

# XBee Module Interfacing with Arduino

Published February 14, 2019

0 1

R Rishabh Jain (/users/rishabh-jain)  
Author



Interfacing XBee Module with Arduino

In previous tutorials we have learned about ZigBee protocol and its architecture (<https://circuitdigest.com/article/zigbee-introduction-architecture-at-commands>), and also learned about wireless communication between two Xbee modules (<https://circuitdigest.com/article/communication-between-two-computers-using-xbee-modules>). Now in this tutorial we will **interface XBee module with Arduino Uno board**. The XBee connected with Arduino board will act as a receiver and it will communicate wirelessly with other XBee module which is serially connected with the laptop using a Explorer Board. So lets explore further for **Arduino wireless communication using XBee**.

## Hardware Requirements

1. 1 x Arduino Uno
2. 2 x XBee Pro S2C modules (any other model can be used)

3. 1 x Xbee explorer board (optional)
4. 1 x Xbee Breakout board (optional)
5. USB cables
6. LEDs

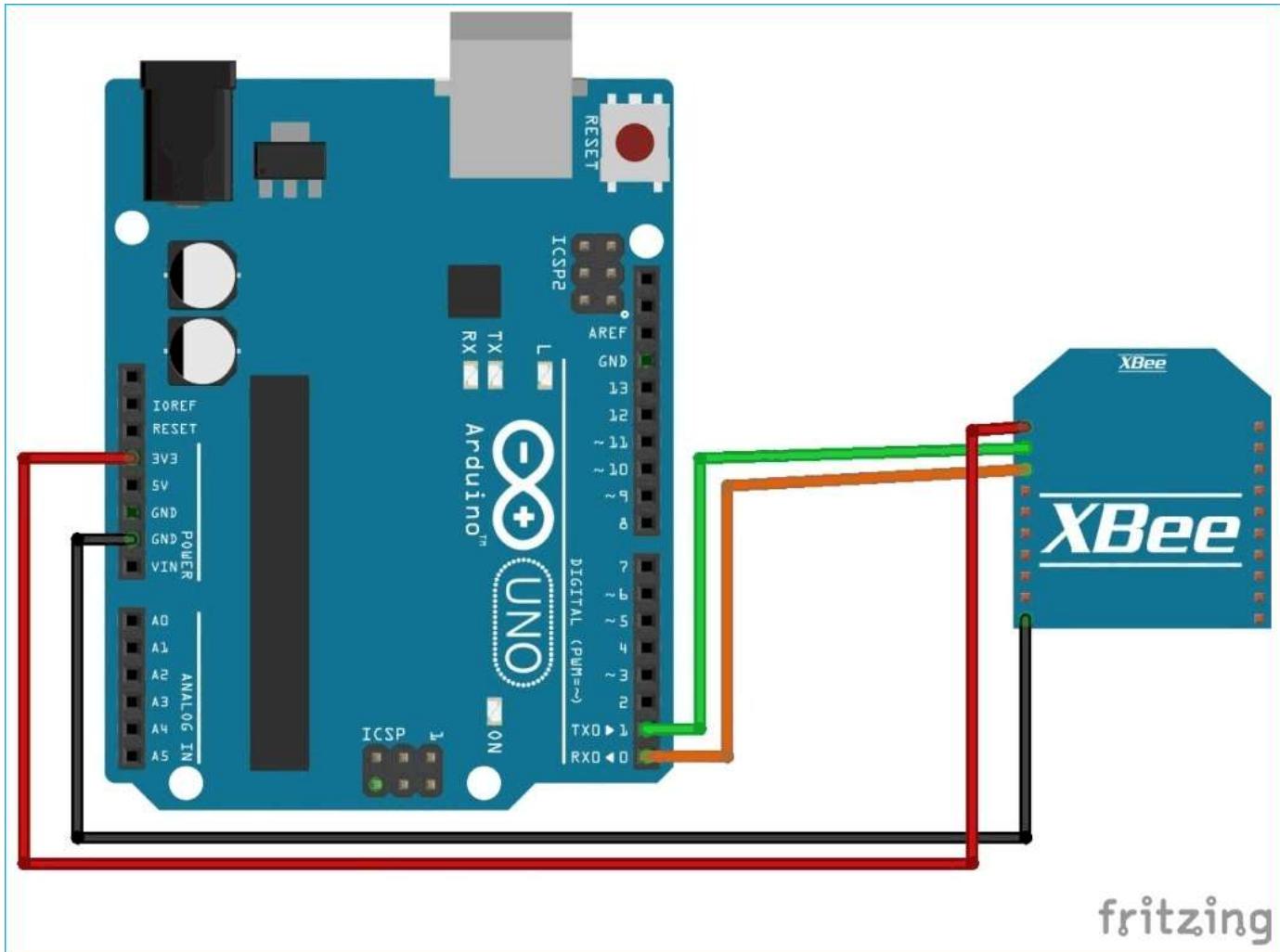
## Configuring Xbee Modules using XCTU

As we have learnt in [previous tutorials](https://circuitdigest.com/article/zigbee-introduction-architecture-at-commands) (<https://circuitdigest.com/article/zigbee-introduction-architecture-at-commands>) that the Xbee module can act as a Coordinator, Router or an End device but it need to be configured to work in desired mode. So before using the **XBee modules with Arduino**, we have to configure these modules using XCTU software.

To connect Xbee module with the laptop, a USB to serial converter or specifically designed explorer board is used. Just hook up the Xbee module to the Explorer board and plug it with the laptop using USB cable.

If you don't have any converter or explorer board, then an **Arduino board can be used as a USB to serial device** which can easily communicate with the Xbee and laptop. Just upload blank sketch in Arduino board and now it can behave like a USB to Serial converter.

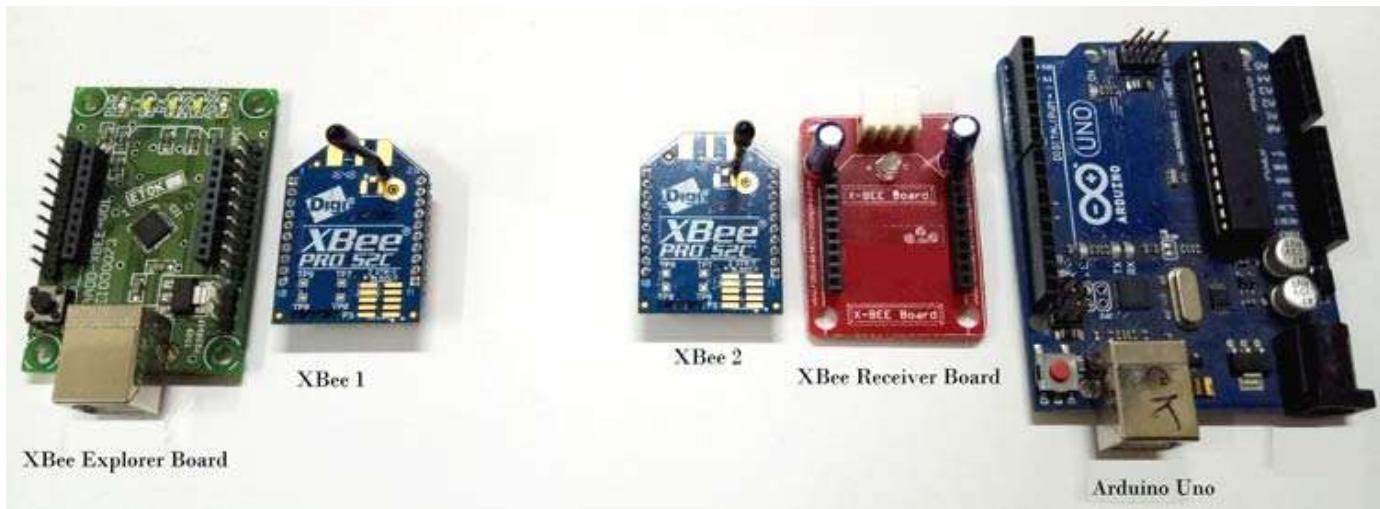
Connections for **interfacing ZigBee module with Arduino** are shown in the circuit diagram.



(/fullimage?i=circuitdiagram\_mic/Circuit-Diagram-for-XBee-Module-Interfacing-with-Arduino.png).

#### Connections:

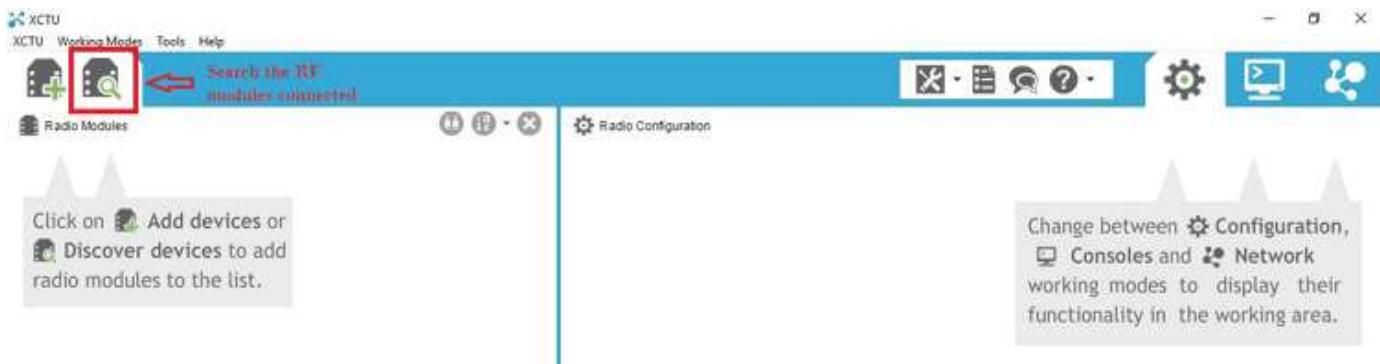
1. Tx (pin2) of XBee -> Tx of Arduino board
2. Rx(pin3) of Xbee -> Rx of Arduino board
3. Gnd(pin10) of Xbee -> GND of Arduino board
4. Vcc (Pin1) of Xbee -> 3.3v of Arduino board



Here in this tutorial, a **Explorer board** is used to configure the XBee modules.

Download the XCTU software from this [link](#) ([https://www.digi.com/resources/documentation/digidocs/90001526/tasks/t\\_download\\_and\\_install\\_xctu.htm](https://www.digi.com/resources/documentation/digidocs/90001526/tasks/t_download_and_install_xctu.htm)) and install it. After downloading and installing the XCTU software, open it and make sure your XBee module is properly connected. Check the COM port of the Arduino board in device manager.

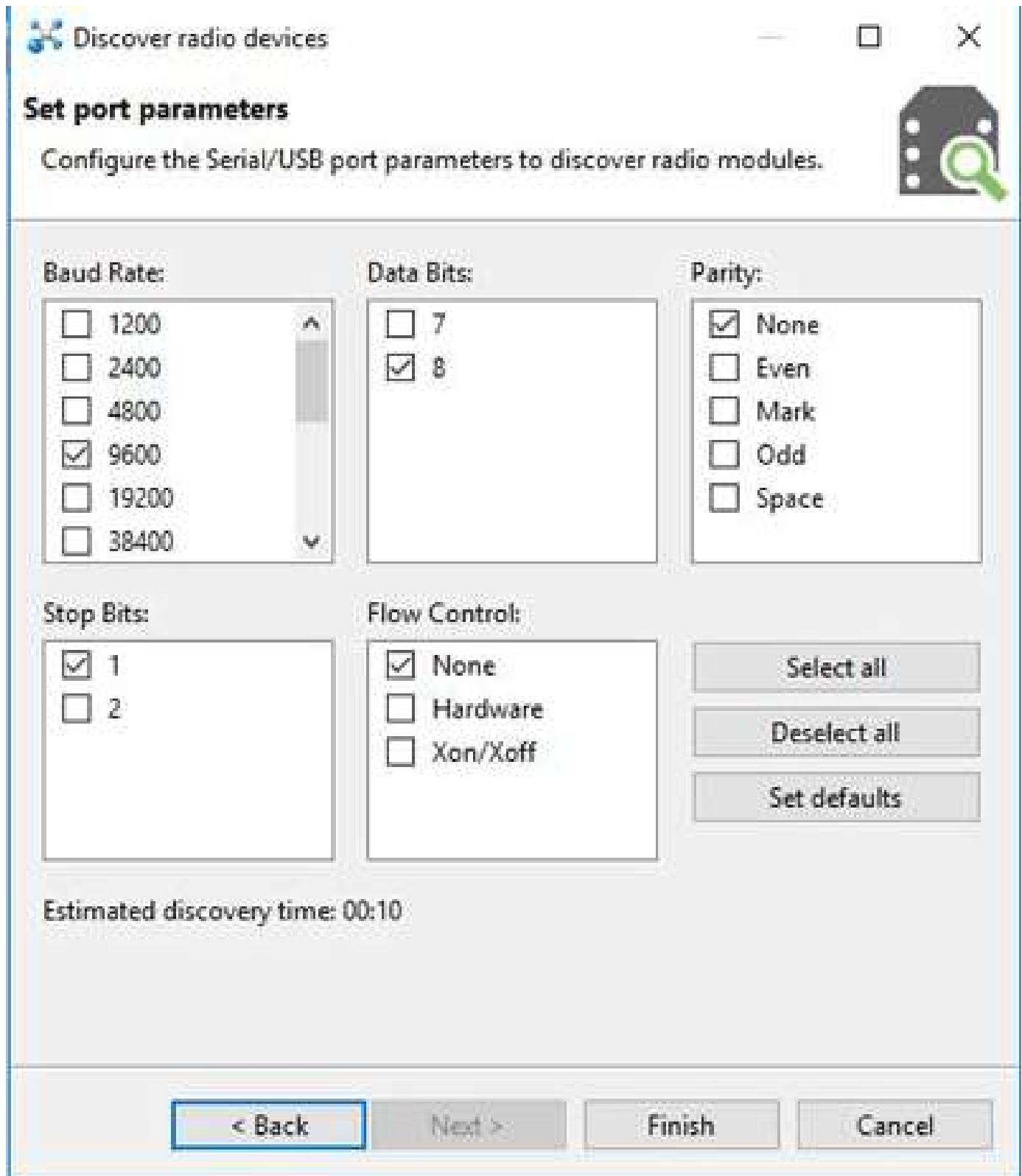
**Step 1:-** Now, click on the search button. This will show you all the RF devices connected with your laptop. In our case, it will show only one XBee module.



**Step 2:-** Select the Serial port of the Explorer board/Arduino board and click on Next.



**Step 3:-** In the next window, set the USB port parameters as shown below and click on Finish.



**Step 4:-** Select the Discovered device and click on **Add selected device**. This process will add your Xbee module to XCTU dashboard.

Discovering radio modules...

Search finished. 1 device(s) found



1 device(s) found

---

Devices discovered:

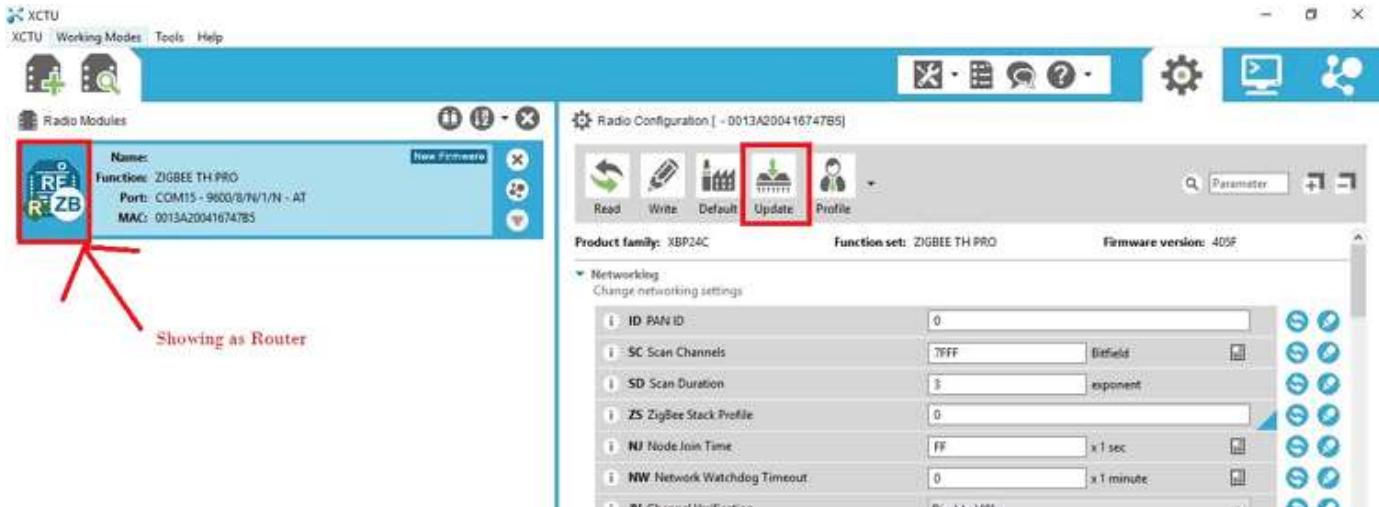
 	<p>Port: COM15 - 9600/8/N/1/N - AT</p> <p>Name:</p> <p>MAC Address: 0013A200416747B5</p>
---	--

---

Your device was not found? [Click here](#)

**Step 5:-** Now, you can configure your Xbee module in this window. Use either AT commands or put the data manually. As you can see, there is R showing on the left panel which means Xbee is in router mode. We have to make it Coordinator for the transmitter part.

First, update the Firmware by clicking on the Update firmware.



**Step 6:-** Choose the Product family of your device which is available on back of XBee module. Select function set and firmware version as highlighted below and click on **Update**.

 Update firmware

### Update the radio module firmware

Configure the firmware that will be flashed to the radio module.



Select the product family of your device, the new function set and the firmware version to flash:

② Product family	Function set	Firmware version
XBP24C	802.15.4 TH PRO DigiMesh 2.4 TH PRO ZIGBEE TH PRO	2001 (Newest)

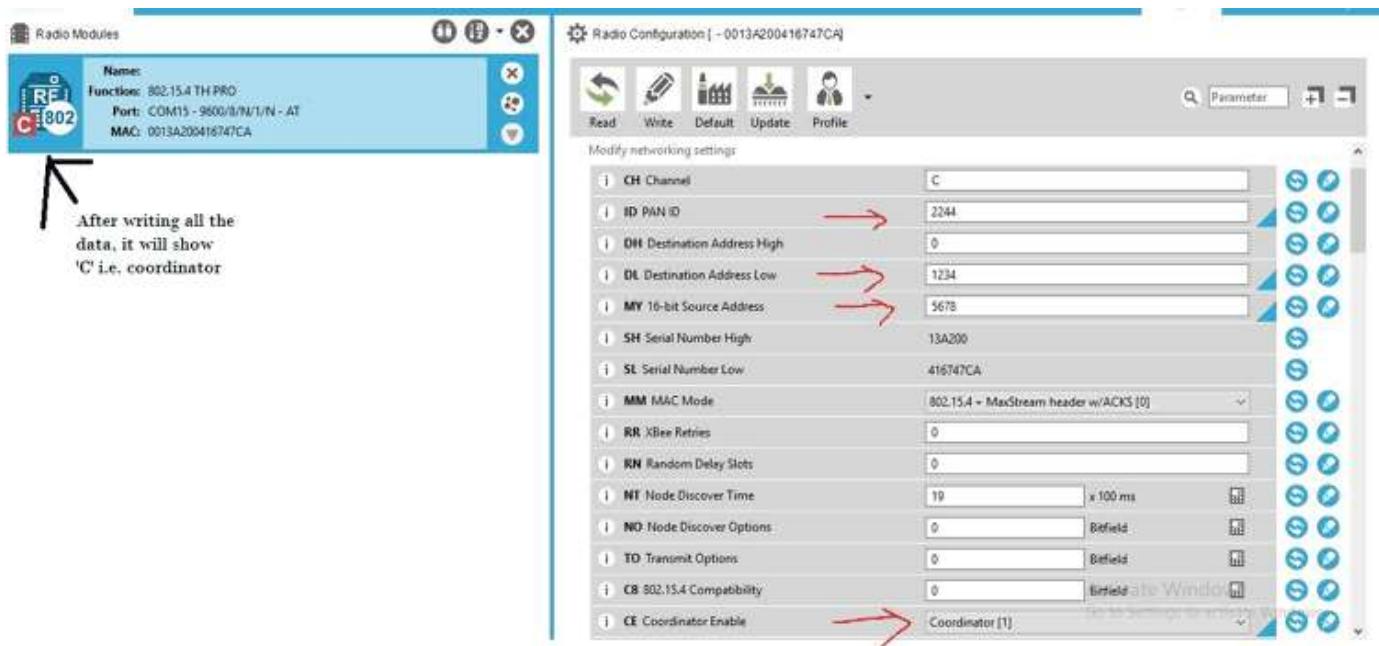
Can't find your firmware? [Click here](#)

[View Release Notes](#)

Force the module to maintain its current configuration. [Select current](#)

[Update](#) [Cancel](#)

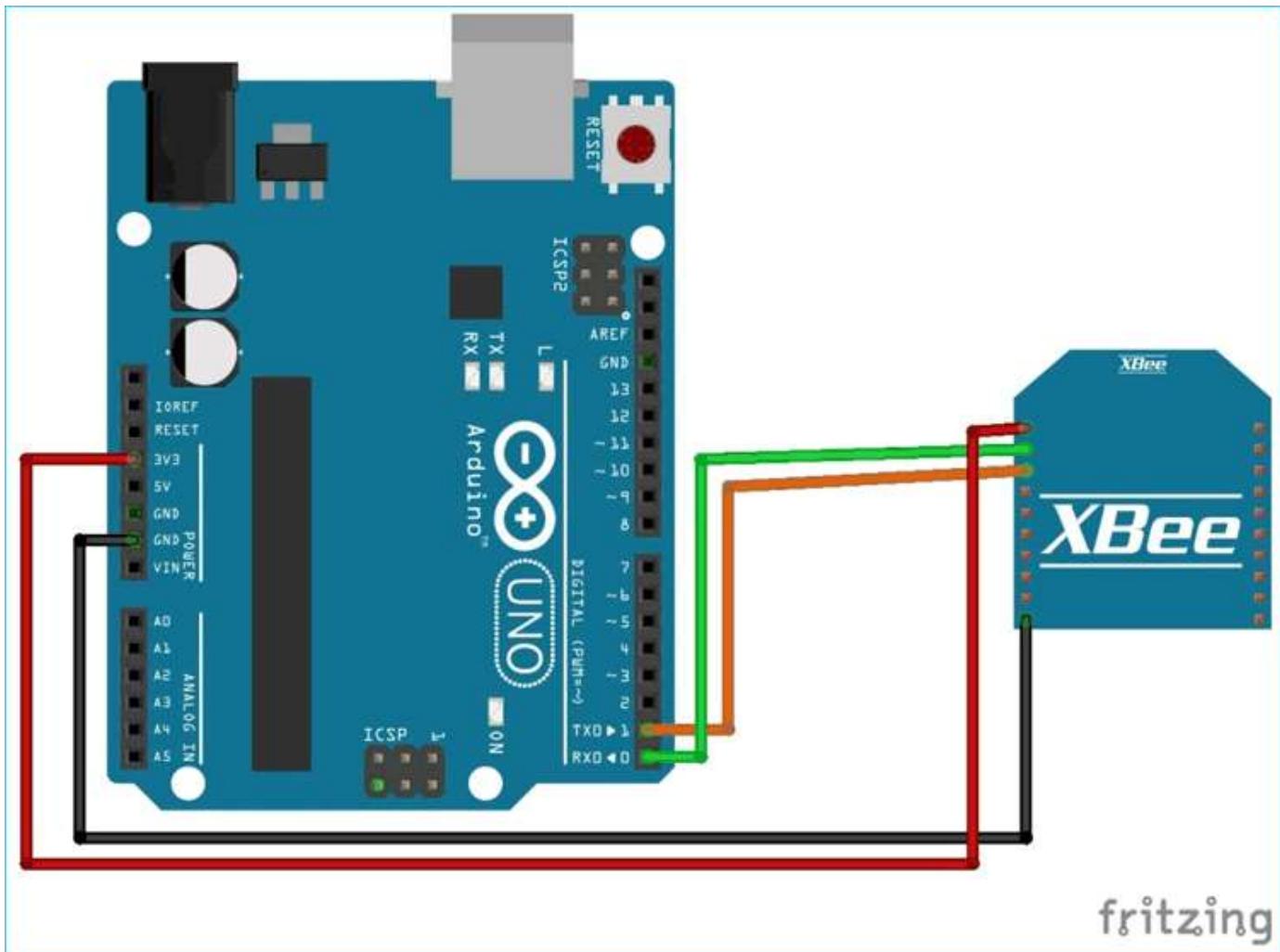
**Step 7:-** Now, you have to give ID, MY and DL data to make connection with other XBee. ID remain same for both the modules. Only MY and DL data interchange i.e. **MY for the receiver XBee becomes DL of the transmitter XBee (coordinator) and DL for the receiver XBee becomes MY of the transmitter XBee.** Make CE as **Coordinator** and then hit the **Write** button. As shown below.



**Step 8:-** After writing the above data to the transmitter part, plug out it from the explorer board and plug it in the second Xbee module. Repeat the same process as above only changes are the DL, MY, and CE. **As we will make the second Xbee as End device so in CE drop down menu, select the End device and hit the Write button.**

**Step 9:-** Now, our Xbee modules are ready to interface with the Arduino board. We will **connect the transmitter Xbee to the laptop and receiver Xbee with the Arduino board**. Then give commands to the receiver part using laptop.

#### Circuit Diagram for Receiver Part:



#### Connections:

1. Tx (pin2)of XBee -> Rx of Arduino board
2. Rx(pin3) of Xbee -> Tx of Arduino board
3. Gnd(pin10) of Xbee -> GND of Arduino board
4. Vcc (Pin1) of Xbee -> 3.3v of Arduino board

If you are using the Arduino board to connect the transmitter ZigBee with the laptop, connections will be same as for the programming the ZigBee.

## Programming and Testing XBee communication using Arduino

Now, we will write a code for the receiver Arduino to **switch ON the LED whenever receiver part receives 'a'**, and **blink the LED whenever it receives 'b'**, for other characters LED will remain OFF.

Code is simple and easily understandable. We will just check the incoming characters using **Serial.available()** function and store this character in a variable using **Serial.read();** function and match it with 'a' and 'b'. If match is correct then perform the task specified in the condition. **Complete code for Receiver part is given in the end.** Upload the code in the Receiver part Arduino. Remove the Tx and Rx wires of Xbee before uploading.

Now, we all set to **test our transmitter and receiver.** To give command to the transmitter part, we will use XCTU's console terminal. Click on the Console icon near the settings option. Then, click on Open button to connect the Xbee to the laptop.

Enter 'a' in Console log. You will see that LED will turn ON for 2 seconds and after that enter 'b' to make the led blink for 5 times.



You can also connect the transmitter Xbee to the Arduino board, just change the receiver code little bit. In place of **Serial.read()** function, use **Serial.println()** function to send the characters.

Check the **Demonstration Video** given below.

This **XBee-Arduino setup** can be used to make many useful wireless applications like Home automation system, chatting room etc.

## Code

```
int led = 13;
int received = 0;
int i;
void setup() {
    Serial.begin(9600);
    pinMode(led, OUTPUT);
}

void loop() {
    if (Serial.available() > 0) {
        received = Serial.read();

        if (received == 'a'){
            digitalWrite(led, HIGH);
            delay(2000);
            digitalWrite(led, LOW);
        }
        else if (received == 'b'){
            for(i=0;i<5;i++){
                digitalWrite(led, HIGH);
                delay(1000);
                digitalWrite(led, LOW);
                delay(1000);
            }
        }
    }
}
```

## Video

Arduino Xbee Module Interfacing



## Tags

[Arduino \(/Tags/Arduino\)](#)   [Arduino Uno \(/Tags/Arduino-Uno\)](#)   [Interfacing \(/Tags/Interfacing\)](#)

[Xbee \(/Tags/Xbee\)](#)   [Zigbee \(/Tags/Zigbee\)](#)   [Wireless \(/Tags/Wireless\)](#)

## Comments

Submitted by Sayali (/users/sayali-0) on Tue, 06/16/2020 - 22:16

Permalink (/comment/31985#comment-31985)

### XBee.begin(9600); gives an error (/comment/31985#comment-31985)

Hi,

I found your article very helpful. I am doing same as you implemented in this article.

I have 1 Arduino UNO and 2 XBee Pro S3B modules and 1 XBee Shield. I have connected 1 XBee module(which is configured as Master) to sheild and another one (Slave) has connected to Arduino UNO.

If I send "a" or "b" letters from xctu then Led on arduino should blink. This is the working of my code.

In the receiver code, I have included XBee.h lib in Arduino. But it gives an error at **XBee.begin(9600);** and all other functions of XBee like **received = XBee.read();**

error: expected unqualified-id before ":" token

I am a learner and I am using Arduino and XBee communication for the first time in my project.

Plz guide.

Thanks

Log in (/user/login?destination=/microcontroller-projects/arduino-xbee-module-interfacing-tutorial%23comment-form) or register (/user/register?destination=/microcontroller-projects/arduino-xbee-module-interfacing-tutorial%23comment-form) to post comments



**G5PZ-X High VDC PCB Relay (<https://bit.ly/3UQwwW6>)**

Omron's new G5PZ-X PCB relay comes in a compact package with 20 A at 200 VDC rated load

(<https://bit.ly/3UQwwW6>)

**3D Power Packaging® for Low Power DC/DC converters**

(<https://bit.ly/3DUq1eB>)

Tiny Size-vs-High Specs, this range of RECOM's DC/DC Converters utilizes minimal PCB footprint

(<https://bit.ly/3DUq1eB>)

**AVR-IoT Cellular Mini Development Board (<https://bit.ly/3KjeOW5>)**

Microchip's AVR-IoT Cellular Mini features Sequans Monarch 2 GM02S cellular module

(<https://bit.ly/3KjeOW5>)

**C4AK Series High Temperature, Long Life DC-Link Film Capacitors**

(<https://bit.ly/3RcSkrY>)

KEMET's C4AK series film capacitors feature long life and high voltage

(<https://bit.ly/3RcSkrY>)

**MYC0409 Series 48 V Charge Pumps (<https://bit.ly/3C5Af9N>)**

Murata's ultra-thin, high-efficiency, 72 W charge pump modules support 48 V Bus architecture

(<https://bit.ly/3C5Af9N>)



(<https://bit.ly/3R7yePY>)

**Multi-Conductor Classics Cables (<https://bit.ly/3R7yePY>)**  
Belden's rugged multi-conductor cables ensure system uptime in rugged environments



(<https://bit.ly/3SxfDhc>)

**High-Current, High-Voltage Cartridge Fuses – 526 Series and 527 Series (<https://bit.ly/3SxfDhc>)**  
Littelfuse's high-current, high-voltage cartridge fuses – 526 series and 527 series



(<https://bit.ly/3To2bxo>)

**Traceability Solutions (<https://bit.ly/3To2bxo>)**  
Omron Automation Traceability Solutions  
Omron's vast track, trace and control experience helps build...



## Join 100K+ Subscribers

Your email is safe with us, we don't spam.

Type your email address

Subscribe

Be a part of our ever growing community.



Semicon Media is a unique collection of online media, focused purely on the Electronics Community across the globe. With a perfectly blended team of Engineers and Journalists, we demystify electronics and its related technologies by providing high value content to our readers.

(<https://www.facebook.com/circuitdigest/>)  (<https://twitter.com/CircuitDigest>)  (<https://www.youtube.com/channel/UCy3CUAIYgZdAOG9k3IPdLmw>)  (<https://www.linkedin.com/company/circuit-digest/>)

## COMPANY

[Privacy Policy \(/privacy-policy\)](#) [Cookie Policy \(/cookie-policy\)](#) [Terms of Use \(/terms-of-use\)](#) [Contact Us \(/contact\)](#) [Advertise \(/advertise\)](#)

## PROJECT

[555 Timer Circuits \(/555-timer-circuits\)](#) [Op-amp Circuits \(/op-amp-circuits\)](#) [Audio Circuits \(/audio-circuits\)](#) [Power Supply Circuits \(/smps-power-supply-circuits\)](#) [Arduino Projects \(/arduino-projects\)](#) [Raspberry Pi Projects \(/simple-raspberry-pi-projects-for-beginners\)](#) [STM32 Projects \(/stm32-projects-and-tutorials\)](#) [ESP8266 Projects \(/esp8266-projects\)](#) [PIC Projects \(/pic-microcontroller-projects\)](#) [AVR Projects \(/avr-microcontroller-projects\)](#) [8051 Projects \(/8051-microcontroller-projects\)](#) [ESP32 Projects \(/esp32-projects\)](#) [IoT Projects \(/internet-of-things-iot-projects\)](#) [PCB Projects \(/diy-pcb-projects\)](#) [Arduino ESP8266 Projects \(/arduino-esp8266-projects\)](#) [All Microcontroller Projects \(/microcontroller-projects\)](#)

## OUR NETWORK



(<https://circuitdigest.com>)



(<https://components101.com>)



(<https://iotdesignpro.com>)

Copyright © 2022 Circuit Digest (/). All rights reserved.