Manual for your SKAARDUINO/UniSketch powered SKAARHOJ controller!

Congratulations with your SKAARHOJ controller! We are really proud of how much we have been able to stuff into this device, and we hope you can see our love and passion for cool and helpful technology shine through when you browse this manual.

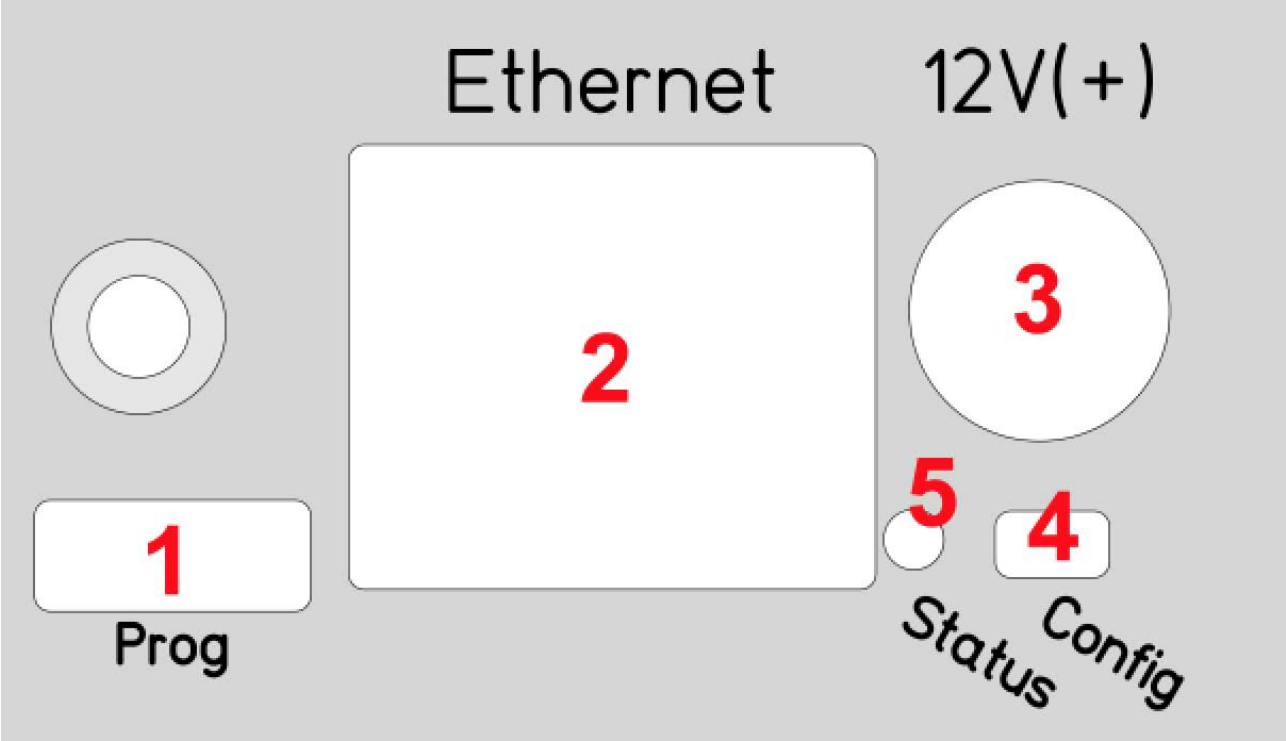
While we really want to make everything intuitive for you, we still need to document some not so obvious facts and conventions and we have tried to put everything you need to know into this document along with a few extra tips too.

Have fun!

Back Connections

Your SKAARDUINO powered SKAARHOJ controller will have a set of connections like the ones you see below.

The ”SKAARDUINO” is the micro processor platform inside which is a variant of the Arduino type micro processor platforms which we have developed ourselves to be smaller and more powerful at the same time so it perfectly fits our products and other professional applications.



1. **Micro USB plug.** Used for the serial monitor and programming new software into the unit (”firmware upgrades”). See ”Programming”
2. **Ethernet Jack.** Connect this to your Ethernet Switch. This jack may also support PoE (Power over Ethernet) if your SKAARHOJ controller was ordered with that option. When connected to a network switch, the yellow LED (lower left) will be on. When data is sent

to/from the controller, the green LED (lower right) will blink. If the device in the other end supports TX/RX auto detection you may be able to connect the SKAARHOJ controller directly to your device, otherwise use a crossed cable or a network switch (the supported setup).

3. **DC input.** Use a standard 2.1mm center pin plug (center = ”+”). Allowed voltage range is 7-18V. We test controllers will work at 12V. The device uses max 500mA at 12V.
4. **Configuration/Reset button.** Use a pencil or tooth pick to press the button. When you press the button shortly, the controller will reset (same as pulling the power plug). If you press and hold the button, you can reset the controller into configuration mode:
    1. Press and hold the button until the status LED becomes blue. Release the button and the controller is in *config* mode. You can access the controller web interface with a web browser on ”http://[CONTROLLER IP]/” where CONTROLLER IP is the IP address used for the currently loaded preset.
    2. Press and hold the button until the status LED becomes white (which is 2 seconds after becoming blue). Release the button and the controller is in *config default* mode. You can access the controller web interface with a web browser on ”http://192.168.10.99/”.
5. **Status LED:** When the controller is just powered up, you will see the status LED blink purple during the boot process. In this process, the hardware is initialized. Eventually the LED should end up blinking slowly (2 sec period) steady green (or blue or white if in config modes). If the LED blinks yellow quickly it indicates that connection to one or more devices is not established. This is perfectly normal for a few seconds between the boot up process (purple blinks) and the operational state (green blinks) when the controller connects to all devices. The status LED should never be permanently on or off, this indicates a potential freeze in the system. In fact, in normal healthy operation the LED should blink with a steady 2 sec period, otherwise it could indicate trouble with connections or hardware.

Status LED overview

Purple blinks, uneven durations	The controller is booting up and for each blink a given step has been completed.
Yellow blinks, quickly	The controller hasn't established necessary connection to one or more devices. At the end of the boot process this is natural for a few seconds as the controller connects to devices for the first time. If you unplug the network cable or turn off an external device the controller is connected to, you will also see this state. Just turn on the external device again or re-insert the cable and the error state should restore itself to normal operation (green, steady blinking) after some time. If this happens during normal operation and without obvious explanations (like removal of a network cable or shutting down an external device), it's an error state you need to pay attention to and bugfix further. If the controller boots up and never stops blinking yellow, you may want to check if you have configured devices for the controller which are not currently present in the network set up. Go to config mode, enter the web interface and check which devices are enabled and their IP addresses.
Green blinks, steady, period of 2 seconds	Normal mode, everything is connected and working properly. Just bliss.
Blue or white blinks, steady, period of 2 seconds	Config mode (white: config default) where you can access the controller web interface.

Connection trouble shooting

When you have a ”blinking-yellow-quickly” situation, you need to figure out which device is not connected. Try some of these things:

- you should bring the controller into config mode and access the web interface in order to check which devices are set up and what their IP addresses are.
- make sure the SKAARHOJ controller itself has the expected IP address and subnet mask.
- Make sure the devices you have setup actually are on the network, can be ping'ed and responds to their respective other types of software connecting over IP.
- Unpower all devices and your network switch for 10 seconds and power them up again.
- Connect a computer to the USB port of the SKAARHOJ controller and open the serial monitor to see the output from that. This provides the most direct information about which devices are not answering and at which IP. You can also see the controller IP and Mac address here. See ”Serial Monitor” section.

Configuration mode

In Configuration Mode all device communication is disabled and instead the controller provides a web interface for configuration of the interface component behaviors. The IP address of the controller in configuration mode (or ”config” mode) depends on how config mode was entered. There are two options: ”config” mode or ”config default” mode. You can enter either mode using a) the Config/Reset button on the controller or b) using the serial monitor command ”config” or ”configd”.

- ”config” mode: The controller IP address is the ”last used” IP address; the one set up for the currently loaded preset. This is convenient most of the time since you probably know your controller IP and just need to boot in config mode and access the web interface with a browser. After 2 minutes in config mode, the controller will run cyclic test programs on the hardware interface components, typically a lot of blinking.
- ”config default” mode changes the IP address to 192.168.10.99 and this is useful if for some reason you forgot the controller IP or otherwise want to make absolutely sure you know the right IP address for the controller. The controller will run cyclic test programs on the hardware interface components immediately as it has booted up.

Notice that the IP address of your SKAARHOJ controller can be different for each preset you have! This makes it easy to have presets for completely different network and device configurations.

Web Interface Troubleshooting:

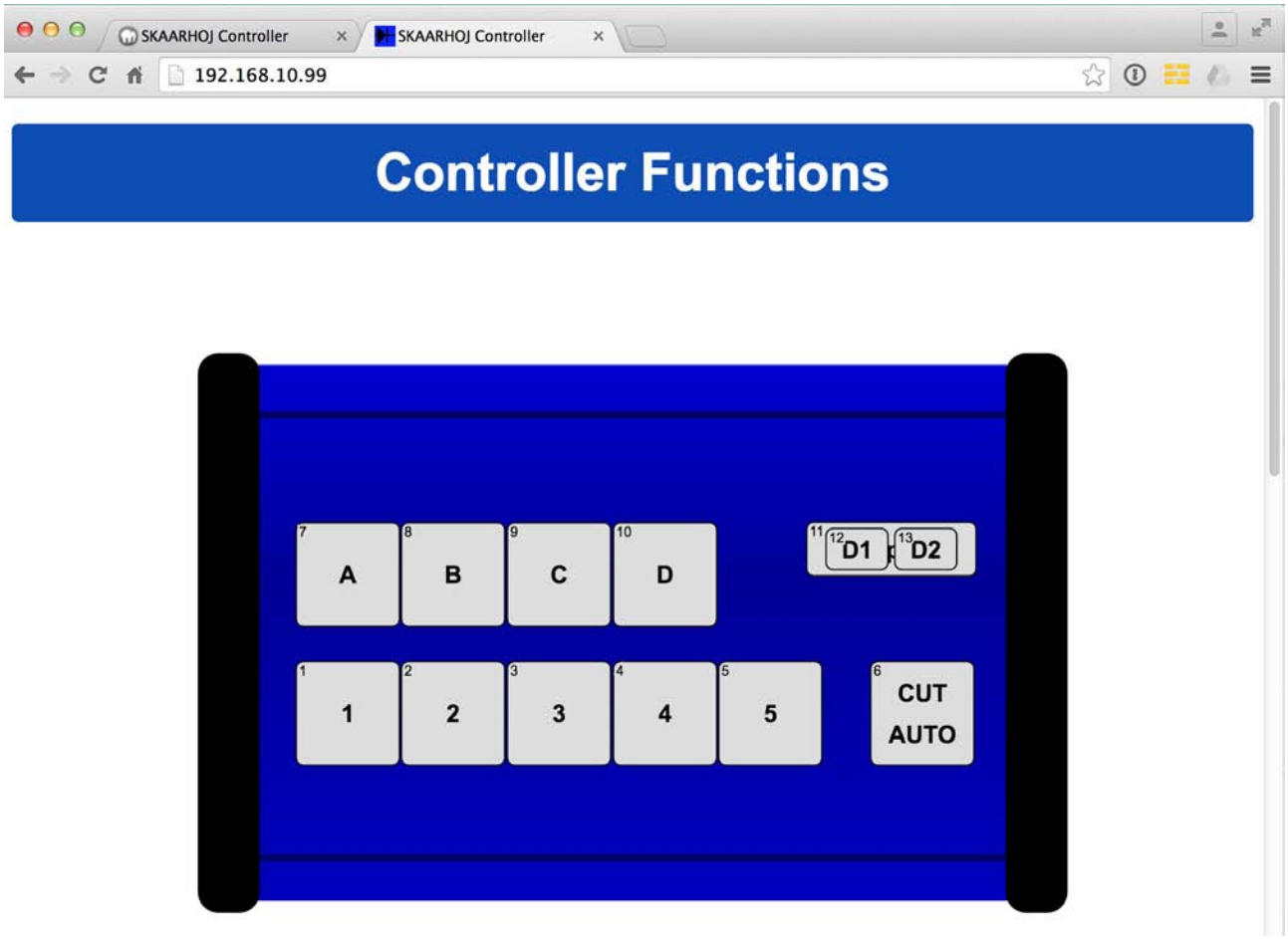
If you are having problems connecting to the web interface there are numerous things that can trick you:

- Reboot your SKAARHOJ controller completely (disconnect for 10 seconds, re-power)
- Reboot your network switch. Sometimes it may hold old information about the controller IP and MAC adresses and to clear this out, reboot it (disconnect for 10 seconds, re-power)
- Check connection to the assumed SKAARHOJ controller IP from your computer with the ”ping” command. You computer IP settings must be correct too.

Web Interface

When your controller is in configuration mode, you will be able to access it's web interface with a web browser. We recommend using the Chrome browser because it's faster, but we have successfully tested it with other modern browsers such as Firefox, Safari and Internet Explorer in the latests version.

Notice that the web interface heavily depends on modern JavaScript and may malfunction with older browsers!



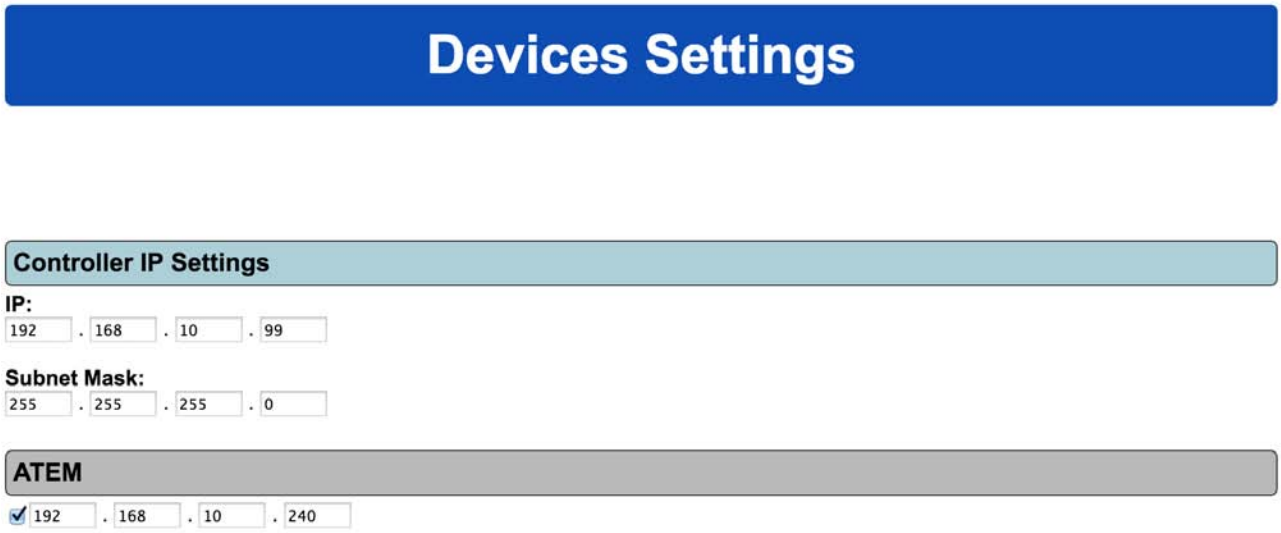
Example of web interface for a SKAARHOJ controller.

Device Settings

You set up the IP address and Subnet mask of your SKAARHOJ controller in the web interface. Likewise any external IP device your controller is configured to work with is listed here. Any device you want to be active must be enabled here and have a valid IP address set up.

Notice that all these IP settings will be saved with each preset in the controller. This is quite awesome because different presets allows you different IP configurations so a controller can easily move between different hardware contexts.

The number and type of external devices listed is compiled into your controller when it is delivered. The list may vary from controller to controller and can potentially be expanded or change to include other hardware by time.



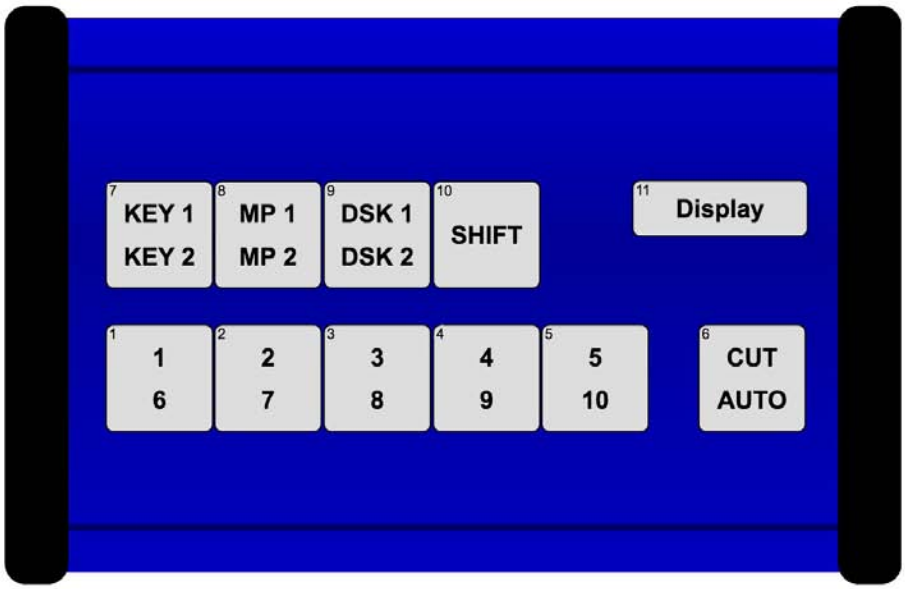
Hardware Interface Components (HWC)

Your SKAARHOJ controller consists of a number of hardware interface components. That is such as buttons, knobs, dials, levers, displays, LEDs, joysticks, plugs, etc. They generally fall into broad categories such as inputs (eg. Buttons and knobs, GPI) or outputs (eg. Displays, LEDs, relay). Sometimes they can be both (most buttons have an illumination color, some even have a display on them). Input elements can be sub categorized as:

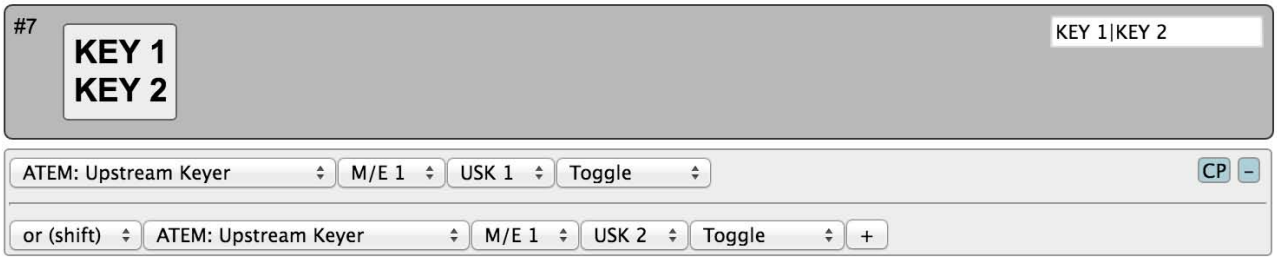
- ”binary” such as a button or GPI trigger which is either pushed/triggered or not. Sometimes holding down a binary input has a special function.
- pulse-generators such as encoders which are knobs that can rotate indefinitely in both directions and sends a corresponding number of pulses. Encoders typically has a fine/coarse adjustment mode which is toggled by pressing it. Pressing and holding an encoder down for 1 second will typically send a binary ”button down” signal to the interface component. Often this can function as a ”reset” feature.
- analog signals such as T-bars or joysticks which provides a free value within some range.

Outputs range from a simple binary output like a relay to red/green LEDs, an array of LEDs (like a VU meter) or a graphical or text based display.

Hardware Interface Components are configured with actions which is what give them their function. This is described in the next sections.



An example of the schematic drawing of a controller as found in the controllers web interface which is available when booted in configuration mode.



This is how the button "KEY1 / KEY2" is configured in the web interface. As it appears, this button will toggle upstream keyer 1 or 2 on an ATEM switcher depending on whether the controller is in shift-state or not.

Graphical Displays

Displays are found in many configurations on SKAARHOJ controllers. One of them is Smart Switches which are buttons with a display on. Otherwise displays are typically stand alone but can be configured to reflect a certain button on the controller (through the "Tie to HWC#" system action). An important convention with displays is whether it works as a label or displays a current status. Take the picture below as an example. Here there are two SmartSwitch buttons apparently showing the same thing:

- A label: The button to the left is configured to set the frame rate of Mix transitions to a fixed "24f". This can be seen from the fact that the button has a non-solid header bar. This is a *label* that simply tells us what will happen if you push the button: You will set a 24 frame mix transition rate.
- A status: The button to the right is configured to also set the frame rate of transitions but is configured to act in "Cycle" mode so when you press the button you will cycle through transition types and values by some scheme. The important thing is that the button shows the current value for Mix transions – 24 frames. And if we change the value to 30 frames, the button will show 30 frames. This is a *status* that informs you about the current value of this ATEM feature and this can be seen from the fact that the button has a *solid* header bar.

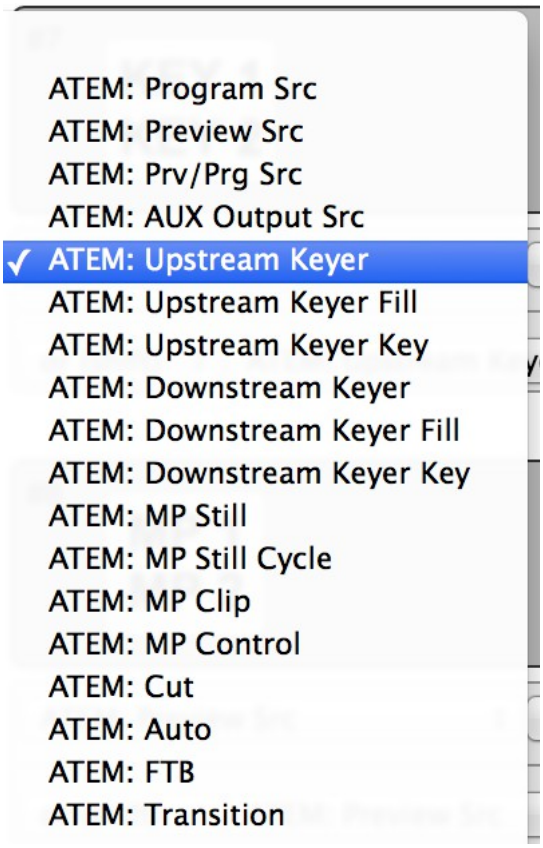
This convention works throughout all displays on your SKAARHOJ controller. Generally, a display will show the value status unless it has been tied to a button-type interface component in which case a label is typically shown – unless the particular button operates in some sort of cyclic mode where a status makes more sense to display.

Devices

You configure your controller by assigning actions to interface components. An action is most typically a command sent to an external device, such as an ATEM switcher, SmartScope, HyperDeck, Videohub or any other supported external hardware.

Some actions may also relate to internal registers or ”system functions”. For instance you can have a button set or clear a ”shift” value which the rest of the interface components will adapt itself to.

Since interface components can be inputs and/or outputs and of various types, the way they affect any given device via an action is a fixed interpretation coded into the system.



An excerpt of the list of ATEM switcher related actions.

System Actions

This is a table of system actions available for any UniSketch powered SKAARHOJ controller:

Shift Level

Sets the controller shift level.

Binary triggers: Sets the selected shift level. If Hold Down is selected, the shift level will fall back to the previous value whenever the trigger is released. Toggle will set the shift level, but on a subsequent trigger, it will fall back to the previous value. If Cycle Up/Down modes are selected, a trigger will set the next/previous shift level up to the level selected.

Pulse inputs: Will cycle through shift levels up to the selected level.

Analog inputs: Will map the analog input range to shift levels up to the selected level.

Binary outputs: On when the shift level matches selected source (or when trigger is held in Cycle modes)

Button colors: Follows binary output: Highlighted, when on.

Displays: Shows the selected shift level either as a number or “Off” (0) or “On” (1)

State

Sets the controller state.

Binary triggers: Sets the selected state. If Hold Down is selected, the state will fall back to the previous value whenever the trigger is released. Toggle will set the state, but on a subsequent trigger, it will fall back to the previous state. If Cycle Up/Down modes are selected, a trigger will set the next/previous state up to the state number selected.

Pulse inputs: Will cycle through states up to the selected state number.

Analog inputs: Will map the analog input range to state numbers up to the selected state.

Binary outputs: On when the controller state matches the selected state (or when trigger is held in Cycle modes)

Button colors: Follows binary output: Highlighted, when on.

Display text: For displays and smart switches, the value will be shown as the preset label entered in the web interface if given, otherwise as a number.

Memory

Sets memory registers A-D. If “Persist” is selected, the value will be stored in EEPROM and recalled from EEPROM upon booting the controller again.

Binary triggers: Sets the selected value for the given register A-D. If Hold Down is selected, the value will fall back to the previous value whenever the trigger is released. Toggle will set the value, but on a subsequent trigger, it will fall back to the previous value. If Cycle Up/Down modes are selected, a trigger will set the next/previous value up to the value selected.

Pulse inputs: Will cycle through values up to the selected value.

Analog inputs: Will map the analog input range to values up to the selected value.

Binary outputs: On when the memory value matches the selected value (or when trigger is held in Cycle modes)

Button colors: Follows binary output: Highlighted, when on.

Cycle Memory 	Sets memory registers A-D with values from specified ranges. If “Persist” is selected, the value will be stored in EEPROM and recalled from EEPROM upon booting the controller again. You can define two From-To ranges (values inclusive) and a single value (“And”, must be different from zero). If both values in the range definitions are zero, the range is ignored. *Binary triggers:* Cycles to the next value given by the range line up. *Pulse inputs:* Will cycle through the values in the range line up. *Analog inputs:* Will map the analog input range to values in the range line up. *Binary outputs:* On when trigger is held in Cycle modes. *Button colors:* Follows binary output: Highlighted, when on. *Displays:* Shows the memory value.
Flag 	Sets a memory flag which is an internal binary value. This can be used to transport binary values around in the system. *Binary triggers:* Sets the selected flag. If Hold Down is selected, the flag will fall back to the previous value whenever the trigger is released. Toggle will set the flag, but on a subsequent trigger, it will fall back to the previous value. *Pulse inputs:* Will flip the value *Analog inputs:* Will clear/set the value when on either side of the middle of the analog value range. *Binary outputs:* On, if the selected feedback flag is set (but subject to modification by the second invert option). Notice that you must select the same feedback flag number as the flag-number if you want it to respond “intuitively”. The idea of the feedback flag is to have a way to send a value out of the system but only reflect a confirmative return value. *Button colors:* Follows binary output: Highlighted, when on *Other features:* If “Invert” is selected, the flag will be cleared (binary zero) instead of set (binary 1) in any of the above cases. If any time frame is set, the value will fall back to the default after that period of time.
Tie to HWC#	This will tie an interface component to another interface component given by it's “Hwc#” number which is the number found in the web interface. This feature is useful if you want a display to be linked to a button to display a label for it. But you can also copy functionality otherwise.
System Info	Displays system information, mainly status on connections.
No Action	No action. Will blank a display
Wait 1/10s second	Waits for a number of 1/10 second periods.
Custom Handler	Calls a custom handler (which need to be compiled into the firmware. Special feature).
Inactivate	Inactivates the panel until pressed again. In inactive mode, no keypresses, turned knobs or pulled handles will result in any action. All displays and button colors will remain active though.

Stop Connect	If pressed, any device which is not yet connected will be disabled (until reboot of the controller).

Device: ATEM Actions

This is a table of actions for BlackMagic Design ATEM Switchers.

Program Src 	Sets Program Source on the given M/E row. *Binary triggers:* Sets the selected source on Program. If Hold Down is selected, the source will fall back to the previous source whenever the trigger is released. Toggle will select the source, but on a subsequent trigger, it will fall back to the previous value. If Cycle mode is selected, a trigger will set the next source on Program (corresponds to a single pulse input). *Pulse inputs:* Will cycle through and set the possible sources for Program limited by the selected source and not including Black, unless Black is selected as source in which case all possible sources are traversed. *Binary outputs:* On when actual Program Src matches selected source (or when trigger is held in Cycle mode) *Button colors:* Will be red when Program Src matches selected source, otherwise dim. In Cycle mode color will be highlighted when button is held down.
Preview Src 	Sets Preview Source on the given M/E row. *Binary inputs:* Sets the select source on Preview. If Cycle mode is selected, a trigger will set the next source on Preview. *Pulse inputs:* Will cycle through and set the possible sources for Preview limited by the selected source and not including Black, unless Black is selected as source in which case all possible sources are traversed. *Binary outputs:* On when actual Preview Src matches selected source (or when trigger is held in Cycle mode) *Button colors:* Will be green when Program Src matches selected source, otherwise dim. In Cycle mode color will be highlighted when button is held down.
Prv/Prg Src 	Set Preview Source on the given M/E row and if the trigger is held down for more than 1 second, it will perform a Cut action too. *Binary inputs:* Sets the select source on Preview. If Cycle mode is selected, a trigger will set the next source on Preview (corresponds to a single pulse input) when released unless the button is held until a Cut is performed in which case no new Preview source is selected. *Pulse inputs:* Will cycle through and set the possible sources for Preview limited by the selected source and not including Black, unless Black is selected as source in which case all possible sources are traversed. *Binary outputs:* On when actual Preview source or Program source matches the selected source (or when trigger is held in Cycle mode) *Button colors:* Will be red or green when Program or Preview Src matches selected source, otherwise dim. In Cycle mode color will be highlighted when button is held

down. For mono-color buttons, the button will blink when the source is on preview (normally green on a multicolor button).

AUX Output Src

Set AUX source on the given AUX bus.

Binary inputs: Sets the select source on the AUX bus. If Hold Down is selected, the source will fall back to the previous source whenever the trigger is released. Toggle will select the source, but on a second trigger, it will fall back to the previous value. Hold Groups will fall back to a previous source for a group of triggers using a queue system and finally to the first previous value before any trigger in the group was activated. If Cycle mode is selected, a trigger will set the next source on the AUX bus (corresponds to a single pulse input).

Pulse inputs: Will cycle through and set the possible sources for AUX limited by the selected source and not including Black, unless Black is selected as source in which case all possible sources are traversed.

Binary outputs: On when actual AUX bus source matches selected source (or when trigger is held in Cycle mode)

Button colors: will be highlighted when AUX bus source matches selected source, otherwise dim. In Cycle mode color will be highlighted when button is held down.

Upstream Keyer

Turns upstream keyers on and off

Binary inputs: If Toggle mode, the given upstream keyer is turned or/off successively. If On or Off the upstream keyer is set On or Off respectively. Hold Down will turn the keyer on as long as the trigger is held. Auto will fade in the keyer (still pending as of June 2016)

Pulse inputs: Will turn on/off

Binary outputs: Follows highlighted button color.

Button colors: Will be highlighted if the keyers state corresponds to the selected mode. For most modes except ”Off” this means the color will be highlighted (but for Off a button is highlight in case the keyer is actually off)

Upstream Keyer Fill

Selects the fill source for Upstream Keyer

Binary inputs: Sets the selected source.

Pulse inputs: Cycles through the available sources. Press and hold will reset to the selected source.

Binary outputs: On if current keyer source is the selected source.

Button colors: Will be highlighted if current keyer source is the selected source.

Upstream Keyer Key

Selects the key source for Upstream Keyer

Binary inputs: Sets the selected source.

Pulse inputs: Cycles through the available sources. Press and hold will reset to the selected source.

Binary outputs: On if current keyer source is the selected source.

Button colors: Will be highlighted if current keyer source is the selected source.

Downstream Keyer

Turns downstream keyers on and off

Binary inputs: If Toggle mode, the given downstream keyer is turned or/off successively. If On or Off the downstream keyer is set On or Off respectively. Hold Down will turn the keyer on as long as the trigger is held. Auto will fade in the keyer.

Pulse inputs: Will turn on/off

Binary outputs: Follows highlighted button color

Button colors: Will be highlighted if the keyers state corresponds to the selected mode. For most modes except ”Off” this means the color will be highlighted (but for Off a button is highlight in case the keyer is actually off)

Downstream Keyer Fill

Selects the fill source for Downstream Keyer

Binary inputs: Sets the selected source.

Pulse inputs: Cycles through the available sources. Press and hold will reset to the selected source.

Binary outputs: On if current keyer source is the selected source.

Button colors: Will be highlighted if current keyer source is the selected source.

Downstream Keyer Key

Selects the key source for Downstream Keyer

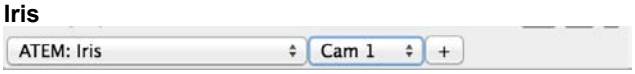
Binary inputs: Sets the selected source.

Pulse inputs: Cycles through the available sources. Press and hold will reset to the selected source.

Binary outputs: On if current keyer source is the selected source.

Button colors: Will be highlighted if current keyer source is the selected source.

Iris



Changes iris value for the selected camera.

Binary inputs: Will trigger auto iris
Pulse inputs: Changes the value up/down.
Analog inputs: Set the value between 0-100%
Displays: Will show the current value

Values:
- Select Camera 1-10. If you choose Mem A-D, the camera value will be taken from the value of this memory register.

About Audio, Video and Camera Sources

Whenever you can select audio, video and camera sources you will find special options in the drop down:

- Whenever you see ”Mem A”-”Mem D” it means the source selected will be the one from the list which the given memory register value currently points to, starting the counting from

zero. For example, if Mem A is 21, the source will be ”Bars” because it's element number 22 in the list (and the first element, ”Black”, has number 0).

- For video sources, selecting AUX1-6 means the source will be whatever source is currently on AUX1-6. This will be dynamically evaluated.
- For video sources, selecting MVx/y means the source will be whatever source is currently on the multiviewer ”x” (1 or 2) in window number ”y”. This will be dynamically evaluated.
- For camera sources, ”Mem A”-”Mem D” will not point to the list, but simply refer to the camera number.

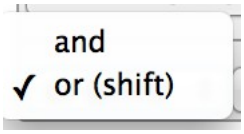
Multiple actions

You can assign multiple actions – even on different devices – to any interface component. This is done by simply pressing the ”+” button and setting up the new action:

In this case, the media player 1 (MP1) is brought on Preview on an ATEM switcher and right after the still number 5 is selected for the media player 1.

The return values, including those driving a display, will always come from the first action in the list (of the current shift level).

Notice that multiple actions are separated by an ”operator” which is either ”and” or ”or (shift)”:



This is explained in the following.

Shift

You can assign a button to set a shift state on your controller. Even though a shift state sounds like an either/or option, we have implemented the possibility to have multiple *shift-levels*. However, in the simple case, a shift button would be configured as shown below:

This will set the shift-level ”1” in the system as long as the button is held down, otherwise it will be ”0” (normal)

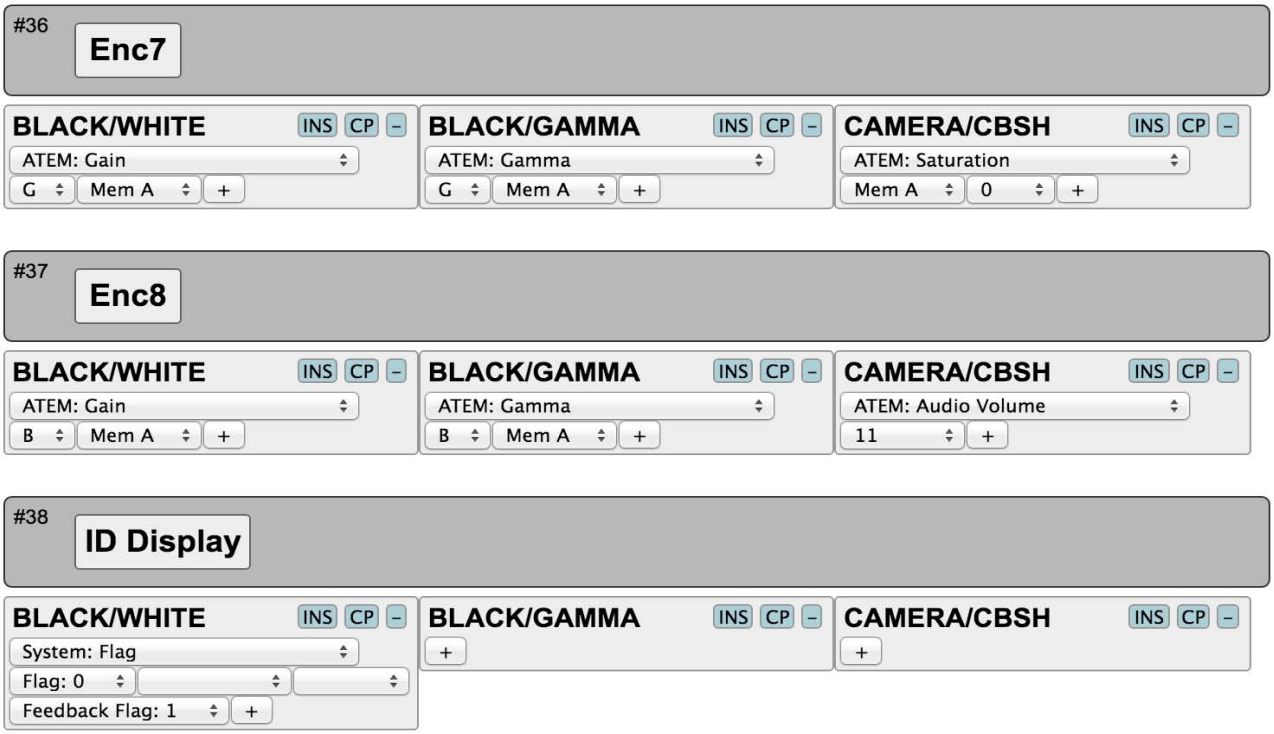
This means another button on the same controller could be configured like this now:

And because the divider between them is ”or (shift)”, the second action is only active when the shift button is held down.

If no specific action is defined for a shift level, the interface component will use the default list of actions.

States

Similar to shift levels you can put your controller in various states. States is mainly different from shift levels by the way the interface lets you set them up. With three states you see three columns of actions for each interface component:



Like with shift levels you can assign other interface components to change the state of the controller. If actions are not defined in any given state, they will fall back to the action list in the first column (Normal state).

States and shift levels can be combined of course; you can have unique shift levels inside each state.

States can also be named. In the above example, they are named ”BLACK/WHITE”, ”BLACK/GAMMA”, ”CAMERA/CBSH”. This is done in the controller web interface as well:

States

This is also where you select how many states the controller should support.

Copy / Paste

Often you will find yourself needing to set up almost the same function on multiple interface components (such as a row of buttons, all sending inputs to an AUX channel). To make this easy, make sure to use the Insert / Copy / Delete functions:

As soon as you make any change to a given interface components action list, this will be copied to memory so you just need to go to the next interface component and press ”INS” for insert.

Presets

At the bottom of the web interface you can load, save and reset your presets. Your controller can theoretically hold any number of presets only subject to the memory usage related to storing them. Pressing the save button in the web interface will save the configuration to the currently selected preset, but you can also select a new or different preset to save to using the selector box.

Load the ”(Default)” preset to get back to the factory configuration.

If your presets seems to be messed up for some reason (could be memory overflow which there is no protection against!) you may need to clear the entire memory by using the serial monitor command ”clearpresets” (see later).

Presets are a very powerful way to make use of your controller in multiple places since it can change the entire behavior of the controller including which devices to connect to and which IPs they are on.

There is a nifty way to load presets on most controllers if you have created more than 1 preset: When you boot your controller, you may see that a number of buttons (corresponding to the number of available presets) light up for a few seconds and one of them being highlighted. The highlighted button indicate the currently loaded preset and the other buttons represent other presets. If at this moment you press and hold any of the other buttons down until that button light up, you will then change the preset of the device (corresponding to selecting it in the web interface and press the ”Load” button).

Programming

(This section is still incomplete...)

Attach your SKAARHOJ controller to your computer via the Micro USB cable. You also need to supply power to the controller via PoE or the DC jack.

You need to have an FTDI driver (link) installed on your Mac/PC/Linux computer to access the controller.

Finally you need the SKAARHOJ Setup application (todo) (or the Arduino IDE environment and necessarily libraries and hardware profiles ready to make the installations, see separate section)

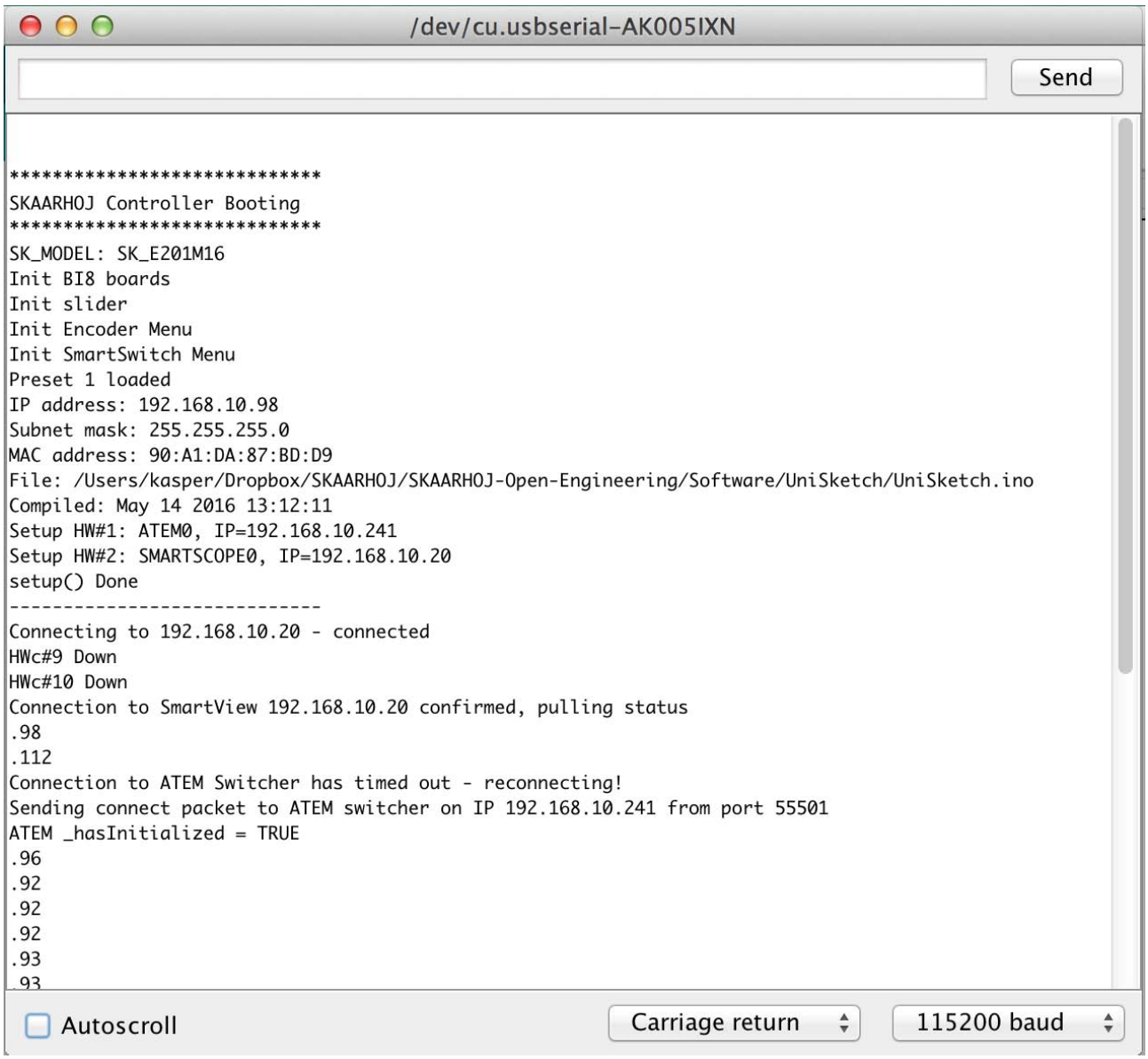
Arduino IDE

(This section is still incomplete...)

If you wish to program your … (explain libraries, serial monitor

Serial Monitor

The serial monitor is an indispensable tool for bugfixing any problems with your SKAARHOJ controller. The serial monitor runs at 115200 baud and a typical output from the boot process looks like this:



This tells us the model name of the controller, how a number of hardware components have been initialized, that preset 1 is loaded, which IP address, subnet mask and MAC address the controller has, which date the software was compiled.

It also shows us which hardware devices it will try to connect to, in this case an ATEM switcher and a SMARTSCOPE at 192.168.10.241 and 192.168.10.20 respectively.

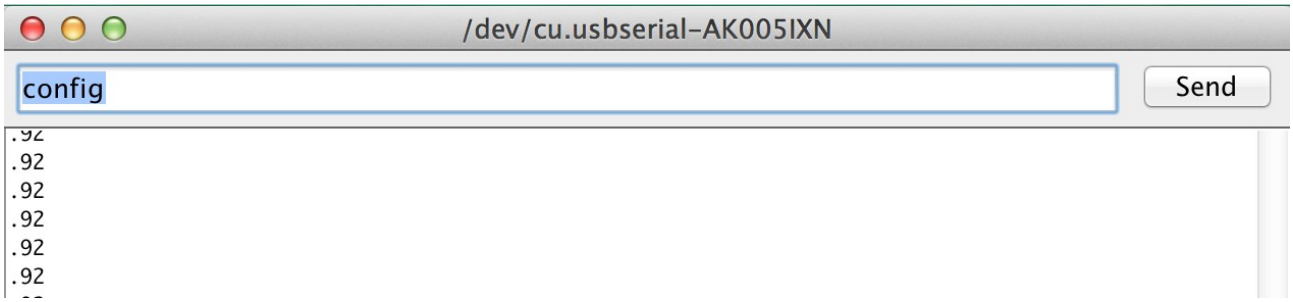
During this process until the ”setup() Done” message is output, the status LED will blink purple.

After the setup, the controller enters normal operational state. You see that it tries to connect to the devices and that it succeeds in this. During this process, the status LED blinks yellow and eventually it will blink green.

The serial monitor will continuously output a small dot and a number every second. If this is not the case permanently, it indicates a crash of the controller. The number indicates the number of times a second the controller manages to check all device connections and hardware components. It should be higher than 25. The higher the better. This number may/will drop if there are problems, if something slows down the controller, if devices are not connected properly or in the process of being connected, if a lot of displays needs to be updated etc. The lower this value, the less responsive the interface will feel. This value will also be impacted by the number and type of actions configured for interface components in the web interface. Network problems may also impact this value. If this value is too low, the controller may further loose connections to devices and may seem unresponsive to interface operations.

Commands in the serial monitor

You can enter commands in the serial monitor to do certain things with the controller. This is particularly useful for developers and also for bugfixing and calibration. Enable CR/LF on the serial monitor dropdown menu in order to send the commands.



List of commands:

”config”	Reboots the device into config mode with its current IP
”configd”	Reboots the device into config default mode (IP always 192.168.10.99)
”debug”	Reboots and enables debug output to serial monitor
”newmac”	Generates a new random MAC address to EEPROM. Power cycle both your controller and network switch after this operation. Useful if you have network problems.
”clearpresets”	Clears the preset memory completely (flushes all!). Useful/necessary after a firmware upgrade.
”reset”	Reboots the controller
”HWvar=XXX”	Set Hardware Variant (byte). This value shouldn't be changed by users. It's significance is to inform the UniSketch software about which hardware revision it's running on in order to take certain specifics into account. Bit 0: Determines model of status LED.
”calibrate”	Starts a calibration process for analog interface components. Instructions will be shown on the serial monitor.

Notice: Any operation from the serial monitor that reboots the controller, does so with a ”soft” reset and the Ethernet chip in the controller may still hold old settings. In most cases this is no problem, but at other times it may lead to strange behaviors and connection problems. In that case; power cycle the unit, press the reset button shortly or close down and reopen the serial monitor which will also act as a hardware reset.