

# Grafické operátorské panely Weintek řady iP



## **Grafické operátorské panely Weintek řady iP**

Copyright © TECON spol. s r. o., Vrchlábí, Česká republika

Tato publikace prošla jen částečnou jazykovou korekturou.

Tato publikace vznikla na základě informací dostupných v době tisku.

Veškeré změny jsou vyhrazeny.

Všechny ochranné známky uvedené v této knize jsou majetkem jejich vlastníků

**Žádná část této publikace nesmí být publikována a šířena žádným způsobem a v žádné podobě bez výslovného svolení firmy TECON s.r.o.**

### **TECON spol. s r. o.**

Komenského 63

Vrchlábí 543 01

Česká republika

Tel.: 499 429 100, 499 429 117

Fax: 499 422 508

e-mail: [info@TECON.cz](mailto:info@TECON.cz) <http://www.TECON.cz>

## Obsah

Technické parametry .....	4
Instalace programu EasyBuilder Pro.....	5
Vytvoření nového projektu .....	6
Bitový přepínač s lampičkou .....	9
Zadávání a zobrazení numerických dat .....	14
Inkrementace a dekrementace .....	17
Alarmy (Eventy) .....	20
Logování dat .....	26
Data transfer .....	35
Více oken na panelu .....	37
Receptury .....	41
Makra.....	48
Security.....	52
Ostatní aktivní prvky .....	56

## Technické parametry

Nové grafické **operátorské panely firmy WEINTEK řady iP** s dotykovou obrazovkou jsou ekonomicky velice výhodným řešením pro vizualizaci a monitoring technologických procesů řízených programovatelnými automaty PLC.

Dotykové panely WEINTEK řady iP jsou aktuálně nabízeny v provedení s úhlopříčkou 4,3 palce, 7 palců a 10 palců TFT v širokoúhlém provedení.

Weintek panely řady iP, jsou hardwarově podobné jako předchozí panely řady iP. Rozdíl zde je v mnoha ohledech. Nové panely řady iP umožňují oproti předchozím panelům rychlejší Boot Time a to až o 10 vteřin. Dále je zde použit nový typ displeje, který disponuje až 16.2M barev oproti dosavadním 65536. Je zde také použita lepší a rychlejší komprese obrázků a textů. Panely řady iP disponují procesorem Cortex A8 32Bit 600 MHz, paměť RAM 128MB, paměť FLASH 256MB a rozsahem provozních teplot 0°C až 50°C.

Sofistikovaný konfigurační software **EasyBuilder PRO je k dispozici zcela ZDARMA.**

	MT8052iP	MT8072iP	MT8106iP
<b>DISPLEJ</b>	4,3" TFT	7" TFT	10.1" TFT
<b>JAS</b>	400	300	350
<b>BARVY</b>	16.7M	16.7M	16.7M
<b>ROZLIŠENÍ</b>	480x272	800x480	1024x600
<b>DOTYKOVÝ PANEL</b>	Rezistivní 4 vodičový		
<b>SÉRIOVÉ PORTY</b>	COM1: RS-232/RS-485 2W/4W, COM3: RS-485 2W	COM1: RS-232 COM3: RS-485 2W/4W	COM1: RS-232 4W COM2: RS-485 2W/4W, COM3: RS-232 2W/RS-485 2W/4W
<b>ETHERNET</b>	10/100 Base-T x 1		
<b>PROCESOR</b>	Dual-core RISC Cortex A8 32Bit		
<b>PAMĚŤ</b>	256 MB		
<b>RAM</b>	128 MB		
<b>NAPÁJENÍ</b>	24±20% VDC		
<b>PROVOZNÍ TEPLOTA</b>	0° ~ 50°C		
<b>SOFTWARE</b>	Easy Builder PRO verze 6 Weincloud Easy Access 2.0 (volitelné)		

**Srovnání všech řad** panelů Weintek z pohledu funkcí a možností naleznete [zde](#).

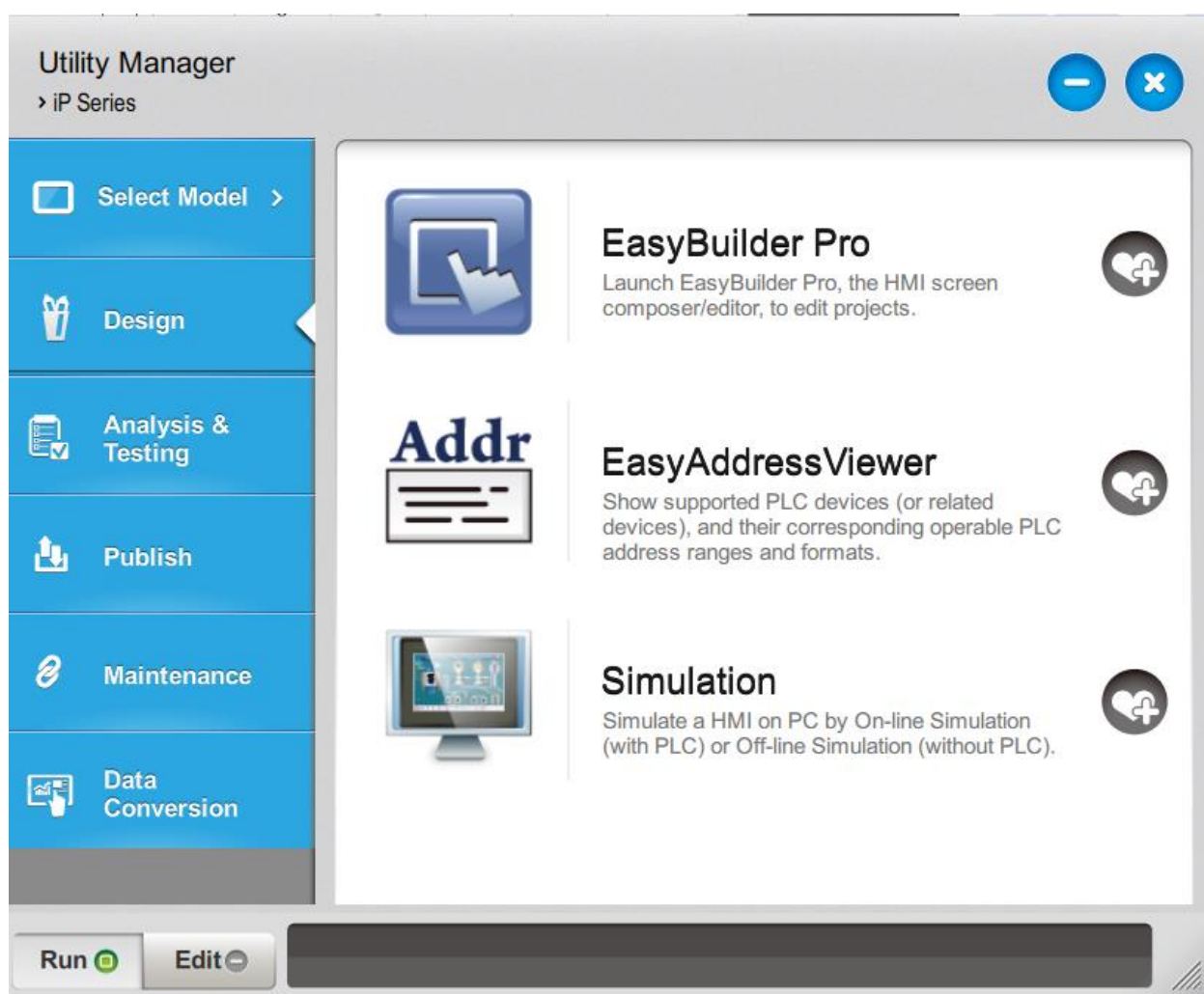
## Instalace programu EasyBuilder Pro

**Spustěte Windows**, jestliže jste tak ještě neučinili, spustěte operační systém **Microsoft Windows**. Jestliže jste doposud nepracovali se systémem Windows, poraďte se s manuálem Windows jak nakonfigurovat systém na optimální výkon a jak spouštět aplikace. Konfigurační software **EasyBuilder Pro** je určen pro 64 bitové operační systémy **Windows od verze 10 výše**.

**Rozbalte archiv** Instalační archiv **EBproV60902.ZIP** (nebo novější) rozbalte na disk a spustěte soubor **SETUP.EXE**

Nejprve se naučíme jak spustit vývojové prostředí EB Pro a zkusíme **napsat** jednoduchou aplikaci, kterou **otestujeme** pomocí **offline simulace** na obrazovce PC. Po té projekt uložíme do panelu a vyzkoušíme jeho funkčnost.

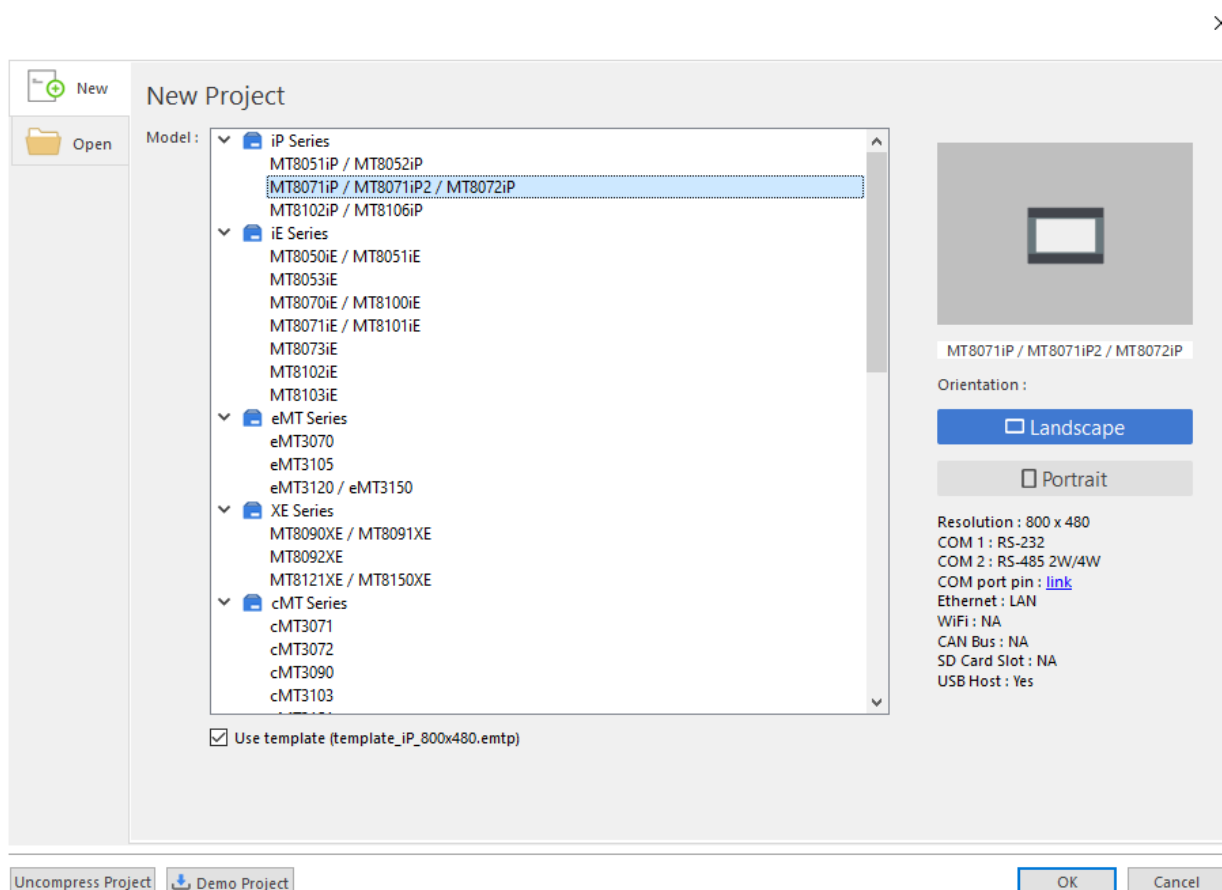
Spustěte **Utility Manager**, který naleznete na ploše. Zobrazí se následující okno. V tomto okně naleznete vývojové prostředí včetně další užitečných funkcí.



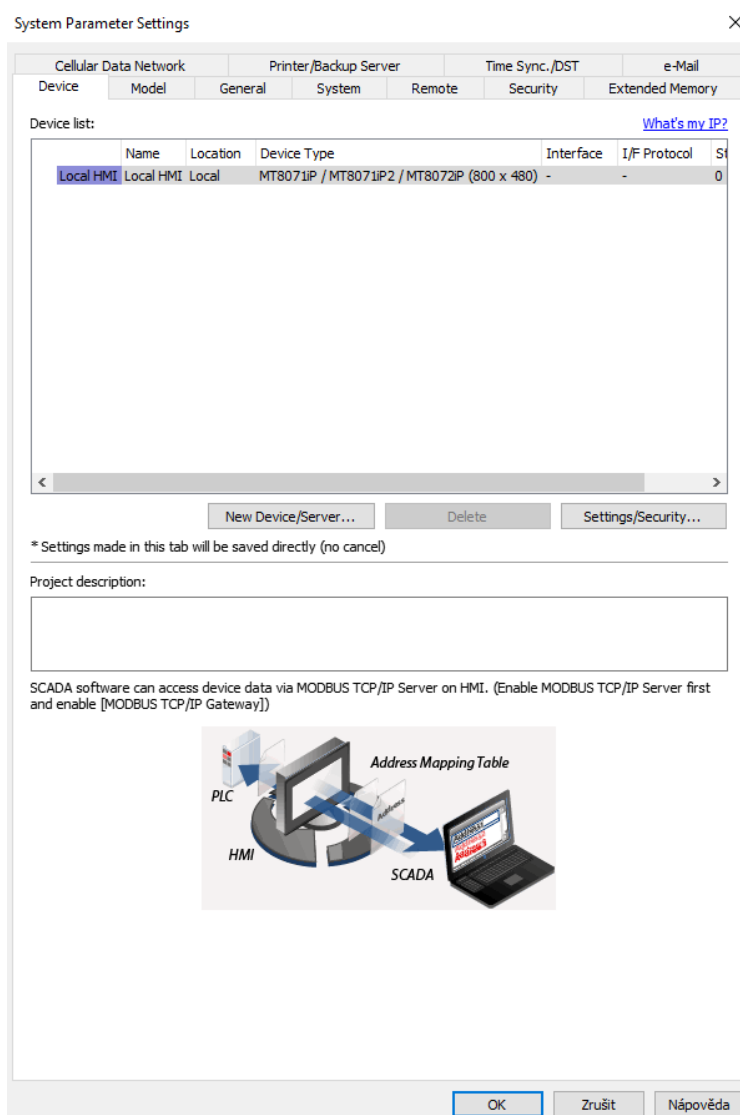
## Vytvoření nového projektu

Kliknutím na **EasyBuilder Pro** spustíme vývojové prostředí. Zobrazí se okno s možností založení nového projektu, otevření projektu z adresáře nebo výběr z posledních použitých projektů.

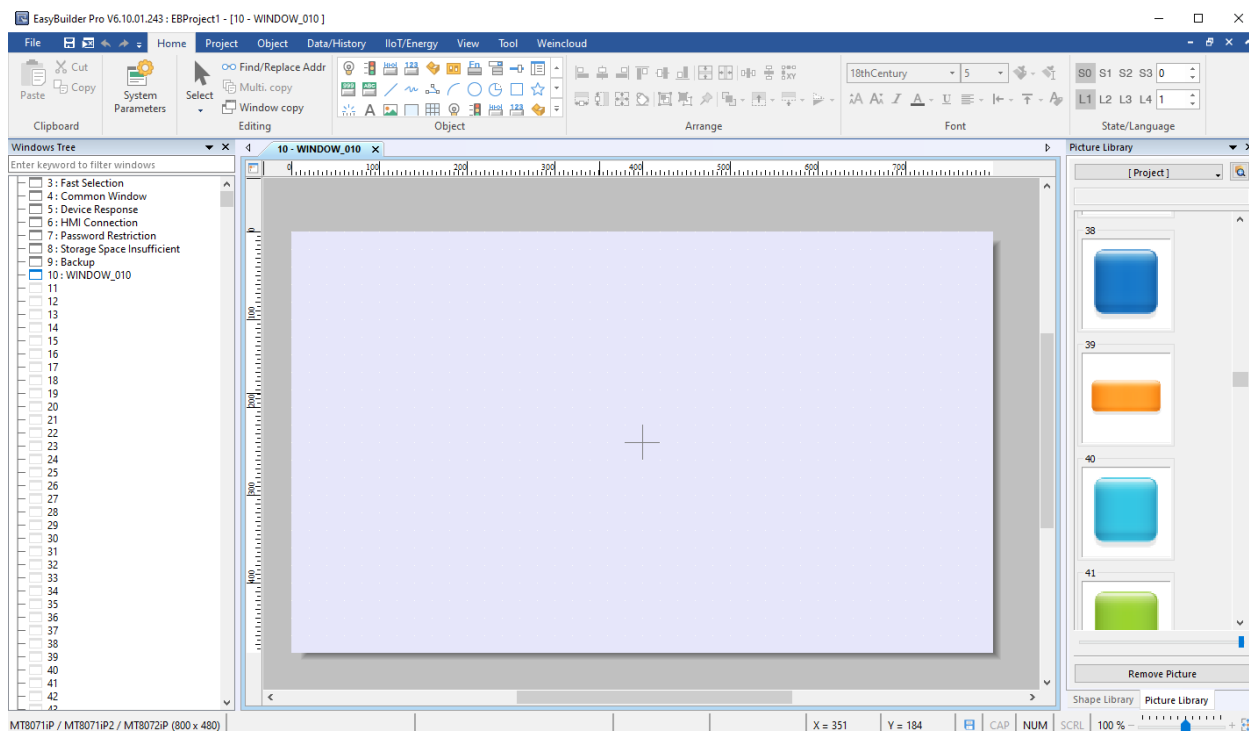
Zvolíme nový projekt. Následně se zobrazí okno s výběrem typu panelu, pro který chceme projekt vytvářet. Pro účely příručky bude stačit panel MT8072iP (800x480)



Po výběru typu panelu se otevře jako první nastavení systémových parametrů.



V tomto okně (**System Parametr Settings**) lze nastavit komunikaci s připojeným zařízením a základní vlastnosti projektu. Tato různá nastavení si ukážeme dále. Klikneme na OK. Dostáváme se tak na prázdnou plochu, která symbolizuje obrazovku panelu.



**V názvu okna** vlevo nahoře vidíte vedle jména konfiguračního software také jméno projektu, se kterým právě pracujete.

Pod ním se nachází **Menu**, ze kterého je přes jednotlivé položky možné dosáhnout na všechny funkce software.

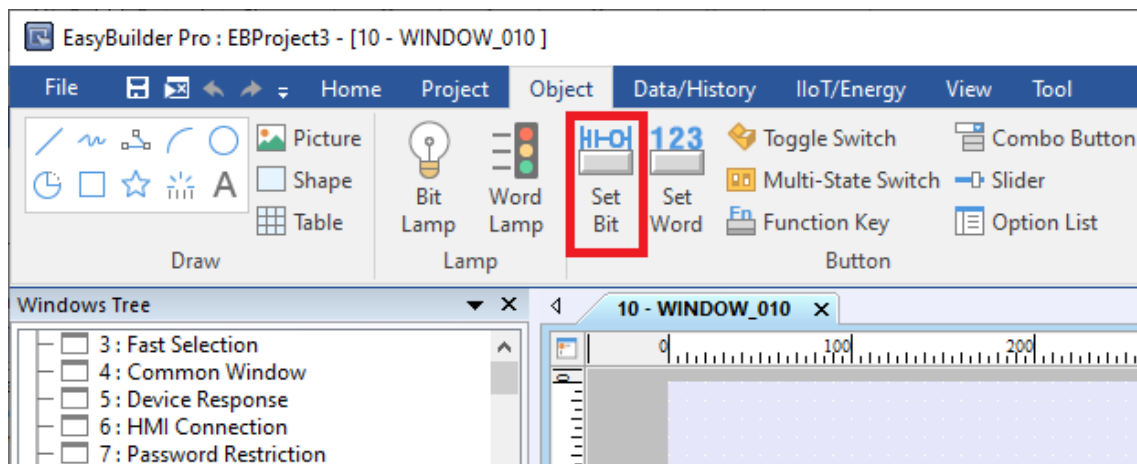
V levém podokně se nachází seznam vytvořených oken. V našem případě WINDOW\_010. Toto okno je tedy zatím prázdná plocha, dále si ukážeme, jak vytvořit první jednoduchý projekt.



## Bitový přepínač s lampičkou

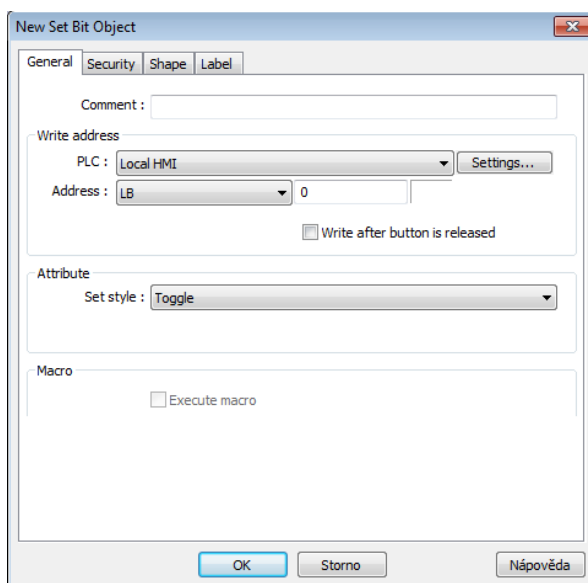
Pro vytvoření první aplikace vložíme na černou plochu nějaký prvek. Jako první projekt vytvoříme bitový přepínač, který nám bude rozsvěcet a zhasínat bitovou lampičku.

Vložíme bitový přepínač. Pomocí hlavního menu v záložce Objects -> Set Bit

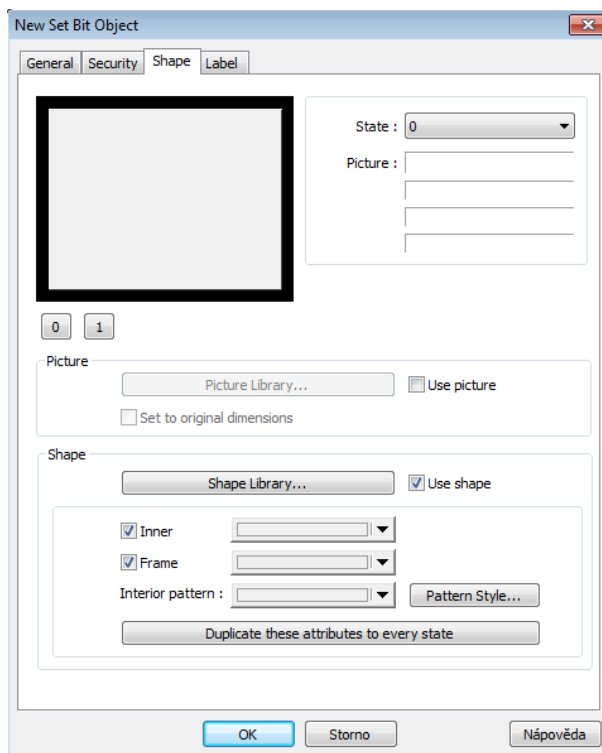


V následujícím okně poté přiřadíme požadované vlastnosti tomuto prvku. V první záložce **General** nastavíme, na jakou adresu se bude bitová hodnota zapisovat (**Write address**).

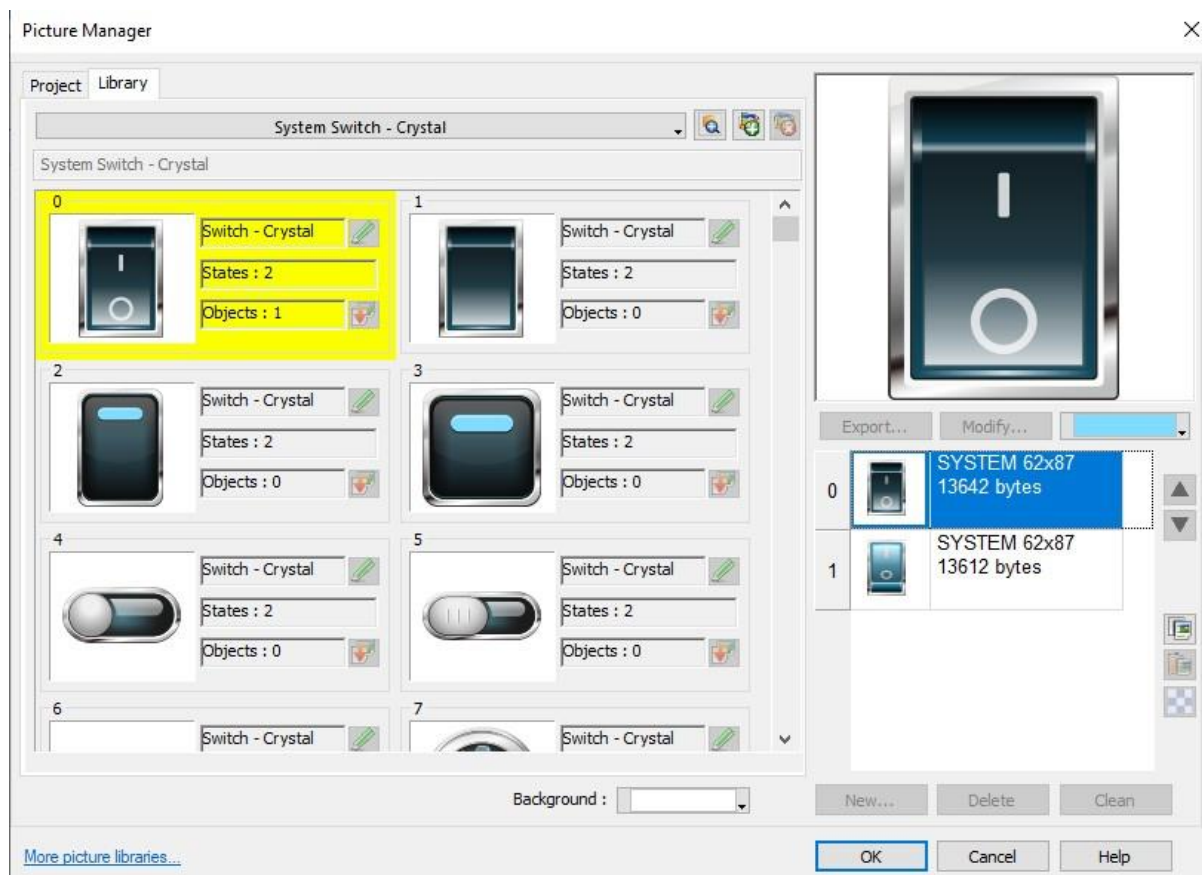
Zde jsou možnosti **LB**, **RW\_Bit**, **RW\_A\_Bit**, **LW\_bit**, **RBI**. Použijeme **LB 0**. Dále nastavíme, jakým způsobem se bude tlačítko chovat (**Attribute**). Lze vybrat mnoho způsobů chování (**Set ON**, **Set OFF**, **Toggle**, **momentary** atd...) Jelikož mi chceme vytvořit přepínač, zvolíme **Toggle**.



V záložce **Shape** můžeme nastavit vzhled daného tlačítka. Pro tuto možnost jsou zde před vytvořenými knihovny s tvary. Klikneme tedy na **Shape Library**.



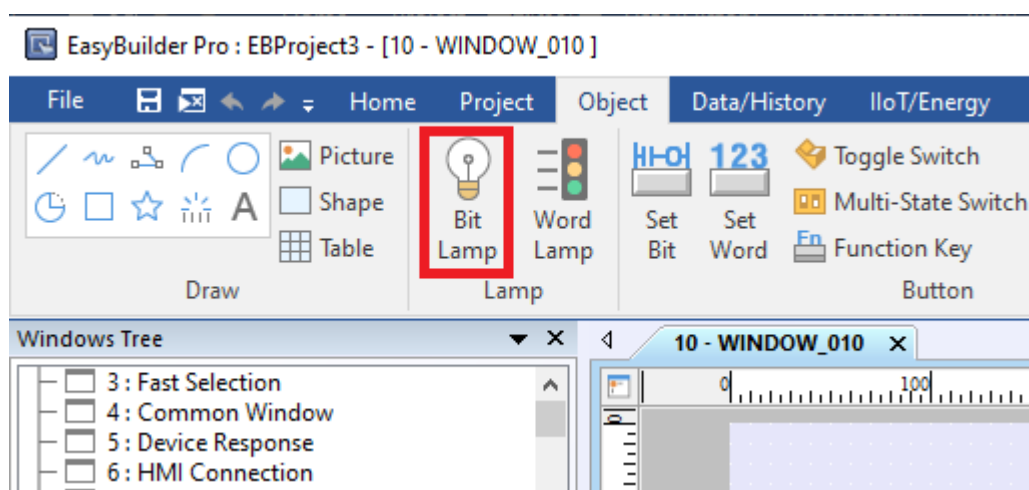
Zde si můžeme zvolit z různých tvarů. Jsou zde vytvořeny nejen tlačítka ale i lampy (signálky) a další užitečné prvky. Zvolíme tedy takovýto tvar pro náš přepínač. (v knihovně **button2**). Tyto knihovny lze rozšiřovat o vlastní knihovny.



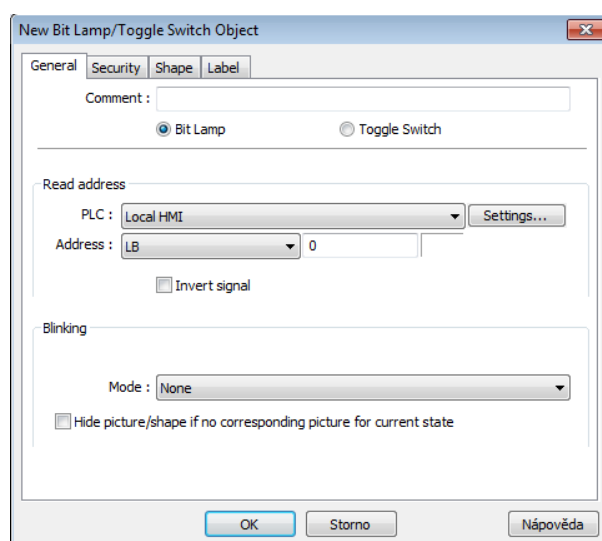
Na ploše obrazovky se nám objevil první prvek - bitový přepínač. Tento přepínač nám bude měnit hodnotu na adrese LB-0 vždy z 1 do 0 a z 0 do 1. Tuto funkci můžeme nechat zobrazit pomocí bitové lampičky, kterou si také vložíme do projektu.



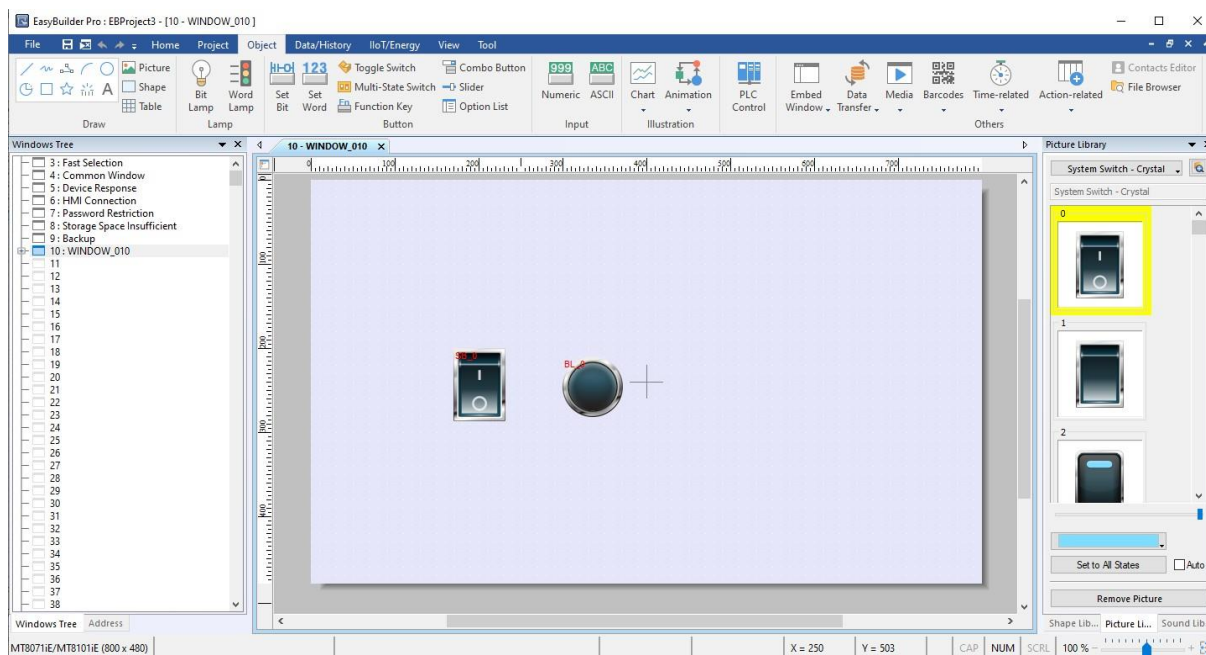
Bitovou lampičku můžeme vybrat v menu Object-> Bit Lamp.



Abychom docílili spárování s předchozím bitovým přepínačem, musíme nastavit Read address na stejnou adresu jako Write address u bitového přepínače, tedy na LB-0. Dále můžeme opět v záložce Shape pomocí Shape Library přiřadit lampičce určitý tvar.

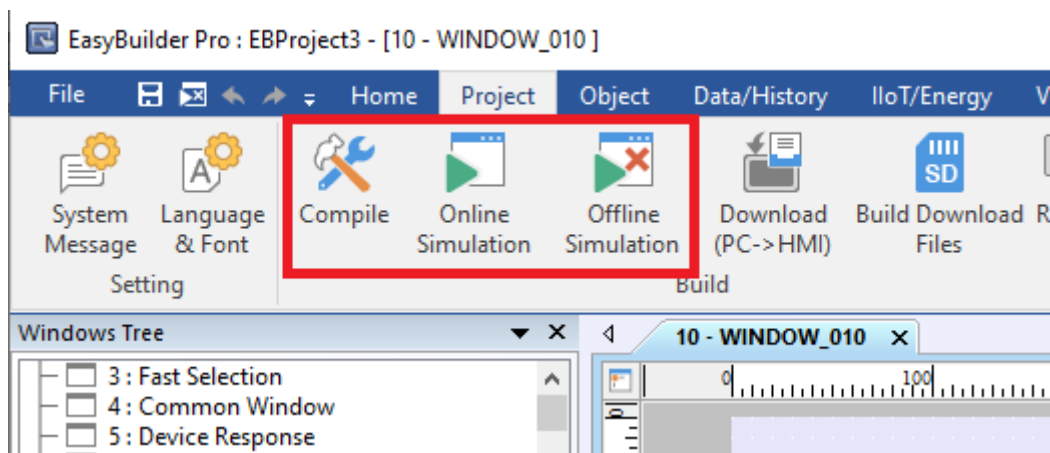


Výsledný projekt potom může vypadat následovně:

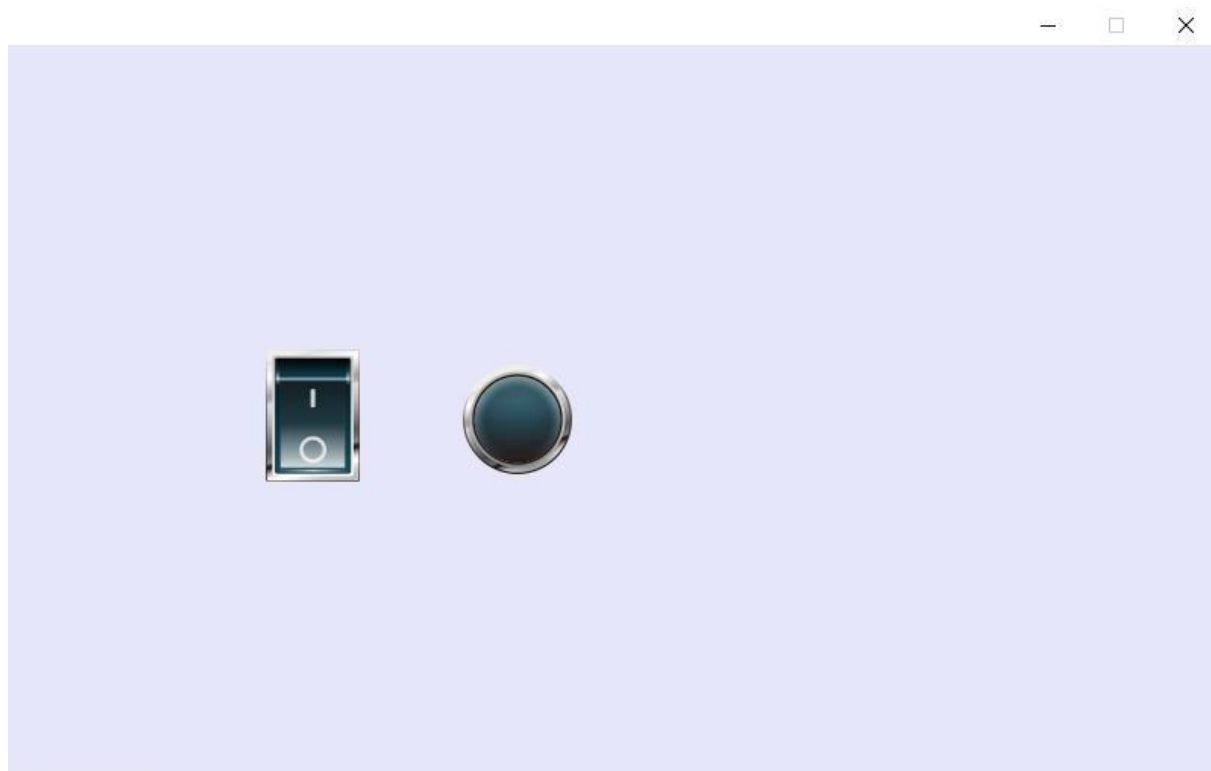


Nyní je čas na uložení a kompilaci projektu. Pro uložení projektu můžete použít, stejně jako u Windows, ikonku diskety.

Kompilovat projekt můžete v menu sekci Project, zde naleznete i spuštění Online či Offline simulace.



Zobrazí se okno simulující obrazovku panelu. Po kliknutí na přepínač se lampička rozsvítí.

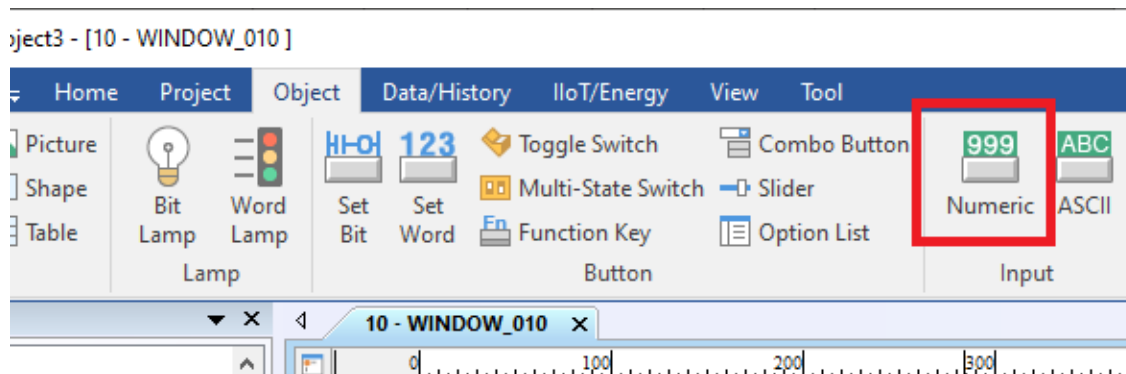


Aplikace ke stažení [zde](#)

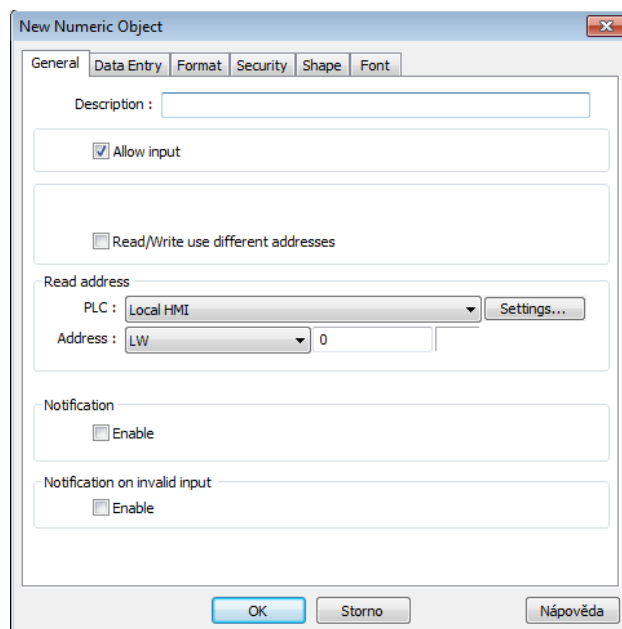
## Zadávání a zobrazení numerických dat

V minulém projektu jsme si ukázali, jak pracovat s bitovou hodnotou. Jak ji měnit a jak ji nechat zobrazovat pomocí bitové lampičky. Nyní si ukážeme, jak pracovat s numerickými hodnotami. Jak je zapisovat a jak je číst a zobrazovat.

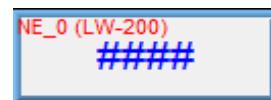
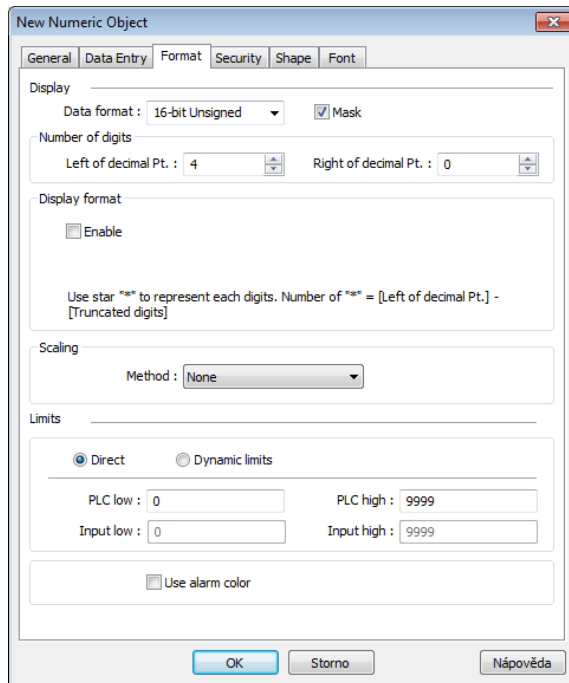
Založíme si nový projekt (File -> New). Pro zobrazení numerických dat použijeme připravený prvek, který naleznete pomocí menu Objects -> Numeric.



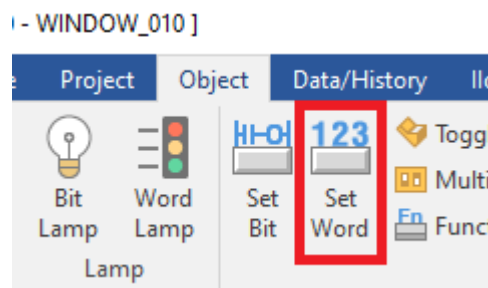
Objeví se opět dialogové okno pro zadání parametrů. V záložce General zvolíme zobrazované místo v paměti. Lze zvolit LW, RW, RW\_A, RWI. Pro možnost i zápisu do dané paměti je nutné zaškrtnout Allow input. U tohoto objektu je také možnost čtení a zápisu do různých částí paměti.



Dále pak v záložce Format můžete nastavit požadovaný formát vstupních dat. Na výběr jsou kombinace 16 a 32 bitové BCD, HEX Unsigned, Signed nebo Float. Dále lze nastavit počet čísel před a za desetinou čárkou. Dále lze nastavit limity pro danou hodnotu včetně alarmového ukazatele. Vše potvrdíme a na obrazovce nám vznikne nový objekt:

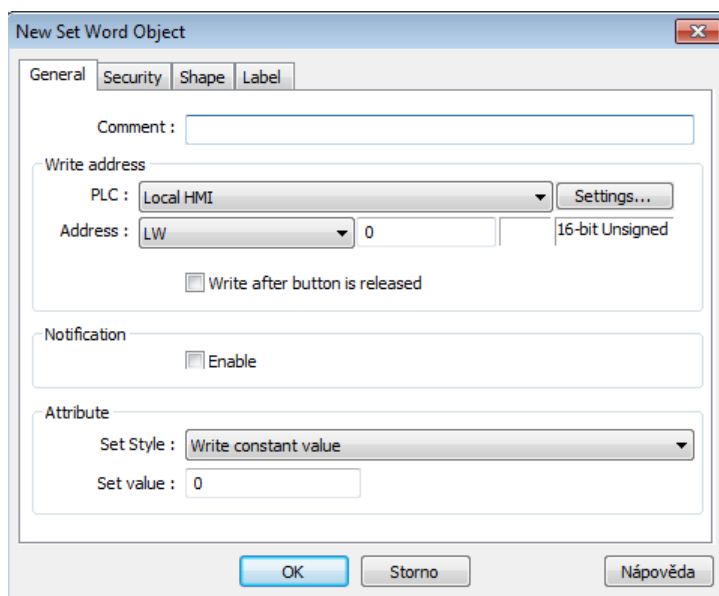


Pro nastavení hodnoty numerické proměnné zvolíme prvek Set Word. Použijeme objekt Set Word, který naleznete přes menu Objects -> Set word.



V nově otevřeném dialogovém okně postupujeme analogicky jako u ostatních prvků. V záložce General nastavíme adresu, kam chceme zapsat danou hodnotu. Dále nastavíme, jak chceme hodnotu zapsat. Zda chceme zapsat konstantu, inkrementovat či dekrementovat stávající hodnotu atd.

Danému tlačítku můžeme opět přiřadit libovolný tvar ze Shape library. Pokud bychom chtěli mít tlačítko s popisem, umožní to záložka Label, kde můžeme danému tlačítku přiřadit text s libovolnou barvou, fontem či zarovnáním.



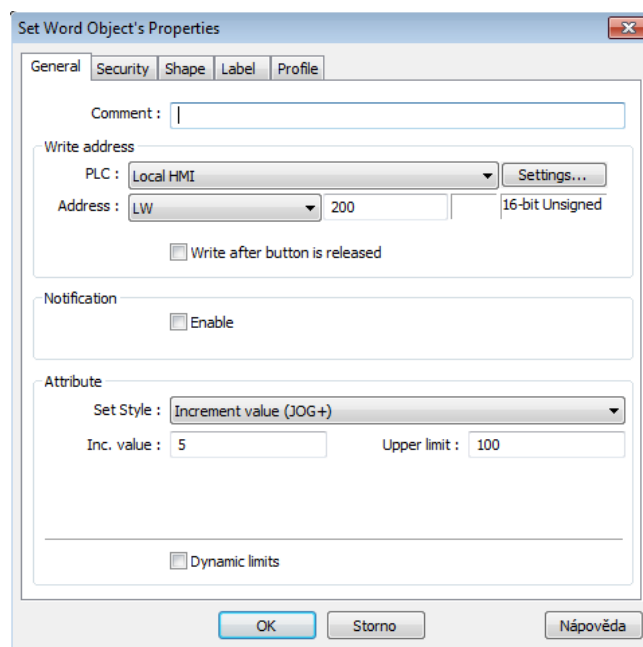
Aplikace ke stažení [zde](#)



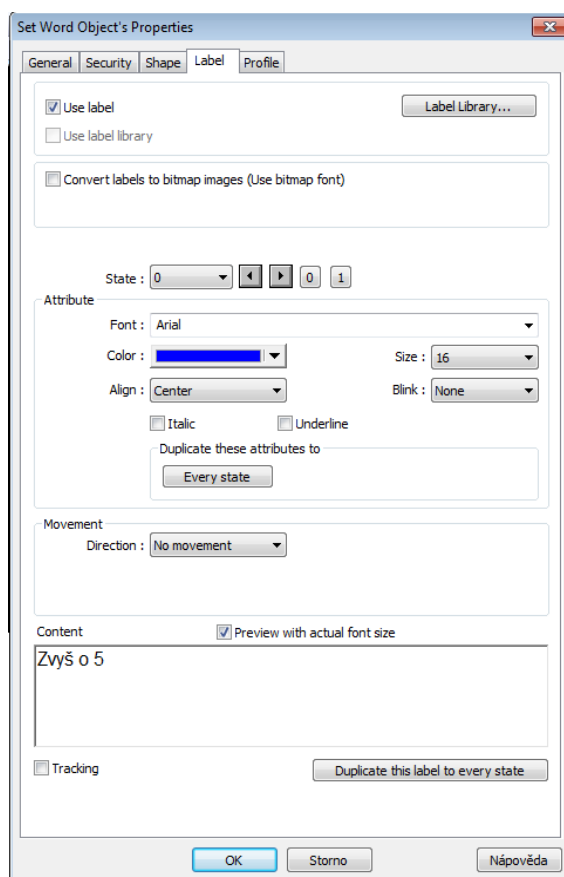
## Inkrementace a dekrementace

Dále si ukážeme, jak můžeme **danou hodnotu ovlivnit tlačítkem**. Vytvoříme si projekt, kde budeme pomocí dvou tlačítek upravovat hodnotu v LW-200 tak, že první tlačítko bude tuto hodnotu **zvyšovat o 5** a druhé ji bude **snížovat o 5**.

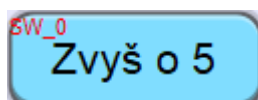
Použijeme opět objekt Set Word, naleznete ho buď jako ikonu nebo pomocí horního menu Objects -> Button -> **Set word**. V nově otevřeném dialogovém okně postupujeme analogicky jako u ostatních prvků. V záložce General nastavíme adresu, kam chceme zapsat danou hodnotu. Dále nastavíme, jak chceme hodnotu zapsat. V našem případě tedy zvolíme **Increment value (JOG +)**. Poté se nám pod tímto výběrem objeví dvě textová pole. První Inc. Value: určuje jaká hodnota se bude přičítat k hodnotě v LW-200. Druhá, Upper limit: nastaví maximální hodnotu do, které bude tlačítko inkrementovat. Dosáhne-li hodnota v LW-200 této hodnoty, tlačítko nebude dále inkrementovat.



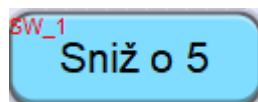
Poté co jsme nastavili funkčnost tlačítka, můžeme nastavit jeho tvar nebo popis. Tvar necháme zachovalí a nastavíme si popis daného tlačítka v záložce Label. Nejprve nahoře zaškrtneme Use Label, poté se nám zobrazí veškeré možnosti textu. Lze tu nastavit font, barva a velikost písma, dále jeho zarovnání nebo efekt v podobě běžícího textu. Ve spodní části se nachází textové pole pro samotný popis tlačítka, v našem případě zadáme Zvyš o 5.



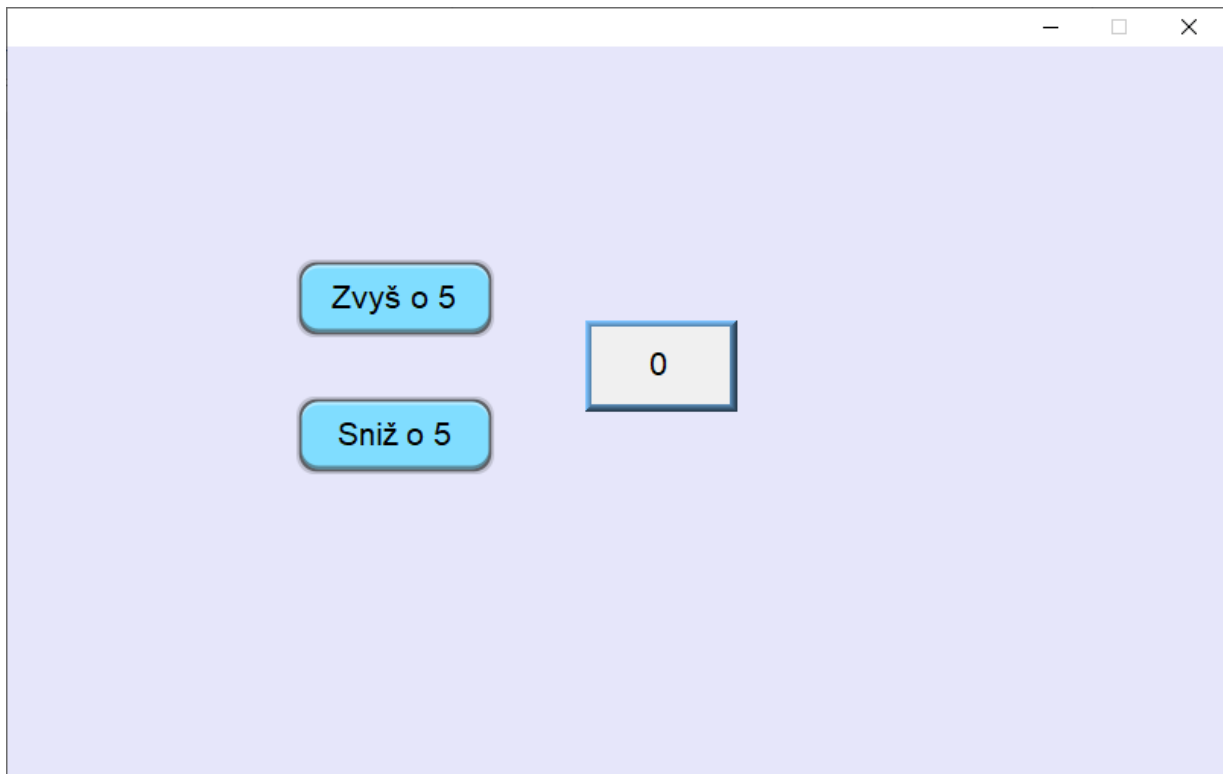
Vše potvrdíme a tlačítko pro inkrementaci máme vytvořeno. Bude vypadat následovně:



Nyní vytvoříme totožné tlačítko pouze s tím rozdílem, že nepoužijeme vlastnost Increment value (JOG +), ale **Decrement value (JOG -)**. U této vlastnosti nastavíme hodnotu, o kterou se bude hodnota v LW-200 snižovat opět na 5 a oproti vlastnosti Increment nevolíme maximální hodnotu, ale minimální. Nastavíme ji tedy na 0. V záložce Label nastavíme popisek Sniž o 5. Výsledné tlačítko bude vypadat následovně:



Teď si vytvoříme Numeric objekt pro zobrazení dané hodnoty. Postupujeme stejně jako v předešlé kapitole. Adresu, ze které budeme číst, nastavíme na LW-200. Objekt vytvoříme a umístíme na obrazovku. Výsledný projekt zkompilujeme a spustíme Off-line simulaci. Výsledný projekt může vypadat takto.



Aplikace ke stažení [zde](#)

## Alarms (Eventy)

V dalším příkladu si ukážeme, jak vytvořit **výpis hlášení** informující obsluhu o důležitých **událostech nebo chybách technologie** – Alarmů.

V konfiguračním software EB Pro se tato funkce jmenuje **Event( Alarm) Log**. Zde si nadefinujeme, jakou událost chceme sledovat. Tuto nastalou událost poté můžeme zobrazovat pomocí tří různých typů objektů a to: **Alarm Bar**, **Alarm Display** a **Event Display**.

Vytvoříme si tedy projekt ve, kterém budeme chybové události „simulovat“ pomocí dvou binárních a jedné numerické proměnné.

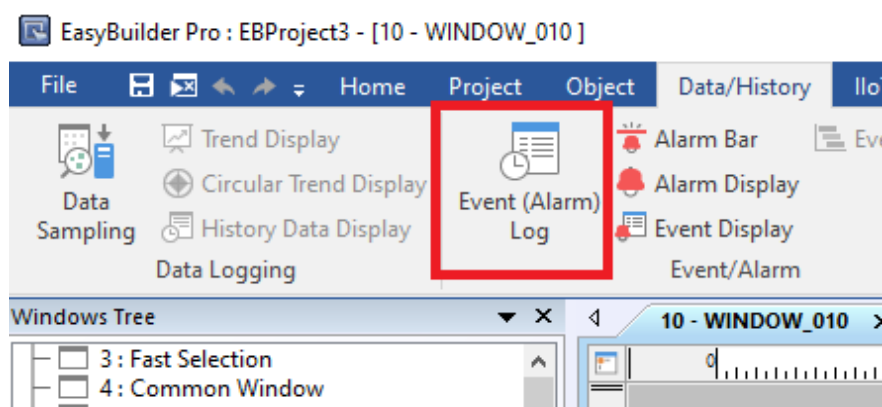
Definujeme si tyto chyby:

**Porucha PR1** – sepnut bit LB-10

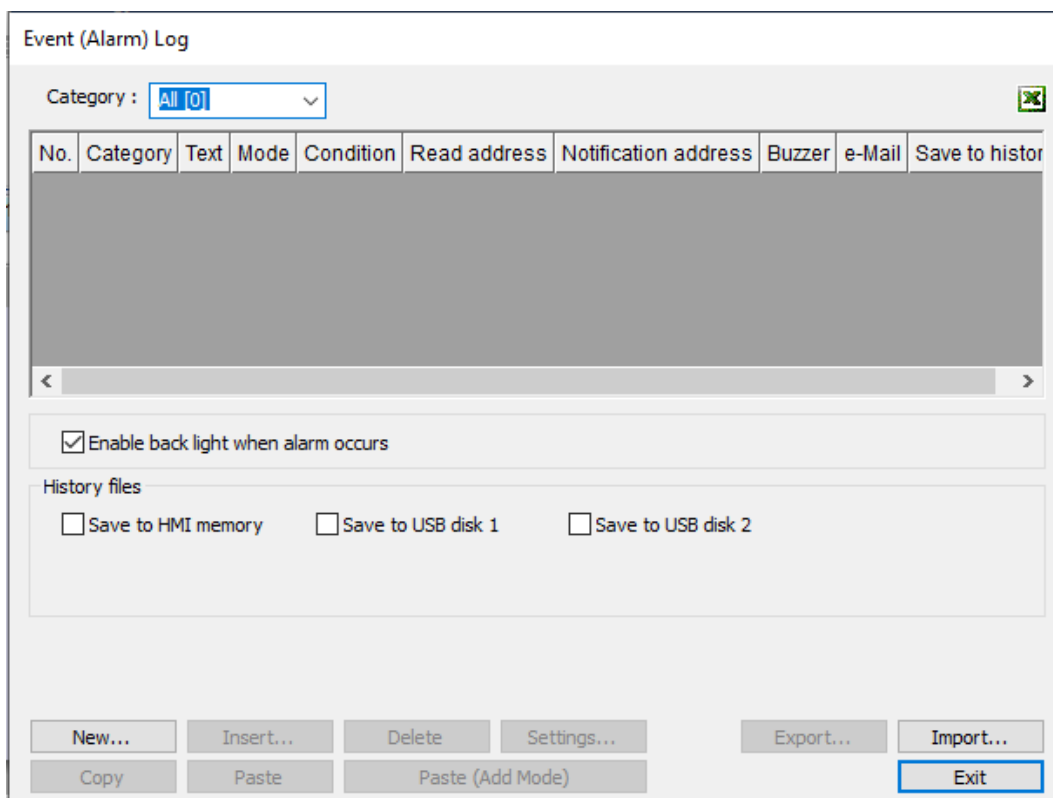
**Porucha PR2** – sepnut bit LB-11

**Max. mez L1** – kritická mez nádrže, numerická hodnota v LW-200, kterou budeme ovládat pomocí již vytvořených inkrementačních tlačítek, přesáhne číslo 20, bude ve výpisu vyhlášen alarm s touto chybou.

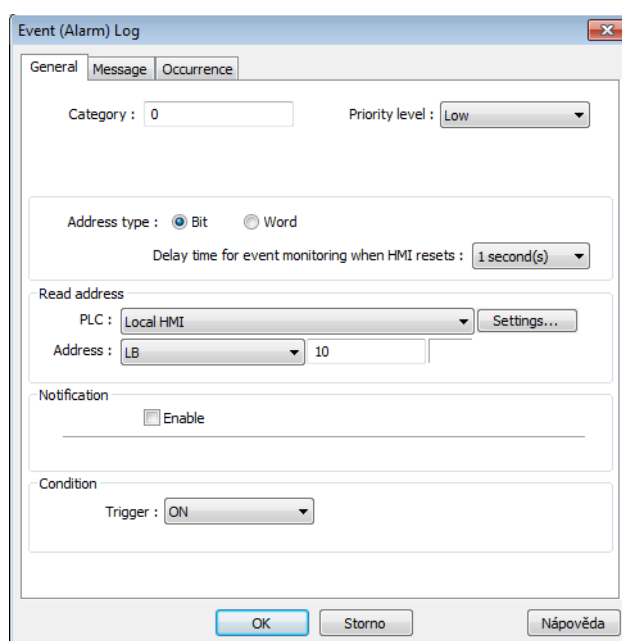
Do předešlého projektu si vytvoříme funkci Event (Alarm) Log. Zde si nadefinujeme proměnné, které chceme sledovat a při jakých podmínkách se mají spustit chyby. Tuto funkci naleznete v menu **Data/History -> Event (Alarm) log**.



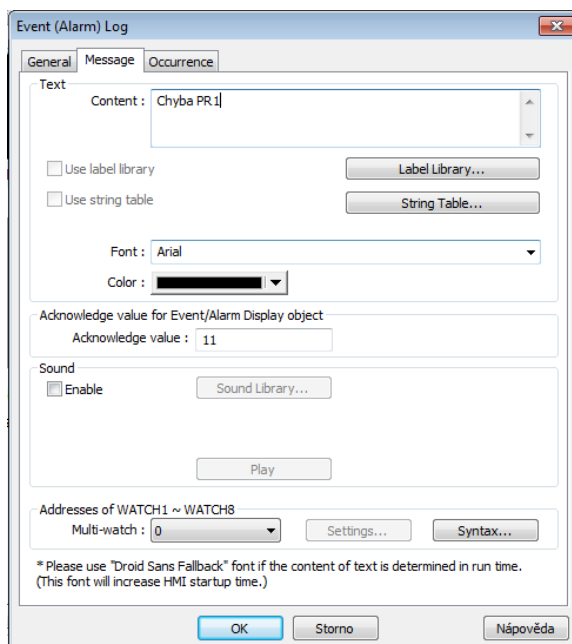
V nově otevřeném dialogovém okně klikneme na tlačítko New a vytvoříme tak novou událost (alarm).



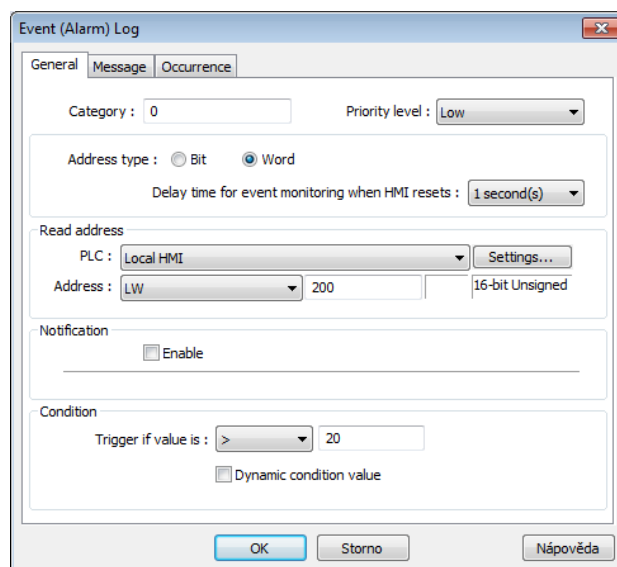
Nejprve nastavíme bitovou poruchu PR1. V záložce General nastavíme jako **Address type Bit**, nastavíme **Read address** na LB-10, jako **Condition** (podmínku) nastavíme ON (lze zvolit mezi ON,OFF,ON->OFF, OFF->ON). Tímto jsme nastavili novou událost, která se uloží v případě kdy se bit LB-10 sepne do stavu ON. Tato událost bude zobrazována, dokud nedojde k uvolnění LB-10 do stavu OFF. Nyní si nastavíme znění zprávy.



V záložce **Message** si nadefinujeme znění samotné chybové hlášky. V textovém poli **Content**: napíšeme znění chyby. V mém případě Chyba PR1. Dále lze nastavit font a barva písma dané chyby. Potvrdíme dané nastavení a zopakujeme ho pro poruchu PR2. Nastavíme **Read address** na LB-11 a textu zprávy napíšeme Chyba PR2.



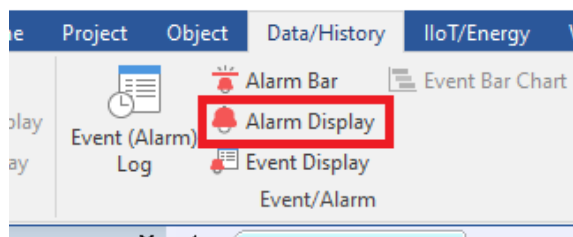
Pro alarm, který se bude spouštět na základě hodnoty v LW-200 musíme zvolit **Address** type Word. V **Read address** nastavíme LW-200. Dále v jako podmínku zvolíme > 20 podle zadání. Poté v záložce **Message** nastavíme text zprávy na Zvýšená hladina.



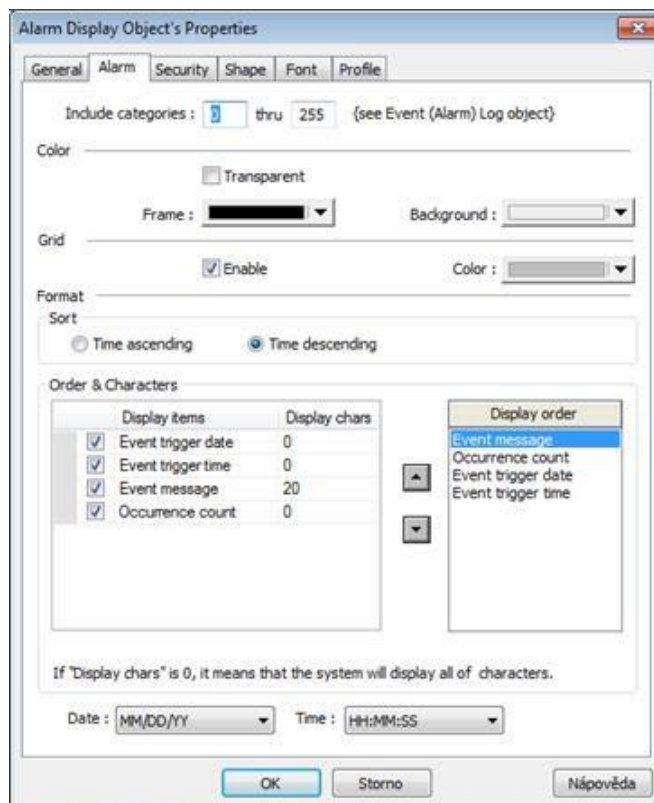
Nyní musíme na obrazovku vložit veškeré prvky pro tento projekt. Nejprve vložíme dva bitové přepínače, které nám budou simulovat poruchy PR1 a PR2. Tyto objekty vložte stejně jako v kapitole výše. Nastavme si u nich, ale Label s popiskem PR1 a PR2. Výsledná tlačítka mohou vypadat následovně.



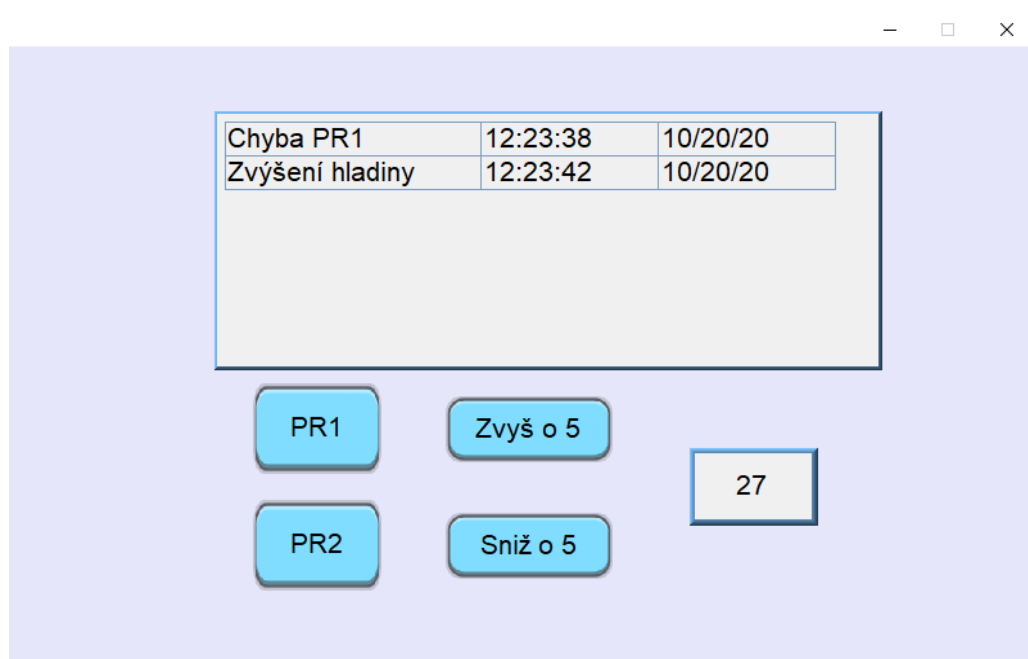
Teď máme na obrazovce veškeré aktivní prvky pro vyvolání vytvořených událostí. Nyní si vytvoříme objekty, které nám dané události (alarmy) zobrazí. Jako první zvolíme **Alarm Display**. Naleznete ho v horním menu Data/History -> Alarm Display.




V nově otevřeném dialogovém okně nás bude zajímat především záložka Alarm. Zde si nahoře můžeme vybrat, které kategorie alarmů chceme zobrazovat. Tyto kategorie se nastavují u vytváření samotných událostí. Lze tak každé události přiřadit jiná kategorie.

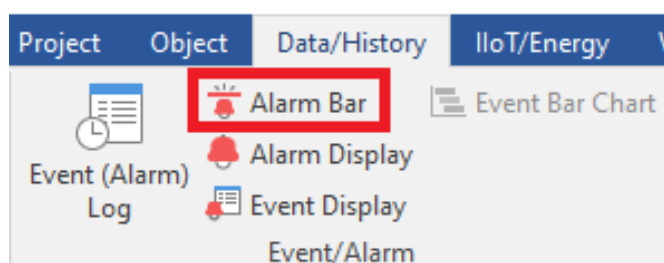


Dále lze nastavit vzhled tohoto objektu. Objekt je zobrazen jako tabulka, lze tedy vybrat barvu pozadí a mřížky a v záložce **Font** i font písma. Další možností je způsob řazení, **Time ascending** nebo **Time descending**, tedy pokud chcete řadit vzestupně či sestupně. Další možností je samotné zobrazení alarmu. Zde můžete zaškrtnout, co vše chcete v tabulce zobrazovat. Je zde na výběr datum, čas, zpráva alarmu nebo počet kolikrát už k alarmu došlo. V pravém sloupci je potom možnost uspořádání sloupců tabulky. Můžete si tedy určit, jak bude daná tabulka vypadat. Vše potvrdíme a na obrazovce se nám vytvoří tabulka. Nyní projekt kompilujeme a spustíme **Off-line** simulaci. Spuštěný projekt tak bude vypadat asi následovně:



Po kliknutí na PR1 nebo PR2 se v tabulce objeví nový řádek s danou chybou, datem, časem a počtem, kolikrát už k chybě došlo. Tato chyba zde bude zobrazena dokud, nedojde k jejímu odstranění. Tedy znovu nestiskneme PR1 nebo PR2. To samé platí o události na hodnotu v LW-200. Dokud bude v LW-200 hodnota vyšší než 20 chyba nezmizí.

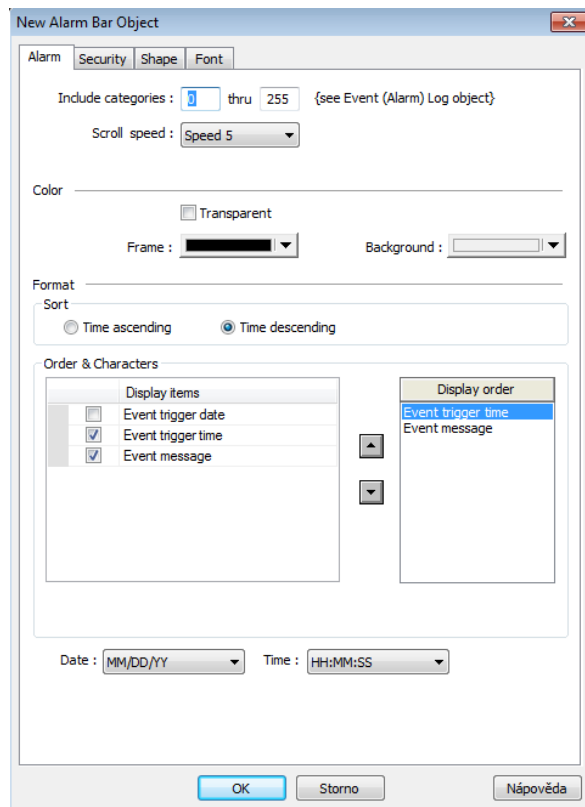
Nyní si přidáme do projektu **Alarm Bar**. Naleznete ho pod ikonkou  nebo opět v horním menu Data/History -> Alarm Bar.



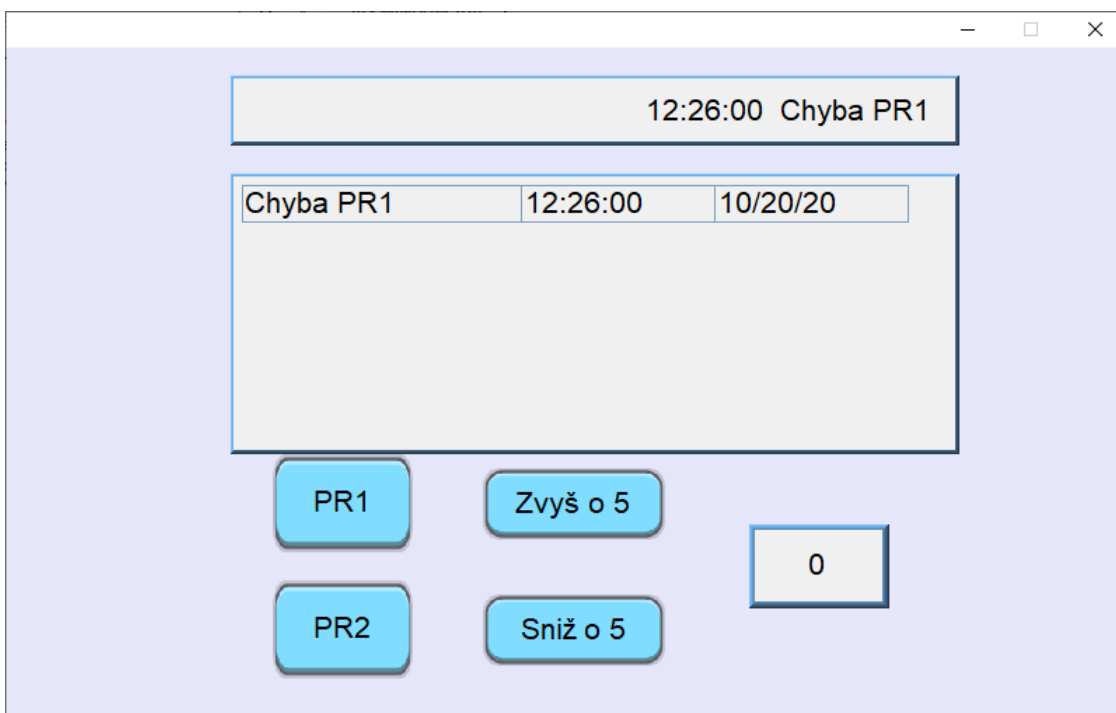
Tento objekt zobrazuje aktuální alarmy v podobě běžícího textu. V horní části lze nastavit, které kategorie alarmů se zde budou zobrazovat. Dále lze nastavit rychlost běhu textu. Dále lze jako u objektu **Alarm display** nastavit řazení vzniklých událostí podle času vzestupně a sestupně.



Opět můžeme vybrat, které informace chceme zobrazovat. Ve spodní části pak můžeme nastavit formát data a času. Vše potvrdíme, zkompilujeme a spustíme **Off-line** simulaci.



Aplikace bude vypadat následovně:

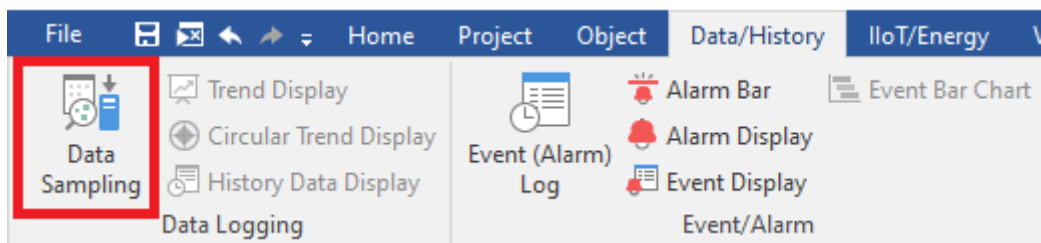


Aplikace ke stažení [zde](#)

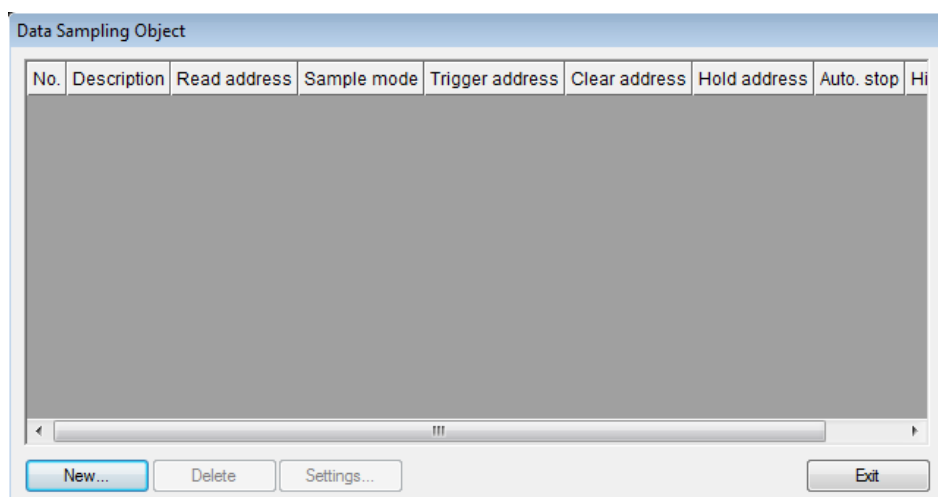
## Logování dat

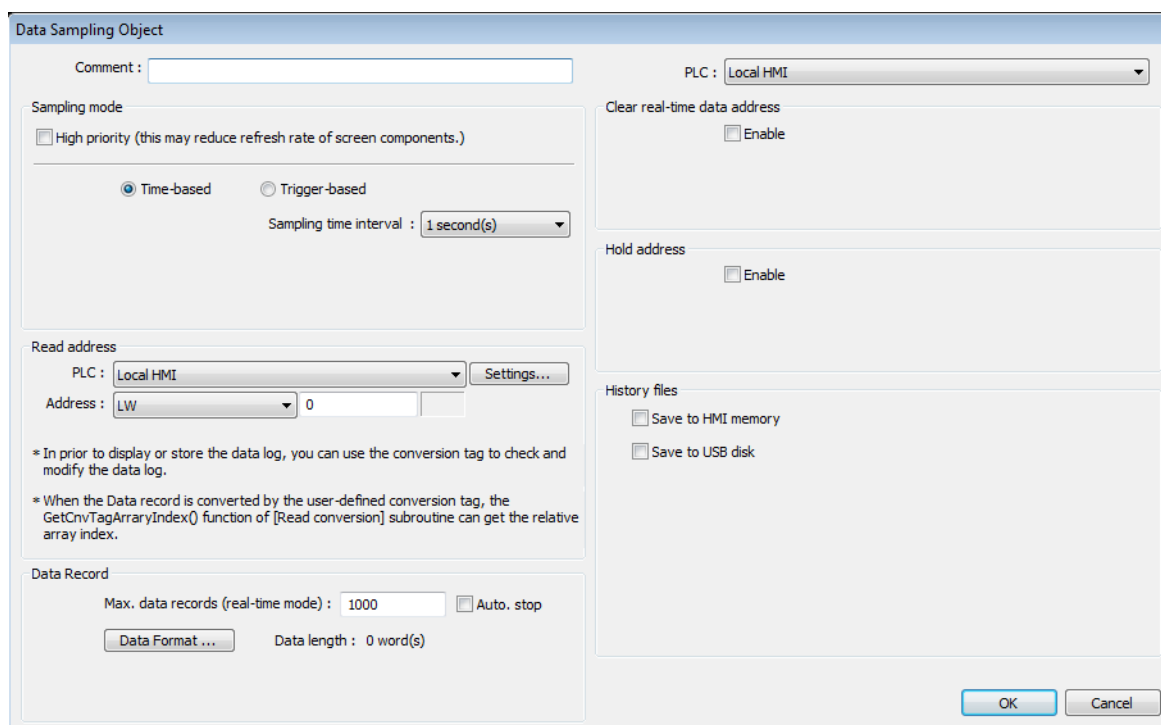
V dalším příkladu si ukážeme, jak vytvořit tzv. **logování dat**. Tedy ukládání načtených dat do paměti panelu nebo na externí zařízení (USB disk).

Založíme si nový projekt. Abychom mohli logovat data, potřebujeme vytvořit tzv. **Data Sampling Object**. Nalezneme ho v menu Data/History -> Data Sampling.



V nově zobrazeném dialogovém okně zvolíme možnost **New...** .

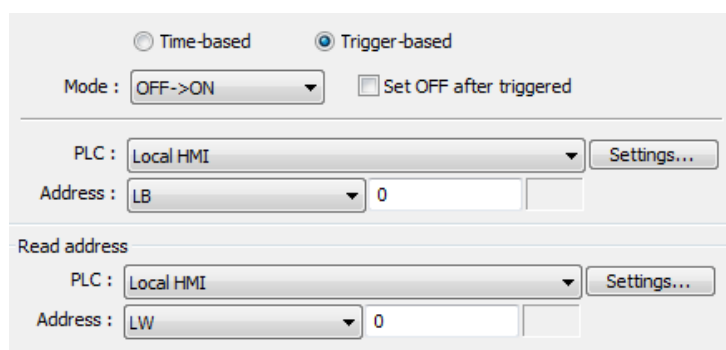




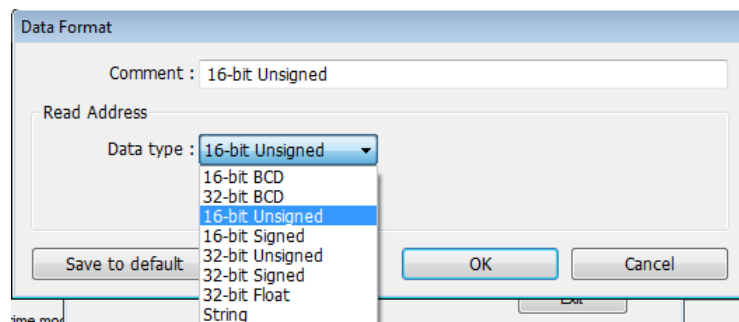
Nejprve si popíšeme nově otevřené okno. V levé části okna vybíráme, kdy a jaká data se budou logovat. V pravé části můžeme nastavit kontrolní adresy, které mohou ovládat logování dat, a také kam a jak se budou data ukládat. Začneme tedy s levou částí, kde specifikujeme, jaká data chceme ukládat.

V levé horní části vybereme způsob, kterým data budeme načítat. Jsou zde dvě možnosti **Time-based** a **Trigger-based**, tedy na základě času nebo změny libovolné proměnné. Možnost **Time-based** bude každý nastavený interval načítat a ukládat vybraná data. Tento interval lze nastavit v rozsahu 0.1 sekundy – 120 minut.

Možnost **Trigger-based** nám umožní zvolit proměnnou, která bude řídit čtení dat. Tato proměnná může být v panelu nebo na straně připojeného zařízení. Pro tuto proměnnou lze zvolit mód změny: OFF->ON, ON->OFF, ON<->OFF.

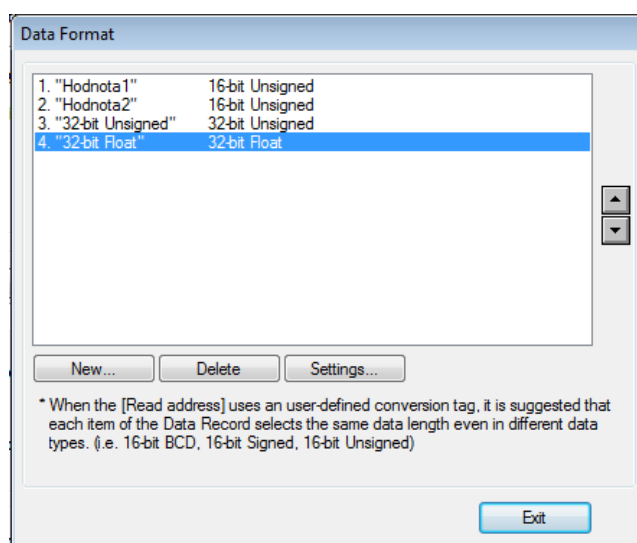


Nyní se dostáváme k samotným datům, které chceme logovat. U možnosti **Read address** zvolíme počáteční adresu, odkud budeme data načítat. Ve spodní části **Data record**, pak zvolíme kolik dat v tzv. **Real-time modu** chceme ukládat. Jedná se o počet dat, které nebudou ukládány do paměti panelu, ale pouze do mezipaměti a po překročení tohoto limitu se budou nejstarší data mazat a nahrazovat novými tak, aby byl zachován maximální počet dat. Nyní musíme zvolit, jaká data se budou načítat. Klikneme na **Data Format**.

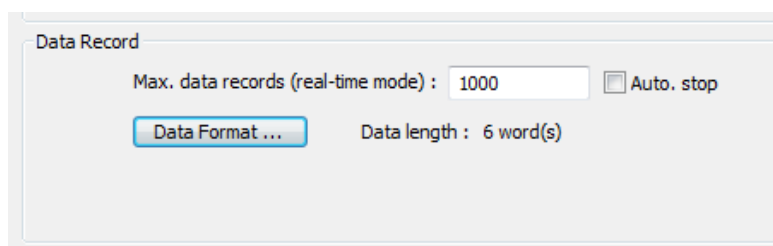


V nově zobrazeném dialogovém okně budeme přidávat formáty daných proměnných, které se budou načítat. Vždy klikneme na **New** a vybereme, jakého datového formátu dané hodnoty budou. Každou hodnotu si můžeme okomentovat a přiřadit jí tak název.

Poté co jsme hodnoty přidali, může jejich výpis vypadat třeba následovně:

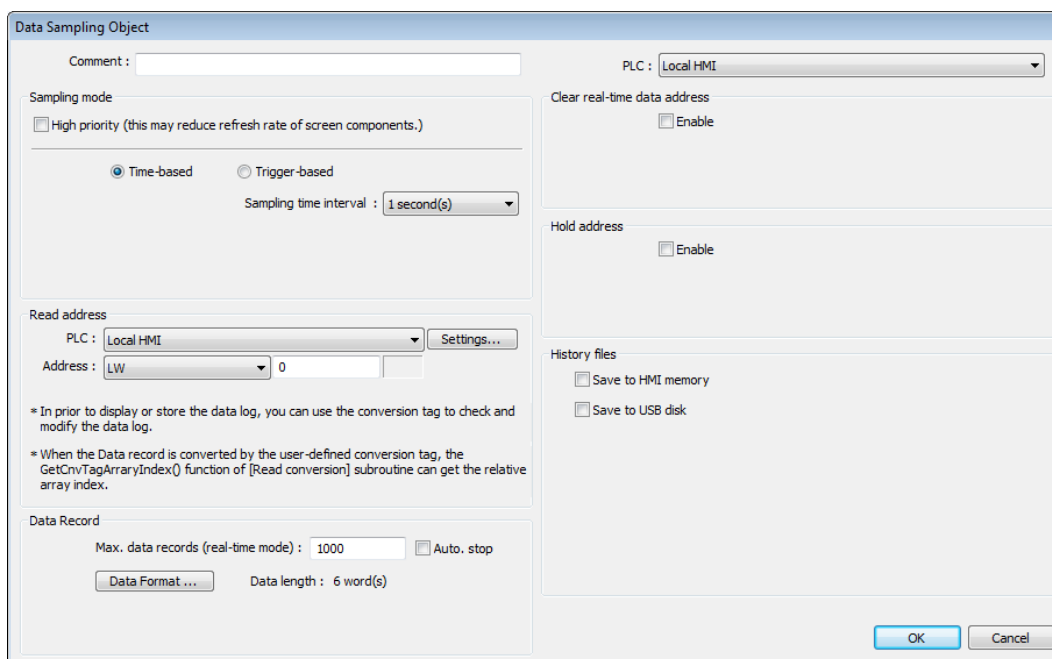


Zavřeme toto okno a ujistíme se, zda se v sekci **Data record** objevila velikost dat.

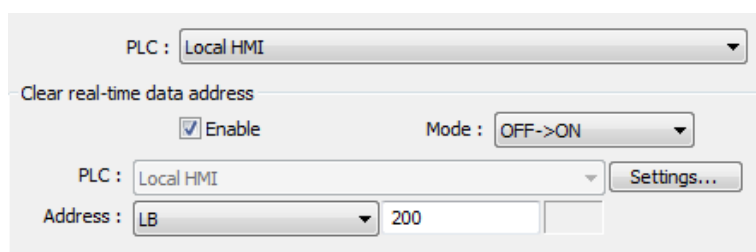


V mém případě je **Data length** 6 slov. Budu tedy načítat přesně 6 16-bitových slov od počáteční adresy, kterou jsem nastavil výše. Maximální počet dat je nastavený na 1000. Data se tedy budou ukládat do mezipaměti a po dosažení maximálního počtu (1000) se nejstarší data budou nahrazovat nejnovějšími.

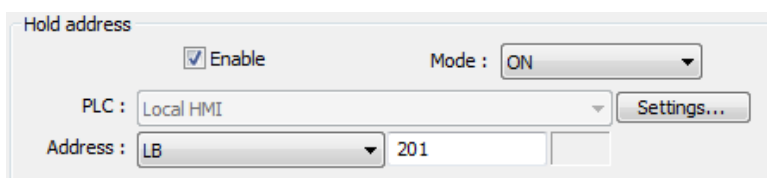
Nyní se dostáváme k pravé části dialogového okna objektu **Data Sampling**. V pravé části můžeme nastavit bity pro jednoduché ovládání dat (mazání real-time dat, pozdržení ukládání), dále můžeme vybrat možnost ukládání dat. Jak jsem zmínil výše, zatím jsme nastavili logování dat do mezipaměti, tak že se data po překročení maximálního počtu, budou přepisovat novějšími. Zde tedy můžeme zvolit ukládání do paměti panelu nebo na USB disk. Nelze obojí.



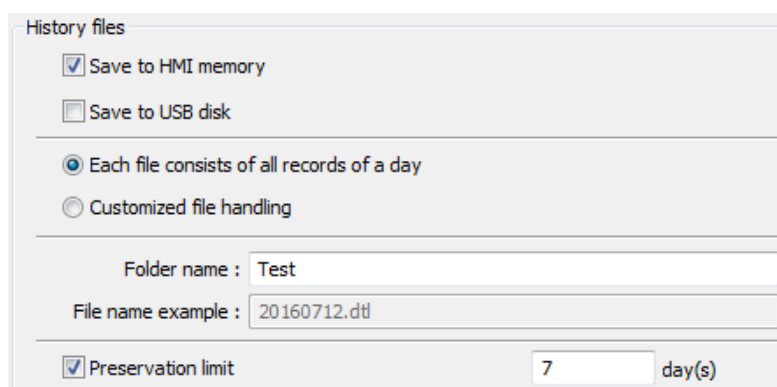
Začneme první částí v pravé horní polovině. Zde je možnost PLC:. Zde si vybíráme kdo bude ovládat daný data sampling (mazání real-time dat, pozdržení logování). Na výběr je vždy Local HMI, tedy vnitřní paměť panelu, dále bude k dispozici jakékoli PLC, které je připojené. Pod touto možností je volba mazání real.time dat.



Zaškrtneme-li **Enable** zobrazí se možnost vybrat ovládací bit z paměti panelu. Následně i mód při, kterém se data smažou (ON -> OFF, OFF -> ON). Pokud tedy dojde ke změně, v mém případě bitu LB 200, z OFF -> ON, smažou se veškerá nalogovaná real-time data.



Další možností je **Hold address**. Pokud zaškrtneme možnost **Enable**, zobrazí se možnost vybrat ovládací bit z paměti panelu. Následně i mód při, kterém se data smažou (ON, OFF). Pokud nastavíme LB 201 na ON data se přestanou logovat, tedy ukládat do paměti i mezi paměti. Pokud v LB 201 bude OFF, data se budou standardně logovat.

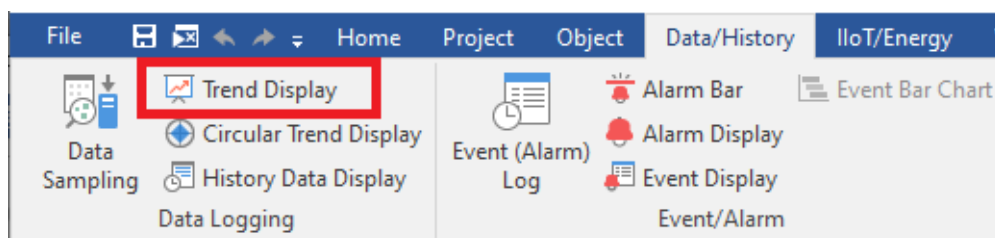


Poslední možností je samotné ukládání dat. Jak bylo napsáno výše je zde výběr mezi **HMI memory** (paměť panelu) a USB diskem. Nelze vybrat obojí. Další možností je, zda se bude soubor s daty ukládat každý den nebo podle uživatelského nastavení. Volba **Preservation limit** umožňuje nastavit maximální počet souborů, které se budou ukládat. Pokud dojde k jeho překročení, opět se budou starší soubory nahrazovat novějšími, tak aby byl zachován tento limit.

Vše potvrdíme a první **Data Sampling** je nyní vytvořený. Bude tedy načítat hodnoty od LW-0, přesně 6 slov. Nastaven je na sbírání dat každou vteřinu. Maximální počet real-time dat je 1000. Tuto hodnotu real.time data nepřekročí. Hodnoty se budou ukládat i do paměti panelu, zde je omezení ve formě počtu souborů, které se budou vytvářet podle nastavení. V našem případě jednou za den s limitem 7 souborů (dnů). Historické data budou ukládány 7 dnů zpětně. K těmto datům lze přistupovat pomocí objektů **Trend Display**, **History Data Display**. Dále pak pomocí ftp nebo lze tyto data vyexportovat pomocí objektu **Backup** na externí zařízení USB disk.

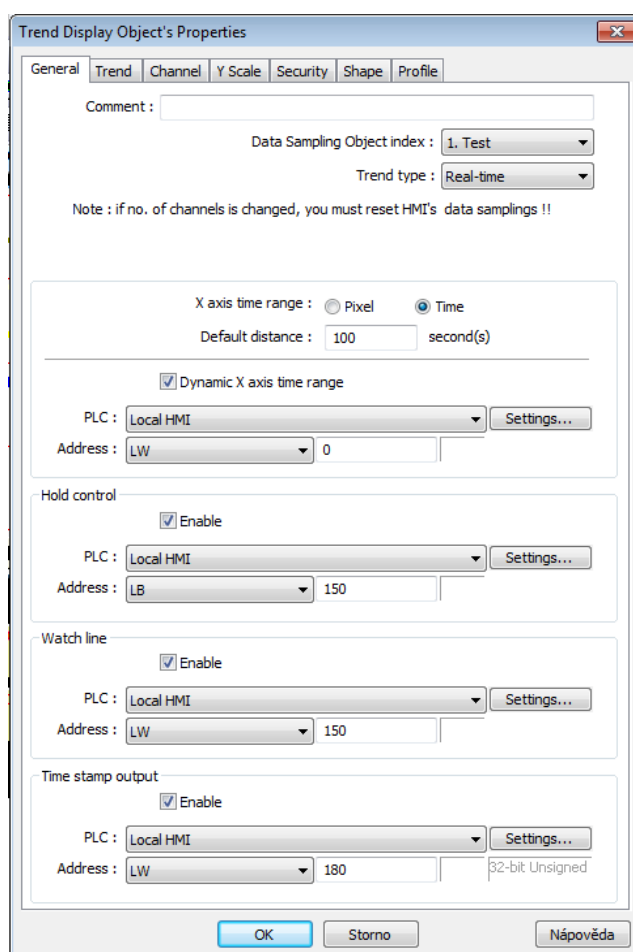
Nyní si ukážeme možnost zobrazení dat pomocí grafu (objekt Trend Display). Abychom mohli použít tento objekt, musíme mít vytvořený **Data Sampling**.

**Trend Display** nalezneme v menu **Data/History -> Trend Display**.



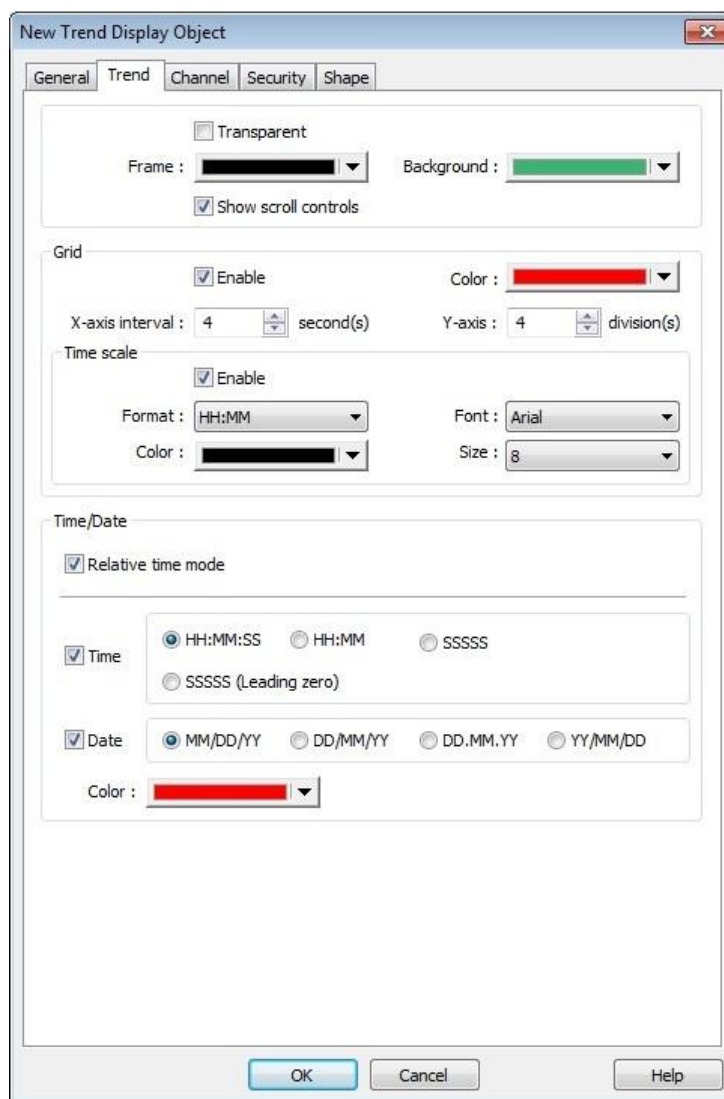
V nově otevřeném dialogovém okně budeme nastavovat jaká data a jakým způsobem chceme zobrazovat.

Začneme záložkou **General**. Nejprve vybereme Data Sampling Object index, tedy sampling, který jsme si vytvořili. Dále zvolíme typ trendu. Zda chceme zobrazovat **Real-time** data či **History**. Jako další možností je určit jakého typu bude X-osa. Lze vybrat možnost **Pixel** nebo **Time**. Pixelová osa nám udává počet dat, které se budou zobrazovat, oproti tomu časová nám zobrazí data za určitý časový úsek. U obou typů lze zaškrtnout **Dynamic X axis range**. Lze tedy libovolně měnit rozsah zobrazovaných dat. Dále lze u typu **Real-time** zvolit **Hold control**. Jedná se o bit, který když bude sepnutý tak zastaví zobrazování dat. Po jeho uvolnění se zobrazí data, která byla pozdržena a zobrazování probíhá stejně jako před tím. Možnost **Watch Line** umožní po kliknutí na libovolné místo na grafu vyčíst dané hodnoty. Zde se nastavuje počáteční adresa. V našem případě je nutné mít prostor 6 slov.



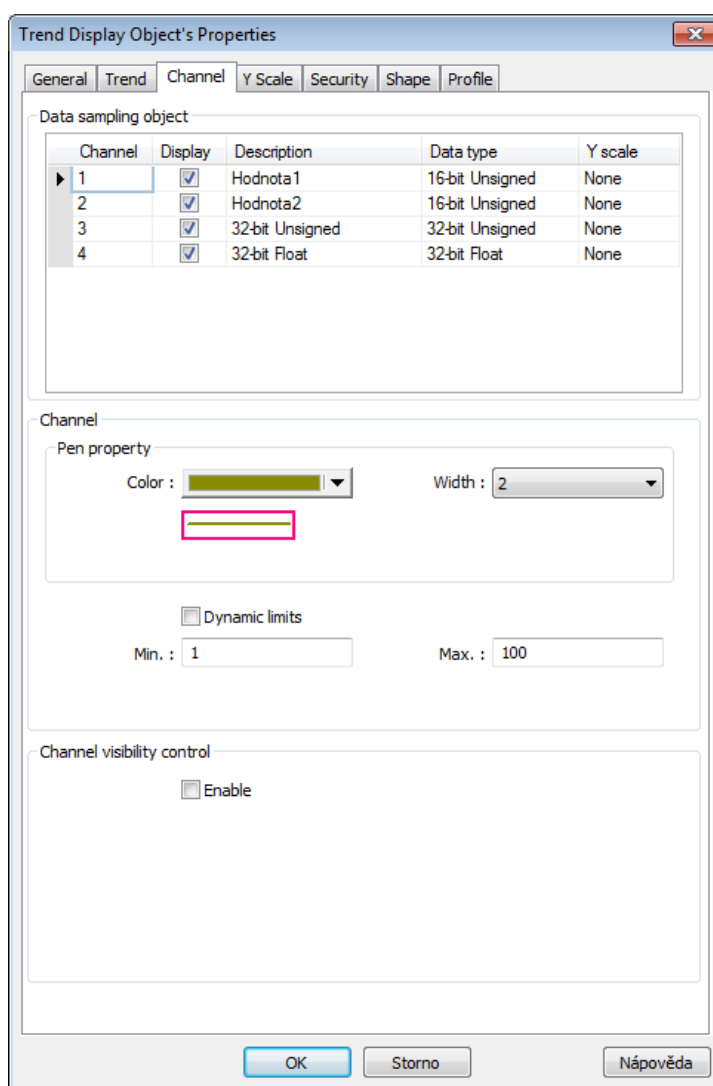
Tato funkce tedy funguje tak, že pokud chceme získat hodnoty z grafu v určitém místě, klikneme na toto místo a hodnoty z grafu (podle počtu čar-kanálů) se naplní od adresy LW 150. Poslední možností je **Time stamp output**, tato funkce zapíše čas na danou adresu. Tento čas reprezentuje čas vzorku na, který bylo kliknuto. Pokud tedy používáme **Watch Line** a potřebujeme zjistit, v jakém čase byly dané vzorky uloženy, nalezneme tento čas právě v dané adrese. Je nutné počítat s tím, že čas je ukládán v 32-bitovém formátu.

Další záložkou je záložka **Trend**. V této záložce se nastavuje samotný vzhled grafu (trendu). Nejprve můžeme nastavit pozadí a ohraničení grafu, nebo můžeme zvolit možnost transparent. Dále nastavujeme mřížku v grafu. Mřížka v grafu být může, ale nemusí. Pokud ji potřebujeme lze nastavit počet vertikálních i horizontálních čar včetně barvy. Dále můžeme nastavit, pokud jsme zvolili v záložce **General** X-axis range jako Time, formát časové osy, včetně fontu, barvy a velikosti písma. Dále můžeme nastavit zobrazení aktuálního času v horní části grafu. Můžeme volit z více formátů zápisu.

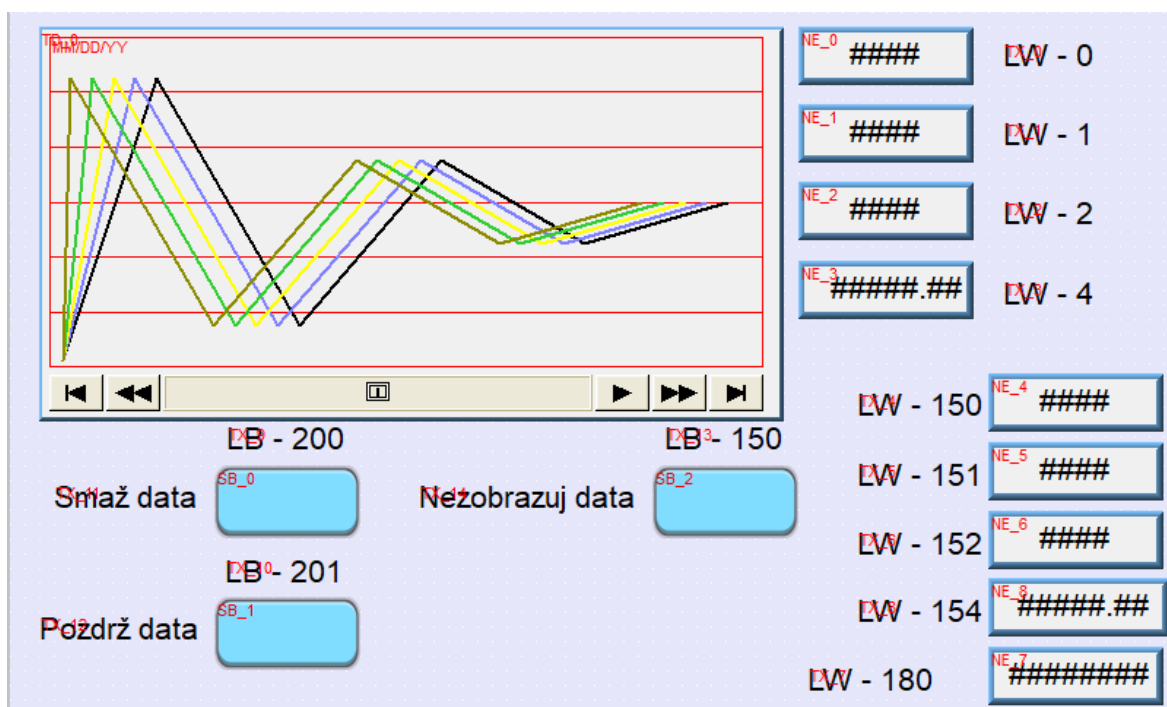




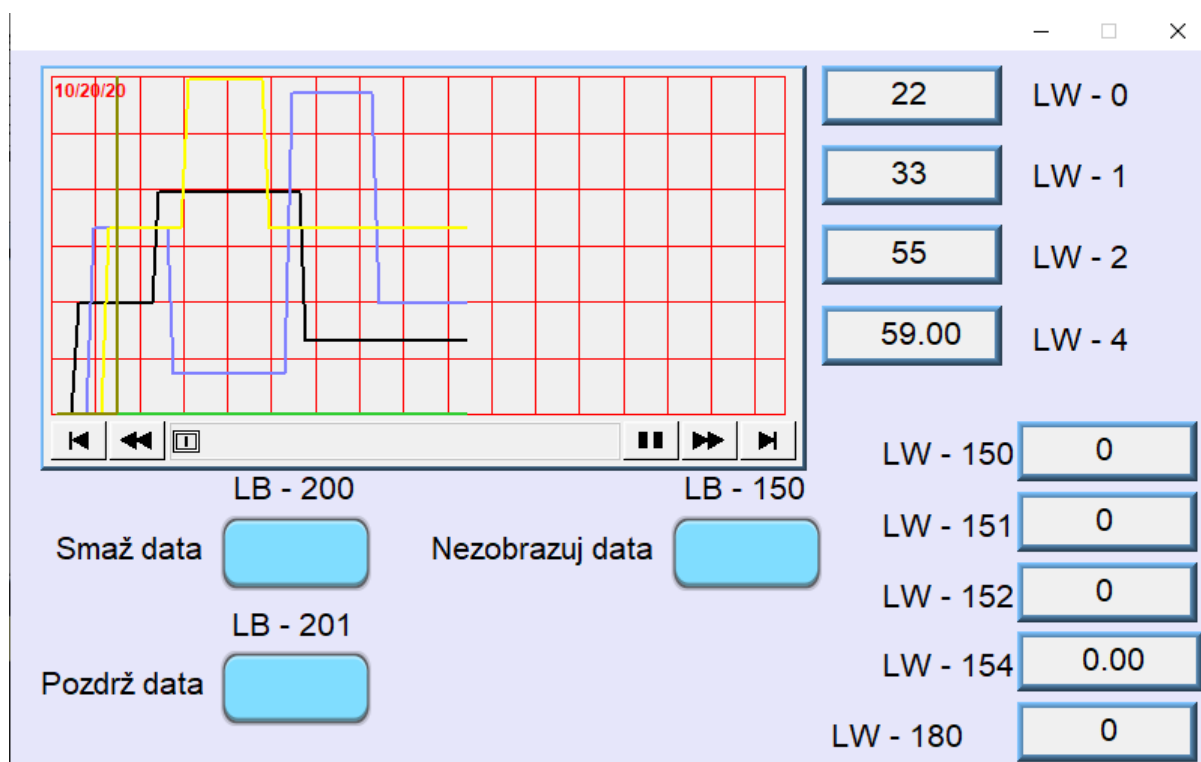
Nyní se dostáváme k záložce **Channel**. V této záložce budeme nastavovat, jak se budou zobrazovat data z **Data Samplingu**. V horní části máme vypsáno všechny proměnné, které jsme si nadefinovali v **Data Samplingu**, pokud je chcete zobrazit, musí být zaškrtnut **Display**. Pro každý **Channel** můžeme zvolit libovolnou barvu, styl a tloušťku čáry. Dále pak její limity. Klikneme na libovolný **Channel** a spodní část se týká právě jeho. Tedy nastavení čáry a limitů. Poslední možnost **Channel visibility Control** je společná pro všechny **kanály**. Po zaškrtnutí **Enable** se zobrazí možnost vybrat libovolné slovo (z panelu či automatu), které bude určovat, které **kanály** budou zobrazeny a které ne. Jednotlivé **kanály** odpovídají jednotlivým bitům tohoto slova v pořadí, jakém jsou uvedeny výše. Zobrazit **kanály** lze buďto při hodnotě ON nebo OFF. Čili pokud bychom chtěli zobrazit pouze **Channel 2** a **3** zapíšeme do slova hodnotu 6 (0110) resp. 9 (1001) podle módu (ON či OFF)



Takto vytvořený graf (**Trend Object**) vložíme na obrazovku. Přidáme veškeré kontrolní objekty. Výsledná aplikace může vypadat následovně:



LB-200 jsme nastavili tak, že bude mazat real-time data. LB-201 zastaví logování dat, jakmile se uvolní, data se budou znovu logovat. Ve slovech LW-0 až LW-4 jsou data, které logujeme. V LW-150 až LW-154 budou vypsaná konkrétní data po kliknutí na libovolné místo v grafu. V LW-180 bude aktuální čas těchto dat. LB-150 pozastaví zobrazování dat v grafu, po jeho uvolnění se data začnou opět zobrazovat. Spustíme **Off-line** simulaci. Výsledná aplikace bude vypadat takto:

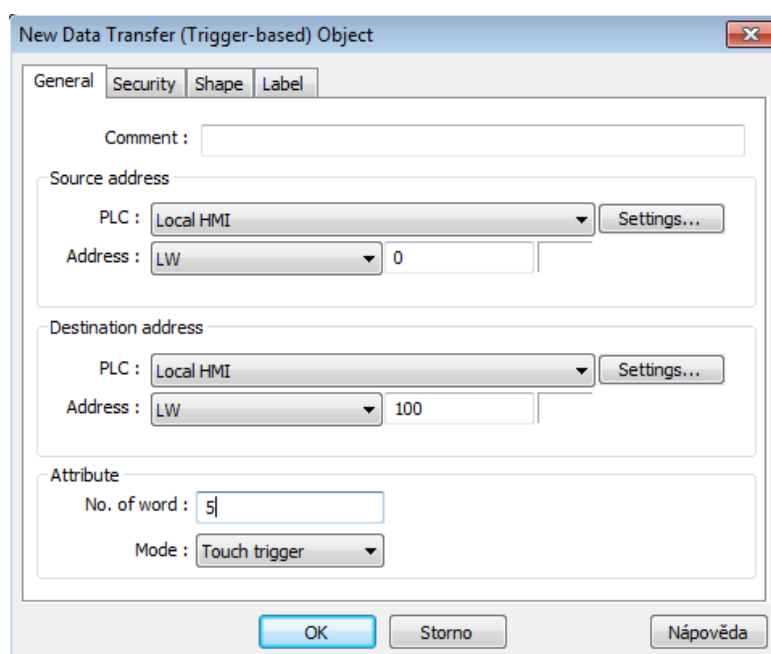
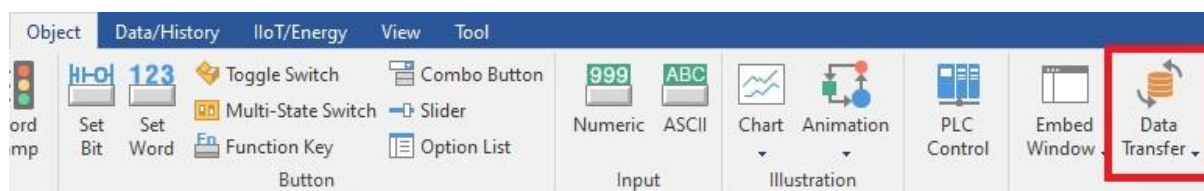


Aplikace ke stažení [zde](#)

## Data transfer

Jestliže potřebujeme posílat větší množství dat mezi panelem a připojeným zařízením, použijeme k tomu objekt **Data Transfer**. Tento objekt slouží k přenosu dat v nastaveném rozsahu. Tento objekt se spouští dvěma způsoby a to, na základě stisku nebo periodicky podle nastaveného času. Data, která chceme přenést, musí být vždy seřazena v paměti za sebou. Data se zde nastavují podle počáteční adresy a množství dat. Nyní si ukážeme jak takový objekt nastavit.

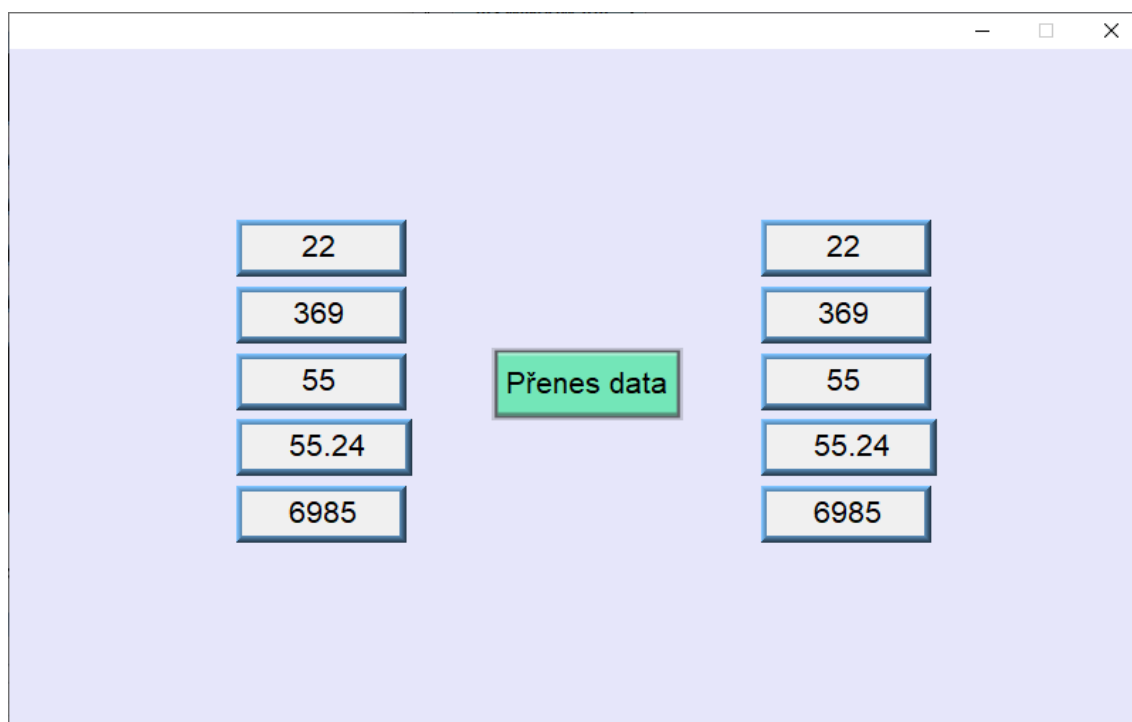
**Data Transfer (Per page)** naleznete v menu Objects -> Data Transfer (Per Page).



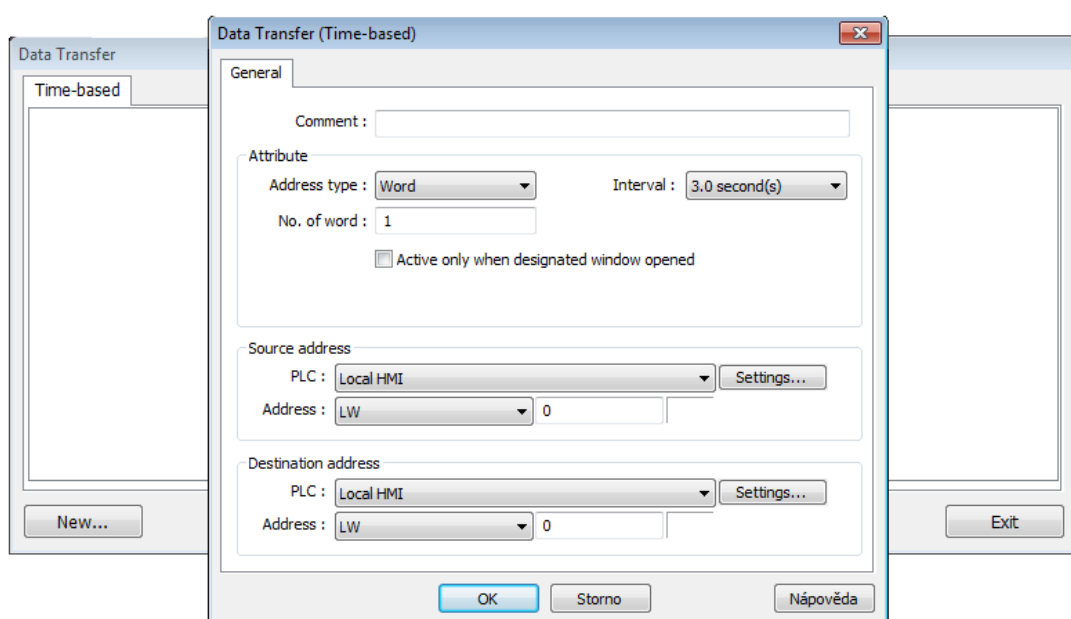
V nově otevřeném okně nastavujeme zdrojovou a cílovou adresu. Dále pak počet slov, které chceme přenést. Nastavíme zdrojovou (Source) adresu na LW-0 a cílovou (Destination) adresu na LW-100. Počet slov nastavíme na 5. Pod nastavením počtu slov nastavujeme mód. Tento mód může být **Touch trigger** nebo **External trigger**. V případě **Touch trigger** se vytvoří tlačítko, které po stisku přenesse data. Pokud zvolíme možnost **External trigger**, můžeme zvolit libovolný bit na straně panelu nebo připojeného zařízení, který po nastavené změně (ON -> OFF, OFF -> ON, ON <-> OFF) také přenesse data. Pokud zvolíme Touch trigger nastavíme tvar a popis tlačítka v záložce Shape resp. Label.

Vše potvrdíme a vložíme vytvořené tlačítko na obrazovku. Vytvoříme k na obrazovce 10 Numeric objektů LW0 – LW4 a LW100 – LW105. Uložíme a spustíme Off-line simulaci.

Výsledná aplikace může vypadat následovně:



**Data Transfer (Global)** naleznete na stejném místě jako Data Transfer (Per page).



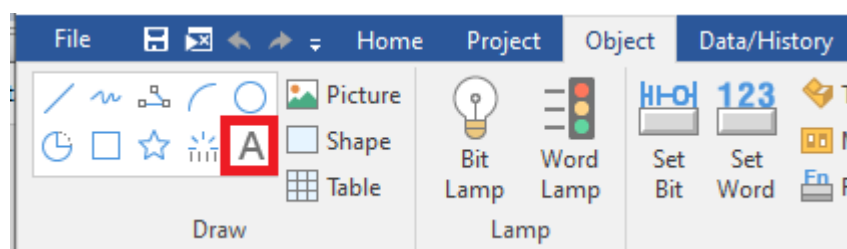
Zde se nastavuje, jaký typ dat budeme přenášet, zda Bity nebo Wordy. Dále pak časový interval po kterém se data vždy přenesou. Data se přenáší stejně jako u trigger-base. Tedy od zdrojové adresy určený počet slov. Nastavíme tedy zdrojovou a cílovou adresu. Tímto je objekt nastaven a vše potvrdíme. Daný objekt nám bude přenášet data ze zdrojové adresy na cílovou v určitém časovém intervalu.

Aplikace ke stažení [zde](#)

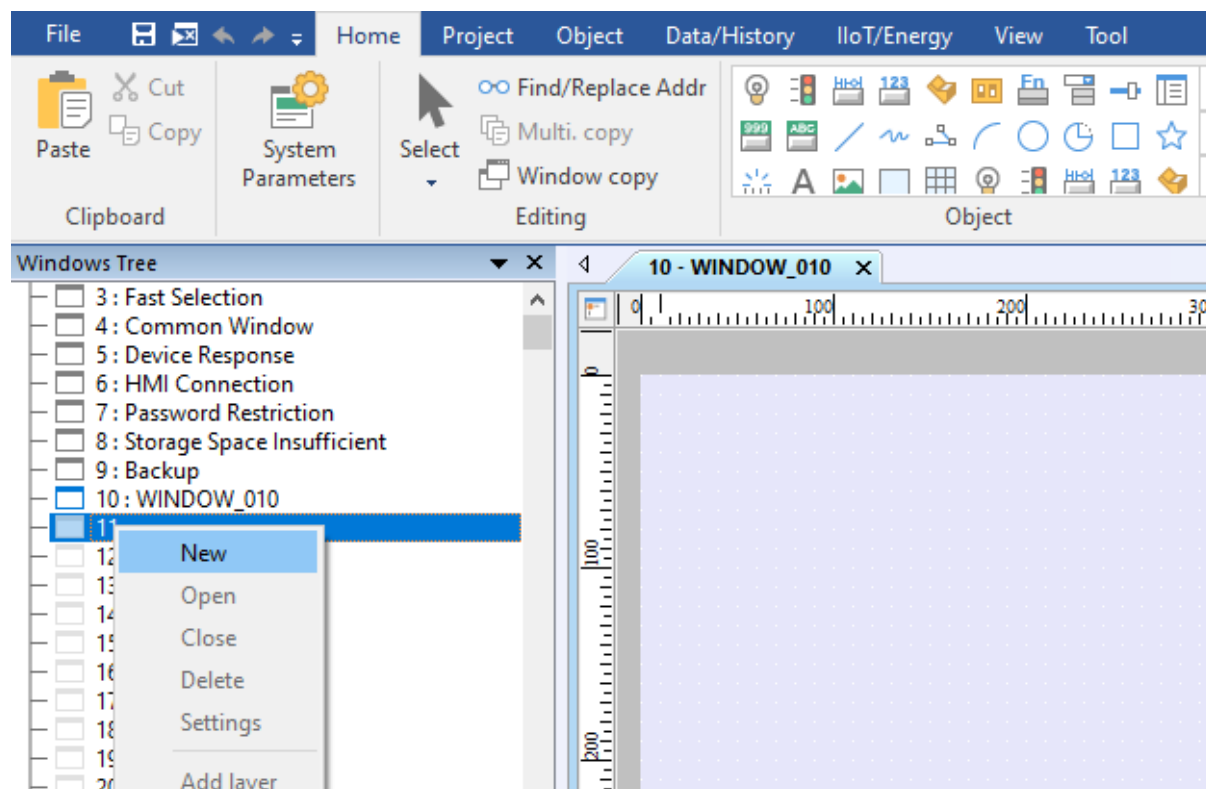
## Více oken na panelu

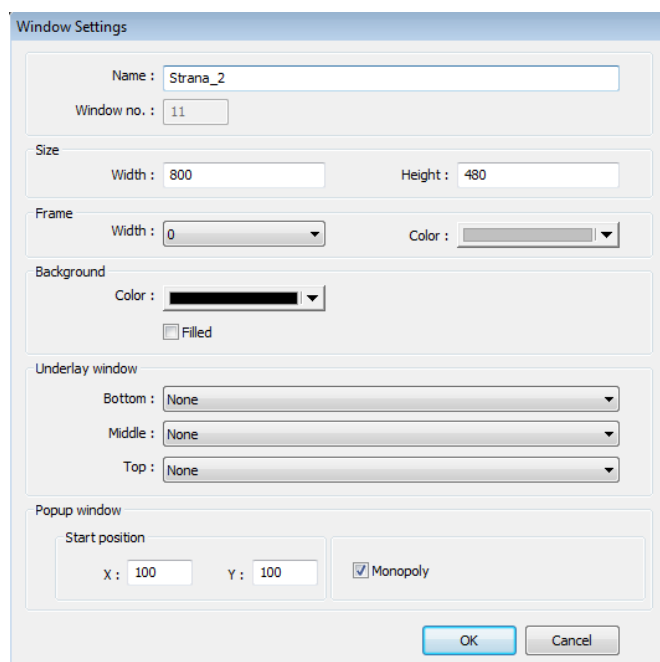
Pokud děláme větší aplikaci a nestačí nám pro zobrazení všech prvků jedna obrazovka, je zapotřebí vytvořit obrazovek více a přepínat mezi nimi. Nyní si ukážeme jak vytvořit nové obrazovky a jak mezi nimi přepínat.

Založíme si nový projekt. Poté si na úvodní obrazovku umístíme nápis „Hlavní Obrazovka“. Použijeme objekt **Text**, naleznete ho v menu Object -> Text.



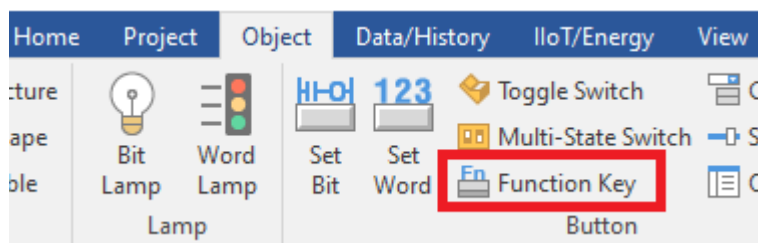
Nyní si vytvoříme novou obrazovku. V levé části vedle hlavní obrazovky se nachází výpis obrazovek. Hlavní obrazovka má název **WINDOW\_010**, klikneme pravým tlačítkem myši na 11 a zvolíme **New**.



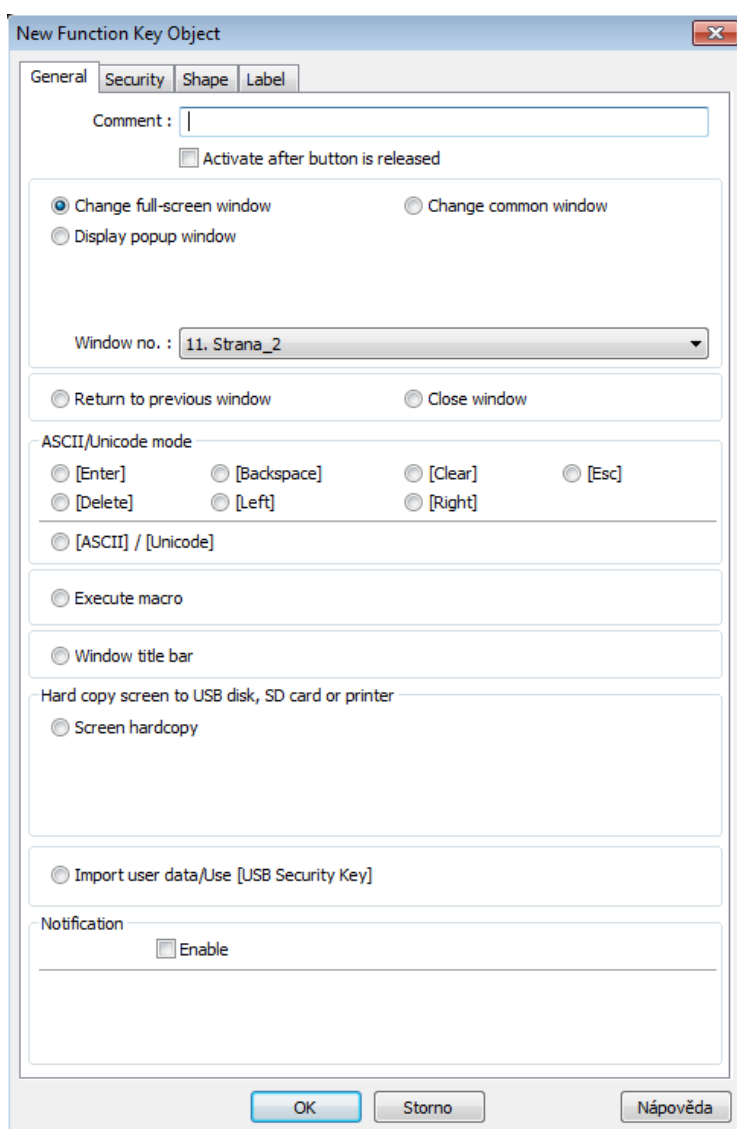


V nově otevřeném dialogovém okně si obrazovku pojmenujeme Strana\_2. Zde lze nastavit i velikost dané obrazovky, její pozadí atd.. Vše potvrdíme OK.

Rozklikneme novou obrazovku Strana\_2. Zobrazí se stejná černá plocha jako u hlavní stránky. Použijeme znovu objekt Text a napíšeme na tuto obrazovku Strana 2. Nyní máme vytvořenou druhou obrazovku. Klikněte zpět na hlavní obrazovku. Teď si ukážeme, jak mezi těmito obrazovkami listovat. K tomuto účelu slouží objekt **Function Key** nalezneteho v menu Objects -> Function Key.

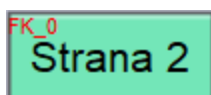


Otevře se dialogové okno.

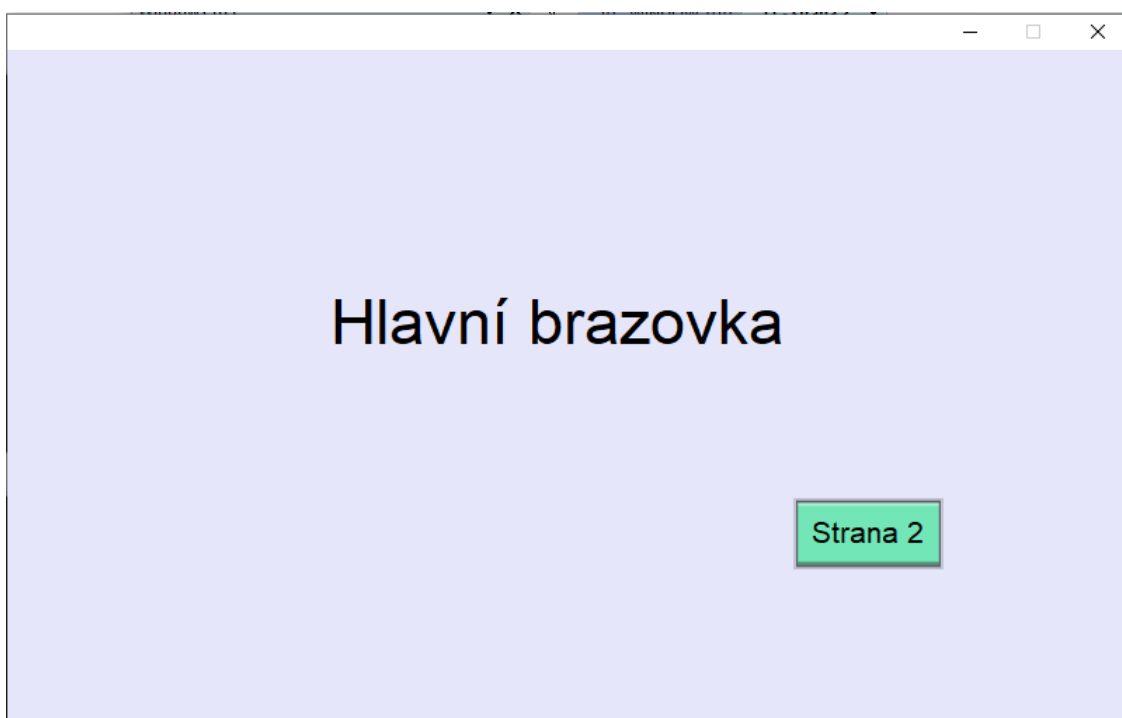


Nás bude zajímat především možnost **Change full-screen window**. Poté stačí vybrat obrazovku, na kterou chceme přejít. Zvolíme tedy 11. Strana\_2. V záložce **Label** si nadefinujeme text „Strana 2“. Umístíme tento objekt na hlavní obrazovku.

Nyní provedeme tu samou operaci na obrazovce Strana\_2. Zvolíme **Function Key** a jako obrazovku, na kterou chceme přejít, zvolíme WINDOW\_010. Opět použijeme záložku **Label** a nadepíšeme si ji Hlavní obr. .



Výsledný projekt bude vypadat následovně:



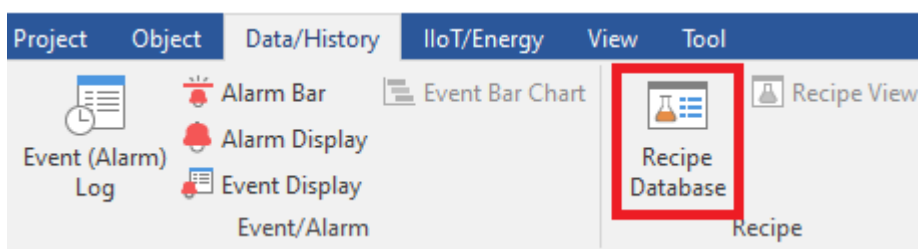
Projekt zkompilujeme a pustíme v **Off-line** simulaci. Obrazovky se budou měnit po stisku daného tlačítka. Tímto způsobem je možno vytvořit několik obrazovek a listovat mezi nimi.

Aplikace ke stažení [zde](#)




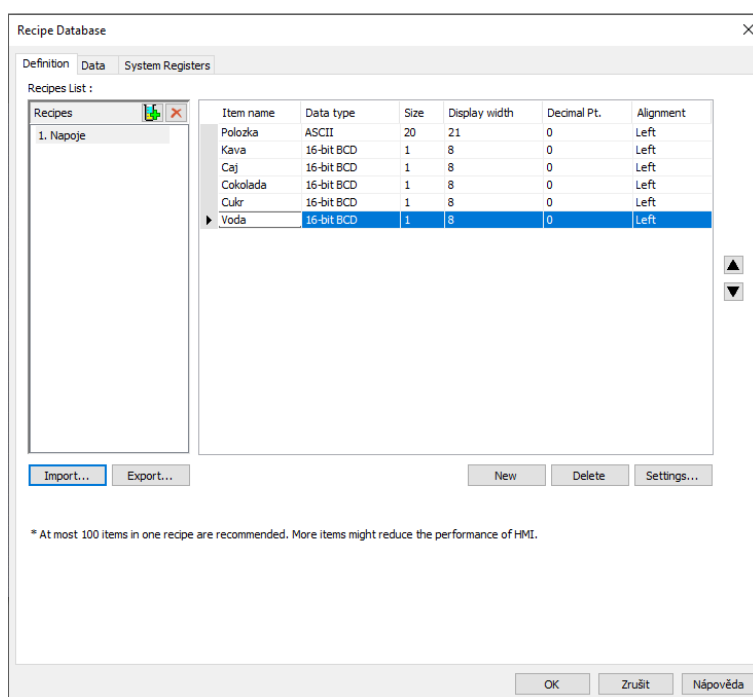
## Receptury

**Receptury** slouží k přiřazování opakovatelných hodnot. Představme si to na příkladu. Vezmeme si pivovar, který vaří různé druhy piva. Každý druh piva se skládá z určitých složek, které se liší pouze množstvím. Pokud by tedy řídicí systém řídil tuto výrobu, bylo by pro něho snazší, kdyby měl tyto informace o hodnotách jednotlivých surovin uložené v určité databázi. Mohl by je tak načíst všechny najednou. K tomu lze použít právě receptury, které vyberou **dané hodnoty z databáze** a naplní jimi určité systémové proměnné, které řídí výrobu. Ukážeme si na **příkladu automatu na nápoje**, jak pracovat s recepturami. Založíme si nový projekt. První, co v novém projektu vytvoříme, bude právě databáze receptur. Seznam receptur nalezneme v menu Data/History -> Recipe Database

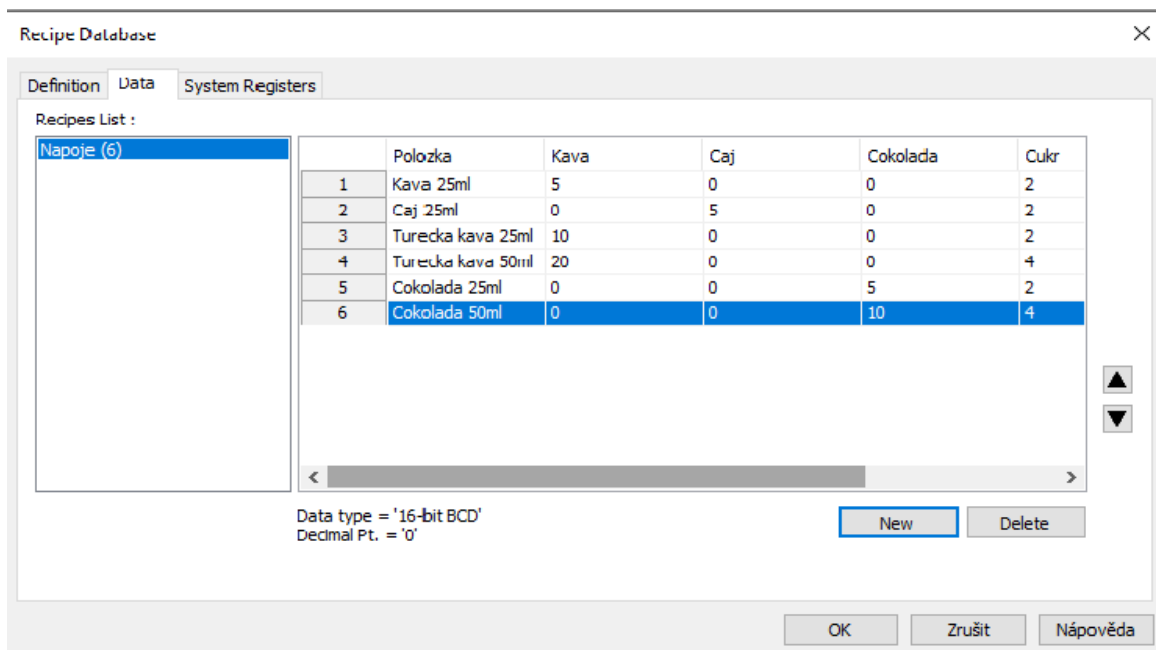


V nově otevřeném dialogovém okně vybereme záložku Recipes. Zde budeme vytvářet seznam receptur.

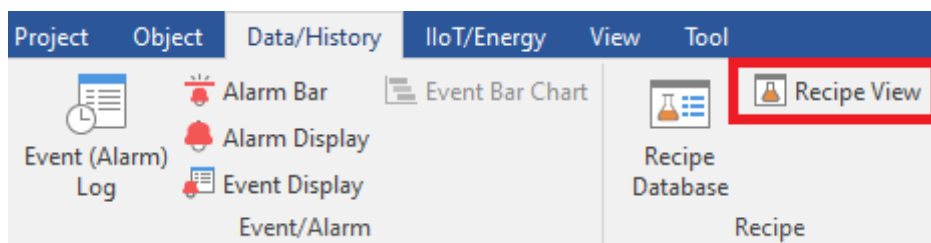
Klikneme na ikonku , abychom vytvořili **novou recepturu**. Nazveme ji Napoje. Dále budeme vytvářet jednotlivé složky dané receptury. V našem případě složky nápojů. Klikneme na tlačítko New a vložíme tyto položky Položka, Kava, Caj, Cokolada, Cukr, Voda. Nejedná se o samotné nápoje, ale pouze o složky, z nichž se nápoj bude skládat. U Položka zvolíme datový typ ASCII s velikostí 20, jelikož to bude text. Ostatní necháme jako 16-bit Unsigned.



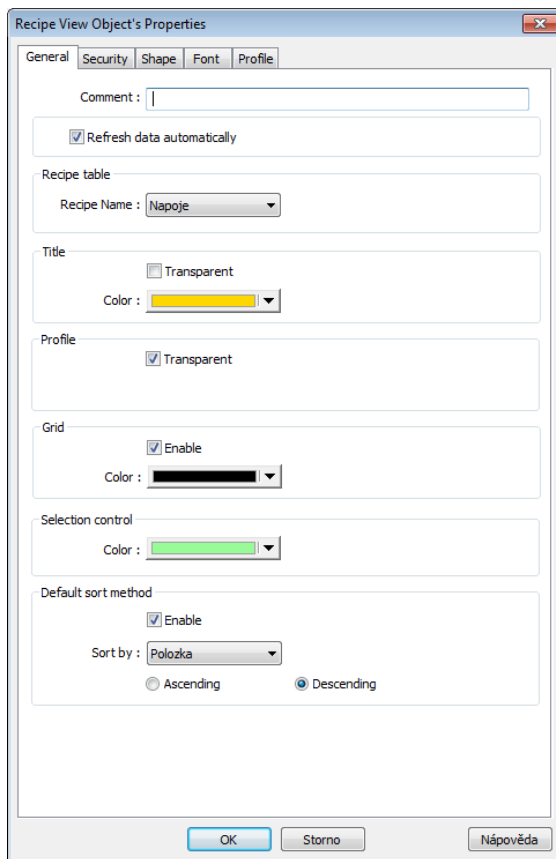
Nyní máme vytvořené jednotlivé **položky receptury** a nyní je můžeme naplnit daty. K tomu slouží záložka Data. V závorce je počet záznamů této receptury. Klikneme na Add a budeme vkládat jednotlivé nápoje a jejich složení. Např. Kava 25ml složení 5 dílků kávy, 2 cukru a 10 vody, Caj 25ml 5 dílků čaje, 2 cukru a 10 vody atd. Těchto nápojů můžete vytvořit několik.



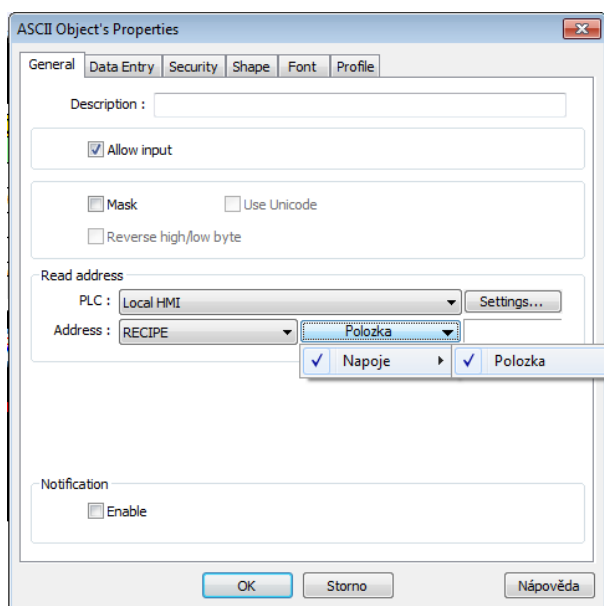
Nyní si ukážeme, jak s těmito daty pracovat. Vytvoříme si na obrazovce několik objektů, které nám k tomu poslouží. Začneme výpisem položek receptury. Zvolíme objekt **Recipe View**, který nalezneme v menu Data/History -> Recipe View



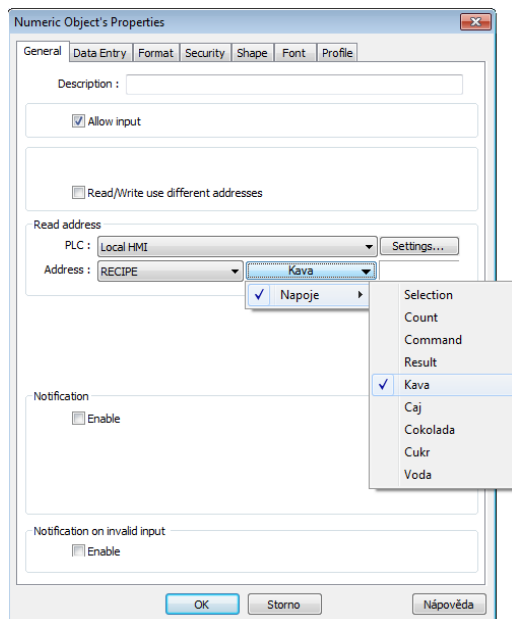
V nově otevřeném okně vybereme, jakou recepturu chceme zobrazit. Tedy Napoje. Dále můžeme vybrat, jak bude tabulka vypadat, tedy barvu pozadí, barvu vybraného řádku, způsob řazení dat v tabulce atd.



Potvrdíme a vložíme tabulku na obrazovku. Nyní si ukážeme jak tyto údaje vytvořit editovat nebo mazat. Nejprve si vložíme do projektu jeden ASCII objekt a pět Numeric objektů. ASCII objekt bude sloužit pro zobrazení názvů nápojů, nastavíme takto:



Jako adresu zvolíme RECIPE dále pak Napoje -> Polozka. Nyní se vytvoří ASCII objekt, do kterého se bude vkládat název nápoje. Takto přiřadíme ostatní položky do Numeric objektu.



Nyní máme vytvořeno 6 objektů, do kterých se budou zapisovat hodnoty z tabulky receptur. **Vždy se do těchto objektů zapíše vybraný řádek.** Nadepišeme si všechny tyto objekty, abychom věděli, který co reprezentuje a spustíme Off-line simulaci. Výsledná aplikace může vypadat následovně:

Polozka	Kava	Caj	Cokolada	Cukr	Voda
Turecka kava 50ml	20	0	0	4	20
Turecka kava 25ml	10	0	0	2	10
Kava 25ml	5	0	0	2	10
<b>Cokolada 50ml</b>	<b>0</b>	<b>0</b>	<b>10</b>	<b>4</b>	<b>20</b>
Cokolada 25ml	0	0	5	2	10

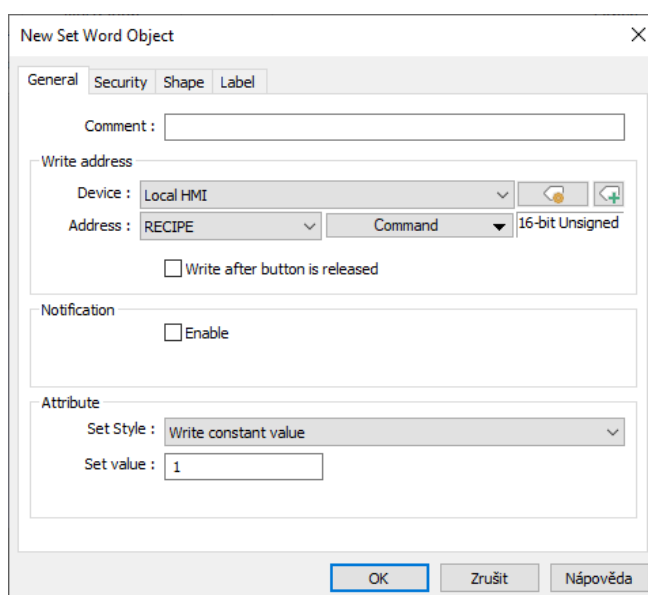
Polozka:

Káva	Čaj	Čokoláda	Cukr	Voda
0	0	10	4	20

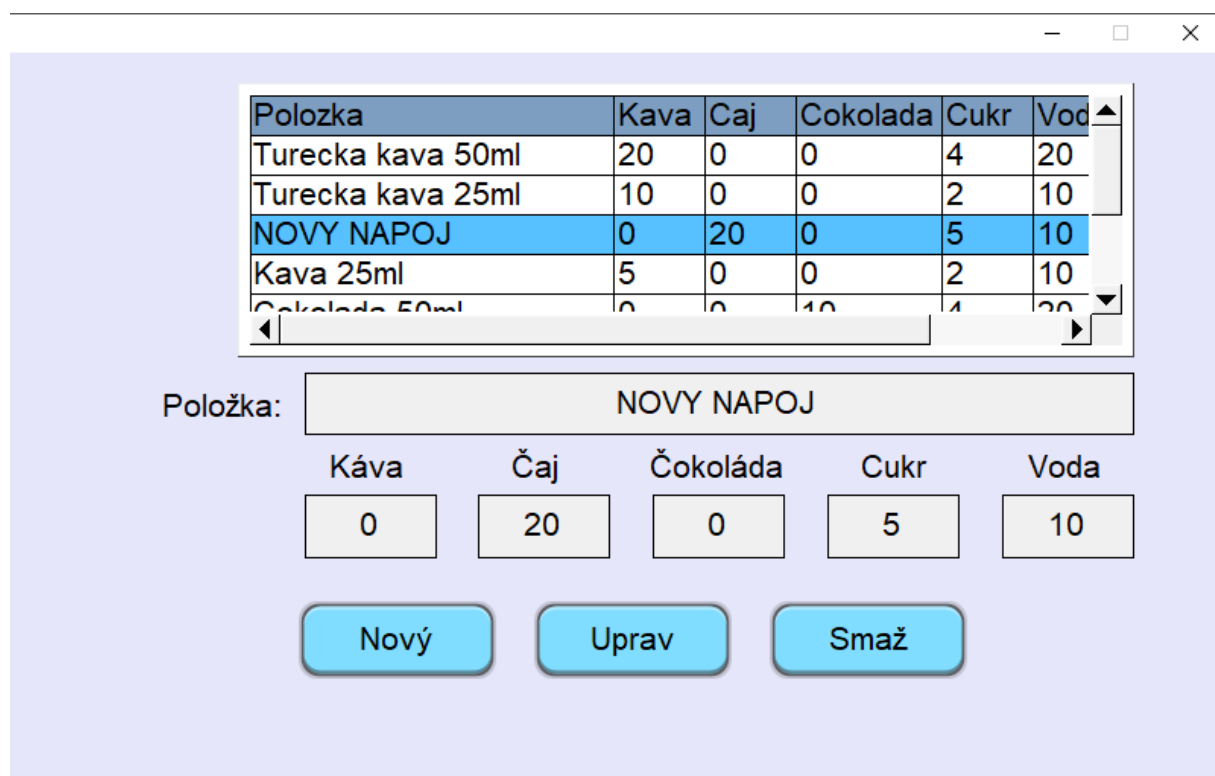
Nyní když máme objekty, které jsou svázané s danou recepturou, můžeme vytvořit funkci, která nám vytvoří nový záznam v tabulce. K tomuto účelu se používá numerická hodnota, která se zapíše do proměnné dané receptury a to do proměnné **Command**. Tato hodnota určí, co se s danými daty stane.

- 1 – Zapiš data do tabulky, jako nový záznam
- 2 – Uprav data ve vybraném řádku tabulky
- 3 – Smaže vybraný řádek v tabulce
- 4 – Smaže celou tabulku a veškerá data

Vytvoříme tedy nový objekt a to **Set Word**, který naleznete v menu Objects -> Set Word. Zde nastavíme adresu na RECIPE -> Napoje -> Command. Jako atribut zvolíme **Write constant value** a nastavíme zde 1. Tlačítko si nadepíšeme textem „Nový“. Toto opakujeme i pro tlačítko „Uprav“ a „Smaž“, jako hodnotu zde, ale zvolíme 2 respektive 3. Vše potvrdíme a zkompilujeme. Projekt spustíme v Off-line simulaci.



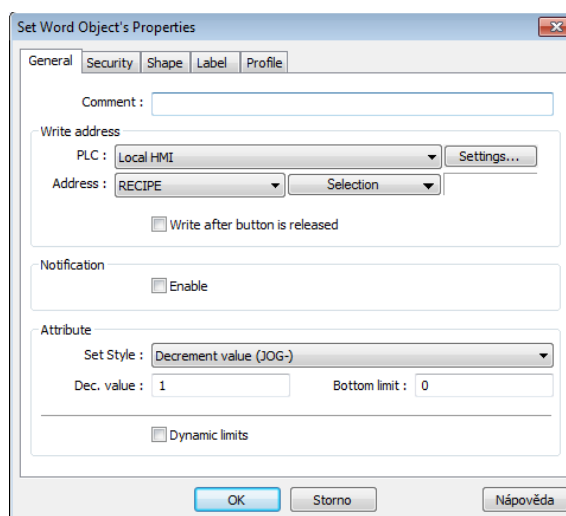
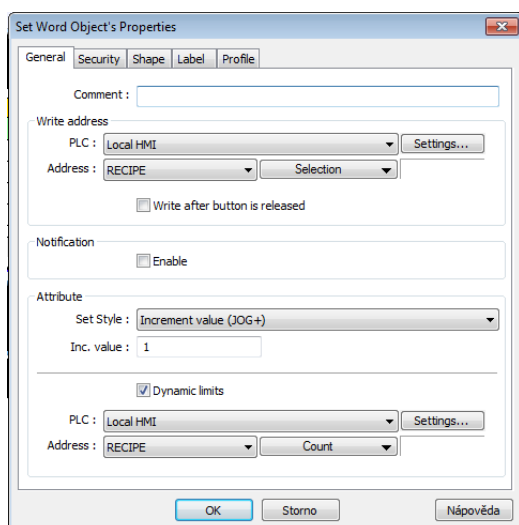
Výsledná aplikace může vypadat takto:



Někdy lze těžko prstem rolovat či dokonce vybrat určitý záznam. Proto si ukážeme, jak lze **listovat v tabulce** pomocí dvou tlačítek.

Tento způsob lze použít efektivně pouze v případě, že necháme záznamy v tabulce seřazeny tak, jak byly vytvořeny. Každý záznam má totiž své číslo a princip listování pomocí dvou tlačítek je založen na inkrementaci a dekrementaci této hodnoty. Pokud si tedy data v tabulce seřadíte podle jiného kritéria, nebude listování pomocí tlačítek fungovat efektivně.

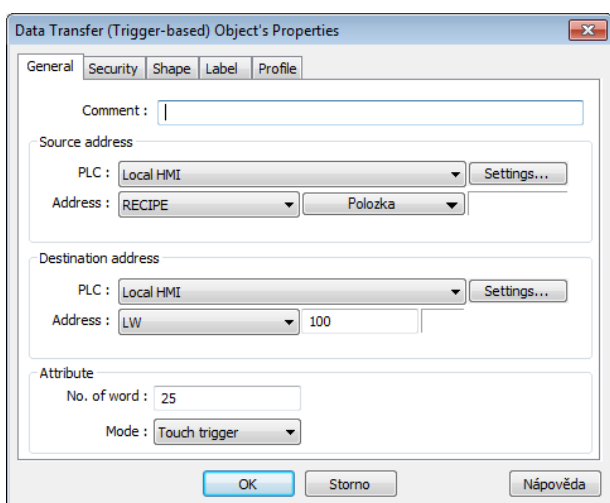
Vložíme do projektu dva nové objekty a opět **Set Word**. Tlačítka nastavíme takto:



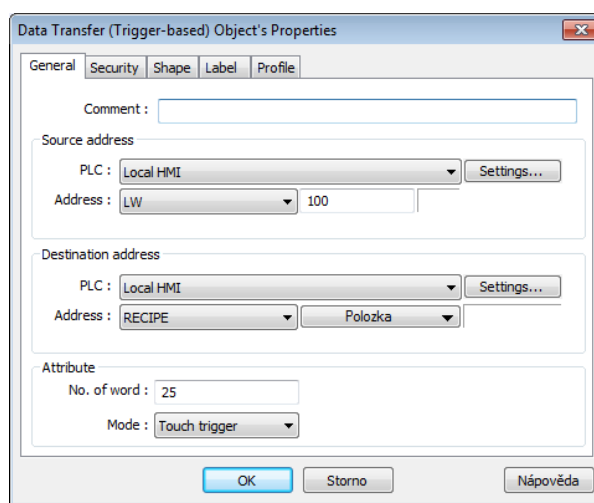
Okno vlevo je pro tlačítko dolu. Je zde použit inkrement o hodnotu 1 a jako horní hranice je zde použit počet receptur, tak abychom zbytečně neinkrementovali. U tlačítka nahoru je použita dekrementace o 1 s limitem 0.

Při práci s recepturami je důležitá funkce zápisu jednotlivých dat receptur do připojeného zařízení. Je tedy nezbytné, abychom byli schopni data z receptury přenést do zařízení podle daného výběru. K tomuto účelu zvolíme objekt **Data Transfer (Trigger-based)**. Tyto objekty budou dva. Jeden pro zápis do zařízení a druhý pro načtení ze zařízení. Zařízení nám bude simulovat paměťový prostor LW-100... . **Data Transfer (Trigger-based)** objekt nastavíme následovně:

Pro zápis do zařízení



Pro načtení ze zařízení



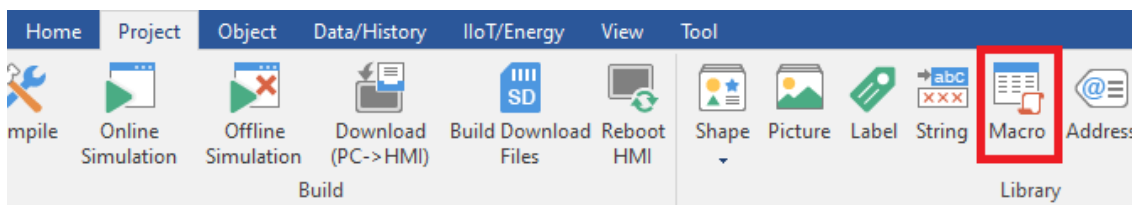
Následovně vložíme na obrazovku **Numeric** objekty a jeden **ASCII** objekt pro každou hodnotu receptury. Spustíme Off-line simulaci. Výsledný projekt může vypadat následovně:



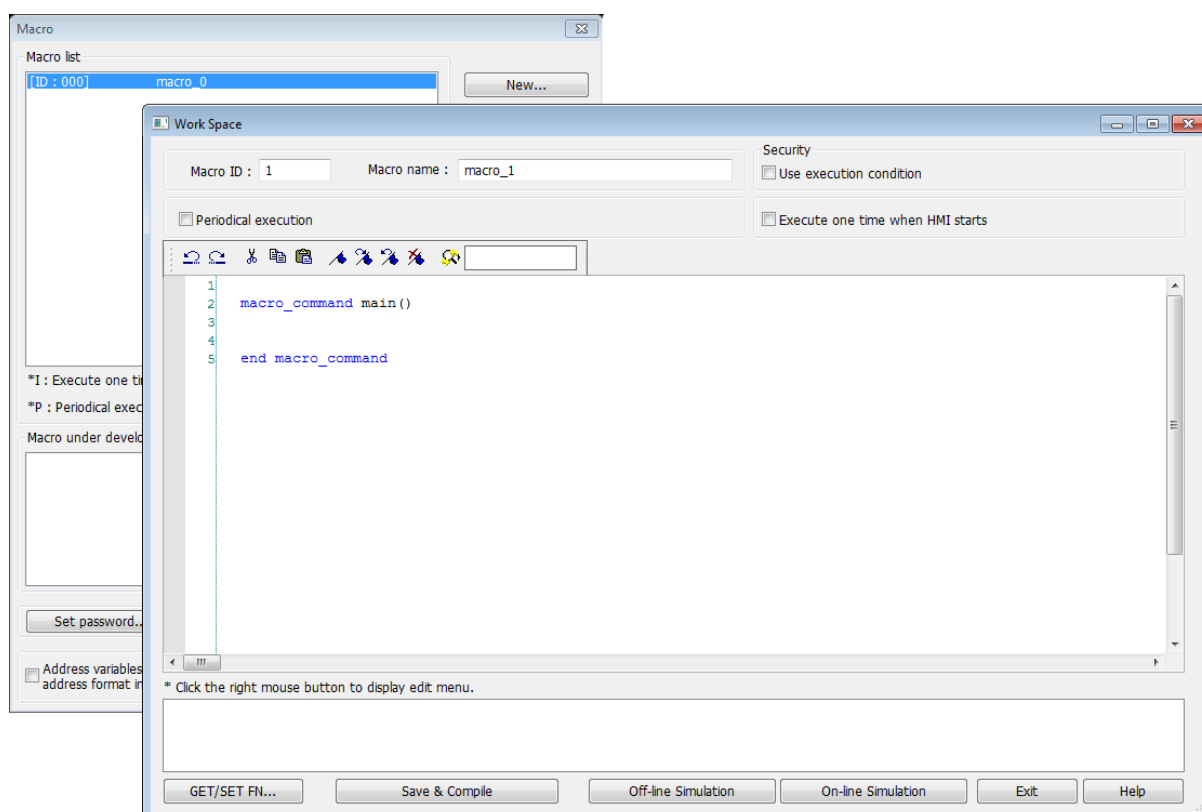
Aplikace ke stažení [zde](#)

## Makra

Pokud bychom potřebovali využít určité funkce, které nejsou předvytvořeny jako objekty, můžeme použít **Makro**. Makro je spustitelný kód, který se vykoná na základě určené podmínky. Lze jej spustit stiskem tlačítka, změnou hodnoty nebo podle časového intervalu. Samotné **Makro** nalezneme v menu Project -> Macro.

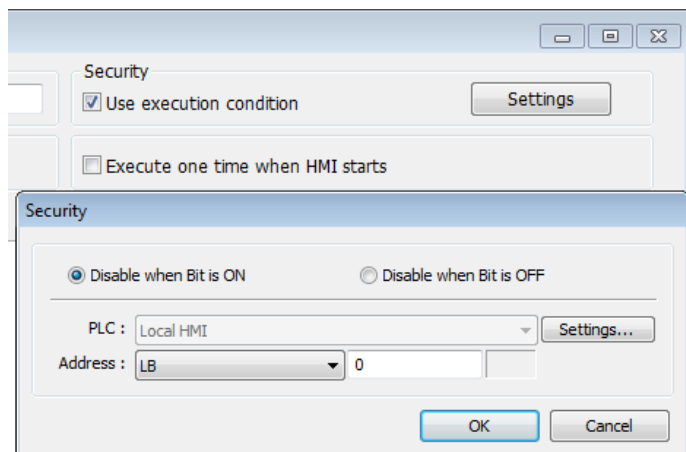


V nově otevřeném dialogovém okně zvolíme možnost New a otevře se nám následující okno:

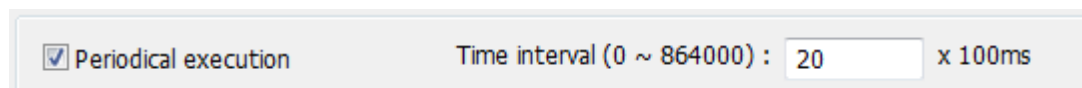


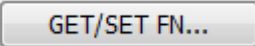
V horní části si makro můžeme pojmenovat a nastavit podmínky jeho spuštění. Pokud zaškrtneme **Use execution condition** a stiskneme **Settings**, tak můžeme nastavit podmínku pro spuštění, resp. nespouštění. Zde můžeme nastavit nadřazenou podmínku, která deaktivuje dané makro. Pokud budeme makro spouštět tlačítkem nebo časovým intervalem, nebude spuštěn, pokud bude tato podmínka splněna.

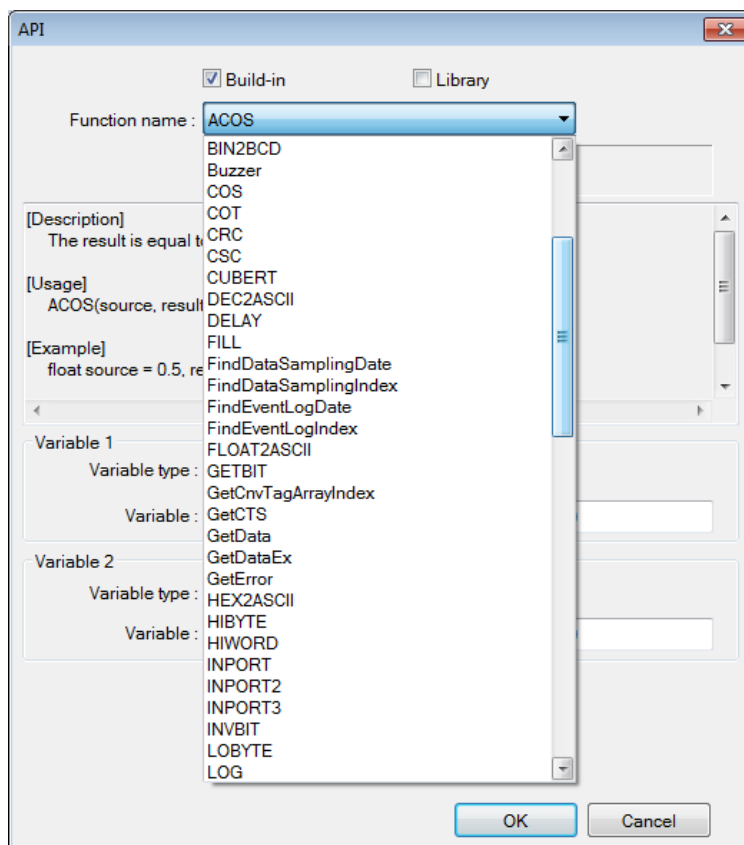




Dále lze nastavit časové spouštění makra. V horní části zaškrtneme možnost **Periodical execution**. Poté stačí nastavit časový interval pro spouštění makra.

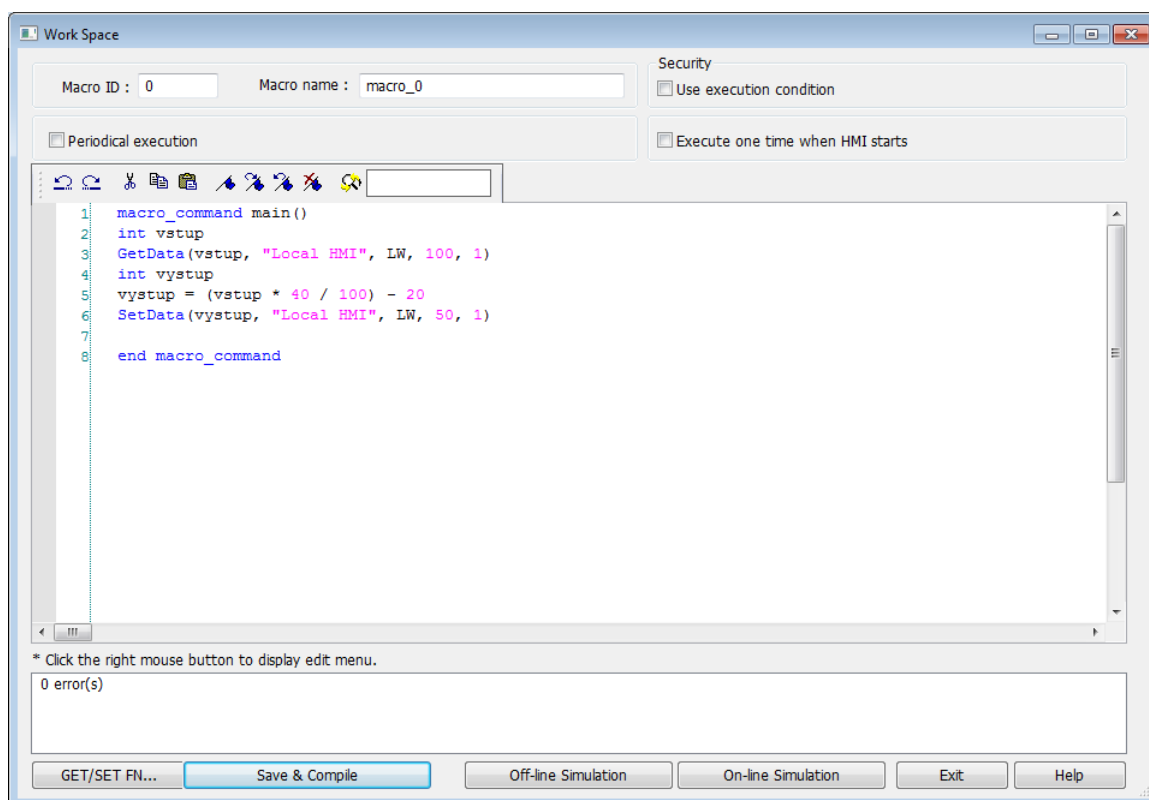


Samotný kód makra je nutné vložit mezi **macro\_command main()** a **end macro\_command**. Kód makra je tvořen před vytvořenými funkcemi. Tyto funkce můžete vložit pomocí tlačítka **GET/SET FN...** . V nově otevřeném dialogovém okně volíme, kterou funkci chceme použít. Popis všech funkcí, jejich použití a zápis, jsou popsány v manuálu EB Pro.



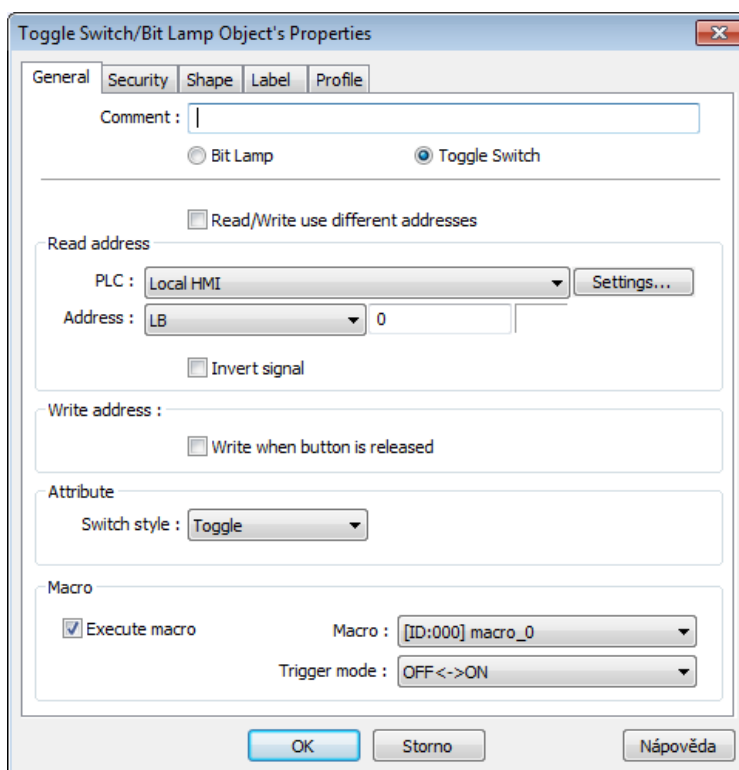
Založíme si nový projekt. Tento projekt bude načítat hodnotu z paměti LW-100. Tato hodnota bude v rozsahu 0-100. Řekněme, že bude simulovat vstup z teplotního čidla. Pomocí makra budeme tuto hodnotu přepočítávat na hodnotu v rozsahu -20° až 20°. Vložíme na obrazovku dva **Numeric** objekty, LW-100 a LW-50 do které budeme zapisovat výslednou hodnotu ve stupních. Pro hodnotu v LW-50 zvolíme datový typ 16-bit Signed. Otevřeme makro v menu Project -> Macro a zvolíme možnost New.

Abychom docílili požadovaných vlastností, zvolíme funkce **GetData()** a **SetData()**. Funkce **GetData()** načte hodnotu z LW-100 a uloží ji do proměnné vstup. Poté se vykoná matematická operace pro přepočet na stupně. Dále použijeme funkci **SetData()** pomocí, které zapíšeme proměnou vystup do paměti LW-50. Makro bude vypadat následovně.

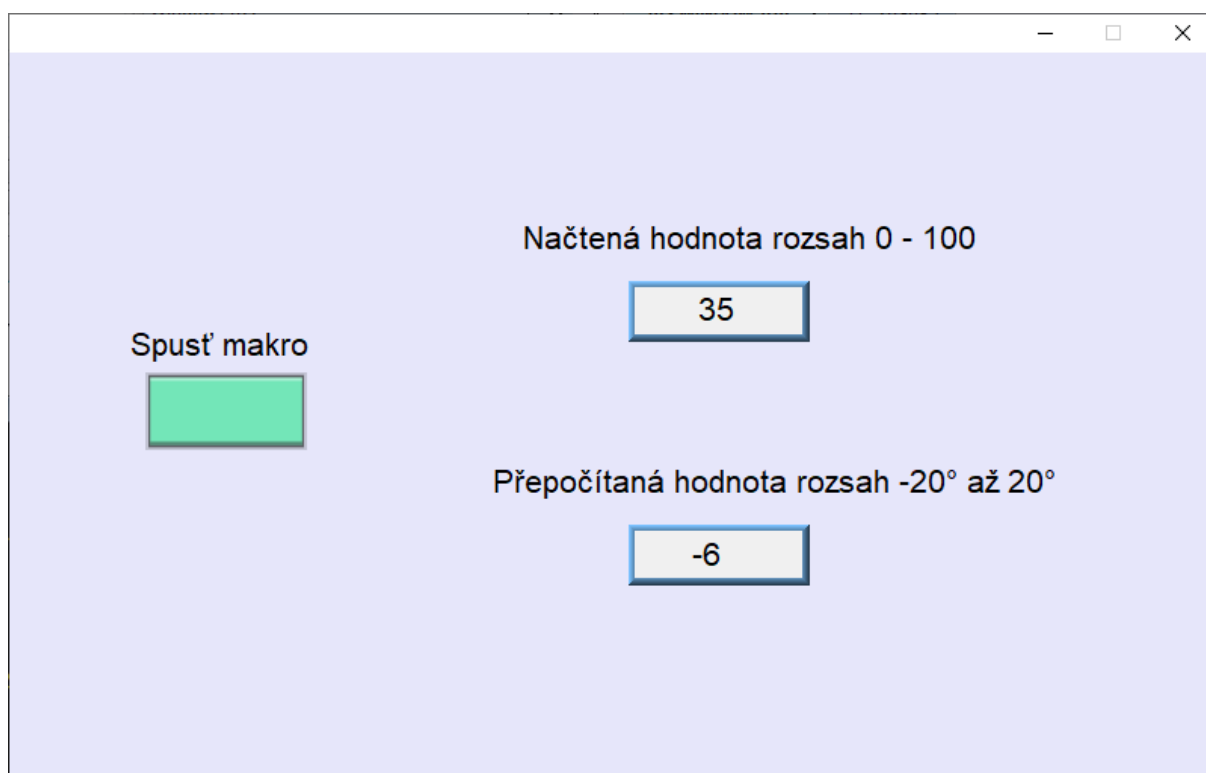


Zkompilujeme a uložíme makro pomocí tlačítka **Save & Compile**. Ve spodním okně se zobrazí, zda proběhla kompilace v pořádku bez chyb. Zavřeme okno.

Dané makro budeme spouštět pomocí bitového tlačítka. Přidáme na obrazovku **Toggle Switch**, který naleznete v menu Objects -> Toggle Switch. V nově otevřeném dialogovém okně, nastavíme vlastnosti daného tlačítka. Důležitá je funkce **Macro** umístěna ve spodní části. Zaškrtneme **Execute macro** a zvolíme makro z uložených. Dále pak mód pro spuštění makra. Vložíme tlačítko na obrazovku.



Projekt zkompilujeme a spustíme **Off-line** simulaci. Výsledný projekt může vypadat následovně:

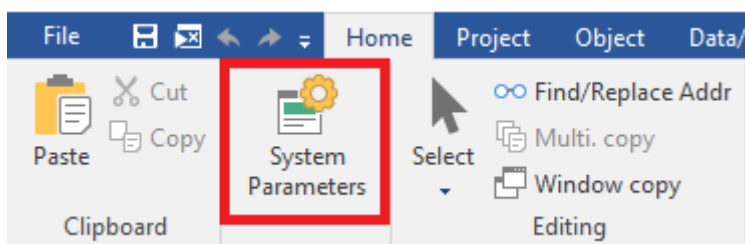


Aplikace ke stažení [zde](#)

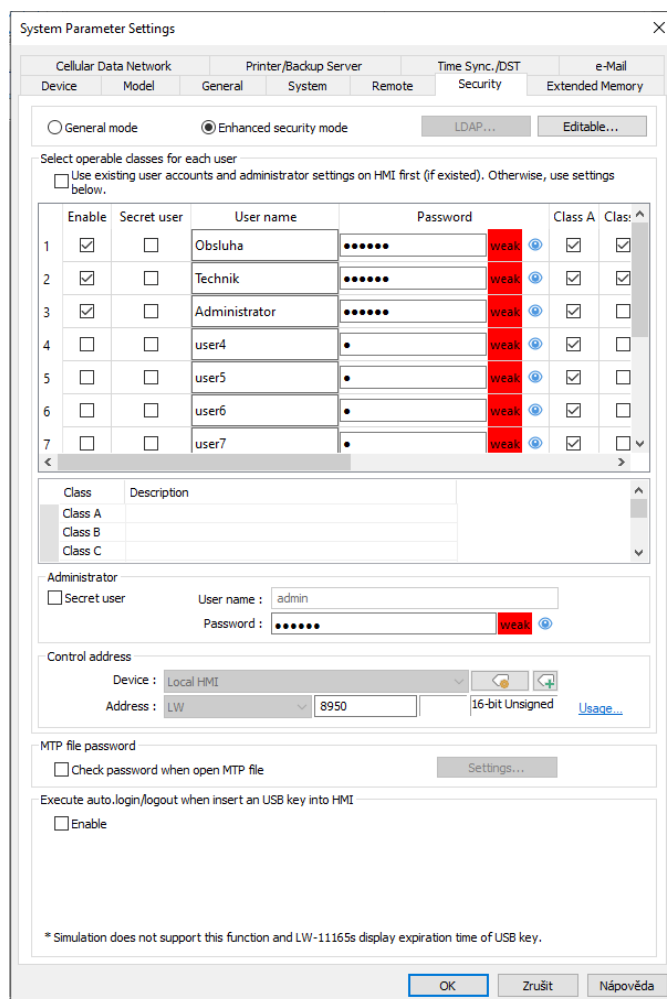
## Security

Mezi další vlastnosti, které lze využít na panelech Weintek, patří možnost využití uživatelů. Lze tak jednotlivým prvkům **nastavit úroveň zabezpečení**. Stejně je to i s uživateli. Lze tak rozdělit uživatele a objekty do různých tříd a udělat tak **různé úrovně přístupů**. Samotné přihlašování a správa uživatelů v projektu je již před vytvořena na obrazovkách 70,71,72,73,74.

**Správu uživatelů** naleznete v menu Home -> System parameters

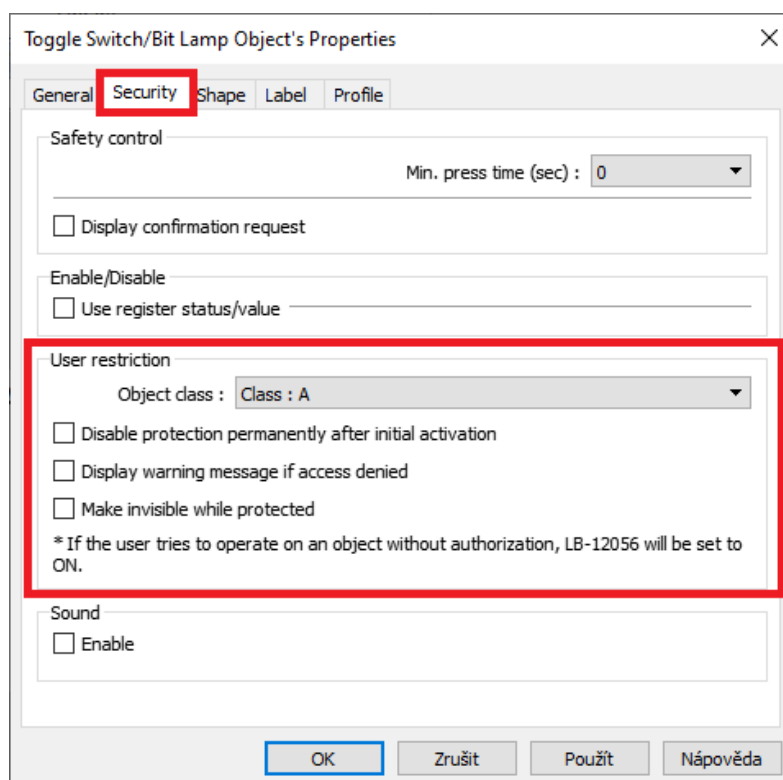


V nově otevřeném okně přejdeme na záložku Security. Zde si vytvoříme tři nové uživatele. Obsluha, Technik a Administrator. Nastavíme jim třídy přístupu. Obsluze pouze Class A, Technik bude mít nastaveno Class A a Class B. Administrator dostane Class A,B,C.

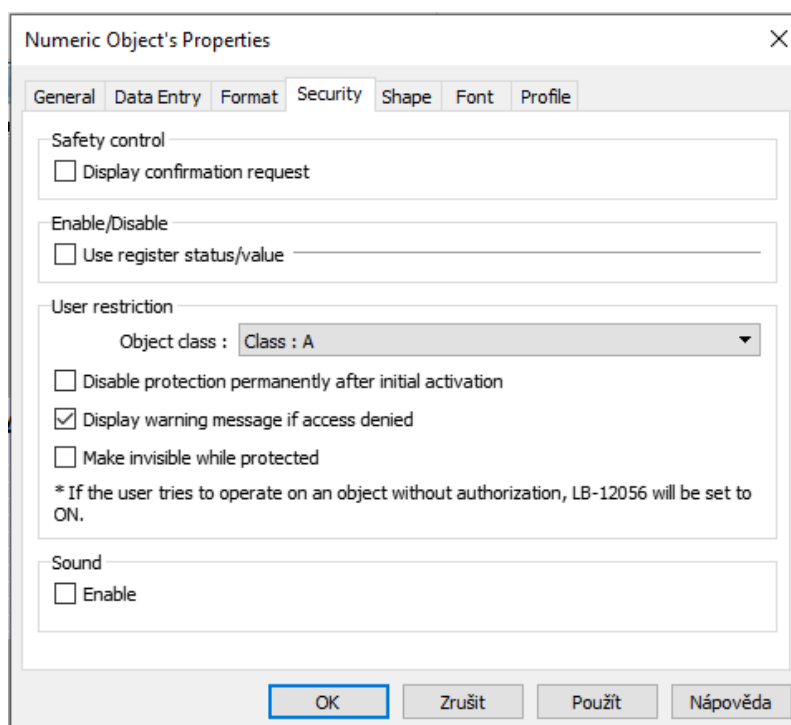


Obsluha bude mít nastavené heslo na „111111“, Technik bude mít heslo „222222“ a Administrator bude mít heslo „admin“

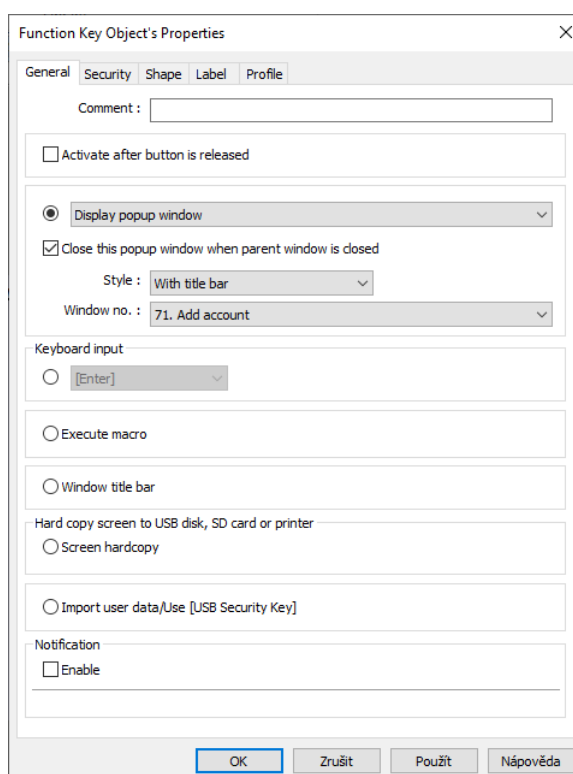
Nyní si vytvoříme několik objektů, kterým nastavíme určité třídy zabezpečení. Vytvoříme si tři bitové lampičky, které nastavíme jako toggle. Každou z nich však nastavíme na jinou úroveň ochrany. Přejdeme do záložky Security a nastavíme pro první Class A, pro druhou Class B atd...



Dále si vytvoříme tři Numeric objekty. Opět v záložce Security nastavíme pro každý jinou Class zabezpečení. Zde ale použijeme jednu z možností pod výběrem úrovně, a to **Display warning message if access denied**.

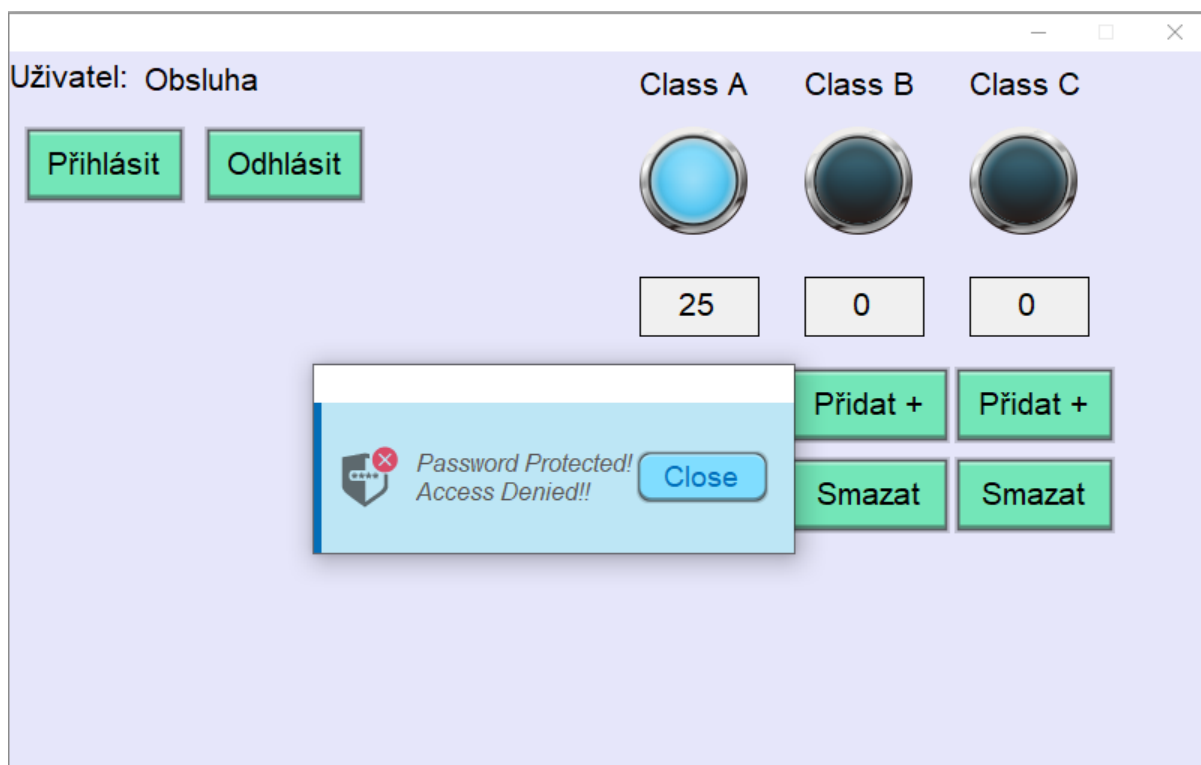


Dále vložíme několik Function Key objektů k vyvolání oken. Nejprve okno pro přidání uživatele. Nastavíme ho tak, aby zobrazil popup okno a poté ze seznamu vybereme okno 71. Add account.



Takto si vytvoříme několik tlačítek na vyvolání i oken 72 pro mazání uživatelů, 73 pro správu práv uživatelů a 74 pro správu hesel. Opět je můžeme rozdělit do různých úrovní přístupu.

Výsledný projekt může vypadat následovně:



## Ostatní aktivní prvky

Na předchozích příkladech jsme si ukázali způsob vytváření obrazovek, definování aktivních prvků, práci s daty, receptury, psaní maker apod. Všechny ostatní prvky se definují velmi obdobným způsobem.

Prvky jako sloupcový graf, ručkový ukazatel, více stavový přepínač, animace, zadávání textů atd. mají velmi intuitivní rozhraní a jejich fungování je z dialogových oken zřejmé.

Pokud jste si prošli všechny uvedené příklady, nebude pro vás jistě problém vytvořit plně funkční aplikaci pro libovolnou řízenou technologii. Narazíte-li přesto na nějakou nejasnost, nahlédněte **do originálního manuálu k software EasyBuilder Pro**.

**Ukázkové projekty**, tak jak jsme je vytvářeli v této příručce, nevyžadují připojení programovatelného automatu a **můžete jejich funkčnost na panelu vyzkoušet** stejně tak, jako jsme je testovali v off-line simulaci na PC. Všechny projekty naleznete [zde](#)