

IUEE Project 4

Testing out the XBee module

Arduino Wireless Shield with the XBee 802.15.4 modules

The Arduino Wireless shield allows your Arduino board to communicate wirelessly using Zigbee. This documentation describes the use of the shield with the XBee 802.15.4 module (sometimes called "Series 1" to distinguish them from the Series 2 modules, although "Series 1" doesn't appear in the official name or product description).

A Simple Example

You should be able to get two Arduino boards with Wireless shields talking to each other without any configuration, using just the standard Arduino serial commands (described in the [reference](#)).

To upload a sketch to an Arduino board with a Wireless shield, remove the Xbee. Then, you can upload a sketch normally from the Arduino environment. In this case, upload the **Communication | Physical Pixel** sketch to one of the boards. This sketch instructs the board to turn on the LED attached to pin 13 whenever it receives an 'H' over its serial connection, and turn the LED off when it gets an 'L'. You can test it by connecting to the board with the Arduino serial monitor (be sure it's set at 9600 baud), typing an H, and pressing enter (or clicking send). The LED should turn on. Send an L and the LED should turn off. If nothing happens, you may have an Arduino board that doesn't have a built-in LED on pin 13 (see the [board index](#) to check for sure), in this case you'll need to supply your own.

Once you've uploaded the Physical Pixel sketch and made sure that it's working, unplug the first Arduino board from the computer. Change the switch to the Micro setting. Now, you need to upload a sketch to the other board. Make sure its switch is in the USB setting. Then upload the following sketch to the board:

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

void loop( )
{
  Serial.print( 'H' );
  delay(1000);
  Serial.print( 'L' );
  delay(1000);
}
```

[\[Get Code\]](#)

When it's finished uploading, you can check that it's working with the Arduino serial monitor. You should see H's and L's arriving one a second. Turn off the serial monitor and unplug the board. Change the switch to the Micro setting. Now connect both boards

Intel Ultimate Engineering Experience (IUEE)

to power. After a few seconds, you should see the LED on the first board turn on and off, once a second. (This is the LED on the Arduino board itself, not the one on the Xbee shield, which conveys information about the state of the Xbee module.) If so, congratulations, your Arduino boards are communicating wirelessly.

A Few Notes

You can use any of the standard Arduino serial commands with the Xbee shield. With the switch in the Micro position, the print and println commands will send data over the Xbee shield and the USB connection (i.e. to other Xbee shields and to the computer at the same time). In this configuration, however, the board will only receive data from the Xbee shield not from the USB connection.

The Xbee module on the shield is set up to work at 9600 baud by default, so unless you reconfigure it, you'll need to make sure you're passing 9600 to the Serial.begin() command in your sketch.

To allow your computer to communicate directly with the Xbee shield, connect it to an Arduino board whose microcontroller has been removed and place the switch in the USB configuration. Then you can send data to and receive data from the Xbee module from any terminal program. This allows you, for example, to see the data that the module is receiving from other Xbee shields (e.g. to collect sensor data wirelessly from a number of locations).

Configuring the Xbee Module

You can configure the Xbee module from code running on the Arduino board or from software on the computer. To configure it from the Arduino board, you'll need to have the switch in the Micro position. To configure it from the computer, you'll need to have the switch in the USB position and have removed the microcontroller from your Arduino board.

To get the module into configuration mode, you need to send it three plus signs: +++ and there needs to be at least one second before and after during which you send no other character to the module. Note that this includes newlines or carriage return characters. Thus, if you're trying to configure the module from the computer, you need to make sure your terminal software is configured to send characters as you type them, without waiting for you to press enter. Otherwise, it will send the plus signs immediately followed by a newline (i.e. you won't get the needed one second delay after the +++). If you successfully enter configuration mode, the module will send back the two characters 'OK', followed by a carriage return.

Send Command	Expected Response
+++	OK<CR>

Once in configuration mode, you can send AT commands to the module. Command strings have the form ATxx (where xx is the name of a setting). To read the current value of the setting, send the command string followed by a carriage return. To write a new value to the setting, send the command string, immediately followed by the new setting (with no spaces or newlines in-between), followed by a carriage return. For example, to read the network ID of the module (which determines which other Xbee modules it will communicate with), use the 'ATID' command:

Intel Ultimate Engineering Experience (IUEE)

Send Command	Expected Response
--------------	-------------------

ATID<enter>	3332<CR>
-------------	----------

To change the network ID of the module:

Send Command	Expected Response
--------------	-------------------

ATID3331<enter>	OK<CR>
-----------------	--------

Now, check that the setting has taken effect:

Send Command	Expected Response
--------------	-------------------

ATID<enter>	3331<CR>
-------------	----------

Unless you tell the module to write the changes to non-volatile (long-term) memory, they will only be in effect until the module loses power. To save the changes permanently (until you explicitly modify them again), use the **ATWR** command:

Send Command	Expected Response
--------------	-------------------

ATWR<enter>	OK<CR>
-------------	--------

To reset the module to the factory settings, use the **ATRE** command:

Send Command	Expected Response
--------------	-------------------

ATRE<enter>	OK<CR>
-------------	--------

Note that like the other commands, the reset will not be permanent unless you follow it with the **ATWR** command.