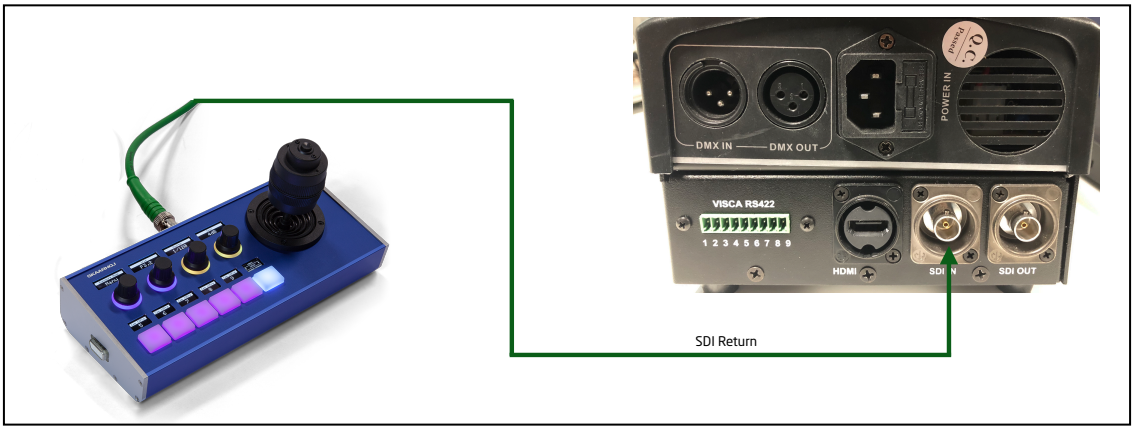
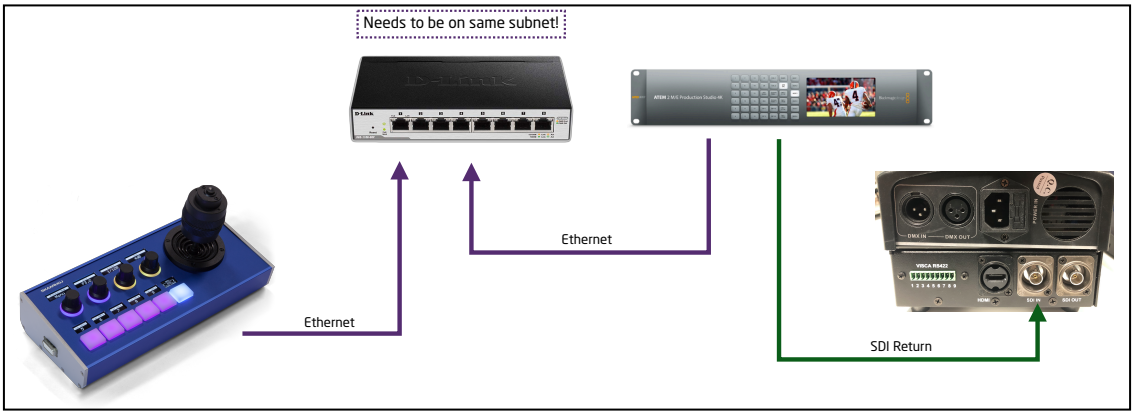
Device: RUSHWORKS PTX + BMD Micro Studio Camera 4K Combination

Introduction

Our controllers work with the RUSHWORKS PTX1 and the Blackmagic Micro Studio Camera 4K combination. This setup can be controlled two ways:

- Via the BMD CamCtrl Device Core (using the Blackmagic Arduino 3G-SDI Shield) where the control signal originates from the integrated Arduino Shield. This requires a controller fitted with the "SDI" option.
- Via the ATEM Device Core where the control signal originates from the BMD ATEM Switcher

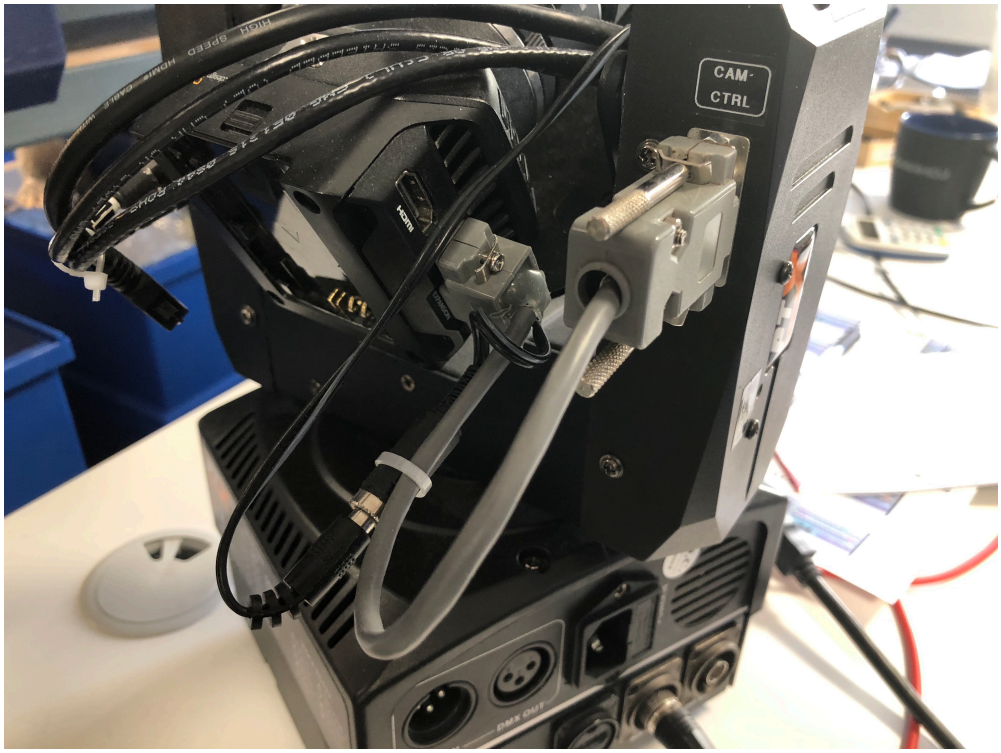




This is the breakout cable used from the Micro Studio Camera 4K to the Rushworks head.

The signal path is as follows

- SDI signal embedded with ancillary data (control data for the camera + head) originating from SKAARHOJ controller or ATEM are fed into the SDI IN on the Rushworks head (an onwards to the camera)
- A breakout cable from the Expansion port on the camera are fed into the "CAM CTRL" plug on the Rushworks head

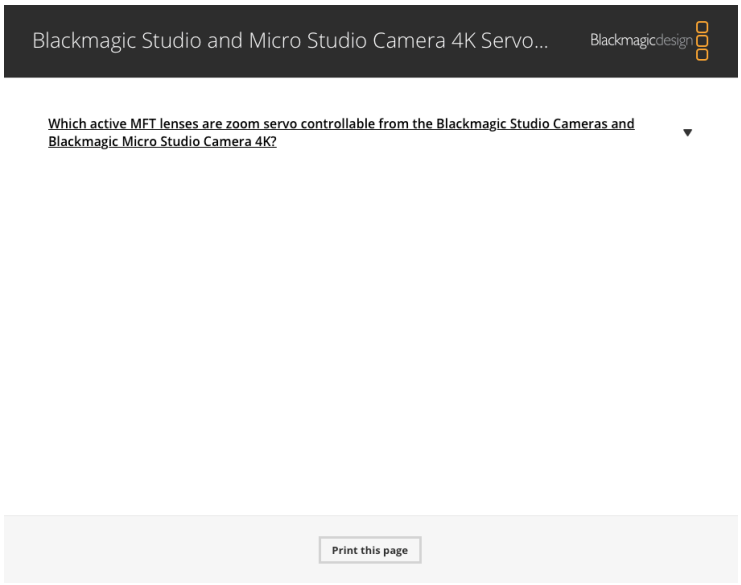


This is how the jumper settings looks like in the setup we have been testing with.

Zoom on the Micro Studio Camera 4K

If a servo zoom lens is put on the camera zoom can be controlled from our controllers as well. See the official Blackmagic note to find a compatible lens:

https://www.blackmagicdesign.com/support/faq/59009



Small Demo

See a small demo of the PTZ Fly with SDI + Rushworks in action here:

https://github.com/kasperskaarhoj/SKAARHOJ-Open-Engineering/raw/master/Manuals/Files/RUSHWORKS_BMD_MicroStudioCamera4K_Demo.MOV

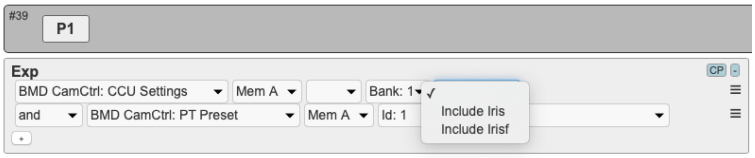
Controlling multiple cameras/PTX Heads

This is possible. The SDI return feed to the camera needs to be distributed from either the SKAARHOJ controller with SDI option or the ATEM. On the BMD camera you need to select propper camera ID.

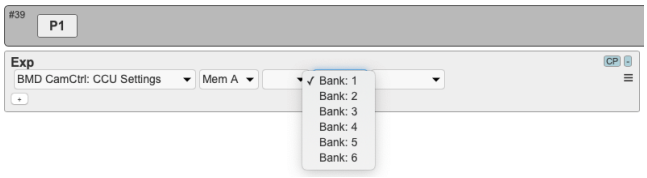
Notice about saving/recalling presets

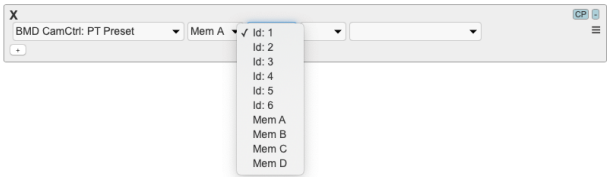
When saving/recalling a preset for a BMD camera/PTX head combination one would typically utilize *two* actions. One saving "CCU Settings" such as white balance, color adjustments and so forth and another action saving pan/tilt presets.

For CCU settings iris can be included or excluded for the preset.

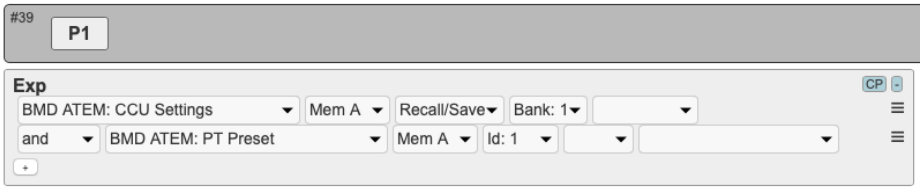


Please notice *only* a total of 6 banks/IDs are available to save CCU settings/PT Presets. Not per camera but **in total**. The reason being is that the actions was originally developed for the RCP which as designed to control just *one* camera at a time.





Please also observe zoom and focus are *not* included in either the CCU Settings or PT Presets



Controlling PTX Head without BMD Camera but via VISCA RS422 (Beta)

Please notice the following instruction are in beta and subject to change.

The PTX Head can also be controlled from a SKAARHOJ panel using the VISCA RS422 port on the PTX head via an Ethernet-Serial converter. We suggest you get a XS1200 from US Converters - http://www.usconverters.com/serial-rs232-device-server

In order to control the head the Device Core "Generic VISCA" must be used. The IP address for the Device Core should match the XS1200 converter box. Only the actions Pan, Tilt and Presets on the Device Core will work.

Device Core Option Generic VISCA

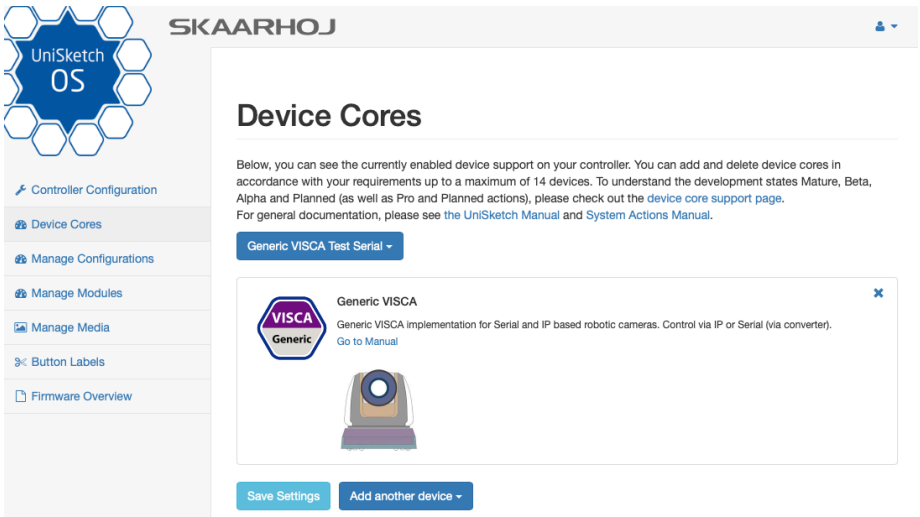
Control via Serial instead of IP must be enabled on the Generic VISCA Device Core

Device configuration options exist:

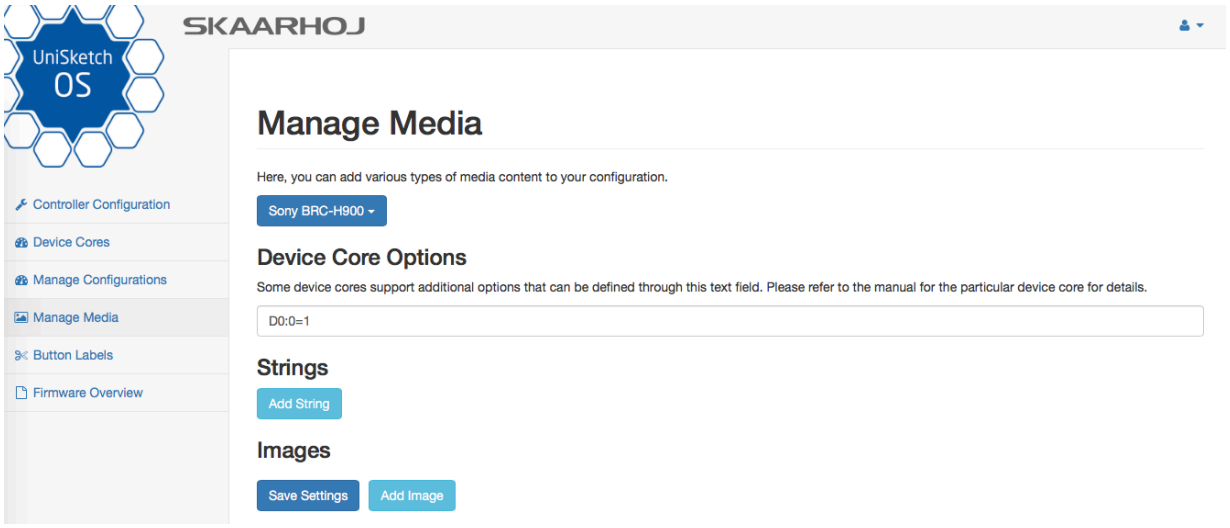
- Index 0: **VISCA over IP/Serial**
  - If "1" = VISCA over Serial

Example:

Enabling VISCA over serial could look like this device configuration code: "D0:0=1" where the general form would be "Dx:y=z" where "x" is the number of the device core as installed on the controller (starting with zero for the first device core), "y" the index number and "z" the value for that index. If the Generic VISCA Device Core is the first like below



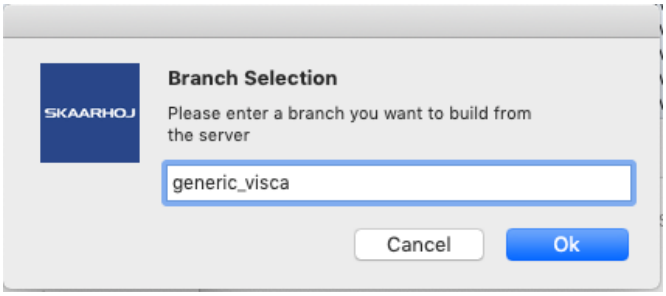
Setting VISCA over serial would be set by this configuration under "Manage Media" on the configuration page for your controller. Access this by pressing "Online Configuration" in the Firmware Application. Remember to save on the configuration page.



Compiling Firmware from Brach

Special note: In order to use the Device Core together with the PTX head a special firmware must be generated from a branch called "**generic_visca**".

Please acquire information with us directly for instructions on how to compile firmware from this branch.



When both options are matched (VISCA over Serial and branch firmware) the boot up process in the Serial Monitor should state this:

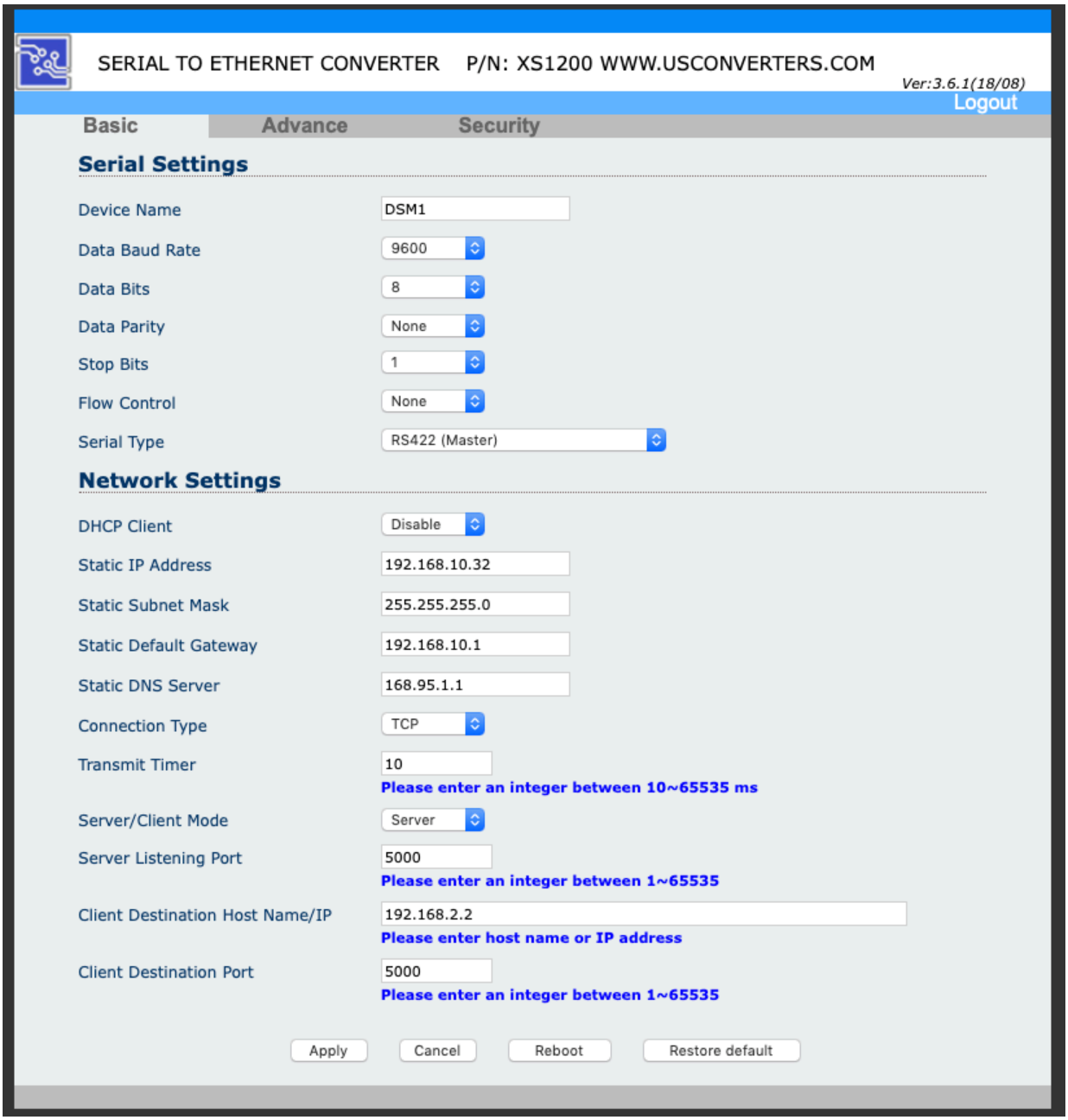
```
DeviceCore #0: GENERICVISCA0, IP = 192.168.10.32
ClientVISCAserialIP:  __deviceIdx: 0
ClientVISCAserialIP::begin()VISCAbase: DISABLING retransmit
VISCAbase: DISABLING acknowledgements
setup() Done
------------------------------
HWc#11 Down Speed: 0
```

There is a quirk you should know about: The XS1200 only accepts a single TCP connection at a time and it will take some time to realise if a client disconnected silently before it allows a new connection. In essence this means if the SKAARHOJ controller was connected and is rebooted without disconnecting, the XS1200 Server may not realise this before after some time. Therefore you may need to powercycle it along with the SKAARHOJ controller to make sure it will accept a connection.

Connection to the XS1200 can be confirmed with the message ".Connected to serial converter"

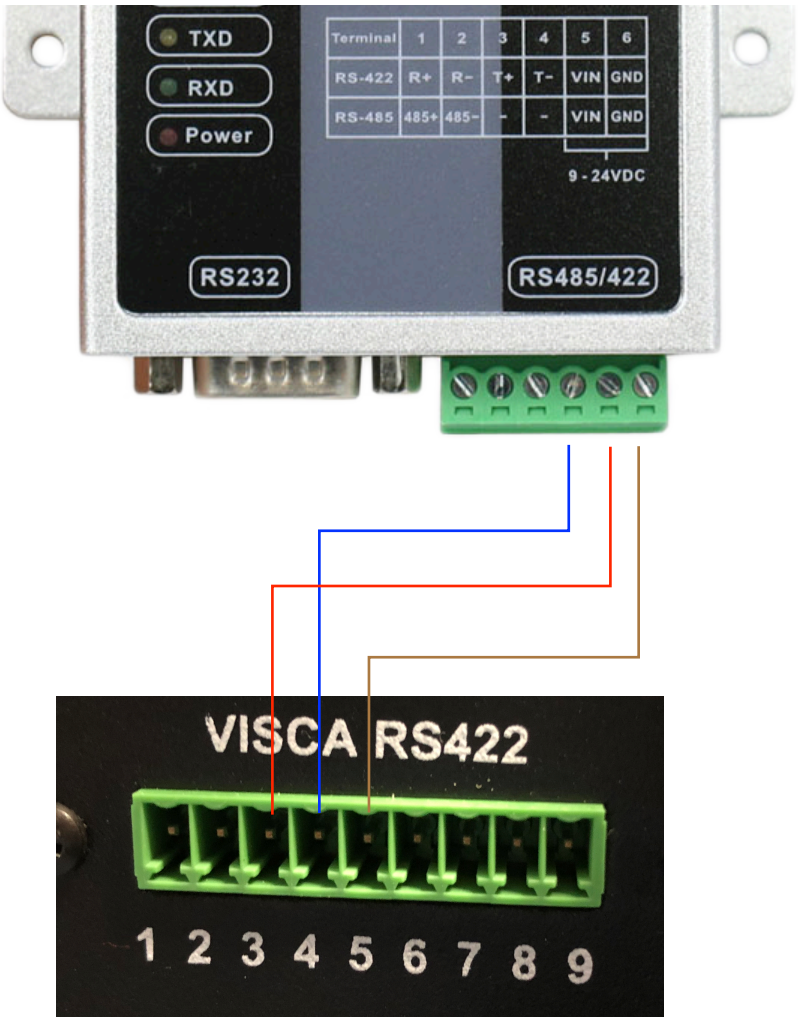
```
.87
.Connected to serial converter
87
.
```

Below you will find screenshots of how to configure the XS1200 converter (found of the web interface of the XS1200).



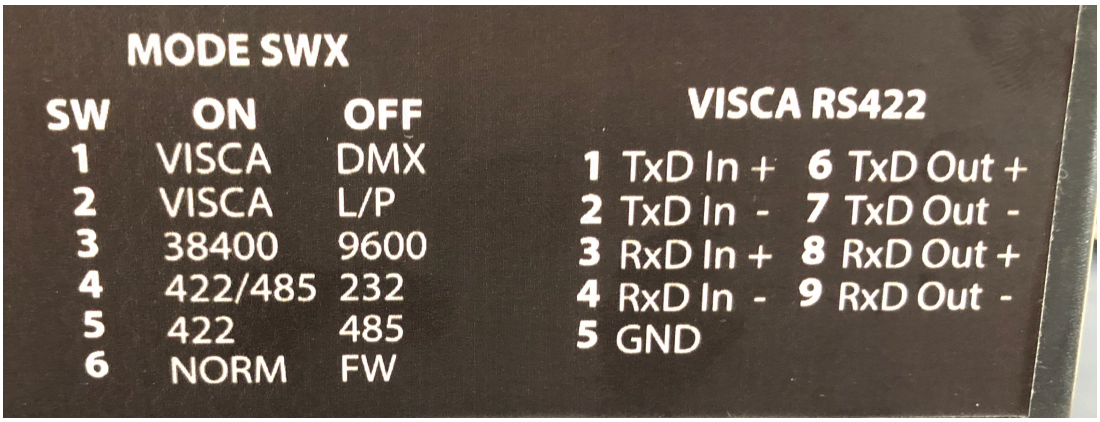
Make sure to set up an IP address in your range here. This is the IP address you must also set up inside the SKAARHOJ controller for the Device Core! Here it is set to 192.168.10.32 and corresponding subnet mask.

Cabling to the XS1200 is via the RS-422 connector. 3 wires are necessary. GND and then T+ and T-.



The coloration between these and the XS1200 is the following:

GND (6) = GND (5)
T+ (3) = RxD In - (4)
T- (4) = RxD In + (3)



The dip switches on the PTX heads should be set to the following:

3 should be off for baud rate of 9600.

A quick and dirty video of the setup can se seen here: https://github.com/kasperskaarhoj/SKAARHOJ-Open-Engineering/raw/master/Manuals/Files/RUSHWORKS_VISCA_RS422_Control_Beta.mov