



Creating a Web Based DM1800 Application

This application note will show you how to use the DMWEB.DLL to allow you to make a web based ASP web page(s) for your custom application. The DMWEB.DLL allows you to use all the features of the DM1800 without having to write a large program. It also enables you to quickly modify your custom web page for other applications without having to rewrite a large program.

ASP is a script based HTML web page language. It will run on any server locally including being able to run on an Internet Information Services (IIS) which allow you to turn any Windows based PC running XP Pro into an internet server. We will discuss how to set your XP PC up to use this feature later in this application note.

The DMWEB.DLL consists of 32 object commands that will allow you the user to communicate to the DM1800 Base and Field Nodes easily including opening and closing the communications comm. port.

The following is a list of the DMWEB.DLL commands and a brief description of there function.

Communication Commands

<u>Command</u>	<u>Description</u>
CloseCom(n)	Closes active com port, n is don't care
OpenCom(n)	Opens Com port n, valid port number 1-8
ComStat(n)	Returns status of com port, n is don't care

Counter Commands

<u>Command</u>	<u>Description</u>
Config_Count_Both(n)	Set counter to increment on both edges, n = node
Config_Count_Dn(n)	Set counter to increment on falling edge, n = node
Config_Count_Up(n)	Set counter to increment on rising edge, n = node



Binary Input Event Commands

<u>Command</u>	<u>Description</u>
Config_Event_Both(n)	Set counter to send a message to the base on <u>both</u> edges, n = node number
Config_Event_Dn(n)	Set counter to send a message to the base on <u>falling</u> edge, n = node number
Config_Event_Up(n)	Set counter to send a message to the base on <u>rising</u> edge, n = node number

Uart Event Commands

<u>Command</u>	<u>Description</u>
Config_Uart_Event_1200(n)	Set Node(n) Uart to 1200 bps and enable event message to base on receipt of data present at Uart
Config_Uart_Event_2400(n)	Set Node(n) Uart to 2400 bps and enable event message to base on receipt of data present at Uart
Config_Uart_Event_4800(n)	Set Node(n) Uart to 4800 bps and enable event message to base on receipt of data present at Uart
Config_Uart_Event_9600(n)	Set Node(n) Uart to 9600 bps and enable event message to base on receipt of data present at Uart

Uart Commands

<u>Command</u>	<u>Description</u>
Config_Uart_1200NE(n)	Set Node(n) Uart to 1200 bps, n = node number
Config_Uart_2400NE(n)	Set Node(n) Uart to 2400 bps, n = node number
Config_Uart_4800NE(n)	Set Node(n) Uart to 4800 bps, n = node number
Config_Uart_9600NE(n)	Set Node(n) Uart to 9600 bps, n = node number



A/D Commands

Command

Read_AD_8Bit(n)

Description

Read the A/D converter on node n, with A/D resolution of **8 bits**, data will be received by a Read_Com(n) command

Read_AD_10Bit(n)

Read the A/D converter on node n, with A/D resolution of **10 bits**, data will be received by a Read_Com(n) command

Binary Input Command

Command

Read_Binary_In(n)

Description

Read binary input pin and report logic state of node(n), data will be received by a Read_Com(n) command, n = node number

Binary Output Commands

Command

Read_Binary_Out(n)

Description

Read binary output pin and report logic state of node(n) data will be received by a Read_Com(n) command, n = node number

Set_Binary_On(n)

Set the binary output pin High on node n

Set_Binary_Off(n)

Set the binary output pin Low on node n

Set_Binary_Default_Hi(n)

Set power on **default** binary output High for node(n)

Set_Binary_Default_Low(n)

Set power on **default** binary output Low for node(n)

Read Com Commands

Command

Read_Com(n)

Description

Read Com port data, n = don't care



User Data Commands

Command

Description

Set_User_A(n,s)

Set user data register A on node n with data string in s, **max** length is 16 bytes

Set_User_B(n,s)

Set user data register B on node n with data string in s, **max** length is 16 bytes

Read_User_A(n)

Read user data register A on node n, data will be received by a Read_Com(n) command

Read_User_B(n)

Read user data register B on node n, data will be received by a Read_Com(n) command

System Commands

Command

Description

Reset_Node(n)

Reset Node n

Reset_Base(n)

Reset Base, n = don't care

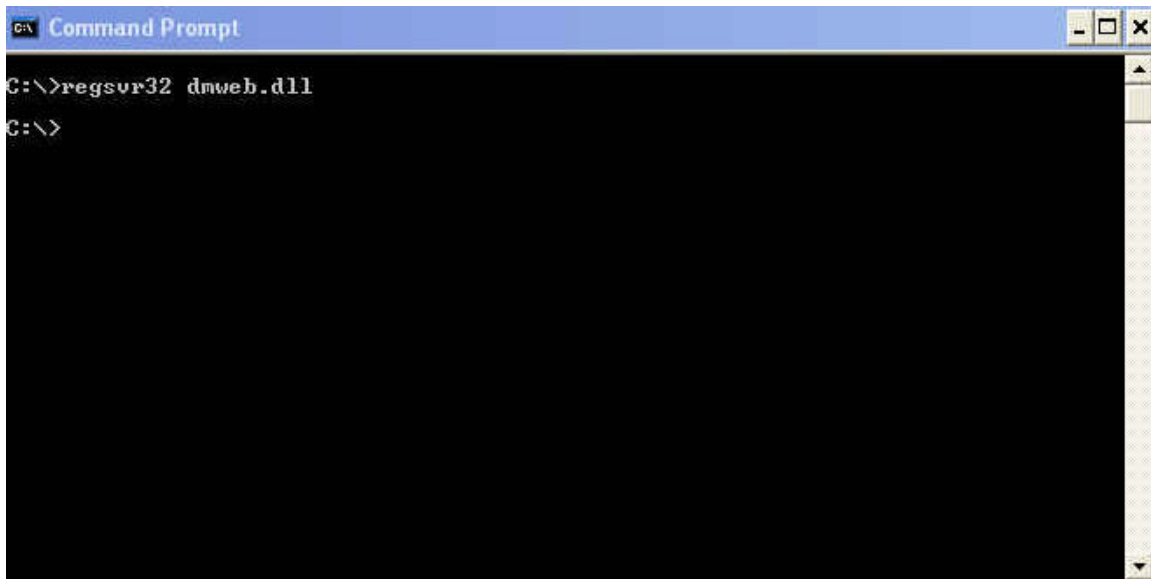


Starting DMWEB.DLL

You will have to run “Regsvr32 DMWEB.DLL” from a DOS window only once.

Example:

Open a DOS window change to the directory you placed DMWEB.DLL and enter
Regsvr32 dmweb.dll



Followed by this window:



Click OK and exit the DOS window.

You may now run any of the dmweb.dll commands from an ASP web page.

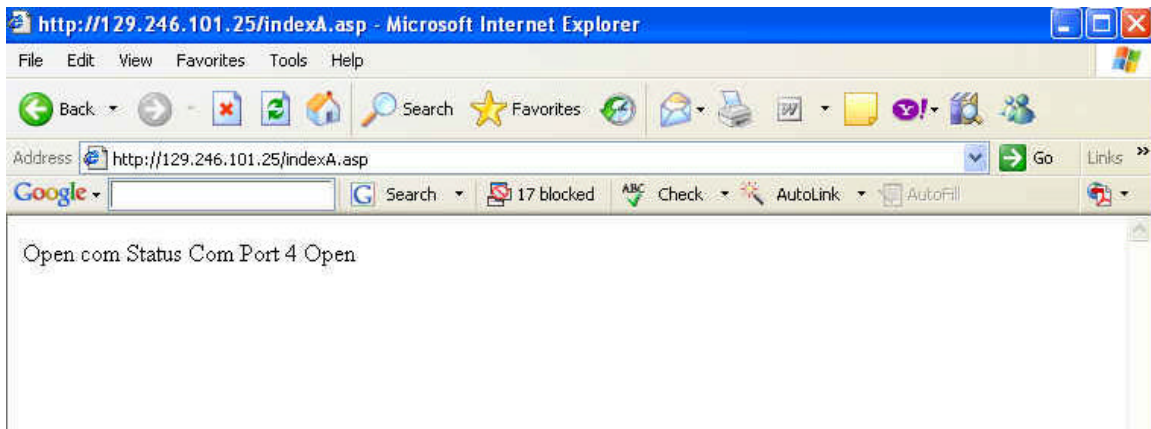


ASP Open Com Port Example

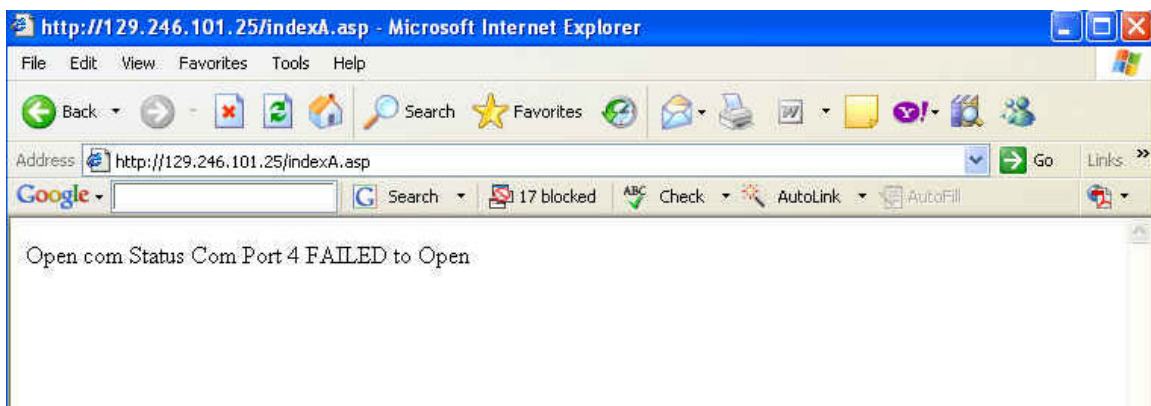
```
<html>
<head>
</head>
<body>

<%
' Declaring variable
Dim sd1
' Creating instance of our Component
Set sd1 = Server.CreateObject("dmweb.OpenCom")
Response.Write "Open com Status " & sd1.OpenCom(4) ' ← open com port 4
Set sd = Nothing
%>
```

Running the above web page will result is a display of:



If successful, else would look like;



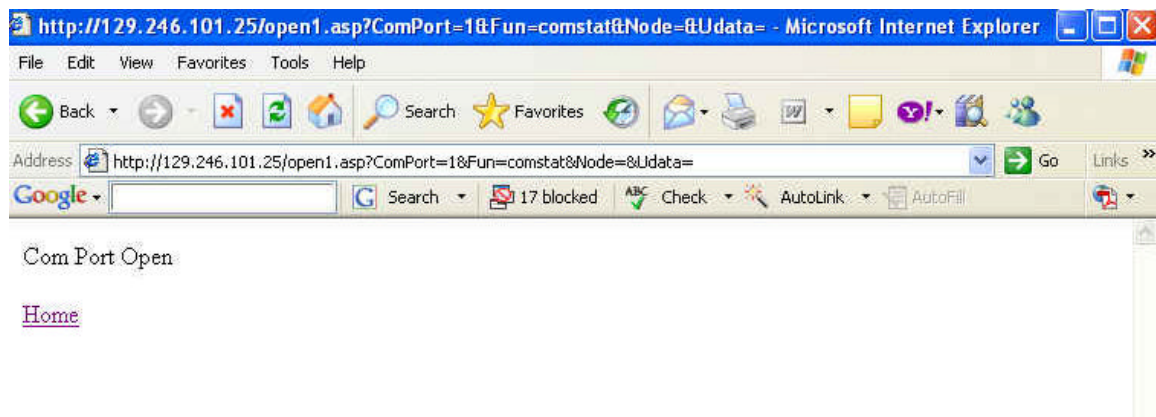


ASP Status Com Port Example

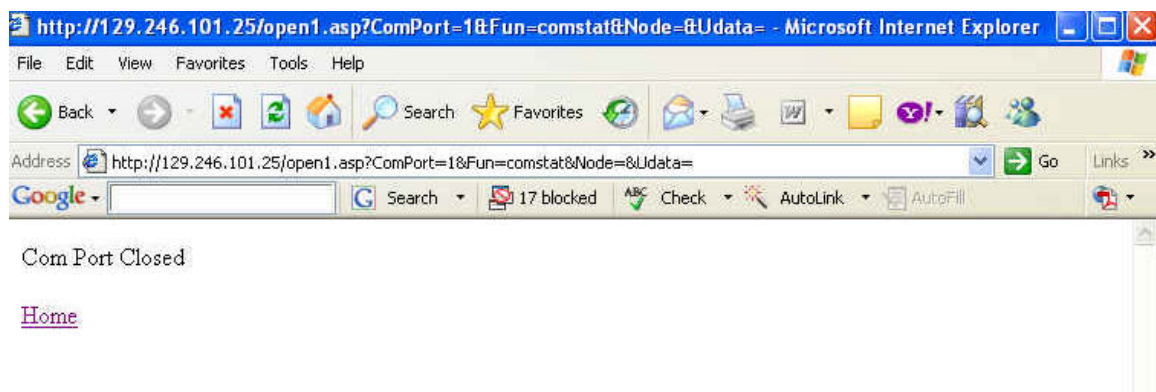
```
<html>
<head>
</head>
<body>

<%
' Declaring variable
Dim sd1
' Creating instance of our Component
Set sd1 = Server.CreateObject("dmweb.ComStat")
Response.Write "Com Status " & sd1.ComStat(4) '← n(4) is don't care
Set sd = Nothing
%>
```

Running the above web page will result is a display of:



If com port is Open or:



If com port is Closed

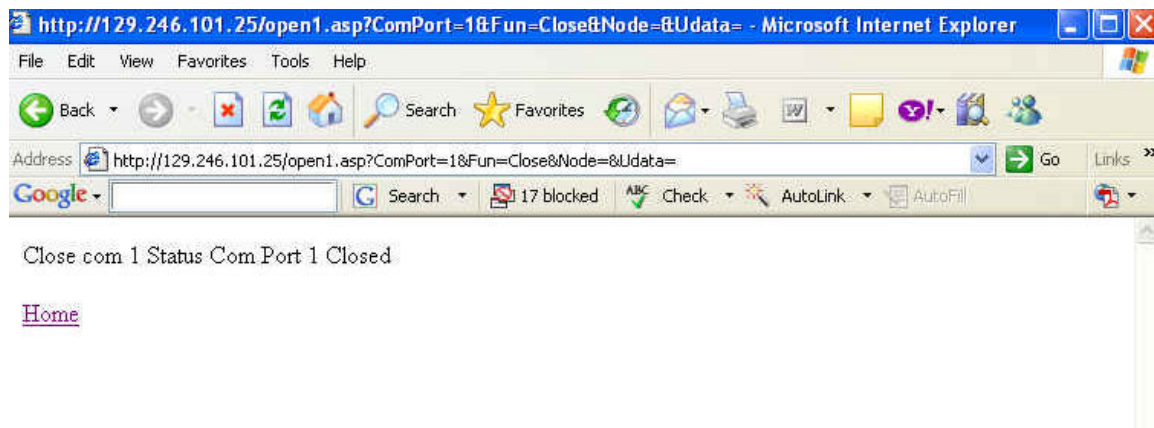


ASP Close Com Port Example

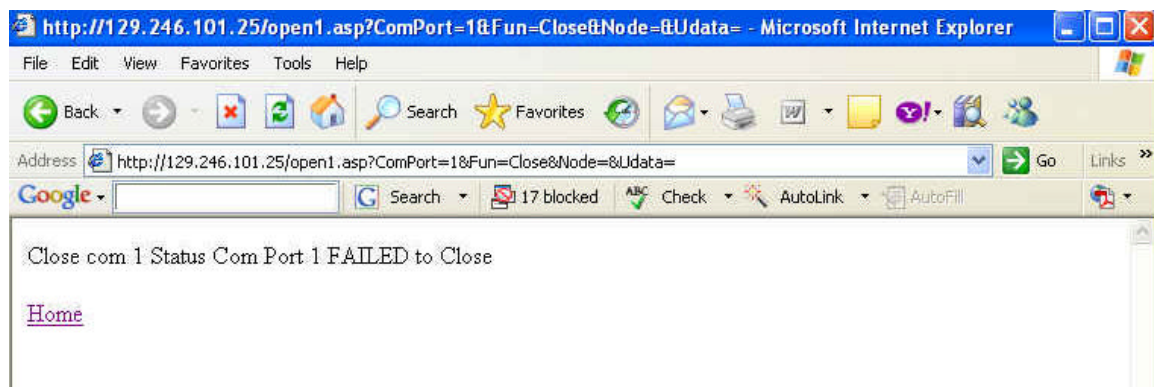
```
<html>  
<head>  
</head>  
<body>
```

```
<%  
' Declaring variable  
Dim sd1  
' Creating instance of our Component  
Set sd1 = Server.CreateObject("dmweb.Close_Com")  
Response.Write "Close Com Status " & sd1.Close_Com(4) '← n(4) is don't care  
Set sd = Nothing  
>%
```

Running the above web page will result is a display of:



Or if it failed:





Simple ASP Example with Drop Downs Menus

Sample Startup Screen:





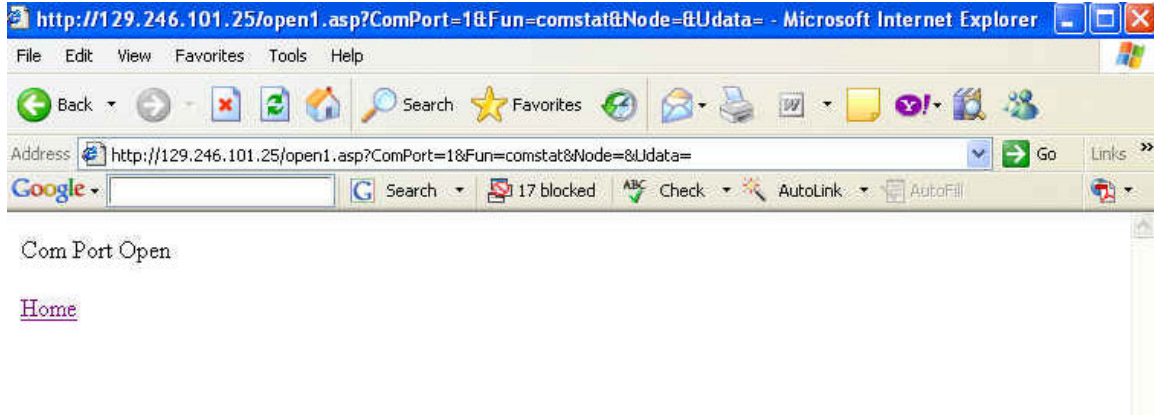
Sample Screen Open Com Port 4



After selecting com port 4 and selecting the Open Com Function click the “Execute” button.



Response from the above:



Now you can execute any of the DMWEB.DLL commands. You can press Home or the Back button to execute another command.



Sample ASP Web Page Demo

```
<html>
<head>
</head>
<body>

<IMG height="68" width="406" border="0" SRC="rfmlogo.gif">
<H1><Center>Simple DM1800 Commands</H1>
<form name="frmForm" action="open1.asp">
<p>
<br>Select Com Port
<select size="1" name="ComPort">
<option value="1">1</option>
<option value="2">2</option>
<option value="3">3</option>
<option value="4">4</option>
<option value="5">5</option>
<option value="6">6</option>
<option value="7">7</option>
<option value="8">8</option>
</select>
<br>
<!<input type="button" name="btnSubmit" onClick="submit()" value="Open Port">
<br><br>
Select Funtion <br>
<select size="1" name="Fun">
<option value="Open">Open Com</option>
<option value="Close">Close Com</option>
<option value="comstat">Com Status</option>
<option value="fOpen">Find Com Port</option>
<option value="bin">Read Binary In</option>
<option value="bout">Read Binary Out</option>
<option value="bon">Set Binary ON</option>
<option value="boff">Set Binary Off</option>
<option value="rad">Read Temperature</option>
<option value="rnode">Reset Node</option>
<option value="rbase">Reset Base</option>
<option value="read">Read Com & Display</option>
<option value="rusera">Read User A</option>
<option value="ruserb">Read User B</option>
<option value="susera">Set User A</option>
<option value="suserb">Set User B</option>
</select>
</p>
```



Node Number

<input type="text" name="Node" size="1">

<input TYPE="Button" onClick="submit()" VALUE="Execute">

</p>

User Data (Max 16 Char's)

<input type="text" name="Udata" size="16">

<input TYPE="Button" onClick="submit()" VALUE="Execute">

</center>

</body></html>

The above code will generate the following web page:





The above page calls the following ASP code to interpret the command being passed from the drop down menus or typed in the text boxes when the “Execute” button is pressed. You can see the commands in the URL window while running the program:

```
<html>
<head>
</head>
<body>

<META HTTP-EQUIV=REFRESH CONTENT=5;>

<%
    ' Start of Script
    ' Declaring variable

Dim port
Dim Node
Dim sd1
port = request("ComPort")           ' Com Port being passed
Node = request("Node")           ' Node number being passed
Fun = request("Fun")             ' Command to be executed being passed
User_data = request("Udata")    ' Text String data being passed

MyString = String(16, "*")          ' load MyString with *, filler
Mylen = Len(User_data)             ' get length of data being passed
If Mylen < 16 Then                 ' if less than 16, make it 16 char.
    fill = 16 - Mylen              ' using the filler
    LeftString = Left(MyString, fill) ' make left string len equal fill len
    us_data = User_data & LeftString ' min/max len now 16 back filled with *
    User_data = us_data
end if

If Mylen > 16 Then                 ' if greater than 16 then
    LeftString = Left(User_data, 16) ' force it to be 16 long only
    User_data = LeftString         ' First 16 chars will be sent only
end if

If Fun = "Open" then             ' check for Com Open command
    ' Creating instance of our Component

    Set sd1 = Server.CreateObject("dmweb.OpenCom")
    Response.Write "Open com " & port & " Status " & sd1.OpenCom(port)
    Set sd1 = Nothing              ' Release the pointer

end if
```



```
If Fun = "Close" then
    ' check for Close Com Command
    ' Creating instance of our Component
    Set sd1 = Server.CreateObject("dmweb.CloseCom")
    Response.Write "Close com " & port & " Status " & sd1.CloseCom(port)
    Set sd1 = Nothing
    ' Release the pointer
```

end if

```
If Fun = "comstat" then
    ' check for Com Status command
    ' Creating instance of our Component
    Set sd1 = Server.CreateObject("dmweb.ComStat")
    Response.Write sd1.ComStat(port)
    Set sd1 = Nothing
    ' Send data to web page
    ' Release the pointer
```

end if

```
If Fun = "fOpen" then
    ' check for Fopen command, find the com port
    ' Creating instance of our Component
    ' Release the pointer
    Set sd4 = Nothing
    Set sd4 = Server.CreateObject("dmweb.ComStat")
    Set sd1 = Server.CreateObject("dmweb.OpenCom")
port = 4
    Count = 1
    while Count < 9
        mydata = sd1.OpenCom(port)
        dat = Instr(1,mydata,"FAILED",1)
        if dat > 0 then
            port = port + 1
            Count = Count + 1
        end if
        mydata1 = sd4.ComStat(port)
        data = Instr(1,mydata1,"Com Port Open",1)
        if data > 0 then Count = 12
    wend
    if Count = 9 then Response.Write "FAILED Opened Port"
    if Count = 12 then Response.Write "Opened Port " & Port & " "
    Set sd1 = Nothing
    Set sd4 = Nothing
```

end if

```
If Fun = "boff" then
    Dim sd2
    ' check for binary out off command
    ' Creating instance of our Component
```



```
Set sd2 = Server.CreateObject("dmweb.Set_Binary_Off")
Response.Write sd2.Set_Binary_Off(Node)      ' Send data to web page
Set sd2 = Nothing                            ' Release the pointer
End if

If Fun = "bon" then                          ' check for binary on command
Dim sd3                                       ' Creating instance of our Component
Set sd3 = Server.CreateObject("dmweb.Set_Binary_On")
Response.Write sd3.Set_Binary_On(Node)      ' Send data to web page
Set sd3 = Nothing                            ' Release the pointer
End if

If Fun = "bout" then                         ' check for read binary command
Dim sd4                                       ' Creating instance of our Component
Set sd4 = Server.CreateObject("dmweb.Read_Binary_Out")
Response.Write sd4.Read_Binary_Out(Node)    ' Send data to web page
Set sd4 = Nothing                            ' Release the pointer

oount = Second(Now)                          ' Get current second
if oount > 57 then oount = 1                 ' Don't get stuck on turn of clock
while ncount <> oount+1                      ' Delay 1 second
    ncount = Second(Now)                    ' windows is never real time!!
wend

                                       ' Creating instance of our Component
Set sd1 = Server.CreateObject("dmweb.Read_Com")
Response.Write sd1.Read_Com(Port)          ' Send data to web page
Set sd1 = Nothing                            ' Release the pointer

End if

If Fun = "rad" then                          ' check for read A/D command
Dim sd5
Dim mydata1
Dim Mypos
Dim sd22
                                       ' Creating instance of our Component
Set sd5 = Server.CreateObject("dmweb.Read_AD_10bit")
sd5.Read_AD_10bit(Node)                    ' Execute the command with no reponse
```



Set sd5 = Nothing

```
oount = Second(Now)
if oount > 57 then oount = 1
while ncount <> oount+1
  ncount = Second(Now)
wend
```

‘ Release the pointer

‘ Get current second
‘ Don’t get stuck on turn of clock
‘ Delay 1 second
‘ windows is never real time!!

```
Set sd22 = Server.CreateObject("dmweb.Read_Com")
mydata = sd22.Read_Com(Port)
```

‘ Creating instance of our Component
‘ Response goes into mydata

```
if StrComp(mydata,Node,1) then
  MyPos = Instr(1,mydata,"Temp. ", 1)
  if Mypos > 1 then
    mydata1 = Right(mydata,7)
    Response.Write mydata1
  end if
end if
```

‘ Look for Node number to match
‘ If so, look for Temp
‘ It’s there if > 1
‘ Only display temperature
‘ Send data to web page

Set sd1 = Nothing

‘ Release the pointer

End if

If **Fun** = "rnode" then

‘ check for reset node command

```
Set sd1 = Server.CreateObject("dmweb.Reset_Node")
Response.Write sd1.Reset_Node(Node)
Set sd1 = Nothing
```

‘ Creating instance of our Component
‘ Send data to web page
‘ Release the pointer

```
Set sd1 = Server.CreateObject("dmweb.Read_Com")
Response.Write sd1.Read_Com(Port)
Set sd1 = Nothing
```

‘ Creating instance of our Component
‘ Send data to web page
‘ Release the pointer

End if

If **Fun** = "rbase" then

‘ check for reset base command

‘ Creating instance of our Component



```
Set sd1 = Server.CreateObject("dmweb.Reset_Base")
Response.Write sd1.Reset_Base(Node) ' Send data to web page
Set sd1 = Nothing ' Release the pointer

' Creating instance of our Component
Set sd1 = Server.CreateObject("dmweb.Read_Com")
Response.Write sd1.Read_Com(Port) ' Send data to web page
Set sd1 = Nothing ' Release the pointer
```

End if

```
If Fun = "read" then ' check for read com command

' Creating instance of our Component
Set sd1 = Server.CreateObject("dmweb.Read_Com")
Response.Write sd1.Read_Com(Port) ' Send data to web page
Set sd1 = Nothing ' Release the pointer
```

End if

```
If Fun = "susera" then ' check for set user data a command

' Creating instance of our Component
Set sd1 = Server.CreateObject("dmweb.Set_User_A")
Response.Write sd1.Set_User_A(Node,User_data) ' Send data to web page
Set sd1 = Nothing ' Release the pointer

' Creating instance of our Component
Set sd1 = Server.CreateObject("dmweb.Read_Com")
Response.Write sd1.Read_Com(Port) ' Send data to web page
Set sd1 = Nothing ' Release the pointer
```

End if

```
If Fun = "suserb" then ' check for et user data b command

' Creating instance of our Component
Set sd1 = Server.CreateObject("dmweb.Set_User_B")
Response.Write sd1.Set_User_B(Node,User_data) ' Send data to web page
Set sd1 = Nothing ' Release the pointer

' Creating instance of our Component
Set sd1 = Server.CreateObject("dmweb.Read_Com")
```



```
Response.Write sd1.Read_Com(Port) ' Send data to web page  
Set sd1 = Nothing ' Release the pointer
```

End if

```
If Fun = "rusera" then ' check for read user data a command
```

```
' Creating instance of our Component
```

```
Set sd1 = Server.CreateObject("dmweb.Read_User_A")
```

```
sd1.Read_User_A(Node)
```

```
Set sd1 = Nothing ' Release the pointer
```

```
oount = Second(Now)
```

```
' Get current second
```

```
if oount > 57 then oount = 1
```

```
' don't get stuck on turn of clock
```

```
while ncount <> oount+1
```

```
' Delay 1 second
```

```
ncount = Second(Now)
```

```
' windows is never real time!!
```

```
wend
```

```
' Creating instance of our Component
```

```
Set sd1 = Server.CreateObject("dmweb.Read_Com")
```

```
mydata = sd1.Read_Com(Port)
```

```
if StrComp(mydata,Node,1) then
```

```
' Look for Node number to match
```

```
MyPos = Instr(1,mydata, "D", 1)
```

```
' If it does look for "D"
```

```
if Mypos >1 then
```

```
' It's there is > 1
```

```
mydata1 = Right(mydata,49)
```

```
' Get data to display
```

```
Response.Write mydata1
```

```
' Send data to web page
```

```
end if
```

```
end if
```

```
Set sd1 = Nothing
```

```
' Release the pointer
```

End if

```
If Fun = "ruserb" then
```

```
' check for read user data b command
```

```
' Creating instance of our Component
```

```
Set sd1 = Server.CreateObject("dmweb.Read_User_B")
```

```
sd1.Read_User_B(Node)
```

```
Set sd1 = Nothing ' Release the pointer
```

```
oount = Second(Now)
```

```
' Get current second
```

```
if oount > 57 then oount = 1
```

```
' Don't get stuck on turn of clock
```

```
while ncount <> oount+1
```

```
' Delay 1 second
```

```
ncount = Second(Now)
```

```
' windows is never real time!!
```

```
wend
```



```
Set sd1 = Server.CreateObject("dmweb.Read_Com")
mydata = sd1.Read_Com(Port)

if StrComp(mydata,Node,1) then
  MyPos = Instr(1,mydata, "D", 1)
  if Mypos >1 then
    mydata1 = Right(mydata,49)
    Response.Write mydata1
  end if
end if
Set sd1 = Nothing
End if

%>

<P><A href="Index.asp">Home </A><BR><BR>
```

' **Creating instance of our Component**
' **Results are put in to mydata**
' **Find Node Number**
' **Look for data**
' **If there will be > 1**
' **Get data to send**
' **Send data to web page**
' **Release the pointer**