Delta Media Server

7thConnect

User Guide

7thConnect : User Guide

Trademark Information

The 7thsense logo, and various hardware and software product names are trademarks of 7thSense Design Ltd. Product or company names that may be mentioned in 7thSense publications are tradenames or trademarks of their respective owners, and such trademarks may also be registered in their respective countries. Windows is a registered trademark of Microsoft Corporation in the United States and other countries.

Copyright Information

The information in this document is subject to change without notice. 7thSense Design Ltd assumes no responsibility for errors, and/or omissions contained in this information.

Printed: February 2021

This edition is for software version 7thConnect 1.1.0.3
Document ref.: M171-4

E: info@7thsense.one
W: 7thsense.one

7thSense Design Ltd
2 The Courtyard, Shoreham Road
Upper Beeding
Steyning
West Sussex
BN44 3TN
UK

T: +44 (0) 1903 812299

7thSense Design LLC, Michigan
332 E Lincoln Ave
Suite 100
Royal Oak, MI 48067
USA

T: +1 248 599 2717

7thSense Design LLC, Orlando
4207 Vineland Rd
Suite M1
Orlando, FL 32811
USA

T: +1 407 505 5200

Contents

Overview

7thConnect is a powerful, single application, used to connect multiple I/O feeds for use with Delta Media Server system.

An an application builder, 7thConnect offers real-time control of Delta from many popular I/O devices such as Kinect®, Leap Motion®, Phidget® I/O devices and TCPIP, and can remove the requirement for an external show controller.

Features and Applications

- Add simplicity to your timeline control via robust physical button inputs.
- Security or tamper avoidance via near-field RFID triggering, or even personalised / language specific media playback.
- Sophistication via motion-based triggering rather than traditional looping.
- Intelligence and audience participation from audience voting systems.
- Multiple Phidgets® can be added to single connect environment for audience voting systems.
- 7thConnect can run from a control PC and communicate with multiple Delta Media Servers.
- Features all the necessary tools to scale and manipulate input data for use as global variables, which may be used by sequences and external controls.

Inputs

Real-time timeline / sequence triggering from wide range of off the shelf input components; IR beams, pressure pads,light sensors, microphones, buttons, knobs and sliders etc.

Outputs

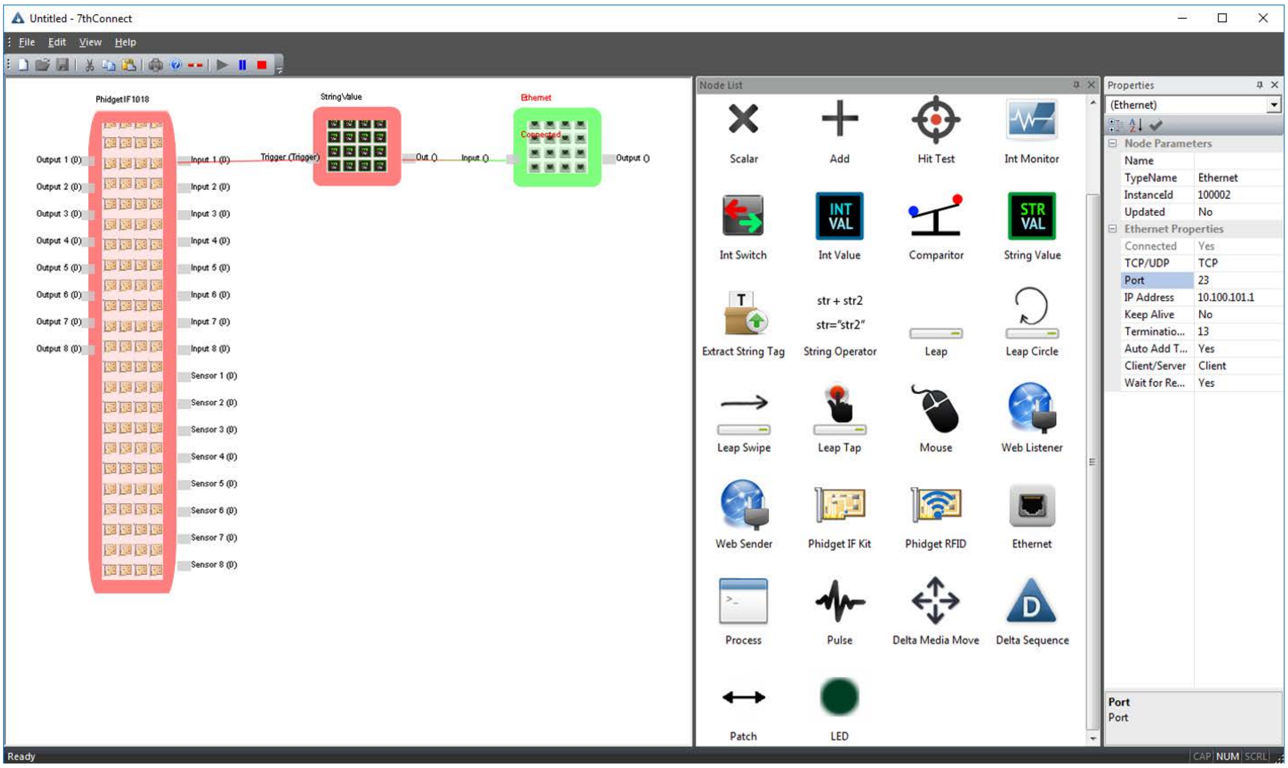
Seamless interaction with open / closed 0v-5v contacts, automatic door systems, and fake snow blowers.

How 7thConnect Works

A library of Nodes (14) is available that the user can drag onto the canvas and connect up to create a system.

Below is a very simple example of a system. The system consists of a Phidget IO controller Node, a String Value Node and an Ethernet Node.

A connection has been made from ‘Input 1’ of the IO Phidget board to the trigger input on a String Value Node. There is then a further connection from the String Value to an Ethernet Node.

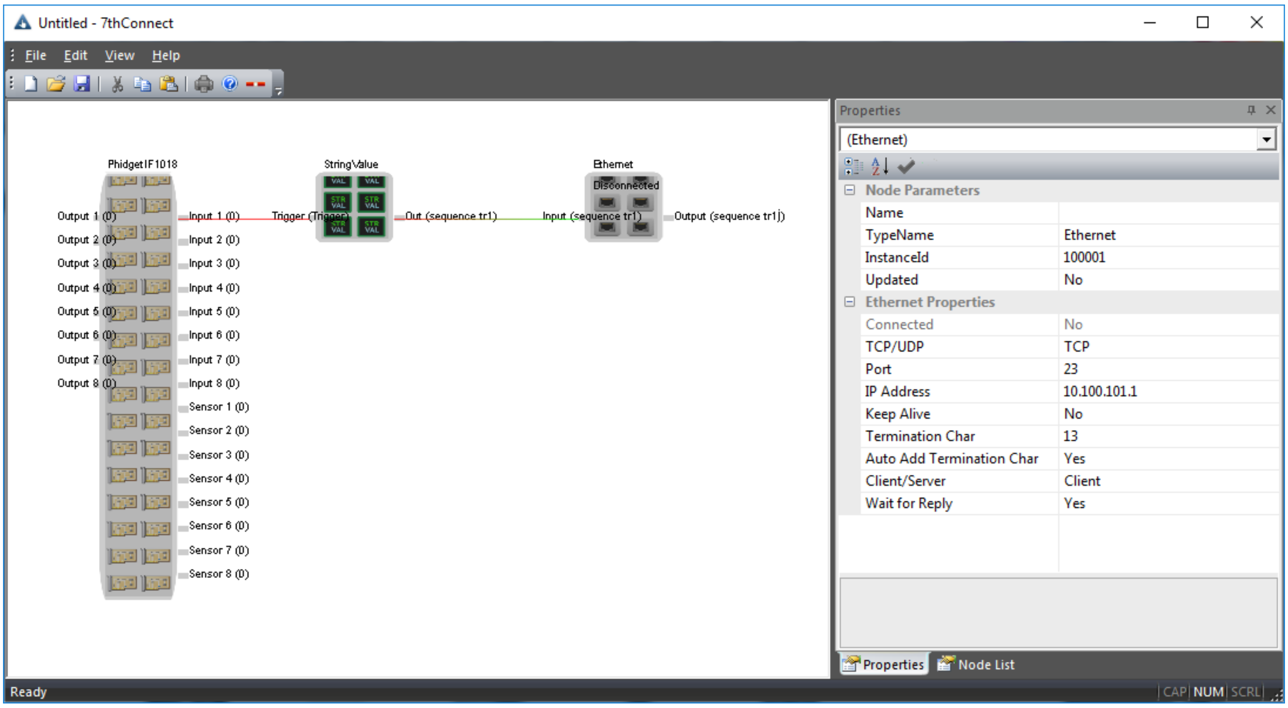


Node Parameters	
Name	
TypeName	Ethernet
InstanceId	100002
Updated	No
Ethernet Properties	
Connected	Yes
TCP/UDP	TCP
Port	23
IP Address	10.100.101.1
Keep Alive	No
Terminatio...	13
Auto Add T...	Yes
Client/Server	Client
Wait for Re...	Yes

Port
Port

The result of this system is that when the user triggers input 1, through a physical connection such as a contact closure or switch, a signal is passed and triggers the string value to send its data. The data is received by the Ethernet node and processed. In this example, the Ethernet node is configured as a client TCP connection to IP address 10.100.100.1 port 23. The Ethernet is also configured to automatically append a carriage return onto the data as this is a common requirement for Delta Media server.

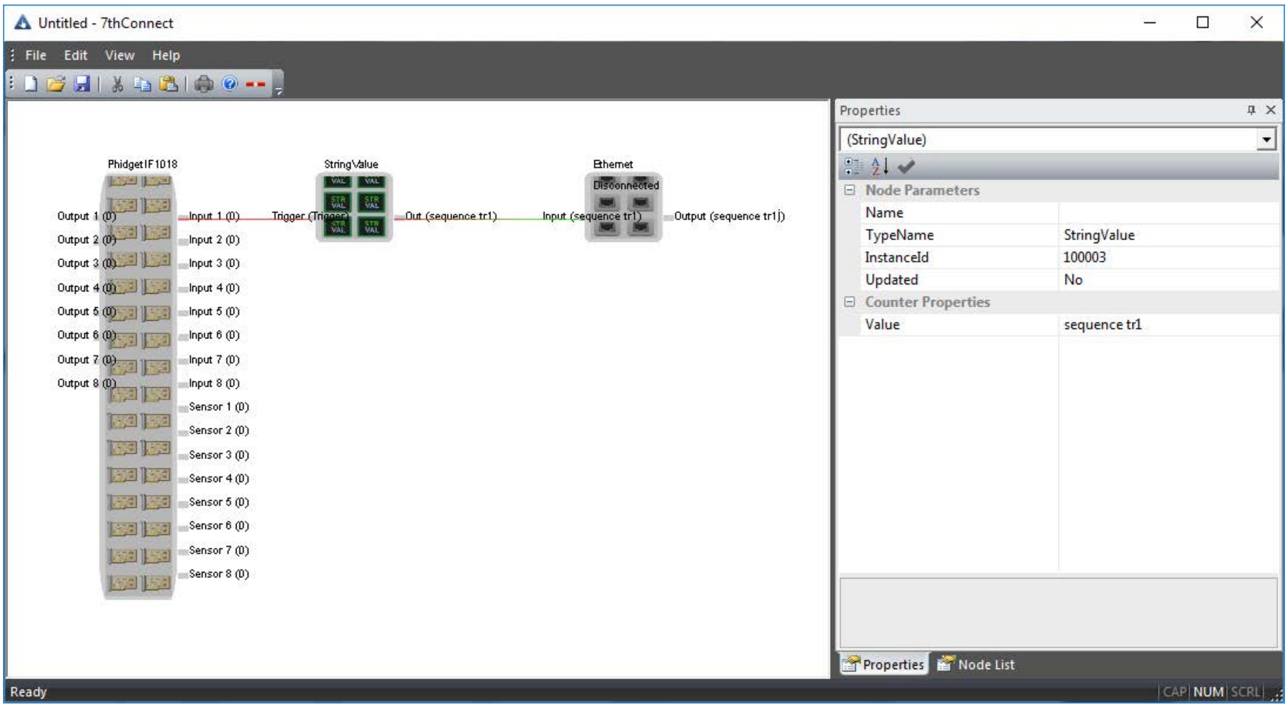
Each node in the system has a set of properties that can be accessed through the Properties docking window. Below are examples of the properties set for the String and Ethernet nodes for this configuration.



Properties

(Ethernet)

Node Parameters	
Name	
TypeName	Ethernet
InstanceId	100001
Updated	No
Ethernet Properties	
Connected	No
TCP/UDP	TCP
Port	23
IP Address	10.100.101.1
Keep Alive	No
Termination Char	13
Auto Add Termination Char	Yes
Client/Server	Client
Wait for Reply	Yes



Properties

(StringValue)

Node Parameters	
Name	
TypeName	StringValue
InstanceId	100003
Updated	No
Counter Properties	
Value	sequence tr1

Installation

7thConnect is a licenced application for 7thSense Delta Media Servers.

The software for 7thConnect can be found on your Delta Server in:
7thSense Data\Software & Drivers\Delta\Utilities\7thConnect. There are three components to install:

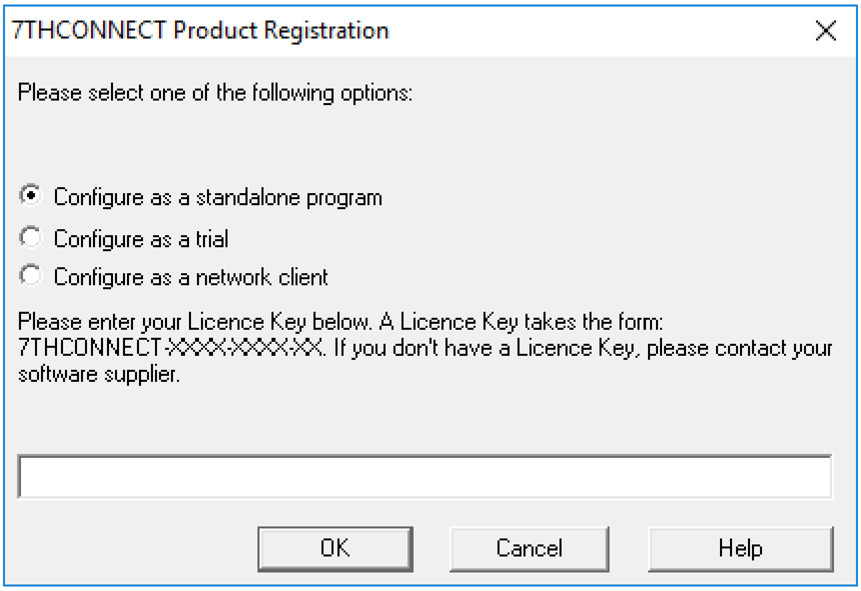
- bonjoursdksetup.exe
- Phidget-x64_2.1.8.20140428.exe
- 7thConnect 1.1.0.3 Setup.exe

Start by installing Bonjour.

The 7thConnect installation will include installation of the Phidget Libraries. Follow through the usual dialogs and Finish.

From the destination installed folder (default: C:\Program Files\7thSense\7thConnect) send a shortcut to your desktop.

You will be prompted to register 7thConnect, using Copyminder, so you will need to have Internet access. Select ‘Configure as standalone program’, and add your licence code:



7THCONNECT Product Registration

Please select one of the following options:

● Configure as a standalone program
○ Configure as a trial
○ Configure as a network client

Please enter your Licence Key below. A Licence Key takes the form: 7THCONNECT-XXXX-XXXX-XX. If you don't have a Licence Key, please contact your software supplier.

Run 7thConnect

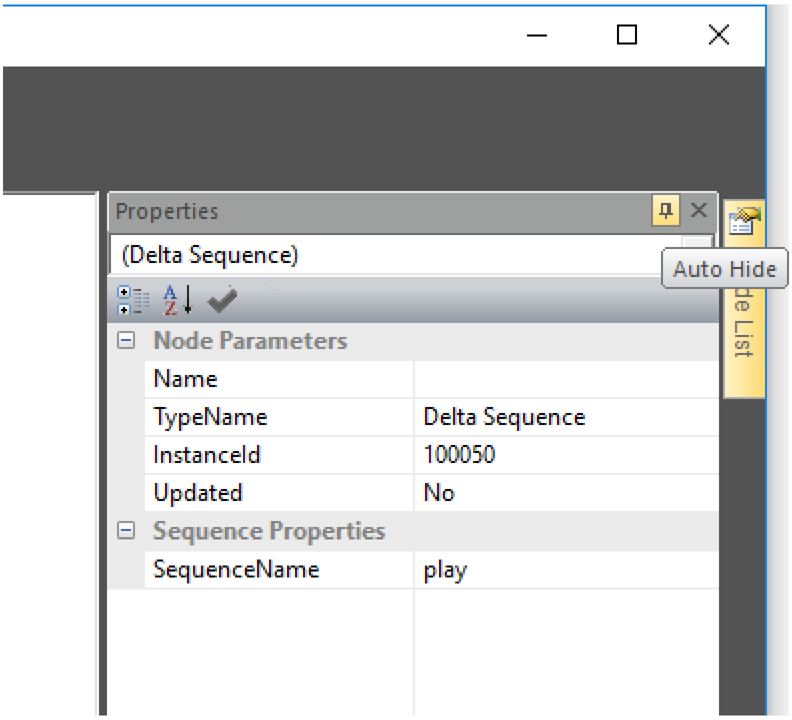
Run DeltaServer and load your show first, then create or load your .7cn node system into 7thConnect. This will ensure the right connections to Delta are made.

Application Tools and Navigation

The 7thConnect working space consists of three panes: the main system space (a blank canvas to receive nodes and connections), a Nodes pane, and a Properties pane.

Undock the Nodes List or Properties by dragging them away, redock them by double-clicking their title bars.

For more working space, autohide either pane with the ‘pin icon’, and slide them out by selecting the new side tab:



(Delta Sequence)

Node Parameters

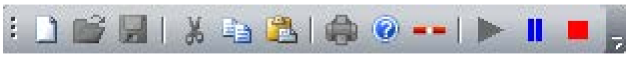
Name	
TypeName	Delta Sequence
InstanceId	100050
Updated	No

Sequence Properties

SequenceName	play

Toolbar

The toolbar buttons for file and edit operation follow convention, as are their normal Ctrl+ shortcut keys (New, Open, Save, Cut, Copy Paste, and Print).



The short red bar deletes a selected node or connection.

You can Load and Save configurations as .7cn files. This is only available with a full licence.

Play, Pause, Stop

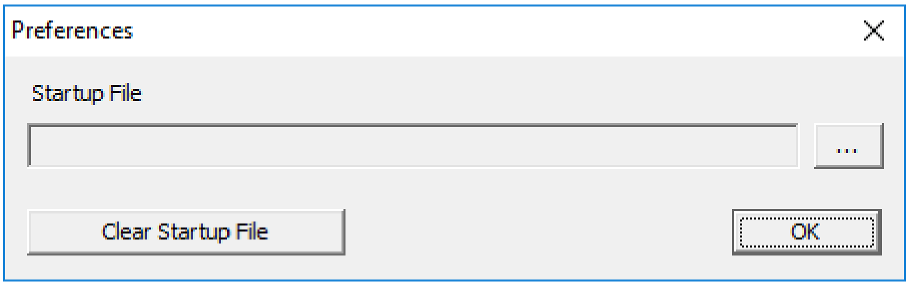
For dynamic nodes, you may need to stop and start. Play will resume where the play point was paused, or at the start point if stopped.

Menu

View > Reset Layout

If you have very many nodes, the system space can fill up. Once connections have been made, you can compact the view so that space between nodes is minimised. Advisory: This may make your system less readable when constructing it. (There is no undo.)

Edit >Preferences > Startup



A startup file can be set to run whenever Connect is run.

System Pane Navigation

Add a node

Drag and drop a node type from the Nodes List into the system pane.

Move a node

Left click and hold to drag the node around.

Move multiple nodes

Hold Ctrl while selecting nodes with left-click. Hold Ctrl to move them all together using the left mouse button.

Move whole layout

Left-click then hold Ctrl, will drag the whole system around on the background.

Delete a node

Left click the node (its label turns red) and press delete, or delete from the toolbar or Edit menu.

Node properties

In the Properties pane, nodes can be named; they can variously have user properties, inputs and outputs.

Make a connection

Drag out a **connection** between an input and an output (contact points turn purple when the connection is made: release the mouse button).

Delete a connection

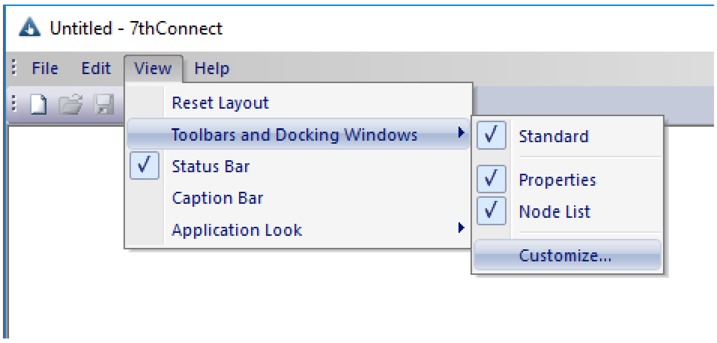
Left-click the connection line (it turns dark green) and press delete, or delete from the toolbar, or Edit menu.

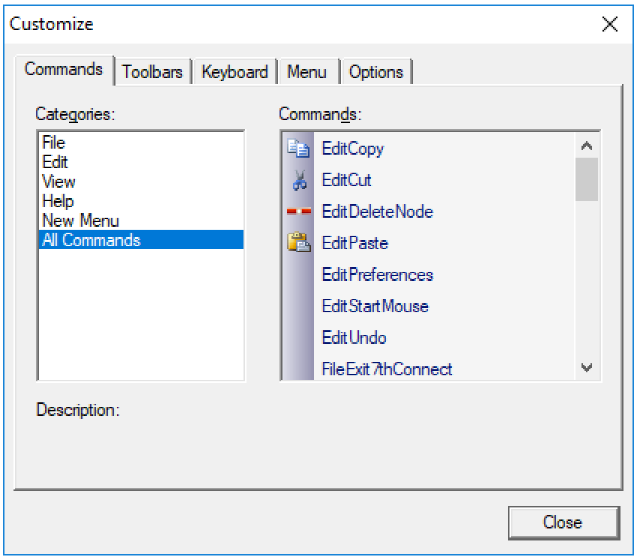
Zoom

The mouse wheel of track pad will zoom in and out.

Customise 7thConnect

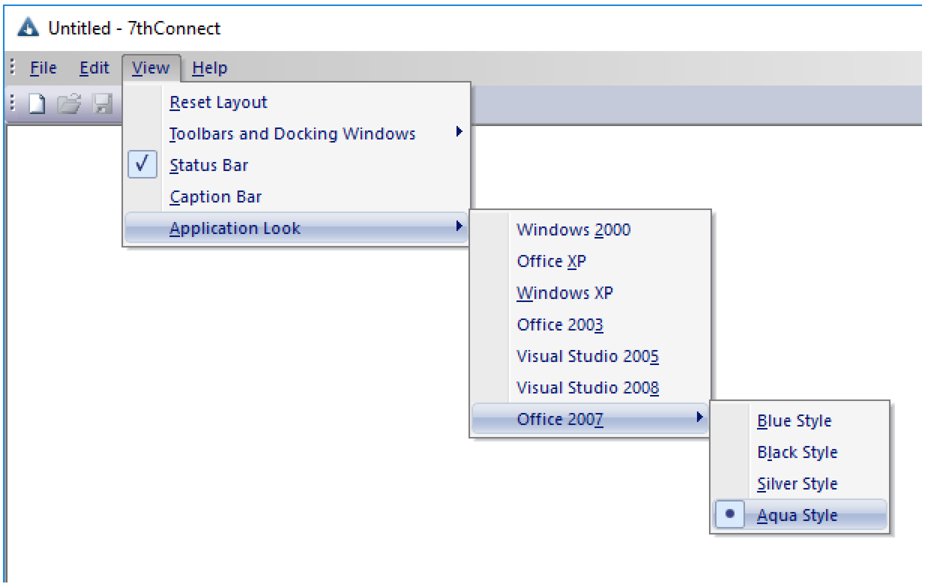
The interface is fully customisable:



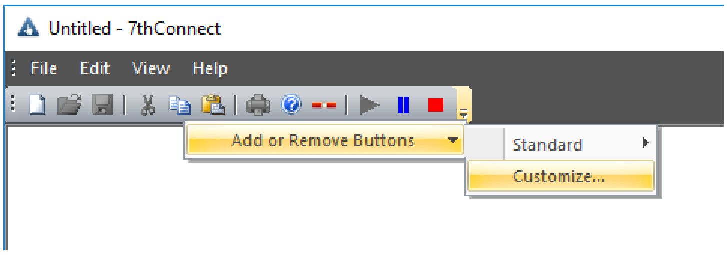


To remove items from menus or toolbars, just drag them off while this is dialog open. To add them, drag from the dialog to the position you want. They can all be reset.

You can also select the overall appearance:



The main toolbar is customisable, including button captions:

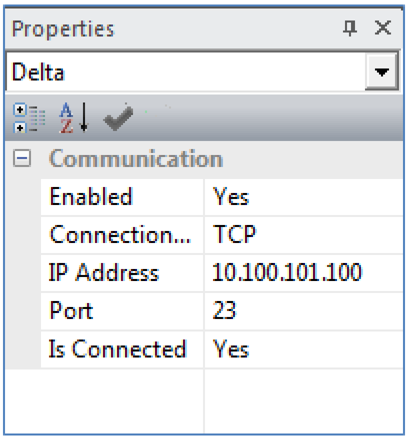


7thSense Delta Connection

7thSense Delta Media Server Connection (talk to Delta)

There is a dedicated Delta connection that can be configured in the drop-down properties list. This is used by Delta Nodes such as MediaMove and Sequence, to talk to Delta. In the future this will support connections to more than one Delta.

In the Properties pane, from the top drop-down, select ‘Delta’. The Connection Type (TCP/UDP), IP Address, and Port can be configured in the ‘Delta’ Selection in the properties drop down.



Properties

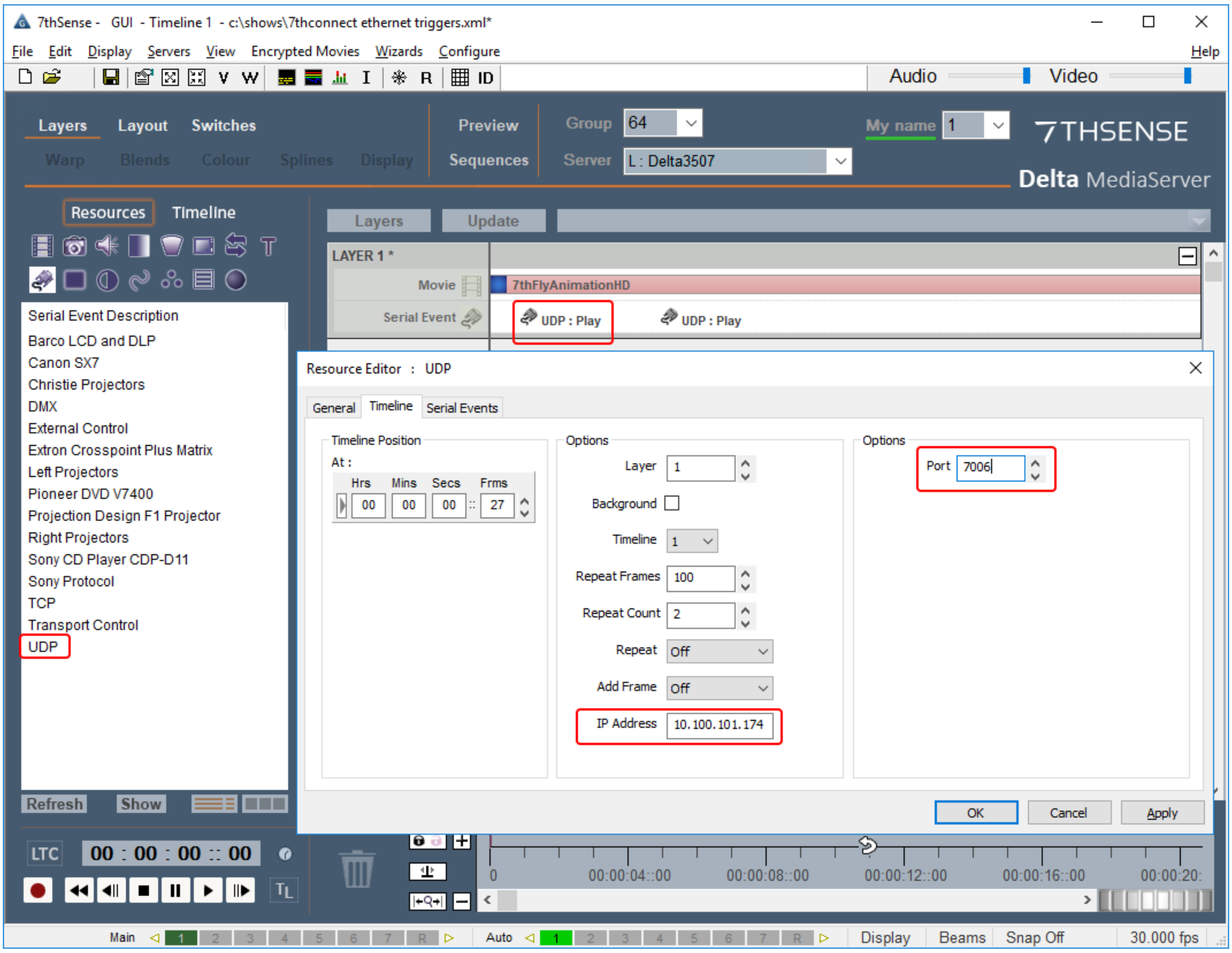
Delta

Communication

Enabled	Yes
Connection...	TCP
IP Address	10.100.101.100
Port	23
Is Connected	Yes

7thSense Delta Media Server Connection (listen to Delta)

7thConnect can listen to Serial Events on the DeltaGUI timeline, and receive ASCII strings via UDP or TCP. For example, add a UDP Serial Event to the timeline and double click for properties:



7thSense - GUI - Timeline 1 - c:\shows\7thconnect ethernet triggers.xml*

Group: 64

Server: L : Delta3507

My name: 1

Resources:

- Serial Event Description
- Barco LCD and DLP
- Canon SX7
- Christie Projectors
- DMX
- External Control
- Extron Crosspoint Plus Matrix
- Left Projectors
- Pioneer DVD V7400
- Projection Design F1 Projector
- Right Projectors
- Sony CD Player CDP-D11
- Sony Protocol
- TCP
- Transport Control
- UDP

LAYER 1 *

Movie: 7thFlyAnimationHD

Serial Event: UDP : Play   UDP : Play

Resource Editor : UDP

General | Timeline | Serial Events

Timeline Position

At : Hrs 00 Mins 00 Secs 00 Frms 27

Options

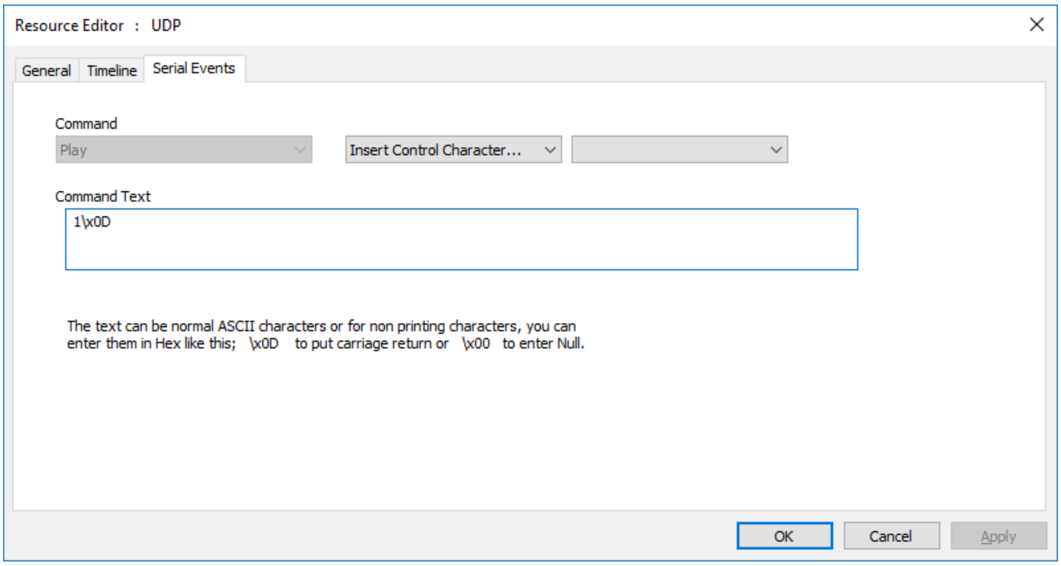
- Layer: 1
- Background: □
- Timeline: 1
- Repeat Frames: 100
- Repeat Count: 2
- Repeat: Off
- Add Frame: Off
- IP Address: 10.100.101.174

Options

- Port: 7006

In the Timeline tab, add the server IP address and assign an available port (see Ports used by Delta).

In the Serial Events tab, add the string you want to send to the Phidget interface via an Ethernet Node:



Command: Play

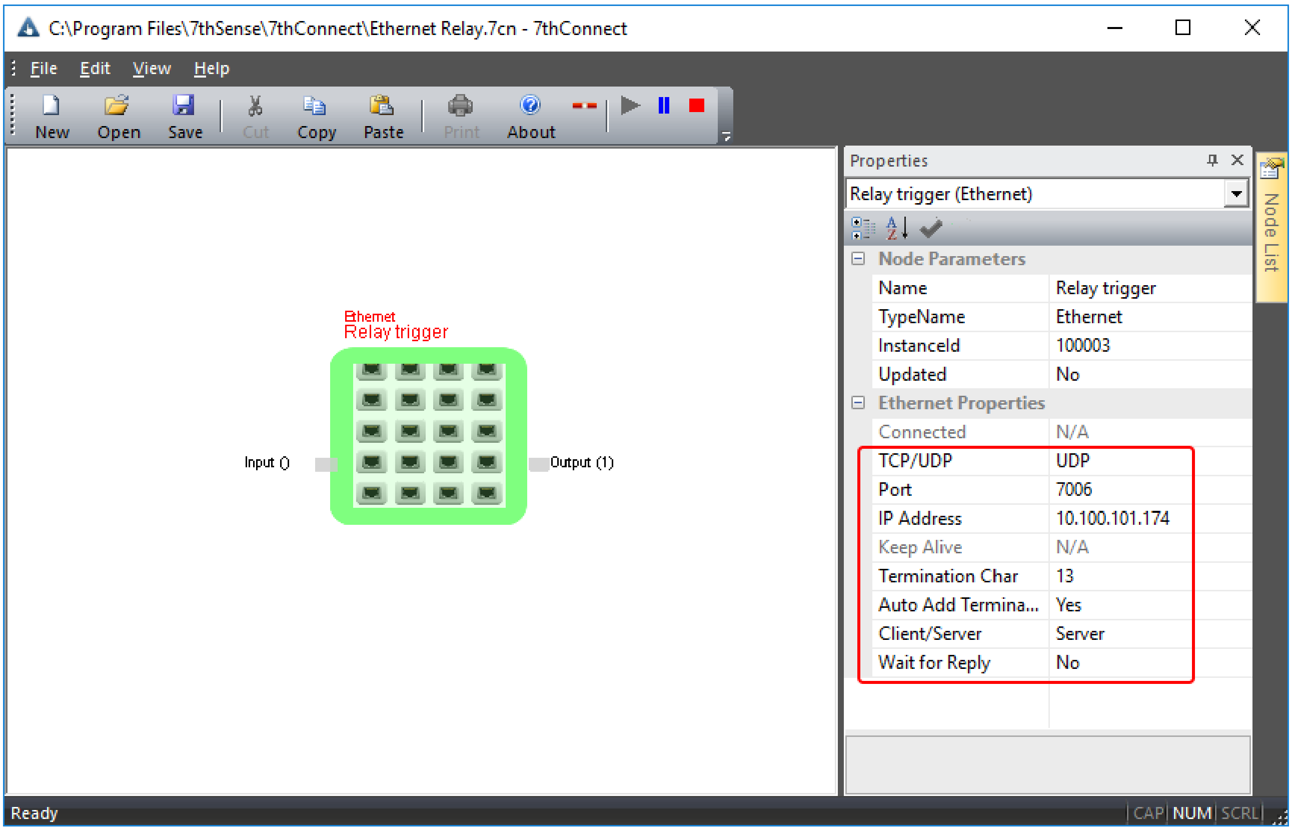
Insert Control Character...

Command Text: 1\x0D

The text can be normal ASCII characters or for non printing characters, you can enter them in Hex like this; \x0D to put carriage return or \x00 to enter Null.

In this example we are simply sending a value of 1 (with carriage return). This might switch something on, so a second UDP Serial Event with the command text of 0\x0D could be used to switch it off again.

In 7thConnect, now set an Ethernet Node to listen to UDP commands on this IP and port, **as a server**:



Relay trigger (Ethernet)

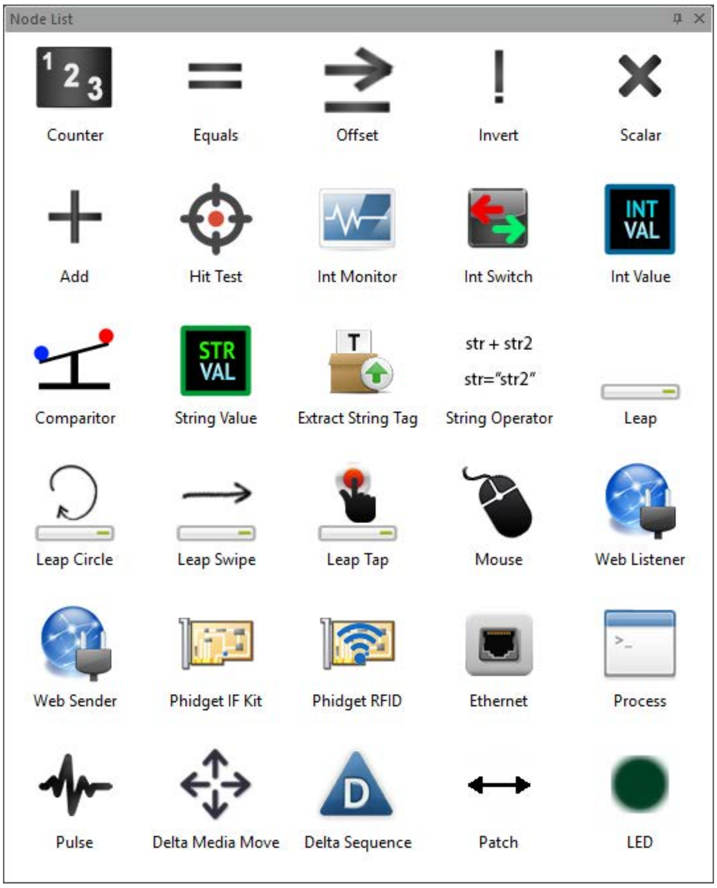
Node Parameters	
Name	Relay trigger
TypeName	Ethernet
InstanceId	100003
Updated	No
Ethernet Properties	
Connected	N/A
TCP/UDP	UDP
Port	7006
IP Address	10.100.101.174
Keep Alive	N/A
Termination Char	13
Auto Add Termina...	Yes
Client/Server	Server
Wait for Reply	No

We have also named this Node 'Relay trigger' because that is what we will use it for, and this name appears in red on the Node icon for clarity. As you play the timeline over the UDP Serial Event, the string value will appear as the Node output.

➢ To create a **TCP connection**, to use TCP Serial Events, see the Delta User guide, TCP Connections.

Nodes

Nodes are the operators that can be connected into a system that combines physical controls with evaluation of their outputs to trigger other events, including Delta sequences. 7thConnect has a built-in set of nodes that can be used to create a system.



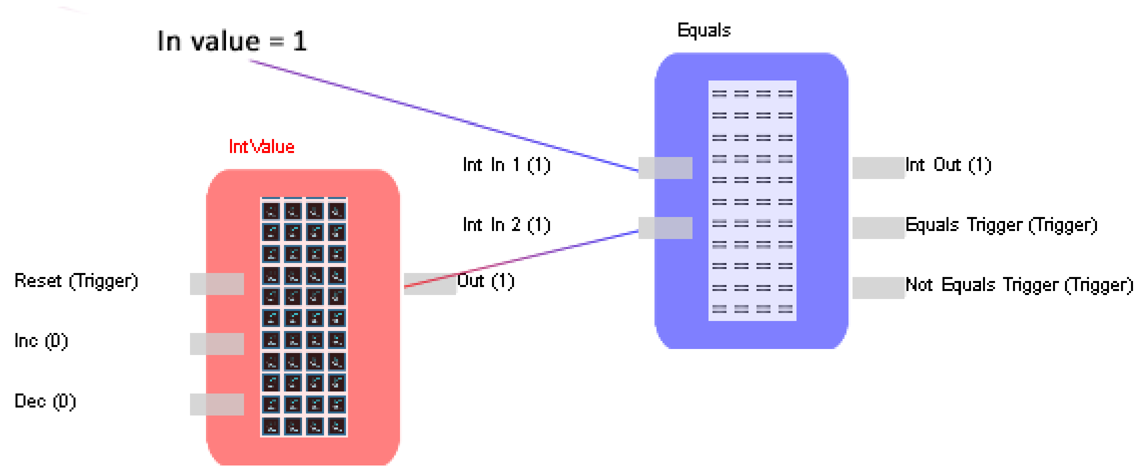
Input and Output types

There are three types of input and output:

- a binary **state** (0 or 1)

- a **value** (string or numerical)
- a **trigger** ('do this')

A value of 1 is not a binary status and will not act as a trigger. A trigger output can be generated by comparing values. Here, an integer value of 1 is being compared with a binary input, using an equals node:



The **LED** node is a visual indicator of any non-zero state or value (=on), or any trigger (switches between on and off).

The **Invert** node simply inverts a binary state.

Node	Trigger in	Trigger out(s)	Binary in	Binary out	Value(s) in	Value out
Add	✗	✗	✗	✗	✓	✓
Comparator	✗	✗	✗	✓	✓	✓
Counter	✗	✗	✗	✗	✗	✓
Delta Media Movie	✗	✗	✗	✗	✓	✗
Delta sequence	✓	✗	✗	✗	✗	✗
Equals	✗	✓	✗	✓	✓	✗
Ethernet	✗	✗	✗	✗	✓	✓
Extract String Tag	✗	✗	✗	✗	✓	✓
Hit test	✓	✓	✗	✗	✓	✗
Int Monitor	✗	✗	✗	✗	✓	✗
Int Switch	✗	✗	✓	✗	✓	✓
Int Value	✓	✗	✗	✗	✓	✓
Leap nodes	✗	✓	✗	✗	✗	✗
Mouse	✗	✓	✗	✓	✗	✓

Node	Trigger in	Trigger out(s)	Binary in	Binary out	Value(s) in	Value out
Offset	✗	✗	✗	✗	✓	✓
Process	✗	✗	✗	✗	✓	✗
Pulse	✓	✓	✗	✗	✗	✗
Scaler	✗	✗	✗	✗	✓	✓
String Operator	✗	✗	✗	✗	✓	✓
String Value	✓	✗	✓	✗	✗	✓
WebListener	✗	✗	✗	✗	✗	✓
WebSender	✗	✗	✗	✗	✓	✓

Node Descriptions

Add

Offset the incoming value by adding a fixed amount to the incoming value. Adds a fixed amount to the incoming value. See also **Offset**.

User Properties

Add (amount to add)

Pins

Inputs

In

Outputs

Out

Comparator

Compare multiple inputs.

User Properties

Num Inputs
(how many inputs to compare)

Mode of Operation for Ints:

Maximum

which input number (integer) has the highest value

Equals

all inputs must be the same for a True output

Less Than

compares 2 inputs, In1 < In2 = True

Greater Than

compares 2 inputs, In1 > In2 = True

Pins

Inputs

Number of inputs defined by Num Inputs property

Outputs

Out: Result True/False

Counter

Counts from ‘Start Value’ to ‘Stop Value’ over a set time.

User Properties

Start Value: count from (integer)

Stop Value: count to (integer)

Loop: repeat the count constantly

Ping Pong: count up and then down between the two values

Pins

Outputs

Out (Value of counter)

Delta MediaMove

Use for 2D Media Move with 7thSense Delta Media Server.

Uses the inbuilt Delta connection (that can be configured in the Properties drop down list – see the 'Built-In Delta Connection section')

User Properties

ControlName
Media external control name

Timeline
Timeline of Media

Pins

Inputs
2D_X_In: X Position to move media to
2D_Y_In: Y Position to move media to
2D_W_In: Width to set media to
2D_H_In: Height to set media to

Outputs
None

Delta Sequence

Use for triggering a sequence in 7thSense Delta Media Server.

Uses the inbuilt Delta connection (that can be configured in the Properties drop-down list – see 7thSense Delta Connection [11]).

User Properties

SequenceName
Name of sequence to trigger

Pins

Inputs
In: Input trigger

Equals

Binary/Logic: compares two input integer conditions to create an output integer value (true/false) and true and false triggers. See also **Int Values**.

User Properties

None

Pins

Inputs

In 1
In 2

Output

Out: Value of 1 if the inputs match, 0 if they don't match
Equals (Trigger): Triggered when the inputs become equal
Not Equals (Trigger): Triggered when the inputs become not equal

Ethernet

Ethernet Receiver and Transmitter Node.

Can be configured as Server or Client for TCP or UDP data.

User Properties

TCP/UDP
Select between TCP or UDP Protocol

Port
Port for Connection

IP Address
IP Address for Client Connection, also used to bind server to an IP Address.

Keep Alive
Maintains the connection open between data sends. Without this set the socket will be opened, data sent, and socket closed.

Termination
Character to append when auto add termination char is set to Yes
e.g. 13 appends a carriage return, as the ASCII character for carriage return is 0x0d or 13

AutoAddTerminationChar
Turn on Termination Character to be appended to all outgoing data.

Client/Server

Act as Client or Server
Client actively creates connections.
Server listens for connections.

Wait for Response

Wait for a response after sending data.

Pins

Inputs

Input: Data to be transmitted

Outputs

Output: Data received

Extract String Tag

Extracts data from a String

User Properties

Tag: Tag to search for. e.g.:

The Input string could be: testdata='test1'
Tag should be set to testdata
Output value would be 'test1'

Pins

Inputs

Input: The Input String

Outputs

Value: The output data

Hit Test

Takes in 2D coordinates X,Y and determines whether they sit within a region. A Trigger is fired out if the input X,Y sits within the region.

User Properties

X Minimum

X Maximum

Y Minimum

Y Maximum

Pins

Inputs

In (Trigger)
X
Y

Outputs

Out (Trigger)

Int Monitor

Monitors an input.

User Properties

None

Pins

Inputs

In

Int Switch

Binary/Logic: pass data through, based on whether the switch is set to On or Off. Output value = Input value when switch value = 1

User Properties

None

Pins

Inputs

OnOff (1=On, 0=Off)
In

Outputs

Out

Int Value

Increment or decrement, and hold or reset, an integer value. Can be used to count trigger events.

User Properties

Value

Current Value

Reset Value

Value to reset to on Reset Trigger

Pins

Inputs

Reset (Trigger)
Inc: Increment by 1
Dec: Decrement by 1

Outputs

Out: Current Value

Invert

Binary/Logic: inverts the state of the input, not the value.

e.g. a Pulse operating an LED directly is **on** when an LED from the same Pulse through an Invert node will be off, and a mouse **press** becomes a mouse **release**.

Leap: all inputs

Connect supports inputs from a Leap Motion USB Hand detection sensor.

Requirements: For Pinch and Grab strength – SDK 2.0 of Leap is required.

Leap Node

User Properties

None

Pins

Inputs

None

Outputs

Num Hands Detected	Num Hands Detected
Hand Detected (Trigger)	Triggered when Hand Detected
Hand Removed (Trigger)	Triggered when Hand removed
Hand1 Num Fingers	Num Fingers Detected
Hand1 Sphere Centre X	Current position of hand
Hand1 Sphere Centre Y	Current position of hand
Hand1 Sphere Centre Z	Current position of hand
Hand1 Pinch Strength	0.0 - 1.0 range. 1.0 – good pinch detected
Hand1 Grab Strength	0.0 - 1.0 range. 1.0 – good grab detected
Hand2 Num Fingers	Num Fingers Detected
Hand2 Sphere Centre X	Current position of hand
Hand2 Sphere Centre Y	Current position of hand
Hand2 Sphere Centre Z	Current position of hand
Hand2 Pinch Strength	0.0 - 1.0 range. 1.0 – good pinch detected
Hand2 Grab Strength	0.0 - 1.0 range. 1.0 – good grab detected

Leap Circle

User Properties

None

Pins

Inputs

None

Outputs

Clockwise Circle Detected (Trigger): Triggered when clockwise circle motion detected
Anti-Clockwise Circle Detected (Trigger): Triggered when clockwise circle motion detected

Leap Swipe

A Swipe represents a swiping motion of a finger or tool.

User Properties

None

Pins

Inputs

None

Outputs

Swipe Detected (Trigger)

Direction: Direction of swipe. The unit direction vector parallel to the swipe motion
X Pos: Position of hand at start for swipe
Y Pos: Position of hand at start for swipe
Z Pos: Position of hand at start for swipe

Leap Tap

A Key Tap gesture

A key tap gesture is recognized when the tip of a finger rotates down toward the palm and then springs back to approximately the original postion, as if tapping. The tapping finger must pause briefly before beginning the tap.

User Properties

None

Pins

Inputs

None

Outputs

Screen Tap: Detected (Trigger)

X Pos: Position of Tap
Y Pos: Position of Tap
Z Pos: Position of Tap

LED

A simple Green = On indicator. Use this also to test results in sections of your system, as a visual indicator.

Output
None

Mouse

Monitors the mouse or pointer on the computer running the Connect application. Positional values can be passed to strings, mouse buttons can be used to trigger events, or run Delta sequences.

User Properties

None

Pins

Inputs
None

Outputs
MouseX: Mouse X Position
MouseY: Mouse X Position
LButton: Clicked (Trigger)
MButton: Clicked (Trigger)
RButton: Clicked (Trigger)
LButton: Released (Trigger)
MButton: Released (Trigger)
RButton: Released (Trigger)
LButton: Down (1 = Down)
MButton: Clicked (1 = Down)
RButton: Clicked (1 = Down)

Offset

Offset the incoming value by adding a fixed amount to the incoming value. See also **Add**.

User Properties

Offset (amount to offset)

Pins

Inputs

In

Outputs

Out

Patch

A simple connection Node with one input and one output.

Phidget 8/8/8 IF Kit

The following Phidgets IF Kits are supported:

- http://www.phidgets.com/products.php?category=0&product_id=1018_2
- https://www.phidgets.com/?tier=3&catid=1&pcid=0&prodid=969
- Phidget interface 1073_0 is supported but is a discontinued item, replaced by PhidgetSBC4

The Phidget IO Nodes Input 1 pin is a 5V digital input trigger. For more information on the Phidget IO devices see www.phidgets.com.

In the Kit Serial Number property, use the drop-down to select your USB-attached kit.

User Properties

None

Pins

Inputs

Input 1	Digital Input Value (1 or 0)
Input 2	
Input 3	
Input 4	
Input 5	
Input 6	

Input 7
Input 8

Sensor1: Value Input from connected analog sensors
Sensor2
Sensor3
Sensor4
Sensor5
Sensor6
Sensor7
Sensor8

Outputs

Output1: Resultant output from connection Digital Outputs (Switches, etc).
Output2
Output3
Output4
Output5
Output6
Output7
Output8

Phidget RFID

The following Phidget RFID board is supported

- http://www.phidgets.com/products.php?category=14&product_id=1024_0

User Properties

RFID Serial Number
Select from drop-down list of detected devices.

Pins

Inputs

Tag Present: Indicates if a tag is present
Tag Name: The name of the currently present Tag

Outputs

Output 1: Indicates if Output 2 Is active
Output 2: Indicates if Output 2 Is active
Antenna On: 1 if Antenna on RFID controller board is On
LED On: 1 if LED on RFID controller board is On

Process

Launch or Kill a Windows process running on the same computer as Connect.

User Properties

None

Pins

Inputs

Launch: String of path to executable to launch.
When a pulse is received with this string, the process will be launched.
Kill: String of path to executable to Kill.
When a pulse is received with this string, the process will be killed.

Outputs

None

Pulse

Send a Pulse / Trigger out. It can be configured for one off, or repeated operation.

User Properties

Repeat Time (ms)
Number of milliseconds between pulses (minimum 16ms)

Enabled
Enabled or Not

Pins

Inputs

Start (Trigger): Trigger to start pulsing
Stop (Trigger): Trigger to stop pulsing

Outputs

Pulse: Output Pulse / Trigger

Scaler

Scale the incoming value. Multiplies the incoming value by a fixed amount.

User Properties

Scaler (amount to multiply by)

Pins

Inputs
In

Outputs
Out

String Operator

Performs String Operations.

User Properties

Mode

Concatenate: joins the two inputs together, e.g. to send a string command with a pointer or sensor value:

In1 = globalvarset $Xpos=
In2 = [output value from sensor]
Out = globalvarset $Xpos=[output value from a sensor]

StringTag: creates string tag from the two inputs, e.g.:

In1 = Test
In2 = Data
Out = test='data'

StringTagWithIdAndItem: creates string tag from the two inputs, with Id and Item. e.g.:

In1 = Test
In2 = Data
Out = id='test'&item='data'

Pins

Inputs
In1: Input String

Outputs
Out: The output data

String Value

A String Value

The String Value node has a string property set to ‘Value’ property. It can be triggered to resend its data.

User Properties

Value: Current String

Pins

Inputs

Trigger: Trigger to (re)send the string data

Outputs

Out: Result

WebListener

Listens to Web requests.

BaseURl of the server is shown in the properties.

User Properties

Relative URL – Defined the additional navigation address for this listener

Pins

Inputs

None

Outputs

Get Rx: Triggered when a Get is received
Get Tx: Triggered when a Post is received
Get Data: The data received with the Get request
Post Data: The data received with the Post request
e.g. Base address http://*:7791/api/7thConnect
Relative URL = test

If the IP Address of the computer running Connect is 10.100.101.1, then navigating to the following URL in a browser would trigger a Get:

http://10.100.101.1:7791/api/7thConnect/test?data=test

The Get Data pin should have a value of: ?data=test

WebSender

Sends Web requests and delivers the response.

User Properties

None

Pins

Inputs

Get: The web address to send a get command to
Post: The web address to send a Post command to

Outputs

Output: The value received back
e.g. Attach a string value node with a value of http://www.google.com to the Get Input pin. The output pin will show the response string received from Google.

Worked Examples: sensors control Delta

Phidget sensors can be used to make something happen in Delta, often by acting as triggers to Delta sequences.

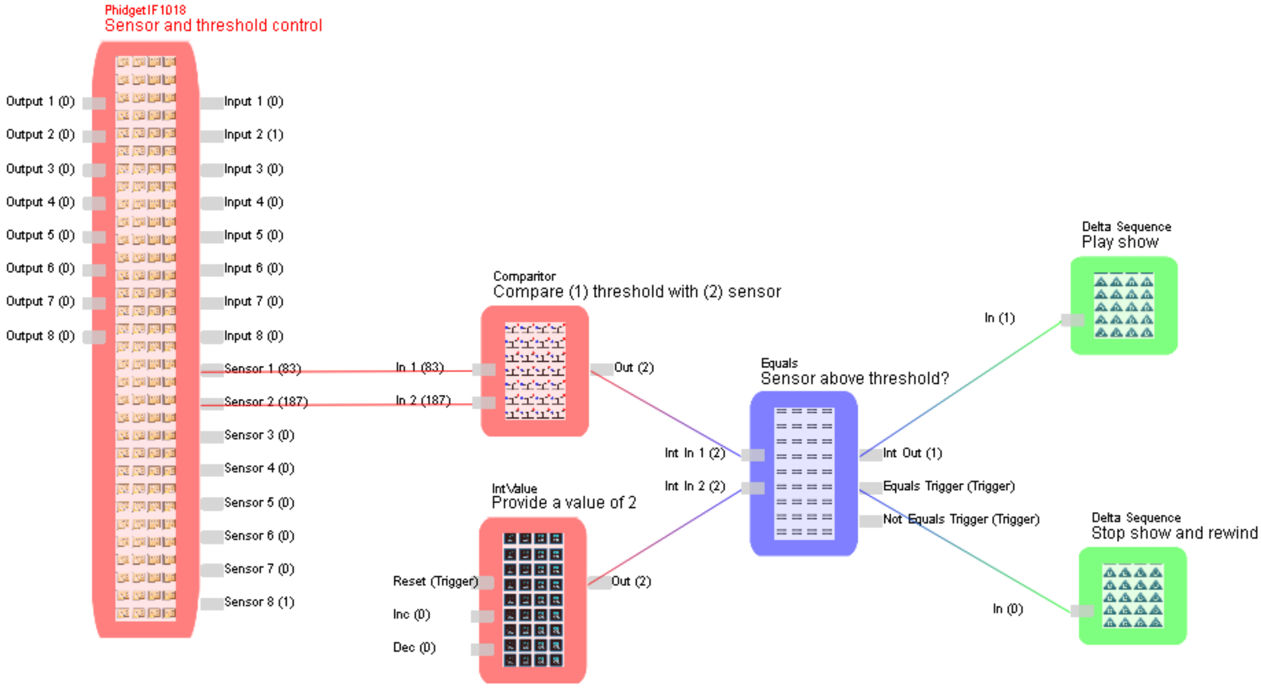
Sensor with threshold setting plays and stops a show

Let's start with a Phidget Interface kit, to which we connect a variable manual control (e.g. a slider) and a sensor (motion, light or sound). The example interface is the 8/8/8 with 8 sensor inputs, and 8 digital in and out.

The manual control (Sensor 1 input to the interface) will allow us to set a threshold value to set against the light sensor (Sensor 2 input) value, and create a trigger.

This example can be seen below. Sensor 1 on the interface kit is the manual control and Sensor 2 is a light or sound sensor. The Delta Sequences (green, on the right) have already been written from DeltaGUI to gracefully start and stop a show timeline. In this way a variable sensor value is converted into a simple on/off trigger to run a sequence that begins a series of steps in Delta.

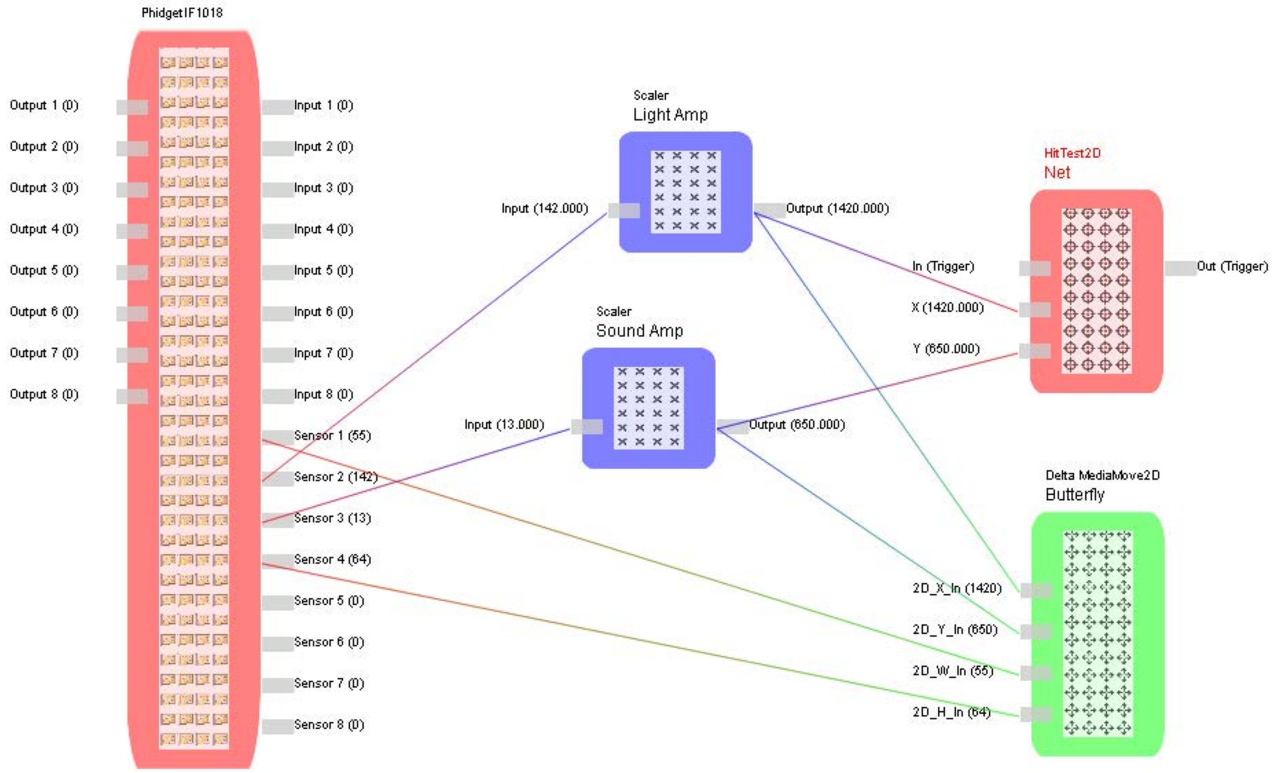
Sensor 1 and Sensor 2 outputs are first compared for which has the greater value. The result is either 1 or 2. By equating this with an integer value of 2, we can trigger a Delta Sequence to play the show when the value of Sensor 2 falls below the threshold (Sensor 1). If Sensor 2 rises above the threshold value (as below) an Equals trigger is sent to another Delta Sequence to stop the show and rewind it.



Note how by applying **node names** you can better explain (or remind yourself) of the logic.

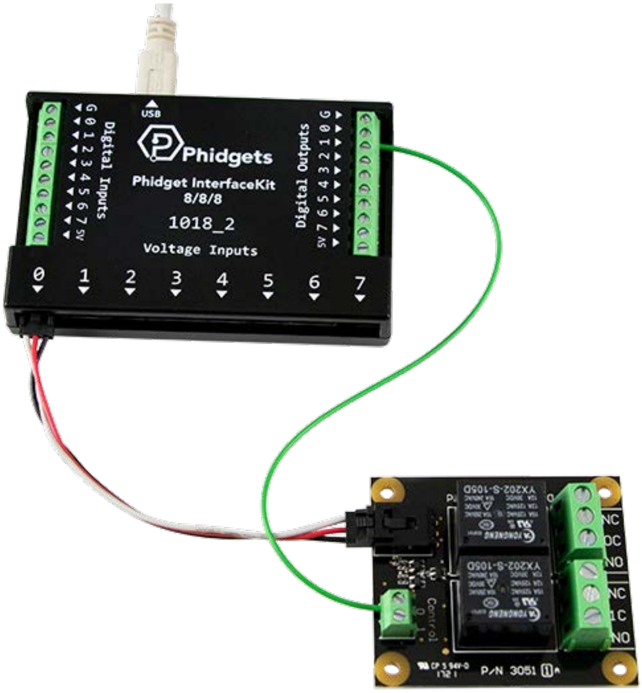
Use sensors to move media on-screen

In this example, from a Phidget kit on the left, a light sensor (2) and a sound sensor (3) are used to move a piece of media around, using the green Delta MediaMove node: it might be catching a butterfly in a net on a display screen. The size of the media (the butterfly, for example) can be changed by two analogue controls (or it could have been just one, to be proportional), but the position co-ordinates need to be amplified with Scaler nodes for sufficient movement on screen. The target area (a HitTest2D node, here called Net) is fed the same positional outputs. Getting the butterfly in the net can then trigger some other event from the HitTest, whether a sound, or a sequence in Delta, or another process.

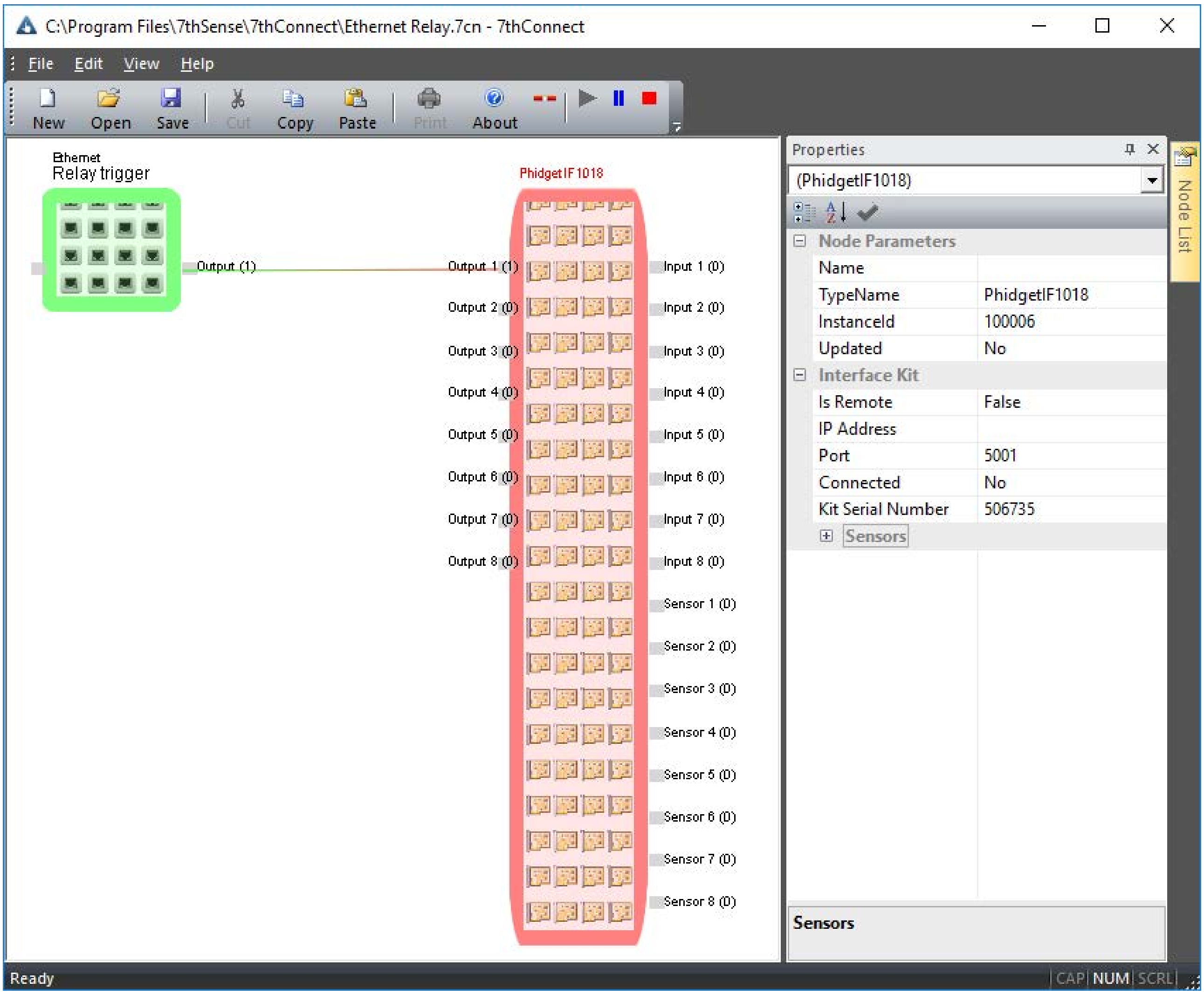


Worked Examples: Delta controls Phidget

Sometime you want Delta to control an active Phidget, such as a remote switch relay to control larger loads, such as to operate a relay or activate a stepping motor. A variety of Phidgets are available; in this example a Phidget 3051_1 Dual Relay Board is connected to an 8/8/8 interface:



This is represented in 7thConnect as this, with the received UDP value on the Ethernet Node output connected to a digital output on the interface:



(PhidgetIF1018)

Node Parameters	
Name	
TypeName	PhidgetIF1018
InstanceId	100006
Updated	No
Interface Kit	
Is Remote	False
IP Address	
Port	5001
Connected	No
Kit Serial Number	506735
Sensors	

Sensors

While the value persists, the relay will be closed. So to operate the relay from a Delta timeline, you can use multiple UDP Serial Events with 1 and 0 values, as illustrated in 7thSense Delta Connection (11) above.

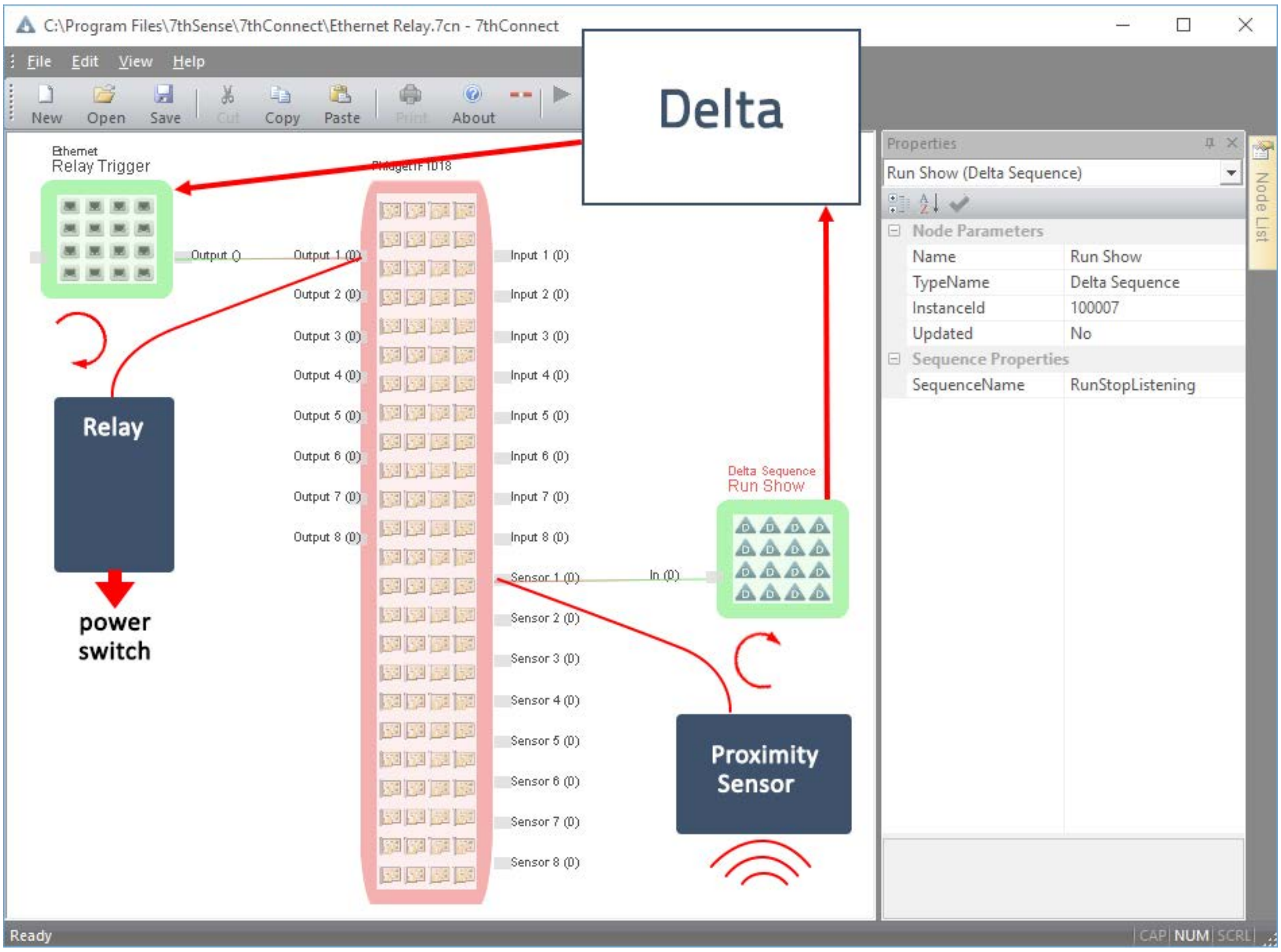
For multiple independently triggered relays, use different UDP ports.

How this might be used in 2-way interaction

- A proximity sensor is connected to the Phidget interface, to detect visitor presence. In 7thConnect, this triggers a ‘Play’ sequence in Delta to start a show.
- To prevent the proximity sensor (a moving visitor or a second visitor) triggering the show to play again, a ‘StopListening’ variable is set in the ‘Play’ sequence in Delta to ignore further messages from 7thConnect until this variable is reset.
- The show has timeline Serial Event(s) where the relay needs to be triggered or released: this UDP Serial Event sends the required ‘1’ or ‘0’ value to the Ethernet Node in 7thConnect.

- A ‘Reset’ sequence runs at the end of the show to rewind the Timeline and also to reset the ‘StopListening’ variable, allowing Delta again to listen to the proximity sensor trigger via 7thConnect and restart the show.

Here is the physical relationship between 7thConnect, Delta, the sensor and the relay:



Run Show (Delta Sequence)

Node Parameters

Name	Run Show
TypeName	Delta Sequence
InstanceId	100007
Updated	No

Sequence Properties

SequenceName	RunStopListening

Document Information

Date	Document edition	Software version	Revision Details	Author/Editor
November 2014	1	1.0	New release	Richard Brown
November 2017	2	1.1.0.3	Reformatted and expanded	Andie Davidson
July 2018	3	1.1.0.3	Improved examples	Andie Davidson
June 2019	4	1.1.0.3	Delta to Phidget examples added	Andie Davidson

E: info@7thsense.one
W: 7thsense.one

7thSense Design Ltd

2 The Courtyard, Shoreham Road
Upper Beeding
Steyning
West Sussex
BN44 3TN
UK

T: +44 (0) 1903 812299

7thSense LLC, Michigan

332 E Lincoln Ave
Suite 100
Royal Oak, MI 48067
USA

T: +1 248 599 2717

7thSense LLC, Orlando

4207 Vineland Rd
Suite M1
Orlando, FL 32811
USA

T: +1 407 505 5200