

Server Script

Funkksteckdosen eintragen

```
# -----  
# use this section to set the parameters for your setup  
# -----  
  
# dictionary with for send command: [group code] [device]  
light_dict = {  
    "Wohnz"      : "11111 5",  
    "Flur1"      : "11101 6",  
    "Garage"     : "10010 3",  
    "Black"      : "00101 5",  
    "Wohnzimmer" : "10011 1",  
    "Flur"       : "10011 2",  
    "Küche"      : "10011 3",  
    "Brunnen"    : "10011 4",  
    "Fernseher"  : "10011 5",  
    "Kinderzimmer" : "10001 1"  
}
```

1-wire Sensoren eintragen

```
# dictionary with for 1-wire sensors: [sensor name] [1-Wire device]  
sensor_dict = {  
    "Wohnzimmer" : "28-00000487bb70",  
    "Balkon"     : "28-00000487bb70",  
    "Speicher"   : "28-000004be39a5"  
}
```

Server starten

- netio_server.py # server wird mit logout beendet
- netio_server.py & # server läuft im Hintergrund

NetIO UI Designer

Globale Einstellungen für das Projekt im Net IO Designer

Global

Config Name	<input type="text" value="433 Client"/>
Description	<div>Demo for 433 Server incl. Timer and 1-Wire Sensor</div>
Device	<input type="text" value="iPad"/>
Orientation	<div>Portrait ▾</div>
Navigation	<div>Fix ▾</div>
Pagebutton Width	<div>Static ▾</div>

Connections [Connection Documentation](#)

Add Attribute ▾

[Remove Connection](#)

name	<input type="text" value="Rasp IP BB"/>
host	<input type="text" value="192.168.2.112"/>
port	<input type="text" value="54321"/>

Eine Seite für den Client definieren:

Bei connection wird der Name eingetragen, der in der Projekt-Einstellung davor definiert wurde.

General

Pages

Selection



Selected Page

Add Attribute▼

name Black Beast

label Black Beast

connection Rasp IP BB

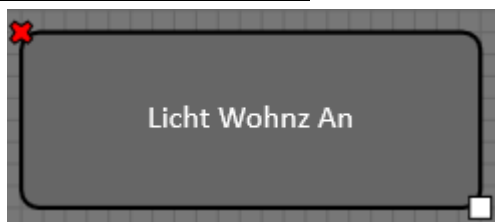
Global Page List

#	Name	Action
---	------	--------

0	Black Beast	
---	-------------	--



Einfacher Ein-Schalter



Button

[Documentation](#)

Add Attribute▼

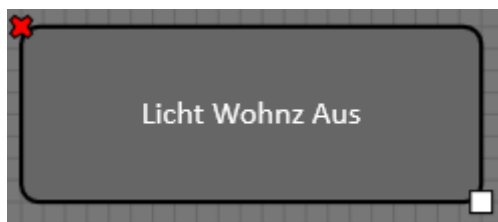
shape rounded ▼

label Licht Wohnz An

sends ⬆ Licht Wohnz An X

[add another 'sends'-value](#)

Einfacher Aus-Schalter



Button

[Documentation](#)

Add Attribute▼

shape rounded ▼

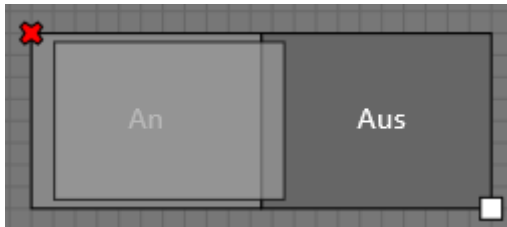
label Licht Wohnz Aus

sends ⬆ Licht Wohnz Aus X

[add another 'sends'-value](#)

Schiebe-Schalter

Dieser Schalter wird entsprechend dem definierten Intervall aktualisiert.
D.h. wenn ein anderer NetIO client die Steckdose schaltet wird der Schalter seinen Zustand ändern.



Switch

[Documentation](#)

Add Attribute▼

onValue

onText

offText

onSend

offSend

reads

interval

parseResponse

requiresSendReport ☒

Temperatur-Anzeige

Der Sensor „Speicher“ wird im netio_server.py eingetragen: `sensor_dict`
Der 1-wire Sensor wird im definierten Intervall ausgelesen.



Label

[Documentation](#)

Add Attribute ▾

text Temp Speicher

reads Temp Speicher

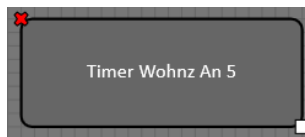
interval 10000

parseResponse \d+

formatResponse {0},{1}°C

Timer

Der Timer spricht eine Steckdose an, die im light_dict definiert wurde.



Button

[Documentation](#)

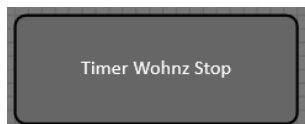
Add Attribute▼

shape rounded ▼

label Timer Wohnz An 5

sends ↑ Timer Wohnz An X

[add another 'sends'-value](#)



Button

[Documentation](#)

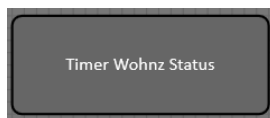
Add Attribute▼

shape rounded ▼

label Timer Wohnz Stop

sends ↑ Timer Wohnz Sto X

[add another 'sends'-value](#)



Button

[Documentation](#)

Add Attribute▼

shape rounded ▼

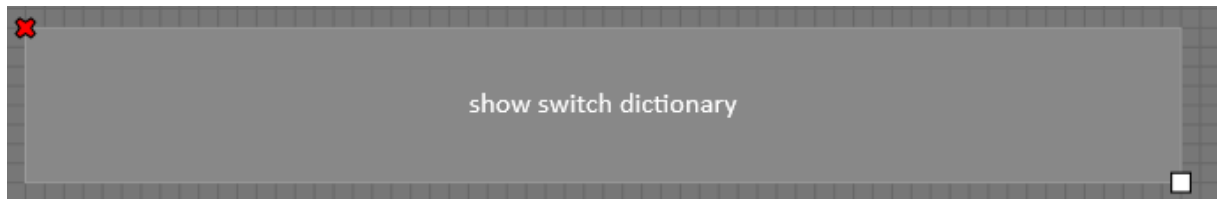
label Timer Wohnz Status

sends ↑ Timer Wohnz Sta X

[add another 'sends'-value](#)

interval 2000

Debug - Anzeige



Label

[Documentation](#)

Add Attribute▼

text show switch dictionar

reads dict

interval 2000

text Tag

fontsize 24

reads Wetter Wiehl Tag

background 77,77,77

parseResponse \w+

formatResponse {1}

text Temp max

fontsize 24

reads Wetter Wiehl tempma

background 0,0,0

parseResponse \d+

formatResponse {1}°C

text Temp min

fontsize 14

reads Wetter Wiehl tempmi

background 0,0,0

parseResponse \d+

formatResponse {1}°C



text Image

reads Wetter Wiehl image

background 77,77,77

parseResponse \w+

formatResponse {1}.png

text Himmel

reads Wetter Wiehl Himmel

background 77,77,77

parseResponse \w+

formatResponse {1}

text Wind

reads Wetter Wiehl Wind

background 43,172,160

parseResponse \w+

formatResponse {1}.png

text Wind

fontsize 16

reads Wetter Wiehl Wind

background 43,172,160

parseResponse \w+

formatResponse {1}

text Windr

fontsize 10

reads Wetter Wiehl Windr

background 43,172,160

parseResponse \d+

formatResponse {1}°

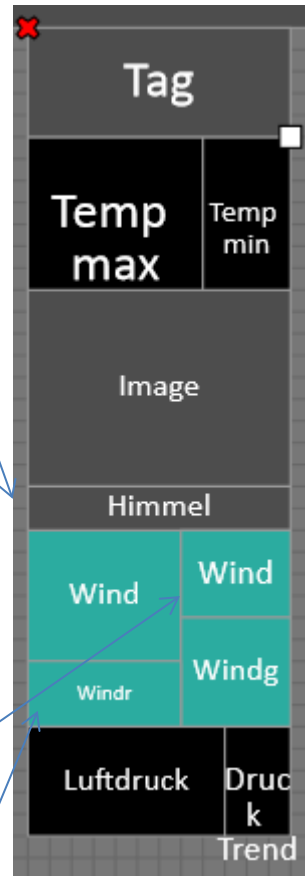
text Windg

reads Wetter Wiehl Windg

background 43,172,160

parseResponse \d+.\d

formatResponse {1} m/s



text

reads

background

parseResponse

formatResponse

text

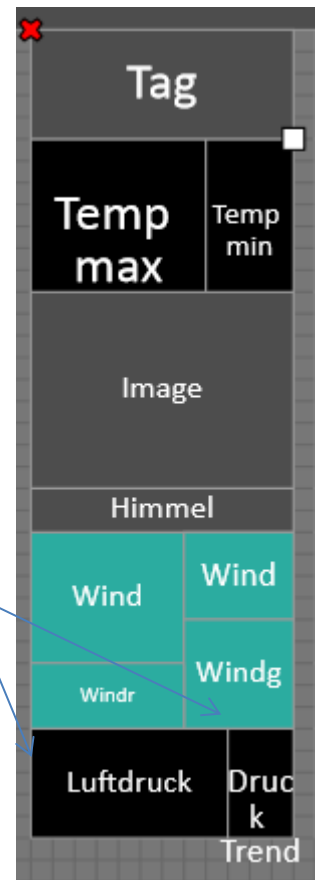
fontsize

reads

background

parseResponse

formatResponse



text

fontsize

reads

interval

background

text

reads

interval

background

