MST Workshop 6.5

Math, Science, and Technology

User's Guide

5/27/2013

May 2013

Math, Science, and Technology Workshop version 6.5

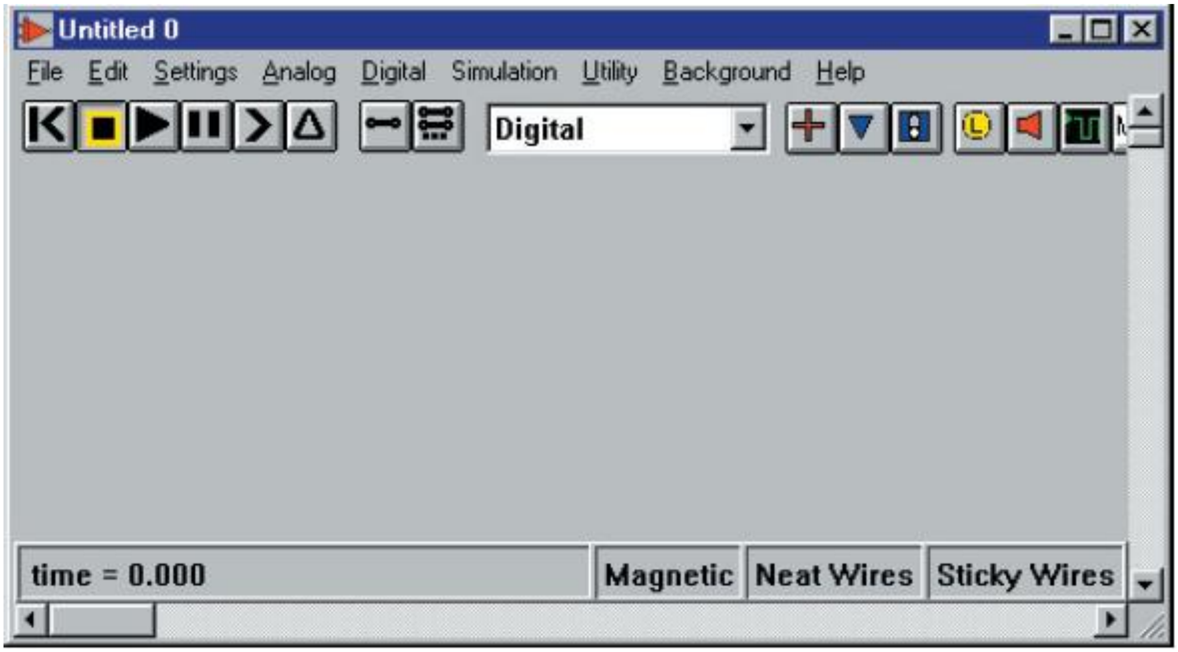
Contents

Chapter 1. Using MST Workshop

This chapter describes the tool bars, workspace, and menus used in MST Workshop, It also explains how to place components on the workspace for building a circuit, and how to operate a circuit.

The MST Workshop Workspace and Tool bars

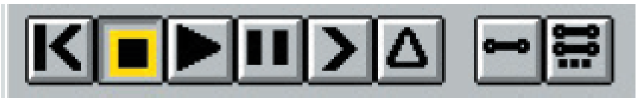
When you run MST Workshop, an empty workspace appears. The tool bar at the top may look different from the figure below depending on your previous settings. The program always displays the tool bar that was active when the program was closed.



The MST Workshop Workspace

Tool bars

To determine the function of a tool bar button, hover over it with the mouse.



The Left side of the tool bar

The first eight buttons are identical on all tool bars. They include:

- Six buttons to operate a circuit once it is built.
- Two wire buttons to connect inputs to outputs.
- A list menu lets you change the tools on the right side of the tool bar.

Circuit Operation buttons and Wires are described in *Operating a Circuit,* later in this chapter.

Menus corresponding to the all the tool bars are always available, regardless of the tool bar that is showing on the right.

Each toolbar has unique components:

- The Digital and Analog toolbars contain sources, functions that perform operations on the inputs, and/or displays.
- The Simulation toolbar contains templates that let you define circuits for building a car, gate, traffic signal, or other simulations. The cars, gates, and signals that you build will animate when placed on the workspace.
- The Communication toolbar provides components that communicate in eight bit codes called ASCII characters.
- The 2D Tools and 2D Shapes toolbars provide two-dimensional shapes and transformations to create animated and interactive drawings.
- The 3D Tools and 3D Shapes toolbars provide three-dimensional shapes and transformations that allow you to create three dimensional animated drawings.
- The Trigonometry toolbar provides access to components that can perform trigonometric functions.
- The Comparison toolbar provides access to components that compare or modify their inputs.
- The More Math toolbar provides access to components that are handy functions.
- The Bitwise Logic toolbar provides access to components that operate on a bit level with integer inputs.
- The Astronomy toolbar provides access to components that can calculate positions of the Sun, Moon, and Planets.
- The Special toolbar provides access to components needed to build integrated circuits.
-

Each of these toolbars is described in later chapters.

Placing Components on the Workspace

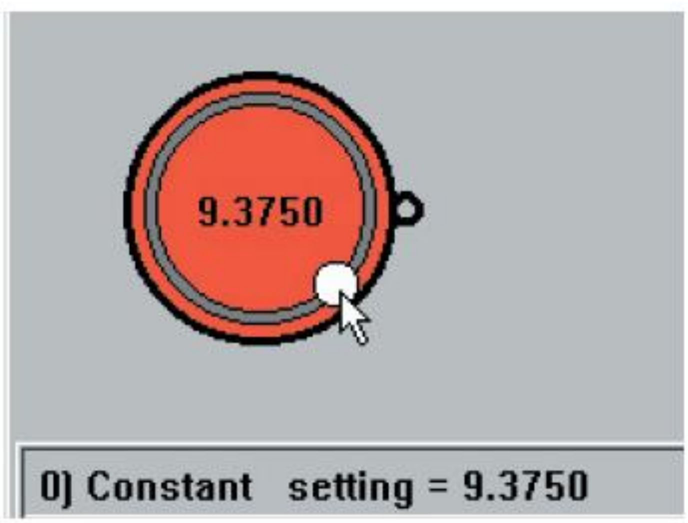
> *To place a toolbar component on the workspace, click it once with the mouse then move the mouse to the desired location. Click the mouse again on the workspace to place the component.*

Note: If you enable the Sticky Buttons menu, you can click a tool multiple times or click various tools to gather components. Then move to the workspace and place one component with each click of the mouse. Components will be placed in the same order they were gathered. If you change your mind and want to forget about the components you have gathered you can click on any of the circuit operation buttons.

Note: Wires require two clicks, one to start it and another to finish it.

The Wires button lets you place multiple wires. When you click on it you will be able to place wires for as long as you want. When you are done click the Wires button again to turn it off.

> *To move a component to a new location on the workspace, drag the body of the component using the mouse. You can also drag over a group of components and wires to select them all and drag them all to a new location. Drag the group by clicking on a component other than a wire for the best results.*
>
> *To get information about the properties of a component on the workspace, hover over the component with the mouse. The information appears in the status bar at the bottom of the window. You can see the values at the input and output connectors.*



Hovering Over a Component Displays Internal Values

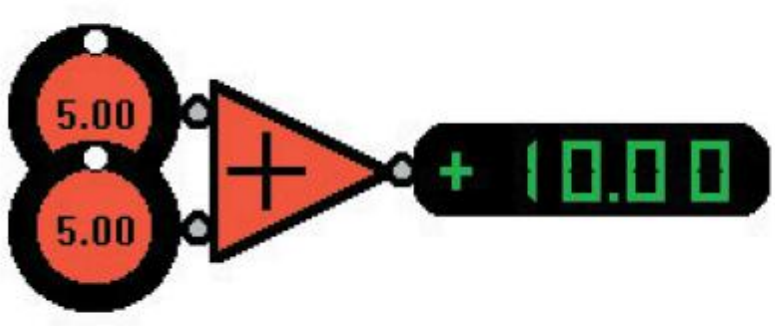
Remember these points when building circuits:

- Most circuits have at least one source, function, and display.
- Analog Cars, Digital Cars, Traffic Signals, and Gates each use a pair of icons. One is called a template and is used to build a circuit. The other is called an Icon and when placed on the workspace will obey the logic in the Template circuit with the same file name.

Input and Output Terminals

Terminals are the points where information goes into or out of components on the workspace. Sources contain only output terminals; that is, they send data to other components. Functions have both input terminals (to receive data from other components) and output terminals (to send data to other components). Displays have only input terminals to receive data.

The following figure shows a complete analog circuit. It contains two sources (constants), a function (adder), and a display (meter) to display the results.



A Simple Analog Circuit

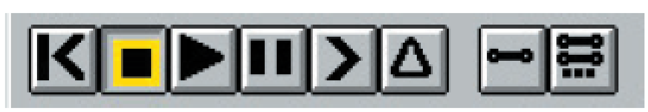
Changing Component Settings

> *To change the settings of a component, double-click the component on the workspace.*
>
> *To delete a component, double-click it then press the delete button in the dialog.*

See chapters 3 and 4, respectively for more information on changing settings of analog and digital components.

Operating a Circuit

All the toolbars contain the following tools. The first six operate a circuit once it is built. The next two are wires to connect components. The next four select the tools on the right side of the toolbar:

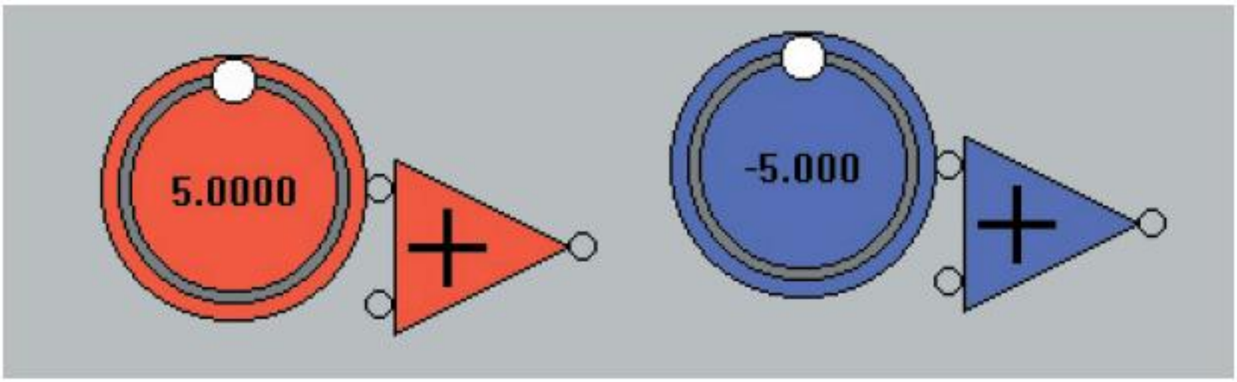


The Left side of the Tool Bar

Starting and Stopping a Circuit

To operate a circuit, press Run. To stop the circuit, press Stop.

When you run a circuit, most components on the workspace change color based on their output value. Digital components change from blue (true) to red (false). Analog components change color based on their output value.



Circuits Showing Color Based on Output Value

The elapsed run time appears at the lower left of the screen.

Pausing a Circuit

To temporarily stop the circuit, press Pause.
To restart the circuit, press Pause again.

Stepping a Circuit

*To step the circuit in 50 millisecond increments, press step***.**
To change the step setting, use Time Setup on the Settings menu (described later in this chapter under **Menus***).*

Resetting a Circuit

*To reset the circuit run time to 0.0, press Reset***.**

Small Steps

Activate the Small Steps button to step at 1 millisecond increments. Press it again to return to normal stepping.

Connecting Components with Wires

Wires are common to both digital and analog circuits. Use wires to connect outputs to inputs. Wires can be bent to have a better appearance. Wires can have up to eight bends. Use the wire button when you need a single wire. Use the wires button when you want to place multiple wires. When you are done placing wires click on the wires button again to turn it off.

To connect a wire to an input or output, drag the wire's connector to the desired component connector.
To bend a wire, click on the wire anywhere and drag it to where you want it.
To reposition a bend in the wire, click on the bend and drag it to a new position.
To remove a bend, click on the bend and move it until the wire is straight again.
To remove all the bends, right click the wire and select Straighten from the popup menu.
To move a wire without bending it hold down the shift key before you click the wire.

Menus

File and Edit Menus

In general these menus function to complete file operations and to select and move components on the workspace, as they do in other software applications.

Search by Content...

Use this item to find all the circuit files that contain a specific type of component. You can pick a file from the list then either open or delete it. Some components in the list have three dots (an ellipsis) after their name. When you select one of these and click the Search Now button, a dialog will pop up so you can enter some text. If you want a list of all the files that contain this component leave the text field empty and click Ok. If you enter text, only those files that have the component and have matching text will be shown on the list. For example you can search for any file containing an Integrated circuit or only show files that have an integrated circuit with a matching name. You can enter the whole name or just a part of the name but be sure the search string has capital letters to match the capitals in the component for which you are searching.

Import Package...

A Package is a file that contains a circuit file and all the other files that are required for the circuit to work. When you import a Package each file is unpacked and placed in the proper folder.

Export Package...

Most circuit files that are created with MST Workshop require additional files to work properly. If you want to send a circuit file to a friend who has MST Workshop, you need to include all the required files. For example, if your circuit contains an Integrated Circuit, you have to send both the original circuit file and the integrated circuit file. The **Export Package** menu will scan the current circuit and list all it's requirements. You can save these files in three ways. **Copy Files** saves all the files in one place. **Copy Files Into Folders** copies each file into an appropriate sub folder. If the sub folder is not present it will be created. This gives your friend a clue to where to put the files in his MST Workshop directories. **Save Requirements List** saves the list of the required files into a text file. **Save Package** creates a single file that contains the original circuit file and all the required files. Export will show you an error message if you haven't saved the file you are trying to export.

Settings Menu

Set Colors

> *Use this option to set colors of various elements of the program. Select an element on the right that you want to change then select the color on the left. You can put all the colors back to the way MST Workshop was installed by clicking on the Default button.*

Background color is the color of the workspace background.

Vehicle Background color is the color of the workspace in the 3D View window when vehicles, gates, or traffic signals are present.

Short Circuit is the color that wires will become when the wire is causing a short circuit. Electric Terminal is the color of the center of a terminal.

Plotter Paper is the color of the paper or background of the strip chart, x y recorder or scatter chart.

Plotter Minor Grid is the color of the minor grid on the strip chart, x y recorder or scatter chart.

Plotter Major Grid is the color of the major grid on the strip chart, x y recorder or scatter chart.

Unused Part is the color of any component when it has nothing connected to its input.

Note Background is the color of the paper or background of a Note.

Positive Voltage 0 through 10 are the colors that components become as their output voltage changes from 0 volts to 10 volts.

Neutral Voltage is the color for components with outputs of zero volts.

Negative Voltage 0 through 10 are the colors that components become as their output voltage changes from 0 volts to -10 volts.

Time Setup

Use Time Between Steps to set the time that the program will wait between each calculation.

Run In Real Time use this choice when you want your circuit to keep up with the computer's real time clock.

Manual Step Size use this choice when you want the circuit to take uniform steps no matter how long the computer takes to update all the components. When you select this choice the Time Step Size choice will enable.

For example, when trying to display the location of a falling object (as on a strip chart recorder), use a smaller time step such as 10 ms. The calculation runs slower, but displays more detail. If the time step is large, you may miss certain steps in the calculation. The default setting is 50 milliseconds.

Clock Sound

When Clock Sound is checked, the program will click once a second while the circuit is running.

Clock Units

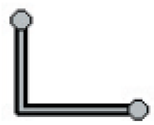
The workspace time is displayed in the status bar below the workspace. Clock Units may be set to Seconds, Minutes, Hours, or Days.

Magnetic Terminals

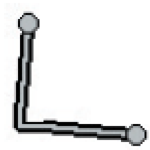
When Magnetic Terminals is checked, connectors will snap together when you attempt to connect them.

Neat Wires

The Neat Wires feature lets you easily create a true 90-degree angle in a wire; that is, both the horizontal and vertical sides of the angle snap straight when you approach 90 degrees. When Neat Wires is not checked, the angle is more difficult to achieve.



With Neat Wires Active



With Neat Wires Inactive

Sticky Wires

This feature makes it possible to move components without dropping the wires connected to the component.

Mouse Sounds

When Mouse Sounds is checked the various mouse activities will make a sound when:
Terminals magnetically snap together or are separated,
Wires are horizontally or vertically neatened or released,
Wires are straightened.

Multiple Selection

When Multiple Selection is checked you can click more than one component button before placing the components. This allows you to gather up a few components then place them on the workspace one at a time.

When This File Opens

When This File Opens contains 4 options: Open Workspace, Run Program, Open 2D View, and Open 3D View. Settings these allows you to create programs that will immediately start a program running, not show the workspace and open a 2D or 3D view window. Programs with these settings can be sent to others along with the MST Player.

Set Canvas Size

The Set Canvas Size feature lets you enlarge the workspace for large circuits. When the workspace is larger than the workspace scroll bars will automatically appear. You can also use this feature to reduce the workspace when trying to constrain vehicles.

Stop Following

The Stop Following feature lets you disable the Follow This Car feature.

Configure Directories (changes require a password)

The Configure Directories dialog lets you specify the location for the six directories that MST Workshop uses. You have three basic choices:

1. You can let MST Workshop decide where to create the directories.
2. You can specify a common beginning for the directory names such as H:\ or H:\LW or H:\LWDIR\. If you end the common beginning with a back slash the directories will be inside the directory you create and specify. Otherwise the directory name such as CIRCUITS will be added to the common beginning you specify such as LWCIRCUITS.
3. You can type in the complete name and path. This way allows you complete control over the directory names and locations.

Note 1: MST Workshop will create the standard directory names but you will have to create the path to that directory if you specify the directory path with multiple layers such as C:\A\B\CIRCUITS.

Note 2: If you make a mistake and MST Workshop can not startup correctly, you can return to MST Workshop's original settings by deleting the USERSET.DAT file in the MSTWorkshop directory.

Configure Constants

The Configure Constants dialog lets you record the Constant values, Slider values and switch positions. To create a configuration:

1. Click the New button to create a new configuration.
2. Click the Copy All Values button or check just the checkboxes for the constants you want to be part of this new configuration.
3. Click Done.

Later when you want to return the constants to their previous settings:

1. Select the configuration from the list at the top

2. Click the Set Constants Now button at the bottom.

If you haven't given the constants unique names, you can click the Fix Ambiguous Names button to make each constant have a unique name. You can also type in the values for the constants if they aren't set to the values you want recorded.

Analog and Digital Menus

These menus contain all the Sources, Functions, and Displays available on the Analog and Digital Toolbars, respectively. These components are described in detail in Chapters 3 and 4.

Simulation Menu

The Simulation Menu contains templates that let you define a circuit that operates a car or a traffic signal. Once you create a circuit using the template, you can save it and load it as an icon.

Communication Menu

The Communication Menu contains components that let circuits communicate with each other and the world outside your computer.

2D Graphics Menu

The 2D Graphics Menu contains components that let you create, manipulate, and display 2-Dimensional shapes.

3D Graphics Menu

The 3D Graphics Menu contains components that let you create, manipulate, and display 3-Dimensional shapes.

Astronomy Menu

The Astronomy Menu contains the items available in the Astronomy Toolbar. These items include components to calculate the positions of the Sun, Moon, and Planets.

Special Menu

The Special Menu contains the items available in the Trigonometry, Comparison, More Math, Bitwise Logic, Arrays and Special toolbars.

Lessons Menu

Lessons are circuits that contain instructions about how to build them and how they work. This menu will contain a list of all the lesson circuits contained in the Lessons directory. When a lesson is selected from the menu a dialog will appear that contains instructions and information. If the instructions say to place a wire or component you should do it then hit the Next button. If here is only information in the dialog you can hit Next to go on to the next step. The Lessons menu is built based on the contents of the Lessons directory each time MST Workshop starts.

Help Menu

About MST Workshop

This menu brings up a dialog that indicates the program's version number and contains information about contacting the author Tom Pandolfi.

Build Demo Circuit

The first time you start MST Workshop after installing on your hard disk a short demo will be built piece by piece on the workspace and then start running. This menu replays that demo.

Open User's Guide

This menu starts Adobe Acrobat Reader and opens the User's Guide. You can also open the User's Guide by starting Acrobat Reader, choosing Open, then navigating to the Programs/MSTWorkshop/Documentation directory and choosing "User's Guide.pdf"

Find A Component

This menu opens a dialog that helps you find a component. When It first opens all the components are listed in the same order they appear in the tool bars. Type a word in the Search For field and hit return or press the Search button. The list will then show components that contain the word you entered somewhere in their name or description. Click the Show Button In Toolbar button and the selected component will be colored yellow in the toolbar. If multiple matches were found, scroll through the list and select one before hitting the Show Button In Toolbar button.

Create New Lesson

You can add to the collection of lessons that MST Workshop shows in the Lesson Menu.

To create a new lesson start by building the final circuit you plan to use in the lesson. Be sure to save it before you start building the lesson -- it's good to have a copy when things don't go as planned.

Step 1. Build the circuit you plan to use in your lesson.

Step 2. Select Create a New Lesson from the Help menu. A new window will appear. Reorganize your workspace and the new window so that both are visible. Notice that the new window has a text entry area at the top, a popup menu for selecting automatic actions, a popup menu for choosing the actions required by the end user and a Next button at the bottom.

Step 3. Type in the lesson text in the text entry area at the top. This is what the user will be shown.

Step 4. If this part of the lesson requires some of the circuit components to be placed, select the components.

Step 5. If you want MST Workshop to automatically select a particular toolbar to help the end user select the action from the popup.

Step 6. Decide if this part of the lesson should require the end user to place the components, ask the end user to place the components but not require it, or have MST Workshop place the components without asking the end user. Make your choice in the actions menu.

Step 7. Click the Next button. The text entry area will clear and the selected components will disappear. Repeat the process starting at Step 3 until you have no circuit components left.

Step 8. Enter a final text message and click the Next button.

Step 9. Click the Save button. A dialog will appear that allows you to enter your name and a password. You can protect your work with a user name and password so that only you will be able to modify the lesson. Click Ok or Don't Password Protect.

Step 10. A standard Save dialog will appear. Name your lesson and save it in the Lessons directory. You can add to or change the directory structure and MST Workshop will build the Lessons menu to match it and the lesson files it contains.

Step 11. Quit MST Workshop and run it again. This gives MST Workshop a chance to rebuild the Lessons menu with your new lesson included.

Edit Existing Lesson

This menu opens a Lesson file for editing. Lesson files can be user name and password protected so not all lessons may be edited by you. Once the lesson is open you can use the Previous and Next buttons to navigate to where you want to change the lesson.
At each step you can edit the lesson text in the text edit window and change the popup menus. The circuit will be shown with the remaining parts visible and the parts that will be used in this part of the lesson will be selected. You can select more or unselect some parts to change the lesson. When all your changes are complete click the Save button.

Popup Menus

When you click the right mouse button in the workspace or over a component a popup menu will appear. The menus will be different depending on where you click.

Over the Workspace

When you click the right mouse button over the workspace, but not over a component, the popup menu will allow you to Reset, Stop, Run, Pause/Continue, Step, and Stop Following. The first five items are just like the first four buttons in the tool bar.

Over Wires

When you click the right mouse button over a wire, the popup menu will allow you to open its Dialog, Duplicate it, Straighten it, and Delete it.

Over Car Icons

When you click the right mouse button over a Digital or Analog Car icon, the popup menu will allow you to open its Dialog, Duplicate it, View the Internal circuit while it is running, Delete it, enable Follow this car, and Stop following the car.

Over Integrated Circuits

When you click the right mouse button over an Integrated Circuit, the popup menu will allow you to open its Dialog, Duplicate it, View the Internal circuit while it is running, and Delete it.

Over Traffic Signals and Gates

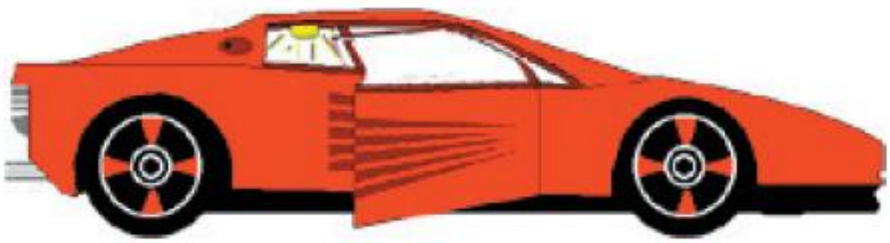
When you click the right mouse button over a Traffic Signal or Gate Icon, the popup menu will allow you to open its Dialog, Duplicate it, View the Internal circuit while it is running, and Delete it.

Other Components

When you click the right mouse button over any other component, the popup menu will allow you to open its Dialog, Duplicate it, and Delete it.

Chapter 2. Building a Simple Circuit

Following the steps in this chapter, you will create a digital circuit that turns on the dome light of a car when either the left front door or the right front door is open.



In MST Workshop, circuits generally proceed from left to right, starting with one or more sources on the left, functions in the middle, and displays on the right. This tutorial follows this principle.

Starting with Sources

In digital logic, everything is either true or false. When the car door is open, it is true; when closed, false. Therefore, our circuit will start with a switch for each of the two car doors. With the mouse you can set either switch to True or False.

1. Start MST Workshop.

The MST Workshop window appears, displaying the toolbar most recently **s**elected.

2. If the Digital toolbar is not displayed, select **Digital** from the drop down menu in the middle of the tool bar.

3. Place a switch on the work area by clicking once on the Switch button, then move the mouse down about an inch, click again to place the switch. This switch represents the left door of the car.

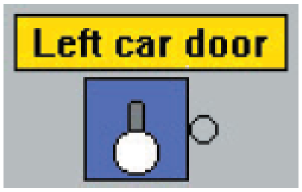
When you place a switch on the work area, the switch is off; that is, the white control point is at the bottom and the switch body is colored blue.

> Note: To reposition a component after placing it you must click and drag on its body. With the exception of wires, components cannot be moved by clicking on their connectors.

Note: Components like switches that can be operated with a mouse have a white dot indicating the control point. Objects cannot be moved by clicking on their control points.

4. Switch to the **Special** toolbar using the drop down menu.

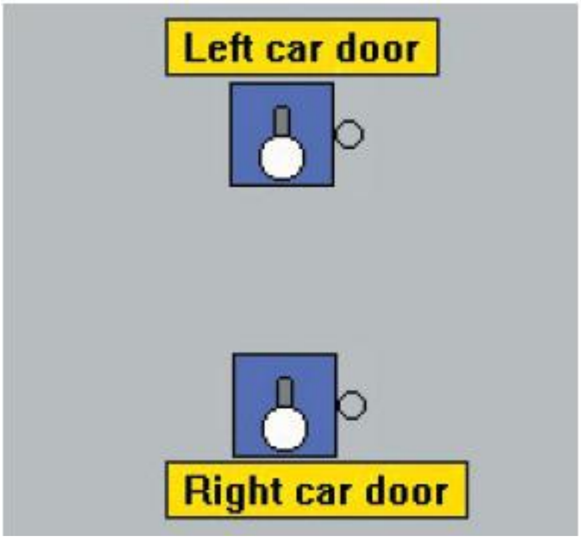
5. Click the Note button and type Left Car Door in the window. Click OK. Move the mouse close to the top of the switch that you just added, and click again. The result should look like this:



6. Place a second switch on the work area under the first switch. This switch represents the right door of the car.

7. Click the Note button and type Right Car Door in the window. Click OK. Move the mouse close to the bottom of the second switch, and click again. The result should look like this:

Your work area should look similar to this one:



8. Place an OR Gate about an inch to the right of the two switches.

Adding Displays

Displays, such as meters, LEDs, and strip chart recorders, must be added to a circuit so that you can read the output from the circuit.

Place an Light Emitting Diode just to the right of the OR Gate that you added.

Connecting Components

Sources, functions, and displays have connectors. Input connectors are generally on the left and output connectors are on the right, which follows our principle of left to right construction.

1. Use wires to connect the switches to the OR Gate:

    Wires let you connect components that are inflexible or in awkward positions. Note: Wires are moved by their connectors. If you attempt to move a wire by clicking and dragging the wire itself, you will bend the wire. To unbend the wire, click on the bend and straighten out the wire.

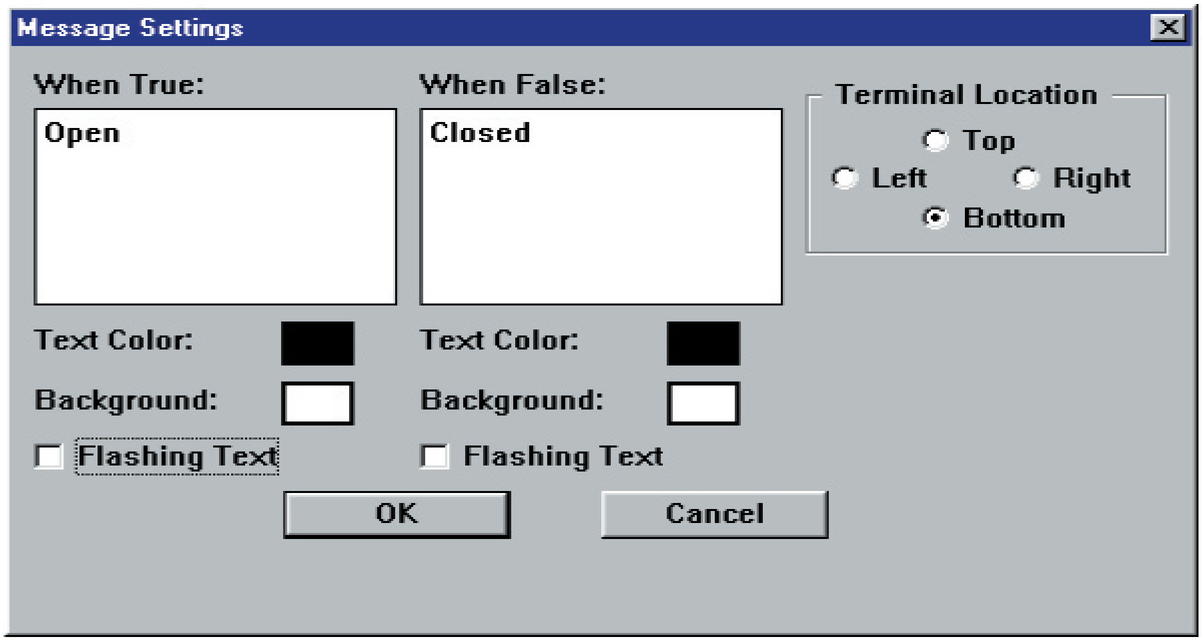
    a. Click on the wire button in the toolbar to place a wire on the work area.
    b. Move the mouse to the output of the top switch, click to start the wire.
    c. Move the mouse to the top input of the OR Gate, click to finish the connection.
    d. Place another wire between the output of the bottom switch and the bottom input of the OR Gate.

2. To complete the circuit connections, simply drag the Light Emitting Diode (LED) so that it's input connects to the output of the OR Gate.

Adding Messages

Messages are special readouts that give you textual information about the state of a component. Circuits can operate without messages, but messages make it easier to see what's going on with a circuit. For our circuit, we need to know whether the car door is open or closed. We'll add a message to do this:

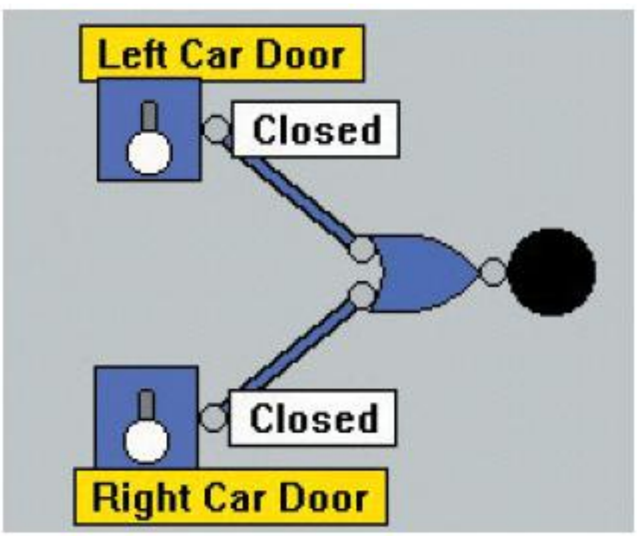
1. Click the Message button to access the Message window.
2. In the When True *box,* type Open. Type it right over the existing text.
3. Click the mouse in the When False box *and t*ype Closed.
4. Make **sure** that Left is selected for Terminal Location.



The Message Dialog

5. Click OK. Place the Message so that its terminal overlaps the upper switches terminal.
6. Repeat steps 1-5 for a second message, or use copy paste on the first message.
7. Place the second Message so that its terminal overlaps the terminal of the bottom switch.

Now you have completed your circuit, which should look like this:



Completed Tutorial

Operating the Circuit

Now that you have created a circuit with Sources, Functions, and Displays, you are ready to operate the circuit.

1. Move the switch on either door to the open position by dragging the white dot and moving it up. The LED turns red, indicating one of the two car doors is open and the dome light is on.
2. Move the switch settings to demonstrate the following door positions and their effect on the dome light:

 Both Closed
 Both Open
 Either door open

Saving the Circuit

Select Save in the File menu, type in a name for your circuit (for *example, practi*ce circuit, choose the location (correct folder), and click Save. MST Workshop saves circuits with the extension CCT.

Chapter 3. Analog Components

This chapter describes the components that are unique to the Analog toolbar and menu. Use these components to build analog circuits. Note that analog and digital components may be used within the same circuit.

Introduction

In an analog circuit, numerical data are represented by measurable physical variables, such as electrical signals. Unlike digital circuits which are either true or false, analog circuits can have inputs and outputs of any value. (MST Workshop imposes practical limits on its analog inputs of between -1000 and +1000.) Analog circuits can be used to solve problems about volume, distance, speed, and acceleration. For example, use an analog circuit to determine gas mileage given a particular distance and number of gallons used.

The analog components of MST Workshop's Analog toolbar are Sources (inputs), Functions that perform operations on the inputs, and Displays (readouts). The toolbar also includes standard buttons for operating a circuit. Sources, Functions, and Displays are described below.

Sources

Constant

The Constant lets you create a number of constant value. For example, to set the acceleration of a falling object, enter *32* for 32 feet per second per second.

> *To change the value of a Constant double click the Constant and type a value in the Value field. Click OK.*

If you want to change a constant without opening the dialog or while the program is executing you can set the Adjustable check box in the constant's settings dialog. Then you can drag the white dot around the Constant to change its value.

Initially, each revolution is a change of 1.00. To change the amount that the constant changes during each revolution, use the slider control in the dialog. You can also change the effect of a revolution by clicking on the little boxes on the dial. The box with a plus (+) increases the effect by a factor of 10, the box with a minus (-) decreases the effect by a factor of 10.

To select the location of the output connector click Up, Down, Left, or Right.

The Constant dialog also allows you to select from a list of common constants.

Slider

The Slider lets you interactively change a value between preset limits. Sliders can be either tall or wide and are adjustable in size. You can set each slider's upper and lower limits. There is an option to set a step size so that the slider's output will change incrementally instead of continuously.

> *To change the Slider's setup double click the Slider, change the dialog and click OK.*

Geometry

The Geometry component is used like a slider. It can have multiple controls (knobs that you can move) and works in both X and Y directions. It can show the controls with or without the circle and controls can be connected together with a line if desired. Use the geometry to create a line and from its outputs you can calculate its slope or length.

Keypad

The Keypad lets you create one or more constants from the keypad or keyboard. From its dialog select how many constants you will need.

> *To enter the constants, select the Keypad component, click the Enable Keypad button at the bottom, click one of the constants in the list, use the keypad or keyboard to type digits, minus, and decimal point.*
>
> *Delete one character at time by with the Delete key or the Backspace key.*
>
> *Change the sign from positive or negative at any time by typing "-" key.*
>
> *Type "E" to add an exponent. Then type the exponent digits. After the letter "E" has been typed, the "-" key will change the sign of the exponent.*

Displays

Analog circuits require Displays for reading the output of the circuit. Analog displays include Meters, Strip Chart Recorders, and X Y Chart Plotters.

Digital Meter

The Digital Meter reads the input and displays that value at a given instant. The digital meter can display values showing up to 16 digits. You can also select the Scientific notation to show extremely large or small values using an exponent. Note: Select the Display and type a digit to quickly set the number of places to display.

Analog Meter

The Analog Meter reads the input and displays that value by deflecting a needle. The analog meter gives a better visual indication of rapidly changing values.

Segmented Display

The Segmented Display shows an Analog value by lighting a series of LED segments. Low and high limit values can be set in the Dialog to allow any range of values to be displayed.

> *To change a display low or high settings, double-click the display to access the Segmented Display Settings dialog box.*
> *To change the size of the display select Large, Medium, or Small.*
> *To change the orientation select Tall or Wide*

Strip Chart Recorder

The Strip Chart Recorder displays the values of from one to eight inputs over a given time period. For example, use the Strip Chart Recorder to watch the output of an integrator as it changes over time.

> *To change a chart setting, double-click the chart to access the Strip Chart Settings dialog box.*
>
> *To change the vertical scale of the chart, enter values for the Top Value and Bottom Value.* If you expect to plot data between 25 and 75 set the Top Value to 75 and the Bottom Value to 25.
>
> *To change the size of the chart select the Height or Width popups.*
>
> *To change the speed of the chart's movement, enter a value in the Chart Speed field. Also select the Hours, Minutes or Seconds in the Units popup.*
> *To change the number of inputs (pens), select the number in the Pens field. If the OK button goes dim you will need to increase the size of the vertical height to accommodate all the inputs.*

To change a pen's color, click it's corresponding color patch and select a new color from the color picker dialog. The eight color patches show the default colors of the eight available pins. The first row is pens 1-4; the second row, pens 5-8, etc.

To make the chart paper move to the left, select the Reverse checkbox.

Sonograph Recorder

The Sonograph Recorder displays the values of an array over a given time period. Each value in the array gets a horizontal band in the display. The color in the band is chosen from the set of colors chosen in the dialog. For example if there are 4 colors chosen for a range of 0 to 100, the first color will be plotted when the array value is between 0 and 25, the second color when between 25 and 50 etc. To change a chart setting, double-click the chart to access the dialog box. To change the input range of the chart, enter values for the Top Value and Bottom Value. If you expect to plot data between 25 and 75 set the Top Value to 75 and the Bottom Value to 25. To change the size of the chart select the size popup. To change the speed of the chart's movement, enter a value in the Chart Speed field. A speed of 10.0 is the maximum speed for the chart. To enter the number of colors, select the number in the Pens field. To change a color, click it's corresponding color patch and select a new color from the color picker dialog. The eight color patches show the default colors of the eight available pins.

X Y Plotter

The X Y Plotter plots the relationship between two inputs at once. For example, it can be used to display miles per gallon for your car at different speeds. The XY Plotter can have from one to eight pens. Each pen has a Y input on the left that moves the pen vertically and an X input on the bottom that moves the pen horizontally.

To change X Y Plotter settings, double-click the plotter.

To change both the vertical and horizontal scale, enter a value in the Chart Range field. The chart range should be greater than the expected value of the inputs. Zero is in the center so values that are negative will be plotted to the left or bottom.

To change the size of the chart select Large, Medium, or Small.

To change the pen color, click the color patch and select a new color from the color picker dialog.

Scatter Chart

The Scatter Chart plots up to eight X Y inputs at once. The X inputs are on the left side of the chart and start at the top. The Y inputs are on the bottom of the chart and start at the left. The chart can be drawn with lines connecting the symbols if desired. For example, the Scatter Chart can be used to plot the x y position of several balls on a billiard table.

To change Scatter Chart *settings, double-click the chart.*

To change the scale, enter a value in the Chart Scale field. The chart scale should be greater than the expected value of the X and Y inputs.

To change the size of the chart select Large, Medium, or Small.

To change the number of symbols plotted at once set the Symbols field.

To show Symbols, Lines, or Lines and Symbols use the Drawing field.

To change the symbol internal color, click the color patch and select a new color from the color picker dialog.

Pie Chart

The Pie Chart plots up to eight inputs at once. The inputs are shown as slices of a pie. The larger an input is in relation to the sum of all inputs, the larger the slice. To change Pie Chart settings, double-click the chart. To change the size of the chart select Large, Medium, or Small. To change the number of inputs set the Slices field. To change the color of a slice, click the color patch and select a new color from the color picker dialog.

To change Pie Chart *settings, double-click the chart.*

To change the size of the chart select Large, Medium, or Small.

To change the number of inputs plotted at once set the Slices field.

Analog Message Display

The analog Message Display can display a series of messages. Each message will be displayed when the input is between values specified for that message. Messages can be in colored text with colored backgrounds. Use the Message Display to show a word or short phrase that is associated with a range of values. The Message Display is preloaded with the days of the week and months of the year. You can add your own table of values and messages with the built in editor. The output of the Message Display can be used to move short messages (8 characters or less) to other components like the any of the 2D shapes. Just connect a wire from the output of the Message Display to the Label input pin on the 2D shape.

Functions

Functions are components that take the values of one or more inputs and generate an output. For example, to determine distance driven, a multiplier function could multiply a car's speed by number of hours the car was driven to show miles driven.

Note: In MST Workshop many functions can take scalar inputs, like from a constant, or arrays. When an array is connected to a function the output of the function may change to an appropriate array. For example, input scalars such as 2 and 3 into a multiplier and the output will be 6. Input an array [1, 2, 3] and a scalar 5 the output will be an array [5, 10, 15]. Input two arrays that are the same size and the output will be a corresponding array of the products. For example [1, 3, 5] times [2, 4, 6] will be [2, 12, 30]. (See additional note for Multiplier)

Adder

Adds its two inputs. Inputs are located on the left side of the Adder.

Subtracter

Subtracts the bottom left input from the top left input.

Multiplier

Use this to multiply two variables. If you need to multiply a variable by a constant, and the constant is between zero and one, use the Attenuator below.
Note: If both inputs are arrays and the top input has 1 dimension more than the bottom input, the result will be an array value that is the product of the two arrays when treated as Matrices.

Divider

Use this to divide two variables. The upper input is the numerator and the lower input is the denominator.

Negative (-)

This analog function returns the negative of the input. The output is the same value as the input but with the opposite sign.

Constant Factor

This function multiplies the input by a constant. For example, dragging an object on a surface causes friction, thus reducing it's velocity. Use an Constant Factor to set the effect of friction on the velocity of an object. If you know that a rock traveling across pavement has a coefficient of friction of 0.6, enter 0.6 in the Constant Factor.

> *To change the value of a Constant Factor, drag the white dot around the circle, or double click the Constant Factor and type a value in the Settings field. Click OK.*

You can select the orientation of the Constant Factor. The input will be marked with a black triangle forming an arrow into the Constant Factor. Because Constant Factors are often used to feedback a voltage from the output of a component to the input, you can set the input on the right and output on the left.

You can override the limit of 0.0 to 1.0 by checking Extended Range. This will allow you to multiply a signal by a positive or negative number.

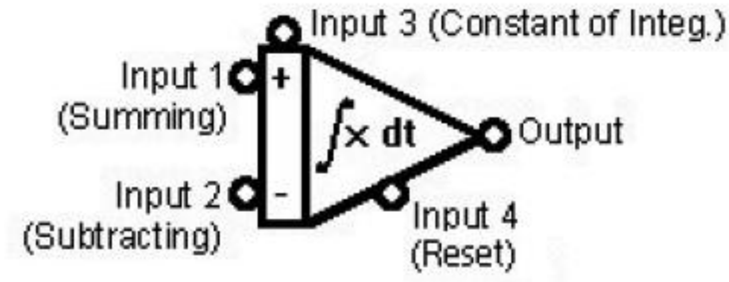
Constant Offset

This function offsets the input value by a constant. It is like the Constant Factor above except that it adds a constant value instead of multiplying by a constant value.

Integrator

The integrator is a component that will sum its inputs and integrate the sum over time. The integrator is a complex component that is the workhorse of an analog computer. For example, if you know the acceleration of an object, you can integrate that value to find the velocity of the object. If you then integrate the velocity, you can determine the position of the object. A full discussion of the Integral is beyond the scope of this manual; however, we suggest you consult a mathematical reference book for detailed information.

> Input 1 (upper left) adds the number you input to the sum, for example, if you input a constant value of 2, 2 is added to the output every second.
> Input 2 (lower left) subtracts the number you input from the sum.
> Input 3 (top) inputs a constant of integration (starting point for the integration).
> Input 4 (bottom) is a control input that restarts the integrator at the constant of integration (input 3) when a True value is connected to it.



An Integrator

Double-click the Integrator to access the Parameter dialog box.

To prevent the function from spinning out of control, check the Stabilizer option and click OK. Instabilities can occur in closed-loop functions, for example, a swinging pendulum.

To invert the output, check Invert Output and click OK.

Differentiator

This component takes the derivative of its input over time. It is the opposite of the integrator. The differentiator is useful in PID (Proportional Integral Differential) control loops. Use the Differentiator when you need to know the rate of change of a value. You can also use it to calculate the acceleration if you know the velocity, or you can use it to calculate the velocity when you know the position of an object.

Latch

The Analog Latch passes analog inputs on the left to the outputs on the right or holds previous values at those outputs. It can run in Strobe mode where inputs are captured when the bottom input goes from False to True or in Gate mode where inputs are passed when the bottom input is True and captured when the bottom input is False. Output values are saved with the file until it is loaded again.

Chapter 4. Digital Components

This chapter describes the components that are unique to the Digital toolbar and menu. It also describes how the Advanced Logic function enables you to build your own digital components, and how to build an integrated circuit. Note that analog and digital components can be used within the same circuit.

Introduction

In a digital circuit, calculations and logical operations are performed using the binary number system; that is, ones and zeros, which are the equivalent of true and false, respectively. Digital circuits can be used to solve problems about states of components, either open or closed. For example, we created a digital circuit in Chapter 2 to represent the on-off state of the dome light of a car when the doors are open or closed.

The Digital toolbar includes Sources (inputs), Functions that perform operations on the inputs, and Displays (readouts).

Sources

True

This input provides a source of 5.0, a value that is consistent with TTL (Transistor-Transistor Logic).

False

This input provides a source of 0.0.

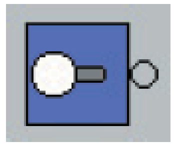
Switch

A switch is a user-controlled input in a circuit. The Switch outputs either a true or false based on the user selection. Switches can be single output or have multiple outputs.

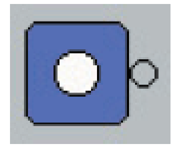
Double-click a switch on the workspace to access the following Switch Settings dialog.

Select Toggle Switch to create a switch that is operated by dragging the white spot on the switch with the mouse.

Select Push Button to create a switch that is turned on by clicking and remains on only as long as the mouse button is held down.



Toggle Switch



Push Button Switch

To operate the switch from the keyboard, enter the key's name in the Keyboard Toggle field. Switches can be used inside integrated circuits and in template circuits. These switches can be changed by pressing the key representing the toggle when the integrated circuit is present or the icon is on the screen.

Displays

Displays are components that take input values of one or more inputs and in some way display the value. For example, a Light Emitting Diode can be used to display a true value by lighting and stay dark for a false value.

Light Emitting Diode (LED)

The Light Emitting Diode lights up when its input is true and is dark otherwise. Light Emitting Diodes can have one to eight inputs. Note: Select the LED and type a digit to quickly set the number of LED's in the package.

Play Sound

The Play Sound component plays a sound when its input goes from false to true. It will play the sound once each time the input becomes true. Only a few sound files are included with MST Workshop. You can find sound files on your computer or on the internet.

If you are running Microsoft Windows search for *.wav files and move them to the Sounds directory. You can record your own sounds by selecting Start >> Programs >> Accessories >> Entertainment >> Sound Recorder.

If you are running Apple Macintosh, search for snd resources and use ResEdit to move them into MST Workshop's resources. You can record your own sounds by selecting Apple Menu >> Extensions >> Monitor & Sound then selecting the Alert button in the Dialog.

Logic Analyzer

The Logic Analyzer displays up to eight inputs at once. The Logic Analyzer is similar to the Strip Chart Recorder but lines are plotted separately and do not overlap.

Digital Message Display

The digital Message Display can display two messages. One message will be displayed when the input is true, a different message can be displayed when the input is false. Messages can be in colored text with colored backgrounds. The output of the Message Display can be used to move short messages (8 characters or less) to other components like the any of the 2D shapes. Just connect a wire from the output of the Message Display to the Label input pin on the 2D shape.

Functions

Functions are components that take the values of one or more inputs and generate an output. For example, an AND Gate can be used to represent a CD-player that plays only if it has a CD loaded and the Play button is pressed.

Note: Like analog functions, the digital functions can have array inputs and create arrays for outputs.

AND Gate

An AND Gate is a component whose output is true if and only if all inputs are true. By selecting Invert Output, the AND gate can be changed into a NAND gate. To show that it is a NAND gate the output terminal will be shown as a bold circle. AND Gates can be started in a True state so that flip flop circuits will be stable. When this option is used the AND Gate symbol will have a small astrerisk inside. AND Gates can have 2 or 3 inputs. Example: A light comes on only if it is nighttime and the switch is turned on.

OR Gate

An OR gate is a component whose output is true if any input is true. By selecting Invert Output, the OR gate can be changed into a NOR gate. To show that it is a NOR gate, the output terminal will be shown as a bold circle. OR Gates can be started in a True state so that flip flop circuits will be stable. When this option is used the OR Gate symbol will have a small astrerisk inside. OR Gates can have 2 or 3 inputs. Example: A car's dome light comes on if either of the two front doors is open.

Exclusive OR Gate (XOR Gate)

The output of an XOR Gate is true if and only if one input is true and the other is false. For example, a stairwell light is on when the switches at the bottom and top of the stairs are positioned so that one is up and the other is down.

Inverter

In an Inverter, the output is true if the input is false, and vice-versa. For example, if you can buy only a daytime sensor for the light that you wish to come on at nighttime, use the inverter to change the daytime sensor to a nighttime sensor.

One Shot (Monostable Multivibrator)

This One Shot outputs a pulse whenever the input changes from false to true. One Shots can output either a single short pulse or a square wave of specified duration. Once the output is activated the input is ignored until the output falls low again. Change the mode and pulse width by double-clicking on the device. Selecting Retriggerable means that the time will be extended each time if the input goes from false to true again before the time has elapsed.

Advanced Logic

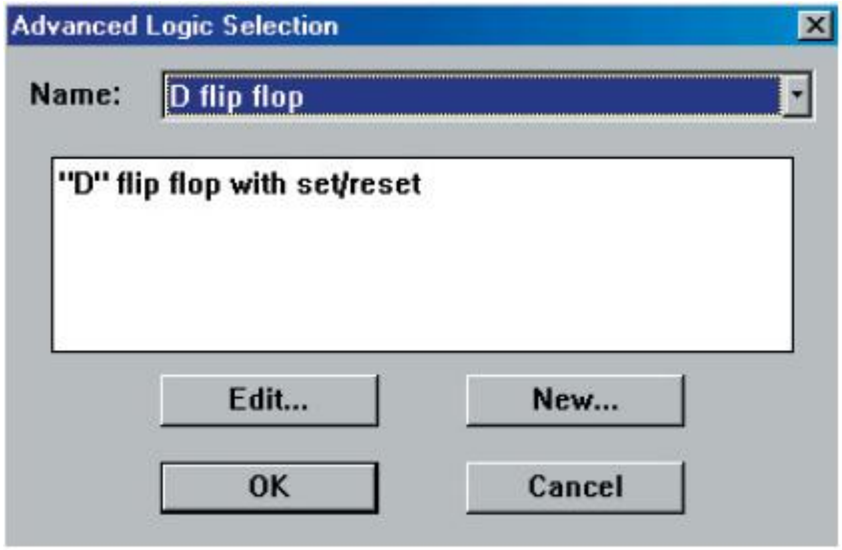
The Advanced Logic function lets you create a new digital function using a table of logic, called a **truth table**. It includes four ready-made advanced logic functions (RS Latch, RS flip-flop, D flip-flop, and J-K flip-flop described below), which you can use in your circuits.

Advanced logic works in three phases:

1. The program gathers the values at all the input pins of the icon in the workspace.
2. It scans down the logic table, row by row, comparing the input pin values with the entries in the table until it finds a row that matches.
3. It takes the output specified on that row and sends each value to the specified output pin.

You can watch this process by right clicking on any advanced logic icon and selecting view internal from the popup menu.

To access the Advanced Logic Settings dialog, click Advanced Logic on the toolbar.

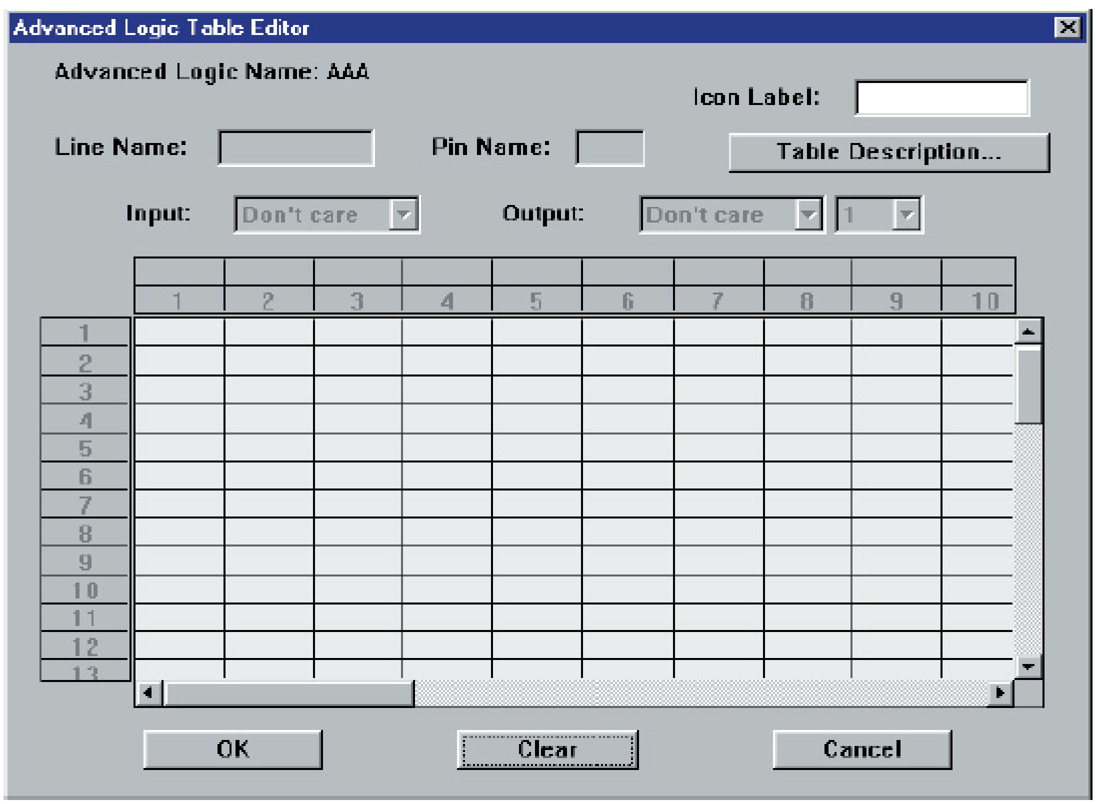


Advanced Logic Selection Dialog

To select an existing function, highlight it's name in the Name field and click OK. The function you selected appears on the workspace.

To create a new function, click New. A new Advanced Logic Table appears.

To edit an existing function, click Edit. The Advanced logic table for the existing function appears:



The Advanced Logic Table

The **Icon Label** is a short label that displays above the icon when it is placed on the workspace.

Pins are numbered starting with *1. To make the Icon more understandable you can change* **Pin names** *by selecting the pin number in the second row of the table and entering a new name in the Pin Name field.*

Rows are numbered also. To help you remember the purpose of each row you can change the **Line Names** *by selecting the row number on the left and entering a short name in the Line Name field.*

Each column represents a pin; each row represents a proposition. Each cell can describe the state of one pin in that proposition.

States of Input Pins

- Don't Care (input signal can be either true or false to match)
- Low (input signal must be false or less than 2 volts to match)
- High (input signal must be true or greater than 2 volts to match)
- Rising Edge (input signal must have just changed from false to true to match)
- Falling Edge (input signal must have just changed from true to false to match)

States of Output Pins

- Don't Care (the pin is not an output pin)
- Low (output signal will be set to false when this row matches)
- High (output signal becomes true when this row matches)
- Internal (output signal becomes the current value of a pin that you specify)
- Int Rev (output signal becomes the opposite of a pin that you specify)

To enter a value in a cell, click the cell and select the desired state from the Input and/or Output popup menus.

Click Save when you have finished editing the logic table.

Enter the Name for the Table. The next time you use the Advanced Logic function, your new function appears on the list.

Advanced Logic Circuits Included in MST Workshop

MST Workshop includes three advanced logic tables: the RS Latch, RS flip-flop and the D flip-flop. Flip-flops and latches are circuits capable of assuming and retaining either of two stable states, true or false.

The RS Latch

An RS Latch has two inputs: Set (S) and Reset (R). A pulse to the S input sets the output to true; a pulse to the R input resets the output to false.

Essentially, the RS Latch represents the smallest unit of memory needed to remember a single state. It is useful for designing circuits that need to maintain a state, such a vending machine remembering that you inserted a quarter.

The RS Flip-Flop

In an RS flip-flop, a true value at the value at the P (preset) input sets the output to true at any time. A true value to the Cl (Clear) input sets the output to false at any time. For the P or Cl inputs to work the other input must be false.

There are also R (reset) and S (set) inputs that can set the output to false or true respectively at the instant that the Ck (clock) input changes from false to true. This transition is called the positive edge (+Edge). Like the P and CL inputs, the R and S inputs work only when one is true and the other is false. Use The RS Flip-Flop to capture data at a specific instant.

The D Flip-Flop

In a D flip-flop, the value at the D (data) input is transferred to the output at the instant that the C (clock) input changes from false to true. This transition is called the Rising Edge. Use it to capture data at a specific time.

A D flip-flop is used in circuits that require an action at a particular instant, as in an automatic garage door opener. The C (clock) input is connected to the open-close button, and the D (data) input is used to decide if the door should be opened or closed when the button is pressed. When the button is pressed and released, the door continues to open or close because the flip-flop remembers the data value until the door is fully open or closed.

The J-K Flip-Flop

In a J-K flip-flop, the values at the J and K inputs are transferred to the output at the instant that the C (clock) input changes from false to true. This transition is called the Rising Edge. If the J input is true the output will become true. If the K input is true the output will become false. If both J and K are true the output will toggle. If both J and K are false the output will not change.

Chapter 5. Simulation Components

This chapter describes the components that are unique to the Simulation toolbar and menu. Use these components to build circuits that will simulate cars driving on roads or through a field of barriers. You can also build other circuits that your cars can interact with such as traffic signals and gates. There are also stand alone components that allow you to test your skills by simulating and elevator, a beaker of water on a hotplate, and a stepper motor.

Introduction

Most simulation components come in two parts: a template and an icon. The template has inputs and output. The template is used in a circuit with analog and digital components. Once this circuit is saved it can be opened as an icon. When the icon is placed on the screen it will respond to its environment as you have programmed the circuit containing its template. Each Icon will perform independently even if it uses the same circuit. In this way a single Digital Car or Traffic light circuit can be programmed and saved then used as an icon as many times as desired to create a workspace with many cars and traffic lights.

Digital Car Template and Icon

The Digital Car simulates a simple car that can go forward and backward and turn left or right. It has sensors to detect the presence of a road, other cars, traffic signals and barriers. Use the Digital Car Template to indicate exactly what the car does when its sensors detect something. For example, the car can be wired to hit the brakes when it detects another car in front of it or turn left when it detects a barrier.

Programming a Digital Car to Follow a Road

1. Start with the Digital Car template ( ) on the workspace.

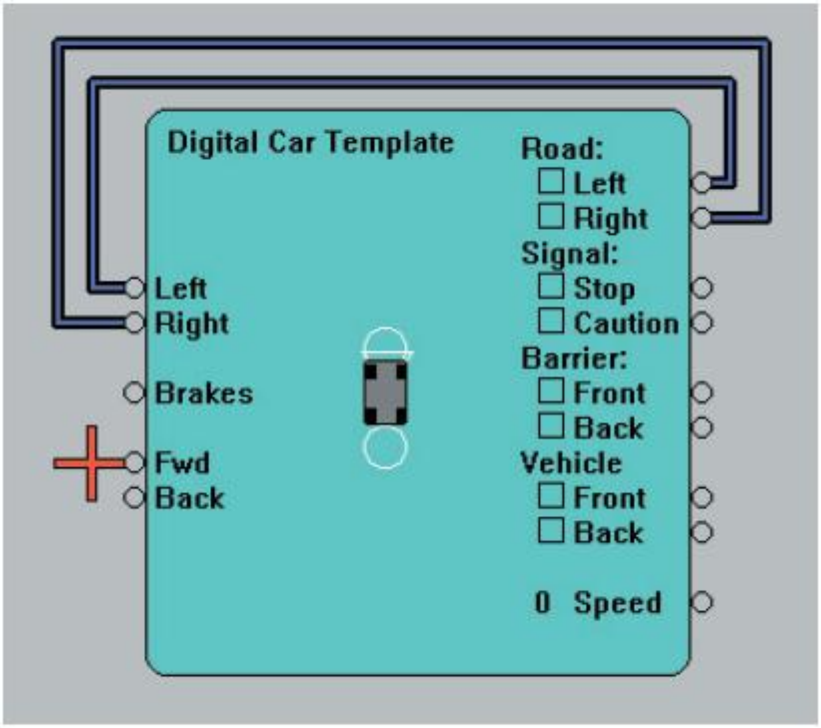
   The Digital Car has two road sensors in front of the car. One is near the left front wheel and the other near the right front wheel. They show up in the simulation as the corners of a white triangle. As the car's speed increases the triangle extends forward to better anticipate the road ahead. When the car is completely on a road both left and right road sensors will give a true signal. If the car wanders off the road to the right the right sensor will go false.

2. Connect the left road sensor to the to the left turn control and the right road sensor to the right road control as shown below.

If left and right turn controls are both true or both false the car will go straight. If the one sensor goes off the road the car will turn away from that sensor and follow the road.

3. To make the car move forward, connect a True to the Fwd control.

2. Save the circuit as *Road Car 1*. You will use it in a few pages.



Template of a Digital Car That Follows a Road

Note: To view the template circuit when the car icon is on the workspace, right click on the icon and select View Internal.

Analog Car Template and Icon

The Analog Car simulates a simple car that can go forward and backward and turn left or right similar to the Digital Car. It has sensors to detect the presence of a road, other cars, traffic signals and barriers also similar to the Digital Car.

The difference is that the Analog car has analog inputs and outputs. The turn and drive inputs will take from minus five to plus five volts, while the brake input will take from zero to plus five volts. The extent of the turning, braking and driving will depend on the voltage. Similarly the outputs are analog. The road sensor indicates how far from the center of the road and the other sensors indicate how close the other object is to the car.

Use the Analog Car Template to indicate exactly what the car does when its sensors detect something. For example, the car can be wired to apply the brakes as it approaches another car or turn hard left when it detects a barrier at mid range.

Programming an Analog Car to Follow a Road

1. Start with the Analog Car template on the workspace.

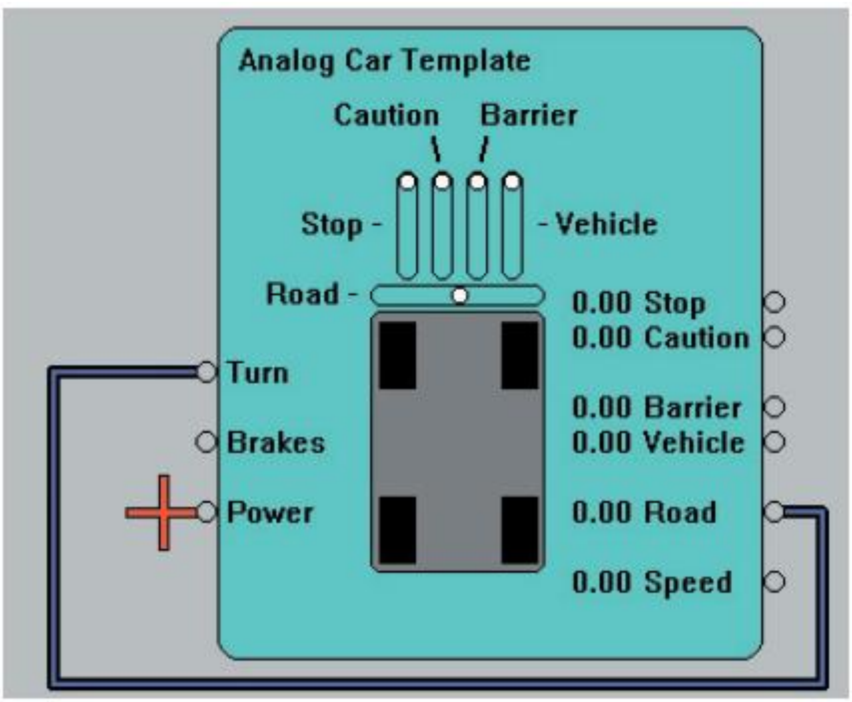
The Analog Car has two road sensors in front of the car. One is near the left front wheel and the other near the right front wheel. They show up in the simulation as the corners of a white triangle. As the car's speed increases the triangle extends forward to better anticipate the road ahead. When the car is centered on a road the output is zero volts. If the road turns to the right the sensor will increase positively and negatively if the road turns to the left.

2. Connect the road sensor to the turn input as shown below.

If the road turns to the left or right the wheels will be turned to bring the car back on the road.

3. To make the car move forward, connect a True to the Drive control.

4. Save the circuit as *Road Car 2*. You will use it in a few pages.



Template of an Analog Car That Follows a Road

Note: To view the template circuit when the car icon is on the workspace, right click on the icon and select View Internal.

Signal Light Template and Icon

The Signal Light simulates a traffic signal, using indicators that can be individually set to red, yellow or green. These indicators point in four directions: North, South, East and West. You could build a timer-based signal to change the colors at time intervals to simulate an old fashioned traffic light. The template also has sensors that can detect approaching cars. These sensors can be used to build a modern signal that changes the light colors based on actual traffic patterns.

Creating a Time Based Traffic Signal

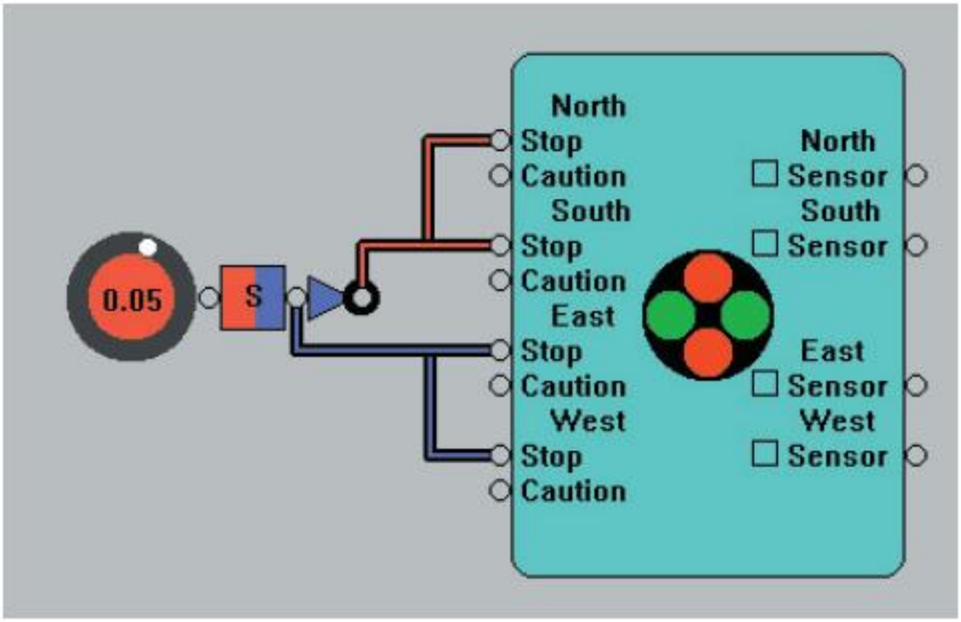
1. Place a Signal Light Template on the workspace and build the circuit below.

   The Constant of 0.05 connected to a Pulse Generator will cause the Traffic Signal to change the lights every 20 seconds. To test the light hit the Run button. If it works save the circuit as “Simple Traffic Light”.

2. Close the workspace.

3. Click on the Signal Light Icon.

4. Place the Icon anywhere on the workspace and press the Run button. The Icon should change colors every 20 seconds.



Template of a signal light based on time.

Note: To view the template circuit when the traffic signal icon is on the workspace, right click on the icon and select View Internal.

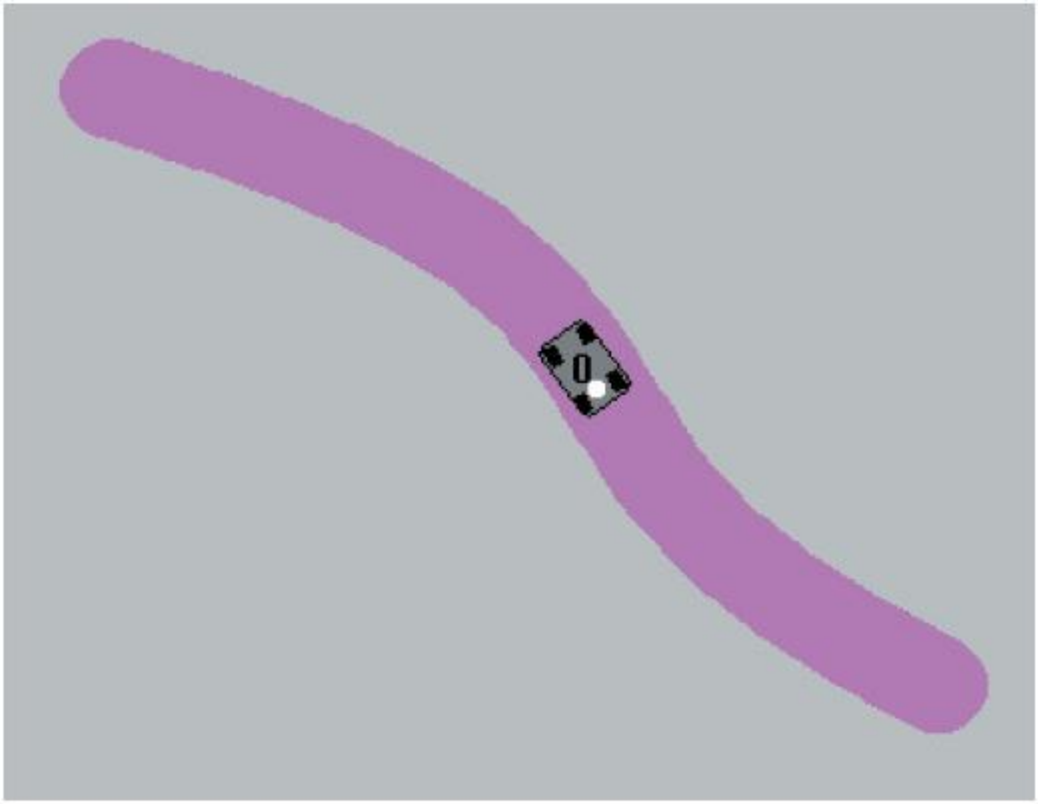
Gate Template and Icon

The Gate Template and Icon can be used to stop vehicles on a road by being placed so that the Icon's arm reaches across a road. Another way to use the gate when no roads are present is to block vehicles by placing the gate icon between two barriers. The Gate teaches simple digital control logic. The Template has just two inputs to command the arm to Raise or Lower. If used without feedback, the arm will Raise and Lower past the normal stopping points. The template has two outputs that indicate when the arm is Up or Down and these outputs should be used to stop the arm's motion at the correct positions. Additionally the gate has two sensors that indicate when a vehicle is to the Left or Right of the arm. These sensor outputs can be used to decide when to lift or lower the arm.

Typically a vehicle approaches from the left and waits for the gate to lift. When the vehicle crosses to the right sensor the gate can be closed. What can you do to prevent the arm from closing on a vehicle? What will your gate circuit do if two cars cue up at once? What happens if the car drives away before the gate is fully closed?

Road

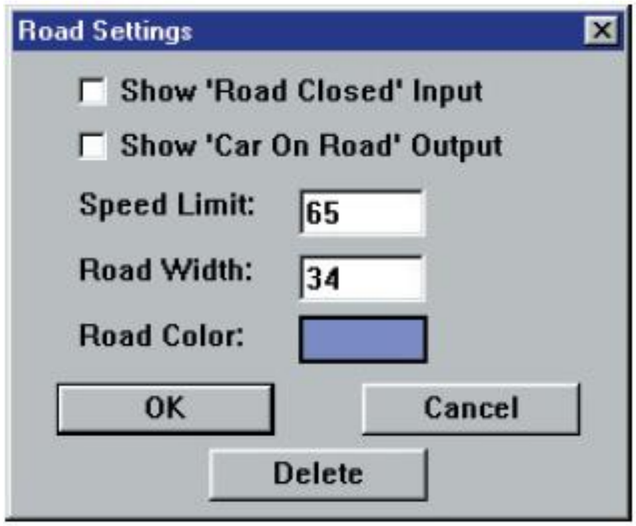
You can use cars without roads but, roads make the simulation much more interesting. Select the Road button to place a road. Click once in the upper left of the workspace then again in the lower right. Roads are similar to wires in the way they are placed and moved. To select a road and see its control points click the mouse at either end. To add new control points click anywhere along the road and drag. Now put some gentle bends in the road as shown:



Workspace with a Road and Car

To place a car on the road, click the Digital Car Icon button or the Analog Car Icon Button then hit OK. When you have created more cars you will need to select the desired car from the popup menu. Move the mouse to the upper left side of the road and click the mouse to place the car. You can move the car by dragging it with the mouse or turn it by clicking on the white spot in the front of the car and dragging it in a circle. Hit the run button and the car should follow the road then drive off the end.

Each road can have up to eight control points. To remove a control point move it so that it is on a straight line between the points on either side of it and when the circle you are dragging disappears release the mouse.



Dialog of Road settings

You can also enable an input on a road that will disable the road when the input is connected to a true value. To enable this feature check the Show Road Closed input in the dialog. When a road is closed while a car is on it the car will continue along without knowing that the road is closed but other cars will not sense the road at all. This feature can be used to create alternate paths that can be enabled or disabled like a switch yard of railroad tracks.

Roads can also have an output that will become true when a car is on the road. To enable this feature check the Show Car On Road Output in the dialog. This output can be used to stop a timer when the first car reaches a road with this output enabled. When there are no roads on the workspace the speed limit for cars is the same everywhere.

When roads are present that all changes. When a car is on the road, the car's speed limit will be the speed limit of the road. You can set the road's speed limit in the road's dialog to a number between 10 and 65. When a car is not on any road, the car's speed limit is reduced. This makes it more important to stay on a road when you are racing other cars.
The width of the road can be set in the road's dialog. Roads can be as narrow as 20 pixels or as wide as 70 pixels. When more complex roads are needed, connect roads together by overlapping the end points.

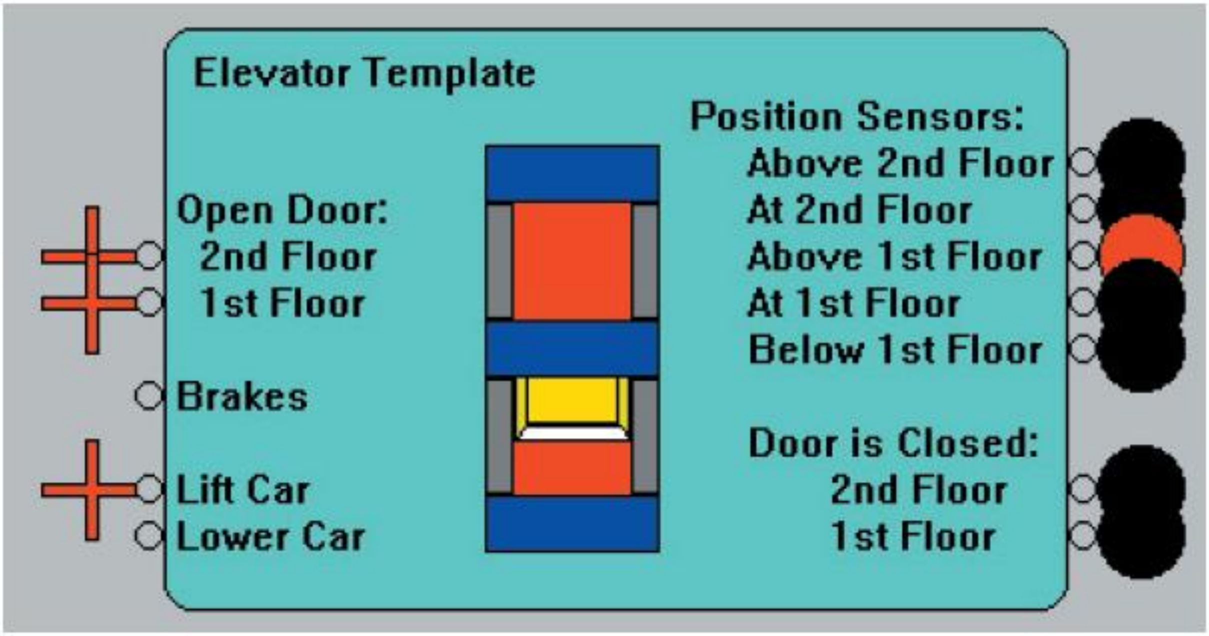
Barrier

Barriers are used to create an obstacle course. Place Barriers of various sizes and colors on the workspace, then test your Digital Car's ability to avoid the Barriers. You can set Barriers to send cars to the upper left corner of the screen when the car crashes into the barrier without stopping.

Elevator Template

The Elevator Template stands alone; it does not have an icon. It is used to simulate the function of an elevator that services a 2-story building. The elevator can go up or down. Each floor has doors that can open or close and a sensor to indicate when the doors are closed.

To get an idea of how the Elevator works, place Trues on the 2 Open Door inputs and a third on the Lift input. Also place LEDs on all the outputs. Move the True away from the Lift input and watch the elevator slowly slip down unless you put a True on the Brakes input. The Door Closed outputs are provided so that you can wait until the doors are completely closed before moving the elevator.



Elevator Template Demo Circuit

Water Beaker Template

The Water Beaker Template stands alone; it does not have an icon. It is used to simulate the function of a beaker of water on a hot plate. The beaker contains 500 ml of water at room temperature (25 degrees centigrade). The hot plate has a coil that will heat to about 600 degrees when 10 volts is connected to the Heater input. The coil heats the plate to about 200 degrees maximum. The water temperature is output on the right. You can drop a cube of ice into the water to see how well your circuit can recover as the ice melts. When the ice has completely melted a fresh cube will be ready to drop.

Using the Water Beaker Template

1. Hold down the button connected to the Heater input for a while and the water temperature will rise. Release it and the temperature will drop after a while as the hotplate cools.
2. Press the button connected to the Drop Ice input and the ice will drop and cool the water while it is melting.

Now build a simple circuit to control the Heater based on the difference between the actual Water Temp output and a desired water temperature (use a Constant and a subtracter). When the temperature is below the desired temperature apply a voltage to the Heater input.

Stepper Motor Template

The Stepper Motor Template stands alone; it does not have an icon. It is used to simulate the function of a four-pole stepper motor. The motor has four inputs, one for each electromagnetic coil.

> *To make the motor's shaft rotate clockwise put a true value on coil 1, then move it to coil 2, then coil 3 then coil 4, then back to coil 1 and repeat the process. Each time the true value is moved to the next coil the shaft rotates one sixteenth of a revolution.*
>
> *To make the shaft rotate one revolution requires sixteen steps.*
>
> *By putting true values on the coils in the reverse order (1, 4, 3, 2, 1, etc.) the shaft will turn counter clockwise. You can also make the motor half step by a slightly more complex procedure. Starting with a true on coil 1 add a true to coil 2. The shaft will step half way between the coils. Then remove the true on coil 1 and the shaft will finish the step. Likewise add a true to coil 2 then remove the true from coil 1 and it will half step again.*

Chapter 6. Communication Components

This chapter describes the components that are unique to the Communication toolbar and menu.

Introduction

The Communication Toolbar contains components needed to communicate between two workspaces or two circuits on a workspace, write data to files on your hard disk, and communicate through your computer's parallel port.

Transmitter and Receiver

The Transmitter lets you send an analog or digital value to a Receiver. This allows you to send information to another circuit in another workspace, an integrated circuit, an analog or digital car, a traffic signal or gate. The transmitter has a Value input that can take either an analog or digital value. When the Send input is true the transmitter is turned on and will send the value to any Receivers on the same channel.
The Receiver picks up an analog or digital value from a Transmitter set to the same channel. When the receiver receives a signal it is output on the Value terminal and the Ready terminal becomes true.

Global Position

The Global Position makes the X, Y position and orientation of an Analog or Digital Car, Traffic Signal or Gate. You can use the Global Position and two transmitters to broadcast the coordinates of a car. Another car could have a circuit with a two receivers and a Global Position wired to calculate the direction to steer the car so that it follows the one broadcasting its coordinates.

Data Logger

The Data Logger saves values to an ASCII text file on your hard disk. The Data Logger can save from 1 to 8 analog values on each line of the file separated by a comma, tab, or space character. To provide the best insurance against data loss the Data Logger saves data by opening the file, writing the line, then closing the file. Use the Delete File button in the Data Logger dialog to remove data already logged to the file.

Data Fetch

The Data Fetch reads values from an ASCII text file on your hard disk. The Data Fetch can read from 1 to 8 analog values on each line of the file separated by a comma, tab, or space character. Each time the Open input goes from false to true, the Data Fetch opens the file and starts reading from the beginning. Each time the Read input goes true one line of the file is read. The Open output indicates that the file was opened successfully. The Ready output indicates the values on the 1 to 8 analog outputs are ready.

Data to Array

The Data to Array, like Data Fetch, reads a data file (created by the Data Logger) into an array.

AES Cypher

The AES Cypher is used to encrypt or decrypt an array using a 256 bit key. The array can be any type and size. The key can be one parameter or the same number of parameters as the input so each will correspond. The key is also an array. The first 32 bytes will be used as a key. Use the same key for encryption to convert from plain data to cypher data that you use for decryption to retrieve the clear data from the cypher data.

Keyboard

The Keyboard reacts to the computer's keyboard. Each time a key is typed the eight bit ASCII code is read and used to set the eight data terminals. As soon as the data is ready the Valid output is set to true then immediately set to false.

Display

The Display waits for the Load input to go from false to true. When it does, the eight data inputs are converted to an ASCII code and the resulting character is displayed on a small screen.

Reader

The Reader opens a file on your hard disk when the Open input goes true. If the file was opened successfully it is rewound to the beginning and the Open output will be set to true. When the Read input goes from false to true eight bits of data are read from the file and the result sent to the eight output terminals. In binary mode bites are read unchanged, in ASCII mode carriage return and line feed characters are handled specifically for the platform you are using. When the Open input goes from true back to false, the file is closed and the Open output is set to false.

Writer

The Writer waits for the Open input to go true. When it does the Writer waits for the Write input to go true. The first time it does either a new file is created on your hard disk or an existing file is opened and characters appended. This depends on the settings in the Writer's dialog box. The Writer writes eight bits of data each time the Write input goes from false to true. In binary mode bites are written unchanged, in ASCII mode carriage return and line feed characters are handled specifically for the platform you are using. When the Open input goes false, any data in the file buffer is flushed to the file and the file is closed. If you don't close the file some or all the data may not be writer to the file.

FTP File Transfer

The FTP File Transfer can move files from your computer to the internet or from the internet to your computer. You can use this to share data with others by creating a file with the Data Logger and sending it to an FTP site so the other person can copy it from the FTP site to their computer and use Data Fetch to read it. You could also use FTP File Transfer to create a live web site. Create graphics with the Exporter, send the graphic to the web site. Incorporate it into the HTML of a web page. Each time the graphic changes you can trigger the Exporter to create a new graphic and use the FTP File Transfer to sent it to the web site. Users can hit the refresh button on their browser to see the latest graphic.

Socket Connect

The Socket Connect lets you connect to another computer. Enter the URL of the other computer and the socket that the other computer is accepting connections. Use this component in conjunction with one or more Socket Clients (see below).

Socket Client

The Socket Client can exchange messages with another computer. Use the Socket Connect to establish a connection. Then connect the output of the Socket Connect to the bottom input of the Socket Client. Each time the Strobe input goes from low to high the command string will be sent and a response string received back from that computer. The command is composed of the string set in the dialog followed by any string connected to the command input pin. Set the Output Mode to convert the response string to a scalar or an array of numbers as needed.

Read Only Memory

When the Read Only Memory is created, type in a string of characters. Then put a true on its Open pin and toggle its Get pin. Each time the Get pin goes from false to true the next character in the string will be put on the data output pins. Connect the Read Only Memory directly to the Display to view the data as it comes out. To restart the string from the beginning, set the Open pin to false then true again.

Memory Stack

The Memory Stack can be used to store 8-bit bytes of data or ASCII characters. Each time the Load pin goes from false to true, the data on the input data pins is loaded into the Memory Stack. Each time the Get pin goes from false to true, a byte of data is moved from the Memory Stack to the output data pins. When the data is valid on the data output pins the Valid output pin will go from false to true. When the Memory Stack has some data the More output pin will be true, when the Memory Stack runs out of data the More pin will go false.

The Memory Stack can be used in either of two modes: LIFO or FIFO. The LIFO mode stands for Last In - First Out. This means that when data goes in it is like stacking up dishes in a pile. When you take a dish off the top of the pile it was the last one you put there. The FIFO mode stands for First In - First Out. This means that when data goes in it is like sending balls through a tunnel. The first one in will be the first one out. Use the Memory Stack's dialog box to set either LIFO or FIFO mode.

Cursor Control

The Cursor Control lets you control other programs with MST Workshop. You can move the computer's cursor, click the computer's mouse, and type text where you clicked. Use this component to Start programs, open files, and enter text into dialogs.

Linear Filter

The Linear Filter is used to create a finite impulse response (FIR) digital filter from array coefficients. Each time the filter is triggered the input is multiplied by the first array coefficient. Inputs are saved and multiplied by succeeding coefficients at each trigger. These values are summed to calculate the output. The array can contain up to 32 coefficients.

If you don't connect the coefs pin at the bottom of the component you can select a filter from inside the dialog. Set the desired filter function and the number of points up to 16. The coefficients will be created internally.

Array Linear Filter

The Array Linear Filter is used to filter an array with coefficients from another array. Unlike the Linear Filter the entire array is processed at once and the coefficients are centered to reduce phase shifts between the input and output arrays.

As with the Linear Filter above, if you don't connect the coefs pin at the bottom of the component you can select a filter from inside the dialog. Set the desired filter function and the number of points up to 16. The coefficients will be created internally.

Input array can be 1 or 2 dimensional. Coefficient array can be 1 or 2 dimensional when the input is 2 dimensional.

Chapter 7. 2D Graphics Components

This chapter describes the components that are unique to the 2D Tools toolbar, the 2D Shapes toolbar, and the 2D Graphics menu.

Introduction

The 2D Graphics Toolbar contains components needed to create and place simple animated shapes. The 2D graphics components have a special terminal at the bottom and/or top. This terminal represents a 3 x 3 matrix. The 3 x 3 matrix contains information about the position, orientation and scale of a Cartesian coordinate system.

The descriptions of the 2D components describe a simple use where the left inputs are scalars and the bottom inputs and top outputs are 3 x 3 matrices. Like most components 2D Graphics can accept arrays at the left input. When used this way, instead of drawing a single shape (oval, rectangle, etc.) a shape will be drawn for each element in the array. Connecting an array to the X input for example will draw a series of shapes along the x axis. Multiple arrays can be used as long as all the arrays have the same number of elements.

The bottom input can also be connected to an array that contains multiple 3 x 3 matrices. If a 3 x 3 x 7 array is connected to a shape, 7 shapes will be drawn one for each 3 x 3 matrix and each will be rotated, scaled, and translated according to the respective 3 x 3 matrix.

Open 2D View

The Open 2D View button opens a window that shows the Axes, Points, Vectors, Ovals, Rectangles, Triangles, Parallelograms, Trapezoids, Lenses, Shapes, and Labels you have created and positioned with the components above.

The 2D View window has menu options to allow skipping the display of the workspace and open the 2D view window directly. This can be used with the MST Player to create programs for others. They will be able to see and interact with the 2D graphic but not be shown the workspace or have to be give instructions to open the 2D view.

In addition to menus to control the scale, axes, and grids in the 2D view there is also an option to display a geographic map background. The map is displayed in decimal degrees of latitude and longitude using the current view scale factors.

2D Reference

The 2D Reference establishes a standard Cartesian coordinate system with a fixed position, scale and orientation. It establishes a position of (0, 0), a scale of (1.0, 1.0), and an orientation of 0.0. Use the Reference as input to any of the other components.

2D Axis

The 2D Axis displays the position, scale and orientation of the matrix connected to the bottom input. The Axis is displayed in the 2D View window. Use the Axis to monitor the results of your transformations.

2D Knob

The 2D Knob allows the user to control the circuit from the 2D View window. When the user moves the knob in the 2D View window the values of the terminals in the circuit change. The Knob can be 1 or 2 dimensional.

2D Dial

The 2D Dial allows the user to control the circuit from the 2D View window. The dial operates like the Constant component, When the user moves the white dot on the dial in the 2D View window the output of the Button changes value.

2D Button

The 2D Button also allows the user to control the circuit from the 2D View window. When the user clicks on the button in the 2D View window the output of the Button goes True.

2D Mouse

The 2D Mouse component picks up the mouse position and button when the mouse is over the 2D View window.

2D Rotate

The 2D Rotate component allows you to rotate a Cartesian coordinate system about its origin. The Angle input determines the amount of rotation. The units can be specified in the 2D Rotate dialog. A value of PI Radians will rotate the axis 180 degrees counter clockwise. Its bottom input can be either the 2D Reference or the output of another 2D transforming component. Its top output can be connected to other 2D transforming components or shape components.

2D Translate

The 2D Translate component allows you to shift the Cartesian coordinate system in X and Y. The X and Y inputs are in pixels. Positive values will shift the coordinate system to the right and up. The bottom input can be connected to either the 2D Reference or the output of another 2D transforming component. Its top output can be connected to other 2D transforming components or shape components.

2D Scale

The 2D Scale component allows you to change the scale of the Cartesian coordinate system in X and Y. Positive values greater than 1.0 will enlarge the coordinate system. Positive values between 0.0 and 1.0 will shrink it. Negative values will flip it. The bottom input can be either the 2D Reference or the output of another 2D transforming component. Its output can be connected to other 2D transforming components or shape components.

Pixel Picker

The Pixel Picker lets you load an image in BMP format on Windows systems or in PICT format on Macintosh systems. You can input the X and Y location of a pixel and if the Get input is TRUE, the color of the pixel will be sent to the Red, Green, and Blue outputs. Color values range from 0 to 255, where 0 is dark and 255 is bright. The Valid input will become TRUE when the colors are available. X values start at the left with 0 and increase to the right. The Y values start at the top with 0 and increase downward. The limit is determined by the size of the image and the size of the Pixel Picker (Small, Medium, or Large).

Pixel Painter

The Pixel Painter lets you paint an image pixel by pixel. You can input the X and Y location of the pixel you want to paint. Then set the Red, Green and Blue values for the desired color of the pixel and set the Paint input to TRUE. Color inputs can range from 0 to 255, where 0 is dark and 255 is bright. You can save the resulting image at any time by using the Save Image button in the Pixel Painter’s dialog. Images can be saved in BMP format on Windows systems or in PICT format on Macintosh systems. If you connect a 2D Matrix to the matrix input on the bottom of the Pixel painter, the image will be drawn in the 2D View window. The scale and position of the input matrix will control the scale and position of the pixels in the 2D View window but the image will never rotate.

2D Compress

The 2D Compress component lets you build a coordinate system matrix from separate values. There are nine inputs that build a 3 x 3 matrix. The input pins are arranged in a matrix format.

2D Expand

The 2D Expand component lets you break down a coordinate system matrix into separate values. The matrix is broken down into nine outputs from the a 3 x 3 matrix. The output pins are arranged in a matrix format.

2D Transform

The 2D Transform lets you rotate, scale, and translate a point in space by inputting the X and Y coordinates and outputs the transformed X and Y values.

Point in Polygon

The 2D Point in Polygon component tests to see if a given point or points are within a given polygon. X and Y can be a single value or two matching arrays of values. When the inputs are arrays the output will be a array of TRUE/FALSE values.

Color Maker

The Color Maker takes Red, Green, and Blue inputs and assembles them into a color. If any input is an array the output is an array of colors.

2D Point

The 2D Point component draws an small dot in the 2D View window. The X - Y position of the dot depends on where the values of the X and Y inputs fall in the Cartesian coordinate system connected to the bottom of the Point component. The size and color of the point are set in the Point's dialog. The point can be labeled with either its input coordinates or its plotted coordinates in the view. The bottom input can be either the Reference or the output of a transforming component. The point also outputs its coordinates in the current view.

2D Vector

The 2D Vector component has three options the first draws a line, the second draws an arrow, the third draws a shape of your creation stretched between the starting and ending points in the 2D View window. The end points are set with the X0, Y0, X1, and Y1 inputs and depends on where these values fall in the Cartesian coordinate system connected to the bottom of the Vector component. The line width, and line color are set in the Vector's dialog.

When the Shape type is selected, you can select the particular shape, specify if you want it outlined or filled, and the fill color. The (0, 0) point of the shape is placed at the (X0, Y0) value. The shape is stretched so that the pickup point is placed at the (X1, Y1) point. If the X or Y pickup point is sent to zero, the shape will be stretched and rotated to match, if both X and Y pickup points are not zero, the shape will be stretched in two dimensions without rotation. The bottom input can be either the 2D Reference or the output of a 2D transforming component.

Common Features of the 2D Solid Shapes

The dialogs for these components are similar. They each have a Fractional Offset section. This lets you move the origin of the shape away from its actual center. For example when the Fractional Offsets are set to zero, the shape will be centered at the origin. If you want the shape to be shifted to the right and up so that the it pivots around its lower left corner, set both the X and Y Fractional Offsets to 1.0. Use -1.0 to offset in the other direction.

The dialog can also be used to set the Outline Color and Line Width of the shape. These shapes can also be labeled with text you choose. The label text will appear around the shape or centered in it as you choose.

You can choose to fill the shape with a Default Color Fill or leave it as an outline. If you connect values to the Red, Green, and Blue color inputs they will override the default color you have set in the dialog. The range of the color fill inputs is 0 (dark) to 255 (bright). If the value goes negative or is disconnected the appropriate value from the Default Color Fill from the dialog will be used.

2D solid shapes will appear in the 2D View window when the matrix input of the shape component is connected to a 3D Reference or transforming component. Alternatively a 2D shape will appear in the 3D View window when it is connected to a 3D Reference or transforming component.

Most 2D shapes have an input called Label. This input can take a value from either Analog or Digital Message Display component, or the Name output from the Planet Statistics component. Values at the Label terminal will be interpreted as 8 characters instead of a floating point number.

Note: If you have a version of MST Workshop that includes Arrays, you can connect an array to many of the 2D shape inputs. The result will be multiple shapes with the appropriate feature changing as the array indicates. For example: if you connect an array containing the values 30, 40, and 50 to the X position of an Oval, you will see three ovals in positions x=30, x=40, and x=50 in the 2D View window. You can connect arrays to more than one input but the arrays must be similar in parameters and dimensions.

Oval

The Oval component draws an oval shape in the 2D or 3D View window. The position and orientation depends on the Cartesian coordinate system connected to the bottom of the Oval component. The size of the oval is determined by the Width and Height inputs, these values are in pixels. The bottom input can be the 2D or 3D Reference or the output of any transforming component.

2D Arc

The Arc component draws an arc shape in the 2D View window. The position and orientation depends on the Cartesian coordinate system connected to the bottom of the Arc component. The size of the arc is determined by the Width and Height inputs, these values are in pixels. The bottom input can be the 2D Reference or the output of any transforming component.

Rectangle

The Rectangle component draws a rectangular shape in the 2D or 3D View window. The position and orientation depends on the Cartesian coordinate system connected to the bottom of the Rectangle component. The size of the rectangle is determined by the Width and Height inputs, these values are in pixels. The bottom input can be the 2D or 3D Reference or the output of any transforming component.

Triangle

The Triangle component draws a triangular shape in the 2D or 3D View window. The position and orientation depends on the Cartesian coordinate system connected to the bottom of the Triangle component. The size of the triangle is determined by the Width and Height inputs, these values are in pixels. The Offset input determines the position of the top of the triangle relative to the base. When set to 0.0 the top will be centered over the base, creating a isosceles triangle. When set to 1.0 the top will be directly over the right side of the base, creating a right triangle. When set to -1.0 the top will be over the left side of the base, also creating a right triangle. The bottom input can be 2D or 3D

Reference or the output of any transforming component.

Parallelogram

The Parallelogram component draws a parallelogram shape in the 2D or 3D View window. The position and orientation depends on the Cartesian coordinate system connected to the bottom of the Parallelogram component. The size of the parallelogram is determined by the Width and Height inputs, these values are in pixels. The Offset input leans the top of the parallelogram to the right for positive and left for negative values. The bottom input can be 2D or 3D Reference or the output of any transforming component.

Trapezoid

The Trapezoid component draws a rectangular shape in the 2D or 3D View window. The position and orientation depends on the Cartesian coordinate system connected to the bottom of the Trapezoid component. The size of the trapezoid is determined by the Width and Height inputs, these values are in pixels. The Inset input shrinks the top of the trapezoid for positive settings from 0.0 to 1.0. The bottom input can be 2D or 3D Reference or the output of any transforming component.

Lens

The Lens component draws the shape of a lens in the 2D View window. The position and orientation depends on the Cartesian coordinate system connected to the bottom of the Lens component. The size of the lens is determined by the Width and Height inputs, these values are in pixels. The Left Radius and Right Radius inputs cause the left and right sides to curve to the specified radius. Positive radii cause either side to bulge to the left, negative values cause the side to bulge to the right. The bottom input should be a 2D Reference or the output of a 2D transforming component.

2D Shape

The 2D Shape component draws a selected shape in the 2D or 3D View window. Shapes can be created by creating a set of x, y coordinates and saving them to a shape file. There are two shapes already created for you named “Spring” and “Star”.

To create a new shape click the Shape button. In the dialog use the New... button to open the shape editor. Move the mouse over the grid in the window and click the mouse at the desired position. By changing the Mode you can Move, Insert, or Delete points. The colors of the dots will change to indicate which points are involved in the change. You can also reposition the entire shape from the Mode menu.

Select A Transform allows you to perform an operation on the entire shape. This operation will take place when you select the operation from the popup menu.

2D Poly

The 2D Poly component draws a polyline, polygon, points, or a spline in the 2D View window. Shapes are created an array input. The array input should have two parameters, the first will be used as the X coordinate and the second as the Y coordinate.

2D Bezier

The 2D Bezier component draws a polyline of a Bezier in the 2D View window. The polyline is created from 8 scalar or array inputs. The Steps input indicates how many steps to create over the curve. The first point (P0x,P0y) is the starting point of the polyline, the last point (P3x,P3y) is the ending point. The other 2 points guide the polyline. The points can be either scalars or arrays but if some are arrays they must have the same number of elements.

2D Label

The Label component draws text in the 2D View window. You can control the position of the text and insert a numeric value between two preset text strings. You could label the weight of a mass by setting the Prefix to “Mass = “ and the Suffix to “ kilograms”. Then connect the Value input to the actual mass value from the circuit. You would then see the following text displayed in the 2D View window:

Mass = 12.4 kilograms

2D Text

The Text component draws a single ASCII character in the 2D View window. You can control the position of the text, the size, and if it is shown normally or highlighted. The text color, background color, border color and size are set in the dialog.

2D Paragraph

The 2D Paragraph component draws wrapped text in the 2D View window. You can control the position and size of the text. The text color, background color, and border color are set in the dialog. Files can be read using the File Array Reader.

Chapter 8. 3D Graphics Components

This chapter describes the components that are unique to the 3D Tools toolbar, the 3D Shapes toolbar, and the 3D Graphics menu.

Introduction

The 3D Graphics Toolbar contains components needed to create and place simple animated shapes. The 3D graphics components have a special terminal at the bottom and/or top. This terminal represents a 4 x 4 matrix. The 4 x 4 matrix contains information about the position, orientation and scale of a Cartesian coordinate system.

The descriptions of the 3D components describe a simple use where the left inputs are scalars and the bottom inputs and top outputs are 4 x 4 matrices. Like most components 3D Graphics can accept arrays at the left input. When used this way, instead of drawing a single shape (cone, sphere, etc.) a shape will be drawn for each element in the array. Connecting an array to the X input for example will draw a series of shapes along the x axis. Arrays can be used at multiple inputs as long as all the arrays have the same number of elements.

The bottom input can also be connected to an array that contains multiple 4 x 4 matrices. If a 4 x 4 x 7 array is connected to a shape, 7 shapes will be drawn one for each 4 x 4 matrix and each will be rotated, scaled, and translated according to the respective 4 x 4 matrix.

Open 3D View

The Open 3D View button opens a window that shows the Axes, Points, Vectors, Spheres, Cubes, Cones, Cylinders, Shapes, Extrusions, Revolutions, Labels, and Imported shapes you have created. The 3D view window can be rotated using the mouse by clicking in the window and dragging the mouse.

3D Reference

The 3D Reference establishes a standard Cartesian coordinate system with a fixed position, scale and orientation. It establishes a position of (0, 0, 0), a scale of (1.0, 1.0, 1.0), and an orientation of (0.0, 0.0, 0.0). Use the Reference as input to any of the other components.

3D Axis

The 3D Axis displays the position, scale and orientation of the matrix connected to the bottom input. The Axis is displayed in the 3D View window. Use the Axis to monitor the results of your transformations.

3D Light

The 3D Light component allows you to create 7 separate light sources. Lights can be Directional lighting, Point Sources, or Spotlights. Directional light sources are like distant light sources with parallel rays. The matrix input determines the direction of the light but the distance is always infinite. Point light sources are like small light bulbs. The matrix input determines the position of the light in space. Spotlights are like point light sources with the added ability to control the direction and width of the light output. In the case of Spotlights the position and direction are determined by the matrix input. Lights have Red, Green and Blue inputs that determine the color and intensity of the light,

Note: 3D Lights do not draw a shape in the 3D View but, you can place a sphere at the same location to create a visible light.

3D Rotate

The 3D Rotate component allows you to rotate a Cartesian coordinate system about any one of its axes. Double click the component to select the X, Y, or Z axis of rotation. The Angle input determines the amount of rotation. Units can be specified in the 3D Rotate dialog. A value of PI Radians will rotate the axis 180 degrees counter clockwise. Its bottom input can be either the 3D Reference or the output of another 3D transforming component. Its top output can be connected to other transforming components or shape components.

3D Translate

The 3D Translate component allows you to shift the Cartesian coordinate system in X, Y, and Z. The X, Y, and Z inputs are in pixels. Positive values will shift the coordinate system to the right, up, and closer. The bottom input can be connected to either the 3D Reference or the output of another 3D transforming component. Its top output can be connected to other transforming components or shape components.

3D Scale

The 3D Scale component allows you to change the scale of the Cartesian coordinate system in X, Y, and Z. Positive values greater than 1.0 will enlarge the coordinate system. Positive values between 0.0 and 1.0 will shrink it. Negative values will flip it.

The bottom input can be either the 3D Reference or the output of another 3D transforming component. Its output can be connected to other transforming components or shape components.

3D Compress

The 3D Compress component lets you build a coordinate system matrix from separate values. There are sixteen inputs that build a 4 x 4 matrix. The input pins are arranged in a matrix format.

3D Expand

The 3D Expand component lets you break down a coordinate system matrix into separate values. The matrix is broken down into sixteen outputs from the a 4 x 4 matrix. The output pins are arranged in a matrix format.

3D Transform

The 3D Transform lets you rotate, scale, and translate a point in space by inputting the X, Y and Z coordinates and outputs the transformed X, Y and Z values.

3D Point

The 3D Point component draws an small dot in the 3D View window. The position of the dot depends on where the values of the X, Y, and Z inputs fall in the Cartesian coordinate system connected to the bottom of the Point component. The size and color of the point are set in the Point's dialog. The bottom input can be either the 3D Reference or the output of a 3D transforming component. The point also outputs its coordinates in the current view.

3D Vector

The 3D Vector component has three options the first draws a line, the second draws an arrow, the third draws a 3D shape of your creation stretched between the starting and ending points in the 3D View window. The end points are set with the X0, Y0, Z0, X1, Y1 and Z1 inputs and depends on where these values fall in the Cartesian coordinate system connected to the bottom of the Vector component. The line width, and line color are set in the Vector's dialog.

When the Shape type is selected, you can select the particular shape, specify if you want it outlined or filled, and the fill color. The (0, 0, 0) point of the shape is placed at the (X0, Y0, Z0) value. The shape is stretched so that the pickup point is placed at the (X1, Y1, Z1) point. If the X, Y, or Z pickup point is sent to zero, the shape will be stretched and rotated to match, if X, Y, and Z pickup points are not zero, the shape will be stretched in three dimensions without rotation. The bottom input can be either the Reference or the output of a transforming component.

Common features of the 3D solid shapes.

The dialogs for these components are similar. They each have a Fractional Offset section. This lets you move the origin of the shape away from its actual center. For example when the Fractional Offsets are set to zero, the shape will be centered at the origin. If you want the shape to be shifted along the X axis so that the it pivots around its left edge, set the X Fractional Offset to 1.0. Use -1.0 to offset in the other direction.

The dialog can also be used to set the Outline Color and Line Width of the shape. You can choose to draw the shape filled with a solid color. You can control outline by setting the line width.

You can create other shapes by reducing the number of points in the shape's structure. For example setting the number of points on the Cone's base to 4 will result in a pyramid shape.

Sphere

The Sphere component draws a spherical shape in the 3D View window. The position depends on the X, Y, and Z inputs. The size of the sphere is determined by the Width, Height, and Depth inputs, these values are in pixels. The position, orientation and size also depend on the Cartesian coordinate system connected to the bottom of the Sphere component. The bottom input can be the 3D Reference or the output of a 3D transforming component.

Note: Spheres can have both reflective color and emissive color. Use emissive color to make the sphere glow.

Torus

The Torus component draws a toroidal shape in the 3D View window. The position depends on the X, Y, and Z inputs. The size of the torus is determined by the other 4 inputs, these values are in pixels. The position, orientation and size also depend on the Cartesian coordinate system connected to the bottom of the Torus component. The bottom input can be the 3D Reference or the output of a 3D transforming component.

Cube

The Cube component draws a cubical shape in the 3D View window. The position depends on the X, Y, and Z inputs. The size of the cube is determined by the Width, Height, and Depth inputs, these values are in pixels. The position, orientation and size also depend on the Cartesian coordinate system connected to the bottom of the Cube component. The bottom input can be the 3D Reference or the output of a 3D transforming component.

Cone

The Cone component draws a conical shape in the 3D View window. The position depends on the X, Y, and Z inputs. The size of the cone is determined by the Width, Height, and Depth inputs, these values are in pixels. The position, orientation and size also depend on the Cartesian coordinate system connected to the bottom of the Cone component. The bottom input can be the 3D Reference or the output of a 3D transforming component.

Cylinder

The Cylinder component draws a Cylindrical shape in the 3D View window. The position depends on the X, Y, and Z inputs. The size of the cylinder is determined by the Width, Height, and Depth inputs, these values are in pixels. The position, orientation and size also depend on the Cartesian coordinate system connected to the bottom of the Cylinder component. The bottom input can be the 3D Reference or the output of a 3D transforming component.

Surface

The Surface component draws a plane in X and Y. The Z direction of the plane is specified by the two-dimensional one-parameter array connected to the last left input labeled Array. If the array is all zeros, the plane will be flat. This component could be used to create a topographic map, for example. The position, orientation and size also depend on the Cartesian coordinate system connected to the bottom of the Surface component. The bottom input can be the 3D Reference or the output of a 3D transforming component.

Blob

The Surface component draws almost any shape in X, Y, and Z. The X, Y, and Z position of each point is specified by the two-dimensional three-parameter array connected to the last left input labeled Array. Think of the array as a two dimensional cloth that will be wrapped over a three dimensional shape. At each point on the cloth there are three values, these are the X, Y, and Z position of the 3D object on which it is wrapped. For example, to create a contour map, you would have to specify the longitude, latitude and altitude at each point on in the array. The position, orientation and size also depend on the Cartesian coordinate system connected to the bottom of the Surface component. The bottom input can be the 3D Reference or the output of a 3D transforming component.

3D Shape

The 3D Shape component draws a user created shape in the 3D View window. The position depends on the X, Y, and Z inputs. The size of the shape is determined by the Width, Height, and Depth inputs, these values are in pixels. The position, orientation and size also depend on the Cartesian coordinate system connected to the bottom of the Sphere component. The bottom input can be the 3D Reference or the output of a 3D transforming component.

To create a new shape click the 3D Shape button. In the dialog use the New... button to open the shape editor. Move the mouse over the grid in the window and click the mouse at the desired position. By changing the Mode you can Move, Insert, or Delete points. The colors of the dots will change to indicate which points are involved in the change. You can also reposition the entire shape from the Mode menu. Use the View to select the Front, Left or Bottom view.

Select A Transform allows you to perform an operation on the entire shape. This operation will take place when you select the operation from the popup menu.

Extrusion

The Extrusion component drags a 2D shape in the Z direction and shows the result in the 3D View window. The position depends on the X, Y, and Z inputs. The size of the extrusion is determined by the Sx, Sy, and Depth inputs, these values are multiplied by the size of the 2D shape. The position, orientation and size also depend on the Cartesian coordinate system connected to the bottom of the Extrusion component. The bottom input can be the 3D Reference or the output of a 3D transforming component.

Array Extrusion

The Array Extrusion component drags a 2D array in the Z direction and shows the result in the 3D View window. The shape array is any array with two parameters, first X then y. Be sure the 2D shape array has its points running clockwise to produce the proper lighting effects. The position depends on the X, Y, and Z inputs. The size of the extrusion is determined by the shape array and the Depth input. The position, orientation and size also depend on the Cartesian coordinate system connected to the bottom of the Array Extrusion component. The bottom input can be the 3D Reference or the output of a 3D transforming component.

Revolution

The Revolution component drags a 2D shape around the Y axis and shows the result in the 3D View window. The position depends on the X, Y, and Z inputs. The size of the revolution is determined by the Sx, Sy, and Sz inputs, these values are multiplied by the size of the 2D shape. The position, orientation and size also depends on the Cartesian coordinate system connected to the bottom of the Revolution component. The bottom input can be the 3D Reference or the output of a 3D transforming component.

Array Revolution

The Array Revolution component drags a 2D shape array around the Y axis and shows the result in the 3D View window. To correct strange lighting effects reverse the direction of the 2D shape array being used. The position depends on the X, Y, and Z inputs. The position, orientation and size depends on the Cartesian coordinate system connected to the bottom of the Revolution component. The bottom input can be the 3D Reference or the output of a 3D transforming component.

Array Bender

The Array Bender component drags a 2D shape along a 3D path with scaling and twist as it progresses. The section array is an array with two parameters, first X then Y. Be sure the section shape array has its points running clockwise to produce the proper lighting effects. The position depends on the X, Y, and Z inputs and the three parameter path array. The scale input controls the relative size of the shape as it moves along the path. The scale array can be one or two parameters. The section can also be rotated as it moves along the path by a one parameter array at the twist input. The position, orientation and size also depend on the Cartesian coordinate system connected to the bottom of the Array Extrusion component. The bottom input can be the 3D Reference or the output of a 3D transforming component.

Import

The Import component imports a shape in DXF format stored in the Shapes directory and draws it in the 3D View window. The position depends on the X, Y, and Z inputs. The size of the import is determined by the Sx, Sy, and Sz inputs, these values are multiplied by the imported shape’s dimensions. The position, orientation and size also depend on the Cartesian coordinate system connected to the bottom of the Import component. The bottom input can be the 3D Reference or the output of a 3D transforming component.

Chapter 9. Advanced Functions

This chapter describes the functions that are available in the Advanced Functions component of the Special Menu. These functions fall into three categories: trigonometric, comparative, and other esoteric functions (for example, random numbers, logarithms, and square root.)

Introduction

The next three Toolbars (Trigonometry, Comparison or More Math) contain an assortment of components. These functions can also be accessed from either the Advanced Functions button on the Special Toolbar or from the Special menu.

Note: You can connect an array to many of the advanced functions inputs. The result will be an array output. For example: if you connect an array containing the values 30, 40, and 50 to the input of the Sine function, the output will be an array containing Sine(30), Sine(40), and Sine(50). For functions that have two inputs you can connect arrays to more than one input but the arrays must be similar in parameters and dimensions.

Trigonometry Toolbar

The Trigonometry Toolbar contains the standard trigonometric functions and their inverses. It also contains the Hyperbolic trigonometric functions.

Sine (Sin)

The output is the sin of the input. The input angle is in radians, degrees or grads. The output will be between -1.0 and 1.0.

Cosine (Cos)

This functions output is the cosine of the input angle is in radians, degrees, or grads. The output is in the range of -1.0 to 1.0.

Tangent (Tan)

The output is the tangent of the input. The input angle is in radians, degrees or grads.

Arc Sine (ASin)

When the input is between -1.0 and 1.0 the output is the arc sine of the input in radians, degrees, or grads. The output will range from +Pi/2 to -Pi/2 when the selected angle mode is radians.

Arc Cosine (ACos)

When the input is between -1.0 and 1.0 the output is the arc cosine of the input in radians, degrees, or grads. The output will range from 0.0 to Pi (3.14159) when the selected angle mode is radians.

Arc Tangent (ATan1)

The output is the Arc Tangent of the input in radians, degrees, or grads. The output will range from +Pi/2 to -Pi/2 when the selected angle mode is radians.

Arc Tangent (ATan2)

This function takes the numerator (y) and the denominator (x) separately and internally divides them in this way it can calculate the arc tangent correctly when both x and y are negative.

Hyperbolic Sine (HSin)

This function’s output is the hyperbolic sine of the input.

Hyperbolic Cosine (HCos)

This function’s output is the hyperbolic cosine of the input.

Hyperbolic Tangent (HTan)

This function’s output is the hyperbolic tangent of the input.

Comparison Toolbar

The Comparison Toolbar contains functions that compare inputs to each other or zero and output a value based on that comparison.

Less Than (<)

Compares the analog inputs, if the top input is less than the bottom input the output will be Logic True (5.0), otherwise Logic False (0.0)

Equals (=)

Compares the analog inputs, if the top input is equal to the bottom input the output will be Logic True (5.0), otherwise Logic False (0.0)

Greater Than (>)

Compares the analog inputs, if the top input is greater than the bottom input the output will be Logic True (5.0), otherwise Logic False (0.0)

Less Than Zero (< 0)

If the input is less than zero the output will be Logic True (5.0), otherwise Logic False (0.0)

Equals Zero (= 0)

If the input is equal to zero the output will be Logic True (5.0), otherwise Logic False (0.0)

Greater Than Zero (> 0)

If the input is greater than zero the output will be Logic True (5.0), otherwise Logic False (0.0)

Maximum (Max)

The output of this function is the larger of the two inputs. You can build a peak detector by connecting the output to one of its inputs. It will output the largest value the other input has seen since the reset button was hit.

Minimum (Min)

The output of this function is the smaller of the two inputs.

Ceiling (Ceil)

This function’s output is the closest integer higher than or equal to the input.

Floor (Flr)

This function’s output is the closest integer smaller than or equal to the input.

Integer (Int)

The Integer function takes in a floating point number and passes the integer part out.

Fraction (Frac)

The Fraction function takes in a floating point number and passes the fractional part out.

More Math Toolbar

The More Math Toolbar contains functions that rarely used but handy to have when you need them.

Absolute Value (Abs)

When the input is positive the output is the same value. When the input is negative the output is converted to a positive number of the same value.

Reciprocal (1/X)

The output is one over the value of the input.

Modulo (Mod)

The output of this function is the remainder of the top input divided by the bottom input.

Power (Pow)

The output is the top input raised to the bottom input.

Random Number (Rand)

The output of this function is a random number between zero and the value of the input.

Coin (Coin)

The Coin function randomly returns either true or false each time the input goes from false to true. Statistically there will be the same number of true results and false results.

Die (Die)

The Die function randomly returns an integer between 1 and 6 each time the input goes from false to true. Statistically there are equal chances of getting each of the 6 possible results.

Square (Sq)

The output is the square of the input.

Square Root (SqRt)

The output is the square root of the input.

Exponential (Exp)

This function's output is the exponent of the input.

Logarithm, Natural (Log)

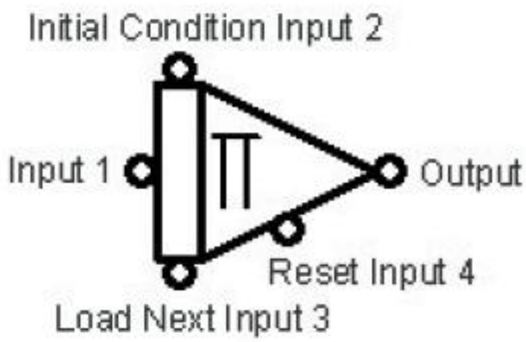
This function's output is the natural logarithm of the input.

Logarithm, Base 10 (Log10)

This function's output is the base 10 log of the input.

Product

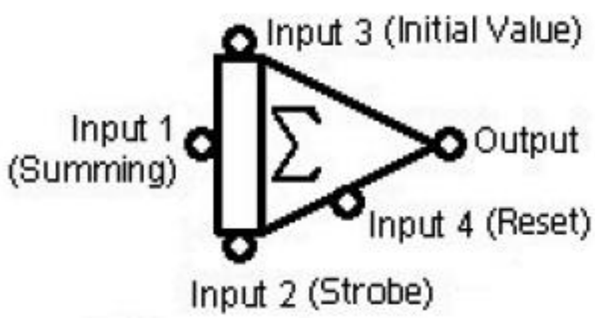
This component captures Input 1 (on the left) each time the Load Next Input 3 (on the bottom) goes from False to True. The values that are captured are multiplied together and sent to the Output. The Reset input 4 (bottom) clears the internal value or loads the value at the Initial Condition Input 2 (top).



Input and output pins on Product component

Sum

This component captures the input each time the Strobe input (bottom) goes from False to True. The Initial value and the sum of the inputs are sent to the Output on the right. The Reset (center) clears the internal values and sets the output to either zero or the value at the Initial Value Input (left).



Input and output pins on Sum component

Bitwise Math Toolbar

The Bitwise Math Toolbar contains functions that deal with numbers as if they were a collection of bits. Each bit is processed separately.

Bitwise And (And)

The Bitwise And function converts the inputs to 32-bit integers and Ands them together, then outputs the 32-bit result. If bits in the same position in each input are 1 the output bit will be 1. Otherwise the output bit will be 0.

Bitwise Or (Or)

The Bitwise Or function converts the inputs to 32-bit integers and Ors them together, then outputs the 32-bit result. If bits in the same position in either input are 1 the output bit will be 1. Otherwise the output bit will be 0.

Bitwise Exclusive Or (Xor)

The Bitwise Exclusive Or function converts the inputs to 32-bit integers and Ors them together, then outputs the 32-bit result. If bits in the same position in each input are different the output bit will be 1. Otherwise the output bit will be 0.

Bitwise Not (Not)

The Bitwise Not function converts the input to a 32-bit integer and inverts each bit, then outputs the 32-bit result. Ones are converted to zero and Zeros are converted to Ones.

Bitwise Shift Left (<Shift)

The Bitwise Shift Left function shifts the top input integer value by the number of bits indicated at the bottom input. A shift of 1 bit is the same as doubling the integer value. The result is output as an integer.

Bitwise Shift Right (Shift>)

The Bitwise Shift Right function shifts the top input integer value by the number of bits indicated at the bottom input. A shift of 1 bit is the same as cutting the integer value in half. The result is output as an integer.

Bitwise Compress

The Bitwise Compress takes in multiple Boolean (TRUE or FALSE) values and combines them bit by bit into the equivalent integer value. The Bitwise Compress can process 8, 16, 24, or 32 bits at once.

Bitwise Expand

The Bitwise Expand takes in an integer value and breaks it into bits. These bits are given out individually as Boolean (TRUE or FALSE) outputs. The Bitwise Expand can process 8, 16, 24, or 32 bits at once.

Chapter 10. Array Components

This chapter describes the components that are unique to the Array toolbars and menu. Arrays can be used with many components other than the components in this section. For example, connecting an array to the Square Root component will result in an array which contains the square roots of the numbers in the original array.

Introduction

This Toolbar contains components that are available from either the Array Toolbar or from the Array menu.

Arrays are collections of data values. Arrays can be 1-Dimensional like a list of numbers, 2-Dimensional like numbers in a spread sheet, 3-Dimensional (imagine a stack of spread sheets), or any Dimensional. Each data value can be a single number like a temperature, a pair of numbers like wind speed and direction, three numbers like the three numbers it takes to represent a three dimensional vector or any set of numbers. A set of data values can also be called a parameter.

Initially MST Workshop labels the parameters “p0”, “p1”, “p2”... You can change the labels to more appropriate names such as “temperature”, “pressure”, etc. The dimensions are labeled “d0”, “d1”, “d2” ... initially but also can be changed. The parameter or dimension labels can each be up to 31 characters in length.

Values in the array can be:

- Double (floating point value with 15 digits and an exponent from -308 to +308),
- Single (floating point value with 6 digits and an exponent from -38 to +38),
- Long (integer value from -2,000,000,000 to +2,000,000,000),
- Short (integer value from -32,767 to +32,767),
- Character (integer value from -127 to +127),
- Unsigned Character (integer value from 0 to 255),
- String 16 (text string of 15 characters),
- String 32 (text string of 31 characters),
- String 64 (text string of 63 characters),
- String 128 (text string of 127 characters),
- String 256 (text string of 255 characters),
- String 512 (text string of 511 characters),
- String 1024 (text string of 1023 characters), or
- String 2048 (text string of 2047 characters).

Note: Arrays can be connected to many of the inputs of many 2D shapes, 3D shapes, and Advanced Functions. For shapes, the result will be multiple shapes with the appropriate feature changing as the array indicates. With Advanced Functions the function output will become an appropriate array. You can connect arrays to more than one input of a shape or function, but the arrays must be similar in parameters and dimensions.

Array Creation Group

These components create arrays from your input.

Array Table

The Array Table creates either a one dimensional array with from one to eight parameters or a two dimensional array with just one parameter. Type in data into the dialog with tabs to space the data values. The array is saved in a file and can be used by other Array Tables or Array Data Stores. You can choose the type of the array ranging from character to integer to double precision floating point data.

Array Synthesizer

The Array Synthesizer creates an array from input parameters. In Parameter mode it creates a one dimensional N parameter array. You can specify up to 8 parameters. For each parameter you have three ways to choose the parameter indexing. 1) Start, Stop, Step; 2) Start, Count, Step; and 3) Start, Count, Stop. The resulting array will have values that range as specified by the inputs connected to the Array Synthesizer. A use of this mode is an array that has the positions for a 2D array of points representing the pixels in an image. The x values are in one parameter and the y values in the other.

In any of the Dimension modes, an array of 1 parameter in N dimensions is created. The inputs are used to specify the number of points in each dimension. The value of the array can be all zeros, all ones, a sum of the inputs, or the product of the inputs. An example of this is an array that can be used to construct the sum of a series of overlapping functions.

Array Grid Input

The Array Grid Input creates a two dimensional array of True/False values from a graphic. Use the dialog to select the size of the array. Blue circles represent False, Red True. Click a circle to change its color and the corresponding True/False value in the array will change to match.

Array Editor

The Array Editor creates a one dimensional array with between 1 and 8 parameters from graphic input. You can create an array with as many points as needed and move them with the mouse. Radio buttons on the right side select which parameter will be edited when the mouse is moved. Each parameter is shown in a different color and the selected parameter is drawn with a bold line. Inputs on the right allow you to save your work to a file and reload it later. Use the dialog to change the range of the chart, the number of points in the array, or the number of parameters. If you save the array to a file, it can be opened by other array components like the Array Data Store and the Array Table.

Array Geometry

The Array Geometry creates a one dimensional, two parameter array from graphic input. You can create an array with as many points as needed and move them with the mouse. The first parameter is changed when you move a point in the X direction (horizontal), the second parameter by moving in the Y direction (vertical). Inputs on the right allow you to save your work to a file and reload it later. Use the dialog to change the range of the chart or the number of points in the array. If you save the array to a file, it can be opened by other array components like the Array Data Store and the Array Table.

Array Data Store

The Array Data Store is the base component for arrays. It can hold up to 6 parameters in up to 6 dimensions. You can tailor the Array Data Store to your needs.
Some example uses:

> Temperatures over time would use 1 parameter in 1 dimension,
> Temperatures on a weather map would use 1 parameter in 2 dimensions,
> Temperatures within a cube would use 1 parameter in 3 dimensions,
> Wind speed and direction over time would use 2 parameters in 1 dimension,
> Winds on a weather map would use 2 parameters in 2 dimensions, and
> Molecules in 3-D motion within a cube would use 3 parameters in 3 dimensions.

The Array Data Store has 9 inputs on the left.

The first four inputs let you store data into the array or accumulate data with the data already in the array. The first input (Data Value) allows you to connect one or more values for each of the parameters. If you are working with one parameter you can just connect it to this input. If there are more than one parameter use an Array Compress to load up to six values into an array and connect the output of the Array Compress to the input. The next input (Position) lets you direct the data to a particular position in the array. If you are working with a one dimensional array just connect the value to the input. If you are working with a multidimensional array use an Array Compress to load up to six values into the position input. The next two inputs take action when triggered by changing the inputs from false to true. If the value at the Set Data Now input pin is triggered, the values on the input pins will be stored in the array at the specified position overwriting any previous value. Alternatively, if the value at the Accumulate Now input pin is triggered, the values on the input pins will be added to the values already in the array at the specified position.

The next inputs let you fill all positions of any parameter. The Fill Value input pin lets you set the value for each parameter separately. Use an Array Compress when you have multiple parameters to set. When the Fill Data Now input is triggered (goes from false to true) each parameter will be completely filled with the value at the Fill Value input pin.

The next input pin (Erase All Now) erases all the values in all parameters when it is triggered. The data are not set to zero but are set to a large negative number that is used to indicate that each value is empty. This allows you to have an array that is not completely filled with data but still be able to determine which positions have data and which are empty. It also makes it possible to calculate the statistics on the array without including the empty positions. For example if a 10 x 10 array has just 3 values the average value will be determined by using just the 3 values present and will not include the 97 empty values.

The last two inputs let you save the array to a file or load a file into the array. You have to decide on a file name and enter it in the Array Data Store’s dialog box before these input pins will work. Data is saved or loaded when the read or write input pin is triggered (goes from false to true).

The outputs on the right indicate the values you entered in the Array Data Store’s dialog These output pins give your circuit access to the number of parameters, dimensions and the size of each dimension.

The output on the top is used to connect the Array Data Store to other Array components. It is actually just a large negative value that other array components use to find the actual array information. This means that you can pass the value into Integrated Circuits or use the Transmitter and Receiver to send the value to other workspaces.

Array Data Gather

This component stores multi-dimensional array data. The Data Gather can be used to fill in the values of an array. The data, minimum, maximum, sum, or sum squared can be gathered. Inputs can be scalar or arrays to create multi parameter or multi dimensional array storage.

Array Constant

The Array Constant lets you create a character array (like a string of letters) by typing the text into a dialog. Simply type into the dialog and the result will be available as an array. This component is handy to use as input to the File Name inputs on the file and image read or write components.

Array Generator

This component builds an array using one or more Gaussian, Pulse, or Triangle functions. The inputs for the center, width and amplitude control each of the functions that will be summed into the output array. The resulting array will be 1 dimensional if the count and step are a single scalar value. If these inputs are arrays of two parameters the resulting array will be 2 dimensional. If they are arrays of three parameters the output will be a 3 dimensional array.

If the mode is Constant the array will be filled with that value. If the mode is Gaussian, Pulse, or Triangle the array will be filled with that shape. In this case the Amplitude, Center, and Width inputs can be scalar to generate a single shape or they can be a mixture of scalars and arrays. But the ones that are arrays must be the same size. When the output is 2D the Center and Width inputs can be 2 parameter arrays to control x and y separately. If the mode is Array Input then the Array Input connection needs to be an array. In the 1D mode the array should be 1D but in the 2D mode the array can be either 1D or 2D.

String Builder

Use the String Builder to construct a string array one character at a time. Load characters by strobing the Load input. Advance to the next string by strobing the Next input.

Array Display Group

These components display arrays in various ways.

Array Lister

The Array Lister lists the values in the array in a spread sheet like format. The inputs on the left allow you to scroll through the data when it won't all fit in the frame. The first dimension increases down the frame and is listed on the left side. The second dimension increases to the right and is listed along the top. The other dimensions (up to 8), are listed along the top of the frame and will page the values in the frame. When multiple parameters are present in the array they are shown in colors and listed at the top using the same colors.

Array Plotter

The Array Plotter lets you view the data in the array as a graphic You can choose the size and type of graphic in the Array Plotter's dialog. Connect the input pin on the bottom of the Array Plotter to any array. Either directly to an Array Data Store, Array List, etc. or any signal that results in an array. The Array Plotter can plot data in many ways depending on the data type:

> Data with 1, 2, or 3 parameters in 1 or 2 dimensions can be plotted as Dots, Connected lines, Connected Dots, Dots on a line, or Bars. The each parameter is shown in a different color.
>
> Data with 2 parameters in 1, 2, or 3 dimensions can be plotted using Scatter Plot that ignores the dimensions and scatters the data based on the 2 parameters.
>
> Data with 1, 2, or 3 parameters in 2 dimensions can be plotted as a Contour Plot. The parameters are contoured in different colors.
>
> Data with 1 parameter in 3 dimensions can be plotted as dots in a cube with the parameter determining the dot size.
>
> Data with 2 or 3 parameters in 3 dimensions can be plotted as vectors in a cube with the parameters determining the length and direction of the vectors.
>
> Data with 2 parameters in 1 dimension can be plotted as an xy spline.

Image Display

The Image Display shows an array as an image on the screen. To magnify or reduce the image display size use the dialog. You can also select the width and height of the frame.

Text Display

The Text Display shows a value as text. Values are floating point numbers that are stored in the computer using 8 bytes of memory. The Text Display displays those 8 bytes as 8 characters. You can create text values using the Analog or Digital Message. Once created these values can be treated just like any other value.

Array Information Group

These components provide information about arrays.

Array Statistics

The Array Statistics lets you get overall data about one of the parameters (up to 32). You can choose the type of data you want and the parameter you want it from in the Array Statistics dialog. When the Ready output is true the data is ready on the upper right pin. Connect the input pin on the bottom of the Array Statistics to an Array Data Store. The statistics currently available are:

Sum - The sum of all entries
Mean - The average of all entries
Minimum Value - The smallest entry value
Maximum Value - The largest entry value
Index of Minimum - The position of the smallest value
Index of Maximum - The position of the largest value
Entries - The number of entries
Variance - The statistical variance of the entries
Standard Deviation - The standard deviation of the entries
Average Deviation - The average deviation of the entries
Product - The product of all entries.
Skewness (3rd) - Variation of the third order.
Kurtosis (4th) - Variation of the fourth order.
Value Range - The maximum value minus the minimum value.
Next Peak - The data value of the second maximum.
Index of Next Peak - The position of the next maximum.
Next Valley - The data value of the second minimum.
Index of Next Valley - The position of the next minimum.

Array Ranges

The Array Ranges gives the range or size of each dimension of the input array. If the input array is a 5 by 4 array the values on the output pins will be 5 and 4 respectively.

Array Parameters and Dimensions

Use the Array Parameters and Dimensions to get the number of parameters and the number of dimensions from an array.

Array Labels

Use the Array Labels to get the parameter labels and the dimension labels from an array.

Array Find Value

The Array Find Value lets you search any one of the first 31 parameters of an array for a value that meets a specified criterion. The output of the Array Find Value will be the position of the first value that matches the criterion.

Array Segmenter

This component is used to split an array into separate arrays. One parameter of the input array is scanned. As long as the data matches, that segment of the array (with all its parameters) is written to the first output pin. When the parameter no longer matches that array is closed and the search continues. When the next match is found that segment is written to the next output pin, and so on.

Array Get Value

The Array Get Value lets you get the value from any one of the parameters (from 0 up to 31) at any position in the array. Connect the input pin on the bottom of the Array Get Value to the array to be accessed. The input pin on the left lets you select the position of the data. When a one dimensional array is connected at the bottom, you can connect a scalar value to indicate the position to get. When a multidimensional array is connected at the bottom, you need to connect an array with the number of values that the array has dimensions. For example: If the array at the bottom is 3 dimensional, connect a one dimensional array with 3 values on the left. Each of the 3 values will indicate the position in one of the 3 dimensions. If data is present for the input at the specified position the data will be available on the upper right pin. On the lower right the Ready pin indicates true (value of 5) when data is available.

Array Do Spline

The Array Do Spline takes a one dimensional, two or three parameter array as input. The points in the array are fit to a spline in 2 or 3 dimensions as appropriate. The left input is used to specify a point along the resulting spline. The input can be either the index into the array or a percentage of the total curve length. Use the Array Do Spline to create curves that pass through each point in the array.

Array Polynomial

This component is used to calculate the Y output value from an X input value and a set of polynomial coefficients.

Array Chebyshev

This component takes an X input value and an array of Chebyshev coefficients and outputs a Y value.

Array Linear Regression

The Array Linear Regression takes an array of any dimension containing 2 parameters. The parameters are treated like x and y coordinates. This component calculates the slope and intercept for a straight line that best fits the collection of points in 2 dimensions. It also outputs an error coefficient known as Chi squared.

Array LOESS

This component is used to plot a smooth curve through x, y data from the array input. Array must have two parameters. Parameter 0 is X and parameter 1 is Y. The X input can be a scalar or an array of values, the Y output will match it.

Array Singular Value Decomposition

This component is used to solve a set of equations or a matrix. This method solves most linear least-squares problems efficiently. Single parameter input will be the Y, double parameter input is X and Y, triple parameter input is X, Y, and sigma.

Array Chebyshev Fit

This component is used to tune parameters for a desired outcome. Solves problems like the Simplex Solver but without iteration.

Array Polynomial Fit

This component is used to calculate the coefficients to a polynomial that fit the input Data

Array Histogram

The Array Histogram creates an array that is a histogram of the input array. A histogram is a counting of the number of times that a range of data appears in the input. If the left input is connected to an array that is 0, 1.5, 3.0, 6.6, the histogram will have three values. The first value is a count of number of times the array connected to the bottom input has a value between 0 and 1.5, the second value output will be the number of values between 1.5 and 3.0, and the third value will be the number of values that are between 3.0 and 6.6.

Vector Dot Product

The Vector Dot Product calculates the dot product of two vectors. The input matrices must both be 1 x N vectors, i.e. arrays of 1 x 2 representing [x, y] or 1 x 3 representing [x, y, z]. The dot product is related to the cosine of the angle between the two vectors.

Matrix Determinant

The Matrix Determinant calculates the determinant of an N x N matrix.

Sudoku Hint

The Sudoku Hint suggests a value for one cell in a 9 x 9 matrix of characters.

Array Zero Crossing

This component takes an array representing a function and finds the first zero crossing. The array can be just Y values or X Y pairs. X Y pairs are created by parameter 0 being the X values and parameter 1 being the Y values.

Array Functions Group

These components modify arrays usually resulting in another array. Solve the entire Sudoku by feeding the suggestion back into the array.

Array Expand

The Array Expand outputs scalar values (normal voltages) from a 1 dimensional array. The dialog controls the number of outputs that the Array Expand shows on its right side. The values are always taken from the first parameter of the input array.

> Note: Select the Expand and type a digit to quickly set the number of Outputs in the package.

Array Compress

The Array Compress works in two ways. It can compress scalar values (normal values) into a 1 dimensional array. Alternatively, it can compress multiple arrays into an array of higher dimension. For example five 1D arrays can be compressed into a 2D array. Note: All the arrays being compressed must have exactly the same number of parameters and dimensions. The ranges of the dimensions must match also. The dialog controls the number of inputs that the Array Compress shows on its left side. Note: Select the Compress and type a digit to quickly set the number of Inputs in the package.

Array Concatenate

The Array Concatenate strings together multiple arrays into a larger array by increasing one if the dimensions. When there are multiple ways for the input arrays to be concatenated, you can specify the desired dimension to concatenate along.

Array Parameter Joiner

The Array Parameter Joiner joins up to 10 arrays into a single multi-parameter array. The output array is composed as follows: The first parameter of the first array is used to populate the first parameter in the new array. The first parameter of the second array is used to populate the second parameter in the new array. If a third array is input, its first parameter is used to populate the third parameter in the new array. The Joiner can be set to use just the first parameters from each input arrays, or all the parameters of each input array. Note: Select the Joiner and type a digit to quickly set the number of Inputs in the package.

Array Parameter Splitter

The Array Parameter Splitter splits a multi-parameter array into two or three separate arrays each having a single parameter. If the input array has three parameters then the Array Splitter will create 3 new arrays. If the input has two parameters it will create just two new arrays. Note: Select the Splitter and type a digit to quickly set the number of Outputs in the package.

Array Scalar Joiner

This component builds a multi-parameter array by joining scalars together. The result is a one-dimensional multi parameter array with a single value for each parameter. Note: Select the Joiner and type a digit to quickly set the number of Inputs in the package.

Array Scalar Splitter

This component splits a multi-parameter array into multiple scalar values. Note: Select the Splitter and type a digit to quickly set the number of Outputs in the package.

Array Parameter

This component splits a multi-parameter array into a single parameter array. The parameter number is chosen by the input to this component or by setting it in the dialog.

Array Set Difference

Use the Set Difference to create an array of unique elements from either input. Elements that are common to both inputs will not be included in the output. Input parameter types must match. Size and dimensions do not need to match. Use the menu to choose the difference type: Symmetric Difference, Items Only in A, Items Only in B.

Array Set Intersection

Use the Set Intersection to create an array of elements found in both inputs. Elements that are only found in one or the other input will not be included in the output. Input parameter types must match. Size and dimensions do not need to match.

Array Set Union

Use the Set Union to create an array of elements from the Union of two input arrays. Elements found in both inputs will be included in the output just once. Input parameter types must match. Size and dimensions do not need to match.

Array Subset

The Array Subset lets you select the ranges of up to 15 dimensions to create an array that is composed from just the data in those ranges. You can set the start position and number of values desired for each dimension. The array that is created is of the same type (Double, Single, Long, Short, or Character) as the input array.

Array Repeat

The Array Repeat will take any array and construct a new array by repeating the array as many times as the input specifies.

Array Resample

The Array Resample is used to create an array that is a resampled version of the input array. The input can be multi-parameter multi-dimensional. The left inputs must match the dimensions of the input array. All or a portion of the input array can be resampled to the output.

Array Sequencer

Use the Array Sequencer to create an array by looping with an Integrated circuit. The input array specifies the starting point and is the first element of the output array. This creates a new array that is dimensionally larger than the input and that dimension is the number of the Times input plus 1. If the input has n parameters the Integrated circuit must have n inputs and n outputs. Each value in the input array is fed to the Integrated circuit, the output is the next value in the output array. This process is repeated for the entire input array. This whole process is then repeated based on the Times input value.

Array Ravel

The Array Ravel takes an array of any dimensions and reforms it into the specified dimensions. Specify the desired dimensions by creating an array with the dimensions in order. To Ravel an array into a 2 dimensional array that is 7 by 8, use the Array Table to make an array with just two numbers in it: 7 and 8. Then connect that array to the left input of the Ravel component. Any array connected to the bottom input will be reformed into a 7 by 8 array.

Array Replace

The Array Replace scans through an array looking for values to replace. Select a comparison method: Equal, Less than, Less than or equal, Greater than, Greater than or Equal, or Not Equal. Then connect a value with which to compare. The values in the input array that match will be replaced by the value connected to the Replace input. Optionally connect a value to the Times input to control whether some or all the values in the array are replaced. The Array Replace component outputs the number of matches in the input array and the number of values actually replaced in the output array.

Array Assemble

The Array Assemble builds an array one layer at a time. Each time the Trigger input goes from false to true, the array at the input is concatenated into the output array. Start over at any time by taking the Clear input from false to true. For example, with a one dimensional input array of [1, 2, 3]. After the Trigger input goes from False to True the output will be [1, 2, 3]. The next time the Trigger input goes from False to True the output will be a two dimensional array with row 1: [1, 2, 3] and row 2: [1, 2, 3]. This component can be used as a FIFO by setting the limit to the size of the FIFO desired.

Array Shuffle

The Array Shuffle can shuffle an array in ways that resemble shuffling a deck of cards. Arrays can be shuffled in five ways: No Change, Perfect Interleave, Sloppy Interleave, Completely Random, Reverse Order, Prefect Cut, and Random Cut. Perfect Interleave is like a robot cutting the deck in half and fanning them back together. Random Interleave is more like a person cutting and fanning them back together. Completely Random uses a random number generator to mix the array. Perfect Cut divides the array in half and puts it back together. Random Cut divides the array in half at random and puts the halves back together. The shuffle can be triggered by an Edge at the input on the left or Continuously which means when the input data changes and the input on the left is high.

Array Labeler

Use the Array Labeler to change or add parameter and dimension labels to an array. Either connect string data to the inputs or open the dialog and enter the label text.

Substring

This component is used to extract a substring of an array of input string. Start sets the first character to be copied to the output, Size indicates how many characters will be copied max.

String Find

This component searches the input array of strings for matches. The output array is the position where the string was found. Negative output indicates the string was not found.

String Length

This component scans the input array of strings. The output array is the length of each string.

String CRC

This component scans the input array of strings. The output array is the CRC of each string.

String Trimmer

Use the String Trimmer to trim leading and/or trailing spaces and/or blanks off a string and output the resulting strings.

String Subset

Use the String Subset to select part of a string by giving a start string to match and an end string stop the match. Use the prefix and suffix inputs to add text before and after the matched text in the output. The inputs can be any types of strings but have to match the start string element count or have just a single element. Optionally add a delimiter character that will be placed between matches. All inputs must be a single parameter.

Array Find Index

The Array Find Index lets you search an array for a values that meet a specified criteria, the output of the Array Find Index will be an array of the positions of the values that matched. Arrays can be up to 3 parameters in 32 dimensions.

Array Find Match

This component is used to search for values of the array. An array is output that contains the positions of all the values that matched. Values can be numeric or text and must match exactly.

Array Lookup

The Array Lookup takes two arrays for input. The array that is connected on the left is used as a list of indices. The array that is connected to the bottom input is used for data values. The output is an array composed of the values from the bottom array taken from the positions specified by the array on the left. The resulting array will have the dimensions of the left array and the parameters and types of the bottom array.

Array Sorter

The Array Sorter is used to sort the values within an array. Values from one parameter are sorted into ascending or descending order. If the array contains more than one parameter, the values of the other parameters are moved along with the sorted parameter. In this way data can be rearranged by using one parameter as a key and the other parameters to contain data values to be rearranged.

Array Transpose

This component is used to re-arrange an array into its transpose. Arrays must be two-dimensional but can be any size and any number of parameters.

Array Inner Sum

The Array Inner Sum works in two different ways. By choosing either Row Sum or Row Product, Array Inner Sum will take in a multi-dimensional array and add or multiply all the values of the array along one of the dimensions. The result is a lower dimensional array. All parameter sets are summed separately into the resulting array. The dimension on which the sum is performed is specified in the Array Inner Sum's dialog as Dimension 0 through 31.

By choosing Add, Subtract, or Multiply, Array Inner Sum will take a one or more dimensional array and accumulate values, difference values or calculate a product – resulting in an array of the same size as the input array. For example in one dimension, if the input array is [1, 2, 3, 4], the Add setting will result in an array that is [1, 3, 6, 10], selecting Subtract will result in [1, 1, 1, 1], and selecting Multiply results in [1, 2, 6, 24].

Array Combine

The Array Combine takes an array with multiple parameters and either adds or multiplies the parameters together. The result is an array with the same dimensions as the input array but with a single parameter.

Array Type Converter

The Array Type Converter converts all the parameters in the input array to the specified type (Unsigned Char, Character, Short Integer, Long Integer, Single precision floating point value, or Double precision floating point value) in the output array. Values that can not be converted to the new type will be folded into some appropriate value in the new type.

Array Mirror

The Array Mirror reflects any dimension of the input array. The data is simply reversed in the specified dimension. If the input array is [1, 2, 3], the output array will be [3, 2, 1].

Array Rotate

This component is used to rotate a 1, 2 or 3 dimensional array. All parameters will be rotated. For example rotating a 1 dimensional array is the same as a mirror. For a 2 dimensional array a -90 degree rotation results in the X and Y directions being swapped. For a 3 dimensional array rotation is specified by an axis and a rotation in degrees.

Array Format

The Array Format creates a character array based on an analog input. Connecting a value of 1.2 to the left input will result in an array with the characters “1”, “.”, and “2”. You can specify formatting information like the number of decimal places to use and the number of characters in the resulting array.

Array Scan

This component scans a string array and returns a numerical value.

Array MotorDrive

This component drives a 4-pole stepper motor from analog inputs or an array. The Set inputs tells the component where the motor actually is. The Go To inputs tell the motor where to go. Connect an array to the bottom input to send the motor along a specified path. The Simple mode runs the motor at the speed input. The Ramp Each Leg mode starts slowly and ramps to full speed then slows down at the end of the leg. In this mode the Ramp Size setting in the dialog controls the number of steps it takes to achieve full speed. Also in the Ramp Min is the fraction of full speed needed to change directions or stop. The array should be a 2-parameter 1-dimensional array.

Array Outer Product

The Array Outer Product takes two arrays and multiplies, adds, subtracts, or divides each value in one array by every value in the other array creating a new array that has a larger dimension and values that are the products, sums, differences, or quotients of the original arrays. Select between the four possible functions in the Array Outer Product’s dialog.

Matrix Inverter

The Matrix Inverter calculates the inverse of a matrix. Multiplying a matrix by its inverse results in the unit matrix.

Matrix Multiply

The Matrix Multiply multiplies two matrices, this is different from multiplying the elements in two arrays using the Multiply component. The input matrices must be 2 dimensional and square, i.e.. N x N.

Vector Cross Product

The Matrix Cross Product calculates the cross product of two 3-dimensional vectors, i.e. a 1 x 3 array representing [x, y, z]. The cross product is a vector that is perpendicular to both input vectors. Additionally, you can input multiple 1 x 3 arrays. The output will them be a 1 x 3 array with multiple parameters.

Array Convolution (may not work correctly)

The Array Convolution multiplies two arrays together and outputs an array that represents the convolution of the two inputs.

Array Integrate

The Array Integrate is used to integrate an array resulting in another array. In the resulting array each value is the accumulation of succeeding values.

Array Differentiate

The Array Differentiate is used to differentiate an array resulting in another array. In the resulting array each value is the difference between neighboring values.

Array FFT

This component is used to calculate the FFT or Fast Fourier Transform (or its inverse) from an array. The result is another array of the same format and dimensions. The input array must be 1 dimensional and have an even power of 2 values (1024 values for example). The input should be two parameters one for the real values and one for the imaginary values. Note: Using just a real data set seems to have problems.

Array Permute

This component takes an array of items (the pool) and permutes them into groups specified by Group Size. The result is a 2-dimensional array the has all the permutations of the pool items.

Array Image Functions Group

These components can modify two dimensional character arrays in grayscale (one parameter) or color (three parameters).

Image Minimum

The Image Minimum searches around each pixel for a minimum value. That value then replaces the old pixel value. The radius of the search is controlled by the left input. The resulting image has enlarged dark areas.

Image Maximum

The Image searches around each pixel for a maximum value. That value then replaces the old pixel value. The radius of the search is controlled by the left input. The resulting image has enlarged light areas.

Image Gaussian

The Image Gaussian softens or blurs an image by combining adjacent pixels. Larger values on the left input increase the blurring effect.

Image Brightness

The Image Brightness shifts the values of each parameter by the value at the left input causing the image to look brighter. Negative values cause the image to look darker. Values that go below zero or over 255 are clipped to zero and 255 respectively. When a two-dimensional array is connected to the brightness input the image will be locally brightened (darkened) by the array. The brightness array will be scaled in X and Y to match the image data.

Image Contrast

The Image Contrast multiplies the values of each parameter by the value at the left input causing the image to have higher contrast. Negative input values are used to reduce the contrast of the image. Values that go below zero or over 255 are clipped to zero and 255 respectively. When a two-dimensional array is connected to the contrast input the image's contrast will be locally increased (decreased) by the array. The contrast array will be scaled in X and Y to match the image data.

Image Invert Levels

The Image Invert Levels creates a negative of the input image by changing each pixel from its value to 255 - its value.

Image Auto Levels

The Image Auto Levels expands the range of the values for an image. All the parameters are scanned, a brightness and contrast are calculated and all the parameters are modified to better fit between the range of 0 to 255.

Mirror Image

The Mirror Image swaps pixels from the left to the right and vise versa. The result is like looking in a mirror.

Image Flip

The Image Flip swaps pixels from top to bottom and vise versa. The result is an upside down image.

Image Rotate 90 Right

The Image Rotate 90 Right rotates an image to the right by moving pixels from the top row to the right side.

Image Rotate 180

The Image Rotate 180 rotates in image to be upside down.

Image Rotate 90 Left

The Image Rotate 90 Left rotates an image to the right by moving pixels from the top row to the left side.

Image Difference

The Image Difference subtracts two images. The result is an image that is dark where the two images match and bright when they don't.

Image Offset

The Image Offset shifts the pixels of an image horizontally and vertically based on the two left inputs. When pixels are shifted off the edge of the image or needed to fill the other side of the image is determined by settings in the dialog.
The Image Offset takes an Image (two-dimensional array), an X offset and a Y offset. If these inputs are scalar it shifts each pixel by the offset values. The effects at the edges can be controlled in the dialog. Choose to copy, fill, or wrap the image as needed. When a two-dimensional array is connected to the X and/or Y offset inputs the image will be locally shifted (distorted) by the inputs. The offset array will be scaled in X and Y to match the image data.

2D Image Project

The 2D Image Project takes an image at the left input (two dimensional array of characters) and a 2D matrix on the bottom input. The resulting image is shifted, rotated and/or scaled by the 2D matrix. The image is processed the same way shapes are processed before being drawn in the 2D view window.

Image Mixer

The Image Mixer takes from 2 to 32 images and combines them into a single image. Images can be color or grayscale and with or without an alpha channel.

Image Fader

The Image Fader takes 2 images and combines them pixel by pixel into a single image using a third control image. Images can be color or grayscale and with or without an alpha channel. Images need to be the same x/y dimensions. Each Pixel of the control image is read; when the control pixel is white a pixel from the White Image is used. When the control pixel is black a pixel from the Black Image is used. Gray control pixels cause pixels from the Black Image and White Image to be mixed in proportion.

Image Threshold

The Image Threshold takes in an image and converts it into a black & white image. Images can be color or grayscale (without an alpha channel). The conversion can be either a simple threshold or with error diffusion. When a two-dimensional array is connected to the threshold input the image will be locally thresholded by the array. The threshold array will be scaled in X and Y to match the image data.

Image Transition

The Image Transition takes in two images and transitions from one to the other based on the mode and input. Images can be color or grayscale (without an alpha channel). The images must be the same size in 2 dimensions and of char type. There are several transitions available. The output array will match the inputs in X and Y dimensions. If either input is color the output will be color. The range of the transition input is 0.0 to 1.0.

Image Silhouette

The Image Silhouette scans an image from one of its four sides. It compares pixel values to the threshold. The output is the distance it had to scan to find the threshold value in the image.

Array File Group

These components can read and write arrays in various ways.

Array Tabbed Reader

This component reads a file into an array. The array must be either a 1-parameter 1 or 2-dimensional array, or a 1-dimensional array of multiple parameters. The file must be a text file with fields delimited with tabs or commas. The result is an array that can be used by other components.

Array Saver

The Array Saver writes an array out to a specified file. Using this component will result in all the information in the array being saved. The file can be read by the Array Data Store, the Array Table, or the Array Geometry components.

Array Tabbed Saver

This component writes the array to a file. The result is a file that can be read with a word processor or spread sheet. It is tab delimited and currently limited to arrays of 1 or 2 dimensions.

Image Reader

The Image Reader reads a JPEG image file into an array. The array will have three parameters red, green and blue, with two dimensions to form an image. The Image Reader can be connected directly to the Image Display component to see the image.

Image Writer

The Image Writer takes a array and writes a JPEG image. If the array has 1 parameter the JPEG file will be a gray scale image, if there are 3 parameters it will be a color image. Set the name of the file to be written in the dialog or use the File Name input to connect a character array to specify the name.

File Array Reader

The File Array Reader reads any file into a character array. The array will be one dimensional and have just one parameter. One use of the File Array Reader is to copy a file by connecting it directly to a File Array Writer.

File Array Writer

The File Array Writer writes a one parameter, one dimension, character array into a file. Each character (byte) is written in sequence. Use this component to create text or binary files to your specifications.

Array File CRC

This component creates an array of CRCs from an array of file names. If the files name has no path, it is assumed to be in the same directory as the application.

Array File Info

This component creates an array of one of various file parameters from an array of file names. If the files name has no path, it is assumed to be in the same directory as the application. Some of the available parameters are file size, creation date, and modified date.

Directory Array

The Directory Array reads the directory at the path you specify, then builds an array with the names of all the files in that directory. The array will be a string array of one parameter and one dimension. The size of the array will be the number of files in the directory. Use the Array Subset to grab one file name at a time from the Directory Array's output.

Make Directory

Use the Make Directory to create a new directory. Use either the dialog or the inputs to set the path where the new directory will be created and the name of the new directory.

File Open

Use the File Open to open a file for read or write using other connected components. The file is opened when the input goes TRUE and is closed when it goes false. Connect the output on the top to the bottom of other file components like File Read One, File Write One, File Get Position, and File Set Position.

File Get Position

Use the Get Position to access the current byte pointer of an open file. Use the File Open to open and close the file. Use the File Set Position to move the file pointer to a new position.

File Set Position

Use the Set Position to move the current byte pointer of an open file. Use the File Open to open and close the file. Use the File Get Position to before or after using the Set to move a specific amount or check if the position moved as desired, respectively.

File Read One

Use the File Read One to read from an open file. Use the File Open to open and close the file. Select the data type and byte order from the dialog.

File Write One

Use the File Write One to write to an open file. Use the File Open to open and close the file. Select the data type and byte order from the dialog.

File Read Many

Use the File Read Many to read from an open file. Use the File Open to open and close the file. Select the data type from the dialog. Use the input to specify the number of values to read. An array will be created with 1 parameter and 1 dimension to hold the data.

File Write Many

Use the File Write Many to write to an open file. Use the File Open to open and close the file. All the data in the array connected to the input will be written to the file. Each parameter's type will indicate the type to write. All the dimensions will be written in a single linear array.

File Read String

Use the File Read String to read from an open file. Use the File Open to open and close the file. Select the file format of the string from the dialog. Use the input to specify the number of strings to read. An array will be created with 1 parameter and 1 dimension to hold the strings.

File Write String

Use the File Write String to write to an open file. Use the File Open to open and close the file. All the string data in the array connected to the input will be written to the file. All the dimensions will be written in a single linear array. The dialog sets the string format in the file

Chapter 11. Astronomy Components

This chapter describes the components that are unique to the Astronomy toolbar and menu.

Introduction

This Toolbar contains components that are available from either the Astronomy Toolbar, from the Astronomy menu.

Compress Date

Many astronomical calculations need a measure of time that is continuous and easily manageable. Compress date takes in the Year, Month, and Day and outputs Julian Day. Note: The Day input and Julian Day output include a fraction that represents the time of day.

Expand Date

Expand Date takes in Julian Day and breaks it down into Year, Month and Day. It is the opposite of Compress Date.

Date to Text

Use the Date to Text to convert a Julian Day into a text string. It is similar to Expand Date but formats the Year, Month, and Day into an easily readable text string.

Sunrise Sunset

Sunrise Sunset takes a Year, Month, Day, Latitude, Longitude, and Time Zone and calculates the time of Sunrise and Sunset on that day at that location. NOTE: Longitude over the United States is in positive values.

Moonrise Moonset

Moonrise Moonset takes a Year, Month, Day, Latitude, Longitude, and Time Zone and calculates the time of Moon rise and set on that day at that location. NOTE: Longitude over the United States is in positive values.

Moon Phase

Moon Phase calculates the times of the four phases of the moon starting with the New Moon on January 6, 2000. Each time the input is increased by one the date and time (Julian Day) of the next New moon will be given. To get the date and time of the First Quarter add 0.25 to the input. To get the date and time of Full Moon add 0.5 and add 0.75 for Last Quarter.

Planet Position

Planet Position takes in a planet number and the Julian Day then calculates the heliocentric coordinates of the Planets in the Solar System. The planet number is the index from the Sun for each planet, for example 3 is Earth. You can use the Planet Position connected to a 2D Oval to create a graphic that shows the planets circling the Sun.

Planet Statistics

The Planet Statistics component takes in a planet number and outputs one of six statistics for that planet. The statistics are:

Planet Name,
Semimajor Axis,
Sidreral Period,
Eccentricity,
Equatorial Radius, and
Inclination

Note: The Planet Name can be connected to a Text Display or the Label input of most 2D shapes.

Moon Position

Moon Position takes the Julian Day and calculates the geocentric coordinates of the Moon. To calculate the position of the Moon for a viewer on earth connect the output of the Moon Position to the Equatorial Coordinates then the output of that to the Horizontal Coordinates. Provide all with the Julian Day and connect the observers Latitude, Longitude, and Time Zone to the Horizontal Coordinates. The result will be the Azimuth and Altitude of the Moon for that day and time.

Sun Position

Sun Position takes the Julian Day and calculates the position of the Sun in geocentric coordinates. To calculate the position of the Sun for a viewer on earth connect the output of the Sun Position to the Equatorial Coordinates then the output of that to the Horizontal Coordinates. Provide all with the Julian Day and connect the observers Latitude, Longitude, and Time Zone to the Horizontal Coordinates. The result will be the Azimuth and Altitude of the Sun for that day and time.

Ecliptical Coordinates

Ecliptical Coordinates takes in equatorial coordinates and converts to ecliptical coordinates. Use this to transform Equatorial coordinates into Ecliptical coordinates. Given Julian Day, Ascension, and Declination as inputs it gives back Latitude and Longitude as outputs.

Equatorial Coordinates

Equatorial Coordinates takes in ecliptical coordinates and converts to equatorial coordinates. Use this to transform Ecliptical coordinates into Equatorial coordinates. Given Julian Day, Latitude, and Longitude as inputs it gives back Ascension and Declination as outputs.

Horizontal Coordinates

Horizontal Coordinates takes in equatorial coordinates and converts to horizontal coordinates. Use this to transform Equatorial coordinates into Horizontal coordinates. Given Julian Day, Ascension, Declination, Latitude, Longitude, and Time Zone as inputs it gives back Azimuth and Altitude as outputs.

Tide Height (requires optional tide data set)

The Tide Height component is part of the astronomy group because the tide height prediction is based solely on the positions of the Sun and Moon. The tide Height component requires tide data files in a directory named “TideData”. The files are the same files used by the Harbor Master program. The file name is the state name with a coded extension. The first two digits of the extension indicate the first year that is available in the file. The last digit of the extension is the number of years contained in the file. If the number of years exceeds 9, letters are used. For example a file covering Connecticut starting in 2005 and usable for 10 years is “CONNECTICUT.05A”. The first two outputs of the Tide Height component are the latitude and longitude of the station in degrees.

The next output is the daily tide data for the selected day. There is a single parameter in a single dimension. The data is in feet and spaced by 5 minutes resulting in 288 data points to cover the 24 hour day. The last output is a list of the highs and lows for the day.

This array has three parameters in one dimension. The number of entries is the number of highs and lows for the day. The first parameter is the height of the tide in feet, the second parameter is the time of the high or low in hours (0.0 to 24.0), the third parameter is an ASCII character either “H” or “L” indicating High or Low tide respectively.

Chapter 12. Special Components

This chapter describes the components that are unique to the Special toolbar and menu.

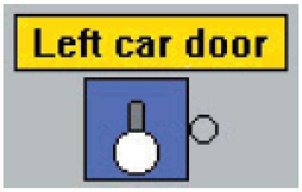
Introduction

The Other Toolbar contains an assortment of components including components needed to create and place Integrated Circuits.

Note

The Note is simply a label to make the circuit easier to understand. Use notes to label circuit components or describe the signal flowing through a wire.
To add a note, click note and type information into the Specify Note Parameters screen. Click OK. Then move the mouse and click on the workspace. The note appears on the workspace.

Following is a simple example of how a note is used:



A Note Added to a Circuit

Information Button

The Information Button can be placed on the workspace to explain something without taking a lot of space like the Note above. When you place the button it will show a dialog. You can enter a title and some information. The title will appear on the button followed by three dots. The information will be shown in a dialog when the button is clicked once.

Note: If you need to select the Information Button to move it, delete it, change its title or information you will have to select it by dragging a selection rectangle over it. Once it is selected you can move it, double click it, or click the right mouse button for the popup menu.

Buss In and Buss Out

The Buss In and Buss out components work together to allow you to connect components without wires. This can be useful if you need to connect lots of components together or when components are separated in the workspace and a wire connecting them would make the purpose of the circuit more difficult to see or understand. Start by connecting a Buss In at the source of the value. Its name will be three letters, you can change this in the dialog to have more meaning if you want. Then place one or more Buss Out components and connect them to where you want the value to be available.

Function Generator

The Function Generator is a table of numbers that represent a function. Enter the input and output values of a function and place the component. When you connect input a value the Function generator will look up the value in the table and output the corresponding value. For example: Suppose you want to calculate the total rainfall in your town for a month. If you know the daily rainfall you can enter the date as input and the rainfall as output. Now you have a function that takes in the day of the month and outputs the rainfall for that day. You can connect the Function Generator to an integrator and it will integrate the rainfall for each day resulting in the total rainfall for the month.You can enter the values for in the Function Generator's table or place a tab delimited text file in the FUNCGEN directory. Be sure to change the file's extension from "txt" to 'fun". To see the exact file format create a simple table and save it. Then use your word processor or spread sheet to open the file in the FUNCGEN directory.

Date Time

The Date Time icon contains sources that output calendar dates, times and workspace time.

Year (Year)
    The output is the calendar year.
Month (Mon.)
    The output is the calendar month. One is January, 12 is December.
Day of Month (Day)
    The output is the calendar day. The range is one to thirty-one.
Day of Week (WDay)
    The output is the day of the week. Zero is Sunday, 1 is Monday, 2 is Tuesday, 3 is Wednesday, 4 is Thursday, 5 is Friday, and 6 is Saturday.
Hour (Hour)
    The output is the hour of the day. Zero is midnight, one is one AM, twelve is noon, and twenty-three is eleven PM.
Minute (Min.)
    The output is the minute of the day. The range is zero to fifty-nine.
Second (Sec.)

The output is the second of the day. The range is zero to fifty-nine.

Real Hours (RHrs.)

The output is the time of day in hours with minutes and seconds as fractions of the hour. For example: 16.5 represents 4:30 PM.

Real Minutes (RMin.)

The output is the time of day in minutes with seconds as fractions of the minute. For example: 65.5 represents 1:05:30 AM (1:05 AM plus 30 seconds)

Real Seconds (RSec.)

The output is the time of day in seconds. For example: 3601 represents 1:00:01 AM (1:00 AM plus 1 second)

Date Time Group (DTG)

The output is the date and time grouped in a manner that is easy for humans to read. For example: 12312000.054321 represents December 31, 2000 at 5:43 AM and 21seconds.

Julian Date Time (JDT)

The output is the Julian day and time. You can use it as input for many of the astronomy functions.

Workspace Hours (WHour)

The output is the same time that is displayed at the bottom of the workspace, translated into hours.

Workspace Minutes (WMin.)

The output is the same time that is displayed at the bottom of the workspace, translated into minutes.

Workspace Seconds (WSec.)

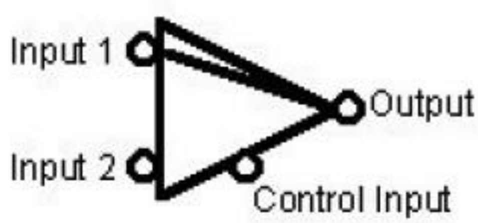
The output is the same time that is displayed at the bottom of the workspace.

Multiplexer

This switch makes a programmable change in a circuit, similar to an electrical relay. When the control input is False, it passes a signal from Input 1 (upper left) to the output. When the input is a True value, it switches to input 2 (lower left). Example: use it to change a boat's friction value changes when sliding down a ramp versus floating in the water.

If an array is connected to the control input, the output becomes an array of the same size. The array connected to the Control Input is treated as an array of True and False values. The content of the output array is either the value at Input 1 or Input 2 depending whether the Control array is True or False. For example: If input 1 is 2.3 and input 2 is 7.5 and the control array is [0, 5, 5, 0] the output array will be [2.3, 7.5, 7.5, 2.3].

If the control input is not connected, the input that changes will be connected to the output. This works for scalar values and arrays.



A Multiplexer

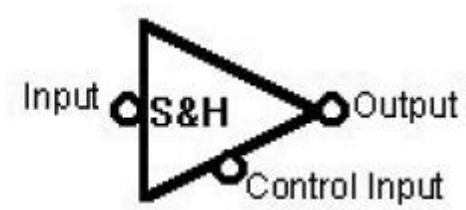
Selector

The Selector makes a programmable change in a circuit, similar to a multi-point relay. It passes a signal from the one of its inputs to the output. If you connect a value to the switch Input (bottom) it switches between the inputs based on the switch input's integer value. The inputs can also be arrays. In this case an array is output based on the switch input. If the switch input is an array, the output is an array built from looking up inputs based on the elements of the switch input.

Sample and Hold

The function of the Sample and Hold depends on the Control Input. When a False or no signal is connected to the Control Input, the Sample and Hold passes the signal unchanged to the output; when a True is connected to the Control Input, the Sample and Hold holds the last input. The device is useful for catching a value at a particular point in a cycle, such as the velocity of a bouncing ball when it hits the ground.

When an array is connected to the input, the entire contents of the array will be held at the output as long as the Control Input remains True.



A Sample & Hold

Limiter

This function passes the value of its input to the output; however, it limits the output to a preset value. If you set the limit value to 5, the output voltage will be limited to 5 when the sum goes above 5. In addition if the input goes below -5 the output will be limited to -5. Example: Use the limiter to enter the maximum amount of water a bucket can hold. The Limiter can take an array as input.

If you set the limit value to 0 the output will be limited to positive values.

To set the value of the input, double-click the Limiter. Enter a value and click OK.

In Range

This function compares the input value of to the limits in its dialog. If the input is within the specified limits, the output will be true (5) otherwise the output will be false (0).

Prime Number

This function outputs True if the input number is a prime and false if it is not a prime number. The input is truncated to an integer before it is checked. The Prime Number can take an array as input.

Exporter

This function captures an image of the workspace within its frame and writes it to a JPG file when the input goes from False to True. You can get a BMP file by using the .bmp extension. Use this component to publish components to a file that others can view.

2D View Exporter

This function captures an image of the 2D view window within its frame and writes it to a JPG file when the input goes from False to True. You can get a BMP file by using the .bmp extension. Use this component to publish components to a file that others can view.

Pulse Generator

This function generates pulses, square waves, triangle waves, sawtooths, or sine waves at a rate or period based on an analog input. In Rate mode the pulses or waveforms are generated at a rate in seconds determined by the input. In Period mode the pulses or waveforms are generated at the specified period in seconds. An input of 2 generates two pulses per second or a pulse every 2 seconds respectively.. The output can be a short pulse, a square wave, a triangle wave, a sawtooth, or a sine wave. The Pause input on the bottom freezes the beam where it is as long as it is true. The Reset input sets the output to zero, then restarts the timing at the beginning as soon as the reset input goes back to false.

Timer

The Timer only works while the workspace is running. It shows the length of time that its left input is true. If the input goes false the timer stops. The count resumes from where it left off when the input goes true again. To reset the timer to zero put a true signal on its bottom input. The output of the timer goes true at a preset time. Set the time in the Timer's dialog.

Dog

The Dog is a component that can output a value even when its input value is not ready. If you construct a circuit that is basically a closed loop, none of the components output will be ready because each is waiting for its input to be ready. So the result is that the loop never becomes ready and active. To get around this problem use the Dog component.

Insert it anywhere convenient in the loop. The Dog will provide an output that will get the next component ready. This will propagate around the loop until the Dog receives a valid input. At this point the Dog will act like a wire, changing its output to match its input.

IsDriven

The IsDriven indicates if its input is driven. Use this in an integrated circuit (IC) to determine if an input is driven (ready) and then take special action.

Delay

The Delay allows feedback in a solution. Use it to correct looping errors. When the workspace is running the input on the left pin will take one cycle to propagate to the output on the right pin. When the bottom input pin is true, the left input is disabled and the input on the top pin is used.

Sound Synthesizer

The Sound Synthesizer creates sound based on its inputs. The sound starts when the Trigger input goes from False to True. The Pitch input controls the frequency of the sound. The Amplitude and Duration control the amplitude and duration of the sound. There are four types of sound that can be produced; Square Wave, Sawtooth, Sine Wave, and Harmonic. In Harmonic Wave Mode you can specify up to 32 harmonics and amplitudes. Each harmonic can be any ratio to the Pitch input. The Envelope can also be specified in three sections: Attach, Sustain, and Decay. Attack = 10, Sustain = 20, and Decay = 70 would create sounds in which the amplitude increases for 10% of the

Duration input, then sustains for 20% and decays for 70%. The values do not have to be percents. You would get the same effect from values of 1, 2, and 7 or even 9, 18, and 63.

Array Sound

The Array Sound generates a sound from a wave form stored in an array. If the array has 1 parameter the result is monophonic sound, 2 parameters creates stereo sound. The array containing the waveform can be any type with a range of -32,767 to +32768 except when the type is character the values will be between 0 and 255.
The Array Sound can also save the waveform into a .wav file for playback with other applications.

Sound Reader

The Sound Reader reads .wav files into an array. The array will contain 1 parameter when monophonic sound files are read and 2 parameters when stereo sound files are read. The outputs on the right will indicate the number of channels read and the original sample rate. You can connect the outputs of the Sound Reader into the Array Sound directly or modify the waveform in any way you like before playing it or saving it back into a .wav file.

MIDI Send

The MIDI Send sends a note command to the MIDI synthesizer in the computer or connected to it through a MIDI cable. The Strobe/Gate input starts the sound. When using the Strobe trigger the sound will start when the input goes from False to True and continue until the sound is finished. When using the Gate trigger the sound will start when the input goes TRUE and play until the input goes False or the sound finishes. The Note input sets the pitch of the note, 60 in middle C. The Velocity input controls how hard the note is played. In the dialog you can select the Program. This determines the instrument that will be played.

Audio Capture

The Audio Capture imports audio signals from the computer’s audio input jack or built-in microphone.

Button

The Button allows a circuit to control itself. You can set the Button to be any one of the following: Reset, Stop, Run, Pause, Step, or Slow.

In each case when the input of the Button goes true the matching button at the top left of the work space will be activated. An example of the use of the Button is to stop a circuit when a falling object hits the ground.

Status

The Status allows a circuit to determine its own state by selecting one of the following: Stop, Run, Pause, or Slow.

The output of the Status will be true when the circuit is Stopped, Running, Paused, or in Slow mode, respectively.

Configure

The Configure allows a circuit to change a group of Constants, Constant Offsets, Constant Factors, Sliders, and Switches. Use the menu Edit >> Configure Constants to setup the desired constants. Then the circuit can recall this group of settings when the input goes from FALSE to TRUE.

Remote

The Remote allows a circuit to change a Constant, Constant Offset, Constant Factor, Slider, or Switch programmatically. Choose the part type then select the part from the labels in the list. There are three possible modes to choose from depending on your needs. Switches that have more than one pole use an integer to control each pole: 1, 2, 4, 8 etc. You can also use the Remote to put a second dial on large workspaces so you don’t have to scroll back and forth.

Dialog Ok

The Dialog Ok allows your circuit to give the user a status message and wait for a response. You can set the Message text in the dialog or connect a string to the Message input.

Dialog Yes/No

Dialog Yes/No allows your circuit to ask the user a multiple choice question. You can set the question text in the dialog or connect a string to the Question input. If you don’t connect anything to the Yes, No, and Extra inputs the dialog will have two buttons one set to Yes the other to No. You can change the button text by connecting strings to the

Yes and No inputs. If you connect a string to the Extra input a third button will show in the dialog with that text. The outputs will go high to indicate which button the user hit.

Dialog Value

The Dialog Value allows your circuit to ask the user for a numerical value and wait for a response. You can set the Message text in the dialog or connect a string to the Message input. If you input a value that value will be shown to the user and used if the user hits the Ok button without typing. This components Value output will be the value typed into the dialog when the Ok button is hit.

State Machine

The State Machine is a component where only one component can be TRUE at any time. Changing the input of any component to true will result in that component's output going true on the next clock tick AND the previously TRUE component will go FALSE You can also specify a group name to create independent groups. One component in a group can be specified as Initial. That component will go TRUE when the workspace is reset.

Any State

The Any State is either Any Terminal State or Any Error State. The component goes TRUE when any Terminal or Error state in the specified group is true. This component only updates its output on the clock tick or when the workspace is reset.

State Name

The State Name component gives the name of the state in the specified group that is currently true. This component only updates its output on the clock tick or when the workspace is reset.

PID Controller (working correctly?)

The PID Controller uses Proportional, Integral and Differential feedback to servo control something.

Simplex Solver

The Simplex Solver can drive an equation to a minimum. The Simplex solver outputs are connected to an equation of from 2 to 9 parameters. The output of the equation is connected to the Program input on the Simplex Solver. Also input a starting point for each of the parameters and an offset from that starting point (use 1.0 if you don't know what to use). As the Solver runs it will output values for the equation and respond to the value returned.

Array Simplex Solver

The Array Simplex Solver can drive an equation to a minimum. Group the inputs and deltas using an Array Compress. If all the deltas are the same you can just connect a scalar to the input pin. The Array Simplex solver outputs an array that you can split usint an Array Expand and connect its outputs to an equation. The output of the equation is connected to the Program input on the Array Simplex Solver. As the Solver runs it will output values for the equation and respond to the value returned. The Simplex Solver and Array Simplex Solver are identical in function. The advantage of the Array Simplex solver is that you can use more than 9 parameters.

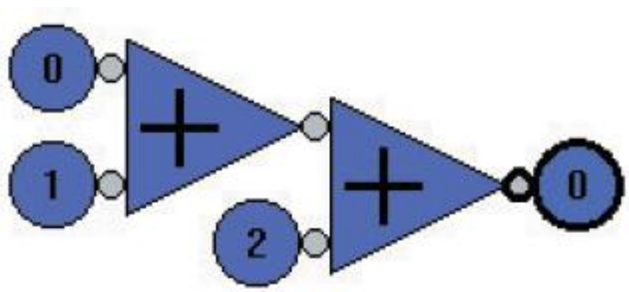
Bezier

The Bezier component creates values along one dimension of a Bezier curve. The points are created from 4 scalar inputs. The t input indicates how many steps to create over the function. The The first value (P0) is the starting point of the function, the last point (P3) is the end of the function. The other 2 values guide the function. The t input can be either scalars or an array. Whether a scalar or an array the t input normally ranges from 0.0 to 1.0.

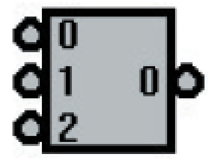
Integrated Circuit (IC)

An integrated circuit results when a number of parts become a collective. To qualify as an Integrated Circuit, a circuit must contain at least one Input Terminal or Output Terminal**.** These terminals are located beside the IC button on the Special toolbar or in the Special menu and are described in detail below.

For example, to create an adder that adds three terminals (rather than the usual two), create a circuit with two adders and place input terminals on three of the inputs. Connect an output terminal to the output of the second adder. The circuit looks like the one shown in the figure on the left below—we'll call this an expanded circuit. When you save an expanded circuit it creates an IC, which looks like the figure on the right below when you place it in the workspace from the Special >> Integrated Circuits menu.



Expanded Circuit IC



IC

Note: You must save the expanded circuit in the Circuits folder in order to open it as an IC.

To save a circuit, select Save on the File menu, type a unique name, and click Save.

To save a circuit so that the resulting IC has a label different from the file name, name the circuit using a carat (^). For example, to save a three-input adder, name the file triple adder^3add*.*

To open the file as an IC, click the integrated circuit icon in the Special tool bar. A list appears of the ICs you have saved so far. If the new IC doesn't show up in the list of IC's you may need to click the Refresh List button.
To open the file in it's original form (as an expanded circuit), use Open on the File menu.

- ICs can include analog, digital and graphic components.
- ICs can contain other ICs.
- Switches that have a Keyboard Toggle setting will work inside ICs.
-

If you place an IC into a circuit, then open and change its expanded circuit you will have to save the expanded circuit, delete the IC and place it into the circuit again for the changes to take effect.

Input and Output Terminals

Use input terminals to add connections that bring signals into a circuit. Use output terminals to bring signals from a circuit. These connections become part of an the integrated circuit that is created when you save a circuit.

MST Workshop labels terminals with numbers as they're added to the workspace.

> *To edit terminal labels, double-click the terminal on the workspace to access the settings dialog.*

To rename the terminal, type up to 6 letters or numbers in the Name field. Note that if each terminal in a complicated circuit has a 6-letter name, the circuit may become cluttered.

To perform a test before you save a circuit, enter a value in the Test Value field and run the circuit. When you open the circuit as an IC, the values will not be used.

Click OK to save terminal parameters.

Viewing IC Internals

Integrated circuits can be viewed while they are running by right clicking on the IC's icon and selecting View Internal. A small window will open with the complete circuit showing. You can enlarge the window and scroll around the circuit. Each integrated circuit will open its own window to show you the state of that particular circuit. You cannot edit the circuit in any way from this window. To change the IC use the Open command in the File menu to access the original circuit.

IC Sequence Index

The IC Sequence Index Terminal brings a value into the IC when the IC is being sequenced with an array; it is a used to get the index of the sequence for use in the IC. This allows the IC to be sensitive to the exact index (1 to times) that is being executed.

IC Sequence Position

The IC Sequence Position Terminal brings an array into the IC when the IC is being sequenced with an array; it is a used to get the position of the sequence. The form of the array will match the form of the input array but with a single parameter. The value will be the position being executed. This allows an IC to be sensitive to the exact element being computed.