



CONTACT SUPPORT:

COMPANY NAME:	Control Concepts Inc.
SUPPORT CONTACT:	Elizabeth Scozzari
EMAIL ADDRESS:	support@controlconcepts.net
PHONE:	(201) 797-7900
ADDRESS:	336 Route 46, Fairfield, NJ 07004

GENERAL INFORMATION

SIMPLWINDOWS NAME:	CCI Websocket Server Lite
CATEGORY:	Utility
VERSION:	1.0.1
SUMMARY:	<p>Note: this is the "Lite" version of the WebSocket server with a limited number of signals and limited number of concurrent connections supported. The "Full" version of this module has the ability to support far more signals of each type and multiple connections (i.e. panels) per server. Contact Control Concepts for more information if you are interested in the full version.</p> <p>This is a Utility module that implements a RFC-6455 WebSocket server capable of having a single connection per instance. The WebSocket server allows for two-way communication to HTML5 compatible web pages. This WebSocket server utilizes a sub-protocol to allow for input/output control into and out of your Crestron 3 Series controller. The sub-protocol allows up to 500 digital, 100 analog, and 100 string input/output signals. A simple HTML5 webpage has been provided as an example (Websocket Webpage Example.html).</p> <p>This WebSocket server is compatible with the following web browsers.</p> <ul style="list-style-type: none"> Internet Explorer 10 Firefox 11 (PC) Firefox 11 (Android) Chrome 16 (PC, Mobile) Safari 6 (Mac, iOS*) Opera 12.10 (PC, Mobile) Android Browser (Not Supported) <p>*iOS must be at least version 6.</p>
GENERAL NOTES:	<p>This module utilizes SIMPL# and is only compatible with 3-Series controllers.</p> <p>To learn more about what other utility modules are available from Control Concepts visit the CCI Utility Module Store.</p>
CRESTRON HARDWARE REQUIRED:	3-Series controller with Ethernet connectivity
SETUP OF CRESTRON HARDWARE:	Ethernet Connectivity
VENDOR FIRMWARE:	n/a
VENDOR SETUP:	n/a
CABLE DIAGRAM:	TCP/IP



PARAMETERS:

IP Port	A	IP Port the WebSocket will listen on. (<i>Must be an available port on the 3-Series controller.</i>)
---------	---	--

CONTROL:

<i>Signal/Function Name</i>	<i>D,S,A</i>	<i>Digital, Serial, Analog signal property definition.</i>
fb*	D	<p>Digital Feedback Signals 1-500.</p> <p>When a signal goes high, it causes a sub-protocol message of <i>ON[<signal_ID>]</i> to be sent into the HTML5 websocket.onmessage event handler of your webpage.</p> <p>When a signal goes low, it causes a sub-protocol message of <i>OFF[<signal_ID>]</i> to be sent into the HTML5 websocket.onmessage event handler of your webpage.</p> <p><i>Examples (fb201):</i> ON[201] OFF[201]</p>
an_fb*	A	<p>Analog Feedback Signals 1-100.</p> <p>When a signal changes, it causes a sub-protocol message of <i>LEVEL[<signal_id>,<value>]</i> to be sent into the HTML5 websocket.onmessage event handler of your webpage</p> <p><i>Example (an_fb50):</i> LEVEL[50,100] LEVEL[50,65535]</p>
text-o*	S	<p>String Output Signals 1-100</p> <p>When a signal changes, it causes a sub-protocol message of <i>STRING[<signal_id>,<value>]</i> to be sent into the HTML5 websocket.onmessage event handler of your webpage</p> <p><i>Example (text-o20):</i> STRING[20,Hello World!] STRING[20,SIMPL# Rocks!]</p>
Start_Server	D	Pressing this signal will cause the WebSocket server to start listening for a client connection.
Stop_Server	D	Pressing this signal will disconnect any WebSocket clients and stop the server from listening for any additional clients.

FEEDBACK:

press*	D	<p>This digital signal will go high when a sub-protocol message of <i>PUSH[<signal_id>]</i> is sent out from your webpage utilizing the <i>websocket.send()</i> method.</p> <p>This signal will go low when a sub-protocol message of <i>RELEASE[<signal_id>]</i> is sent out from your webpage utilizing the <i>websocket.send()</i> method.</p> <p><i>Example (press101):</i> PUSH[101] RELEASE[101]</p>
an_act*	A	<p>This analog signal will change when a sub-protocol message of <i>LEVEL[<signal_id>,<value>]</i> is sent out from your webpage utilizing the <i>websocket.send()</i> method.</p> <p><i>Example (an_act51):</i> LEVEL[51,255] LEVEL[51,100]</p>
text-i*	S	<p>This string signal will change when a sub-protocol message of <i>STRING[<signal_id>,<data>]</i> is sent out from your webpage utilizing the <i>websocket.send()</i> method.</p> <p><i>Example (text-i32):</i> STRING[32,Testing 123] STRING[32,PIN-1234]</p>
Server_Is_Running	D	Indicates high when the server is running.
Client_Is_Connected	D	Indicates high when a client is connected and successfully authenticated using RFC-6455 rules.



TESTING:

OPS USED FOR TESTING:	CP3: 1.601.3935.27221
SIMPL WINDOWS USED FOR TESTING:	4.14.20.00
DEVICE DB USED FOR TESTING:	200.05.001.00
CRES DB USED FOR TESTING:	200.00.004.00
SYMBOL LIBRARY USED FOR TESTING:	1114
SAMPLE PROGRAM:	CCI Websocket Server v1.0 Demo
REVISION HISTORY:	1.0.0 – Initial Release 1.0.1 – Change default port and removed fixed server IP address