In this tutorial I am going to show how-to communicate a Java program with an Arduino board through the serial port.

**Getting started**

Before start, be sure you have installed Java JDK correctly. You can read this blog to do it. Once this is done, check the Java compiler version.

```
$ javac -version
javac 1.8.0_171
```

**Installing jRXTX library**

I work with jRXTX. This library is native in Ubuntu and you have to install following these steps:

Install the library from repositories.

```
$ sudo apt install librxtx-java
```

This will install the correct native library in **/usr/lib/jni** and the jar library in **/usr/share/java**

```
$ ls /usr/lib/jni
librxtxI2C-2.2pre1.so
librxtxRaw-2.2pre1.so
librxtxSerial-2.2pre1.so
librxtxI2C.so
librxtxRaw.so
librxtxSerial.so
librxtxParallel-2.2pre1.so
librxtxRS485-2.2pre1.so
librxtxParallel.so
librxtxRS485.so
$ ls /usr/share/java
. . .
RXTXcomm-2.2pre2.jar
RXTXcomm.jar
. . .
```

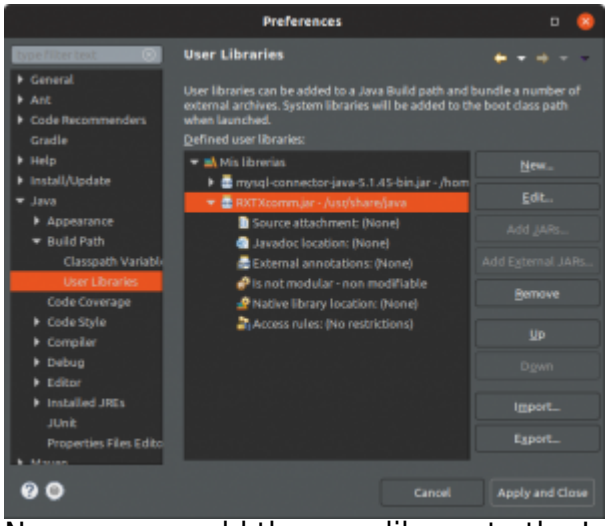
The serial ports /dev/tty* are only accessible to members belonging to groups **dialout** and **tty**. You therefore have to add your user to those groups

```
$ sudo adduser <user> dialout
$ sudo adduser <user> tty
```

Afterwards you have to **reboot** your computer in in order for the group changes to become active.

**Configure Eclipse**

In order to use the library jRXTX on Eclipse IDE. You have to add it to user library.




Now, ensure add the user library to the Java project.

**The Java program**

I used the same java program showed in rxtx example page to test the library.

```
package serial;

import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;

import gnu.io.CommPort;
import gnu.io.CommPortIdentifier;
import gnu.io.SerialPort;

public class TwoWaySerialCom {
    public TwoWaySerialCom() {
        super();
    }

    void connect(String portName) throws Exception {
        CommPortIdentifier portIdentifier =
CommPortIdentifier.getPortIdentifier(portName);
        if (portIdentifier.isCurrentlyOwned()) {
            System.out.println("Error: Port is currently in use");
        } else {
            CommPort commPort =
portIdentifier.open(this.getClass().getName(), 2000);

            if (commPort instanceof SerialPort) {
                SerialPort serialPort = (SerialPort) commPort;
                serialPort.setSerialPortParams(9600,
SerialPort.DATABITS_8, SerialPort.STOPBITS_1,
                        SerialPort.PARITY_NONE);

                InputStream in = serialPort.getInputStream();
                OutputStream out = serialPort.getOutputStream();
```

```
                (new Thread(new SerialReader(in))).start();
                (new Thread(new SerialWriter(out))).start();

            } else {
                System.out.println("Error: Only serial ports are
handled by this example.");
            }
        }
    }

    /** */
    public static class SerialReader implements Runnable {
        InputStream in;

        public SerialReader(InputStream in) {
            this.in = in;
        }

        public void run() {
            byte[] buffer = new byte[1024];
            int len = -1;
            try {
                while ((len = this.in.read(buffer)) > -1) {
                    System.out.print(new String(buffer, , len));
                }
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }

    /** */
    public static class SerialWriter implements Runnable {
        OutputStream out;

        public SerialWriter(OutputStream out) {
            this.out = out;
        }

        public void run() {
            try {
                int c = ;
                while ((c = System.in.read()) > -1) {
                    this.out.write(c);
                }
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
```

```
    public static void main(String[] args) {
        try {
            (new TwoWaySerialCom()).connect("/dev/ttyACM0");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

**Arduino program**

I created a basic arduino program to respond a string received by serial port.

```
/**
 * Arduino serial test
 * Author: Rodrigo Tufino <rtufino@ups.edu.ec>
 * Date: 26-06-2018
 */
void setup() {
  Serial.begin(9600);
  Serial.println("Serial Echo Device");
  delay(1000);
}

void loop() {
  delay(100);
  if (Serial.available() > ) {
    String dato = Serial.readString();
    Serial.println(dato);
  }
}
```

**Test**

```
$ java serial.TwoWaySerialCom
```

```
Serial Echo Device
Hello
Hello

how are you?
how are you?
```

**Working in Microsoft® Windows®**

You need two files to use this library on Microsoft Windows:
rxtxSerial.dll
rxtx-2.2pre2.jar
You have to copy the **rxtxSerial.dll** file to system folder **C:\winsows\system32.**
The **rxtx-2.2pre2.jar** file is the library to add to Java's project. You can download these archives from here.