# User Guide

**JWebSocket**

**Arduino Remote Control Demo**

**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, client server have a structure, following the first 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 one who has to initiate the communication process, so wait for the response of the device, otherwise the communication establishing 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, to manipulate the on and off 4 LEDs in blue, red, green and yellow. And transmits the movements of a physical joystick 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, ie the controller is not obliged to initiate the communication process, allowing the exchange of data bi-directionally. The aforementioned aspects: real-time two-way communication, encourage a new paradigm of communication on the web. This is accomplished by use of the framework jWebSocket, a new technology aimed at developing 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 a1, 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 WebSocket protocol.

**2. Main functionalities**

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

* 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 for use in different processes that need to control a device through the web, and more so when this control that is required in real time. Specifically, this solution can be modified to control lights instead of LEDs, plus programming Arduino micro-controller circuit can be changed and adapted to manipulate devices such as motors, air conditioners, televisions in order to contain any device type controls here to handling. It is also feasible to use as a base to create applications for home automation solutions that handle security cameras, monitoring sensors, robotics, in order 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 4 LEDs in real time, yet 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 an apparatus, mechanism or system.

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

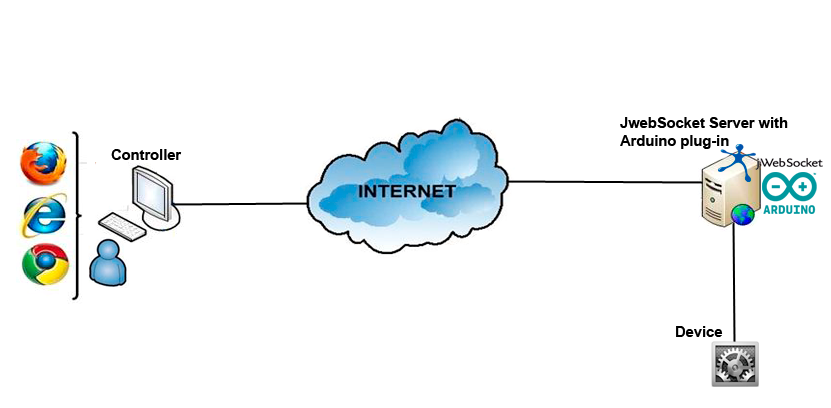
**Real time on the web:** Group of technologies and practical that they allow the users to receive information as soon as it is published by their authors, instead of checking a source of information periodically.

**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 servant. Later on the communication remains active until he/she closes, allowing to exchange messages between the client and the servant.

**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**

****

To carry out the remote control using the application, the user should have a navigator web that supports the protocol websocket that will allow him to consent to the controller application. After establishing connection with the servant jWebSocket, which possesses the necessary elements to consent to the controller interface of the platform Arduino; you can manipulate the ignition and out of the 4LEDs that are connected to the micro-controller, one also can monitoring 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 toward the servant jWebSocket, and this he/she sends toward the circuit a command, indicating which LED to turn off or to light. For the monitoring of the joystick the circuit sends toward the servant jWebSocket 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:

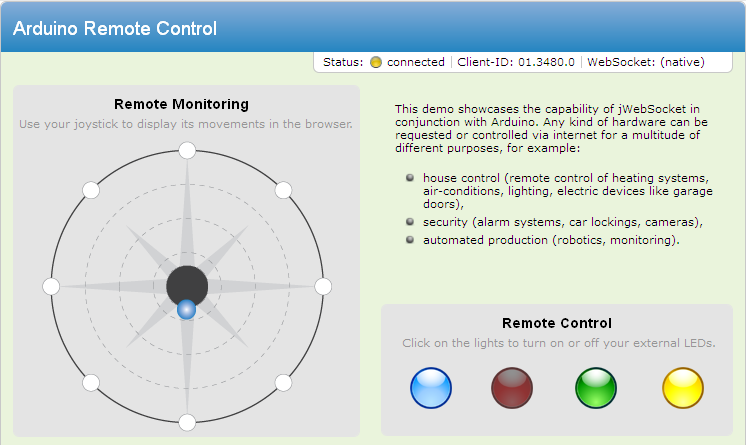
* OpenJDK 1.7 installed.
* Native library RxTx copied in the binary folder on the Java Virtual Machine(JVM).
* Circuit micro-controller of the hardware platform Arduino.
* Browser with support for the protocol websocket.

To obtain the packages of the controller application, the servant jWebSocket, 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 on and off 4 LEDs also can monitor the movements of the joystick.

**9. System Operation**

****

**2**

**3**

**4**

**1**

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.

**Port**

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, not connected the Arduino circuit, or incorrectly configured port, 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 server jWebSocket start with the circuit connected, and that the port setting is correct.