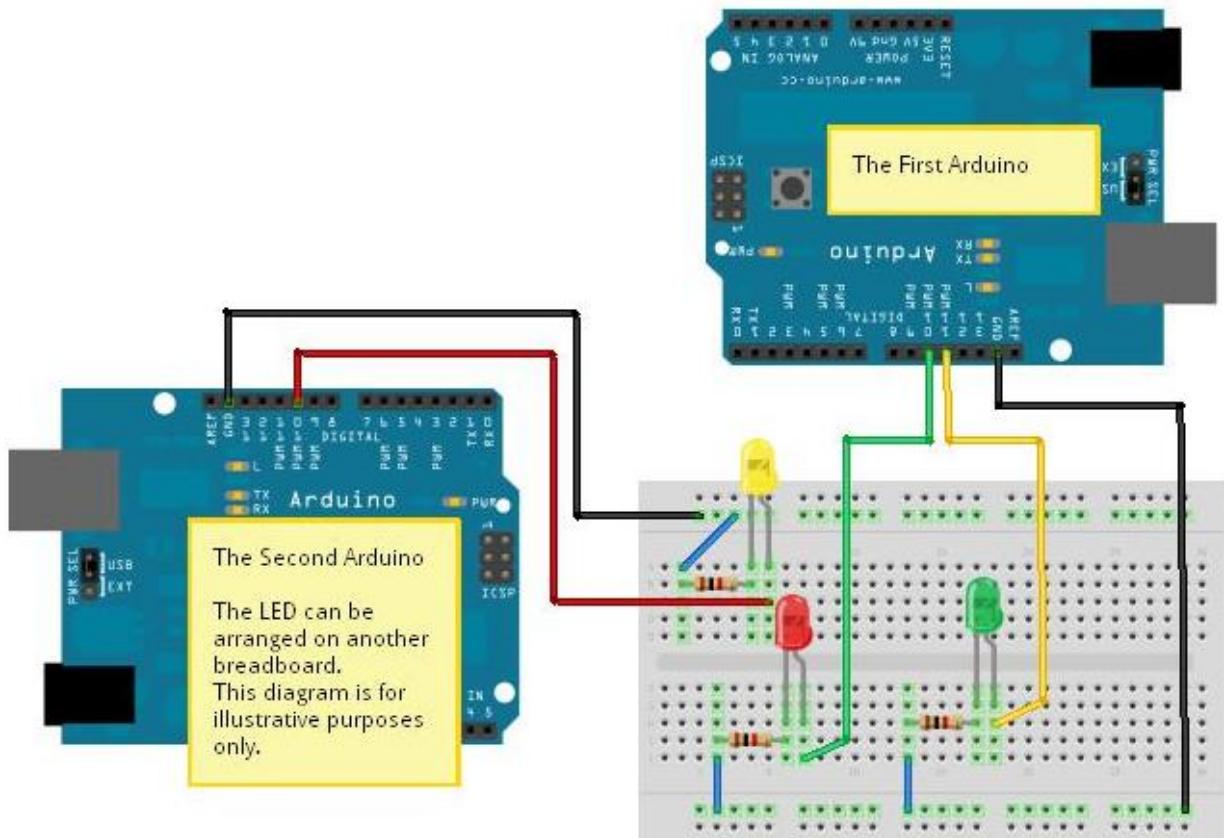
Once you have successfully programmed the receiver Arduino, unplug power of the Arduino and make sure to set the jumper setting of the shield to 'XBEE' side (See Pin Diagram above). Fire it up! Plug up both Arduino development boards and watch the action! Isn't it amazing???





Part 2: Transceiver. Let me hear you say something and reply!

Now let's get each Arduino to send a message to each other and make them reply (a.k.a. transceivers). So now we need another LED setup to get feedback for the other Arduino. Take the breadboard and connect 1K Ohm Resistor to ground and connect that in series to the LED and then to digital pin 10 of the Second Arduino. See the diagram below. Let's keep the previous configuration of LEDs for the first Arduino.



Please make sure that both Jumpers of the XBee shield are placed on 'USB' side or else you will not be able to program the Arduino (See Pin Diagram Above). Then plug in power to start programming.

For the second Arduino that initiates the conversation, we are going to use this code:

```

int LEDredPin =10;                                //Red LED assigned to Digital Pin 10
void setup() {                                     //setup procedures
pinMode(LEDredPin, OUTPUT);                      //Make digital pin 10 an output pin
  
```

```

    Serial.begin(9600);                      //Serial communication
}

void loop() {                                //loop sequence

    delay(1000);                            //wait for a second (optional, but recommended for startup
    Serial.print('x',BYTE);                //Send the letter x as a byte

    if (Serial.available() > 0){           //if the serial port is ready to receive

        byte retcommand = Serial.read();   //create a byte variable retcommand and read serial input
        if (retcommand == byte(44))        //if the retcommand is equal to byte 44

        {
            digitalWrite(LEDredPin,HIGH);      //turn on the LED
            delay(4000);                     //hold it for 4 seconds
            digitalWrite(LEDredPin,LOW);       //turn off the LED

        }

        else if (retcommand == byte('n'))    //else if the return is equal to byte n

        {
            digitalWrite(LEDredPin,HIGH);      //then turn LED on
            delay(1000);                     //wait for 1 second
            digitalWrite(LEDredPin,LOW);       //turn off LED
            delay(1000);                     //wait for 1 second
            digitalWrite(LEDredPin,HIGH);      //then turn on LED again
            delay(1000);                     //wait for 1 second
            digitalWrite(LEDredPin,LOW);       //turn off LED again
            delay(1000);                     //wait for 1 second

        }
        delay(6000);                         //wait for 6 seconds (optional or value can be decreased)
    }

}

```

Once you have successfully programmed the second Arduino, unplug power of the Arduino and make sure to set the jumper setting of the shield to 'XBEE' side (See Pin diagram above).

Now let's setup the program with the first Arduino

Please make sure that both Jumpers of the XBee shield are placed on 'USB' side or else you will not be able to program the Arduino (See Pin Diagram Above). Then plug in power to start programming.

Here's the programming for the first Arduino:

```

int LEDredPin = 10;                      //set Red LED for digital pin 10
int LEDgreenPin = 11;                     //set Green LED for digital pin 11

void setup() {                           //initial setup

    Serial.begin(9600);                  //set baud rate of Arduino to 9600
    pinMode(LEDredPin, OUTPUT);         //Set digital pin 10 to Output
    pinMode(LEDgreenPin, OUTPUT);       //Set digital pin 11 to Output

}

void loop() {                            //loop program

    if (Serial.available() > 0){        //if serial line is available

        byte command = Serial.read();   //store data into byte command from serial
        if (command == byte('x'))       //if command = byte x

        {
            delay(3000);               //then wait 3 seconds (optional, but recommended)
            Serial.print(44,BYTE);       //send byte 44
            digitalWrite(LEDredPin,HIGH); //turn on the Red LED
            digitalWrite(LEDgreenPin,LOW); //turn off the Green LED

        }

        Else                          //else

        {
            delay(3000);               //wait 3 seconds
            Serial.print('n',BYTE);     //send byte of character n
            digitalWrite(LEDredPin,LOW); //turn off Red LED

        }

    }

}

```

```

    digitalWrite(LEDgreenPin,HIGH);      //turn on Green LED
}

}

```

After that, Fire it up! Plug up both Arduino development boards and watch the action! Wow, they are both chatting, excellent! Also, try hitting the reset button on the second Arduino and you will see that the first Arduino does not recognize the code and responds accordingly.

Imagine

Now that you have learned, programmed and hopefully enjoyed this tutorial, here are some ideas that you could use to spark the imagination:

XBee Communicator with LCD Display: Use two Serial LCDs and find out what each Arduino is talking about.

Door Controller or Alarm: Use a door sensor and a XBee module with Arduino to notify if someone has entered your home.

Brewski Detector: Use a temperature/door sensor and a XBee module with Arduino to notify if your beer fridge is down or if someone is grabbing your beer!

Robot Guidance control: Use an XBee module with Arduino to control a robot!

Additional Information

For more information on other tutorials of the Arduino and the XBee Shield, we recommend visit the Arduino's site and/or software: <http://www.arduino.cc>

Filed under: by Hanguru

[24 Comments »](#)

24 Responses to “Arduino with XBee (ZigBee) Tutorial”

1. [orchard bank](#), on [November 25th, 2010 at 9:15 pm](#) Said:

I'm thankful for this precious great web page; this is the kind of subject that sustains me though out the day. We've usually heard been not too long ago desiring shut to inside your web-site proper just after I noticed about these from a shut pal and was delighted when I was in a position to acquire it pursuing seeking out for some time. Becoming a enthusiastic blogger, I'm happy to find out other many people taking gumption and adding towards the neighborhood. I just wished to remark to exhibit my comprehension to get a distribute because it is especially inviting, and lots of writers really don't get the credit score they deserve. I'm constructive I'll be back again yet again and can send a couple of of my friends.

2. [Hanguru](#), on [November 29th, 2010 at 8:48 am](#) Said:

Thanks for the comment. We try our best to make sure you learn, enjoy and imagine electronics.
We love what we do! If you have questions or even suggestions, let us know because we care how you think.

3. [Devora Brechner](#), on [December 3rd, 2010 at 3:26 am](#) Said:

i'm adding your blog rss feed so that i can see your new posts. keep up the good work!

4. [healthy diet for kids](#), on [January 8th, 2011 at 4:22 am](#) Said:

I give rise to look on behalf of such a article on behalf of a sustained era, credit a percentage.

5. [Cody](#), on [March 17th, 2011 at 8:32 am](#) Said:

This guide was VERY helpful. I am a high school senior and I'm currently working on a robotic hand project that will require two arduinos to talk to each other. One arduino is sending signals to the other telling it that one of our 5 bend sensors has changed resistor value and the second arduino responds by moving the corresponding servo. Is it possible for the second arduino to send back information and turn an LED attached to the first board if it comes in contact with an object? i.e. grasps an object and the servo doesn't complete its full rotation. Thank you so much. Any and all help is appreciated.

6. Hanguru, on March 17th, 2011 at 9:42 am Said:

Yes it is possible but you are going to need some type of feedback from the second Arduino when it has sensed an object. e.g. contact switch, photocell sensor.

Then the next step would be to have the second board to send a message to the first board using an interrupt or you could have the second board stop operation when it has sensed a touching of the object or change in light and then send a command to the first board.

You will require if/else or do/while statements to complete this operation for stopping the operation of the second board.

If you could give a little more detail and then we will gladly be able to help.

In the meantime, check out our photocell tutorial([Click here](#)) to see if this works, you can attach a photocell to the robotic finger and then detect darkness of the object, or use a contact switch.

Chris, Support, Electrojoystick.com

7. Cody, on March 17th, 2011 at 12:54 pm Said:

My friend and i are trying to make a robot hand wirelessly controlled by a data glove with bend sensors. There will be two arduinos working together. Arduino 1 will receive data from the glove. Arduino 2 will control the hand. The problem we seem to be having is getting number 2 to send feedback to number 1.

we want to have number 2 send data back to number 1 tell it that there is an object in its grasp, stopping all motor movement. The photocell option seems like it would work, but how complicated would it be to implement? Wouldn't we have to constantly recalibrate it depending on the current lighting?

We don't necessarily want a code directly written for us. We both are novices to programming, but would love a good shove in the right direction. Code snippets and some short examples would be greatly appreciated.

Also having a LED lighting up telling us which servo(s) are gripping something.

Thank you for all of your help

8. Hanguru, on March 17th, 2011 at 2:09 pm Said:

Ok,

it seems I had a logical error in understanding what you were trying to accomplish.

If Arduino 1 controls the glove and Arduino 2 controls the robotic hand with servos, then there is no way possible to for Arduino 2, to send a message to Arduino that it is gripping an object. All of this operation is determined from Arduino 1 receiving the glove information.

So what you'll need to is determine the resistance value (or voltage value) for holding that object and you will have to determine the set value and place that as the threshold (grip value) in your program.

Then, Arduino 1 will need to send a data to Arduino 2 to stop the motor.

If you want to achieve what you're trying to do then both of those devices (bend sensor and control the robotic hand) will need to be attached to Arduino 2. Arduino 1 can send data to start the movement, then when Arduino 2 senses the resistance value to a certain gripped value, then you can have Arduino 2 send a signal to Arduino 1. Then Arduino 1 can reply to Arduino 2 to stop the motor. This is one method of doing it but it would seem rather illogical to do but you can do it this way!

If you have another question or reply, please feel free.

Chris, Support, Electrojoystick.com

9. Hanguru, on March 17th, 2011 at 2:29 pm Said:

Another thought...

You could use a LM339 (voltage comparator) to determine the logic voltage coming from the motor and then have Arduino 2 send a message to Arduino 1 that the motor has stopped. It is a little complex and kind of tricky but I believe you can do it.

Let me know, if you are having trouble.

10. **Cody**, on [March 17th, 2011 at 7:23 pm](#) Said:

Alright, so we kind of had it wrong. How would we go about using the LM339? What would the programming be like if we were to use it to sense if the motor was moving or not? You have already answered many of our questions. Thank you again Hanguru. The speedy responses are very great!

11. **Cody**, on [March 29th, 2011 at 7:26 am](#) Said:

Any reply to the comment of above would be very very helpful.
thank you

12. **Hanguru**, on [March 29th, 2011 at 6:29 pm](#) Said:

Hi Cody,
I apologize for the delay. After thinking about using the LM339, I don't think you should use it because you may have some issues. Normally, LM339 is similar to the analog to digital port on the Arduino. Thus, if you were to read the logic voltage from the motor then you may have some issues with motor operating properly. So, it is recommended to have the bend sensors and the motors connected to one Arduino as previously stated and read the resistance values from the bend sensors .

13. **Sean**, on [June 20th, 2011 at 9:53 am](#) Said:

Great tutorial! I was wondering if you could touch on Zigbee with more than two devices? What changes? Is there any addressing involved? If you set-up a mesh type network how is data managed when devices are transmitting at the same time? I guess the assumption is that this is all done via the Xbee modules – correct?

Thanks

14. **Hanguru**, on [July 9th, 2011 at 7:23 am](#) Said:

Ok we will see what we can do about having another tutorial of more than two Zigbees with the Arduino. Thanks for your comment!

15. **Agustin Sevilla**, on [August 4th, 2011 at 1:06 pm](#) Said:

Firstly, thank you so much for a simple and excellent tutorial. This is the first example I have found that shows how quickly you can get xbees running without jumping into how to configure them right off the bat. I am currently trying to figure out how to send several values from one Arduino to another to control a small robot I have hacked together. I want to send it values from 2 pots and several buttons. A sample of values would be:

512 512 0 0

Any tips on how to go about this? If I understand correctly, xbee series 1 can only send one byte at a time, so do I need to set up some sort of loop to listen for all 4 values? Or perhaps send a "stop" bit once transmission is complete?

Thanks again for your awesome work!

16. **Hanguru**, on [August 4th, 2011 at 11:04 pm](#) Said:

Hi and thanks for your comment.

For your question of receiving bytes, you need to determine the number of bytes available.

So you would count the number of bytes coming in; e.g. 4 bytes in your case that you have determined.

Then you need to put the incoming data into an array; and yes, you would need to put it into a loop if the Arduino is constantly getting data from the robot.

Thus you would use:

```
int ser_byte[3]; // up here declare your array ser_byte to an integer, remember array starts at 0, not 1  
  
void setup()  
{  
//setup your serial parameters  
}  
  
void loop()  
{  
while (Serial.available() <4) //read to see if there are less than five bytes available in the queue
```

```
{
//intentionally left empty to count number of bytes
}

for (int count =0; count <4; count++)
{
ser_bytes[count] = Serial.read; //store the data into each array
}

//Then do what you must with data afterwards e.g. compare it to another variable, etc.
}
```

I hope this helps, sorry for the delay and please try to use our Forum. Thanks again for your contribution.

17. **Agustin Sevilla**, on [November 8th, 2011 at 2:16 pm](#) Said:

Hi Hanguru! Sorry for the delay, and thank you so much for your reply! This was super duper helpful! I was able to get my XBee RC system working on my robot perfectly! You can have a look here: <http://bit.ly/puLZ4V>

18. **Hanguru**, on [November 9th, 2011 at 5:32 pm](#) Said:

Hey Awesome work! Keep up the good work and always remember to learn, enjoy and imagine Electronics!

19. **Protoplasm**, on [December 3rd, 2011 at 7:37 pm](#) Said:

Hi

I am planning to build a plane with the capacity to fly for long distance (4-6km) and i would like to ask if the 2 Xbee must be the same model. If i have an Xbee Pro as a transmitter and an Xbee as a receiver, will it make a difference? Is there any antenna that will provide these type of range

With regards
Protoplasm

20. **Bobb Craig**, on [December 10th, 2011 at 1:03 am](#) Said:

I was just wondering if you gone any further with a tutorial about multiple Xbee's? I am trying to figure out (just started planning) how to connect a weather shield, garden shield and a home monitoring (temp/humidity and basement water level) all into one system. So if you write a multi-zbee as well as you wrote the single zbee tutorial ... I would be way ahead of the game on the planning of the communications side of this project.

21. **Hanguru**, on [December 20th, 2011 at 5:39 pm](#) Said:

Hi, Sorry for the delay.

Here's what you can do:

You will keep the XBee devices channels the same (default). They are automatically designed to communicate with each other out of the box according to the datasheet from Digi. Therefore, this makes life a little easier.

I presume you're going to have one XBee device as a master that receives all data coming in?

If so, then you will have XBee A send a byte (e.g. 0xFA or whatever data) to XBee B and XBee B will reply the data. Your XBee B has to be configured to acknowledge the data coming in.
so you will need something like (sample code below)

```
byte dataref = 0xFA; //reference data
byte data_in; //incoming data
...
void setup(){
Serial.begin(9600);
}

...
void loop()
```

```
{  
if (Serial.available() >0)  
data_in = Serial.read(); //read what is coming in from the Serial Input  
}  
//then compare that byte to the reference;  
if (dataref == datain) {  
Serial.print("This is XBee B replying")  
}
```

With XBee C, you will have XBee A send a byte (e.g. 0xFB) to XBee C and XBee B will reply data.

You can also do the same for XBees' B and C. It's a matter of how you set it up.

If you can send me a sample code of what you have started then I can be able to help you.

Once again, my apologies for the delay. Hanguru.

22. **Hanguru**, on [December 20th, 2011 at 5:45 pm](#) Said:

Hi, I'm afraid to say that the XBee module is not going to fit your criteria because the range for this device is 40 m – 1.6 km.

Is there a reason why you're trying to use the XBee modules?

23. **Sreejith**, on [December 22nd, 2011 at 9:11 am](#) Said:

HI,

Thanx for the simplest tutorial.

I am in the midst of a project in which i need to communicate one remote PLANT with a central computer. i wish to implement arduino+ xbee in the remote plant and can u please tell me how to implement a two way communication between (computer+ xbee) and (arduino+ xbee). thanx in advance

24. **Hanguru**, on [December 24th, 2011 at 12:52 pm](#) Said:

Hi,

Yeah there are several ways to do this:

One, you can use an RS232 chip and connect that to the computer and the XBee or purchase the RS232 from Digi. If you want to use USB, then there is one from Adafruit.com, it is called the XBee Adapter Kit.

Any of these configurations will help connect XBee to your computer.

Leave a Reply

<input type="text"/>	Name (required)
<input type="text"/>	Mail (will not be published) (required)
<input type="text"/>	Website
<input type="text"/>	

Submit Comment


 CAPTCHA Code *

seven \times 6 =

Login Status

Not logged in

» [Login](#)

Our Products

[Development Boards](#)
[Components](#)
[Displays](#)
[Sensors](#)
[Network / Wireless](#)
[Building / Testing](#)
[Electronic Kits](#)
[Tools/ Cables](#)
[Motors and Gears](#)

Service

- [About Us](#)
- [Shipping FAQ](#)
- [Payment Options](#)
- [Return Policy](#)
- [Terms & Conditions](#)
- [Tracking](#)
- [Order Status](#)
- [Privacy Policy](#)
- [Contact Us](#)

Tutorial

Categories

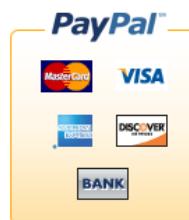
- [Electronics Tutorial 101 – Basics](#) (1)
- [Inspiration](#) (1)
- [Product Tutorials](#) (9)

Archives

- [March 2012](#)
- [January 2012](#)
- [July 2011](#)
- [January 2011](#)
- [December 2010](#)
- [November 2010](#)

- [October 2010](#)

[Email Marketing](#)



Copyright (c) 2012 [Electrojoystick.com](#)
Learn, Enjoy, Imagine Electronics!

2010-11-10