# User Guide

**JWebSocket**

**Arduino Remote Control Demo**

**Dariel Noa Graverán**

**1. Characteristics of the solution**

The remote control device using the framework jWebSocket, allows manipulation of an electronic module in real time, coupled with the benefits provided by the Web.

Usually remote control systems based on the Web possess a client server structure, following the principle request - reply. These systems establish communication using the HTTP protocol, which ensures that an order is sent from the controller, in this case the client to the server, which is associated with the device. However, the use of this protocol for the exchange of information has disadvantages: the first is that the driver is always the one who has to initiate the communication process, to wait for the response of the device; otherwise the communication establishing is not in real time, which prevents you from controlling some devices that require high precision time.

The demo application of remote control via web, developed with the framework jWebSocket and Arduino hardware platform, allows manipulating the turning on and off of 4 LEDs in blue, red, green and yellow. It also transmits the movements of a physical joystick to the controller application.

This demonstration of manipulation brings the possibility of controlling hardware in real time using jWebSocket, which can then be used in some remote scenarios, such as: high precision equipment, the camera control robotic, pan-tilt heads and associated with medical devices.

The application also shows the type of communication established, the controller is not forced to initiate the communication process, allowing the exchange of data bi-directionally. The aspects mentioned before: real-time two-way communication, encourage a new paradigm of communication on the web. This is accomplished by use of the jWebSocket framework, a new technology aimed to the developing of web applications on the communication protocol WebSocket.  
Today there are solutions that control a device remotely via the web, but these applications do not use the WebSocket protocol for communication, which provides high levels of security, scalability and speed. Comparisons between HTTP and WebSocket show that the reduction of network traffic has a rate of 500 to 1, bearing in mind that setting the WebSocket communication between the client and the server there is only one shipment of 2 bits, eliminating the HTTP headers. This ensures speed in communication, allowing the creation of real-time applications using WebSocket protocol.

**2. Main functionalities**

The demo application for controlling remote devices, developed with jWebSocket framework, has the following features:

* Turn On/Off, 4 physical LEDs.
* Monitor the physical position of the joystick.
* Send data to the micro-controller.
* Receive data from the micro-controller.

The solution can be adapted to use in different processes that need to control a device through the web, and more so when this control requires to be in real time. Specifically, this solution can be modified to control lights instead of LEDs, also programming Arduino micro-controller circuit can be changed and adapted to manipulate devices such as motors, air conditioners, televisions, finally any device that contains controls of click type for its manipulation. It is also viable to use it as a base to create applications for home automation, solutions that handle security cameras, monitoring sensors, robotics, or at any stage that requires control a device in real time using a web application.

**3. Problems to be solved**

|  |  |
| --- | --- |
| **Problem** | **Contributions of the solution** |
| The remote control device via the web is not performed in real time. | The system allows remote control of 4 LEDs in real time, yet it can be easily adapted for controlling and monitoring devices located in different sectors of society. |
| The communication established for the control and monitoring devices is not bidirectional. | The remote control system developed in the framework jWebSocket ensures two-way communication, allowing the driver is not who initiates the communication process. |

**4. Glossary of Terms**

**Remote Control:** Device that regulates at distance the operation of a device, mechanism or system.

**Device:** Mechanism or willing artifice to produce a foreseen action.

**Real time on the web:** Group of technologies and practices that allow the users to receive information as soon as it is published by their authors, instead of checking a source of information periodically.

**Websocket:** The protocol websocket defines the procedures to upgrade the connection through HTTP to a connection by means of completely bidirectional websocket using TCP. The client sends a petition HTTP GET to establish a communication WebSocket with the server. Later on the communication remains active until he/she closes, allowing exchanging messages between the client and the server.

**JWebSocket:** It is a new technology guided to the development of applications based on websocket that provide high levels of speed, scalability, security and the work in real time, key element for the web nowadays.

**Arduino:** It is a platform of development of physical calculation (physical computing) of open code, based on a badge with a simple micro-controller and a development environment to create software that then will be gone up to the badge.

**5. Model of the solution**

|  |
| --- |
| [Model of the solution](Images/Model_of_the_Solution.png) |

*Ref. to Fig.1: Model of the solution*

To carry out the remote control using the application, the user should have a navigator web that supports the protocol websocket that allows him to consent to the controller application. After establishing connection with the jWebSocket server, which possesses the necessary elements to consent to the controller interface of the platform Arduino; you can manipulate the turning on or out of the 4LEDs that are connected to the micro-controller, one also allows the monitoring of the movements of the physical joystick that it is also connected to the circuit. For the control of the LEDs an event rushes from the controller towards the jWebSocket server, and it sends towards the circuit a command, indicating which LED to turn off or on. For the monitoring of the joystick the circuit sends towards the jWebSocket server the position, then an event rushes to each one of the controllers, to visualize in real time the position of the joystick.

**7. Requirements for the use**

The application has the advantage of working on the operating systems Linux, Windows and Mac OS, however for its total operation he/she needs of the following elements:

* OpenJDK 1.7 installed.
* Native library RxTx copied in the binary folder on the Java Virtual Machine (JVM).
* Circuit micro-controller of the hardware platform Arduino.
* Browser with support for the protocol websocket.

To obtain the packages of the controller application, the jWebSocket server, the native library for serial port access, and the program that contains the circuit micro-controller, you should access the URL: <http://jwebsocket.org/download/>

**8. Roles of the solution**

**User of the system:**  It can remotely control the turning on and off of 4 LEDs also it allows monitoring of the joystick movements.

**9. System Operation**

|  |
| --- |
| [Main view](Images/Main_view.PNG) |

*Ref. to Fig.2: Main view*

1. This part of the application represents the movements of the joystick.
2. In the bar shows the status of the application (on / off), the identifier of the client and the type of connection established with the server (native / flash-bridge).
3. In the text area, is disclosed in which fields you can use a remote control system in real-time via the web.
4. In the section represent the LEDs, which are connected to the Arduino micro-controller circuit to effect on and off of them.

**9. Configurations of the solution**

To run the application, you must specify which port is connected to the micro-controller circuit; this requires access to the configuration of the plug-in Arduino located in the file:

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

Then you must specify the file where the port is connected to the microcontroller circuit.

|  |
| --- |
| [Ref. to Port settings in the rc.xml file](Images/PlugIn_Port_Configuration.PNG) |

*Ref. to Fig.3: Port settings in the rc.xml file*

This configuration can be performed by system users who can access the server files jWebSocket.

**10. Rules of the solution**

**R1:** If the computer running the server jWebSocket is not connected to the Arduino circuit, or the port is incorrectly configured, the application throws an error message indicating that the Arduino circuit is not available. The user before accessing the application controller should verify that the jWebSocket server starts with the connected circuit, and that the port configuration is correct.