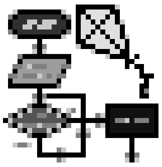
Secondary School Activities for Flowol™



By: Rod & Anthony Bowker

System Requirements for the Secondary Mimic Disc:

- Flowol 2 or later.
- 2Mb of free Hard Disc Space
- Microsoft Windows 95, 98, NT4, 2000 or XP
- We recommend that all the latest critical updates are installed from http://windowsupdates.microsoft.com

Installation Notes:

1. Flowol 2 or later needs to be installed before you can install the Secondary Mimics.
2. You need to be logged on as an Administrator to install these mimics
3. This CD contains an msi installer. This can be assigned to workstations as a package. (There are no shortcuts with the mimics).
4. If the Flowol program was open (in use) before installation, exit and restart Flowol before trying to use the mimics.

Activities

These activities are a pupil based text to support some examples in the Flowol tutorial and those on the Secondary Mimic Disc. The controllable tasks are progressive, similar to those in the tutorial, but focus more on the problem solving aspects of the examples rather than acquiring the skills to use the software.

The examples explore the programming needed to operate a zebra crossing and a set of traffic lights. The multiple functions of a lighthouse, which respond to an input, are then considered and also the use of subroutines to keep the programming compact.

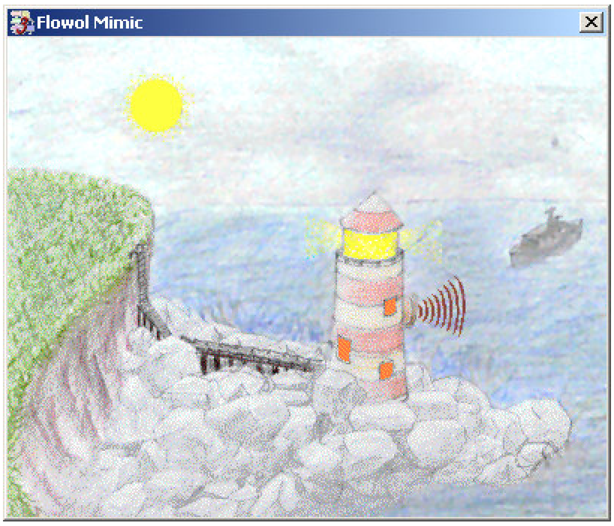
The students are then encouraged to use their acquired skills to manipulate a pelican crossing, a railway level crossing and explore the automatic systems which might be possible in the home. Analogue values for light level and temperature are introduced to the auto-home and also applied to a greenhouse. Variables are finally considered to count vehicles as they pass through automatic car parking barriers.

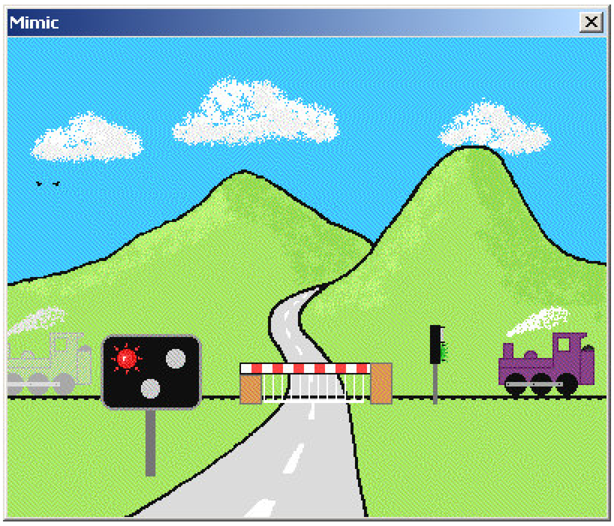
Other mimic packs are available, check http://www.flowol.com for details.

Let *Flowol*™ help you take control.

Automatic systems are all around us; keeping us safe, making life comfortable and helping us with difficult and unpleasant tasks.

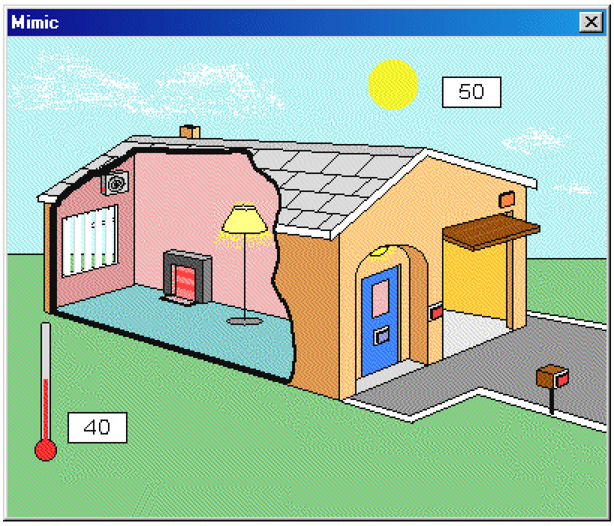
Flowol will allow you to produce your own solutions to many of these situations.
We will start by guiding you through the tasks to control traffic signals and warning lights.



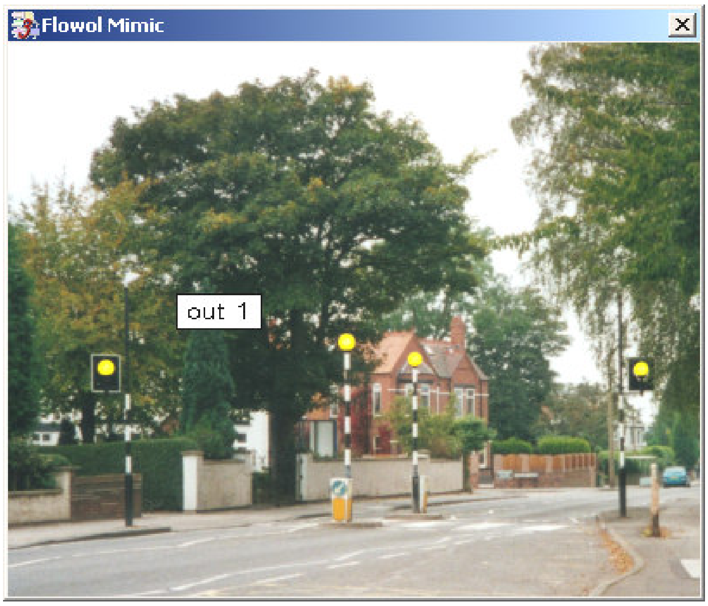


With your skills, you will soon be able to solve more complex examples such as an automatic level crossing or the control systems needed to help people in their homes.

What other situations can you think of where automatic control might help?



Zebra crossing. This picture shows a fairly quiet road.



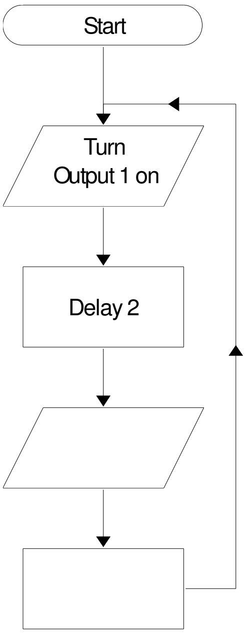
Questions

Where is there a crossing like this in your neighbourhood?

What is special about the yellow lights and why are they there?

What must drivers do when they get near to this type of crossing?

The pictures in Flowol are call 'Mimics' and can be controlled by you. Load the Flowol program and open the 'Zebra Crossing'. To see what the mimic can do, click on one of the lights on the picture. We will call these lights 'Output 1'.



Activity 1

Create the instructions (a program) to control the lights automatically by building up this flowchart.

Click on the symbol you want from the toolbar and click again on the workspace to position it. Use the 'prompt box' at the bottom of the screen to put the instructions in each symbol. Finally use the 'Line' tool to join up the symbols.

Remember to add your own instructions to the blank symbols.

Click on ▶■ /Run to see if the lights work.

Activity 2

Now use to 'Edit' (hand) tool to change the delay times until you find the most effective and energy efficient routine.

Save your program.

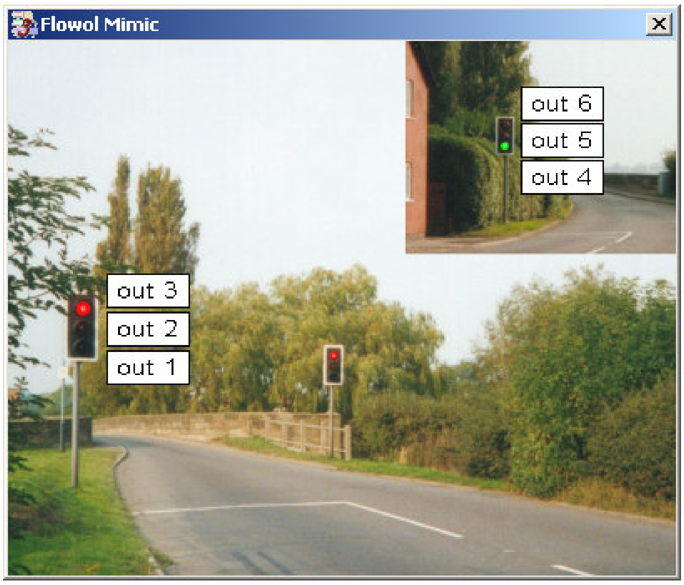
Questions

What improvements would you like to see to help people cross a road?
What other systems already exist?

__**Traffic Lights.**__ This picture shows the traffic lights at both ends of a narrow bridge.

Where are the traffic lights in your neighbourhood?

Open the 'Bridge Lights' mimic and show the labels. See what the mimic can do by clicking on the three main lights.



Activity 3

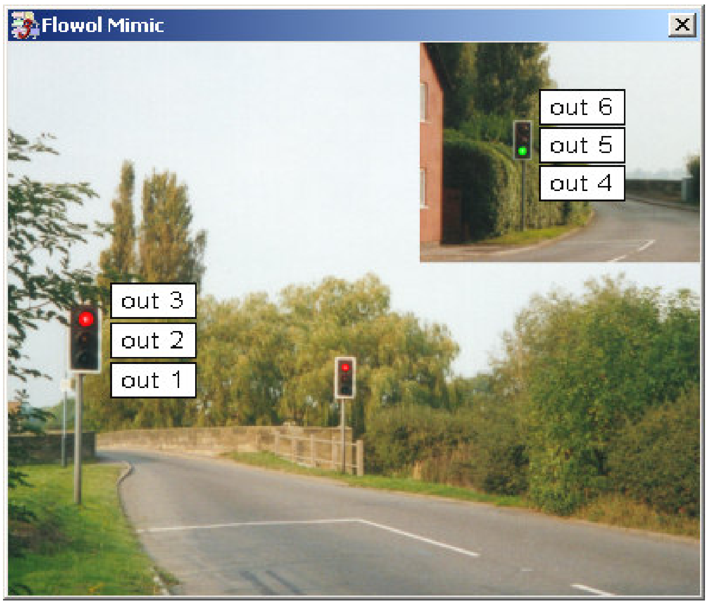
Call the three main lights outputs 1, 2 and 3, and create a program to control the set of lights at one end of the bridge.

Run the program to check your sequence and then make any changes which might improve it. **Save** your new program. Call it '3lights'.

Now use the T (text) tool and add a heading or comment to your flowchart. **Save** your '3lights' program again.

Two sets of Traffic Lights

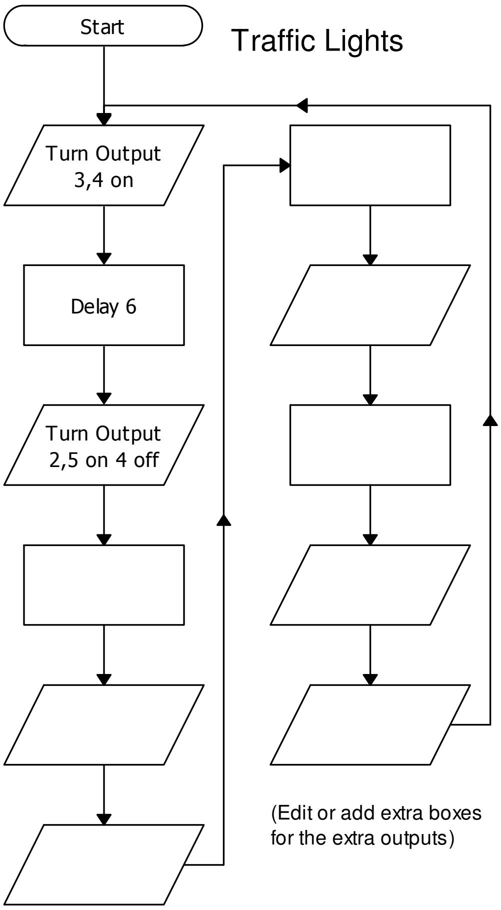
I'm sure you got the sequence for one set of lights correct, but what is the combined sequence when both sets work together?



Open the mimic again but explore the mimic this time by clicking on the outputs on the monitor bar or window at the bottom of the screen.

Activity 4

Now build either another program to control the other set of lights and run both programs together. Or add extra symbols and edit your first program to control both sets in a single flowchart. It might look something like this.



(Edit or add extra boxes for the extra outputs)

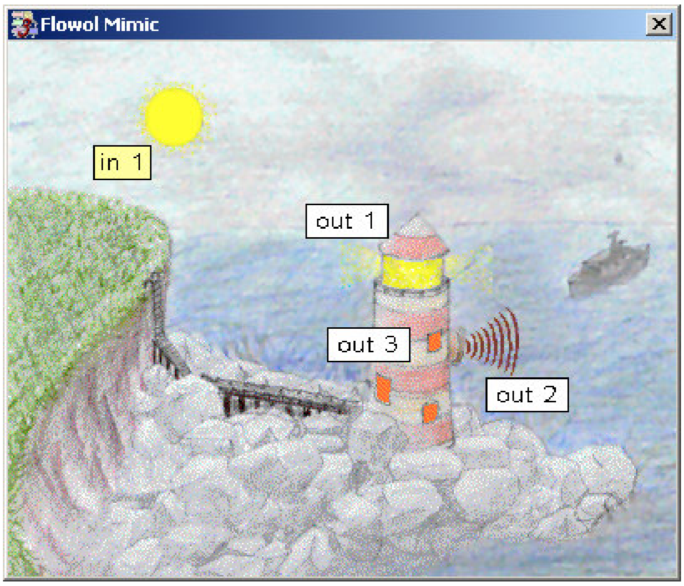
It might also be useful to re-scale/Zoom the flowchart to 80% Now add some text.

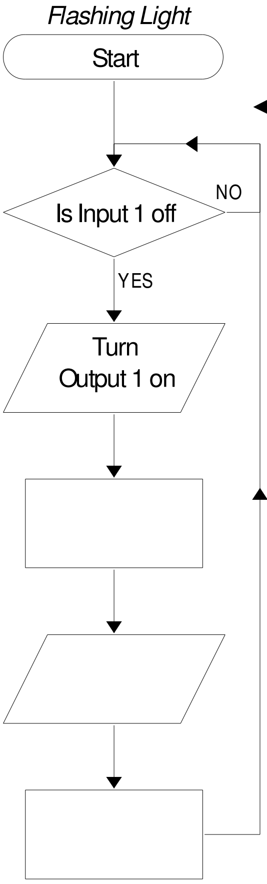
Save your '6lights' program.

So far the systems have been controlled by a set of instructions which are remembered and repeated. In the following examples, the situations may need to respond to an external event such as a button being pressed or the daylight [brightness] changing.

The Lighthouse.

Open the 'Light House' mimic and use the monitor to explore the mimic by clicking on outputs 1, 2 & 3 and input 1. Input 1 is like a light sensor which is on when it is daylight. [Click on the Moon/Sun]





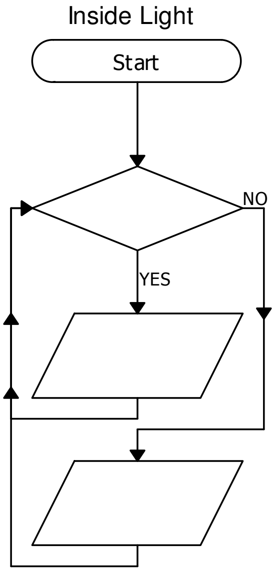
Activity 1

Construct this control flowchart to turn on the flashing beacon of the lighthouse only at night-time. Add some labels to your flowchart.

In this routine you use the 'decision' symbol. You will need to choose the 'Yes' and 'No' lines.

Activity 2 →

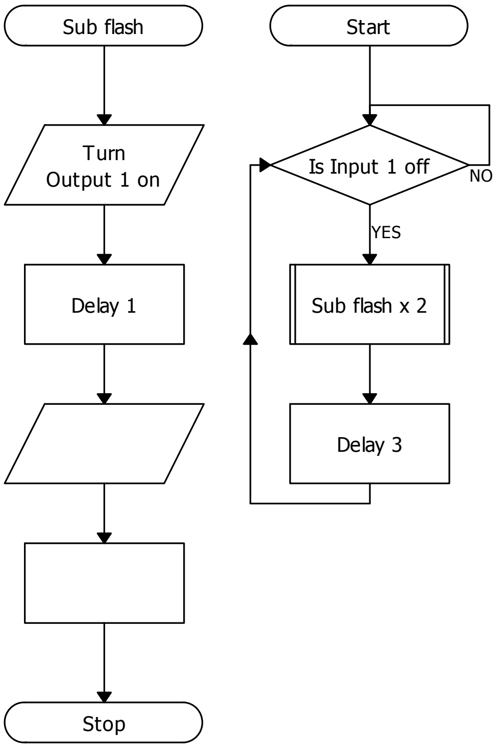
Now create this flowchart to control the inside light. This light should stay on when it is dark and go off automatically in the daytime. [Both flowcharts will run together].



Save your 'Lighthouse' program.

Activity 3

Create a more interesting flashing sequence with a subroutine.



Subroutines: Use the Subroutine symbol and build a flowchart for a single flash 'flash'. Now produce the main routine which has the subroutine 'flash' nested in it. In my example the subroutine runs through twice (x 2).

Now change the main routine to create your own interesting flashing sequence and **save** your program.

Activity 4

Control the foghorn by constructing another flowchart. Use Input 2 on the monitor bar to turn the foghorn on.

Activity 5

Give an "Mmmmm" sound to your foghorn [PC version only].

Sound: If you are using a Windows version of Flowol then you may be able to record some sounds or comments for your program. Connect a microphone into your computer's sound card and open the 'Sound recorder'. Make your recordings and save them.

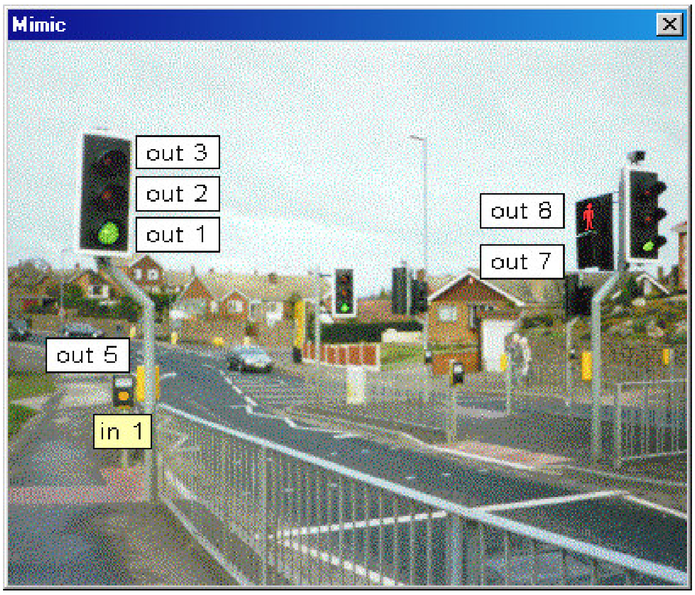
You can now add these sounds to your control program by inserting an extra output symbol at the right place in the flowchart. Select 'Sound' in the prompt box and browse for the appropriate sound file.

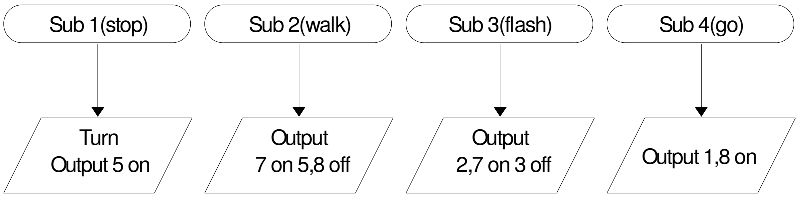
Record your "Mmmmm" foghorn sound and add it to your program.

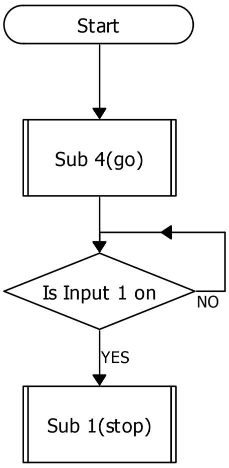
Pelican Crossing

Where is there a Pelican crossing in your neighbourhood?

Open the 'PELICAN' mimic and use the monitor window to see what the mimic can do. Discuss with a friend to see if you both know how the lights and 'people' symbols change when the button is pressed.
Split your solution into separate statements.







Activity 1

Construct and complete these four subroutines:

- To stop the vehicles.
- Indicate when it is safe to walk across.
- Warn the pedestrians to clear the crossing.
- Allow the vehicles to move.

Activity 2

Complete the main routine to call up the subroutines correctly.

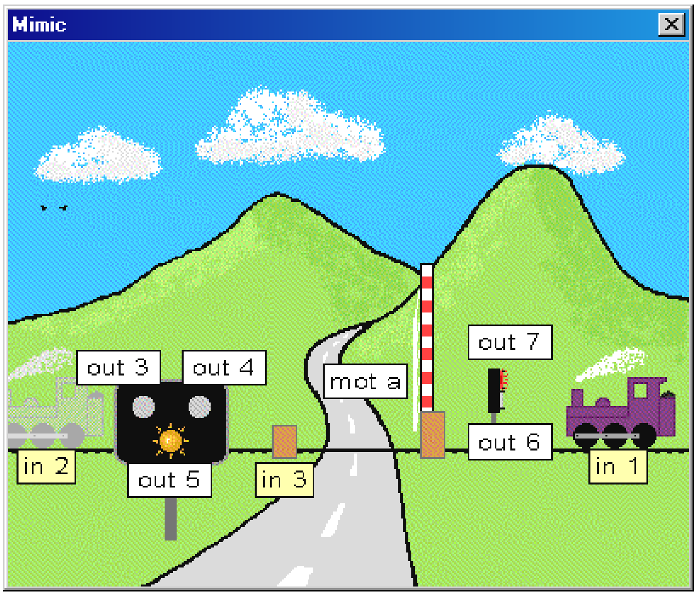
Activity 3

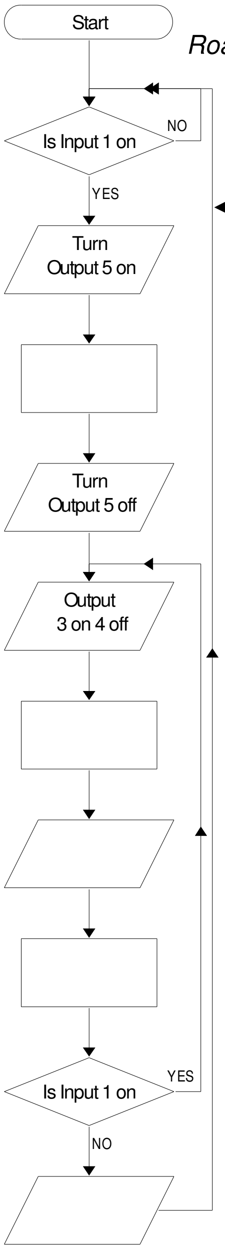
If you can produce sound, record a beep sound and add a 'beep' subroutine to indicate when it is safe to cross.

Refine and **Save** your 'pelican' program.

Level Crossing Barrier

Open the 'Levcross' mimic and use the monitor bar to see what this mimic can do. Note: if the first train is 'on' when you click on the second train, the first one will fade out suggesting that the train has passed by.



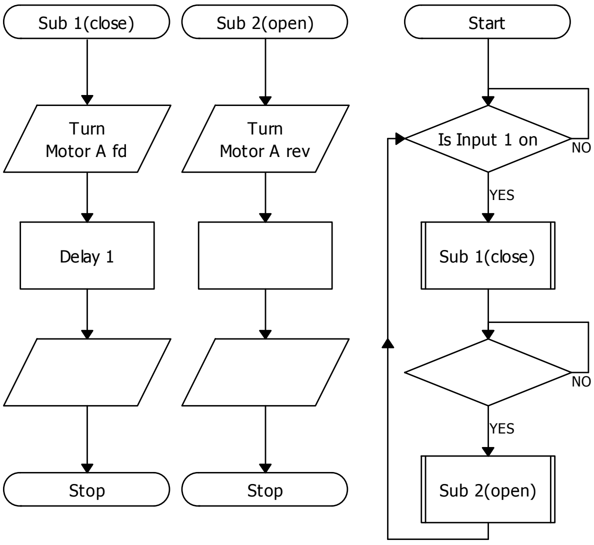

Road Lights

Activity 1

Construct and complete the flowchart to control the road lights. The flashing red lights should only go off when the train has passed by.

Activity 2

A motor moves the gate, so we use forward and reverse. Finish off the routines to control the crossing gate.



Activity 3

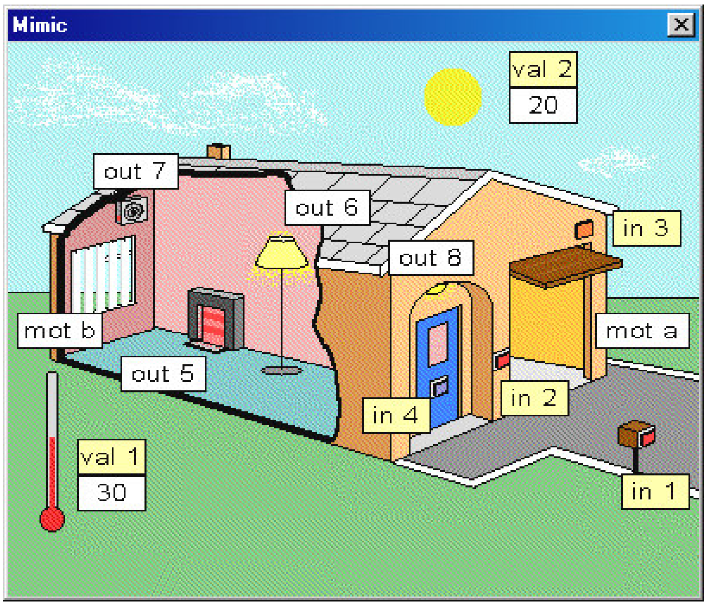
Control the lights

for the train, by adding extra symbols and commands to suitable places in your flowcharts. Refine the program by inserting extra delays. **Save** your 'LevelX' program.

The Auto-home

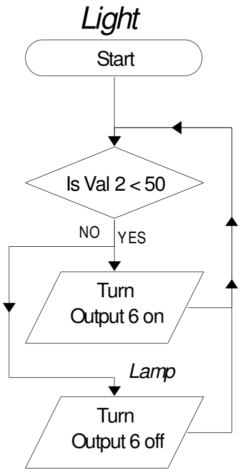
What automatic control features do you have in your home?

Open the 'AUTOHOME' mimic and the monitor window to see what it can do.



Activity 1
Construct a program to open the garage door, when input 1 is pressed, and to close it again when input 2 is pressed. Label your flowcharts and **save** 'autoH'

Activity 2
Assume input 3 above the garage door is a movement/heat sensor which can detect a person on your driveway. Construct a program to turn on the light above the front door, when a person is detected.



Daylight brightness & temperature. If you click on the numbers, near the Sun and thermometer, they will increase or decrease in steps of 5 or 10. This is our way of simulating the changes in the values (val) of daylight brightness and temperature.

Activity 3
Construct this flowchart to turn the inside light on, only when the light value goes below 50 units.

Activity 4 Construct another flowchart to make the electric fire to come on, only when the temperature value goes lower than 30 units.

Continued

Activity 5 (Auto-home continued)

Now control the electric cooling fan above the window. Think about the temperature you choose. Remember to add labels to your routines and **re-save** your program.

Activity 6

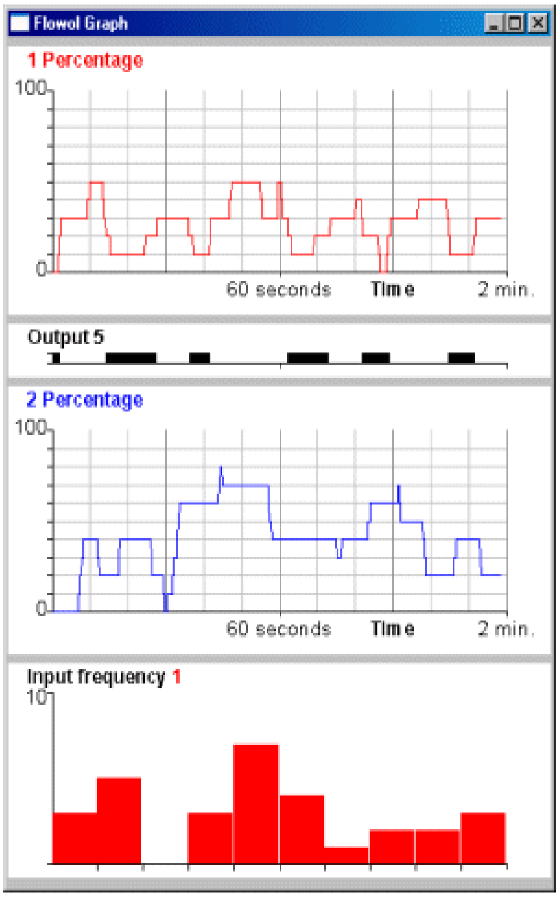
What else can we do? Well the window blinds can be controlled electrically. Give it a go!

Activity 7

Have a look at activity 2 again, the one with the person detector. How can that system be improved? Try it.

Activity 8 Make the push button on the front door do something. Make sure you **re-saved** your 'autoH' program at each stage.

<u>Monitoring and logging</u> data can be shown with the Flowol Graph. Open a graph window 'Show graph'. Select what information you want to observe from the Graph toolbar [Your tutor may have to show you this the first time].



Activity 9

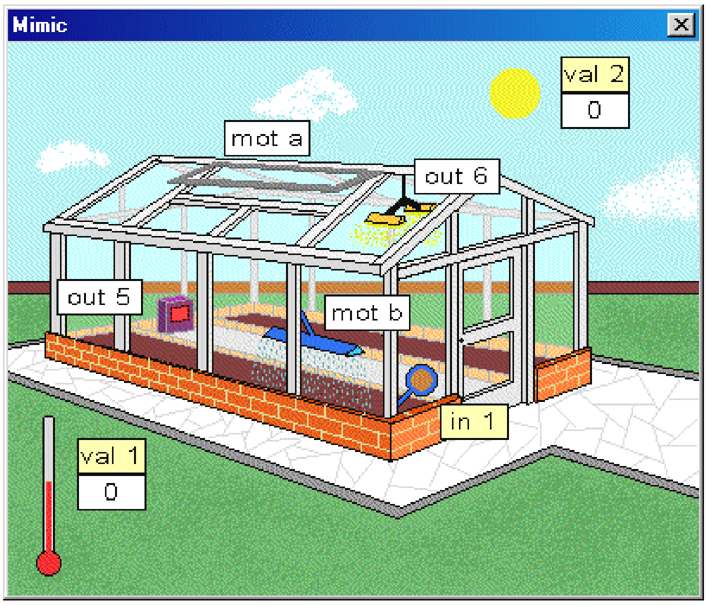
Run your 'autoH' program and keep clicking on the different inputs. My graphs are logging: temperature, the electric fire, brightness level and how often the garage door is opened.

The Intelligent Greenhouse

Why do we have greenhouses? Do some research and determine the best conditions for growing plants.

Open the 'greenh' mimic and use the monitor window to explore the functions of this greenhouse.

Input 1 is a moisture sensor [off when dry]. Temperature and daylight and can also be detected.



You can control the lights, heater, window and water sprinkling system.

You can produce your own warning sounds if you need them.

Assignment
This is an open assignment. Apply your knowledge and skills to make this greenhouse look after the plants for you.
Make sure you label the different flowcharts to show clearly what you are trying to achieve.

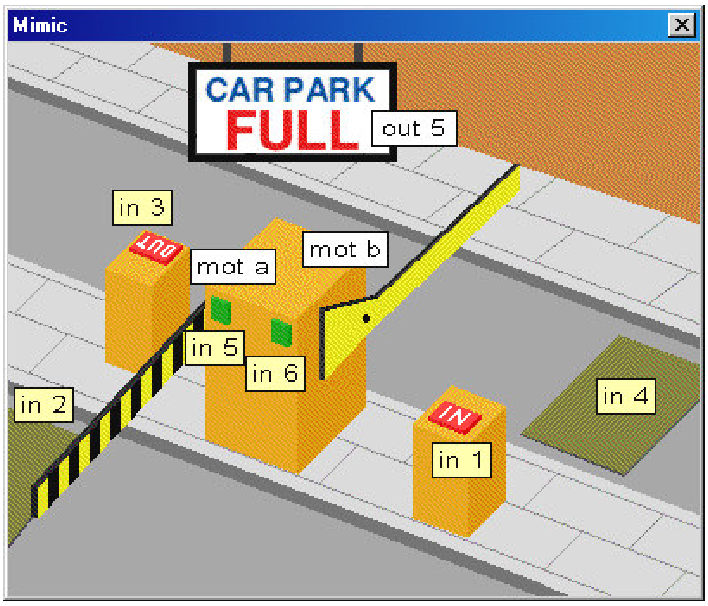
Save your 'greenh' program.

<u>Using a control box</u> can bring assignments alive. You can construct models, monitor the input conditions, use graphs which show actual readings, and use real mechanisms to operate your systems.

Car Park Barriers

Parking restrictions can sometimes be inconvenient but there are some advantages. What advantages can you think of?

Open the 'CARPARK' mimic and use the monitor window to explore the mimic.



Activity 1
Construct a program to open the 'IN' barrier when input 1 is pressed and close it when input 2 is pressed.

Activity 2
Construct a similar program to control the 'OUT' barrier.

Activity 3 [Advance/Optional]. Ask you tutor if you should try this.

When exploring the 'CARPARK' mimics you may have noticed the two green buttons on the side of the barrier post. These are 'feedback' or movement limiting switches. Input 5 should go on when the IN barrier reaches its highest or lowest position, and input 6 should go on when the OUT barrier reaches its highest or lowest position. Note: these switches will turn off automatically when the barriers move again.

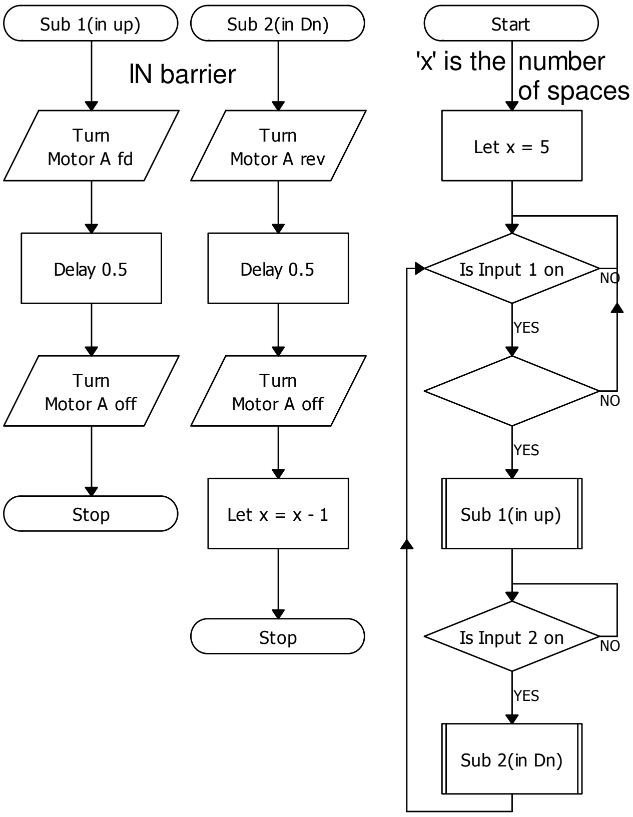
Now modify your routines to stop the barrier motors when the barriers have reached their highest and lowest positions.

[**Note:** The 'Level Crossing Barrier' can also be modified in a similar way. Check the functions of inputs 3 and 4 on the 'Levcross' mimic.]

Applying Variables (Car park continued)

<u>A variable</u> e.g. x or n, can be given a number or can be changed mathematically in a process symbol. The variable can also be used in a decision symbol.

This flowchart shows where I have modified my 'IN' program to stop cars entering if a space is not available. ['x' is the number of spaces in the car park].



Activity 4

Modify your 'IN' and 'OUT' routines to stop the IN barrier working if there are no spaces available. The barrier should operate again when spaces become available.

Activity 5

Now refine your program or draw a separate flowchart to control the large 'Full' sign.

Extra Activities

Other mimic packs are available to reinforce or extend your control work with Flowol. Check http://www.flowol.com for detail.