Development of a Design-Aiding and Self-Learning Tool for Pneumatic Sequential Circuits

By Prof. Dr. Waladin K. Sa'id, and Fadi Ahmed Said Zeyd*

Received on:30/1/2003

Accepted on:11/9/2004

Abstract

A windows-based computer package for pneumatic sequential circuits is implemented using Microsoft Visual Basic 6.0 as a programming language. This package enables the user to construct pneumatic sequential circuits and simulate their operation. The package has a library consisting of four types of cylinders, delay valve, signal generating valve, compressed air unit, four types of logic valves, and twenty-one types of the most commonly used directional valves. The package is enhanced with powerful animation features that enable the user to construct pneumatic sequential circuit and test its operation in a friendly interaction. The workspace provided within the package is sufficient to add about 67500 pneumatic components. The package is useful for teaching, self-learning, troubleshooting, and testing new design ideas by going through the cycle step-by-step.

الخلاصة

نفذت حقيبة محاكاة لعمل الدارات التتابعية الرئوية تعتمد أسلوب النوافذ في التعامُل مع المُستخدم، و أستخدمت لغة البرمجة Microsoft Visual Basic 6.0 لهذا الغرض. تمكن هذه الحقيبة المستخدم من بناء الدارات التتابعية الرئوية عن طريق مكتبة تحتوي على أربعة أنواع من الاسطوانات، وصمام تأخير، وصمام لتوليد النبضة، ووحدة للهواء المضغوط، وأربعة صمامات منطقية، وأحد وعشرون صمام سيطرة اتجاهية من بين الصمامات الأكثر استخداماً. وقد عززت هذه الحقيبة بمُميزات، مثل الرسوم المتحركة والتأثيرات اللونية. إن المميزات الموجودة في هذه الحقيبة تساعد المُستخدم على بناء أي دارة تتابعية رئوية وفحص عملها بأسلوب سهل. إن المساحة المهيأة لبناء الدارات الرئوية في هذه الحقيبة تتسع لحوالي ٦٧٥٠٠ قطعة رئوية. يمكن الاستفادة من برنامج المُحاكاة لأغراض التعليم والتعلّم الذاتي وتحري الخلل و إصلاحه و اختبار أفكار تصميمية جديدة. ومما يُساعد على إتمام هذهِ العمليات إمكانية تنفيذ الدارة خُطوة بخُطوة.

Key Words

Pneumatic, sequential, package, simulation, design-aiding, self-learning, and teaching.

*Control and Systems Engineering Department, University of Technology, Baghdad, Iraq.

1.Introduction

Pneumatic actuators are cheap, reliable and widely used in automated manufacturing industries. Pneumatic technology is a good tool for implementing low-cost automation technology which is characterized by using ready, available, simple, reusable, and flexible components to be applied to small and medium batch productions [1]. For industries that require non-contaminating work media such as food, wood, textile, and leather industries, pneumatic automation is most suitable.

In recent years, computers with their great computational speed abilities and visual facilities have found applications in the field of simulation and design of pneumatic sequential circuits. This paper describes a Windows-based computer package for the simulation of the pneumatic sequential circuits. This package allows the users to construct a pneumatic sequential circuit and check its operation through visual animation features. The user can interact with the package through a friendly-designed graphical-user interface. The graphical-user interface is the part of the program that the user sees and interacts with. It is built with graphical objects, such as buttons, text boxes, scroll bars, and menus. In general, these objects already have meanings to most computer users. For example, when the user presses an "OK" button on a dialog box, the settings are applied and the dialog box is dismissed. Packages provided with graphical-user interface are generally easier to learn [2].

A number of computer packages are reported in the literature that deals with pneumatic circuits [3, 4, 5]. However, the survey showed that such programs have their own shortages.

The Microsoft Visual Basic 6.0 is used to implement the package reported in this paper [6]. Henceforth, it will be referred to as PACAD in short for Pneumatic Automation Computer Aiding Design. It represents a very good tool for building graphical-user interface. It also consists of simple and fully featured tools that make the programming process flexible enough to fit the task it builds.

2.Pneumatic Automation Package (PACAD)

This package allows the user to design, simulate, animate and analyze complex sequential pneumatic circuits. There is no software restriction on the number of components being used. The program doesn't consider the value of pressure supplied. It only recognizes whether or not a pressure exists. The splash screen of this package is shown in Fig. (1). The main window of the package is shown in Fig. (2), which consists of workspace window, toolbar, and horizontal and vertical scroll bars. The workspace window is the space into which the sequential pneumatic circuit is constructed and analyzed. Horizontal and vertical scroll bars enable the user to explore all contents of the workspace.

The package has two modes of operation, User-Design mode and Run-Time mode [7]. The User-Design mode enables the user to construct a sequential pneumatic circuit, while the Run-Time mode is used to simulate the operation of the constructed circuit.

2. 1 .Toolbar Elements

The friendly-user uses toolbar consisting of two control boxes and several buttons, which enable the user to construct, analyze, and save the constructed circuits into a file, in addition to other functions. The toolbar contains the following items:

1. *Circuit component buttons*: These buttons are the square buttons at the bottom of the toolbar, Fig. (2). Some of these are group-buttons, which have a small triangle at the bottom. When clicking one of these group-buttons a small window appears containing other additional buttons. All these buttons (including those contained within the group-buttons) can be dragged and dropped onto the workspace window, with the exception of the group buttons, by pressing the left button of the mouse at the selected component and then dragging the component –by moving the cursor– to the workspace. Releasing mouse button at the desired location in the workspace will create the selected component at the desired location.

Circuit component buttons are activated only at User-Design mode. A description of these buttons is given below:

- Cylinders group button: This button introduces the following cylinder types, Fig. (3-a):
- Single-rod, double-acting.
- Single-rod, single-acting.
- Double-rod, double-acting.
- Double-rod, single-acting.

In this work, these cylinders operate in discrete form. So, their pistons are either fully extended or fully retracted. Although only linear actuators are available as components, however this will not limit the use of this package to other actuators. The cylinders in the menu can be used to represent all actuators that have two states of operation (ON and OFF). Examples of this are conveyor belts motors, drilling motors, milling motors ... etc.

- Directional valves buttons: These buttons (including group and non-group buttons) provide most of the directional valves in use. In total, they are 22 valves. The symbol in each group button represents the type of valves contained within it (i.e. the number of switching positions and input-output ports), as shown in Figs. (3-b) to (3-g).
- Group button of logic gates: This button, as shown in Fig (3-h), introduces AND and OR gates. To avoid the connection of multi AND and multi OR gates which complicates circuit layout, additional components are introduced, Multi-AND and Multi-OR. When dragging one of these to the workspace, a dialog box will appear in which the required number of input ports is entered, as shown in Fig. (4). The allowable number of input ports is from 2 to 10.
- Signal generating valve button: The button contains a pulse symbol.
- Time delay valve button: This component is designated by the word "Delay" on its button, as

shown in Fig. (2). The value of the delay is adjustable.

- Compressed air source button: This is the third button in the second row of the toolbar of the main window (Fig. (2)).
- Pipe-line button: This is explained later in section 2.2.
- Shortcut button: This is explained later in section 2.2.

2. *File management buttons*: These are the three buttons located at the upper-left of the Toolbar. They are activated at User-Design mode only. The function of these buttons is:

- New button: Clears the workspace by deleting all current components to start a new design project.
- Save button: Enables the user to save the current circuit into a file through a suitable dialog box (Fig. (5)).
- Open button: Clears the workspace by deleting all current components, and then reconstructs the circuit saved into a file. This file is selected from the dialog box shown in Fig. (6) that appears when "Open" button is clicked.

3. *Start button*: This button is located at the upper-right corner of the toolbar (Fig. (2)). It takes the form of a switch. Each time this button is clicked, the operation mode is toggled between User-Design mode and Run-Time mode.

4. *Zoom combo box*: Combo box is a text box with an additional drop-down list. The list appears when the button attached to the text box is clicked. User can enter information to text box by selecting an item from the drop-down list. Zoom combo box is used to change the layout size of the circuit, by selecting one of the items into the drop-down list. There are three zooming options 100%, 75%, and 50%. Zoom combo box is activated only at User-Design mode.

5. The other buttons will be explained later.

2. 2 .User-Design Mode Activities

In the User-Design mode, the user can construct any pneumatic circuit. The activities that help the user to construct a circuit, which makes it easy to get an access to components described in previous section, are to be described next.

- Any component available in the toolbar menus can be added to the workspace by pressing on this component using the left button of the mouse, and drag it to the workspace to a desired location, and then releasing the button.
- For each cylinder dragged to the workspace, a reference and unique number appears behind it, as shown in Fig. (7-a). When double-clicking on this cylinder, a dialog window appears, as shown in Fig. (7-b). The window has two check boxes. If the first box "Add limit switches" is checked, then two symbols appear behind the piston. These symbols represent the physical location of limit switches, as shown in Fig. (7-c). These symbols are used for position sensing, as will be explained later. The second check box "Initially extended" is used to extend the piston of the cylinder initially. It should be noted that this check box is enabled only if double-acting cylinders are selected.

- Actuation methods of dragged directional valves can be changed by double-clicking on the selected valve. A dialog window will appear depending on the number of positions of the valve being selected, as shown in Fig. (8). The available actuation methods are spring, push-button, plunger, direct application of pneumatic pressure, and lever.

When the plunger is selected to actuate a particular valve, the user can choose the position that should be sensed by the plunger. Where, by double clicking on the plunger, a dialog window will appear as shown in Fig. (9-a). This window contains a combo box, its drop-down list items (Fig. (9-b)) are the symbols written behind cylinder pistons that appear when "Add limit switch" check box is checked (Fig. (7)). By selecting the desired symbol, the plunger will operate as if it is physically located at the selected position. When a piston depresses the plunger, then it is forced in. This action is illustrated visually in the workspace, as shown in Fig. (10-a).

The lever position can also be actuated when the left button of the mouse clicks it. Similarly, the push-button can be pressed in the same manner. The movement of the lever and push button are illustrated visually as shown in Figs. (10-b), and (10-c).

- The time delay valve dragged to workspace has a text box, which displays the delayed time, as shown in Fig. (11-a). If the user needs to adjust this delay time value, a dialog window will appear, as shown in Fig. (11-b), when double-clicking at the device. This window will enable the user to set desired time.

- To connect between an input port and an output port of any two components in the workspace, the mouse cursor should be moved to be at one of these ports. The right button of the mouse should be pressed then the mouse should be moved toward the target port without releasing the mouse button. During this process a line appears from first port to cursor location. When the cursor is near to the target port, a small node will appear at the head of the cursor. This means that if the button is released here, then a connection will be made. If the user attempts to connect two input ports or two output ports together, then it is auto ignored.

Each connection line between two ports consists of five lines and four joints. This structure enables connection to be powerful and flexible. The line connection between any two ports is usually done without passing through any component. But if this happens, then the user can move the three middle line segments freely (vertical lines can be moved horizontally, while horizontal lines can be moved vertically). Pressing the left button of the mouse when the cursor is at the desired line segment, and then moving the cursor leads to shifting the line. The button should be released at the desired location.

If there is an output port, which is connected to more than one input port, then there are two ways to do task:

- By using pipeline button (Fig. (2)), a pipeline can be dragged to the workspace. The output port of a pneumatic component can be inputted to the pipe line. This pipe in turn can be used to distribute pressure to all desired input ports, as shown in Fig. (12). Pipe line length can be changed by pressing on the left button of the mouse at one of its ends and move it left or right.

- By using shortcut button (Fig.(2)), the connection can be edited in a format that reduces circuit layout complexity. This method enables a connection between an output port and any number of input ports without using a line connection between them. By pressing at the shortcut button in toolbar and dragging it to workspace, and releasing the mouse button when the mouse cursor is over the desired output port, a dialog window will appear, as shown in Fig. (13-a), which enables user to input a desired label. The label is written in a rectangular box, which is connected to the output port through a connection line. This box represents the shortcut source, as shown in Fig. (13-b). The dialog window usually shows a default label that represents unique number, the user can either select or change it.

The output port can be connected to any desired input port (or more) as follows. Select the shortcut source of the output port, press the right button of the mouse, move the mouse cursor to the desired input port, and release the mouse button there. A rectangle connected to the desired input port will appear. It contains the same label of its shortcut source but the border of the rectangle is thicker than the source one. Figure (13-b) is an illustration of the shortcut application.

- The colour of all components and connection lines used in the workspace is black. The colour is turned to red when selecting any item, by clicking or pressing by the left mouse button at it. If the "Delete" button (Fig. (2)) is clicked, then the selected item will be deleted. This button is activated only at User-Design mode. It should be noted that any component could not be deleted until all its connections are disengaged including the plunger related to piston position. If the user attempts to delete a component before releasing its connections, then an error message will appear.
- To modify the location of any component in workspace, press the left mouse button when the cursor is at the desired component. The component colour will become red, moving the cursor to move the component. Release the button when the component reaches the desired location. If the moved component has connections with other components, then the connection lines will be adjusted automatically.
- "Label" button is used to set or modify the label of cylinders and directional valves components in workspace. Selecting the desired component and clicking "Label" button (Fig. (2)) can do this. A dialog window will appear, as shown in Fig. (14-a), which enables the user to enter a label that will be written under the selected component in green colour (Fig. (14-b)). This button is activated only at User-Design mode.
- The "Enable switch bar" button shown in Fig. (4.2), when clicking a vertical bar named "Switch Bar",

appears at the left side of workspace. The text written inside this button will turn immediately to "Disable switch bar". A second click of this button will make the switch bar disappear. The text written inside the button will return to "Enable switch bar". This button is used to toggle appearance of "Switch Bar". "Switch Bar" is a vertical bar that contains buttons. Each button represents a push-button or lever that is used in directional valves in the workspace. The label under each button represents the valve label to which the push-button or lever belongs. A click at any button in the bar is like clicking on the push-button or the lever of the valve itself. So, "Switch Bar" introduces links between buttons displayed by the bar, and the push buttons or levers used in workspace, as shown in Fig. (15).

2. 3 .Run-Time Mode Activities

Run-Time mode is the mode which enables the user to check and test a designed circuit visually. The user can follow up in details the operation of the automation circuit. Animation and colouring effects of components according to their status are used to show the behavior of components during circuit operation. The activities that can be carried out by the user are explained below.

- To observe or monitor cylinders' operation (their sequence), there are two methods available to the user:
- The cylinders in workspace themselves can be monitored graphically. The pistons of the cylinders are extended and retracted graphically. This method is limited to the area of workspace covered by the monitor. If the number of cylinders is large, the workspace can be explored by using scroll bars during circuit operation to observe the status of cylinders outside the viewed window. This procedure is not convenient especially if the circuit is very large even with 50% reduction in size.
- Event-diagram window can also be used to monitor cylinders operation. The "Event-diagram" button on toolbar (Fig. 2) is activated during both User-Design and Run-Time modes. Event-diagram window is used to monitor at most ten cylinders; these cylinders can be selected by using ten combo boxes, as shown in Fig. (16). Each combo box corresponds to a certain cylinder and its drop-down list items are the reference numbers of the cylinders in use. Each combo box can be used to select a given cylinder. The selection of cylinders to be observed should be done through User-Design mode only. The combo boxes are inactive during Run-Time mode. The "Set default" button, in event-diagram window, is used to make each combo box select a distinct cylinder automatically. If the used cylinders are more than ten, then ten cylinders with lowest reference numbers will be selected. This button is activated only at User-Design mode.

In Run-Time mode, a curve is drawn for each combo box that corresponds to the cylinder selected by that box. Each curve is switched between two levels: lower and upper level, which mean the piston is retracted and extended, respectively. Rising of the curve to the upper level designates the beginning of piston extension operation. Falling of the curve to the lower level designates the beginning of the retraction operation of the piston. The Event-diagram window is also useful since it shows the history of movement of the cylinders.

- The directional valves have two or three switching positions. During the running operation at Run-Time mode, the colour of active position of each directional valve is turned to violet. For clarity purposes, the figures in this paper animate active valve position by gray background colour, as shown in Fig. (17). Thus the user is able to follow how the directional valves are operating through the circuit operation.
- Each limit switch plunger is moved graphically when a piston, as shown in Fig. (10-a) presses it.
- The status of any type of logic device used in this work can also be observed during operation. At Run-Time mode a binary number – "0" or "1" – is printed at each logic valve input port. This number reflects the input of that port, "1" for pressurizing and "0" for depressurizing. Similarly, a binary number is also printed at output port to reflect its output status. Figure (18) illustrates the animation method used.
- The operation of the (3/2) signal generating directional valve can be observed by the colour of its operating state box. When there is no pressure at its input port, the active switching position is the right one as indicated by Fig. (19-a). When a pressure is set at its input port, the active switching position will turn to the left state box after a preset delay time. This is indicated by the change to violet colour (here, gray background) of the state box, as shown in Fig. (19-b).
- The time delay valve has a text box that indicates its delay time setting. At Run-Time mode, when there is no pressure at its input port, the active position of the (3/2) directional valve is the right one, as shown in Fig. (20-a). When a pressure is applied at the input port of the time delay valve, a new text box will appear under the main text box as shown in Fig. (20-b). This text box displays continuously decreasing number starting from the set delay time (displayed by main text box) to zero. The number displayed by the lower text box is the time that should elapse before compressed air is allowed to pass to output port. During the count down, the active position of the (3/2) directional valve is the right one and the colour of the count down text box is red, as shown in Fig. (20-b). At zero count down time, the text box colour turns to green and the valve's active position turns to the left side, as shown in Fig. (20-c). The compressed air now passes to the output port. This state is maintained until the input pressure is removed where the valve returns to its initial state, as shown in Fig. (20-a). In addition, the valve returns to its initial state if input pressure is removed before end of delay time is reached.
- The colour of any shortcut source is turned to violet when the output linked to it is pressurized. The colour of any shortcut linked to this source which is

designated an input to any device also turns to violet. Figure (21) illustrates this type of animation.

- For large circuit size, switch bar can be used to simplify changing the state of any lever or push-button used in workspace.
- The user can further control the Run-Time pace with the "Step-By-Step" check box and "Next step" button (Fig.2). If the check box is checked during the Run-Time mode, then the designed circuit can be run step-by-step, and the "Next step" button should be used to advance the circuit operation one step forward. The "Step-By-Step" check box can also be used to pause the running of the circuit.

3.PACAD as a Learning Tool

For learning (particularly, self-learning) use, the monitoring features of the simulation program allows visualizing of the states of components used during circuit running. In addition, this is a powerful tool for guided training of teachers, students, and engineers. It can be used to save valuable time for new engineers to gain design skill.

This package can also be of big aid to a skilled designer. He can use the package to test new design ideas. He can initially construct the designed system using the simulation program. The design can be tested and if errors are indicated, the errors can be easily tracked and corrected until the new design idea is proved.

To illustrate this, a sequential circuit will be tested at Run-Time mode through a step-by-step operation. The circuit is designed by intuitive method with the sequence (1^+, 2^+, 2^-, 1^-). A circuit with such a sequence is designed as in Fig. (22-a). However, a deliberate mistake is made by unavailability of the signal breaker (signal generating valve) at the input of activation signal of (2^+) operation.

Figure (22) illustrates the monitoring features of the simulation program when it is run step-by-step. Briefly:

- Figure (22-a) illustrates the initial state of actuators and valves used. Cylinders (1) and (2) are both retracted, and so plungers of limits switches (1-) and (2-) are pressed.
- Figure (22-b) illustrates that the signal-generating valve is switched to the left position to turn off the activation signal of (1^-) operation at port "Z".
- When Start/Stop lever is operated, as shown in Fig. (22-c), pressurized air passes through "Start/Stop" valve and (1-) limit switch to activate the (1^+) operation to extend cylinder (1).
- The plunger of limit switch (1-) is released when the piston of cylinder (1) is extended which presses the plunger of limit switch (1+), as shown in Fig. (22-d).
- Limit switch (1+) activates the (2^+) operation which causes cylinder (2) to extend, as shown in Fig. (22-e)

- The plunger of limit switch (2-) is released by the extension of the piston of cylinder (2). The latter presses the plunger of limit switch (2+), as shown in Fig. (22-f).
- The signal coming from limit switch (2+) should activate the (2⁻) operation at port "Z". However, due to the presence of the signal at port "Y", the activation is halted because the valve fails to change its status, as shown in Fig. (22-f).

The conflict in actuation signal requirement (presence of signals at ports "Y" and "Z" simultaneously) is detected by the internal valve logic built in the simulation program. It is up to the designer to deduce the cause of failure. This should not be a difficult task if the designer makes use of the visual animation of the valve status. It is obvious that this problem can only be overcome by breaking the signal at port "Y".

4 .Software Implementation Aspects

As mentioned before, this package is implemented by using the Microsoft Visual Basic 6.0 programming language, and the following sections illustrate key aspects of the software implementation [7].

4. 1 .Components as ActiveX Control

An ActiveX control is an object that the programmer can place it on a form to enable or enhance a user's interaction with an application. ActiveX controls have interface elements, which are properties (variables), methods (functions), and events (actions done by the mouse or the keyboard). These controls have an ".ocx" file name extension [6].

The components used in the PACAD package are built as ActiveX controls (Table 1). Table (2) illustrates the interface elements of the "DirValve.ocx" as an example. Most of the ActiveX controls in Table (1) have the method "Job" to do their action according to their input(s).

4. 2 .How to Connect the Ports

There is a random access file "Connections.con" used for saving the connection between any two ports. Each record in this file consists of six fields as illustrated in Table (3). The first two fields are used to identify the source component, and the third field is used to identify the output port in this component. The last three fields are used to identify the target component and its input port.

4. 3 .Run-Time Mode Procedure

This procedure simulates the operation of the pneumatic sequential circuit designed in the workspace. This procedure is executed when the user clicks at the "Start" button. Figure (25) illustrates the flowchart of this procedure.

4. 4 .The Function of New, Save, and Open Buttons

The "New" button is used to start with a new circuit by unloading all the ActiveX controls in the workspace, and deleting the content of the "Connections.con" file.

The "Save" button is used to keep the designed circuit in the workspace by storing the following items in a data file:

1. The location of each ActiveX controls in the workspace, and their static properties (the properties that are not changed after loading their ActiveX controls).
2. The content of "Connections.con" file.
3. The zoom value.
4. The locations of the horizontal and the vertical scroll bars.

The "Open" button is used to reconstruct a pre-designed circuit in the workspace by opening the corresponding data file and then executing the following items:

1. Unloading all the ActiveX controls in the workspace.
2. Deleting the content of the file "Connections.con".
3. Loading the ActiveX controls, which are stored in the data file, and placing them in their locations in the workspace, and then setting the values of their static properties.
4. Setting the content of the "Connection.con" file.
5. Setting the zoom value.
6. Setting locations of the horizontal and the vertical scroll bars.

5.Conclusion

The implemented package is easy to use through the friendly built-in graphical-user interface. The user can build a circuit and test it in a simple manner by the simulation program. If there is a contradiction, the internal logic built in the simulation program will halt the step-by-step sequence run. The user can make use of the colouring features to simplify error tracking.

The simulation program of pneumatic automation is an essential and very useful tool to aid in the design and testing of sequential pneumatic circuits. Corrections, modifications, and checking can be made before the practical realization of the system.

References

[1] Stoll, K., "*Low Cost Automation (LCA) Through Special Purpose Machines (SPM) with the Help of Pneumatics*", Proc. Inter. Conf. on advance in manufacturing ocl., Singapore, pp. 111-118, 9-11 Oct. 1984.

[2] "*Matlab*", 2000. http://www.glue.umd.edu/~nsw/matlab.doc/buildgui.pdf

[3] Jacic'. Lj., "*Computer-Aided-Design of Pneumatic Digital Systems- A System Approach*", Fluid Control And Measurement Vol.2, pp.95-100, 1986.

[4] Al-Timimi, A.M., "*CAD Package for Pneumatic Automation*", M.Sc. thesis, University of Technology (UOT), Baghdad, Iraq, 1990.

[5] "*Automation Studio 3.0*", 2002. www.automationstudio.com/plc/plc.pdf

[6] Siler, B. and Spotts, J., "*Using Visual Basic 6*", Que, 1998.

[7] Zeyd, F.A.S., "*Development of a Pneumatic Automation CAD Package with Discrete Actuators*", M.Sc. thesis, UOT, Baghdad, Iraq, 2002.

Fig. (1) The splash screen of the package.

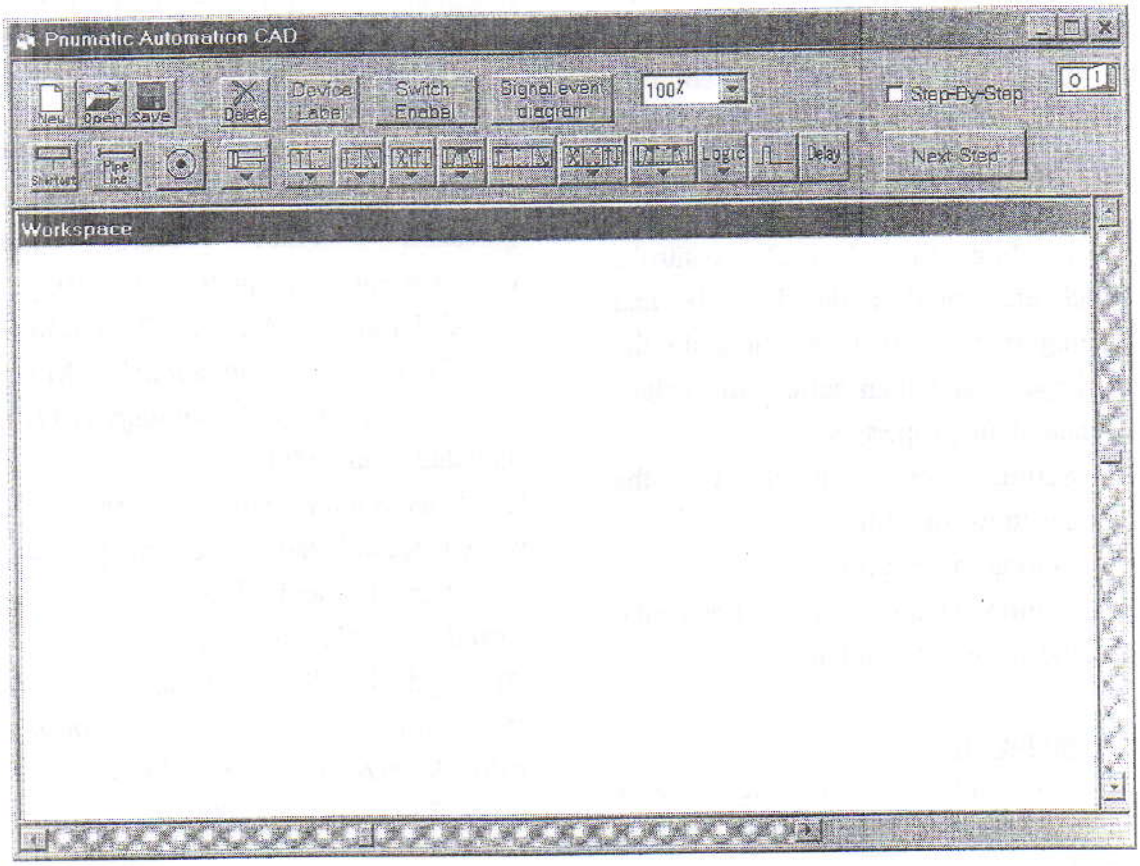
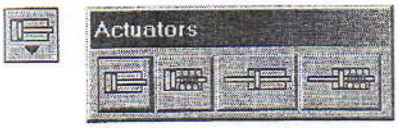


Fig. (2) Main window of the package.



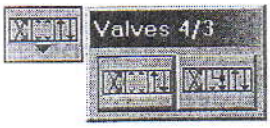
a- Cylinders group.

b - Group of (2/2) directional valves.

c - Group of (3/2) directional valves.

d - Group of (4/2) directional valves.

e - Group of (5/2) directional valves.



f - Group of (4/3) directional valves.

g - Group of (5/3) directional valves.

h – Logic valves group.

Fig. (3) Group-buttons and their contents.

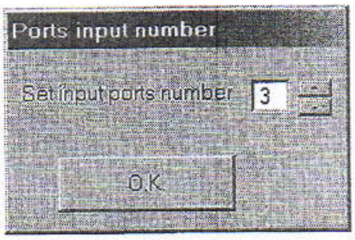


Fig. (4) Dialog box for Multi-AND and Multi-OR gates.

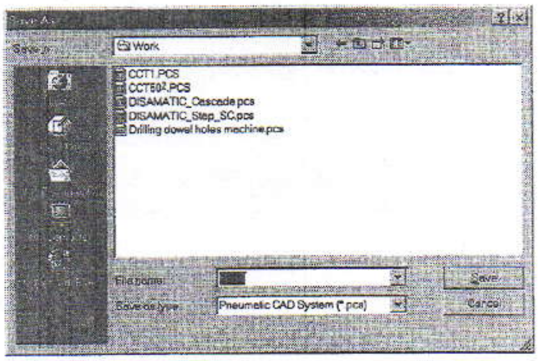


Fig. (5) Dialog box for saving a file.

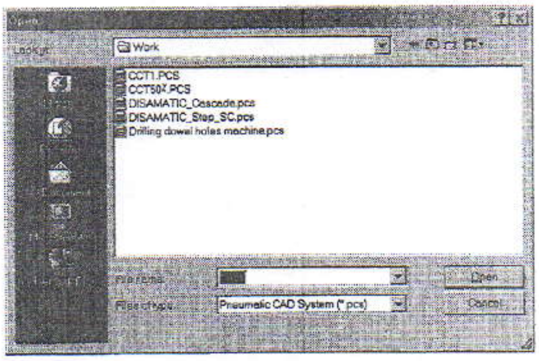


Fig. (6) Dialog box for opening a file.

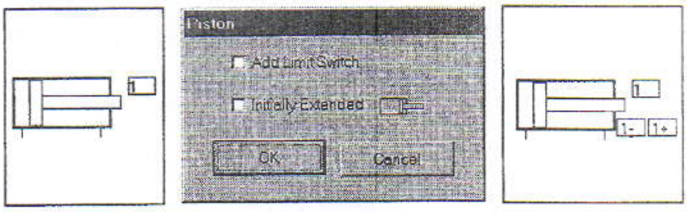
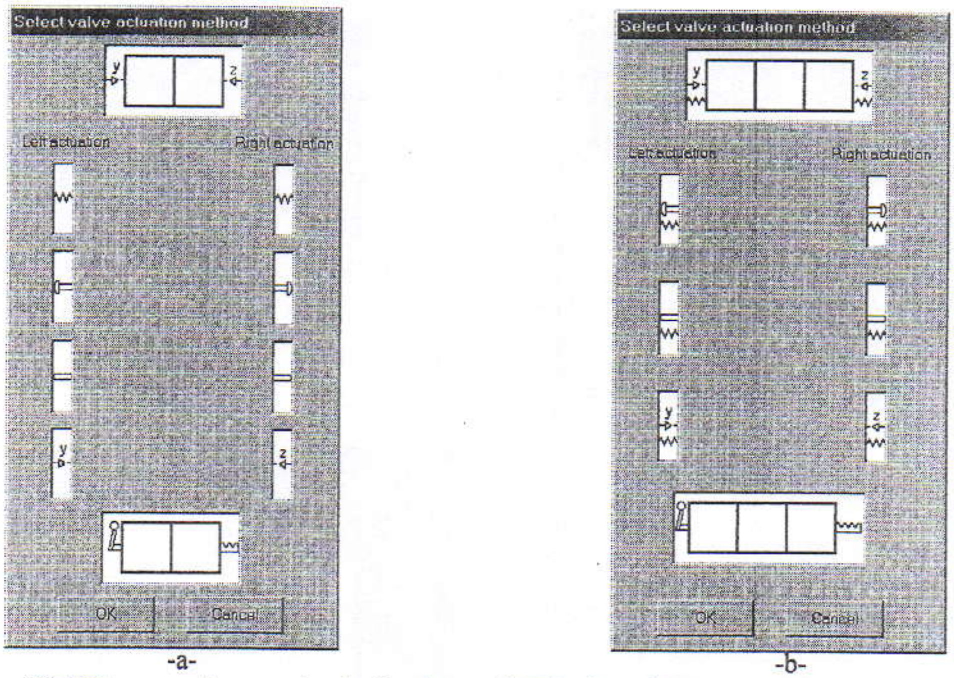


Fig. (7) Edit cylinder, a) Cylinder symbol without limit switches, b) Cylinder dialog window, and c) Cylinder symbol with limit



-a- -b-

Fig. (8) Edit actuation method of valves, a) Window for two positions valves, and b) Window for three positions valves.

-a- -b-

Fig. (9) Plunger dialog window.

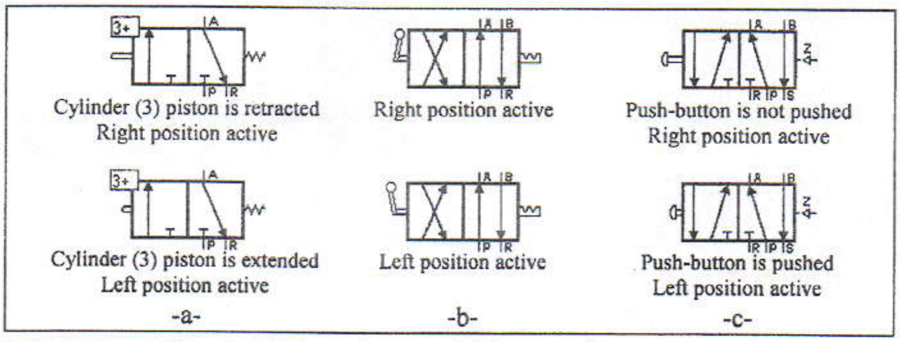


Fig. (10) Animated actuation methods, a) Plunger, b) Lever, and c) Push-button.

-a- -b-

Fig. (11) Delay time valve, a) Symbol, and b) Dialog window.

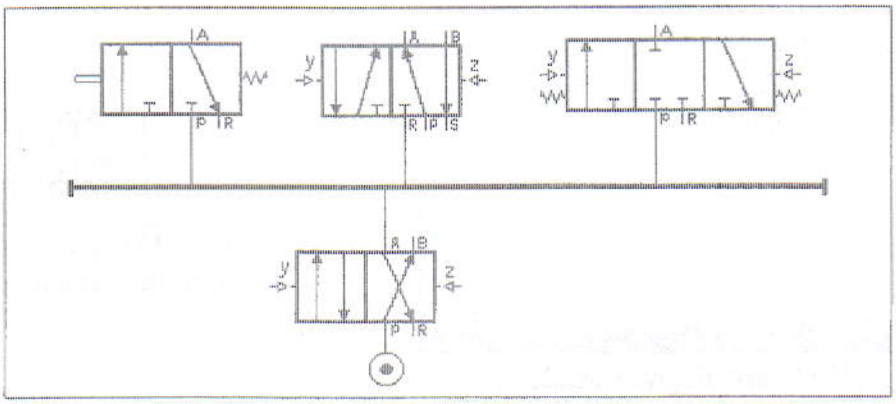
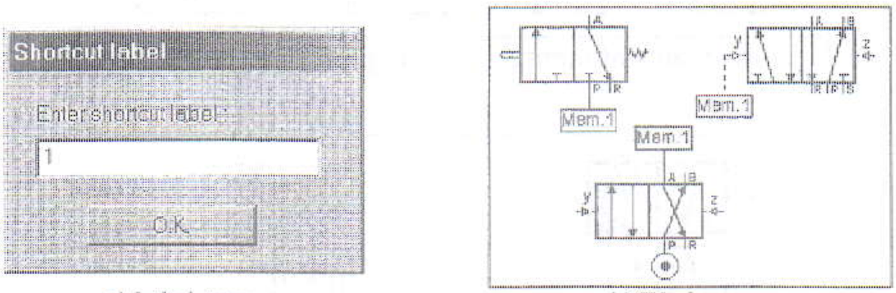
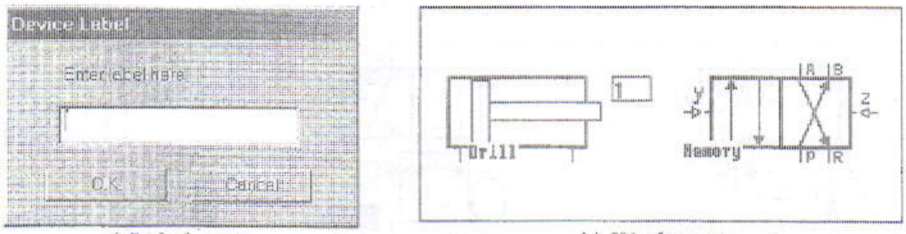


Fig. (12) Pipe line use.



a) Label entry. b) Workspace.

Fig. (13) Shortcut application.



a) Label entry. b) Workspace.

Fig. (14) Label writing.

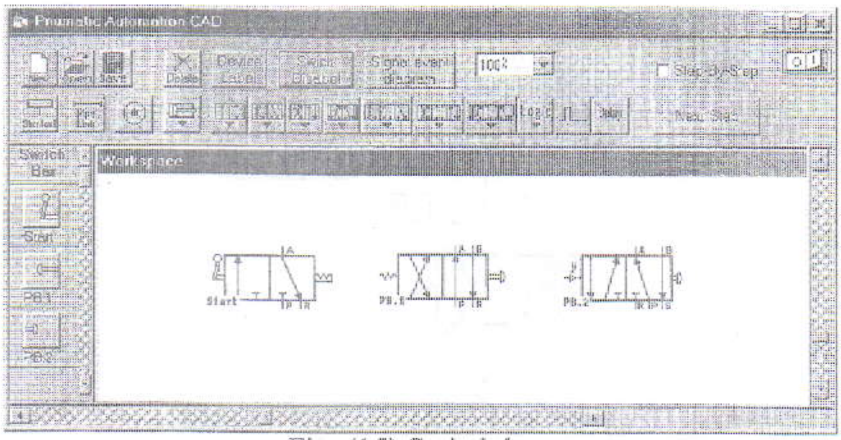


Fig. (15) Switch bar.

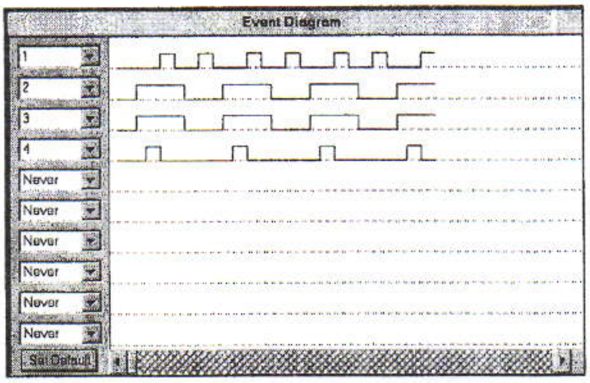


Fig. (16) Event diagram window.

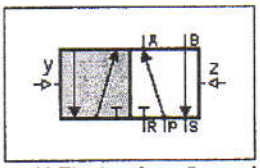


Fig. (17) Active-Inactive valve state visual message.

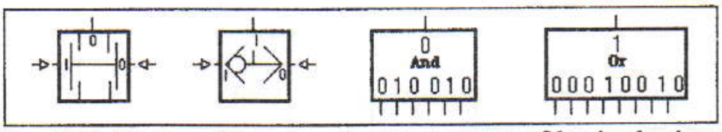
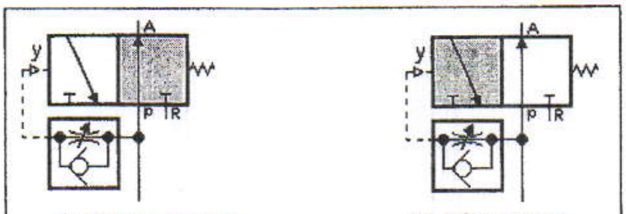


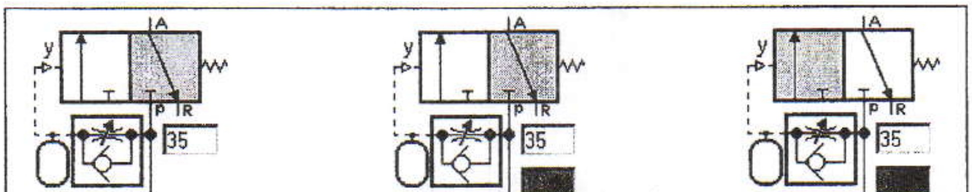
Fig. (18) Status of input and output ports of logic devices.



a) During preset delay time.

b) After preset delay time.

Fig. (19) Signal generating valve state visual message.



a) Port "P" has no pressure.

b) Port "P" is pressurized, and (22) seconds are elapsed.

c) End of delay time.

Fig. (20) Time delay valve state visual message.

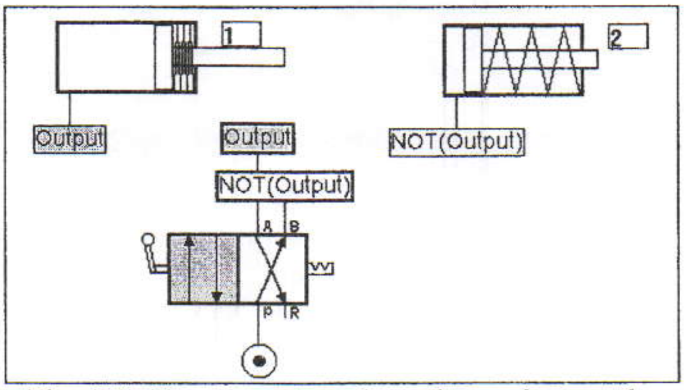


Fig. (21) Shortcut colour through run-time mode.

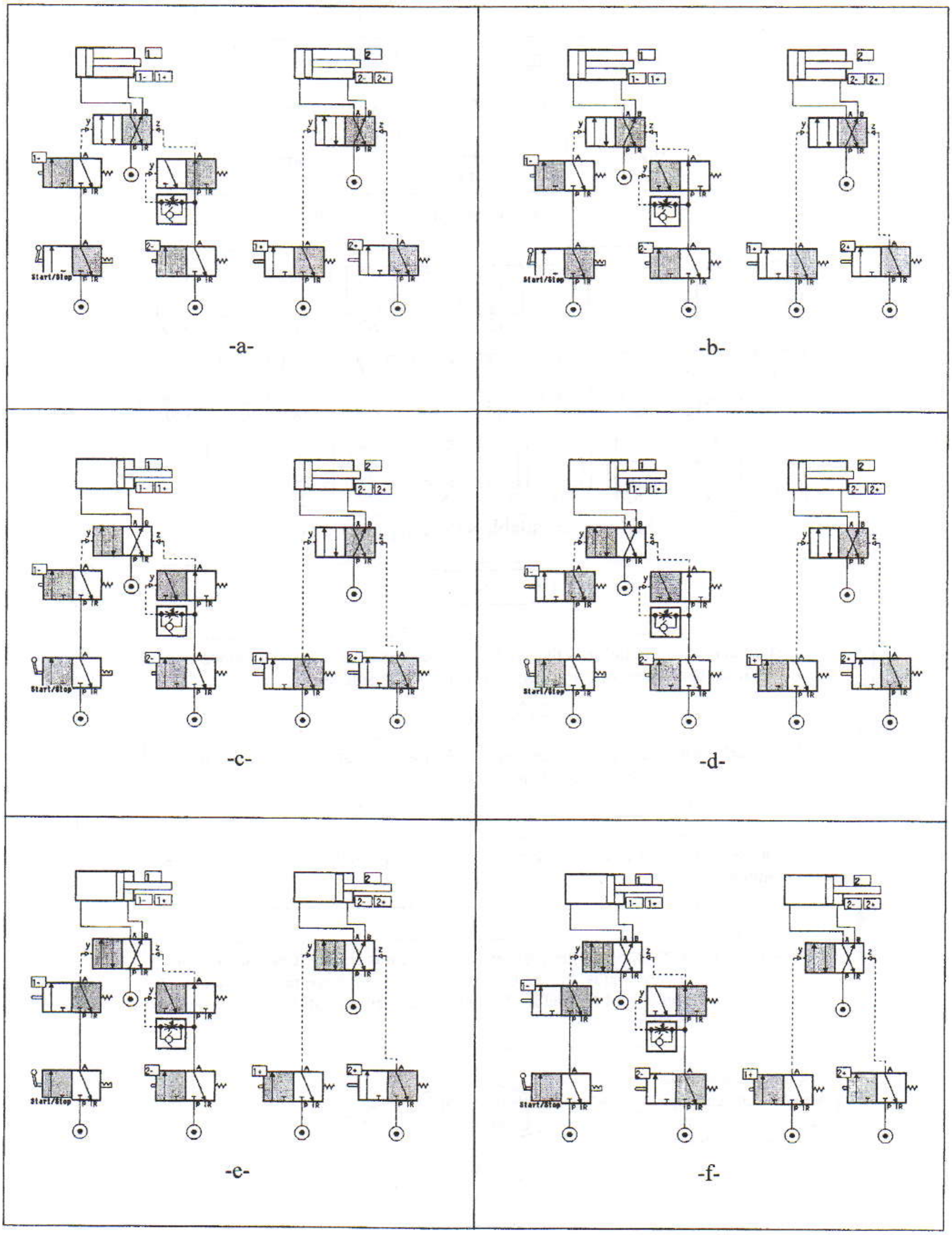


Fig. (22) Step-by-step operation of sequential pneumatic circuit.

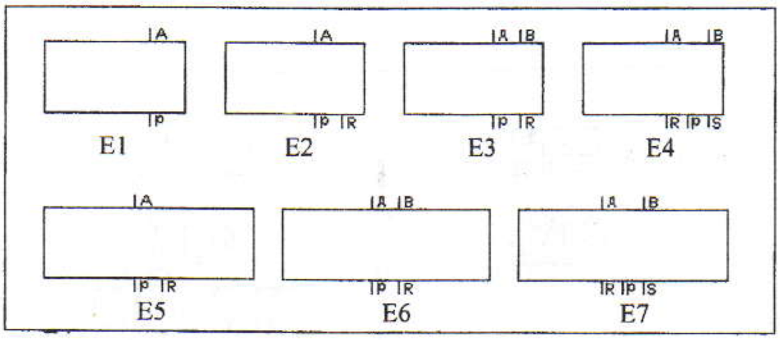


Fig. (23) Directional valves envelopes.

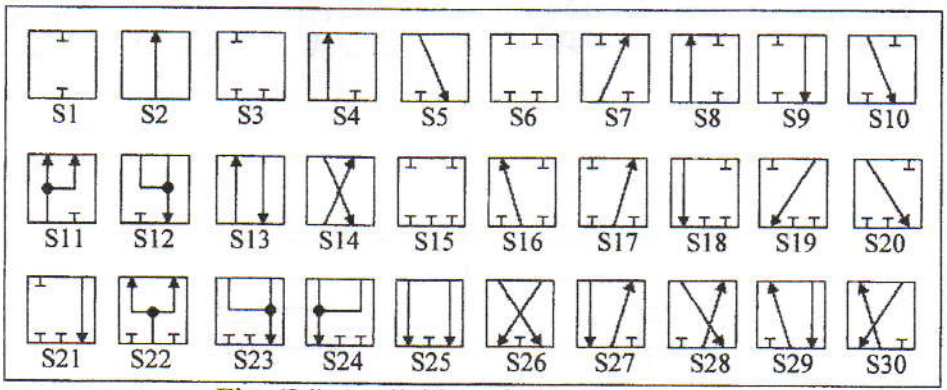


Fig. (24) Available switching positions

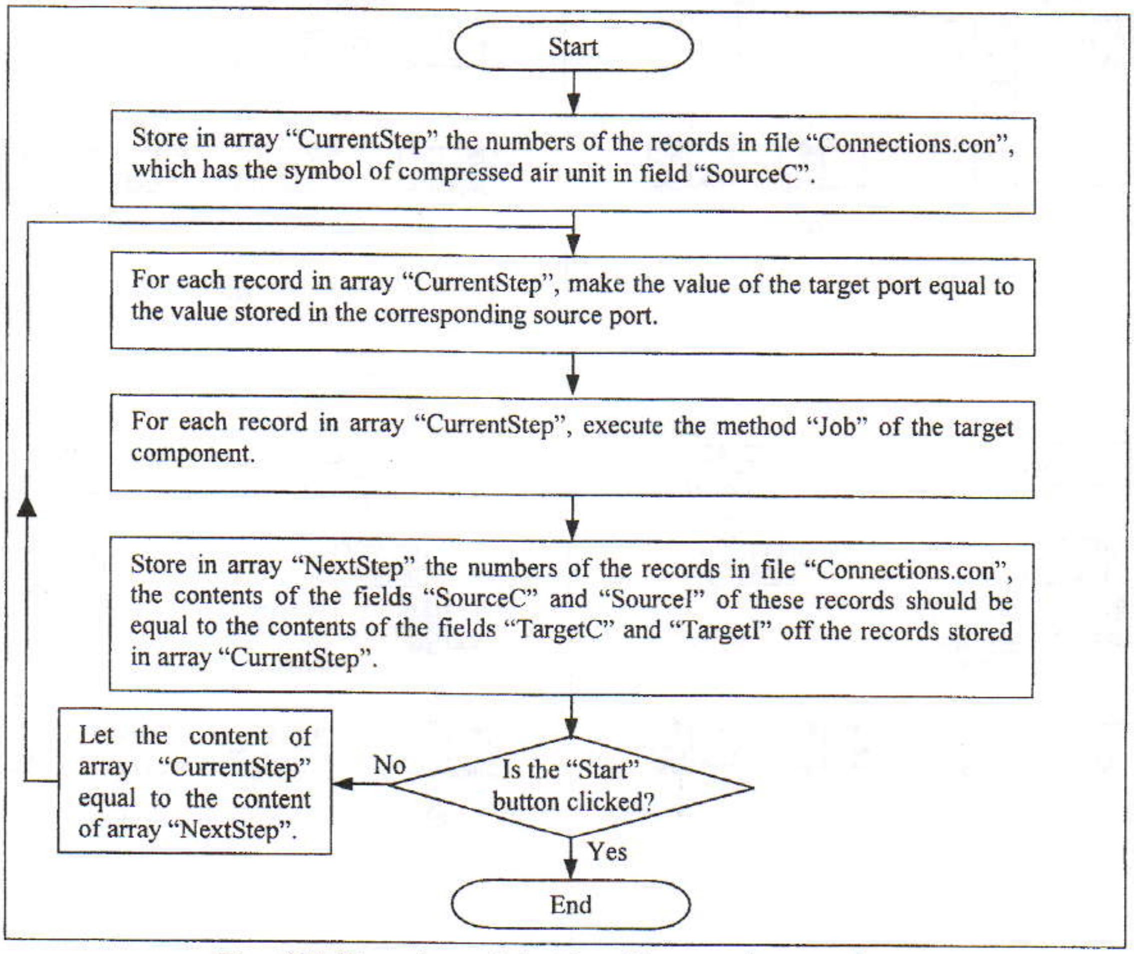


Fig. (25) Flowchart of the Run-Time mode procedure.

Table (1) The constructed ActiveX Controls.

	ActiveX Controls	Comments
1	ShortCut.ocx	Simulate the Shortcut connection.
2	PipeLine.ocx	Simulate the pipe line.
3	AirSourse.ocx	Simulate the compressed air source.
4	Cylinder.ocx	Simulate the four cylinder types.
5	DirValve.ocx	Simulate all the directional valves used in the package.
6	Logic.ocx	Simulate the AND and OR gates.
7	MultiLogic.ocx	Simulate the multiAND and multiOR gates
8	SigGen.ocx	Simulate the signal generating valve.
9	DelayValve.ocx	Simulate the time delay valve.
10	ConnLine.ocx	Simulte the connection line between the ports.

Table (2) The interface elements of the "DirValve.ocx".

	Interface Name	Interface Type	Comments
1	Envelope	Property	Byte (1 to 7): Detect the envelope type of the directional valves illustrated in Fig. (23).
2	SP1 SP2 SP3	Properties	Integer (1 to 30): Detect one of the switching positions shown in Fig. (24).
3	LeftAM RightAM	Properties	Byte (0 to 7): Detect one of the available actuation methods shown in Fig. (8).
4	ActiveFlag	Property	Byte (1 to 3): Indicate the active switching position during the Run-Time mode in the package.
5	A B R P S Y Z	Properties	Byte (0 to 2): Indicate if the status, where: 0: No pressure , 1: Under pressure , 2: Trapped air.
6	Draw	Method	This function is used to draw the directional valve according to the information stored in the properties (1,2, and 3).
7	F1 F2 ...... F30	Methods	Each function represents the task of on of the switching positions shown in Fig. (24). These functions set the values of properties "A" and "B", which represents the output ports of the valve. For example the function "F7" code (which simulate the operation of "S7" in Fig. (24)) is: B=P IF A<>0 THEN A=2
8	Job	Method	This function do the following three tasks: 1. Set the property "ActiveFlag" according to the properties "Y" and "Z". 2. Make the colour of the active switching position (the value of property "ActiveFlag") is violet. 3. Call one of the methods (F1, F2, ..., F30) depending on the active switching position.
9	Click	Event	This event is used in the main program to activate the lever and push button of the directional valves in the workspace.
10	DoubleClick	Event	This event is used in the main program to change the actuation method of the valve. It is also used to make the plunger senses the cylinder's piston position.
11	MouseDown	Event	These two events are used in the main program to change the valve location in the workspace. They are also used to connect the valve ports to other ports.
12	MouseMove	Event	

Table (3) The fields of the random access file "Connections.con".

	Feilds	Comments
1	SourceC	It is an integer value to indicate the type of the source component, which has the output port.
2	SourceI	It is an integer value represents the index of the source component in workspace.
3	SourceP	It is an integer value to indicate the output port in the source component.
4	TargetC	It is an integer value to indicate the type of the target component, which has the input port.
5	TargetI	It is an integer value represents the index of the target component in workspace.
6	TargetP	It is an integer value to indicate the input port in the target component.