

SoftPLC IDE

UŽIVATELSKÁ
PŘÍRUČKA

RC•Ware[®]
SoftPLC

Obsah

Obsah.....	2
1 Úvod	3
1.1 Předpoklady a cíle.....	3
1.2 Základy systému	3
2 IDE.....	5
2.1 Požadavky na systém	5
2.2 Instalace.....	5
2.3 Okna a panely IDE	6
2.4 Vytváření projektu	8
2.5 Spouštění programu.....	19
3 Licencování.....	20
3.1 Základy.....	20
3.2 Hardwarový klíč.....	21
3.3 Softwarové licencování	22

1 Úvod

1.1 Předpoklady a cíle

Předpoklady:

Existuje technický popis a výkresy projektu (nebo alespoň tabulka vstupů a výstupů s přiřazením periférií vstupům a výstupům I/O modulů). Je definovaná topologie systému včetně řídicí úrovně (RcWare Vision).

Pro funkční testy jsou v rozvaděčích namontovány I/O moduly, připojeny k perifériím a I/O busu, adresovány a oživeny.

Cíle:

Pro všechny procesní stanice je navržen, napsán, parametrován, kompilován a nahrán aplikační software. Program je otestován s I/O moduly včetně výměny dat mezi podstanicemi. Pro snadný import do RcWare Vision jsou exportovány seznamy datových bodů. U dotykových displejů jsou proměnné připraveny pro import do HMI editoru.

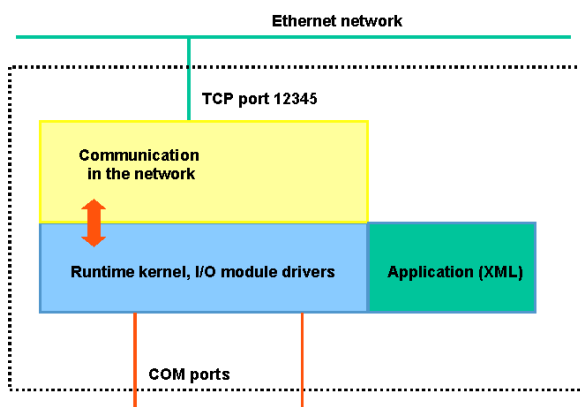
1.2 Základy systému

Aplikace SoftPLC se skládá ze dvou hlavních komponent:

- runtime
- aplikační program.

Runtime je program instalovaný v podstanici. Zpracovává aplikaci, která je do podstanice nahrána. Cesta k aplikaci se nastavuje v parametrech runtime – viz Konfigurace runtime.

Procesní stanice s runtime

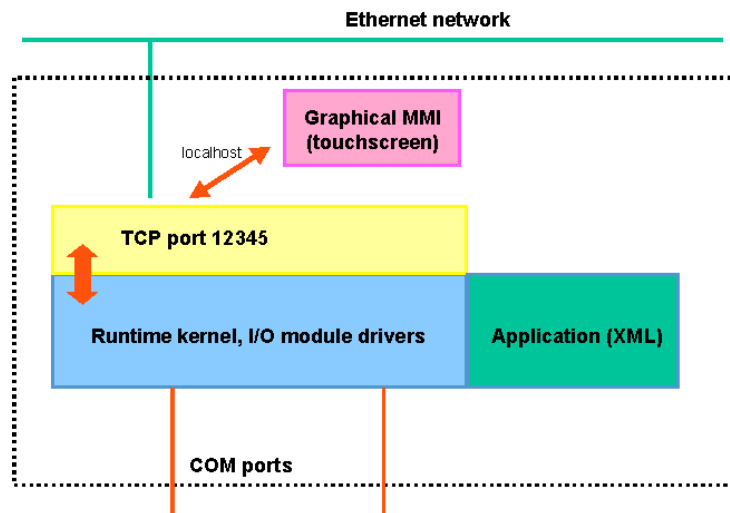


Na COM porty jsou připojeny I/O moduly. Po startu runtime se kontroluje licence (viz Licencování), pak se načítají aplikační soubory (dále zvané *projekt* nebo *aplikace*), kontroluje se jejich integrita a začnou se vykonávat funkce programu.

Pro sdílení dat je v runtime komunikační vrstva. Přijímá připojení na TCP portu 12345 (tuto hodnotu lze změnit v konfiguraci). K tomuto portu se připojuje např. aplikace touchscreen nebo OPC server pro integraci do RcWare Vision. Je to proto, že spojení TCP používá specifický protokol, je stabilní a nezávislé na platformě.

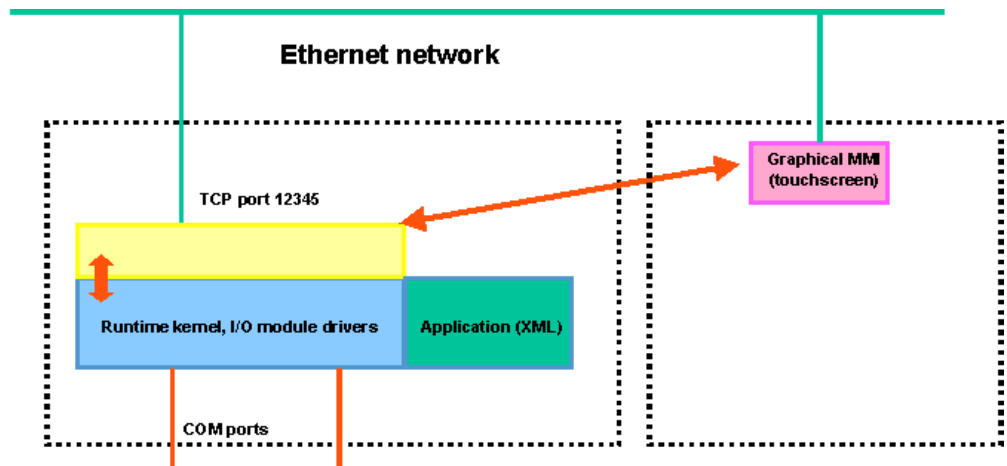
Touchscreen se připojuje přes TCP port 12345:

Procesní stanice s runtimem a touchscreenem (např. IPCT.1)

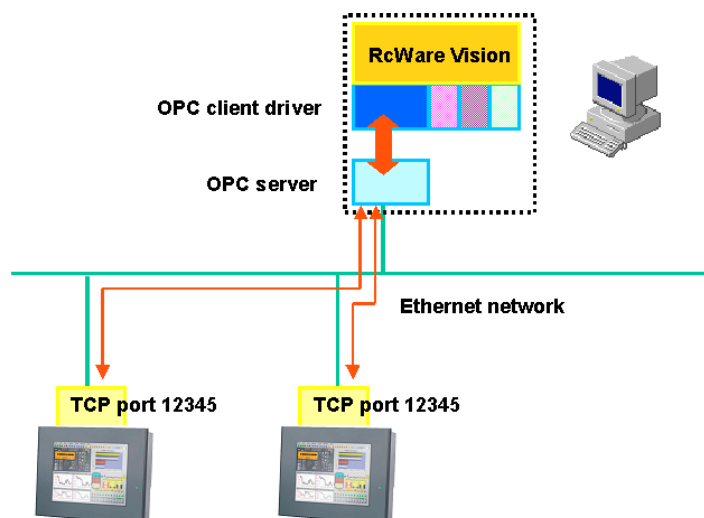


Díky této síťové topologii je také možné spouštět aplikaci Touchscreen na jiném počítači, takže procesní stanice je u technologie a ovládací rozhraní může být umístěno např. u domovního technika.

Procesní stanice s runtimem, touchscreen na jiném stroji



Podobně se připojí více podstanic na jeden OPC server:



Podrobnosti najdete v části Integrace do RcWare Vision.

2 IDE

IDE (Integrated Development Environment) je nástroj pro tvorbu, parametrování, oživování a testování aplikací SoftPLC.

2.1 Požadavky na systém

IDE vyžaduje PC s těmito minimálními parametry:

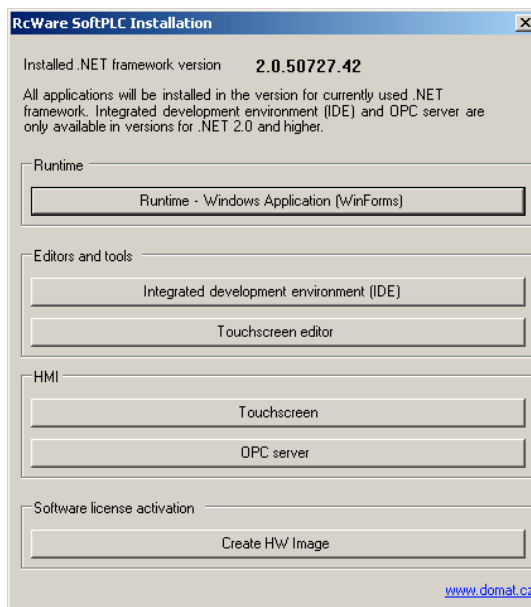
- Pentium 1 GHz
- 256 MB RAM již funguje, 512 MB pro běžnou práci, 1 GB pro největší aplikace
- 30MB místa na disku (plus projekty)
- Ethernet
- sériový port nebo adaptér USB na COM pro testování I/O modulů
- XGA display (1024 x 768 a více)
- Windows 2000 SP2 nebo Windows XP
- Microsoft.NET V2.0 (ke stažení na <http://msdn.microsoft.com/netframework> zdarma, hledejte „.NET Framework Version 2.0 Redistributable Package“).

2.2 Instalace

Kompletní balík instalací se distribuuje na CD nebo ke stažení jako soubor ZIP. Po rozbalení se vytvoří stromová struktura, která obsahuje

- instalátory (.msi) pro různé komponenty (runtime, IDE, touchscreen, OPC server...)
- licencovací program (ESG.SoftPLC.Licenses.HWInfo.exe)
- malý program pro snadné spouštění instalátorů (SoftPLC.InstallationHub.exe).

1. Spustíte **SoftPLC.InstallationHub.exe**.



2. V horní části okna se zobrazuje instalovaná verze .NET Frameworku. Zkontrolujte, zda je 2.0 nebo vyšší.
3. Instalujte „**Integrated development Environment (IDE)**“. Během instalace ponechte výchozí volby.
4. Pokud budete používat **Touchscreen editor**, můžete ho instalovat nyní. Během instalace opět ponechte výchozí volby.
5. IDE je nyní instalováno. Poběží jako **demo** verze, jediné omezení je to, že komunikace s moduly se *po 10 minutách zastaví*. Všechny ostatní funkce jsou

Demo verze, plná verze

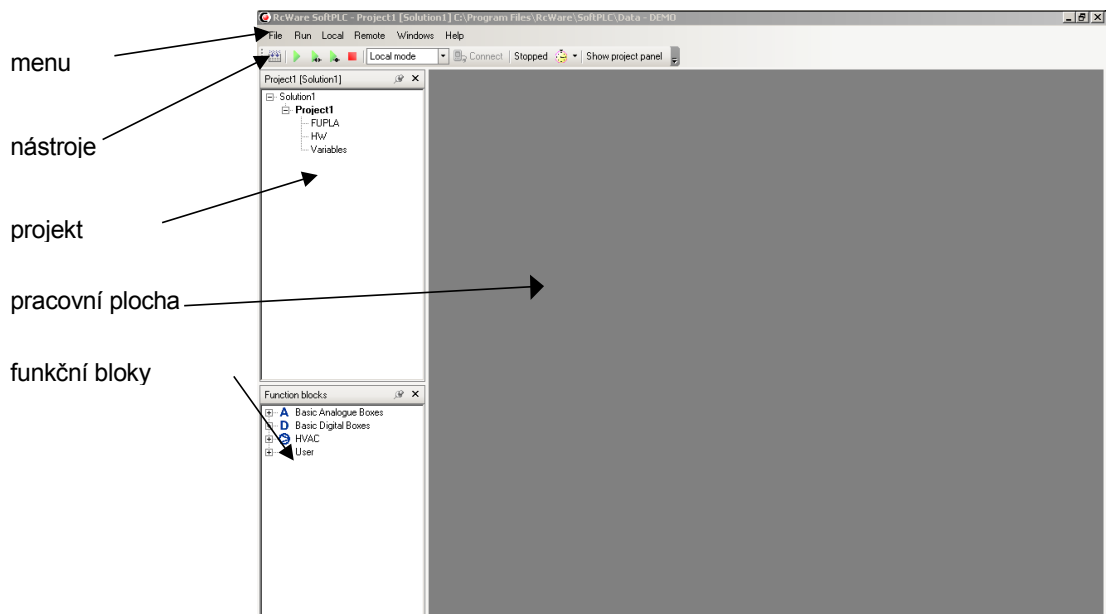
stejně, jako v **plné** verzi. Pokud chcete používat plnou verzi, licencujte ji (viz Licencování).

6. Ukončete Installation Hub.

2.3 Okna a panely IDE

2.3.1 Okno programu

Spusťte **Start – Programs – RcWare – SoftPLC – SoftPLC**. Úvodní obrazovka zobrazí logo SoftPLC a postup nahrávání komponentů. Po startu se otevře úvodní okno:



Skládá se z těchto částí:

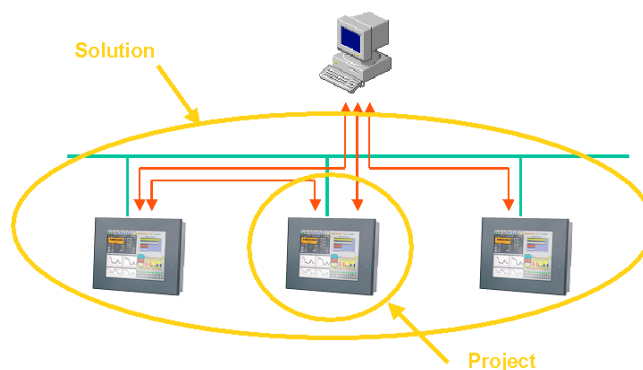
- lišta s menu
- lišta s nástroji
- panel projektu
- panel s funkčními bloky
- pracovní plocha
- panel výstupu compileru (nyní neviditelný).

2.3.2 Panel projektu

Projekt, solution

V panelu projektu se vybírají projekty a sítě (solution). Co je **project** a **solution**?

Project je vlastně aplikace pro jednu procesní stanici nebo regulátor. **Solution**



obsahuje jeden nebo více projektů, které jsou spojeny sítí a mohou sdílet data nebo být připojeny k jedné (nebo několika) pracovní stanici (vizualizaci). Každá síť obsahuje alespoň jeden projekt. Síť musí být definována i v tom případě, že máme jedinou autonomní podstanici (tato síť obsahuje tuto jedinou podstanici nebo regulátor).

Každý projekt obsahuje tři položky:

FUPLA

Zkratka pro FUnction PLAn. Grafická reprezentace funkce programu, skládající se z proměnných a funkčních bloků, které tyto proměnné čtou, zpracovávají a zapisují výsledky do jiných proměnných.

HW

Konfigurace hardwaru procesní stanice a I/O modulů, především osazení sériových linek a síťových kanálů, komunikujících s I/O moduly, jejich adresování, konfigurace, typy a adresy I/O modulů, jejich nastavení a další parametry.

Pro moduly s komunikací Modbus a Advantech má konfigurátor funkci autodetekce – na sběrnici najde připojené moduly a může je automaticky založit ve stromu modulů na sběrnici.

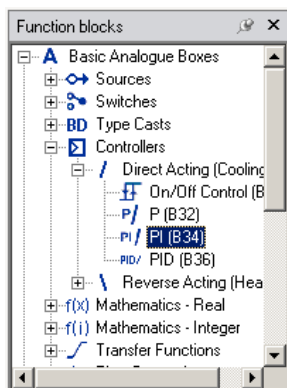
Zde se též definuje vzájemná komunikace mezi podstanicemi.

Variables

Seznam všech proměnných, které aplikace SoftPLC obsahuje. Zde se prohlížejí a nastavují hodnoty proměnných. Pro rychlé vyhledání jsou zde funkce třídění a filtrování.

Po výběru dvojklikem se záložky HW a Variables otevřou v pracovním panelu.

2.3.3 Panel funkčních bloků



Zde je knihovna všech dostupných funkčních bloků SoftPLC, které lze přetahovat (drag & drop) do schématu (tedy do listu s funkčními bloky, proměnnými a vazbami) a v němž je definována funkce programu.

Bloky jsou rozříděny do stromu podle jejich funkce. Hlavní skupiny jsou tyto:

- Basic analogue blocks – základní analogové bloky
- Basic digital blocks – základní digitální bloky
- HVAC blocks – bloky funkcí VVK (větrání, vytápění a klimatizace).

Popisy všech bloků a jejich funkcí najdete v nápovědě **Function block documentation** (soubor .pdf, který se automaticky instaluje zároveň s IDE). Odkaz na dokumentaci je v menu Start, ve skupině SoftPLC, a vytváří se automaticky během instalace.

2.3.4 Pracovní panel

V tomto panelu se otevírají schémata, tabulka s proměnnými a konfigurátor hardwaru.

Je možné mít otevřených více záložek najednou a přepínat mezi nimi kliknutím na název záložky (schématu).

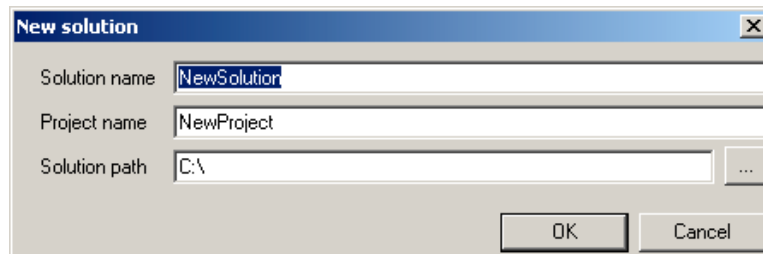


Hvězdička u názvu záložky znamená, že v záložce jsou neuložené změny. Aktivní záložku můžeme uzavřít kliknutím na „X“ v pravém horním rohu okna.

2.4 Vytváření projektu

2.4.1 Nová síť

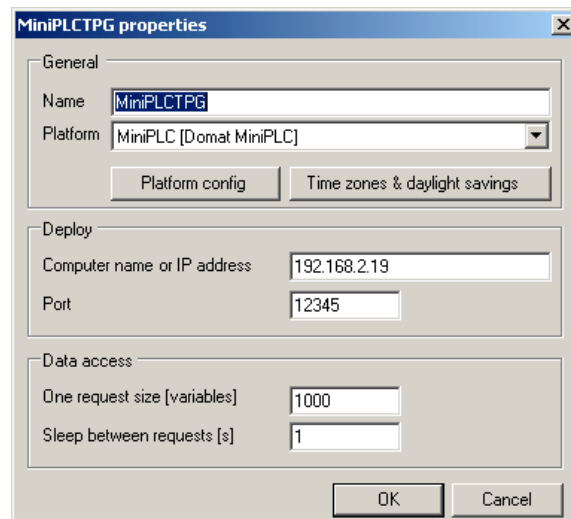
1. Vyberte **File, New, Solution**.
2. Zadejte název sítě (typicky jméno akce, např. **HotelLhotka**) a název projektu



(typicky rozvaděč, v němž je stanice instalována, nebo funkce podstanice (např. **BA, Topení, BudovaC**)) Cesta v definici sítě zadává adresář, v němž je projekt uložen.

Všechny údaje lze později měnit.

3. Klikněte pravým tlačítkem na název projektu a zvolte **Properties**.

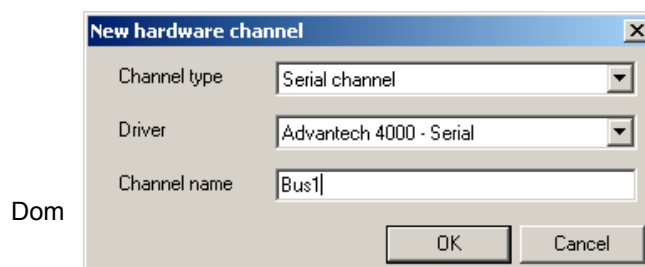


Pokud je již známa IP adresa podstanice, je možné ji zadat v rámečku *Deploy*. Dialog *Time zones & daylight savings* použijte pouze pokud bude podstanice instalována v jiné oblasti, než je Evropa (UTC + 1h, posun letního času o 1 h). Pro ostatní oblasti se průběh přechodů léto – zima a zpět zadává manuálně.

Rámeček *Data access* neměňte. Hodnoty se upravují pouze při zvláštních požadavcích na komunikaci.

2.4.2 Komunikační kanály

1. Dvakrát klikněte na větev **HW**. V panelu se otevře prázdná záložka.
2. Klikněte pravým tlačítkem na prázdný panel a zvolte **Create channel**.



Pro sériovou komunikaci s I/O moduly Domat vyberte *Serial channel* a *Advantech 4000 – Serial driver* nebo *Modbus RTU*, podle toho, jaké I/O moduly používáte. Přejmenujte sběrnici na vhodné krátké jméno (název bude součástí názvu I/O proměnných).

Typy komunikačních kanálů a drivery

Typy komunikačních kanálů jsou:

OPC channel: SoftPLC funguje jako OPC client pro jakýkoli OPC server běžící na stejném počítači.

Serial channel: sériová linka RS232/RS485, COM port. V současnosti jsou pro ni dostupné tyto drivery:

Advantech 4000 serial – domat I/O moduly s protokolem Advantech

Modbus RTU TCP/serial – domat I/O moduly s protokolem Modbus nebo libovolné I/O moduly Modbus nebo přístroje fungující jako Modbus servery

EMU Electricity meters – měřiče elektřiny (<http://www.emuag.ch/>)

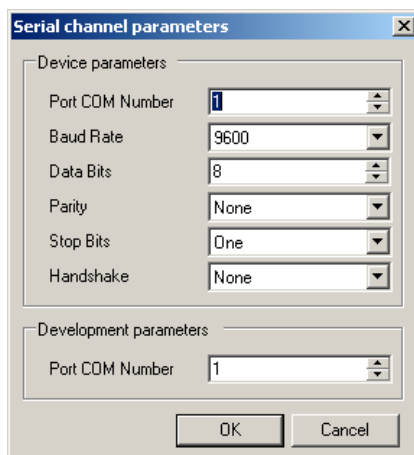
RSZ serial – podstanice Sauter fungující jako I/O moduly

SoftPLC native channel: používá se pro výměnu dat mezi podstanicemi

TCP channel: pro Modbus over TCP

UDP channel: rezervováno.

3. Klikněte pravým tlačítkem na kanál a zvolte **Edit, Channel properties**.



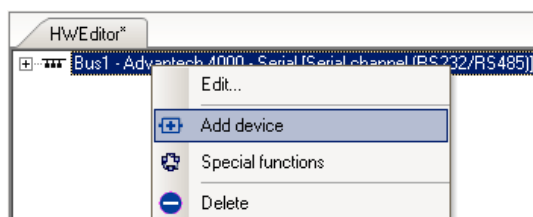
Nastavte parametry COM portu podle potřeby. Pozor, číslo COM portu se může lišit pro notebook (např. COM1 pro přímé připojení na I/O sběrnici) a pro podstanici (např. COM2). Je tedy možné nastavit “vývojový” COM port zvlášť. To se používá pro uvádění do provozu, zatímco COM port nastavený v sekci *Device parameters* platí pro podstanici (runtime).

Výchozí nastavení jsou vhodná pro I/O moduly domat Mxxx s převodníkem M010.

Development parameters

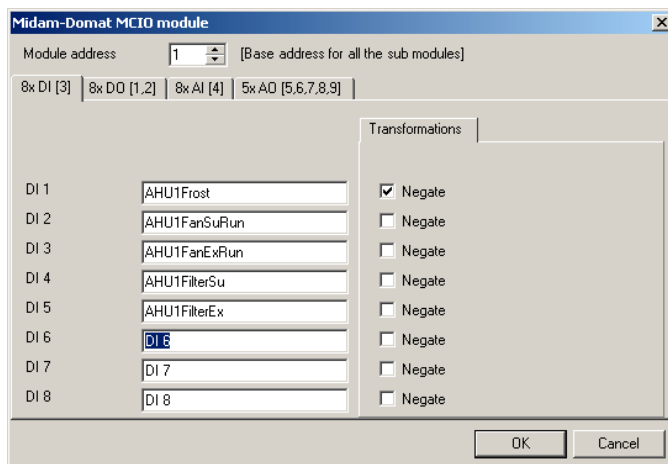
2.4.3 Přidávání modulů

1. Nyní přidáme na sběrnici I/O moduly. Typy a adresy by měly být uvedeny v zapojovacích schématech projektu, pokud nejsou, adresaci určuje autor softwaru. Pravým tlačítkem klikněte na kanál a vyberte **Add device**.



2. Vyberte ze seznamu první modul na sběrnici a klikněte na **OK**. Modul se objeví v seznamu a dvojklikem na jeho ikonku se otevře okno s nastavením modulu.

- Nastavete adresu modulu (dekadicky). Zadejte názvy vstupních a výstupních proměnných (případně přepočty signálů).



- Pokračujte se všemi moduly na sběrnici. Zkontrolujte, že žádné dvě adresy nejsou stejné.

2.4.4 Variables viewer

Moduly jsou seřazeny ve stromovém náhledu. V panelu Project, **Variables** nyní můžete vidět seznam proměnných, které se týkají vstupů a výstupů.

OPC	Name pr...	Name	Value	Unit	Type	UpdateTime	Quali...	Blocks	Origin
<input type="checkbox"/>	<input type="checkbox"/>	HW	Bus1.AHU1FanExRun	false	Boolean	01.01 0001 01:00:00...	Bad	None	HW
<input type="checkbox"/>	<input type="checkbox"/>	HWBlock	Bus1.AHU1FanExRun_blk...	0	Int64	25.07 2006 11:00:29...	Good	N/A	HWBlockControl
<input type="checkbox"/>	<input type="checkbox"/>	HW	Bus1.AHU1FanSuRun	false	Boolean	01.01 0001 01:00:00...	Bad	None	HW
<input type="checkbox"/>	<input type="checkbox"/>	HWBlock	Bus1.AHU1FanSuRun_blk...	0	Int64	25.07 2006 11:00:29...	Good	N/A	HWBlockControl
<input type="checkbox"/>	<input type="checkbox"/>	HW	Bus1.AHU1FilterEx	false	Boolean	01.01 0001 01:00:00...	Bad	None	HW

První sloupec se používá pro značení a odznačování proměnných.

OPC Vybrané proměnné jsou viditelné v RcWare OPC serveru (typicky se označují ty, které jsou pak použity v RcWare Vision).

Name prefix je část názvu proměnné a používá se pro označení původu proměnné (HW, HWBlock, SW (systém)) nebo jejího umístění (název listu FUPLA).

Name: Název proměnné, na který se pak odvoláváme při vkládání do žebříčku v grafice FUPLA. Tento název je rovněž částí názvu OPC proměnné.

Value: Aktuální hodnota proměnné, dynamicky se mění.

Unit: Fyzikální jednotka (°C, %, %rH, kPa atd.)

U vstupů a výstupů (proměnné HW) se jednotky nastavují kliknutím na příslušný řádek a stisknutím **Ctrl+E**. Otevře se dialog s nastavením parametrů proměnné.

Type: Typ proměnné. Používají se tyto typy:

Boolean: Binární proměnná, nabývá hodnot False nebo True.

Double: Reálná čísla s dvojitou přesností: 64 bit floating point.

Int64: 64 bit integer se znaménkem.

String: Rezervováno.

Date Time: Pro operace s datem a časem.

Nastavení jednotek u HW proměnných

Podrobný popis typů naleznete v nápovědě k funkcím.

Update time: Datum a čas poslední aktualizace proměnné. Zde je možné kontrolovat komunikaci a zjistit, zda jsou komunikační linky optimálně nastavené pro danou konfiguraci I/O.

Quality: Bad nebo Good, podle stavu aktualizace.

Blocks: Každá hardwarová proměnná může být blokována proti čtení, zápisu nebo obojímu. Blokování se nastaví v blokovací proměnné, která přísluší každému vstupu nebo výstupu a má název **<jméno_proměnné>_blk_mode**. Tyto blokovací proměnné mohou mít následující hodnoty:

- 0 Bez blokování
- 1 Blok čtení (tj. proměnná se nenačítá)
- 2 Blok zápisu (tj. do proměnné se nezapisuje)
- 3 Blok čtení i zápisu

(Platí jen nejnižší dva bity, tedy např. 8 = Bez blokování, 7 = Blok čtení i zápisu atd.)

Proč používat blokování?

Blokování se používá hlavně v případech, že v softwaru máme definovanou již celou I/O sběrnici, ale nainstalovaná a oživená je jen část modulů. Normálně by se driver snažil komunikovat i s nepřipojenými moduly a pravidelně je obesílal, což by znamenalo pomalou komunikaci a výpadky. Proto se chybějící moduly mohou ručně zablokovat proti komunikaci.

Celý modul najednou (nebo spíše všechny HW proměnné v něm) můžeme zablokovat na ikoně modulu v záložce **HW** výběrem *Reads and writes block*. Všechny proměnné v modulu jsou pak blokovány (viz **Variables viewer**.)

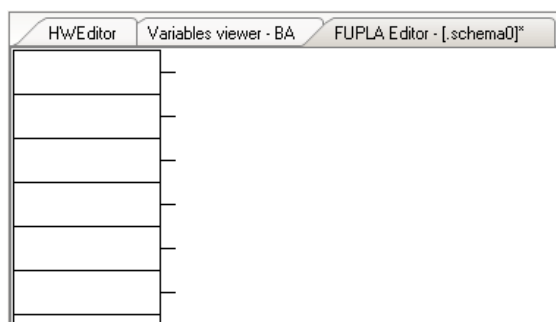
Modul je možné opět odblokovat pravým kliknutím na ikonu modulu v záložce **HW** a výběrem *Cancel all blocks*.

Origin: Komentář o původu proměnné. I po případném přejmenování je dále zřejmé, o jaký typ proměnné se jedná.



2.4.5 Schémata

Ve stromě Project, pod FUPLA, je pravým kliknutím přidáme nové schéma – výchozí název je *schema0*. Schéma je prázdné. Dvojklikem na *schema0* se v pracovním panelu otevře schéma.



Žebříčky na levé a pravé straně slouží ke vkládání proměnných. Tak jsou hardwarové proměnné připojeny k funkčním blokům.

Přidání schématu

Projekt může obsahovat více schémat. Nové schéma přidáme pravým kliknutím na FUPLA a výběrem **Add schema**. Schémata je možné přejmenovat, takže jejich názvy popisují funkci schématu. Názvy schémat jsou součástí názvů proměnných ve stromu OPC.

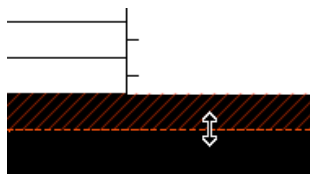
Schémata též lze skládat do stromové struktury (pravý klik na schéma, nikoli na FUPLA, a **Add schema**). Toto řazení slouží jen pro zpřehlednění stromu, na funkci aplikace nemá žádný vliv.

Každé schéma může obsahovat libovolný počet funkčních bloků.

2.4.6 Grafické funkce

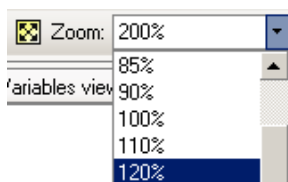
Schéma může měnit svou velikost potažením za okraje:

Změna velikosti schématu



Pokud okraje nejsou viditelné, použijte zoomovací nástroje na liště. Odzoomujte tak, že schéma je menší než aktivní plocha panelu a schéma je obklopeno černými okraji. Pak uchopte schéma za okraj a potažením změňte jeho velikost. Pro zoom funguje také Ctrl+kolečko myši.

Zoom

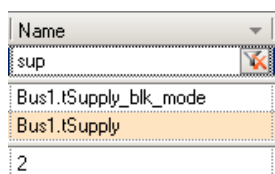


Při natahování svislým směrem se v žebříku objevují nové pozice, do nichž lze vložit další proměnné.

Fit to screen 

Schéma lze kdykoli přizpůsobit velikosti obrazovky kliknutím na ikonu „Fit to screen“.

2.4.7 Vložení proměnné do schématu



Aby se dala proměnná přiřadit funkčnímu bloku, tj. například přivést signál z čidla na vstup PID regulátoru, musíme vložit hardwarovou proměnnou do žebříčku s proměnnými.

1. Pravým tlačítkem klikněte na pozici v žebříčku.
2. Vyberte **Place variable**.
3. Vyberte proměnnou ze seznamu. Používejte třídění a filtrování: v boxu Name napište několik písmen, které obsahuje název proměnné, a stiskněte <Enter>.
4. Seznam se zkrátí na proměnné, které vyhovují filtrovacím kritériím.
5. Dvojitým kliknutím na název proměnné se dialog zavře a proměnná je vložena do žebříčku. Nyní ji lze propojovat s funkčními bloky.



2.4.8 Softwarové proměnné

Je také možné definovat *softwarové proměnné*. Ty nejsou součástí žádného funkčního bloku ani nejsou připojeny k hardwaru. Příkladem softwarových proměnných jsou *systémové proměnné*.

Podívejme se na systémové proměnné (které jsou nyní jedinými softwarovými proměnnými v projektu). Odfiltrujeme proměnné podle prefixu – tím bude **SW**.

<input type="checkbox"/>	OPC	Name prefix	Name	Value	Unit	Type	UpdateTime	Quality	Blocks	Origin
		SW								
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	SW	System.RTC	25.07.2006 10:55:32.980		DateTime	25.07.2006 10:55:32.	Good	N/A	SW
<input type="checkbox"/>	<input type="checkbox"/>	SW	System.RTCDaylight	false		Boolean	25.07.2006 10:55:32.	Good	N/A	SW
<input type="checkbox"/>	<input type="checkbox"/>	SW	System.RTCUTC	25.07.2006 08:55:32.980		DateTime	25.07.2006 10:55:32.	Good	N/A	SW
<input type="checkbox"/>	<input type="checkbox"/>	SW	System.StopAllowed	false		Boolean	25.07.2006 10:55:32.	Good	N/A	SW
<input type="checkbox"/>	<input type="checkbox"/>	SW	System.StopRequest	false		Boolean	25.07.2006 10:55:32.	Good	N/A	SW
<input type="checkbox"/>	<input type="checkbox"/>	SW	System.WaitForApplicatio...	false		Boolean	25.07.2006 10:55:32.	Good	N/A	SW
6										

System.RTC: Poskytuje aktuální systémový čas v místním formátu.

System.RTCDaylight: Stav true znamená, že je aktivní letní čas.

System.RTCUTC: Poskytuje aktuální systémový čas ve formátu Universal Time Coordinated.

System.WaitForApplicationStop: Pokud je false (default), systém nečeká na *System.StopAllowed* a po požadavku na zastavení programu (vyvolaném tlačítkem „Stop“ v okně runtime) zpracování funkčních bloků ihned ukončí.

Pokud je true, požadavek na stop nastaví proměnnou *System.StopRequest* na True a systém běží, dokud není *System.StopAllowed* ve stavu True. Proměnná *System.StopAllowed* musí být nastavena aplikací, např. pokud je zavřen ventil na páře nebo je dochlazen elektroohřev.

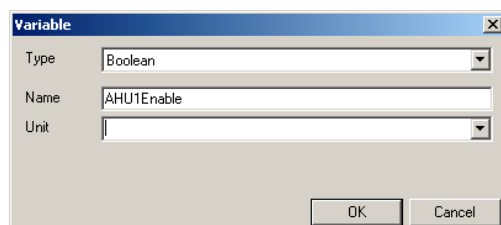
Tato funkce umožňuje řídit ukončení programu a podmínit ho splněním určitých podmínek.

Systémové proměnné

Vložení nové proměnné

Novou softwarovou proměnnou můžeme založit v záložce **Variables viewer**:

1. Přejdeme do **Variables viewer**
2. Stiskneme <Ins>



3. Vybereme typ proměnné, zadáme název a volitelně jednotku (výběrem ze seznamu nebo zapsáním jiné, dosud nepředdefinované jednotky)
4. Klikneme na **OK**. Nová proměnná se objeví v seznamu.

K čemu softwarové proměnné?

Softwarové proměnné se používají především pro přenos signálů z jednoho schématu do druhého. Ve zdrojovém schématu se proměnná vloží do výstupního žebříčku (napravo), zatímco ve schématu cílovém do žebříčku vstupního. Tak se signál přenáší mezi schémata.

Z proměnné může být čteno na více místech, zapisovat se do ní však může jen na jednom místě – z jednoho funkčního bloku. Podívejme se nyní na funkční bloky.

2.4.9 Funkční bloky

Podrobný popis funkčních bloků a jejich vlastností najdete v Nápovědě (klávesou F1 na zvýrazněném funkčním bloku).

Funkční bloky si vyměňují data s aplikací pomocí vstupů, parametrů a výstupů. Dohromady se vstupům, parametrům a výstupům říká svorky.

Některé svorky mohou být definovány buď uvnitř (*inside*) nebo vně (*outside*) funkčního bloku.

Uvnitř bloku: hodnota je pevná a lze ji měnit pouze jako parametr (přes LCD displej, touch screen nebo řídicí stanici).

Vně bloku: svorka musí být připojena k proměnné nebo výstupu jiného bloku (což je ostatně také proměnná) a mění se během chodu programu. Nelze ji ovlivňovat z HMI nebo řídicí stanice (s výjimkou případu, kdy "zdrojový" blok je v manuálním módu, viz níže).

Vstupy

B30				
Inputs		Parameters	Outputs	
Name	Outsi...	NOTed	Type	Description
enable	<input type="checkbox"/>	<input type="checkbox"/>	Boolean	If enable=true then the controller is switched off and by=false
w	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Double64	Setpoint
x	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Double64	Signal input

Vstupy poskytují funkčním blokům hodnoty, které bloky zpracovávají. Dvoustavové (binární) hodnoty typu bool mohou být na vstupech invertovány zaškrtnutím boxu **NOTed**.

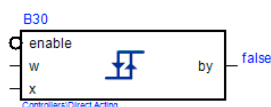
Každý vstup má několik vlastností, které jsou vypsány v tabulce:

Name: Popis svorky - zkratka, která je v rámci bloku unikátní. Je součástí jména proměnné a nelze ji měnit.

Outside: Při zaškrtnutí se svorka objeví vně bloku a lze ji připojit k jinému bloku nebo proměnné.

NOTed: Při zaškrtnutí je binární hodnota negována. Zvenčí je negace znázorněna malým kroužkem před svorkou.

Blok s invertovaným vstupem **enable**.



Type: Typ proměnné (Boolean, Double64, Integer64, String, DateTime). Není možné měnit.

Description: Krátký popis funkce svorky v logice bloku. Podrobný popis funkce bloku získáme skokem do bloku a stisknutím <F1>. Nápověda se otevře na popisu aktuálního bloku.

Kontextová nápověda v bloku

Parameters

B30							
Inputs		Parameters			Outputs		
Name	OPC	Outside	NOTed	Default value	Unit	Type	Description
Xp	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1		Double64	Bandwidth or hysteresis
Of	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0		Double64	Adds to setpoint / moves the whole sequence

Parametry jsou funkčně podobné vstupům. Mají však některé vlastnosti navíc:

OPC: Při zaškrtnutí se parametr objeví mezi proměnnými vybranými pro OPC a exportuje se do definičního souboru OPC, tedy do seznamu proměnných, které budou integrovány do RcWare Vision nebo jiné vizualizace. Zaškrtněte, pokud má být parametr viditelný a ovladatelný na uživatelské úrovni (parametry ekvitermní křivky, noční útlumy atd.).

Default value: Pokud parametr není venku (*Outside*), zde je jeho hodnota. Je možné ji kdykoli za běhu programu měnit, buď z vizualizace (pokud je zaškrtnuto OPC), nebo z IDE. Hodnoty se ukládají, takže po restartu runtime jsou zachovány.

Každý funkční blok má alespoň jeden výstup (kromě bloku B108, Beep).

Outputs

B84							
Inputs		Parameters		Outputs			
Name	OPC	Outside	Manual	Manual value	Unit	Type	Description
byn	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	true		Boolean	Negation of by
by	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	false		Boolean	
Tres	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0	s	Double64	

Výstupy jsou výsledky výpočtů bloku. Výstupy mají jisté zvláštní vlastnosti:

Manual: Při zaškrtnutí se na výstupu neobjeví vypočítaná hodnota, ale hodnota ručního přeřazení.

Manual value: Hodnota posílaná na výstup pokud je výstup v módu *Manual*.

Přeřazení je užitečné při testování a uvádění do provozu. Ve schématu jsou přeřazené svorky označeny žlutou ručičkou. Pokud je v programu alespoň jeden blok přeřazen, žlutá ručička je i na liště s nástroji.

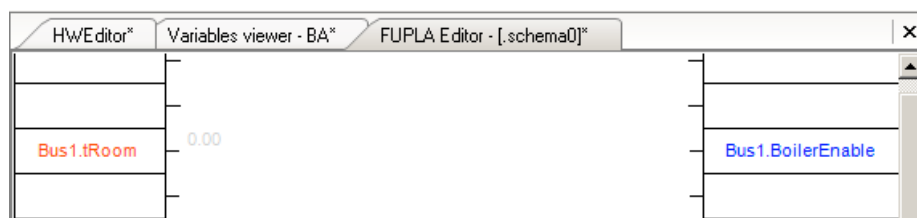


2.4.10 Vkládání funkčních bloků

Sestavme jednoduchý regulátor – pokojový termostat:

1. V testovacím projektu definujeme jeden vstup a jeden výstup:
 - čidlo teploty
 - povolení kotle.
2. Vložíme proměnné do žebříku prázdného schématu: čidlo vlevo, povolení kotle vpravo.
Viz 2.4.7, Vložení proměnné do schématu.

Proměnné ve schématu

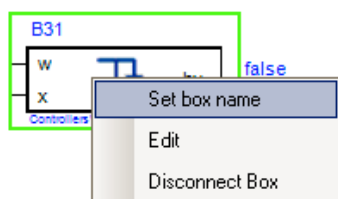


3. V panelu funkcí **Function block panel** přejdeme do **Basic analogue boxes, Controllers, Reverse acting (Heating), On/Off control (B31)**.
4. Přetáhneme tento blok do schématu.
5. Blok můžeme po schématu posouvat a umístit na vhodnou pozici. Je-li ve schématu více bloků, aktivní blok vybereme kliknutím do jeho obvodu. Aktivní blok je označen zeleným rámečkem.

Vybraný blok



6. Pravý klik do bloku vyvolá kontextové menu:



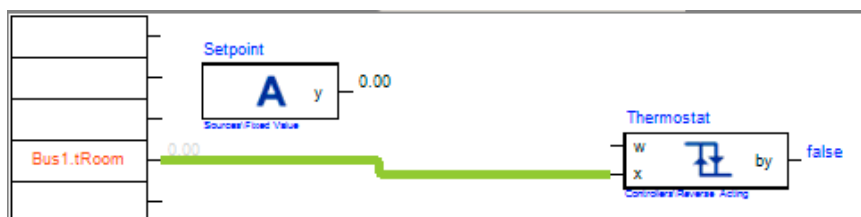
Vyberte **Set box name** a přejmenujte blok, pokud je to nutné. Dbejte na to, že název bloku je součástí jména proměnné jak v SoftPLC, tak v OPC (SCADA).

7. Z cvičných důvodů pojmenujte blok *Thermostat*. Potvrďte <Enter> nebo **OK**.
8. Nyní se podívejme do okna **Variables viewer**. Nastavme filtr v kolonce **Name** na např. *Ther* (podřetězec „Thermostat“) a podívejme se na seznam odpovídajících proměnných. Všechny z nich se vztahují ke bloku *Thermostat*.

2.4.11 Propojování funkčních bloků

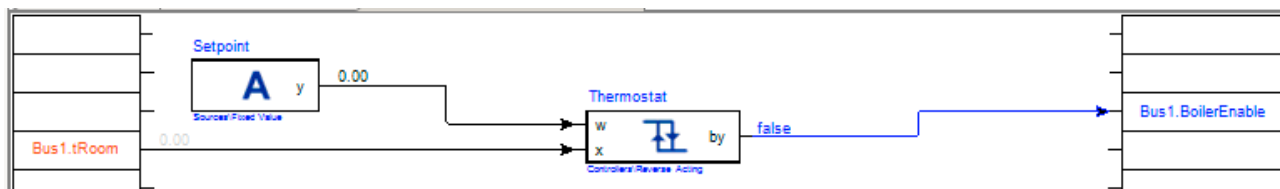
Vraťme se nyní do záložky **FUPLA Editor**. Nyní připojíme blok k proměnným, takže proměnné budou blokem zpracovávány.

1. Vložme proměnnou s požadovanou hodnotou. Může to být buď SW proměnná nebo jiný funkční blok (například pevná hodnota Basic analogue boxes, Sources, Fixed value (B1)). Softwarová proměnná tolik nezatěžuje systém a možná je v tomto případě přehlednější.
2. Nastavme požadovanou hodnotu na 20°C.
3. Přejmenujme blok B1 na *Setpoint*.
4. Proměnná s teplotou z čidla má v žebříku „drátek“. Po najetí kurzorem na drátek se šipka změní v ručku.
5. Levým tlačítkem myši přetáhneme zelený spoj až na svorku „x“ bloku *Thermostat*. Spoj se automaticky uchytl na svorku.
6. Pustíme levé tlačítko myši.



Nesoulad typů signálů

7. Proměnná je nyní připojena na svorku „x“ regulátoru.
8. Pokud typy spojovaných signálů nesouhlasí (například při pokusu propojit proměnnou Boolean na vstup Double), spoj změní barvu na červenou a k cílové svorce se nepřichytí. Totéž se stane, pokud do cílové svorky již zapisuje jiná proměnná.
9. Obdobně propojíme blok *Setpoint* na svorku „w“ a výstup bloku na proměnnou povolení kotle (*Boiler enable*) (ta v našem příkladě řídí fyzický výstup – relé v I/O modulu, které spíná kotel).



Aplikace je nyní připravená ke kompilaci.

2.4.12 Kompilace

Ikona kompilace

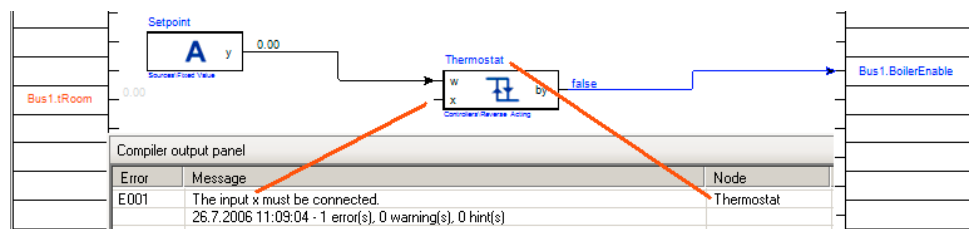


1. Klikněte na ikonu kompilace **Compile** na liště s nástroji *nebo* stiskněte <Ctrl>+<Shift>+ *nebo* vyberte **Local, Compile** z menu.
2. IDE se zeptá, zda má být projekt uložen. Potvrďte **Ano**. Neuložený projekt nelze kompilovat. Všimněte si, že po uložení zmizí hvězdičky ze záložek panelů (= bez neuložených změn).
3. Ve spodní části pracovního panelu se otevře okno kompilátoru *Compiler output panel* a zobrazí výsledek kompilace.

Error	Message	Node
	26.7.2006 11:05:21 - 0 error(s), 0 warning(s), 0 hint(s)	

Lokalizace chyby

4. Jsou-li v programu chyby bránící překladu, vypíše se v panelu překladače a je třeba je opravit. Následuje příklad nejčastější chyby. Sloupec **Node** zobrazuje název bloku. Levým dvojklikem na řádku s chybou se označí příslušný blok, což zjednodušuje lokalizaci chyby.



5. Po úspěšné kompilaci můžeme projekt spustit.

2.4.13 Hlášení kompilátoru

Hlášení kompilátoru jsou trojího typu:

1. **Exxx** – (Error) – chyba bránící kompilaci, je nutno ji opravit

"E001" : "The input ... must be connected." – nepřipojený vstup

"E002" : "The parameter ... must be connected." – nepřipojený parametr

"E003" : "The input ... is connected to an output with incompatible type." – nekompatibilita signálů

"E004" : "The parameter ... is connected to an output with incompatible type." - nekompatibilita signálů

EMxxx – (Error MiniPLC) – chyby specifické pro MiniPLC

"EM001" : "Function block ... is not implemented under MiniPLC platform. You must replace it with something else." – funkční blok je možné použít pouze v projektu Full featured SoftPLC (viz projekt, kontextové menu Properties, Platform). Smažte jej a jeho funkčnost nahraďte kombinací bloků implementovaných pro MiniPLC.

"EM002" : "Unsupported channel type in the HW definition: You must remove it from the project." – Tento typ kanálu není pro MiniPLC implementován.

"EM003" : "Unsupported device driver in the HW definition: You must remove it from the project." – Tento driver není pro MiniPLC implementován.

"EM004" : "Unsupported serial port number [...] in the HW definition. Only ports 1-3 are supported." – MiniPLC má pouze COM porty 1, 2 a 3.

"EM005" : "The HW variable \"...\" is not a part of a device box. You must remove it from the project." – Hardwarová proměnná není obsažena v modulech a driver jí nerozumí. Odstraňte ji z projektu.

2. **Wxxx** – (Warning) – varování, kompilace je možná, ale stav může způsobit to, že se aplikace chová jinak než autor zamýšlel (např. v projektu jsou dvě proměnné stejného jména, což je díky rozlišení pomocí GUID možné, ale není vyloučeno, že signál bude omylem v grafice přiveden do jedné a čten z druhé)

"W001" : "The variable ... should not appear among the outputs." – do proměnné se neočekávaně zapisuje

"W002" : "The node ... is obsolete. Consider its replacement." – blok byl v knihovně nahrazen novější verzí, doporučuje se jej smazat a nahradit novým blokem stejného typu z knihovny

"W003" : "The name of the variable ... is not unique." – dvě proměnné stejného jména

3. **Hxxx** – (Hint) – tip pro optimalizaci nebo efektivnější zapojení bloků.

2.5 Spouštění programu

Program může běžet buď **místně**, nebo na **vzdálené** procesní stanici.

S dálkovým připojením je možné:

- připojit se k podstanici protokolem TCP/IP
- konfigurovat runtime
- nahrát do podstanice novou verzi aplikace
- spustit a zastavit aplikaci
- monitorovat a měnit proměnné za běhu aplikace
- nahrát aplikaci a její parametry zpět do IDE jako zálohu.

To je typické použití IDE. My však nyní spustíme aplikaci přímo v IDE (local connection) – I/O moduly musejí být připojeny na COM počítače, kde běží IDE.

Existují tři módy běhu:



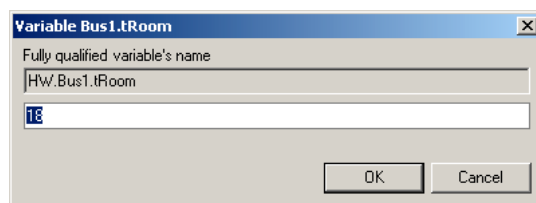
Full run

Plný běh – jede jak komunikace, tak vyhodnocování bloků. To je normální chod programu. Pozor, v nelicencované verzi se **komunikace po deseti minutách zastaví**.



Evaluation only

Pouze vyhodnocování bloků – komunikační driver je neaktivní („odpojen“). Hodnoty vstupů lze nastavovat manuálně po kliknutí na proměnné v žebříku vlevo na schématech.



Když je možné nastavit hodnotu proměnné ručně, po kliknutí na proměnnou se otevře dialogové okno. Nastavená hodnota není přepsána nakomunikovanými hodnotami z I/O modulů, protože komunikace neběží.

K čemu je vyhodnocování ?



Tento mód můžeme použít pro zkoušení programu ve chvíli, kdy ještě nejsou oživeny periferie a technologie není funkceschopná.

Communication only

Pouze komunikace – pro kontrolu periférií, aniž bychom je ovlivňovali výstupy z aplikace. Jsou aktivní pouze komunikační drivery, funkční bloky se nevyhodnocují a nezapisují do HW proměnných.

Aktuální hodnoty vstupních proměnných se zobrazují v záložce **Variables viewer** i v žebřících schémat.

Výstupní proměnné lze nastavit manuálně zápisem do žebříčku.

Spustíme aplikaci nejprve ve vyhodnocovacím módu. Později připojíme I/O modul a čidlo.

1. Klikneme na ikonu Evaluation only
nebo
vybereme **Run – Run evaluation only** v menu.
2. Vedle výstupních svorek bloků se budou online zobrazovat jejich digitální a analogové hodnoty. Tuto funkci lze přepínat tlačítkem **Show links' values** na liště s nástroji.
3. Nastavte požadovanou teplotu v místnosti na 20°C v bloku *Setpoint*.
4. Nastavte teplotu v místnosti na 17°C (viz Evaluation only). Výstup je ve stavu *True*.
5. Nastavte teplotu v místnosti na 23°C a sledujte výstup regulátoru. Přejde do stavu *False*. Kotel vypnul, protože teplota v místnosti se zvýšila.

Další doporučené kroky:

6. Pohrajte si s nastavením parametrů bloku Thermostat.
7. Místo termostatu vložte PI regulátor a ovládejte spojité ventily.
8. Přidejte časový program, který bude řídit noční útlum. Použijte buď binární program (HVAC – TPG - ... v knihovně bloků) a přepínač, nebo analogový časový program.
9. K regulátoru přidejte svorku Enable (blokovanou např. signálem z EPS).
10. Vložte do projektu další bloky (např. ekvitermní křivku) a navrhnete kompletní topný okruh s ekvitermním řízením a korekcí podle místnosti..

Zobrazení hodnot proměnných



3 Licencování

3.1 Základy

Protože SoftPLC může pracovat na jakékoli platformě PC (ne nutně IPCT.1 nebo jiný hardware Domat Control System), IDE i runtime jsou licencovány.

Licence má podobu:

- hardware (klíč pro COM port) *nebo*
- software (soubor).

Hardwarový klíč

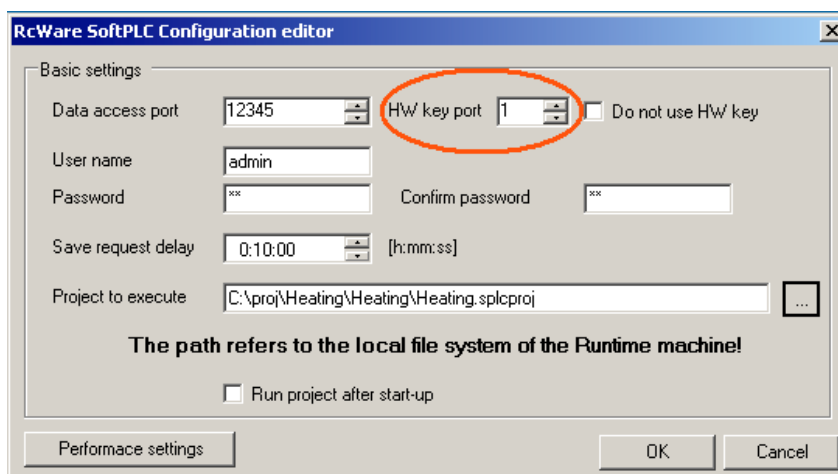
Hardwarový klíč (objednáací číslo **HWS01**) se připojuje na sériový port stroje s běžícím softwarem. Je transparentní pro veškerou komunikaci RS232 a může (ale nemusí) být instalován na stejném portu, který komunikuje s I/O moduly. Umožňuje zpracovávat neomezené množství datových bodů jak v runtime, tak v IDE. Je **volně přenosný** (není vázán na určitou podstanici nebo PC) a nevyžaduje instalaci ani aktivaci.

Tento způsob licencování je velmi flexibilní a užitečný především pro testy, uvádění do provozu a nouzové stavy.

3.2 Hardwarový klíč

Konfigurace runtime pro licencování hardwarovým klíčem:

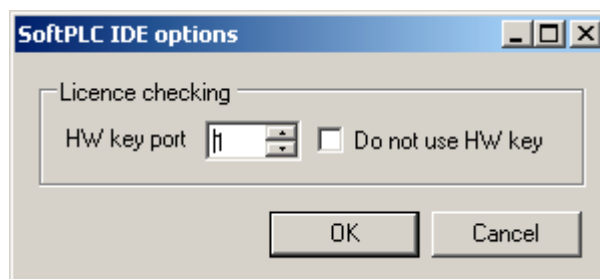
1. V runtime vyberte **Options**.
2. Nastavte číslo COM portu, na nějž je připojen hardwarový klíč.



3. Zkontrolujte, že položka „Do not use HW key“ je nezaškrtnutá.
4. Klikněte na OK.
5. Zasuňte HW klíč do příslušného portu.
6. Restartujte runtime.

Konfigurace IDE:

V menu **Tools – Options** nastavte číslo COM portu, na kterém bude hardwarový klíč.



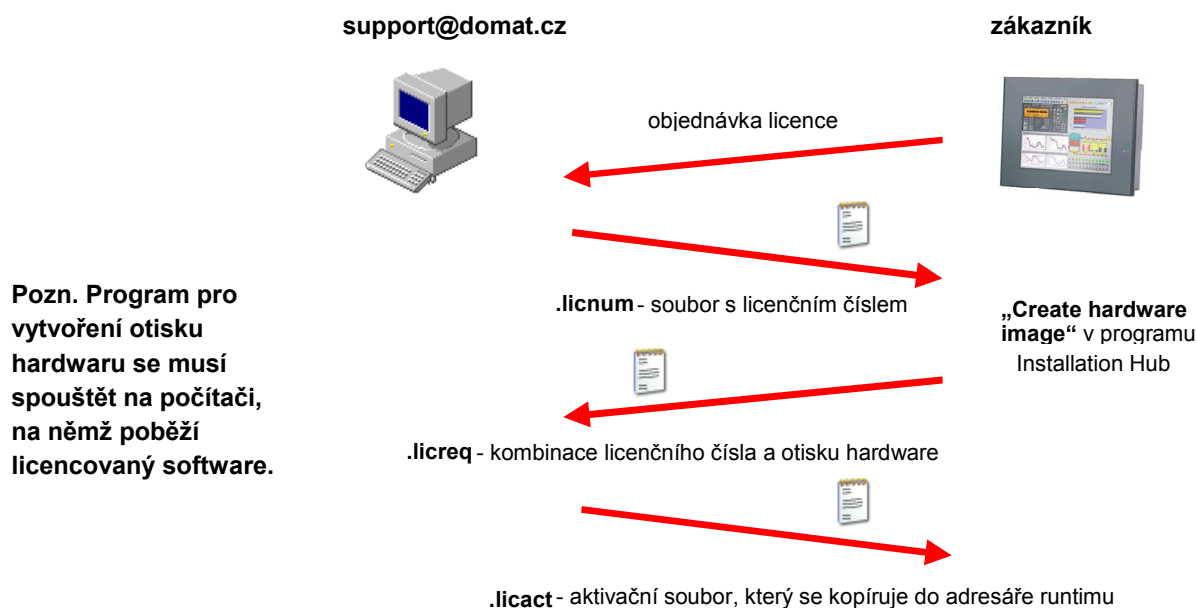
Pak zasuňte HW klíč do příslušného portu a restartujte IDE.

Pokud klíč nepracuje správně, zkopírujte soubor **ESG.IO.Serial.dll** (v **C:\Program Files\RcWare\SoftPLC\Channels**) do adresáře **C:\Program Files\RcWare\SoftPLC**.

3.3 Softwarové licencování

Pokud máme vyhrazený hardware, jako je například procesní stanice, může být jednodušší použít softwarovou licenci. Licence se vydávají na požádání a aktivují se pro konkrétní stroj.

Licencovací proces vypadá takto:

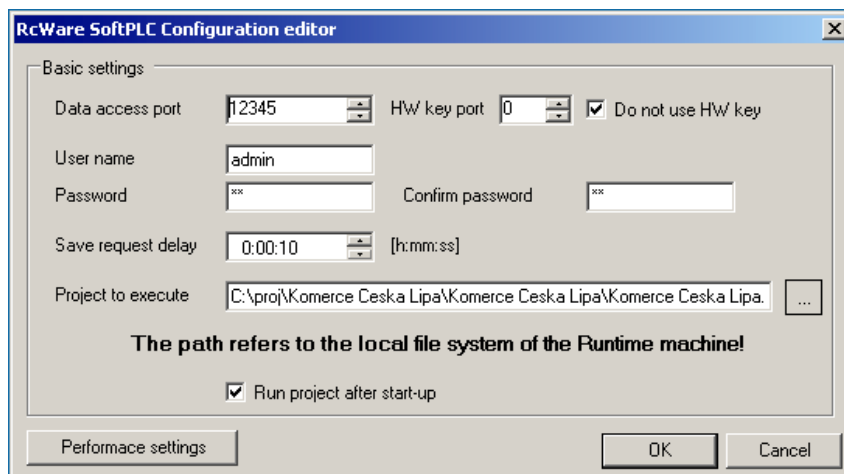


Aktivaci může provádět i zákazník na <https://licenses.rcware.eu>.

Aktivační soubor doporučujeme uschovat pro případ reinstalace operačního systému apod. Soubor lze ovšem kdykoli znovu zdarma poslat po dotazu na support@domat.cz.

3.3.1 Konfigurace runtime pro softwarové licencování:

1. Získejte aktivační soubor podle popisu výše.
2. Zkopírujte jej do adresáře s runtime (tam, kde je soubor **ESG.SoftPLC.Host.WinForms.exe**, výchozí adresář je **C:\Program Files\RcWare\SoftPLC.Runtime**).



Pozn. Pokud pracujete v OS Windows XP Embedded, po dokončení konfigurace runtime nezapomeňte uložit data (Commit CF a reboot).

3. V runtime vyberte **Options**.
4. Vyberte volbu "Do not use HW key".
5. Klikněte na OK.
6. Restartujte runtime.

Licence má neomezený počet datových bodů a neomezenou dobu platnosti. Existují nicméně "měkké limity" pro počet datových bodů připojených na I/O sběrnici – viz příručka pro projektanty.