

Kontron Czech s.r.o

U Sirotčince 353/7

460 01 Liberec

Czech Republic

tel.: +420 485 100 272

+420 485 108 636

fax: +420 485 100 273

sales@kontron-czech.comwww.kontron-czech.com

Příklad postupu vytvoření jednoduchého projektu v Aspic 3.30 z dat OPC serveru Modbus Master

Březen 2006



Zadání příkladu

Záměrem tohoto dokumentu je seznámit případné implementátory s možnostmi SCADA systému Aspic 3.30. V projektu je použit jako zdroj dat OPC server General Modbus & Jbus Master, který běží v simulačním režimu. Celý projekt lze realizovat za použití demo verzí uvedeného software, který si můžete zdarma stáhnout na našich WWW stránkách www.kontron-czech.com.

Cíl projektu:

Cílem ukázkového projektu v Aspic 3.30 je zobrazování 5ti hodnot z OPC serveru General Modbus & Jbus Master a dále zobrazování dalších 3 interních hodnot Aspic 3.30.

OPC server bude pracovat v simulačním režimu. Je třeba nastavit všem pěti hodnotách různé limity tak, aby se hodnoty nepřekrývaly a dalo se s tím dále pracovat.

Nastavení limit hodnot generovaných OPC serverem

Hodnota 1 1-10....pouze pro čtení
Hodnota 2..... 15-20....pouze pro čtení
Hodnota 3.....30-35....pouze pro čtení
Hodnota 4.....100-110....pouze pro čtení
Hodnota 5.....0-2....čtení i zápis

Paměťové hodnoty Aspic 3.30

Hodnota A.....matematicky vypočítaná sinusovka
Hodnota B.....bude ovládána pomocí tlačítka, má dva stavy
Hodnota C.....napojeno na posuvník, má 4 stavy

První vizualizační stránka musí obsahovat následující prvky:

- 1) graf s min třemi průběhy (min 2 hodnoty z OPC serveru (Hodnoty 1 a 2) + Hodnota A). Každý z průběhů má svou barvu.
- 2) min 2x objekt „Value“ (Hodnoty 1 a 2)
- 3) objekt „posuvník“ (Hodnota C)
- 4) objekt „progressbar“ (Hodnota 3)
- 5) objekt simulující svítící kontrolku (zap/vyp...zelená/černá) – (Hodnota B)
- 6) tlačítko ovládající kontrolku - dva stavy (Hodnota B)
- 7) obrázek závislý na stavech Hodnoty 5. Obrázek bude existovat ve třech stavech a bude se zobrazovat podle průběhu hodnoty.
Příklad motor stojí (šedý) = Hodnota 5 je 0
Příklad motor v chodu (zelený) = Hodnota 5 je 1
Příklad motor v poruše (žlutý) = Hodnota 5 je 2

Druhá vizualizační stránka obsahuje objekt graf přes celou stránku. Nastavení je stejné jako v grafu na stránce 1. Mezi stránkami se lze přepínat pomocí tlačítek.

Obecně:

- 1) Všechny Hodnoty 1-5 a A-C se archivují. Použito je ukládání do Aspic archivů.
- 2) V projektu jsou následující alarmy, každý si vyžaduje potvrzení operátora + každý je popsán
 - a. Zvukový alarm na Hodnotě B – aktivní v případě kdy je kontrolka zelená
 - b. Alarm na Hodnotě 5 – aktivní když je motor v poruše
 - c. Alarm na Hodnotě 1 – aktivní při dosažení horní kritické úrovně 9
 - d. Alarm na Hodnotě 2 – aktivní při dosažení horní kritické úrovně 19
 - e. Alarm na Hodnotě 2 – aktivní při dosažení dolní kritické úrovně 15
 - f. Alarm na Hodnotě 3 – aktivní při dosažení horní kritické úrovně 30
- 3) Součástí budou následující reporty:
 - a. Seznam vzniklých alarmů
 - b. Graf průběhu Hodnoty 1
 - c. Graf průběhu Hodnoty 2
 - d. Minutové hodnoty od Hodnoty 3

Seznam software použitých v příkladu:

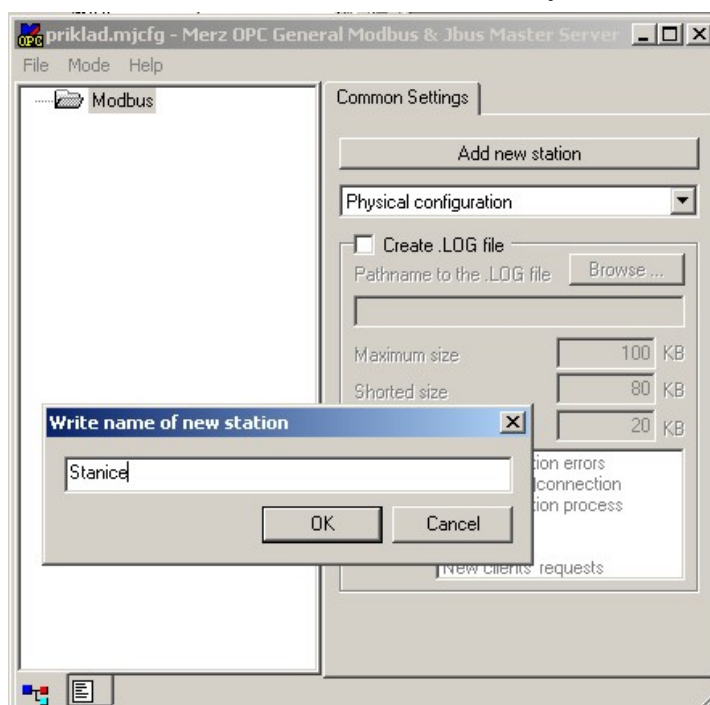
- OPC General Modbus & Jbus Master Server (ke stažení na stránkách www.kontron-czech.com)
- Aspic 3.30 (ke stažení na stránkách www.kontron-czech.com)

OPC server, zdroj dat a jeho konfigurace

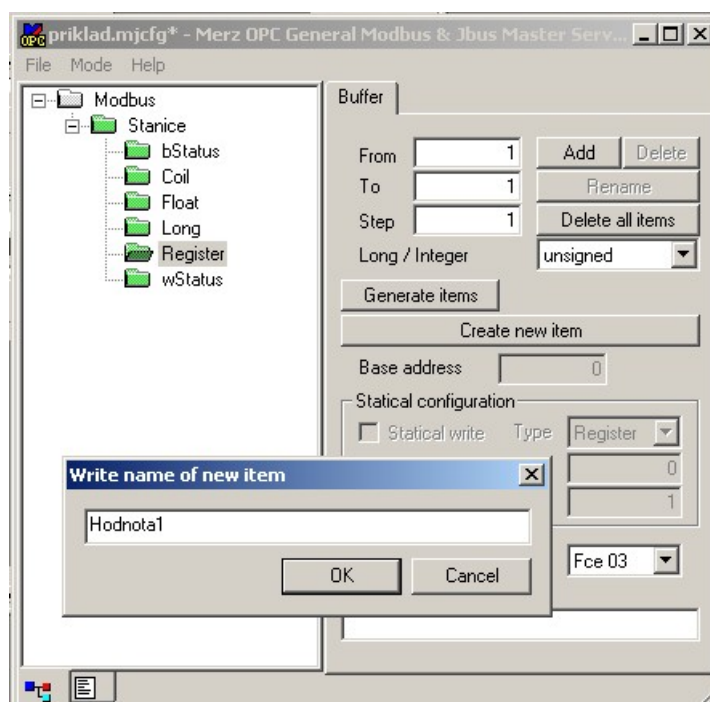
Zdrojem dat pro náš příklad bude OPC server Modbus & Jbus spuštěný v simulačním režimu. Simulační režim OPC serveru umožní generovat data (náhodná data) bez připojení k PLC. Výchozím požadavkem je nainstalovaný OPC server Modbus & Jbus. Jeho nastavení dle požadavků našeho příkladu si ukážeme v následujícím postupu.

Postup nastavení:

1. Spustíme konfiguraci OPC serveru a založíme novou stanici se jménem „Stanice“.

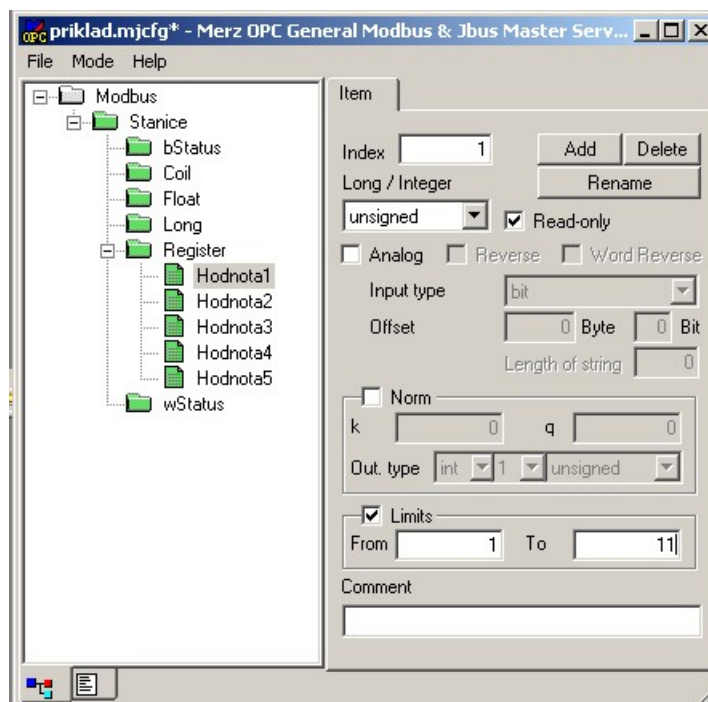


2. Do bufferu „Register“ přidáme všech pět názvů hodnot s kterými budeme v příkladu pracovat. Pojmenujeme je „Hodnota1“ až „Hodnota5“.

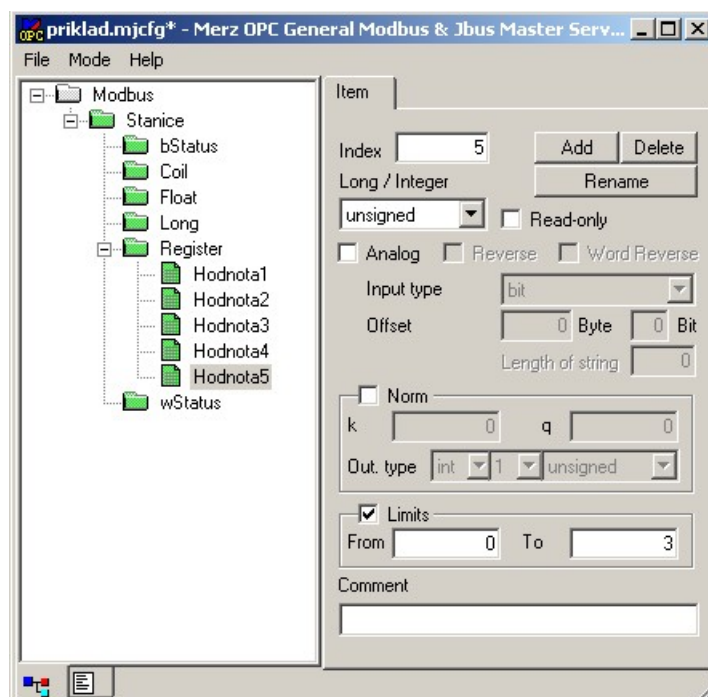


- Pro „Hodnota1“ nastavíme následující vlastnosti dle zadání. Index na 1, typ „unsigned“, zaškrtneme vlastnost „Read-only“ a nastavíme limity od 1 do 11.

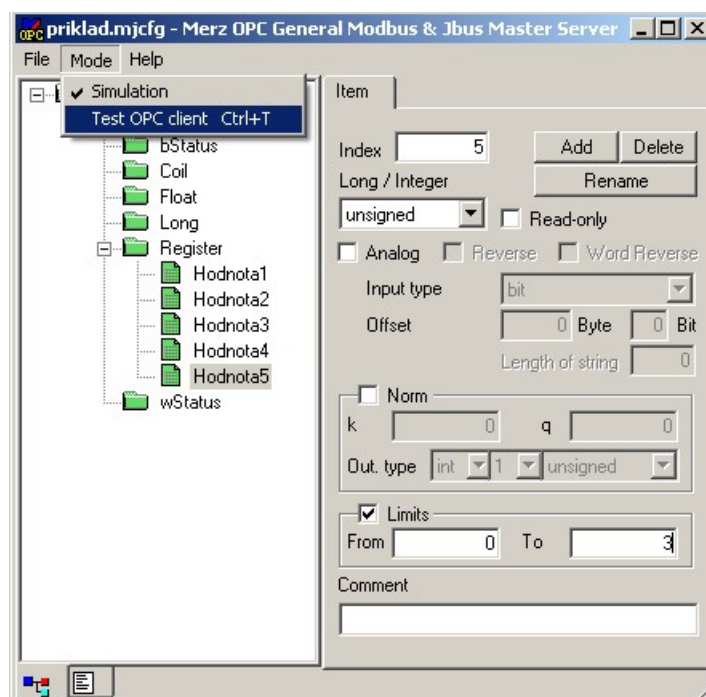
Poznámka: Nastavení limit je nutné, jelikož požadujeme, aby simulační režim OPC serveru generoval hodnoty pouze v námi žádaném rozsahu. Pro „Hodnota1“ byl požadavek, aby hodnota nabývala velikosti 1 až 10. Simulační server generuje hodnoty do zadané nikoliv včetně zadané. Jelikož pracujeme s celými čísly, musíme nastavit hodnotu 11.



- Obdobně nastavíme dle zadání i vlastnosti ostatních hodnot. Hodnotu vlastnosti index použijeme jedinečnou pro každou hodnotu. Pro příklad uvádím „Hodnota5“, která se od předchozích liší nastavením vlastnosti „Read-only“. V našem příkladu toto nastavení nemá velký význam. Nastavení by se uplatnilo, pokud by OPC server nepracoval v simulačním režimu, ale byl připojen k PLC.

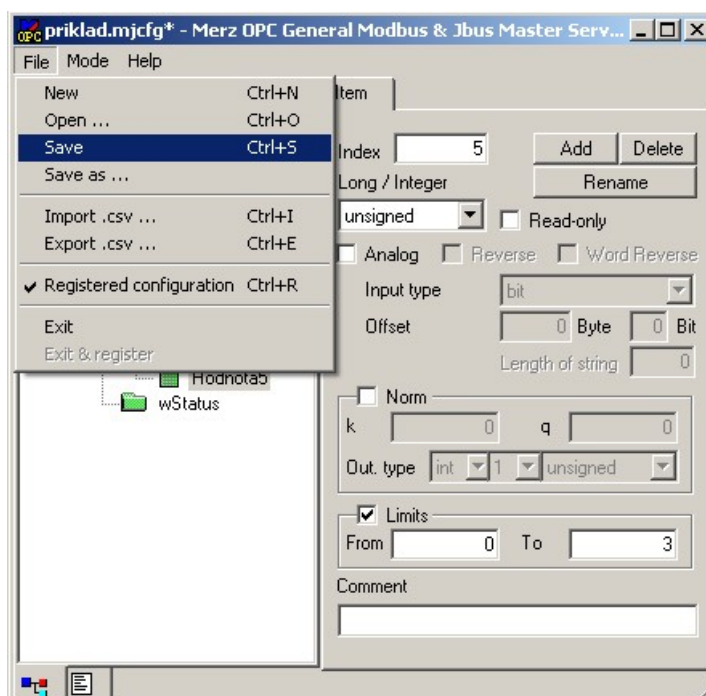


5. Nastavíme simulační režim OPC serveru

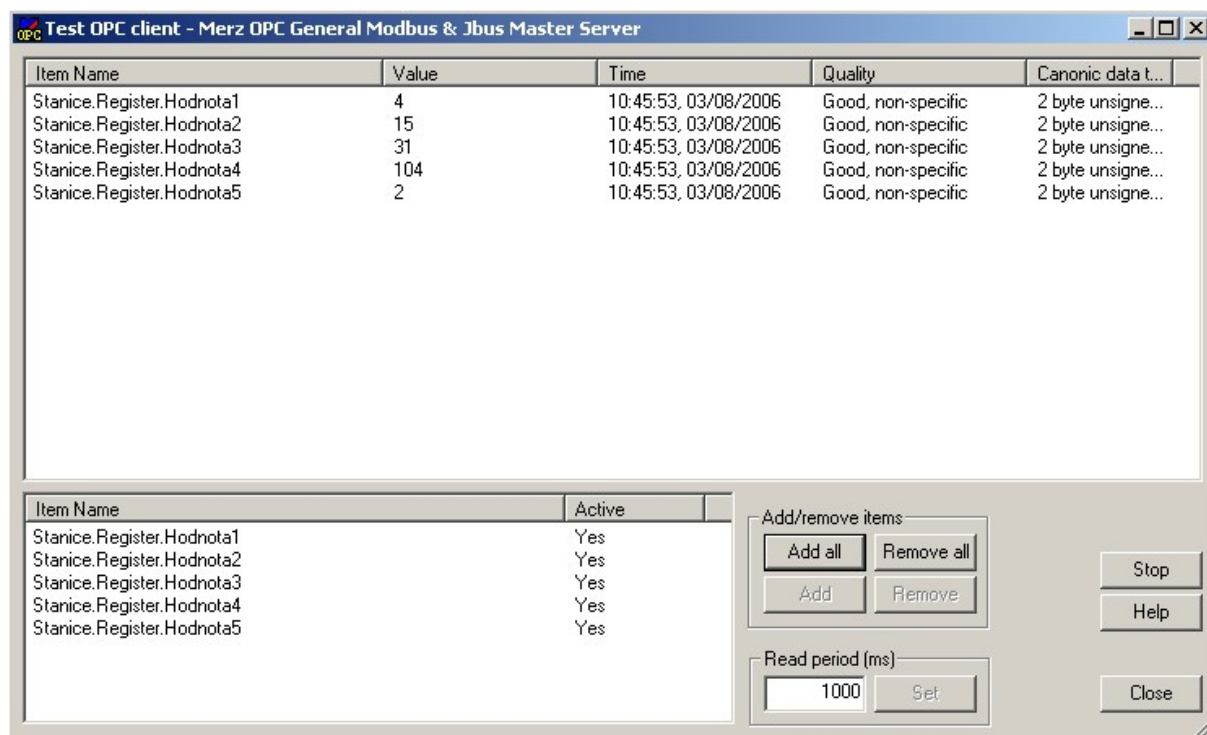


6. Vytvořenou konfiguraci uložíme a označíme tuto konfiguraci za registrovanou.

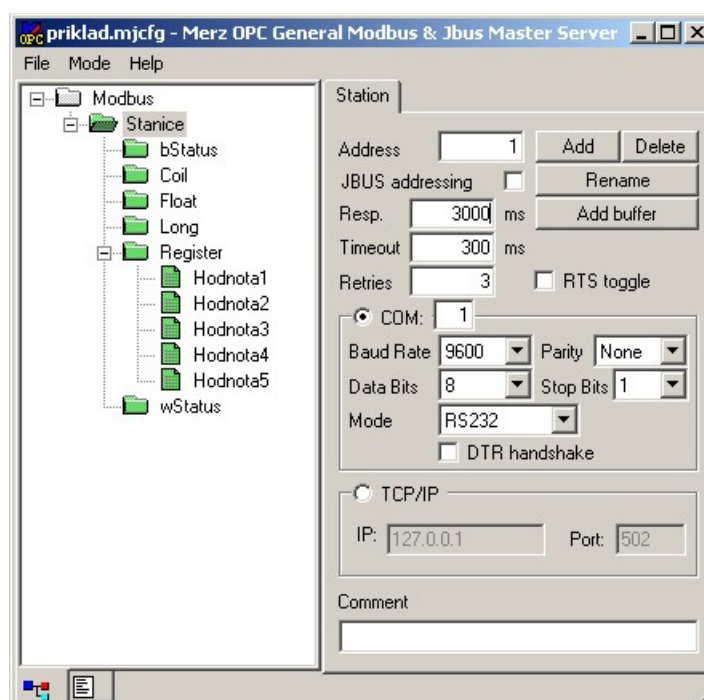
Poznámka: Při spuštění serveru je načítána registrovaná konfigurace. Registrovaná konfigurace může být nejvýše jedna.



7. Spustíme testovacího OPC klienta a přidáme vše. Pokud je zobrazena kvalita „dobrá“ a dochází ke změnám hodnot, je zdroj dat „OPC server v simulačním režimu“ pro náš příklad připraven.



Poznámka: Pro přehlednost našeho příkladu je vhodné nastavit rychlost občerstvování dat (vyčítání dat z PLC nebo v našem případě generování dat simulačním režimem serveru na hodnotu 3000 ms. Změny hodnot budou pomalejší a bude je možno lépe sledovat. Nastavení provedeme změnou hodnoty vlastnosti „Resp.“ na 3000 ms dle následujícího obrázku. Ostatní nastavení (Address, COM port) jsou v našem případě nepodstatná a uplatní se pouze pokud je OPC server připojen k PLC (není použit simulační režim).

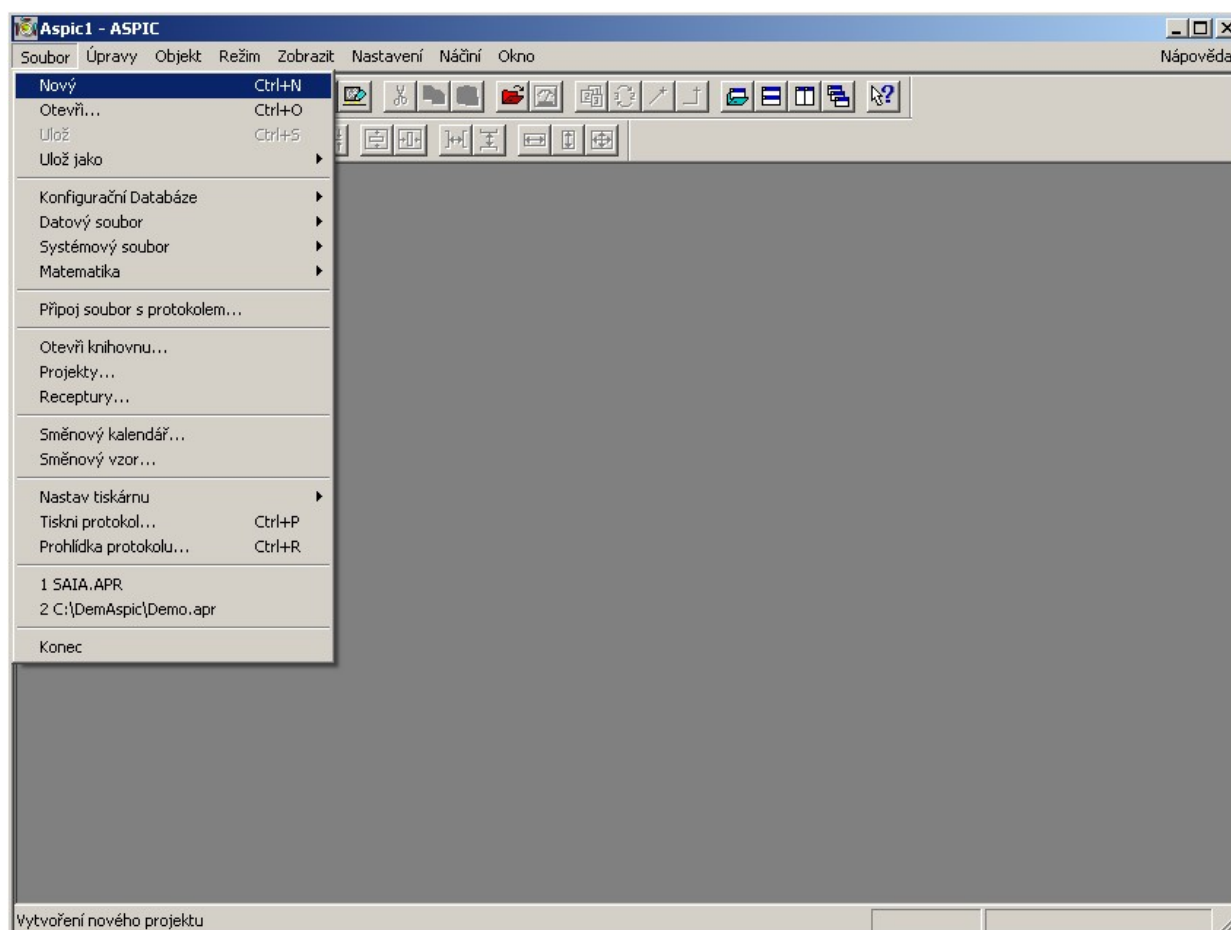


Aspic 3.30, vytvoření projektu dle zadání příkladu

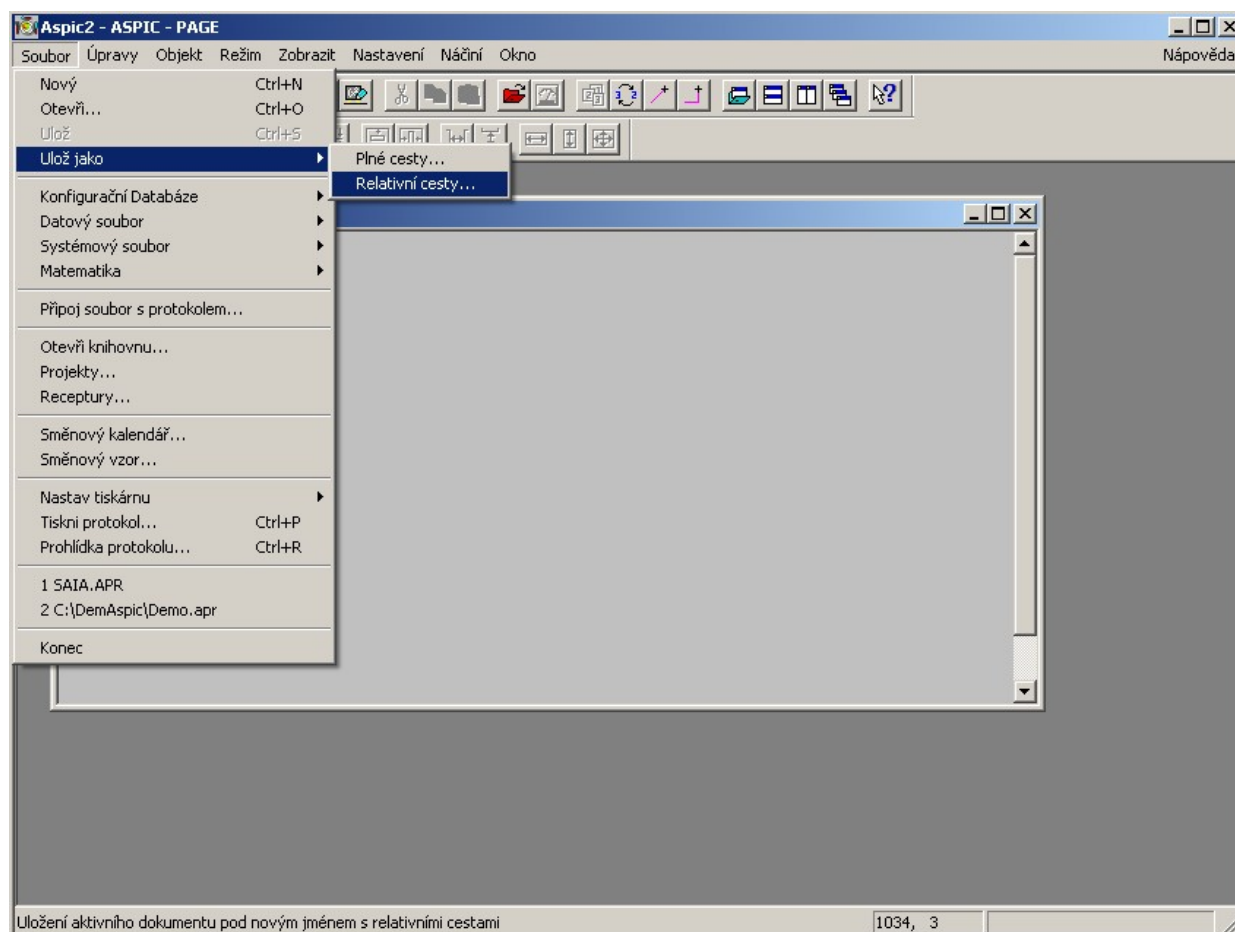
Podrobné návody pro práci se systémem naleznete na stránce Aspic 3.30 On-line Wizard (<http://www.kontron-czech.com> > Produkty > SCADA HMI > Aspic 3.30).

Postup vytváření projektu:

1. Vytvoříme nový projekt a Aspic 3.30

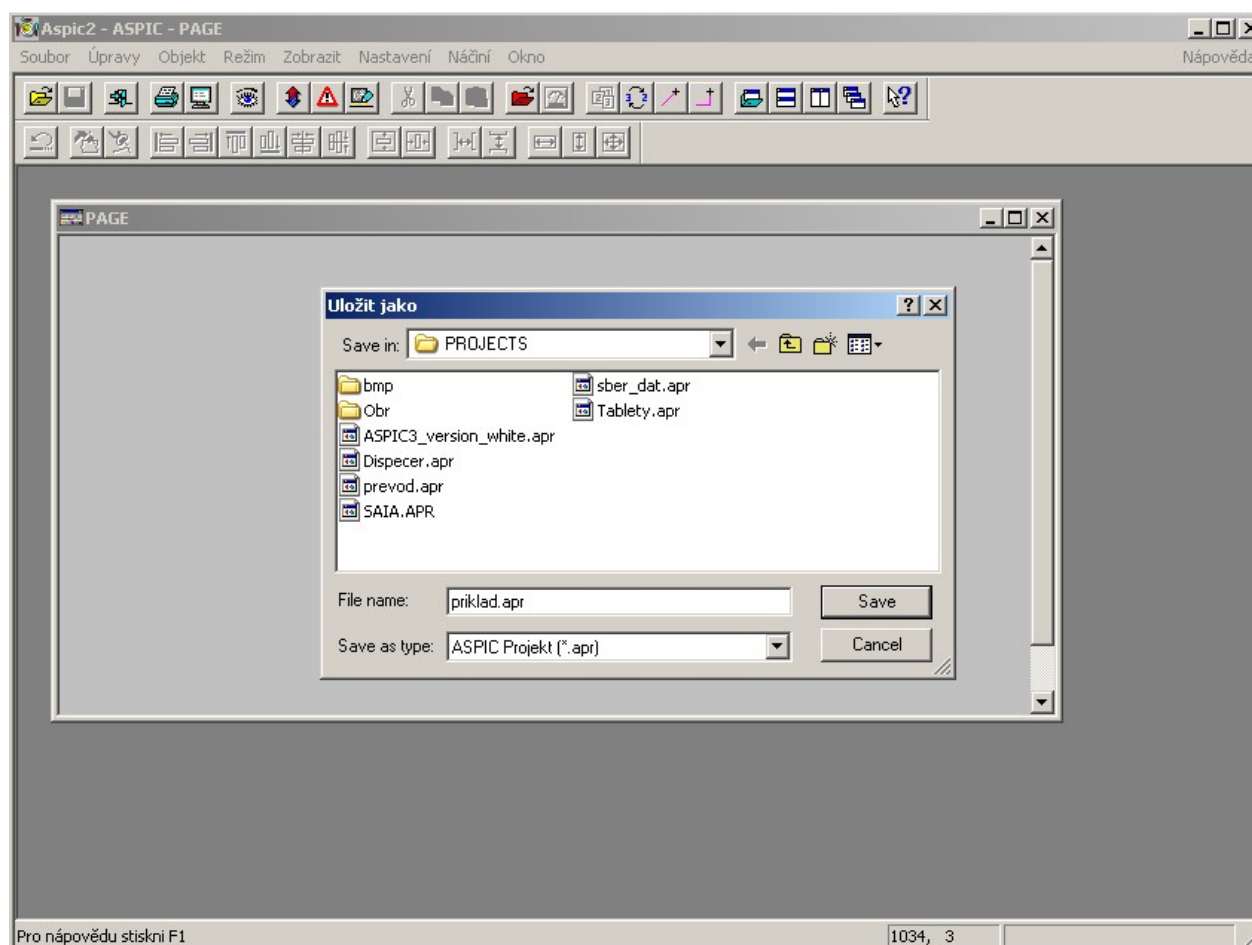


2. Nově vytvořený projekt uložíme s relativní cestou

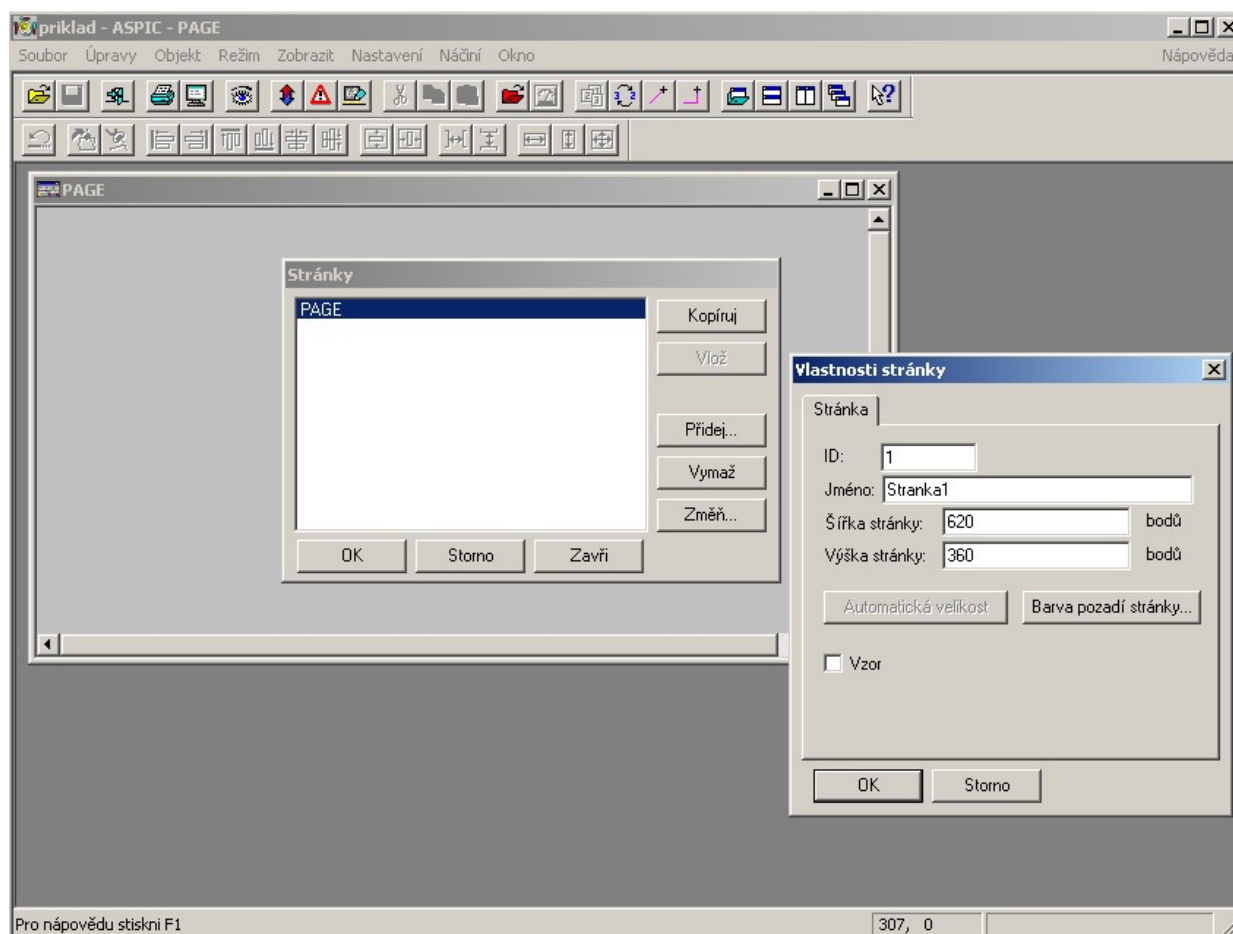


Poznámka: Uložení projektu s relativní cestou je výhodné pokud budeme chtít v budoucnu přesunout projekt do jiného adresáře.

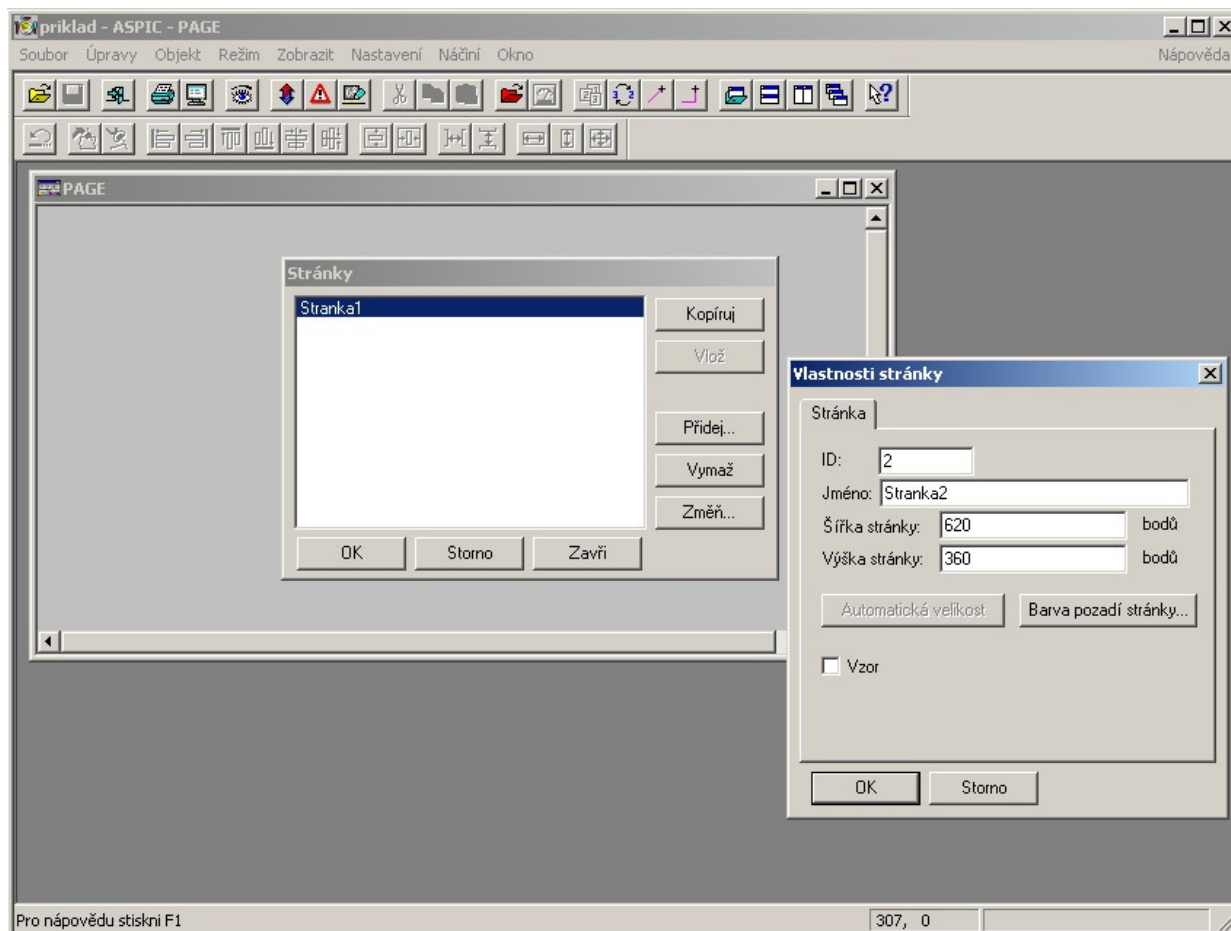
3. Projekt pojmenujeme „příklad“ a uložíme všechny jeho části. Projektový soubor *.apr, systémový soubor *.asf a datový soubor *.adf. Pro přehlednost je vhodné, aby všechny tři soubory měly stejný název.



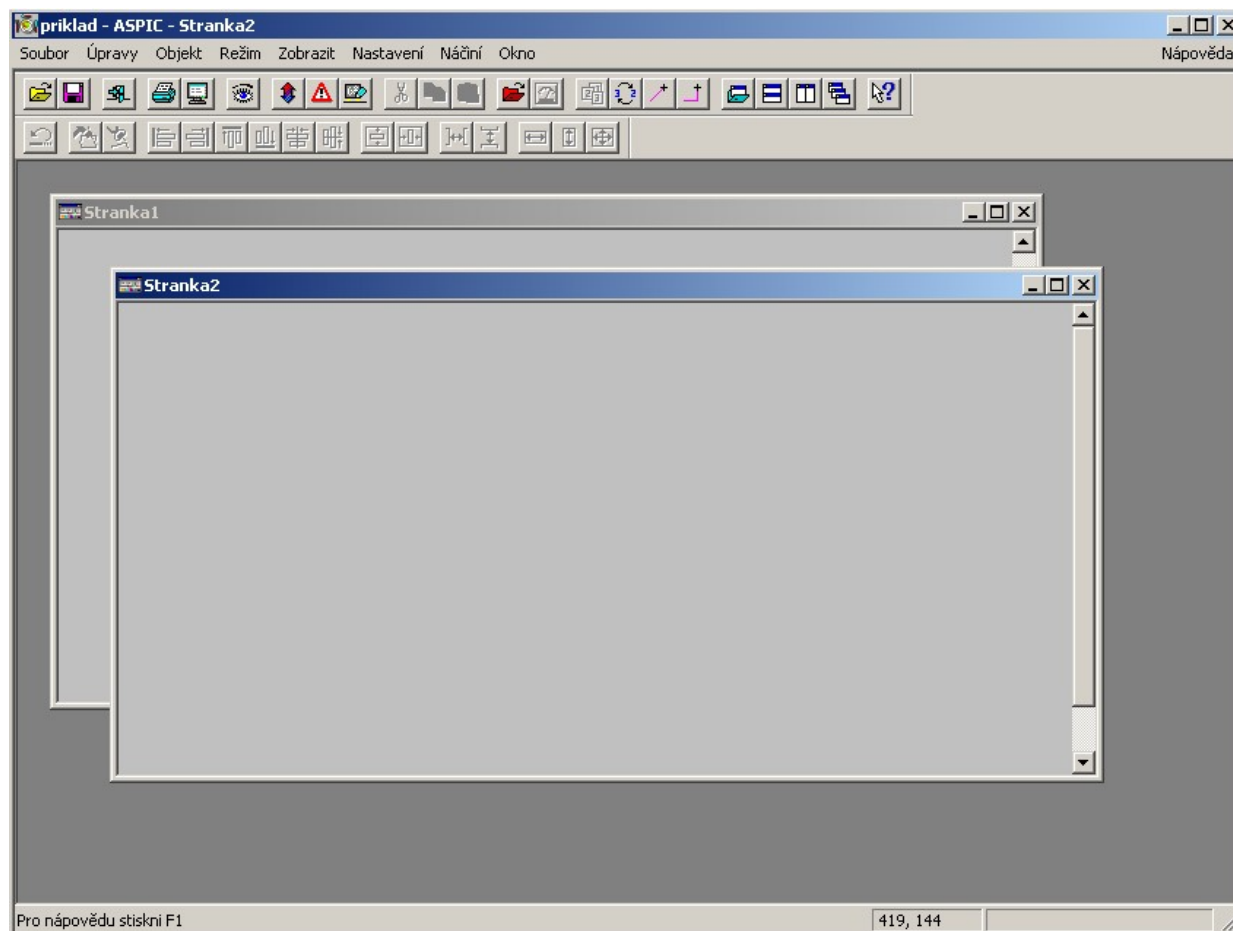
4. Spustíme nastavení vlastností stránky Okno->Okno (CTRL+w), kde jsou zobrazeny veškeré vizualizační stránky projektu. Vybereme „PAGE“ a změníme její vlastnost Jméno na „Stranka1“.



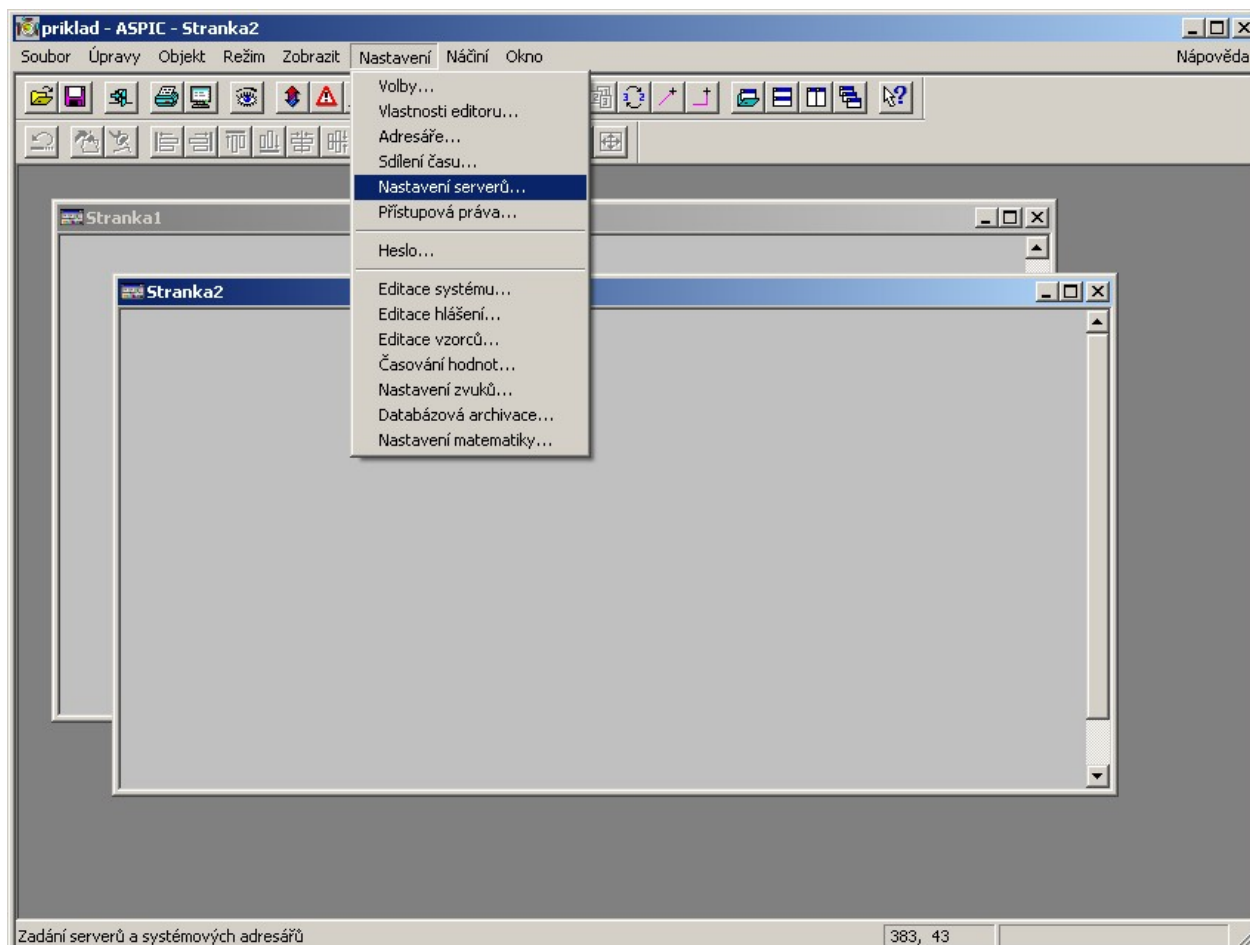
5. Následně, dle zadání našeho příkladu, přidáme druhou vizualizační stránku. Stiskneme tlačítko „Přidej“. Druhou vizualizační stránku pojmenujeme „Stranka2“.



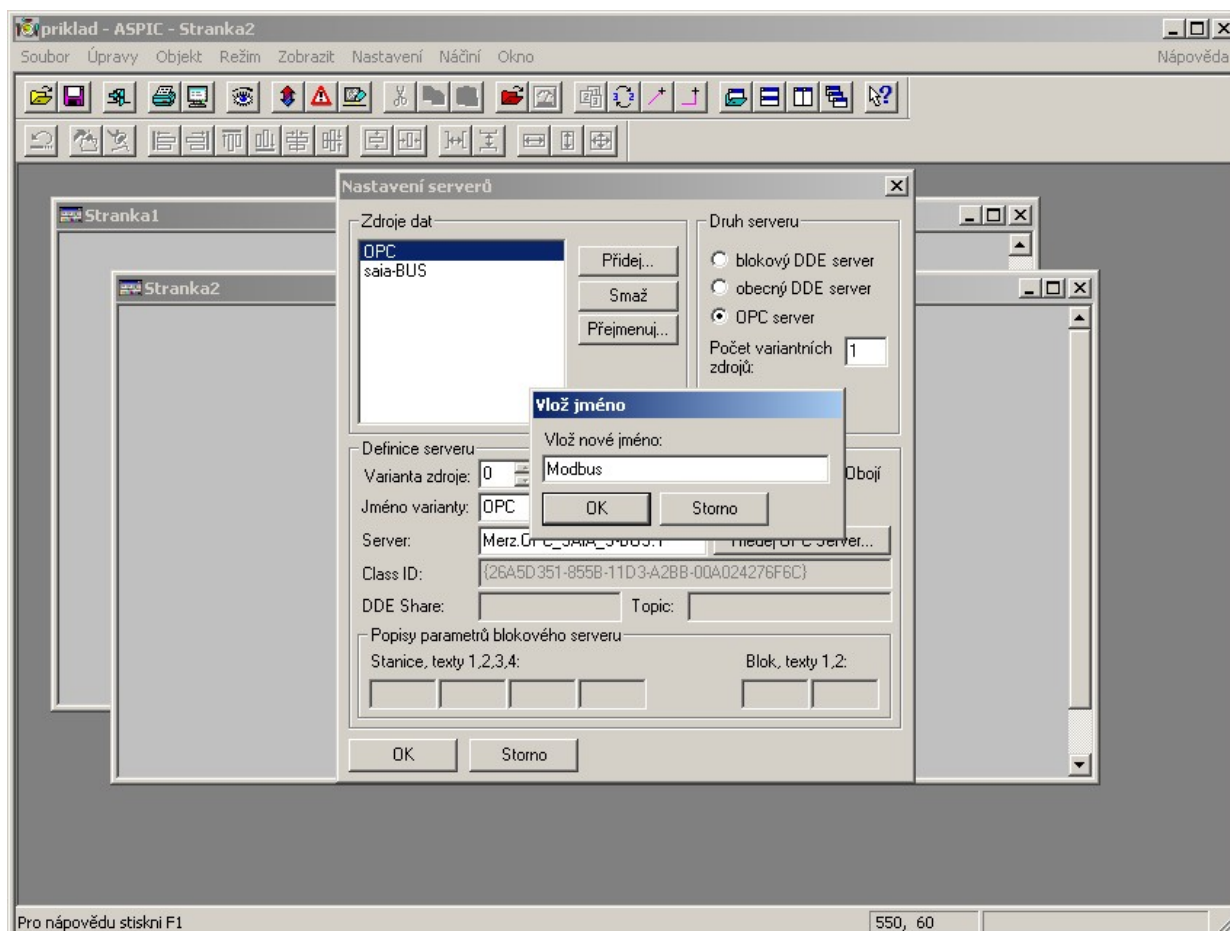
6. Takto vypadá plocha Aspic 3.30 po vytvoření dvou vizualizačních obrazovek



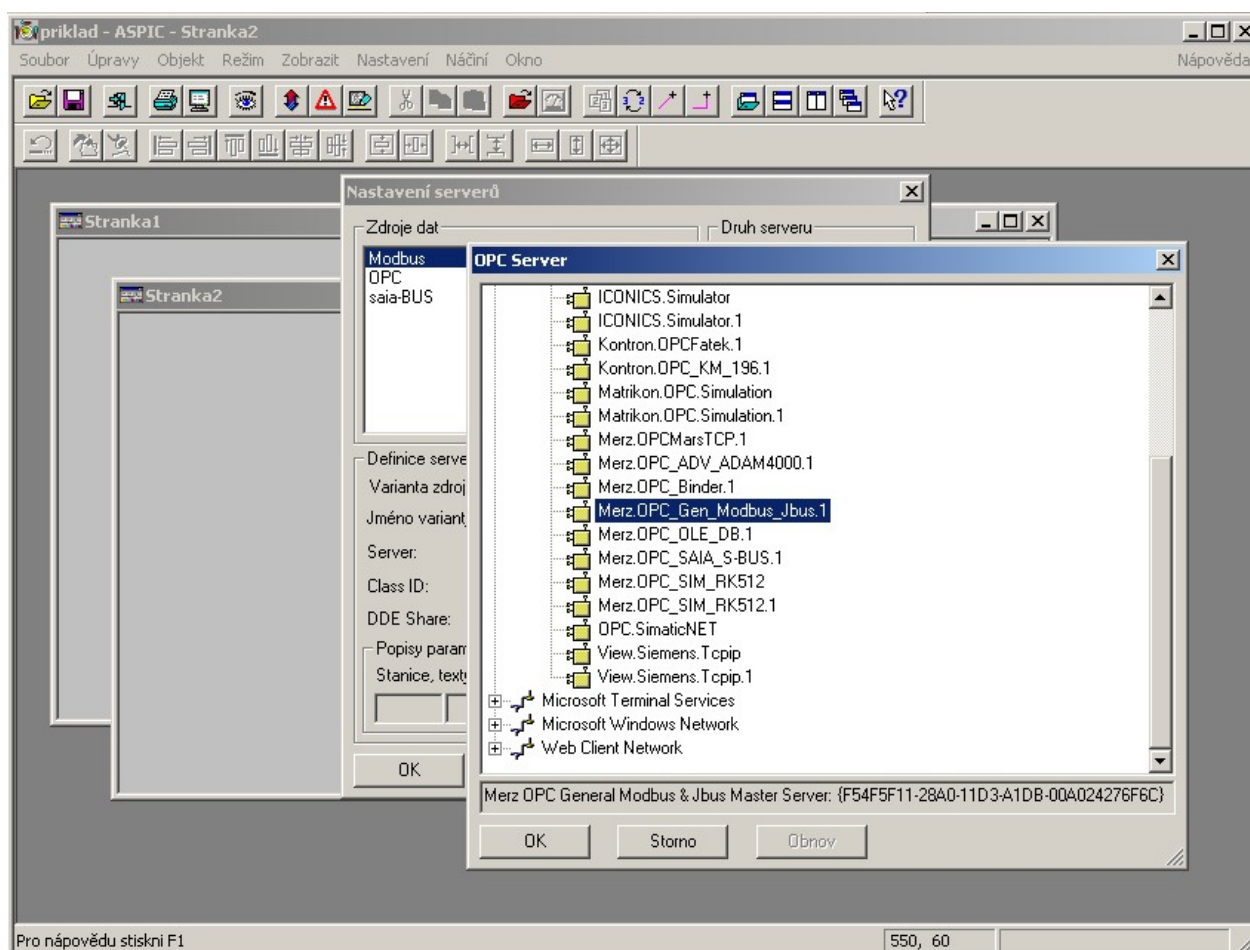
7. Pokračovat budeme nastavením serverů. V Aspic 3.30 je třeba nastavit zdroje dat. Jako zdroje dat lze v Aspic 3.30 použít DDE serverů nebo OPC serverů. V našem příkladu budeme pracovat s OPC serverem Modbus & Jbus v simulačním režimu. Tento server jsme si v předchozí části již nakonfigurovali tak, aby vyhovoval zadání našeho příkladu.



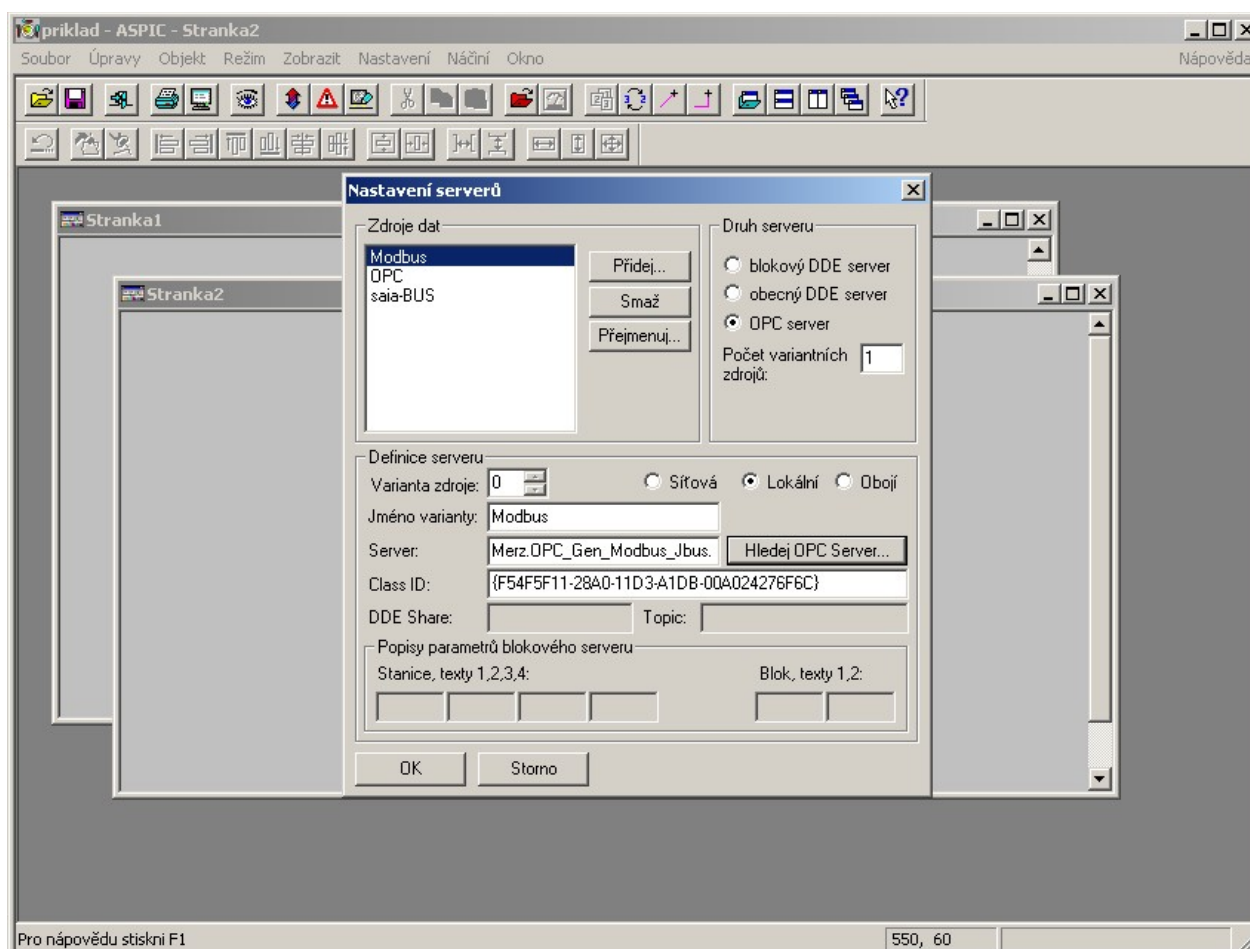
8. Stiskneme tlačítko „Přidej“ a vložíme jméno nového serveru. V našem příkladu použijeme jméno „Modbus“. Jméno je symbolické v rámci Aspic 3.30, můžete použít libovolné jiné.




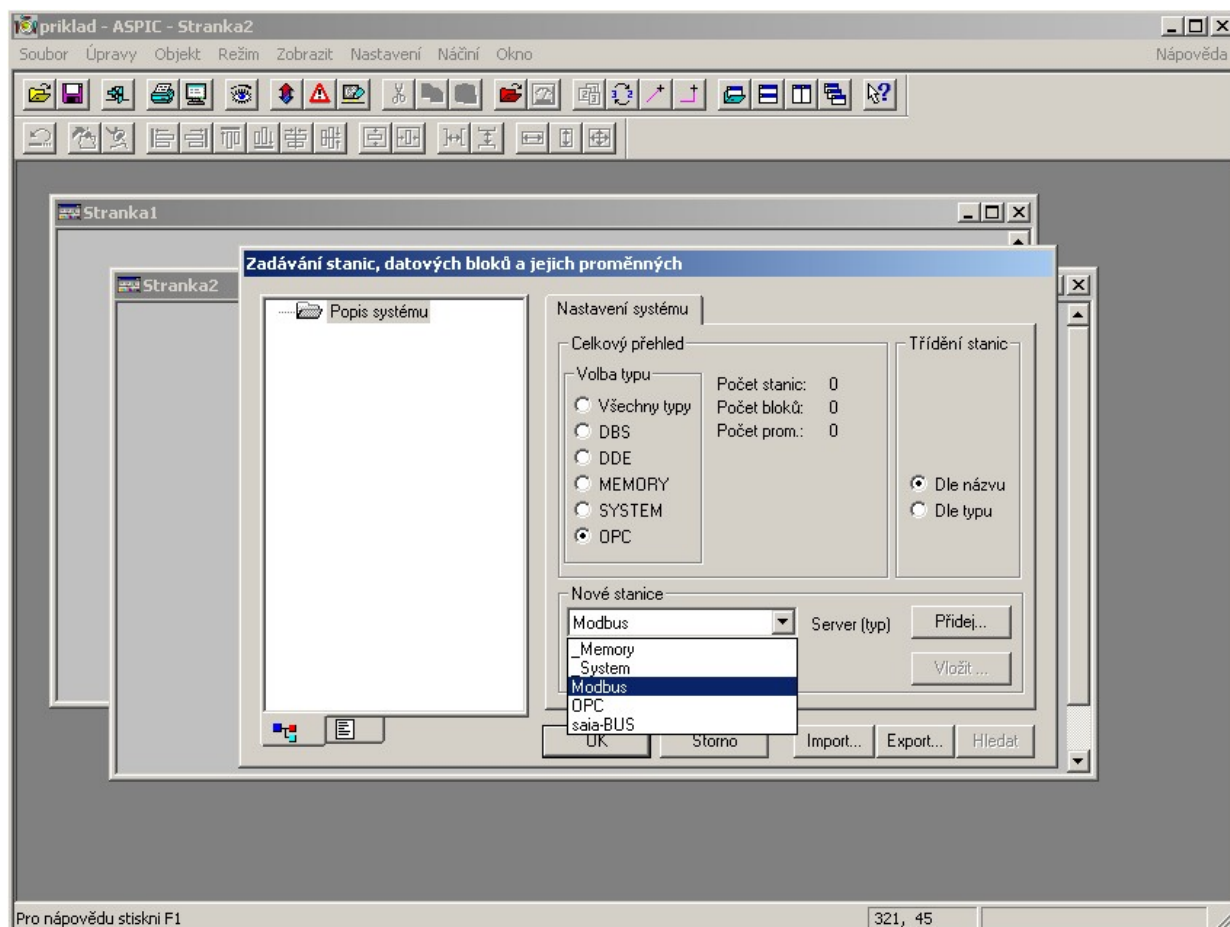
9. Máme vytvořeno symbolické jméno OPC serveru v rámci Aspic 3.30, nyní je třeba nastavit spojení na fyzický OPC server. V dialogu „Nastavení serverů“ vyberte „Zdroje dat“ „Modbus“, tím je řečeno, že další nastavení se budou týkat symbolického jména OPC serveru „Modbus“. Jako „Druh serveru“ nastavte „OPC server“, počet variantních zdrojů ponechte na přednastavené hodnotě 1. Dále v „Definice serveru“ zvolte „Lokální“ variantu. To znamená, že OPC server bude vyhledáván pouze na stejném počítači jako je provozován Aspic 3.30. Ve skutečnosti se OPC server může nacházet i na jiném počítači, pak by v úvahu přicházela volba „Síťová“ nebo „Obojí“. Toto je však nad rámec příkladu. Nyní stiskněte tlačítko „Vyhledat“. Budou zobrazeny veškeré OPC servery registrované na lokálním počítači. Dle specifikace OPC serverů jsou OPC servery zařazeny do skupin OPC 1.0a by 1.0a, OPC 1.0a by 2.0 a OPC 2.0. Rozklikněte skupinu OPC 2.0 a vyberte námi požadovaný OPC server „Merz.OPC_Gen_Modbus_Jbus.1“. Stiskněte „OK“.



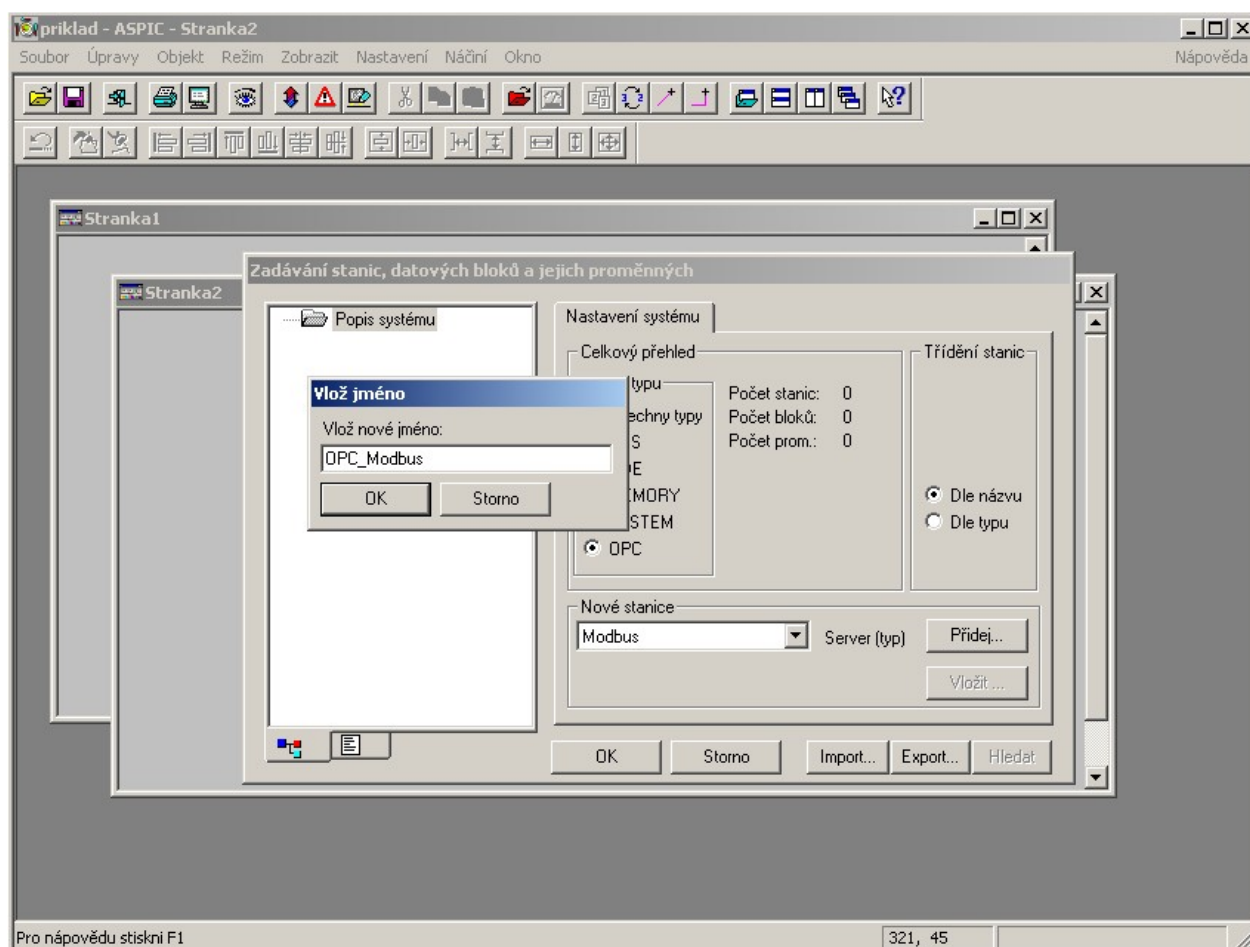
10. Na níže uvedeném obrázku je možné si ověřit, zda navázání OPC serveru na symbolické jméno proběhlo v pořádku. Došlo k automatickému vyplnění položky „Server“ a „Class ID“, které jsou jedinečné pro každý OPC server.



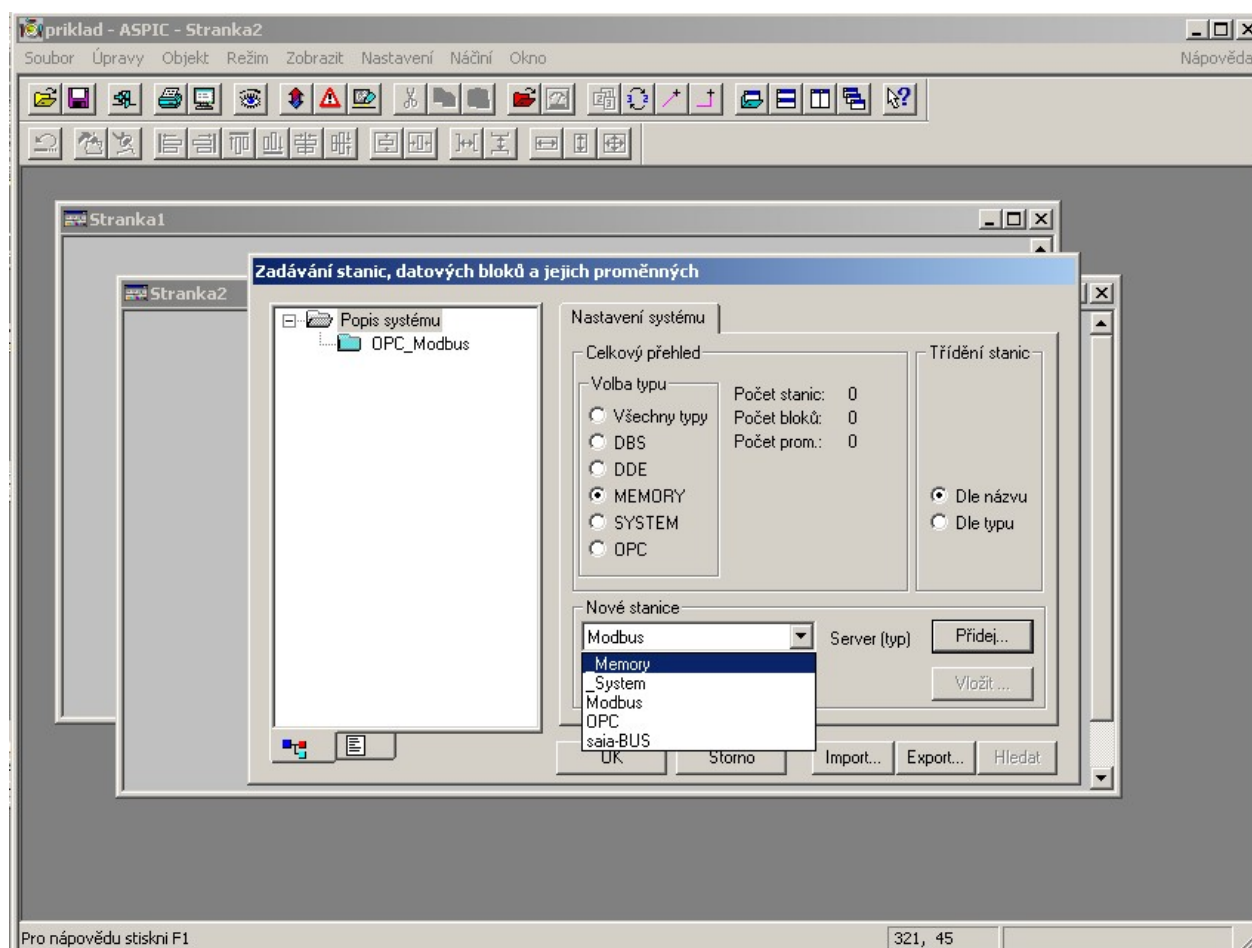
11. Vytvořili jsme spojení na zdroj dat. Nyní je potřeba vytvořit proměnné Aspic 3.30 a navázat data. Stiskněte  v panelu nástrojů. Dialog slouží k zadávání stanic, datových bloků a jejich proměnných. Vyberte novou stanicí typu „Modbus“, jedná se o symbolické jméno OPC serveru vytvořené v předchozích krocích a stiskněte tlačítko „Přidej“.



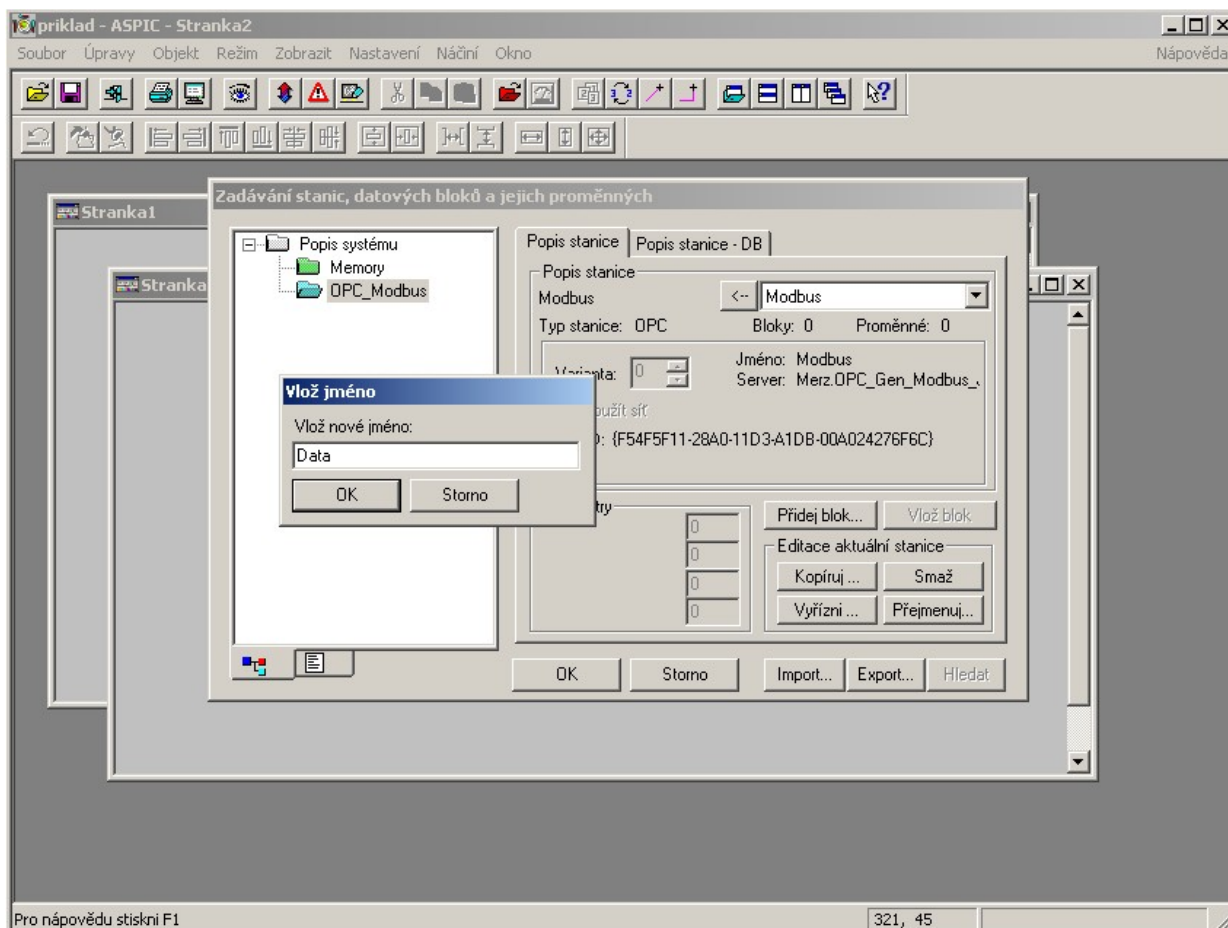
12. Vložte jméno nové stanice. Může být libovolné, názvem nejlépe vystihující zdroj dat. V našem příkladu použijeme jméno stanice „OPC_Modbus“. Do podpoložek této stanice budeme později doplňovat proměnné patřící OPC serveru Modbus & Jbus.



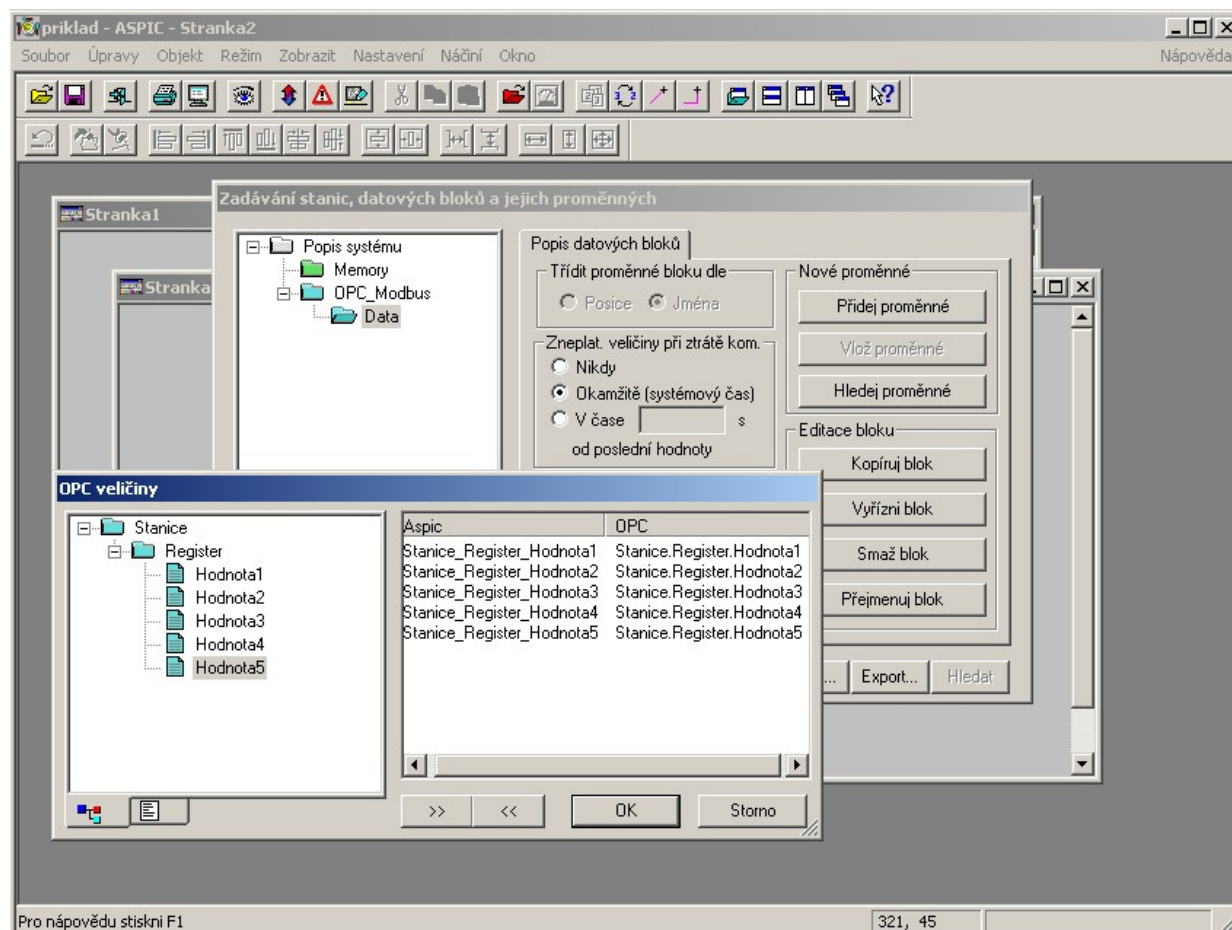
13. V našem příkladě a ve většině „ostrých“ projektů se neobejdeme bez proměnných, které nemají přímou návaznost na zdroje dat (OPC, DDE). Tyto proměnné obsahují data, která jsou nutné při výpočtech skriptovacího jazyka (matematický modul Aspic 3.30). Říkáme jim paměťové (memory) proměnné. Mají stejnou organizační strukturu (rozdělení do stanic, datových bloků) jako proměnné s napojením na ostatní zdroje dat (OPC, DDE). Vytvoříme novou stanici typu „Memory“. Vyberte ze seznamu dostupných stanic „Memory“ a stiskněte tlačítko „Přidej“. V následujícím dialogu vložte jméno nové stanice. Jméno může být libovolné, názvem nejlépe vystihující zdroj dat. V našem příkladu použijeme jméno stanice paměťového typu „Memory“. Do podpoložek této stanice budeme později doplňovat proměnné využívané matematickým modulem



14. Organizační struktura proměnných v Aspic 3.30 je následující. Proměnné jsou součástí datových bloků a datové bloky jsou součástí stanic. Tato organizace slouží pouze k zřehlednění. Na proměnnou se odkazujeme pouze jejím jménem, zařazení do datového bloku či stanice není podstatné. Vytvoříme datový blok, který bude součástí stanice „OPC_Modbus“. Vyberte v levé části dialogu (popis systému) stanici „OPC_Modbus“ a stiskněte tlačítko „Přidej blok“. Pojmenujte jej „Data“.

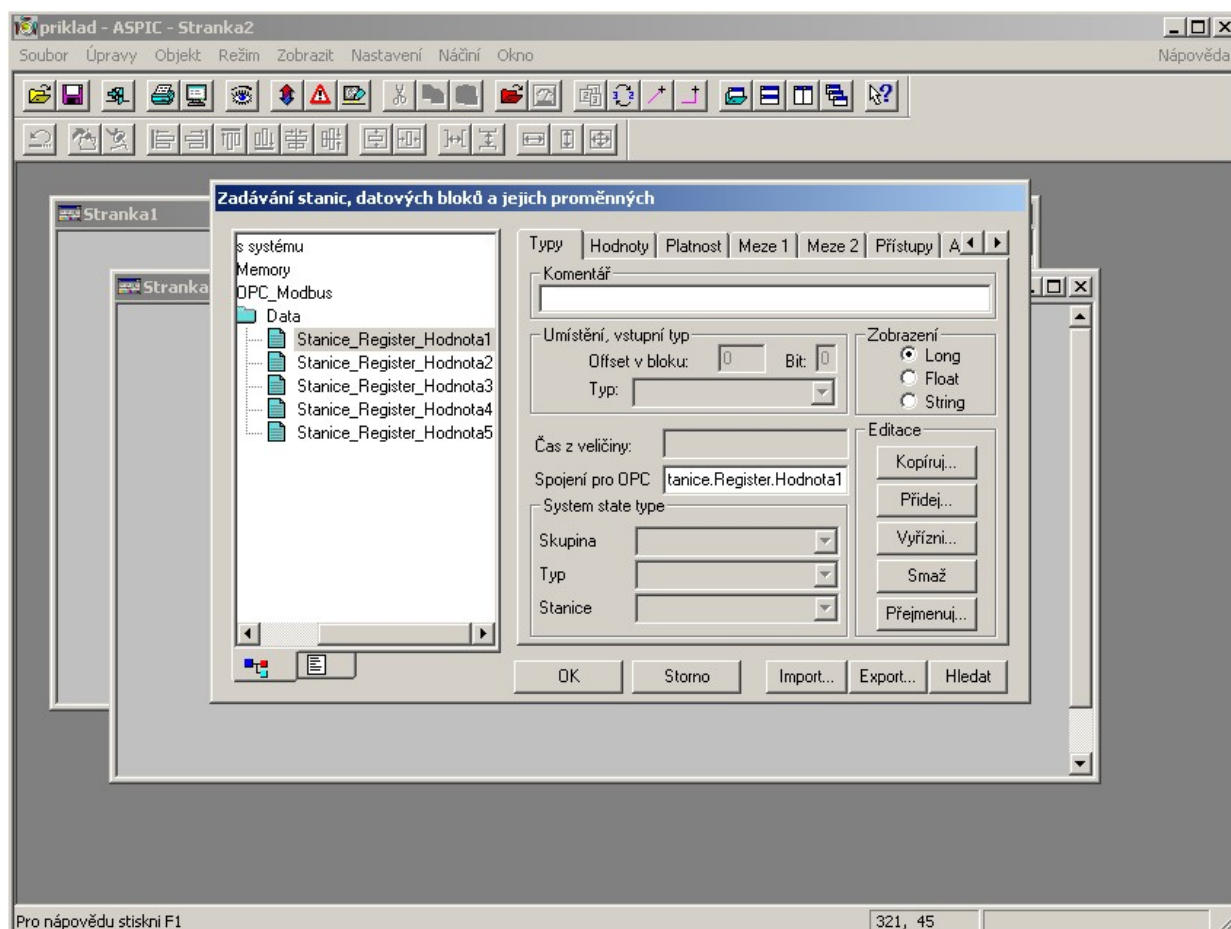


15. Vyberte nově vytvořený datový blok „Data“ a stiskněte tlačítko „Hledej proměnné“. Otevře se dialog, kde v levé části jsou zobrazeny proměnné vytvořené v rámci konfigurace OPC serveru. Vyberte proměnnou z levé části dialogu a stiskem tlačítka „>>“ ji vložte na pravou stranu dialogu. Na pravé straně dialogu jsou zobrazeny proměnné, které chceme přidat do Aspic 3.30. Pokud omylem vyberete proměnnou (máte ji zobrazenou na pravé straně dialogu) použijte „<<“ pro její odstranění ze seznamu přidávaných proměnných do Aspic 3.30. Vzhledem k zadání našeho příkladu budeme chtít přidat všechny dostupné proměnné „Hodnota1“ až „Hodnota5“. Přidání ukončíme stiskem tlačítka „OK“.




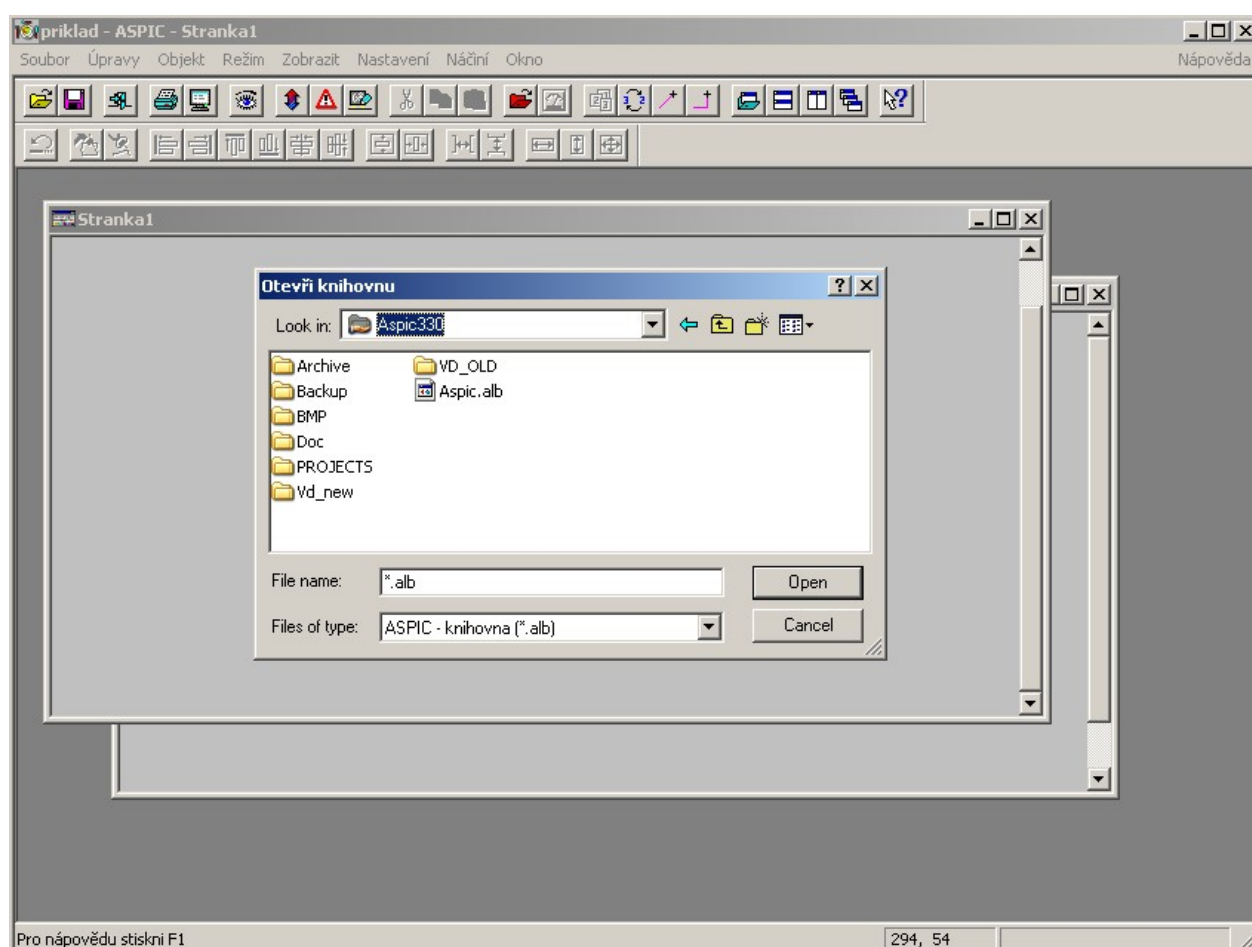
Poznámka: Vkládání proměnných lze provést hromadně označením pomocí „shift“.

16. Stav po vložení proměnných do Aspic 3.30 dokumentuje následující obrázek. Interní názvy proměnných jsou po vložení do Aspic 3.30 nastaveny dle struktury konfigurace OPC serveru. OPC server pracuje se strukturou Stanice.Blok.Proměnná. Aspic 3.30 přejímá názvy stanic, bloků a proměnných a odděluje je tečkami. Vytváří tak interní název proměnné Stanice_Blok_Proměnná dle struktury konfigurace OPC serveru. Tento interní název je možno libovolně měnit. Pokud vyberete již konkrétní proměnnou, např. „Stanice_Register_Hodnota1“, budou v pravé části dialogu zobrazeny volby pro tuto konkrétní proměnnou. Volby jsou umístěny na několika kartách a týkají se např. typů, platností, archivací, mezí, atd... V našem příkladu není nyní nutno měnit jakékoliv vlastnosti, vystačíme si s přednastavenými hodnotami vlastností. Později se k vlastnostem proměnných a jejich nastavení vrátíme (nastavení archivace).



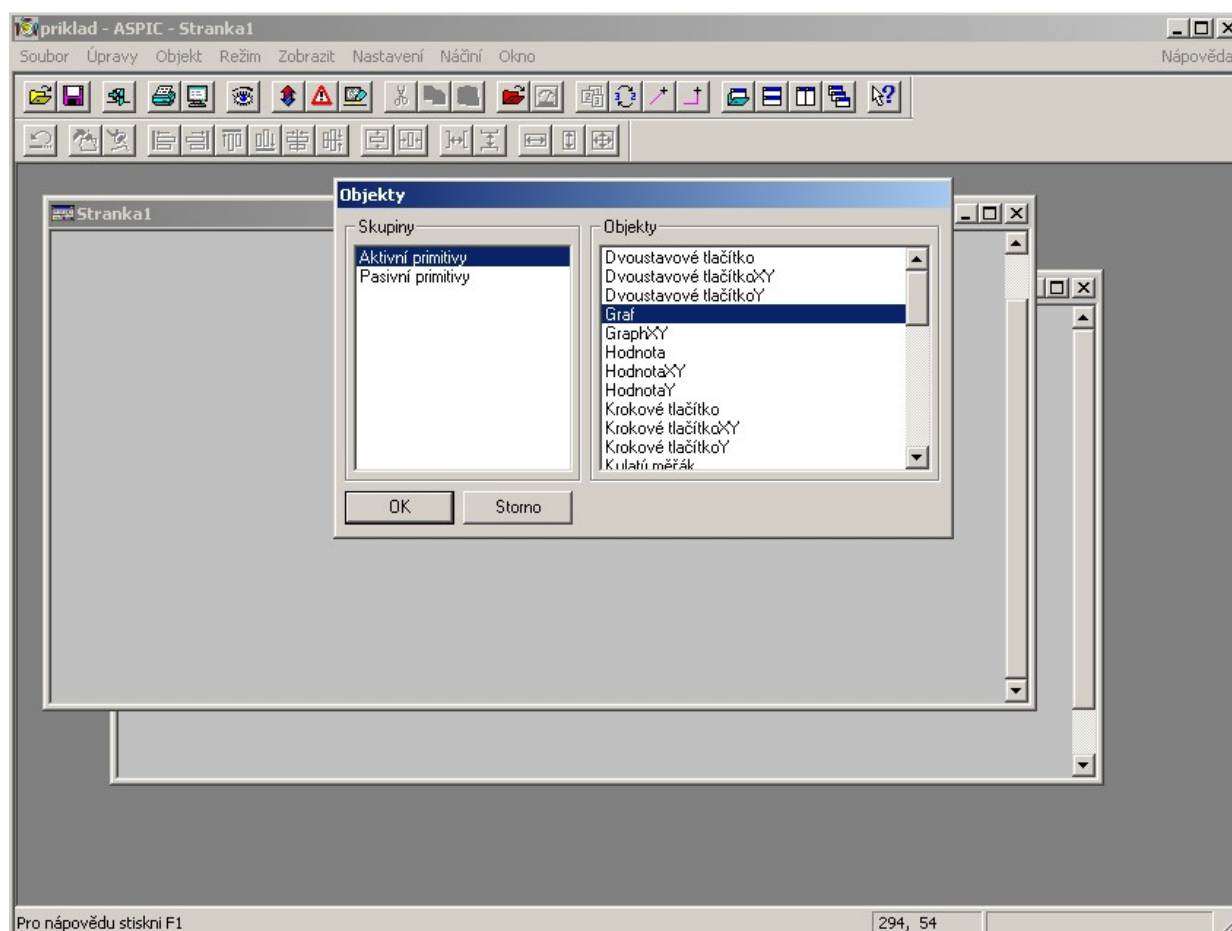
Poznámka: Interní název proměnné je vhodné ponechat ve formátu vygenerovaném z konfigurace OPC serveru. Pokud budeme v Aspic 3.30 vytvářet rozsáhlejší projekt můžeme díky názvu interní proměnné v tomto formátu velice snadno zjistit k jaké stanici (PLC) patří a jakého je typu. Pokud jsou proměnné při tvorbě konfigurace OPC serveru pojmenovány „rozumně“ lze velmi snadno jejich názvy využít v Aspic 3.30.

17. Nyní nastal čas pro vytvoření grafického vzhledu vizualizačních obrazovek. Vizualizační obrazovky používáme v našem příkladu dvě. Jsou vytvořeny a mají jména „Stranka1“ a „Stranka2“. Zatím na nich nejsou umístěny žádné objekty. Objekty které lze umístit na vizualizační obrazovku jsou dvojího typu, aktivní a pasivní. Mezi aktivní zařadíme například „hodnota“, „graf“, „dvoustavové tlačítko“, atd... , jsou to objekty, na které lze navázat proměnné Aspic 3.30 a tyto objekty dle hodnoty navázané proměnné mění svůj stav nebo vzhled. Pasivní objekty, např. „obdelník“, „text“, atd... nejsou navázané na proměnné a během spuštění vizualizace nemění svůj vzhled. Základní pasivní i aktivní prvky jsou obsaženy v knihovně Aspic.alb. Knihovna je dodávána s Aspic 3.30 a je umístěna v adresáři c:\Aspic330\ (pokud jste použili default nastavení při instalaci). Pokud chcete začít pracovat s knihovnou objektů, je třeba ji nejdříve otevřít. Knihovnu otevřeme klepnutím na  v panelu nástrojů. Vyberte knihovnu objektů Aspic.alb (na default cestě c:\Aspic330\Aspic.alb) a otevřete ji stiskem tlačítka „Open“.






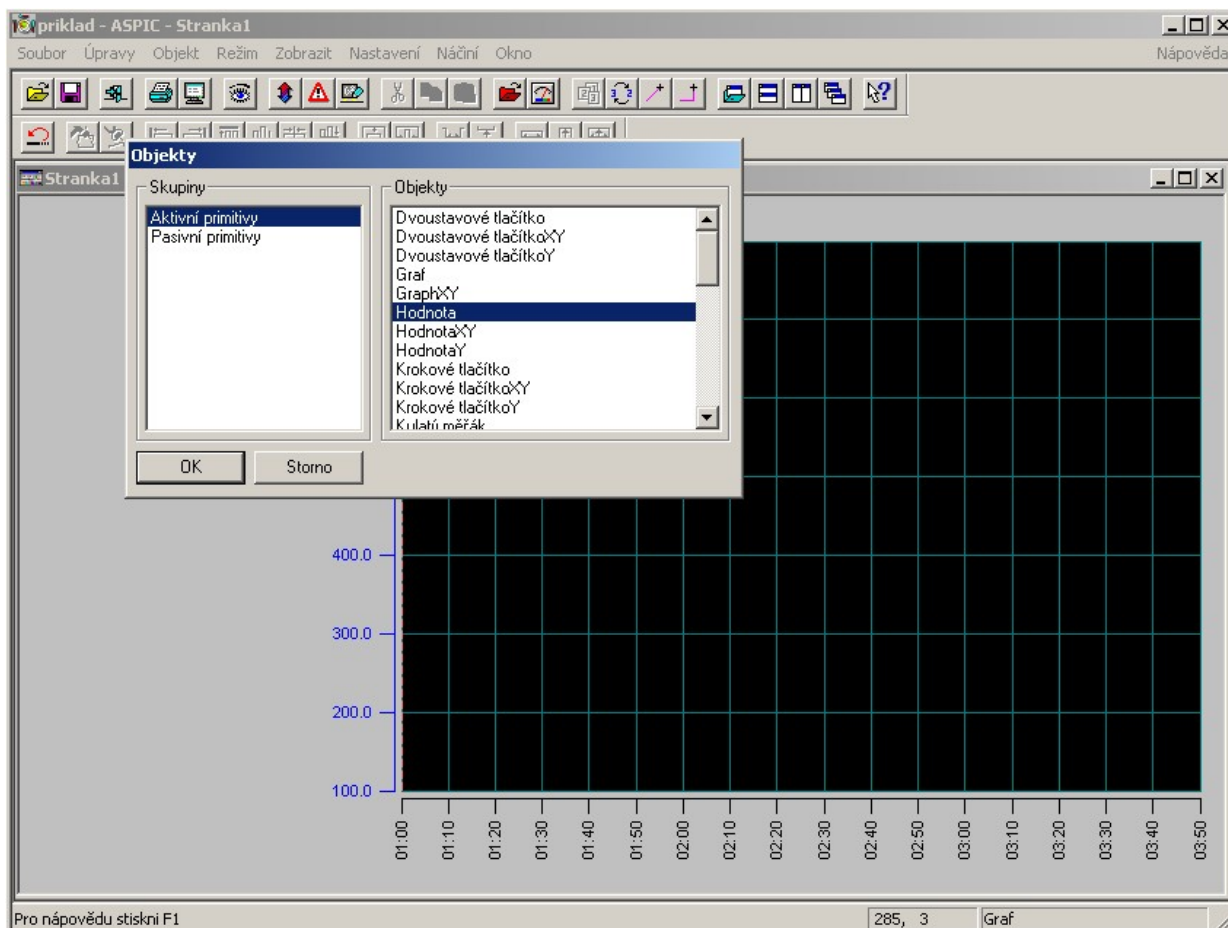
Poznámka: Aspic 3.30 umožňuje uživateli vytvářet knihovny vlastních objektů skládáním primitivních objektů. Vytváření uživatelských knihoven je nad rámec tohoto příkladu.

18. Po otevření knihovny objektů je zobrazen její obsah. V levé části je výběr aktivních a pasivních skupin objektů. V pravé části je uveden seznam dostupných objektů. Vyberte skupinu „Aktivní primitivy“ a objekt „Graf“. Objekt umístíte na obrazovku (Stránka1) stisknutím tlačítka „OK“.

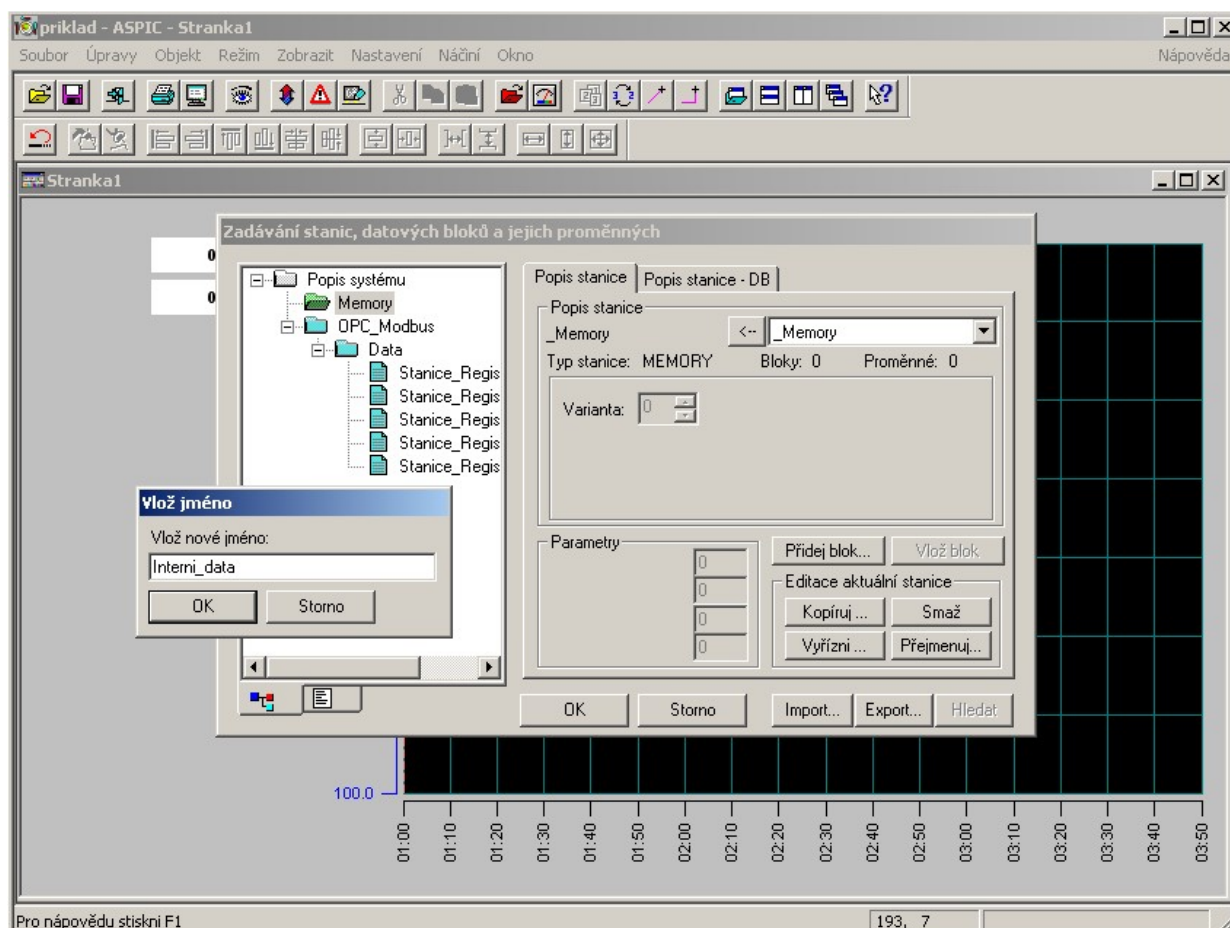


Poznámka: Vybraný objekt z knihovny je umístěn na vizualizační obrazovku, která byla aktivní před otevřením knihovny objektů.

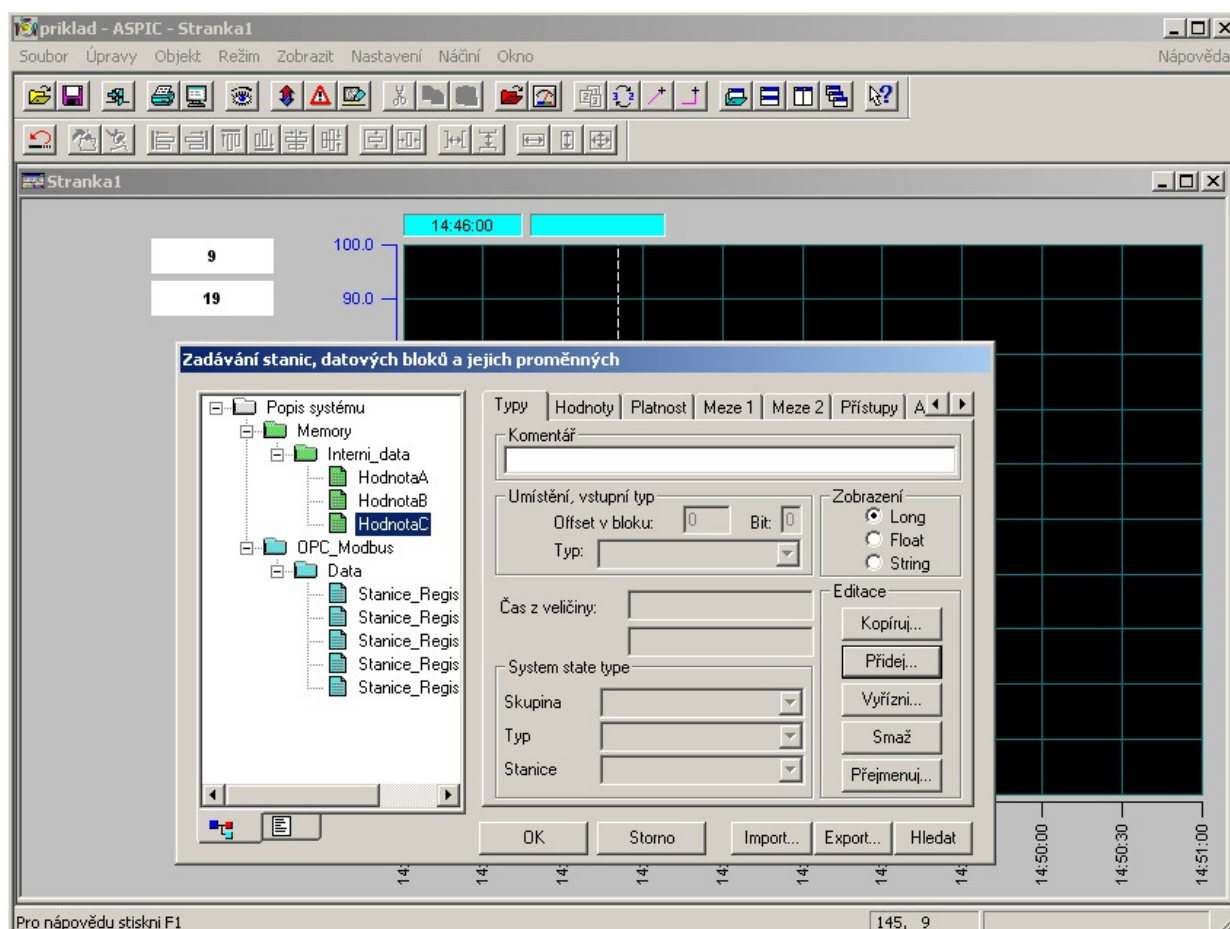
19. Dle potřeby přemístěte a roztáhněte objekt graf po ploše vizualizační obrazovky. Na panelu nástrojů, napravo od ikony pro otevření knihovny objektů  je zobrazena ikona  , která symbolizuje, že je již knihovna objektů otevřena. Stiskem ikony  je otevřen dialog s dostupnými objekty knihovny (stejný dialog jako v minulém bodu). Z nabídky objektů vyberte objekt „Hodnota“ a umístěte jej na vizualizační obrazovku „Stranka1“ stiskem tlačítka „OK“. Objekty „Hodnota“ umístěte na vizualizační obrazovku „Stranka1“ celkem dva.



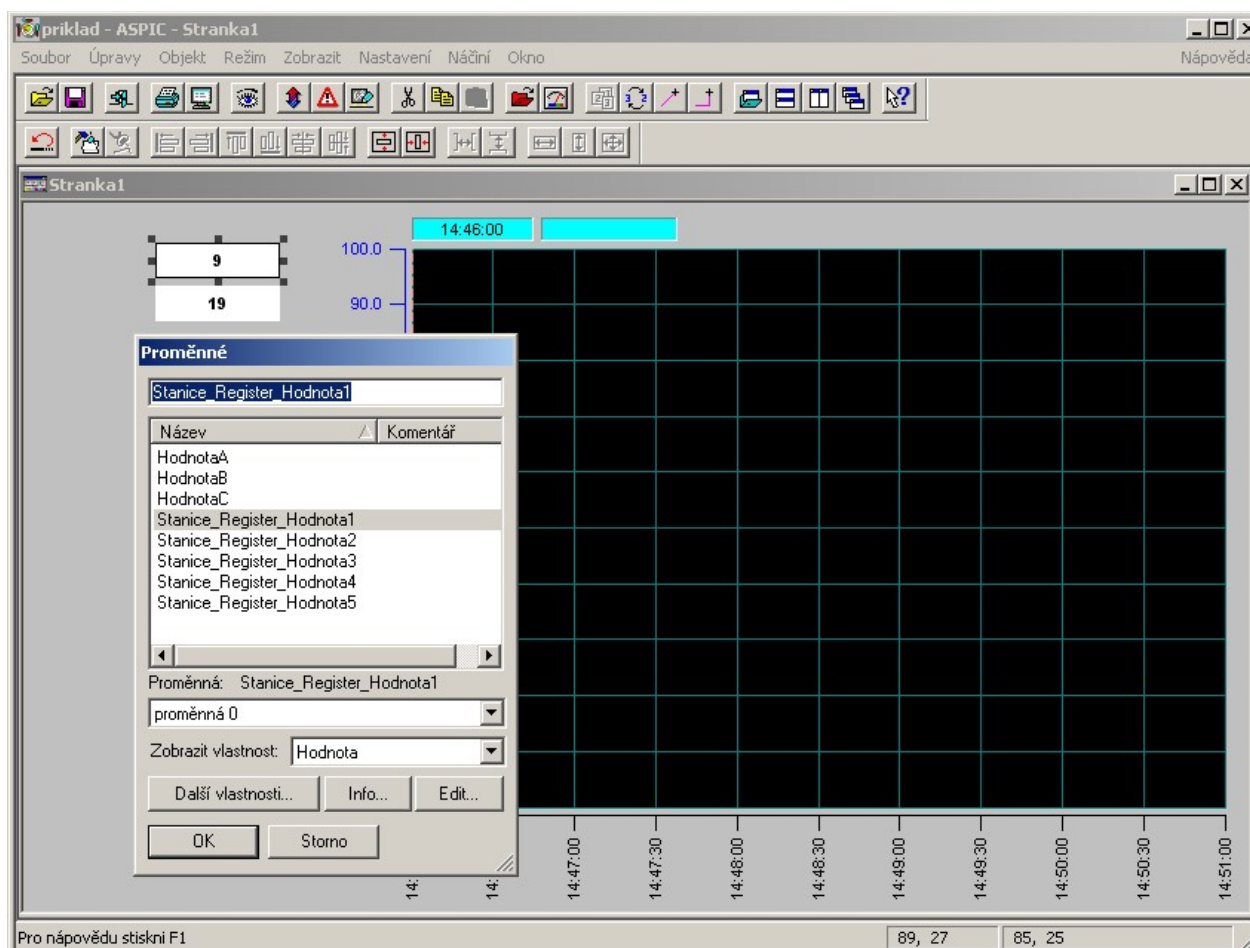
20. Nyní máme na vizualizační obrazovce „Stranka1“ umístěny tři aktivní objekty, jeden objekt „graf“ a dva objekty „hodnota“. Dle zadání našeho příkladu bude graf zobrazovat průběhy tří proměnných (Hodnota1, Hodnota2 a HodnotaA). Poslední jmenovaná je proměnná paměťového typu. V předchozích krocích jsme vytvořili stanici „Memory“, ta však neobsahuje žádné datové bloky ani proměnné. Vytvoříme tedy v rámci stanice „Memory“ datový blok jménem „Interni_data“.




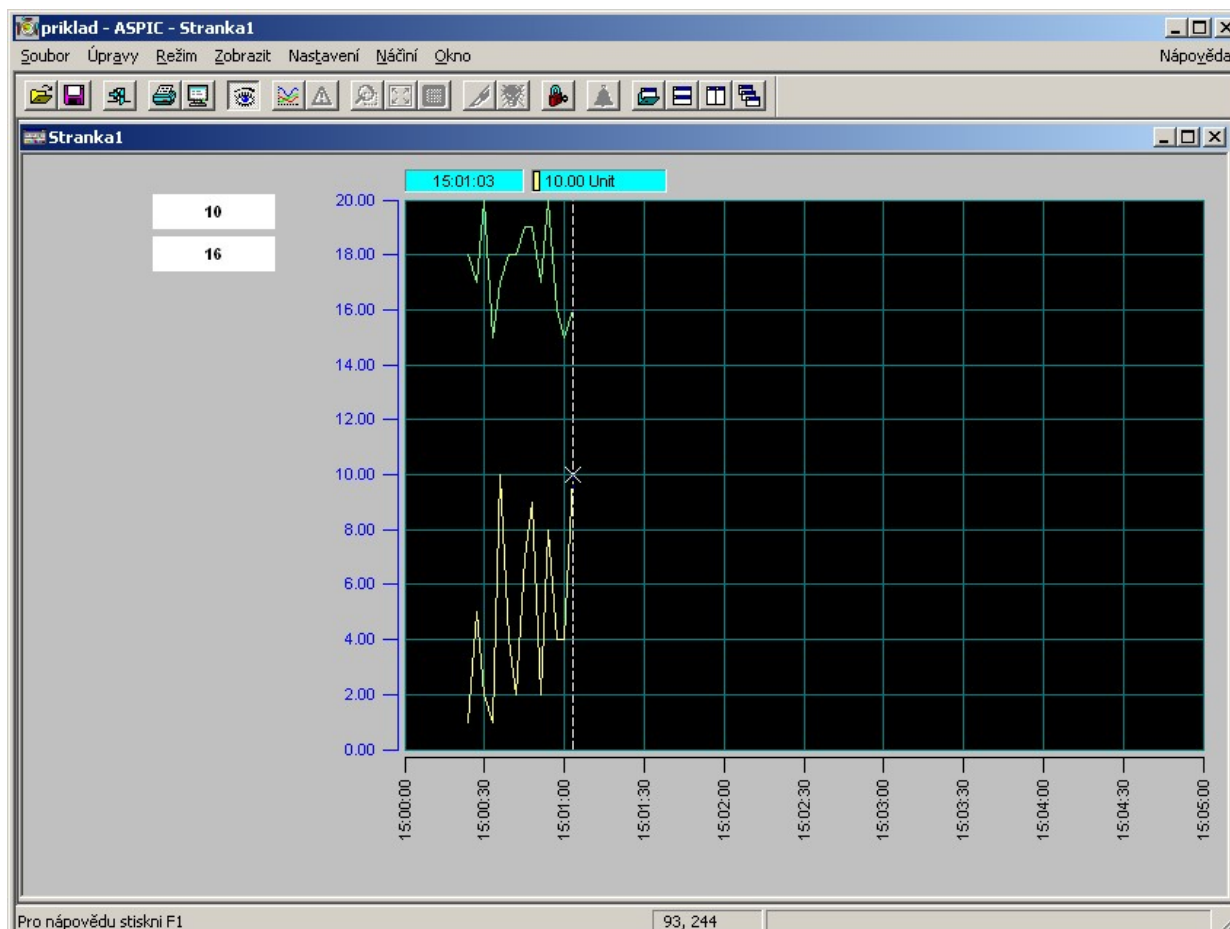
21. V rámci datového bloku „Interni_data“ vytvořte tři proměnné s názvy „HodnotaA“ až „HodnotaC“. Jelikož plánujeme využít paměťovou proměnnou „HodnotaA“ pro zobrazení sinusového průběhu, je potřeba nastavit pro tuto proměnnou vlastnost „Zobrazení“ na „Float“ ! Vlastnosti ostatních paměťových proměnných ponecháme zatím v default nastavení.



23. Obdobně jako při napojování proměnných na objekt „graf“ provedeme napojení na dva objekty „hodnota“ umístěné na vizualizační obrazovce „Stranka1“. Klikněte pravým tlačítkem myši na první z objektů hodnota a v otevřeném dialogu „Proměnné“ Proveďte navázání na interní proměnnou Aspic 3.30 s názvem „Stanice_Register_Hodnota1“. Dialog ukončete stiskem tlačítka „OK“. Obdobné nastavení proveďte i pro druhý objekt „hodnota“ ovšem proveďte navázání na proměnnou „Stanice_Register_Hodnota2“.

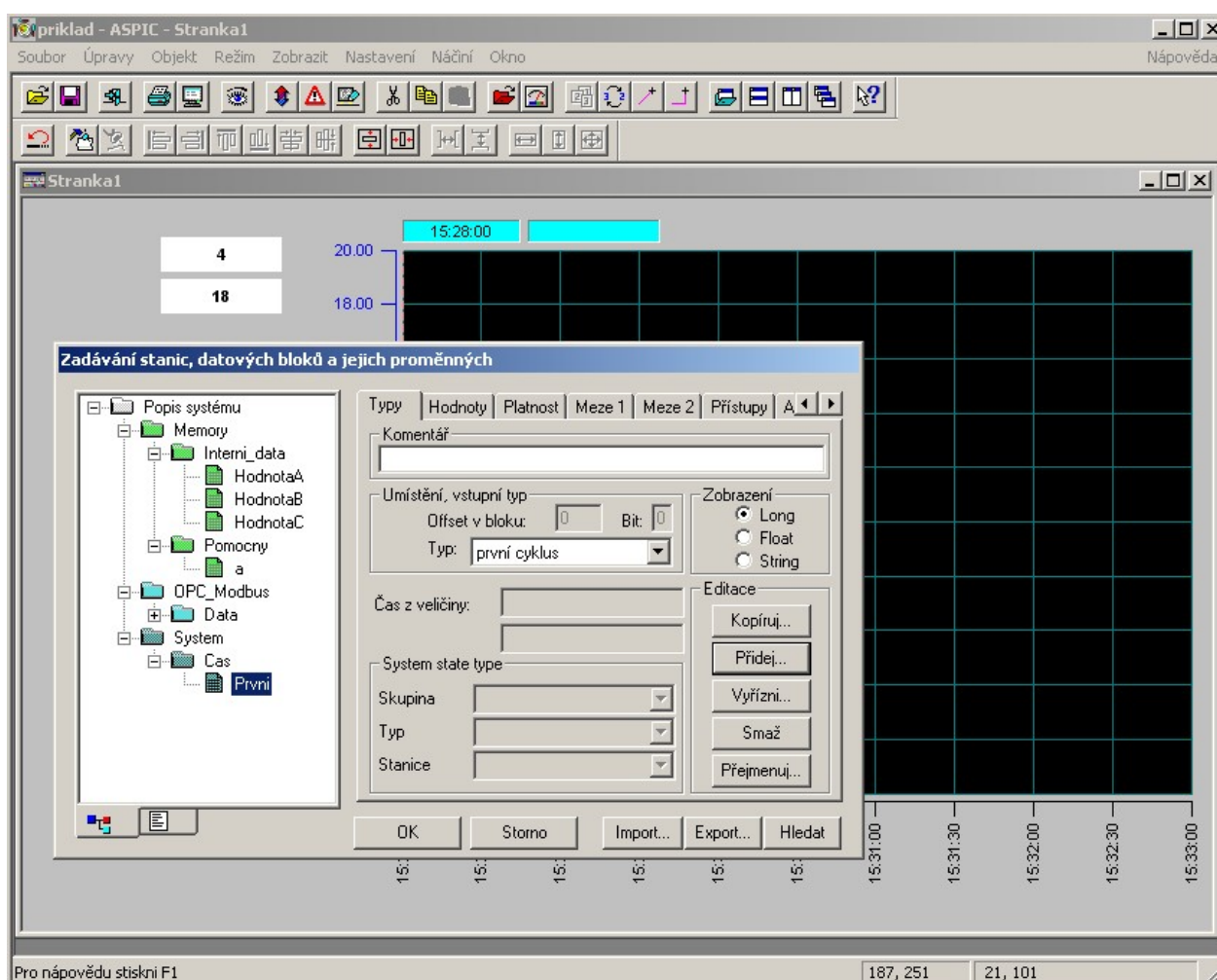



24. Nyní je projekt připraven na první spuštění. Zdaleka ještě neodpovídá našemu zadání, ovšem je možno jej spustit a pokud jste všechna nastavení provedli správně bude na vizualizační obrazovce zobrazen „objekt“ graf s průběhem proměnných Stanice_Register_Hodnota1 a Stanice_Register_Hodnota2 (další dva průběhy nejsou zatím viditelné). Dále pak dva objekty „hodnota“ se zobrazením proměnných Stanice_Register_Hodnota1 a Stanice_Register_Hodnota2. Vizualizaci spustíte stiskem ikony  nebo (CTRL+I) v panelu nástrojů. Vypnete ji stejným způsobem.



Poznámka: Pokud se hodnoty v objektech „hodnota“ mění velmi rychle (rychleji než jednou za vteřinu), ponechali jste default nastavení vlastnosti „Resp. time“ v konfiguraci OPC serveru. Vraťte se k bodu 7 tohoto příkladu a změňte nastavení na 3000 ms. Toto nastavení v našem příkladu je z důvodu přehlednosti.

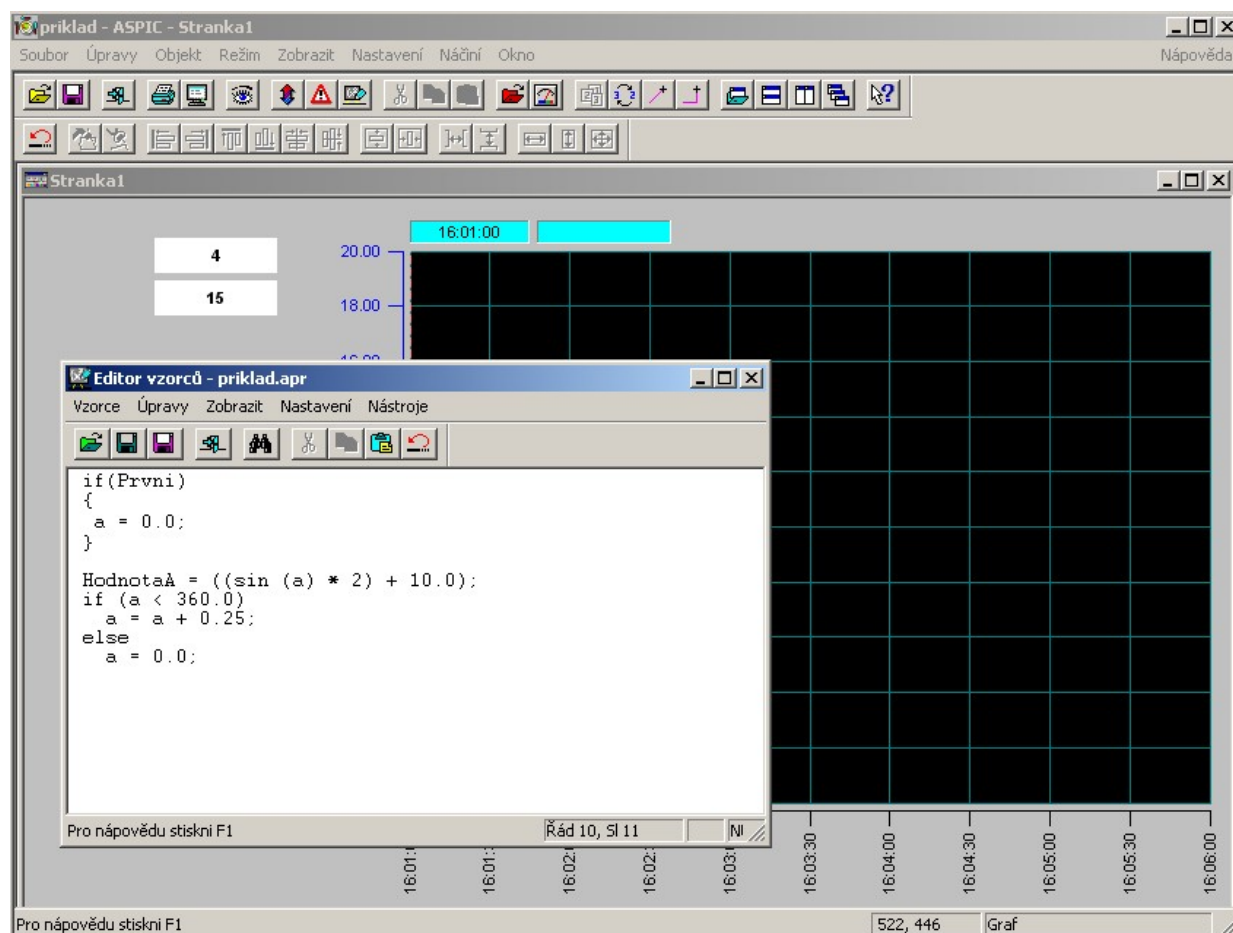
25. Zdokonalujeme projekt. Otevřete dialog pro zadávání stanic, datových bloků a proměnných (viz. bod 11. tohoto příkladu) a přidejte stanici typu „System“. Pojmenujte ji „Systém“ a vytvořte v ní datový blok s názvem „Cas“. Uvnitř datového bloku „Cas“ vytvořte proměnnou s názvem „První“, jejíž typ nastavte na „první cyklus“. K čemu nám tato proměnná bude? Aspic 3.30 nastaví hodnotu této proměnné na (true) pouze v okamžiku, kdy provádí po spuštění vizualizace první průchod programem skriptovacího modulu. Program skriptovacího modulu je prováděn periodicky (perioda může být nastavena, ovšem v našem příkladu jsme ponechali default nastavení, které vykonává program „jak nejčastěji to jde“). Téměř vždy potřebujeme při spuštění vizualizace provést inicializaci hodnot paměťových proměnných. Díky proměnné „První“ jsme schopni detekovat spuštění vizualizace (první provedení, cyklus, programu skriptovacího modulu) a provést potřebné inicializace. Dále přidejte do již existující stanice „Memory“ datový blok s názvem „Pomocny“. Budeme ho používat pro pomocné paměťové proměnné, které potřebujeme při psaní programu skriptovacího modulu. Uvnitř datového bloku „Pomocny“ vytvořte novou proměnnou s názvem „a“ a nastavte její vlastnost „Zobrazení“ na hodnotu „Float“. Proč? Uvidíte dále.



26. Zajistíme, aby paměťová proměnná „HodnotaA“ byla plněna „sinusovým“ průběhem. Plnění zajistíme na úrovni matematického modulu. Spustíte „Editor vzorců“ stiskem  a vložte následující kód.

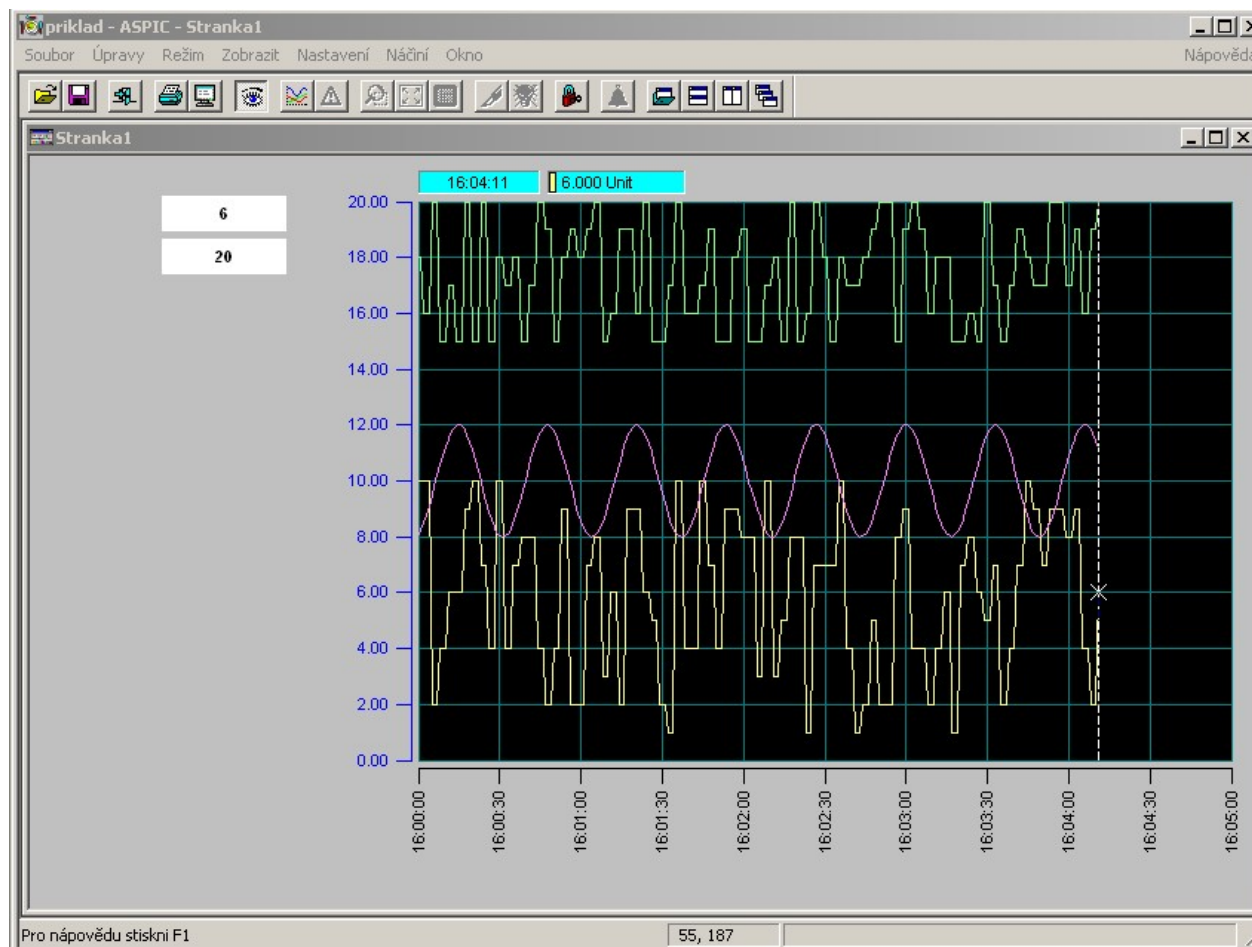
```
if (Prvni)
{
    a = 0.0;
}
HodnotaA = ((sin (a) * 2) + 10.0);
if (a < 360.0)
    a = a + 0.25;
else
    a = 0.0;
```

Při prvním průchodu programu bude inicializována proměnná „a“. Při každém průběhu programu bude vypočítán sinus hodnoty proměnné „a“ a přiřazen paměťové proměnné „HodnotaA“. Sinus je pro lepší viditelnost v grafu násoben dvěma a celá sinusovka je posunuta o 10. Proměnná „a“ je každým zpracováním programu inkrementována o 0.25, v případě dosažení hodnoty 360.0 je nulována. Kód uložte a opusťte „Editor vzorců“.

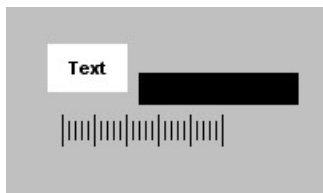


Poznámka: Jelikož není pevně nastavena perioda provádění programu matematického modulu, bude tvar výsledného průběhu odlišný v závislosti na výkonu počítače.

27. Spustíte vizualizace stiskem ikony  . V grafu bude vykreslena sinusovka.



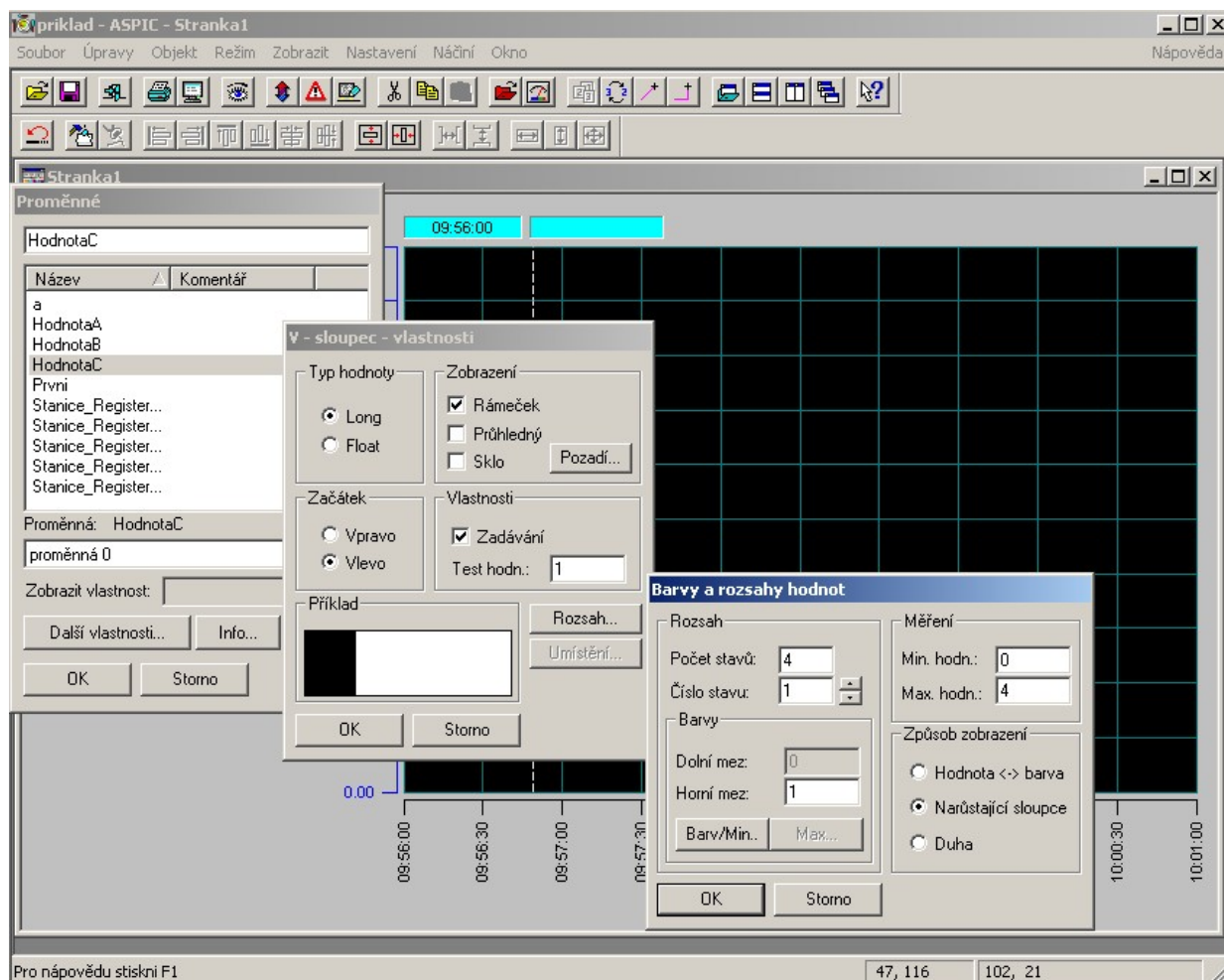
28. Zdokonalíme projekt. Umístěte na vizualizační obrazovku „Stranka1“ aktivní objekt „V-sloupec“, dále jeden pasivní objekt „Vodorovná stupnice“ a jeden pasivní objekty „Text“. Celkem potřebujeme objekty typu „Text“ tři, avšak z knihovny umístíme na obrazovku pouze jeden a po nastavení jeho vlastností jej dvakrát zkopírujeme. Objekty umístěné na vizualizační obrazovce ilustruje následující obrázek.



Nyní nastavte vlastnosti aktivního objektu „V-sloupec“. Od tohoto objektu požadujeme aby zobrazoval a umožňoval nastavení paměťové proměnné „HodnotaC“. Hodnota proměnné může nabývat celých čísel v rozsahu 0 až 4. Umožněte „Zadávání“ hodnot a změňte barvu „Pozadí“ objektu „V-sloupec“ na bílou. Dále stiskněte tlačítko „Rozsah“ a nastavte „Počet stavů“ na 4, „Min. hodn.“ na 0, „Max. hodn.“ na 4 a „Způsob zobrazení“ na „Narůstající sloupce“.

Ke každému číslu stavu, je možno zvolit dolní a horní mez a barvu. Pro stav „1“ nastavte dolní mez na 0, horní mez na 1 a barvu černou. Pro stav „2“ nastavte dolní mez na 1, horní mez na 2 a barvu černou. Pro ostatní stavy nastavte meze obdobně jako pro předchozí stavy, vzrůstající až do 4, s barvou vždy černou.

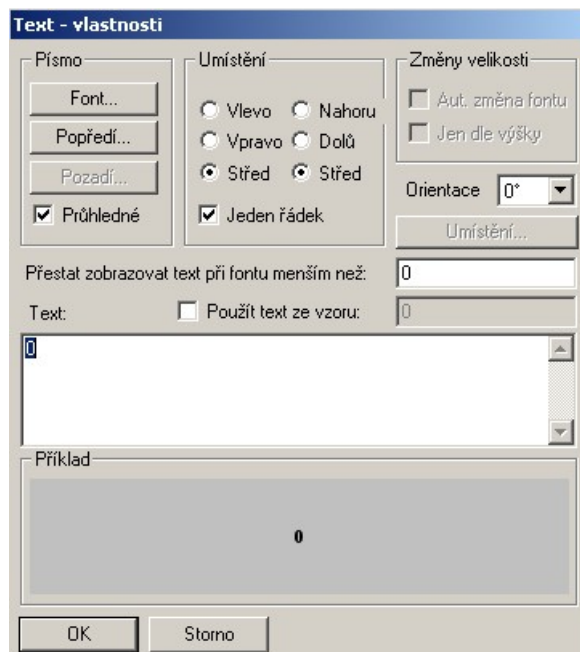
Proved'te napojení objektu „V-sloupec“ na proměnnou „HodnotaC“.



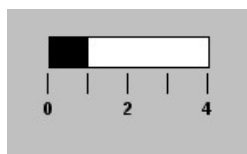
Nastavte vlastnosti pasivního objektu „Vodorovná stupnice“ dle níže uvedeného obrázku.



Nastavte vlastnosti objektu „Text“ dle níže uvedeného obrázku. Tímto objektem chceme vytvořit čísla pro „Vodorovnou stupnici“. Zvolte průhledné pozadí a do pole „Text“ vložte 0. Stisknete tlačítko OK. Objekt dvakrát zkopírujte pomocí (CTRL + C, CTRL + V) a u kopií změňte zobrazovaný text na 2 a 4.

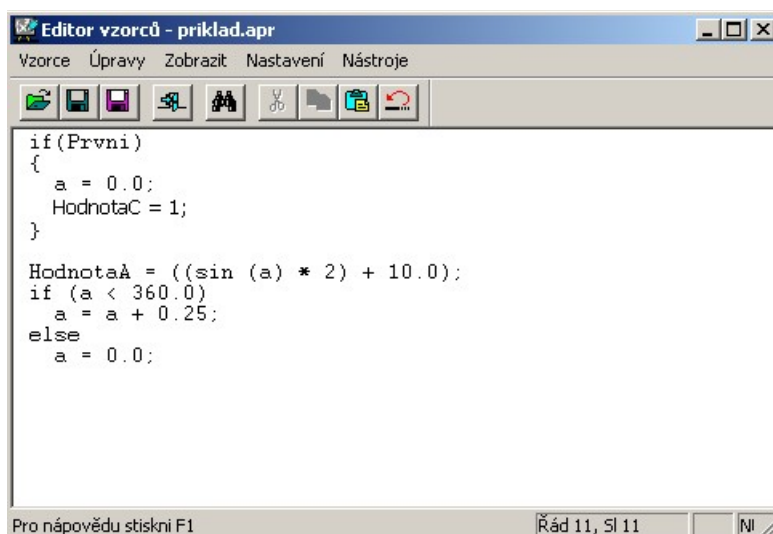


Nyní změňte pozici a velikost objektů, přibližně podle uvedeného obrázku. Aspic 3.30 umožňuje prohlásit tuto skupinu objektů za nový uživatelský objekt a provést jeho uložení do uživatelské knihovny a následně jej velmi jednoduše používat. Uživatelské knihovny jsou nad rámec tohoto příkladu. Hledejte prosím v uživatelské příručce k Aspic 3.30.



Nezapomeňte provést inicializaci paměťové proměnné „HodnotaC“. Do kódu matematického modulu, do sekce „První cyklus“ přidejte následující řádek. Inicializace proměnné na hodnotu 1.

```
HodnotaC = 1;
```



```

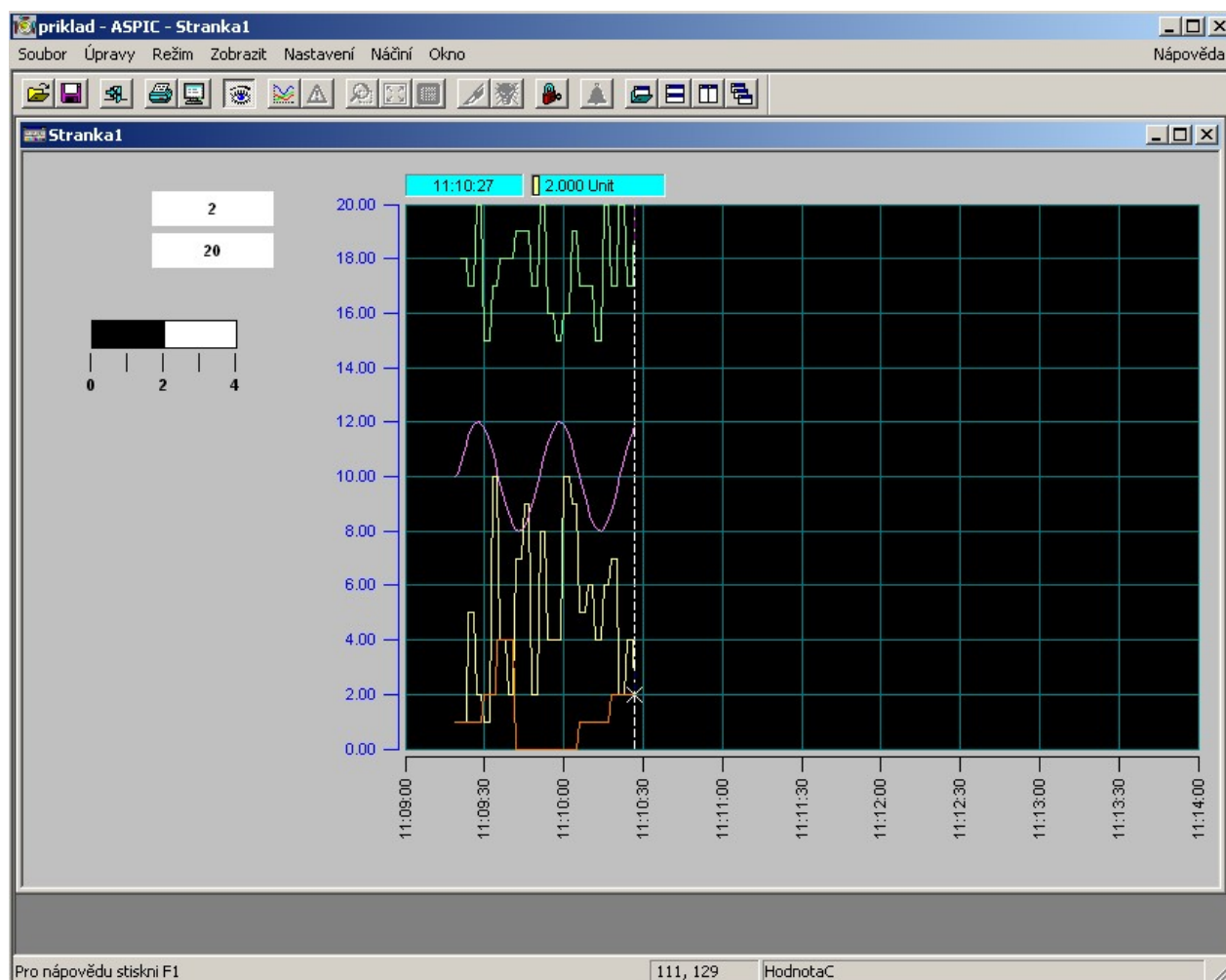
if(Prvni)
{
  a = 0.0;
  HodnotaC = 1;
}

HodnotaA = ((sin (a) * 2) + 10.0);
if (a < 360.0)
  a = a + 0.25;
else
  a = 0.0;
  
```

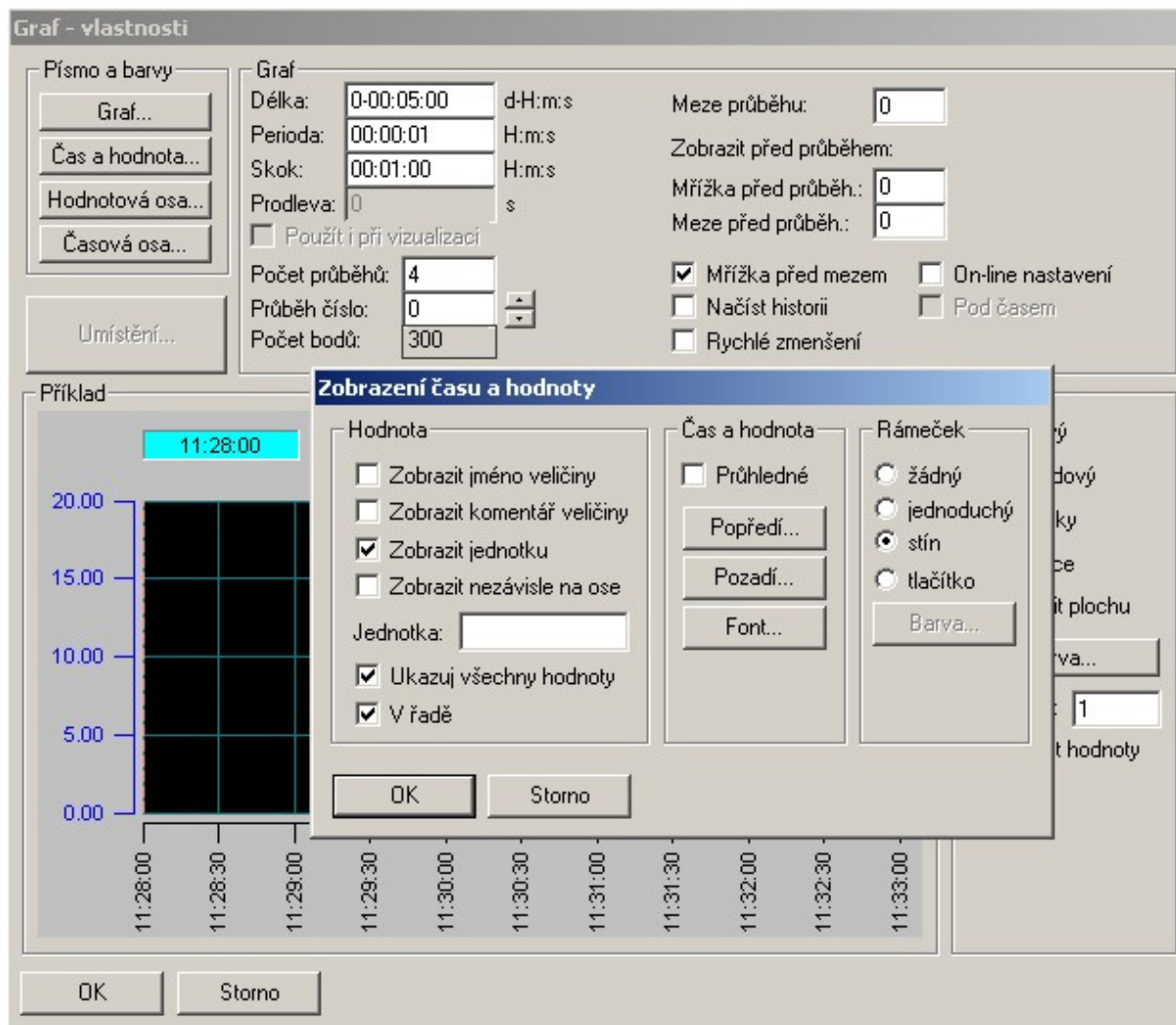
Pro nápovědu stiskni F1

Řád 11, Sl 11

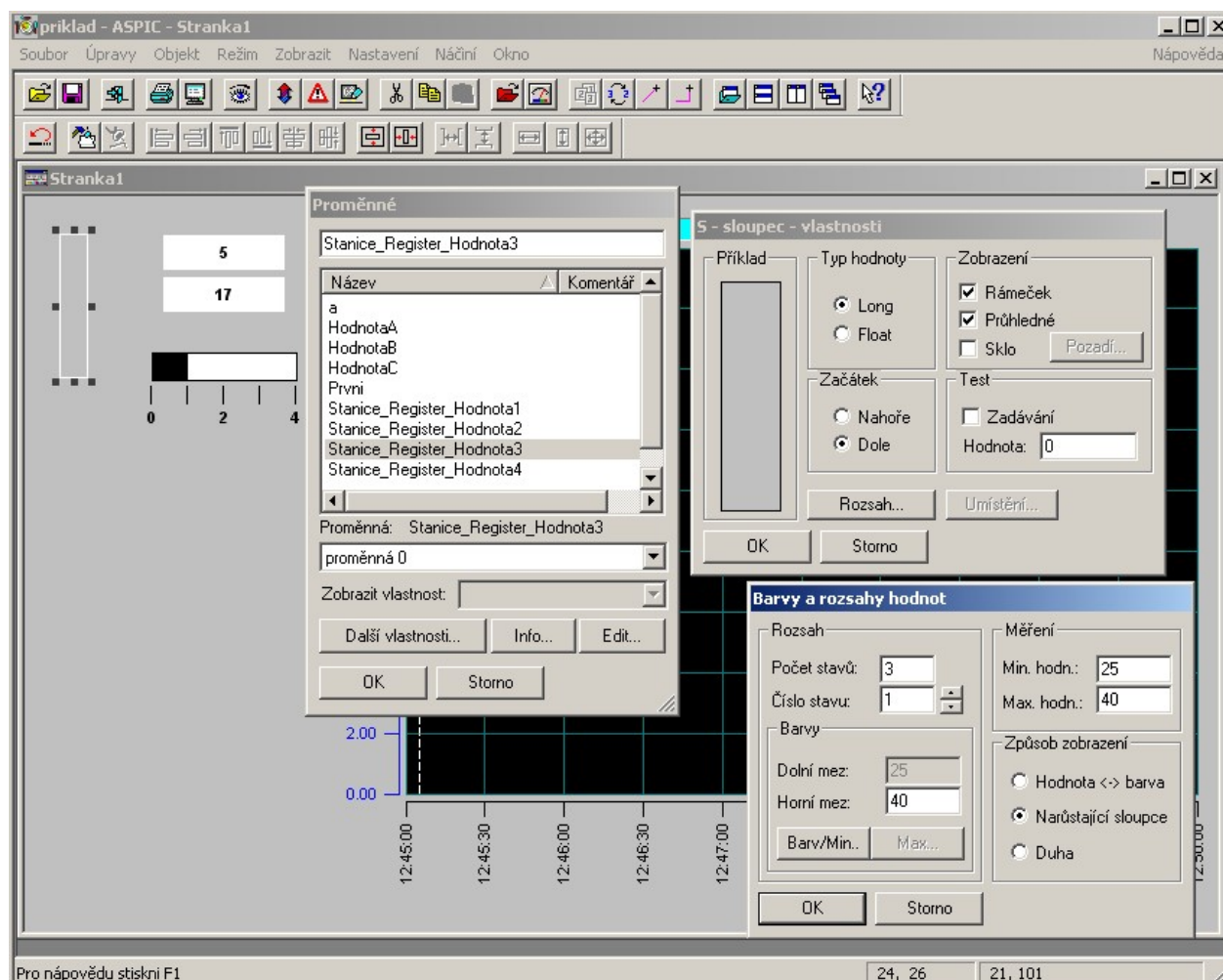
Spustíte vizualizaci a kliknutím myši na objekt „V-sloupec“ dojde ke změně hodnoty proměnné „HodnotaC“. Proměnná je také vykreslována do grafu.



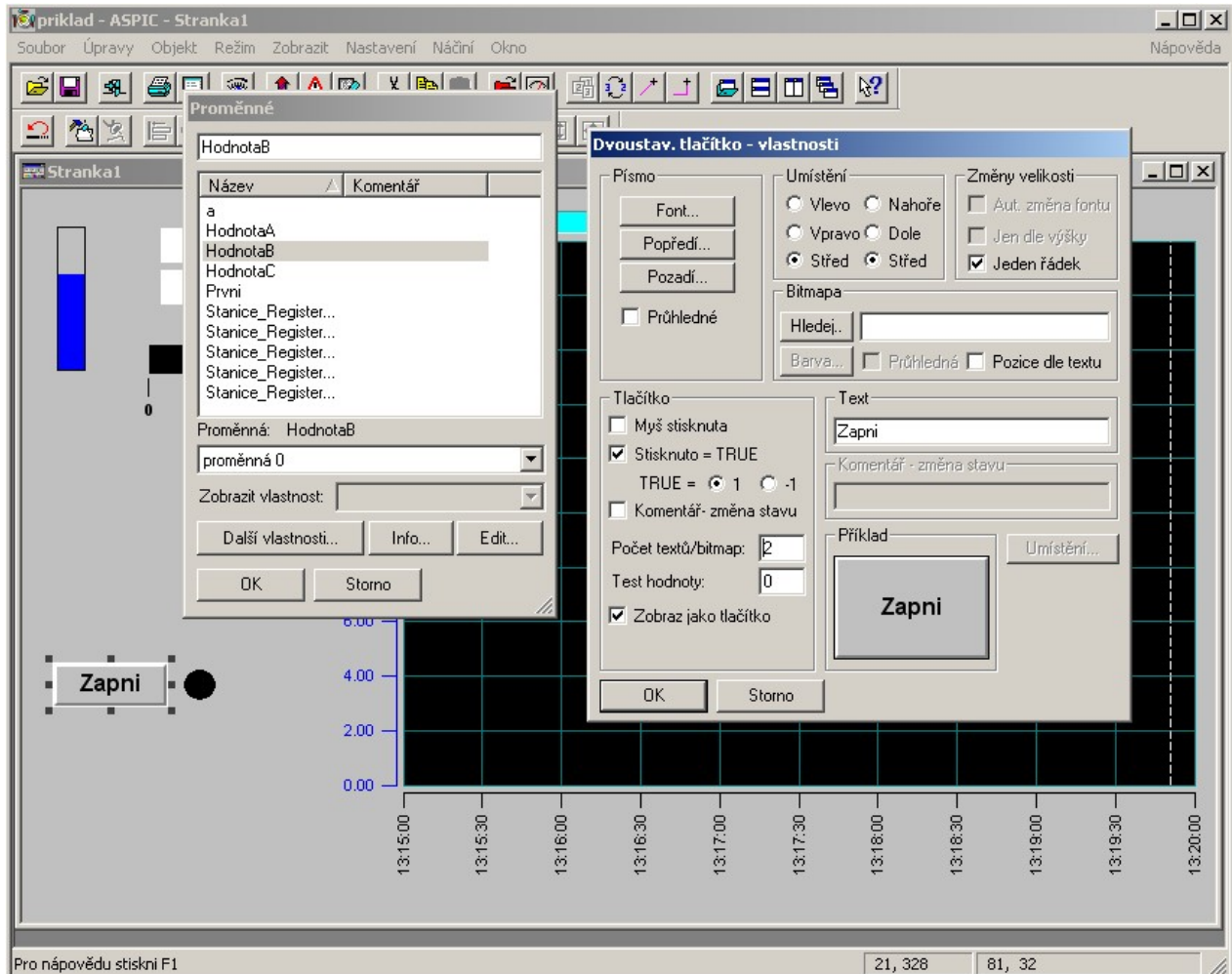
29. Upravte nastavení vlastností grafu. Zobrazte dialog „Graf – vlastnosti“ a stiskněte tlačítko „Čas a hodnota“. Jelikož nepoužíváme žádný konkrétní rozměr proměnných zobrazených v grafu, ponechte prázdnou položku „Jednotka“ a zaškrtněte „Ukazuj všechny hodnoty“ a „V řadě“. Při spuštěné vizualizaci budou zobrazovány hodnoty všech proměnných připojených do objektu „Graf“.



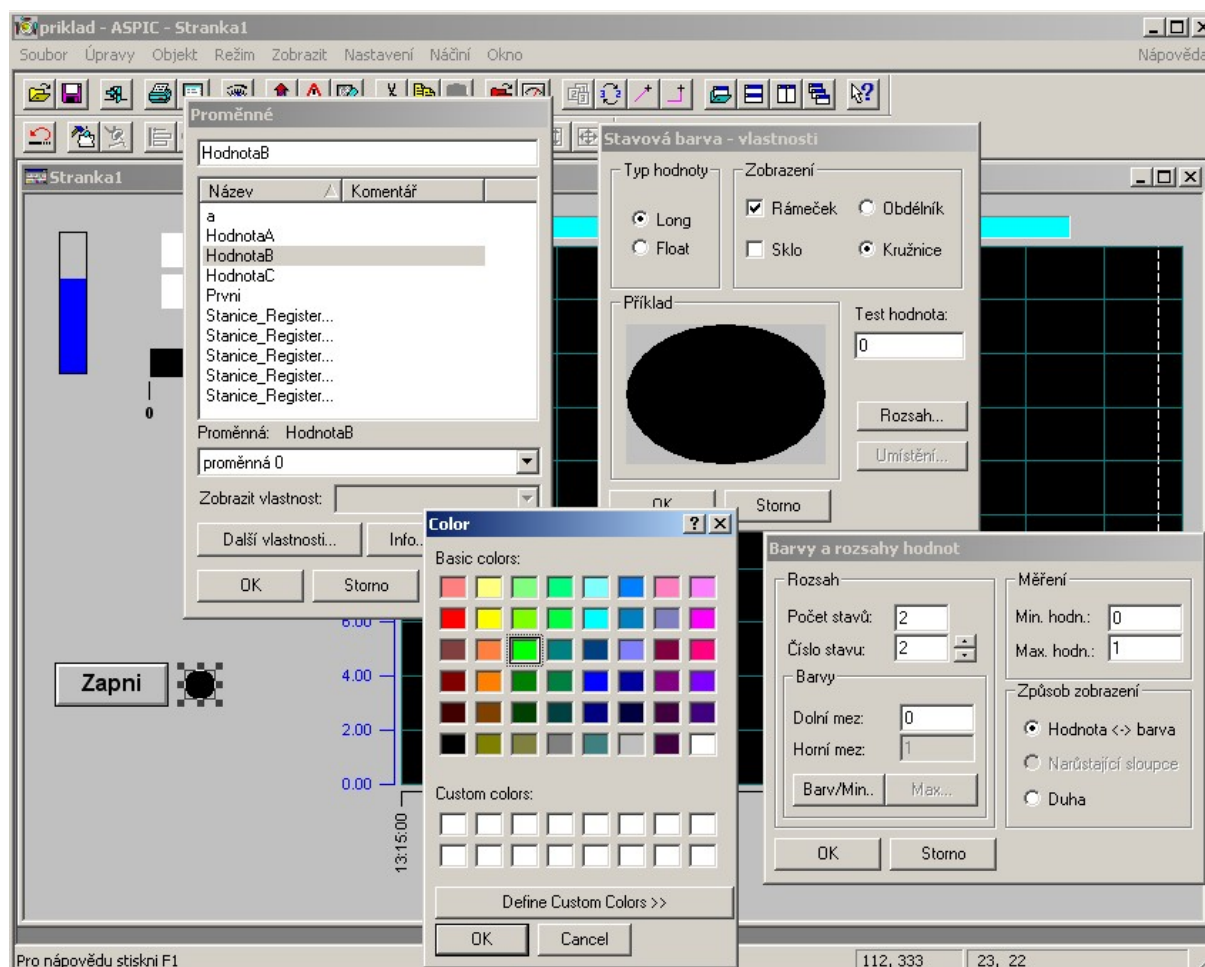
30. Na vizualizační obrazovku „Stranka1“ přidejte aktivní prvek „S – sloupec“, napojte ho na proměnnou „Stanice_Register_Hodnota3“ a jeho vlastnosti nastavte dle uvedeného obrázku. Rozsahy měření jsou pro větší názornost nastaveny na 25 a 40. Hodnota připojené proměnné může nabývat hodnot 30 a 35.



31. Přidáme aktivní objekty „Dvoustavové tlačítko“ a „Stavová barva“. Budeme simulovat žárovku která je ovládána spínačem. Oba objekty připojíme na paměťovou proměnnou „HodnotaB“. HodnotaB je paměťová proměnná, nezapomeňte ji inicializovat (viz. bod 28.) například na hodnotu 0. Pro „Dvoustavové tlačítko“ vytvořte dva texty. Pro hodnotu 0 bude text „Zapni“ a pro hodnotu 1 text „Vypni“.

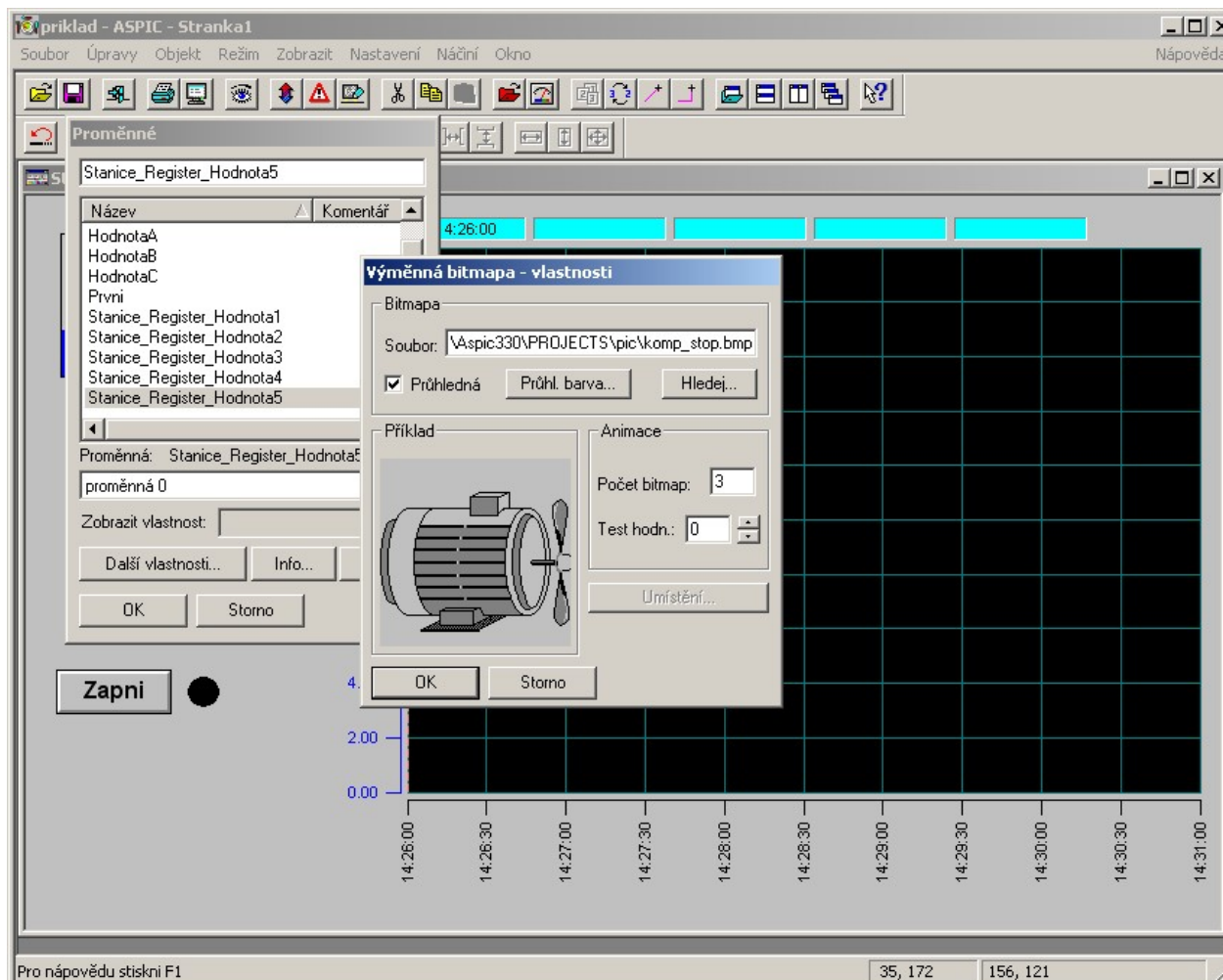


Objektu „Stavová barva“ nastavte dva stavy a rozsah měření 0 a 1. Budeme simulovat žárovku, tedy zapnuto a vypnuto. Každému ze stavů přiřadte různou barvu. Stav „1“ černou (vypnuto) a stavu „2“ zelenou (zapnuto).

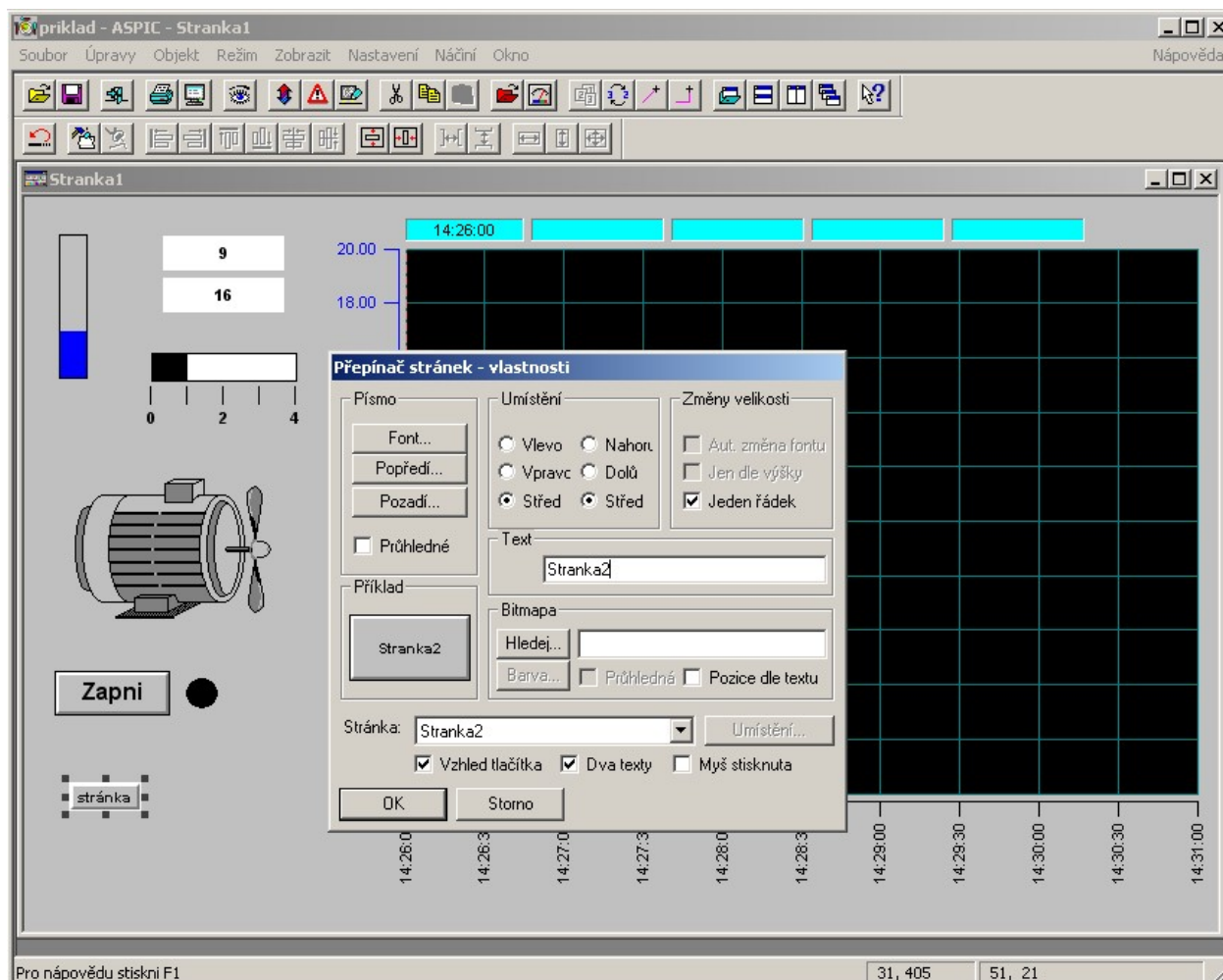


Vše vyzkoušejte spuštěním vizualizace.

32. Nyní si ukážeme, jakým způsobem lze na vizualizační obrazovce zobrazit stav zařízení, například motoru. V příkladu budeme používat obrázky, které naleznete jako součást ukázkového projektu. Vložte na vizualizační obrazovku „Stranka1“ jeden aktivní objekt „Výměnná bitmapa“ a připojte ho na proměnnou „Stanice_Register_Hodnota5“. Jeho vlastnost „Počet bitmap“ nastavte na hodnotu 3 a pro každý stav připojte obrázek. Pro stav „0“ obrázek znázorňující klid motoru, pro stav „1“ obrázek znázorňující chod motoru a pro stav „3“ obrázek poruchy motoru. Na vizualizační obrazovce bude zobrazen obrázek odpovídající hodnotě připojené proměnné.



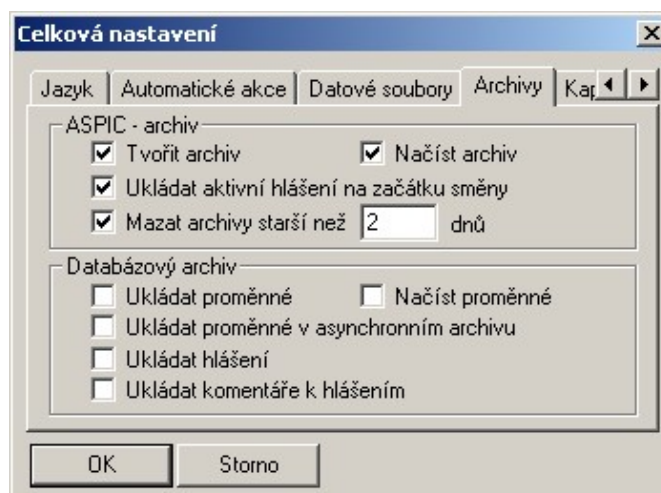
33. Na vizualizační obrazovku „Stranka2“ přidejte aktivní objekt graf, připojte na něj libovolné dostupné proměnné a nakonfigurujte jeho vlastnosti. Můžete například změnit vlastnosti „Délka grafu“ a „Průběh“. Dále přidejte na každou vizualizační obrazovku pasivní prvek „Přepínač stránky“. Proveďte změnu jeho vlastností tak, aby bylo možno navzájem přecházet mezi stránkami „Stranka1“ a „Stranka2“. Spusťte vizualizaci a ověřte chování projektu.



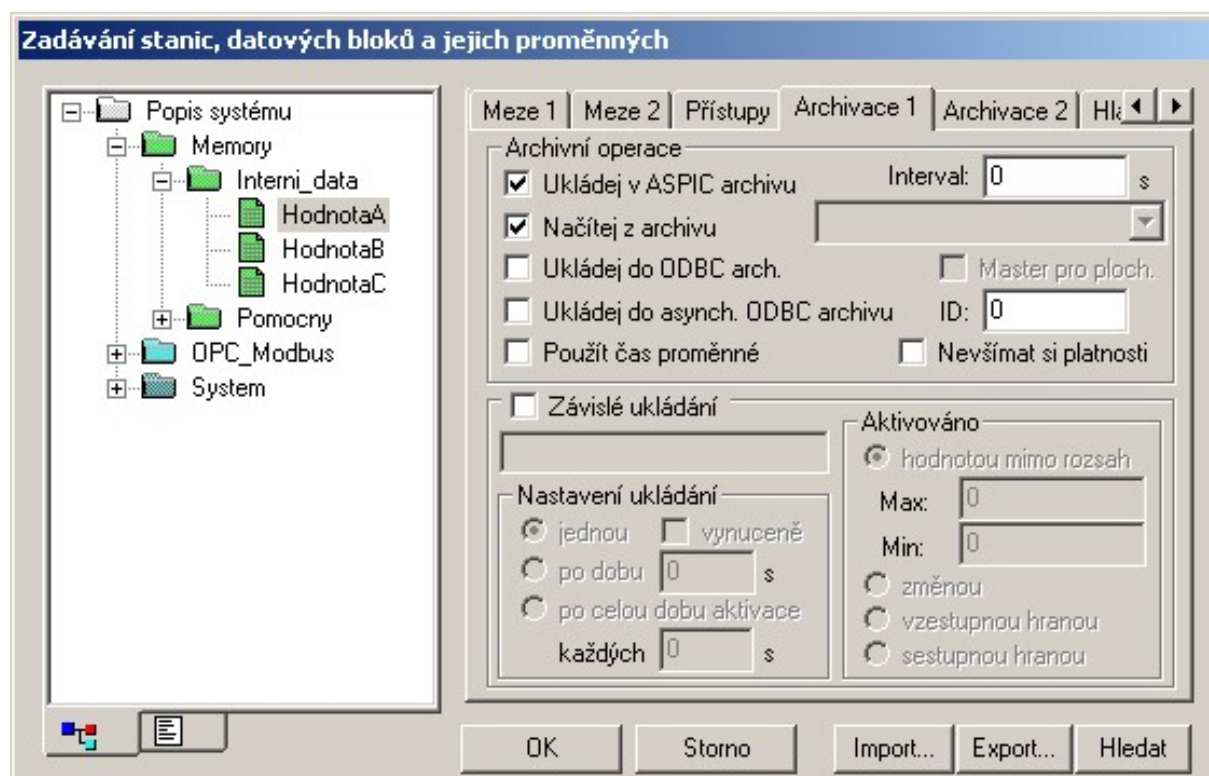
Poznámka: V návrhovém režimu Aspice 3.30 je možno se mezi vizualizačními obrazovkami přepínat výběrem menu „Okno“

34. V tuto chvíli máme umístěny na vizualizačních obrazovkách všechny potřebné objekty. Objekty jsou připojeny na proměnné a zobrazují správné hodnoty. Požadavkem tohoto příkladu je hodnoty proměnných „Stanice_Register_Hodnota1“ až „Stanice_Register_Hodnota5“ a „HodnotaA“ až „HodnotaC“ archivovat. Archivaci hodnot lze v Aspic 3.30 provádět různě, my si pro ukázkou vybereme nejjednodušší z nich a tou je archivace dat „do souboru“, ASPIC archivu.

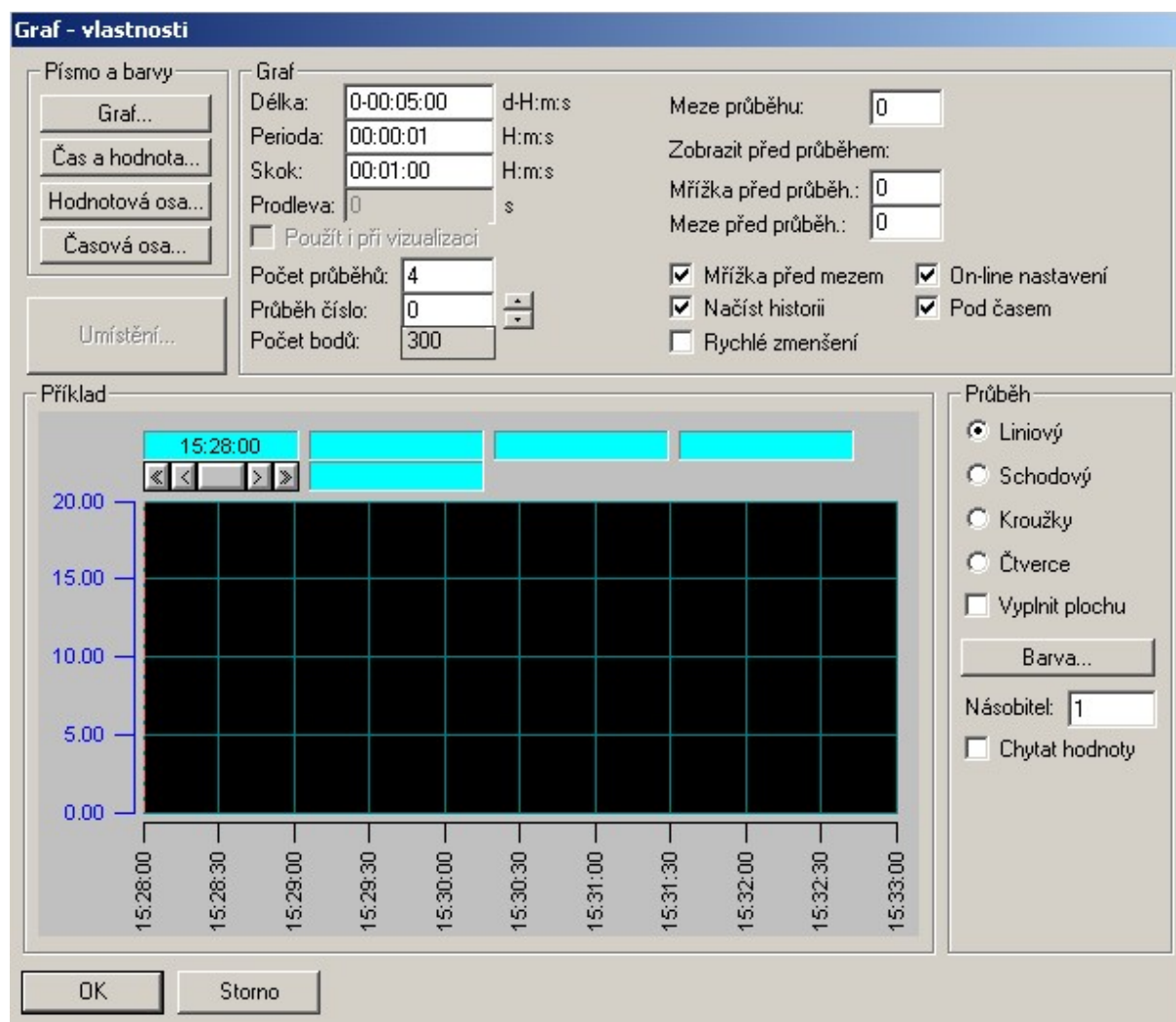
Nejprve je potřeba zaškrtnout „Tvořit archiv“ a „Načíst archiv“ v dialogu Nastavení->Volby->karta Archivy



Dále pro proměnné, které chceme do ASPIC archivu ukládat musíme zaškrtnout „Ukládej v ASPIC archivu“ a „Načítej z archivu“. Toto nastavení proveďte pro všech osm proměnných, které chceme archivovat.



Dále bychom rádi, aby byly hodnoty automaticky po startu vizualizace dearchivovány (vykresleny do grafu). Zaškrtněte „Načíst historii“ v dialogu „Graf – vlastnosti“. Toto nastavení je globální pro graf a týká se všech proměnných zobrazovaných v grafu.



Nyní spusťte vizualizaci, nechte ji chvíli (postačí 10 vteřin) běžet, poté ji vypněte a znovu spusťte. Pokud jste všechna nastavení provedli správně, bude do grafu načtena historie hodnot proměnných.

Poznámka: Nebudou načteny všechny proměnné. Bude chybět průběh „HodnotaA“. Jde o proměnnou, která je vypočítávána v matematickém modulu a během procesu dearchivace (při spuštění vizualizace) jsou dearchivované hodnoty přepsány aktuálně vypočítanými. Pro odstranění tohoto nedostatku stačí založit novou proměnnou stanice „System“ se jménem např. „Dearch“ a přiřadit ji typ „dearchivace“. Tato proměnná bude nabývat hodnoty „true“ pouze při dearchivaci. Následně stačí výpočet hodnoty „HodnotaA“ v matematickém modulu podmínit hodnotou proměnné „Dearch“. Viz. následující obrázky. Úpravu vyzkoušejte.

Zadáání stanic, datových bloků a jejich proměnných

Popis systému

- Memory
 - Interni_data
 - HodnotaA
 - HodnotaB
 - HodnotaC
 - Pomocny
 - OPC_Modbus
 - System
 - Archivace
 - Dearch**
 - Cas
 - Prvni

Typy
Hodnoty
Platnost
Meze 1
Meze 2
Přístupy
A

Komentář

Umístění, vstupní typ

Offset v bloku: 0
Bit: 0

Typ: dearchivace

Zobrazení

☒ Long
☐ Float
☐ String

Editace

Kopíruj...

Přidej...

Vyřizni...

Smaž

Přejmenuj...

Čas z veličiny:

System state type

Skupina

Typ

Stanice

OK
Storno
Import...
Export...
Hledat

Editor vzorců - příklad.apr

Vzorce Úpravy Zobrazit Nastavení Nástroje

```


if(Prvni)
{
    a = 0.0;
    HodnotaC = 1;
    HodnotaB = 0;
}

if (!Dearch)
{
    HodnotaA = ((sin (a) * 2) + 10.0);
    if (a < 360.0)
        a = a + 0.25;
    else
        a = 0.0;
}

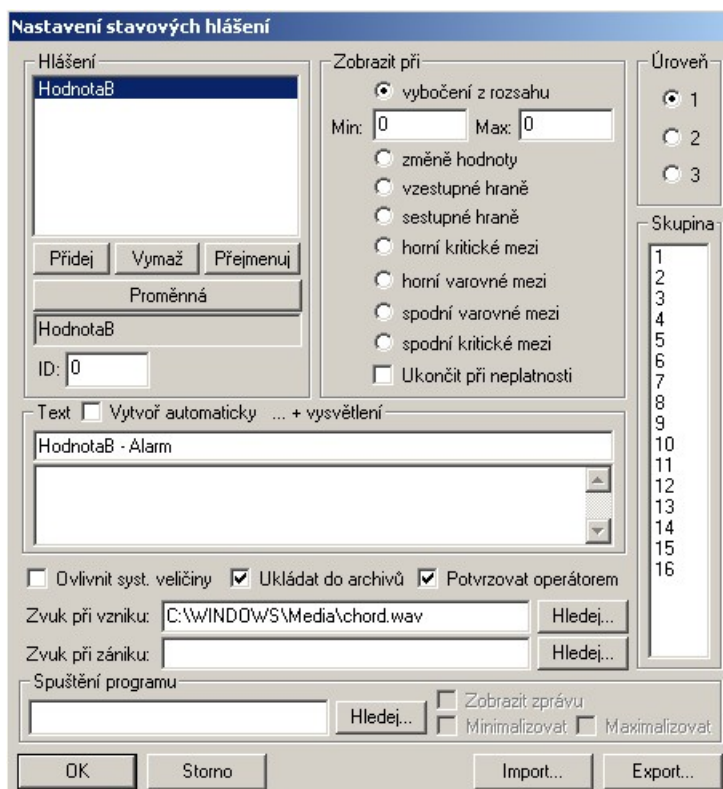
```

Pro nápovědu stiskni F1 Řád 1, Sl 1

35. Dalším úkolem příkladu je vytvoření alarmového systému. Ukážeme si, jakým způsobem alarmy vytvářet a jak je archivovat pro využití reportovacího nástroje.

Dle požadavků na náš projekt budeme chtít vytvořit alarmy, které vyžadují potvrzení operátora a pouze jeden z požadovaných alarmů bude „Zvukový“. Alarmy se v Aspic 3.30 definují v okně „Nastavení stavových hlášení“, okno otevřete klepnutím na ikonu  v panelu nástrojů.

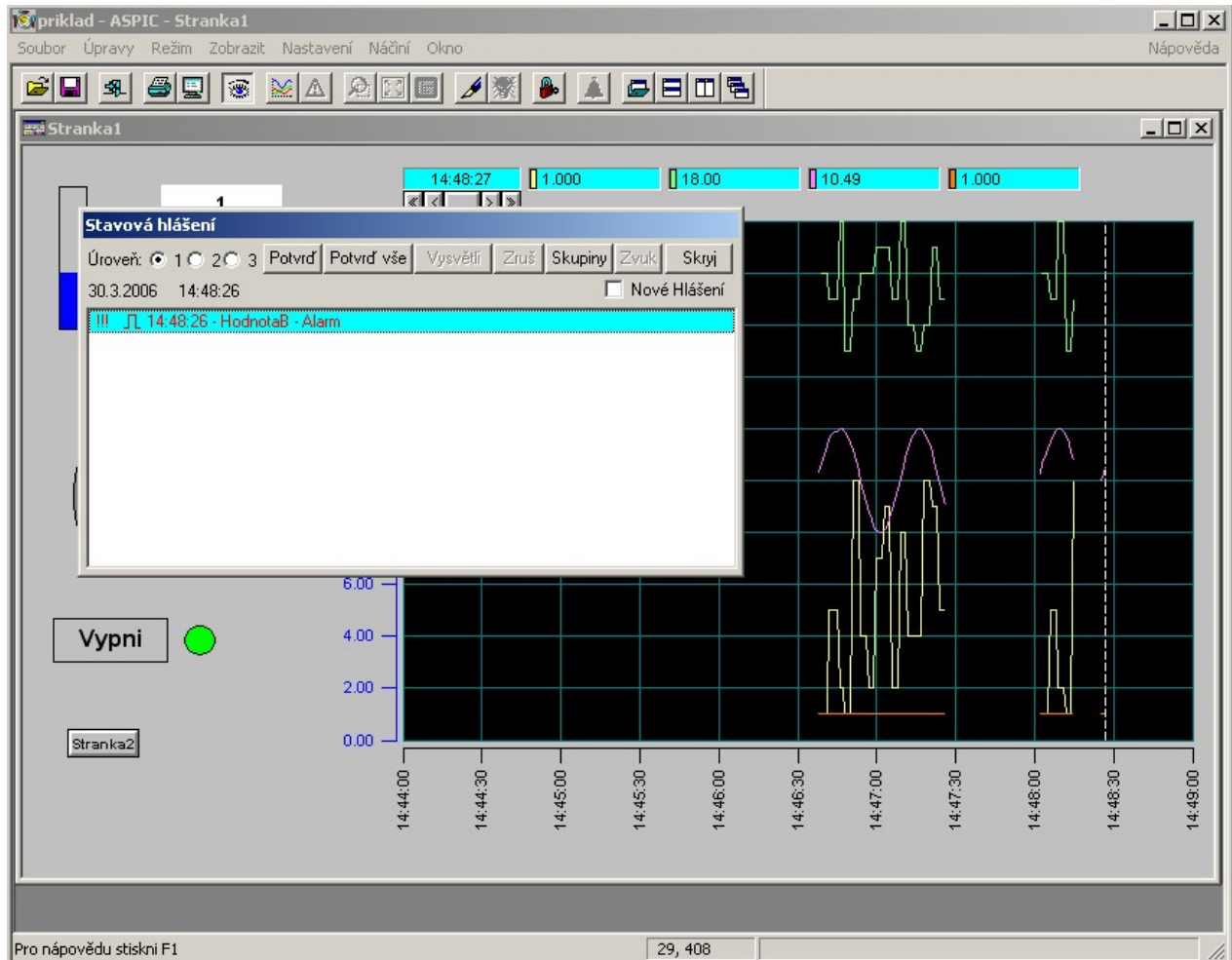
- a) nadefinujeme zvukový alarm, který bude vyhlášen v případě, že hodnota proměnné HodnotaB je aktivní (1). Proměnnou HodnotaB jsme navázali na tlačítko umístěné na vizualizační obrazovce „Stranka1“, kterým můžeme snadno pro naše testovací účely měnit hodnotu proměnné a vyhlásit alarm. Otevřete dialog „Nastavení stavových hlášení“ a přidejte nové hlášení „HodnotaB“ a navažte toto hlášení na proměnnou „HodnotaB“. Proměnná „HodnotaB“ nabývá pouze dvou stavů (0,1), při stavu „1“ chceme vyhlásit alarm. „Zobrazit při“ nastavte na „vybočení z rozsahu“ (0 – 0). „ID“ ponechte 0, „Úroveň“ nastavte na 1 (nejvyšší úroveň alarmů), do pole „Text“ napište pro obsluhu srozumitelnou hlášku o vzniklé události, připojte „Zvuk při vzniku“ na nějaký vhodný zvuk (může se jednat i o lidským hlasem mluvenou hlášku) v našem příkladu jsme použili systémový zvuk z kolekce operačního systému a zaškrtněte „Potvrzovat operátorem“ a „Ukládat data do archivů“.



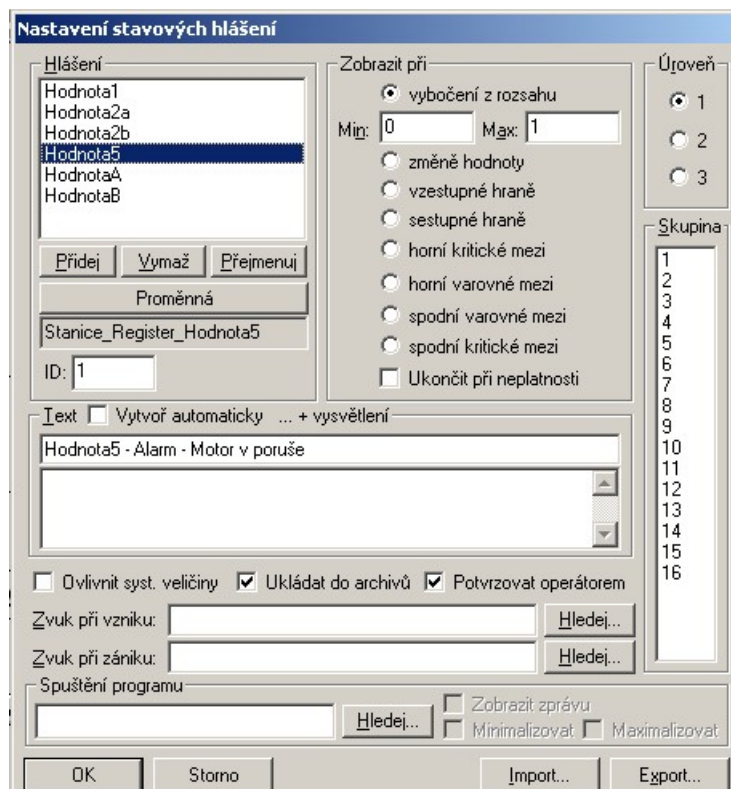
Poznámka: V okně „Nastavení stavových hlášení“ je k dispozici mnoho nastavení. Práce s nimi je nad rámec tohoto příkladu a musíme Vás odkázat na uživatelskou příručku Aspic 3.30. Jedinou oblast zmíníme a tou je možnost v případě vzniku alarmu spustit externí program a předat mu libovolné parametry. Tímto způsobem lze velmi snadno předávat alarmy pomocí sms, emailů, atd...

Nyní spusťte vizualizaci a stiskněte tlačítko „Zapni“ umístěné na vizualizační obrazovce „Stranka1“. Hodnota proměnné „HodnotaB“ se změní z 0 na 1 a dojde k vyhlášení alarmu. Alarm je zobrazen v okně „Stavová hlášení“.

Vzniklý alarm je možno potvrdit, vložit komentář obsluhy nebo ho v případě dostatečných práv smazat.



- b) dále je třeba přidat alarm signalizující poruchu motoru. Stav motoru vyjadřuje hodnota proměnné „Stanice_Register_Hodnota5“. Pokud proměnná nabude hodnoty „2“, došlo k poruše motoru.

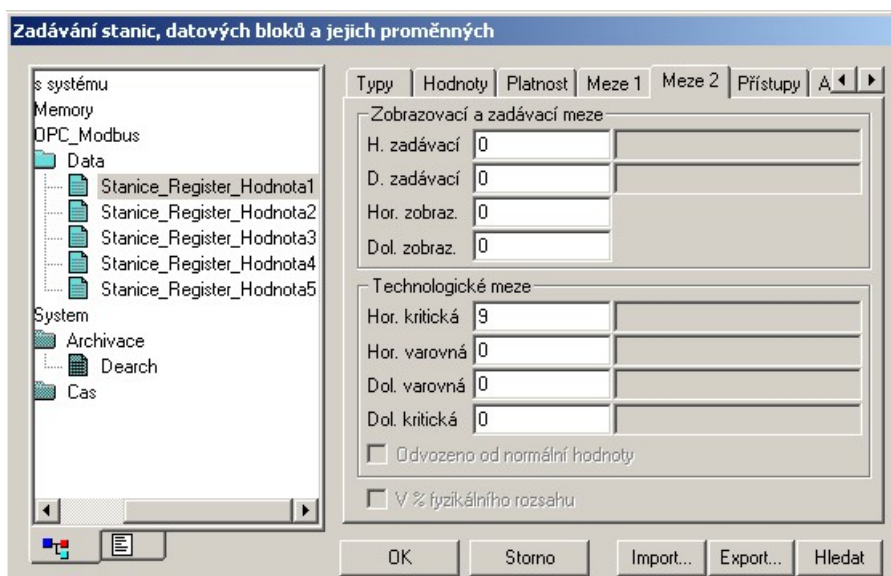
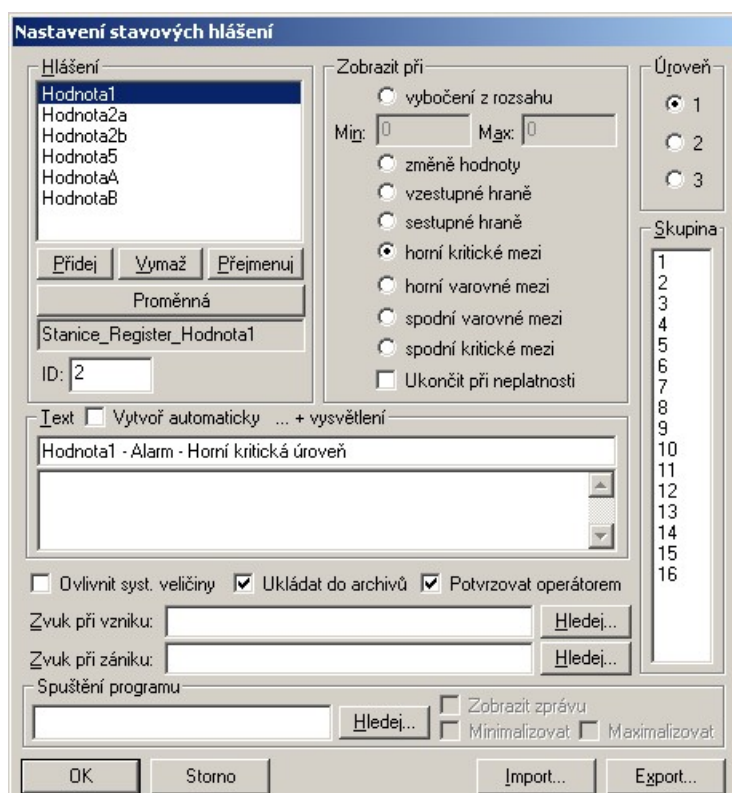


Alarm „Hodnota5“ navažte na proměnnou „Stanice_Register_Hodnota5“. Volbu „Zobrazit při“ ponechte ve stavu „vybočení z rozsahu“, pouze změňte meze rozsahu (0 – 1). Hodnota, při níž musí být aktivován alarm je „2“.

Úpravu vyzkoušejte.

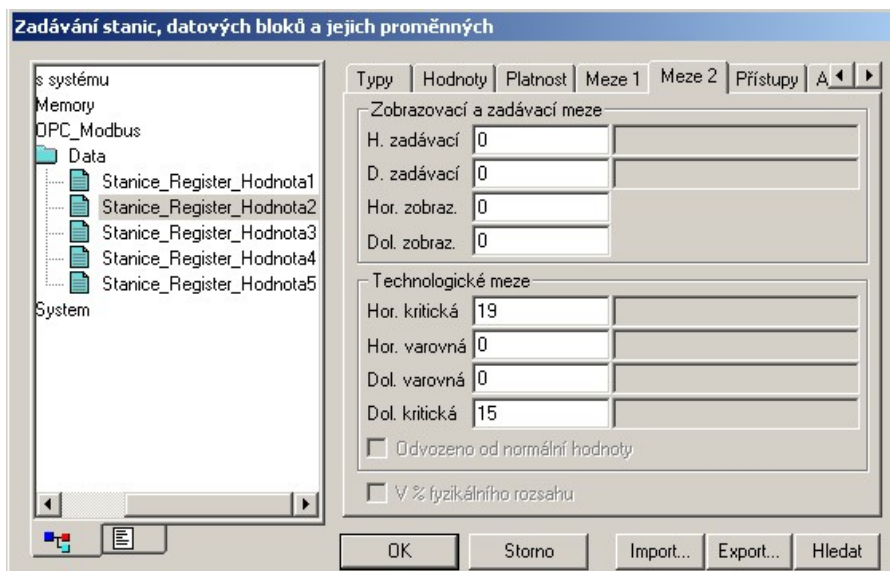
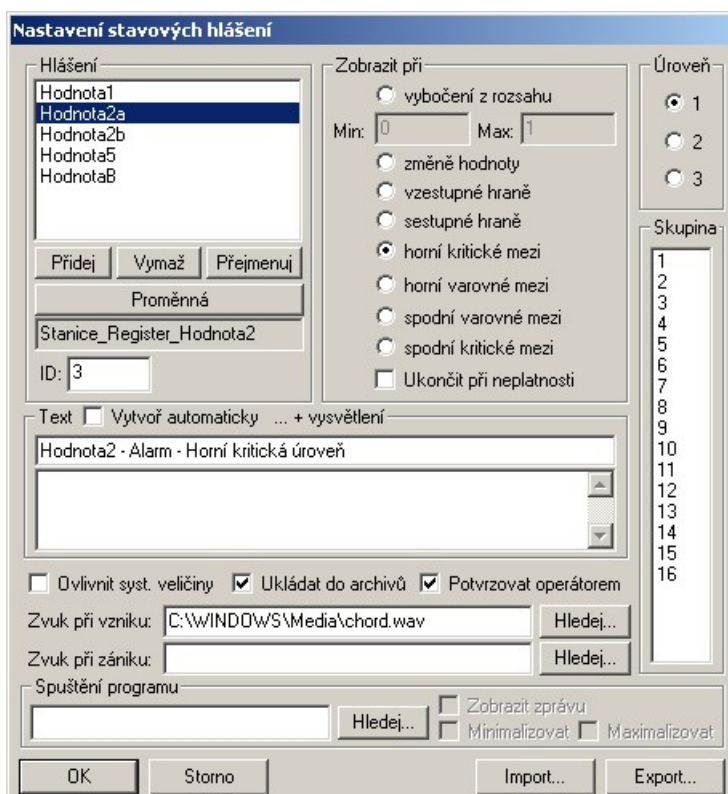
Poznámka: Zvuky při vzniku alarmů lze v Aspic 3.30 také definovat globálně pro určité úrovně alarmů. Případné zájemce odkazují na uživatelskou příručku k Aspic 3.30.

- c) dále přidáme alarm, který bude vyhlášen, dosáhne-li hodnota proměnné „Stanice_Register_Hodnota1“ horní kritické úrovně. Horní kritická úroveň se definuje pro jednotlivou proměnnou. Zobrazte již známý dialog „Zadávání stanic, datových bloků a jejich proměnných“ a pro proměnnou „Stanice_Register_Hodnota1“ na kartě „Meze2“ zadejte horní kritickou mez.

Dále v dialogu „Nastavení stavových hlášení“ přidejte alarm s názvem „Hodnota1“ a zvolte „Zobrazit při horní kritické mezi“. Další vlastosti alarmu nastavte podle uvedeného obrázku.

- d) Obdobně jako předchozí alarmy přidejte alarm pro dosažení horní i dolní kritické úrovně proměnné „Stanice_Register_Hodnota2“ (dolní mez = 15, horní mez =19) a alarm pro dolní kritickou mez proměnné „HodnotaA“ (9). Nezapomeňte nejen vytvořit alarmy, ale i v dialogu „Zadávání stanic, datových bloků a jejich proměnných“ upravit požadované horní a dolní kritické meze dle potřeb našeho projektu.

Nastavení stavových hlášení

Hlášení

- Hodnota1
- Hodnota2a
- Hodnota2b**
- Hodnota5
- Hodnota8

Přidej Vymaž Přejmenuj

Proměnná

Stanice_Register_Hodnota2

ID: 4

Zobrazit při

☐ vybočení z rozsahu
Min: 0 Max: 1

☐ změně hodnoty

☐ vzestupné hraně

☐ sestupné hraně

☐ horní kritické meze

☐ horní varovné meze

☐ spodní varovné meze

☒ spodní kritické meze

☐ Ukončit při neplatnosti

Úroveň

☒ 1

☐ 2

☐ 3

Skupina

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16

Text ☐ Vytvoř automaticky ... + vysvětlení

Hodnota2 - Alarm - Spodní kritická úroveň

☐ Ovlivnit syst. veličiny ☒ Ukládat do archivů ☒ Potvrzovat operátorem

Zvuk při vzniku: C:\WINDOWS\Media\chord.wav Hledej...

Zvuk při zániku: Hledej...

Spuštění programu Hledej...

☐ Zobrazit zprávu ☐ Minimalizovat ☐ Maximalizovat

OK Storno Import... Export...

Zadávání stanic, datových bloků a jejich proměnných

Popis systému

- Memory
 - Interni_data
 - HodnotaA**
 - HodnotaB
 - HodnotaC
 - Pomocny
- OPC_Modbus
- System

Typy Hodnoty Platnost Meze 1 Meze 2 Přístupy A B C

Zobrazovací a zadávací meze

H. zadávací 0

D. zadávací 0

Hor. zobraz. 0

Dol. zobraz. 0

Technologické meze

Hor. kritická 0

Hor. varovná 0

Dol. varovná 0

Dol. kritická 9

☐ Odvozeno od normální hodnoty

☐ V % fyzikálního rozsahu

OK Storno Import... Export... Hledat

Nastavení stavových hlášení

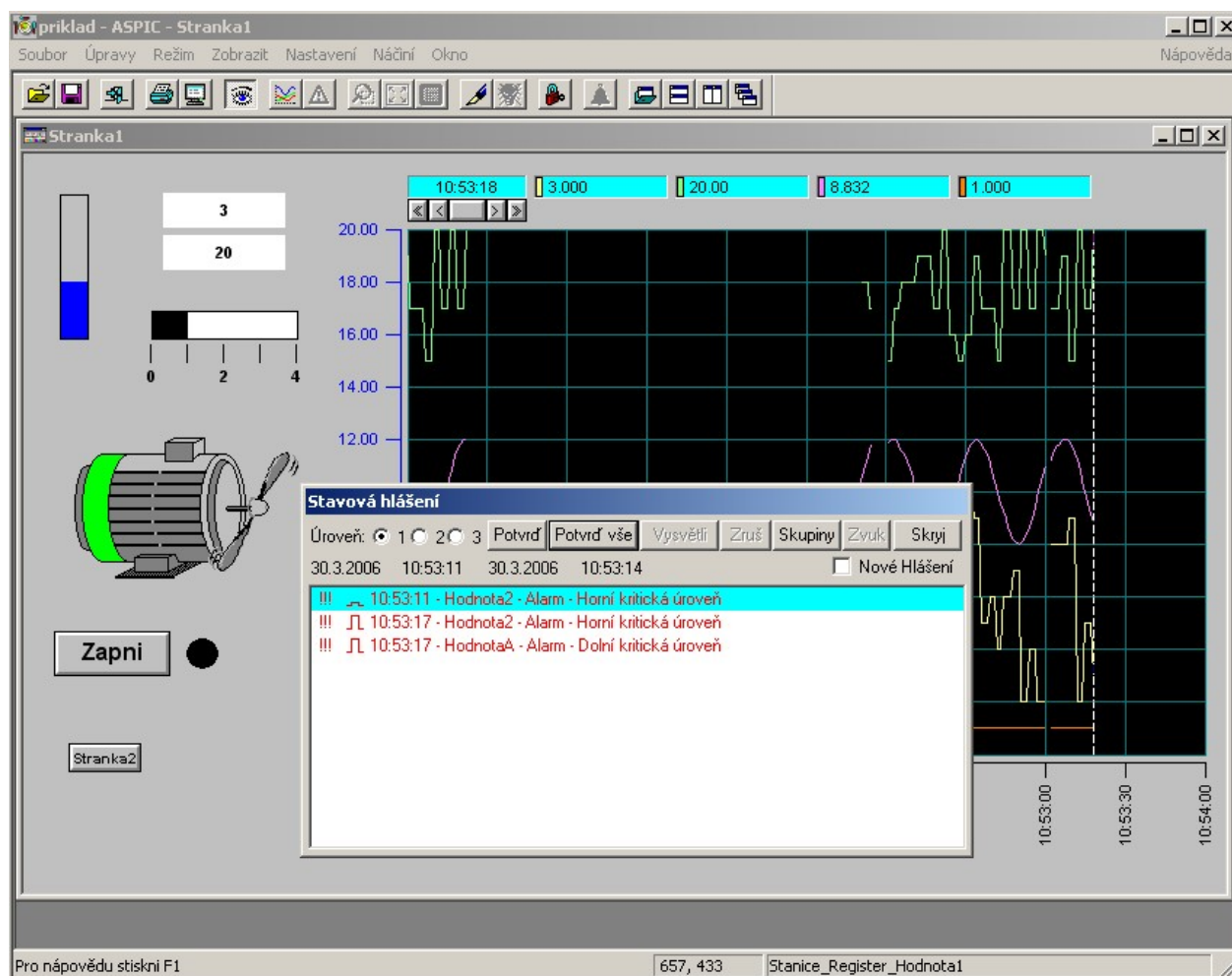
Hlášení
 Hodnota1
 Hodnota2a
 Hodnota2b
 Hodnota5
HodnotaA
 HodnotaB
 Přidej Vymaž Přejmenuj
 Proměnná
 HodnotaA
 ID: 0

Zobrazit při
☐ vybočení z rozsahu
 Min: 0 Max: 1
☐ změně hodnoty
☐ vzestupné hraně
☐ sestupné hraně
☐ horní kritické meze
☐ horní varovné meze
☐ spodní varovné meze
☒ spodní kritické meze
☐ Ukončit při neplatnosti

Úroveň
☒ 1
☐ 2
☐ 3
Skupina
 1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

Text ☐ Vytvoř automaticky ... + vysvětlení
 HodnotaA - Alarm - Dolní kritická úroveň
☐ Ovlivnit syst. veličiny ☒ Ukládat do archivů ☒ Potvrzovat operátorem
 Zvuk při vzniku: C:\WINDOWS\Media\chord.wav Hledej...
 Zvuk při zániku: Hledej...
 Spuštění programu: Hledej... ☐ Zobrazit zprávu ☐ Minimalizovat ☐ Maximalizovat
 OK Storno Import... Export...

Alarmový systém ověřte spuštěním vizualizace.

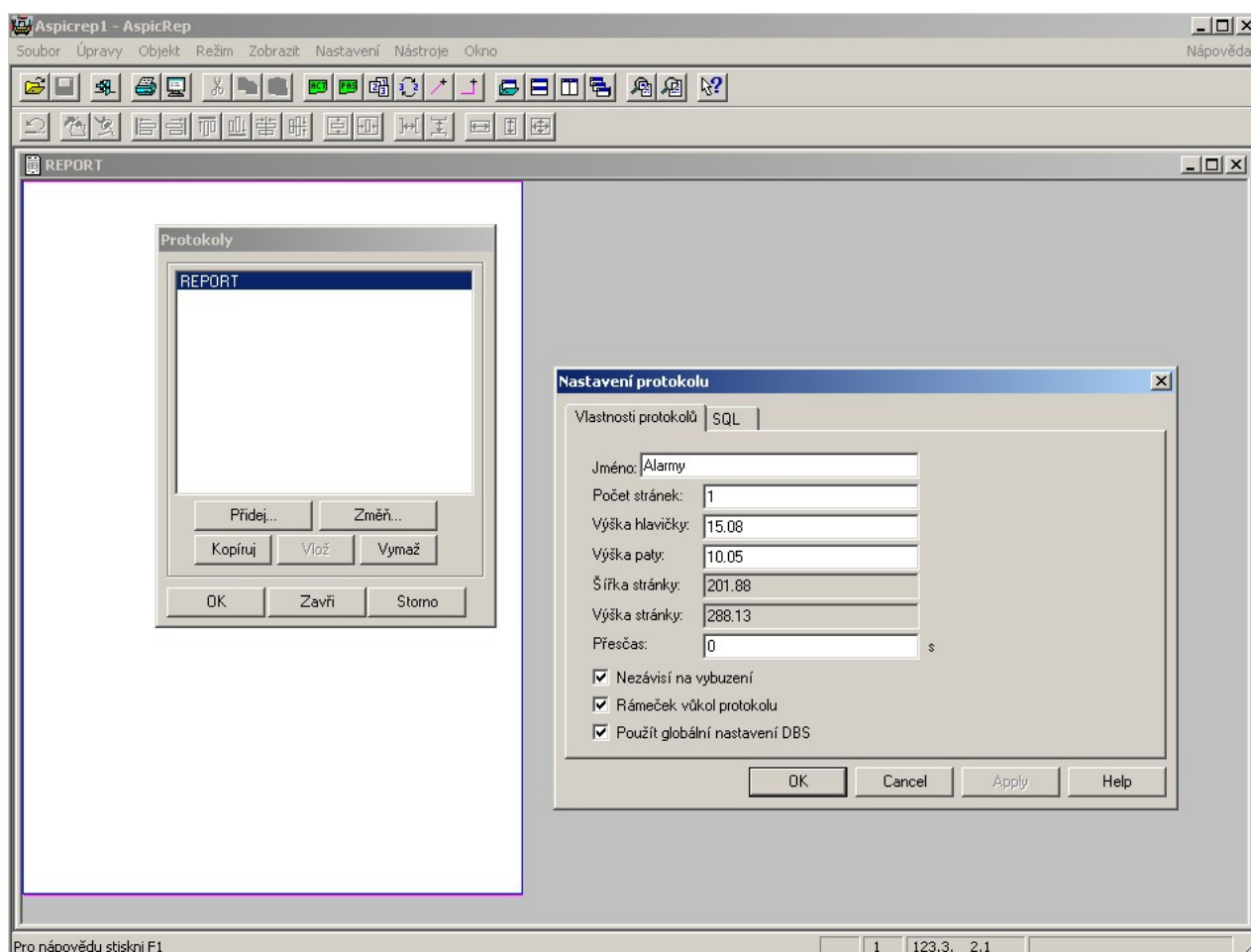


36. Poslední částí našeho projektu je tvorba protokolů „reportů“. Systém Aspic 3.30 pro tvorbu a prohlížení reportů disponuje samostatným nástrojem nazývaným „AspicRep“. AspicRep umožňuje vytvářet tiskové sestavy reportů. Využívá k tomu aktivní i pasivní objekty, obdobné jako vizualizační objekty Aspic, které jsou po umístění na zvolené místo tiskové sestavy naplněny historickými daty z databáze dle výběru uživatele.

Spusťte AspicRep, zvolte „Soubor->Nový“, pokud již není založen nový soubor s reporty.


Vyberte z menu „Okno -> Okno“, v zobrazeném dialogu jsou vypsané vytvořené „listy“ v rámci projektu reportů. Nyní je uveden jeden, s default jménem „REPORT“. Vyberte jej a stiskněte tlačítko „Změň“. Objeví se dialog „Nastavení protokolu“. Protokol pojmenujte „Alarmy“, máme v úmyslu vytvářet protokol alarmových hlášení. Do pole „Výška hlavičky“ zadejte 15 a do pole „Výška paty“ zadejte 10. Hlavička a pata se bude na každém tiskovém listu opakovat. Z obou dialogů odejděte stiskem tlačítek OK.

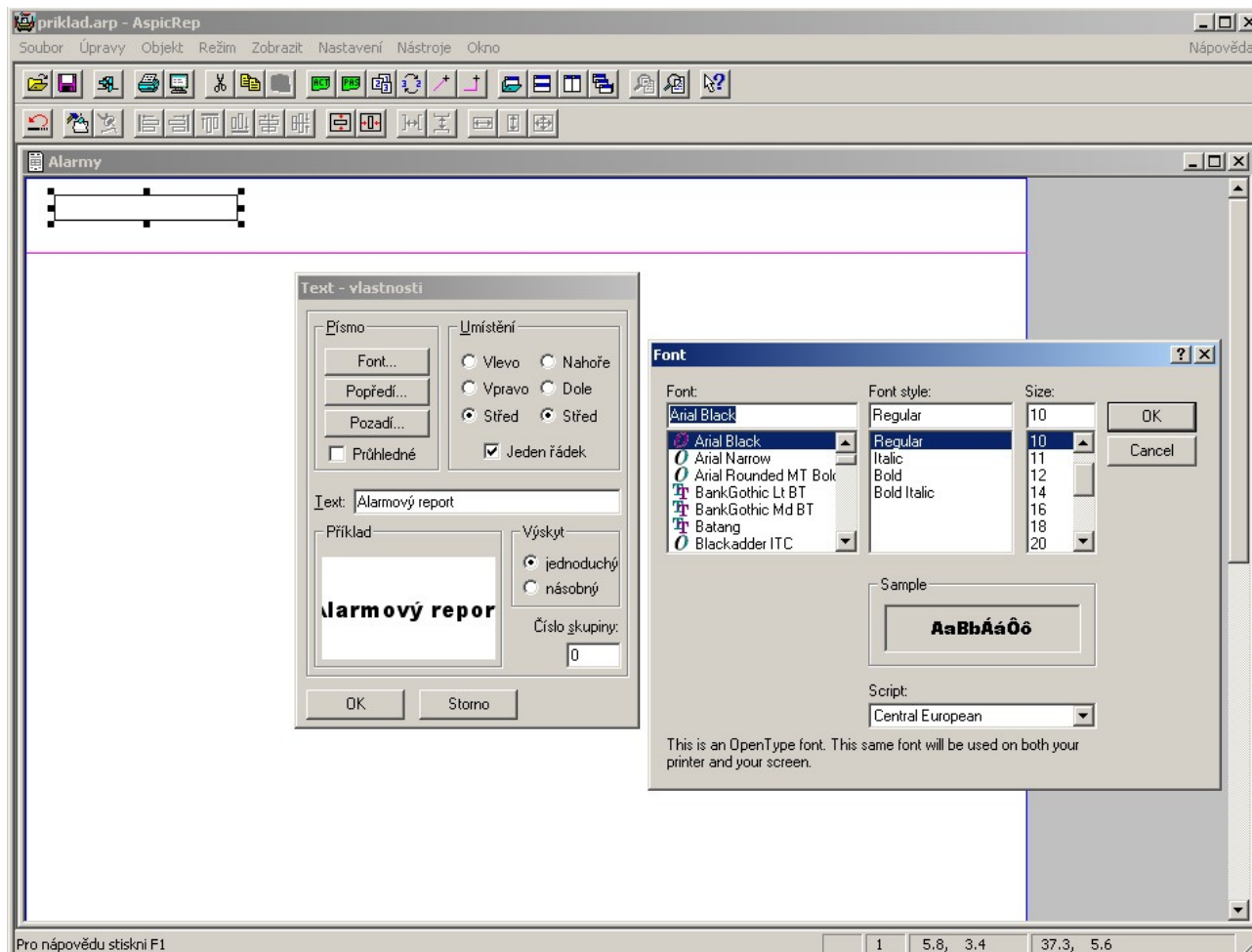
Nyní je potřeba projektový soubor s reportem uložit. Vyberte položku v menu „Soubor->Ulož jako..“. Projektový soubor s reportem uložte, nejlépe do stejného adresáře s projektovými soubory vizualizace, není to ale podmínkou. Pojmenujte ho „příklad.apr“.



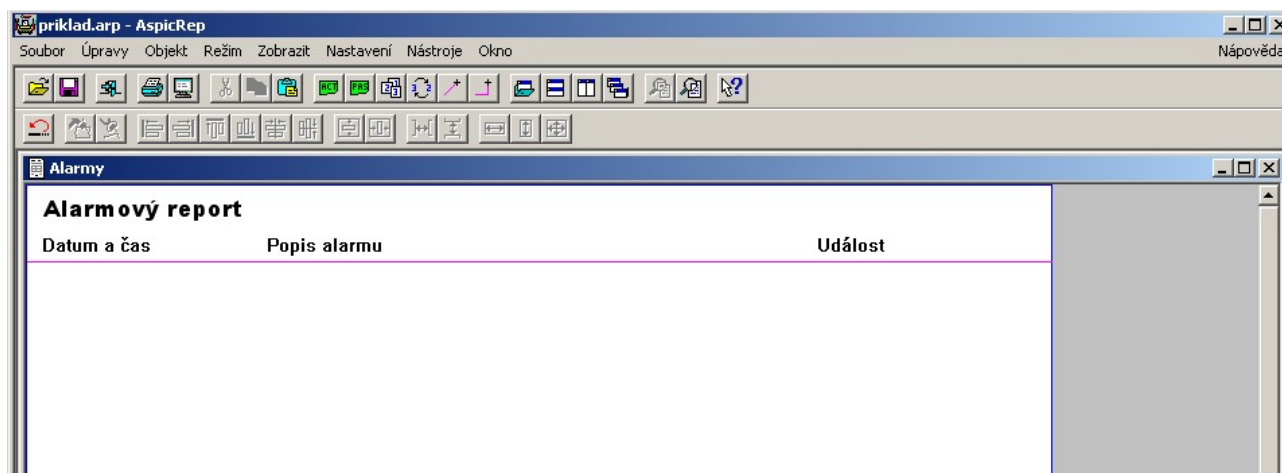
Poznámka: Tiskový list lze orientovat tzv. „na ležato“ nebo „na stojato“. Toto nastavení proveďte v menu „Soubor -> Nastav Tiskárnu -> Všechny protokoly (Vybraný protokol)“. Nastavení aplikujte buď na všechny protokoly v projektu reportů, nebo pouze na jeden. (Výpis alarmů bychom měli rádi na A4 na stojato, kdežto graf průběhu nějaké veličiny raději A4 na ležato). Formát stránky záleží na zvolené tiskárně.

37. Na list umístíte první pasivní objekt. Bude to objekt „Text“. Bude realizovat název reportu a bude umístěn v hlavičce.

Klikněte na ikonu pasivních objektů v panelu nástrojů  a vyberte objekt „Text“. Umístěte ho na list, jeho polohu a velikost zvolte přibližně dle uvedeného obrázku. Dvojklikem na tento objekt je otevřen dialog vlastností objektu. Vepište požadovaný text „Alarmový report“ a upravte jeho další vlastnosti dle potřeby (font, umístění).



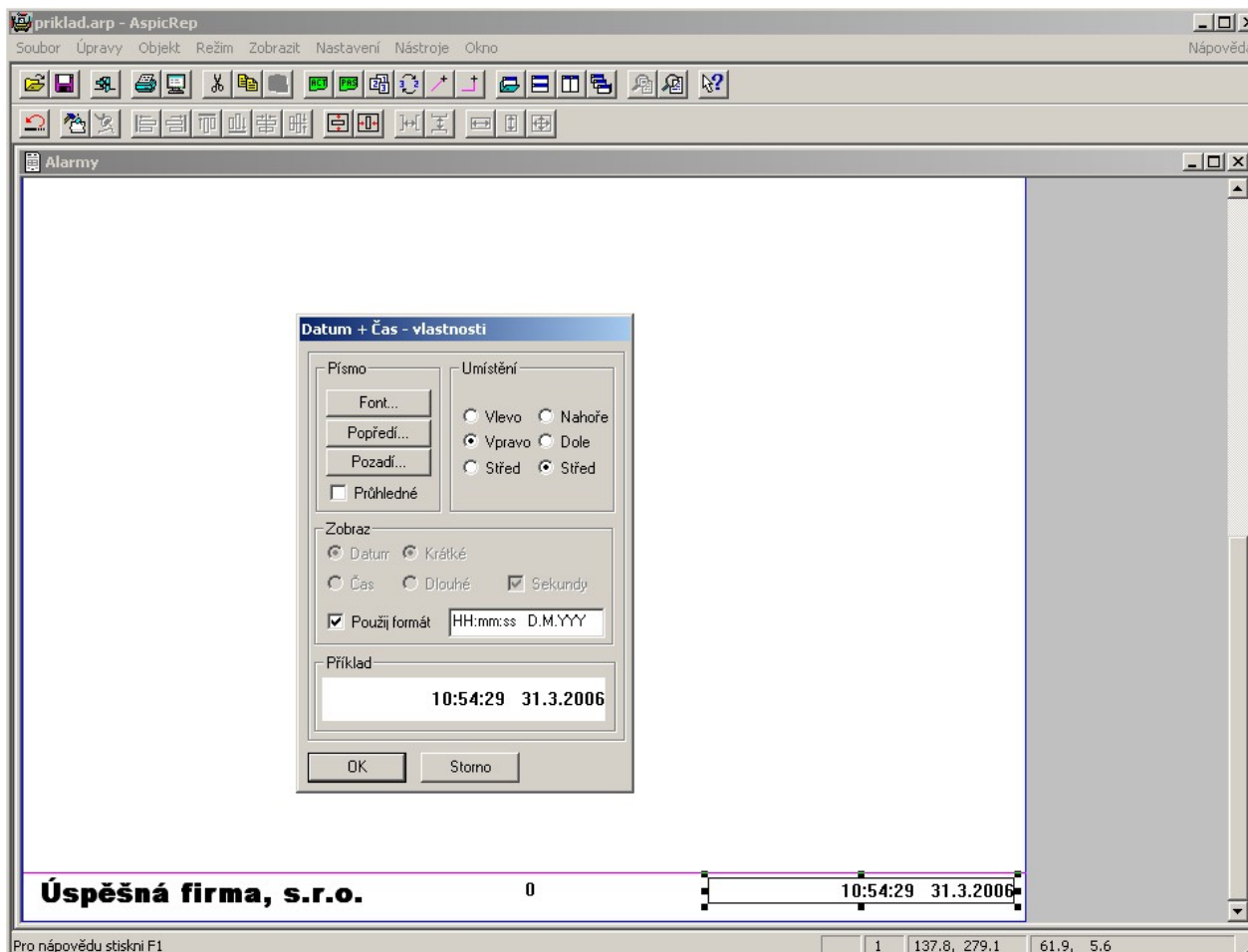
Dále vložte do hlavičky listu tři další pasivní objekty typu „Text“, které budou vytvářet záhlaví pro zobrazované alarmy. Texty a umístění volte dle následujícího obrázku.



Poznámka: Při příjemné vytváření tiskových listů je velmi užitečné zvětšování a zmenšování tiskového listu. Pro zvětšení klikněte na , pro zmenšení na .


Dále vytvoříme zápatí tiskového listu. Umístíme do něj název firmy (lze umístit i logo), číslo stránky a datum vytvoření reportu.

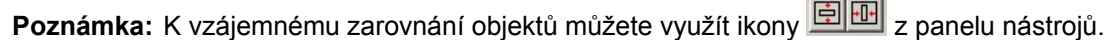
Vložte pasivní objekt typu „Text“ a vepište do něj název firmy. Dále do zápatí umístěte pasivní objekt typu „Číslo stránky“ a nakonec pasivní objekt typu „Datum + čas“. Vlastnosti posledně jmenovaného objektu nastavte dle obrázku. Důležitý je formátovací řetězec „HH:mm:ss D.M.YYY“. Význam formátovacích řetězců je uveden v uživatelské příručce Aspic 3.30.

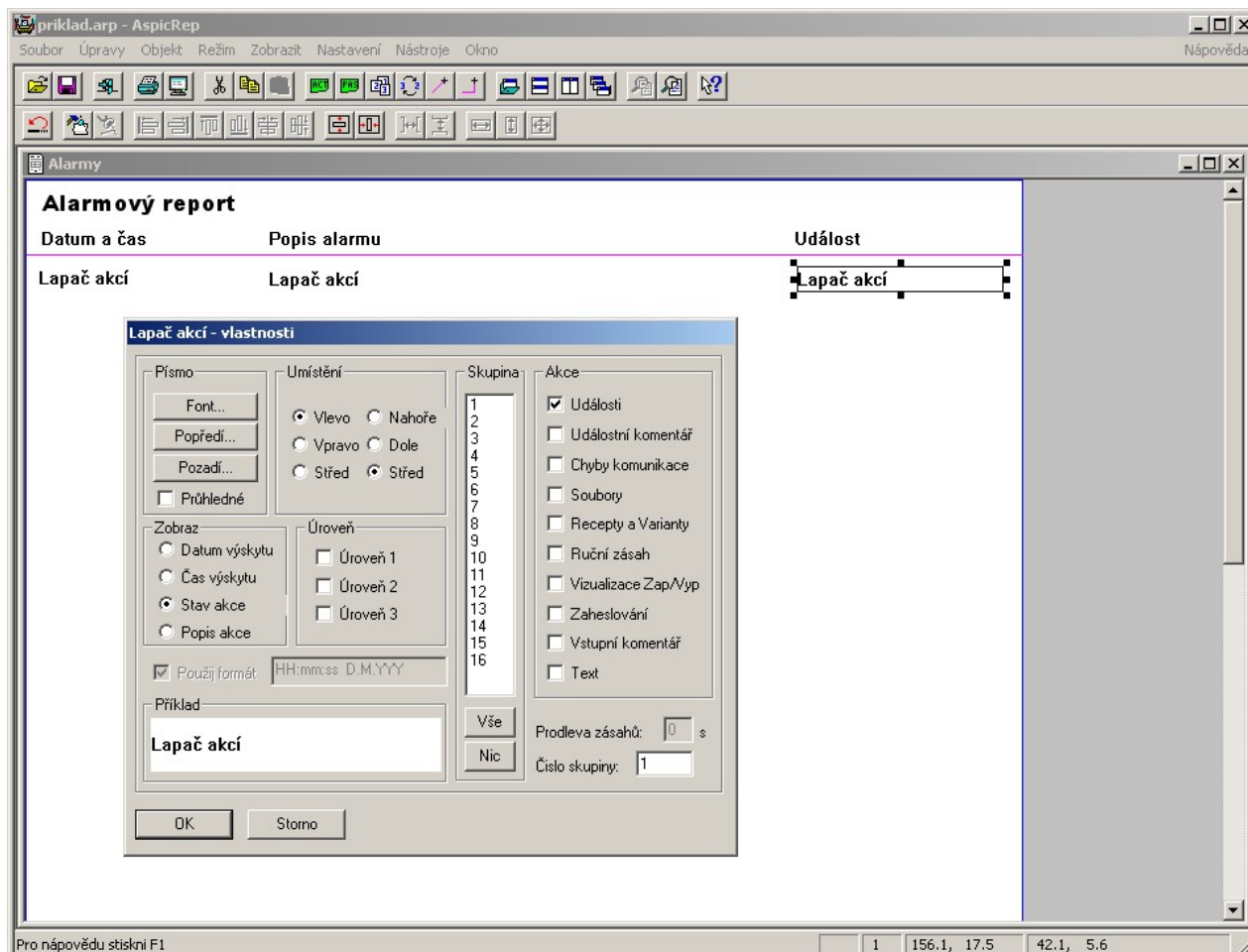
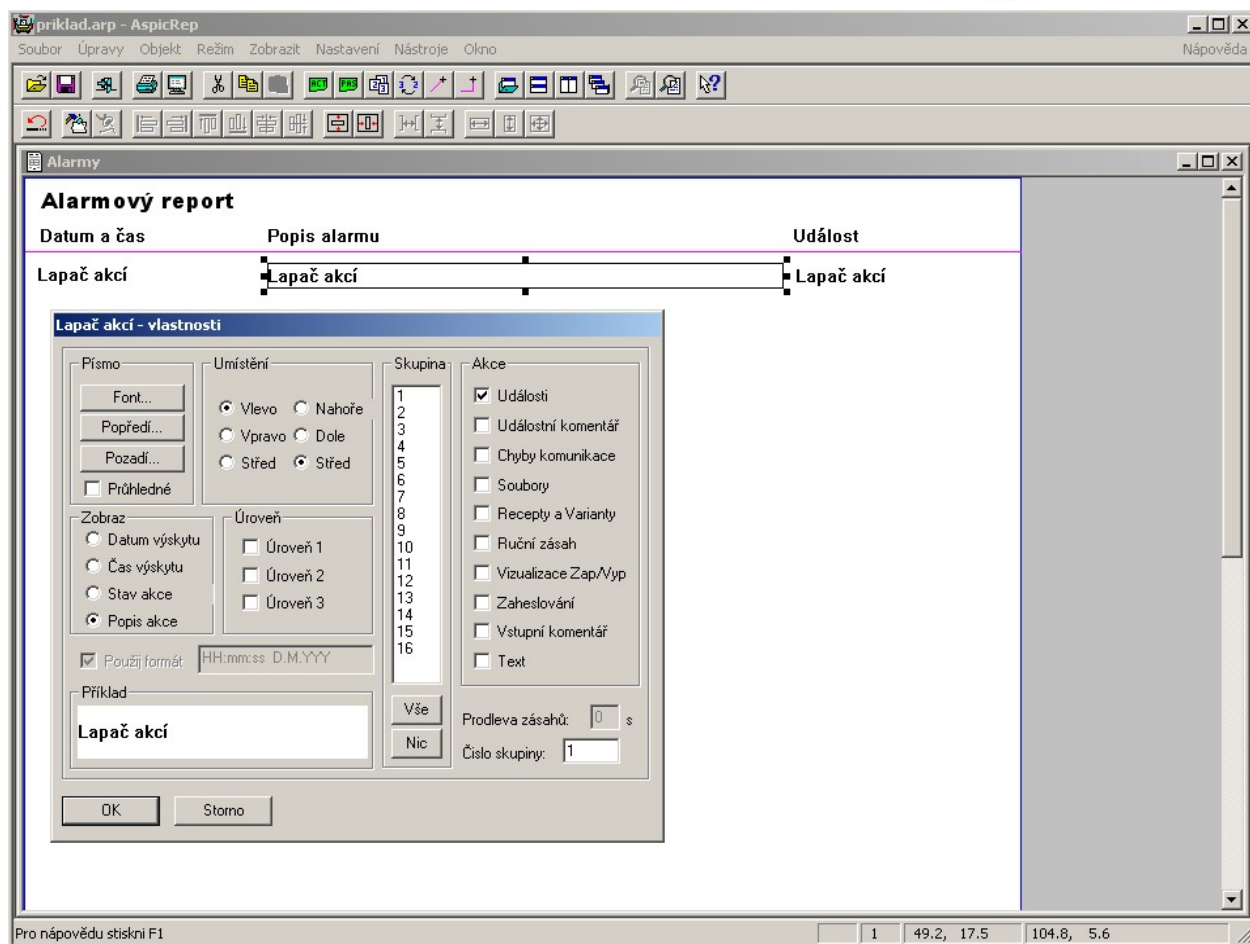



Nyní máme připraveno záhlaví a zápatí tiskového listu. Zbývá list naplnit daty. Nejvhodnějším objektem pro naplnění alarmového reportu je aktivní objekt „Lapač akcí“.

Aktivní objekty umístěné na listu je třeba navázat na proměnné, které jsou definovány v systémovém souboru Aspic 3.30. Abychom mohli používat proměnné, je třeba projekt reportů svázat se systémovým souborem. Vyberte z menu „Soubor -> Otevři systémový soubor“ a vyberte systémový soubor našeho projektu „priklad.asf“.

Obdobně jako je potřeba svázat projekt reportů se systémovým souborem Aspic 3.30, je nutné Aspic 3.30 svázat s projektovým souborem reportů. Přepněte se do aplikace Aspic 3.30 a vyberte položku menu „Soubor -> Připoj soubor s protokolem“. Vyberte soubor „priklad.arp“. Nyní je možno z aplikace Aspic 3.30 kliknutím na ikonu  spustit generování reportu. Můžete vyzkoušet, nicméně, report bude prázdný, bez dat. Na tiskovém listu bude zobrazena pouze hlava a pata.



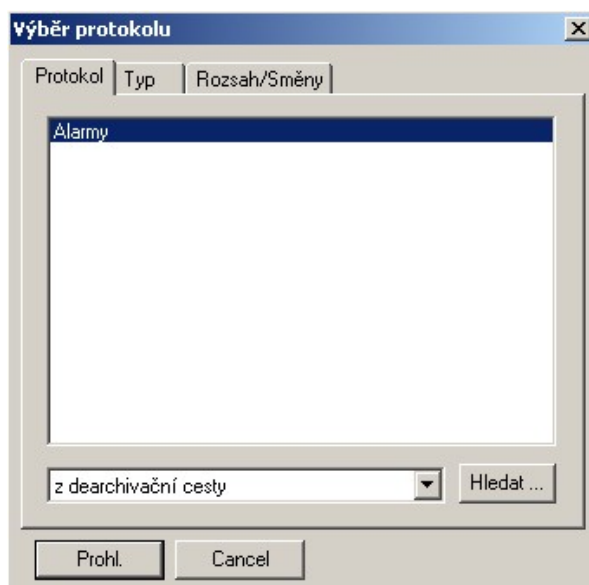
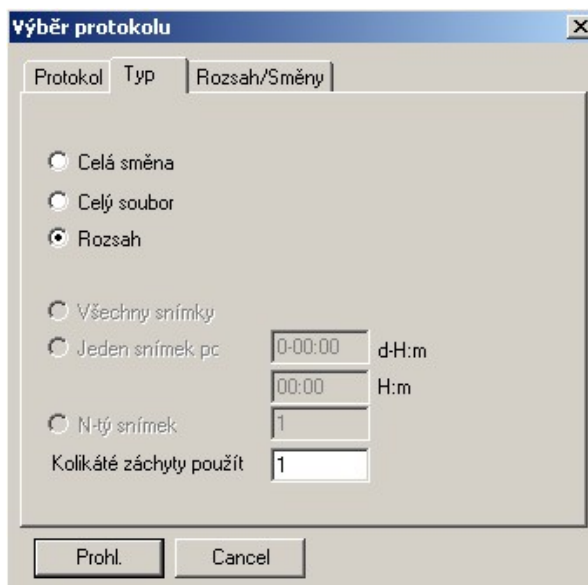
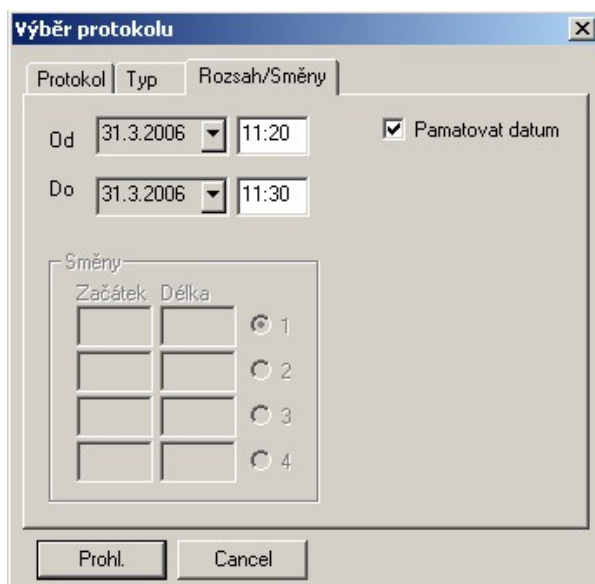


Uložte projekt reportů a ukončete aplikaci AspicRep. Přesunete se do aplikace Aspic 3.30 a klikněte na ikonu pro zobrazení reportů . Není podstatné, jestli je spuštěna vizualizace.

V otevřeném dialogu „Výběr protokolu“ zvolte požadovaný report (v současnosti máme pouze jeden, Alarmy) a vyberte, že ho chcete tvořit z dearchivační cesty (používáme archivaci do souboru Aspic 3.30).

Přesuňte se na další kartu dialogu „Typ“ a zvolte požadovaný typ protokolu. Pro ukázku si vytvoříme protokol, pro časový rozsah.

V další kartě dialogu zadejte časový rozsah, pro který má být report vytvořen. Nevolte jej příliš dlouhý (alarmů máme mnoho a pro ukázku plně postačuje období 10 minut). Časy volte s ohledem na to, kdy jste měli vizualizaci spuštěnou (alarmy musí pro dané období existovat v archivním souboru).

Směny	Začátek	Délka	
			<input checked="" type="radio"/> 1
			<input type="radio"/> 2
			<input type="radio"/> 3
			<input type="radio"/> 4

Na následující stránce je příklad vytvořeného Alarmového reportu.

Alarmový report

Datum a čas	Popis alarmu	Událost
11:27:11 31.3.2006	Hodnota5 - Alarm - Motor v poruše	Příchod
11:27:14 31.3.2006	Hodnota5 - Alarm - Motor v poruše	Konec
11:27:14 31.3.2006	Hodnota2 - Alarm - Horní kritická úroveň	Příchod
11:27:17 31.3.2006	Hodnota2 - Alarm - Horní kritická úroveň	Konec
11:27:20 31.3.2006	Hodnota1 - Alarm - Horní kritická úroveň	Příchod
11:27:23 31.3.2006	Hodnota1 - Alarm - Horní kritická úroveň	Konec
11:27:24 31.3.2006	HodnotaA - Alarm - Dolní kritická úroveň	Příchod
11:27:34 31.3.2006	HodnotaA - Alarm - Dolní kritická úroveň	Konec
11:27:38 31.3.2006	Hodnota2 - Alarm - Horní kritická úroveň	Příchod
11:27:41 31.3.2006	Hodnota2 - Alarm - Horní kritická úroveň	Konec
11:27:47 31.3.2006	Hodnota5 - Alarm - Motor v poruše	Příchod
11:27:47 31.3.2006	Hodnota1 - Alarm - Horní kritická úroveň	Příchod
11:27:50 31.3.2006	Hodnota1 - Alarm - Horní kritická úroveň	Konec
11:27:53 31.3.2006	Hodnota5 - Alarm - Motor v poruše	Konec
11:27:54 31.3.2006	HodnotaA - Alarm - Dolní kritická úroveň	Příchod
11:27:59 31.3.2006	Hodnota5 - Alarm - Motor v poruše	Příchod
11:28:02 31.3.2006	Hodnota5 - Alarm - Motor v poruše	Konec
11:28:02 31.3.2006	Hodnota2 - Alarm - Horní kritická úroveň	Příchod
11:28:04 31.3.2006	HodnotaA - Alarm - Dolní kritická úroveň	Konec
11:28:05 31.3.2006	Hodnota2 - Alarm - Horní kritická úroveň	Konec
11:28:08 31.3.2006	Hodnota5 - Alarm - Motor v poruše	Příchod
11:28:08 31.3.2006	Hodnota2 - Alarm - Horní kritická úroveň	Příchod
11:28:11 31.3.2006	Hodnota2 - Alarm - Horní kritická úroveň	Konec
11:28:14 31.3.2006	Hodnota5 - Alarm - Motor v poruše	Konec
11:28:14 31.3.2006	Hodnota2 - Alarm - Horní kritická úroveň	Příchod
11:28:17 31.3.2006	Hodnota2 - Alarm - Horní kritická úroveň	Konec
11:28:20 31.3.2006	Hodnota5 - Alarm - Motor v poruše	Příchod
11:28:23 31.3.2006	Hodnota5 - Alarm - Motor v poruše	Konec
11:28:24 31.3.2006	HodnotaA - Alarm - Dolní kritická úroveň	Příchod
11:28:29 31.3.2006	Hodnota2 - Alarm - Horní kritická úroveň	Příchod
11:28:32 31.3.2006	Hodnota2 - Alarm - Horní kritická úroveň	Konec
11:28:35 31.3.2006	Hodnota2 - Alarm - Horní kritická úroveň	Příchod
11:28:35 31.3.2006	HodnotaA - Alarm - Dolní kritická úroveň	Konec
11:28:38 31.3.2006	Hodnota2 - Alarm - Horní kritická úroveň	Konec
11:28:47 31.3.2006	Hodnota5 - Alarm - Motor v poruše	Příchod
11:28:47 31.3.2006	Hodnota2 - Alarm - Horní kritická úroveň	Příchod
11:28:50 31.3.2006	Hodnota5 - Alarm - Motor v poruše	Konec
11:28:53 31.3.2006	Hodnota5 - Alarm - Motor v poruše	Příchod

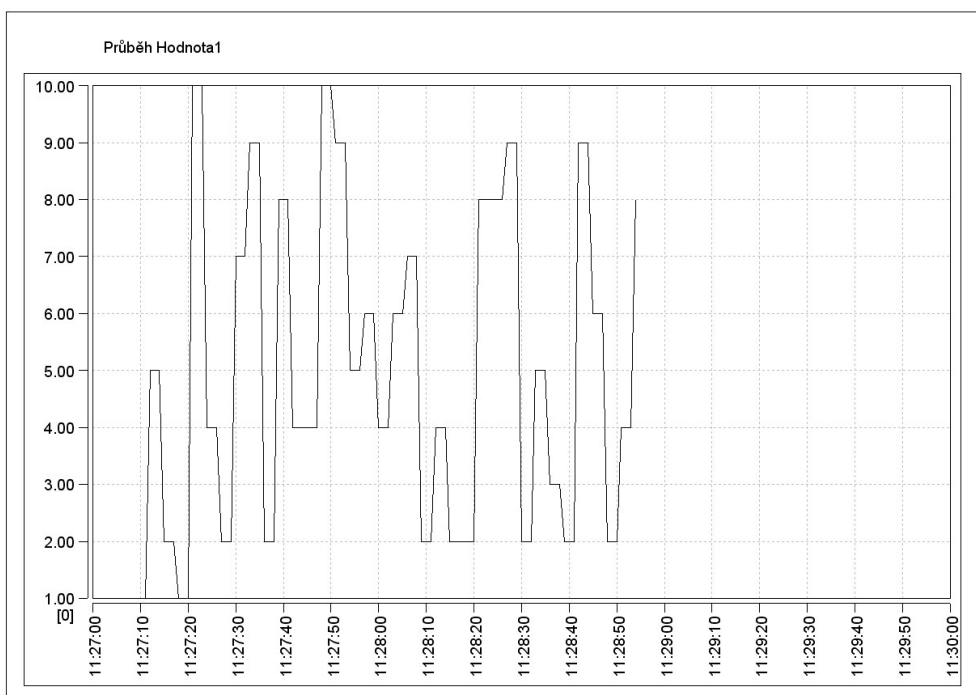
38. Dle požadavků na ukázkový projekt je potřeba vytvořit mimo alarmového reportu ještě grafický report, který bude zobrazovat průběh proměnné „Stanice_Register_Hodnota1“ a „HodnotaA“ za uživatelem zadané časové období. Dále pak report (tabulku) minutových hodnot proměnné „Stanice_Register_Hodnota3“.

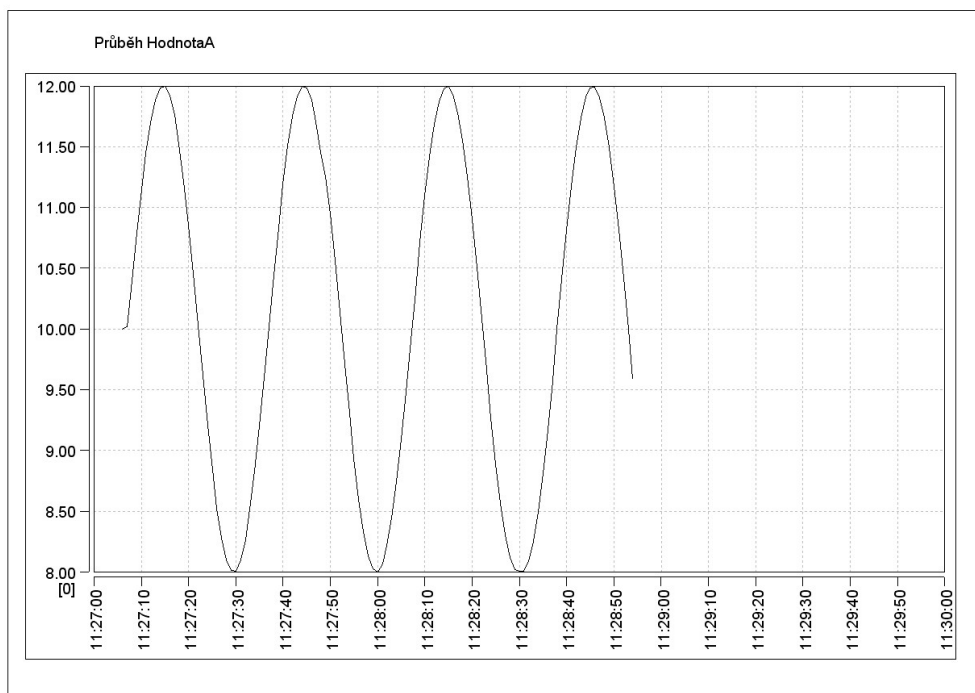
Spustíte aplikaci AspicRep a ve stávajícím projektu reportů přidejte nový report „Okno->Okno“, „Přidej“. Report pojmenujte „Hodnota1f“, hlavičku ani patu nenastavujte (ponechte nulovou).

Nastavte orientaci listu „na ležato“ a na list umístěte aktivní objekt „Plný graf“. Roztáhněte jej po ploše listu a dvojklikem na něj zobrazte dialog „Plný graf – vlastnosti“. Přenastavte vlastnost „Perioda“ na 00:00:01 (v příkladu data budeme načítat po vteřině, v praxi musíte zvolit jinou, technologii odpovídající hodnotu). Dále proveďte napojení na proměnnou „Stanice_Register_Hodnota1“. Můžete změnit vlastnosti týkající se barev a fontů dle libosti. Je dokonce možné, zobrazit do jednoho grafu více průběhů. V našem příkladu necháme průběh jeden.

Dále na list přidejte pasivní objekt „Text“ kterým list pojmenujete. Můžete přidat i další objekty „Text“ pro popisy a jednotky os grafu.

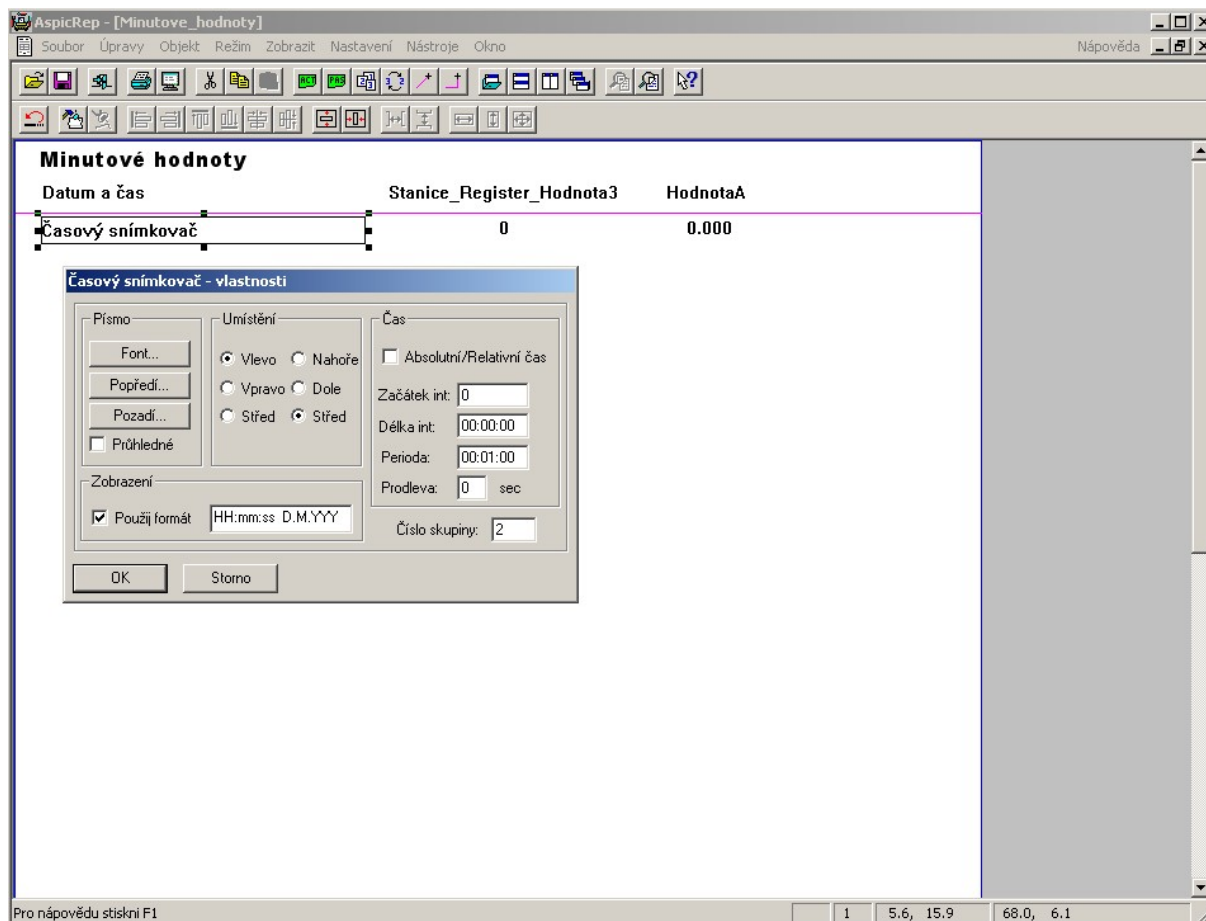
Uložte projekt reportů a ukončete aplikaci AspicRep. Přesuňte se do aplikace Aspic 3.30 a zobrazte si vytvořený report obdobně jako předchozí alarmový report.





Poslední požadovaným reportem jsou minutové hodnoty proměnné „Stanice_Register_Hodnota3“. Přidejte další report, pojmenujte jej „Minutove_hodnoty“. Záhloví a zápatí vytvořte stejné jako v případě alarmového reportu, jen vyměňte texty. Použijte aktivní objekt „Časový snímkořač“ s nastavením uvedeným na obrázku. Pro zobrazení hodnoty proměnné použijte objekt „Závislá proměnná“ který napojte na odpovídající proměnnou.

Nezapomeňte všechny aktivní objekty umístit do jedné skupiny.



Vyzkoušejte vytvoření reportu. Volte delší časové období, vypisujeme minutové hodnoty. Je nutno zvolit takové časové období, za které jsou v archivním souboru uložena data, jinak by byl vygenerovaný report prázdný.

Minutové hodnoty			
Datum a čas		Stanice_Register_Hodnota3	HodnotaA
12:00:00	30.3.2006	35	8.127
12:01:00	30.3.2006	31	8.142
12:02:00	30.3.2006	32	8.184
12:03:00	30.3.2006	30	8.018
12:04:00	30.3.2006	34	8.000
12:05:00	30.3.2006	30	8.003
12:06:00	30.3.2006	30	8.000
12:07:00	30.3.2006	35	8.004
12:08:00	30.3.2006	33	8.037
12:09:00	30.3.2006	33	8.047
12:10:00	30.3.2006	32	8.032
12:11:00	30.3.2006	33	8.026
12:12:00	30.3.2006	35	8.015
12:13:00	30.3.2006	30	8.149
12:14:00	30.3.2006	34	8.156
12:15:00	30.3.2006	31	8.124
12:16:00	30.3.2006	35	8.059
12:17:00	30.3.2006	30	8.044
12:18:00	30.3.2006	31	8.222
12:19:00	30.3.2006	32	8.176
12:20:00	30.3.2006	34	8.139
12:21:00	30.3.2006	35	8.106
12:22:00	30.3.2006	34	8.173
12:23:00	30.3.2006	32	8.446
12:24:00	30.3.2006	34	8.429
12:25:00	30.3.2006	34	8.382
12:26:00	30.3.2006	30	8.309
12:27:00	30.3.2006	34	8.226
12:28:00	30.3.2006	32	8.537
12:29:00	30.3.2006	33	8.496
12:30:00	30.3.2006	31	8.462
12:31:00	30.3.2006	30	8.413
12:32:00	30.3.2006	33	8.367
12:33:00	30.3.2006	30	8.940
12:34:00	30.3.2006	30	8.955
12:35:00	30.3.2006	32	8.874
12:36:00	30.3.2006	33	8.796
12:37:00	30.3.2006	34	8.655
12:38:00	30.3.2006	30	9.084
12:39:00	30.3.2006	35	9.008
12:40:00	30.3.2006	33	8.948
12:41:00	30.3.2006	34	8.903
12:42:00	30.3.2006	34	8.839
12:43:00	30.3.2006	34	9.283
12:44:00	30.3.2006	30	9.283
12:45:00	30.3.2006	35	9.457
12:46:00	30.3.2006	35	9.382
Úspěšná firma, s.r.o.		1	16:07:35 31.3.2006

Poznámka: Do reportu můžete přidat více proměnných. Do tohoto reportu jsem přidal ještě minutové hodnoty proměnné „HodnotaA“.