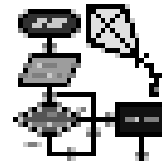


# Primary School Activities for *Flowol*<sup>TM</sup>



By: Rod & Anthony Bowker  
Copyright © Keep I.T. Easy 2003

## System Requirements for the Primary 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:

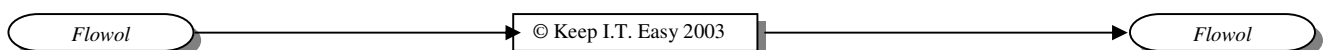
1. Flowol 2 or later needs to be installed before you can install the Primary Mimics.
2. You need to be logged on as an Administrator to install these mimics.
3. This CD contains an msi installer. This can be assigned to workstations as a package. (There are no shortcuts with the mimics).
4. If the Flowol 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 part of the Flowol tutorial, and the extra mimics on the Primary 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 in a variety of ways: to manipulate a robot, a pelican crossing, a railway level crossing and finally the automatic systems which might be possible in the home.

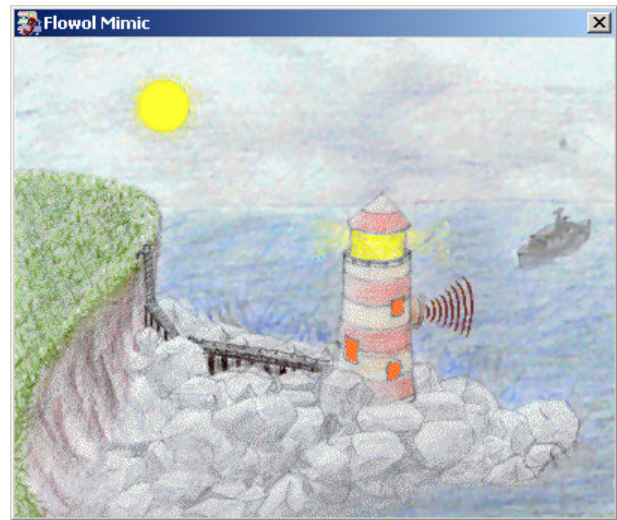
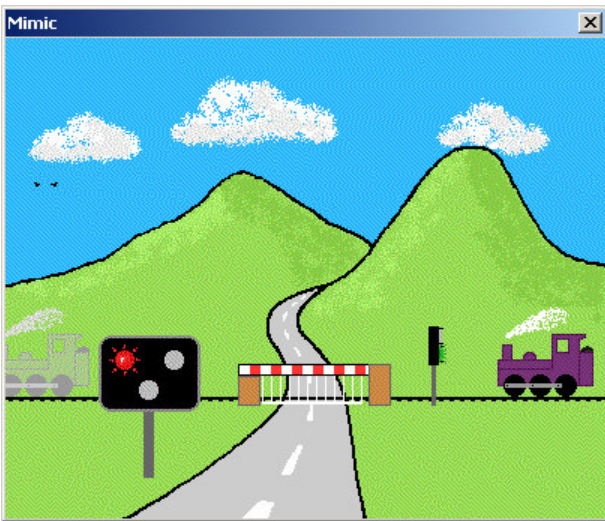
Other mimic packs are available, check <http://www.flowol.com> for details.



## Let Flowol™ help you take control.

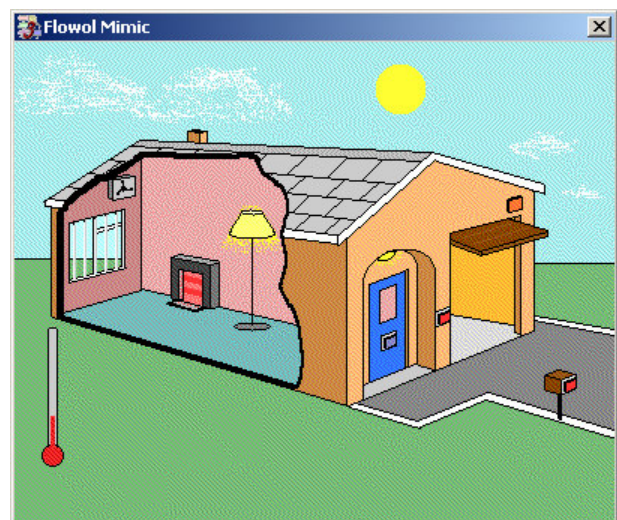
Automatic systems are all around us; keeping us safe, making life comfortable and helping us with difficult and unpleasant tasks.

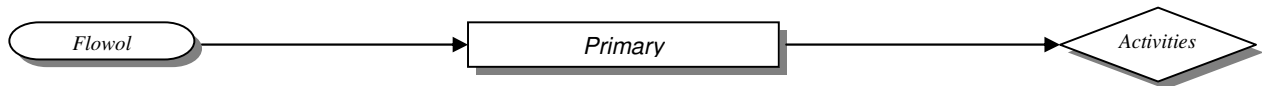
Flowol will allow you to produce your own solutions to many of these situations. We will start by guiding you through the simple tasks to control traffic signals and warning lights.



With your skills, you will soon be able to solve other 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 can help?





**Zebra crossing.** This picture shows a fairly quiet road.

**Questions**

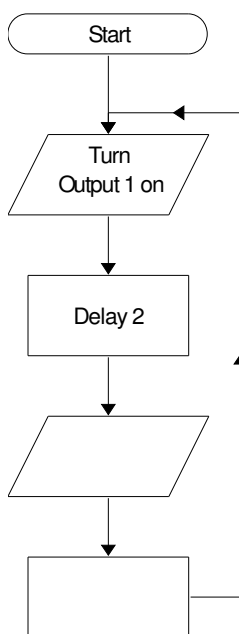
Where is there a crossing like this near to your school?

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' mimic. 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 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 the 'Edit' (hand) tool to change the delay times until you get the best flashing routine.

Now **Save** your program.

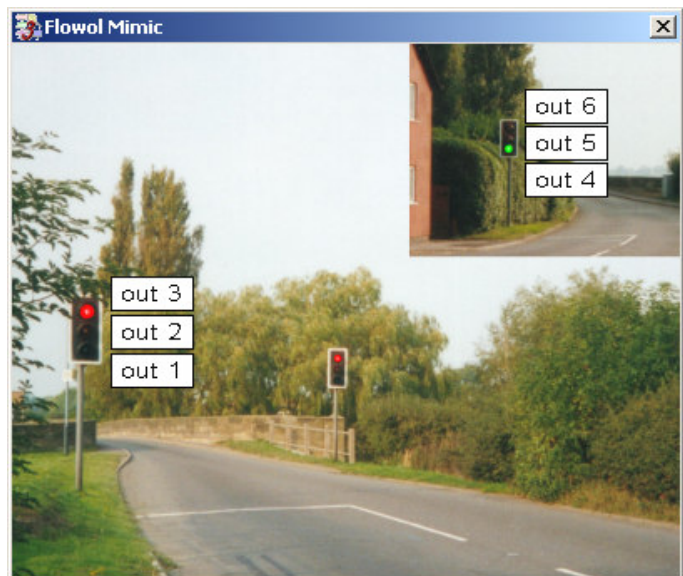
## Questions

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

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


Where are there traffic lights near to your school?

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

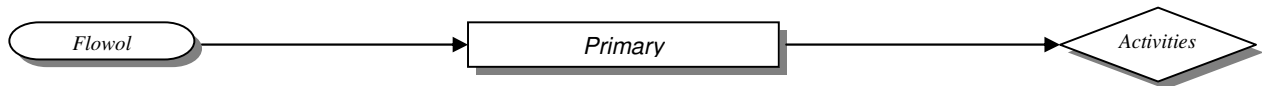


## Activity 3

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.

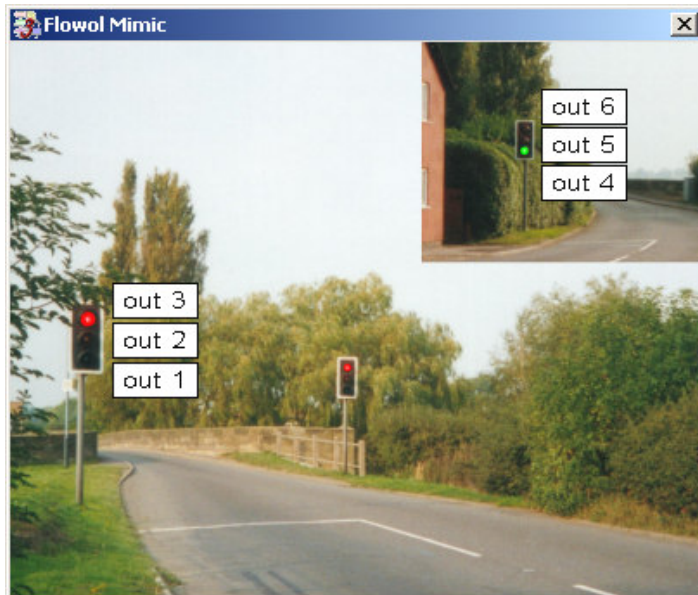
Click on  'Run' to check your program, and then make any changes which might improve it. **Save** your new program. Call it '3lights'.

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 right, but what is the combined sequence when both sets work together?



Open the 'Bridge Lights' mimic again. Explore the mimic this time by clicking on the outputs on the monitor bar/window at the bottom of your screen.

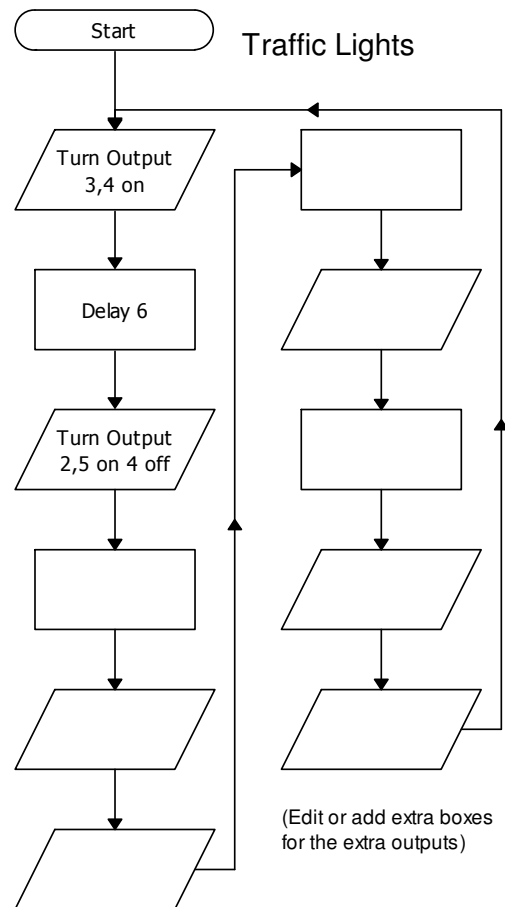
### Activity 4

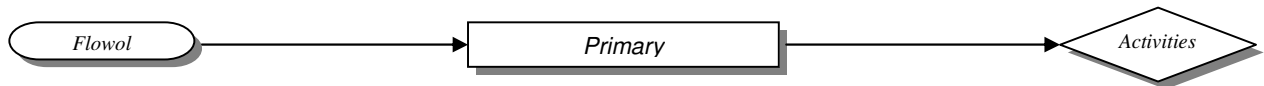
Now build another program to control the other set of lights and run them both together.

Or you could add extra symbols and edit your first program to control both sets in a single flowchart. It might look something like this.

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

**Save** your '6lights' program.

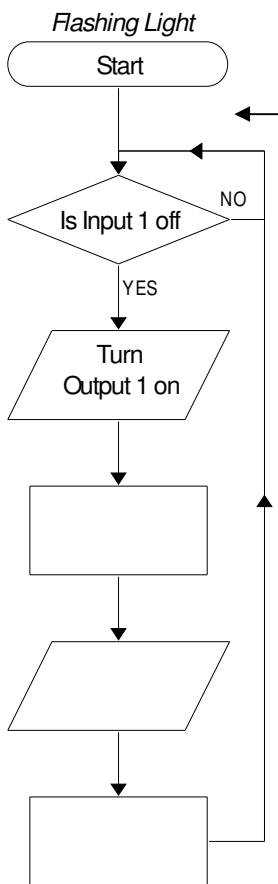
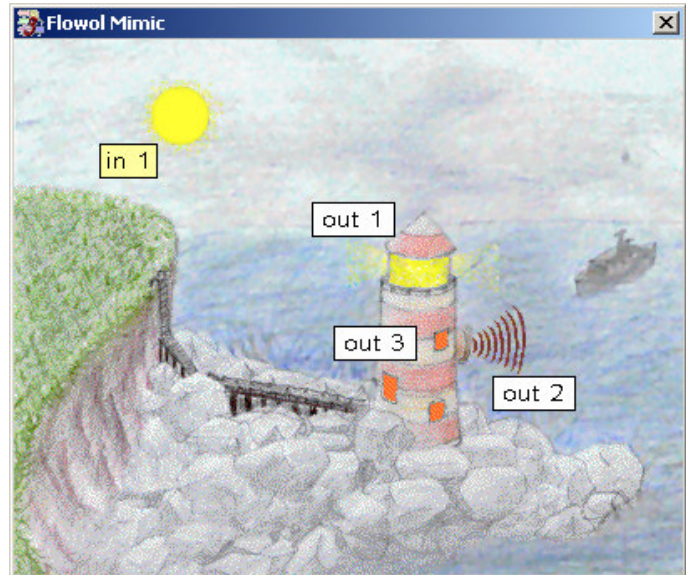




So far the systems have been controlled by a set of instructions which are remembered and repeated. In the next examples, the situations may need to respond to an external event such as a button being pressed or the daylight [brightness] changing.

### The Lighthouse.

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



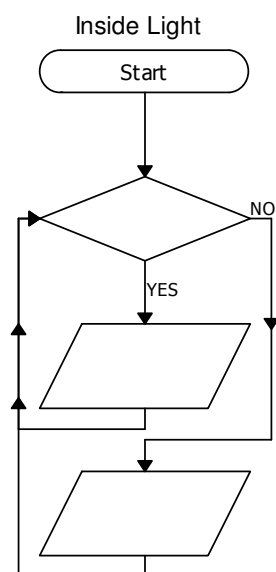
#### Activity 1

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

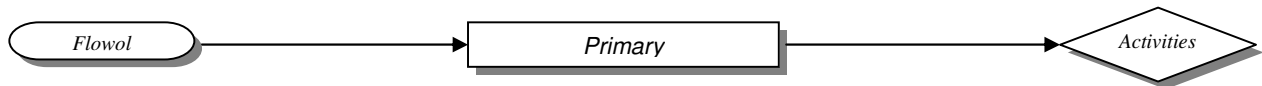
In this routine you need to use the 'decision' symbol and the 'Yes' and 'No' lines.

#### Activity 2

Now create another 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.

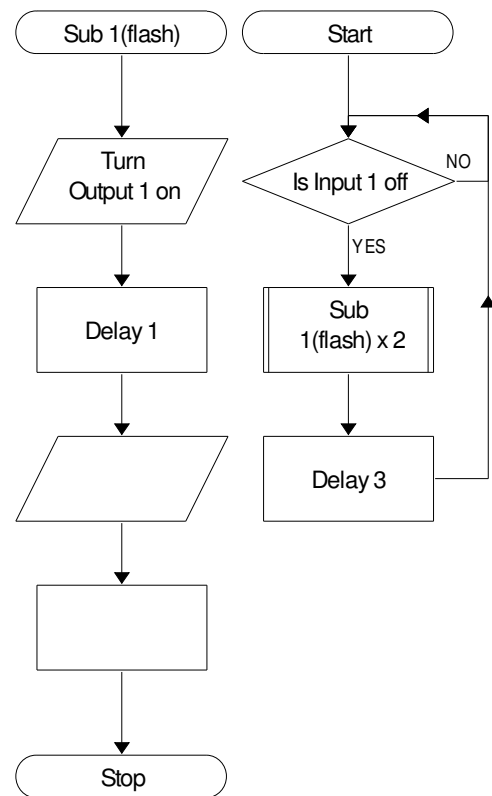


### Activity 3

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 'flash' subroutine 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

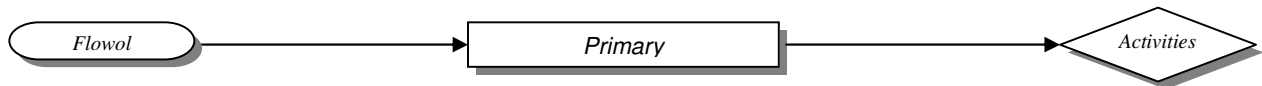
Control the foghorn by constructing another flowchart. Use Input 2 to turn the foghorn on.

### Activity 5

Give an "Mmmm" 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.

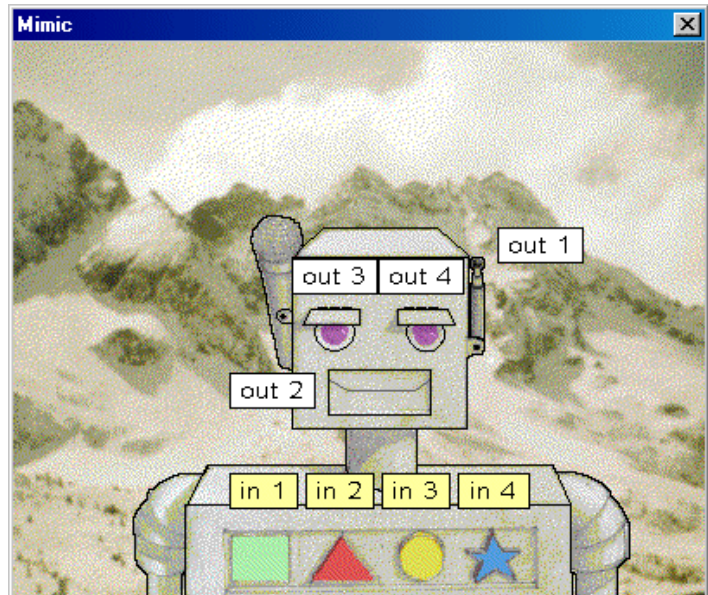
Record your "Mmmm" foghorn sound and add it to your program.



## Testing your skills and having a little fun

This Robot mimic has four inputs and four functions which can be controlled.

Open the 'Robot' mimic and use the monitor window to explore the mimic by clicking on the inputs and outputs on the monitor.



### Activity 1

Amuse your friends by constructing three or four separate flowcharts to control different movements of the robot.

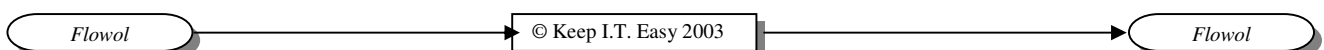
**Save** your 'Robot' program.

**Activity 2:** Make your robot talk [PC version only and if your tutor gives you permission].

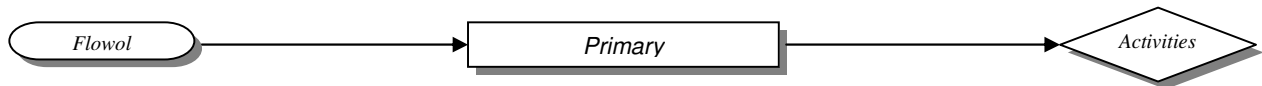
Use the 'Sound Recorder' and record some [sensible] statements or 'jingles' which your robot can say or sing.

Construct a program to control both the sounds and movements of your robot. Remember a robot cannot say two things at the same time.

**Save** this 'Robot2' program.







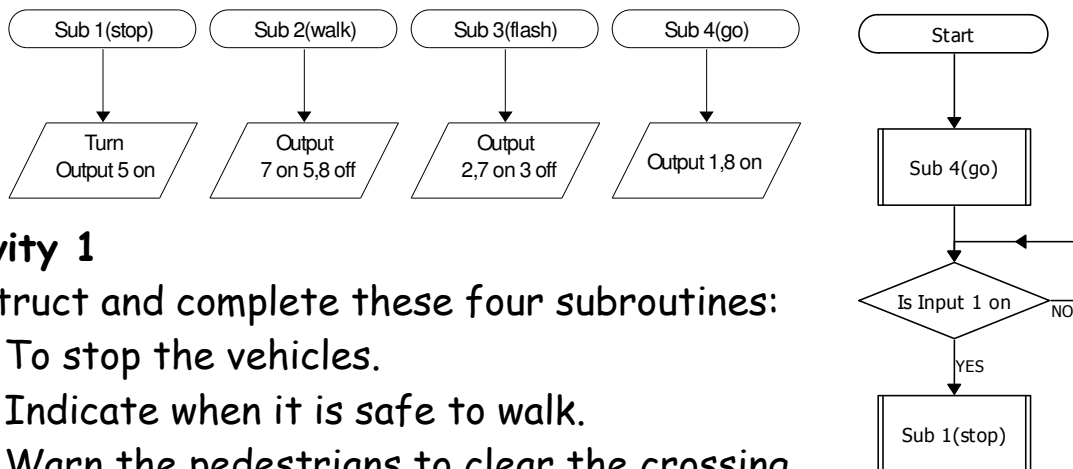
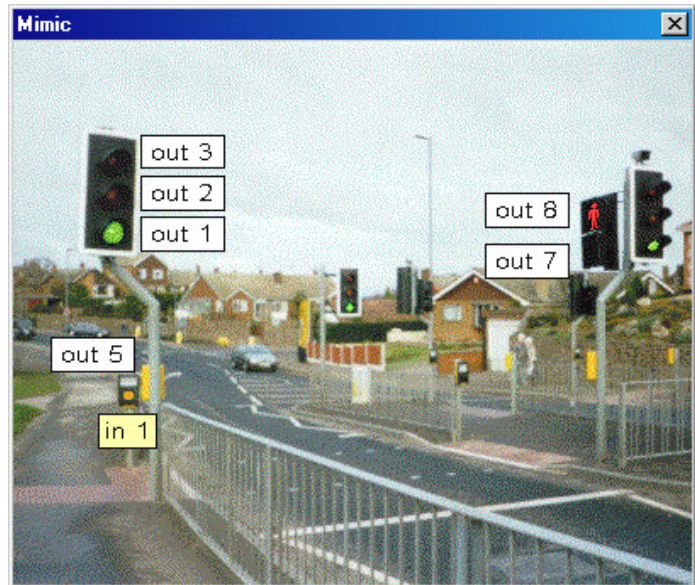
## Pelican Crossing

Where is there a crossing like this near to your school?

Open the 'PELICAN' mimic and use the monitor screen 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.
- Warn the pedestrians to clear the crossing.
- Allow the vehicles to move.

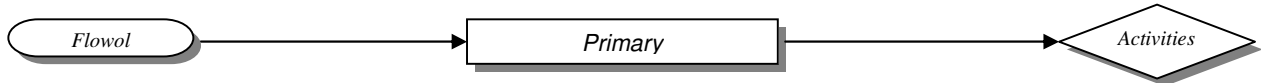
### Activity 2

Now complete the main routine to call up the subroutines correctly.

### Activity 3

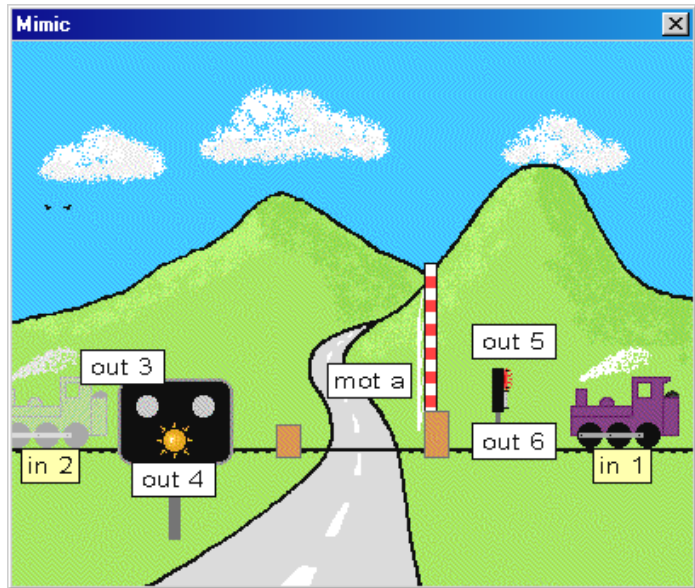
If you can produce sound, record a beep sound and add a 'beep' subroutine to indicate when it is safe to cross.

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



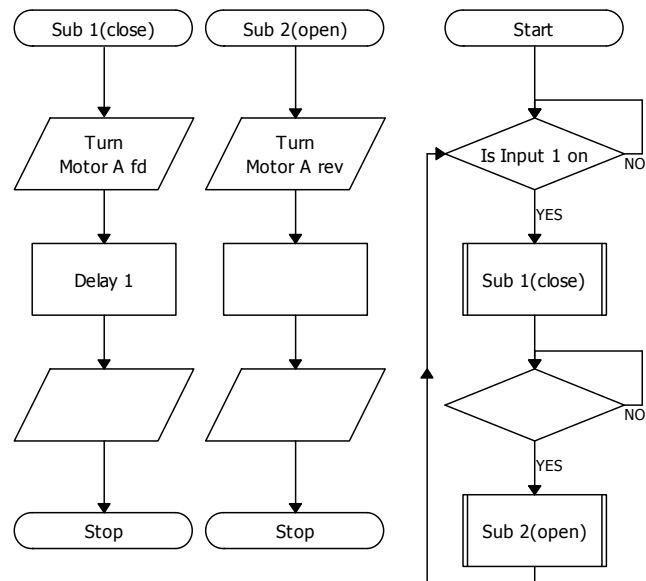
## Level Crossing

Open the 'LevelP' mimic and use the monitor window 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 through the crossing.



### Activity 1

A motor closes and opens the gate, so we need to use the forward and reverse commands. Copy these routines and finish them off to control the crossing gate.



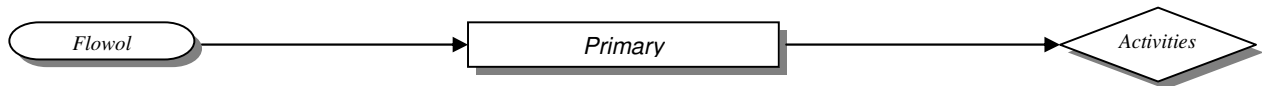
### Activity 2

The barrier should only close after the road traffic has been stopped. Add more symbols or another subroutine to control the road lights.

### Activity 3

Now operate the signal lights for the train, by adding extra symbols and commands to the right places in your flowcharts.

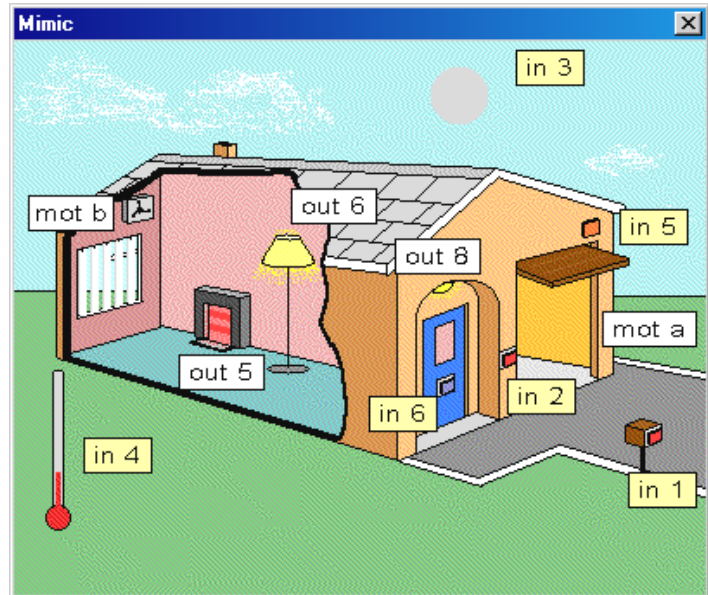
**Save** your 'LevelX' program.



## The Auto-home

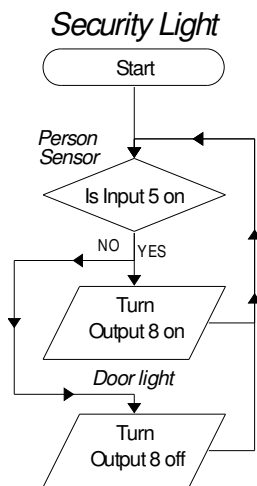
What automatic control features do you have in your home?

Open the 'AutoHP' mimic. Use the monitor window to see what this auto-home 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'



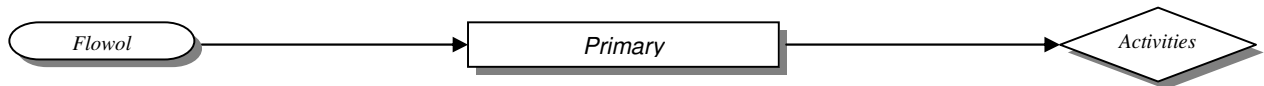
### ← Activity 2

Assume input 5 above the garage door is a movement/heat sensor which can detect a person on your driveway. Construct this program to turn the light above the front door on when a person is detected.

### Activity 3

Now construct a flowchart, similar to the one shown, to turn the inside light 'on' when it is dark, but 'off' again when it is daytime.

**Activity 4** Now control the electric fire to come on, only when it is cold. *Continued*



**Activity 5** (Auto-home continued)

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

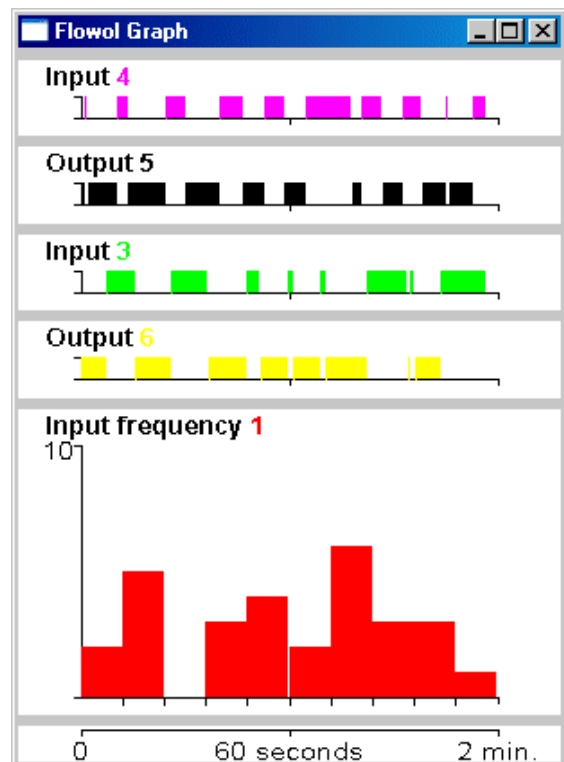
**Activity 6**

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

**Activity 7**

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

Monitoring and logging 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 8**

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

**Extra Activities**  
Other mimic packs are available to reinforce or extend your control work with Flowol. Check <http://www.flowol.com> for details.