How To Monitor & Control ControlByWeb™ Devices From An Independent Web Server

Overview:
ControlByWeb™ devices can be easily be monitored and controlled using their built-in web servers and default web pages. It may be desirable however, to design custom web pages to control one or more ControlByWeb™ devices. This can be accomplished by using a stand alone web server (a web server other than the one that is built into the device). In this scenario, custom web pages are installed on the web server, and the user views the web pages directly from the stand-alone web server. The web pages control the device by sending simple commands directly to it. In addition, the custom web pages can display the status of the relays, inputs, sensors, etc. Status information is obtained by periodically reading an XML page from device which contains this information.

Disclaimer:
This example illustrates one way to control ControlByWeb™ devices from web pages that are installed on a stand alone web server. There are many ways to accomplish this task and this example illustrates only one of those ways. This example is provided for reference only and is given without warranty and is not supported by Xytronix Research & Design, Inc.

How It Works:
This example uses a combination of html, javascript, and xml to create a web page that monitors and controls a ControlByWeb™ device. We first create the layout of the web page using html. For this example we have created a basic table that displays the status of the device, similar to the control page index.html that we can retrieve directly from the device itself. We then use javascript to communicate with the ControlByWeb™ device. To retrieve the state of the device we request the file state.xml from the devices built in web server and then parse through this information, updating our web page as necessary. This is handled by the function getState(), which is called once when the web page is first opened, and then repeatedly called at the interval specified by the variable refreshTimeSec which by default is set to 3 seconds.

To send control messages to the device we have created another function called setState(relayNumber, state). To change the state of a relay on a ControlByWeb™ device, this function can be called in response to a button being pressed or a link being clicked. This function takes two parameters, the relay number which we want to control and the new state of the relay. The relay on a single relay device is numbered 0, while devices with multiple relays start numbering the relays at 1. Valid commands that can be sent to the relays are:

0 – Turn relay off
1 – Turn relay on
2 – Pulse relay
5 – Toggle relay state

How To Use:
To get started, find the html file in this zip file that matches your device. For example, if the device to by monitored is a WebRelay-Dual, then choose the file webrelay-dual.html. By default all the examples are configured to communicate with a ControlByWeb™ device at the ip address 192.168.1.2 with a refresh rate of 3 seconds. To change the ip address and refresh rate find the lines at the top of the html

file that look like this:

```
// specify the ip address of the webrelay-dual here
var ipaddress = "192.168.1.2";
// specify the automatic refresh rate here (in seconds)
var refreshTimeSec = 3;
```

Here you can change the ip address to match the ip address assigned to the device to be monitored. The refresh rate can also be changed from its default of 3 seconds. Once the ip address has been configured try opening the html file in a web browser. Depending on the security settings of the web browser, you might be prompted to allow the running of scripts. Allow scripts to run in this case. If you get more prompts saying permission denied then read the section 'Permission Denied – Cross-Domain XMLHttpRequest Calls' below.

You now have a basic web page that will read the status of the ControlByWeb™ device and control any relays on the device.

Permission Denied: Cross-Domain XMLHttpRequest Calls

Most new web browsers have added security measures that stop XMLHttpRequest calls from requesting information from a different web server than the one where the web page originated from. In other words, if the file webrelay-dual.html is opened from a directory on a local machine, the web browser will complain if the javascript in that file tries to request information from a different location other than that same machine. This is good for security, but it makes it difficult for our custom web pages to function correctly. Fortunately many web pages use the same techniques we have incorporated into our custom web pages and therefore there are workarounds.

Workaround 1 – Lower the security settings of the web browser.

Most web browsers will allow you to adjust the security settings to allow our custom web page to function. This is probably the easiest method to get the page to work, although it sacrifices security.

Workaround 2 – Use a Web-Proxy

This workaround is a little more involved, but more secure. It is definitely the way you would ultimately want to use if this custom web page were to be placed on a web server for others to view and use. With this method, a small custom application or script is created to communicate with the ControlByWeb™ device. This script is then placed on the same server as our custom web page and we modify our custom web page to send requests to the custom script instead of directly to the device. The script then takes these requests, sends them to the ControlByWeb™ device, and returns the response to our custom web page. Since our custom web page and the script reside on the same server the web browser does not complain.

Included in this zip file is a directory called proxy. In this directory you will find examples that make use of a proxy. The only difference between these examples and the others is that the examples in the proxy directory request information from the proxy instead of directly from the device. To test these examples place the html files and php file on your web server in the same directory as each other. Locate the line

```
// hostname
```

```
define('HOSTNAME', 'http://xx.x.xx.xx:xxxx/');
```

in the file proxy.php and replace xx.x.xx.xx with the ip address and port number of the device you are going to monitor. For this to work, make sure your web server is able to locate the ControlByWeb™ device.

For each device a separate proxy file should be created and configured. The name of the file can also be changed. (If you are going to monitor multiple devices you will have to rename each proxy script.) If the name of the proxy script is changed, then the html file that is to use this proxy must be configured as well. To tell the html file which proxy script to use find the lines:

```
// name of the proxy script
var proxyName = "proxy.php";
```

at the top of the html file and replace proxy.php with the new name of the script file.