**Administrator Guide**

**jWebSocket**

**Arduino Remote Control Demo**

**Dariel Noa Graverán**

**Version history?**

1. **Process application download**

The application is divided in two parts, the application client, and the server, it is also necessary to download and install the native library that allows that the server to write data in the port USB. The following list shows where to download the mentioned components:

|  |  |  |
| --- | --- | --- |
| **Package** | **Size** | **Link** |
| Arduino Remote Control PlugIn | **78 KB** | [Download](https://jwsdev.org:9443/svn/jWebSocket/branches/jWebSocket-1.0/jWebSocketPlugIns/jWebSocketArduinoPlugIn/) |
| Arduino Remote Control Client | **126 KB** | [Download](https://jwsdev.org:9443/svn/jWebSocket/branches/jWebSocket-1.0/jWebSocketClient/web/demos/jwsArduino/) |
| Native library rxtx | **-** | [Download](http://rxtx.qbang.org/wiki/index.php/Download) |
| Arduino Environment | **-** | [Download](http://arduino.cc/en/Main/Software) |
| Arduino Program | **4 KB** | [Download](https://jwsdev.org:9443/svn/jWebSocket/branches/jWebSocket-1.0/arduino/rc_demo) |

1. **Characteristics of the installation environment**

The application should be executed in an environment that fulfills the following requirements:

* Operating system Windows XP or higher, GNU Linux x86/x64 or Mac OSX.
* Java Runtime Environment 7. Java 1.6 not supported anymore? OpenJDK 7 ?
* A port free USB.
* Apache Web Server or similar (it is not necessary to install PHP).
* Native library rxtx 2.1.7 or 2.2, for the communication with the USB port.
* The Arduino micro-controller should have housed in its memory the program with which one will work.

1. **Installation process**

First of all you must copy the client application into the root directory of the web server. This is necessary because some browsers require a http:// site and cannot establish a websocket connection, when the site is started using the file:// protocol. For development purposes in the configuration of the server the application client e.g. can be under the URL: http://localhost.

Then you proceed to indicate to the virtual machine of Java the files that it should use so that the application can manage the port, to accomplish that you must copy the files of the native libraries for the serial port control in the binary folder of JRE (Java Runtime Environment): No good English!

To allow the jWebSocketArduinoPlugIn to communicate with the Arduino Hardware you need to install the native rxtx library. This allows the plug-in to communicate via a serial port which is mapped to the USB port for Arduino. You must copy the files of the native libraries for the serial port control in the binary folder of JRE (Java Runtime Environment) as follows:

* Windows: Copy the files rxtxParallel.dll and rxtxSerial.dll in the location C:/Program Files/Java/jre7/bin/
* Linux: Copy the file librxtxSerial.so in the location /jre/lib/, below the folder of the Java Virtual Machine.
* Mac OS X: Copy the file librxtxSerial.jnilib in the folder /Library/Java/ Extensions/

Subsequently you should connect the circuit Arduino to a USB port, then verify once connected that all the earth (GND) indicators are turned on. For the development of the solution the Arduino Mega ADK was used, however for other types of circuits of the platform Arduino, the behavior would be same. It is necessary to have 4 LEDs of blue, red, green and yellow colors that will be connected to the entrance / exit (I/O) pines 12, 8, 7 and 4, also you must have a joystick of two connections corresponding to the coordinates (x, y), which will be connected to the entrance pin 0 (I0) the connector x and to the entrance pin 1 (I1) the connector y. Once connected the joystick you must verify that the indicative earth (GND) are turned on. To facilitate the connections it would be very good to have a Tinker-Kit, this is convenient to connect the printed circuit board (PCB) with the LEDs and the joystick.

To run the application it is not necessary to have the source code of the project, you will have a file of type .jar that will be executed by console. This file after being downloaded can be copied in any directory.

1. **Hardware**

The application has as fundamental requirement the existence of a micro-controller Arduino Mega ADK. This device is connected to the PC by USB port, so it is necessary to install drivers so that the application can manage it.

**Installation of the drivers in Windows**

Once the PCB has been connected in Windows, the system will show the assistant to add new hardware. Here you must indicate not to connect to Windows Update, and later click next.

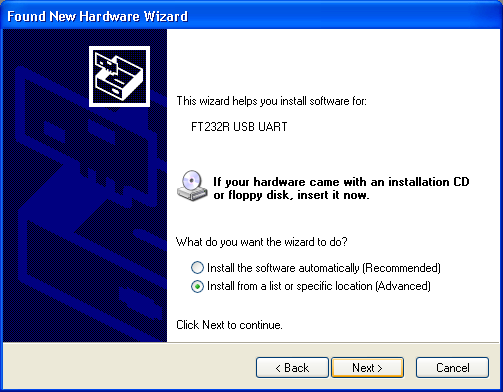
Important: we will specify only the installation on Windows, to install Arduino in other operating systems find documentation at <http://arduino.cc>.



|  |
| --- |
|  |

*Ref. to Fig.1: Assistant for New Hardware in MS-Windows. Step 1.*

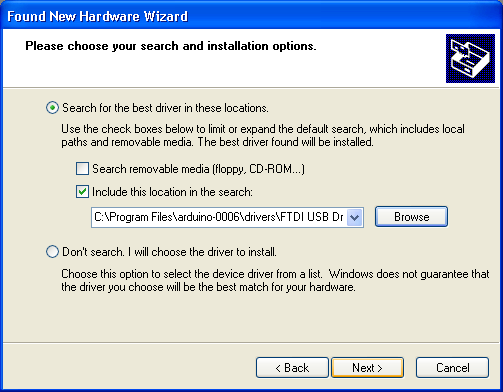
In the following window you must select: Install from a list or specific (Advanced) location and then click next.



|  |
| --- |
|  |

*Ref. to Fig.2: Assistant for New Hardware in MS-Windows. Step 2.*

Select Search for the best driver in these locations. This location will be the directory where the downloaded driver was stored.



|  |
| --- |
|  |

*Ref. to Fig.3: Assistant for New Hardware in MS-Windows. Step 3.*

The assistant will indicate that it found Arduino Mega ADK, then give click in Finish.



|  |
| --- |
|  |

*Ref. to Fig.4: Assistant for New Hardware in MS-Windows. Step 4.*

Then search in the devices manager which is the name of the port that has assigned Arduino Mega ADK.

1. **Configuration Options**

The application has only a point for its configuration (not completely true, you missed the jWebSocket.xml where the plug-in needs to be registered, please add that! This is essential); it consists on a XML file that is the file associated to the one plug-in: rc.xml on which the port is specified where they should send and receive the data. This file is located in the address:

|  |
| --- |
| *$JWEBSOCKET\_HOME/conf/EventsPlugIn/rc-application/app-plugins/rc.xml* |

When opening the file with a text editor you must modify the line 8 specifying which will be the port where Arduino is connected, example:

|  |
| --- |
| *<constructor-arg><value>****COM3****</value></constructor-arg>* |

In Windows the port would be: COM0. . ., COM3; in Linux: /dev/tty/USB0, /dev/tty/USB1 or similar and in MAC it is probably: /dev/tty.usbserial-1B1.

Where can I see which ports are possible, especially in Linux and Mac OS? Please explain how I can figure that out.

1. **Setting in operation**

The reader will also expect some words here about the Arduino program, including some screenshots with a description how to upload. BTW: this is also a good location to describe how to figure the existing port, which need to be configured in the the rc.xml! Please extend the Administrator Guide into this direction!

After the program is in the PCB and to have configured the port, you proceed to execute the server for the console by the following command:

|  |
| --- |
| *java -jar $JWEBSOCKET\_HOME/libs/jWebSocketServer-Bundle-1.0.jar* |

Once the server is executing, you proceed to begin the client application from his local URL, by a web navigator that has support for WebSocket. In the superior right part it should indicate that the connection settled down with the server, if this occurs the application is ready to be tested.

1. **Application Management**

For the administration of the Arduino Remote Control Demo application you proceed in the same way that EventsPlugIn is configured in the jWebSocket server (this is not enough! Here the reader needs the configuration in the jWebSocket.xml and an explanation of the options, especially since there are a few plug-in specific settings). In case of any problem during the installation process and/or configuration of the application you can consult to:

**Dariel Noa Graverán**

**Mail:** dariel87@gmail.com