**Installation guide**

**JWebSocket**

**Arduino Remote Control Demo**

1. **Process application download**

The application is divided in two parts, the application client, and the server, it is also necessary to discharge the native library that allow that the servant can write data in the port USB, subsequently they will be been able to discharge the mentioned elements:

|  |  |  |
| --- | --- | --- |
| **Package** | **Size** | **Link** |
| Arduino Remote Control Server | **83.9 Mb** | [Download](http://repo.hab.uci.cu/svn/tesis/Segundo_Corte_de_Tesis/JWS/Dariel_Noa/2do_Corte/Implementacion/server-side) |
| Arduino Remote Control Client | **6.84 Mb** | [Download](http://repo.hab.uci.cu/svn/tesis/Segundo_Corte_de_Tesis/JWS/Dariel_Noa/2do_Corte/Implementacion/client-side/demos) |
| Native library rxtx | **2.04 Mb** | [Download](http://repo.hab.uci.cu/svn/tesis/Segundo_Corte_de_Tesis/JWS/Dariel_Noa/2do_Corte/Implementacion/native-library) |
| Drivers Arduino Mega ADK | **4 Kb** | [Download](http://repo.hab.uci.cu/svn/tesis/Segundo_Corte_de_Tesis/JWS/Dariel_Noa/2do_Corte/Implementacion/driver-ArduinoMegaADK/Arduino_ADK.zip) |

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.
* 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 port USB.
* The Arduino micro-controller should have housed in its memory the program with which one will work.

1. **Installation process**

Him first that it should be made, it is to copy the application client toward the directory root of the servant web. This is necessary because the used version of jWebSocket, he/she needs that the client this under an address web, since this validated by defect, in the configuration of the servant that the application client this under URL: http://localhost.

Then you proceeds to indicate to the virtual machine of Java that files should use so that the application can manage the port, for it they should copy you the files of the native bookstores of control of serial port for the portfolio of binary of JRE (Java Runtime Environment):

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

Subsequently you should connect the circuit Arduino to a port USB, to verify once connected that all the earth (GND) indicators are lit. For the development of the solution Arduino Mega ADK was used, however for other types of badge 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 a joystick of two connections corresponding to the coordinates (x, y) should respectively also be had, which will be connected to the entrance pin 0 (I0) the connector x and to the entrance pin 1 (I1) the connector and. Once connected the joystick to verify that the indicative earth (GND) is lit. To facilitate the connections it would be very good to have a TinkerKit, This is good to connect to the circuit the LEDs and the joystick.

To begin to walk the application it is not necessary to have the source code of the project, that is to say he/she will have a file of type jar that will be executed by means of the console. This file after being discharged 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 means of the port USB, for what is necessary to install drivers so that the application can manage it.

**Installation of the drivers in Windows**

Once the badge has been connected in Windows, the system showed the assistant to add new hardware. Here we indicate that it is not connected Windows Update, and later we give him/her following.

Important: alone the installation was specified in Windows, to install Arduino in other operating systems, to be documented in the place <http://arduino.cc>.



Fig. 3.1: Assistant for New Hardware in MS-Windows. Step 1.

In the following window to select: To install from a list or specified (Advanced) location and then click in following.



Fig. 3.2: Assistant for New Hardware in MS-Windows. Step 2.

To mark to look for the drivers from a location in specific. This location will be the directory where is the discharged driver.



Fig. 3.3: Assistant for New Hardware in MS-Windows. Step 3.

The assistant indicated that he found Arduino Mega ADK, then to give click in concluding.

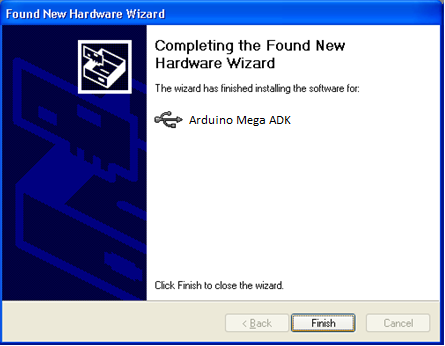


Fig. 3.4: Assistant for New Hardware in MS-Windows. Step 4.

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

1. **Configuration Options**

The alone application has a point for its configuration, it consists on a XML file, that is to say the file associated to the one plug-in: rc.xml in which the port is specified where they should be sent and to 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 he/she should modify the line 8 specifying which it will be the port where it is connected Arduino, 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.

1. **Setting in operation**

After the program is in the badge and to have configured the port, you proceeds to execute the servant for the mediating console the following command:

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

Once the servant is executing, you proceeds to begin the application client from his local URL, by means of a navigator web that has support for websocket. In the superior right part it should indicate that the connection settled down with the servant, if this already happens the application this list to be proven.

1. **Application Management**

For the administration of the application Arduino Remote Control Demo you comes in the same way in that EventsPlugIn is configured in the servant jWebSocket. In the event of happening any problem during the installation process and/or configuration of the application can consult to:

**Dariel Noa Graverán**

**Mail:** dariel87@gmail.com