Talking Ember+ to Junger equipment

The **Ember+** protocol was designed by Co. l-s-b : http://www.l-s-b.de/en.html
(see also the product **VSM** - Virtual Studio Manager and panel solutions).

For Junger Audio devices the implementation of the **Ember+** protocol allows 3rd party application to gain access to parameters of the respective device. But you can also receive status information and react to it.
These parameters are available via the GUI interface as well. So during implementation it might be a good idea to have a parallel look at the GUI if you must understand the parameters in their context.

Usually the tasks are simple. E.g. load pre-defined settings (AKN presets), fire soft GPIs that on the other hand may trigger actions inside the device or change a parameter like the gain of a processing channel to name a few.

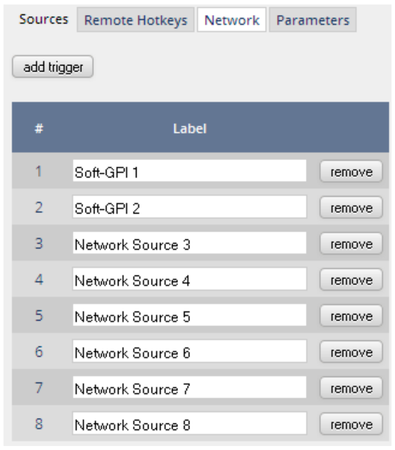
You must implement an **Ember+** client for your application (e.g. for a broadcast automation system) that talks to the **DUC**. You will find **Ember+** protocol details and implementation guide-lines as well as example codes here : https://github.com/Lawo/ember-plus/

We suggest that you download a package to get hands-on experience. For a software developer it is no virgin soil …
If you want to get hands on experience and you have a Junger device that has **Ember+** protocol implemented we would recommend installing the latest **Ember+ viewer** that comes with each package.

Here we want to briefly discuss as an example the control of **Network triggers** of the ***AP** series of 1RU devices and how to load presets for the c8k system.

The triggers are set / reset by a control instance (e.g. VSM, a broadcast automation system or a mixing desk). For the VSM system you may assign these triggers to virtual panels or to buttons of physical hardware panels (e.g. LBP). A broadcast automation system on the other hand may bind these triggers to events of a play list while a mixing desk may assign it to control buttons.

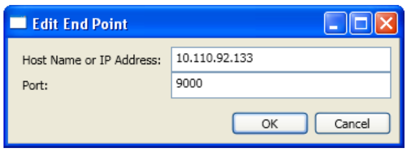
We have named the first two Network triggers "Soft GPI-1" and "Soft GPI-2".
They will be activated from the automation system if the respective event has been reached in the play list or if the operating guy has pressed the respective control button.
EVENTS > Triggers > Sources = Network



#	Label
1	Soft-GPI 1
2	Soft-GPI 2
3	Network Source 3
4	Network Source 4
5	Network Source 5
6	Network Source 6
7	Network Source 7
8	Network Source 8

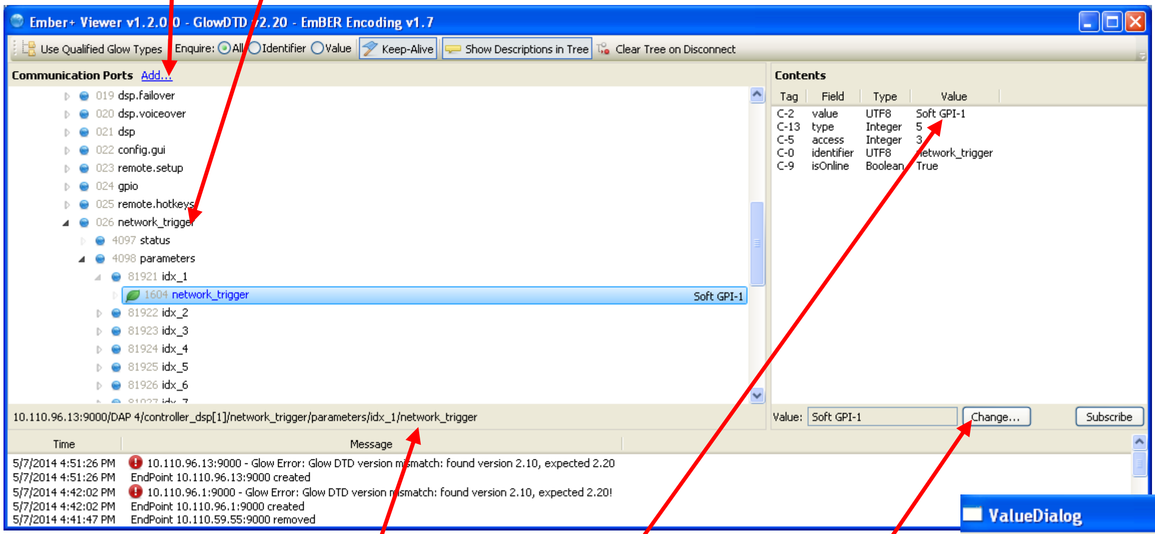
#	Order number of the network trigger.
Label	Label of that network trigger. It appears on the **Configuration** pane as well as in the **Ember+** tree.

Lets have a look at the **Ember+** tree to find the "Soft GPI-1" trigger.
At this stage we are using the **Ember+ viewer tool** (EmberPlusView.exe).
You must **add** a communication port :



Host Name or IP Address:	10.110.92.133
Port:	9000

If you have selected that port you must drill down to **network_trigger** :


Ember+ Viewer v1.2.0.0 - GlowDTD v2.20 - EmBER Encoding v1.7
Use Qualified Glow Types
Enquire: All Identifier Value
Keep-Alive
Show Descriptions in Tree
Clear Tree on Disconnect
Communication Ports Add...
Contents
Change...
Subscribe

Enquire: [✓ All] Identifier Value

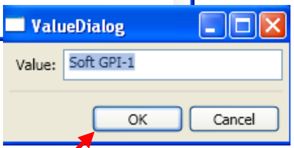
```
019 dsp.failover
020 dsp.voiceover
021 dsp
022 config.gui
023 remote.setup
024 gpio
025 remote.hotkeys
026 network_trigger
  4097 status
  4098 parameters
    81921 idx_1
      1604 network_trigger                Soft GPI-1
    81922 idx_2
    81923 idx_3
    81924 idx_4
    81925 idx_5
    81926 idx_6
```

Tag	Field	Type	Value
C-2	value	UTF8	Soft GPI-1
C-13	type	Integer	5
C-5	access	Integer	3
C-0	identifier	UTF8	network_trigger
C-9	isOnline	Boolean	True

Value: Soft GPI-1

```
10.110.96.13:9000/DAP 4/controller_dsp[1]/network_trigger/parameters/idx_1/network_trigger
```

```
Time                  Message
5/7/2014 4:51:26 PM   10.110.96.13:9000 - Glow Error: Glow DTD version mismatch: found version 2.10, expected 2.20
5/7/2014 4:51:26 PM   EndPoint 10.110.96.13:9000 created
5/7/2014 4:42:02 PM   10.110.96.1:9000 - Glow Error: Glow DTD version mismatch: found version 2.10, expected 2.20!
5/7/2014 4:42:02 PM   EndPoint 10.110.96.1:9000 created
5/7/2014 4:41:47 PM   EndPoint 10.110.59.55:9000 removed
```



Value: Soft GPI-1

The path is also shown at the
bottom of the main window :

As a value you will receive the trigger name **"Soft GPI-1"**
you have assigned to the first network trigger.

You may now fire that trigger by pressing **<Change … >** and **<OK>** in the pop-up.
This is similar to sending the trigger as a pulse (behaves like a push button).

But you may also give it a value of **"1"** to set the trigger and later on a **"0"** to reset it depending on the functionality you need for a specific application (if you implement a toggle switch).

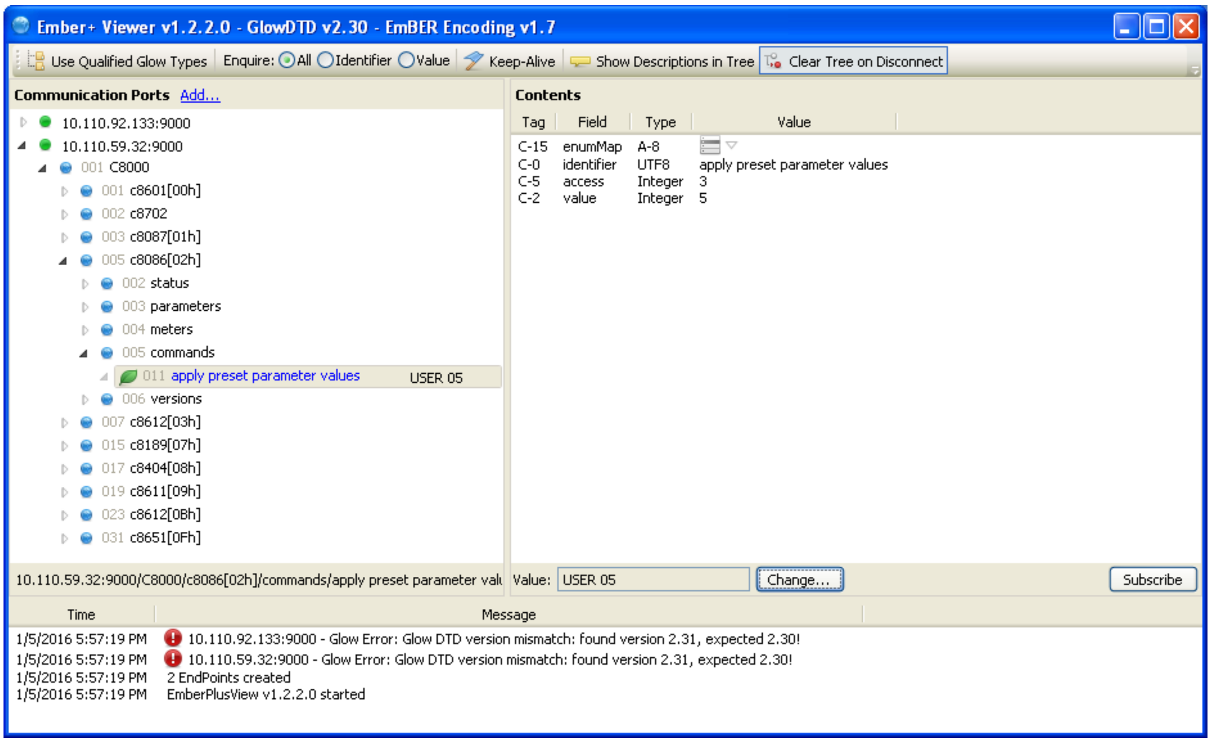
To change a parameter value a 3rd party **Ember+** client implementation must provide the partial tree up to idx_1 > network_trigger with a new value.

To round up this brief explanation, here is the way how to load presets of a specific module inside a c8k frame.
It is mandatory that you are familiar with how the c8k frame works and how you change things using the web GUI.

The screen shot below shows a c8k frame (IP address 10.110.59.32) containing 9 modules beside the frame controller C8702 that runs the **Ember+** server. In square brackets you will find the physical module address [N] that was assigned by the person who did the frame configuration. The leading digits are the identifiers of the modules. They are generated from the logical CAN bus address (N*2 +1) that is used internally by the frame controller to communicate with the modules.

If you now go to the LevelMagic processor **C8086[02h]** and drill down to the **"commands"** identifier you will see: **"apply preset parameter values"**. These values are the preset numbers of that module
(refer to the PRESET page of the C8086 for details):
01 – 16 for DSP
17 – 32 for DOLBY METADATA
33 – 40 for SETUP

You can now change this value. Press **<Change>** and set a value between 1 and 40 and press <OK>.


Ember+ Viewer v1.2.2.0 - GlowDTD v2.30 - EmBER Encoding v1.7

Use Qualified Glow Types | Enquire: [✓ All] Identifier Value | Keep-Alive | Show Descriptions in Tree | Clear Tree on Disconnect

Communication Ports Add...

```
10.110.92.133:9000
10.110.59.32:9000
  001 C8000
    001 c8601[00h]
    002 c8702
    003 c8087[01h]
    005 c8086[02h]
      002 status
      003 parameters
      004 meters
      005 commands
        011 apply preset parameter values        USER 05
      006 versions
    007 c8612[03h]
    015 c8189[07h]
    017 c8404[08h]
    019 c8611[09h]
    023 c8612[0Bh]
    031 c8651[0Fh]
```

Contents

Tag	Field	Type	Value
C-15	enumMap	A-8	
C-0	identifier	UTF8	apply preset parameter values
C-5	access	Integer	3
C-2	value	Integer	5

10.110.59.32:9000/C8000/c8086[02h]/commands/apply preset parameter valı Value: USER 05 Change... Subscribe

```
Time                  Message
1/5/2016 5:57:19 PM   10.110.92.133:9000 - Glow Error: Glow DTD version mismatch: found version 2.31, expected 2.30!
1/5/2016 5:57:19 PM   10.110.59.32:9000 - Glow Error: Glow DTD version mismatch: found version 2.31, expected 2.30!
1/5/2016 5:57:19 PM   2 EndPoints created
1/5/2016 5:57:19 PM   EmberPlusView v1.2.2.0 started
```

Changing a parameter value [an excerpt from the implementation manual]

If a consumer wants to change a parameter value, he has to provide the partial tree and the new value.
The following sample shows what a "change network mask" request looks like. This is a code example from the **Ember+** manual.

```
root = GlowRootElementCollection::create();
device = new GlowNode(1);
network = new GlowNode(3);

mask = new GlowParameter(2);
mask->setValue(“255.255.252.0”);

root->insert(root->end(), device);
device->children()->insert(device->children()->end(), network);
network->children()->insert(network->children()->end(), mask);
```

When a provider detects a parameter containing a value, he must treat this value like a value change request.
That's it.