A Web-based Distance PLC Laboratory

Yu-Chi Wu　　Wei-Fu Chang　　Chui-Wen Chiu　　Wen-Ching Yu

Department of Electrical Engineering

*Department of Industrial Engineering and Management

National Lien-Ho Institute of Technology

Abstract

Thanks to the Internet technology, there have been many distance learning (web-based) courses offered by academic institutes around the world nowadays through the Internet. And these courses have benefited many students who might be constrained by distance and time. Nevertheless, most web-based courses are "lecturing courses" that cannot fulfill the need for engineering technology education. In this paper, we propose hardware/software architecture for a web-based distance laboratory for programmable logic controllers (PLCs). This Web-based PLC Lab allows the laboratory facilities shared among several academic institutes and can be accessed by registered students to practice PLC experiments at their own home, enhancing the quality of education without much increasing in the overall cost. With the help of web cameras, the students can even observe the live video of PLC experiments through the Internet.

Keyword: Visual Basic, Distance Lab, Active X, PLC, Internet

I. Introduction

The Internet is now providing a new and increasing important medium for distributing information world wide without time constraint, permitting information to be displayed numerically and graphically on any client platform. This has generated great impact to the way of “information/knowledge acquisition” as well as to the manufacturing/commerce automation. The ability to acquire information and even to control instruments/devices at fingertips over the Internet is becoming desirable not only to the general public but also to the professionals.

Thanks to the development of Internet technology, distance learning and distance control of devices are emerging realities. Students nowadays can access information and receiving knowledge/education at home over the Internet. There have been many distance learning (web-based) courses offered by academic institutes around the world through the Internet to benefit many students who might be constrained by distance and time. However, most of web-based courses are “lecturing courses” that still cannot fulfill the need of engineering technology education.

In engineering technology program, most courses have a laboratory component. The quality and availability of laboratory facilities are important factors to the quality of engineering education. However, high-quality laboratories are very expensive to build and most laboratories are only used part of the time by students due to time and/or distance constraints. Sharing laboratories among several academic institutes and allowing students to practice laboratory facilities remotely without time and distance constraints would enhance the quality of education without much increasing in the overall cost. Therefore, the functions offered by distance learning have to be extended to the real engineering laboratories whose physical processes can be remote-controlled. This has led to the setting up of web-based laboratories for teaching and/or research purposes.

In this paper, we propose hardware/software architecture of a web-based distance laboratory (Web-Lab) for programmable logic controllers (PLCs). In the last 20 years, there has been an increasing emphasis on manufacturing automation in order to cut the cost of production and/or to increase product quality, and therefore the field of automatic control has been undergoing a transformation. The number of control engineering positions in manufacturing has been dramatically increasing to the point that the majority of new control engineering positions is now in manufacturing and involves PLCs. With the technology improvement, PLCs now offer unprecedented flexibility of sequencing and reliable control and have been widely used in most industrial applications [1]-[8]. Many academic institutes have recognized this trend, organizing a course or laboratory on PLCs in their engineering program [9].

The presented web-based PLC Lab uses VB, Active X, HTML, and ASP (Active Server Page) to build graphical control interfaces and a windows-based ladder logic program editor (WinProladder) to upload/download ladder programs to/from PLCs, providing easy access to the students. All a student has to do is to access the web-based PLC Lab web site to start the client window at the press of a button. Following the instructions on the web page, the student can practice PLC experiments by clicking the button icons. With a web camera installed beside the PLC, the user can even observe the live video of PLC operations. For security reasons, a device that can control the switches to turn on/off power of PLCs is installed on the server side. The student has to pass an ID check-up to initiate the power of the PLC and then to start experiments. The choice of VB, HTML, and ASP gives us simplicity of implementation and makes Web-based-PLC-Lab browser-independent.

The paper is organized as follows. Section II describes the structures of the presented web-based PLC Lab. Section III presents the software design. Section IV presents a test system to demonstrate the proposed system. Conclusions are given in Section V.

II. Architecture of Web-based PLC Lab

The hardware of the system consists of PLCs, RS-232/TCP-IP converters, PCs, a controllable switch device, and a web camera server with CCDs. Fig. 1 shows the structure of the system. The controllable switch device in Fig.1 is connected to the server through RS-232 and can turn on/off the power for 8 different sets of PLCs and RS-232/TCP-IP converters. An ID check-up web page program on the server was designed to check student's ID and password for security reasons. Once the student logins the server web page and passes the ID check-up, the server sends signals through RS-232 to the controllable switch device to turn on the power for a designated set of PLC and RS-232/TCP-IP converter. And then the student enters into another web site controlled by a PC that directly communicates with this designated PLC through RS-232. Some examples of PLC ladder programs were pre-installed on this PC. All the student has to do is to follow the instructions on the web page and then can practice these examples. The statuses of digital inputs/outputs (IOs) of the PLC are displayed on the web page through Active X controls. The student can also control these digital inputs by clicking their button icons. A program was developed to deal with hardware service process and the communication between the client and the PLC. The student also can observe the live video of PLC operations on the web page through the web camera. A timer mechanism is designed in the ID check-up program to allow registered students to operate the designated PLC for a limited period (1 hour in our implementation but it is changeable). This timer prohibits the same student from using a PLC too long or a PLC from being idle too long if the student forgets to log off. The RS-232/TCP-IP converter is connected to another port of the PLC. The student can run WinProladder and upload/download ladder programs through the Internet. The WinProladder program can be downloaded from the server to the student's PC. Once it is installed, the student can use it to upload, download, and even edit ladder programs. Fig. 2 shows the client/server structure of the PC and PLC.

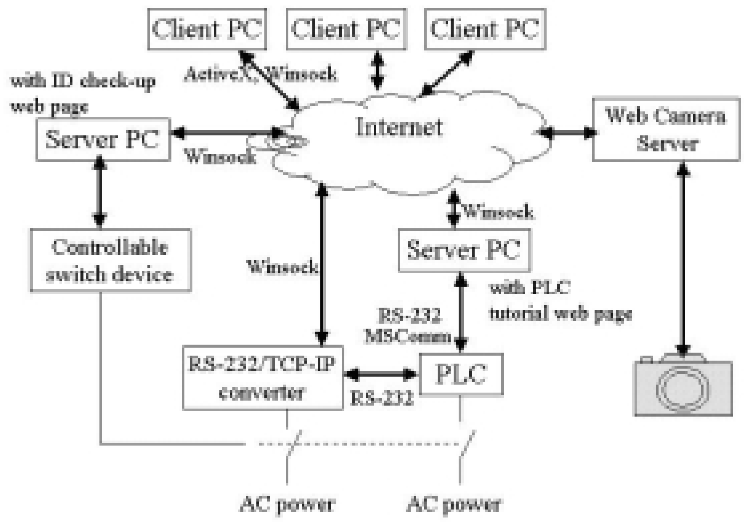

Fig. 1 system structure of web-based PLC Lab

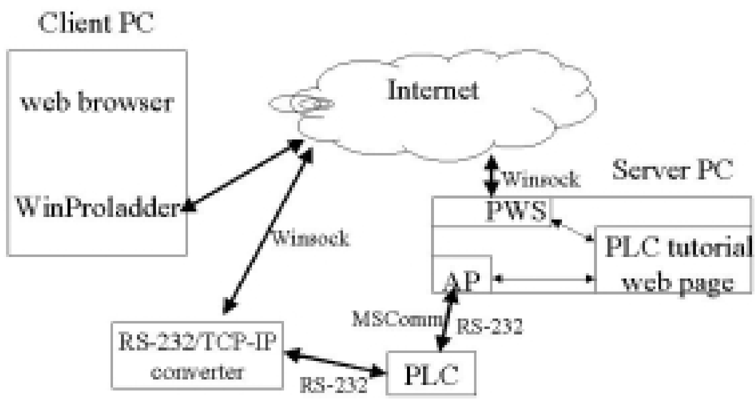

Fig. 2 client/server structure of PLC and PC

III. Software Design

The graphical control interface is written in Visual Basic (VB). VB is based on an event-driven programming model and supports a number of features that make it an excellent language for quickly creating full-featured solutions, taking advantage of the graphical user interface in Microsoft Windows. These features include data access, Active X technologies, Internet capabilities, rapid application development (RAD), etc. It can also use system-provided APIs or external DLL/OCX to extend its functionality. For instance, we can easily use winsock.ocx (Winsock control) to develop an Active X control in the ASP program that gives the Internet accessibility to the server and the client.

In our implementation, the server program uses Winsock control of VB to communicate with the client and the RS-232/TCP-IP converter for data acquisition and supervisory controls of the PLC. Using the Winsock control, it is possible to create sockets and perform TCP- and UDP-style communications over TCP/IP network. The following is a program example for the server using Microsoft Winsock Control.

```
Option Explicit
Public gSockInstance As Long    ' the number of persons connected
Private Sub Form_Load()
    Winsock1(0).Listen      ' the socket is listening
End Sub

Private Sub Winsock1_Close(Index As Integer)
    Winsock1(gSockInstance).Close                    ' the socket is closed
End Sub

Private Sub Winsock1_ConnectionRequest(Index As Integer, ByVal requestID As Long)
    gSockInstance = gSockInstance + 1              ' increase the number of person connected by 1
    Load Winsock1(gSockInstance)                      ' create object
    Winsock1(gSockInstance).Accept requestID    ' accept a client connection request
    Winsock1(gSockInstance).SendData gSockInstance    ' send data to remote computer over
                                                      ' network
End Sub

Private Sub Winsock1_DataArrival(Index As Integer, ByVal bytesTotal As Long)
  ' store received string
End Sub

Private Sub Winsock1_Error(Index As Integer, ByVal Number As Integer, Description As String,
ByVal Scode As Long, ByVal Source As String, ByVal HelpFile As String, ByVal HelpContext As
Long, CancelDisplay As Boolean)
    MsgBox Description, , Number                    'display connection error message
End Sub
```

Active X components for the client communicate with the RS-232/TPC-IP converter also through the Winsock control [10]. The communication program for the client is similar to the one for the server except it needs to perform supervisory controls of hardware. The following is a program example for the client using Microsoft Winsock Control.

```
Option Explicit
Dim ConnectOK As Boolean      ' connection OK?
Dim break    As Long                ' the times of disconnection

Private Sub Form_Load()
      Winsock1.Connect "203.64.199.199", "5555"    ' executed by client to request a connection
                                                                            ' with server at designated IP address and
                                                                            ' port number
      ConnectOK = False
End Sub

Private Sub Winsock1_Close()
      ConnectOK = False
End Sub

Private Sub Winsock1_Connect()
      ConnectOK = True
End Sub

Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)

End Sub
```

```
Private Sub Winsock1_Error(ByVal Number As Integer, Description As String, ByVal Scode As
Long, ByVal Source As String, ByVal HelpFile As String, ByVal HelpContext As Long,
CancelDisplay As Boolean)
      Select Case Number
            Case 11001
                  If break > 3 Then
                        MsgBox "The server is not found, please try later."
                        Unload Me
                  End If
                  break = break + 1
                  Call Form_Load
            Case Else
                  MsgBox Description, , Number                          ' display connection error message
      End Select
End Sub
```

In Fig. 2, the Active X control on the web page sends commands through socket to communicate with the application program (AP) on the server. This is a typical client/server structure. The AP on the server feeds back what the client requests. Figure 3 shows the whole process flowchart as the client requests to get the statuses of the PLC. Since the PLC is connected to the server PC through RS-232, after the AP received the request from the client the AP sends command to the PLC through RS-232. Once the server received the feedback from the PLC, the AP interprets the data and then responds back to the client. A program example that uses MSComm control [20] to monitor/control PLC through RS-232 port is as follows.

```
Private Sub Form_Load()
    On Error GoTo errfix
    MSComm1.PortOpen = True ' open the RS-232 port
finally:
    Exit Sub
errfix:
    MsgBox Err.Description, , Err.Number
End Sub

Private Sub Form_Unload(Cancel As Integer)
    On Error GoTo errfix
    If MSComm1.PortOpen Then MSComm1.PortOpen = False    ' close RS-232 port
finally:
    Exit Sub
errfix:
    MsgBox Err.Description, , Err.Number
End Sub
```

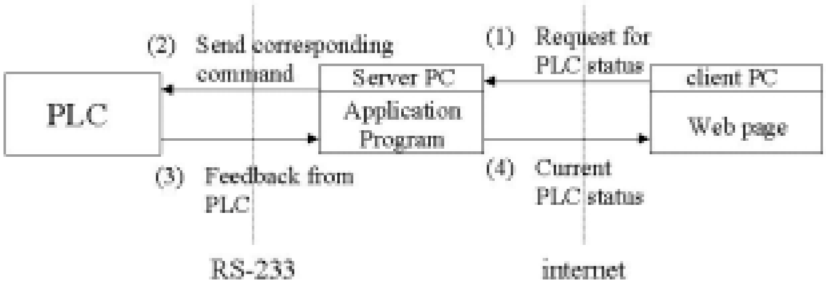

Fig. 3 process flow between the client and the server

There are several PLC program examples on the web page for the web-based PLC laboratory. This tutorial web page was coded using Active X technology and HTML/ASP. The students can download these examples and edit them using WinProladder. Having the programs tested under the ladder program editor environment, the students can upload the program back to the PLC and then observe the statuses of the PLC through the graphical control interface on the web page and the live video of PLC operations through the web camera. The graphical control interface was developed based using Active X controls.

IV. Implementation

To operate the PLC remotely, the students only need a web browser and WinProladder that can be easily downloaded from the server of web-based PLC Lab. The operation steps are as follows.

1. Use a web browser, e.g. Microsoft Internet Explorer, to connect to the server that provides the ID check-up web page (Fig. 4), and then choose an available PLC to turn on the power for the PLC and RS-232/TCP-IP converter by clicking the button icon. A PLC tutorial web page (Fig. 5) will be opened on your PC. Click the web-based PLC Lab item to open another web page for the detailed tutorial instructions (Fig. 6).
2. If WinProladder has not been downloaded, download it and install it.
3. Choose an example on the PLC tutorial web page, e.g. the double blinking circuit (Fig. 7), and read all its relevant instructions on the web page. Then download the example to your PC.
4. Execute WinProladder and set up the communication parameters to connect the PLC. Then under the WinProladder environment import the "double blinking circuit" program that was just downloaded, upload this program to the PLC, and run the PLC.
5. On the PLC tutorial web page, click the "statuses of IO" item to get the graphical control interface (Fig. 8). For the "double blinking circuit" example, as X0 is on, two timers start working to turn on Y0 and Y1 alternately.
6. To observe the live video of PLC operations (Fig. 9), click the "web cam" icon on the PLC tutorial web page.

On the server, the developed application program (AP) provides hardware controls and connection management. The connection management can control the maximum number of clients connected to the server and prohibits some specific bad IPs from connecting to the server. The graphical control interface of the AP is shown in Fig. 10. The instructor or the teacher can monitor and control the statuses of PLC IOs.

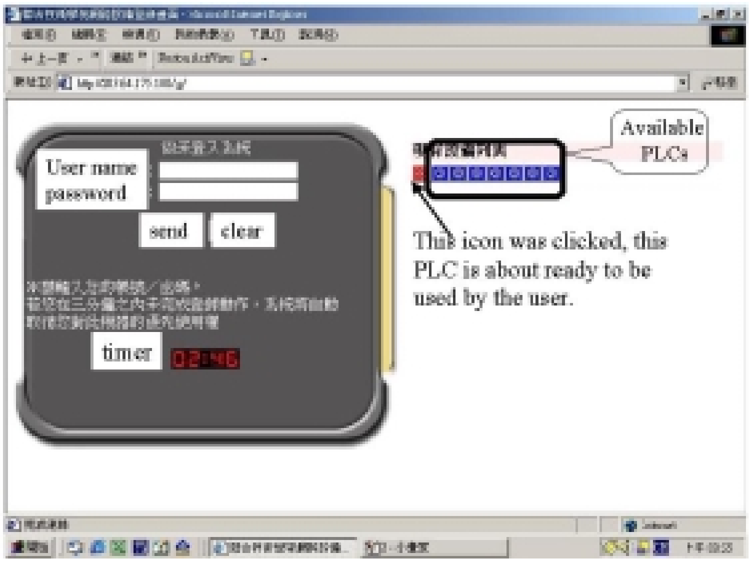

Fig. 4 ID check-up web page

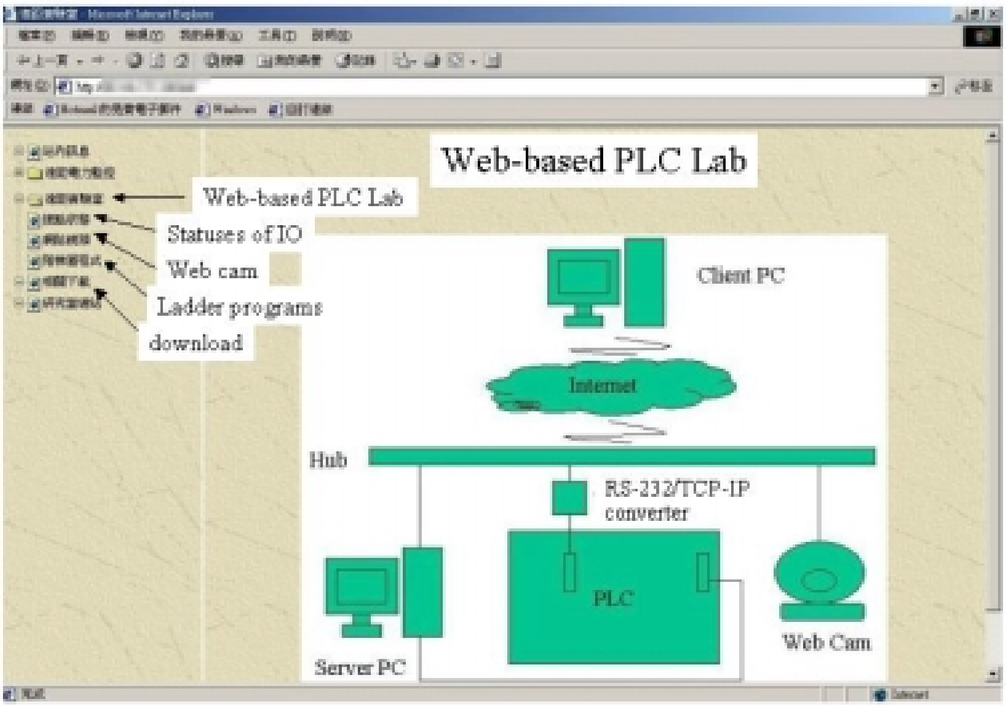

Fig. 5 Tutorial PLC web page

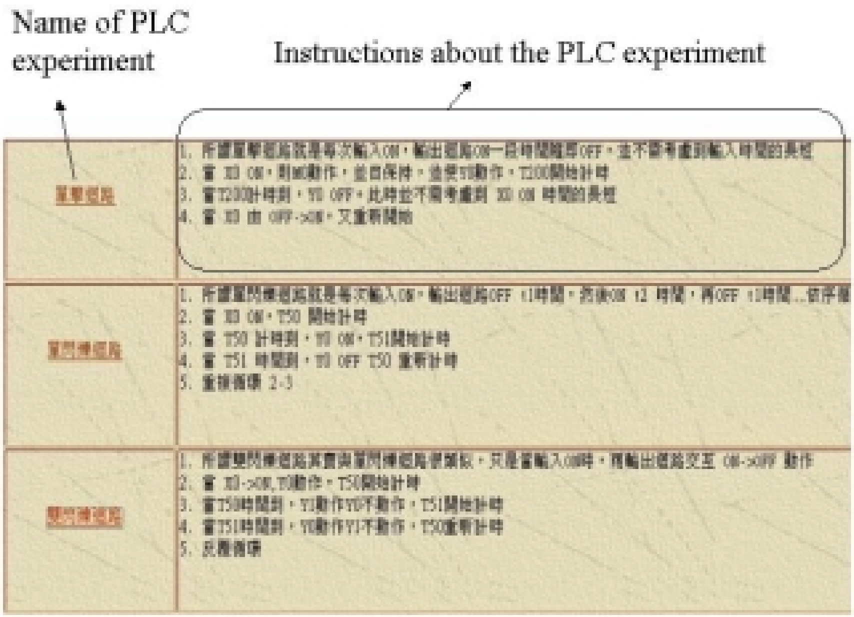

Fig. 6 detailed tutorial instructions

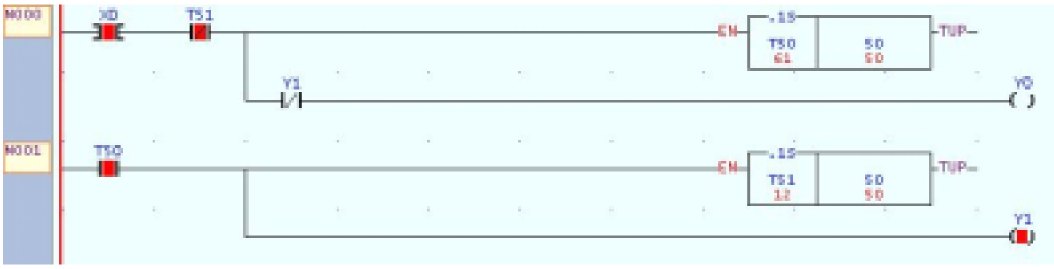

Fig. 7 Ladder program for double blinking circuit

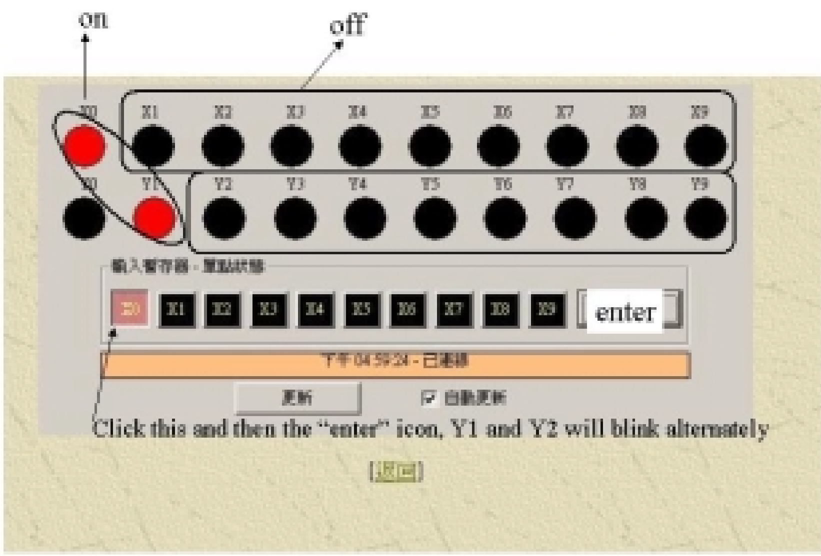

Fig. 8 Graphical control interface

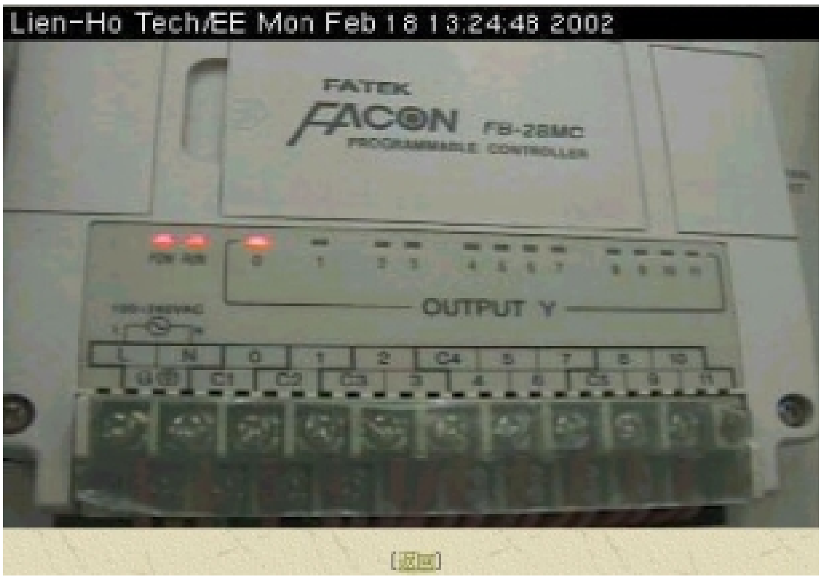

Fig. 9 Live video of PLC operations

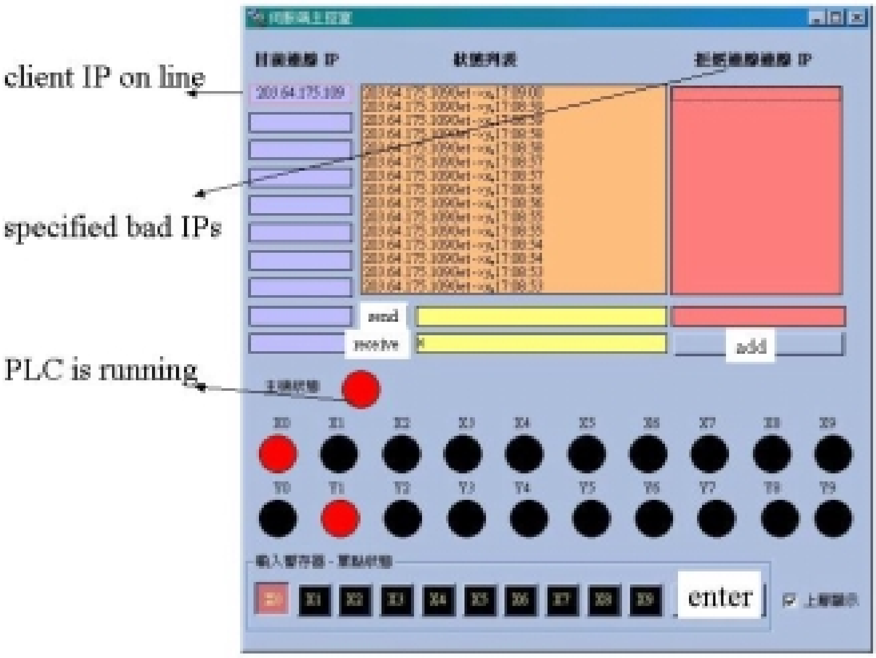

Fig. 10 Graphical control interface of AP on the server

V. Conclusions

Sharing laboratories among several academic institutes and allowing students to practice laboratory facilities remotely without time and distance constraints would enhance the quality of education without much increasing in the overall cost. In this paper, we present a web-based distance PLC laboratory to allow registered students to use PLC facilities remotely through the Internet. Since PLCs now offer unprecedented flexibility of sequencing and reliable control, the number of control engineering positions in manufacturing has been dramatically increasing to the point that the majority of new control engineering positions is now in manufacturing and involves PLCs. This Web-based PLC Lab allows the laboratory facilities shared among several academic institutes and can be accessed by registered students to practice PLC experiments remotely at their own home. With the help of web cameras, the students can even observe the live video of PLC experiments through the Internet. The set up of this web-based distance PLC laboratory can help engineering students to get ready to be involved at control engineering positions.

VI. References

1. Byres, E.J., (2000) "Designing secure networks for process control," IEEE Industry Applications Magazine, 33-39
2. Zhou, M.C. and Twiss, E., (1998) "Design of industrial automated systems via relay ladder logic programming and Petri nets," IEEE Transactions on Systems, Man, and Cybernetics, Part C, 28(1), 137-150
3. Ramirez-Serrano, A., Zhu, S.C., Chan, S.K.H., Chan, S.S.W., Benhabib, B., (2000) "A hybrid PC/PLC architecture for manufacturing system control: implementation," IEEE International Conference on Systems, Man, and Cybernetics, (3), 1679-1702
4. Halang, W.A., Kramer, B.J., (1994) "Safety assurance in process control," IEEE Software, 11(1), 61-67
5. Lauzon, S.C., Ma, A.K.L., Mills, J.K., Benhabib, B., (1996) "Application of discrete-event-system theory to flexible manufacturing," IEEE Control Systems Magzine, 16(1), pp. 41-48
6. Boothroyd, D., (1995) "Failsafe systems gain the flexibility of the PLC approach," IEEE

Computing & Control Engineering Journal, 6(1), 43-44

7. Roman, R. and Wilson, R., (1995) “Commercial demand side management using a programmable logic controller,” IEEE Transactions on Power Systems, 10(1), 376-379
8. Graham, A.M., Tezadi-Amoli, M., (2000) “Design, implementation, and simulation of a PLC based speed controller using fuzzy logic,” IEEE Power Engineering Society Summer Meeting, 4, 2475 –2480
9. Erickson, K.T., (2001) “Factory automation: a controls course for every university,” Proceedings of the 2001 American Control Conference, 2, 1167 –1172
10. Russo, M.F. and Echols, M.M., (1999) *Automating Science and Engineering Laboratories with Visual Basic*, John Wiley & Sons, Inc.