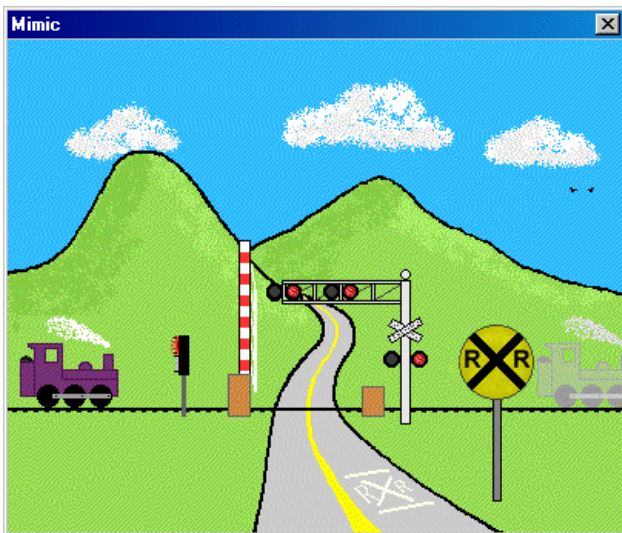


## Let Flowol™ help you take control

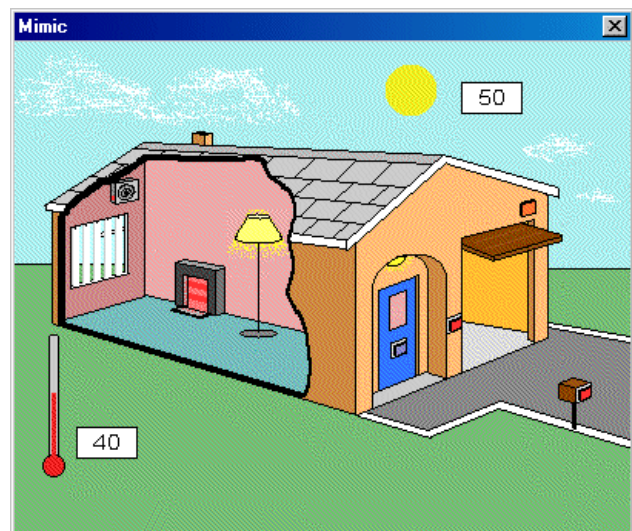
Automatic systems are all around us, keeping us safe, making life comfortable and helping us with difficult and unpleasant tasks.

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 more complex examples such as an automatic railroad crossing or the control systems needed to help people in their homes.

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



**All Stop** This picture shows a fairly quiet intersection.

**Questions**

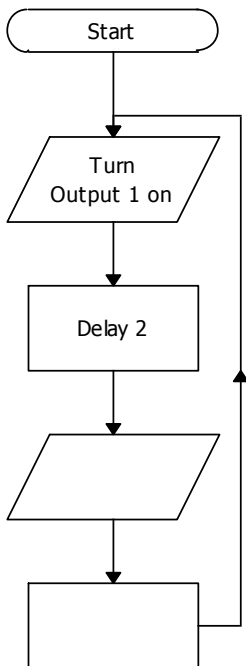
Where is there a street sign like this in your neighborhood?

What is special about the light hanging over the intersection and why is it there?

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



The pictures in Flowol are call 'Mimics' and can be controlled by you. Load the Flowol program and open the 'All Stop' mimic from the 'Open a mimic' icon. To see what the mimic can do, click on the light on the picture. We will call this light 'Output 1'.



**Activity 1**

Create the instructions (a program) to control the light 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 'Start/Stop' to see if the light works.

## 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 and call it 'allstop.flo'.

## Questions

What improvements would you like to see made at road intersections?  
What other systems already exist?

The next mimic shows a **Crosswalk** which uses two lights.

Where is there a crossing like this in your neighborhood?

## Activity 3

Open the 'Crosswalk' mimic and also click on the label icon. See what the mimic can do by clicking on the two lights. These lights are called Outputs 1 & 2.

Now create a program to control these two lights.



Click on 'Start/Stop' to check your program and then make any changes which might improve it. **Save** this 'Xwalk.flo' program.

**Question:** How can crossing a road be made even safer?

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

## Traffic Lights

Where are the traffic signals in your neighborhood? They may hang at the middle of the intersection like these, or from supports which reach over the streets.

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



Open the 'Intersection Lights' mimic and display its labels. Explore the mimic this time by clicking on the outputs on the monitor bar at the bottom of the screen.

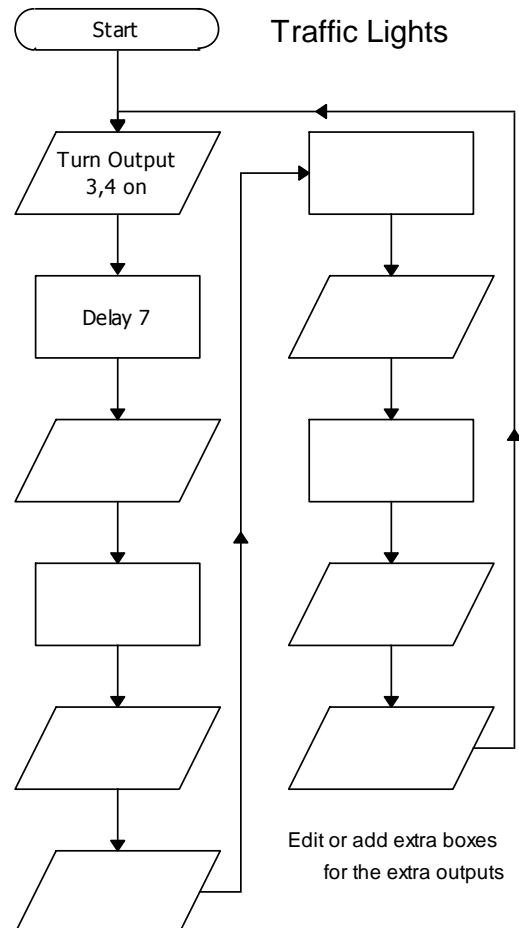
### Activity 4

Now build a new program to control all six lights together. The flowchart might look something like this.

It might be useful to 'Zoom' the flowchart to 80% so it can be seen more easily. Add some text e.g. a heading to your flowchart.

**Save** your 'Xlights.flo' program.

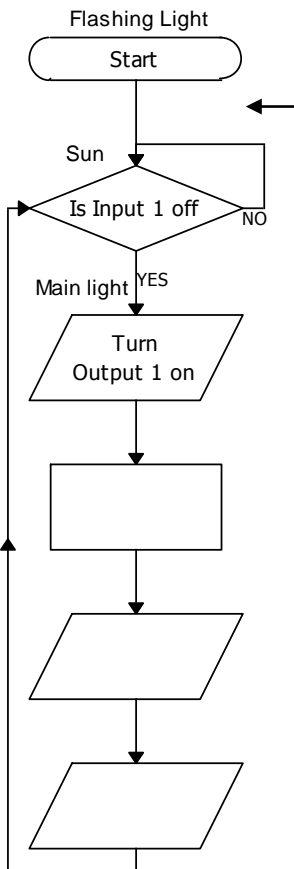
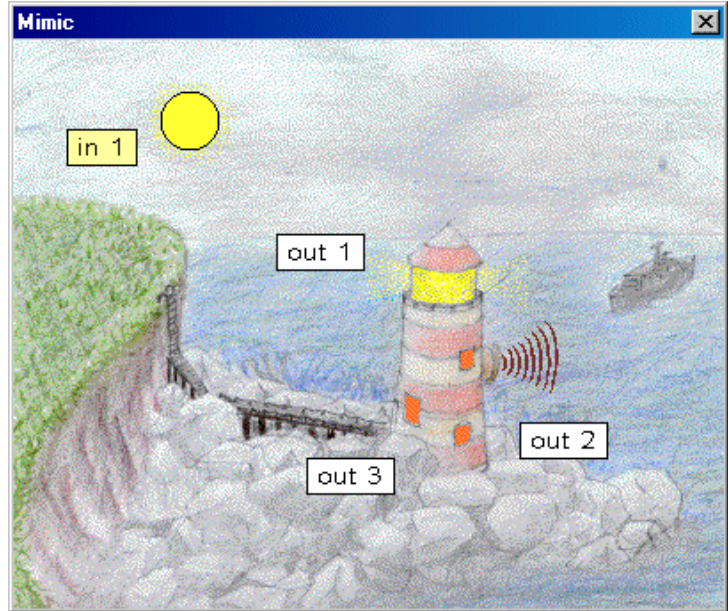
What modifications did you make?



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 and show the labels. Explore the mimic by clicking on outputs 1, 2 & 3 and input 1. Input 1 is like a light probe which is on when it is daylight. [Click on the Moon/Sun]



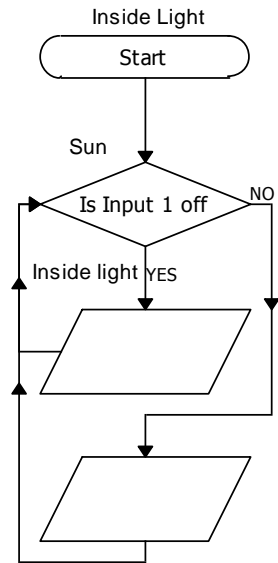
#### Activity 1

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

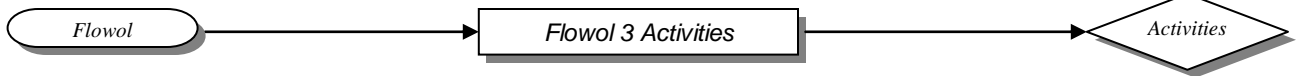
In this routine you need to use the 'decision' symbol with a 'Yes' and a 'No' line.

#### 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.flo' program.

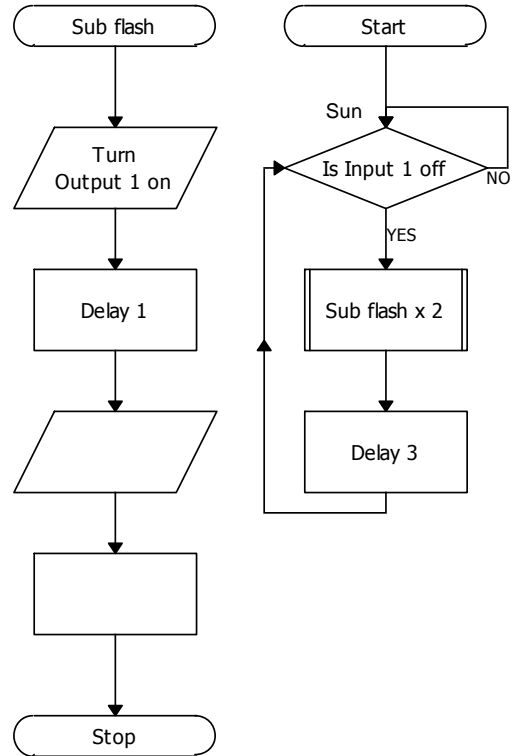


### Activity 3

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 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.

### Activity 5

Give an "Mmmm" sound to your foghorn.

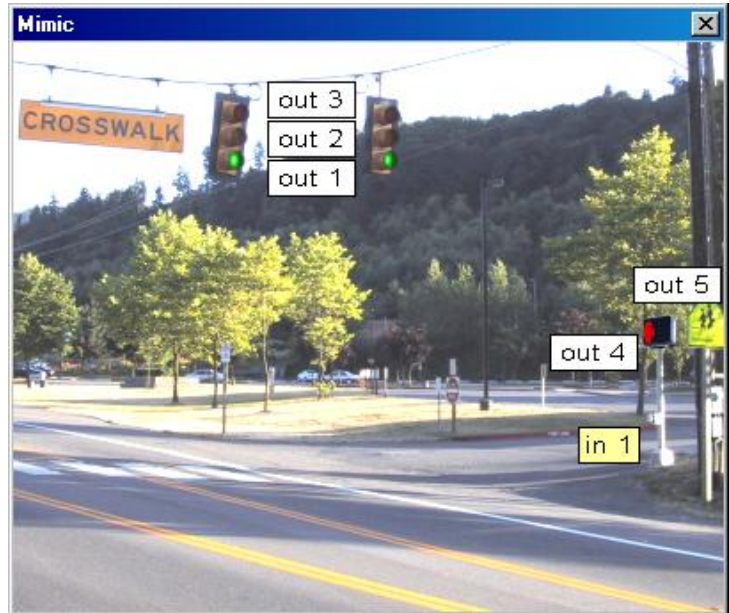
**Sound:** If you are using a Windows version of Flowol then you can record some sounds or comments for your program. Connect a microphone into your computer's sound card and open the 'Sound recorder' using the icon at the top of the screen. Make your recordings and save them. You can now add these sounds to your control program by inserting an extra output symbol at the right place in the flowchart, and browsing for your sound file.

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

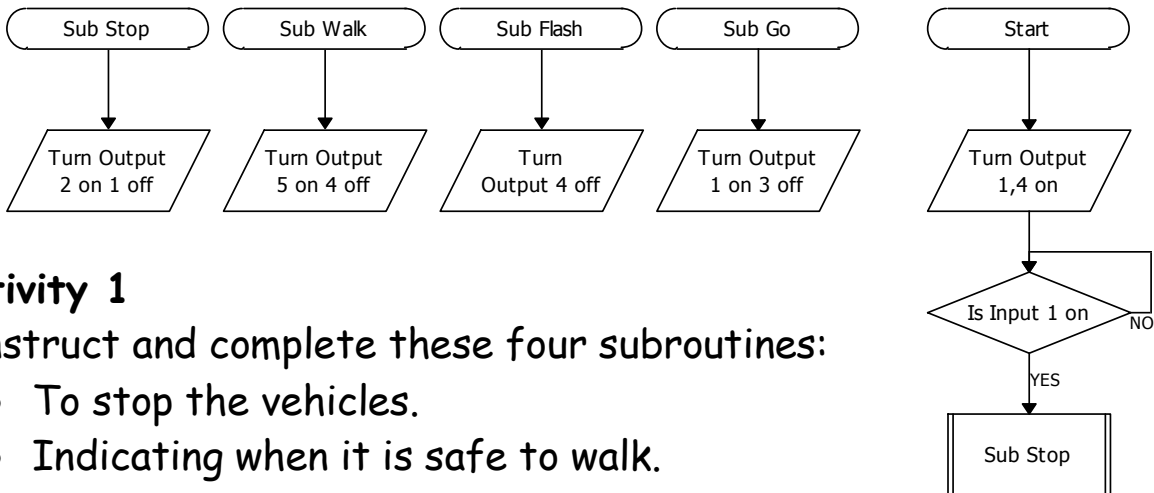
# Crosswalk with lights

Where is there a crossing like this in your neighborhood?

Open the 'Crosswalk 2' mimic and use the monitor bars to see what the mimic can do.



Discuss with a friend to see if you both know how the lights and symbols change when the button is pressed. Split your solution into four separate statements.



## Activity 1

Construct and complete these four subroutines:

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

## Activity 2

Complete the main routine to call up the subroutines correctly.

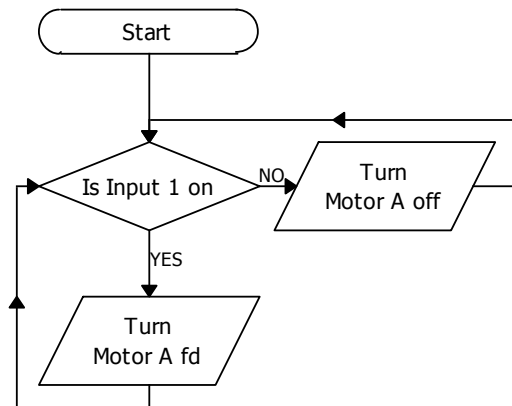
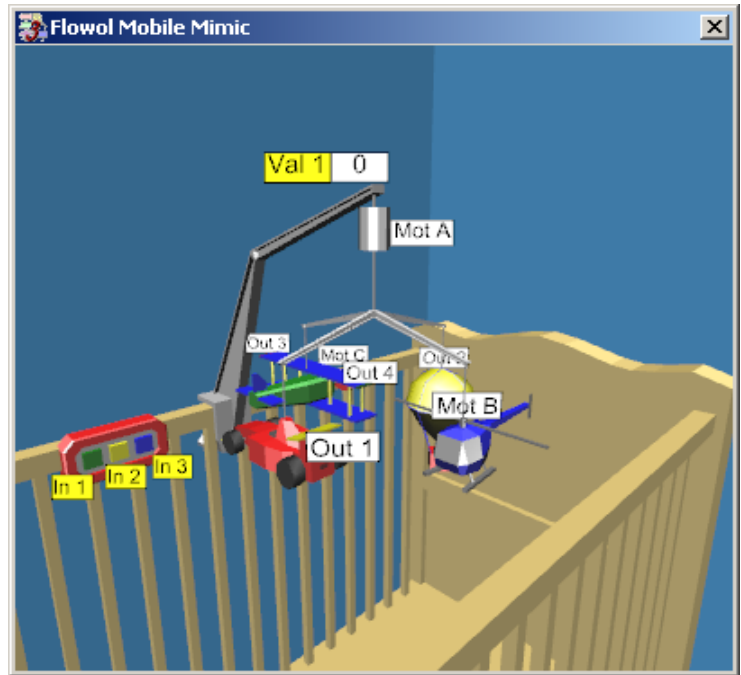
Refine and **Save** your 'crosswalk.flo' program.

## Automatic Crib Mobile

A crib mobile can comfort and entertain a baby but how could an automatic crib mobile be an advantage?

Open the 'Mobile' mimic, add its labels and explore what the mimic can do by clicking on the outputs and motors on the monitor bars.

This mobile has a host of activities ranging from the gentle effect of the lights to the movement of the rotors, propeller and the motion of the whole mobile



**Activity 1:** Controlling the motor to rotate the whole mobile is the most impressive effect. Build this program to use Input 1 to start and stop the main Motor A

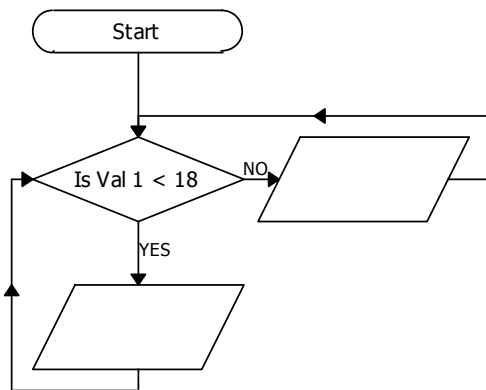
**Activity 2:** Motors of course can go forwards and reverse. Construct another two flowcharts to give some forward and reverse movement to the rotors and propeller when Inputs 2 and 3 are used. Running this whole program should give some interesting combinations when the three switches are changed.



**Activity 3:** Another important control feature for a motor is to change its speed [or Set motor power]. Modify your first flowchart to reduce the main rotation speed of the mobile for a while and then speed it up again. [Remember, if you reduce the motor power percentage, then it must be returned back to 100% for full power].

So far we have only used digital inputs which can only be either on or off. Another type of input can be from an analogue sensor which detects a range of input values [e.g. analogue values could be from different levels of light brightness, different temperatures or different volumes of sound].

On our mimic we have an analogue sensor on the end of the mobile arm which changes in value when you click on the number with the left and right mouse buttons.



**Activity 4:** If our analogue sensor can detect changes in light level, construct an automatic light program to turn on the light in the hot air balloon if the light level goes below a value of 18 units [e.g. Val 1 < 18]

**Activity 5:** The baby should still be occupied before it gets quite this dark, so produce another one or two flowcharts to make the rear light on the car and aeroplane wing tip lights flash if the light level value goes below 60 units.

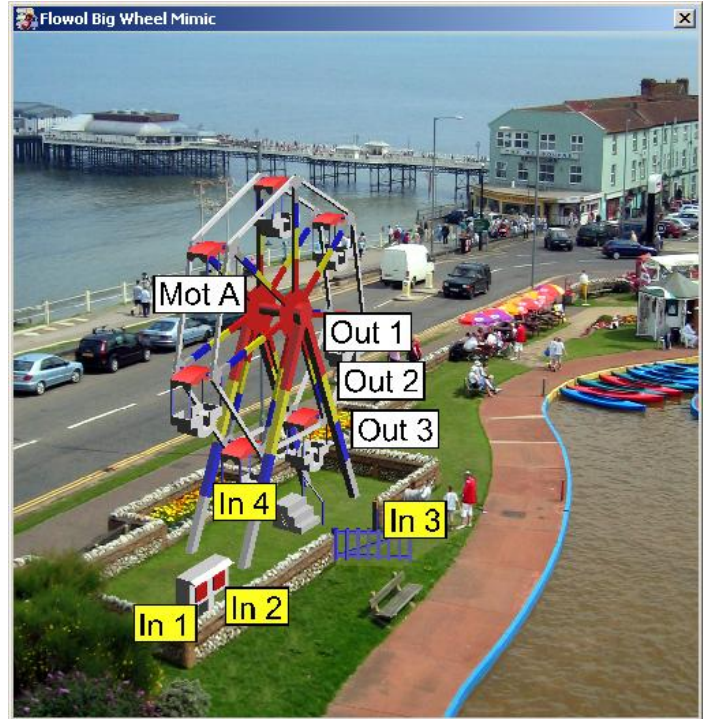
**Activity 6:** The cot mobile would be most interactive if it could respond to the baby's sound. E.g. if the baby makes a quiet noise, some of the lights could turn on or flash for a while. If the noise is louder the rotors and propeller could start to move and if the baby is very noisy the whole mobile could become very active. [Assume the analogue sensor can detect sound for this activity].

# Controlling the Big Wheel

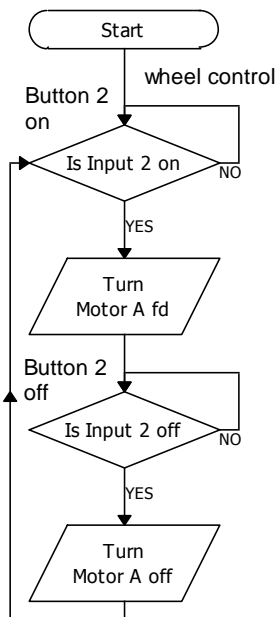
If you were the operator of a funfair ride, how would you produce a pleasurable but safe experience?

Open the 'Big Wheel' mimic and explore what the mimic can do by clicking on the inputs, outputs and motors on the monitor bars.

Input buttons 1 and 2 are normal inputs but when the wheel rotates you may notice that Input 4 flashes each time a seat passes over the steps. Also if you click on the gate with the left mouse button you will find that Input 3 comes on when the gate is shut.



These inputs [3 and 4] are 'virtual inputs'. They are not changed by you clicking on the mimic but are changed by features within the mimic itself.



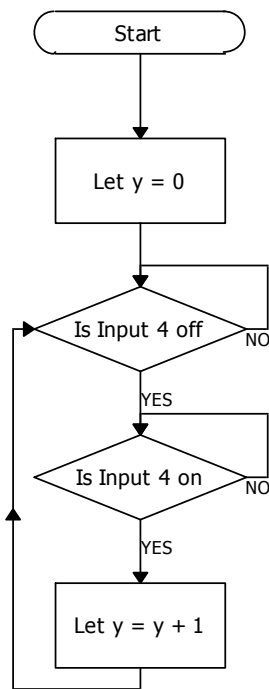
**Activity 1:** To attract the crowd, use the input 1 [button] to control the lighting effects on the wheel. This might be a simple on/off routine but flashing sequences are more exciting [use subroutines].

**Activity 2:** Use the input 2 to control the simple Go/Stop movement of the wheel. [You could perhaps make the wheel speed up and slow down in stages by changing the motor power. Using subroutines would be useful for this].

**Activity 3:** Modify activity 2 to include the safety feature of the gate: i.e. the gate must also be closed before the wheel will start [e.g. if both Input 2 AND Input 3 are on] and the wheel should stop if either input 2 is turned off, or the gate is opened [e.g. if either Input 2 OR input 3 are off].

**Activity 4: Using Speech** [if you have, Windows XP, a sound card and speakers]

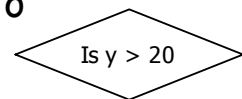
To increase safety even further use the 'Speak' command to give the passengers an automatic verbal instruction to "Hold tight please" just before the ride begins to move.



**Activity 5:** Stopping the wheel automatically. Construct this counting program to increase the variable 'y' each time a seat passes the steps i.e. each time the 'virtual' input 4 goes off and on.

Since there are seven seats, each rotation of the wheel should increase the variable 'y' by 7.

Now modify your program, by introducing a decision symbol, to stop the wheel automatically after it has rotated 3 times.



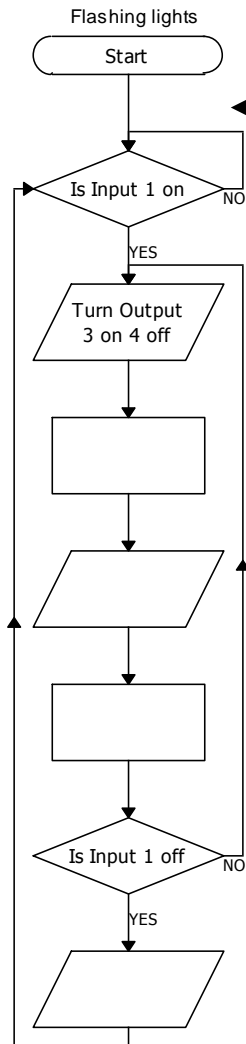
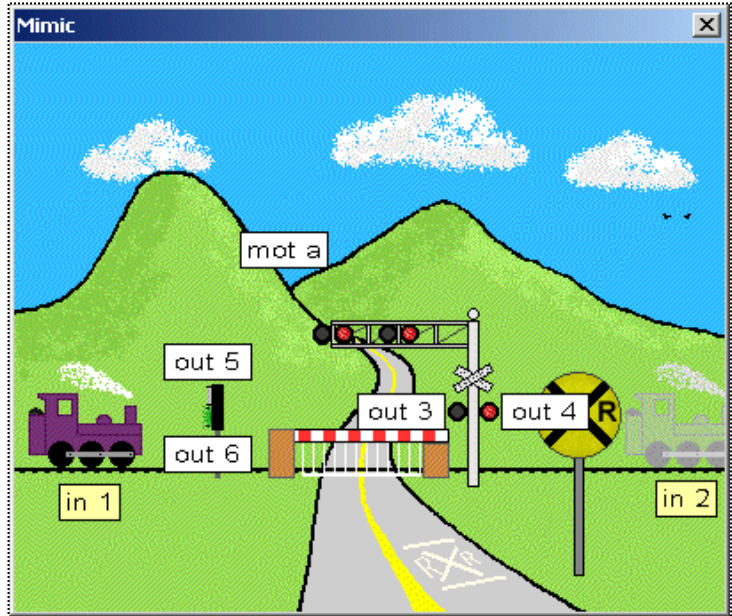
**Activity 6: [Extension]**

Now you have learnt how to use Input 4, modify the program so that each of the seven seats stops automatically at the bottom for a short time for the people to get off. [Note a variable is not needed for this solution]

# Railroad Crossing

Open the 'Railroad Crossing' mimic and use the monitor bars to see what the 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.



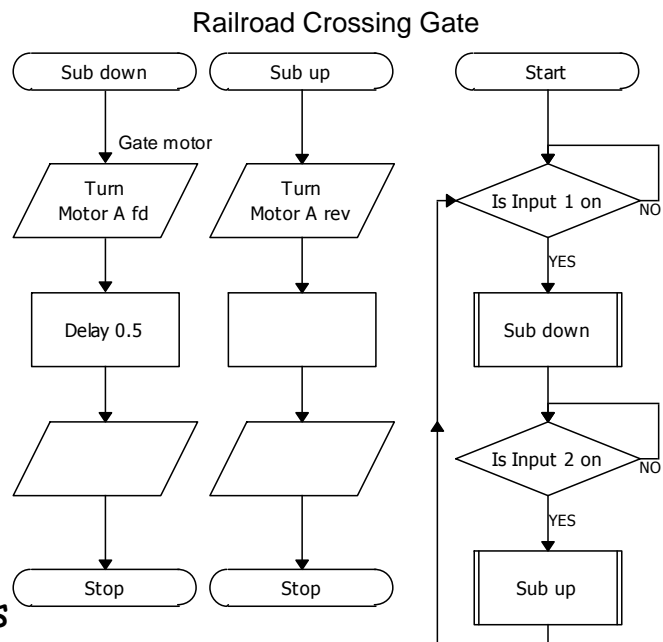
### Activity 1

Construct and complete the flowchart to flash the gantry lights when the train approaches, and turn them all 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 the right places in your flowcharts. **Save** your 'RailX.flo' program.

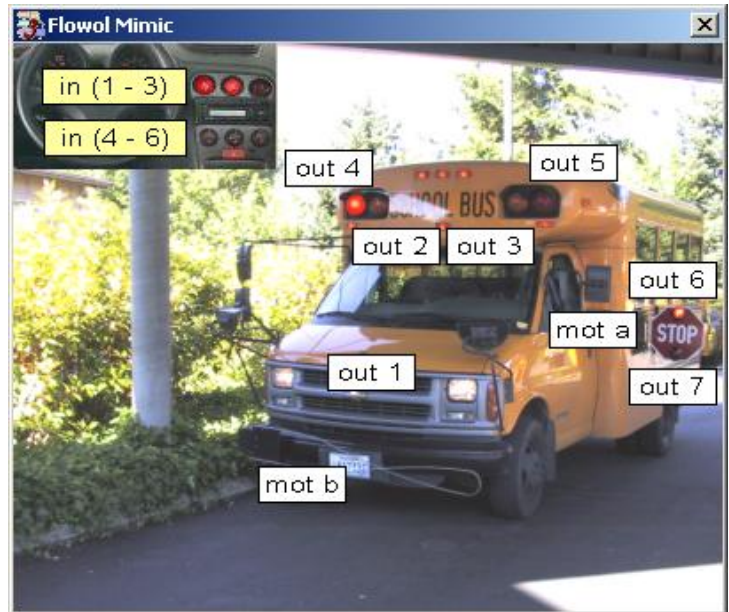


## Keeping students safe

How do the special features on a school bus keep the students safe?

Open the 'School Bus' mimic and explore its special features.

The driver has six button [Inputs 1 to 6] available to control the various lights, Stop paddle, guard etc but these buttons should be used in sequence to operate the safety features in the right order.



**Activity 1:** Use the first input and build a program to operate the general side lights.

**Activity 2:** Create another program to control the flashing amber lights to indicate to passing motorist that the bus is about to stop.

**Activity 3:** The crossing arm should then be deployed to ensure the children cannot pass near to the front of the bus. Why would crossing just in front of the bus be a hazard?

**Activity 4:** The red alternating traffic warning lights should then be activated.

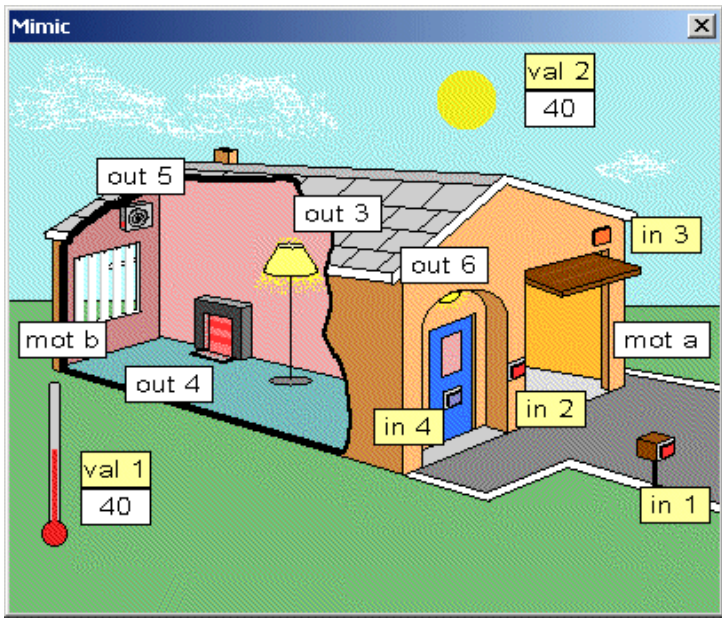
**Activity 5:** The Stop paddle [arm] should then be displayed.

**Activity 6:** Finally the alternating lights on the stop paddle illuminated.

# The Automatic Home

What automatic control features do you have in your home?

Open the 'Auto Rambler' mimic and explore its features.



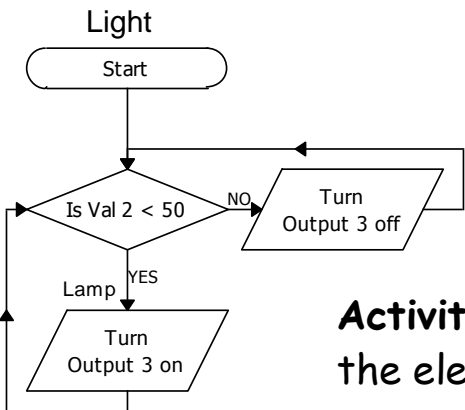
### Activity 1:

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

### 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. By clicking on the numbers, near the Sun and thermometer, you can make them increase or decrease in steps of 5. This simulates the change in the values (val) of daylight brightness and temperature.



**Activity 3:** Construct a flowchart, like the one shown, to turn on the inside light, only when the daylight 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. Make them automatic.

**Activity 7:** Have a look at activity 2 again, the one with the person detector. How can you improve this system?

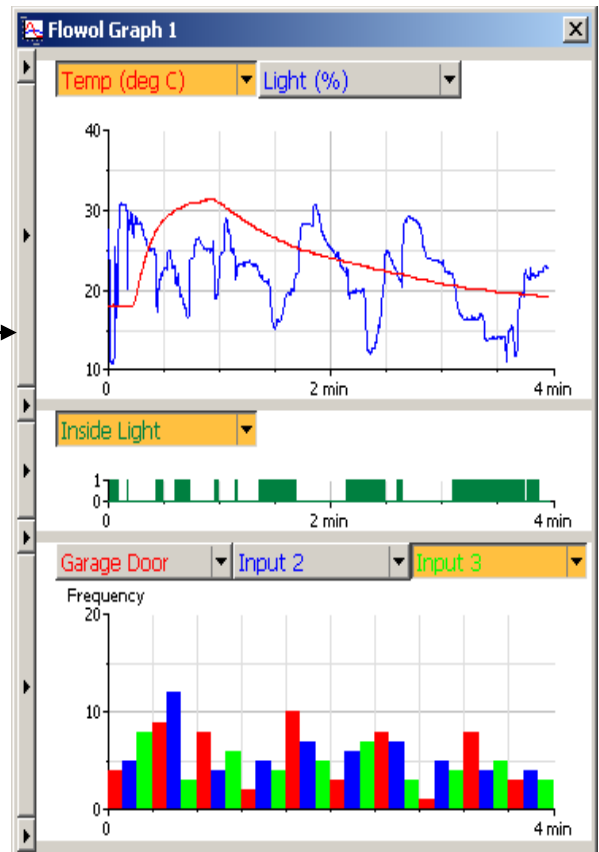
**Activity 8:** Make the push button on the front door trigger the sound of a bell or a recording of a dog barking.

Make sure you **re-saved** your 'autoR.flo' program at each stage.

Monitoring and logging data can be done with the Flowol Graph. Choose 'Show graph' from the icon bar at the top of the screen. Select what information you want to observe from the side toolbars. [Your tutor may have to show you this the first time].

**Activity 9**

Run your 'autoR.flo' program and keep changing the different inputs. My graphs are logging: temperature, daylight brightness, the inside light and how often the garage door is opened.



## The Intelligent Greenhouse

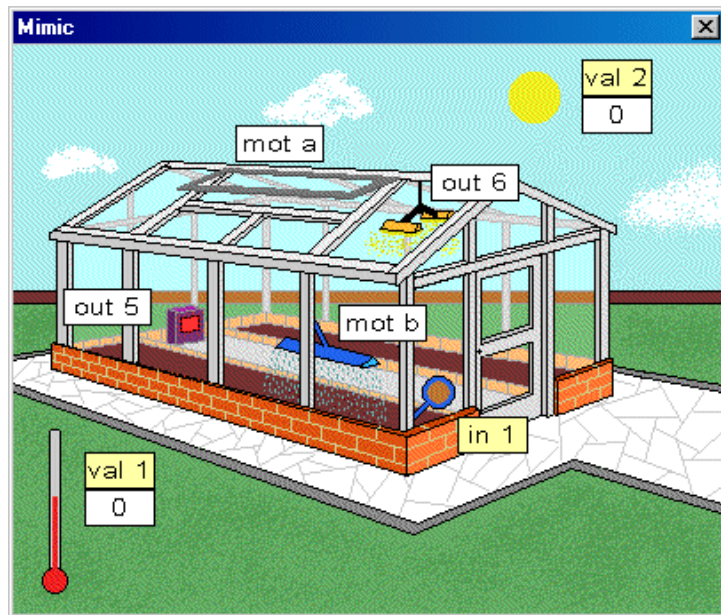
Why do we have greenhouses? Research and determine the best conditions for growing plants.

Open the 'Greenhouse' mimic, and explore the functions of this greenhouse.

Input 1 is a moisture sensor [off when dry]. The temperature and light level 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.flo' program.

### Extra Activities

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