

# MST Workshop

3/1/2025

Tom Pandolfi – MSTWorkshop.com

Windows Version 12.3

## Math Science and Technology Workshop

This application lets you create programs by connecting components. There are hundreds of components available. The menus on the top left group components into categories. Select a menu and look for the component you need. You can place the component anywhere on this workspace (window). Start with a component that generates a value, for example, the **Slider** in the **Controls** menu. Then place some function to the right of the **Slider**. For example the **Square** component in the **Functions** menu. Now you can connect the two components by letting their terminals touch. Alternatively, you can click on one of the terminals and drag the cursor to the other terminal. Now we need to display the output of the **Square** function. Use the **Display List** component in the **Controls** menu. Connect it to the output of the **Square** function. Finally, you can drag the “knob” (white circle) in the **Slider** and instantly view the square of the number selected. An easy way to search for a component is to start typing the component's name. In the example above, to find the slider type “SLIDE” - all the components will dim and only the components with “SLIDE” in their name will remain bright. To clear your search use either the Enter or the Esc key on the keyboard. You can also search by clicking the **Find** button in the upper right corner of the workspace. Type a word and hit OK. The program searches through the names and details of all the components. All the matching components will be listed. Now you can click on any of the lines in that dialog to create that component.

## Inputs and Outputs

Some components have inputs, some have outputs and some have both. You can connect components by snapping an input to an output or using a wire. The values carried between output and input terminals can be numerical values, strings or images. When a wire is carrying a single numerical value it is a thin black line. If it is carrying a string it is a thin green line. If the wire is carrying an array the colored line becomes thicker. Numerical values are what is called Double Precision which means that it is in the range of  $\pm 10$  to the  $\pm 308$  power. This wide range of numbers has at most 16 decimal digits of resolution. Strings can be any length only limited by your computer's memory available. Some components can handle more than 16 digits of precision by accepting strings of digits.

## Arrays

Some Components take arrays for inputs or yield arrays as outputs. For example, the **Count** component takes an array at its input and outputs a number that is the number of elements in the array. An example of a component that takes numbers and outputs an array is the **Synthesizer** component, it takes numbers and synthesizes an array specified by those inputs.

Additionally, many components that normally take single numbers can also take arrays. For example that **Add** component takes two numbers and outputs a number that is the sum. You can connect an array to one of the **Add** component's inputs and the output will be an array that has the same number added to each element. You can also connect two arrays of the same size and the **Add** component will add the arrays and output an array that is the sum of the respective elements.

## Time

Most components respond instantly to changes in their inputs by updating their outputs. There are also components that are time-based. These components consider time in their calculations. An example of a time-based component is the **Square Wave** component. Normally its output doesn't change, but when the workspace clock is running, the output changes based on the **Square Wave's** inputs and settings.

You can control the workspace clock with the buttons at the bottom left of the workspace. These buttons are disabled when there are no components in the workspace that are time-based. When you place a time-based component in the workspace these buttons will become active. Time-based components also change their color. When the clock is not running, time-based components are gray, when the clock is running they change to the color of their menu.

## Connections

There are a few ways to connect components. The simplest is to make their terminals overlap (when close they should snap together). Another way is to connect them with a virtual wire. Click and hold the mouse button on any terminal and drag away, a wire will follow the cursor. Move to another terminal and release the mouse button to connect the terminals. Note: While you are dragging and holding the left mouse button, you can click the right mouse button to put a bend at that point. This is handy when you need to go around another component.

To delete a wire you can right-click anywhere along the wire, a dialog will pop up with 3 choices. You can also delete a wire by clicking on one end of the wire and dragging it off the terminal. You can also bend wires after they are created by clicking and dragging at any point along the wire, a new bend will be created. To remove a bend, right-click on the point where the wire is bent and it will be removed. Wires can have as many bends as needed to look neat.

You can label wires by right clicking on a wire and selecting **Label...** Wire labels are handy when you create a subroutine. If a labeled wire goes into or out of a subroutine when its created, the terminal will use that label for its label.

Another way to connect components is with bus components. One type is the **Bus In** and **Bus Out** components. Bus components act like invisible wires. One **Bus In** component will connect to all the **Bus Out** components with the same name. If you need multiple independent buses you can give buses names by double clicking on the **Bus In** or **Bus Out** then tap **Enter Name**.

Another type of bus is the **Bus 2 In** and **Bus 2 Out** components. These buses can have multiple inputs with names, you can think of them as a cable with several wires inside. Set the number of inputs on the **Bus 2 In** component, then you can set the name of each input if you like. When you place a **Bus 2 Out** component, the program will search for the matching **Bus 2 In** component and automatically set up the number and names of the outputs.

## Subroutines

When your workspace gets cluttered or confusing you can break things into pieces. This is where the subroutine feature comes into play. For example, suppose you are calculating the volume of a cylinder. That calculation takes several components. Instead of putting that calculation in the main workspace, you can open a subroutine (click the **Edit 1** button in the **Subroutines** menu) and put the components there. To get variables into this new page you use an **Input** component (on the **Controls** menu). To get your results back out of the page you use the **Output** component. After creating this new page, return to the Main workspace (click **Subroutines** then the first **Edit**). Next, select the **Subroutines** menu again and you will see the **Place 1** button is active. Click and drag that button and a **Subroutine** component will be placed on the main workspace. The **Subroutine** component will now take inputs from the left terminals and pass those values to the subroutine's **Input** components. Next, it will calculate the volume of a cylinder and take the value of the **Output** components and pass it to the **Subroutine's** right terminals on the main workspace. You can place this **Subroutine** component anywhere you need to calculate a cylinder volume on your Main workspace. You can add labels to the **Input** and **Output** components for clarity. You can also name the **Subroutine** by clicking **Subroutines** then the **Sub 1** button and giving it a title. See an example in the **Input** or **Output** descriptions below.

In the bottom right corner of the workspace there is a button that will highlight when you select some components that you wish to convert into a subroutine. When you click **Make Subroutine** the components in the selection rectangle will be moved into the next available subroutine and will be replaced on the current workspace with a Subroutine icon. If there are wires that cross into or out of the rectangle they will become terminals on the subroutine. If these wires are labeled, the inputs or outputs on the new subroutine will label the terminals to match the wire label.

## Scrolling and Zooming

You can shift the workspace by right-clicking in a blank area where there are no components, holding the right button down, and then dragging the mouse. You can also use the arrow keys and **Home** key (when no **Keyboard** component is in the workspace). There are also buttons in the Options menu that shift the workspace: **Page Left, Page Right, Page Up, and Page Down. Reset Page** returns you to the center of the workspace. If you have some components selected, the arrow keys move the selection up, down, left, or right by one pixel at a time.

To get a closer look at something in the workspace or to zoom out for a wider view, you can use the wheel on the mouse to zoom in or out. If your mouse doesn't have a wheel, you can use the **Zoom In** and **Zoom Out** menu buttons in the **Options** menu. **Reset Zoom** restores the original screen size and position. The mouse wheel is also a button, if you activate that button while scrolling the wheel you will get a ten times magnification of the zoom effect.

A few controls can be operated using the mouse wheel. Hover over the control you want to move vertically and rotate the mouse. The components with this ability are Slider, Slide Factor, Slide Offset, Knob, Knob Factor, Knob Offset, Rack, Rack Factor, Rack Offset, Function, Function X1, Function X2, Switch, Switch 2, Gate, and Display List. Note: When you are trying to get an exact number and using the mouse is too coarse and skips the number you want, try the wheel.

You can also enlarge and reduce the menus. The Options menu has three menus that relate to the size of the menus: **Reset Menus, Enlarge Menus, and Reduce Menus.**

## Selection

To modify a component right click on the component and a menu will pop up. You can **Delete**, get **Details**, **Duplicate**, or give the component a label. Some components have extra buttons. Often you can change the inputs that are available. This allows you to only have inputs that you need and minimize complex components. Some components have **Options** that are available from this menu and are different for each component.

To select a group of components for deletion or duplication, start by clicking where there are no components, then drag to select some components. the workspace where there are no wires or components. A red frame will appear as you drag, when you release the mouse the selected components will become bold. Now you can drag the components as a group. You can also right-click on one of the components in the group to **Delete** or **Duplicate** the group. Note: After duplicating and while the group is still bold, you can drag the new copy to a more desirable location.

## Hovering

If you hover the mouse over an input terminal a text will appear at the bottom right of the workspace indicating the component's number followed by the word “dim” followed by the dimensions of the data being input to that component. Next will be a description of the type of input: numerical, string, or image. Lastly will be the first few values of the data. Similar results will be displayed when you hover over an output terminal.

When you hover over a component, the text will usually show “operating normally” to indicate that the component's inputs are satisfied. When there is a problem with the inputs, the component will change its color to gray and the message will describe which input is not connected to an acceptable value. If the component is designed to operate when the clock is running (buttons at the lower left of the workspace), it will be gray and say “requires clock running.”



## Colors

Several dialogs control graphics colors in the workspace. The colors of the seven menus at the top of the workspace are set from a dialog that is activated by clicking the **Set Button Colors...** item in the **Options** menu. The **Chart**, **XY Chart**, **Recorder**, and **XY Recorder** pen colors are controlled in dialog activated from the **Set Pen Colors...** dialog. The Workspace background, various charts, and control backgrounds are activated from the **Set Background Colors...** dialog. The major and minor gridlines in the charts and recorders, the wire colors, and the lines that connect the various knobs are activated from the **Set Line Colors...** dialog. The various texts shown in and around components are controlled by the **Set Text Colors...** dialog. You can also check the **Dark Theme** item to reverse video everything in the workspace. Finally, if you want to return to the colors that were originally set you can select the **Restore Default Colors** item.

## Errors

When a component's inputs are not sufficient to allow the component to operate, the component will change color to gray. If you hover the mouse over the component, the text will appear in the lower right of the workspace indicating the problem. Components that require a changing time will also be gray when the time is not running as described above.

## Automatic Backup

After you save a file the first time or after opening a file, the Automatic Backup will be activated. You can disable it from the checkbox in the Options menu. When enabled, a copy of the program will be saved periodically as you make progress on your solution. The file name will be the name you have chosen with the word *backup* added.

## Default Values

Most component inputs have a default value of zero. Some color inputs default to 255 to make the view in the 2D or 3D window look reasonable when nothing is connected. A new feature allows you to change the default values for many inputs. The advantage is that you don't need to create a constant for every input. The disadvantage is that unlike a constant you can't see the default value. I suggest you only use the "Defaults..." menu when knowing the default value is not critical to understanding the overall diagram (solution).

## Printing

You can print the Workspace or the 2D View. I noticed that after printing that a second printing has problems. While I figure out how to fix it, I added a warning dialog that recommends saving your work and restarting MST Workshop.

## Table of Contents

!= (not equal to).....	16	Average.....	59	3D Clutch .....	104
!=0 (not equal to zero).....	17	3D Axis .....	60	Color Bars .....	105
< (less than).....	18	Beep .....	61	2D Color Bars .....	106
<= (less or equal to).....	19	Bezier .....	62	Color Image Modify .....	107
<=0 (less or equal to zero)....	20	Bezier Derivative .....	63	Color Patch .....	108
<0 (less than zero).....	21	Binary Counter.....	64	Color to Grayscale .....	109
= (equals).....	22	Binary Tree .....	65	Combine .....	110
=0 (equals zero).....	23	Bit AND .....	66	Common Factor .....	111
> (greater than).....	24	Bit Joiner .....	67	Compare .....	112
>= (greater or equal to).....	25	Bit NOT .....	68	3D Cone .....	113
>=0 (greater or equal to zero)	26	Bit OR .....	69	Concatenate .....	114
.....	26	Bit Splitter .....	70	Constant .....	115
>0 (greater than zero).....	27	Bit XOR .....	71	Constant Factor.....	116
Abs (absolute value).....	28	Bits to Number .....	72	Constant Offset .....	117
Accumulator .....	29	2D Bitmap .....	73	2D Contrast.....	118
Add .....	30	3D Bitmap Text.....	74	Copy To Clipboard.....	119
Add Fractions .....	31	2D Box .....	75	Correlation .....	120
AES 128 Decryption .....	32	3D Bridge .....	76	Cosine .....	121
AES 128 Encryption.....	33	2D Brightness .....	77	Count .....	122
AES 256 Decryption.....	34	Buffer .....	78	Cows Bulls Eval .....	123
AES 256 Encryption.....	35	Burst Generator .....	79	Cows Bulls Guess.....	124
Alpha Rank.....	36	Bus 2 In .....	80	Cows Bulls List .....	125
Alpha Sort .....	37	Bus 2 Out .....	81	Cows Bulls Poss .....	126
Alternate.....	38	Bus In .....	82	Create Folder .....	127
Anagram .....	39	Bus Out .....	83	2D Cross .....	128
AND .....	40	Button .....	84	Cross Product .....	129
2D Angled Slider .....	41	Button 2 .....	85	Cube .....	130
2D Aperture.....	42	Button Grid.....	86	3D Cube .....	131
3D Aperture .....	43	Camera .....	87	3D Cube Surface .....	132
Arc Cos.....	44	Ceil .....	88	3D Cube Surface 2 .....	134
Arc Sin .....	45	Center to Edge .....	89	Cylinder Shape.....	136
2D Arc Slider .....	46	2D Chain .....	90	3D Cylinder .....	137
Arc Tan .....	47	Characters to String .....	91	3D Cylinder Surface .....	139
Arc Tan2 .....	48	Chart .....	92	3D Cylinder Surface 2 .....	140
Arrange .....	49	2D Chart .....	93	3D Cylindrical Mirror.....	141
Array Diffs .....	50	2D Checkbox .....	94	D Flip Flop .....	142
Array Rands .....	51	Checkers.....	95	D to A .....	143
2D Array Slider .....	52	Chord Transpose.....	96	Data Range .....	144
Array Sums .....	53	Chroma Key.....	97	Data[1] to Grayscale .....	145
Array to Sound .....	54	3D Circle .....	99	Data[3] to Image .....	146
2D Arrow .....	55	Circle Center .....	100	Date to Seconds.....	147
3D Arrow .....	56	Classify.....	101	Day .....	148
Audio Input.....	57	Closest Value.....	102	Day Of Week .....	149
Ave .....	58	2D Clutch .....	103	Days In Month .....	150

2D DC Motor .....	151	2D Field Plot .....	199	Get Startup Directory .....	246
3D DC Motor .....	152	FIFO .....	200	Get Until .....	247
Deal .....	153	File SHA-256.....	201	2D Glass Plate.....	248
Dealer .....	154	Filter .....	202	3D Glass Plate.....	249
Decimate .....	155	Find Angle .....	203	GraphBase .....	250
Delay.....	156	Find Bool .....	204	Gray Bar.....	251
Delete File .....	157	Find Char Index .....	205	2D Gray Bars.....	252
Delta Sensor.....	158	Find Index .....	206	Gray Colorize .....	253
Delta to Point.....	159	Find Peak .....	207	Gray Image Modify .....	254
Deselector .....	160	Find String .....	208	Grayscale to Data[1] .....	255
Detent .....	161	Finite State .....	209	3D Grid .....	256
Difference .....	162	First N .....	210	Harmonic Gen .....	257
Digital Solver.....	163	2D Flat Distance .....	211	Harmonic Gen 2 .....	258
Digits To Value .....	164	2D Flip.....	212	Hex to String.....	259
Dimensions .....	165	Flip Flop .....	213	3D Hexahedron.....	260
Display Grid .....	166	Floor .....	214	Hi-Res Timer .....	261
Display List .....	167	Fold.....	215	Histogram .....	262
Display Plot.....	168	Format .....	216	Hold .....	263
3D Distance .....	169	Fourier Transform .....	217	2D Horizontal Axis .....	264
Divide .....	170	Frame .....	218	2D Horizontal Slider .....	266
Divide Fractions .....	171	Func. Gen. ....	219	Horizontal Stripes .....	267
3D Dodecahedron.....	172	Func. Points .....	220	Hour .....	268
2D Dot Product .....	173	Func. Rack At.....	221	HSV to RGB .....	269
3D Dot Product .....	174	Func. with Mode.....	222	2D HSV Tint.....	270
3D Draw Shape.....	175	Func. XY At.....	223	Huffman Compress.....	271
Duplicate .....	176	Func. Y At.....	224	Huffman Expand.....	272
Edge Enhancer .....	177	Gate .....	225	Image Angle.....	273
Edge to Center .....	178	Gather .....	226	Image Area .....	274
Eliminate Value .....	179	Gaussian.....	227	Image Arrow .....	275
Elliptical Path .....	180	2D Gear .....	228	Image Bilateral Filter .....	276
2D Ellipse on Center .....	181	3D Gear.....	229	Image Bounds .....	277
2D Ellipse on Focus .....	182	3D Gear 2 .....	230	Image Box .....	278
2D Encoder.....	183	Generate Prime .....	231	Image Brightness .....	279
3D Encoder .....	184	Geodesic Shape.....	232	Image Centroid .....	280
English Words.....	185	3D Geodesic.....	233	Image Circle .....	281
Excise .....	186	Geometry .....	234	Image Color Index .....	282
Expander .....	187	2D Geometry .....	235	Image Contours.....	283
Exponential .....	188	Get At .....	236	Image Contrast .....	284
3D Extrude .....	189	Get At Reverse.....	237	Image Crop .....	285
3D Extrude 2.....	190	Get Bitmap .....	238	Image Cross .....	286
3D Extrude 3.....	191	Get Char At .....	239	Image Dilate .....	287
Extruded Shape.....	192	3D Get Direction .....	240	Image Display .....	288
3D Faces .....	193	Get Directory .....	241	Image Ellipse .....	289
Factor .....	195	Get File Info .....	242	Image Equalize .....	290
Factorial .....	196	Get File Name.....	243	Image Erode .....	291
Fader .....	197	Get From Clipboard .....	244	Image Fader .....	292
2D Fenestrate.....	198	Get Pixel .....	245	Image Flip.....	293

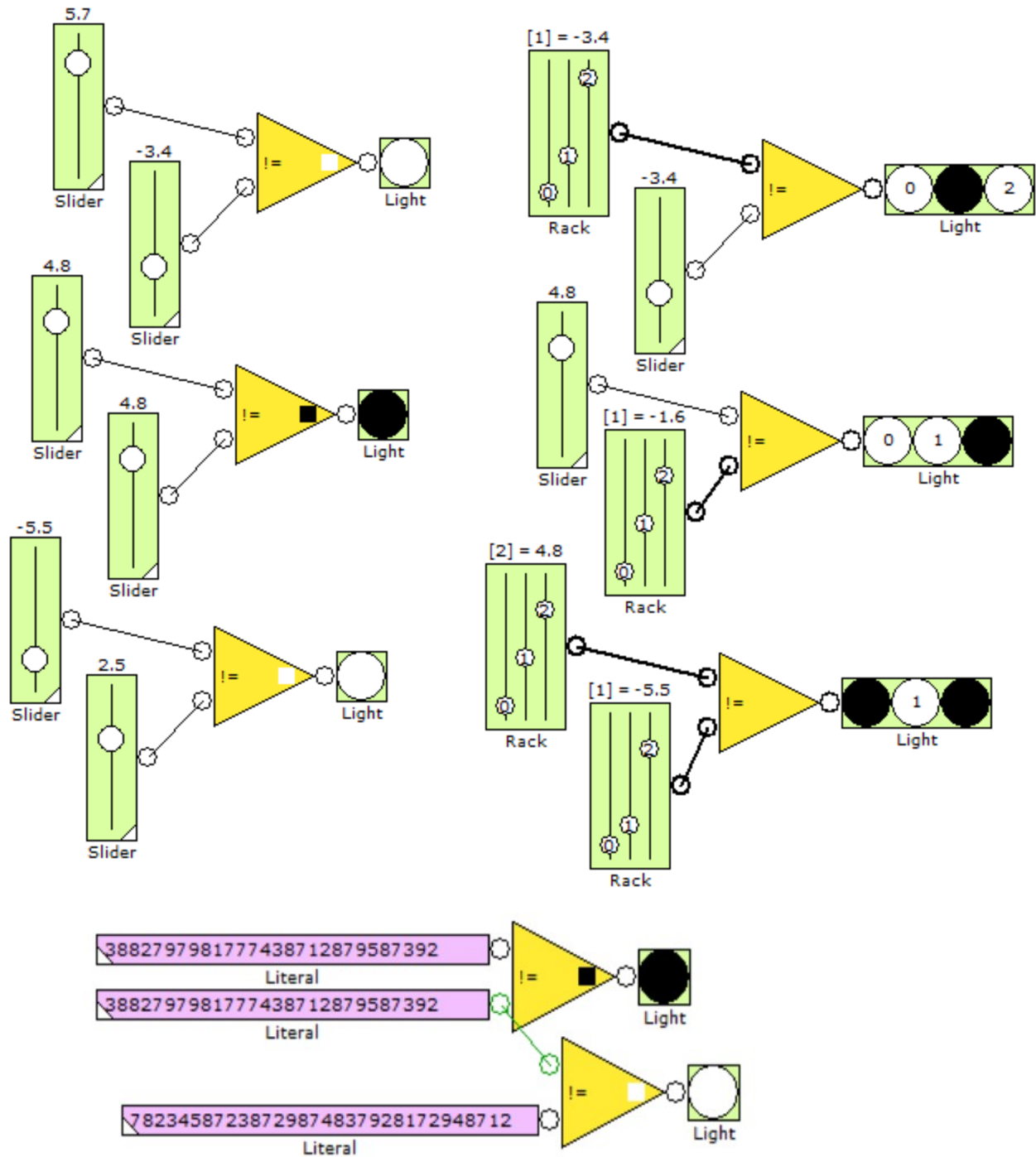
Image Gaussian .....	294	Key Send .....	343	2D Marker .....	394
Image Gradient .....	295	2D Keyboard .....	344	Matrix .....	395
Image Insert .....	296	Keyboard .....	345	Matrix Determinant .....	396
Image Invert .....	297	Knob .....	346	Matrix Excise .....	397
Image Line .....	298	2D Knob .....	347	Matrix Get At .....	398
Image Median Blur .....	299	Knob Factor .....	348	Matrix Interpolate .....	399
Image Moments .....	300	Knob Offset .....	349	Matrix Invert .....	400
Image Multiplexer .....	301	Largest .....	350	Matrix Joiner .....	401
Image Polygon .....	303	Last N .....	351	Matrix Mirror .....	403
Image Reader .....	304	Lathe Shape.....	352	Matrix Multiply .....	404
Image Resize .....	305	3D Lathe .....	353	Matrix Rotate .....	405
Image Rotate .....	306	Lathe Path .....	355	Matrix Splitter .....	406
Image Size .....	307	LatLon to XYZ .....	356	Matrix Subset .....	408
Image Subtract .....	308	Layer Slider.....	357	Matrix Transpose .....	409
Image Text .....	309	2D Length.....	358	Max .....	410
Image Threshold .....	310	2D Lens.....	359	Max Index .....	411
Image to 2D View.....	311	3D Lens .....	360	Maximum .....	412
Image to Data[3] .....	312	Letter Boxed.....	361	Median .....	413
Image Trace Edge .....	313	3D Life Cycle.....	362	Memory.....	414
Image Writer .....	314	LIFO .....	363	Merge .....	415
In Polygon .....	315	Light .....	364	Mesh .....	416
In Range .....	316	Light 2 .....	365	2D Mesh Plot.....	417
Index to Position .....	317	3D Light .....	366	Meter.....	418
Inflector.....	318	2D Light Ray.....	368	2D Meter .....	419
Inner Joiner .....	319	3D Light Ray.....	369	MIDI Percussion .....	420
Inner Mixer .....	320	2D Light Sensor.....	370	MIDI Pitch .....	421
Inner Product .....	321	3D Light Sensor.....	371	MIDI Play File .....	422
Inner Repeat .....	323	Limit .....	372	MIDI Play Note .....	423
Inner Sum .....	324	Line Clipper.....	373	Min .....	424
Input .....	325	3D Line .....	374	Min Index .....	425
Inst. Harmonics.....	326	2D Line Segment .....	375	Minimum .....	426
Integrator .....	327	3D Line to Line.....	376	Minute .....	427
2D Intensify.....	328	Linear Regression .....	377	Mirror .....	428
Interpolate .....	329	2D Linear Slider .....	378	Mode .....	429
Intersect Circles .....	330	List.....	379	Modulo .....	430
Intersect Line/Circle .....	331	Literal .....	380	Month .....	431
Intersect Line/Sphere .....	332	Loan .....	381	Moon Phase .....	432
Intersect Lines .....	333	Log .....	382	Morse Code .....	433
Intersect Rays.....	334	Log Scale .....	383	Morse Reader .....	434
Intersection .....	335	Logic .....	384	2D Motor.....	435
Invert .....	336	Low Case .....	387	3D Motor.....	436
Joiner .....	337	2D Magnify.....	388	2D Mouse.....	437
2D Kaleidoscope.....	338	Mailslot Client.....	389	Mouse Click.....	438
Keep .....	339	Mailslot Server.....	390	Mouse Get.....	439
Keep First .....	340	Manage Text File.....	391	Mouse Set.....	440
Keep Last .....	341	Map.....	392	Multiplexer .....	441
Keep Unique.....	342	Map Projection.....	393	Multiply .....	442

Multiply Fractions .....	443	Polygon Area.....	491	2D RGB Tint.....	539
n-Tuple.....	444	Polygon To Vector.....	492	RGB to HSV .....	540
Name to Type.....	445	Polynomial .....	493	Ring Counter .....	541
NAND.....	446	Polynomial Derivative .....	494	Roll .....	542
Negate .....	447	Polynomial Integral .....	495	2D Rotate.....	543
NOR .....	448	Poser.....	496	3D Rotate.....	544
NOT .....	449	Power .....	497	Rotation Slider.....	545
2D Number Spinner .....	450	Precision .....	498	Round .....	546
Number to Bits .....	451	Primes .....	499	2D Round Button .....	547
Numerical Index .....	452	3D Prism.....	500	RT Clock .....	548
Numerical Rank .....	453	Pulse 2 Generator .....	501	RT Filter.....	549
Numerical Sort .....	454	Pulse Generator .....	502	Running .....	550
One Shot .....	455	Pulse Width .....	503	Sample .....	551
Only in A .....	456	Put At .....	504	Savings .....	552
Only in B .....	457	3D Pyramid.....	505	Sawtooth .....	553
OR .....	458	QR Generator.....	506	Scalar Counter.....	554
Orthographic .....	459	Quadratic .....	507	Scale .....	555
Outer Product .....	460	r-permute .....	508	3D Scale .....	556
Outer Sum .....	461	Rack .....	509	Second .....	557
Output .....	462	Rack Factor .....	510	Seconds to Date.....	558
2D Oval .....	463	Rack Offset .....	511	Scramble .....	559
Palindrome .....	464	Radio .....	512	Segmenter .....	560
2D Parabola .....	465	2D Radio Group .....	513	Selector .....	561
Parabolic Path .....	466	Random Choice .....	514	Sequencer .....	563
Path Interpolate .....	467	Random Key Gen.....	515	Serial Port .....	564
Pattern.....	468	Random Position .....	516	3D Serpentine.....	565
2D Pendulum .....	469	Random Read File .....	517	3D Shaft .....	566
Periodic .....	470	Random Seed .....	518	Shape Arranger.....	567
Permute .....	471	Random String.....	519	Shape Bender.....	568
Persistent.....	472	Random Write File .....	520	Shape Expander.....	569
3D Pipe .....	473	2D Ratchet.....	521	Shape Joiner.....	570
3D Plane Surface .....	474	3D Ratchet .....	522	Shape Outliner.....	571
3D Plane Surface 2 .....	475	Ravel .....	523	Shape Presser.....	572
Planet .....	477	2D Ray Sensor.....	524	Shape Shifter.....	573
Play Sound File .....	478	Read Data File .....	525	Shape Store.....	574
Point to Ellipse .....	479	2D Read Image.....	526	Shape Twister.....	575
Point to Line .....	480	Read Text File .....	527	Shell Command .....	576
3D Point to Line .....	481	Read URL.....	528	2D Shift .....	577
2D Pointer .....	482	Read Zip Directory.....	529	3D Shift .....	578
3D Pointer .....	483	Reasonable .....	530	Shuffle .....	579
Points to Delta.....	484	Recorder.....	531	Sign .....	580
Polar to XY .....	485	Reduce .....	533	Simplex Solver .....	581
3D Poly .....	486	2D Reflector.....	534	Sine .....	582
2D Poly Line .....	487	3D Reflector .....	535	2D Sine Cosine .....	583
Poly Synth .....	488	2D Remap Color.....	536	Sine Wave.....	584
2D Polygon .....	489	Replace.....	537	Slide Factor .....	585
3D Polygon .....	490	Reset Clock .....	538	Slide Offset.....	586

Slider .....	587	Sun Position .....	633	Value Terms .....	682
Slope Intercept.....	588	Sunrise .....	634	Value To Digits .....	683
Smallest .....	589	3D Surface.....	635	Variance .....	684
Snap Point .....	590	Switch .....	637	2D Vector .....	685
Snap Point+ .....	591	Switch 2.....	638	Vector Angle .....	686
Solid Color .....	592	Synthesizer .....	639	Vector Delta .....	687
Sort.....	593	Table .....	640	Vector Length .....	688
Speak.....	594	2D Tachometer.....	641	Vector Scale .....	689
3D Sphere .....	595	Tangent .....	642	Vector To Affine .....	690
3D Sphere Surface.....	597	Terminal.....	643	Version .....	691
3D Sphere Surface 2 .....	599	2D Text .....	644	2D Vertical Axis .....	692
Spherical Distance .....	600	2D Text Entry .....	645	2D Vertical Slider .....	694
2D Spinner .....	601	2D Thin Lens.....	646	Vertical Stripes .....	695
Splice In .....	602	3D Thin Lens .....	647	Video Reader .....	696
Splice Out .....	603	Tic and Label .....	648	Video Writer .....	697
Spline.....	604	Time .....	649	2D View Control .....	698
Spline Gen .....	605	Time Random .....	650	3D View Control .....	699
Spline XY .....	606	Time Step .....	651	3D Wheel .....	700
Splitter .....	607	Timer .....	652	2D Window Control .....	701
Sprocket.....	608	Timer 2 .....	653	Wire.....	702
Sqrt .....	609	To ASCII .....	654	Word Finder.....	703
Square .....	610	To Char .....	655	Wordus.....	704
2D Square Button .....	611	To HM .....	656	World Cities .....	705
Square Wave .....	612	To HMS .....	657	Write Data File .....	706
Stack CPU.....	613	To HSV .....	658	Write Text File .....	707
Standard Deviation .....	614	To Number .....	659	3D X Rotate .....	708
Start Clock .....	615	To RGB .....	660	XOR .....	709
Step Clock .....	616	To String .....	661	2D XY Axis .....	710
2D Stepper Motor .....	617	Toggle .....	662	XY Chart .....	711
Stop Clock .....	618	Tomograph X Y.....	663	2D XY Chart .....	712
Stopwatch.....	619	Tomograph Z.....	664	XY Recorder .....	713
String 2D Width .....	620	3D Torus .....	665	XY Rotate .....	714
String Joiner .....	621	Tracker.....	668	XY Scribble .....	715
String Length .....	622	3D Translation Slider .....	669	XY Slide Factor .....	716
String Parse .....	623	Triangle Area .....	670	XY Slide Offset .....	717
String SHA-256.....	624	Triangular Wave .....	671	XY Slider .....	718
String to Binary .....	625	Triggered Random .....	672	XY to Polar .....	719
String to Characters .....	626	Triggered Sum .....	673	XYZ to LatLon .....	720
String to Hex .....	627	3D U-Joint .....	674	XYZ Transform.....	721
String to Key.....	628	Union .....	676	3D Y Rotate .....	722
3D Stroke Text.....	629	Up Case .....	677	Year .....	723
Sub String .....	630	US Cities .....	678	3D Z Rotate .....	724
Subset .....	631	UTF-8 .....	679		
Subtract .....	632	2D Value .....	680		

# != (not equal to)

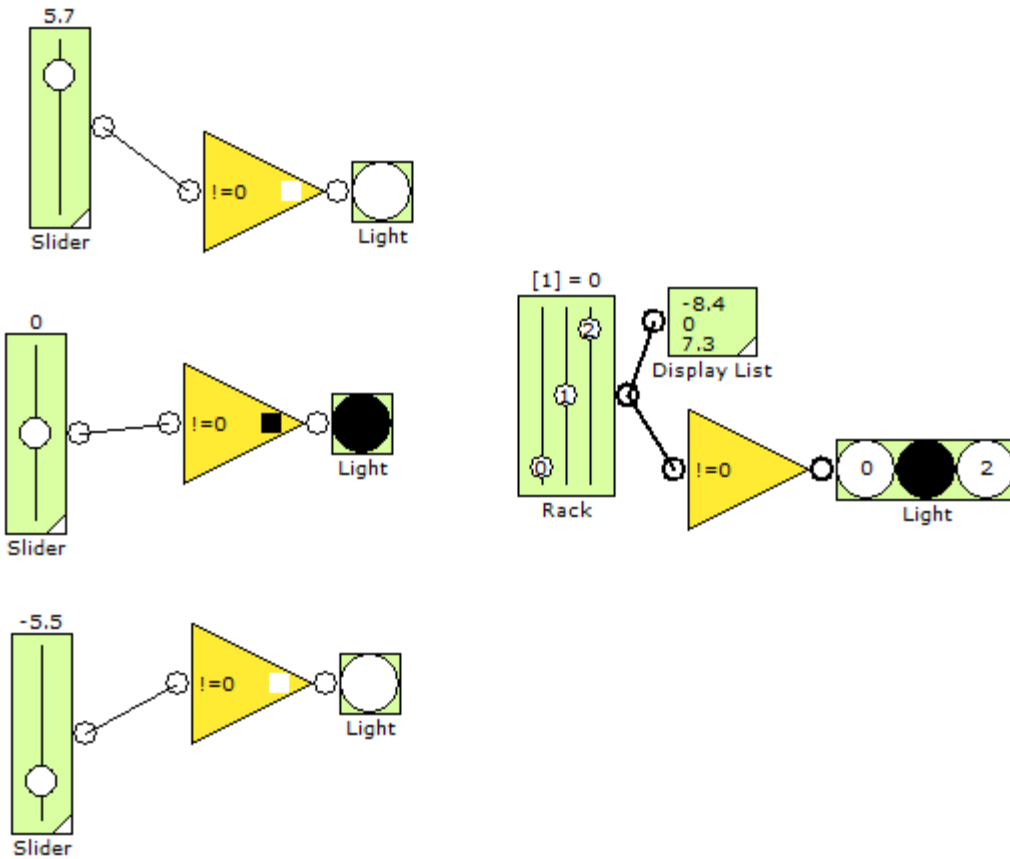
The != component compares the inputs and returns true if the top value is not equal to the bottom value. Inputs can be scalars or arrays. Functions - column 4





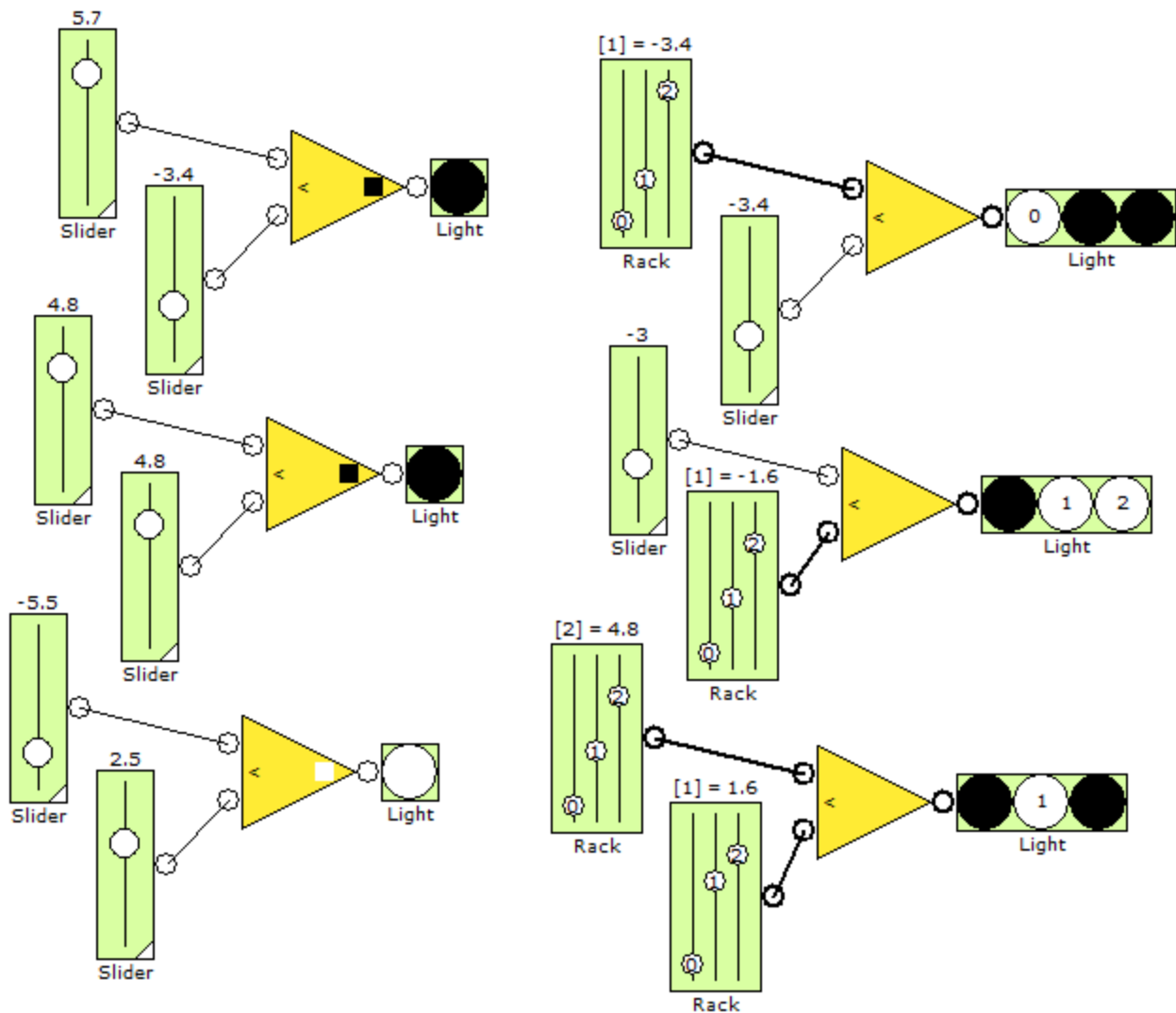
## !=0 (not equal to zero)

The !=0 component returns true if the input value is not equal to zero. Input can be scalar or an array.  
Functions - column 4



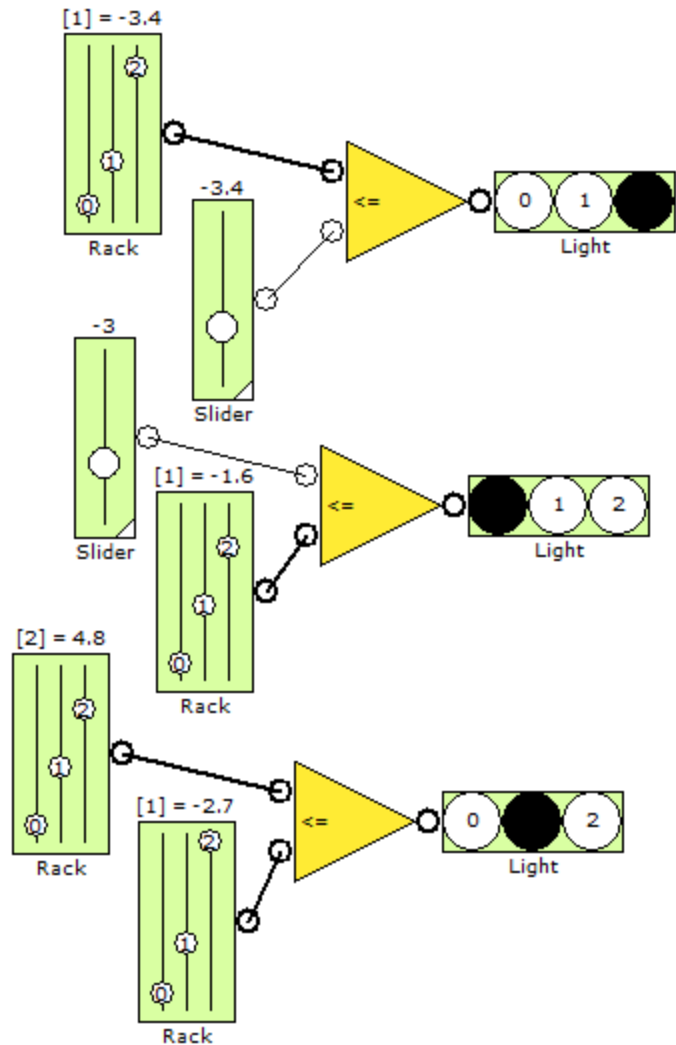
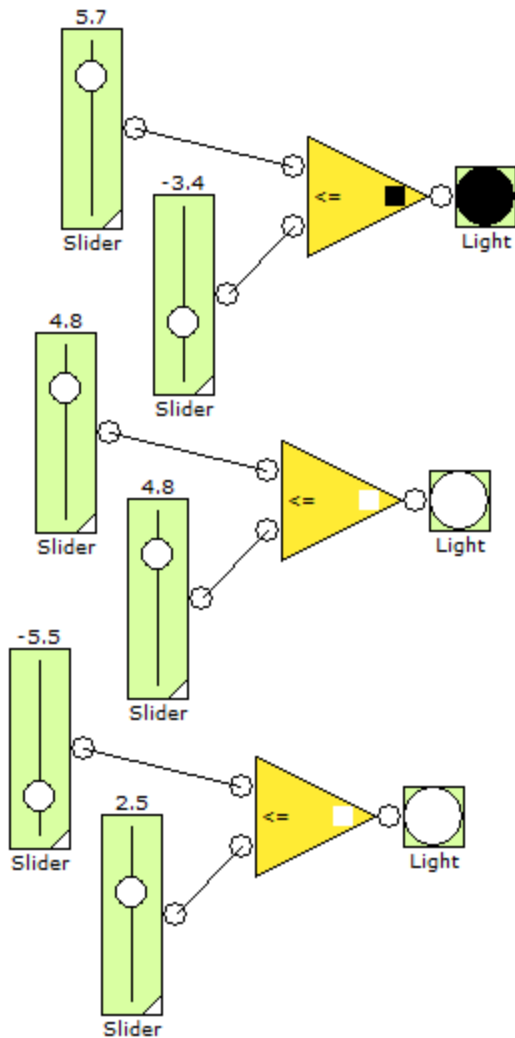
## < (less than)

The < component compares the inputs and returns true if the top value is less than the bottom value. Inputs can be scalars or arrays. Functions - column 4



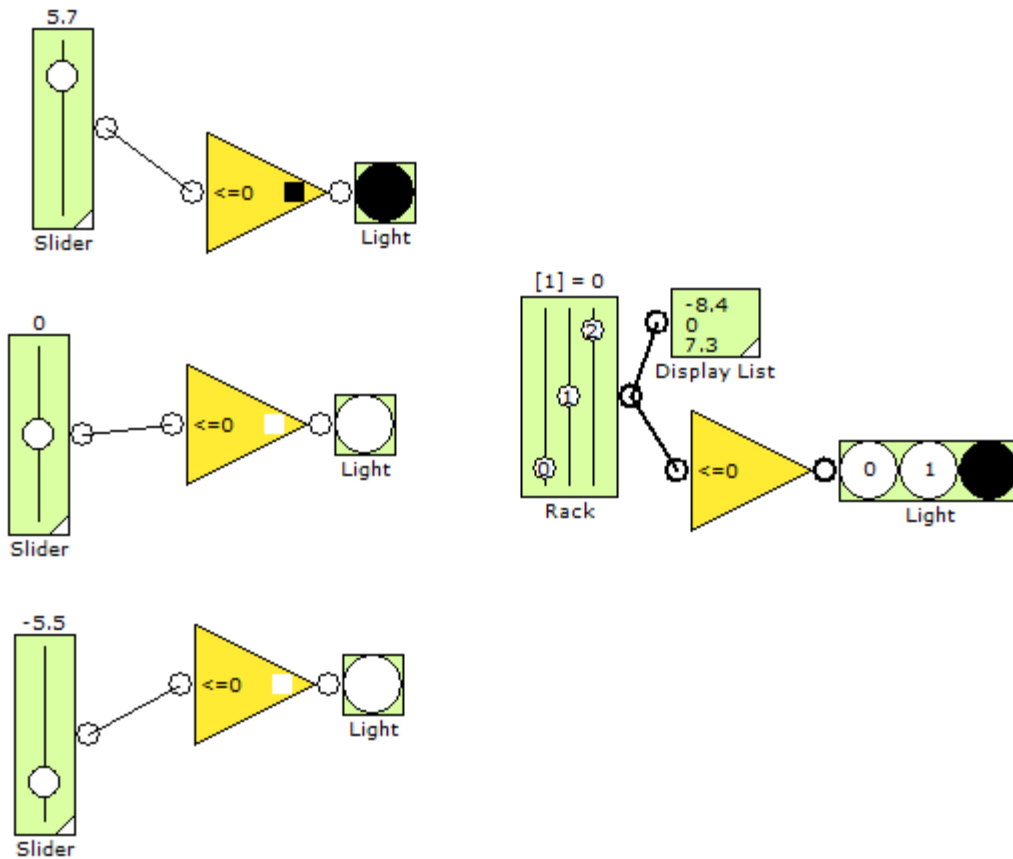
## <= (less or equal to)

The <= component compares the inputs and returns true if the top value is less than or equal to the bottom value. Inputs can be scalars or arrays. Functions - column 4

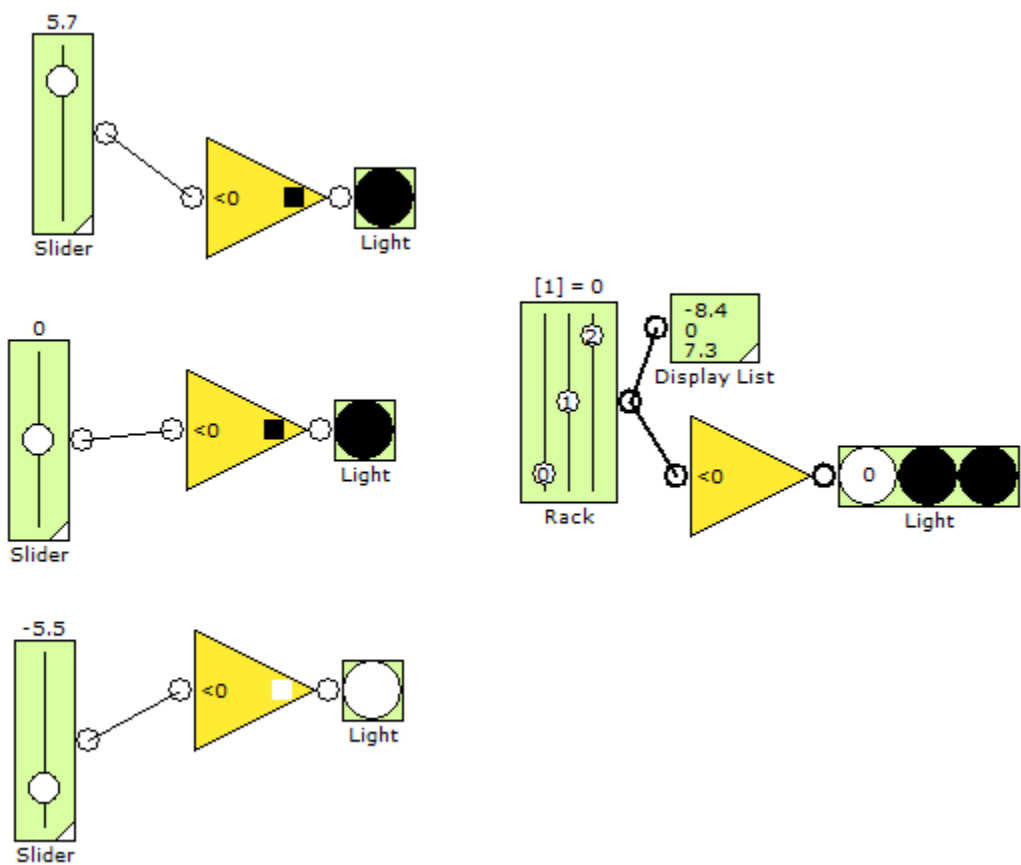


## **$\leq 0$ (less or equal to zero)**

The  $\leq 0$  component returns true if the input value is less than or equal to zero. Input can be scalar or an array. Functions - column 4



## <0 (less than zero)

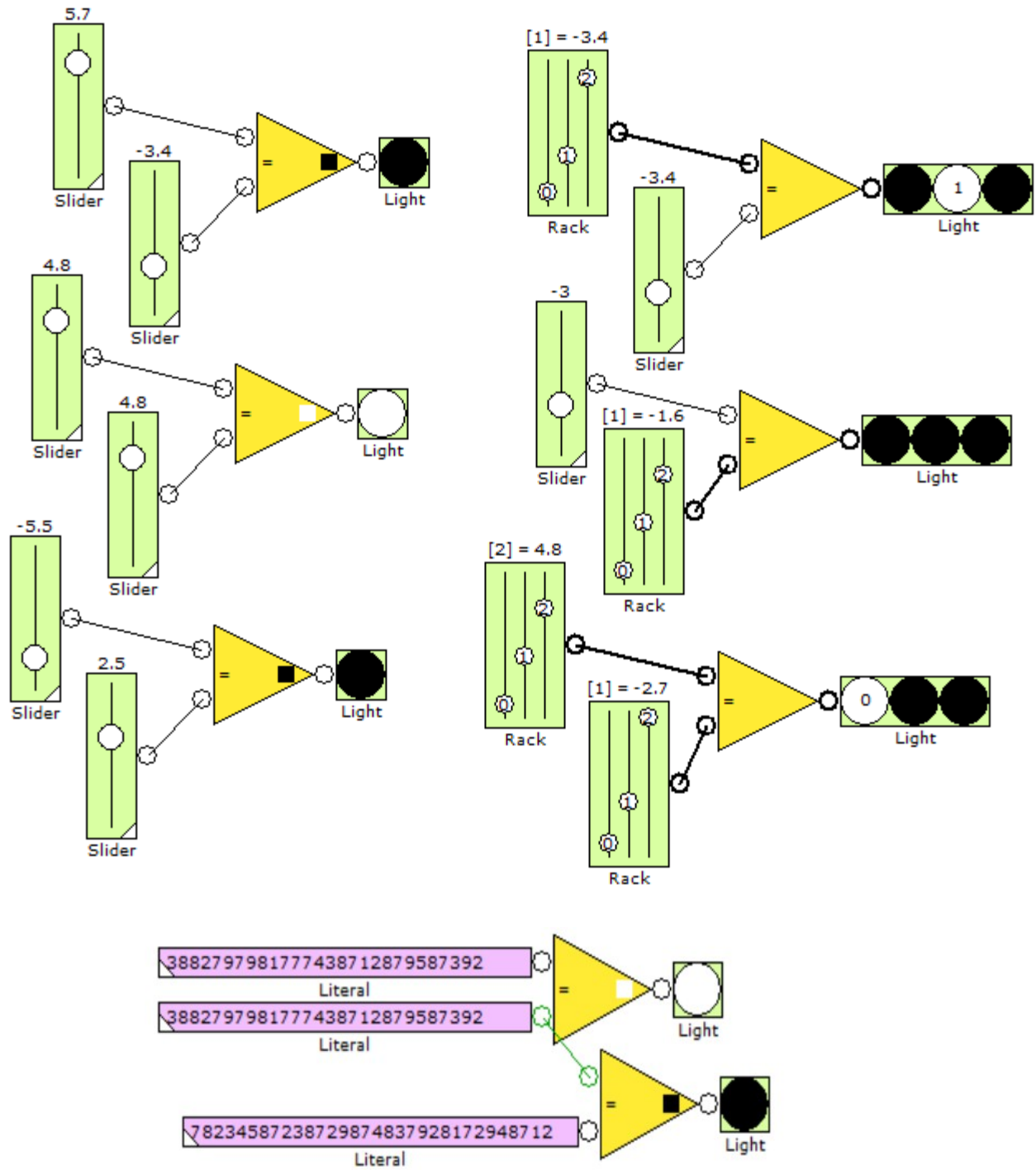


The <0 component returns true if the input value is less than zero. Input can be scalar or an array.

Functions - column 4

# = (equals)

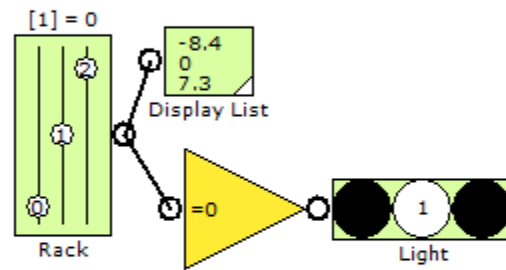
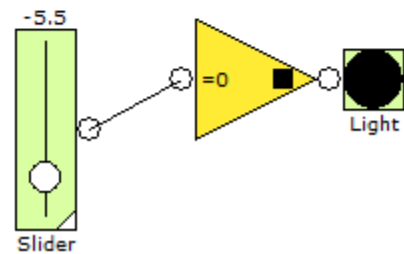
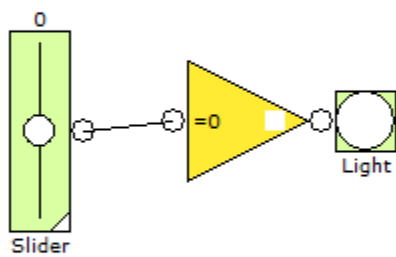
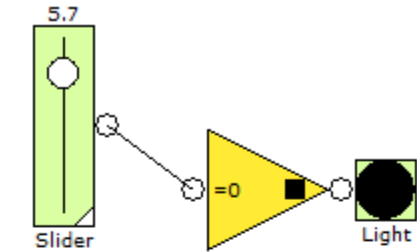
The = component compares the inputs and returns true if the top value is equal to the bottom value. Inputs can be scalars or arrays. Functions - column 4



## =0 (equals zero)

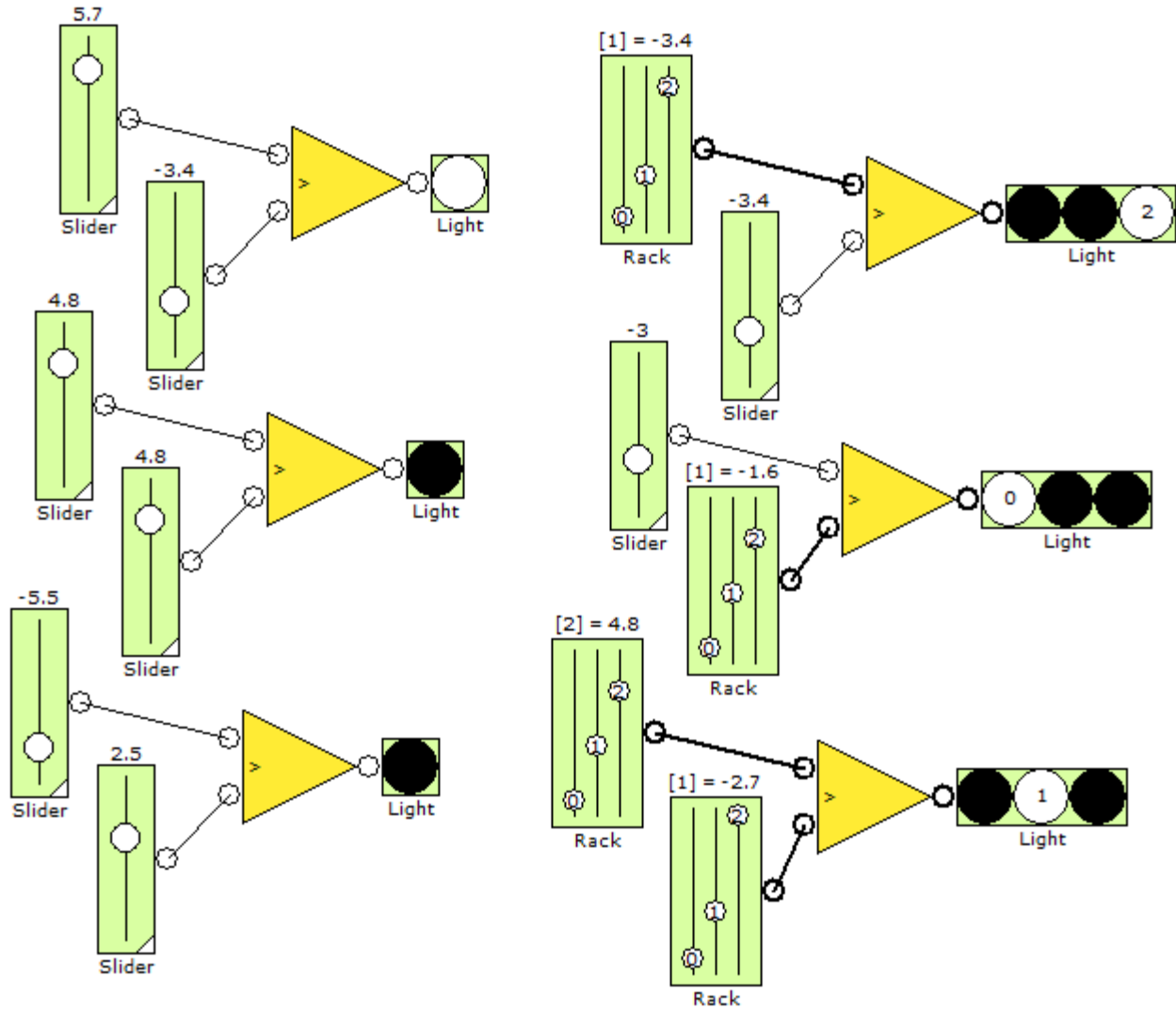
The =0 component returns true if the input value is equal to zero. Input can be scalar or an array.

Functions - column 4



## > (greater than)

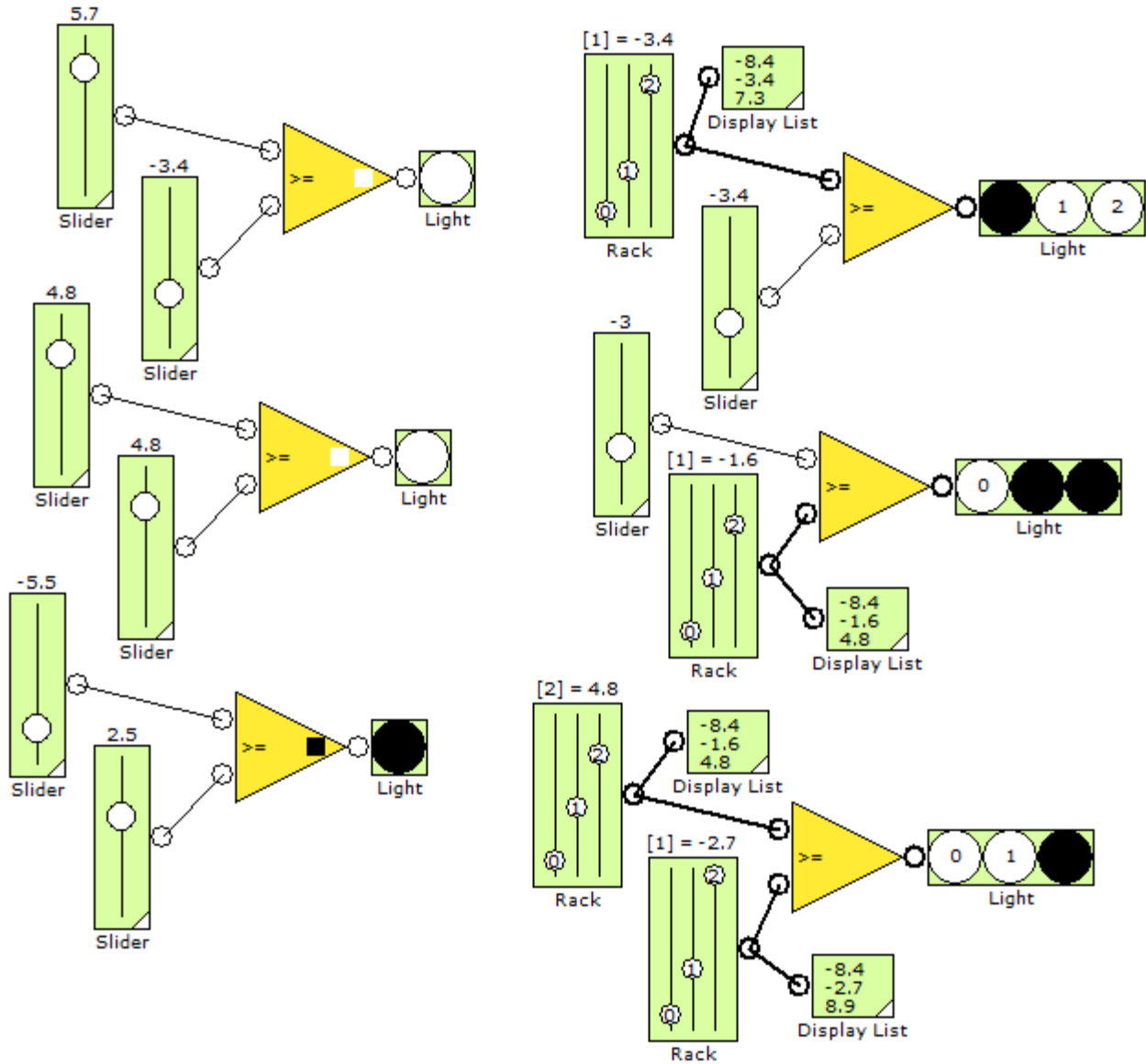
The > component compares the inputs and returns true if the top value is greater than the top value. Inputs can be scalars or arrays. Functions - column 4





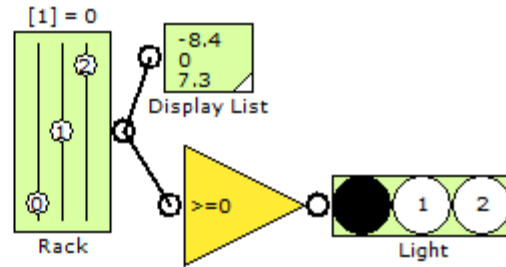
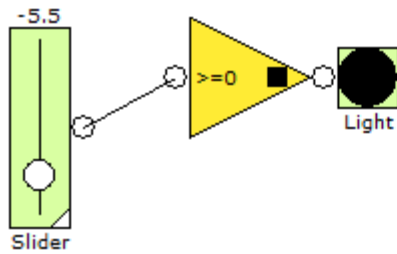
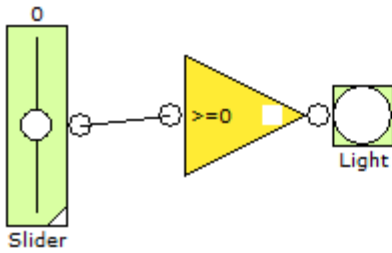
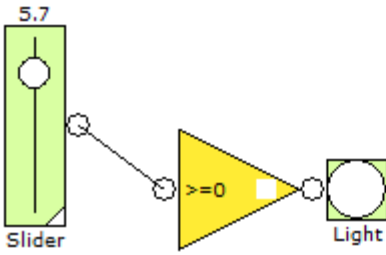
## >= (greater or equal to)

The >= component compares the inputs and returns true if the top value is greater than or equal to the bottom value. Inputs can be scalars or arrays. Functions - column 4



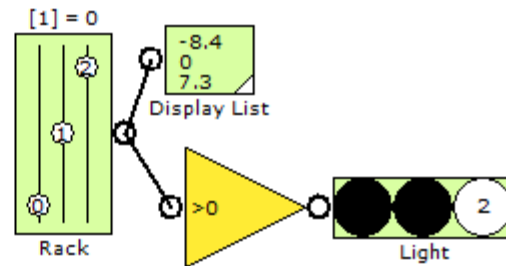
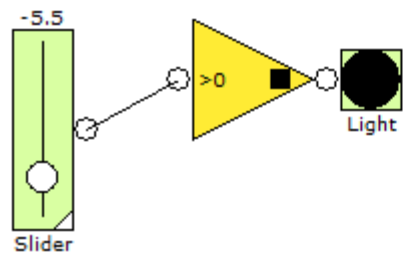
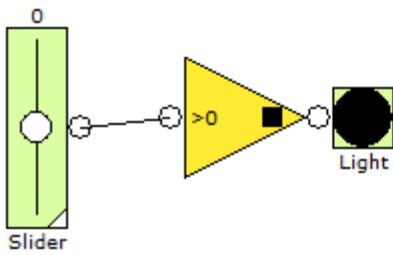
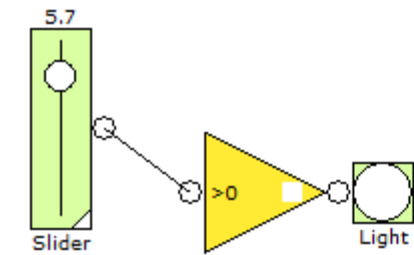
## $\geq 0$ (greater or equal to zero)

The  $\geq 0$  component returns true if the input value is greater than or equal to zero. Input can be scalar or an array.



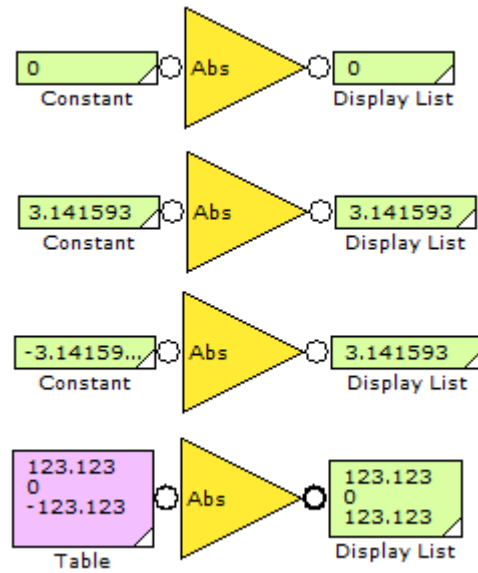
## >0 (greater than zero)

The >0 component returns true if the input value is greater than zero. Input can be scalar or an array.  
Functions - column 4



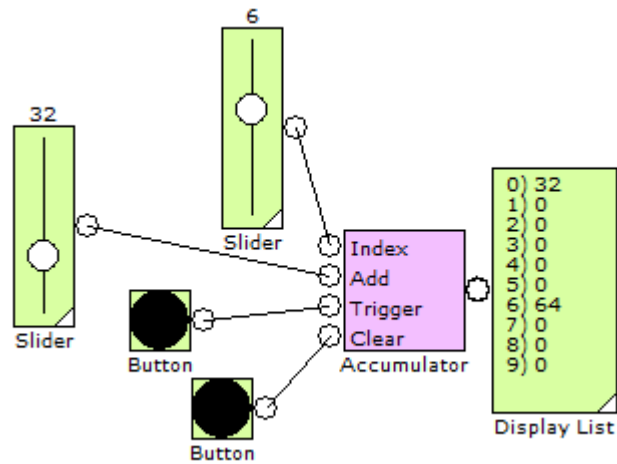
## Abs (absolute value)

The Abs component outputs the absolute value of the input value. Input can be scalar or an array.  
Functions - column 1



# Accumulator

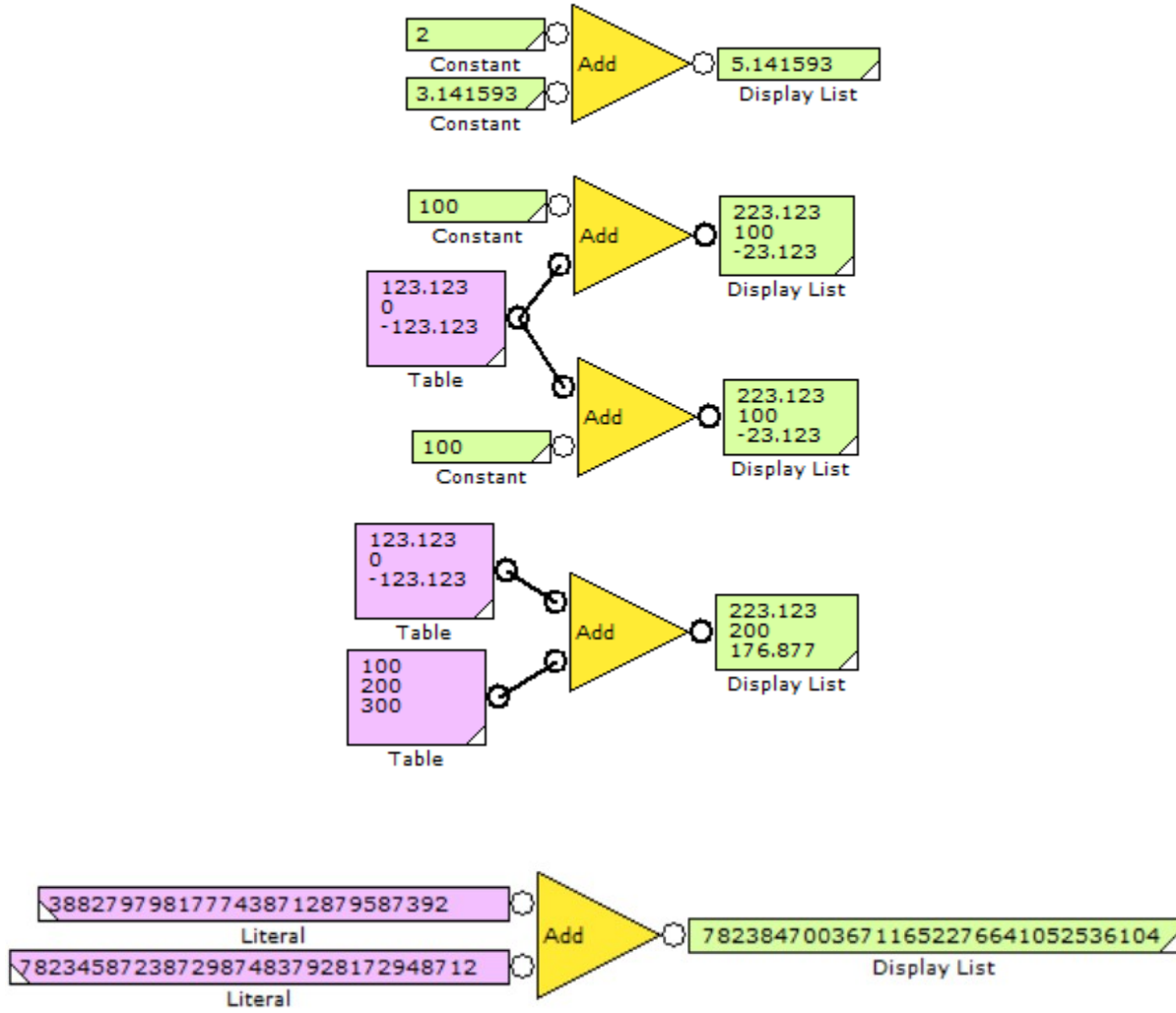
The Accumulator component accumulates data into an internal buffer and outputs the data. When the trigger goes true the value at the Add input is added to the total of the item in the Cell input. The cell number can be the index of any of the items in the array. You can put a negative number at the Add input to decrease the value in the array.      Arrays - column 1



# Add

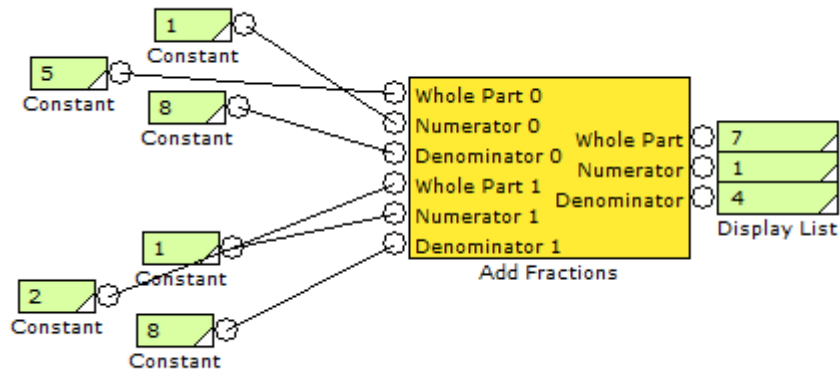
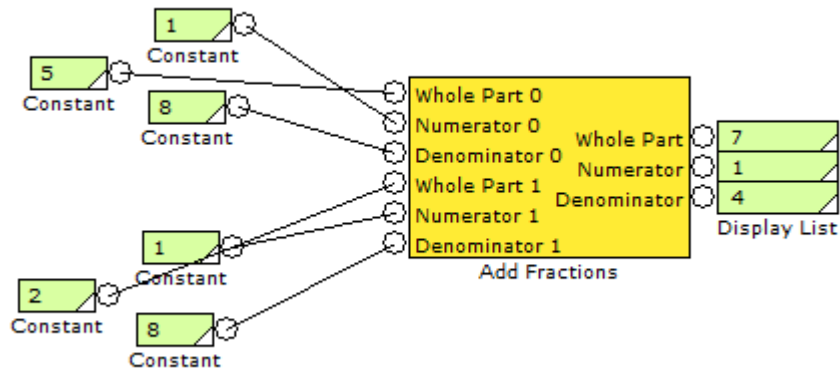
The Add component adds the inputs together. Inputs can be scalar and /or arrays.

Functions - column 1



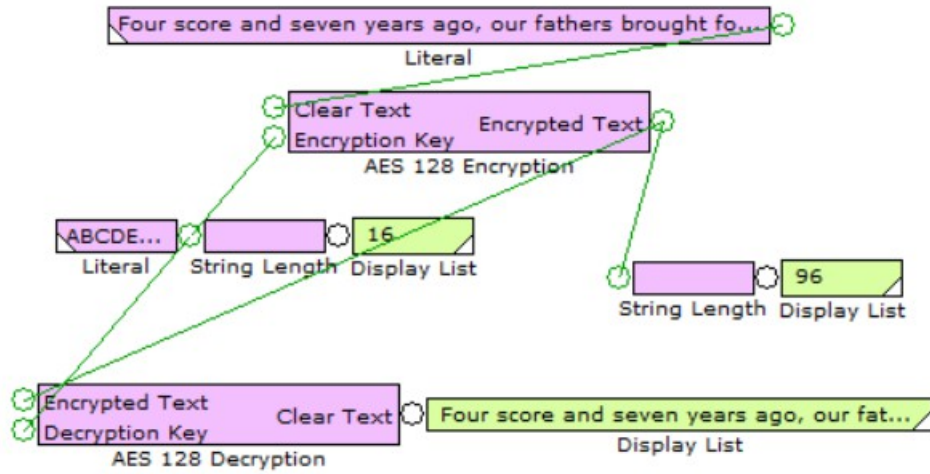
# Add Fractions

The Add Fractions component adds two numbers with fractions. The output is a whole number with a fraction as needed. Functions - column 9



# AES 128 Decryption

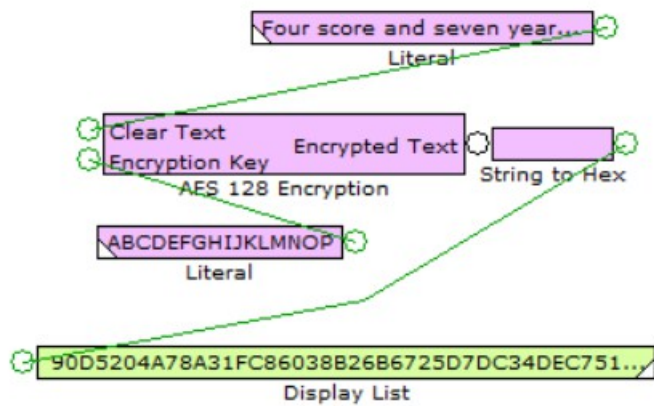
The AES 128 Decryption component decrypts the input string using AES 128 bit decryption. Arrays – column 5





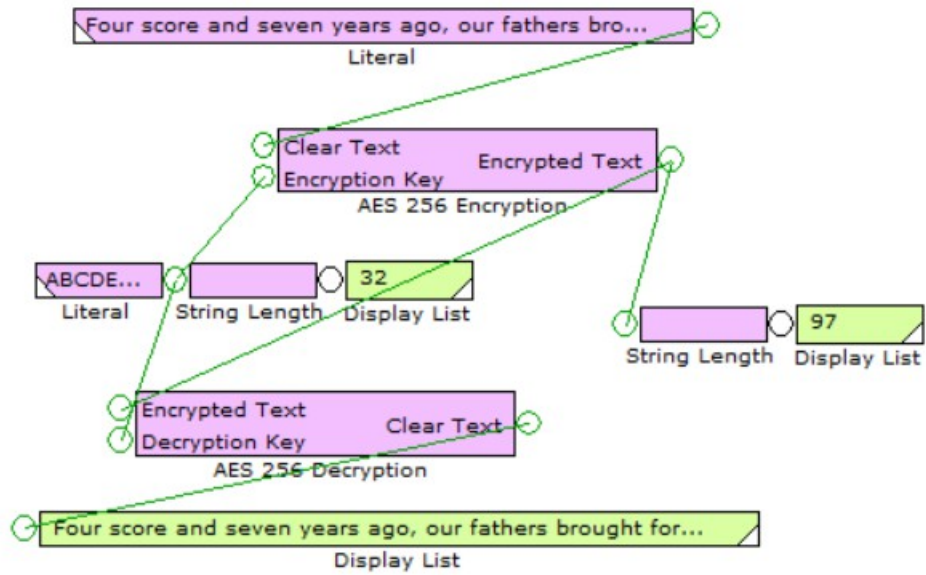
# AES 128 Encryption

The AES 128 Encryption component encrypts the input string using AES 128 bit encryption. Arrays – column 5



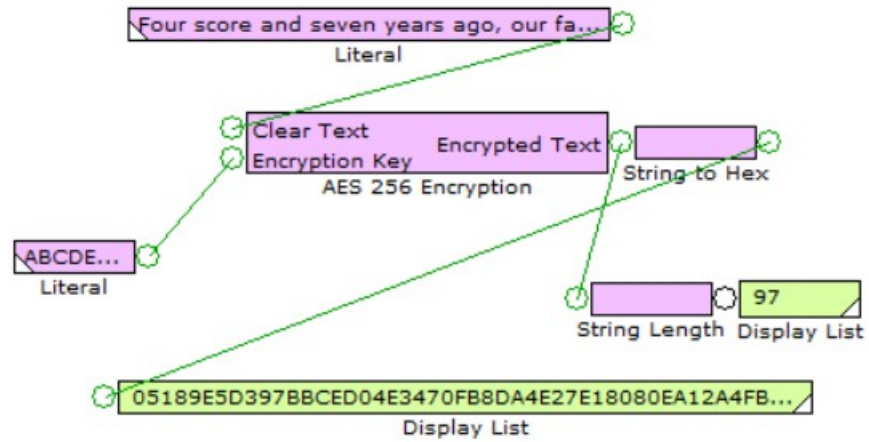
# AES 256 Decryption

The AES 256 Decryption component decrypts the input string using AES 256 bit decryption. Arrays – column 5



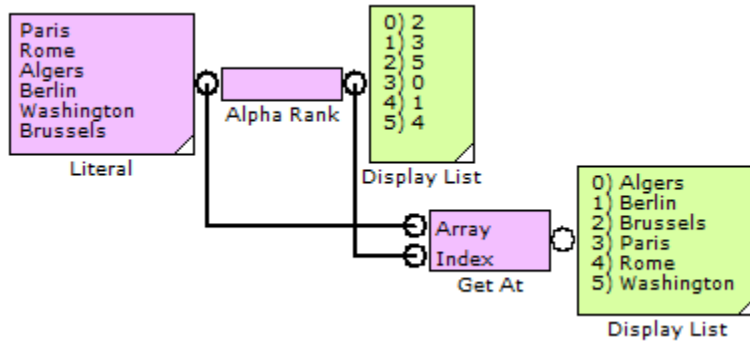
# AES 256 Encryption

The AES 256 Encryption component encrypts the input string using AES 256 bit encryption. Arrays – column 5



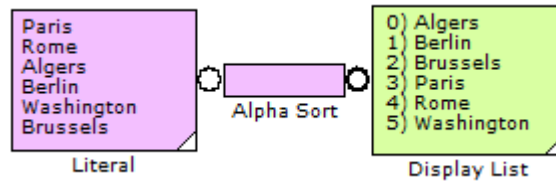
# Alpha Rank

The Alpha Rank component outputs a list of numbers in the order of the sorted alpha numerical input.  
Arrays - column 6



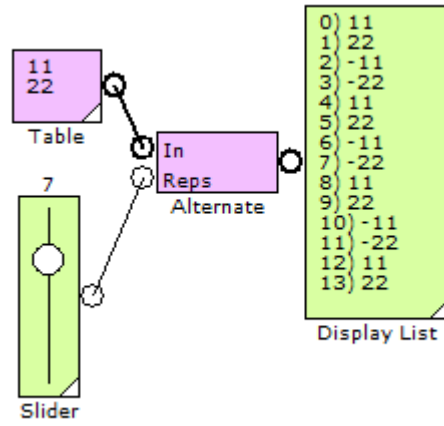
# Alpha Sort

The Alpha Sort component sorts the input strings into alpha numerical order. Arrays - column 6



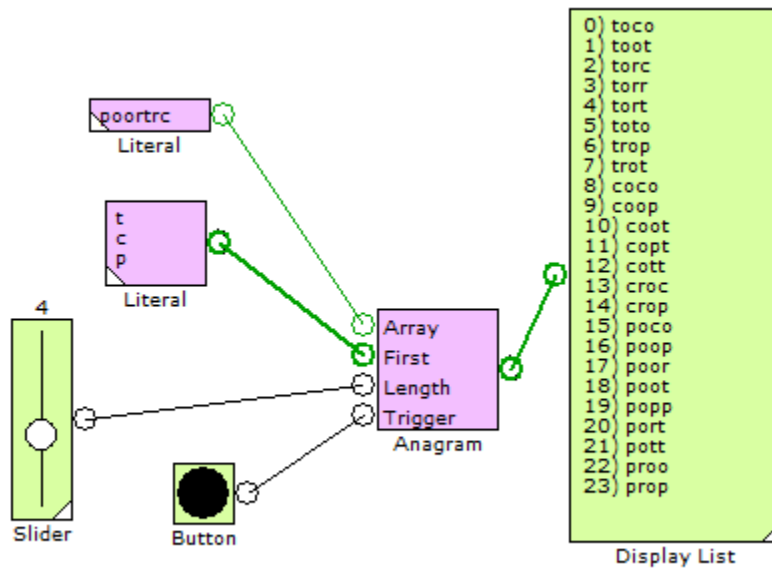
# Alternate

The Alternate component outputs the input alternated in sign repeatedly. Arrays - column 3



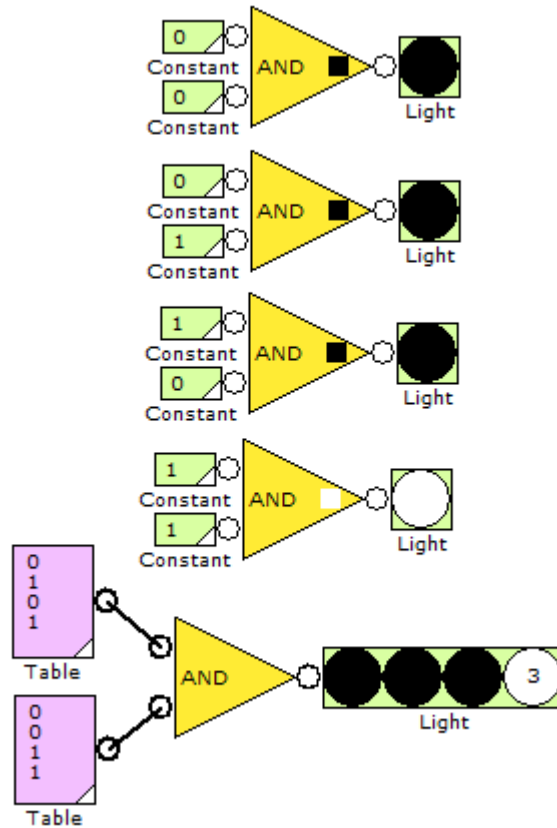
# Anagram

The Anagram lists anagrams of the input letters. Set Trigger to true to generate the list.



# AND

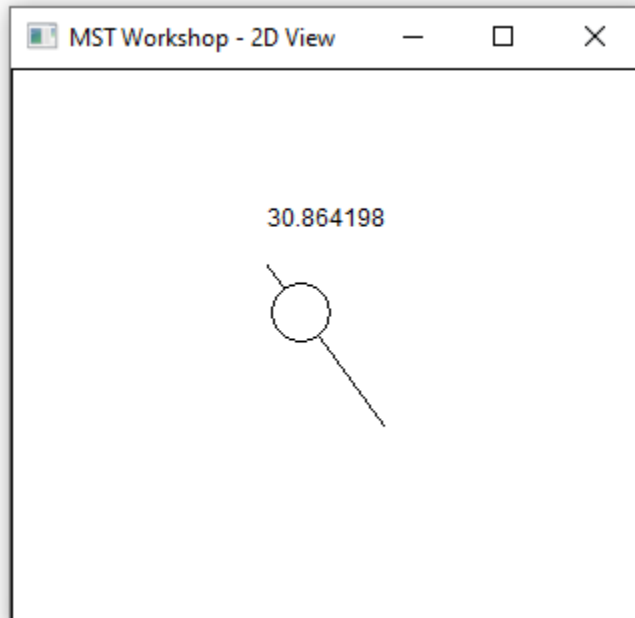
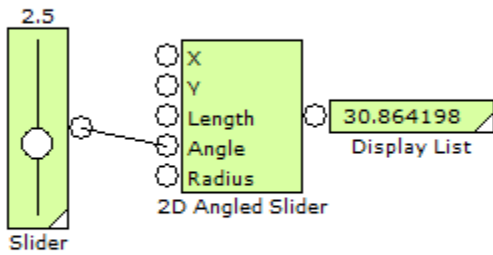
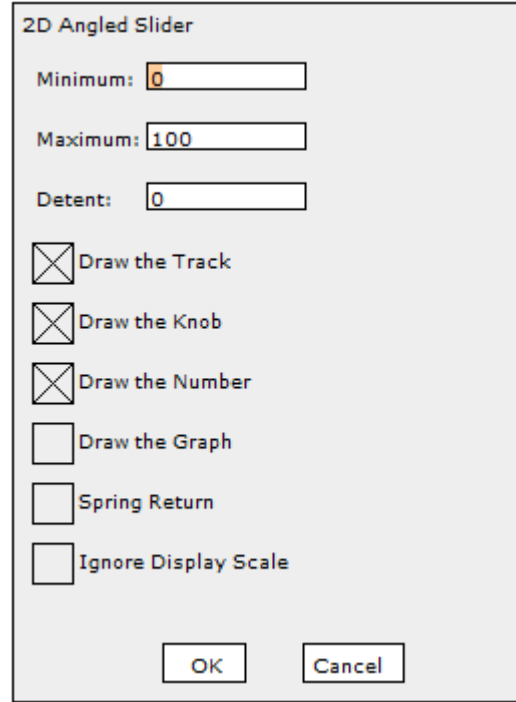
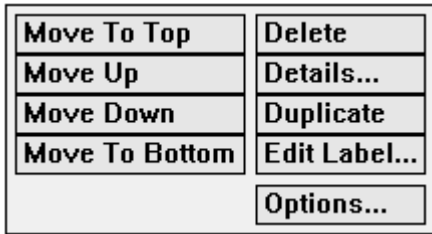
The AND component compares the inputs and returns true if both values are true. Inputs can be scalars or arrays. Functions - column 3





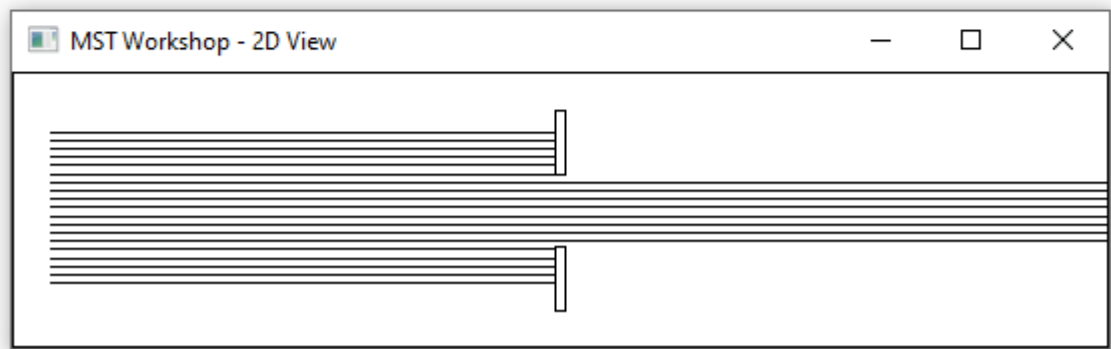
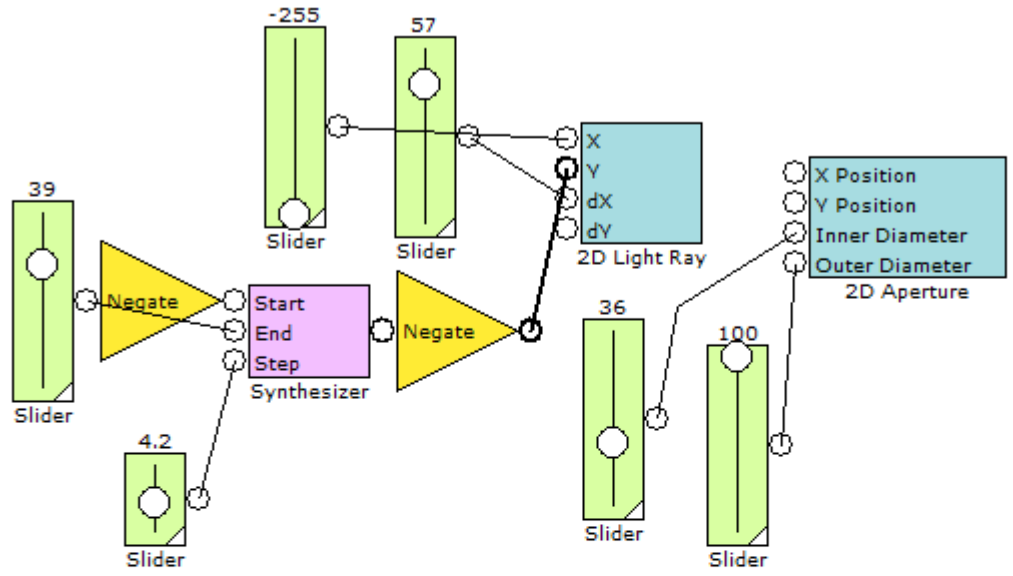
# 2D Angled Slider

The 2D Angled Slider component outputs a number based on its dial position. Controls - column 1



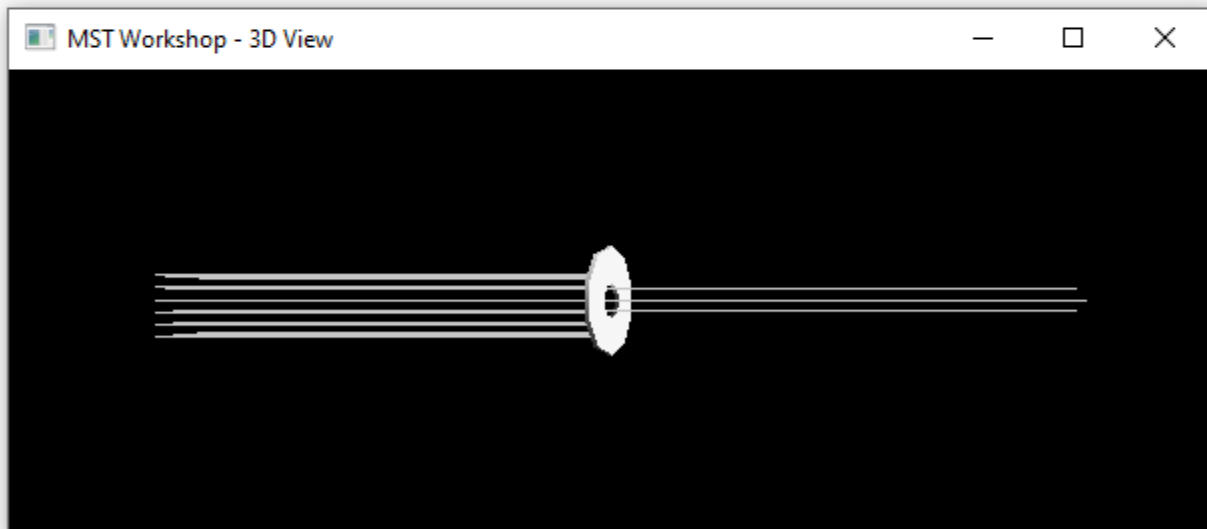
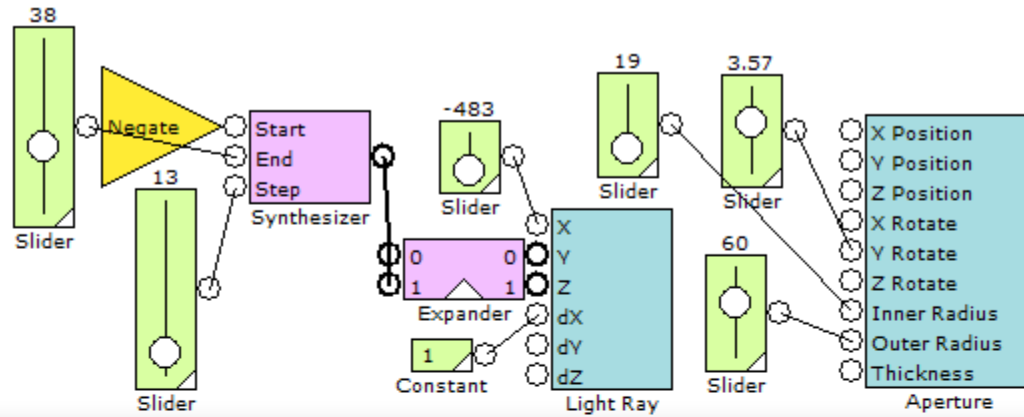
## 2D Aperture

The 2D Aperture component passes some light rays and block others in the 2D view window. The settings determine the size of the opening in the center of the aperture. Light will pass through the center but will be blocked by the rest of the aperture. 2D - column 3



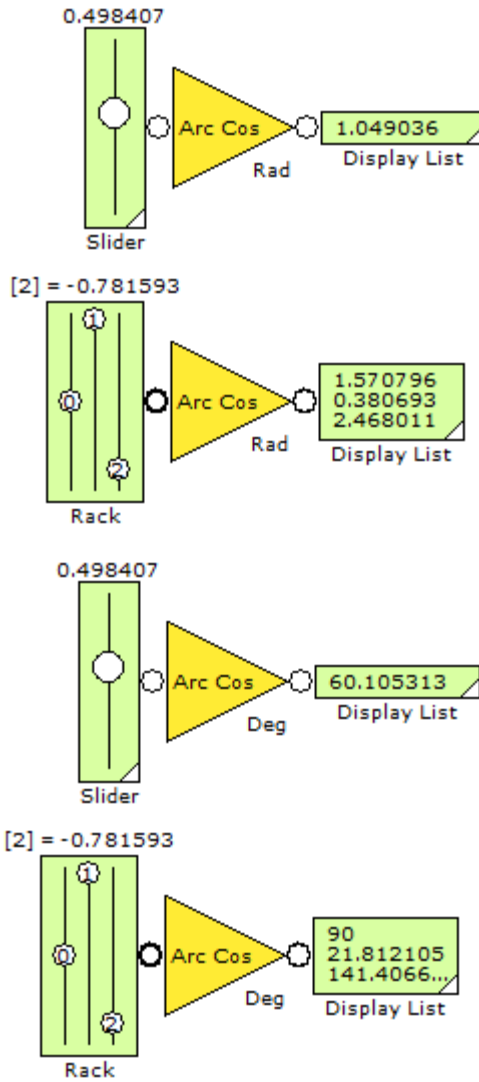
# 3D Aperture

The 3D Aperture component passes some light rays and block others in the 3D view window. The settings determine the size of the opening in the center of the aperture. Light will pass through the center but will be blocked by the rest of the aperture. The Affine [4x4] array input modifies the size, position and orientation of the aperture. 3D - column 5



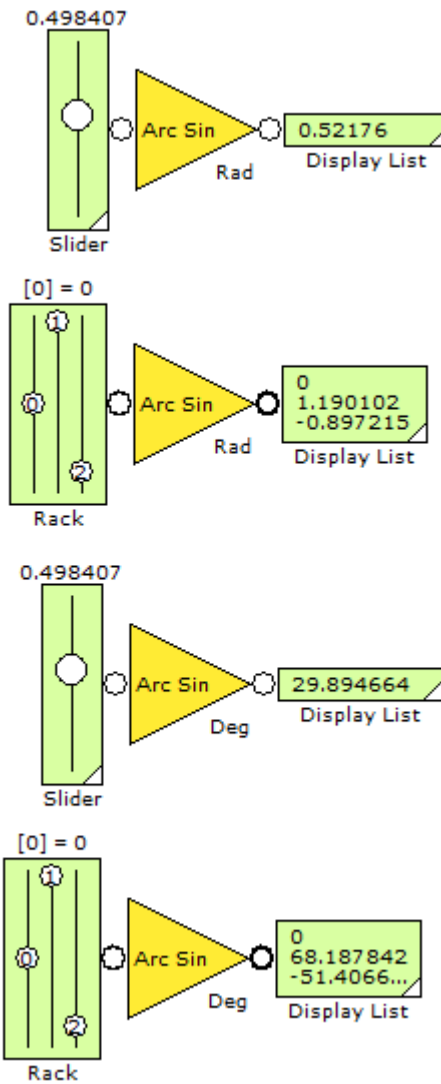
# Arc Cos

The Arc Cos component calculates (in radians or degrees) the inverse trigonometric cosine of the input. Input can be scalar or an array. Functions - column 5



# Arc Sin

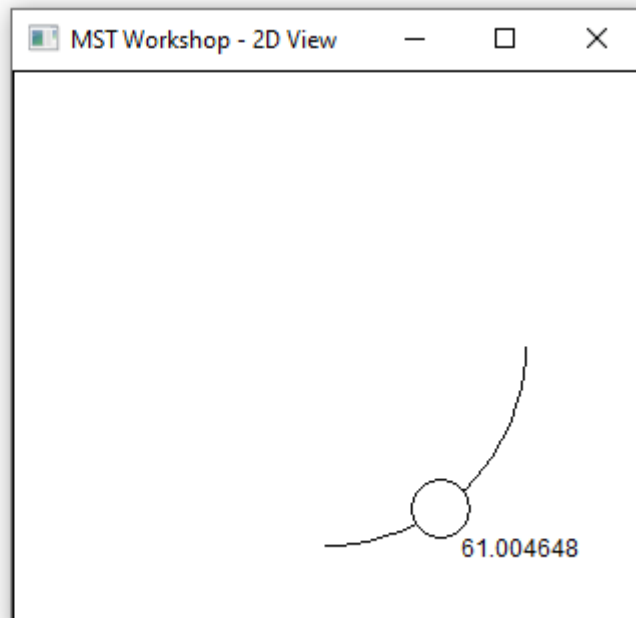
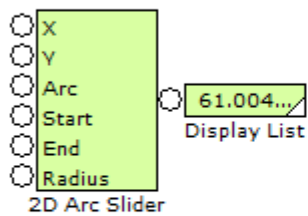
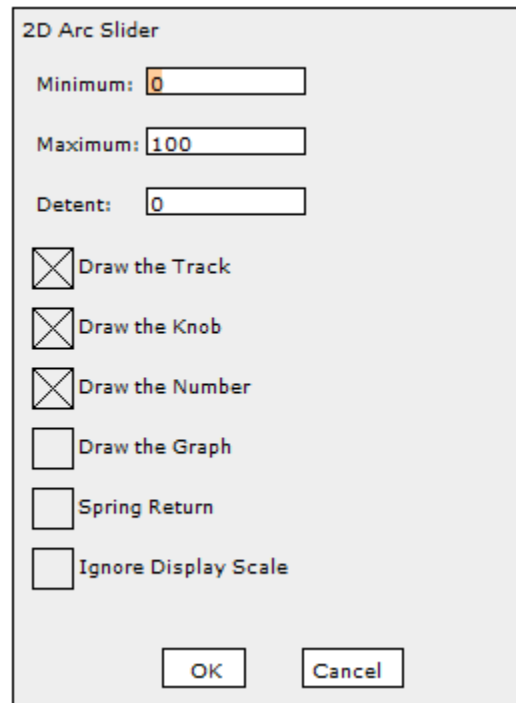
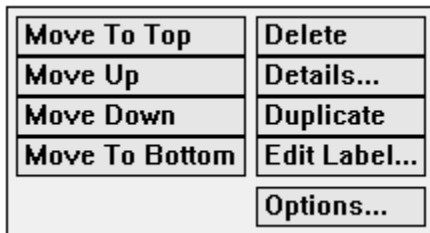
The Arc Sin component calculates (in radians or degrees) the inverse trigonometric sine of the input. Input can be scalar or an array. Functions - column 5



## 2D Arc Slider

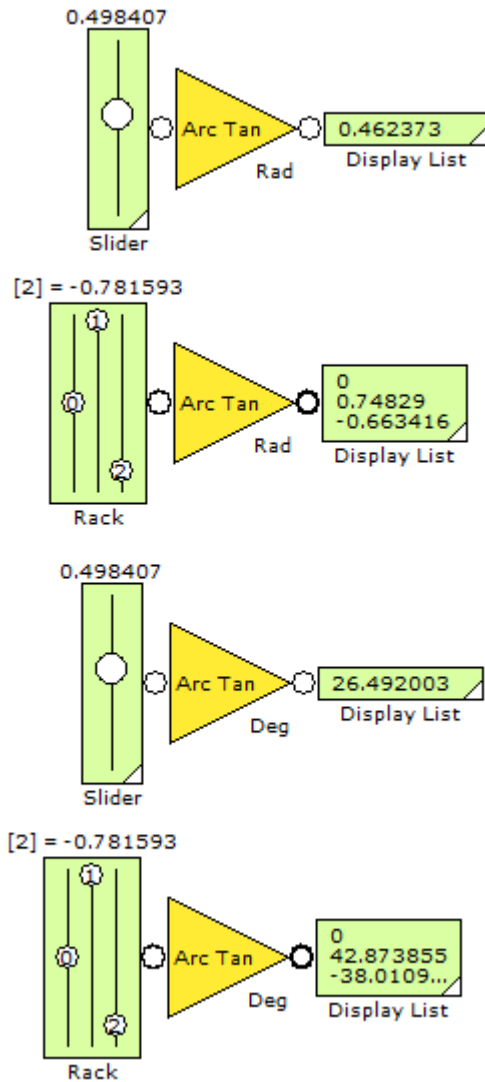
The 2D Arc Slider component outputs a number based on its dial position.

Controls - column 2



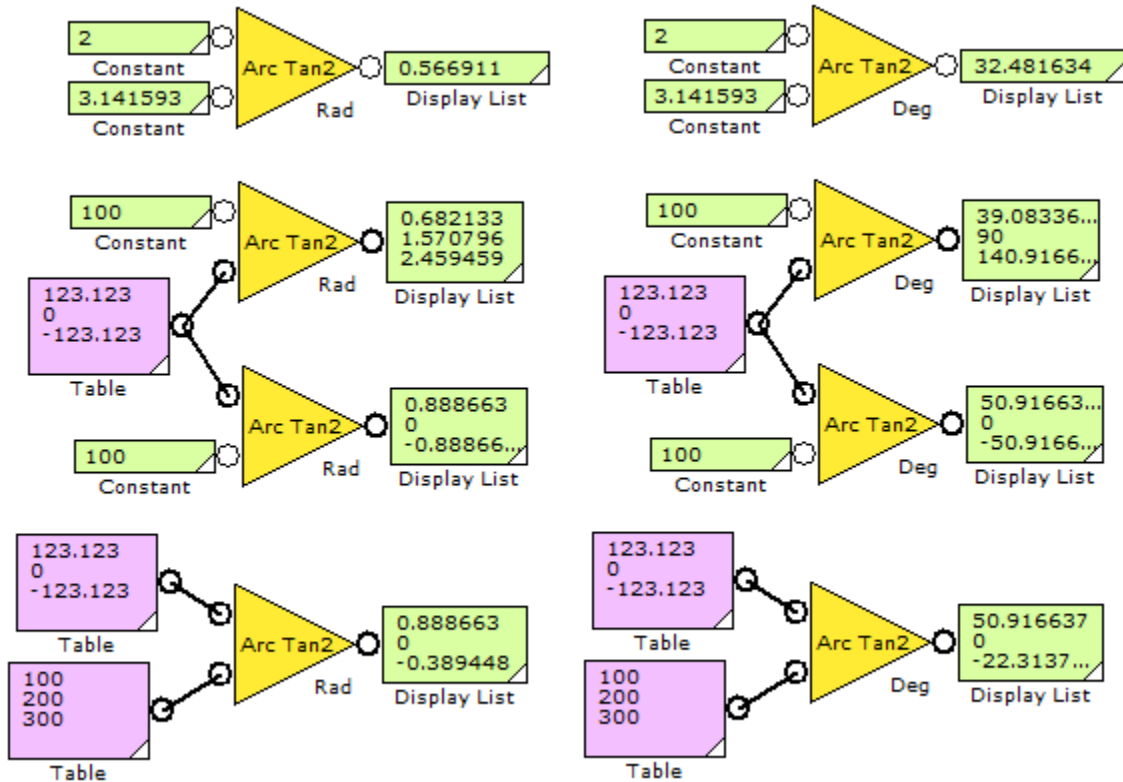
# Arc Tan

The Arc Tan component calculates (in radians or degrees) the inverse trigonometric tangent of the input. Input can be scalar or an array. Functions - column 5



# Arc Tan2

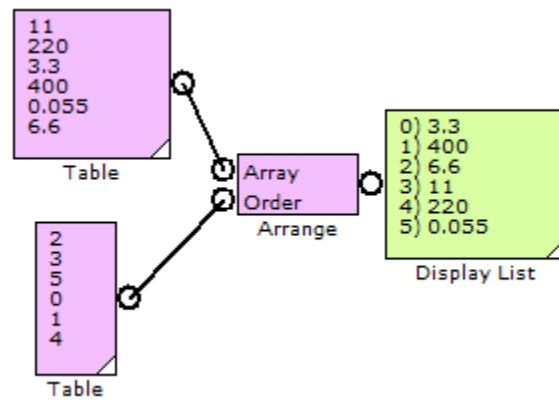
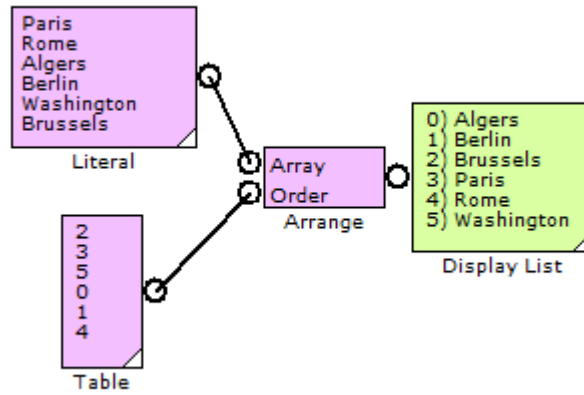
The Arc Tan2 component calculates (in radians or degrees) the inverse trigonometric tangent of the inputs after dividing them. Inputs can be scalar or arrays. Functions - column 5





# Arrange

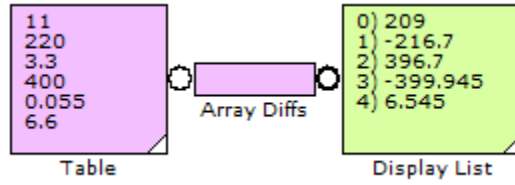
The Arrange component rearranges the input array values based on the array connected to the Order input. Arrays - column 2



# Array Diffs

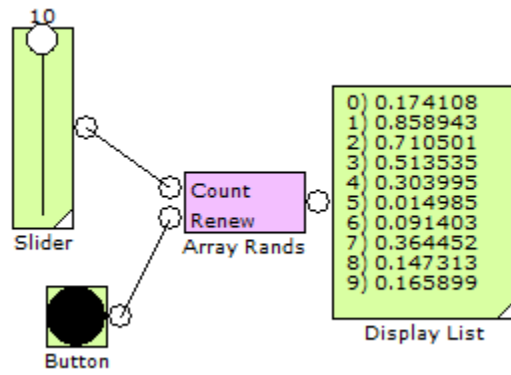
The Array Diffs component creates an array from the differences in the input array. column 3

Arrays -



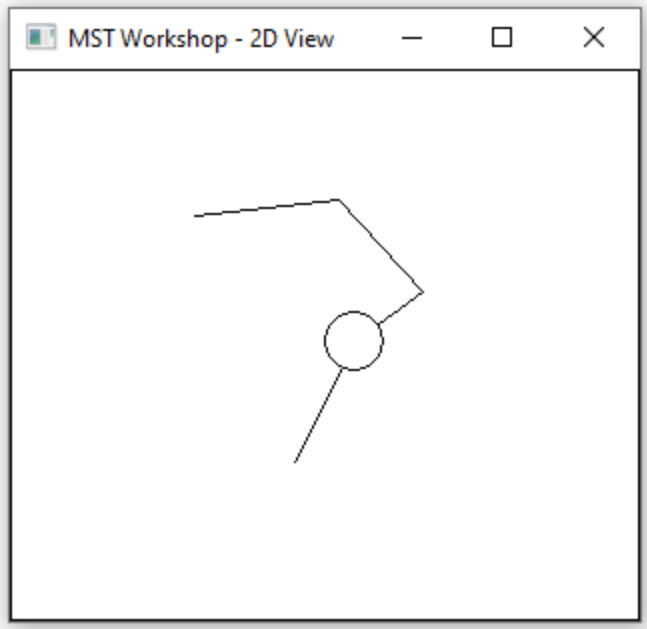
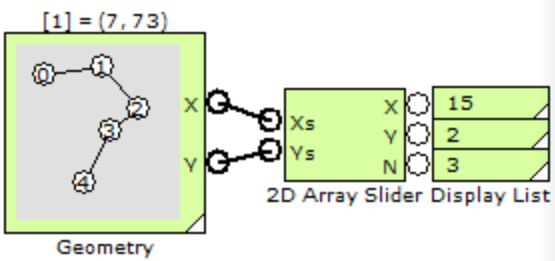
# Array Rands

The Array Rands component creates an array of random numbers between 0.0 and 1.0. If connected the Renew input provides a new array of random numbers. Arrays - column 3



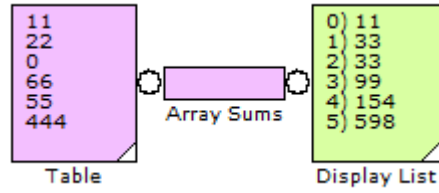
# 2D Array Slider

The 2D Array Slider gives you a control that is an array of XY points on 2D canvas. The output(s) are integer positions of the slider(s). Controls - column 1



# Array Sums

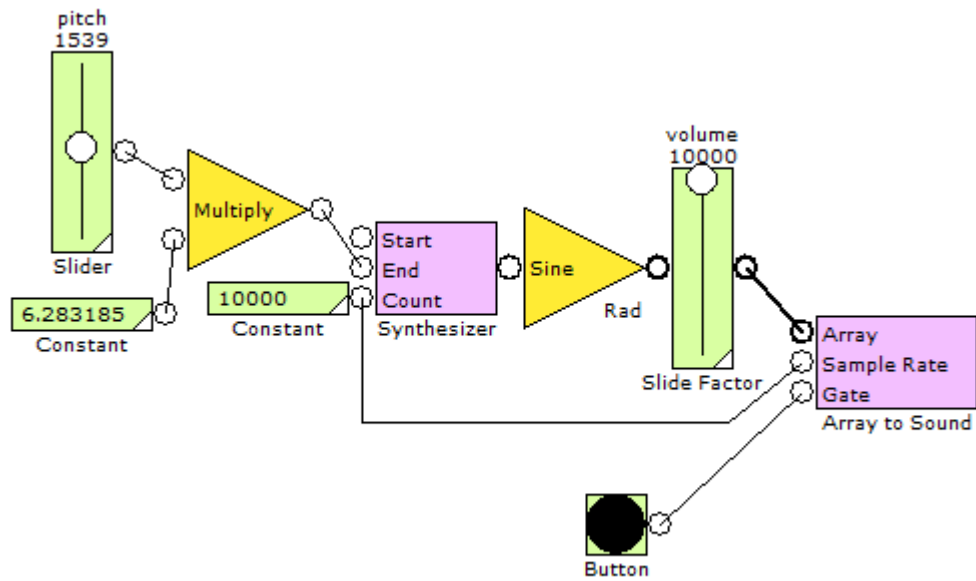
The Array Sums component creates an array from the sums of the input array.    Arrays - column 3



# Array to Sound

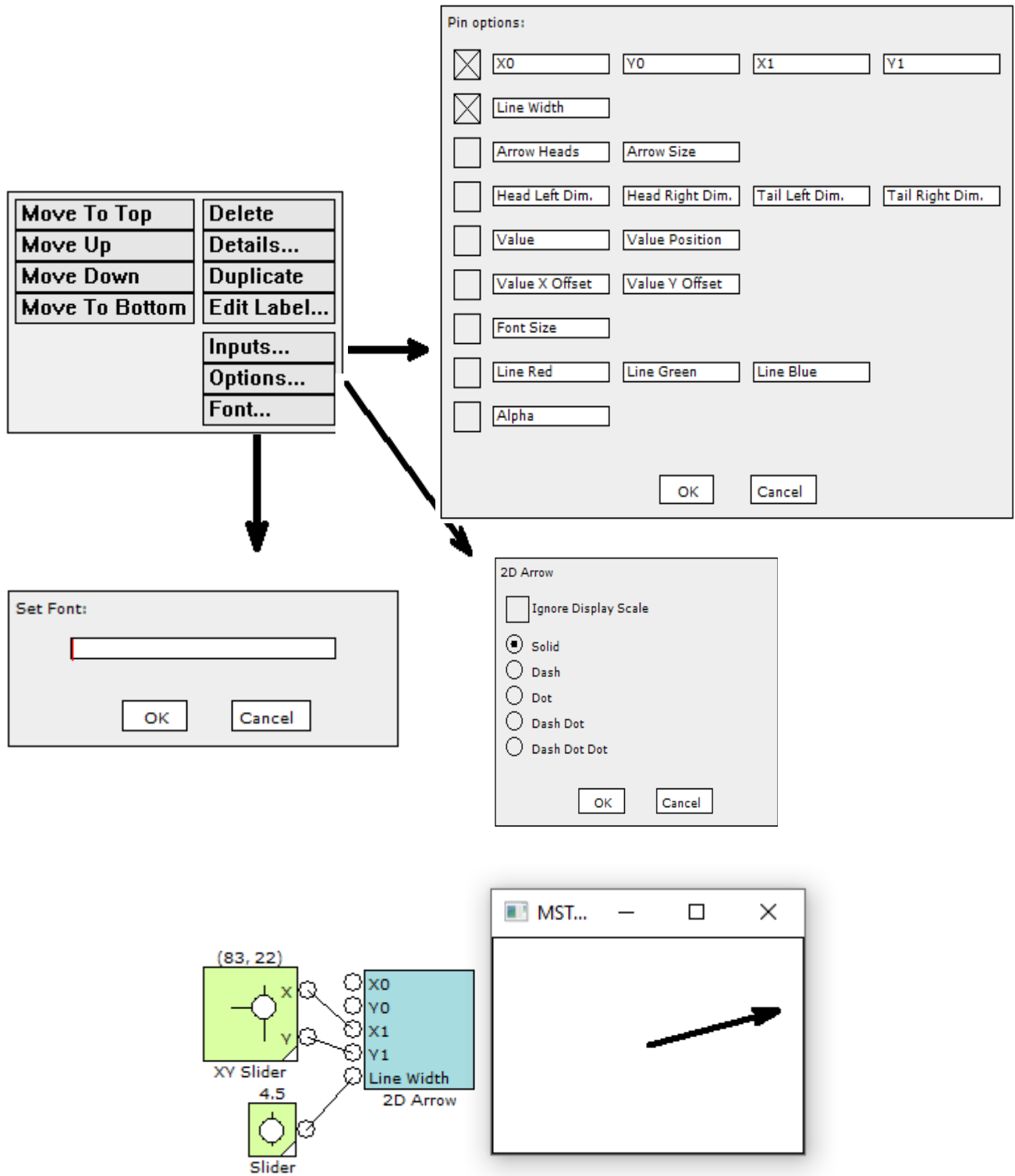
The Array to Sound component sends the array at the input to the system speaker.

Arrays -  
column 7



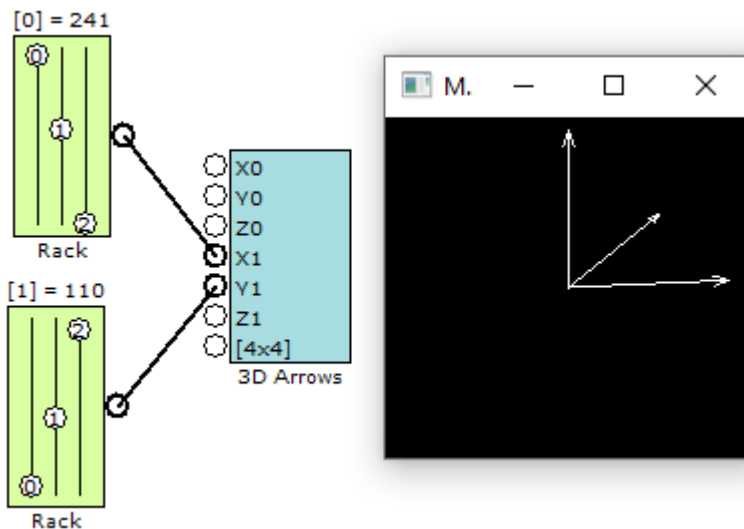
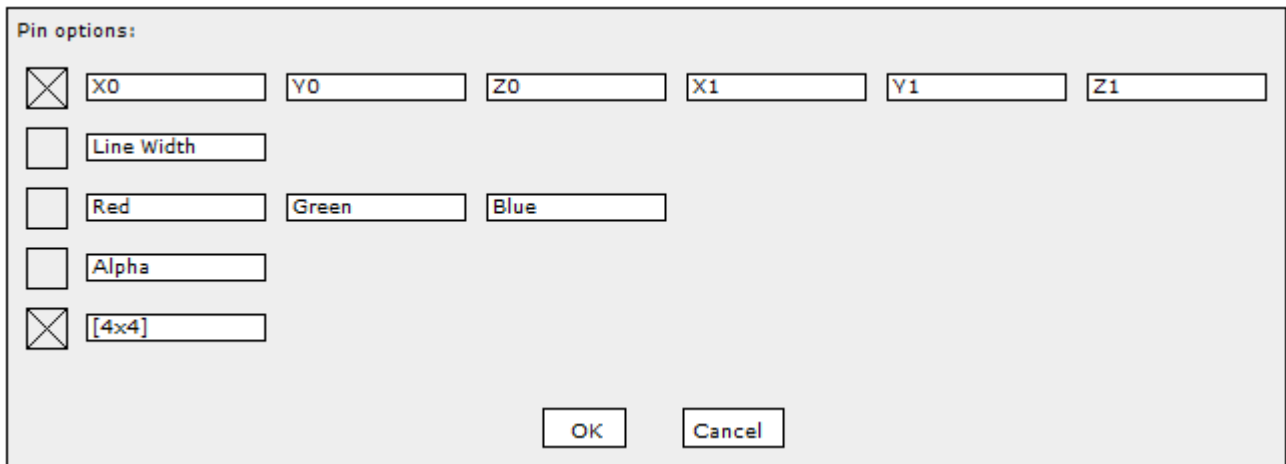
# 2D Arrow

The 2D Arrow component draws one or more arrows on the 2D canvas. 2D - column 1



### 3D Arrow

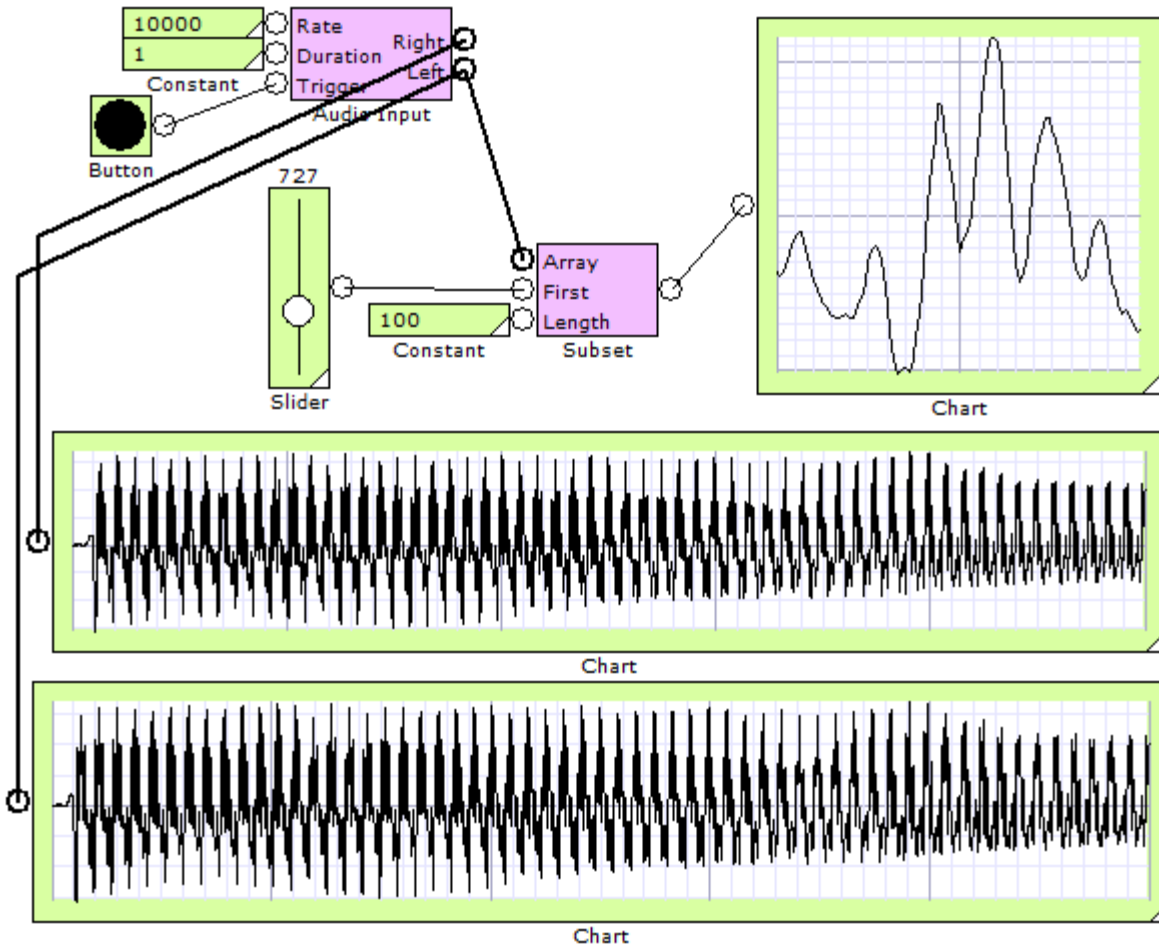
The 3D Arrow component draws multiple arrows in the 3D view window. The X, Y, and Z inputs set the start and end points. The Affine [4x4] array input modifies the size, position and orientation.





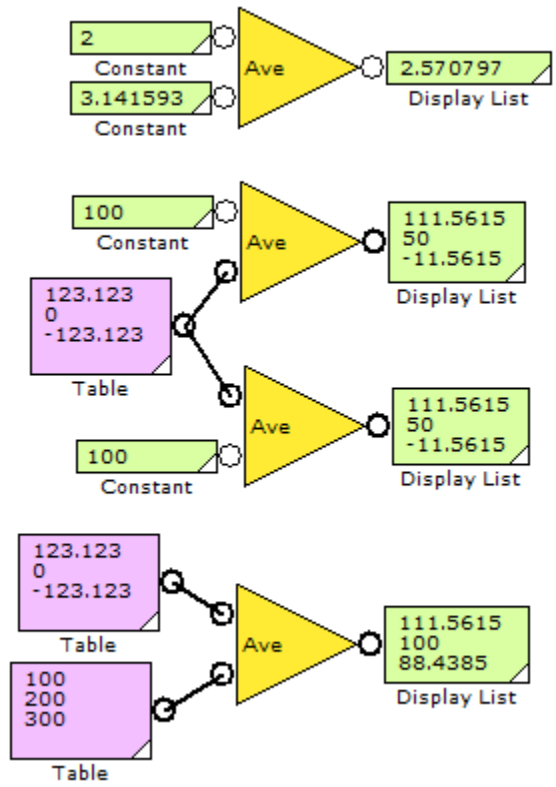
# Audio Input

The Audio Input component reads the default audio input channel. Arrays - column 7



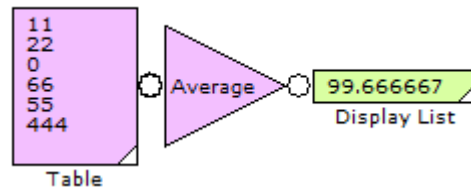
# Ave

The Ave component outputs the average of the inputs. Inputs can be scalars or arrays. Functions - column 2



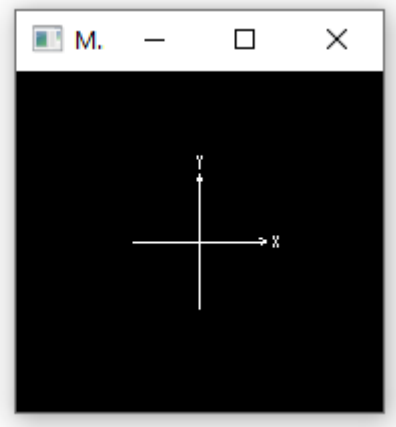
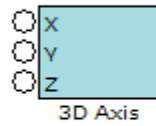
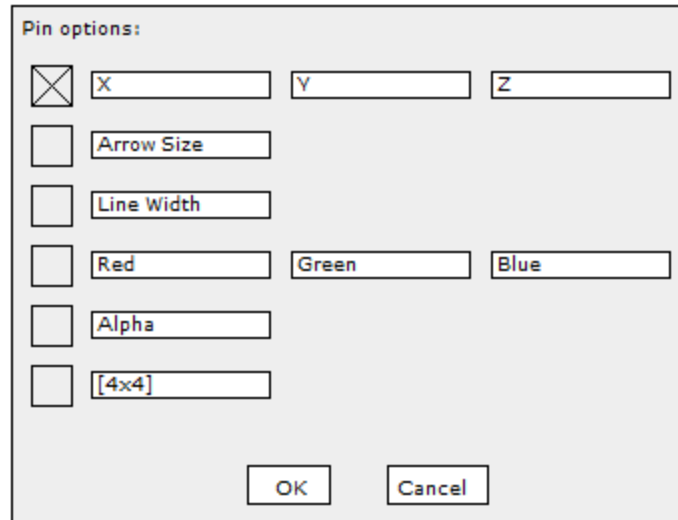
# Average

The Average component scans the input array and averages the values. Arrays - column 3



## 3D Axis

The 3D Axis draws an X, Y, Z axis in the 3D view window. The Affine [4x4] array input modifies the size, position and orientation. 3D - column 4

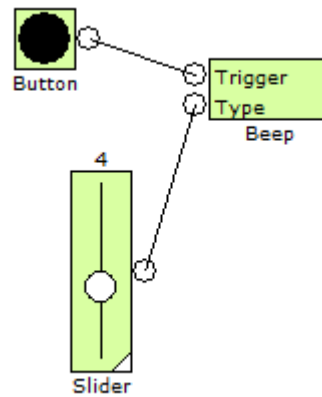


# Beep

The Beep component plays the sound of one of the system beeps. The code ranges from 0 to 8.

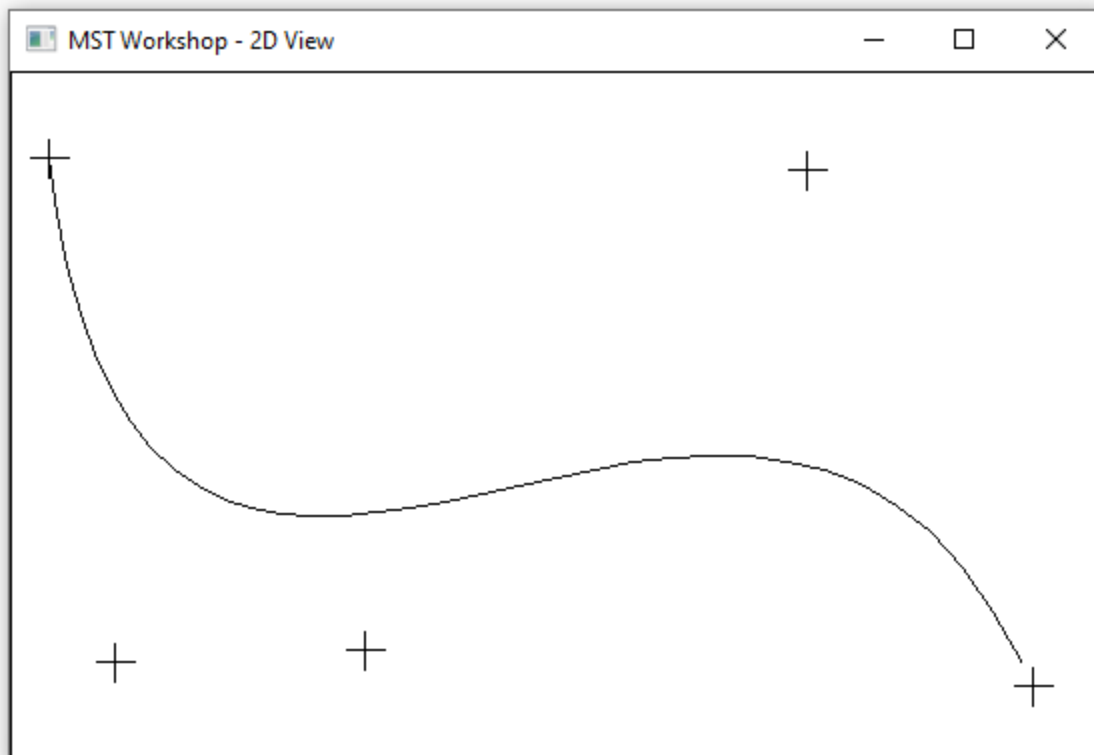
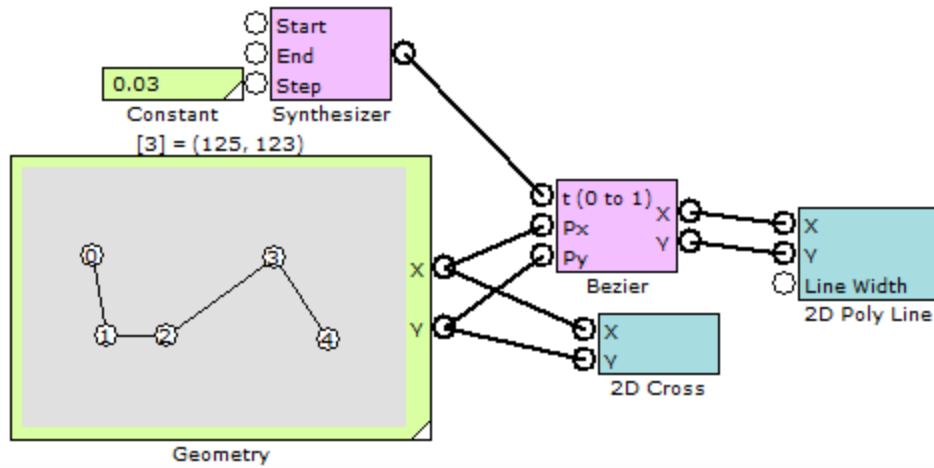
- 0 = MB\_OK
- 1 = MB\_ICONASTERISK
- 2 = MB\_ICONEXCLAMATION
- 3 = MB\_ICONERROR
- 4 = MB\_ICONHAND
- 5 = MB\_ICONINFORMATION
- 6 = MB\_ICONQUESTION
- 7 = MB\_ICONSTOP
- 8 = MB\_ICONWARNING

Controls - column 4



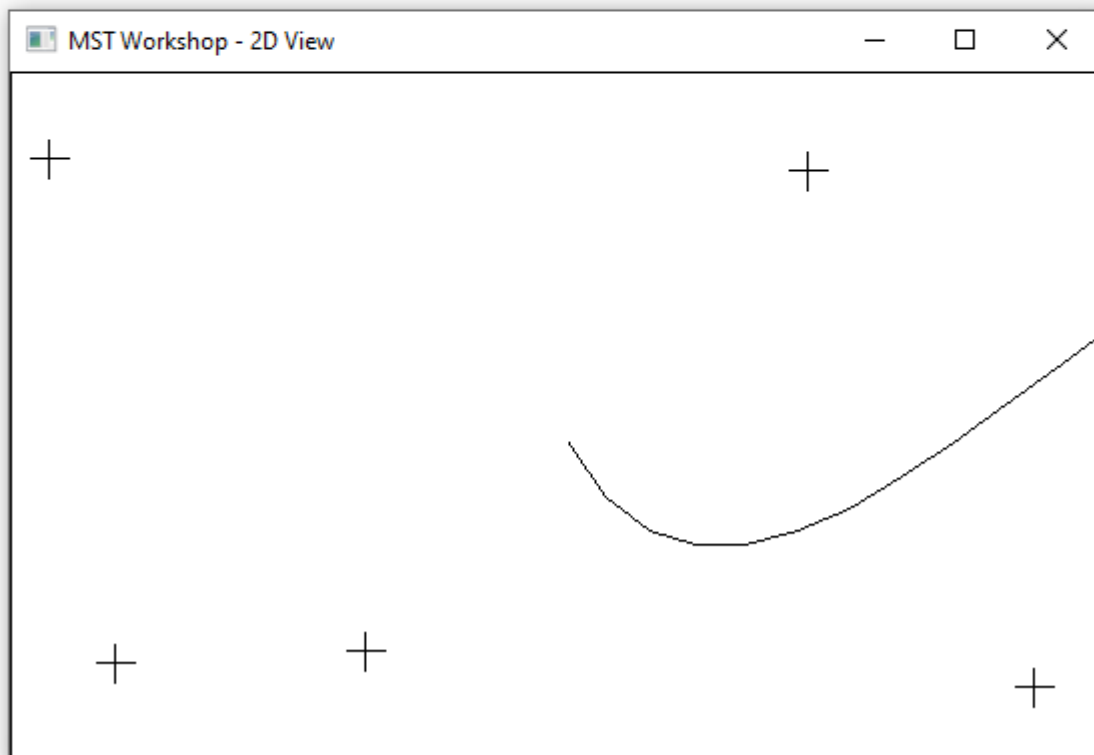
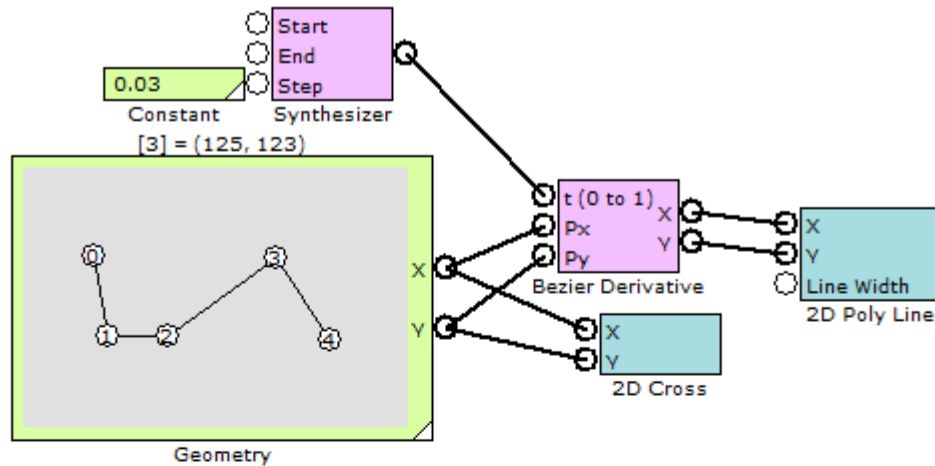
# Bezier

The Bezier component interpolates a Bezier curve based on up to 10 input points. It outputs the X and Y value(s) at the positions given by t, ranging from 0 to 1. Arrays - column 4



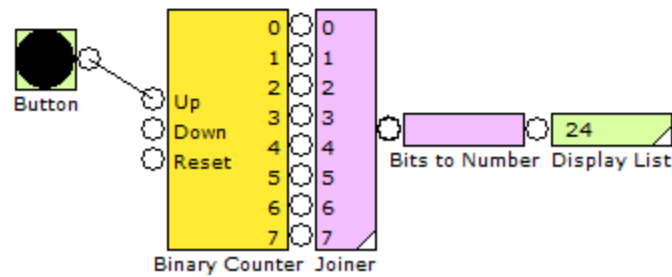
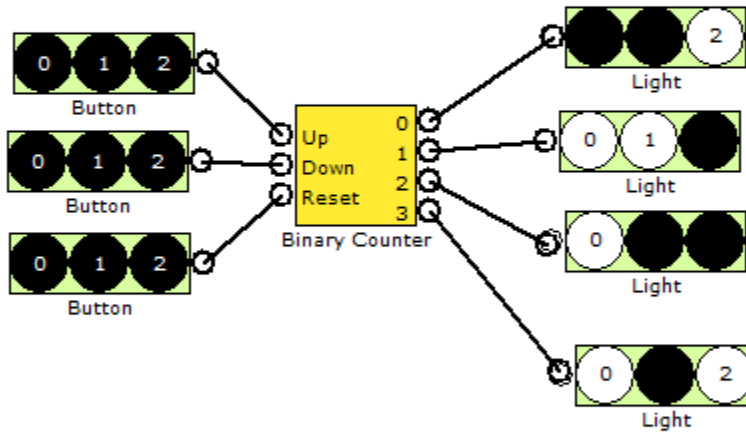
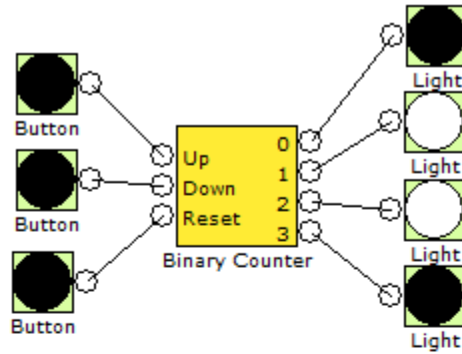
# Bezier Derivative

The Bezier Derivative component interpolates a Bezier's Derivative curve based on up to 10 input points. It outputs the X and Y value(s) at the positions given by t, ranging from 0 to 1. Arrays - column 4



# Binary Counter

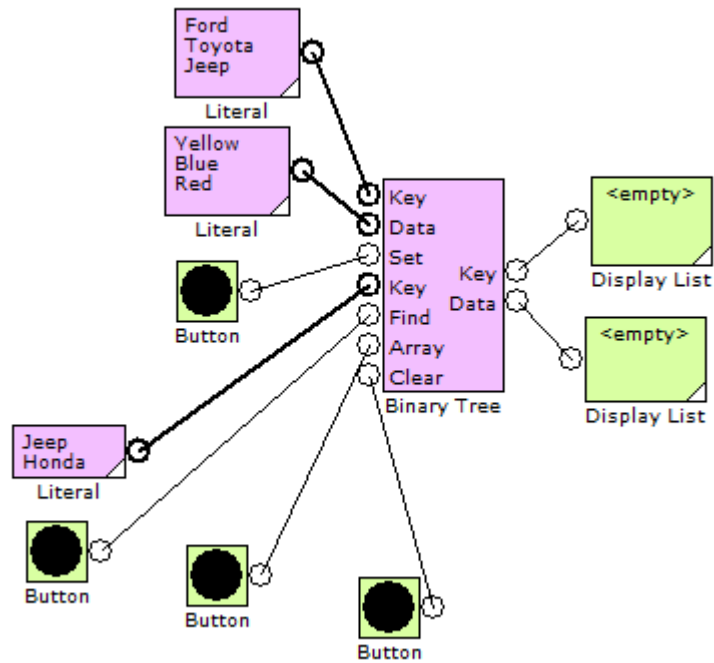
The Binary Counter component counts up or down and outputs binary bits. You can set the number of outputs desired. Functions - column 8





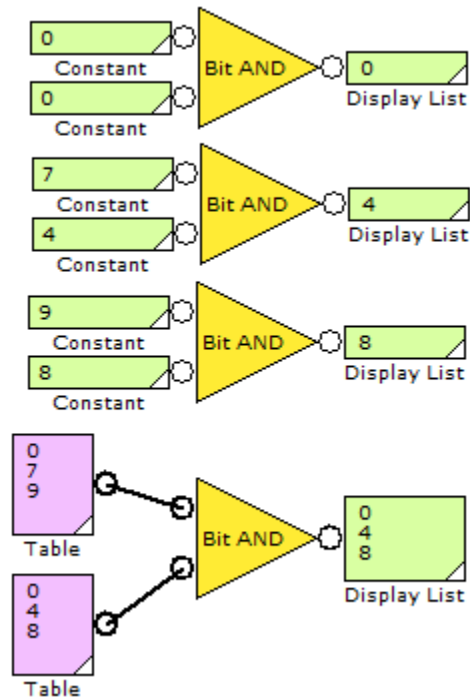
# Binary Tree

The Binary Tree can save key/data pairs in a binary tree on Set, then recall the data given the key on Find. Arrays - column 5



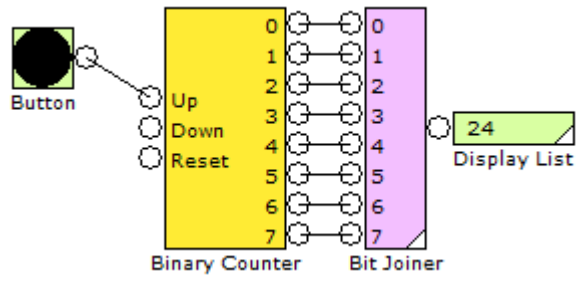
# Bit AND

The Bit AND component compares the inputs and returns the result of a bitwise AND process on the inputs Functions - column 3



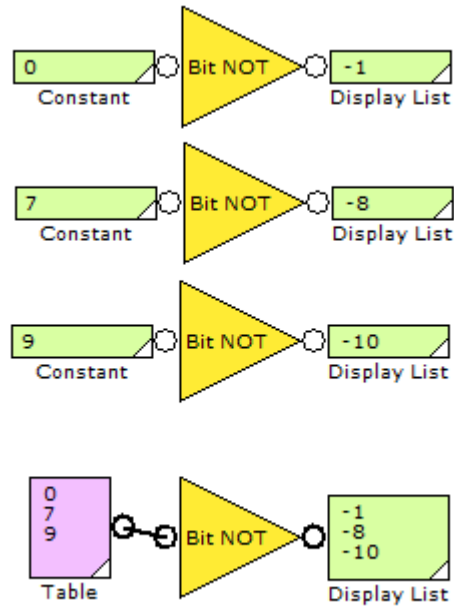
# Bit Joiner

The Bit Joiner component joins all the bits for all its inputs into a single output. You can set the number of inputs. Arrays - column 4



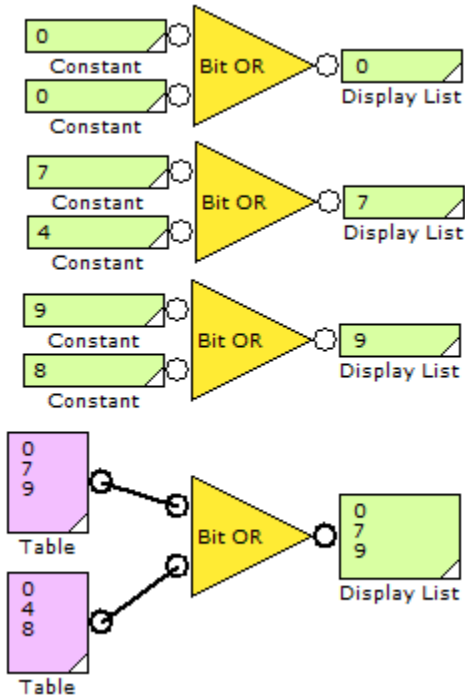
# Bit NOT

The Bit NOT component compares the inputs and returns the result of a bitwise inversion process on the input. Functions - column 3



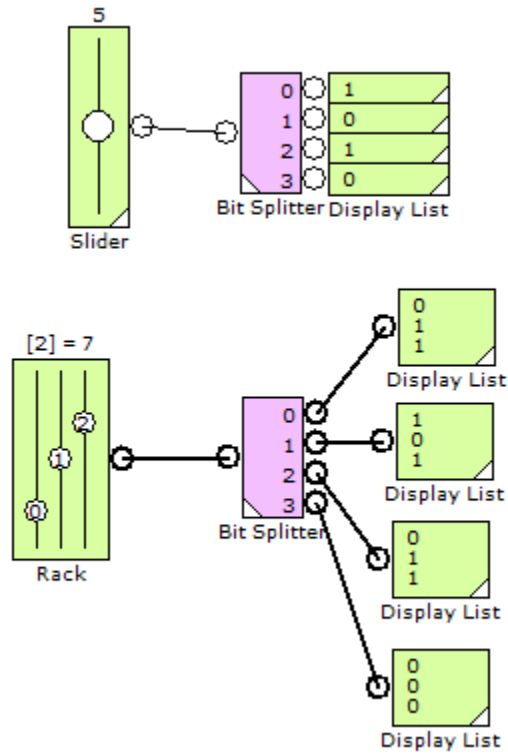
# Bit OR

The Bit OR component compares the inputs and returns the result of a bitwise OR process on the inputs. Functions - column 3



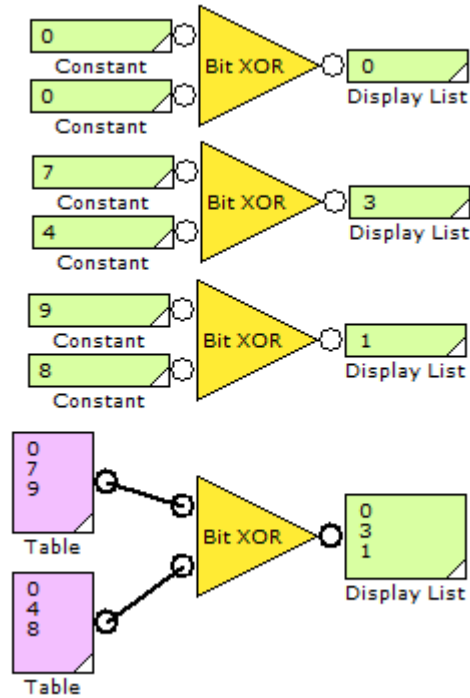
# Bit Splitter

The Bit Splitter component breaks the input into separate bits. You can set the number of outputs desired. Arrays - column 4



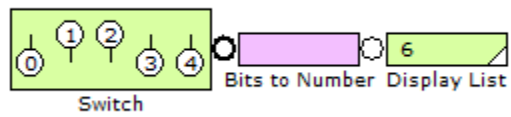
# Bit XOR

The Bit XOR component compares the inputs and returns the result of a bitwise Exclusive OR process on the inputs. Functions - column 3



## Bits to Number

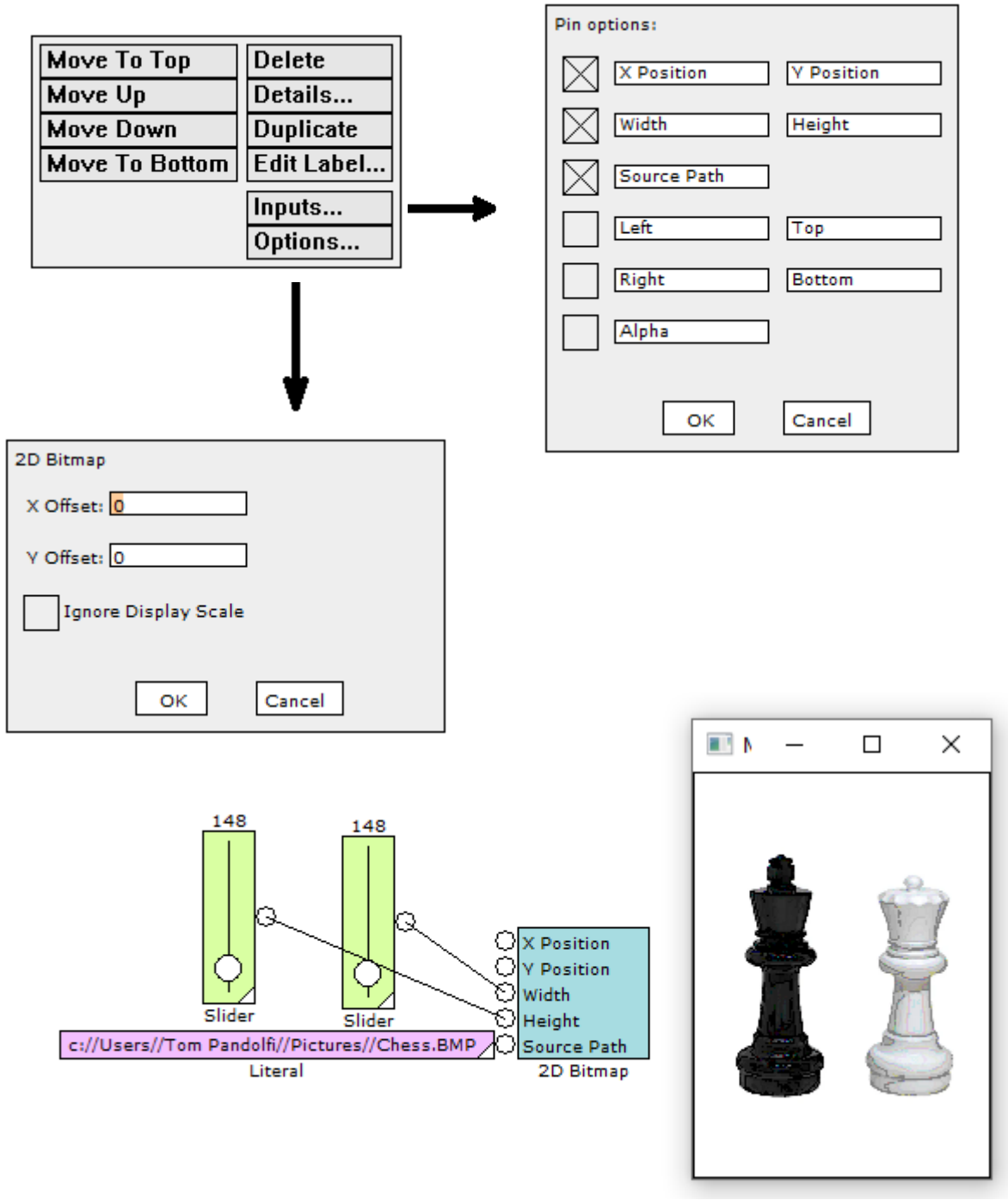
The Bits to Number component joins an array of bit values into a single output. Arrays - column 4





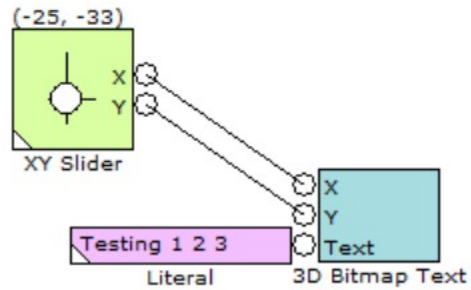
## 2D Bitmap

The 2D Bitmap reads a BMP image file given its path. It then paints that bitmap in the 2D View at the specified location and size. You can paint part of a bitmap by specifying the top, left, bottom, and right pixel values of the bitmap. 2D - column 2



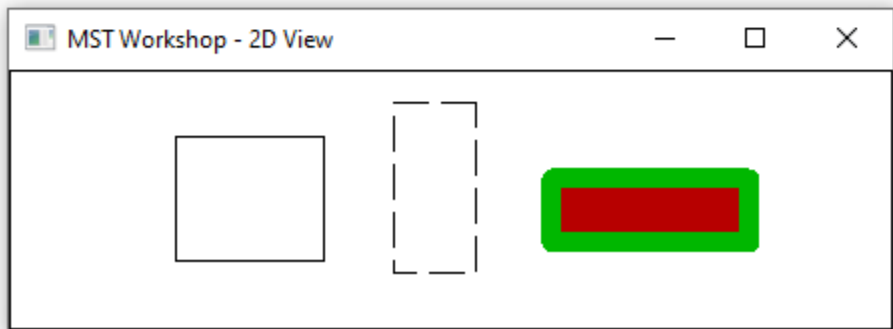
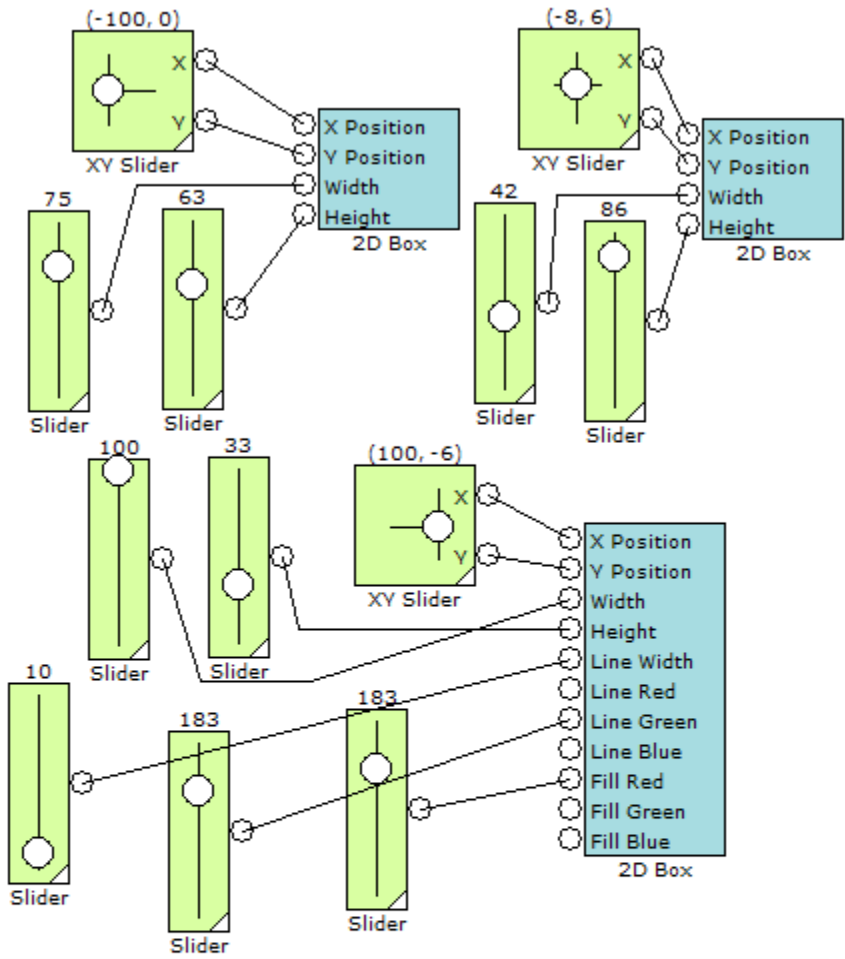
## 3D Bitmap Text

The 3D Bitmap Text draws a string of text in the 3D view window. The X, Y, and Z inputs set the position of the text. 3D – column 1



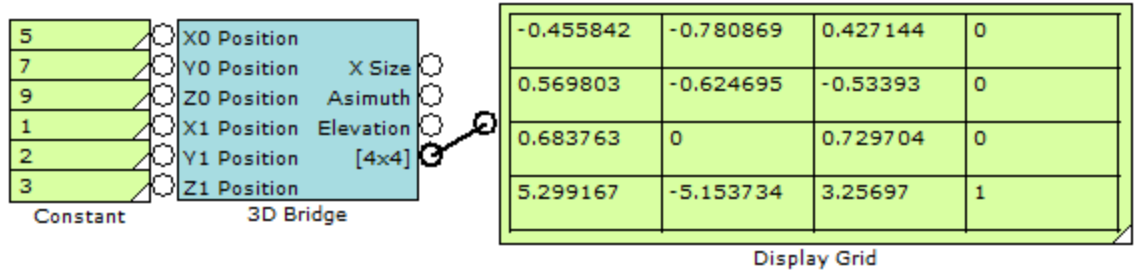
# 2D Box

The 2D Box component draws a rectangle on the 2D canvas. 2D - column 1



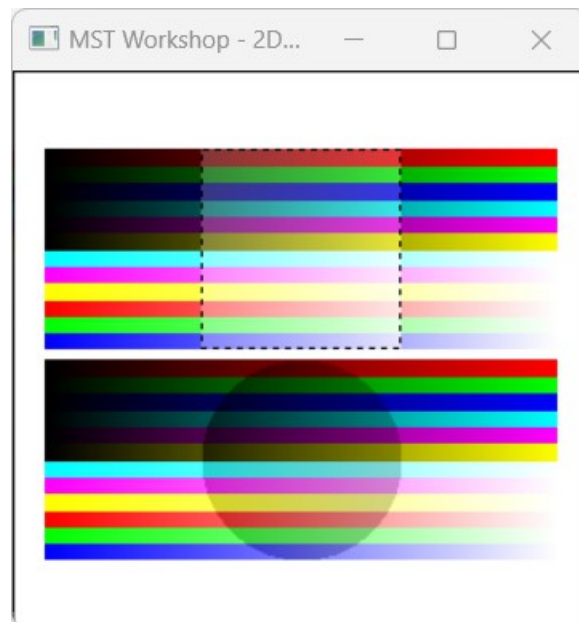
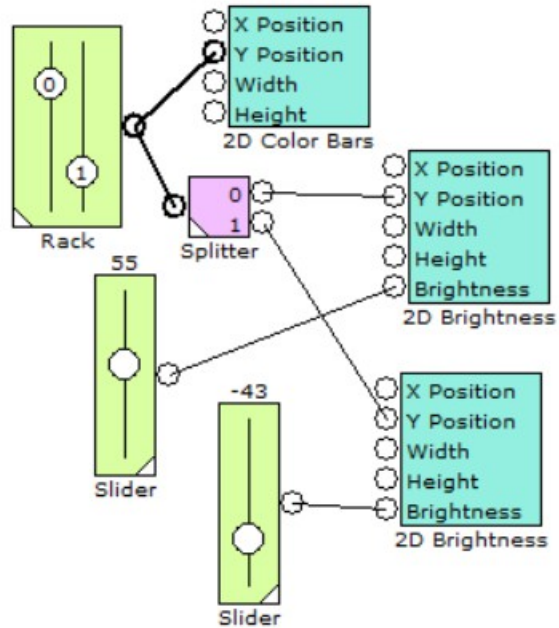
# 3D Bridge

The 3D Bridge takes a pair of X, Y, Z coordinates, then returns information about how to bridge the two points including a matrix that will connect to a 3D Cube to bridge the two points. 3D - column 3



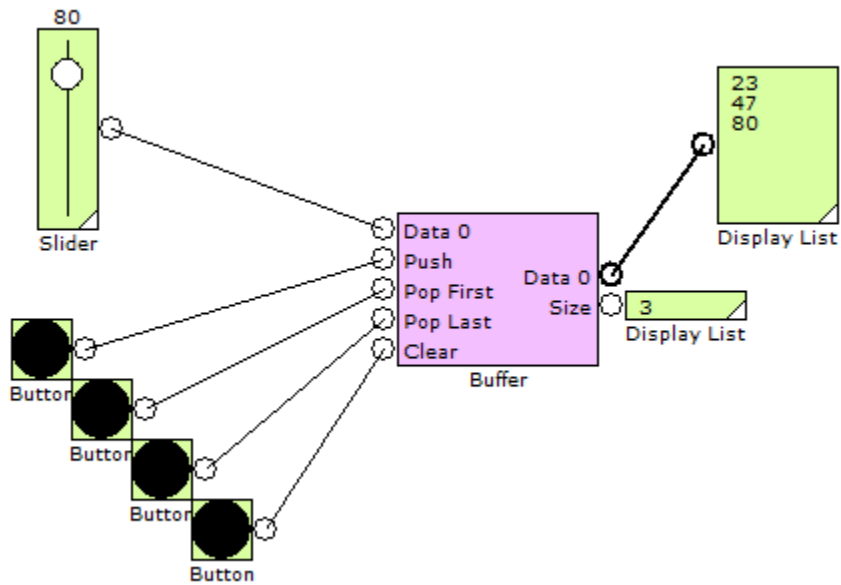
## 2D Brightness

The 2D Brightness lightens or darkens the 2D drawing it covers. 2D – column 1



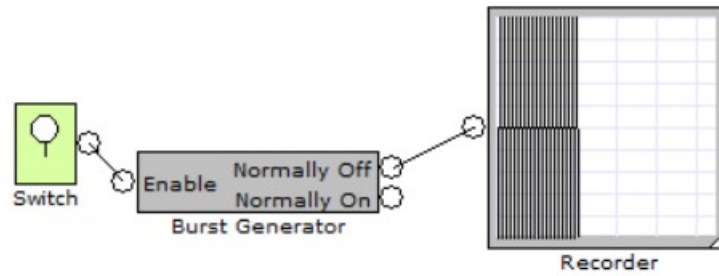
# Buffer

The Buffer lets you create a list that you can insert values into or delete values from. Arrays - column 1



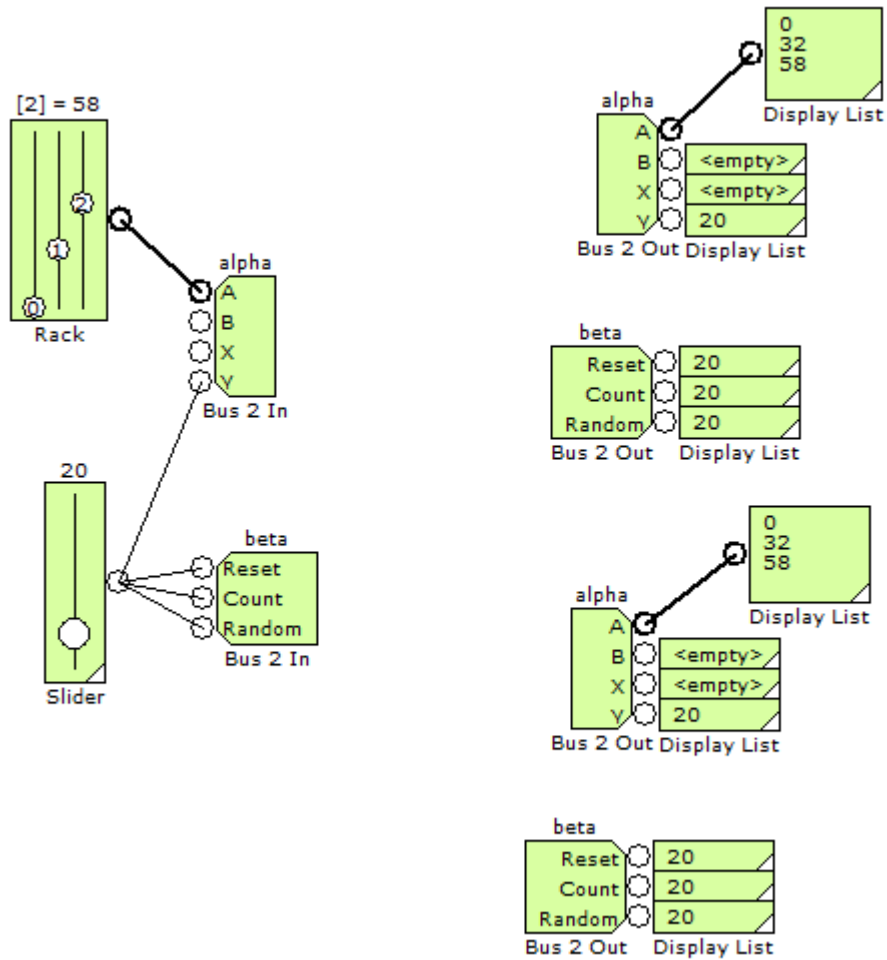
# Burst Generator

The Burst Generator component outputs rapid pulses alternating between output A and B at half the current clock rate. Functions – column 8



# Bus 2 In

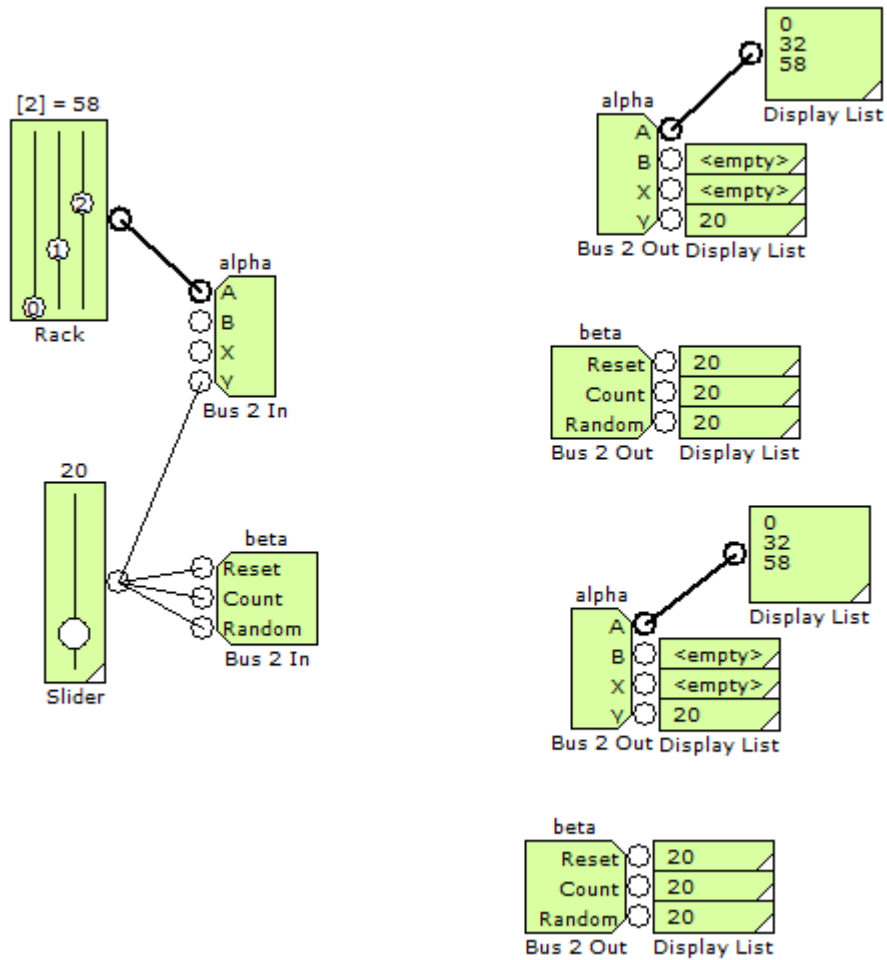
The Bus 2 In component passes a value to all the Bus 2 Out components with the same name.  
Controls - column 5





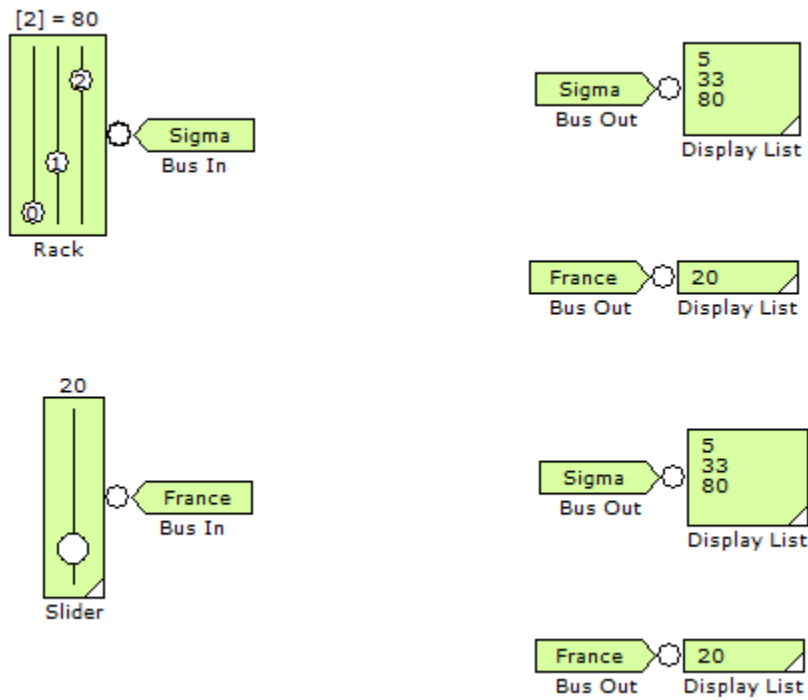
# Bus 2 Out

The Bus 2 Out component outputs the value that is connected to the Bus 2 In of the same name.  
 Controls - column 5



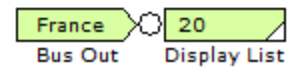
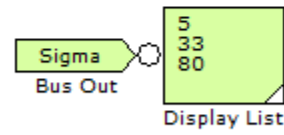
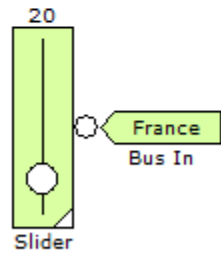
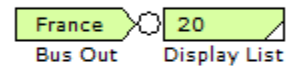
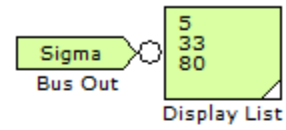
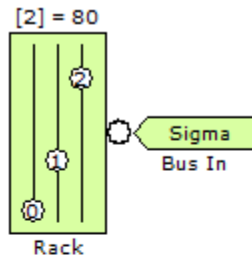
# Bus In

The Bus In component passes a value to all the Bus Out components with the same name.  
Controls - column 5



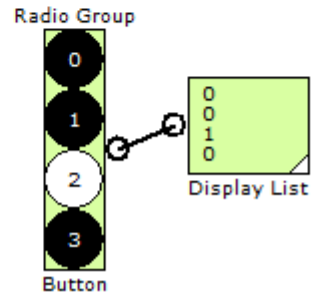
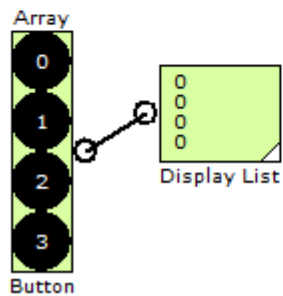
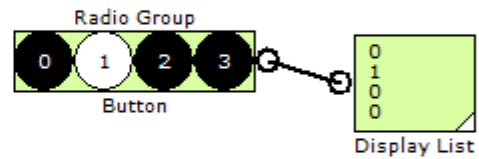
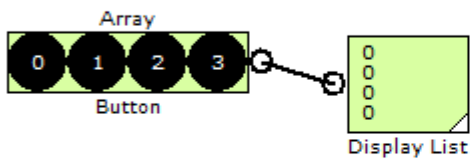
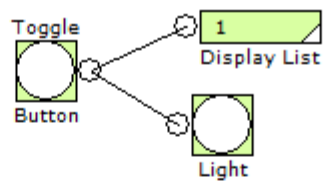
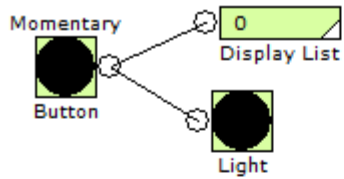
# Bus Out

The Bus Out component outputs the value that is connected to the Bus In of the same name.  
Controls - column 5



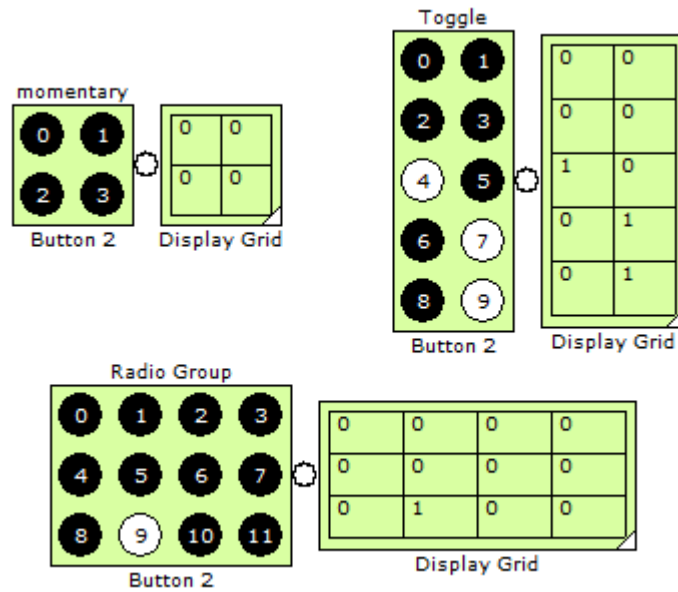
# Button

The Button is a one dimensional array of buttons. They can be momentary, toggle, or radio buttons.  
Controls - column 4



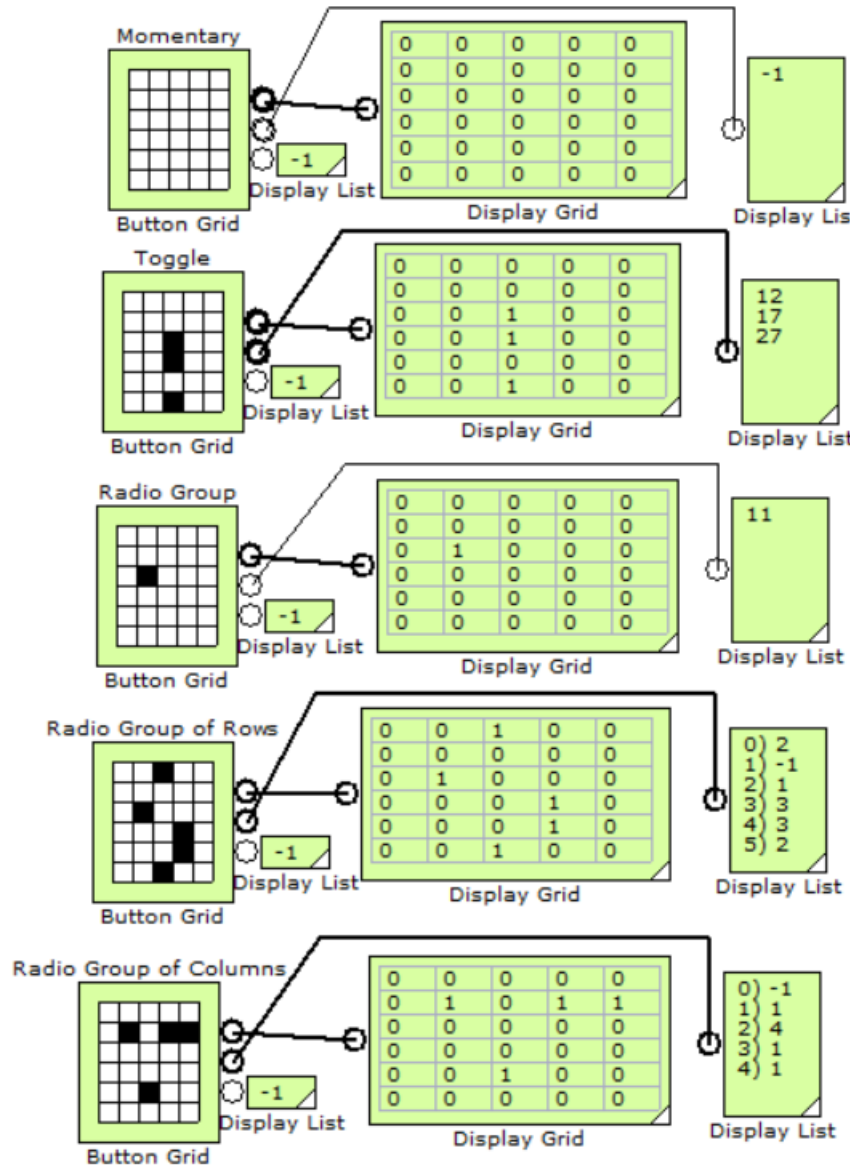
## Button 2

The Button 2 is a 2 dimensional array of buttons. They can be momentary, toggle, or radio buttons.  
Controls - column 4



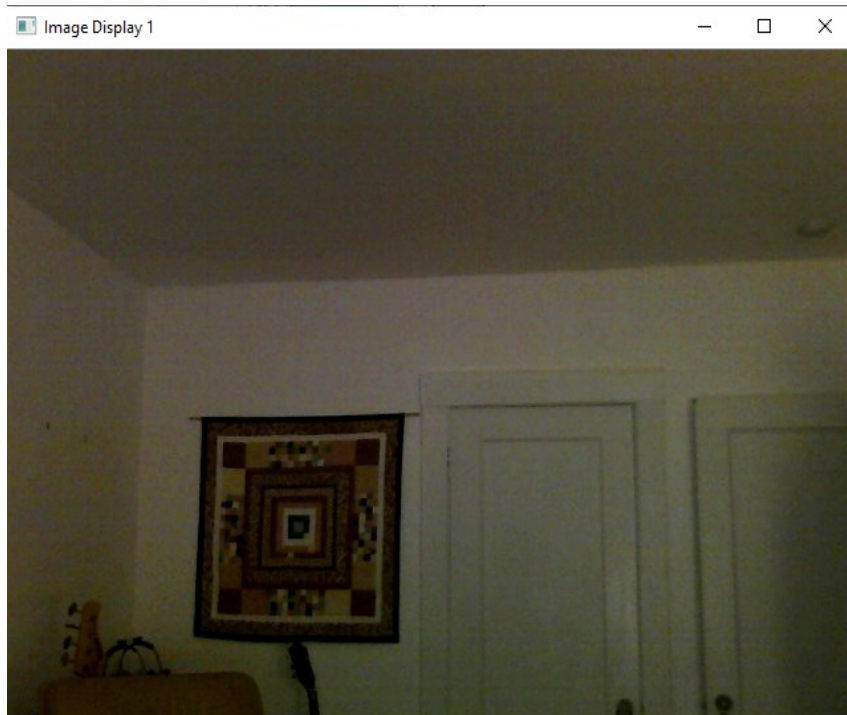
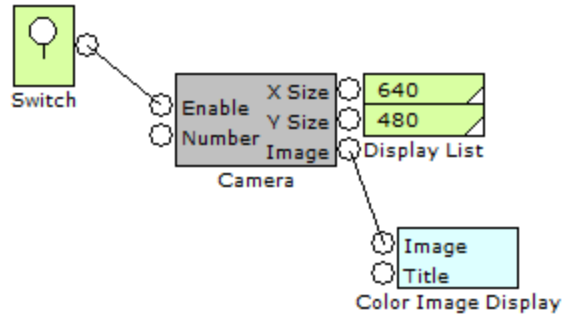
# Button Grid

The Button Grid is a 2-dimensional array of buttons. They can be momentary, toggle, or radio buttons. The top output is a matrix corresponding to the button down. The middle output is a list of the active buttons. The bottom output is the value of the most recent button down or change. Controls – column 4



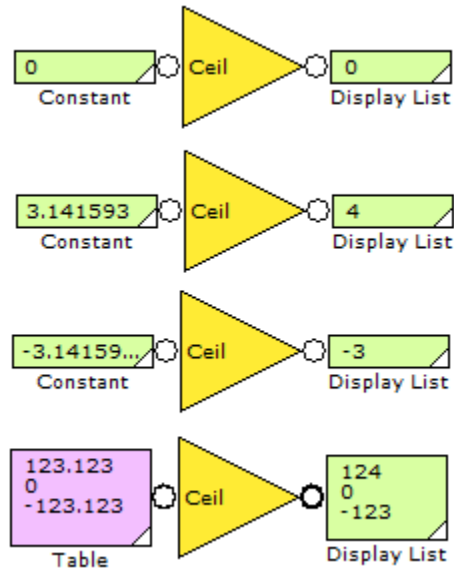
# Camera

The Camera component grabs a frame from a camera connected to the computer. Images - column 1



# Ceil

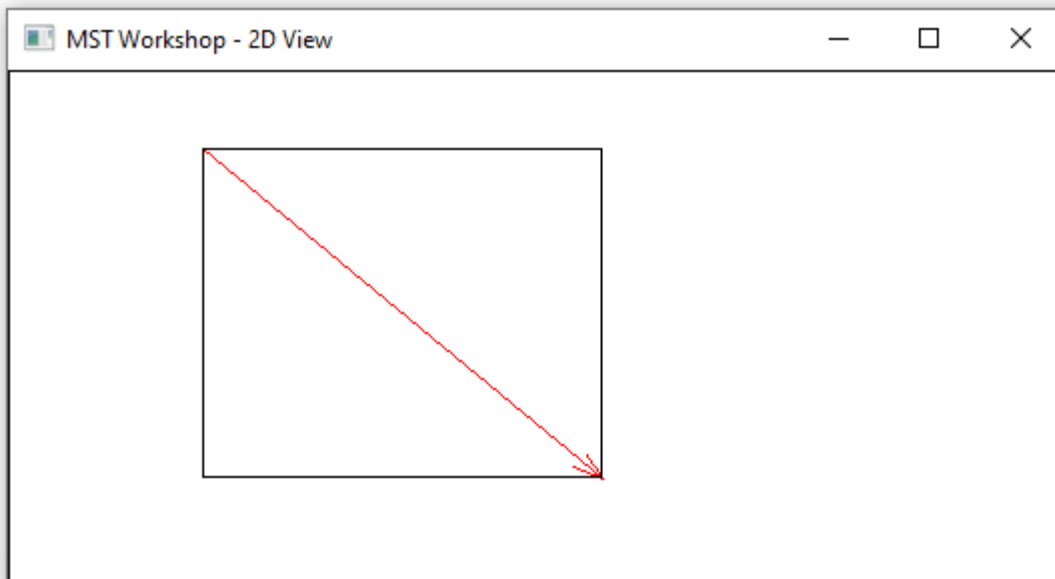
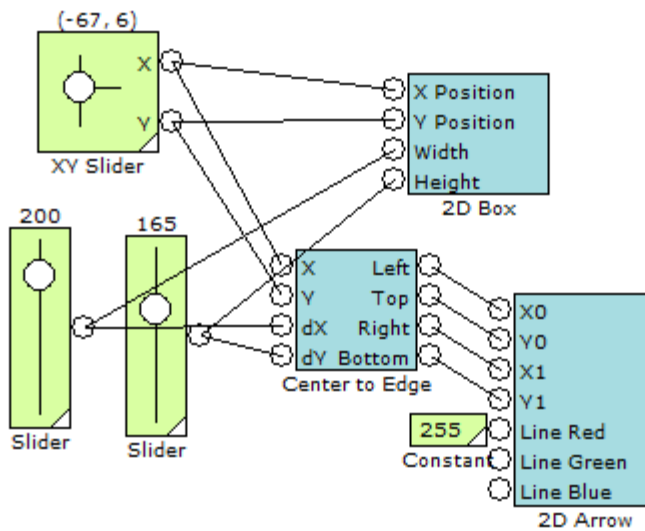
The Ceiling component outputs an integer greater than or equal to the input. Input can be scalar or an array. Functions - column 2





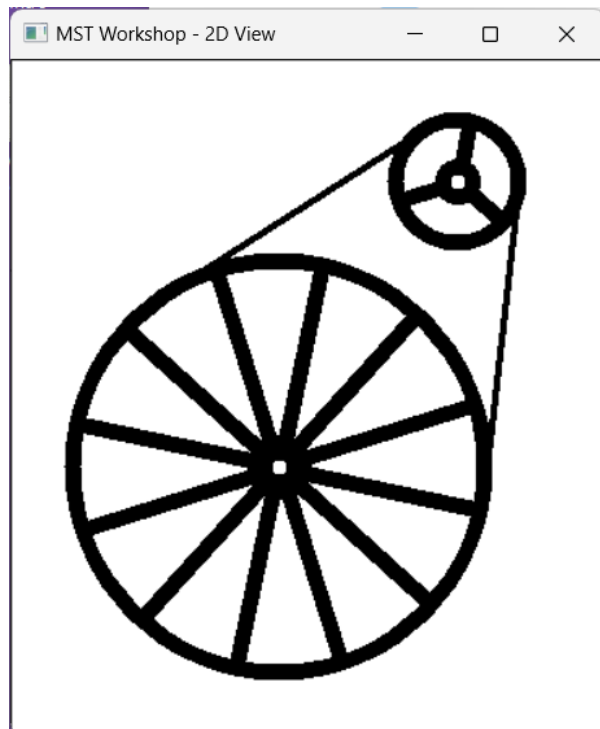
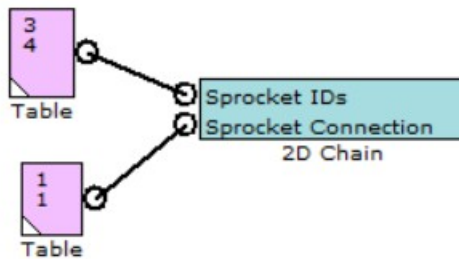
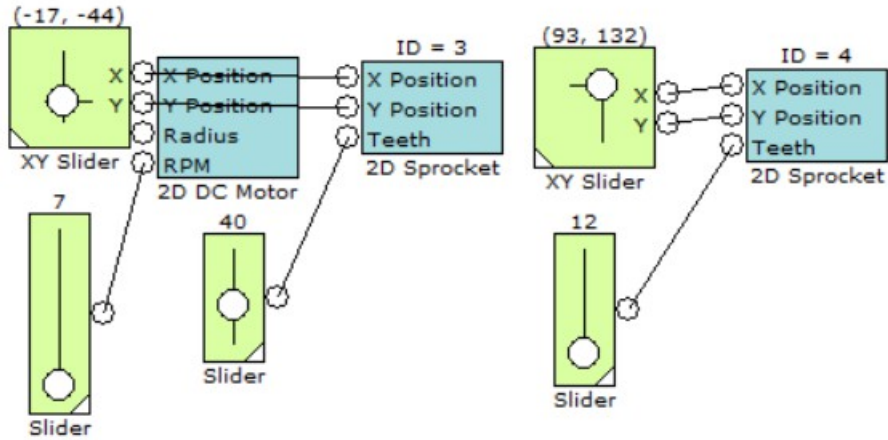
# Center to Edge

The Center to Edge component takes X Y coordinates and deltas and outputs values that can be used to put a vector from corner to corner to corner. 2D - column 8



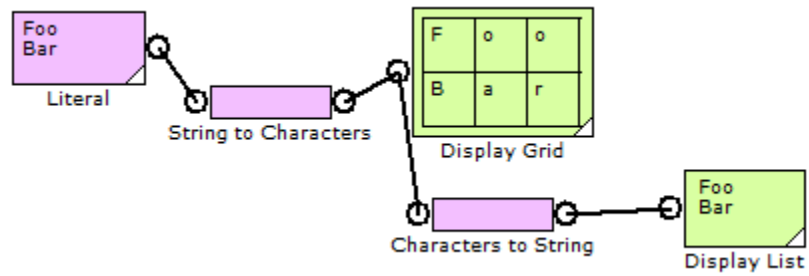
## 2D Chain

The 2D Chain component creates a Chain that can connect multiple sprockets. Indicate the ID of each sprocket and the direction of rotation. 2D - column 3



## Characters to String

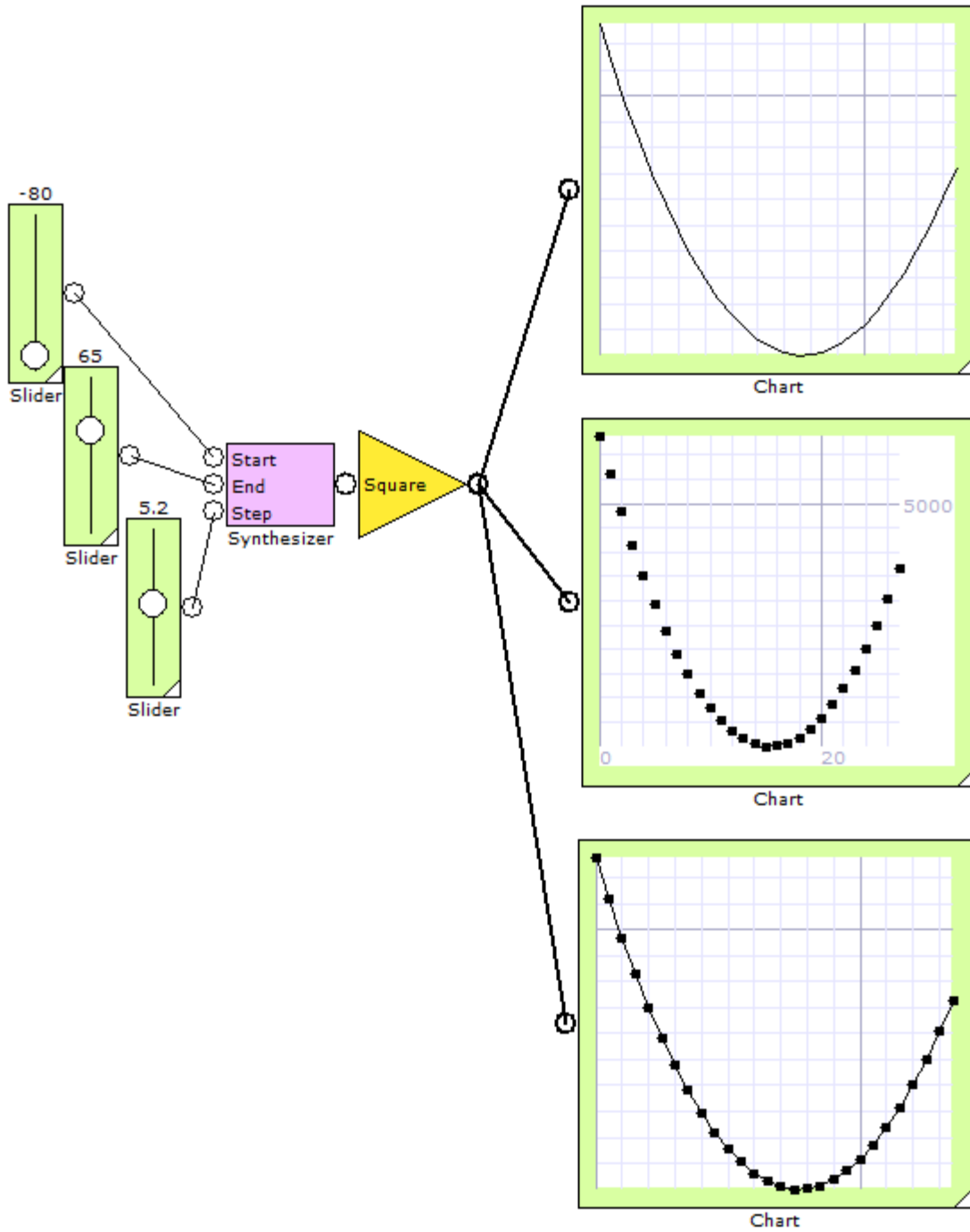
The Characters to String component takes a string of characters and creates an array of characters.  
Arrays - column 7



# Chart

The Chart shows a line or dot chart of the array at its input.

Controls - column 4



# 2D Chart

The 2D Chart shows the array at its input in the 2D display window. 2D - column 2

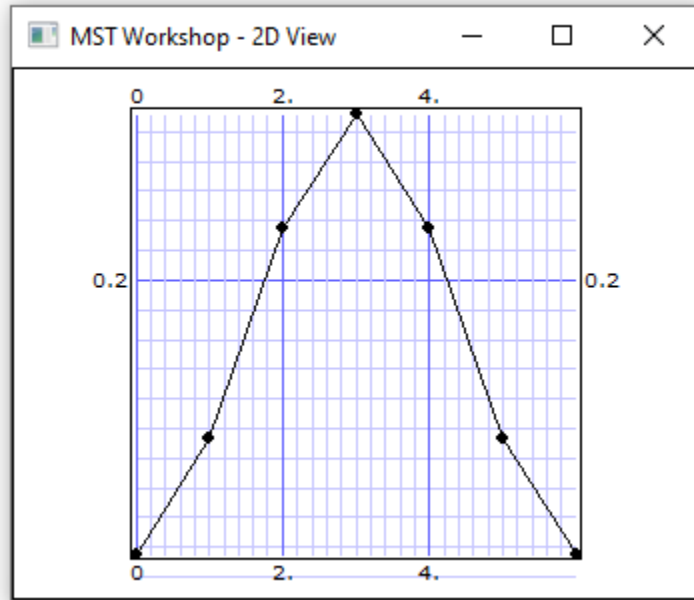
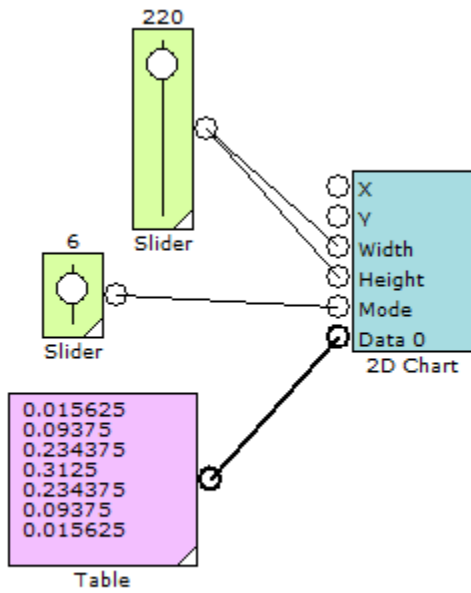
Move To Top	Delete
Move Up	Details...
Move Down	Duplicate
Move To Bottom	Edit Label...
	Inputs...
	Options...



Pin options:

<input checked="" type="checkbox"/>	X	Y	
<input checked="" type="checkbox"/>	Width	Height	
<input checked="" type="checkbox"/>	Mode		
<input type="checkbox"/>	Delta X		
<input type="checkbox"/>	Line Width	Dot Size	
<input type="checkbox"/>	Fill Red	Fill Green	Fill Blue
<input type="checkbox"/>	Alpha		
<input type="checkbox"/>	X Axis Range		
<input type="checkbox"/>	Minimum Y	Maximum Y	
<input checked="" type="checkbox"/>	Data 0		
<input type="checkbox"/>	Data 1		
<input type="checkbox"/>	Data 2		

OK Cancel



## 2D Checkbox

The 2D Checkbox is used to enable an event by clicking on it.

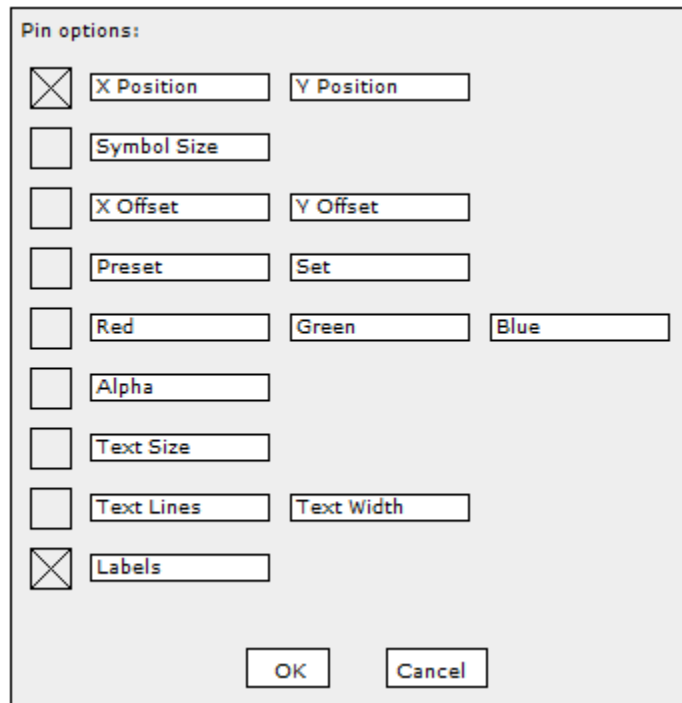
The X, Y and Size inputs control the position and font size of the text.

The Offsets control the position of the text relative to the boxes.

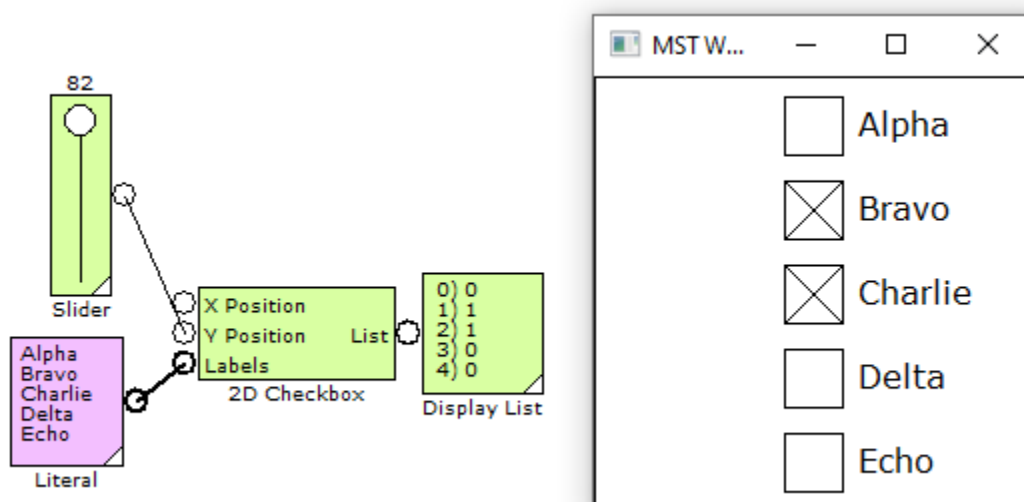
The Preset inputs allow external setting of the selections. Use an array of ones and zeros to set the individual boxes.

The Color input sets the color of the boxes and text.

The Alpha input fades the boxes and text.

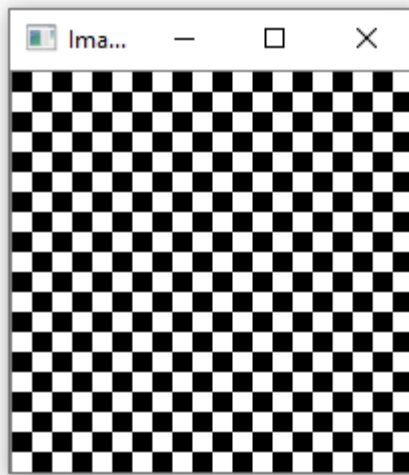
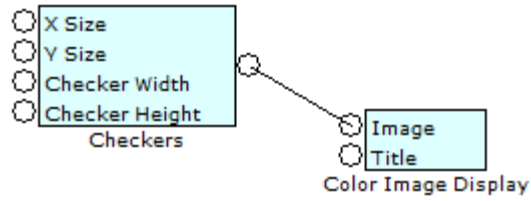


The List output provides an array with a one at the selected positions. Controls - column 2



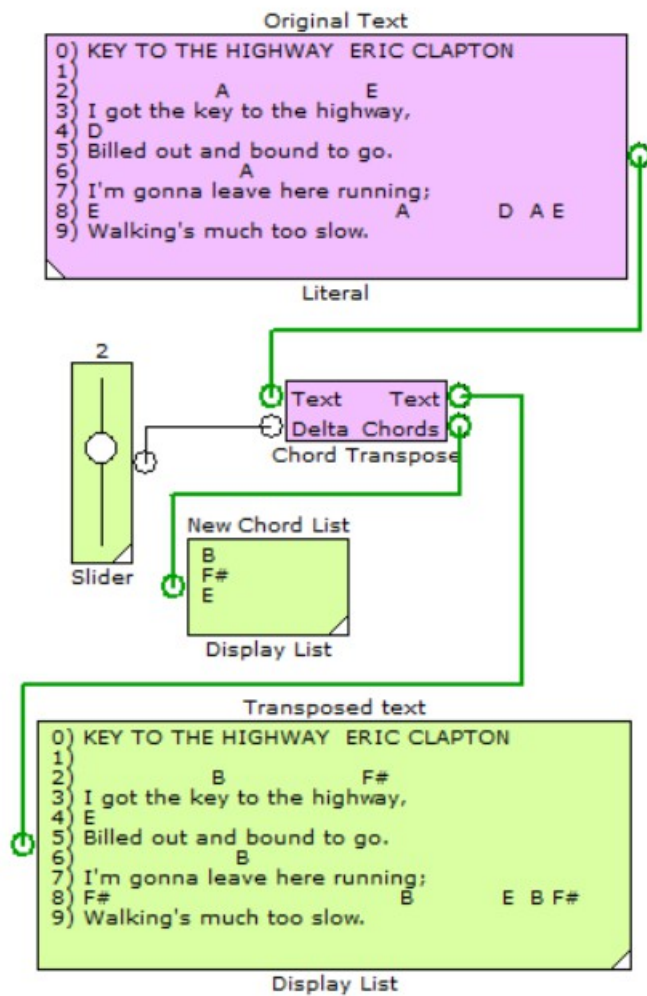
# Checkers

The Checkers component creates an image with a checkered test pattern. Images - column 1



# Chord Transpose

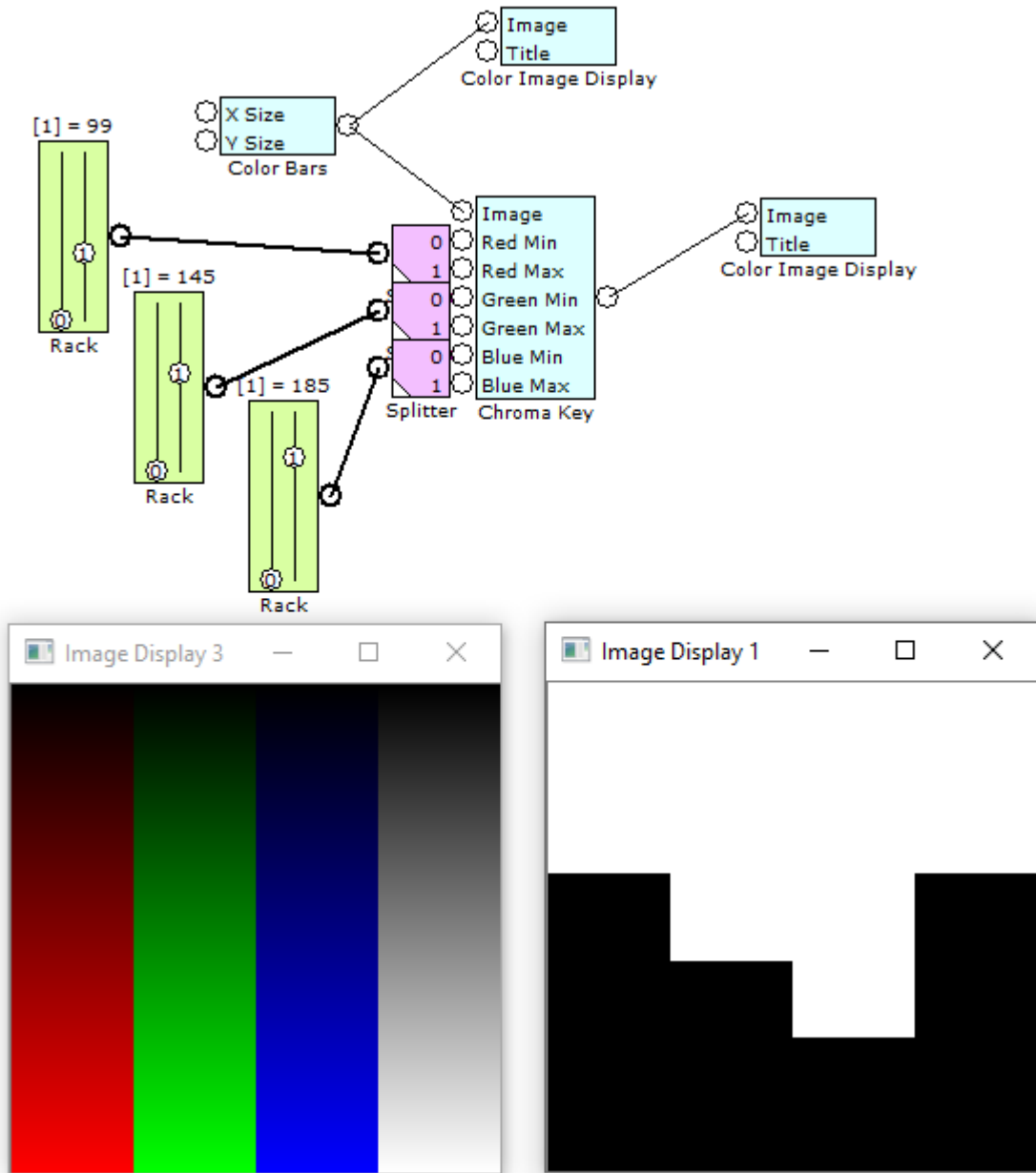
The Chord Transpose component takes an array of text lines and transposes the chord progression to a new key by the delta indicated. Only lines that contain A through G followed by a blank or a sharp (#) or flat (b) then a space are considered. Text from Rich Text Format files also works correctly.

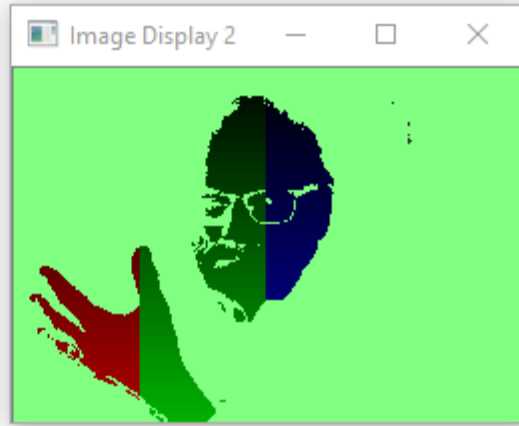
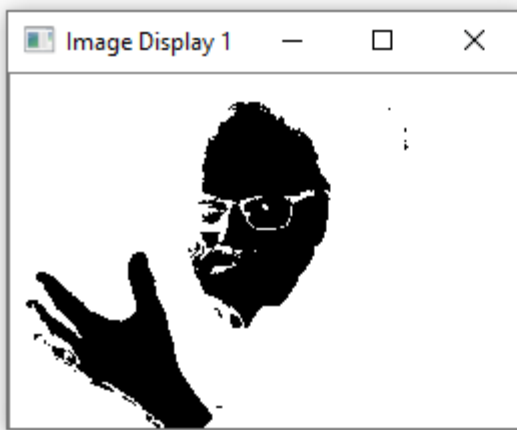
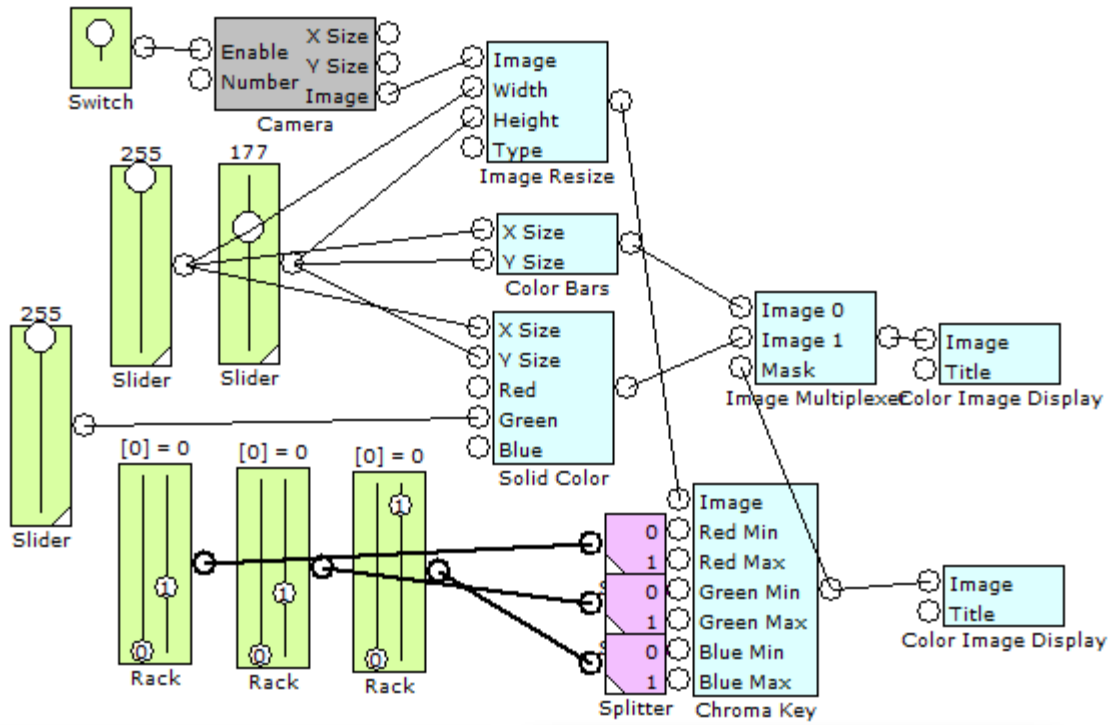




# Chroma Key

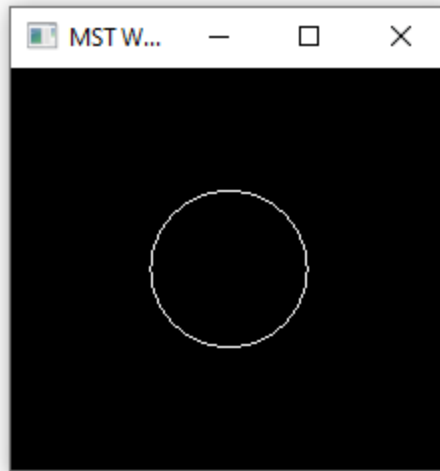
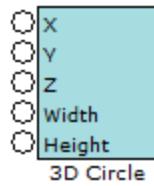
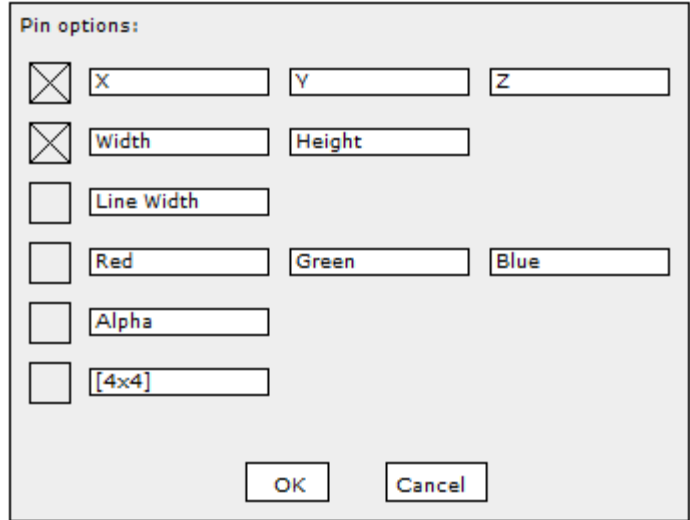
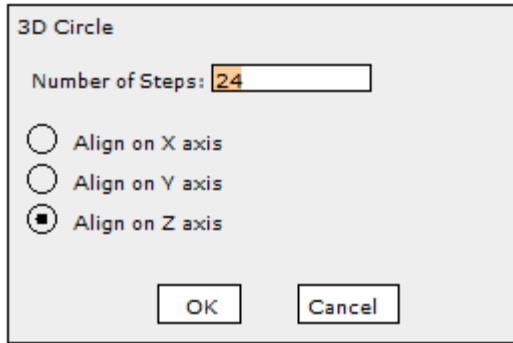
The Chroma Key component outputs white where it finds pixels that are within the limits of the three color inputs. Images - column 2





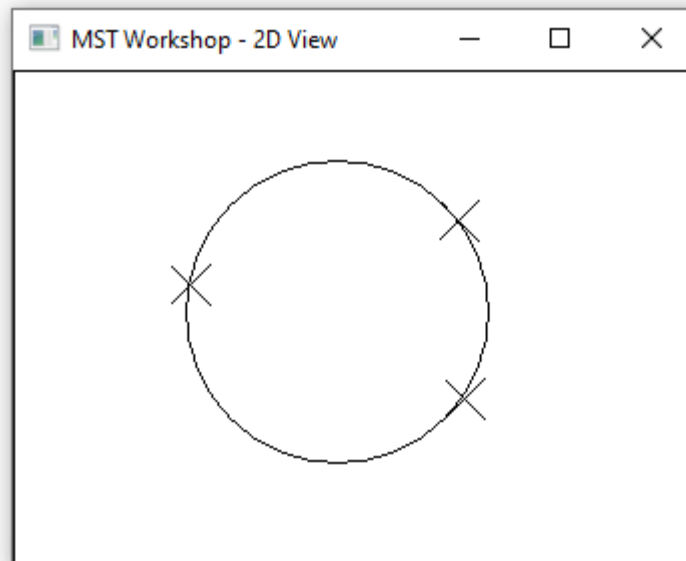
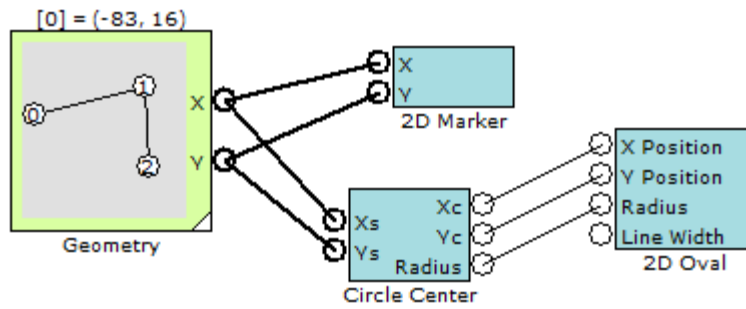
# 3D Circle

The 3D Circle draws a disc in the 3D view window. The X, Y, and Z inputs set the position and size of the circle. The Affine [4x4] array input modifies the size, position and orientation. 3D - column 4



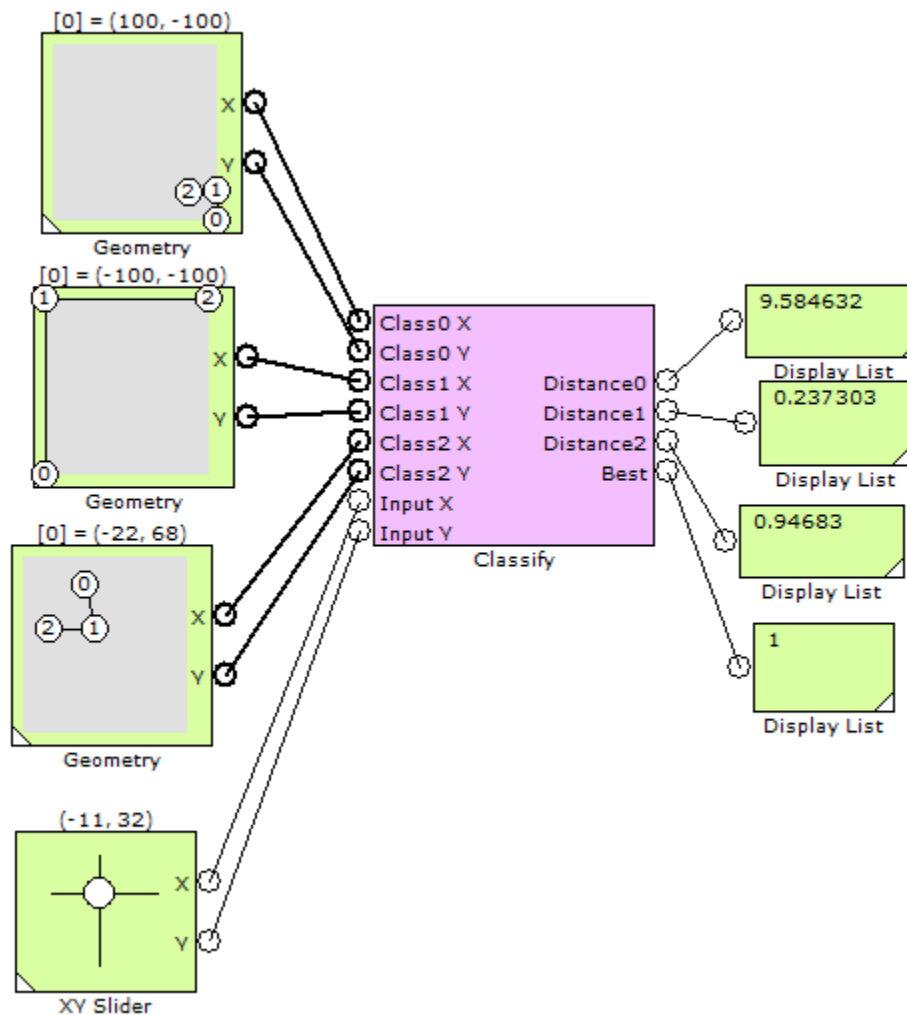
# Circle Center

The Circle Center calculates the center of a circle from 3 points on its circumference. 2D - column 2



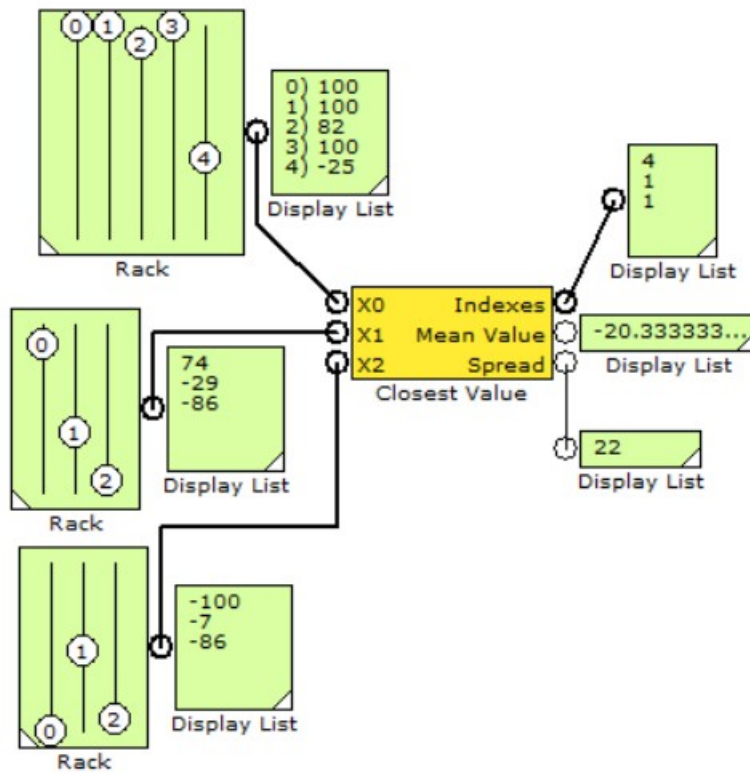
# Classify

The Classify component is given coordinates of a number of classes. The centroid and standard deviation are calculated for each class. The distance from the centroid of the class to the input is given in standard deviation units, along with the index of the closest class. Arrays – column 2



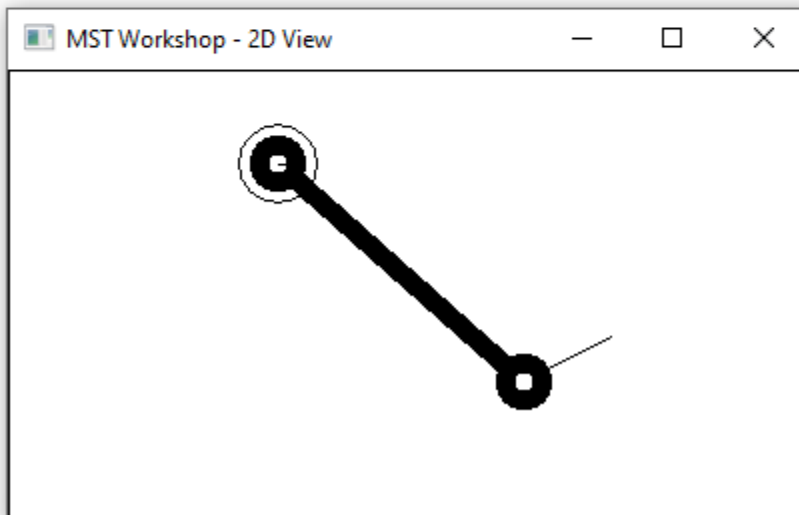
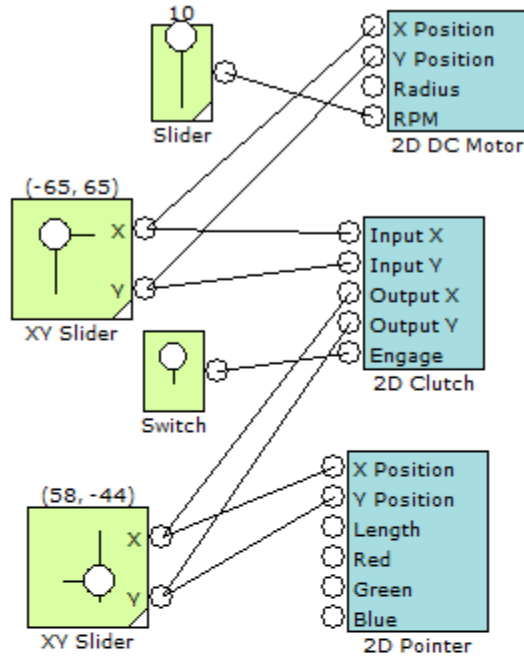
# Closest Value

The Closest Value component takes arrays of coordinates at several inputs and outputs the indexes of the closest value in 1, 2, or 3 dimensions. Functions – column 8



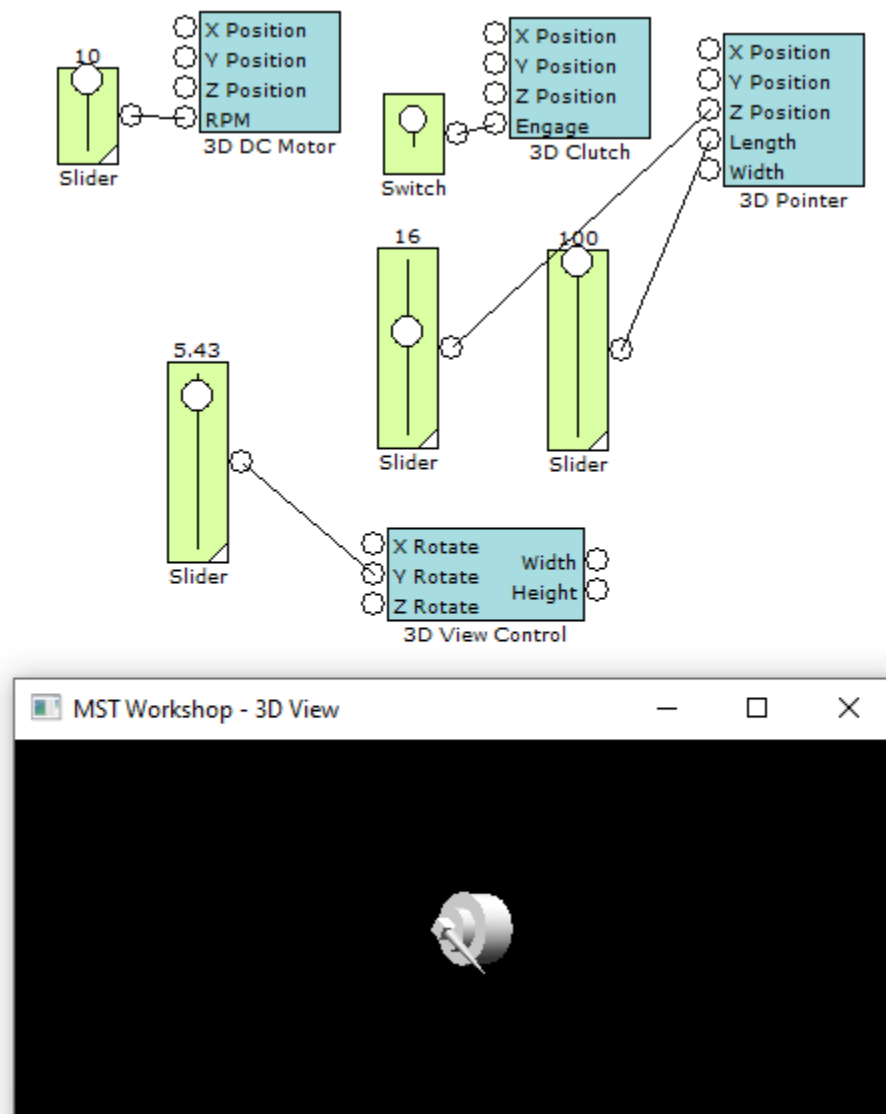
## 2D Clutch

The Clutch component creates a connection between two shafts that can be connected or disconnected. You can attach a driver to one shaft and drive something at the other shaft. Place the Clutch at the same X, Y position as a motor and place a gear at the X, Y coordinates as the other shaft. 2D - column 3



## 3D Clutch

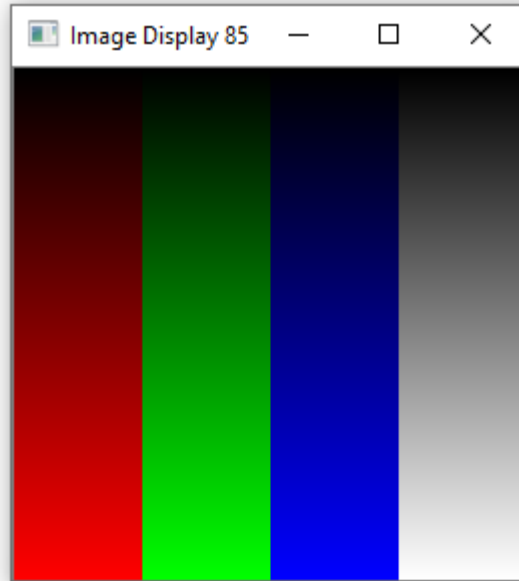
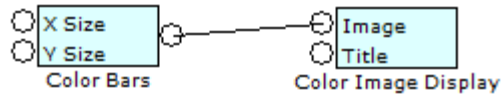
The 3D Clutch component creates a connection between two shafts that can be connected or disconnected. Values between 0 and 1 cause a fraction of the input to get to the output. You can attach a driver to the input and drive something at the end of the shaft. Place the Clutch at the same X, Y, Z position as a motor and place a gear at the X, Y, Z coordinates as the other end. 3D - column 4





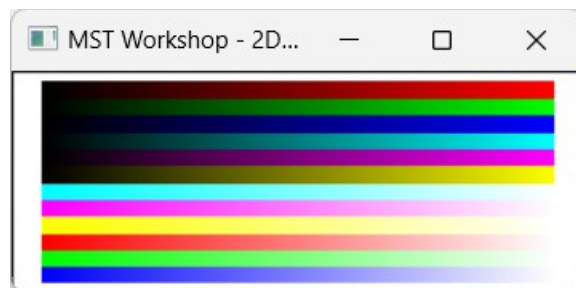
# Color Bars

The Color Bars component creates an image with a test pattern of color bars. Images - column 1



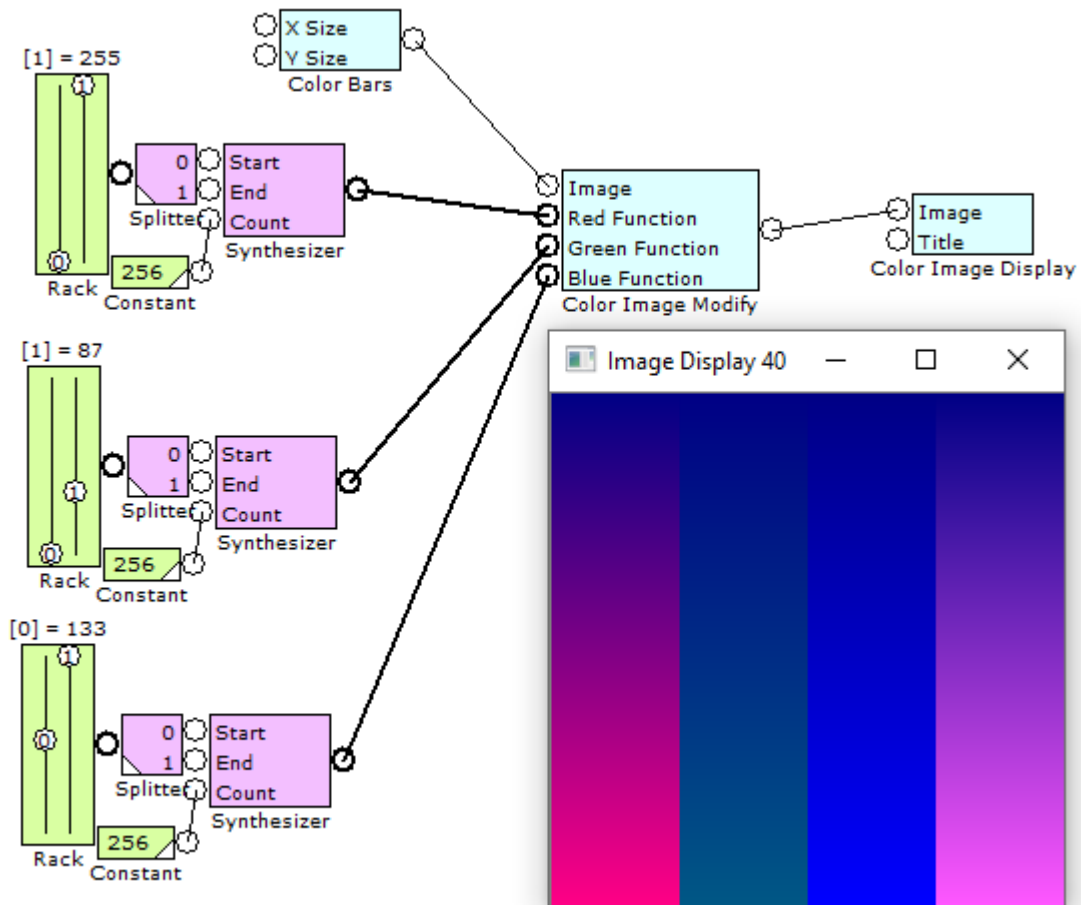
## 2D Color Bars

The 2D Color Bars component draws color bars on the 2D view canvas. 2D – column 2



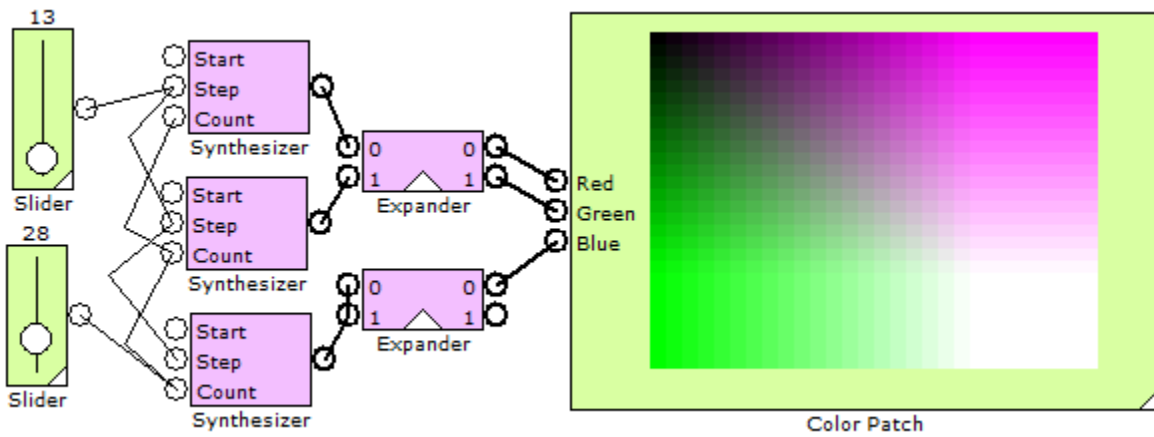
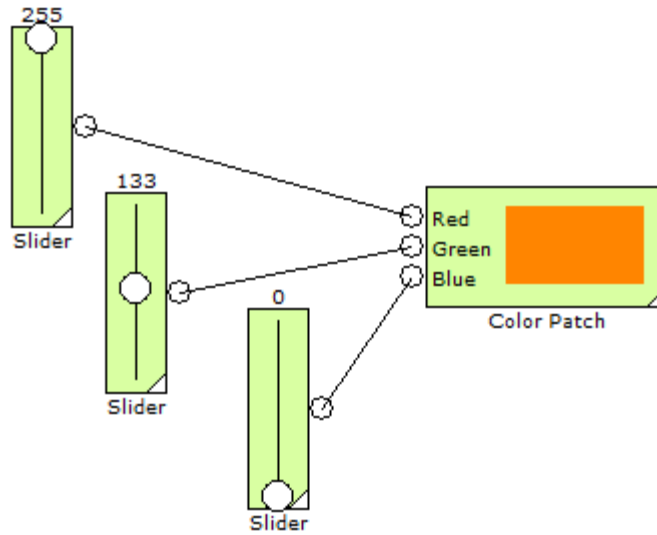
# Color Image Modify

The Color Image Modify component remaps the color image based on the 3 color function inputs.  
Images - column 2



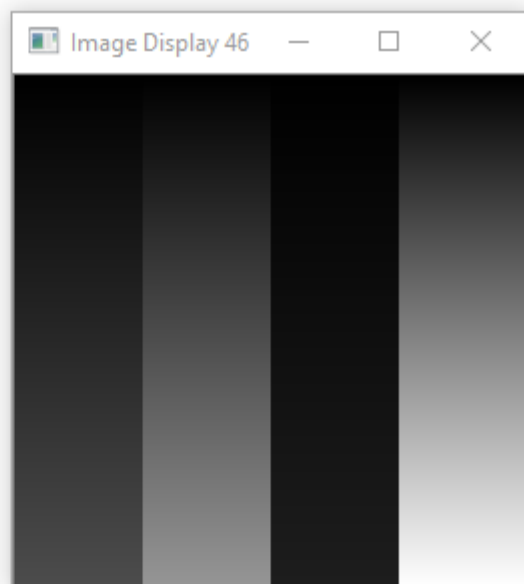
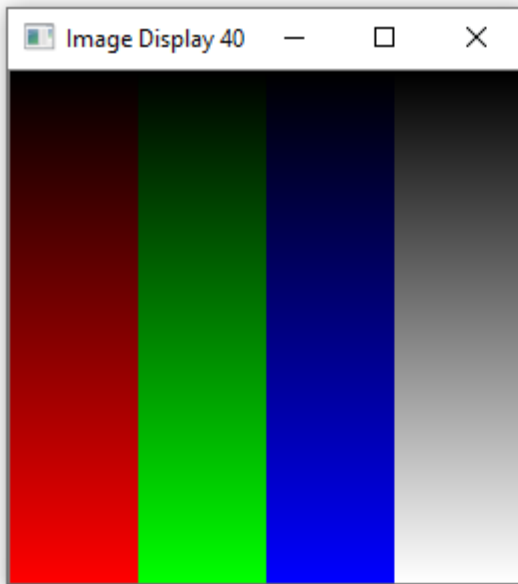
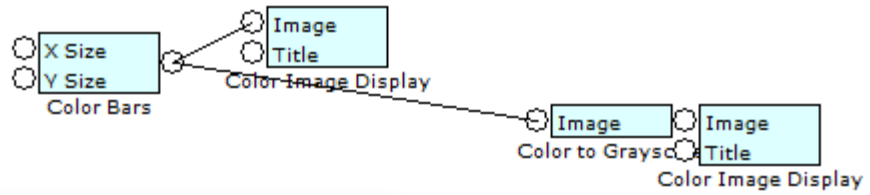
# Color Patch

The Color Patch shows you any color based on the Red, Green and Blue inputs. Controls - column 4



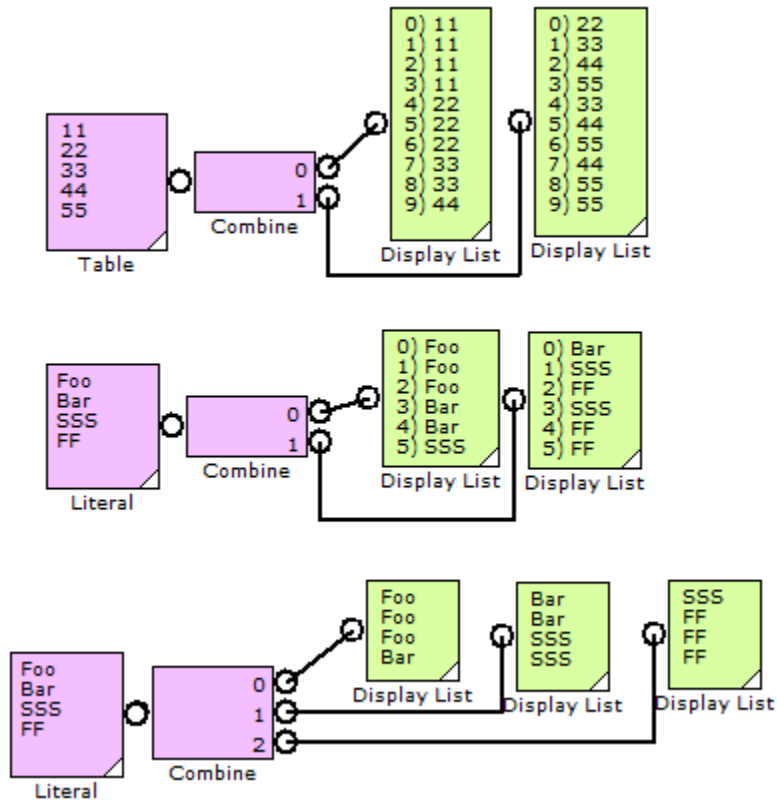
# Color to Grayscale

The Color to Grayscale component converts the color image to grayscale. Images - column 1



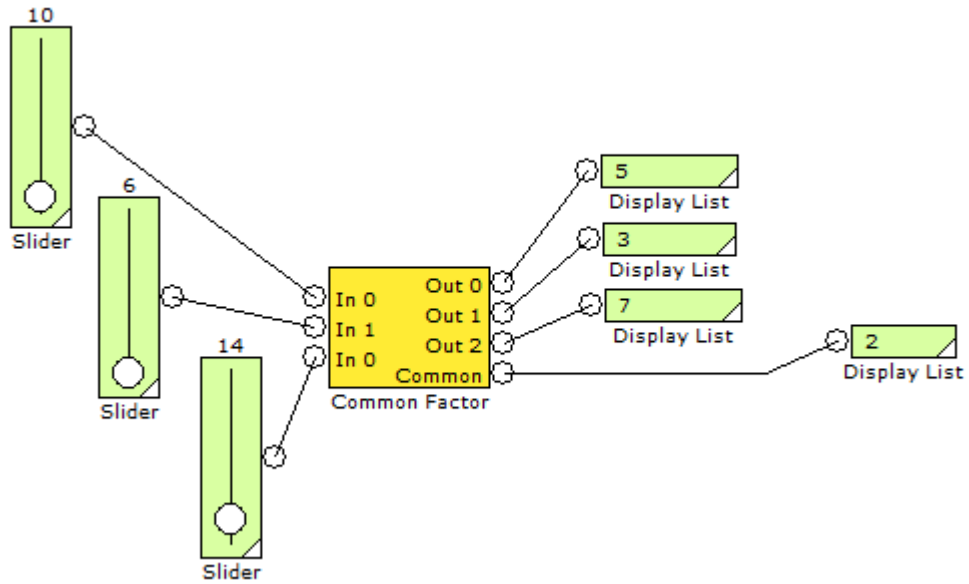
# Combine

The Combine component outputs a list of numbers taken from the Pool input with the specified Group Size. Arrays - column 5



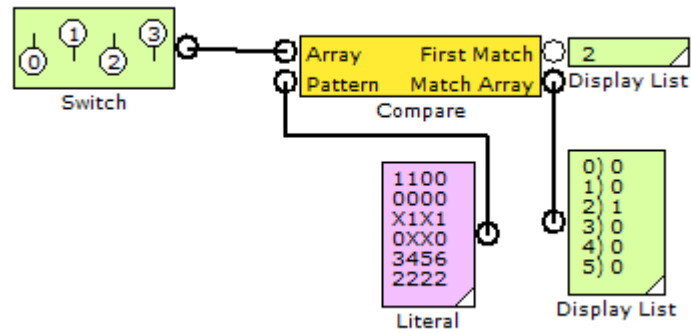
# Common Factor

The Common Factor component takes a list of numbers then removes and outputs the common factor.  
Functions - column 9



# Compare

The Compare component compares the input array of Booleans with the patterns. The output is the first pattern that matches and an array of all the patterns that match. Use 'X' for don't care. You can start the string with 'ALL ' or 'ANY ' to control if ALL the bits must match or if ANY matching bits are a match. An 'X' in place of a '1' or '0' skips the comparison. Functions - column 2





# 3D Cone

The 3D Cone draws a cone in the 3D view window. The X, Y, and Z inputs set the position and size of the cone. The Affine [4x4] array input modifies the size, position and orientation.

Rotation:

X Clockwise  
 X Counter Clockwise

Y Clockwise  
 Y Counter Clockwise

Z Clockwise  
 Z Counter Clockwise

X zero is North  
 X zero is East  
 X zero is South  
 X zero is West

Y zero is North  
 Y zero is East  
 Y zero is South  
 Y zero is West

Z zero is North  
 Z zero is East  
 Z zero is South  
 Z zero is West

OK Cancel

Pin options:

X Position    Y Position    Z Position

X Rotate    Y Rotate    Z Rotate

Radius    Height

Red    Green    Blue

Alpha

[4x4]

OK Cancel

3D Cone

Number of Slices:

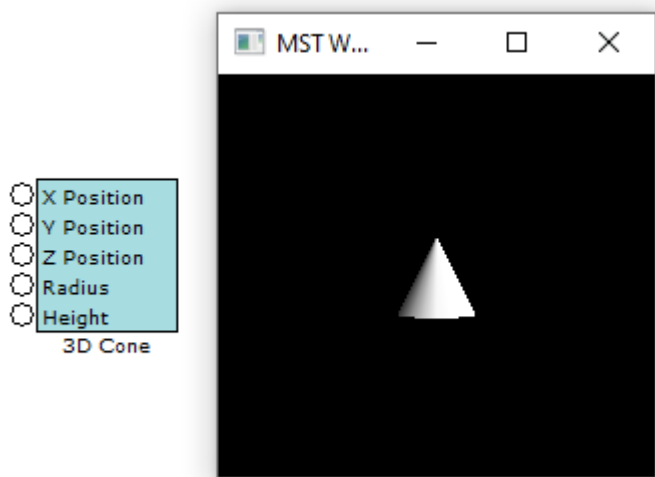
Number of Stacks:

X Offset:

Y Offset:

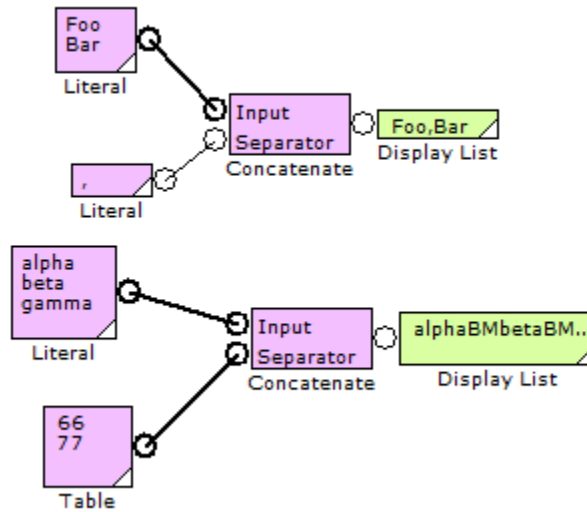
Z Offset:

OK Cancel



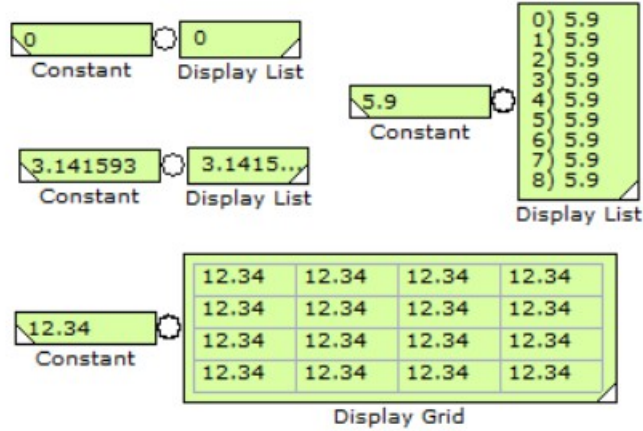
# Concatenate

The Concatenate component combines all the values at its input into a single output. You can optionally provide a separator. The separator can be either a string or an array of ASCII values. Arrays - column 6



# Constant

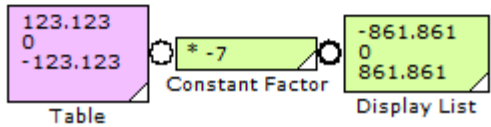
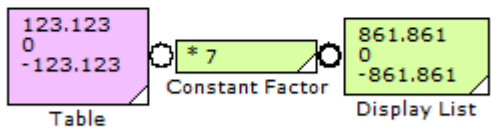
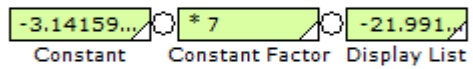
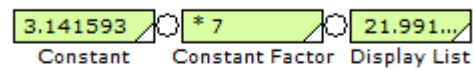
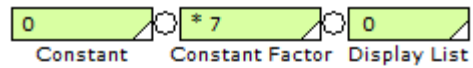
The Constant component is used to create a single number or an array of that number. Controls - column 1



<b>Move To Top</b>	Delete
<b>Move Up</b>	Details...
<b>Move Down</b>	Duplicate
<b>Move To Bottom</b>	Edit Label...
<b>Feet to Meters</b>	Enter Constant...
<b>Meters to Feet</b>	Pi
<b>Miles to Kilom.</b>	2 * Pi
<b>Kilom. to Miles</b>	Pi / 2
<b>Pound to Kilo</b>	e
<b>Kilo to Pound</b>	log(10)
<b>Degree to Radian</b>	Earth Radius (mi)
<b>Radian to Degree</b>	Speed of Light (m/sec)
<b>Use 2D mouse X</b>	Gravity (m/sec/sec)
<b>Use 2D mouse Y</b>	Mass of Proton (kg)
	Mass of Electron (kg)
	Coulomb's (m/F)
	Gravitational (N-m <sup>2</sup> /kg <sup>2</sup> )

# Constant Factor

The Constant Factor component multiplies its inputs by a single number. Controls - column 2

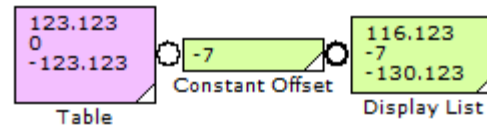
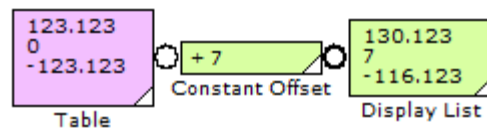
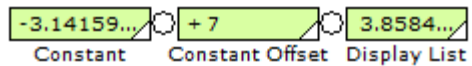
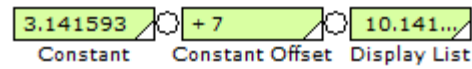
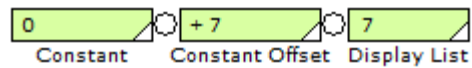


Move To Top	Delete
Move Up	Details...
Move Down	Duplicate
Move To Bottom	Edit Label...
Feet to Meters	Enter Constant...
Meters to Feet	Pi
Miles to Kilom.	2 * Pi
Kilom. to Miles	Pi / 2
Pound to Kilo	e
Kilo to Pound	log(10)
Degree to Radian	Earth Radius (mi)
Radian to Degree	Speed of Light (m/sec)
Use 2D mouse X	Gravity (m/sec/sec)
Use 2D mouse Y	Mass of Proton (kg)
	Mass of Electron (kg)
	Coulomb's (m/F)
	Gravitational (N-m2/kg2)

# Constant Offset

The Constant Offset component add a constant number to its inputs.

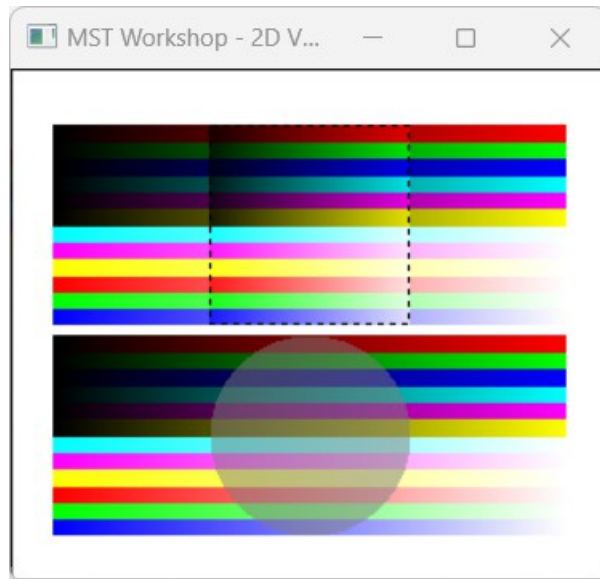
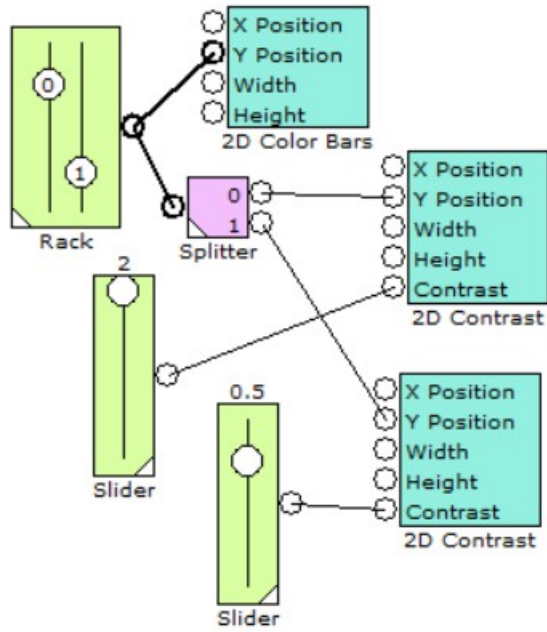
Controls - column 3



Move To Top	Delete
Move Up	Details...
Move Down	Duplicate
Move To Bottom	Edit Label...
Feet to Meters	Enter Constant...
Meters to Feet	Pi
Miles to Kilom.	2 * Pi
Kilom. to Miles	Pi / 2
Pound to Kilo	e
Kilo to Pound	log(10)
Degree to Radian	Earth Radius (mi)
Radian to Degree	Speed of Light (m/sec)
Use 2D mouse X	Gravity (m/sec/sec)
Use 2D mouse Y	Mass of Proton (kg)
	Mass of Electron (kg)
	Coulomb's (m/F)
	Gravitational (N-m2/kg2)

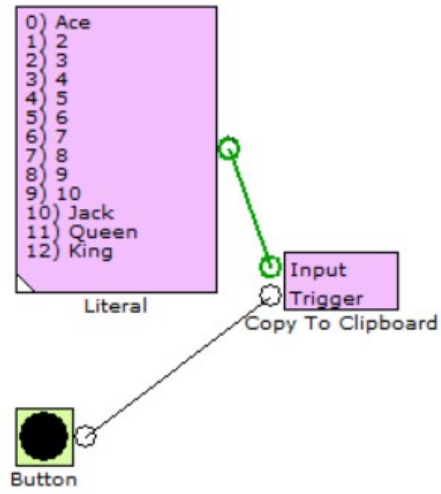
# 2D Contrast

The 2D Contrast component intensifies or softens the 2D drawing it covers. 2D – column 1



# Copy To Clipboard

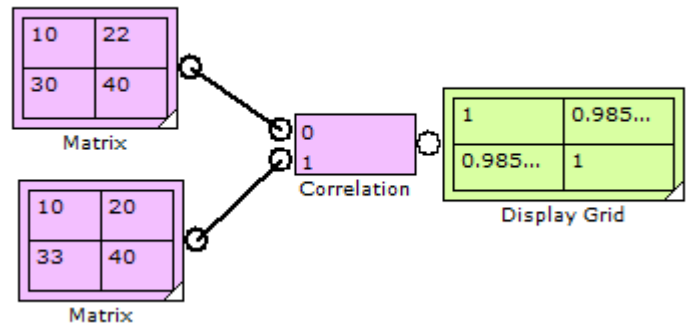
The Copy To Clipboard component puts text into the clipboard. Arrays – column 5



# Correlation

The Correlation component calculates the Pearson Correlation matrix for the input matrix.  
- column 3

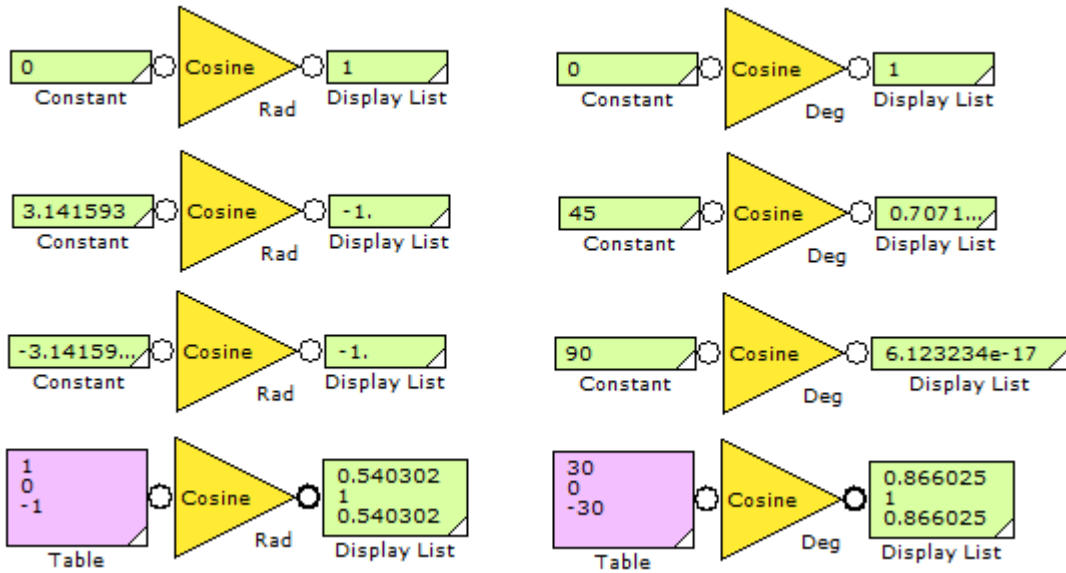
Arrays





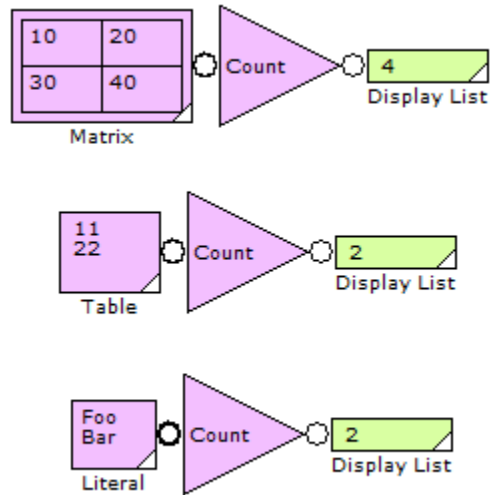
# Cosine

The Cosine component calculates the trigonometric cosine of the input (in radians). Input can be scalar or an array. Functions - column 5



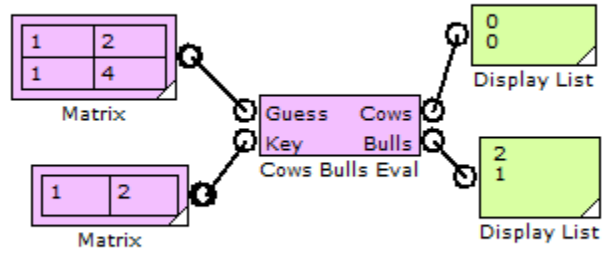
# Count

The Count component scans the input array and outputs the number of elements in the array. Arrays - column 3



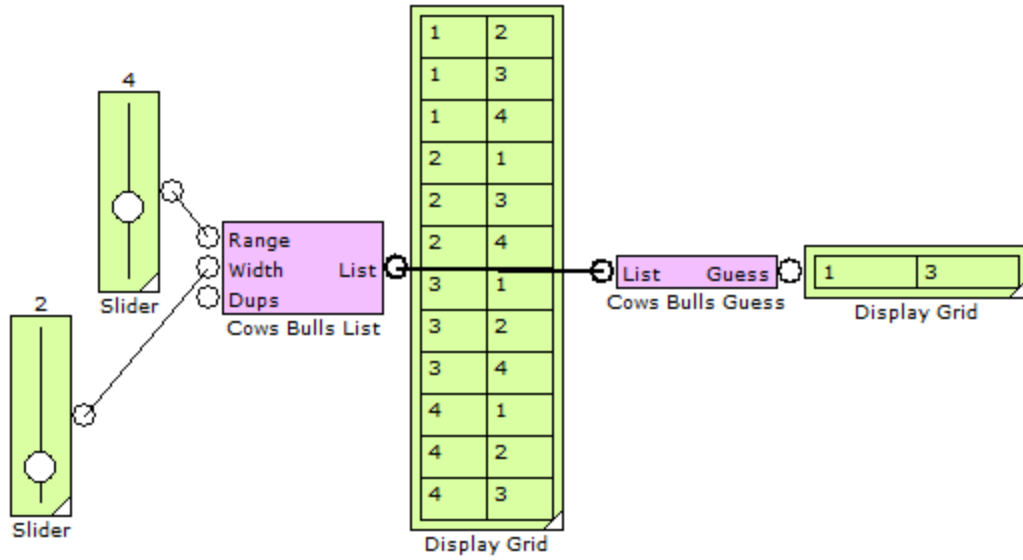
# Cows Bulls Eval

The Cows Bulls Eval component evaluates the Guess inputs against the Key input and outputs Cows and Bulls based on how the values match. Arrays - column 5



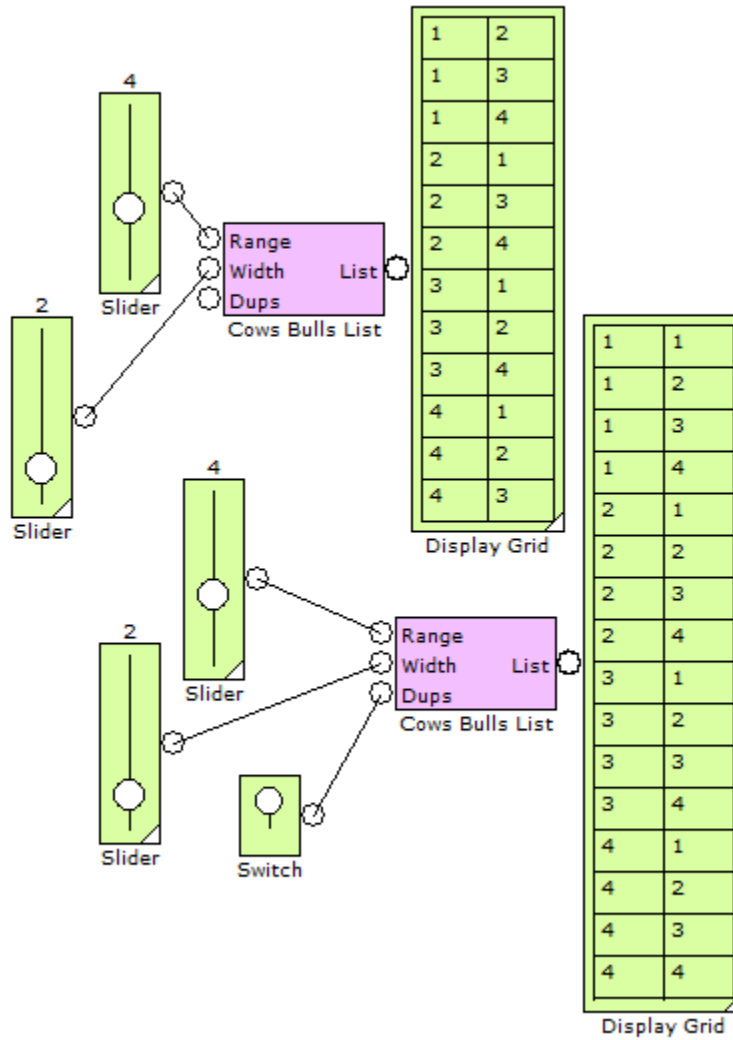
# Cows Bulls Guess

The Cows Bulls Guess component the best guess to minimize future steps to success. Arrays - column 5



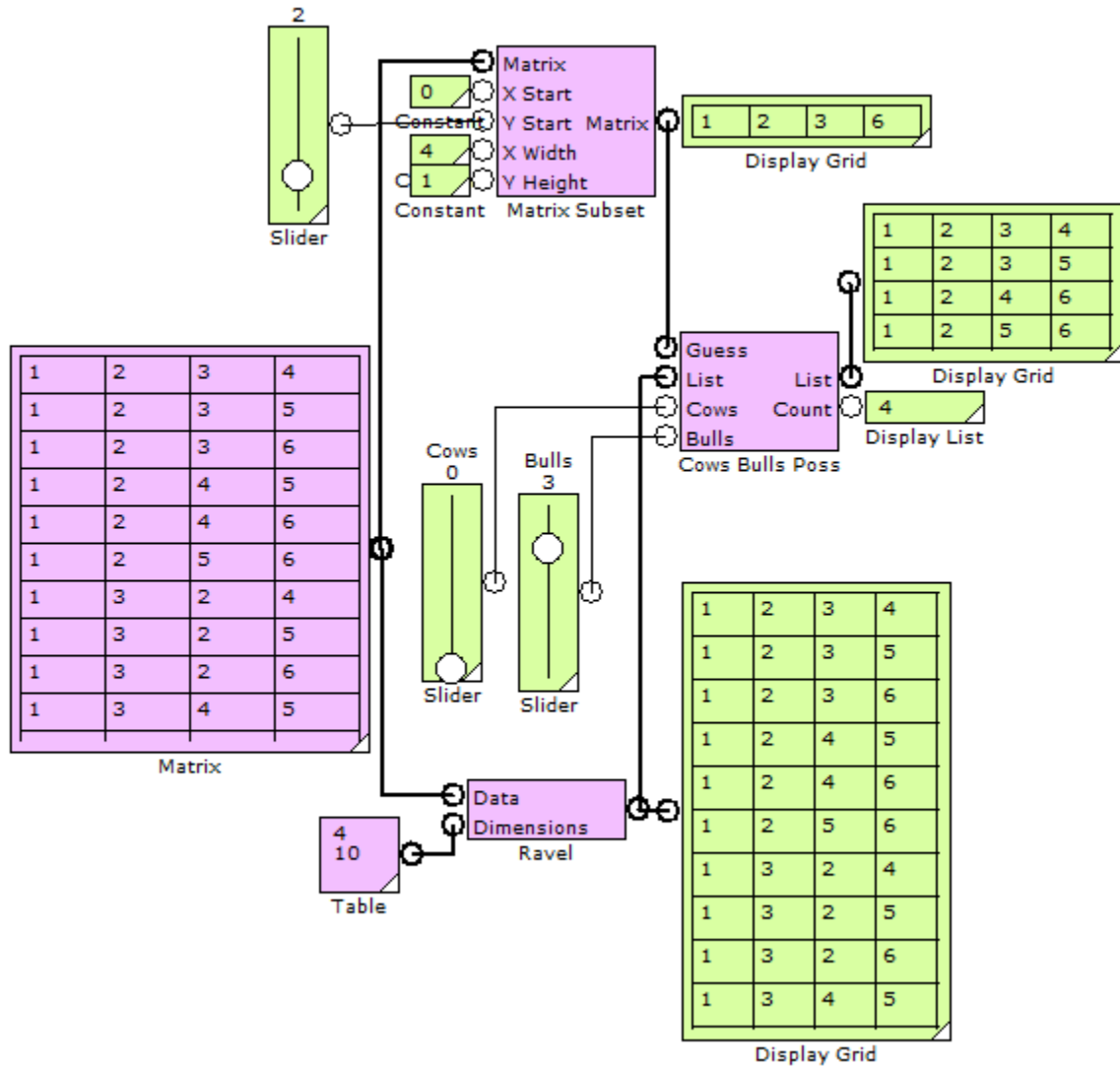
# Cows Bulls List

The Cows Bulls List component creates a list of all possible combinations based on the inputs. Arrays - column 5



# Cows Bulls Poss

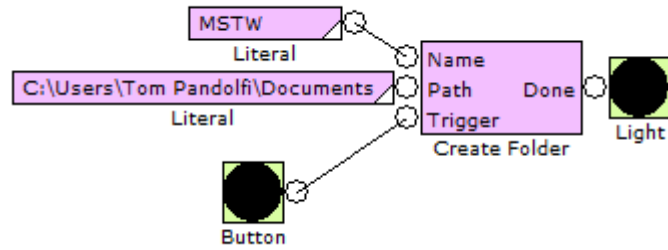
The Cows Bulls Poss component lists all the possible combinations that match the inputs. Arrays - column 5



# Create Folder

The Create Folder component creates a new folder with the specified name.

Arrays - column 7



## 2D Cross

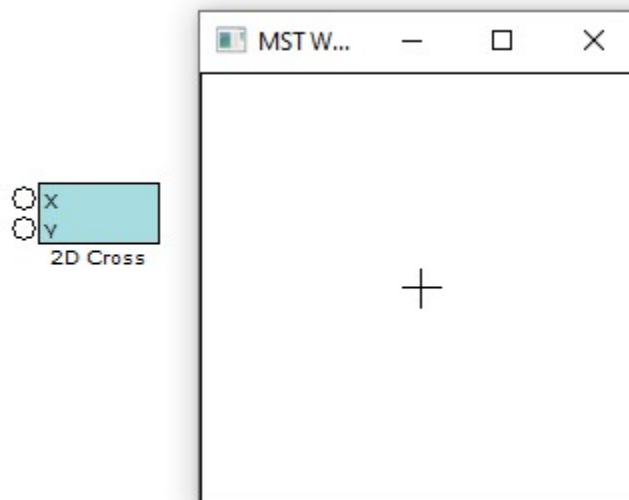
The 2D Cross component draws a small cross at the X, Y location.

2D - column 1

Pin options:

<input checked="" type="checkbox"/>	X	Y	
<input type="checkbox"/>	Size		
<input type="checkbox"/>	Line Width		
<input type="checkbox"/>	Line Red	Line Green	Line Blue
<input type="checkbox"/>	Alpha		

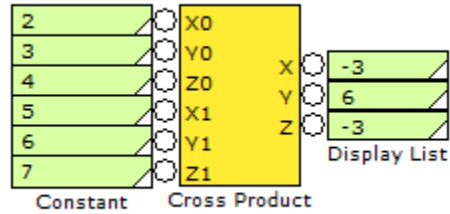
OK Cancel





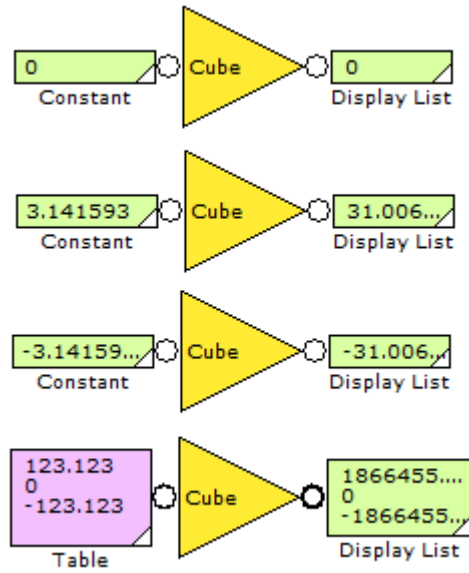
# Cross Product

The Cross Product component calculates the cross product of 2 3D vectors. The result is a vector that is perpendicular to both inputs. Functions - column 8



# Cube

The Cube component multiplies the input value times itself twice. Input can be scalar or an array.  
Functions - column 1



# 3D Cube

The 3D Cube draws a box in the 3D view window. The X, Y, and Z inputs set the position and size of the box. The Affine [4x4] array input modifies the size, position and orientation. 3D - column 4

Rotation:

X Clockwise  
 X Counter Clockwise  
 Y Clockwise  
 Y Counter Clockwise  
 Z Clockwise  
 Z Counter Clockwise

X zero is North  
 X zero is East  
 X zero is South  
 X zero is West  
 Y zero is North  
 Y zero is East  
 Y zero is South  
 Y zero is West  
 Z zero is North  
 Z zero is East  
 Z zero is South  
 Z zero is West

OK Cancel

Pin options:

X Position     Y Position     Z Position  
 X Rotate     Y Rotate     Z Rotate  
 X Size     Y Size     Z Size  
 Red     Green     Blue  
 Alpha  
 [4x4]

OK Cancel

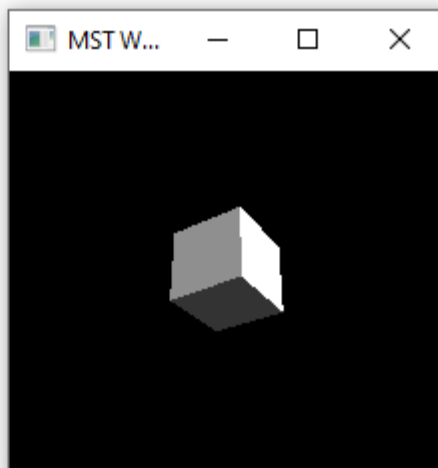
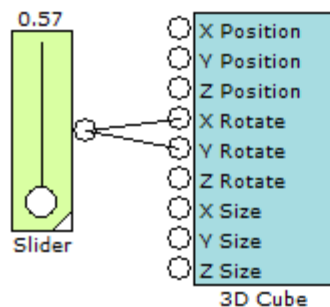
3D Cube

X Offset:

Y Offset:

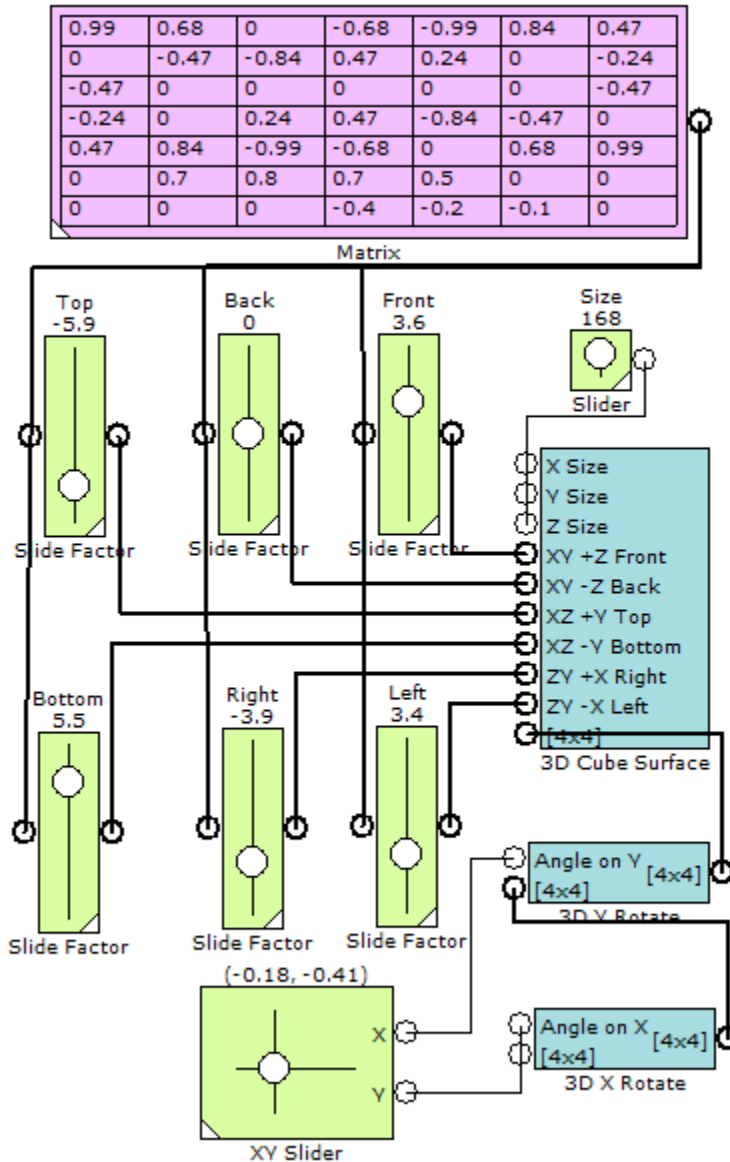
Z Offset:

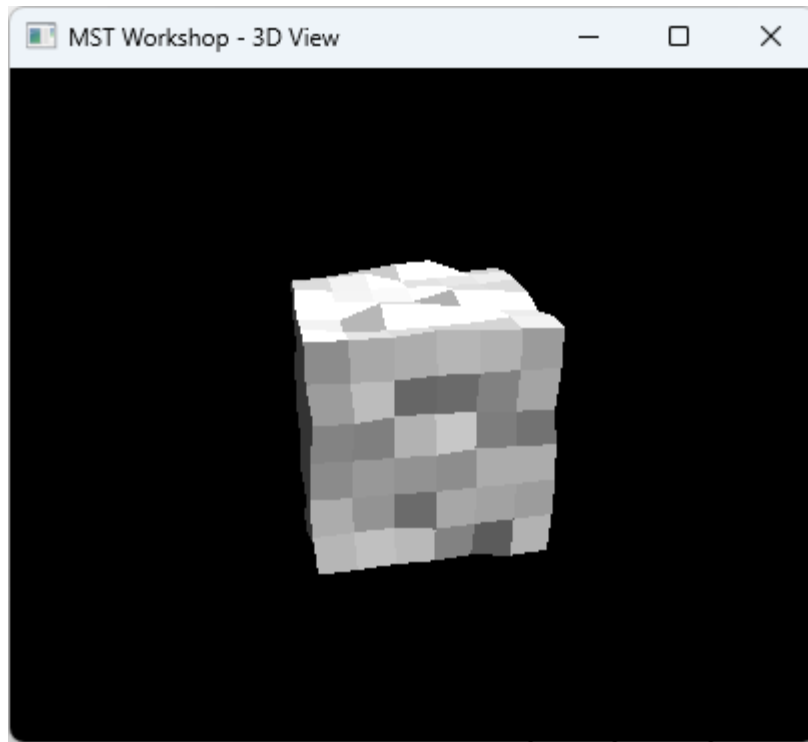
OK Cancel



## 3D Cube Surface

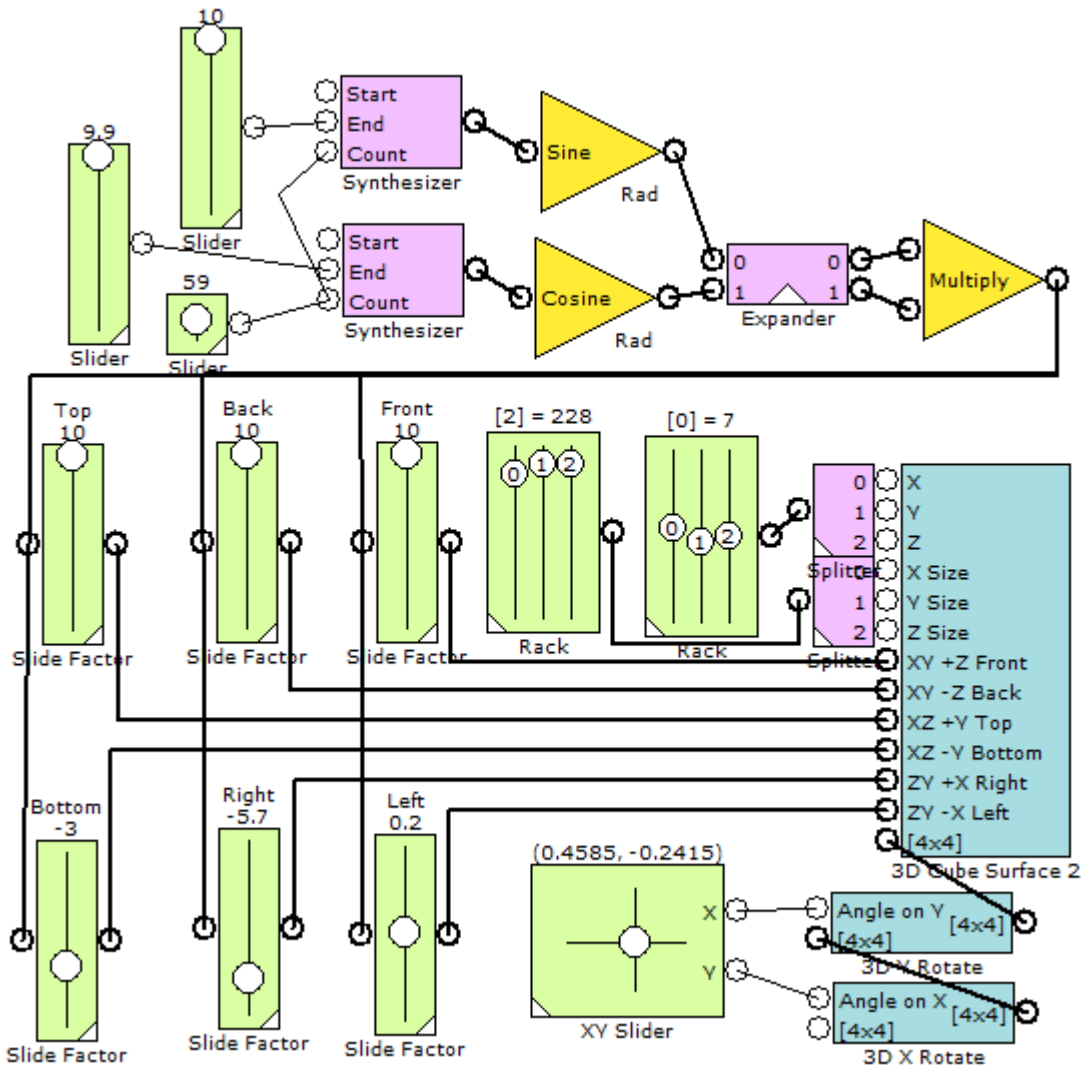
The 3D Cube Surface component draws a cube in the 3D view window using up to 6 surfaces. The 6 surfaces are enabled in the Options menu. Each surface must be a 2D matrix. Each surface matrix must match the others in X, Y, and Z sides of the matrix. The Affine [4x4] array input modifies the size, position and orientation. 3D - column 5

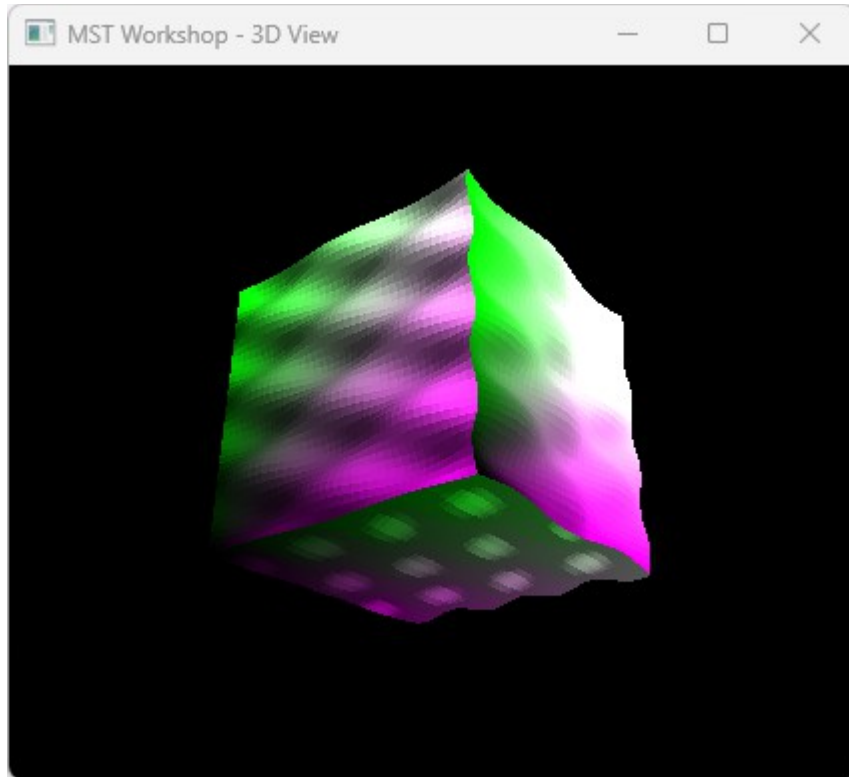
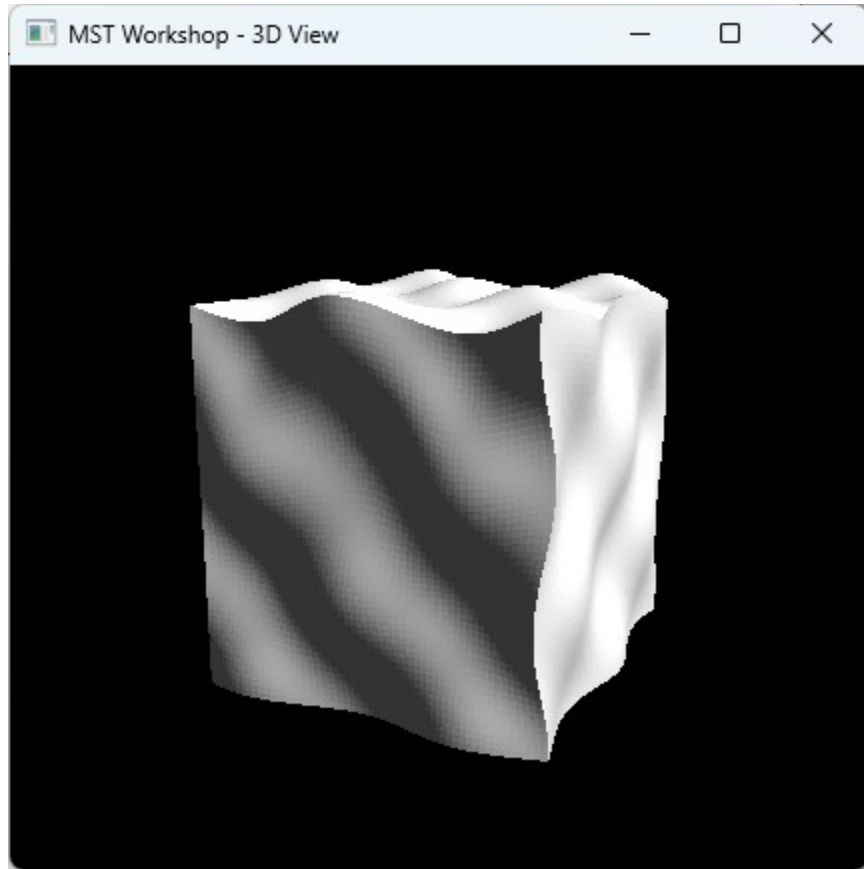




## 3D Cube Surface 2

The 3D Cube Surface 3D Cube Surface 2 component draws a cube in the 3D view window using up to 6 surfaces. The 6 surfaces are enabled in the Options menu. Each surface must be a 2D matrix. Each surface matrix must match the others in X, Y, and Z sides of the matrix. The Affine [4x4] array input modifies the size, position and orientation. Each surface can be a different color or if arrays are input to the colors each facet will be a unique color. 3D - column 5





## Cylinder Shape

The Cylinder Shape component creates a triangular mesh composed of points and faces for a cylinder. Use its outputs to reshape or draw it in the 3D View by connecting it to the 3D Draw Shape component.  
3D – column 6



## 3D Cylinder

The 3D Cylinder draws a cylinder in the 3D view window. The X, Y, and Z inputs set the position and size of the cylinder. The Affine [4x4] array input modifies the size, position and orientation. 3D - column 4

3D Cylinder

Number of Slices:

Number of Stacks:

X Offset:

Y Offset:

Z Offset:

Close the Top

Close the Bottom

Pin options:

X Position   Y Position  Z Position

X Rotate  Y Rotate  Z Rotate

Top Radius  Bottom Radius  Height

Red  Green  Blue

Alpha

[4x4]

Rotation:

X Clockwise  X Counter Clockwise

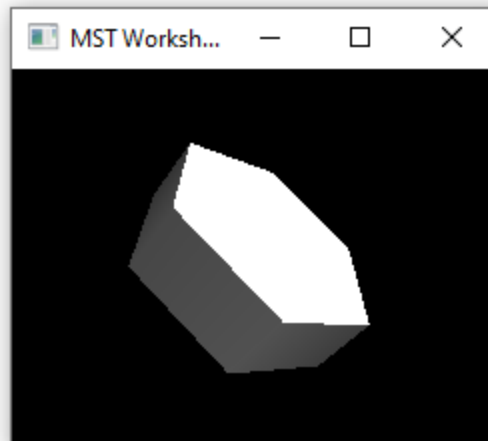
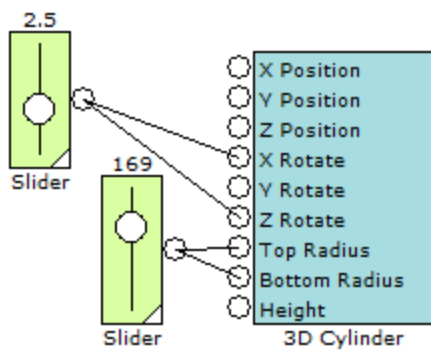
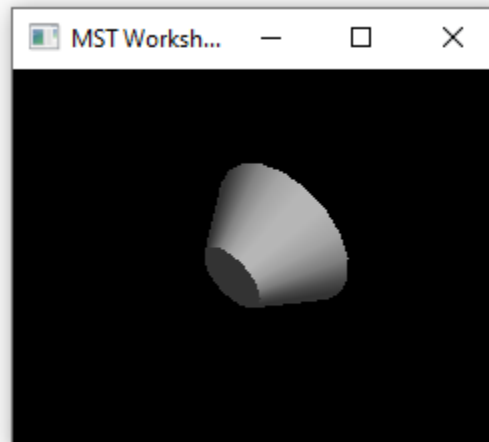
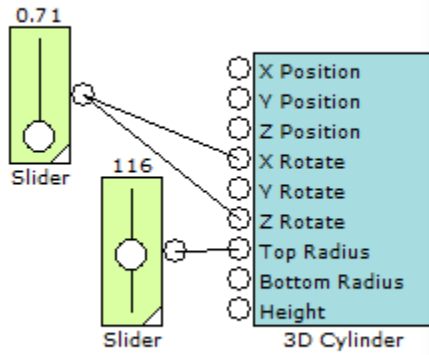
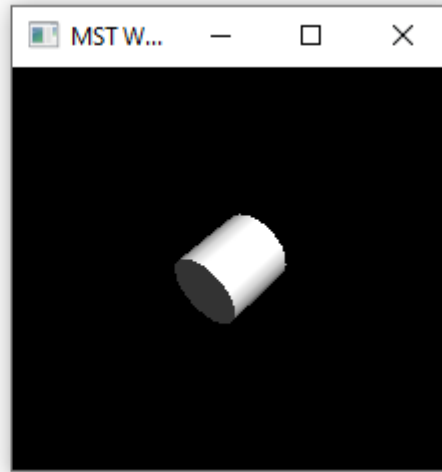
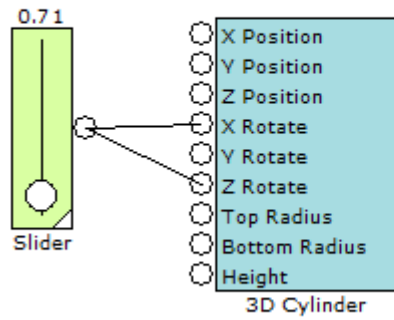
Y Clockwise  Y Counter Clockwise

Z Clockwise  Z Counter Clockwise

X zero is North  X zero is East  X zero is South  X zero is West

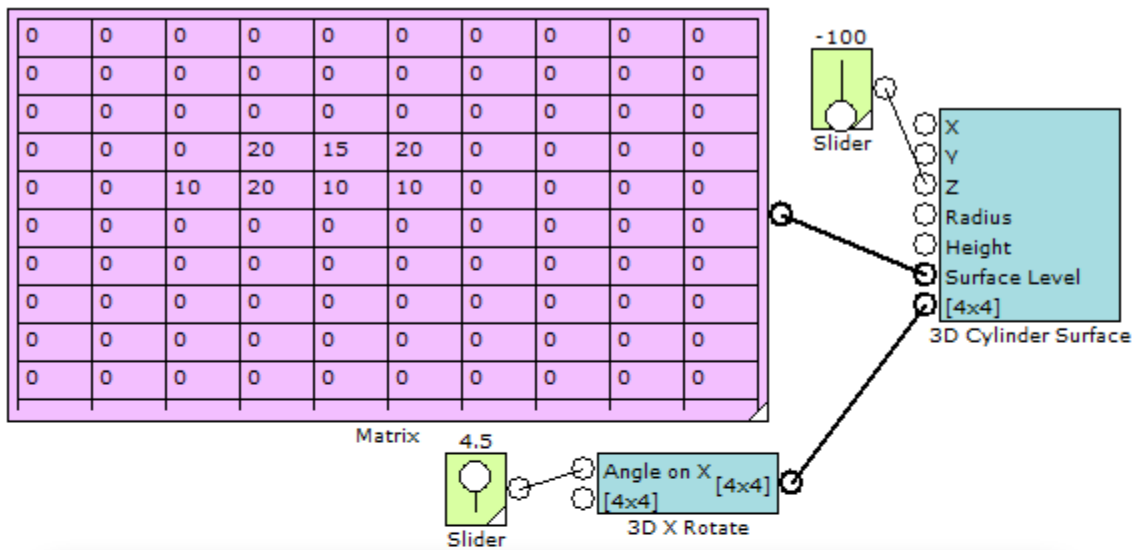
Y zero is North  Y zero is East  Y zero is South  Y zero is West

Z zero is North  Z zero is East  Z zero is South  Z zero is West



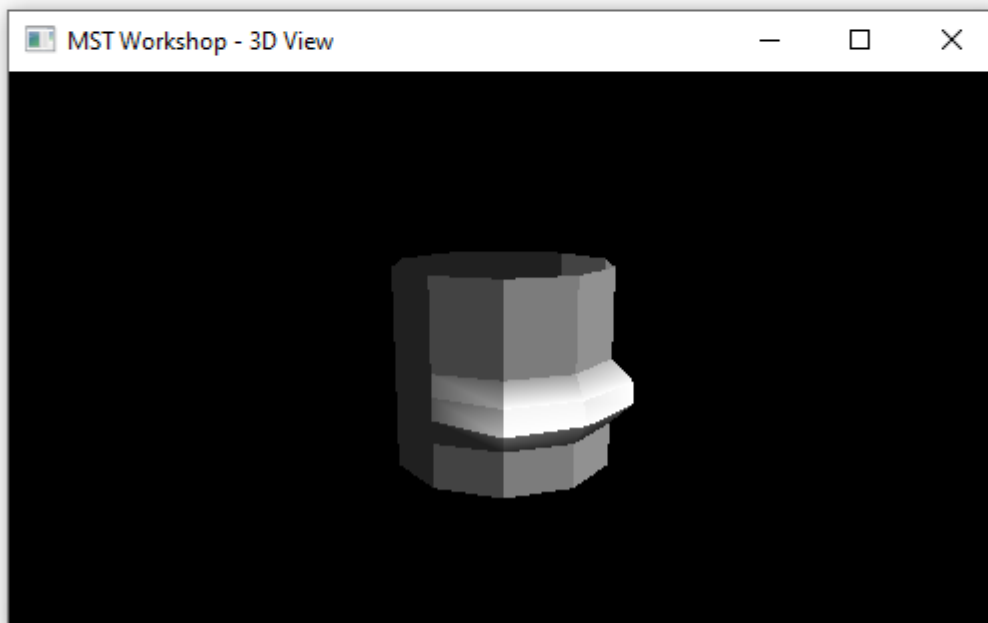
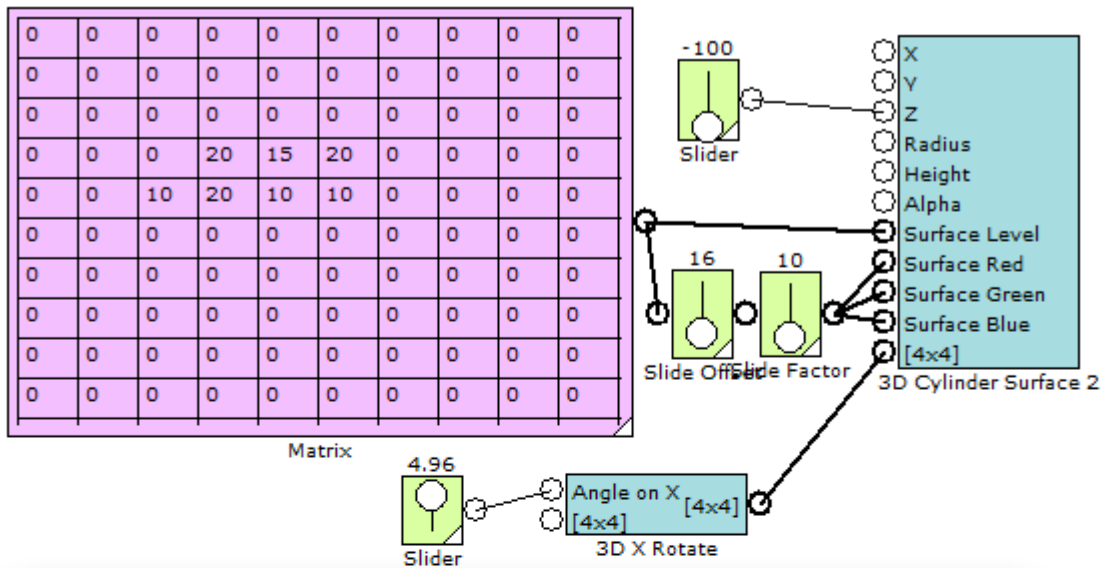
## 3D Cylinder Surface

The 3D Cylinder Surface component draws a cylinder with an embossed surface in the 3D view window. The Matrix and Mat Row Size inputs set the 2D data to be plotted. The Affine [4x4] array input modifies the size, position and orientation. 3D - column 5



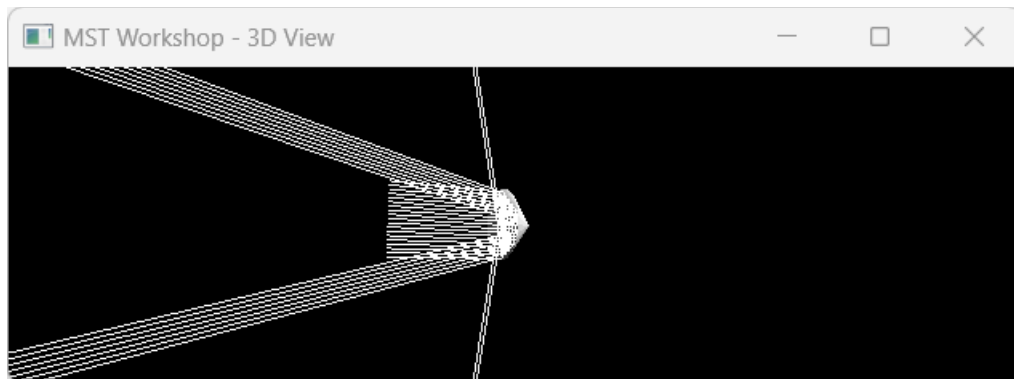
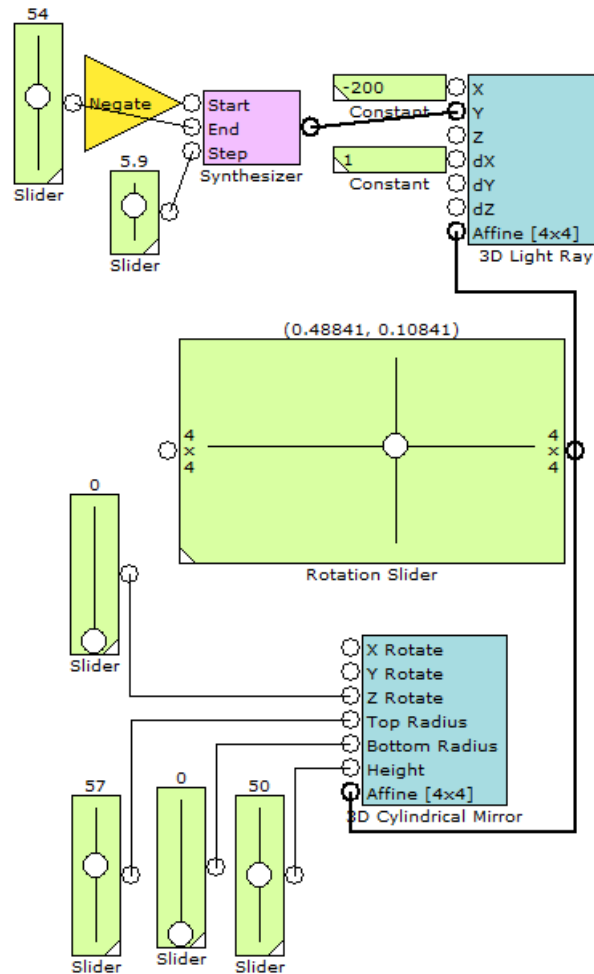
## 3D Cylinder Surface 2

The 3D Cylinder Surface 2 component draws a cylinder with an embossed and color surface in the 3D view window. The Matrix and Mat Row Size inputs set the 2D data to be plotted. The Affine [4x4] array input modifies the size, position and orientation. 3D - column 5



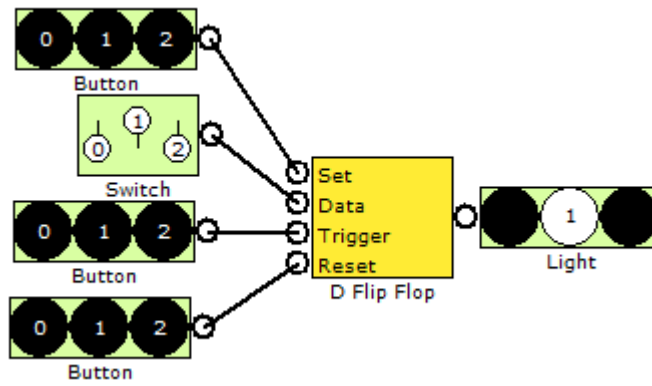
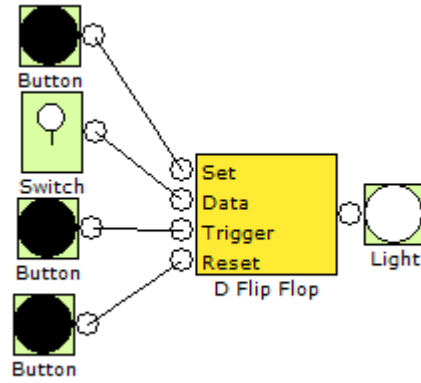
# 3D Cylindrical Mirror

The 3D Cylindrical Mirror component reflects light rays in the 3D view window. The settings determine the fraction of light that reflects and passes through. The Affine [4x4] array input modifies the size, position, and orientation of the mirror. 3D - column 5



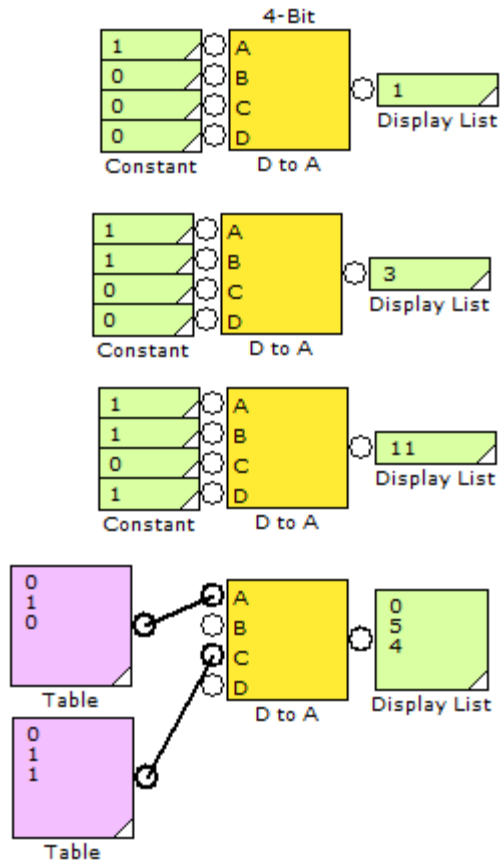
# D Flip Flop

The D Flip Flop component is a memory device that can toggle between true and false. Functions - column 3



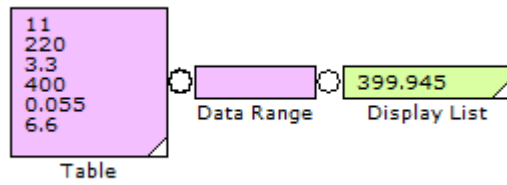
# D to A

The D to A component lets you convert digital input into analog outputs. Functions - column 3



## Data Range

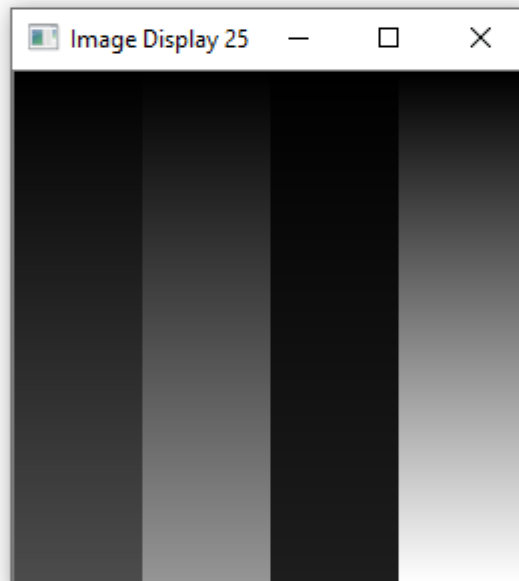
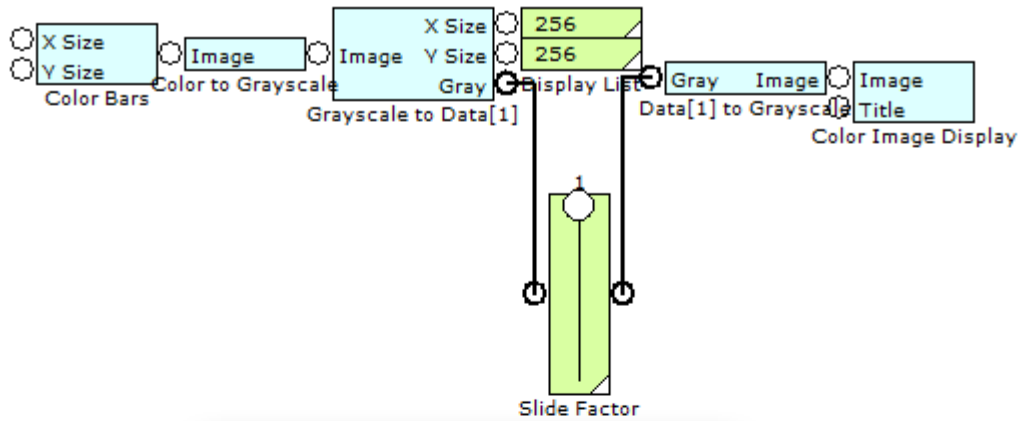
The Data Range component scans the input array and outputs the maximum value minus the minimum value. Arrays - column 3





## Data[1] to Grayscale

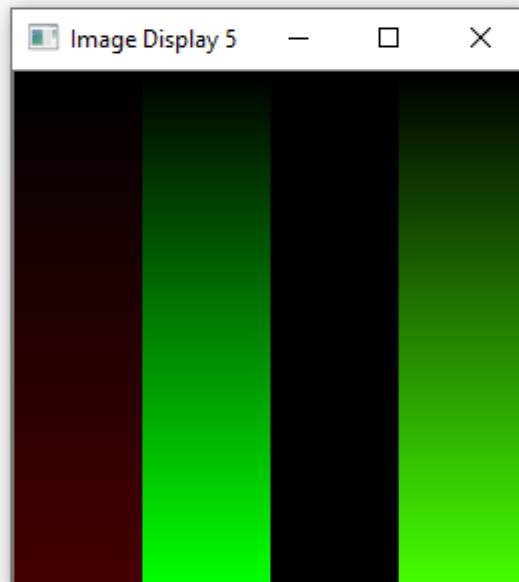
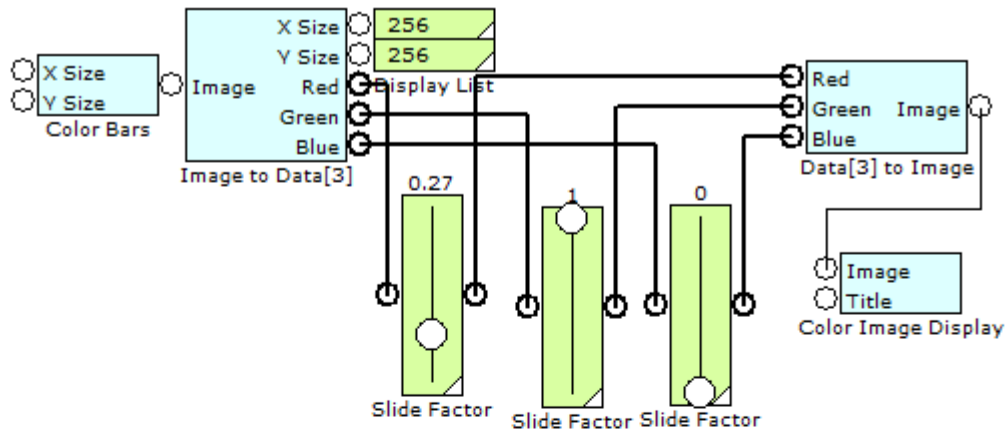
The Data[1] to Grayscale component creates a grayscale image from the input data. Images - column 5



## Data[3] to Image

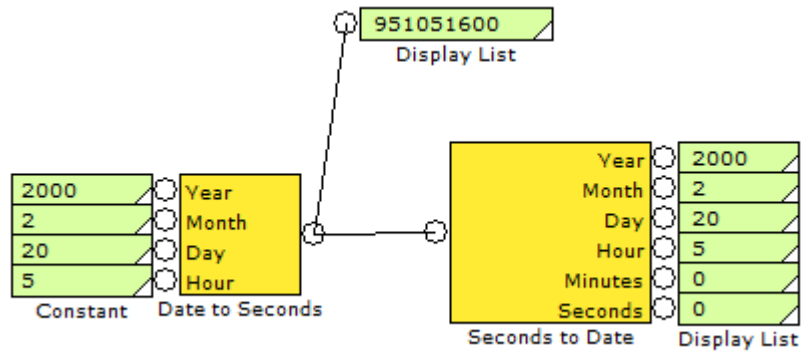
The Data[3] to Image component creates a three color image from the color data provided.  
- column 5

Images



## Date to Seconds

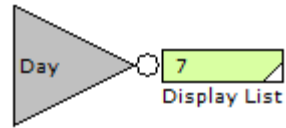
The Date to Seconds component takes a date and converts it to a number of seconds. when the input is not connected it uses the computer's internal clock. Functions - column 6



# Day

The Day component outputs the current calendar day of the month.

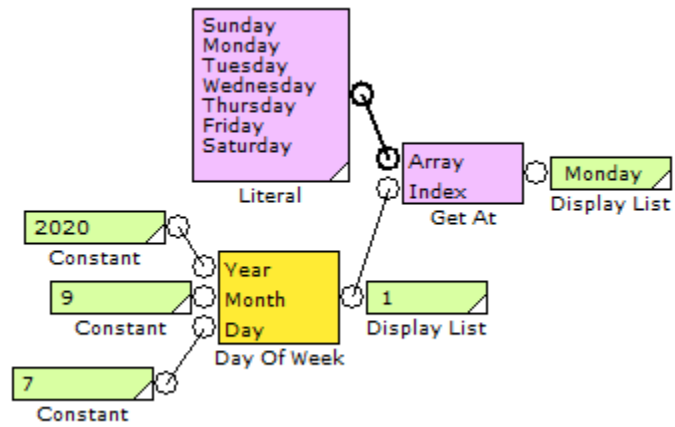
Functions - column 7



# Day Of Week

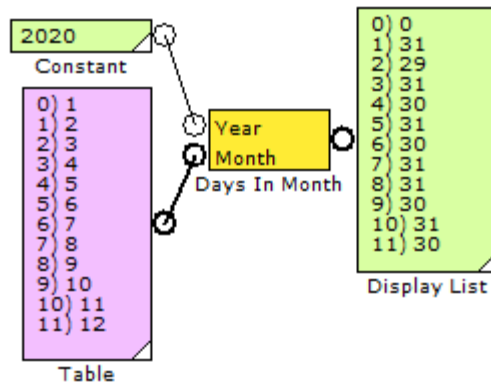
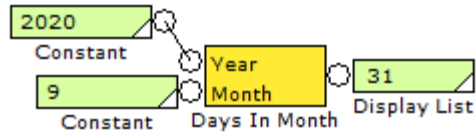
The Day Of Week component outputs 0 for Sunday, 1 for Monday... 6 for Saturday.

Functions -  
column 7



# Days In Month

The Days In Month component outputs the number of days in the given month. Functions - column 7



## 2D DC Motor

The 2D DC Motor creates a motor that can drive other simulation objects like Gears. Place a Gear at the same X, Y position to connect the motor to the gear. 2D - column 3

Rotation:

<input checked="" type="radio"/> X Clockwise	<input checked="" type="radio"/> X zero is North
<input type="radio"/> X Counter Clockwise	<input type="radio"/> X zero is East
	<input type="radio"/> X zero is South
	<input type="radio"/> X zero is West
<input checked="" type="radio"/> Y Clockwise	<input checked="" type="radio"/> Y zero is North
<input type="radio"/> Y Counter Clockwise	<input type="radio"/> Y zero is East
	<input type="radio"/> Y zero is South
	<input type="radio"/> Y zero is West
<input checked="" type="radio"/> Z Clockwise	<input checked="" type="radio"/> Z zero is North
<input type="radio"/> Z Counter Clockwise	<input type="radio"/> Z zero is East
	<input type="radio"/> Z zero is South
	<input type="radio"/> Z zero is West

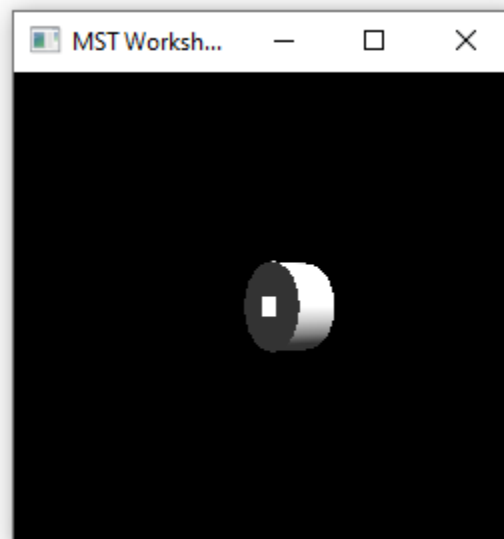
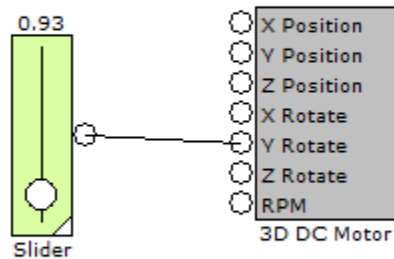
OK Cancel

3D DC Motor

Spin Up Time:

Spin Down Time:

OK Cancel



# 3D DC Motor

The 3D DC Motor creates a motor that can drive other 3D simulation objects like 3D Gears. Place a Gear at the same X, Y, Z position and orientation to connect the motor to a gear. You specify the RPM of them motor. 3D - column 4

**3D DC Motor**

Spin Up Time:

Spin Down Time:

**Rotation:**

X Clockwise  
 X Counter Clockwise

Y Clockwise  
 Y Counter Clockwise

Z Clockwise  
 Z Counter Clockwise

X zero is North  
 X zero is East  
 X zero is South  
 X zero is West

Y zero is North  
 Y zero is East  
 Y zero is South  
 Y zero is West

Z zero is North  
 Z zero is East  
 Z zero is South  
 Z zero is West

**Pin options:**

X Position  Y Position  Z Position

X Rotate  Y Rotate  Z Rotate

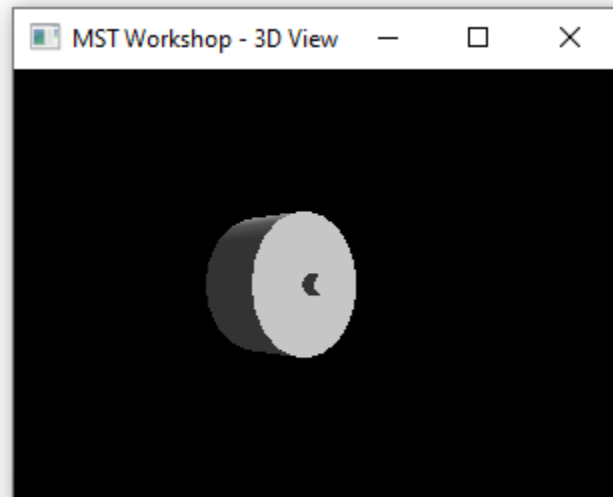
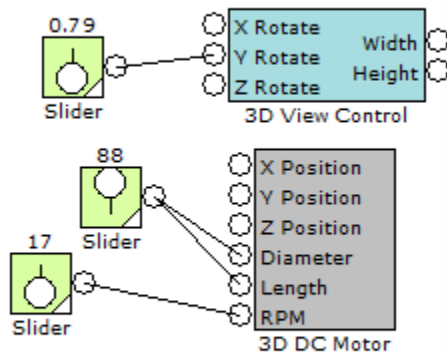
Diameter  Length

RPM

Red  Green  Blue

Alpha

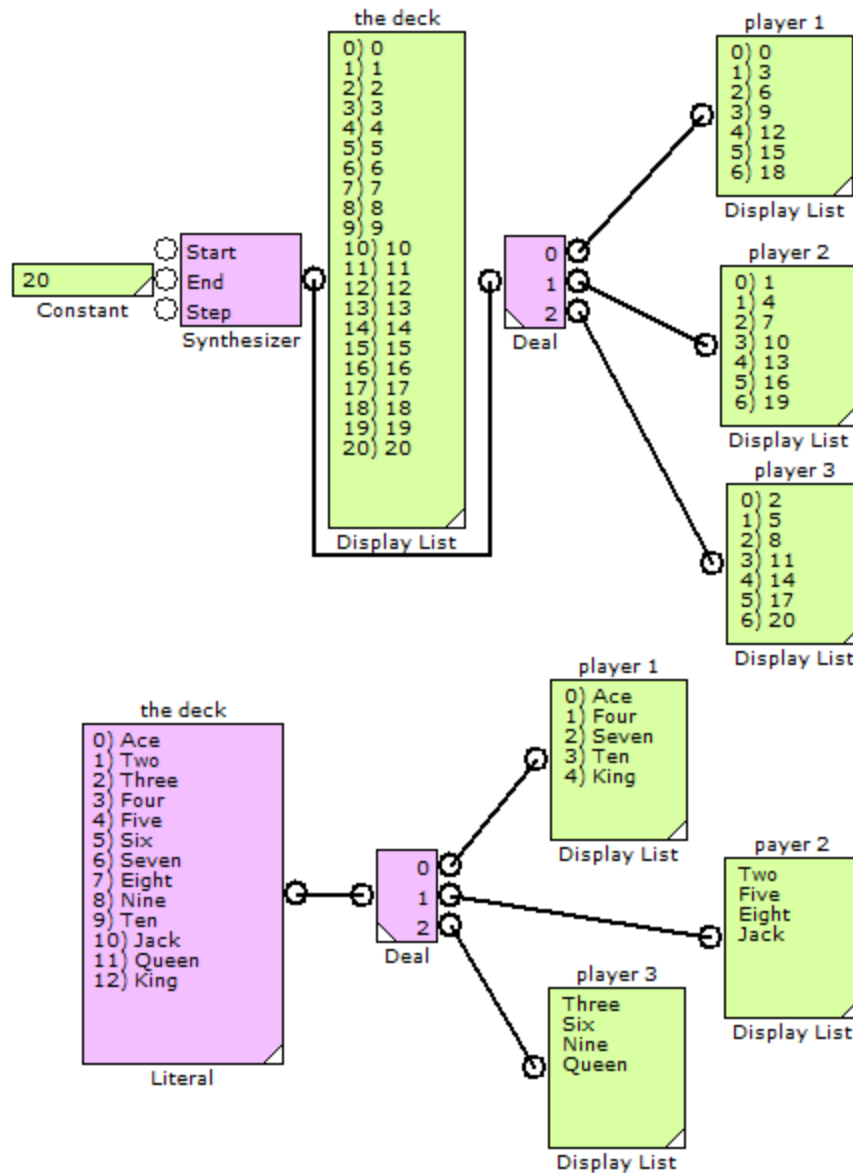
[4x4]





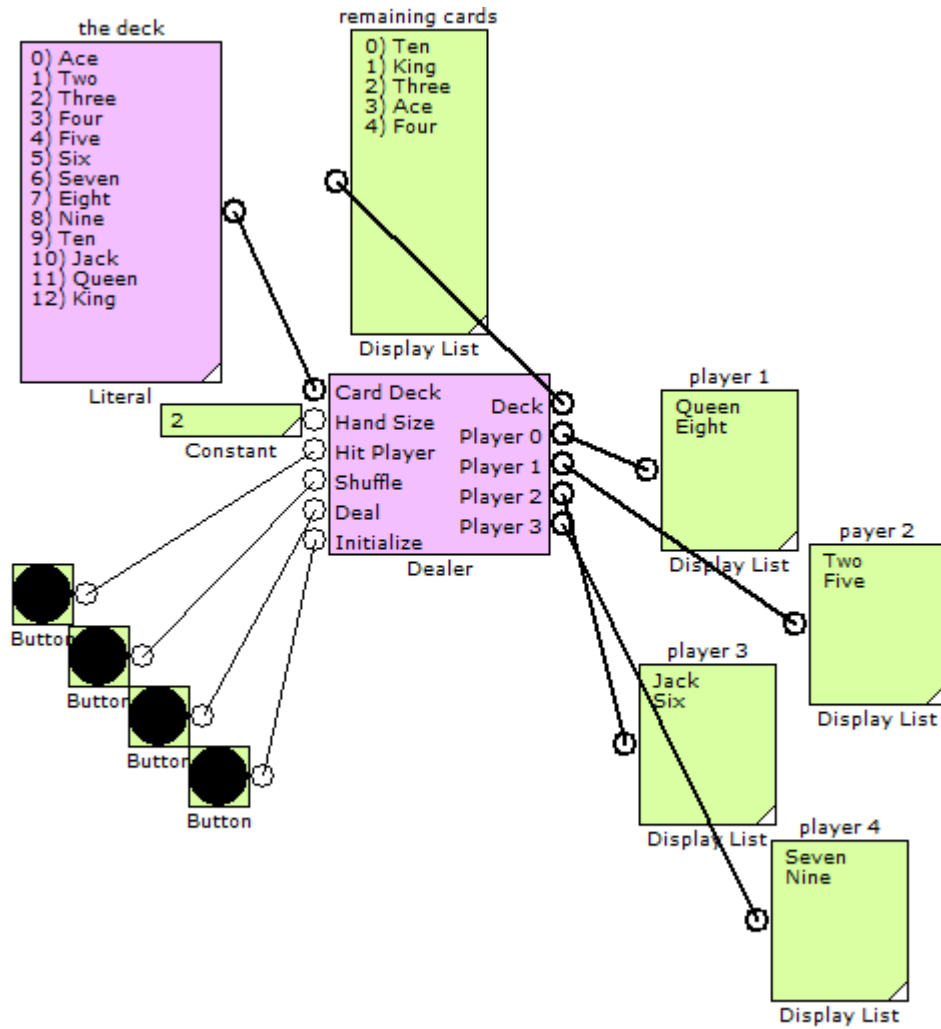
# Deal

The Deal component breaks the input into separate outputs. The outputs each receive one value at a time, like in a card game. You can set the number of outputs desired. Arrays - column 1



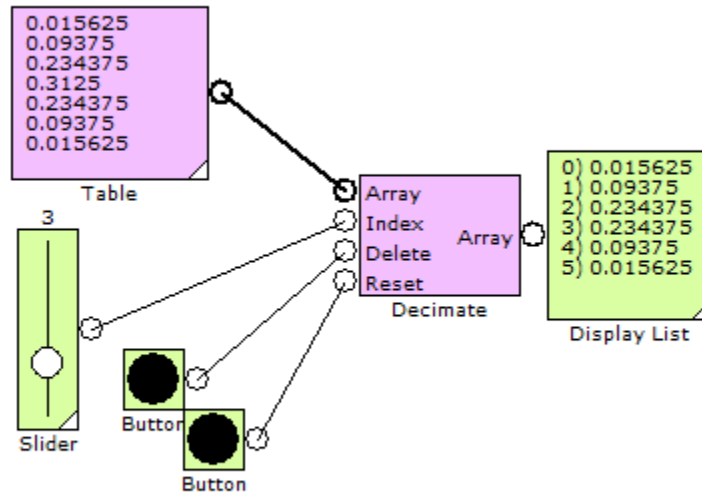
# Dealer

The Dealer component simulates a dealer in a card game. Arrays - column 6



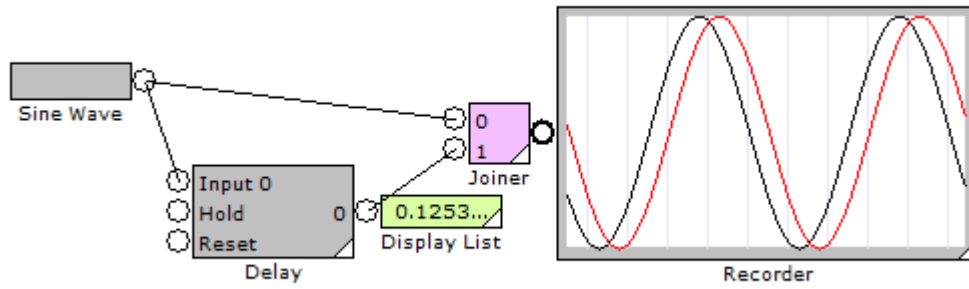
# Decimate

The Decimate component starts with a copy of the Array input. Each time the Delete input goes true an element of the array is removed from the output indicated by the Index input. The input Array is copied to the output when the Reset input goes true. Arrays - column 2



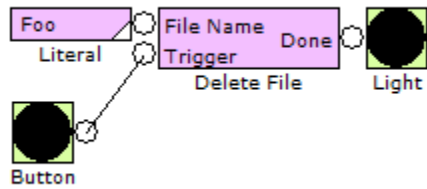
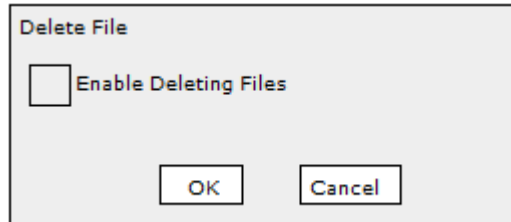
# Delay

The Delay component delays a signal for the number of clock ticks set in the Options dialog.  
Functions - column 3



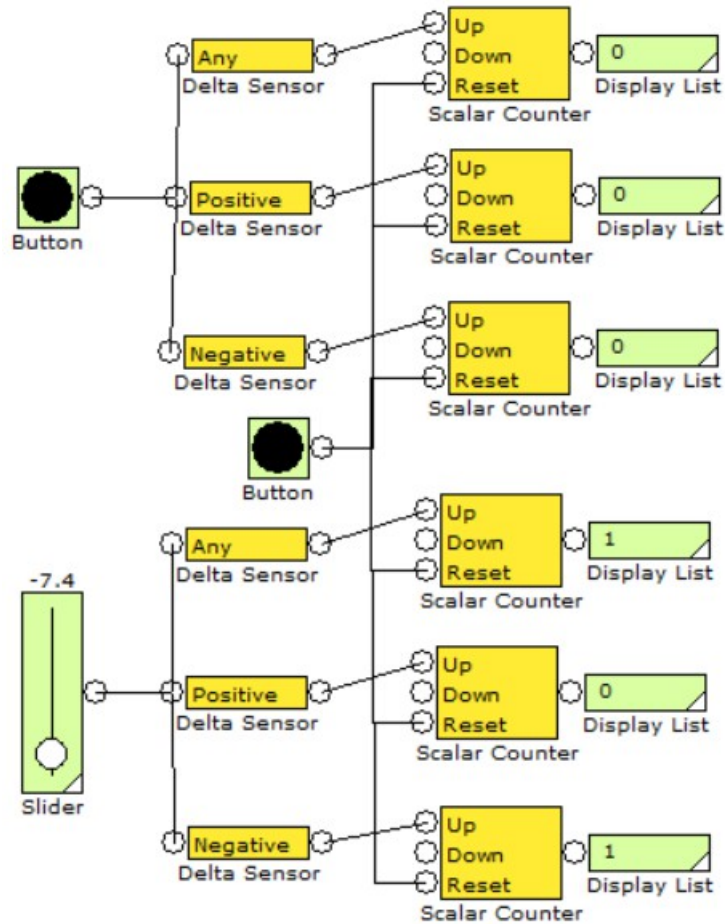
# Delete File

The Delete File component permanently deletes the specified file. BE CAREFUL! To insure that programs you load from outside sources cannot delete your files, you must enable the component each time it is loaded using the component's Options menu. Arrays - column 7



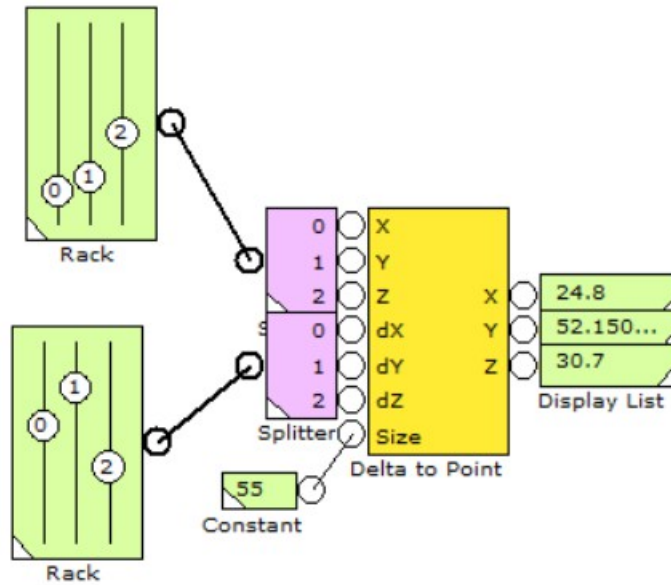
# Delta Sensor

The Delta Sensor component emits a short pulse when the input changes. You can accept all changes or limit it to Positive or Negative changes. Functions – column 7



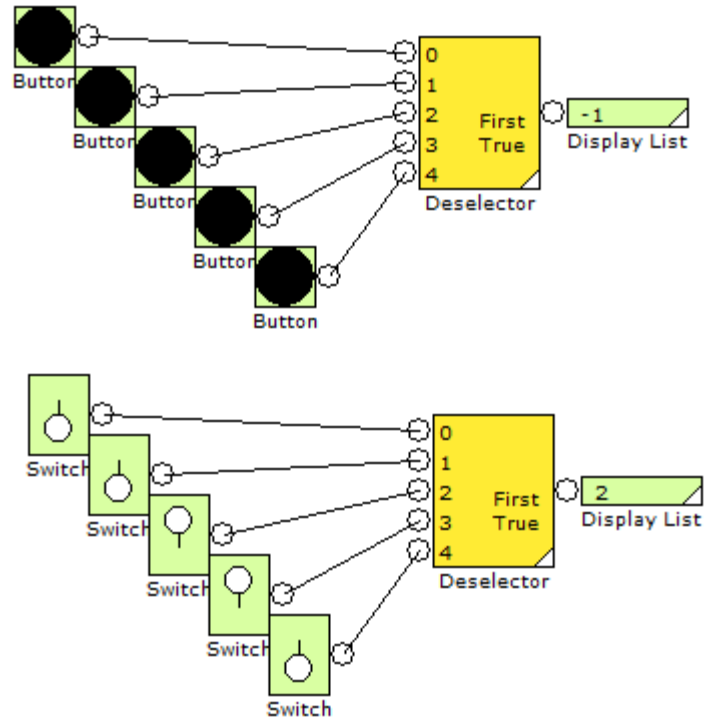
# Delta to Point

The Delta to Point component calculates a point from the start point, a delta vector and a size.  
Functions – column 8



# Deselector

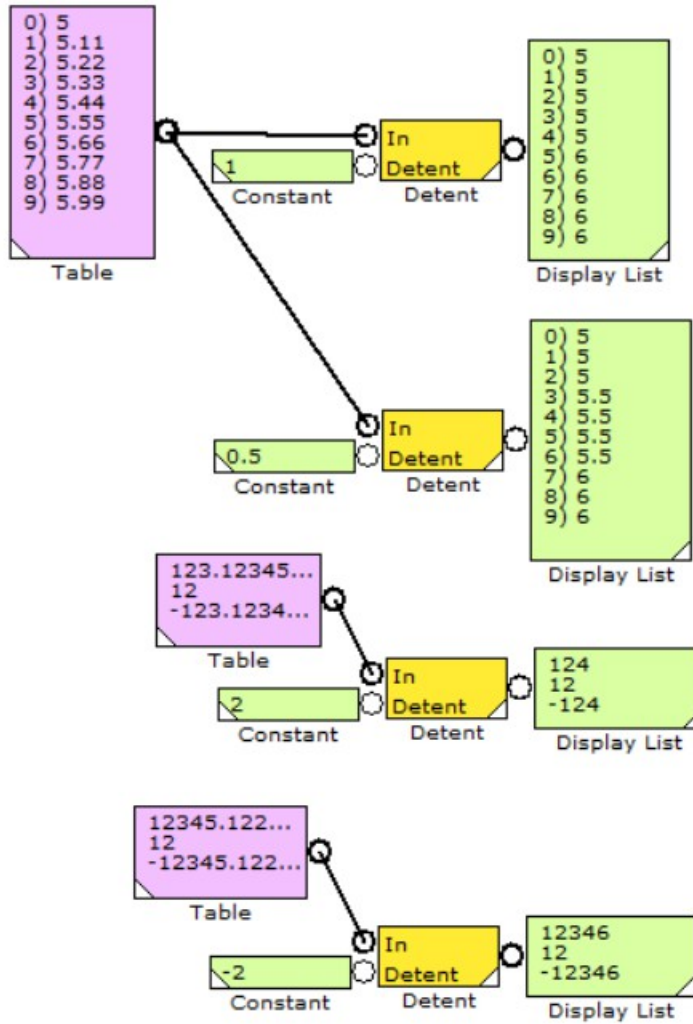
The Deselector component scans the inputs and outputs the number of the first true input. You can set the number of inputs. Functions - column 6





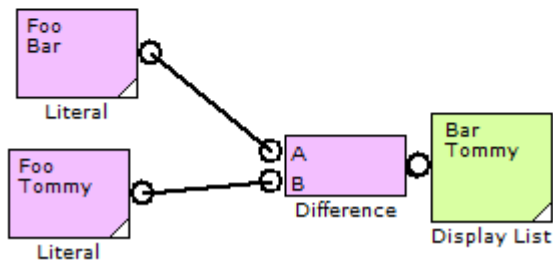
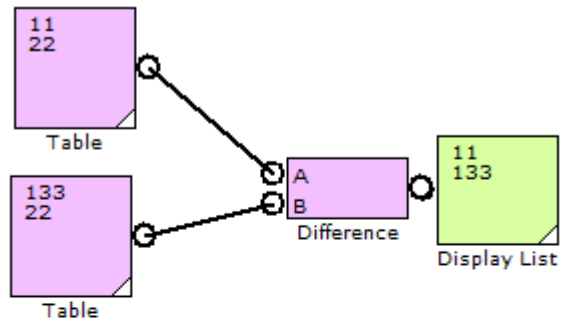
# Detent

The Detent component outputs a value with the specified detent. Inputs can be scalar or arrays.  
 Functions - column 2



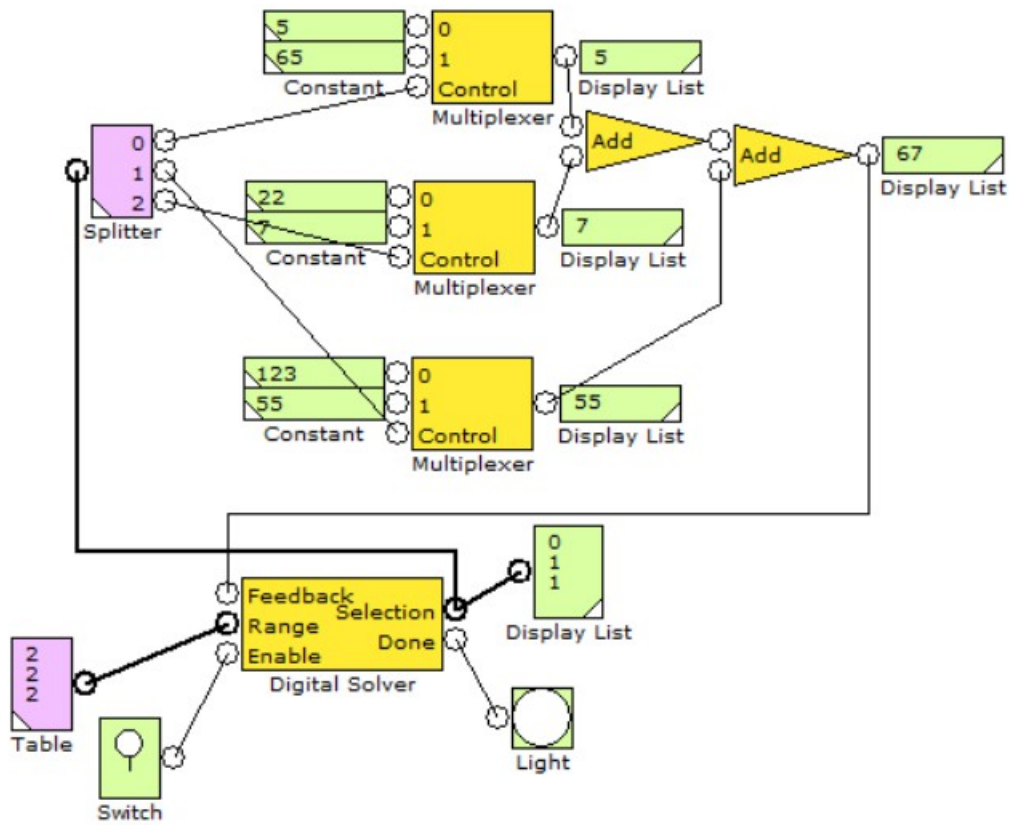
# Difference

The Difference component scans the input arrays and outputs the elements that are not present at both inputs. Arrays - column 5



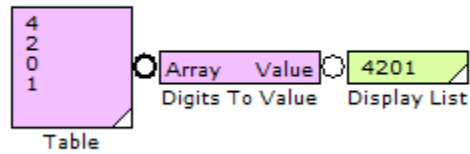
# Digital Solver

The Digital Solver component finds the selection that results in the minimum feedback. Functions – column 6



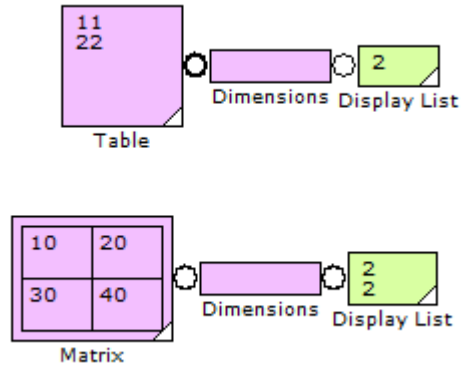
## Digits To Value

The Digits To Value component converts an array of digits into a value. -1 indicates a blank in the input. 10 indicates a negative sign. 11 indicates a plus sign.      Arrays - column 5



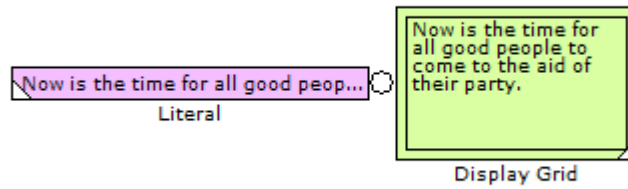
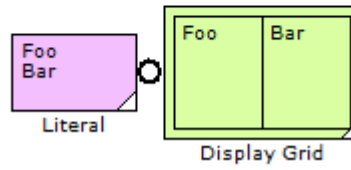
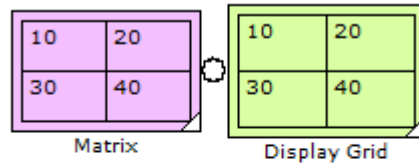
# Dimensions

The Dimensions component shows the dimensions for the input data. Use it to tell a scalar from a vector from a matrix. Arrays - column 2



# Display Grid

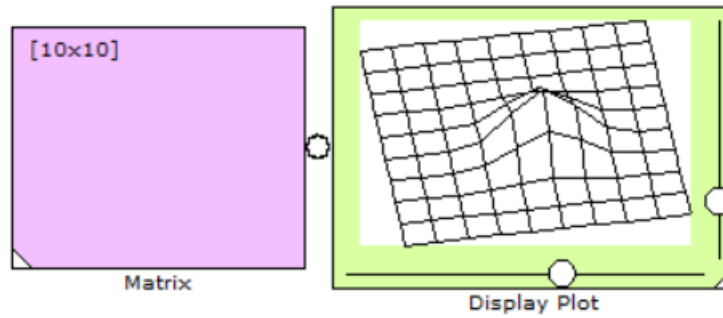
The Display Grid is used to show matrix values. Make it larger to see matrix values better.  
Controls - column 4





## Display Plot

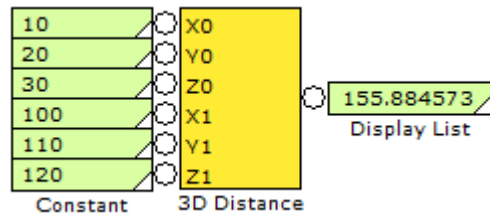
The Display Plot draws a graphic from a 2-dimensional data set. Rotate the display using the side and bottom controls. Controls – column 4





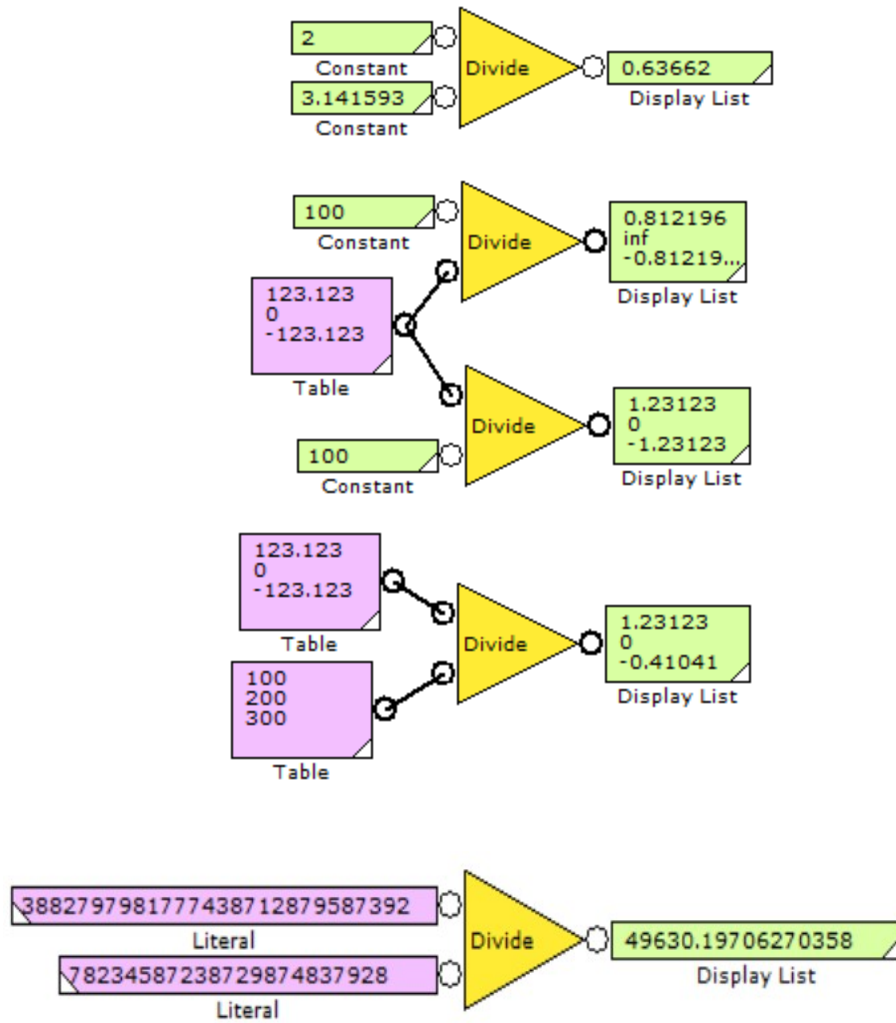
## 3D Distance

The 3D Distance component calculates the distance between two points in space. Functions - column 8



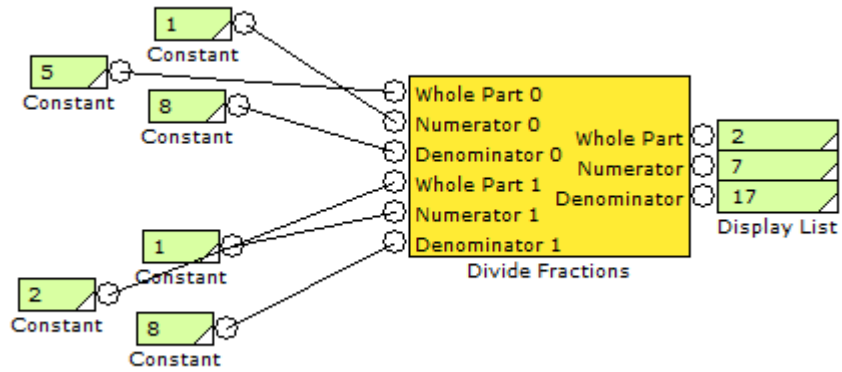
# Divide

The Divide component divides the inputs. Inputs can be scalar and /or arrays. Functions - column 1



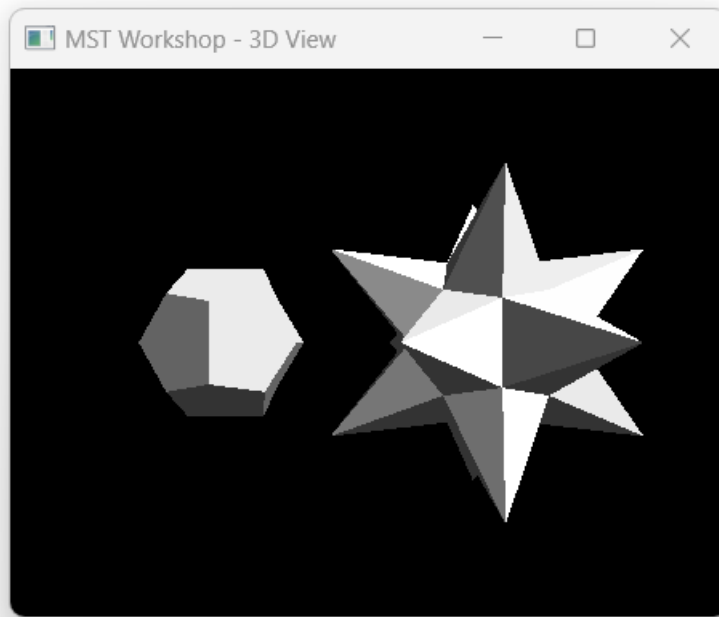
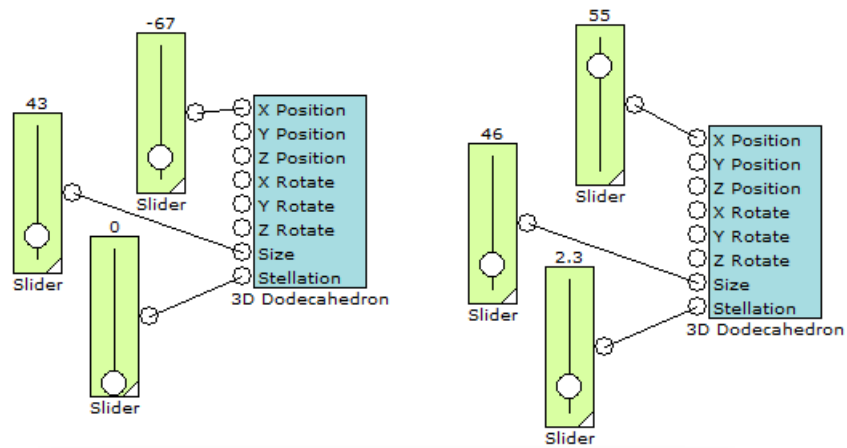
# Divide Fractions

The Divide Fractions component divides two numbers with fractions. The output is a whole number with a fraction as needed. Functions - column 9



### 3D Dodecahedron

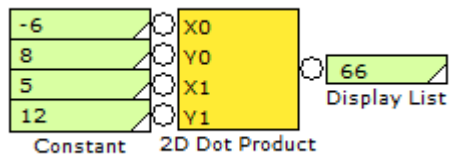
The 3D Dodecahedron draws a dodecahedron in the 3D view window. The X, Y, and Z inputs set the position and size of the dodecahedron. The Affine [4x4] array input modifies the size, position, and orientation of the dodecahedron. 3D – column 4



## 2D Dot Product

The 2D Dot Product component calculates the dot product of 2 2D vectors.

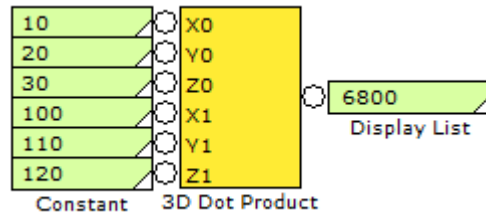
Functions - column 8



## 3D Dot Product

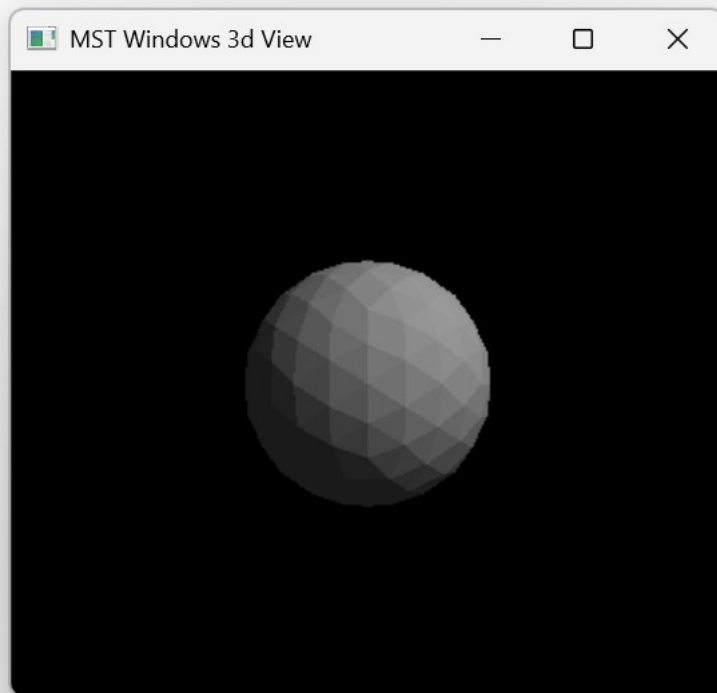
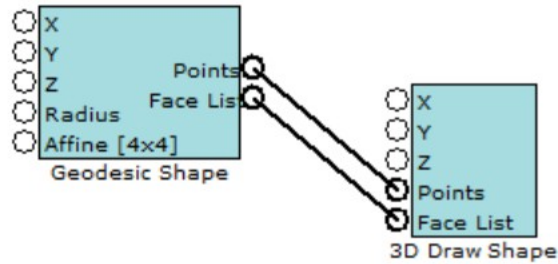
The 3D Dot Product component calculates the dot product of 2 3D vectors.

Functions - column 8



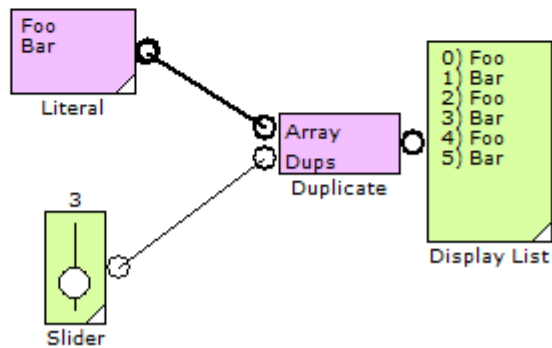
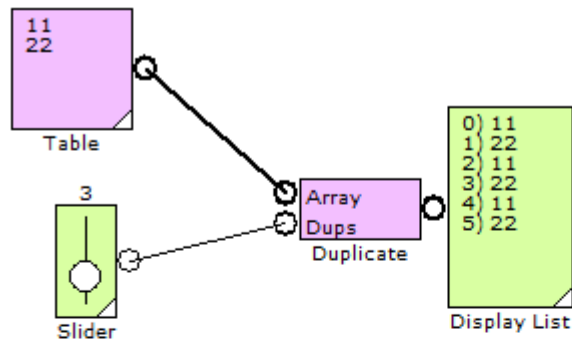
## 3D Draw Shape

The 3D Draw Shape component draws a shape composed of triangular meshes in the 3D view window. The X, Y, and Z-Points establish points in 3-dimensions. The list of faces is a list of 3 points for each triangle. For example if you specify 4 3D coordinates to create a triangular pyramid, you would create the face by specifying 0, 1, 2 for the first face. Then 0, 3, 1 for the second face, then 1, 3, 2 for the third face and 0, 2,3 for the bottom. Also the order of the points in the list indicates the normal direction (right hand rule). The Affine [4x4] array input modifies the size, position, and orientation. 3D - column 6



# Duplicate

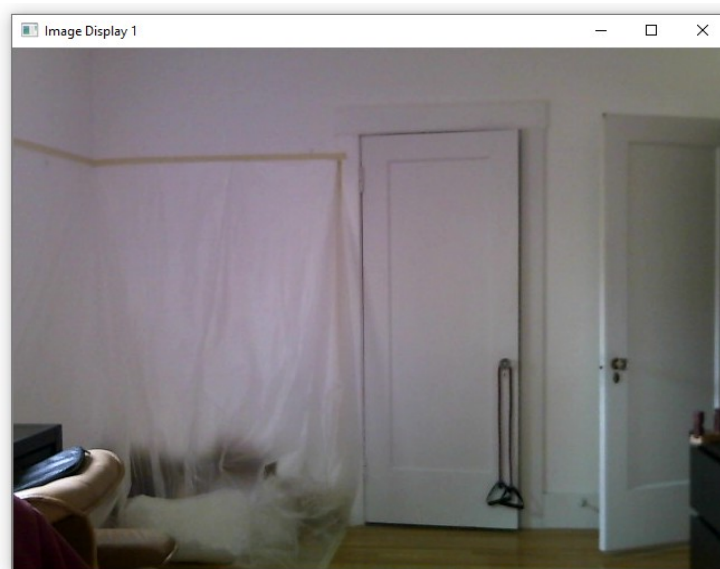
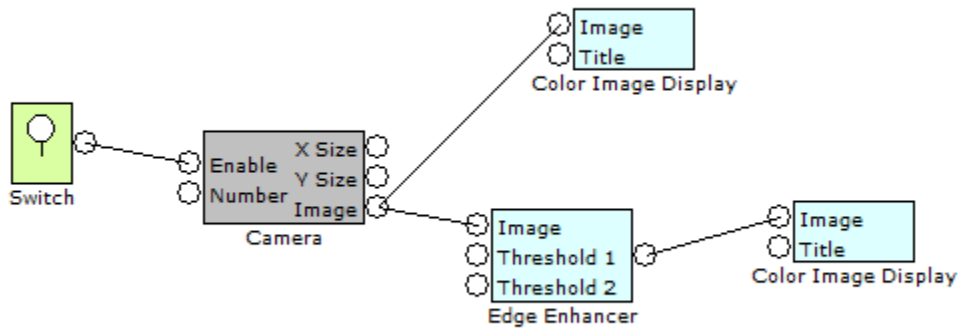
The Duplicate component builds an array by duplicating each element at the Array input. The Array input can be a scalar or an array. For example: an input of 1, 2, 3 with a Dups input of 2 will output 1, 2, 3, 1, 2, 3. Arrays - column 2





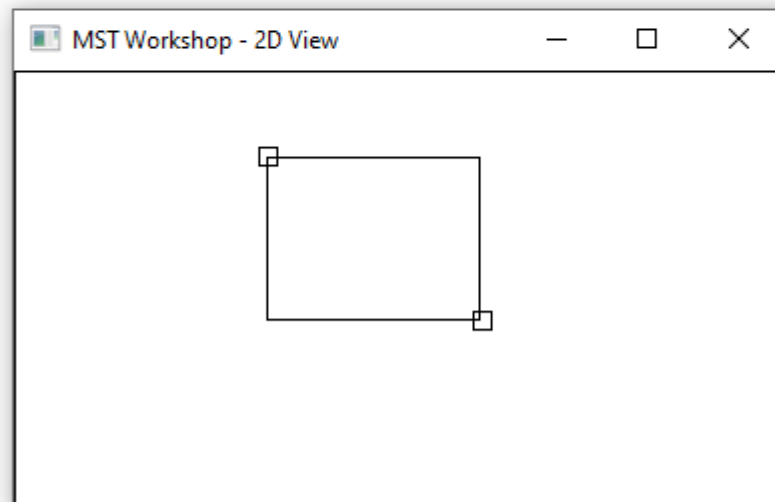
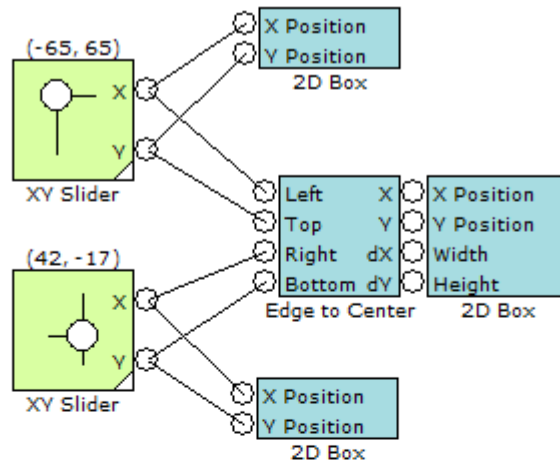
# Edge Enhancer

The Edge Enhancer component creates lines based on the gradient of the image. Images - column 2



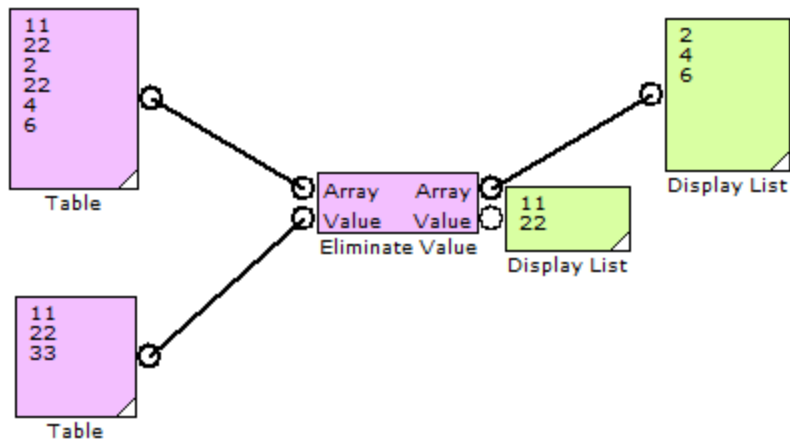
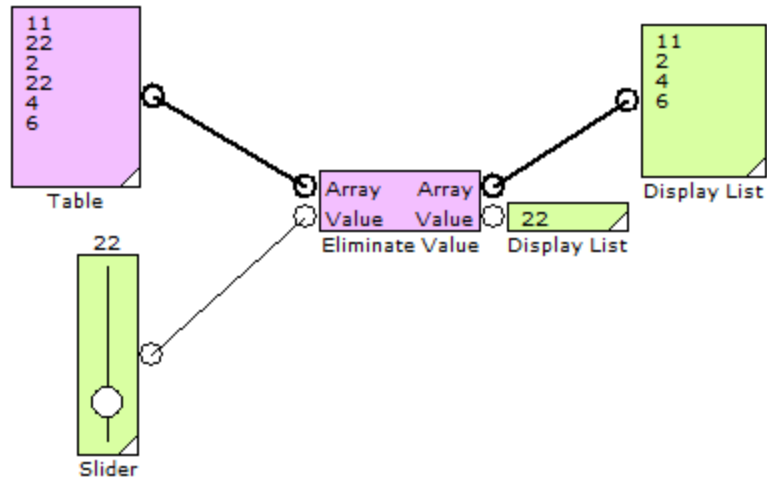
## Edge to Center

The Edge to Center component takes bounding coordinates and outputs the center and size values that could be used to draw an object there. 2D - column 8



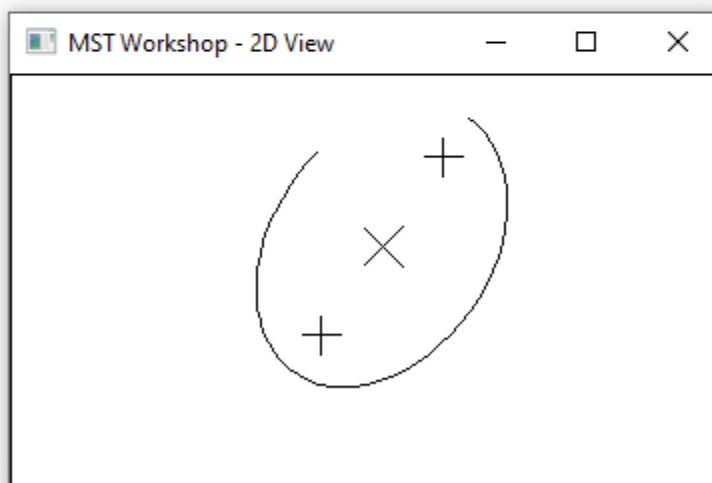
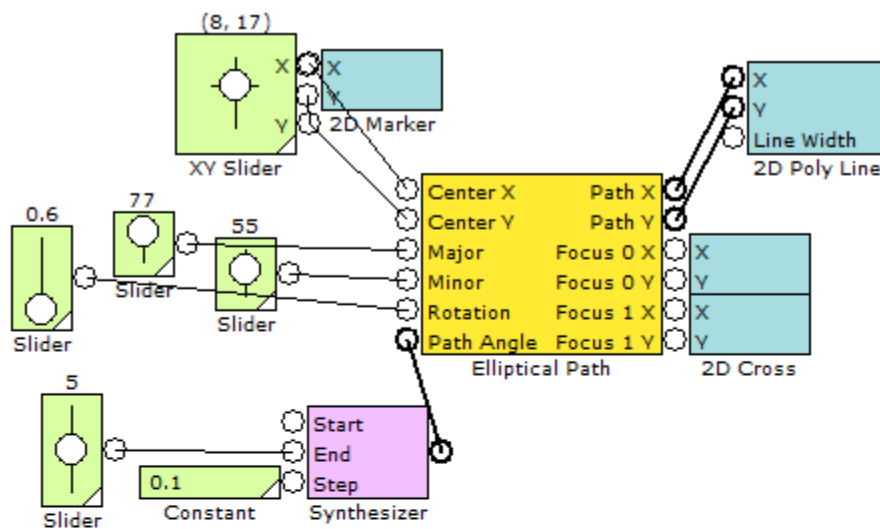
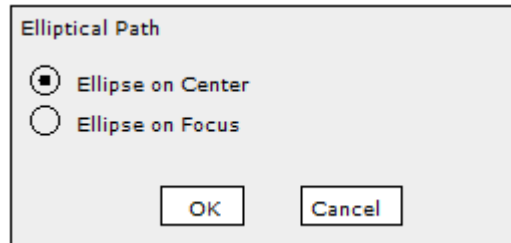
# Eliminate Value

The Eliminate Value component removes elements found on the Value input. Arrays - column 5



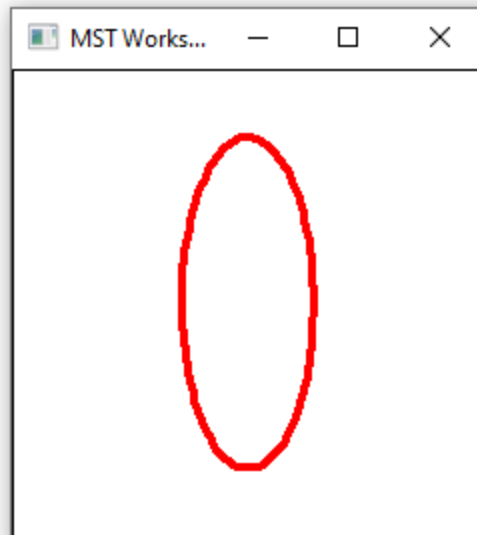
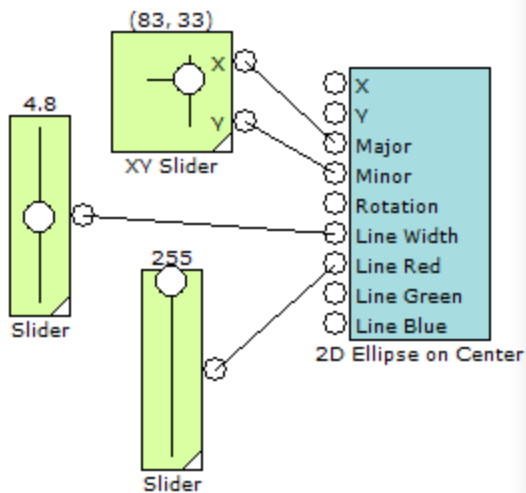
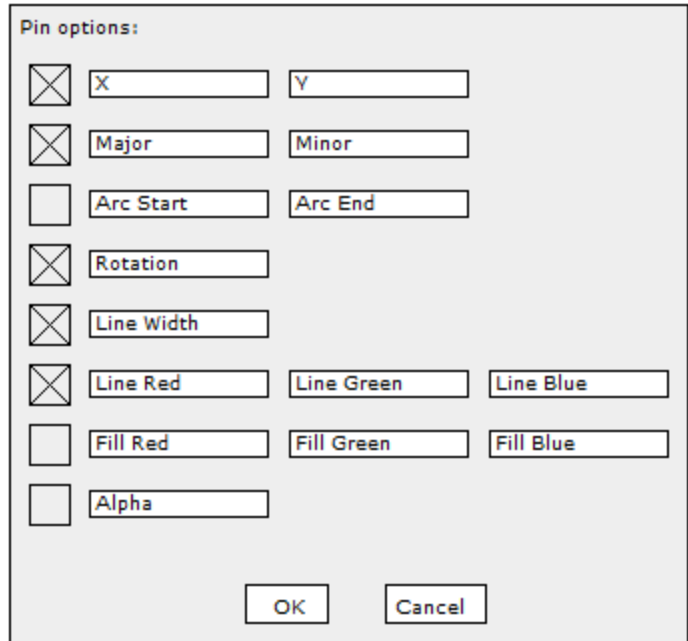
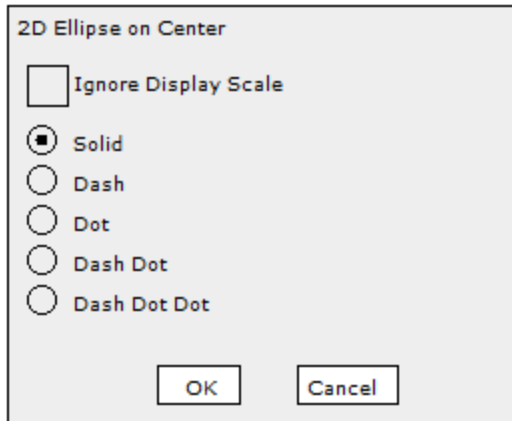
# Elliptical Path

The Elliptical Path component traces the ellipse (X, Y) given the inputs and a range of angles.  
 Functions - column 9



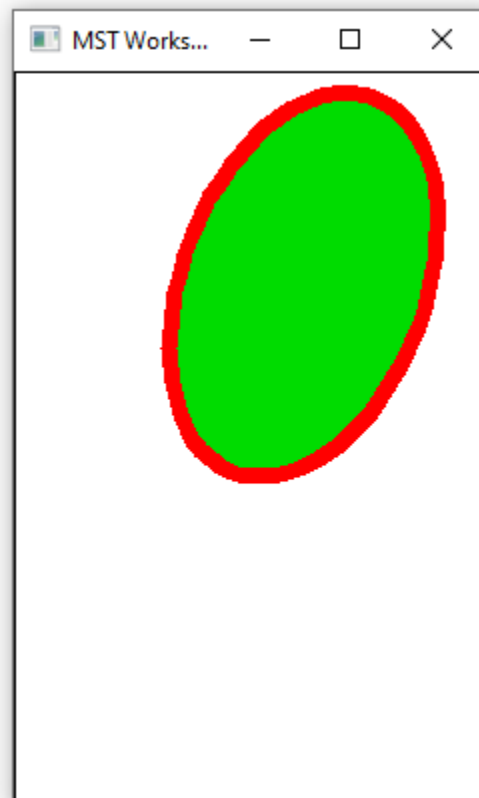
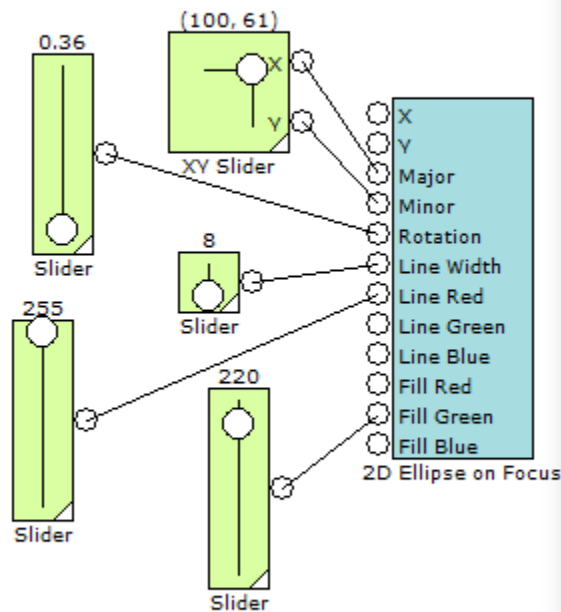
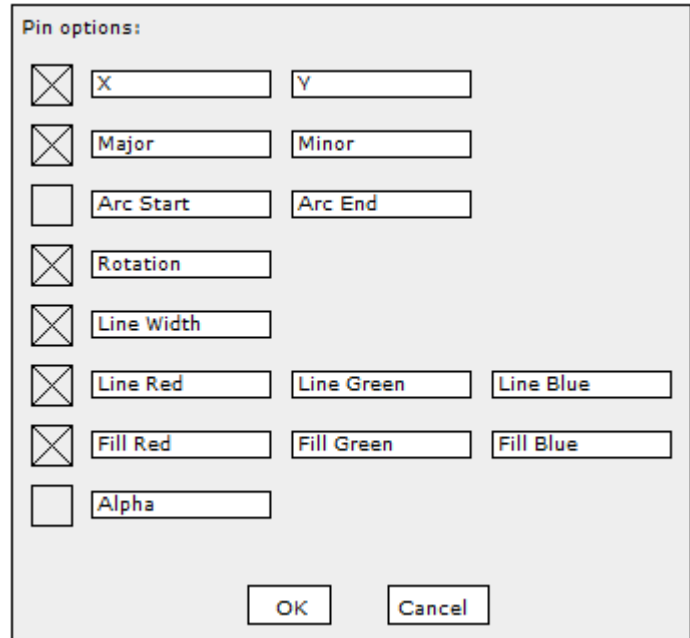
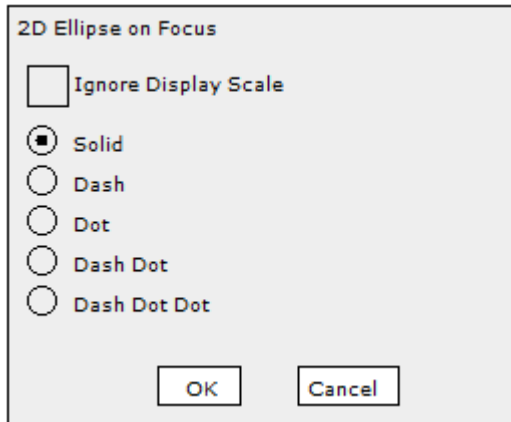
## 2D Ellipse on Center

The 2D Ellipse on Center component draws a ellipse outline using the X and Y points given. You can control the line thickness. 2D - column 1



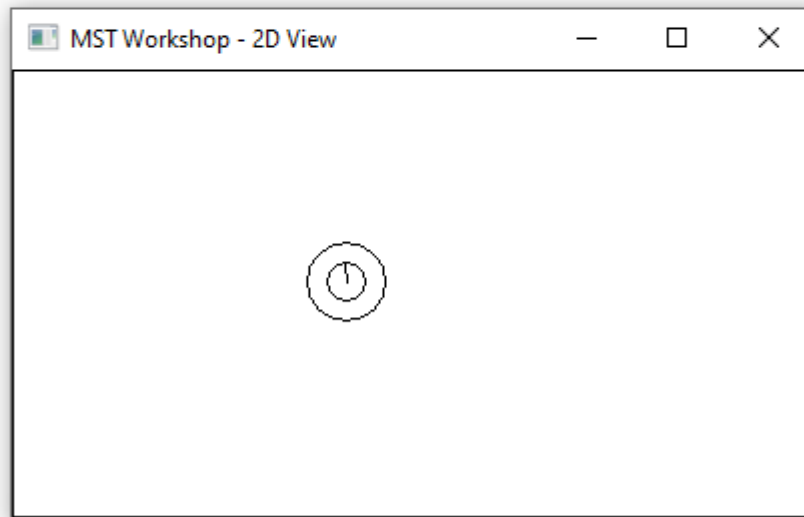
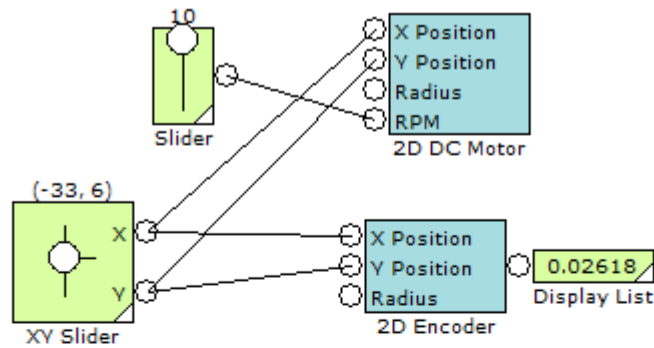
## 2D Ellipse on Focus

The 2D Ellipse on Focus component draws a ellipse outline using the X and Y points given. You can control the line thickness. 2D - column 1



## 2D Encoder

The Rotation Encoder outputs the angle of the shaft. You can attach it to Gears or other simulation objects like Motors. Place the encoder at the same X, Y position as a gear to connect them. 2D - column 3



# 3D Encoder

The 3D Encoder component reads the position and rotational speed of a motor or shaft. Place the encoder at the X, Y, Z coordinates of the motor or end of the shaft. 3D - column 4

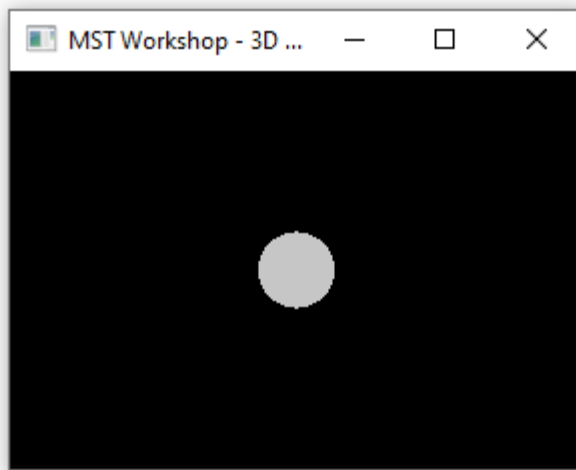
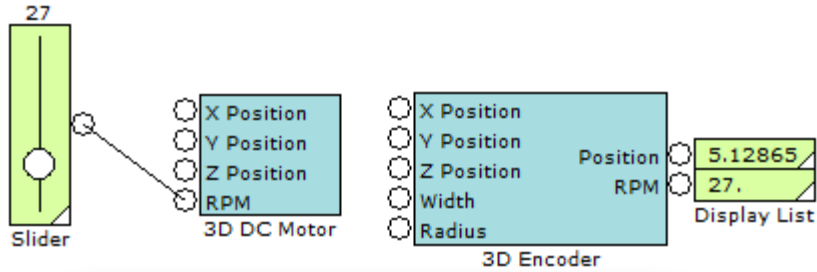
3D Encoder

Width Steps:

Radius Steps:

Pin options:

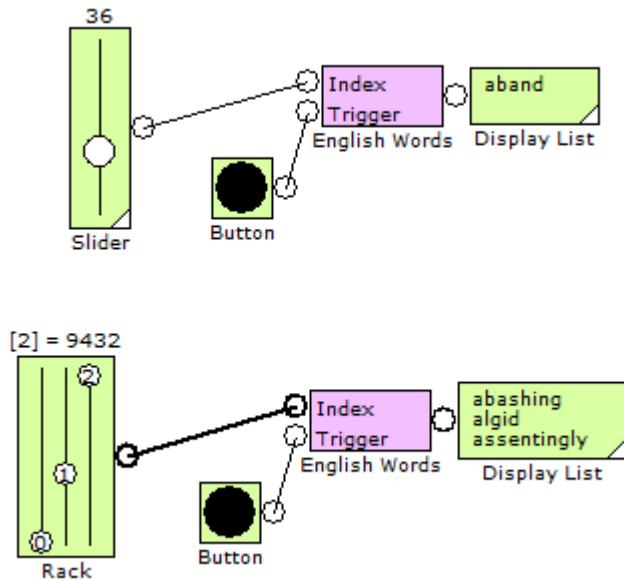
<input checked="" type="checkbox"/>	<input type="text" value="X Position"/>	<input type="text" value="Y Position"/>	<input type="text" value="Z Position"/>
<input type="checkbox"/>	<input type="text" value="X Rotate"/>	<input type="text" value="Y Rotate"/>	<input type="text" value="Z Rotate"/>
<input checked="" type="checkbox"/>	<input type="text" value="Width"/>	<input type="text" value="Radius"/>	
<input type="checkbox"/>	<input type="text" value="Red"/>	<input type="text" value="Green"/>	<input type="text" value="Blue"/>
<input type="checkbox"/>	<input type="text" value="Alpha"/>		
<input type="checkbox"/>	<input type="text" value="[4x4]"/>		





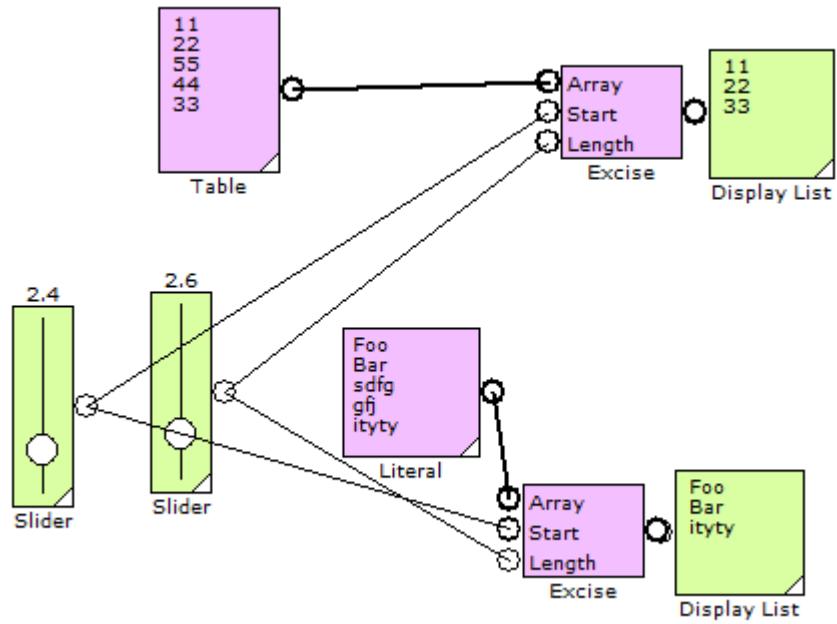
# English Words

The English Words component is a list of 58,000 English words downloaded from the Mieliestronk website.



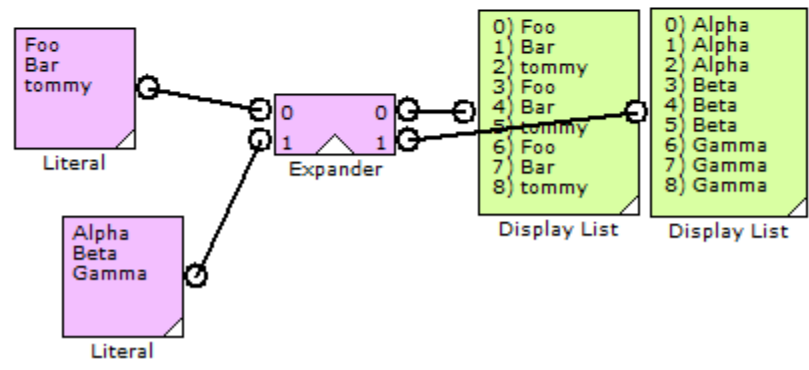
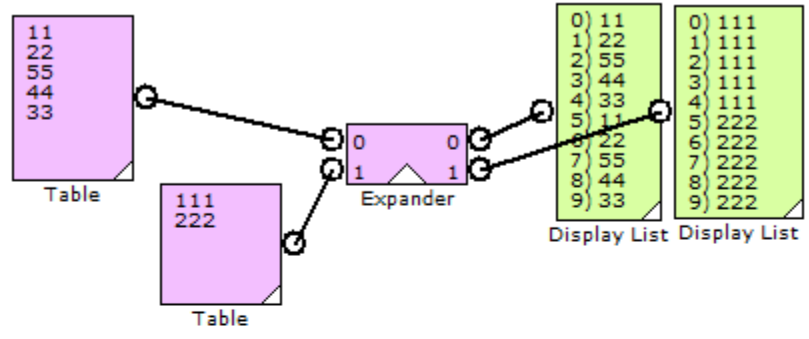
# Excise

The Excise component eliminates a portion of the values in an array. Input the number of the first value and the number of values you want to remove.      Arrays - column 1



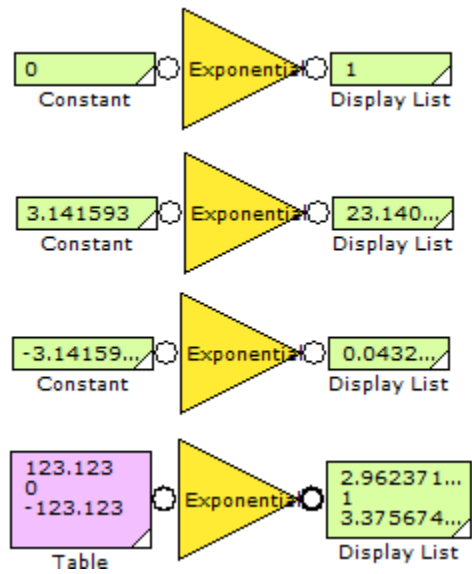
# Expander

The Expander component outputs all the combinations for all its inputs. You can set the number of inputs/outputs. Arrays - column 1



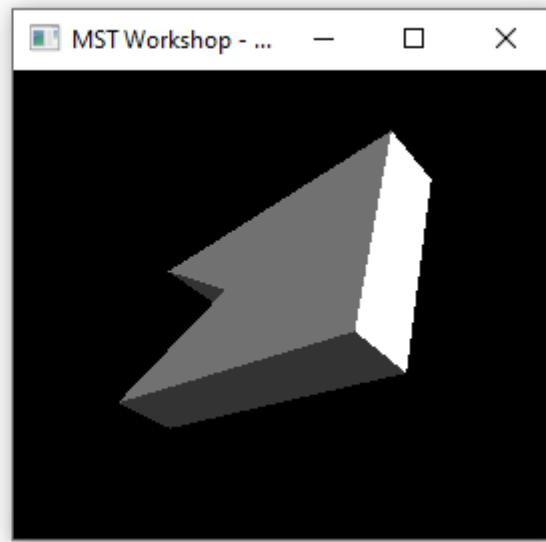
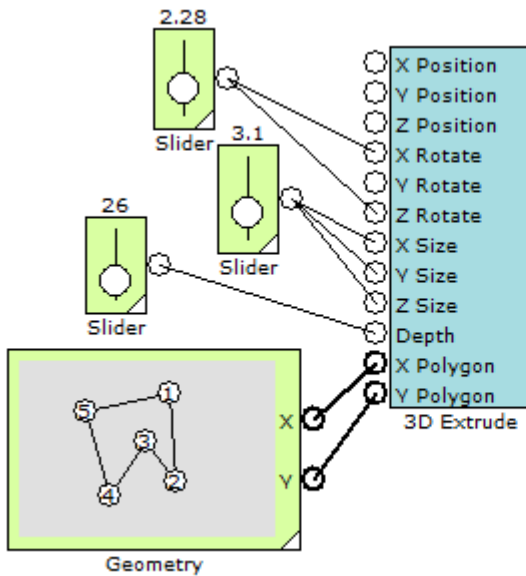
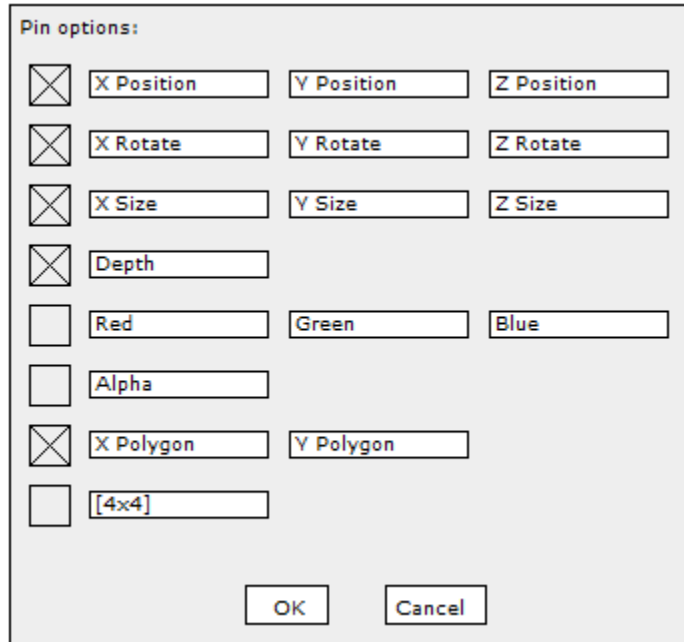
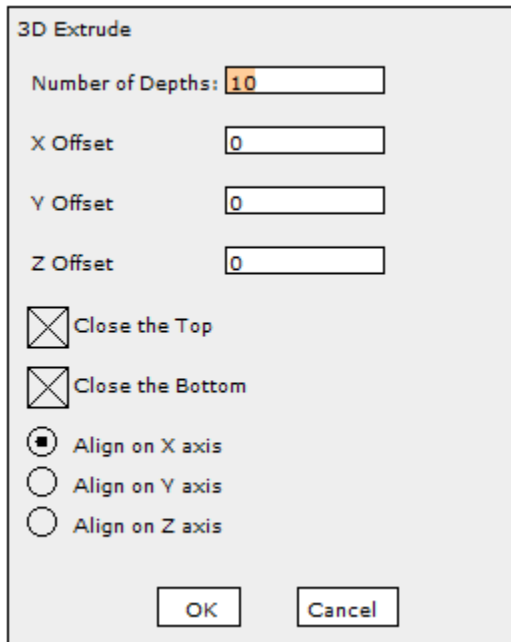
# Exponential

The Exponential component calculates the exponential of the input base  $e$ . Input can be scalar or an array. Functions - column 1



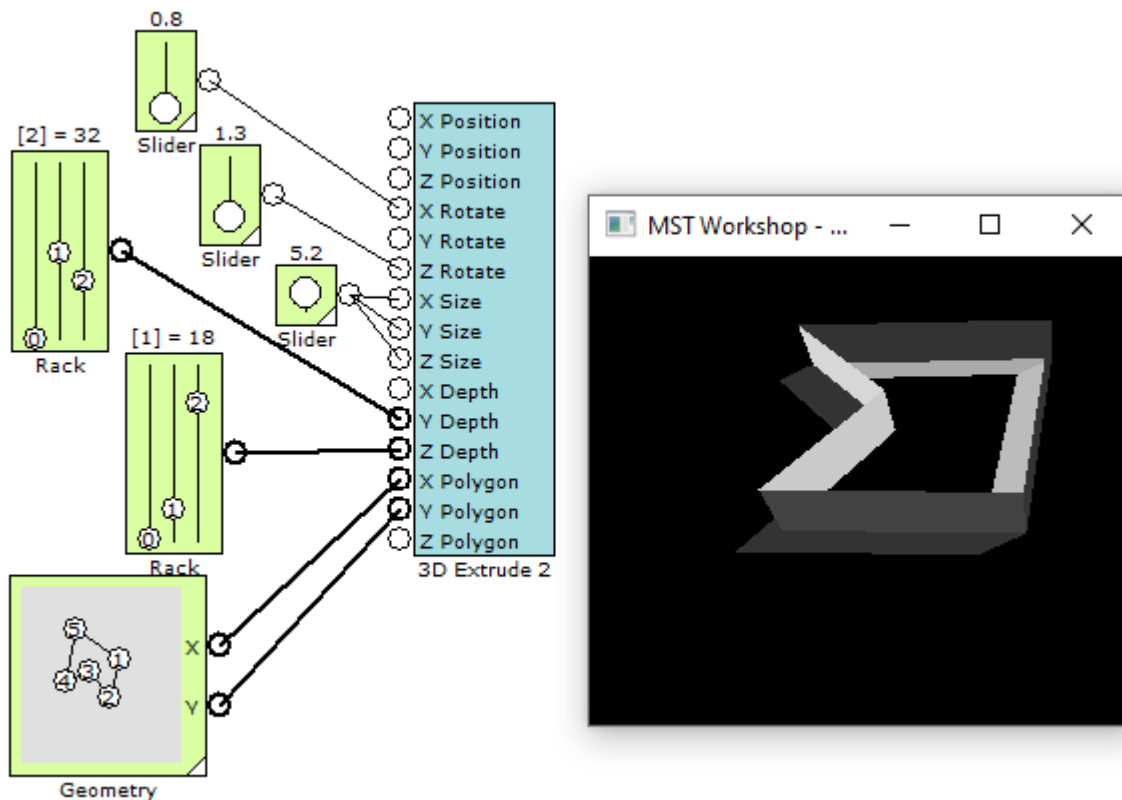
# 3D Extrude

The 3D Extrude draws a 2D shape extruded into the 3D view window. The X, Y, and Z inputs set the position and size of the extrude. The Affine [4x4] array input modifies the size, position and orientation. 3D - column 5



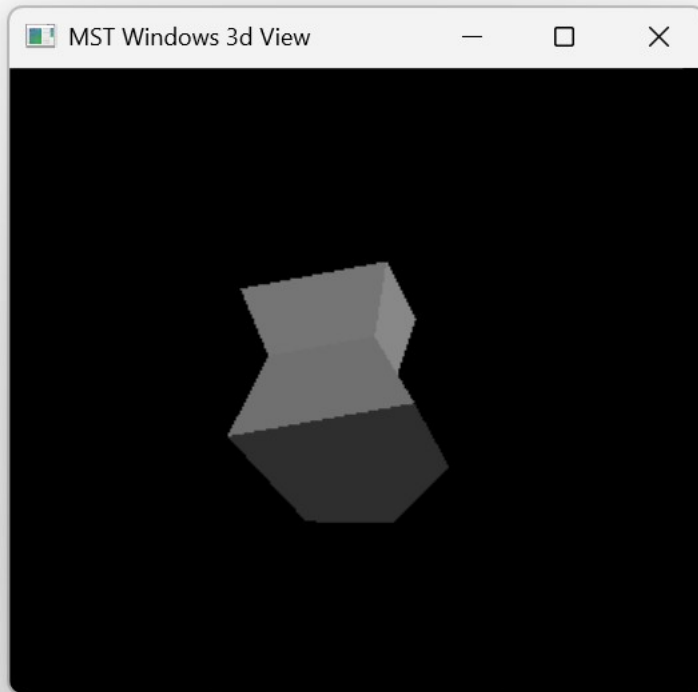
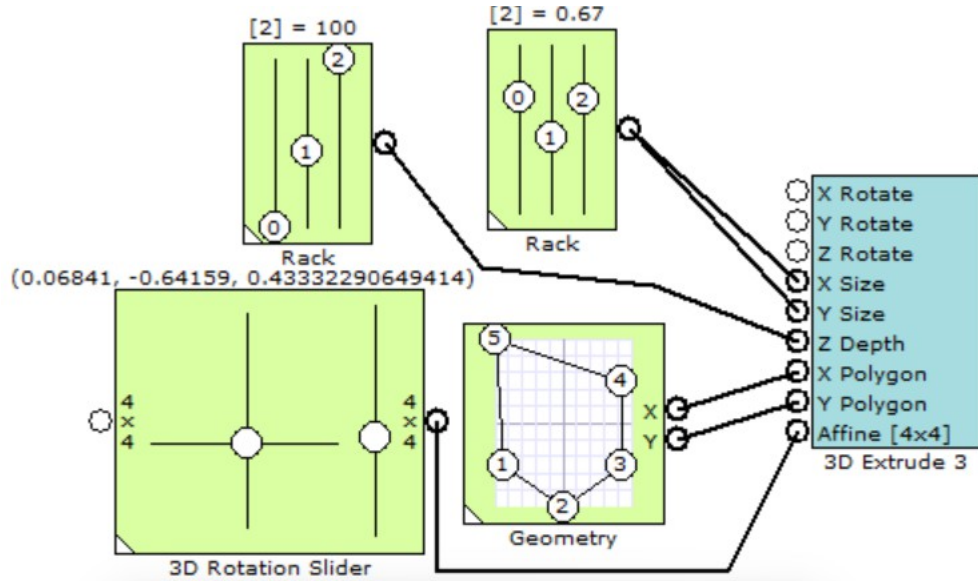
## 3D Extrude 2

The 3D Extrude 2 draws a 2D shape extruded into the 3D view window. The X, Y, and Z inputs set the position and size of the extrude. The Affine [4x4] array input modifies the size, position and orientation. 3D - column 5



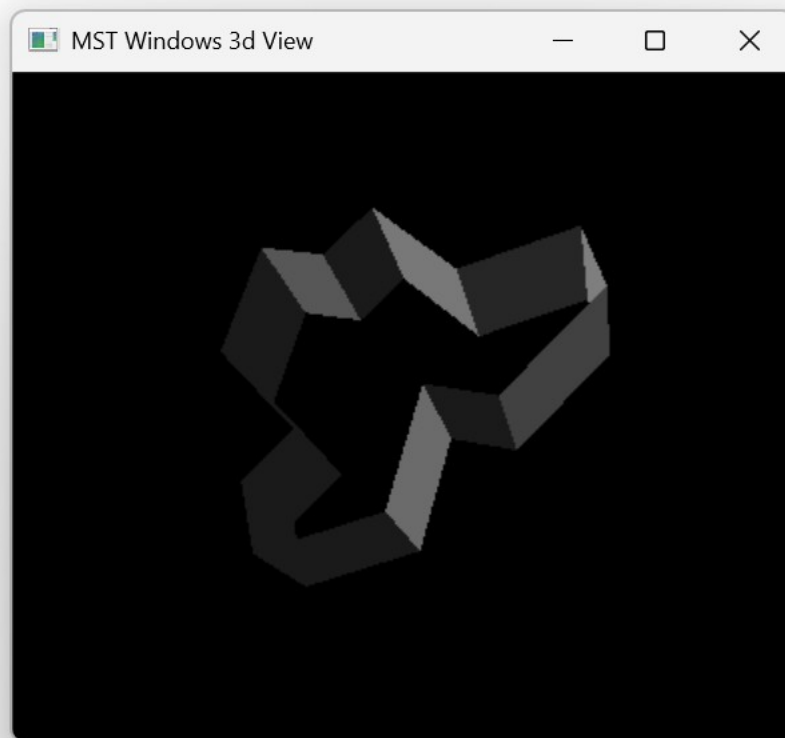
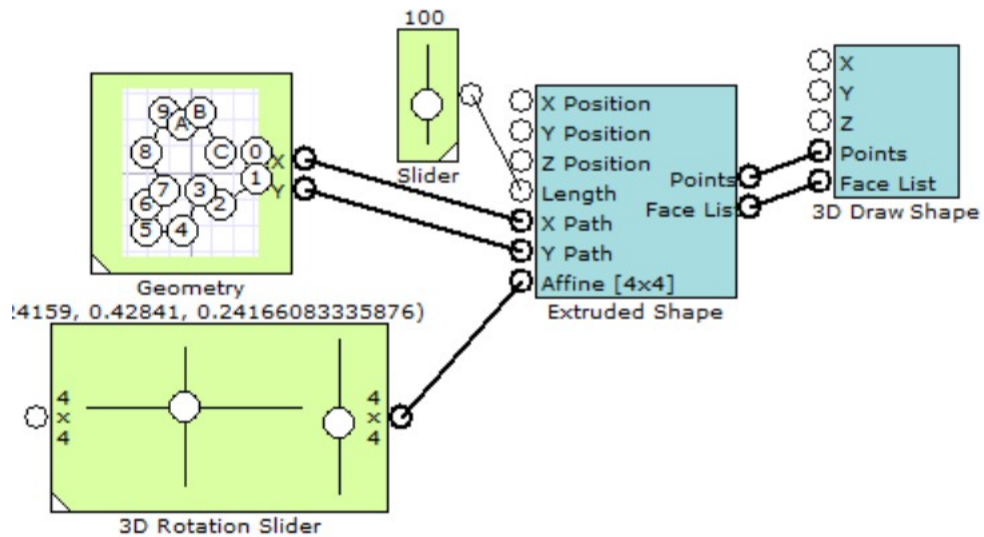
### 3D Extrude 3

The 3D Extrude 3 draws a 2D shape extruded into the 3D view window. The X, Y, and Z inputs set the position and size of the extrude. The Affine [4x4] array input modifies the size, position and orientation. 3D - column 5



## Extruded Shape

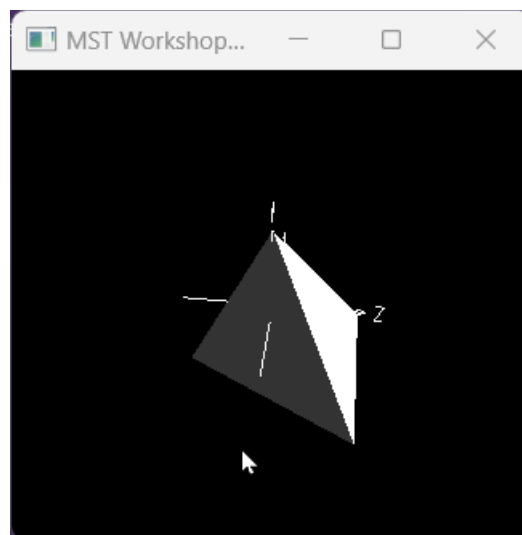
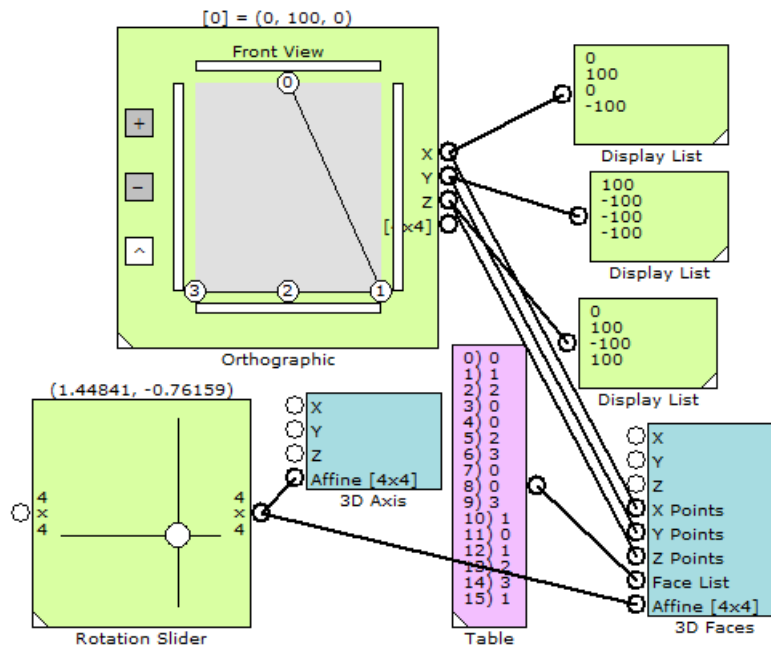
The Extruded Shape component creates triangular mesh composed of points and faces from an extrusion. You input an X, Y array to define the outline of the extrusion and its length. Use its outputs to reshape or draw it in the 3D View by connecting it to the 3D Draw Shape component. 3D - column 6

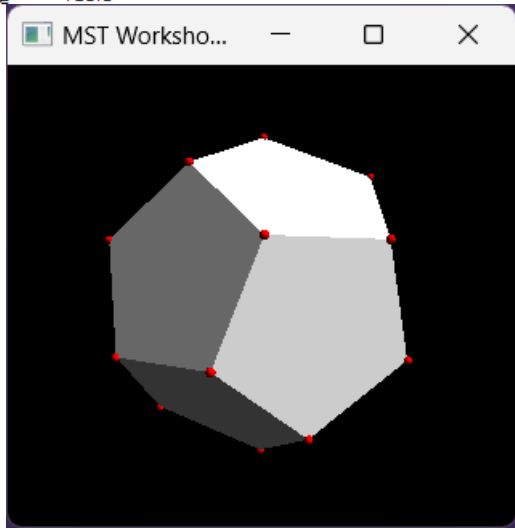
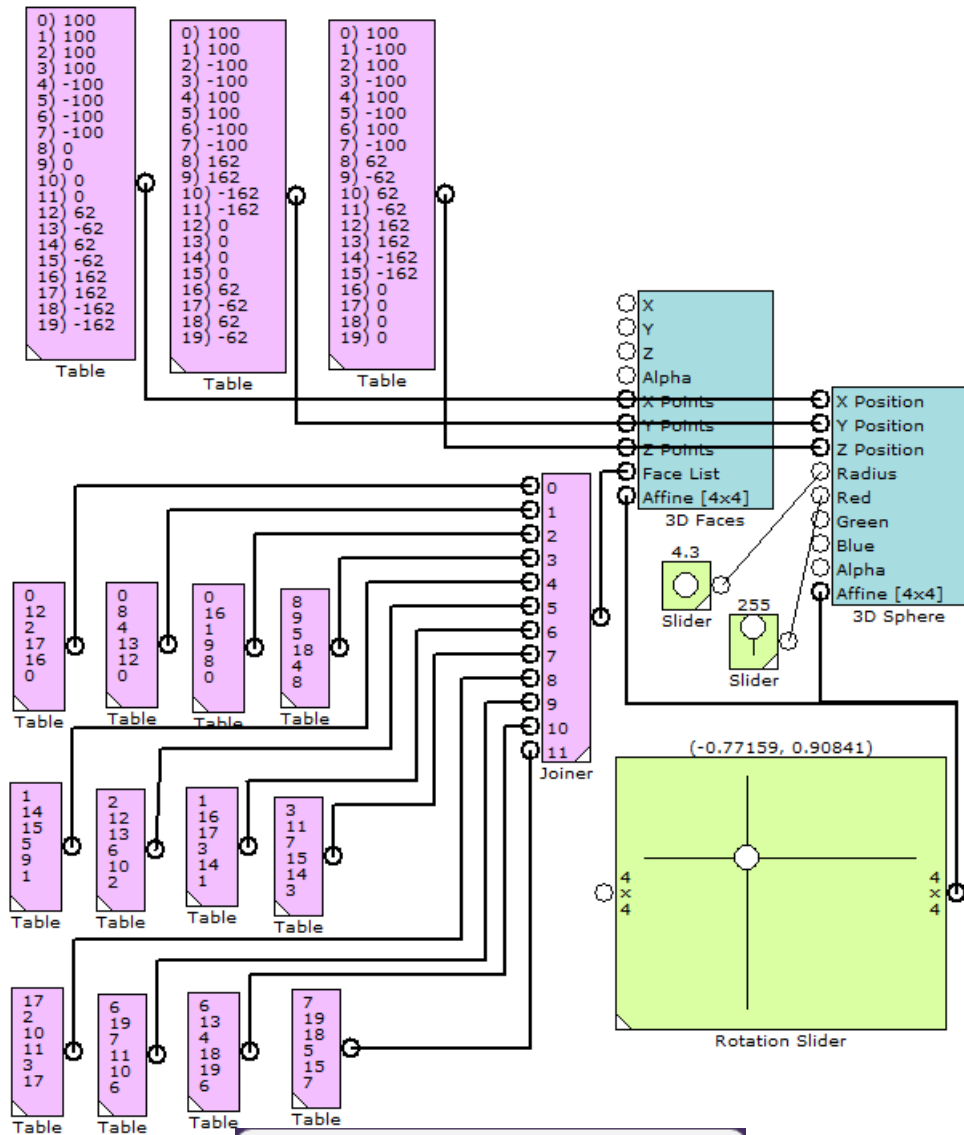




## 3D Faces

The 3D Faces component draws a shape in the 3D view window. The X-, Y-, and Z-Points establish points in 3-dimensions. Then create a list of faces by listing the point for each face. For example if you specify 4 3D coordinates to create a tetrahedron, you would create the first face by specifying 0, 1, 2, 0 for the first face. Then 0, 2, 3, 0 for the second face, then 0, 3, 1, 0 for the third face and 1, 3, 2, 1 for the bottom. Be sure to start and end each face with the same point number. Also the order of the points in the list indicates the normal direction (right hand rule). The Affine [4x4] array input modifies the size, position, and orientation. 3D - column 5

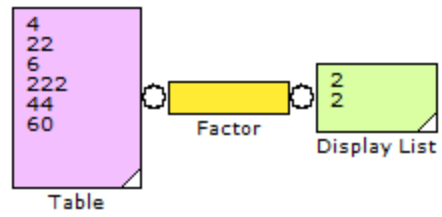




# Factor

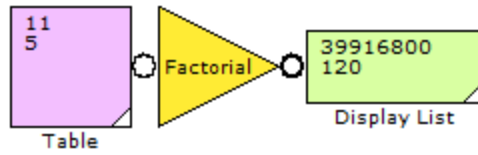
The Factor component returns a list of the prime factors of the input.

Functions - column 9



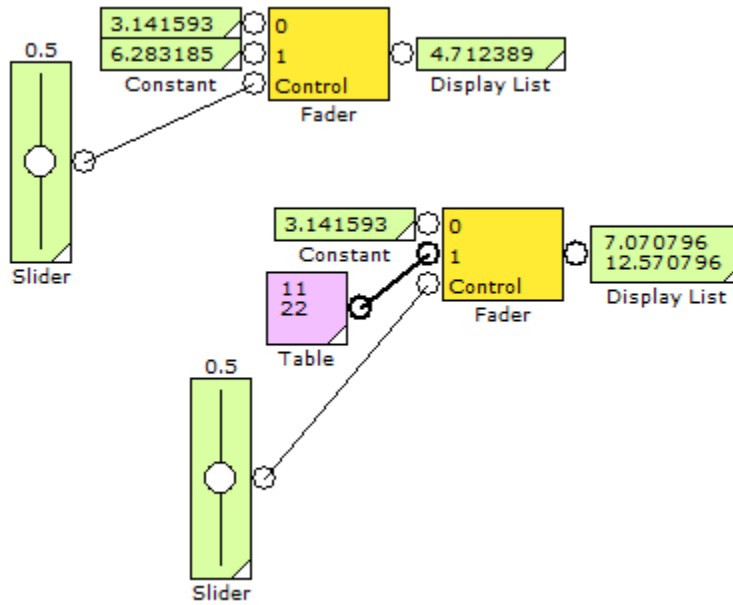
# Factorial

The Factorial component calculates the factorial of a number. Functions - column 5



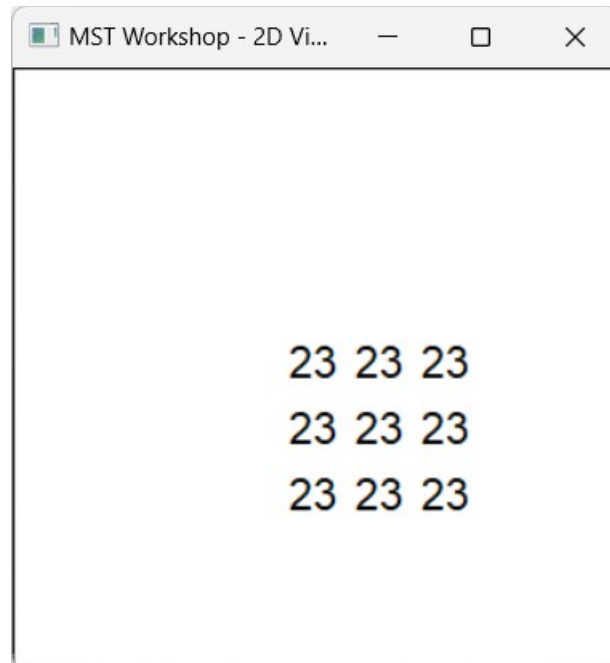
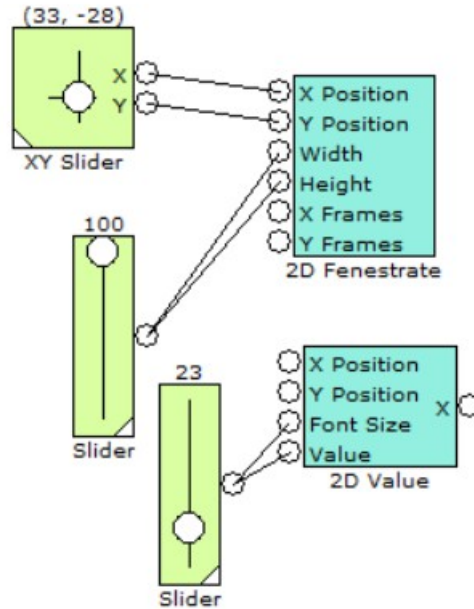
# Fader

The Fader can fade between inputs based on the Control input going from zero to one. Functions - column 6



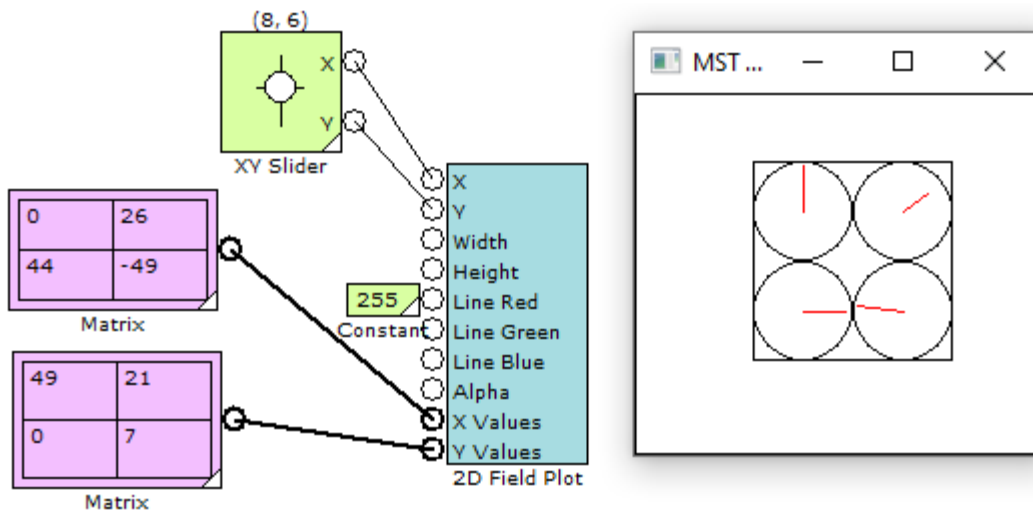
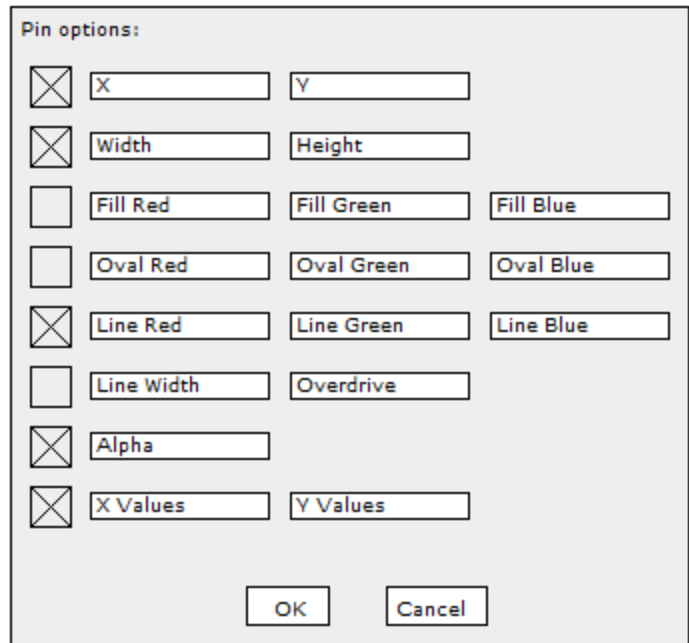
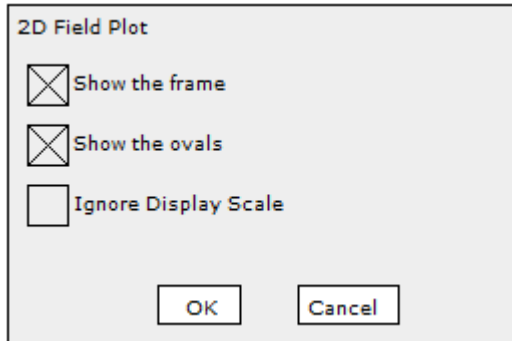
## 2D Fenestrate

The 2D Fenestrate creates the repeating windows effect on the image in the 2D drawing it covers. Note: This component only affects the items in the 2D View window that are under it. 2D – column 7



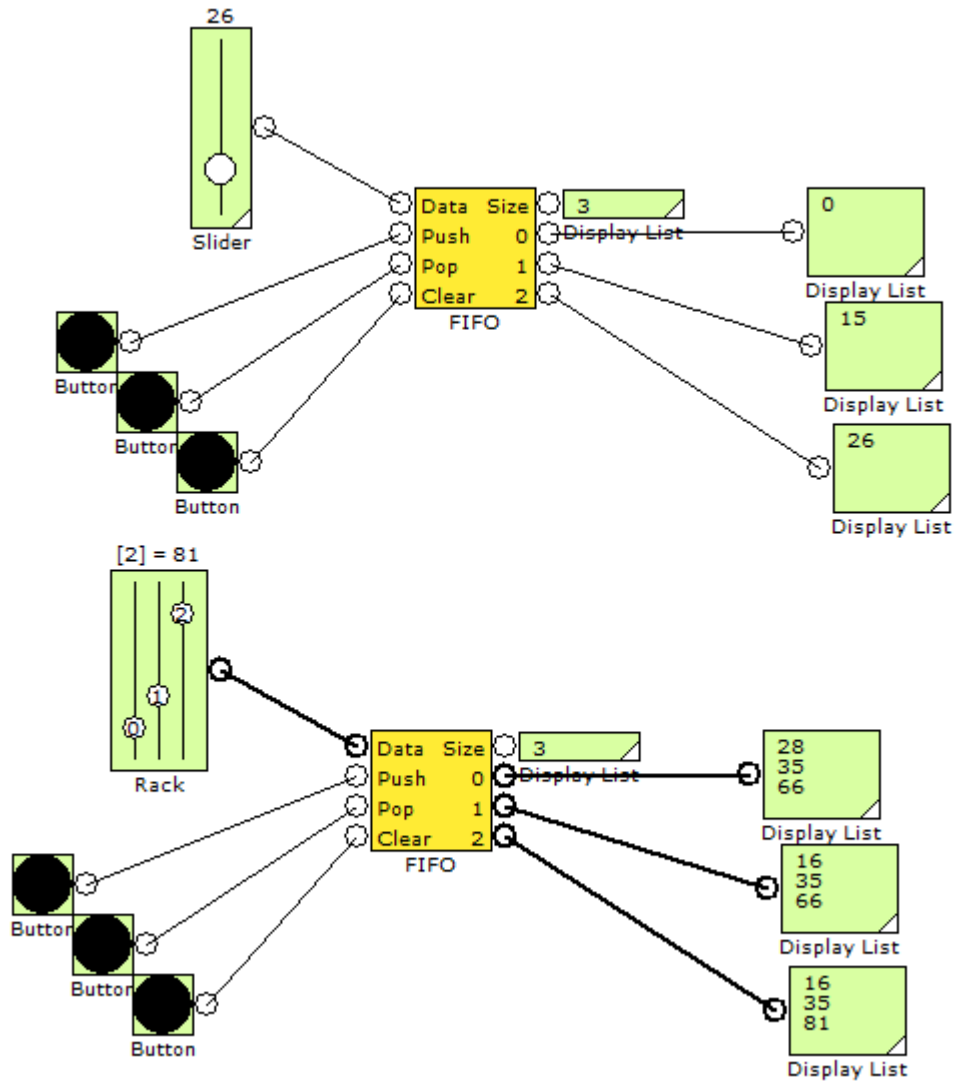
## 2D Field Plot

The 2D Field Plot plots 2D data creating a field plot of the data in the 2D display window. The inputs to the X Values and Y Values must be two 2 dimensional arrays. 2D - column 2



# FIFO

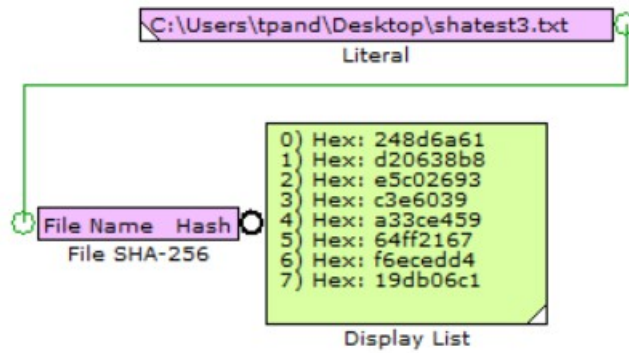
The FIFO component stores data each time the in clock goes true. The output changes when the out clock goes true. Functions - column 4





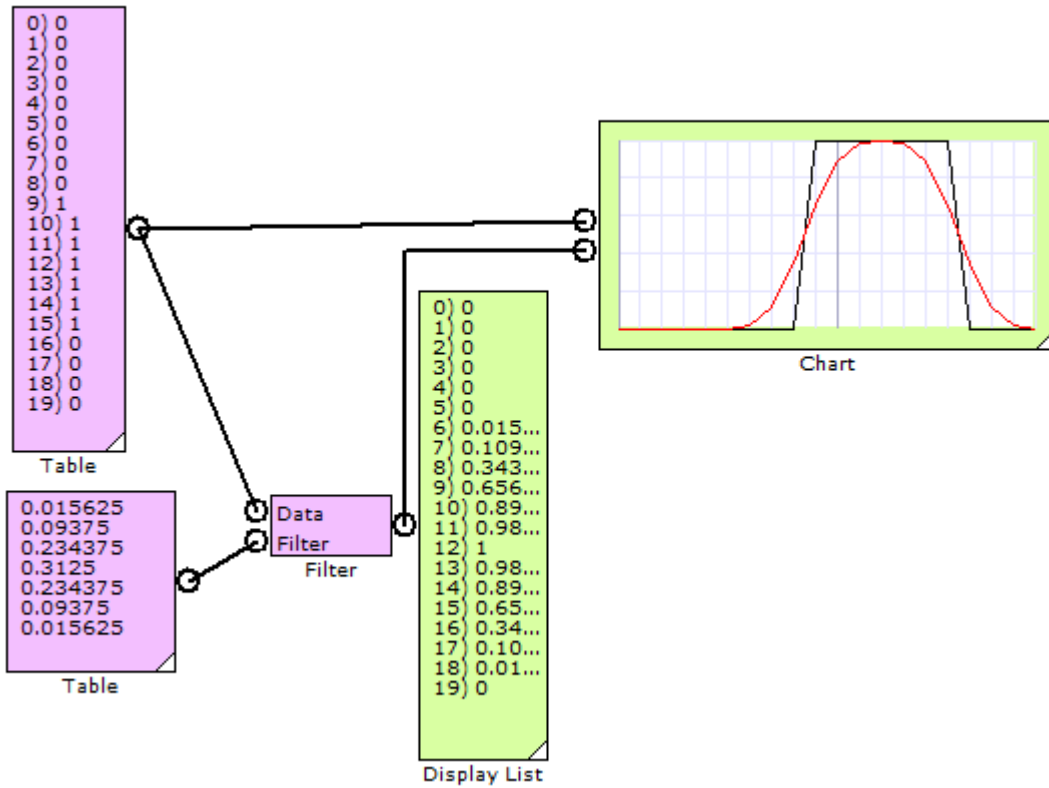
## File SHA-256

The File SHA-256 component calculates the US Secure Hash Algorithm on the file pointed to by the input. Arrays – column 5



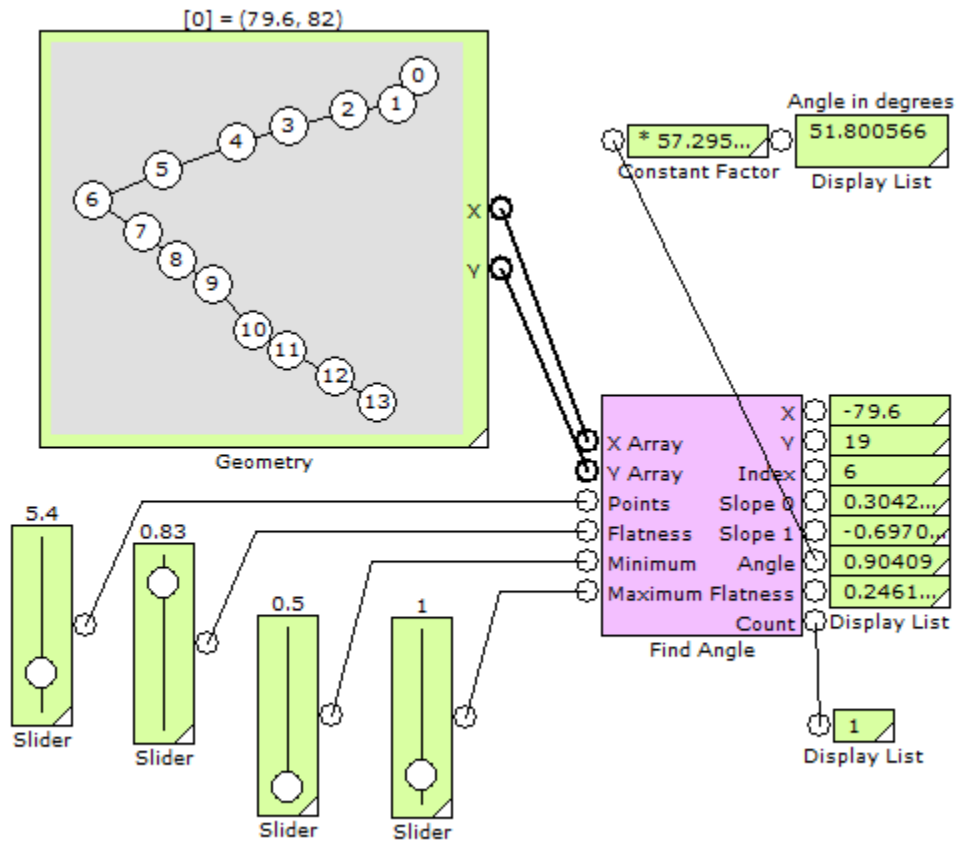
# Filter

The Filter component processes the Data array with the Filter array. The output is the filtered array.  
Arrays - column 2



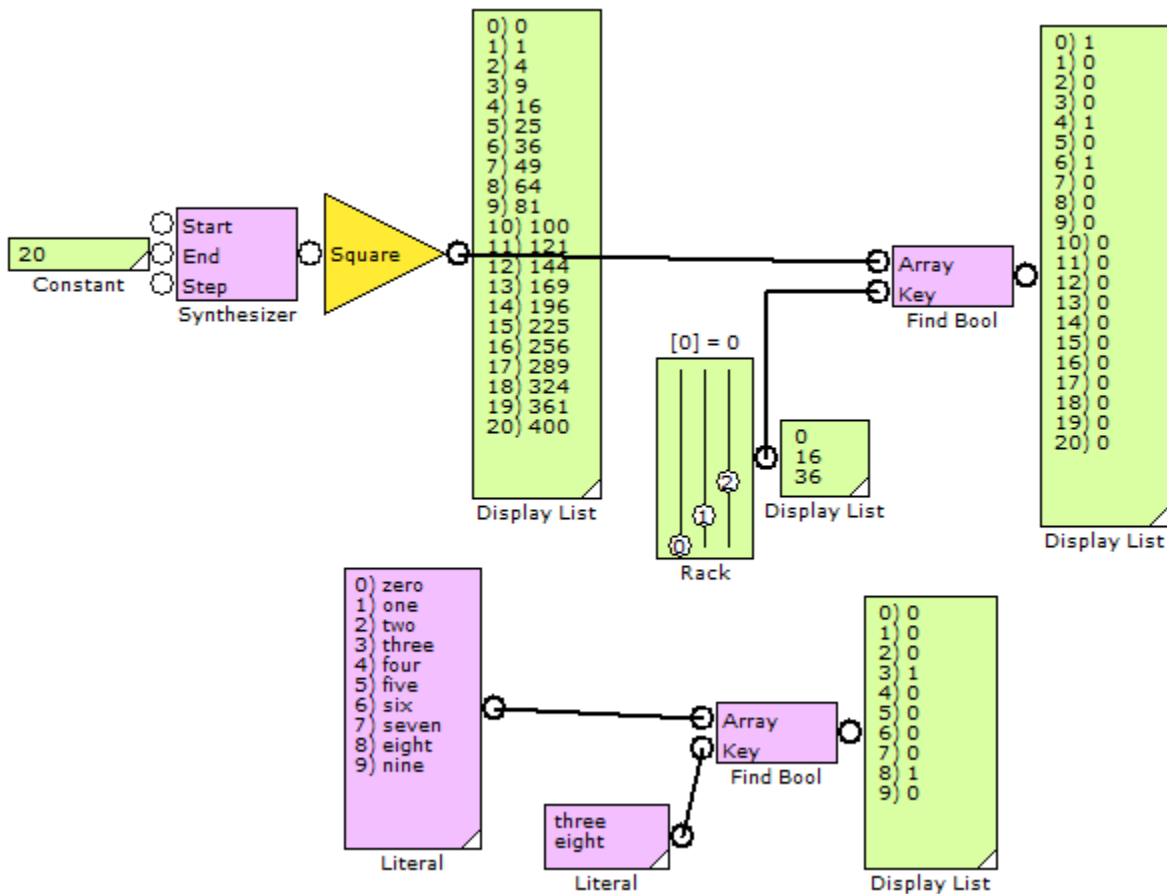
# Find Angle

The Find Angle scans along a 2D polyline looking for a sharp change in direction. It breaks the segment into two segments and fits each to a straight line. Arrays - column 5



# Find Bool

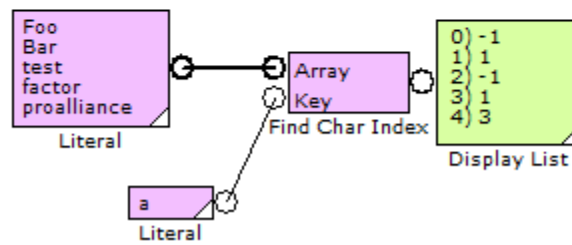
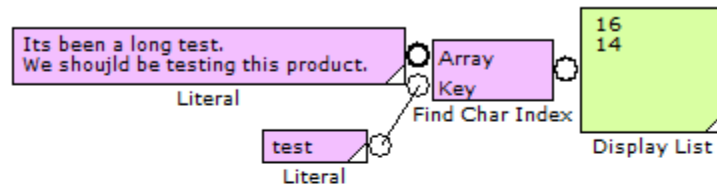
The Find Bool component scans the Array for values matching the Key input. It outputs an array with true in positions that matched and false where it didn't match. Arrays - column 2



# Find Char Index

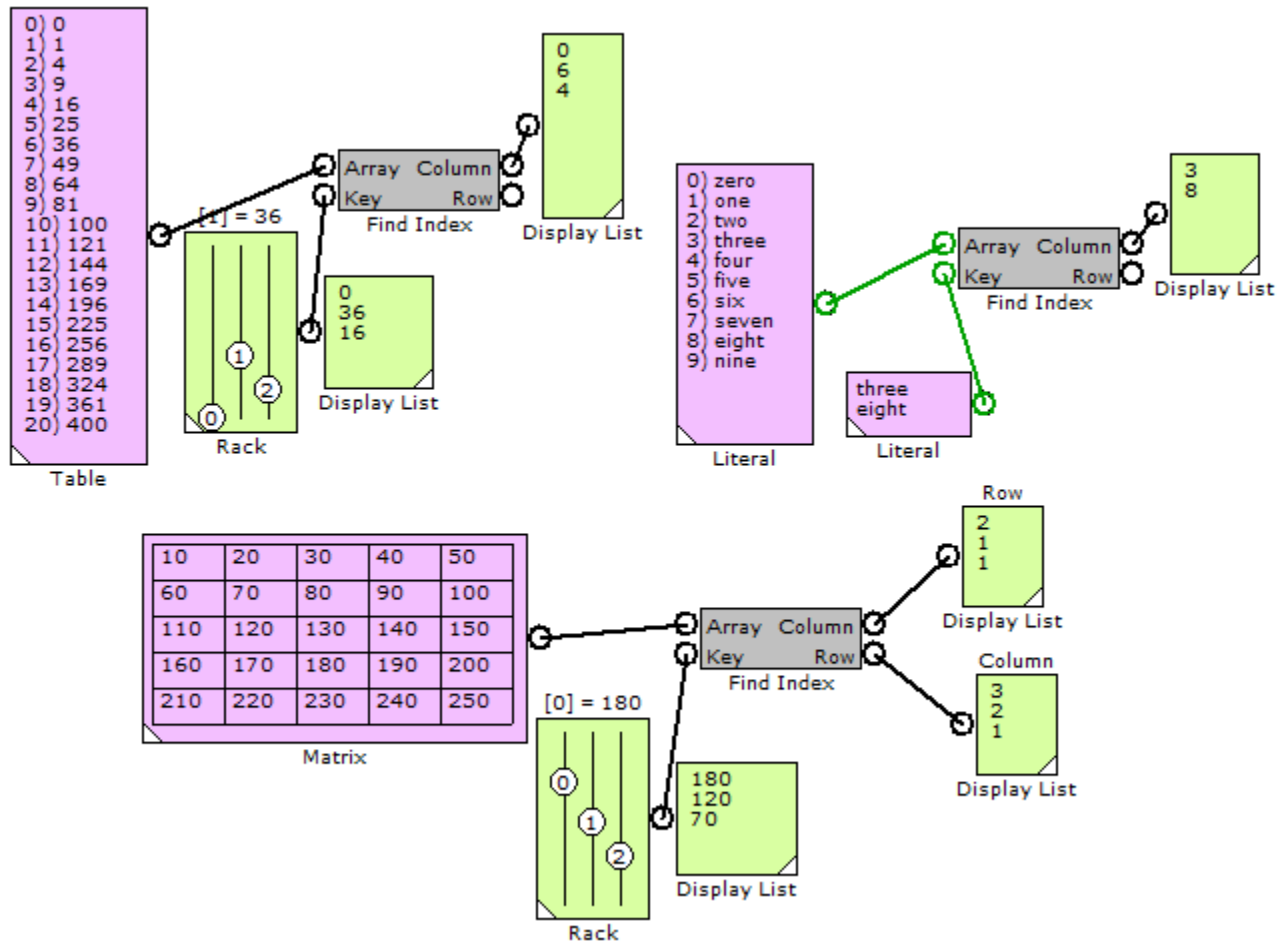
The Find Char Index component outputs the position of the key in the input array. column 6

Arrays -



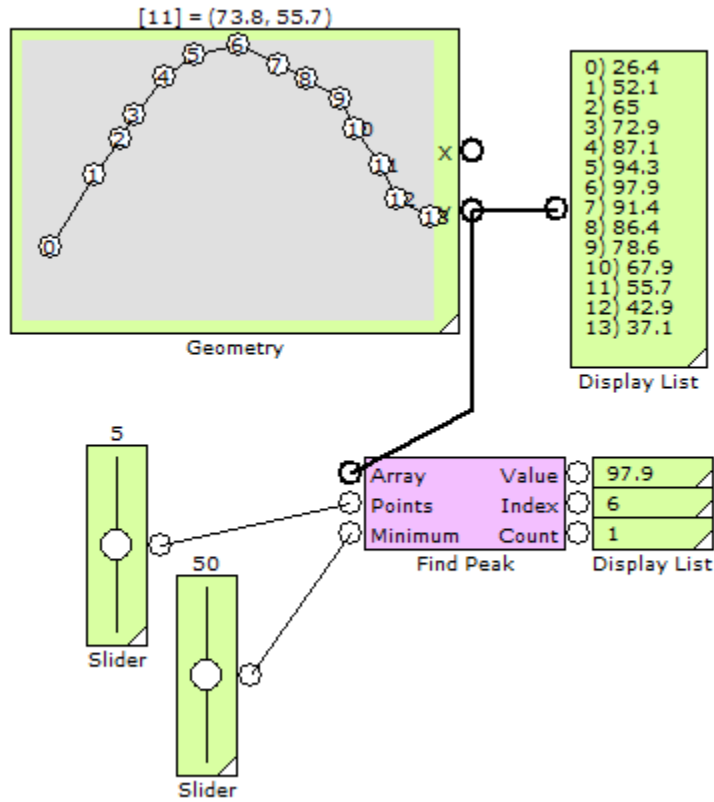
# Find Index

The Find Index component searches the Array for values matching the Key input. It outputs the index of each match in the Array. Arrays - column 2



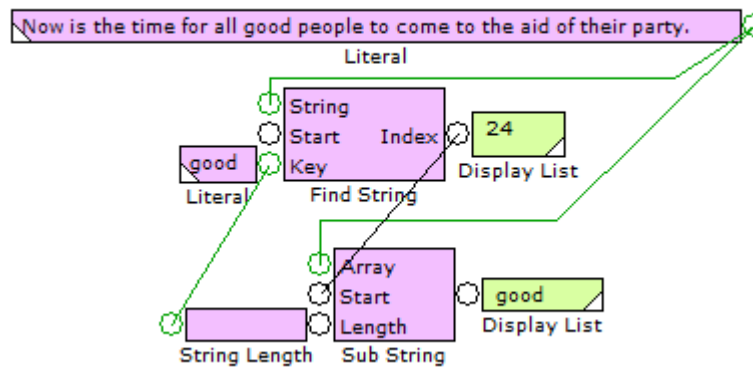
# Find Peak

The Find Peak scans along an array looking for peaks. Peaks must be in a group of Points that is at least Minimum in value. Arrays - column 5



## Find String

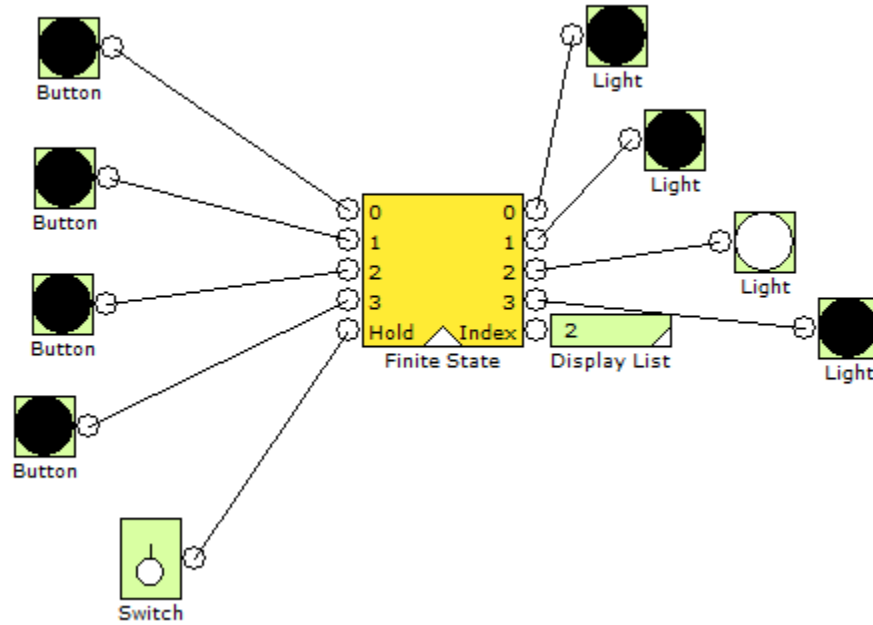
The Find String component outputs the index of where it finds the Key string of the String input. If the Start input is connected, the search starts at that index. If the Start input is negative, the search is started from the end of the string. Arrays – column 6





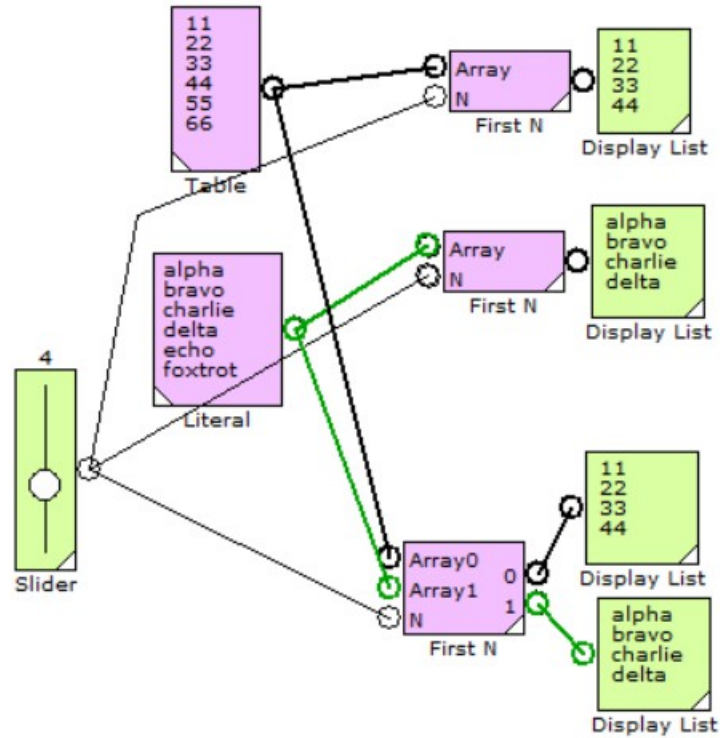
# Finite State

The Finite State component is used to create sequences. Only one output will be true at any time. To change states the input for that state has to be true. Functions - column 2



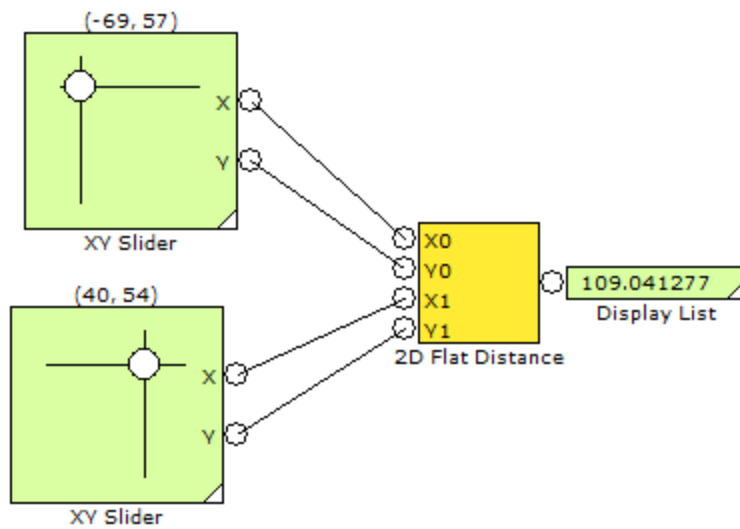
# First N

The First N component outputs just the first few values in an array. Input the number N to indicate the number of values you want at the output. Arrays - column 1



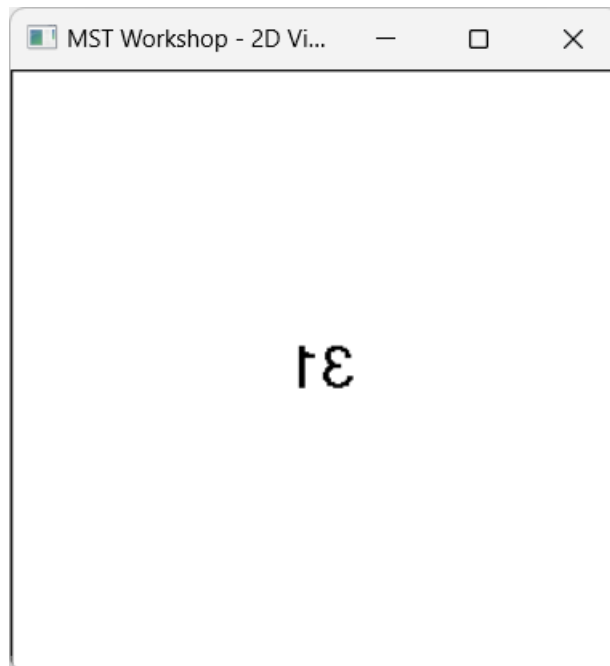
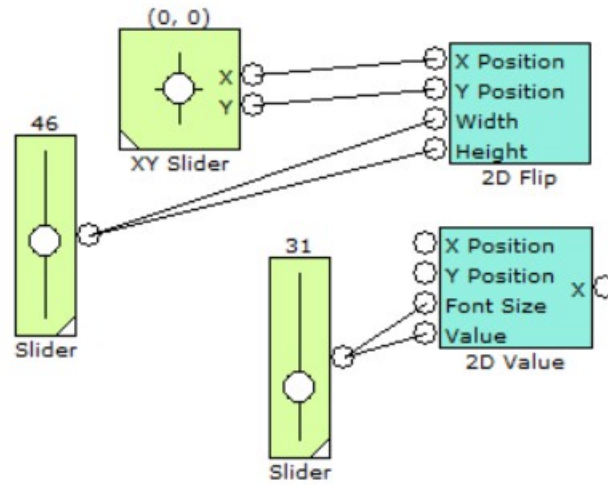
## 2D Flat Distance

The 2D Flat Distance component calculates the distance between two points on a plane. Functions - column 8



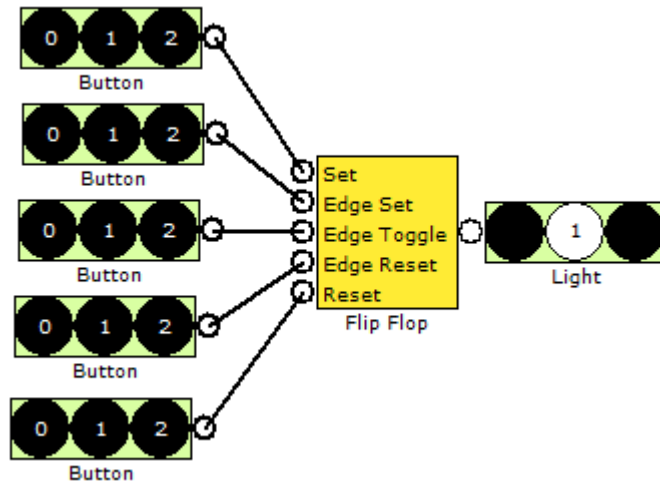
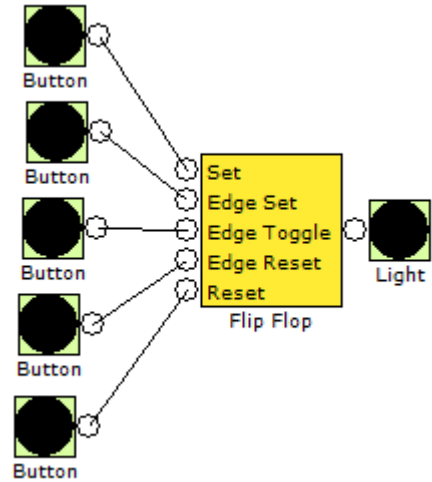
## 2D Flip

The 2D Flip component can flip the image in the 2D Window. Optionally you can flip horizontally or vertically or both. Note: This component only affects the items in the 2D View window that are under it. 2D – column 7



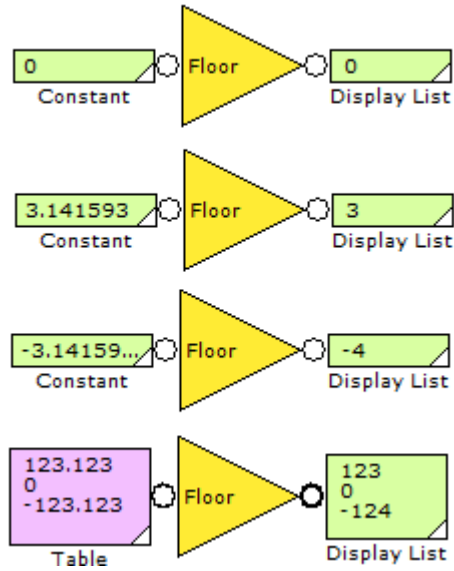
# Flip Flop

The Flip Flop component is a memory device that can toggle between true and false. Functions - column 3



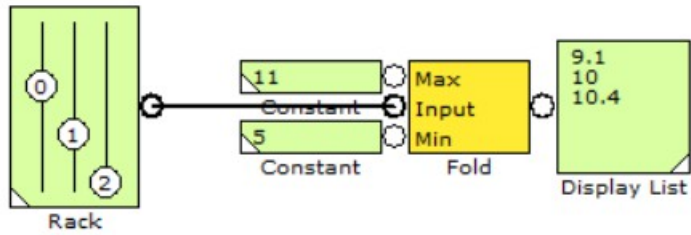
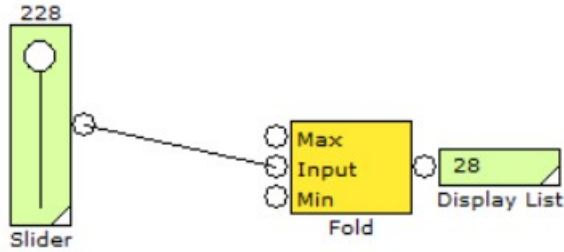
# Floor

The Floor component outputs an integer less than or equal to the input. Input can be scalar or an array.  
Functions - column 2



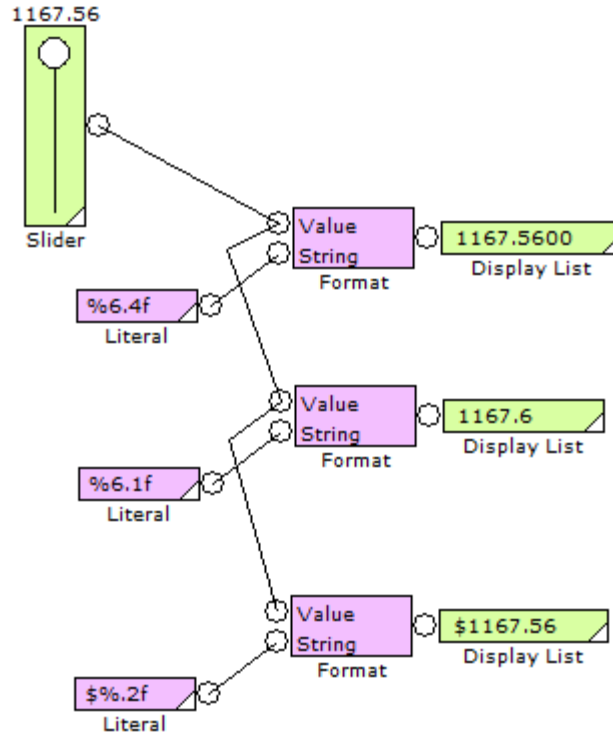
# Fold

The Fold component folds the input value until it is between the High and Low values. Functions – column 2



# Format

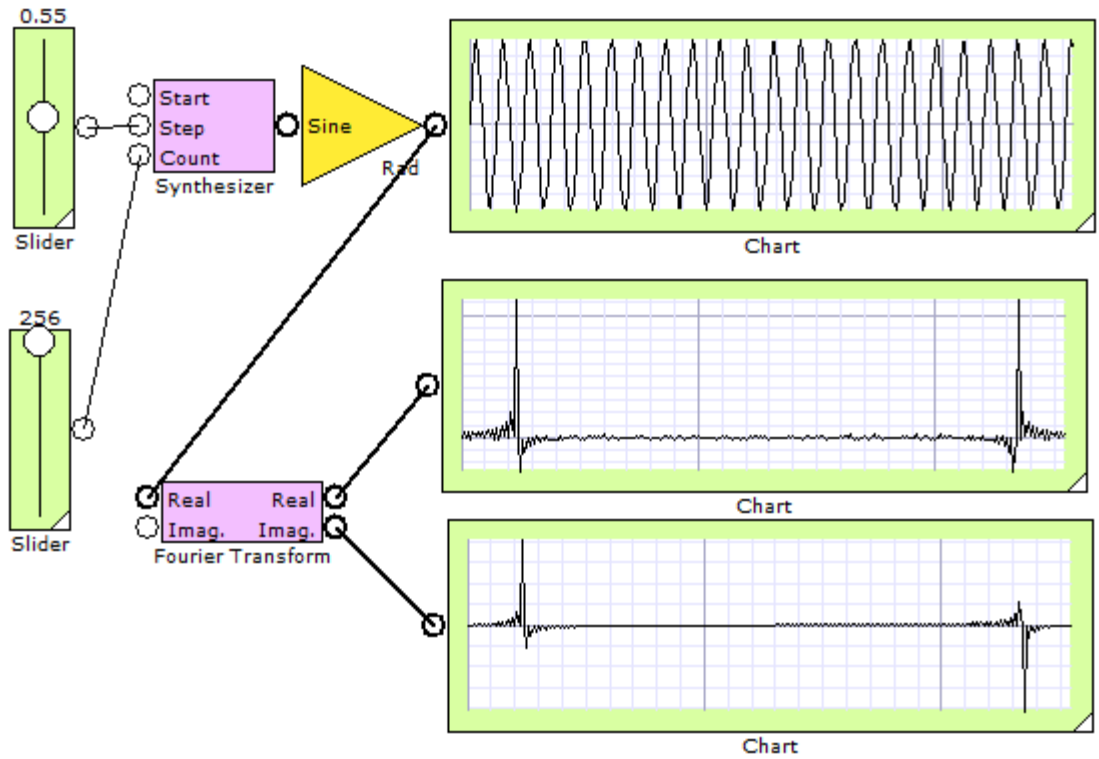
The Format component converts the input numbers to strings. Arrays - column 6





# Fourier Transform

The Fourier Transform calculates the Fourier transform of the input. The result is to generate a spectrum analysis of the input. Arrays - column 5



# Frame

The Frame component is used to draw a frame around a section of your solution for clarity only.  
Controls - column 5

Frame:

- White LED
- Red LED
- Orange LED
- Yellow LED
- Green LED
- Blue LED
- Violet LED
- Gray LED

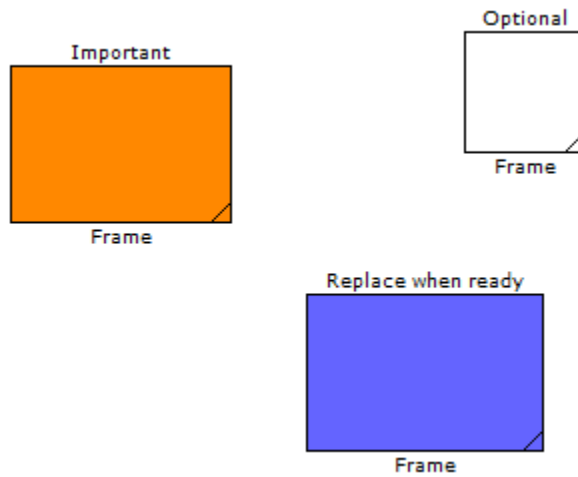
Other:

Red

Green

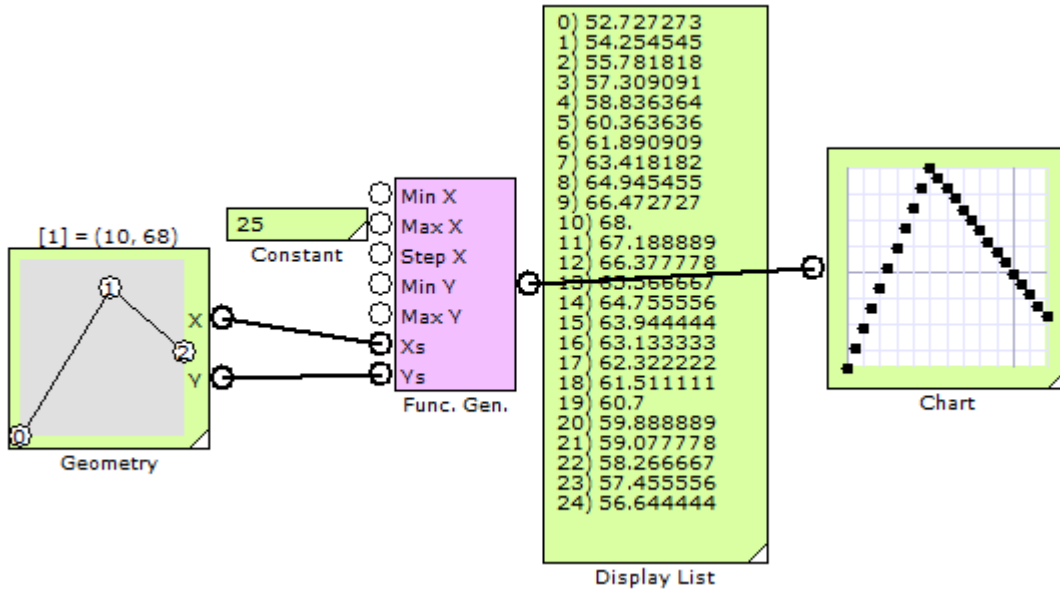
Blue

OK Cancel



# Func. Gen.

The Func. Gen. component takes a few xy points and creates a function in the range specified at the inputs. Arrays - column 1



# Func. Points

The Func. Points lets you create a function by setting the Y values on the surface of the component component. Note: This component can be operated using the wheel on the mouse. Controls - column 1

Move To Top	Delete
Move Up	Details...
Move Down	Duplicate
Move To Bottom	Edit Label...
Copy	Options...
Paste	

Function

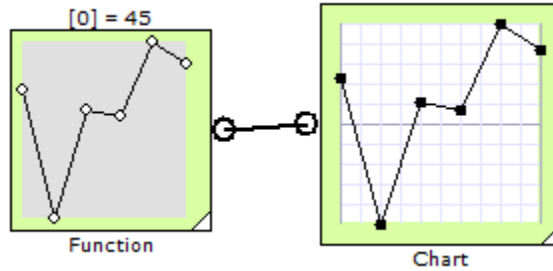
Set Point Count:

Set Minimum Y:

Set Maximum Y:

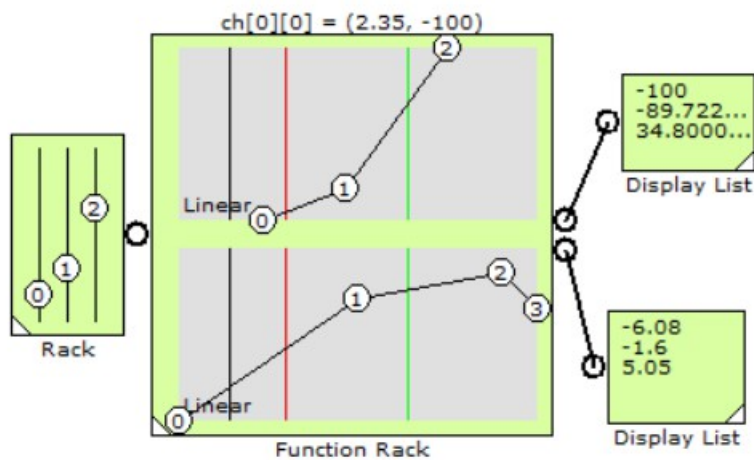
Set Detent:

OK Cancel



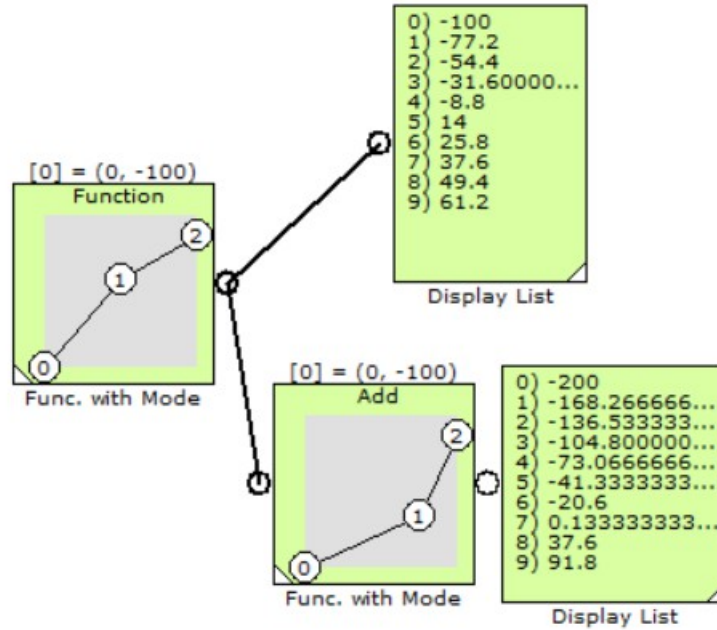
## Func. Rack At

The Func. Rack At component lets you create several functions by drawing them on the surface of the component. This version outputs a value for each channel. You can have up to 8 channels. This version allows you to change the X and Y position of each point. The output can be linear interpolation, a sinusoid between points, a spline, or simple stair steps. Note: This component can be operated using the wheel on the mouse. Controls – column 1



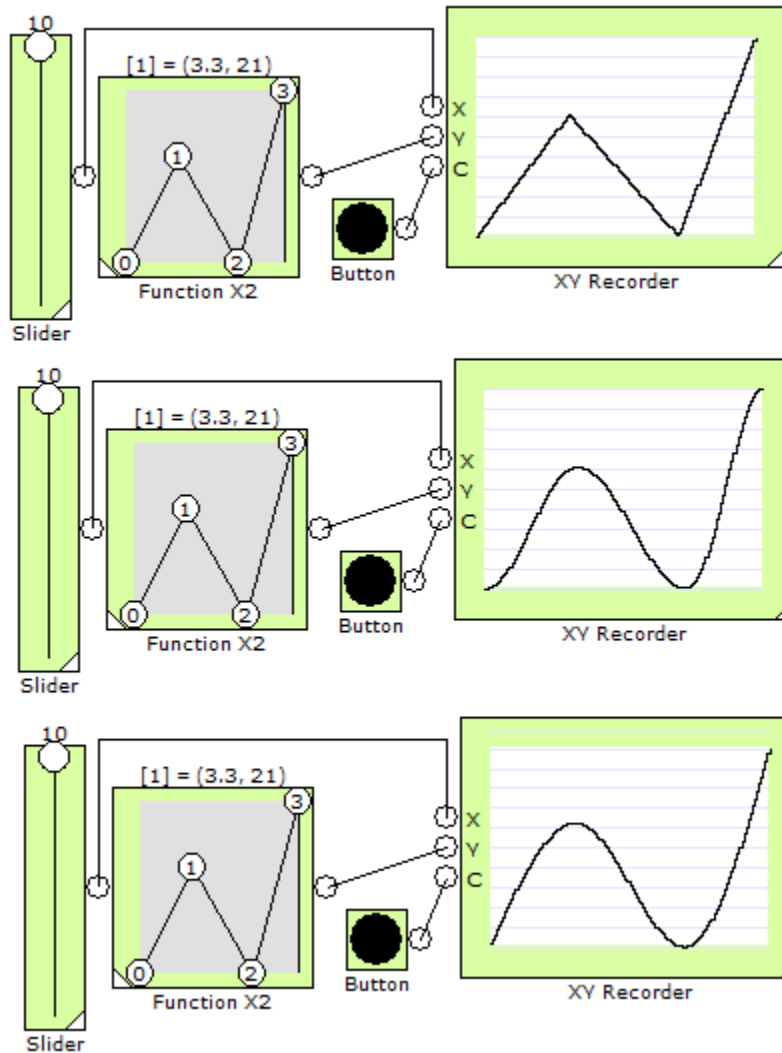
## Func. with Mode

The Function with Mode component lets you create a function by setting the XY position of points on the surface of the component. This version outputs an array based on the positions of the knobs. This version allows you to change the X and Y position of each point. The output can be a linear interpolation, a sinusoid between points, a spline, or simple stair steps. Note: This component can be operated using the wheel on the mouse. Controls - column 1



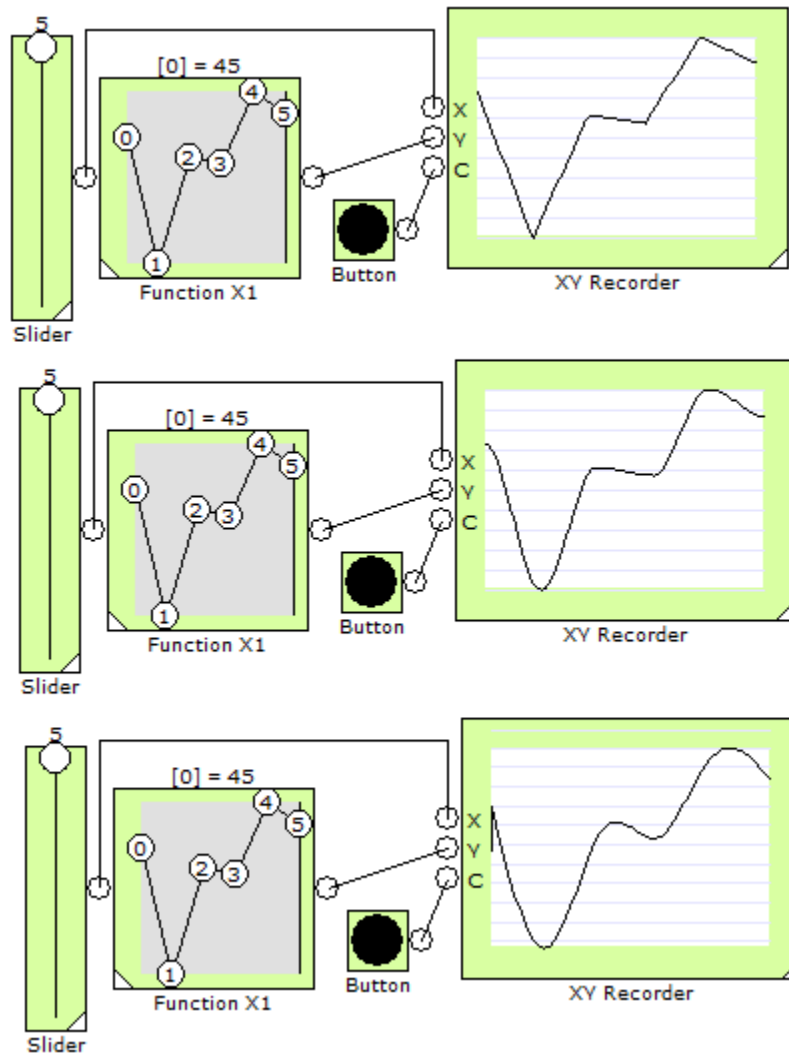
## Func. XY At

The Func. XY At lets you create a function by setting the XY position of points on the surface of the component. This version outputs a single value based on a single input or an array of outputs if the input is an array. This version allows you to change the X and Y position of each point. The output can be a linear interpolation, a sinusoid between points or a simple stair step. Note: This component can be operated using the wheel on the mouse. Controls - column 1



## Func. Y At

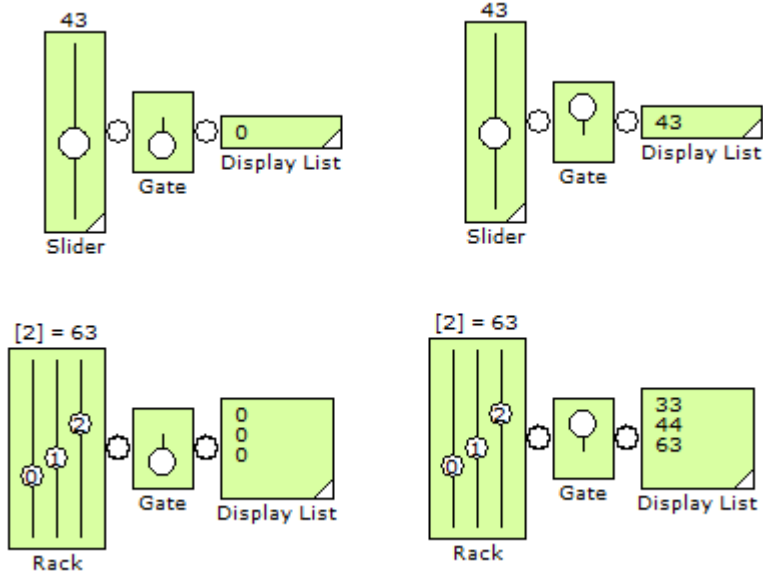
The Function Y At lets you create a function by setting the Y position of points on the surface of the component. This version outputs a single value based on a single input or an array of outputs if the input is an array. This version allows you to change the Y position of each point. The output can be a linear interpolation, a sinusoid between points or a simple stair step. Note: This component can be operated using the wheel on the mouse. Controls - column 1





# Gate

The Gate component connects the input to the output when it is on. When it is off, the output is zero.  
Note: This component can be operated using the wheel on the mouse. Controls - column 4



# Gather

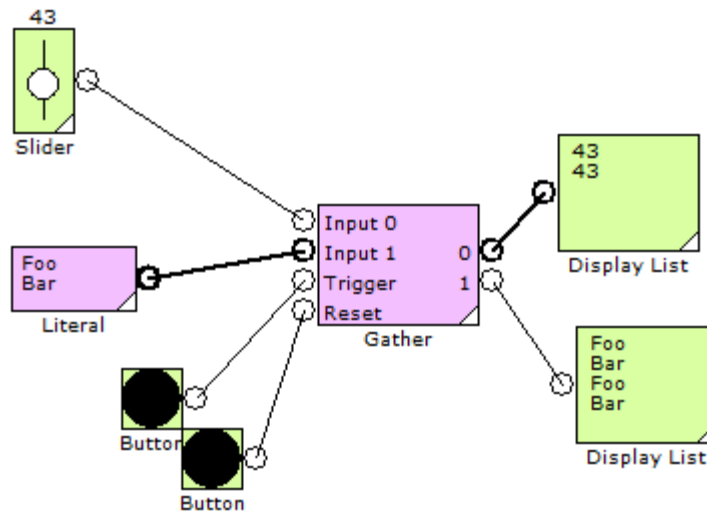
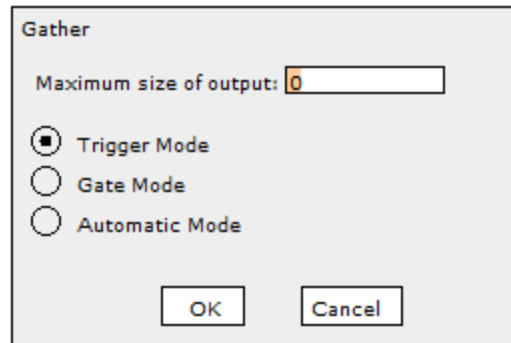
The Gather component stores the data at the input terminal each time the trigger goes true. The input can be numbers or strings and scalar or an array. There are three modes:

Trigger mode gathers inputs on the positive edge of the Trigger input.

Gate mode gathers inputs when the Gate is true and the clock is running.

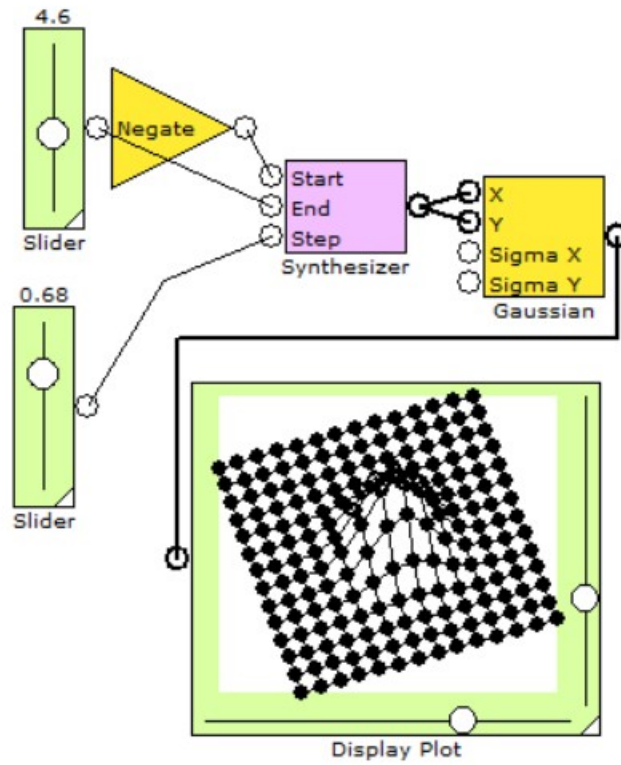
Automatic mode gathers inputs when the Automatic input is true and any input changes.

Arrays - column 1



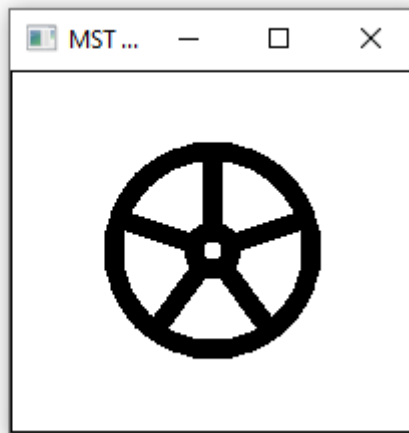
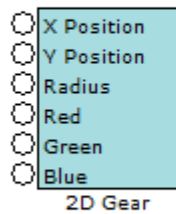
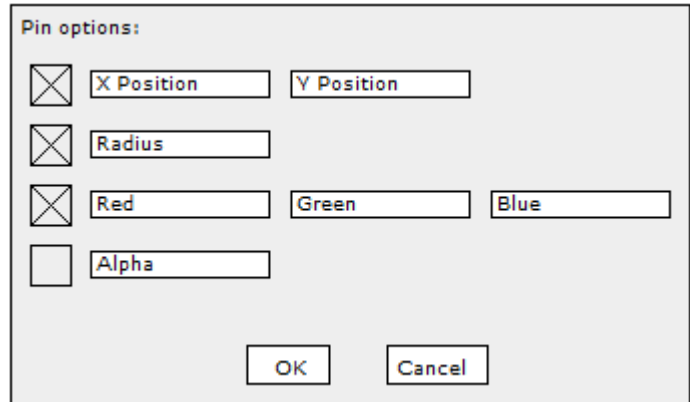
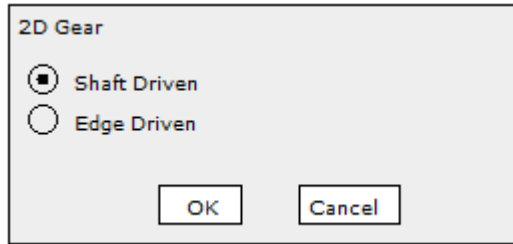
# Gaussian

The Gaussian component calculates the Gaussian in 1, 2, or 3-Dimensions. Functions – column 5



## 2D Gear

The Gear component creates a gear that can attach to other Gears or other simulation objects like Motors. Place the Gear at the same X, Y position as a motor to connect the gear to the motor. 2D - column 3



## 3D Gear

The 3D Gear component creates a gear that can attach to other 3D Gears or other simulation objects like 3D Motors. Place the Gear at the same X, Y, Z position and orientation as a motor to connect the gear to the motor. 3D - column 4

**3D Gear**

Tube Steps:

Ring Steps:

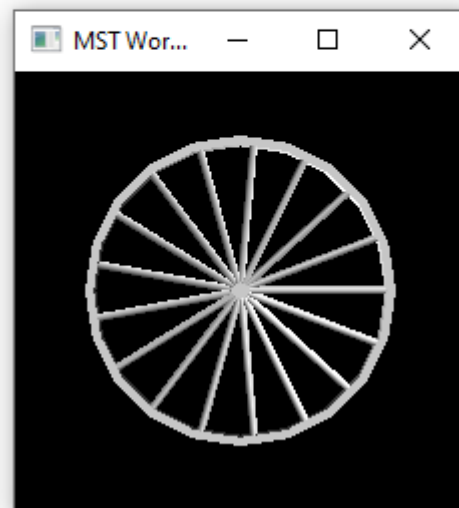
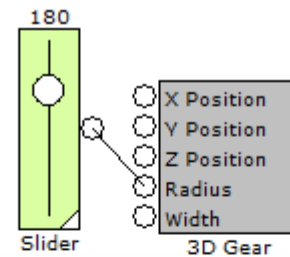
Shaft Driven  
 Edge Driven

**Pin options:**

X Position     Y Position     Z Position  
 X Rotate     Y Rotate     Z Rotate  
 Radius     Width  
 Red     Green     Blue  
 Alpha  
 [4x4]

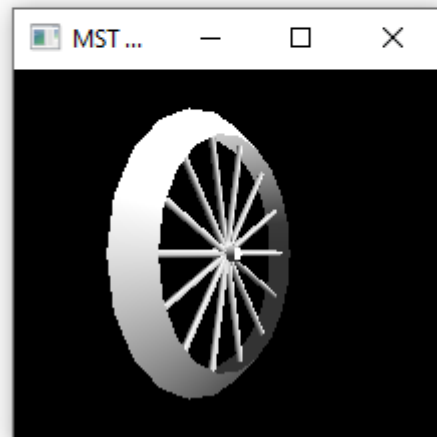
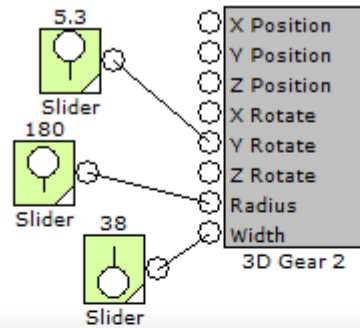
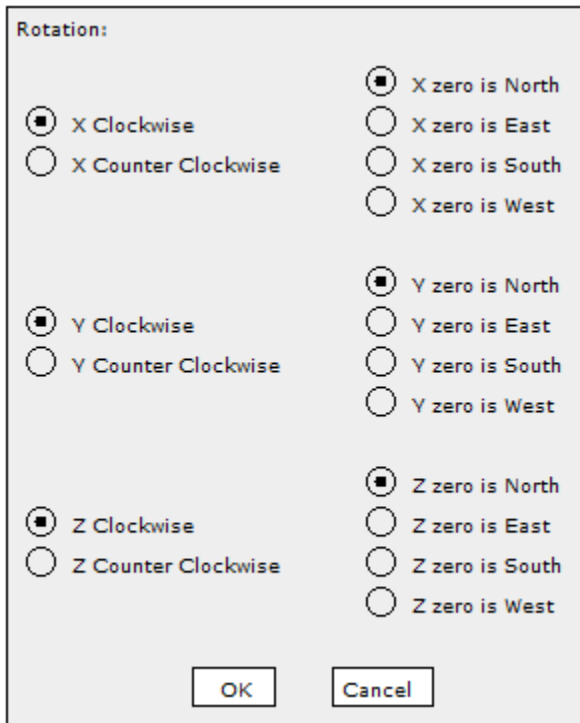
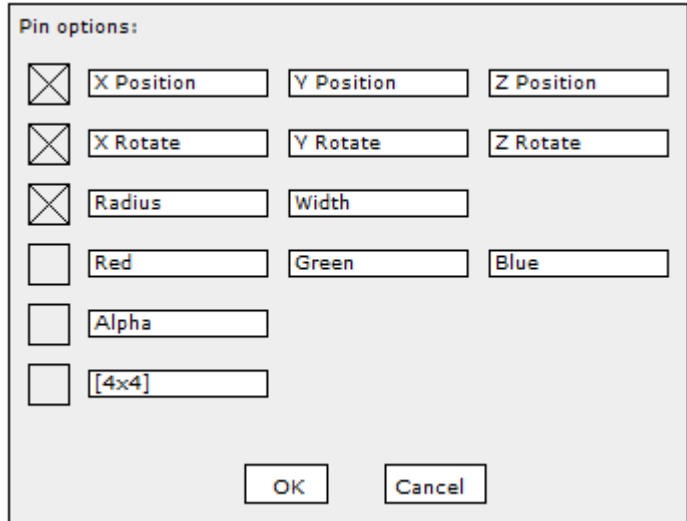
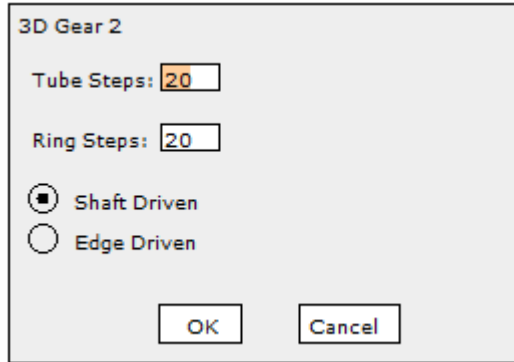
**Rotation:**

<input checked="" type="radio"/> X Clockwise <input type="radio"/> X Counter Clockwise  <input checked="" type="radio"/> Y Clockwise <input type="radio"/> Y Counter Clockwise  <input checked="" type="radio"/> Z Clockwise <input type="radio"/> Z Counter Clockwise	<input checked="" type="radio"/> X zero is North <input type="radio"/> X zero is East <input type="radio"/> X zero is South <input type="radio"/> X zero is West  <input checked="" type="radio"/> Y zero is North <input type="radio"/> Y zero is East <input type="radio"/> Y zero is South <input type="radio"/> Y zero is West  <input checked="" type="radio"/> Z zero is North <input type="radio"/> Z zero is East <input type="radio"/> Z zero is South <input type="radio"/> Z zero is West
---	---



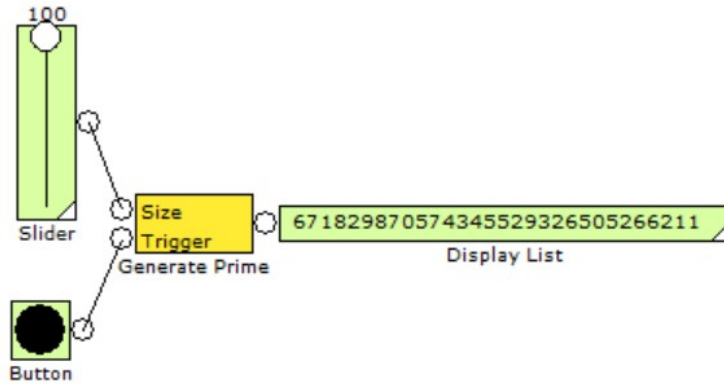
## 3D Gear 2

The 3D Gear 2 component creates a gear that can attach to other 3D Gear 2s or other simulation objects like 3D Motors. Place the Gear 2 at the same X, Y, Z position and orientation as a driven object then the opposite end will drive another object. 3D - column 4



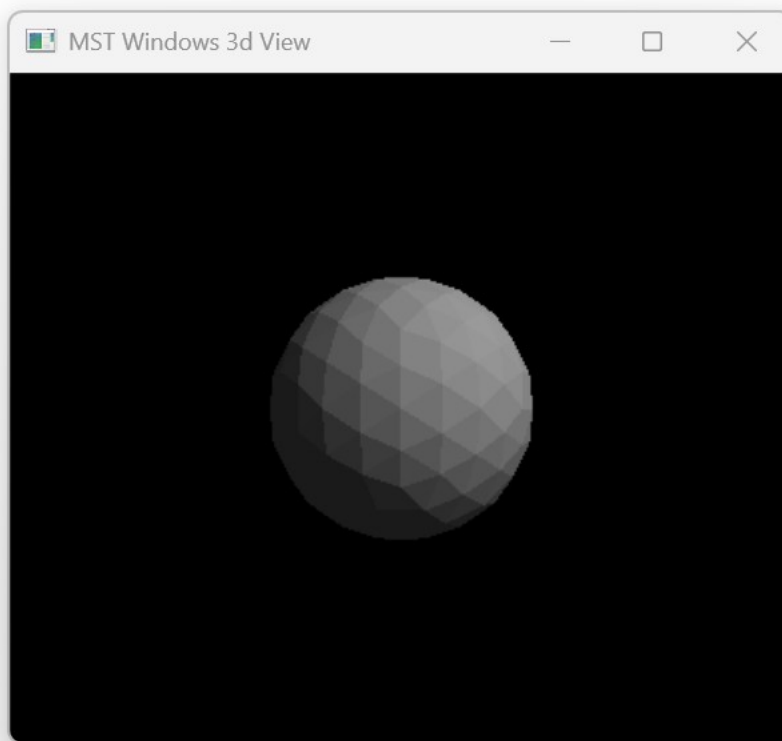
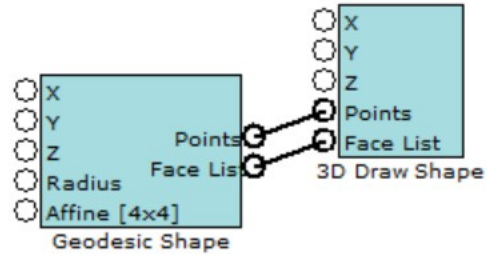
# Generate Prime

The Generate Prime component finds a prime number with the number of bits specified. The output is a string. Functions - column 8



## Geodesic Shape

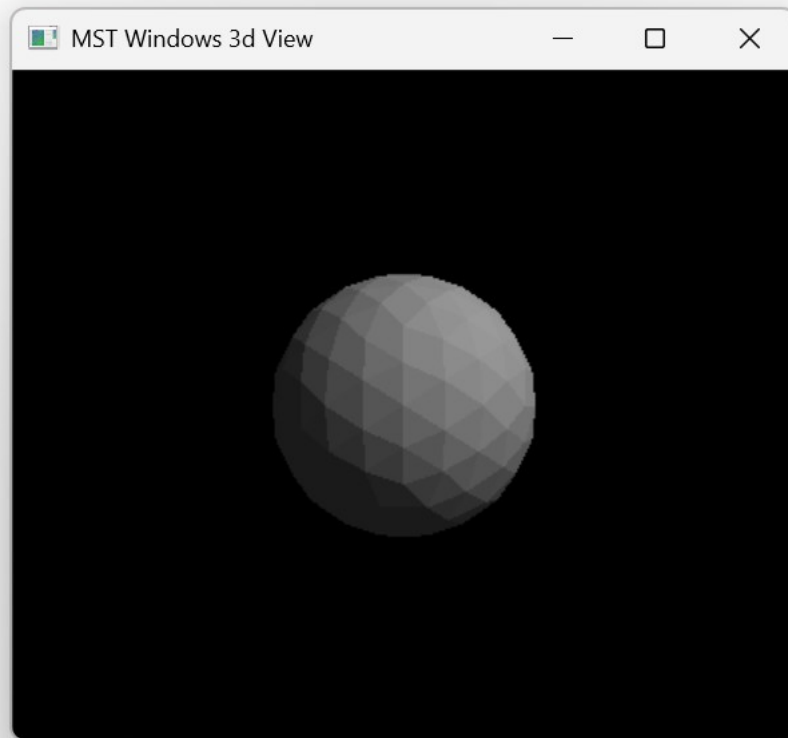
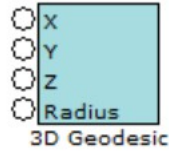
The Geodesic Shape component creates triangular mesh composed of points and faces for a geodesic. Use its outputs to reshape or draw it in the 3D View by connecting it to the 3D Draw Shape component. 3D - column 6





## 3D Geodesic

The 3D Geodesic draws a spherical geodesic in the 3D View Window. The X, Y, Z, and Radius inputs set the position and size of the geodesic. The Affine [4x4] array input modifies the size, position and orientation. 3D – column 2

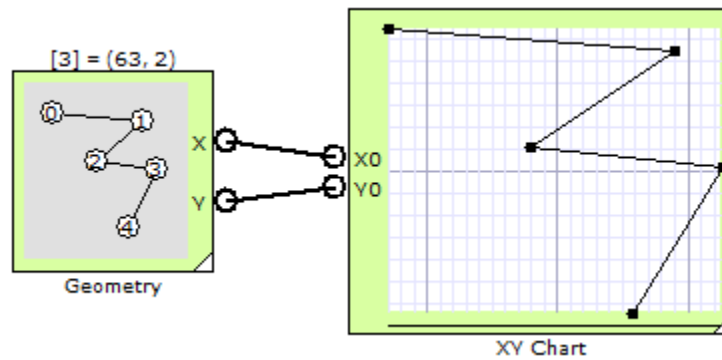


# Geometry

The Geometry lets you control a number of constants in two directions, horizontal (x) and vertical (y).  
Controls - column 1

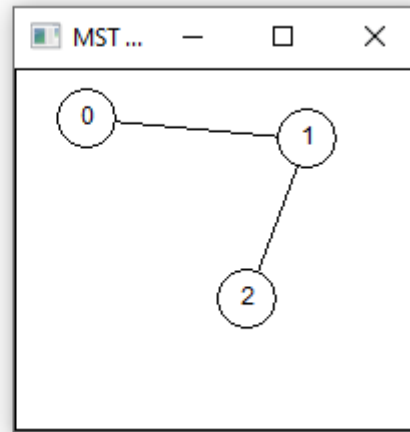
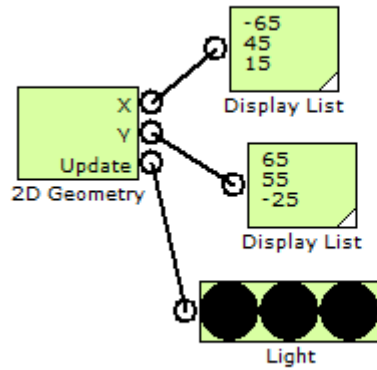
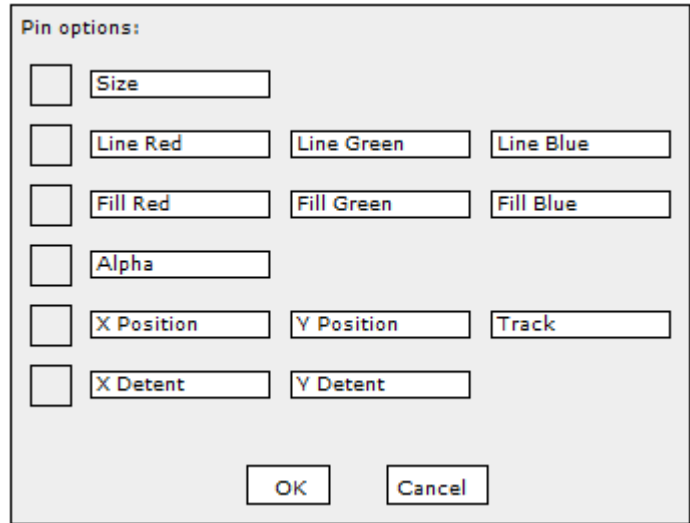
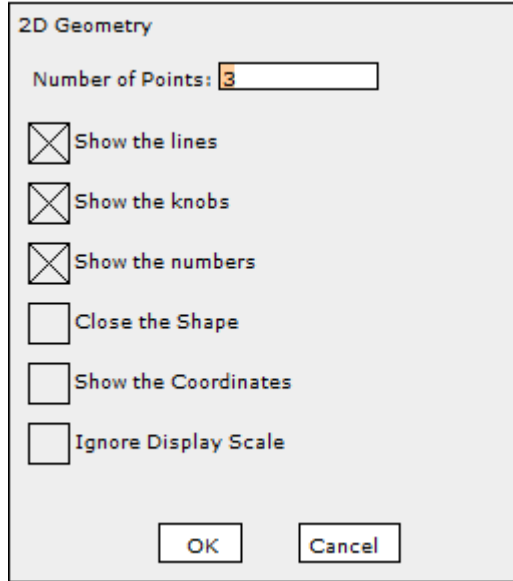
Move To Top	Delete
Move Up	Details...
Move Down	Duplicate
Move To Bottom	Edit Label...
Copy	Options...
Paste	

Geometry	
Set Point Count:	<input type="text" value="5"/>
Set Minimum X:	<input type="text" value="-100"/>
Set Maximum X:	<input type="text" value="100"/>
Set Minimum Y:	<input type="text" value="-100"/>
Set Maximum Y:	<input type="text" value="100"/>
<input type="button" value="OK"/> <input type="button" value="Cancel"/>	



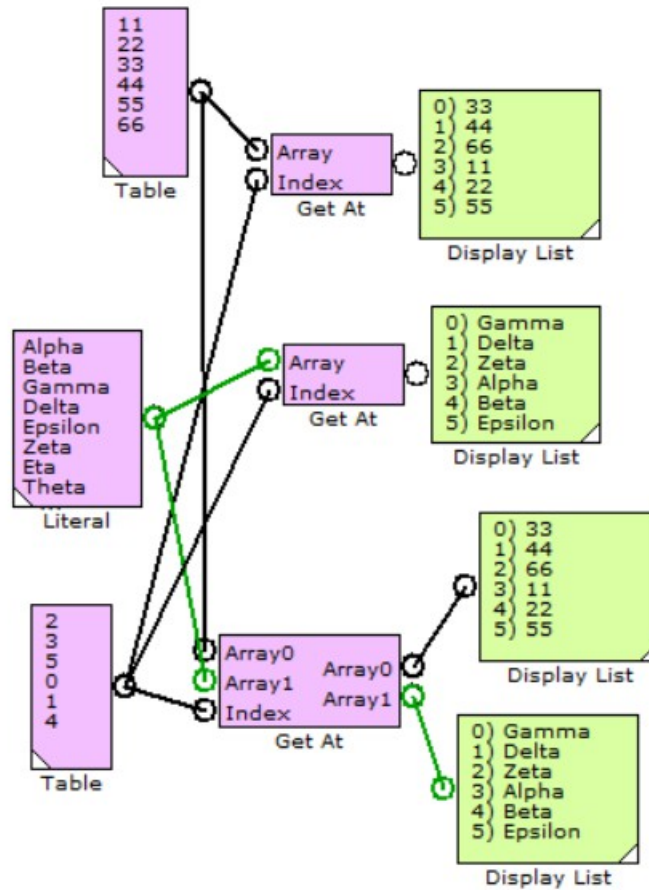
# 2D Geometry

The 2D Geometry lets you control a number of constants on the 2D canvas. Controls - column 2



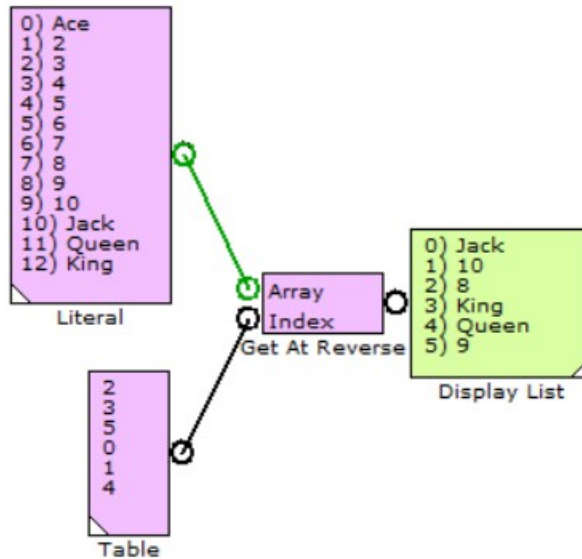
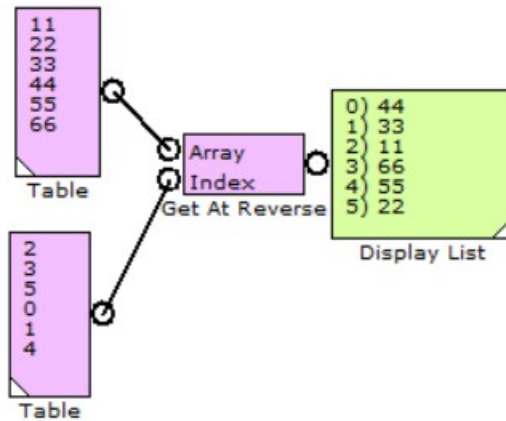
# Get At

The Get At component gets the Array value at the Index input. It outputs the value(s) at the positions given by the Index(s). Optionally you can specify what to return when the Index is out of the Array's range. Arrays - column 2



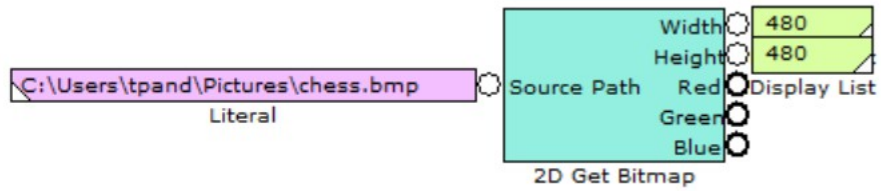
# Get At Reverse

The Get At Reverse component gets the Array value at the Index input starting at the end of the Array. It outputs the value(s) at the positions taken from the end of the array. An index of zero returns the last element of the Array. Optionally you can specify what to return when the Index is out of the Array's range. Arrays - column 2



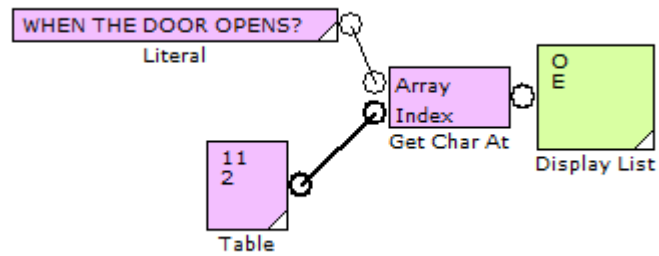
# Get Bitmap

The Get Bitmap reads a BMP image file given its path. It outputs the Width and Height of the bitmap in the file. The Red, Green, and Blue outputs have the 2 dimensional data of the image. 2D - column 2



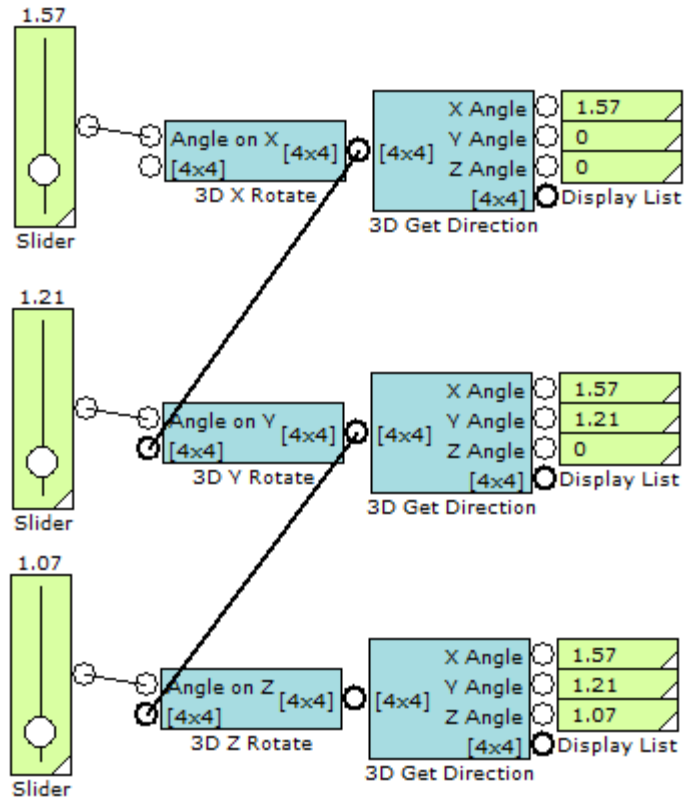
# Get Char At

The Get Char At component outputs one character of the input string. Arrays - column 6



### 3D Get Direction

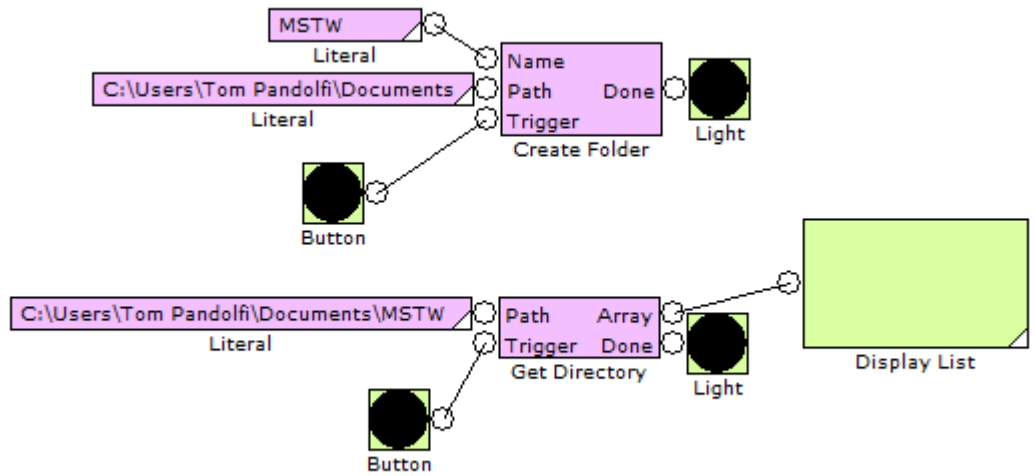
The 3D Get Direction returns the rotation angles of a 4 by 4 matrix on the X, Y, and Z axes and also returns the matrix with the rotation removed. 3D - column 3





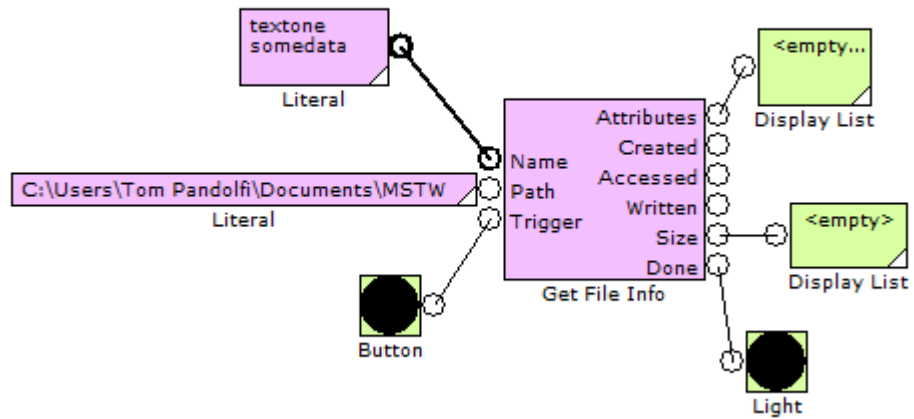
# Get Directory

The Get Directory component lists the contents of the specified directory. You can specify if you want files or sub directories to be listed. Arrays - column 7



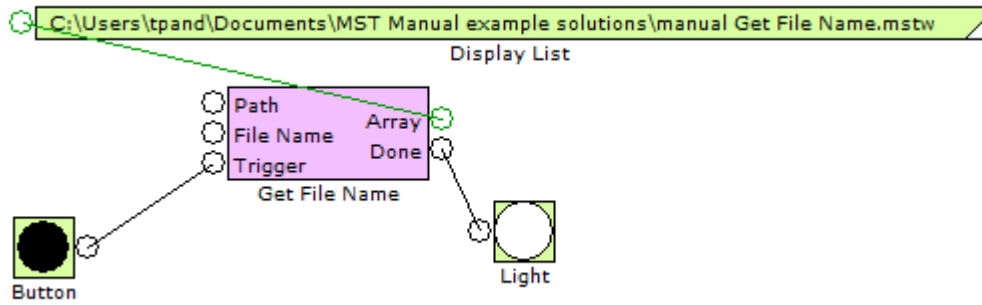
## Get File Info

The Get File Info component outputs information about the file specified. Including the size and time of creation. Arrays - column 7



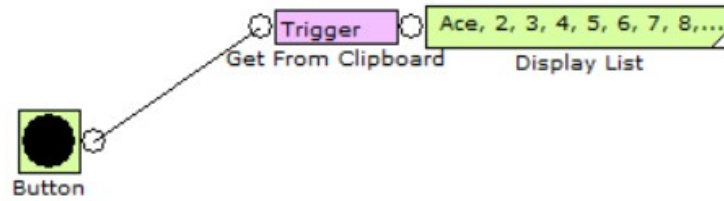
# Get File Name

The Get File Name component asks the user to select a file to open or save. Arrays - column 7



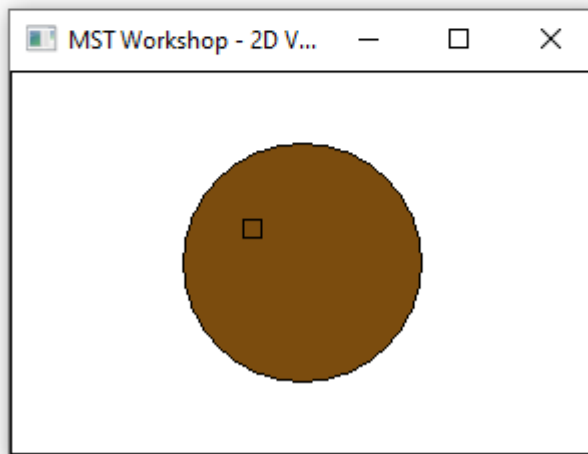
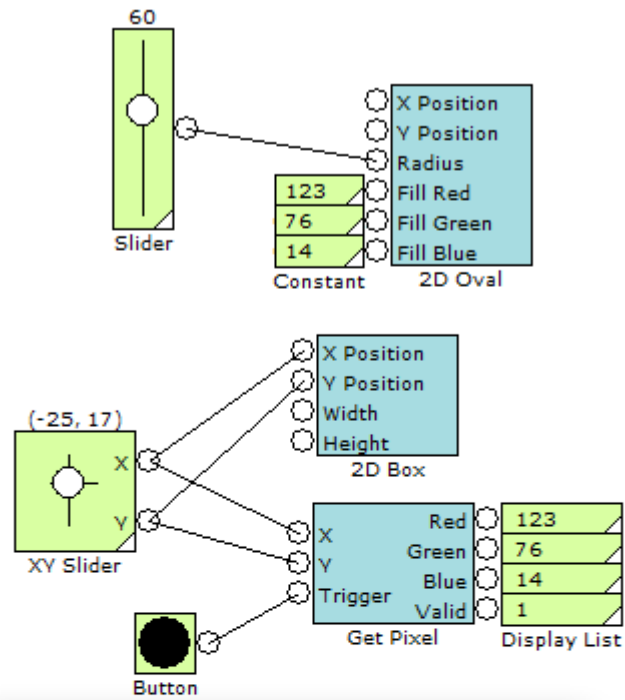
## Get From Clipboard

The Get From Clipboard component gets the text from the clipboard. Arrays – column 5



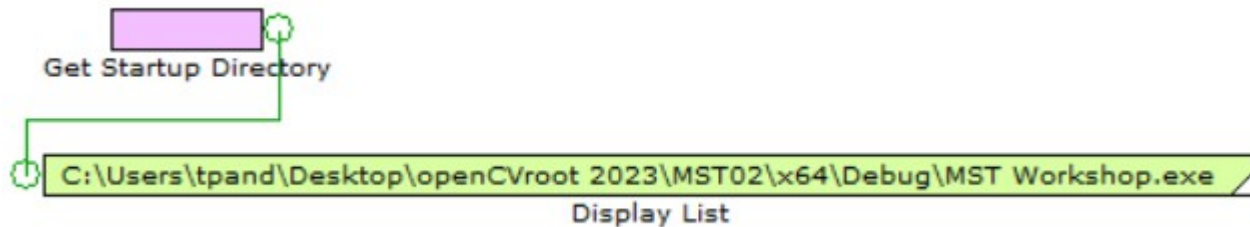
# Get Pixel

The Get Pixel component reads the canvas and outputs the color components of the pixel at the x, y coordinate. 2D - column 2



## Get Startup Directory

The Get Startup Directory component gets the directory path to where this program started. Arrays – column 7

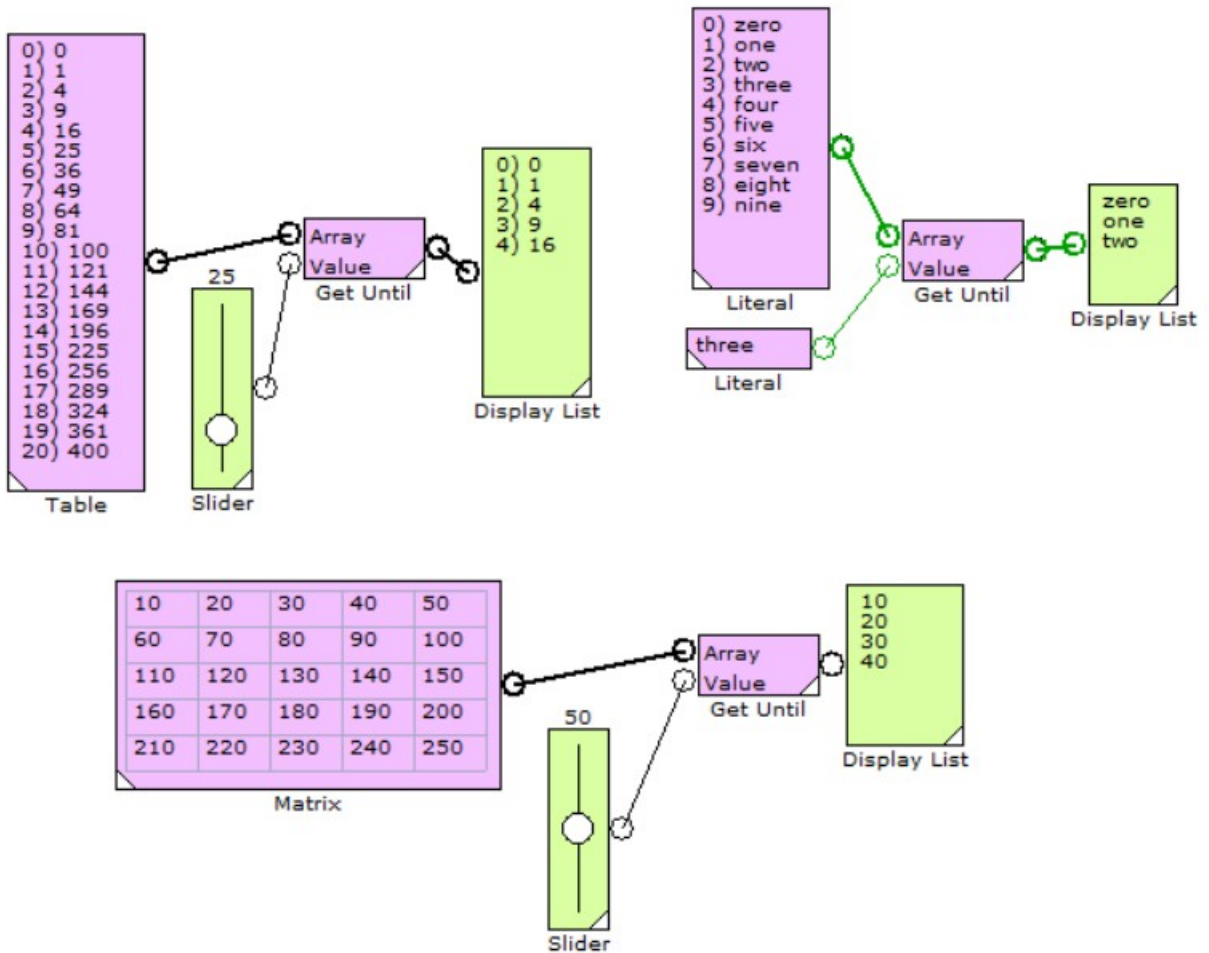


N

# Get Until

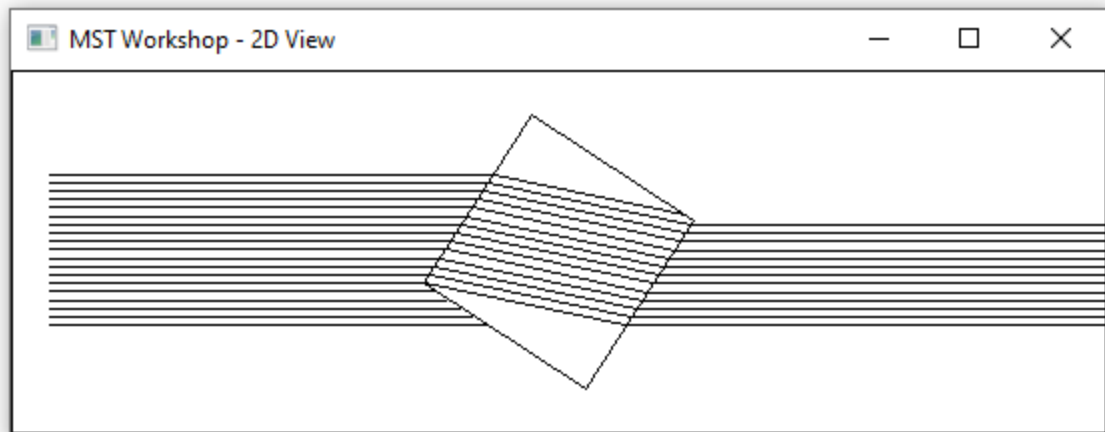
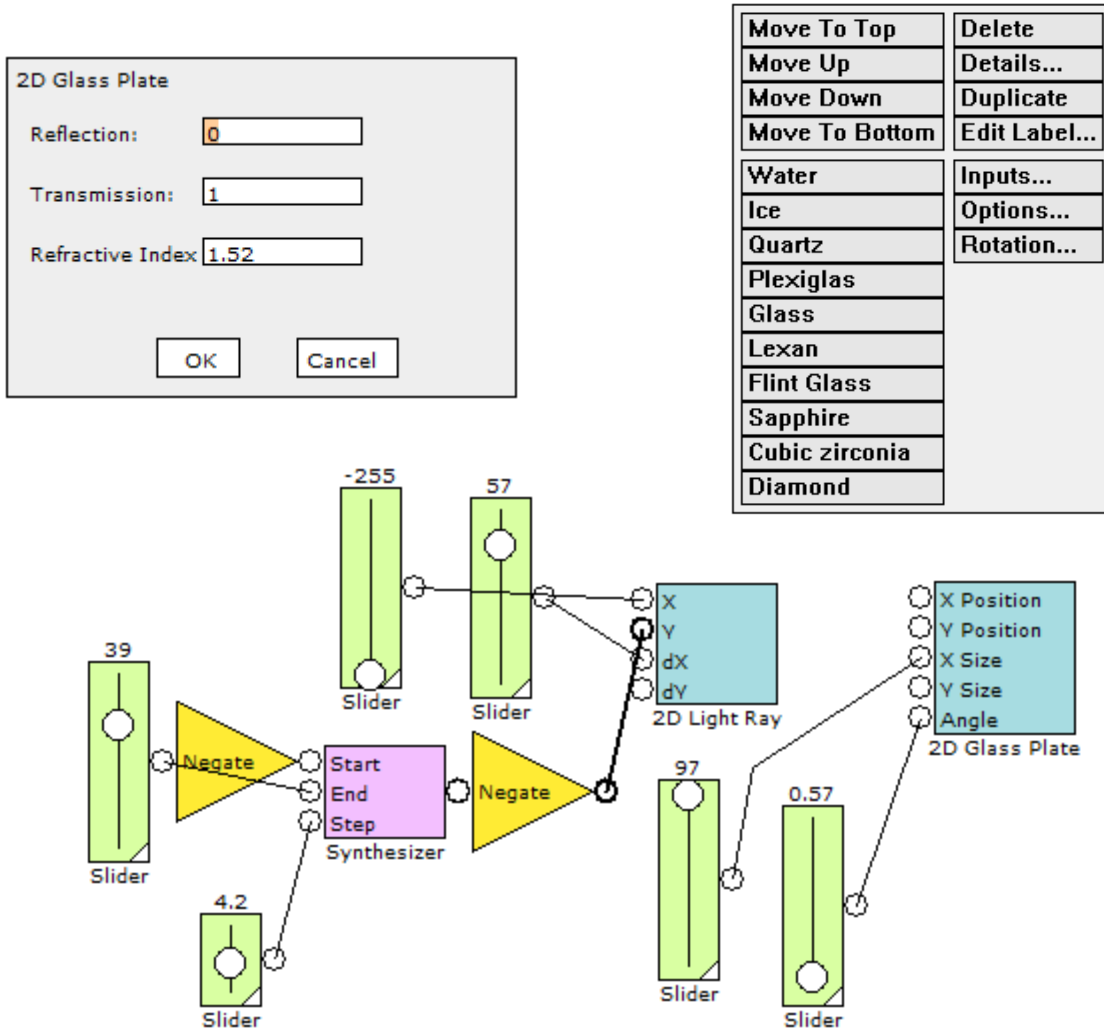
The Get Until component copies the Array values until the value at the Until input is reached. If nothing is connected to the Until input, then the array is copied until a blank item in the array is found.

Arrays - column 2



# 2D Glass Plate

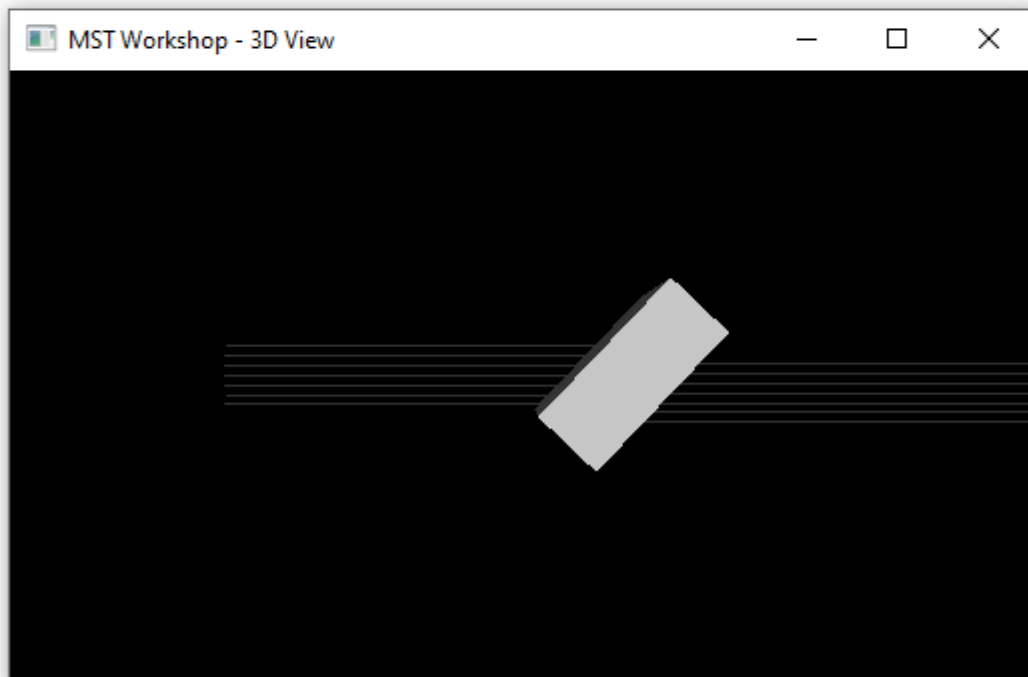
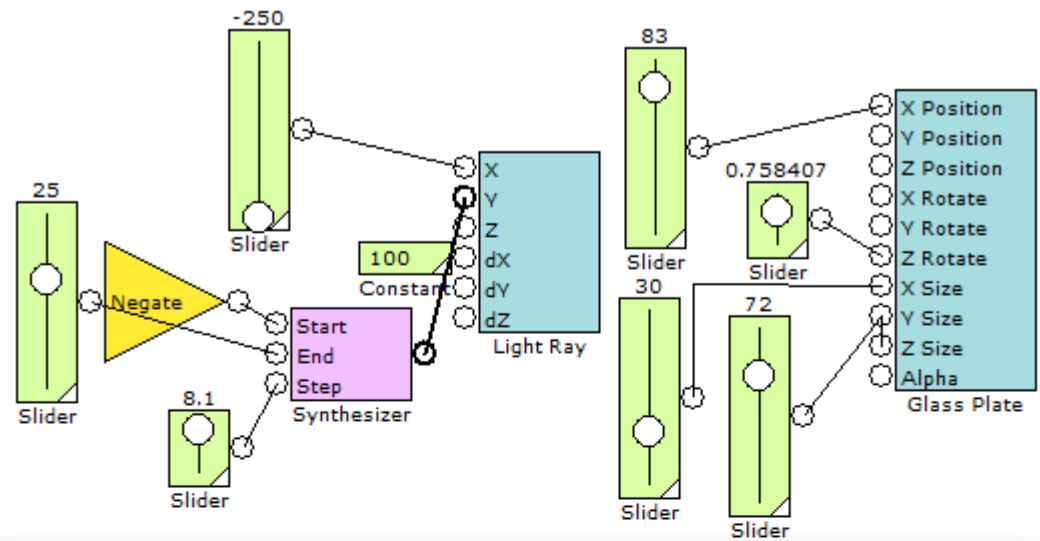
The 2D Glass Plate component passes light rays through a solid piece of glass in the 2D view window. The settings determine the refractive index of the glass. 2D - column 3





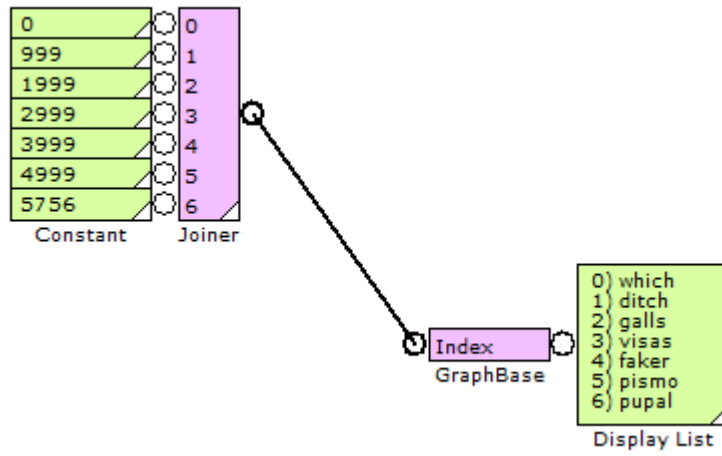
### 3D Glass Plate

The 3D Glass Plate component passes light rays through a solid piece of glass in the 3D view window. The settings determine the refractive index of the glass. The Affine [4x4] array input modifies the size, position and orientation of the glass plate. 3D - column 5



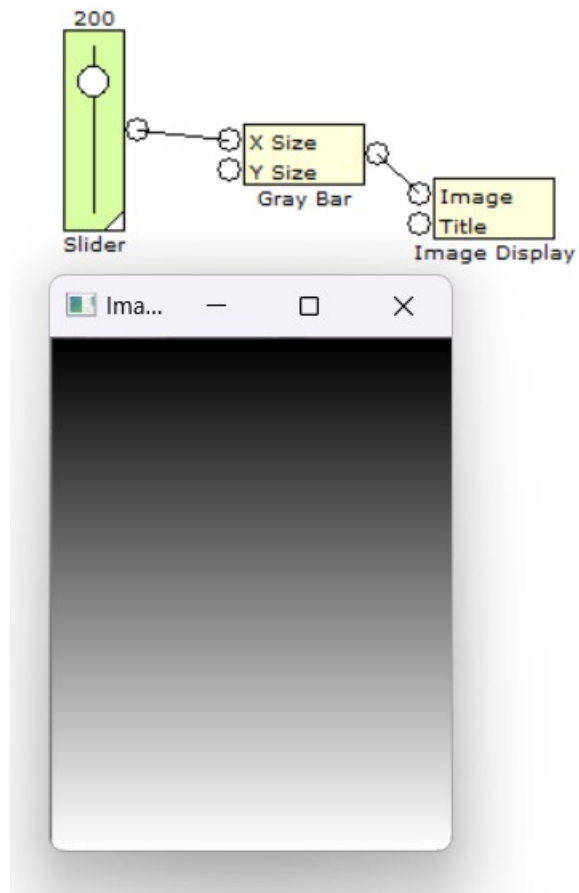
# GraphBase

The GraphBase Words component is a list of 5757 five letter words provided by the Stanford University GraphBase. Arrays - column 7



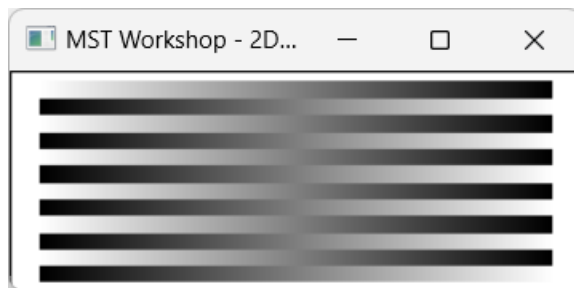
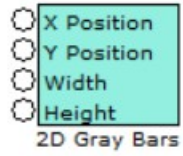
# Gray Bar

The Gray Bar component creates an image with a test pattern of a Grayscale bar. Images – column 1



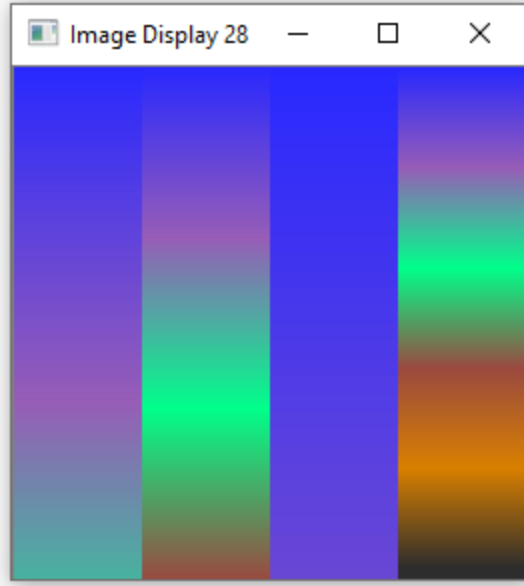
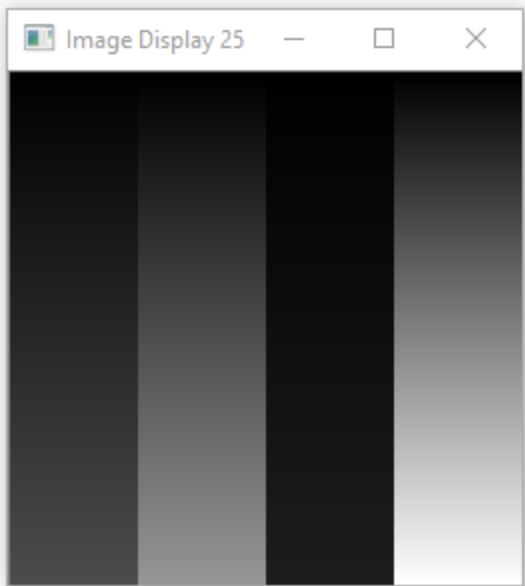
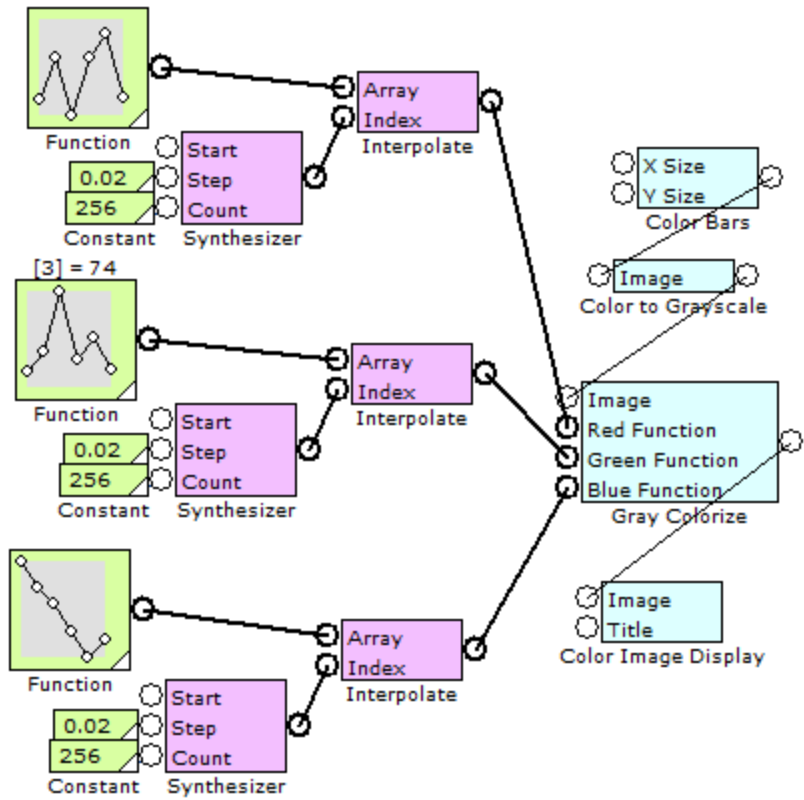
## 2D Gray Bars

The 2D Gray Bars component draws grayscale bars on the 2D view canvas. 2D – column 2



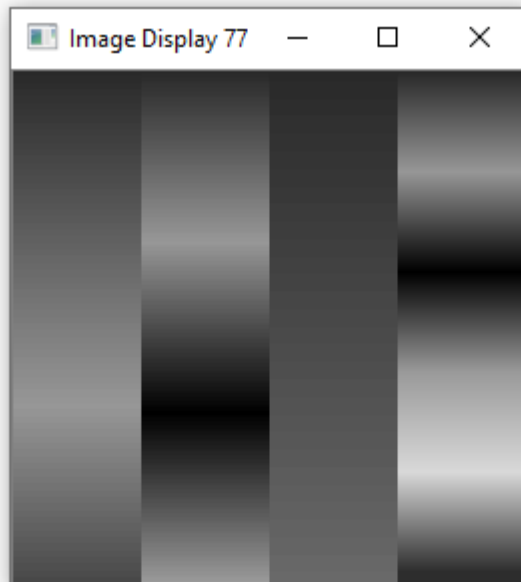
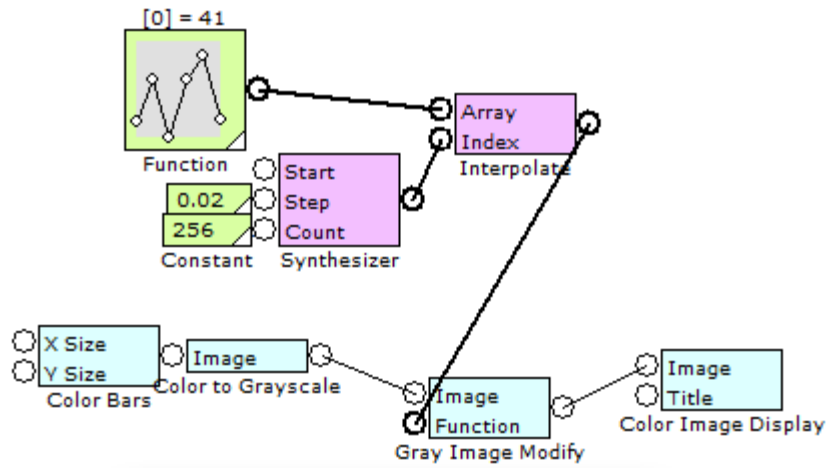
# Gray Colorize

The Gray Colorize component remaps the grayscale image based on the 3 color function inputs. Images - column 2



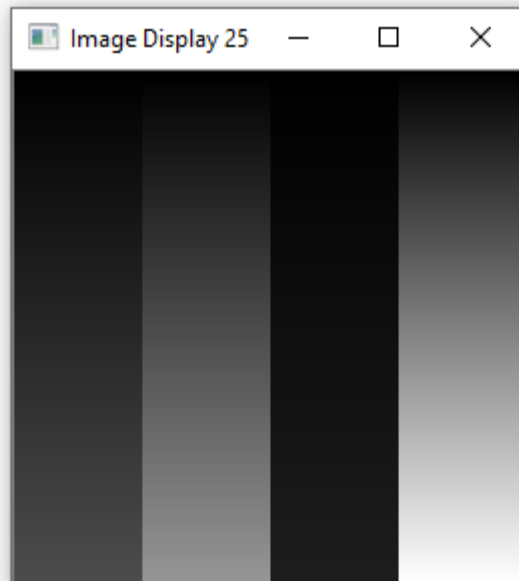
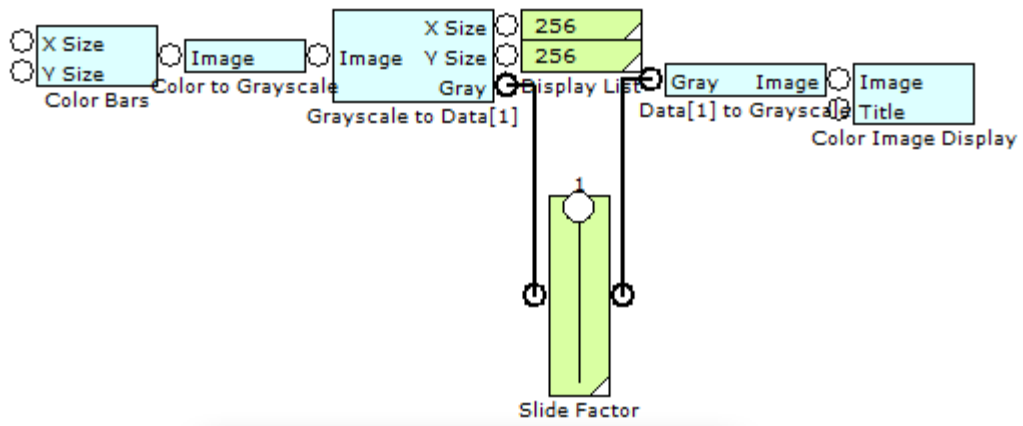
# Gray Image Modify

The Gray Image Modify component remaps the gray data based on the input function. Images - column 2



# Grayscale to Data[1]

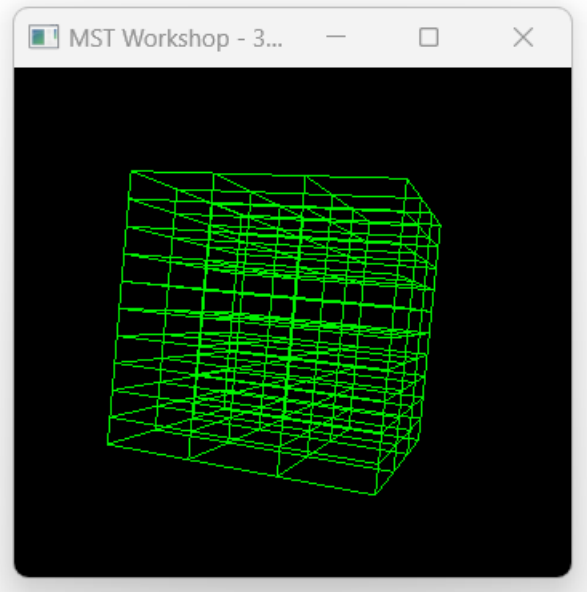
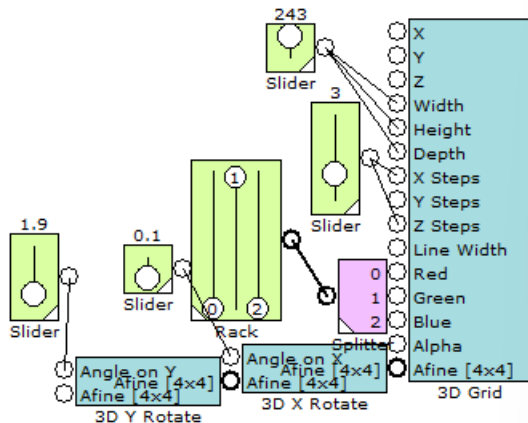
The Grayscale to Data[1] component outputs the grayscale data for the image. Images - column 5



### 3D Grid

The 3D Grid component draws a grid of lines in the 3D view window. The inputs control the size and number of the grid lines. The Affine [4x4] array input modifies the size, position and orientation.

3D - column 4

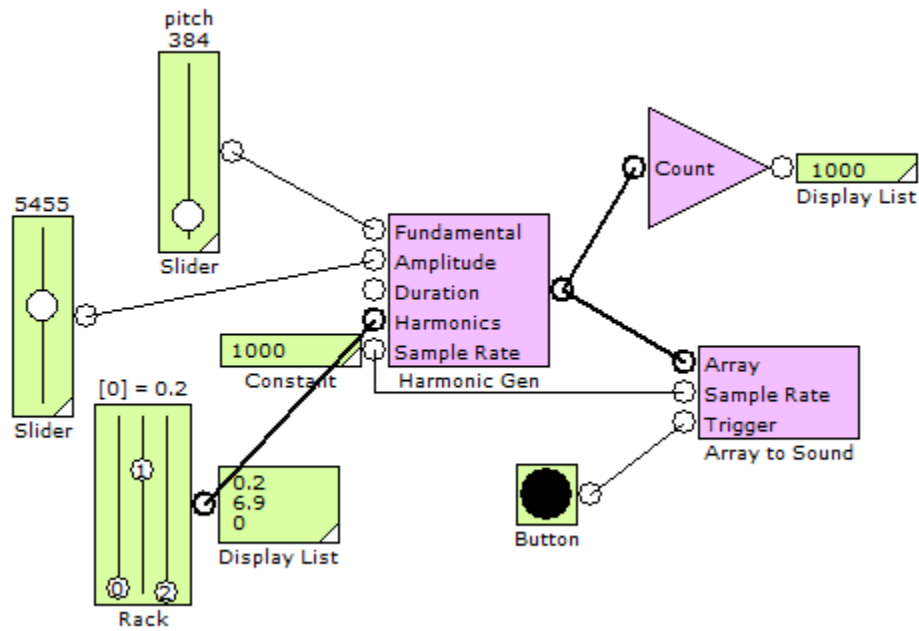




# Harmonic Gen

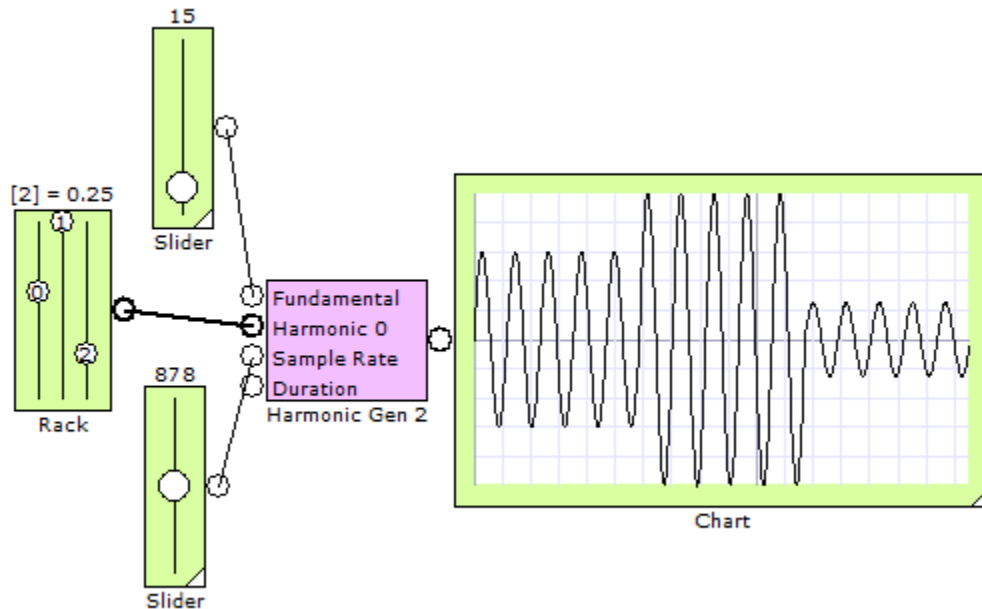
The Harmonic Gen component creates a wave output based on the inputs. The output can be heard by connecting it to an Array to Sound component. The Fundamental, Amplitude, and Duration inputs can be arrays of the same size. This will generate a wave with multiple pitches. The Harmonics input can also be an array of any size, this array should be the relative amplitudes of the desired harmonics.

Arrays - column 7



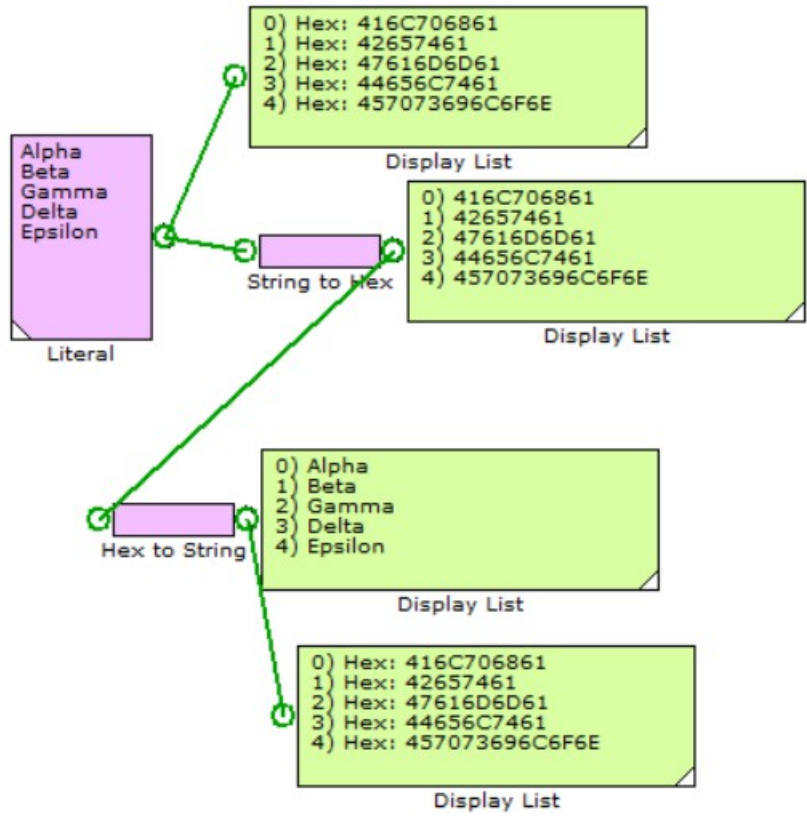
## Harmonic Gen 2

The Harmonic Gen 2 component creates a wave output based on the inputs. If the Fundamental and Harmonic inputs are arrays, the wave will change over the duration specified. The Fundamental input can be an array of any size and will allow the changing of the overall pitch. The Harmonic N inputs are amplitudes of the harmonics, they can also change over time independently. Arrays - column 7



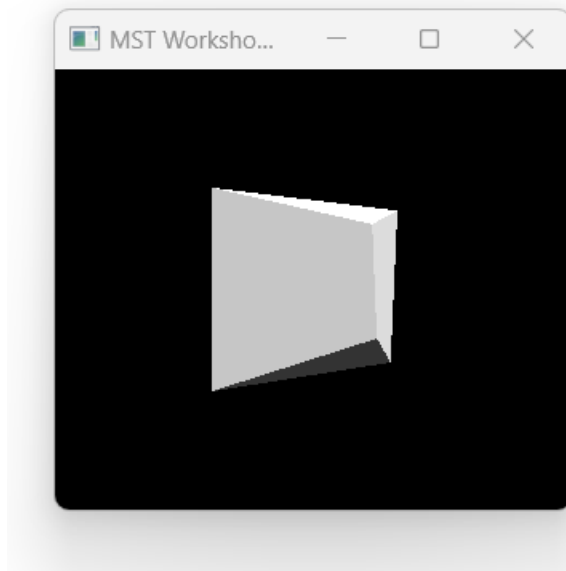
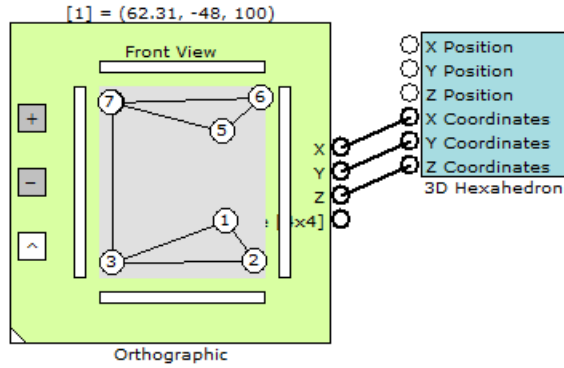
# Hex to String

The Hex to String component converts a string to its Hexadecimal string representation. Arrays – column 3



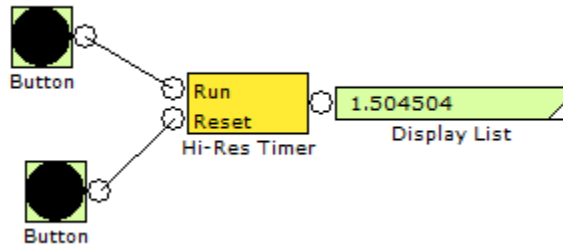
# 3D Hexahedron

The 3D Hexahedron draws a hexahedron in the 3D view window. The X, Y, and Z inputs set the position and size of the hexahedron. The Affine [4x] array input modifies the size, position, and orientation. 3D – column 4



# Hi-Res Timer

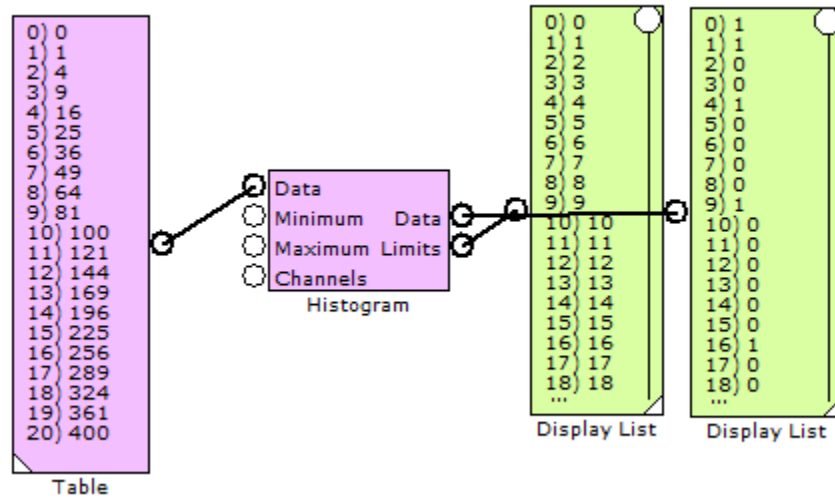
The Hi-Res Timer component starts when the Run input goes high and runs until the Run input goes low. It is reset when the Reset input goes high. This timer runs in real-time and quietly continues when the local time is stopped or reset. Functions - column 7



# Histogram

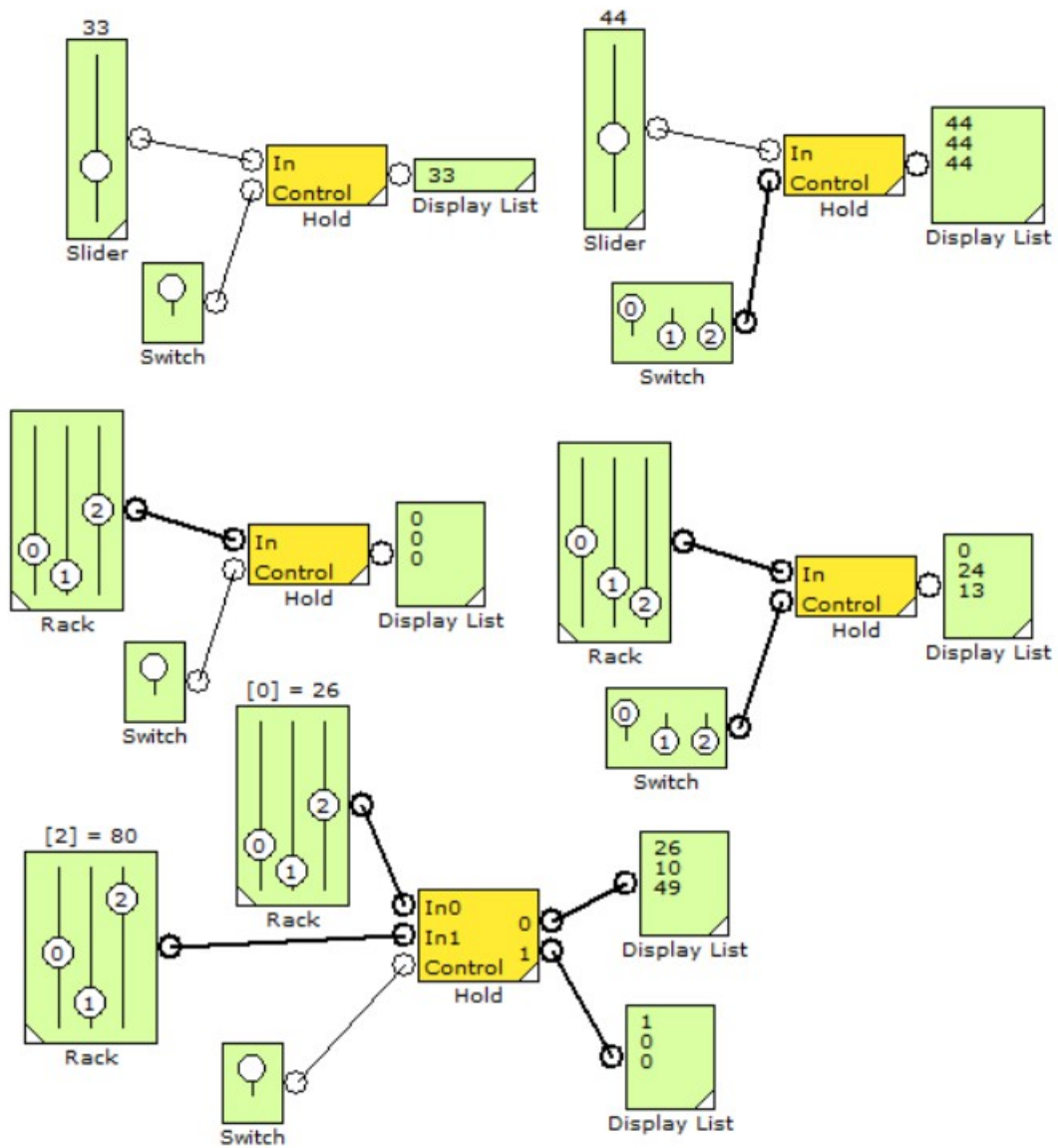
The Histogram component scans the input array and outputs a histogram of the data. column 4

Arrays -



# Hold

The Hold will hold the input at the output when the control input is true. Functions - column 2



## 2D Horizontal Axis

The 2D Horizontal Axis component draws labeled axes on the 2D canvas.

2D - column 1

2D Horizontal Axis

Arrow Size:

Small Tic Size:

Large Tic Size:

Labels on Top

Show Zero Label

Arrow Head

Ignore Display Scale

Pin options:

X

Size

Start  End

Tic  Label

Line Red  Line Green  Line Blue

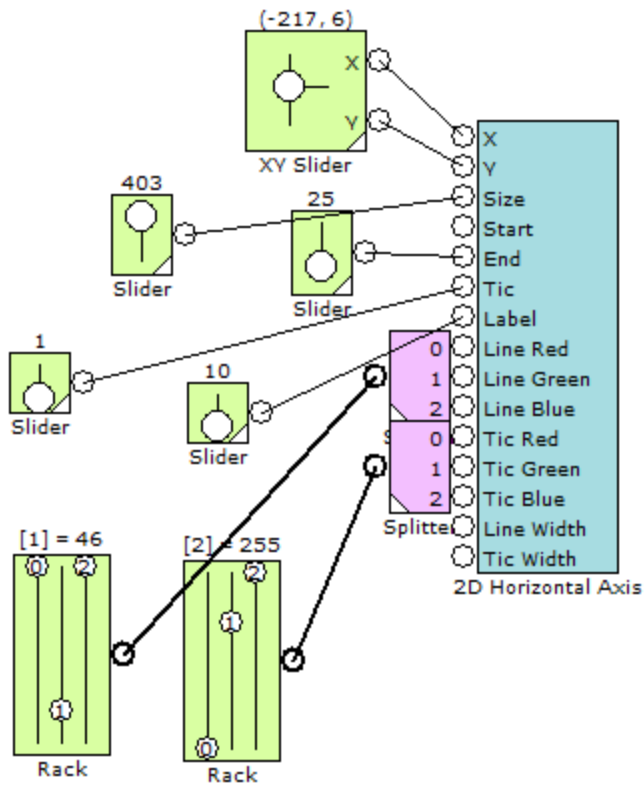
Tic Red  Tic Green  Tic Blue

Line Width  Tic Width

Label Red  Label Green  Label Blue

Alpha

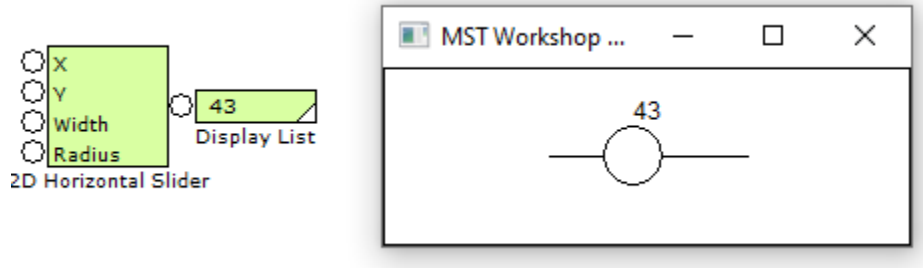
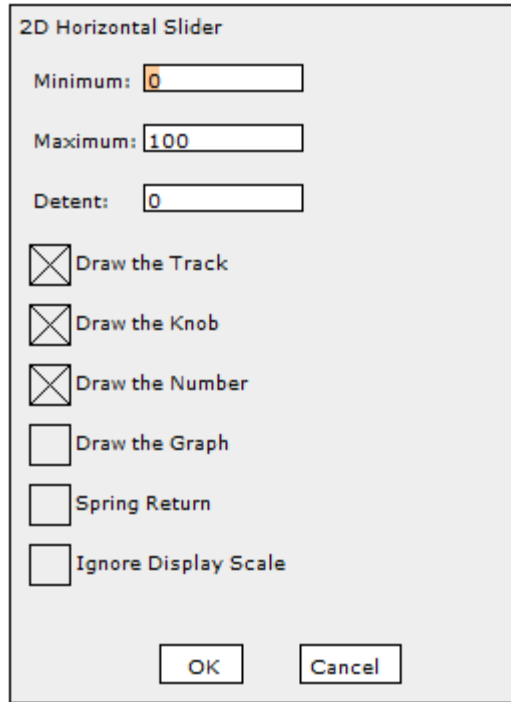




# 2D Horizontal Slider

The 2D Horizontal Slider component outputs a number based on its dial position. column 2

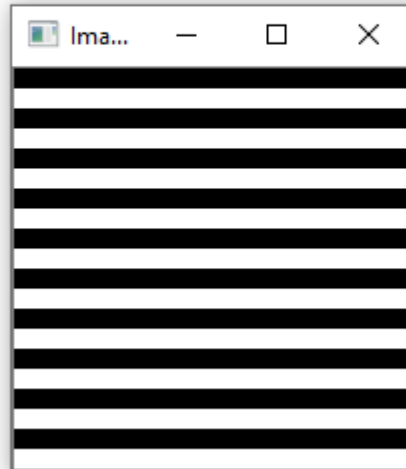
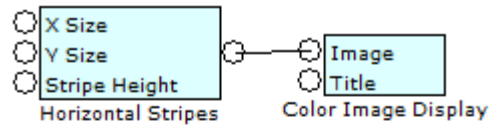
Controls -



# Horizontal Stripes

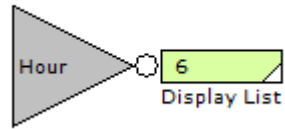
The Horizontal Stripes component creates an image with a horizontal striped test pattern.  
- column 1

Images



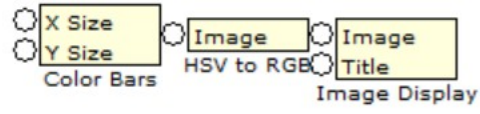
# Hour

The Hour component outputs the hour from the computers internal clock. Functions - column 7



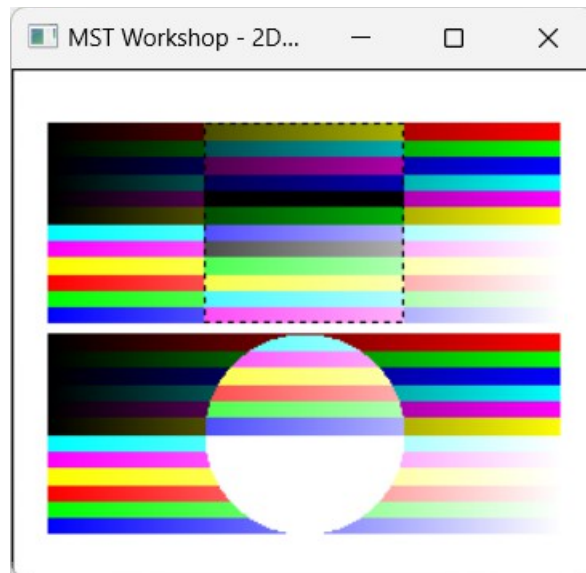
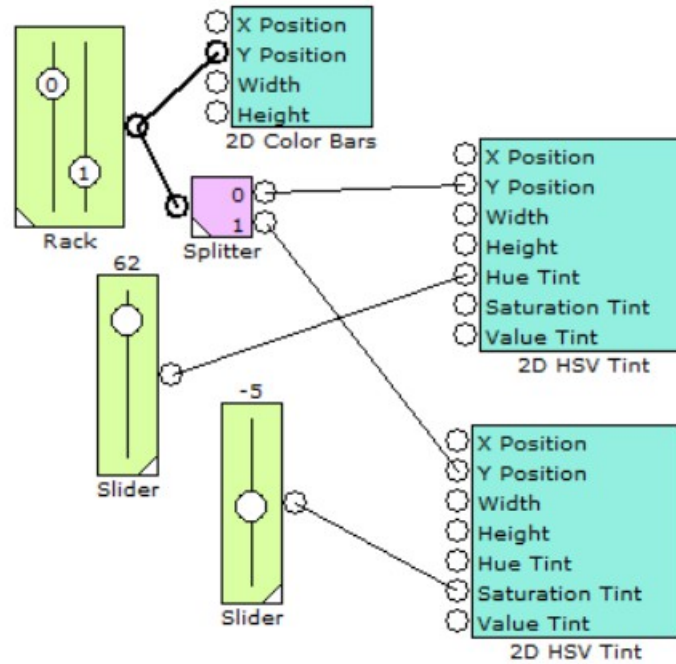
# HSV to RGB

The HSV to RGB component converts the image from HSV color to RGB color. Images - column 1



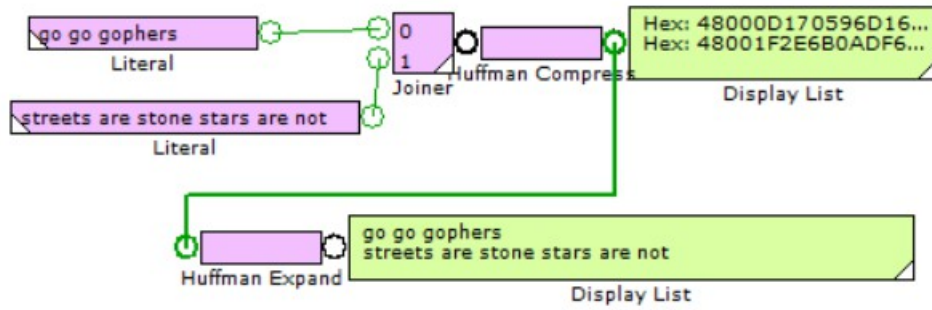
## 2D HSV Tint

The 2D HSV Tint lightens or darkens each of the Hue Saturation and Value of the components in the 2D drawing it covers. 2D – column 1



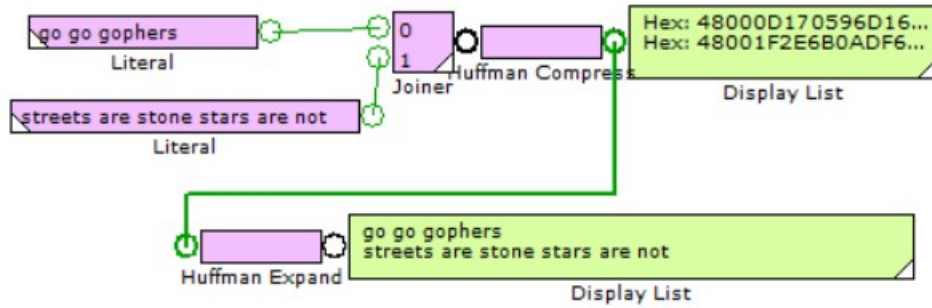
# Huffman Compress

The Huffman Compress component uses Huffman compression to compress a string into compressed bits.



# Huffman Expand

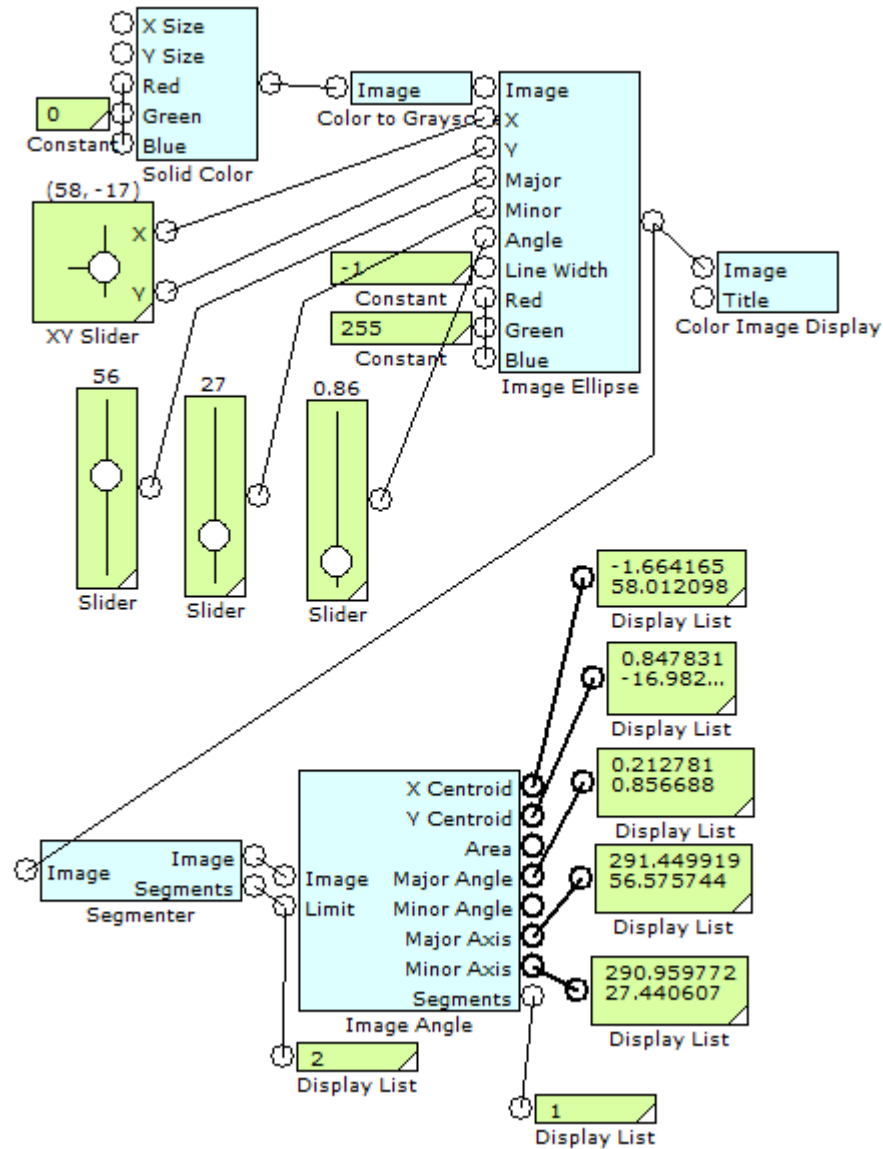
The Huffman Expand component uses Huffman decompression to return compressed bits into a string.





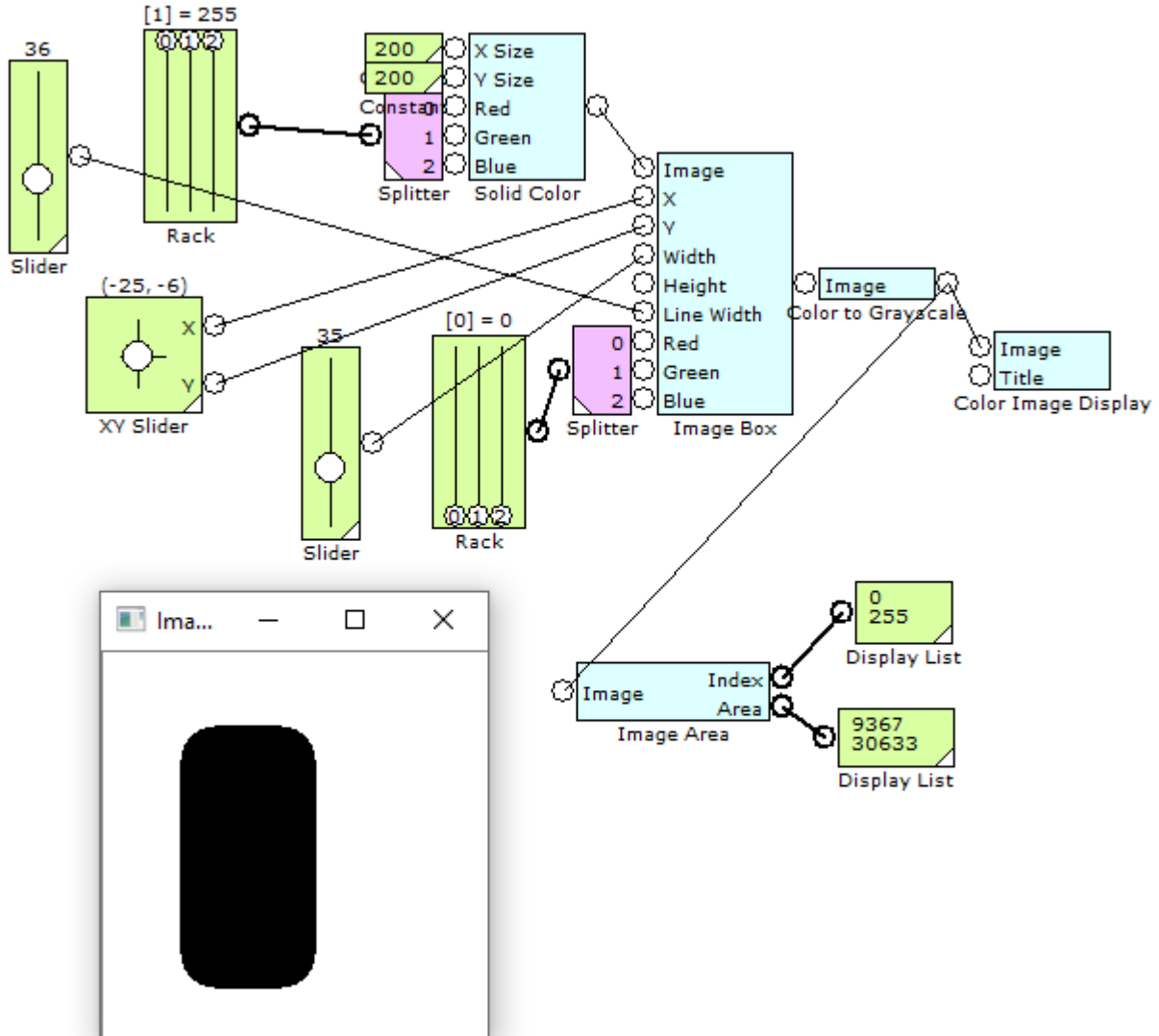
# Image Angle

The Image Angle component inputs a grayscale image and returns the angles and sizes of the major and minor axes of the object in the image. Images - column 4



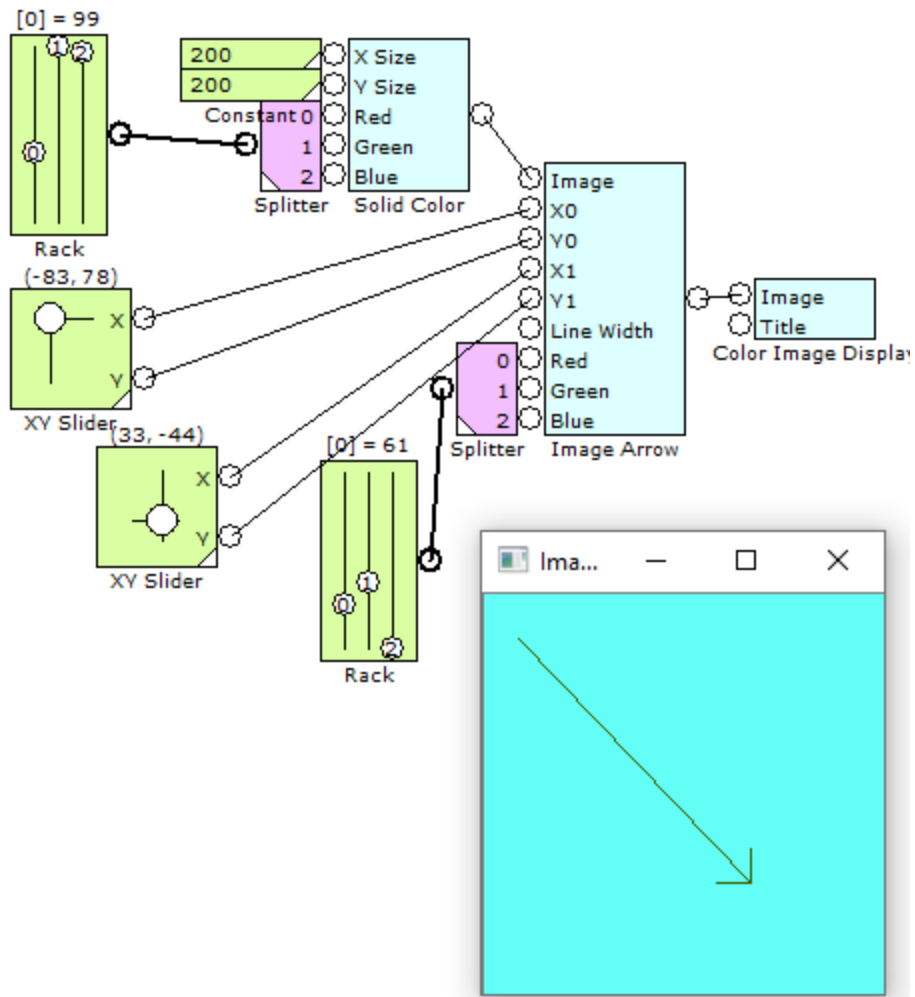
# Image Area

The Image Area component inputs a grayscale image and returns the area of the object in the image.  
 Images - column 4



# Image Arrow

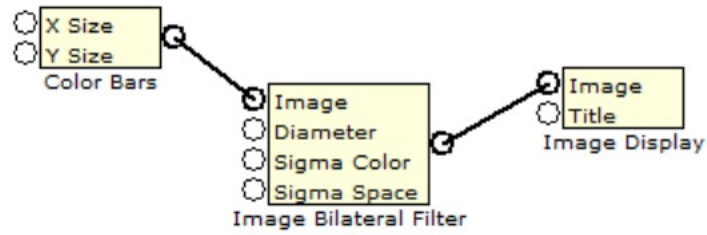
The Image Arrow component draws an arrow on an image. Images - column 3



# Image Bilateral Filter

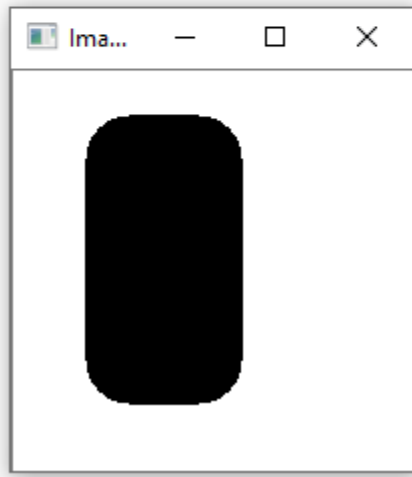
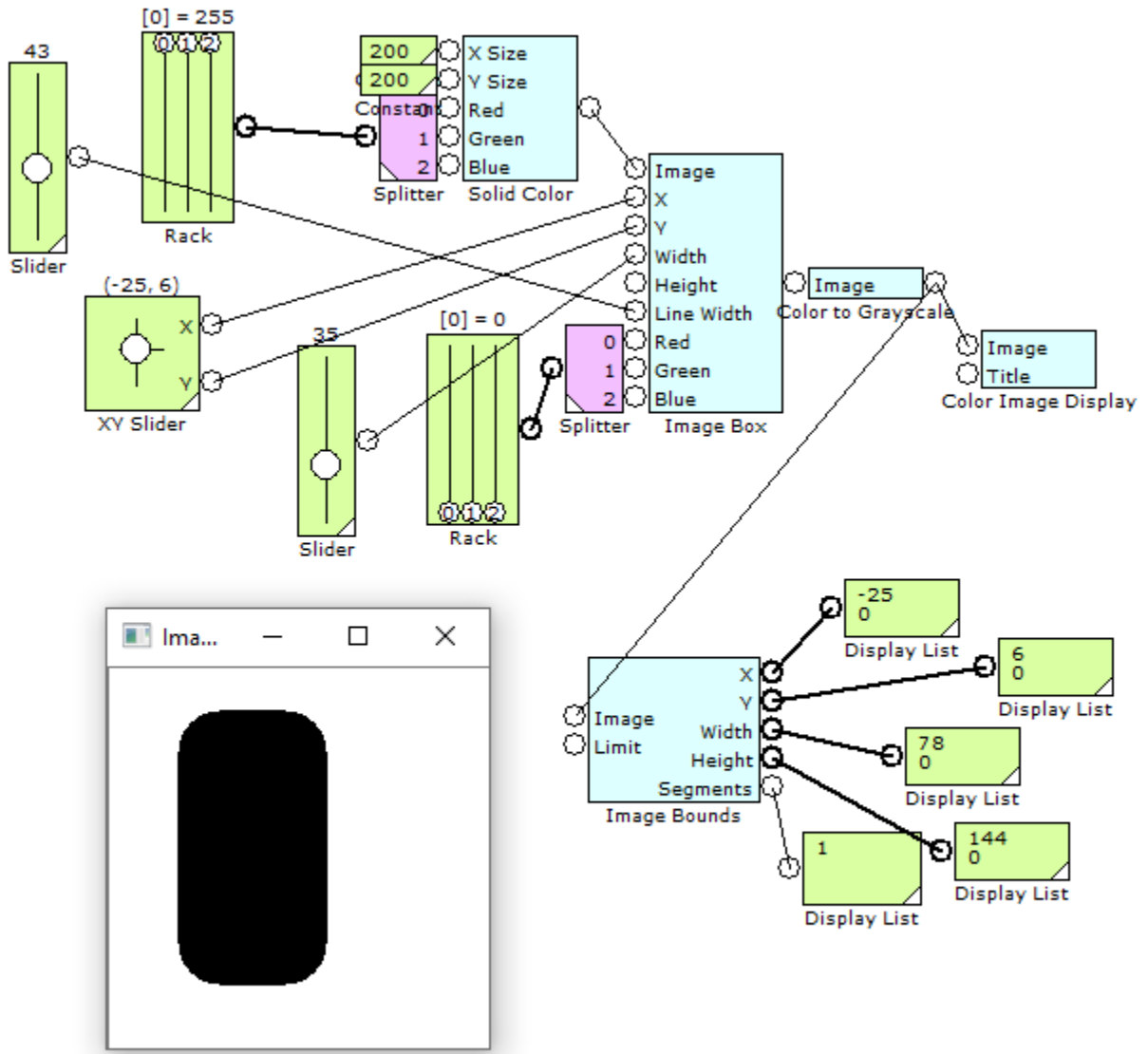
The Image Bilateral Filter component smooths an image based on a bilateral filter.

Images -  
column 2



# Image Bounds

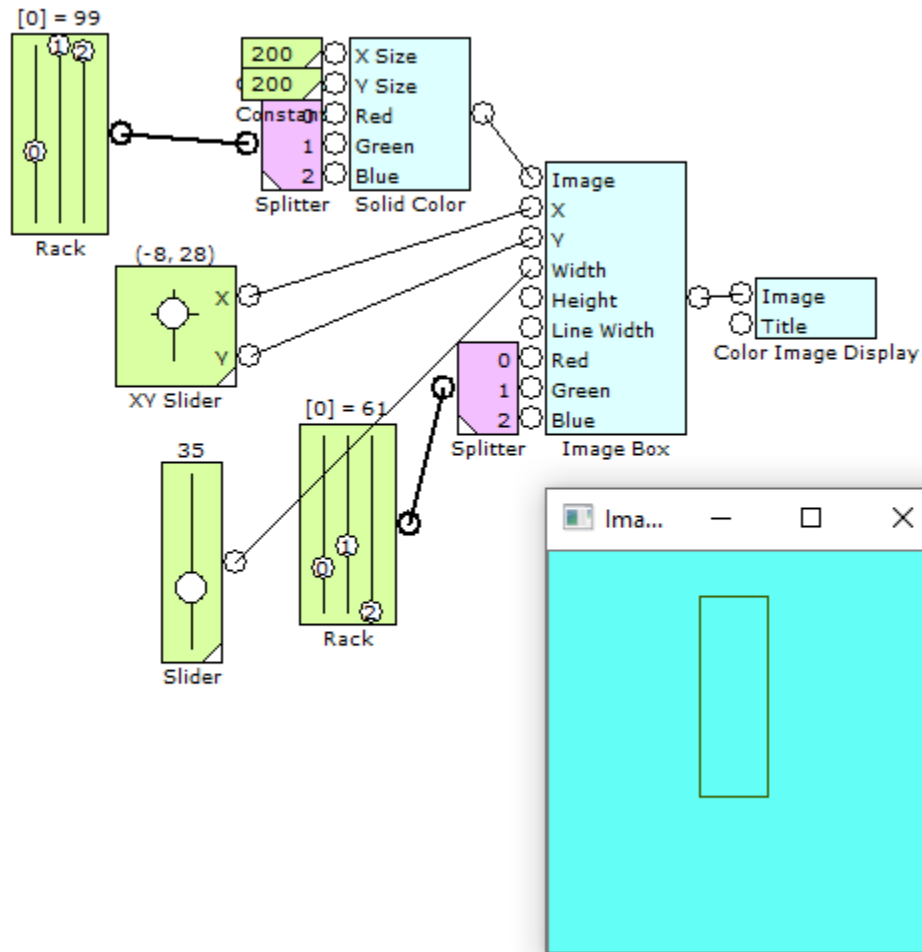
The Image Bounds component scans a grayscale image and returns the bounding box for pixels above the threshold. Images - column 4



# Image Box

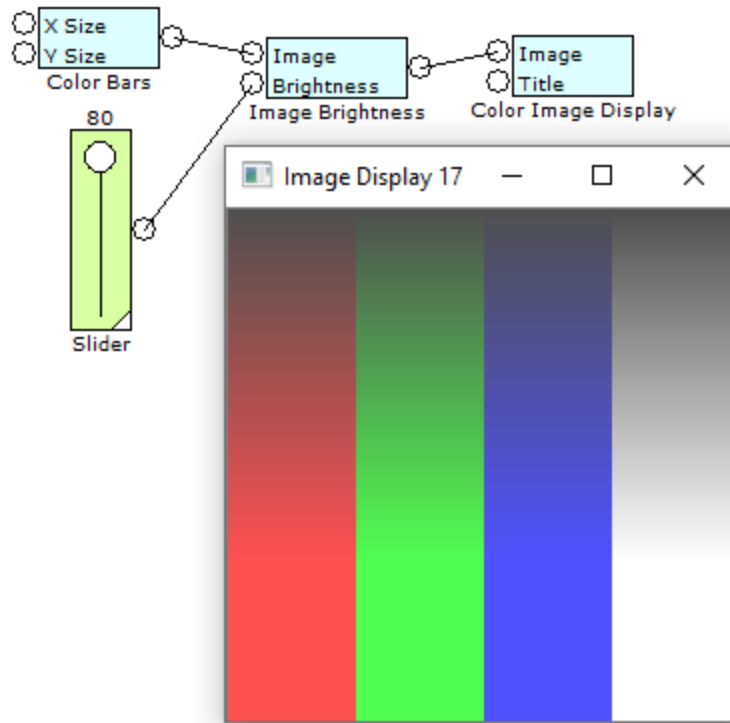
The Image Box component draws a box on an image.

Images - column 3



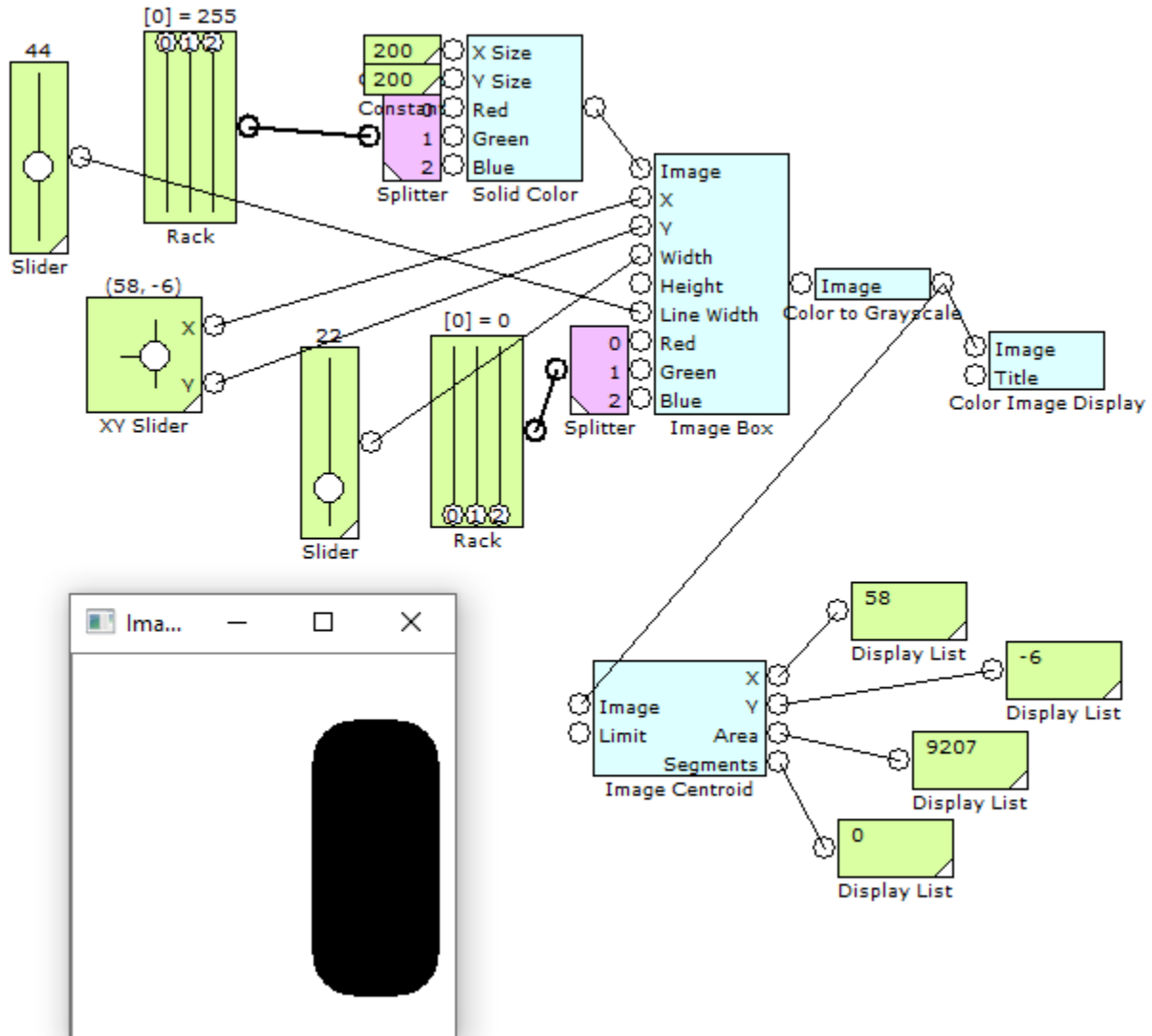
# Image Brightness

The Image Brightness component changes the brightness of an image. Images - column 2



# Image Centroid

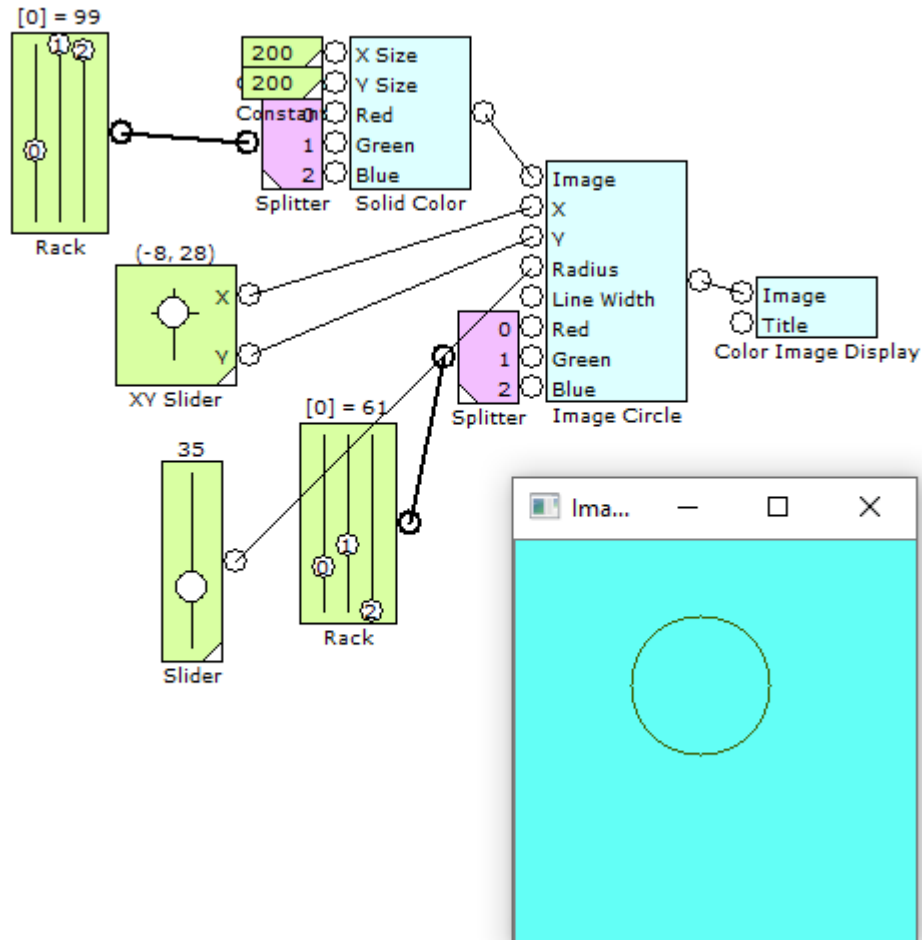
The Image Centroid component inputs a grayscale image and returns the x and y centroids of the object in the image. Images - column 4





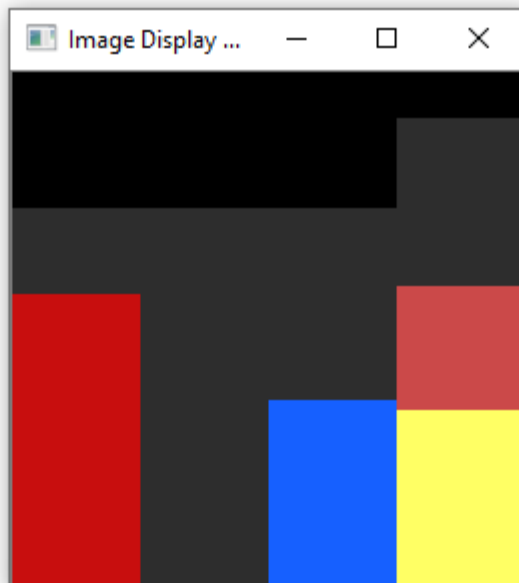
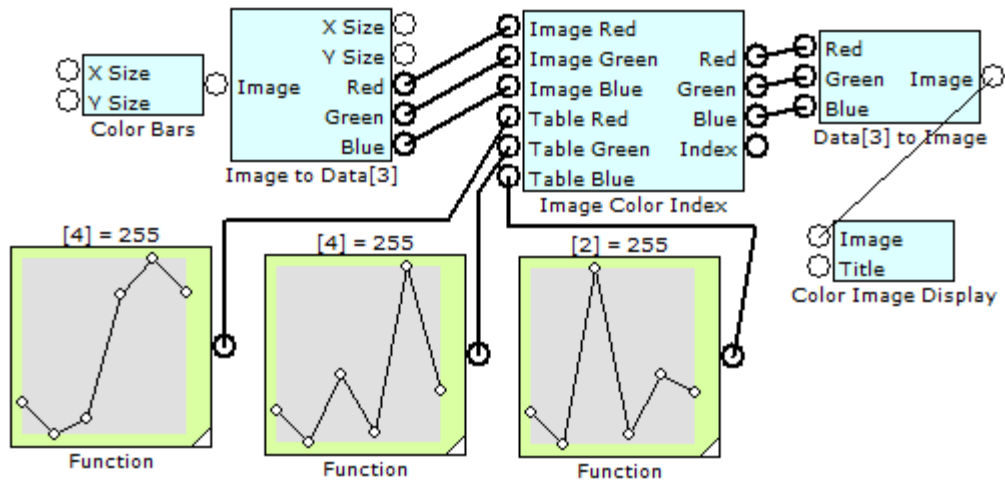
# Image Circle

The Image Circle component draws a circle on an image. Set the Line Width to a negative number to fill the circle. Images - column 3



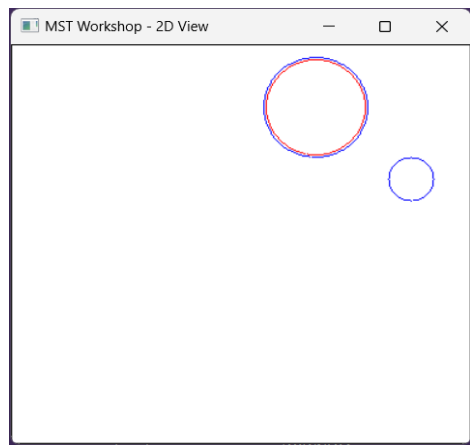
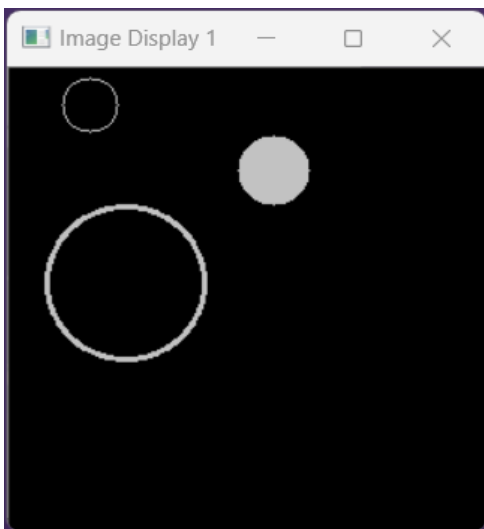
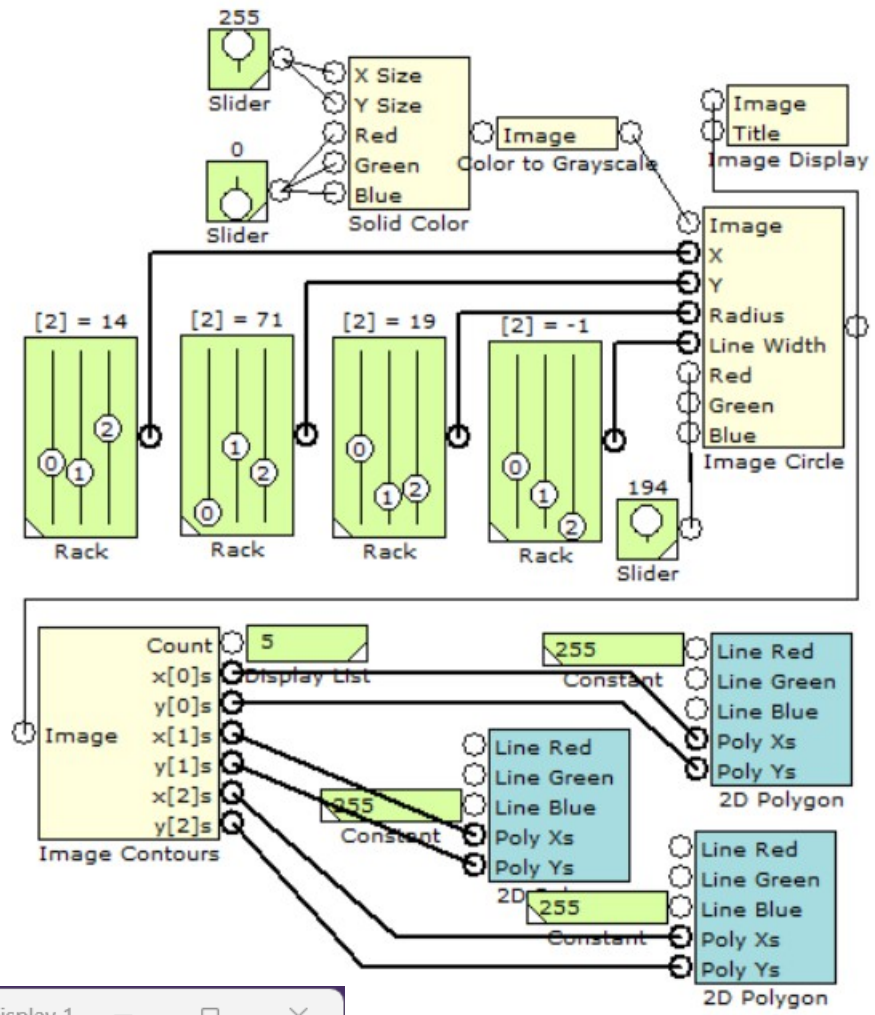
# Image Color Index

The Image Color Index searches the table of colors for the closest match to each pixel in the image.  
 Images - column 5



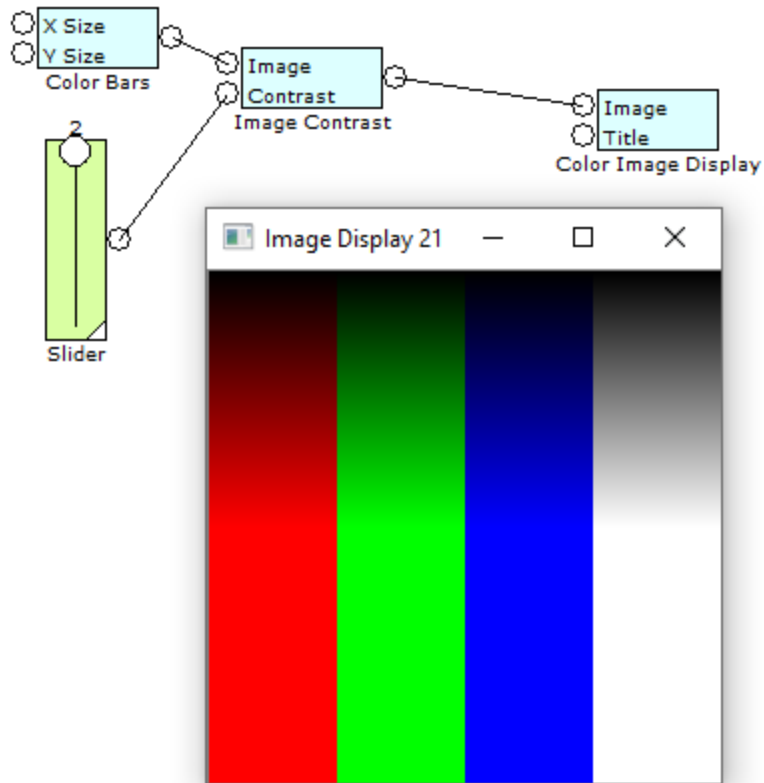
# Image Contours

The Image Contours component inputs a grayscale image and returns the contours. Images – column 4



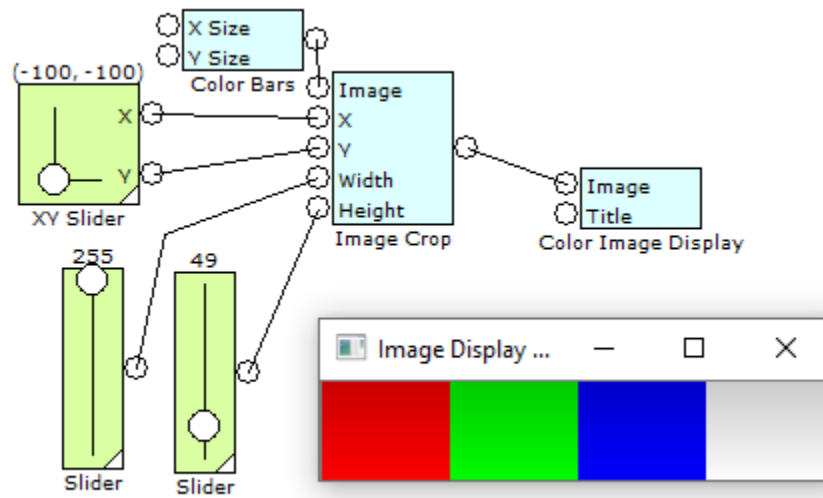
# Image Contrast

The Image Contrast component changes the contrast of an image. Images - column 2



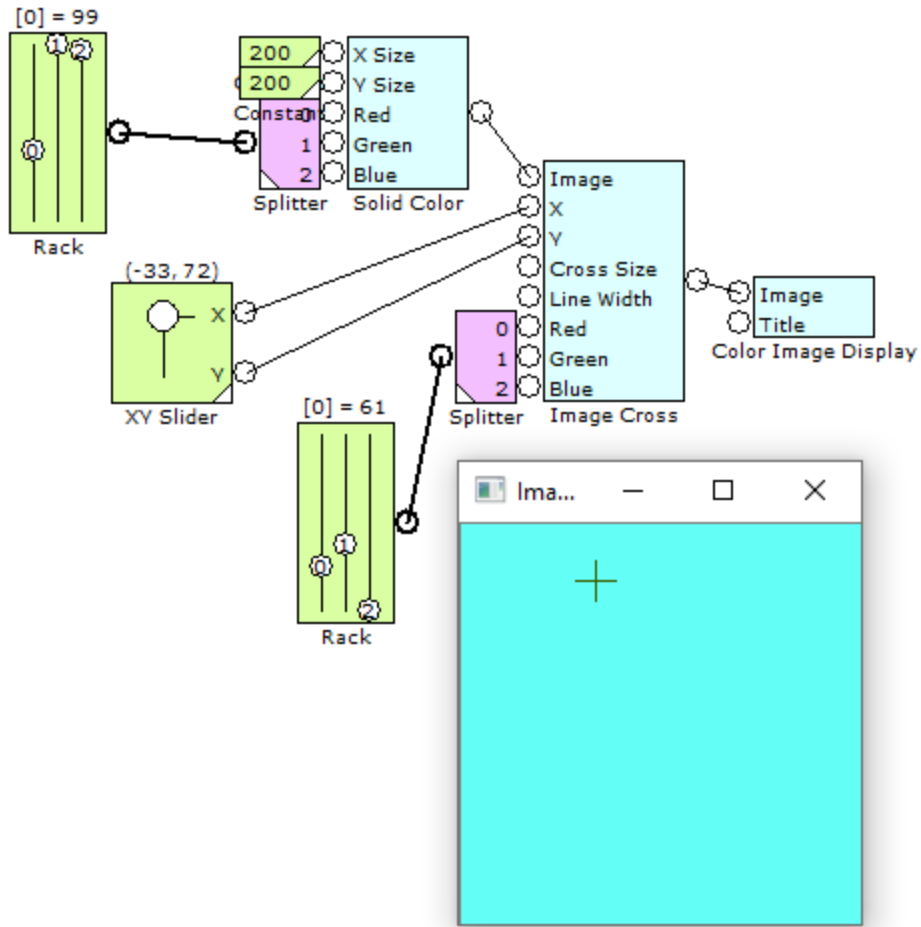
# Image Crop

The Image Crop component crops the image at the specified offset to the desired size. Images - column 1



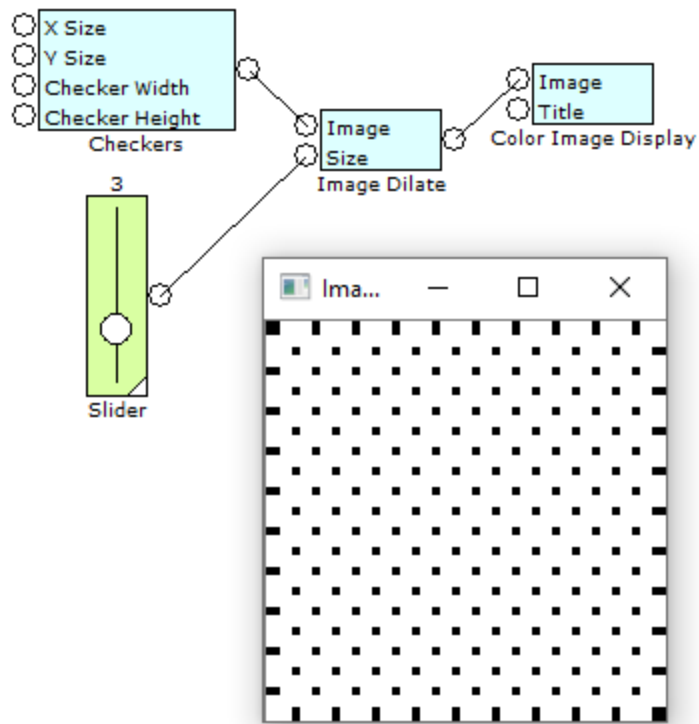
# Image Cross

The Image Cross component draws a cross on an image. Images - column 3



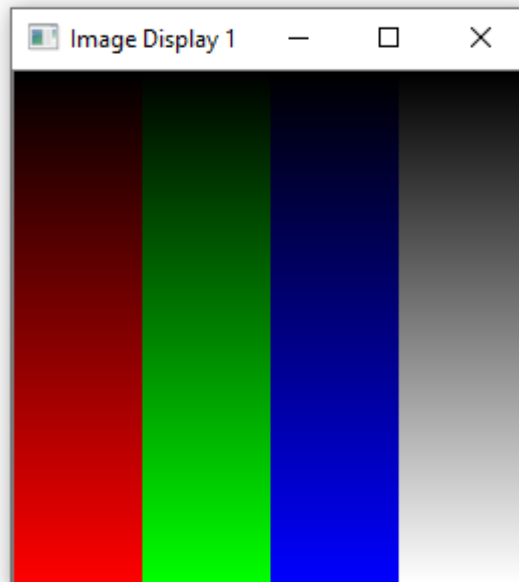
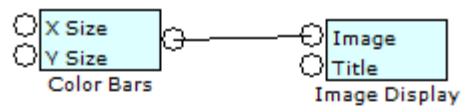
# Image Dilate

The Image Dilate component grows darker areas in an image. Images - column 2



# Image Display

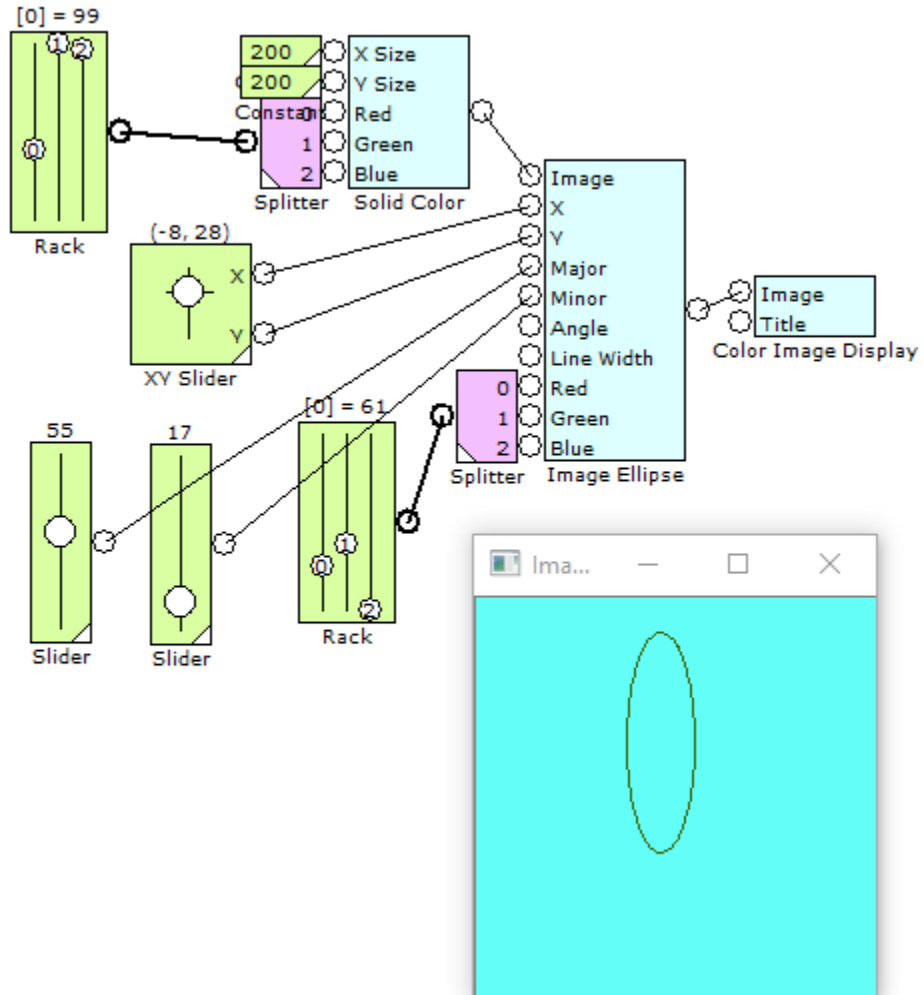
The Image Display component shows an image preview in color. Images - column 5





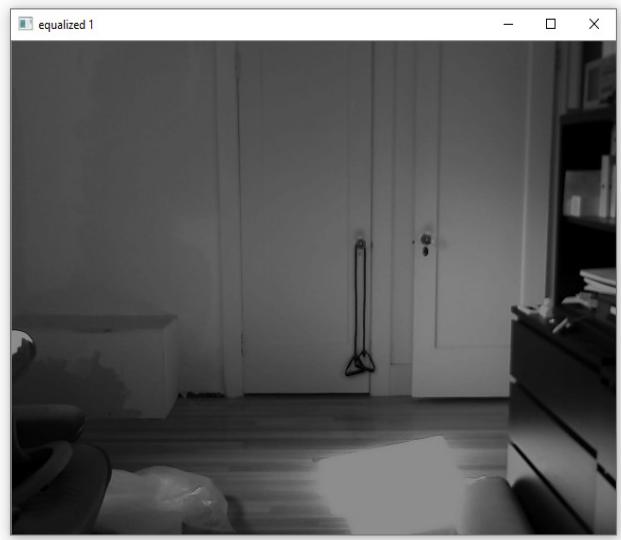
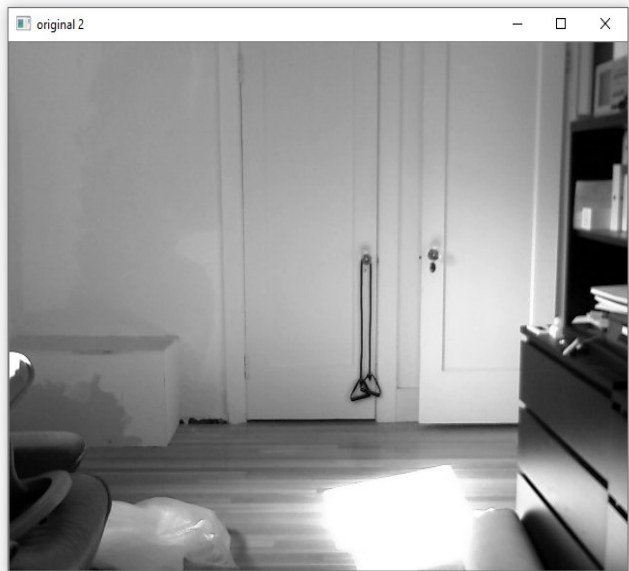
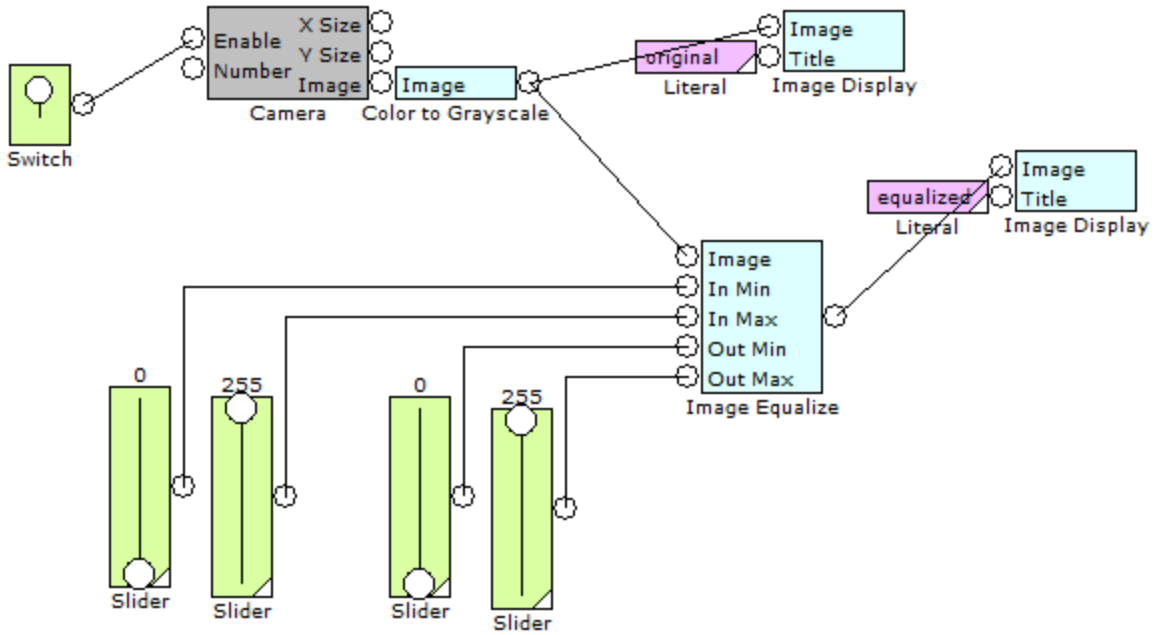
# Image Ellipse

The Image Ellipse component draws an ellipse on an image. Set the Line Width to a negative number to fill the ellipse. Images - column 3



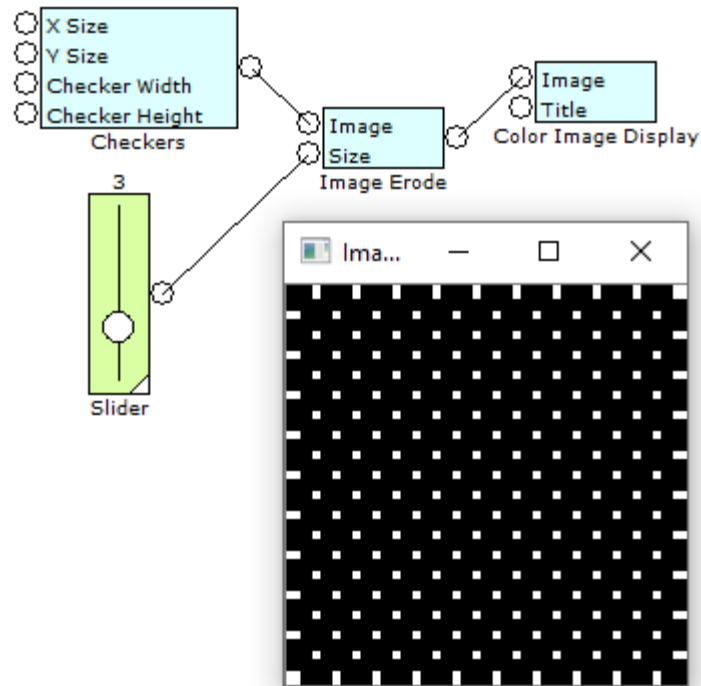
# Image Equalize

The Image Equalize component spreads the levels equally over the image based on the number of times they are used in the image. Images - column 3



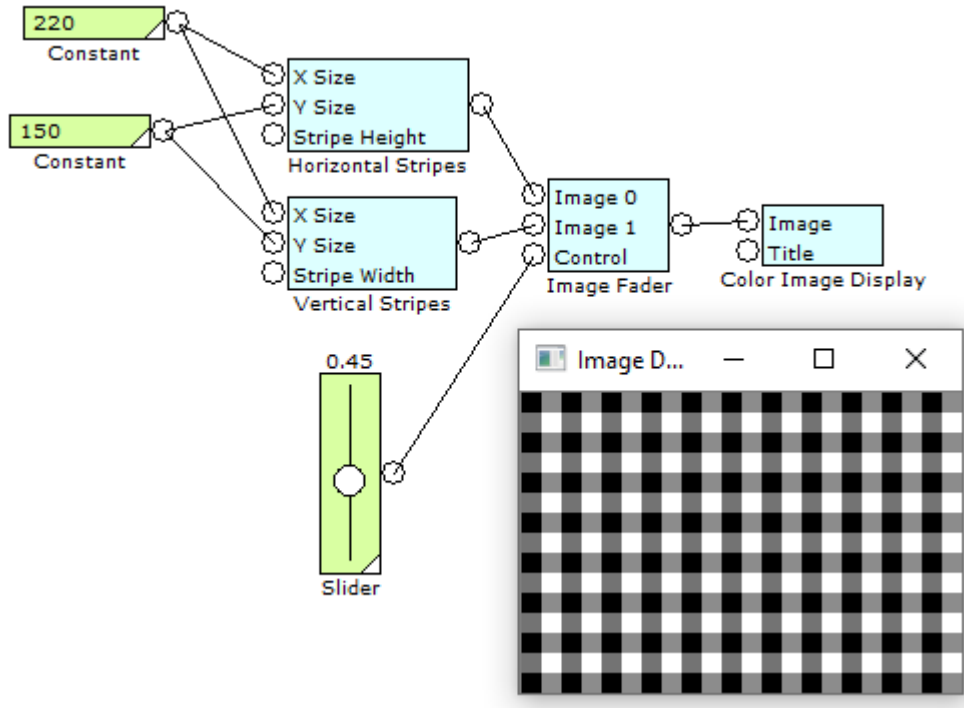
# Image Erode

The Image Erode component grows lighter areas in an image. Images - column 2



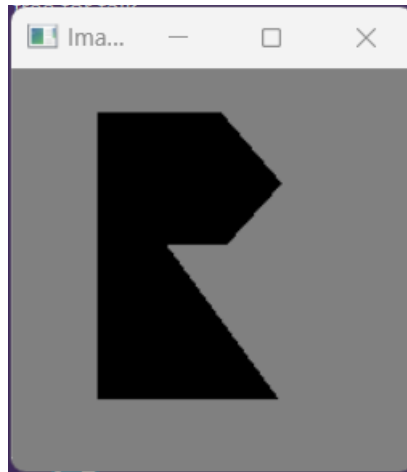
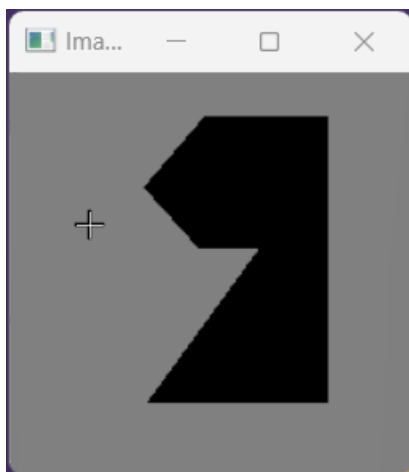
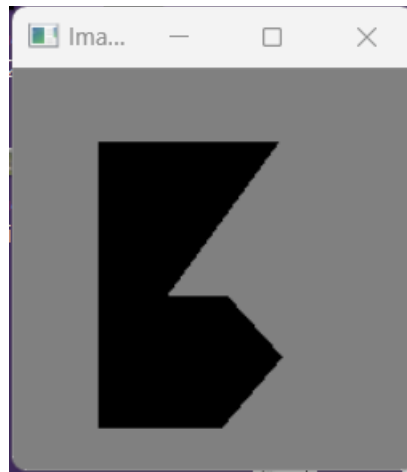
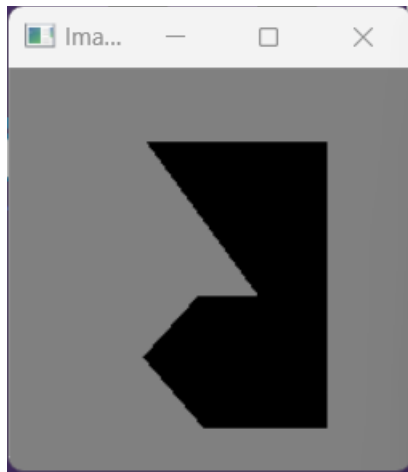
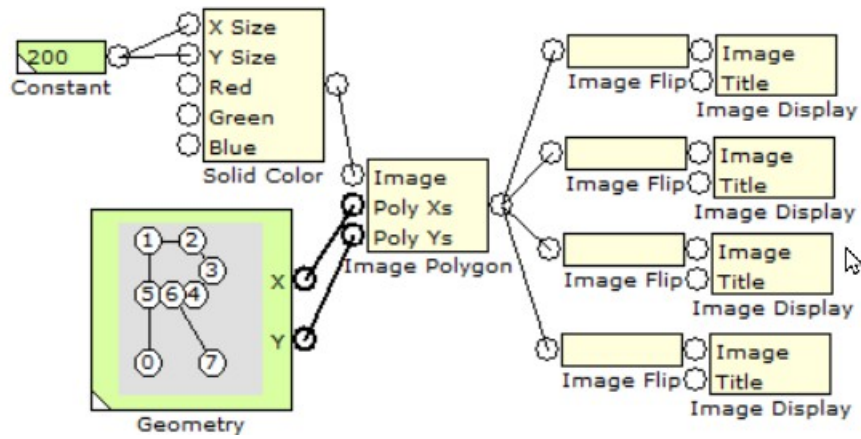
# Image Fader

The Image Fader component creates an image by fading between the 2 input images based on the Control input. The Control value is between 0.0 and 1.0. Images - column 3



# Image Flip

The Image Flip component flips the image either horizontally, vertically, or both. Images – column 1



# Image Gaussian

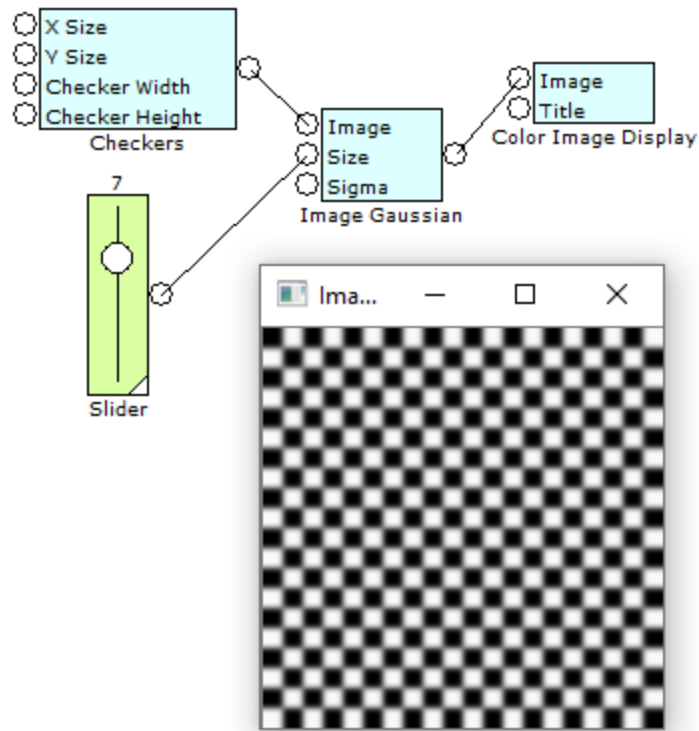
The Image Gaussian component smooths an image using a two dimensional Gaussian filter.

The Size input is the size of a two dimensional filter and can range from 3 to 31 and has a default of 7.

The Sigma input is the intensity of the filtering and has a default of 1.5.

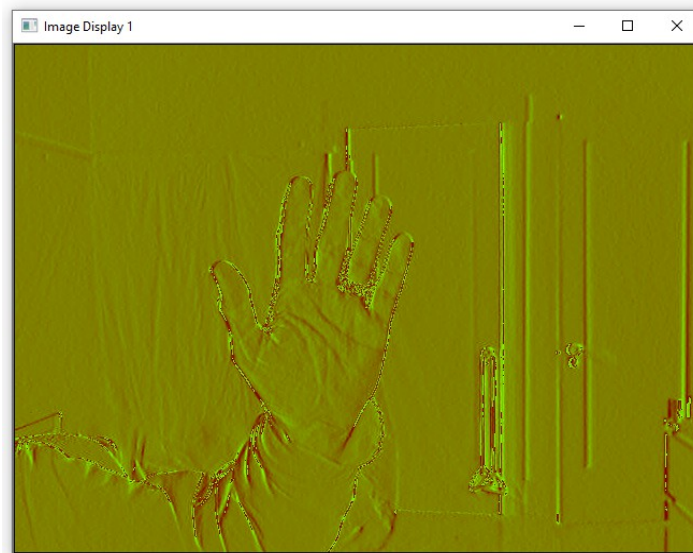
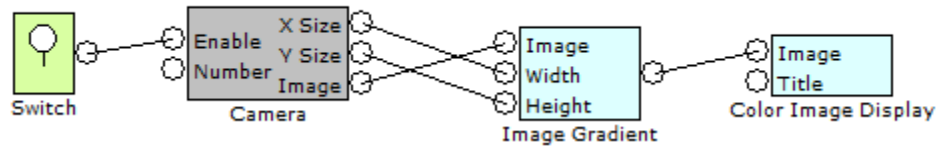
The Size and Sigma inputs can be a single value or a pair to give different values in x and y.

Images - column 2



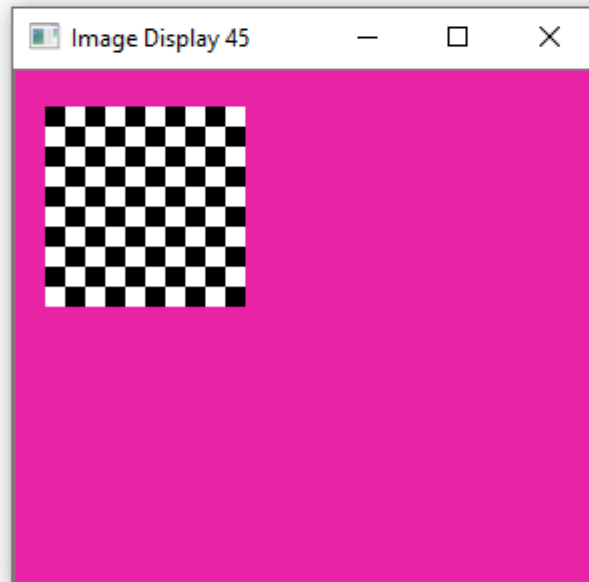
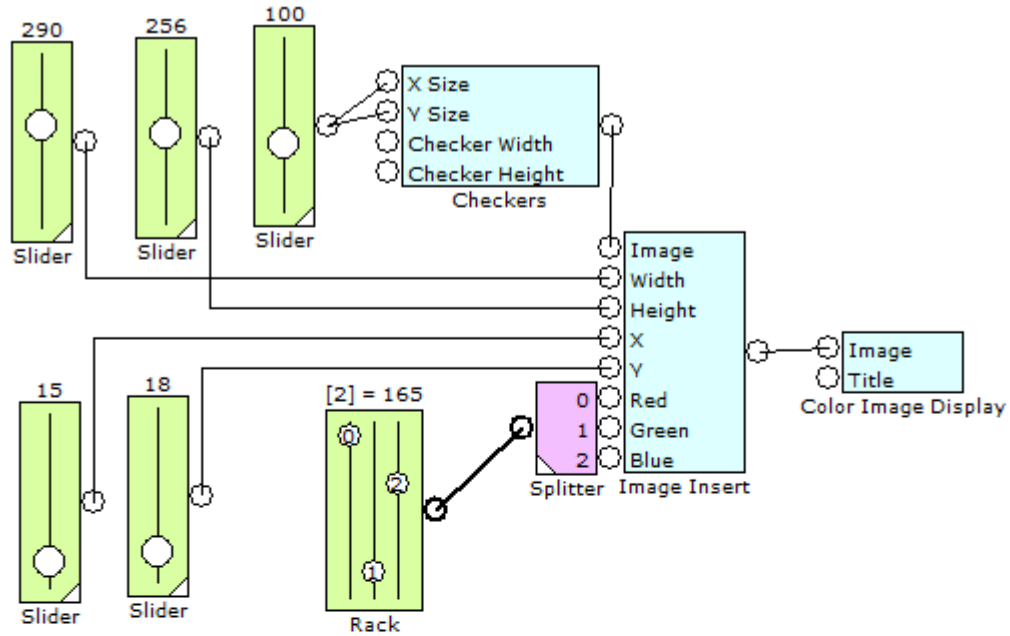
# Image Gradient

The Image Gradient component finds gradients in x and y and creates an image where red is the x gradient and green is the y gradient. Images - column 2



# Image Insert

The Image Insert component inserts the image at the specified offset into an otherwise blank image.  
Images - column 1

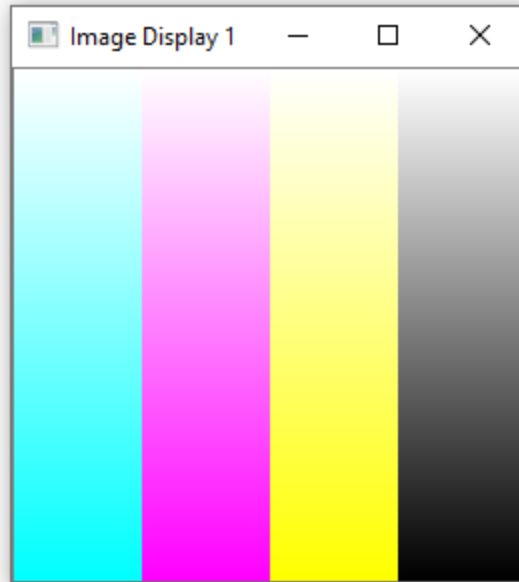
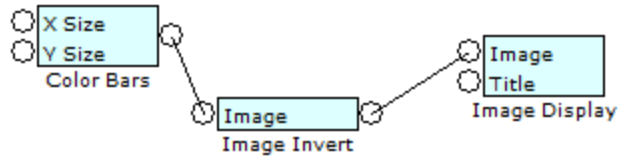




# Image Invert

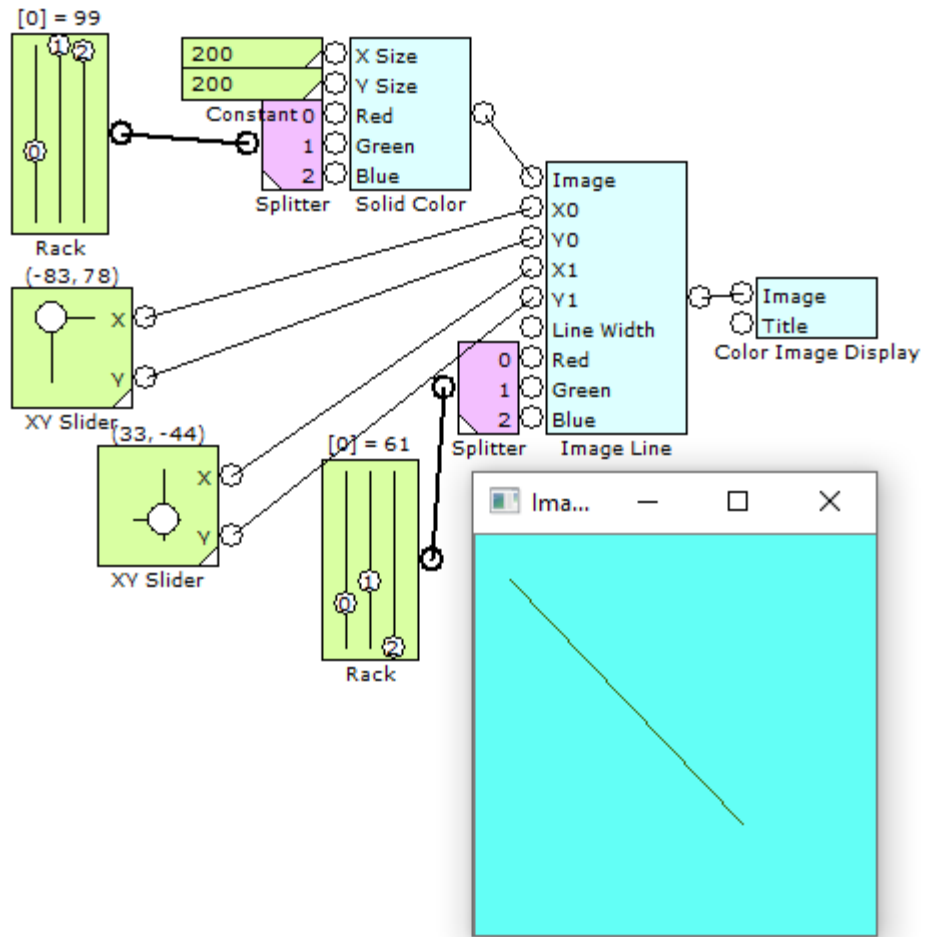
The Image Invert component creates a negative of an image.

Images - column 2



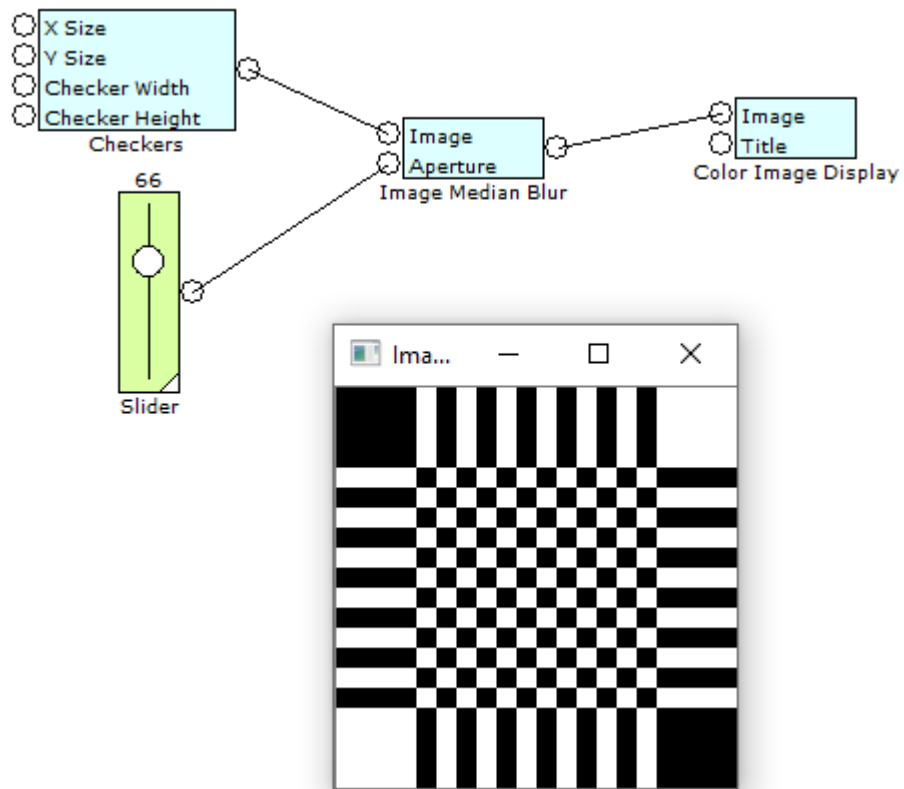
# Image Line

The Image Line component draws a line on an image. Images - column 3



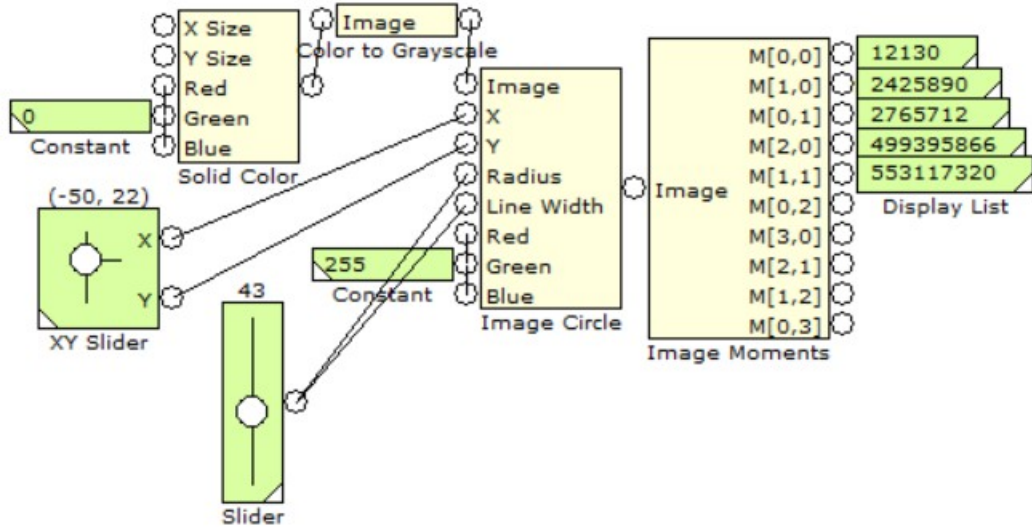
## Image Median Blur

The Image Median Blur component smooths an image based on a median filter. The aperture must be greater than 1 and less than 102. Images - column 2



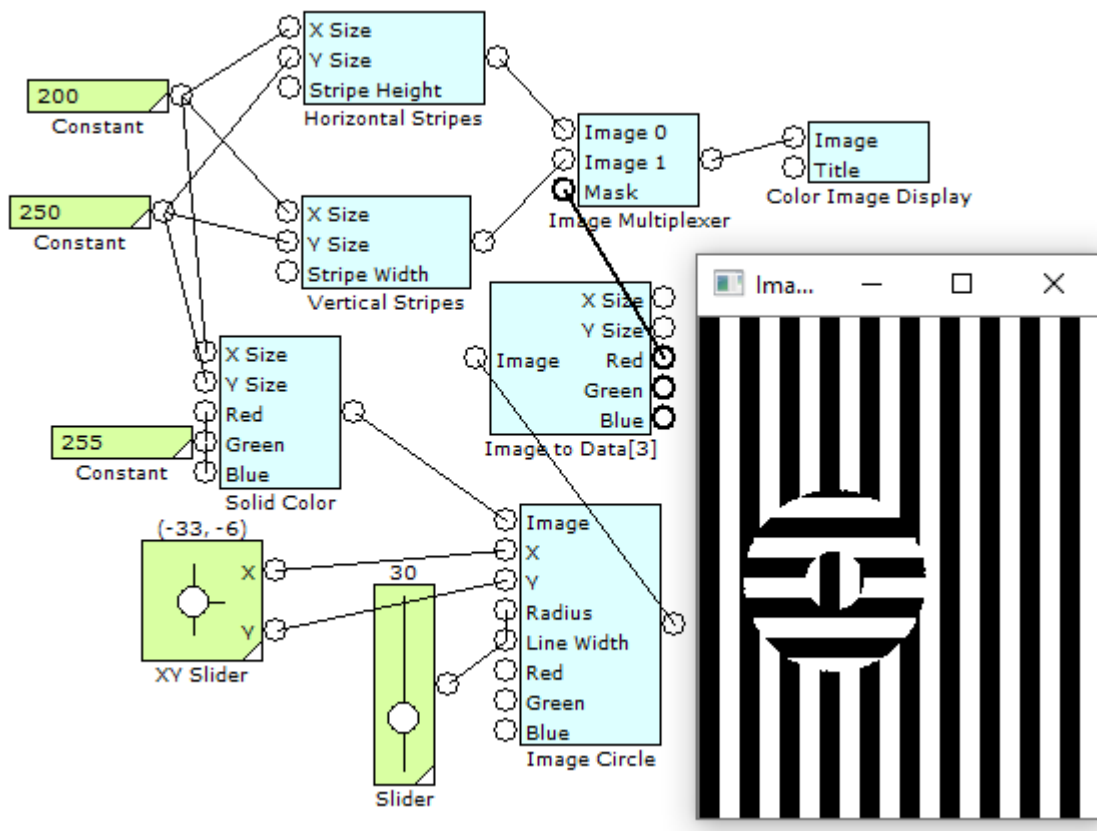
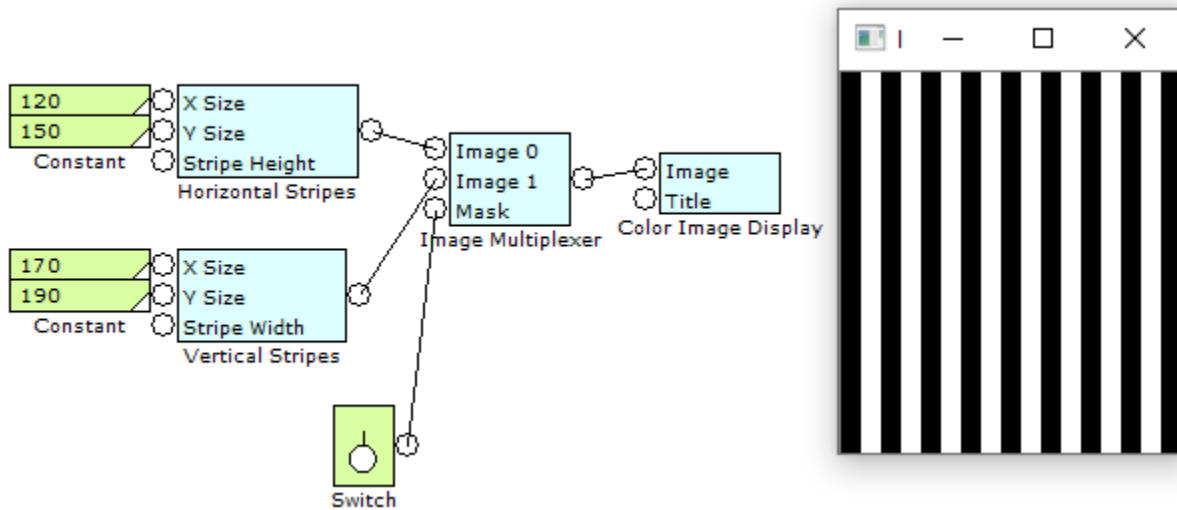
# Image Moments

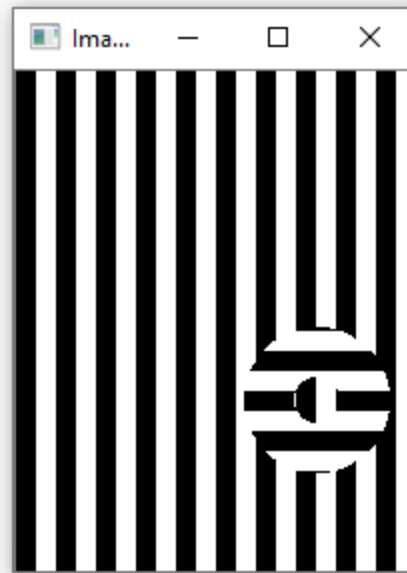
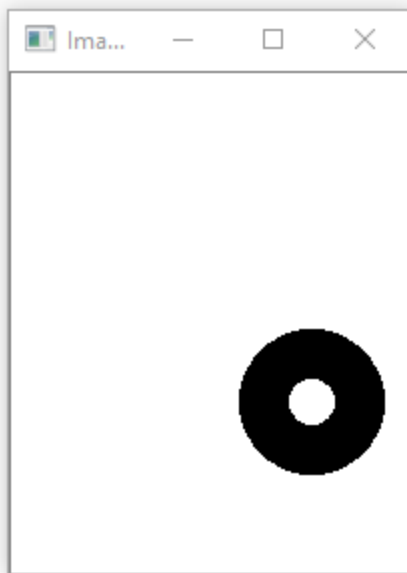
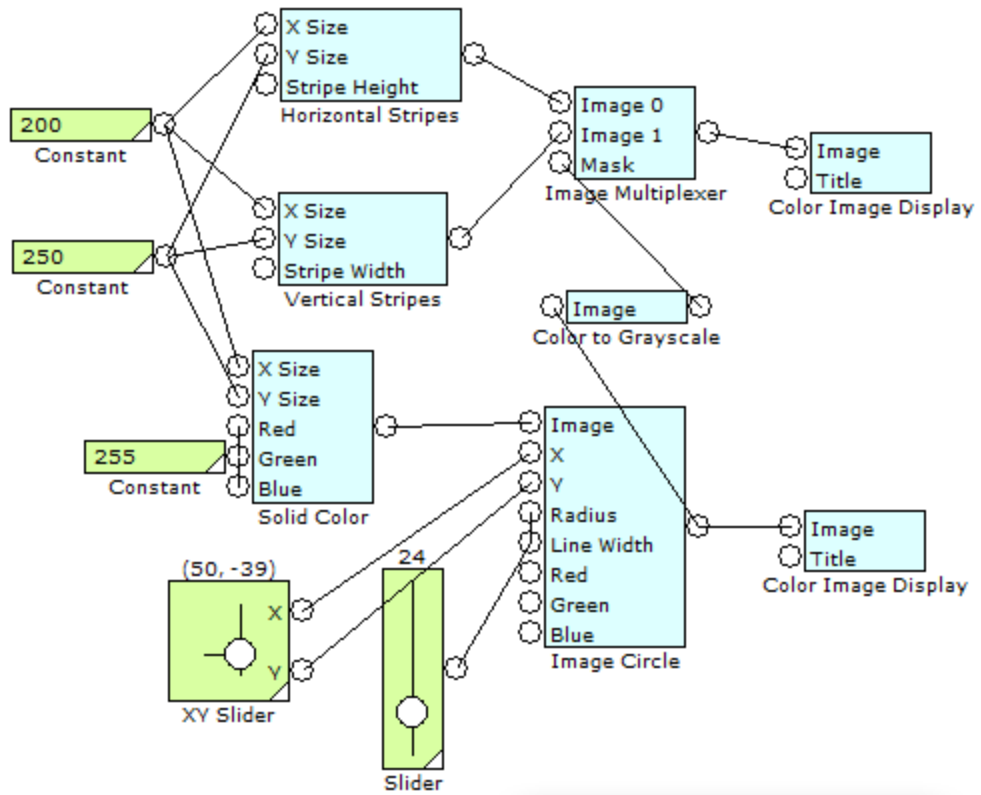
The Image Moments component inputs a grayscale image and returns the moments. Images – column 4



# Image Multiplexer

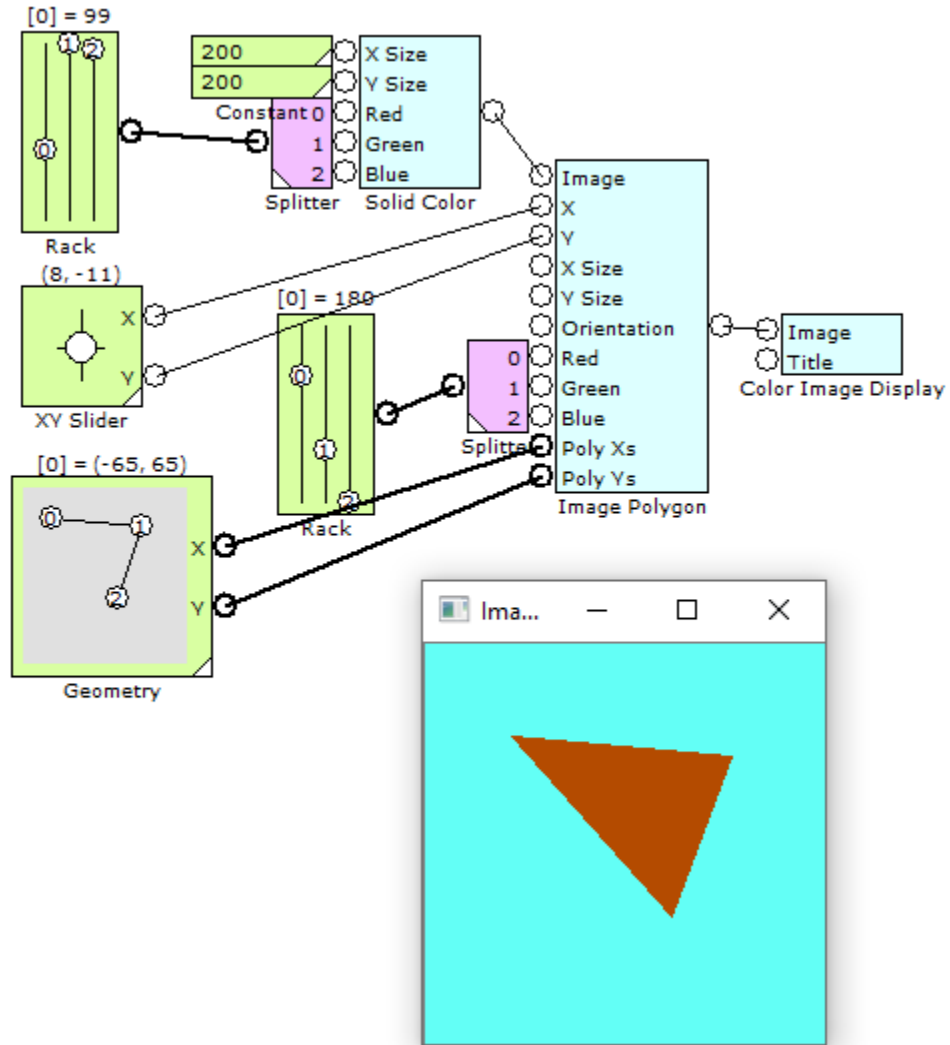
The Image Multiplexer component creates an image by switching between the 2 input images based on the Mask input. Images - column 3





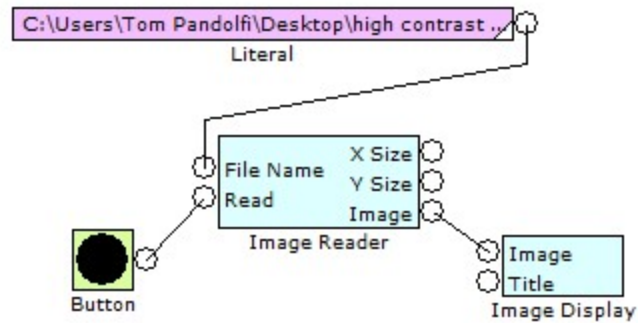
# Image Polygon

The Image Polygon component draws a polygon on an image. Set the Line Width to a negative number to fill the polygon. Images - column 3



# Image Reader

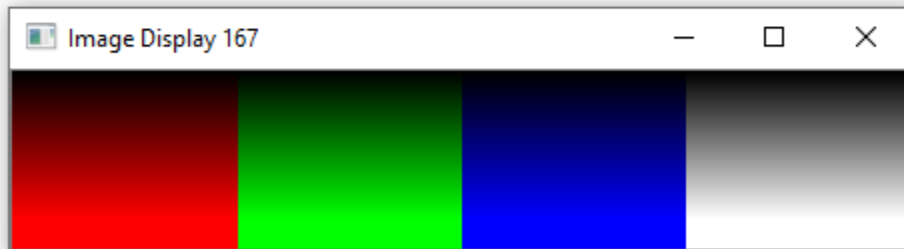
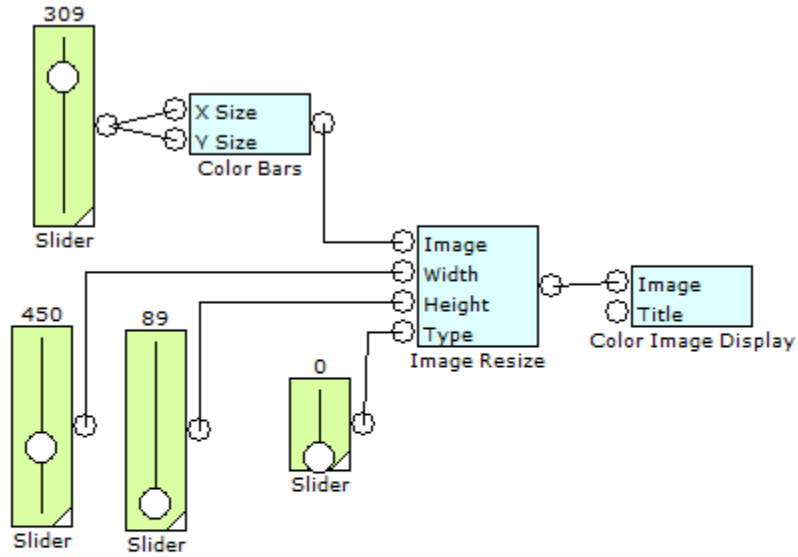
The Image Reader component reads a color or grayscale image at the path given. Images - column 5





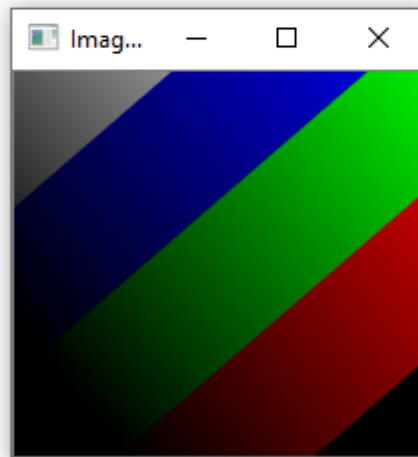
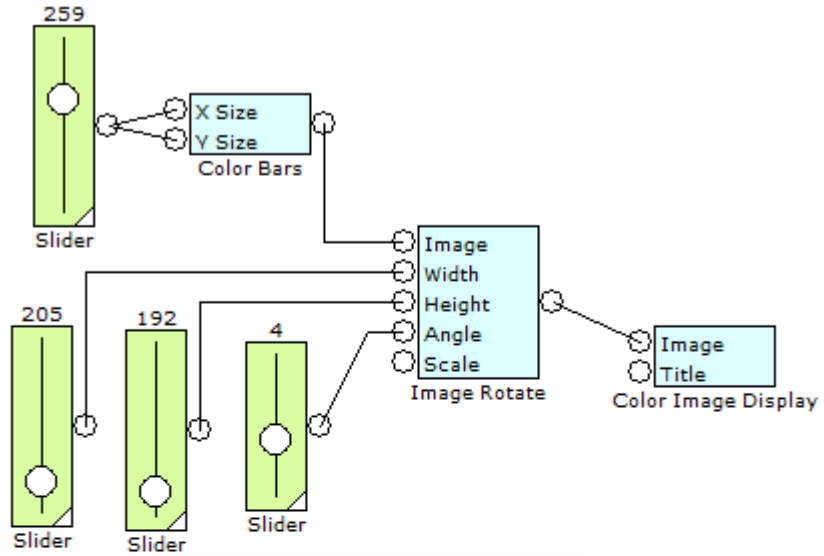
# Image Resize

The Image Resize component resizes the image to the width and height given. Type: 0-Nearest neighbor, 1-Bi-linear, 2-Bi-cubic, 3-Area, 4-Lanczos (8x8), 5-Bit exact bi-linear. Images - column 1



# Image Rotate

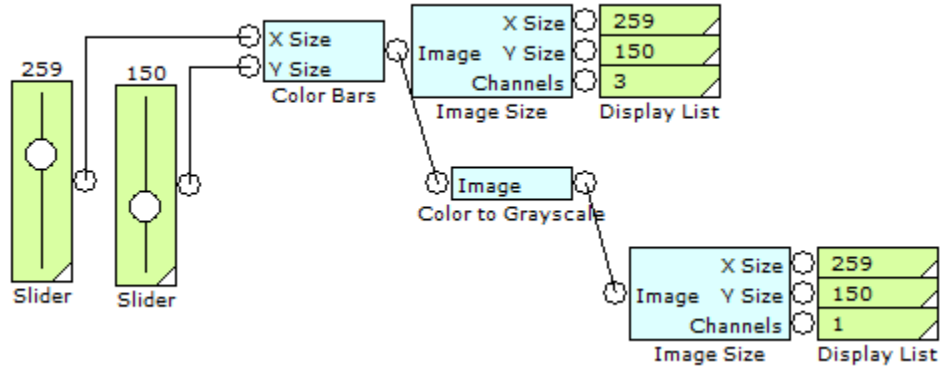
The Image Rotate component rotates the image at the specified angle and scale into an otherwise blank image. Images - column 1



# Image Size

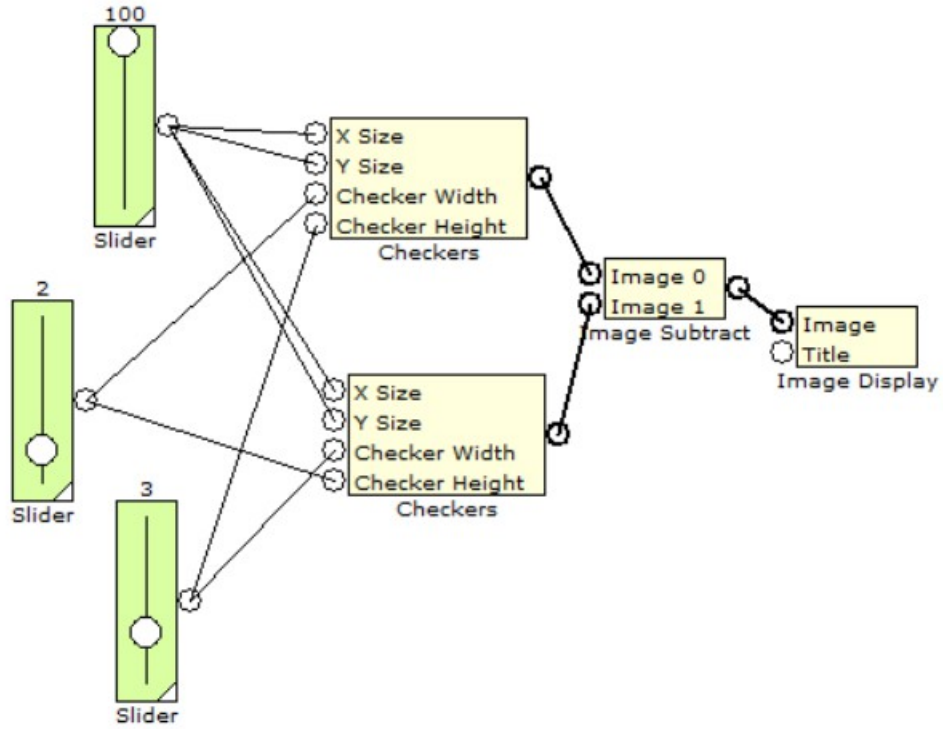
The Image Size component reads the image width, height, and number of channels.

Images - column 1



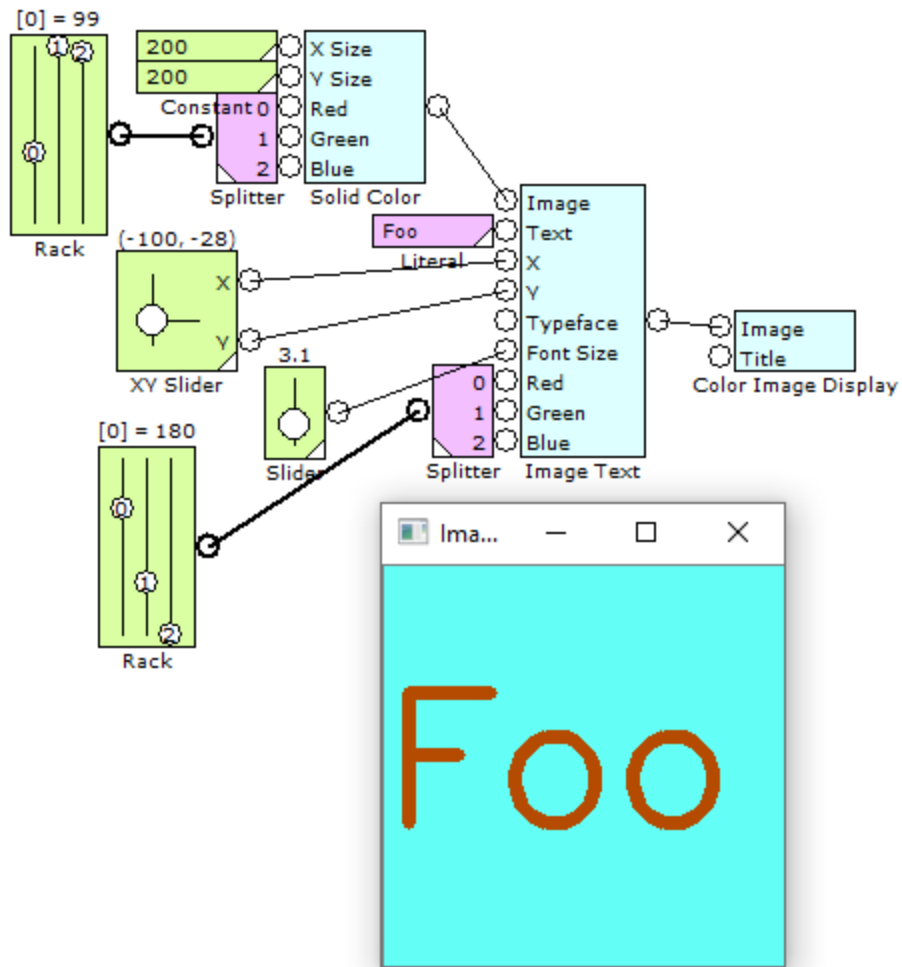
# Image Subtract

The Image Subtract component subtracts the pixels of the second image from the first image. Images – column 2



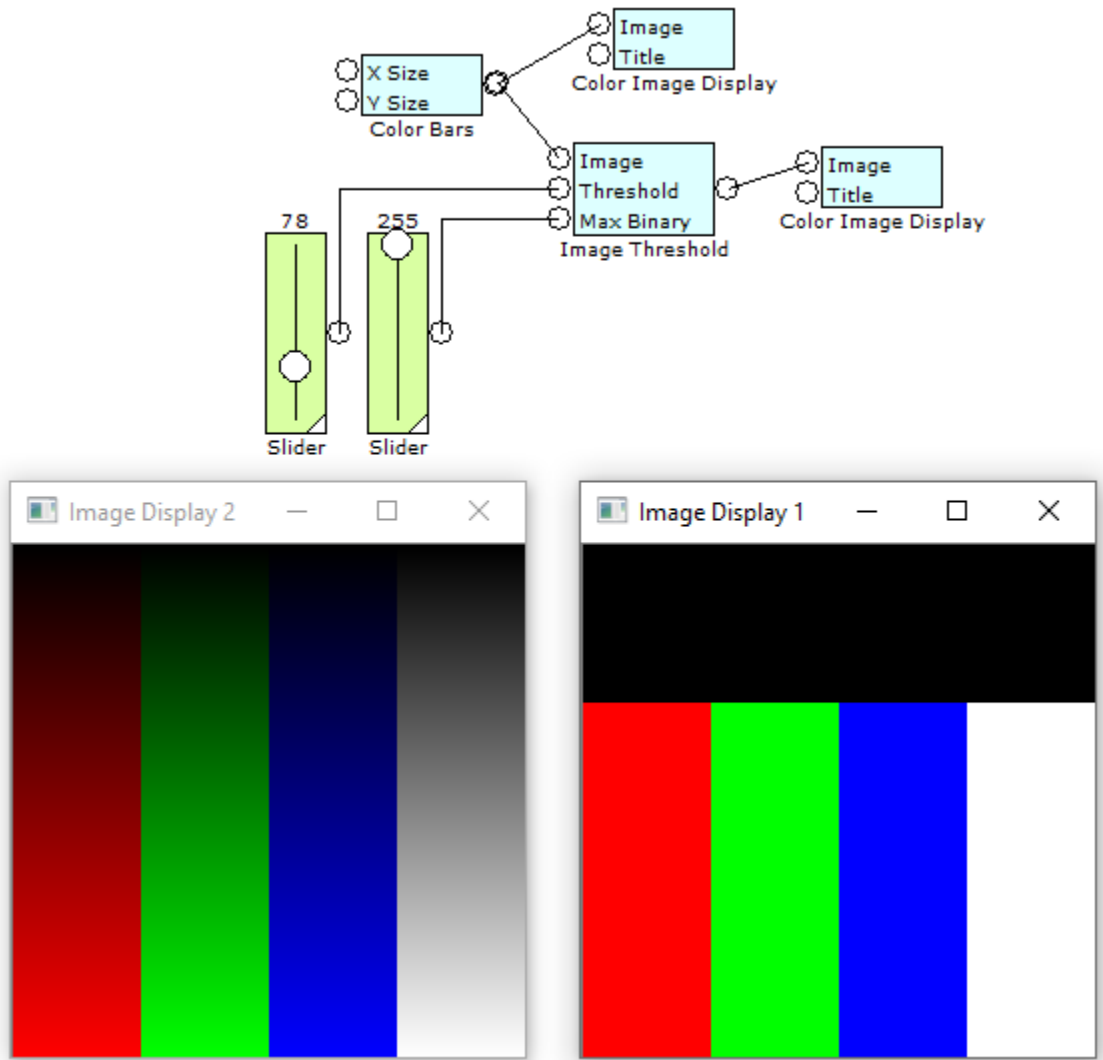
# Image Text

The Image Text component draws the text given on an image. Images - column 3



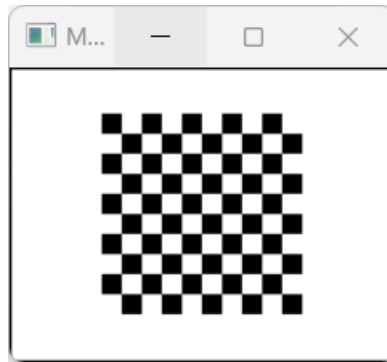
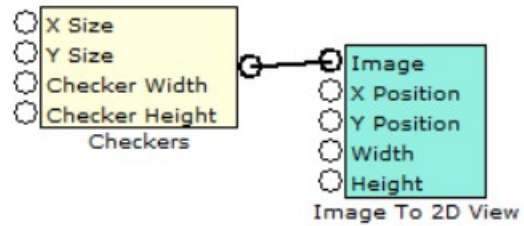
# Image Threshold

The Image Threshold component converts an image to grayscale usually with higher contrast. Images - column 2



## Image to 2D View

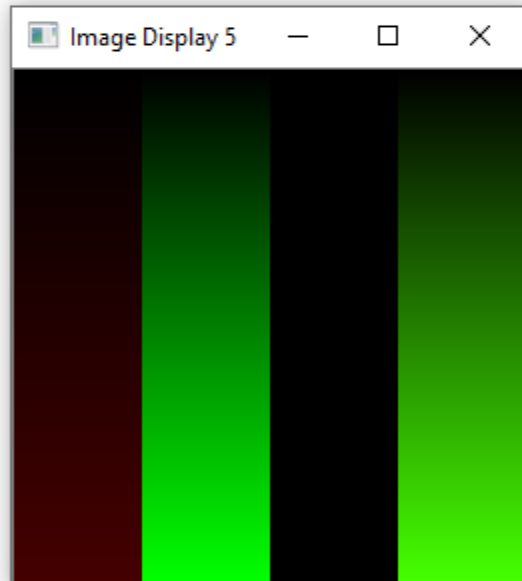
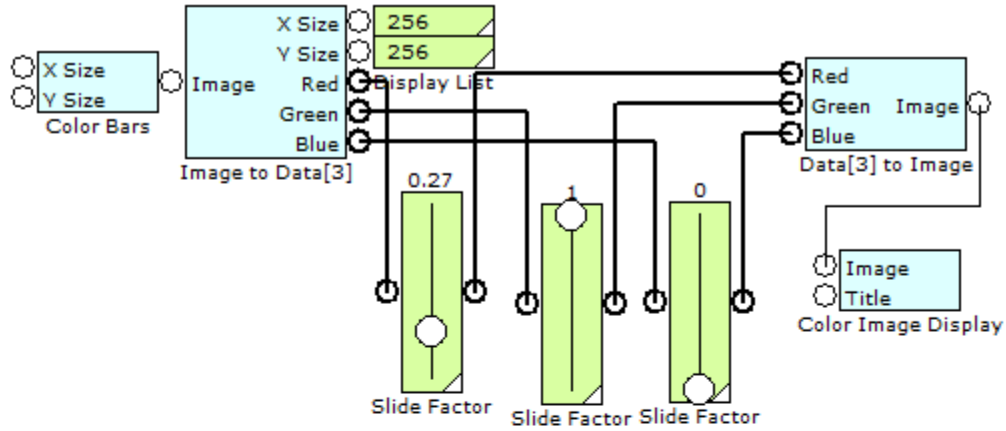
The Image to 2D View component takes an Image and position and size coordinates and draws the image in the 2D view canvas. 2D – column 2



# Image to Data[3]

The Image to Data[3] component outputs the three color matrices from the image.

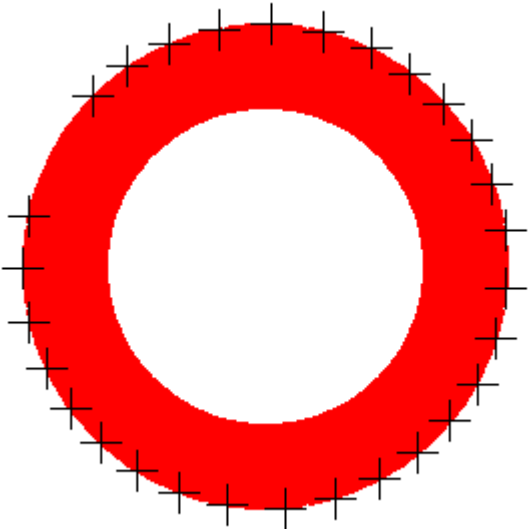
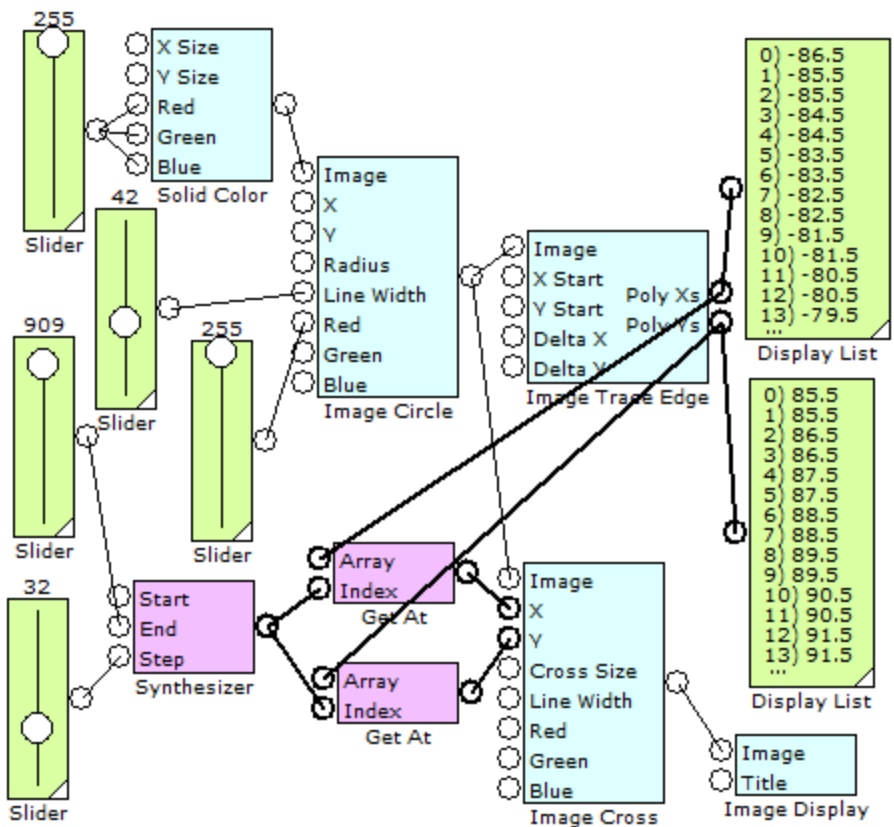
Images - column 5





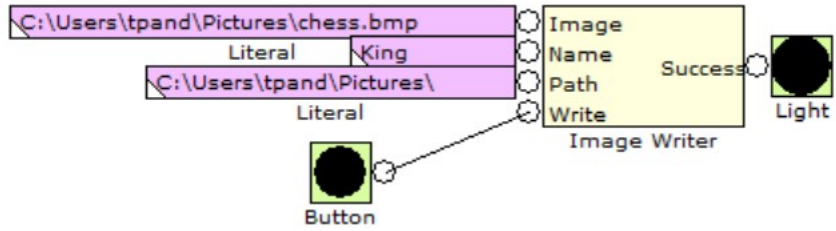
# Image Trace Edge

The Image Trace Edge component follows an edge and outputs X and Y coordinates along that polyline. Images - column 3



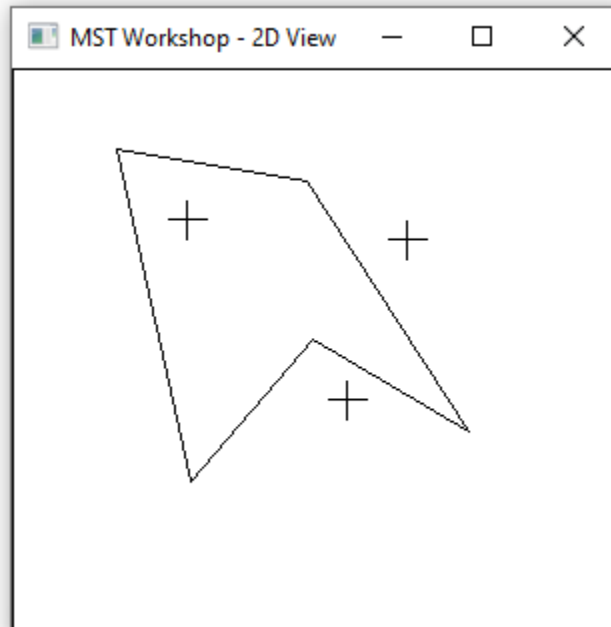
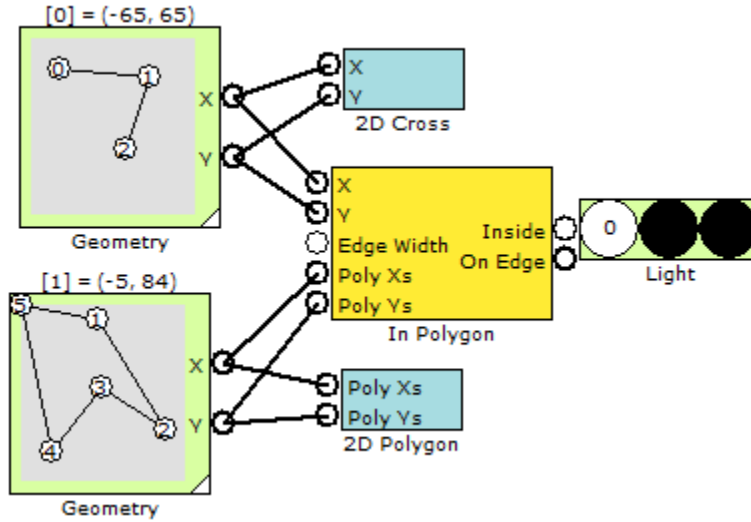
# Image Writer

The Image Writer component writes an image to the path given. Images - column 5



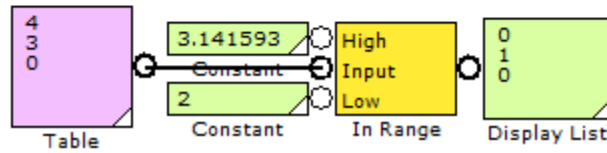
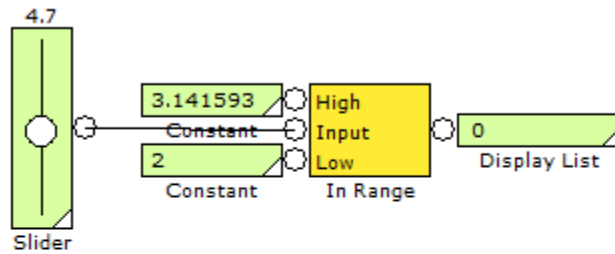
# In Polygon

The In Polygon component gives true if the input is inside the polygon. The edge output is the number of the edge when you are on an edge or -1 when you are not on an edge. Functions - column 2



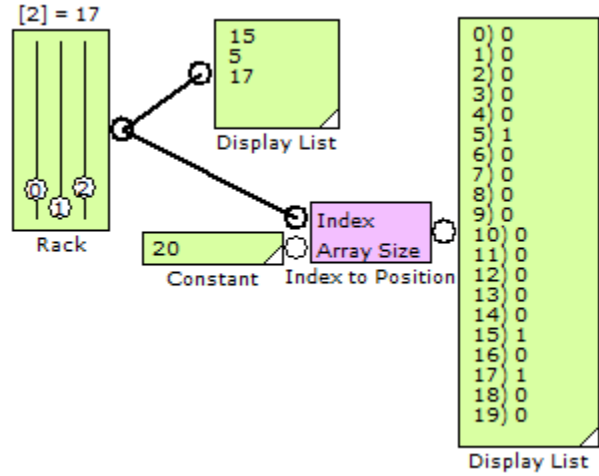
# In Range

The In Range component gives true if the input is between the High and Low values. Functions - column 2



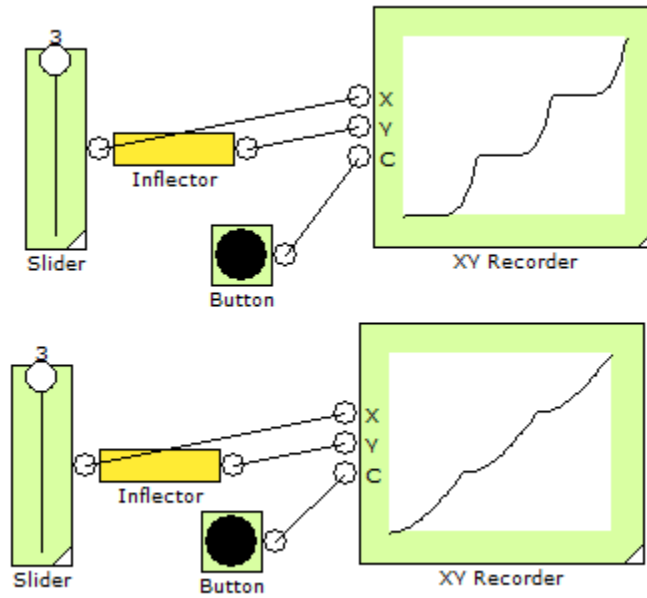
# Index to Position

The Index to Position component creates an array of Array Size and puts 1's in the positions specified by the Index input. Arrays - column 4



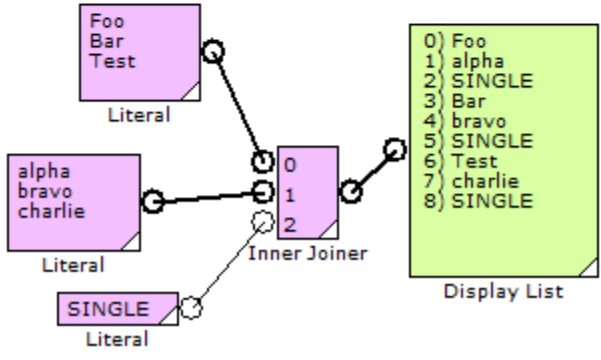
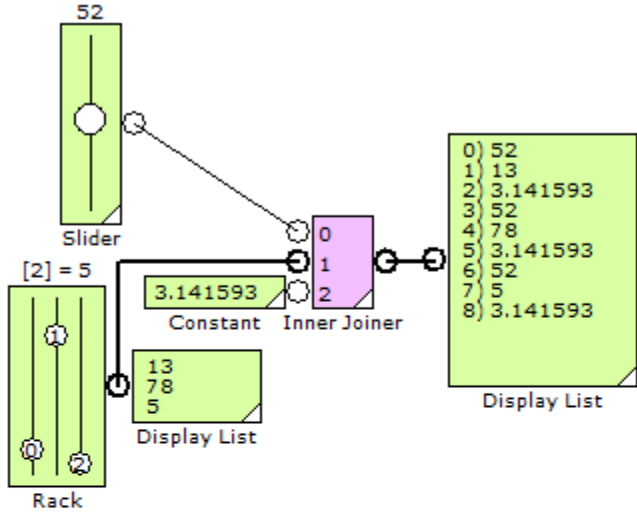
# Inflector

The Inflector component modifies the input to provide inflection at each integer value. Input can be scalar or an array. Functions – column 1



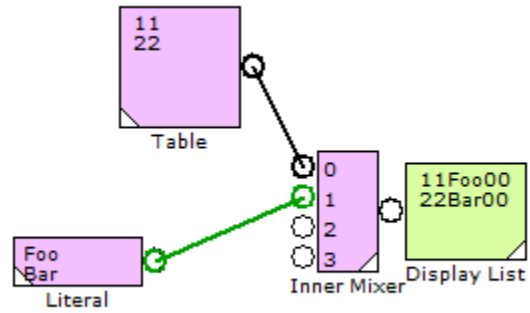
# Inner Joiner

The Inner Joiner component builds an array by sequencing through all the inputs. Each pass uses the next element in that input's array. The inputs can be a scalar, array, string or an array of strings. Arrays - column 2



## Inner Mixer

The Inner Mixer component builds a string array by sequencing through all the inputs. Each pass uses the next element in that input's array. The inputs can be a scalar, array, string or an array of strings.





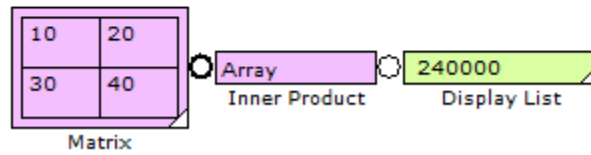
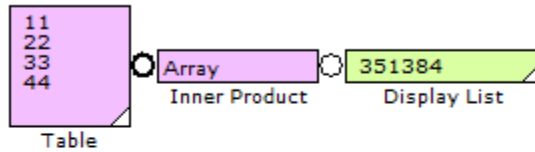
# Inner Product

The Inner Product component scans the input array multiplying the values together. You can set the number of dimensions. Arrays - column 3

Inner Product

One Dimension  
 Two Dimensions  
 Three Dimensions

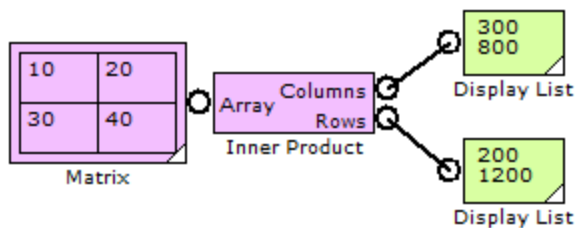
OK Cancel



Inner Product

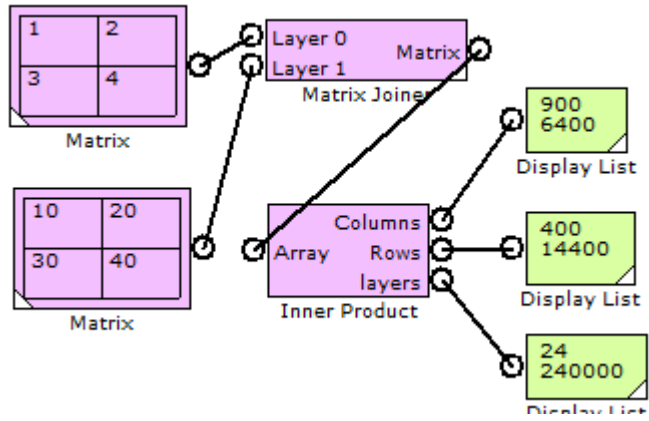
One Dimension  
 Two Dimensions  
 Three Dimensions

OK Cancel



Inner Product

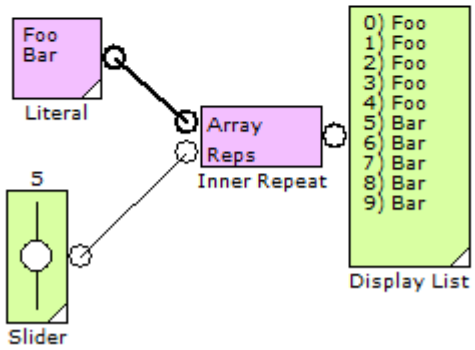
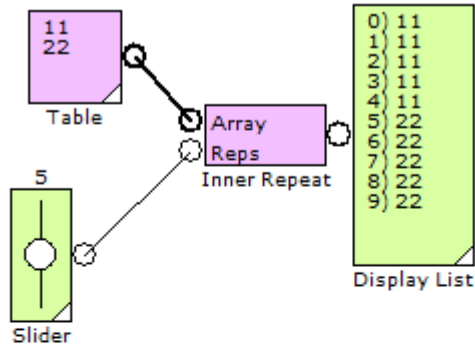
One Dimension  
 Two Dimensions  
 Three Dimensions



# Inner Repeat

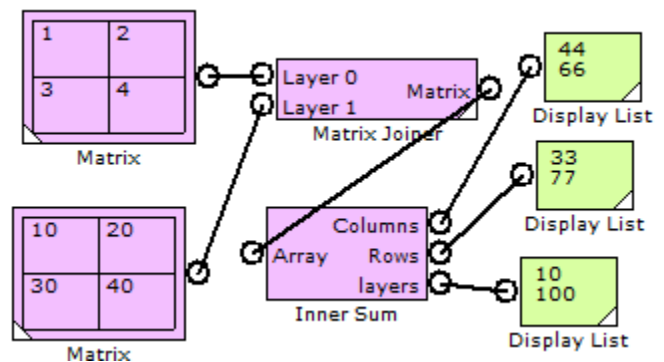
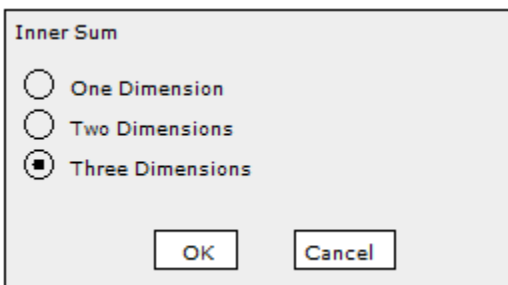
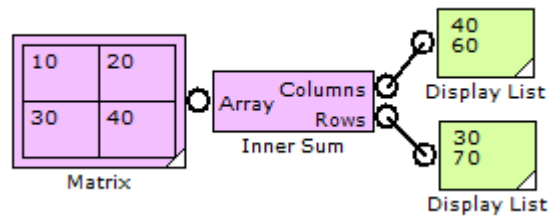
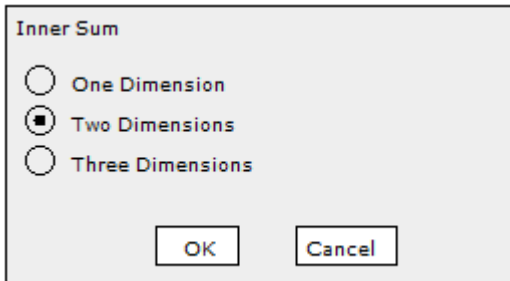
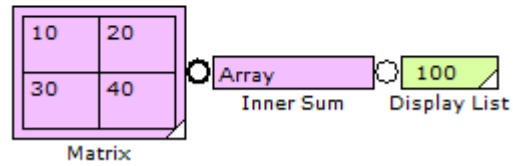
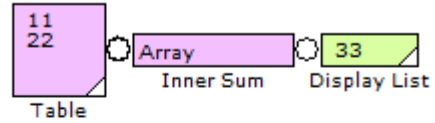
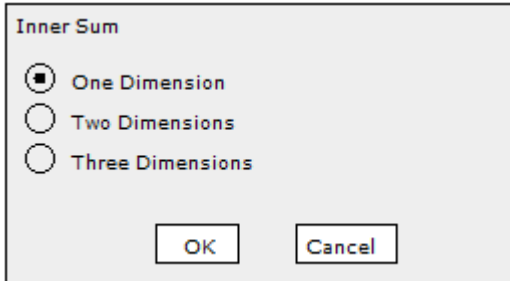
The Inner Repeat component builds an array by repeating the Array input. The Array input can be a scalar or an array. For example: an input of 1, 2, 3 with a Reps input of 2 will output 1, 1, 2, 2, 3, 3.

Arrays - column 2



# Inner Sum

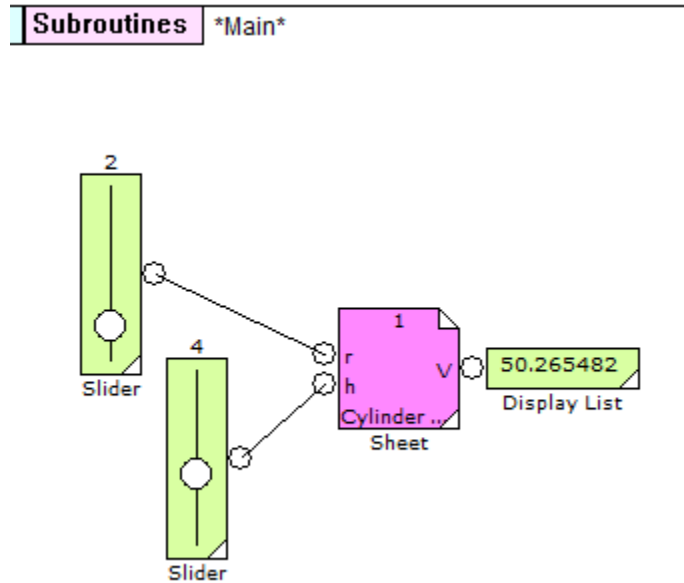
The Inner Sum component scans the input array summing the values together. You can set the number of dimensions. Arrays - column 3



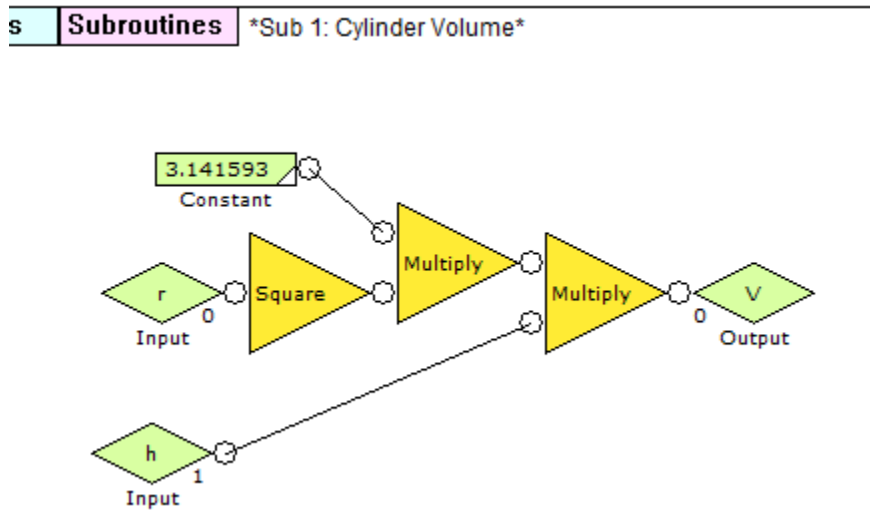
# Input

The Input component is used in a subroutine to carry values into the subroutine. Each Input shows as an input pin on the subroutine icon. You can set a Default Value when nothing is connected to this input pin. Controls - column 5

In the main workspace:

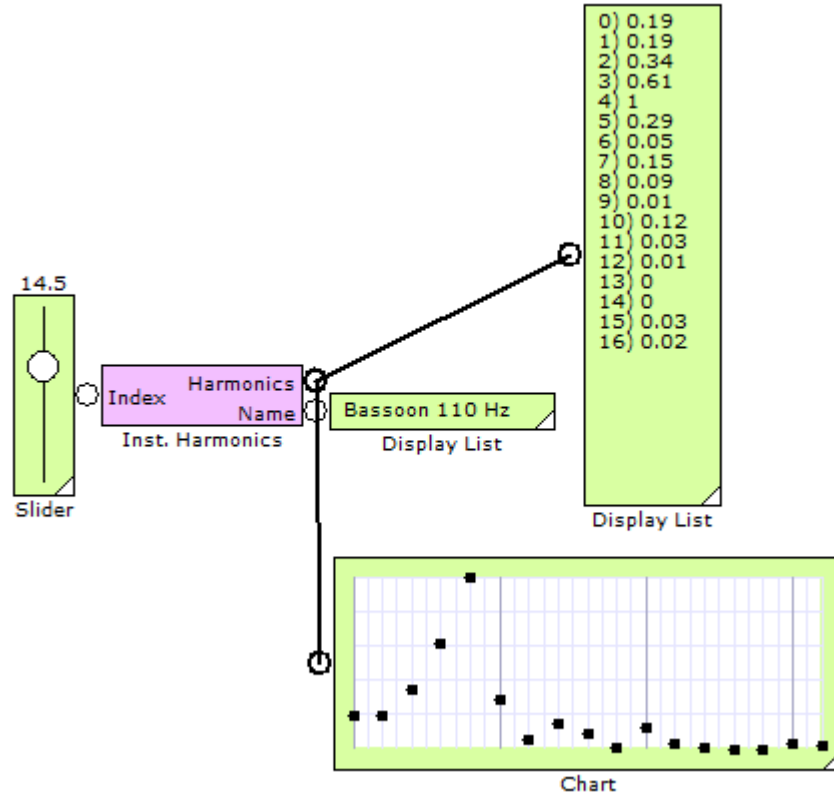


In the subroutine:



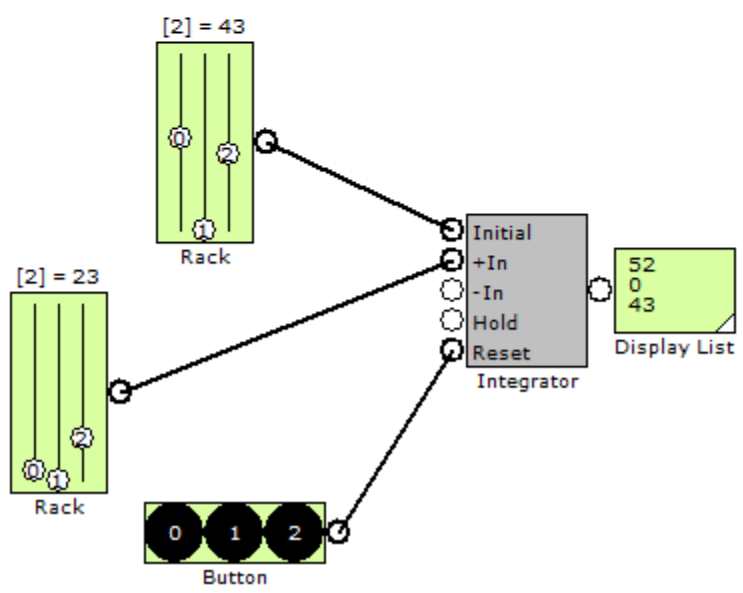
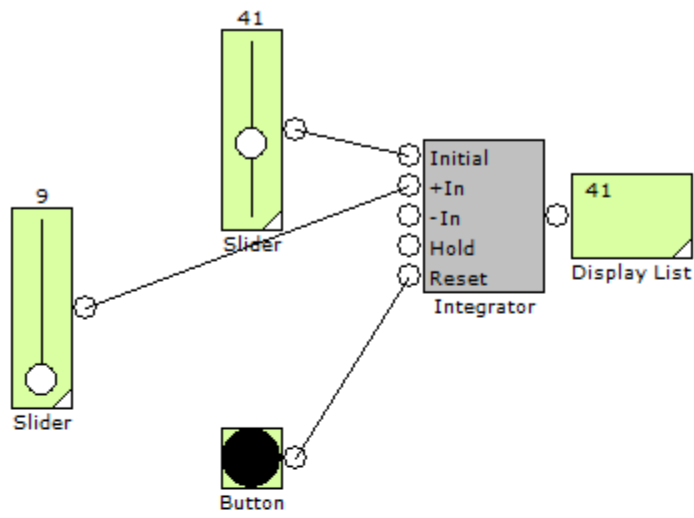
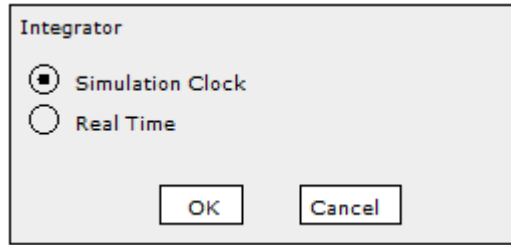
# Inst. Harmonics

The Inst. Harmonics component lets you choose from a list of instruments and provides the harmonic amplitudes of the instrument's sound. Arrays - column 7



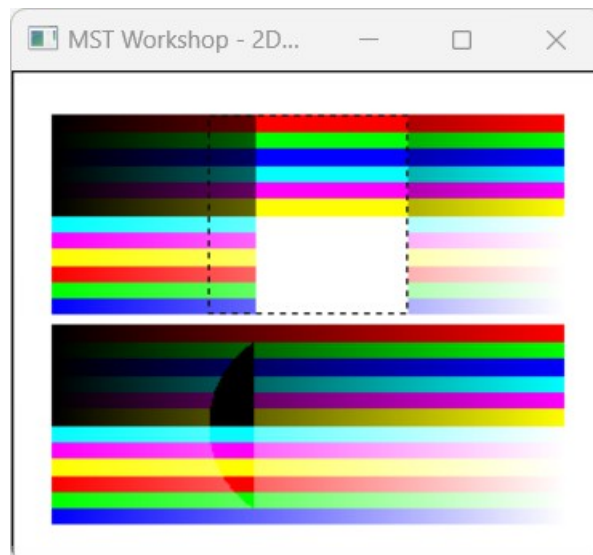
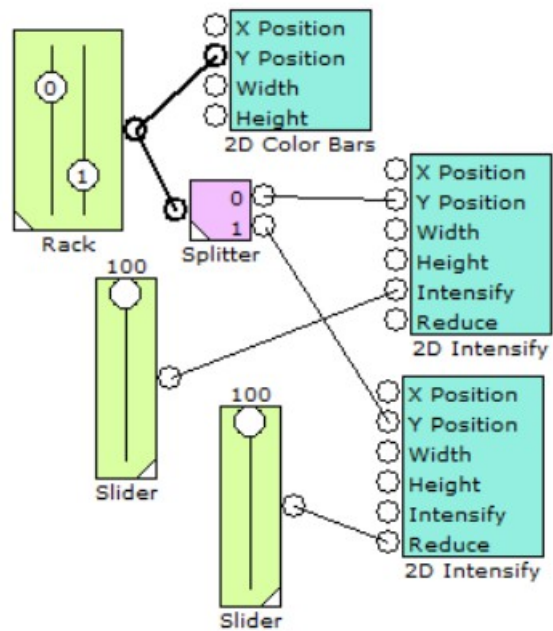
# Integrator

The Integrator component integrates the inputs over time. Functions - column 6



## 2D Intensify

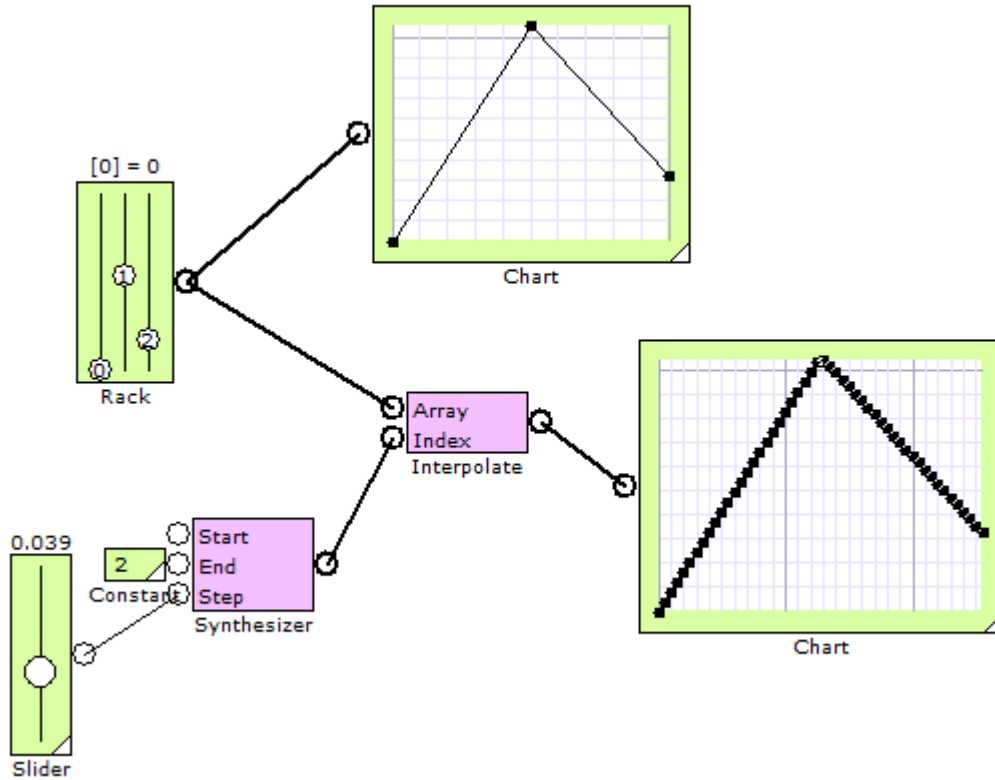
The 2D Intensify reduces or intensifies the 2D drawing it covers. 2D – column 1





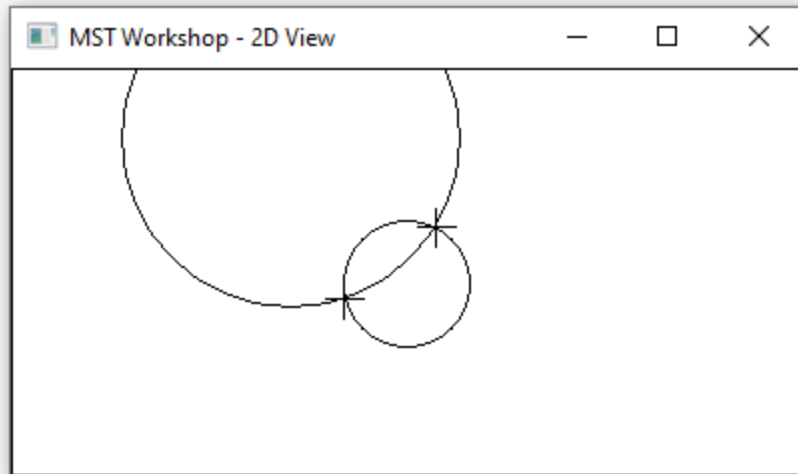
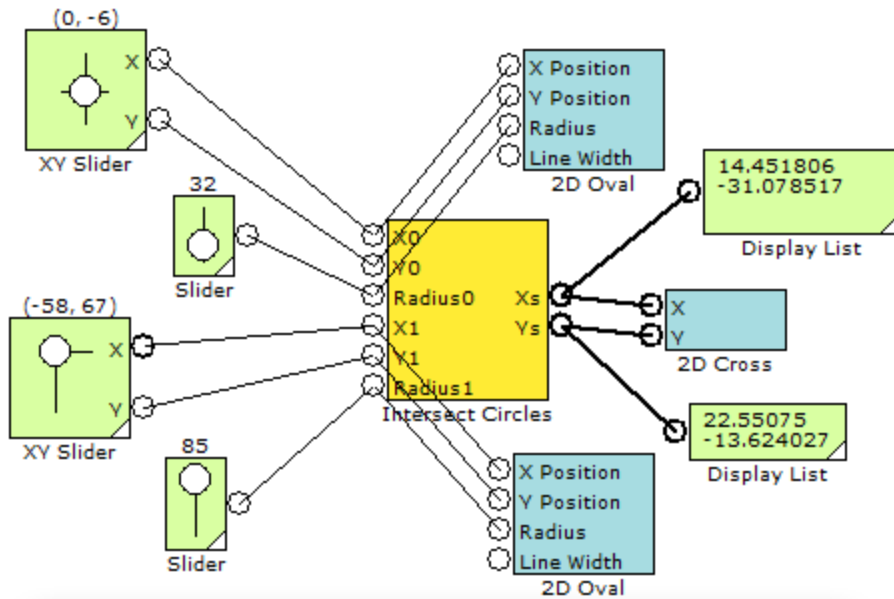
# Interpolate

The Interpolate component gets the Array values around the Index input and interpolates. It outputs the value(s) at the positions given by the Index(s).      Arrays - column 2



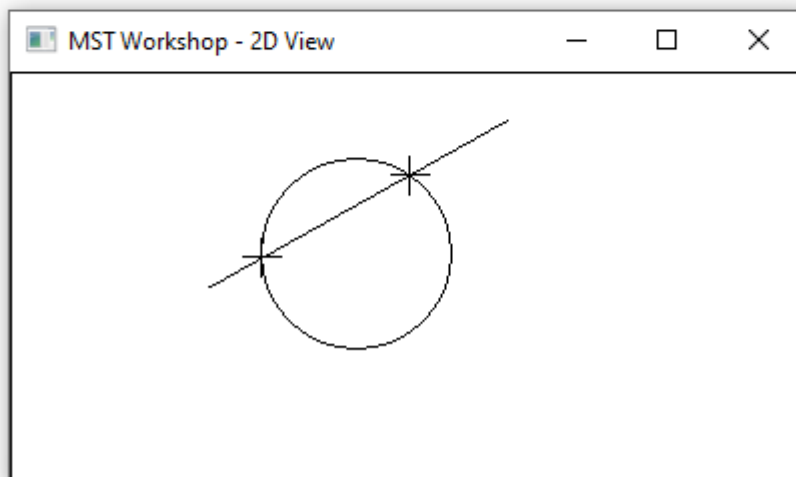
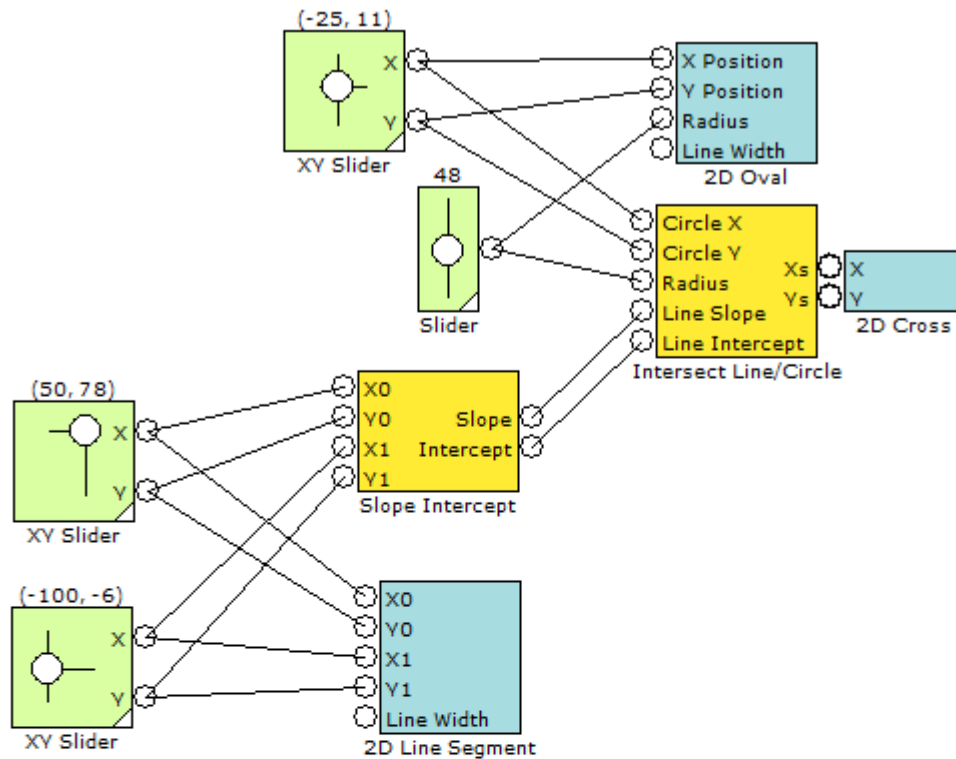
# Intersect Circles

The Intersect Circles component calculates the intersection between two circles. Functions - column 8



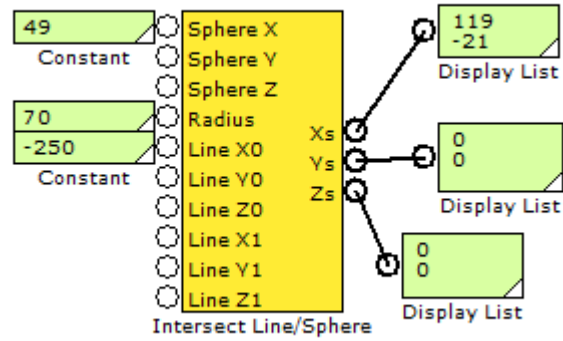
# Intersect Line/Circle

The Intersect Line/Circle component calculates the intersection of a line and a circle. Functions - column 8



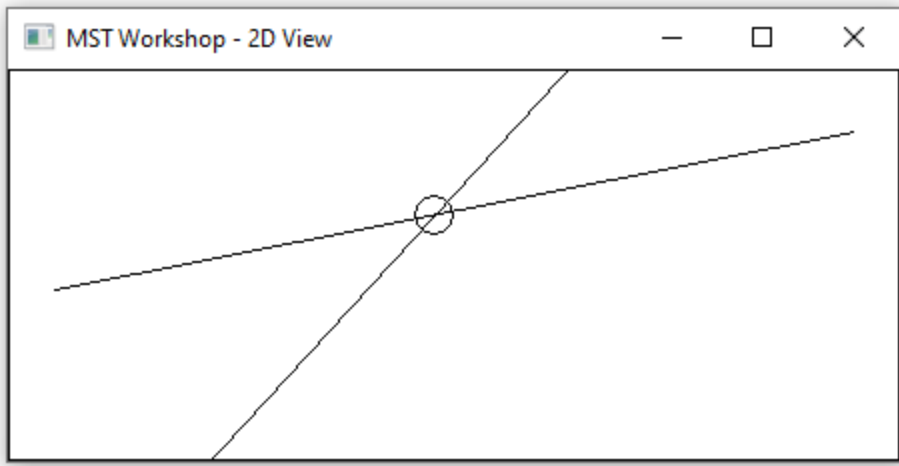
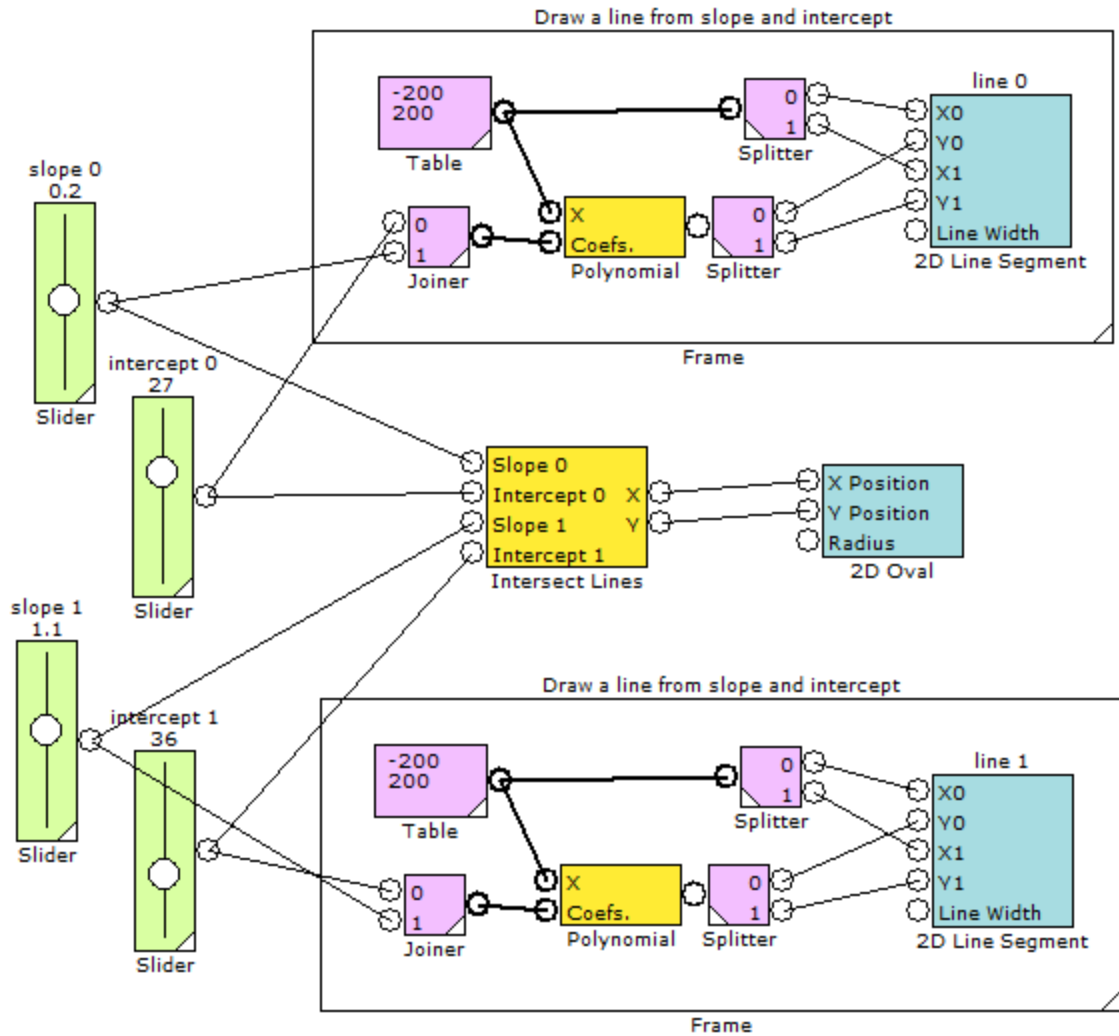
## Intersect Line/Sphere

The Intersect Line/Sphere component calculates the intersection of a line and a sphere.



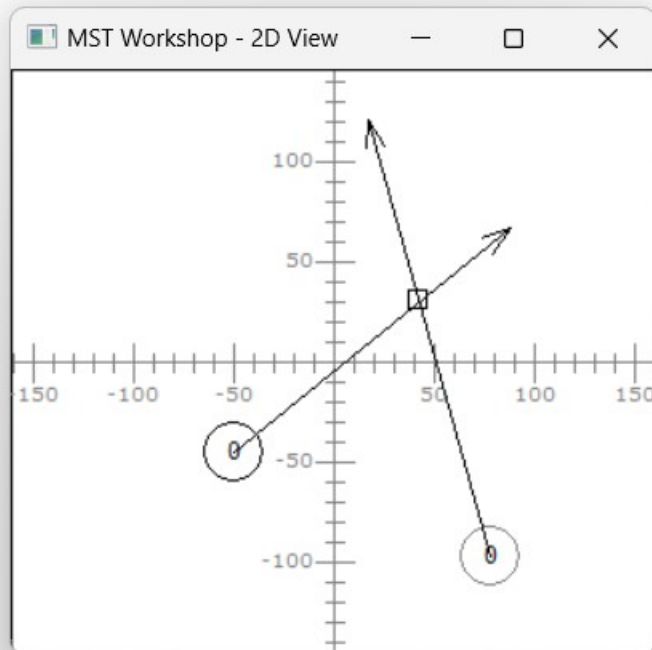
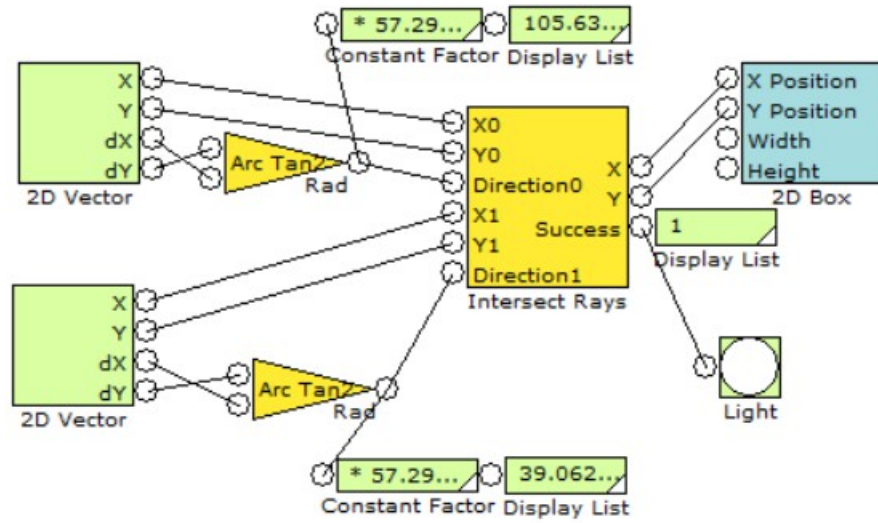
# Intersect Lines

The Intersect Lines component calculates the intersection between two lines. Functions - column 8



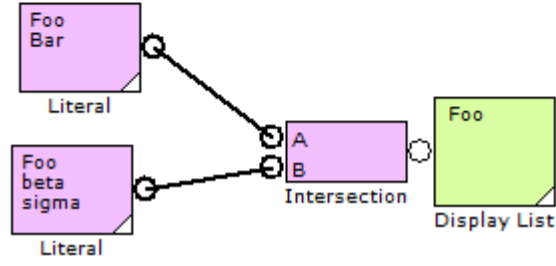
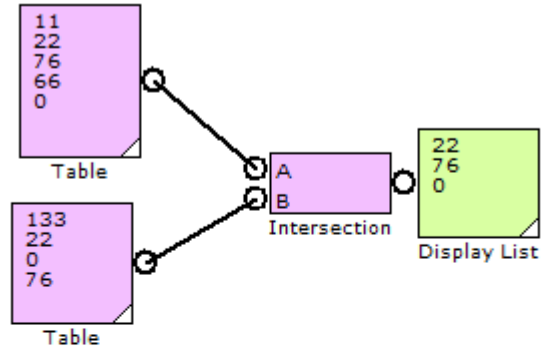
# Intersect Rays

The Intersect Rays component calculates the intersection between two rays. Functions – column 8



# Intersection

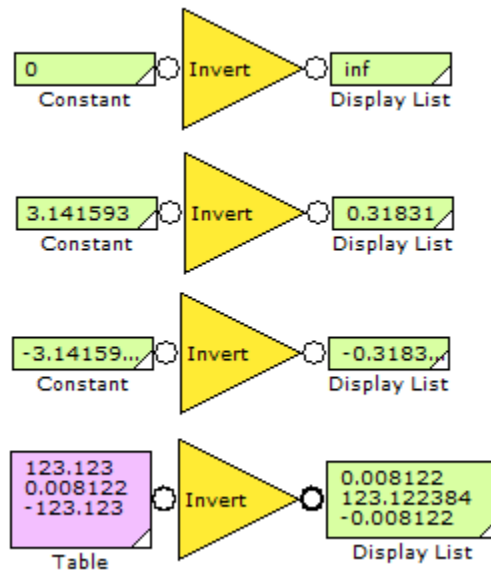
The Intersection component scans the input arrays and outputs the elements common to both inputs.  
Arrays - column 5



# Invert

The Invert component calculates one over input. Input can be a scalar or an array.

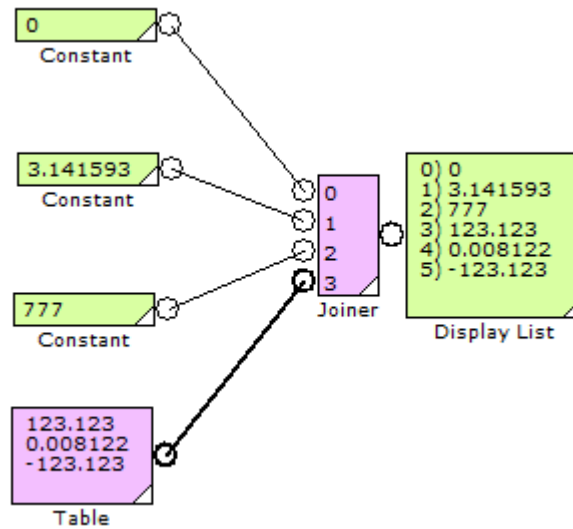
Functions -  
column 1





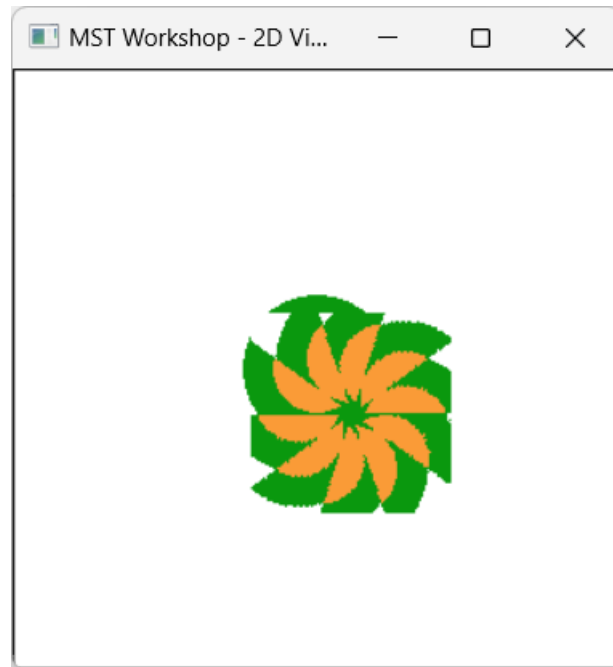
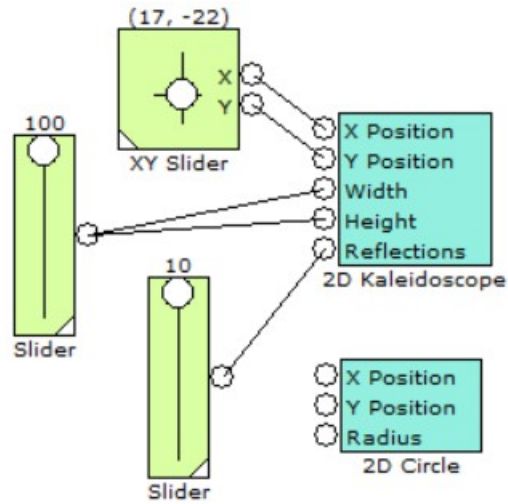
## Joiner

The Joiner component joins all the values for all its inputs into a single output. You can set the number of inputs. Arrays - column 1



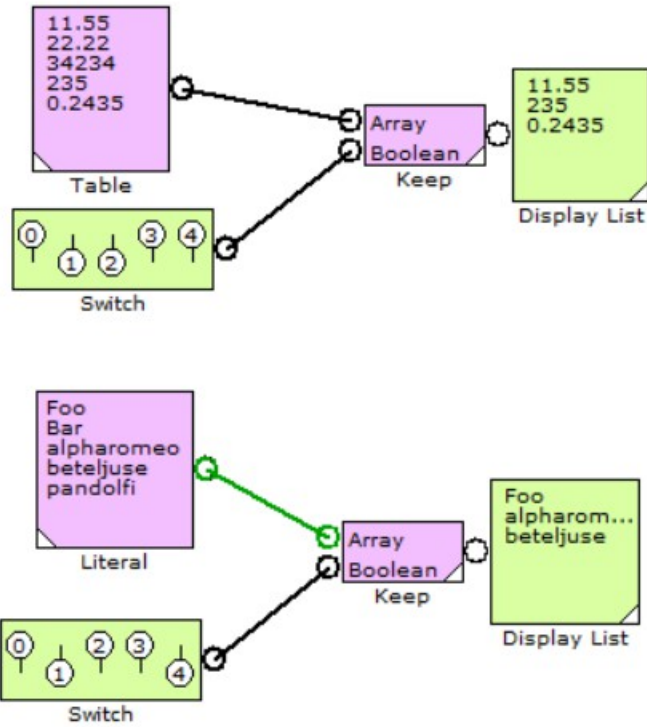
## 2D Kaleidoscope

The 2D Kaleidoscope creates the repeating kaleidoscope effect on the image in the 2D drawing it covers. Note: This component only affects the items in the 2D View window that are under it. 2D – column 7



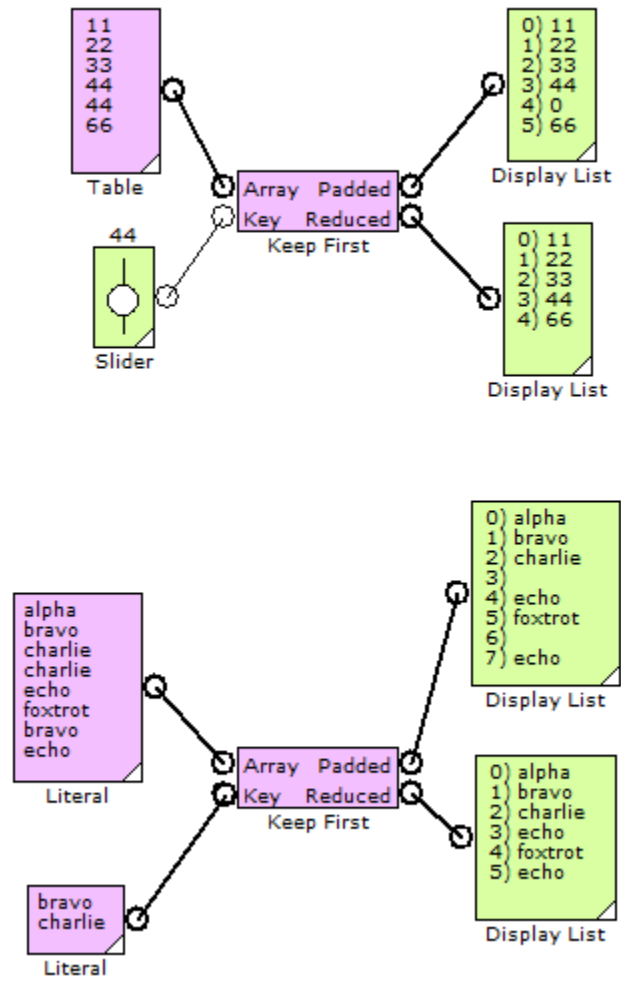
# Keep

The Keep component outputs only the values of the input array that correspond to the true values in the Boolean input. Arrays - column 2



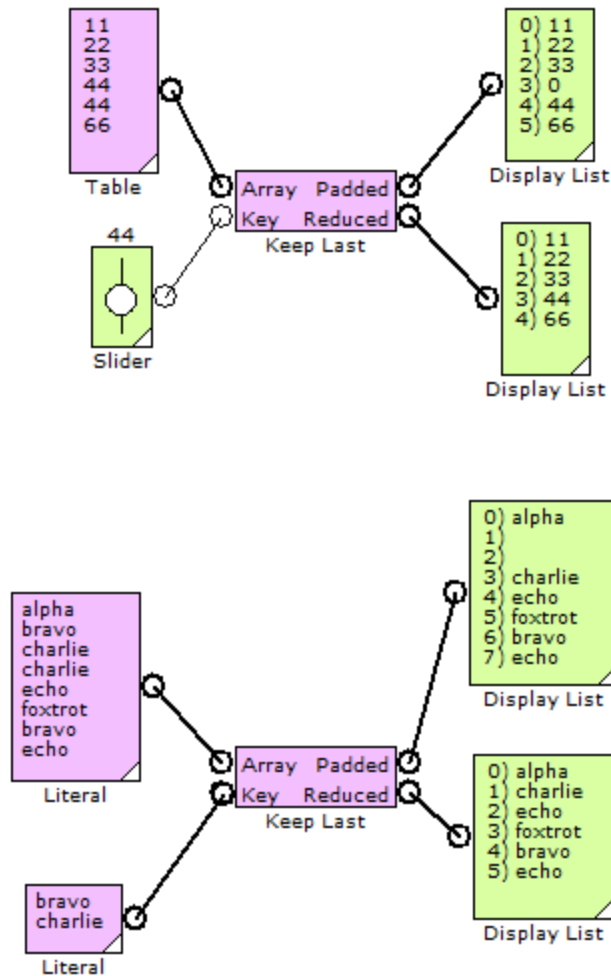
# Keep First

The Keep First component outputs the values of the input array but removes extra copies of the key, keeping only the first occurrence. Arrays - column 2



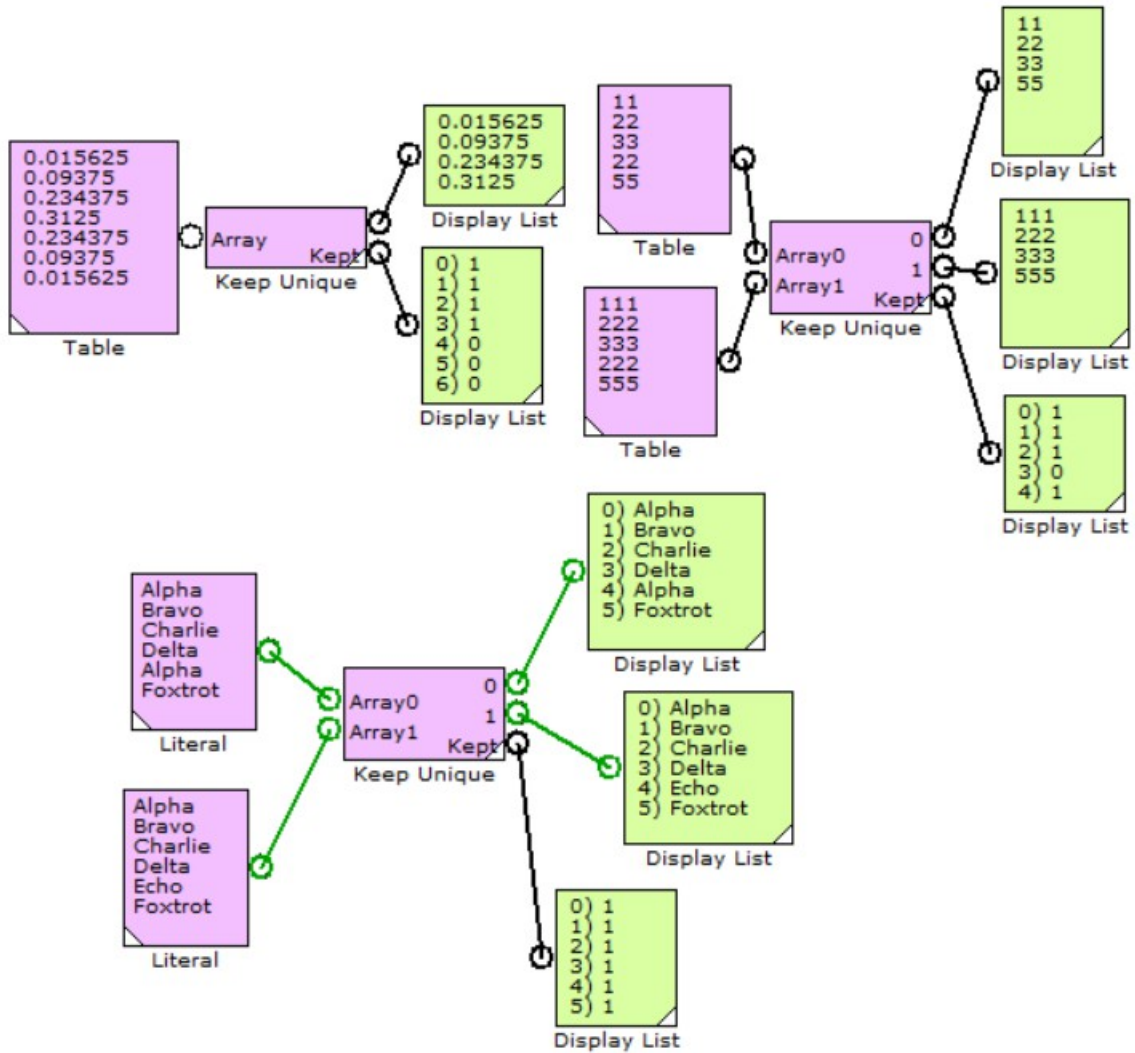
# Keep Last

The Keep Last component outputs the values of the input array but removes extra copies of the key, keeping only the last occurrence. Arrays - column 2



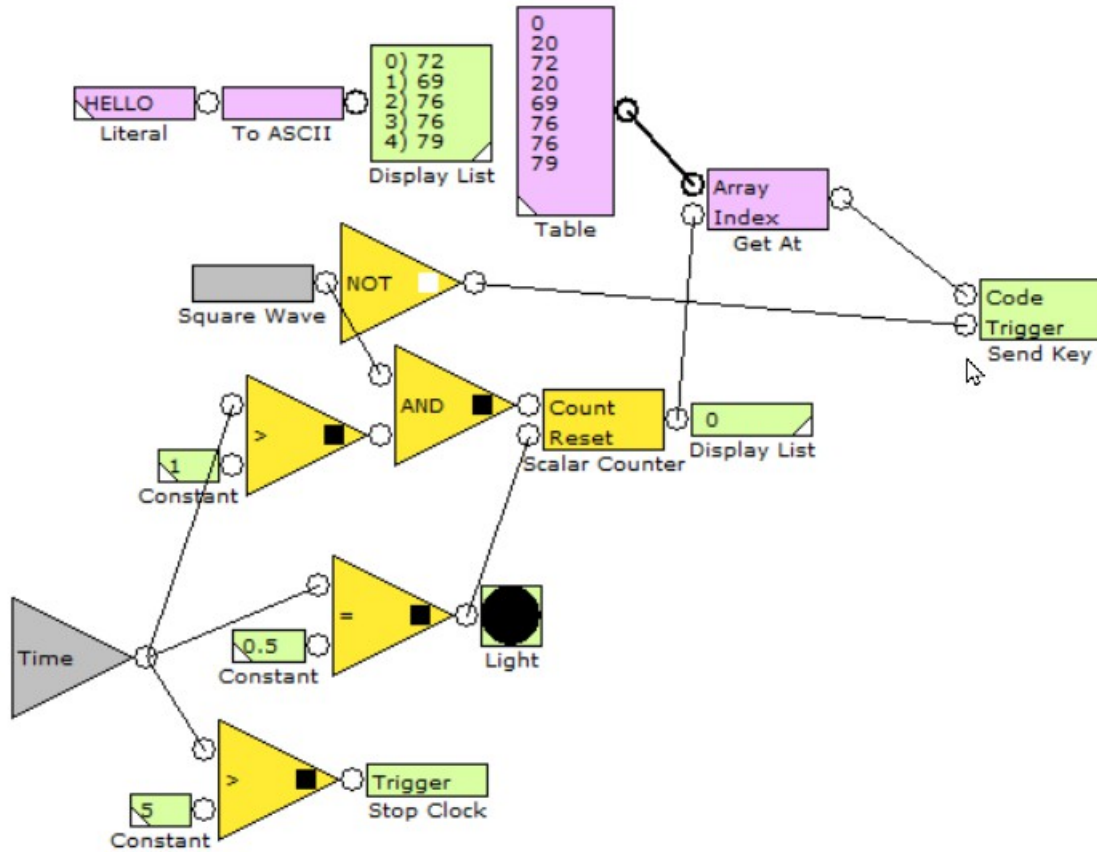
# Keep Unique

The Keep Unique component outputs only the values of the input array that are unique. In other words it removes duplicates. When multiple inputs are enabled, all inputs are considered as a group. Arrays – column 2



# Key Send

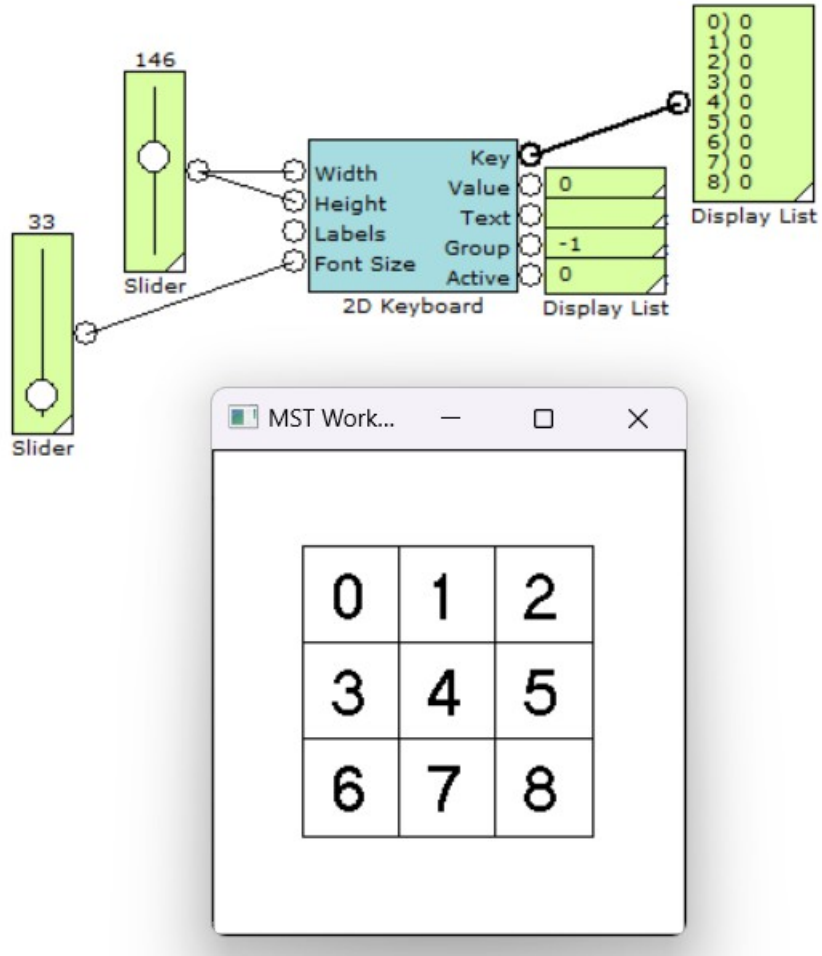
The Key Send component simulates a key click as if someone is typing on the computer's keyboard. The Code input is the Virtual Key Code not ASCII. For example you can click the Caps Lock key using a Code of 20. The Ctrl key is 17. Controls – column 4



This solution will type the word “Hello” one second after the clock is started. Notice the code of 20 before and after the “H” key is sent to make it a capital.

# 2D Keyboard

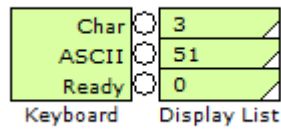
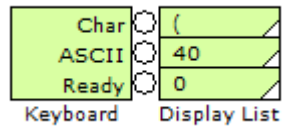
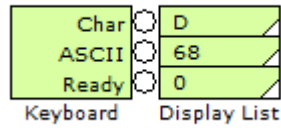
The 2D Keyboard draws a keyboard in the 2D View window. 2D – column1





# Keyboard

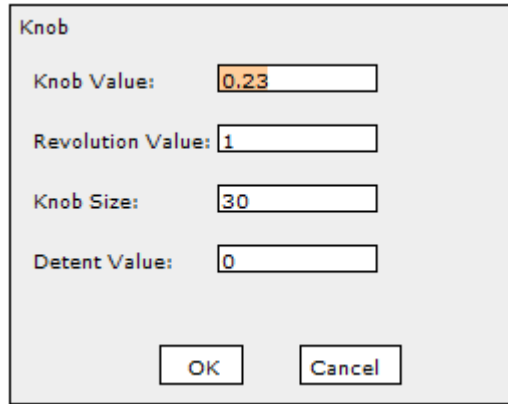
The Keyboard outputs the ASCII value and the character typed at the computer's keyboard.  
Controls - column 4



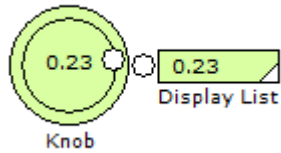
# Knob

The Knob is used to create a constant value by setting the dial. You can control how much change you get from each revolution. Note: This component can be operated using the wheel on the mouse.

Controls - column 1

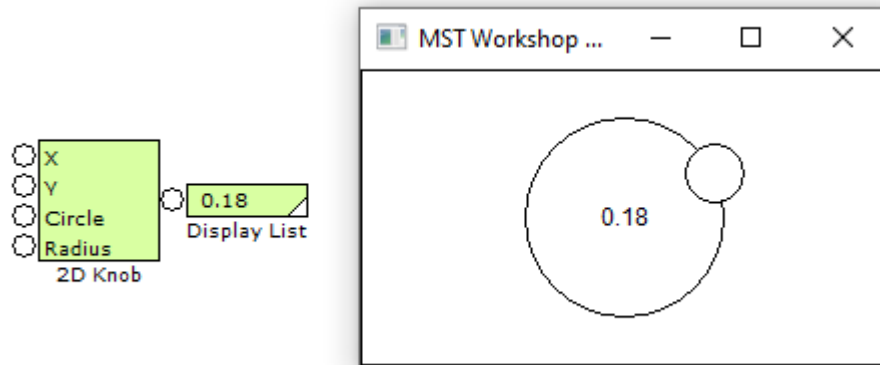


A dialog box titled "Knob" with four input fields and two buttons. The fields are: "Knob Value:" with the value "0.23", "Revolution Value:" with the value "1", "Knob Size:" with the value "30", and "Detent Value:" with the value "0". The "OK" and "Cancel" buttons are at the bottom.



## 2D Knob

The 2D Knob is used to create a constant value by setting the dial. You can control how much change you get from each revolution. Controls - column 1



# Knob Factor

The Knob Factor multiplies the inputs by the constant in the dial. Controls - column 2

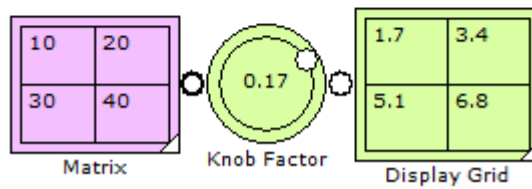
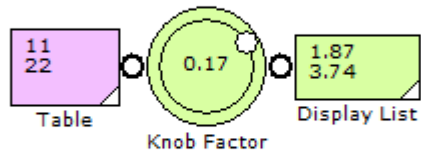
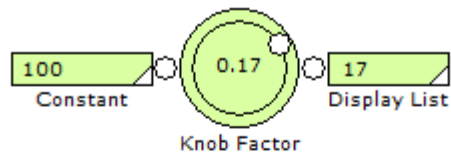
Knob Factor

Knob Value:

Revolution Value:

Knob Size:

Detent Value:



# Knob Offset

The Knob Offset adds its value to each of its inputs.

Controls - column 3

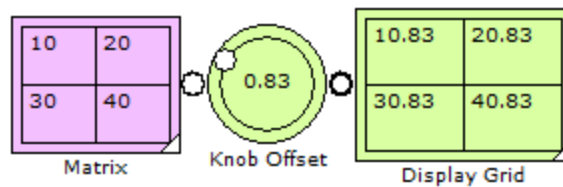
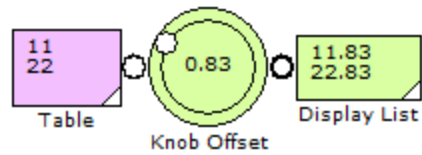
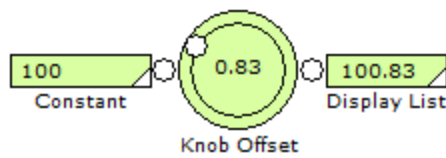
Knob Offset

Knob Value:

Revolution Value:

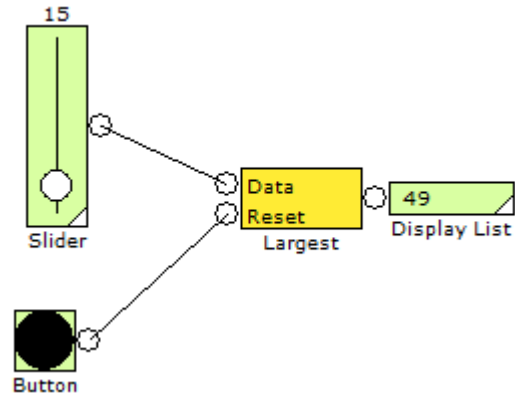
Knob Size:

Detent Value:



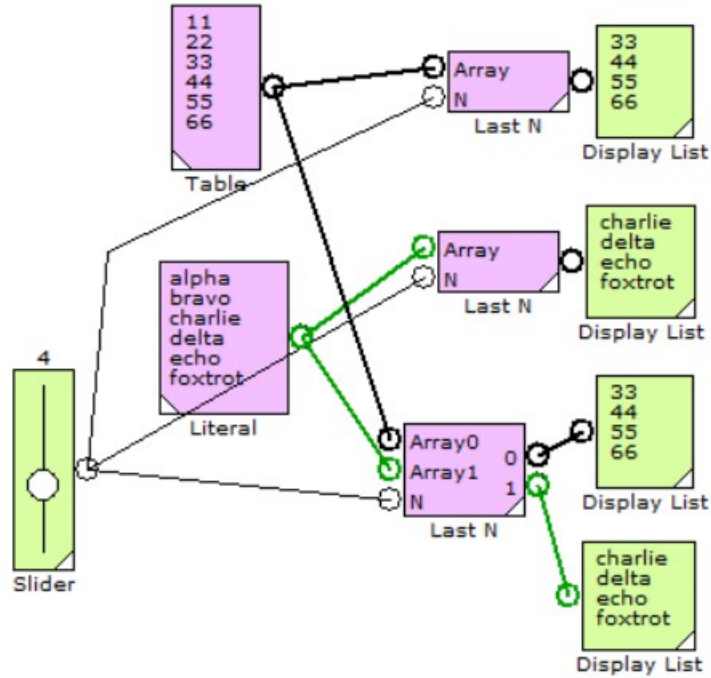
# Largest

The Largest component watches the Data input and outputs the largest value since it was reset.  
Functions - column 2



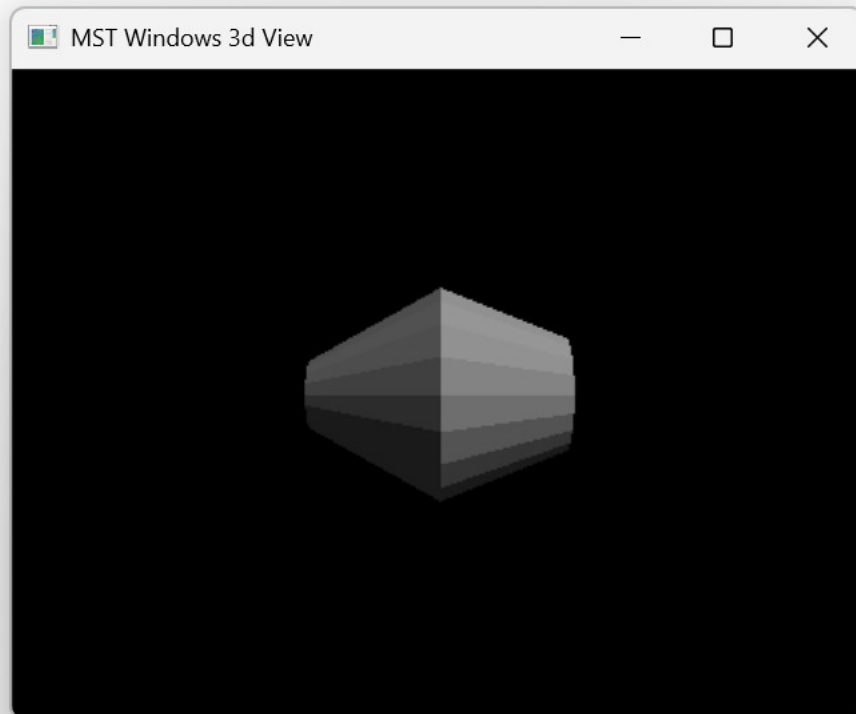
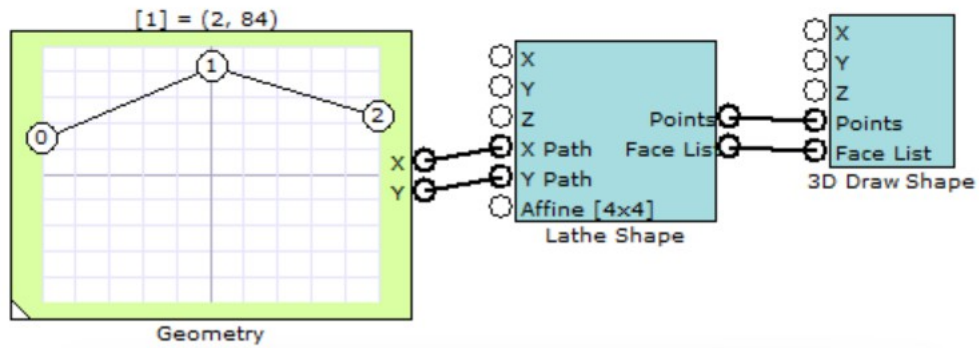
# Last N

The Last N component outputs just the last few values in an array. Input the number N to indicate the number of values you want at the output. Arrays - column 1



# Lathe Shape

The Lathe Shape component creates triangular mesh composed of points and faces from a lathe. You input an X, Y array to define the outline of the Lathe. Use its outputs to reshape or draw it in the 3D View by connecting it to the 3D Draw Shape component. 3D - column 6





## 3D Lathe

The 3D Lathe rotates a 2D line into the 3D object. The X, Y, and Z inputs set the position and size of the object. The Affine [4x4] array input modifies the size, position and orientation. 3D - column 5

**3D Lathe**

Number of Steps:

X Offset

Y Offset

Z Offset

Close the Top

Close the Bottom

Align on X axis

Align on Y axis

Align on Z axis

**Pin options:**

X Position    Z Position

X Rotate    Z Rotate

X Size    Z Size

Red    Blue

Alpha

X Polyline   Y Polyline

[4x4]

**Rotation:**

X Clockwise

X Counter Clockwise

Y Clockwise

Y Counter Clockwise

Z Clockwise

Z Counter Clockwise

X zero is North

X zero is East

X zero is South

X zero is West

Y zero is North

Y zero is East

Y zero is South

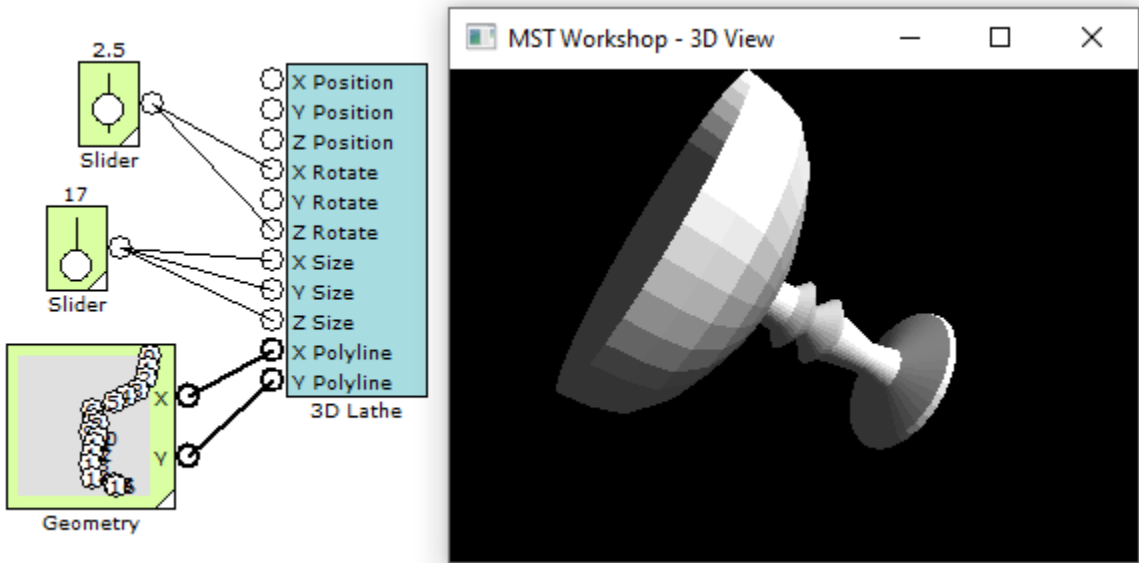
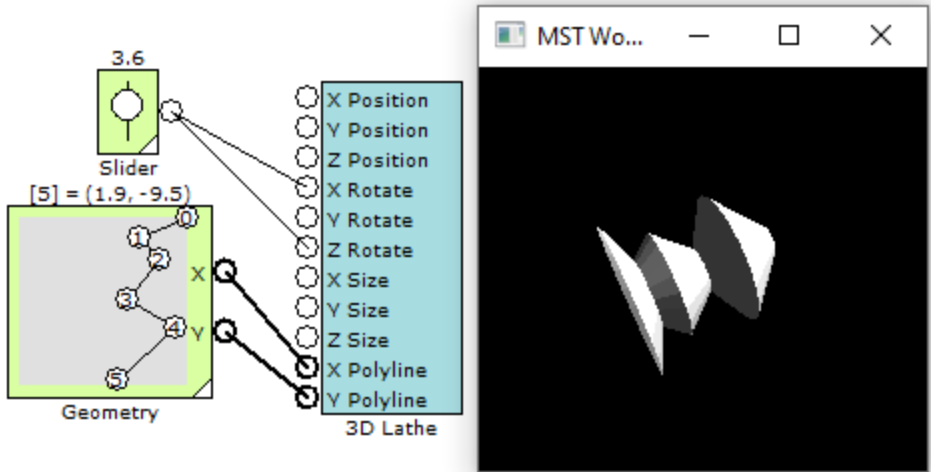
Y zero is West

Z zero is North

Z zero is East

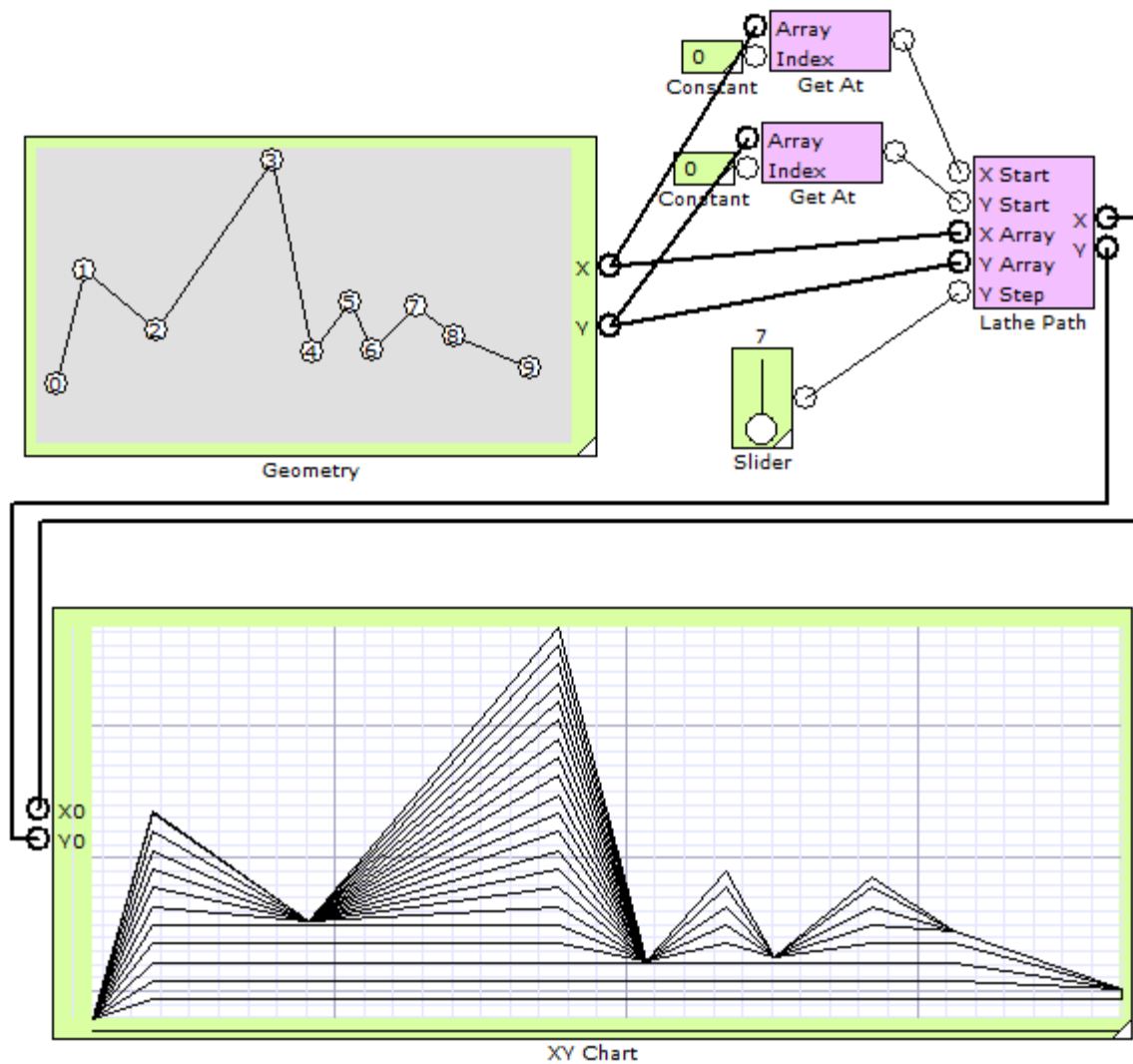
Z zero is South

Z zero is West



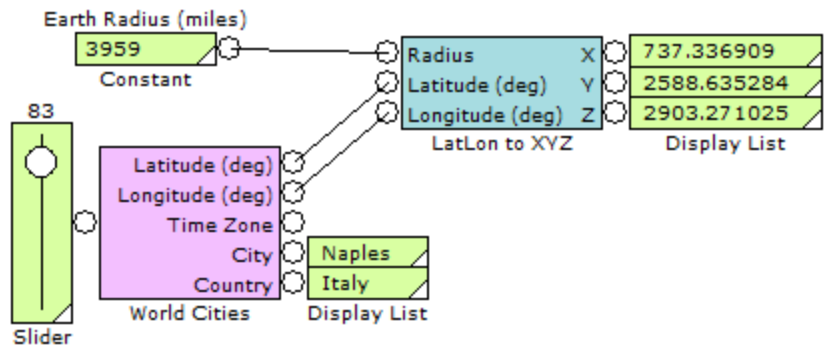
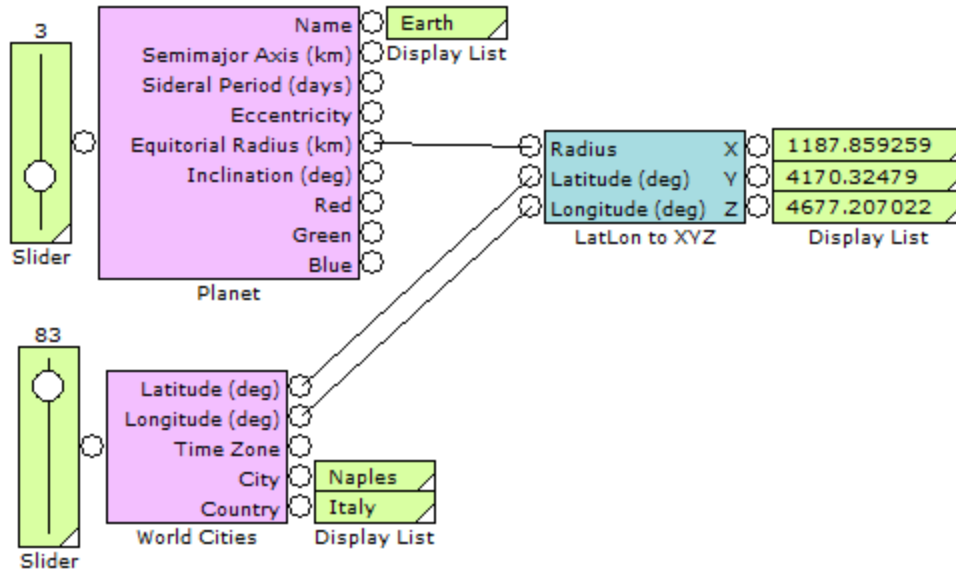
# Lathe Path

The Lathe Path component builds an array of X, Y points to guide a lathe cutter to achieve the input array. Arrays - column 7



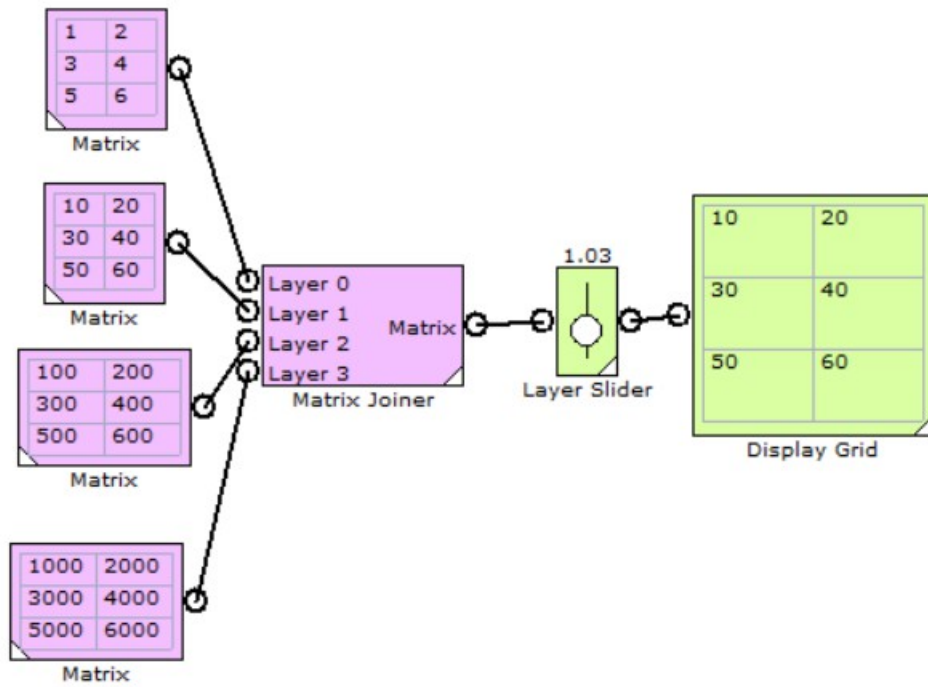
# LatLon to XYZ

The LatLon to XYZ component takes a Radius, Latitude, and Longitude then returns an X, Y, and Z coordinate. 3D - column 6



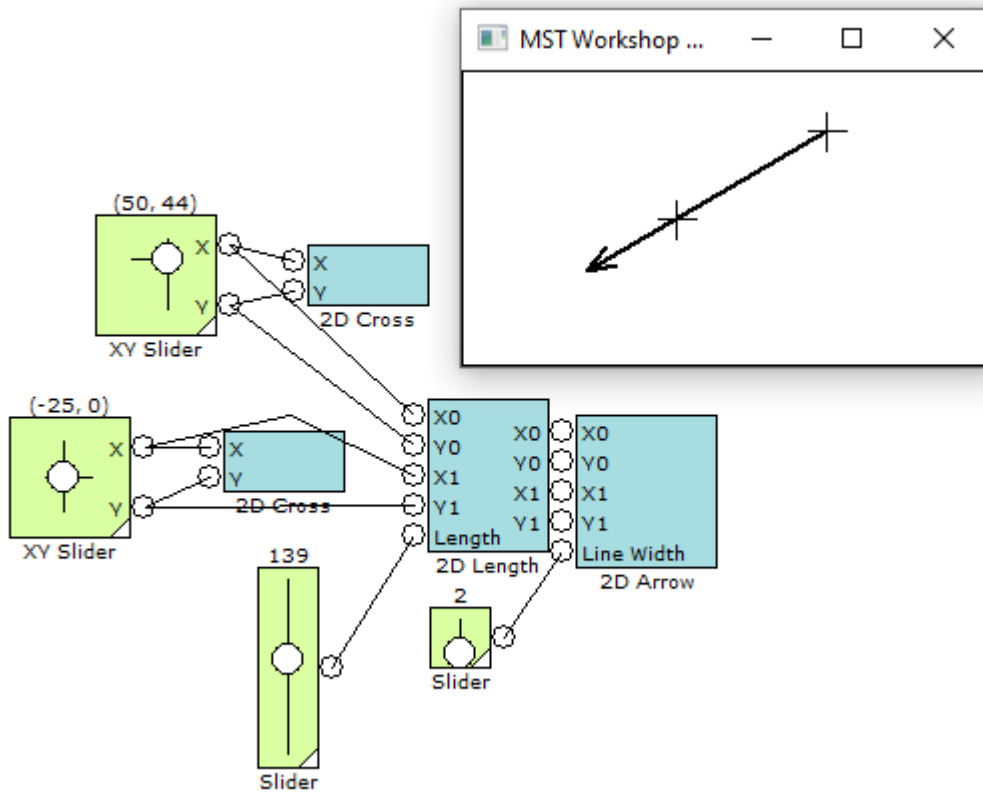
# Layer Slider

The Layer Slider component selects one layer from a 3D array of values and outputs a 2D Matrix of that layer. Controls – column 2.



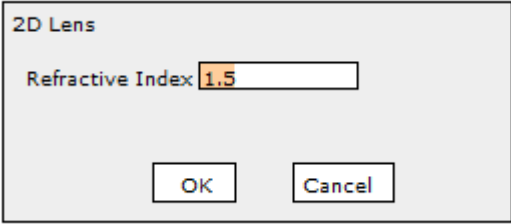
## 2D Length

The 2D Length component takes a vector begin and end coordinates with a length and outputs vector begin and end coordinates of that length. 2D - column 2

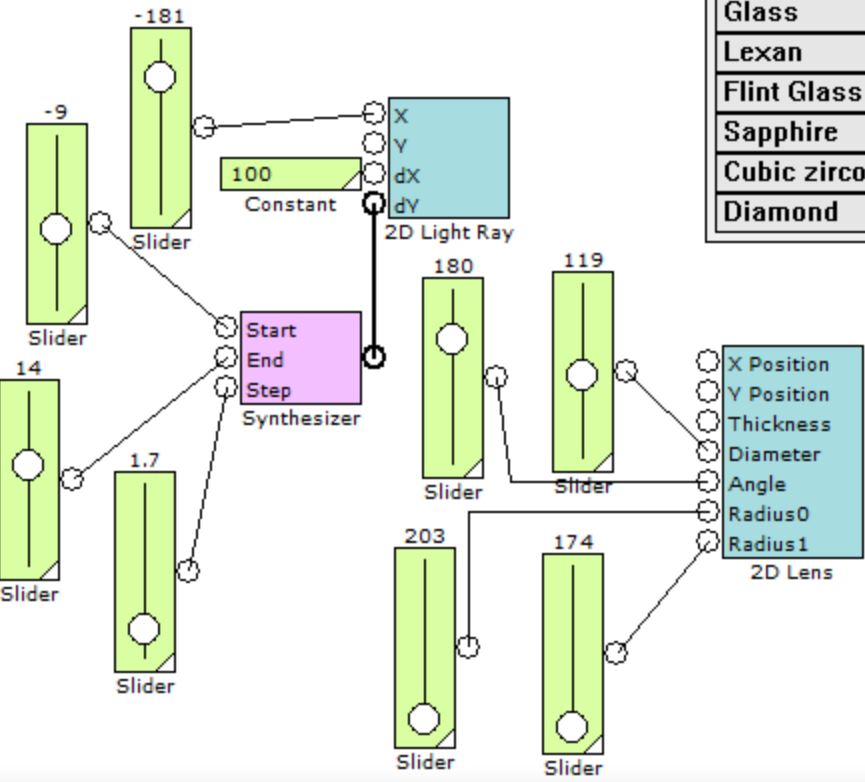


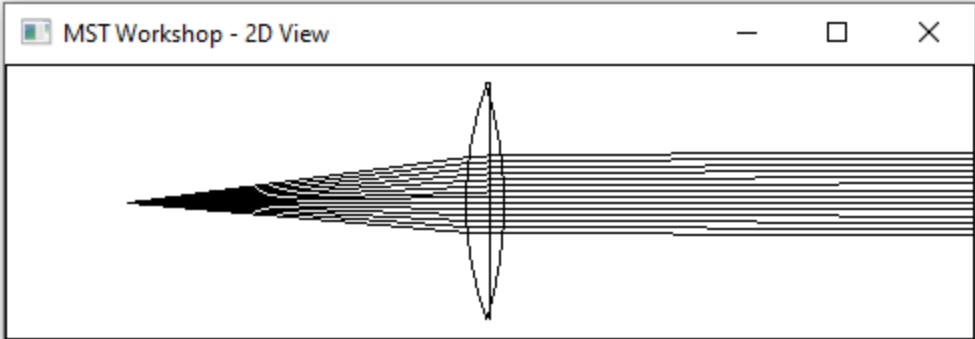
## 2D Lens

The 2D Lens component bends light rays in the 2D view window. The settings determine how the ray progresses through the lens. 2D - column 3



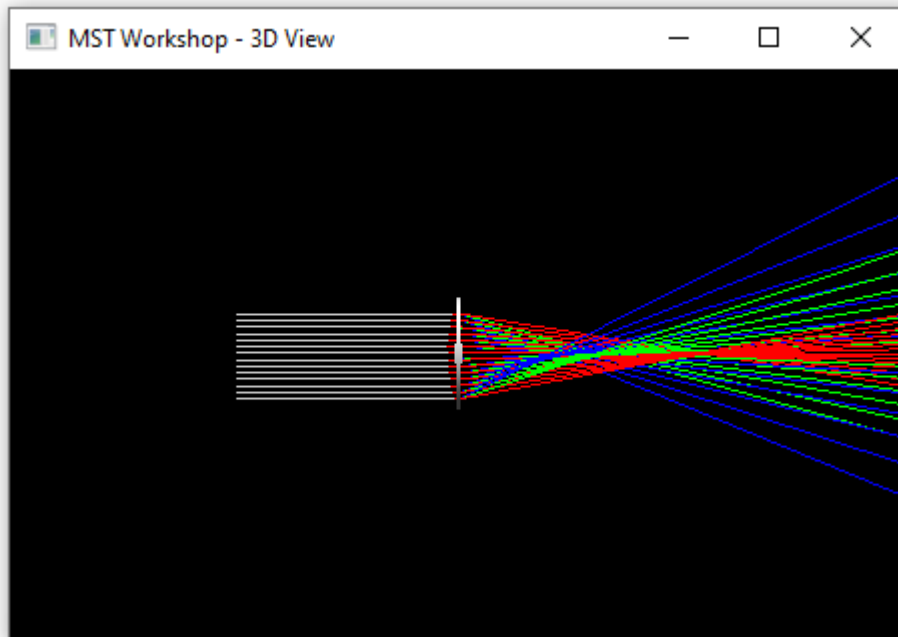
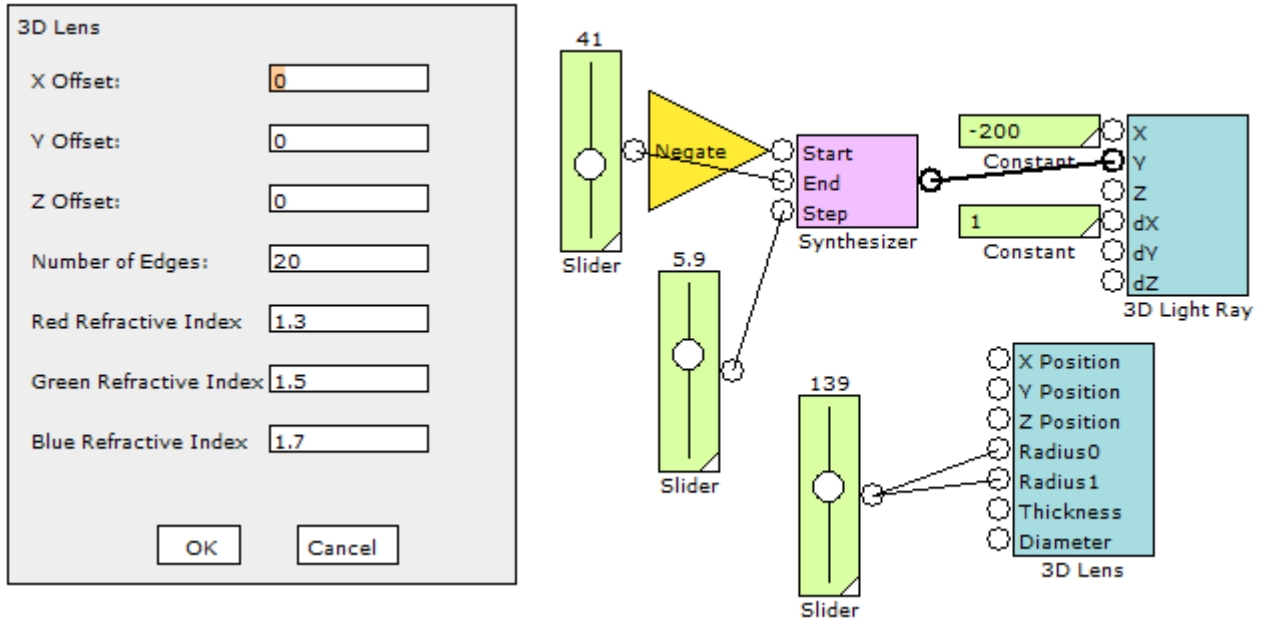
Move To Top	Delete
Move Up	Details...
Move Down	Duplicate
Move To Bottom	Edit Label...
Water	Inputs...
Ice	Options...
Quartz	Rotation...
Plexiglas	
Glass	
Lexan	
Flint Glass	
Sapphire	
Cubic zirconia	
Diamond	





# 3D Lens

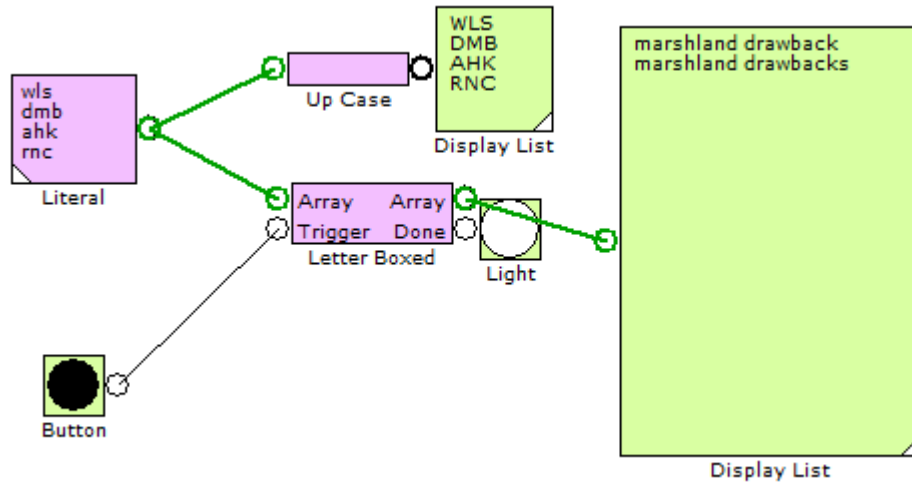
The 3D Lens component bends light rays in the 3D view window. The settings determine how the ray progresses through the lens. The Affine [4x4] array input modifies the size, position and orientation of the lens. 3D - column 5





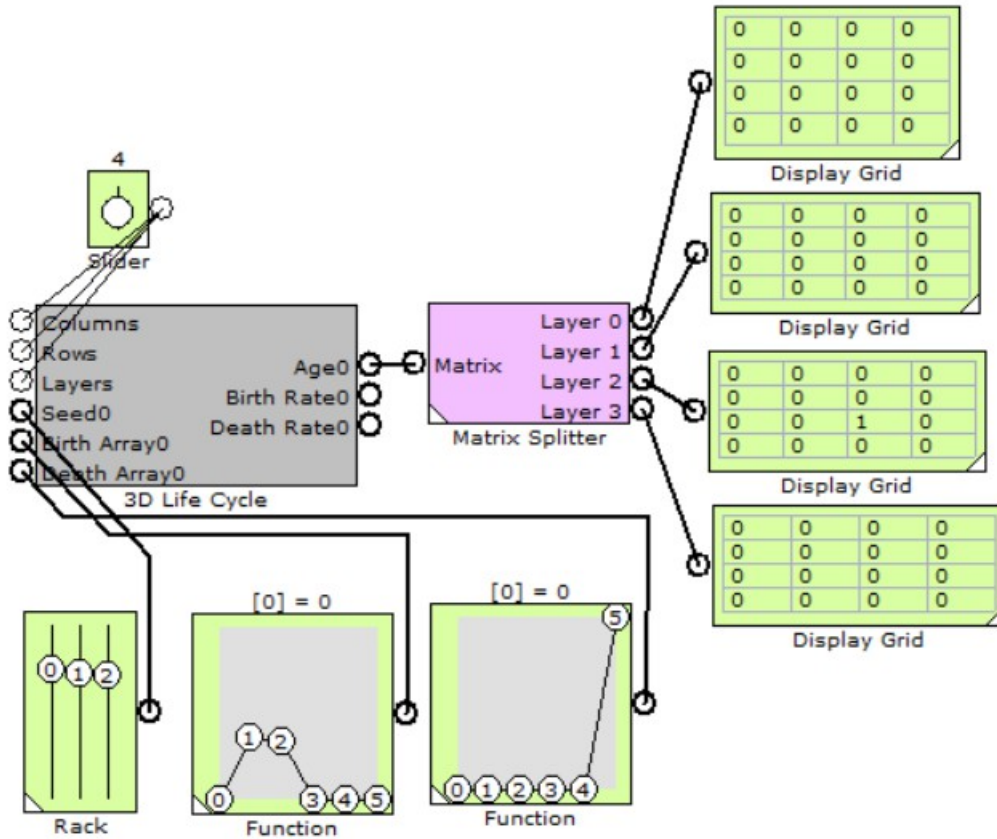
# Letter Boxed

The Letter Boxed solves the Letter Boxed puzzle. Input a string array of characters for each side of the puzzle. The array can be 4 lines of 3 characters each. Set Trigger to true to solve the puzzle.



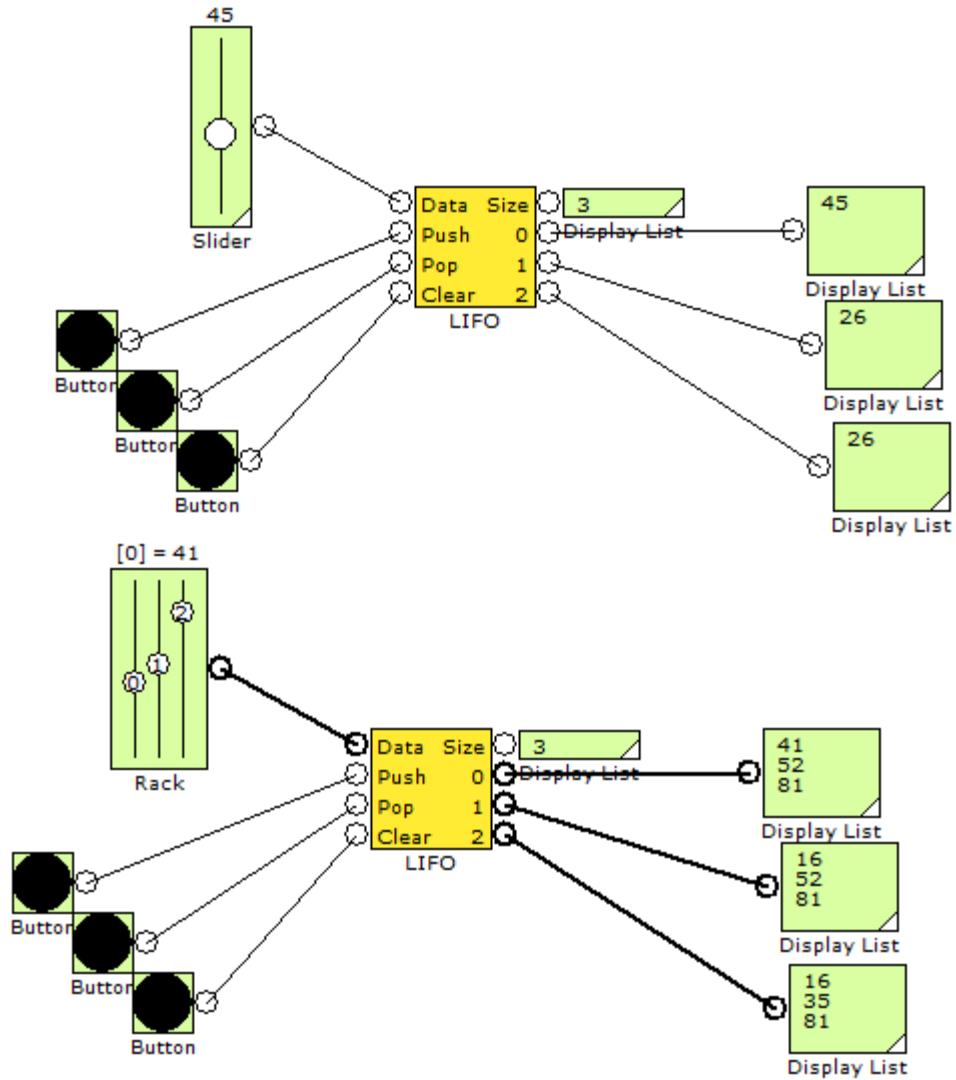
# 3D Life Cycle

The 3D Life Cycle component creates a 3D array of potential life forces. Birth, growth and death proceed based on the input parameters. Arrays – column 2



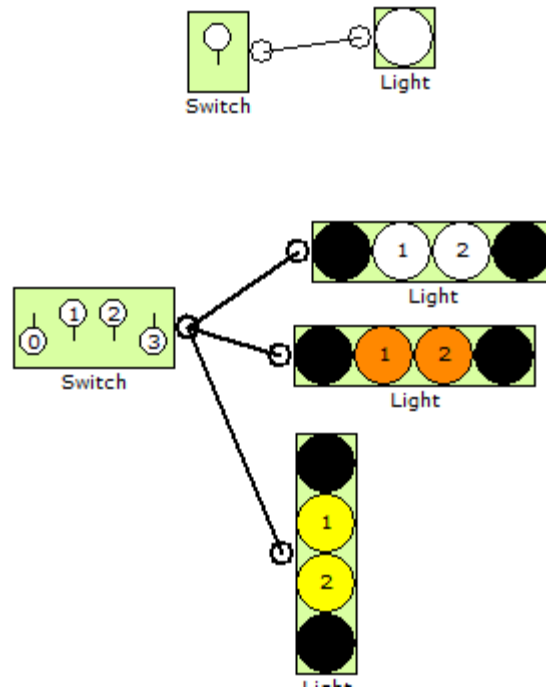
# LIFO

The LIFO component stores data each time the in clock goes true. The output changes when the out clock goes true. Functions - column 4



# Light

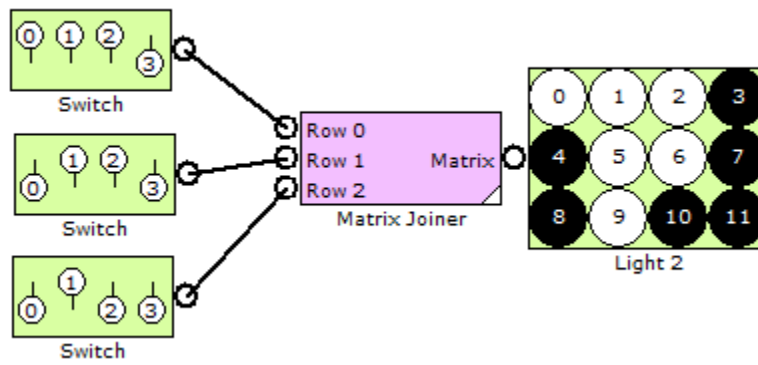
The Light goes on when the input is not zero or logic true. Controls - column 4



## Light 2

The Light 2 goes on when the input is not zero or logic true.

Controls - column 4



## 3D Light

The 3D Light controls various aspects of lighting in the 3D view window. The X, Y, and Z inputs set the position or direction of various aspects of the light. The Affine [4x4] array input modifies the position. 3D - column 5

Pin options:

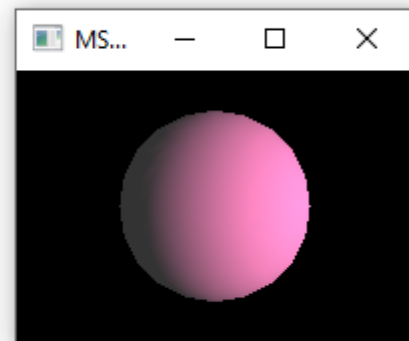
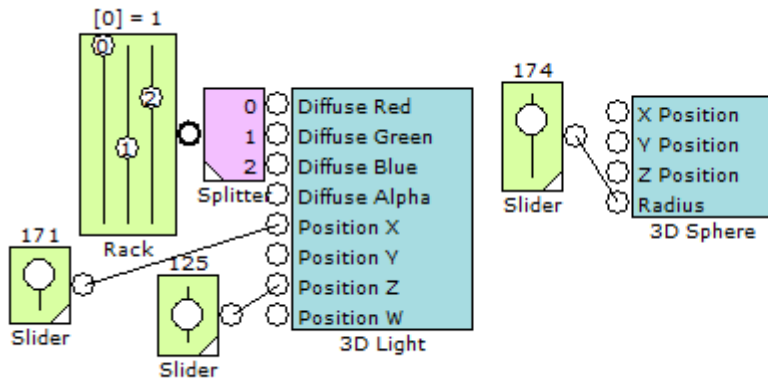
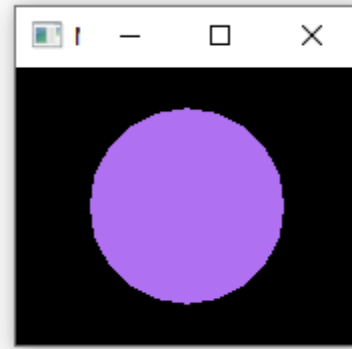
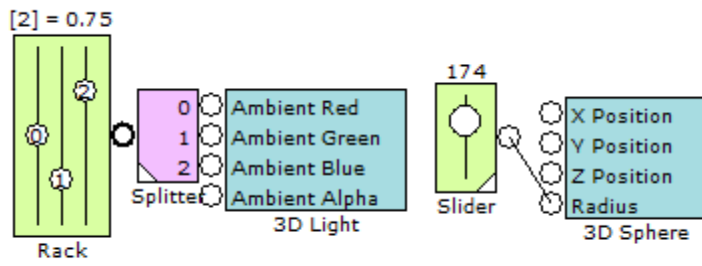
<input type="checkbox"/>	Ambient Red	Ambient Green	Ambient Blue	Ambient Alpha
<input checked="" type="checkbox"/>	Diffuse Red	Diffuse Green	Diffuse Blue	Diffuse Alpha
<input type="checkbox"/>	Specular Red	Specular Green	Specular Blue	Specular Alpha
<input checked="" type="checkbox"/>	Position X	Position Y	Position Z	Position W
<input type="checkbox"/>	Direction X	Direction Y	Direction Z	
<input type="checkbox"/>	Spot Exponent			
<input type="checkbox"/>	Spot Cutoff			

OK Cancel

3D Light

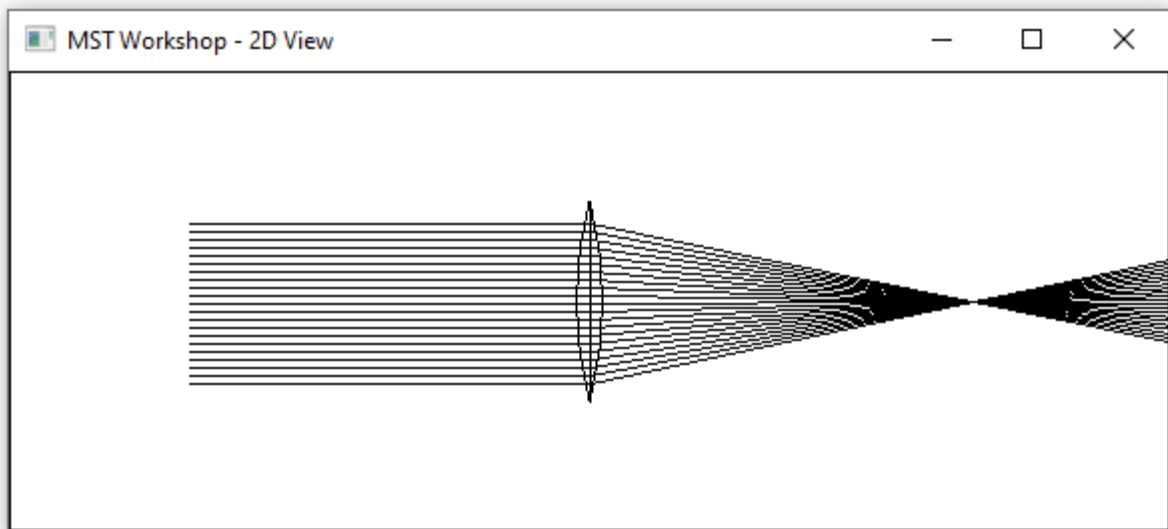
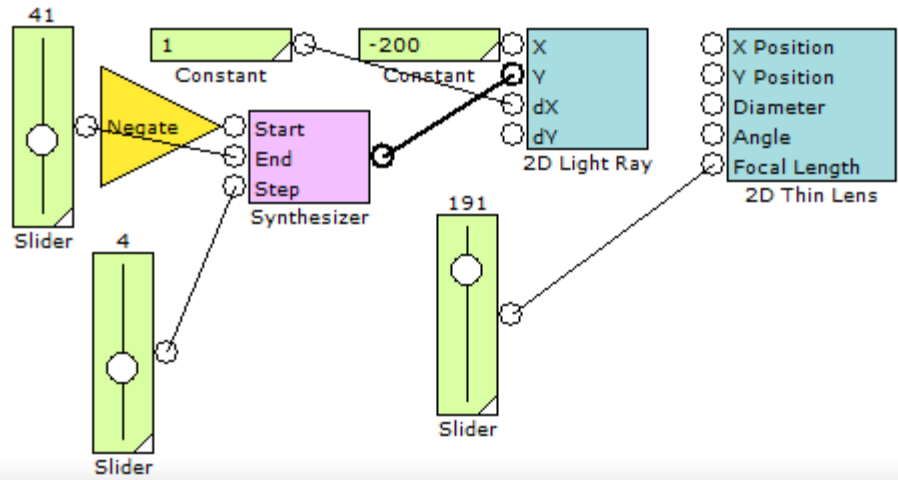
- Light 1
- Light 2
- Light 3
- Light 4
- Light 5
- Light 6
- Light 7
- Light 8
- Light 9

OK Cancel



## 2D Light Ray

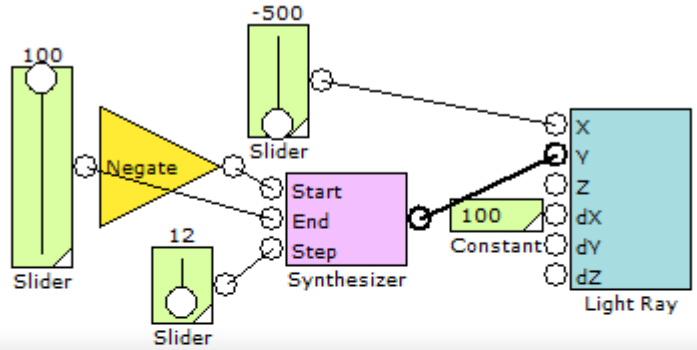
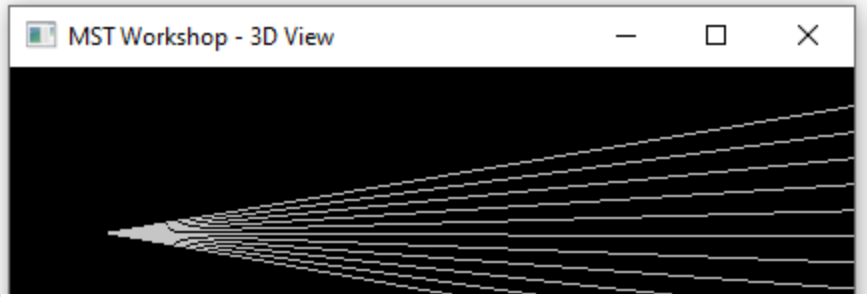
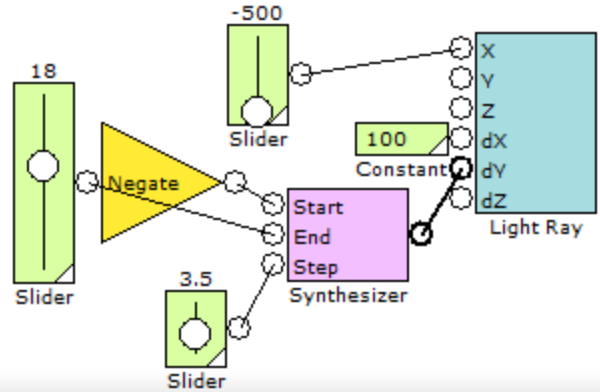
The 2D Light Ray component draws a ray of light in the 3D view window. The origin and direction determine how the ray progresses through the lenses. 2D - column 3





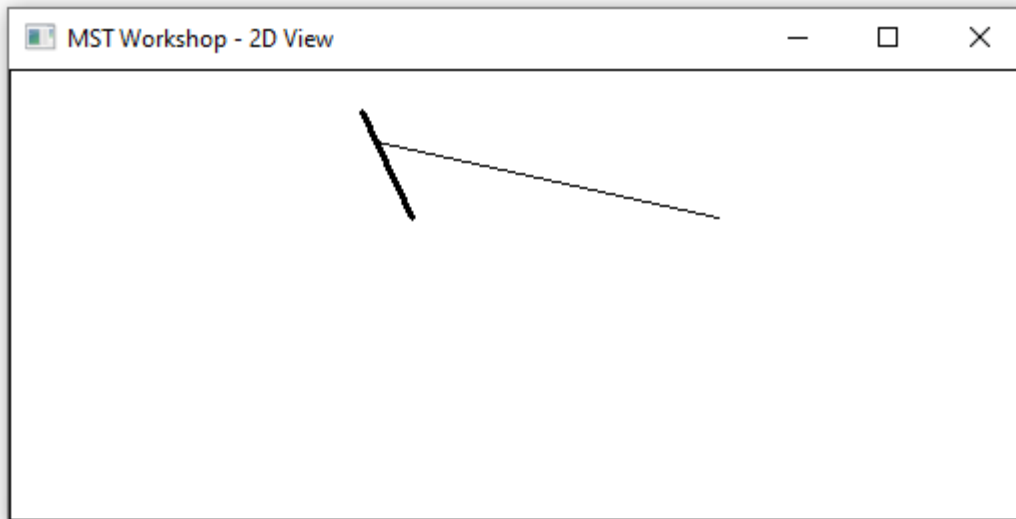
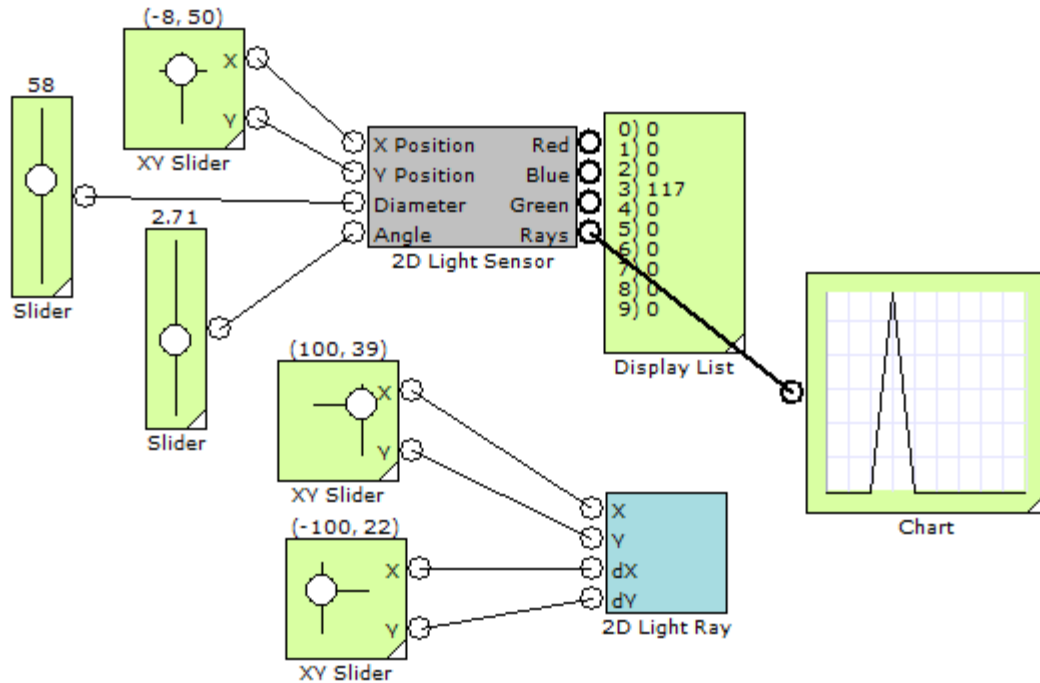
# 3D Light Ray

The 3D Light Ray component draws a ray of light in the 3D view window. The origin and direction determine how the ray progresses through the lenses. The Affine [4x4] array input modifies the size, position and orientation of the ray. 3D - column 5



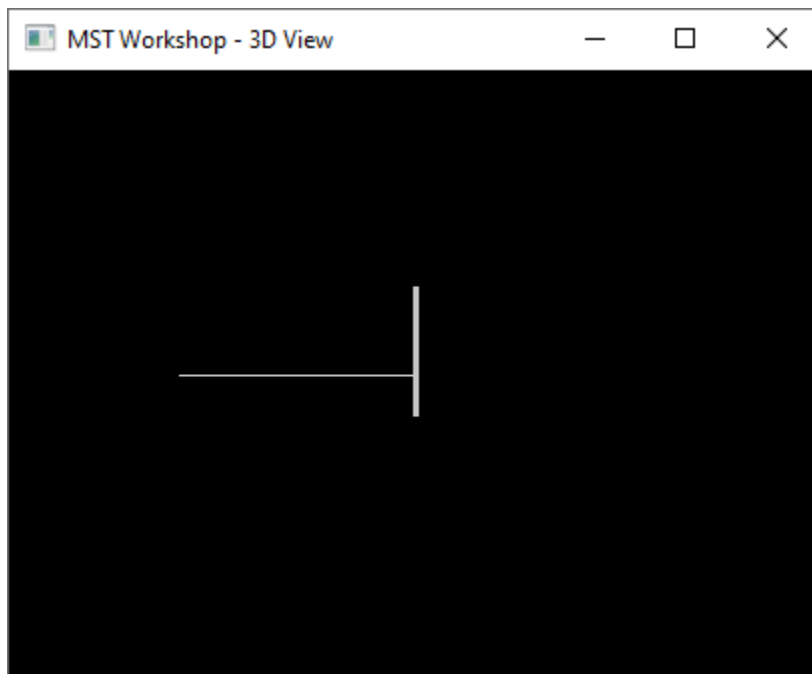
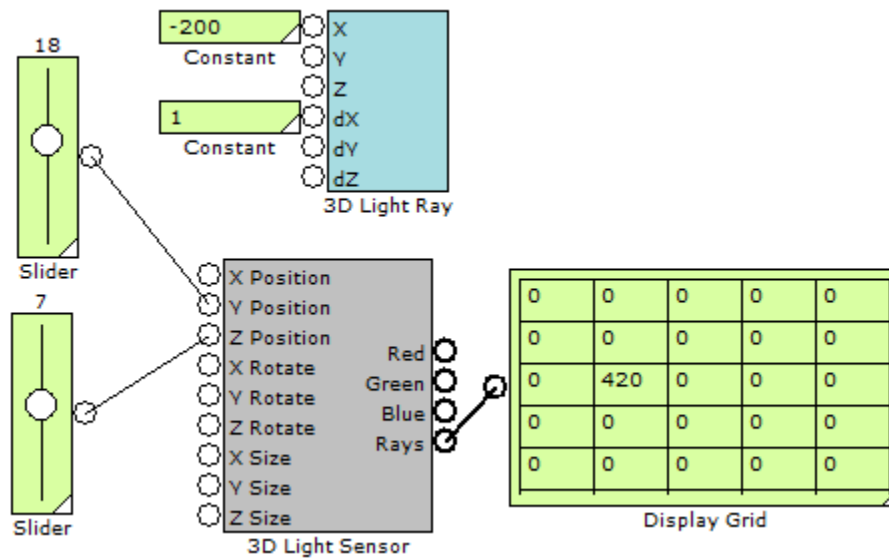
## 2D Light Sensor

The 2D Light Sensor component collects rays of light in the 2D view window. The origin and direction controls the placement of the sensor. 2D - column 3



## 3D Light Sensor

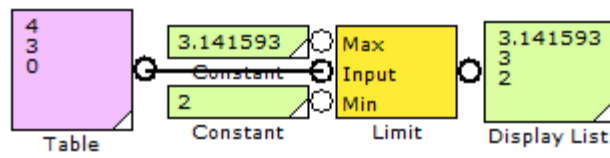
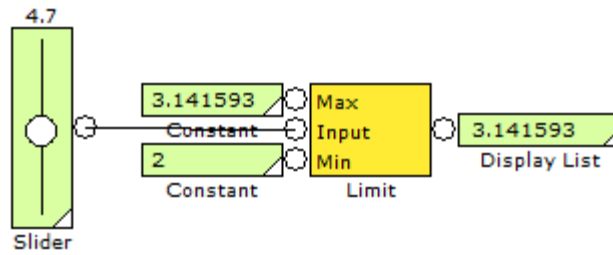
The 3D Light Sensor component collects rays of light in the 3D view window. The origin and direction controls the placement of the sensor. The Affine [4x4] array input modifies the size, position and orientation of the collector. 3D - column 3



# Limit

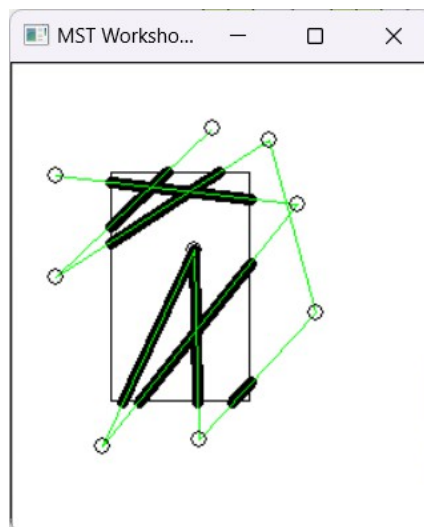
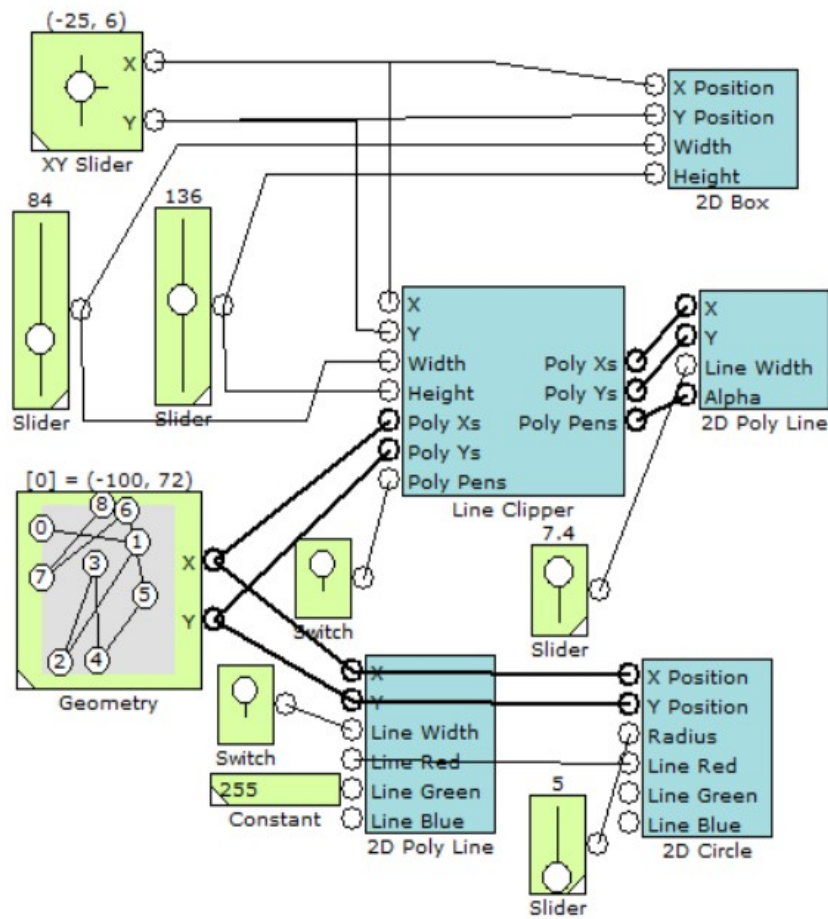
The Limit component limits the output to between the Max and Min values.

Functions - column 2



# Line Clipper

The Line Clipper component clips the input lines to fit exactly in the given rectangle. 2D – column 2



# 3D Line

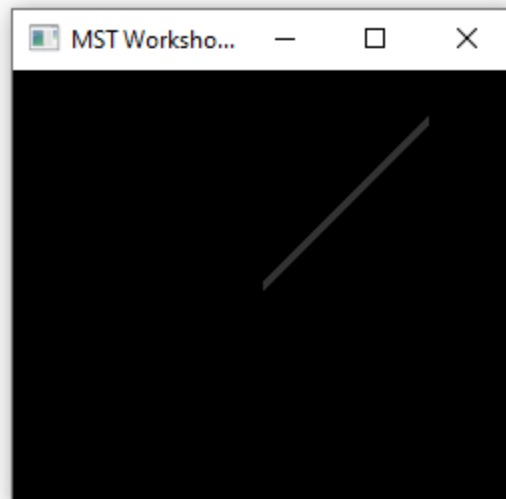
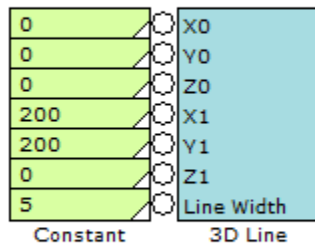
The 3D Line Segment component draws multiple lines in the 3D view window. The X, Y, and Z inputs set the start and end points. The Affine [4x4] array input modifies the size, position and orientation.

3D - column 4

Pin options:

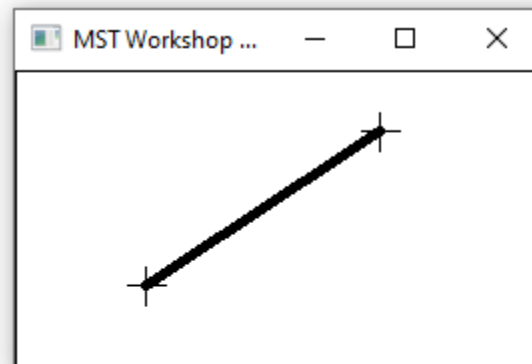
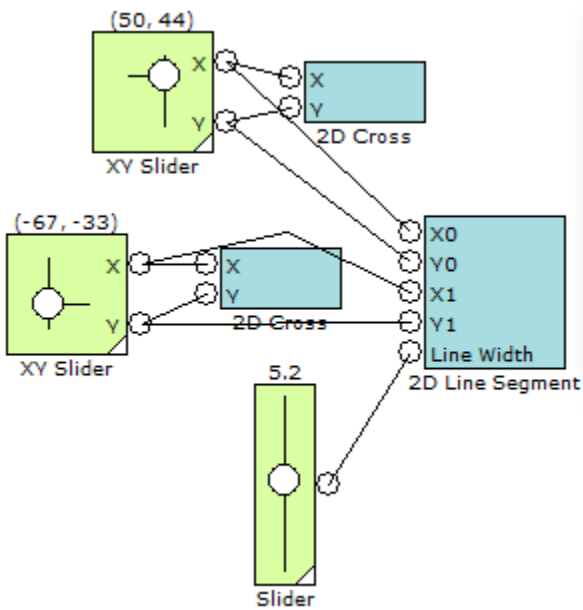
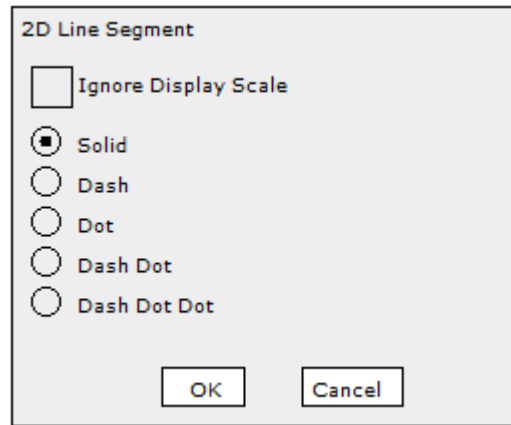
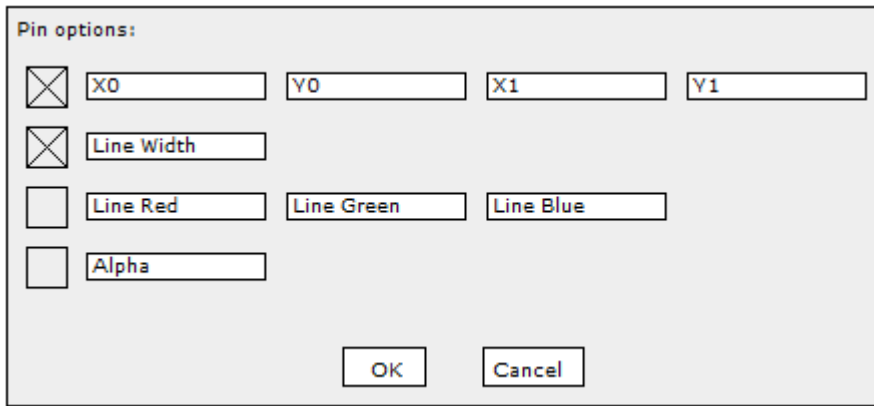
<input checked="" type="checkbox"/>	X0	Y0	Z0
<input checked="" type="checkbox"/>	X1	Y1	Z1
<input checked="" type="checkbox"/>	Line Width		
<input type="checkbox"/>	Red	Green	Blue
<input type="checkbox"/>	Alpha		
<input type="checkbox"/>	[4x4]		

OK Cancel



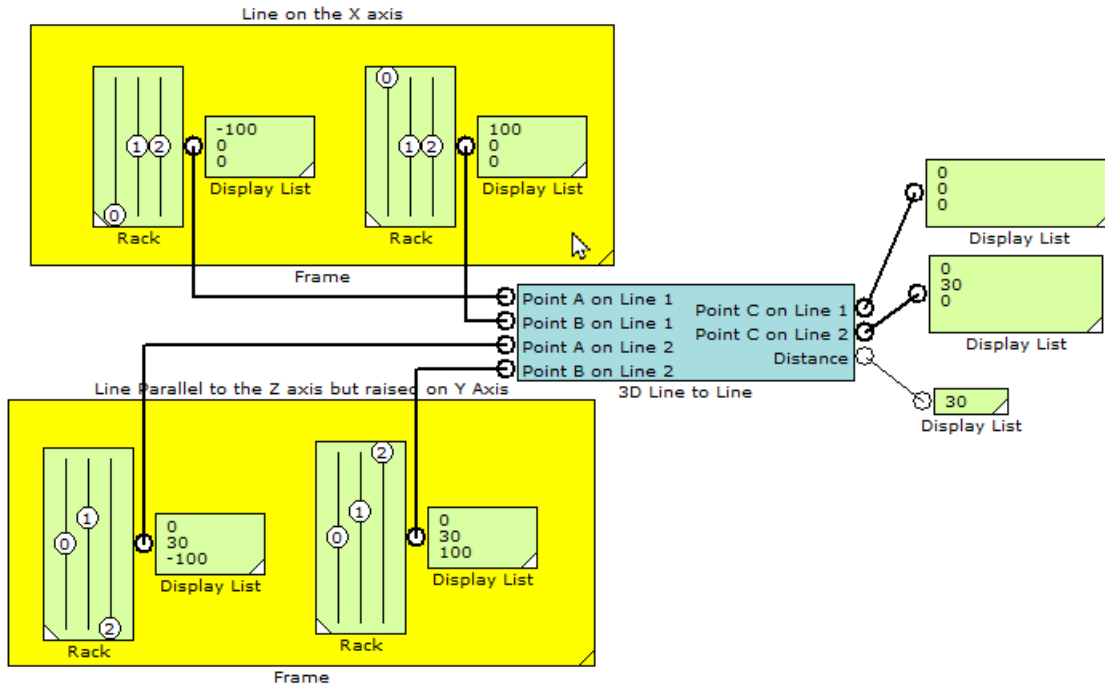
# 2D Line Segment

The 2D Line Segment component draws one or more line segments on the 2D canvas. 2D - column 1



# 3D Line to Line

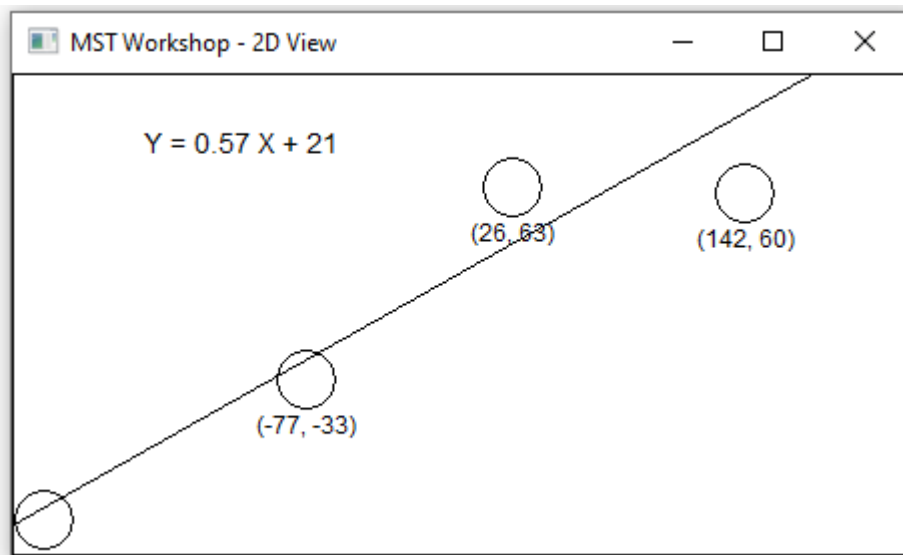
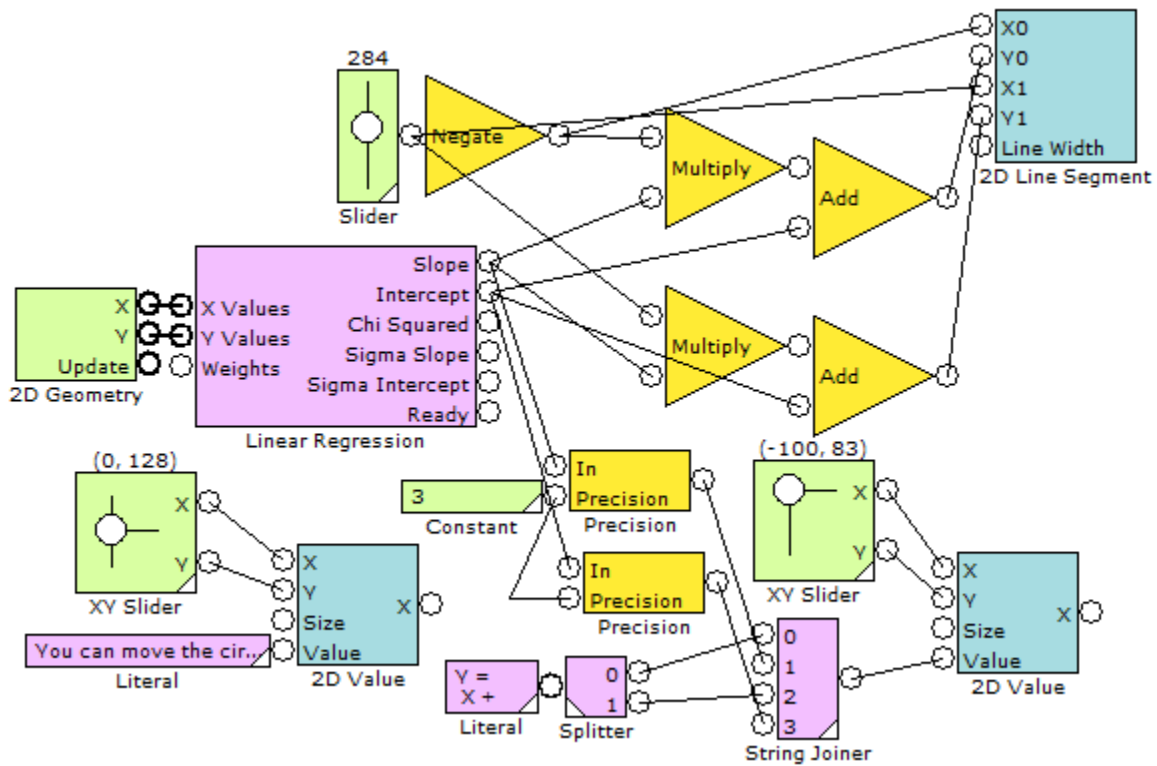
The 3D Line to Line component calculates the distance from a line to another line. It also returns the closest points on each line. 3D - column 2





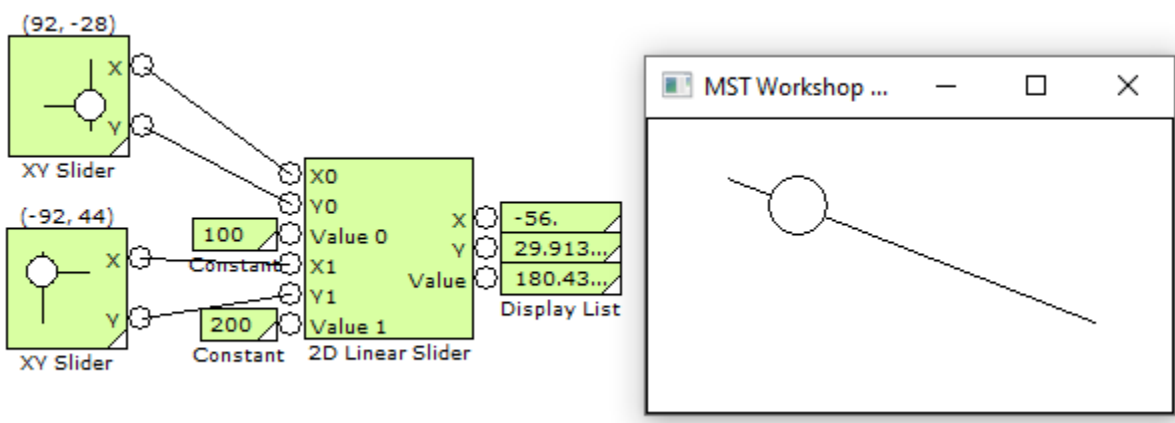
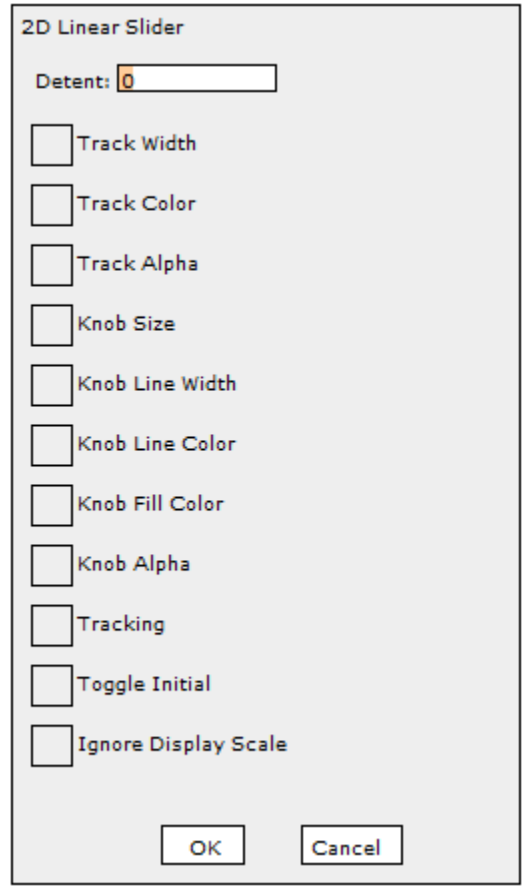
# Linear Regression

The Linear Regression component calculates a linear fit to the input data. Arrays - column 5



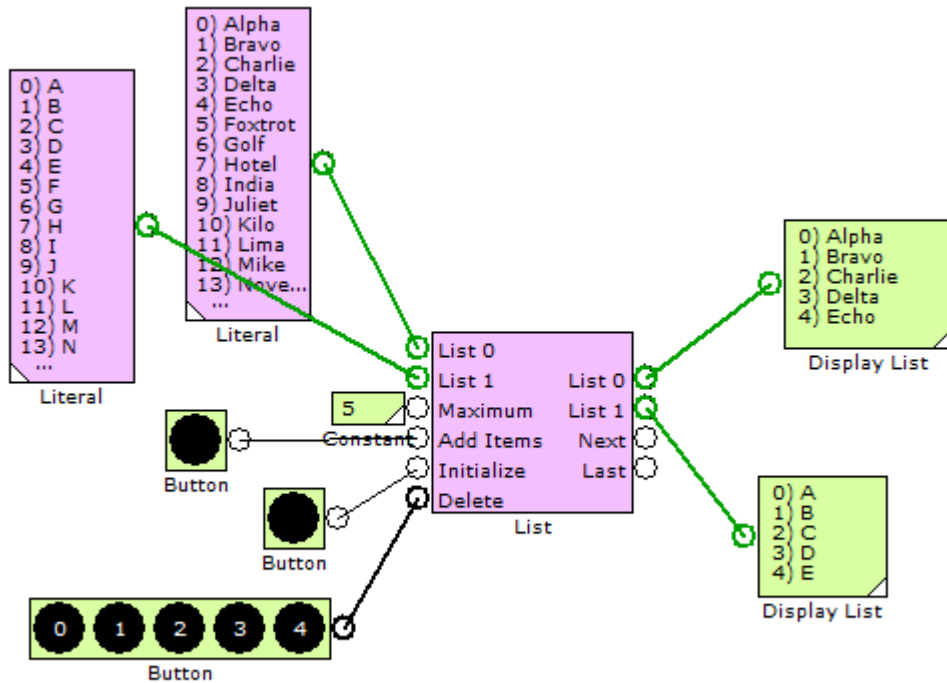
## 2D Linear Slider

The 2D Linear Slider outputs a number based on its dial(s) position(s). Controls - column 2



# List

The List component lets you process lists starting at the beginning of the list. Input one or more lists and they will be output when the Initialize input is triggered. You can restrict the number of items output using the Maximum input. Enable the Add Items input to refill the list after some are deleted. The Delete input indicates by a boolean array which items to delete. Arrays - column 1



# Literal

The Literal component lets you create a list of text strings. Arrays - column 6

Move To Top	Delete
Move Up	Details...
Move Down	Duplicate
Move To Bottom	Edit Label...
Paste w/commas	Set Value Count...
Paste w/tabs	Edit Literal Values...
Paste w/NewLines	Get a File Path...
Read File w/commas...	Alphabet
Read File w/tabs...	Phonetic
Read File w/NewLines...	Greek
US State Names	Cards
US Capital Names	Lorem Ipsum
Stress Test	Pi 100 places
	e 100 places

Literal

Number of Values:

- 0) A
- 1) B
- 2) C
- 3) D
- 4) E
- 5) F
- 6) G
- 7) H
- 8) I
- 9) J
- 10) K
- 11) L
- 12) M
- 13) N
- 14) O
- 15) P
- 16) Q
- 17) R
- 18) S
- 19) T
- 20) U
- 21) V
- 22) W
- 23) X
- 24) Y
- 25) Z

Literal

- 0) Alpha
- 1) Bravo
- 2) Charlie
- 3) Delta
- 4) Echo
- 5) Foxtrot
- 6) Golf
- 7) Hotel
- 8) India
- 9) Juliet
- 10) Kilo
- 11) Lima
- 12) Mike
- 13) November
- 14) Oscar
- 15) Papa
- 16) Quebec
- 17) Romeo
- 18) Sierra
- 19) Tango
- 20) Uniform
- 21) Victor
- 22) Whiskey
- 23) X-ray
- 24) Yankee
- 25) Zulu

Literal

- 0) Alpha
- 1) Beta
- 2) Gamma
- 3) Delta
- 4) Epsilon
- 5) Zeta
- 6) Eta
- 7) Theta
- 8) Iota
- 9) Kappa
- 10) Lambda
- 11) Mu
- 12) Nu
- 13) Xi
- 14) Omicron
- 15) Pi
- 16) Rho
- 17) Sigma
- 18) Tau
- 19) Upsilon
- 20) Phi
- 21) Chi
- 22) Psi
- 23) Omega

Literal

- 0) Alabama
- 1) Alaska
- 2) Arizona
- 3) Arkansas
- 4) California
- 5) Colorado
- 6) Connecticut
- 7) Delaware
- 8) Florida
- 9) Georgia
- 10) Hawaii
- 11) Idaho
- 12) Illinois
- 13) Indiana
- 14) Iowa
- 15) Kansas
- 16) Kentucky
- 17) Louisiana
- 18) Maine
- 19) Maryland
- 20) Massachus...
- 21) Michigan
- 22) Minnesota
- 23) Mississippi
- 24) Missouri
- ...

Literal

- 0) Montgomery
- 1) Juneau
- 2) Phoenix
- 3) Little Rock
- 4) Sacramento
- 5) Denver
- 6) Hartford
- 7) Dover
- 8) Tallahassee
- 9) Atlanta
- 10) Honolulu
- 11) Boise
- 12) Springfield
- 13) Indianapolis
- 14) Des Moines
- 15) Topeka
- 16) Frankfort
- 17) Baton Rou...
- 18) Augusta
- 19) Annapolis
- 20) Boston
- 21) Lansing
- 22) Saint Paul
- 23) Jackson
- 24) Jefferson C...
- ...

Literal

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut lab... ○

Literal

3.1415926535897932384626433832795028841971693993751058209749445923078164062862... ○

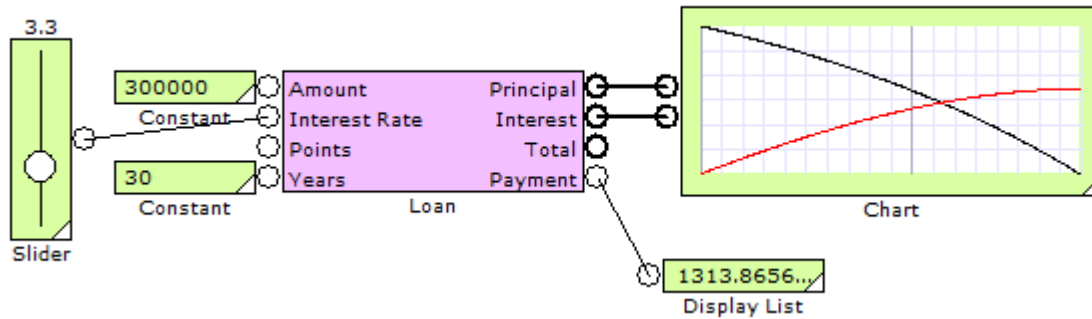
Literal

2.7182818284590452353602874713526624977572470936999595749669676277240766303535... ○

Literal

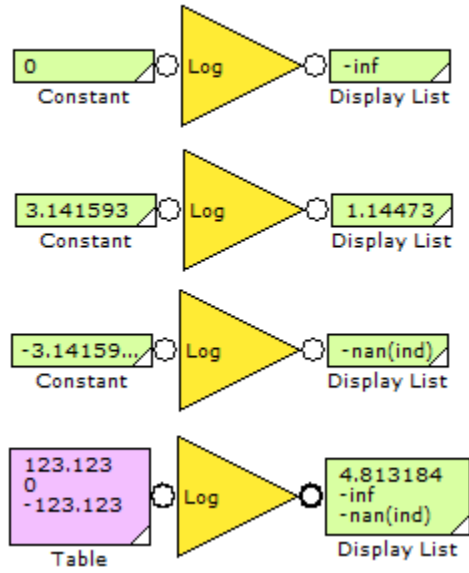
# Loan

The Loan component calculate the monthly payment on an loan. Arrays - column 5



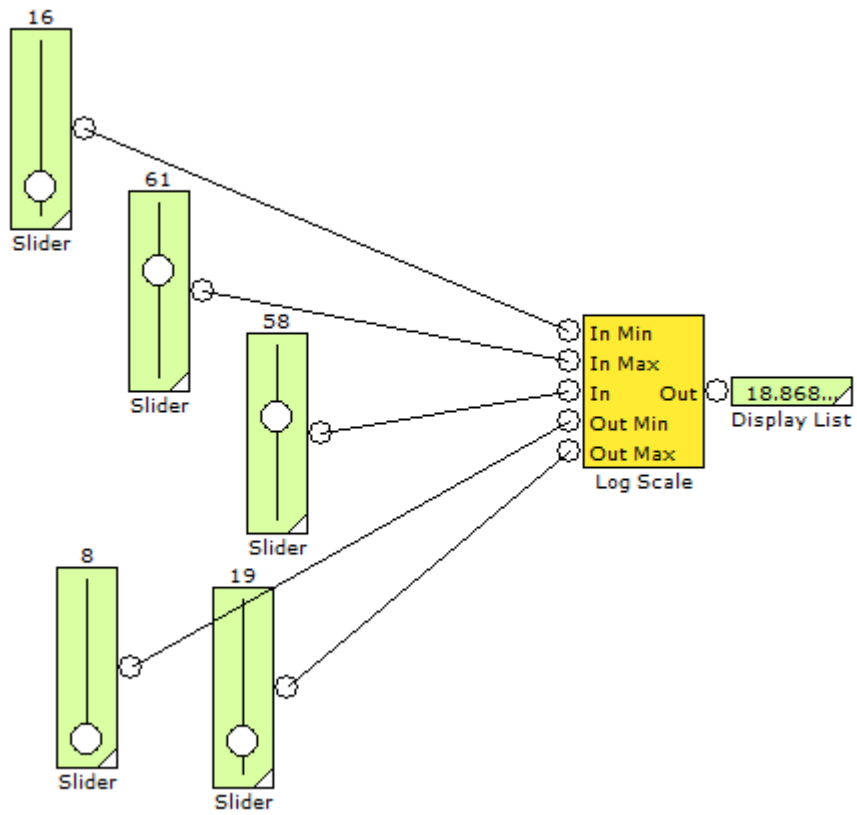
# Log

The Log component calculate the logarithm of the input base e. Input can be scalar or an array.  
Functions - column 1



# Log Scale

The Log Scale component scales its inputs from a linear range to a logarithmic range. Functions - column 5



# Logic

The Logic component lets you create advanced logic in a single component.

Functions - column 3

Move To Top	Delete
Move Up	Details...
Move Down	Duplicate
Move To Bottom	Edit Label...
4 input AND	Set Options...
4 input OR	Edit internal logic...
D type Latch	
4 Bit Rotate	
Full Adder	
SR Flip Flop	

Logic Options:

Number of Inputs:

Number of Outputs:

Number of Logic Lines:

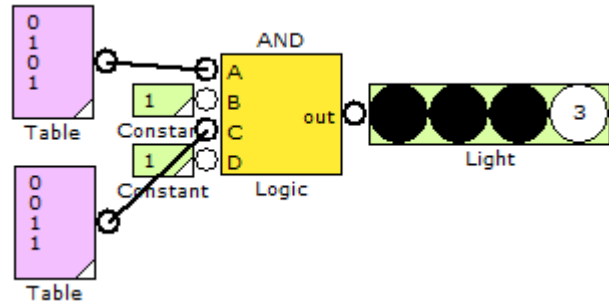
OK Cancel

## 4 input AND

Logic Lines:

In 0:	In 1:	In 2:	In 3:	Out 0:
A	B	C	D	out
Line 0:	True	True	True	True
Line 1:				False

OK Cancel

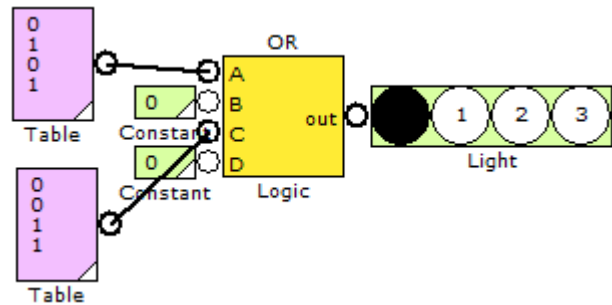


## 4 input OR

Logic Lines:

In 0:	In 1:	In 2:	In 3:	Out 0:
A	B	C	D	out
Line 0:	False	False	False	False
Line 1:				True

OK Cancel



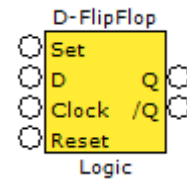


## D type Latch

Logic Lines:

	In 0:	In 1:	In 2:	In 3:	Out 0:	Out 1:
	Set	D	Clock	Reset	Q	/Q
Line 0:	True			False	True	False
Line 1:	False			True	False	True
Line 2:	False	True	Rise	False	True	False
Line 3:	False	False	Rise	False	False	True

OK Cancel

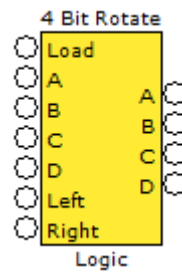


## 4 Bit Rotate

Logic Lines:

	In 0:	In 1:	In 2:	In 3:	In 4:	In 5:	In 6:	Out 0:	Out 1:	Out 2:	Out 3:
	Load	A	B	C	D	Left	Right	A	B	C	D
Line 0:	Rise					False	False	In 1	In 2	In 3	In 4
Line 1:	False					Rise	False	Out 1	Out 2	Out 3	Out 0
Line 2:	False					False	Rise	Out 3	Out 0	Out 1	Out 2

OK Cancel

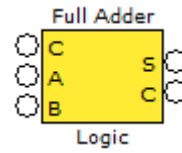


## Full Adder

Logic Lines:

	In 0:	In 1:	In 2:	Out 0:	Out 1:
	C	A	B	S	C
Line 0:	False	False	False	False	False
Line 1:	True	False	False	True	False
Line 2:	False	True	False	True	False
Line 3:	True	True	False	False	True
Line 4:	False	False	True	True	False
Line 5:	True	False	True	False	True
Line 6:	False	True	True	False	True
Line 7:	True	True	True	True	True

OK Cancel

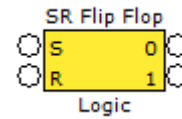


## SR Flip Flop

Logic Lines:

	In 0:	In 1:	Out 0:	Out 1:
	S	R	0	1
Line 0:	Rise		True	False
Line 1:		Rise	False	True

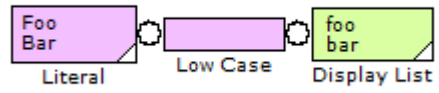
OK Cancel



## Low Case

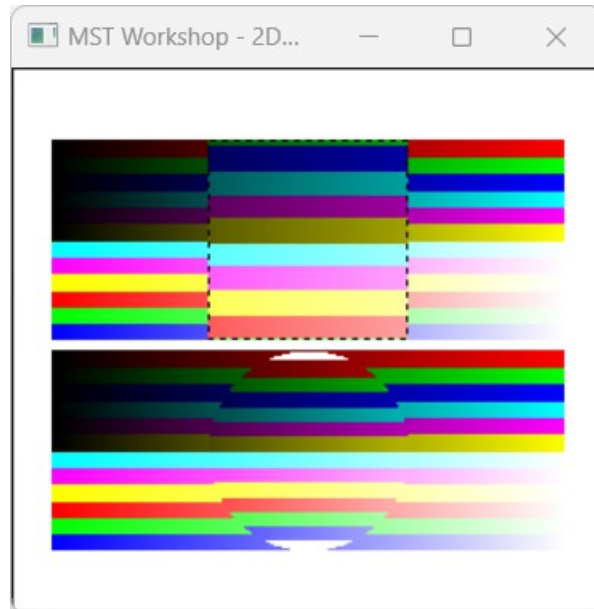
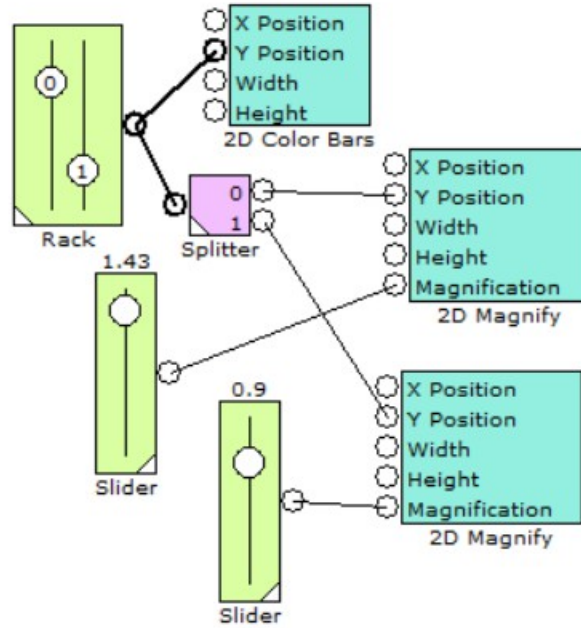
The Low Case component changes the input string to lower case characters.

Arrays - column 6



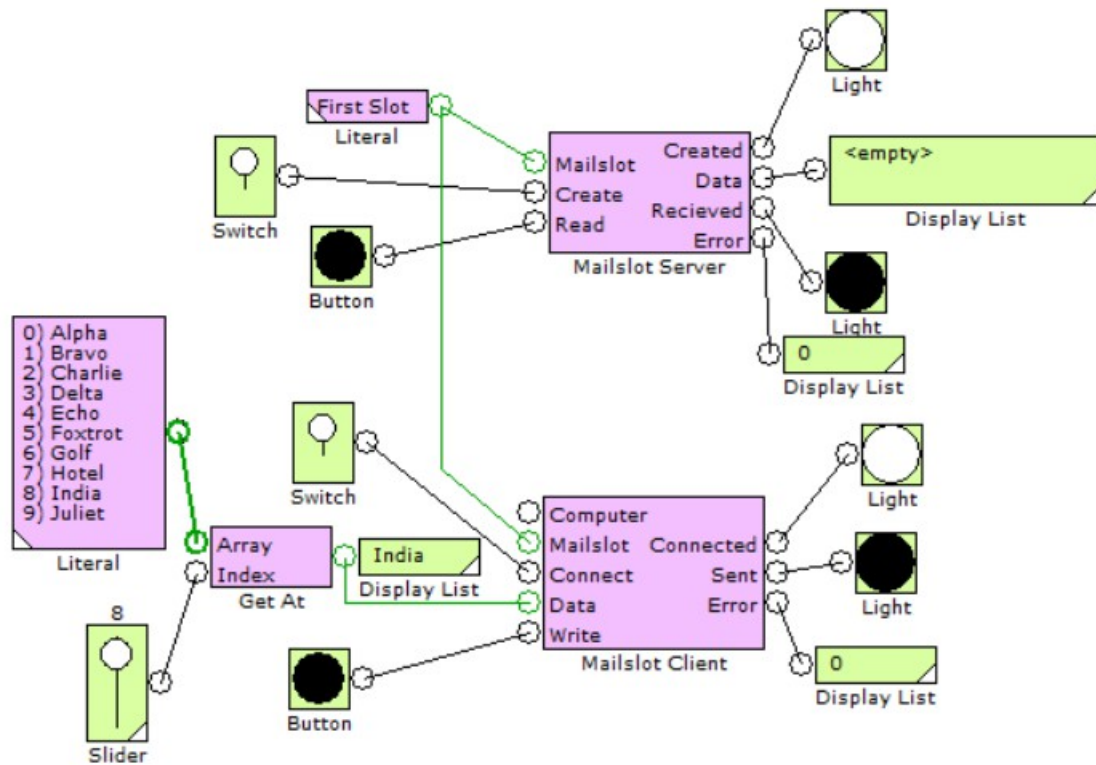
# 2D Magnify

The 2D Magnify magnifies the image in the 2D drawing it covers. 2D – column 1



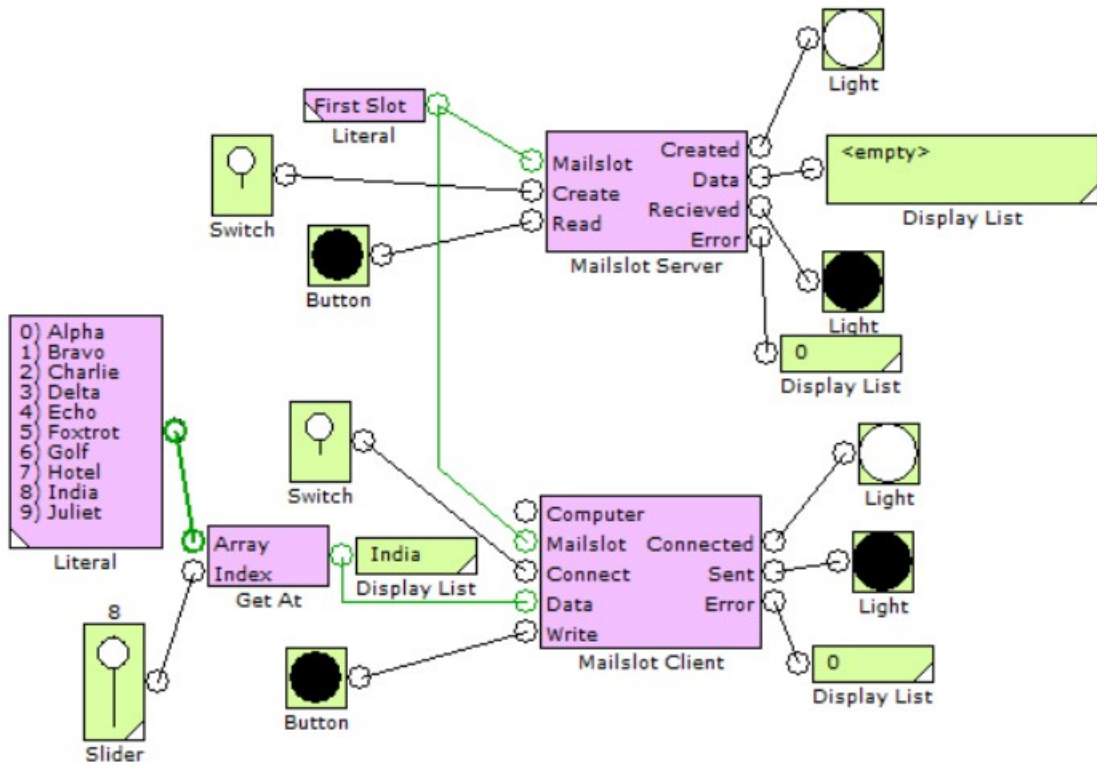
# Mailslot Client

The Mailslot Client component can open a Mailslot and write text to it. Mailslots can be on the same computer or other computers on the same local network. Arrays - column 6



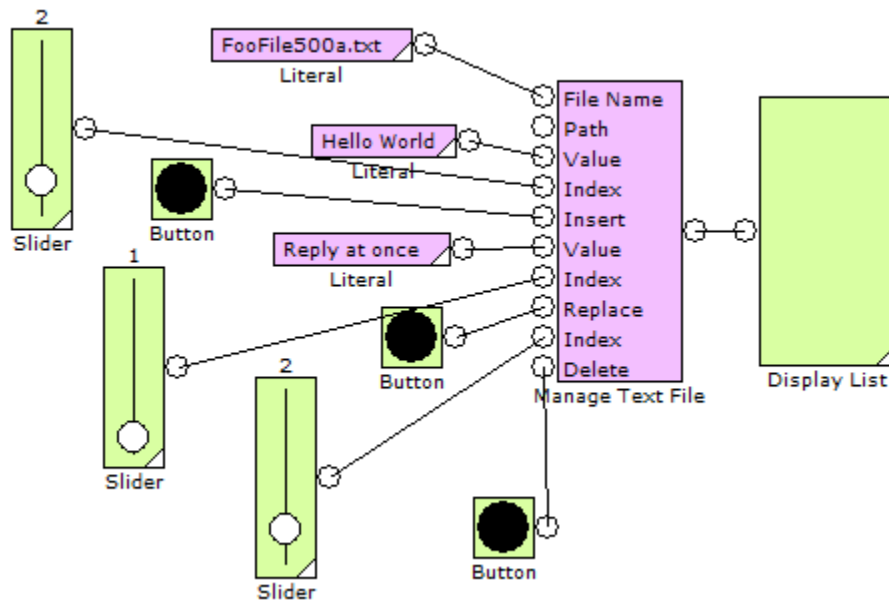
# Mailslot Server

The Mailslot Server component can create Mailslots and read from them. Mailslots can be on the same computer or other computers on the same local network. Arrays - column 6



## Manage Text File

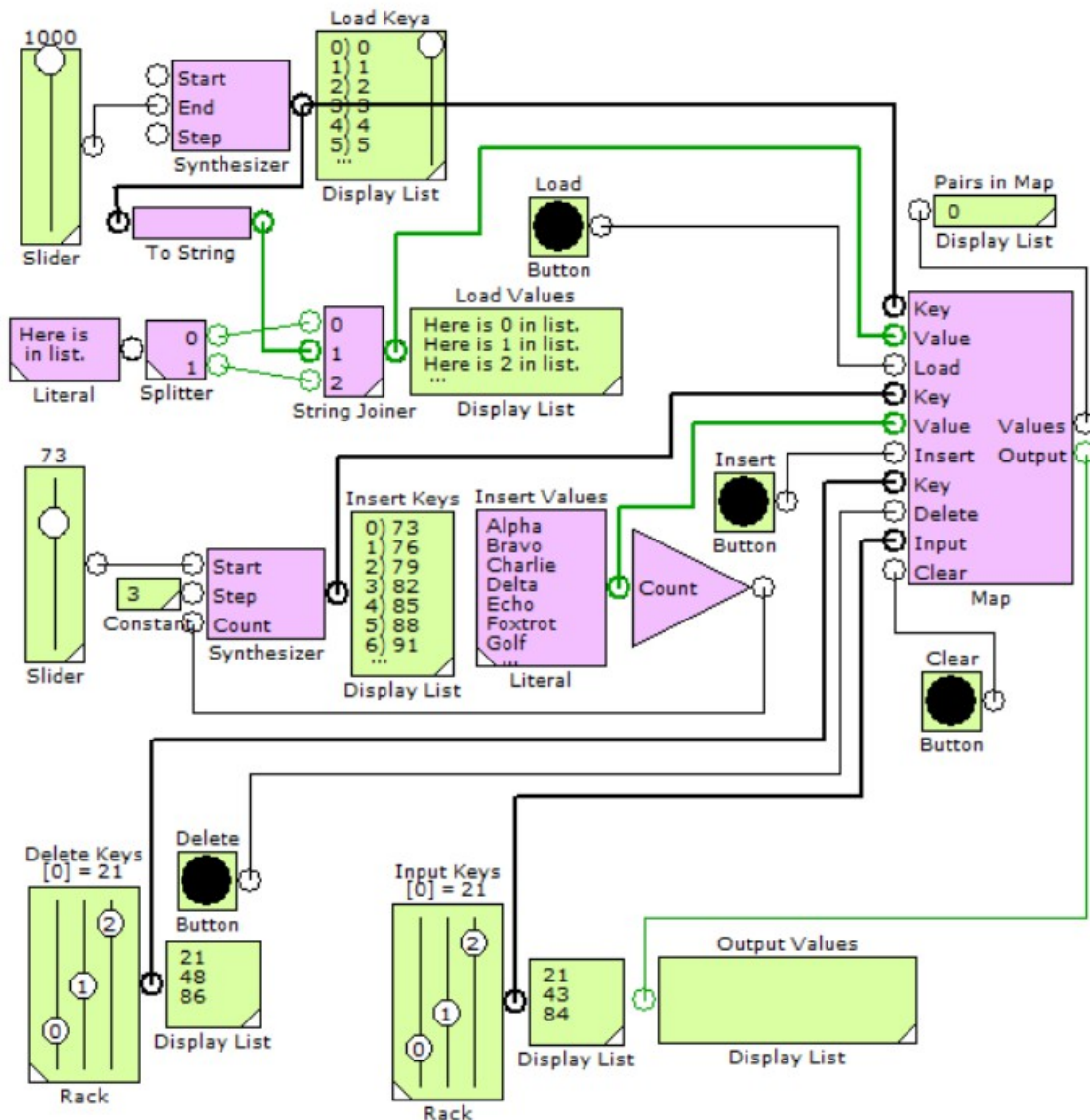
The Manage Text File component creates a text file. Use the Value, Index and Insert inputs to add new values. Use the Value, Index and Replace inputs to replace a value. Use the Index and Delete inputs to delete a value. Arrays - column 7



# Map

The Map component holds a list of Key Value pairs. Keys are integers and Values are strings. You can load the pairs all at once with the Load command or insert and delete them one at a time or in groups with the Insert and Delete commands. Clear empties the Map completely. A key(s) at the Input will result in the corresponding Value(s) at the Output. Keys are unique, so inserting duplicate keys will be ignored.

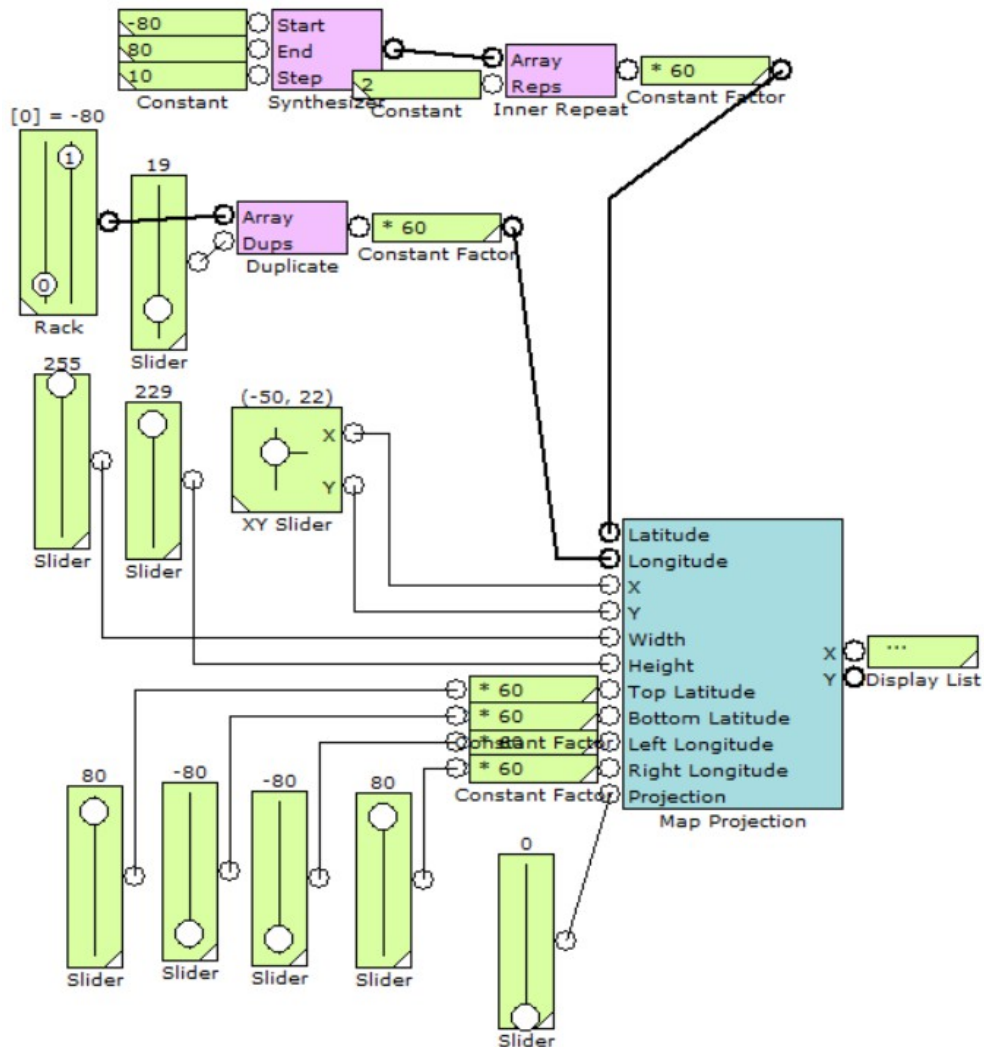
Arrays – column 1





# Map Projection

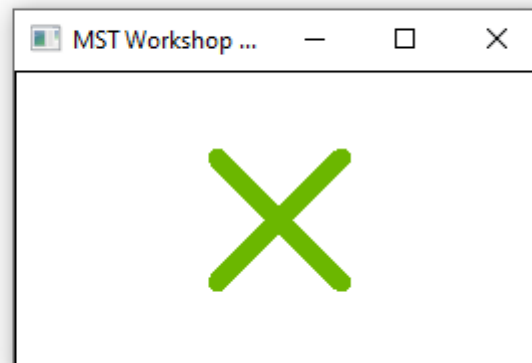
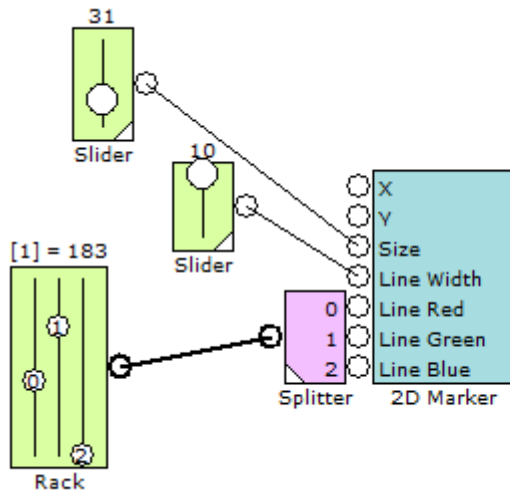
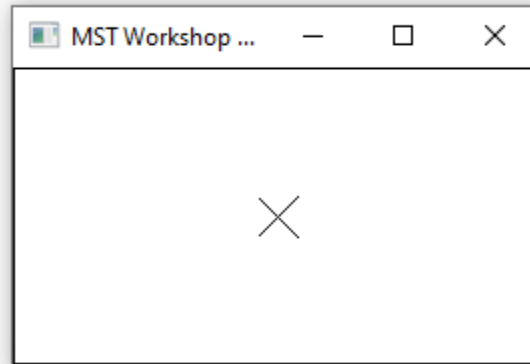
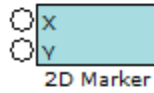
The Map Projection component projects a drawing from Latitude, Longitude coordinates to X, Y coordinates. 2D – column 1



## 2D Marker

The 2D Marker component draws an X at the X, Y location.

2D - column 1



# Matrix

The Matrix component is used to create a two dimensional matrix of numbers. Arrays - column 1

Move To Top	Delete
Move Up	Details...
Move Down	Duplicate
Move To Bottom	Edit Label...
Copy	Set Matrix Size...
Paste	Edit Matrix Values...
Load Resistor	Load Zener Diode
Load Inductor	Load Polar Capacitor
Load Diode	Load Ground
Load Capacitor	Load AC Source
Load Battery	

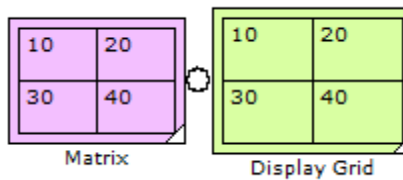
Matrix

Columns

Rows

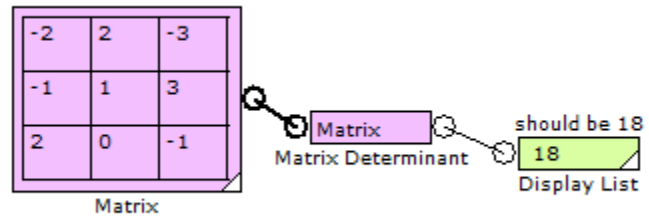
Matrix:

	Col 1:	Col 2:
Row 0:	<input type="text" value="10"/>	<input type="text" value="20"/>
Row 1:	<input type="text" value="30"/>	<input type="text" value="40"/>



## Matrix Determinant

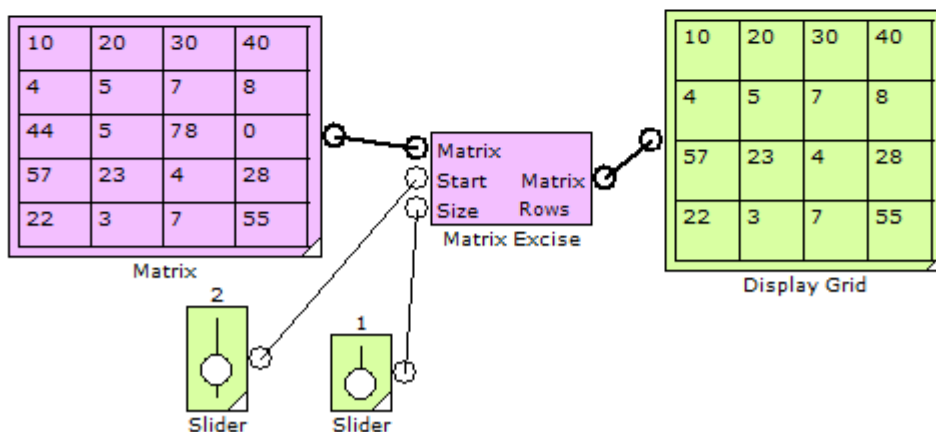
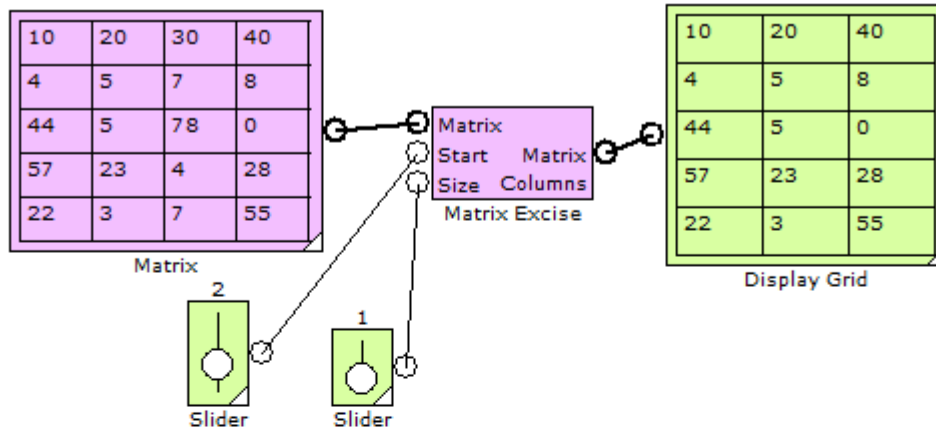
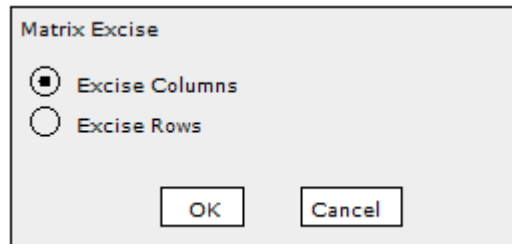
The Matrix Determinant component returns the determinant of a matrix. Input matrix must be square.  
Arrays - column 4



# Matrix Excise

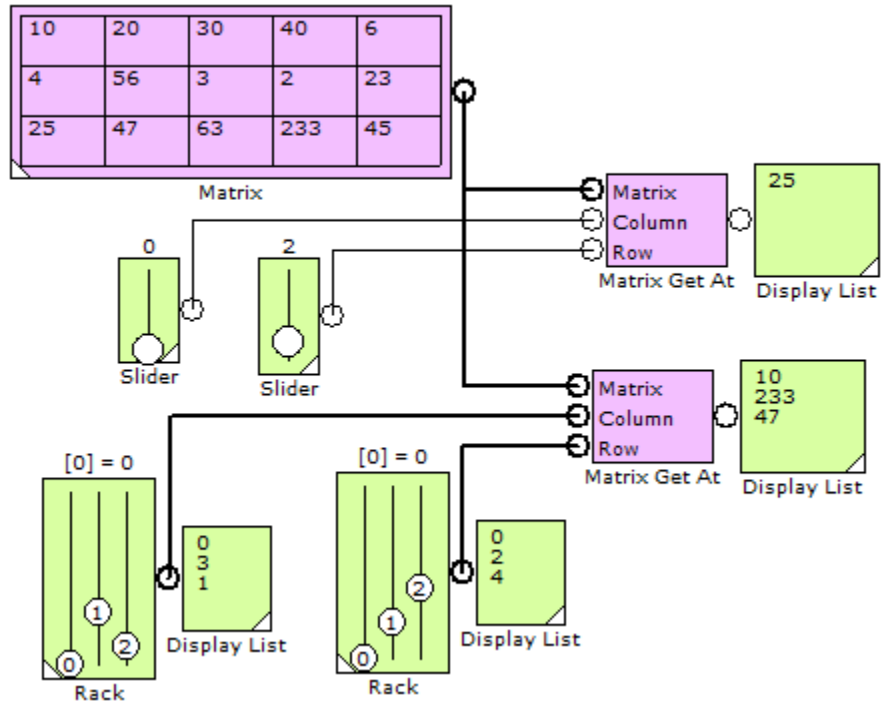
The Matrix Excise component removes a portion of the array. From the menu choose either X or Y direction. Input the number of the first value in X or Y, and the size in X or Y you want to remove.

Arrays - column 4



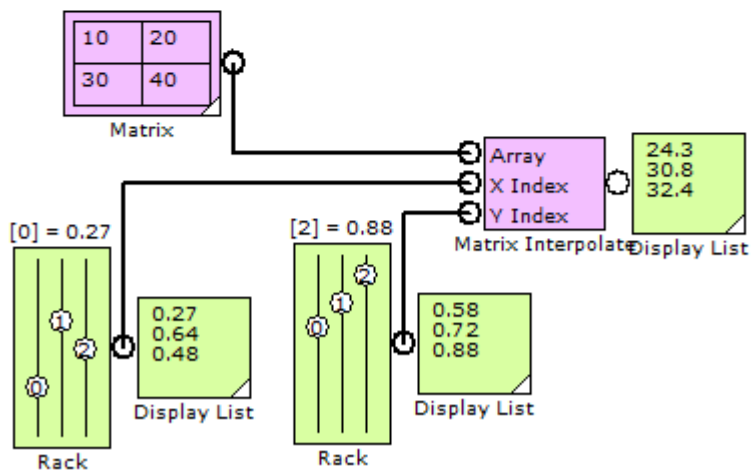
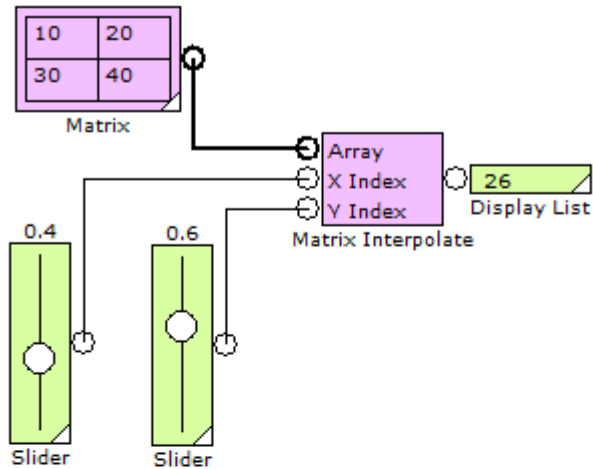
# Matrix Get At

The Matrix Get At component gets the Array value at the X and Y Index inputs. It outputs the value(s) at the positions given by the XY Index(s). Arrays - column 4



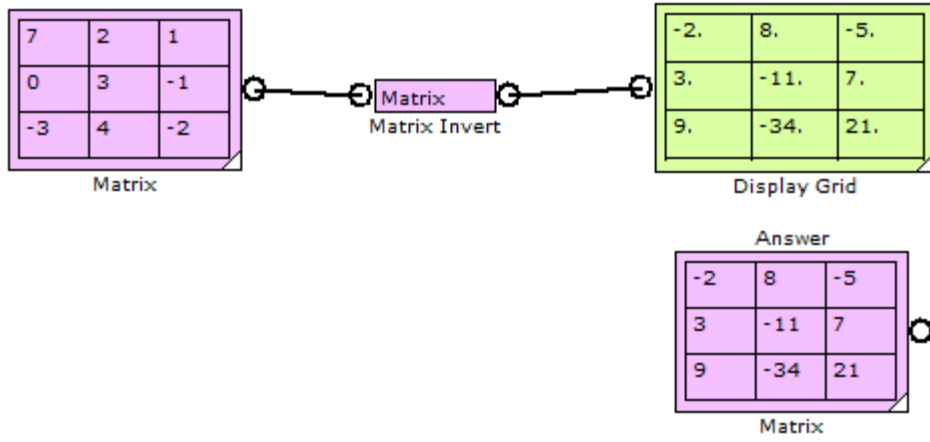
# Matrix Interpolate

The Matrix Interpolate component gets the Matrix values around the Indexes input and interpolates. It outputs the value(s) at the xy positions given by the Index(s). Arrays - column 4



# Matrix Invert

The Matrix Invert component computes the inverse of a square matrix. Arrays - column 4

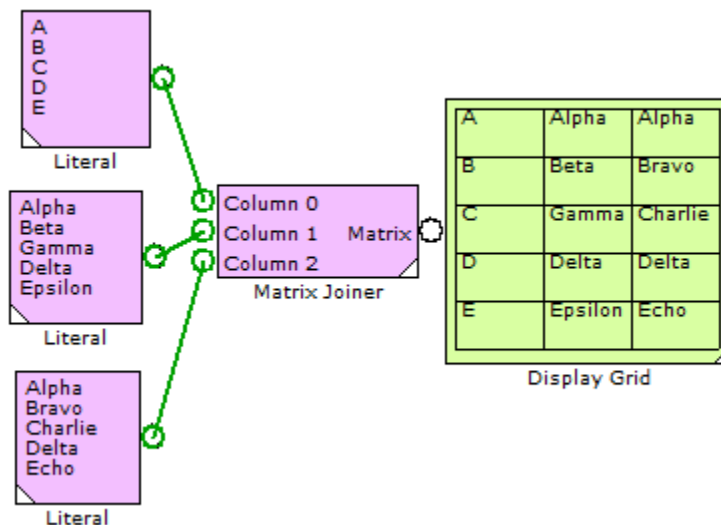
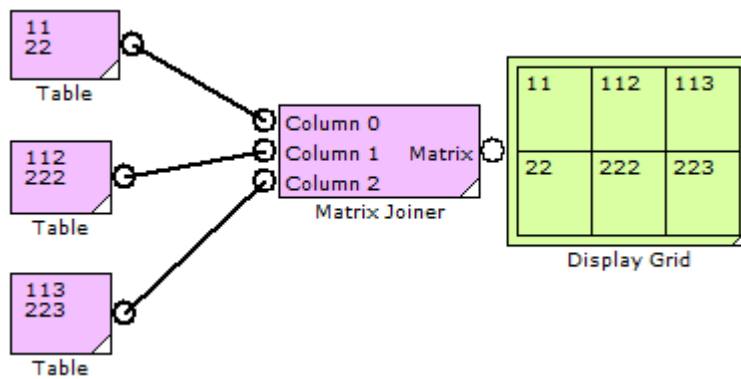
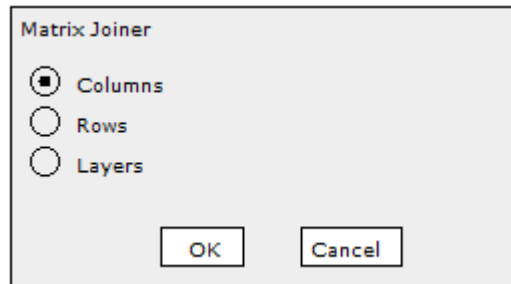


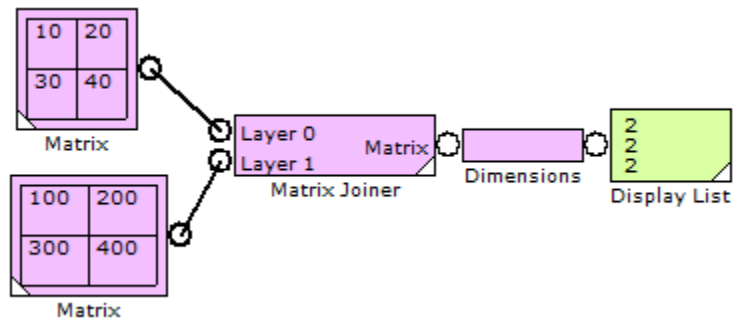
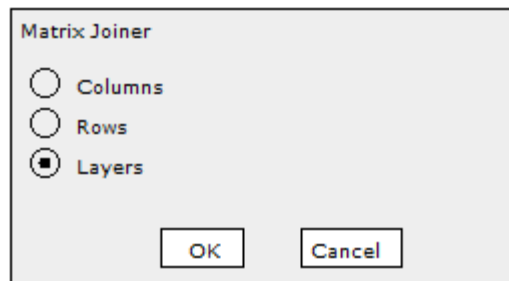
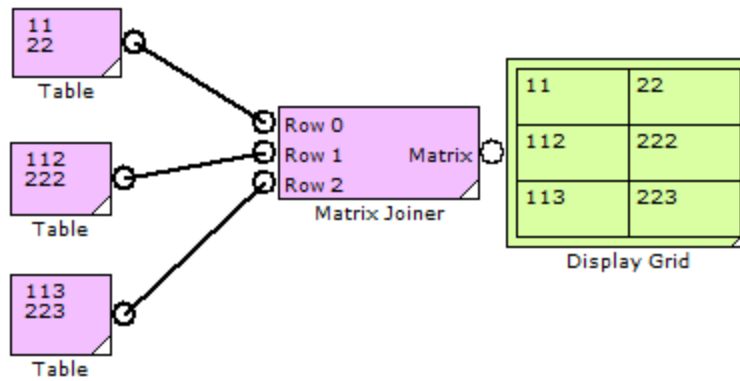
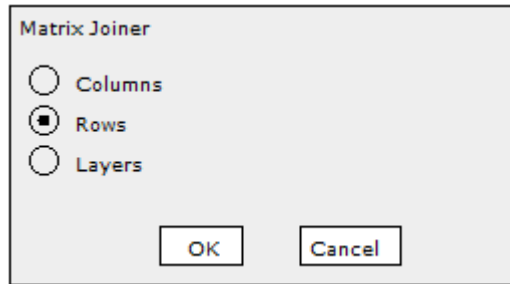


# Matrix Joiner

The Matrix Joiner component joins matrices in the x direction into a single matrix. column 4

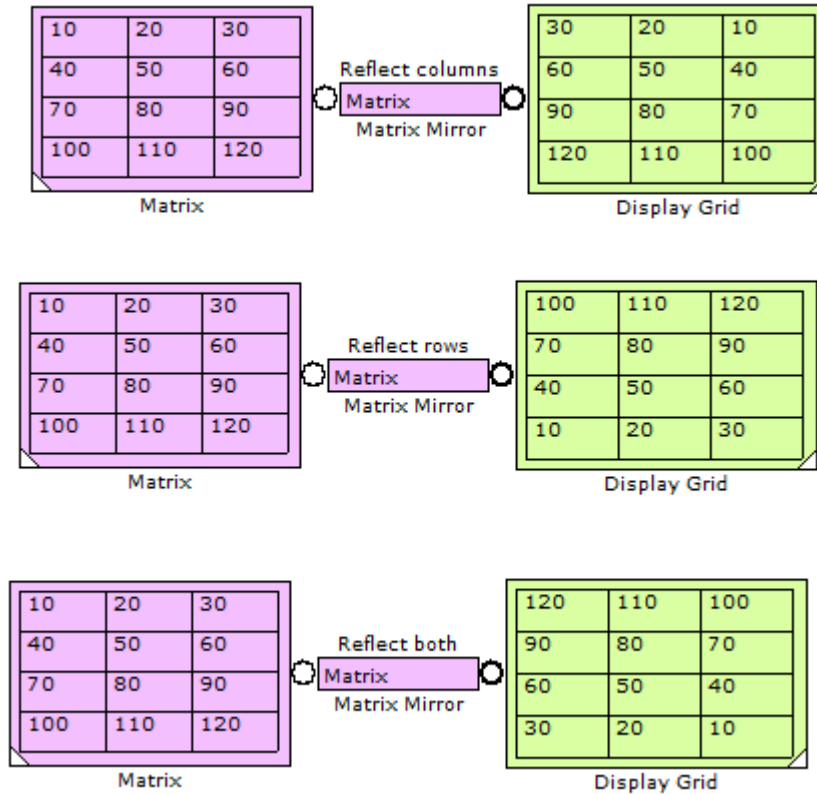
Arrays -





# Matrix Mirror

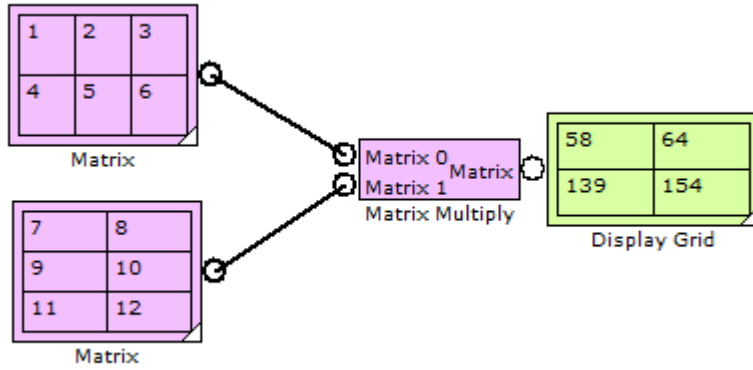
The Matrix Mirror component returns the matrix mirrored in rows, columns, or both. Use the Options menu to decide. Arrays - column 4



# Matrix Multiply

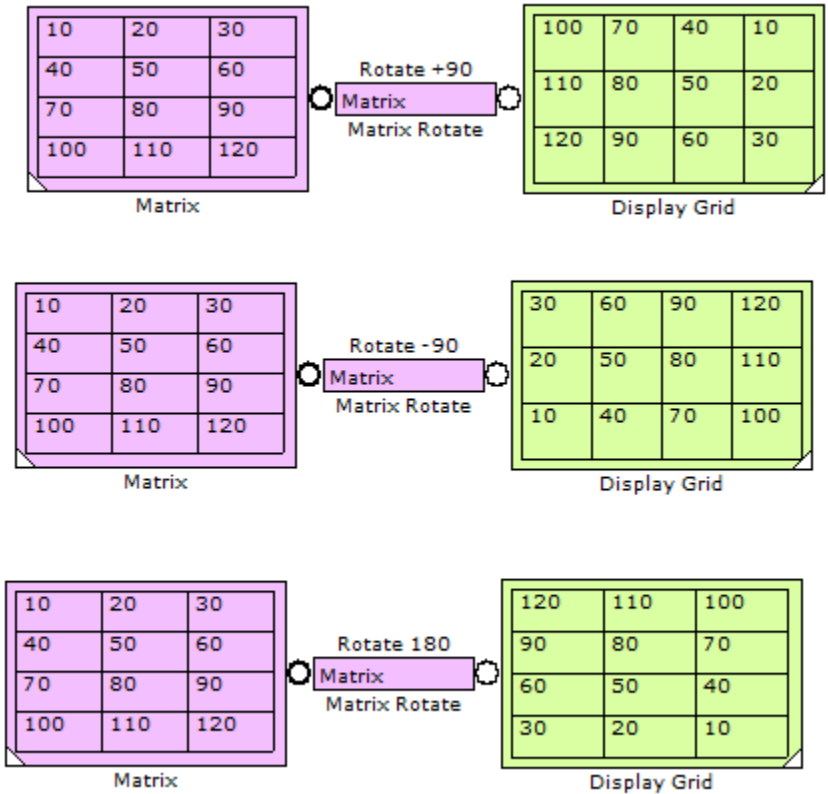
The Matrix Multiply component multiplies two matrices and outputs a matrix and its size.  
- column 4

Arrays



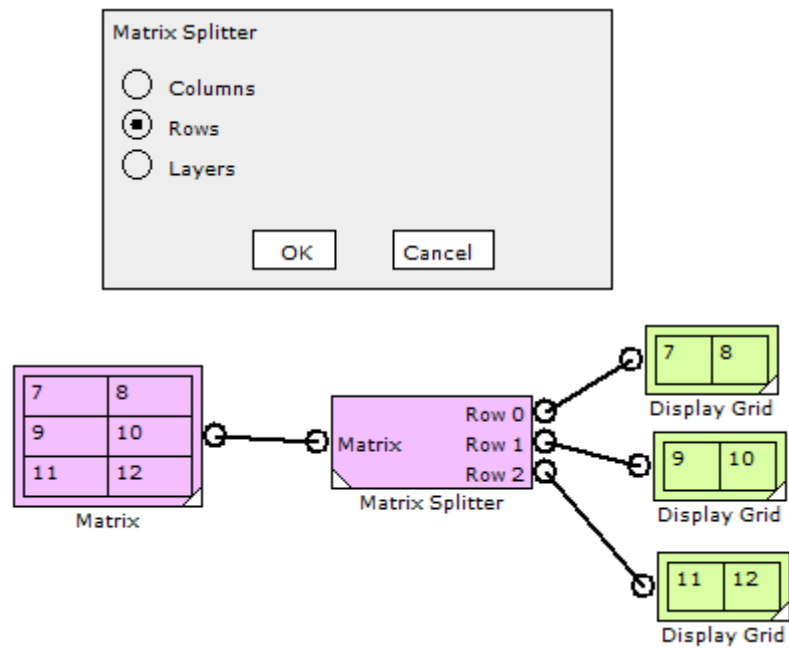
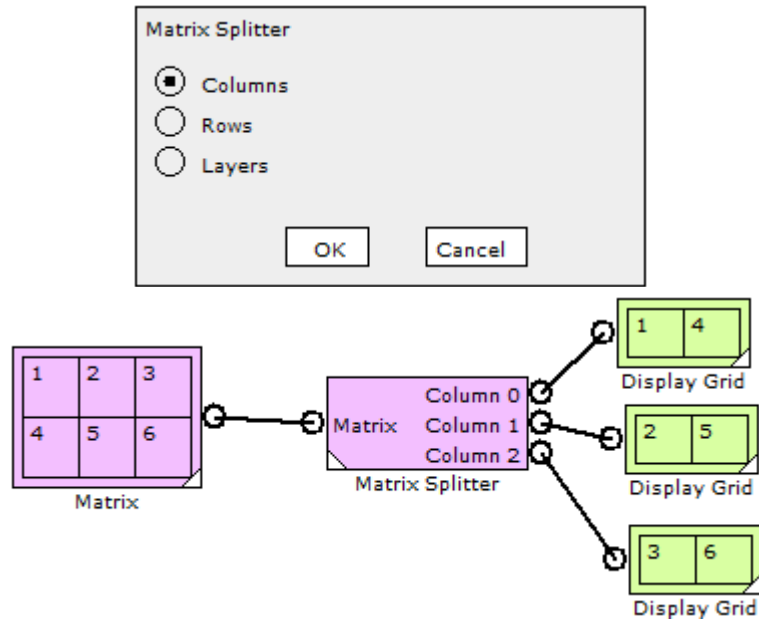
# Matrix Rotate

The Matrix Rotate component returns the matrix rotated by +90, -90, or 180 degrees. Use the Options menu to decide. Arrays - column 4



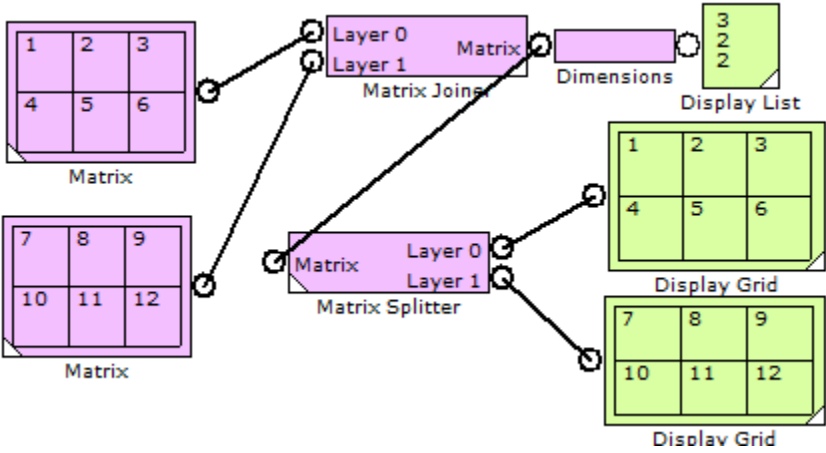
# Matrix Splitter

The Matrix Splitter component breaks the input matrix into separate vector arrays. You can set the number of outputs desired. Arrays - column 4



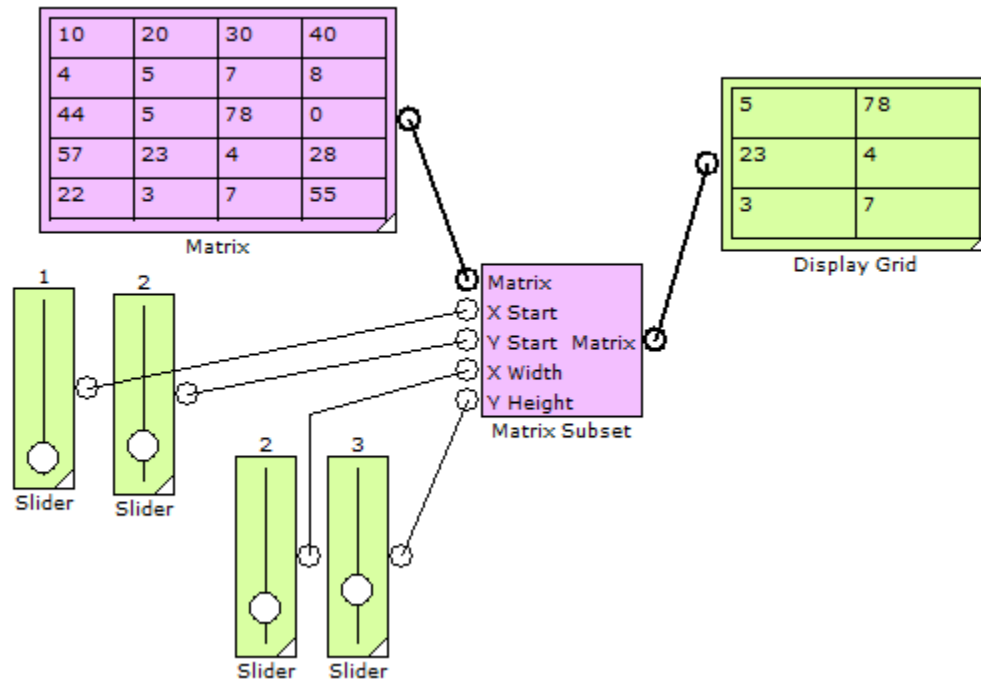
Matrix Splitter

Columns  
 Rows  
 Layers



# Matrix Subset

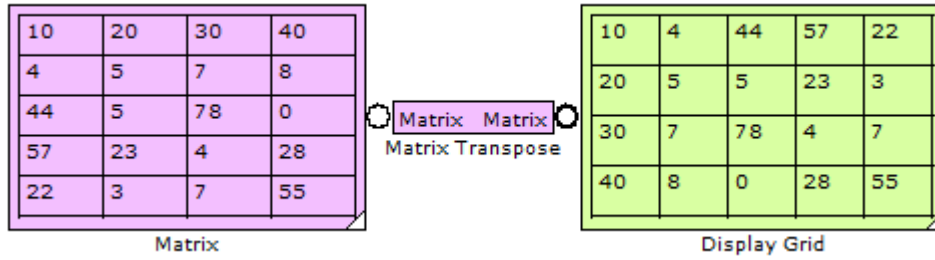
The Matrix Subset component outputs a portion of the values in an array. Input the number of the first value in X and Y, and the size in X and Y you want at the output. Arrays - column 4





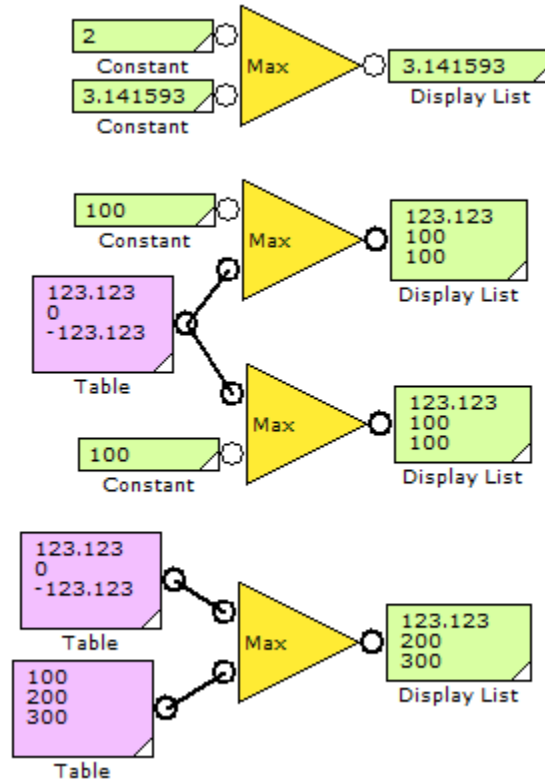
# Matrix Transpose

The Matrix Transpose component returns the transpose of a matrix. Values are shifted along the diagonal. Arrays - column 4



# Max

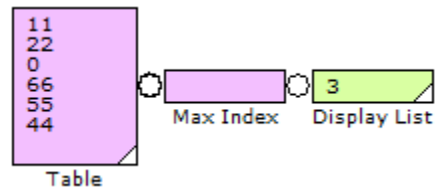
The Max component outputs the larger of the two input values. Inputs can be scalars or arrays.  
Functions - column 1



## Max Index

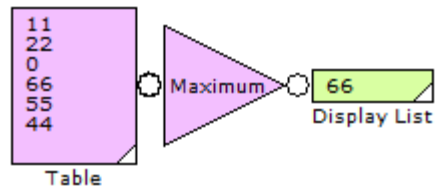
The Max Index component scans the input array and outputs the index of the largest value.  
- column 3

Arrays



# Maximum

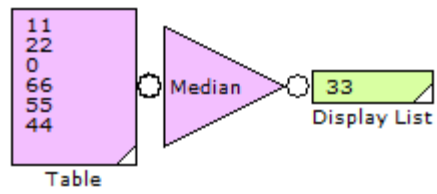
The Maximum component scans the input array and outputs the largest value. Arrays - column 3



# Median

The Median component scans the input array and outputs the middle value.

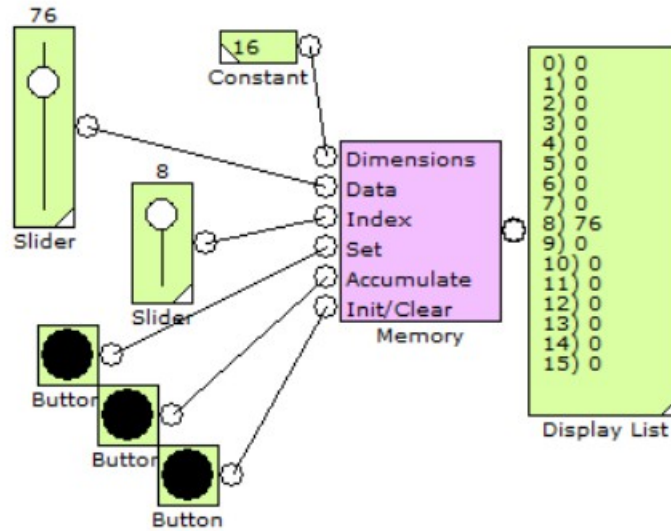
Arrays - column 3



# Memory

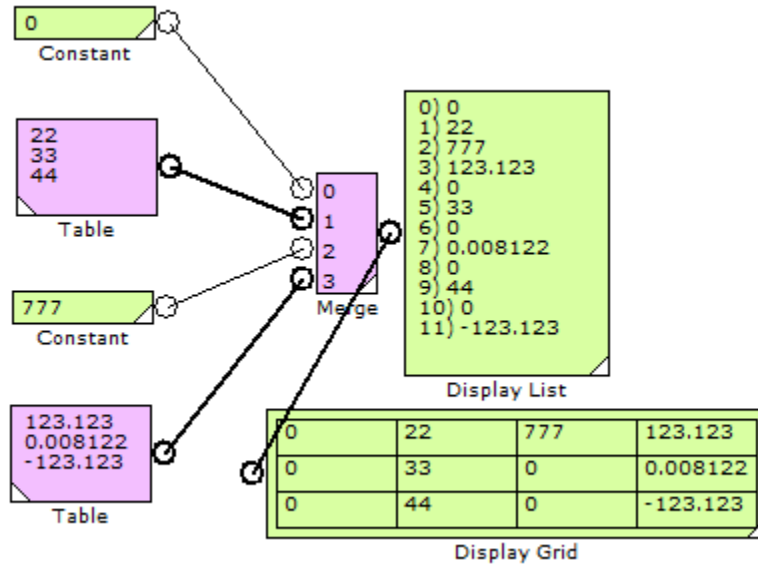
The Memory component creates a table of the input Dimensions filled with zeros when the Init/Clear input goes true. When the Set input goes true the values at the Data input are put at the positions indicated by the Index input. When the Accumulate input goes true the values at the Data input are added to the values at the positions indicated by the Index input.

Arrays - column 1



# Merge

The Merge component joins all the values for all its inputs into a single output. The inputs are interlaced one at a time like stacking a deck of cards. You can set the number of inputs. Arrays - column 1



# Mesh

The Mesh component lets you control a number of constants in two dimensions, horizontal (x) and vertical (y). Controls - column 1

<b>Move To Top</b>	<b>Delete</b>
<b>Move Up</b>	<b>Details...</b>
<b>Move Down</b>	<b>Duplicate</b>
<b>Move To Bottom</b>	<b>Edit Label...</b>
<b>Copy</b>	<b>Options...</b>
<b>Paste</b>	<b>Redistribute</b>

**Mesh**

Set X Points:

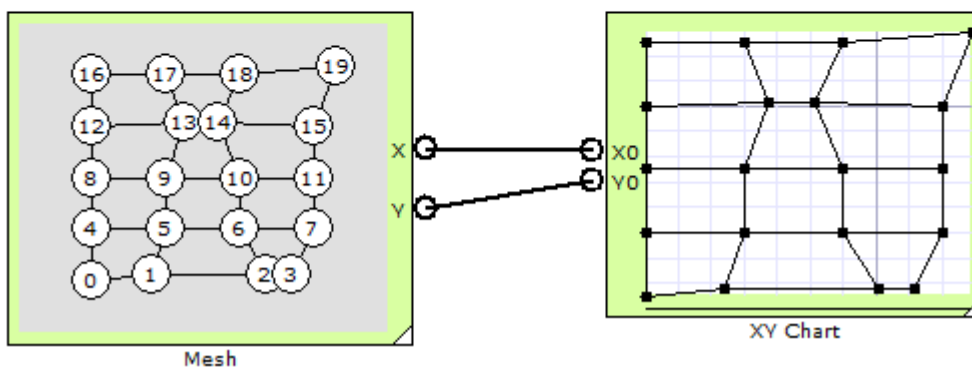
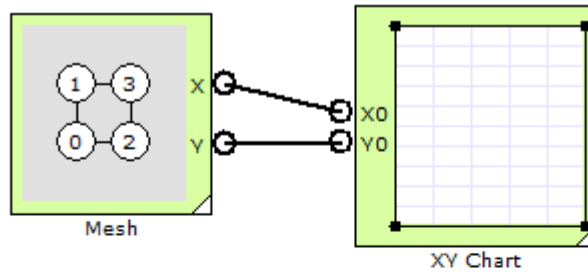
Set Y Points:

Set Minimum X:

Set Maximum X:

Set Minimum Y:

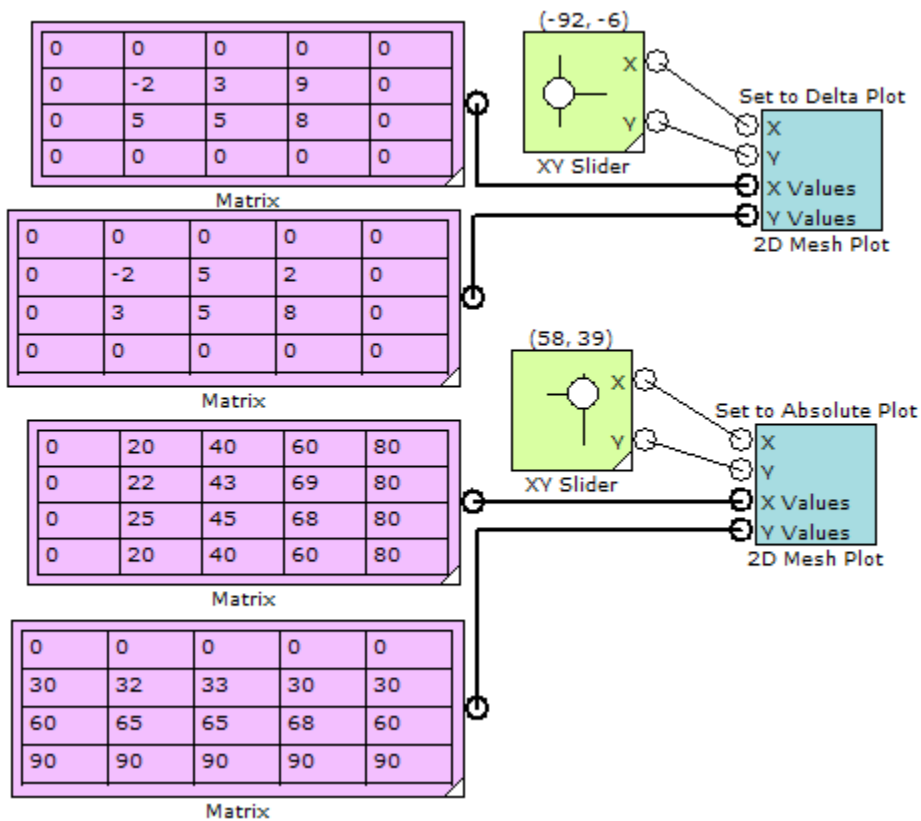
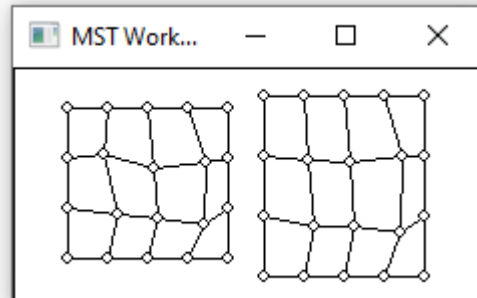
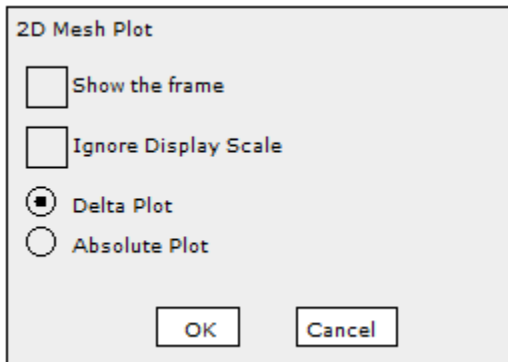
Set Maximum Y:





## 2D Mesh Plot

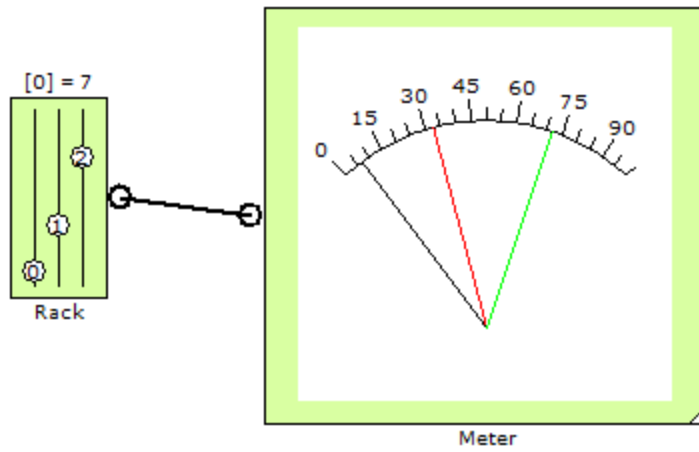
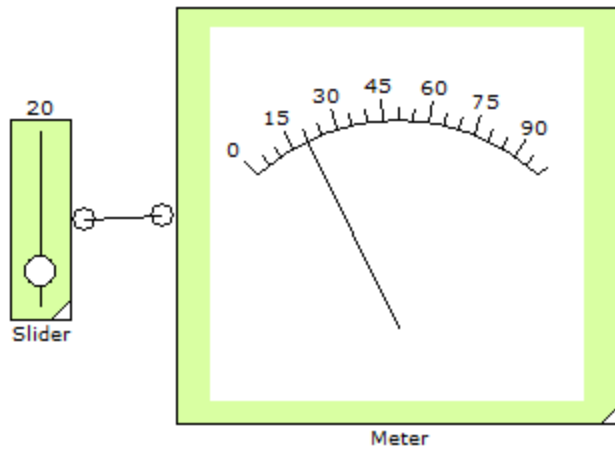
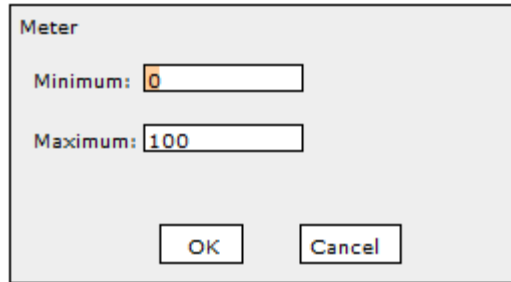
The 2D Mesh Plot plots 2D data creating a mesh plot of the data in the 2D display window. In the mesh plot, a grid with dots is created from the X, Y points. The component has 2 modes: Absolute and Delta. The inputs to the X Values and Y Values must be two 2 dimensional arrays. 2D - column 2



# Meter

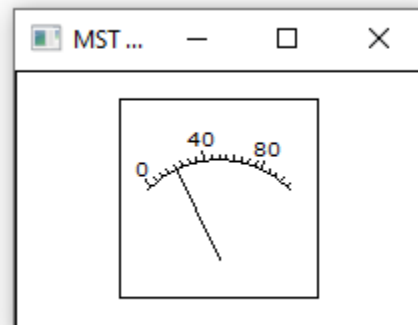
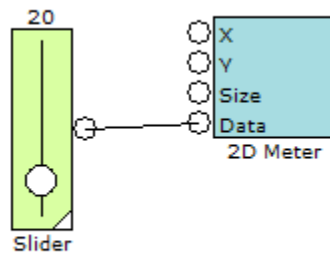
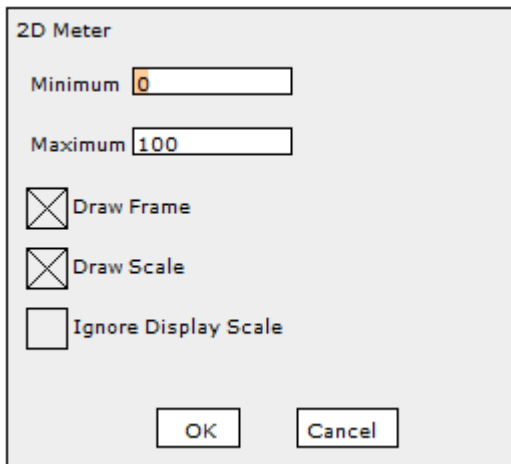
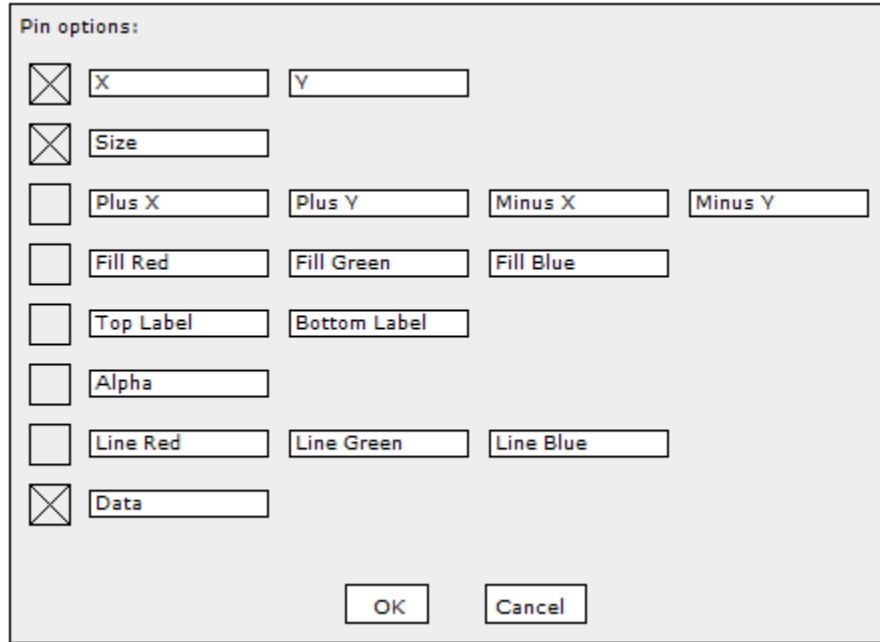
The Meter is used to show the values at any terminal using pointers.

Controls - column 4



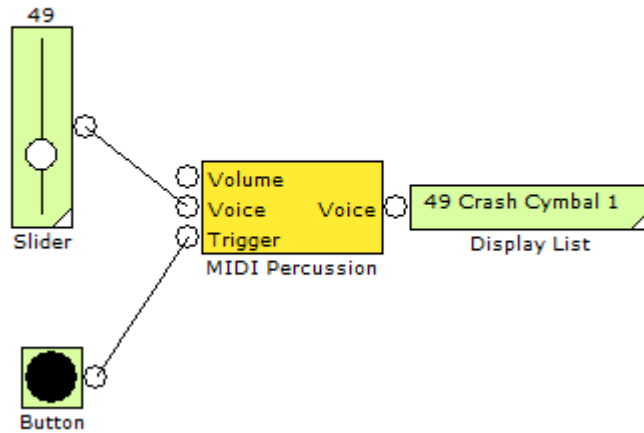
# 2D Meter

The 2D Meter is used to show the values at any terminal using pointers. It draws in the 2D display window. 2D - column 2



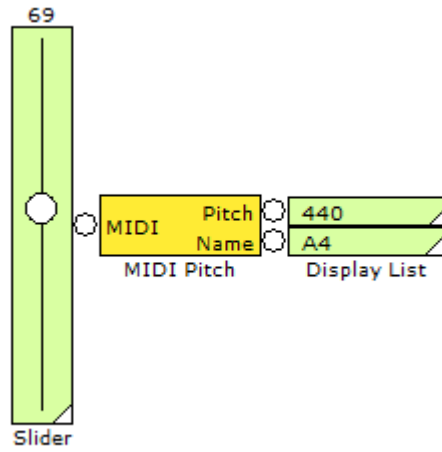
# MIDI Percussion

The MIDI Percussion component uses the internal MIDI synthesizer to play a percussive sound. Duration is milliseconds, Volume is 0 to 127. Voice is the instrument to play. Functions - column 6



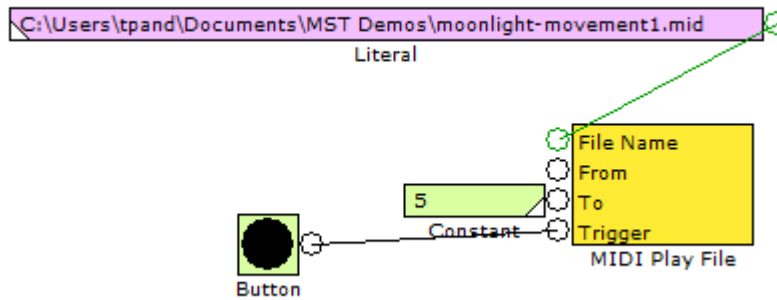
# MIDI Pitch

The MIDI Pitch component given the MIDI number of a note, this component outputs the frequency and name of the note. Functions - column 6



## MIDI Play File

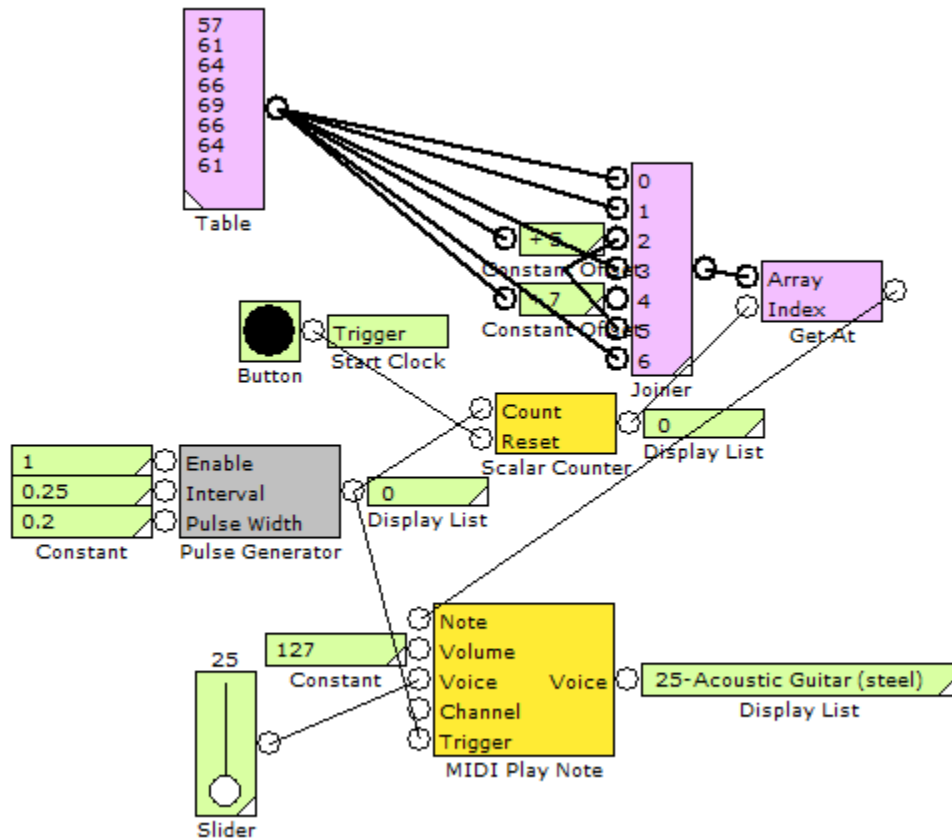
The MIDI Play File component plays the input file with the MIDI synthesizer. Functions – column 6



# MIDI Play Note

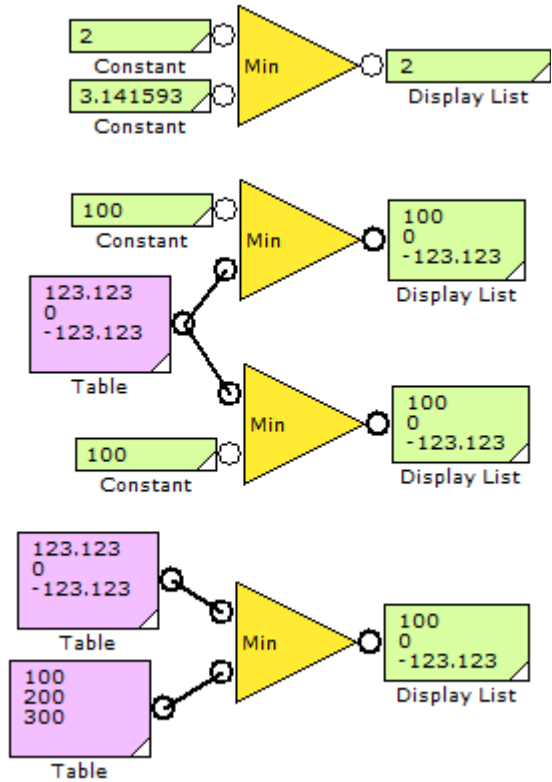
The MIDI Play Note component uses the internal MIDI synthesizer to play a sound. Note is the MIDI note (60 is middle C), Duration is milliseconds, Volume is 0 to 127. Voice is the instrument to play.

Functions - column 6



# Min

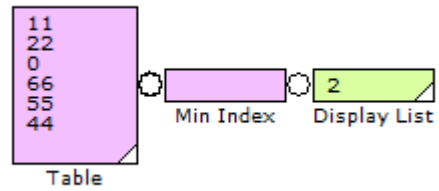
The Min component outputs the smaller of the two input values. Inputs can be scalars and /or arrays.  
Functions - column 1





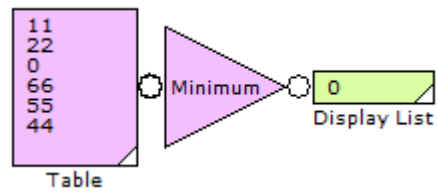
## Min Index

The Min Index component scans the input array and outputs the index of the smallest value.     Arrays  
- column 3



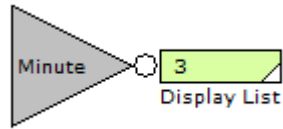
# Minimum

The Minimum component scans the input array and outputs the smallest value. Arrays - column 3



# Minute

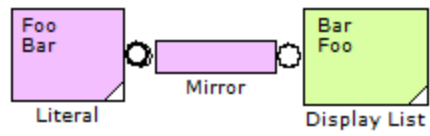
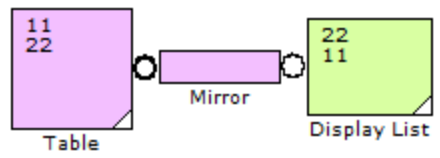
The Minute component outputs the clock minutes from the computers internal clock. Functions - column 7



# Mirror

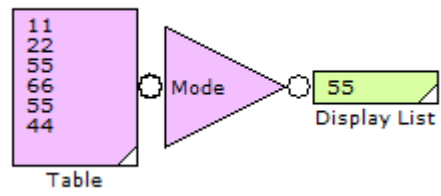
The Mirror component reverses the order of the input array.

Arrays - column 2



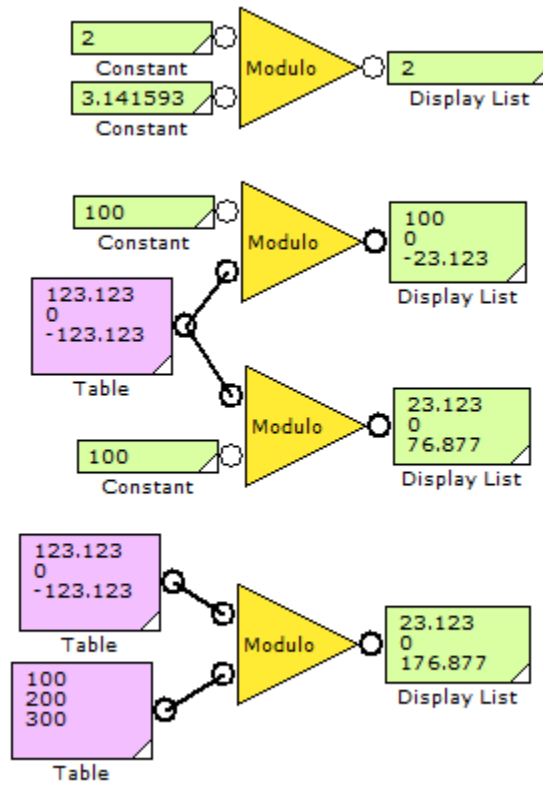
# Mode

The Mode component scans the input array and outputs the most common value. Arrays - column 3



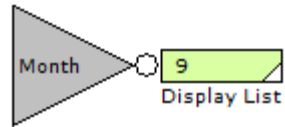
# Modulo

The Modulo component divides the inputs and outputs the remainder. Inputs can be scalars or arrays.  
Functions - column 1



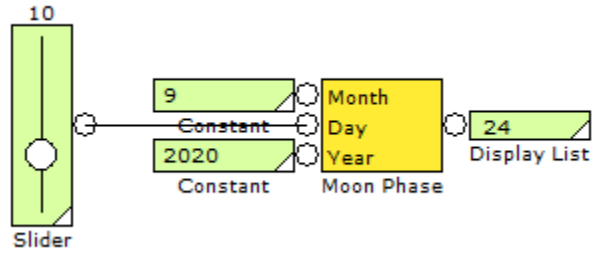
# Month

The Month component outputs the current calendar month. Functions - column 7



# Moon Phase

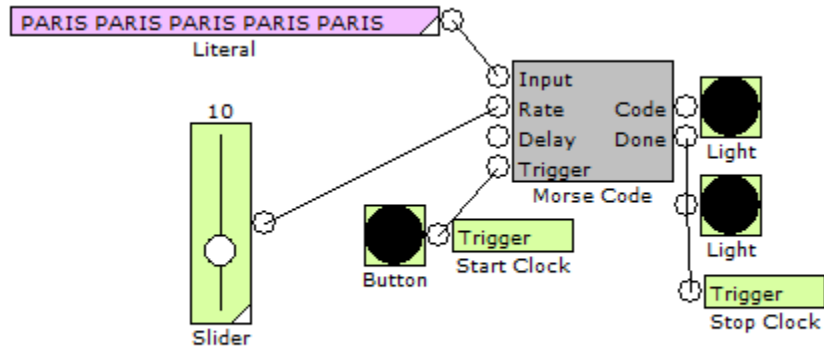
The Moon Phase component calculates the Phase of the Moon. Output is days since the New Moon (0 to 29, where 0=new moon, 15=full) for the selected date. Functions - column 9





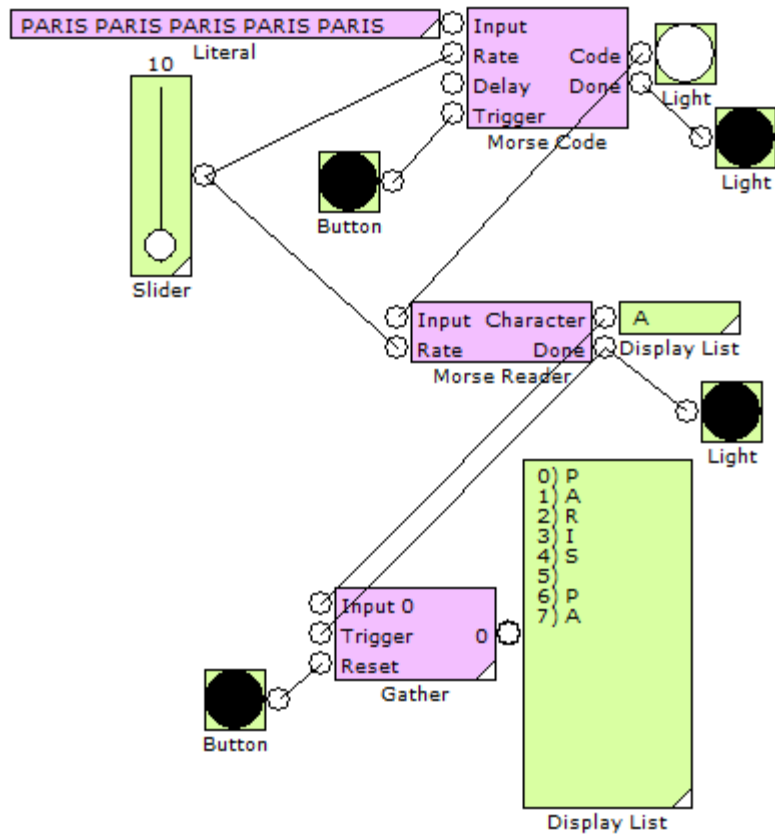
# Morse Code

The Morse Code component generates pulses based on the inputs. These pulses can be connected to a light or sound input. You can use the delay to give you time to respond to the Morse code while learning the character sounds at a higher speed. Arrays - column 7



# Morse Reader

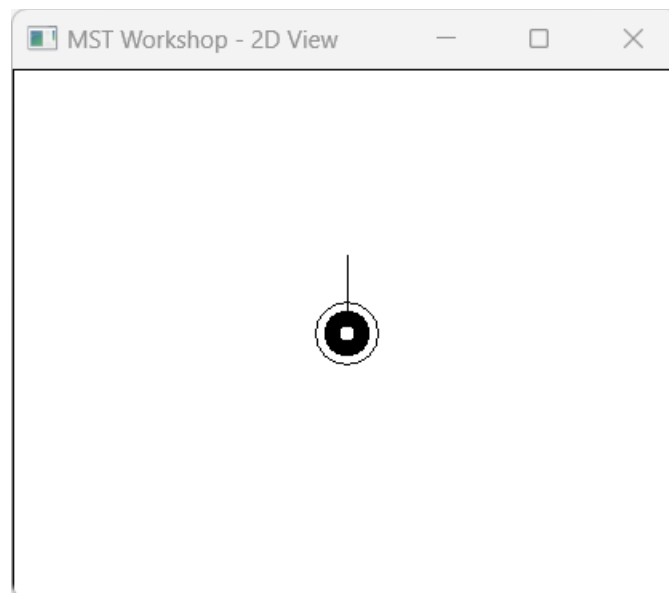
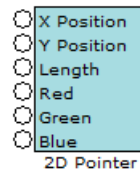
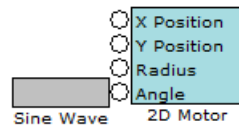
The Morse Reader component receives pulses and converts them to the equivalent Morse Code. Arrays  
- column 7



## 2D Motor

The 2D Motor converts its input directly into shaft rotation. Place a Gear at the same X, Y position to connect the motor to the gear.

2D - column 3



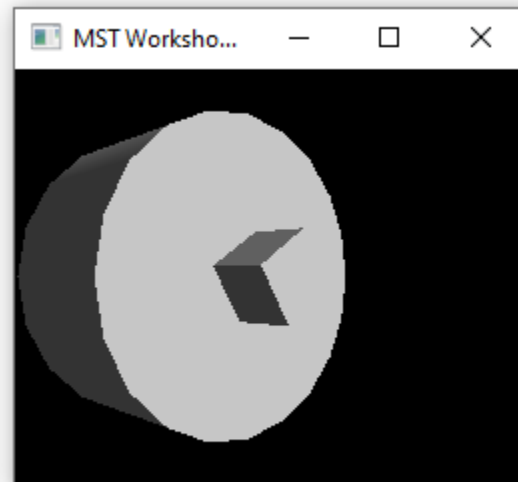
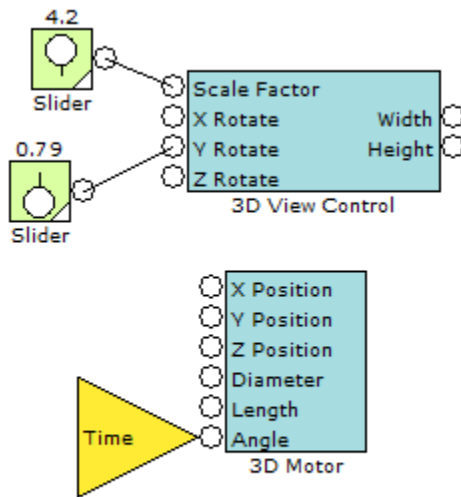
# 3D Motor

The 3D Motor creates a motor that can drive other 3D simulation objects like 3D Gears. Place a Gear at the same X, Y, Z position and orientation to connect the gear to a motor. You specify the angle of the motor's shaft. 3D - column 4

Pin options:

<input checked="" type="checkbox"/>	X Position	Y Position	Z Position
<input type="checkbox"/>	X Rotate	Y Rotate	Z Rotate
<input checked="" type="checkbox"/>	Diameter	Length	
<input checked="" type="checkbox"/>	Angle		
<input type="checkbox"/>	Red	Green	Blue
<input type="checkbox"/>	Alpha		
<input type="checkbox"/>	[4x4]		

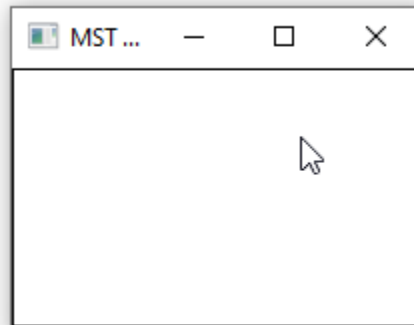
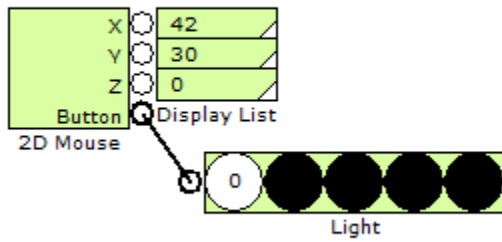
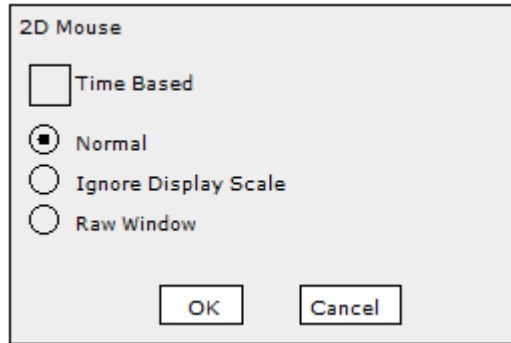
OK Cancel



## 2D Mouse

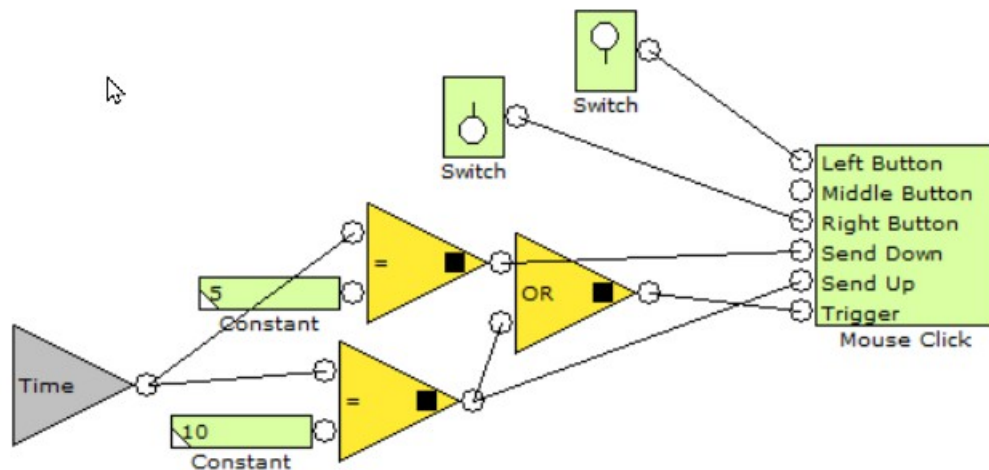
The 2D Mouse gives you the X and Y position of the cursor in the 2D display window. There are Options for how the X and Y values are measured. The Z output is from the wheel on the mouse. The Button output gives 5 mouse buttons in this order: Left, Middle, Right, Extra Left, Extra Right.

Controls - column 1



## Mouse Click

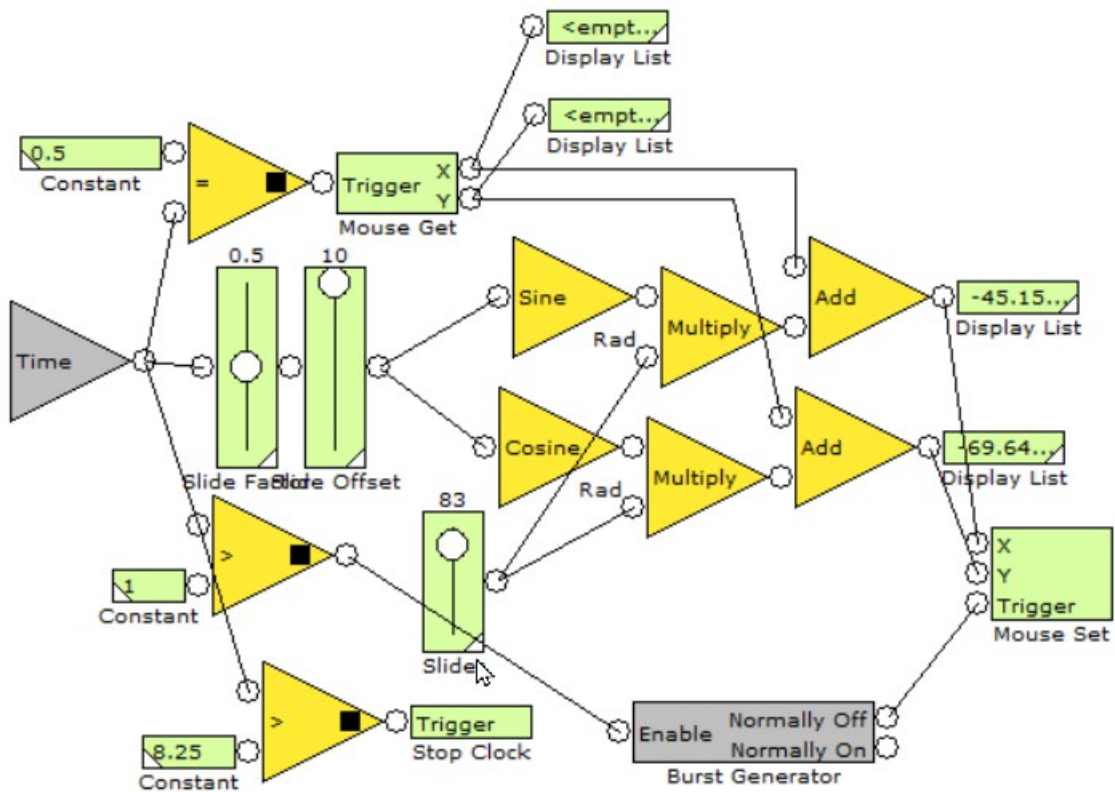
The Mouse Click component simulates a mouse click as if someone is clicking the computer's mouse buttons. Connect True or False for the Left and Right inputs to select which button and True or False for the desired Up and Down action. Controls – column 4



Reset and start this solution and it will click the left button down at 5 seconds and release it at 10 seconds. You can see the effect by opening a word processor and dragging the mouse through the text as this solution runs. A section of the text will be selected.

# Mouse Get

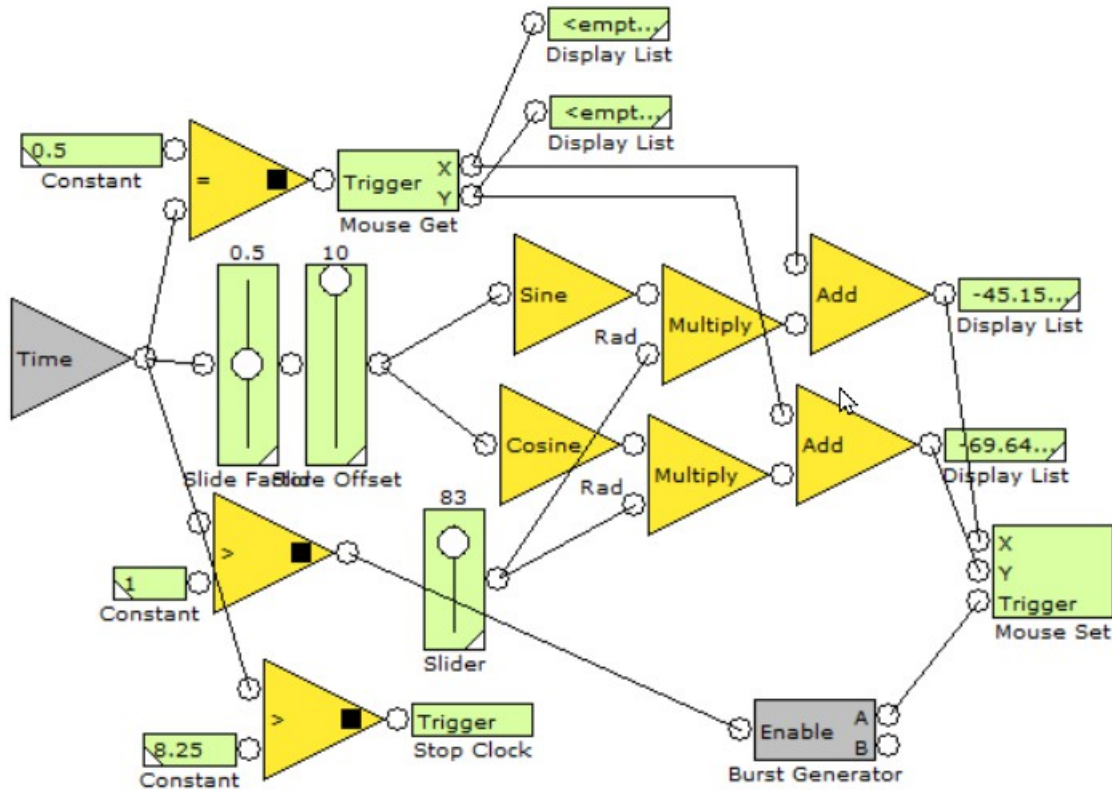
The Mouse Get component gets the x and y coordinates of the computer's mouse. Controls – column 4



This solution gets the current mouse position and drives it in a semicircle around that position when the clock is started.

# Mouse Set

The Mouse Set component sets the x and y coordinates of the mouse as if someone moved the computer's mouse. Controls – column 4

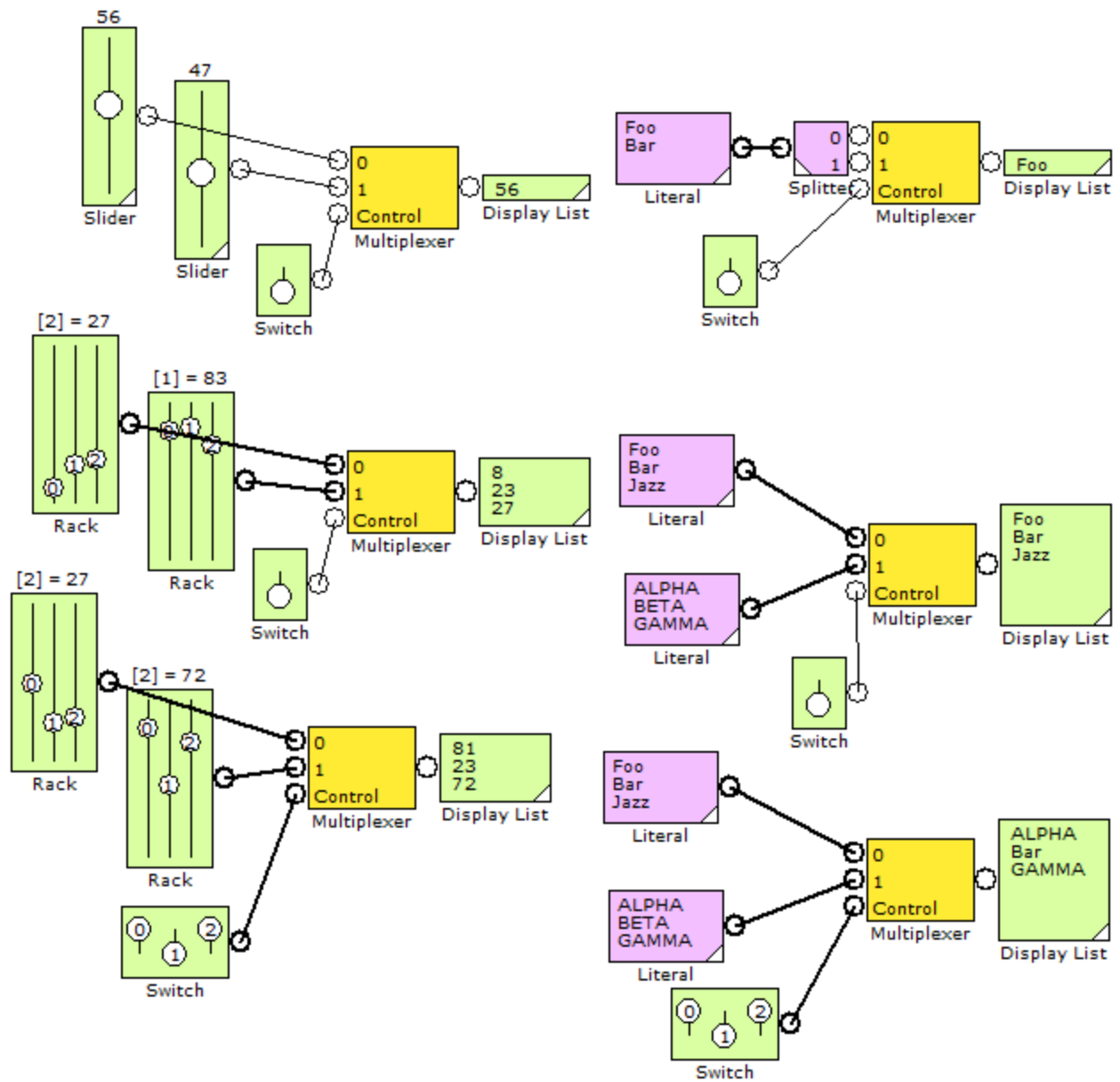


This solution gets the current mouse position and drives it in a semicircle around that position when the clock is started.



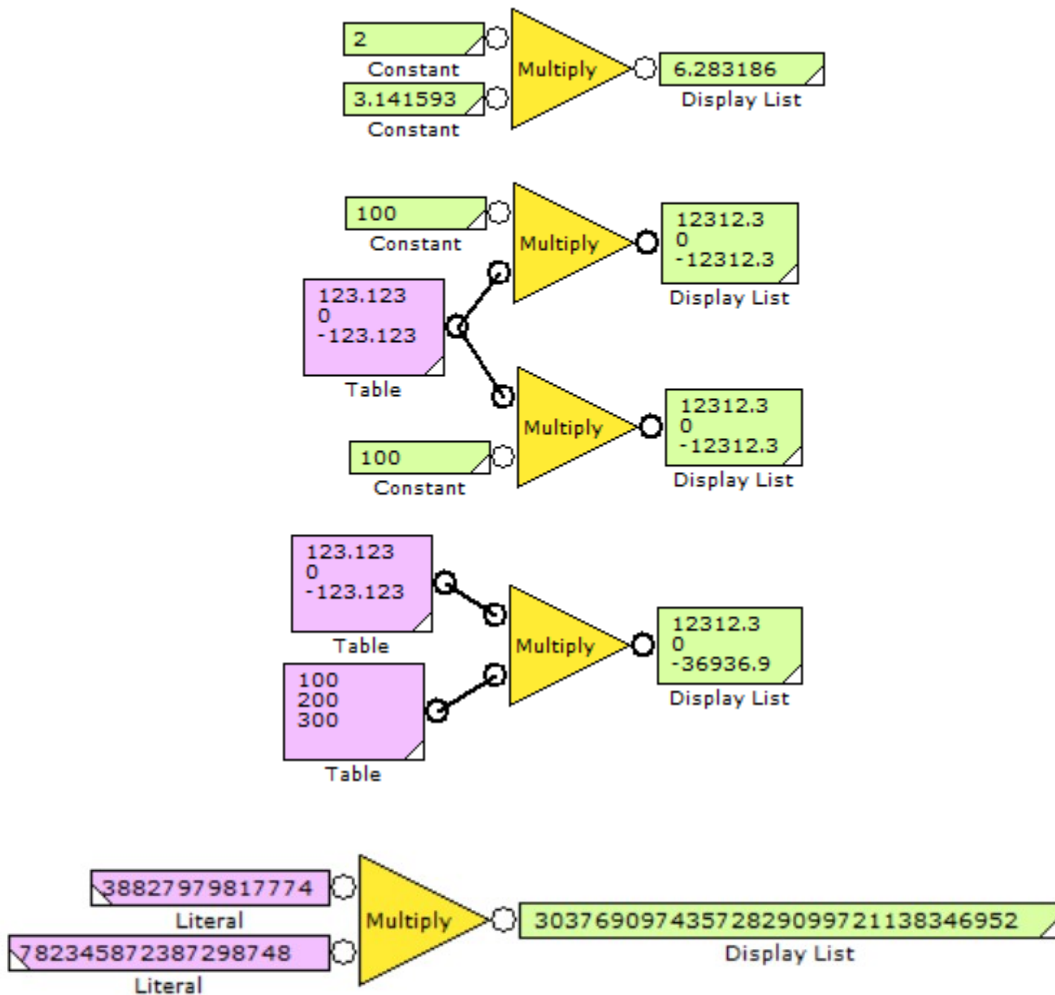
# Multiplexer

The Multiplexer can switch inputs based on the Control input. Functions - column 6



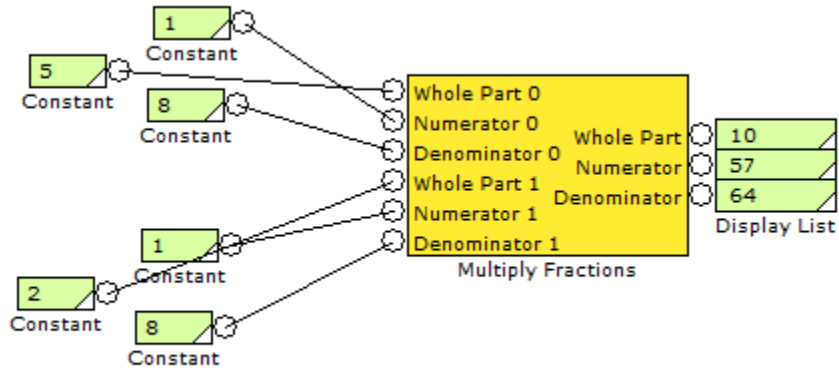
# Multiply

The Multiply component multiplies the inputs together. Inputs can be scalar and /or arrays.  
Functions - column 1



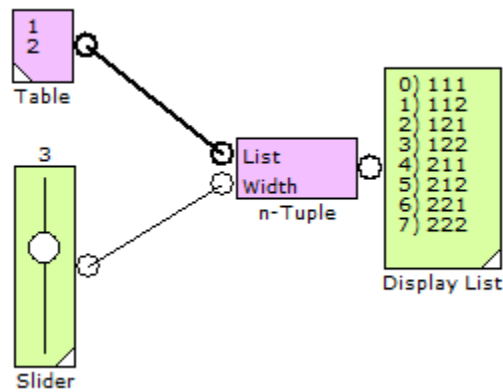
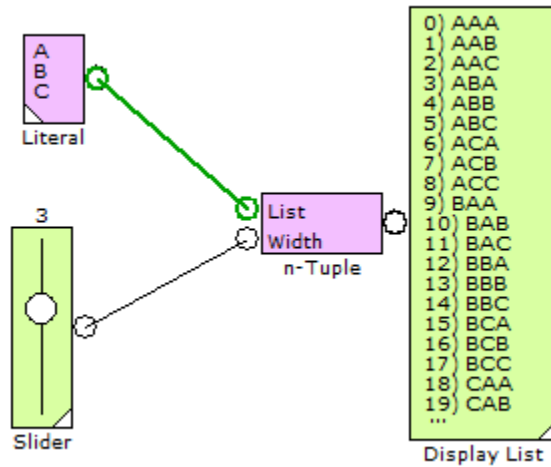
# Multiply Fractions

The Multiply Fractions component multiplies two numbers with fractions. The output is a whole number with a fraction as needed. Functions - column 9



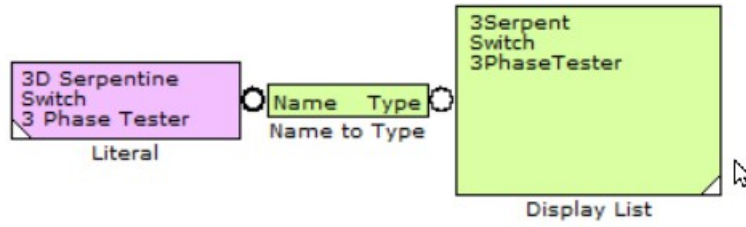
# n-Tuple

The n-Tuple component outputs a list of values using the set of inputs with the specified width.  
Arrays – column 5.



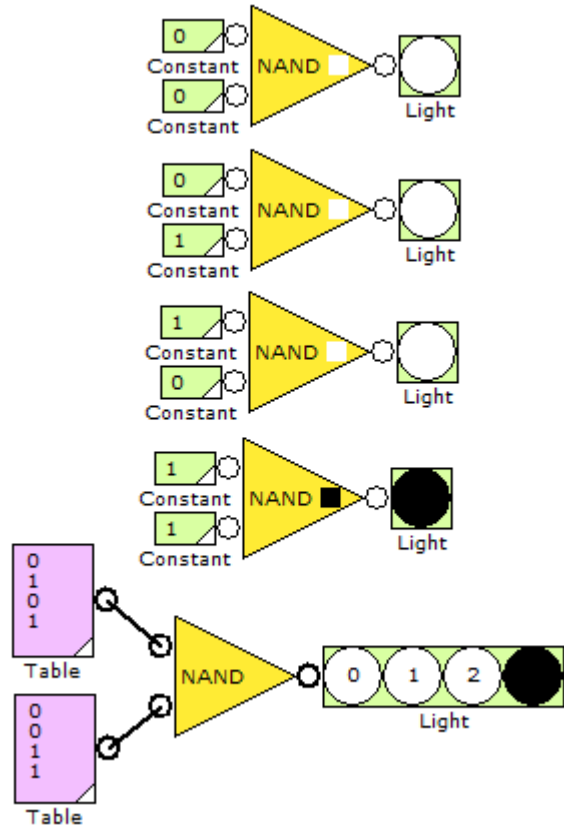
## Name to Type

The Name to Type Component returns the internal type for each component name input.  
Controls - column 5



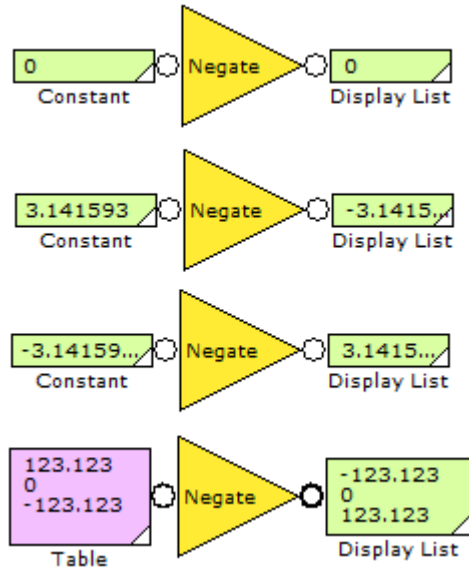
# NAND

The NAND component compares the inputs and returns false if both values are true. It's output is the opposite of the AND component. Inputs can be scalars or arrays. Functions - column 3



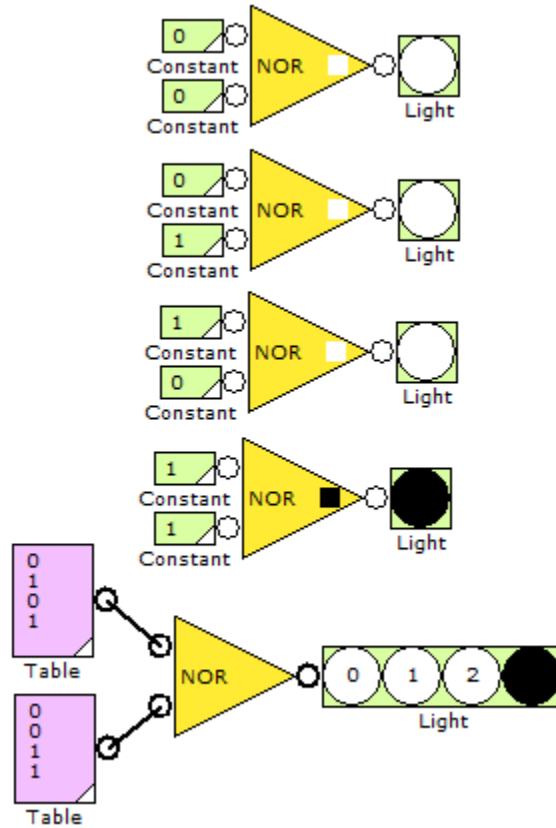
# Negate

The Negate component changes the sign of the input. Input can be a scalar or an array. Functions - column 1



# NOR

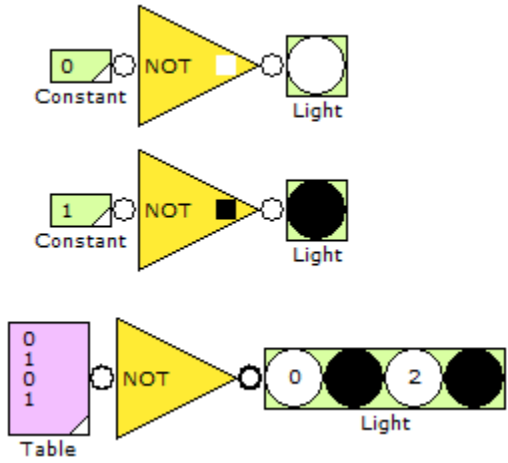
The NOR component compares the inputs and returns false if either or both values are true. Its output is the opposite of the OR component. Inputs can be scalars or arrays. Functions - column 3





# NOT

The NOT component compares the inputs and returns true if the input is false, returns false if the input is true. Inputs can be scalars or arrays. Functions - column 3



## 2D Number Spinner

The 2D Number Spinner is used to enter a number on the 2D canvas.

The X, Y and Size inputs control the position and font size of the number.

The Anchor inputs offset the number to left, center or right justify.

The Chevron Position inputs control the horizontal position and size.

The Frame input frames the number.

The Preset inputs allow setting the value externally.

The Decimal input sets the number of decimal places displayed.

The Limit inputs allow setting minimum and maximum values allowed.

The Color inputs color the number.

The Alpha input fades the number.

The Text input takes a list of text options to display.

The Horizontal outputs give the left and right frame coordinates.

The Vertical outputs give the top and bottom frame coordinates.

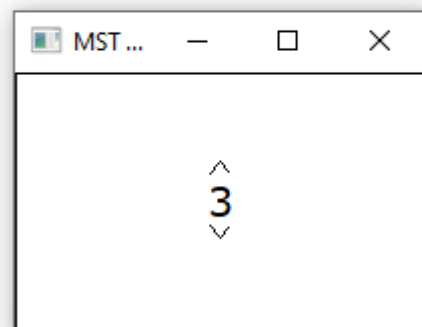
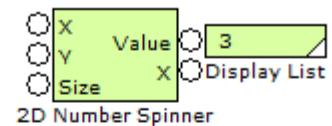
The Sign output is one when the value is negative AND suppresses the sign on the canvas.

Controls - column 2

Pin options:

<input checked="" type="checkbox"/>	X	Y	Size
<input type="checkbox"/>	Horiz Anchor	Vert Anchor	
<input type="checkbox"/>	Chev Position	Chev Size	Chev Count
<input type="checkbox"/>	Frame		
<input type="checkbox"/>	Preset	Set	
<input type="checkbox"/>	Decimal		
<input type="checkbox"/>	Minimum	Maximum	
<input type="checkbox"/>	Red	Green	Blue
<input type="checkbox"/>	Alpha		
<input type="checkbox"/>	Strip Sign	Trim Zeros	

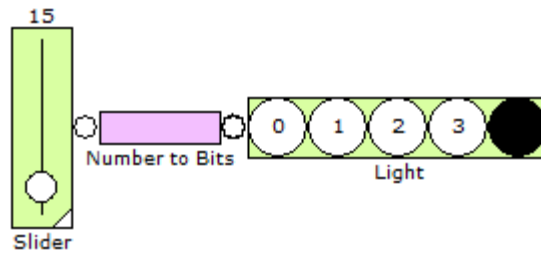
OK Cancel



# Number to Bits

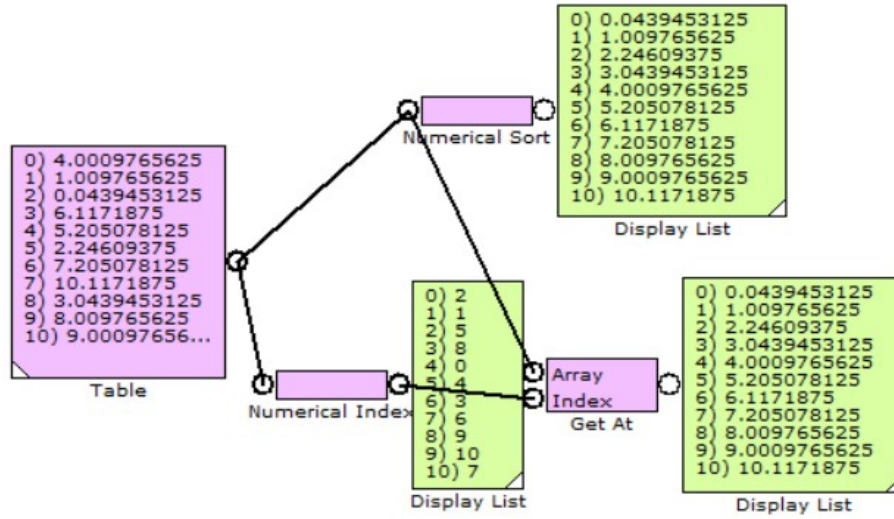
The Number to Bits component breaks the input into an array of bit values.

Arrays - column 4



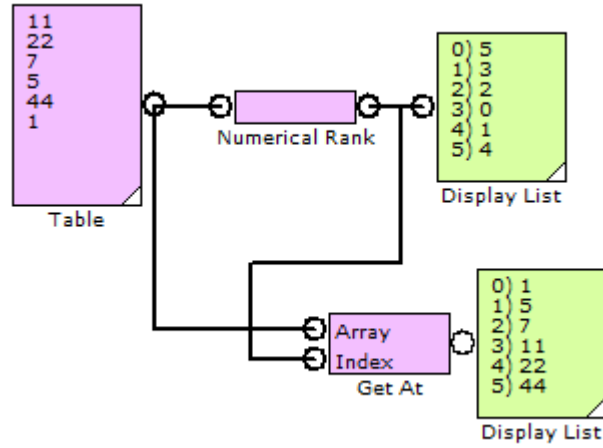
# Numerical Index

The Numerical Index component outputs a list of numbers in the order of the sorted numerical input.  
Arrays – column 4



# Numerical Rank

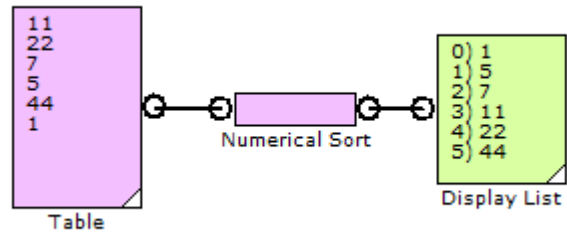
The Numerical Rank component outputs a list of numbers in the order of the sorted numerical input.  
Arrays - column 4



# Numerical Sort

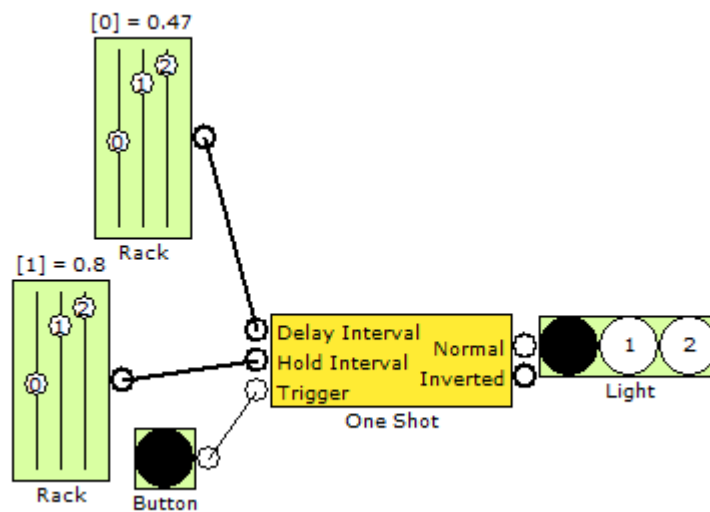
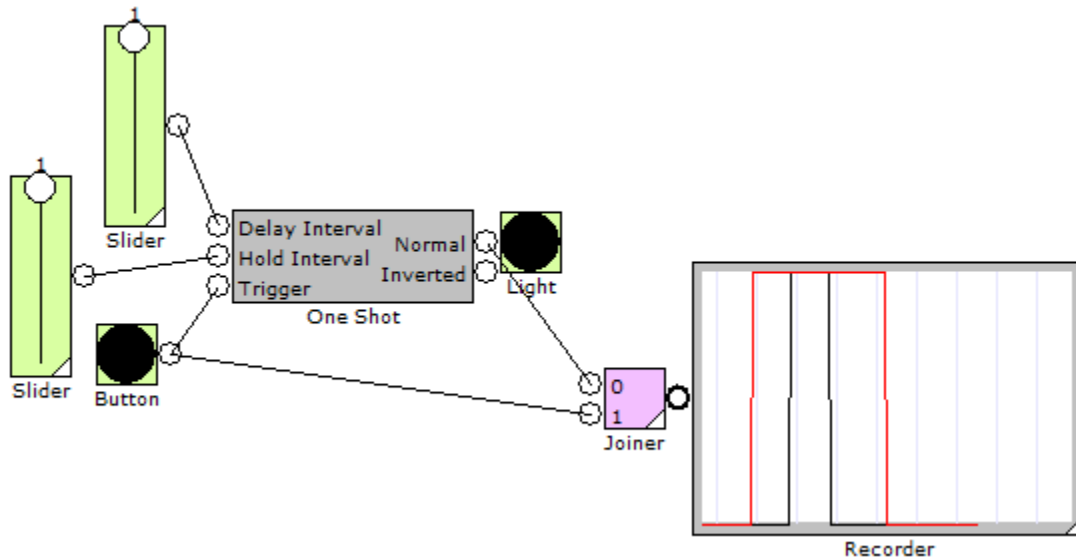
The Numerical Sort component sorts the input values into numerical order.

Arrays - column 4



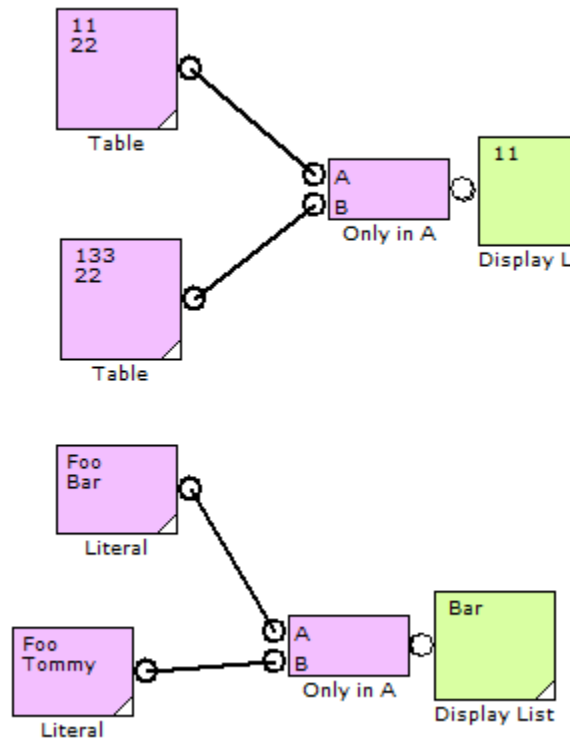
# One Shot

The One Shot component starts when the Trigger input is triggered, then delays for the Delay interval, then goes high for the Hold interval, then resets. Inputs are milliseconds. Functions - column 7



# Only in A

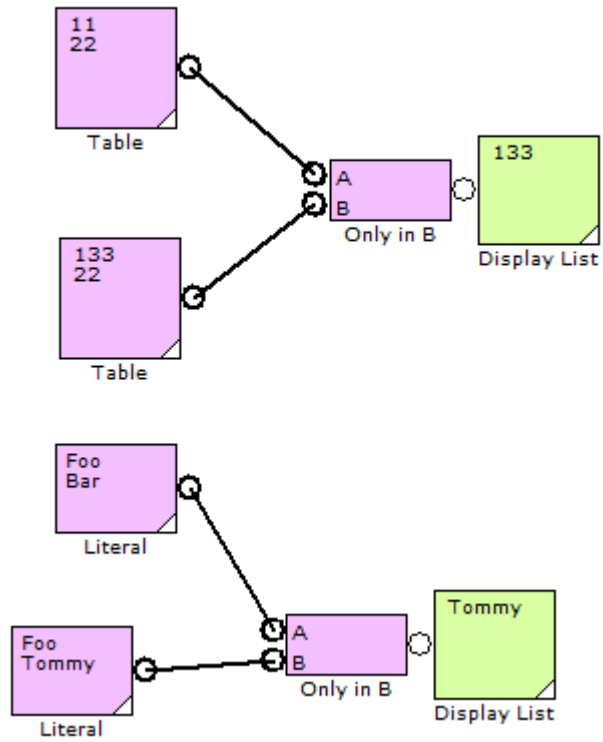
The Only in A component scans the input arrays and outputs the elements that are only found in the top input array. Arrays - column 5





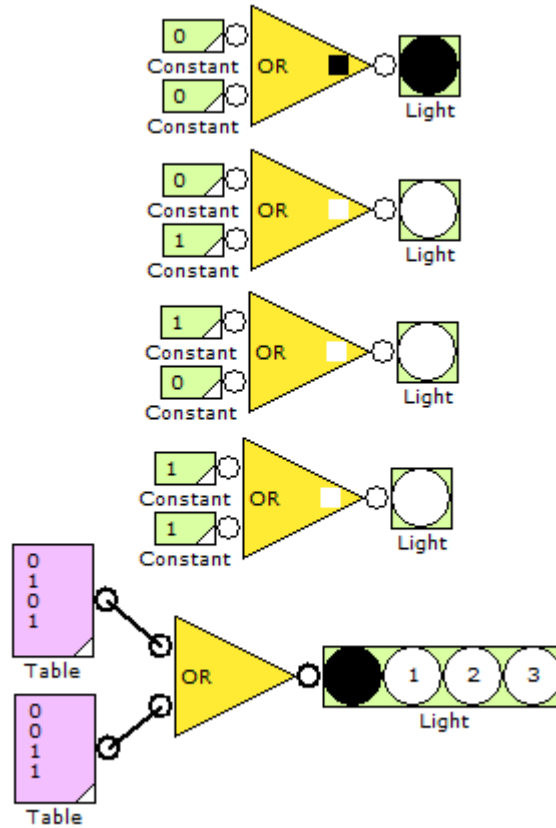
## Only in B

The Only in B component scans the input array and outputs the elements that are only found in the bottom input array. Arrays - column 5



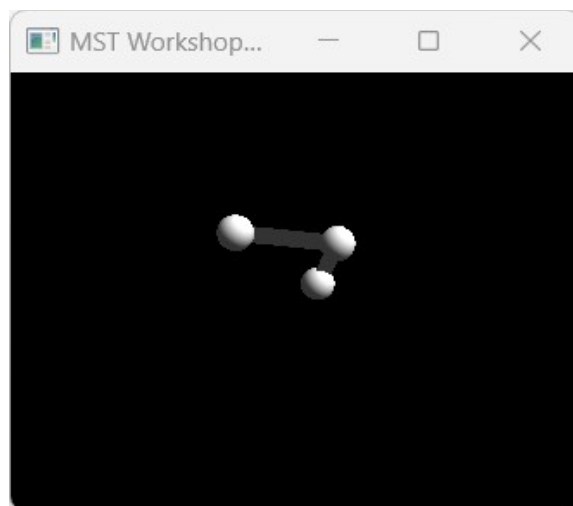
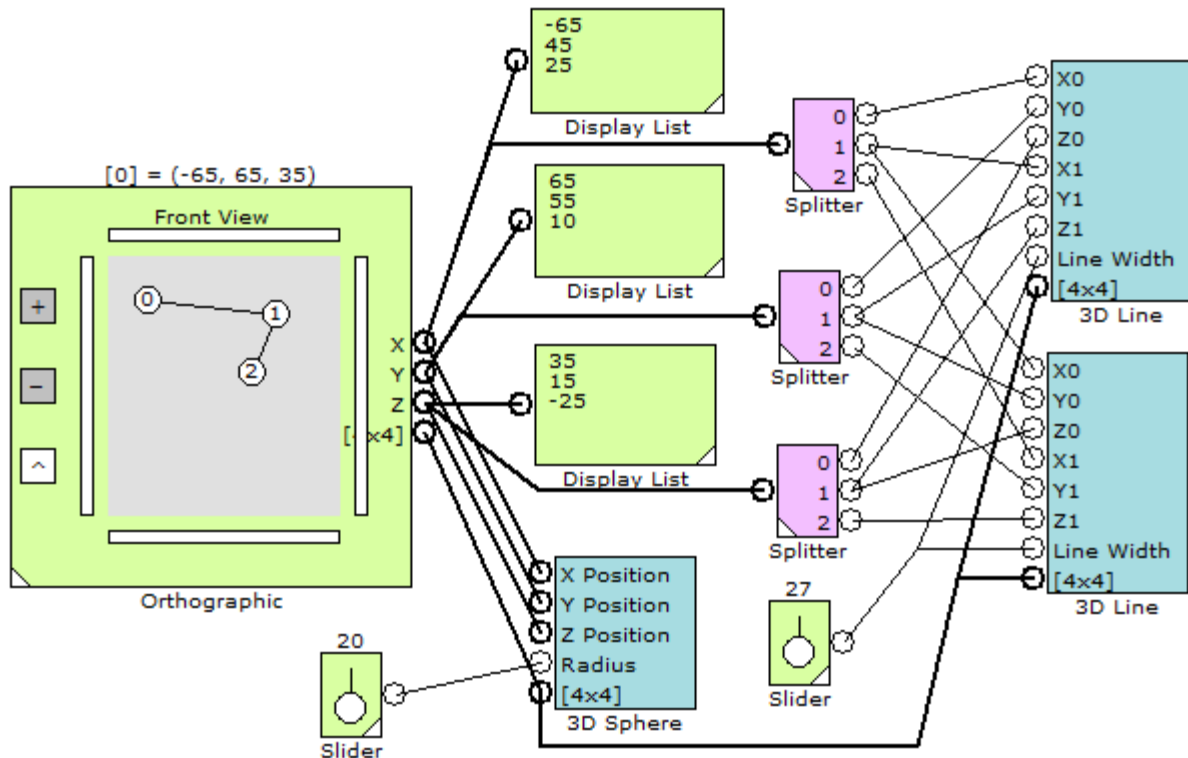
# OR

The OR component compares the inputs and returns true if either or both values are true. Inputs can be scalars or arrays. Functions - column 3



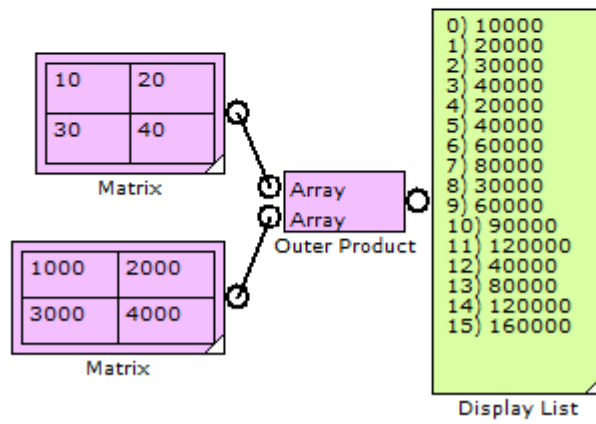
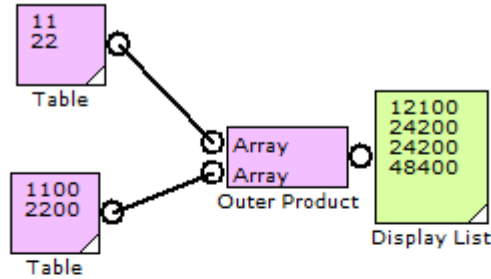
# Orthographic

The Orthographic lets you generate a function in 3 dimensions. You can move the points in X and Y at first but then shift from the front view to top, bottom, left, right or back dragging the vertical controls on the edges of the component. Depending on the view, moving the points left/right or up/down changes the X, Y and Z coordinates of the view. If you have a wheel on your mouse, you can use it to move in the in/out direction to allow control of all 3 coordinates in each view. Note: This component can be operated using the wheel on the mouse. Controls - column 2



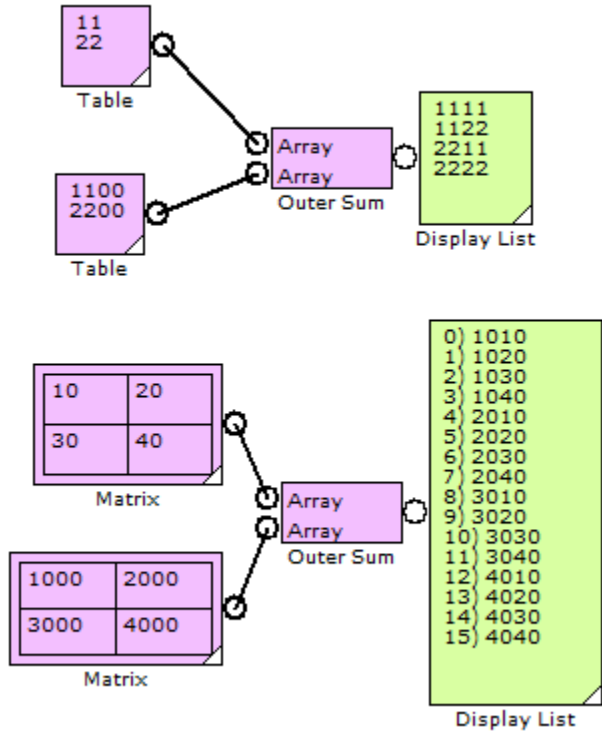
# Outer Product

The Outer Product component combines the input arrays multiplying the values from each input into a larger array. Arrays - column 3



# Outer Sum

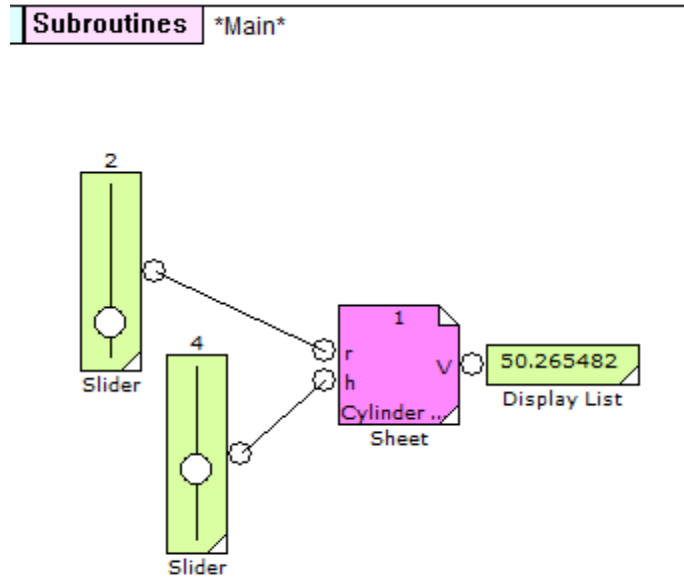
The Outer Sum component combines the input arrays summing the values from each input into a larger array. Arrays - column 3



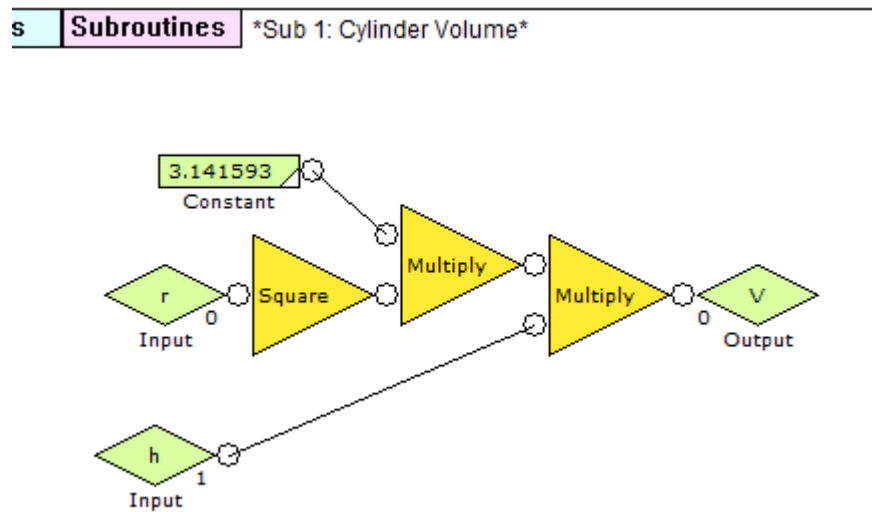
# Output

The Output component is used in a subroutine to carry output values out of the subroutine. Each Output shows as an output pin on the subroutine icon. Controls - column 5

In the main workspace:

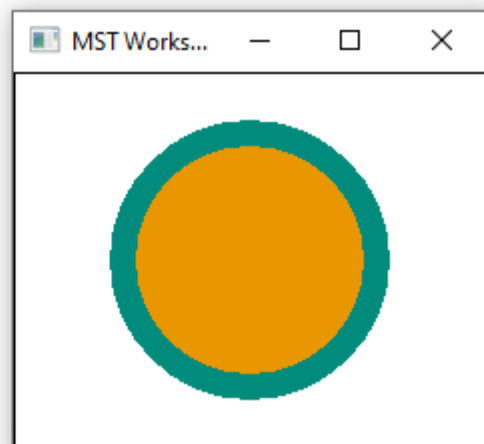
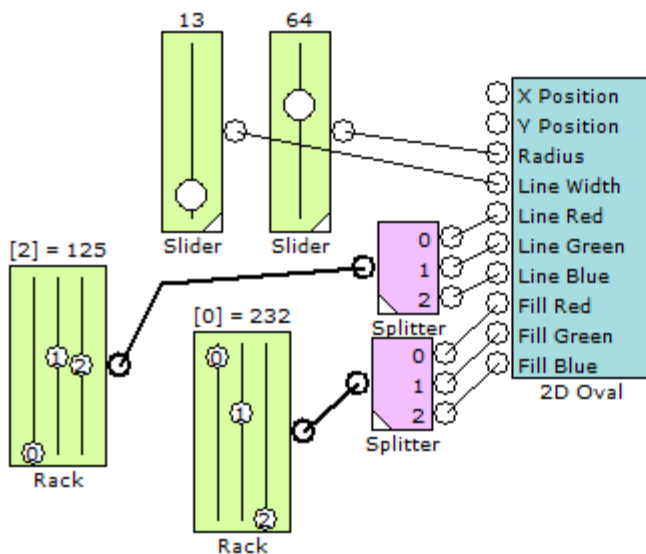
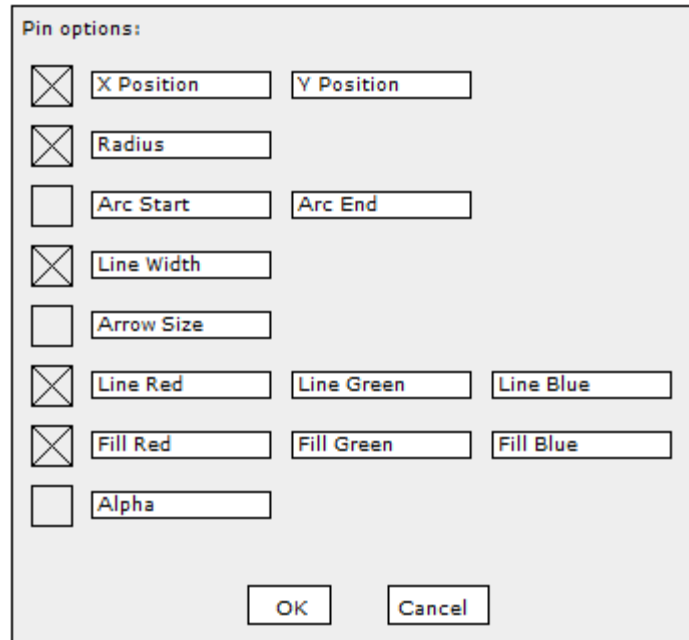
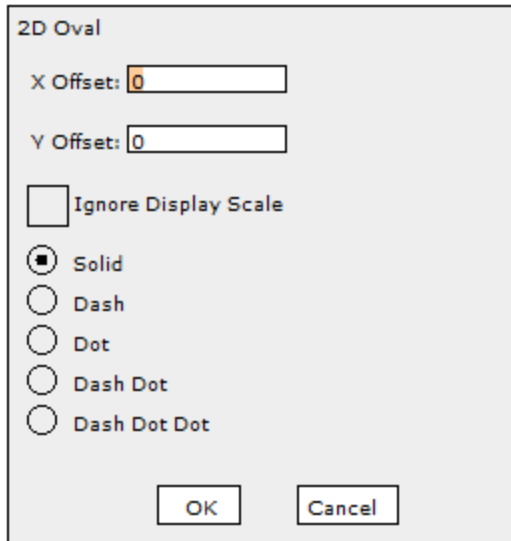


In the subroutine:



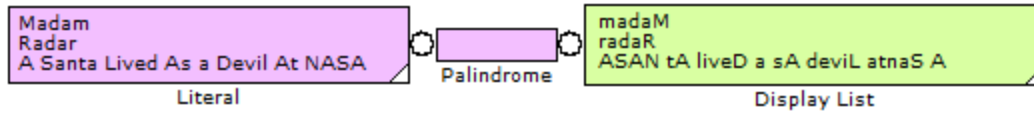
## 2D Oval

The 2D Oval component draws an oval on the 2D canvas. You can control a portion of the oval to make an arc. 2D - column 1



# Palindrome

The Palindrome component reverses the characters in a string. Arrays - column 6





# 2D Parabola

The 2D Parabola component draws a parabola outline using the X and Y points given. You can control the line thickness. 2D - column 1

Rotation:

Clockwise                       Zero is North  
 Counter Clockwise               Zero is East  
 Radians                               Zero is South  
 Degrees                                 Zero is West

OK      Cancel

Pin options:

X      Y  
 Width      Focus  
 Rotation  
 Line Width  
 Line Red      Line Green      Line Blue  
 Fill Red      Fill Green      Fill Blue  
 Alpha

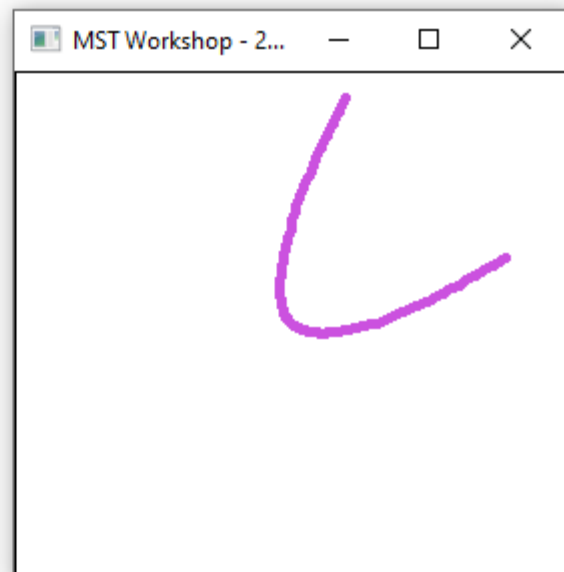
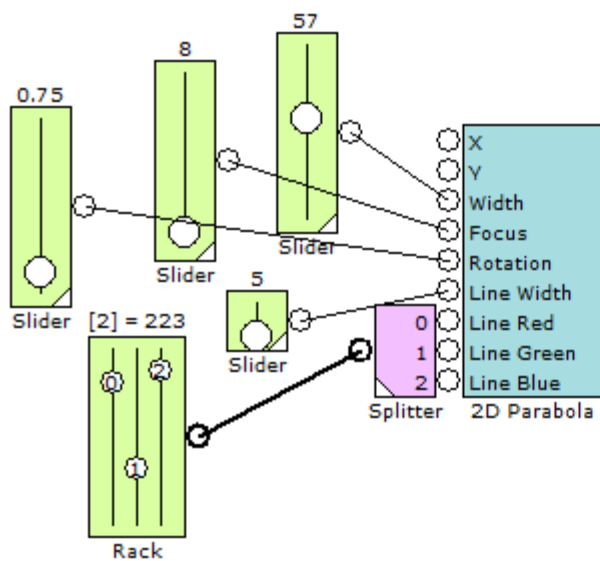
OK      Cancel

2D Parabola

Ignore Display Scale

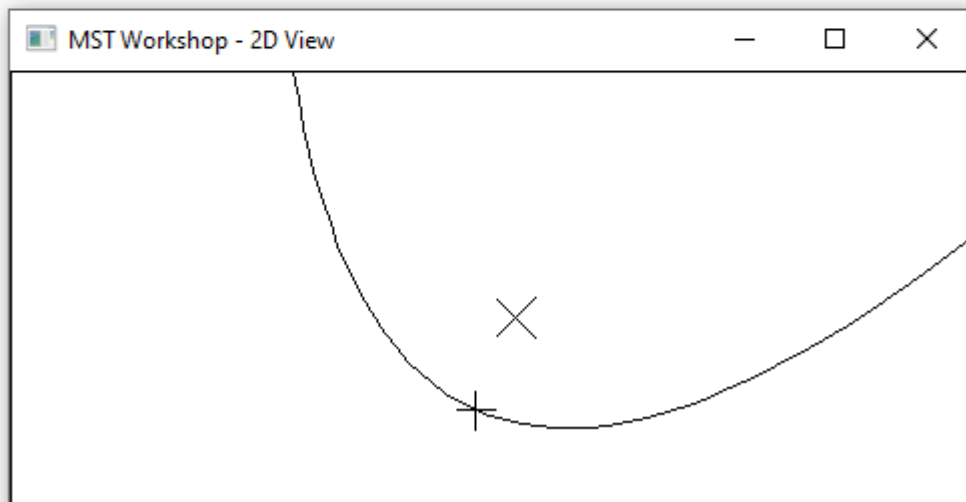
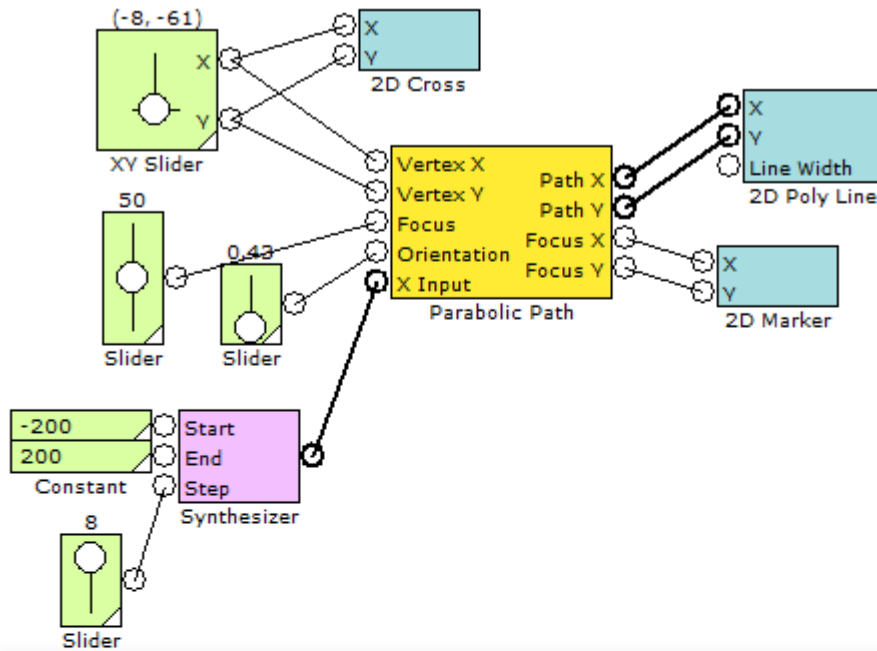
Solid  
 Dash  
 Dot  
 Dash Dot  
 Dash Dot Dot

OK      Cancel



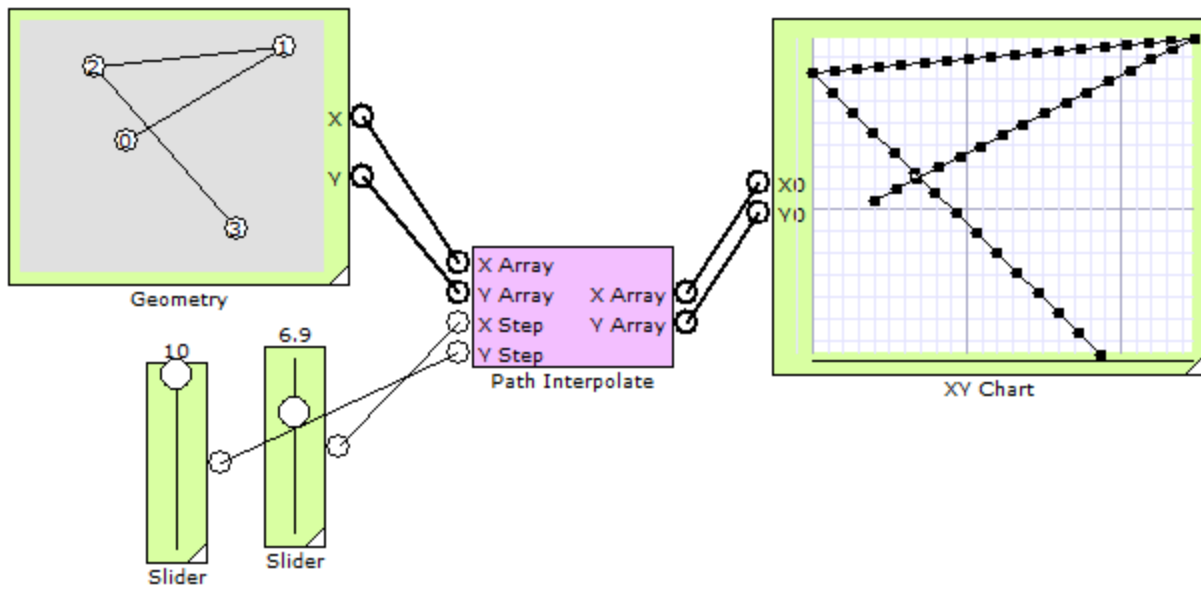
# Parabolic Path

The Parabolic Path component traces the parabola(X, Y) given the inputs and a range of X Inputs.  
Functions - column 9



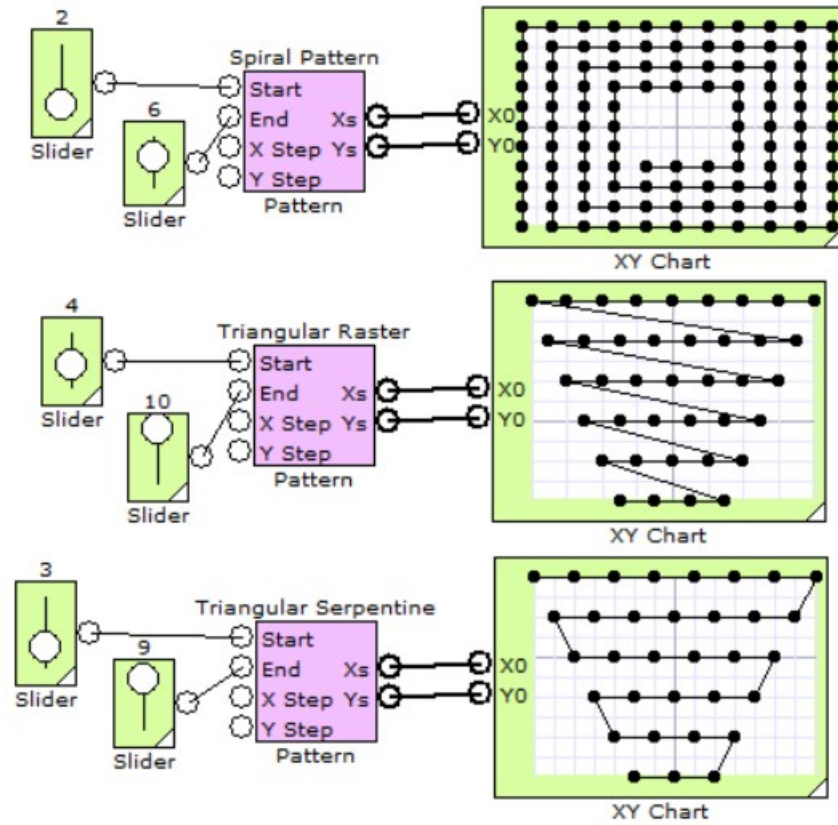
# Path Interpolate

The Path Interpolate component adds intermediate points to an X, Y path. Arrays - column 7



# Pattern

The Pattern component creates a set of X and Y coordinates arranged in a triangular or square spiral pattern. Arrays – column 1



# 2D Pendulum

The Pendulum component can drive other simulation objects like Gears. Place a Gear at the same X, Y position to connect the pendulum to the gear. 2D - column 3

Pin options:

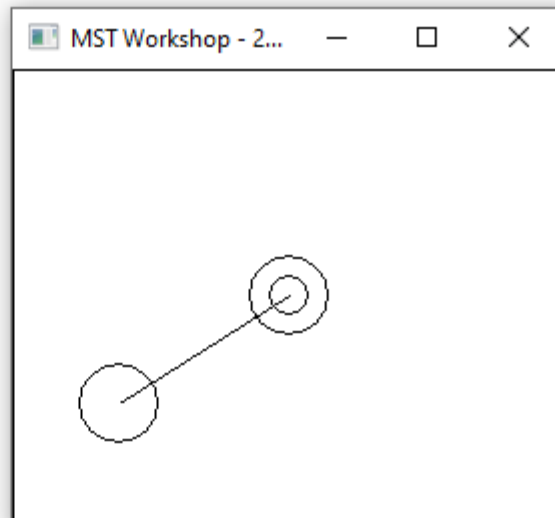
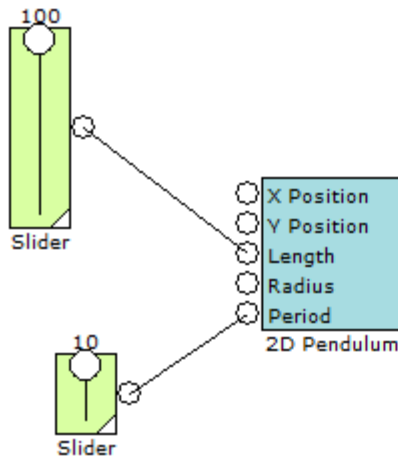
<input checked="" type="checkbox"/>	X Position	Y Position	
<input checked="" type="checkbox"/>	Length	Radius	
<input checked="" type="checkbox"/>	Period		
<input type="checkbox"/>	Red	Green	Blue
<input type="checkbox"/>	Alpha		

OK Cancel

2D Pendulum

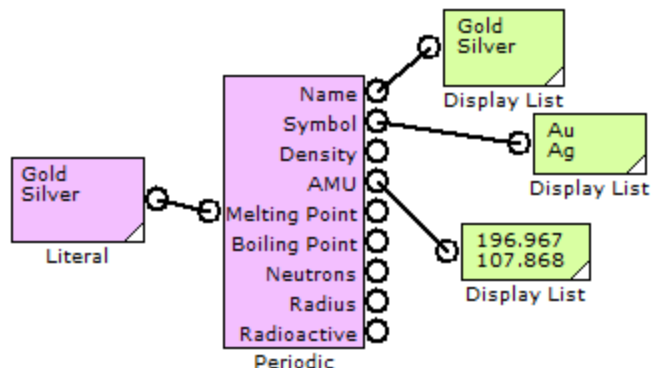
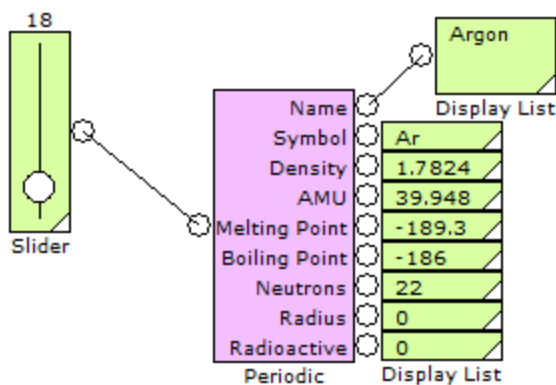
Sweep Angle:

OK Cancel



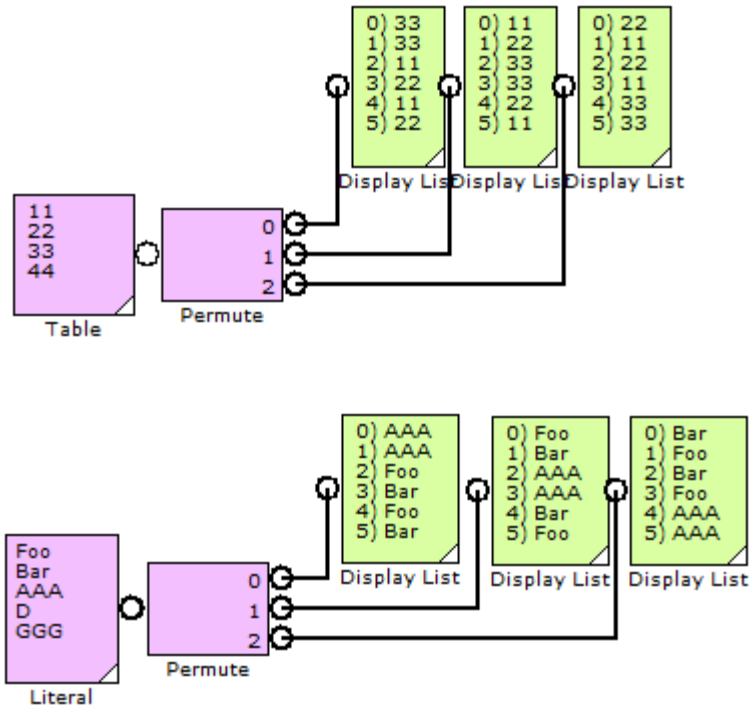
# Periodic

The Periodic component contains important information found in the periodic table of elements. Input the atomic number, the name, or the symbol for the desired element. Arrays - column 5



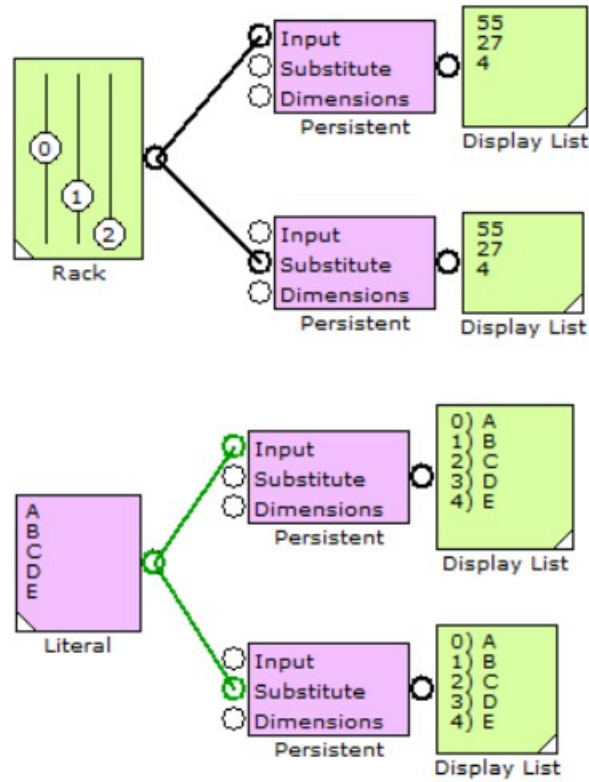
# Permute

The Permute component outputs a list of all arrangements of the input values. Arrays - column 5



# Persistent

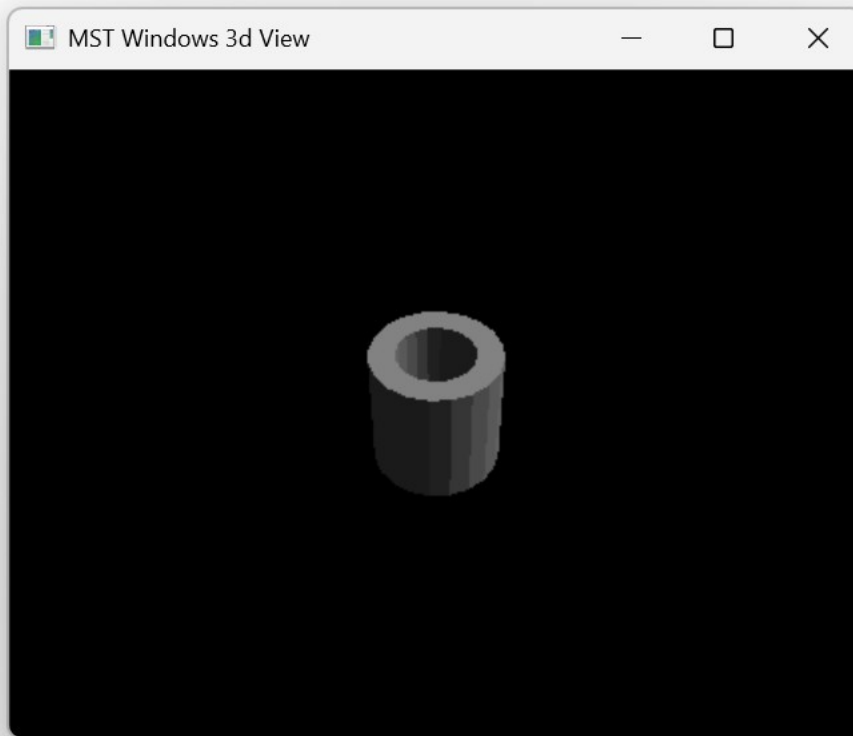
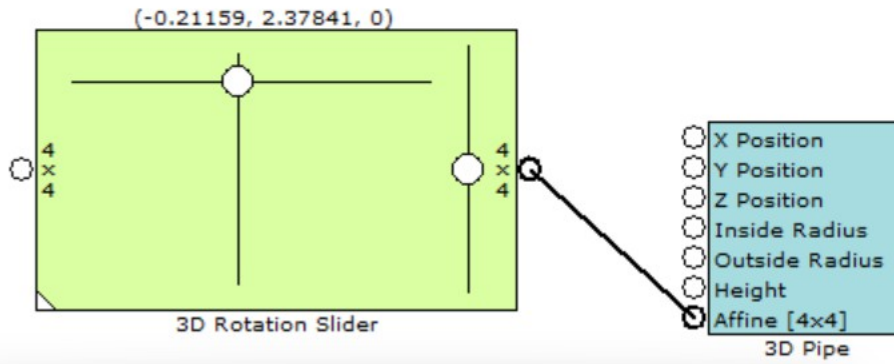
The Persistent component passes the input data when its available. Otherwise it passes the Substitute data. When no Input or Substitute data is present the Dimension input will be used to create an array of zeros with the specified dimensions. Arrays - column 2





## 3D Pipe

The 3D Pipe draws a Pipe in the 3D view window. The X, Y, and Z inputs set the position and size of the Pipe. The Affine [4x4] array input modifies the size, position, and orientation. 3D - column 2



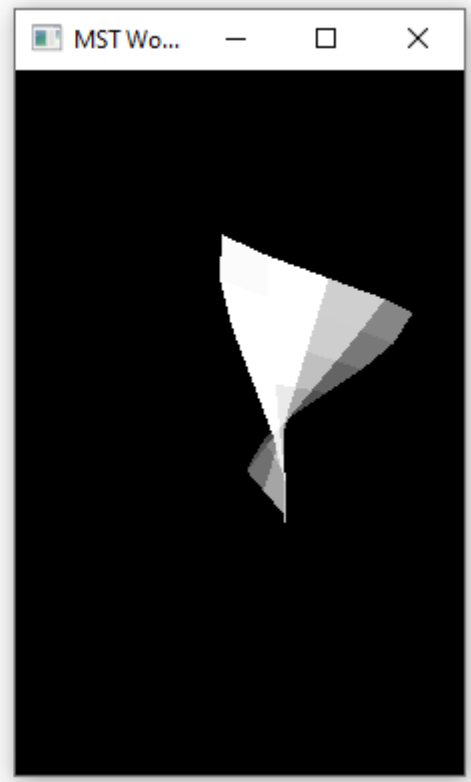
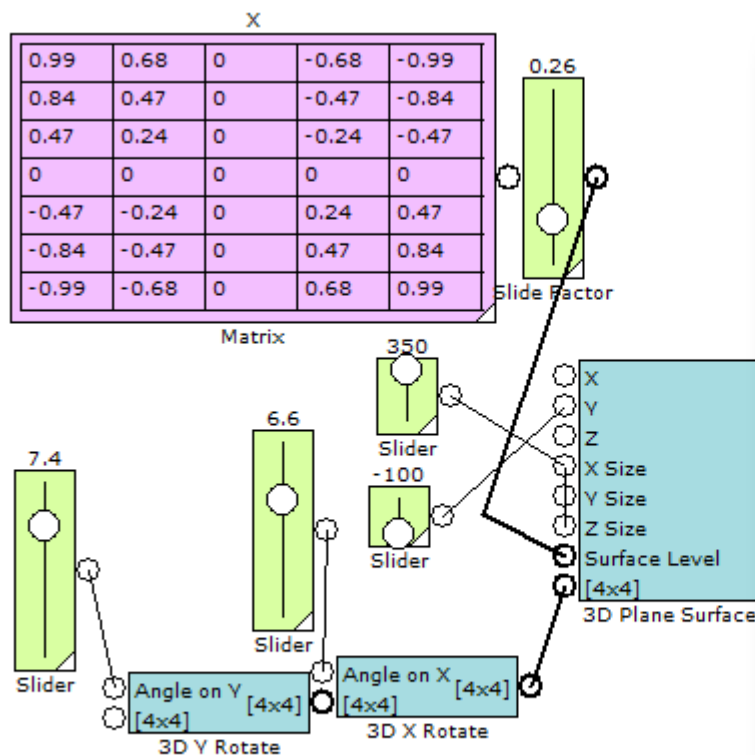
## 3D Plane Surface

The 3D Plane Surface component draws a plane with an embossed surface in the 3D view window. The Matrix and Mat Row Size inputs set the 2D data to be plotted. The Affine [4x4] array input modifies the size, position and orientation. 3D - column 5

Pin options:

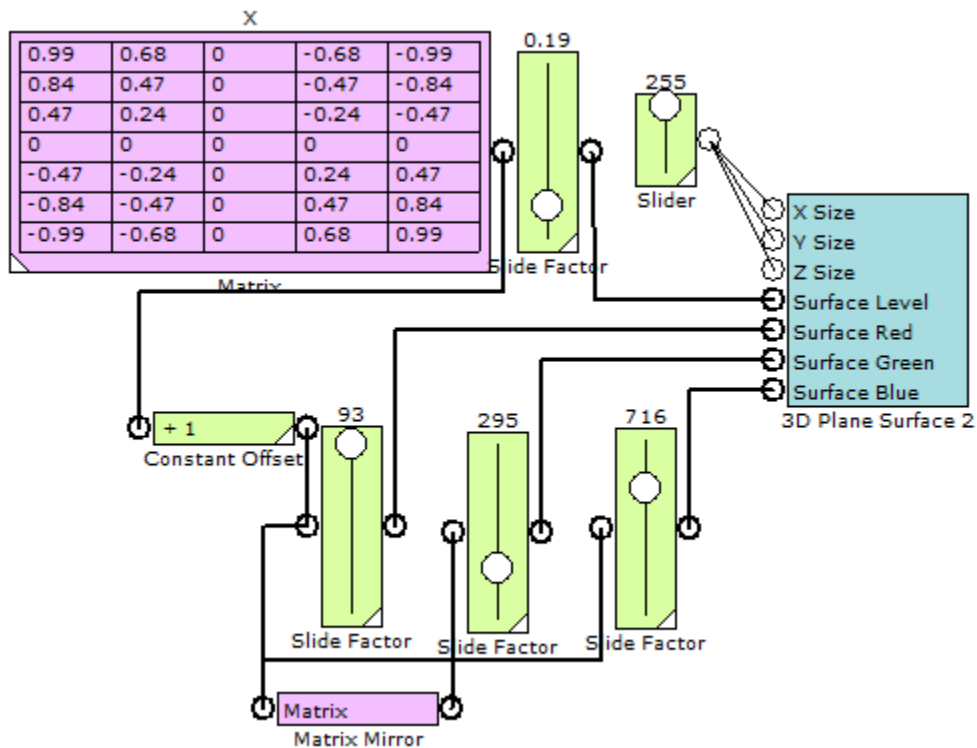
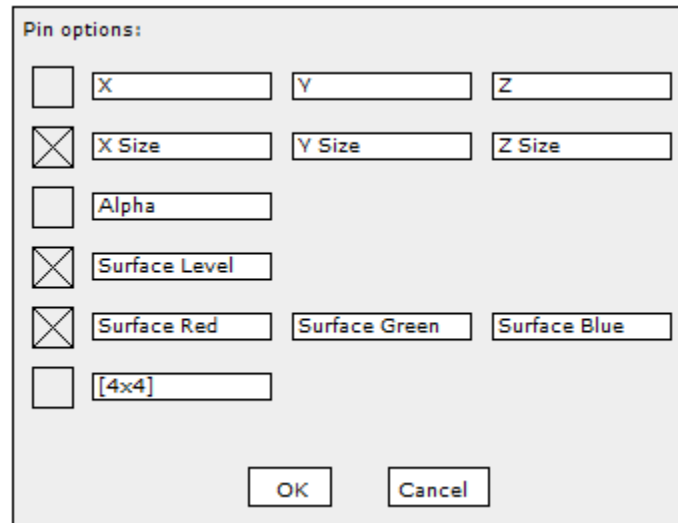
<input checked="" type="checkbox"/>	X	Y	Z
<input checked="" type="checkbox"/>	X Size	Y Size	Z Size
<input type="checkbox"/>	Red	Green	Blue
<input type="checkbox"/>	Alpha		
<input checked="" type="checkbox"/>	Surface Level		
<input checked="" type="checkbox"/>	[4x4]		

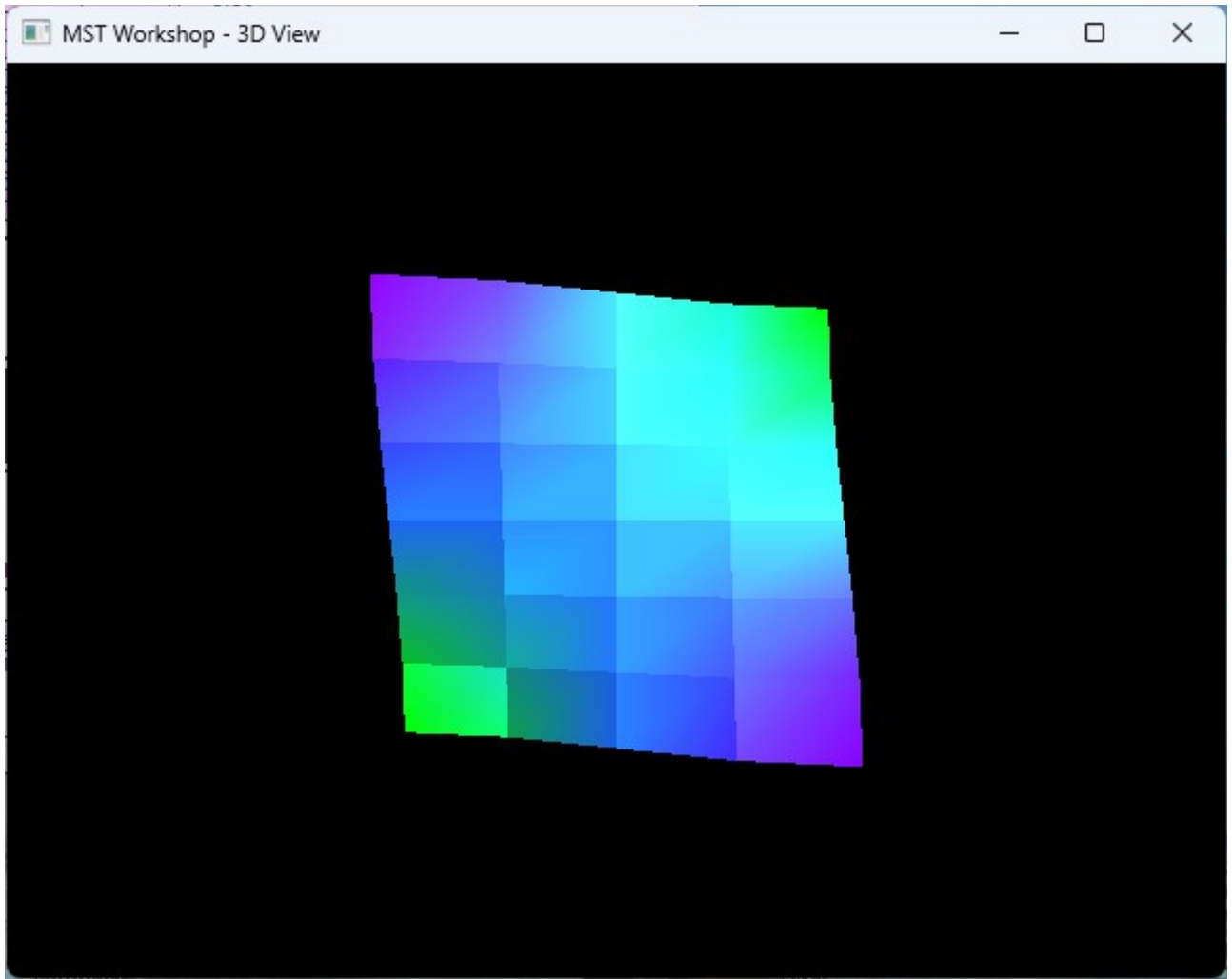
OK Cancel



## 3D Plane Surface 2

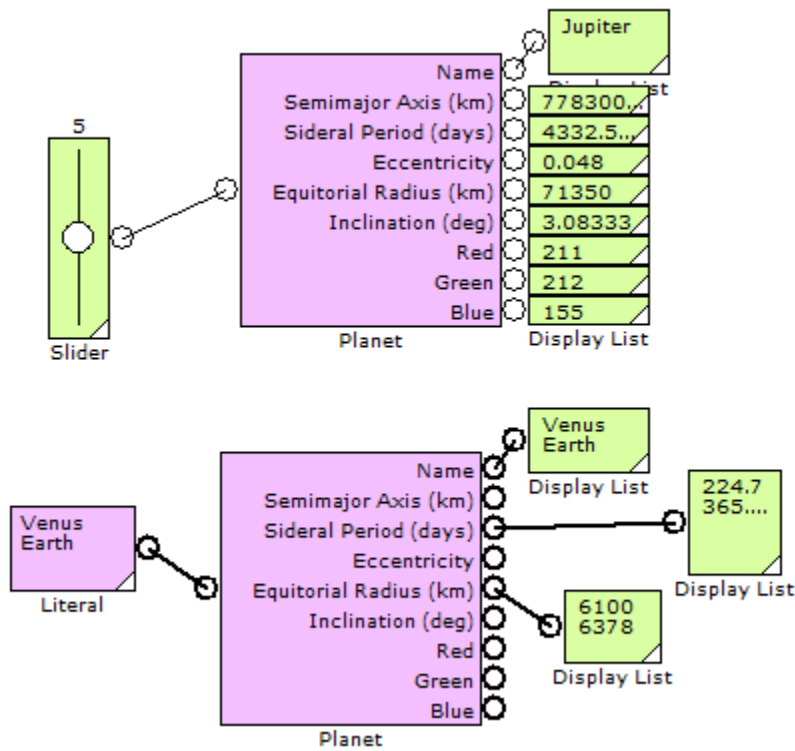
The 3D Plane Surface 2 component draws a plane with an embossed and color surface in the 3D view window. The Matrix and Mat Row Size inputs set the 2D data to be plotted. The Affine [4x4] array input modifies the size, position and orientation. 3D - column 5





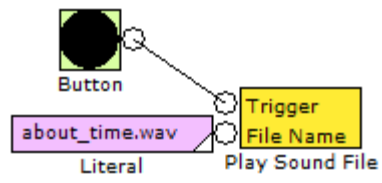
# Planet

The Planet component contains important information about the planets in our Solar System. Input a number 0 for Sun, 1 and up for planets. Alternatively you can input a string with the name of the planet desired. Arrays - column 5



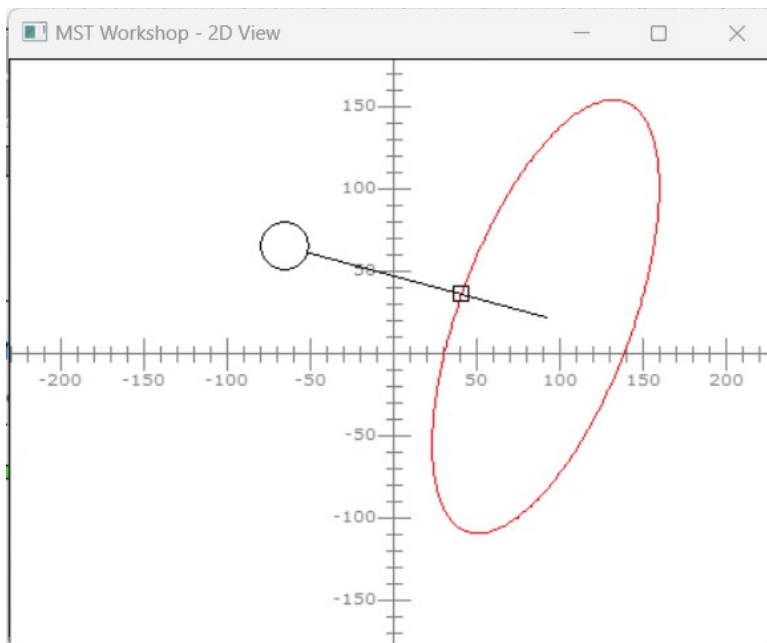
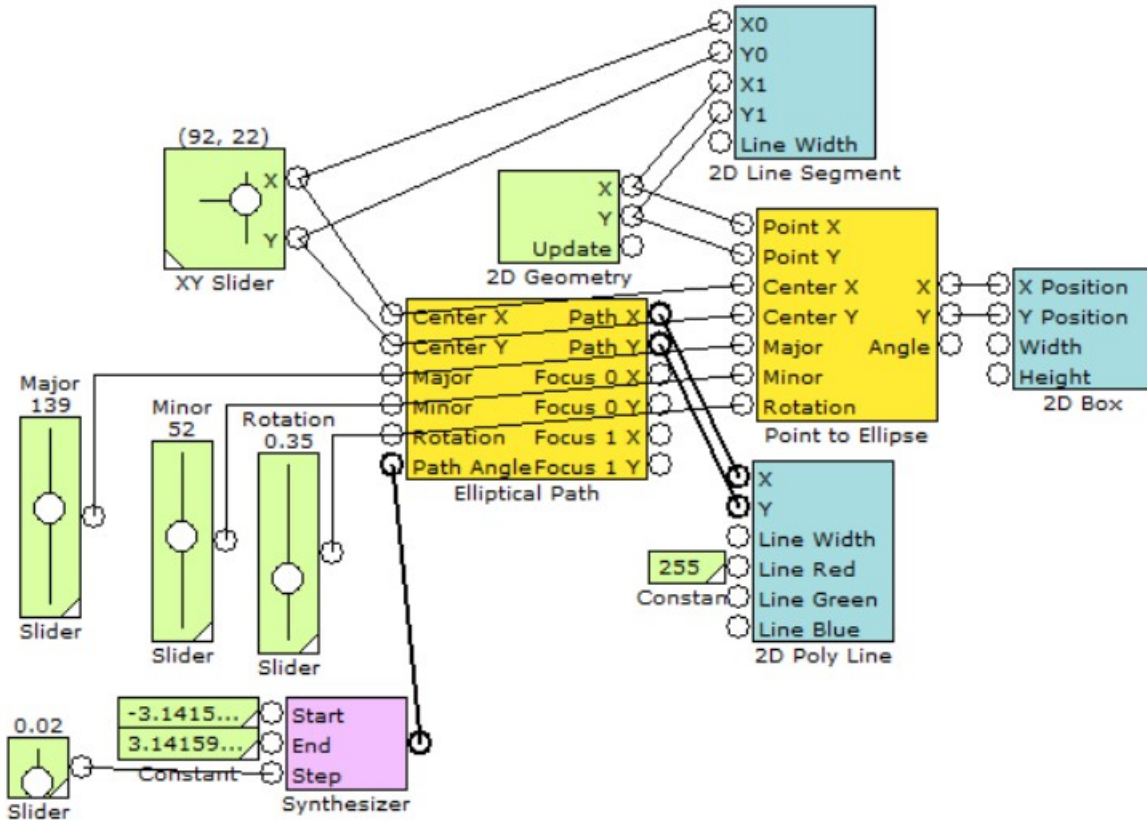
## Play Sound File

The Play Sound File component reads the given sound file and plays it.



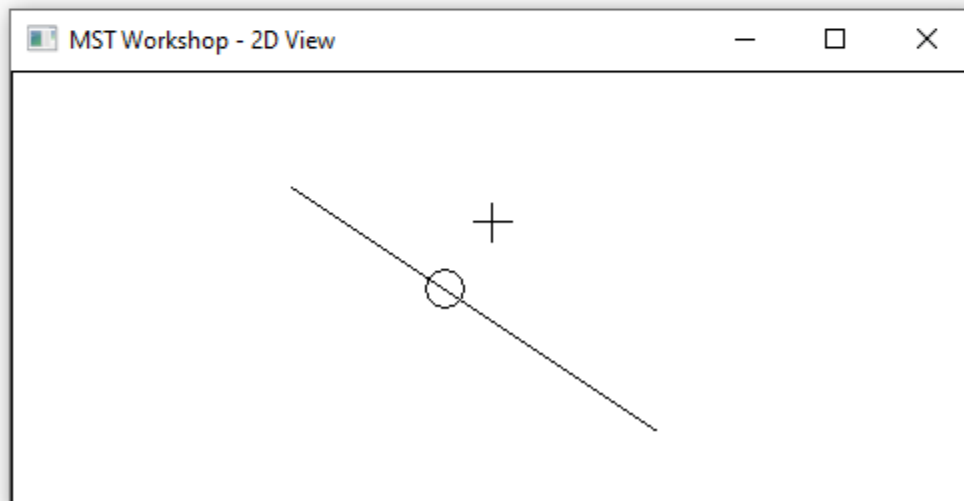
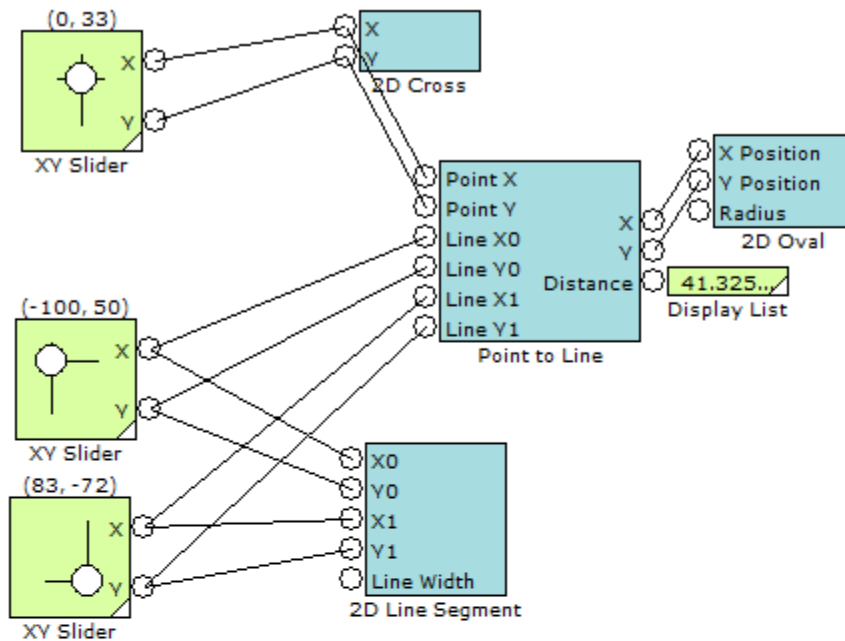
# Point to Ellipse

The Point to Ellipse component finds the coordinates and angle of the ellipse that will radiate from the center of the ellipse to the input point.



# Point to Line

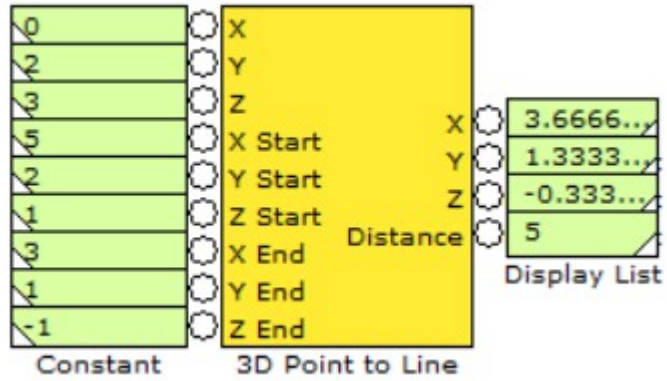
The Point to Line component computes the distance from a point to a line and gives the coordinates of the point on the line. 2D - column 2





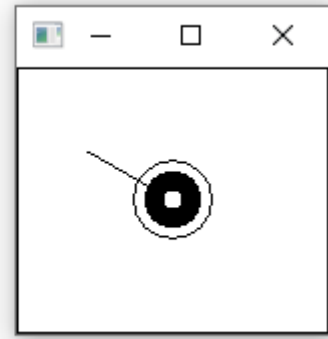
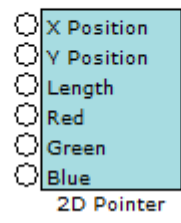
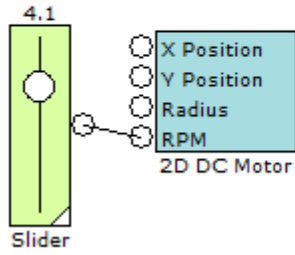
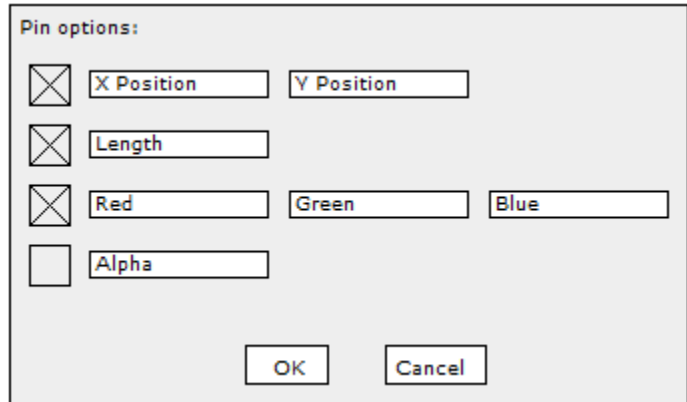
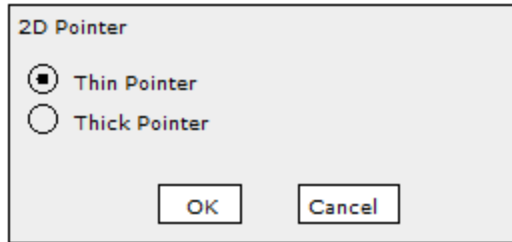
### 3D Point to Line

The 3D Point to Line component computes the distance from a point to a line and gives the coordinates of the point on the line. 3D - column 2



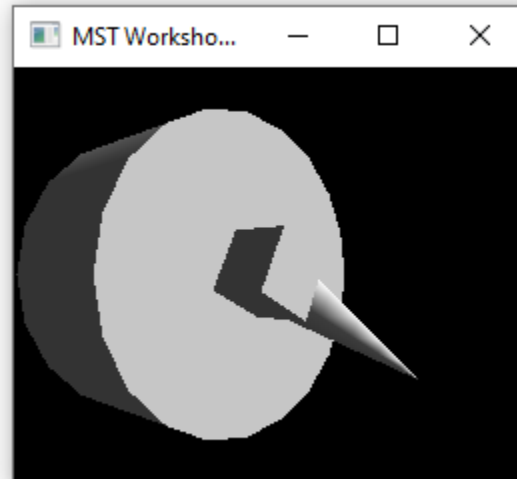
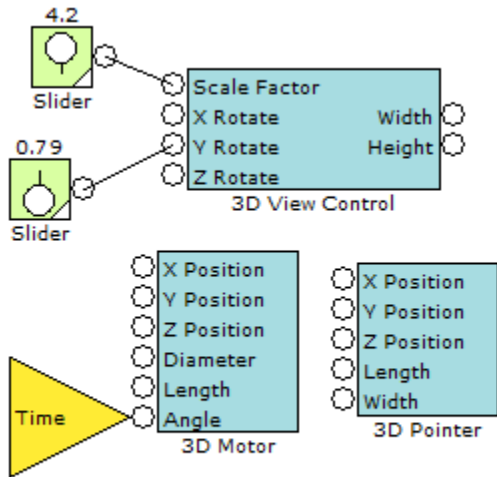
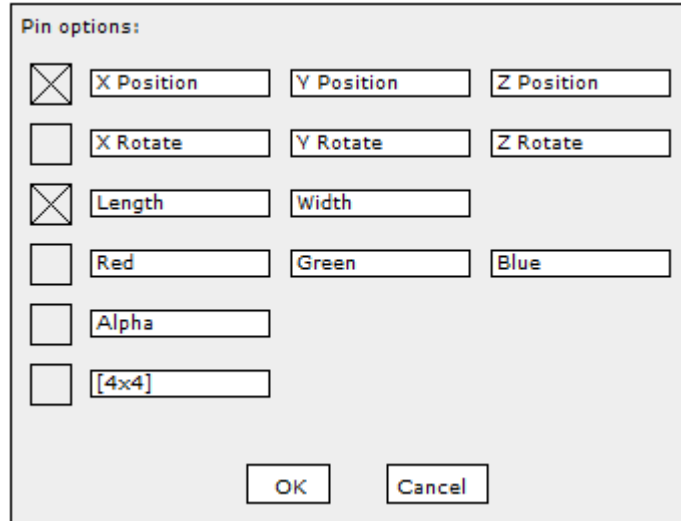
# 2D Pointer

The Pointer creates an indicator that can attach to a shaft on objects like Motors. Place the Pointer at the same X, Y position as a motor to connect the pointer to the motor. 2D - column 3



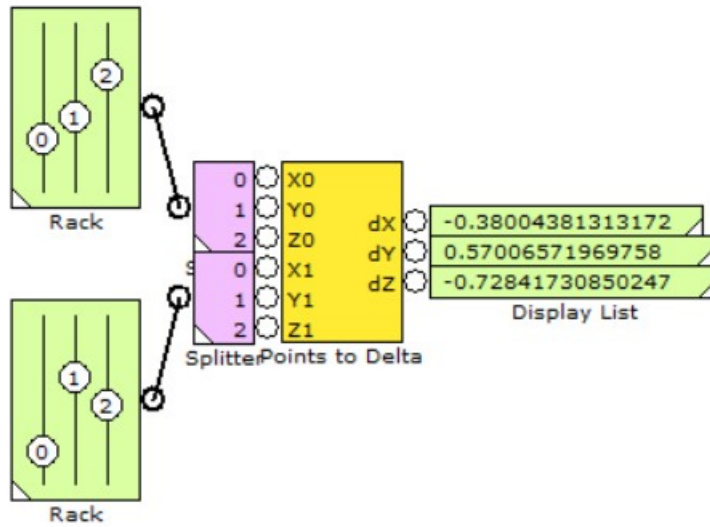
# 3D Pointer

The 3D Pointer component can be used to indicate the position of the motor or gear shaft, for example in a clock or stopwatch. Place the pointer at the X, Y, Z position and orientation of a motor, gear, or shaft. 3D - column 4



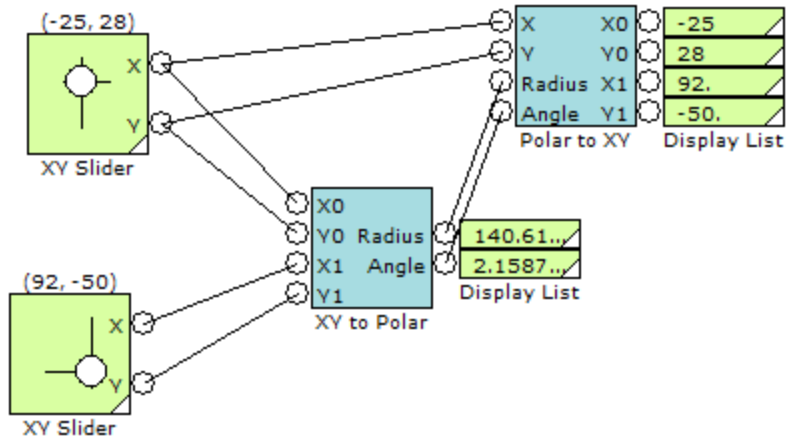
# Points to Delta

The Points to Delta component calculates the unit delta between two points. Functions – column 8



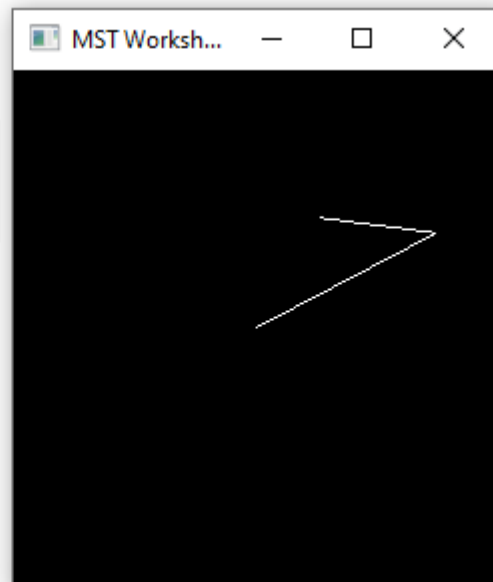
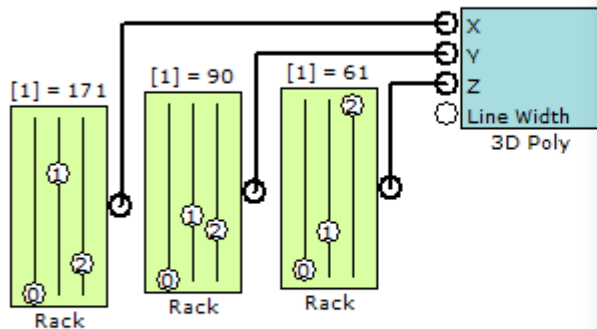
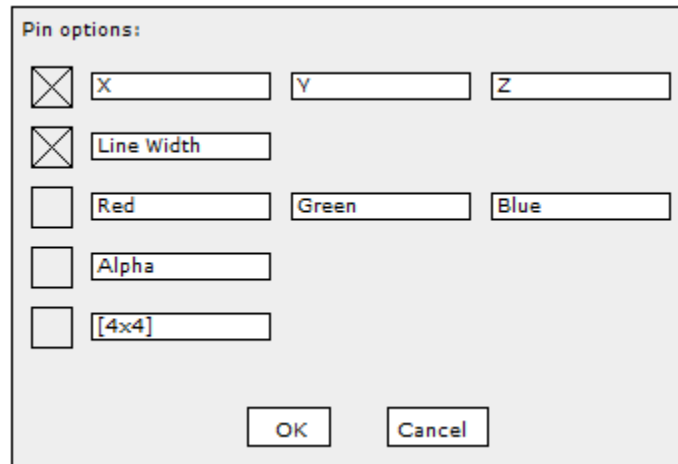
# Polar to XY

The Polar to XY component takes a radius and an angle then returns X, Y coordinates. 2D - column 5



## 3D Poly

The 3D Poly Line component draws a single segmented line in the 3D view window. The X, Y, and Z array inputs set the points along the line. The Affine [4x4] array input modifies the size, position and orientation. 3D - column 4



# 2D Poly Line

The 2D Poly Line component draws a line from point to point. Input a series of X and Y points. 2D - column 1

2D Poly Line

Ignore Display Scale

Solid

Dash

Dot

Dash Dot

Dash Dot Dot

OK Cancel

Pin options:

X  Y

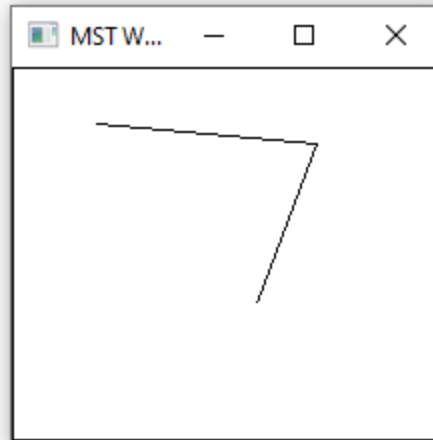
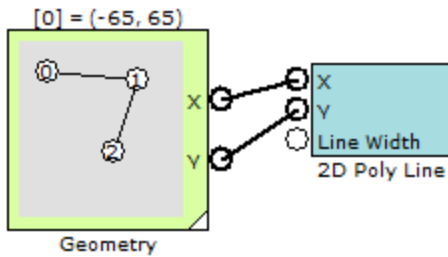
Line Width

Line Red  Line Green  Line Blue

Alpha

Corners

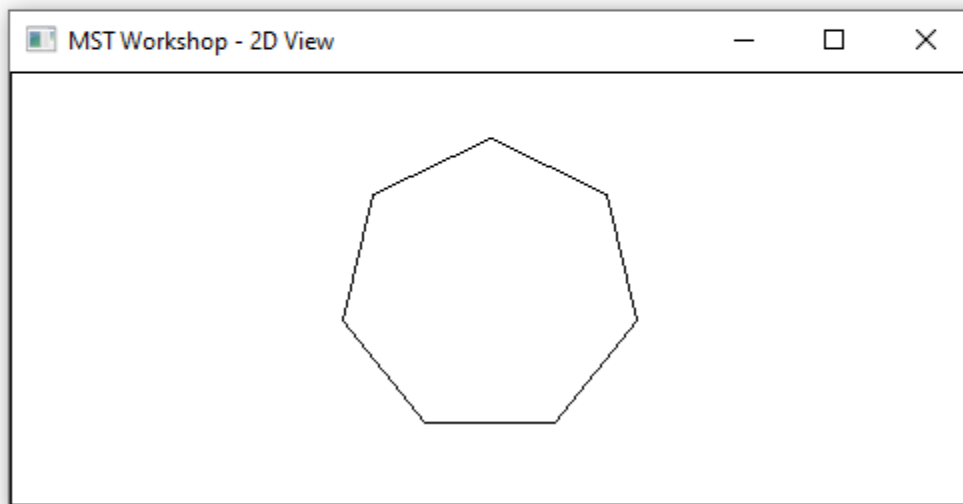
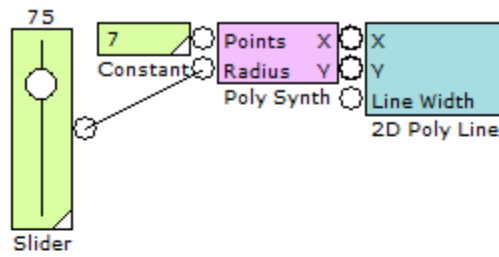
OK Cancel



# Poly Synth

The Poly Synth component creates a regular polygon and outputs its X and Y coordinates.  
- column 1

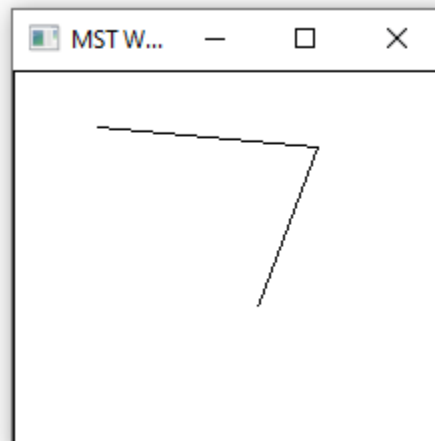
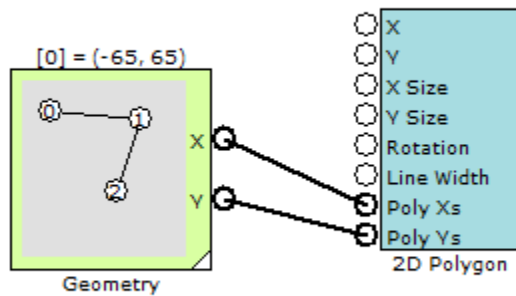
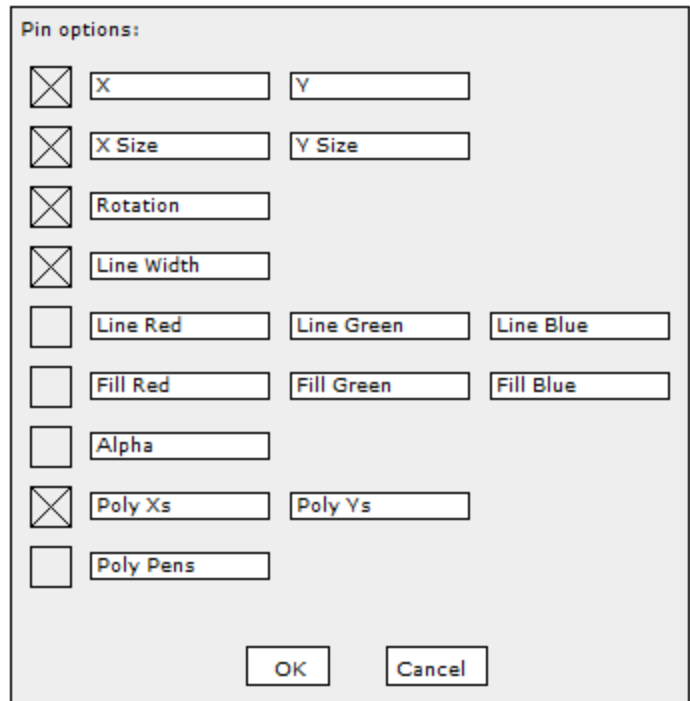
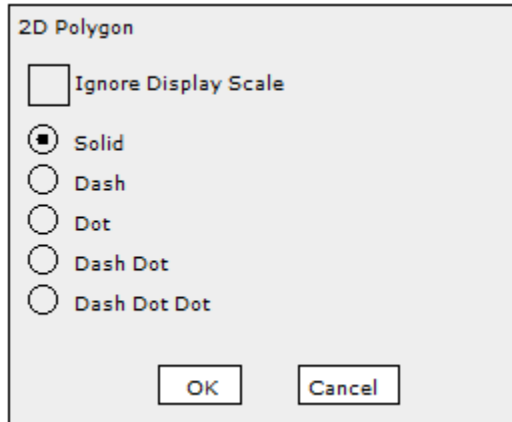
Arrays





# 2D Polygon

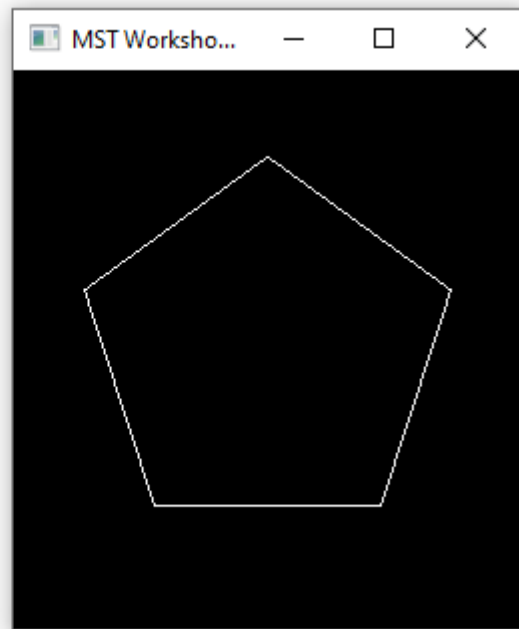
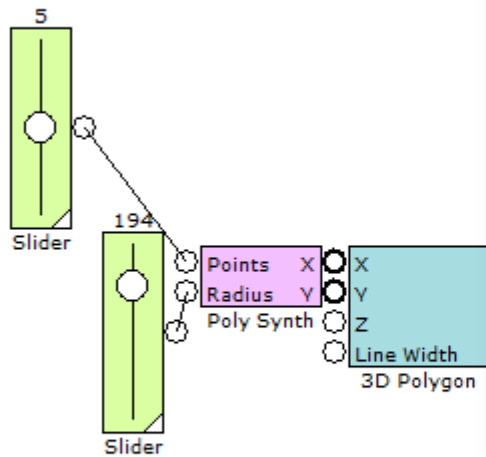
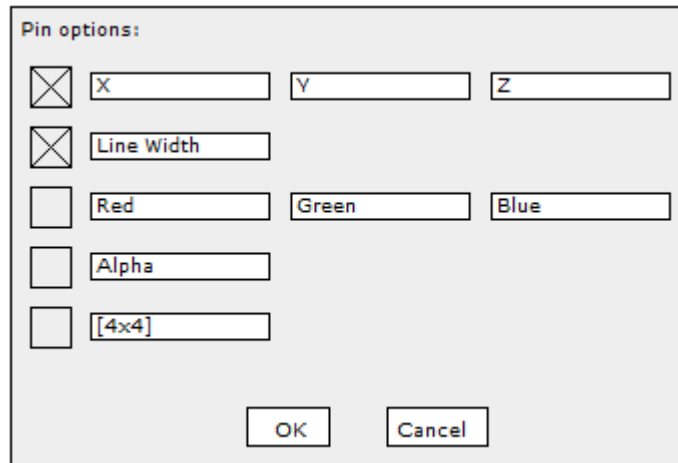
The 2D Polygon component draws a polygon outline using the X and Y points given. You can control the line thickness. 2D - column 1



# 3D Polygon

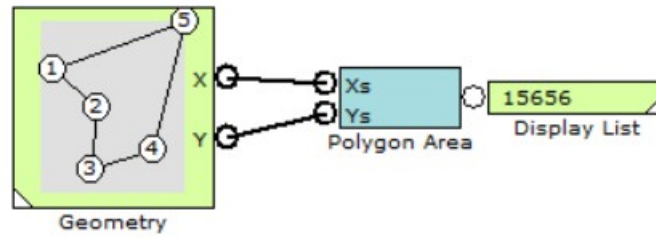
The 3D Polygon draws a 3D polygon in the 3D view window. The X, Y, and Z inputs set the position and size of the polygon. The Affine [4x4] array input modifies the size, position and orientation.

3D - column 4



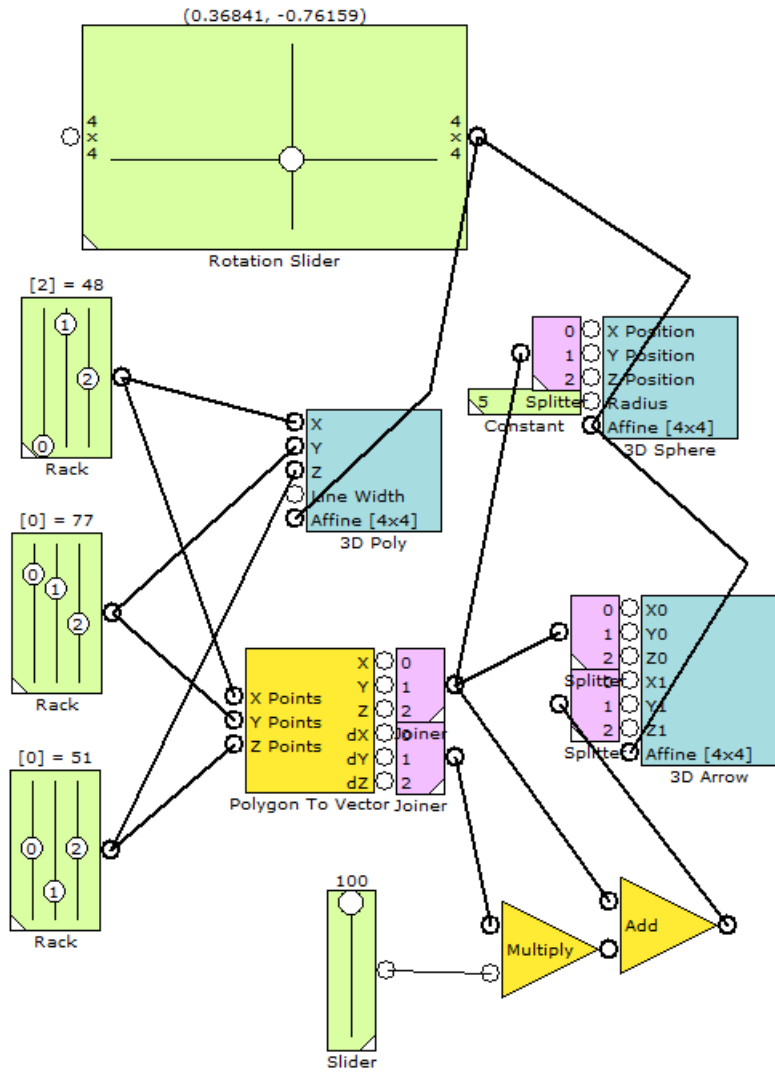
# Polygon Area

The Polygon Area component computes the area of a polygon from its XY coordinates. 2D – column 2



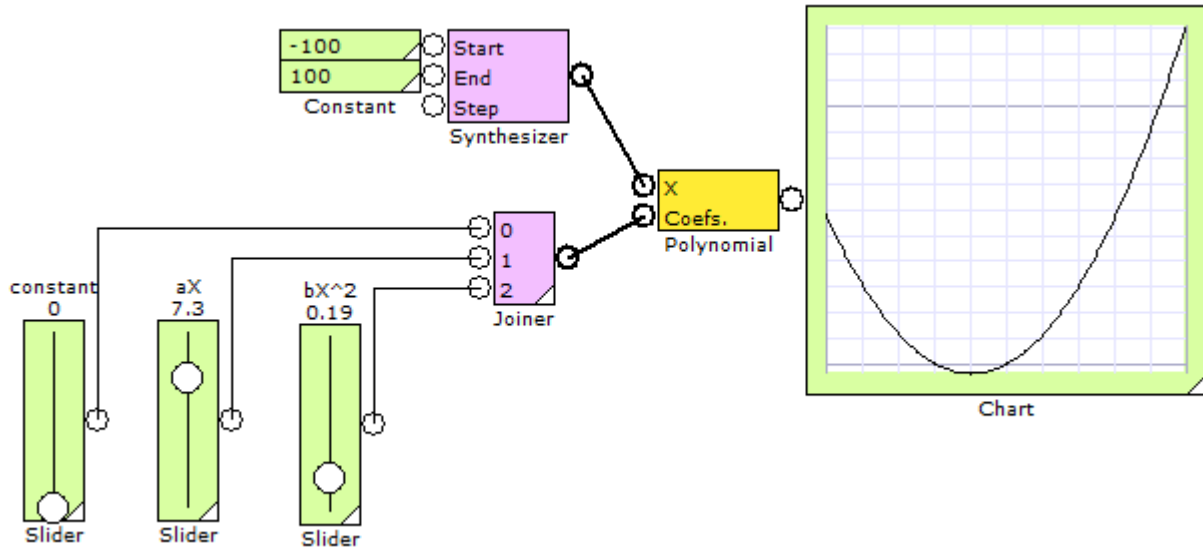
# Polygon To Vector

The Polygon To Vector component calculates the origin of a vector from the center of a polygon that is perpendicular to the polygon. Functions – column 8



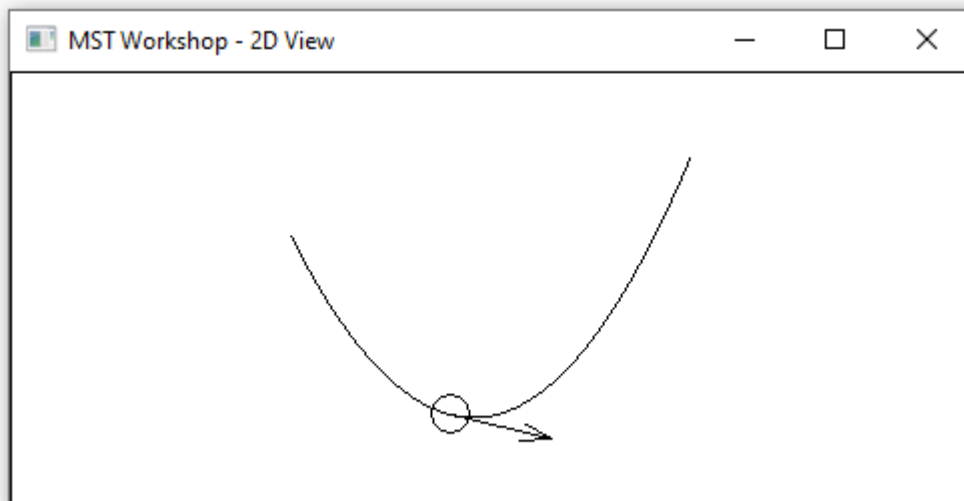
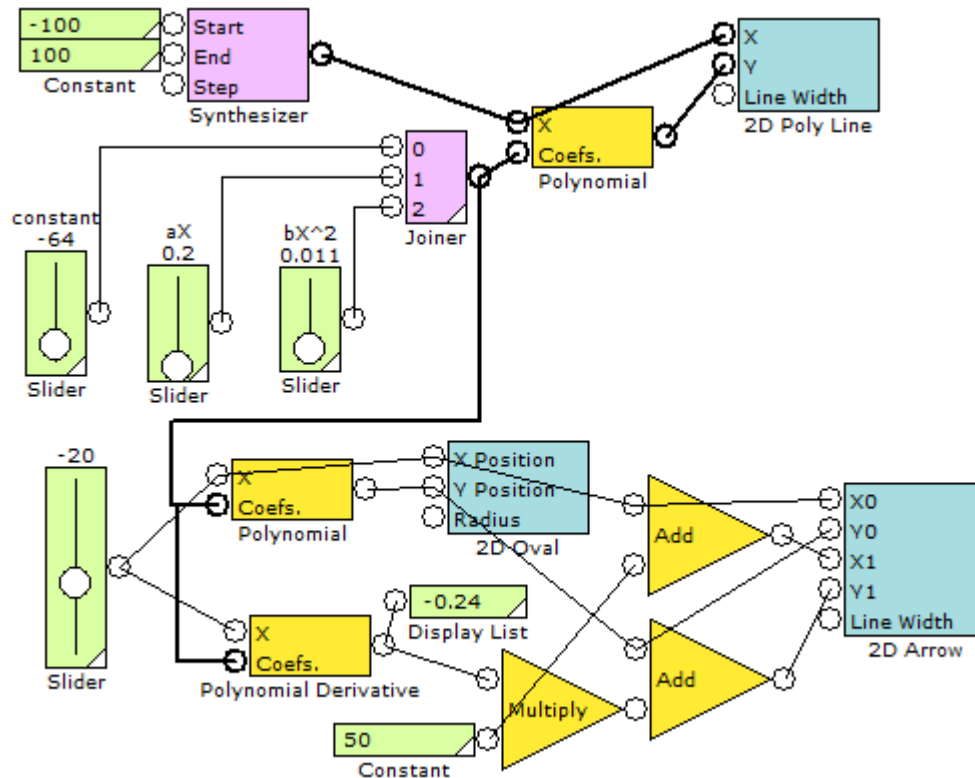
# Polynomial

The Polynomial component calculates the polynomial using the coefficients and the X input.  
Functions - column 6



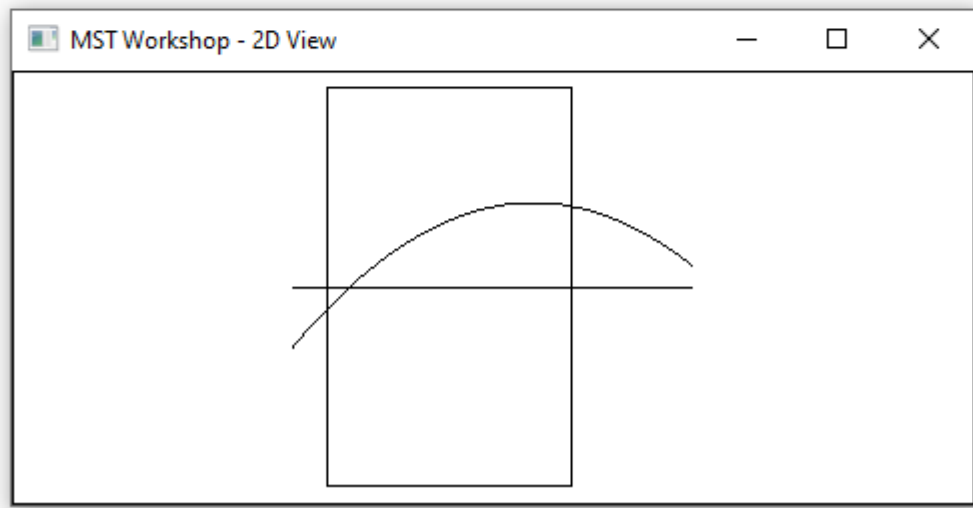
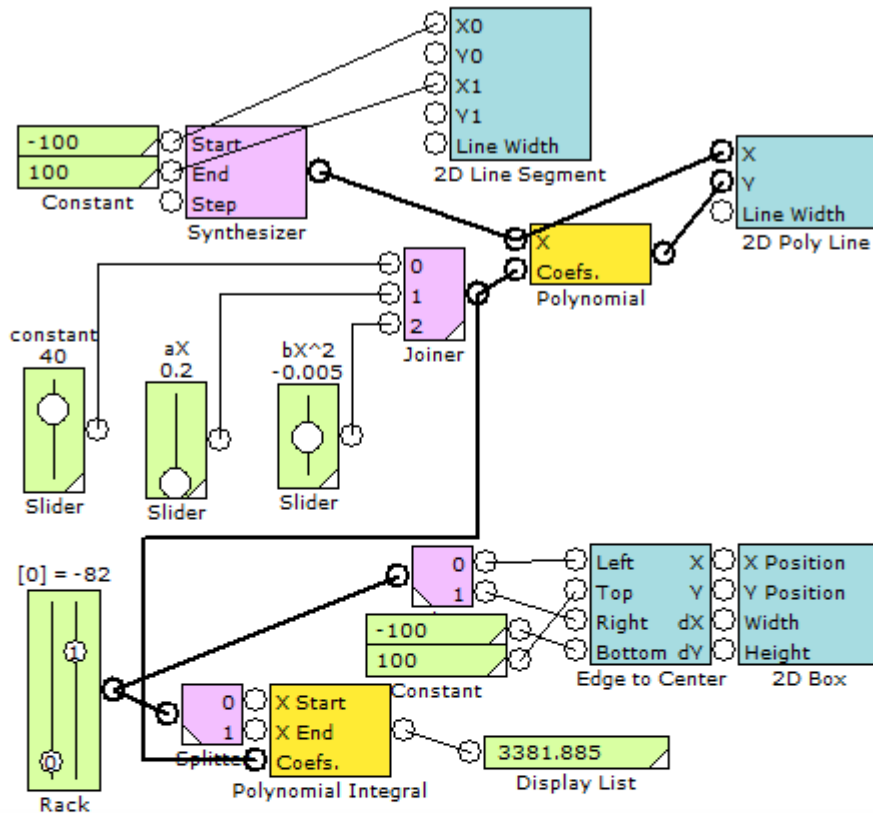
# Polynomial Derivative

The Polynomial Derivative component calculates the derivative of a polynomial using the coefficients and the X input. Functions - column 6



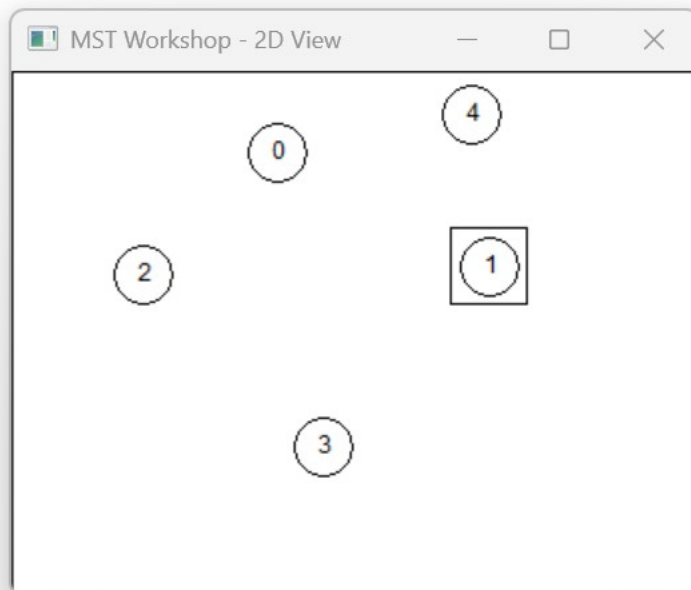
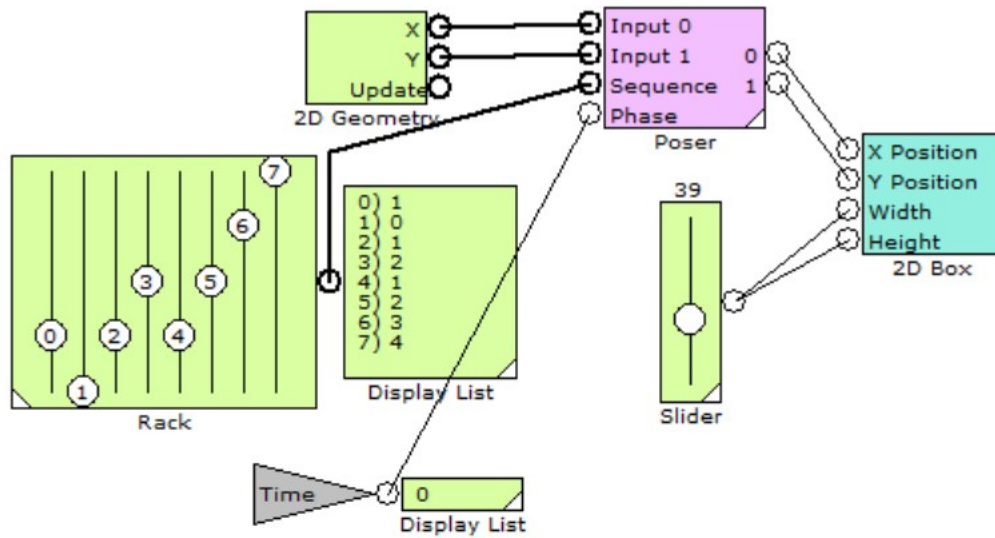
# Polynomial Integral

The Polynomial Integral component calculates the integral of a polynomial using the coefficients and the X input from X Start to X End. Functions - column 6



# Poser

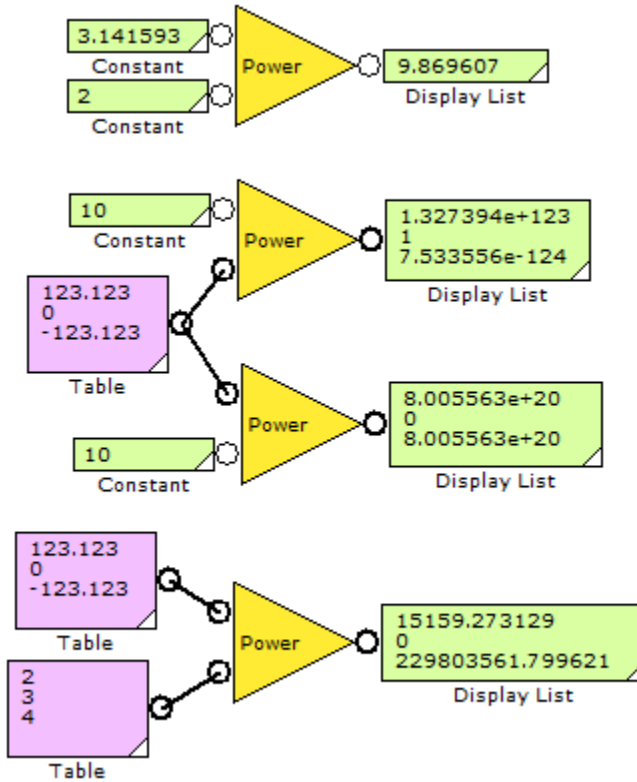
The Poser component takes arrays of values and transitions from one Pose to another based on the Phase input. It outputs a smooth transition from one pose to another.      Arrays - column 2





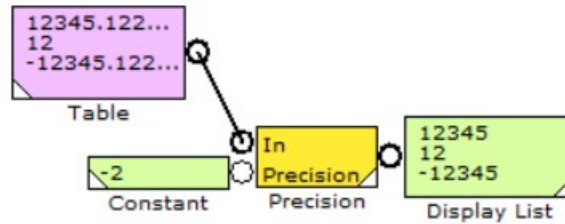
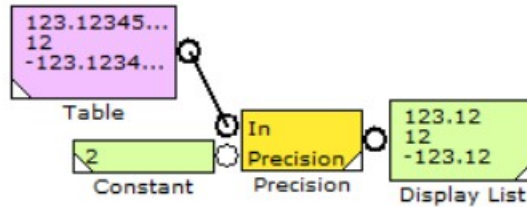
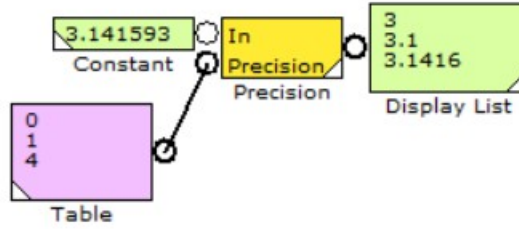
# Power

The Power component calculates the top input value raised to the bottom input value. Inputs can be scalar or arrays. Functions - column 1



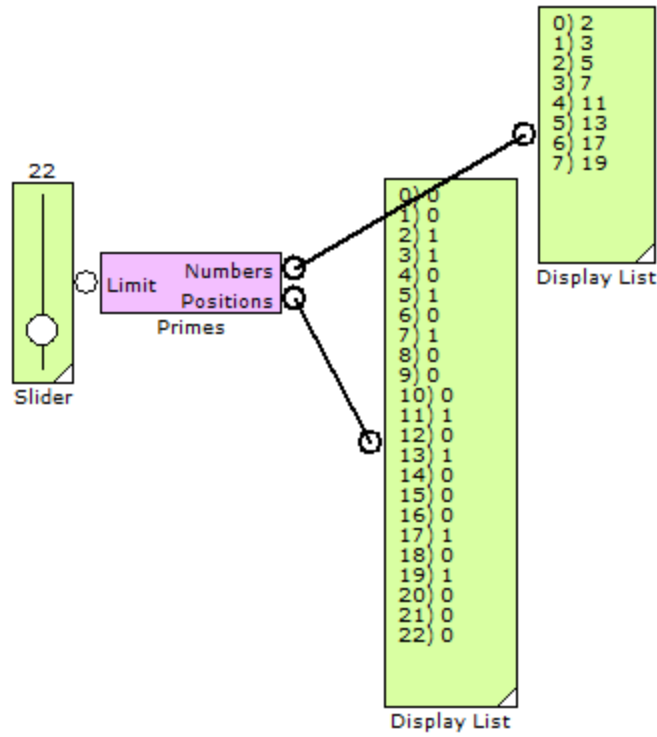
# Precision

The Precision component outputs a value with the specified precision. Inputs can be scalar or arrays.  
Functions - column 2



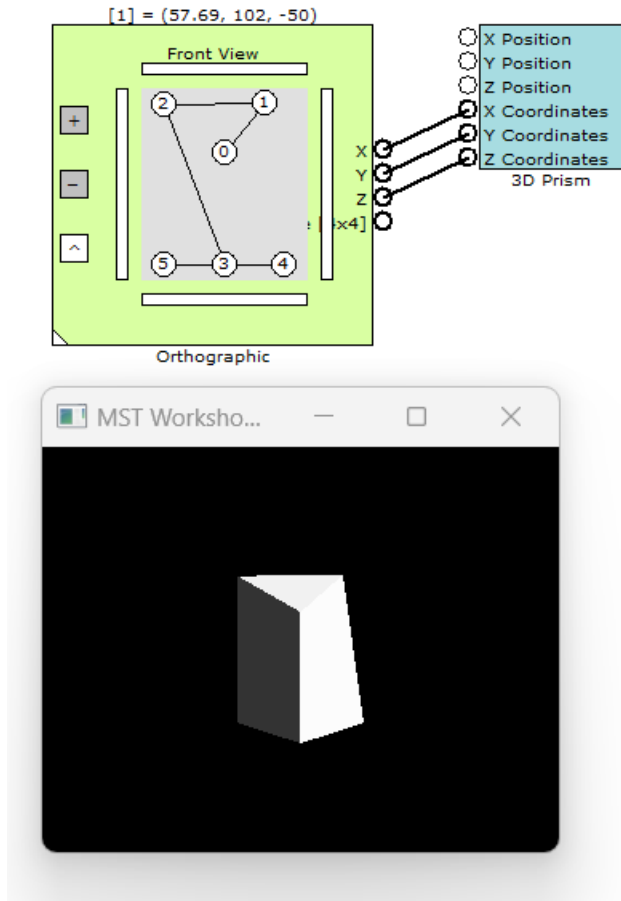
# Primes

The Primes component outputs a list of primes less than the input limit. Arrays - column 5



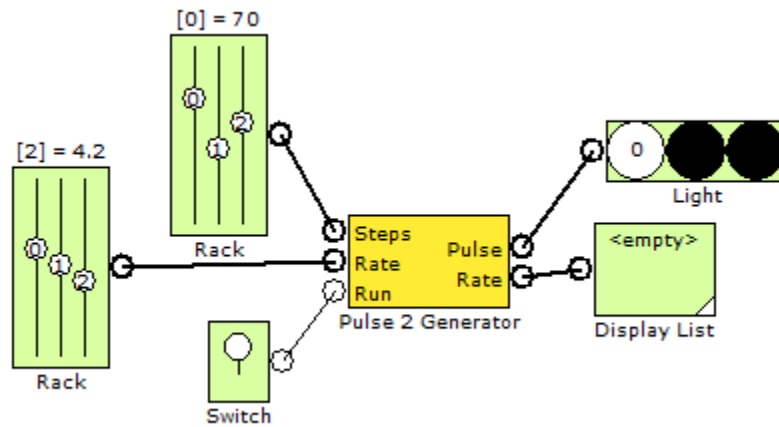
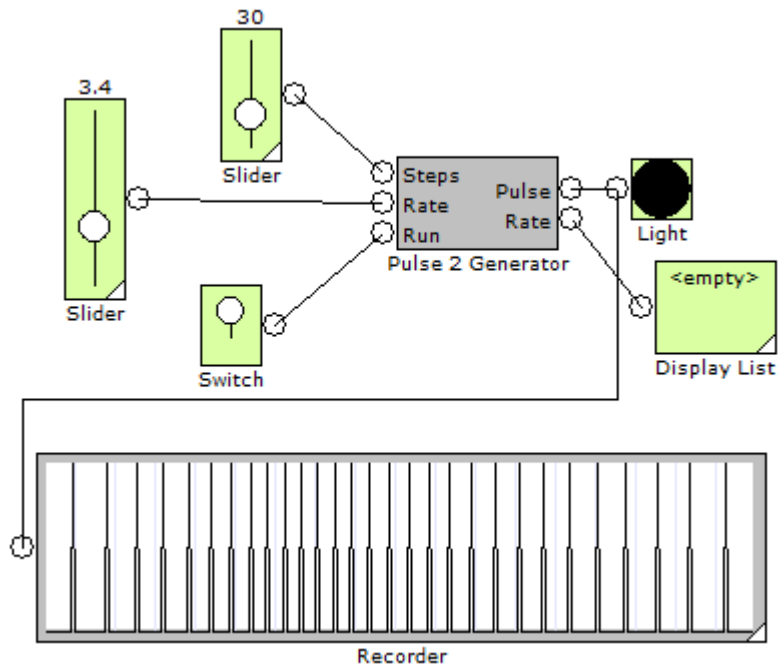
# 3D Prism

The 3D Prism draws a prism in the 3D view window. The X, Y, and Z inputs set the position and size of the prism. The Affine [4x4] array input modifies the size, position, and orientation of the pyramid. 3D – column 4



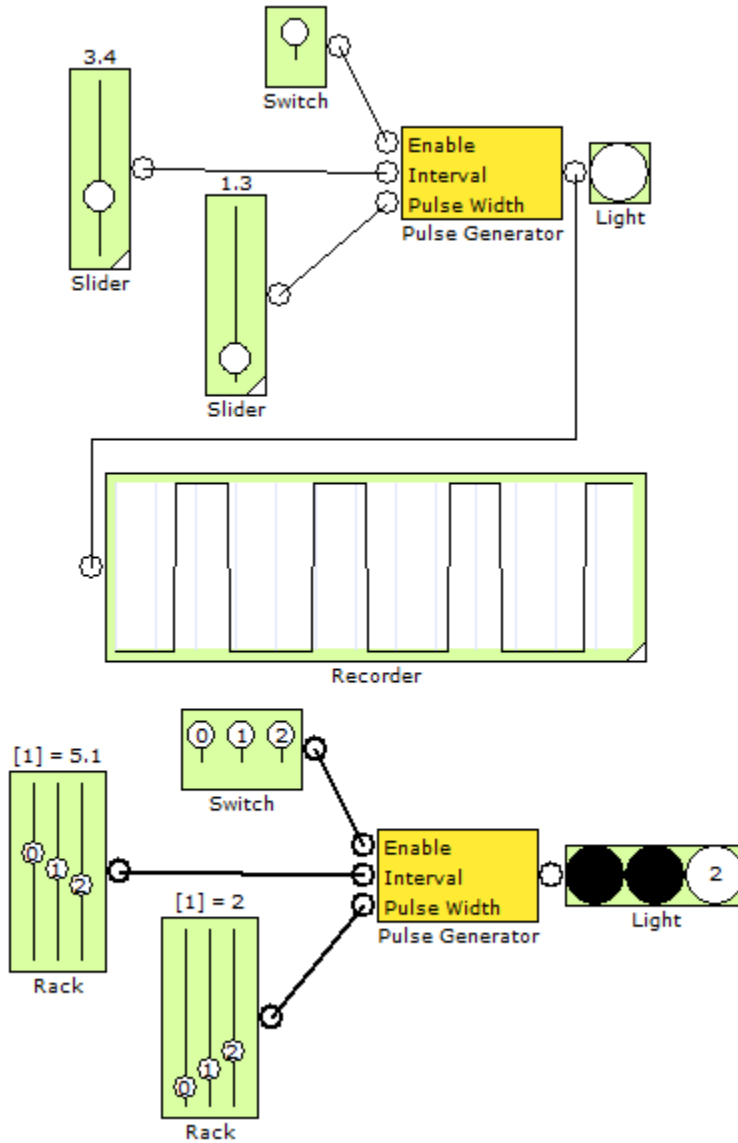
## Pulse 2 Generator

The Pulse 2 Generator component outputs a short pulse at a specified interval. Use this pulse generator to drive a stepper motor. Inputs are seconds. Functions - column 8



# Pulse Generator

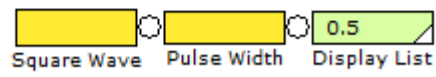
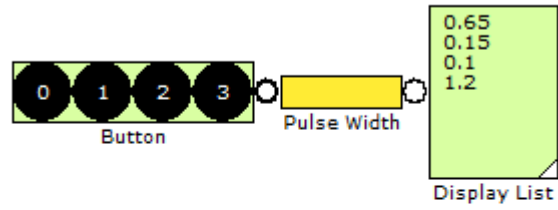
The Pulse Generator component outputs a short pulse at a specified interval. Inputs are seconds.  
Functions - column 8



# Pulse Width

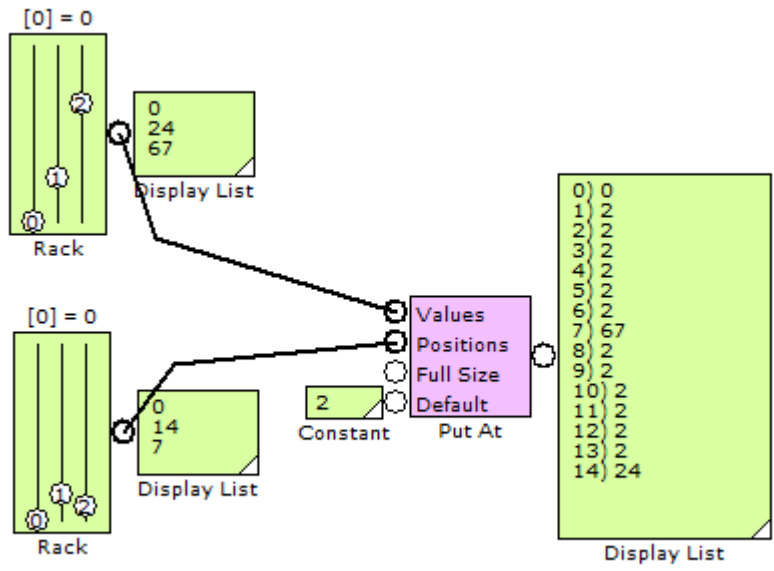
The Pulse Width component measures the width of the input pulse.

Functions - column 7



# Put At

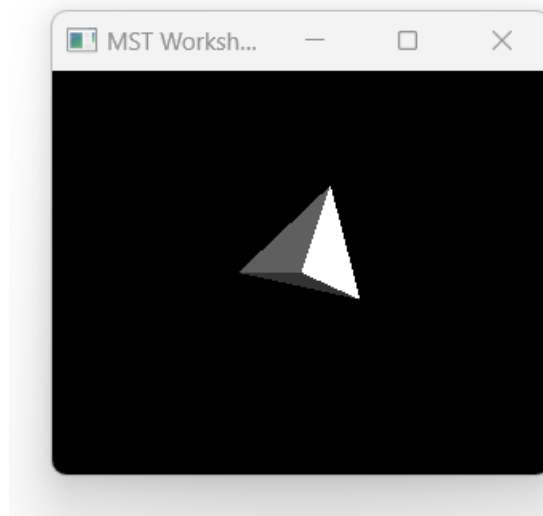
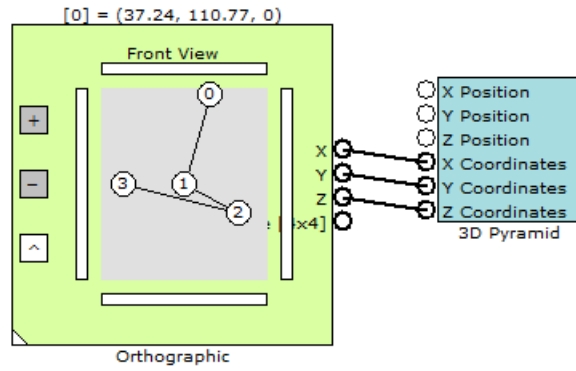
The Put At component outputs a list of values in the specified positions, open positions are filled with the default. Arrays - column 2





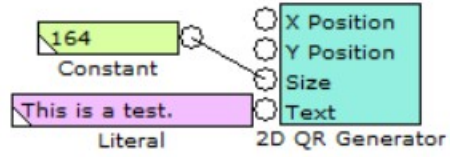
## 3D Pyramid

The 3D Pyramid draws a pyramid in the 3D view window. The X, Y, and Z inputs set the position and size of the pyramid. The Affine [4x4] array input modifies the size, position, and orientation of the pyramid. 3D – column 4



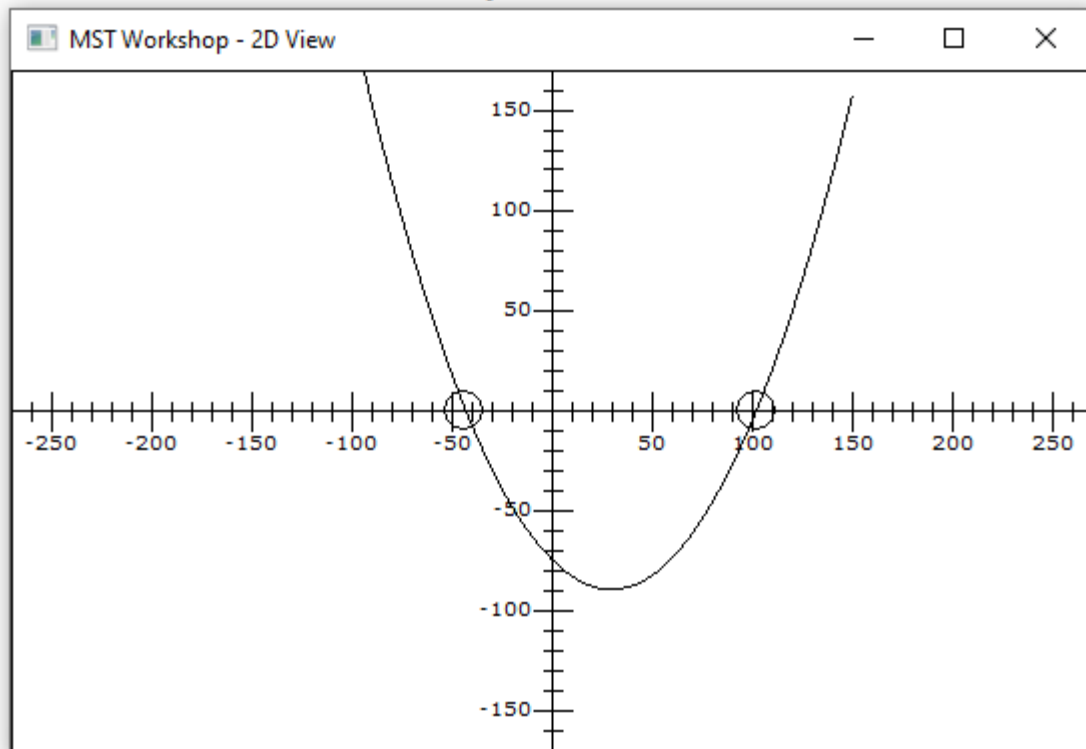
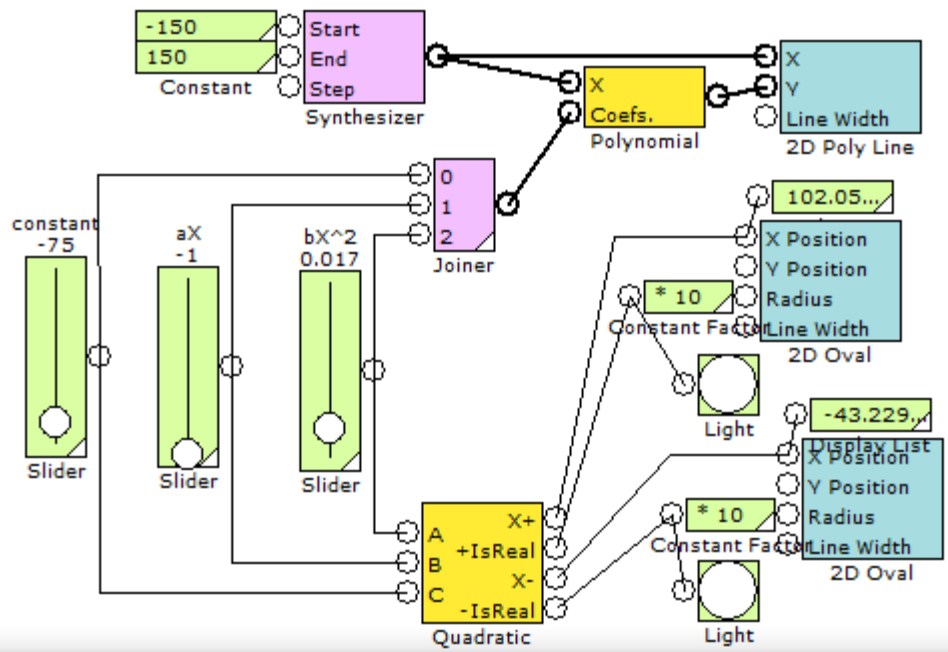
# QR Generator

The QR Generator component creates a QR graphic in the 2D View. 2D – column 3



# Quadratic

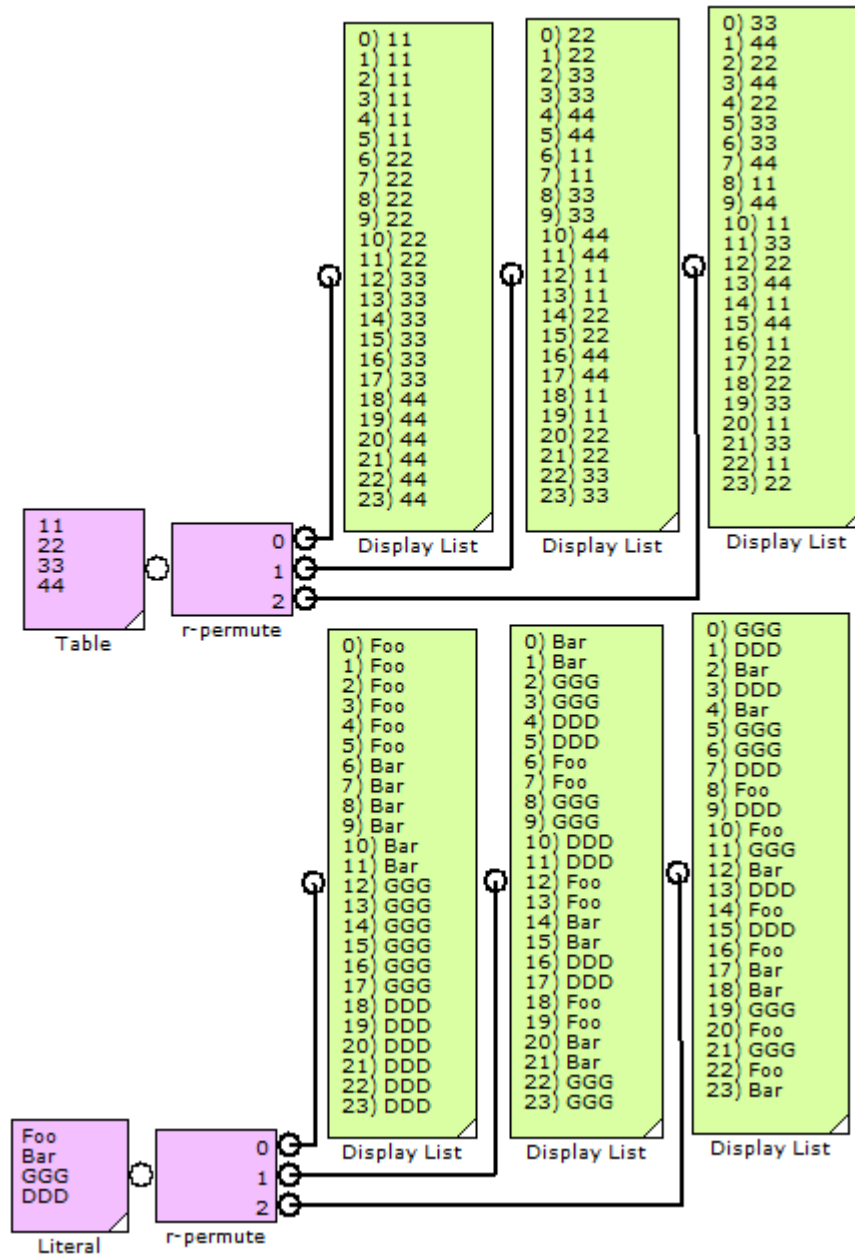
The Quadratic component factors a second degree polynomial. Input can be scalar or an array.  
 Functions - column 1



# r-permute

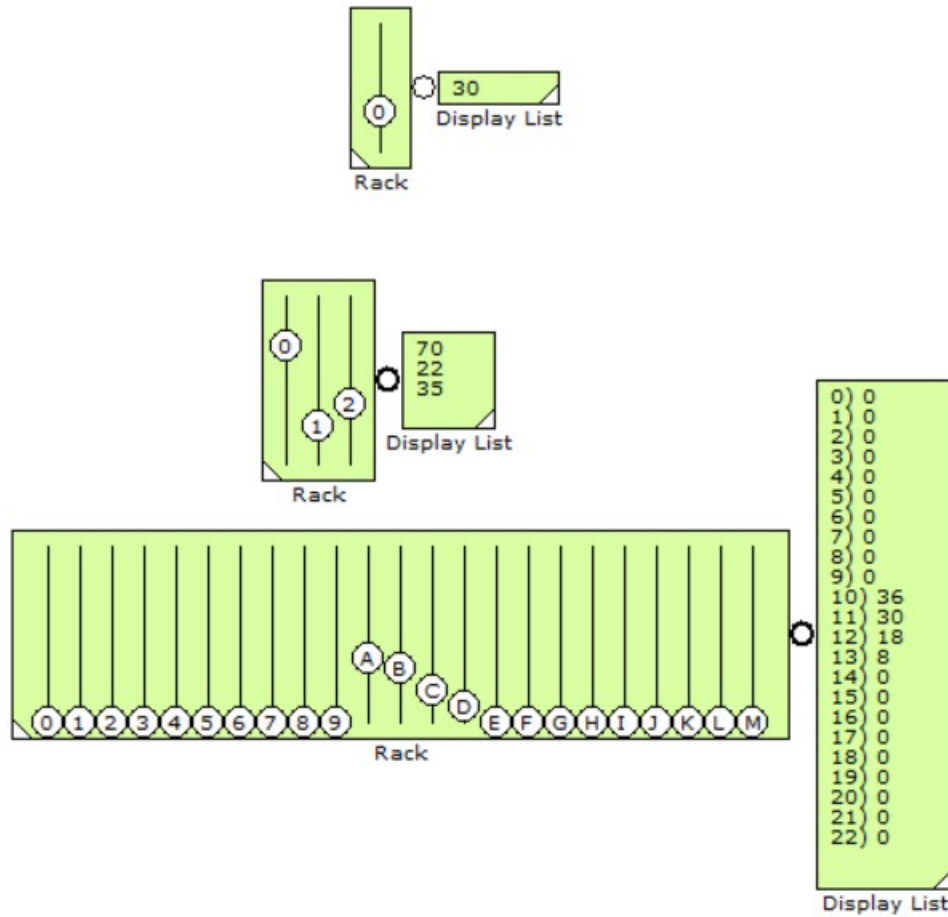
The r-permute component outputs a list of arrangements of a number of the input values.  
 - column 5

Arrays



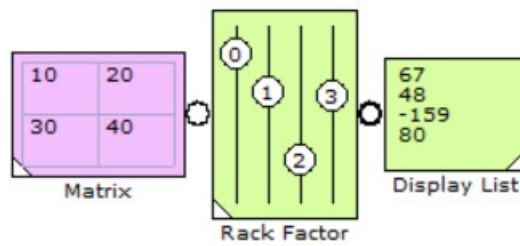
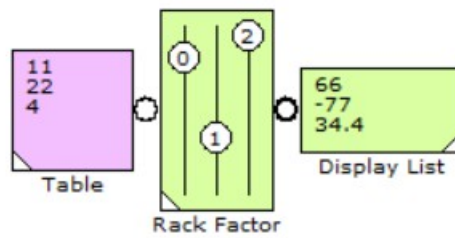
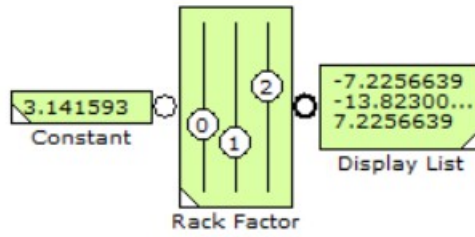
# Rack

The Rack component outputs a bunch of constant numbers based on its dial positions. Controls - column 1



# Rack Factor

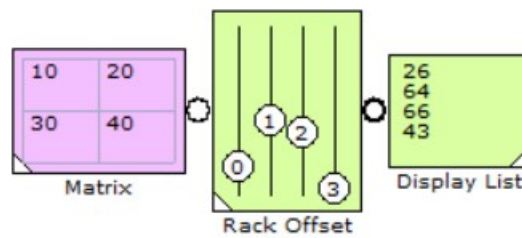
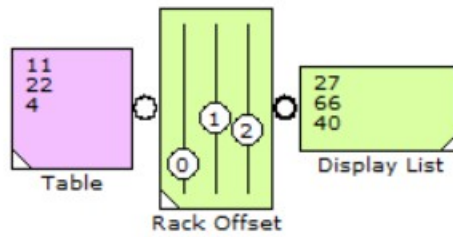
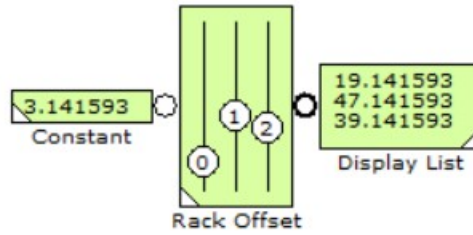
The Rack Factor component multiplies the inputs based on its dial positions. Controls - column 2



# Rack Offset

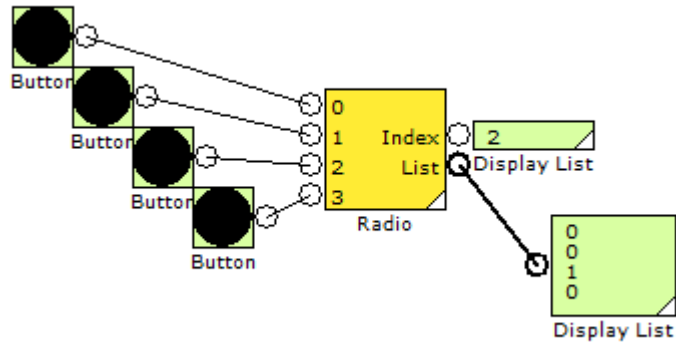
The Rack Offset component adds constants to the inputs based on its dial positions.

Controls - column 3



# Radio

The Radio component is like the function of a group of radio buttons. It outputs the number of the last input that went high. Functions - column 3





## 2D Radio Group

The 2D Radio Group is used to choose one of several buttons.

The X, Y and Size inputs control the position and font size of the text.

The Offsets control the position of the text relative to the buttons.

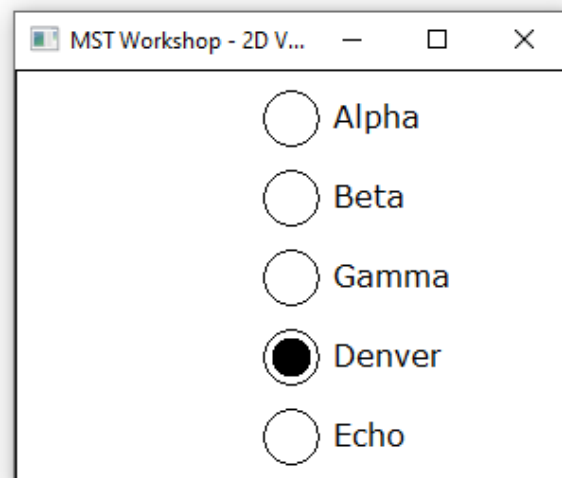
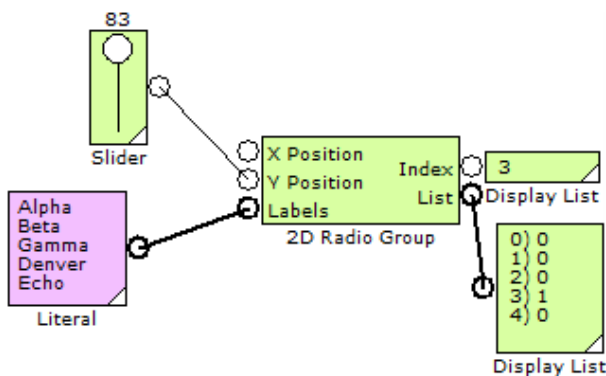
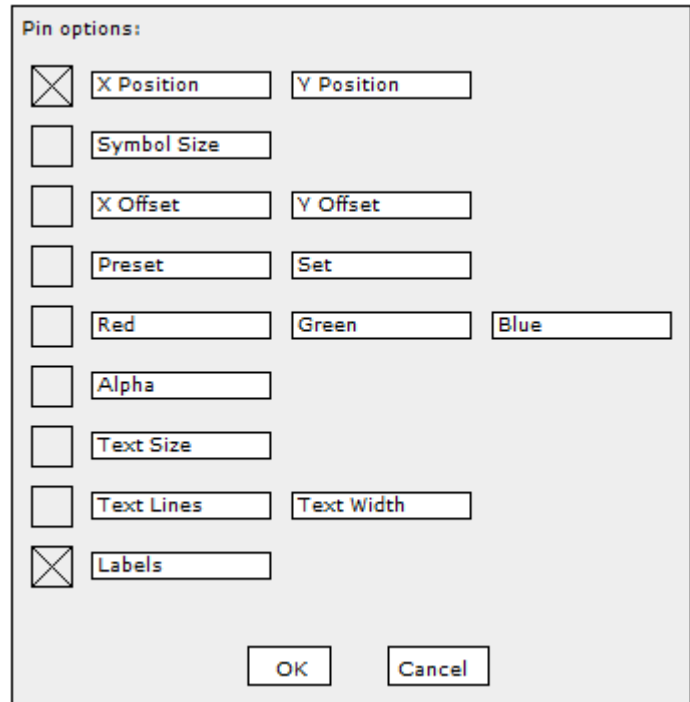
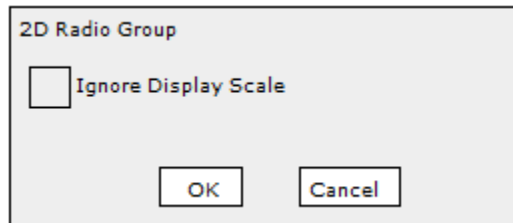
The Preset inputs allow external setting of the selection.

The Color input sets the color of the buttons and text.

The Alpha input fades the buttons and text.

The Index output indicates which item is chosen.

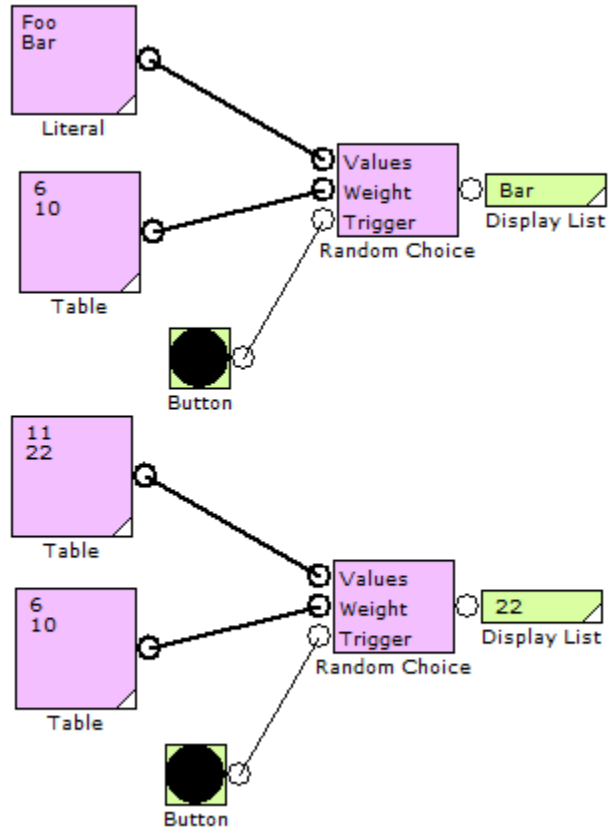
The List output provides an array with a one at the selected position. Controls - column 2



# Random Choice

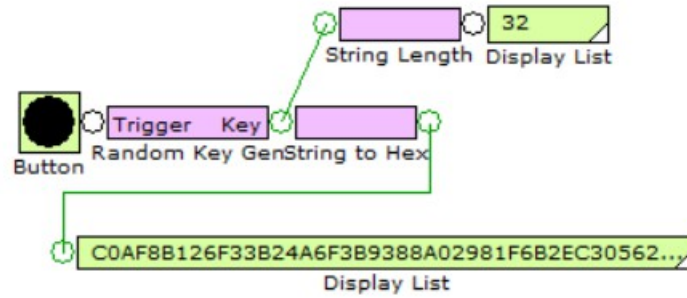
The Random Choice component randomly chooses one of the values at Values input. If the Weights input is used the choice is based on the weights at that input otherwise the choices are evenly weighted.

Arrays - column 3



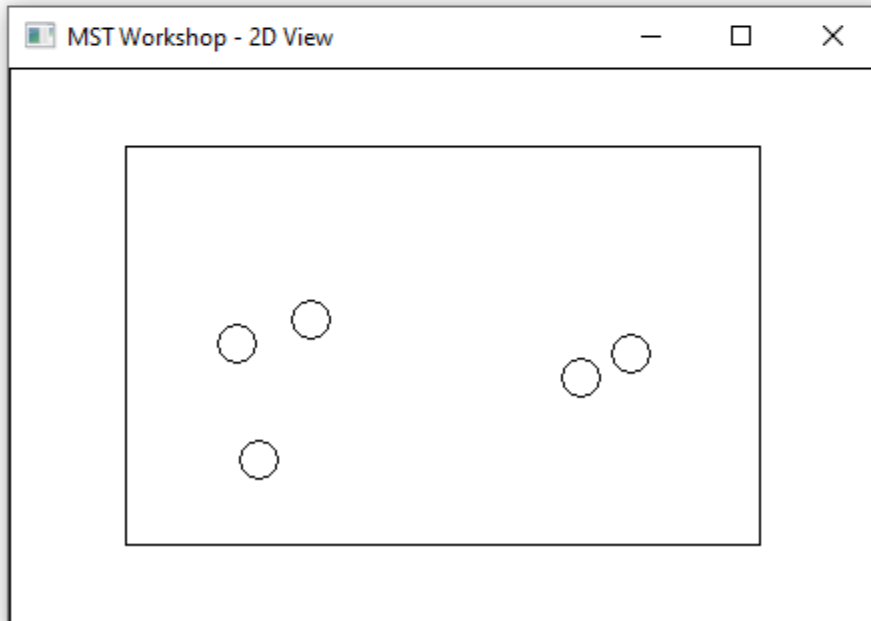
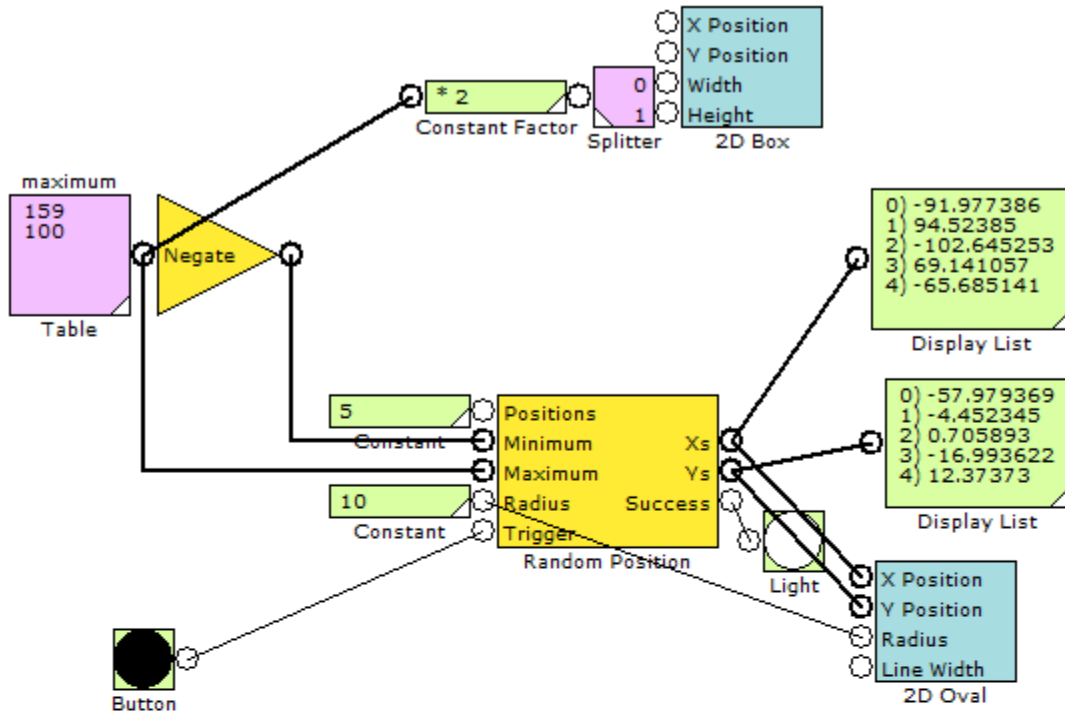
# Random Key Gen

The Random Key Gen component creates a key from random numbers. Arrays – column 3



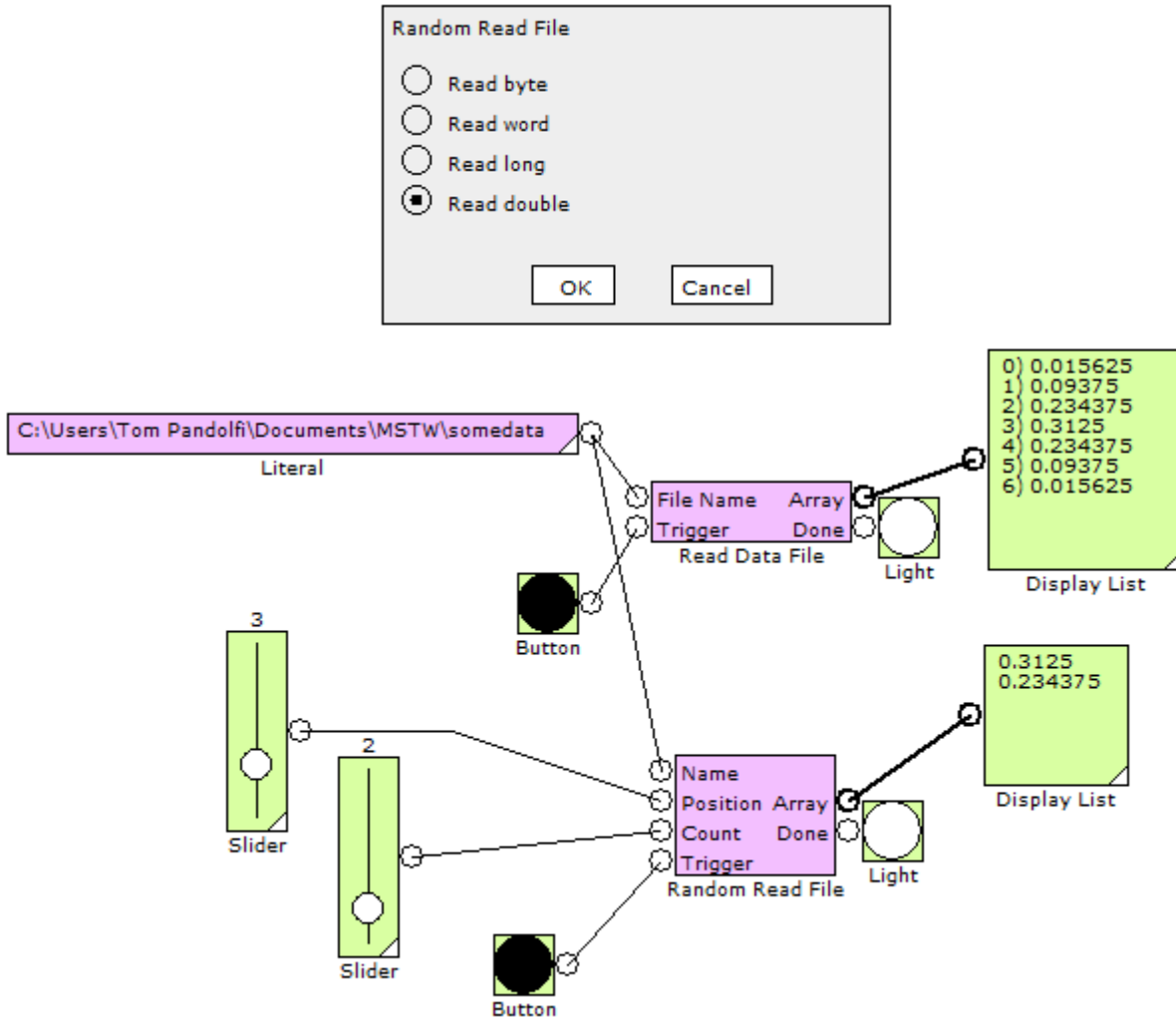
# Random Position

The Random Position component outputs random positions between between the minimum and maximum when the trigger goes from between false to true. The positions will be separated using the Radius input. Set the number of dimensions in the menu. Functions - column 6



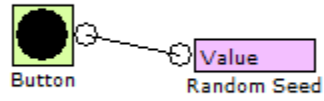
# Random Read File

The Random Read File component seeks to the specified position and reads the specified count of data type. The data type can be bytes, words, longs, or doubles, depending on what you specify in the options. Arrays - column 7



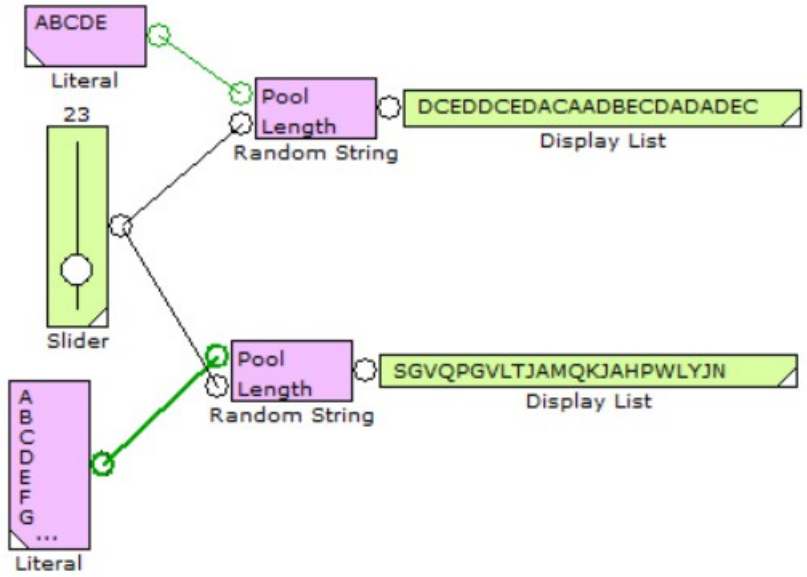
## Random Seed

The Seed component sets a new starting point for the random generator. If no input is connected the current time is used to set the random sequence. Arrays - column 3



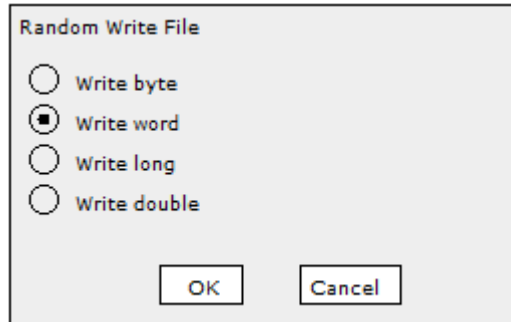
# Random String

The Random String component generates a random string from characters in the input Pool.  
Arrays – column 5



## Random Write File

The Random Write File component seeks to the specified position and writes the array in the specified format. The array data type can be bytes, words, longs, or doubles, depending on what you specify in the options. You can set an option to append the data to an existing file. Arrays - column 7

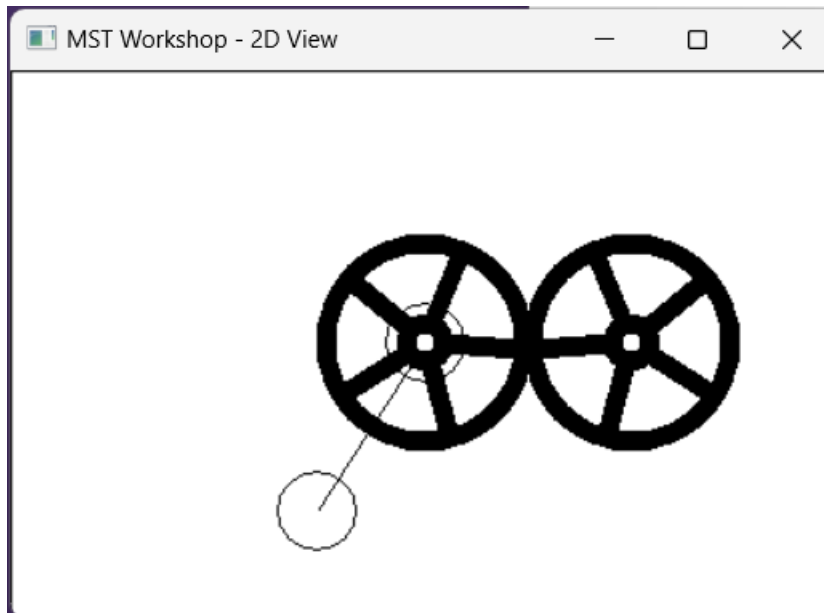
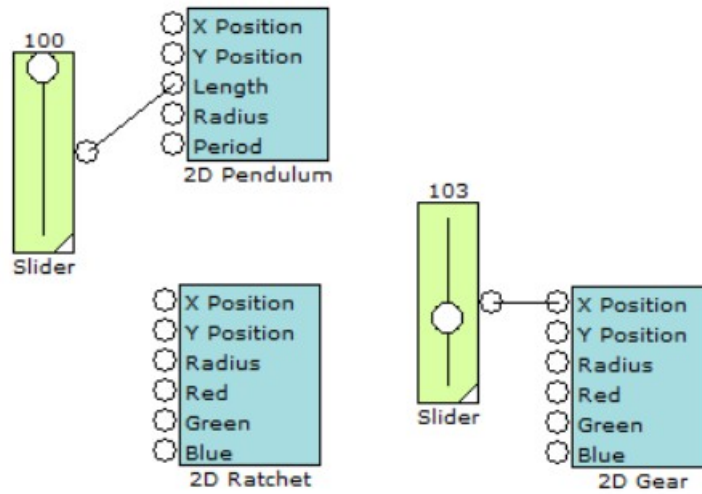




## 2D Ratchet

The 2D Ratchet component is like a gear that transmits rotation only in one direction. You can attach a driver to the shaft and drive something from the edge, or vice versa. Options indicate which direction is transmitted from hub to edge or vice versa.

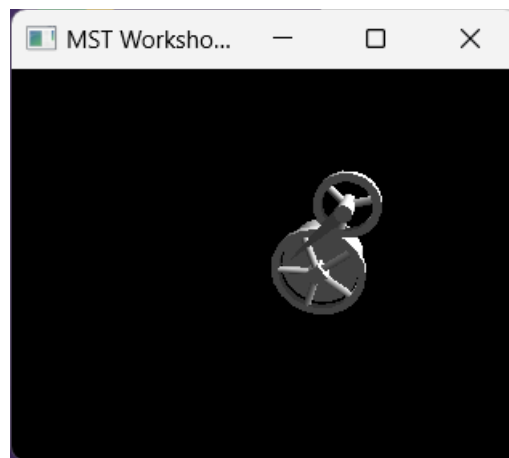
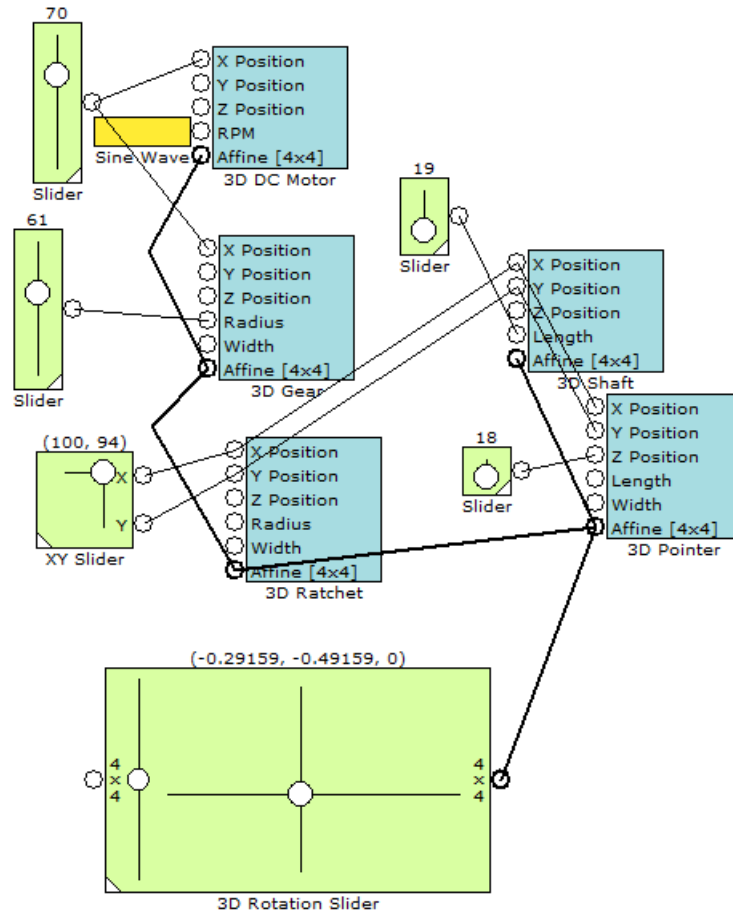
2D – column 3



# 3D Ratchet

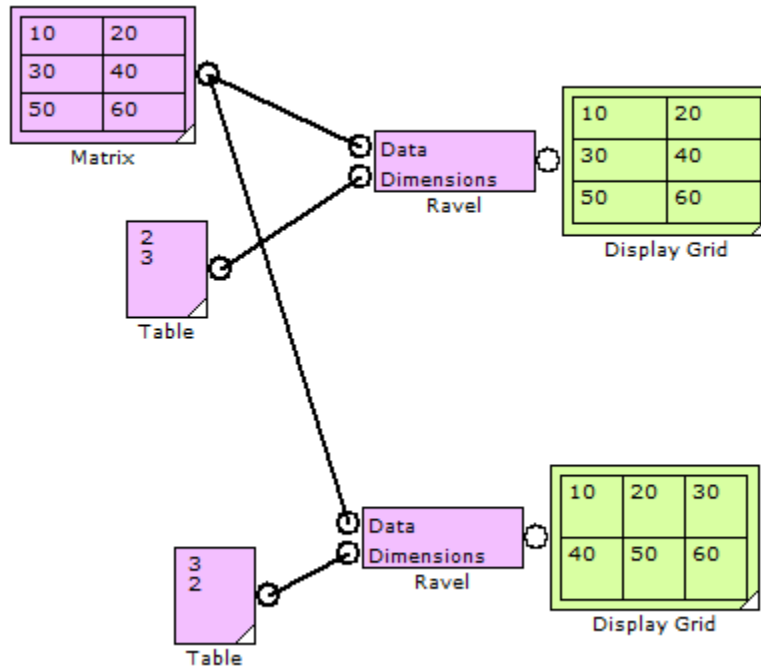
The 3D Ratchet component is like a gear that transmits rotation only in one direction. You can attach a driver to the shaft and drive something from the edge, or vice versa. Options indicate which direction is transmitted from hub to edge or vice versa.

3D – column 4



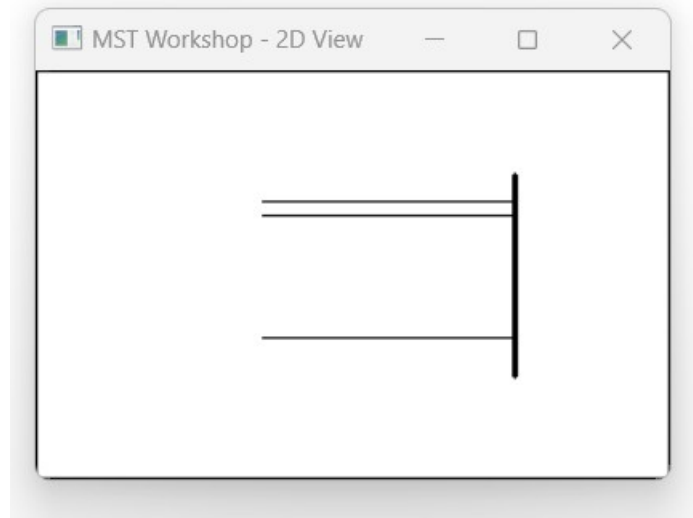
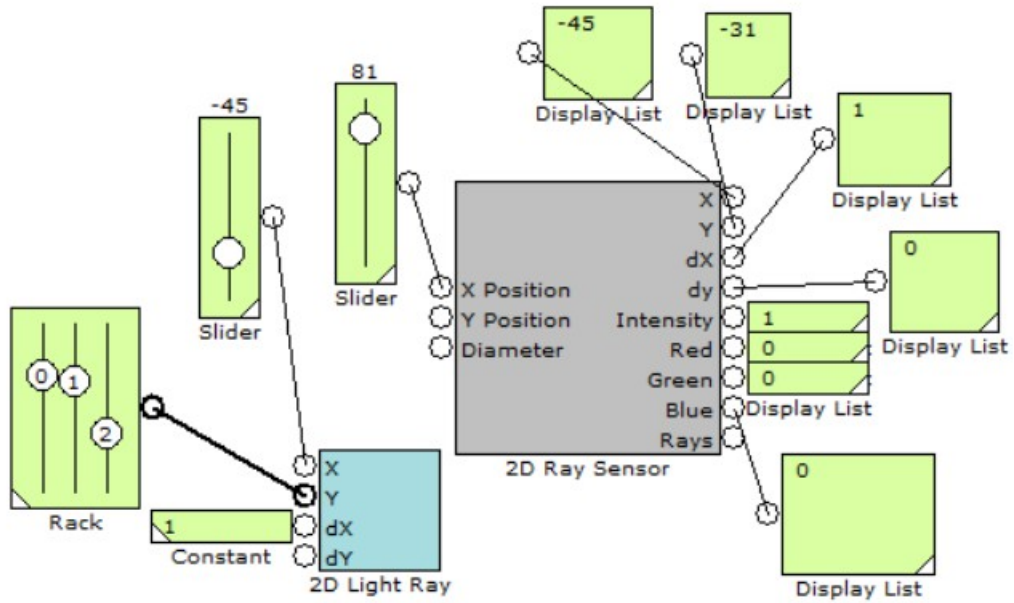
# Ravel

The Ravel component sets the dimensions for the input data. Use it to create a matrix from a vector or to change the dimensions of the input data. Arrays - column 2



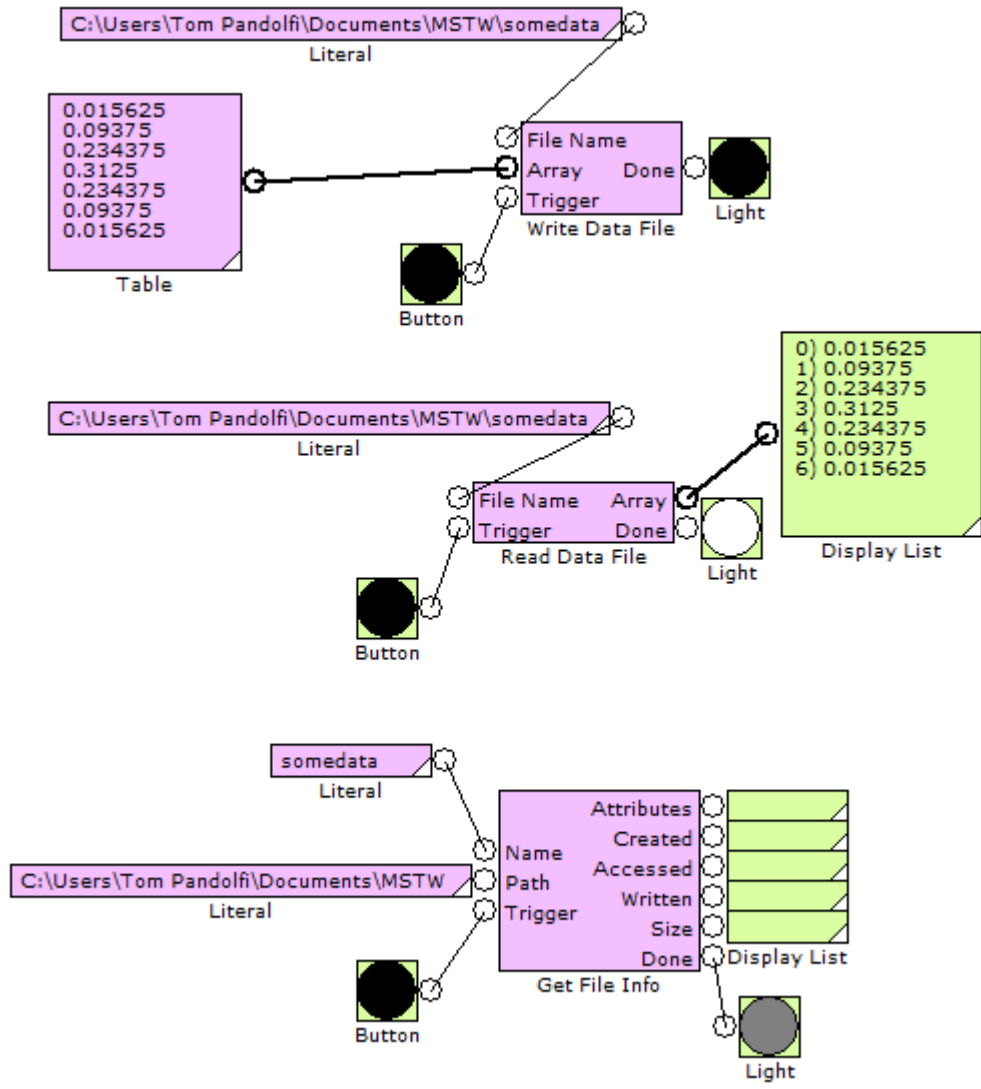
## 2D Ray Sensor

The 2D Ray Sensor component collects rays of light in the 2D view window. The position and diameter control the position and size of the sensor. The outputs give information about where the ray is coming from. 2D – column 3



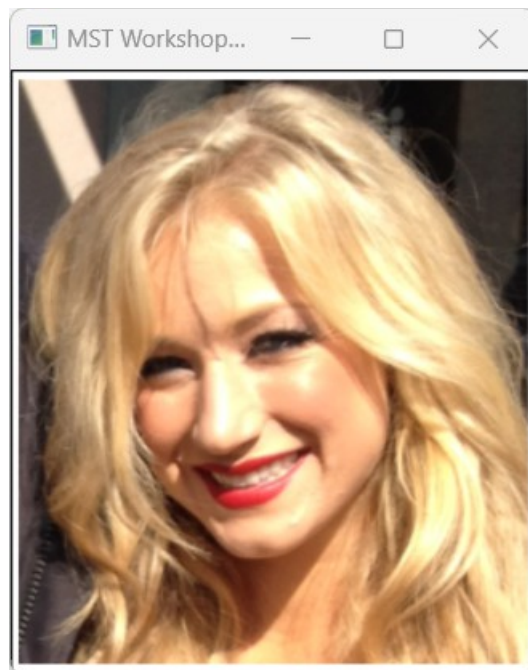
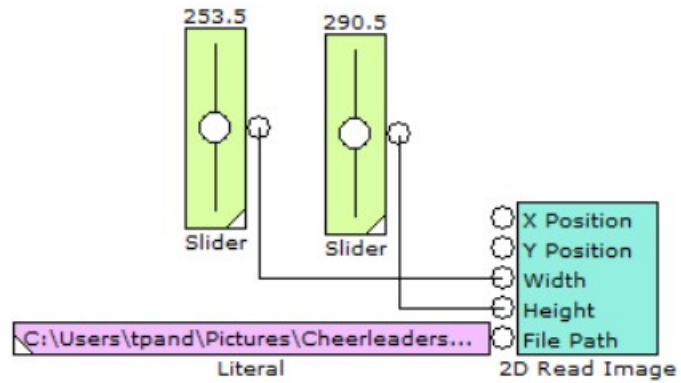
# Read Data File

The Read Data File component reads a binary file. The file can be bytes, words, longs, or doubles, depending on what you specify in the options.      Arrays - column 7



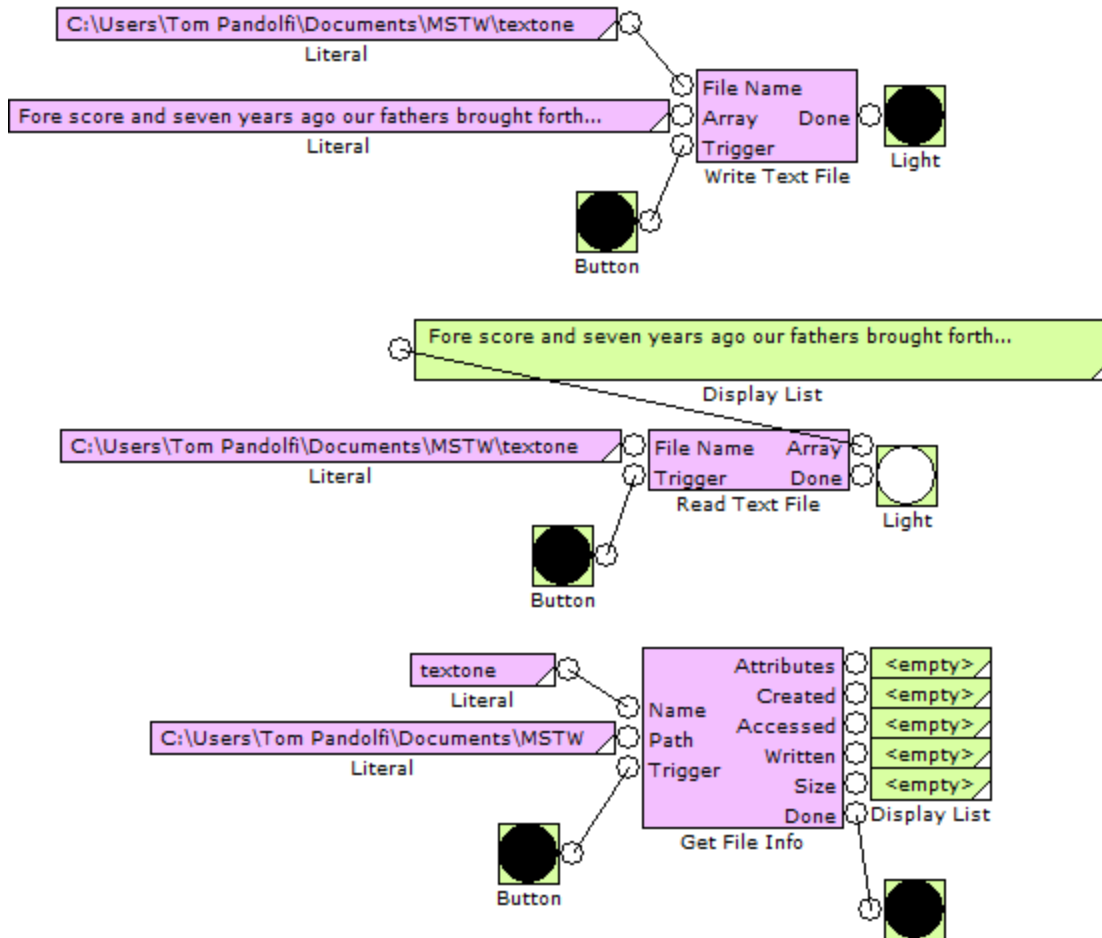
## 2D Read Image

The 2D Read Image component reads an image and draws it in the 2D window. 2D – column 1



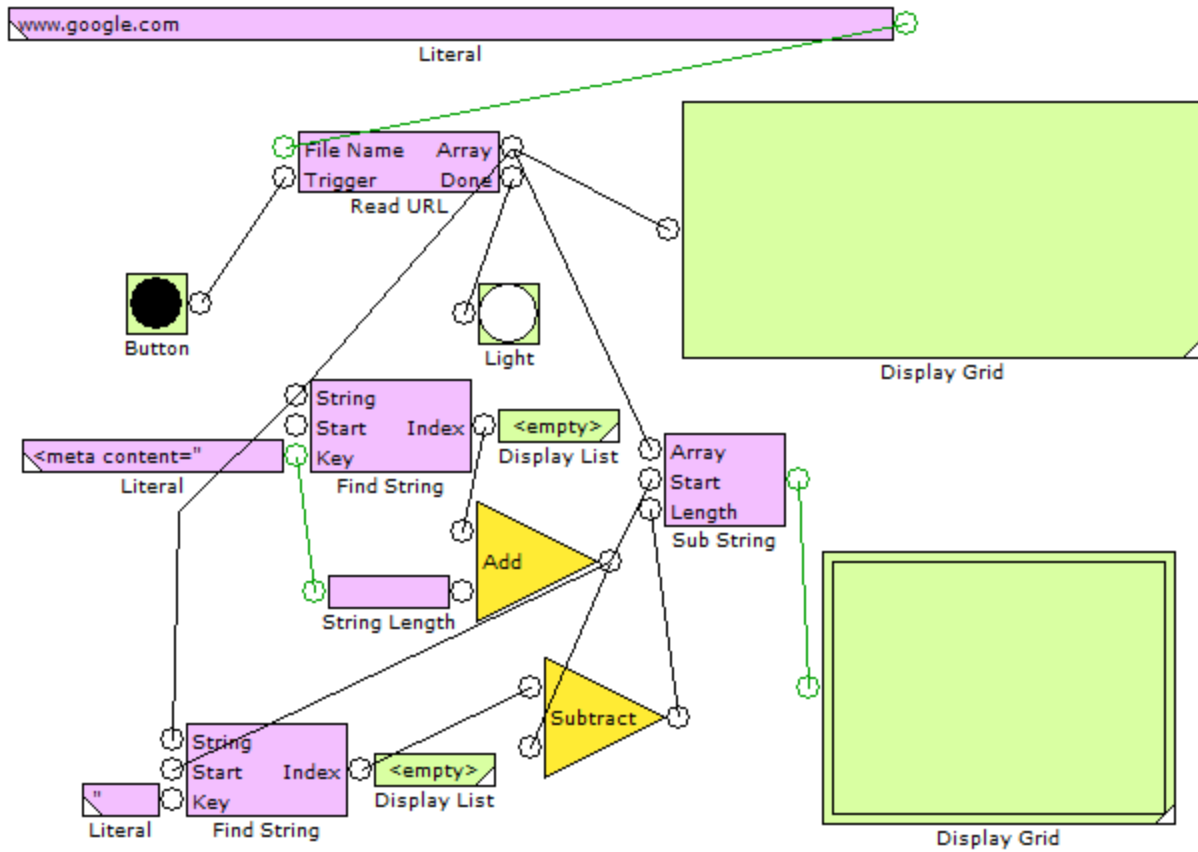
# Read Text File

The Read Text File component reads a text file. Arrays - column 7



# Read URL

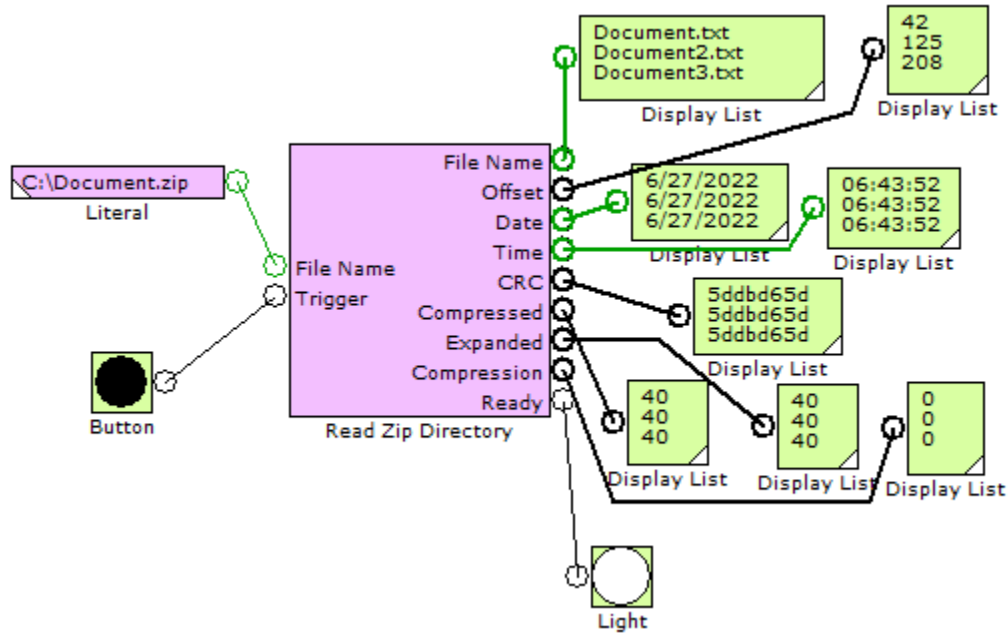
The Read URL component reads a file on the internet given the URL of the file. Arrays - column 7





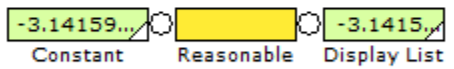
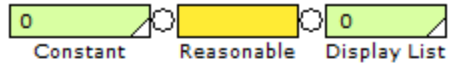
# Read Zip Directory

The Read Zip Directory component reads a Zip file and outputs the inner file names, positions and sizes. Use the position and size to read any of the inner files. Arrays – column 7



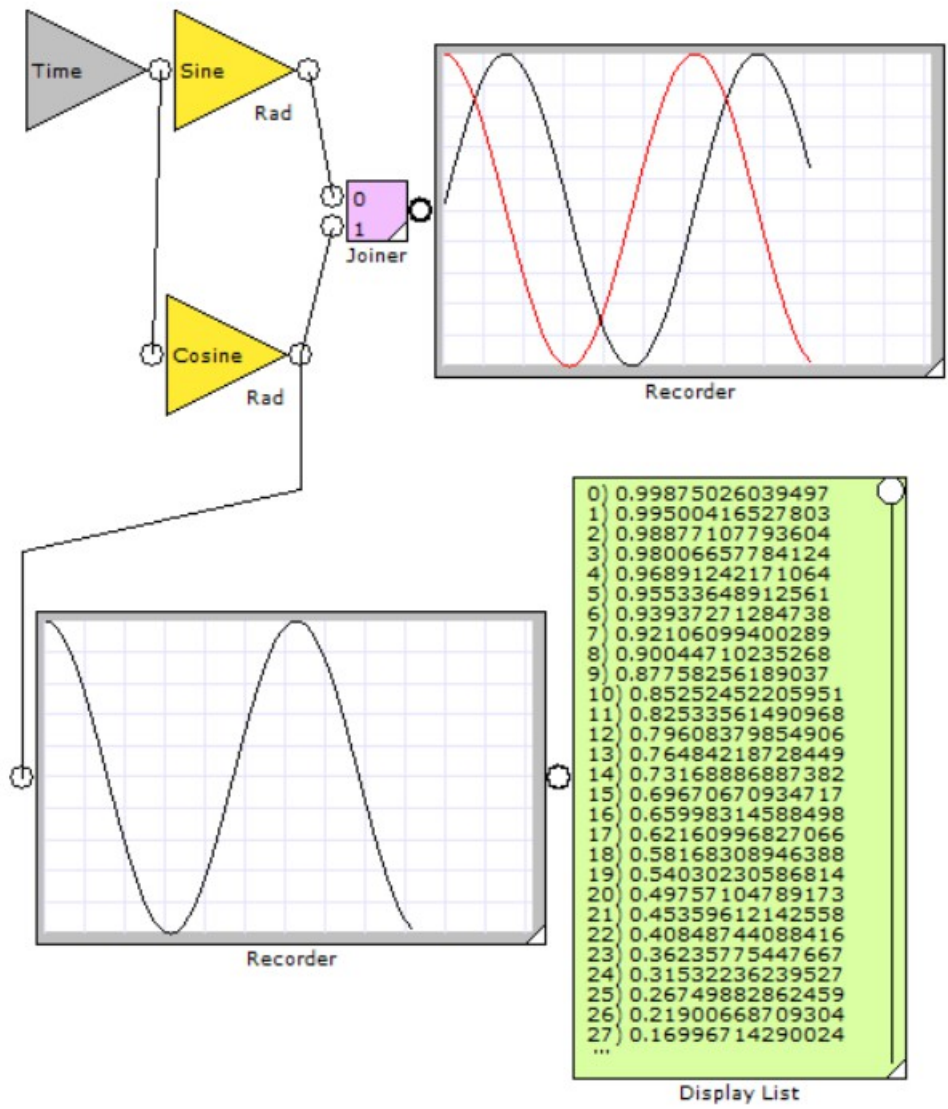
# Reasonable

The Reasonable component converts values to a reasonable value. Inputs can be scalar or arrays.  
Functions - column 2



# Recorder

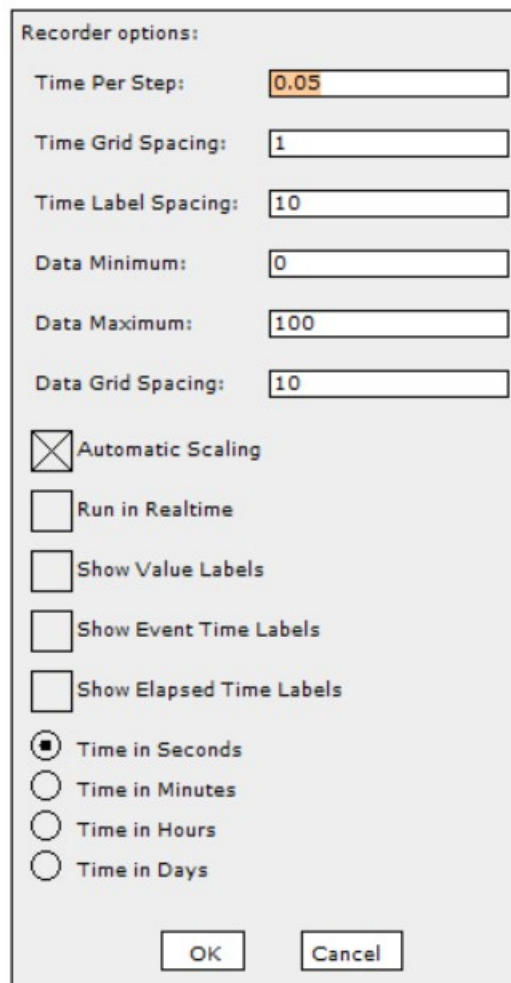
The Recorder plots the value of the inputs over time. Hit the Start button to run this recorder. You can access the values shown on the recorder by enabling the Outputs. Controls - column 4



You can set the time steps using the Time Per Step input and the radio buttons to select if this time is seconds, minutes, hours or days. As shown below the setting is 0.05 seconds per step. The Time Grid Spacing indicates how many steps before a grid line is drawn. The Time Label Spacing is how many steps before a bold line and label are drawn.

For the vertical grid, you can control the minimum and maximum values allowed in the plot or select Automatic Scaling.

For the horizontal grid, three options are available: Event Time, Elapsed Time, or both. If you select Event Time vertical lines will be drawn and labeled at the bottom to indicate the time of the event starting at the time the Recorder was started. If you select Elapsed Time vertical lines will be drawn and labeled at the top indicating the time elapsed since the Recorder was started. If you select both the labels will show as normally but the vertical lines will be drawn to the mid point of the chart.



Recorder options:

Time Per Step:

Time Grid Spacing:

Time Label Spacing:

Data Minimum:

Data Maximum:

Data Grid Spacing:

Automatic Scaling

Run in Realtime

Show Value Labels

Show Event Time Labels

Show Elapsed Time Labels

Time in Seconds

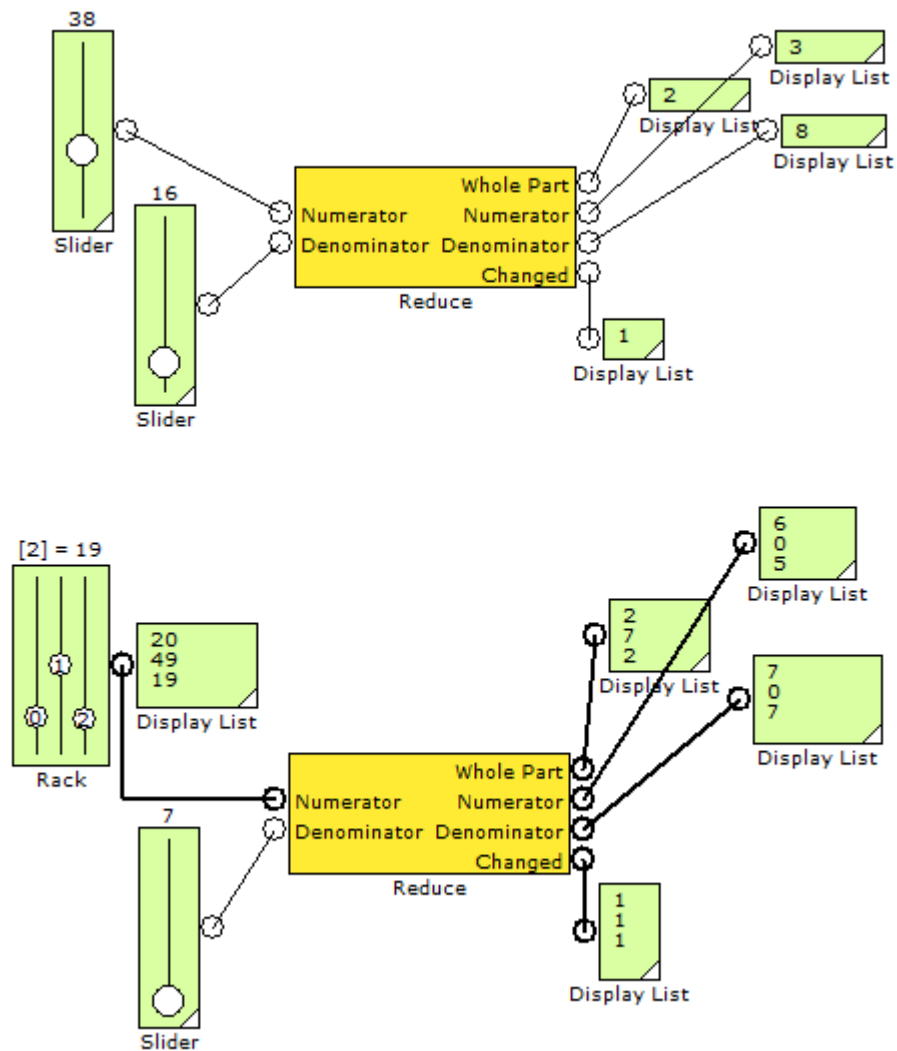
Time in Minutes

Time in Hours

Time in Days

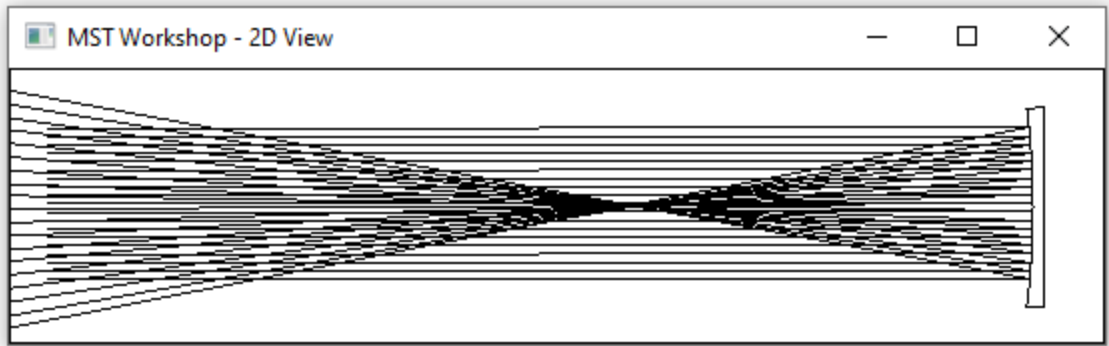
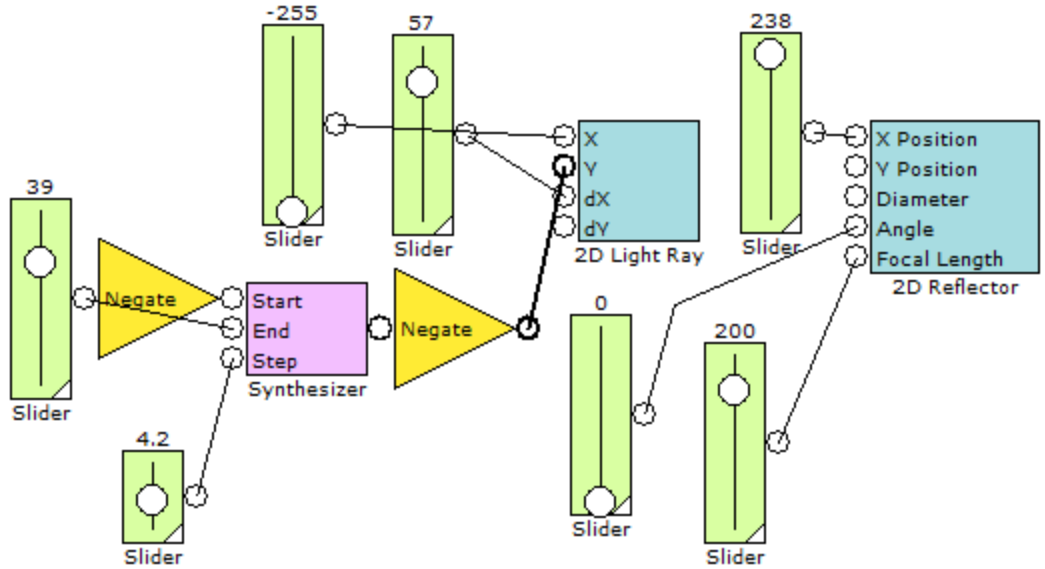
# Reduce

The Reduce component reduces the input fraction to a proper fraction. Possibly with a whole part.  
 Functions - column 9



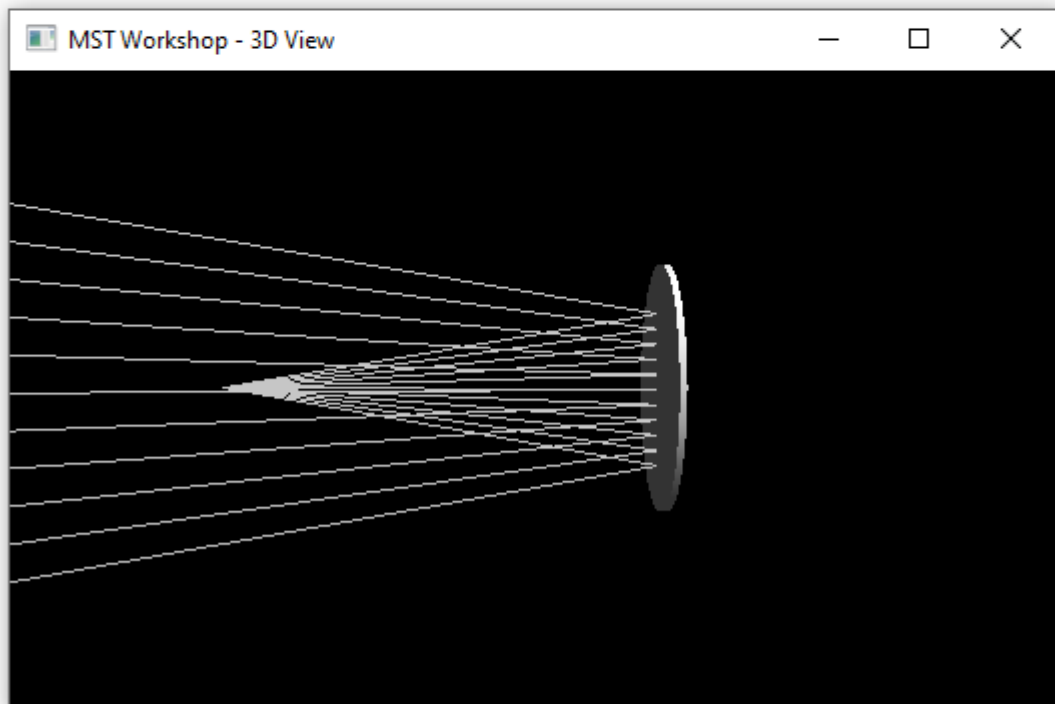
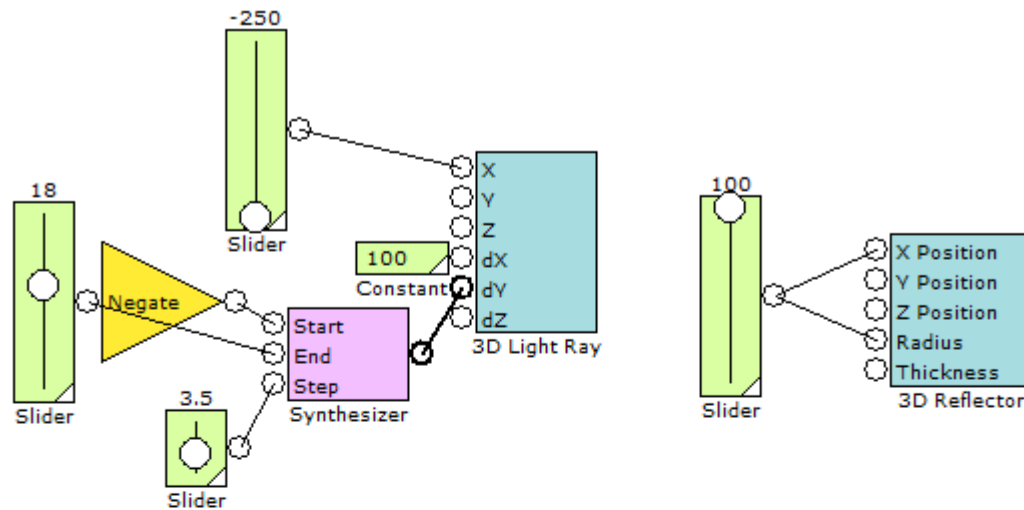
## 2D Reflector

The 2D Reflector component reflects light rays in the 2D view window. The settings determine the fraction of light that reflects and passes through. 2D - column 3



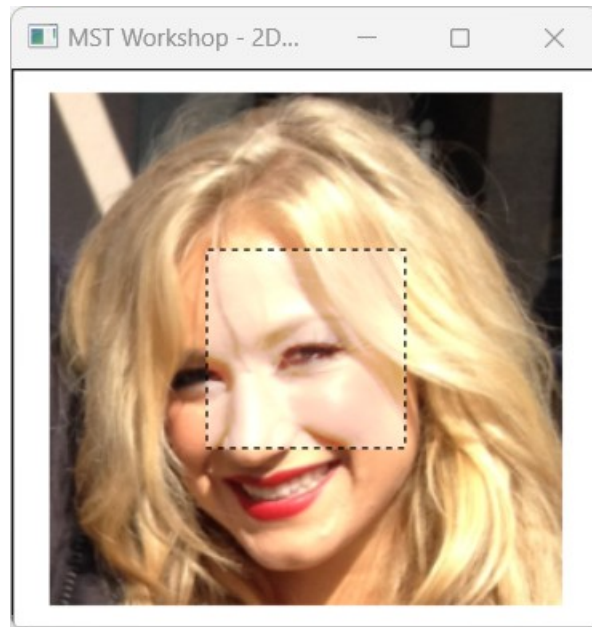
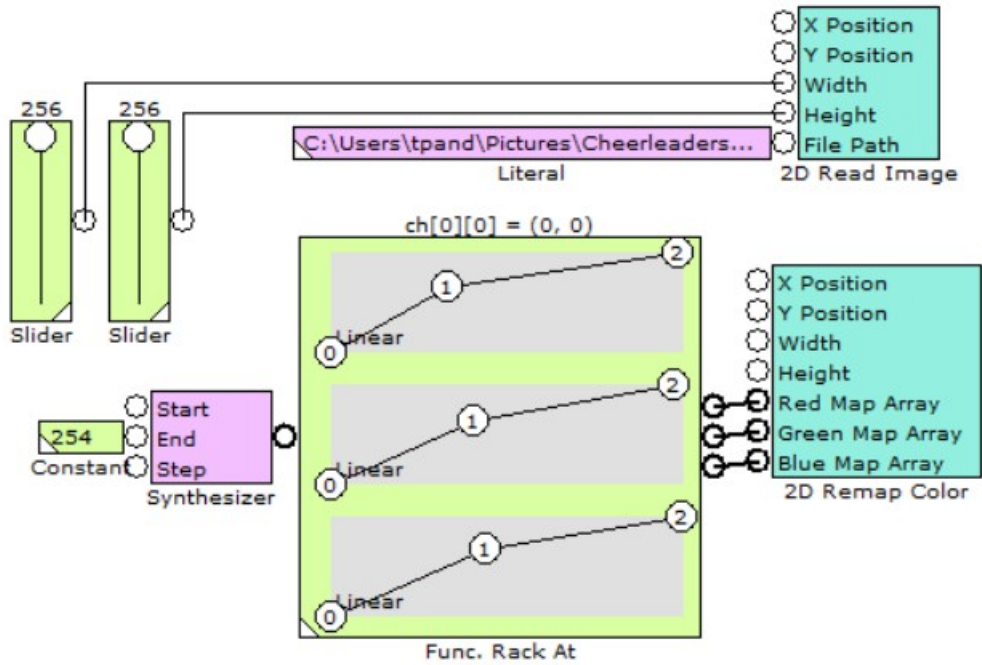
## 3D Reflector

The 3D Reflector component reflects light rays in the 3D view window. The settings determine the fraction of light that reflects and passes through. The Affine [4x4] array input modifies the size, position and orientation of the reflector. 3D - column 5



## 2D Remap Color

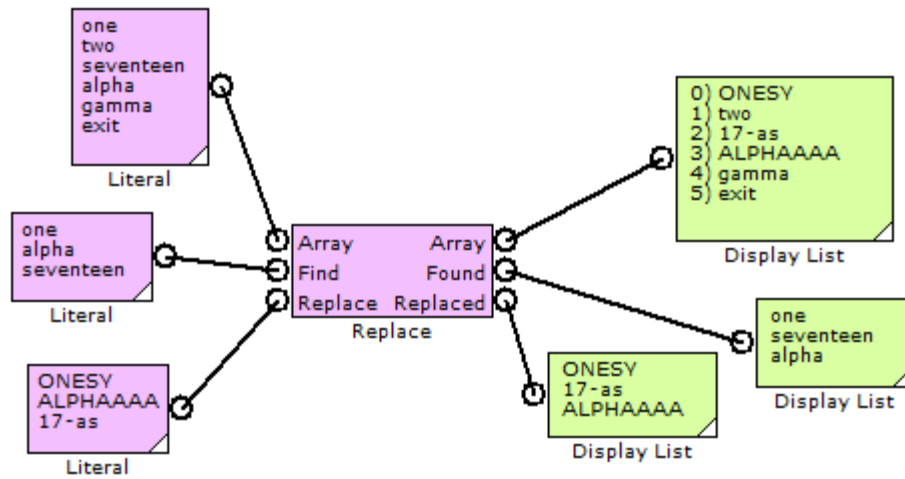
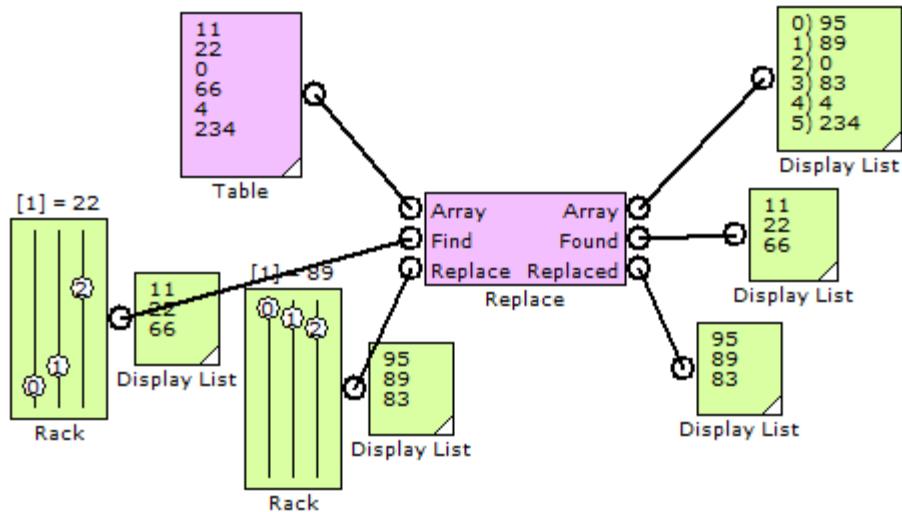
The 2D Remap Color uses the input look-up table to modify the color of each pixel. 2D – column 1





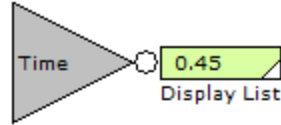
# Replace

The Replace component scans the input, when it finds the Match input value it replaces it with the Replace input string. Arrays - column 2



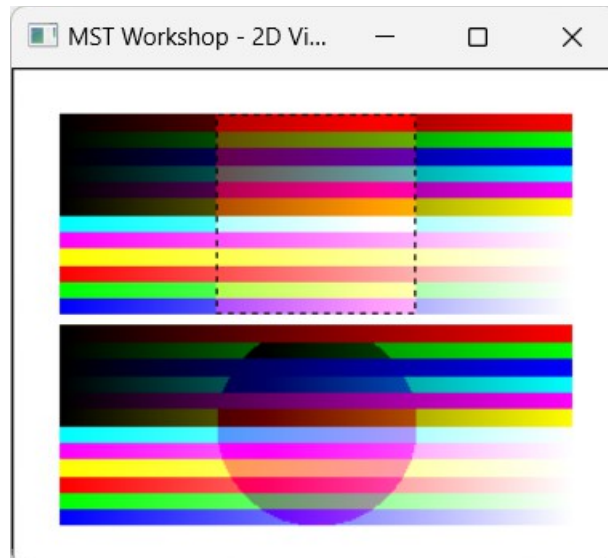
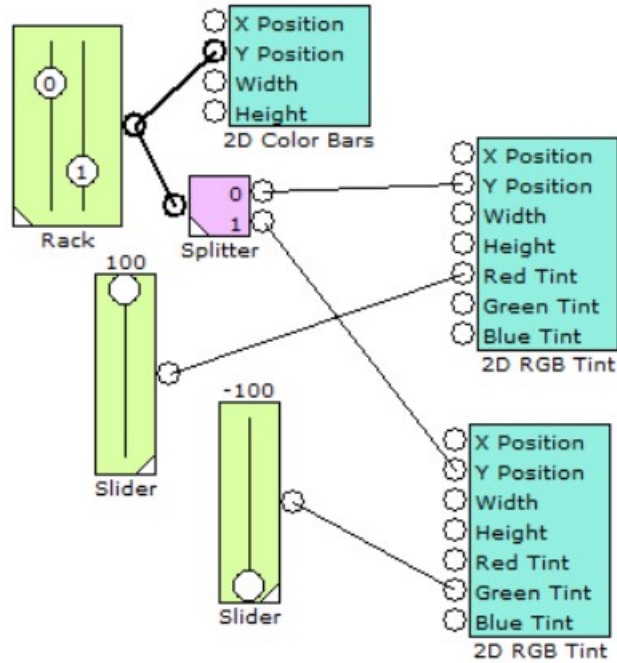
# Reset Clock

The Reset Clock component sets the solution clock to zero when the trigger goes from false to true.  
Controls - column 3



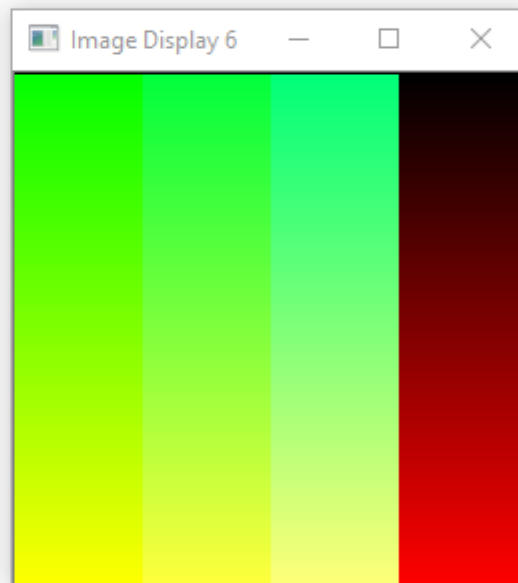
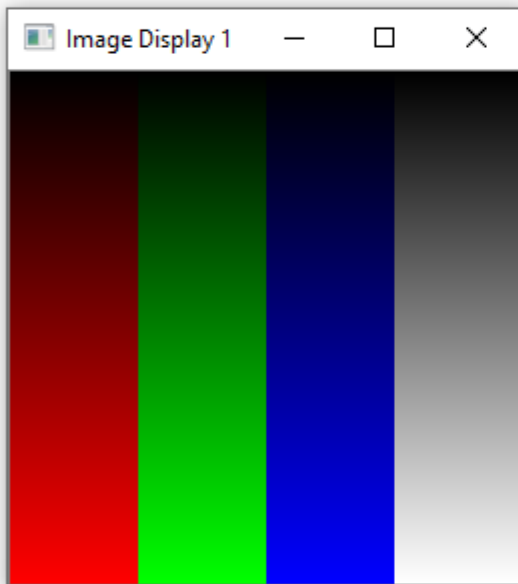
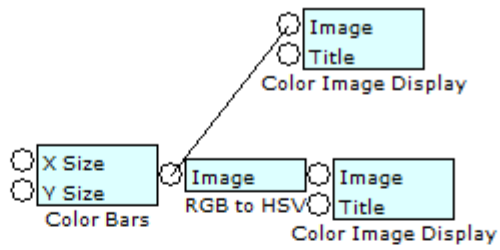
## 2D RGB Tint

The 2D RGB Tint lightens or darkens each of the Red Green and Blue components in the 2D drawing it covers. 2D – column 1



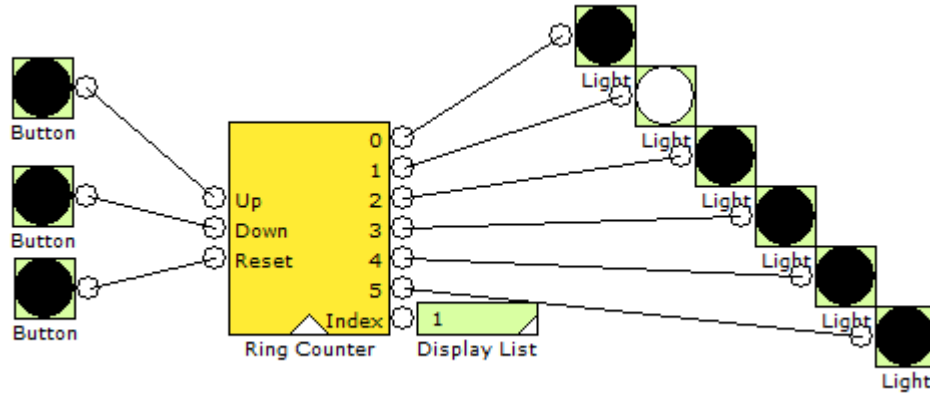
# RGB to HSV

The RGB to HSV component converts the image from RGB color to HSV color. Images - column 1



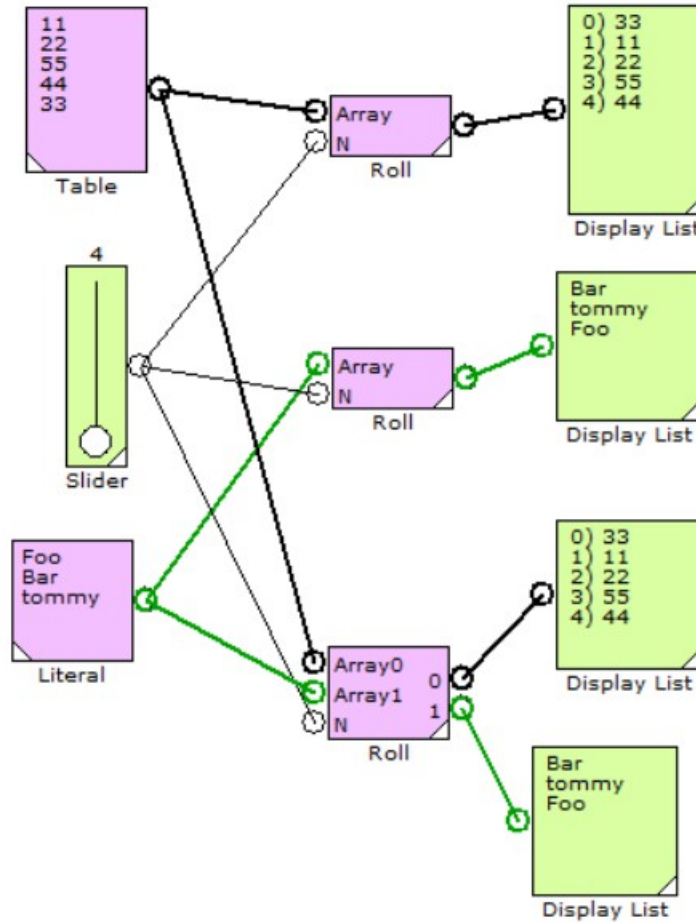
# Ring Counter

The Ring Counter component counts up or down and outputs one of n true values and the index of that value. Functions - column 8



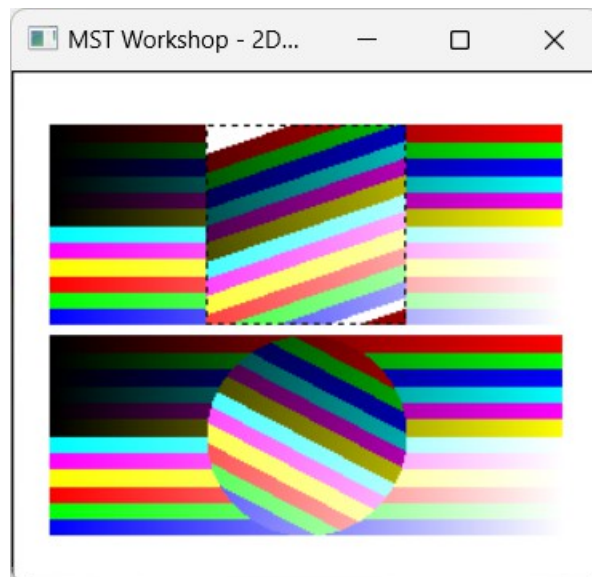
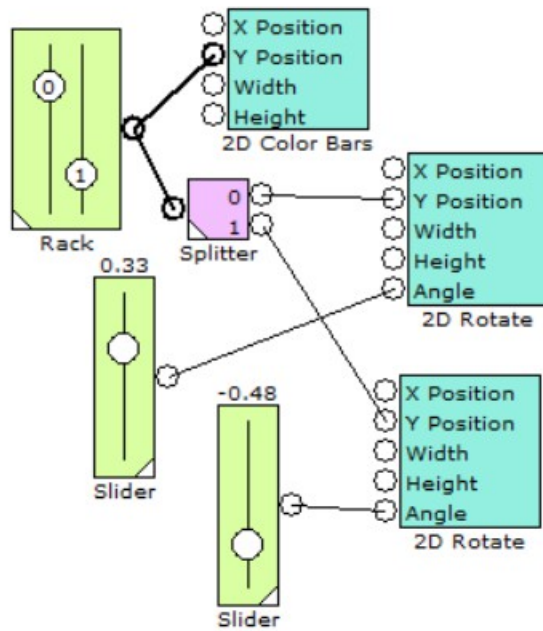
# Roll

The Roll component rotates the values by the Amount given. Negative numbers rotate backwards.  
Arrays - column 1



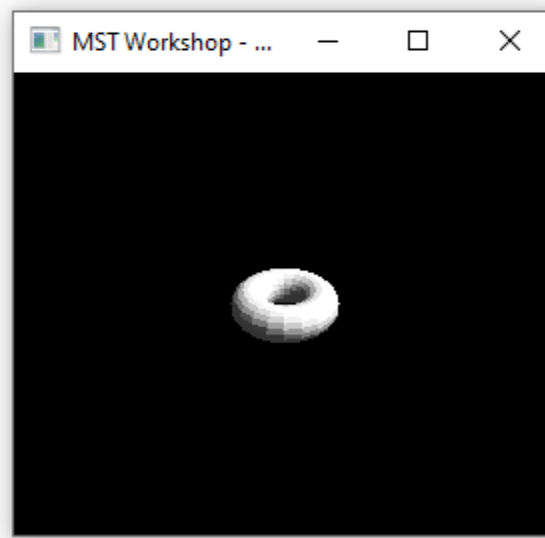
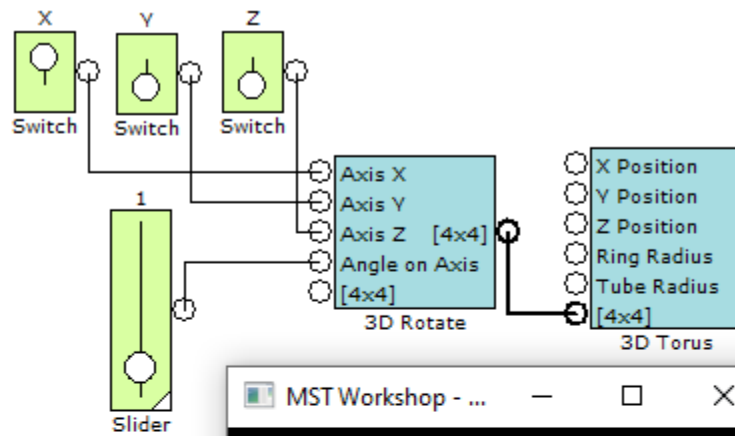
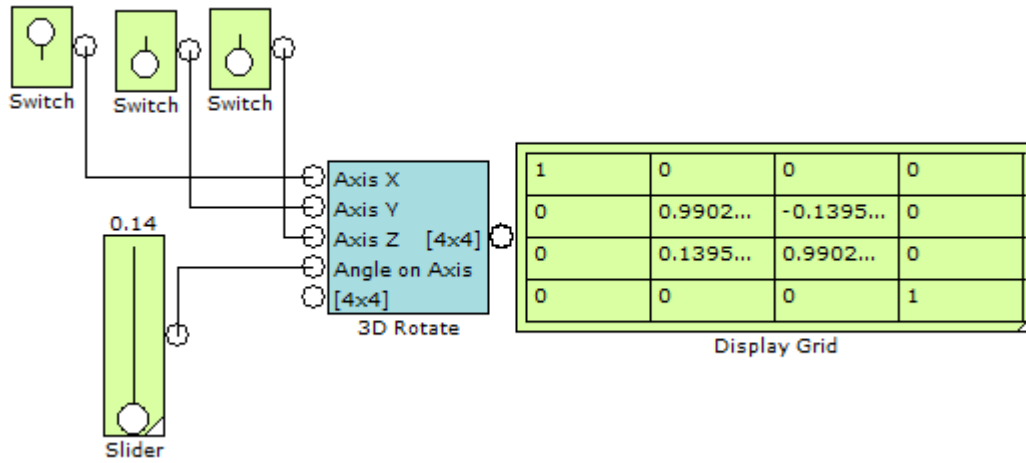
## 2D Rotate

The 2D Rotate rotates the image in the 2D Drawing it covers. 2D – column 1



### 3D Rotate

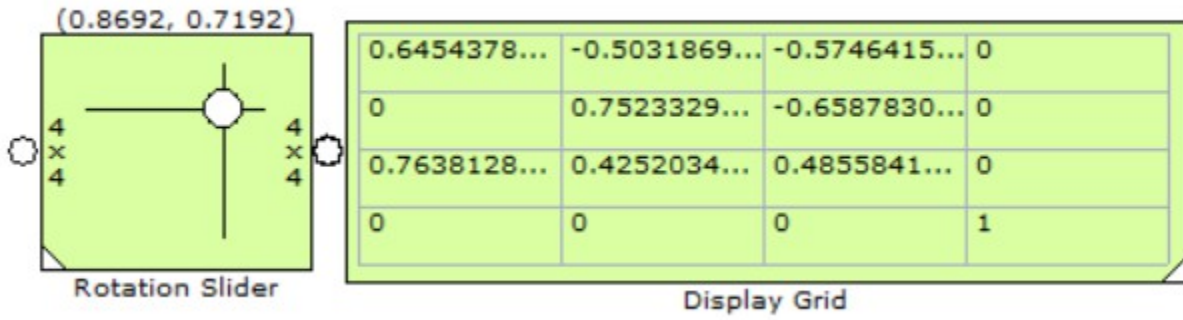
The 3D Rotate rotates a 4 by 4 matrix on the X, Y, and/or Z axis. 3D - column 3





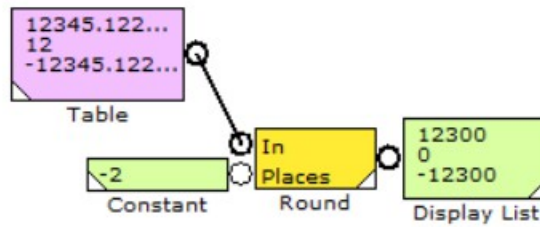
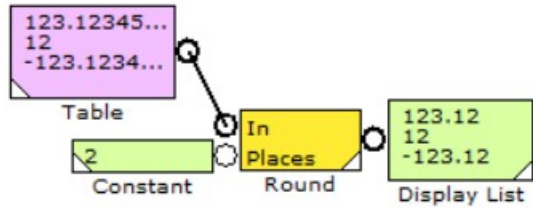
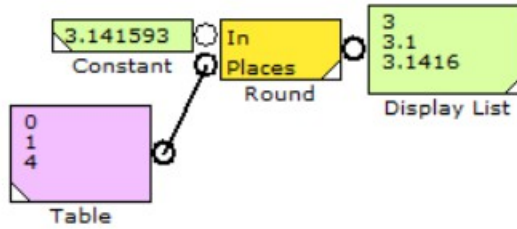
# Rotation Slider

The Rotation Slider component creates a 4x4 rotation matrix from the position of the control knob.  
Controls – column 2



# Round

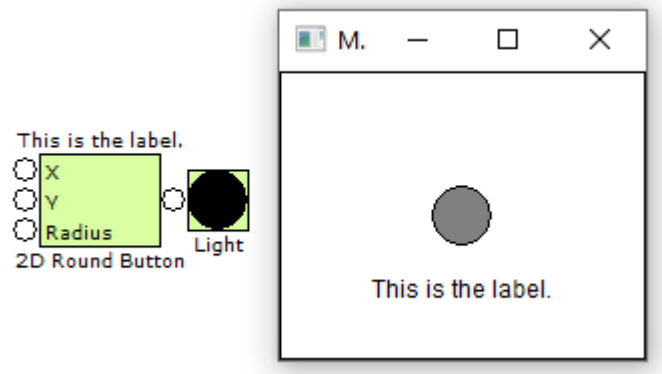
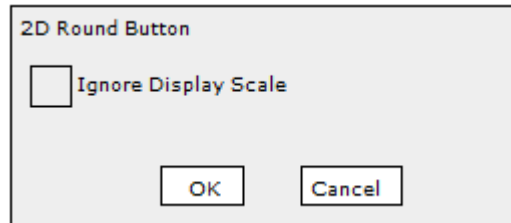
The Round component outputs a value with the specified decimal places. Inputs can be scalar or arrays.  
Functions - column 2



## 2D Round Button

The 2D Round Button is used to trigger an event by clicking on it.

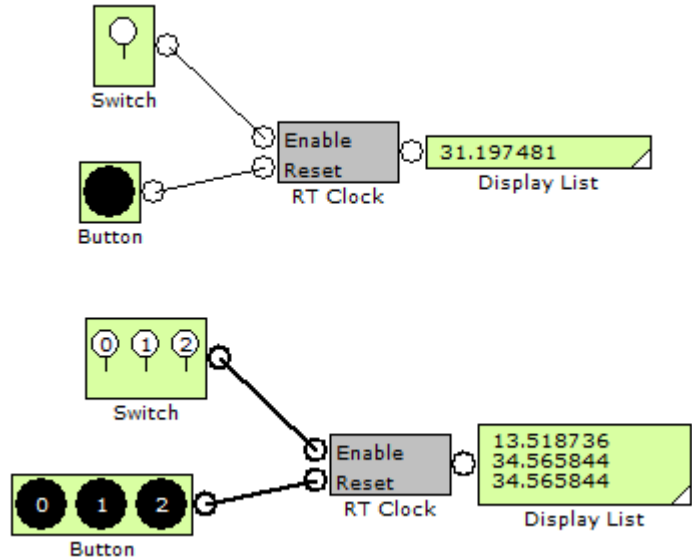
Controls - column 1



# RT Clock

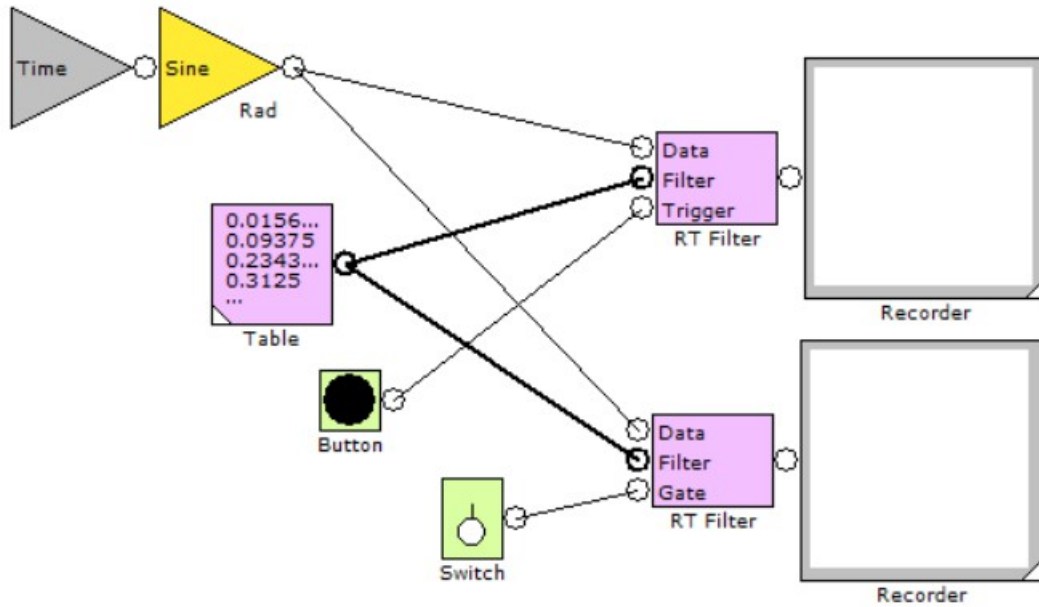
The RT Clock component lets you measure time using the computer's internal clock (not the workspace time). The output is in seconds with microsecond resolution. When the RT Clock is enabled it is running even though it's output is only updated when the workspace clock increments.

Functions - column 7



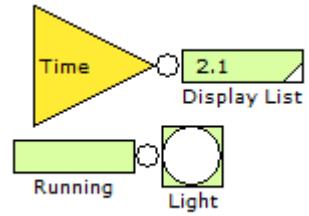
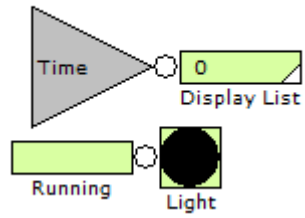
## RT Filter

The RT Filter component processes the data as it comes into the filter in realtime. It uses the input from the Filter array. The output is the filtered value. Arrays – column 2



# Running

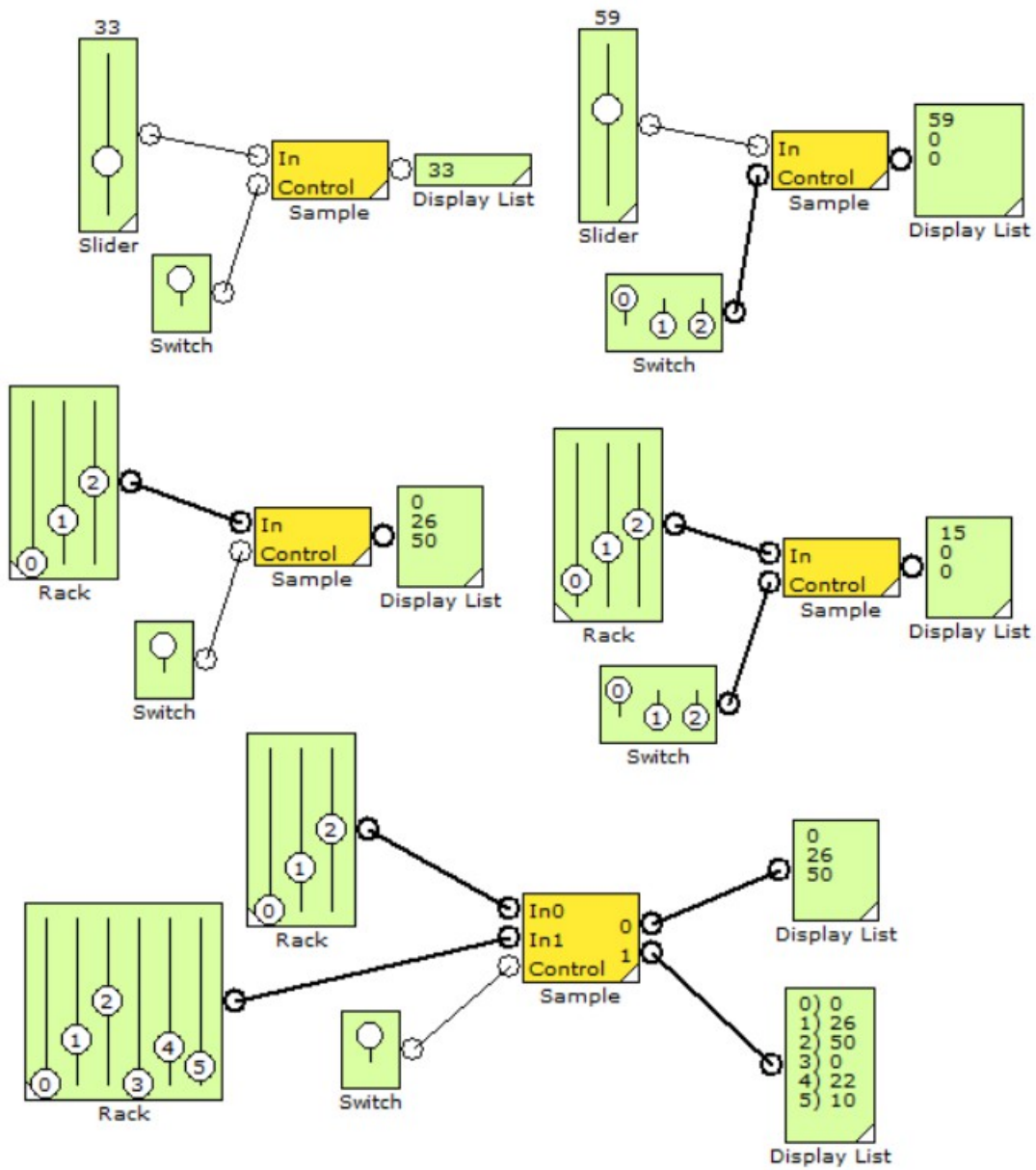
The Running component outputs a 1 when the clock is running or 0 when the clock is stopped.  
Controls - column 3



# Sample

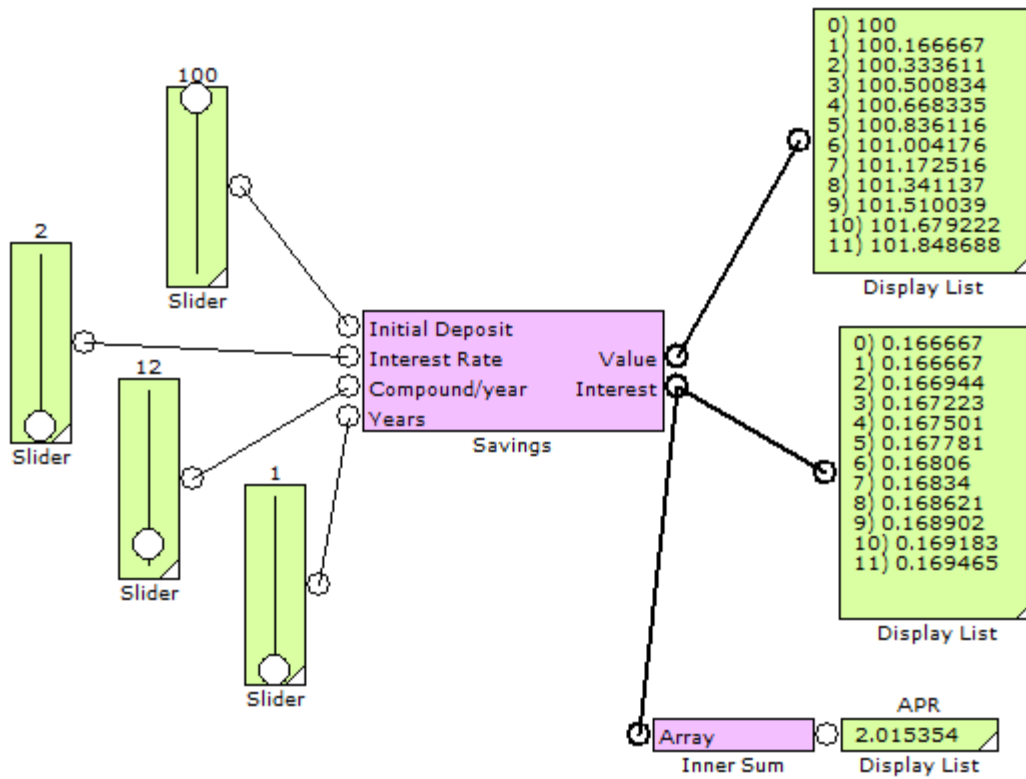
The Sample will send the input to the output when the control input is true.

Functions - column 2



# Savings

The Savings component calculate the interest on an investment. Arrays - column 5





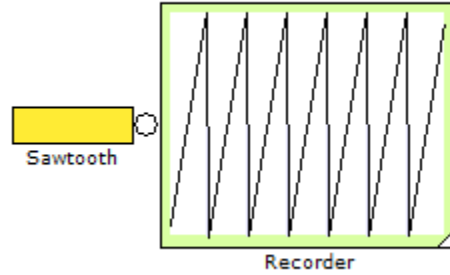
# Sawtooth

The Sawtooth component outputs a ramp that drops sharply when the clock is running. Functions - column 8

Sawtooth

Output Period:

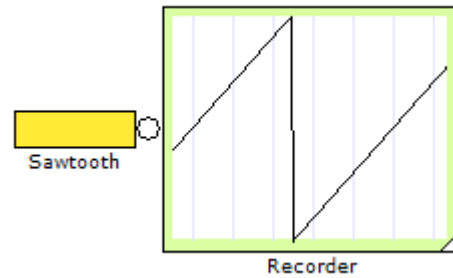
Output Maximum:



Sawtooth

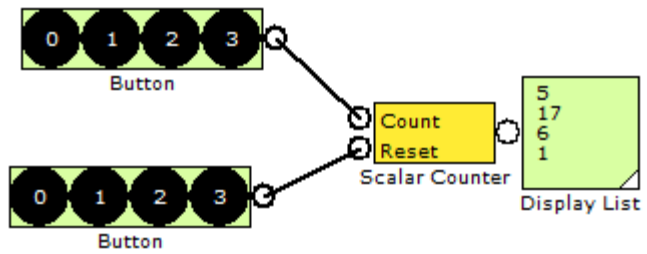
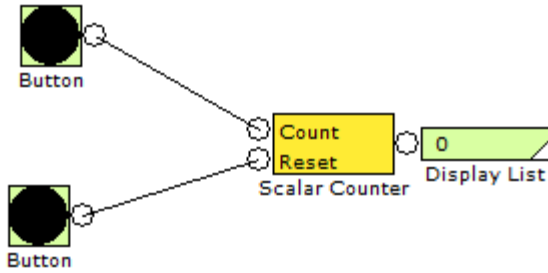
Output Period:

Output Maximum:



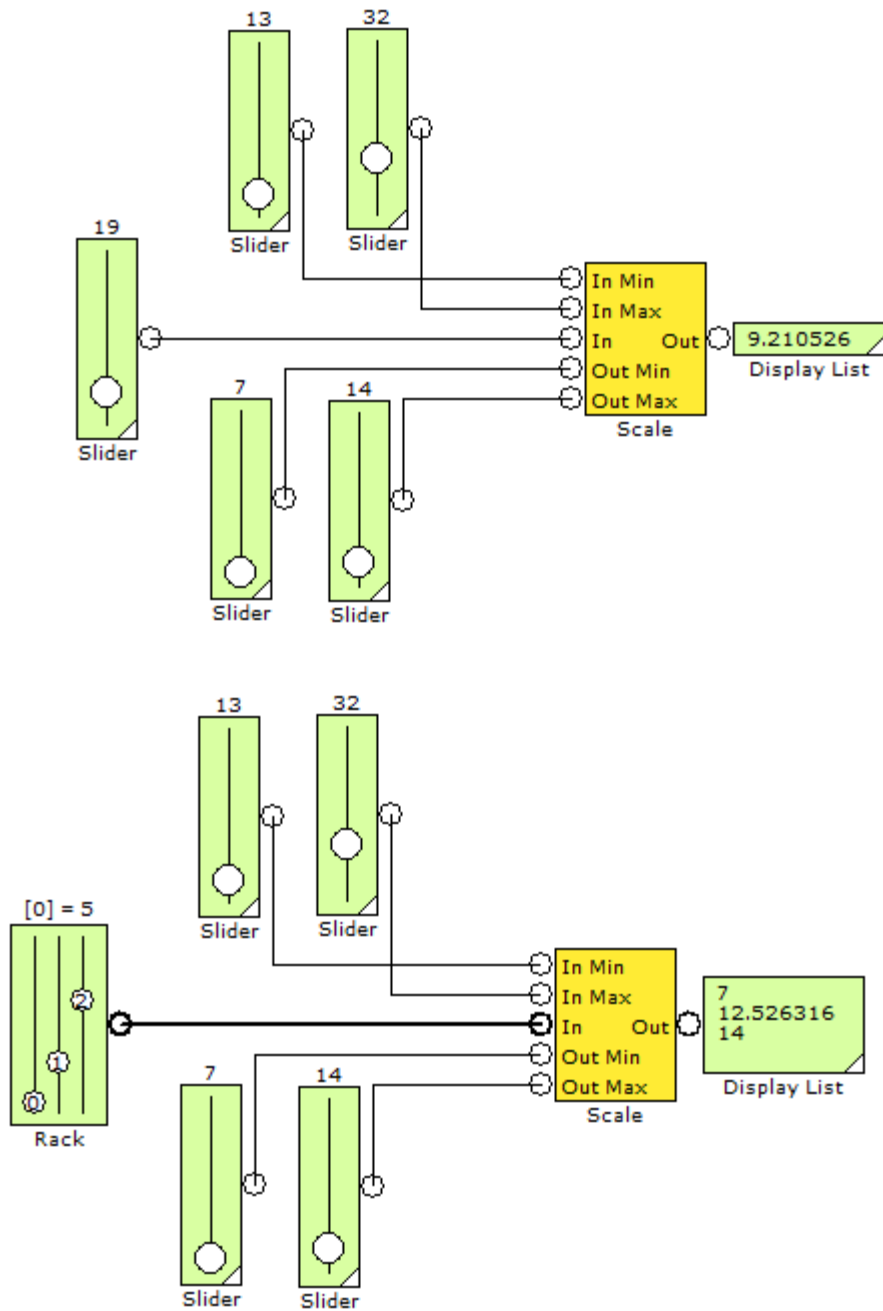
# Scalar Counter

The Scalar Counter component counts up or down and outputs the value. Functions - column 8



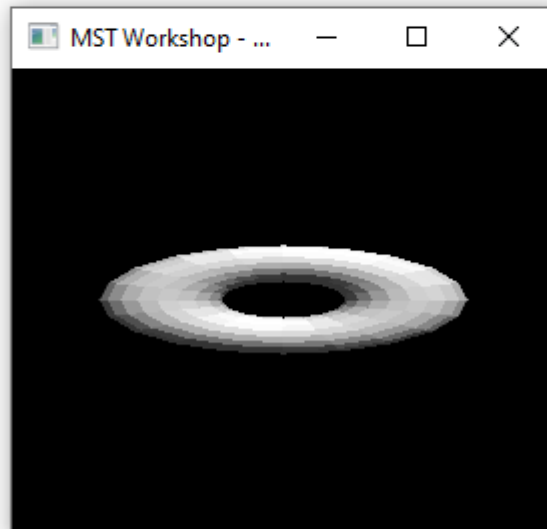
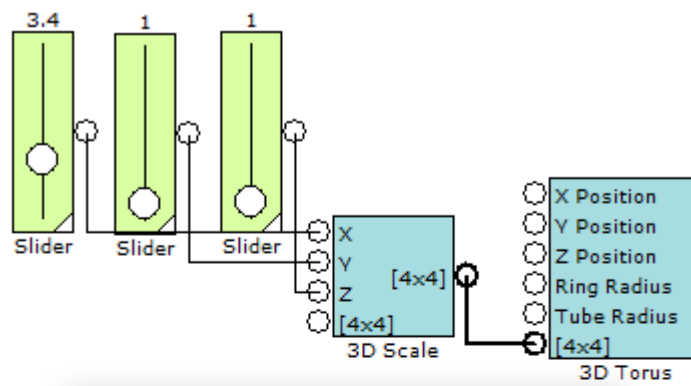
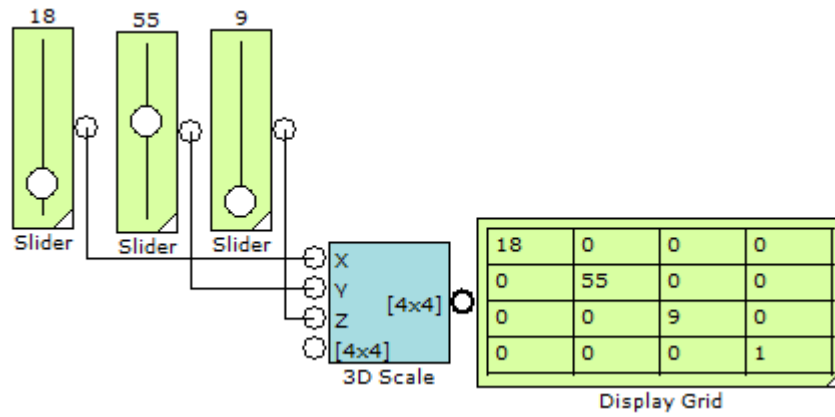
# Scale

The Scale component linearly scales its inputs from one range to another. Functions - column 5



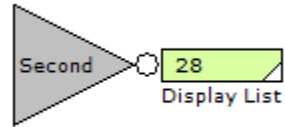
# 3D Scale

The 3D Scale magnifies a 4 by 4 matrix in the X, Y, and Z directions. 3D - column 3



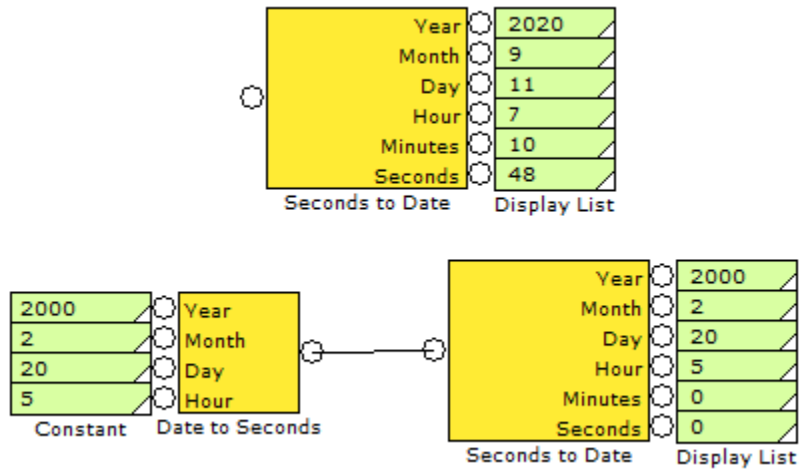
## Second

The Second component outputs the clock seconds from the computers internal clock. Functions - column 7



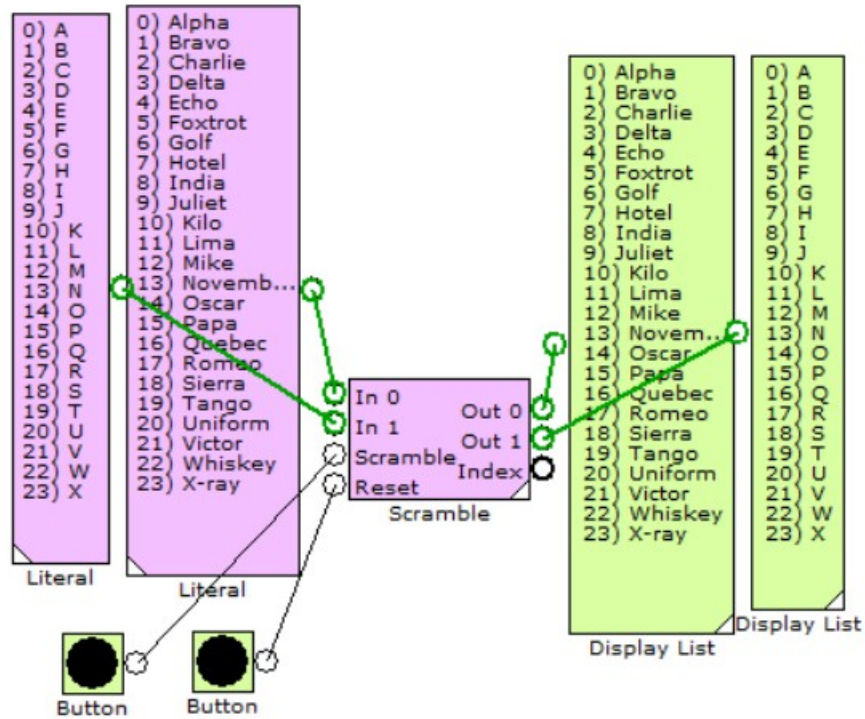
# Seconds to Date

The Seconds to Date component a number of seconds and converts it to a date. Functions - column 6



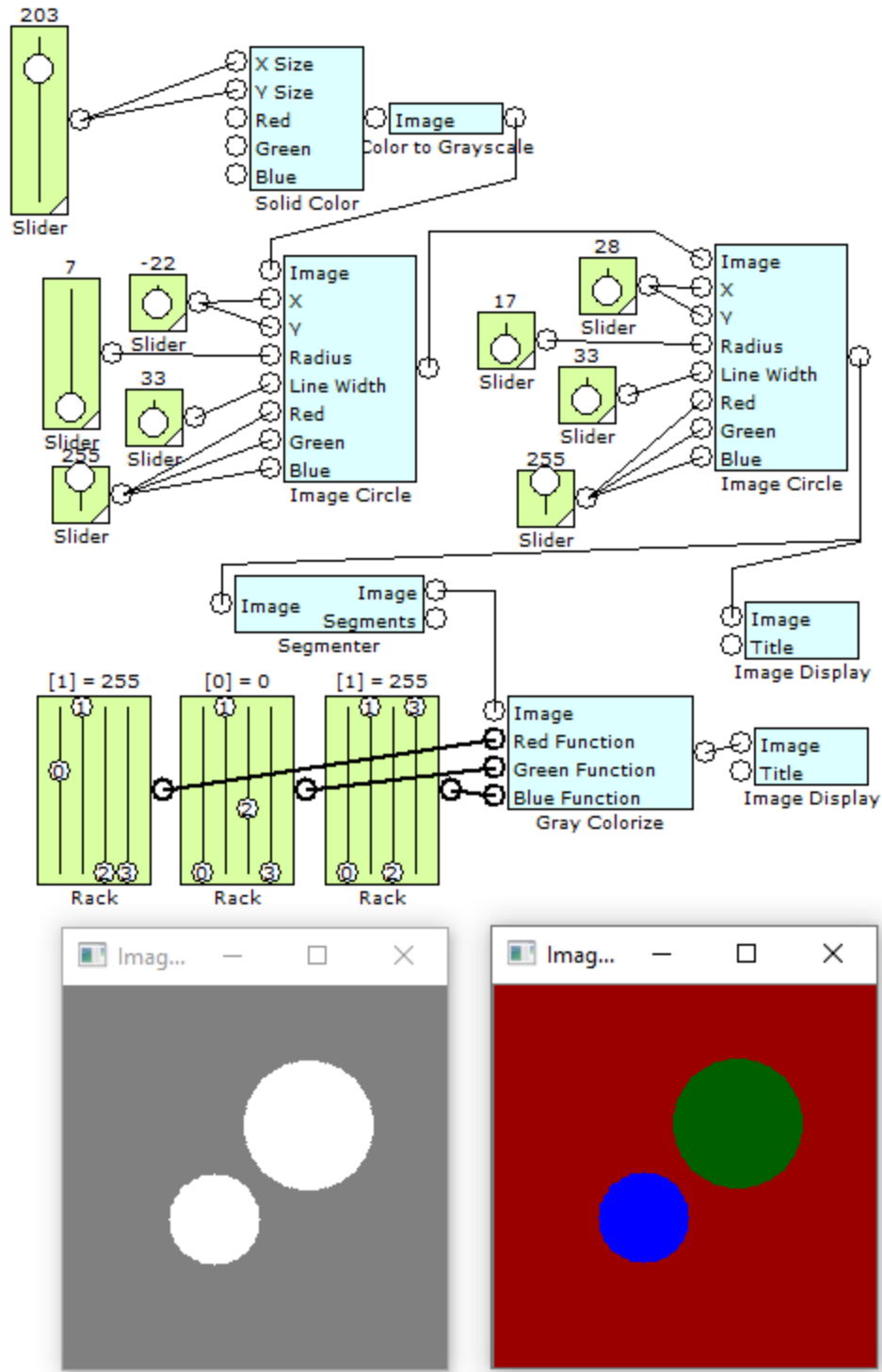
# Scramble

The Scramble component randomly rearranges the input(s). If there are multiple inputs they are scrambled with the same random index. Arrays – column 1



# Segmenter

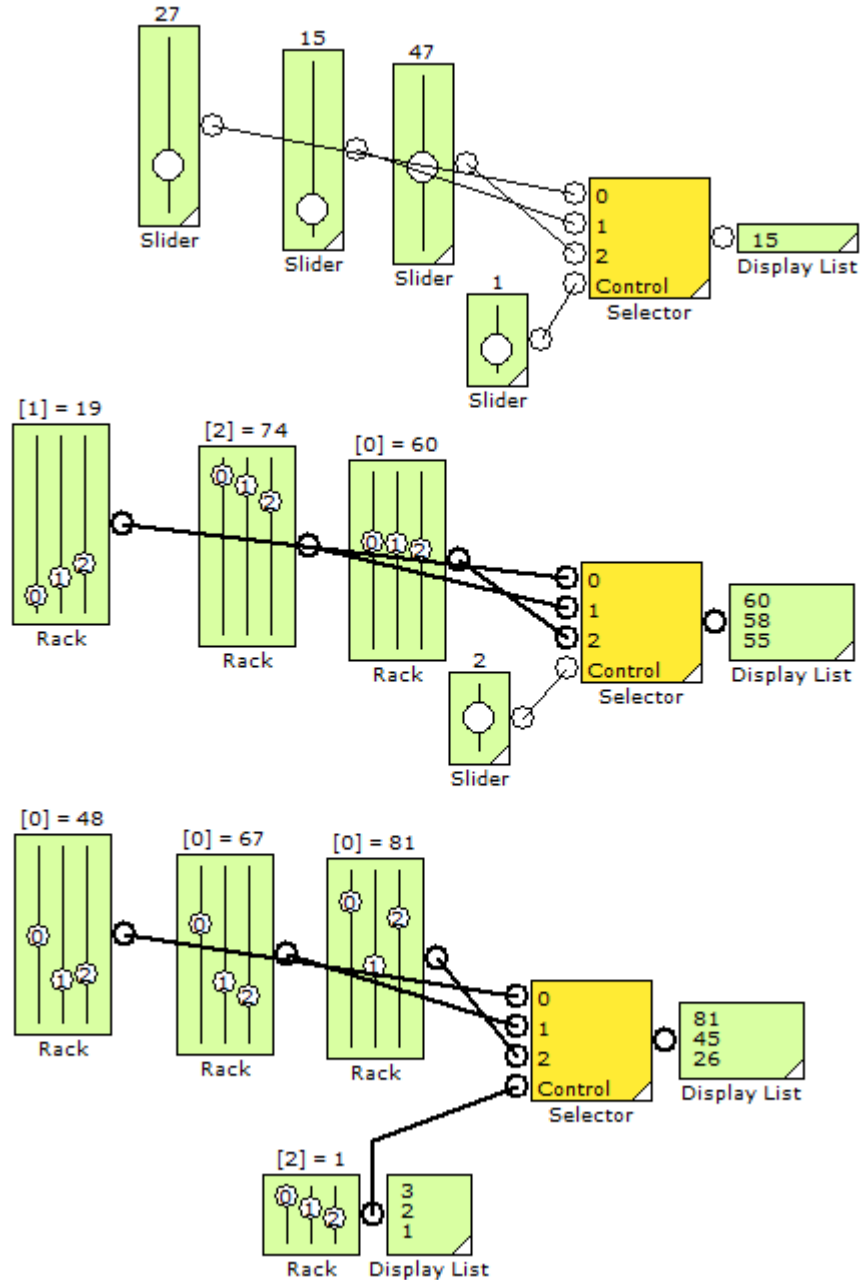
The Segmenter component inputs a grayscale image and outputs a grayscale image that has a different gray level for each segment found. Images - column 4

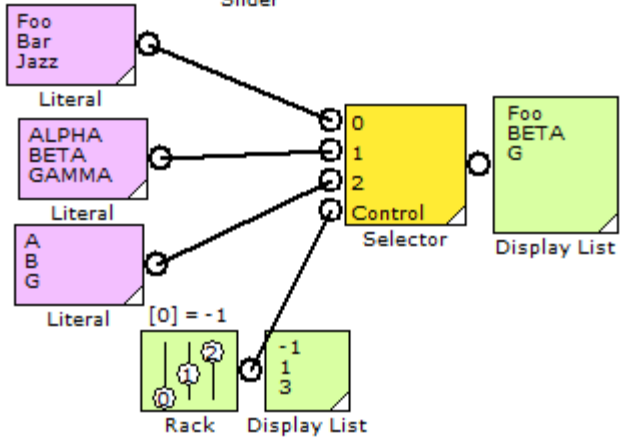
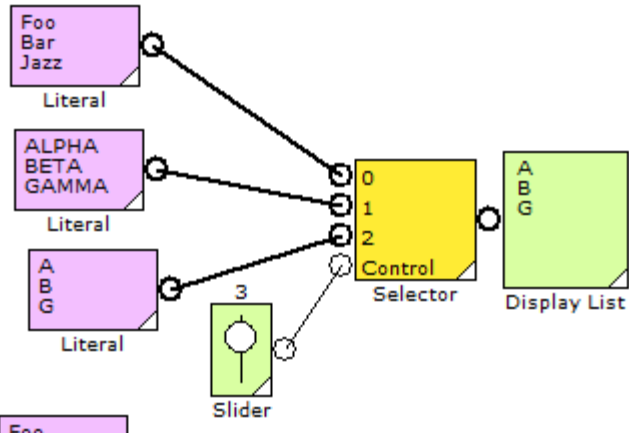
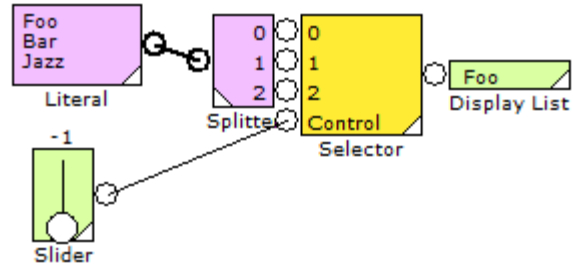




# Selector

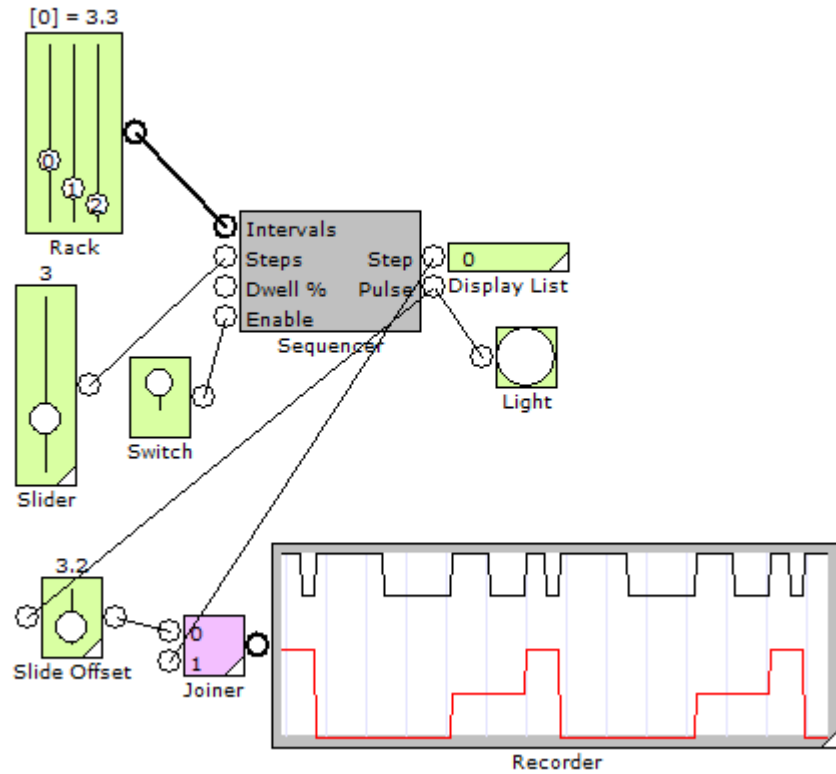
The Selector component picks one of the input values based on the control input. You can set the number of inputs. Functions - column 6





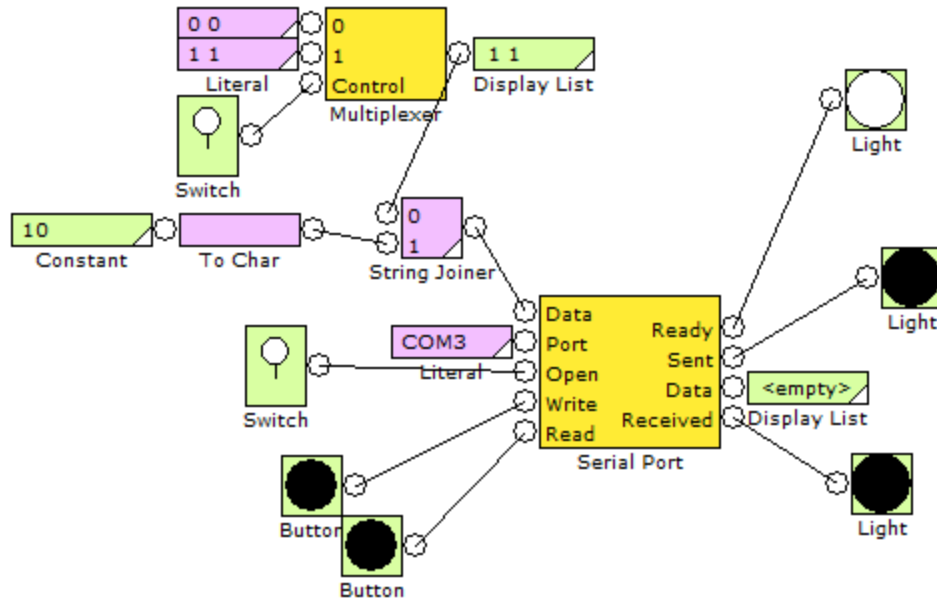
# Sequencer

The Sequencer component increments the Step output based on the number of Interval inputs. Each step is timed by the number of seconds in the Interval array. The Pulse output is high for the percent of the interval step specified in the Dwell % input. Arrays - column 7



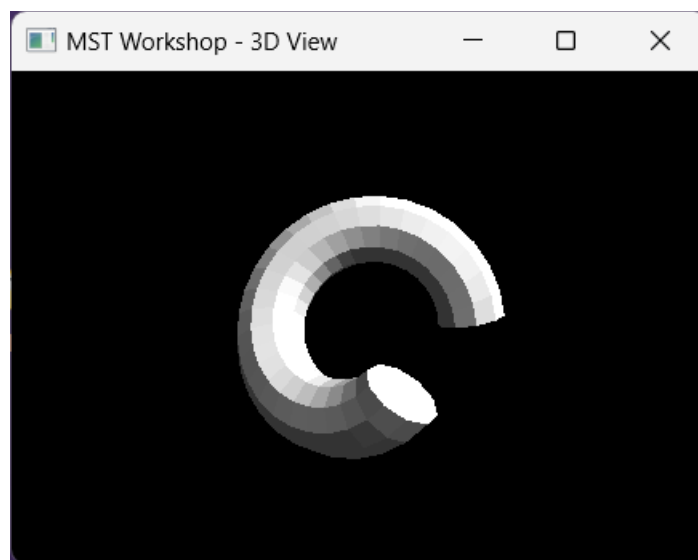
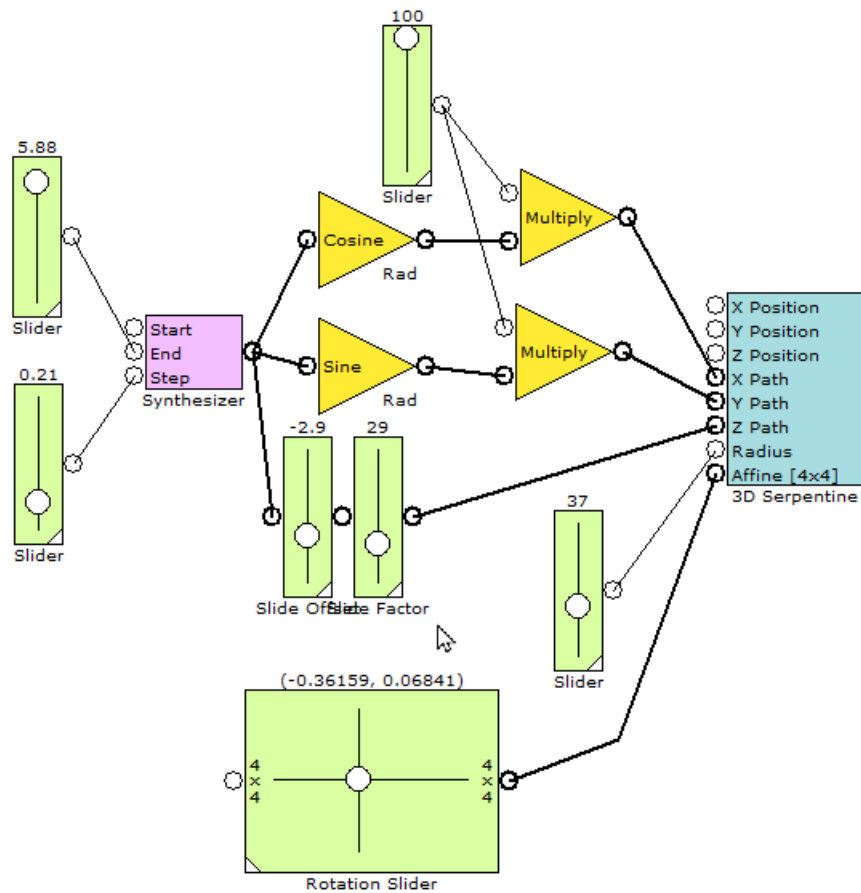
# Serial Port

The Serial Port component lets you send and receive messages through the computer's serial ports.  
Functions - column 7



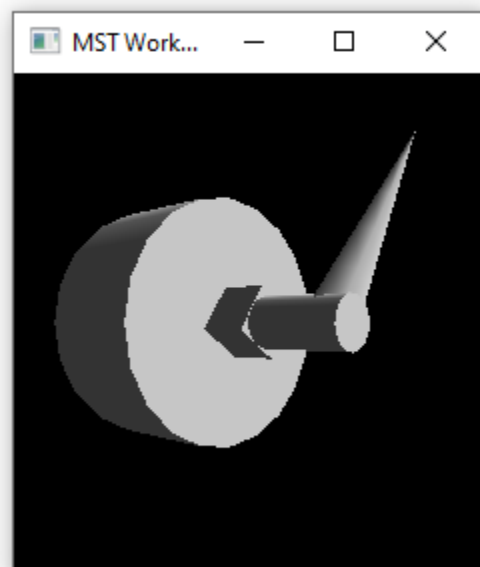
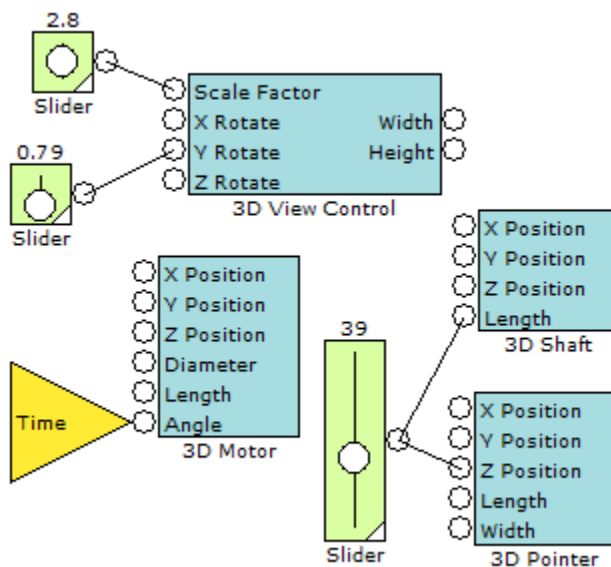
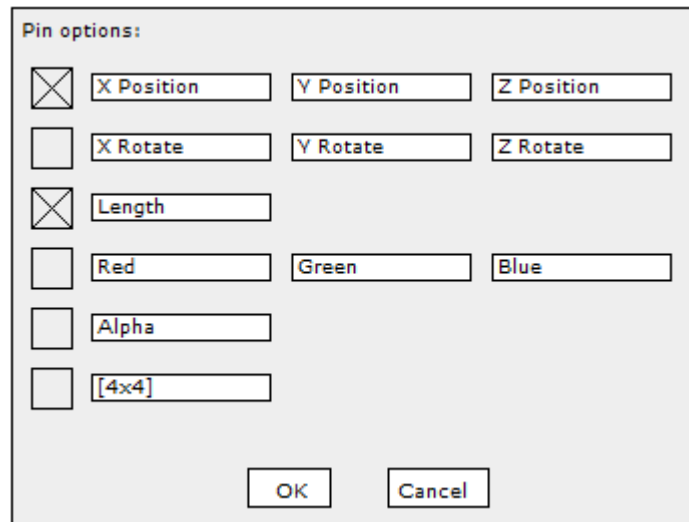
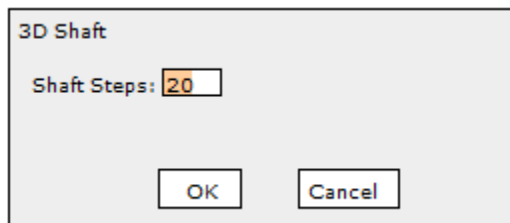
## 3D Serpentine

The 3D Serpentine component draws a circular extrusion in the 3D view window. The serpentine follows the given Path in X, Y and Z. The Radius can be constant or an array the same size as the path points. The X, Y, and Z inputs set the position and size of the Serpentine. The Affine [4x4] array input modifies the size, position and orientation. 3D - column 4



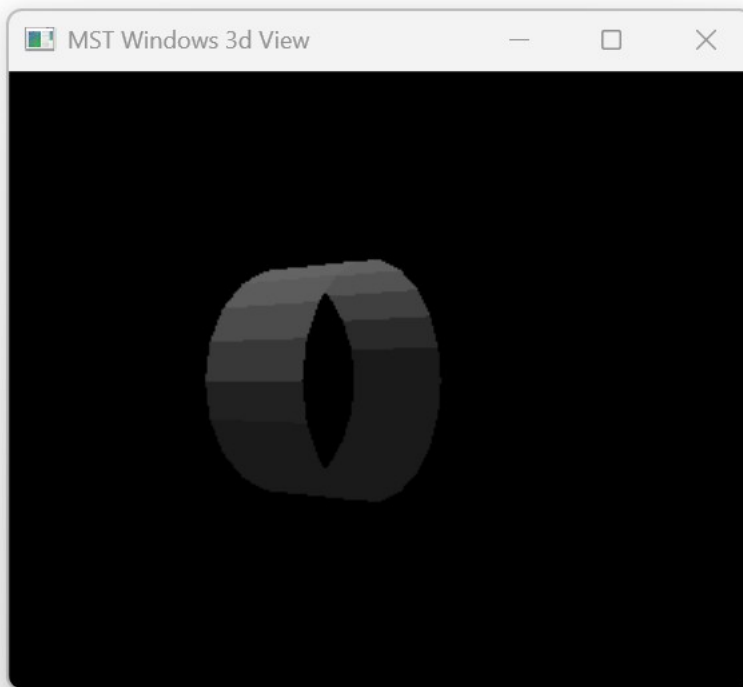
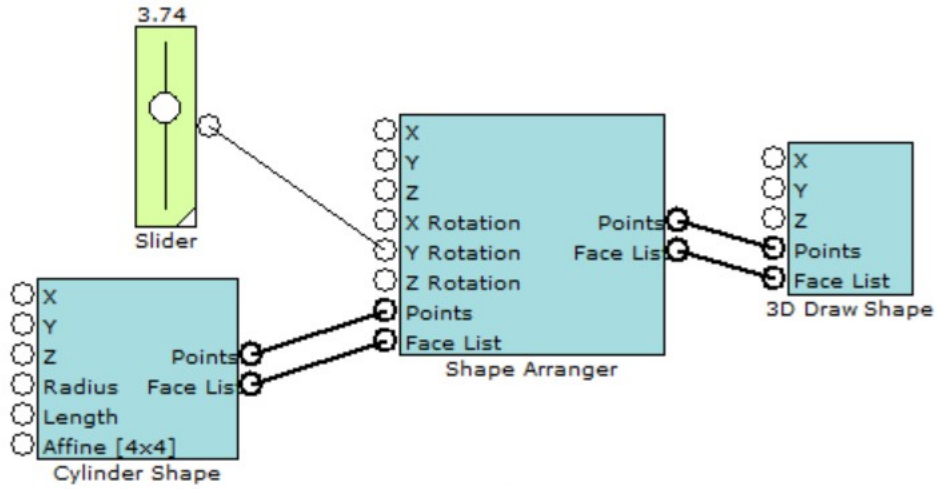
## 3D Shaft

The 3D Shaft component creates a connection between drivers like motors and driven objects like gears or other simulation objects. Place the Shaft at the same X, Y, Z position and orientation as a motor to connect the gear to the motor. 3D - column 4



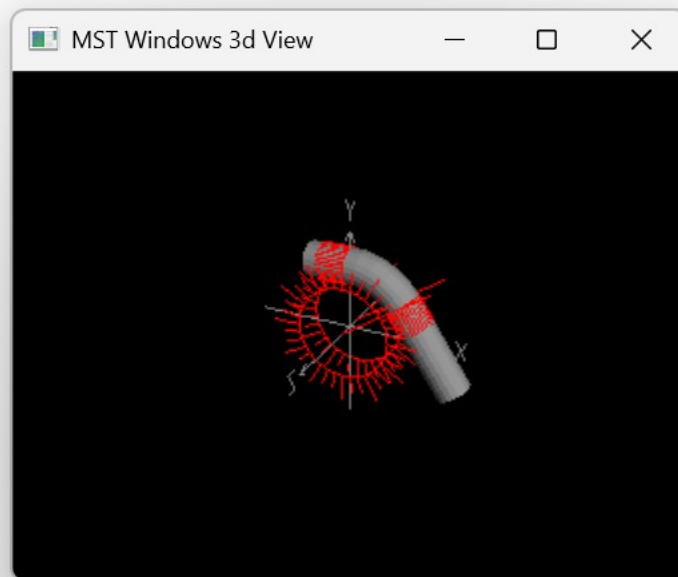
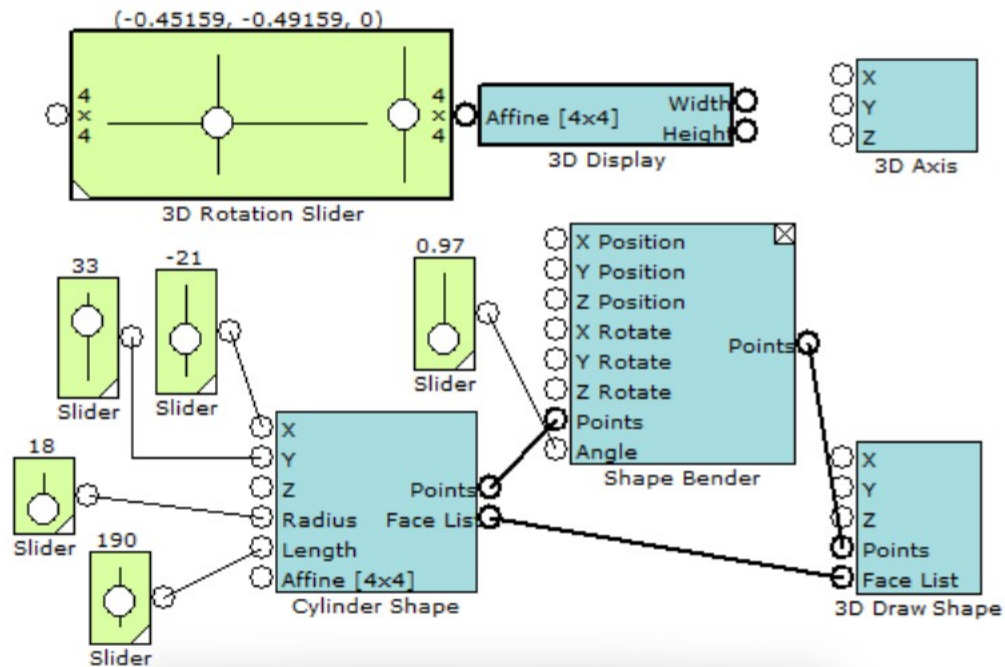
# Shape Arranger

The Shape Arranger component can move, resize, or rotate triangular meshes. If inputs are arrays, it can replicate parts. Use it to reshape shapes before plotting in the 3D view. 3D - column 6



# Shape Bender

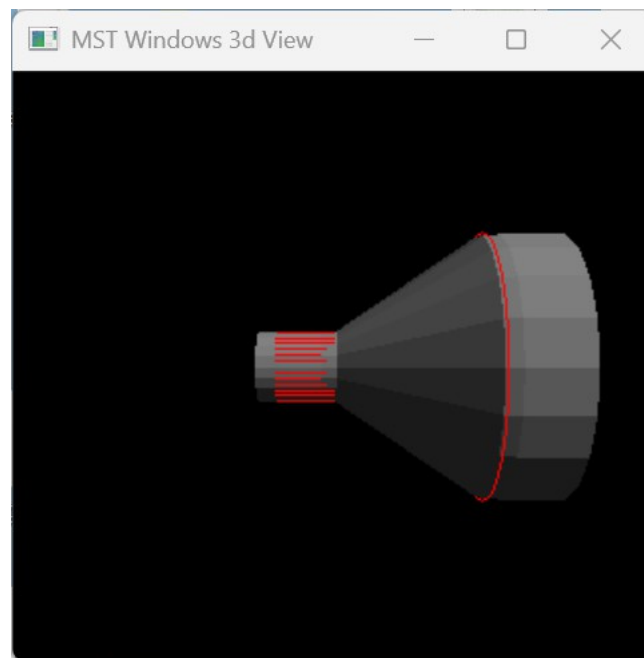
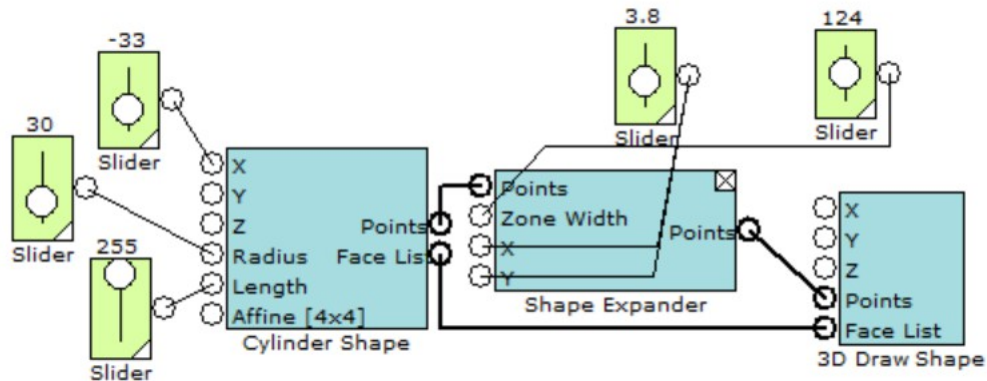
The Shape Bender component bends points along the chosen axis. When the Zone Width is zero, points are rotated on the axis by the input angle. When the Zone Width is positive, points along the axis that are negative will be held fixed. Points beyond the Zone Width will be bent by the input Angle. Use it to bend shapes before plotting or additional operations in the 3D view. 3D - column 6





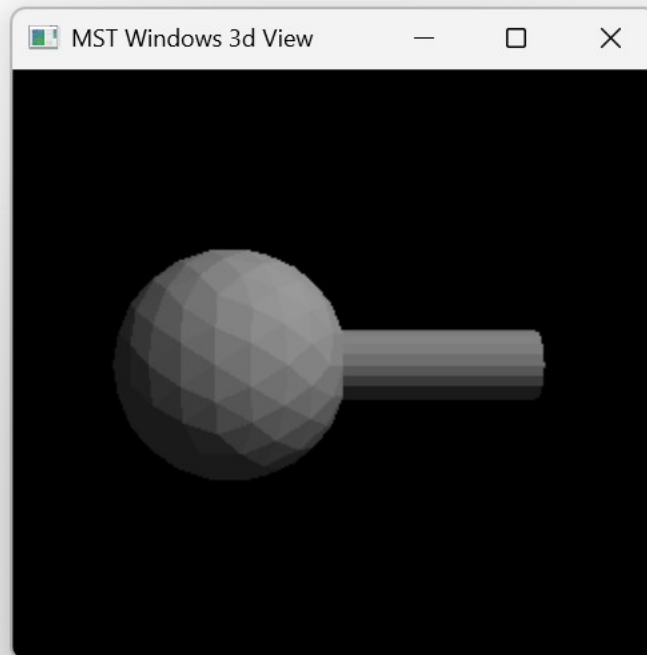
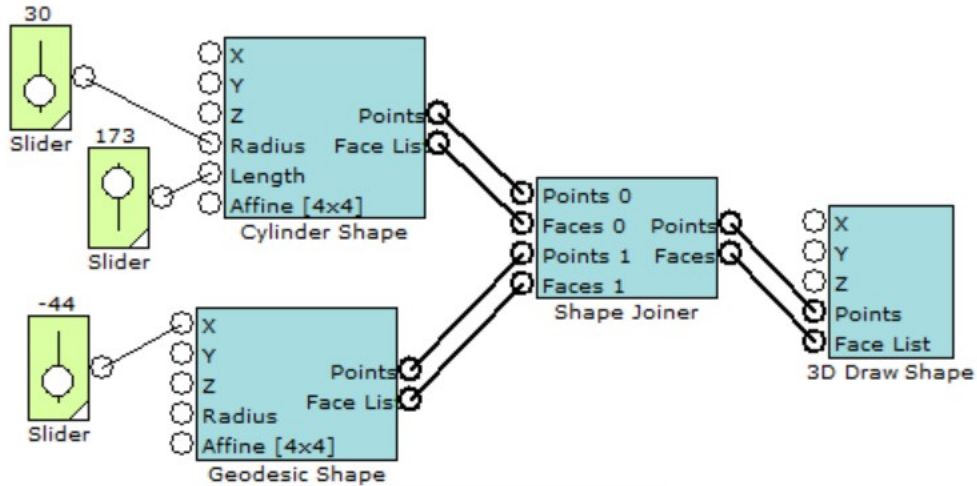
## Shape Expander

The Shape Expander component enlarges points between the negative part of the chosen axis and the Zone Width. Use it to reshape shapes before further operations of drawing in the 3D view. When the Zone Width is zero, the whole shape will be expanded. When the Zone Width is positive, Points on the part that are less than zero on the X axis will remain fixed. Points on the part that are beyond the Zone Width on the chosen axis will be expanded by the X and Y input values. 3D - column 6



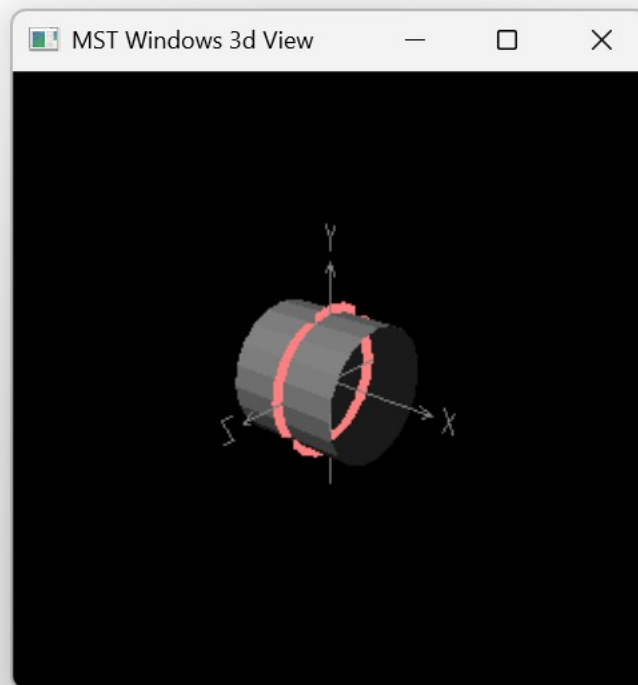
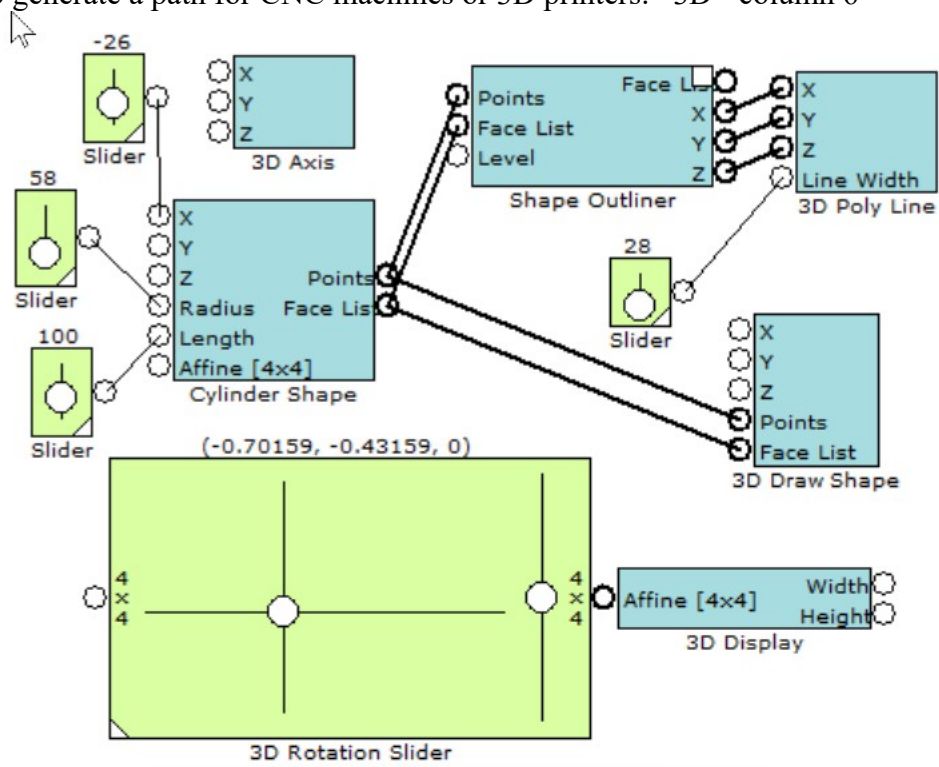
## Shape Joiner

The Shape Joiner component joins two triangular meshes. Use it to reshape shapes before plotting in the 3D view. 3D - column 6



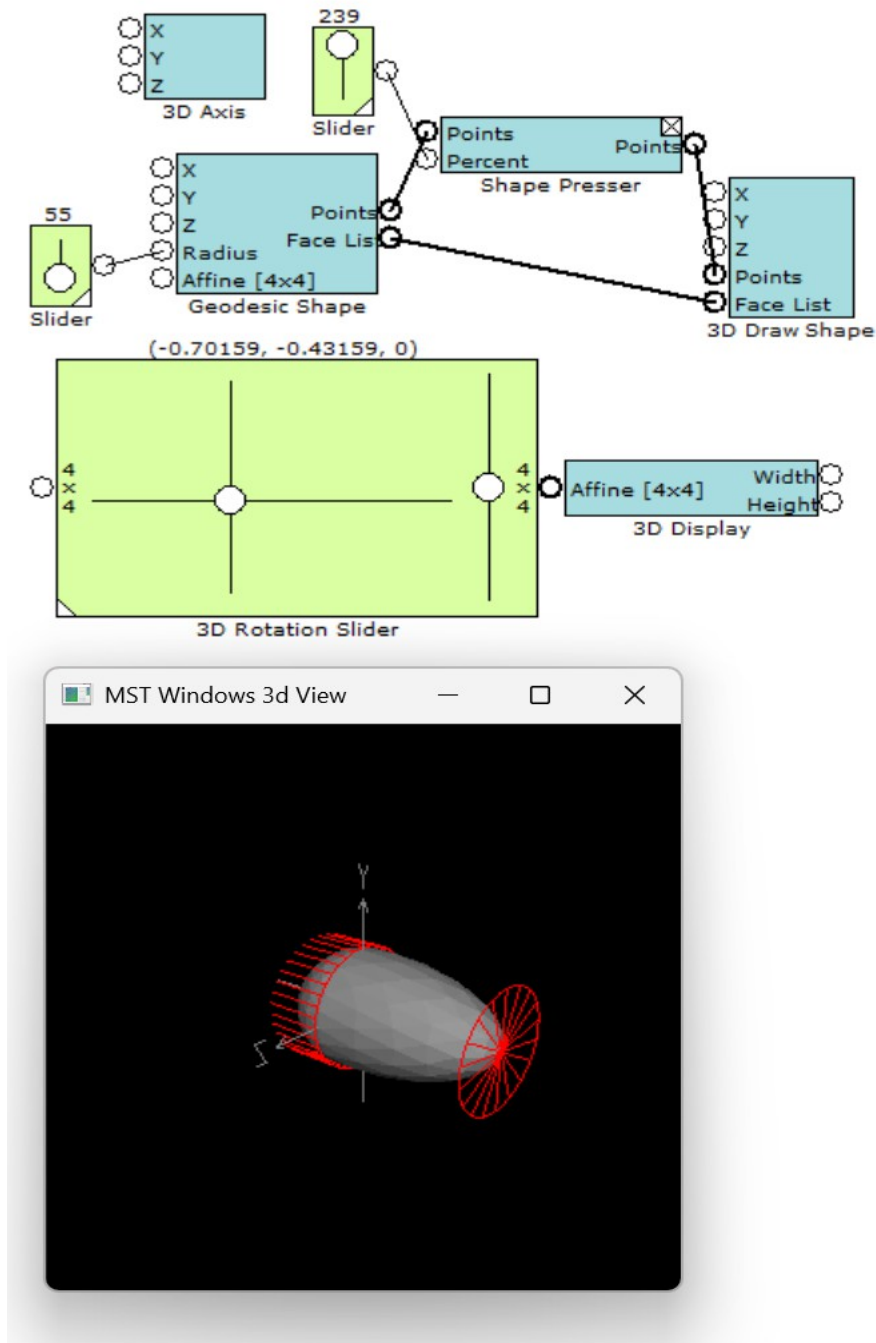
# Shape Outliner

The Shape Outliner component captures an outline of the input shape of a triangular mesh. The outline can be used to generate a path for CNC machines or 3D printers. 3D - column 6



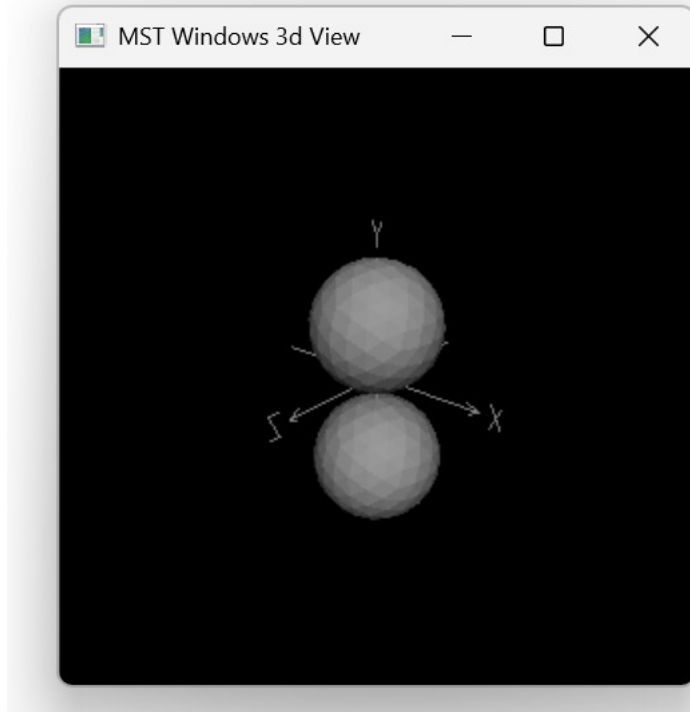
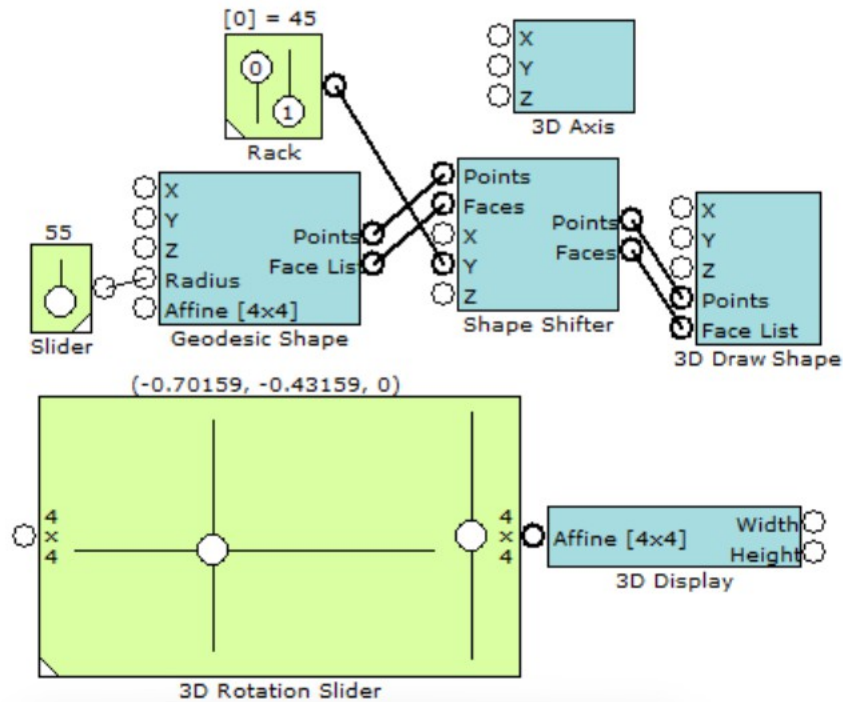
# Shape Presser

The Shape Presser component shrinks points between the negative part of the chosen axis and the Zone Width. Use it to reshape shapes before further operations of drawing in the 3D view. When the Zone Width is zero, the whole shape will be expanded. When the Zone Width is positive, Points on the part that are less than zero on the X axis will remain fixed. Points on the part that are beyond the Zone Width on the chosen axis will be expanded by the X and Y input values. 3D - column 6



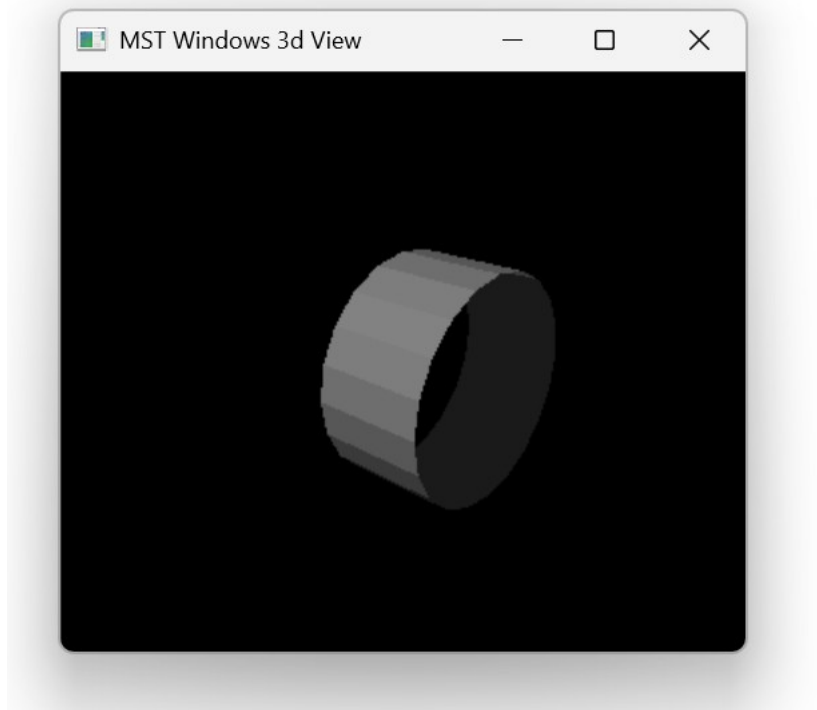
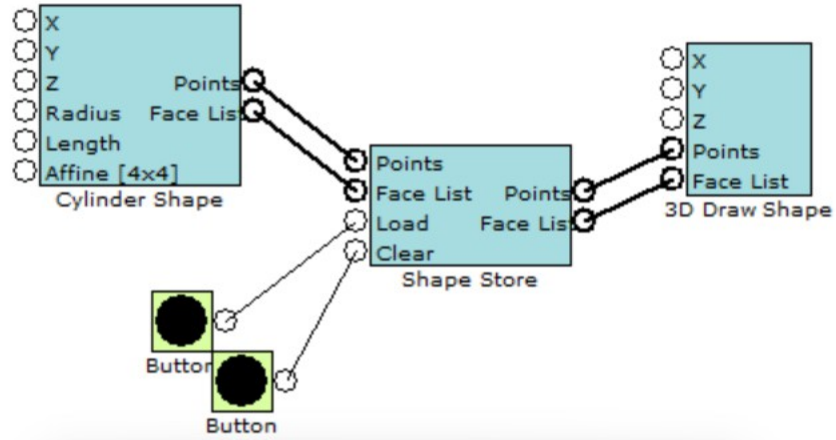
# Shape Shifter

The Shape Shifter component Shifts points in 3D. Use it to reshape shapes before plotting in the 3D view. 3D - column 6



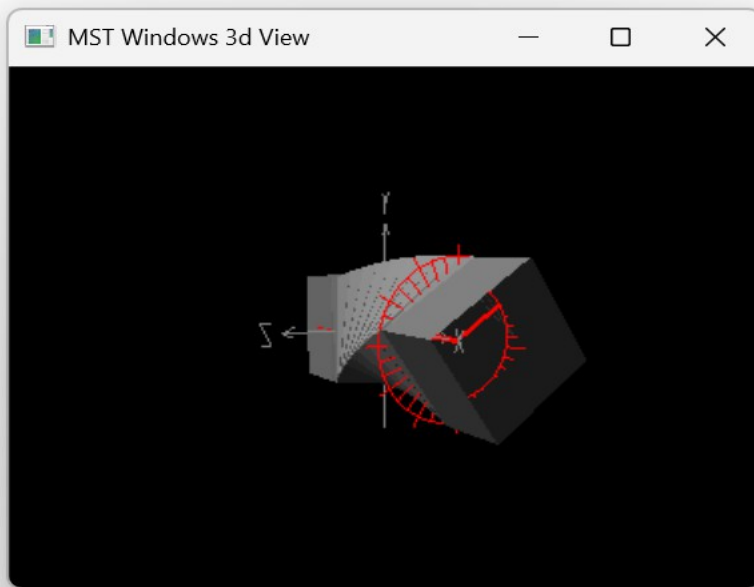
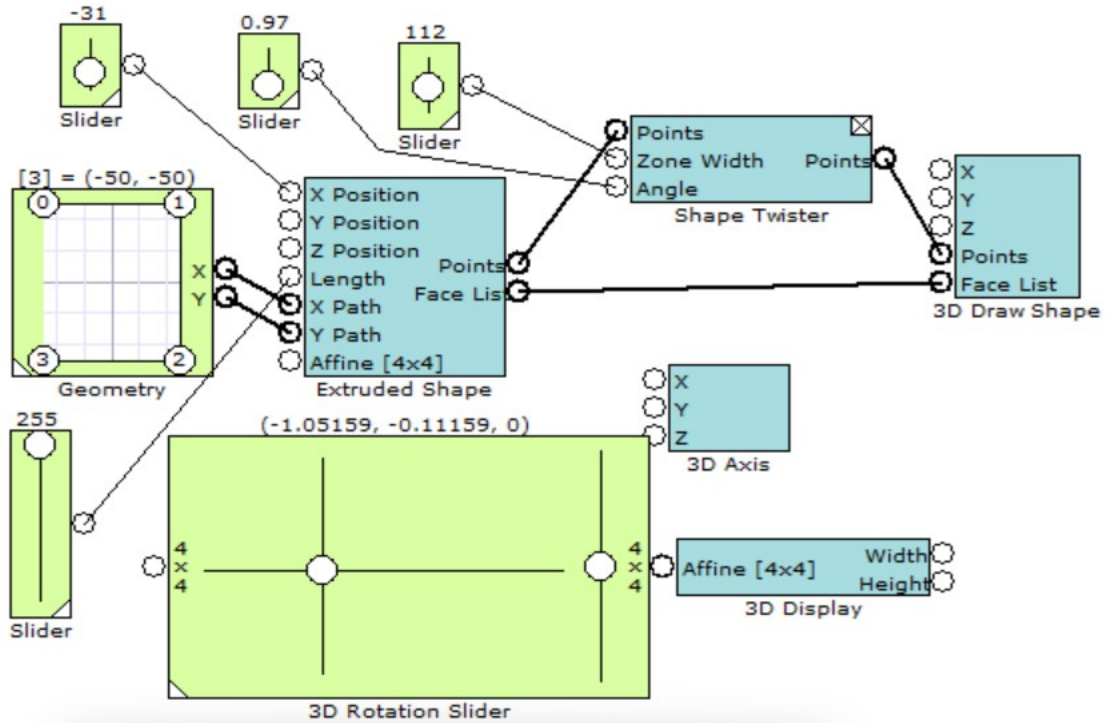
# Shape Store

The Shape Store component stores a shape's triangular mesh composed of points and face list. After bending or twisting you can save the bent or twisted part for later use. 3D - column 6



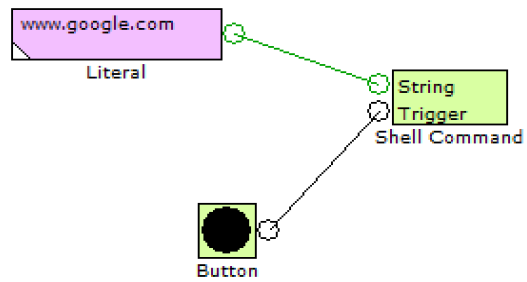
# Shape Twister

The Shape Twister component rotates points around the chosen axis. Use it to reshape Parts before further operations of drawing in the 3D view. When the Zone Width is zero, the whole shape will be rotated. When the Zone Width is positive, Points on the part that are less than zero on the X axis will remain fixed. Points on the part that are beyond the Zone Width on the chosen axis will be twisted by the input angle.



## Shell Command

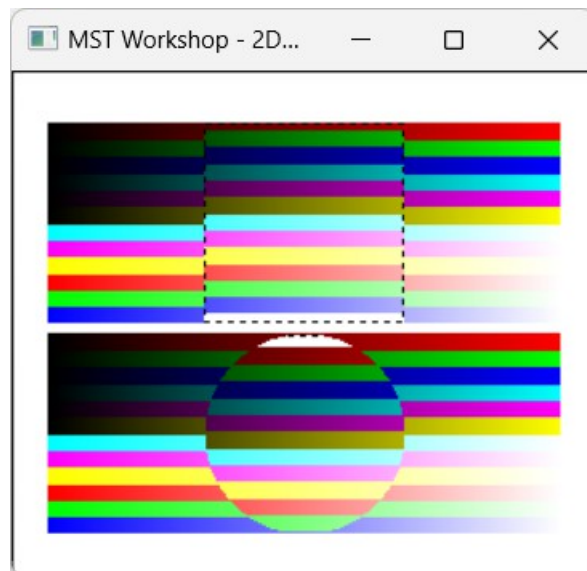
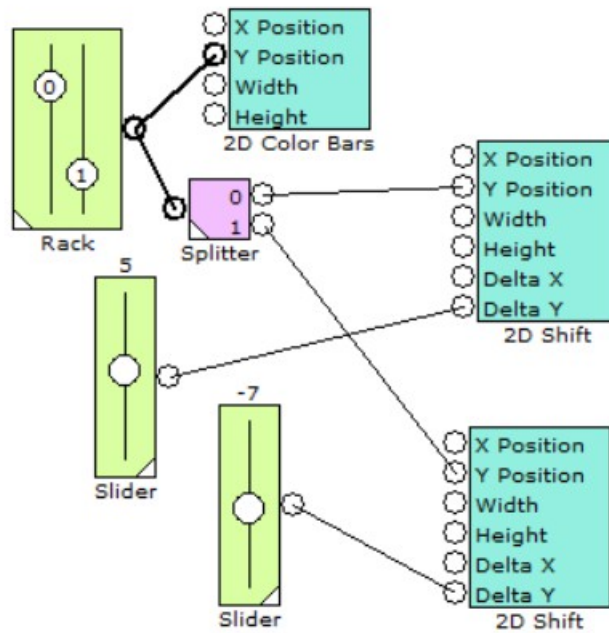
The Shell Command opens a new window using the input shell command. For example the string "[www.Google.com](http://www.Google.com)" will open Google's web site. Controls – column 3





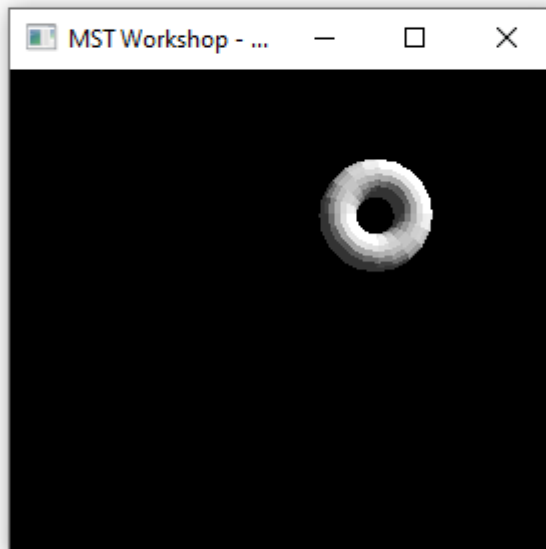
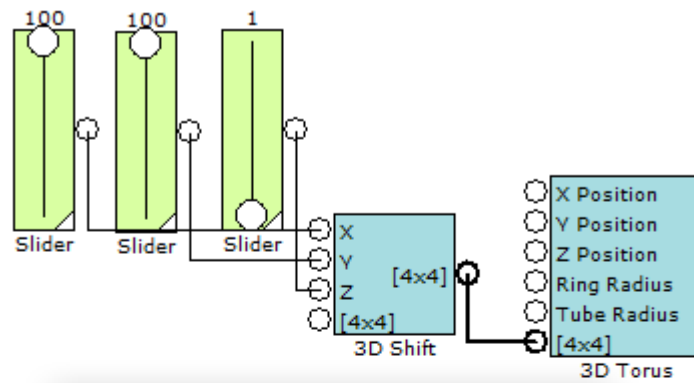
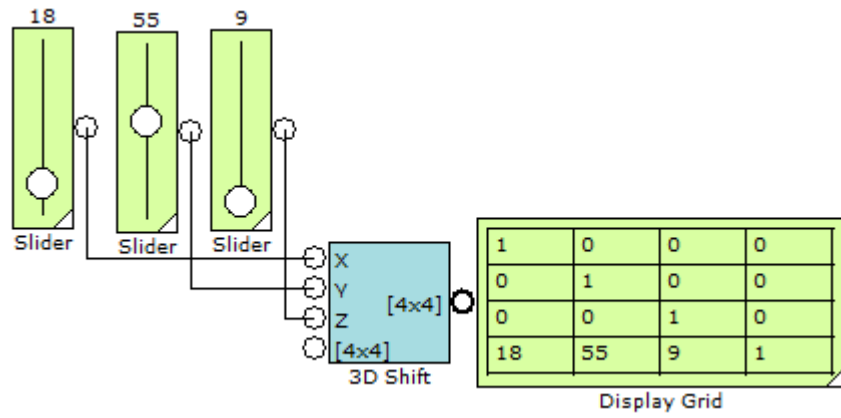
## 2D Shift

The 2D Shift shifts the image in the 2D drawing it covers. 2D – column 1



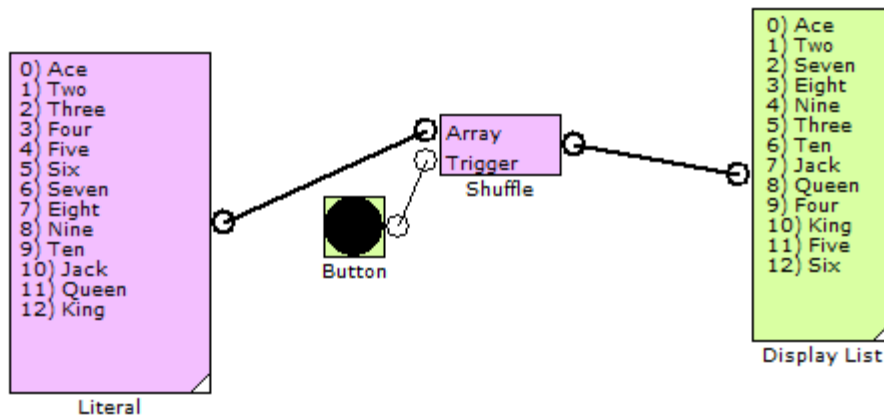
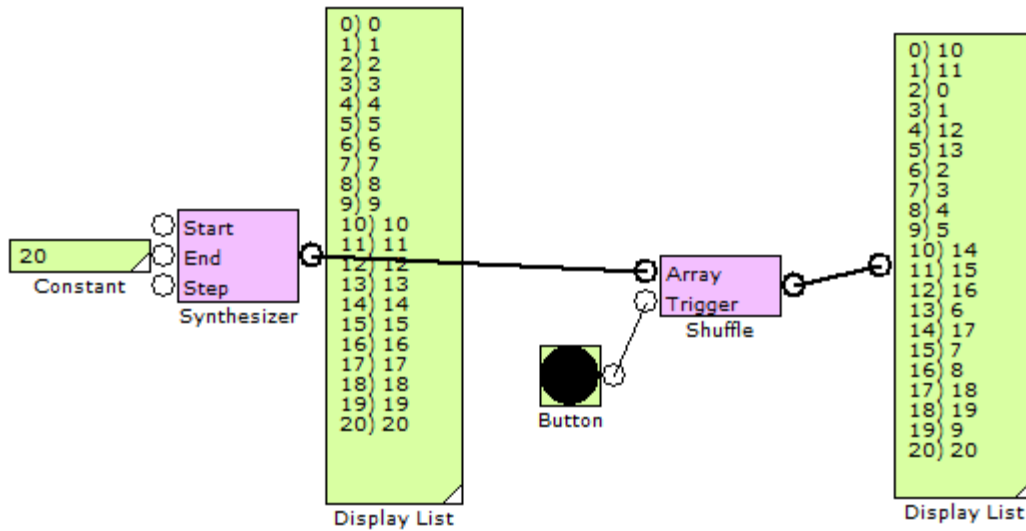
# 3D Shift

The 3D Shift translates a 4 by 4 matrix in the X, Y, and Z directions. 3D - column 3



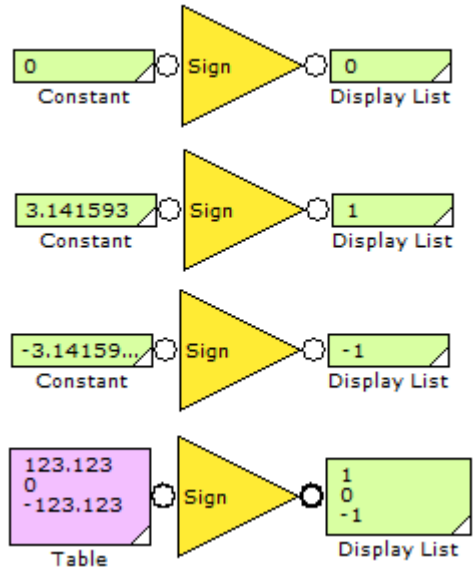
# Shuffle

The Shuffle component cuts the input list in two nearly equal groups, then shuffles them back together. Like cutting a deck cards and merging them together. Arrays - column 1



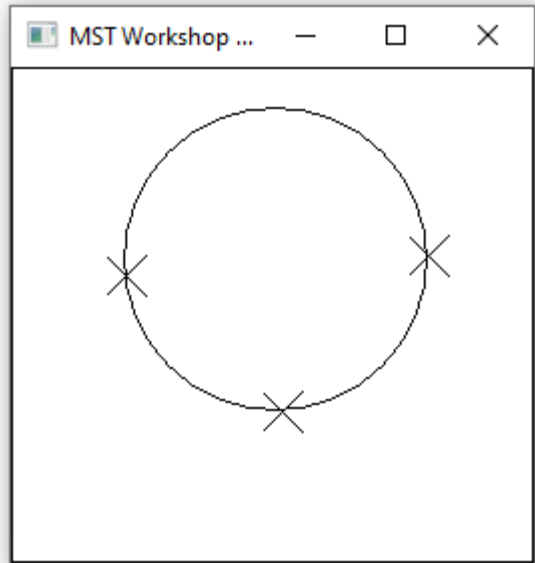
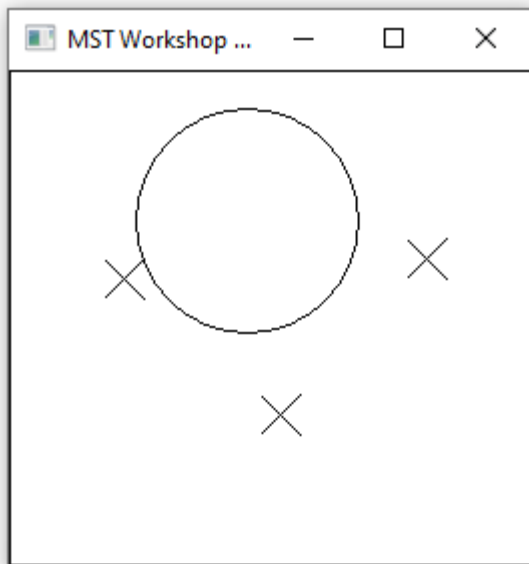
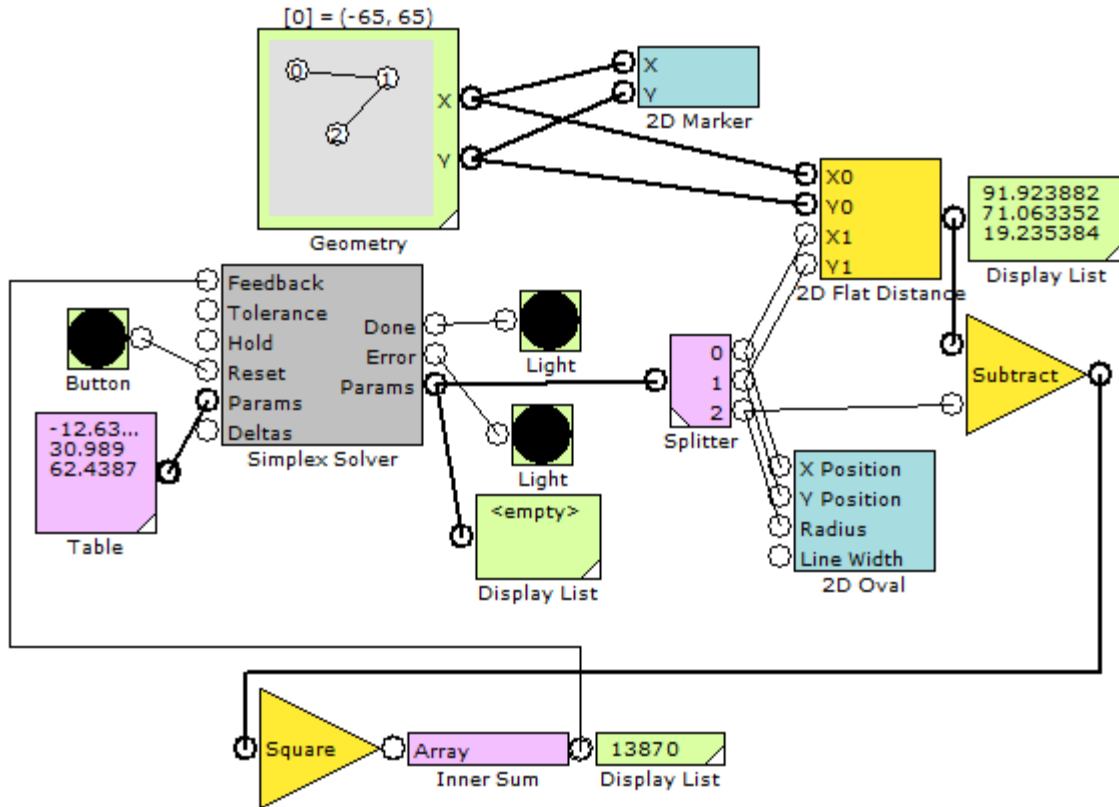
# Sign

The Sign component outputs zero if the input is zero, one if the input is positive or minus one if the input is negative. Input can be scalar or an array. Functions - column 2



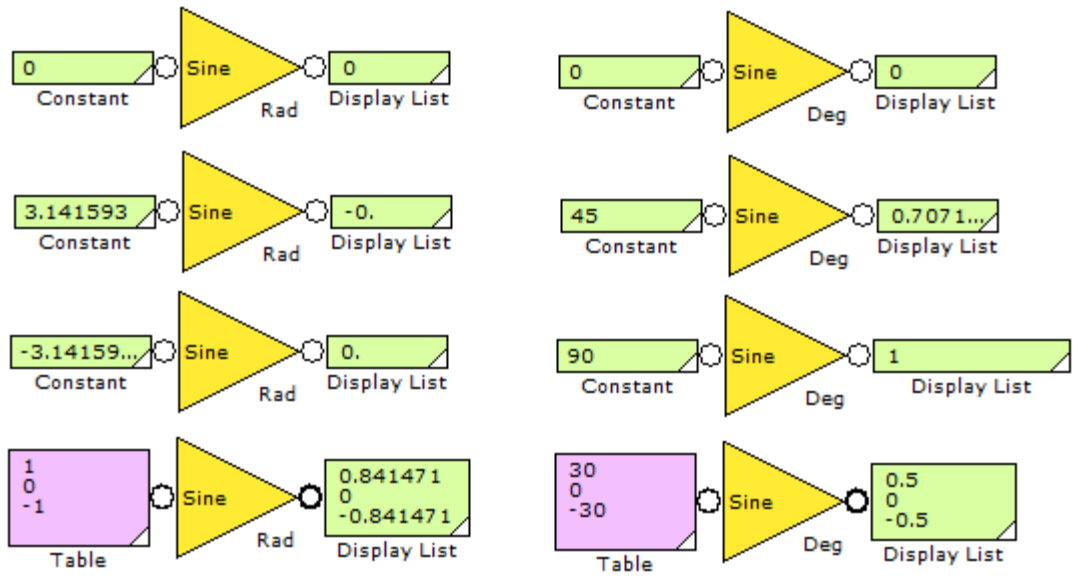
# Simplex Solver

The Simplex Solver component finds the best parameters to reduce feedback. Functions - column 6



# Sine

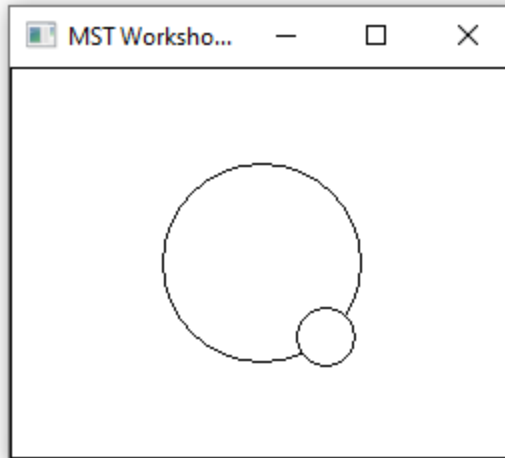
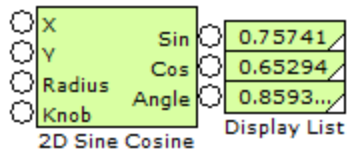
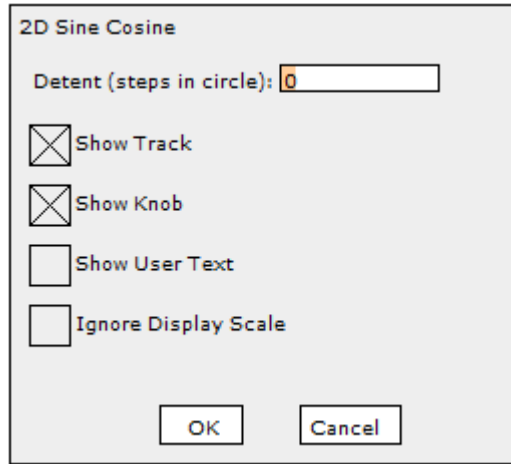
The Sine component calculates the trigonometric sine of the input (in radians). Input can be scalar or an array. Functions - column 5



# 2D Sine Cosine

The 2D Sin/Cos component outputs sine and cosine based on the angle of the dial. column 1

Controls -



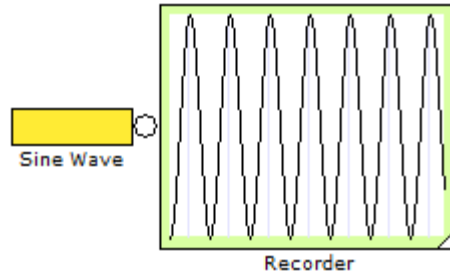
# Sine Wave

The Sine Wave component outputs a sinusoidal wave when the clock is running. Functions - column 8

Sine Wave

Output Period:

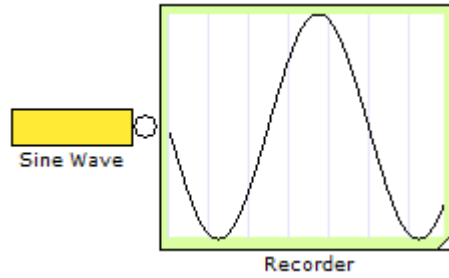
Output Maximum:



Sine Wave

Output Period:

Output Maximum:

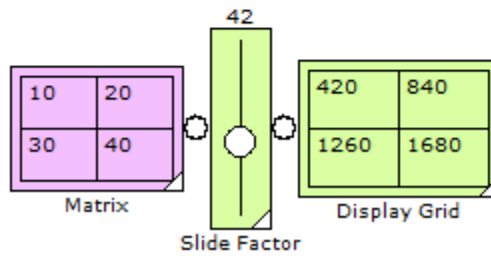
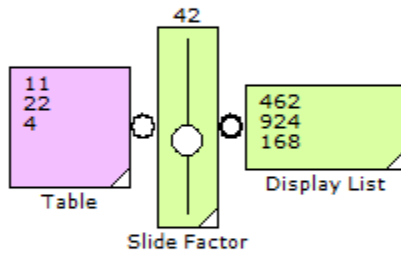
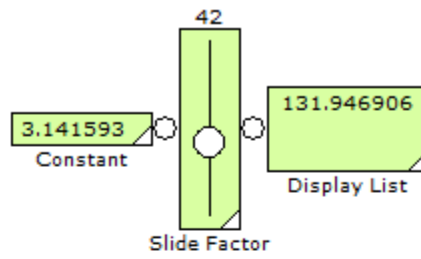




# Slide Factor

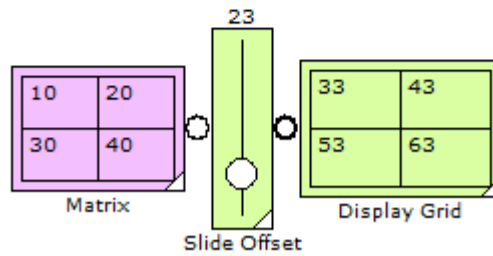
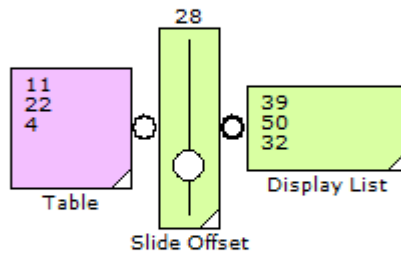
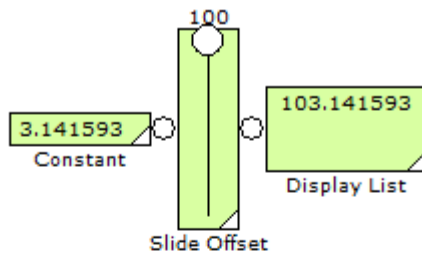
The Slide Factor component multiplies its inputs by its dial position.

Controls - column 2



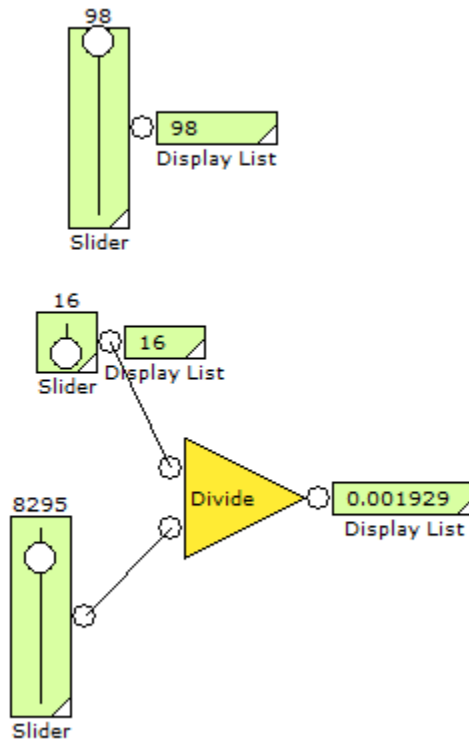
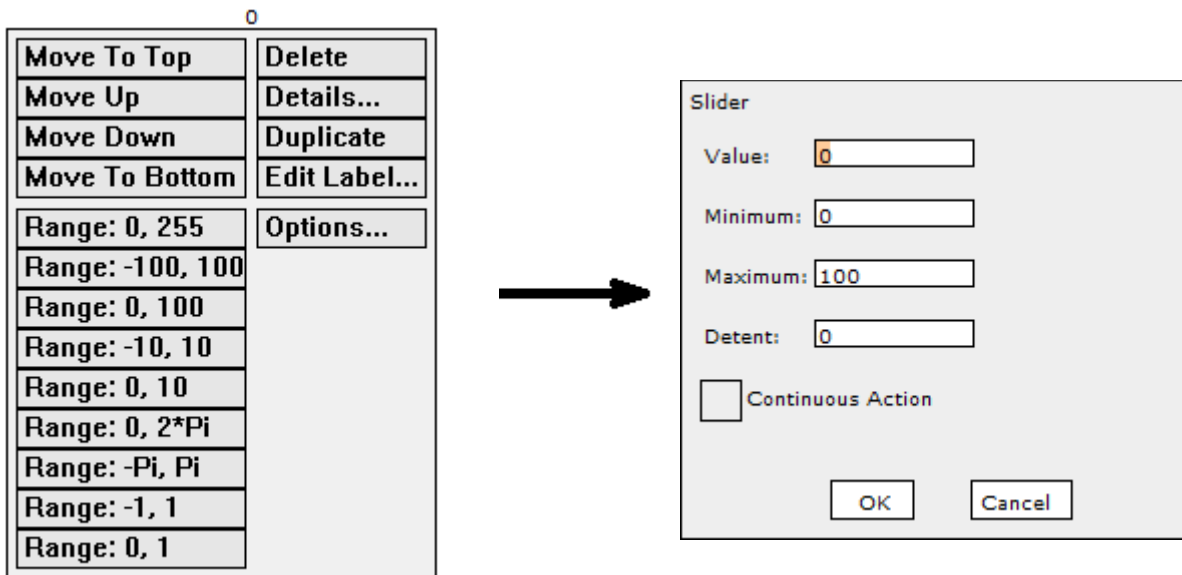
# Slide Offset

The Slide Offset component adds its dial position to its inputs. Controls - column 3



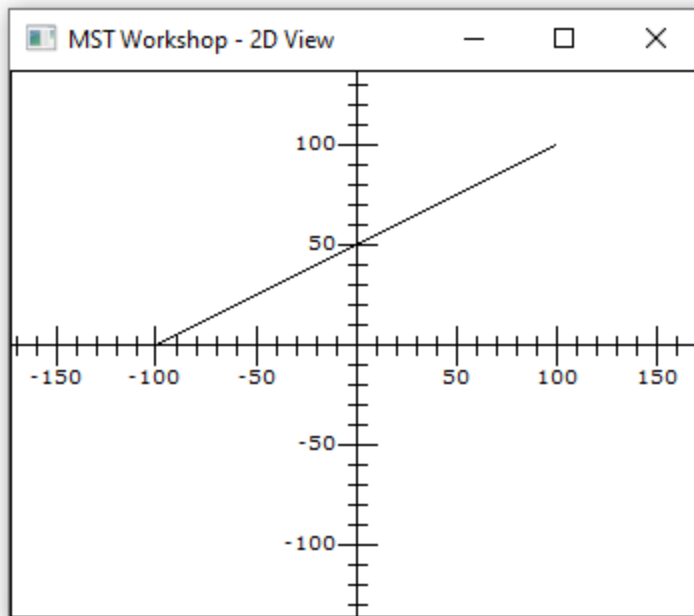
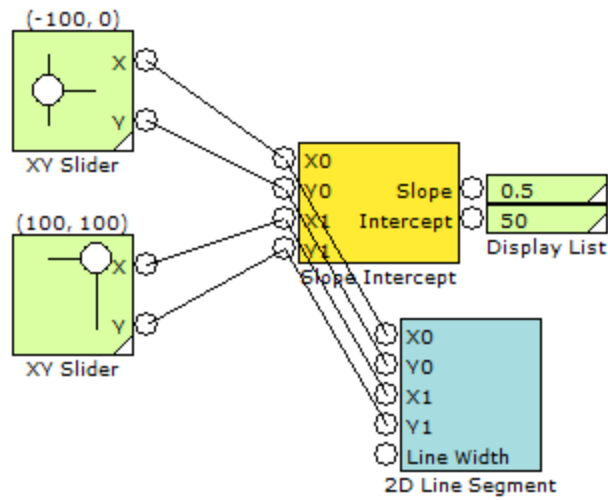
# Slider

The Slider component outputs a constant number based on its dial position. Note: This component can be operated using the wheel on the mouse. Controls - column 1



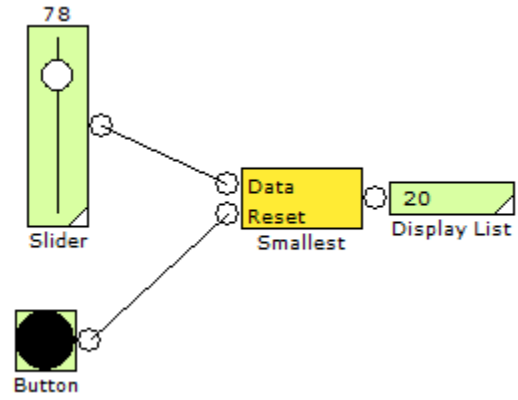
# Slope Intercept

The Slope Intercept component calculates the slope and intercept between two points. Functions - column 8



# Smallest

The Smallest component watches the Data input and outputs the smallest value since it was reset.  
Functions - column 2



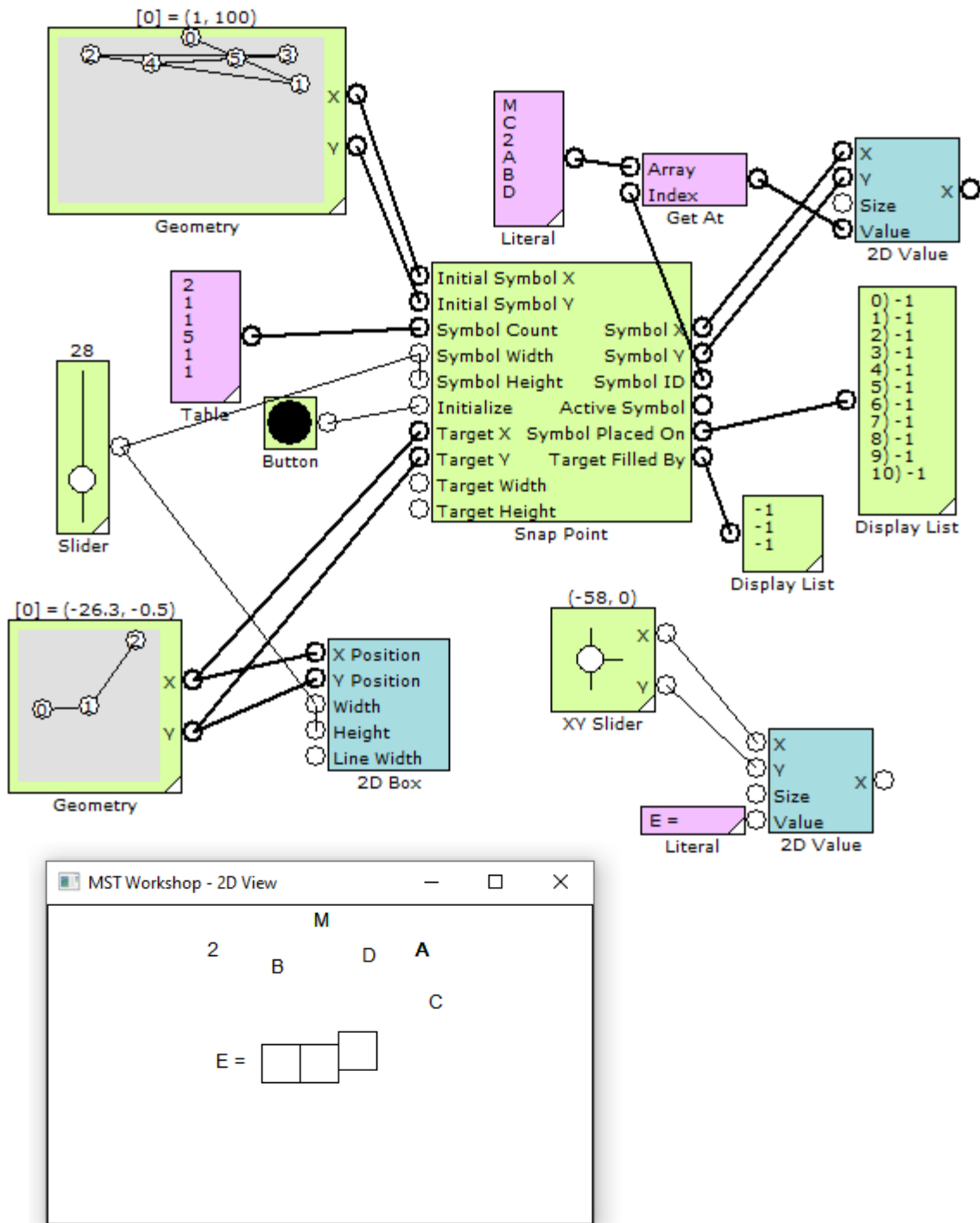
# Snap Point

The Snap Point allows you to guide symbols to defined targets.

Initial inputs are the XY coordinates of the 2D symbols at the start.

Target inputs are the XY coordinates of the destinations.

Snapped outputs indicate the part is in place. Filled outputs indicate the target position has been filled. Controls - column 3



## Snap Point+

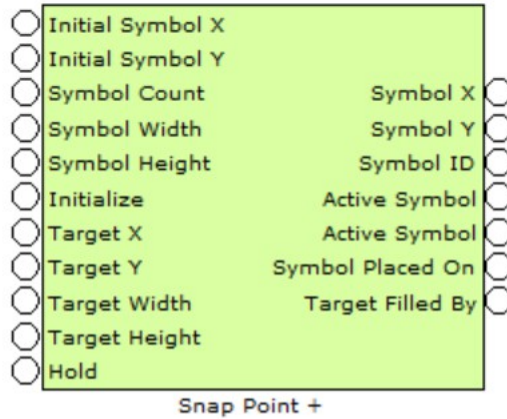
The Snap Point+ allows you to guide symbols to defined targets. This component has an added Hold input. Everything else is the same as Snap Point.

Initial inputs are the XY coordinates of the 2D symbols at the start.

Target inputs are the XY coordinates of the destinations.

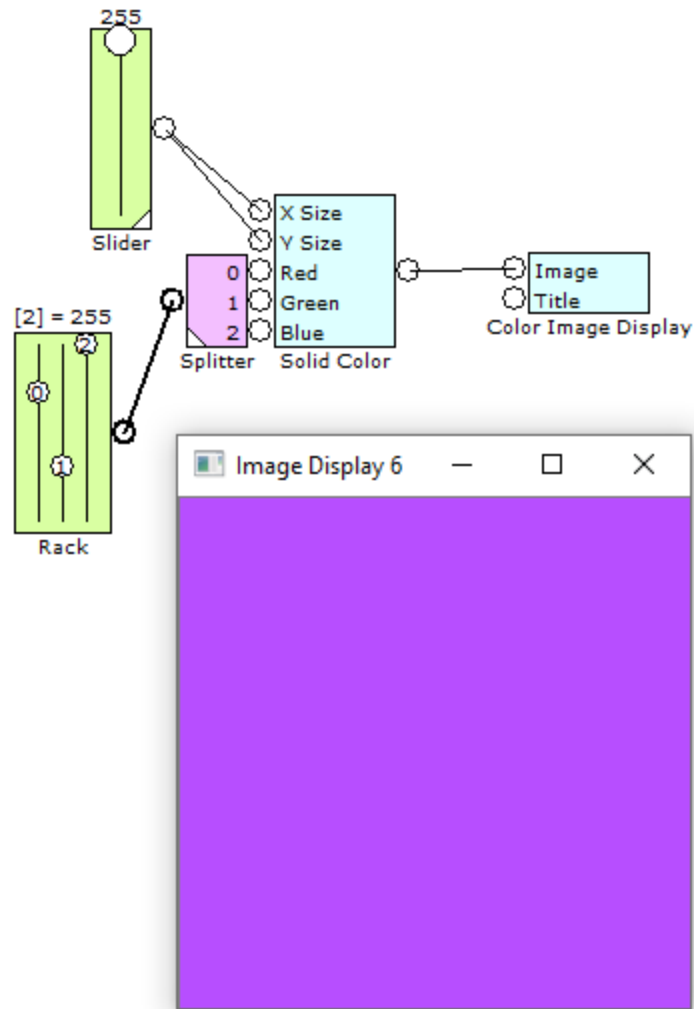
Snapped outputs indicate the part is in place.

Filled outputs indicate the target position has been filled. Controls - column 3



# Solid Color

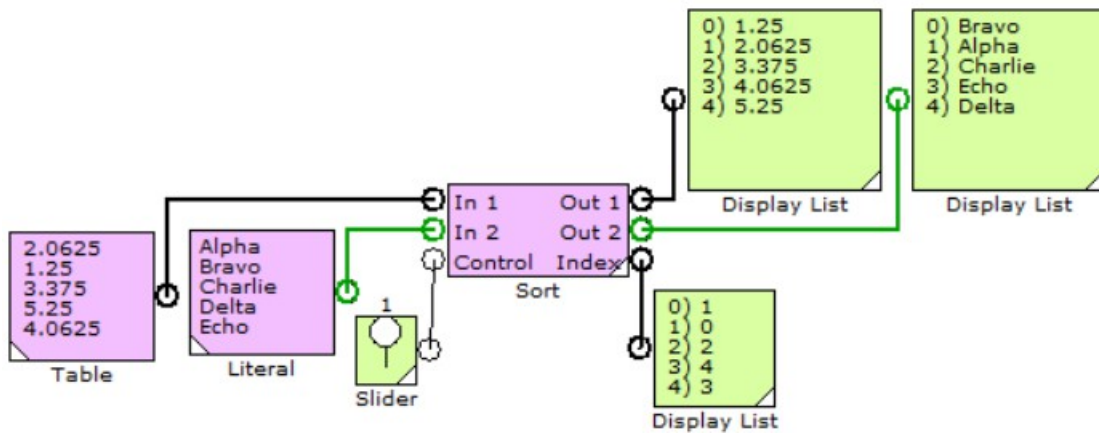
The Solid Color component creates an image with a single color. Images - column 1





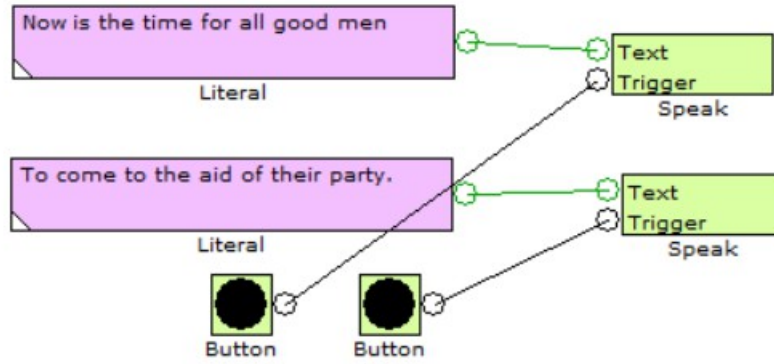
# Sort

The Sort component sorts the input data. The input(s) can be arrays of numbers or strings. The Array at the control input indicates how each data input should be sorted. A single number at the control input indicates which input is primary. A negative number indicates sorting in reverse. If an array is at the control input, the second value indicates which data input is used as a secondary sort when primaries are identical. Arrays – column 1



# Speak

The Speak component uses text to speech technology to say the string at it's input. Controls – column 4



# 3D Sphere

The 3D Sphere draws a sphere in the 3D view window. The X, Y, and Z inputs set the position and size

**3D Sphere**

Latitude Steps:

First Latitude Step:

Last Latitude Step:

Longitude Steps:

First Longitude Step:

Last Longitude Step:

X Offset:

Y Offset:

Z Offset:

Faceted Surface  
 Smooth Surface

**Pin options:**

X Position  Y Position  Z Position

X Rotate  Y Rotate  Z Rotate

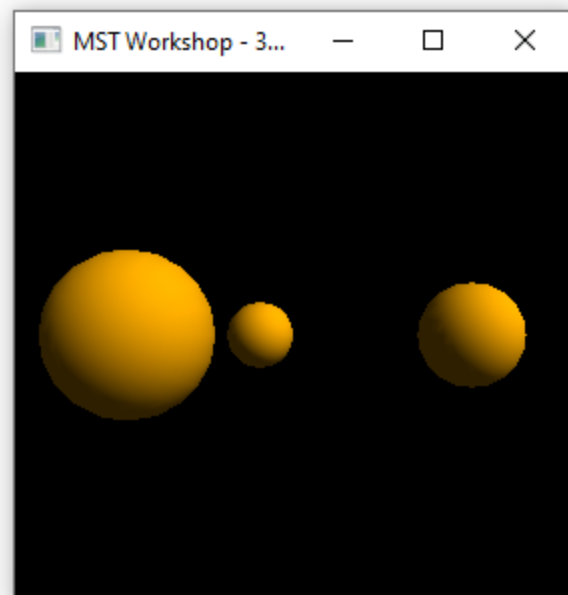
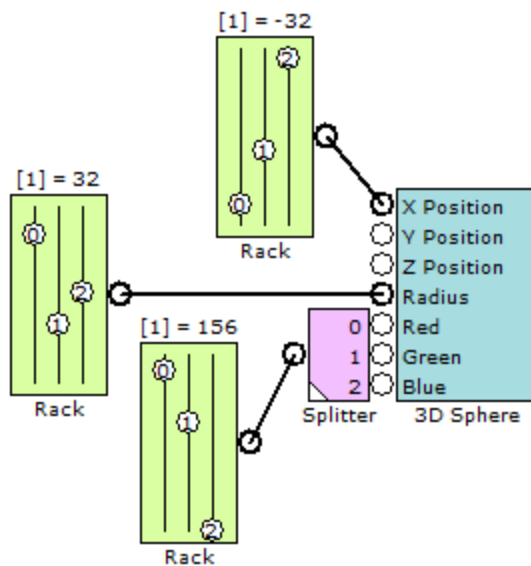
Radius

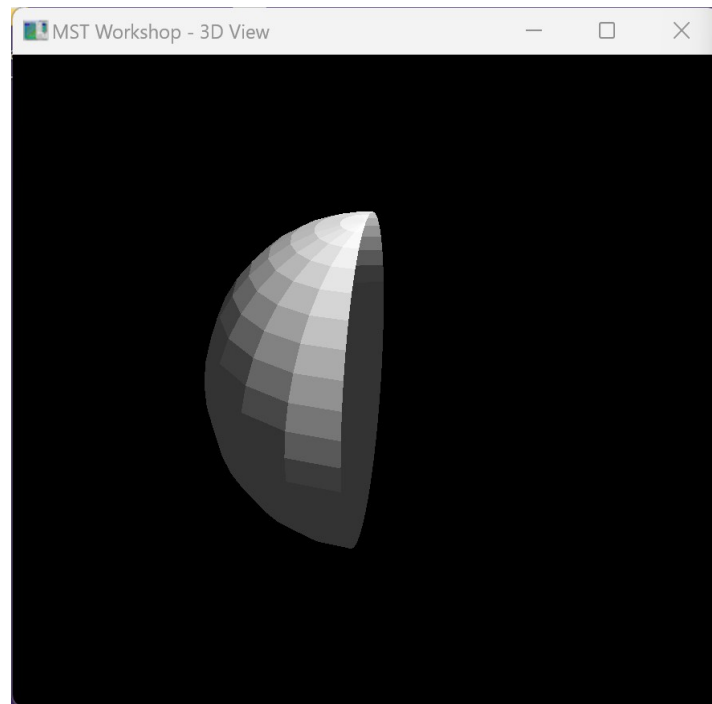
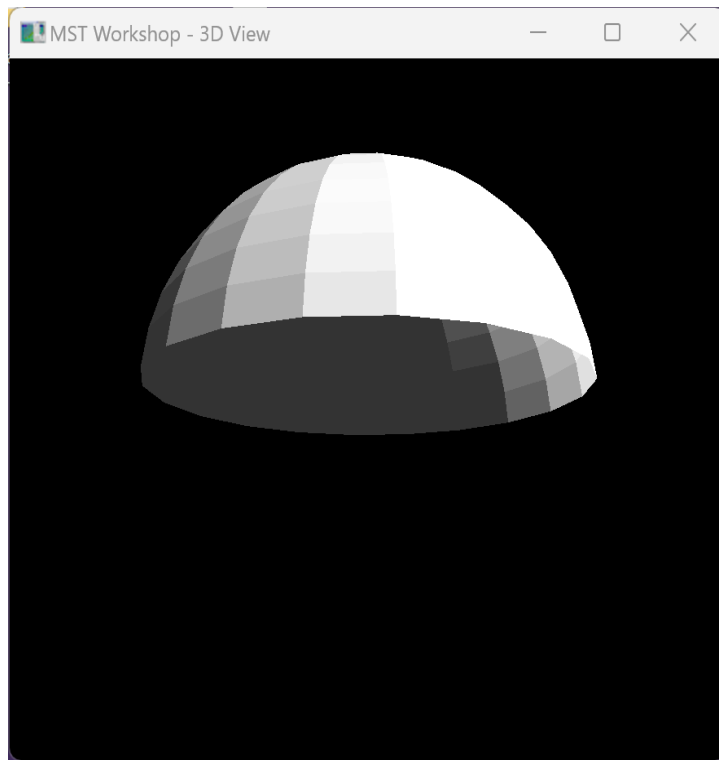
Red  Green  Blue

Alpha

[4x4]

of the sphere. The Affine [4x4] array input modifies the size, position and orientation. 3D - column 4





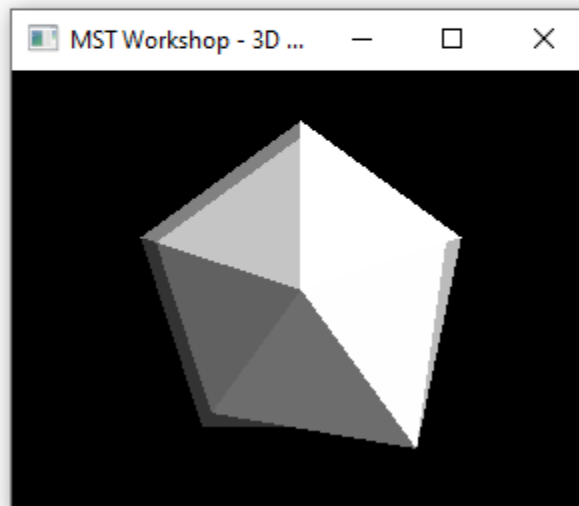
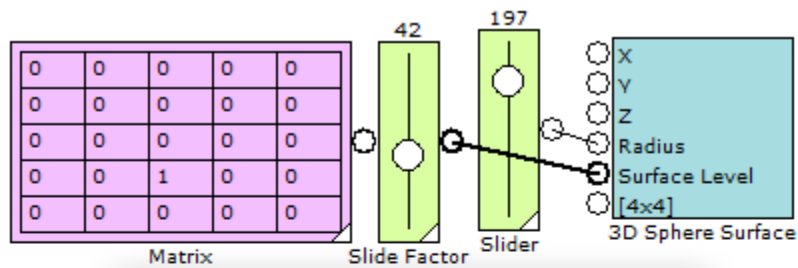
### 3D Sphere Surface

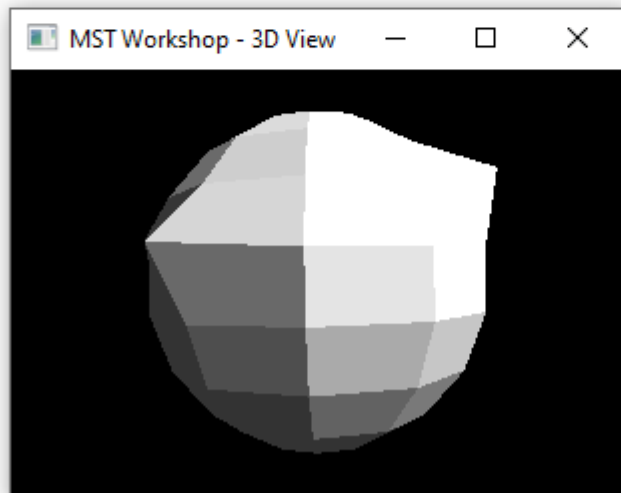
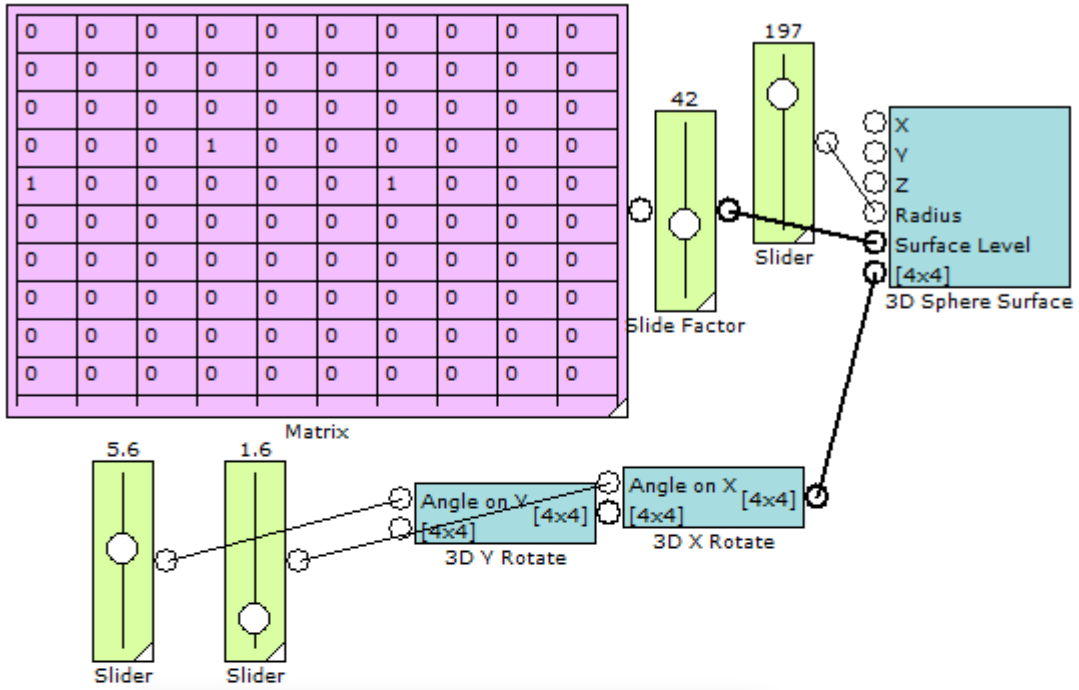
The 3D Sphere Surface component draws a sphere with an embossed surface in the 3D view window. The Matrix and Mat Row Size inputs set the 2D data to be plotted. The Affine [4x4] array input modifies the size, position and orientation. 3D - column 5

Pin options:

<input checked="" type="checkbox"/>	X	Y	Z
<input checked="" type="checkbox"/>	Radius		
<input type="checkbox"/>	Red	Green	Blue
<input type="checkbox"/>	Alpha		
<input checked="" type="checkbox"/>	Surface Level		
<input checked="" type="checkbox"/>	[4x4]		

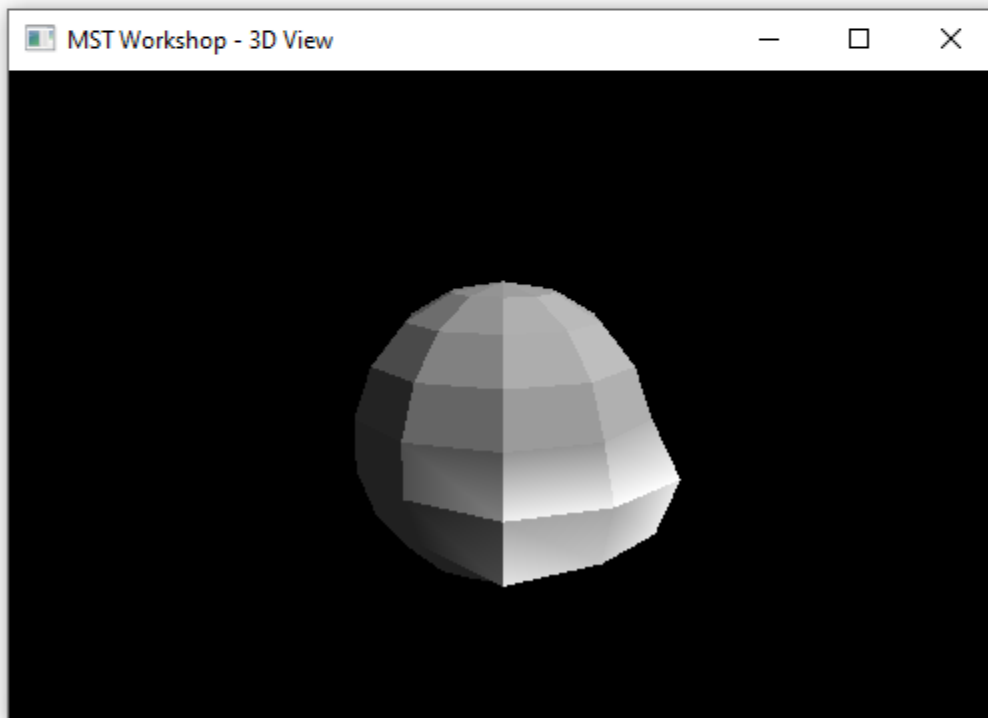
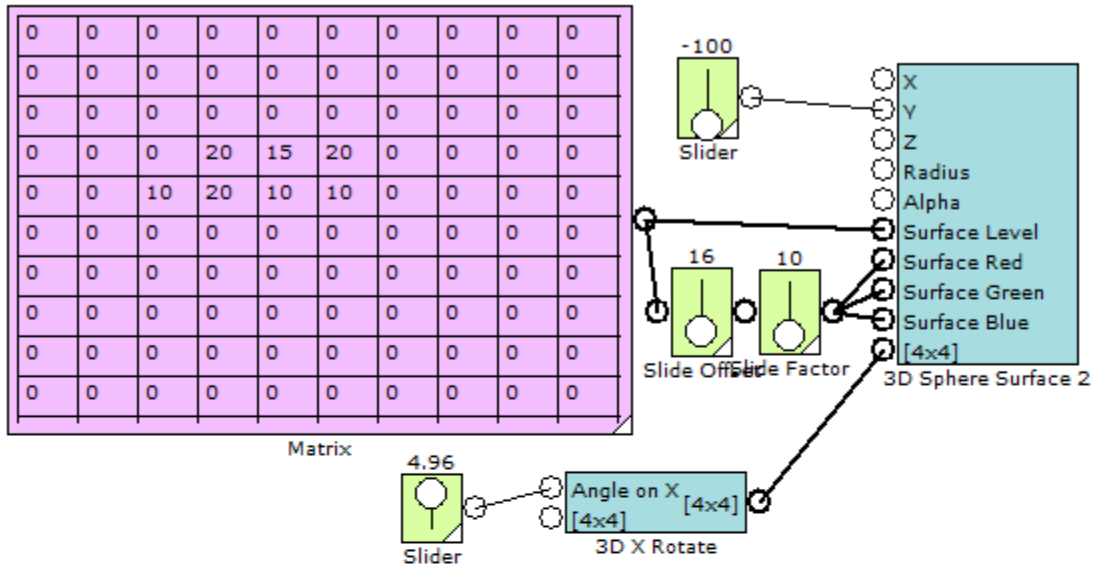
OK Cancel





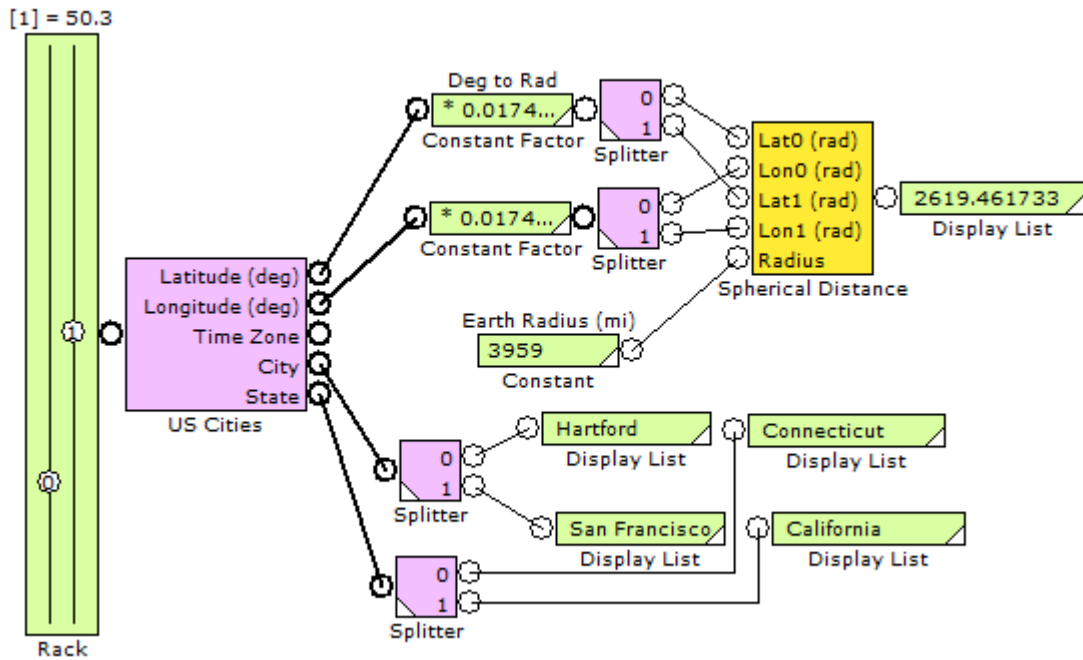
## 3D Sphere Surface 2

The 3D Sphere Surface 2 component draws a sphere with an embossed and color surface in the 3D view window. The Matrix and Mat Row Size inputs set the 2D data to be plotted. The Affine [4x4] array input modifies the size, position and orientation. 3D - column 5



# Spherical Distance

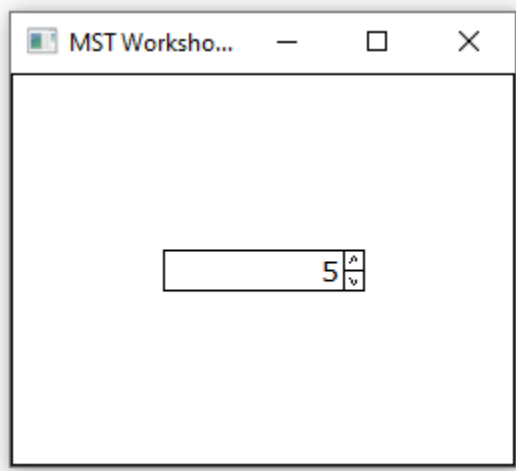
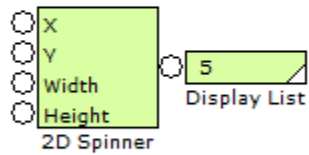
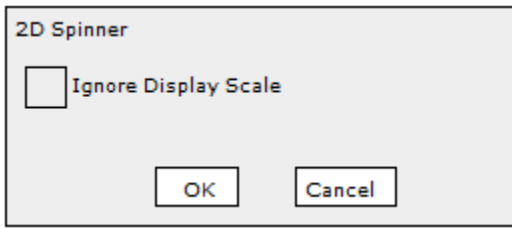
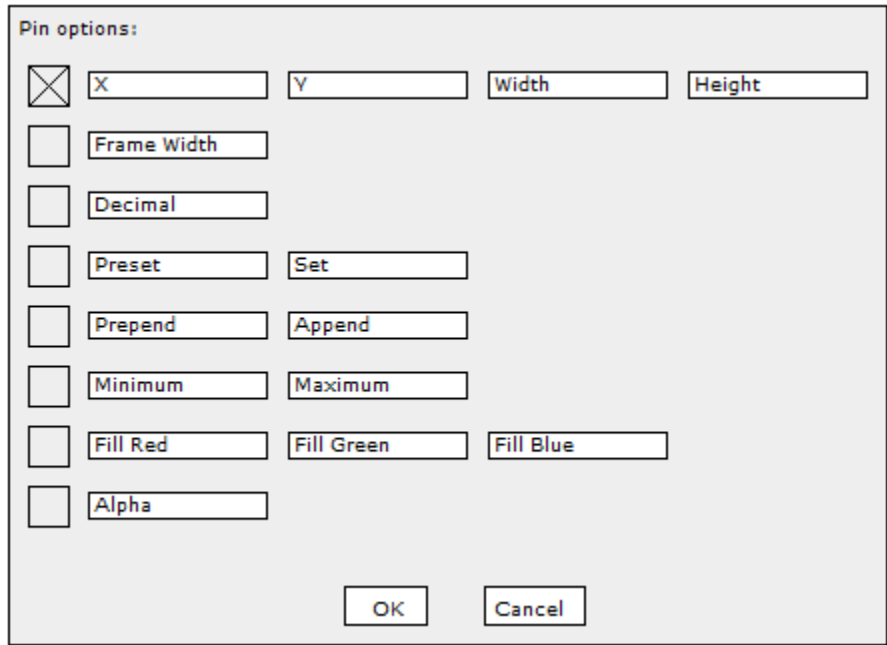
The Spherical Distance component calculates the distance between two points on a sphere.  
Functions - column 8





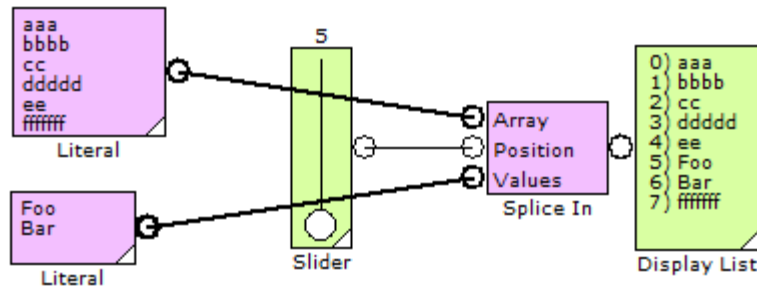
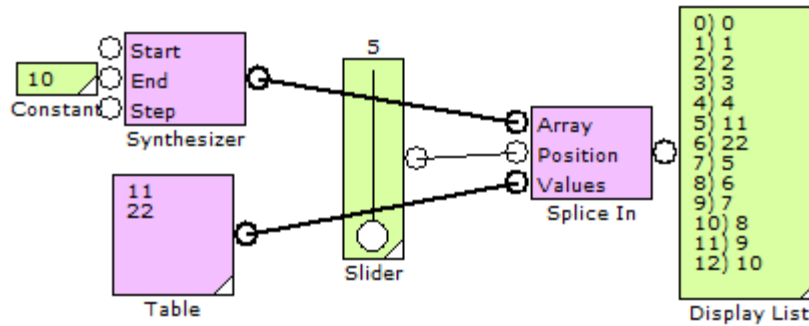
# 2D Spinner

The 2D Spinner is used to enter a number on the 2D canvas. Controls - column 2



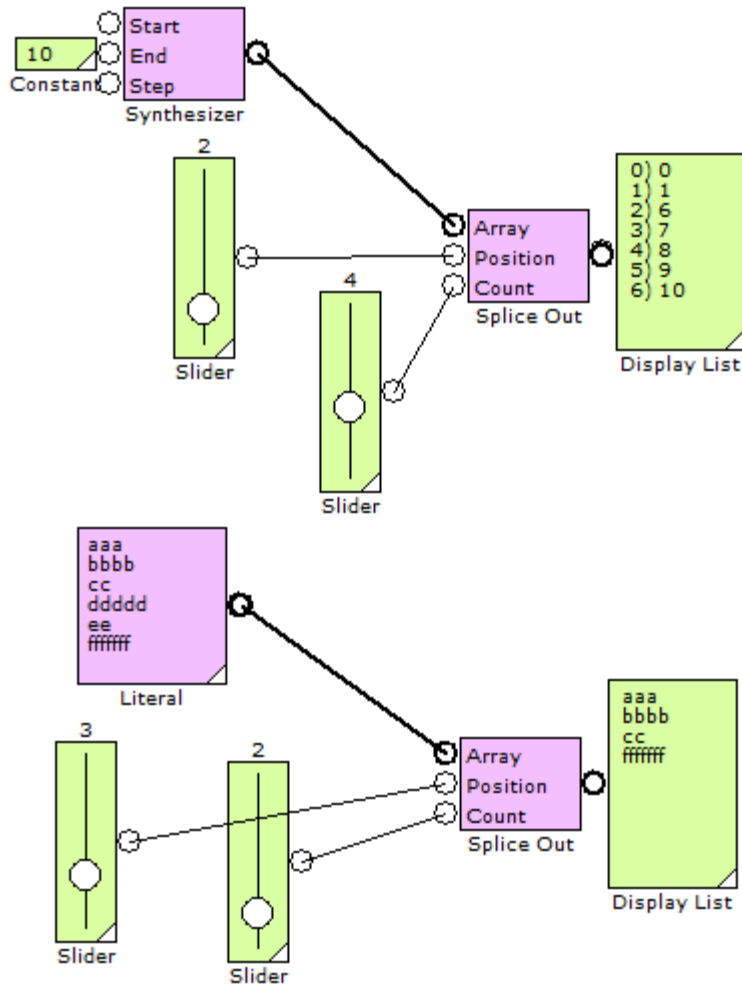
# Splice In

The Splice In component inserts values into an array at the desired position.      Arrays - column 1



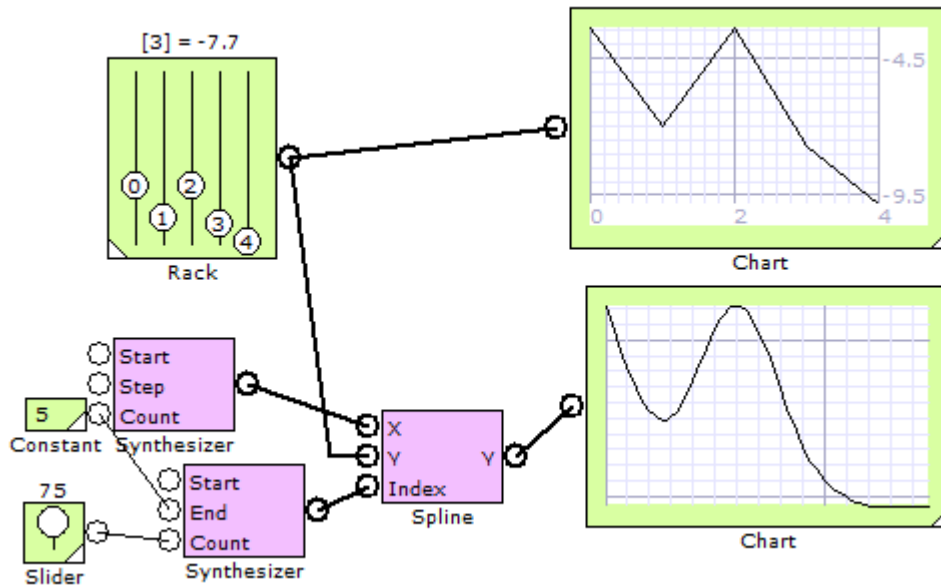
# Splice Out

The Splice Out component removes values from the specified position. Arrays - column 1



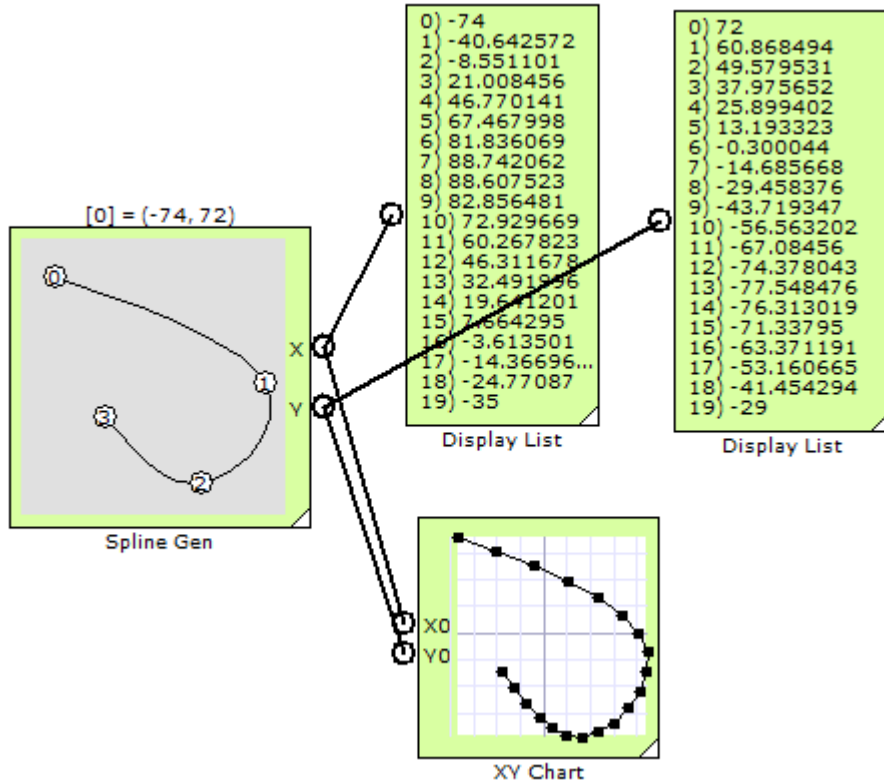
# Spline

The Spline component interpolates a spline curve based on the input array. It outputs the Y value(s) at the positions given by the Index(s) Arrays - column 2



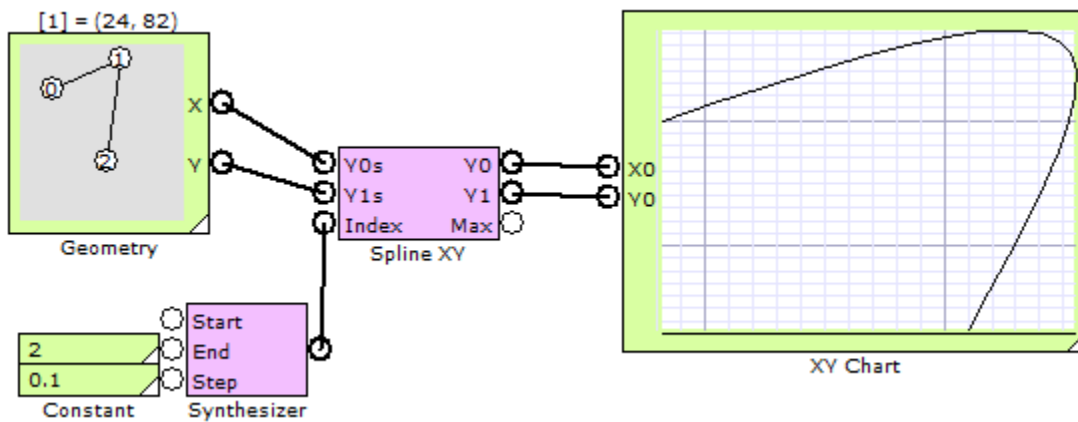
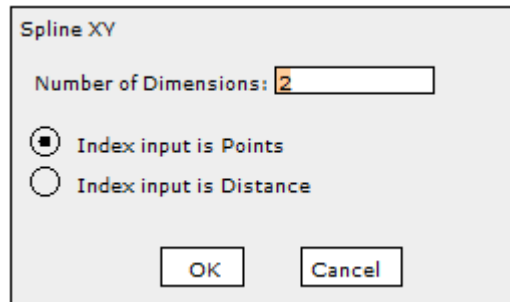
# Spline Gen

The Spline Gen lets you generate a smooth curve from a small number of control points in X and Y.  
Controls - column 1



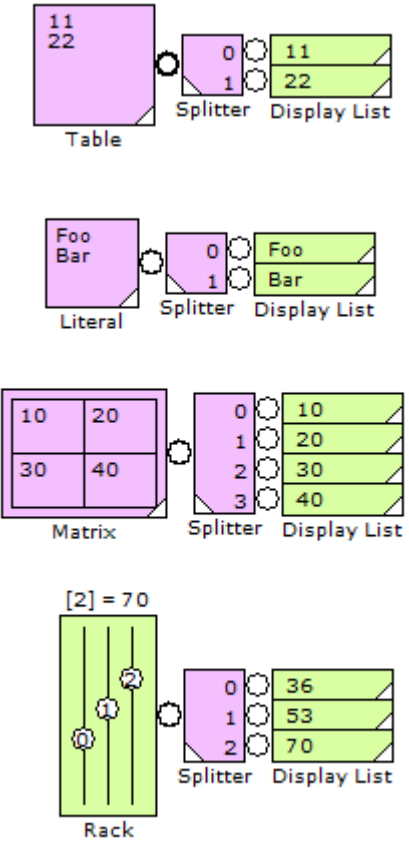
## Spline XY

The Spline XY component interpolates a spline curve based on the input array. It outputs the X and Y value(s) at the positions given by the Index(s). The number of dimensions can be expanded. Arrays - column 2



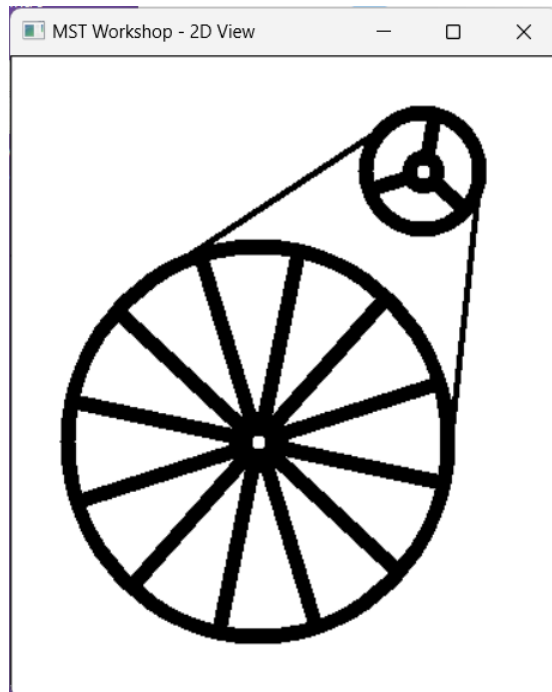
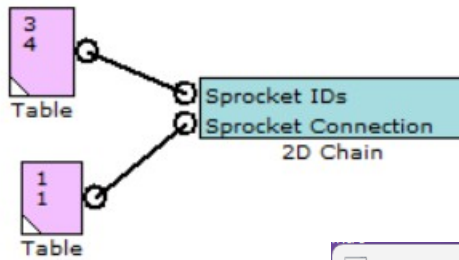
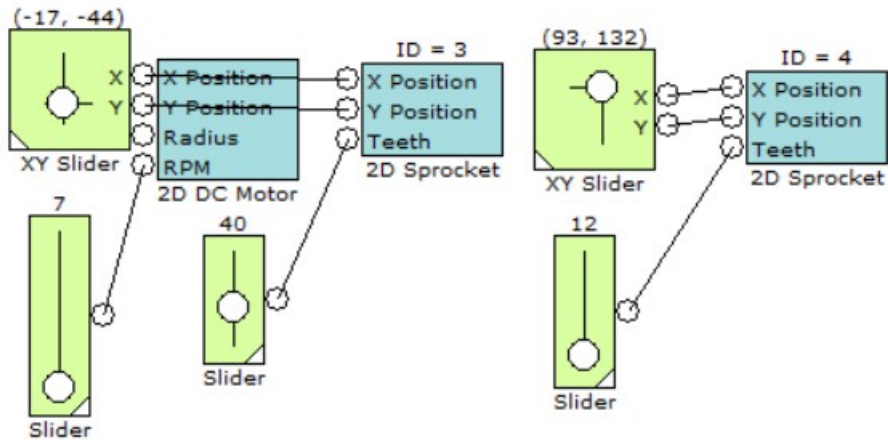
# Splitter

The Splitter component breaks the input into separate outputs. You can set the number of outputs desired.      Arrays - column 1



# Sprocket

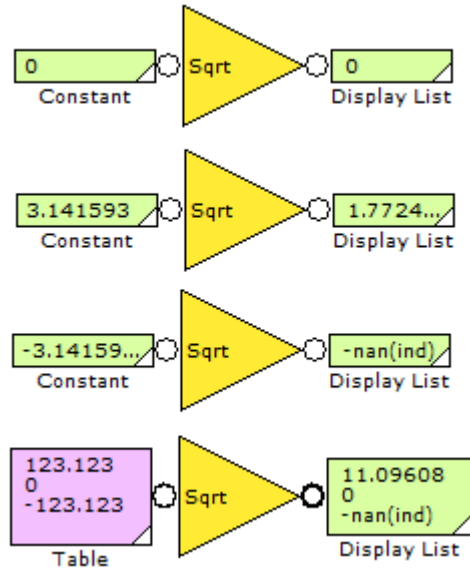
The 2D Sprocket component creates a sprocket that can attach to a Chain or other simulation objects like Motors. Place the Sprocket at the same X, Y position as a motor to connect the Sprocket to the motor. Use the Chain component to hook two or more Sprockets together. 2D - column 3





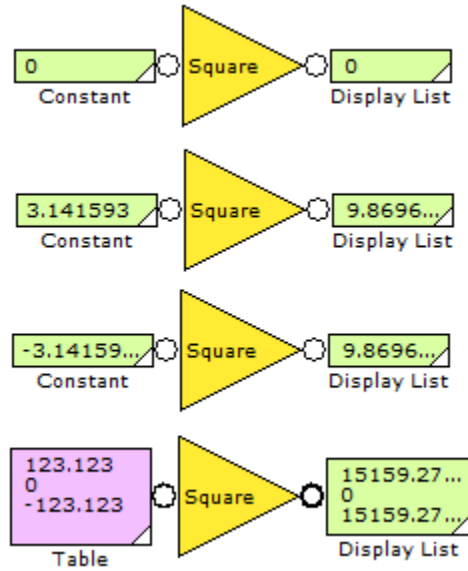
# Sqrt

The Sqrt component outputs square root of the input value. Input can be scalar or an array.  
Functions - column 1



# Square

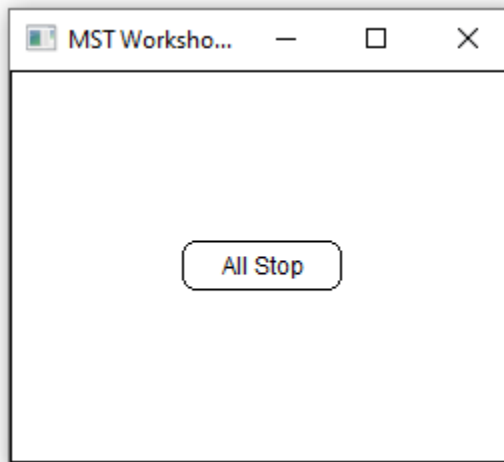
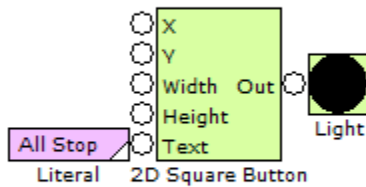
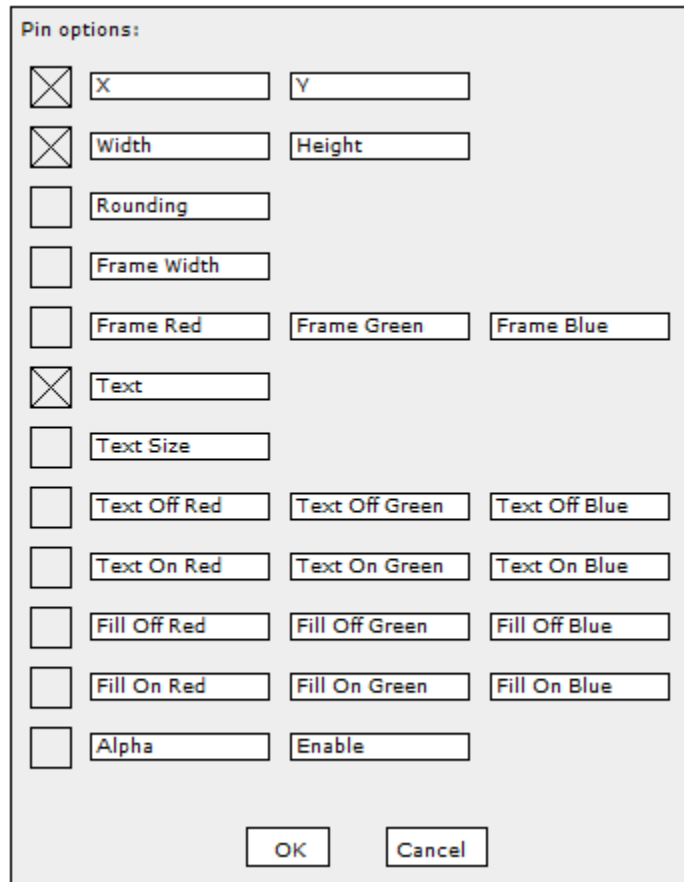
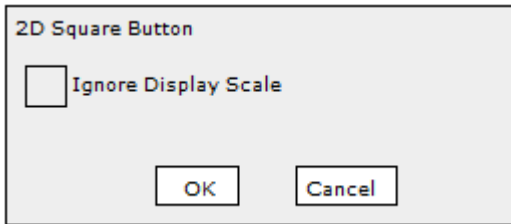
The Square component multiplies the input value times itself. Input can be scalar or an array.  
Functions - column 1



# 2D Square Button

The 2D Square Button is used to trigger an event by clicking on it.

Controls - column 1



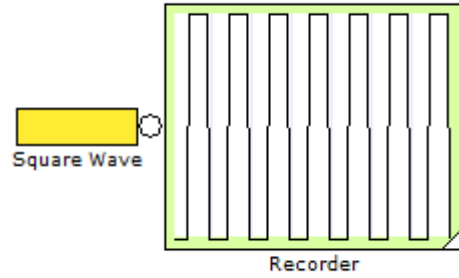
# Square Wave

The Square Wave component outputs a square wave when the clock is running. Functions - column 8

Square Wave

Output Period:

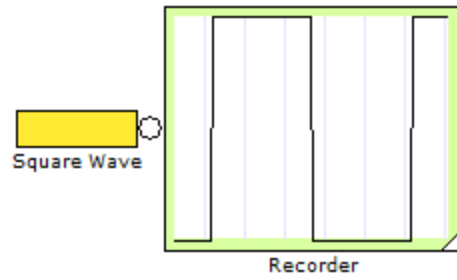
Output Maximum:



Square Wave

Output Period:

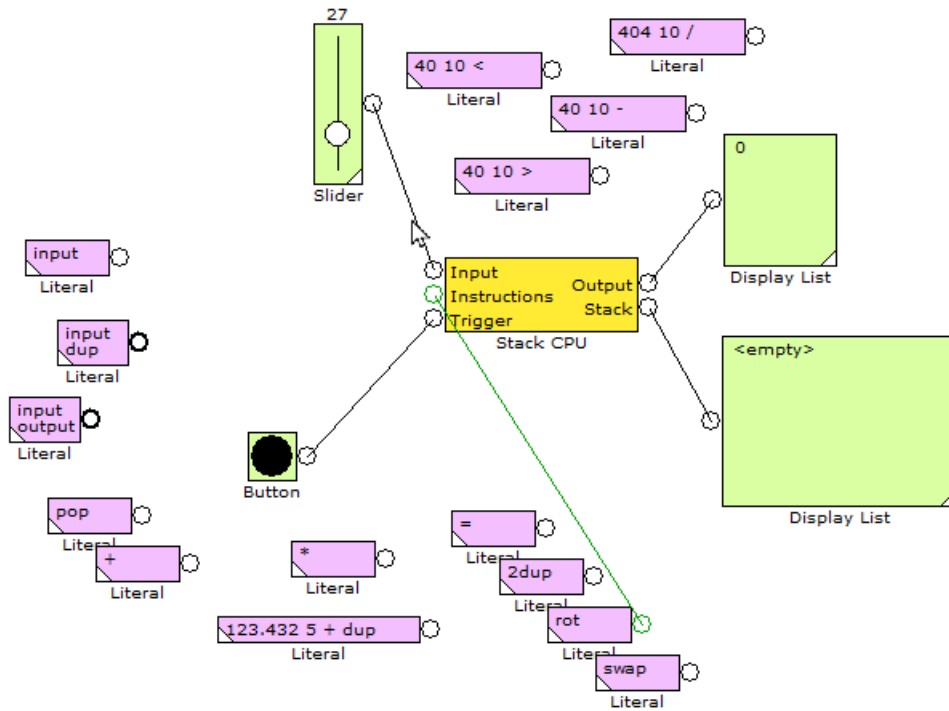
Output Maximum:



# Stack CPU

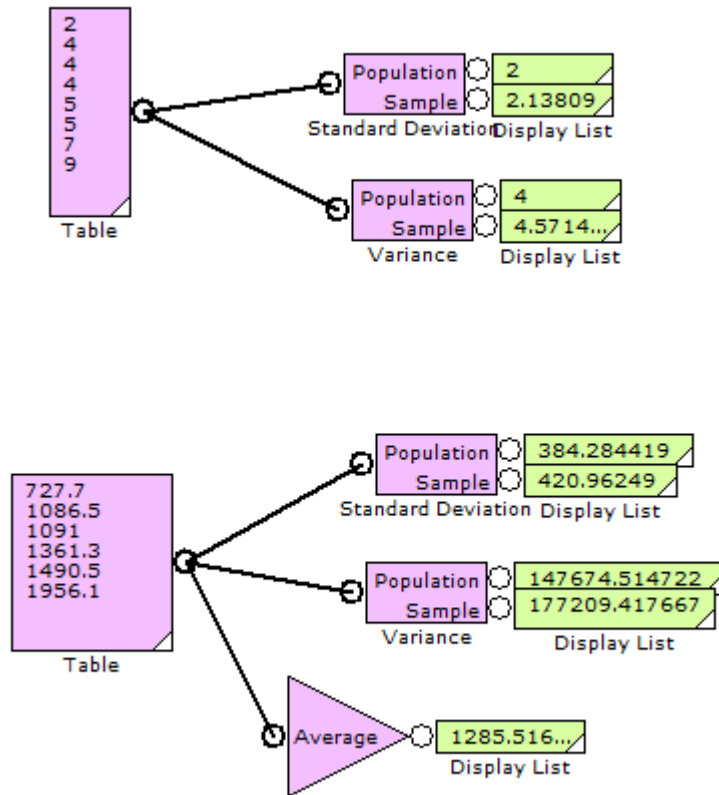
The Stack CPU component is a stack based computer that you can program in a language similar to Forth. Functions - column 3

NOTE: Stack CPU is just in its beta phase and only understands a few words.



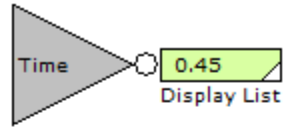
# Standard Deviation

The Standard Deviation component scans the input array and calculates the standard deviation of those values. Arrays - column 3



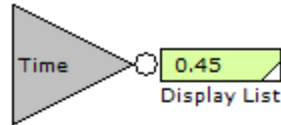
# Start Clock

The Start Clock component starts the solution clock when the trigger goes from false to true.  
Controls - column 3



# Step Clock

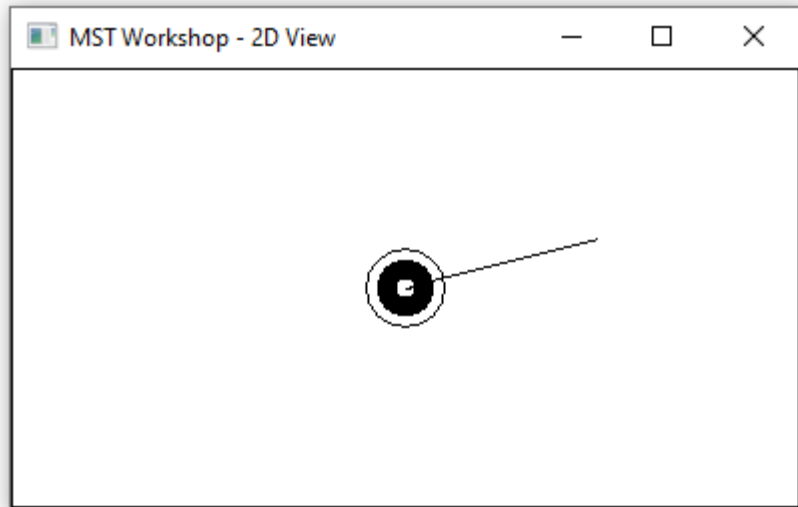
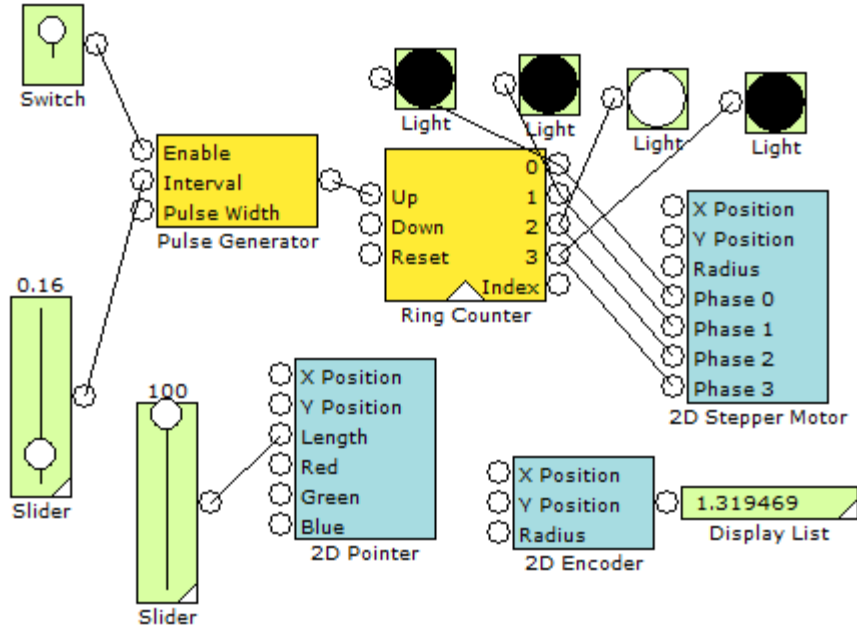
The Step Clock component steps the solution clock one time interval when the trigger goes from false to true. Controls - column 3





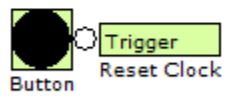
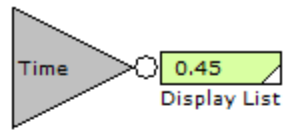
## 2D Stepper Motor

The Stepper Motor creates a motor that can drive other simulation objects like Gears. Place a Gear at the same X, Y position to connect the motor to the gear. 2D - column 3



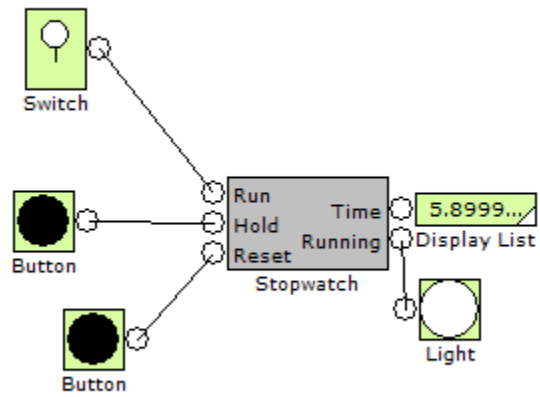
# Stop Clock

The Stop Clock component stops the solution clock when the trigger goes from false to true.  
Controls - column 3



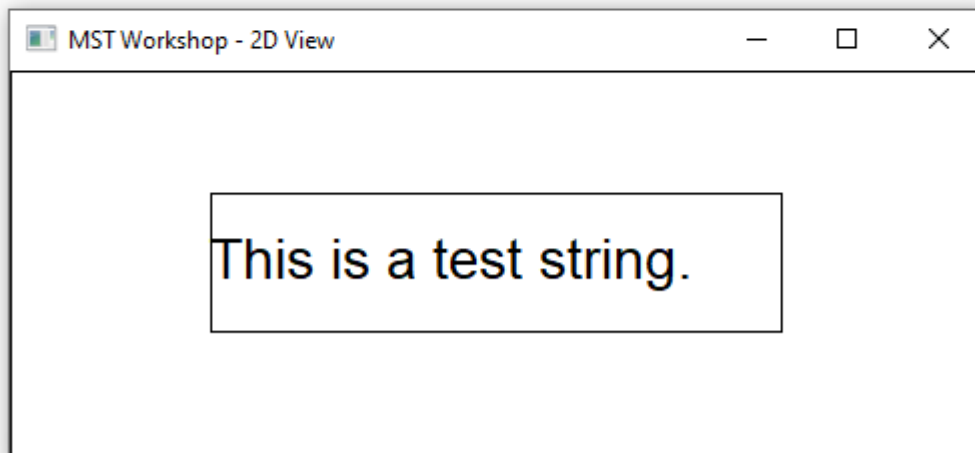
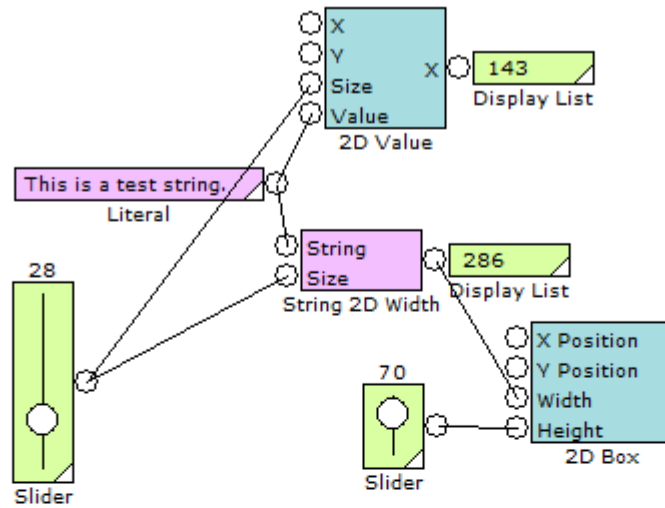
# Stopwatch

The Stopwatch component starts when the Run input goes high and runs until the Run input goes low. It can hold when the Hold input is high. It runs on the workspace clock. Functions - column 7



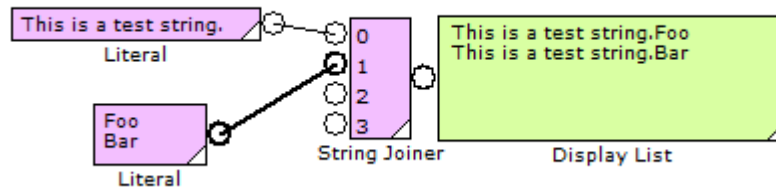
# String 2D Width

The String 2D Width component measures the width of the input string in 2D pixels. Arrays - column 6



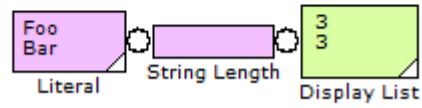
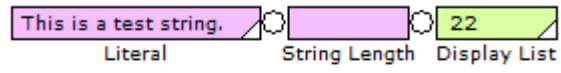
## String Joiner

The String Joiner component combines all the values for all its inputs into a single output. You can set the number of inputs. Arrays - column 6



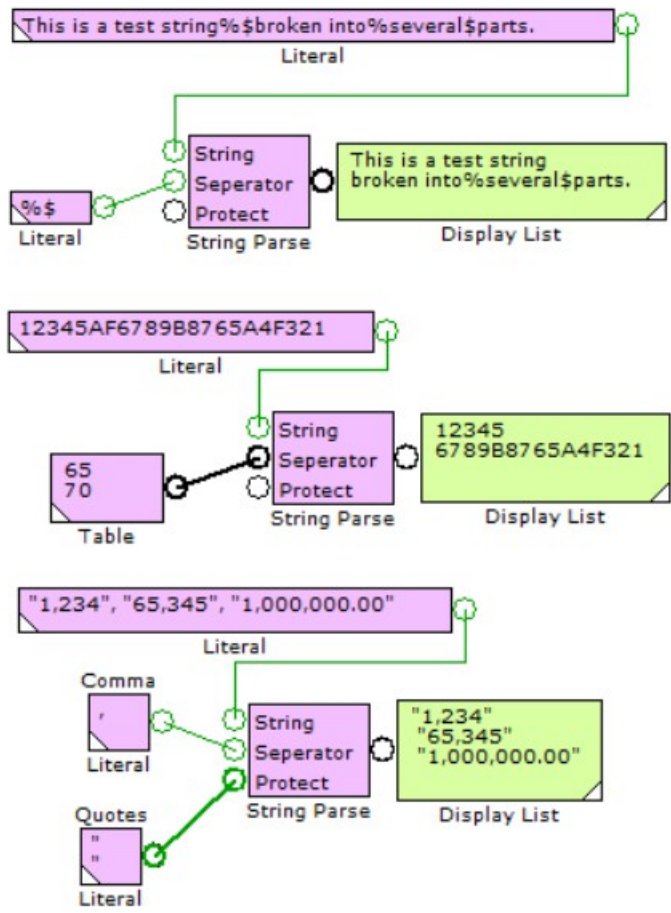
# String Length

The String Length component outputs the number of characters in the string at the input. Arrays - column 6



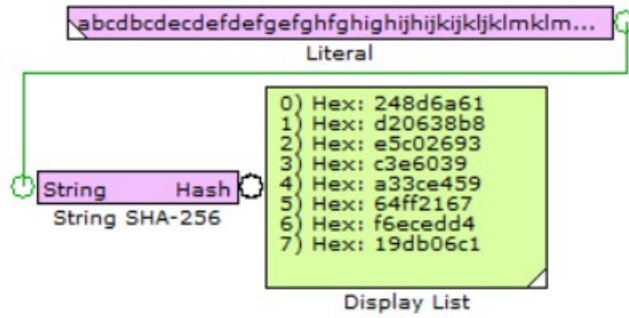
# String Parse

The String Parse component takes the string input and breaks it into array elements based on the separator. The separator can be either a string or an array of numerical values of ASCII characters. You can also specify a pair of strings to Protect ranges of characters from being parsed. Protect examples: " and "; [ and ]; or ( and ) . Arrays - column 6



# String SHA-256

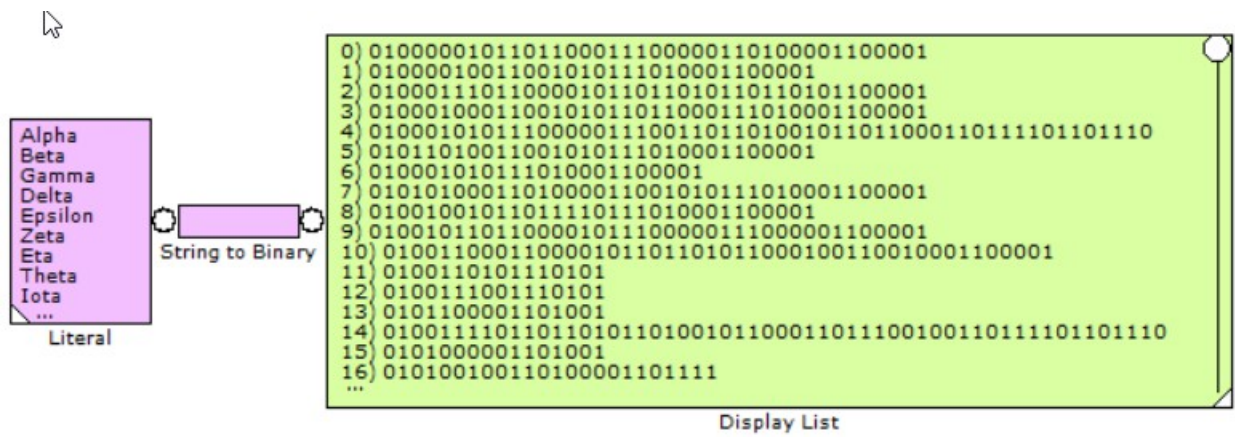
The String SHA-256 component calculates the US Secure Hash Algorithm of the input String.





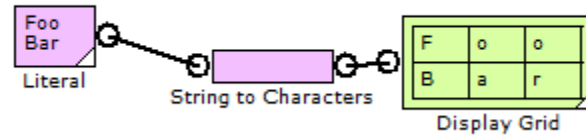
# String to Binary

The String to Binary component converts a string to its Binary representation. Arrays – column 3



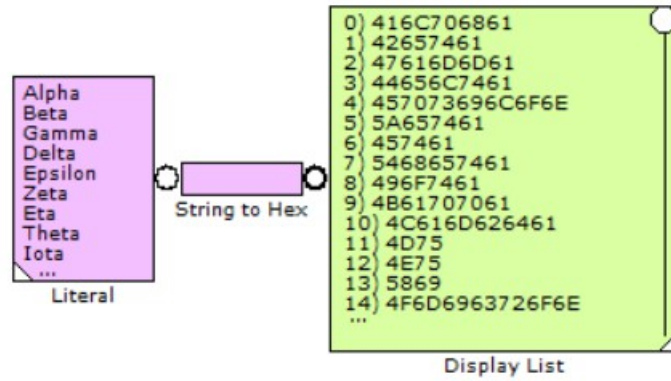
## String to Characters

The String to Characters component outputs an array of characters based on the input string. Arrays  
- column 7



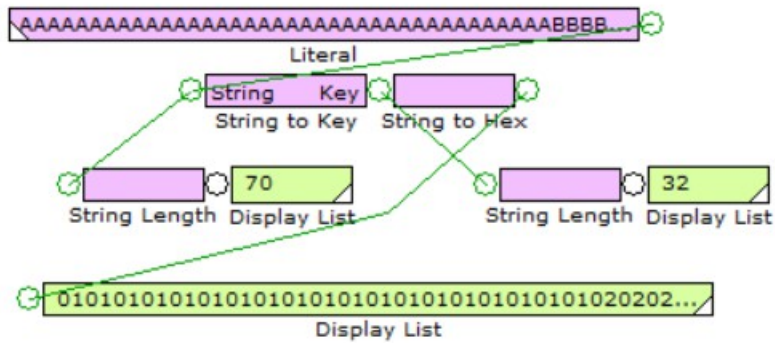
## String to Hex

The String to Hex component converts a string to its Hexadecimal representation. Arrays – column 3



# String to Key

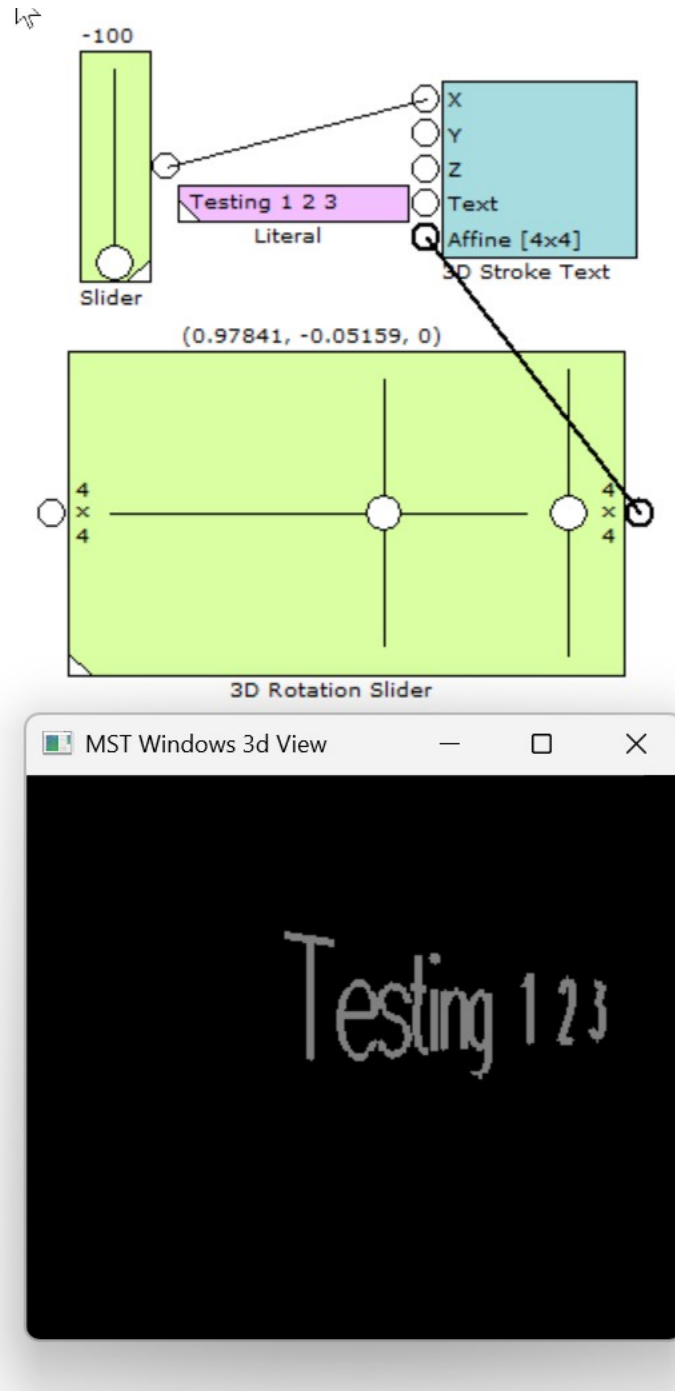
The String to Key component takes a string and packs it into a key. Arrays – column 3



## 3D Stroke Text

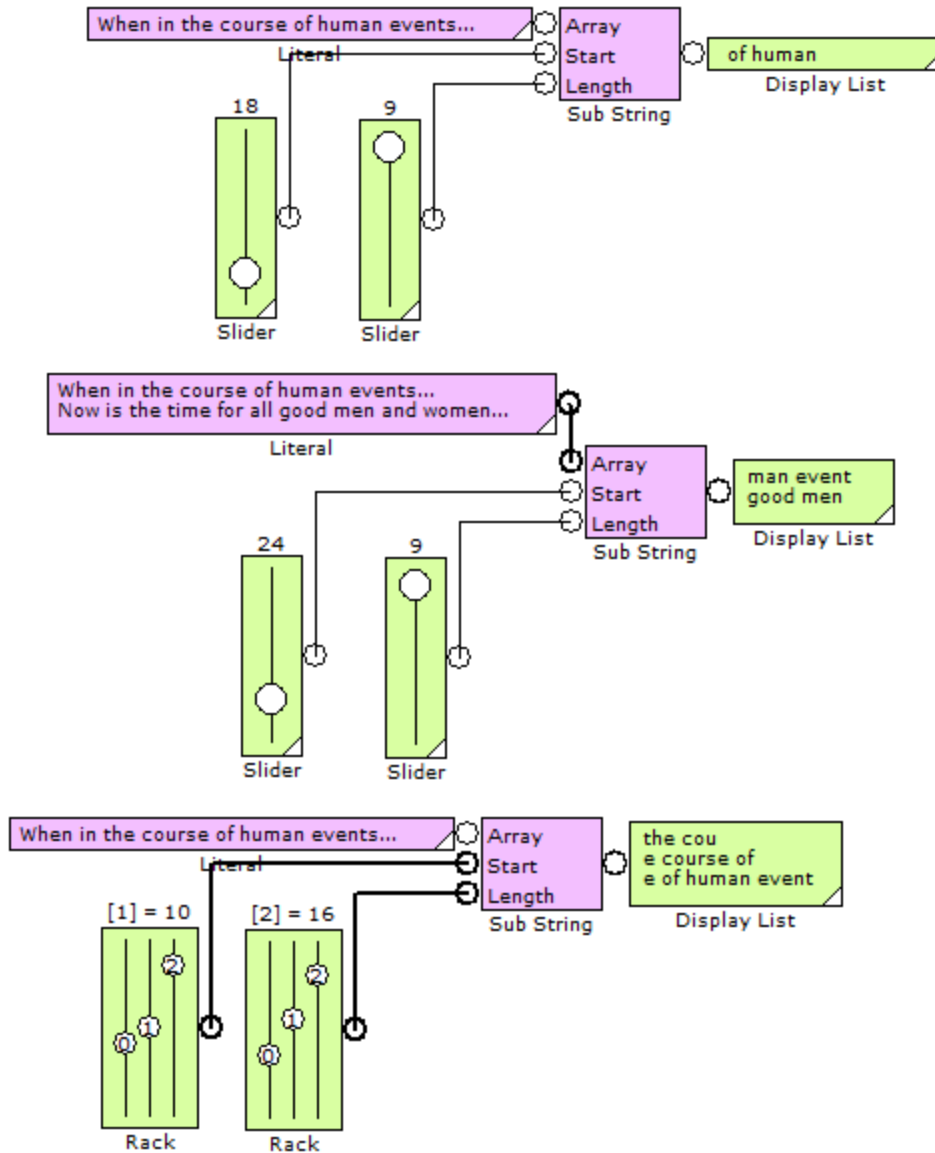
The 3D Stroke Text draws a string of text in the 3D view window. The X, Y, and Z inputs set the position and size of the text. The Affine [4x4] array input modifies the size, position, and orientation.

3D - column 1



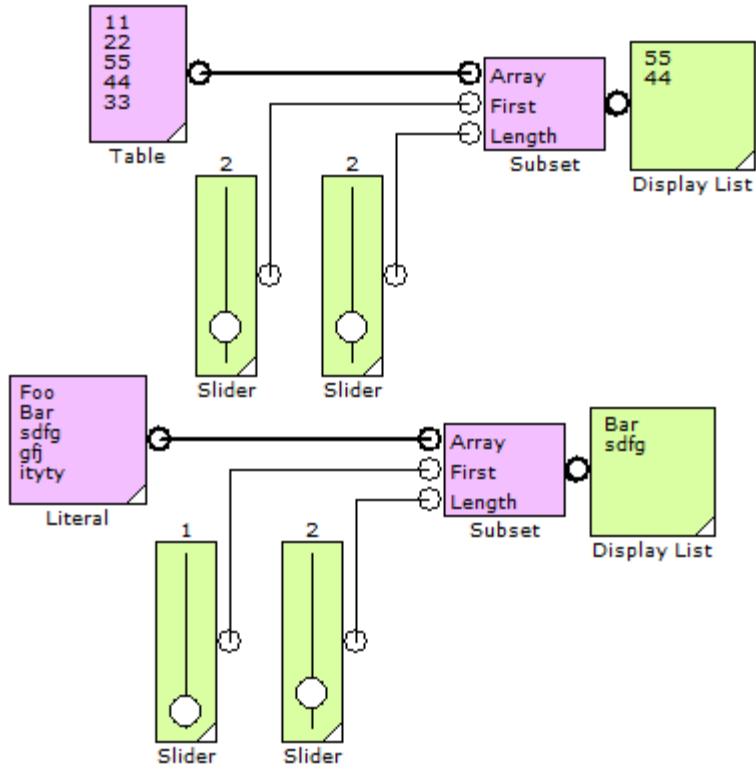
# Sub String

The Sub String component outputs a portion of the input string. Arrays - column 6



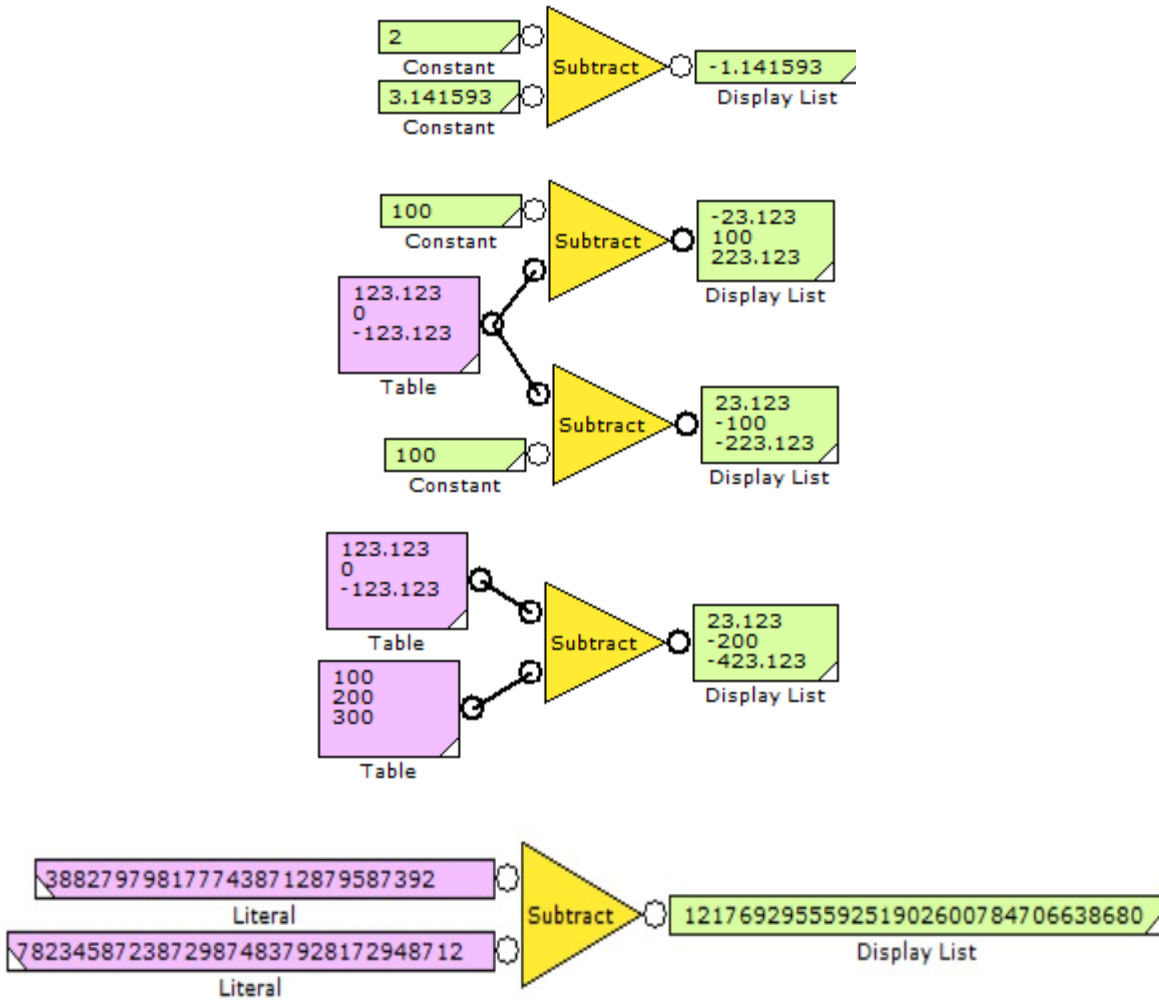
# Subset

The Subset component outputs a portion of the values in an array. Input the number of the first value and the number of values you want at the output. Arrays - column 1



# Subtract

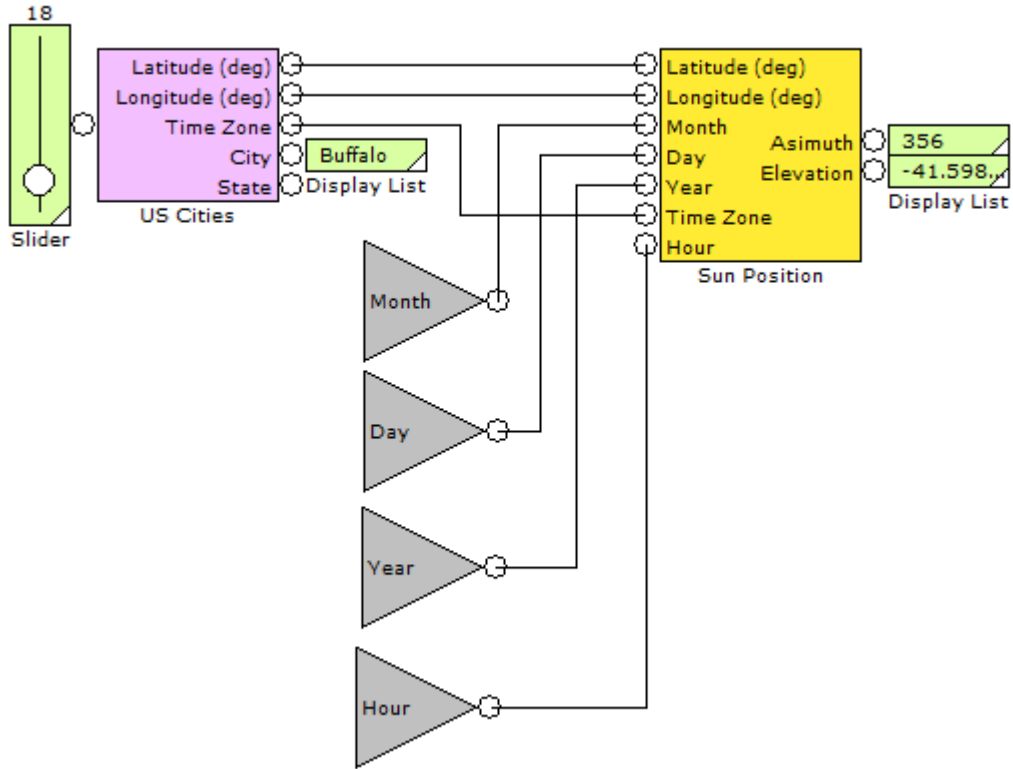
The Subtract component subtracts the inputs. Inputs can be scalar and /or arrays. Functions - column 1





# Sun Position

The Sun Position component calculates the Elevation and Azimuth of the Sun. Latitude and Longitude are in degrees. North Latitude is positive. West Longitude is negative. The outputs are in degrees.

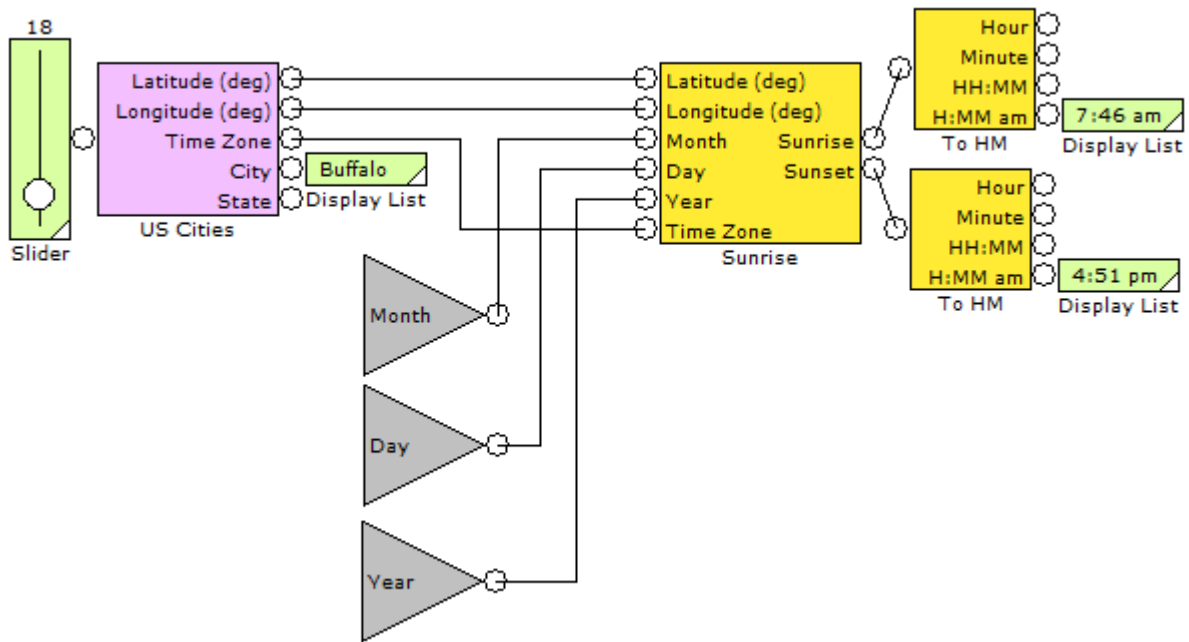


Functions - column 9

# Sunrise

The Sunrise component calculates the time of Sunrise and Sunset. Latitude and Longitude are in degrees. North Latitude is positive. West Longitude is negative. The output is a fraction of a day.

Functions - column 9



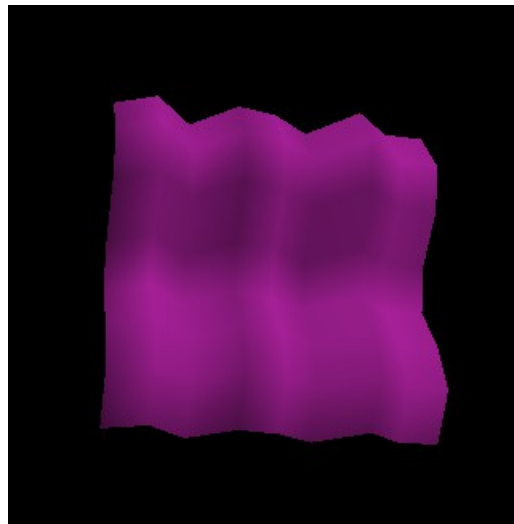
## 3D Surface

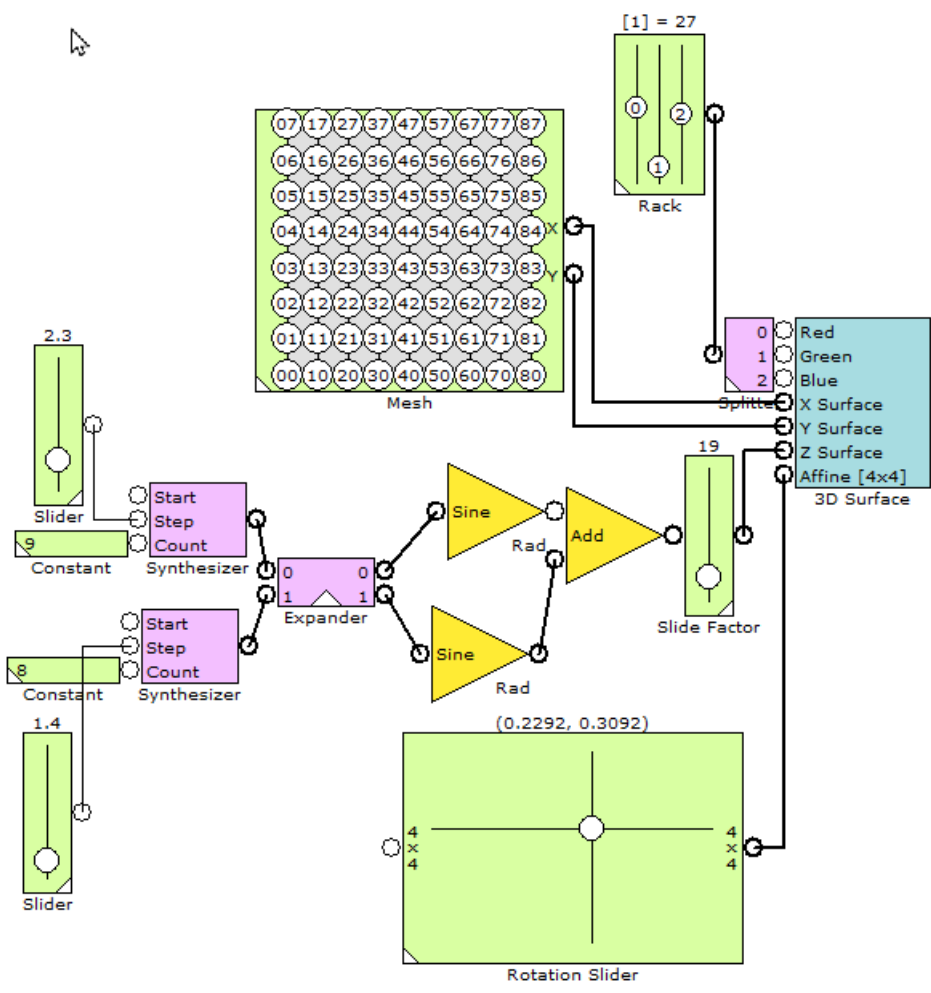
The 3D Surface component draws a surface in the 3D view window. The X, Y, and Z Matrix and Mat Row Size inputs set the 3D data to be plotted. The Affine [4x4] array input modifies the size, position and orientation. 3D - column 5

Pin options:

<input checked="" type="checkbox"/>	X	Y	Z
<input checked="" type="checkbox"/>	X Size	Y Size	Z Size
<input type="checkbox"/>	Red	Green	Blue
<input type="checkbox"/>	Alpha		
<input checked="" type="checkbox"/>	X Surface	Y Surface	Z Surface
<input checked="" type="checkbox"/>	[4x4]		

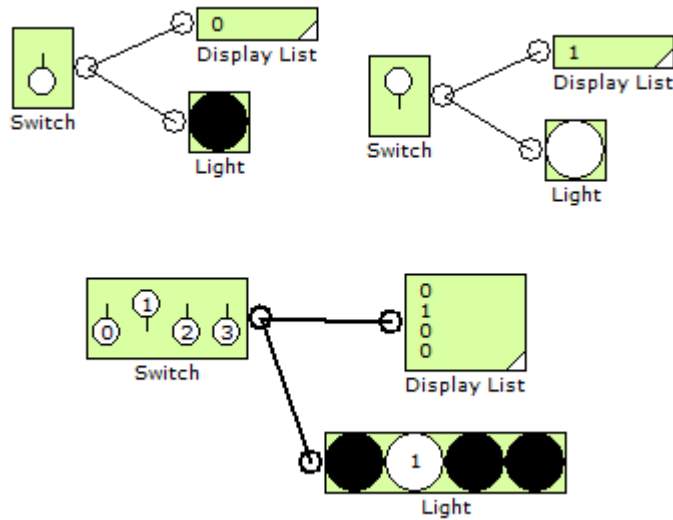
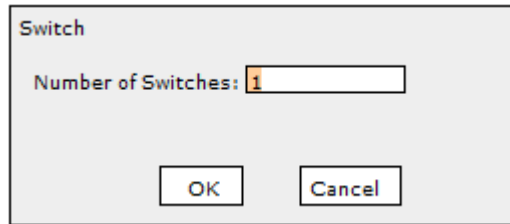
OK Cancel





# Switch

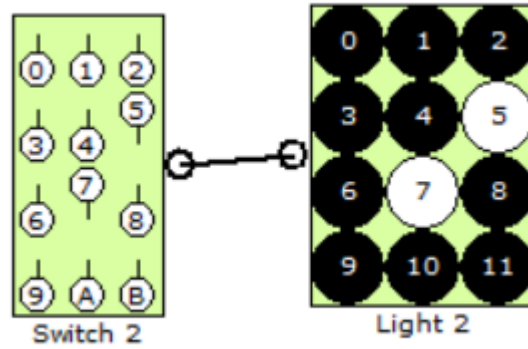
The Switch goes on when mouse moves the knob up, off when it is moved down. Note: This component can be operated using the wheel on the mouse. Controls - column 4



## Switch 2

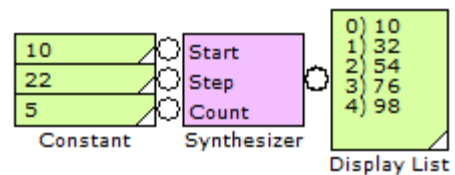
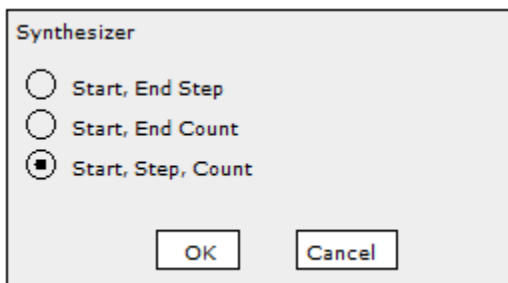
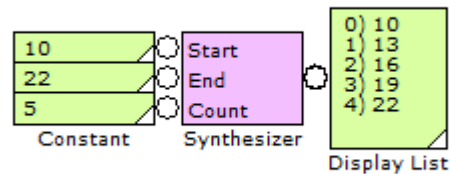
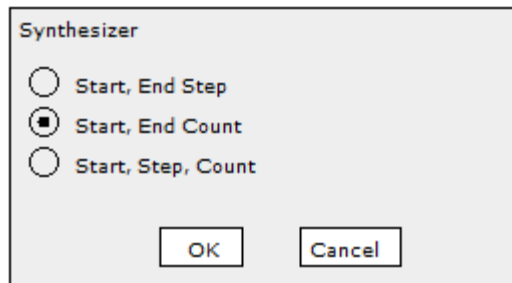
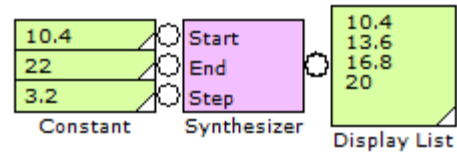
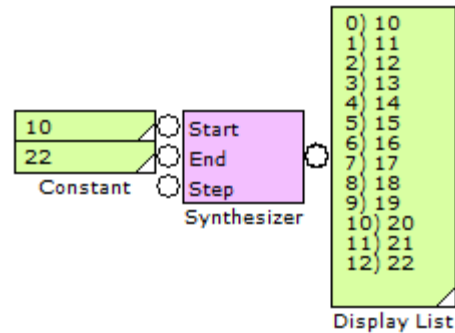
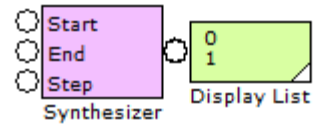
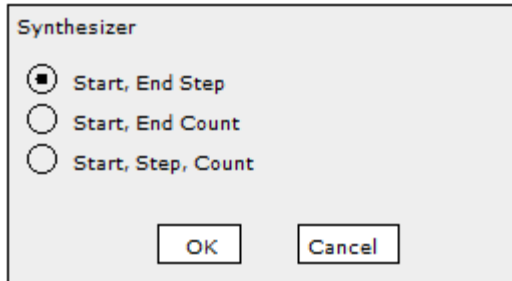
The Switch 2 goes on when the mouse moves the knob up, off when it is moved down. Switch 2's output is 2 dimensional. Note: This component can be operated using the wheel on the mouse.

Controls – column 4



# Synthesizer

The Synthesizer component creates an array of numbers. Arrays - column 1



# Table

The Table component is used to create a one dimensional table of numbers.      Arrays - column 1

<b>Move To Top</b>	<b>Delete</b>
<b>Move Up</b>	<b>Details...</b>
<b>Move Down</b>	<b>Duplicate</b>
<b>Move To Bottom</b>	<b>Edit Label...</b>
<b>Set Value Count...</b>	
<b>Edit Table Values...</b>	
<b>Load 3 pt. Gaussian</b>	
<b>Load 5 pt. Gaussian</b>	
<b>Load 7 pt. Gaussian</b>	
<b>Paste...</b>	
<b>Read File ...</b>	

Table

Number of Items

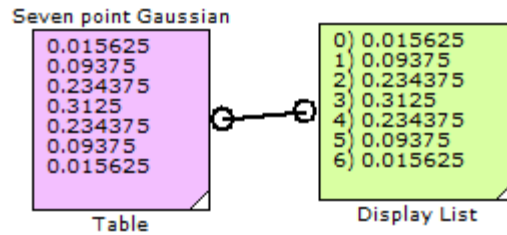
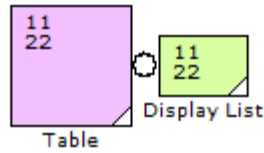
OK      Cancel

Table:

Value 0:

Value 1:

OK      Cancel

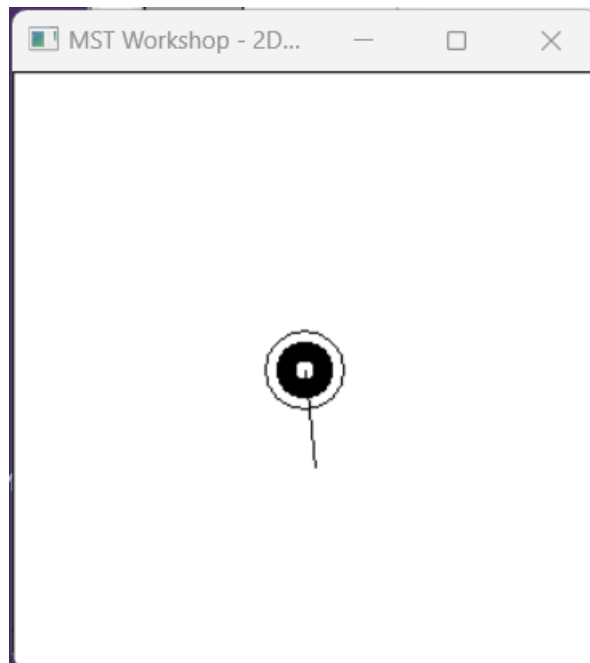
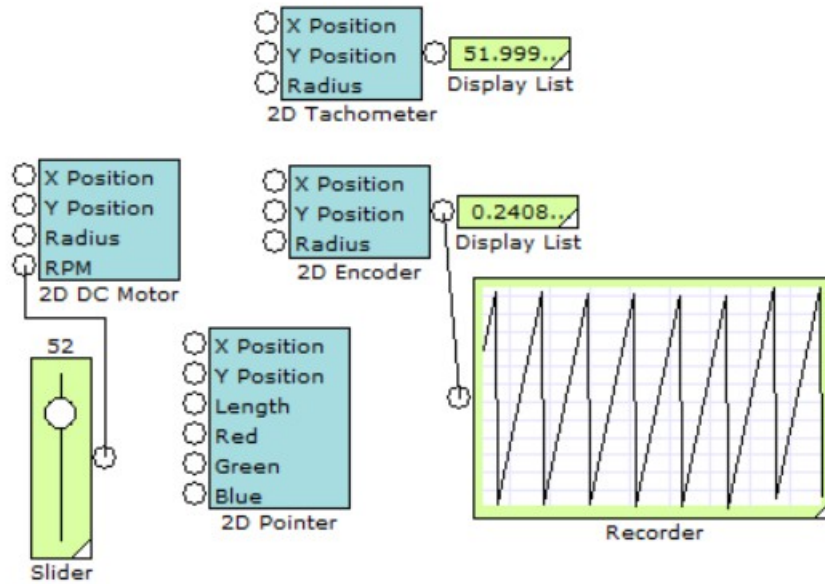




## 2D Tachometer

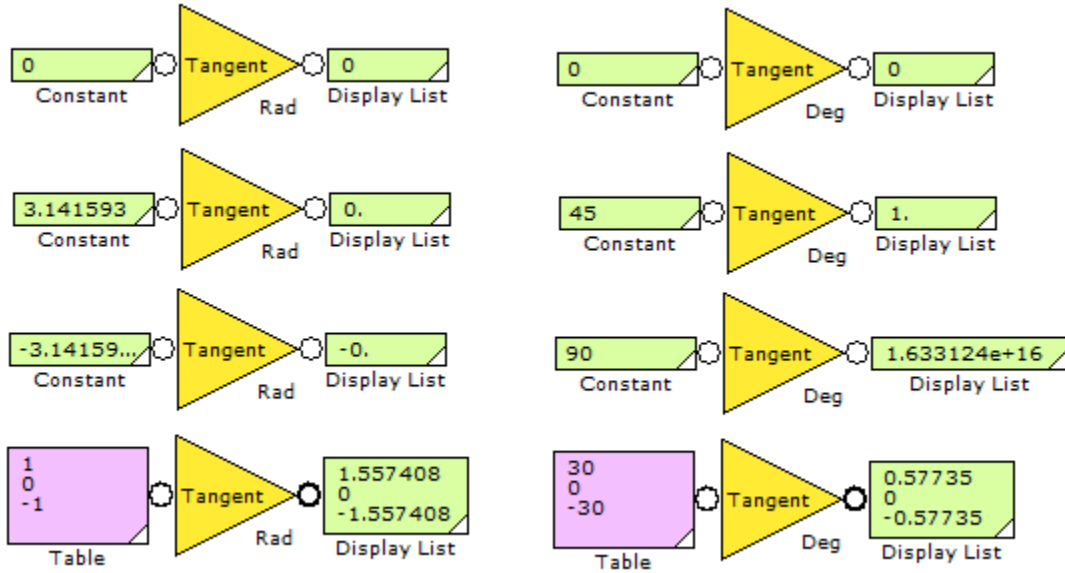
The 2D Tachometer outputs the speed of rotation of the shaft. You can attach it to Gears or other simulation objects like Motors. Place the tachometer at the same X, Y position as a gear to connect them.

2D – column 3



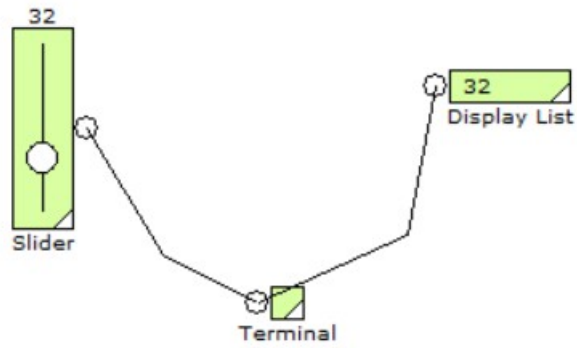
# Tangent

The Tangent component calculates the trigonometric tangent of the input (in radians). Input can be scalar or an array. Functions - column 5



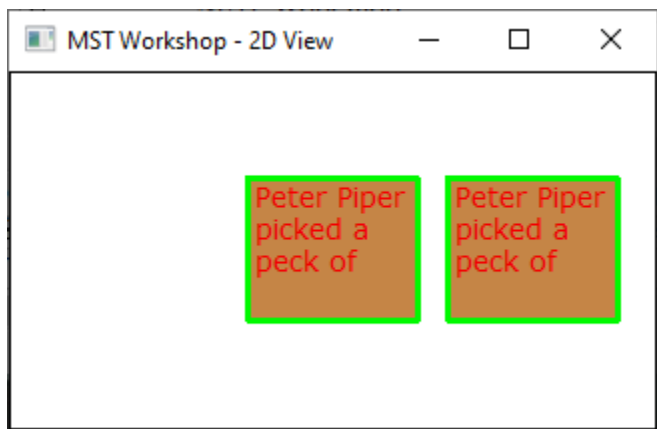
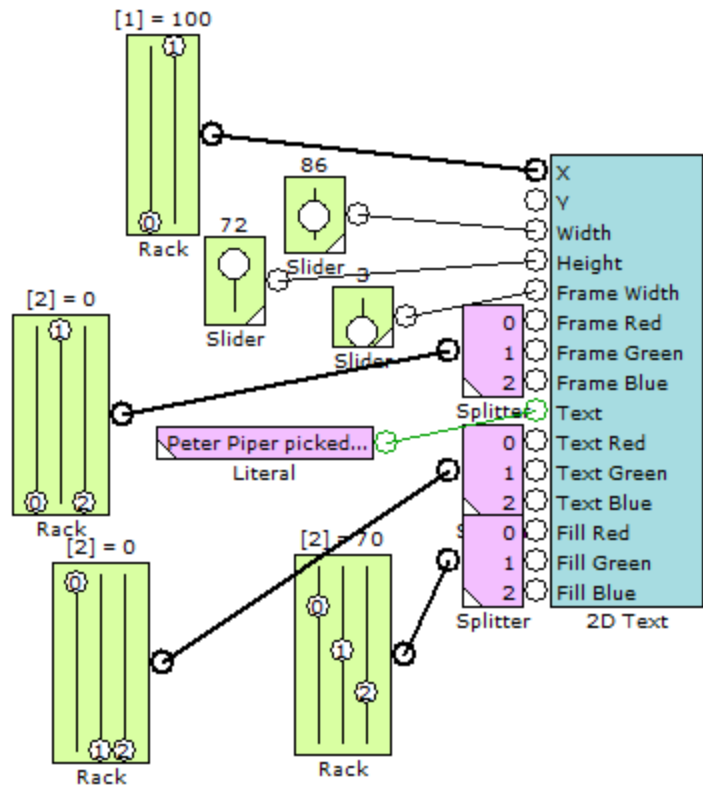
# Terminal

The Terminal is used when you need a mid point to connect two distant components. Controls - column 5



# 2D Text

The 2D Text is used put formatted text on the 2D canvas.



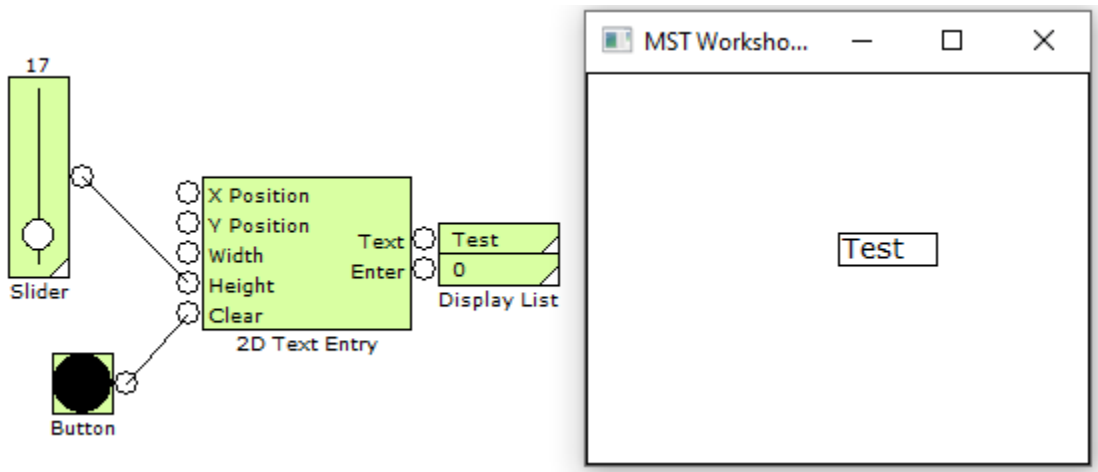
## 2D Text Entry

The 2D Text Entry allows the user to type text in the 2D view at the specified location and with the specified size. Controls - column 2

Pin options:

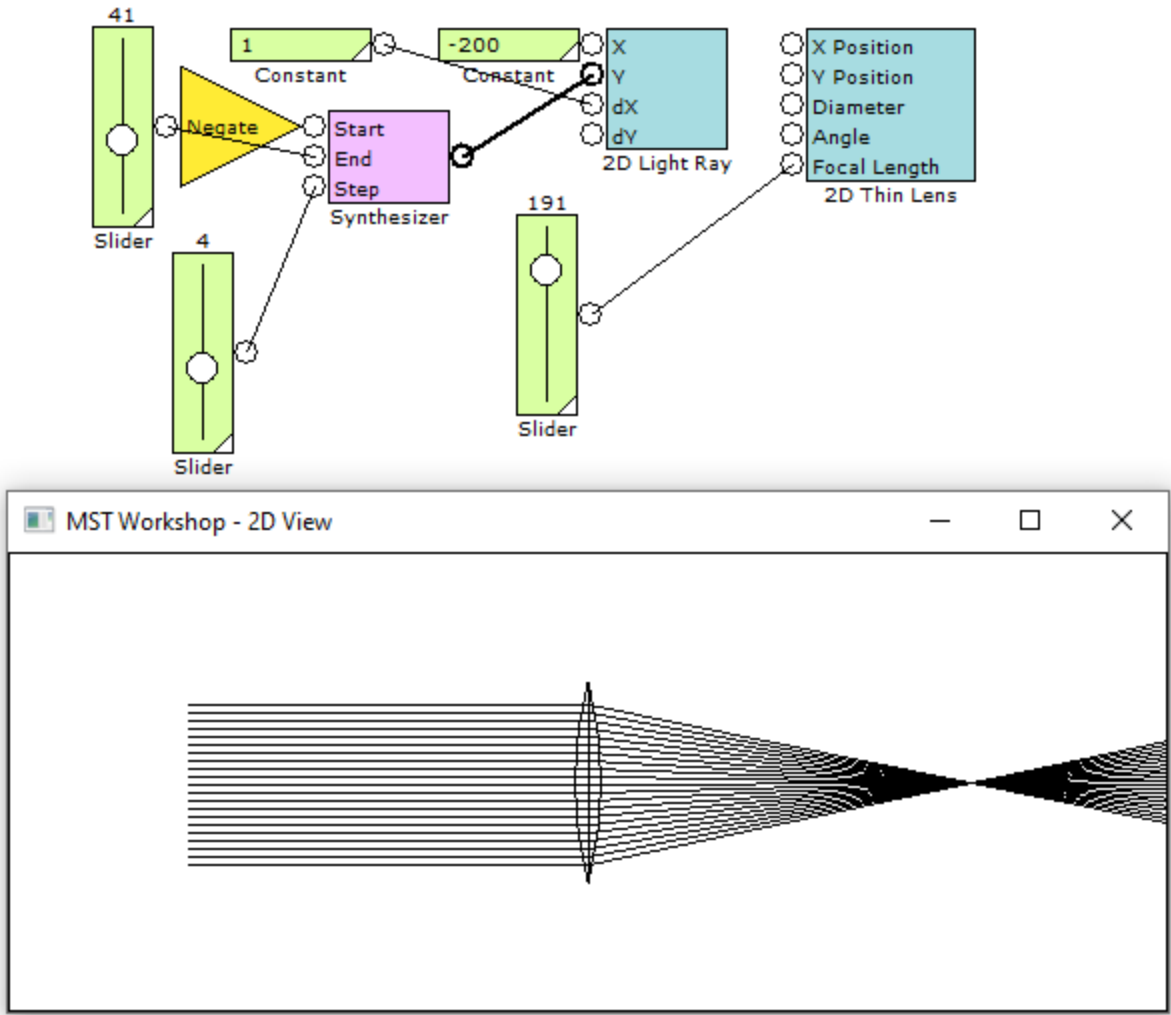
<input checked="" type="checkbox"/>	X Position	Y Position	
<input checked="" type="checkbox"/>	Width	Height	
<input type="checkbox"/>	Line Width		
<input type="checkbox"/>	Line Red	Line Green	Line Blue
<input type="checkbox"/>	Fill Red	Fill Green	Fill Blue
<input type="checkbox"/>	Selected Red	Selected Green	Selected Blue
<input checked="" type="checkbox"/>	Clear		
<input type="checkbox"/>	Alpha		

OK Cancel



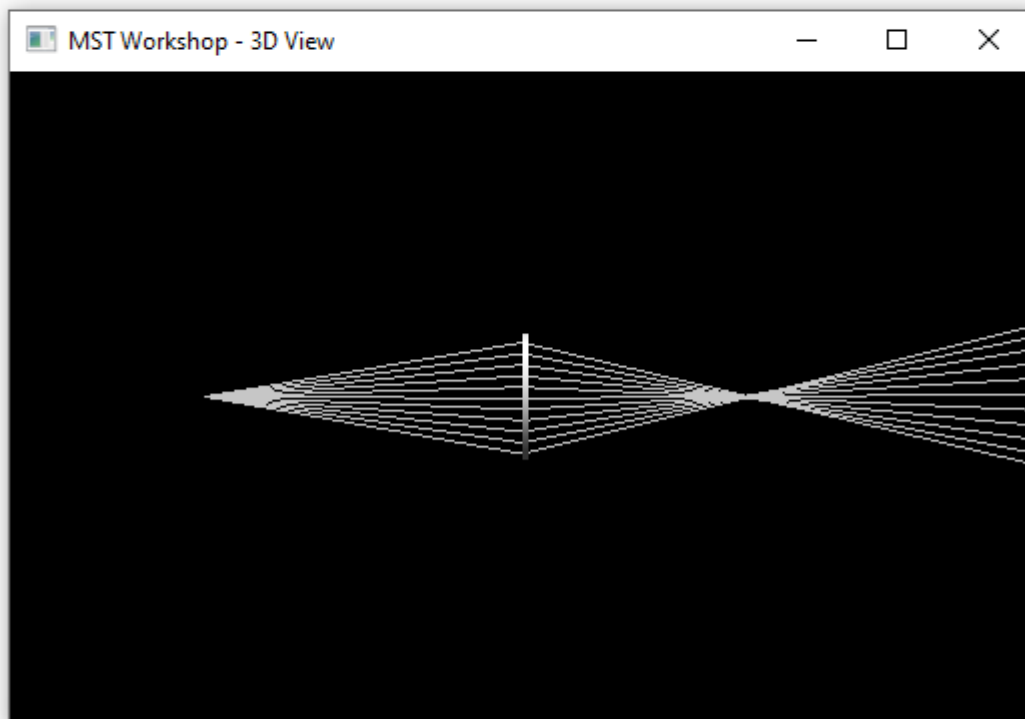
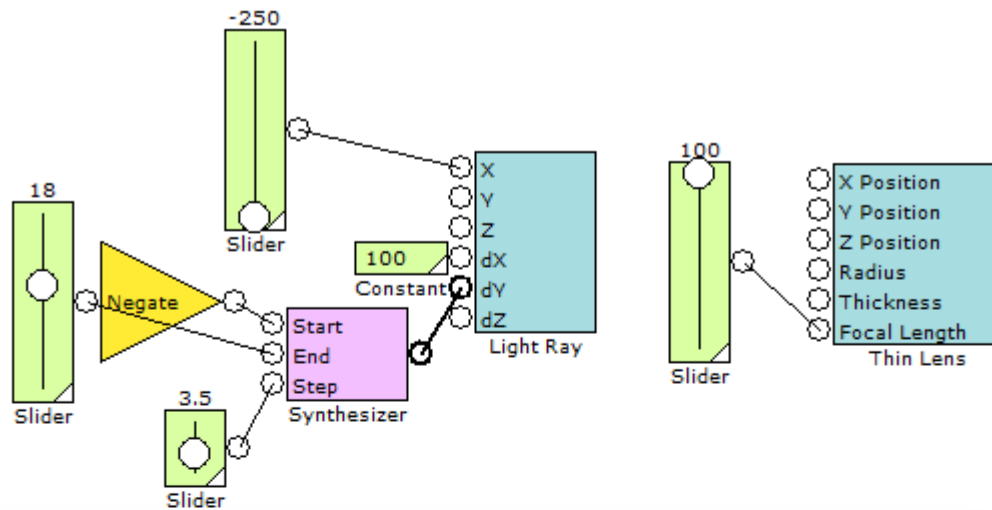
## 2D Thin Lens

The 2D Thin Lens component bends light rays in the 2D view window. The settings determine how the ray progresses through the lens. 2D - column 3



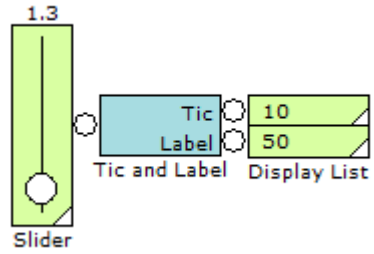
## 3D Thin Lens

The Thin Lens component bends light rays in the 3D view window. The settings determine how the ray progresses through the lens. The Affine [4x4] array input modifies the size, position and orientation of the lens. 3D - column 5



## Tic and Label

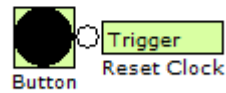
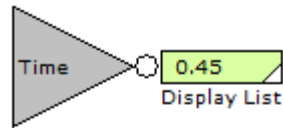
The Tic and Label component calculates the best separation for tics and labels for the Vertical or Horizontal Axis component. 2D - column 1





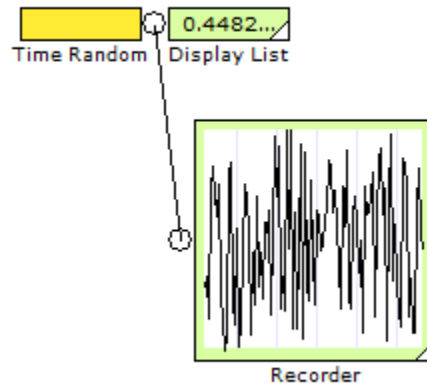
# Time

The Time component outputs the number of milliseconds since start was pressed. This time is based on the components in the workspace not real time. Functions - column 7



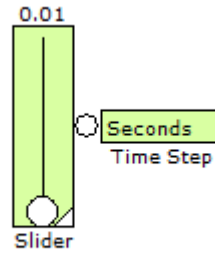
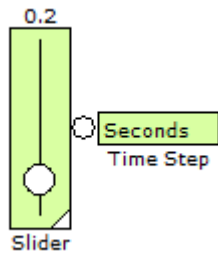
# Time Random

The Time Random component outputs a new random number between 0 and 1 when the clock runs.  
Functions - column 6

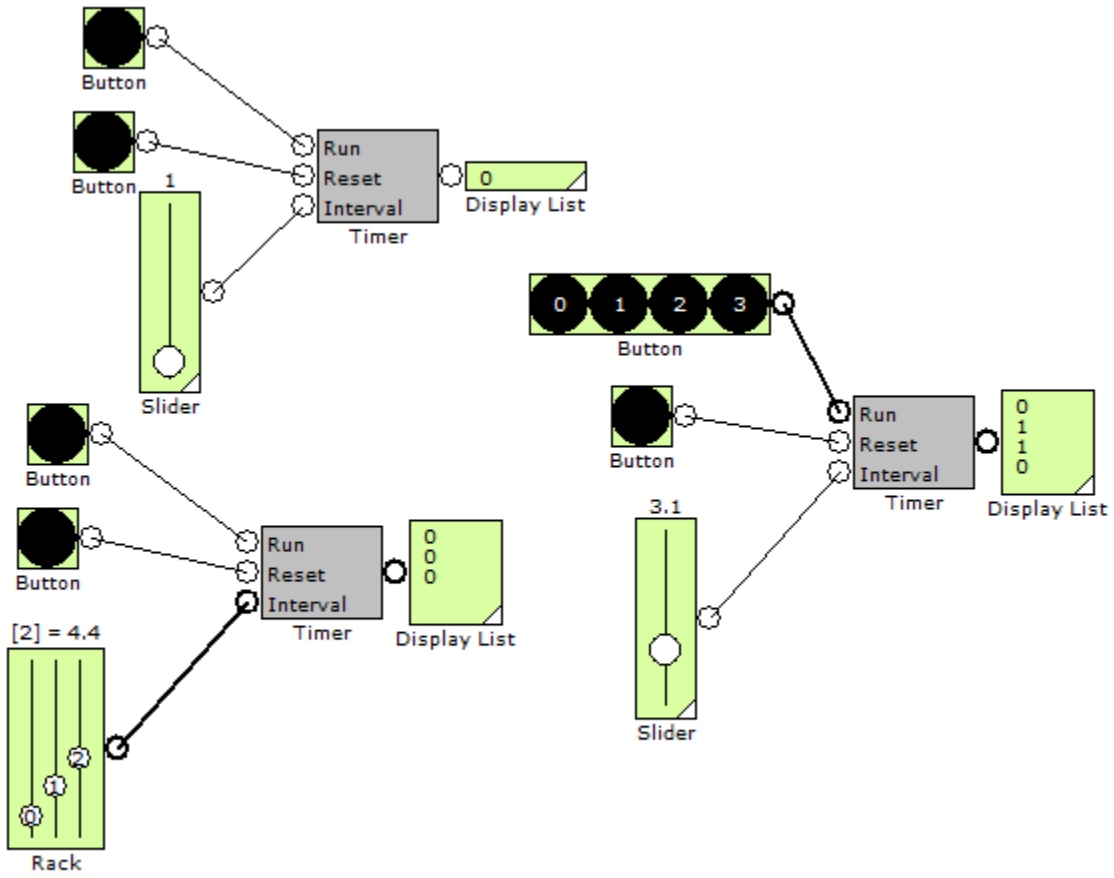


## Time Step

The Time Step component changes the time interval between calculations. It does not change the time given to execute, it changes the time interval that the solution will use to advance each time based component. The minimum time step allowed is 0.001 Seconds (1 millisecond) Controls - column 3



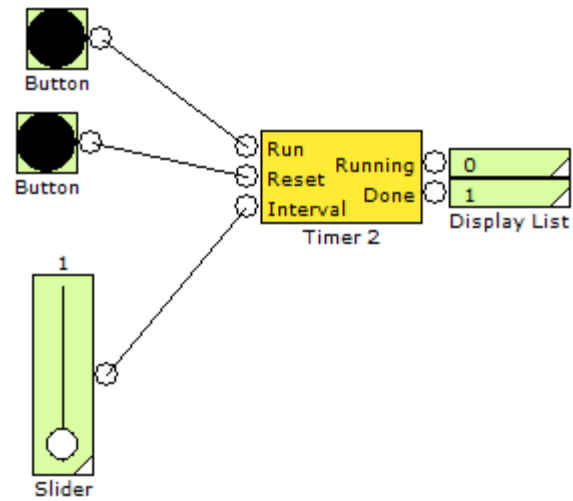
# Timer



The Timer component measures time whenever the Run input is true, when the interval is reached the output goes high and remains high until it is reset. Functions - column 7

## Timer 2

The Timer 2 component starts when the Start input is triggered and runs until the Interval time has elapsed or it is reset. Functions - column 7



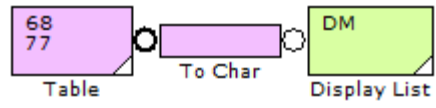
## To ASCII

The To ASCII component converts the one letter input to the ASCII number. Arrays - column 6



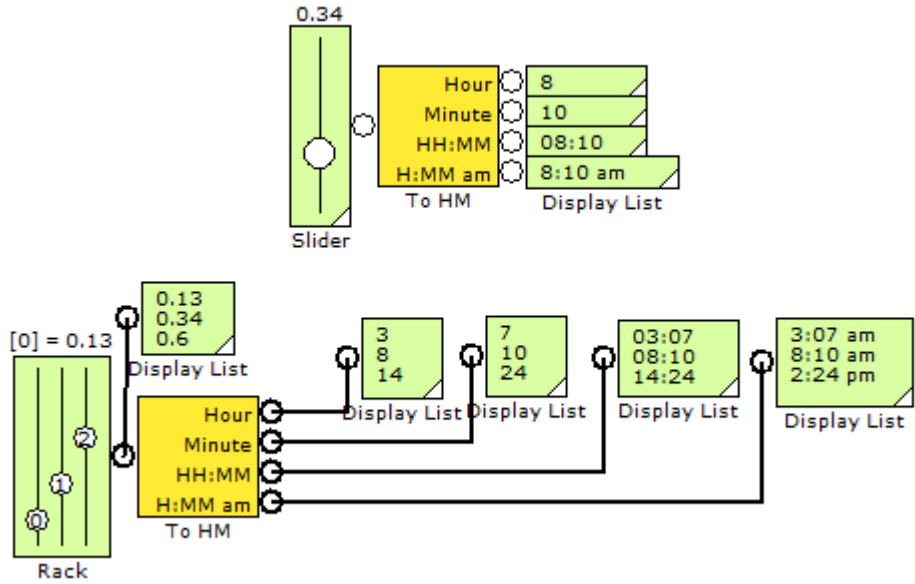
## To Char

The To Char component converts the input numbers to characters. Arrays - column 6



# To HM

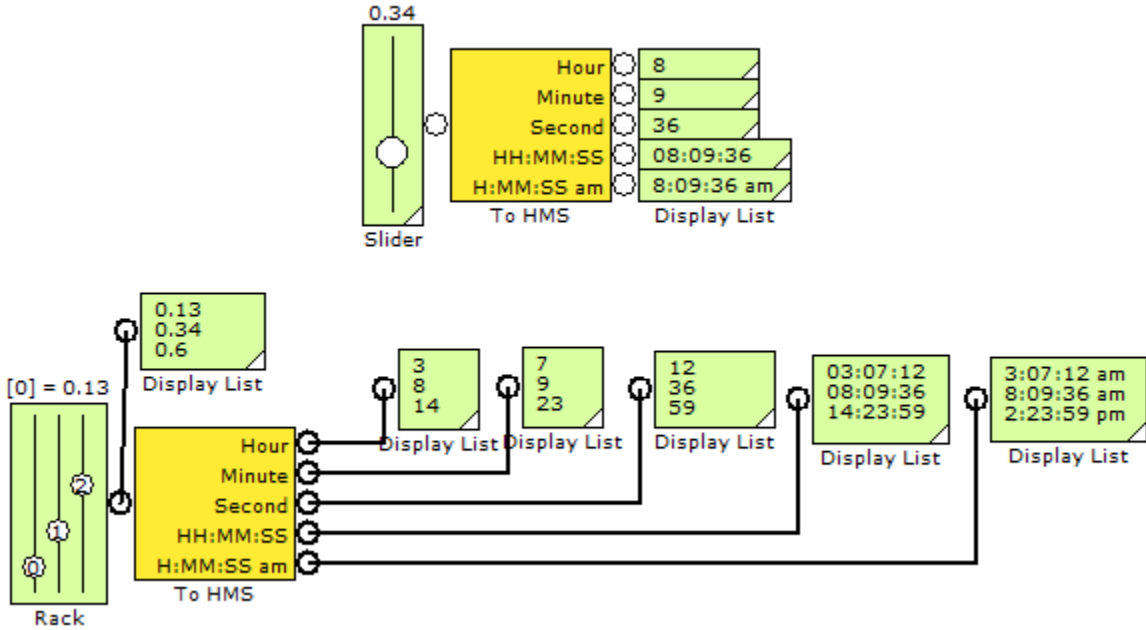
The To HM component converts a day to Hours and Minutes. Also provides 24 and 12 hour time formats. Functions - column 5





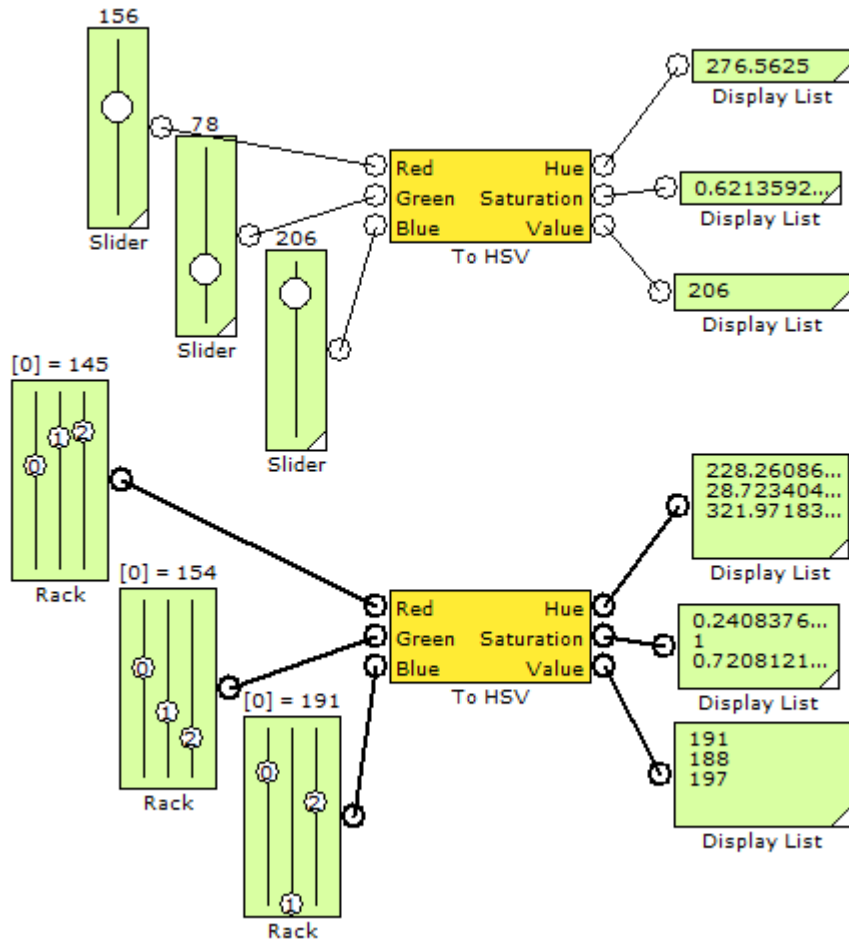
# To HMS

The To HMS component converts a day to Hours, Minutes and Seconds. Also provides 24 and 12 hour time formats. Functions - column 5



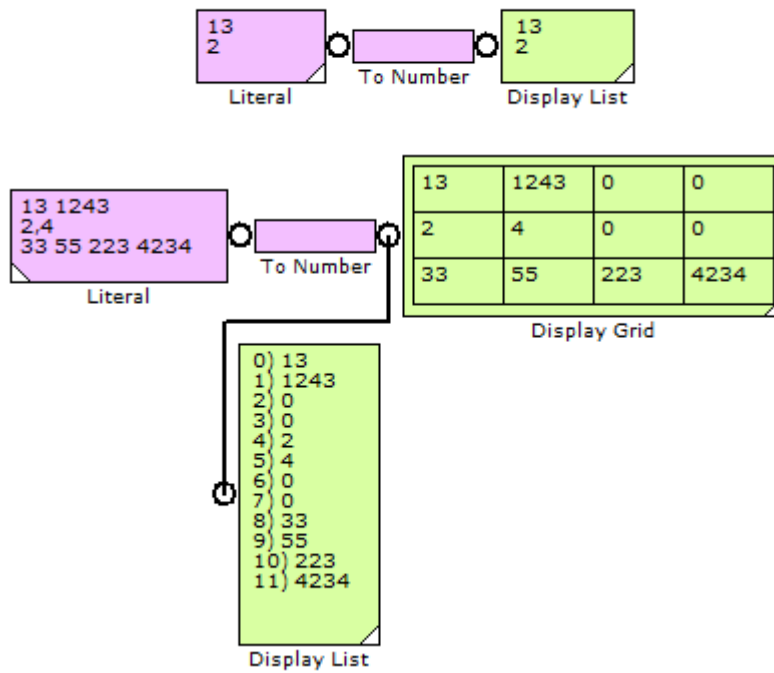
# To HSV

The To HSV component outputs a color from Red, Green, Blue to Hue, Saturation, Value.  
Functions - column 5



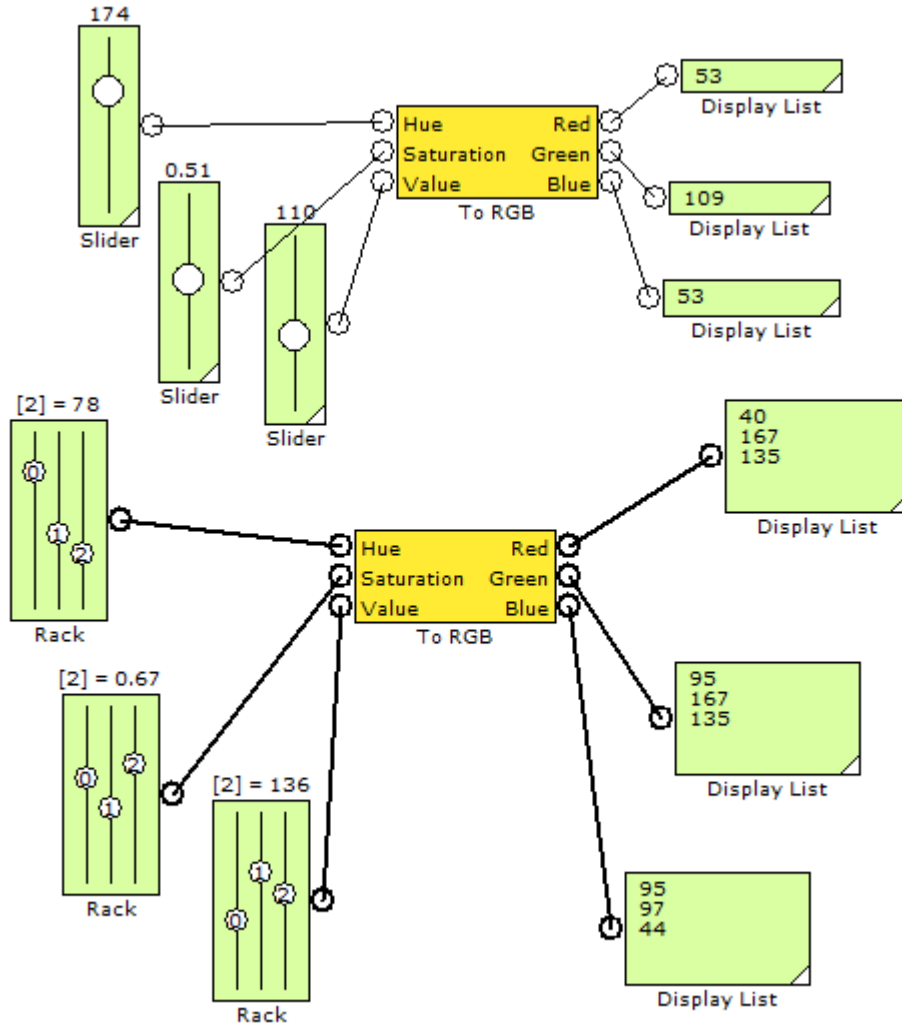
# To Number

The To Number component converts the text input to numerical values. Arrays - column 6



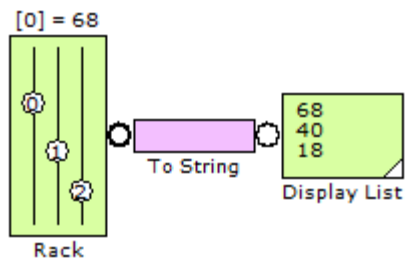
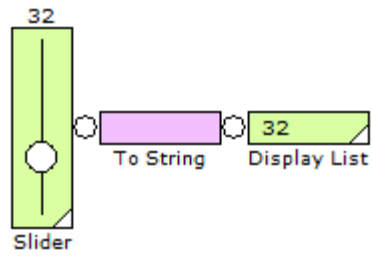
# To RGB

The To RGB component converts a color from Hue, Saturation, Value to Red, Green, Blue.  
Functions - column 5



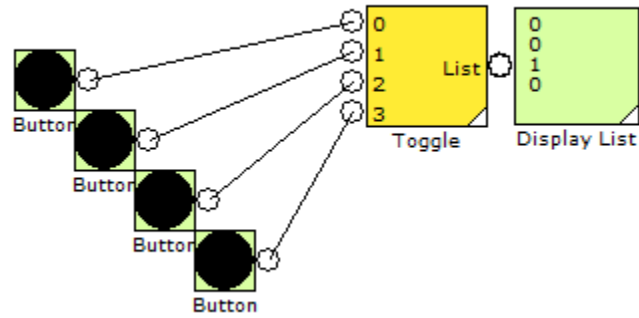
# To String

The To String component converts the input numbers to strings. Arrays - column 6



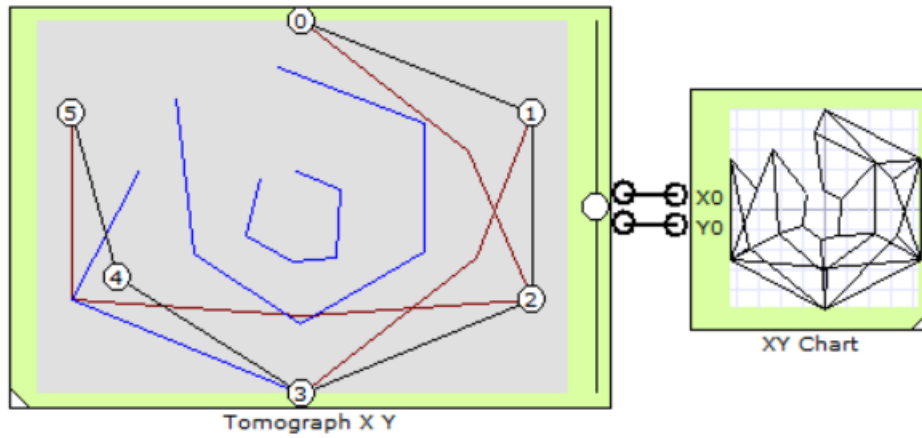
# Toggle

The Toggle component is like the function of the checkbox. It toggles values in the output each time the input goes high. Functions - column 3



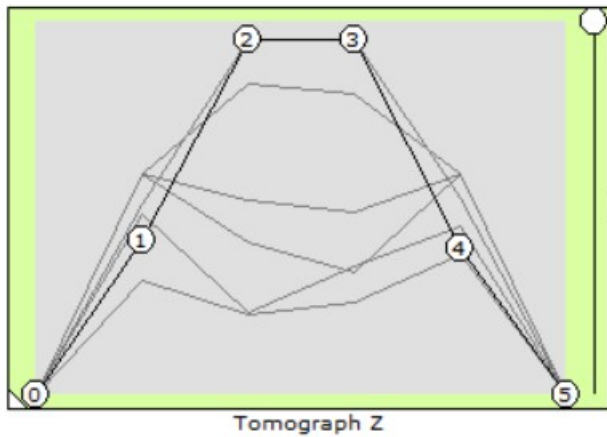
# Tomograph X Y

The Tomograph X Y component lets you create 3 dimensional data sets by drawing individual curves.  
Controls – column 2



## Tomograph Z

The Tomograph Z component lets you create a 2 dimensional matrix by drawing individual curves. You can adjust the view to front (X vs. Z) or side (Y vs. Z). Controls – column 2



-100	-18	90.2...	90.2...	-22	-99.9...
-100	2	90.2...	90.2...	5	-99.9...
-100	17.5...	66	61	17.5...	-99.9...
-100	17.5...	4	-3	17.5...	-99.9...
-100	17.5...	-19	-35	17.5...	-99.9...
-100	-4	-56	-32	-10	-99.9...
-100	-39	-57	-51	-25	-99.9...

Display Grid



# 3D Torus

The 3D Torus draws a torus in the 3D view window. The X, Y, and Z inputs set the position and size of the torus. The Affine [4x4] array input modifies the size, position and orientation. 3D - column 4

**3D Torus**

Tube Steps:

First Tube Step:

Last Tube Step:

Ring Steps:

First Ring Step:

Last Ring Step:

X Offset:

Y Offset:

Z Offset:

Faceted Surface  
 Smooth Surface

**Pin options:**

X Position     Y Position     Z Position

X Rotate     Y Rotate     Z Rotate

Ring Radius     Tube Radius

Red     Green     Blue

Alpha

[4x4]

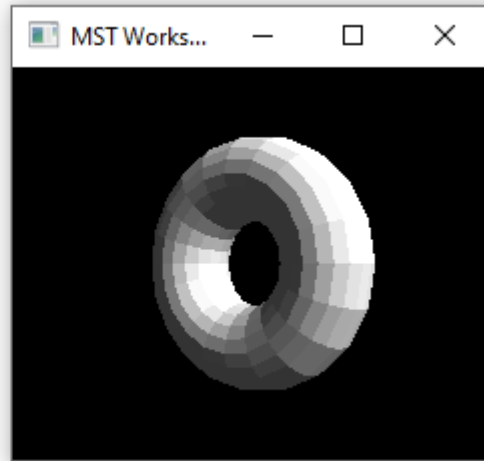
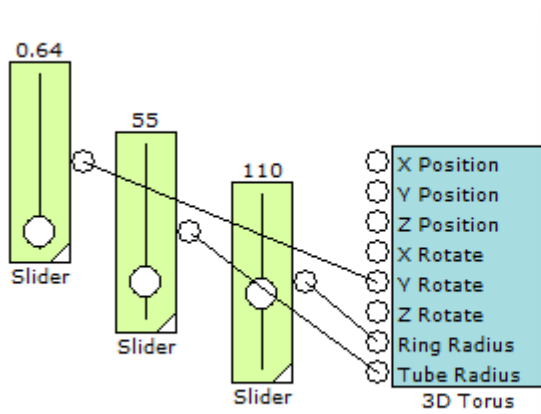
**Rotation:**

X Clockwise     X zero is North  
 X Counter Clockwise     X zero is East  
 X zero is South  
 X zero is West

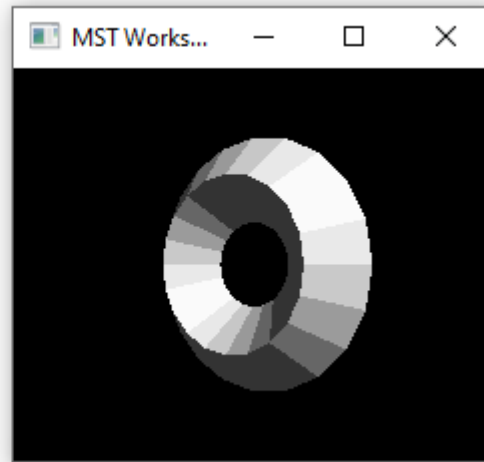
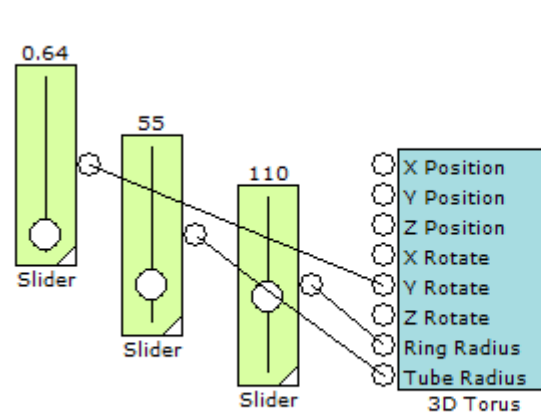
Y Clockwise     Y zero is North  
 Y Counter Clockwise     Y zero is East  
 Y zero is South  
 Y zero is West

Z Clockwise     Z zero is North  
 Z Counter Clockwise     Z zero is East  
 Z zero is South  
 Z zero is West

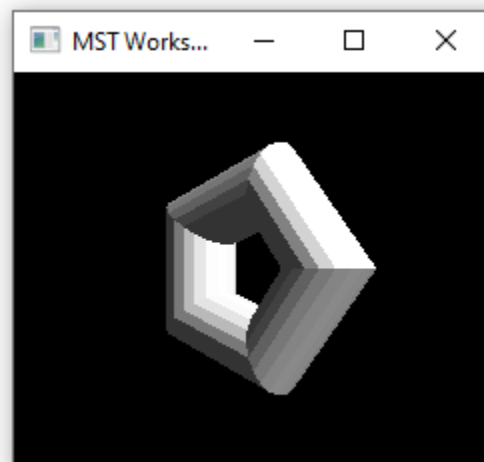
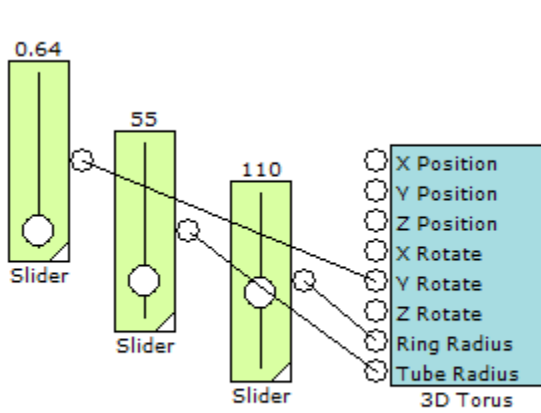
Tube Steps = 20, Ring Steps = 20

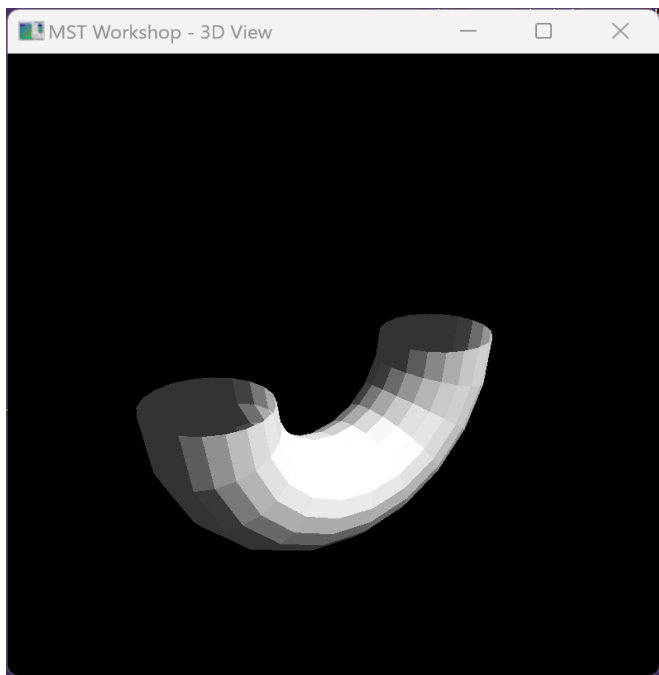
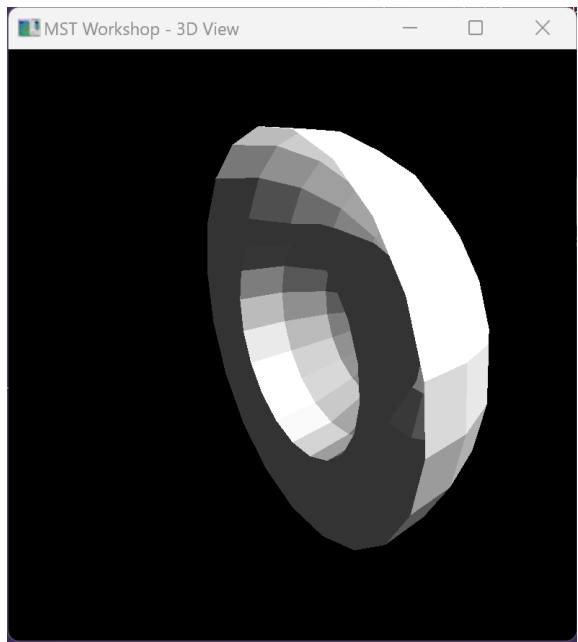


Tube Steps = 4, Ring Steps = 20



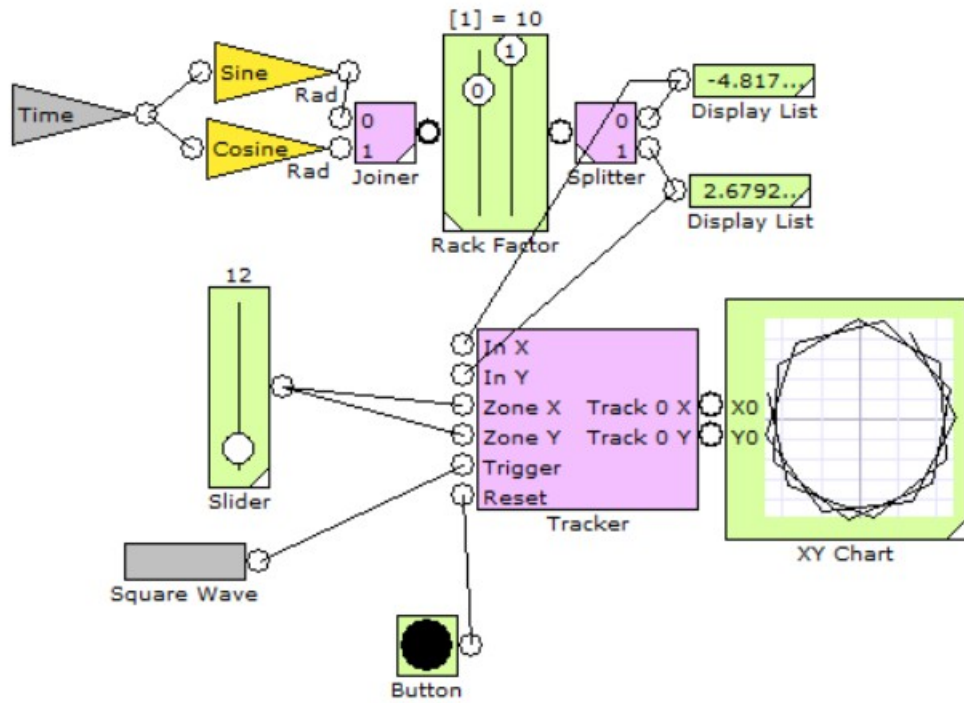
Tube Steps = 20, Ring Steps = 4





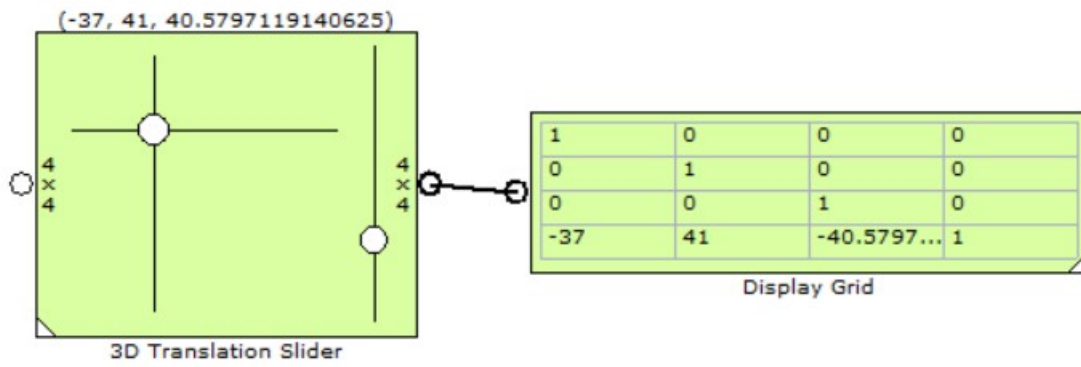
# Tracker

The Tracker component creates a path from input points. Array – column 2



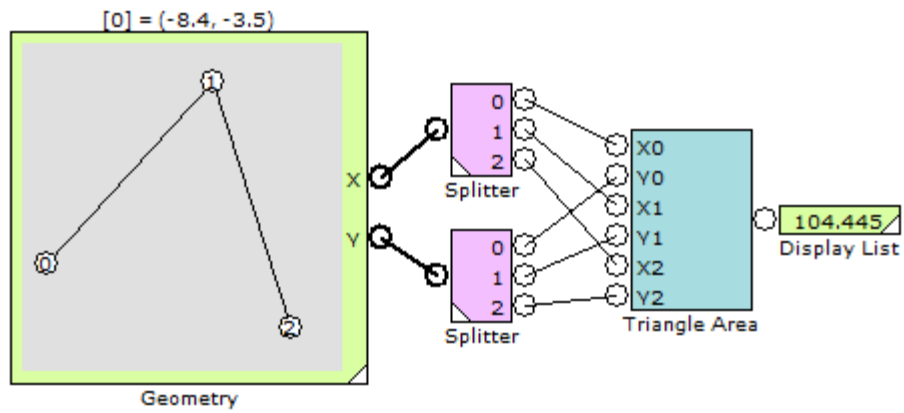
### 3D Translation Slider

The 3D Translation Slider component creates a 4x4 translation matrix from the position of the control knobs. Controls – column 2



# Triangle Area

The Triangle Area component computes the area of a triangle from its 3 XY coordinates. 2D - column 2



# Triangular Wave

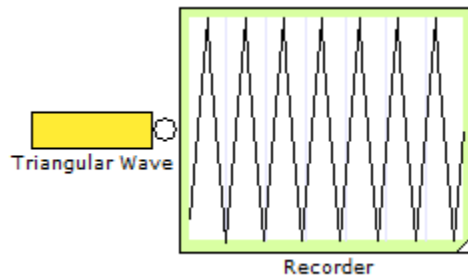
The Triangular Wave component outputs a triangular wave that slopes up then down when the clock is running. Functions - column 8

Triangular Wave

Output Period:

Output Maximum:

OK Cancel

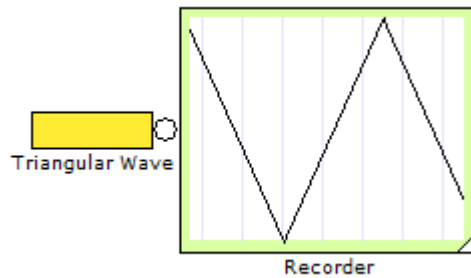


Triangular Wave

Output Period:

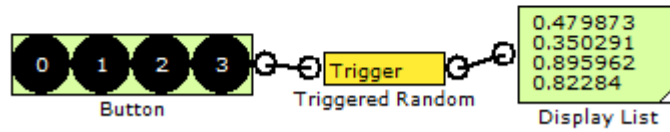
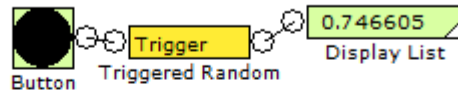
Output Maximum:

OK Cancel



# Triggered Random

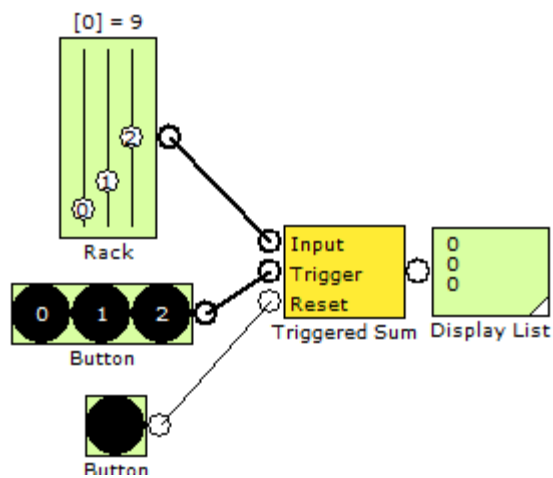
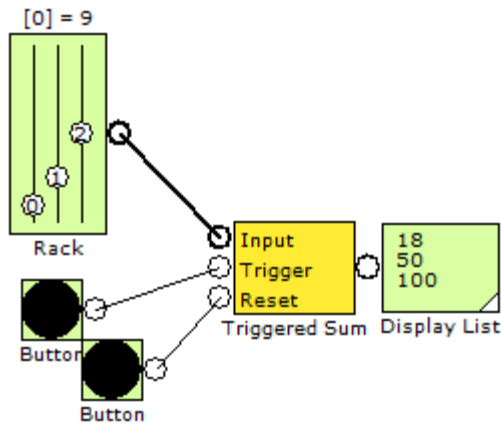
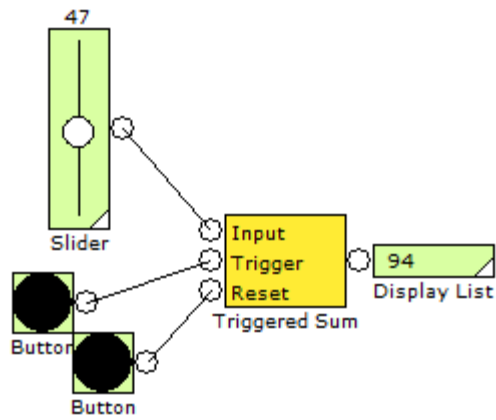
The Triggered Random component delivers random number between 0 and 1 when the trigger goes from between false to true. Functions - column 6





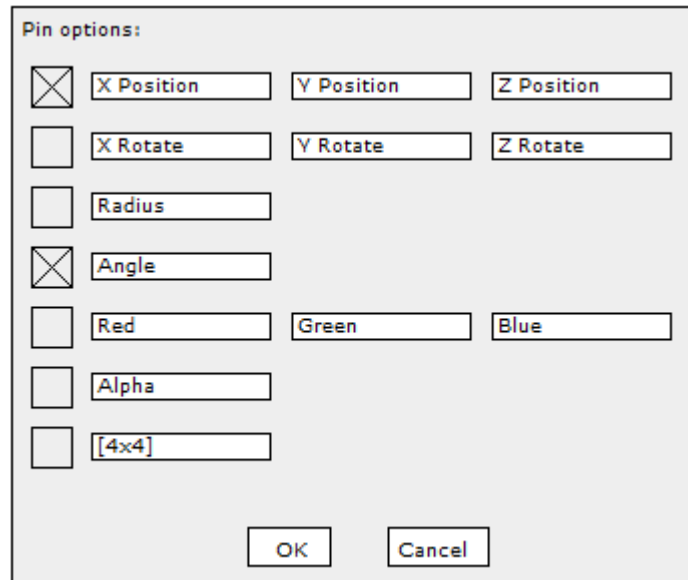
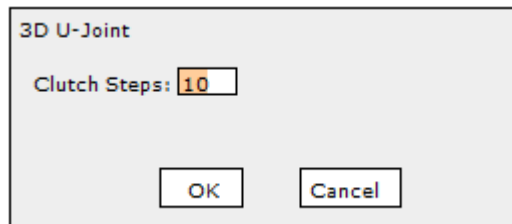
# Triggered Sum

The Triggered Sum component sums the input each time the trigger goes true. Functions - column 3

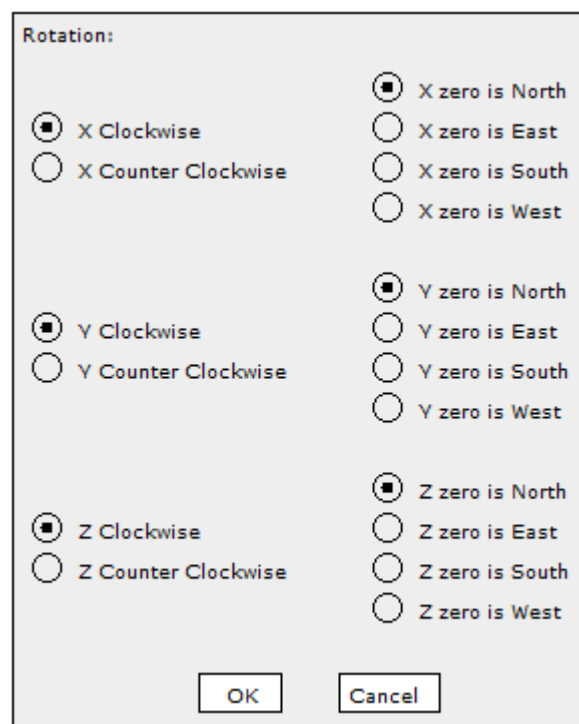


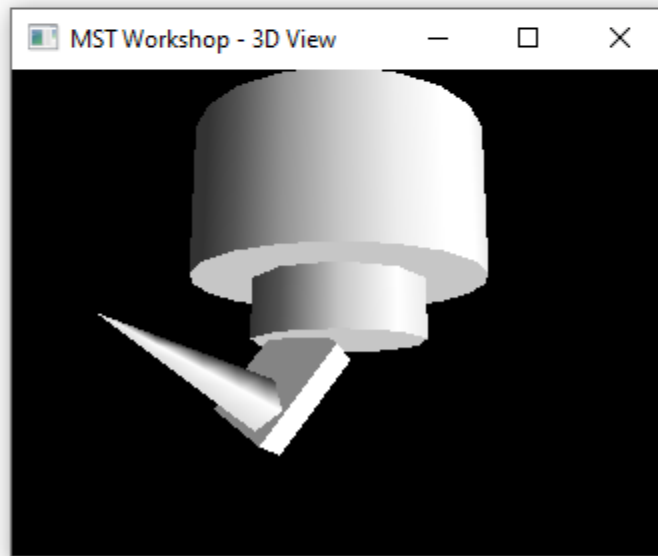
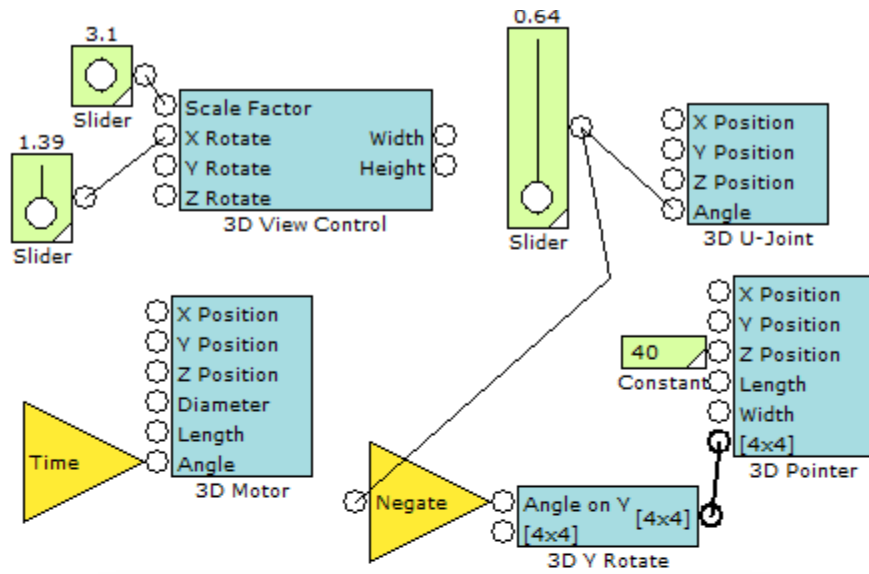
## 3D U-Joint

The 3D U-Joint component creates a connection between two shafts that can be at an angle. You can attach a driver to the input and drive something at the end of the shaft. Place the U-Joint at the same X,



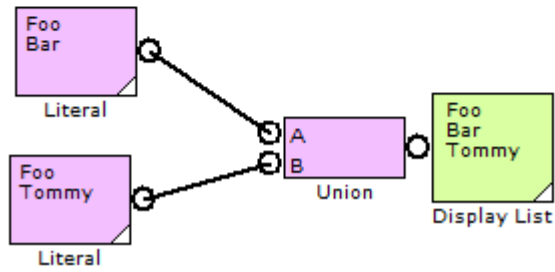
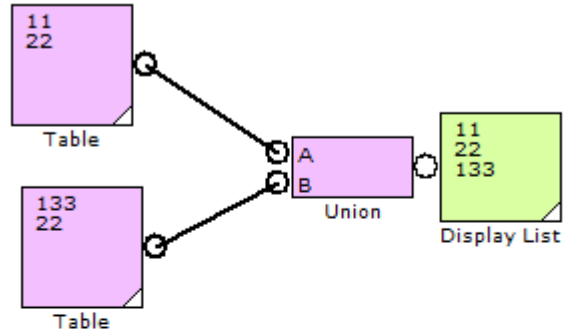
Y, Z position as a motor and place a gear at the X, Y, Z coordinates as the other end. 3D - column 4





# Union

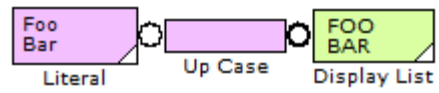
The Union component combines the elements of the two inputs and outputs each value just once.  
Arrays - column 5



## Up Case

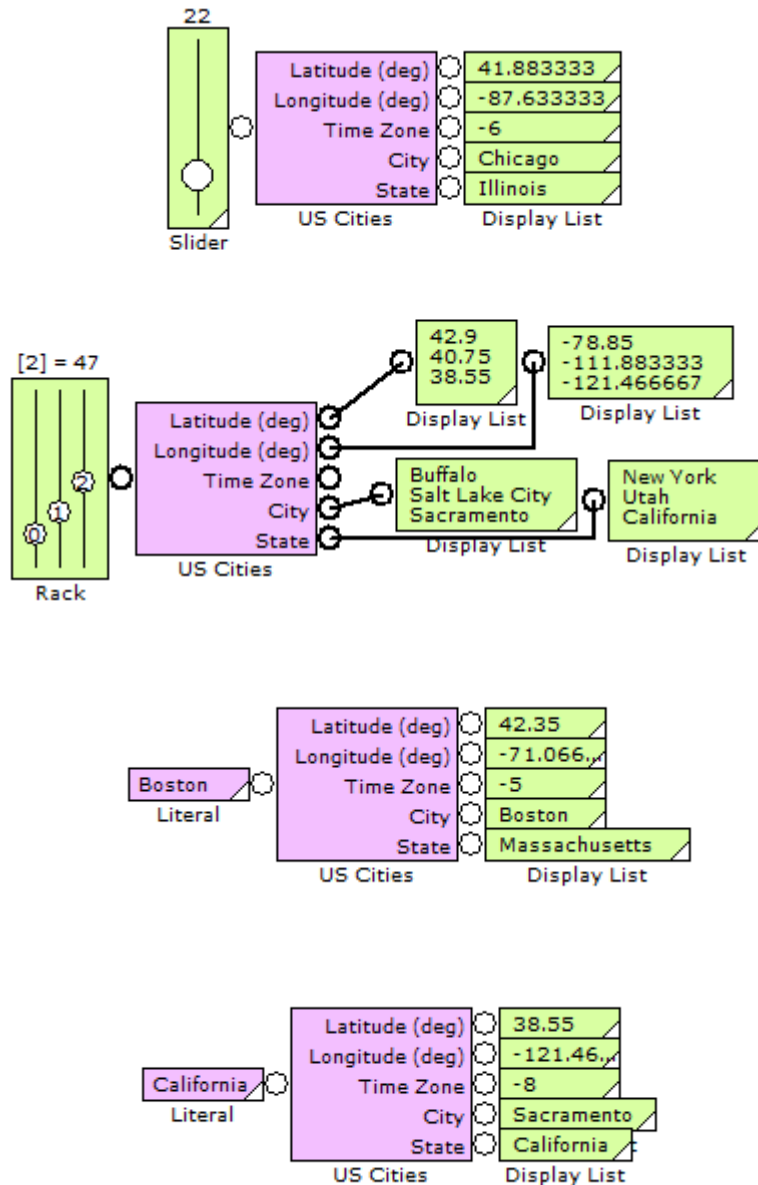
The Up Case component changes the input string to upper case characters.

Arrays - column 6



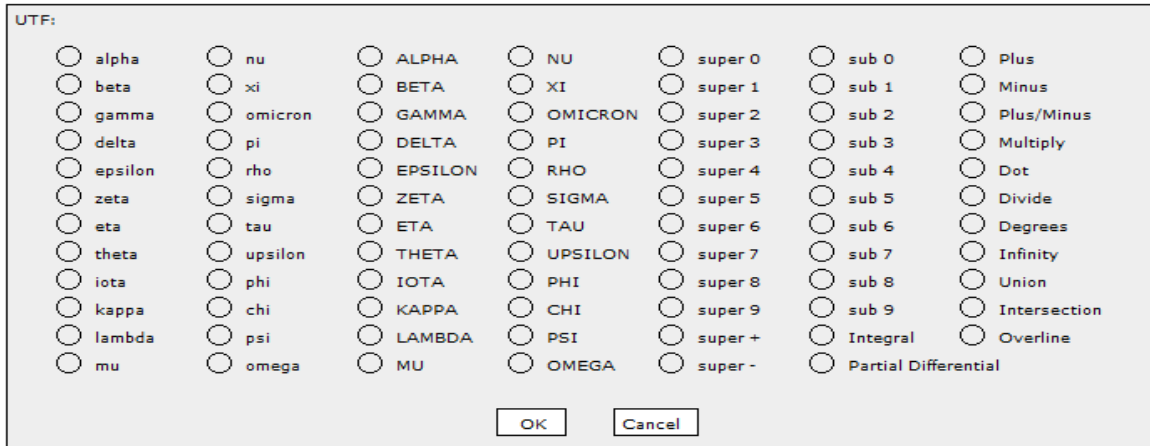
# US Cities

The US Cities component contains Latitude, Longitude, and Time Zone for some cities in the US.  
 Arrays - column 5

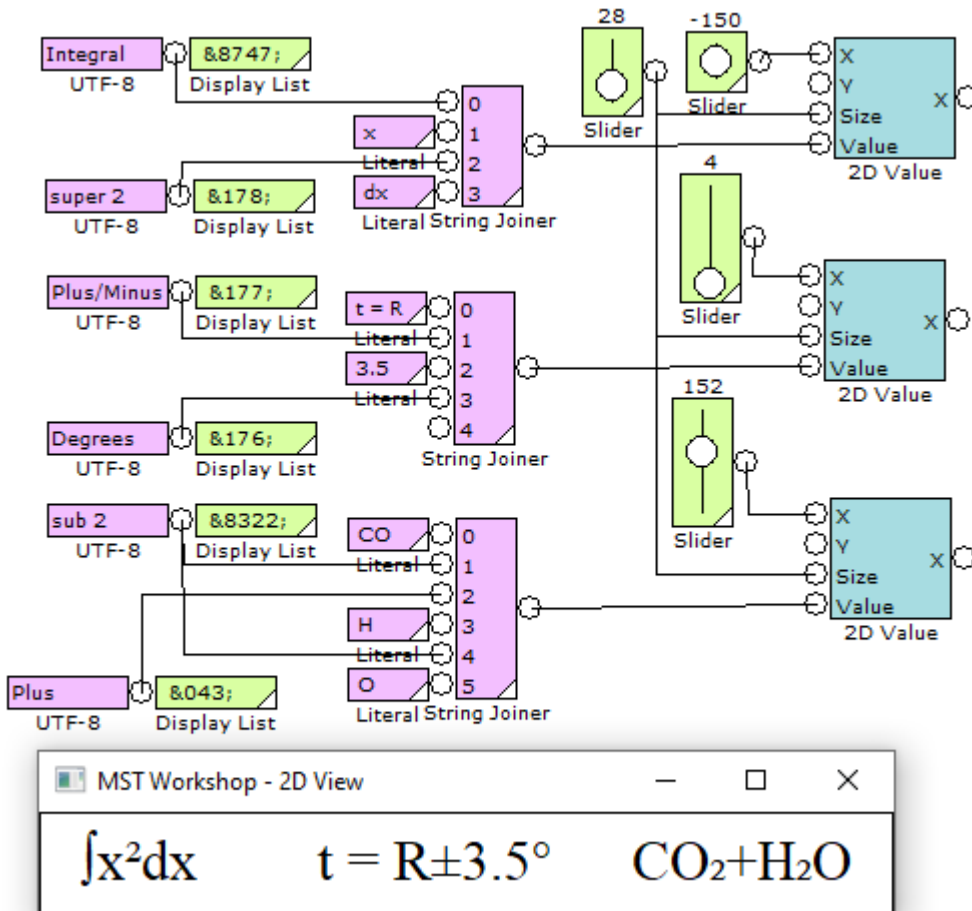


# UTF-8

The UTF-8 component outputs a constant string that represents the UTF-8 character chosen in the options dialog. Use it to display special characters in the 2D Value and 2D Text components. Arrays - column 6



Note: The 2D Value is set to Times font.



## 2D Value

The 2D Value component draws a number or text on the 2D canvas.

The Size input sets the font size.

The Background input enables a white background behind the value.

The Color input sets the color of the value.

The Alpha input fades the value on the canvas.

The Gradient inputs override the color and alpha inputs.

The Gradient input needs 2 xy pairs to indicate the start and end points of the gradient on the canvas.

The Stops input takes 2 or more color inputs. Each stop is 4 or 5 values.

Each of them starts with a fraction to indicate where along the line between the 2 xy pairs to place the color.

If each stop is 4 values: first is the fraction of the gradient.

Usually the first stop is at zero and the last stop is at 1.

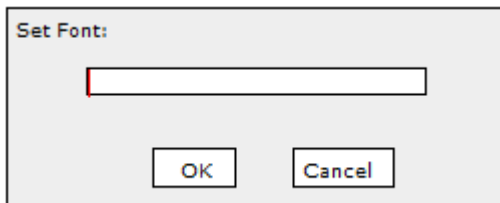
The next 3 values are the Red, Green, Blue components of the color at that fraction.

You can add a fifth value after color to indicate the alpha value at that point.

The Horizontal outputs are the left and right limits of the text on the canvas.

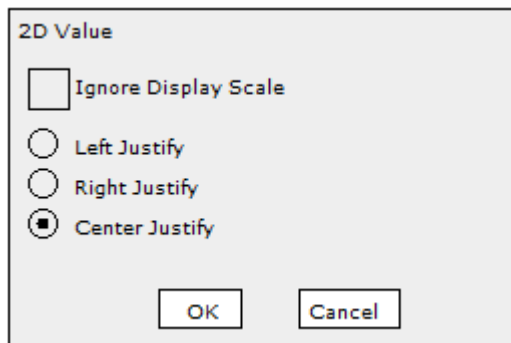
The Vertical outputs are the top and bottom limits of the text on the canvas.

The Center, Left, Right and Decimal Justify control the position of the value on the canvas. 2D - column 1



Set Font:

OK Cancel



2D Value

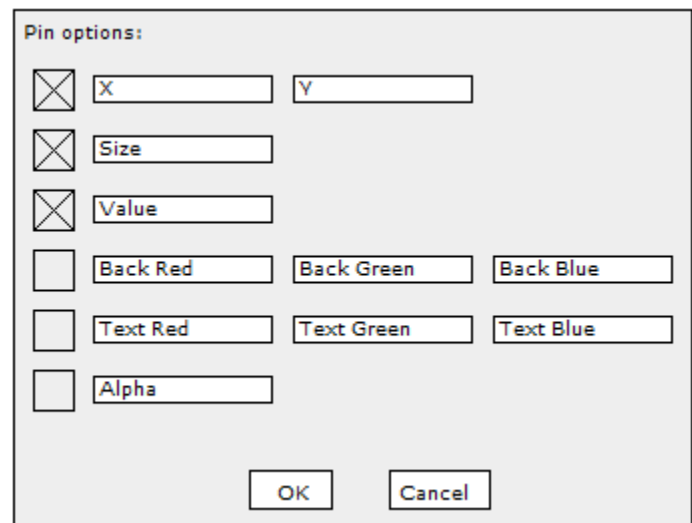
Ignore Display Scale

Left Justify

Right Justify

Center Justify

OK Cancel



Pin options:

X  Y

Size

Value

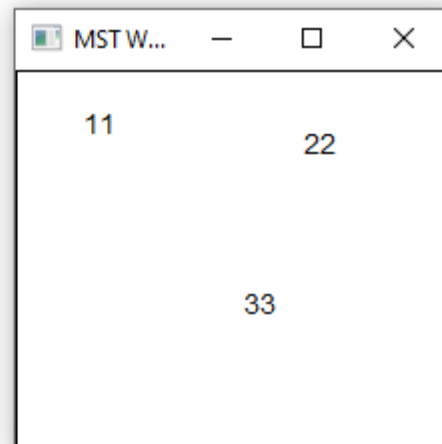
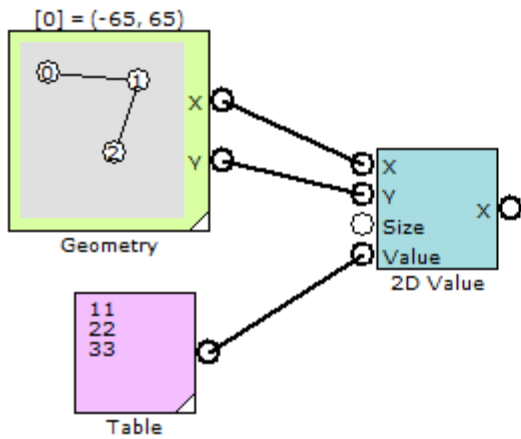
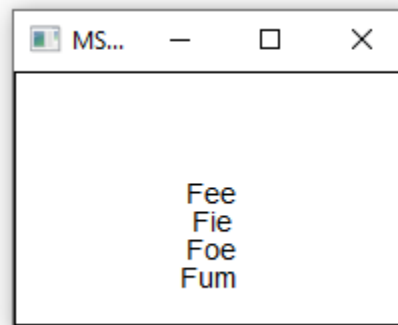
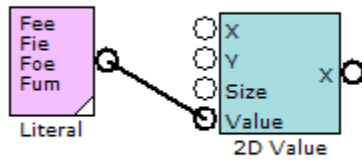
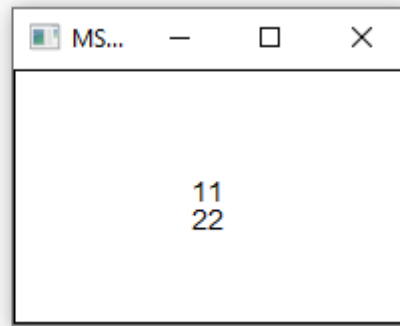
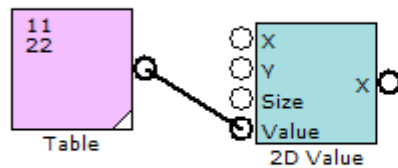
Back Red  Back Green  Back Blue

Text Red  Text Green  Text Blue

Alpha

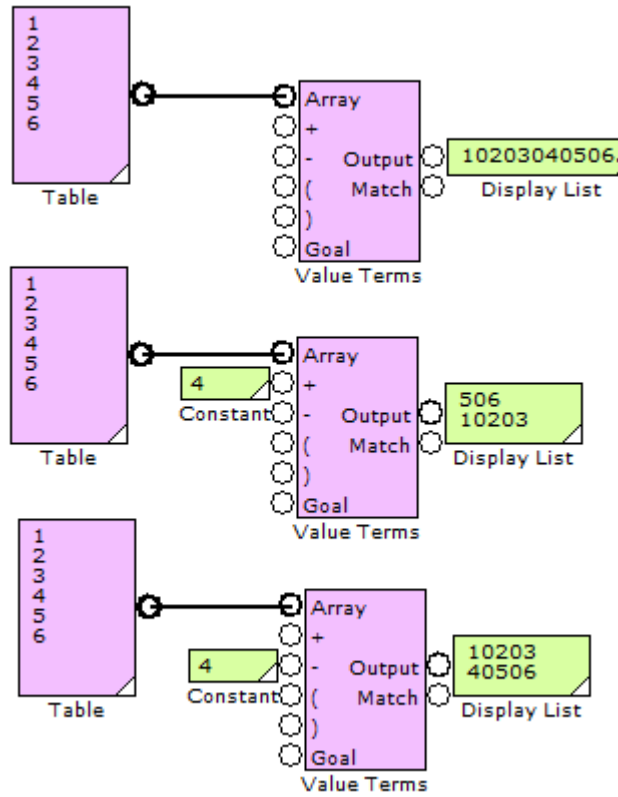
OK Cancel





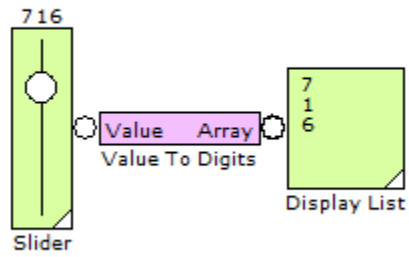
# Value Terms

The Value Terms component converts the input to terms using standard rules of addition, subtraction, multiplication, and grouping. Arrays - column 5



## Value To Digits

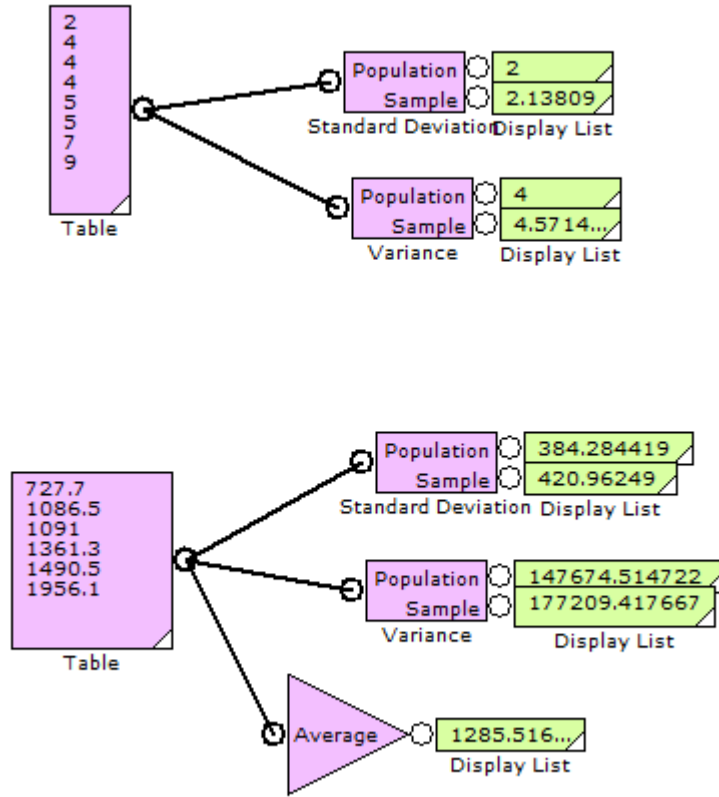
The Value To Digits component converts a value to an array of digits. 10 indicates a negative sign.  
Arrays - column 5



# Variance

The Variance component scans the input array and calculates the variance of those values.  
- column 3

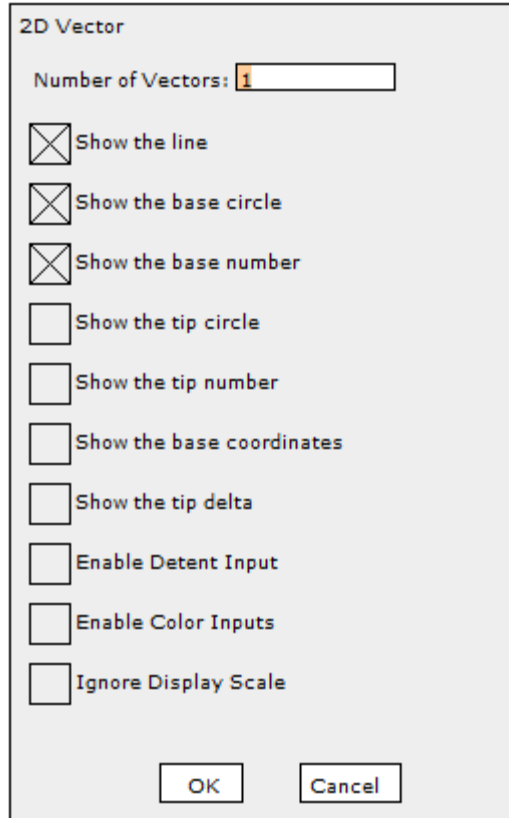
Arrays



# 2D Vector

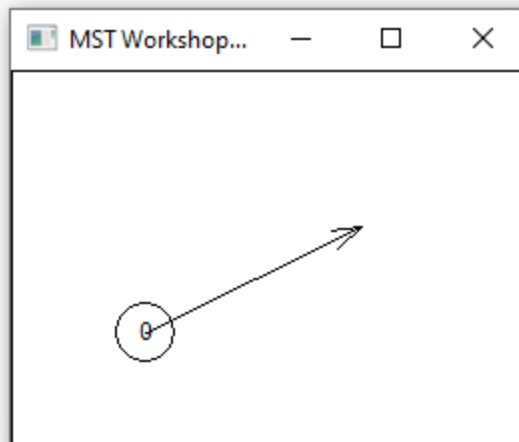
The 2D Vector component outputs the XY coordinates and the dX/dY of the vector.

Controls - column 2



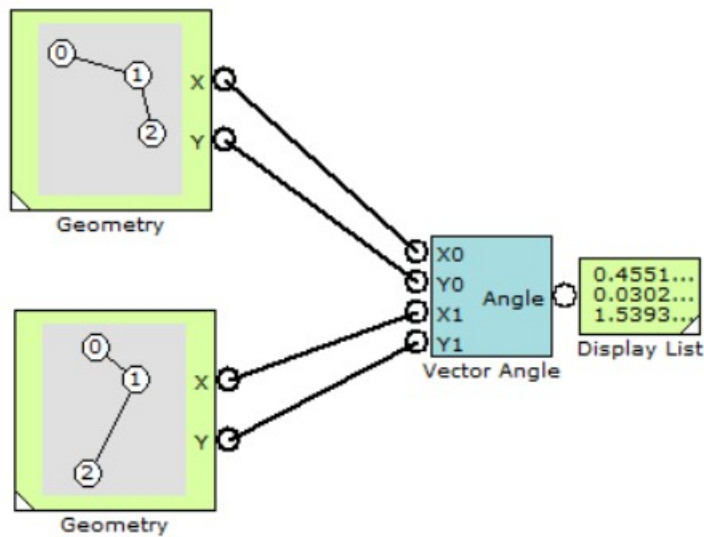
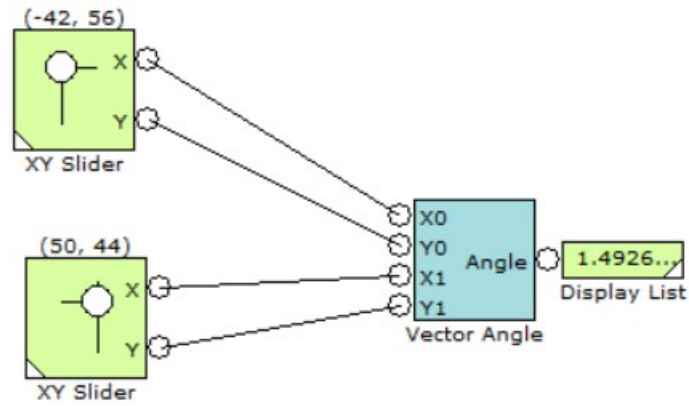
X	-62
Y	-37
dX	108
dY	53

2D Vector      Display List



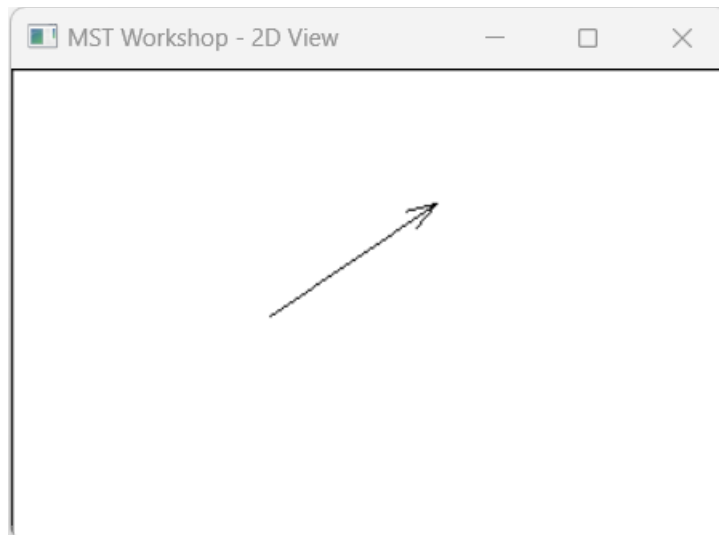
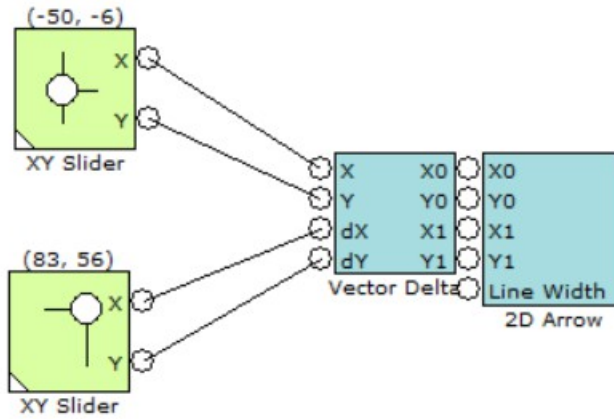
# Vector Angle

The Vector Angle component calculates the angle between 2 vectors. 2D - column 2



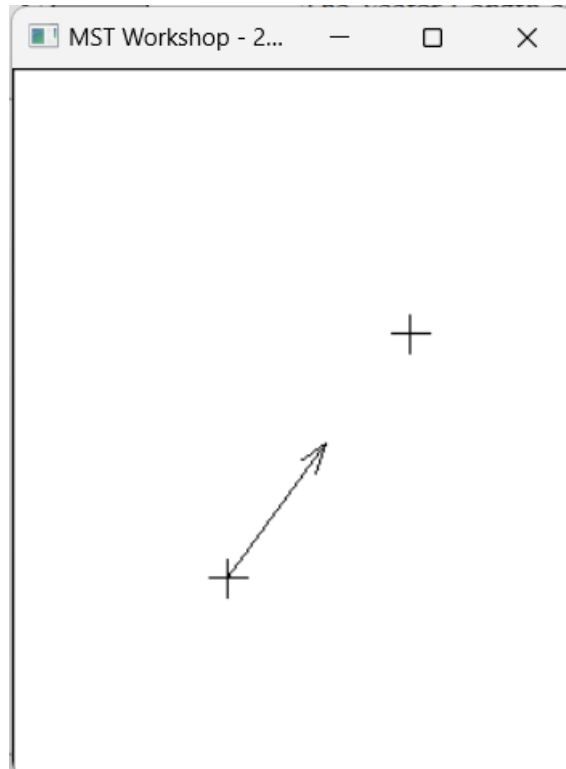
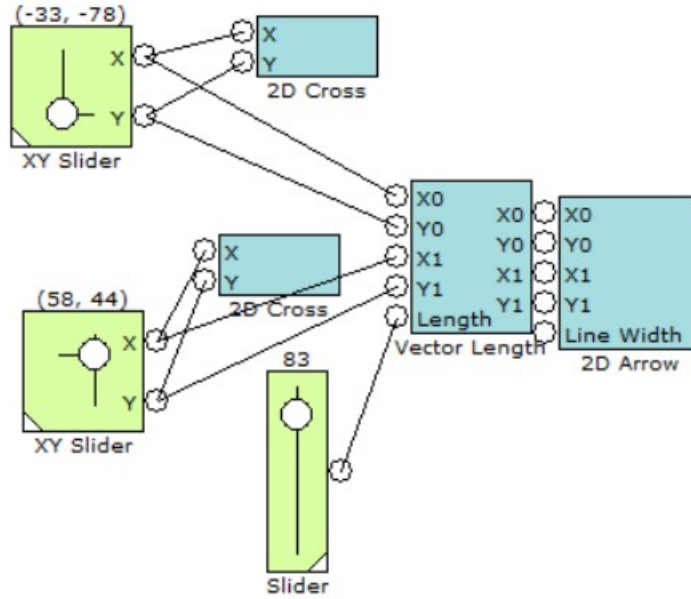
# Vector Delta

The Vector Delta component takes X Y coordinates and deltas and outputs the resulting vector. 2D - column 2



# Vector Length

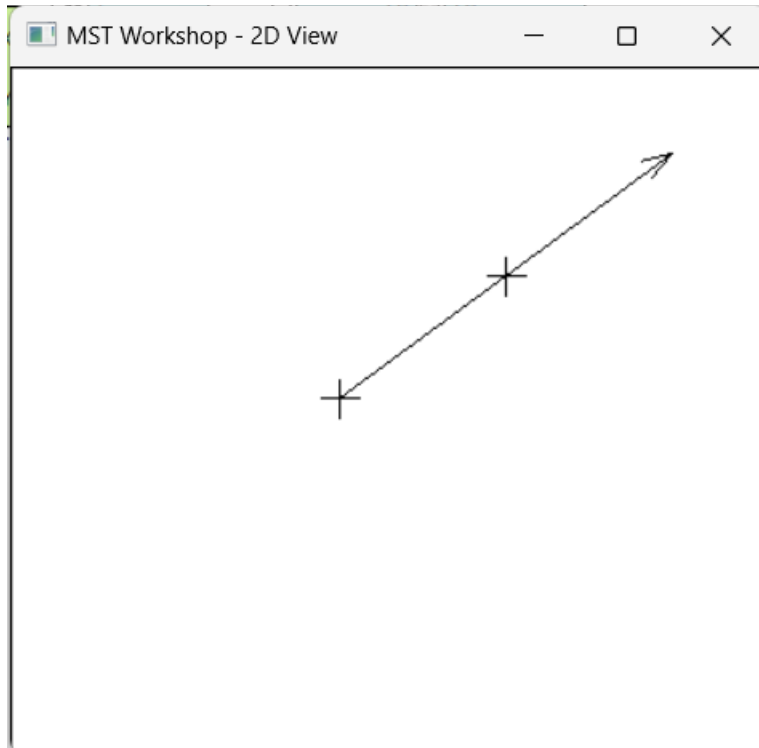
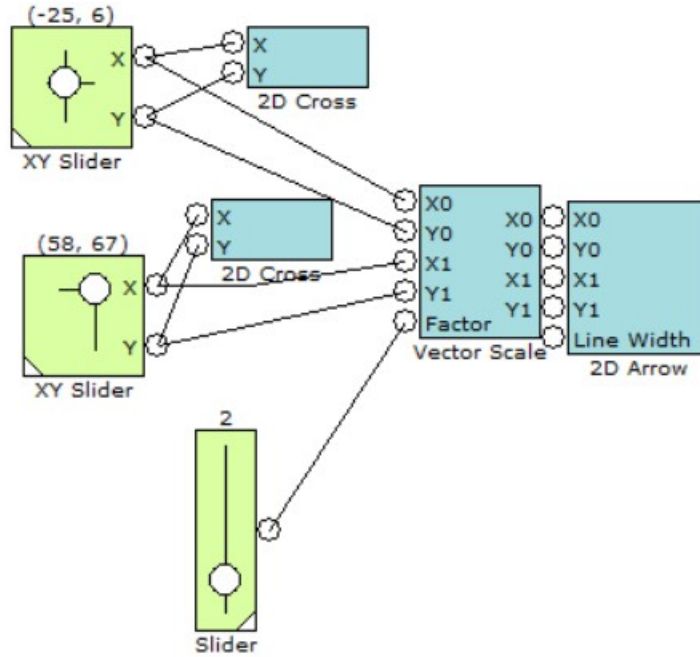
The Vector Length component takes a vector begin and end coordinates with a length and outputs vector begin and end coordinates of that length. 2D - column 2





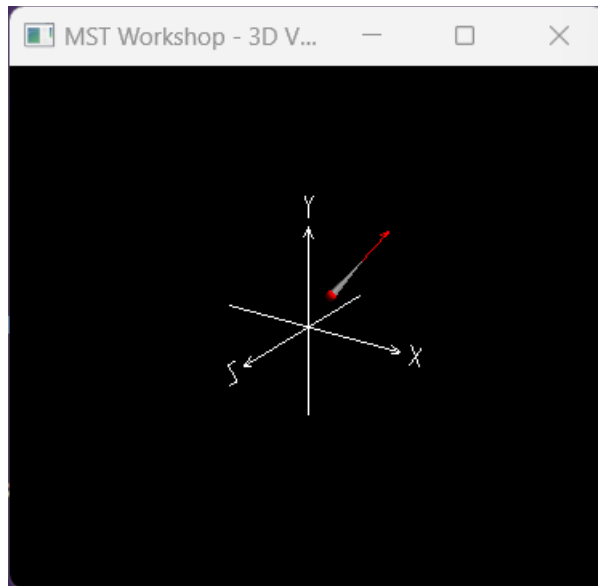
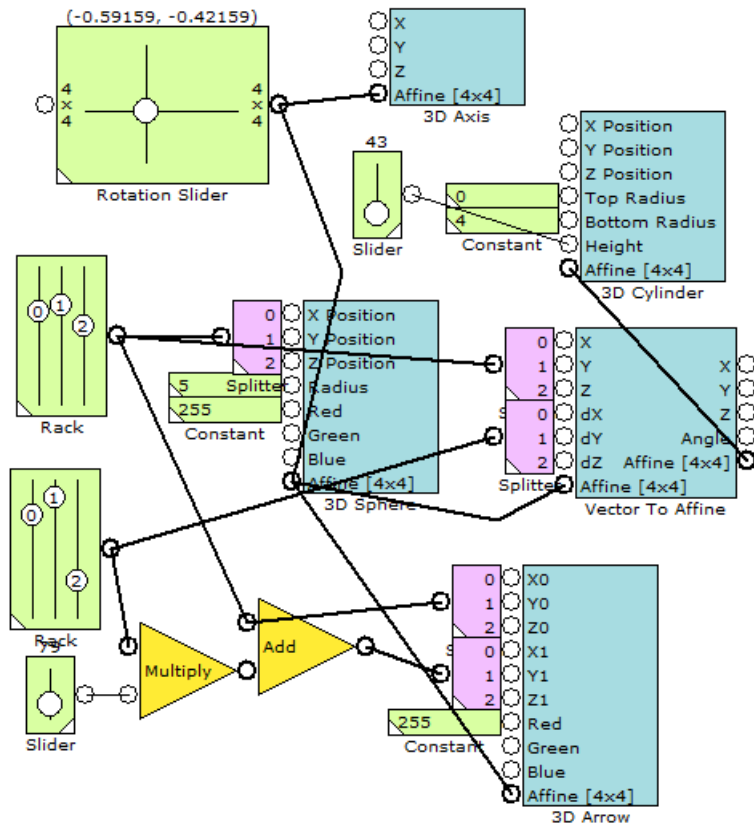
# Vector Scale

The Vector Scale component takes a vector begin and end coordinates with a factor and outputs vector begin and end coordinates scaled by that factor. 2D - column 2



# Vector To Affine

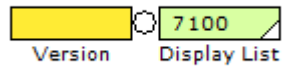
The Vector To Affine component creates an Affine [4x4] matrix that will translate a 3D object to align with the input vector position and direction. Functions – column 8



# Version

The Version component outputs the current software version.

Functions - column 9



## 2D Vertical Axis

The 2D Vertical Axis component draws labeled axes on the 2D canvas. 2D - column 1

2D Vertical Axis

Arrow Size:

Small Tic Size:

Large Tic Size:

Labels on Right

Show Zero Label

Arrow Head

Ignore Display Scale

Pin options:

X  Y

Size

Start  End

Tic  Label

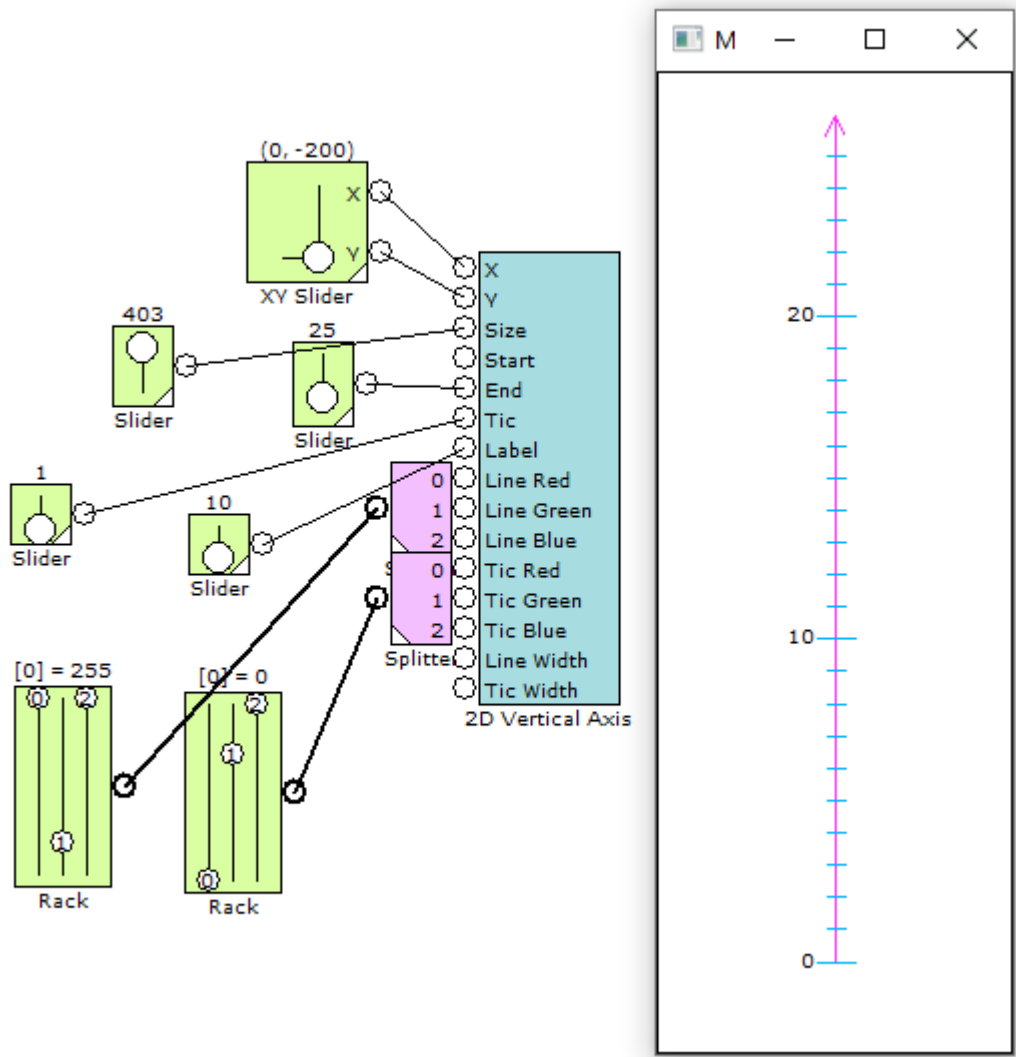
Line Red  Line Green  Line Blue

Tic Red  Tic Green  Tic Blue

Line Width  Tic Width

Label Red  Label Green  Label Blue

Alpha



## 2D Vertical Slider

The 2D Vertical Slider component outputs a number based on its dial position. Controls - column 1

2D Vertical Slider

Minimum:

Maximum:

Detent:

Draw the Track

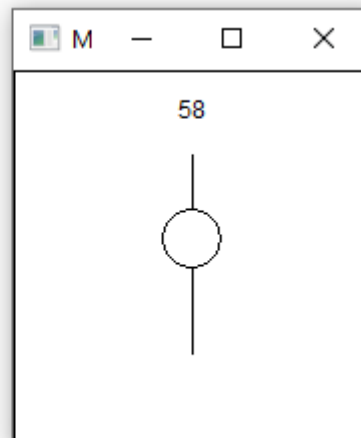
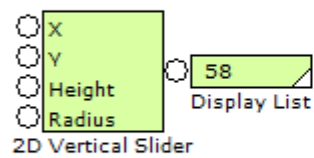
Draw the Knob

Draw the Number

Draw the Graph

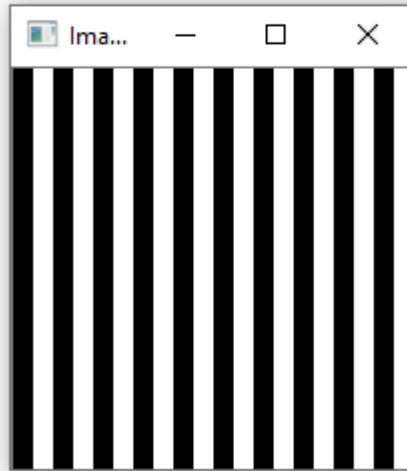
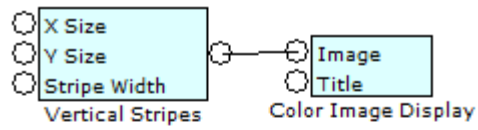
Spring Return

Ignore Display Scale



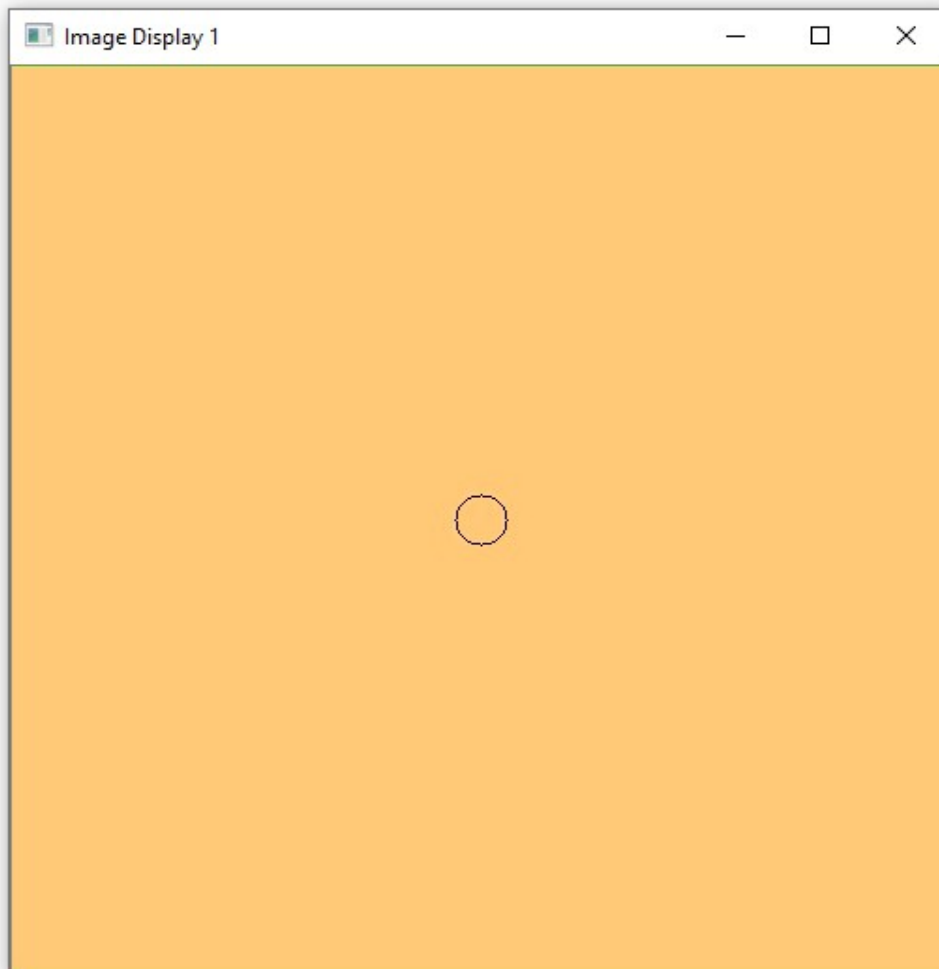
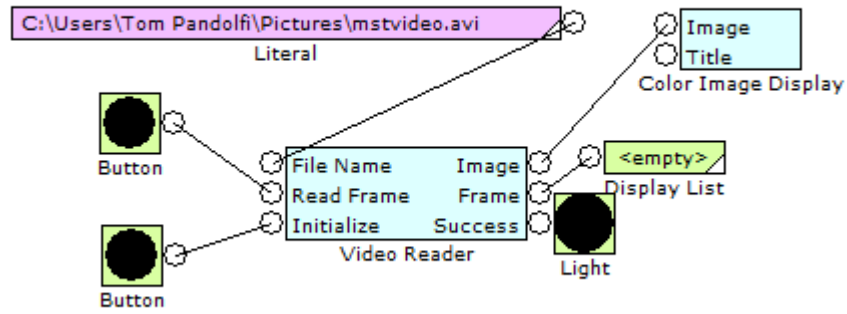
# Vertical Stripes

The Vertical Stripes component creates an image with a vertical striped test pattern. Images - column 1



# Video Reader

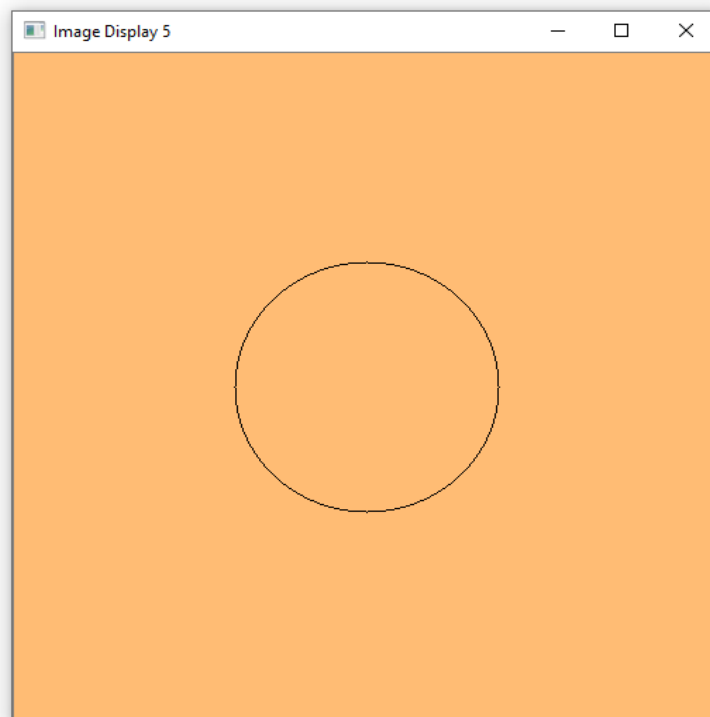
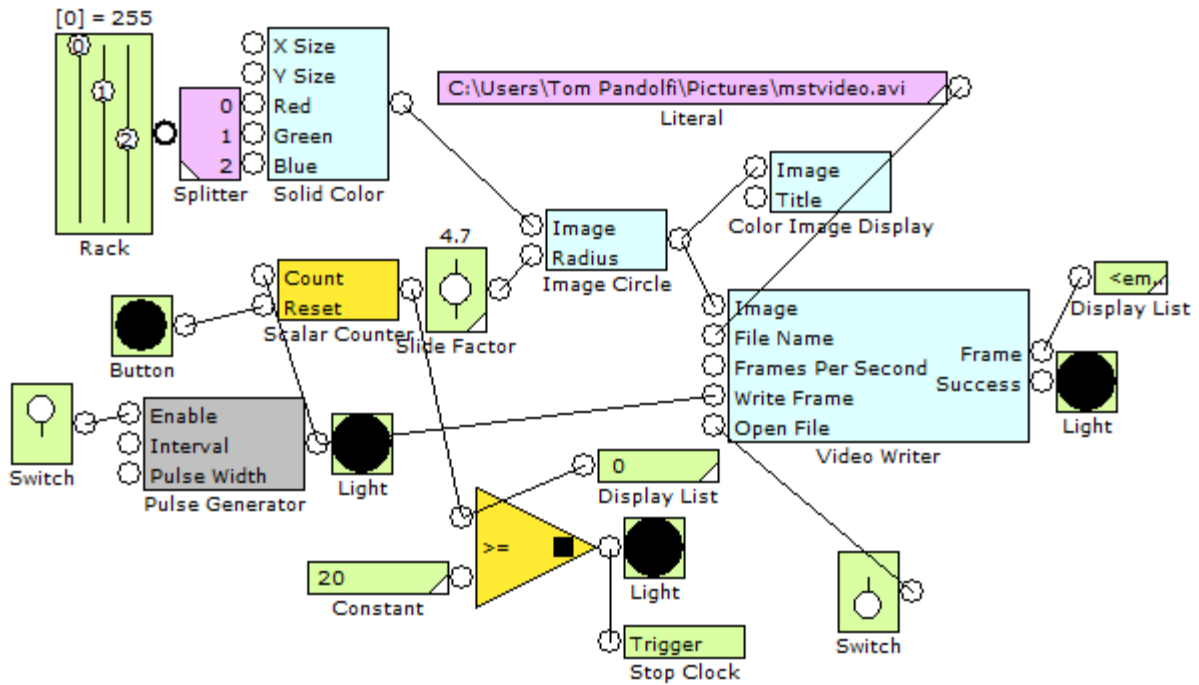
The Video Reader component reads a video file at the path given and outputs it frame by frame. Images - column 5





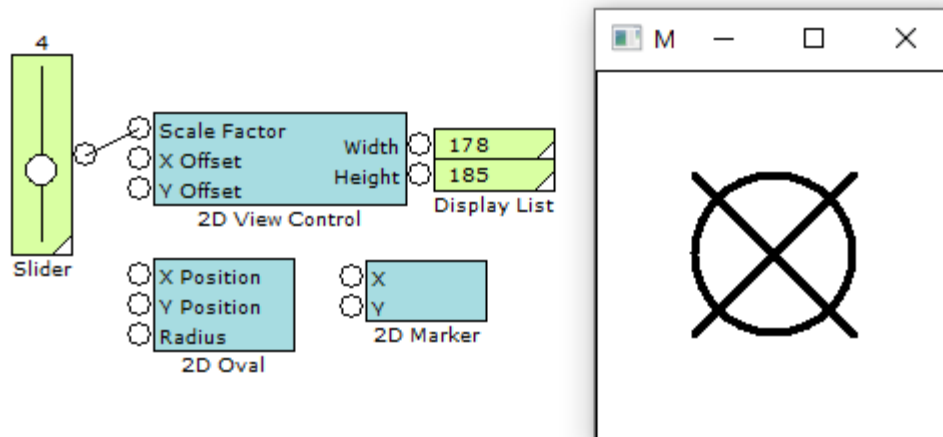
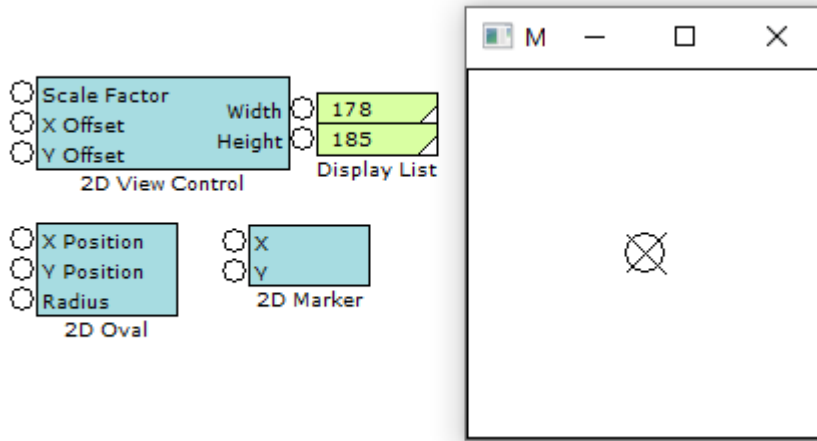
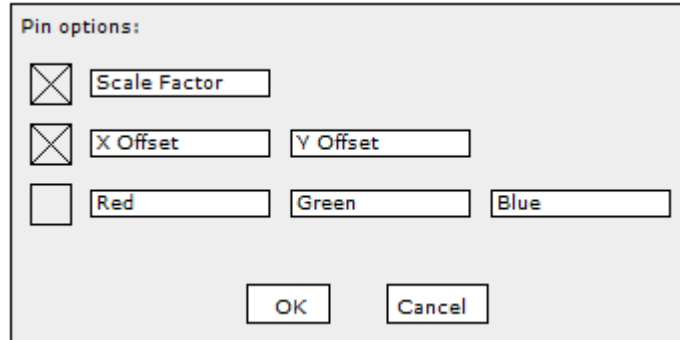
# Video Writer

The Video Writer component writes a series of images to a video file at the path given. Images - column 5



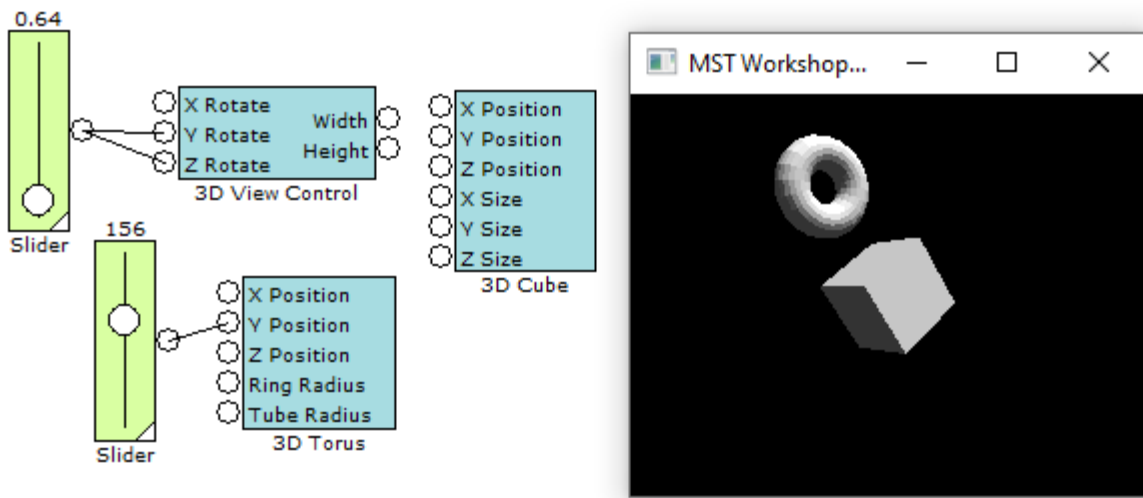
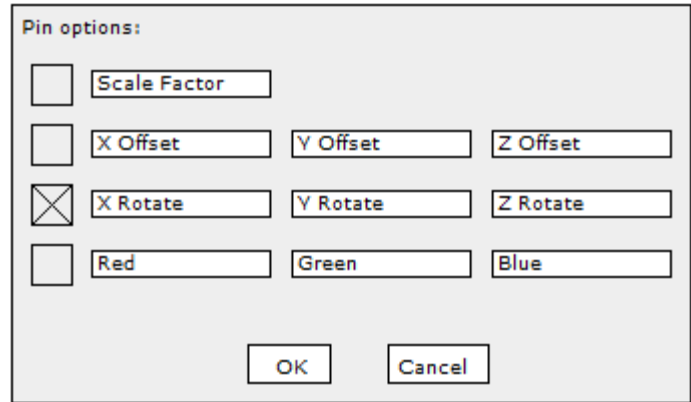
# 2D View Control

The 2D View Control component lets you set the scale, X and Y offsets and size of the 2D display window. 2D - column 1



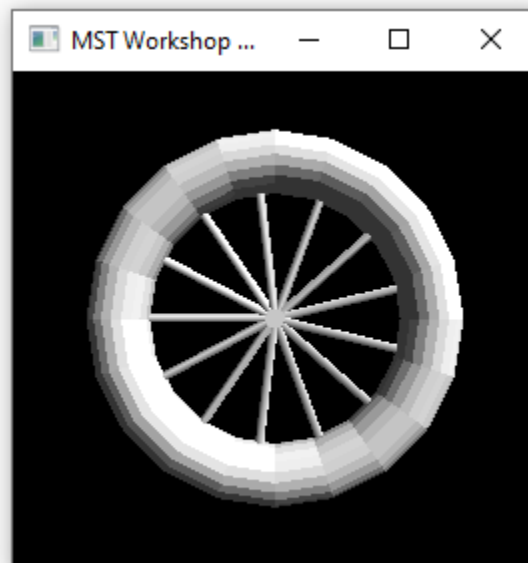
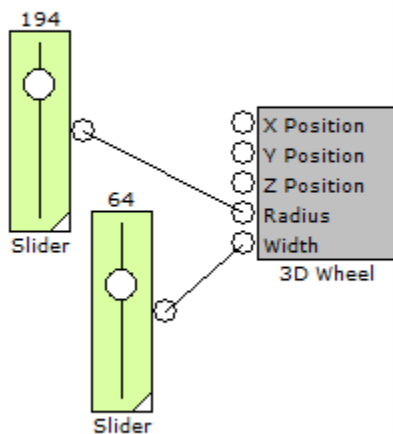
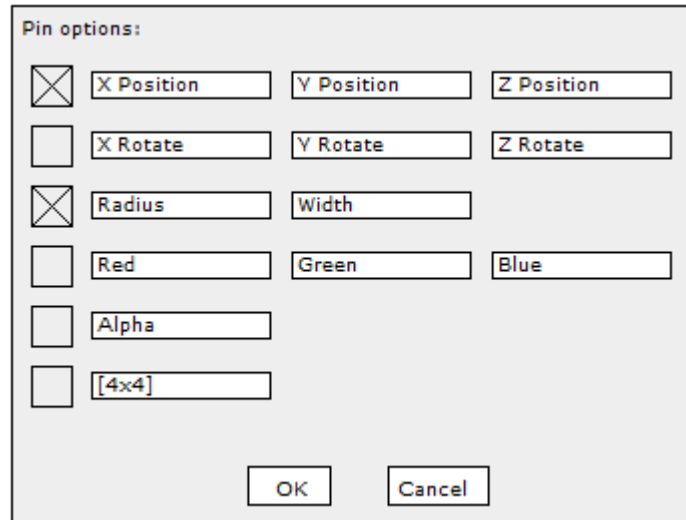
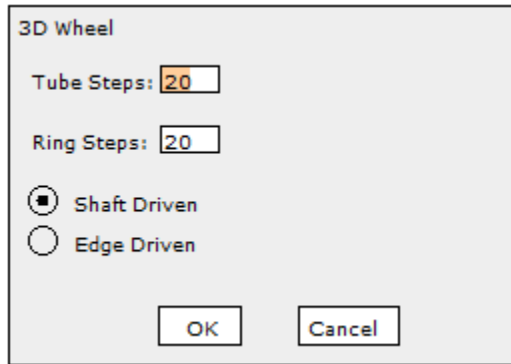
# 3D View Control

The 3D View Control sets the scale, offset, rotation, and background color of the 3D View window.  
3D - column 3



## 3D Wheel

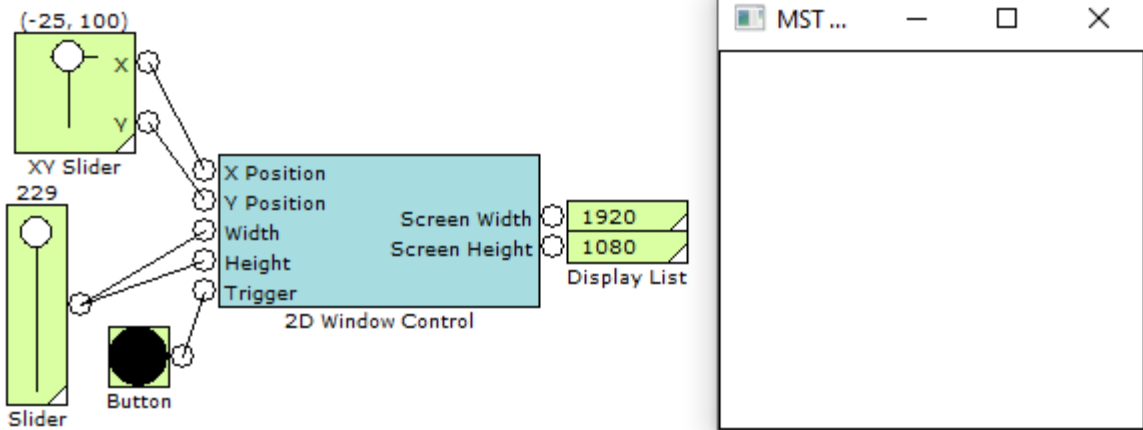
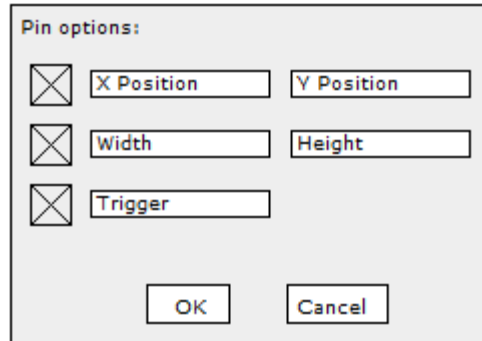
The 3D Wheel component creates a wheel that can attach to other simulation objects like 3D Motors. Place the Wheel at the same X, Y, Z position and orientation as a motor to connect the wheel to the motor. 3D - column 4



# 2D Window Control

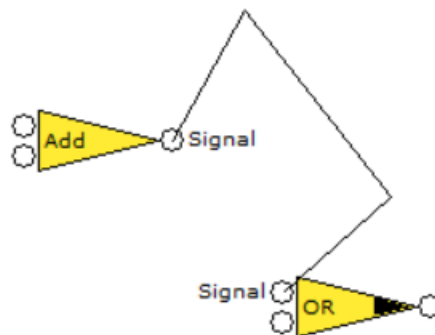
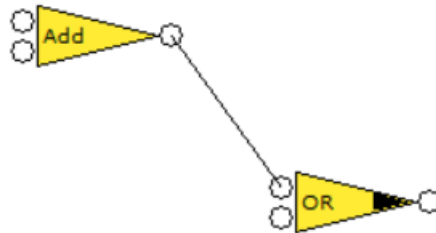
The 2D Window Control component lets you set the position and size of the 2D window.

2D - column 1



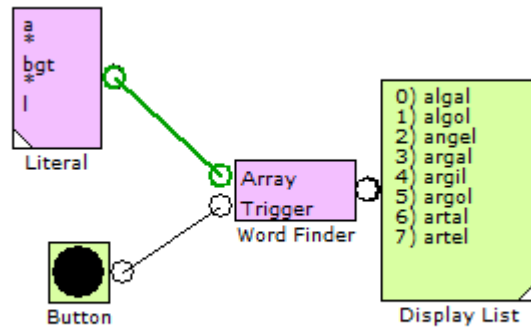
## Wire

The Wires connect components. Wires are created when you click an empty terminal and hold the mouse button while you drag the cursor to another terminal. You can bend wires by clicking on any point along the wire and dragging that point. To remove a bend in a wire Right Click on the bend. To delete the wire, straighten a wire or add a label to a wire Right Click on the wire and choose from the pop-up menu.



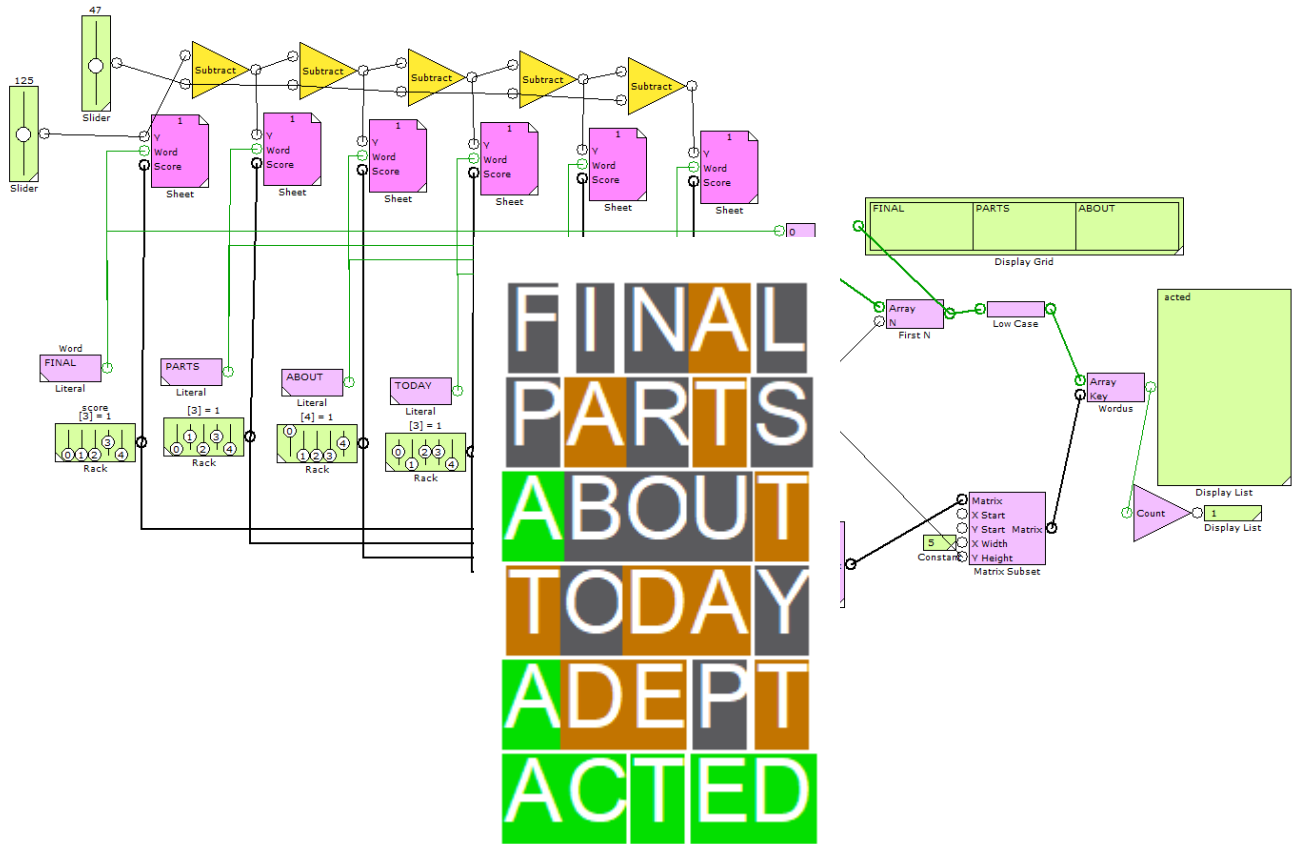
## Word Finder

The Word Finder lists words that fit the input specification. Input should be a set of strings that contains the possible letters desired in each position. The number of input strings is the number of letters in the result. Use a single star to indicate any letter is allowed. Set Trigger to true to generate the list.



# Wordus

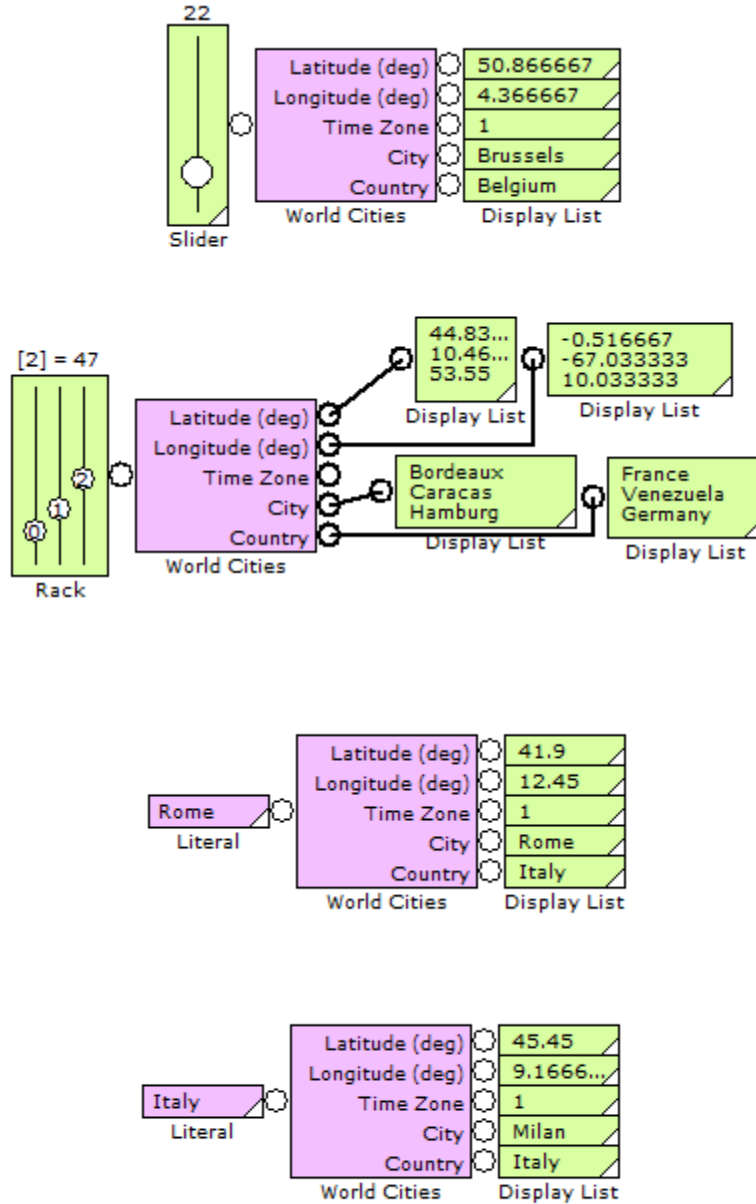
The Wordus recommends words possible in the Wordus game. Connect an array of 6 words in the Array input. Connect 6 switches each of 5 knobs using Join to the Key input. Set the switches to 0 for bad letters, 1 for good letters in wrong position, and 2 for letters in the correct position. Arrays - column 6





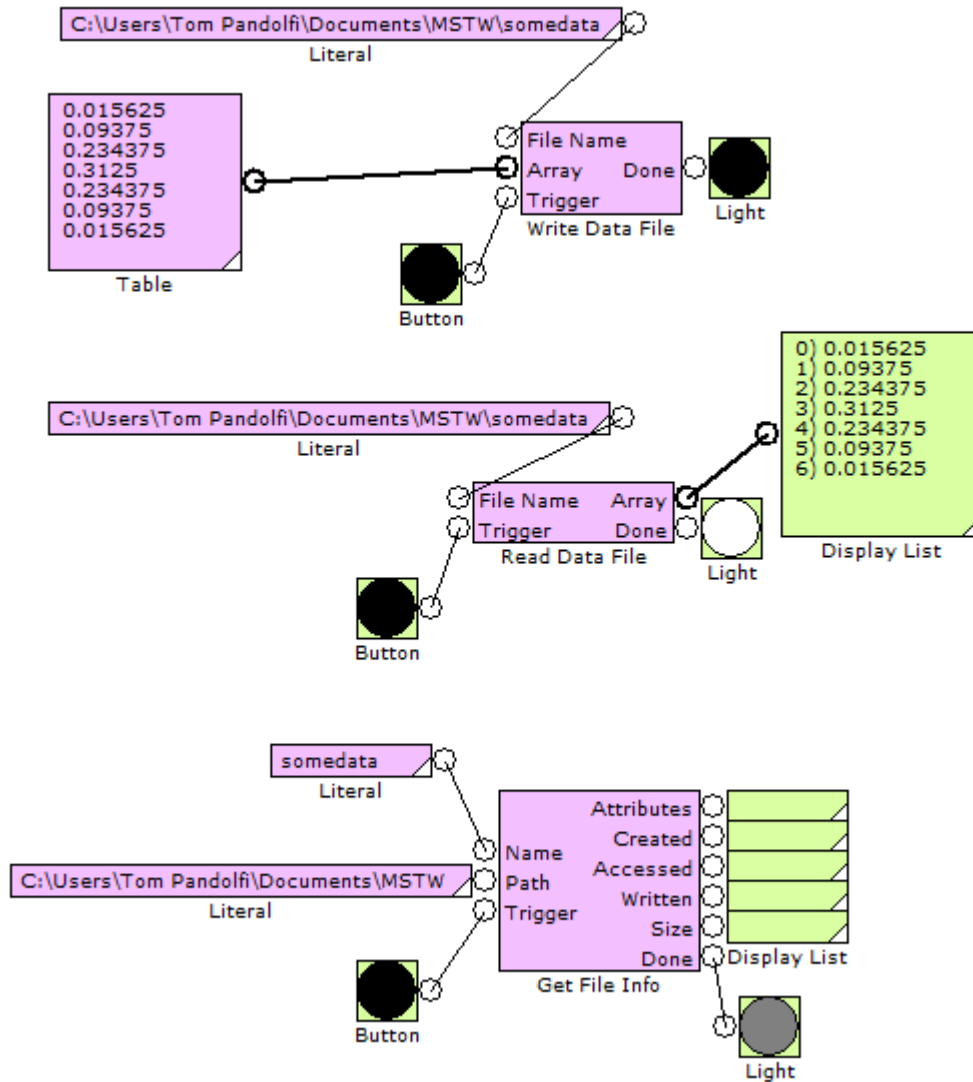
# World Cities

The World Cities component contains Latitude, Longitude, and Time Zone for some cities around the World. Arrays - column 5



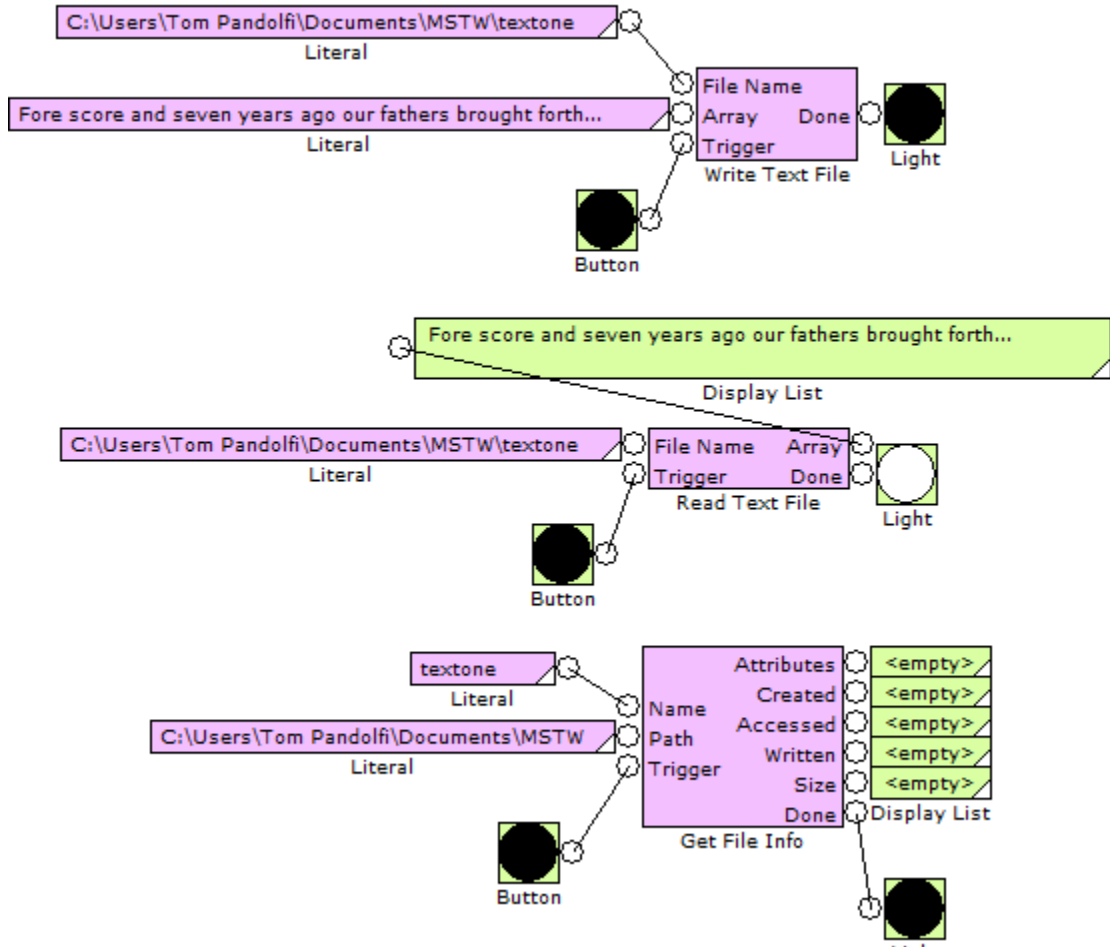
## Write Data File

The Write Data File component writes a binary file. The file can be bytes, words, longs, or doubles, depending on what you specify in the options. You can set an option to append the data to an existing file. Arrays - column 7



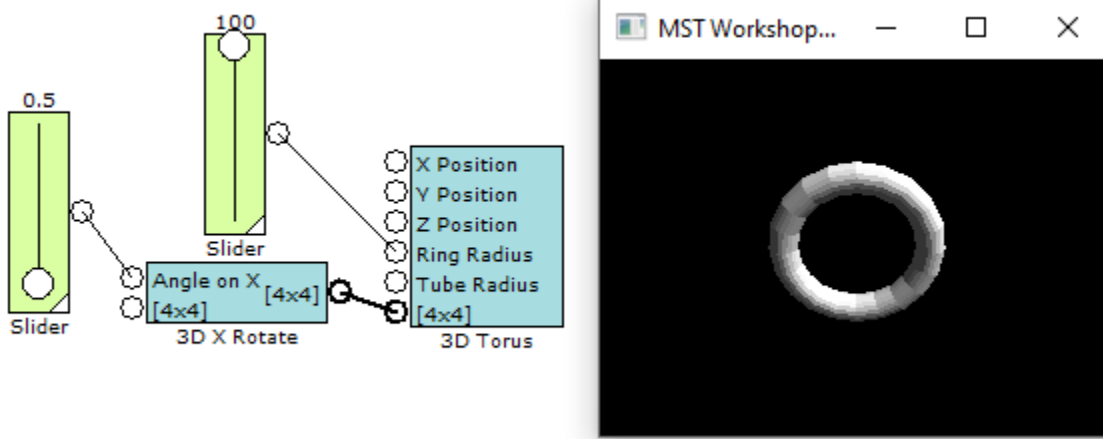
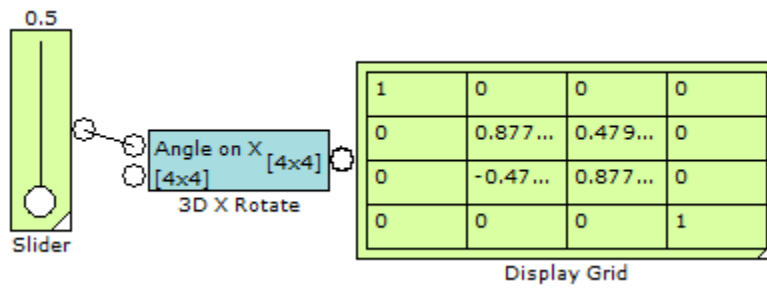
# Write Text File

The Write Text File component writes a text file. You can set an option to append the data to an existing file. Arrays - column 7



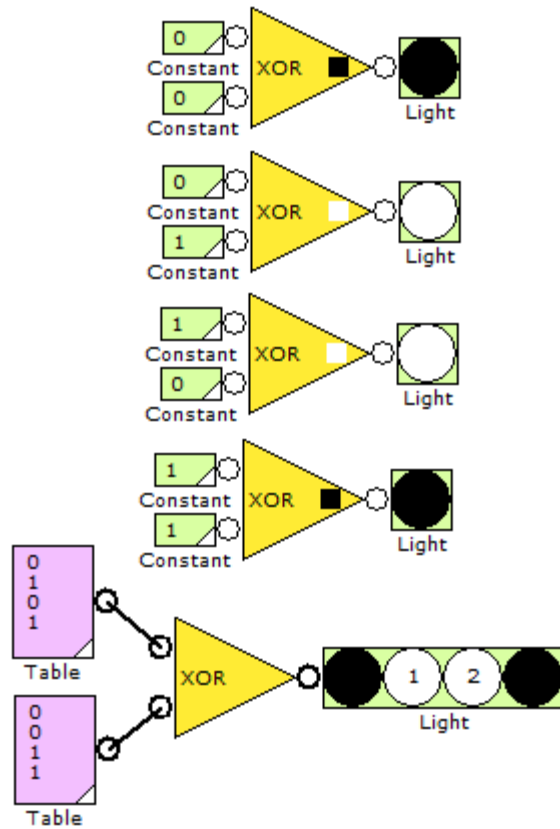
### 3D X Rotate

The 3D X Rotate rotates a 4 by 4 matrix around the X axis by Radians. 3D - column 3



# XOR

The XOR component compares the inputs and returns true if one value is true and the other is false. Inputs can be scalars or arrays. Functions - column 3



# 2D XY Axis

The 2D XY Axis component draws a pair of labeled axes on the 2D canvas. 2D - column 1

2D XY Axis

Arrow Size:

Small Tic Size:

Large Tic Size:

Labels on Top

Show Zero Label

Arrow Head

Ignore Display Scale

Pin options:

X

Size

Range

Tic

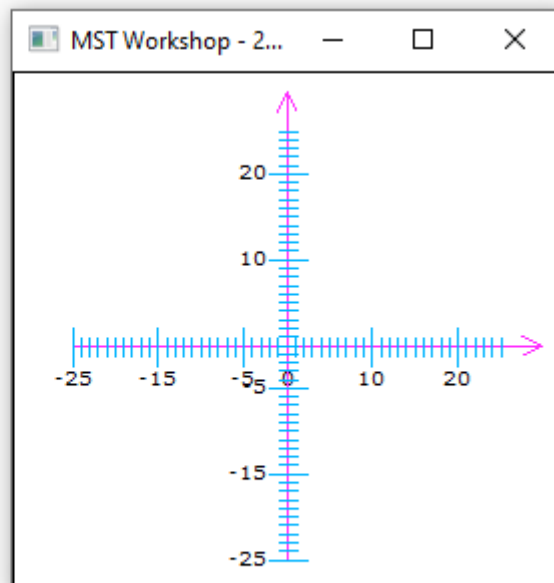
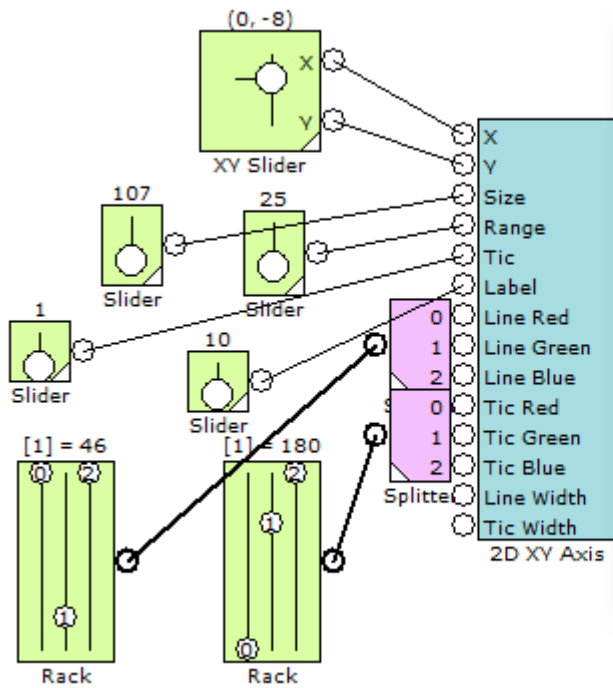
Line Red

Tic Red

Line Width

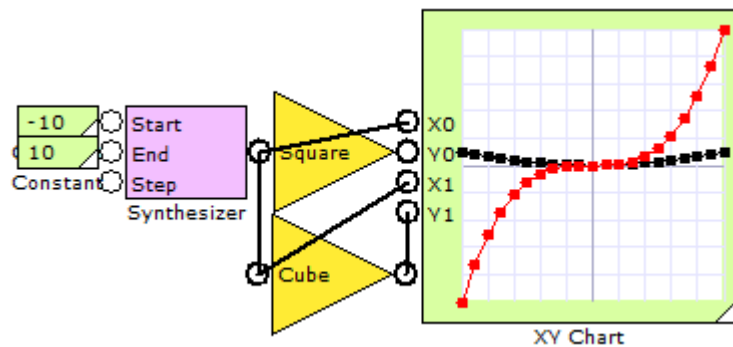
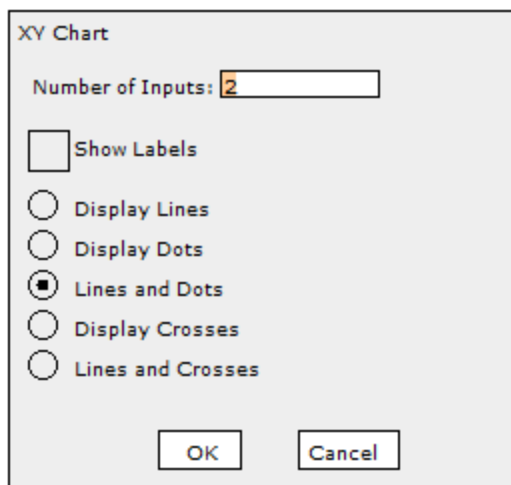
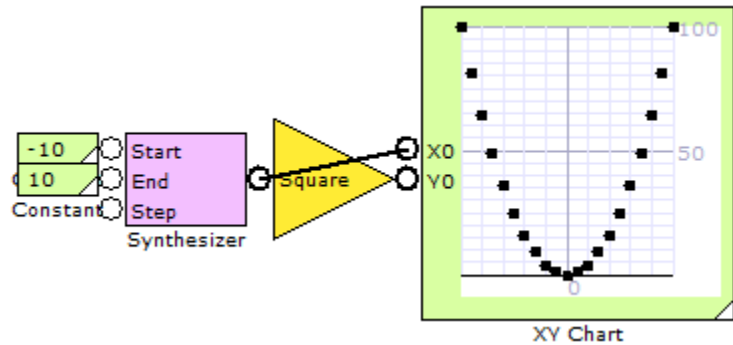
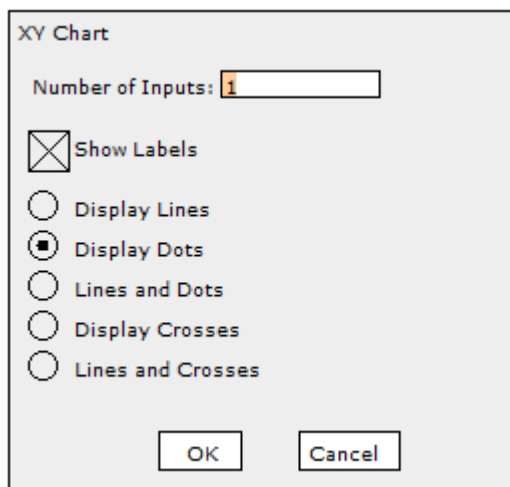
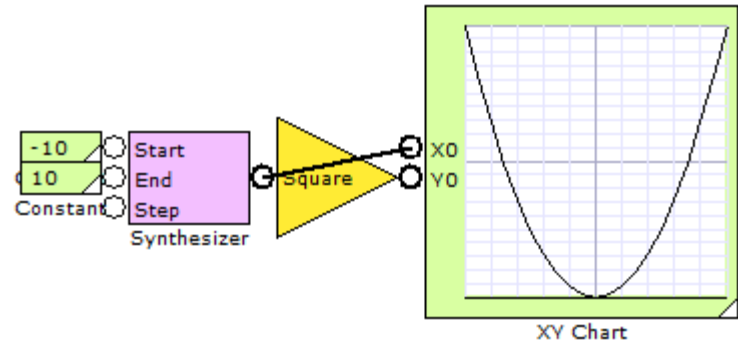
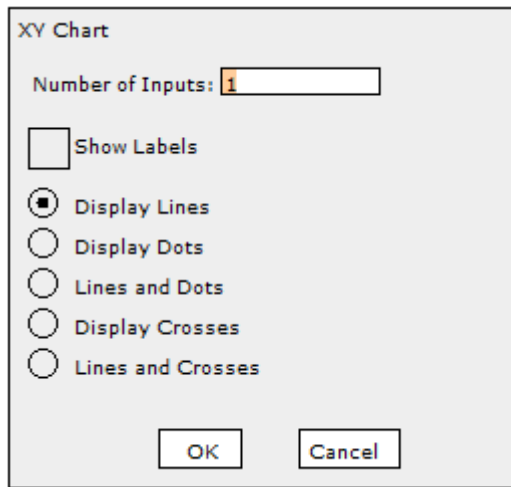
Label Red

Alpha



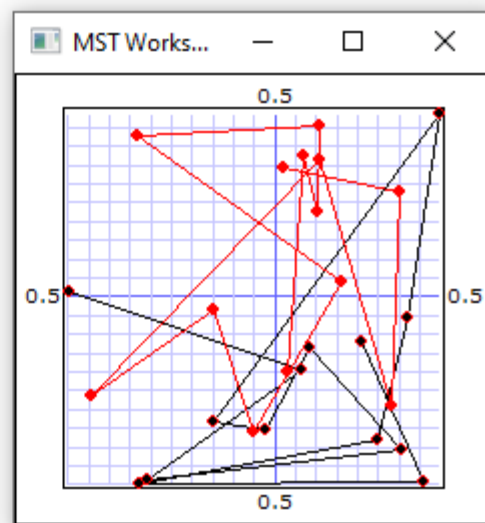
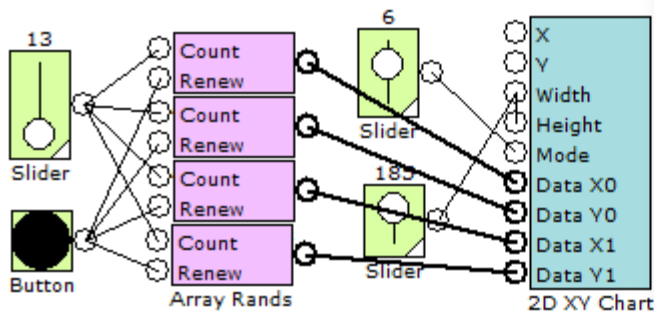
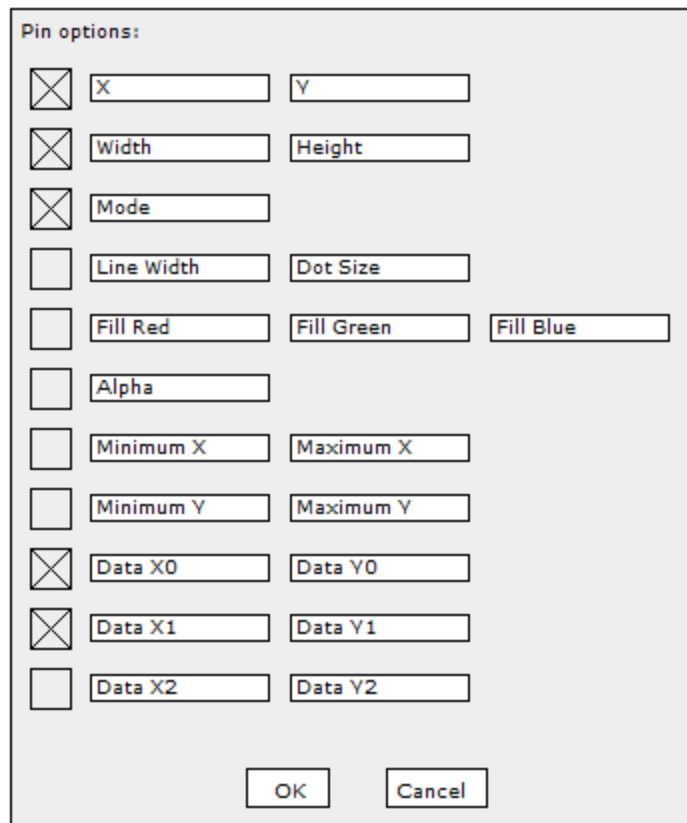
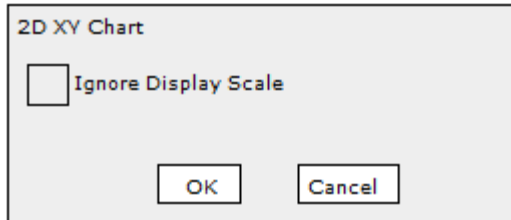
# XY Chart

The XY Chart shows X vs. Y creating a scatter plot of the data. Controls - column 4



## 2D XY Chart

The 2D XY Chart shows X vs. Y creating a scatter plot of the data in the 2D display window. 2D - column 2

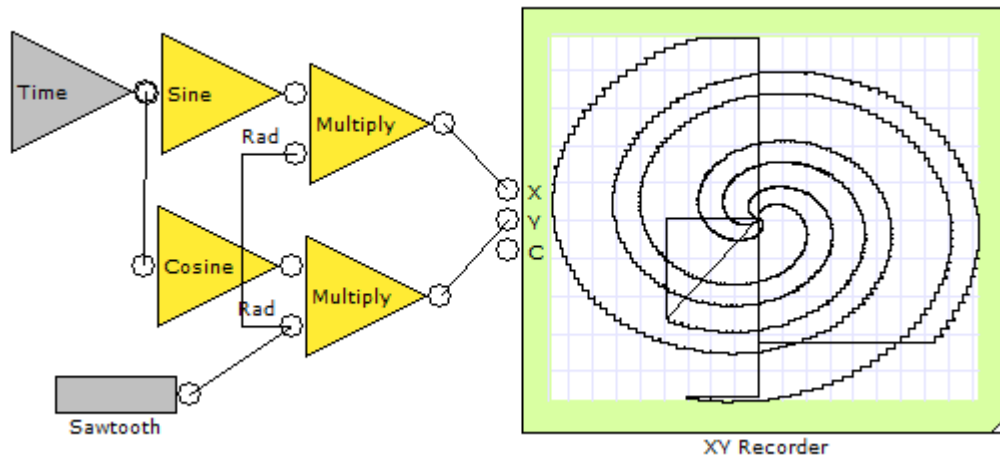




# XY Recorder

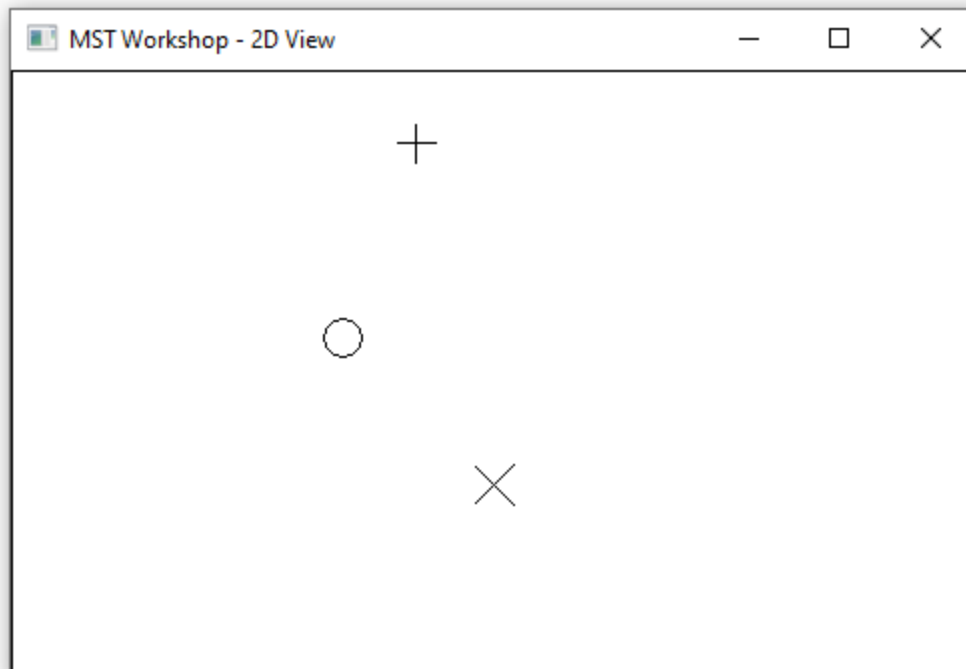
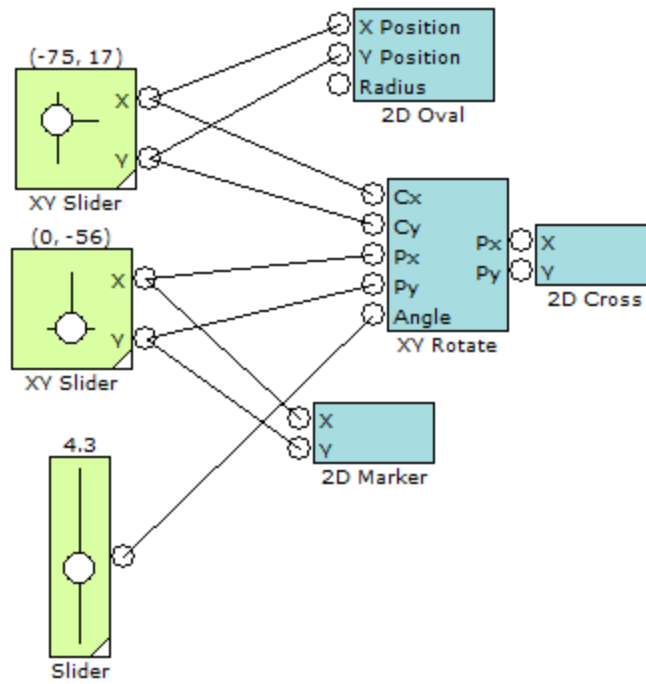
The XY Recorder plots the value of the 2 inputs over time.

Controls - column 4



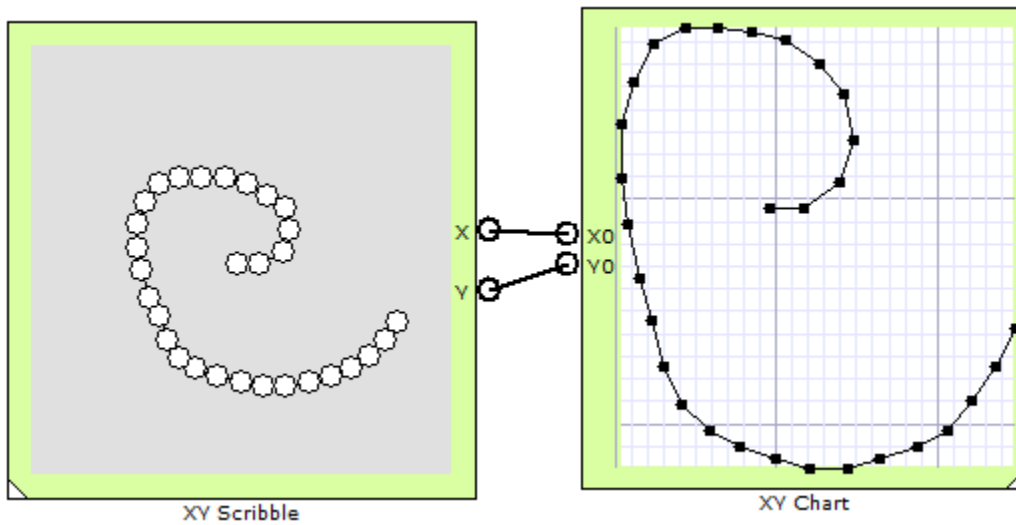
# XY Rotate

The XY Rotate calculates new XY coordinates after a rotation. 2D - column 5



## XY Scribble

The XY Scribble component lets you draw a curve by moving the mouse inside the control. You can add to either end by dragging the end points. Restart by clicking on an empty space. You can adjust the position of any of the inner points after first drawing.



# XY Slide Factor

The XY Slide Factor takes in two inputs and multiplies one by the X position and one by the Y position of the slide. Controls - column 2

Move To Top	Delete
Move Up	Details...
Move Down	Duplicate
Move To Bottom	Edit Label...
Range: 0, 255	Options...
Range: -100, 100	
Range: 0, 100	
Range: -10, 10	
Range: 0, 10	
Range: 0, 2*Pi	
Range: -Pi, Pi	
Range: -1, 1	
Range: 0, 1	

XY Slide Factor

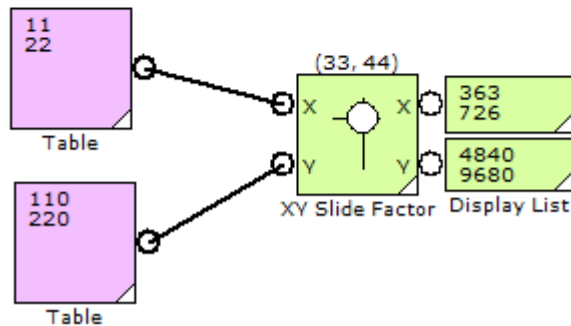
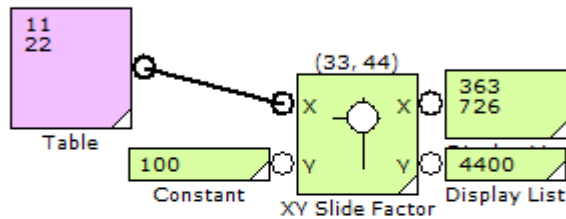
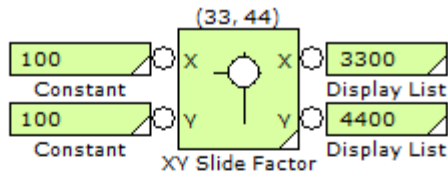
Minimum X:

Maximum X:

Minimum Y:

Maximum Y:

Continuous Action



# XY Slide Offset

The XY Slide Offset adds the X and Y positions of the slider to the two inputs. Controls - column 3

Move To Top	Delete
Move Up	Details...
Move Down	Duplicate
Move To Bottom	Edit Label...
Range: 0, 255	Options...
Range: -100, 100	
Range: 0, 100	
Range: -10, 10	
Range: 0, 10	
Range: 0, 2*Pi	
Range: -Pi, Pi	
Range: -1, 1	
Range: 0, 1	

XY Slide Offset

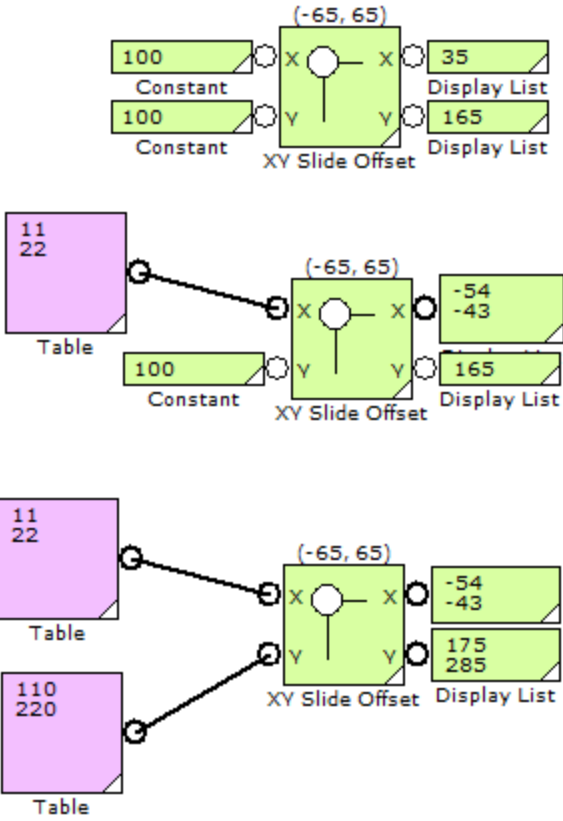
Minimum X:

Maximum X:

Minimum Y:

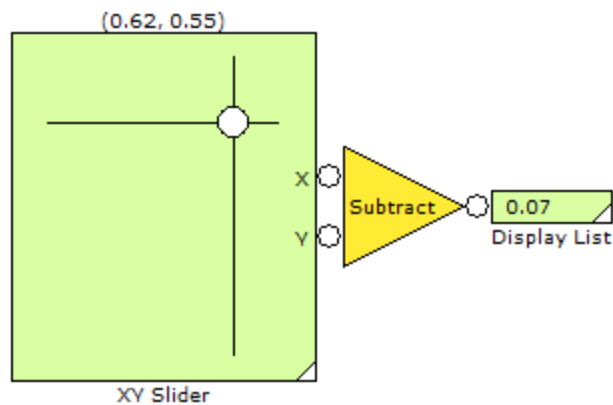
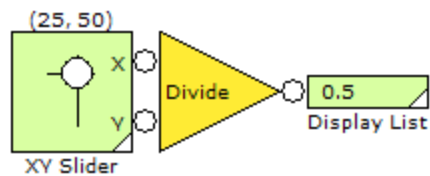
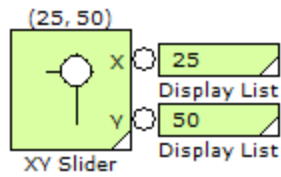
Maximum Y:

Continuous Action



# XY Slider

The XY Slider lets you control two constants in two directions, horizontal (x) and vertical (y).  
Controls - column 1



Move To Top	Delete
Move Up	Details...
Move Down	Duplicate
Move To Bottom	Edit Label...
Range: 0, 255	Options...
Range: -100, 100	
Range: 0, 100	
Range: -10, 10	
Range: 0, 10	
Range: 0, 2*Pi	
Range: -Pi, Pi	
Range: -1, 1	
Range: 0, 1	



XY Slider

Minimum X:

Maximum X:

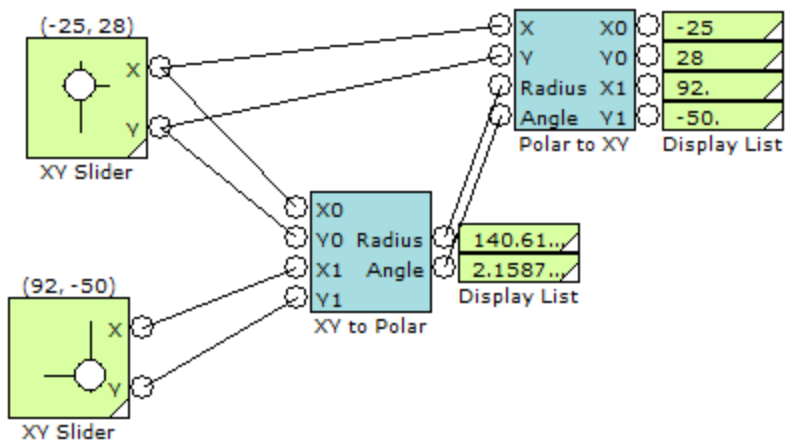
Minimum Y:

Maximum Y:

Continuous Action

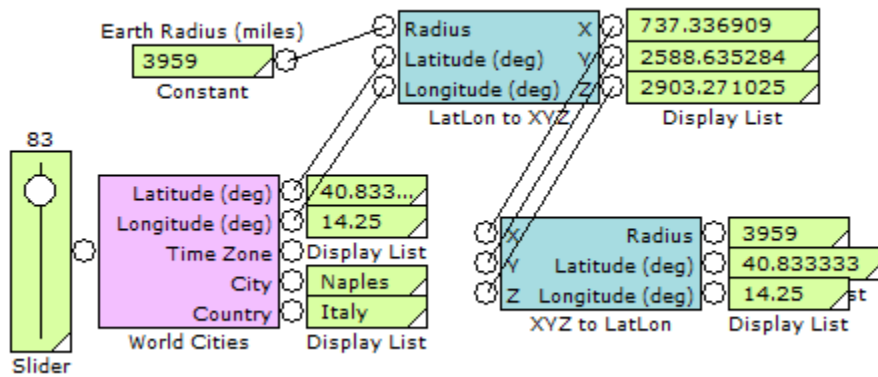
# XY to Polar

The XY to Polar component takes an X,Y coordinate and returns a radius and angle. 2D - column 5



# XYZ to LatLon

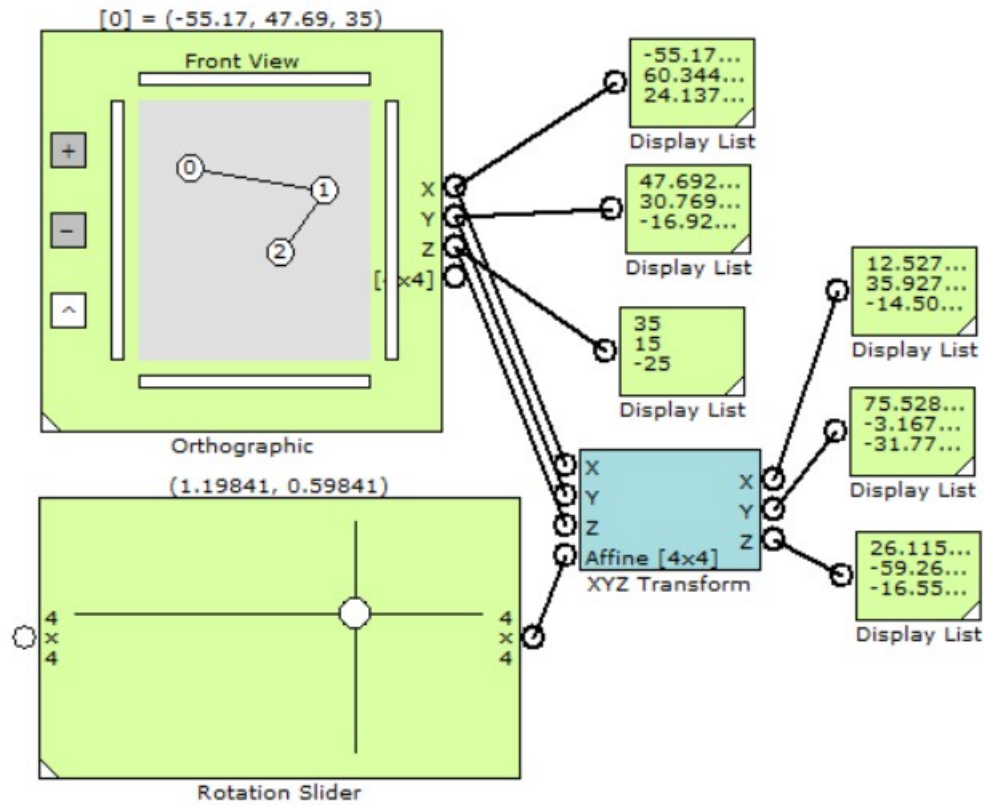
The XYZ to LatLon component takes an X,Y,Z coordinate and returns a Radius, Latitude and Longitude. 3D - column 6





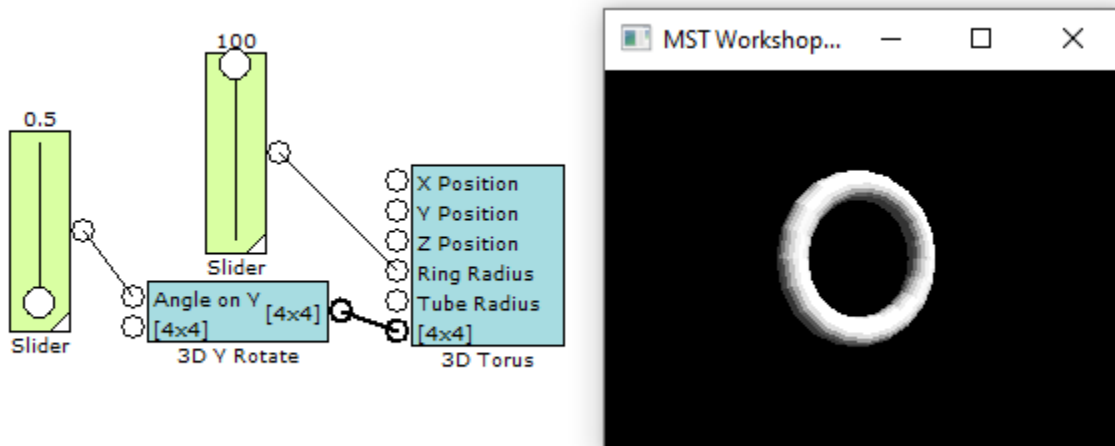
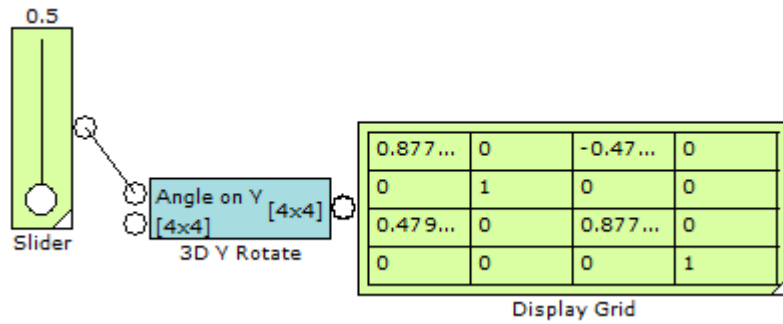
# XYZ Transform

The XYZ Transform takes a point's X, Y and Z coordinates, transforms them with the Affine [4x4] input and outputs new X, Y, and Z coordinates. 3D – column 5



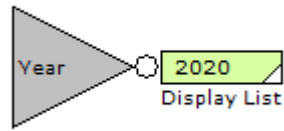
### 3D Y Rotate

The 3D Y Rotate rotates a 4 by 4 matrix around the Y axis by Radians. 3D - column 3



# Year

The Year component outputs the current calendar year. Functions - column 7



# 3D Z Rotate

The 3D Z Rotate rotates a 4 by 4 matrix around the Z axis by Radians. 3D - column 3

