How To Monitor & Control WebRelay™ from An Independent Web Server

Note: This example works with Internet Explorer only!

Overview:
WebRelay can be easily be monitored and controlled using its built-in web server and default web pages. It may be desirable however, to design custom web pages to control one or more WebRelay devices. This can be accomplished by using a stand alone web server (a web server other than the one that is built into WebRelay). 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 WebRelay by sending simple commands directly to WebRelay. In addition, the custom web pages can display the status of the relay and/or the input. Status information is obtained by periodically reading an XML page from WebRelay.

Disclaimer:
This example illustrates one way to control WebRelay 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:
Three files are installed on the WebServer (demo.html, webrelay.html, and junk.html). The demo is started by pointing a web browser to demo.html. Demo.html creates two frames. One frame fills almost the entire window and displays webrelay.html. The other frame is named 'trash'. The trash frame is very small so that it is not visible and initially contains the file junk.html. The trash frame is very useful for controlling WebRelay. Each time webrelay.html submits a command to WebRelay, the command is submitted through the trash frame instead of directly from the main frame. The purpose for submitting commands through the trash frame is to prevent the response from WebRelay from being displayed in the main frame. This leaves the main frame that displays webrelay.html un-interrupted. The status of the input and the relay is read every two seconds from WebRelay. The status is obtained by requesting state.xml from WebRelay and parsing the XML file. The function loadDoc() reads the xml file every two seconds. The function UpdateValues() parses the XML file and causes the current status to be displayed on the web page.

How To Use:
Change the IP address in webrelay.html to the IP address of the WebRelay unit that you wish to control. Note that there are four places to change the IP address. Install all three html files on the web server. Run the example by pointing your web browser to demo.html.