

Introduction

Many systems allow integration over an Ethernet protocol (UDP or TCP based). These Ethernet Protocols pass data from one device to the other. All Pharos Controllers are capable of Ethernet communications, but in a multi-controller installation, only the Time Server (main controller) is able to communicate with external systems over Ethernet.

Communications

Ethernet communications with Pharos Systems don't have a standard communications syntax, but typically the other device involved will have a standard syntax, so the Pharos Controller should be configured to match that syntax. The way Ethernet communication is configured allows you to enter any ASCII, Hex or Decimal strings for the controller to listen for, or to output to other devices.

All Ethernet communications with a Pharos System are handled within the Trigger functionality, but the communication is configured within the Network Mode. Select the Time server, then go to Interfaces and setup an Ethernet Bus with the required IP protocol and Port.



Ethernet
Configure buses for Ethernet triggering. These settings are associated with the Time Server.

Bus 1	Type	UDP
Bus 2	Port	800
Bus 3		
Bus 4		
Bus 5	IP address	

Ethernet Input

An Ethernet Input trigger is included to allow you to receive data into the Pharos controller. To configure the Trigger, select the required Ethernet Bus and add the string that you want to listen for (e.g. start_timeline_1)

Ethernet Output

An Output Ethernet Action is available which will output an ethernet message whenever the associated trigger is fired. To configure the Action, enter the IP Address of the device that the message should be sent to and the receiving port, then enter the string that you want to output (e.g. timeline_1_started)

String Matching

Using a single trigger for each message that you want to receive can make it easier to see what is happening within the project, but can lead to some unexpected behaviour due to the string matching process.

Whenever the Pharos System receives a character (letter, number, punctuation or control character) in the Ethernet Bus, it is stored in a buffer. If the characters in the buffer match the string in any of the Ethernet Input triggers in your project, the trigger will be fired and the buffer cleared.

Example:

If you have a series of triggers to start different timelines, you could have a trigger with the string start_timeline_1 and another trigger with the string start_timeline_11.

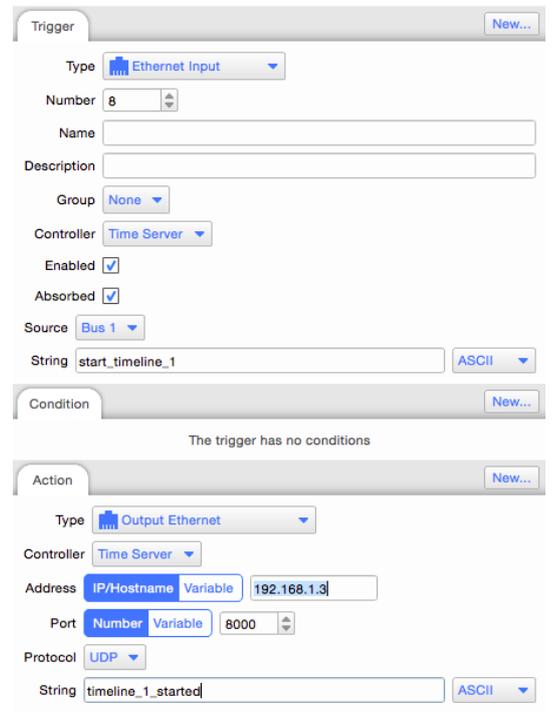
If this happens, then whenever the controller receives the string start_timeline_11 it is read in as follows:

s t a r t _ t i m e l i n e _ 1 1

The red characters are the same as the trigger configured to receive start_timeline_1 and so that trigger will be fired and the buffer cleared, then the second 1 will be received.

There are two ways around this:

1. Add leading zeroes to the strings:



Trigger New...

Type: Ethernet Input

Number: 8

Name:

Description:

Group: None

Controller: Time Server

Enabled:

Absorbed:

Source: Bus 1

String: start_timeline_1 ASCII

Condition New...

The trigger has no conditions

Action New...

Type: Output Ethernet

Controller: Time Server

Address: IP/Hostname Variable 192.168.1.3

Port: Number Variable 8000

Protocol: UDP

String: timeline_1_started ASCII

```
start_timeline_01  
start_timeline_11
```

2. Add a terminating character after the number, typically a newline or carriage return character

```
start_timeline_1\r (\r is the ASCII representation of a carriage return character)  
start_timeline_11\r
```

This will force the trigger to wait for the character after the number before firing, ensuring that the whole number is received.

Variable Data

Ethernet Input

If you have a large number of timelines that you want to start using an Ethernet message, or you want to be able to pass in colour information for example, then having individual triggers for every possible value will become unwieldy quickly. To get around this, you can capture bits of data from the string e.g. `start_timeline_<2d>`

The angle brackets (<>) define the capture region and the data inside the angle brackets determines what sort of data is expected.

In the example above <2d> will capture any two digit number (00-99), meaning that the trigger will be fired whenever a string containing "start_timeline_" followed by a two digit number.

The two digit number is then captured into a variable, which can then be used in the action associated with the trigger (e.g. Start Timeline).

Ethernet Output

An Ethernet output action can also use variables to output a variable piece of data.

If we wanted to reply to the external device to say that we have started the timeline that was requested, we can form an Ethernet output string including a capture in the same format as the input and the number will be passed from the Trigger to the Action and the number will be sent back to the external device.



The screenshot shows a configuration window with three sections: Trigger, Condition, and Action. The Trigger section is set to 'Ethernet Input' with a number of 8, a name field, a description field, a group of 'None', a controller of 'Time Server', and is enabled and absorbed. The source is 'Bus 1' and the string is 'start_timeline_<2d>' with an ASCII encoding. The Condition section is empty, stating 'The trigger has no conditions'. The Action section contains two actions: 'Start Timeline' with controller 'All Controllers' and a variable '1' for the timeline, and 'Output Ethernet' with controller 'Time Server', address 'IP/Hostname Variable 192.168.1.2', port 'Number Variable 8000', protocol 'UDP', and string 'timeline_<2d>_started' with an ASCII encoding.

Further information

Further information can be found in the Triggers section of the software help or by contacting Pharos Support.