

TCP Server Device Core

The TCP Server Device Core for Blue Pill, known as *core-skaarhoj-tcpserver*, enables users to connect one or multiple TCP clients to the device core, and read/write values to numerous internally stored data types. It's also an implementation of the data register section of the [TCP Server device core for UniSketch](#), mainly for legacy purposes. The key attraction for Blue Pill users is likely the significantly improved capacities:

- **Strings:** It supports the read/write operations for 200 strings, which also include titles for each.
- **String Matrix:** Users can read/write into a 20x100 string matrix.
- **Integers:** This system allows reading and writing of 200 32-bit signed integers, along with titles for each. Every integer can be set with distinct permitted value ranges, such as "-10 to 20".
- **Booleans:** It supports read/write operations for 200 boolean values, including titles for each.
- **Option Lists:** It allows the read/write of 200 option list values, with titles for each. Each option list can be set with an individual number of options, for example, "On" and "Off" or "Red", "Green", "Blue", "White", etc.
- **Timers:** It enables reading and writing of 200 boolean values, which automatically revert to false after a specific time period.
- **Images:** It lets users write 100 JPEG images and 100 PNG images into the device core (primarily intended for small images used for display graphics).
- **From UniSketch:** The device core implements Mem A-L, Flags 0-63, as well as Shift and State registers, mostly for legacy reasons.

The device core is generally multi-directional. Most values are set to read/write, and alterations from any client - whether the connected TCP clients or the device core clients - are propagated to other connected clients, ensuring synchronization at all times.

Use cases

The Device Core is an essential tool for ad hoc interactions with Reactor, which is the panel management application found on the majority of SKAARHOJ products. This utility is particularly beneficial for addressing custom integration needs. Below are a few examples:

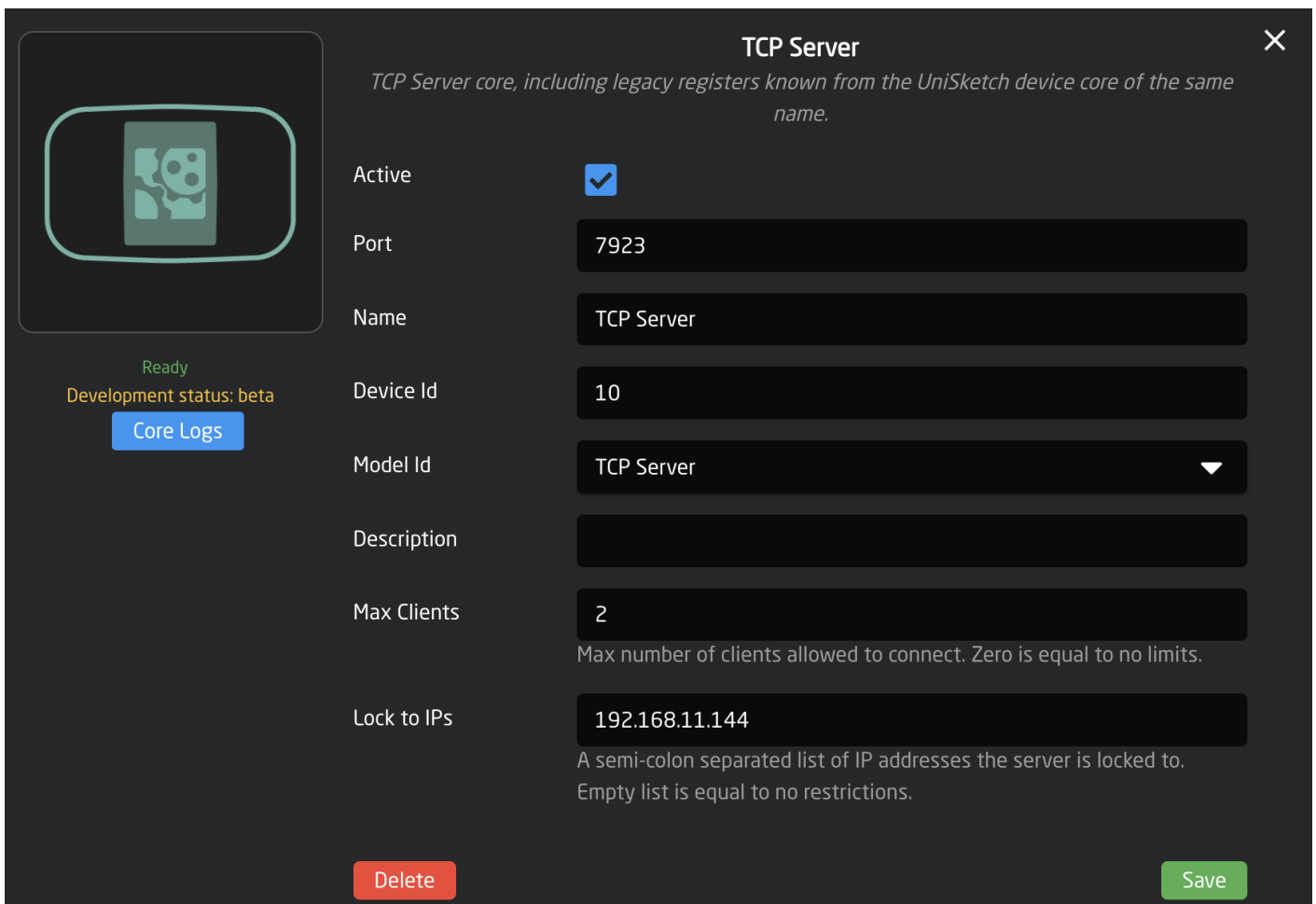
- Create ad hoc Device Core parameters using integers and option lists. Leverage value ranges and custom options to allow Reactor to control custom applications through TCP.
- Utilize Reactor's Virtual Triggers to initiate actions on broadcast devices based on strings, integers, booleans, and option list values that are altered via TCP.

- Use a timer to either send a cut trigger or initiate recording on a device for a specific duration.
- Regulate a device's volume using an integer within a predetermined range.
- Select sources using an integer.
- Enable or disable functions using a boolean or option list.
- Create dynamic labels and icons

<https://www.youtube.com/embed/SdICSrmS0rQ>

Device Core Setup

In Reactor you would add the device core to the device list. The setup page looks like this:



TCP Server ✕

TCP Server core, including legacy registers known from the UniSketch device core of the same name.

Ready
Development status: beta
Core Logs

Active	<input checked="" type="checkbox"/>
Port	7923
Name	TCP Server
Device Id	10
Model Id	TCP Server ▼
Description	
Max Clients	2 <small>Max number of clients allowed to connect. Zero is equal to no limits.</small>
Lock to IPs	192.168.11.144 <small>A semi-colon separated list of IP addresses the server is locked to. Empty list is equal to no restrictions.</small>

Delete Save

- Port is the TCP port where the TCP server is created. If you create multiple servers, make sure these ports are different and not used by anything else. Please check the logs of core-skaarhoj-tcpserver under the Packages tab if you experience problems: there will be information about any issues in the logs.

- The Device ID is usually "1" and is used to reference the parameters inside the core.
- Max Clients is the maximum number of clients that can connect at the same time. A light weight security measure.
- Lock to IPs are specific IP addresses which are the only ones allowed to connect. A light weight security measure.

Connect as a TCP Client

Using Putty, telnet or nc depending on your platform, you connect to the TCP server by using the IP address of the Blue Pill device hosting the device core and the port number. Here is an example. Connection is silent, there will be no introduction message sent to the connecting client, but typing in "help" and ending with newline (\n or char 10) will show available commands:

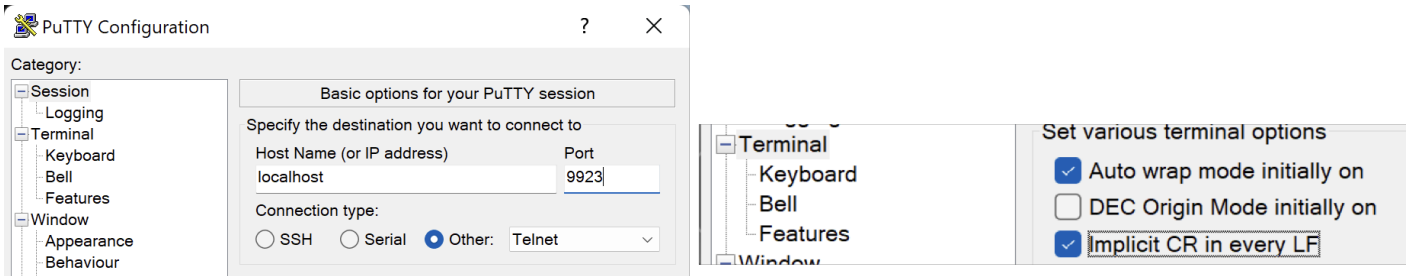
The screenshot shows a terminal window titled "kasper — nc 192.168.11.5 7923 — nc — nc 192.168.11.5 7923 — 103x28". The prompt is "[~ > nc 192.168.11.5 7923" and the user has entered "help". The output is a list of commands and their descriptions:

```

[~ > nc 192.168.11.5 7923
help
List of commands:
- help : This message
- Mem[A-L]=y : Set Memory value (byte)
- Flag#[0-63]=[1/0] : Set Flag bit
- Shift[/A/B/C/D]=y : Set Shift register value (byte)
- State[/P/Q/R/S]=y : Set State register value [0-9] (byte)
- Str#[1-200]=str : Set String register value
- StrTitle#[1-200]=str : Set title for string register value
- StrMatrix#[1-20].[1-100]=str : Set String Matrix register value
- Int#[1-200]=y : Set Integer register value
- IntRange#[1-200]=min:max : Set Integer range values (min < max)
- IntTitle#[1-200]=str : Set title for integer
- Bool#[1-200]=[1/0] : Set Boolean register value
- BoolTitle#[1-200]=str : Set title for boolean
- Opt#[1-200]=y : Set Option register value (10 values by default, 0-9)
- OptTitle#[1-200]=str : Set title for option list
- OptLabels#[1-200]=str;...;str : Set dynamic option labels (values follow, starting at 0)
- Timer#[1-200]=y : Set/Refresh timer for y milliseconds
- ImageJPEG#[1-100]=(base64 encoded jpeg < 10kb) : Set JPEG image
- ImagePNG#[1-100]=(base64 encoded png < 10kb) : Set PNG image
- HeartBeatTimer=[ms] : Sets frequency (>200 or zero for stop) by which the server will send the client a "ping"
- ping : Responds with ACK
- reset : Resets all memory
- list : Dump state

```

If you are using Putty on Windows, make sure to set it up correctly to send newlines for line endings:

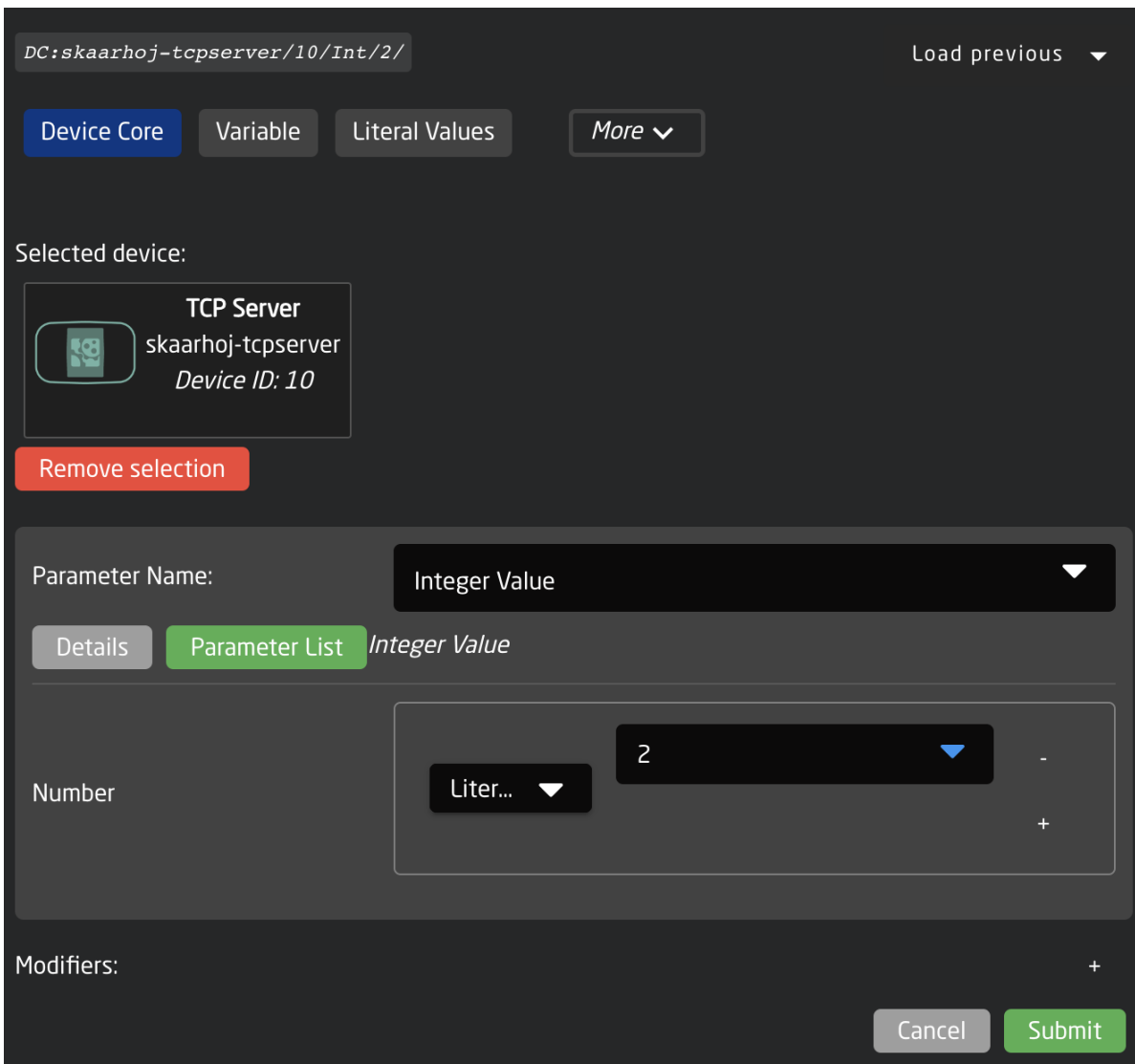


Try the different commands to get the feel for it.

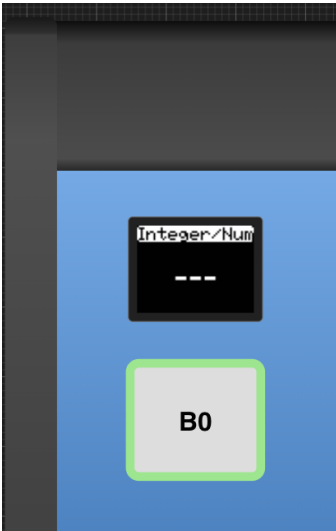
Example in Reactor

Here is an example of how Reactor can interface with an integer from the device core:

Click a hardware component in Configuration tab and assign a new behavior with an IO reference pointing to the Integer Value parameter of the TCP Server. Select integer #2 for example:



You will see it in the display of the component, but it has no value:



Probably the "Change by Step" master behavior was assigned to the button as you selected the parameter, so if you press the button it should assume a value:



Simultaneously, you should see values reported in the connected TCP Client window:

```
Int#2=1
Int#2=2
Int#2=3
Int#2=4
█
```

In the TCP Client you can try to type in "Int#2=111" and you should see the value reported back in all connected TCP Clients as well as update in Reactor:



The same can be done with all the other registers.

Revision #5

Created 7 June 2023 19:05:08 by Kasper

Updated 23 June 2026 08:05:48 by Kasper