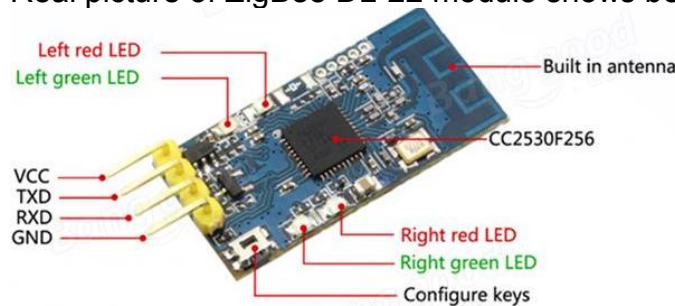


## Appendix A | Hardware Setting

### ZigBee mode debugging

Real picture of ZigBee DL-22 module shows below:



Module-configuration of DL-22 follows five steps:



1. **Enter setting-mode:** Firstly, Holding the configure key when the module is power-off. Then powering up the module and you will see that four LEDs is flashing continually. Now, you should release the keys. You have entered setting-mode.
2. **Set Baud rate:** After finishing the first step, the module will use four LEDs to indicate current baud rate. Pressing configure key will change the baud rate. Different baud rate with corresponding LED can be seen below:

	2400		4800
	9600		14400
	19200		38400
	57600		115200

3. **Set Channel:** After finishing setting baud rate, you should hold the key until the four LEDs are flashing continually and then release the key. You can set channel now. In this step, the LED which is flashing represents a channel. Pressing the key will change the channel. This module provides sixteen channels corresponding to sixteen statuses of the four LEDs (including four LED off).
4. **Set host/client:** To set host or client, you should do the same as like previous. You will need to hold the key until the four LEDs are flashing and then release the key. You can set host and client now. In this section, the lighting LED will keep flashing slowly. Pressing the key will change the pattern among three modes. Three patterns can be seen below:

	Point - Point A
	Point- Point B
	Broadcasting

5. **Confirm Setting:** After finishing all settings, you should hold the key again until the four LED are flashing and then release the key. You can see the four LEDs will

keep lighting for two seconds. Now, the settings of the module are reserved, and the module can work normally.

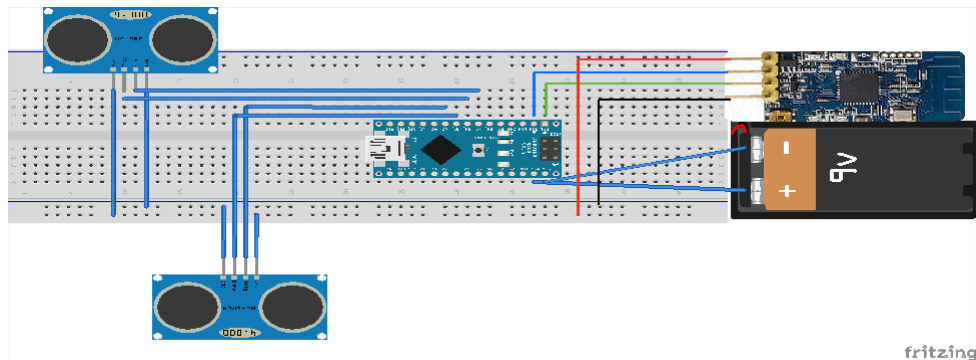
All of ZigBee modules in this project should have same channels and baud rate.

### **Arduino Nano setting with ZigBee module and sensors**

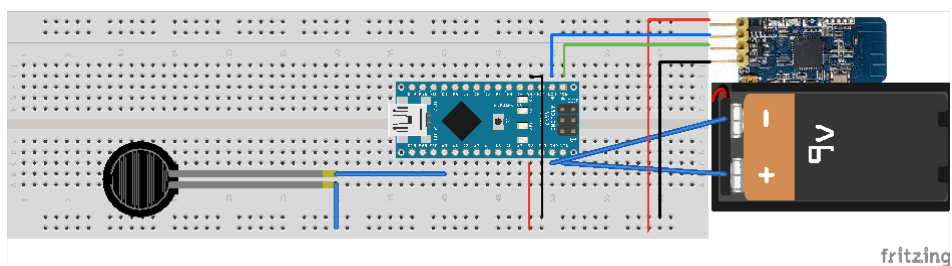
All of the Arduino Nano are connected with ZigBee modules and powered by 9-volt battery.

Arduino Nano connecting different sensors can be seen as below:

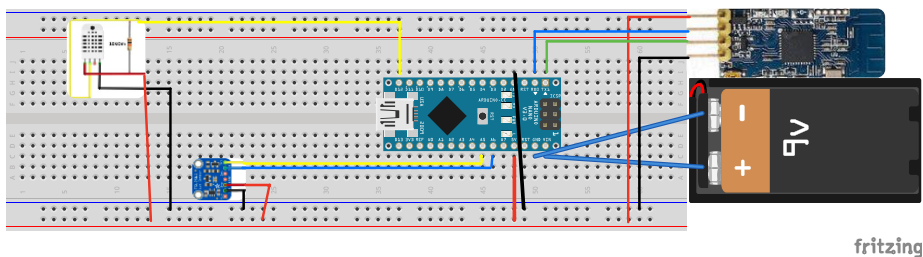
#### *HC-SR04 Ultrasonic sensor*



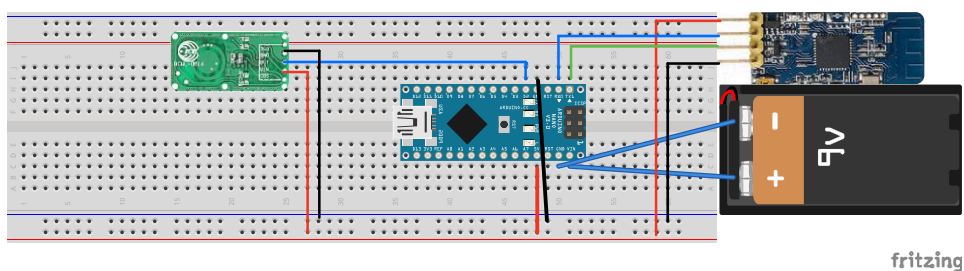
#### *RFP Force-sensitive sensor*



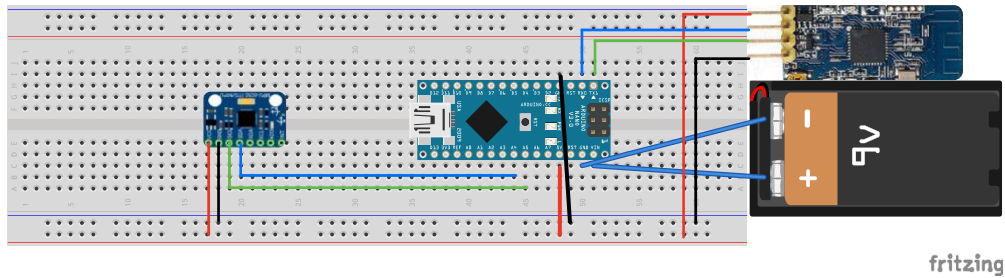
#### *DHT22 Temperature sensor & TSL2561 Digital Light sensor*



#### *RCWL-0561 Microwave Radar Motion sensor*

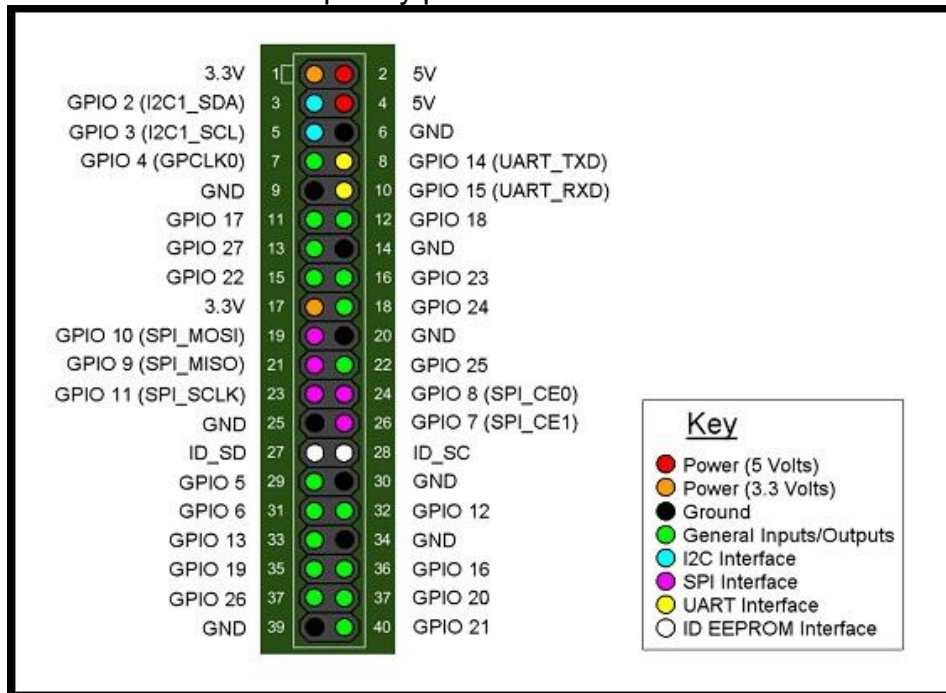


#### *MPU-6050 Accelerometer sensor*

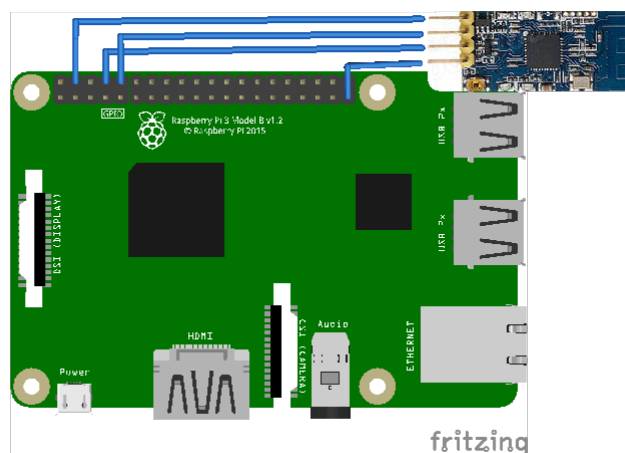


### Raspberry pi connected with ZigBee Module

GPIO overview of Raspberry pi mode 3 can be seen as below:



A ZigBee module will be connected to raspberry pi with  
VIN-5V; RXD - UART\_TXD; TXD – UART\_RXD; GND-GND  
Picture can be seen as below



## Appendix B | User Interface Guidance

To install the IOS application, you will need XCode tools and an iPhone device (E.g. iPhone 5/5s, iPhone 6/6s, iPhone 7). How to install the application via XCode can refer to:

<https://codewithchris.com/deploy-your-app-on-an-iphone/>

1. Initial interface of the application looks like Figure 3-1:
2. If the Bluetooth service have not been started, a warning message would be alert like Figure 3-2:
3. After turning on Bluetooth, you need to press the Scan button on the upper right corner. If there are Bluetooth devices nearby, the name of devices along with signal strength will show in table. An example can refer to Figure 3-3.
4. Next, you will need to select a target device. In this project, the raspberrypi was selected.
5. If the device was connected successfully, interface will turn to a Conversation page. An example can refer to Figure 3-4;

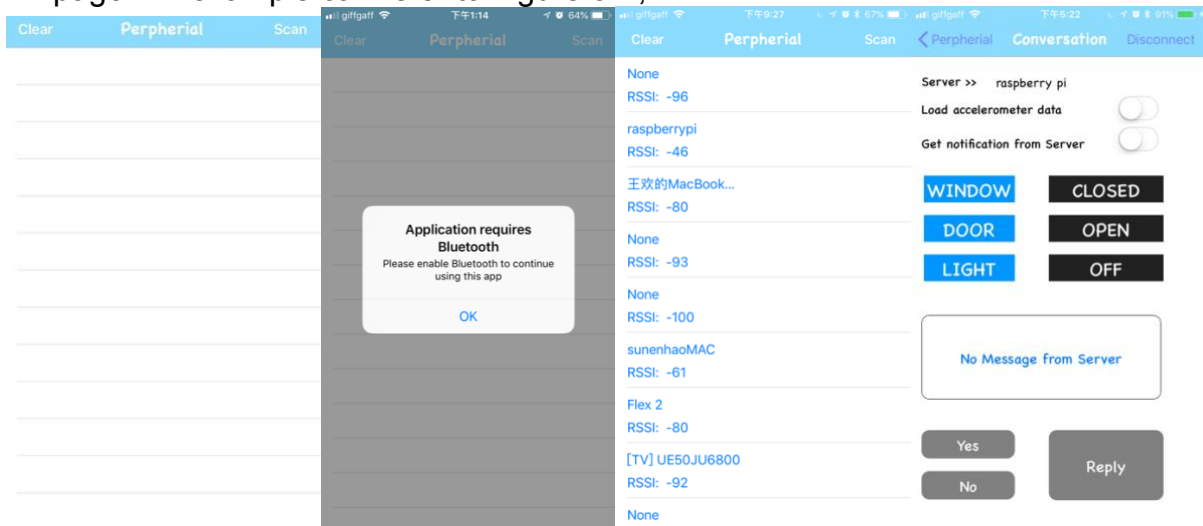


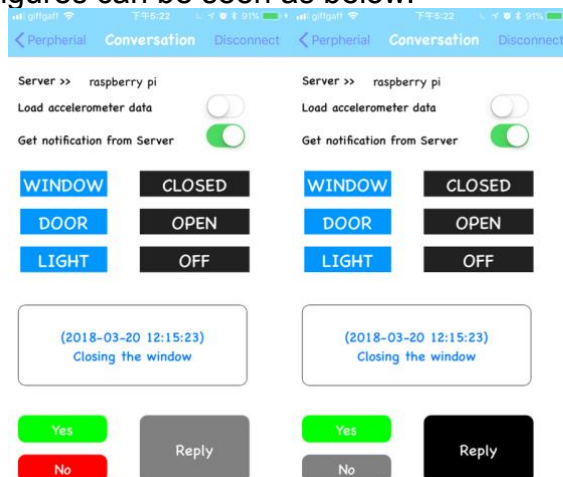
Figure 3-1

Figure 3-2

Figure 3-3

Figure 3-4

6. Now that you have connected to raspberry pi successfully, you can start subscribing the server or loading accelerometer data by switch on the button alongside the labels. If the application received message from the server, the message would be displayed in the box on the bottom. Before reply to server, you need to either select Yes or No button indicating that you agree or disagree with the message. Figures can be seen as below.



### **Running services on the server**

There are two services running on the server: Nodejs Bluetooth service and python serial port service. Both of them are in Server-programs folder. To execute the program, following the instructions as below:

#### **Nodejs server programs are in NodeJS folder.**

To run the program, firstly, you will need reach the root folder.

```
cd NodeJS/node_modules/bleno/blue
```

Next step is to run the main program:

```
sudo node main.js
```

The service is now running.

#### **Programs of data receiving from serial port are in Python\_server folder.**

To run the program, firstly you will need reach the root folder.

```
cd Python_server
```

Next step is to run the main program:

```
sudo python3 receive.py
```

The service is now running.

Other services for collecting data from a single sensor include pressReceive.py, sonarReceive.py, daReceive.py, waReceive.py.

### **Load Arduino programs**

To load Arduino programs to your board, you will need to download the IDE from the website:

<https://www.arduino.cc/en/Main/Software>

The libraries should also be installed by IDE.

Further guidance about how to download libraries and load the sketch can refer to the website:

<https://www.arduino.cc/en/Guide/HomePage>