Isadora Actors for using TCPIP connection of the LanBox

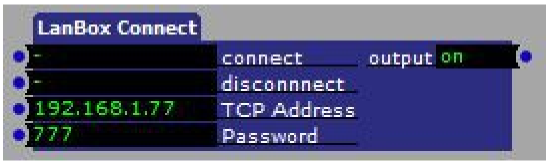
Here is a collection of Actors which use the LanBox via TCPIP. My goal was to create a lightdesk in Isadora, which uses the LanBox as DMX interface. Along on this work there reseulted a couple of actors which I needed in the more advanced actors.

The highlights in this collection are the actors **LanBox Scene Fader** and **LanBox Store Cuestep**. The LanBox Scene Fader actor uses a Cuestep of the LanBox and outputs channel values in relation to a fader value between 0 and 100%. The main intention was to use a MIDI faderboard like the Korg nano kontrol on the LanBox, but with this actor you can also create a Sound to Light function. The LanBox Store Cuestep actor provides you with a tool to store the actual output as cuestep. You can also use this actor to copy DMX input values (from an external light console) into the LanBox as a cuestep.

Thanks to jhoepffner (http://forum.troikatronix.com/cgi-bin/forum/gforum.cgi?username=jhoepffner;t=search_engine), I got the ideas how to connect via TCPIP.

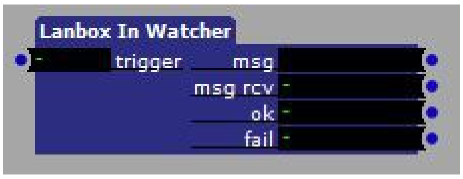
Part 1: Basics

LanBox Connect



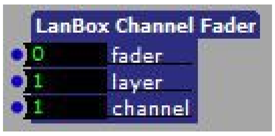
This is an adaption of jhoepffner's actor, I added the password as user input. After a connection is made, the LanBox command Common16BitMode is executed. For using this actor in a program make sure that the scene which contains the LanBox Connect actor is active. Otherwise you will loose the connection after a while. In my programs I include the LanBox Connect actor in the first scene, and in every subsequent scene I include an Activate Scene actor which points to this first scene.

LanBox In Watcher



This actor is an adaption of the TCP In watcher. It listens to a TCP stream for only one input and only for one second. It is used in conjunction with a TCP send data actor, triggered at the same time. The LanBox In Watcher actor triggers „ok“, if it receives '>' or '*' as first character of the returning string. It triggers „fail“ if it receives a '?' or after 1 second without an answer.
This is a very useful actor if you want to send a sequence of TCP commands. You can decide in your program depending on the LanBox answer. And you have a trigger to be sure that the LanBox has done a request before you send the next command.

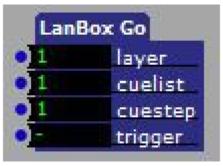
LanBox Channel Fader


LanBox Channel Fader
0 fader
1 layer
1 channel

With this actor you get access to the channel value in a layer. Depending on the layer status, the channel value will be included in the DMX output. By LanBox default, Layer A is on top of all layers and in Copy mode. So for a first try, fade channel values in layer A (= layer 1).

Input parameters: fader, layer, channel.

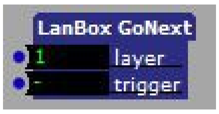
LanBox Go


LanBox Go
1 layer
1 cuelist
1 cuestep
- trigger

This is a basic LanBox command. In this case it makes sense to put that into the Global Toolbox. If you have the programming of your cuesteps done, this actor activates the cuesteps in the desired layer.

Input parameters: layer, cuelist and cuestep, trigger.

LanBox GoNext


LanBox GoNext
1 layer
- trigger

This actor is even simpler, you trigger the next cuestep in the cuelist. Also makes sense to have this actor in the Global Toolbox.

Input parameters: layer, trigger.

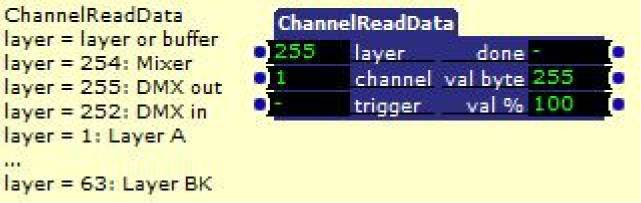
Basic LanBox Commands

The following list of LanBox commands can be found in the examples. These are mainly TCP Send Data actors with correct LanBox notation and the corresponding parameters as input values. The 'Get'- and 'Read'-commands are used in conjunction with the LanBox In Watcher, triggered at the same time. Please have a look at the examples. I did not put these actors to the Global Toolbox, since this would result in a lack of clarity in the toolbox. If you compare the following list with the LanBox commands, you will find a great deal ready to use with Isadora! I grouped the collection according to the List of Commands of the LanBox Reference Chart Version 3.01.

General Commands
Common16BitMode: Included in LanBoxConnect
CommonReboot
CommonSaveAll: Not documented in the Reference Chart 3.01. I found this command in the Command Chart version 1.1a2. I suppose that is equal to the Save LanBox Data command in LC Edit. After executing this command, the cuelists are stored and you can safely reboot or shut down the LanBox.

Channel Control
ChannelSetData

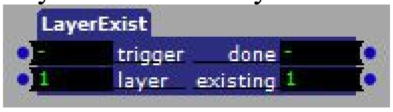
ChannelReadData



ChannelReadData
layer = layer or buffer
layer = 254: Mixer
layer = 255: DMX out
layer = 252: DMX in
layer = 1: Layer A
...
layer = 63: Layer BK

ChannelSetOutputEnable
ChannelSetActive

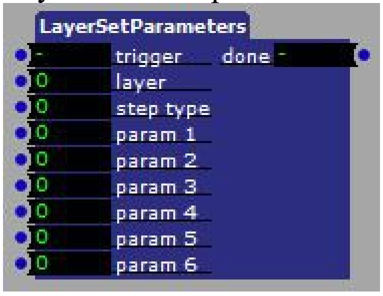
Layer General Control
LayerGetStatus: LayerExist



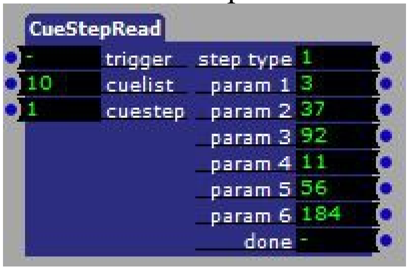
LayerSetMixMode
LayerConfigure: Create Layer and Delete Layer

Layer Playback Control
LayerGo: LanBox Go and LanBox GoNext
LayerClear

Layer Real Time Editing
LayerSetEditRunMode
LayerUsesCuelist
CuelistCreate
LayerInsertStep and LayerInsertStep Append
LayerReplaceStep
LayerSetCueStepType
LayerSetCueStepParameters: all Cuestep parameters are set in a macro – LayerSetParameters



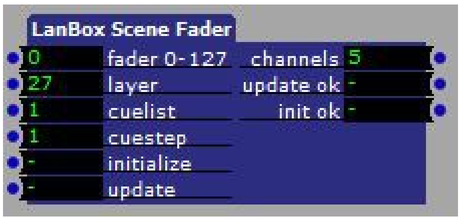
Cue List Control
CueListRead: all parameters of one cuestep are read in one macro: CueStepRead



CuelistRemoveStep.

Part 2: Applications

LanBox Scene Fader



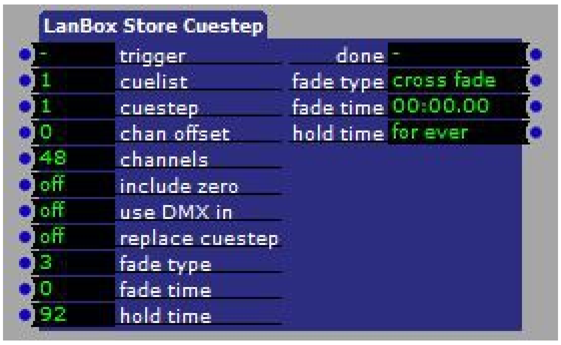
LanBox Scene Fader

0 fader 0-127
27 layer
1 cuelist
1 cuestep
- initialize
- update

channels 5
update ok -
init ok -

This actor is an improvement of the LanBox Submaster actor I put to the forum in march, see http://forum.troikatronix.com/cgi-bin/forum/gforum.cgi?post=13725;t=search_engine#13725)
With this actor you can send a DMX values, which are calculated by multiplying channel values of a given cuestep with a value between 0 and 100%. The idea was to have certain light scenes (i.e. cuesteps in the LanBox) on a fader and maybe adding this to a running cue. For example you want to have applause light, audience light or a follow spot separated from the rest of the show.
This actor needs an init command, which creates a new layer and prepares it by switching to mix mode HTP. This new layer is on top of the existing layers, it's channel values are sent to the DMX output if they are highest of all layers. After initializing, a cuestep of the LanBox is read and calculations are made with every channel value (by multiplying with the fader value) and are written to the layer.
Every time the fader value comes from a positive value to zero, the cuestep values are updated, which means that changes in the input parameters cuelist and cuestep and changes in the LanBox memory are read. This auto-update functionality is the change between the old LanBox Submaster actor and this new one.
Input parameters:
fader 0-127: Because of the primary intention of using this actor in conjunction with a MIDI fader board, the range of the fader is from 0 to 127. The fader value determines the intensity of the cuestep. When going down from a positive value to zero, changes of the input parameters cuelist and cuestep are detected and the cuestep is loaded into the actor. If the requested cuestep does not exist, no channel values are sent to the layer.
layer: Integer from 1 to 63, the range of layers in the LanBox. In the LanBox terminology, these numbers correspond to Layer IDs from A to BK. Please use different numbers for every use of this actor.
cuelist: Integer from 1 to 999.
cuestep: Integer from 1 to 99. These parameters determine which cuestep is used in the actor.
initialize: After connecting to the LanBox, this input should be triggered.
Output parameters:
channels: Here you see how many channels are used in this cuestep. Please note that the LanBox command for reading a cuestep provides not more than 250 channel values. I think that you can live with that limitation.
update ok: This trigger indicates that the requested cuestep exists and is read into the actor.
init ok: After initializing this output is triggered. This is useful when initializing more than one LanBox Scene Fader Actor – you can chain all initialize triggers and get one input trigger for initializing all.

LanBox Store Cuestep


LanBox Store Cuestep

- trigger
1 cuelist
1 cuestep
0 chan offset
48 channels
off include zero
off use DMX in
off replace cuestep
3 fade type
0 fade time
92 hold time

done -
fade type cross fade
fade time 00:00.00
hold time for ever

The use of this actor is to store DMX values into a cuestep. The cuestep will be created if not existing. The parameters of a new cuestep are set by input values. The DMX values, which will be stored in the cuestep are either the actual DMX output values or the actual DMX input values. So you can store what you see as a cuestep or you can copy a step of another desk to a cuestep of the LanBox.

A cuelist in the LanBox consists of cuesteps, which are numbered consecutively from 1. So the actor includes comment-lines if there is a gap between the actual last and the new cuestep. Depending on the input parameter 'replace cuestep', the new cuestep will be inserted at the desired position or the actual channel values replace the channel values of the cuestep. In case of replacing the cuestep parameters remain unchanged. This makes sense when you have to make corrections in the light intensities. The fading parameters stay as they were set.

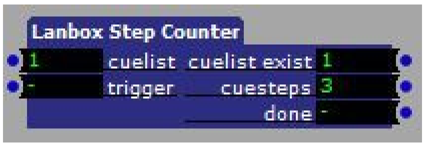
The step type of the cuestep has to be mentioned. In the LanBox a cuestep can have over 30 different step types. So a cuestep can be used not only to show a scene (which is step type 1) or declare a comment-line (which is step type 112). With other step types you can declare jump addresses or write MIDI or serial streams for example. The LanBox Store Cuestep actor only uses cuesteps with type 1 and 112. If a cuestep is stored with this actor the step type is set to 1.

Input Parameters:
trigger: when activated, the copy process starts. Depending on the number of channels and the speed of the tcpip connection this may take a few seconds.
cuelist: integer value between 1 and 999. This is the cuelist number in the LanBox. If this cuelist does not exist, it will be created.
cuestep: integer value between 1 and 99. This is the number of the cuestep to be created, if not existing. Since a cuelist consists of a list of subsequent cuesteps, lines between the actual last cuestep of the list and the desired cuestep are filled up with comment-lines.
chan offset: integer value between 0 and 511. Defines the starting point of value read/write. A chan offset of 0 starts the process at DMX 1.
channels: integer value between 1 and 512. Defines the number of channels, which are stored/replaced in the new cuelist. The channels parameter is mainly responsible for the speed of the copy process.
include zero: Toggle with values on/off. If off, zero values of DMX channels are skipped in the copy process. if on, zero values of DMX channels are written to the cuestep.
use DMX in: Toggle with values on/off. If off, the actual DMX output will be stored as cuestep. If on, the incoming DMX values are stored as cuestep. Control the DMX setting in your LanBox, you find it in the LCedit software: Tools-Global settings-DMX.
replace cuestep: Toggle with values on/off. If on, the channel values of an existing cuestep will be replaced by the new values. If off, the new cuestep will be inserted at the desired line and the subsequent cuesteps change their number. This input parameter has no meaning when the cuestep is appended at the end of the cuelist. Please note: If 'replace cuestep ' is on, the cuestep parameters

remain unchanged.
fade type: integer between 1 and 7. There is a fade type table in the LanBox. You can see in the output parameter the meaning of every number translated to a String.
fade time: integer between 0 and 91. The fade time is encoded in a table, you can see the actual fade time in the output parameter as a string.
hold time: integer between 0 and 92. The translation from number to time is seen in the corresponding output parameter.

Output Parameters:
done: trigger at the end of the process.
fade type, fade time and hold time: These parameters are string translations of the corresponding input parameters.

LanBox Step Counter


Lanbox Step Counter
cuelist 1
trigger -
cuelist exist 1
cuesteps 3
done -

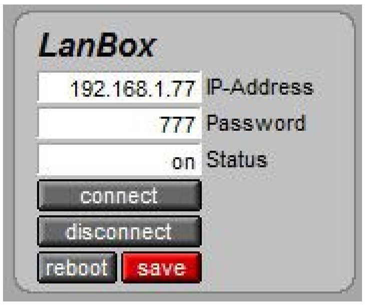
With this actor you get information about a cuelist: If the cuelist exists and of how many cuesteps it consists. This actor was necessary for programming the LanBox Store Cuestep actor, but it is useful by itself also.

Input parameters:
cuelist: integer between 1 and 999. The cuelist to be investigated.
trigger: sart the process.

Output parameters:
cuelist exist: 1 indicates that the cuelist exists. 0 indicates that the cuelist does not exist.
cuesteps: the number of cuesteps (integer between 0 and 99). If the cuelist does not exist, the output is set to 0.
done: end trigger.

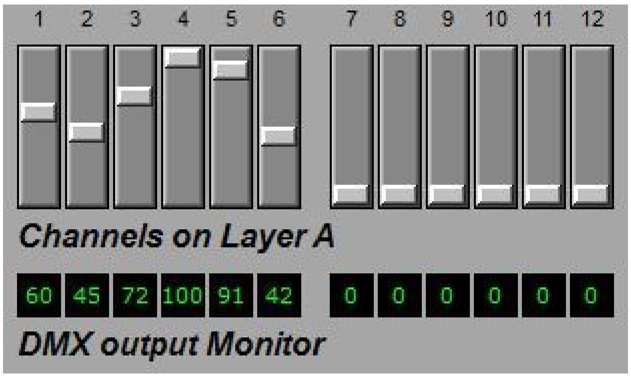
Part 3: Example

In the example LanBox TCP basics.izz you find all these actors and a working light desk in a simple form: 24 channel faders, a 24 channel DMX monitor, a scene fader with rec-function and of course the control for the LanBox connestion.

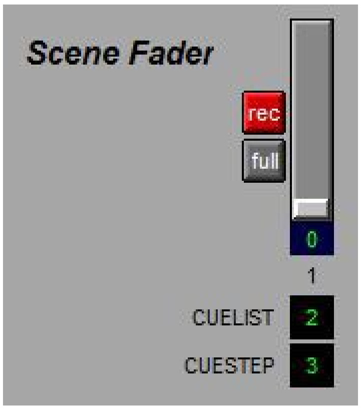

LanBox
IP-Address 192.168.1.77
Password 777
Status on

As you can see, you find inputs for IP-address and password. After a successful connection is established, the (Isadora) scene „advanced“ is activated. This starts the lightdesk. To run the lightdesk, stay on the (Isadora) scene „global“.

You can make a light scene using the channel faders. The DMX output monitor shows you percentage values of the channels. Of course you can also run the LCedit+ software and compare in



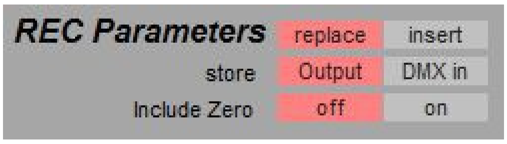
the DMX output window. For testing the scene fader, you can save the DMX output as a cuestep in the LanBox:



CUELIST 2

CUESTEP 3

Type in the desired cuelist and cuestep an press the rec button. If neccessary, the cuelist will be created. According to the rec parameters you will create or replace an existing cuestep. Control the result in the Cuelist editor of Lcedit+.



	replace	insert
store	Output	DMX in
Include Zero	off	on

After creating cuesteps you shuld save the LanBox data with the save button. If not, the cuesteps are gone after rebooting the LanBox. You can see in the Control window of LCedit+ that there is a new layer AA, which is neccessary to run the scene fader. If you plan to use more scene faders (I tested 9 faders) make suer that each scen fader actor gets a unique layer number. I start with layer number 27 to create a dynamic layer. This layer is deleted when the LanBox is shut down or rebooted.