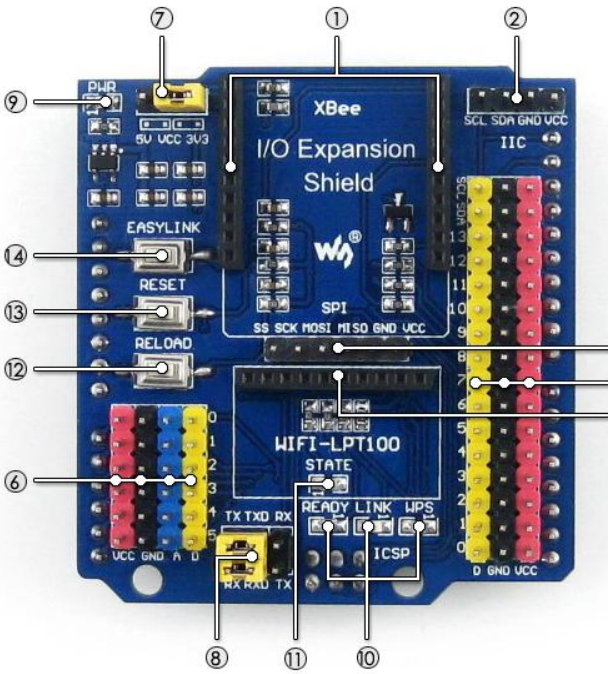
IO Expansion Shield User Manual

1 Features

- 3-pin & 4-pin sensor interfaces, supports connecting sensors directly without complicate custom connections
- XBee module connector
- WIFI-LPT100 wireless module connector

2 What's on Board

1. **XBee module connector**
2. **IIC interface**
3. **SPI interface**
4. **3-pin sensor interface**
   - VCC : power positive
   - GND : ground
   - D : digital pin, correspond to the Arduino board
5. **WIFI-LPT100 connector**
6. **4-pin sensor interface**
   - VCC : power positive
   - GND : ground
   - A : analog pin, correspond to the Arduino board
   - D : digital pin, correspond to the Arduino board
7. **VCC configuration :** for selecting sensor interfaces power voltage
8. **Debugging/Communication selection jumper**
   - when connecting TXD and TX, RXD and RX respectively, the Arduino board may debug/config the XBee module or WIFI-LPT100 through serial port
   - when connecting TXD and RX, RXD and TX respectively, the Arduino board may communicate with the XBee module or WIFI-LPT100 through serial port



9. **Power indicator**
10. **WIFI-LPT100 state indicator**
11. **XBee module state indicator**
12. **WIFI-LPT100 RELOAD button :** for restoring to factory setting
13. **XBee module and WIFI-LPT100 RESET button**
14. **XBee module EASYLINK button**

3 Getting started with IO Expansion Shield

3.1 Preparations

- One WIFI-LPT100 module
- Two Arduino development boards
- Two XBee modules
- Two IO Expansion Shields
- TCP232 serial software

3.2 How to use with WIFI-LPT100 module

WIFI-LPT100 is a Wi-Fi wireless communication module developed by Waveshare. It adopts LPT100 wireless technology. This module features with Wi-Fi to serial conversion function, making it possible to be operated as a serial port and providing good user experience in MCU communication.

1) Insert the WIFI-LPT100 module into the WIFI-LPT100 interface of IO Expansion Shield, and set the jumpers of IO Expansion Shield:
   - Connect TXD toTX;
   - Connect RXD to RX.
2) Connect IO Expansion Shield to Arduino development board.
3) Power up Arduino development board, and connect it to a PC via a USB cable. Waiting a few seconds, the READY LED indicator will light up.
4) Open Wi-Fi on your PC, and check the wireless network status (Usually, there will be a symbol in stair-step shape appears on the right of the taskbar. Clicking this symbol, you will see the available Wi-Fi connection(s)). Select WIFI-LPT100 and Connect.

5) Start the software TCP232, and select corresponding COM port. Then, set the Baud rate: 115200, and click the button **OPEN**. Configure the parameters in option NetSettings :
   - Protocol->TCP Client
   - Server IP->10.10.100.254
   - Server Port->8899

   And click the button **Connect**.
6) Enter a message in the data transmission box on the button left, and click the button **Send** nearby. You can find that the message will be received by the network data receive box. Similarly, when you input a message in the data transmission box on the button right, and click the button **Send** nearby, you can see that the message you inputted will be received by the COM port data receive

box.

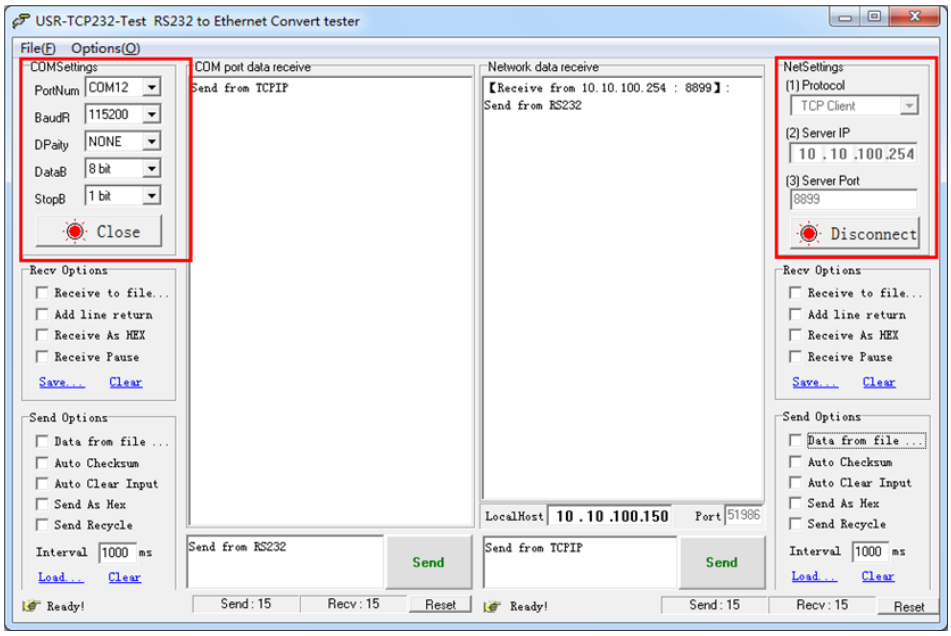


Figure 1: Setting TCP232

7) If you need to use the Arduino development board to deal with the data received by WIFI-LPT100, you should set the jumpers of IO Expansion Shield:
   - Connect TXD to RX;
   - Connect RXD to TX.

   Now, you can transmit data between WIFI-LPT100 and the MCU of Arduino development board.

3.3 How to use with XBee module

XBee of MaxStream is a wireless communication module based on ZigBee technology. In an easy-to-use design, it can automatically transmit the inputted data to another XBee module by wireless connection. And it also supports AT commands for advance configuration.

1) Prepare two XBee modules, two IO Expansion Shields and two Arduino development boards. In this document, we will divide them into two groups: Group A and Group B, of which Group A includes XBee-A, IP Expansion Shield-A and Arduino development board-A, and Group B includes XBee-B, IP Expansion Shield-B and Arduino development board-B.
2) Insert the XBee-A into the XBee interfaces of IO Expansion Shield-A, and insert the XBee-B into the XBee interfaces of IO Expansion Shield-B respectively.
3) Connect the IO Expansion Shield-A to the Arduino development board-A, and connect the IO Expansion Shield-B to the Arduino development board-B respectively. And power up the Arduino development boards, and connect them to your PC via USB cables.
4) Set IO Expansion Shield jumpers:

- Connect TXD to TX
- Connect RXD to RX

Notices: The USB interface, TX and RX pins of Arduino development board should be used in the processes described below, so please make sure that the projects running on the Arduino development board will not occupy any serial port. If necessary, you can remove the AVR chip from Arduino development board before performing the following operations.

5) Start X-CTU software to configure XBee module on your PC.

a) Select the corresponding COM port in PC Settings bar, and set relative parameters, such as baud rate. The factory default settings of XBee module is as followed:

- Baud rate: 9600
- Data Bits: 8
- Parity: NONE
- Stop Bits: 1

(To make sure XBee module can operate properly, the relative settings of it should be configured correctly. You can configure the XBee module by X-CTU tool, or by AT commands via a serial terminal. This document will introduce how to use X-CUT tool to configure XBee merely.)

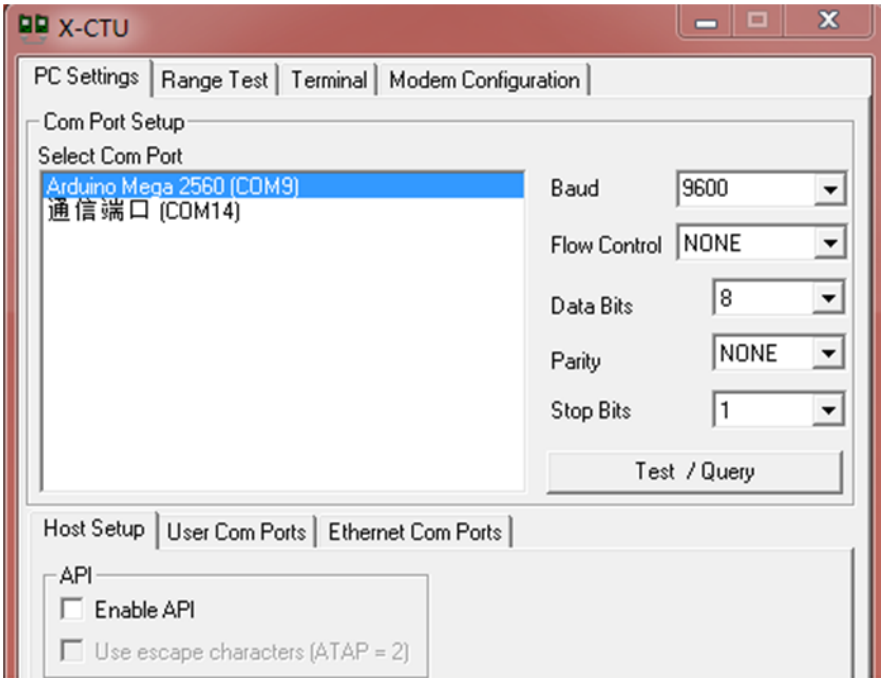


Figure 2: X-CTU operation interface

b) Click the button **Test/Query** to test whether the software is connected to the XBee module properly. Normally, we will see the dialog box popped out, as Figure 3 shows. If it cannot communicate with the XBee module but all the communication parameters are set correctly, please check the USB connection and the jumper settings on the Arduino development board and the XBee expansion board. (The main control chip of Arduino UNO R3 is a DIP

AVR MCU. You can remove it to retry, if necessary.)

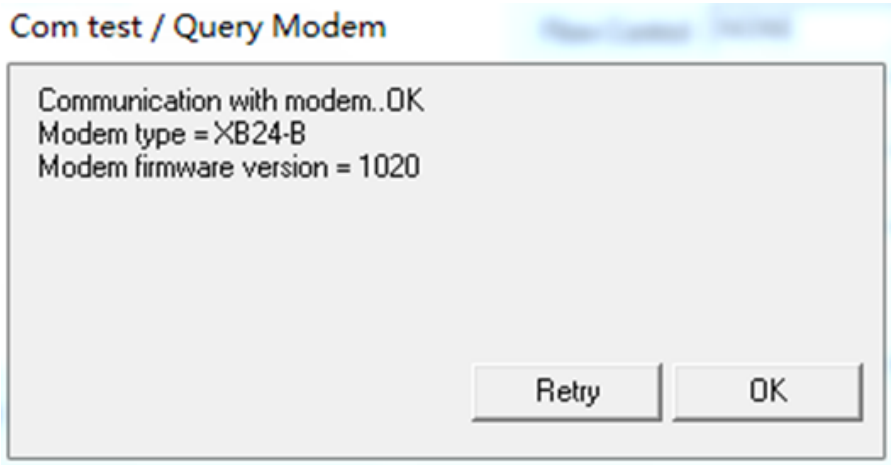


Figure 3: Testing the connection to the XBee module

c) After a successful test, click the option Modem Configuration.

d) Then, click the button **Read** to get the current parameters of XBee-A.

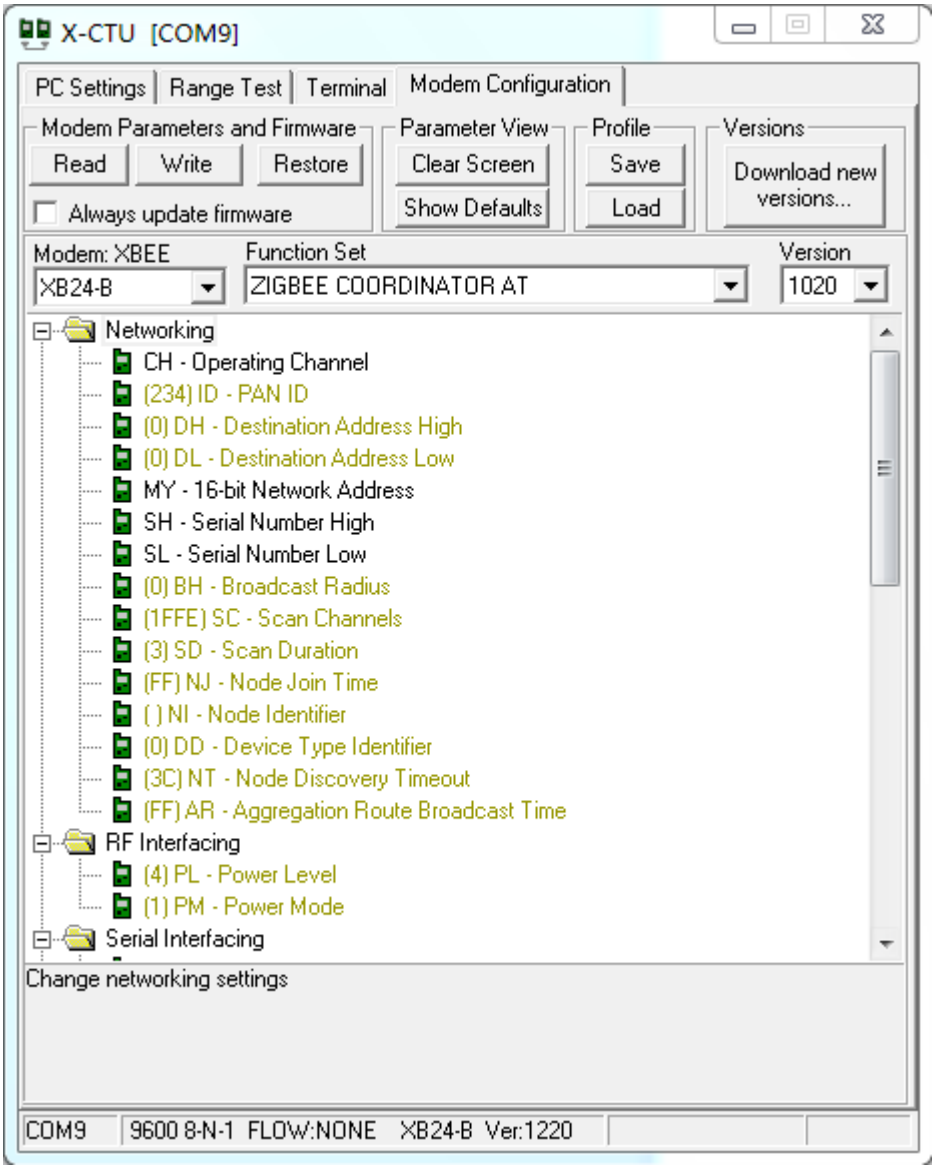


Figure 4: Getting the current parameters of XBee-A

e) Select the option ZIBGEE ROUTER/END DEVICE AT under the pull-down menu Function Set of the XBee-A.

Figure 5: Setting the Function Set of XBee-A

f) Set the read Networking settings: ID: 234, DH: 0, and DL: 0.

g) Click the button **Write** to download the configured parameters into the XBee-A.

h) Select the option ZIBGEE ROUTER/END DEVICE AT under the pull-down menu Function Set of the XBee- B.

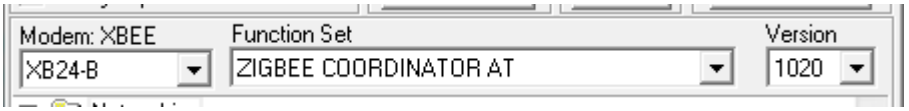


Figure 6: Setting the Function Set of XBee-B

i) Set the read Networking settings: ID: 234, DH: 0, and DL: FFFF.

j) Click the button **Write** to download the configured parameters into the XBee-B.

6) In order to implement a simple P2P network, please configure XBee-A and XBee-B according to the processes described above. Start two X-CTU tools, and select different COM interfaces in the option PC Settings to control Group A and Group B respectively.

7) Input the data to be transmitted in the X-CTU Terminal of XBee-A, then, you can find that the inputted data will be sent to XBee-B automatically, and displayed in the X-CTU Terminal of XBee-B. In the X-CTU, data in blue is the data to be sent, and data in red is the received data.

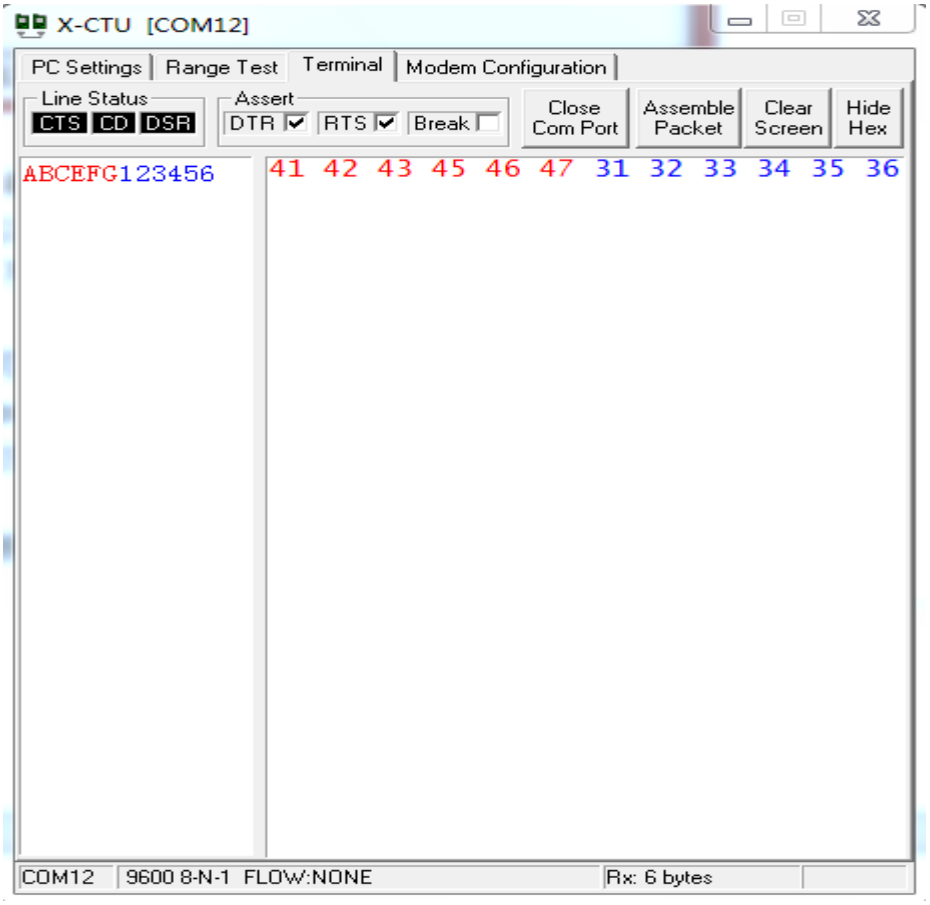


Figure 7: The X-CTU terminal of XBee-A

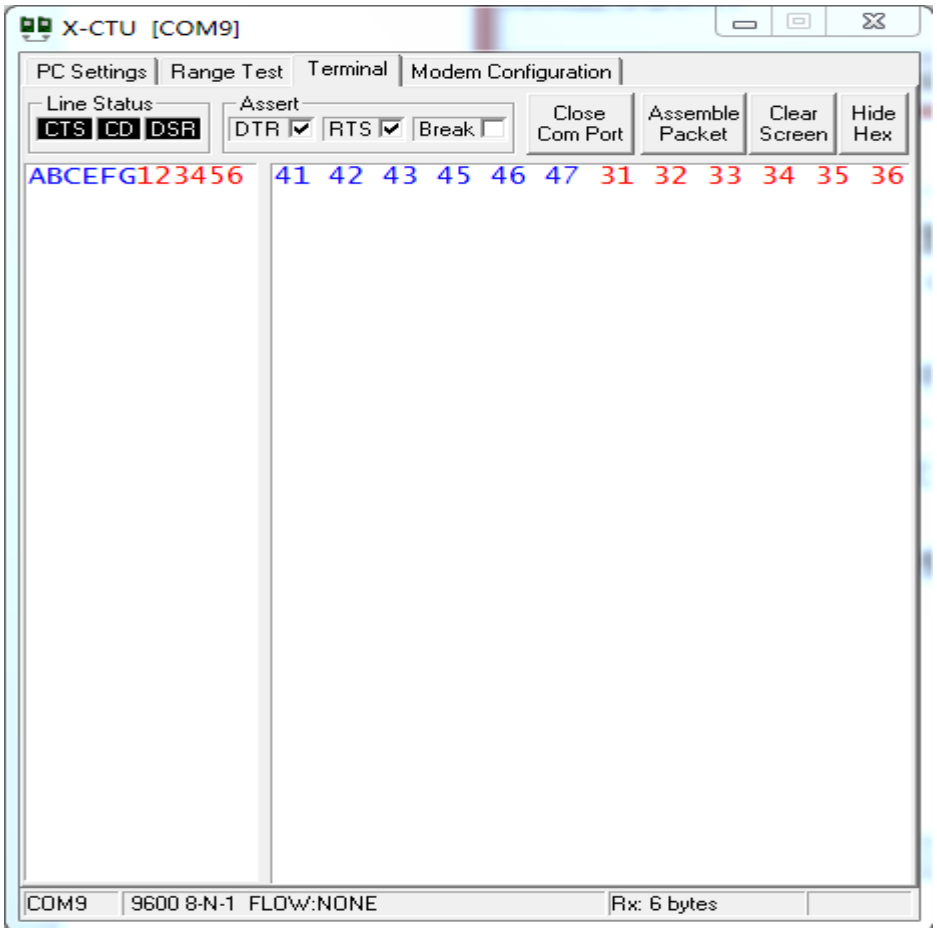


Figure 8: The X-CTU terminal of XBee-B

8) Figure 8 shows the normal operating state of XBee module. You can use XBee modules to perform wireless communication in Arduino projects. It is easy to done, transmitting the data to be sent to XBee-A via the serial port of Arduino development board-A, then, receiving the data by XBee-B and reading it out on PC via the serial port of Arduino development board-B.

Notices: In this operation, you should set the jumpers of IO Expansion Shield:

- **Connect TXD to RX;**
- **Connect RXD to TX.**

Then, XBee module can communicate with the MCU of Arduino development board via a serial port.