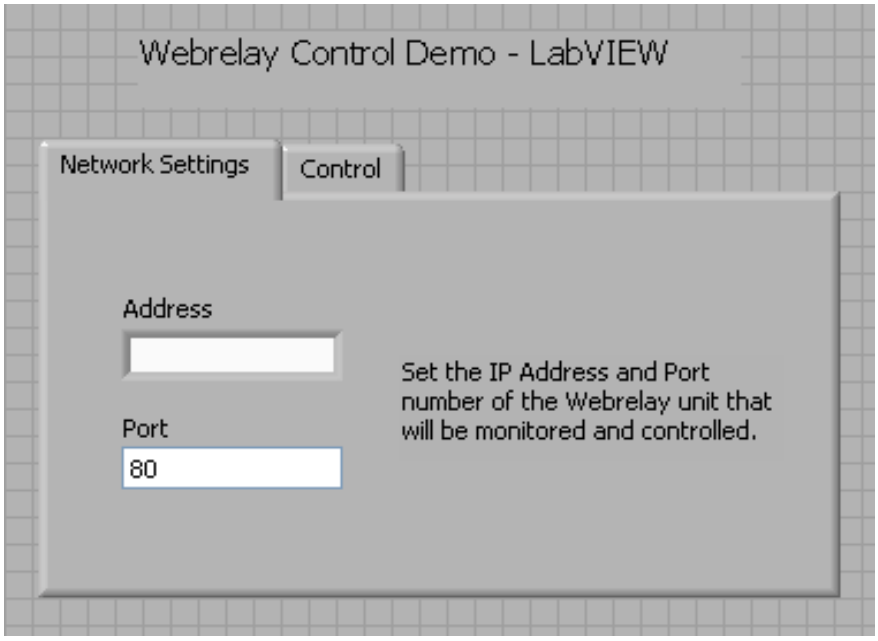
How to control/monitor Webrelay™ with LabVIEW™ 8.0

1. Introduction

Webrelay can be monitored and controlled through the use of TCP and XML. TCP is used to create a network connection with the Webrelay unit, and XML is the language used to communicate with it. LabVIEW 8.0 is capable of handling both of these protocols/languages, making it a convenient way to create applications that are capable of controlling and monitoring the Webrelay unit. The following document explains how we were able to create the sample VI webrelay.vi to control and monitor Webrelay.

2. The Front Panel



The following controls were used on the front panel for this example.

Tab Control:

This divides the network settings from the actual relay control GUI.

String Control:

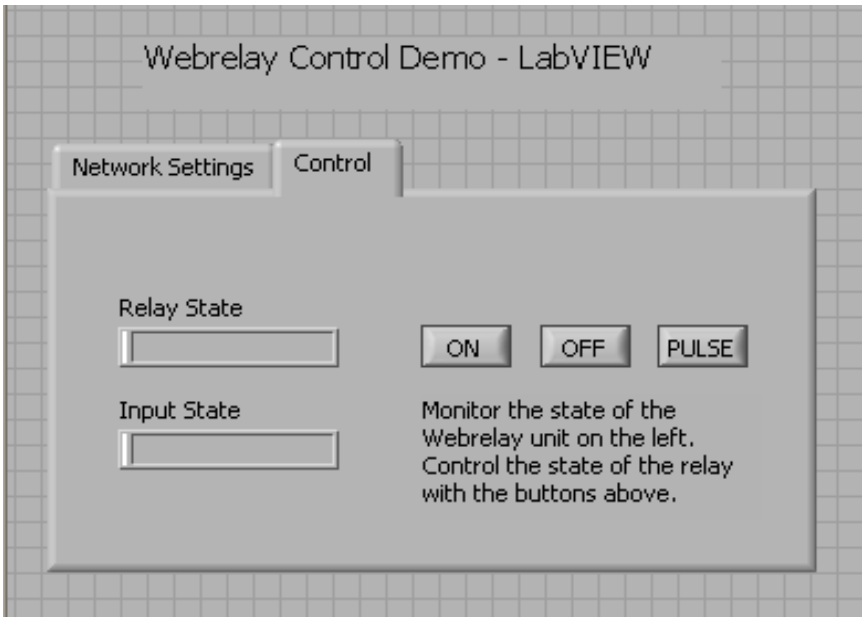
This control is used to get the IP Address of the Webrelay unit to be monitored/controlled.

System Numeric Control:

This control is used to get the port number of the Webrelay unit to be monitored and controlled.

System Labels

The labels in this example are there to give basic instruction about the use of the application.



String Indicators

The string indicators display the current status of the Webrelay units relay and input. If errors occur, they are displayed in these controls as well.

Text Buttons

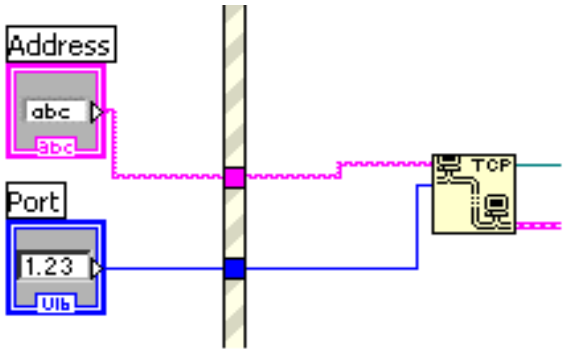
These buttons signal events in the Block Diagram, which in turn send XML messages to the Webrelay unit instructing it to turn ON/OFF, or PULSE. The properties of each of these buttons were set such that the on and off colors were the same, and the button behaviors were set to switch when pressed.

3. The Block Diagram

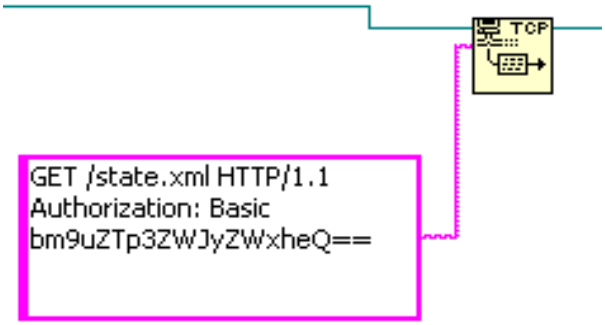
There are three main Programming Structures used in the Webrelay example: the While Loop, the Event Structure, and the Case Statement. The While Loop encapsulates everything else in the Block Diagram. It allows the program to run continually until the user stops it.

Inside the While Loop there is an Event Structure. The first event of the Event Structure is the Timeout event. This event is setup to occur every 500ms, at which time the status of the Webrelay unit is read, and the String Indicator controls on the Front Panel are updated. The other three Events of the Event Structure occur when one of the Text Buttons on the Front Panel are pressed. When one of the button pressed events occur, the TCP connection procedure is started. Upon success, the appropriate XML message is sent to the Webrelay unit, a response is received, the String Indicators are updated, and the TCP connection is closed.

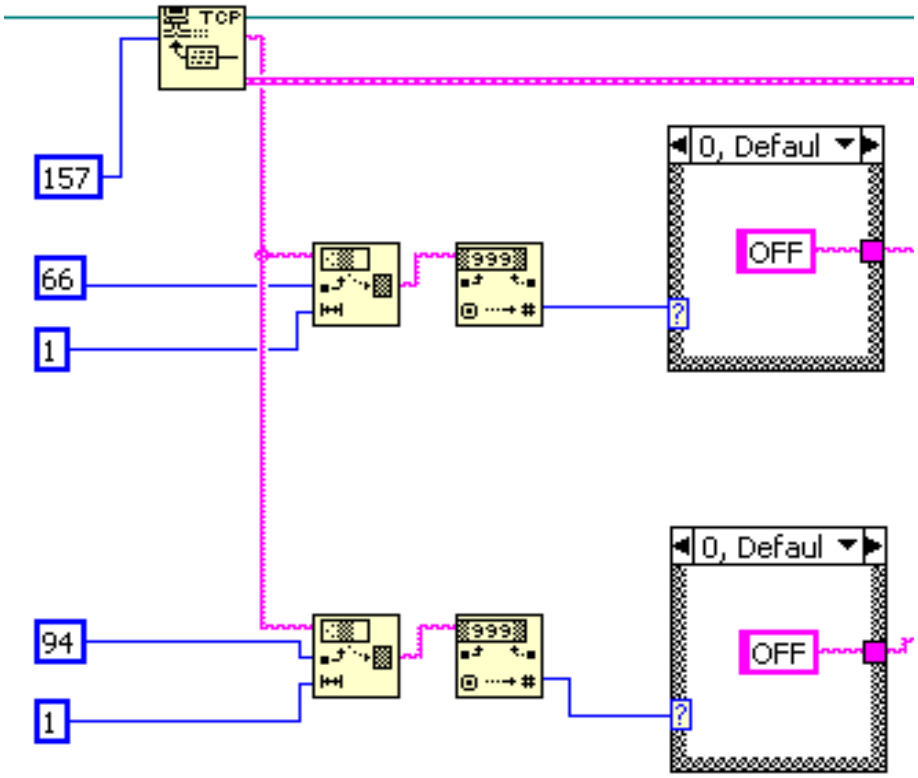
To open a TCP connection, the TCP Open Connection VI is used. This VI allows a TCP connection to be established with a server at a specified IP Address and Port number. In our case, the address and port come from the Network Settings tab on the Front Panel of the project, and reflect the IP Address and Port number of the Webrelay unit being monitored.



To send a XML message to the Webrelay unit, the TCP Write VI is used. It takes the connection ID from the TCP Open Connection VI, and the actual message string to be sent as inputs.



To wait for a XML response from the Webrelay unit, the TCP Read VI is used. For this example, we expect to receive a response consisting of 157 bytes. The TCP Read VI allows us to connect a numeric constant defining the number of bytes to be read. If the number of bytes to be read isn't received, the TCP Read VI will timeout automatically after 25000ms. Once a response has been received, the XML is parsed using the String Subset VI, and the results are converted to integers using the Decimal String To Number VI. These integers are then passed to case structures which send a "ON" or "OFF" string to the String Indicators on the Front Panel.



4. Format of XML request.

Note: The following notation is used to indicate a carriage return and line feed in the XML requests.

<CR> = Carriage return (hex 0d)
<LF> = Line feed (hex 0a)

These characters aren't normally visible, but they must be present for correct operation of the Webrelay unit.

There are 4 different XML request that are recognized by Webrelay :

1. To retrieve the state of the Webrelay unit, send the following

GET /state.xml?noReply=0 HTTP/1.1<CR><LF>
Authorization: Basic bm9uZTp3ZWJyZWxheQ==<CR><LF><CR><LF>

2. To turn the relay of the Webrelay unit on, send the following

GET /state.xml?relayState=1 HTTP/1.1<CR><LF>
Authorization: Basic bm9uZTp3ZWJyZWxheQ==<CR><LF><CR><LF>

3. To turn the relay of the Webrelay unit off, send the following

GET /state.xml?relayState=0 HTTP/1.1<CR><LF>
Authorization: Basic bm9uZTp3ZWJyZWxheQ==<CR><LF><CR><LF>

4. To pulse the relay of the Webrelay unit, send the following

GET /state.xml?relayState=2 HTTP/1.1<CR><LF>
Authorization: Basic bm9uZTp3ZWJyZWxheQ==<CR><LF><CR><LF>

Note: The Second Line of each of the XML requests above specifies the password to be used. The password needs to be in base64 form. For example, the password "webrelay" is "bm9uZTp3ZWJyZWxheQ==" in base64 form. To encode a password, go to the webpage http://www.controlbyweb.com/encoder.html.

5. Format of XML response

Each XML response from Webrelay is 157 bytes long. Below is an example XML response from a webrelay unit whose input is off, and relay state is on.

```
<?xml version='1.0' encoding='utf-8'?>
<datavalues>
<relaystate>1</relaystate>
<inputstate>0</inputstate>
<rebootstate>0</rebootstate>
</datavalues>
```

Each XML response begins with a header that indicates the version of XML used and the character encoding used. Next, there is a tag labeled <datavalues>. It is in between the tags <datavalues> and </datavalues> that the state of Webrelay can be found. The tags <relaystate></relaystate> surround the byte that signifies the relay state of the webrelay unit.

1 = RELAY ON
0 = RELAY OFF

Similarly, the tags <inputstate></inputstate> surround the byte that signifies the input state of the webrelay unit.

1 = INPUT ON
0 = INPUT OFF

The relay state is found at an offset of 66 bytes from the beginning of the response. The input state is found at an offset of 94 bytes from the beginning of the response.