Lecture 7 Java常駐程式

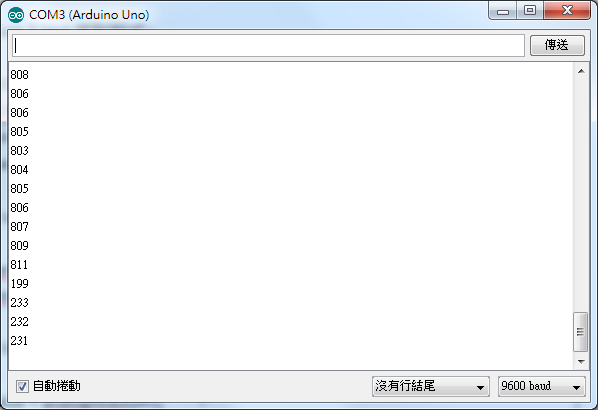
Step1:

arduino program: 取得光敏電阻值

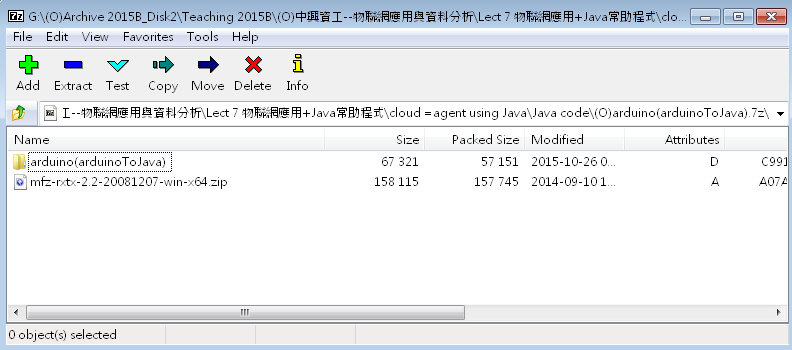
|  |
| --- |
| int pin = A2; // select the input pin for the potentiometer  int value = 0; // variable to store the value coming from the sensor  void setup()  {  Serial.begin(9600);  pinMode(pin, INPUT);  }  void loop()  {  value = analogRead(pin);  Serial.println(value);  delay(1000); //一秒取得一次值  } |

Step 2 testing from serial port monitor in arduino

輸出結果(序列埠監控視窗)會如下圖



Step 3 Java 常駐程式

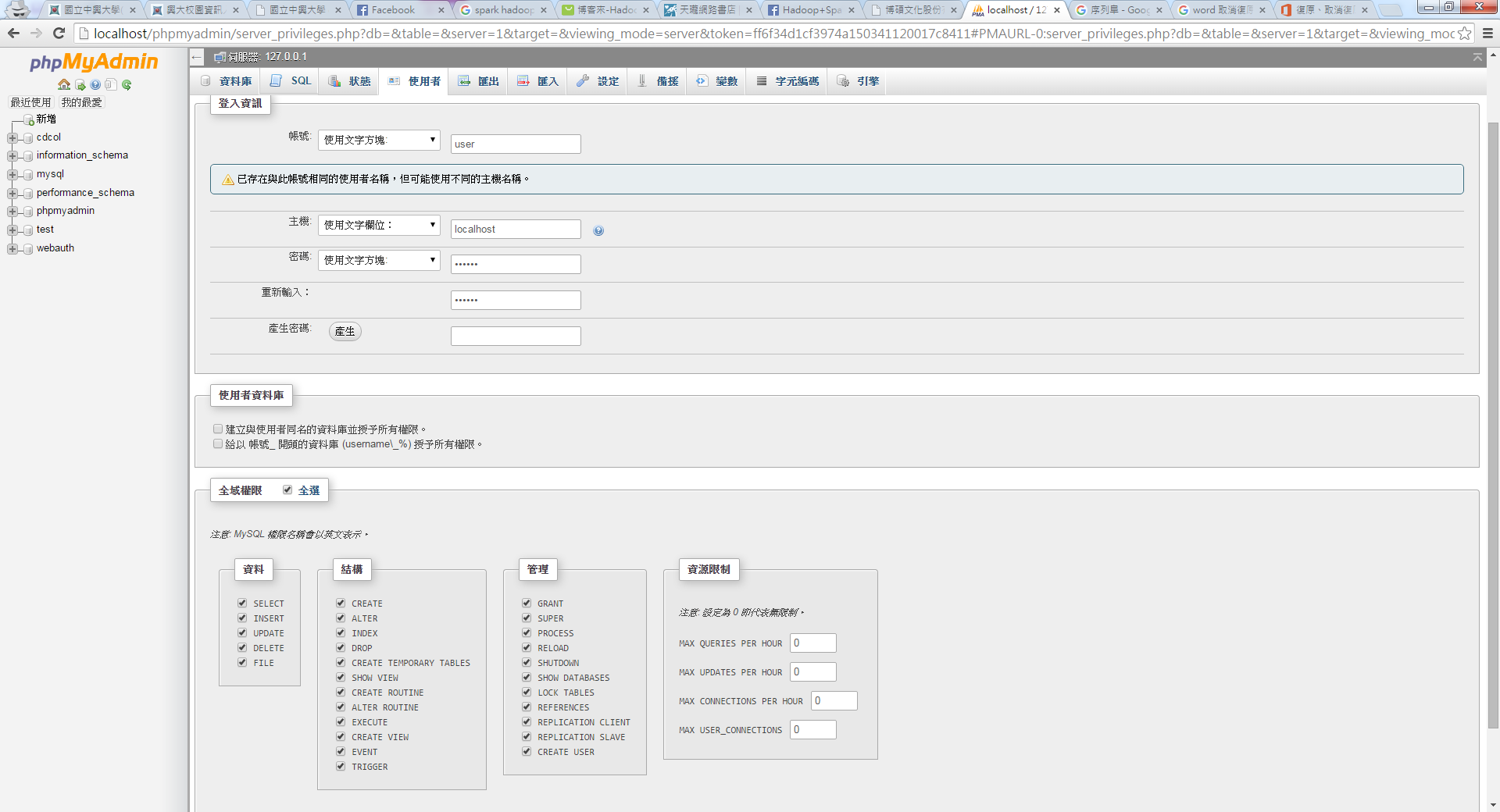
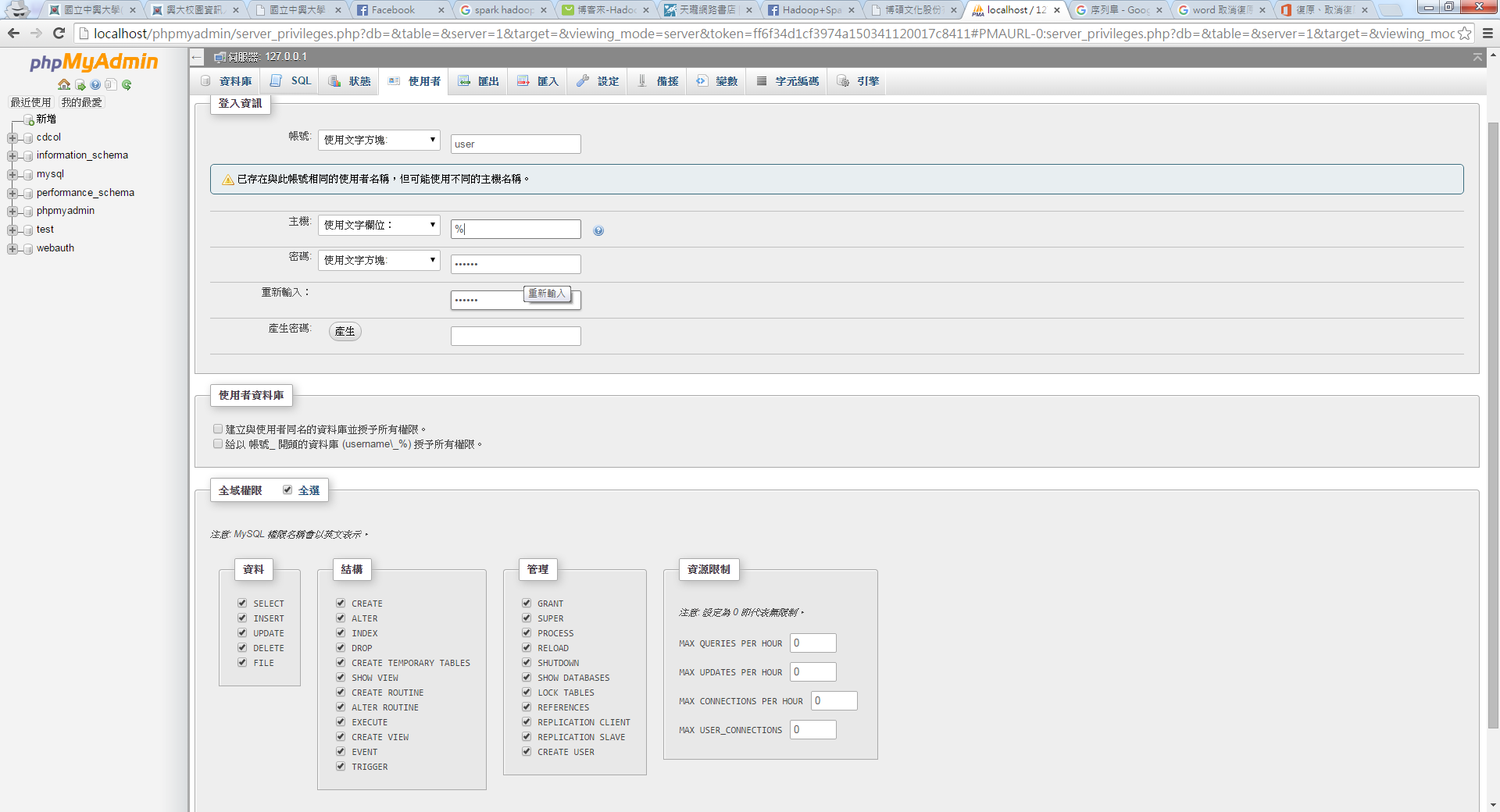


http://puremonkey2010.blogspot.tw/2011/07/java-java-rs232.html

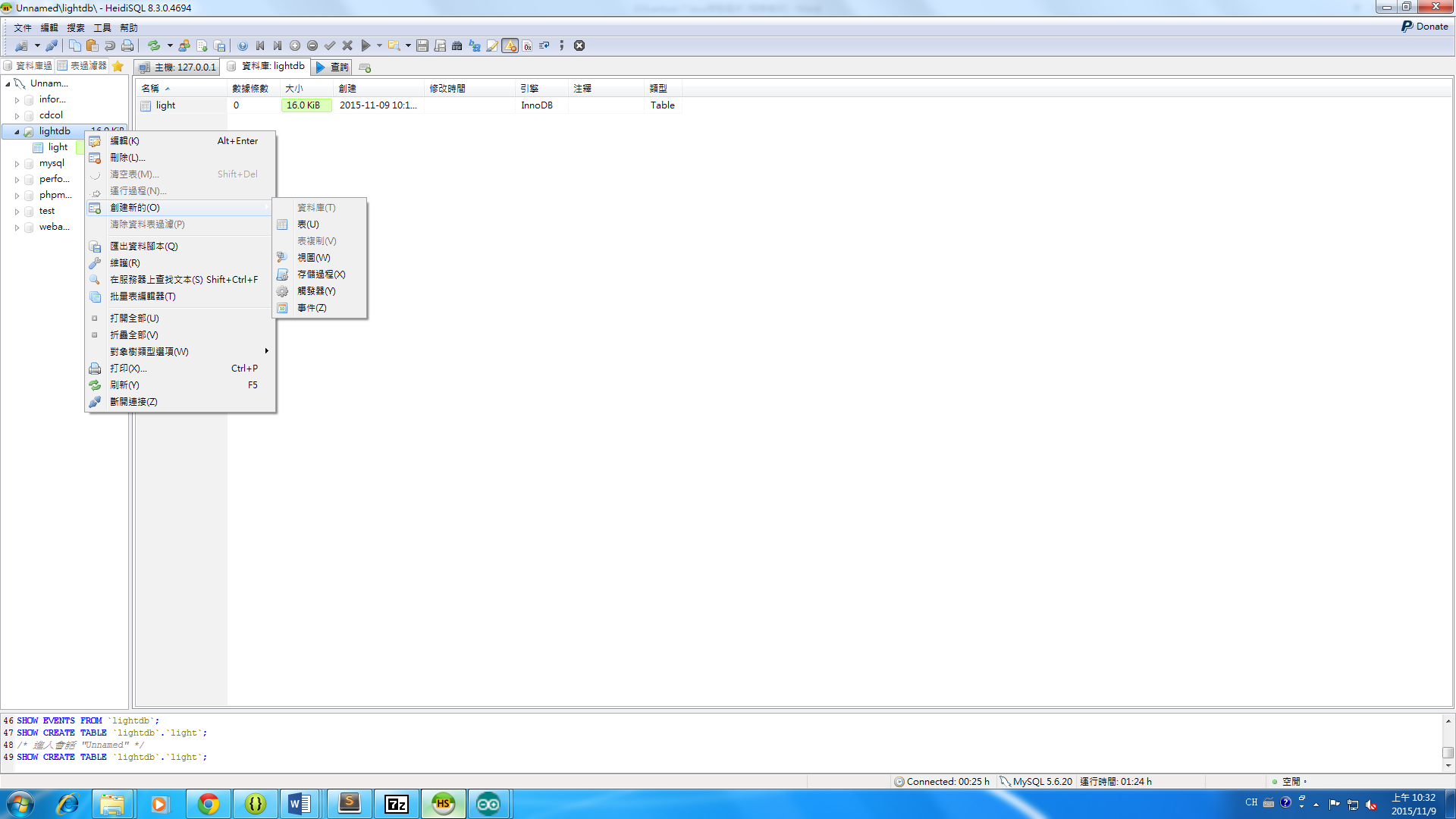
Step4: follow ppt

* put C library dll (rxtxSerial.dll、rxtxParallel.dll) 放入 Java/jre/bin
* 加入 java wrapper ==> RxTxComm.jar into project
* 加入 RxTxCom.jar , mysql-connector-java-5.1.32-bin.jar into build path讓link/build 在一起

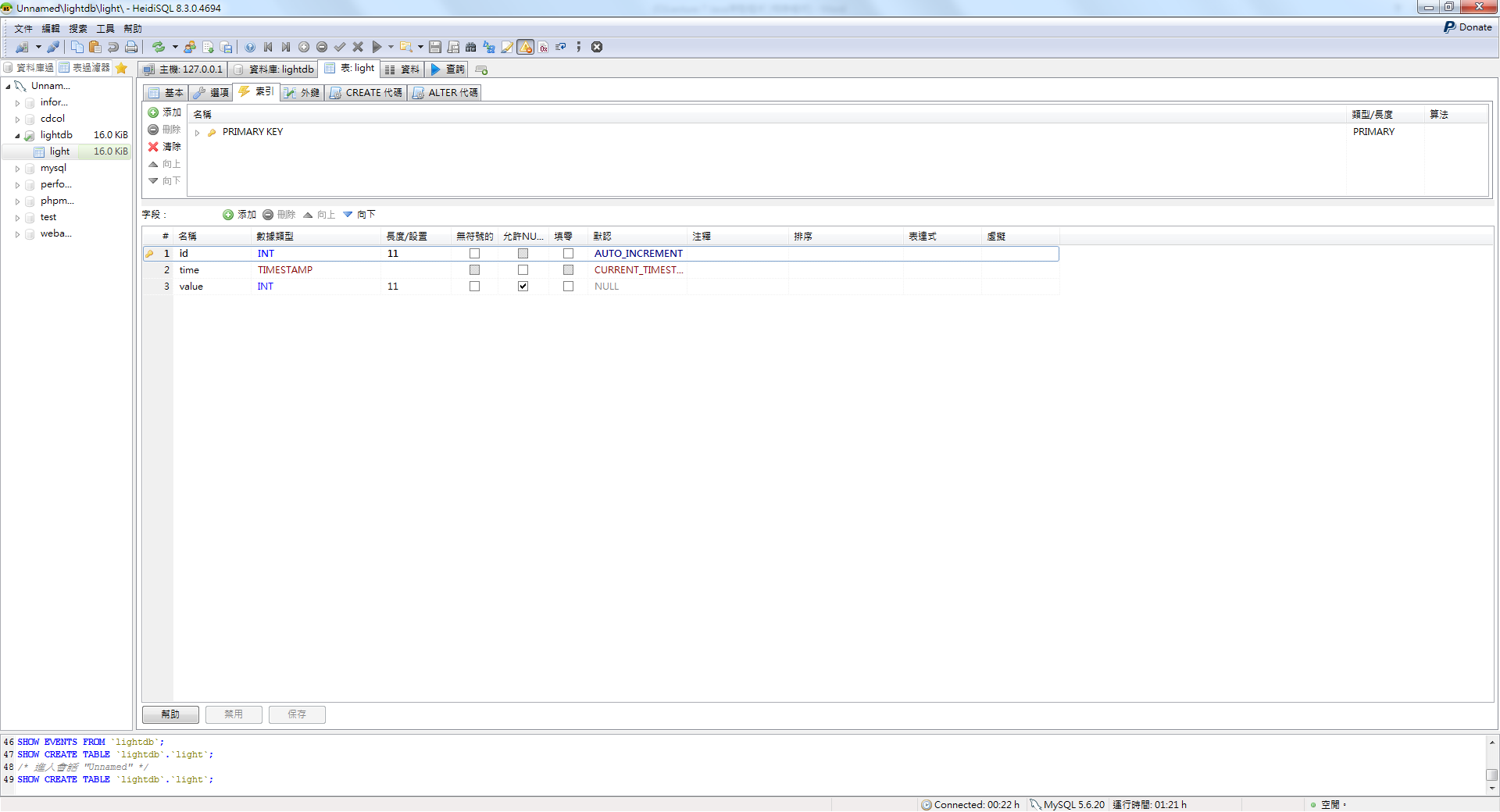
新建使用者

需要新建資料庫lightDB 再建light table



Light table的內容如下



Java to資料庫(code)

**static** **final** String *USER* = "user"; =>改成剛剛新創的使用者

**static** **final** String *PASS* = "pass"; =>設定的使用者密碼

**if** (currPortId.getName().equals("COM3")) =>把PORT改成自己所接收的PORT號

**package** datatomysql;

**import** java.io.BufferedReader;

**import** java.io.InputStreamReader;

**import** gnu.io.CommPortIdentifier;

**import** gnu.io.SerialPort;

**import** gnu.io.SerialPortEvent;

**import** gnu.io.SerialPortEventListener;

**import** java.util.Enumeration;

**import** java.sql.\*;

**public** **class** light **implements** SerialPortEventListener {

SerialPort serialPort;//定義serialport物件

**private** BufferedReader input;//宣告input buffer

**private** **static** **final** **int** *TIME\_OUT* = 2000;//設定等待port開啟的時間，單位為毫秒

**private** **static** **final** **int** *DATA\_RATE* = 9600;//設定baud rate為9600

//設定server IP,帳號,密碼

**static** **final** String *JDBC\_DRIVER* = "com.mysql.jdbc.Driver";//設定JDBC driver

**static** **final** String *DB\_URL* = "jdbc:mysql://localhost/lightDB";

//server IP後街資料庫名稱

**static** **final** String *USER* = "user";

**static** **final** String *PASS* = "pass";

**public** **void** initialize() {

CommPortIdentifier portId = **null**;//定義CommPortIdentifier物件，控制接收接收port

Enumeration portEnum = CommPortIdentifier.*getPortIdentifiers*();//儲存所有有效的port

**while** (portEnum.hasMoreElements()) {//掃過所有的port

CommPortIdentifier currPortId = (CommPortIdentifier) portEnum.nextElement();//定義currPortId

**if** (currPortId.getName().equals("COM3")) {//設定arduino serial port

portId = currPortId;

**break**;

}

}

**if** (portId == **null**) {//如果com port設定錯誤，會出現這一行

System.*out*.println("Could not find COM port.");

**return**;

}

**try** {//連接port

//open serial port

serialPort = (SerialPort) portId.open(**this**.getClass().getName(),*TIME\_OUT*);

//設定port parameters

serialPort.setSerialPortParams(*DATA\_RATE*,SerialPort.*DATABITS\_8*,SerialPort.*STOPBITS\_1*,SerialPort.*PARITY\_NONE*);

//open the streams

input = **new** BufferedReader(**new** InputStreamReader(serialPort.getInputStream()));

// add event listeners

serialPort.addEventListener(**this**);// Registers a SerialPortEventListener object to listen for SerialEvents.

serialPort.notifyOnDataAvailable(**true**);//Expresses interest in receiving notification when input data is available.

} **catch** (Exception e) {

System.*err*.println(e.toString());

}

}

//處理serial port事件,讀取資料並print出來

**public** **void** serialEvent(SerialPortEvent oEvent) {

Connection connection = **null**;//建立Connection物件

Statement statement = **null**;//建立Statement物件

**if** (oEvent.getEventType() == SerialPortEvent.*DATA\_AVAILABLE*) {//if data available on serial port

**try** {

//連接mysql database

connection = DriverManager.*getConnection*(*DB\_URL*, *USER*, *PASS*);

System.*out*.println("SQL Connection to database established!");

String inputLine=input.readLine();

System.*out*.println(inputLine);

//執行query

statement = connection.createStatement();

PreparedStatement pstmt = connection.prepareStatement("INSERT INTO light SET value = ('"+inputLine+"')");

pstmt.executeUpdate();

pstmt.close();

statement.close();

connection.close();

} **catch** (SQLException e) {

//Handle errors for JDBC

System.*out*.println("Connection Failed! Check output console");

**return**;

} **catch** (Exception e) {

//System.err.println(e.toString());

}**finally** {

**try**

{

**if**(connection != **null**)

connection.close();

System.*out*.println("Connection closed !!");

} **catch** (SQLException e) {

e.printStackTrace();

}

}

}

}

**public** **static** **void** main(String[] args) **throws** Exception {

light main = **new** light();//creates an object of the class

main.initialize();

//call to ensure the driver is registered

**try**

{

Class.*forName*(*JDBC\_DRIVER*);

}

**catch** (ClassNotFoundException e) {

System.*out*.println("MySQL JDBC Driver not found !!");

**return**;

}

System.*out*.println("Started");

}

}

**public** **void** initialize() {

CommPortIdentifier portId = **null**;

//定義CommPortIdentifier物件，控制接收接收port

Enumeration portEnum = CommPortIdentifier.*getPortIdentifiers*();

//儲存所有有效的port

**while** (portEnum.hasMoreElements()) {//掃過所有的port

CommPortIdentifier currPortId = (CommPortIdentifier) portEnum.nextElement();//定義currPortId

**if** (currPortId.getName().equals("COM3")) {//設定arduino serial port

portId = currPortId;

**break**;

}

}

**if** (portId == **null**) {//如果com port設定錯誤，會出現這一行

System.*out*.println("Could not find COM port.");

**return**;

}

**try** {//連接port //open serial port

serialPort = (SerialPort) portId.open(**this**.getClass().getName(),*TIME\_OUT*);

//設定port parameters

serialPort.setSerialPortParams(*DATA\_RATE*,SerialPort.*DATABITS\_8*,SerialPort.*STOPBITS\_1*,SerialPort.*PARITY\_NONE*);

//open the streams

input = **new** BufferedReader(**new** InputStreamReader(serialPort.getInputStream()));

// add event listeners

serialPort.addEventListener(**this**);// Registers a SerialPortEventListener object to listen for SerialEvents.

serialPort.notifyOnDataAvailable(**true**);//Expresses interest in receiving notification when input data is available.

} **catch** (Exception e) {

System.*err*.println(e.toString());

}

}

//處理serial port事件,讀取資料並print出來

**public** **void** serialEvent(SerialPortEvent oEvent) {

Connection connection = **null**;//建立Connection物件

Statement statement = **null**;//建立Statement物件

**if** (oEvent.getEventType() == SerialPortEvent.*DATA\_AVAILABLE*) {

//if data available on serial port

**try** {

//連接mysql database

connection = DriverManager.*getConnection*(*DB\_URL*, *USER*, *PASS*);

System.*out*.println("SQL Connection to database established!");

String inputLine=input.readLine();

System.*out*.println(inputLine);

//執行query

statement = connection.createStatement();

PreparedStatement pstmt = connection.prepareStatement("INSERT INTO light SET value = ('"+inputLine+"')");

pstmt.executeUpdate();

pstmt.close();

statement.close();

connection.close();

} **catch** (SQLException e) {

//Handle errors for JDBC

System.*out*.println("Connection Failed! Check output console");

**return**;

} **catch** (Exception e) {

//System.err.println(e.toString());

}**finally** {

**try**

{

**if**(connection != **null**)

connection.close();

System.*out*.println("Connection closed !!");

} **catch** (SQLException e) {

e.printStackTrace();

}

}

}

}

**public** **static** **void** main(String[] args) **throws** Exception {

light main = **new** light();//creates an object of the class

main.initialize();

//call to ensure the driver is registered

**try**

{

Class.*forName*(*JDBC\_DRIVER*);

}

**catch** (ClassNotFoundException e) {

System.*out*.println("MySQL JDBC Driver not found !!");

**return**;

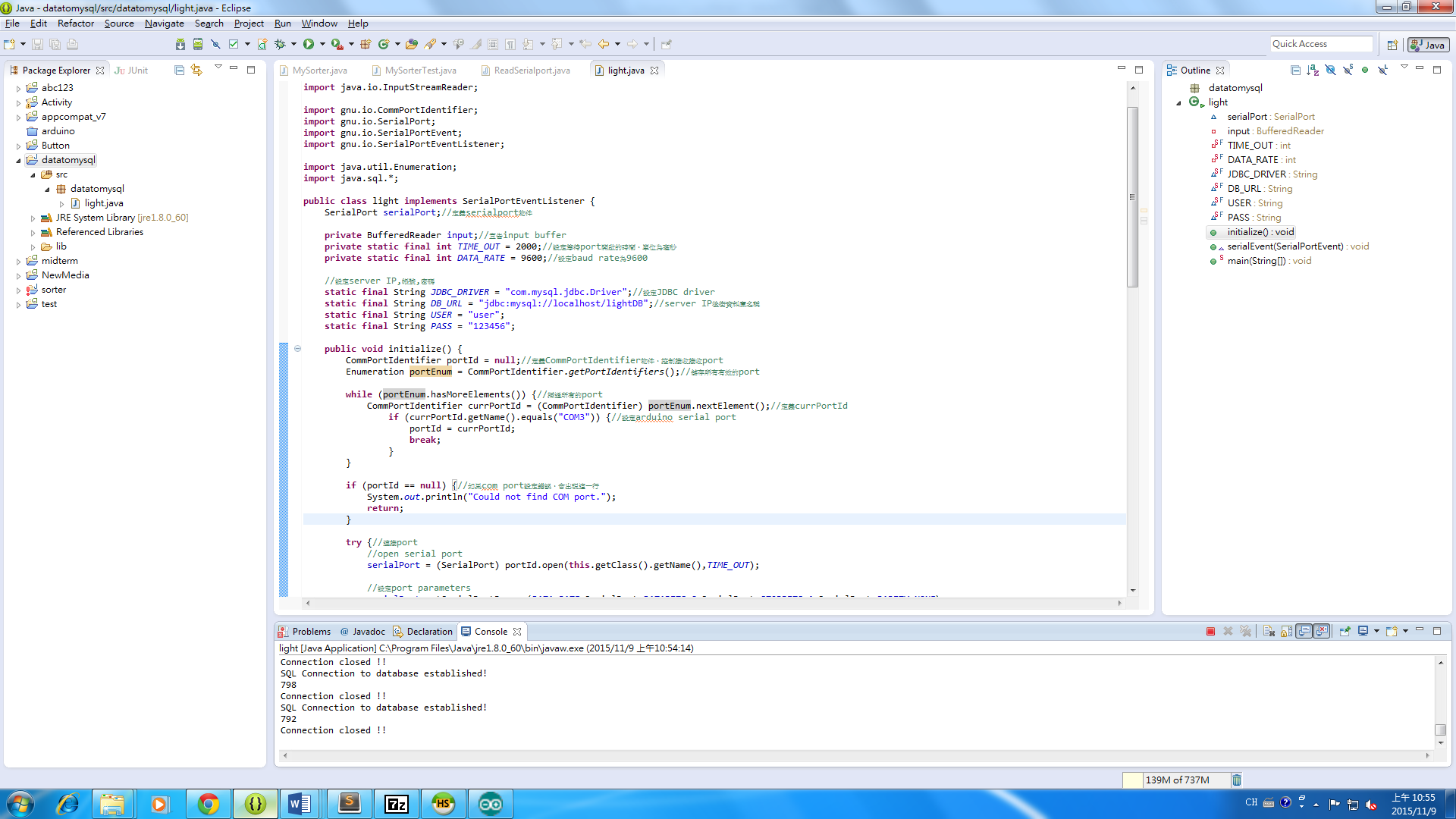
}

System.*out*.println("Started");

}

}

輸出結果(JAVA)



輸出結果(資料庫)

