



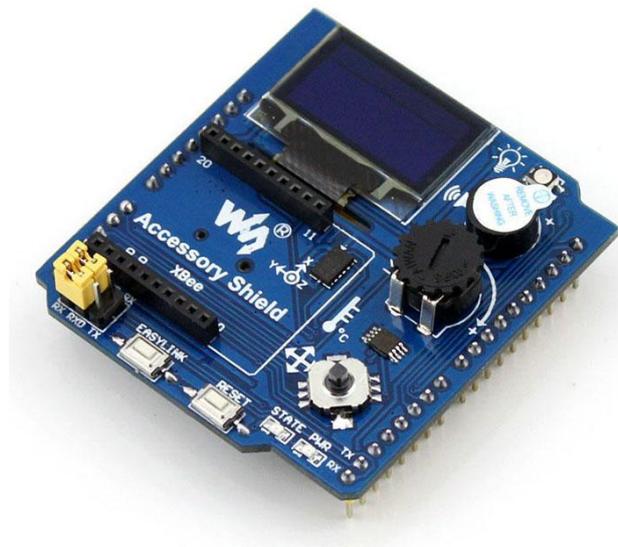
# Accessory Shield User Manual

## Overview

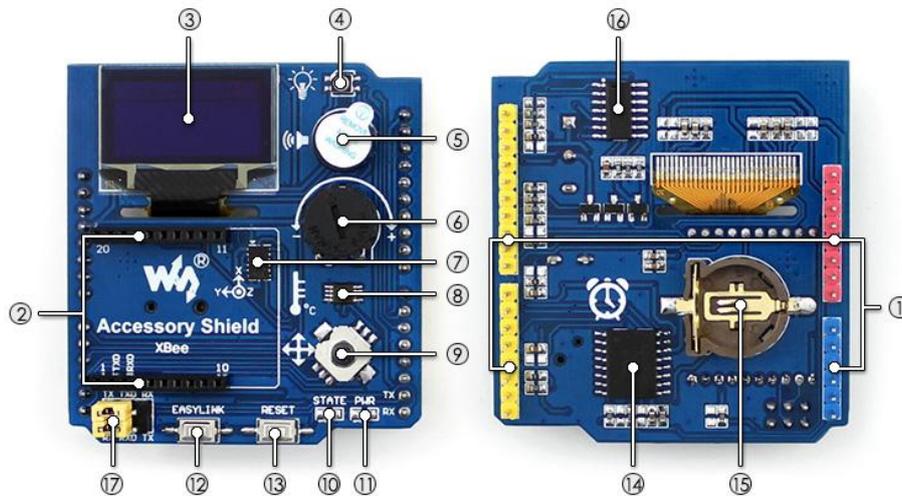
Accessory Shield is an Arduino expansion board developed by Waveshare. It integrates many popular peripheral devices, providing a great support to users in learning Arduino series products and shortening development period.

### Features

- 5-way joystick
- Adjustable potentiometer
- Buzzer
- RGB LED
- Temperature Sensor
- 3-axis accelerometer
- Extremely accurate real-time clock
- 0.96 inch OLED
- XBee interface



## 1. On-board source



### [Interface]

1. **Arduino interface**  
Easy to connect to different main control boards with Arduino interface.
2. **XBee interface**  
Easy to connect to different XBee communication modules.

### [Device]

3. **0.96inch OLED**  
Flexible for various display experiments; resolution: 128×64.
4. **RGB LED**  
RBG LED with 256×256×256 color display.
5. **Buzzer**  
PWM control function, supports simple music performance.
6. **10K adjustable potentiometer**
7. **ADXL345**  
3-axis accelerometer, measurement range: ±16g.
8. **LM75BDP**  
Temperature sensor with 0.125 °C accuracy.

### 9. 5-way joystick

5 action statuses: UP, DOWN, LEFT, RIGHT and ENTER.

### 10. XBee status indicator

### 11. Power light

### 12. XBee EASYLINK press key

### 13. Arduino and XBee reset key

### 14. DS3231

Extremely accurate real-time clock.

### 15. CR1220 battery socket

Power supply the real-time clock.

### 16. P9813

RGB LED driver chip

### [Jumper]

### 17. Jumper configuration for debugging/communication

Connecting TXD to TX, and RXD to RX, the XBee module can be debugged and configured by the relative software installed on the PC via the serial port on the Arduino main control board; Connecting TXD to RX, and RXD to TX, the XBee module can communicate with the Arduino main control board via the serial port.

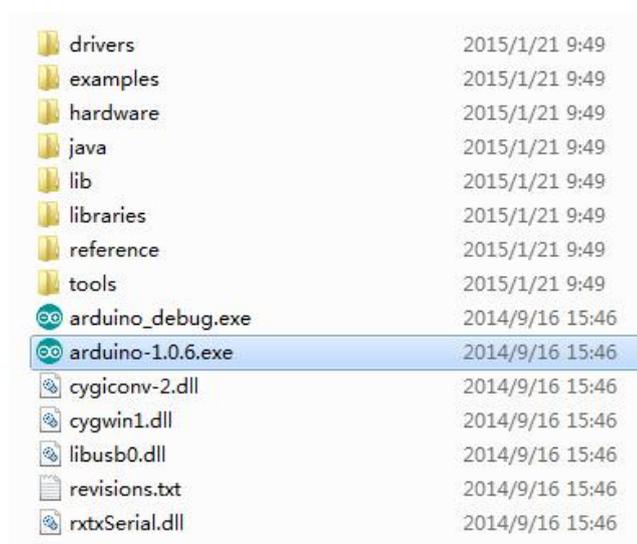
## 2. How to use

In this chapter, we will introduce the basic operations of Arduino IDE in the section 2.1, including custom library importation, program compiling and uploading. The detailed operations and experiments of the devices on the board are presented in the section 2.2. The development board we used here is Arduino UNO PLUS. For more detailed information about this development board, please refer to Appendix.

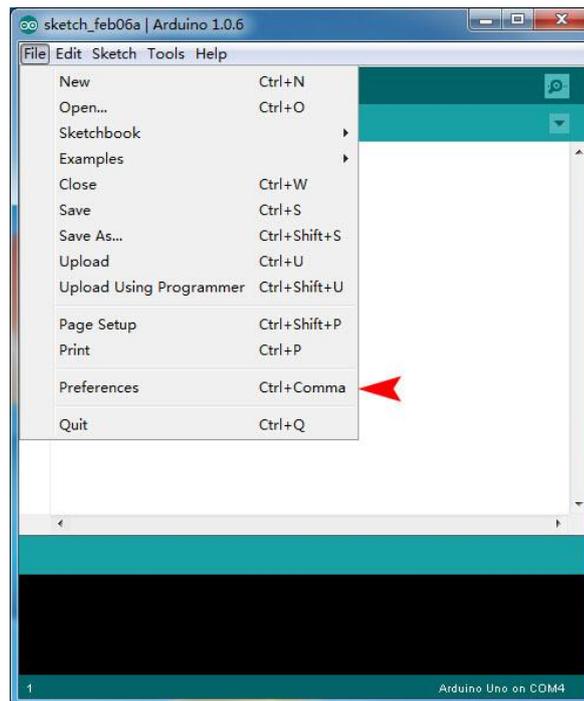
### 2.1. Arduino IDE

#### 2.1.1. Library importation

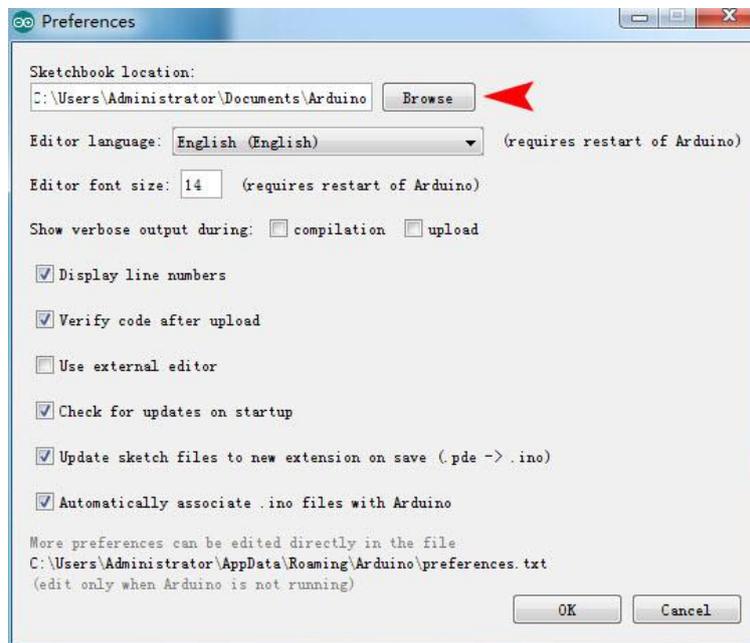
Here we take the Arduino 1.0.6 (free installation) as an example to illustrate how to import libraries. Firstly, start the software arduino-1.0.6.exe.



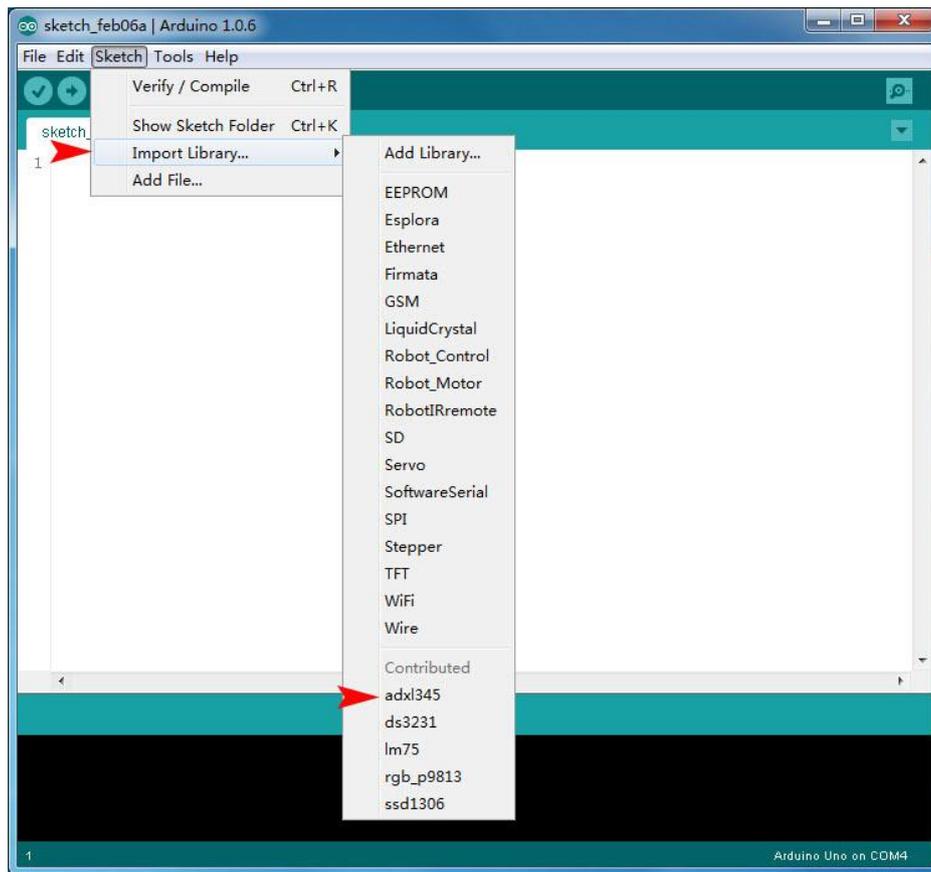
Then, click File -> Preferences, to enter the configuration interface.



Find out the folder *libraries*, and copy it to the folder pointed by the option **Sketchbook location**:

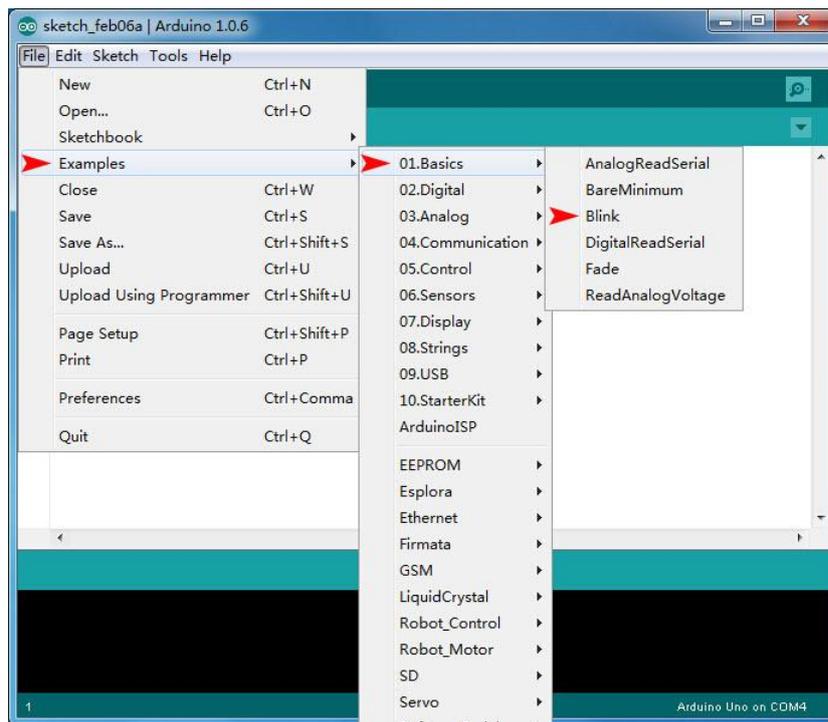


Restart the software, and then click Sketch -> Import Library, to check the library options: **adxl345**, **ds3231**, **lm75**, **rgb\_p9813** and **ssd1306**. If all of them are available, the libraries are imported successfully.

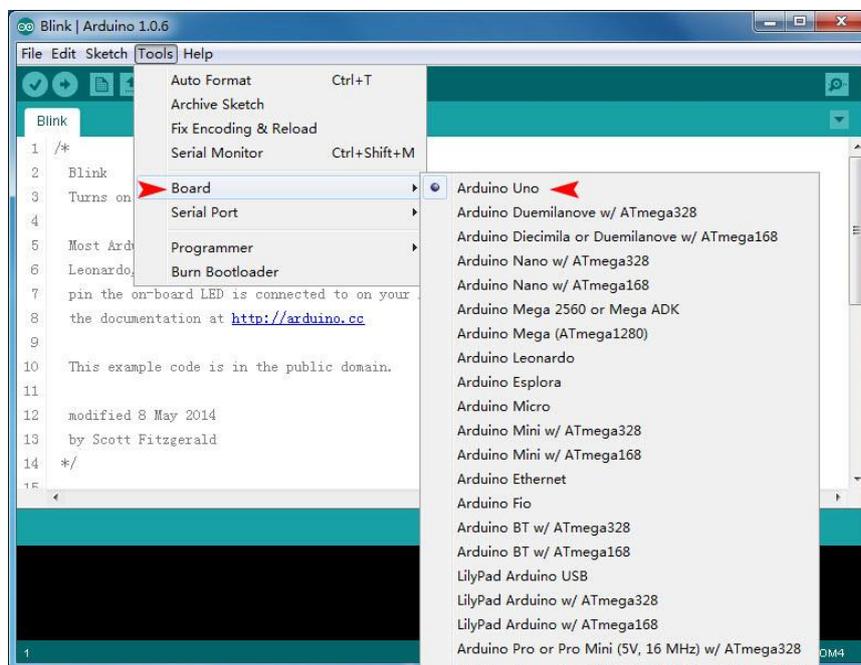


### 2.1.2. Program compiling and uploading

- 1) Switch the BOOT mode of UNO PLUS development board to ON, and set VCC to 5V;
- 2) Connect the Accessory Shield module to the UNO PLUS development board, and then the development board to your PC via a USB cable. You can see the power light on the module lights up when the module is working properly.
- 3) This document presents the method of program compiling and uploading with the Demo provided by Arduino IDE. Click File -> Example -> 01.Basics -> Blink, to open the Demo.

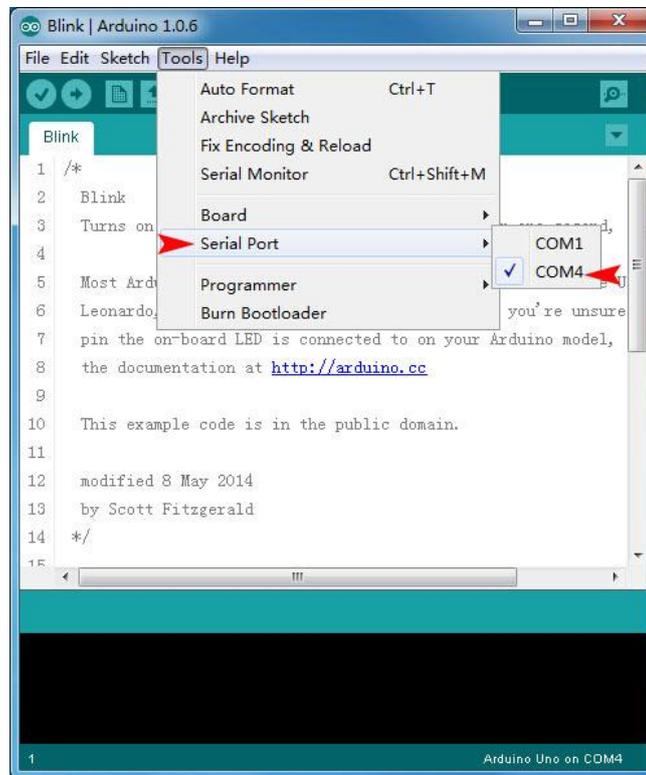


- 4) Click Tools -> Board -> Arduino Uno, to select relative Arduino development board. Since the setting will be saved automatically, you only need to configure it once.

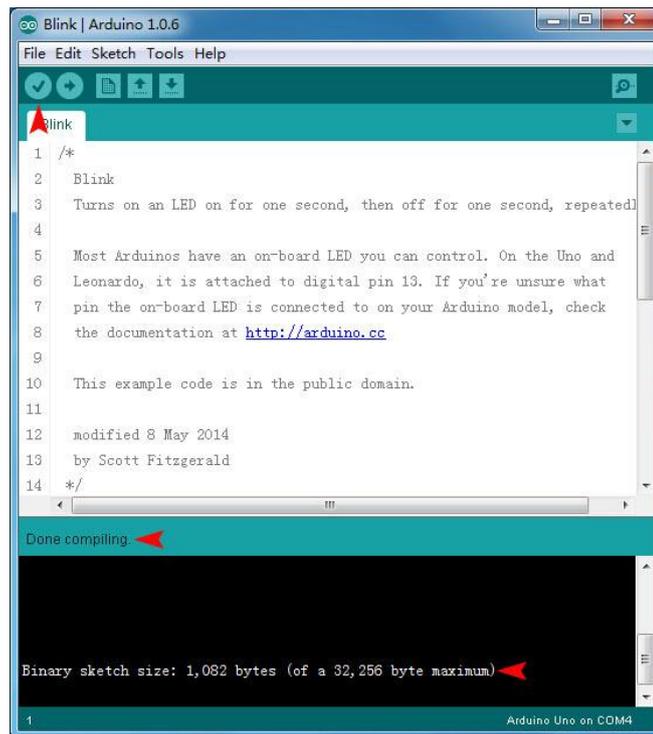


- 5) Click Tools -> Serial Port -> COM4, to configure the serial port of the Arduino development board for uploading. Please take a note that the port number in used is

various with different PCs, so you need to check the actual serial port number by Device Manager before making the setting. In here, the serial port number is COM4.



- 6) Click the button  to start program compiling. If there is no error, the software will give a prompt “Done compiling” and display the size of generated file.



- 7) Click the button  to start uploading. When finished, you can see the LED on the UNO PLUS development board blinking in a frequency of 1Hz if the program runs properly.

## 2.2. Basic experiments

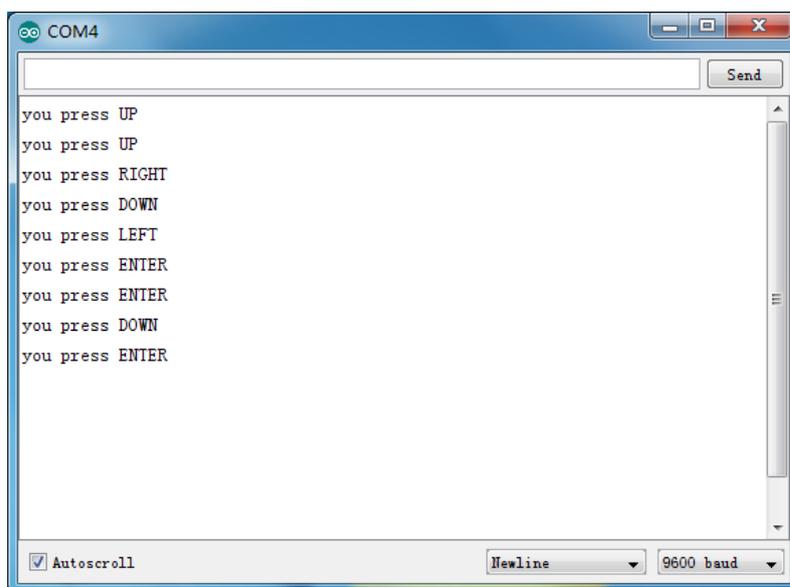
Please make sure the libraries in the Demo are imported correctly, before performing the basic experiments presented in the sections below.

### 2.2.1. 5-way joystick

The Joystick has 5 action statuses: UP, DOWN, LEFT, RIGHT and ENTER. The keys relative to different statuses are connected to corresponding Arduino pins. The key-press can be judged by checking the corresponding IO levels.

#### Operation and phenomena:

Perform program compiling and uploading, and then click the button  to start the serial port monitor of Arduino IDE. If the program is running properly, the on-board LED will blink in the frequency of 1Hz, indicating that the whole system works well. When any key is pressed, the monitor will show the one has been pressed, for example: *you press UP*, as the figure shows below.



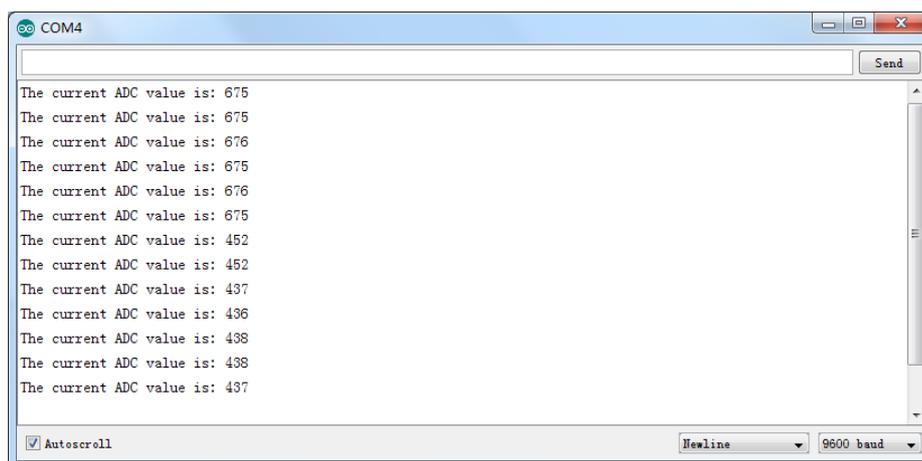
### 2.2.2. ADJUSTABLE potentiometer

ATMEGA328P-AU is the main control chip of UNO PLUS development board. It has a wide power voltage range, supporting 3.3V and 5V power supply. With a 10-bit AD converter embedded, it can provide  $2^{10}=1024$  levels of AD conversion. For 3.3V power supply, the resolution is up to  $3300\text{mv} / 1024 = 3.22\text{mv}$ . When the AD value is 1, it corresponds to the output voltage 3.22mv. For 5V power supply, the resolution is  $5000\text{mv} / 1024 = 4.88\text{mv}$ . When the AD value is 1, it corresponds to the output voltage 4.88mv.

Accessory Shield provides an on-board 10k adjustable potentiometer with the maximum adjustable output voltage of 3.3V. When the development board is powered supply by 3.3V, the ADC output value is in the range of 0 – 1023. For 5V power supply, the ADC resolution is 4.88mv, and the AD value corresponding to 3.3V, the maximum output voltage, is  $3300 / 4.88 = 676$ , so the ADC output value is in the range of 0 – 676 in this case.

**Operation and phenomena:**

Perform program compiling and uploading, and then click the button  to start the serial port monitor of Arduino IDE. If the program is running properly, the monitor will display the current ADC value. When the potentiometer is adjusted, the outputted sample value will change too, as the figure shows below.



### 2.2.3. Buzzer

In applications, active buzzer can be directly connected to the rated power supply to work properly. But passive buzzer, as well as electromagnetic loudspeaker, requires an audio output circuit to work with. The buzzer we use here is an active buzzer, which can either perform simple frequency sound with DC power source, or play simple music with AC power source.

**Operation and phenomena:**

Perform program compiling and uploading to the board. If the program is running well, you can hear the buzzer play some simple music.

### 2.2.4. RGB LED

RGB LED use PWM function to control the brightness generated respectively by three LED chips (R, G and B) embedded in the LED light. And the mixture of RGB can display rich colors.

### Operation and phenomena:

Perform program compiling and uploading to the board. If the program is running well, you can see the RGB LED displays gradient colors.

#### 2.2.5. Temperature sensor: LM75B

The LM75B is a temperature-to-digital converter with I<sup>2</sup>C interface. In a simple application design, it only requires a decoupling capacitance to work with. And there are three selectable logic address pins so that eight devices can be connected on the same bus without address conflict.

It has the following features:

- Pin-for-pin replacement for industry standard LM75 and LM75A;
- I<sup>2</sup>C-bus interface with up to 8 devices on the same bus, frequency range from 20 Hz to 400 kHz with bus fault time-out to prevent hanging up the bus;
- Power supply range from 2.8 V to 5.5 V;
- Temperatures range from -55°C to 125°C;
- 11-bit ADC that offers a temperature resolution of 0.125°C;
- Temperature accuracy of:
  - ±2°C from -25°C to +100°C;
  - ±3°C from -55°C to 125°C;
- Supply current of 1.0µA in shutdown mode for power conservation;
- ESD protection.

When LM75 servers as a slave, a 7-bit slave address is used.

1= HIGH, 0= LOW

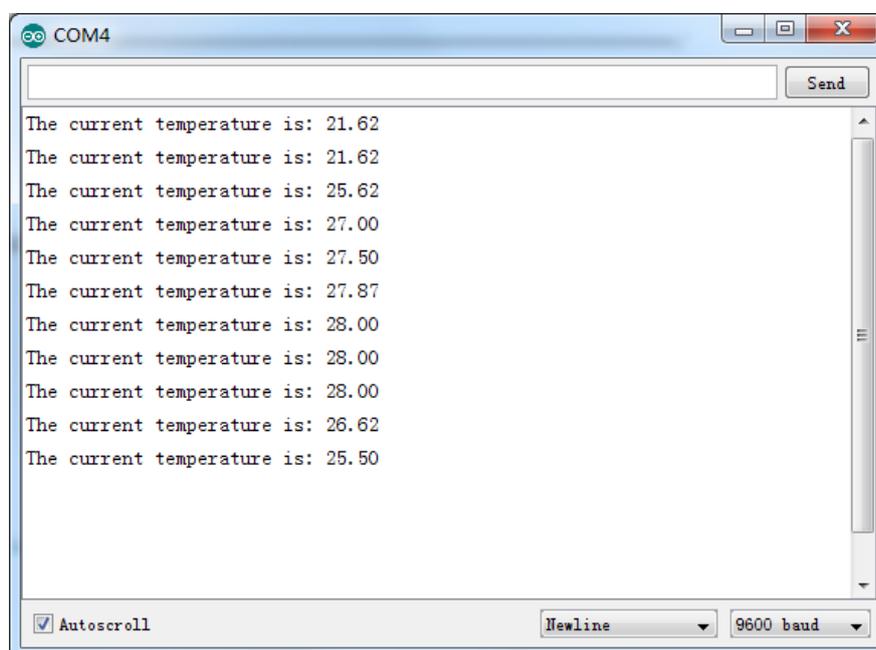
MSB				LSB		
1	0	0	1	A2	A1	A0

The pins A2/A1/A0 of this module are tied to Ground, so the slave address is 0x48 in here.

### Operation and phenomena:

Perform program compiling and uploading, and then click the button  to start the serial port monitor of Arduino IDE. If the program is running properly, the monitor will display the

current temperature value. Touching LM75 with your hand, you can see that the outputted sample values change too, as the figure shows below.



### 2.2.6. 3-axis accelerometer: ADXL345

The ADXL345 of ADI is a small, thin, ultralow power, 3-axis accelerometer with high resolution (13-bit) measurement at up to  $\pm 16 g$ . Digital output data is formatted as 16-bit twos complement and is accessible through either a SPI (3- or 4-wire) or I2C digital interface.

The ADXL345 is well suited for mobile device applications. It measures the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion or shock. Its high resolution (3.9 mg/LSB) enables measurement of inclination changes less than  $1.0^\circ$ .

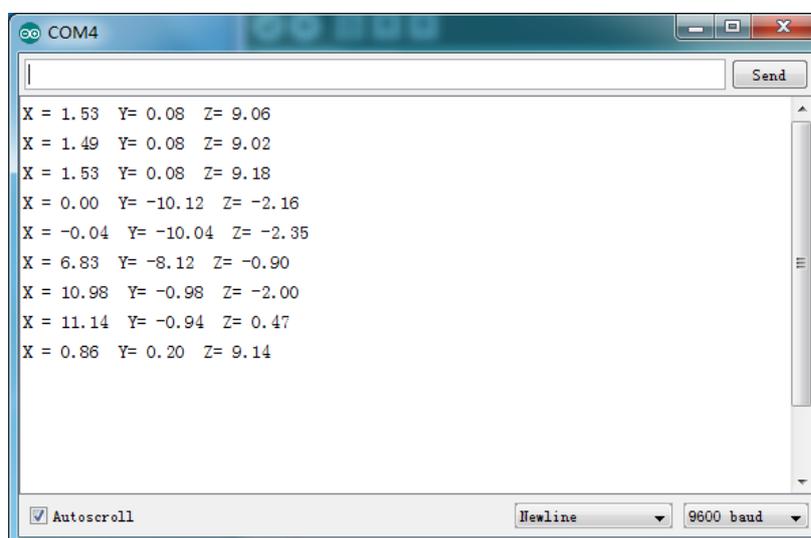
It has the following features:

- Power supply voltage range: 2.0V to 3.6V;
- I/O voltage range: 1.7V to VCC;
- Temperature range:  $-40^\circ\text{C}$  to  $+85^\circ\text{C}$ ;
- Dimensions: 3mm x 5mm x 1mm, LGA package.

#### Operation and phenomena:

Perform program compiling and uploading, and then click the button  to start the serial

port monitor of Arduino IDE. If the program is running properly, the monitor will display the current accelerated velocity of the development board based on the 3-axis space position. When you rotate the development board, the outputted sample value will change too, as the figure shows below.



### 2.2.7. Extremely accurate real-time clock: DS3231

The DS3231 is a low-cost, extremely accurate  $I^2C$  real-time clock (RTC) with an integrated temperature compensated crystal oscillator (TCXO) and crystal. The device incorporates a battery input, and maintains accurate timekeeping when main power to the device is interrupted. The integration of the crystal resonator enhances the long-term accuracy of the device.

The RTC maintains seconds, minutes, hours, day, date, month, and year information. The date at the end of the month is automatically adjusted for months with fewer than 31 days, including corrections for leap year.

It has the following features:

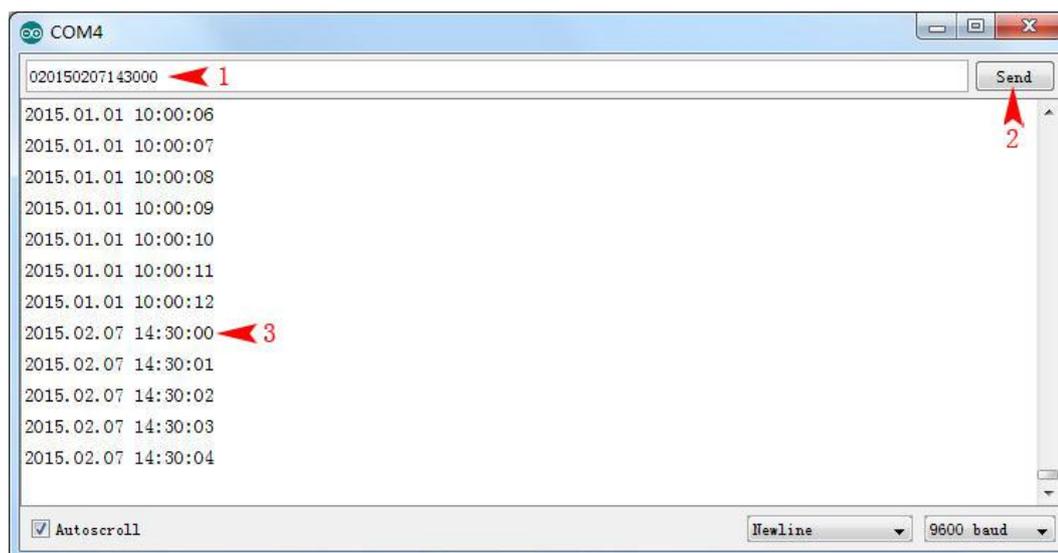
- Operating voltage range: 2.3V to 5.5V;
- Operating temperature range:  
Commercial:  $0^{\circ}\text{C}$  to  $+70^{\circ}\text{C}$ ;  
Industrial:  $-40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ ;
- Accuracy  $\pm 2\text{ppm}$  from  $0^{\circ}\text{C}$  to  $+40^{\circ}\text{C}$ ;
- Accuracy  $\pm 3.5\text{ppm}$  from  $-40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ .

### Operation and phenomena:

Perform program compiling and uploading, and then click the button  to start the serial port monitor of Arduino IDE. This Demo performs two functions: outputting current time information and setting new time via the serial port.

As the figure shows below:

Arrow 1 points to the position for inputting new time setting. The new time should be set in a specified format with 15 characters length, such as 020150207143000, in which the first character is 0, then the followings are the settings of year(2015), month(02), date(07), hours(14), minutes(30) and seconds(00). When finished the new time setting, click the button **Send** pointed by Arrow 2. You will see that the time information is updated to the new one (pointed by Arrow 3). **Notices that the time format should be set in 24-hour system and the whole data length is of 15 characters. For example, February is written into 02, and the like.**



### 2.2.8. 0.96 inch OLED: SSD1306

Accessory Shield is provided with a 0.96 inch 128 x 64 OLED display module. It has the following features:

- Driver chip: SSD1306;
- Interface: I<sup>2</sup>C;
- Color: yellow and blue;
- Active display area: >160°;

- Operating temperature range: -20°C to 70°C;
- Storage temperature range: -30°C to 80°C.

**Operation and phenomena:**

Perform program compiling and uploading to the board. The relative Demo shows the basic drawing operations, including lines, circles and rectangle boxes drawings.

---

**2.2.9. XBee interface**

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.

**Preparation:**

Prepare two XBee modules, two Accessory Shields and two Arduino UNO PLUS development boards. In this document, we will divide them into two groups: Group A and Group B, of which Group A includes XBee-A, Accessory Shield-A and Arduino development board-A, and Group B includes XBee-B, Accessory Shield-B and Arduino development board-B.

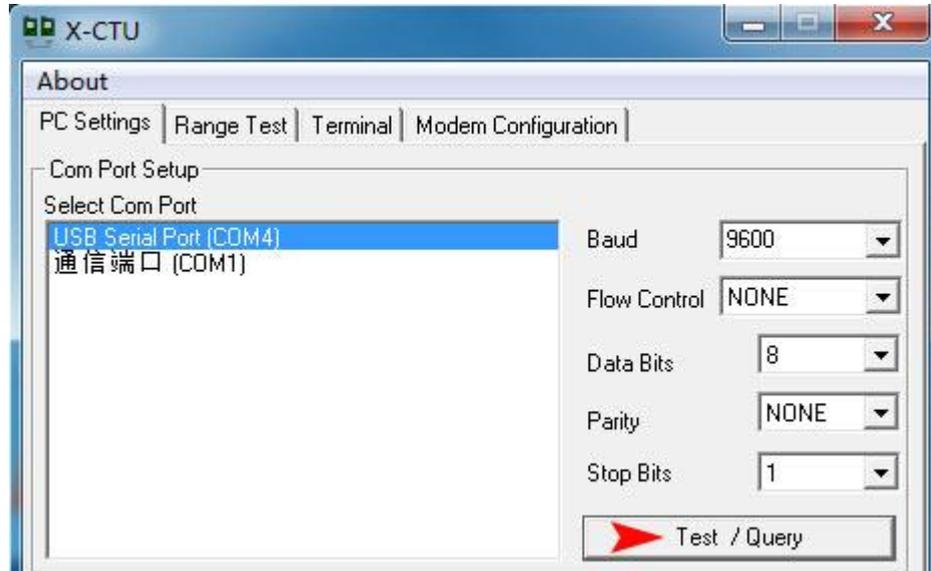
**Operation and phenomena:**

- 1) Insert the XBee-A into the XBee interfaces of Accessory Shield-A, and insert the XBee-B into the XBee interfaces of Accessory Shield-B respectively. Then, connect the Accessory Shield-A to the Arduino development board-A, and the Accessory Shield-B to the Arduino development board-B respectively. Power up the Arduino development boards, and connect them to your PC via USB cables.
- 2) Set Accessory Shield jumpers:
  - Connect TXD to TX;
  - Connect RXD to RX.

**Notices: The USB interface, TX and RX pins of Arduino development board will be used in the processes described below, so please make sure that the program running on the**

**Arduino development board does not occupy any serial port. If it does, please download and rewrite a new program, such as buzzer Demo, to the board.**

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



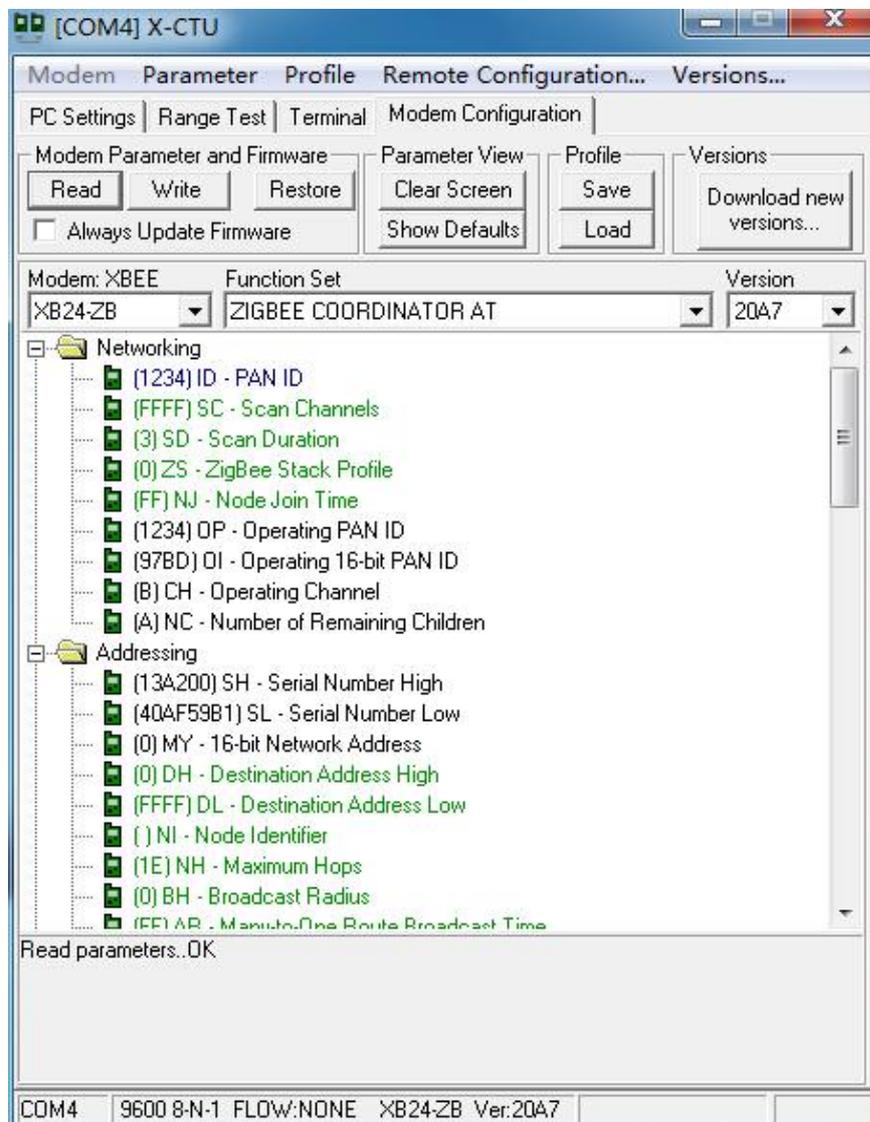
- 4) Select the corresponding COM port in PC Settings bar, and set relative parameters, such as baud rate. The factory default settings of XBee module is as followed:
- Baud: 9600;
  - Data Bits: 8;
  - Flow Control: NONE;
  - Parity: NONE;
  - Stop Bits: 1.

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.

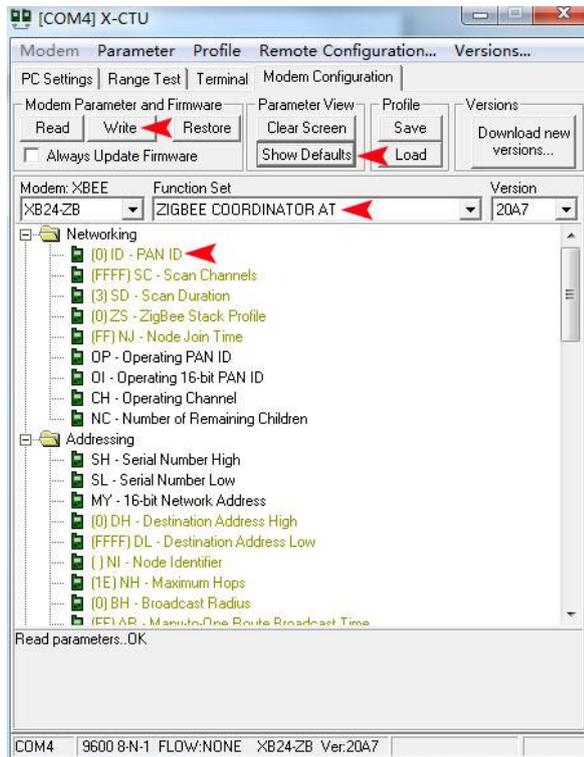
- 5) Click the button **Test/Query** to test whether the software is connected to the XBee module properly. Normally, we will see the dialog box popped out, as figure shows below. If it cannot communicate with the XBee module but all the communication parameters are set correctly, please check the USB connection and the jumper settings on the Arduino development board and the XBee module.



- 6) After a successful test, select the option **Modem Configuration**. Then, click the button **Read** to get the current parameters of the XBee module.

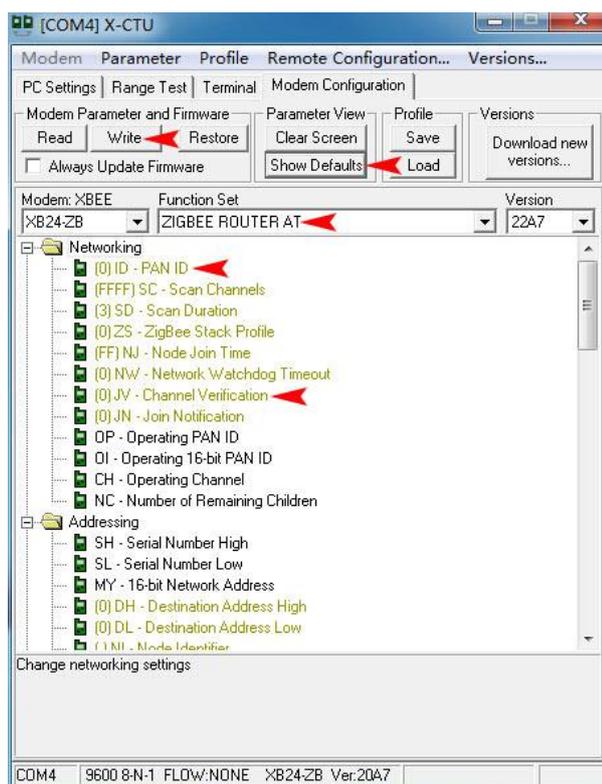


- XBee-A configuraton

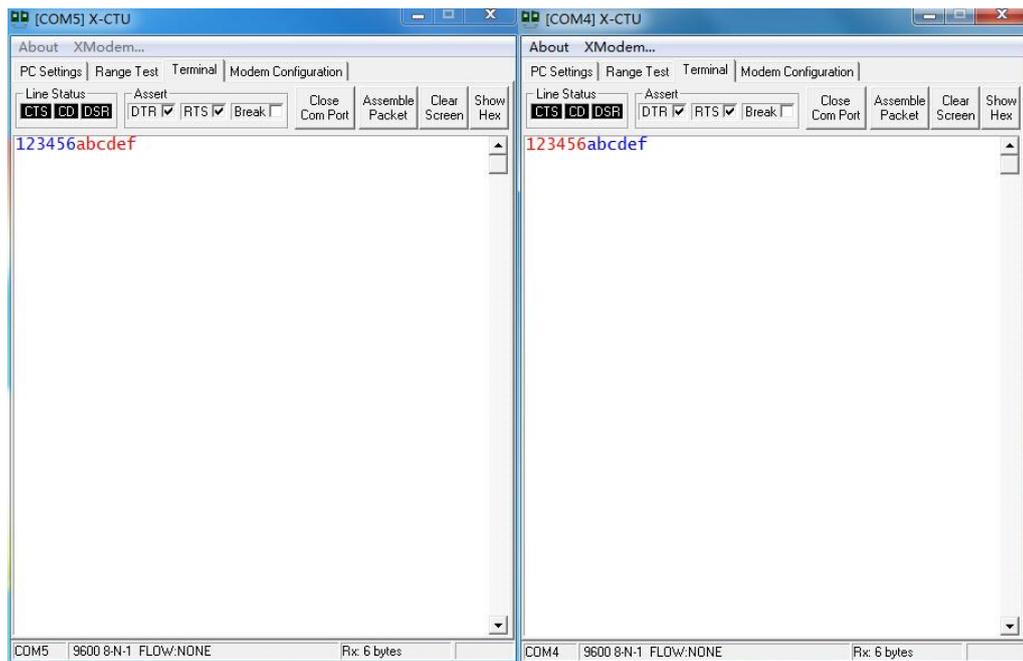


- Select the option **ZIGBEE ROUTER/END DEVICE AT** under the pull-down menu Function Set;
- Click the button **Show Defaults**;
- Select the option **PAN ID** and modify its parameter to 1234;
- Click the button **Write** to finish the setting of XBee-A.

- XBee-B configuration



- a) Select the option **ZIGBEE ROUTER/END DEVICE AT** under the pull-down menu Function Set;
  - b) Click the button **Show Defaults**;
  - c) Select the option **PAN ID** and modify its parameter to 1234;
  - d) Select the option **Channel Verification(JV)**, and modify its parameter to 1-ENABLED;
  - e) Click the button **Write** to finish the setting of XBee-B.
- 7) Configure XBee-A and XBee-B according to the processes described above. When finished, start two X-CTU tools, and select different COM interfaces in the option **PC Settings** to control Group A and Group B respectively.
  - 8) 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.



- 9) The figure above 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.

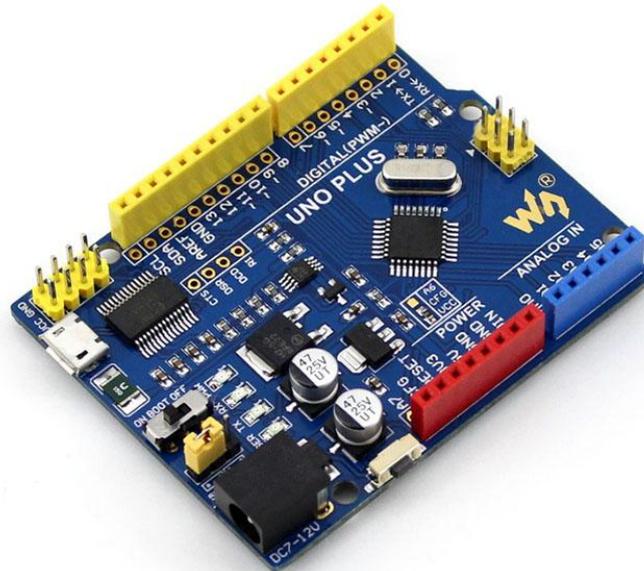
**Notices: In this operation, you should set the jumpers of Accessory Shield:**

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

### 3. Appendix:

#### 3.1. Arduino UNO PLUS

Arduino UNO PLUS is a development board based on ATMEGA328P-AU. It is compatible with Arduino UNO R3 development board, and has some improvements based on UNO R3.



For more information about Arduino UNO PLUS development board referred in this document, please refer to the links listed below:

Chinese datasheet: <http://www.waveshare.net/shop/UNO-PLUS.htm>

English datasheet: <http://www.waveshare.com/product/UNO-PLUS.htm>