Mindo SDK Instruction Manual

Mindo_connect.jar library specification

Package mindo.connect

Class	
Mindo_connect	The class Mindo_connect read bluetooth data from Mindo.

Constructor
Mindo_connect(int x, int y, int z) Creates a new Mindo_connect instance by Mindo parameters. **Parameters:** x – Resolution (16 or 24bits) x – Channel (up to 64) z – SampleRate (up to 512)

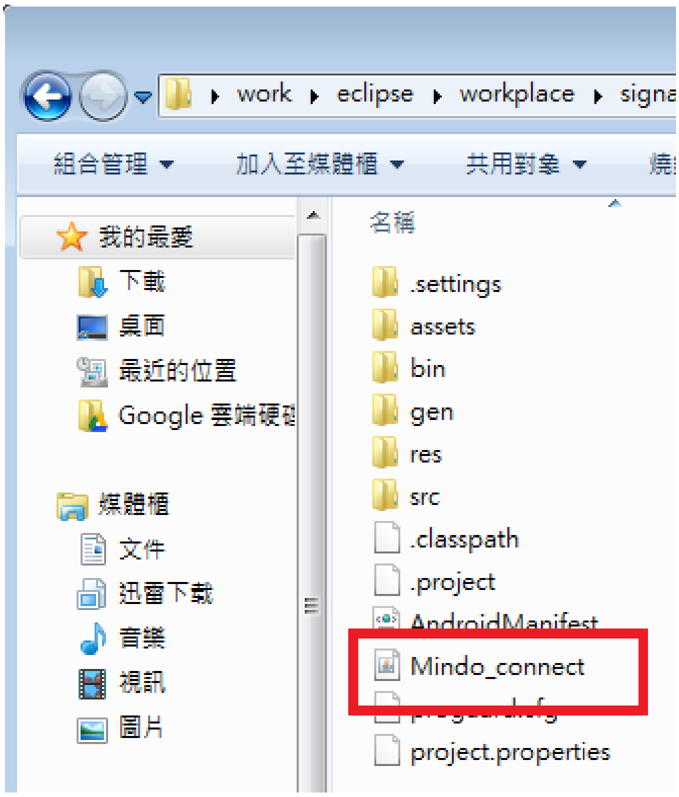
Method	
byte[]	parseStartCommand(int x) Use the return value to start connecting. **Parameters:** x - gain value
byte[]	parseStopCommand(int x) Use the return value to stop connecting. **Parameters:** x - gain value
byte[]	NotchFilter_on(int x) Use the return value to filter 60Hz data. **Parameters:** x - gain value
byte[]	NotchFilter_off(int x) Use the return value to turn off the filter function. **Parameters:** x - gain value
byte[]	Change_Gain(int x) Use the return value to change the gain from Mindo device. **Parameters:** x - gain value (1,2,3,4,6,8,12)
int[]	getMindoData (InputStream x); Return number of channels data. (-2.4V~2.4V) **Parameters:** x - the next byte of input.

int[]	getDataLostCount(); Return number of data lost when it happened.

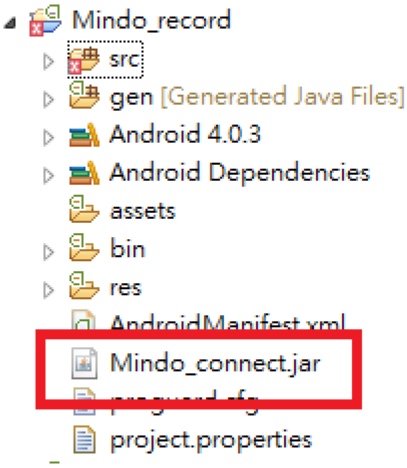
※Each use method “getMindoData (InputStream x);”, it will return number of channels data.(If SampleRate is 256, it will receive 256 part of data.)

How to import jar file into Android project?

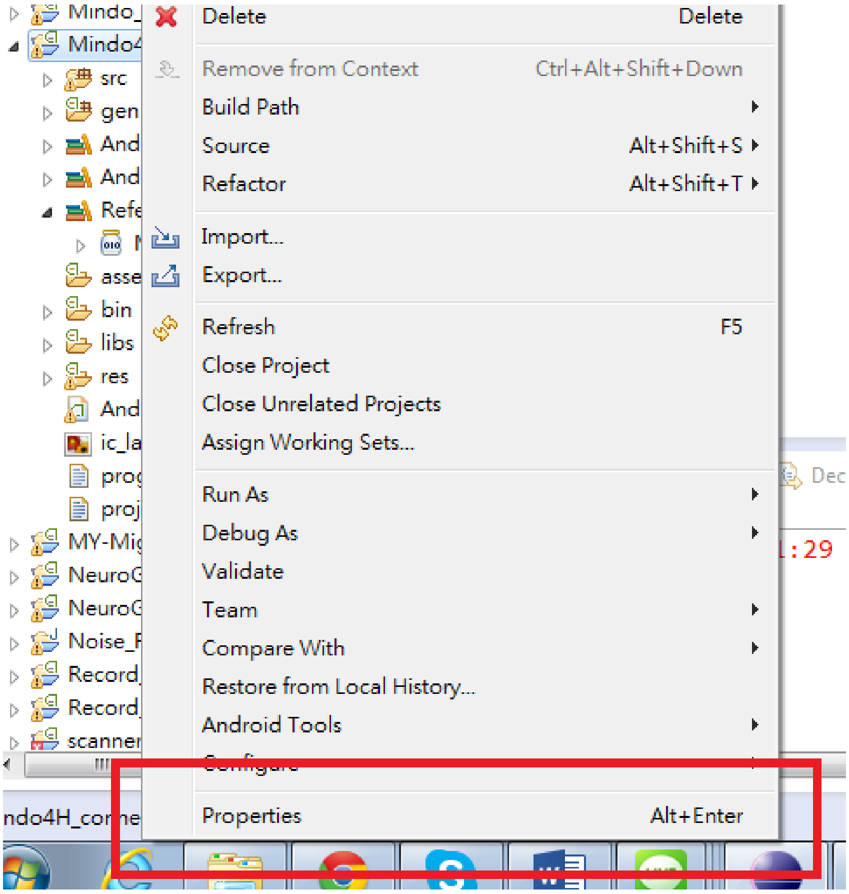
◎ Step 1: Copy Mindo_connect.jar into Android project.



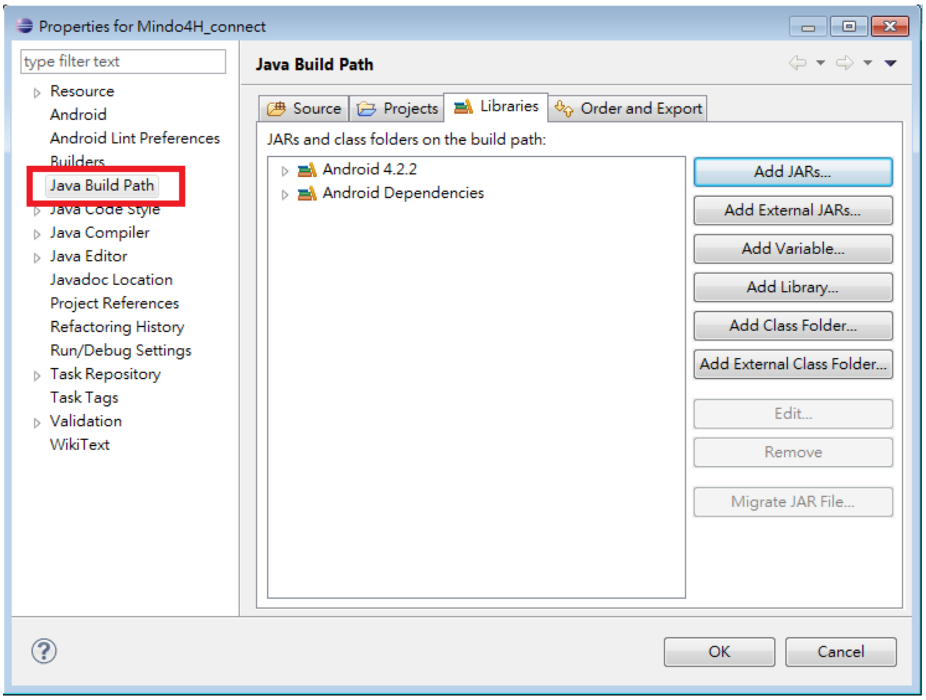
◎ Step 2: You can find the Mindo_connect.jar file in the Android project in the Eclipse.



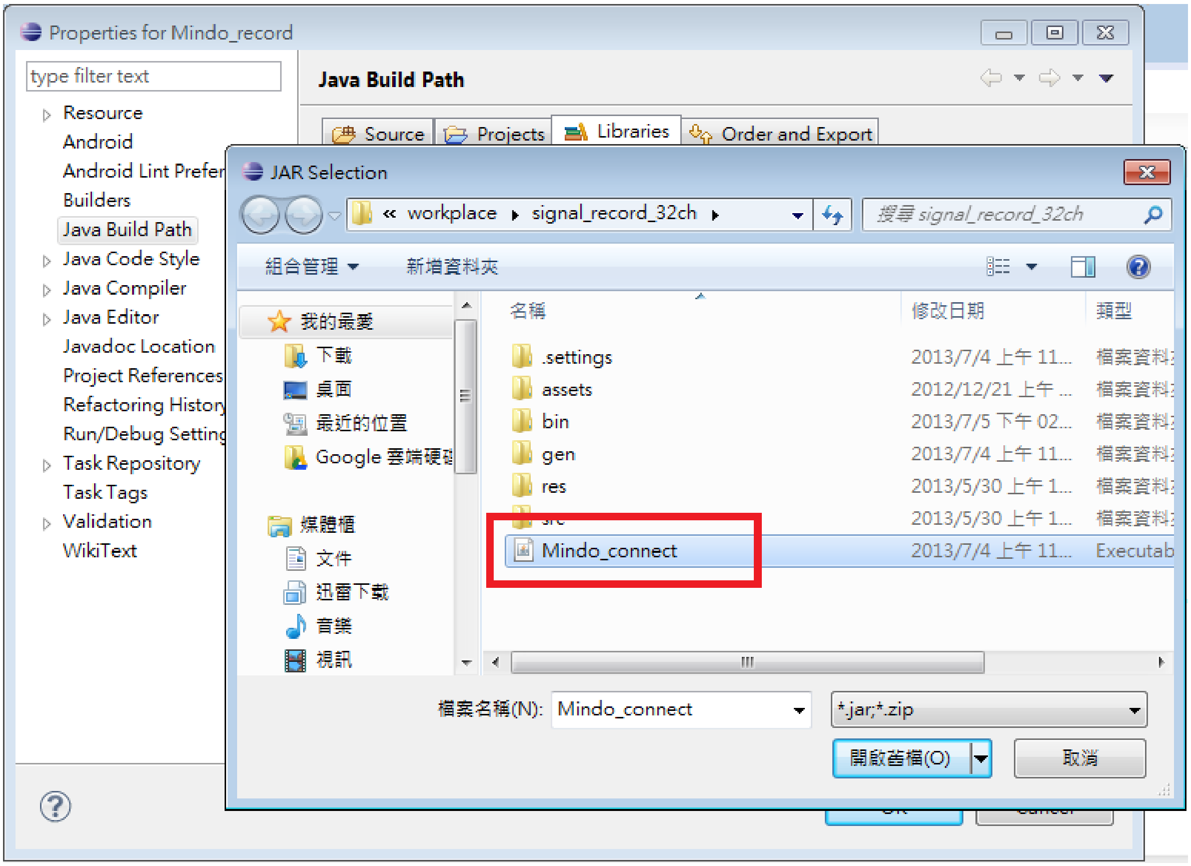
◎ Step 3: Click the right button on the mouse on the Android project, then select Properties option.



◎ Step 4: Select Java Build Path



◎ Step 5: Select Add External JARs..，then select Mindo_connect.jar file to finish this step.

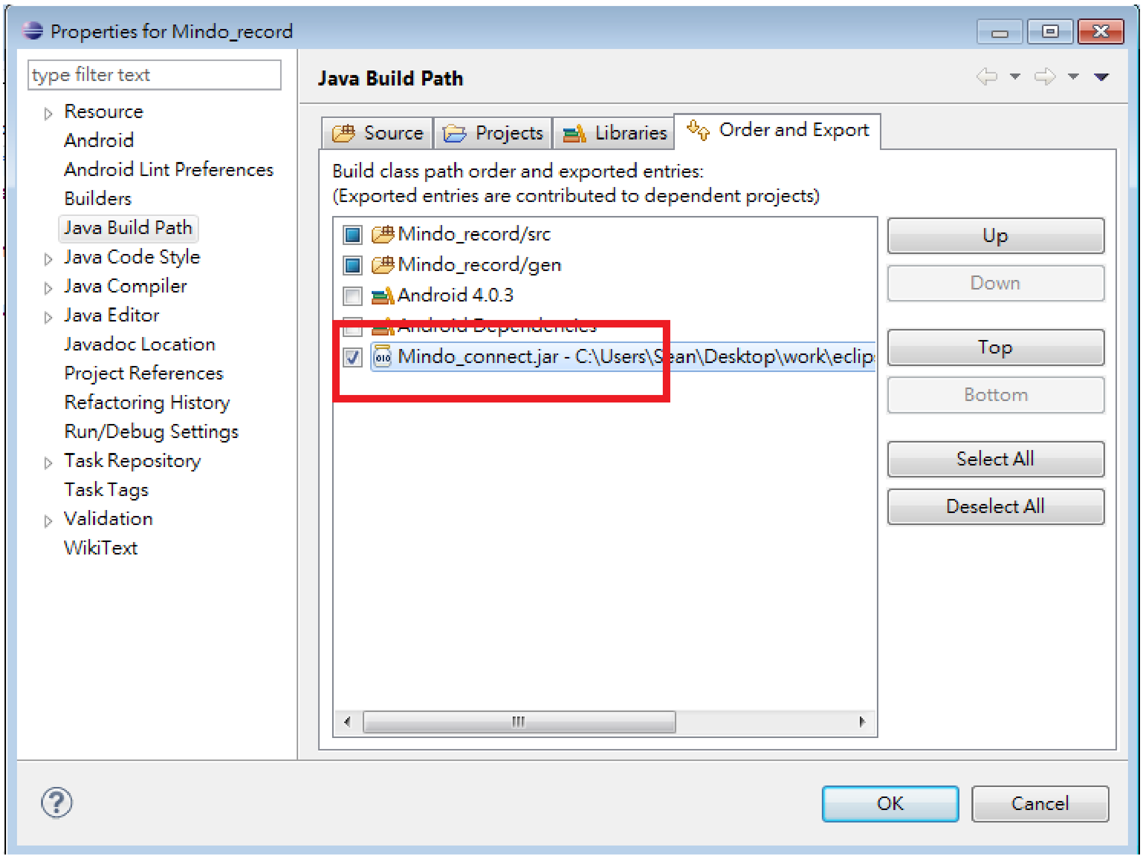


名稱	修改日期	類型
.settings	2013/7/4 上午 11...	檔案資料
assets	2012/12/21 上午 ...	檔案資料
bin	2013/7/5 下午 02...	檔案資料
gen	2013/7/4 上午 11...	檔案資料
res	2013/5/30 上午 1...	檔案資料
src	2013/5/30 上午 1...	檔案資料
Mindo_connect	2013/7/4 上午 11...	Executab

檔案名稱(N): Mindo_connect

.jar;.zip

◎ Step 6: Select Order and Export option，then check Mindo_connect.jar option。



Properties for Mindo_record

Java Build Path

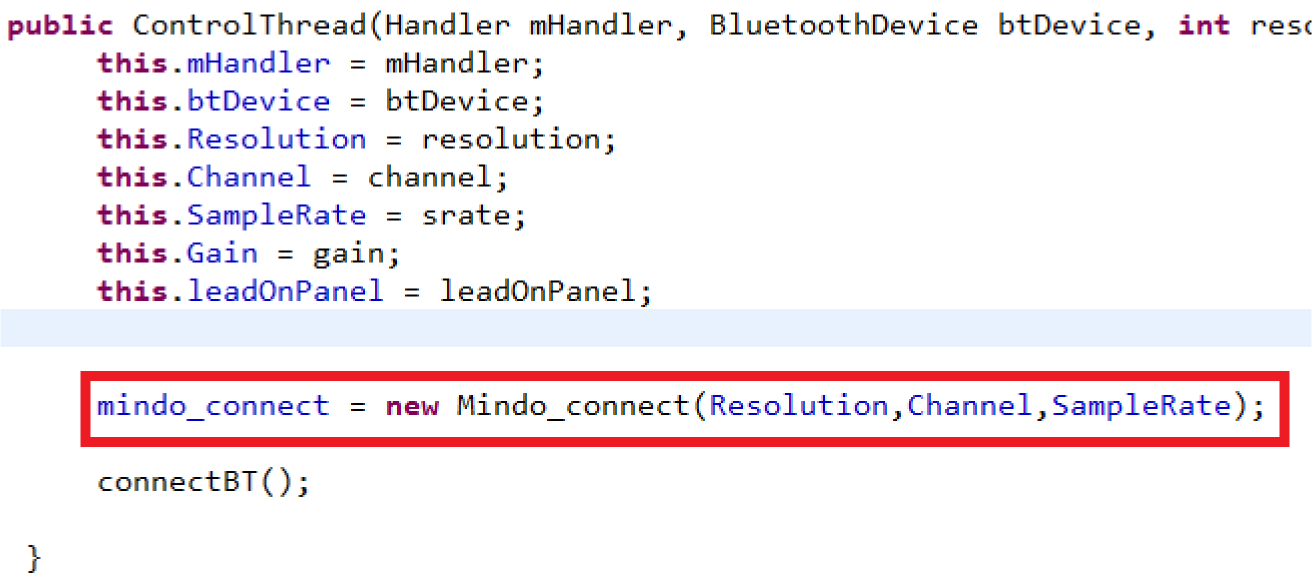
Build class path order and exported entries:
(Exported entries are contributed to dependent projects)

- ■ Mindo_record/src
- ■ Mindo_record/gen
- □ Android 4.0.3
- □ Android Dependencies
- ☑ Mindo_connect.jar - C:\Users\Sean\Desktop\work\eclip

※Please make sure to check this option, or it is unable to run normally.

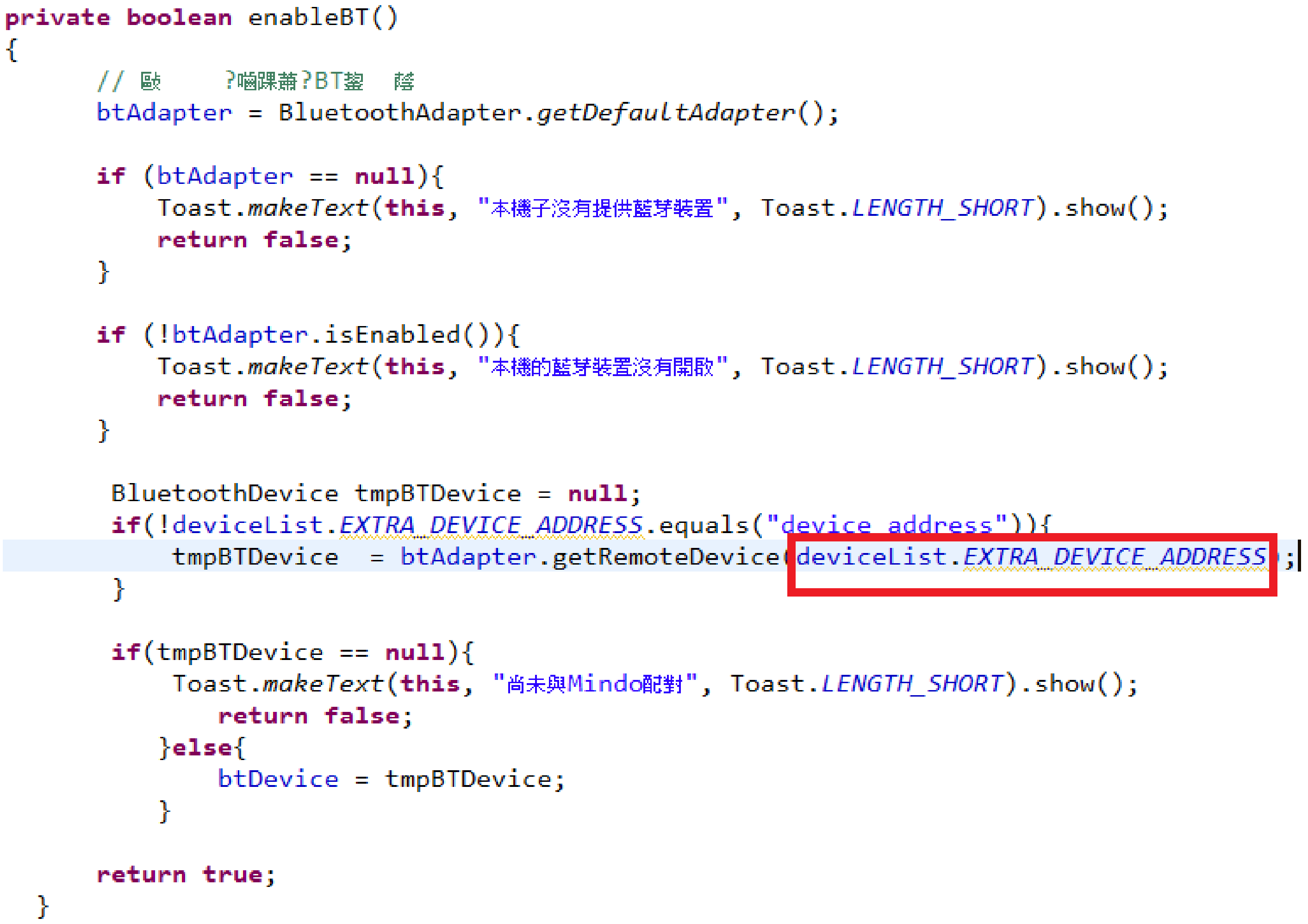
How to read the Bluetooth data from Mindo using Android code?

◎ Step 1: Create Mindo_connect instance in the Constructor.



```
public ControlThread(Handler mHandler, BluetoothDevice btDevice, int reso
     this.mHandler = mHandler;
     this.btDevice = btDevice;
     this.Resolution = resolution;
     this.Channel = channel;
     this.SampleRate = srate;
     this.Gain = gain;
     this.leadOnPanel = leadOnPanel;

     mindo_connect = new Mindo_connect(Resolution,Channel,SampleRate);

     connectBT();

 }
```

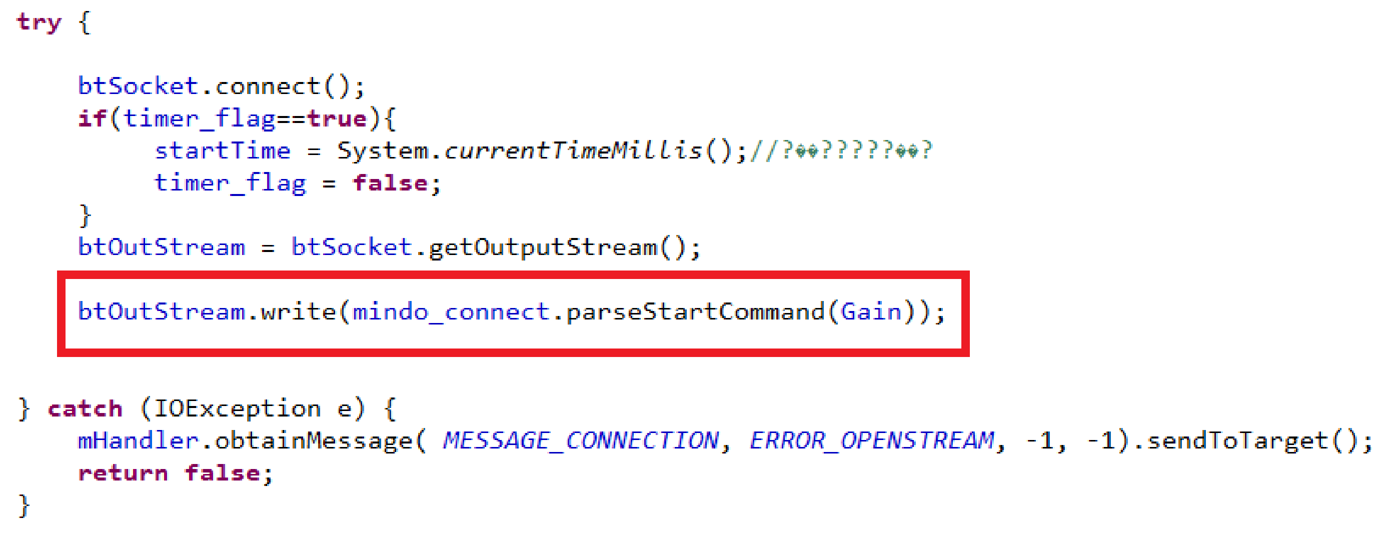
◎ Step 2: Create variables for Bluetooth connection.



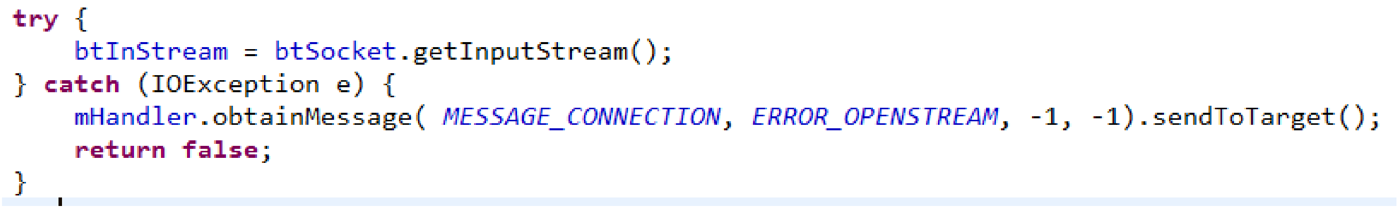
```java
private boolean enableBT()
{
      // 啟      ?啲踝蕭?BT鍳   瓘
      btAdapter = BluetoothAdapter.getDefaultAdapter();

      if (btAdapter == null){
          Toast.makeText(this, "本機子沒有提供藍芽裝置", Toast.LENGTH_SHORT).show();
          return false;
      }

      if (!btAdapter.isEnabled()){
          Toast.makeText(this, "本機的藍芽裝置沒有開啟", Toast.LENGTH_SHORT).show();
          return false;
      }

       BluetoothDevice tmpBTDevice = null;
       if(!deviceList.EXTRA_DEVICE_ADDRESS.equals("device address")){
           tmpBTDevice  = btAdapter.getRemoteDevice(deviceList.EXTRA_DEVICE_ADDRESS);
       }

       if(tmpBTDevice == null){
           Toast.makeText(this, "尚未與Mindo配對", Toast.LENGTH_SHORT).show();
              return false;
          }else{
              btDevice = tmpBTDevice;
          }

      return true;
  }
```

※ You might need to write the Bluetooth address in "BluetoothAdapter.getRemoteDevice"and input the proper parameter.

◎ Step 2: After getting variables "BluetoothDevice", now you can connect with Mindo and get variables "InputStream".



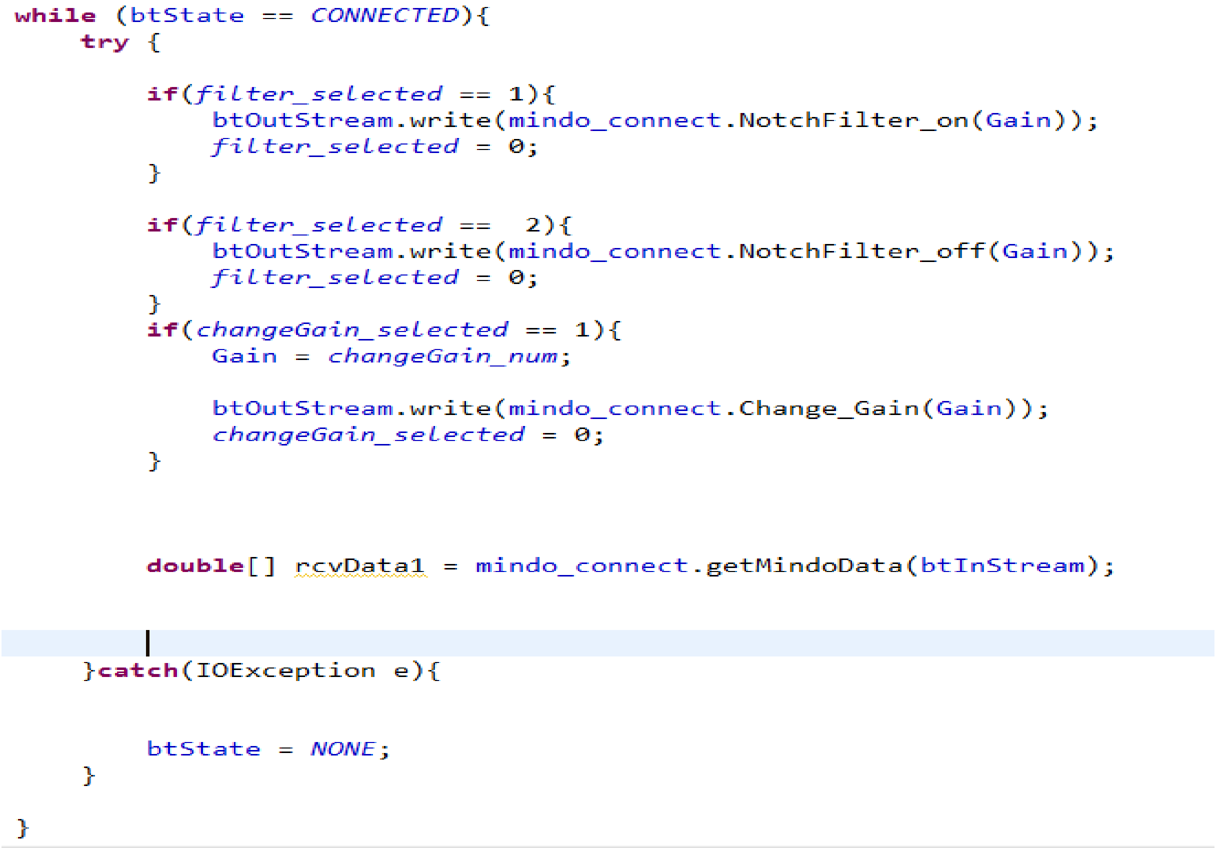
```java
try {

    btSocket.connect();
    if(timer_flag==true){
         startTime = System.currentTimeMillis();//?��?????��?
         timer_flag = false;
    }
    btOutStream = btSocket.getOutputStream();

    btOutStream.write(mindo_connect.parseStartCommand(Gain));

} catch (IOException e) {
    mHandler.obtainMessage( MESSAGE_CONNECTION, ERROR_OPENSTREAM, -1, -1).sendToTarget();
    return false;
}
```



```java
try {
    btInStream = btSocket.getInputStream();
} catch (IOException e) {
    mHandler.obtainMessage( MESSAGE_CONNECTION, ERROR_OPENSTREAM, -1, -1).sendToTarget();
    return false;
}
```

※ If you want to start to read Bluetooth data from Mindo, you need to call method “parseStartCommand(int x)” from Mindo_connect class.

◎ Step 3: After linking to the Mindo device via Bluetooth, now you can call the method “getMindoData(InputStream x)” and start to read data.



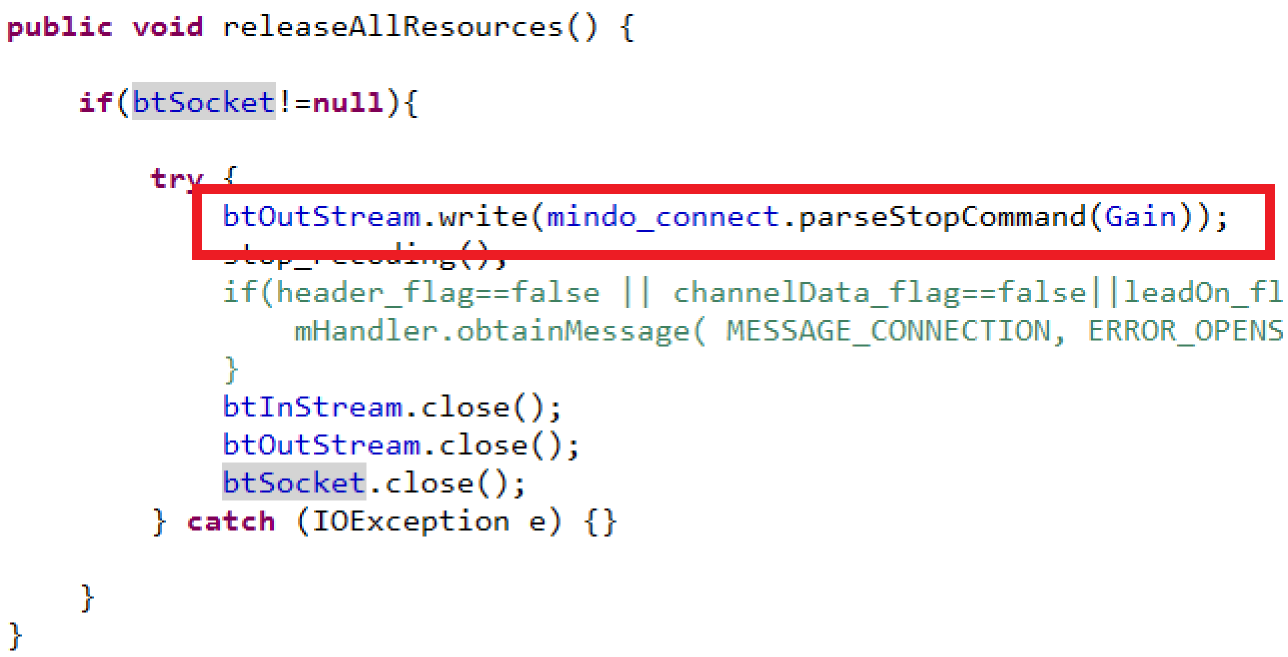
```
while (btState == CONNECTED){
    try {

        if(filter_selected == 1){
            btOutStream.write(mindo_connect.NotchFilter_on(Gain));
            filter_selected = 0;
        }

        if(filter_selected ==  2){
            btOutStream.write(mindo_connect.NotchFilter_off(Gain));
            filter_selected = 0;
        }
        if(changeGain_selected == 1){
            Gain = changeGain_num;

            btOutStream.write(mindo_connect.Change_Gain(Gain));
            changeGain_selected = 0;
        }



        double[] rcvData1 = mindo_connect.getMindoData(btInStream);



    }catch(IOException e){


        btState = NONE;
    }

}
```

※ If you want to turn on the filter function, you need to call method NotchFilter_on(int x) from Mindo_connect class.

※ If you want to turn off the filter function, you need to call method NotchFilter_off(int x) from Mindo_connect class.

※ If you want to change the gain value, you need to call method Change_Gain(int x) from Mindo_connect class.

◎ Step 4: If you want to stop to read data, you need to call method “parseStopCommand(int x)”.



```
public void releaseAllResources() {

    if(btSocket!=null){

        try {
            btOutStream.write(mindo_connect.parseStopCommand(Gain));
            stop_recording();
            if(header_flag==false || channelData_flag==false||leadOn_fl
                mHandler.obtainMessage( MESSAGE_CONNECTION, ERROR_OPENS
            }
            btInStream.close();
            btOutStream.close();
            btSocket.close();
        } catch (IOException e) {}

    }
}
```