GET STARTED TO WORK WITH XBee
LEARN HOW TO INTERFACE WITH MINDSTORMS AND ARDUINO
Content

XBee Configuration ................................................................................................................................................. 3

Hardware: ................................................................................................................................................................. 3

Must Configure the Xbee modules to speak the same ID First .................................................................................. 3

To Run a communication test with simple echo server and client code on Mindstorms/nxt...................................... 4

Basic low level serial APIs from RobotC: .................................................................................................................. 4

Echo server code (receiver) on Mindstorms/NXT ........................................................................................................ 4

Echo client code (sender) on Mindstorms/NXT ........................................................................................................... 5

To Run a communication test with simple echo server and client code on Arduino .................................................... 6

Important hardware setup with the xbee shield: ........................................................................................................... 6

Test 1 - Basic Echo test using the Digi programmable board ...................................................................................... 7

Test 2: Write your own transmitter and receiver using software Serial ................................................................... 9

Reference: .................................................................................................................................................................. 9
XBee Communication

COMPUTATIONAL THINKING AND PRE-COLLEGE ENGINEERING

XBee Configuration

Hardware:

- 2+ XBee with XBee Shield for Arduino or Arduino Wireless XBee shield
- Your XBees must be on the same channel to communicate with one another. The operating frequency ranges within the 2.4GHz 802.15.4 band
- install Xtcu - use this to configure Xbee

Must configure the XBee modules to speak the same ID first

1. Need to download the Xtcu software.
2. Configure their unique PAN ID (Personal Area Network ID). Your XBees must share the same PAN ID to communicate with one another. Valid value: 0 to 0xFFFF.
3. There is a default Addressing range; which you should not need to change.
4. Snap one Xbee module onto the XBee shield hooked up with an Arduino
5. At the Xtcu software:
   a. Add a radio module - the Xbee module thru this:
   b. Select the proper com port and baud rate 9600.
   c. Setup unique channel and pan id thru:
   d. Working Modes - configuration working mode

6. Done with one xbee.
7. Repeat 2 to 4.d for the 2nd xbee module
TO RUN A COMMUNICATION TEST WITH SIMPLE ECHO SERVER AND CLIENT CODE ON MINDSTORMS/NXT

BASIC LOW LEVEL SERIAL APIs FROM ROBOTC:

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nxtEnableHSPort()</td>
<td>Enable High Speed Port #4</td>
</tr>
<tr>
<td>nxtSetHSBaudRate(9600)</td>
<td>Xbee Default Speed</td>
</tr>
<tr>
<td>nxtHS_Mode = hsRawMode;</td>
<td>Raw Mode</td>
</tr>
<tr>
<td>nxtGetAvailHSBytes()</td>
<td>any available byte to read</td>
</tr>
<tr>
<td>nxtReadRawHS(buf, N)</td>
<td>read N-bytes into the buffer array</td>
</tr>
<tr>
<td>nxtWriteRawHS(buf, N)</td>
<td>send N-bytes of outgoing data</td>
</tr>
<tr>
<td>nxtDisableHSPort()</td>
<td>Disable HS Port #4</td>
</tr>
</tbody>
</table>

ECHO SERVER CODE (RECIEVER) ON MINDSTORMS/NXT

```c

task main()
{
    char incomingData[10];  // may allow arbitrary size of buffer. Use only 1 byte here

    // initial setup -----------------------------------------------
    clearDebugStream();
    nxtEnableHSPort();  //Enable High Speed Port #4
    nxtSetHSBaudRate(9600);  //Xbee Default Speed
    nxtHS_Mode = hsRawMode;  //Set to Raw Mode (vs. Master/Slave Mode)

    // ---------------------------------------------------------------
    while (true) {  // listen for message and echo
        if (nxtGetAvailHSBytes() == 0)  // anything available to read?
            delay(1);
        continue;
    }
    if (nxtReadRawHS(incomingData, 1)) {
        writeDebugStreamLine("%d", incomingData[0]);
        delay(1);
    }
    nxtDisableHSPort();  // Disable HS Port #4
}
```
XBee Communication

ECHo CLIENT CODE (SENDER) ON MINDSTORMS/NXT

task main()
{
    char buf; // may allow arbitrary size of buffer. Use only 1 byte here
    // initial setup ----------------------------------
    … Do the same setup as in echo server code …
    // ---------------------------------------------
    while (true) {
        nxtWriteRawHS((char*)&buf,1);
        wait1Msec(1000);
        writeDebugStream("Sender: %c", ++buf);
    }
}
TO RUN A COMMUNICATION TEST WITH SIMPLE ECHO SERVER AND CLIENT CODE ON ARDUINO

For testing, you will need designate at least:

- One transmitter controller
- One receiver controller

IMPORTANT HARDWARE SETUP WITH THE XBEE SHIELD:

A) If you use the XBee Shield from Itead Studio, you need to make sure your jumpers are set properly – see the specification document. If you use the Arduino Wireless SD Shield module, Make sure you have toggle from usb to micro connection on the xbee shield.

Switch to Mico mode to allow Xbee communication

For regular TX/RX communication. So, when you download the code, you must be in this mode.

B) If you use the Arduino Xbee Shield:

I use

- pin 5 as RX pin
- Pin 4 as TX pin

Since I ran these test using 5V on Arduino, I disabled the 3.3V connection.

Or

Just remove these jumpers entirely
TEST 1 - BASIC ECHO TEST USING THE DIGI PROGRAMMABLE BOARD

A) Bare Basic Echo Server

//Bare Basic Echo Server code : wait for incoming data and echo it back
int snd = 0;

void setup() {
    Serial.begin(9600);
    Serial.println("Echo-Starting....");
}

void loop() {
    if (Serial.available() > 0)
        Serial.write(Serial.read());
}

B) Echo Receiver

B.1) With the Digi Programmable board

1. I use theDigi International XBIB-U-DEV as the transmitter. Thus I don’t even need to write the ode for this echo test because it will be like simple typing right at your computer console. To do this:

2. At Xtcu, go to Working Modes – console working mode. Make sure it is connected

Then, simply hitting keys at the console, you should see each being echoed back.
B.2) Instead of the programmable board, using Another Arduino with an already configured XBee module

```c
int LED = 13;
int snd = 0;

void setup() {
    pinMode(LED, OUTPUT);
    Serial.begin(9600);
    Serial.println("RX-Starting....");
}

void loop() {
    snd = (++snd)%5;
    Serial.write(snd); delay(10);
    if (Serial.available() > 0) {
        int ch = Serial.read();
        for (int i=0; i<ch; i++) {
            digitalWrite(LED, HIGH);   // turn the LED on
            delay(500);
        }
        delay(2000);
    }
}
```
**TEST 2: WRITE YOUR OWN TRANSMITTER AND RECEIVER USING SOFTWARE SERIAL**

- **XBee Sender**
  - Send 1 byte via software serial
  - Get the byte back from echo server
  - Display the data to the hardware serial monitor.

- **XBee Echo Server**
  - Receive 1 byte via Hardware Serial
  - Send it back via Hardware Serial

---

```c
// Sender
#include <SoftwareSerial.h>

int irx = 5, itx = 4;
SoftwareSerial Ss( irx, itx);

void setup() {
    Ss.begin(9600);
    Serial.begin(9600);
    Serial.println("TX-Starting....");
}

void loop() {
    for (int i=0; i<20; i++)
    Ss.write(i);
    if ( Ss.available() > 0 )
    {
        Serial.print( Ss.read());
        Serial.print(" ");
    }
    delay(1000);
}
```

```c
//Echo Server Bare Basic Sample
void setup() {
    Serial.begin(9600);
    Serial.println("Echo-Starting....");
}
void loop() {
    if ( Serial.available() > 0)
    Serial.write( Serial.read()+1);
}
```
Learn about the AT commands you may use with XBee:

https://dlnmh9ip6v2uc.cloudfront.net/learn/materials/29/22AT%20Commands.pdf

More development:

- XBIB-U-DEV from Digi - about $70 configuration board. This is one of the RF Evaluation and Development Kits Board. View the datasheet here.
- Two XBee Series-1 modules

Another option: Purchase NXTBee

1) Download the software - XCTU Next Gen Installer, Windows x32/x64
2) Upload the firmware using this software