

# Webová vizualizace pro systém REXYGEN

## Uživatelská příručka

REX Controls s.r.o.

Verze 3.0.1  
1.3.2024  
Plzeň

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>WebWatch</b>	<b>6</b>
2.1	Automatické generování HMI z programu REXYGEN Studio	6
2.2	Pokročilé použití	6
<b>3</b>	<b>REXYGEN HMI Designer</b>	<b>9</b>
3.1	Grafický designer webového HMI	9
3.2	Obecná konfigurace HMI	10
3.2.1	Záložka Připojení (Connections tab)	11
3.2.2	Záložka Možnosti (Options tab)	13
3.2.3	Záložka Export HTML (HTML Export tab)	13
3.3	Rozšíření pro REXYGEN HMI Designer	14
3.3.1	Animace skupin	14
3.3.2	Aktualizovat komponenty	15
3.4	Seznam dostupných komponent	15
3.4.1	HMIConfig	15
3.4.2	CHARTS	16
3.4.3	GENERAL	19
3.4.4	HOME	44
3.4.5	HVAC	51
3.4.6	PROC	61
<b>4</b>	<b>WebBuDi</b>	<b>67</b>
4.1	Jednoduchá tlačítka a displeje na webu	67
4.2	Dostupné řádky a komponenty	70
4.2.1	AnalogLookupTable	71
4.2.2	AnalogReadRow	71
4.2.3	AnalogWriteRow	72
4.2.4	DigitalReadRow	73
4.2.5	DigitalWriteRow	73
4.2.6	EmptySpace	74
4.2.7	LinkButton	74

4.2.8	ManualPulseRow . . . . .	74
4.2.9	PushButton . . . . .	75
<b>5</b>	<b>Knihovna REX.HMI</b>	<b>76</b>
5.1	Jak používat knihovnu . . . . .	76
5.2	Referenční příručka pro REX.HMI . . . . .	76
5.3	Referenční příručka pro REX.HMI.Graph . . . . .	84
	<b>Literatura</b>	<b>87</b>

# Kapitola 1

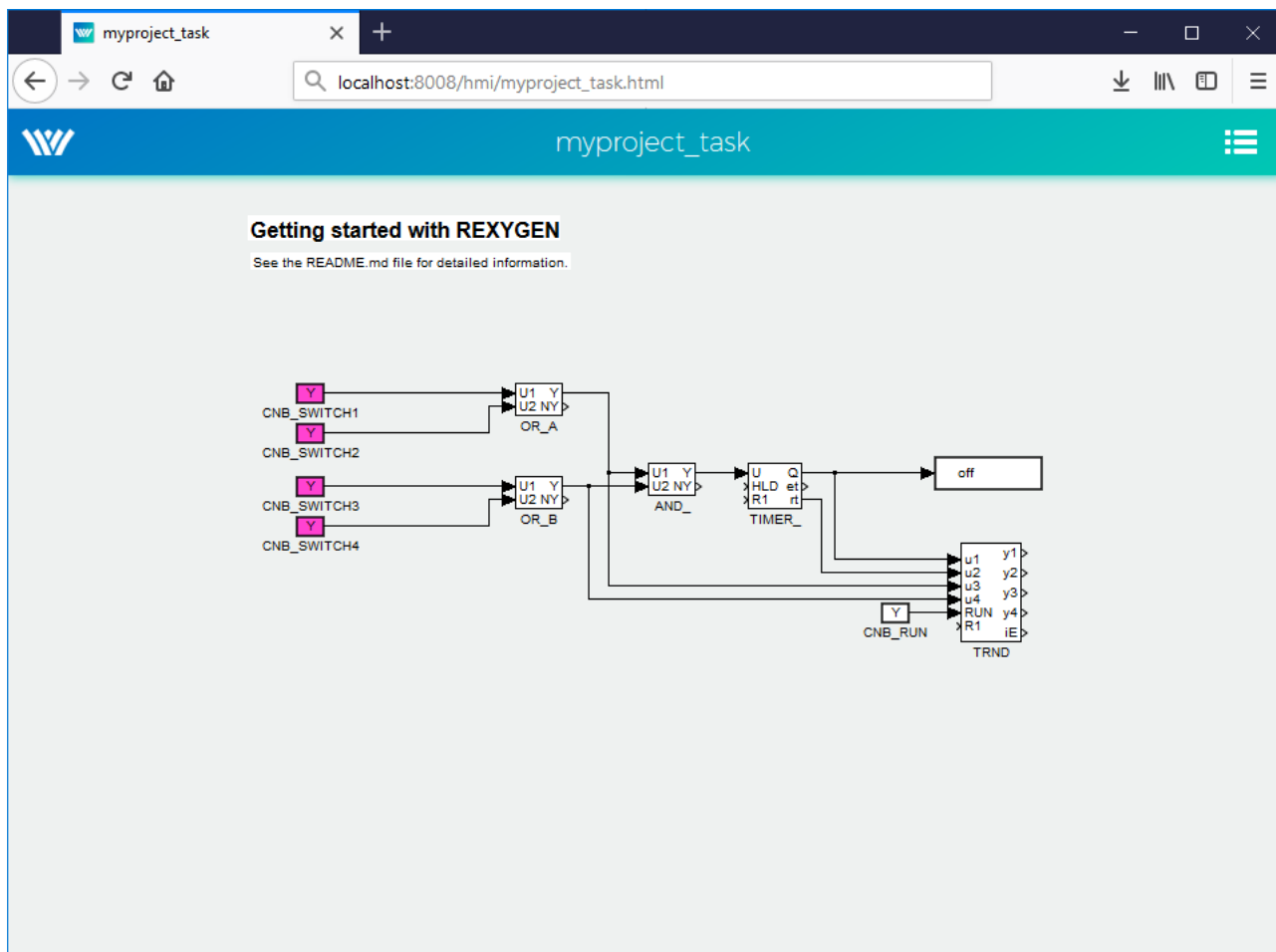
## Úvod

RexHMI je souhrnný název pro nástroje jejichž pomocí lze vytvářet webové vizualizace pro systém REXYGEN. V tuto chvíli existují tři možnosti jak si vytvořit vlastní vizualizaci - WebWatch, WebBuDi a REXYGEN HMI Designer.

- WebWatch(Kapitola 2) je určen k automatickému generování vizualizací z řídicího schématu. Má podobný vzhled, atributy a funkce jako mód sledování v programu REXYGEN Studio. WebWatch je perfektní nástroj pro okamžité vytvoření HMI vhodného pro vývojáře nebo integrátory systému. WebWatch poskytuje grafickou interakci s téměř všemi signály v řídicím algoritmu.
- REXYGEN HMI Designer(Kapitola 3) vytváří standardní SVG soubor s rozšířeními pro *RexHMI*. REXYGEN HMI Designer je skvělý nástroj pro vytváření grafického HMI vhodného pro operátory a jiné koncové uživatele.
- WebBuDi(Kapitola 4, zkratka pro Web Buttons and Displays) je jednoduchý JavaScriptový soubor s několika deklarativními bloky, které popisují datové body, ke kterým je HMI připojeno. Bloky se skládají do tabulky, ve které jsou všechna data prezentována. Poskytuje textovou interakci se zvolenými signály a je vhodným nástrojem pro vývojáře nebo integrátory systému, případně může sloužit jako HMI v nouzovém režimu pro nestandardní situace.

HMI je obvykle na cílové zařízení staženo prostřednictvím bloku *HMI* v souboru *exec.mdl*. V adresáři projektu vytvořte soubor *exec.mdl* (nebo použijte předdefinované šablony ze start-up menu). Následně do exekutivy přidejte blok *HMI*, nastavte parametr *IncludeHMI* a poté povolte vizualizaci *Web Watch* (parametr *GenerateWeb Watch*), případně si vytvořte vlastní. Jakmile je vše nakonfigurováno, vizualizaci lze stáhnout do cílového zařízení prostřednictvím funkce *Compile and Download* v programu REXYGEN Studio. HMI je přístupné z interního serveru cílového zařízení, například na URL adrese <http://192.168.1.100:8008/hmi>.

URL adresa může obsahovat doplňující parametry ovlivňující chování HMI:

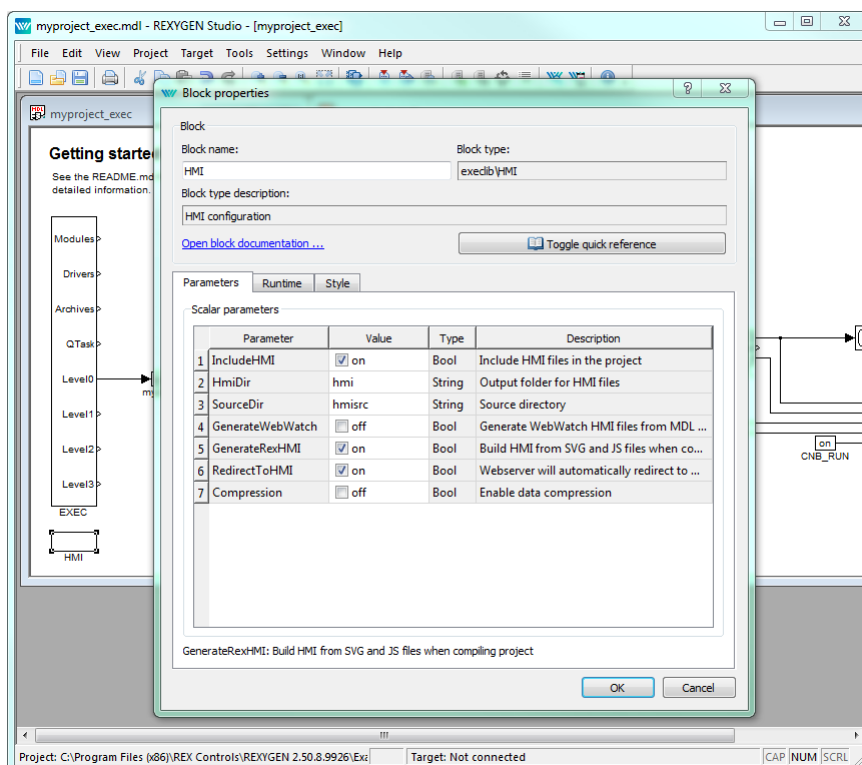


Obrázek 1.1: Příklad vizualizace ve *Web Watch*

- virtualKeyboard - aktivace virtuální klávesnice na zařízeních s dotykovou obrazovkou. Používá se typicky u panelových počítačů s operačním systémem Linux, kde samotný operační systém toto neumožňuje nativně.
- noLogout - skrytí odkazu *Logout*. Používá se typicky u zařízení pracujících v kioskovém režimu.

Příklad: <http://192.168.1.100:8008/hmi/index.html?virtualKeyboard=true&noLogout=true>

Pro speciální případy může uživatel vytvořit vlastní HMI založené na JavaScriptu a vlastní HTML nebo SVG komponenty. Pro tento účel je k dispozici knihovna funkcí *REX.HMI*. Kapitola 5 popisuje všechny funkce a jejich parametry.



Obrázek 1.2: Konfigurace bloku *HMI* v programu REXYGEN Studio

## Kapitola 2

# WebWatch

### 2.1 Automatické generování HMI z programu REXYGEN Studio

WebWatch je automaticky vygenerované HMI založené na struktuře projektu a je podobné *Watch mode* v programu REXYGEN Studio. Celé schéma je vygenerováno do webové stránky. Uživatel může monitorovat všechny signály ze zvolených bloků, měnit parametry bloků a číst data z bloků TRND. WebWatch je generováno automaticky použitím bloku *HMI* v hlavním souboru projektu.

1. Vložte blok HMI do souboru *exec.mdl* vašeho projektu.
2. Zaškrtněte `GenerateWebWatch` a `IncludeHMI` pro aktivaci generování WebWatch.
3. Spusťte *Compile and Download*.
4. Otevřete webový prohlížeč a jděte na adresu cílového zařízení, např. <http://192.168.1.100:8008/hmi>.
5. *Klikněte levým tlačítkem myši (dotek)* pro změnu parametrů bloku a pro povolení monitorování bloku použijte *stisknutí pravého tlačítka myši (dlouhý dotek)*.

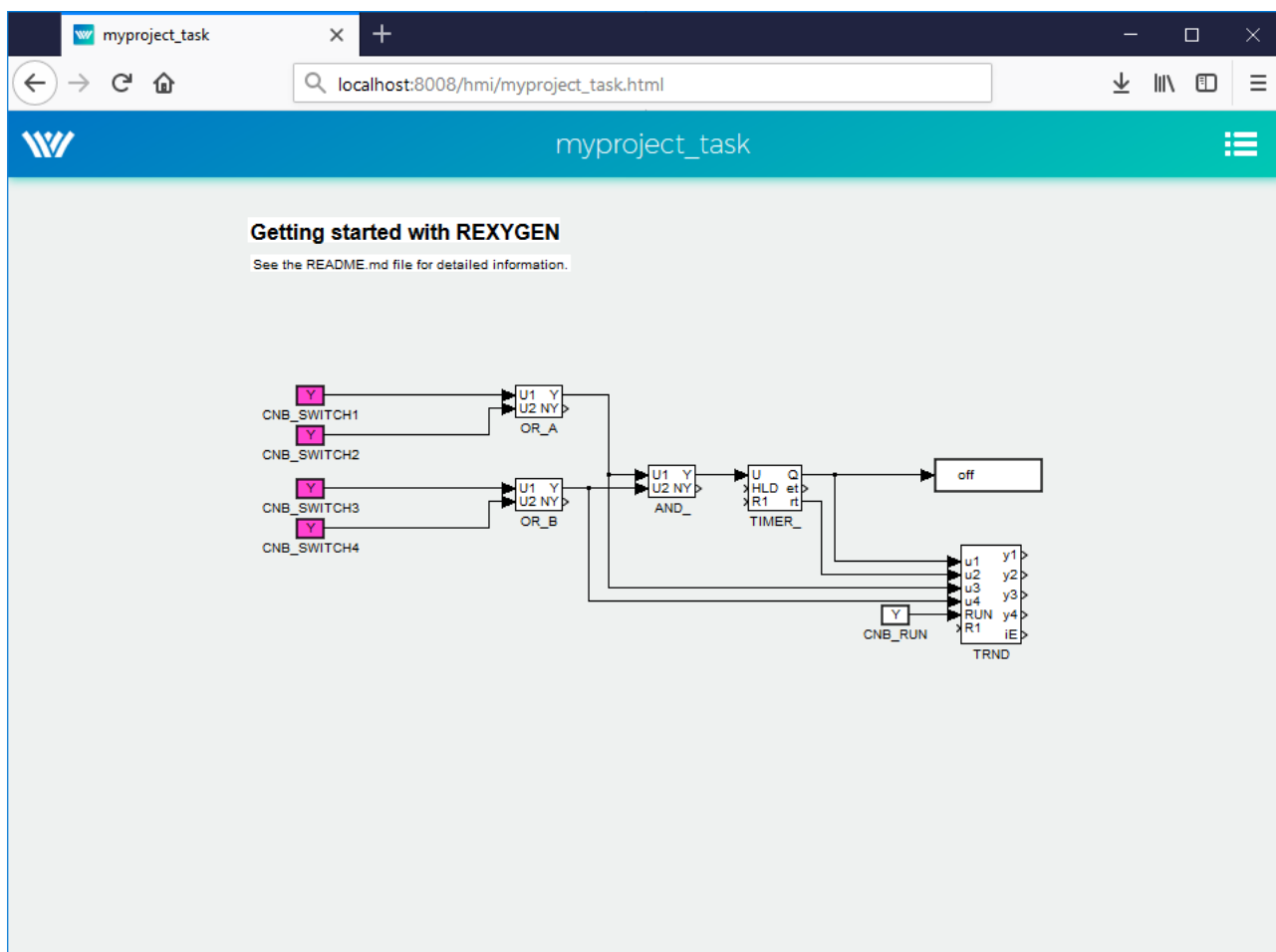
### 2.2 Pokročilé použití

Automaticky vygenerované schéma lze přizpůsobit manuálně. <sup>1</sup>

Upravte `REX.HMI.init = function(){}` funkci ve zvoleném `*.html` souboru. Pro WebWatch HMI existuje pouze několik funkcí, ale můžete používat všechny funkce popsané v kapitole 5.

---

<sup>1</sup>Poznámka: Pokud je *GenerateWebWatch* v bloku HMI aktivováno, HMI je generováno při každé kompilaci schématu. Pro manuální přizpůsobení deaktivujte možnost *GenerateWebWatch* v bloku HMI.



Obrázek 2.1: Příklad vizualizace WebWatch

**REX.WebWatch.enableMonitoring(blockIDs)** – Aktivace monitorování zvolených bloků okamžitě od načtení webové stránky.

Parametr	Typ	Popis
blockIDs	Array	Seznam všech bloků, popsáný spojovacím řetězcem .<String> (např.["task.block1", "task.block2"])

#### Příklad

```

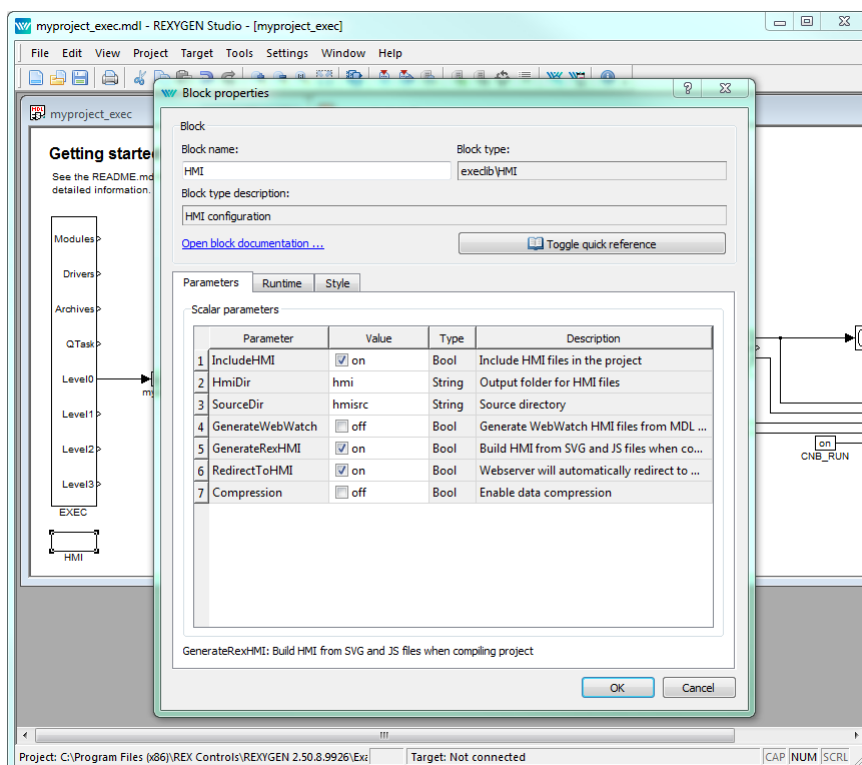
1 // Aktivace monitorování zvolených bloků
2 REX.HMI.init = function(){
3   REX.WebWatch.enableMonitoring(["pidcontrol_control.CNR_sp",
4     pidcontrol_control.PIDU"]);
5 }

```

**REX.WebWatch.disableHint()** – Deaktivuje nápovědu po načtení stránky

**REX.WebWatch.showHint()** – Zobrazí nápovědu





Obrázek 2.2: Zaškrtněte GenerateWebWatch pro vytvoření WebWatch HMI.

## Kapitola 3

# REXYGEN HMI Designer

### 3.1 Grafický designer webového HMI

REXYGEN HMI Designer<sup>1</sup> je nástroj pro designování vlastních vizualizací za použití předdefinovaných komponent. Celé HMI je uloženo v souboru SVG (Scalable Vector Graphics) se specifickými rozšířeními REXYGEN.

Když je připravena vizualizace, vygeneruje se HTML stránka s veškerými potřebnými knihovnamí. Takovou webovou stránku lze snadno vytvořit a stáhnout na cílové zařízení z adresáře `./hmi` pomocí funkce *Compile and Download* vývojového nástroje REXYGEN Studio.

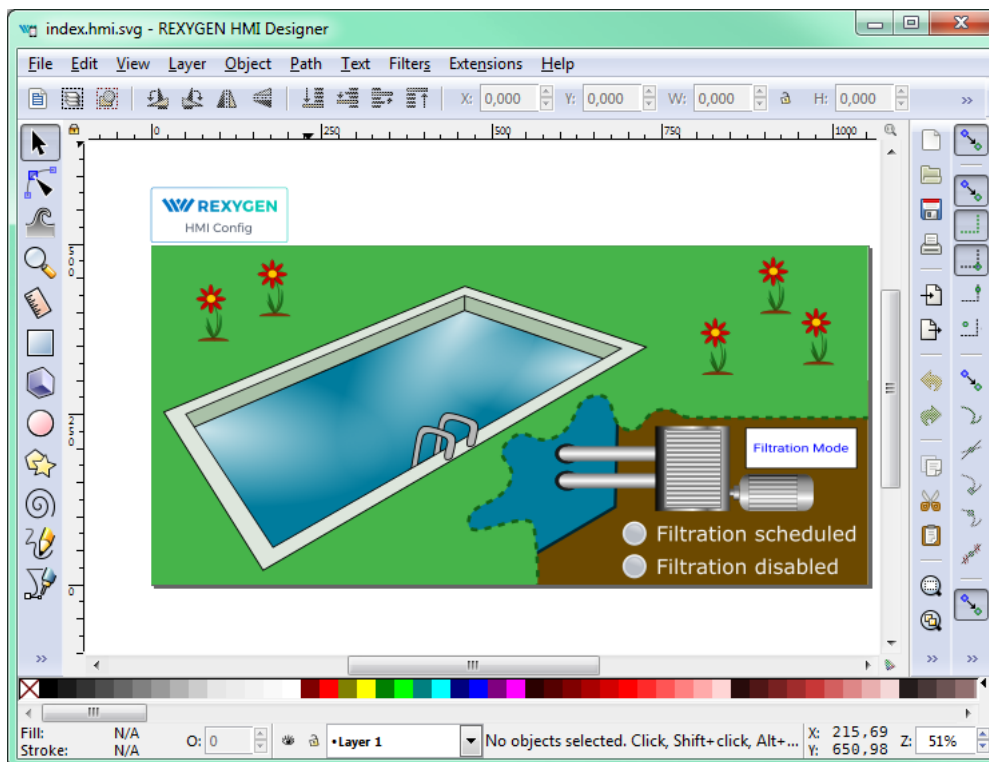
Každé HMI schéma se skládá z několika komponent, které jsou propojeny se signály z řídicího systému. Tyto komponenty jsou organizovány v knihovnách. Příklad HMI na obr. 3.1 obsahuje jedno tlačítko (*Button*), dva LED displeje (*LEDs*) a několik obecných komponent (*GeneralComponents*) pro animace bazénu a květin.

HMI komponenty jsou ovládány prostřednictvím rozšíření (viz obr. 3.2), která upravují specifické parametry pro REXYGEN HMI a umožňují export finálního HMI. K dispozici jsou následující rozšíření:

- **Procházet knihovnu komponent** (Ctrl + L) - Otevře složku se všemi dostupnými komponentami. Každá komponenta je uložena jako soubor SVG, který lze přetáhnout do aktuálního HMI.
- **Upravit komponentu** (Ctrl + E) - Otevře okno pro úpravu vybrané komponenty nebo celého projektu, pokud není vybrána žádná komponenta.
- **Upravit konfiguraci HMI** - Otevře okno pro úpravu obecné konfigurace HMI.
- **Exportovat HMI** (Ctrl + H) - Exportuje obsah souboru SVG do webové stránky HTML5 s veškerými potřebnými knihovnamí. Toto rozšíření se konfiguruje prostřednictvím parametrů celého projektu.

---

<sup>1</sup>REXYGEN HMI Designer je postavený na známém SVG editoru Inkscape<sup>TM</sup> <https://inkscape.org/en/>.

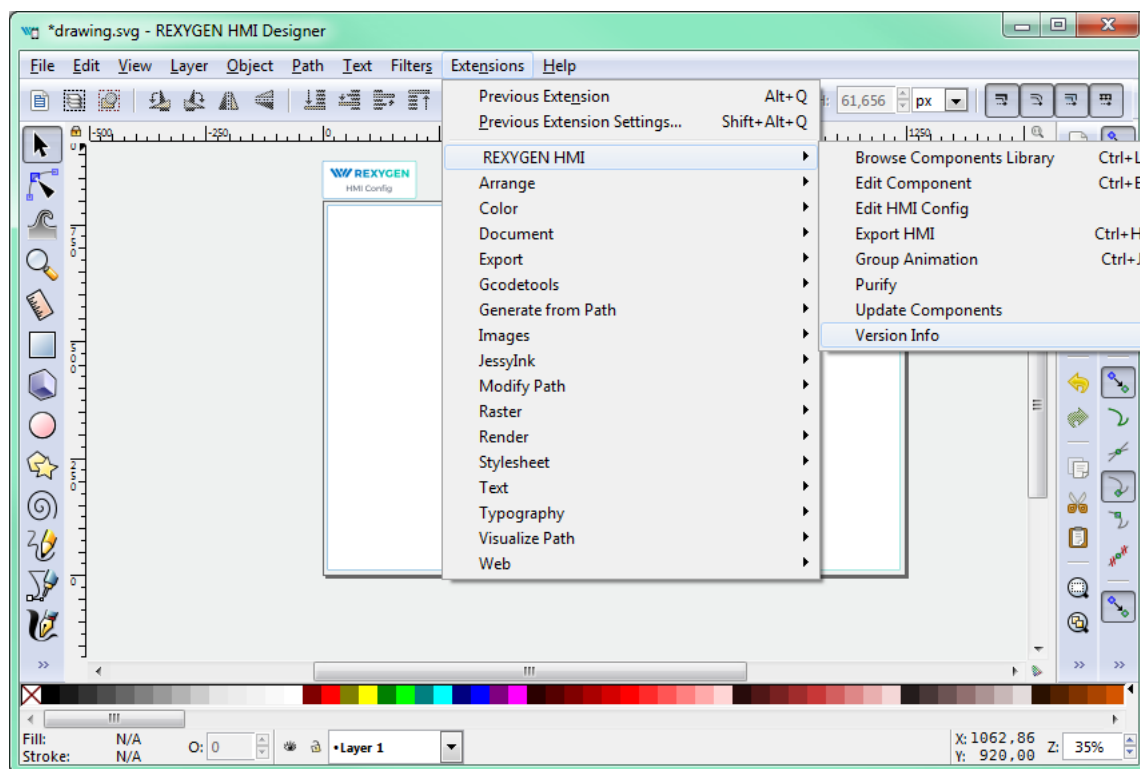


Obrázek 3.1: Příklad vizualizace REXYGEN HMI Designer

- **Seskupit animaci** (Ctrl + J) - Povolí animaci transformací (posun, rotace, změna velikosti), průhlednosti nebo barvy vybraného objektu (skupiny SVG) na základě hodnot z řídicího systému. Lze také použít rozšíření pro vytvoření klikacího tlačítka.
- **Vyčistit** - Najde všechny neplatné komponenty v dokumentu a odstraní je. Komponenta je neplatná, pokud neobsahuje žádnou platnou konfiguraci JSON nebo viditelný obsah jako obdélníky, texty atd.
- **Aktualizovat komponenty** - Rozšíření pro aktualizaci schématu na novou verzi.
- **Informace o verzi** - Zobrazí aktuální verzi REXYGEN HMI Designer a nástrojů pro HMI.

## 3.2 Obecná konfigurace HMI

Každá vlastní vizualizace vytvořená pomocí nástroje REXYGEN HMI Designer musí obsahovat obecnou konfiguraci, která je uložena v prvku HMI Config. Tento prvek je přidán automaticky s novým dokumentem nebo po prvním použití rozšíření *Upravit komponentu*. Obecná konfigurace se skládá ze tří částí



Obrázek 3.2: Seznam rozšíření RexHMI

- *Connections* - Seznam spojení mezi HMI a systémem REXYGEN.
- *Options* - Obecné nastavení HMI. Viz subsekcí 3.4.1 pro všechny možnosti.
- *HTML export* - Konfigurace HTML exportu.

### 3.2.1 Záložka Připojení (Connections tab)

Záložka Připojení obsahuje tabulky map mezi *Aliasem* (název datového bodu v HMI) a připojovacím řetězcem (cesta, odkud je hodnota čtena nebo zapisována do systému REXYGEN). Každé připojení může mít jeden ze dvou typů:

- *R* – Čtecí prvek, hodnoty jsou čteny z cíle,
- *W* – Zapisovatelný prvek, hodnota může být po změně zapsána do cíle.

Existují také dva speciální typy:

- *L* – Lokální prvek,
- *X* – Prvek s výrazem (bude popsán později).

Když je HMI konfigurace otevřena, *Aliases* datových bodů použitých v HMI komponentách se načtou automaticky. Také je vybrán typ *R* nebo *W* na základě typu komponenty (např. datový bod tlačítka *value* ukazuje na připojení *Write*, zatímco datový typ LED *value* ukazuje na připojení *Read*). Pro tyto připojení pouze použijte funkci pro procházení (*Browse*) k vyplnění připojovacího řetězce.

Stiskněte tlačítko *Procházet*. Otevře se dialog pro přihlášení. Pokud jste neprováděli změnu přihlašovacích údajů, použijte výchozí uživatelské jméno **admin** s prázdným heslem. Po úspěšném přihlášení se záložka Připojení rozšíří o stromovou strukturu běžícího algoritmu. Vyberte pole *Připojovací řetězec* daného *Aliasu*, poté projděte strukturu až k požadovanému bloku a proveďte **dvojklik** na parametr. Připojovací řetězec parametru je zkopírován do aliasu. Postupujte tímto způsobem pro vyplnění všech připojení *R* a *W*. Pro základní příklad viz [1].

**Lokální typ připojení (*L*)** reprezentuje proměnnou. Uživatel může definovat vlastní prvek, který lze číst nebo zapisovat pomocí standardních HMI komponent. Výchozí hodnota je nastavena pomocí pole *Připojovací řetězec*. *Lokální* prvek je dostupný pouze pro aktuální webovou stránku. Pokud tedy obnovíte webovou stránku nebo otevřete novou v jiné záložce, načte se výchozí hodnota.

**Výrazový typ připojení (*X*)** se používá pro vyhodnocení výrazu nad jinými aliasy. Alias je pouze pro čtení a aktualizuje se pokaždé, když se změní některý z vybraných prvků. Jméno aliasu použitého ve výrazu je uzavřeno v složených závorkách. Například:

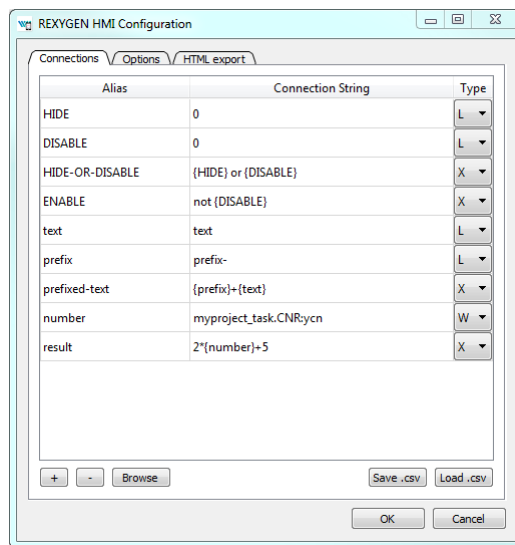
```
not {ALIAS}; {ALIAS-1} + {ALIAS-2}; 2*{ALIAS-1} + 5
```

Dostupné operace jsou:

- *Aritmetické operace* – +, -, \*, /, %
- *Bitové operace* – not (~), and (&), or (|), xor (^|)
- *Posun* – left (<<), right (>>), right logical (>>>)
- *Logické operace* – not, and, or, xor
- *Relační operace* – ==, !=, <, >, <=, >=
- *Konstanty* – pi, e, true, false, null

Výraz může směřovat na typ prvku. Může obsahovat konstanty, jako jsou čísla nebo řetězce. Typ hodnoty je nastaven na základě typu hodnoty z použitého připojení. Pokud není výraz možné vyhodnotit, uvidíte v chybovém protokolu na webové stránce HMI chybové hlášení. Můžete také použít výraz pro spojování řetězců. Příklad jednoduchého použití lokálního a výrazového typu je zobrazen na následujícím obrázku 3.3.

**Poznámka:** Pokud prvek s výrazem odkazuje na jiný prvek s výrazem, cílový prvek MUSÍ být vždy definován předem. Položky jsou řazeny abecedně podle použitého *Aliasu*, takže cílový prvek musí mít správný název.



Obrázek 3.3: Příklad použití připojení lokálního a výrazového typu.

### 3.2.2 Záložka Možnosti (Options tab)

Viz podsekcí 3.4.1 pro všechny dostupné možnosti.

### 3.2.3 Záložka Export HTML (HTML Export tab)

Existují dvě úrovně konfigurace exportu HTML. Základní úroveň umožňuje uživateli změnit titul webové stránky a titul v hlavičce webové stránky. Je také možné zakázat generování nápověd.

Když je aktivován *Expertní režim*, uživatel může změnit následující pokročilé možnosti:

- *Cesta ke konfiguračnímu souboru* – Pro pokročilé účely.
- *Vytvořit zobrazení* – Pokud je vybráno, dokument SVG se vejde do zobrazení vykreslené webové stránky. Poměr stran je zachován. Pokud je tato volba zrušena, velikost SVG se nemění, takže je viditelná pouze část dokumentu a může být zobrazena posuvníkem.
- *Cesta k uživatelské knihovně* – Každá komponenta HMI používá kombinaci souboru SVG a prostředků JavaScriptu. Výchozí knihovny jsou do projektu zařazeny automaticky, pokud položka je *Cesta k knihovně* prázdná. Pokud není, uživatel musí vybrat cesty k adresářům, kde se nacházejí prostředky pro komponenty HMI. Kontrola cesty během exportu HTML probíhá následovně:
  - *Absolutní cesta* – Pokud je zadaná cesta absolutní, použije se nebo se zobrazí chyba, pokud neexistuje.

- *Relativní cesta (není zaškrtnuto Vyjma výchozích knihoven)* – Zkontroluje, zda se řetězec v cestě ke knihovně neshoduje s názvem výchozí knihovny. Pokud ano, použije se výchozí knihovna. Pokud ne, zkontroluje se, zda existuje, a použije se, nebo se zobrazí chyba.
- *Relativní cesta (je zaškrtnuto Vyjma výchozích knihoven)* – Během exportu HTML se zkontroluje, zda adresář na relativní cestě existuje, nebo se zobrazí chyba.

## 3.3 Rozšíření pro REXYGEN HMI Designer

### 3.3.1 Animace skupin

Rozšíření animace skupin umožňuje změnit SVG skupinu na vlastní HMI komponentu. Nejprve nakreslete nějaký útvar nebo seskupte několik prvků dohromady pomocí klávesové zkratky *Ctrl + G*. Následně máte několik možností:

- *Posunout X* – Posune zvolenou skupinu ve směru osy X podle signálu ze zvoleného datového bodu.
- *Posunout Y* – Posune zvolenou skupinu ve směru osy Y podle signálu ze zvoleného datového bodu.
- *Otočit* – Otočí zvolenou skupinu kolem středu rotace zvoleného ve skupině před zavoláním rozšíření. Otočení je nastaveno podle signálu ze zvoleného datového bodu.
- *Změnit velikost* – Změní velikost ve zvoleném směru podle signálu ze zvoleného datového bodu.
- *Průhlednost* – Změní průhlednost celé skupiny podle signálu ze zvoleného datového bodu.
- *Změnit barvu* – Změní barvu obrysu nebo výplně podle signálu ze zvoleného datového bodu.
- *Změnit skupinu na tlačítko* – Vytvoří vlastní tlačítko, které má stejné parametry a chování jako komponenta *Tlačítko*.

**Poznámka:** Skupina může buď být tlačítkem, nebo mít několik animací.

Když jsou všechny animace zkontrolované, stiskněte OK pro vytvoření vlastní komponenty. Od této chvíle jsou parametry komponenty upravitelné pomocí rozšíření *Editor prvků*. Pokud již není některá animace zapotřebí, můžete jednoduše znovu otevřít rozšíření animace skupin a změnit, co je potřeba.

### 3.3.2 Aktualizovat komponenty

Rozšíření aktualizace komponent zkontroluje všechny komponenty v souboru a porovná je mezi různými verzemi. Pokud je verze jiná, navrhne typ aktualizace. Existují tři takové typy:

- *Žádná (None)* – Nprovede nic.
- *Částečná (Partial)* – Změní metadata a zkopíruje parametry. Nezasahuje do podkladového SVG.
- *Úplná (Full)* – Aktualizuje celé SVG komponenty a metadata

Aktualizační algoritmus zvolí nejvíce vyhovující možnost.

## 3.4 Seznam dostupných komponent

Existuje několik předdefinovaných HMI komponent v knihovně komponent. Každá komponenta je reprezentována jako jeden SVG soubor, který může být přetáhnut do aktuálního vizualizačního schématu. Knihovna je rozdělena na pět základních skupin:

- **GENERAL** - Skupina běžných komponent jako tlačítka, vstupy, posuvníky nebo měřidla pro vizualizaci základních proměnných.
- **HOME** - Skupina komponent pro automatizaci domácnosti, jako různá potrubí, ventily, radiátory, bojler apod.
- **HVAC** - Specializovaná knihovna pro vizualizace zateplování, ventilace a větrání.
- **CHARTS** - Specializovaná knihovna pro různé grafy.
- **PROC** - Specializovaná knihovna pro vizualizace řízení procesů.

Každá komponenta má určité datové body, které poskytují interakci mezi živými daty a stavem HMI. Většinou lze narazit na:

- `value` - hlavní datový bod, kde se čte nebo zapisuje hodnota.
- `disable_by` - komponenta je deaktivována při nenulové hodnotě.
- `hide_by` - komponenta je skryta při nenulové hodnotě.

a další možnosti, které jsou popsány v následujícím textu. Komponenty jsou řazené abecedně.

### 3.4.1 HMIConfig

Globální konfigurační komponenta REXYGEN HMI. Uživatel může měnit nastavení projektu, konfigurovat HTML export a definovat seznam spojení mezi REXYGEN targetem a HMI.



Vlastnost	Typ	Popis
<code>refresh_rate</code>	int	Změní výchozí časový interval pro čtení hodnot z targetu. Výchozí hodnota je 500ms.
<code>log_level</code>	enum	Nastaví globální úroveň logování pro celé RexHMI. Zprávy jsou zapsány jak do menu v pravém horním rohu, tak do konzole webového prohlížeče. Výchozí úroveň je INFO.
<code>show_clock</code>	bool	Pokud je True, hodiny jsou zobrazeny v pravém horním rohu šablony místo blikajícího čtverce. Pokud je target připojen, hodiny se aktualizují, pokud dojde k chybě připojení nebo ztrátě času.
<code>use_client_time</code>	bool	Pokud je True, <code>show_clock</code> zobrazí čas z počítače klienta namísto targetu. To lze použít, když zařízení targetu nemá synchronizaci NTP nebo čip RTC.
<code>header_mode</code>	enum	Změňte výchozí velikost záhlaví (tenké, balené). Pokud je baleno, klikněte na logo, abyste zobrazili záhlaví.
<code>kiosk_mode</code>	bool	Pokud je True, odkazy z hlavičky (z loga a hlavičky) jsou odstraněny a uživatel nemůže změnit časový interval obnovení ani cíl.
<code>target_url</code>	str	URL cílového zařízení. Zatím je podporován pouze protokol WebSocket. URL musí také obsahovat PORT, ale protokol lze vynechat. Výchozí hodnota je prázdný řetězec, což znamená <code>ws://127.0.0.1:8008</code> Příklady: <code>192.168.1.100:8008</code> nebo <code>wss://192.168.1.100:8009</code> <i>Poznámka: Pokud je HMI hostováno na interním webovém serveru, je tato možnost ignorována a HMI se připojuje k hostname / IP, kterou používáte v URL webového prohlížeče.</i>

### 3.4.2 CHARTS

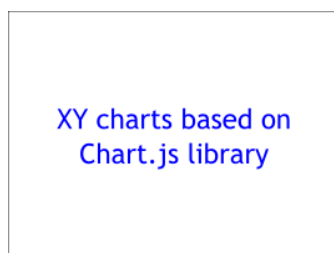
#### RadarChart



Radarový graf založený na knihovně Chart.js.

Data	Volitelné	Popis
<code>values</code>		Pole signálů.
<code>hide_by</code>	True	Komponenta je skryta, pokud hodnota není nulová.
<code>disable_by</code>	True	Komponenta je zakázána, pokud hodnota není nulová a data se již neaktualizují.
Vlastnost	Typ	Popis
<code>chartJsOptions</code>	object	Vlastní nastavení pro knihovnu Chart.js. Více informací na <a href="https://www.chartjs.org">https://www.chartjs.org</a> .
<code>min</code>	int	Minimální hodnota osy
<code>max</code>	int	Maximální hodnota osy
<code>step</code>	int	Úhel mezi dvěma vzorky [stupně]
<code>labelStep</code>	int	Počet kroků, kdy se vygeneruje popisek.

## XYChart



XY graf založený na knihovně Chart.js. Signály jsou čteny z bloku TRND(V) nebo pole. Každé dva signály jsou použity jako X a Y souřadnice pro každý signál.

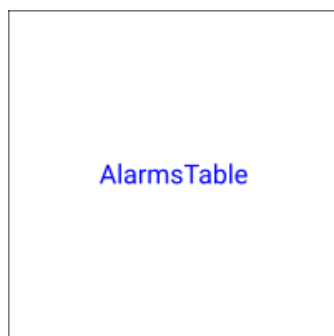
Data	Volitelné	Popis
<code>values</code>		Pole XY dat ve sloupcích nebo odkaz na blok TRND.
<code>hide_by</code>	True	Komponenta je skryta, pokud hodnota není nulová.
<code>disable_by</code>	True	Komponenta je zakázána, pokud hodnota není nulová a data se již neaktualizují.
<code>reset_by</code>	True	Pokud není nulová, data v grafu jsou vymazána.
Vlastnost	Typ	Popis
<code>title</code>	str	Název grafu
<code>signals</code>	table	Tabulka popisků signálů. Zadejte popisky ve stejném pořadí, v jakém jste signály připojili k bloku Array nebo TRND.
<code>bufferSize</code>	int	Použije se pouze při připojení k bloku TRND. Maximální počet uložených vzorků. Pokud je -1, počet vzorků se čte z konfigurace bloku TRND.
<code>hideLegend</code>	bool	Pokud je TRUE, legenda je skryta.

<code>showLines</code>	bool	Pokud je TRUE, body XY jsou spojeny pomocí čáry.
<code>xAxis</code>	table	Tabulka základních parametrů pro osu X. Použije se pouze první řádek!
<code>yAxis</code>	table	Tabulka základních parametrů pro osu Y. Použije se pouze první řádek!
<code>chartJsOptions</code>	object	Vlastní možnosti pro knihovnu Chart.js. Viz <a href="https://www.chartjs.org">https://www.chartjs.org</a> pro více informací.

---

### 3.4.3 GENERAL

#### AlarmsTable



Komponenta tabulky, která zobrazuje aktivní alarmy z interního subsystému alarmů REXYGENu.

Vlastnost	Typ	Popis
<code>columns</code>	array	Určuje, které sloupce budou viditelné v tabulce alarmů. Sloupce se zobrazují v pořadí, jak jsou uvedeny v poli.
<code>fontScale</code>	float	Tento parametr slouží k nastavení velikosti písma.
<code>timeFormat</code>	string	Formát data a času zobrazených časových údajů v tabulce. Pro více informací viz <code>luxon.js</code>

---

#### BarGraph



Bargraph komponenta, která vykresluje stupnici na levé straně a obsahuje barevné zóny a zobrazuje aktuální hodnoty

Data	Volitelné	Popis
<code>value</code>		Zobrazená hodnota
<code>disable_by</code>	True	Komponenta je zakázána, pokud hodnota není nulová.
<code>hide_by</code>	True	Komponenta je skryta, pokud hodnota není nulová.
Vlastnost	Typ	Popis
<code>rangeMin</code>	float	Dolní limit pro hodnotu bargraphu
<code>rangeMax</code>	float	Horní limit pro hodnotu bargraphu
<code>tickStep</code>	float	Hodnota pro dílčí krok mezi hlavními kroky.
<code>mainTickStep</code>	float	Hodnota pro hlavní krok stupnice.
<code>decimals</code>	float	Kolik desetinných míst je zobrazeno
<code>units</code>	str	Text, který je zobrazen na spodní straně bargraphu. Obvykle název jednotek.
<code>colorZones</code>	table	Uživatel může definovat několik barevných zón, které mohou představovat nějakou úroveň varování. Zóny jsou zobrazeny na pravé straně bargraphu. Každá má vlastní barvu a rozsah definovaný <code>startValue</code> a <code>endValue</code> . Stačí dvakrát kliknout na <code>colorZones</code> a změnit je v tabulce.
<code>colorOffLimits</code>	color	Barva okraje bargraphu, když je hodnota limitů <code>rangeMin</code> a <code>rangeMax</code> .
<code>levelColor1</code>	color	Změna hlavní barvy gradientu bargraphu
<code>levelColor2</code>	color	Změna doplňkové barvy gradientu bargraphu

## Battery



Zobrazí stav baterie od 0 do 100 %.

Data	Volitelné	Popis
------	-----------	-------

value		Úroveň baterie od 0 do 100
Vlastnost	Typ	Popis

## Button



Komponenta tlačítka mění logickou hodnotu. Obvykle stav ON znamená hodnotu TRUE a stav OFF znamená hodnotu FALSE. Pokud je to nutné, parametr `reverseMeaning` přepne zapsané hodnoty.

Existují tři různé režimy provozu:

`PushButton` při stisku nebo doteku se stav ON zapíše do datového bodu `value` cíle. Po uvolnění je zapsán stav OFF.

`ManualPulse` vždy zapíše hodnotu TRUE po stisknutí tlačítka, uvolnění je ignorováno. (Vhodné pro řízení bloku MP)

`ToggleButton` když je tlačítko stisknuto, hodnota se přepne (ON -> OFF -> ON).

Data	Volitelné	Popis
value		Hlavní datový bod, do kterého je zapsána logická hodnota.
refresh_from	True	Pokud je nastaveno, stav komponenty je aktualizován na základě hodnoty přečtené z tohoto datového bodu. Jinak se použije hodnota z hlavního datového bodu.
disable_by	True	Pokud není nula, komponenta je zakázána.
hide_by	True	Komponenta je skryta, pokud hodnota není nulová.
readonly_by	True	Komponenta přejde do režimu pouze pro čtení, pokud hodnota tohoto parametru není nulová.

Vlastnost	Typ	Popis
type	enum	Změní typ tlačítka. <code>PushButton</code> při stisku nebo doteku se stav ON zapíše do datového bodu <code>value</code> cíle. Po uvolnění je zapsán stav OFF. <code>ManualPulse</code> zapíše hodnotu TRUE jednou po stisknutí tlačítka, uvolnění je ignorováno. <code>ManualPulseRpt</code> zapisuje hodnotu TRUE periodicky (20 ms perioda) po stisknutí tlačítka, uvolnění zastaví zápis. (Vhodné pro řízení bloku MP) <code>ToggleButton</code> když je tlačítko stisknuto, hodnota se přepne (ON -> OFF -> ON).

<code>reverseMeaning</code>	bool	Pokud je hodnota True, význam hodnoty je přepnut. Stav ON je tedy False a stav OFF je True.
<code>fontScale</code>	float	Pomocí tohoto parametru můžete měnit velikost písma, protože tlačítko používá komponentu tlačítka HTML, která není svázána s SVG.
<code>labelFalse</code>	str	Popisek tlačítka, když je stav OFF
<code>labelTrue</code>	str	Popisek tlačítka, když je stav ON. Pokud není definováno, použije se <code>labelFalse</code> .
<code>colorFalse</code>	color	Barva pozadí tlačítka, když je stav OFF
<code>colorTrue</code>	color	Barva pozadí tlačítka, když je stav ON
<code>labelColorFalse</code>	color	Barva popisku (textu), když je stav OFF.
<code>labelColorTrue</code>	color	Barva popisku (textu), když je stav ON
<code>writePerm</code>	table	Seznam přihlášení, která mají povoleno zapisovat do datového bodu. Pokud nejsou v seznamu, bude tlačítko zakázáno.

---

## Checkbox



Checkbox.svg

Data	Volitelné	Popis
<code>value</code>		Datapoint, kam se zapisuje boolean hodnota.
<code>refresh_from</code>	True	Pokud je nastaveno, stav komponenty se aktualizuje na základě hodnoty z tohoto datapointu. Jinak se použije hodnota z hlavního datapointu.
<code>hide_by</code>	True	Komponenta je skryta, pokud hodnota není nulová.
<code>disable_by</code>	True	Komponenta je zakázána, pokud hodnota není nulová a data se již neaktualizují.
<code>readonly_by</code>	True	Komponenta přejde do režimu pouze pro čtení, pokud hodnota tohoto parametru není nulová.
Vlastnost	Typ	Popis
<code>reverse_meaning</code>	bool	Pokud je True, mění se význam hodnoty. Takže stav ON je False a stav OFF je True.
<code>show_cross</code>	bool	Pokud je True, křížek se zobrazí v případě hodnoty OFF.

---

## ComboBox



Combobox umožňuje vybrat hodnotu z předdefinovaného seznamu. Pokud hodnota v systému není v seznamu, je automaticky přidána.

Data	Volitelné	Popis
<code>value</code>		Datapoint, kam se zapisují hodnoty po výběru.
<code>refresh_from</code>	True	Pokud je nastaveno, stav komponenty je aktualizován na základě hodnoty přečtené z tohoto datového bodu. Jinak se použije hodnota z hlavního datového bodu.
<code>values</code>	True	Datapoint, který ukazuje na seznam hodnot / popisů oddělených   např. 0: NOK   1: OK (typicky parametr <code>pupstr</code> bloku CNE). Pokud je nastaven, parametr <code>values</code> je ignorován.
<code>hide_by</code>	True	Komponenta je skryta, pokud hodnota není nulová.
<code>disable_by</code>	True	Komponenta je zakázána, pokud hodnota není nulová.
<code>readonly_by</code>	True	Komponenta přejde do režimu pouze pro čtení, pokud hodnota tohoto parametru není nulová.
Vlastnost	Typ	Popis
<code>values</code>	table	Definuje páry hodnota:popis pro seznam comboboxu. Hodnota je číslo nebo řetězec a popis je jeho řetězcový popis.
<code>showValue</code>	bool	Pokud je True, hodnoty jsou zobrazeny v seznamu např. 0:OK
<code>valueType</code>	enum	Určuje, zda je hodnota čtena jako číslo nebo řetězec.
<code>fontScale</code>	float	Pomocí tohoto parametru můžete měnit velikost písma, protože Combobox používá HTML komponentu select, která není svázána s SVG.
<code>writePerm</code>	table	Seznam přihlášení, která mají povoleno zapisovat do datového bodu. Pokud není v seznamu, bude ComboBox zakázán.

## ControlLed





Jednoduchý indikátor stavu. Prázdný nebo nula znamená VYPNUTO, jinak ZAPNUTO.

Data	Volitelné	Popis
<code>value</code>		Datapoint ze kterého se čte stav LED.
<code>disable_by</code>	True	Komponenta je zakázána, pokud hodnota není nulová.
<code>hide_by</code>	True	Komponenta je skryta, pokud hodnota není nulová.
Vlastnost	Typ	Popis
<code>reverseMeaning</code>	bool	Pokud je True, význam hodnoty je přepnut. Takže stav ON je False a stav OFF je True.
<code>color_true</code>	color	Barva pro stav ZAPNUTO.
<code>color_false</code>	color	Barva pro stav VYPNUTO.

### CustomHTML



Komponenta CustomHTML umožňuje vložit vlastní HTML obsah do HMI. Vytvoří prázdný DIV element, který je naplněn obsahem z parametru `html`. Může být použit například pro vložení živého obrazu z kamery.

Vlastnost	Typ	Popis
<code>html</code>	str	Vlastní HTML obsah, který je přímo vkládán do HMI.

## Display

**X.XX**

Komponenta zobrazuje hodnotu ze systému jako textový řetězec. Může to být buď `number`, `date`, `datetime` nebo jednoduchá hodnota jako `text` řetězec v závislosti na vlastnosti `format`. Každý displej může mít jednotky (`units`), které jsou zobrazeny na konci zobrazeného řetězce.

Data	Volitelné	Popis
<code>value</code>		Datapoint, ze kterého je hodnota čtena.
<code>disable_by</code>	True	Komponenta je zakázána, pokud hodnota není nulová a data se již neaktualizují.
<code>hide_by</code>	True	Komponenta je skryta, pokud hodnota není nulová.
Vlastnost	Typ	Popis
<code>format</code>	enum	<code>number</code> - Pokud je vybráno, je přečtená hodnota převedena na číslo, buď celé nebo desetinné. Také parametry <code>scale</code> , <code>offset</code> a <code>decimals</code> se používají během konverze. Barva textu závisí na vlastnosti <code>color</code> , pokud je hodnota vyšší než <code>rangeMax</code> , je použita barva <code>colorAbove</code> , stejně tak, pokud je hodnota nižší než <code>rangeMin</code> , je použita barva <code>colorBelow</code> . <code>time</code> a <code>datetime</code> - Zobrazovač převede přečtenou hodnotu v sekundách od REXYGEN Epoch na aktuální čas nebo datum a čas. Vlastnost <code>text_format</code> se v tomto kontextu používá, kde lze vybrat formát zobrazeného času. <code>text</code> - Jednoduše zobrazí hodnotu ve formátu, jak byla přijata. Vhodné pro krátké řetězce.
<code>text_format</code>	enum	Vybere formát řetězce času pro <code>time</code> a <code>datetime</code> .
<code>scale</code>	float	Změna měřítka přečtené hodnoty. Použitelné pouze pro formát <code>number</code> .
<code>offset</code>	float	Přidá offset k přečtené hodnotě. Použitelné pouze pro formát <code>number</code> .
<code>decimals</code>	int	Zaokrouhlí přečtenou hodnotu na definovaný počet desetinných míst. Použitelné pouze pro formát <code>number</code> .
<code>color</code>	color	Normální barva písma displeje. Použitelné pouze pro formát <code>number</code> .
<code>colorAbove</code>	color	Barva písma displeje, když je přečtená hodnota vyšší než <code>rangeMax</code> . Použitelné pouze pro formát <code>number</code> .

<code>colorBelow</code>	color	Barva písma displeje, když je přečtená hodnota nižší než <code>rangeMin</code> . Použitelné pouze pro formát <code>number</code> .
<code>rangeMax</code>	float	Když je přečtená hodnota vyšší než <code>rangeMax</code> , je použita barva <code>colorAbove</code> . Použitelné pouze pro formát <code>number</code> .
<code>rangeMin</code>	float	Když je přečtená hodnota nižší než <code>rangeMin</code> , je použita barva <code>colorBelow</code> . Použitelné pouze pro formát <code>number</code> .
<code>units</code>	str	Připojí nějaký řetězec na konec zobrazené hodnoty. Obvykle název jednotek.

### DisplayMatrix



Zobrazí matici z libovolného pole. Stačí předat do `value` datový bod, který obsahuje pole. Například `CNA:acn` nebo `MX_MAT:ay`.

Data	Volitelné	Popis
<code>value</code>		Hodnota z datového bodu bude zobrazena jako matice.
<code>disable_by</code>	True	Komponenta je zakázána, pokud hodnota není nulová a data se již neaktualizují.
<code>hide_by</code>	True	Komponenta je skryta, pokud hodnota není nulová.
Vlastnost	Typ	Popis
<code>transpose</code>	bool	Pokud je True, zobrazí transponovanou matici.
<code>fontScale</code>	float	Pomocí tohoto parametru můžete změnit velikost písma.
<code>decimals</code>	int	Zaokrouhlí hodnotu v buňce na zadaný počet desetinných míst.
<code>header</code>	table	Hlavička tabulky, pole řetězců pro každý sloupec.
<code>showLineNumbers</code>	bool	Pokud je True, bude přidáno číslo řádku do prvního sloupce.
<code>divClass</code>	string	Vlastní CSS třída pro hlavní DIV element komponenty. Seznam řetězců oddělených mezerami.

<code>tableClass</code>	string	Vlastní CSS třída pro HTML tabulku. Seznam řetězců oddělených mezerami.
-------------------------	--------	---

---

### DisplayMatrixExt



Zobrazí matici z libovolného pole. Stačí předat do `value` datový bod, který obsahuje pole. Například `CNA:acn` nebo `MX_MAT:ay`.

Data	Volitelné	Popis
<code>value</code>		Hodnota z datového bodu bude zobrazena jako matice.
<code>row</code>		Pokud není nulová, řádek bude vybrán.
<code>disable_by</code>	True	Komponenta je zakázána, pokud hodnota není nulová a data se již neaktualizují.
<code>hide_by</code>	True	Komponenta je skryta, pokud hodnota není nulová.
Vlastnost	Typ	Popis
<code>transpose</code>	bool	Pokud je True, zobrazí transponovanou matici.
<code>fontScale</code>	float	Pomocí tohoto parametru můžete změnit velikost písma.
<code>decimals</code>	int	Zaokrouhlí hodnotu v buňce na zadaný počet desetinných míst.
<code>header</code>	table	Hlavička tabulky, pole řetězců pro každý sloupec.
<code>showLineNumbers</code>	bool	Pokud je True, číslo řádku bude přidáno do prvního sloupce.
<code>divClass</code>	string	Vlastní CSS třída pro hlavní DIV element komponenty. Seznam řetězců oddělených mezerami.
<code>tableClass</code>	string	Vlastní CSS třída pro HTML tabulku. Seznam řetězců oddělených mezerami.
<code>transformFcn</code>	string	Tělo funkce bude vyhodnoceno jako <code>'function transform(row,column,value)'</code> a očekává <code>'value'</code> jako výsledek.

---

## DisplayString

### XXX DISPLAY STRING XXX

Komponenta `Display string` čte hodnotu z cíle a zobrazí ji jako text. Může zobrazit hodnotu přímo jako `format: text` nebo vyhledat hodnotu v tabulce `alt` (alternativní tabulka) popisů.

*Poznámka: Pokud hodnota obsahuje znak nového řádku, je text zalamován.*

Data	Volitelné	Popis
<code>value</code>		Datapoint, ze kterého je hodnota čtena.
<code>disable_by</code>	True	Komponenta je zakázána, pokud hodnota není nulová a data se již neaktualizují.
<code>hide_by</code>	True	Komponenta je skryta, pokud hodnota není nulová.
Vlastnost	Typ	Popis
<code>format</code>	enum	<code>text</code> - zobrazí přečtenou hodnotu tak, jak je. <code>alt</code> - Vyhledá popis pro přečtenou hodnotu v seznamu <code>values</code> . Pokud není nalezen, zobrazí hodnotu tak, jak je.
<code>showValue</code>	bool	Pokud je hodnota True, přidá před popisem hodnotu oddělenou dvojtečkou. Příklad: 0: Status OK
<code>values</code>	table	Seznam hodnota:popis, kde je popis vyhledán při načtení hodnoty.

## DisplayWithBox



Komponenta zobrazuje hodnotu ze systému jako řetězec. Může to být buď `number`, `date` nebo `datetime` řetězec v závislosti na vlastnosti `format`. Každý displej může mít nastaveny jednotky (`units`), které jsou zobrazeny ve spodní části.

Data	Volitelné	Popis
<code>value</code>		Datapoint, ze kterého je hodnota čtena.
<code>disable_by</code>	True	Komponenta je zakázána, pokud hodnota není nulová a data se již neaktualizují.
<code>hide_by</code>	True	Komponenta je skryta, pokud hodnota není nulová.

Vlastnost	Typ	Popis
<code>format</code>	enum	<code>number</code> - Pokud je vybráno, je přečtená hodnota převedena na celé nebo desetinné číslo. Také jsou použity parametry <code>scale</code> , <code>offset</code> a <code>decimals</code> . Barva textu závisí na vlastnosti <code>color</code> , pokud je hodnota vyšší než <code>rangeMax</code> , je použita barva <code>colorAbove</code> , stejně tak, pokud je hodnota nižší než <code>rangeMin</code> , je použita barva <code>colorBelow</code> . <code>time</code> a <code>datetime</code> - Displej převede přečtenou hodnotu v sekundách od REXYGEN Epoch na aktuální čas nebo datum a čas.
<code>scale</code>	float	Změna měřítka přečtené hodnoty. Použitelné pouze pro formát <code>number</code> .
<code>offset</code>	float	Přidá offset k přečtené hodnotě. Použitelné pouze pro formát <code>number</code> .
<code>decimals</code>	int	Zaokrouhlí přečtenou hodnotu na definovaný počet desetinných míst. Použitelné pouze pro formát <code>number</code> .
<code>color</code>	color	Normální barva písma displeje. Použitelné pouze pro formát <code>number</code> .
<code>colorAbove</code>	color	Barva písma displeje, když je přečtená hodnota vyšší než <code>rangeMax</code> . Použitelné pouze pro formát <code>number</code> .
<code>colorBelow</code>	color	Barva písma displeje, když je přečtená hodnota nižší než <code>rangeMin</code> . Použitelné pouze pro formát <code>number</code> .
<code>rangeMax</code>	float	Když je přečtená hodnota vyšší než <code>rangeMax</code> , je použita barva <code>colorAbove</code> . Použitelné pouze pro formát <code>number</code> .
<code>rangeMin</code>	float	Když je přečtená hodnota nižší než <code>rangeMin</code> , je použita barva <code>colorBelow</code> . Použitelné pouze pro formát <code>number</code> .
<code>units</code>	str	Připojí řetězec na konec zobrazené hodnoty. Obvykle název jednotek.

## Gauge180



Gauge zobrazuje přečtenou hodnotu na stupnici pomocí ručičky. Rozsah stupnice je omezen parametry `rangeMin` a `rangeMax`. Stupnice je dále dělena na hlavní kroky pomocí parametru `mainTickStep` a na menší pomocí `tickStep`. Přečtená hodnota je zobrazena

také jako číslo zaokrouhlené na `decimals` desetinných míst. Pokud je přečtená hodnota mimo rozsah, změní se barva okraje na `colorOffLimits`. Gauge také obsahuje speciální druh stupnice, která je definována tabulkou `colorZones`. Například může být použita pro definování normálních, varovných a chybových úrovní.

Data	Volitelné	Popis
<code>value</code>		Datapoint, ze kterého je hodnota čtena.
<code>disable_by</code>	True	Komponenta je zakázána, pokud hodnota není nulová a data se již neaktualizují.
<code>hide_by</code>	True	Komponenta je skryta, pokud hodnota není nulová.
Vlastnost	Typ	Popis
<code>rangeMin</code>	float	Minimální hodnota pro stupnici Gauge.
<code>rangeMax</code>	float	Maximální hodnota pro stupnici Gauge.
<code>tickStep</code>	float	Nakreslí kroky na stupnici každých <code>tickStep</code> hodnot.
<code>mainTickStep</code>	float	Nakreslí hlavní kroky na stupnici každých <code>mainTickStep</code> hodnot s textovým popisem.
<code>decimals</code>	float	Zaokrouhlí přečtenou hodnotu na definovaný počet desetinných míst.
<code>units</code>	str	Jednotky nebo jen textový popis.
<code>colorZones</code>	table	Definuje speciální stupnici s barevnými segmenty začínající na <code>startValue</code> a končící na <code>endValue</code> obarvené barvou <code>color</code> .
<code>colorOffLimits</code>	color	Barva okraje, když je přečtená hodnota mimo rozsah.

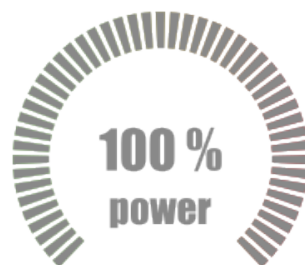
## Gauge270



Gauge zobrazuje přečtenou hodnotu na stupnici pomocí ručičky. Rozsah stupnice je omezen parametry `rangeMin` a `rangeMax`. Stupnice je dále dělena na hlavní kroky pomocí parametru `mainTickStep` a na menší pomocí `tickStep`. Přečtená hodnota je zobrazena také jako číslo zaokrouhlené na `decimals` desetinných míst. Pokud je přečtená hodnota mimo rozsah, změní se barva okraje na `colorOffLimits`. Gauge také obsahuje speciální druh stupnice definované tabulkou `colorZones`. Například může být použita pro definici normální, varovné a chybové úrovně.

Data	Volitelné	Popis
<code>value</code>		Datapoint, ze kterého je hodnota čtena.
<code>disable_by</code>	True	Komponenta je zakázána, pokud hodnota není nulová a data se již neaktualizují.
<code>hide_by</code>	True	Komponenta je skryta, pokud hodnota není nulová.
Vlastnost	Typ	Popis
<code>rangeMin</code>	float	Minimální hodnota pro stupnici Gauge.
<code>rangeMax</code>	float	Maximální hodnota pro stupnici Gauge.
<code>tickStep</code>	float	Nakreslí kroky na stupnici každých <code>tickStep</code> hodnot.
<code>mainTickStep</code>	float	Nakreslí hlavní kroky na stupnici každých <code>mainTickStep</code> hodnot s textovým popisem.
<code>decimals</code>	float	Zaokrouhlí přečtenou hodnotu na definovaný počet desetinných míst.
<code>units</code>	str	Jednotky nebo jen textový popis.
<code>colorZones</code>	table	Definuje speciální stupnici s barevnými segmenty začínajícími na <code>startValue</code> a končícími na <code>endValue</code> obarvené barvou <code>color</code> .
<code>colorOffLimits</code>	color	Barva okraje, když je přečtená hodnota mimo rozsah.

## GaugeBars



Komponenta `GaugeBars` zobrazuje aktuální hodnotu jako počet zvýrazněných čárek. Hodnota by měla být mezi `rangeMin` a `rangeMax`. Také je zobrazena přečtená hodnota zaokrouhlená na `decimals` s případným řetězcem `units`.

Data	Volitelné	Popis
<code>value</code>		Datapoint, ze kterého je hodnota čtena.
Vlastnost	Typ	Popis
<code>rangeMin</code>	float	Minimální hodnota pro Gauge.
<code>rangeMax</code>	float	Maximální hodnota pro Gauge.



<code>decimals</code>	<code>int</code>	Zaokrouhlí přečtenou hodnotu na definovaný počet desetinných míst.
<code>units</code>	<code>str</code>	Název jednotek nebo jen textový popis.

---

### GaugeGradient

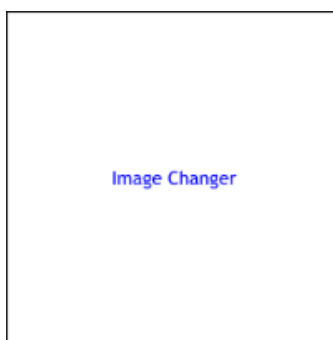


Komponenta GaugeGradient zobrazuje aktuální hodnotu jako vyplnění oblouku. Hodnota by měla být mezi `rangeMin` a `rangeMax`. Minimální hodnota znamená, že celý oblouk je červený, maximální hodnota je celý oblouk zelený.

Data	Volitelné	Popis
<code>value</code>		Datapoint, ze kterého je hodnota čtena.
Vlastnost	Typ	Popis
<code>rangeMin</code>	<code>float</code>	Minimální hodnota pro Gauge.
<code>rangeMax</code>	<code>float</code>	Maximální hodnota pro Gauge.

---

### ImageChanger



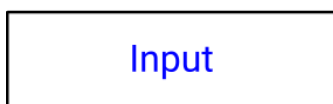
Komponenta Image Changer načte obrázek (\*.png,\*.jpg,\*.bmp,\*.svg atd.) a zobrazí jej ve svých hranicích. Obrázek je obnovován při změně hodnoty `refresh_from` nebo po uplynutí času `period`. Cesta k obrázku `imagePath` je relativní k rootu webového serveru. Pokud je v `imagePath` nastaveno {0} a `appendKey` je True, je {0} nahrazeno hodnotou

z `refresh_from`.

Příklad `imagePath /hmi/images/img- $\{0\}$ .png` a hodnota může být 0 až 10 atd.

Data	Volitelné	Popis
<code>refresh_from</code>	True	Při změně hodnoty je obrázek aktualizován.
<code>hide_by</code>	True	Komponenta je skryta, pokud hodnota není nulová.
<code>disable_by</code>	True	Komponenta je zakázána, pokud hodnota není nulová a data se již neaktualizují.
Vlastnost	Typ	Popis
<code>imagePath</code>	str	Relativní cesta k obrázku od rootu webového serveru.
<code>appendKey</code>	bool	Pokud je True, $\{0\}$ je nahrazeno aktuální hodnotou <code>refresh_from</code> v <code>imagePath</code> .
<code>period</code>	int	Pokud je větší než nula, obrázek bude automaticky obnovován podle zadaného parametru <code>period</code> v milisekundách. Datapoint <code>refresh_from</code> je použit pouze pro režim <code>appendKey</code> .
<code>preserveAspectRatio</code>	enum	Hodnota pro atribut <code>preserveAspectRatio</code> elementu SVG image.

## Input



Komponenta umožňuje zapisovat čísla, texty, datумы nebo časy do cílového bodu. Pokud je vybrán formát `number`, hodnota může být převedena pomocí `scale` a `offset` a zaokrouhlena na počet `decimals`. Vstup umožňuje omezit vstup uživatele na rozsah definovaný parametry `min` a `max`. Formát `date` a `datetime` má speciální widget pro zadávání dat. Hodnota z tohoto formátu je převedena na počet milisekund od REXYGEN epochy. (`datetime` - 'HH:mm', `datetime-seconds` - 'HH:mm:ss')

Formát `time` převádí hodnotu na počet sekund. (`time` - 'HH:mm', `time-seconds` - 'HH:mm:ss')

*Poznámka: Hodnota je zapsána do datového bodu `value`, pokud je to nutné, může být obnovena z jiného místa definovaného v datovém bodu `refresh_from`. Může být použito, pokud je hodnota omezena saturací.*

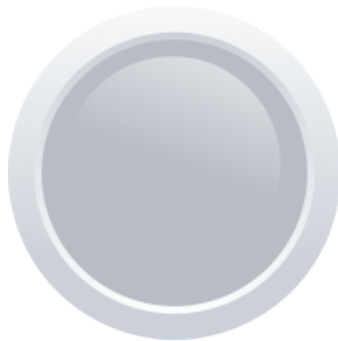
Data	Volitelné	Popis
<code>value</code>		Datový bod, kam je hodnota zapsána.

<code>refresh_from</code>	True	Volitelný datový bod, kde je hodnota aktualizována.
<code>hide_by</code>	True	Komponenta je skryta, pokud hodnota není nulová.
<code>disable_by</code>	True	Komponenta je zakázána, pokud hodnota není nulová a data se již neaktualizují.
<code>readonly_by</code>	True	Komponenta přejde do režimu pouze pro čtení, pokud hodnota tohoto parametru není nulová.
Vlastnost	Typ	Popis
<code>format</code>	enum	<b>number</b> - Pokud je vybrán, hodnota je převedena na číslo, buď celé nebo desetinné. Také se používají parametry <b>scale</b> , <b>offset</b> a <b>decimals</b> . <b>date</b> a <b>datetime</b> - Vstup převede datum nebo datum a čas na počet sekund od REXYGEN Epoch. <b>time</b> - Vstup převede čas na počet sekund. <b>text</b> - Jednoduše zadejte hodnotu ve formátu, jak je. Vhodné pro krátké řetězce. <b>password</b> - Textové pole, které zobrazuje hvězdičky.
<code>fontScale</code>	float	Vstup používá komponentu HTML input, která není svázána s SVG. Použijte tento parametr k změně měřítka písma.
<code>textAlign</code>	enum	Vstup používá komponentu HTML input, která není svázána s SVG. Použijte tento parametr k změně zarovnání textu.
<code>scale</code>	float	Změna měřítka přečtené hodnoty. Použitelné pouze pro formát <b>number</b> .
<code>offset</code>	float	Přidá offset k přečtené hodnotě. Použitelné pouze pro formát <b>number</b> .
<code>decimals</code>	int	Zaokrouhlí přečtenou hodnotu na definovaný počet desetinných míst. Použitelné pouze pro formát <b>number</b> .
<code>digits</code>	int	Celkový počet číslic bez desetinných míst. Zbývající číslice budou vyplněny nulami. Použitelné pouze pro formát <b>number</b> .
<code>min</code>	float	Minimální hodnota, kterou může uživatel zadat. Použitelné pouze pro formát <b>number</b> .
<code>max</code>	float	Maximální hodnota, kterou může uživatel zadat. Použitelné pouze pro formát <b>number</b> .
<code>setOnBlur</code>	bool	Pokud je True, hodnota vstupu dat je nastavena na událost blur (stiskněte ESC pro zrušení zápisu).
<code>css</code>	object	Vlastní CSS styl pro HTML vstupní objekt. Například <code>{"background-color":"#00ffff"}</code>

<code>writePerm</code>	table	Seznam přihlášení, která mají povoleno zapisovat do datového bodu. Pokud není v seznamu, bude vstup zakázán.
------------------------	-------	--

---

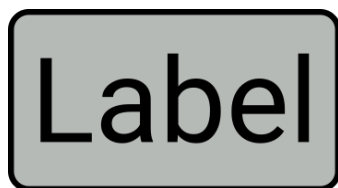
## Led



Indikátor stavu. Prázdná nebo nulová hodnota znamená VYPNUTO, jinak ZAPNUTO. Toto lze přepnout pomocí parametru `reverseMeaning`.

Data	Volitelné	Popis
<code>value</code>		Datapoint, ze kterého se čte hodnota.
<code>error_by</code>	True	Pokud je nastaveno a TRUE, LED zobrazí <code>colorError</code>
<code>disable_by</code>	True	Komponenta je zakázána, pokud hodnota není nulová a data se již neaktualizují.
<code>hide_by</code>	True	Komponenta je skryta, pokud hodnota není nulová.
Vlastnost	Typ	Popis
<code>reverseMeaning</code>	bool	Pokud je True, význam hodnoty je přepnut. Takže stav ZAPNUTO je False a stav VYPNUTO je True.
<code>colorTrue</code>	color	LED pozadí, když je signál ZAPNUTO, PŘIPOJENO, ...
<code>colorFalse</code>	color	LED pozadí, když je signál VYPNUTO, ZAKÁZÁNO, ...
<code>colorError</code>	color	LED pozadí, když je chyba signálu TRUE.

## LedLabel



Indikátor stavu s textovým popisem. Stavů definujete pomocí tabulky `values`. Pokud není barva použita, nechte pole prázdné (např. barva rámečku)

Data	Volitelné	Popis
<code>value</code>		Datapoint, ze kterého se čte stav LED.
<code>hide_by</code>	True	Komponenta je skryta, pokud hodnota není nulová.
<code>disable_by</code>	True	Komponenta je zakázána, pokud hodnota není nulová a data se již neaktualizují.
Vlastnost	Typ	Popis
<code>values</code>	table	Seznam stavů a jejich barev, které se zobrazí při změně <code>value</code> . Priorita je vzestupná. Pokud je některá položka v tabulce <code>values</code> prázdný řetězec, vzhled se nemění.
<code>line_height</code>	str	Výška řádku víceřádkového textu použitá pro dy atribut <code>tspan</code> . TENTO parametr se použije pouze v případě, že není dostatečný počet <code>tspan</code> uzlů.

### PushOnOff



Tlačítko `PushOnOff` mění hodnotu na datovém bodě. Normálně je ON stav `TRUE` a OFF stav `FALSE`. Pokud je potřeba, parametr `reverseMeaning` obrátí zapisované hodnoty.

Existují tři různé režimy provozu:

`PushButton` když je stisknuto nebo dotknuto, je zapsán stav ON do cílového datového bodu `value`. Po uvolnění je zapsán stav OFF.

`ManualPulse` vždy zapíše hodnotu `TRUE` po stisknutí tlačítka, uvolnění je ignorováno. (Vhodné pro řízení bloku MP)

`ToggleButton` když je tlačítko stisknuto, hodnota se přepne (ON -> OFF -> ON).

Data	Volitelné	Popis
<code>value</code>		Datový bod, kam se zapisuje boolean hodnota.
<code>refresh_from</code>	True	Pokud je nastaveno, stav komponenty se aktualizuje na základě hodnoty přečtené z tohoto datového bodu. Jinak se použije hodnota z hlavního datového bodu.

<code>hide_by</code>	True	Komponenta je skryta, pokud hodnota není nulová.
<code>disable_by</code>	True	Komponenta je zakázána, pokud hodnota není nulová a data se již neaktualizují.
<code>readonly_by</code>	True	Komponenta přejde do režimu pouze pro čtení, pokud hodnota tohoto parametru není nulová.
Vlastnost	Typ	Popis
<code>type</code>	enum	Změní typ tlačítka. <code>PushButton</code> když je stisknuto nebo dotknuto, je zapsán stav ON do cílového datového bodu <code>value</code> . Po uvolnění je zapsán stav OFF. <code>ManualPulse</code> zapíše hodnotu TRUE po stisknutí tlačítka, uvolnění je ignorováno. <code>ManualPulseRpt</code> zapíše hodnotu TRUE periodicky (20ms perioda) po stisknutí tlačítka, uvolnění zastaví zápis. (Vhodné pro řízení bloku MP) <code>ToggleButton</code> když je tlačítko stisknuto, hodnota se přepne (ON -> OFF -> ON).
<code>reverseMeaning</code>	bool	Pokud je True, význam hodnoty je přepnut. Takže stav ON je False a stav OFF je True.
<code>colorFalse</code>	color	Barva pozadí tlačítka, když je stav OFF
<code>colorTrue</code>	color	Barva pozadí tlačítka, když je stav ON
<code>writePerm</code>	table	Seznam přihlášení, která mají povoleno zapisovat do datového bodu. Pokud nejsou v seznamu, bude tlačítko zakázáno.

## SimpleLogger



Jednoduchá komponenta pro zápis seznamu oddělených řádků / logů. Může být použita ve dvou režimech na základě parametru `format`. Každý řádek může být předřazen časovým razítkem pomocí lokálního (klientského) času. Celý log je omezen na maximální počet `lines`. Parametr `format` má dvě možnosti:

`text` - Jednoduchý log, když se změní hodnota datového bodu `value`.

`alt` - Když se změní hodnota datového bodu `value`, je v tabulce `texts` vyhledána popis pro hodnotu. Pokud není nalezena, je zaznamenána hodnota. Hodnota je předřazena popisu, pokud je povoleno `showValue`.

Data	Volitelné	Popis
------	-----------	-------

<code>value</code>		Hodnota z datového bodu bude zaznamenána v logu.
<code>disable_by</code>	True	Komponenta je zakázána, pokud hodnota není nulová a data se již neaktualizují.
<code>hide_by</code>	True	Komponenta je skryta, pokud hodnota není nulová.
Vlastnost	Typ	Popis
<code>format</code>	enum	Formát zaznamenané zprávy. Buď čistý <code>text</code> nebo <code>alt</code> tabulka pro vyhledávání analogových hodnot.
<code>texts</code>	table	Tabulka pro vyhledávání analogových hodnot a jejich popisů ( <code>desc</code> ). Dostupné pro <code>alt format</code> .
<code>lines</code>	int	Maximální počet řádků v logu.
<code>showValue</code>	bool	Pokud je povoleno, hodnota je předřazena před popisem z tabulky <code>texts</code> .
<code>ignoreEmptyValues</code>	bool	Pokud je povoleno, logger ignoruje všechny hodnoty bez popisu.
<code>timestamp</code>	bool	Pokud je povoleno, je zobrazeno lokální časové razítka klienta.
<code>timestampFormat</code>	enum	Formát řetězce časového razítka. Viz GlobalizeJS knihovna pro další formáty.
<code>useTargetTime</code>	bool	Pokud je povoleno, hodnota časového razítka je přečtena z cíle.
<code>css</code>	object	Vlastní CSS styl pro HTML logger jako JSON objekt. Například <code>{'background-color': '#00ffff'}</code>

## SliderHorizontal



Posuvník nastavuje celočíselnou nebo desetinnou hodnotu cílovému bodu. Rozsah posuvníku je omezen mezi hodnotami `min` a `max`. Pohyb posuvníku je rozdělen na několik `stepů`. Hodnota `value` může být převedena pomocí `scale` a `offset` a zaokrouhlena na zvolený počet `decimals`. `fontScale` se vztahuje na zobrazenou hodnotu. Každý posuvník má `label`.

Data	Volitelné	Popis
<code>value</code>		Hlavní bod, kam je zapsána číselná hodnota.
<code>active</code>	True	Pokud je nastaveno, posuvník periodicky zapisuje hodnotu TRUE do datového bodu, když je posuvník stisknut nebo se pohybuje.
<code>refresh_from</code>	True	Pokud je nastaveno, stav komponenty je aktualizován na základě hodnoty přečtené z tohoto datového bodu. Jinak se použije hodnota z hlavního datového bodu.
<code>disable_by</code>	True	Komponenta je zakázána, pokud hodnota není nulová a data se již neaktualizují.

<code>hide_by</code>	True	Komponenta je skryta, pokud hodnota není nulová.
<code>readonly_by</code>	True	Komponenta přejde do režimu pouze pro čtení, pokud hodnota tohoto parametru není nulová.
Vlastnost	Typ	Popis
<code>min</code>	float	Minimální hodnota posuvníku.
<code>max</code>	float	Maximální hodnota posuvníku.
<code>step</code>	float	Velikost kroku (inkrement) při pohybu posuvníku.
<code>label</code>	str	Textový popis.
<code>fontScale</code>	float	Změna velikosti písma zobrazené hodnoty. Pokud je nastaveno na 1, velikost písma je výchozí. Desetinné číslo větší než 1 zvětší velikost písma.
<code>scale</code>	float	Změní měřítko přečtené hodnoty. Rovnice je $scale * val + offset$ . Číslo je převedeno zpět při zápisu.
<code>offset</code>	float	Přidá offset k přečtené hodnotě. Rovnice je $scale * val + offset$ . Číslo je převedeno zpět při zápisu.
<code>decimals</code>	int	Zaokrouhlí číslo při čtení.
<code>writeOnChange</code>	bool	Pokud je pravda, hodnota je zapsána do cíle, když se posuvník pohybuje.

## SliderVertical



Posuvník nastavuje celočíselnou nebo desetinnou hodnotu cílovému bodu. Rozsah posuvníku je omezen mezi hodnotami `min` a `max`. Pohyb posuvníku je rozdělen na několik `stepů`. Hodnota `value` může být převedena pomocí `scale` a `offset` a zaokrouhlena na zvolený počet `decimals`. `fontScale` se vztahuje na zobrazenou hodnotu. Každý posuvník má `label`.

Data	Volitelné	Popis
<code>value</code>		Hlavní bod, kam je zapsána číselná hodnota.
<code>active</code>	True	Pokud je nastaveno, posuvník periodicky zapisuje hodnotu TRUE do bodu, když je posuvník stisknut nebo se pohybuje.



<code>refresh_from</code>	True	Pokud je nastaveno, stav komponenty je aktualizován na základě hodnoty přečtené z tohoto datového bodu. Jinak se použije hodnota z hlavního datového bodu.
<code>disable_by</code>	True	Komponenta je zakázána, pokud hodnota není nulová a data se již neaktualizují.
<code>hide_by</code>	True	Komponenta je skryta, pokud hodnota není nulová.
<code>readonly_by</code>	True	Komponenta přejde do režimu pouze pro čtení, pokud hodnota tohoto parametru není nulová.
Vlastnost	Typ	Popis
<code>min</code>	float	Minimální hodnota posuvníku.
<code>max</code>	float	Maximální hodnota posuvníku.
<code>step</code>	float	Velikost kroku (inkrement) při pohybu posuvníku.
<code>label</code>	str	Textový popis.
<code>fontScale</code>	float	Změna velikosti písma zobrazené hodnoty. Pokud je nastaveno na 1, velikost písma je výchozí. Desetinné číslo větší než 1 zvětší velikost písma.
<code>scale</code>	float	Změna měřítka přečtené hodnoty. Použitelné pouze pro formát <code>number</code> .
<code>offset</code>	float	Přidá offset k přečtené hodnotě. Použitelné pouze pro formát <code>number</code> .
<code>decimals</code>	int	Zaokrouhlit přečtené číslo.
<code>writeOnChange</code>	bool	Pokud je nastaveno na true, hodnota je zapsána do cíle, když se posuvník pohybuje.

## Switch

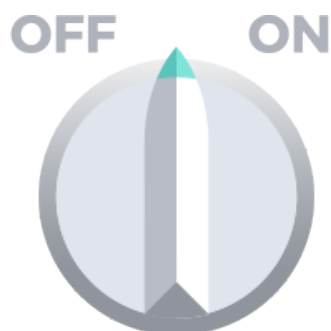


Spínač mění hodnoty z předdefinovaného seznamu. Každá pozice zapisuje definovanou hodnotu do cílového zařízení. Kliknutím nebo klepnutím změňte pozici ve směru hodinových ručiček (CW). Dvojklikem nebo dvojným klepnutím změňte pozici proti směru hodinových ručiček (CCW).

Data	Volitelné	Popis
<code>value</code>		Datapoint, kam se zapisuje hodnota.

<code>refresh_from</code>	True	Pokud je nastaveno, stav komponenty je aktualizován na základě hodnoty přečtené z tohoto datového bodu. Jinak se použije hodnota z hlavního datového bodu.
<code>disable_by</code>	True	Komponenta je zakázána, pokud hodnota není nulová a data se již neaktualizují.
<code>hide_by</code>	True	Komponenta je skryta, pokud hodnota není nulová.
<code>readonly_by</code>	True	Komponenta přejde do režimu pouze pro čtení, pokud hodnota tohoto parametru není nulová.
Vlastnost	Typ	Popis
<code>positions</code>	table	Seznam pozic spínače s hodnotami.
<code>hideTooltips</code>	bool	Pokud je TRUE, tak se nezobrazují tooltips s hodnotami na každé pozici.

### SwitchOnOff



Přepínač zapnuto/vypnuto

Data	Volitelné	Popis
<code>value</code>		Datový bod, do kterého se zapisuje logická hodnota.
<code>refresh_from</code>	True	Pokud je nastaveno, stav komponenty je aktualizován na základě hodnoty přečtené z tohoto datového bodu. Jinak se použije hodnota z hlavního datového bodu.
<code>hide_by</code>	True	Komponenta je skryta, pokud hodnota není nulová.
<code>disable_by</code>	True	Komponenta je zakázána, pokud hodnota není nulová a data se již neaktualizují.
<code>readonly_by</code>	True	Komponenta přejde do režimu pouze pro čtení, pokud hodnota tohoto parametru není nulová.
Vlastnost	Typ	Popis
<code>reverse_meaning</code>	bool	Pokud je True, význam hodnoty je přepnut. Takže stav ON je False a stav OFF je True.
<code>labelColorFalse</code>	color	Barva popisku (textu) pokud je stav OFF.
<code>labelColorTrue</code>	color	Barva popisku (textu) pokud je stav ON.

<code>writePerm</code>	table	Seznam přihlašovacích jmen, která mají povolen zápis do datového bodu. Pokud není v seznamu, bude tlačítko zakázáno.
------------------------	-------	--

---

## SwitchOnOff2



Komponenta přepínač mění hodnotu typu boolean. Obvykle je stav ON roven hodnotě TRUE a stav OFF hodnotě FALSE. Pokud je potřeba, parametr `reverseMeaning` obrátí zapsané hodnoty.

Data	Volitelné	Popis
<code>value</code>		Hlavní datový bod, do kterého je zapisována hodnota typu boolean.
<code>refresh_from</code>	True	Pokud je nastaveno, stav komponenty je aktualizován na základě hodnoty přečtené z tohoto datového bodu. Jinak se použije hodnota z hlavního datového bodu.
<code>hide_by</code>	True	Komponenta je skryta, pokud hodnota není nulová.
<code>disable_by</code>	True	Komponenta je zakázána, pokud hodnota není nulová.
<code>readonly_by</code>	True	Komponenta přejde do režimu pouze pro čtení, pokud hodnota tohoto parametru není nulová.
Vlastnost	Typ	Popis
<code>reverseMeaning</code>	bool	Pokud je True, význam hodnoty je přepnut. Stav ON je tedy False a stav OFF je True.
<code>label</code>	str	Text, který je zapsán vedle přepínače
<code>writePerm</code>	table	Seznam přihlašovacích jmen, která mají povolen zápis do datového bodu. Pokud není v seznamu, bude tlačítko zakázáno.

---

## TRND

Real time trend  
from TRND block

Komponenta TRND slouží pro zobrazení signálů z bloku TRND. Stačí nastavit TRND datový bod na libovolný parametr bloku TRND např. `taskname.TRND:u1`. Všechny signály budou zobrazeny v trendu. Každý signál lze pojmenovat pomocí tabulky `signals`.

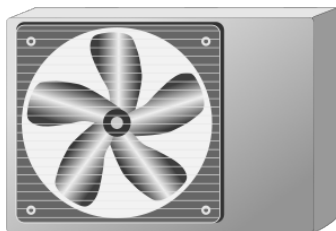
Data	Volitelné	Popis
TRND		Nastavte datový bod na libovolný parametr bloku TRND např. <code>taskname.TRND:u1</code> pro povolení čtení všech signálů.
ViewConfig	True	Pokud je parametr TRUE, konfigurace grafu je načtena ze řetězce JSON. Pokud není nastaven, je konfigurace načtena z bloku TRND. Konfigurace je načtena jednou po načtení stránky. Obnovení musí být provedeno ručně z menu.
reset_by	True	Pokud není nulová, data z trendu jsou smazána.
hide_by	True	Komponenta je skryta, pokud hodnota není nulová.
disable_by	True	Komponenta je zakázána, pokud hodnota není nulová a data se již neaktualizují.
Vlastnost	Typ	Popis
title	str	Název trendu
signals	table	Tabulka popisků signálů. Zadejte popisky ve stejném pořadí, jako jste signály připojili k bloku TRND.
bufferSize	int	Velikost bufferu v prohlížeči. Pokud je -1, použijte se velikost bufferu na cílovém bloku TRND. <i>Poznámka: pokud je buffer větší než buffer na zařízení, je historie ztracena při obnovení stránky.</i> <i>Poznámka: Velký buffer může zpomalit vaše zařízení, zejména na pomalejších CPU, jako je RaspberryPi</i>
refreshRate	int	Vlastní obnovovací frekvence trendu. Obvykle je trend obnovován s periodou hlavní obnovy HMI. Pokud je to nutné, může mít každý trend vlastní.
hideLegend	bool	Skrýt legendu grafu.
hideRangeSelector	bool	Skrýt selektor rozsahu pod grafem.
yAxis	table	Tabulka základních parametrů pro osu Y. Použijte se pouze první řádek!

`dygraphOptions` object Vlastní možnosti pro knihovnu Dygraph. Více informací naleznete na <http://dygraphs.com/options.html>.

---

### 3.4.4 HOME

#### AirCirculator



Data	Volitelné	Popis
<code>POWER</code>		Pokud není nula, komponenta je zapnuta.
Vlastnost	Typ	Popis
<code>colorRun</code>	color	Barva komponenty, když je zapnuta / běží.
<code>colorStop</code>	color	Barva komponenty, když je vypnuta / zastavena.

#### Boiler



Kotel ukazuje postup vytápění. Pokud je `currentTemp` menší než `setTemp` jsou viditelné plameny.

Data	Volitelné	Popis
<code>POWER</code>		Pokud není nula, komponenta je zapnutá.
<code>setTemp</code>		Cílová teplota kotle.
<code>currentTemp</code>		Aktuální teplota.
Vlastnost	Typ	Popis

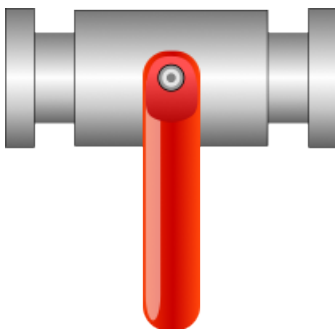
## Filter



Filtr ukazuje, zda je filtrace v provozu.

Data	Volitelné	Popis
POWER		Pokud není nula, komponenta je zapnuta.
Vlastnost	Typ	Popis
colorOn	color	Barva komponenty, když je zapnuta / běží.
colorOff	color	Barva komponenty, když je vypnuta / zastavena.

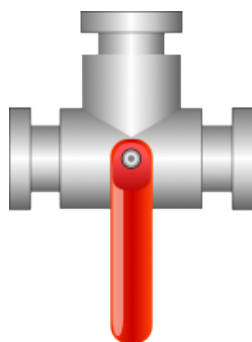
## HandleValve



Ovládání ventilu. Přepíná ventil po kliknutí.

Data	Volitelné	Popis
FLOW		Stav ventilu.
refresh_from	True	Pokud je nastaveno, stav je čten z tohoto bodu.
Vlastnost	Typ	Popis

## HandleValveT



Třístavový ventil. Po kliknutí přepíná dvě boolean hodnoty.

Data	Volitelné	Popis
FLOW1		Stav ventilu.
FLOW2		Stav ventilu.
refresh_from	True	Pokud je nastaveno, stav ventilu je čten z tohoto bodu. Očekává se celé číslo 0 - <i>neznámý</i> , 1 - <i>True-False stav</i> , 2 - <i>False-True stav</i> .
Vlastnost	Typ	Popis

### Heater



Zobrazuje stav topení a aktuální teplotu. Barva se mění z `colorMin` když je hodnota menší než `colorSignalMin` a `colorMax` když je hodnota větší než `colorSignalMax`.

Data	Volitelné	Popis
POWER		Pokud není nula, topení je aktivní.
currentTemp		Aktuální teplota.
Vlastnost	Typ	Popis
heatingColor	color	Barva topné spirály když je aktivní.
colorMin	color	Barva média když je aktuální teplota nízká.
colorMax	color	Barva média když je aktuální teplota vysoká.
colorSignalMin	float	Minimální hodnota teploty.

colorSignalMax float Maximální hodnota teploty.

---

### Motor



Indikátor stavu motoru.

---

Data	Volitelné	Popis
POWER		Pokud není nula, motor je aktivní.
Vlastnost	Typ	Popis
colorRun	color	Barva, když je motor aktivní / běží.
colorStop	color	Barva, když je motor neaktivní / zastaven.

---

### PipeElbow



Trubka mění barvu podle signálu **COLOR**.

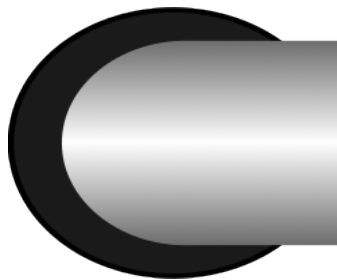
---

Data	Volitelné	Popis
COLOR		Hodnota pro změnu barvy (např. teplota trubky)
Vlastnost	Typ	Popis
colorMin	color	Barva, když je teplota pod minimem.
colorMax	color	Barva, když je teplota nad maximem.
colorSignalMin	float	Minimální teplota.
colorSignalMax	float	Maximální teplota.

---

### PipeEnding





Trubka mění barvu podle signálu `COLOR`.

Data	Volitelné	Popis
<code>COLOR</code>		Hodnota pro změnu barvy (např. teplota trubky)
Vlastnost	Typ	Popis
<code>colorMin</code>	color	Barva, když je teplota pod minimem.
<code>colorMax</code>	color	Barva, když je teplota nad maximem.
<code>colorSignalMin</code>	float	Minimální teplota.
<code>colorSignalMax</code>	float	Maximální teplota.

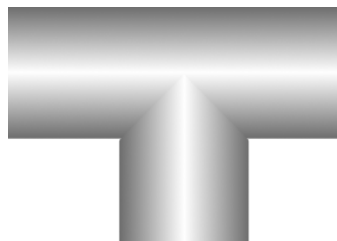
### PipeStraight



Trubka mění barvu podle signálu `COLOR`.

Data	Volitelné	Popis
<code>COLOR</code>		Hodnota pro změnu barvy (např. teplota trubky)
Vlastnost	Typ	Popis
<code>colorMin</code>	color	Barva, když je teplota pod minimem.
<code>colorMax</code>	color	Barva, když je teplota nad maximem.
<code>colorSignalMin</code>	float	Minimální teplota.
<code>colorSignalMax</code>	float	Maximální teplota.

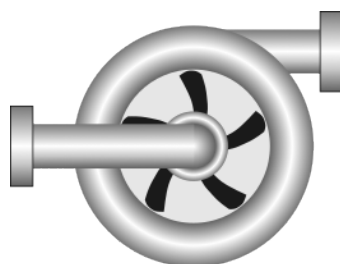
### PipeT



Trubka mění barvu podle signálu **COLOR**.

Data	Volitelné	Popis
<b>COLOR</b>		Hodnota pro změnu barvy (např. teplota trubky)
Vlastnost	Typ	Popis
<b>colorMin</b>	color	Barva, když je teplota pod minimem.
<b>colorMax</b>	color	Barva, když je teplota nad maximem.
<b>colorSignalMin</b>	float	Minimální teplota.
<b>colorSignalMax</b>	float	Maximální teplota.

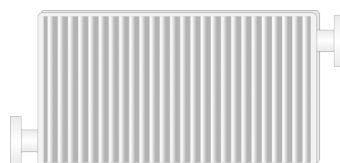
## Pump



Indikátor stavu čerpadla.

Data	Volitelné	Popis
<b>POWER</b>		Pokud není nula, čerpadlo je aktivní.
Vlastnost	Typ	Popis
<b>colorRun</b>	color	Barva, když je čerpadlo aktivní / běží.
<b>colorStop</b>	color	Barva, když je čerpadlo neaktivní / zastaveno.

## Radiator



Indikátor stavu radiátoru.

Data	Volitelné	Popis
<b>POWER</b>		Indikátor stavu radiátoru.
<b>currentTemp</b>		Aktuální teplota.
Vlastnost	Typ	Popis

<code>inOutColor</code>	color	Barva vstupu a výstupu, když je radiátor aktivní.
<code>colorMin</code>	color	Barva radiátoru, když je <code>currentTemp</code> menší než <code>colorSignalMin</code> .
<code>colorMax</code>	color	Barva radiátoru, když je <code>currentTemp</code> větší než <code>colorSignalMax</code> .
<code>colorSignalMin</code>	float	Minimální hodnota teploty.
<code>colorSignalMax</code>	float	Maximální hodnota teploty..

---

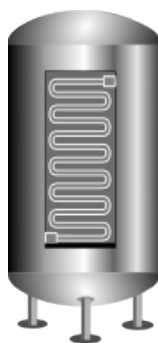
### Shower



Sprcha

Data	Volitelné	Popis
<b>POWER</b>		
Vlastnost	Typ	Popis

### WaterBoiler



Komponenta zobrazuje stav ohřevu a chlazení vody. Stav závisí na hodnotě `power`: 1 - ohřev  
0 - vypnuto  
-1 - chlazení

Data	Volitelné	Popis
<code>power</code>		Stav ohřevu a chlazení vody: 1 - ohřev 0 - vypnuto -1 - chlazení
<code>currentTemp</code>		
Vlastnost	Typ	Popis
<code>heatingColor</code>	color	Barva spirály při ohřevu vody.
<code>coolingColor</code>	color	Barva spirály při chlazení vody.
<code>colorMax</code>	color	Barva při překročení maximální teploty.
<code>colorMin</code>	color	Barva při překročení minimální teploty.
<code>colorSignalMax</code>	float	Maximální teplota.
<code>colorSignalMin</code>	float	Minimální teplota.

### WaterTank

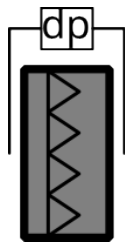


Ukazuje naplnění nádrže.

Data	Volitelné	Popis
<code>level</code>		Úroveň kapaliny v nádrži.
Vlastnost	Typ	Popis
<code>capacity</code>	float	Maximální kapacita.
<code>colorOfLevel</code>	color	Barva obsahu.

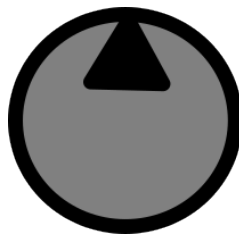
### 3.4.5 HVAC

#### AirFilter



Data	Volitelné	Popis
DIRTY		Pokud není nula, je filtr vzduchu špinavý.
Vlastnost	Typ	Popis

### Circulator



Cirkulace.svg

Data	Volitelné	Popis
ENABLE OHEAT		
Vlastnost	Typ	Popis

### ClosingDamperDown



Zavírací klapka dolní

Data	Volitelné	Popis
POSITION		

Vlastnost	Typ	Popis
open_time	int	

### ClosingDamperLeft



Zavírací klapka vlevo

Data	Volitelné	Popis
POSITION		
Vlastnost	Typ	Popis
open_time	int	

### ClosingDamperRight



Zavírací klapka vpravo

Data	Volitelné	Popis
POSITION		
Vlastnost	Typ	Popis
open_time	int	

### ClosingDamperUp



Zavírací klapka horní

Data	Volitelné	Popis
POSITION		
Vlastnost	Typ	Popis
open_time	int	

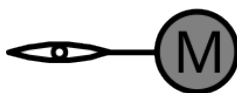
### DamperDown



DamperDown.svg

Data	Volitelné	Popis
POSITION		
Vlastnost	Typ	Popis

### DamperLeft



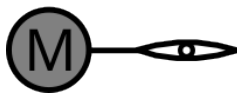
DamperLeft.svg

Data	Volitelné	Popis

POSITION

Vlastnost	Typ	Popis
-----------	-----	-------

**DamperRight**



DamperRight.svg

Data	Volitelné	Popis
------	-----------	-------

POSITION

Vlastnost	Typ	Popis
-----------	-----	-------

**DamperUp**



DamperUp.svg

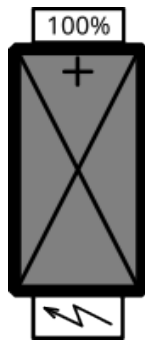
Data	Volitelné	Popis
------	-----------	-------

POSITION

Vlastnost	Typ	Popis
-----------	-----	-------

**ElAirHeater**

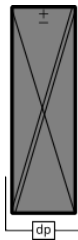




ElAirHeater.svg

Data	Voliteľné	Popis
CIRCUITBREAKER		
ENABLE		
POWER		
Vlastnosť	Typ	Popis

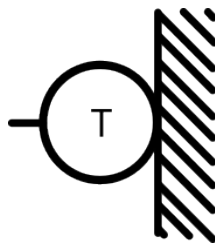
### Exchanger



Exchanger.svg

Data	Voliteľné	Popis
ICE		
Vlastnosť	Typ	Popis

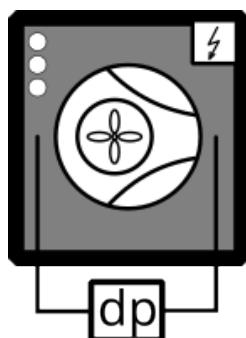
### ExternalThermometer



## Externí teploměr

Data	Volitelné	Popis
temp		
Vlastnost	Typ	Popis

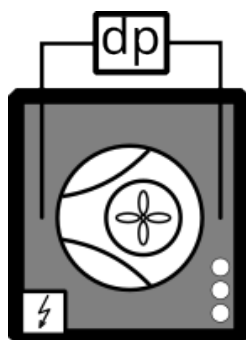
### Fan



## Ventilátor

Data	Volitelné	Popis
L1		
L2	True	
L3	True	
OHEAT		
RUNNING		
Vlastnost	Typ	Popis

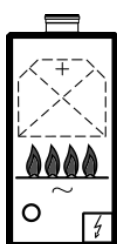
### FanLeft



FanLeft.svg

Data	Voliteľné	Popis
L1		
L2	True	
L3	True	
OHEAT RUNNING		
Vlastnosť	Typ	Popis

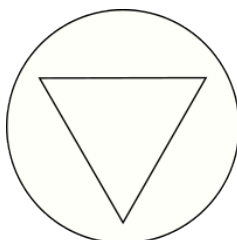
### GasBoiler



GasBoiler.svg

Data	Voliteľné	Popis
ERROR POWER		
Vlastnosť	Typ	Popis
flame1_on	float	
flame2_on	float	
flame3_on	float	
flame4_on	float	
flame_color	color	
innerFlame_color	color	
led_color	color	

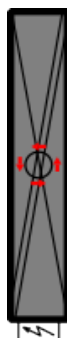
### Pump



Čerpadlo

Data	Volitelné	Popis
ON		
Vlastnost	Typ	Popis
on_color	color	

### RotaryExchanger



RotaryExchanger.svg

Data	Volitelné	Popis
ENABLE		
ICE		
OHEAT		
POWER		
Vlastnost	Typ	Popis

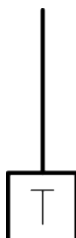
### Thermometer



Teplomer.svg

Data	Volitelné	Popis
temp		
Vlastnost	Typ	Popis

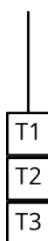
### Termostat



Termostat.svg

Data	Voliteľné	Popis
THERM		
Vlastnosť	Typ	Popis

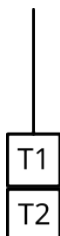
### TermostatThree



TermostatThree.svg

Data	Voliteľné	Popis
THERM		
THERMTHREE		
THERMTWO		
Vlastnosť	Typ	Popis

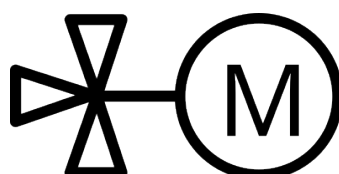
### TermostatTwo



TermostatTwo.svg

Data	Volitelné	Popis
THERM		
THERMTWO		
Vlastnost	Typ	Popis

### TValve

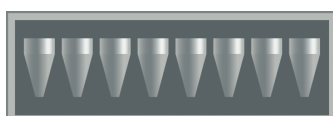


### T-Ventil

Data	Volitelné	Popis
OPEN		
Vlastnost	Typ	Popis
open_color	color	

## 3.4.6 PROC

### HydroCyclons



### HydroCyclons.svg

Data	Volitelné	Popis
value	True	Počet aktivních cyklonů
disable_by	True	Komponenta je zakázána, pokud hodnota není nulová a data se již neaktualizují.
hide_by	True	Komponenta je skryta, pokud hodnota není nulová.
Vlastnost	Typ	Popis

### HydroCyclons10



Grafická komponenta pro hydrocyklony

Data	Volitelné	Popis
<code>active_by</code>	True	Bitový maska, která nastavuje aktivní barvu pro příslušný hydrocyklon. Nejnižší bit odpovídá nejpravějšímu hydrocyklonu.
<code>blocked_by</code>	True	Bitový maska, která nastavuje blokovanou barvu pro příslušný hydrocyklon. Nejnižší bit odpovídá nejpravějšímu hydrocyklonu. Blok má vyšší prioritu než aktivní.
<code>disable_by</code>	True	Komponenta je zakázána, pokud hodnota není nulová a data se již neaktualizují.
<code>hide_by</code>	True	Komponenta je skryta, pokud hodnota není nulová.
Vlastnost	Typ	Popis
<code>colorActive</code>	color	Barva aktivního hydrocyklonu.
<code>colorBlocked</code>	color	Barva blokováného hydrocyklonu.

### HydroCyclons2



HydroCyclons2.svg

Data	Volitelné	Popis
<code>value</code>	True	Počet aktivních cyklonů.
<code>hide_by</code>	True	Komponenta je skryta, pokud hodnota není nulová.
<code>disable_by</code>	True	Komponenta je zakázána, pokud hodnota není nulová a data se již neaktualizují.
Vlastnost	Typ	Popis

### Pump



## Čerpadlo

Data	Volitelné	Popis
<code>active_by</code>	True	Čerpadlo je zapnuto, pokud hodnota není nulová.
<code>error_by</code>	True	Čerpadlo je v chybě, pokud hodnota není nulová.
<code>disable_by</code>	True	Komponenta je zakázána, pokud hodnota není nulová a data se již neaktualizují.
<code>hide_by</code>	True	Komponenta je skryta, pokud hodnota není nulová.
Vlastnost	Typ	Popis

## Reaktor



## Reaktor.svg

Data	Volitelné	Popis
<code>value</code>	True	Výška hladiny v nádrži.
<code>active_by</code>	True	Reaktor je zapnutý, pokud hodnota není nulová.
<code>error_by</code>	True	Reaktor je v chybě, pokud hodnota není nulová.
<code>disable_by</code>	True	Komponenta je zakázána, pokud hodnota není nulová a data se již neaktualizují.
<code>hide_by</code>	True	Komponenta je skryta, pokud hodnota není nulová.
Vlastnost	Typ	Popis



valueMin	float
valueMax	float

---

### ReactorSmall



Reaktor malý

Data	Volitelné	Popis
value	True	Výška reaktoru v nádrži.
active_by	True	Reaktor je zapnut, pokud hodnota není nulová.
error_by	True	Reaktor je v chybovém stavu, pokud hodnota není nulová.
disable_by	True	Komponenta je zakázána, pokud hodnota není nulová a data se již neaktualizují.
hide_by	True	Komponenta je skryta, pokud hodnota není nulová.
Vlastnost	Typ	Popis
valueMin	float	
valueMax	float	

### Tank

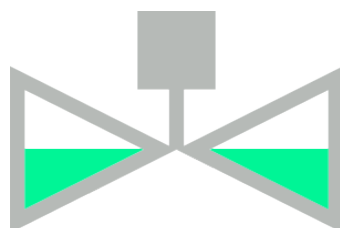


Tank.svg

Data	Volitelné	Popis
value	True	Výška hladiny v nádrži.
disable_by	True	Komponenta je zakázána, pokud hodnota není nulová a data se již neaktualizují.

hide_by	True	Komponenta je skryta, pokud hodnota není nulová.
Vlastnost	Typ	Popis
valueMin	float	
valueMax	float	

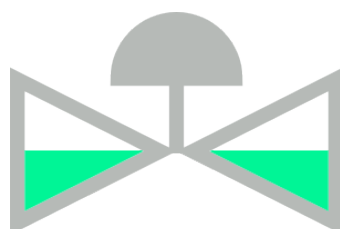
### Valve



Ventil.svg

Data	Volitelné	Popis
value	True	Výška hladiny v nádrži.
active_by	True	Ventil je otevřen, pokud hodnota není nulová.
error_by	True	Ventil je v chybovém stavu, pokud hodnota není nulová.
disable_by	True	Komponenta je zakázána, pokud hodnota není nulová a data se již neaktualizují.
hide_by	True	Komponenta je skryta, pokud hodnota není nulová.
Vlastnost	Typ	Popis
valueMin	float	
valueMax	float	

### Valve2



Ventil2.svg

Data	Volitelné	Popis
value	True	Výška hladiny v nádrži.
active_by	True	Pokud není nulová, ventil je otevřený.

<code>error_by</code>	True	Pokud není nulová, ventil je v chybovém stavu.
<code>disable_by</code>	True	Komponenta je zakázána, pokud hodnota není nulová a data se již neaktualizují.
<code>hide_by</code>	True	Komponenta je skryta, pokud hodnota není nulová.
Vlastnost	Typ	Popis
<code>valueMin</code>	float	
<code>valueMax</code>	float	

# Kapitola 4

## WebBuDi

### 4.1 Jednoduchá tlačítka a displeje na webu

WebBuDi je zkratka pro Web Buttons and Displays, což je jednoduchý JavaScriptový soubor s několika deklarativními bloky, které popisují datové body (datapointy), ke kterým je HMI připojeno. Z bloků je možné sestavit tabulku, ve které jsou všechna data prezentována. WebBuDi poskytuje textovou interakci s vybranými signály a hodí se pro vývojáře a integrátory systémů nebo může sloužit jako záložní režim HMI pro nestandardní situace.

WebBuDi je složen z několika **řádků** (*rows* - grafických komponent s předdefinovanou funkcí a vzhledem) připojených k jednotlivé položce v řídicím systému (specifikováno pomocí vlastnosti *alias* nebo *cstring*). Existují různé řádky podle typu, který mění (pro booleovské hodnoty, čísla, data atd.). Všechny **řádky** jsou organizovány v *sekcích* (barevné bloky, které mohou mít nadpis). Sekce jsou poté organizovány do několika sloupců.

Konfigurace je provedena pomocí JavaScriptových objektů. Viz [4.1](#) pro více detailů.

#### Příklad

```
1 REX.HMI.init = function(){
2     // Jednoduchy priklad PID regulatoru
3
4     // Volitelne - pridani polozky
5     REX.WebBuDi.addItems([
6         {alias: 'PID_MAN', cstring: 'pidcontrol_control.CNB_MAN:YCN', write:
7             true}
8     ]);
9
10    // Pridani sekce WebBuDi
11    REX.WebBuDi.addSection({
12        column: 1,
13        title: 'Ovladaci prvky pro uzivatele',
14        rows: [
15            {alias: 'PID_MAN', desc: 'Rezim regulatoru', type: 'DW', label_false:
16                'AUT', label_true: 'MAN'},
17            {type: 'ES'},
18            {alias: 'SP_AW', cstring: 'pidcontrol_control.CNR_sp:ycn', desc: '
19                Setpoint', type: 'AW'}
```

```

17     ]
18   });
19
20   // Zobrazení grafu z bloku TRND
21   REX.HMI.Graph.addSignal({cstring: 'pidcontrol_control.TRND_PIDU',
22     labels: ['Hodnota procesu', 'Manipulovaná proměnná', 'Setpoint']});
23
24   // Nastavení jiné cílové adresy
25   // REX.HMI.setTargetUrl('ws://192.168.1.100:8008/rex');
26
27   // Nastavení obnovovací frekvence (Vychozi: 500 ms)
28   REX.HMI.setRefreshRate(100);
29
30   // Změna názvu stránky
31   REX.HMI.setTitle('Jednoduchý PID regulátor');
32 }

```

`.WebBuDi` : objekt

- `.addSection(opt)` => `REX.HMI.WebBuDi`
- `.add()` => `REX.HMI.WebBuDi`
- `.addItem(items)` => `Pole.<REX.WS.Item>`

**REX.WebBuDi.addSection(opt)** => **REX.HMI.WebBuDi** Funkce `addSection` přidává nové řádky / HMI komponenty na webovou stránku.

Sekce obsahuje všechny komponenty definované v poli `rows` (řádky). Může mít `title` (titulek) zobrazený v záhlaví. Celá sekce může být řízena prostřednictvím položek `disable_by` a `hide_by`. Nakonec je sekce umístěna do `column` (indexovaného sloupce).

Volání funkcí může být řetězeno nebo voláno přes alias `REX.WebBuDi.add()`.

**Typ:** statická metoda [WebBuDi](#)

Parametr	Typ	Výchozí	Popis
<code>opt</code>			Hlavní konfigurační objekt
<code>opt.rows</code>	<code>Pole.&lt;RowOption&gt;</code>		Definice všech HMI komponent/řádků. Viz seznam pro více detailů
<code>[opt.title]</code>	Řetězec		Název sekce zobrazený v záhlaví.
<code>[opt.column]</code>	Číslo	1	Index sloupce (začíná od 1).
<code>[opt.background_color]</code>	Řetězec		Vlastní barva pozadí sekce.
<code>[opt.text_color]</code>	Řetězec		Vlastní barva textu záhlaví sekce
<code>[opt.disable_by]</code>	Řetězec   Objekt		Pokud je definováno 'aliasem' nebo objektem <code>{alias:"XXX", cstring:"XXX", reverse_meaning:false}</code> , změní se stav komponenty (povoleno / zakázáno).
<code>[opt.hide_by]</code>	Řetězec   Objekt		Pokud je definováno 'aliasem' nebo objektem <code>{alias:"XXX", cstring:"XXX", reverse_meaning:false}</code> , změní se viditelnost řádku.

Parametr	Typ	Výchozí	Popis
[opt.customDivID]	Řetězec		"content" ID elementu, kam budou připojeny všechny sloupce / sekce.

### Příklad

```

1 // Jednoduché HMI pro PIDU_Simple_PID_Controller
2
3 REX.WebBuDi.addSection({
4     column: 1,
5     title: 'Ovladaci prvky',
6     rows: [
7         // Digitalni zapis
8         {alias: 'PID_MAN', desc:'Rezim regulatoru', type: 'DW', label_false:
9           'AUT', label_true: 'MAN'},
10        // Analogovy zapis
11        {alias: 'SP', desc:'Zadani', cstring: 'pidcontrol_control.CNR_sp:ycn'
12          , type: 'AW'},
13        {type: 'ES'}, // Prazdny prostor
14        ],
15        hide_by:"",
16        disable_by:""
17    });

```

**REX.WebBuDi.add()** => **REX.HMI.WebBuDi** Zkratka pro funkci **REX.WebBuDi.addSection**.

**Typ:** statická metoda [WebBuDi](#)

**REX.WebBuDi.addItems(items)** => **Pole.<REX.WS.Item>** Přidá několik položek najednou. Je to užitečný způsob, jak definovat aliasy a používat je v různých rádcích.

**Typ:** statická metoda [WebBuDi](#)

Parametr	Typ	Popis
items	Pole. <Objekt>	Pole položek k registraci. Zkratka pro <b>REX.HMI.addItem()</b> .
items.alias	Řetězec	
items.cstring	Řetězec	
items.write	boolean	Nastavte na true, pokud je položka zapisovatelná

### Příklad

```

1 REX.WebBuDi.addItems([
2     {alias: 'PID_MAN', cstring: 'pidcontrol_control.CNB_MAN:YCN', write:
3       true},

```

```

3     {alias: 'SP_AW', cstring: 'pidcontrol_control.CNR_sp:ycn', write:
      true},
4     {alias: 'HV', cstring: 'pidcontrol_control.PIDU:hv'}
5 ];

```

## 4.2 Dostupné řádky a komponenty

- AnalogLookupTable ('ALT')
- Analog Write ('AW')
- Analog Read ('AR')
- Digital Read ('DR')
- Digital Write ('DW')
- Empty Space ('ES')
- Link Button ('LINK')
- Manual Pulse ('MP')
- Push Button ('PB')

**Obecné možnosti řádku** - Každý řádek je konfigurován s běžnými a specifickými vlastnostmi řádku. Následující objekt představuje běžnou část.

### Vlastnosti

Název	Typ	Popis
type	Řetězec	Typ komponenty / řádku (např. "AR")
alias	alias	UNIKÁTNÍ identifikace položky z řídicího systému (např. 'SP'). Nesmí obsahovat mezery nebo diakritiku
desc	Řetězec	Název komponenty zobrazený v popisu (např. "Setpoint"). Pokud není <code>alias</code> definován, použije se místo toho <code>desc</code> . <code>desc</code> je převeden na malá písmena, mezery jsou nahrazeny podtržítkem a všechny ne-ascii znaky vynechány nebo nahrazeny ascii ekvivalentem.
cstring	Řetězec	Připojovací řetězec obsahující celou cestu k cílovému zařízení (např. "task.CNR:ycn")
disable_by	Řetězec   Objekt	Pokud je definováno 'aliasem' nebo objektem <code>{alias:"XXX", cstring:"XXX", reverse_meaning:false}</code> , změní se stav komponenty (povoleno / zakázáno).
hide_by	Řetězec   Objekt	Pokud je definováno 'aliasem' nebo objektem <code>{alias:"XXX", cstring:"XXX", reverse_meaning:false}</code> , změní se viditelnost řádku.
refresh_from	Řetězec	Pokud je definováno 'aliasem' nebo objektem <code>{alias:"XXX", cstring:"XXX"}</code> , hodnota položky je čtena z jiného místa než zapisována. Je to aplikovatelné pouze pro komponenty WRITE

```
1 // Vsechny vlastnosti
```

```
2 {alias:"SP", desc:"Setpoint", cstring:"task.CNR:ycn", disable_by:null,
  hide_by:null refresh_from:null}
```

### 4.2.1 AnalogLookupTable

**AnalogLookupTable** AnalogLookupTable ('ALT') - Výběr z možností. Používá se pro uživatelsky definované enumy.

Vlastnost	Typ	Popis
show_key	bool	Zobrazit klíče v seznamu
values	table	Seznam všech hodnot, které mohou být vybrány. Je to objekt s klíčovými hodnotami (např. '{"1":"Options 1", "2":"Options 2"}')

#### Příklad

```
1 {type:'ALT', values:{"1":"Options 1", "2":"Options 2"}, show_key: false}
```

### 4.2.2 AnalogReadRow

**AnalogReadRow** Analogové čtení ('AR') - Periodické čtení vybrané hodnoty ('date', 'time', 'datetime', 'text', 'date', 'datetime' - Zobrazí datum vypočítané ze sekund od REX Epoch timestamp. 'time' - Zobrazí čas vypočítaný ze sekund. 'text' - Zobrazí hodnotu bez jakékoliv transformace (vhodné pro řetězcové hodnoty). 'number' - Zobrazí číslo v měřítku 'scale', posunuté o 'offset' a zaokrouhlené na počet desetinných míst 'decimals'.

Vlastnost	Typ	Popis
format	bool	Jedna z následujících 'date', 'time', 'datetime', 'text', 'number'
scale	int	Měřítko
offset	int	Posunutí zobrazené hodnoty
decimals	int	Počet desetinných míst
convert	function	Pokud je definováno, hodnota 'format='number' je transformována pomocí funkce 'convert=function(val){return val+1;}'

#### Příklad

```
1 // Show number rounded to 4 decimal places
2 {type:'AR'}
3
4 // Show date
5 {type:'AR', format:'date'}
6
7 // Show string values
```



```

8 {type: 'AR', format: 'text'}
9
10 // Convert radians to degrees
11 {type: 'AR', format: 'number', scale:(Math.PI/180), offset:0, decimals:0}

```

### 4.2.3 AnalogWriteRow

**AnalogWriteRow** Analog Write ('AW') - Nastaví 'date', 'time', 'datetime', 'text' nebo 'number' do řídicího systému. 'date', 'datetime' - R/W datum vypočítané ze sekund od REX Epoch timestamp. 'time' - R/W čas vypočítaný ze sekund. 'text' - Zapiše hodnotu bez jakékoliv transformace (vhodné pro řetězcové hodnoty). 'number' - R/W číslo v měřítku 'scale', posunutě o 'offset' a zaokrouhlené na počet desetinných míst 'decimals'.

Vlastnost	Typ	Popis
format	bool	Jedna z následujících 'date', 'time', 'datetime', 'text', 'number'
scale	int	Měřítko
offset	int	Posunutí zobrazené hodnoty
min	int	Omezení minimální hodnoty
max	int	Omezení maximální hodnoty
decimals	int	Počet desetinných míst
set_on_blur	bool	Pokud je nastaveno, hodnota je zapsána po ztrátě fokusu. Použijte *ESC* pro zrušení změn
convert	function	Pokud je definováno, je hodnota 'format='number' transformována pomocí funkce convert např. 'convert=function(val){return val+1;}'
convertW	function	Pokud je definováno, je hodnota 'format='number' transformována před zápisem pomocí funkce convert např. 'convertW=function(val){return val+1;}'

#### Příklad

```

1 // Change number rounded to 4 decimal places
2 {type: 'AW'}
3
4 // Set date
5 {type: 'AW', format: 'date'}
6
7 // Set string values
8 {type: 'AW', format: 'text'}
9
10 // Display degrees, read and write as radians
11 {type: 'AW', format: 'number', scale:(Math.PI/180), offset:0, decimals:0}
12
13 // Default options
14 {type: 'AW', format: 'number',
15   scale:1, offset:0, min: -Number.MAX_VALUE, max: Number.MAX_VALUE,

```

```
16 set_on_blur:false, convert:null, convertW:null}
```

#### 4.2.4 DigitalReadRow

**DigitalReadRow** Digitální vstup ('DR') - Periodické čtení logické hodnoty.

Vlastnost	Typ	Popis
label_false	string	Popisek pro hodnotu false / off / disable.
label_true	string	Popisek pro hodnotu true / on / enable.
label_color_false	string	Barva popisku pro hodnotu false / off / disable.
label_color_true	string	Barva popisku pro hodnotu true / on / enable.
reverse_meaning	bool	Pokud je nastaveno, tak '0' (nula) znamená zapnuto / ON a '1' vypnuto / OFF
color_false	string	Změní barvu tlačítka FALSE, když je aktivní
color_true	string	Změní barvu tlačítka TRUE, když je aktivní

##### Příklad

```
1 {alias: 'DR', desc:'Controller mode', type: 'DR', label_false: 'AUT',
  label_true: 'MAN'}
2
3 // All options
4 {type: 'DR', label_false: 'OFF(0)', label_true: 'ON(1)', reverse_meaning:
  false, color_false:'', color_true:''}
```

#### 4.2.5 DigitalWriteRow

**DigitalWriteRow** Digital Write ('DW') - Nastavuje boolean hodnotu

Vlastnost	Typ	Popis
label_false	string	Popisek pro hodnotu false / off / disable.
label_true	string	Popisek pro hodnotu true / on / enable.
label_color_false	string	Barva popisku pro hodnotu false / off / disable.
label_color_true	string	Barva popisku pro hodnotu true / on / enable.
reverse_meaning	bool	Pokud je nastaveno, tak '0' (nula) znamená zapnuto / ON a '1' vypnuto / OFF
color_false	string	Změní barvu tlačítka FALSE, když je aktivní
color_true	string	Změní barvu tlačítka TRUE, když je aktivní
flip	bool	Pokud je nastaveno, tak se prohodí pozice tlačítek TRUE/FALSE.

##### Příklad

```
1 {alias: 'DW', desc:'Controller mode', type: 'DW', label_false: 'AUT',
  label_true: 'MAN'}
```

```

2
3 // All options
4 {type: 'DW', label_false: 'OFF(0)', label_true: 'ON(1)', reverse_meaning:
    false, color_false:'', color_true:'', flip:false}

```

#### 4.2.6 EmptySpace

**EmptySpace** Prázdný prostor ('ES') - Vytvoří prázdný řádek pro vyplnění mezer

Vlastnost	Typ	Popis
-----------	-----	-------

##### Příklad

```
1 {type: 'ES'}
```

#### 4.2.7 LinkButton

**LinkButton** Tlačítko s odkazem ('LINK') - Vytvoří odkaz na jinou stránku

Vlastnost	Typ	Popis
target_url	string	URL na kterou odkaz vede
label	string	Popisek tlačítka

##### Příklad

```
1 {type: 'LINK', target_url: '/hmi/index.html', desc: 'Go to index page ...',
    label: 'To index'}
```

#### 4.2.8 ManualPulseRow

**ManualPulseRow** Manuální puls ('MP') - Manuální puls (pro blok MP). Existují dva typy této komponenty. Buď 'type:'MP' který zapíše hodnotu TRUE do cstringu po stisknutí tlačítka. Druhá možnost 'type:'MPRpt' zapisuje hodnotu TRUE periodicky každých 20 ms když je tlačítko stisknuto. To lze použít pro prodloužení výstupu tlačítka MP.

Vlastnost	Typ	Popis
color	bool	Barva pozadí tlačítka
label	int	Popisek tlačítka
labelColor	int	Barva popisku tlačítka

##### Příklad

```
1 {alias: 'MP_RUN', cstring: "task.MP_RUN:BSTATE", type: 'MP'}
2
```

```

3 {alias: 'MP_RUN', cstring:"task.MP_RUN:BSTATE", type: 'MPRpt'}
4
5 {alias: 'MP_RUN', label:'Press me', type: 'MP', color:REX.UTIL.CONSTANTS.
  COLORS.error, labelColor:'#000000'}

```

### 4.2.9 PushButton

**PushButton** Tlačítko ('PB') - Jedno tlačítko pro nastavení různých hodnot při stisku a uvolnění.

Vlastnost	Typ	Popis
label_false	string	Popisek pro hodnotu false / off / disable.
label_true	string	Popisek pro hodnotu true / on / enable.
reverse_meaning	bool	Pokud je nastaveno, tak '0' znamená zapnuto / ON a '1' vypnuto / OFF
color_false	string	Změní barvu tlačítka FALSE, když je aktivní
color_true	string	Změní barvu tlačítka TRUE, když je aktivní
value_release	int   string	Nastaví hodnotu, která se má nastavit při uvolnění ('reverse_meaning se nepoužije').
value_press	int   string	Nastaví hodnotu, která se má nastavit při stisku ('reverse_meaning se nepoužije').

#### Příklad

```

1 // Write `true` on press and `false` on release
2 {type: 'PB'}
3
4 // Write float value
5 {type: 'PB', value_release: 0, value_press: 0.1}
6
7 // All options
8 {type: 'PB', label_false: 'OFF(0)', label_true: 'ON(1)', reverse_meaning:
  false,
9   color_false:'', color_true:'', value_release: 0, value_press: 1}

```

## Kapitola 5

# Knihovna REX.HMI

### 5.1 Jak používat knihovnu

Uživatel může programovat HMI prostřednictvím rozhraní REX.HMI. Obsahuje několik veřejných metod pro čtení a zápis položek z REXYGEN cílů. Pro vytvoření a sestavení vlastního HMI založeného na knihovně RexHMI postupujte podle těchto kroků:

1. Vytvořte adresář projektu s *exec.mdl*. Přidejte blok *HMI* do exekutivy (viz [2] pro více detailů).
2. Vytvořte soubor *\*.hmi.js* v adresáři *./hmisrc* s funkcí `REX.HMI.init = function() {` uvnitř.
3. Začněte psát svůj skript.

Když je skript připraven, povolte generování RexHMI pomocí parametru *Generate-RexHMI* v bloku *HMI* exekutivy. Poté spusťte funkci *Compile and Download* v REXYGEN Studio. Obsah souboru *\*.hmi.js* je vložen do šablony RexHMI se všemi potřebnými skripty a knihovnami, také veškerý další obsah adresáře *hmisrc* je zkopírován do cílového adresáře *./hmi*. HMI je poté staženo do cíle a je dostupné z interního webového serveru.

Rozhraní REX.HMI je popsáno v následujících sekcích. Každá funkce obsahuje krátký příklad s běžnými parametry.

```
1 // Bezne pouziti
2 // Pripojte tuto funkci k HTML dokumentu nebo pouzijte preddefinovany
   zastupny symbol v sablone
3 REX.HMI.init = function(){
4 // Zmente nektera zakladni nastaveni
5
6 }
```

### 5.2 Referenční příručka pro REX.HMI

Hlavní vstupní bod pro všechnu vizualizaci RexHMI. Tato třída zpřístupňuje všechny metody potřebné pro čtení a zápis proměnných v řídicím schématu přes WebSockets.

**Vydává:** událost:čas, událost:online, událost:offline

### Vlastnosti

Název	Typ	Popis
REX.HMI.kioskMode	boolean	Nastavte na true pro povolení kiosk módu HMI
REX.HMI.disableAutoReload	boolean	Nastavte na true, pokud má být zakázáno automatické znovunačtení webové stránky. Automatické znovunačtení je voláno, když se změní exekutiva REXYGEN nebo HMI. To může být užitečné, pokud jsou obecné signály čteny do Grafu a nechcete ztratit historii

### Seznam všech dostupných funkcí

- `.init()`
- `.connect() =>Promise`
- `.disconnect()`
- `.getTarget(url) =>WSTarget`
- `.setTargetUrl(url, force) =>Promise`
- `.setRefreshRate(period)`
- `.addItem(opt) =>Item`
- `.addItems(items) =>Array.<Item>`
- `.removeItem(alias) =>Promise`
- `.get(alias) =>Item`
- `.$i(alias) =>Item`
- `.addGroup(g)`
- `.removeGroup(g)`
- `.addTrend(t)`
- `.removeTrend(t)`
- `.writeValues(aliases, values) =>Promise`
- `.setTitle(title)`

- `.setHeaderTitle(title)`
- `.showHeartBeatClock(show)`
- `.useClientTime(use)`
- `.getItemsEventSynchronizer(aliases, events) =>EventSync`
- `.commit() =>Promise`

**REX.HMI.init()** Tuto metodu může přepsat uživatel. Metoda *init* je volána, když je otevřeno websocket spojení a můžete přidat vlastní položky pro operace RW. Metoda může být volána buď synchronně nebo asynchronně s callbackem. Příklad funkce *REX.HMI.init* je součástí HTML šablony

#### Příklad

```

1 // Synchronní verze
2 REX.HMI.init = function(){
3 REX.HMI.addItem({alias:"SP", cstring:"task.block:param"});
4 }
5
6 // Init s callbackem
7 REX.HMI.init = function(done){
8 done();
9 }

```

**REX.HMI.connect() =>Promise** Připojí RexHMI k cíli. Tato funkce je volána automaticky

**REX.HMI.disconnect()** Odpojí všechny položky od cíle. Zastaví čtení a zruší spojení se všemi cíli.

**REX.HMI.getTarget(url) =>WSTarget** Vrátí základní REXYGEN cíl na základě dané URL. Pokud je URL null (nejčastější případ), pak vrátí výchozí cíl.

Parametr	Typ	Popis
url	Řetězec	URL požadovaného cíle

#### Příklad

```

1 // Nactení verze vychoziho cile
2 REX.HMI.getTarget().getVersion().then((data)=>{console.log(data)})

```

**REX.HMI.setTargetUrl(url, force) =>Promise** Nastaví novou výchozí URL cíle. Když je stránka poskytována ze serveru (ne z localhost) a je nastaven `location.hostname`,

funkce `setTargetURL` nastaví NULL pro použití výchozí `targetURL`. Takže cíl se připojí k místu, odkud je stránka poskytována.

Toto chování lze změnit nastavením parametru `force` na `true`. Pak bude 'url' použita za každé situace.

Parametr	Popis
url	URL cíle
force	nastavit na <code>true</code> , pokud má být URL nastavena i při spuštění ze serveru s hostname

### Příklad

```

1 // Nejcastejsi pouziti
2 // Pokud bezi lokalne z file:// pripoj se na IP, pri nahrani na server (
  RexCore)
3 // pak se pripojit na location.hostname
4 REX.HMI.setTargetUrl("ws://192.168.0.100:8008");
5
6 // Vzdy se pripojit na localhost
7 REX.HMI.setTargetUrl("ws://127.0.0.1:8008", true);
8
9 // Vzdy pouzivat zabezpeceny WebSocket
10 REX.HMI.setTargetUrl("wss://192.168.0.100:8008", true);

```

**REX.HMI.setRefreshRate(period)** Změní výchozí rychlost obnovy (jak rychle budou čtena data z `RexCore`)

Parametr	Typ	Výchozí	Popis
period	číslo	500	Nové období obnovy [ms]

### Příklad

```

1 // Zmenit vchozi obnovovaci periodu na 1000 ms (1s)
2 REX.HMI.setRefreshRate(1000);

```

**REX.HMI.addItem(opt) =>Item** Registrovat novou *Položku* definovanou `alias` a `cstring` pro periodické čtení a asynchronní zápis.

**Vrací:** Položka - - Registrovaná položka

Parametr	Typ	Popis
opt	Objekt	Hlavní konfigurační objekt
opt.alias	řetězec	Alias pro připojovací řetězec
opt.cstring	řetězec	Připojovací řetězec signálu z REXYGEN
[opt.url]	řetězec	URL cíle, pokud NULL pak se použije výchozí



Parametr	Typ	Popis
[opt.period]	číslo	Období obnovy položky [ms]
[opt.writeCString]	řetězec	Pokud definováno, hodnota položky bude zapsána do tohoto <i>cstring</i>

### Příklad

```

1 // Nejbeznejsi pouziti
2 var sp = REX.HMI.addItem({alias:"SP", cstring:"task.block:param"});
3 sp.on('change',(data)={console.log(data)});
4
5 // Rozdilne mista pro cteni a zapis (napr. Zapsat pred saturaci a cist po
6 REX.HMI.addItem({alias:"SP", cstring:"task.SAT:y", writeCString:"task.CNR
   :ycn"});

```

**REX.HMI.addItem(items) => Array.<Item>** Přidat několik položek najednou.

Viz [HMI#addItem](#) pro více detailů

**Vrací:** Array.<Item> - - Vrací pole přidanych položek

Parametr	Typ	Popis
items	Array. <Objekt>	Pole položek

**REX.HMI.removeItem(alias) => Promise** Odstranit položku na základě jejího `alias`

Parametr	Popis
alias	Položky <code>alias</code> použitý během registrace

**REX.HMI.get(alias) => Item** Najít položku pomocí jejího `alias`

Parametr	Popis
alias	Položky <code>alias</code> použitý během registrace

**REX.HMI.\$i(alias) => Item** Najít položku pomocí jejího `alias`

Parametr	Popis
alias	Položky <code>alias</code> použitý během registrace

**REX.HMI.addGroup(g)** Zaregistruje vlastní skupinu položek pro operace čtení/zápisu

Parametr	Typ	Popis
g	REX.WS.Group	Skupina pro registraci

#### Příklad

```
1 // Vytvoreni skupiny, viz REX.WS.Group
2 var g = new REX.WS.Group({id:"group1", period:100, url:""});
3
4 // Pridani nekterych polozek
5 g.addItem(new REX.WS.Item({id:"ITEM-1", cstring:"task.block:param", url:"
  "}));
6
7 // Registrace udalosti
8 g.on('read',(data)=>{console.log(data)});
9
10 // Registrace skupiny
11 REX.HMI.addGroup(g);
```

**REX.HMI.removeGroup(g)** Zrušení registrace vlastní skupiny

Parametr	Typ	Popis
g	REX.WS.Group	Instance skupiny, která bude zrušena

**REX.HMI.addTrend(t)** Na rozdíl od REX.WS.Group Trend čte data z bloků TRND\*. Tyto bloky ukládají několik signálů s vyrovnávacími daty. Po registraci může uživatel zpracovávat data z bloků TRND\*.

Parametr	Typ	Popis
t	REX.WS.Trend	Trend, který bude zaregistrován

#### Příklad

```
1 // Vytvoreni noveho trendu
2 var trend = REX.WS.Trend({cstring:"task.TRND", id:"TRND-1", period:500,
  readWholeBuffer:true});
3
4 // Registrace
5 REX.HMI.addTrend(trend);
6 // Pridani obsluznych rutin udalosti
7 trend.on('read',(data)=>{console.log(data)});
```

**REX.HMI.removeTrend(t)** Zrušení registrace trendu

Parametr	Typ
t	REX.WS.Trend

**REX.HMI.writeValues(aliasy, hodnoty) => Promise** Zapsání jedné nebo několika hodnot do řídicího systému. Použitím již registrovaných položek (aliasů).

Parametr	Typ	Popis
aliases	Pole. <Řetězec>	Pole již registrovaných aliasů
hodnoty	Pole	Pole hodnot k zápisu

### Příklad

```

1 // Registrace nekterych polozek
2 REX.HMI.addItems([{alias:"A1",cstring:"task.A1:ycn"},{alias:"A2",cstring:
   "task.A2:ycn"}]);
3
4 // Zapis hodnot
5 REX.HMI.zapsatHodnoty(["A1","A2"],[0.5, 0.7]);

```

**REX.HMI.setTitle(title)** Změna názvu v záhlaví a názvu webové stránky

Parametr	Typ	Popis
title	Řetězec	Nový název pro záhlaví a webovou stránku

### Příklad

```

1 REX.HMI.setTitle("Moje HMI");

```

**REX.HMI.setHeaderTitle(title)** Změna pouze názvu záhlaví

Parametr	Typ	Popis
title	Řetězec	Nový název pro záhlaví

**REX.HMI.showHeartBeatClock(show)** Pokud je hodnota pravda, šablona zobrazí HODINY v pravém horním rohu hlavní obrazovky. Když aktualizace času zastaví, cílové zařízení není připojeno a HMI se neaktualizuje

Parametr	Typ	Popis
show	Boolean	Pravda pro zobrazení hodin

**REX.HMI.useClientTime(use)** Nastavte na True, pokud má být čas zobrazen v klientském čase, nikoli v čase cílového zařízení. Když cíl není schopen synchronizovat se s nějakým časovým serverem, je možné používat a zobrazovat časy v klientském čase.

Param	Type	Description
use	Boolean	True pro použití klientskeho casu misto ciloveho

**REX.HMI.commit() => Promise** REX.HMI.commit

Když je voláno, všechny položky přidané po připojení HMI k cíli jsou inicializovány najednou.

**Druh:** instanční metoda **HMI Vrací:** Promise - - Promise, že všechny položky jsou inicializovány.

**REX.HMI.getItemsEventSynchronizer(aliases, events) =>EventSync** Vrátil objekt EventSync, který emituje události, když všechny registrované položky emitují stejnou událost.

Parametr	Typ	Popis
aliases	Array String	Pole aliasů položek nebo jeden alias jako řetězec
events	Array String	Pole sledovaných událostí

### Příklad

```

1 // Registrace nekterych polozek
2 REX.HMI.addItems([{alias:"A1",cstring:"task.A1:ycn"},{alias:"A2",cstring:
   "task.A2:ycn"}]);
3
4 var sync = REX.HMI.getItemsEventSynchronizer(["A1","A2"],["read"]);
5
6 sync.on("read",()=>{console.log("Vsechny udalosti cteni byly emitovany")
   });

```

## 5.3 Referenční příručka pro REX.HMI.Graph

Časový grafický komponent, který je zobrazen na spodní části webové stránky. Graf může číst libovolný signál připojený prostřednictvím `alias` a `cstring` nebo všechny signály z bloků TRND\*. Graf se zobrazí, když je přes funkci `addSignal` nebo `addTrend` přidán první signál.

REX.HMI.Graph má následující funkce:

- `.resume()`
- `.pause()`
- `.show()`
- `.hide()`
- `.addSignal(opt)`
- `.addTrend(opt)`
- `.setSize(size)`
- `.setRefreshRate(period)`
- `.hideLegend()`

**REX.HMI.Graph.resume()** Obnoví překreslování grafu.

**Příklad**

```
1 REX.HMI.Graph.resume()
```

**REX.HMI.Graph.pause()** Pozastaví překreslování grafu.

**Příklad**

```
1 REX.HMI.Graph.pause()
```

**REX.HMI.Graph.show()** Zobrazí graf.

**Příklad**

```
1 REX.HMI.Graph.show()
```

**REX.HMI.Graph.hide()** Skryje graf.

**Příklad**

```
1 REX.HMI.Graph.hide()
```

**REX.HMI.Graph.addSignal(opt)** Přidá libovolný signál z exekutivy REXY-GEN do trendu. Upozornění! Data jsou ukládána uvnitř webové stránky, po obnovení stránky budou všechna data ztracena.

Parametr	Typ	Popis
<code>opt</code>	Objekt	Hlavní konfigurační objekt

Parametr	Typ	Popis
opt.alias	string	Alias pro připojení
[opt.cstring]	string	Připojovací řetězec signálu z REXYGEN
[opt.desc]	Pole. <string>	Popis signálu
[opt.period]	číslo	Období obnovení signálu [ms]

### Příklad

```
1 REX.HMI.Graph.addSignal({alias:"Signal-1", cstring:"task.CNR:ycn", desc:"
  Nastaveni bodu", period:500});
```

**REX.HMI.Graph.addTrend(opt)** Přidá signály z bloků TRND\* do společného grafu v HMI.

Parametr	Typ	Popis
opt	Objekt	Hlavní konfigurační objekt
opt.cstring	string	Připojovací řetězec pro blok TRND*
[opt.labels]	Pole. <string>	Pole štítků signálů
[opt.period]	číslo	Perioda překreslení grafu

### Příklad

```
1 REX.HMI.Graph.addTrend({cstring:"task.TRND", labels:["signal-1","signal-2
  "], period:500});
```

**REX.HMI.Graph.setSize(size)** Změna velikosti grafu. Velikost je v procentech stránky.

Parametr	Typ	Výchozí	Popis
size	číslo	0.39	Velikost grafu v procentech <0;1>

### Příklad

```
1 REX.HMI.Graph.setSize(0.39); // Default
```

**REX.HMI.Graph.setRefreshRate(period)** Změna frekvence obnovování všech signálů a trendů.

Parametr	Popis
period	Perioda obnovování [ms]

### **Příklad**

```
1 REX.HMI.Graph.setRefreshRate(250);
```

**REX.HMI.Graph.hideLegend()** Skryje legendu grafu.

### **Příklad**

```
1 REX.HMI.Graph.hideLegend();
```

# Literatura

- [1] REX Controls s.r.o.. *Getting started with REXYGEN*, 2020. [→](#).
- [2] REX Controls s.r.o.. *Function blocks of REXYGEN – reference manual*, 2020. [→](#).