

KEEP I. T. EASY

Flowol 4: Control Systems and Robotics Software

Flowol 4 Tutorial

KEEP I.T. EASY

Flowol 4 Tutorial

© 2010-2014 Keep I.T. Easy

www.flowol.com
support@flowol.com

For Flowol 4, version 4.17

Copyright © 2010-2014 Keep I.T. Easy

This documentation is subject to copyright.

If you have a licensed copy of the Flowol 4 software, you may print as many copies of all or part of this tutorial for your use under your Flowol 4 software end user license.

If you do not have a licensed copy of Flowol 4, you may use this documentation to understand more about Flowol 4 and whether it may be suitable for your purposes. In this case you may not print and use this documentation in a teaching setting.

Keep I.T. Easy acknowledge that there may be errors or omissions in this publication for which responsibility cannot be assumed. No liability will be accepted for loss or damage resulting from the use of information contained in this documentation or from uses as described.

Table of Contents

Introduction	4
Chapter 1: Launching Flowol 4	5
Downloading and Installing Flowol 4	5
Activating Flowol 4	6
Registering Flowol 4	6
Launching Flowol 4	7
Chapter 2: Rapid Start	8
The Flowol 4 Screen	8
Dragging Flowchart Symbols onto the Workspace	9
Core Symbols	9
Tools	10
Running the Flowchart	11
Connecting to an Interface	11
Flowol Options	12
Chapter 3: Tutorial	13
Flowol 4 Bundled Mimics	13
Zebra Crossing	13
Crossing Patrol Mimic	16
All Stop	17
Crosswalk Mimic	19
Double Traffic Lights	20
Hiding and Showing the Mimic Window	21
Zooming the Flowchart View	22
Adding Labels to the Flowchart	22
Slowing and Pausing the Flowchart	22
The Lighthouse Mimic	23
Lighthouse with a Switch	24
Lighthouse with Subroutines	25
Other Mimics	25
Sound and Speech (not available in Linux beta)	26
The Clipboard: Using Cut, Copy and Paste	28
Controlling Electric Motors	29
The Mobile Mimic	29

Using Analog values with the Crib Mobile	31
Controlling the Big Wheel/Ferris Wheel	32
Using Variables.....	33
Using Variables with the Big Wheel/Ferris Wheel	34
Controlling a Motorized Barrier	35
Other uses of Variables.....	36
More Mimics.....	37
Mimic Packs.....	37
Printing and Workspace Size (not available in Linux beta)	37
Flowol 4 Options	38
Chapter 4: Hardware Interfaces	39
VEX IQ.....	42
VEX Cortex	46
FlowGo.....	50
Deltronics Junior Control IT Box	51
Commotion CoCo 3/Deltronics USB	51
Fischertechnik Robo TX.....	52
Fischertechnik Robo LT	57
Fischertechnik Robo	57
Fischertechnik Intelligent Interface.....	58
Control Station	58
Smart Box	59
Contact Controller/Contact Controller Plus	59
Commotion Coco (earlier metal version).....	60
Deltronics/Commotion Junior Serial.....	60
Deltronics/Commotion Serial Interface.....	60
Deltronics Serial Plus.....	60
Deltronics Digital Serial Adapter	60
LEGO Dacta Control Lab (LEGO Interface B).....	61
Arduino.....	61
PICAXE.....	65
Brainy-USB and Brainy-Motor Kit.....	66
Intelligent FirmwerX FlowIC Kit	67
Solo18/Solo28 Microcontroller Kits	68
Numbered Interface	69
Chapter 5: Advanced Features.....	70
Graphs	70
Solar Water Heating Panel.....	74
Random Numbers.....	75
Subroutine Parameters	76

Using multiple Mimics and/or Interfaces.....	76
Advanced Thread Control	77
Chapter 6: Mimic Activities	78
Zebra Crossing	79
Crossing Patrol	80
Bridge Lights	81
All-way Stop	82
Crosswalk	83
Intersection Lights	84
Lighthouse	85
Crosswalk with Stop Light	87
Pelican Crossing	88
Robot	89
Santa.....	90
Mobile	91
Ferris Wheel/Big Wheel	93
Level Crossing	95
Railroad Crossing	97
Automatic Home	99
Greenhouse	101
School Bus.....	102
Train Set 1	103
Train Set 2	105
Index.....	106

Introduction

This tutorial is organized into several chapters:

Chapter 1: Launching Flowol 4

Instructions for downloading, installing and launching Flowol 4.

Chapter 2: Rapid Start

A quick overview of the Flowol screen and flowchart symbols. *This chapter is suitable for users who are experienced with Flowol or other graphical computer programming software.*

Chapter 3: Tutorial

This chapter is targeted for all users. The comprehensive tutorial provides a hands-on approach to learning Flowol with the mimics, taking the reader through a series of problem solving exercises. Each step uses a control program to explore different elements of control systems. Programming techniques and features of Flowol are introduced along the way.

Chapter 4: Hardware Interfaces

Find out how to connect your hardware interface. This section contains brief descriptions of all the hardware interfaces that Flowol supports. Some interfaces have special features and/or limitations which are detailed in this chapter.

Chapter 5: Advanced Features

Learn how to use advanced features such as graphs, random number generation, subroutine parameters and advanced thread control.

Chapter 6: Mimic Activities

Student activity sheets for each mimic included with the Flowol 4 software.



Chapter 1: Launching Flowol 4

Flowol 4 allows students to develop logical reasoning and problem solving talents, develop programming skills and explore the world of automatic control systems and robots.

Downloading and Installing Flowol 4

Flowol 4 is distributed as an internet download for both Windows PC and Apple Mac computers. A beta (test) download for the Linux desktop is also available.

Downloading Flowol 4

1. Navigate to <http://www.flowol.com/Download.aspx>.
2. Enter the license key that was delivered with your purchase. The license key is of the form **XXXX-XXXX-XXXX-XXXX-XXXX-XXXX** where **X** is either a letter or number. Letters will be capitalized as you type them, but you do need to enter the dashes. If your license key was delivered to you in an e-mail, you may find it easier to copy and paste the key into the field. Click on **OK**.
3. If you have a multi-machine license, or a whole-site license, then you will be given the opportunity to register your purchase of Flowol 4 at this point. (See section on **Registration** below for more information)
4. At the top of the page is a link to the end-user license agreement for the level of license that you have. Please click on the link, read and print the license for your records. **Note that installing the software indicates your acceptance of the terms of the license agreement.**
5. Every Flowol 4 license key provides the Flowol 4 software for Windows PC, Apple Mac or Linux computers (or more than one type if you have a multi-computer license).

Installing Flowol 4 on a single computer

Follow the instructions on the download webpage to install Flowol 4 onto a single computer. On the left you will find the download **Flowol4.msi** for Windows PC, in the middle **Flowol4.dmg** for Apple Mac and on the right the beta **Flowol4.zip** for Linux.

Installing onto multiple computers (on a school network)

Below the instructions for a single computer on the webpage are the instructions for installing Flowol 4 onto multiple workstations. **Follow the instructions on the webpage carefully**; otherwise you may have to enter the license key on each computer during activation.

For more detailed information on installing Flowol 4, please consult the following webpage: <http://www.flowol.com/flowol4/Installation.aspx>

Activating Flowol 4

Once Flowol 4 has been installed, it will need to be activated. Activation verifies that Flowol 4 is only used on the number of computers granted in the license. This only needs to be done once.

Activation may prompt you to re-enter your license key. If your license key was delivered to you in an e-mail, you may find it easier to copy and paste the key into the field.



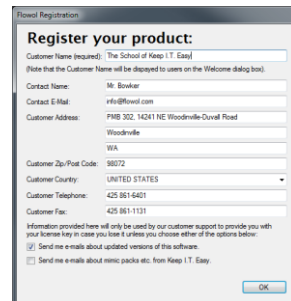
Activation requires connecting over the internet. No personal information will be sent in order to activate the software.

To find out more please see <http://www.flowol.com/flowol4/Activation.aspx>.

Registering Flowol 4

After activation, you have the opportunity to register your purchase of Flowol 4 with Keep I.T. Easy, the developers of Flowol.

Only the first field is required here, and if you have purchased Flowol for use at your school or college, please use the school/college name here as it will be displayed on the Welcome dialog box.



If you do complete the rest of the information, then our customer support can use it to provide you with your license key in case you ever lose it.

To receive e-mails about updated versions of Flowol, or other products that Keep I.T. Easy may develop for Flowol (for example mimic packs), please check the appropriate boxes. Keep I.T. Easy will only use your information as you have specified.

Launching Flowol 4

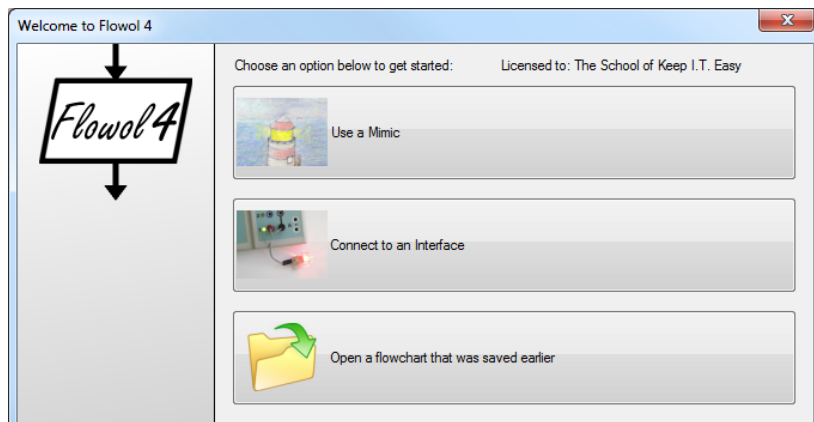
On Windows PC, launch Flowol 4 by clicking on the Flowol 4 program icon in the Start Menu/Start Screen.

On Apple Mac, launch Flowol 4 by double-clicking on the Flowol 4 application bundle.

On Linux desktop, launch Flowol 4 from under the Education section in your list/menu of applications.

The Welcome dialog

When Flowol is first launched, the Welcome dialog box will be displayed. Choose one of the options to get started:



Use a Mimic (an on-screen control system simulation)

Click on the top button to start a new flowchart with a mimic. A mimic is an on-screen pictorial simulation of a control system. Flowol 4 is bundled with a progressive series of mimics which introduce programming concepts in manageable steps. See the Activities section of this document for a set of student tasks for each mimic.

Mimics are an ideal tool to allow all of the students in the course to work at their own pace. The results provide differentiation and allow the teacher to assess the students' achievements individually.



Connect to an Interface

Click the second button to connect to a hardware interface. Flowol supports a wide variety of hardware interfaces including control boxes and microcontrollers.

If you have already used a hardware interface on this computer, then an additional option will appear allowing you to directly reselect that interface.



Open a flowchart file

Click the last button to load an existing flowchart (.flo) file. The flowchart file contains information specifying which mimic and/or interface it uses.

Flowol 4 can load flowcharts that were created with Flowol 2 or Flowol 3.

Chapter 2

Chapter 2: Rapid Start

New users to Flowol, please skip to the **Tutorial** on page 13

A quick overview of the Flowol screen and flowchart symbols. This section is suitable for users who are experienced with Flowol or other graphical programming software.

The Flowol 4 Screen

Below is the Flowol 4 screen once the Ferris Wheel/Big Wheel mimic has been chosen:

New, Open, Save and Print buttons

Cut, Copy, Paste and Delete buttons

Undo and Redo buttons

Click to connect to an **Interface**, use a **Mimic** or use **Variables**. Or click **More...** for more features.

Drag new **flowchart symbols** from here

Add a **Label**

Select mode

Add connecting **line** between symbols

The **Workspace** is where your flowchart is constructed

The **Mimic Window** is a two or three dimensional graphical representation of a control system or robot. Move the mouse pointer over the mimic window to reveal the **Show** icon in the top left and click there to toggle the mimic labels. Then click on inputs/outputs in the mimic to toggle their state. Drag the corner of the mimic window to resize it.

The **Status Panel** shows the current status of all inputs, outputs and variables in use. When the flowchart is not running, click to toggle the state of outputs to test their effect. When no interface is connected, click on the inputs to simulate a state change.

Drag the divider: left and right to adjust the size of the Status Panel.

Run/Stop the flowchart, adjust flowchart **Speed Pause** and **Single Step**

Zoom buttons

Flowol Ferris Wheel Mimic

File Settings Edit Help

Start

Turn Red on, Yellow on, Blue on

Delay 1

Turn Red off, Yellow off, Blue off

Stop

Ferris Wheel

Button 1
Button 2
Gate
Steps
Red
Yellow
Blue
Wheel

Input

Output

Slow Fast

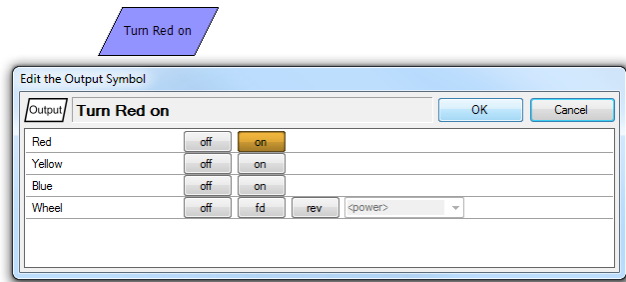
100%

Dragging Flowchart Symbols onto the Workspace

To construct your flowchart, drag symbols from the left toolbar onto the workspace. When a symbol is placed, a **Prompt Box** opens at the bottom of the Flowol screen.


Prompt Boxes

Click on the buttons in the prompt box to complete the instructions in the symbol.



When the contents of the symbol are correct, click **OK** on the prompt box to apply the changes to the flowchart.

Core Symbols

The left toolbar shows those symbols which can be dragged onto the workspace. To keep the toolbar clean and uncluttered, only the symbols which are currently relevant are shown. An example of this is with variables. If you want to assign a value to a variable, first add the Variables to the Status Panel by clicking on the  button.



Start/Stop/Sub Symbol

Use the **Start symbol** at the beginning of the flowchart program. Multiple Start symbols can be placed, and their programs will run in parallel.

Use the **Stop symbol** to end a program. Or use the Stop symbol to end a subroutine.

Use the **Sub symbol** to head a subroutine. Note that you will need to define subroutines before they can be called from the main program.



Output Symbol

Use the **Output symbol** to turn a digital output on or off. Or turn a motor to the forward, reverse or off state. The motors on the mimics and some interfaces can also have their speed defined as a percentage.

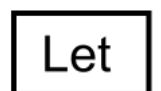
The VEX Cortex and fischertechnik Robo TX also support encoder motors which can be programmed to turn a certain amount. See the Hardware Interfaces section for more information.

The VEX Cortex also supports Servo motors which can be instructed to turn to a particular angle and hold that position.




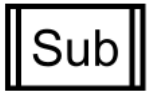
Delay Symbol

A **Delay symbol** pauses the processing of the flowchart for a given number of seconds.




Let Symbol

A **Let symbol** assigns a numeric value to a variable. To use variables in the flowchart, first click the  button to add variables to the Status Panel.



Call Subroutine Symbol

This symbol will run a **subroutine**. When the subroutine is complete (reaches a Stop), the program will return to the call subroutine symbol. Usually subroutines are called once, but this symbol can repeat the subroutine a number of times. Before using a call subroutine symbol, define your subroutine with a  symbol.



Decision Symbol

Use the **Decision symbol** to branch the flowchart based on a condition. If the condition is true (if **YES**) then proceed one way, if false (if **NO**), proceed another way. Decision symbols can check the status of an input switch (Is Input 1 on?) or of an analog sensor (Is Temperature > 50°C?). A decision symbol is also used to check the value of a variable (Is $x = 10$?).

Every decision symbol must have both a YES and a NO line proceeding from it.

Tools



The Text Tool

The text tool is used to add a label to the flowchart. Click and drag the *T* onto the workspace to add a label. Then use the keyboard to adjust the text.



The Select Tool

When the **Select tool** is chosen, clicking on the flowchart will select symbols, lines and labels. Single items can be selected by clicking on them. Additional items can be added or removed from the selection by holding the **Ctrl/control** key on the keyboard and clicking. Alternatively, click and drag to create a bounding rectangle to select multiple items.

A selected symbol can be edited with the prompt box. Edit a label can be by double-clicking on it.

Symbols and labels can be moved by dragging them around the workspace. When symbols are moved, the lines will follow.

Use the appropriate toolbar buttons to cut, copy or delete selected items.



The Line Tool

Use the **Line tool** to connect symbols together and define the flow of the program. All symbols except for a **Stop** symbol need a line proceeding from them to tell the computer what to run next. Every **Decision** symbol needs both a **YES** and a **NO** line.

To add a line, select the line tool from the tool bar. Then, click on the symbol from which you want the line to flow and it will turn green. If the symbol is a decision symbol, then a small popup will appear allowing you to choose whether this should be a **YES** or a **NO** line. Finally, click on the destination symbol to draw the line. Flowol will choose an appropriate route for the line around any existing symbols.

Running the Flowchart



To run the flowchart, click the **Run button** at the bottom-left of the screen. Execution will begin at all Start symbols and proceed fairly quickly. Click the **Stop button** to stop running.



To adjust the speed that the flowchart runs, drag the **speed slider** to the left to slow the flowchart down, and to the right to speed it up. Return the slider to the middle for normal speed.



Use the **Pause button** to pause execution, and then use the **Single Step button** to advance the flowchart by one symbol. If the flowchart doesn't run when you click on the Run button, check that the Pause button isn't selected.

Connecting to an Interface

Flowol 4 supports a variety of interface hardware from a range of manufacturers. Please see the section at the end of this document which details each interface and notes any special features that they might have.

When an interface is selected, it will be displayed in the Status Panel.



Interface Options

Some interfaces have options that can be set. Click on the Options button to view and modify these options.

An important option is the **connection**. This is the port (Serial, USB or Bluetooth) to which your interface is connected. Select the connection from the dropdown in the Options dialog.

Connecting

Click on the black connection icon to instruct Flowol to make a connection to your interface.

Once a connection has been made, the icon turns green. Click again to disconnect from the interface.

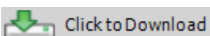
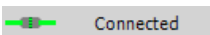
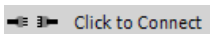
Downloading

Some interfaces have an integrated microcontroller which can run a downloaded Flowol flowchart remotely. If this is the case, the download button will be present. Click this button to compile and download the flowchart to the connected interface.

Inputs

Digital inputs (initially named Input x) are switches and can be either on or off. Analog inputs (initially named Val x) are sensors which report a numeric reading. When the

FlowGo				
Click to Connect				
Download Supported				
Input	1	Input 1	I	
	2	Input 2	I	
	3	Input 3	I	
	4	Input 4	I	
	1	Val 1	0.0 %	I ▾
	2	Val 2	0.0 %	I ▾
	Output	1	Output 1	I
		2	Output 2	I
3		Output 3	I	
4		Output 4	I	
5		Output 5	I	
6		Output 6	I	
	A	Motor A	I	
	B	Motor B	I	



interface is disconnected, click on the inputs to toggle their state, and click and drag the analog value left and right. When the interface is connected, the states of the inputs on the interface itself are shown.



Analog sensors can be of various types, including a light sensor, temperature sensor and sound sensor. When the mouse pointer is over the status panel, dropdown arrows appear to the right of the analog readings. Click on this dropdown arrow to choose which sensor is connected and calibrate the reading.

Outputs

Digital outputs (initially named Output x) are lights or buzzers and can be either on or off. Motors (initially named Motor A) can be either off, turning forwards or in reverse. When the flowchart is not running, click on the outputs to toggle their state. With motors, click with the right mouse button to put the motor into reverse.



Naming Inputs and Outputs

Once you have connected your control system or robot to your interface, it is useful to name the inputs and outputs appropriately. Once they are named, it will be no longer necessary to remember which numbered connection on the interface it is attached to.

When the mouse pointer is over the status panel, renaming icons will appear to the right of the inputs/outputs that can be renamed. Click on this icon to name the input/output with a suitable name (for example *barrier* or *left wheel*).

Flowol Options

See page 37 for details on adjusting the workspace size.

See page 38 for details on adjusting the toolbar icon size, locating 2D mimics from specific directories and changing the colors used.

Chapter 3: Tutorial

A hands-on approach to learning Flowol 4 using the mimics.

Flowol 4 Bundled Mimics

Flowol 4 is bundled with a series of mimics. Some of these mimics show road and traffic features. In these cases there are often two mimics, one for the United Kingdom market (and countries where traffic drives on the left) and another for the United States market (and countries where traffic drives on the right).



The set of mimics that are shown is configured by the Country Setting. This is automatically set when you first activate Flowol 4 and should match the country in which you purchased Flowol 4. The country setting can be changed by opening the **Settings** → **Options...** menu item and selecting your country from the Country dropdown.

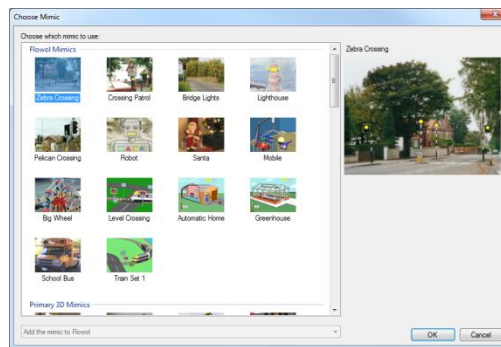


United States users please turn to page 17 for the **All Stop** mimic.

Zebra Crossing

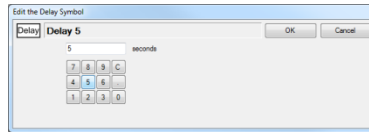
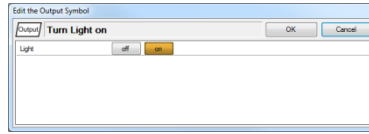
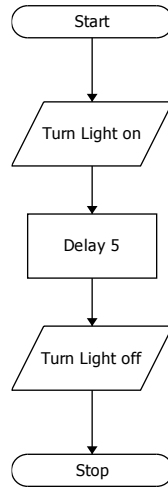
Set up a new Flowol workspace with the Zebra Crossing mimic:

1. If necessary, click on the **New Document** button: 
2. Choose **Use a Mimic**: 
3. Then select the **Zebra Crossing mimic** and click OK.



Constructing the Flowchart

1. Drag a Start/Stop symbol from the left toolbar onto the workspace. If the symbol outline is red, move it slightly and try again in order to place it in the desired position. In the prompt box, click **Start**.
2. Drag an Output symbol and position it directly beneath the Start symbol. Use the prompt box to select **Turn Light on**. Click OK.
3. Drag a Delay symbol and position it beneath. Use the prompt box to select **Delay 5**. Click OK.
4. Drag another Output symbol and position it beneath. Use the prompts to select **Turn Light off**. Click OK.
5. Drag a Start/Stop symbol and position it last. In the prompt box, click **Stop**.

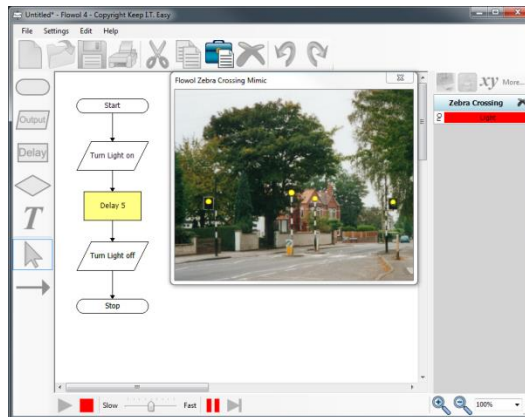


-
- a. Click on the **Line Tool**.
 - b. Then click on the **Start Symbol**. It will highlight green.
 - c. Then click on the symbol beneath it, **Turn Light on**, to draw the line.
 - d. Continue clicking on the **from** and **to** symbols to complete the rest of the lines to match the flowchart shown above.



Running the Flowchart

Click on the Run button at the bottom-left to run this short flowchart. Notice the lights illuminate in the mimic and the state of the **Light** output is shown in the Status Panel.



Saving the Flowchart

If the flowchart is still running, first click on the Stop button to stop the flowchart. Then click on the Save button on the top toolbar, enter a filename and click Save.

Removing a Mistake

Symbols or lines placed incorrectly can be deleted:



1. Select the **Edit Tool** on the left toolbar.
2. Click on the incorrect symbol or line to select it. Selected symbols or lines turn blue. (To skip step 1, click directly on the symbol or line with the right mouse button).



3. Then click on the **Delete button** on the top toolbar to delete the selection.

Using Undo and Redo



Click on the **Undo button** on the top toolbar to undo the last action that was taken. The Undo button can be used repeatedly to undo all actions back to the empty workspace, or the last time a flowchart was loaded.



If you press the Undo button too many times, press the **Redo button** to re-do the action that was just undone.

Changing a Symbol

Let's change the delay on our zebra crossing to 2 seconds, instead of 5 seconds:

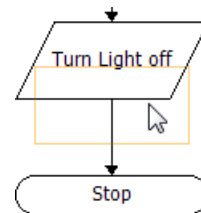


1. Select the **Edit Tool**.
2. Click on the Delay symbol in the flowchart to select it.
3. In the prompt box click on the **C** to clear the number and click on **2**. Then click on OK.

Inserting an additional Symbol

To insert a second delay symbol just before the Stop:

1. Click and drag a delay symbol from the left toolbar onto the workspace, and drop just above the Stop symbol when the outline of the new delay symbol is an orange rectangle.
2. Placing a symbol when the outline is orange will insert the new symbol in the place shown. The Stop symbol will be moved down and the line will automatically be redrawn.
3. Use the prompt box to make this also a delay of 2 seconds.



To make the Light flash forever (An Infinite Loop)

Currently, the flowchart just runs through once. But if we delete the Stop symbol and connect, the second Delay back to the first Output symbol (as shown) we will have a flowchart that continues for ever:



1. Select the **Edit Tool** on the left toolbar.
2. Click on the Stop symbol.
3. Click on the **Delete button** on the top toolbar to remove the Stop symbol.



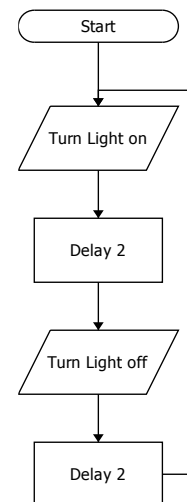
4. Click on the **Line tool** on the left toolbar.
5. Click on the second Delay symbol (turning it green).
6. Then click on the top Output symbol to draw the line back to the top.



7. Run the flowchart and see the light continue to flash.



8. Stop the flowchart with the **Stop button**. ■



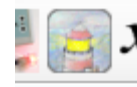


Crossing Patrol Mimic

While keeping the current flashing light flowchart, let's change the mimic to the Crossing Patrol:



1. Click on the **Select Mimic** button above the Status Panel.
2. Choose the **Crossing Patrol** Mimic and click on OK.

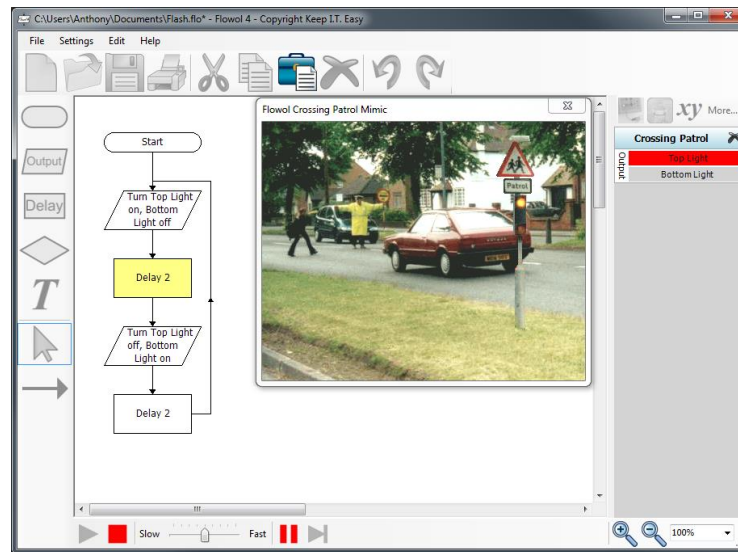
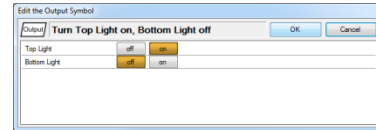


Now the Zebra Crossing Mimic has been replaced by the Crossing Patrol, the output used in the flowchart now refers to the **Top Light**.

Controlling both Outputs

Adjust the flowchart to flash the two outputs alternately by editing each output symbol:

1. Select the Edit Tool.
2. Select the first Output symbol and modify it to read **Turn Top Light on, Bottom Light off**. Click OK.
3. Select the second Output symbol and modify it to read **Turn Top Light off, Bottom Light on**. Click OK.
4. Run the flowchart and observe that the two lights flash alternately.





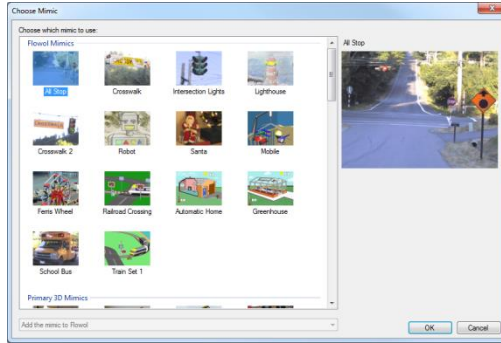
All Stop

Set up a new Flowol workspace with the All Stop mimic:



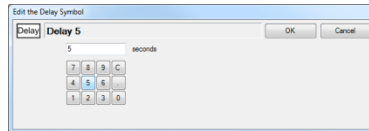
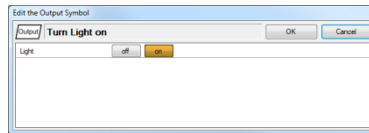
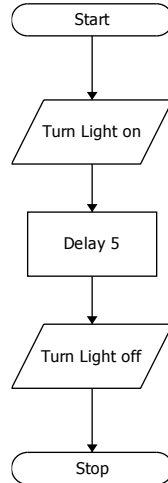
United Kingdom users please turn to page 20 for the **Double Traffic Lights** mimic.

1. If necessary, click on the **New Document** button: 
2. Choose **Use a Mimic**: 
3. Then select the **All Stop mimic** and click OK.



Constructing the Flowchart

1. Drag a Start/Stop symbol from the left toolbar onto the workspace. If the symbol outline is red, move it slightly and try again in order to place it in the desired position. In the prompt box, click **Start**.
2. Drag an Output symbol and position it directly beneath the Start symbol. Use the prompt box to select **Turn Light on**. Click OK.
3. Drag a Delay symbol and position it beneath. Use the prompt box to select **Delay 5**. Click OK.
4. Drag another Output symbol and position it beneath. Use the prompts to select **Turn Light off**. Click OK.
5. Drag a Start/Stop symbol and position it last. In the prompt box, click **Stop**.



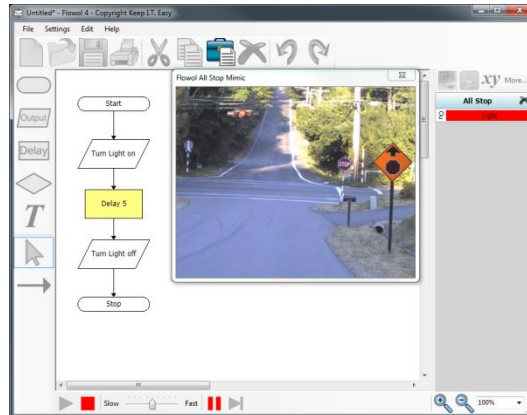
6. If when dragging the symbols onto the workspace they were dropped close enough to the symbol above, a line will have automatically been drawn connecting the two symbols. When this happens the outline of the symbol will turn orange. Otherwise, the lines can be explicitly added using the Line Tool:



- a. Click on the **Line Tool**.
- b. Then click on the **Start** Symbol. It will highlight green.
- c. Then click on the symbol beneath it, **Turn Light on**, to draw the line.
- d. Continue clicking on the **from** and **to** symbols to complete the rest of the lines to match the flowchart shown above.

Running the Flowchart

▶ Click on the Run button at the bottom-left to run this short flowchart. Notice the lights illuminate in the mimic and the state of the **Light** output is shown in the Status Panel.



Saving the Flowchart

If the flowchart is still running, first click on the Stop button to stop the flowchart. Then click on the Save button on the top toolbar, enter a filename and click Save.

Removing a Mistake

Symbols or lines placed incorrectly can be deleted:



1. Select the **Edit Tool** on the left toolbar.
2. Click on the incorrect symbol or line to select it. Selected symbols or lines turn blue. (To skip step 1, click directly on the symbol or line with the right mouse button).



3. Then click on the **Delete button** on the top toolbar to delete the selection.

Using Undo and Redo



Click on the **Undo button** on the top toolbar to undo the last action that was taken. The Undo button can be used repeatedly to undo all actions back to the empty workspace, or the last time a flowchart was loaded.



If you press the Undo button too many times, press the **Redo button** to re-do the action that was just undone.

Changing a Symbol

Let's change the delay on our stop light to 2 seconds, instead of 5 seconds:

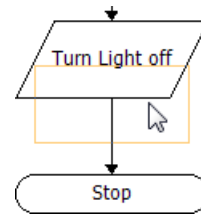


1. Select the **Edit Tool**.
2. Click on the Delay symbol in the flowchart to select it.
3. In the prompt box click on the **C** to clear the number and click on **2**. Then click on OK.

Inserting an additional Symbol

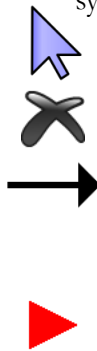
To insert a second delay symbol just before the Stop:

1. Click and drag a delay symbol from the left toolbar onto the workspace, and drop just above the Stop symbol when the outline of the new delay symbol is an orange rectangle.
2. Placing a symbol when the outline is orange will insert the new symbol in the place shown. The Stop symbol will be moved down and the line will automatically be redrawn.
3. Use the prompt box to make this also a delay of 2 seconds.

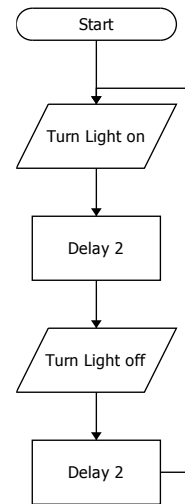


To make the Light flash forever (An Infinite Loop)

Currently the flowchart just runs through once. But if we delete the Stop symbol and connect, the second Delay back to the first Output symbol (as shown) we will have a flowchart that continues forever:



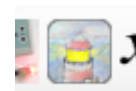
1. Select the **Edit Tool** on the left toolbar.
2. Click on the Stop symbol.
3. Click on the **Delete button** on the top toolbar to remove the Stop symbol.
4. Click on the **Line tool** on the left toolbar.
5. Click on the second Delay symbol (turning it green).
6. Then click on the top Output symbol to draw the line back to the top.
7. Run the flowchart and see the light continue to flash.
8. Stop the flowchart with the **Stop button**. ■



Crosswalk Mimic

While keeping the current flashing light flowchart, let's change the mimic to the Crosswalk:

1. Click on the **Select Mimic** button above the Status Panel.
2. Choose the **Crosswalk Mimic** and click on OK.

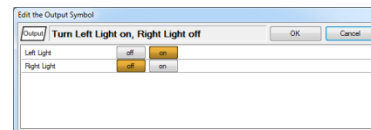


Now the All Stop Mimic has been replaced by the Crosswalk, the output used in the flowchart now refers to the **Left Light**.

Controlling both Outputs

Adjust the flowchart to flash the two outputs alternately by editing each output symbol:

1. Select the Edit Tool.
2. Select the first Output symbol and modify it to read **Turn Left Light on, Right Light off**. Click OK.
3. Select the second Output symbol and modify it to read **Turn Left Light off, Right Light on**. Click OK.
4. Run the flowchart and observe that the two lights flash alternately.





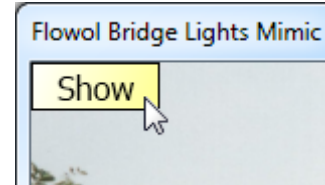
Double Traffic Lights

The double traffic lights extend what was learned with the previous mimics to 6 outputs: 3 outputs for each set of traffic lights.

1. If necessary, click on the **New Document** button:
2. Choose **Use a Mimic**:
3. Then select either the **Bridge Lights (UK)** or **Intersection Lights (US)** mimic.

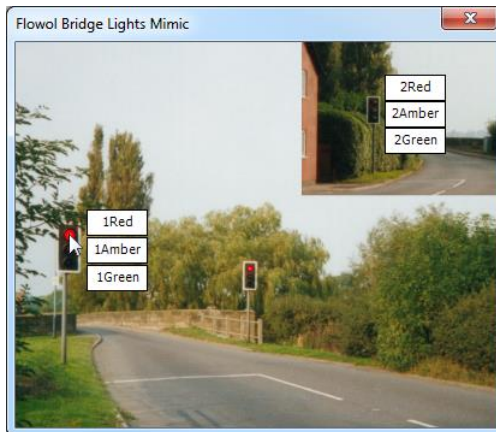
Showing Labels on the Mimic

When the mouse pointer hovers over the mimic window, an icon appears in the top-left corner. Click this icon to show input/output labels on the mimic window. Click the icon again to hide the labels:



Testing the Outputs by Clicking on the Mimic

In order to test the appearance of the outputs, click on each light directly on the mimic window. The state can also be toggled by clicking on the outputs in the status panel.

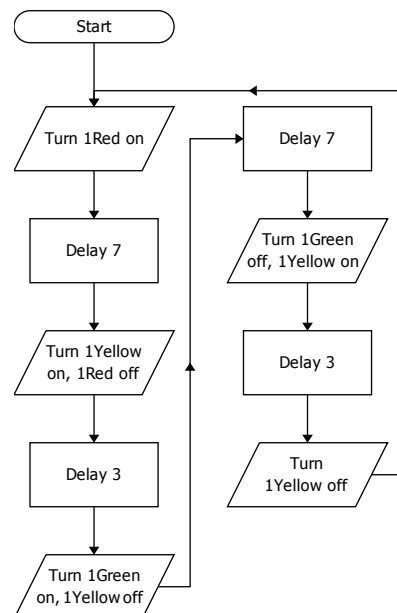


Construct and Run the Flowchart

Copy the flowchart on the right to control a single set of traffic lights. If you need guidance on how to position, fill and connect the symbols review the last few pages of this tutorial. Click on the **Run** button at the bottom-left to run this short flowchart.

Single Traffic Light Sequence

The flowchart to the right is not quite correct for either the UK or US traffic light sequences. Use the **Edit tool** to select the symbols to make the correction.



Both Sets of Traffic Lights

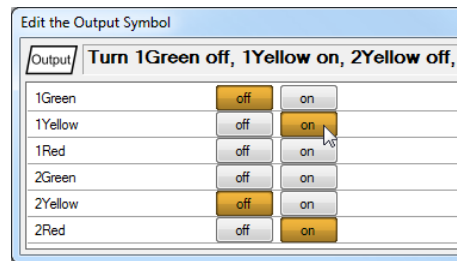
The final step is to develop the flowchart to control both sets of traffic lights, so that they provide a safe control of traffic.

First, edit each of the output symbols to add control commands for the second set of traffic lights.

This results in both directions having the yellow/amber light displayed at the same time. A better solution would be to insert additional delay and output symbols so that one direction would complete the sequence to red before the other direction would move to yellow/amber and then green.

Editing Output Symbols

The Output symbol prompt box enables the setting of multiple outputs in the same symbol. Click on the **On** and **Off** buttons to determine which outputs are controlled. To remove an output from the symbol all together, click on its orange **On/Off** button. The buttons toggle.

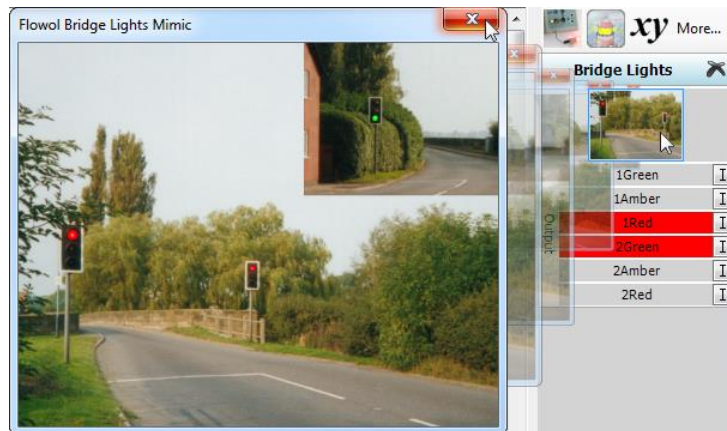


Tooltips

When an output symbol is controlling 3 or more outputs, the text may be truncated if it does not fit in the symbol. When this is the case, and you move the mouse pointer over the symbol, a tooltip will appear showing the full text.

Hiding and Showing the Mimic Window

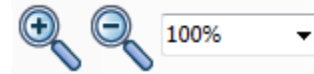
Click on the close icon of the Mimic window to hide it. A mimic icon will then appear in the Mimic area of the Status Panel. Click on this icon to show the mimic window again.



Zooming the Flowchart View

As the sequence of symbols gets longer it quickly fills the screen. The scroll bars can be used to scroll the workspace to view the relevant part of a large flowchart.

Also use the zoom tool to adjust the scale that the workspace is displayed on the screen. The zoom buttons can be found in the bottom-right of the Flowol window.



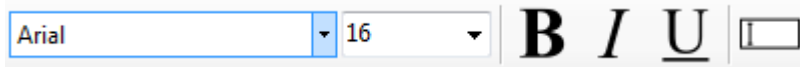
If the scale is less than 80% then tooltips will be displayed on the workspace when the mouse pointer is positioned over any symbol.

T Adding Labels to the Flowchart

The **Text Tool** is used to add a label to the flowchart. Click and drag the *T* onto the workspace to add a label. Then use the keyboard to adjust the text.

Labels can be selected and moved in a similar way to symbols using the **Edit Tool**. To select and move multiple labels and symbols at once, choose the Edit Tool, and then drag a bounding rectangle around a group of items to select them all. Then click and drag the group to a new location.

When selected, the label's font, size and style can be adjusted using the buttons on the Font toolbar at the top of the window:



To edit the text in a label, double click on the label with the edit tool, or click once to select it and then use the **Edit Label** button to edit the text.

Slowing and Pausing the Flowchart



To run the flowchart, click the **Run button** at the bottom-left of the screen. Execution will begin at all Start symbols and proceed fairly quickly. Click the **Stop button** to stop running.



To adjust the speed that the flowchart runs, drag the **speed slider** to the left to slow the flowchart down, and to the right to speed it up. Return the slider to the middle for normal speed.



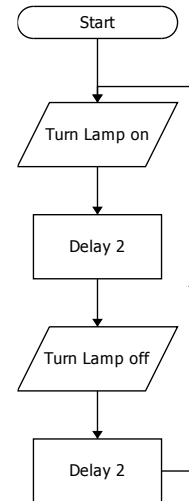
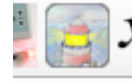
Use the **Pause button** to pause execution, and then use the **Single Step button** to advance the flowchart by one symbol. If the flowchart doesn't run when you click on the Run button, check that the Pause button isn't selected.



The Lighthouse Mimic

Either create a new document, choose the **Lighthouse mimic** and construct the flowchart to the right.

Or load the flowchart created earlier for the Zebra Crossing/All Stop mimic and change the mimic to the **Lighthouse** by clicking on the **Select Mimic** button above the status panel.



Moving Sections of Flowchart around the Workspace





1. Select the **Edit Tool**.
2. Drag a rectangle around all 5 symbols to select them.
3. Click and drag any symbol to move the whole group.

Copying a Section of Flowchart

1. First select the 5 symbols in the flowchart (the same way as steps 1 and 2 above).
2. Then there are two ways of copying the symbols:

Either hold down the **Shift key** and drag any symbol. The mouse cursor will have a little + sign next to it:



Or click the **Copy** button  on the toolbar to copy the selection to the clipboard, and then click the **Paste** button  to place a copy.

Parallel Programming – More than one Start

Now that there are two **Start symbols** in the flowchart they will both run together, simultaneously when you run the flowchart.

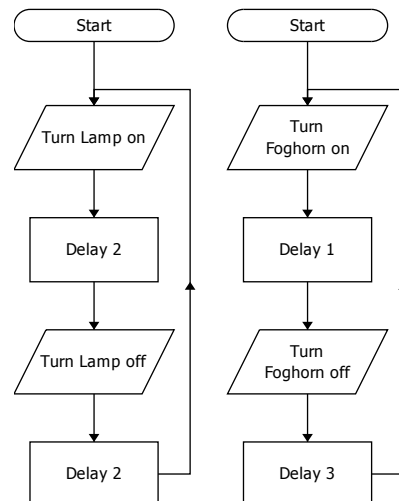
This is **parallel programming** and each Start begins a separate **programming thread**.

Adjust the Second Flowchart to control the Foghorn

Currently the second flowchart is an exact copy of the first, so let's make edits so that it will control the

Foghorn:

1. Adjust both output symbols.
2. Adjust the delays. Experiment with different values.
3. Run the flowchart.



Lighthouse with a Switch

Outputs are the means the control system has of **acting** on the environment. In the Lighthouse mimic there are three outputs: the **Lamp**, the **Foghorn** and the inside **Lights**.

Inputs are **sensory devices** the control system uses to **detect** and **measure** the environment. The Lighthouse mimic has one input, the **Sun**. It represents a digital light sensor and can be either on or off.

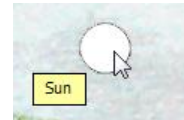
Using a real light sensor with an Interface

Turn to the section on interface hardware (page 39) to connect an interface to Flowol which is linked to the Lighthouse mimic. When an interface is connected, the state of the inputs is the real-time state of the real light sensor.

Simulating the light sensor input

When no interface hardware is connected, the state of the digital input can be simulated in two ways:

1. Click on the Sun on the **mimic window** to toggle the input on and off.
2. Click on the Sun input in the **status panel** to toggle the input on and off.



Lighthouse		
In	Sun	I
	Lamp	I
Output	Foghorn	I
	Lights	I

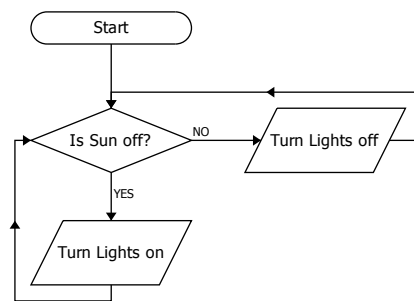
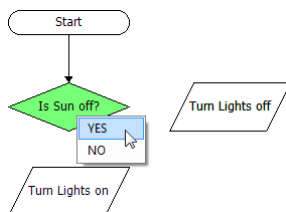
The current state of the input is visible both on the mimic window and in the status panel.

Responding to the Input in the Flowchart

The flowchart must respond to the state of the input, therefore a **decision symbol** is used. The **condition** tests the state of the Input and the flowchart **branches** on the result. The condition is only evaluated, and the state of the input is only observed, for the fraction of a second when the decision symbol is being run.

Construct the following flowchart to use the Sun input to control the inside lights in the lighthouse.

Two lines need to be drawn from the decision symbol. Choose the line tool and when you click the decision symbol a small popup will ask whether you are drawing the **YES** or the **NO** line.



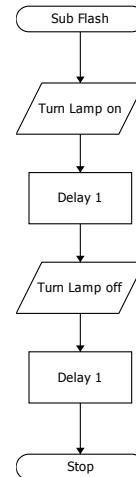
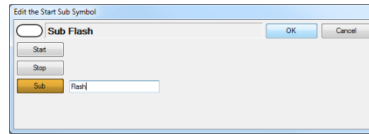
After the Lights output has been set, the flowchart loops back around to the decision symbol to continually re-check the status of the Sun input.

Insert decision symbols into the other flowcharts so that the Lamp and Foghorn is also automatically controlled by the Sun.

Lighthouse with Subroutines

When a section of flowchart needs to be repeated, or reused, it is ideal for placement inside a **subroutine**. Let's rebuild part of the lighthouse flowchart using a subroutine:

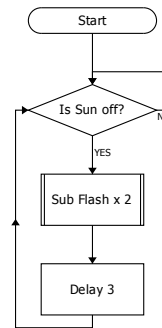
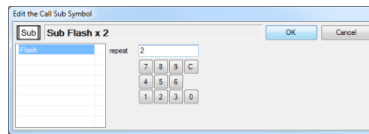
1. Start with a New document and choose the Lighthouse mimic.
2. First build a subroutine called **Flash**:
 - a. Drag a Start/Stop symbol onto the workspace, but this time use it to define a subroutine.
 - b. Follow on with outputs and delays to flash the Lamp once.
 - c. End the subroutine definition with a Stop symbol.



Calling the Subroutine from the main flowchart

For a subroutine to be executed, it needs to be called (invoked) from a main flowchart (from a thread which begins with a Start symbol).

Now that there is a subroutine defined on the workspace, the Call Sub symbol has become available on the left toolbar. Use it to build the main flowchart.



When the flowchart runs and encounters a Call Sub symbol, the subroutine is then run. After the subroutine is complete (reaches a Stop) execution will return back to the Call Sub symbol. In the above example, the Flash subroutine is repeated twice.

Subroutines can be called from within other subroutines. This is called **nesting** subroutines.

Distinct flashing sequence of a lighthouse

Lighthouses have distinct flashing sequences so that ships can identify which lighthouse they can see. If they can see two or more lighthouses they can triangulate their position.

Adjust the flowchart to create a distinct flashing sequence.

Other Mimics

Flowol 4 is bundled with other mimics that make use of digital inputs and outputs. Apply what you've learned so far in the tutorial to complete these mimics:



Pelican Crossing



Crosswalk 2



Robot



Santa

Review the Student Activity sheets at the end of this document for ideas.

Sound and Speech (not available in Linux beta)

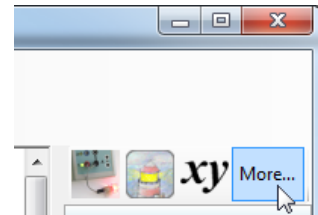
Sound can give extra realism to the control situations. For example, consider the foghorn on the lighthouse, audible warning on level/railroad crossings, or sounds to help the visually impaired cope more easily.



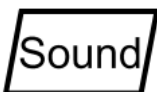
Adding the Sound and Speech Feature to Flowol

To use Sound and Speech, the feature needs to be first added to your workspace:

1. Click on the **More...** button at the top-right of the window, above the Status Panel.
2. Choose the **Sound and Speech** feature and click **OK**.

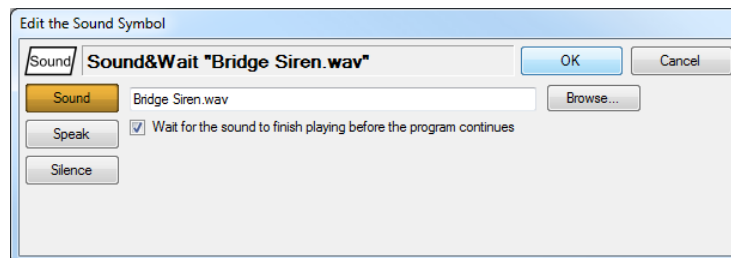


Once the feature has been added, the Sound symbol is now available on the left toolbar. Incorporate this into your flowchart.



Playing a Sound file

Drag a sound symbol onto the workspace, choose Sound in the prompt box and then browse to the sound file. Flowol 4 can play **WAV files** which can be recorded by most popular sound recording software (not included):



If the box **'Wait for the sound to finish playing before the program continues'** is checked, then when run, the flowchart will pause at the Sound symbol until the WAV file has played completely. Otherwise execution will proceed immediately while the sound plays in the background.

For some sample sound files, go to <http://www.flowol.com/flowol4/Sounds.aspx>

Relative and Absolute File paths

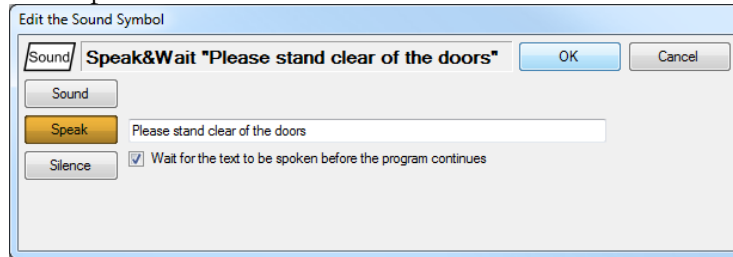
A **relative path** will be used to reference the sound file if it is located in the same directory or subdirectory of the saved flowchart (.flo) file. Otherwise an **absolute path** (a full path from the root directory) will be used. If the flowchart is subsequently saved, then the references to the sound files will be updated, using relative paths if possible.

Note that the flowchart (.flo file) saves a reference to the sound file. It does not embed a copy of the sound file within it. Therefore, if you modify the sound file (.wav file) on disc, then the modified sound file will play in Flowol. Also, when copying your saved flowchart files to a different computer, be sure to also copy the referenced sound files.

Speaking Text

Modern computers now include **Speech Synthesis** capability as part of the Operating System (either Windows or Mac OS X). Flowol uses the Operating System's **Text-To-Speech** system to artificially synthesize human speech from text.

Drag a sound symbol onto the workspace, choose Speak in the prompt box and then enter the text to be spoken:



If the box **'Wait for the text to be spoken before the program continues'** is checked, then when run, the flowchart will pause at the Sound symbol until all the text has been spoken. Otherwise execution will proceed immediately while the voice speaks in the background.

Silence

If you have sound or speech playing in the background, then the Silence option can be used to silence it.

The Clipboard: Using Cut, Copy and Paste

In Flowol, symbols (with their connecting lines) and labels can be cut/copied to the clipboard. They can then be pasted into either the same workspace, or as a graphic into another piece of software (e.g. a Word processor or Paint program).

Selecting Symbols and Labels



Select the **Edit tool** on the left toolbar then either:

- Click on a single symbol or label to select it (it will turn blue),
- Or click in space and drag a rectangle around a set of symbols and labels to select them.

Hold down the **Ctrl/control** key when clicking or dragging to toggle items in and out of the selection.

Copying and Cutting



Click the **Copy button** on the top toolbar to copy the selection to the clipboard. Copying leaves the selection intact.



Click the **Cut button** on the top toolbar to cut the selection to the clipboard. Cutting deletes the selection from the workspace, leaving it on the clipboard.

Pasting into the Workspace

When there is content (symbols or labels) currently on the clipboard, the Paste button will be enabled. Click the **paste button** to get a copy of the clipboard contents, then position the outline image and click again to paste it onto the workspace.

A flowchart cannot have two different subroutines with the same name. Therefore if you paste a copy of a subroutine definition, it will be pasted with a slightly changed name.

In Flowol 4, a flowchart that's placed onto the clipboard can only be pasted back into the same workspace.

Copying into a word processor (Windows PC only)

When there is a fragment of flowchart on the clipboard it may be pasted into other software (e.g. a word processor) as a graphic (an Enhanced Windows Metafile).

It is also possible to copy and paste the mimic image. Click on the mimic window's title bar and hold down the **Alt** key and press the **PrtScn** key. Then paste into the word processor.

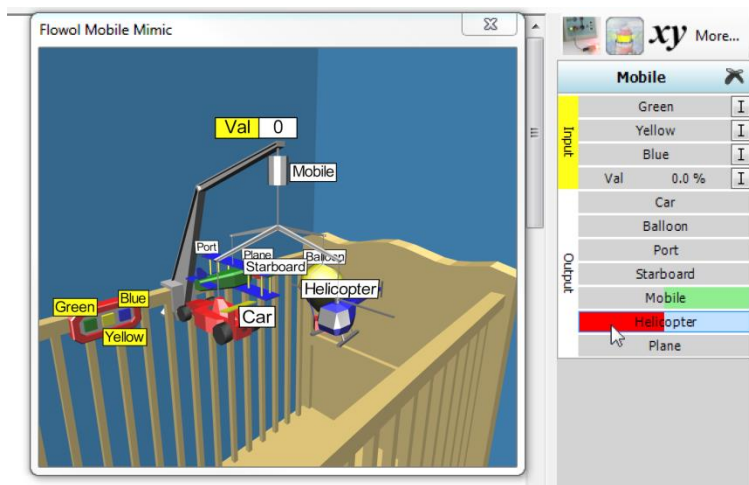
Controlling Electric Motors

Electric motors produce movement. This can result in the continuous rotation of a crib mobile, the forwards and reverse movement of a barrier or two motors (left wheel and right wheel) can control the movement of a floor robot.



The Mobile Mimic

The mobile mimic uses **3D graphics** so that the rotation of the motors can be realistically animated. If you observe jerky animation with the 3D mimics, consult the notes on the website: <http://www.flowol.com/flowol4/MimicPerformance.aspx>

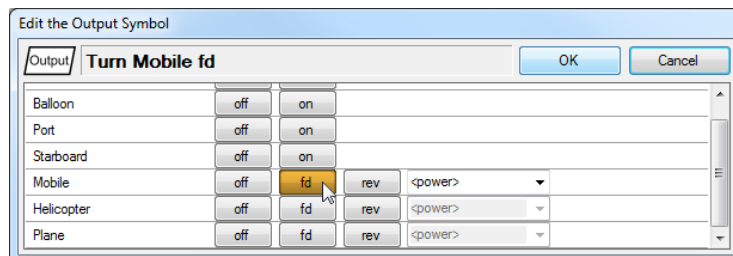


Show the labels on the mimic window, by moving the mouse over the mimic window and clicking on the **Show** icon in the top-left of the window.

Then click on the outputs in the Status Panel and on the mimic window to see what the Mobile can do. When clicking on the **Mobile**, **Helicopter** and **Plane** motors, use the right mouse button to turn the motor in the reverse direction.

Controlling a motor in the flowchart

Motors are controlled with an Output symbol. In the prompt box, the direction of the motor can be set. Use **fd** to make the motor rotate forwards, and use **rev** for reverse.



Multiple outputs (digital and motor) can be set in a single output symbol. Sometimes this will rapidly fill the symbol with text, so separate logically different operations into several symbols to make your flowchart more readable.

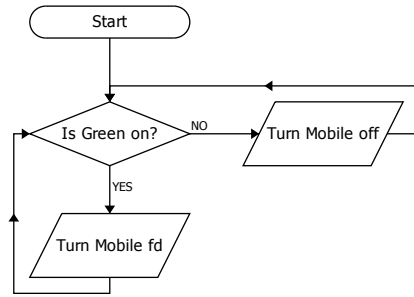
FLOWOL 4 TUTORIAL

Construct the following flowchart to control the mobile with the Green input switch:

Run the program and click on the green input switch to toggle its state:

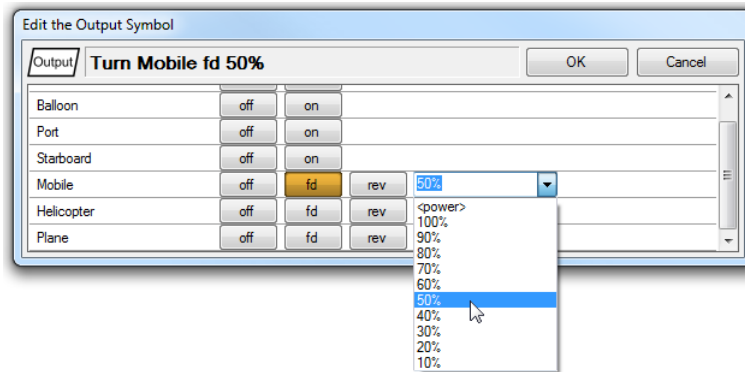


Now build some more programs to control the other lights and motors with the other input switches.



Motor Speed or Power Control

Real motors on models/robots and those on 3D mimics often rotate too quickly. To slow them down we need to give the motors a percentage speed setting. To do this, use the dropdown in the Output symbol prompt box:



Remember to set the power back to 100% in a subsequent symbol if necessary.

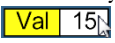
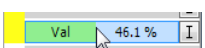
Using Analog values with the Crib Mobile

A switch is a digital input because it can only be either on or off. Analog sensors on the other hand have a range of input values:

- A light sensor can detect variations in the brightness.
- A sound sensor can detect different levels of noise.
- A temperature sensor (a thermometer) reads the current temperature.

Adjusting the Analog values on the Mimic

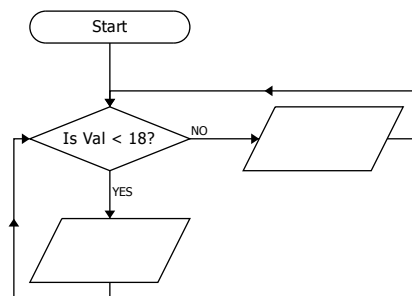
The analog values can be adjusted in two ways:

- **Either** click the reading on the mimic window (clicking with the left mouse button will increase the value by 5 units, clicking with the right mouse button will decrease by 5 units) 
- **Or** click and drag the reading horizontally on the Status panel: 

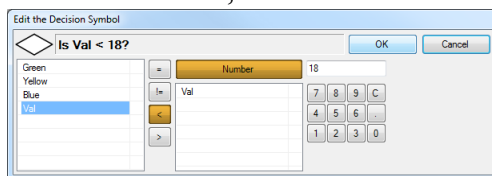
Using Analog values in the Flowchart

As inputs, the analog values feature in the Decision Symbol. Let's make a night light with the light in the hot air balloon. We'll treat the analog value as a light sensor on the top of the mobile's arm. A light sensor gives a high reading when it is light, and a low reading when dark:

Build and complete the flowchart on the right. The instructions in the two output symbols are left as an exercise.



When creating the decision symbol, the condition is a comparison between the analog **Val** and a constant, in this case 18 units:



When creating conditions with analog sensor values, it is best to use $<$ or $>$ comparisons rather than $=$. The sensor reading may vary quickly and may not ever equal the number in question.

Further Activities

To occupy the baby create another program to automatically flash the car rear light and airplane wing tip lights if the light level falls below 60 units.

Every parent's dream is a content baby. Now assume that the analog value is from a sound sensor which gives a high reading when noisy. Create a flowchart which brings the mobile to life if the baby wakes up during the night and cries.

Make the mobile interesting using multiple lights and motors and perhaps even a soothing sound file (see Sound and Speech on page 26). Perhaps the mobile should become more active the greater the noise, and then gradually slow down over time.



Controlling the Big Wheel/Ferris Wheel

Open the **Big Wheel** or **Ferris Wheel** mimic and explore what it can do by clicking on the outputs in the Status Panel.

Red, Yellow and **Blue** are lights embedded into the frame of the wheel structure. **Wheel** is the motor which controls the wheel's rotation.

When the Wheel motor is running, look closely at the triangle on the steps under the wheel. Whenever one of the cars is directly above

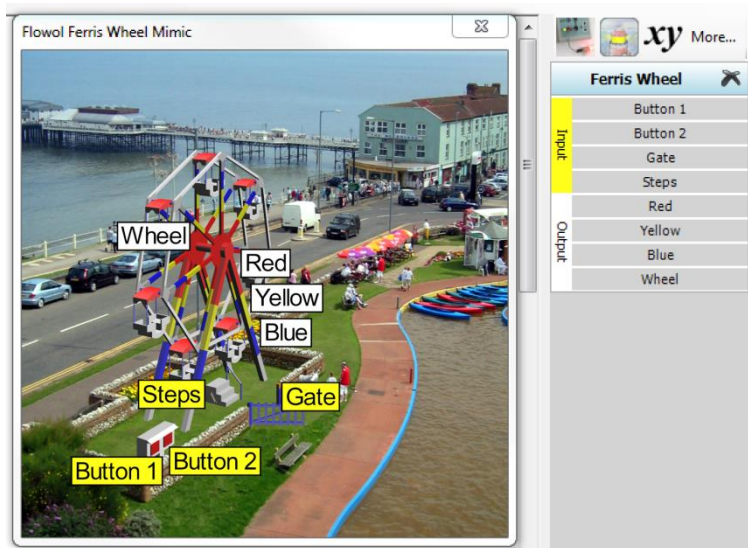


the steps, the triangle turns yellow and the **Steps** input is switched on. This is a **virtual input**, controlled by the mimic itself. Notice that clicking on the **Steps** input on the Status panel has no effect.

Now click on the gate on the mimic window with the left mouse button. The gate closes, click again to open it. When the gate is closed, the **Gate** input is on and is represented by the yellow light at the top of the gatepost.

Flowcharts to Construct

1. To attract the crowd, use **Button 1** to control the lighting effects on the wheel. This might be a simple on/off routine, but flashing sequences are more exciting (use subroutines).
2. Use **Button 2** to control a simple start/stop movement of the wheel. Consider using motor power control to speed up and slow down the wheel.
3. For safety, include a check on the **Gate** switch so that the gate must be closed before the wheel starts to move. The wheel should stop if either the gate is opened, or **Button 2** is turned off.
4. Use speech (see Sound and Speech on page 26) to give the passengers an automatic, verbal instruction to "Hold tight please" just before the wheel begins to move.



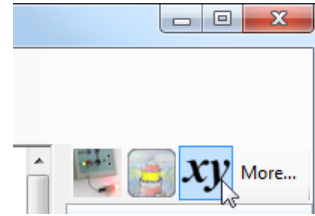
Using Variables

A **variable** is a named, numeric value.

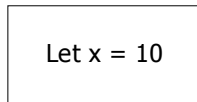
xy

Adding the Variables Feature to Flowol

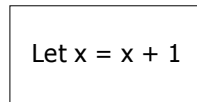
First, add the variables feature to your workspace by clicking on the **xy** button at the top-right of the screen. If this button is not visible because the Status Panel is too narrow, click on the **More...** button and choose **Global Variables**. This will add the Global Variables feature to the Status Panel, listing the values of variables **x** and **y**.



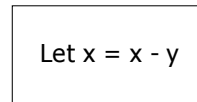
Assigning a value to the variable **x**



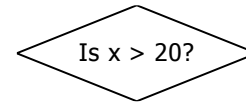
Incrementing **x** by 1



Subtracting the value of **y** from **x**

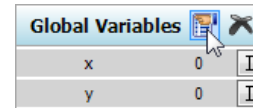


Comparing **x** in a Decision symbol.




Creating More Variables

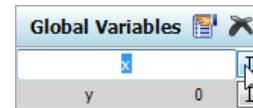
By default, two variables **x** and **y** are provided. To create more, click on the options button on the Global Variables panel. The options dialog allows variables to be added and removed.



Renaming Variables

While the names **x** and **y** might be suitable for small flowcharts, it's often clearer to rename the variables to match their purpose:

1. Move the mouse pointer over the Global Variables status panel to reveal the rename icons.
2. Click on the rename icon  to the right of the variable.
3. Edit the variable name and press Enter/return or click outside the text box to confirm the change.



Everywhere in the flowchart where this variable is referenced will be updated to use the new name.

Variable Scope

The variables have global scope, meaning they can be used anywhere in the flowchart, including in all threads and within subroutines.

Since variables are visible in all threads (simultaneously running parallel flowcharts), they can be set in one thread to trigger the behavior in another. [See page 36 for an example].

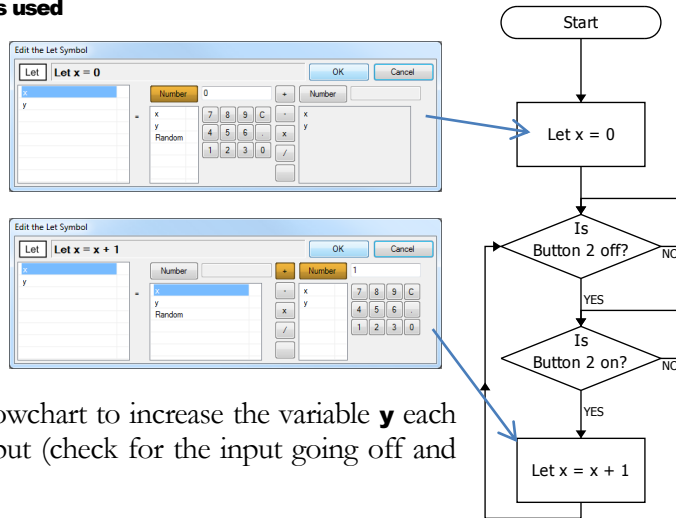
Variable Values

The variables can contain any numeric value between -1.79×10^{308} and $+1.79 \times 10^{308}$ with a precision of 64bits.

Using Variables with the Big Wheel/Ferris Wheel

Counting how many times the ride is used

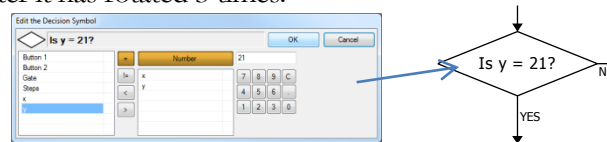
Use variable **x** to count how many times the ride is used. This could be done either by adding the increment **Let** instruction to the existing flowchart, or a separate flowchart could be used, as shown here.



Counting the Cars

Construct a similar counting flowchart to increase the variable **y** each time a car passes the **Steps** input (check for the input going off and then on as above).

Since there are 7 seats, each full rotation of the wheel should increase the value of the variable **y** by 7. Now modify the main flowchart, by introducing a decision symbol, to stop the wheel after it has rotated 3 times.



Stopping at each Car

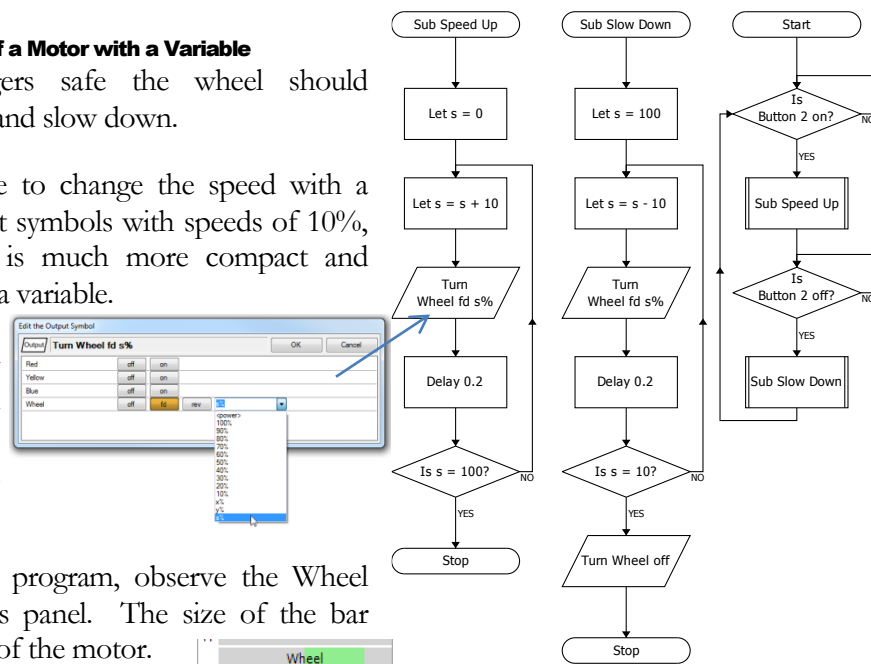
Now we are using the **Steps** input, modify the program so that each of the seven cars stops automatically at the bottom for a short time for the people to get off. [Note a variable is not needed for this solution].

Changing the Speed of a Motor with a Variable

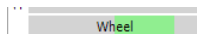
To keep passengers safe the wheel should gradually speed up and slow down.

While it is possible to change the speed with a sequence of Output symbols with speeds of 10%, 20%, 30% etc. it is much more compact and reusable to employ a variable.

Create the new variable **s** and use a speed of **s%** in the output symbols in the subroutines.



When running the program, observe the Wheel motor in the status panel. The size of the bar indicates the speed of the motor.



Controlling a Motorized Barrier

A motorized barrier or garage door moves in one direction for a certain distance and then has to move in the reverse direction by the same amount. Deciding how long the motor should be turned on for can be determined in two ways:

- **Either** by considering the speed of the motor and the distance the barrier has to move, determine the delay time needed while the motor is on.
- **Or** by having a **feedback switch** so the flowchart can turn the motor off when the barrier has reached its final position.

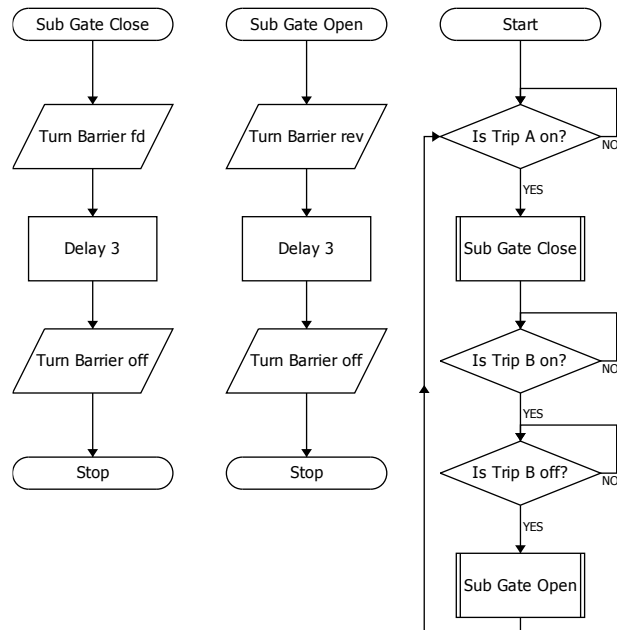


The Level Crossing/Railroad Crossing Mimic

The Level Crossing/Railroad Crossing mimic provides both these scenarios. Open the mimic and by default the barrier will be without feedback switches, just physical buffers at the top and bottom of its movement.

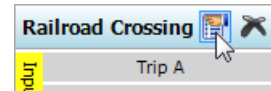
Construct the following flowchart to control the barrier when the train approaches. When the flowchart is run, the train will begin to move around the track, triggering the **Trip A** and **Trip B** inputs as it goes (these are represented by the yellow triangles on the mimic).

The flowchart is structured to use two subroutines: **Gate Close** and **Gate Open**. They are called from the main flowchart when the train is in the appropriate position. Consider carefully the three decision symbols in the main flowchart and how they operate.



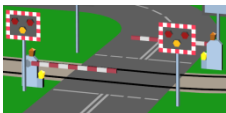
Changing Mimic Options

Click on the Mimic Options button in the mimic's Status Panel. Check the one option **Use feedback switches on the barriers** and click **OK**. This adds two new inputs **Bar Down** and **Bar Up**.

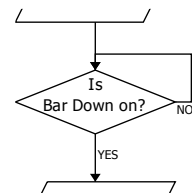


Feedback switches

Run your existing flowchart again. When the barrier reaches its down or up position, either the **Bar Down** or **Bar Up** feedback switches will be on (small yellow indicators on the mimic). Since the delay of 3 seconds is longer than the barrier's motion, the barrier will vibrate indicating that the motor is under stress as it tries to move against the feedback switch.



Modify the **Gate Close** subroutine to use the **Bar Down** feedback switch instead of the Delay as shown here. Make the corresponding modification to the **Gate Open** subroutine.



Other uses of Variables

Counting cars in a parking area

A controlled parking area usually has two barriers, one for entering traffic, and one for exiting traffic. The flowchart will likely have subroutines for opening and closing each barrier.

There are two ways of counting:

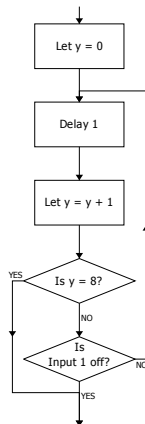
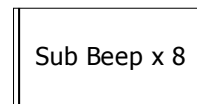
- Use a variable, **cars**, which counts the number of cars in the parking area
- Use a variable, **spaces**, which counts the number of available spaces.

The parking area would then likely have a sign which illuminates when there are no more available spaces. Also, the entry barrier should deny entry to cars who arrive while the parking area is full.

A repeated loop

A variable can be used to repeat an operation a certain number of times. In this example the beep occurs 8 times.

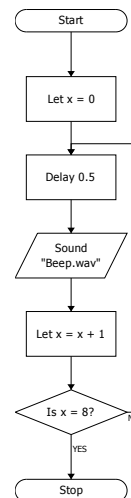
Alternatively, the Beep could be placed into a subroutine. The subroutine could be then called 8 times with the **Call Sub** symbol:



A delay that can be interrupted

A fairground ride should have a specific run time, but it should also be possible for the ride operator to stop it at any time.

Here the flowchart delays for a total of 8 seconds, but after every second, if Input 1 has gone off, the delay is interrupted.



Main thread controlling the operation in another thread

It is possible for one thread to have influence over another through the use of global variables. For example, in a railroad crossing/level crossing scenario the main thread could set the variable **x** to be 1 when it needs the two red lights to start flashing, and then set it back to 0 again to stop; the worker thread would only flash lights when **x = 1**.

Another scenario where this is useful is in a conveyor belt process in a factory. When the belt stops, several simultaneous processes need to be done to the item that's stopped on the belt. These processes may take different lengths of time. So, when each process is finished it increments a variable, and only when the variable count is equal to the number of expected completed processes should the conveyor belt start moving again.

More Mimics

Flowol 4 is bundled with four more mimics that make use of motors. Apply what you've learned so far in the tutorial to complete these mimics:



Automatic Home



Greenhouse



School Bus



Train Set 1

Review the Student Activity sheets at the end of this document for ideas.

Mimic Packs

Keep I.T. Easy sells additional mimic packs which go beyond the mimics included with Flowol 4. Each additional mimic pack includes a set of student activities.

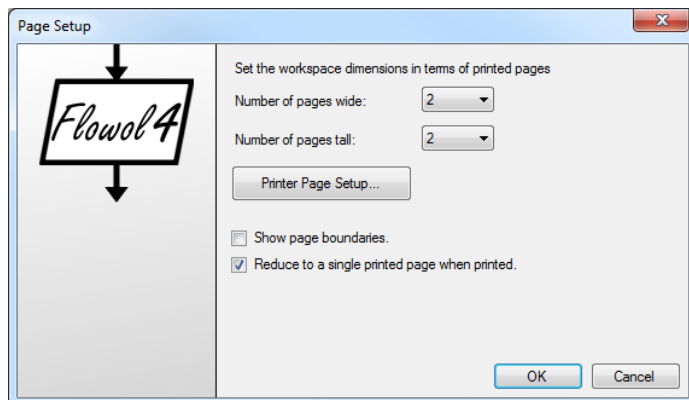
For more information, please see www.flowol.com

Printing and Workspace Size (not available in Linux beta)

The workspace has a default size of four sheets of paper, 2 wide and 2 tall (usually A4 or Letter 8.5" x 11" depending on your printer settings). This can hold about 220 flowchart symbols.

Adjust the workspace size by opening the **Page Setup** dialog (from the File menu) and choosing how many pages wide and tall should form the workspace.

To adjust the actual paper size or orientation (portrait or landscape) that the printer is using, click **Printer Page Setup** to modify the Printer page setting.



By default, Flowol will reduce the flowchart to fit onto a single printed page when printed. To maintain a 100% true-size print, uncheck **Reduce to a single printed page when printed** and Flowol will print onto multiple pages which can be taped together. Check **Show page boundaries** to see the page boundaries on the workspace.

Print Preview

Windows: To see a print preview, click the **Print Preview** option on the File Menu

Mac OS X: Choose **Print** from the file menu, and then click **Preview** on the dialog.

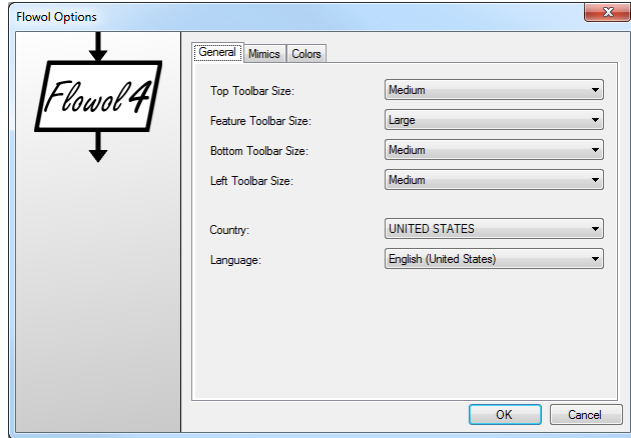
Flowol 4 Options

Open the Options from the **Settings** → **Options...** menu. All of these settings are stored on a per-user basis.

General Options

The icons in the toolbars can be set to different sizes. If your monitor has the space, try a larger size.

The **Country** setting is used to determine which mimics to list. Some mimics will adjust their behavior based on this setting, e.g. whether road traffic drives on the left or the right.

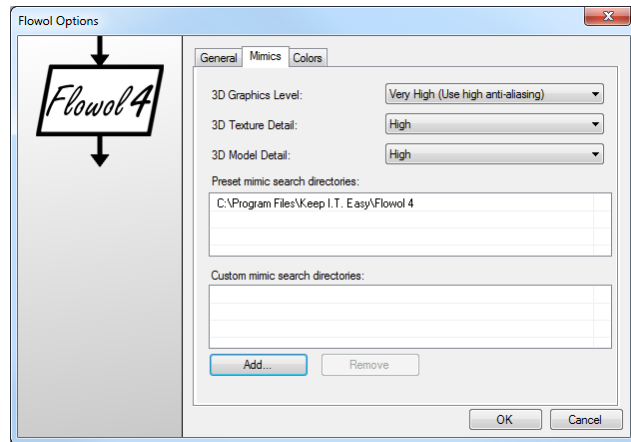


Adjust the **Language** setting to change the text used in the menus and dialog boxes.

Mimic Options

If the 3D mimics are running slowly on your computer, or you get errors when opening the 3D mimics, reduce the settings at the top of the Mimics tab until the mimics work satisfactorily.

Flowol 4 will automatically find any 2D mimics that were installed on the computer with Flowol 2 or 3. If you have created your own 2D mimics with the Mimic Creator, use **Add** to add their directory to the Custom mimic search.



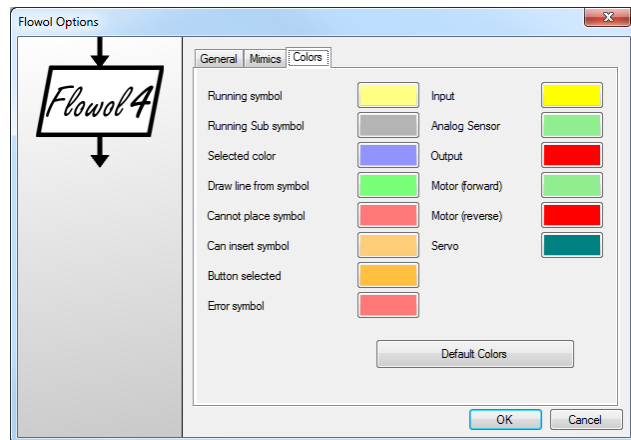
Add to add their directory to the Custom mimic search.

Colors

Use the colors tab to customize the colors used for the symbols and for the Status Panel.

Click on each block of color to adjust the color with the Color dialog box.

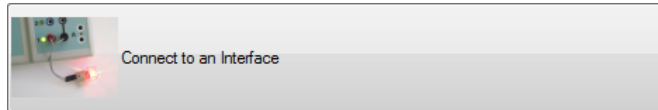
Click on Default Colors to return to the default Flowol 4 color scheme.



Chapter 4

Chapter 4: Hardware Interfaces

Hardware Interfaces are the controllers that bring your control systems and robots to life.
From the Welcome Screen, choose the Connect to an Interface button:



This opens a list of interfaces supported by Flowol 4:



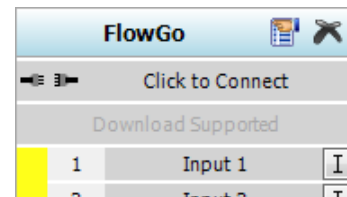
Control Boxes are robust controllers which have sockets for inputs and outputs to be connected.

Microcontrollers are single microchips often with a circuit board to support the inputs and outputs.

Scroll down the list until you find the interface you are looking for. Select it and click on **OK**.

Once you have selected an interface, it will appear on the Welcome dialog box as a separate option making it easier to select next time.

Choosing an interface will add it to the Status Panel.



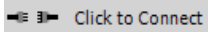


Interface Options

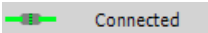
Some interfaces have options that can be set. Click on the Options button to view and modify these options. An important option is the **connection** dropdown. This is the port (**Serial**, **USB** or **Bluetooth**) to which your interface is connected.

To use a Serial connection on a Mac or newer PC computer, you will need to use a **USB to Serial adapter**. If the interface has a USB connection, it's usually better to use a USB cable directly.

Connecting

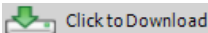


Click on the black connection icon to instruct Flowol to make a connection to your interface.



Once a connection has been made, the icon turns green. Click again to disconnect from the interface.

Downloading



Some interfaces have an integrated microcontroller which can run a downloaded Flowol flowchart remotely. If this is the case, the download button will be present. Click this button to compile and download the flowchart to the connected interface.

Inputs

Digital inputs (initially named Input x) are switches and can be either on or off. Analog inputs (initially named Val x) are sensors which report a numeric reading. When the interface is disconnected, click on the inputs to toggle their state, and click and drag the analog value left and right. When the interface is connected, the states of the inputs on the interface itself are shown.



Analog sensors can be of various types, including a light sensor, temperature sensor and sound sensor. When the mouse pointer is over the status panel, dropdown arrows appear to the right of the analog readings. Click on this dropdown arrow to choose which sensor is connected and calibrate the reading.

Outputs

Digital outputs (initially named Output x) are lights or buzzers and can be either on or off. Motors (initially named Motor A) can be either off, turning forwards or in reverse. When the flowchart is not running, click on the outputs to toggle their state. With motors, click with the right mouse button to put the motor into reverse.



Naming Inputs and Outputs

Once you have connected your control system or robot to your interface, it is useful to name the inputs and outputs appropriately. Once they are named, it will be no longer necessary to remember which numbered connection on the interface it is attached to.

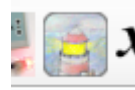
When the mouse pointer is over the status panel, renaming icons will appear to the right of the inputs/outputs that can be renamed. Click on this icon to name the input/output with a suitable name (for example *barrier* or *left wheel*).

Using a Mimic with the Interface

Some mimics, for example the Lighthouse, would work well on screen when simultaneously connected to a real model Lighthouse, connected with an interface.

To use both a mimic and interface in your flowchart:

- From the Welcome dialog, choose the interface to use.
- Then click the **Mimic Button** above the Status Panel.
- Choose the mimic and click **OK**.




The Status Panel will then show how the inputs and outputs on the mimic have been linked to the hardware interface. In the example on the right, **Input 1** is connected to the **Sun** light switch, **Output 1** is connected to the **Lamp**, **Output 2** to the **Foghorn** and **Output 3** to the **Lights**.

Flow		Lighthou	
Connect			
Download			
1	Sun		
2	Input 2		
3	Input 3		
4	Input 4		
1	Val 1	0.0 %	
2	Val 2	0.0 %	
1	Lamp		
2	Foghorn		
3	Lights		
4	Output 4		

Binary I/O Values (Advanced Feature)

Some of the interfaces allow the flowchart to use the input **InBin** and output **OutBin** binary values directly. **InBin** is a read-only decimal value, which when considered in a binary representation, each bit represents the state of an input. **OutBin** is a read-write decimal variable, which when considered in a binary representation, each bit represents the state of an output.

To enable the Binary I/O values, click on the interfaces' options button  and check the **Use binary I/O values** option and click **OK**. The **InBin** and **OutBin** values will be displayed in the Status Panel. Switch on some inputs and outputs to see how the binary values behave.



VEX IQ

The **VEX IQ** is the brain of the VEX IQ robotics platform. There are no dedicated input, output or motor connections on the VEX IQ Brain. There are 12 Smart Ports which can have any device connected (VEX IQ Smart Motor or any Smart Sensor) into any port.

First update the VEX IQ Firmware

Go to <http://www.vexiq.com/firmware> and download the VEX IQ Firmware Update Utility (versions for Windows and Mac OS X are available). Follow the instructions there to update the firmware in your VEX IQ Brain, Controller and devices (motors and sensors).

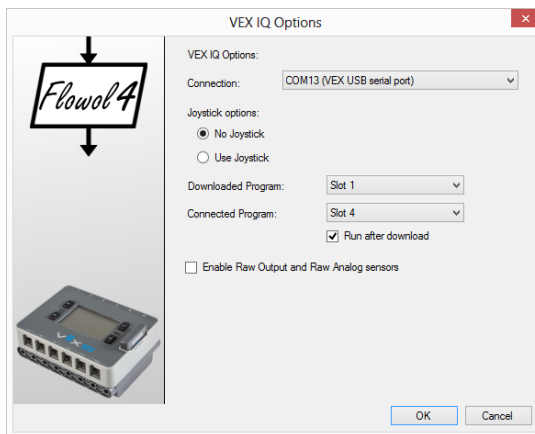
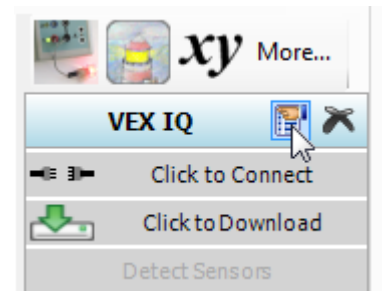
Flowol 4 requires that you have the VEX IQ firmware version 1.09 or later.

The VEX IQ Firmware Update Utility will install the required USB Device Drivers on Windows for the VEX IQ Brain. No driver is required on Mac OS X.

Connecting the VEX IQ to Flowol

Connect the VEX IQ Brain to the computer using the USB Cable. Exit the VEX IQ Firmware Update Utility if you have it open. Then open Flowol, in the Welcome Screen, choose to connect to an interface and choose the VEX IQ and click OK.

Then click on the **Interface Options** button (pictured right) to ensure that the correct COM port is selected and to configure the VEX IQ.



Use the top dropdown to select the correct connection. If the VEX USB serial port (or the tty.usbmodem on Mac OS X) is not shown, check your connection, close the Options dialog box and re-open it to refresh the list of available connections.

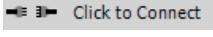
By default, Flowol is setup to create an autonomous program. Click 'Use Joystick' to add the VEX IQ controller to Flowol as inputs to the system to build a program which can use both

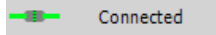
the joystick input as well as sensors.

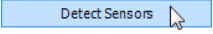

The VEX IQ has 4 slots in its flash storage where programs can be stored. Flowol uses one of these slots (by default Slot 4) for a special program to facilitate Flowol's Connected mode. By default, Flowol will download the program to Slot 1. The slot choices can be adjusted here in the Options dialog.

FLOWOL 4 TUTORIAL

The check box to **Enable Raw Output and Raw Analog sensors** is an advanced feature. Please contact Flowol support (support@flowol.com) if you would like to use it at this time.

After adjusting settings in the Options dialog, click  to connect to the VEX IQ in connected mode. The icon will turn green when connected:



Connect your sensors and motors to the VEX IQ brain and click  to automatically detect which sensors are connected to which port. You can now use the interface monitor below to test motors and see sensor values. And use the  tool to name devices as appropriate. See pages 40 for more details.

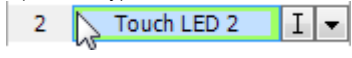


Bumper Switch

The bumper switch is the simplest of all sensors. It's a digital input and can either be **on** (pressed) or **off** (released). Use it with the decision symbol in Flowol to detect if your robot has bumped into an obstacle or another robot. It can also be used as a limit switch.



Touch LED

The Touch LED is both a digital input and an output. Use the Output symbol in Flowol to turn the LED (the illuminated ring on to one of the preset colors). To test the LED, you can click (carefully) around the outside of the Touch LED pane in the Interface Status Panel. . Use the LED as a visual indicator of the status of your program.

The top of the Touch LED is a touch input. When you lightly touch the top of the Touch LED, the input becomes ON. Use the Decision symbol in Flowol to check the status of the input.

The Touch LED adds an element of human interaction with the robot. Light up the LEDs to show the human operator which touch inputs are active.



Distance Sensor

The Distance Sensor uses ultrasonic sound waves to measure distance. By default the sensor is calibrated in centimeters. Use the drop-down to change calibration to inches if preferred.



Color Sensor

The Color Sensor can operate in two modes. By default it detects the color of an object in front of the sensor and yields a numeric value from this table:

Value	Color
0	No object is detected in front of the sensor. If necessary, position the sensor closer to the object. Optimal position is between 2cm and 3cm. This uses the IR reading to return zero if no object is close enough.
1	Red
2	Red-Orange
3	Orange
4	Yellow-Orange
5	Yellow
6	Yellow-Green
7	Green
8	Blue-Green
9	Blue
10	Blue-Violet
11	Violet
12	Red-Violet

(Colors are approximate)

Therefore you might want to detect ranges in value in order to identify the colors desired.

Alternatively the Color Sensor can be configured into Grey Scale mode using the dropdown. In Grey Scale mode, the LED on the color sensor is turned on and the sensor returns a percentage value (0 to 100%). Gray Scale mode is effective as a line tracker sensor.



Gyro Sensor


The Gyro sensor measures turn rate and this in turn is computed as an angle. The best use for a gyro is to make accurate turns. If you want a robot to drive to a location then turn around and return to the same spot, you will need to move an accurate distance, turn exactly 180 degrees, and then drive back the same distance. Gyro's make the turn very accurate.

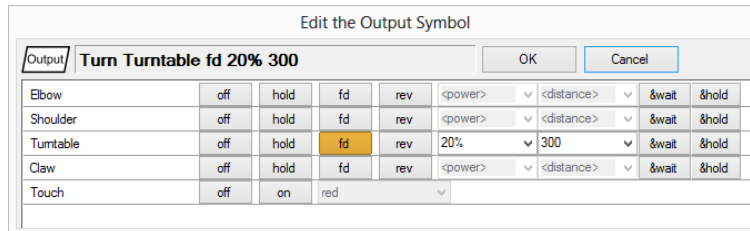
When first connected the gyro calibrates itself so that the observed drift will be less than 0.56 degrees per minute. This takes about 4 seconds.




Smart Motor

The VEX IQ Smart Motor really lives up to its name. First, click on the motor in the status panel to test it. Left-click to go “forward” and right-click for “reverse”. If the motor doesn’t go in the direction that you expect, use the dropdown and choose the **[Reverse Polarity]** option to switch the effect of “forward” and “reverse”.

The motor is controlled using the Output symbol  which can control multiple motors in one symbol. After placing the output symbol you can see the Prompt Box:



off turns the motor off. When off, the motor can be turned by hand.
hold sets the motor to actively hold its position. This is useful for robot arms which may have to hold their position while holding a load. When held, the motor has a faint line in the status panel: 

fd turns the motor forward
rev turns the motor reverse

<power> set the power of the motor movement as a percentage. The Smart Motor will maintain the velocity even if the VEX IQ battery is low, or the motor is under load. If left unset, the motor will use any previously set power or will default to 100%. If power is set to a negative value it will reverse the motor.

<distance> instructs the motor to turn a particular number of degrees and follow a deceleration profile to stop at that target position. There are 360 degrees per revolution of the motor shaft.

&wait when selected will tell the flowchart to wait (and not proceed to the next symbol) until the motor has reached its target distance. Setting this for multiple motors in the same Output symbol will wait until they have all reached their target position.

&hold will instruct the motor to enter a hold state once it’s reached its target position. Otherwise it will enter a Brake state.



LCD

Flowcharts can output information to the LCD display on the VEX IQ. Five lines of text are available.

The LCD is especially useful for displaying variable or analog value information. Enter the name of a variable (or analog value) in braces in the text string to display the corresponding value on the LCD. For example: “Value of x is {x} ”.



VEX Cortex

The **VEX Cortex** is the controller for the VEX Robotics System. It is a very powerful controller supporting many different outputs, motors and sensors. The VEX Cortex can be connected to the computer directly with the USB type A to A cable or via the VEXnet wireless system and the Programming Hardware Kit.

First Update the Master Code on the VEX Cortex

Before using your VEX Cortex with Flowol, check that the Master Code is up to date. On a Windows PC, use the USB cable and the **VEXnet Firmware Upgrade Utility** (from http://www.vexforum.com/wiki/index.php/Software_Downloads) to update the Master Code on your VEX Cortex.



Connecting directly with USB cable

[Windows: Install the VEXnet Serial USB driver (available from **C:\Program Files\Keep I.T. Easy\Flowol 4\Interfaces\VexWindowsDriver**). Use either the 32-bit or 64-bit installer, as appropriate for your version of Windows.]

Launch Flowol 4 and choose the VEX Cortex. If your computer has additional serial ports, use the connection dropdown in the VEX Cortex Options Dialog to choose the appropriate connection.

[Mac: If this is the first time you have used the VEX Cortex with Flowol on the computer, a message will prompt you to install the **VEX Cortex Serial driver**. You will be prompted for the username and password of an admin account to complete installation of the drivers. After installation of the drivers, you may be prompted to restart your Mac.]



Connecting with the VEXnet Joystick

Alternatively, connect the computer to a VEXnet Joystick using the VEX Programming Hardware Kit. The computer and Joystick can then connect over the VEXnet wireless radio with the VEX Cortex (1.0 and 2.0 VEXnet keys supported).


This configuration allows remote programming and debugging of the program as well as using the joystick inputs in your flowchart.

First, pair the Cortex and the VEXnet Joystick using the USB cable to connect them together (see the **Cortex Microcontroller and VEXnet Joystick User Guide** on the vexrobotics.com website).

Then connect the VEX Programming Kit to the computer.

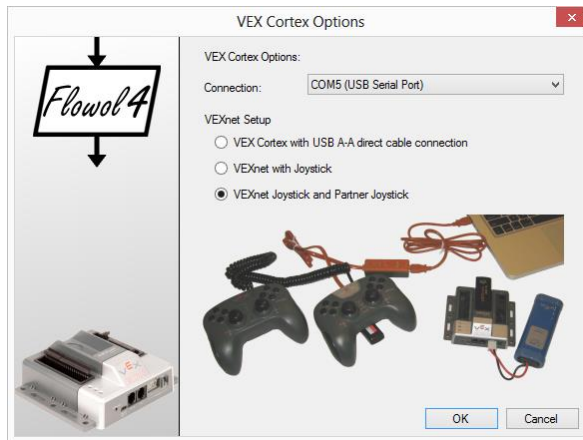
[Windows: Download and install the suitable USB device driver from http://www.vexforum.com/wiki/Software_Downloads.]

[Mac: If you have the Kit from June 2012 or later, connect the USB connection and click Cancel on the 'A new network interface has been detected' dialog box. If you have the earlier Kit from before June 2012, install the Prolific USB driver from http://www.prolific.com.tw/US/ShowProduct.aspx?p_id=229&pcid=41]

Finally, configure Flowol with your chosen VEXnet setup by opening the VEX Cortex Options dialog  and choosing the connection and setup.


The VEXnet joystick has 4 analog axes (range -127 to +127) and 12 input buttons. It also has an X and Y accelerometer which measures how you move the joystick itself.

A VEX Partner Joystick can be connected to the VEXnet Joystick for another operator. This adds 4 more axis and another 12 inputs. Try the joysticks while connected to see how the joystick inputs work.

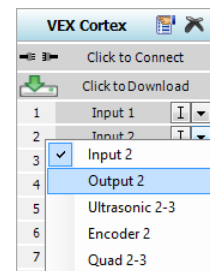


This setup can also be programmed by using the USB cable directly connected to the VEX Cortex. The downloaded program will wait until a connection with the VEXnet joystick is made before beginning.

Digital I/O

There are 12 general purpose digital I/O connections on the VEX Cortex. Initially the first 8 are configured as digital inputs, and then 4 digital outputs. Click on the down arrow  to configure:

- Choose **Input** for Limit Switch, Bumper Switch or Jumper Clip.
- Choose **Output** for LEDs.
- Choose **Ultrasonic** for the Ultrasonic Range Finder. The range finder uses two connections. Connect the one labeled 'Input' into the first connection (e.g. 2 according to the screenshot above) and connect the 'Output' into the second (e.g. 3).
- Choose **Encoder** to use just one of the connections from the Optical Shaft Encoder. When using just one connection, the value will increase whether the shaft is rotated forward or reverse.
- Choose **Quad** to use both connections from the Optical Shaft Encoder. In this configuration, the value will increase when the shaft is turned in one direction and will decrease when turned in the other.



The values from the Encoder and Quad can be used in the Decision and Assignment symbols just as you would a variable (values are integers).

Analog

There are 8 analog connections on the VEX Cortex. Initially these are calibrated as a percentage (0 to 100%). The VEX Cortex has **12-bit ADC** so the system is sensitive to small changes in the sensor reading. Click on the down arrow ▾ to calibrate for different sensors. If the type of sensor you are using is not in the dropdown, use **Val (percent)**.

Motors

There are 10 motor connections on the VEX Cortex. Motor 1 and Motor 10 accept the 2-wire motors directly. Motor 2 through Motor 9 can use a 3-wire motor controller or a Servo motor. Use the down arrow to configure your motors ▾:

- Initially all 10 are configured for a **Motor**. Motors have PWM power control.
- Choose **Encoder Motor** to use a Motor with an Integrated Encoder Module. The encoders connect in a chain to the I2C connection on the VEX Cortex. Flowol requires that the Encoder Modules are connected in order so that the Encoder Module for the lowest numbered motor is connected in the I2C chain closest to the VEX Cortex and so on.
- Choose the **[Reverse Polarity]** option from the dropdown to switch the effect of the Forward and Reverse directions of the motor. This is very useful with wheeled robots when the motor is already connected and you want ‘forward’ to mean forward!
- Choose **Servo** when using a Servo motor.
- If power is set to a negative value it will reverse the motor.

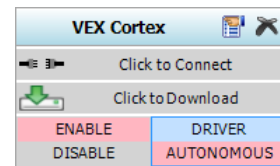
Downloading

Click the Download button to download your flowchart to the VEX Cortex to run autonomously from the computer. The VEX Cortex can run all of the Flowol flowchart symbols. The downloaded program is not erased from the VEX Cortex when it is turned off. However, when reconnecting the VEX Cortex to Flowol 4, the downloaded program is erased and must be downloaded again if you want to use it while away from your computer.

If downloading via the VEXnet joystick, ensure that the VEXnet setup is properly configured in the VEX Cortex Options dialog to an option with Joystick.

VEX Robotics Competition Modes

If you have selected a setup with a VEXnet Joystick, there is the option to **Show Competition Switch**. This adds the **Enable/Disable** and **Driver/Autonomous** switches to the Status Panel on the right of the screen.



If a real competition switch is connected to your VEXnet joystick, then the current mode set on the switch is shown. Otherwise, you can click on the options here to simulate the effect of the competition switch (while the VEX Cortex is connected to Flowol).

When competing, your robot will need to be running with the flowchart downloaded to the VEX Cortex controller. So it is useful to test with the real competition switch and a downloaded flowchart. Read the rules for your competition.

The Start symbols on the flowchart can now have their execution mode set in the prompt box to one of the following settings:

Execution Mode	Effect	Usage
Always (default)	This Start will run whenever the switch is Enabled	This can be used for common operations, e.g. timers or obstacle avoidance logic which you might want during all stages of the competition.
Driver Only	This Start will run only when the switch is in DRIVER mode	The flowchart here should respond to the joystick and control the robot during the Driver state of the competition
Autonomous Only	This Start will run only when the switch is in AUTONOMOUS mode	The flowchart here should drive and control the robot autonomously (without any joystick input at all). All joystick input is disabled in this mode.

When the competition switch is set to DISABLE, all inputs and outputs on the robot are disabled, and none of the flowchart Starts will run.

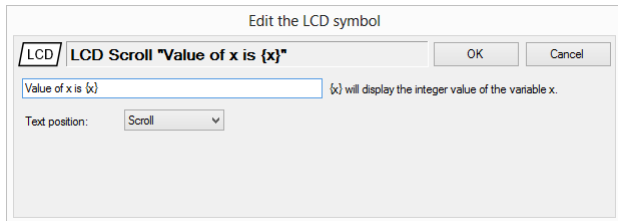
When a flowchart thread isn't running (because the switch is set to DISABLE, or the appropriate mode isn't set on the competition switch to match its execution mode), the flowchart is paused and will resume when the switch changes. If you want to restart a flowchart (rather than pausing), then use the Advanced Thread Control feature of Flowol.



Using the VEX LCD Module

Connect the VEX LCD module to the UART1 socket on the VEX Cortex and select the **Use LCD on UART 1** option in the VEX Cortex Options dialog. This adds the LCD symbol to the left-hand tool bar which sends a line of text to the LCD.

The text can be displayed on either the Top or Bottom lines, or choose Scroll to write the text to the bottom line (and scroll the old text up).



The LCD is especially useful for displaying variable or analog value information. Enter the name of a variable (or analog value) in braces in the text string to display the corresponding value on the LCD. For example: "Value of x is {x}"

Scroll history mode

If Scroll history mode is selected, the buttons on the LCD can be used to scroll back through the last text messages. Left button scrolls up, right button scrolls down, middle button scrolls to end. If scroll history mode isn't used, the LCD buttons are available as inputs in the flowchart.



FlowGo

<http://www.flowol.com/flowol4/Interfaces/FlowGo.aspx>



The **FlowGo** interface by Data Harvest is ideal for Primary or Secondary monitoring and control. It has 4 digital and 2 analog inputs (calibrated sensors), 6 digital outputs and 2 motors with speed control. By default, Flowol tries to connect to the FlowGo via the USB connection. To use a Serial Port connection, open the Options dialog and choose which Serial (COM) port you are using.

Windows: the device driver (32-bit or 64-bit) for FlowGo's USB connection is installed automatically with Flowol 4. Just plug the USB cable to the FlowGo and to the computer and wait a moment for the driver to be automatically loaded.

Mac OS X/Linux: no device driver is required.

Downloading

Flowol 4 can also download the flowchart to the FlowGo to be run remotely:

1. Connect the cable to the interface and click on the Connect button  **Click to Connect** to ensure that Flowol is connected to the interface.
2. Run your flowchart while the interface is connected to check that everything works as expected with your model. Sometimes the delay values or the threshold in analog sensor conditions need to be refined.
3. Click the Download button  **Click to Download** to compile and download the flowchart to the FlowGo.
4. Press the **Go button** in the top of the FlowGo box to run the downloaded program (the green LED will flash). While the downloaded program is running, Flowol cannot connect to the interface. Stop the program running by pressing the Go button again (the green LED will be solid) and then click the Connect button on the screen for Flowol to reconnect to the FlowGo.

There are some limitations to what a downloaded program can do:

- A maximum of two variables can be used in the flowchart. The variables are stored as single byte values and so they can only be integers between 0 and 255.
- Variable assignments are limited to the forms **Let x = constant** or **Let x = x +/- constant/x/y**.
- Motor speeds can only be set with a constant value (e.g. **50%**, not **x%**).
- **InBin** and **OutBin** values are not supported.
- A maximum of 4 parallel threads (separate Start symbols) can be used.
- A maximum of 16 defined subroutines.
- A maximum of 16 distinct delay constants
- About 100 symbols can be used.
- Calls to subroutines with repeat values can only be nested to a depth of 2.
- The Random function is not supported.

If exceeded, these limitations will be pointed out when you try to download your flowchart and you will have to adjust your flowchart. However if you choose the **Limit**

features to those that will work with a flowchart downloaded to the interface option in the interface options, then Flowol will enforce many of these limitations in the symbol prompt boxes, making it easier to create a flowchart that can be downloaded.



Deltronics Junior Control IT Box

<http://www.flowol.com/flowol4/Interfaces/DeltronicsJuniorControlItBox.aspx>

The Deltronics Junior Control IT Box is ideal for Primary monitoring and control. It has 4 digital and 4 analog inputs (with calibrated sensors) and 4 digital outputs and 2 motors with power control. Input and output devices are connected via standard 4mm sockets.

The interface connects via a USB connection. No USB device driver is required.

The Deltronics USB Interfaces are not supported on Linux beta.



Commotion CoCo 3/Deltronics USB

<http://www.flowol.com/flowol4/Interfaces/CommotionCoCo3.aspx>

<http://www.flowol.com/flowol4/Interfaces/DeltronicsUSB.aspx>

The Commotion CoCo 3 or Deltronics USB control boxes are ideal for Primary or Secondary monitoring and control. They each have 6 digital and 4 analog inputs (with calibrated sensors) and 6 digital outputs and 4 motors with power control. Input and output devices are connected via standard 4mm sockets.

The interface connects via a USB connection. No USB device driver is required.

The Deltronics USB Interfaces are not supported on Linux beta.





Fischertechnik Robo TX

<http://www.flowol.com/flowol4/Interfaces/FischertechnikRoboTX.aspx>

The fischertechnik Robo TX Controller is a modern and very powerful controller. It uses 1mm sockets to connect to the whole range of fischertechnik sensors and outputs. The controller has slots for connecting with the fischertechnik construction kit.

Windows USB Connection

- Connect the Robo TX to the computer with the supplied USB Cable.
- Switch on the Robo TX, and if Windows fails to locate the device driver, choose the option to Install from a specific location. Include this location in the search: **C:\Program Files\Keep I.T. Easy\Flowol 4\Interfaces\RoboTXDeviceDriver** (if Flowol 4 was installed to the default directory).
- Launch Flowol 4 and choose the Robo TX. In the interface options, choose the connection that is labeled **COMn (fischertechnik USB ROBO TX Controller)**.

Windows Bluetooth Connection

- Follow the instructions included with the Robo TX to pair the controller with the computer.
- Launch Flowol 4 and choose the Robo TX. In the interface options, choose the connection that is labeled **COMn (Bluetooth ROBO TX-*nnn* MAC...)**.

Mac OS X USB Connection

- Connect the Robo TX to the Mac with the supplied USB Cable.
- If a dialog box opens asking to configure the new connection, just close it.
- Launch Flowol 4 and choose the Robo TX. In the interface options, choose the connection that is labeled **/dev/tty.usbmodem*nnn***.

The Bluetooth connection is not currently supported on Mac OS X.

Linux USB Connection

- Connect the Robo TX to the computer with the supplied USB Cable.
- The device driver is included in the Linux kernel, so simply choose the appropriate serial port.

Linux Bluetooth Connection

- To use the ROBO TX with a Bluetooth connection, add the following to the **/etc/bluetooth/rfcomm.conf** file:

```
rfcomm0 {
  bind yes;
  device xx:xx:xx:xx:xx:xx;
  channel 1;
  comment "Robo TX"
}
```

Where **xx:xx:xx:xx:xx:xx** is the Bluetooth address of the ROBO TX.

- Then choose the appropriate serial device from within Flowol 4.

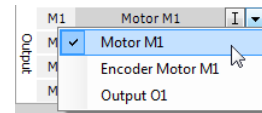
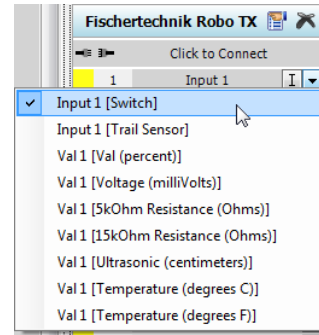
Inputs

There are 8 general purpose inputs on the Robo TX (**I1 – I8**) which can connect to either digital inputs (switches) or analog inputs. Hover the mouse pointer over the Status Panel and then click on the dropdown arrow ▾ to configure the input.

A switch can be connected to the counting inputs (**C1-C4**) if they are not being used with an **Encoder Motor** (see below).

Motors and Digital Outputs

There are 4 outputs on the Robo TX (**M1-M4**). By default they are configured for the connection of a regular motor. Hover over the Status Panel and click on the dropdown arrow ▾ to reconfigure for either an **Encoder Motor** or two digital outputs.

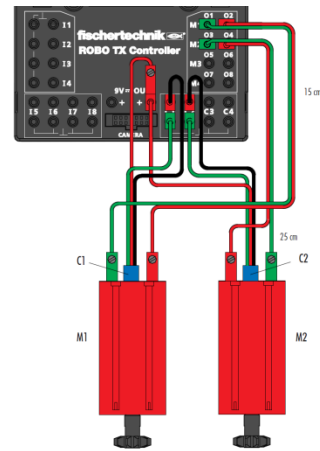


Configuring Encoder Motors

An encoder motor is a smart motor. The encoder motor contains a mechanism for counting the revolutions of the motor’s shaft. There is also a gearbox included inside the encoder motor with a transmission ratio of 25:1 to slow down the rotation of the output shaft. A count of 75 pulses is generated per rotation of the output shaft.

Connect the encoder motor to both the motor output (**M1**) and the corresponding counter input (**C1**) as shown in the wiring diagram to the right.

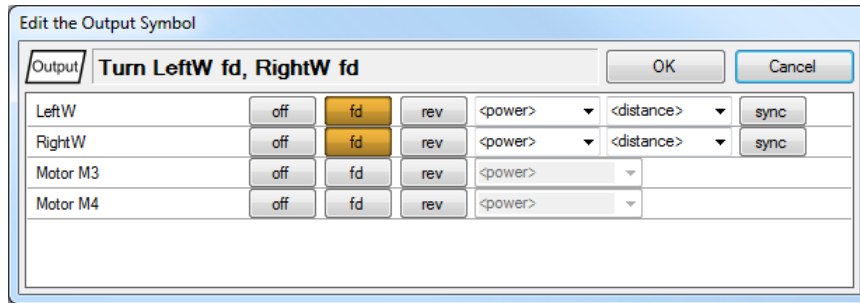
When an encoder motor is used, the counter input cannot be used with a switch, and so the separate input disappears from the Status Panel.



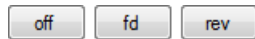
Controlling Encoder Motors in your Flowchart

Encoder motors can be used anywhere that precise motor control is needed. A very common use is to control the two wheels of a floor robot. Using one motor on the left wheel and another on the right enables the robot to move forwards, backwards and make turns. Consider the floor robot and label the encoder motors in **M1** and **M2 LeftW** and **RightW** for the left and right wheels respectively.

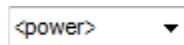
Then place an output symbol onto the flowchart and open its prompt box:



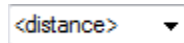
Encoder motors present a few new options:



These have same effect as a regular motor. Turn the motor off, forward or reverse.

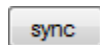


(Optional) Specify the speed of the motor as a percentage. Use either a number, or a variable.



(Optional) Enter a distance for the motor to turn. The motor will run until it has turned this distance at which point it will break. The control system also calculates the breaking distance of the motor in order to automatically apply the break early enough to stop at the correct distance. A distance value of 75 represents a rotation of the motor's output shaft.

Even if a distance value is set, the running flowchart will progress immediately to the next symbol. This enables the flowchart to continue reacting to the environment and can stop the motor short if necessary.



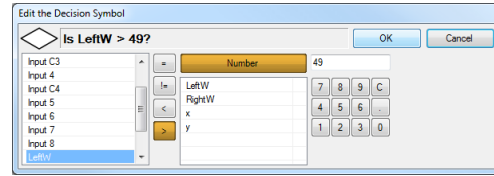
If two or more motors are in use in this output symbol, you may select **sync** on two or more of them to synchronize their rotation. When synchronized, these motors will make the same number of turns in the same time, if one motor encounters resistance and turns slower, the system will automatically slow down the other motor.

Measuring Encoder Motors

The encoder motors track their distance in terms of a count. This count is displayed in the Status Panel.

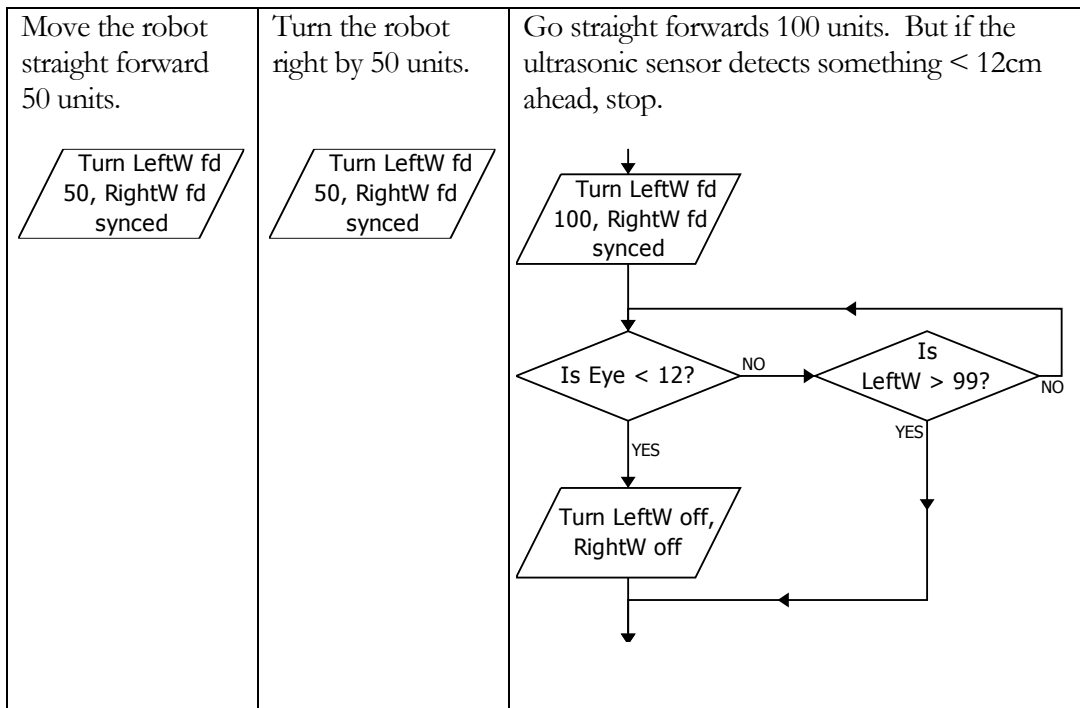
M1	LeftW	50
M2	RightW	50

The count is available to the flowchart as a read-only variable, and can be used in decision and assignment symbols.



Encoder Motor Examples with a Floor Robot

Consider again the floor robot, with two encoder motors named **LeftW** and **RightW** attached to the left and right wheels respectively. Also attach the ultrasound distance sensor to **I1**, calibrate it for **Ultrasonic (cm)** and rename it **Eye**. For example, see the **Basic Model** in the **ROBO TX Training Lab**.

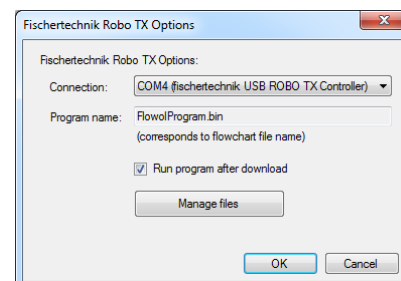


Downloading

Click the Download button to download your flowchart to the RoboTX controller. The RoboTX can run all of the Flowol flowchart symbols.

The flowchart is compiled and downloaded to as a .bin file to the **flash memory** on the RoboTX. Flash memory is not erased when the RoboTX is turned off. Flowol names your program using the same name as your saved flowchart.

By default, the RoboTX will run the program after download. You can modify this behavior in the Robo TX options dialog.



FLOWOL 4 TUTORIAL

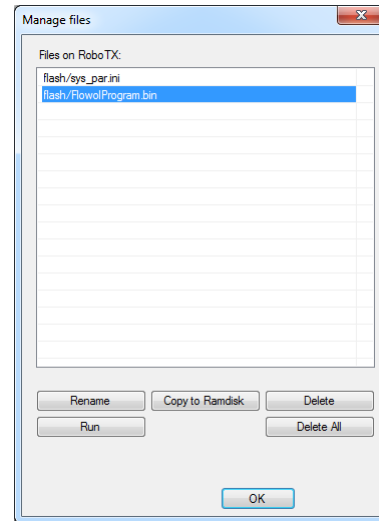
Flowchart programs run very quickly when downloaded to the RoboTX (1000 symbols can be executed every second). Therefore, downloading the program is very useful when controlling autonomous robots as your program can respond to changes in the robot's environment much more quickly than when run on the computer over a Bluetooth connection.

Managing Files Downloaded to the RoboTX

Within the RoboTX Options dialog, click on the Manage Files button to list, rename, run and delete the programs you have downloaded.

Making changes to files in the flash memory can sometimes take a little time.

Any programs stored on the Ramdisk will be erased when you switch off the RoboTX controller.





Fischertechnik Robo LT

<http://www.flowol.com/flowol4/Interfaces/FischertechnikRoboLT.aspx>

The Robo LT is the complete beginner's robotics package for kids age 8 and above using the fischertechnik digital sensors and outputs. The Robo LT interface does not support encoder motors.

Windows USB Connection

- Connect the Robo LT to the computer with the supplied USB Cable.
- Connect the Robo LT to a power source, and if Windows fails to locate the device driver, choose the option to Install from a specific location. Search this location (if Flowol 4 was installed to the default directory): **C:\Program Files\Keep I.T. Easy\Flowol 4\Interfaces\RoboDeviceDriver**
- Launch Flowol 4 and choose the Robo LT.

Mac OS X/Linux USB Connection

No device driver is required. Launch Flowol 4 and choose the Robo LT.

The ftlib.dll file (Windows) and the Robo USB Device Driver (Windows) are Copyright Knobloch GmbH (www.knobloch-gmbh.de) and are included in Flowol 4 with permission.



Fischertechnik Robo

<http://www.flowol.com/flowol4/Interfaces/FischertechnikRobo.aspx>

The Robo is the predecessor to the Robo TX, and with 1mm sockets can connect to the range of fischertechnik sensors and outputs. The Robo interface does not support encoder motors.

Windows USB Connection

- Connect the Robo to the computer with the supplied USB Cable.
- Connect the Robo to a power source, and if Windows fails to locate the device driver, choose the option to Install from a specific location. Search this location: **C:\Program Files\Keep I.T. Easy\Flowol 4\Interfaces\RoboDeviceDriver** (if Flowol 4 was installed to the default directory).
- Launch Flowol 4 and choose the Robo. The connection should default to USB.

Mac OS X/Linux USB Connection

No device driver is required. Launch Flowol 4 and choose the Robo. The connection should default to USB.

Serial Connection

- Connect the Robo to a serial port on the computer.
- Launch Flowol 4 and choose the Robo.

- Open the Robo Options and choose the correct serial port from the Connections dropdown.


The fischertechnik Robo interface is not supported on 64-bit versions of Windows.

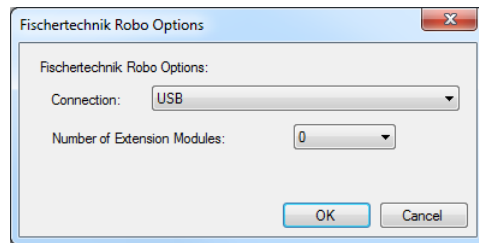
Analog Inputs

The **A1** and **A2** analog inputs map to **Val 1** and **Val 2** respectively. The analog inputs **AX** and **AY** map to **Val 3** and **Val 4** respectively.

Extension Modules

The Robo interface supports up to three extension modules. These are connected to the Robo interface with a short ribbon cable.

After connecting the extension module(s), open the Fischertechnik Robo Options dialog  to adjust the **Number of Extension Modules** setting.



The fplib.dll file (Windows) and the Robo USB Device Driver (Windows) are Copyright Knobloch GmbH (www.knobloch-gmbh.de) and are included in Flowol 4 with permission.



Fischertechnik Intelligent Interface

<http://www.flowol.com/flowol4/Interfaces/FischertechnikIntelligent.aspx>

The older Intelligent Interface connects via the Serial port. It provides 8 digital and 2 analog inputs, and 4 motor outputs.



Control Station

<http://www.flowol.com/flowol4/Interfaces/ControlStation.aspx>

The Control Station is a general control box which uses 4mm sockets for connection of inputs and outputs. It has 6 digital and 2 analog inputs and 4 digital outputs and 2 motors with power control.

Windows: Connect either to the USB port or connect with to a serial port (see <http://www.flowol.com/flowol4/Interfaces/ControlStation.aspx> to download and install the USB device driver).

Mac OS X/Linux: Connect to a serial port via a USB to Serial converter.



Smart Box

<http://www.flowol.com/flowol4/Interfaces/SmartBox.aspx>

The Smart Box is a general control box which uses 4mm sockets for connection of inputs and outputs. It has 8 digital and 4 analog inputs and 8 digital outputs and 4 motors with power control.


Windows: Connect to either the USB port (if necessary, install the 'Smart Box USB Drivers' that came on the CD ROM with your Smart Box) or connect with a Serial Port.

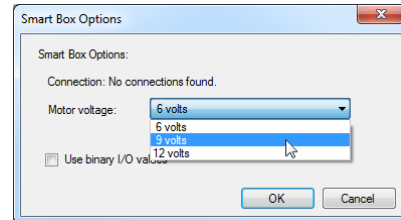
Mac OS X/Linux: Connect to a serial port via a USB to Serial converter.

Analog Sensors

The Smart Box has a wide range of available analog sensors. To calibrate the readings, hover the mouse pointer over the Status Panel, and click on the dropdown arrow ▾ to choose the correct sensor from the list.

Output Voltage

By default, the Smart Box has an output voltage of 6 volts. Click on the  and use the dropdown in the Smart Box Option dialog to adjust this to either 9 or 12 volts if necessary.



Caution, setting the Smart Box output voltage to a value higher than the rating on your outputs (motors and lights) can damage your output devices.

Therefore a password is required to make this change. The password is printed in the Smart Box 'User Information' leaflet that came with your Smart Box.



Contact Controller/Contact Controller Plus

<http://www.flowol.com/flowol4/Interfaces/ContactController.aspx>

<http://www.flowol.com/flowol4/Interfaces/ContactControllerPlus.aspx>



The Contact Controller and Contact Controller Plus are robust serial interfaces. They use 4mm sockets for inputs and outputs.

The Contact Controller has 4 digital and 4 analog inputs, and 4 digital and 2 motor outputs with power control.

The Contact Controller Plus has 8 digital and 4 analog inputs, and 8 digital and 4 motor outputs with power control.



Commotion Coco (earlier metal version)

<http://www.flowol.com/flowol4/Interfaces/CommotionCoCo.aspx>

The earlier version of the Commotion Coco interface has a USB connection (no device driver required for Windows or Mac OS X). It uses 4mm sockets for inputs and outputs.

It has 4 digital and 6 analog inputs. Two of the analog inputs are the internal light and sound sensors. It has 6 digital and 2 motor outputs with power control.



Deltronics/Commotion Junior Serial

<http://www.flowol.com/flowol4/Interfaces/DeltronicsJuniorSerial.aspx>

The Junior Serial interface connects to the computer via the Serial port. It uses 4mm sockets for inputs and outputs.

It has 4 digital and 4 analog inputs, and 4 digital and 2 motor outputs with power control.



Deltronics/Commotion Serial Interface

<http://www.flowol.com/flowol4/Interfaces/DeltronicsSerial.aspx>

The Serial Interface connects to the computer via the Serial Port. It uses 4mm sockets for inputs and outputs.

It has 8 digital and 4 analog inputs. It has 8 digital outputs. The outputs are dual use as 4 bidirectional motors. This means that for every 2 digital outputs you can use either 2 lights OR a motor.



Deltronics Serial Plus

<http://www.flowol.com/flowol4/Interfaces/DeltronicsSerialPlus.aspx>

The Deltronics Serial Plus connects to the computer via the Serial Port. It uses 4mm sockets for inputs and outputs.

It has 8 digital and 4 analog inputs, and 8 digital and 4 motor outputs with power control.



Deltronics Digital Serial Adapter

<http://www.flowol.com/flowol4/Interfaces/DeltronicsSerialAdapter.aspx>

The Digital Serial Adapter enables an old Deltronics Buffer Box, which was designed for connection to a BBC Microcomputer, to be connected via a Serial Port to a PC or Mac.

The adapter supports 8 digital inputs, and 8 digital outputs. The outputs are dual use as 4 bidirectional motors. This means that for every 2 digital outputs you can use either 2 lights OR a motor.



LEGO Dacta Control Lab (LEGO Interface B)

<http://www.flowol.com/flowol4/Interfaces/LegoDactaControlLab.aspx>

The old LEGO Dacta Control Lab connects to a Windows PC or Apple Mac via a serial port. It uses the older 9v LEGO motors and sensors which are connected with the square brick plugs.

Flowol will operate lights connected to the 8 outputs using digital **Outputs 1-8**. Alternatively, connect motors to outputs A-D and operate with **Motors A-D**.

Flowol will respond to switches connected to **Inputs 1-8**. Or connect analog sensors to inputs 5-8 and use with **Val 1-4**.



Arduino

<http://www.flowol.com/flowol4/Interfaces/Arduino.aspx>

The Arduino prototyping hardware platform consists of several I/O boards (based on the ATmega line of microcontrollers). The I/O sockets on the board are in a fixed position therefore allowing it to be extended with a variety of custom daughter-boards (called shields).

Flowol 4 supports the Arduino Uno, Arduino Leonardo and the Arduino Duemilanove (with either the ATmega328p or 168 microcontrollers). There are also many Arduino-compatible boards made by other companies (e.g. Freeduino and DFRobot boards). Flowol may support many compatible boards with a 16MHz resonator.

See the Flowol end-user license agreement for details of use of the Arduino library components.

Flowol 4 supports connection to the Arduino via USB cable. [Note that the USB cable can be used to provide power to the board, although this will not drive motors].

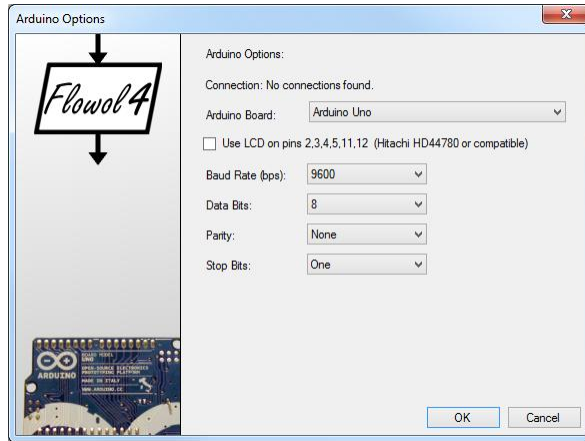
Windows: If when connecting the Arduino Uno/Leonardo, the driver has not already been installed, Windows will prompt you that the driver cannot be found. Direct Windows to install the USB device driver from **C:\Program Files\Keep I.T. Easy\Flowol 4\Interfaces\ArduinoDeviceDriver**. The FTDI driver for the Arduino Duemilanove is automatically installed with the Flowol installation.

Mac OS X: No driver should be needed for the Arduino Uno/Leonardo. If using the Arduino Duemilanove, download and install the FTDI device driver from the Arduino website.

Linux: No driver required. The port should be listed on the dropdown.

Configuring the Arduino

Once you have selected the Arduino option from the list of Interfaces, the **Arduino Options** dialog box will open.



Choose the appropriate **Connection** and **Arduino Board** from the dropdown. The board choice will be remembered for the next time Flowol is used.

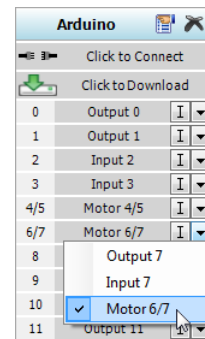
Flowol supports both online (connected) operation as well as downloading the flowchart for remote operation.

Pin Configuration

With the Uno or Duemilanove, pins 0 and 1 cannot be used when Flowol connects to the interface in online mode as they are used for the serial communication with the computer (RX/TX).

Digital Inputs/Outputs

By default, all of the digital pins are configured as outputs and all of the analog pins are configured as percentage readings. Use the dropdown arrows in the Status Panel to reconfigure the pins.



Motors

Pins 4/5 and 6/7 can be configured to support bi-directional motors. This supports the DFRobot Motor Shield and the DFRobot RoMeo board. The motors support PWM power control and use the following pins:

Pin	Use
4	Motor 4/5 Direction Control
5	Motor 4/5 PWM Control
6	Motor 6/7 PWM Control
7	Motor 6/7 Direction Control

Servo Motors

Any of the pins can drive servo motors. Choose **Servo** from the dropdown list and then use the Output symbol to set the position of the servo (0° to 180°).

Note that if any servos are used, then PWM outputs cannot be used on pins 9 or 10.

PWM Outputs

PWM (Pulse-width modulation) outputs can be selected from the dropdown on pins 3, 5, 6, 9, 10 and 11. Then use the Output symbol to set the power of the output to a value (0 to 255).

Note that the motors use PWM outputs to vary the speed of the connected motors.

Digital Encoders

Digital encoders are sensors which count holes or color changes on a disk attached to a rotating shaft or axel. Connect digital encoders to either pin 2 or pin 3, then choose Encoder from the dropdown. Flowol will then automatically count the on/off cycle of the input and make the value available as a variable. The changes trigger a hardware interrupt in the Arduino microcontroller so the change in input state is never missed.

Use the encoder variable in decision symbols to see if your robot has moved the required distance. The variable can also be reset (**Let Encoder 2 = 0**) in an assignment symbol.

HC-SR04 Ultrasonic Distance Sensor

Up to two HC-SR04 (or compatible) ultrasonic distance sensors can be used. These sensors can measure the distance to a wall or other object by sending a short series of ultrasonic sound pulses and listening for the echo. Flowol will time the echo and will compute the distance in centimeters.

HC-SR04 sensors have 4 connections. Connect the Gnd to GND on the Arduino and connect the Vcc to 5V. Then connect the Echo to pin 2 (or pin 3) and the Trig to pin 9 (or pin 10). Flowol continually triggers the sound sensor to get a current reading of the distance.

Parallax PING))) Distance Sensor

Up to two Parallax PING))) distance sensors can be used on either pins 2 or 3. These sensors can measure the distance to a wall or other object by sending a short series of ultrasonic sound pulses and listening for the echo. Flowol will time the echo and will compute the distance in centimeters.

The Parallax PING))) sensors have the advantage over other ultrasonic distance sensors that they only need to connect to a single pin (either pin 2 or pin 3). Flowol continually triggers the sound sensor to get a current reading of the distance.

Analog Inputs

The Analog pins can be configured for one of the following:

- Val (percent) a linear scaling of the analog reading to 0-100%
- Val (Raw 10-bit) the raw 10-bit (0-1023) analog reading.
- Temperature (LM35) (degrees C) calibrates a LM35 linear temperature sensor (e.g. the DFRobot DFR0023).

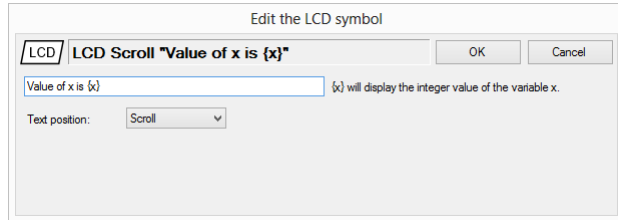


LCD Displays

A Hitachi HD44780 or compatible 16x2 character display can be connected and controller by Flowol. Only one connection is supported:

GND to 0V, VCC to 5V	D4 to pin 5
RS to pin 12	D5 to pin 4
E to pin 11	D6 to pin 3
	D7 to pin 2

To use the LCD, check the box in the Arduino interface settings dialog box. Then a new LCD symbol will be added to Flowol which sends a line of text to the LCD.



The text can be displayed on either the Top or Bottom lines, or choose Scroll to write the text to the bottom line (and scroll the old text up).

The LCD is especially useful for displaying variable or analog value information. Enter the name of a variable (or analog value) in braces in the text string to display the corresponding value on the LCD. For example: “Value of x is {x}”.

Serial

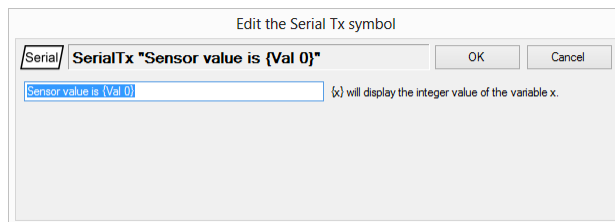
Serial Communication

When you have downloaded your program to the Arduino and disconnected from the computer, pins 0 and 1 become available either as general purpose Input/Output connections or they can communicate with a serial device.

If you choose SerialRx from the dropdown on Pin1, then the downloaded flowchart will setup pins 0 and 1 as a serial port using the configuration information chosen in the Arduino interface properties dialog (default 9600 baud, 8 data bits, no parity, 1 stop bit).

Serial communication is useful for connecting to another control device or for receiving commands from a serial terminal. Extra hardware can be added to send the serial protocol over Bluetooth to a terminal application on a smart phone (e.g. Android). In this way, the terminal could send command bytes to the flowchart running on the Arduino. And the Arduino can send textual messages back to the terminal.

Send text using the Serial output symbol (available once SerialRx is selected on Pin 0). Enter the name of a variable (or analog value) in braces in the text string to send the corresponding value. For example “Sensor value is {Val 0}”.



Read bytes send as integer values using the **SerialRx** variable. This variable will automatically be filled with the next byte received over the serial connection. If there are no more bytes available, the variable will be unchanged. The SerialRx variable can also be reset (**Let SerialRx = 0**) in an assignment symbol which is useful after a command has been processed.

Programming Features

The Arduino supports most of Flowol’s programming features with the following notes:

- All variables are treated as 16-bit signed values (-32768 to 32767).
- Parameters cannot be passed to subroutines.



PICAXE

<http://www.flowol.com/flowol4/Interfaces/Picaxe.aspx>

The PICAXE microcontrollers from Revolution Education are a versatile and well supported set of programmable chips. Flowol 4 supports the following PICAXE chips: 08, 08M, 08M2/08M2+, 08M2LE, 14M, 14M2, 18, 18A, 18M, 18M2, 18M2+, 18X, 20M, 20M2, 20X2, 28A, 28X, 28X1, 28X2, 40X, 40X1 and 40X2.

The PICAXE Plug-in files installed in the Compilers directory are Copyright Revolution Education Limited and are included with Flowol 4 under license.

Revolution Education produces a wide range of project boards for the PICAXE chips. They provide a 3.5mm jack socket for connection to the computer via either the AXE027 USB cable or AXE026 Serial cable.

Windows: If necessary, install the USB device driver from **C:\Program Files\Keep I.T. EasyFlowol 4\Interfaces\Picaxe027Driver**.

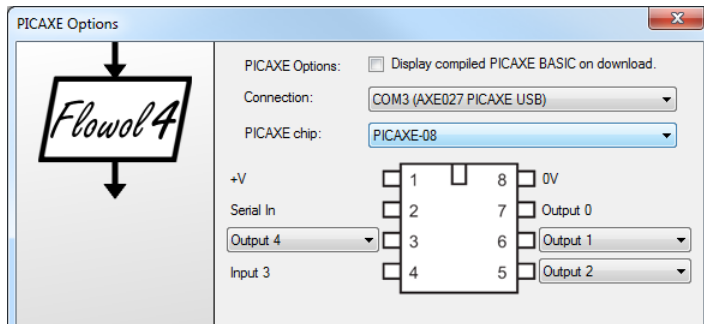
Mac OS X: If necessary, install the USB device driver from the PICAXE website.

Linux: No driver is necessary. The port should be listed on the dropdown.

Configuring the PICAXE chip

Once you have selected the PICAXE option from the list of Interfaces, the **PICAXE Options** dialog box will open.

Check **Display compiled PICAXE BASIC on download** in order to view the compiled BASIC code that's sent to the PICAXE chip.

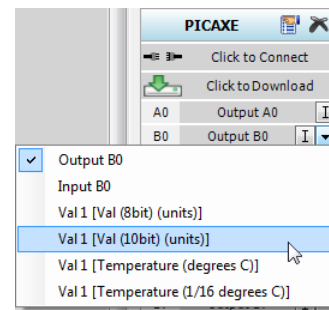


Choose a connection from the connection dropdown. Then choose the appropriate PICAXE chip from the dropdown. The chip diagram below will show the pin out for the chip chosen. If using the Microbot, choose the PICAXE-MICROBOT option.


On many PICAXE chips there are pins which can be configured as either a digital or analog input, or a digital output. Use the dropdowns here to configure those pins. Click on OK.

Return to the options dialog by clicking on the  button in the PICAXE section of the Status Panel.

It is also possible to configure the pins (and calibrate analog sensors) using the dropdowns in the Status Panel.



Connected Online Operation

The 08M2/08M2+, 08M2LE, 14M, 14M2, 18M, 18M2, 18M2+, 20M, 20M2, 20X2, 28X1, 28X2, 40X1 and 40X2 chips support online operation. Click on the connect button  to connect in online mode. Note that this will erase and replace the last program that has been downloaded to the PICAXE chip.

If there is nothing connected to an input pin on the chip then its state may fluctuate.

Downloading

Click the Download button  to compile and download the flowchart to the PICAXE.

Consider the following points when creating your flowchart with PICAXE in mind:

- The 08M2/08M2+, 14M2, 18M2 and 20M2 chips supports up to 4 parallel flowchart threads (4 Start symbols placed on the page). The 18M2+ chip supports up to 8 parallel flowchart threads. They also support the Advanced Thread Control symbol. All other PICAXE chips only support a single threaded flowchart (one Start symbol).
- PICAXE interrupts are not currently supported by Flowol.
- All variables are compiled to the PICAXE 16-bit (word) values and can therefore store integer values between 0 and 65535.
- When using a pin for an analog reading, use the dropdown in the Status Panel to configure for either an 8bit or 10bit analog reading. Or configure for using a DS18B20 digital temperature sensor.
- Flowol will provide optional motor outputs (pairing two digital outputs to achieve a bi-directional motor) for those chips which have a bank of non-configurable outputs.



Brainy-USB and Brainy-Motor Kit

<http://www.flowol.com/flowol4/Interfaces/BrainyUSB.aspx>



<http://www.flowol.com/flowol4/Interfaces/BrainyMotor.aspx>

The Brainy-USB and Brainy-Motor kits from Kre8 use the FlowIC microcontroller to provide a cost-effective, yet versatile microcontroller solution.

The Brainy-USB board has 1 digital input and 3 digital outputs (Outputs 1 and 2 are low-power outputs suitable for LEDs, Output 3 is a medium power output suitable for a buzzer or small motor).

The Brainy-Motor board has 4 digital inputs and 2 digital outputs (Output 1 is a low power LED output; Output 2 is a medium power output suitable for a buzzer of small motor). It also has two bi-directional motor outputs, Motor A and Motor B.

FLOWOL 4 TUTORIAL

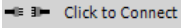
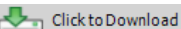
The Brainy circuit board is connected to the computer via the Brainy USB Cable. Connect this to the 3.5mm jack socket on the board and to a USB socket on your computer.

Windows: The USB Device Driver is automatically installed with Flowol 4.

Mac OS X/Linux: No device driver is required.

Downloading

Flowol 4 can operate the Brainy boards in connected mode (also called In-Circuit Emulation) where the inputs and outputs are displayed in the Status Panel. Flowol 4 can also download the flowchart to the Brainy boards to be run remotely:

1. Connect the cable to the interface and click on the Connect button  to ensure that Flowol is connected to the interface.
2. Run your flowchart while the interface is connected to check that everything works as expected with your model. Sometimes delay values need to be refined.
3. Click the Download button  to compile and download the flowchart to the FlowGo.
4. When the download has completed, Flowol will disconnect from the Brainy board and instruct the microcontroller to run the downloaded program. To reconnect to the board click on the Connect button, this will halt any program running on the microcontroller and allow Flowol to either run connected, or download a modified flowchart.

There are some limitations to what a downloaded program can do:

- A maximum of two variables can be used in the flowchart. The variables are stored as single byte values and so they can only be integers between 0 and 255.
- Variable assignments are limited to the forms **Let x = constant** or **Let x = x +/- constant/x/y**.
- A maximum of 3 parallel threads (separate Start symbols) can be used.
- A maximum of 16 defined subroutines.
- A maximum of 16 distinct delay constants
- About 150 symbols can be used.
- Calls to subroutines with repeat values can only be nested to a depth of 2.

If exceeded, these limitations will be pointed out when you try to download your flowchart and you will have to adjust your flowchart. However if you choose the **Limit features to those that will work with a flowchart downloaded to the interface** option in the interface options, then Flowol will enforce many of these limitations in the symbol prompt boxes, making it easier to create a flowchart that can be downloaded.



Intelligent FirmwerX FlowIC Kit

<http://www.flowol.com/flowol4/Interfaces/FlowIC.aspx>

Intelligent FirmwerX produce two boards, one for the 8-pin FlowIC microcontroller and another for the 14-pin FlowIC microcontroller.

The 8-pin board supports 1 digital input and 3 digital outputs. The 14-pin board supports 4 digital inputs and 6 digital outputs. An optional motor PCB will use 4 of the digital outputs to drive 2 motors.

The Intelligent FirmwerX boards are connected to the computer with a USB download cable. Connect to the 6-pin right connector on the board and to a USB port on the computer.

Windows: The USB Device Driver is automatically installed with Flowol4.

Mac OS X/Linux: No device driver is required.

Downloading

Flowol 4 can operate the Intelligent FirmwerX boards in connected mode (also called In-Circuit Emulation) where the inputs and outputs are displayed in the Status Panel. Flowol 4 can also download the flowchart to the FlowIC microcontroller to be run remotely

The limitations for remote operation are the same as for the Brainy boards (see above).



Solo18/Solo28 Microcontroller Kits

<http://www.flowol.com/flowol4/Interfaces/Solo18.aspx>

<http://www.flowol.com/flowol4/Interfaces/Solo28.aspx>



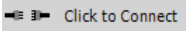
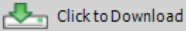
The **Solo** (stand-alone) PIC Programming System from Data Harvest can be programmed with Flowol 4. Programs can be run in a connected mode, or downloaded and run remotely. The Solo 18 has 4 digital inputs and 7 outputs (or with 2 motors). The Solo 28 has 8 digital and 2 analog inputs, 8 outputs (or with 4 motors).

Windows: the device driver (32-bit or 64-bit) for FlowGo's USB connection is installed automatically with Flowol 4. Just plug the USB cable to the FlowGo and to the computer and wait a moment for the driver to be automatically loaded.

Mac OS X/Linux: no device driver is required.

Downloading

Flowol 4 can also download the flowchart to the Solo to be run remotely:

1. Connect the cable to the interface and click on the Connect button  to ensure that Flowol is connected to the interface.
2. Run your flowchart while the interface is connected to check that everything works as expected with your model. Sometimes the threshold in an analog sensor condition or a delay value needs to be refined.
3. Click the Download button  to compile and download the flowchart to the Solo.

There are some limitations to what a downloaded program can do:

- A maximum of two variables can be used in the flowchart. The variables are stored as single byte values and so they can only be integers between 0 and 255.

- Variable assignments are limited to the forms **Let x = constant** or **Let x = x +/- constant/x/y**.
- Motor speeds can only be set with a constant value (e.g. **50%**, not **x%**).
- A maximum of 4 parallel threads (separate Start symbols) can be used.
- A maximum of 16 defined subroutines.
- A maximum of 16 distinct delay constants
- About 100 symbols can be used.
- Calls to subroutines with repeat values can only be nested to a depth of 2.
- The Random function is not supported.

If exceeded, these limitations will be pointed out when you try to download your flowchart and you will have to adjust your flowchart. However if you choose the **Limit features to those that will work with a flowchart downloaded to the interface** option in the interface options, then Flowol will enforce many of these limitations in the symbol prompt boxes, making it easier to create a flowchart that can be downloaded.



Display on Solo LCD

Data Harvest also produces an add-on LCD for the Solo. This connects to the Solo PCB. Information can be displayed on the LCD by using the Display symbol which is now available on the left toolbar. The LCD will only operate when the flowchart has been downloaded to the Solo and it is running remotely.

The Display symbol instructs the Solo to display one of the following:

- A static text string.
- A number, either in decimal or hexadecimal format.
- The current value of one of the variables.



Numbered Interface

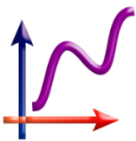
You will not need to use the **Numbered Interface** in Flowol 4. It is listed because it is used when loading flowchart (.flo) files saved by Flowol 2 or Flowol 3.

It is the equivalent of the **No Interface** option present in Flowol 3.



Chapter 5: Advanced Features

Flowol 4 has several advanced features.

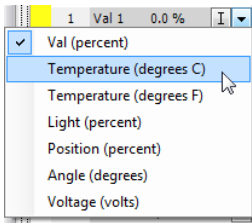


Graphs

Any real-world control system will need to be monitored. Monitoring is essential to verify correct operation and check the efficiency of the system.

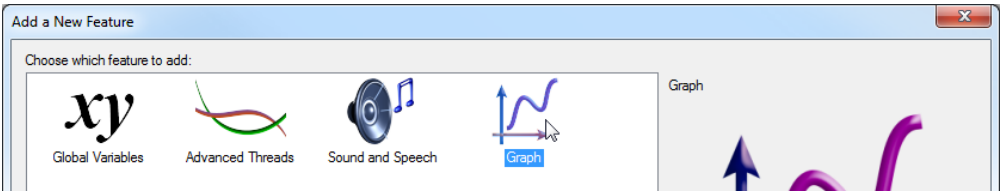
Monitoring is best done with a real model connected to Flowol via a hardware interface. Please review the last chapter to select and configure your hardware interface.

Connect appropriate analog sensors to your model and calibrate the reading to the appropriate units by selecting the right sensor from the dropdown in the Status Panel. For example use a temperature sensor to measure the temperature of the air inside a greenhouse.

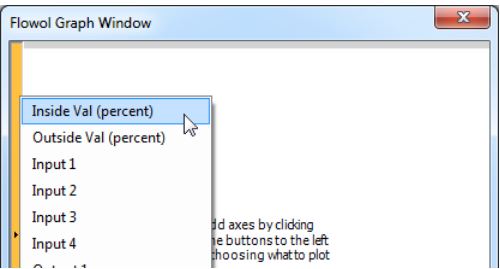


Opening and adding Axes to a Graph Window

1. Click on the More... button  at the top-right of the window.
2. Choose Graph from the **Add a New Feature** dialog:



3. When the graph window opens, click on the left edge (it turns orange) to open the menu of available axes. In the greenhouse example, I have labeled Val1 to Inside and Val2 to Outside.
4. Choose an analog sensor from the menu to create an axis.



FLOWOL 4 TUTORIAL

The graph window will now look like this:

Click the top button to add a new axis above the current one.

Click the middle button to add a new set of data on the current axis.

Click the bottom button to add a new axis below the current one.

Click the dropdown arrow to open a submenu about this axis.

Properties... will open a properties dialog relating to this axis. Change the grid, color and y-axis limits.

Remove will remove the axis from the window.



To start logging click the Run button at the bottom-left of the Flowol screen. By default the axis will log 2 minutes of data. After that the axis duration will double to 4 minutes and Flowol will discard every other logged data reading so the logging interval will also double.

Hiding and Showing the Graph Window

Click on the close icon of the Graph window to hide it. A graph icon will then appear in the Graph area of the Status Panel. Click on this icon to show the graph window again.


The screenshot shows the Flowol Graph Window with a red data line plotted on a grid. The y-axis ranges from -10.0 to 110.0, and the x-axis ranges from 0 to 2 minutes. The Status Panel on the right shows a list of outputs and motors, and a Graph icon with a red arrow pointing to it.

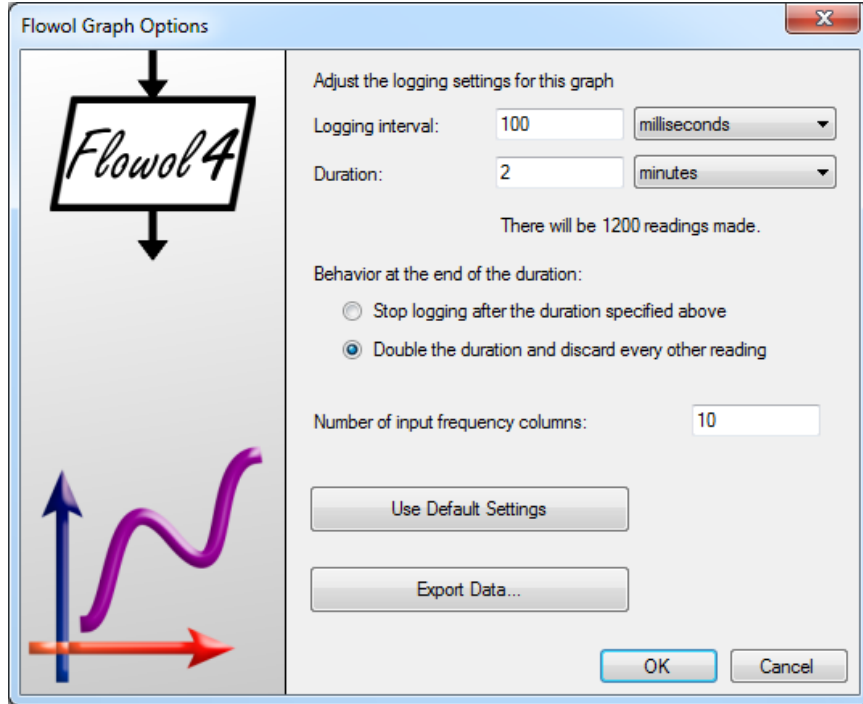
To remove the graph window entirely, click on the black Delete icon:



Note that this operation can be undone with the Undo button.

Graph Options

Click on the options button  in the Graph area of the Status Bar to view and adjust the graph options.



By default logging occurs at an interval of 100ms for a duration of 2 minutes, after which the duration and interval are doubled and every other reading is discarded.

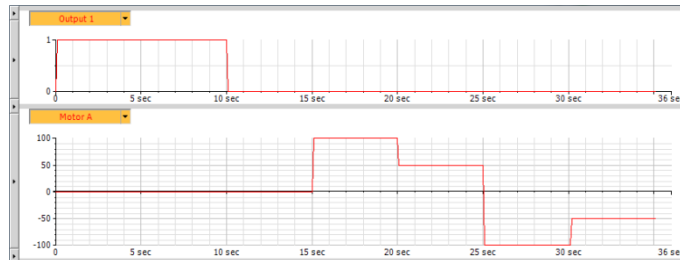
Adjust the duration and interval to suit your project. Flowol 4 can log a maximum of 20,000 readings. Click on **Use Default Settings** to reset the values back to the defaults.

To export the logged data as a Comma Separated Value (.csv) file, click on the **Export Data...** button. CSV files can be loaded into a database or spreadsheet (e.g. Microsoft Excel) for further analysis.

The number of input frequency columns defaults to 10. See below for details on how digital inputs are logged and displayed.

Digital Output and Motor axes

Digital Outputs and Motors can also be displayed on the graph window. Here Output 1 and Motor A are displayed on separate axes:



FLOWOL 4 TUTORIAL

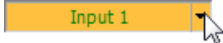
A digital output has a value of 1 when on and 0 when off. The motor's value is the speed of the motor, with negative values when the motor is going in reverse.

Consider again the model greenhouse. Two temperature sensors could be used to log the inside and outside temperatures. A motor could be used to control a window, and a flowchart constructed to open the window to ventilate the greenhouse if it gets too hot. To measure the benefit of the window system, log both temperatures on one axis and the window motor on another.

Digital Input axis

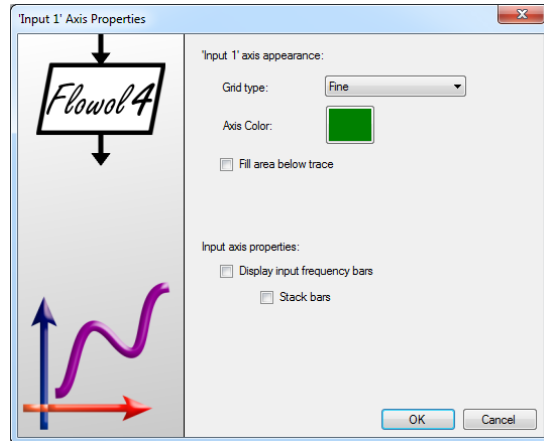
Digital Inputs can also be displayed on the graph window. Like a digital output, an input has a value of 1 when on and 0 when off.

Alternatively, digital inputs can be displayed on a frequency chart. To use a frequency display:

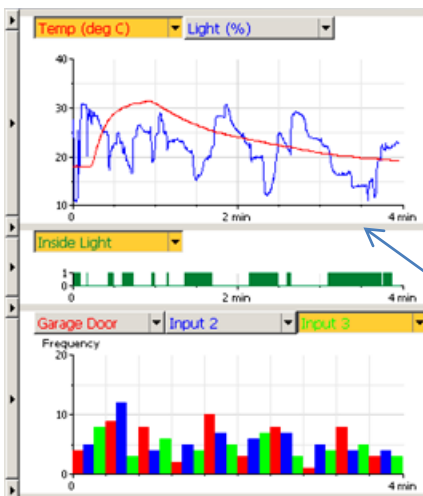
1. Add the input axis to the graph using the button at the left edge of the graph window.
2. Click on the dropdown arrow on the Input label at the top: 
3. Choose **Properties**, and then in the properties dialog, check the **Display input frequency bars** option. Then Click OK.

If more than one input frequency axis is on the same graph, then the frequency bars can be stacked.

Consider the Ferris Wheel, or other fairground ride. Log the frequency of an input used to start the ride over the period of a day. This would show when the ride was most popular.



Multiple Axes Overview



The number of sets of axes and the data displayed on those axes is your choice.

When multiple values are displayed on the same axis, click on the title (to turn it orange) to select that trace's y-axis information to be displayed on the graph.

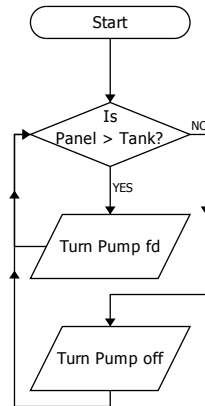
Click and drag the horizontal grey gutters up/down to resize the axes heights.

If required, additional graph windows can be added. Simply open the **New Features** dialog by clicking **More...** and chose again **Graph**.

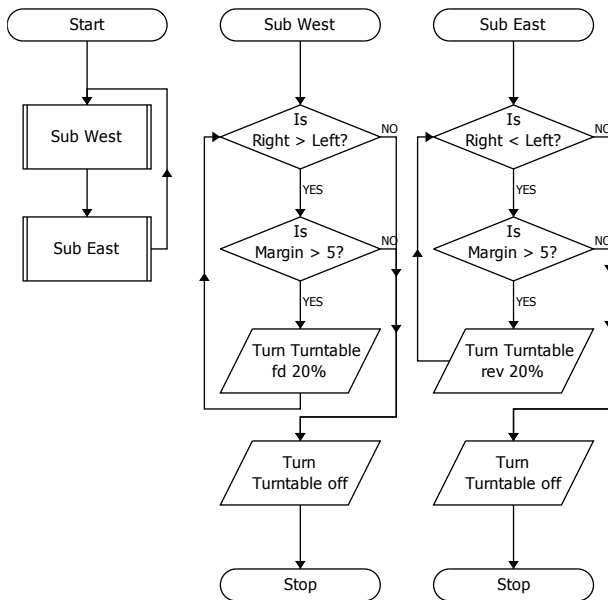
Solar Water Heating Panel

In this example, a solar panel will heat water. The solar panel is a large, square glass panel behind which is a black tube of water snaking across the surface. The solar energy from the sun heats the water in the tube.

The tube is connected in a loop with a pump and water tank. There are two temperature sensors, one in the water tank and the other in the panel. The following flowchart controls the pump so that the water is circulated only when the panel temperature is hotter than the tank.



The Sun Seeker



To make the Solar Panel more efficient we would like it to turn to face the sun. This can be done by placing the panel on a turntable controlled by a motor. Two light sensors separated by a piece of cardboard are attached to the front of the panel so a shadow is cast over one of the sensors when the sun is not directly overhead.

The **Margin function** calculates the positive difference between the two values used at the comparison in the preceding decision symbol.

Therefore the flowchart to the left only turns the turntable when the

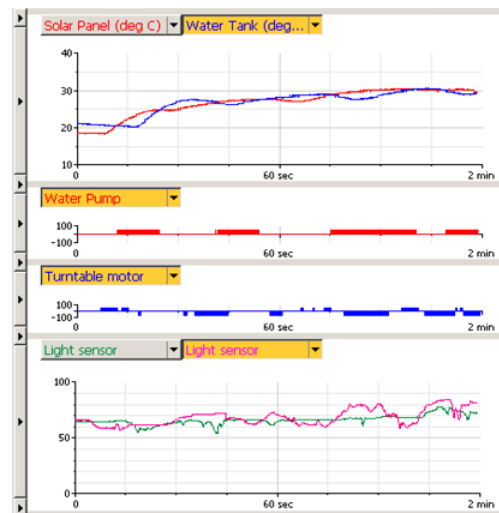
difference is greater than 5 units. This removes the tendency for the turntable to twitch back and forth with small differences in the light intensity.

Data Logging

The cause and effect in the system is easily demonstrated by using the Flowol Graphs:

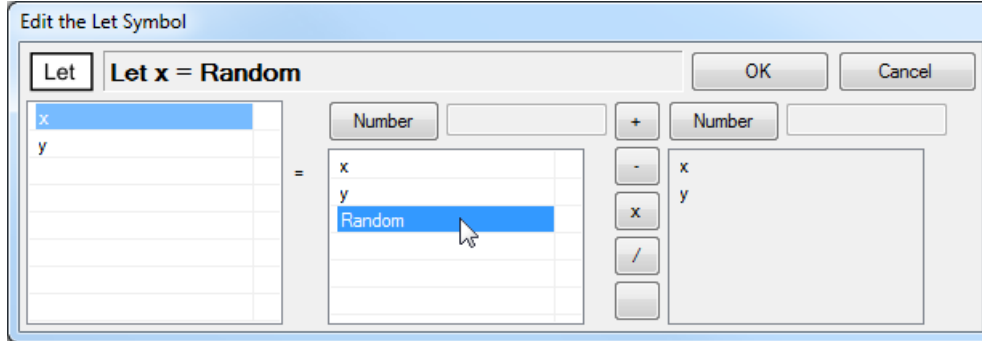
- The Pump motor responds to the temperature difference.
- The Turntable responds to the light difference.

In the graphs, the temperature sensors are calibrated in degrees Celsius.



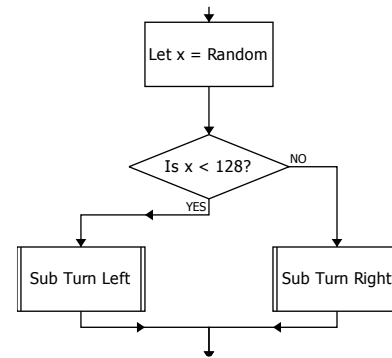
Random Numbers

The Random function will return a pseudo random number between 0 and 255. Use the random function by assigning the random number to a variable in the Assignment symbol:



A common pattern is to branch based on the random value giving the behavior of the system a random component.

Consider a floor robot which needs to explore its environment. It could do this deterministically (in an entirely predictable way), but the robot might be able to quickly explore more of the room using a random movement. The flowchart shown here randomly turns the robot left or right with an approximately even distribution.



Random branching like this could also be used to give the lights on a theme park ride (e.g. the Ferris Wheel/Big Wheel) a more interesting, irregular pattern.

Subroutine Parameters

Subroutine parameters are local values that can be defined in a subroutine definition. The parameter values are set when the subroutine is called.

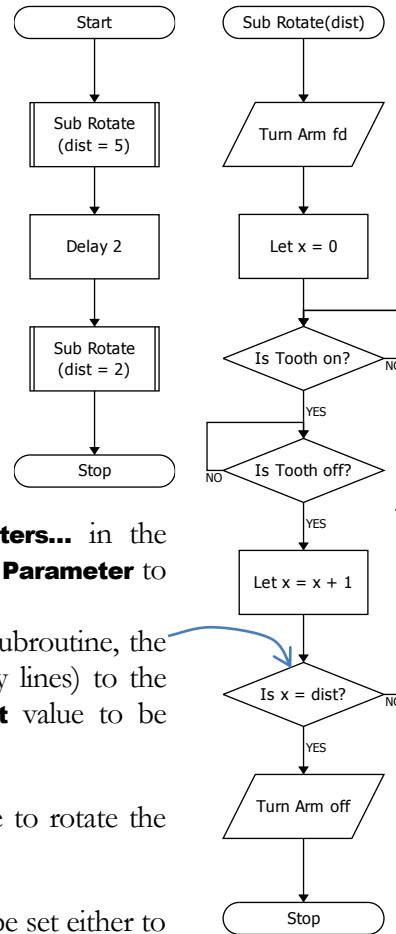
In the example to the right, there is a robot arm on a turntable (motor **Arm**) which can be rotated. The motor has a gear (cog) with a switch (input **Tooth**) that is pressed by each tooth on the cog. The tooth switch can be used to count the rotation of the turntable.

Construct the example to the left:

1. Make sure that the Global Variables are added.
2. Define the subroutine first, click on **Parameters...** in the subroutine definition prompt box, and then **Add Parameter** to add the **dist** parameter.
3. Since the **dist** value is only local to the Rotate subroutine, the decision symbol needs to be first connected (by lines) to the **Sub Rotate(dist)** symbol in order for the **dist** value to be available in the prompt box.

The subroutine is called twice by the main thread: once to rotate the turntable by 5 teeth, and a second time by 2 teeth.


When the subroutine is called, the parameter values can be set either to a constant number, or to the value of a global variable. If set to a global variable, note that the value is passed to the subroutine, not a reference.



Using multiple Mimics and/or Interfaces

Flowol 4 can support multiple mimics and/or multiple interfaces from the same flowchart.

Adding a Mimic

Click the Select Mimic button  at the top-right of the Flowol window. The **Choose Mimic** dialog has a dropdown at the bottom left. The dropdown will give you some of the following options:

- Add the mimic to Flowol.
- Replace the 'Old' mimic with this mimic.
- Add the mimic and link with the 'Hardware Interface'.
- Add the mimic independent of the 'Hardware Interface'.

For example, add two of the fairground themed mimics and control them both with the same flowchart.

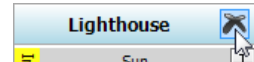
Adding a Hardware Interface

Click the Select Hardware Interface button  at the top-right of the Flowol window. The Choose Hardware Interface dialog also has a dropdown at the bottom left:

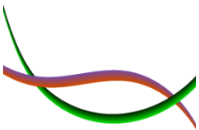
- Add the interface to Flowol.
- Replace the 'Old' interface with this interface.
- Add the interface and link with the 'Mimic'.
- Add the interface independent of the 'Mimic'.

Using multiple interfaces can be useful for controlling really big models which use many inputs and outputs.

To remove mimics or interfaces from the workspace, click on their delete button in the Status Panel. Use **Undo** to get it back.



Advanced Thread Control



When the flowchart is run, each Start symbol creates a new **programming thread**. These threads run until they reach a Stop, and once all threads have stopped Flowol stops running the flowchart.

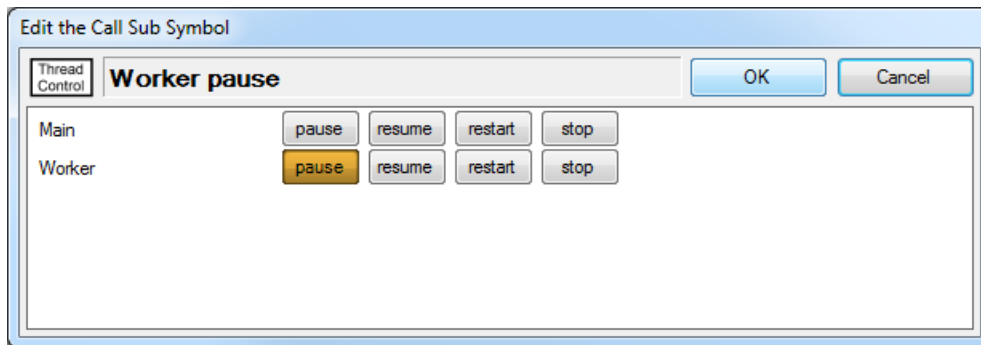
It is possible for one thread to have influence over another through the use of global variables. For example in a railroad crossing/level crossing scenario the main thread could set the variable **x** to be 1 when it needs the two red lights to start flashing, and then set it back to 0 again to stop; the worker thread would only flash lights when **x = 1**.

Advanced Thread Control provides an alternative mechanism. Click on the **More...** button at the top right and then, in the Add a New Feature dialog, choose **Advanced Threads**.

Once Advanced Threads has been added to the flowchart, threads can now be named. This is done in the Start symbol prompt box. Existing threads will be named 1, 2 etc.



The named threads can now be controlled with the new Thread Control symbol. Drag and drop this symbol onto the workspace. This symbol pauses, resumes, restarts and stops any thread:



Multiple threads can be controlled from one Thread Control symbol.

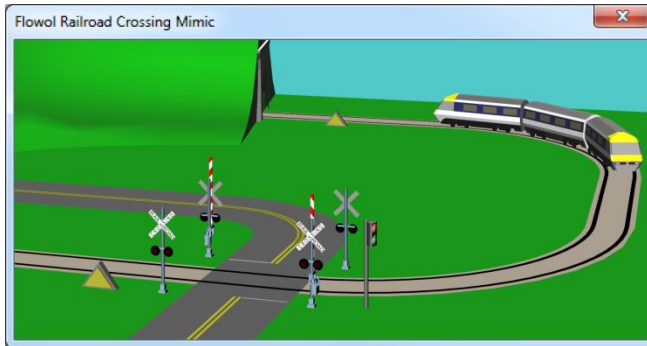
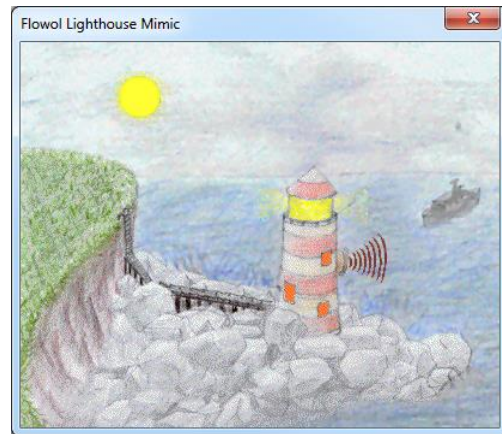
Chapter
6

Chapter 6: Mimic Activities

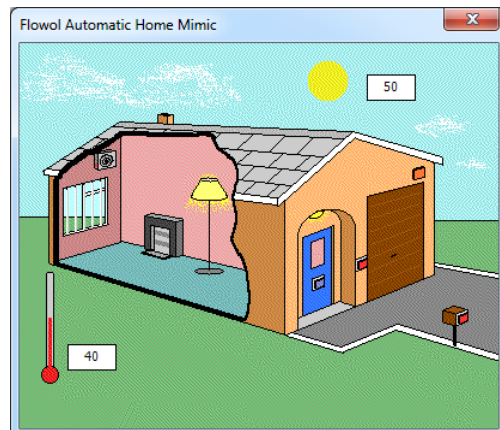
Automatic systems are all around us, keeping us safe, making life comfortable and helping us with difficult and unpleasant tasks.

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 learn to be able to solve more complex examples such as an automatic railroad crossing/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?



Use the grey boxes for teacher marks, grades or notes.

Print copies of these student activities for use in the classroom

NAME	_____
CLASS	_____
DATE	_____



Zebra Crossing

Open the Zebra Crossing mimic.

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

What is special about the Belisha beacons (amber lights) at the crossing? Why are they there?

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

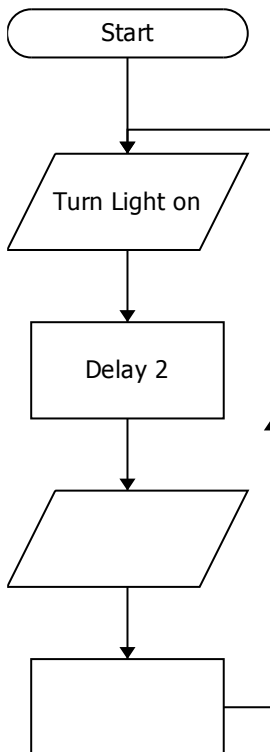
The pictures in Flowol are called Mimics and you can control them. To see what the mimic can do, click on the light in the picture.

See p. 13 in the tutorial for more help.



Activity 1

Create the instructions (a program) to control the light by building this flowchart.



Click and drag each symbol from the left toolbar and place it on the workspace. 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.

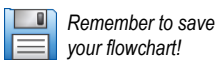
Remember to add your own instructions to the blank symbols.

Click on Run ► to see if your flowchart works.

The example uses a delay of 2 seconds. Is this a good time? Why would the Belisha beacon be less effective if the delay was too short or too long?

What improvements would you like to see made at pedestrian crossings?

What other types of road crossings already exist?



NAME	_____
CLASS	_____
DATE	_____



Crossing Patrol

See p. 16 in the tutorial for more help.

Open the Crossing Patrol mimic.



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

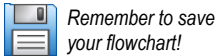
There are two lights on the signpost.

What are these lights for?

Do the lights come on together, or alternately?



Create a flowchart to control these two lights.



Click on Run to check your flowchart and then make any improvements.

How can crossing a road be made even safer for school children?

Use the Label Tool **T** to add a title to your flowchart.

NAME _____
 CLASS _____
 DATE _____



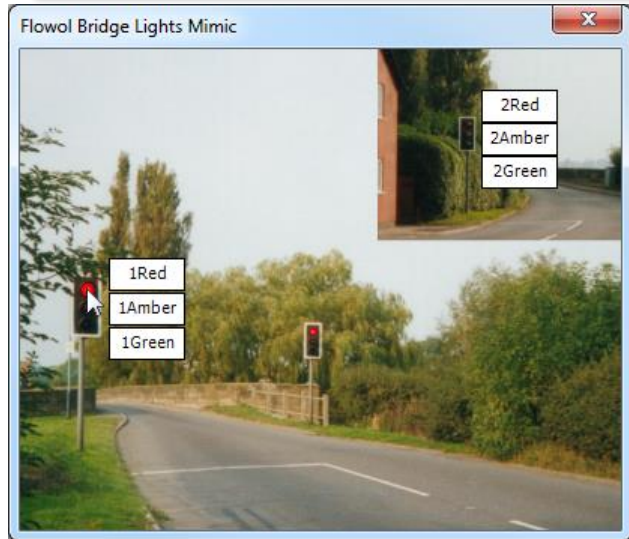
Bridge Lights

Is there a narrow bridge or road near your school that requires traffic lights in order to avoid a collision?

You've seen a single traffic light sequence many times before. But what is the combined sequence when both sets work together?

Explore how the mimic looks when the outputs are turned on by clicking on the outputs in the Status Panel.

See p. 20 in the tutorial for more help.



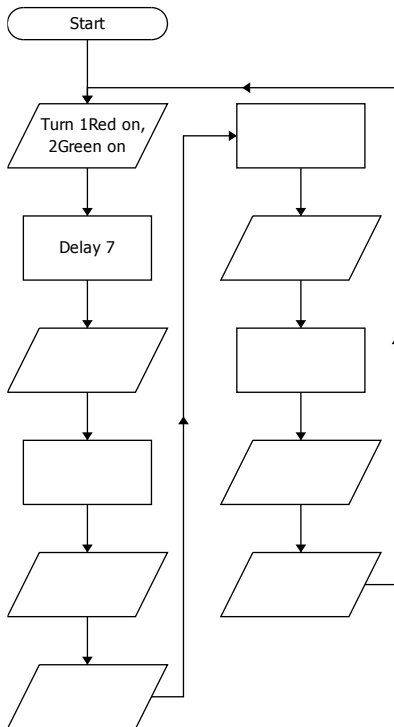
Bridge Lights	
1Green	I
1Amber	I
1Red	I
2Green	I
2Amber	I
2Red	I

Activity 1

First, create a flowchart to control a single set of traffic lights.

Activity 2

Now, modify your flowchart to control both sets of lights together. The flowchart might look like the one below.



Remember to save your flowchart!

Remember to fill in the empty symbols.

Click on Run to see if your flowchart works. And make any refinements or modifications if necessary.

Tooltips

When an output symbol is controlling 3 or more outputs, the text may be truncated if it does not fit in the symbol. When this is the case, and you move the mouse pointer over the symbol, a tooltip will appear showing the full text.

NAME	_____
CLASS	_____
DATE	_____

See p. 17 in the tutorial for more help.



All-way Stop

Open the All Stop mimic. It shows a fairly quiet intersection.



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

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

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

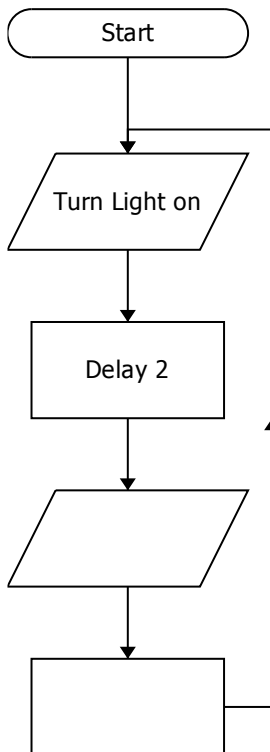


The pictures in Flowol are called Mimics and you can control them. To see what the mimic can do, click on the light in the picture.



Activity 1

Create the instructions (a program) to control the light by building this flowchart.



Click and drag each symbol from the left toolbar and place it on the workspace. 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.

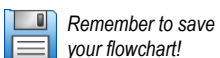
Remember to add your own instructions to the blank symbols.

Click on Run ► to see if your flowchart works.

The example uses a delay of 2 seconds. Is this a good time? Why would the stop light be less effective if the delay was too short or too long?

What improvements would you like to see made at road intersections?

What other types of stop light are there?



NAME	_____
CLASS	_____
DATE	_____



Crosswalk

See p. 19 in the tutorial for more help.

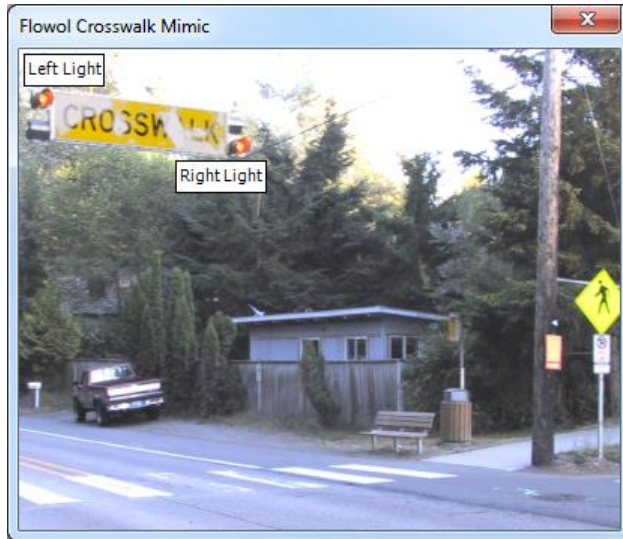
Open the Crosswalk mimic.

Where is there a crossing like this in your neighborhood?

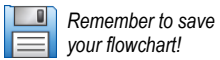
A crosswalk has two lights suspended above it.

What are these lights for?

Do the lights come on together, or alternately?



Create a flowchart to control these two lights.



Click on Run to check your flowchart and then make any improvements.

How can crossing a road be made even safer?

Use the Label Tool **T** to add a title to your flowchart.

NAME _____
 CLASS _____
 DATE _____



Intersection Lights

Where are the traffic signals in your neighborhood? They may hang from cables in the middle of the street like these, or be fixed to metal supports which reach over the road.

You've seen a single traffic light sequence many times before. But what is the combined sequence when both sets work together?

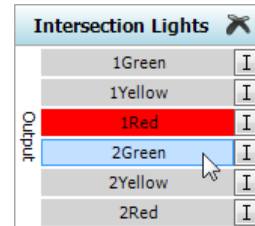
See p. 20 in the tutorial for more help.



Explore how the mimic looks when the outputs are turned on by clicking on the outputs in the Status Panel.

Activity 1

First, create a flowchart to control a single set of traffic lights.




Activity 2

Now, modify your flowchart to control both sets of lights

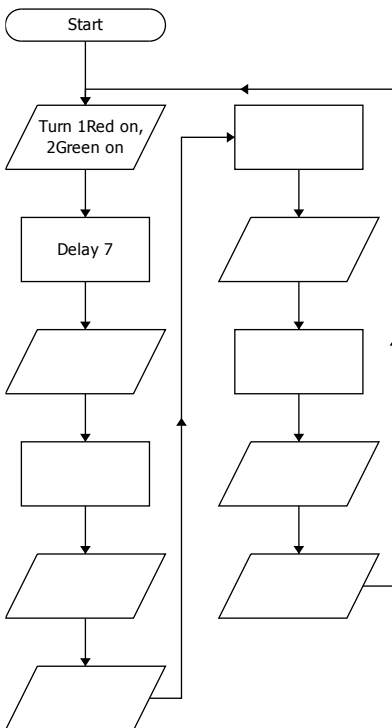
together. The flowchart might look like the one below.


Remember to fill in the empty symbols.

Click on Run  to see if your flowchart works. And make any refinements or modifications if necessary.

Tooltips

When an output symbol is controlling 3 or more outputs, the text may be truncated if it does not fit in the symbol. When this is the case, and you move the mouse pointer over the symbol, a tooltip will appear showing the full text.



 Remember to save your flowchart!

NAME	_____
CLASS	_____
DATE	_____

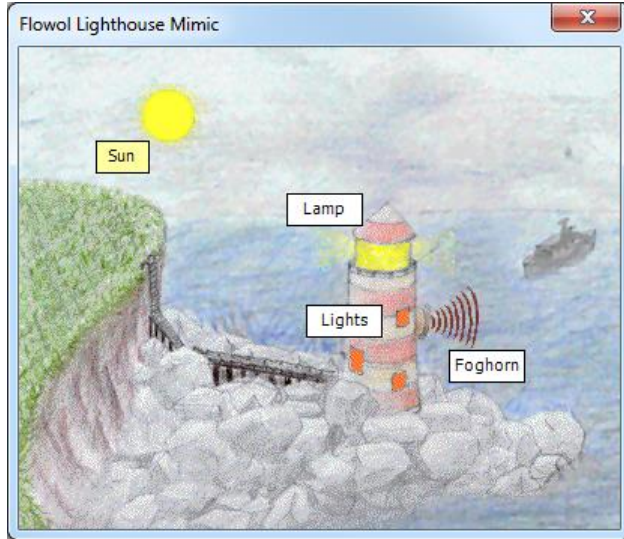


Lighthouse

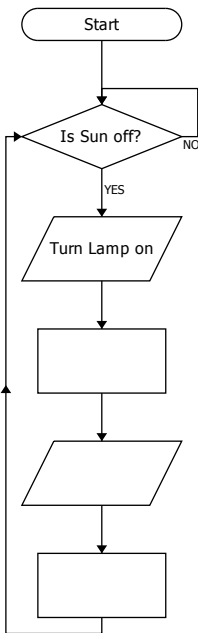
See p. 23 in the tutorial for more help.

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

Open the **Lighthouse** mimic and explore by clicking on the three outputs: **Lamp**, **Lights** and **Foghorn**, and the input **Sun**.



The Sun input is representing a digital light sensor which is on when it is daylight. Click on the sun/moon to toggle it.



Activity 1

Construct this control flowchart to turn on the flashing Lamp of the lighthouse at nighttime. Add some labels to your flowchart.

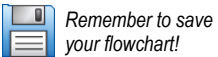
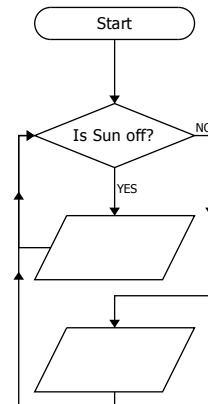
Note that you always need both a **YES** and an **NO** line from a decision symbol.



Activity 2

Now create another flowchart on the same workspace to control the inside Lights. The inside lights should stay on when it is dark and go off automatically in the daytime.



Since both flowcharts have a Start, they will both run in parallel when you click ▶.



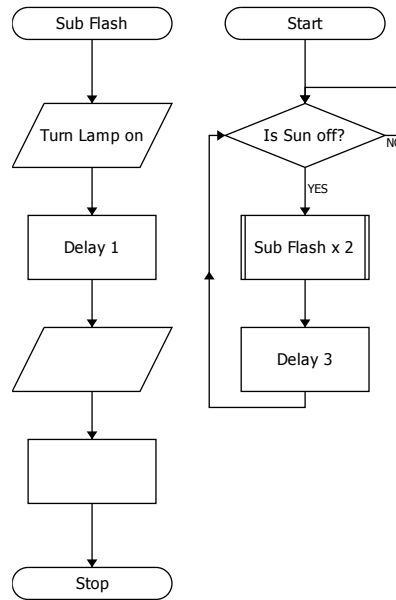


Activity 3

Create a more interesting flashing sequence with a subroutine.

A subroutine must first be defined with the  symbol. Once the subroutine has been defined, the **Call Sub** symbol  will appear on the left toolbar. Use it in the main flowchart to call (invoke) the subroutine. In the example to the right, the **Flash** subroutine is run twice (x 2).

Now adjust the main flowchart to create your own interesting flashing sequence.



Activity 4

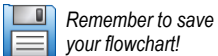
Control the foghorn by constructing another flowchart.



Activity 5

Add an "Mmmmm" sound to your foghorn. If you have a microphone and suitable recording software, record your own sound.

See page 26 in the tutorial for help playing a sound.



NAME	_____
CLASS	_____
DATE	_____

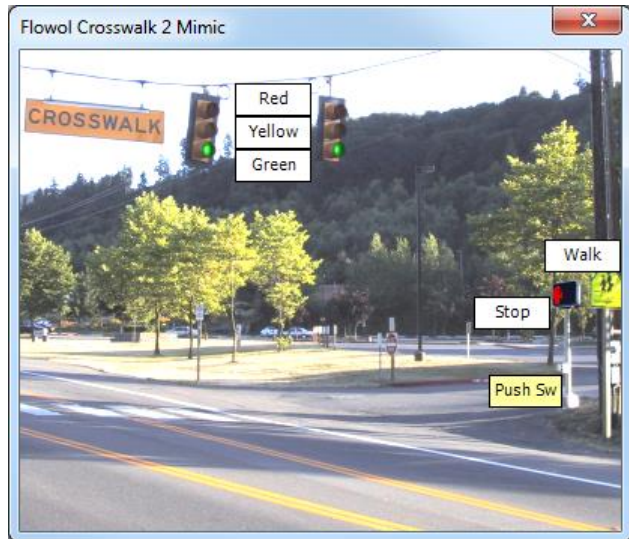


Crosswalk with Stop Light

Open the **Crosswalk 2** mimic.

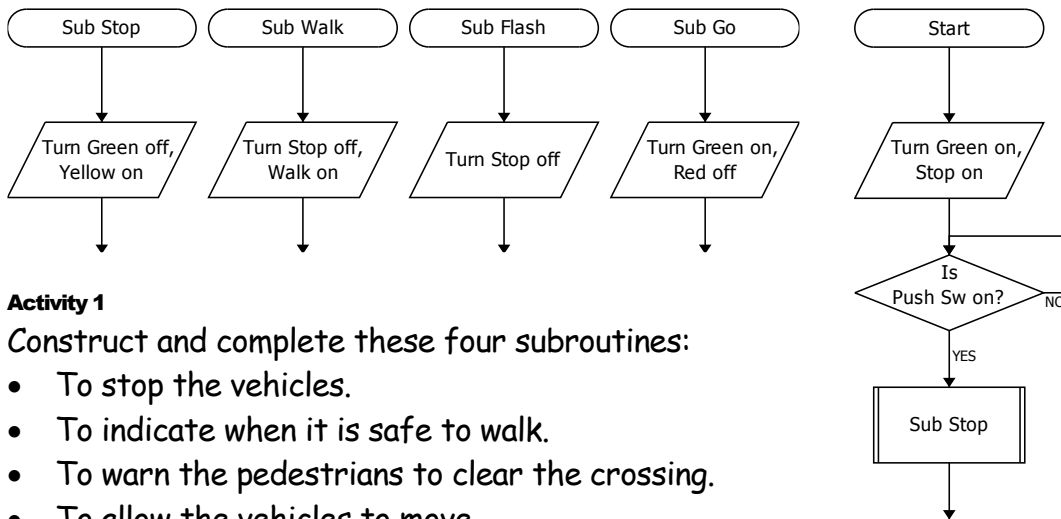
Click on the **Push Sw** Input (the small white circle) on the mimic window. This is a push switch with goes off after $\frac{1}{2}$ a second.

Also explore the Outputs 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.



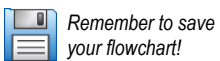
Activity 1

Construct and complete these four subroutines:

- To stop the vehicles.
- To indicate when it is safe to walk.
- To warn the pedestrians to clear the crossing.
- To allow the vehicles to move.

Activity 2

Complete the main flowchart to call (invoke) the subroutines correctly.



NAME	_____
CLASS	_____
DATE	_____

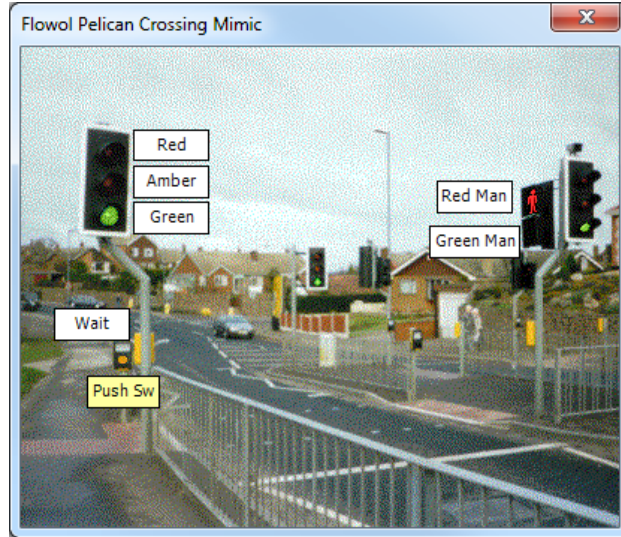


Pelican Crossing

Open the **Pelican Crossing** mimic.

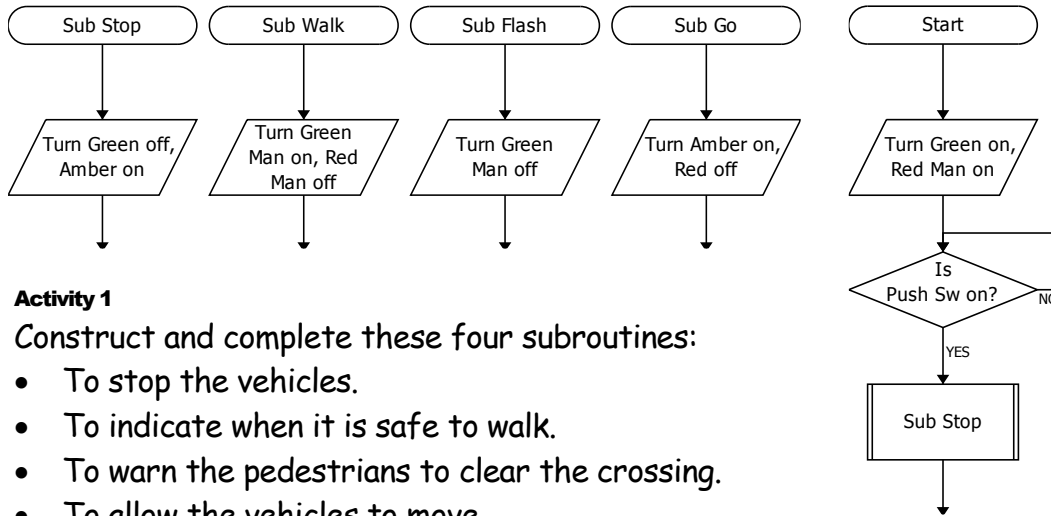
Click on the **Push Sw** Input (the orange circle) on the mimic window. This is a push switch with goes off after $\frac{1}{2}$ a second.

Also explore the Outputs 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.



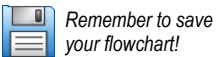
Activity 1

Construct and complete these four subroutines:

- To stop the vehicles.
- To indicate when it is safe to walk.
- To warn the pedestrians to clear the crossing.
- To allow the vehicles to move.

Activity 2

Complete the main flowchart to call (invoke) the subroutines correctly.



NAME _____
 CLASS _____
 DATE _____

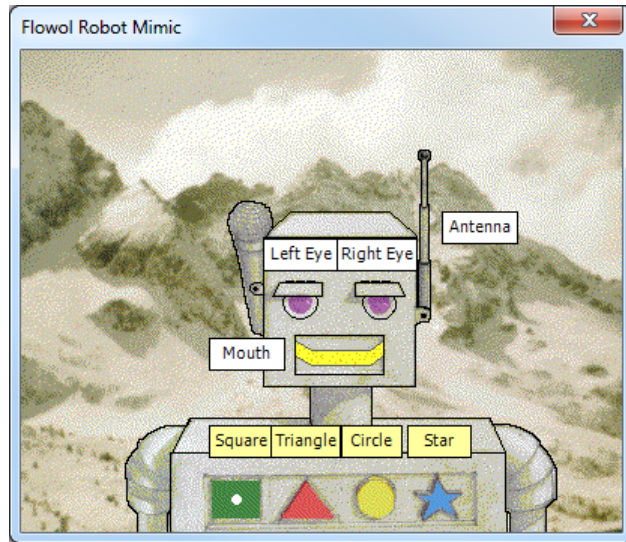


Robot

Let's have a little fun with the **Robot** mimic.

It has 4 inputs along its chest, and 4 outputs. Explore the mimic by clicking on the Status Panel:

Robot		
Input	Square	I
	Triangle	I
	Circle	I
	Star	I
Output	Antenna	I
	Mouth	I
	Left Eye	I
	Right Eye	I



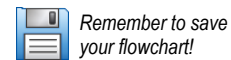
Activity 1

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



Activity 2

Make your robot talk using the Speech feature.



See page 27 in the tutorial for help using speech.



Activity 3

Construct a program to animate the robots mouth while the computer is speaking so it appears as if the robot is speaking.



Activity 4 (Larger project)

Imagine that the robot is a toy for a young child who is just learning their colors and shapes.

Create a program for the robot toy which uses Speech to ask the child to press a certain shape or colored button. Make the robot tell them if they got it correct or whether they need to try again.

Remember to break the program into subroutines.

When the child gets the correct button make the robot react in an excited way opening and closing its eyes etc.

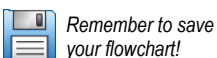


Activity 5 (Advanced)

Use the Random feature of Flowol to extend the solution above so that the robot randomly asks the child to find different buttons. Ask by color or shape.

See page 37 in the tutorial to get a larger flowchart workspace.

See page 75 in the tutorial for help using Random numbers.



NAME	_____
CLASS	_____
DATE	_____



Santa

Season's Greetings!

Open the **Santa** mimic. Use the Status Panel to see what the mimic can do.

Santa	
In	Switch
Output	Light 1
	Light 2
	Light 3
	Light 4
	Light 5
	Nose
	Fire
	Santa



Activity 1

Using the **Switch**, construct a flowchart to turn the electric **Fire** on and off.

Activity 2

Construct another routine to make Rudolph's **Nose** flash when the **Switch** is on.

Activity 3

The Christmas tree lights could be controlled in several ways. You could make them all come on together when the **Switch** is on. You could also make them all flash together, or they could twinkle if you have different groups coming on and off at different times. Create your own program to control the lights.

Activity 4

Santa's movement can be controlled with the **Santa** digital output. Write a subroutine to move Santa once. Then create a main program to call (invoke) the subroutine and make him dance.

Activity 5

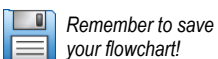
Use the Sound feature to play a recorded Christmas greeting and make Santa sing it to you.

See page 26 in the tutorial for help using sound.

Activity 6 (Advanced)

Use the Random function to have the Christmas tree lights flash in a subtle, but random fashion.

See page 75 in the tutorial for help using Random numbers.



NAME	_____
CLASS	_____
DATE	_____



Mobile

See p. 29 in the tutorial for more help.

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

Mobile	
Input	Green
	Yellow
	Blue
	Val 0.0 %
Output	Car
	Balloon
	Port
	Starboard
	Mobile
	Helicopter
	Plane

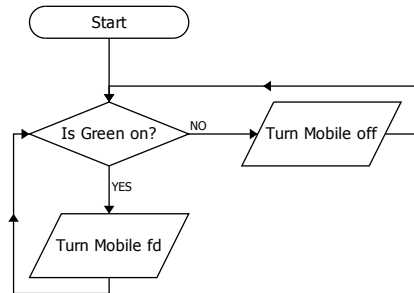
Open the **Mobile** mimic, and show its labels. Explore what it can do by clicking on the outputs and motors in the Status Panel.



Click with the right mouse button on the motors to reverse them.



Activity 1
Controlling the motor to rotate the whole mobile is the most impressive effect. Build this program to use the **Green** digital input to start and stop the **Mobile** motor.

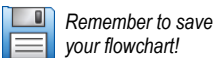


Activity 2
Motors can of course go forwards and reverse. Construct another two flowcharts to give some forward and reverse movement to the **Helicopter** rotors and the **Plane** propeller when the **Yellow** and **Blue** inputs are used.

Run the whole program. This should give some interesting combination movements when the three switches are changed.



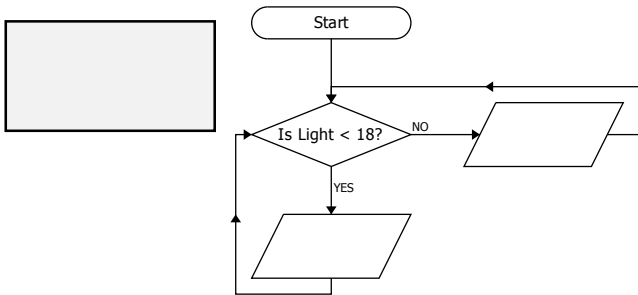
Activity 3
Another important control feature for a motor is to change its speed (or power). Modify your first flowchart to reduce the main rotation speed of the mobile for a while and then speed it up again. [Remember, if you reduce the motor speed percentage, then it must be returned back to 100% for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



See page 34 in the tutorial for help using variables to control the speed of the motors.

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

The **Mobile** mimic has an analog sensor at the top of the mobile support arm which changes value when you click on the number with the left and right mouse buttons.



Activity 4

Rename the analog value from **Val** to **Light** and treat it as a light level sensor. Construct an automatic light flowchart to turn on the light in the hot air **Balloon** if the light level goes below a value of 18 units (i.e. Light < 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 the **Port** and **Starboard** plane wing tip lights flash if the light level value goes below 60 units.

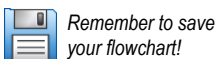


Activity 6

The cot mobile would be most interactive if it could respond to the baby's sound. So rename the analog sensor to **Sound**. Larger values represent a louder sound.

Rebuild the flowchart to respond to the sound that the baby is making. If the baby makes a quiet noise, some of the lights could turn on or flash for a while. If the noise gets louder, the rotors and propeller could start to move slowly, and if the baby gets very noisy, the whole mobile could become very active.

As the baby settles down, and makes less noise, the mobile should also slow its activity to sooth the baby back to sleep.



A gentle musical sound could also be added to the mobile.

Finally, use a word processor or the label feature in Flowol to write some brief instructions for the parents.

NAME	_____
CLASS	_____
DATE	_____



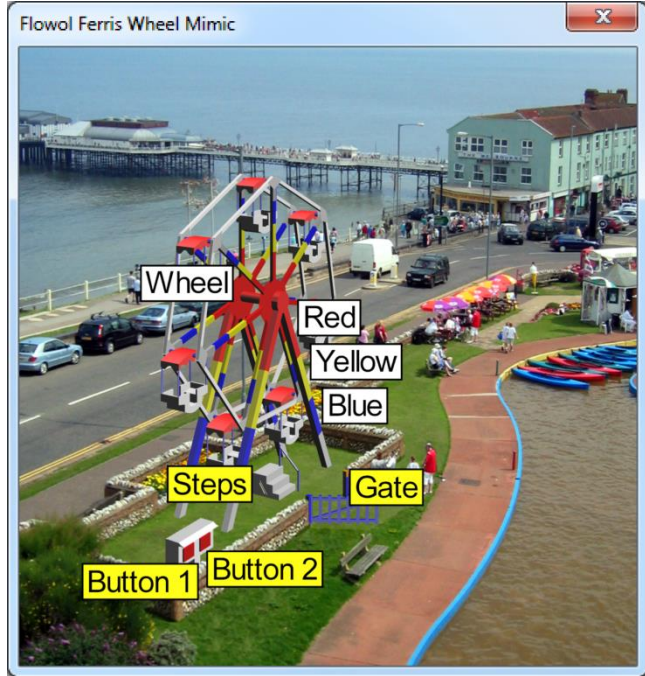
Ferris Wheel/Big Wheel

See p. 32 in the tutorial for more help.

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

Open the **Ferris Wheel/Big Wheel** mimic. Explore what the mimic can do by clicking on the inputs, outputs and motors on the Status Panel.

Button 1 and **Button 2** are normal inputs but, when the wheel is rotating, you may notice that the **Steps** input flashes on each time a seat passes over the steps. Also, if you click on the **Gate** with the left mouse button you will find that the **Gate** input comes on when the gate is shut.



The Steps and Gate inputs are called virtual inputs. They cannot be changed by directly clicking on the mimic, but are changed by features within the mimic itself.



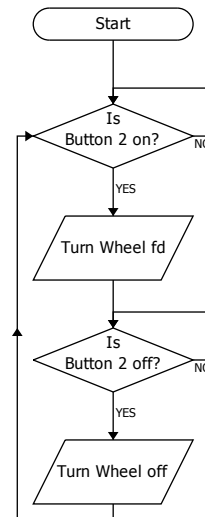
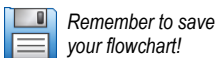
Activity 1

To attract the crowd, use the **Button 1** input to control the lighting effects on the wheel's frame. This might be a simple on/off routine but flashing sequences are more exciting. Use subroutines.



Activity 2

Use the **Button 2** input to control the simple Go/Stop movement of the wheel. You could perhaps make the wheel speedup and slowdown in stages by changing the motor power.





Activity 3

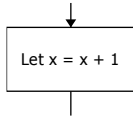
Modify activity 2 to include the safety feature of the gate so that the gate must be closed before the wheel will start (i.e. both **Button 2** and **Gate** are on). The wheel should stop if either **Button 2** is turned off, or the **Gate** is opened (i.e. if either **Button 2** or **Gate** are off).



Activity 4

To increase the 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.

See p. 27 in the tutorial for help using speech.



Activity 5: Counting how many times the ride is used

Add this symbol to the flowchart created above in Activity 2. This uses variable x to count how many times the ride is used.



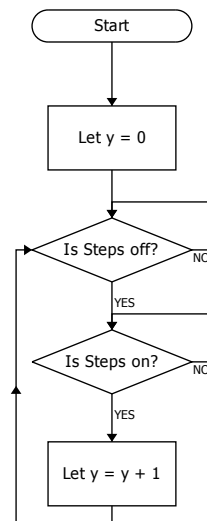
Activity 6: Stopping the Wheel automatically

Construct this counting program to increase the variable y each time a seat passes the steps, i.e. each time the virtual input goes off and on.

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

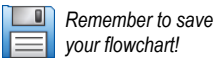
Now modify your program by introducing a decision symbol, to stop the wheel automatically after it has rotated 3 times.

See page 33 in the tutorial for help using variables.



Activity 7

Now that you have learned how to use the **Steps** input, create a subroutine which rotates the wheel and stops briefly at each of the seven seats for passengers to get on or off.



Call (invoke) this subroutine twice; once at the beginning to load the wheel with passengers, and then at the end to unload.

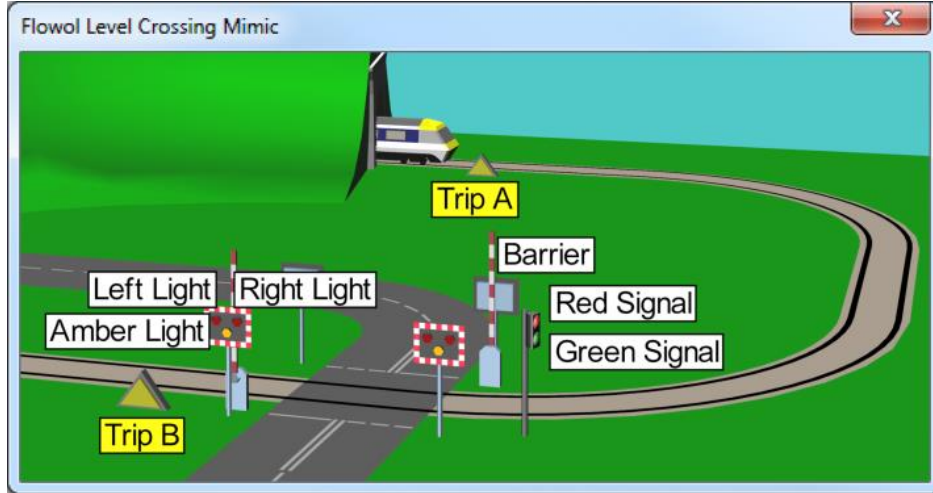
NAME _____
 CLASS _____
 DATE _____



Level Crossing

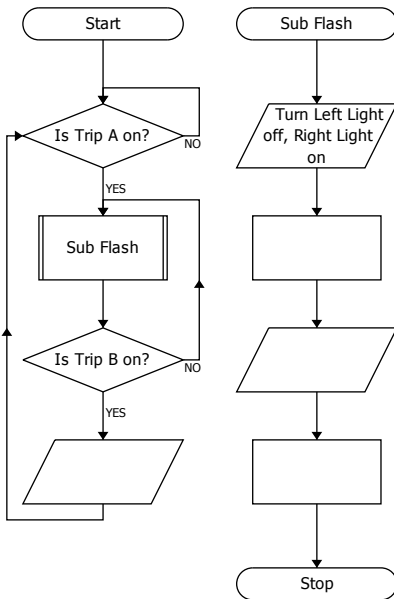
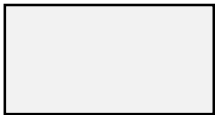
See p. 35 in the tutorial for more help.

Open the **Level Crossing** mimic.



There are two inputs: **Trip A** and **Trip B** which cannot be clicked on. Instead they are triggered when the train goes past.

The train runs automatically whenever the flowchart is running. Click Run now, even without any flowchart, to see the train move.



Activity 1

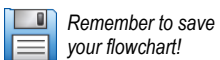
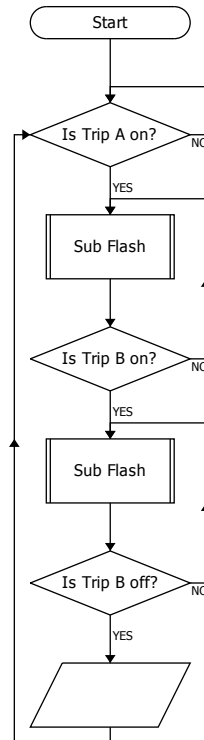
Construct and complete the flowchart to the left to flash the **Left Light** and **Right Light** outputs when the train approaches, and turn them off when the train has passed by.

Activity 2

Add symbols to control the **Amber Light**.

Activity 3

The solution above is not quite correct since the lights stop flashing as soon as the front of the train reaches **Trip B**.



Modify the main flowchart to match the one on the right to correct this.

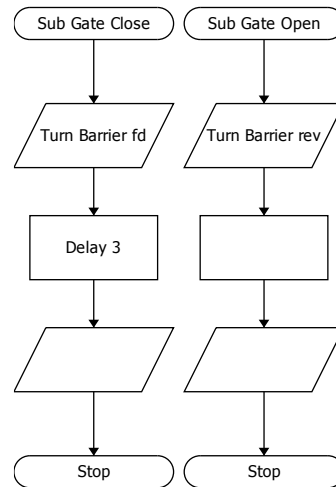


Activity 4

The **Barrier** motor moves the gate. Turn it forward for a certain time period to close the gate and reverse to open the gate.

Create two subroutines, **Gate Close** to close the gate, and **Gate Open** to open it. Call (invoke) these subroutines from the main flowchart to close and then reopen the gate.

Replace the Delay in the **Gate Close** subroutine to call the Flash subroutine an appropriate number of times to keep the lights flashing while the barriers are closing.



Activity 5


The train can be stopped in an emergency with the signal. To stop the train, turn **Red Signal** on.

Add output symbols in appropriate places to your flowchart so that the Red Signal is shown whenever the barrier is not down.

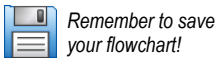
To verify that the signal works correctly, reduce the speed of the barrier so that it doesn't have time to completely close by the time the train arrives at the signal.



Activity 6 (Advanced)

Adjust the mimic options (click the  button on the Status Panel) to use feedback switches on the barriers.

Re-run your existing flowchart to see how the barriers vibrate when they reach the feedback switches. Modify your flowchart to use the feedback switches to stop the barrier movement.



Now that the feedback switches are used to control the barriers, it's hard to keep the lights flashing because you don't know exactly how long it will take to lower the barrier. Therefore, create a separate, parallel flowchart which flashes the lights whenever variable **x = 1**. Then set variable **x** to be **1** in the main flowchart to start flashing the lights, and set it back to **0** to stop the lights flashing.

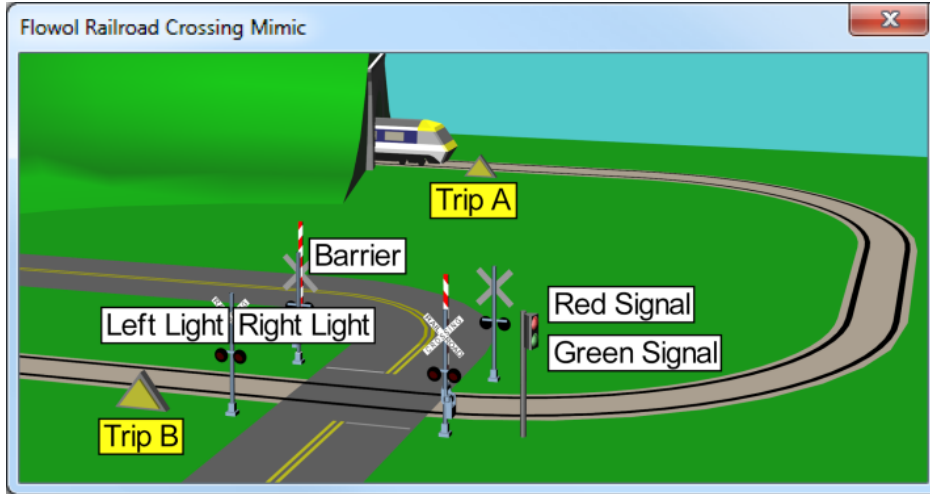
NAME _____
 CLASS _____
 DATE _____



Railroad Crossing

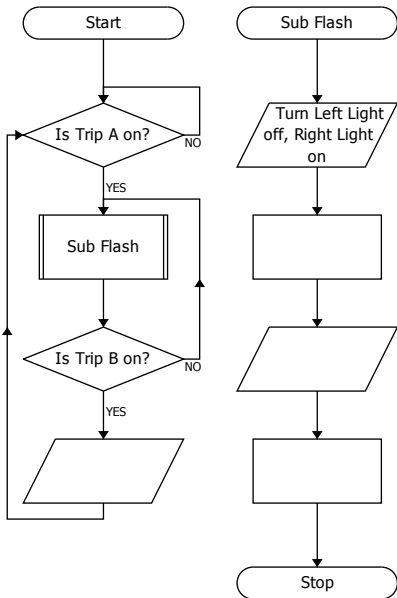
See p. 35 in the tutorial for more help.

Open the **Railroad Crossing** mimic.



There are two inputs: **Trip A** and **Trip B** which cannot be clicked on. Instead they are triggered when the train goes past.

The train runs automatically whenever the flowchart is running. Click Run ► now, even without any flowchart, to see the train move.



Activity 1

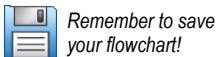
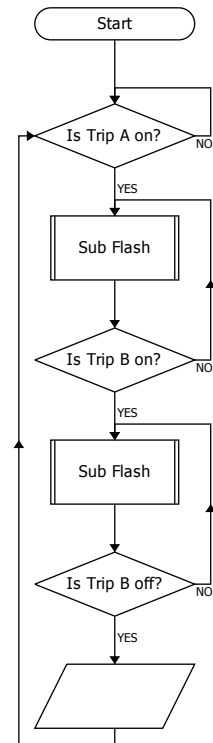
Construct and complete the flowchart to the left to flash the **Left Light** and **Right Light** outputs when the train approaches, and turn them off when the train has passed by.



Activity 2

The solution above is not quite correct since the lights stop flashing as soon as the front of the train reaches **Trip B**.

Modify the main flowchart to match the one on the right to correct this.



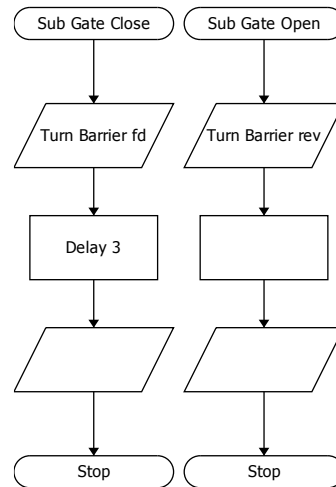


Activity 3

The **Barrier** motor moves the gate. Turn it forward for a certain time period to close the gate and reverse to open the gate.

Create two subroutines, **Gate Close** to close the gate, and **Gate Open** to open it. Call (invoke) these subroutines from the main flowchart to close and then reopen the gate.

Replace the Delay in the **Gate Close** subroutine to call the Flash subroutine an appropriate number of times to keep the lights flashing while the barriers are closing.



Activity 4


The train can be stopped in an emergency with the signal. To stop the train, turn **Red Signal** on.

Add output symbols in appropriate places to your flowchart so that the Red Signal is shown whenever the barrier is not down.

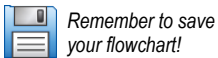
To verify that the signal works correctly, reduce the speed of the barrier so that it doesn't have time to completely close by the time the train arrives at the signal.



Activity 5 (Advanced)

Adjust the mimic options (click the  button on the Status Panel) to use feedback switches on the barriers.

Re-run your existing flowchart to see how the barriers vibrate when they reach the feedback switches. Modify your flowchart to use the feedback switches to stop the barrier movement.



Now that the feedback switches are used to control the barriers, it's hard to keep the lights flashing because you don't know exactly how long it will take to lower the barrier. Therefore, create a separate, parallel flowchart which flashes the lights whenever variable **x = 1**. Then set variable **x** to be **1** in the main flowchart to start flashing the lights, and set it back to **0** to stop the lights flashing.

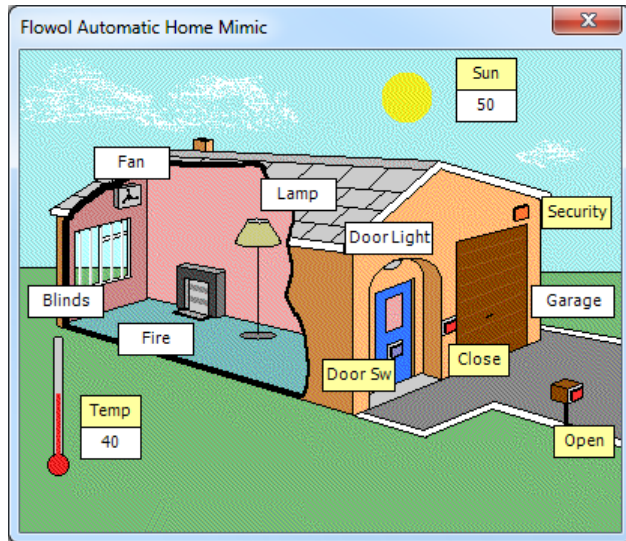
NAME	_____
CLASS	_____
DATE	_____



Automatic Home

What automatic control features do you have in your home? What control features do you want?

Open the **Automatic Home** mimic and explore.



Activity 1

Construct a program to open the garage door when the **Open** button is pressed, and close it when the **Close** button is pressed.

Activity 2

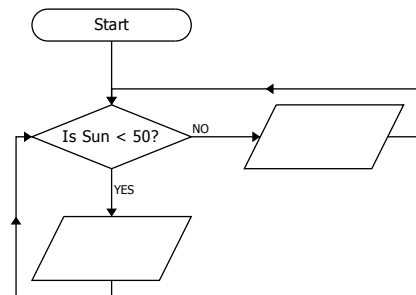
Assume that the **Security** digital input is a movement/infra-red heat sensor which can detect a person on your driveway. Construct a program to turn on the **Door Light** when a person is detected.

Daylight Brightness and Temperature

By clicking on the numbers near the **Sun** and **Temp** thermometer, you can make the sensor readings increase or decrease in steps of 5. The sensors are calibrated by default as a percentage. If you have interface hardware, connect it and calibrate the temperature sensor to °Celsius or °Fahrenheit.

Activity 3

Construct a flowchart, like the one shown, to turn on the inside **Lamp** only when the daylight **Sun** value goes below 50%.

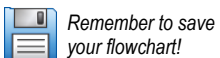


Activity 4

Construct another flowchart to make the electric **Fire** come on when the temperature goes lower than 30 units.

Activity 5

Now control the electric cooling fan above the window. Think about the temperature threshold that you choose.





Activity 6

What else can we do? The window **Blinds** can be controlled electrically. Make them automatic.



Activity 7

Look at Activity 2 again, the one with the person detector. How can you improve this system?



Activity 8

Make the push button (**Door Sw**) on the front door trigger the sound of a bell, or a recording of a dog barking.

Monitoring and Logging

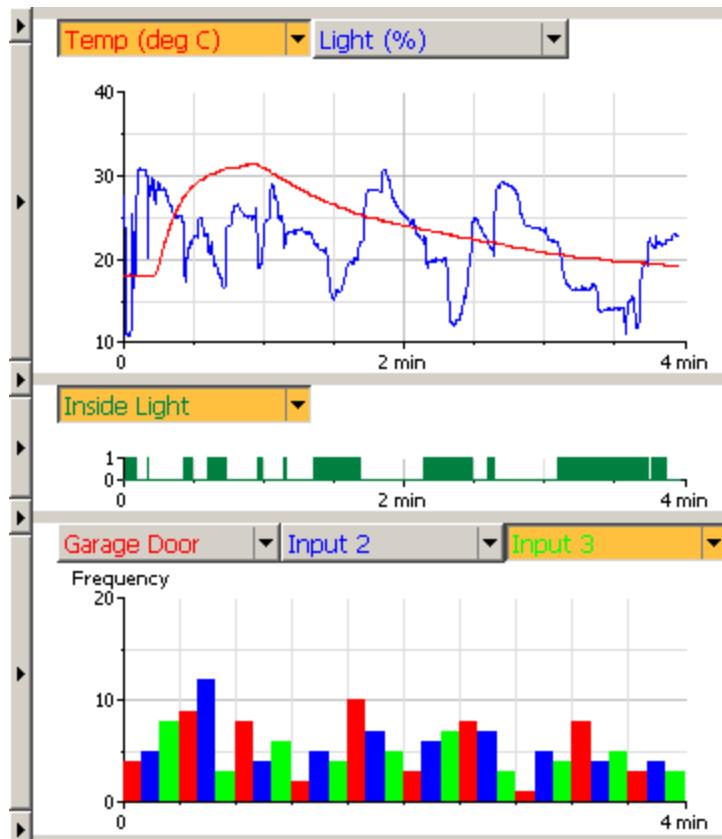
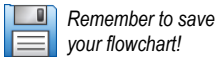
The home can be monitored and data logged using the Flowol Graph feature. Configure the graph window to show the information you want to observe.

See page 70 in the tutorial for help using graphs.



Activity 9

Re-run your program and keep changing the different inputs. The graph shown here is logging temperature, daylight brightness, the inside light and how often the garage door is opened.



NAME	_____
CLASS	_____
DATE	_____



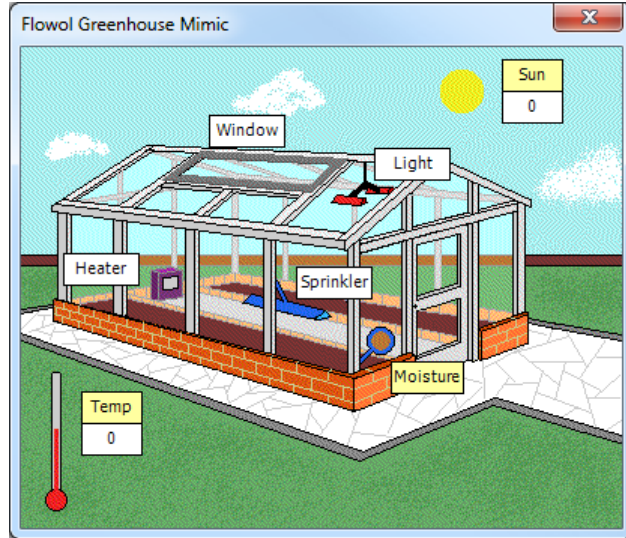
Greenhouse

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

Open the **Greenhouse** mimic and explore its functions.

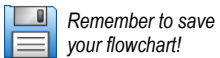
Moisture is a digital input that is off when the soil is dry and on when it is moist. The ambient temperature and light can also be measured with the analog **Temp** and **Sun** sensors.

You can control the **Lights**, **Heater**, **Window** and **Sprinkler**.



Assignment

This is an open assignment. Apply your knowledge and skills to make this greenhouse look after the plants for you.



Make sure to label the different flowcharts to show clearly what you are trying to achieve.

NAME	_____
CLASS	_____
DATE	_____

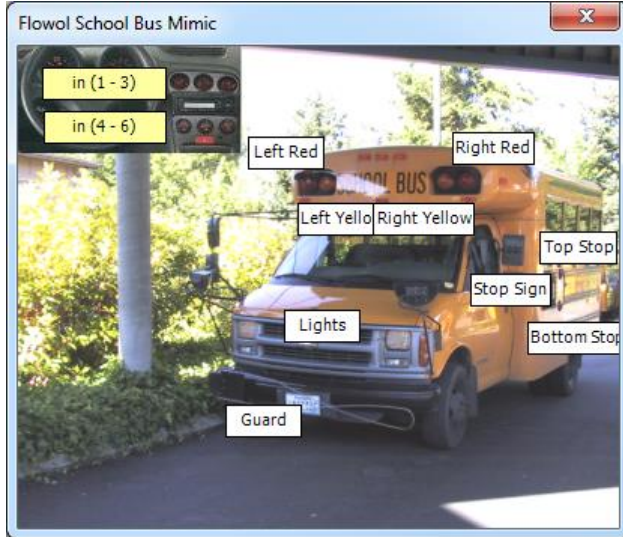


School Bus

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

Open the School Bus mimic and explore its features.

The driver has six buttons available to control the various lights, **Stop Sign** and **Guard** paddle. These buttons should be used in sequence to operate the safety features in the right order.



Activity 1

Use the first button and build a program to operate the general front **Lights**.

Activity 2

Create another program to control the flashing yellow lights to indicate to passing motorist that the bus is about to stop.

Activity 3

The **Guard** paddle 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

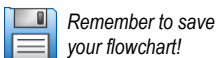
The red alternating traffic warning lights should then be activated.

Activity 5

The **Stop Sign** should then be deployed.

Activity 6

Finally the alternating lights on the stop sign should be illuminated.

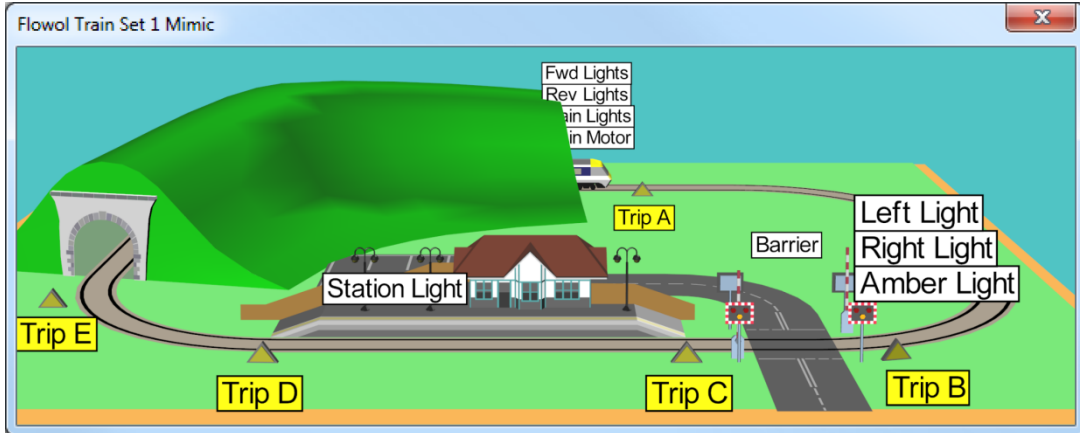


NAME	_____
CLASS	_____
DATE	_____



Train Set 1

Open the Train Set 1 mimic, add the labels and use the Status Panel to explore what the mimic can do.



Now imagine that you are the engine driver, the guard, the signal operator and station manager and manually control the train, its lights, the station lights, the crossing lights and of course the barrier gates. How easy did you find that?

You would probably be more successful, and safer, if you are just the engine driver and the other functions are controlled by a system created by you using Flowol 4.



Activity 1

Since you are the engine driver, build this flowchart allowing you to use **Input 1** on the input Status Panel to control the clockwise movement of the train. What do the **Fwd Lights** do?



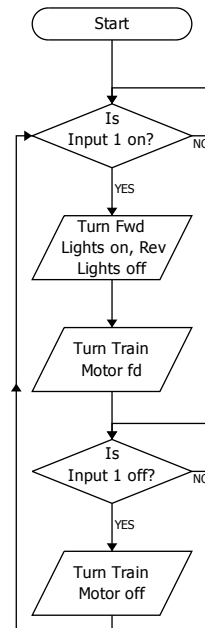
Activity 2

Now build a similar program to control the reverse movement of the train by using **Input 2**.

Trip Switches

For the next activities we will assume that the train will be moving clockwise around the track.

You may have noticed that the moving train turns several input switches on automatically when it passes over them (the yellow triangles light up).





Activity 3

To be energy efficient, build a program to turn the passenger carriage lights (**Train Lights**) on automatically only while the train is passing through the tunnel.

Note that the carriage lights should go on when the front of the train enters the tunnel and go off again when the back of the train leaves the tunnel.

See page 95 for an example of a similar flowchart.



Activity 4

Now build a similar program to turn the **Station Lights** on only while the train is passing or is stopped at the platform.



Activity 5

Apply what you learned from the **Level Crossing/Railroad Crossing** mimic to control the barrier and lights for vehicular traffic.



Activity 6

Perhaps you can add an air horn sound to your program when the train passes through the tunnel. You could produce your own sound using a microphone and suitable recording software, or look for a .wav file from the internet.



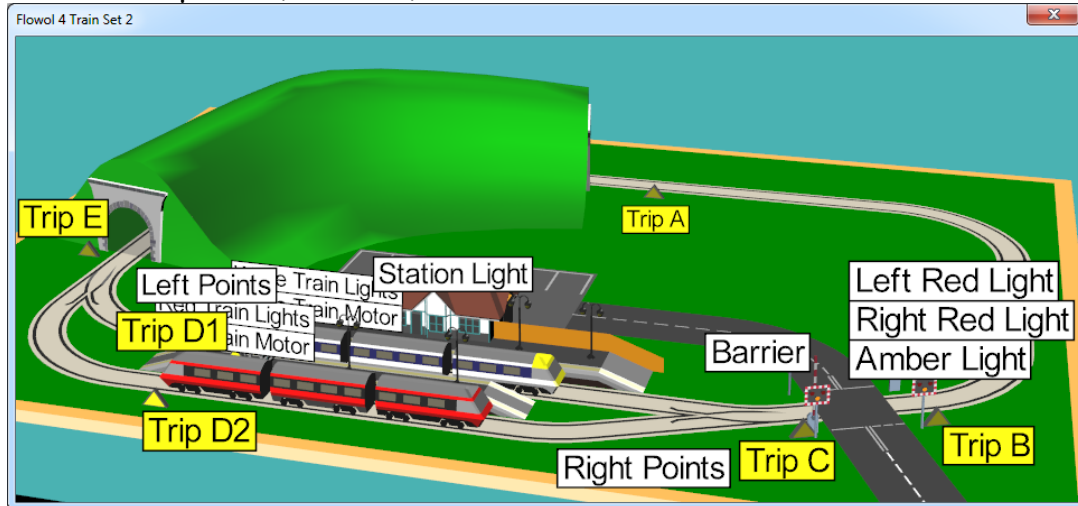
Remember to save your flowchart!

NAME	_____
CLASS	_____
DATE	_____



Train Set 2

The Train Set 2 mimic is more complicated. There are now two trains, and two sets of points (switches).



Each train has its own motor and lights. Click on the motors in the Status Panel to drive each train (right-click the mouse to turn the motor in reverse).

Each set of points (switches) are controlled by the **Left Points** and **Right Points** motors. Turn the motor forward to switch to the outside track and reverse to switch to the inside track.

Activity 1

Create two sub-routines, one for switching to the outside track, and another to switch to the inside track.

Activity 2

Create a master program which drives each train around the track one at a time.

Activity 3

Incorporate all of the features from the TrainSet1.

Activity 4

Have fun!

Index

- absolute path, 26
- All Stop, 17, 82
- Analog values, 31
- Analog values in the Flowchart, 31
- Apple Mac, 5
- Arduino, 61
- Automatic Home, 99
- Barrier, 35
- Belisha beacons, 79
- Big Wheel, 32, 93
- Binary I/O Values, 41
- Bluetooth, 40, 52
- Brainy-Motor, 66
- Brainy-USB, 66
- branches, 24
- Bridge Lights, 20, 81
- Calling the Subroutine, 25
- Clipboard, 28
- CoCo, 51
- Colors, 38
- Commotion, 51, 60
- Commotion Coco, 60
- condition, 24
- Connecting, 40
- Contact Controller, 59
- Contact Controller Plus, 59
- Control Boxes, 39
- Control Station, 58
- Copy, 28
- Copying into a word processor, 28
- Copyright, i
- Counting, 34
- Country, 38
- Country Setting, 13
- crib mobile, 91
- Crossing Patrol, 16, 80
- Crosswalk, 19, 83, 87
- Cut, 28
- Data Logging, 74
- decision symbol, 24
- Decision symbol, 10
- Delay symbol, 9
- Deltronics, 51, 60
- Deltronics Serial Plus, 60
- DFRobot, 61
- Digital Encoders, 63
- Downloading, 40
- DS18B20 digital temperature sensor, 66
- Duemilanove, 61, 62
- Encoder Motor, 48
- Encoder Motors, 53
- Encoder Motors in your Flowchart, 54
- Extension Modules, 58
- feedback switch, 35
- Ferris Wheel, 32, 93
- File paths, 26
- Fischertechnik Intelligent, 58
- Fischertechnik Robo, 57
- Fischertechnik Robo LT, 57
- Fischertechnik Robo TX, 52
- Flowchart file, 7
- Flowchart Symbols, 9
- FlowGo, 50
- FlowIC Kit, 67
- Flowol 2, 7
- Flowol 3, 7
- Foghorn, 23
- font, 22
- Freeduino, 61
- frequency bars, 73
- Global Variables, 33
- Graph Options, 72
- Graphs, 70
- Greenhouse, 101
- Hardware interface, 7
- Hardware Interfaces, 39
- HC-SR04, 63
- Helicopter, 29
- Hitachi HD44780, 63
- InBin, 41
- Input in the Flowchart, 24
- Inputs, 11, 40
- Inserting an additional Symbol, 15
- Installing Flowol, 5
- Interface, 11, 39
- Interface Options, 40
- interrupted, 36
- Intersection Lights, 20, 84
- Introduction, 4
- Junior Serial, 60
- Labels, 20, 22
- Language, 38
- LCD Displays, 63
- LEGO Dacta Control Lab, 61
- Let symbol, 9
- Level Crossing, 35, 95
- License key, 5
- light sensor, 31
- Lighthouse, 23, 85
- Line tool, 10
- LM35 linear temperature, 63
- loop, 36

- Loop, 15
- Margin function, 74
- Microcontrollers, 39
- Mimic, 7
- Mimic Creator, 38
- Mimic Options, 35
- Mimic Packs, 37
- Mimic Window, 21
- Mimic with the Interface, 41
- Mimics, 13
- Mobile, 29, 91
- Monitoring, 70
- motor in the flowchart, 29
- Motor Shield, 62
- Motors, 29
- Moving Sections of Flowchart, 23
- multiple interfaces, 76
- multiple Mimics, 76
- Naming Inputs and Outputs, 40
- nesting, 25
- Numbered Interface, 69
- Optical Shaft Encoder, 47
- Options, 38
- OutBin, 41
- Output symbol, 9
- Output Voltage, 59
- Outputs, 12, 40
- Page Setup, 37
- Parallel Programming, 23
- Paste, 28
- Pause, 11, 22
- Pelican Crossing, 88
- PICAXE, 65
- PICAXE chip, 65
- Plane, 29
- Playing a Sound, 26
- Power Control, 30
- Print Preview, 37
- Printing, 37
- Prompt Box, 9
- PWM Outputs, 62
- Railroad Crossing, 35, 97
- Random Numbers, 75
- relative path, 26
- Renaming Variables, 33
- Robot mimic, 89
- Running the Flowchart, 11
- Santa mimic, 90
- School Bus, 102
- scroll bars, 22
- Select tool, 10
- Serial, 40
- Serial Communication, 64
- Serial Interface, 60
- SerialRx, 64
- Servo motor, 48
- Servo Motors, 62
- Silence, 27
- Simulation. See Mimic
- Single Step, 22
- Smart Box, 59
- solar panel, 74
- Solar Water Heating Panel, 74
- Solo LCD, 69
- Solo18/Solo28, 68
- Sound, 26, 31, 32
- sound sensor, 31
- Speech, 26
- Speed, 30
- Speed of a Motor with a Variable, 34
- speed slider, 22
- Start symbol, 9
- Stop Light, 87
- Stop symbol, 9
- Sub symbol, 9
- Subroutine, 10
- Subroutine Parameters, 76
- Subroutines, 25
- Sun Seeker, 74
- temperature sensor, 31
- Text Tool, 10, 22
- Text-To-Speech, 27
- thread, 23
- Thread Control, 77
- Tooltips, 21
- Traffic Lights, 20
- traffic signals, 84
- Train Set 1, 103
- Turntable, 74
- Tutorial, 13
- Ultrasonic Distance Sensor, 63
- Ultrasonic Range Finder, 47
- Undo and Redo, 15
- Uno, 61
- USB, 40
- USB to Serial adapter, 40
- Variables, 33
- VEX Cortex, 42, 46
- VEX Partner Joystick, 47
- VEX Programming Kit, 46
- VEXnet Firmware, 46
- VEXnet Joystick, 46
- virtual input, 32
- WAV files, 26
- Welcome dialog, 7
- Windows PC, 5
- Workspace, 9
- Workspace Size, 37
- Zebra Crossing, 13, 79
- Zooming, 22

