How to Remotely Control AC Power for a WEB Server Using an Acromag ES2113 or a 989EN-4C16 Ethernet Digital I/O Module

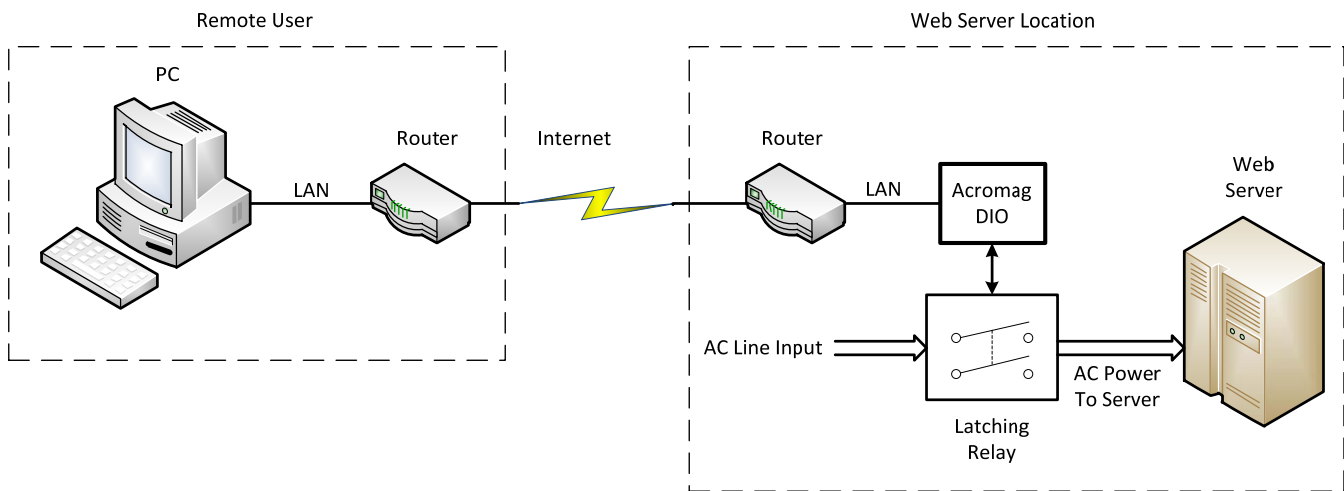# How to Remotely Control AC Power for a WEB Server Using Acromag Ethernet

Objective

This document outlines the steps taken to use an ES2113 or 989EN DIO to control a latching relay that is used to open or close the AC power (115 VAC) path used by a multi domain WEB server. In this mode of control the DIO is configured to a static IP address and is therefore accessible from anywhere on the Internet. The web server described in this outline has been setup to automatically boot up at power on. The ES2113 might be viewed as major overkill for this project, however with a little ingenuity the user could easily increase the number of relays and use the same circuitry to control house lighting both inside and out as well as putting some of the ports to use to return digital environmental data monitored at the server site (which in my case is my house) all while on vacation.

Overview

System Control Overview

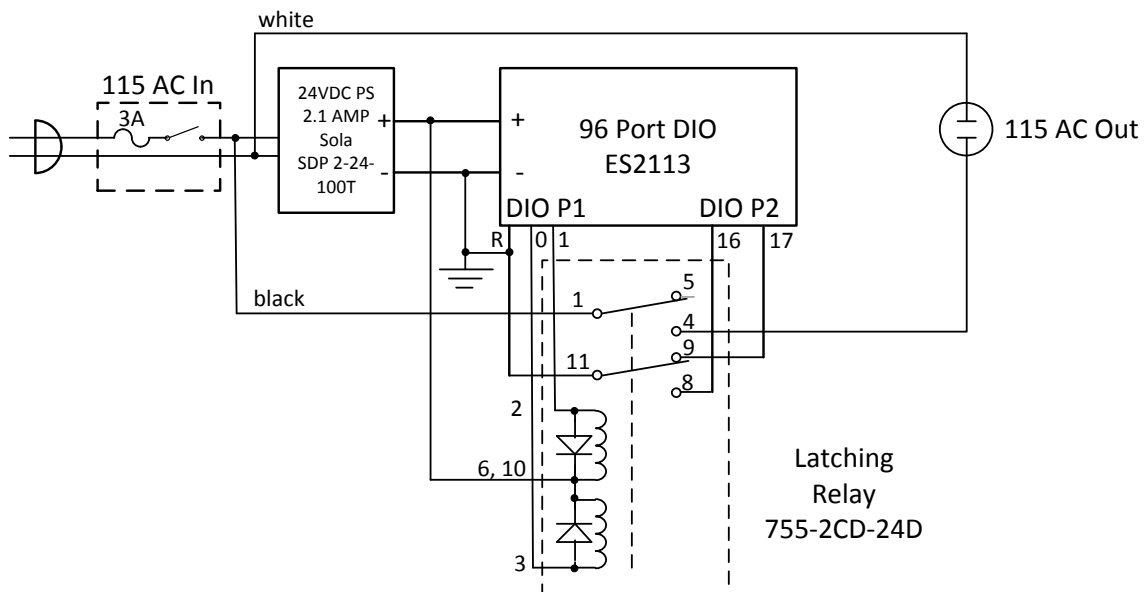The following diagram (figure 1) provides the schematic for the AC Control System.


Figure 1 - AC power control schematic

In the above diagram the user has made use of two ports when using the ES2113. Port 1 is being used to control the Set and Reset coils in the latching relay and Port 2 is being used for reading back the telemetry contacts in the latching relay. If a 989EN DIO is used the same functionality can be accomplished using a single 16 bit port in which case the telemetry would be monitored on bits 14 and 15 of the port and the user would mask for these two bits to read telemetry from the single 16 bit port.
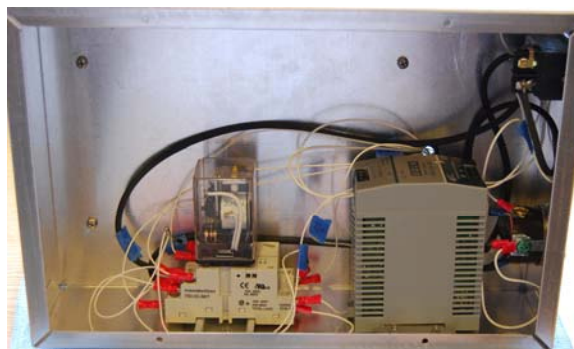
The following images show the finished prototype used to develop and test the Server AC Power Control Circuitry.
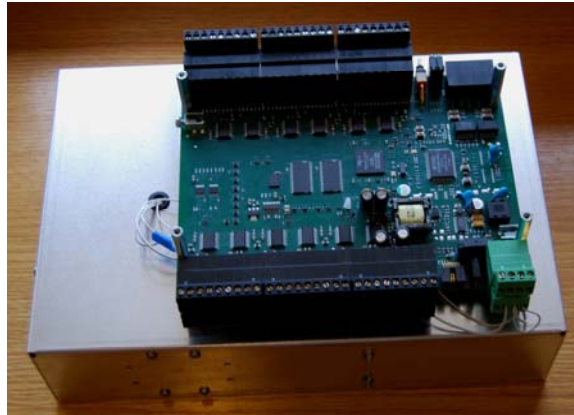
Figure 2 – Top view showing mounting of ES2113
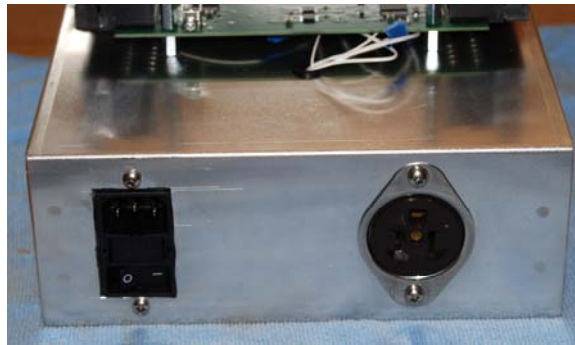
Figure 3 – End view showing the AC input & output connectors

Figure 4 – Under chassis view showing DIN mounted latching relay & 24 VDC power supply

Control Software
The DIO software control was developed using Microsoft Visual Studio using a large quantity of C and C++ for all of the DIO read and write functions including a C++ library of ModBus calls to communicate with the Acromag DIO. After all the software function calls were tested out and debugged, a C# windows application was developed to permit user friendly interaction that involved a drop down menu selection for performing all of the user available functions. These front panel controls and indicators are shown and described in the following User Interface Section.

User Interface Section

IP Address

Figure 5 below shows the front panel with the DIO IP Address input control outlined in red. **NOTE:** The IP address show in this image is the default IP address for the DIO and is NOT the final Static IP address that was assigned to the control system and has been changed for obvious security reasons.
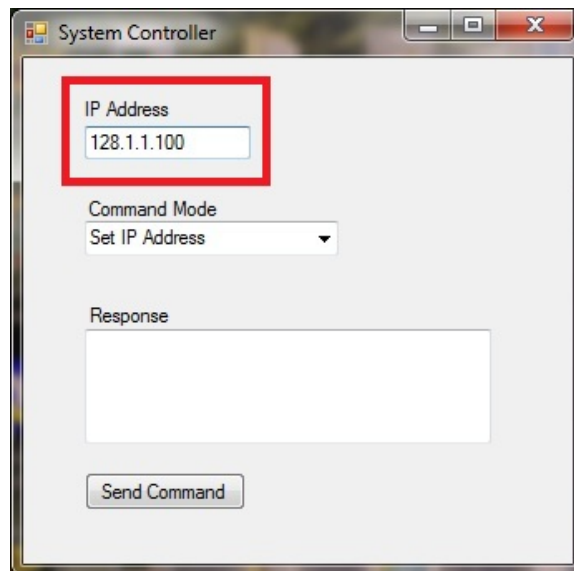
Figure 5 – Front Panel view showing control input for DIO IP Address

The IP Address has been set to the Static default value in the software and always comes up showing this default value when this system controller is started. The IP Address control window was added to allow the user to change the IP Address if for any reason the IP address were changed in the future. This is explained further in the section for the Command Mode.

Command Mode

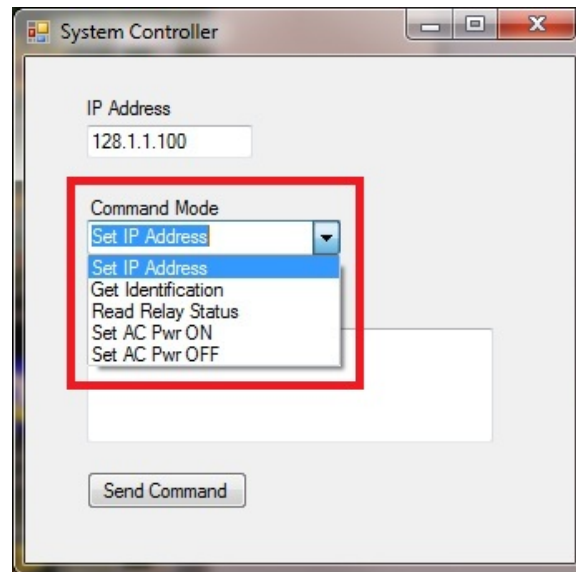Figure 6 below shows the front panel with the Command Mode drop down menu activated and outlined in red.

Figure 6 – Front Panel view of Command Mode Control with menu choices displayed

The above view shows the five commands modes displayed in the drop down menu selection for this control. The five commands are outlined in detail below:

- **Set IP Address** – When this menu choice is selected the IP Address displayed in the "IP Address" window will be used to communicate with the DIO when the "Send Command" button is clicked. This command mode is only required when the IP Address has been set by the user to a value other than the Default value which is displayed in the "IP Address" window when the System controller is first executed.
- **Get Identification** – When this menu choice is selected the DIO identification string will be requested from the DIO and returned in the "Response" window when the "Send Command" button is clicked. The typical response from this request (for a ES2113 DIO) will be "ACROMAG,9300-170C,ES2113-xxxx232996F". This is an excellent method to determine if your software has a functional tcp/ip connection with the DIO.
- **Read Relay Status** – When this menu choice is selected the present position of the latching relay is returned to the user in the "Response" window when the "Send Command" button is clicked. The returned status will be one of two text strings listed below:
    - **Telem: AC Power ON**
    - **Telem: AC Power OFF**
- **Set AC Pwr ON –** When this menu choice is selected a command is written to the DIO that will place the latching relay into the position to apply AC power to the server. As part of this command the telemetry contacts on the latching relay are read back and the present status of the latching relay will be displayed in the "Response" window. The response should be the following: **Telem: AC Power ON**
- **Set AC Pwr OFF -** When this menu choice is selected a command is written to the DIO that will place the latching relay into the position to remove AC power from the server. As part of this command the telemetry contacts on the latching relay are read back and the present status of the latching relay will be displayed in the "Response" window. The response should be the following: **Telem: AC Power OFF**

Response

Figure 7 below shows the front panel with the Response window location outlined in red.
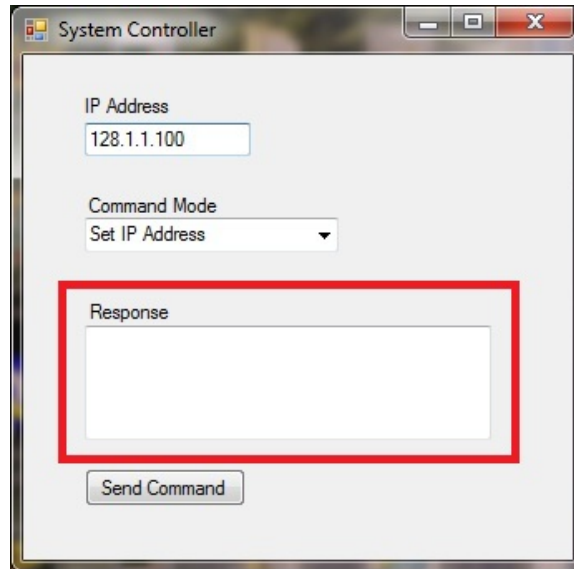
Figure 7 – Front Panel view of Response window

The response window is where all software generated text messages are displayed. Messages are associated with the following commands:

- Get Identification
- Read Relay Status
- Set AC Pwr ON
- Set AC Pwr OFF

Send Command

Figure 8 below shows the front panel with the Send Command button location outlined in red.
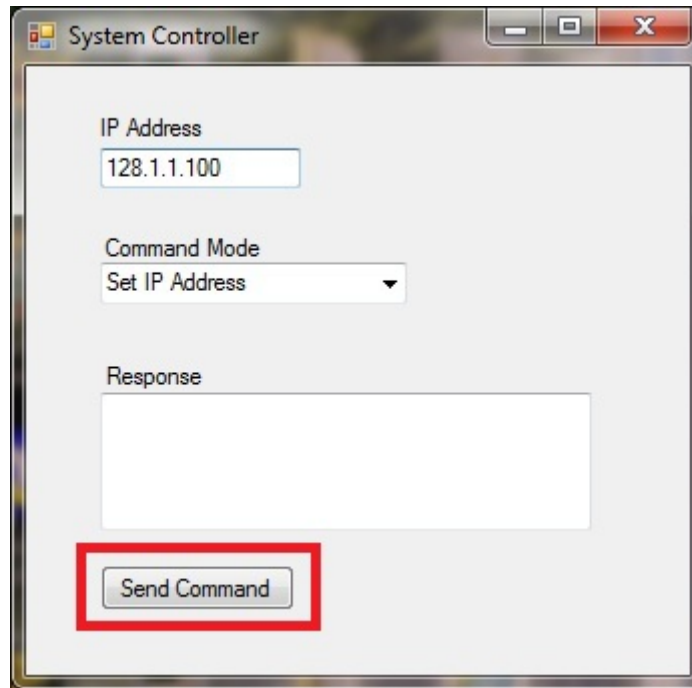
Figure 8 – Front Panel view of Send Command button

The Send Command button generates an "on click" event when the button is left clicked with the user's mouse. This event then reads the command mode menu selection and executes the software associated with the selected mode.

Software
The following C# software is associated with the front panel user interface

```csharp
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;




namespace WindowForm
{
  public partial class Form1 : Form
  {
    public Form1()
    {
      InitializeComponent();
      cmbBx1.SelectedIndex = 0;
      txtBx1.Text = "128.1.1.100";
    }
    private void btn1_Click(object sender, EventArgs e)
```

```csharp
{
    TheCore.ACPwr_CLR.ACPwr rfsw = new TheCore.ACPwr_CLR.ACPwr();
    string strAddress = txtBx1.Text;
    Object item = cmbBx1.SelectedItem;
    String caseStr = item.ToString();

    switch (caseStr)
    {
        case "Set IP Address":          // Set IP Address
        {
            strAddress = txtBx1.Text;
            if (strAddress == "")
                txtBx2.Text = "failed - IP Address=NULL";
            else
                txtBx2.Text = "IP Address set to: " + strAddress;
        }
        break;
        case "Get Identification":      // Get Acromag ID String
        {
            string id = null;
            int status = rfsw.ID(strAddress, ref id);
            if (status != 0)
                txtBx2.Text = "failed";
            else
                txtBx2.Text = id;
        }
        break;
        case "Read Relay Status":       // read status of requested ports
        {
            string settings = "All";
            string telemetry = null;
            int status = rfsw.GETSW(strAddress, settings, ref telemetry);
            if (status != 0)
                txtBx2.Text = "failed";
            else
            {
                string Prompt = String.Format(telemetry);
                switch (Prompt)
                {
                    case "10":
                    {
                        txtBx2.Text = "Telem: AC Power ON";
                    }
                    break;
                    case "01":
                    {
                        txtBx2.Text = "Telem: AC Power OFF";
                    }
                    break;
                    case "00":
                    {
                        txtBx2.Text = "Telem: 00 - Indicates Relay Failure";
                    }
                    break;
                    case "11":
                    {
                        txtBx2.Text = "Telem: 11 - Indicates Relay Failure";
                    }
                    break;
```

```csharp
          }
        }
      }
    break;

    case "Set AC Pwr ON":          // SET SWITCHES, READ STATUS
      {
        string settings = "ON";
        string telemetry = null;
        int status = rfsw.SETSW(strAddress, settings, ref telemetry);
        if (status != 0)
          txtBx2.Text = "failed";
        else
        {
          string Prompt = String.Format(telemetry);
          switch (Prompt)
          {
            case "10":
              {
                txtBx2.Text = "Telem: AC Power ON";
              }
              break;
            case "01":
              {
                txtBx2.Text = "Telem: AC Power OFF";
              }
              break;
            case "00":
              {
                txtBx2.Text = "Telem: 00 - Indicates Relay Failure";
              }
              break;
            case "11":
              {
                txtBx2.Text = "Telem: 11 - Indicates Relay Failure";
              }
              break;
          }
        }
      }
    break;
    case "Set AC Pwr OFF":          // SET SWITCHES, READ STATUS
      {
        string settings = "OFF";
        string telemetry = null;
        int status = rfsw.SETSW(strAddress, settings, ref telemetry);
        if (status != 0)
          txtBx2.Text = "failed";
        else
        {
          switch (telemetry)
          {
            case "10":
              {
                txtBx2.Text = "Telem: AC Power ON";
              }
              break;
            case "01":
              {
```

```
                    txtBx2.Text = "Telem: AC Power OFF";
                }
              break;
            case "00":
                {
                    txtBx2.Text = "Telem: 00 - Indicates Relay Failure";
                }
              break;
            case "11":
                {
                    txtBx2.Text = "Telem: 11 - Indicates Relay Failure";
                }
              break;
            }
          }
        }
      break;
        }
      }

    }
}
```

How to Remotely Control AC Power for a WEB Server Using Acromag Ethernet

Flow Chart

```
                          ┌─────────────────┐
                          │  Initialize S/W │
                          └─────────────────┘
                                   │
                          ┌─────────────────┐
                   ┌─────▶│    "START"      │◀─────┐
                   │      └─────────────────┘      │
                   │               │               │
                   │          ◇ Send Cmd ◇   no     │
                   │          ◇ Btn Pressed? ◇──────┘
                   │               │ yes
                   │      ┌─────────────────┐
                   │      │ Get Command Mode│
                   │      └─────────────────┘
```

Set IP Address?
- no → branches to Get Identification
- yes → Read IP Address Input

IP Address Empty?
- yes → Return "Failed" Err Msg
- no → Write IP Address into Local variable "strAddress" to be used By all other mode calls

Get Identification?
- no
- yes → Pass ID request to C++ subroutine

1. Open Socket to DIO using "strAddress".
2. Format ID request into ModBus command and write to DIO using TCP/IP

1. Read back Identification string from DIO.
2. Pass Identification string back to user interface S/W.

Identification string Empty?
- yes → Return "Failed" Err Msg
- no

Read Relay Status?
- no
- yes → Pass Read Relay Status request to C++ subroutine

1. Open Socket to DIO using "strAddress".
2. Format Read Relay Status request into ModBus command and write to DIO using TCP/IP

1. Read back Relay Status string from DIO.
2. Pass Relay Status string (telemetry contacts) back to user interface S/W.

Relay Status string Empty?
- yes → Return "Failed" Err Msg
- no

Set AC Pwr ON?
- no
- yes → Pass Set ON Relay request to C++ subroutine

1. Open Socket to DIO using "strAddress".
2. Format ON Relay request into ModBus command and write to DIO using TCP/IP

1. Read back Relay Status string from DIO.
2. Pass Relay Status string (telemetry contacts) back to user interface S/W.

Relay Status string Empty?
- yes → Return "Failed" Err Msg
- no

(Set AC Pwr OFF)
Pass Set OFF Relay request to C++ subroutine

1. Open Socket to DIO using "strAddress".
2. Format OFF Relay request into ModBus command and write to DIO using TCP/IP

1. Read back Relay Status string from DIO.
2. Pass Relay Status string (telemetry contacts) back to user interface S/W.

Relay Status string Empty?
- yes → Return "Failed" Err Msg
- no

Return to "START"

Application note courtesy of Dan Weise, Lesman Instrument Company

Acromag
THE LEADER IN INDUSTRIAL I/O

About Acromag

Acromag is a leading manufacturer of temperature transmitters and signal conditioners for use with thermocouple and RTD sensors. These instruments convert the sensor input to a proportional 4-20mA DC current, DC voltage, Modbus, Profibus, or Ethernet signal to interface with a PC, PLC, DCS, or other control equipment.

Acromag has designed and manufactured measurement and control products for more than 60 years. They are an AS9100 and ISO 9001-certified international corporation with a world headquarters near Detroit, Michigan and a global network of sales representatives and distributors. Acromag offers a complete line of industrial I/O products including a variety of process instruments, signal conditioners, and distributed fieldbus I/O modules. Industries served include chemical processing, manufacturing, defense, energy, and water services.

For more information about Acromag products, call the Inside Sales Department at (248) 295-0880. E-mail sales@acromag.com or write Acromag at 30765 South Wixom Road, Wixom, MI 48393-7037 USA. The web site is www.acromag.com.