LADSIM Demonstration Guide

For those who hate Manuals …

We realise that most people only look at manuals as a very last resort and, if you are very short of time and would like a quick 'taster', then just stick the disk in and try it. However, if you have a few minutes to spare, then the tutorial given here will take you through the main features of the program.

Installation

To install LADSIM , you will require:

>   386 processor or above
>   4MB RAM as a minimum
>   4MB Hard Disk Space
>   VGA monitor
>   Mouse
>   Microsoft Windows version 3.1 or later.
>   (*__Note__ : Problems may be experienced if you use earlier versions*.)

To install LADSIM place the DEMO disk into a 3.5'' drive, ensure that Windows is running and select the **RUN** command from the **F**ile menu of Program Manager or, if you are using Windows 95, select **RUN** from the Start Menu

Type: **A:SETUP.EXE**
        (replace **A**: with your floppy drive if different).
Click on the **OK** button.
Follow the on-screen prompts.

Introduction to LADSIM

This Demo Copy contains all the features of the full program with the exception of the Save and Print commands. It may be used as many times as you like, however it will not run for more than 25 minutes.

LADSIM has three main features, namely:

1. It enables the PC to simulate a PLC such that Ladder Logic PLC programs can be built up on the screen in a simple 'Drag and Drop' manner.

2. These PLC programs can then be used to control one of seven different 'on screen' simulations of control problems including a Drinks Dispensing Machine and a Lift (Elevator). Of course, if the program that the student has written is incorrect then the simulation will not work properly!

3. After the student has become familiar with the programming techniques used for a PLC by controlling the simulations s/he can then move on to control real external items of equipment via a Bytronic Interface if required. Bytronic manufacture a range of products designed for this purpose including conveyor systems, traffic control units, rotary transfer units and many more. Please contact your supplier for more details.

In order to illustrate the potential of LADSIM the procedure for building up a Ladder Program to control the Car Park simulation is given over the next few pages. It is not suggested that the solution given is the best one but it serves as a good example. If you follow the procedure you will become familiar with many of the features of the program.

Procedure for producing a Ladder Logic Program for the Car Park Simulation

1. To start the program, double click on the LADSIM icon.

2. Click on the <u>Simulate</u> menu and then on the <u>Car Park</u>. You will be presented with the car park simulation screen for which you are about to develop some ladder code. The simulation objectives and identification of the various input and output signals used can be found in LADSIM's help facility. You can quickly access this by pressing the F1 key. After you have finished viewing the help screen return to the simulation. You will also see the I/O Browser and part of the Ladder Diagram, but we will consider these later.

3. Click on the car and two pointing hands will be displayed. Click on the left hand and the car will go left and try to get into the car park. It will break the input barrier beam (IP0) but the barrier will not lift The first part of our program must therefore make the barrier open when the beam is broken. At the bottom of the I/O Browser are controls similar to those of a Video Recorder. These control the execution of the Ladder Program. At present the program is running (even though there is nothing in it yet) so click on the stop button and you are returned to the Ladder Logic Diagram.

4. The input barrier beam (IP0) is a **N**ormally **O**pen **C**ontact (**NOC**) in this particular application. If you move the mouse cursor slowly over the symbols in the **Controls** box you will see that NOC is displayed over the top left symbol. Hold down the mouse button and drag the symbol to the left hand side of Rung 0 and release it. You will then be asked via the **Inputs** dialog box to select the identification of this input. We want IP0, so click OK and this will be displayed on the ladder. When IP0 is closed we want the barrier to open so drag the symbol for an output to the right hand side of the rung and identify it as OP3, i.e. the barrier, by clicking OP3 in the **OUTPUTS** dialogue box and then clicking OK. We now have the first rung of our ladder as shown:
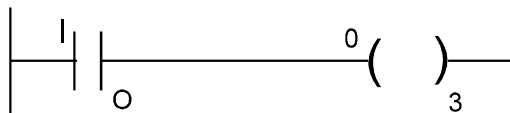
Lets try it and see if it works by following the simple step given in (5).

5. Select the Car Park simulation again and send the car into the car park. You should see that the car goes quite happily into the car park but unfortunately the barrier closes before the car has entered properly so the car would be damaged. Obviously we can't allow this so we need to 'latch' the barrier open. Thus, return to the ladder editor by clicking the 'stop' button as before and, from the Controls button menu, click on Del Control. Follow the instructions at the bottom of the screen and delete OP3. Replace it with a latched version (OPL). Your ladder diagram should now look like this:
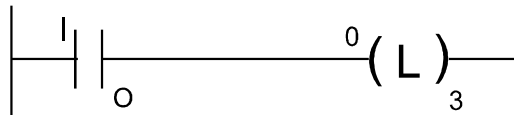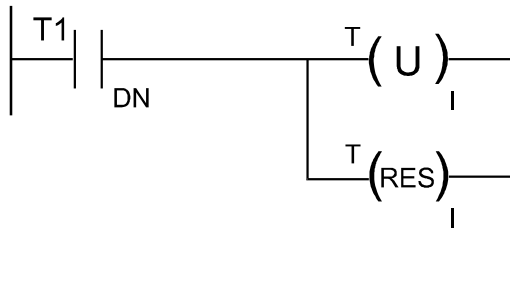

6. Now try this version. You will see that the barrier stays open and the car is not damaged. However, following cars can now get in free of charge and the cars can get out of the entrance! Click on a car that is in the car park to see what happens. Thus we need to add a timer to control the amount of time the barrier stays latched open. As an intermediate step we will add a timer, see how it works and then use it to control the barrier. (Return to the ladder editor.) Click on the Add Rung button to get a second rung. We want the timer to start when the IP0 beam is broken so drag IP0 to the second rung, as with the first, and drag a Timer to the right hand side. Choose T1 and Preset the time delay to 3 seconds in the dialogue box that pops up, then click on OK. We should now have:
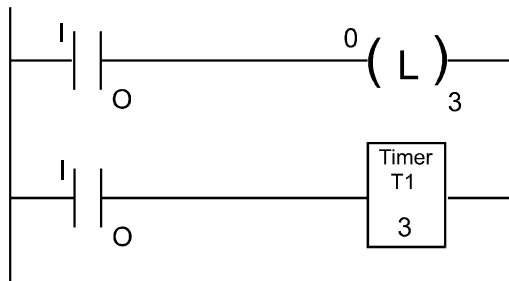
7. Run the simulation as before but click on the Timers card on the I/O Browser. When a car breaks the beam, notice the Accumulator (Accum) display for the timer increases by one and, after 3 seconds, the DN bit is set. We now have a timer which starts timing when the IP0 beam is broken and the T1/DN bit is set 3 seconds later. We can thus add a rung which will unlatch OP3 when the T1/DN bit is set. So, add a rung as before and drag a NOC to the left hand side. Identify it as T1/DN from the dialogue box. On the right hand side drag a OPU (output unlatch) and identify it as O/P3 thus:
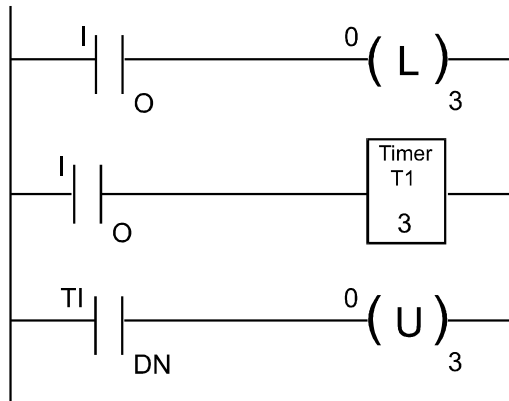
8. Try the new program and you should see that it works OK, but only for the first car. Why is this? You can get some idea by looking at the ladder diagram that is displayed on the same screen as the simulation. You will notice that active elements are highlighted and, in particular, you will notice that the T1/DN input is not highlighted until 3 seconds after the first car goes through. The T1/DN is then permanently set and no other car is allowed through. You can check this by clicking the Re-wind button, looking at T1/DN then clicking the Play button and sending a car into the car park again and observing that T1/DN becomes highlighted 3 seconds later. In order to overcome this problem it is necessary to Reset the timer at the same time that the barrier is unlatched. Thus, add a Reset to the right of rung 2 and identify it as T1. You will notice that the Reset is automatically added in parallel with the Unlatch thus:

```
        T1
       ┌─┤ ├──────────────────────┬─────( U )────────┐
       │  DN                      │      T     I      │
       │                          │                   │
       │                          └────(RES)──────────┤
       │                                 T    I        │
                                                       │
```
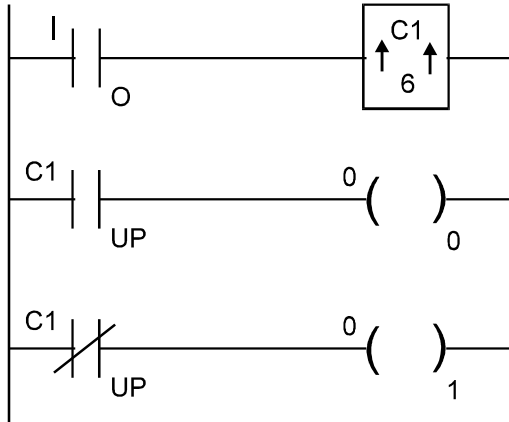
9. Try this and you should see that it works well. If not, try altering the time delay for T1. Congratulations, you've now solved the problem of getting cars into the car park. The problem now, however, is that the cars are trapped in the car park! Click on one and see what happens. This can be solved by adding a similar bit of program for the output barrier. If you add this you should now have a working car park, with a bit of luck!
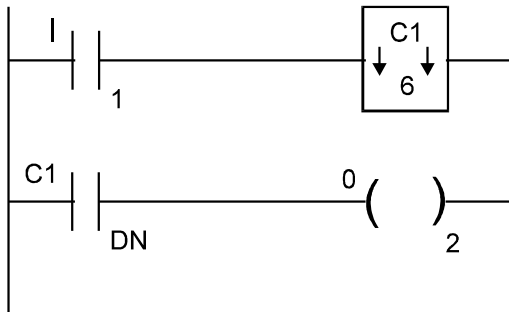
10. You should have noticed that the car park has a number of displays describing various aspects of the car parks function. The next problem is to control these Car Park displays. In LADSIM, the accumulator of Counter 1 is directly connected to the Count Display. This is equivalent to a connection made directly to a counter within a PLC and is normal practice when required.

11. Thus, if we use Counter 1 (C1) in the 'count up' mode such that every time the entrance barrier beam is broken it triggers the counter, the count will be displayed. Also, if we set the preset to 6 to represent the capacity of the car park, when the C1/UP bit is set, i.e. 6 is reached, we can turn on the Full display (OP0). Conversely, this same bit (or flag) can be used to control OP1 such that, when it is not set, the SPACES display is turned on.

12. To implement the above, add a rung and put IP0 on the left and a counter, identified as C1 on the right. C1 should have a preset of 6 and be an UP counter. Add another rung with a NOC on the left identified as C1/UP and OP0 on the right. Finally, add another rung with a NCC on the left identified as C1/UP, and OP1 (SPACES) on the right:

```
 I                                        ┌─────────┐
─┤ ├─────────────────────────────────────│ ↑  C1  ↑│──
   O                                      │    6    │
                                          └─────────┘

 C1                                          0
─┤ ├──────────────────────────────────────( )──
   UP                                        0

 C1                                          0
─┤/├──────────────────────────────────────( )──
   UP                                        1
```

13. If you try this you will find that the FULL, SPACES and COUNT displays work but only when the cars are entering the car park. When a car exits, the count is not changed. This can be rectified by using the exit beam to trigger the same counter but this time in the 'count down' mode. We can also use the C1/DN flag to control the EMPTY display (OP/2):

```
 I                                        ┌─────────┐
─┤ ├─────────────────────────────────────│ ↓  C1  ↓│──
   1                                      │    6    │
                                          └─────────┘

 C1                                          0
─┤ ├──────────────────────────────────────( )──
   DN                                        2
```

14. When you try this you will find that the program works well until we allow 7 cars into the car park. The last one cannot find a space so it exits and the counter counts down to 5. This is obviously wrong as there are still 6 cars in the car park.

15. To overcome this final problem we must prevent cars entering the car park when it is full. Thus we could add a condition to the first rung that the FULL display was not on by modifying it as shown below.

```
  |    I      0       0            |
  |---| |---|/|------( L )---------|
  |     O     0        3           |
```
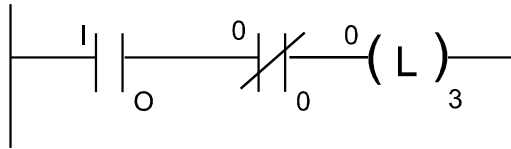
16. We must also modify the second rung (rung1) in a similar way.

17. You have now completed a ladder program to control the car park simulation. The final version of the program is given at the end of this document. We hope that you have found this demonstration exercise useful. We think it demonstrates many of LADSIM's features. There are, however, many other features of LADSIM that we have not covered. For example :

a) Comments can be added to every rung allowing you to clearly document your program - click the RUNG COMMENTS button on the tool bar.

b) Programs can be debugged using the DEBUGGER before attempting control of simulations or external devices - the debugger is found on the SIMULATE pull down menu.

c) The I/O BROWSER is a useful supplement to the debugger allowing you to view a particular type of control.

d) LADSIM also includes the use of FLAGS and SHIFT REGISTERS.

e) Your ladder diagrams can be printed with and without comments.

f) LADSIM includes another 6 simulations that will test your students ability to write control programs.

g) LADSIM includes a direct link to the Bytronic range of interface cards that will allow you to control real external plant via the interface card.

LADSIM Ordering Information

LADSIM, the Ladder Logic Editor and PLC Simulator is available in the UK direct from Bytronic or one of our Authorised Distributors if you are outside the UK.

If you would like to purchase LADSIM or require a written quotation, please mail or fax your purchase orders to Bytronic or one of our representatives.

Bytronic International Ltd. manufacture a wide range of Educational Training Equipment in the field of engineering. Areas covered include :

> PLC Control Systems
> SCADA Control
> Process Control
> Fuzzy Logic
> Microprocessors
> Control Applications
> and many more ……

A full-colour catalogue is available free on request from Bytronic. Simply pick-up the phone and call us.

> Bytronic International Ltd.
> The Courtyard,
> Reddicap Trading Estate,
> Sutton Coldfield,
> West Midlands.
> B75 7BU.
>
> Tel. +44 (121) 378 0613
> Fax. +44 (121) 311 1774
> Internet: http://www.bytronic.co.uk
> Email: sales@bytronic.co.uk

LADSIM - Suggested Solution for Car Park Simulation
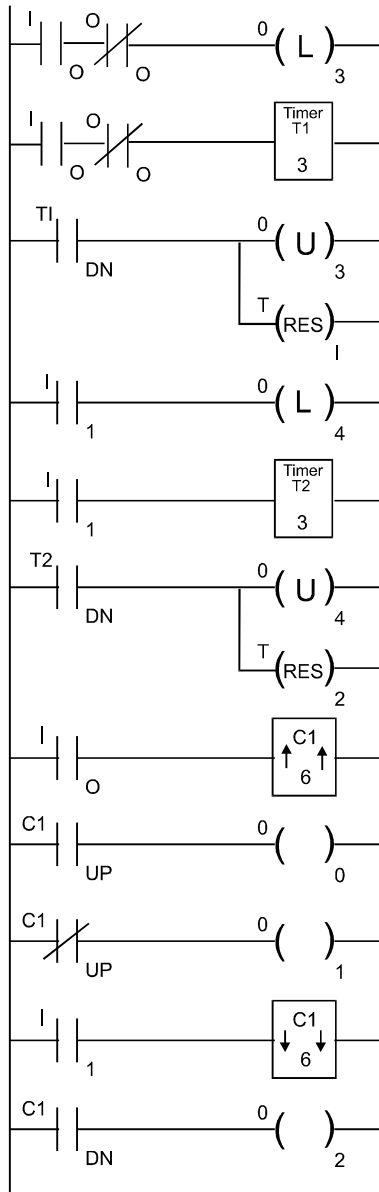
N.B. This is not the solution to the competition. An alternative program has been written to challenge your Ladder Logic knowledge.