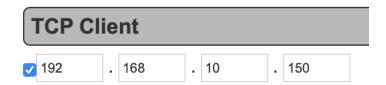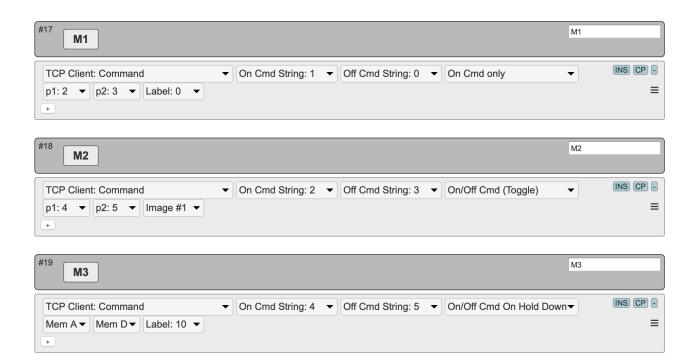# Generic TCP Client

This device core will let the panel connect as a TCP Client to a given IP address on a given port.

The IP to connect to is set via the standard device core configuration:

For example:

There is one action to add to buttons, "Command". Below are three examples of configuration:

"On Cmd only": This means when the button is triggered on down press, the command string with index 1 ("On Cmd String: 1") will be sent to the server.

"On/Off Cmd (Toggle)": This means when the button is pressed repeatedly, the action will alter between sending the "On Cmd String" and the "Off Cmd String". The button will light up after sending the On Smd String and be dimmed after sending the Off Cmd String
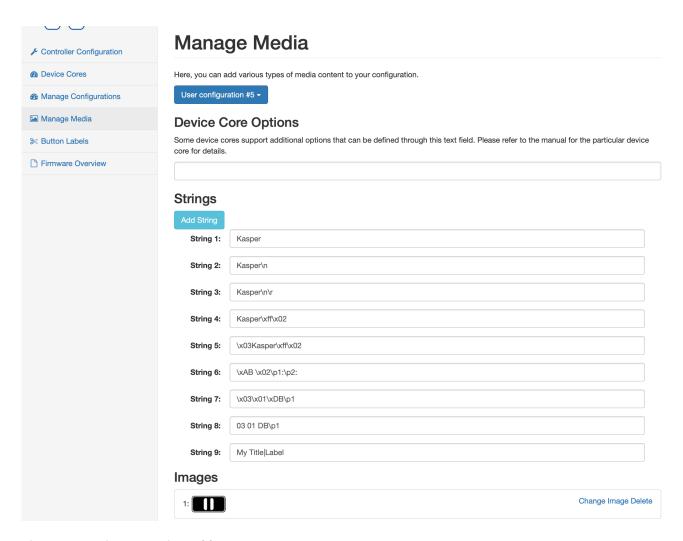
"On/Off Cmd On Hold Down": This means the "On Cmd String" is sent when button is pressed and the "Off Cmd String" is sent when the button is released again.

The "Label" selector (last parameter) will select a custom label for the button if it has a display (see for example how this works for System: Memory or similar actions.)

The p1 and p2 selectors are parameters. This can either be a fixed value from 0-200 or it can be dynamic, retrieved from Mem A-D. The two parameters can be incorporated into the command strings via codes. Read on…

On Cmd String and **Off Cmd String:**

When different from 0 (zero), this is a reference to a media string compiled into the controller from the "Manage Media" tab on the online configuration:

Manage Media

Here, you can add various types of media content to your configuration.

[User configuration #5 ▾]

Device Core Options

Some device cores support additional options that can be defined through this text field. Please refer to the manual for the particular device core for details.

[   ]

Strings

[Add String]

String 1:	Kasper
String 2:	Kasper\n
String 3:	Kasper\n\r
String 4:	Kasper\xff\x02
String 5:	\x03Kasper\xff\x02
String 6:	\xAB \x02\p1:\p2:
String 7:	\x03\x01\xDB\p1
String 8:	03 01 DB\p1
String 9:	My Title|Label

Images

1: ▮▮   Change Image   Delete

These strings has a number of features:

⁻ By default they are ASCII and exactly what you see in the string will be sent as ASCII, unless:

⁻ \n and \r will send a newline or carriage return (10 and 13)

⁻ \xHH will send a byte with the value HH (in hex) so \xFF will send a byte with value "255". \x00 will however terminate the string, so don't do that. You can't send the byte "zero" at all.

⁻ \p1 or \p2 (alternatively \d1, \d2, \h1, \h2) will insert a parameter value as defined by the selectors p1 and p2 in the action. The difference of using \p, \d and \h is whether the value is inserted as the byte value (\p - in this case, don't insert zeroes - it will terminate the string!), inserted as a decimal number like "255" or "37" or "3" (\d - there is no leading zeros) or in hexadecimal like FF or D0 or 00 (\h - in this case it's always two characters and uppercase).

Example

Say you want to add a command to fire DMEMs on GVG100, in that case you want to send the text string "03 01 DB 00" to fire DMEM 01, send "03 01 DB 01" to fire DMEM 02 etc.

So adding a string like "03 01 DB \h1" and then using the "p1" selector for the action assigned to multiple keys referring to this string would insert a dynamic value for the DMEM identification.

Test:

For testing the output from the TCP Client, we suggest you use the Python Script "TCPserver.py" on https://github.com/SKAARHOJ/Support/tree/master/Files/UniSketchTCPClient

Running that on a computer will open port 23 and you can let your SKAARHOJ panel connect to it and see what it sends over.

Change of port:

Can be done with device core option 0 (zero): Insert "D0:0=1234" to set the port to 1234 in case the device core is the first one installed ("DC0"). Default port is 23.