



## 1.1 Obsah

1.1 Obsah .....	2
<b>Instalace, konfigurace a prostředí programu.....</b>	<b>3</b>
2.1 Požadavky na hardware a operační systém.....	3
2.2 Instalace a konfigurace programu .....	3
2.3 Hlavní okno, pracovní plocha, nástrojový panel, stavový řádek.....	3
2.4 Používané datové soubory .....	4
2.5 Přehled příkazů hlavního menu .....	4
2.6 Nastavení programu.....	6
2.7 Nástroje.....	6
<b>Aplikace v prostředí APS Config .....</b>	<b>9</b>
3.1 Struktura a vlastnosti aplikace .....	9
3.2 Moduly.....	10
3.3 Událost a makro .....	11
3.4 Script.....	12
3.5 Překlad programu, přenos do paměti řídicího modulu.....	13
<b>Programování systému .....</b>	<b>14</b>
4.1 Standardní funkce systému .....	14
4.2 Makra.....	16
<b>Přílohy .....</b>	<b>22</b>
5.1 Moduly systému APS 400 .....	22
5.2 Zařízení systému APS 400.....	26
5.3 Přehled nejdůležitějších konstrukcí programovacího jazyka .....	34
5.4 Přehled systémových archivních značek.....	37
5.5 Ucelený příklad aplikace systému APS 400 .....	39
5.6 Programovací protokol .....	42

## 2

## Instalace, konfigurace a prostředí programu

Program APS 400 Config je určen pro uživatelskou konfiguraci identifikačního systému APS 400. Obsahuje nástroje pro jeho naprogramování, jednoduchou diagnostiku a vizualizaci všech jeho stavů, včetně možnosti on-line ovládání výstupů a vybraných logických stavů.

### 2.1 Požadavky na hardware a operační systém

Program je určen pro operační systémy Microsoft Windows NT 4.0 (SP4), a vyšší (Windows 2000, Windows XP). Pro efektivní práci je nutné hardwarové vybavení, výkonově dobře vyhovující použitému operačnímu systému. Pro komfortní práci s programem je vhodný monitor s rozlišením alespoň 1024x768 bodů. Požadavky na prostor na pevném disku jsou minimální.

### 2.2 Instalace a konfigurace programu

Program je k dispozici zdarma. Odkaz na aktuální instalační soubory je umístěn na stránce [http://www.techfass.cz/aps\\_400\\_dn\\_cz.html](http://www.techfass.cz/aps_400_dn_cz.html), na objednávku jej lze dodat i na disku CD ROM nebo jiném médiu. Po spuštění instalačního programu (setup.exe) se, prosím, řiďte jeho pokyny. Po instalaci není nutná žádná další konfigurace. Konfigurační soubor (APSConfig.ini) se ukládá při ukončení programu (APSConfig.exe). Dostanete-li se při nastavování nejrůznějších parametrů prostředí do situace, kdy je vhodné použití výchozí konfigurace, stačí program ukončit, smazat konfigurační soubor a opět program spustit.

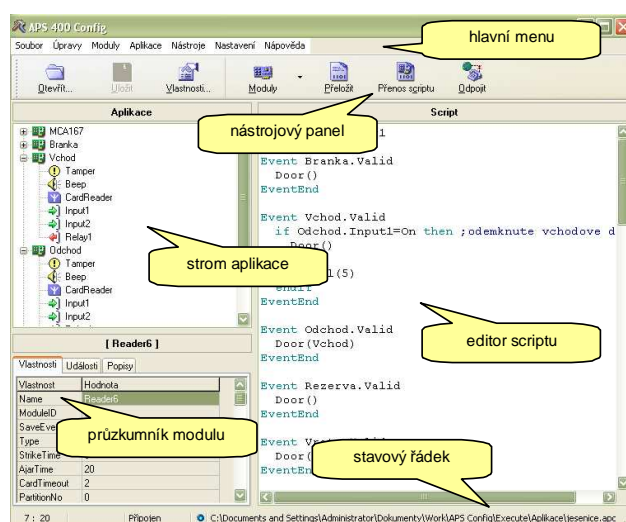
### 2.3 Hlavní okno, pracovní plocha, nástrojový panel, stavový řádek

Hlavní okno programu APS Config (*obr. 2.1*) obsahuje standardní prvky, obvyklé u obdobných programů (menu, nástrojový panel, stavový řádek,...).

Pracovní plocha hlavního okna programu je rozdělena do tří částí:

- *Strom aplikace* ... popisuje hardwarovou strukturu aplikace,
- *průzkumník modulu* ... zobrazuje a umožňuje editaci vlastností objektu vybraného v aplikačním stromu,
- *editor scriptu* ... textový editor pro tvorbu uživatelského programu (*scriptu*).

Všechny tři základní části jsou funkčně provázány, takže změny v jedné z nich se automaticky promítají v obou zbývajících.



Obr. 2.1: Hlavní okno programu

## 2.4 Používané datové soubory

Zdrojový soubor aplikace se ukládá do jediného souboru s příponou `.apc`. Při každém otevření souboru aplikace se automaticky vytváří záložní soubor se stejným názvem a příponou doplněnou o „.bak“. Po každém úspěšném překladu programu se v adresáři, v němž je uložen zdrojový soubor, vytváří konfigurační soubor pro administrační program. Konfigurační soubor má identický název jako soubor zdrojový, ale příponu „.cfg“.

## 2.5 Přehled příkazů hlavního menu

Přehled všech příkazů umístěných v hlavním menu programu popisuje tabulka na *obr. 2.3*. Aby byl seznam příkazů kompletní, jsou v tabulce na *obr. 2.2* uvedeny i příkazy, které nejsou obsaženy v hlavním menu (mají pouze klávesovou zkratku). Význam řady příkazů odpovídá běžným zvyklostem (práce se soubory, úpravy textu, nápověda,...); v dalším textu budou podrobněji popsány jen příkazy specifické pro tento program.

Název	zkratka	popis
Inteligentní vkládání	Ctrl+mezera	používá se při psaní scriptu – po stisku klávesové zkratky se zobrazí nabídka modulů a zařízení, které lze na daném místě scriptu použít
Rychlé konstrukce	Ctrl+J	používá se při psaní scriptu – po stisku klávesové zkratky se zobrazí nabídka předdefinovaných částí scriptu, které se vloží do editoru
Nalezení modulu	Ctrl+M	používá se při psaní scriptu – po stisku klávesové zkratky se ve stromu aplikace vybere modul, jemuž přísluší právě editované makro
Odsazení bloku doprava	Shift+Ctrl+I	odsadí všechny vybrané řádky o dvě mezery doprava
Posun bloku doleva	Shift+Ctrl+U	odsadí všechny vybrané řádky o dvě mezery doleva (je-li to možné)

*Obr. 2.2: Přehled příkazů, které nejsou obsaženy v hlavním menu programu*

název menu / název příkazu		zkratka	popis
<b>Soubor</b>		<b>Alt+S</b>	<b>práce se soubory</b>
	Nový	-	vytvoří nový prázdný soubor
	Otevřít	-	otevře soubor s aplikací
	Uložit	-	uloží aktuální soubor s aplikací
	Uložit jako	-	uloží aktuální soubor s aplikací do nového souboru
	Tisk	-	vytiskne programovací protokol
	Konec	Alt+F4	ukončí program APS Config
	Naposledy otevřené soubory	-	4 příkazy pro rychlé otevření naposledy editovaných souborů
<b>Úpravy</b>		<b>Alt+A</b>	<b>úpravy textu v editoru scriptu</b>
	Zpět	Alt+←	vrátí zpět poslední změnu v editoru scriptu (je-li to možné)
	Vymout	Ctrl+X	vyjme označený text a zkopíruje ho do schránky
	Kopírovat	Ctrl+C	zkopíruje označený text do schránky
	Vložit	Ctrl+V	vloží text ze schránky
	Odstranit	Del	odstraní označený text
	Vybrat vše	Ctrl+A	označí celý obsah editoru scriptu
	Najít	-	zobrazí dialog pro hledání textu v editoru scriptu
	Nahradit	-	zobrazí dialog pro nahrazování textu v editoru scriptu
	Převést na komentář	Ctrl+K	převede označené řádky na komentáře
	Odstranit komentář	Ctrl+L	převede řádky označené jako komentáře zpět
	Sbalit aplikační strom	Ctrl+B	sbalí všechny uzly aplikačního stromu
	Seřadit makra	-	seřadí všechna makra v editoru scriptu podle pořadí v aplikačním stromu

*Obr. 2.3: Přehled příkazů hlavního menu programu*

Moduly		Alt+M	práce s moduly (řídící modul, síťové moduly, zdroje,...)
	Načíst moduly ze systému	-	v online režimu vytvoří strom aplikace podle skutečného stavu systému
	Standardní funkce	-	zobrazí konfigurační dialog pro definici standardní funkce modulu vybraného ve stromu aplikace
	Smazat	-	smaže modul vybraný ve stromu aplikace
	... další položky menu	-	seznam modulů, které lze do aplikace vložit
Aplikace		Alt+E	práce s aplikací
	Přeložit	F9	přeloží aktuální aplikaci do binárního kódu
	Přenos scriptu	Alt+C	přenesení přeloženého binárního kódu do řídicího modulu, pokud je aplikace změněna provede se nejdříve její překlad
	Vlastnosti	Alt+V	zobrazí dialog pro definici vlastností aplikace
	Připojit/Odpojit	-	připojí, resp. ukončí připojení komunikačního serveru k řídicímu modulu
	Změnit heslo aplikace	-	zobrazí dialog pro změnu připojovacího hesla v řídicím modulu
Nástroje		Alt+R	diagnostika a vizualizace
	Načíst přístupové karty	-	načte čísla přístupových karet z paměti řídicího modulu
	Načíst přístupové skupiny	-	načte definici přístupových skupin z paměti řídicího modulu
	Načíst časové plány	-	načte definici časových plánů z paměti řídicího modulu
	Načíst tabulku svátků	-	načte definici tabulky svátků z paměti řídicího modulu
	Parametry aplikace	-	načte parametry aplikace z paměti řídicího modulu
	Vlastnosti řídicího modulu	-	načte vlastnosti řídicího modulu
	Stav sítě	-	spustí vizualizaci stavu komunikace na sběrnici RS 485
	Stav řídicího modulu	-	spustí vizualizaci stavů řídicího modulu
	Stav síťové čtečky	-	spustí vizualizaci stavů vybraného síťového modulu
	Stav síťového zdroje	-	spustí vizualizaci stavů vybraného síťového zdroje
	Vyčítat archiv událostí	-	spustí vyčítání archivu událostí řídicího modulu
	Stav registrů	-	spustí vizualizaci registrů řídicího modulu
	Stav časovačů	-	spustí vizualizaci softwarových časovačů řídicího modulu
	Oblasti EZS	-	spustí vizualizaci stavu oblastí EZS
	Smyčky a příznaky EZS	-	spustí vizualizaci stavů smyček a příznaků EZS
	Nahrát servisní nastavení	-	nahrádá servisní nastavení do řídicího modulu
	Nastavit adresu čtecího modulu	-	otevře dialog pro nastavení adresy síťového modulu
Nastavení		Alt+N	nastavení parametrů programu
	Rychlé konstrukce	-	definice rychlých konstrukcí
	Komunikační servery	-	nastavení cest k zákaznickým komunikačním serverům
	Nástroje a prostředí	-	nastavení obecných parametrů programu (servisní karty, nástroje, editor zdrojového kódu)
	Tiskové sestavy	-	nastavení parametrů tiskových sestav
	Otevřít scénář	-	Otevře scénář vizualizačních oken a zobrazí je na monitoru
	Uložit scénář	-	Uloží scénář vizualizačních oken
Nápověda		-	práce s nápovědou
	Obsah	F1	zobrazí obsah nápovědy
	Programování	-	zobrazí kapitolu s popisem základů programování systému
	O aplikaci APS Config	-	zobrazí základní informace o programu

Obr. 2.3: Přehled příkazů hlavního menu programu (pokračování)

### 2.6 Nastavení programu

Nastavením programu APS 400 Config je možné ovlivnit jak jeho vzhled, tak i chování při některých operacích. Dialogy pro veškerá nastavení se otevírají z menu *Nastavení*.

- Rychlé konstrukce jsou předdefinované části textu, které je možné vkládat do editoru po stisku klávesové zkratky CTRL+J. Dialog obsahuje seznam konstrukcí a tlačítka pro jejich vytvoření, editaci a smazání. Pro každou konstrukci se definují dva textové parametry: titulek nabídky a vkládaný text (obr. 2.4). Titulek nabídky je libovolný text, který se bude zobrazovat v nabídce pro vložení. Vkládaný text je text, který se po výběru v nabídce do editoru vloží. Znakem „%“ lze definovat pozici kam má být po vložení umístěn kurzor.
- Standardní komunikační servery jsou součástí instalace programu. Popis nastavení a konfigurace nestandardních komunikačních modulů je součástí jejich dokumentace.
- Dialog *Nástroje a prostředí* obsahuje 3 záložky, umožňující zadat:
  - kódy servisních přístupových karet a časových plánů,
  - kód karty, která bude použita pro potvrzení nastavení adresy čtecího modulu (u modulů, které tento způsob nastavení podporují) a počet vyčítaných karet při diagnostice paměti řídicího modulu,
  - barvy, které se mají používat v editoru zdrojového kódu.
- Nastavení tiskových sestav umožňuje do hlavičky programovacího protokolu vložit obrázek s firemním logem a čtyři řádky textu. Aby při tisku nedošlo ke zkreslení obrázku vkládejte jen obrázky ve formátu bmp, s poměrem stran 2:1.
- Občas je užitečné uložit aktuální nastavení pozic vizualizačních oken pro jejich pozdější obnovení. K tomu slouží příkazy *Uložit scénář* a *Otevřít scénář* v menu *Nastavení*. Podmínkou pro práci se scénáři je navázaná komunikace s řídicím modulem.



Obr. 2.4: Definice rychlé konstrukce

### 2.7 Nástroje

Program APS 400 Config obsahuje řadu nástrojů určených pro usnadnění tvorby aplikací systému APS 400. Jejich použití vyžaduje navázanou komunikaci s řídicím modulem (podrobněji viz. kapitola 3.1: „Struktura a vlastnosti aplikace“). Funkčně je lze rozdělit do tří kategorií:

- nástroje pro *kontrolu obsahu paměti řídicího modulu*,
- nástroje pro *vizualizaci stavů systému*,
- nástroje pro nastavení *parametrů systému* (servisní karty, HW adresy některých typů síťových modulů).

#### *Nástroje pro kontrolu obsahu paměti řídicího modulu*

Vyčíst lze tabulky přístupových karet, přístupových skupin, časových plánů, svátků, parametrů aplikace a vlastností řídicího modulu. Korektní vyčtení tabulky parametrů je podmíněno souladem mezi aplikací načtenou v programu a aplikací nahanou v řídicím modulu.

Po volbě požadovaného příkazu jsou vyslány požadavky komunikačnímu serveru a je zobrazen dialog, který se postupně plní získanými daty. Každý obsahuje tlačítka *OK* a *Storno*, po jejichž stisku se uzavře. Po stisku tlačítka *OK* jsou navíc vyčtená data uložena do textového souboru, v němž každému řádku tabulky odpovídá jeden řádek textu a jednotlivé sloupce jsou odděleny středníky.



Soubory jsou ukládány do adresáře, v němž je uložen spustitelný soubor APSConfig.exe a jejich pojmenování je následující:

- ReportCards.txt pro přístupové karty,
- ReportPlans.txt pro časové plány,
- ReportHolidays.txt pro svátky,
- ReportParams.txt pro parametry aplikace,
- ReportGroups.txt a ReportIDS.txt pro přístupové skupiny a odpovídající autorizační příznaky,
- ReportProperties.txt pro parametry řídicího modulu.

K vyčítání přístupových skupin je nutné doplnit několik detailů:

Přístupové oprávnění je definováno hodnotou uloženou na průsečku čísla skupiny, která je přístupové kartě přiřazena a adresy síťového modulu, pro který je přístupové oprávnění definováno. Na každém z průsečků může být definováno buď generální povolení (znak „A“ v zeleném poli) nebo odepření (znak „N“ v červeném poli) přístupu, nebo přístup podle časového plánu (číslo plánu ve žlutém poli), viz. *obr. 2.5*.

Není-li přístup generálně odepřen, lze definovat i tzv. autorizační příznak (může mít význam oprávnění k ovládání ústředny EZS). V případě, že je autorizační příznak definován, zobrazí se v pravém horním rohu políčka černý trojúhelník. V textovém souboru ReportIDS.txt pak jsou tyto hodnoty uloženy jako „A“ – autorizováno a „N“ – neautorizováno.

Přístupové skupiny											
GRP	USR	1	2	3	4	5	6	7	8	9	10
1	1	A	A	A	A	A	A	A	A	A	A
2	2	N	N	N	N	N	N	N	N	N	N
3	3	1	1	1	1	1	1	1	1	1	1
4	0	N	N	N	N	N	N	N	N	N	N
5	0	N	N	N	N	N	N	N	N	N	N
6	0	N	N	N	N	N	N	N	N	N	N

*Obr. 2.5: Přístupové skupiny*

### Nástroje pro vizualizaci stavů systému

Nástroje pro vizualizaci stavů systému umožňují online sledování stavů všech vstupů, výstupů a logických stavů jednotlivých systémových modulů, jejich vzdálené ovládání, aktivaci uživatelských funkcí a částečně i řízení komunikace na sběrnici APS Bus. Všechna vizualizační okna jsou plovoucí a mohou být otevřena současně. S programem APS 400 Config přitom lze dále pracovat.

Základním diagnostickým nástrojem je sledování stavu sítě (*obr. 2.6*). Každý síťový modul je reprezentován jednou ikonou, která vyjadřuje stav komunikace s ním. Šedou barvou jsou zobrazeny moduly (dle HW adresy), které nejsou připojeny, zelenou barvou moduly s nimiž se normálně komunikuje a konečně barvou červenou ty, s nimiž se řídicí modul pokouší komunikaci navázat. Po kliknutí pravým tlačítkem myši na libovolný modul se zobrazí lokální menu s následujícími příkazy:

- *Ukázat podrobně ...* zobrazí okno s podrobnou vizualizací stavů síťového modulu.
- *Připojit modul ...* předá řídicímu modulu příkaz k pokusu o navázání komunikace s modulem.
- *Odpojit modul ...* předá řídicímu modulu příkaz k odpojení modulu.

Podrobné sledování stavů síťového modulu lze spustit dvojím způsobem. Buď, jak je popsáno výše, z okna pro sledování stavu sítě, nebo příkazem *Stav síťové čtečky* z menu *Nástroje*. Stavové okno (*obr. 2.7*) obsahuje (zleva):

- ikonu pro sledování stavu komunikace s modulem (se stejným významem jako u zobrazení stavu sítě),
- vstup TAMPER (žlutá = narušení),
- čtecí zařízení (šedá = neaktivní, zelená = platné ID, modrá = neplatné ID a červená neznámé ID),
- logické stavy “vyražené dveře” a “dlouho otevřené dveře”,
- oba vstupy a oba výstupy (zelená = rozepnut, červená = sepnut).

Sledování stavů síťových zdrojů lze otevřít stejně, jako sledování stavů síťových modulů. Stavové okno (*obr. 2.8*) obsahuje (zleva):

- ikonu pro sledování stavu komunikace s modulem (se stejným významem jako u zobrazení stavu sítě),
- vstupu TAMPER (žlutá = narušení),
- stav AC (zelená = připojeno, šedá = odpojeno),
- stav baterie (zelená = v pořádku, červená = vybitá, šedá = odpojena),
- stav výstupu DC (zelená = v pořádku, červená = přetížen / zkrat, černá = vypnut).



*Obr. 2.6: Stav sítě*



*Obr. 2.7: Síťový modul*



*Obr. 2.8: Stav zdroje*

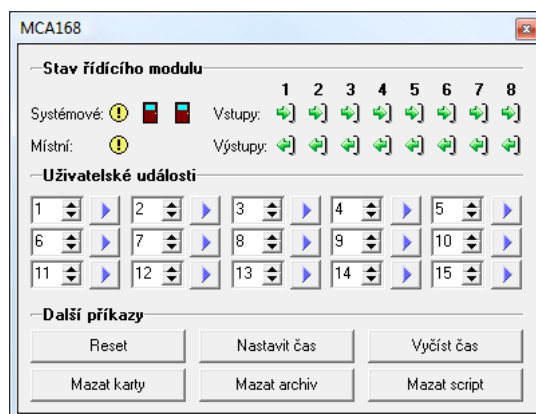
Pro podrobné zobrazení stavů řídicího modulu slouží okno *Stav řídicího modulu*. Část plochy okna je opět tvořena ikonami reprezentujícími vstupy a výstupy modulu a globální logické stavy systému, zbytek obsahuje tlačítka, jejichž význam je zřejmý z jejich popisu.

Ve všech případech lze výstupy ovládat z lokálního menu, dostupného po kliknutí na odpovídající ikonu (obsahuje dvojici příkazů *Sepnout* a *Rozepnout*).

Dalšími nástroji tohoto typu jsou: Okno pro sledování stavů *oblastí* připojeného systému EZS NX-8, kde lze z lokálních menu ovládat stavy střežení jednotlivých oblastí (*obr. 2.9*) a okno pro sledování stavů smyček a příznaků EZS, kde lze navíc sledovat vstupy a výstupy osazené na komunikačním interface IFNIO 844.

Zvláštní postavení mají okna pro vizualizaci stavů *registrů* a *softwarových časovačů*. Aby bylo zobrazení korektní, je nutné mít v programu APS Config přeloženou aplikaci přesně odpovídající aplikaci nahané v řídicím modulu. Aktuální hodnoty registrů i časovačů lze vzdáleně nastavovat.

Posledním nástrojem z této kategorie je online *vyčítání archivu událostí*. Předpokladem přehledného zobrazení je opět přeložená aplikace odpovídající programu v řídicím modulu.



**Obr. 2.9: Stavový panel řídicího modulu**



**Obr. 2.10: Stavy oblastí EZS**

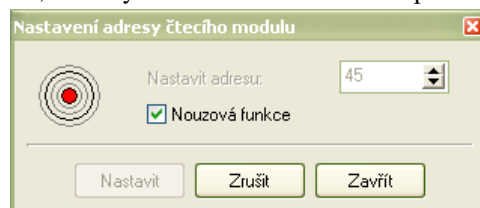
### Nástroje pro nastavení parametrů systému

*Nahrání servisního nastavení řídicího modulu* slouží k usnadnění ladění systému. Do paměti řídicího modulu přenese přístupová oprávnění definovaná v nastavení nástrojů – pro odladění aplikace tedy nutné instalovat server a administrační program.

*Nastavení adresy síťového modulu* (*obr. 2.11*) se používá u modulů, u kterých nelze adresu nastavit pomocí klávesnice, propojkami či přepínači (např. NREM 61).

Do vstupního pole zadejte požadovanou adresu a stiskněte tlačítko „Nastavit“. Poté je nutné vzít kartu, která je v konfiguraci nástrojů zadána jako nastavovací a načíst její kód na příslušné čtečce. Akceptování adresy modul potvrdí dlouhým pípnutím.

Nenastavujte adresu, která již je na sběrnici připojena.



**Obr. 2.11: Nastavení adresy síťového modulu**



## 3

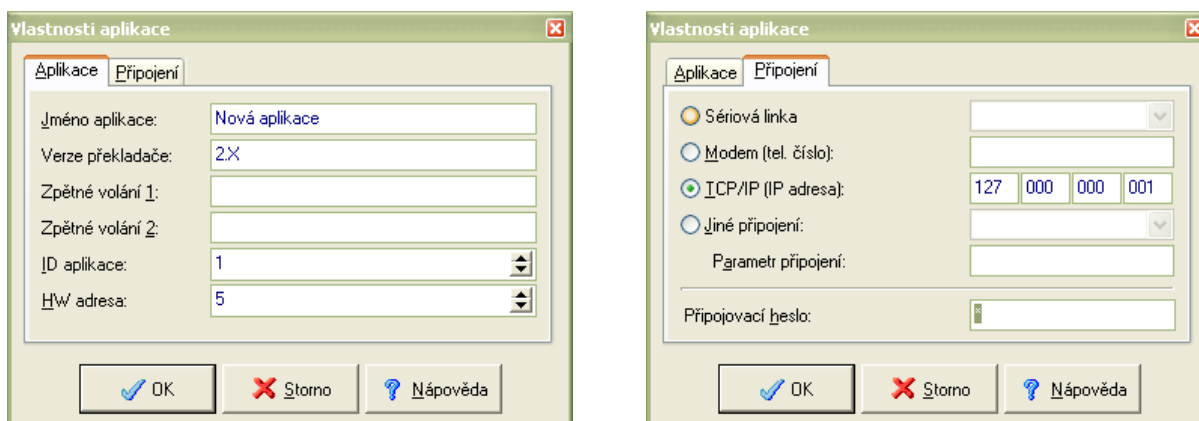
## Aplikace v prostředí APS Config

## 3.1 Struktura a vlastnosti aplikace

Aplikace v prostředí APS 400 Config obsahuje popis hardwarové struktury, nastavení vlastností jednotlivých modulů a skript, podle kterého má systém pracovat.

Hardwarová struktura aplikace je reprezentována stromem aplikace. Součástí každé aplikace je jeden řídicí modul (MCA 167(8)), max. 64 síťových modulů, max. 8 systémových zdrojů napájecího napětí (PSU 71) a ústředna EZS (NX-8). Jednotlivé moduly se do stromu aplikace vkládají z nabídky *Moduly*, nebo z lokálního menu dostupného po kliknutí pravým tlačítkem na stroměček. Mazat je lze příkazem *Smazat modul* z téže nabídky.

Vlastní aplikace má několik obecných vlastností, které lze zadat v dialogu *Vlastnosti aplikace* (obr. 3.1). Obsahuje dvě záložky, jednu pro definici obecných vlastností aplikace (*Aplikace*) a druhou pro nastavení způsobu komunikace s řídicím modulem (*Připojení*), zobrazí se příkazem *Vlastnosti* z nabídky *Aplikace*.



Obr. 3.1: Vlastnosti aplikace

Obecnými vlastnostmi aplikace jsou:

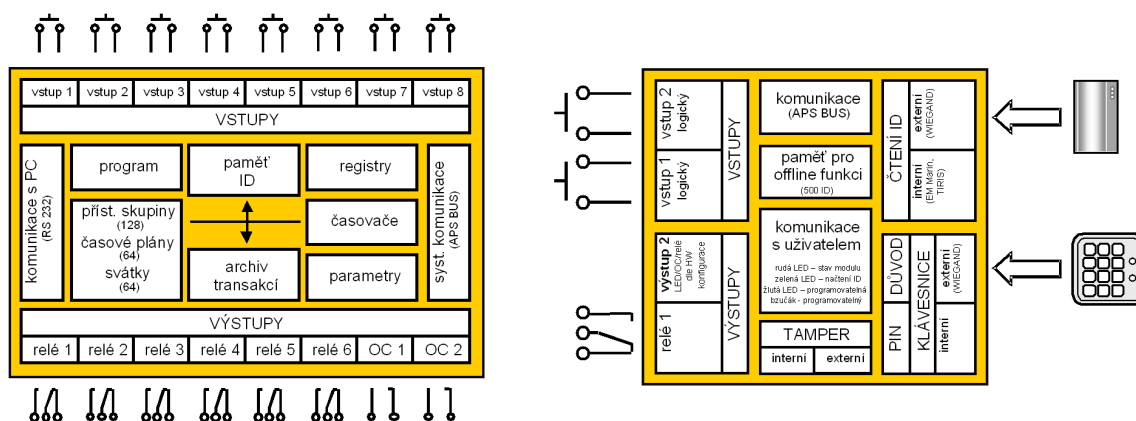
- *Jméno aplikace* ... libovolný text, max. 30 znaků,
- *verze překladače* ... číslo verze překladače pro kterou je aplikace vytvořena, tento údaj nelze měnit,
- *zpětné volání 1* a *zpětné volání 2* ... je-li aplikace připojena pomocí modemu pak lze v programu použít příkaz pro volání nadřazeného systému (PC) na tato čísla,
- *ID aplikace* ... má zvláštní význam při modemovém připojení, kromě speciálních případů (zpětné volání modemem) není nutné jeho hodnotu nastavovat,
- *HW adresa řídicího modulu* ... nastavení HW adresy řídicího modulu.

V záložce s vlastnostmi připojení je nutné definovat způsob komunikace s řídícím modulem, připojení modulu MCA 168 je možné pouze přes nastavení TCP/IP připojení ke komunikačnímu serveru APS 400 nServer.NET:

- *Sériová linka* ... komunikace prostřednictvím rozhraní RS 232, parametrem připojení je použitý komunikační port.
- *Modem* ... komunikace prostřednictvím modemu, parametrem modemového připojení je telefonní číslo linky ke které je připojen řídicí modul.
- *TCP/IP* ... komunikace prostřednictvím komunikačního serveru a TCP/IP sítě (nejčastější způsob komunikace), parametrem připojení je IP adresa serveru.
- *Jiné připojení* ... řídicí modul je připojen jinak. Popis nastavení nestandardních způsobů komunikace je obsažen v dokumentaci k příslušným komunikačním programům.
- *Připojovací heslo* je číslo z intervalu <0;99999999>, jehož hodnota musí odpovídat heslu uloženému v paměti řídicího modulu. Výchozí hodnota připojovacího hesla je 0, tato hodnota se v řídicím modulu nastaví po jeho restartu se zapnutým přepínačem SW konfigurace č. 10 (viz. katalogový list řídicího modulu). Změnu připojovacího hesla je možné provést po úspěšném navázání komunikace s řídicím modulem a volbou příkazu *Změna hesla* z nabídky Aplikace.

### 3.2 Moduly

Každý hardwarový modul je ve stromu aplikace reprezentován objektem, který obsahuje vstupy, výstupy a další obvody. Obecná struktura řídicího a síťového modulu je zřejmá z blokových schémat na *obr. 3.2*.



*Obr. 3.2: Bloková schémata řídicího a síťového modulu*

Každý modul i zařízení má řadu parametrů, definujících jeho chování a reprezentaci v administračním software, lze je rozdělit do tří kategorií:

- *Vlastnosti* ... definují obecné vlastnosti modulu nebo zařízení,
- *události* ... definují chování systému při konkrétní změně stavu (například sepnutí vstupu),
- *popisy* ... ke každé archivní značce definují text, který má být zobrazen při prohlížení archivu událostí.

Pro jejich editaci slouží *Průzkumník modulu* (*obr. 3.3*), umístěný v levé dolní části hlavního okna programu.

[ Branka ]	
Vlastnosti	Události
Vlastnost	Hodnota
Name	Branka
ModuleID	1
SaveEvents	true
Type	General
StrikeTime	5
AjarTime	20
CardTimeout	2
PartitionNo	0
Description	Branka

[ Branka ]	
Vlastnosti	Události
Událost	Makro
DoorAjar	Branka.DoorAjar
ForcedDoor	Branka.ForcedDoor

[ Branka ]	
Vlastnosti	Události
Značka	Popis
DoorAjar	Dlouho otevřené dveře
ForcedDoor	Vyražené dveře
CommLost	Ztráta komunikace
CommRestore	Obnovení komunikace

Obr. 3.3: Průzkumník modulu

Pro všechny *moduly* jsou povinné tři vlastnosti:

- Name (Jméno) ... jedinečný textový identifikátor modulu v aplikaci. Může obsahovat znaky „A“ .. „Z“, „a“.. „z“, „0“ .. „9“, „\_“ a „-“. Maximální délka jména je 25 znaků, musí být v rámci aplikace unikátní a nesmí být totožné se žádným z klíčových slov makrojazyka ani s jiným jménem definovaným v aplikaci.
- ModuleID (ID Modulu) ... jedinečné číslo přiřazené modulu. ID Modulu se přiřazuje podle následujících pravidel:

- 1) Řídicí modul ... vždy ModuleID = 0.
- 2) Síťový modul ... ModuleID = HW adresa modulu (1 - 64).
- 3) Ústředna EZS ... vždy ModuleID = 100.
- 4) Oblast EZS ... ModuleID = 100 + Číslo oblasti.
- 5) Napájecí zdroj ... Module ID = HW adresa modulu (71 - 78).

Stejně jako jméno musí být i ID Modulu v rámci aplikace jednoznačné.

- Description (Popis) ... je libovolný textový popis modulu. Přenáší se do konfigurace administračního software.

U všech *zařízení* je povinný název (je přiřazen implicitně a nelze jej měnit). Množinu datových typů vlastností a způsob jejich editace v průzkumníku udává tabulka na *obr. 3.4*.

datový typ	rozsah hodnot	příklad	způsob editace
číselný	0..255	ModuleID	zápisem do editačního řádku
logický	true, false	SaveEvents	zápisem do editačního řádku, poklepáním levým tlačítkem myši
Vyjmenovaný – typ čtecího modulu	General, Door, DoorWithHandle	Type (u síťového modulu)	zápisem do editačního řádku, poklepáním levým tlačítkem myši
textový	libovolný text, max. 50 znaků	Description	zápisem do editačního řádku

*Obr. 3.4: Datové typy vlastností*

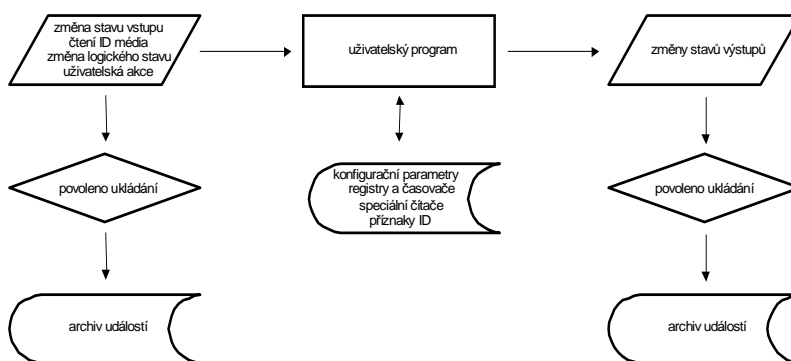
Seznam *událostí* modulu nebo zařízení zobrazuje všechny události, na které systém může reagovat vykonáním makra. Při vzniku události se zapisuje archivní značka do archivu událostí systému (je-li její zápis povolen).

Je-li pro nějakou událost vytvořeno *makro*, zobrazí se jeho název v odpovídajícím editačním řádku Průzkumníku modulu. Při vytváření makra stačí na tento řádek poklepat levým tlačítkem myši – program zapíše název makra do Průzkumníku a současně jej vytvoří i v editoru scriptu.

Seznam *popisů* archivních značek umožňuje upravit textové popisy, které se mají zobrazovat obsluze při prohlížení archivu provozních událostí. Kompletní přehled systémových archivních značek je obsahem přílohy 5.4: „Přehled systémových archivních značek“.

## 3.3 Událost a makro

Program v systému APS 400 vykonává řídicí modul. Ten detekuje změny stavů jak vlastních, tak i podřízených síťových modulů a na jejich základě generuje tzv. *události*. Na každou událost pak může reagovat vykonáním uživatelského programu, tzv. *makra*, které předepisuje reakci systému. Mechanismus vzniku a zpracování události přibližuje diagram na *obr. 3.5*.



*Obr. 3.5: Postup zpracování události*

Při provádění makra, přesněji při provádění příkazu Delay, může dojít k opakované aktivaci události, která makro spustila. V této situaci se řídicí modul musí rozhodnout, zda má běžící makro přerušit (a provádět znovu od začátku) nebo jej nechat proběhnout do konce. Jeho chování je dáno nastavením 5. přepínače SW konfigurace řídicího modulu (viz. katalogový list řídicího modulu).

Poznámka: U většiny aplikací je výhodné nechat makra přerušovat, v případě nejistoty raději volby konzultujte s technickou podporou.

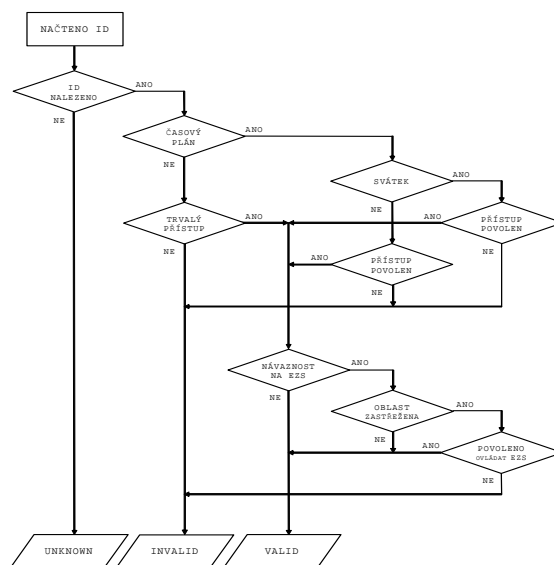
Většina událostí je aktivována samotnou změnou nějakého stavu. U událostí aktivovaných identifikací uživatele, tj. událostí Valid, Invalid a Unknown, je mechanismus o něco složitější. Po načtení ID uživatele síťovým modulem dojde v řídicím modulu k rozhodování, kterou z těchto událostí zvolit. Průběh tohoto procesu je zřejmý z *obr. 3.6*.

Celý model přístupových oprávnění je poměrně komplexní. Obsahuje čítače přítomných osob, programovatelné příznaky vázané na konkrétní ID a další mechanismy, vhodné pro nejrůznější typy aplikací. Jeho popis je obsahem dokumentů:

- [http://www.techfass.cz/files/t\\_aps\\_400\\_opravneni\\_cz.pdf](http://www.techfass.cz/files/t_aps_400_opravneni_cz.pdf).

Podrobný technický popis nejběžnějších aplikací:

- [http://www.techfass.cz/files/t\\_aps\\_400\\_ijidelna\\_cz.pdf](http://www.techfass.cz/files/t_aps_400_ijidelna_cz.pdf),
- [http://www.techfass.cz/files/t\\_aps\\_400\\_iadministrator\\_vis\\_cz.pdf](http://www.techfass.cz/files/t_aps_400_iadministrator_vis_cz.pdf).



*Obr. 3.6: Přístupová oprávnění*

### 3.4 Script

Ve scriptu jsou zapsána všechna makra definující chování systému při vzniku nejrůznějších událostí, komentáře, deklarace registrů, softwarových časovačů, uživatelských archivních značek a parametrů.

Ve scriptu se nerozlišují velká a malá písmena, například pro jméno modulu tedy platí:

```
Reader1=reader1=READER1
```

#### Makra

Makro má následující strukturu:

```
Event JménoModulu.IdentifikačtorUdálosti
... tělo makra ...
EventEnd
```

Tělo makra tvoří posloupnost příkazů definujících chování systému. Klíčová slova `Event` a `EventEnd` jsou povinná a vymezují začátek a konec makra. `JménoModulu` je jméno, které je modulu přiřazeno vlastností `Name` v průzkumníku modulů. `IdentifikačtorUdálosti` je pojmenování konkrétní události a nelze jej měnit.

#### Komentáře

Komentář je část řádku scriptu začínající znakem „;“. Všechny znaky do konce řádku jsou při překladu ignorovány. Komentáře slouží pro vysvětlení programového kódu, významu použitých proměnných apod. Program je vhodné bohatě komentovat – usnadní se tím jeho pochopení při pozdějších úpravách.

```
Event JménoModulu.IdentifikačtorUdálosti
; toto je komentář
EventEnd
```

### Registry

Registr je číselná proměnná dostupná ve všech makrech, je možné do něj zapsat libovolnou číselnou hodnotu z intervalu <0;255> a na jiném místě scriptu tuto hodnotu testovat. V aplikaci je možné použít až 250 registrů. Deklarace registru může být umístěna v kterémkoliv místě scriptu, z důvodu přehlednosti programu je však vhodné veškeré deklarace umístit na začátek scriptu.

Deklarace registru má následující formát:

```
define register JménoRegistru
```

### Softwarové časovače

Softwarový časovač je, stejně jako registr, číselná proměnná, avšak s odlišným významem. Je-li jeho hodnota rovna 255, časovač stojí. Je-li hodnota menší, automaticky se každou sekundu snižuje o jedničku až k nule. Ve scriptu je možné časovači přiřadit číselnou hodnotu (smysl mají hodnoty 1-254), čímž se spustí časování intervalu odpovídajícího zadané hodnotě v sekundách. Kdekoliv ve scriptu je pak možné testovat zda časování ještě probíhá (hodnota > 0), nebo zda je již časování dokončeno (hodnota=0). Je-li podmínka pro nulovou hodnotu časovače vyhodnocena jako pravda, je automaticky nastavena na 255 (časovač je zastaven). V aplikaci je možné použít až 250 softwarových časovačů.

Deklarace softwarového časovače má následující formát:

```
define timer JménoČasovače
```

### Uživatelské parametry

Uživatelský parametr je rovněž platný v celém scriptu. V aplikaci je možné použít až 254 uživatelských parametrů. Uživatelský parametr může být použit jako parametr příkazu Delay, přiřazení hodnoty registru a časovači.

Deklarace uživatelského parametru má následující formát:

```
define parameter JménoParametru=Hodnota
```

Uživatelský parametr je vhodné použít tam, kde je pracovat se stejnou hodnotou na více místech programu. Případnou změnu hodnoty pak stačí provést na jediném místě.

### Uživatelské archivní značky

Uživatelské archivní značky se používají pro zápis informace o průchodu určitou částí programu do archivu událostí, jsou platné v celém scriptu, v aplikaci je možné deklarovat až 254 archivních značek.

Deklarace uživatelského parametru má následující formát:

```
define mark JménoArchivníZnačky=PopisnýText
```

Uživatelské archivní značky se často používají při ladění programu, v „ostrých“ aplikacích se používají pro zaznamenání těch stavů systému, které nelze rozpoznat ze systémových archivních značek.

---

## 3.5 Překlad programu, přenos do paměti řídicího modulu

Po sestavení programu je nutné jej přeložit do kódu, kterému „rozumí“ řídicí modul. Překlad se provede příkazem *Přeložit* z nabídky *Aplikace*. Nejsou-li při překladu nalezeny syntaktické chyby, lze výsledný kód přenést do řídicího modulu (příkazem *Přenos scriptu*) a odzkoušet chování systému. V opačném případě se ve spodní části editoru scriptu zobrazí výpis nalezených chyb. Překladač přitom dokáže zkontrolovat jen chyby syntaktické – tzn. pokud narazí na nějaký zápis, kterému nerozumí, ohlásí chybu. Logické chyby v naprogramovaných algoritmech odhalit nedokáže.

Poznámka 1: Je-li tělo makra prázdné, je při překladu ze scriptu odstraněno.

Poznámka 2: Při přejmenování modulu v průzkumníku se změna automaticky projeví i v editoru scriptu.

## 4

## Programování systému

## 4.1 Standardní funkce systému

Standardní funkce systému je možné v prostředí APS 400 Config naprogramovat dvojím způsobem:

- "Ručně", jak je popsáno v dalších kapitolách,
- pomocí konfiguračních dialogů pro standardní funkce jednotlivých modulů.

Konfigurační dialog, příslušný modulu vybranému v aplikačním stromu, lze zobrazit volbou příkazu *Standardní funkce* z nabídky *Moduly* hlavního menu programu a z lokálního popup menu aplikačního stromu.

**Funkce konfiguračních dialogů**

Aby přístupový systém vykonával nějakou funkci, je nutné ji definovat scriptem, případně nastavením hodnot nějakých vlastností. Konfigurační dialogy nedělají nic jiného, než že tyto činnosti pro standardní funkce provedou automaticky, na základě zadaných hodnot.

**Společné prvky konfiguračních dialogů**

U všech konfiguračních dialogů jsou společné následující prvky:

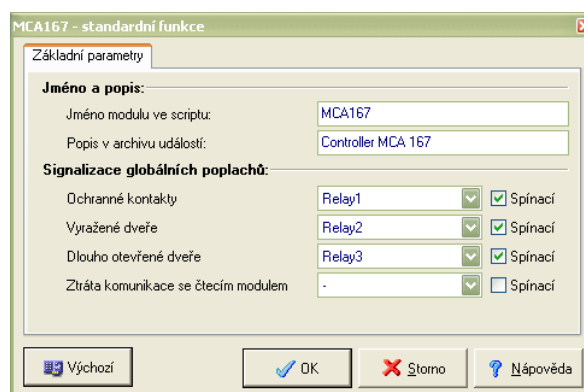
- Pole *Jméno modulu ve scriptu* pro nastavení vlastnosti Name,
- pole *Popis v archivu událostí* pro nastavení vlastnosti Description,
- tlačítko *Výchozí*, po jehož stisknutí se v dialogu nastaví výchozí standardní funkce modulu.

**Standardní funkce řídicího modulu**

Řídicímu modulu je v konfiguračním dialogu (*obr. 4.1*) možné přiřadit hlášení globálních poplachových stavů systému. Poplachové hlášení je realizováno nastavením volitelného výstupu řídicího modulu (Relé 1 až Relé 8).

Logika hlášení je konfigurovatelná. Standardně se předpokládá hlášení poplachového stavu rozepnutím výstupu. Opačnou logiku lze definovat zatržením boxu *Spínací*.

Význam jednotlivých poplachových hlášení a jejich výchozí nastavení uvádí tabulka na *obr. 4.2*.



Obr. 4.1: Standardní funkce řídicího modulu



Hlášení	Význam	Výstup	Logika
ochranné kontakty	byl narušen alespoň jeden z ochranných kontaktů v systému (řídící modul, síťové moduly)	Relé 1	rozpínací
vyražené dveře	alespoň jeden ze čtecích modulů je ve stavu vyražené dveře	Relé 2	rozpínací
dlouho otevřené dveře	alespoň jeden ze čtecích modulů je ve stavu dlouho otevřené dveře	Relé 3	rozpínací
ztráta komunikace...	je ztracena komunikace alespoň s jedním ze síťových modulů	Relé 4	rozpínací

Obr. 4.2: Poplachová hlášení řídícího modulu

## Standardní funkce síťového čtecího modulu

Na záložce Základní parametry se definuje způsob řízení dveří, tedy jsou-li dveře čtecím modulem ovládané jednostranně, nebo oboustranně v páru s druhým čtecím modulem (v tom případě je nutné druhý čtecí modul zvolit v příslušném výběrovém poli – obr. 4.3a). Poslední položkou je pole pro výběr typu modulu, význam jednotlivých typů je popsán dále.

Pro oboustranné ovládání dveří platí, že by měly být ovládány modulem na straně vyšší bezpečnosti (uvnitř objektu). Modul na straně nižší bezpečnosti zajišťuje pouze načtení kódu identifikační karty, případně kódu PIN a jeho výstup a vstupy jsou pak volné pro jiné použití.

Obr. 4.3a: Standardní funkce síťového modulu

## Pokročilé parametry

Záložka (obr. 4.3b) Pokročilé parametry obsahuje pole pro zadání doby uvolnění zámku, maximální doby otevření dveří (po jejím překročení je hlášen stav „Dlouho otevřené dveře“ – je-li typ modulu nastaven na hodnoty Door nebo DoorWithHandle) a doby prodlevy čtení poslední karty.

Obr. 4.3b: Standardní funkce síťového modulu

## Archivní značky

Záložka „Archivní značky“ obsahuje pole pro definici textových popisů vybraných událostí.

## Výchozí nastavení

Po stisku tlačítka Výchozí nastavení se nastavení modulu nakonfiguruje pro jednostranné řízení dveří, typ modulu Průchozí, doba otevření zámku 5s, max. doba otevření dveří 20s, timeout poslední karty 1s, odchozí tlačítko. Současně se nastaví popisy vybraných událostí na hodnoty uvedené na obr. 4.3c.

Obr. 4.3c: Standardní funkce síťového modulu

## 4.2 Makra

Tělo makra, jak již bylo napsáno dříve, tvoří posloupnost příkazů, které má systém po spuštění makra provádět. Jednotlivé příkazy se provádějí postupně, v pořadí, v jakém jsou v těle makra zapsány. Na každém řádku může být nejvýše jeden příkaz, případně doplněný komentářem. V těle makra lze pracovat se všemi moduly a zařízeními vloženými do stromečku aplikace. Každé zařízení má v aplikaci jednoznačný identifikátor, který má následující formát:

JménoModulu.JménoZařízení

kde JménoModulu je jméno, které jsme danému modulu přiřadili ve vlastnosti Name v průzkumníku modulů a JménoZařízení je pevně dané jméno zařízení, jednoznačné v rámci modulu. Zařízení, která jsou součástí modulu, který makro aktivoval, mohou být ve scriptu identifikována jen svým jménem (překladač automaticky předpokládá, že zařízení je součástí modulu generujícího událost).

Příkazy makrojazyka lze rozdělit do 3 skupin: Příkaz přiřazení, logická podmínka a příkazy pro spuštění různých funkcí systému.

### Příkaz přiřazení

Příkaz přiřazení umožňuje programové nastavení stavu výstupů systému, registrů a časovačů. Formát příkazu přiřazení je následující:

Identifikátor=hodnota

Typ a rozsah hodnot použitelných v konkrétním případě závisí na typu nastavovaného stavu. Pro stavy logických zařízení (vstupy, výstupy,...) jsou vyhrazeny hodnoty On (sepnuto) a Off (rozepnuto), pro číselné hodnoty registrů a časovačů hodnoty z intervalu <0;255>. Použití příkazu přiřazení demonstruje *příklad 4.1*. V příkladu jsou definována dvě makra – pro událost sepnutí a rozepnutí vstupu (Input1) síťového modulu (Reader1). Makra realizují kopírování stavu tohoto vstupu na výstup 1 (Relay1) řídicího modulu (MasterModule).

příklad 4.1: Kopírování stavu vstupu

```
Event Reader1.Input1On      ; makro pro sepnutí vstupu
  MasterModule.Relay1=On    ; sepnutí výstupu
EventEnd

Event Reader1.Input1Off     ; makro pro rozepnutí vstupu
  MasterModule.Relay1=Off   ; rozepnutí výstupu
EventEnd
```

### Podmínka

V makrojazyku lze vyhodnocovat jednoduché logické výrazy a větvit tak řídicí program. Pro vyhodnocení logických výrazů slouží následující konstrukce:

```
If LogickýVýraz then
  ... blok příkazů, které se provedou při pravdivosti výrazu ...
Else
  ... blok příkazů, které se provedou při nepravdivosti výrazu ...
EndIf
```

Pokud není nutný kód pro obě větve podmínky, stačí kratší zápis:

```
If LogickýVýraz then
  ... tělo podmínky ...
EndIf
```

Logický výraz může testovat logické stavy vstupů a výstupů systému, číselné hodnoty registrů, stavy softwarových časovačů, vybrané stavy systému EZS, případně testovat různé stavy systému. Přehled podporovaných podmínek je uveden v příloze 5.3: „Přehled nejdůležitějších konstrukcí programovacího jazyka“.

V podmínkách lze vyhodnocovat jen jednoduché logické výrazy (jeden logický stav, jednu číselnou hodnotu nebo jednu systémovou podmínku). Jediným logickým operátorem použitelným v logických výrazech je operátor NOT (negace). Podmínky však lze vnořovat do sebe (až 20 vnoření) a nepřítomnost operátorů AND a OR tak nahradit vhodnou kombinací podmínek. *Příklad 4.2* demonstruje náhradu operátoru AND, *příklad 4.3* náhradu operátoru OR a *příklad 4.4* testování logických proměnných systému.

Příklad 4.2: Náhrada operátoru AND (vnořené podmínky)

```
Event Reader1.Input1On
  if Reader2.Input1=On then
    if Reader3.Input1=On then
      Relay1=On
    endif
  endif
EventEnd
```

Příklad 4.3: Náhrada operátoru OR (podmínky se stejným tělem)

```
Event Reader1.Input1On
  if Reader2.Input1=On then
    Relay1=On
  endif
  if Reader3.Input1=On then
    Relay1=On
  endif
EventEnd
```

Příklad 4.4: Testování logických proměnných systému

```
Event MasterModule.Input1On
  if AllCountersEmpty then ; není-li nikdo v objektu
    Signal(Reader1,2)      ; dvě krátká pípnutí na modulu Reader1
  Else                     ; je-li někdo v objektu
    Signal(Reader1,4)      ; dlouhé pípnutí na modulu Reader1
  endif
EventEnd
```

## Příkazy pro spouštění funkcí systému

Poslední skupina příkazů spouští nejrůznější funkce implementované v samotném systému. Jejich přehled je uveden v příloze 5.3: „Přehled nejdůležitějších konstrukcí programovacího jazyka“. Příkladem je použití příkazu Signal v *příkladu 4.4*.

## Použití registrů

Registru je možné přiřadit hodnotu z intervalu <0;255>, přičíst nebo odečíst konstantu a testovat hodnotu registru v podmínce.

Základní příkazy pro práci s registry jsou následující:

JménoRegistru = Hodnota	; přiřazení hodnoty registru
JménoRegistru = JménoRegistru + konstanta	; přičtení konstanty k registru
JménoRegistru = JménoRegistru - konstanta	; odečtení konstanty od registru
if JménoRegistru = Hodnota then	; testování hodnoty registru
if JménoRegistru > Hodnota then	; testování hodnoty registru

V obou podmínkách je opět možné použít ještě logický operátor NOT.

V *příkladu 4.5* je registr využit pro větvení programu v závislosti na stavu systému (normální provoz / evakuace objektu).

Zapojení a funkce systému je v daném případě následující:

Na vstup 1 řídicího modulu je připojen přepínač, jímž se nastavuje stav „Evakuace“ (Sepnuto = Evakuace, Rozepnuto = Normální provoz). V systému jsou nainstalovány dva síťové čtecí moduly, jejichž výstupy mají být při evakuaci trvale sepnuty. Každých 10 s mají síťové čtecí moduly akusticky indikovat stav evakuace.

příklad 4.5: Využití registrů

```
define register Evakuace           ; definice registru

Event MasterModule.Init           ; událost od inicializace scriptu
  if Input1=Off then               ; test stavu vstupu 1
    Evakuace = 0                   ; nastavení stavu na normální provoz
  endif
  if Input1=On then                ; test stavu vstupu 1
    Evakuace = 1                   ; nastavení stavu na evakuaci
  endif
EventEnd

Event MasterModule.Timer10sec      ; událost od hardwarového časovače 10 s
  if Evakuace = 1 then             ; test na stav evakuace
    Reader1.Beep = On              ; sepnutí bzučáku čtecího modulu 1
    Reader2.Beep = On              ; sepnutí bzučáku čtecího modulu 2
    Delay(2)                       ; čekání 2 s
    Reader1.Beep = Off             ; vypnutí bzučáku čtecího modulu 1
    Reader2.Beep = Off             ; vypnutí bzučáku čtecího modulu 2
  EndIf
EventEnd

Event MasterModule.Input1On        ; událost od sepnutí vstupu 1 řídicího modulu
  Evakuace = 1                     ; nastavení stavu evakuace
  Reader1.Relay1 = On              ; sepnutí výstupu čtecího modulu 1
  Reader2.Relay1 = On              ; sepnutí výstupu čtecího modulu 2
EventEnd

Event MasterModule.Input1Off       ; událost od rozepnutí vstupu 1 řídicího modulu
  Evakuace = 0                     ; nastavení stavu normální provoz
  Reader1.Relay1 = Off             ; rozepnutí výstupu čtecího modulu 1
  Reader2.Relay1 = Off             ; rozepnutí výstupu čtecího modulu 2
EventEnd

Event Reader1.Valid                ; platná identifikace na čtecím modulu 1
  if Evakuace = 0 then             ; test na normální provoz
    Door()                         ; volání funkce pro obsluhu dveří
  EndIf
EventEnd

Event Reader2.Valid                ; platná identifikace na čtecím modulu 2
  if Evakuace = 0 then             ; test na normální provoz
    Door()                         ; volání funkce pro obsluhu dveří
  EndIf
EventEnd
```

## Použití softwarových časovačů

Nastavování a testování hodnoty časovače je stejné jako u registru, není však podporováno přičítání a odčítání konstanty. *Příklad 4.6* je rozšířením *příkladu 4.5*. Časovač je zde využit pro ovládání akustické signalizace při evakuaci.

Zapojení je rozšířeno takto:

Na výstup 1 řídicího modulu je připojena siréna, která má být spuštěna první 2 minuty stavu evakuace. Na výstup 2 řídicího modulu je připojena optická indikace stavu evakuace, která má být aktivní po celou dobu trvání tohoto stavu.

příklad 4.6: Využití softwarových časovačů

```
define register Evakuace           ; definice registru
define timer Sirena                ; definice registru

Event MasterModule.Init           ; událost od inicializace scriptu
  if Input1=Off then               ; test stavu vstupu 1
    Evakuace = 0                   ; nastavení stavu na normální provoz
    Sirena = 255                   ; vypnutí časovače Siréna
    Relay1 = Off                  ; zhasnutí optické signalizace
    Relay2 = Off                  ; vypnutí sirény
  endif
  if Input1=On then                ; test stavu vstupu 1
    Evakuace = 1                   ; nastavení stavu na evakuaci
    Sirena = 120                   ; aktivace časovače Siréna na 120s
    Relay1 = On                   ; rozsvícení optické signalizace
    Relay2 = On                   ; zapnutí sirény
  endif
EventEnd

Event MasterModule.Timer1sec       ; událost od hardwarového časovače 1 s
  if Sirena = 0 then               ; doběhl časovač?
    Relay2 = Off                  ; vypnutí sirény
  endif
EventEnd

Event MasterModule.Timer10sec      ; událost od hardwarového časovače 10 s
  if Evakuace = 1 then             ; test na stav evakuace
    Reader1.Beep = On              ; sepnutí bzučáku čtecího modulu 1
    Reader2.Beep = On              ; sepnutí bzučáku čtecího modulu 2
    Delay(2)                       ; čekání 2 s
    Reader1.Beep = Off             ; vypnutí bzučáku čtecího modulu 1
    Reader2.Beep = Off             ; vypnutí bzučáku čtecího modulu 2
  endif
EventEnd

Event MasterModule.Input1On        ; událost od sepnutí vstupu 1 řídicího modulu
  Evakuace = 1                     ; nastavení stavu evakuace
  Reader1.Relay1 = On              ; sepnutí výstupu čtecího modulu 1
  Reader2.Relay1 = On              ; sepnutí výstupu čtecího modulu 2
  Sirena = 120                     ; aktivace časovače Siréna na 120s
  Relay1 = On                      ; rozsvícení optické signalizace
  Relay2 = On                      ; zapnutí sirény
EventEnd

Event MasterModule.Input1Off       ; událost od rozepnutí vstupu 1 řídicího modulu
  Evakuace = 0                     ; nastavení stavu normální provoz
  Reader1.Relay1 = Off             ; rozepnutí výstupu čtecího modulu 1
  Reader2.Relay1 = Off             ; rozepnutí výstupu čtecího modulu 2
  Sirena = 255                     ; vypnutí časovače Siréna
  Relay1 = Off                     ; zhasnutí optické signalizace
  Relay2 = Off                     ; vypnutí sirény
EventEnd
```

```
Event Reader1.Valid          ; platná identifikace na čtecím modulu 1
  if Evakuace = 0 then       ; test na normální provoz
    Door()                  ; volání funkce pro obsluhu dveří
  EndIf
EventEnd

Event Reader2.Valid          ; platná identifikace na čtecím modulu 2
  if Evakuace = 0 then       ; test na normální provoz
    Door()                  ; volání funkce pro obsluhu dveří
  endif
EventEnd
```

### Použití uživatelských parametrů

Uživatelské parametry lze použít v následujících konstrukcích:

```
Delay(JménoParametru)
JménoRegistru=JménoParametru
JménoČasovače=JménoParametru
Delay(JménoParametru)
```

**Příklad 4.7** ukazuje typické použití uživatelského parametru pro nastavení časování doby otevření zámku dveří po stisknutí tlačítka. Zapojení a funkce systému je následující:

V systému je instalován jeden síťový čtecí modul Reader1. Na jeho výstup 1 (Relay1) je připojen zámek dveří. Na vstup 1 (Input1) je připojen koncový spínač indikující otevření dveří (spínač je sepnut, jsou-li dveře zavřené). Zámek dveří má být aktivován po stisku tlačítka připojeného na vstup 2 (Input2) síťového čtecího modulu a deaktivován při indikaci otevření dveří koncovým spínačem, nejdéle však smí být aktivován dobu definovanou uživatelským parametrem DobaOtevreni.

příklad 4.7: Využití uživatelských parametrů

```
define parameter DobaOtevreni=10 ; definice uživatelského parametru

Event Reader1.Input2On          ; událost od sepnutí vstupu 2 čtecího modulu
  if Input1=On then             ; test stavu dveří
    Relay1 = On                ; aktivace zámku
    Delay(DobaOtevreni)         ; čekání po dobu definovanou parametrem
    Relay1 = Off                ; deaktivace zámku
  endif
EventEnd

Event Reader1.Input1Off          ; událost od rozepnutí hlásiče stavu dveří
  Relay1 = Off                  ; deaktivace zámku
EventEnd
```



## Použití uživatelských archivních značek

Uživatelské archivní značky se do ukládají příkazem:

Mark(JménoArchivníZnačky)

*Příklad 4.8* je rozšířením *příkladu 4.7*. Funkce systému je rozšířena o uložení archivní značky v případě, že je zámek aktivován tlačítkem, avšak po dobu definovanou uživatelským parametrem nedojde k otevření dveří.

příklad 4.8: Využití uživatelských značek

```
define parameter DobaOtevreni=10 ; definice uživatelského parametru
define register DvereOtevreny ; definice pomocného registru
define mark DvereNeotevreny=Dveře nebyly otevřeny ; definice uživatelské
; archivní značky

Event MasterModule.Init ; událost od inicializace skriptu
DvereOtevreny=0 ; nastavení pomocného registru
EventEnd

Event Reader1.Input2On ; událost od sepnutí vstupu 2 čtecího modulu
if Input1=On then ; test stavu dveří
DvereOtevreny=0 ; nastavení pomocného registru
Relay1 = On ; aktivace zámku
Delay(DobaOtevreni) ; čekání po dobu definovanou parametrem
Relay1 = Off ; deaktivace zámku
if DvereOtevreny=0 then ; test stavu pomocného registru
Mark(DvereNeotevreny) ; uložení archivní značky
endif
endif
EventEnd

Event Reader1.Input1Off ; událost od rozepnutí koncového spínače
Relay1 = Off ; deaktivace zámku
DvereOtevreny = 1 ; nastavení pomocného registru
EventEnd
```

## 5.1 Moduly systému APS 400

### Řídící modul MCA 167(8)

Obr. 5.1: Řídící modul MCA 167(8) – pokračování

## Sítový modul

Sítový modul – vlastnosti

Název	Typ	Poznámka
Name	textový	Definuje jednoznačný textový identifikátor modulu, kterým je reprezentován ve scriptu. Znak použitelné při definici jména modulu jsou z množiny ('0..9', 'a'..'z', 'A'..'Z', '-', '_').
ModuleID	číselný	Definuje číselný identifikátor modulu pro vnitřní použití systému. U síťových čtecích musí být roven nastavené HW adrese.
SaveEvents	logický	Určuje mají-li být archivní značky generované modulem ukládány do archivu událostí.
Type	vyjmenovaný – typ čtecího modulu	Určuje typ čtecího modulu, který ovlivňuje jeho standardní chování. Podrobný přehled jednotlivých typů a popis jejich vlastností je uveden v tabulce na <b>obr. 5.3</b> .
StrikeTime	číselný	Doba uvolnění zámku po volání funkce Door [s].
AjarTime	číselný	Omezení doby otevření dveří [s], po jejím překročení přechází čtecí modul do stavu DoorAjar a současně generuje příslušné události.
CardTimeout	číselný	Doba po kterou bude poslední přístupová karta přečtená čtecím modulem tímto modulem ignorována [s].
Description	textový	Libovolný text, přenáší se jako popis modulu do administračního software APS 400.

Sítový modul – události

Název	Podmínka pro vznik události
DoorAjar	Dojde-li k přechodu modulu do stavu DoorAjar (dlouho otevřené dveře).
ForcedDoor	Dojde-li k přechodu modulu do stavu ForcedDoor (vyražené dveře).

Sítový modul – archivní značky

Název	Podmínka pro zapsání značky do archivu
DoorAjar	Při vzniku události DoorAjar.
ForcedDoor	Při vzniku události ForcedDoor.
CommLost	Při ztrátě komunikace s řídícím modulem.
CommRestore	Při obnovení komunikace s řídícím modulem.

Sítový modul – seznam zařízení

Tamper, Beep, CardReader, Input1, Input2, Relay1, AuxOutput
---

Obr. 5.2: Sítový modul

## Vlastnost Type

Vlastnost Type síťového čtecího modulu vyžaduje podrobnější vysvětlení funkce modulu při *standardním zapojení* a použití příkazu Door.

Standardní zapojení předpokládá připojení hlásiče stavu dveří k prvnímu vstupu síťového modulu (zavřené dveře = sepnutý vstup), zámek připojený k relé (k dispozici jsou jak NO, tak i NC kontakty) a případný kontakt kliky ke druhému vstupu (stisknutá klika = sepnutý vstup). Podrobná schémata zapojení pro různé typy zámků, otvíračů a síťových modulů najdete na stránce [http://www.techfass.cz/aplikace\\_cz.html](http://www.techfass.cz/aplikace_cz.html).

Na *příkladech 5.1, 5.2 a 5.3* je naznačeno řešení základních případů využití nastavení typu modulu při volání funkce Door. Při studiu příkladů je vhodné vyzkoušet jejich funkci „živě“ a otestovat i chování systému při změnách hodnot vlastností StrikeTime, AjarTime a CardTimeout.

Typ	Význam a použití
General	Obecný modul, není naprogramován standardně.
Door	Modul se standardně připojenými dveřmi a standardní funkcí. Automaticky hlásí vyražené a dlouho otevřené dveře.
DoorWithHandle	Modul se standardně připojenými dveřmi, kontaktem kliky připojeným je vstupu 2 a standardní funkcí. Automaticky hlásí vyražené a dlouho otevřené dveře, zohledňuje při tom stisknutí kliky.

Obr. 5.3: Význam a použití vlastnosti Type síťového čtecího modulu

### **Funkce Door**

Po spuštění funkce Door dojde k sepnutí relé a aktivaci bzučáku do doby otevření dveří, avšak nejdéle po dobu danou hodnotou vlastností StrikeTime. Dojde-li k neautorizovanému otevření dveří (tedy rozepnutí vstupu 1 bez předchozího příkazu řídicího modulu nebo stisknutí kliky) je aktivována událost „Vyražené dveře“ (ForcedDoor). Jsou-li dveře otevřené déle než udává parametr AjarTime, je aktivována událost „Dlouho otevřené dveře“ (DoorAjar).

### **Událost „Vyražené dveře“ (ForcedDoor)**

Událost „Vyražené dveře“ je aktivována u typů Door a DoorWithHandle. Podmínkou její aktivace je v obou případech uvolnění prvního vstupu síťového modulu při neaktivním výstupu 1 (relé). U typu DoorWithHandle navíc nesmí být sepnut druhý vstup, který je považován za stisknutí kliky, tedy za oprávněné otevření dveří.

### **Událost „Dlouho otevřené dveře“ (DoorAjar)**

Událost „Dlouho otevřené dveře“ je aktivována u typů Door a DoorWithHandle je-li první vstup otevřen déle než udává hodnota vlastnosti AjarTime. Je-li modul současně ve stavu „Vyražené dveře“, „Dlouho otevřené dveře“ se nehlásí.

### **Příklady**

Příklad 5.1: Jednostranné řízení dveří modulem Reader1, typ DoorWithHandle, na 2. vstup připojen kontakt kliky

```
Event Reader1.Valid
  Door()          ; Otevření dveří na základě platného oprávnění uživatele
EventEnd
```

Příklad 5.2: Jednostranné řízení dveří modulem Reader1, typ Door, na 2. vstup připojeno odchozí tlačítko

```
Event Reader1.Valid
  Door()          ; Otevření dveří na základě platného oprávnění uživatele
EventEnd
```

```
Event Reader1.Input2On
  Door()          ; Otevření dveří na základě stisknutí odchozího tlačítka
EventEnd
```

Příklad 5.3: Dvoustranné řízení dveří dvojicí síťových modulů, Reader1 typu Door a Reader2 typu General, dveře standardně připojeny k modulu Reader1

```
Event Reader1.Valid
  Door()
EventEnd

Event Reader2.Valid
  Door(Reader1)
EventEnd
```

## Systémový zdroj

<

Obr.5.4: Systémový zdroj PSU 71

## Ústředna EZS (NX-8), interface IFNIO 844

Modul ústředny EZS NX-8 – vlastnosti		
Název	Typ	Poznámka
Name	textový	Definuje jednoznačný textový identifikátor modulu, kterým je reprezentován ve scriptu. Znak použitelné při definici jména modulu jsou z množiny ('0'..'9', 'a'..'z', 'A'..'Z', '-', '_').
ModuleID	číselný	Definuje číselný identifikátor modulu pro vnitřní použití systému. U ústředny EZS musí být roven 100.
Description	textový	Libovolný text, přenáší se jako popis modulu do administračního software APS 400.
Modul ústředny EZS NX-8 – události		
Název	Podmínka pro vznik události	
StatusChanged	Při změně stavu oblastí, příznaků nebo zařízení EZS.	
Modul ústředny EZS NX-8 – archivní značky		
Název	Podmínka pro zapsání značky do archivu	
CommLost	Ztráta komunikace s interface	
CommRestored	Obnovení komunikace s interface	
NXCommLost	Ztráta komunikace s NX-8	
NXCommRestored	Obnovení komunikace s NX-8	
Modul ústředny EZS NX-8 – seznam zařízení		
Zones, Flags		

Obr. 5.5: Modul ústředny EZS (NX-8)

Poznámka: Interface IFNIO 844 lze používat samostatně, jako modul vstupů a výstupů, bez připojené ústředny EZS (NX-8).

## Oblast EZS (pro NX-8)

Modul oblasti EZS – vlastnosti		
Název	Typ	Poznámka
Name	textový	Definuje jednoznačný textový identifikátor modulu, kterým je reprezentován ve scriptu. Znak použitelné při definici jména modulu jsou z množiny ('0..9', 'a'..'z', 'A'..'Z', '-', '_').
ModuleID	číselný	Definuje číselný identifikátor modulu pro vnitřní použití systému. U oblastí EZS je roven 100+číslo_oblasti.
Description	textový	Libovolný text, přenáší se jako popis modulu do administračního software APS 400.
Modul oblasti EZS – události		
Název	Podmínka pro vznik události	
StatusChanged	Při změně stavu oblasti EZS.	
Modul oblasti EZS – seznam zařízení		
Alarm, AlarmSignal, Armed, Fire, Ready, SystemKey.		

Obr. 5.6: Modul oblasti EZS (NX-8)

## 5.2 Zařízení systému APS 400

V této kapitole jsou podrobně popsány vlastnosti, události a archivní značky všech typů zařízení systému APS 400. Vyskytuje-li se u nějakého modulu více zařízení téhož typu, jsou v názvu odlišeny pořadovým číslem (např. Input1, Input2, ...), zde jsou označeny obecně (InputX, OutputX, ...), kde X je pořadové číslo zařízení.

### Zařízení Alarm

Logické zařízení Alarm reprezentuje stav alarmu v oblasti EZS. Lze testovat v podmínce.

Zařízení Alarm – vlastnosti		
Název	Typ	Poznámka
Name	textový	Definuje jednoznačný textový identifikátor zařízení v rámci nadřazeného modulu. Jméno přiřazuje prostředí automaticky a nelze jej měnit.
Zařízení Alarm – archivní značky		
Název	Podmínka pro zapsání značky do archivu	
AlarmOn	Spuštění alarmu v oblasti.	
AlarmOff	Ukončení alarmu v oblasti.	

Obr. 5.7: Zařízení Alarm

### Zařízení AlarmSignal

Logické zařízení AlarmSignal slouží pro vypnutí akustické signalizace (od alarmu i poruchy) v nadřazené oblasti, případně globálně, je-li nadřazeným modulem ústředna EZS. Ovládá se příkazem přiřazení (AlarmSignal=Off), jeho ovládání je omezeno na událost Valid (musí být autorizováno ovládání systému EZS) a uživatelské události. Ovlivňuje systémovou proměnnou Passed.

Zařízení AlarmSignal – vlastnosti		
Název	Typ	Poznámka
Name	textový	Definuje jednoznačný textový identifikátor zařízení v rámci nadřazeného modulu. Jméno přiřazuje prostředí automaticky a nelze jej měnit.
Zařízení AlarmSignal – archivní značky		
Název	Podmínka pro zapsání značky do archivu	
AlarmSignalOff	Provedení příkazu AlarmSignal=Off.	

Obr. 5.8: Zařízení AlarmSignal

### Zařízení Armed

Zařízení Armed reprezentuje stav střežení oblasti EZS, stav lze testovat v podmínce i programově ovládat příkazem přiřazení (hodnoty On a Off). Nastavení hodnoty Off (tedy odstřežení oblasti) je omezeno na událost Valid (musí být autorizováno ovládání systému EZS) a uživatelské události.



Zařízení Armed – vlastnosti		
Název	Typ	Poznámka
Name	textový	Definuje jednoznačný textový identifikátor zařízení v rámci nadřazeného modulu. Jméno přiřazuje prostředí automaticky a nelze jej měnit.
Zařízení Armed – archivní značky		
Název	Podmínka pro zapsání značky do archivu	
ArmedOn	Zastřežení oblasti.	
ArmedOff	Odstřežení oblasti.	

Obr. 5.9: Zařízení Armed

**Zařízení Beep**

Zařízení Beep reprezentuje bzučák modulu. Stav bzučáku lze programově ovládat příkazem přiřazení (hodnoty On, Off), může být i argumentem funkce Invert.

Zařízení Beep – vlastnosti		
Název	Typ	Poznámka
Name	textový	Definuje jednoznačný textový identifikátor zařízení v rámci nadřazeného modulu. Jméno přiřazuje prostředí automaticky a nelze jej měnit.

Obr. 5.10: Zařízení Beep

**Zařízení CardReader**

Zařízení CardReader reprezentuje čtecí zařízení síťového čtecího modulu; jeho stav nelze programově ovládat. Po přečtení přístupové karty je vyhodnocena platnost přístupového oprávnění (*obr. 3.6*) a aktivována odpovídající událost Valid, Invalid, nebo Unknown. Ukládání archivních značek tohoto zařízení nelze zakázat.

Zařízení CardReader – vlastnosti		
Název	Typ	Poznámka
Name	textový	Definuje jednoznačný textový identifikátor zařízení v rámci nadřazeného modulu. Jméno přiřazuje prostředí automaticky a nelze jej měnit.
Zařízení CardReader – události		
Název	Podmínka pro vznik události	
Valid	Přečtení platné přístupové karty.	
Invalid	Přečtení neplatné přístupové karty.	
Unknown	Přečtení neznámé přístupové karty.	
Invalid PIN	5x po sobě zadán neplatný PIN kód	
Zařízení CardReader – archivní značky		
Název	Podmínka pro zapsání značky do archivu	
Valid	Při události Valid.	
Invalid	Při události Invalid.	
Unknown	Při události Unknown.	

Obr. 5.11: Zařízení CardReader

## Zařízení Flags ústředny EZS

Zařízení Flags reprezentuje stavy příznaků ústředny EZS, stavy lze testovat v podmínce (viz. *příklad 5.5*).

Zařízení Flags – vlastnosti		
Název	Typ	Poznámka
Name	textový	Definuje jednoznačný textový identifikátor zařízení v rámci nadřazeného modulu. Jméno přiřazuje prostředí automaticky a nelze jej měnit.

*Obr. 5.12: Zařízení Flags*

Příklad 5.5: Testování příznaků ústředny EZS

```

Event EZS.StatusChanged      ; změna stavu ústředny EZS
  if Flags(fLowBattery) then  ; test příznaku fLowBattery
    MasterModule.Relay1=On    ; aktivace výstupu řídicího modulu
                              ; např. pro optickou signalizaci
  endif
EventEnd

```

Seznam všech příznaků, které lze testovat a jejich význam udává tabulka na *obr. 5.13*.

Příznak	Význam
fDownload	je navázáno telefonní spojení mezi nadřazeným PC a ústřednou EZS
fExpanderTamper	ochranný kontakt expanderu EZS
fExpanderLoss	ztráta komunikace s expanderem EZS
fSmokeReset	probíhá reset požárních hlásičů
fBoxTamper	ochranný kontakt ústředny EZS
fLowBattery	vybitá baterie EZS
fACFail	porucha síťového napájení ústředny EZS
fSystemTimeLoss	ztráta systémového času ústředny EZS
fGlobalBuzzer	aktivní bzučák alespoň jedné klávesnice
fGlobalSirene	aktivní siréna
fSteadySirene	aktivní požární siréna
fNXLoss	ztráta komunikace mezi řídicím modulem a ústřednou EZS

*Obr. 5.13: Příznaky ústředny EZS*

## Zařízení Flags systémového zdroje

Zařízení Flags reprezentuje jednotlivé stavové informace systémového zdroje (PSU71). Stavy lze testovat v podmínce (viz. *příklad 5.6*).

Zařízení Flags – vlastnosti		
Název	Typ	Poznámka
Name	textový	Definuje jednoznačný textový identifikátor zařízení v rámci nadřazeného modulu. Jméno přiřazuje prostředí automaticky a nelze jej měnit.

*Obr.5.14: Zařízení PSU71.Flags*

Příklad 5.6: Testování příznaků systémového zdroje

```

Event PSU71.StatusChanged    ; změna stavu zdroje s adresou 71
  if not Flags(fACPresent) then ; test příznaku fACPresent
    MasterModule.Relay1=On    ; aktivace výstupu řídicího modulu
                              ; např. pro optickou signalizaci
  endif
EventEnd

```



Seznam všech příznaků, které lze testovat a jejich význam udává tabulka na *obr. 5.15*.

Příznak	Význam
fOverload	Přetížení výstupu.
fBatteryPresent	Baterie připojena.
fLowBattery	Slabá baterie.
fBoxTamper	Ochranný kontakt rozepnut.
fDCOn	Výstupní napětí zapnuto.
fACPresent	Napájení ze sítě.

Obr.5.15: Seznam příznaků systémového zdroje (PSU 71)

**Zařízení Fire**




Zařízení Fire reprezentuje stav příznaku požáru v oblasti EZS. Lze testovat v podmínce.

 Zařízení Fire – vlastnosti		
Název	Typ	Poznámka
Name	textový	Definuje jednoznačný textový identifikátor zařízení v rámci nadřazeného modulu. Jméno přiřazuje prostředí automaticky a nelze jej měnit.
 Zařízení Fire – archivní značky		
Název	Podmínka pro zapsání značky do archivu	
FireOn	Vznik příznaku Fire dané oblasti EZS.	
FireOff	Zánik příznaku Fire dané oblasti EZS.	

Obr. 5.16: Zařízení Fire

**Zařízení InputX**

Zařízení InputX reprezentuje X-tý vstup modulu. Stav vstupu nelze programově ovládat. Stav vstupu lze testovat v podmínce.

 Zařízení InputX – vlastnosti		
Název	Typ	Poznámka
Name	textový	Definuje jednoznačný textový identifikátor zařízení v rámci nadřazeného modulu. Jméno přiřazuje prostředí automaticky a nelze jej měnit.
SaveEvents	logický	Určuje mají-li být archivní značky generované zařízením ukládány do archivu událostí.
 Zařízení InputX – události		
Název	Podmínka pro vznik události	
InputXOn	Sepnutí vstupu InputX	
InputXOff	Rozepnutí vstupu InputX	
 Zařízení InputX – archivní značky		
Název	Podmínka pro zapsání značky do archivu	
InputXOn	Při události InputXOn	
InputXOff	Při události InputXOff	

Obr. 5.17: Zařízení Input

## Zařízení PSU (Power Supply Unit)

Zařízení PSU reprezentuje speciální vstup řídicího modulu, určený pro připojení výstupu zdroje napájecího napětí. Stav vstupu lze testovat v podmínce.

Zařízení PSU – vlastnosti		
Název	Typ	Poznámka
Name	textový	Definuje jednoznačný textový identifikátor zařízení v rámci nadřazeného modulu. Jméno přiřazuje prostředí automaticky a nelze jej měnit.
SaveEvents	logický	Určuje mají-li být archivní značky generované zařízením ukládány do archivu událostí.
Zařízení PSU – události		
Název	Podmínka pro vznik události	
PSUOn	Výstupní napětí O.K.	
PSUOff	Výstupní napětí K.O.	
Zařízení PSU – archivní značky		
Název	Podmínka pro zapsání značky do archivu	
PSUOn	Při události PSUOn.	
PSUOff	Při události PSUOff.	

Obr. 5.18: Zařízení PSU

## Zařízení Ready

Zařízení Ready reprezentuje stav „připraveno“ oblasti EZS. Stav lze testovat v podmínce.

Zařízení Ready – vlastnosti		
Název	Typ	Poznámka
Name	textový	Definuje jednoznačný textový identifikátor zařízení v rámci nadřazeného modulu. Jméno přiřazuje prostředí automaticky a nelze jej měnit.

Obr. 5.19: Zařízení Ready

## Zařízení RelayX a AuxOutput

Zařízení RelayX a AuxOutput reprezentují výstupy hardwarových modulů. Stav výstupů lze programově ovládat příkazem přiřazení (lze přiřadit hodnoty On, Off), případně může být argumentem funkce Invert. Stav výstupů lze testovat v podmínce.

Zařízení RelayX a AuxOutput – vlastnosti		
Název	Typ	Poznámka
Name	textový	Definuje jednoznačný textový identifikátor zařízení v rámci nadřazeného modulu. Jméno přiřazuje prostředí automaticky a nelze jej měnit.
SaveEvents	logický	Určuje mají-li být archivní značky generované zařízením ukládány do archivu událostí.
Zařízení RelayX, AuxOutput – archivní značky		
Název	Podmínka pro zapsání značky do archivu	
RelayXOn, AuxOutputOn	Sepnutí výstupu RelayX, AuxOutput.	
RelayXOff, AuxOutputOff	Rozepnutí výstupu RelayX, AuxOutput.	

Obr. 5.20: Zařízení RelayX a AuxOutput

**Zařízení SystemKey**

Zařízení SystemKey slouží pro vyslání systémového kódu klávesnice NX-8 (viz. *příklad 5.7*).

Zařízení SystemKey – vlastnosti		
Název	Typ	Poznámka
Name	textový	Definuje jednoznačný textový identifikátor zařízení v rámci nadřízeného modulu. Jméno přiřazuje prostředí automaticky a nelze jej měnit.

*Obr. 5.21: Zařízení SystemKey*

Příklad 5.7: Vyslání klávesnicového kódu ústředny EZS

```
Event Reader1.Valid           ; platná karta na čtecím modulu
  Oblast1.SystemKey=Kód       ; vyslání systémového kódu
EventEnd
```

Seznam všech systémových kódů, které lze vyslat a jejich význam udává tabulka na *obr. 5.22*.

Kód	Význam
cChime	gong na klávesnici
cFPanic	stisk tlačítka F Panic na klávesnici
cAPanic	stisk tlačítka A Panic na klávesnici
cPPanic	stisk tlačítka P Panic na klávesnici
cDisarm	odstřežení
cArmAway	zastřežení
cArmStay	zastřežení
cInitiateAutoArm	inicializace automatického zastřežení
cSmokeReset	reset požárních hlásičů
cSounder	spuštění bzučáku klávesnice

*Obr. 5.22: Klávesnicové kódy EZS*

**Zařízení Tamper**

Zařízení Tamper reprezentuje ochranný kontakt modulu. Stav ochranného kontaktu nelze testovat v podmínce (generuje události). Změna stavu ochranného kontaktu libovolného modulu ovlivňuje globální stav GlobalTamper systému.

Zařízení Tamper – vlastnosti		
Název	Typ	Poznámka
Name	textový	Definuje jednoznačný textový identifikátor zařízení v rámci nadřazeného modulu. Jméno přiřazuje prostředí automaticky a nelze jej měnit.
Zařízení Tamper – události		
Název	Podmínka pro vznik události	
Tamper	Rozepnutí vstupu Tamper.	
Zařízení Tamper – archivní značky		
Název	Podmínka pro zapsání značky do archivu	
Tamper	Vygenerování události tamper.	
Tamper OK	Sepnutí vstupu Tamper.	

*Obr. 5.23: Zařízení Tamper*

## Zařízení Timer1sec, Timer10sec

Zařízení Timer1sec, resp. Timer10sec reprezentují hardwarové časovače řídicího modulu. Hardwarové časovače průběžně generují každou 1s, resp. 10s událost. Na tyto události lze reagovat makry. Při tvorbě maker pro tyto události je nutné mít na zřeteli, že makra jsou volána pravidelně a jejich kód se musí vykonat než dojde ke spuštění téhož makra. Nemí tedy vhodné zde používat příkazy Delay a Door ani ovládat systém EZS.

Zařízení Timer1s, Timer10s – vlastnosti		
Název	Typ	Poznámka
Name	textový	Definuje jednoznačný textový identifikátor zařízení v rámci nadřízeného modulu. Jméno přiřazuje prostředí automaticky a nelze jej měnit.
Zařízení Timer1s, Timer10s – události		
Název	Podmínka pro vznik události	
Timer1sec	Události Timer1sec, resp. Timer10sec jsou systémem generovány každou 1s, resp. každých 10s.	
Timer10sec		

Obr. 5.24: Zařízení Timer1sec, Timer10sec

## Zařízení TroubleSignal

Logické zařízení TroubleSignal slouží pro deaktivaci akustické signalizace od poruchy ústředny EZS. Ovládá se příkazem přiřazení (TroubleSignal=Off), ovlivňuje systémovou proměnnou Passed.

Zařízení TroubleSignal – vlastnosti		
Název	Typ	Poznámka
Name	textový	Definuje jednoznačný textový identifikátor zařízení v rámci nadřízeného modulu. Jméno přiřazuje prostředí automaticky a nelze jej měnit.
Zařízení TroubleSignal – archivní značky		
Název	Podmínka pro zapsání značky do archivu	
TroubleSignalOff	Provedení příkazu TroubleSignal=Off.	

Obr. 5.25: Zařízení TroubleSignal

## Zařízení UserEvents

Zařízení UserEvents reprezentuje 8 událostí, které lze generovat z nadřízeného PC. Uživatelské události lze spouštět i z jiných maker příkazem Execute(X), kde X je číslo uživatelské události. Příkaz Execute není možné používat v makrech pro uživatelské události.

Zařízení UserEvents – vlastnosti		
Název	Typ	Poznámka
Name	textový	Definuje jednoznačný textový identifikátor zařízení v rámci nadřazeného modulu. Jméno přiřazuje prostředí automaticky a nelze jej měnit.
SaveEvents	logický	Určuje mají-li být značky generované zařízením ukládány do archivu událostí.
Zařízení UserEvents – události		
Název	Podmínka pro vznik události	
UserEvent1	Odeslání příkazu pro vygenerování události UserEventX z nadřazeného PC, nebo volání funkce Execute(X) z nějakého makra ve scriptu.	
...		
UserEvent8		
Zařízení UserEvents – archivní značky		
Název	Podmínka pro zapsání značky do archivu	
UserEvent1	Při události UserEventX.	
...		
UserEvent8		

Obr. 5.26: Zařízení UserEvents



**Zařízení Zones**

Zařízení Zones reprezentuje stavy smyček EZS. Stavy lze testovat v podmínce (viz. příklad 5.8).

Poznámka: Změna stavu smyčky negeneruje událost StatusChanged modulu zabezpečovací ústředny.

Zařízení Zones– vlastnosti		
Název	Typ	Poznámka
Name	textový	Definuje jednoznačný textový identifikátor zařízení v rámci nadřazeného modulu. Jméno přiřazuje prostředí automaticky a nelze jej měnit.

*Obr. 5.27: Zařízení Zones*

Příklad 5.8: Testování stavu smyčky ústředny EZS

```

Event MasterModule.Timer10sec    ; událost od časovače 10 s
  if Zones(1) then                ; test stavu smyčky 1
    MasterModule.Relay1=On        ; aktivace výstupu řídicího modulu
                                  ; např. pro optickou signalizaci
  endif
EventEnd

```

### 5.3 Přehled nejdůležitějších konstrukcí programovacího jazyka

Vzhledem k průběžnému rozšiřování možností systému doporučujeme občas seznam aktualizovat podle seznamu uvedeného na stránce [http://www.techfass.cz/aps\\_400\\_makrojazyk\\_cz.html](http://www.techfass.cz/aps_400_makrojazyk_cz.html), tamtéž jsou uvedeny odkazy na příklady jejich použití a další související zdroje informací. Dále uvedený seznam odpovídá verzi firmware řídicího modulu 202 a překladače 2.15, ze dne 6.6.2005.

#### **Break**

Ukončí vykonávání makra. Použití ve všech událostech.

#### **ClearAllCounters**

Vymaže všechny příznaky *PRESENT* u přístupových karet a současně i čítače počtu přítomných osob u skupin a uživatelských kódů. Použití ve všech událostech.

#### **ClearGroupCounter(x)**

Vymaže čítač přítomných osob u skupiny definované parametrem *x*, současně nuluje příznak *PRESENT* u přístupových karet patřících do definované skupiny a odpovídajícím způsobem dekrementuje čítač u uživatelského kódu. Použití ve všech událostech.

#### **ClearPresent**

Nuluje příznak *PRESENT* načtené karty a současně dekrementuje odpovídající čítače počtu přítomných osob u skupin a uživatelských kódů. Použití v události *Valid*.

#### **ClearUserCounter(x)**

Vymaže čítač přítomných osob u uživatelského kódu definovaného parametrem *x*, současně nuluje příznak *PRESENT* u přístupových karet patřících do skupin s nastaveným uživatelským kódem *x* a odpovídajícím způsobem dekrementuje čítače přítomných osob dotčených přístupových skupin. Použití v události *Valid*.

#### **define mark x=popisný text**

Definice uživatelské archivní značky. Zapisuje se mimo kód událostí.

#### **define parameter x=jmeno**

Definice uživatelského parametru. Zapisuje se mimo kód událostí.

#### **define register x=jmeno**

Definice číselného registru. Zapisuje se mimo kód událostí.

#### **define timer x=jmeno**

Definice softwarového časovače. Zapisuje se mimo kód událostí.

#### **Delay(x)**

Pozastaví vykonávání makra na počet sekund definovaný parametrem *x*. Použití ve všech událostech, vyjma *Timer1sec* a *Timer10sec*.

#### **Dial(x,y)**

Naváže modemové spojení s PC. Parametr *x* definuje index vytáčeného telefonního čísla (1 nebo 2), parametr *y* definuje uživatelskou archivní značku, popisující důvod navázání spojení. Použití ve všech událostech.

#### **Display(x) / Display(ReaderN,X)**

Zobrazí symbol na displeji jehož makro je prováděno, resp. na displeji definovaným názvem *ReaderN*. Význam parametru *x* je následující:

- 0 - 9 ... číslice 0 - 9,
- 10 ... pomlčka,
- 11 ... tři vodorovné čáry nad sebou,
- 12 ... znak E,

- 13 ... desetinná tečka,
- 14 ... rozsvítí všechny segmenty (včetně desetinné tečky),
- 15 ... zhasne všechny segmenty.

***Door()* / *Door (ReaderN)***

Provede standardní otevření dveří. Použití v událostech *Valid*.

***EraseCard***

Vymaže načtenou kartu z tabulky přístupových karet. Použití v události *Valid*.

***Execute(x)***

Aktivuje uživatelskou událost definovanou parametrem *x*. Použití ve všech událostech vyjma *UserEventN* (rekurze není povolena).

***if AllCountersEmpty then***

Vyhodnotí nulovost všech čítačů přítomných osob přístupových skupin i uživatelských kódů. Použití ve všech událostech.

***if Authorized then***

Vyhodnotí autorizační příznak přístupového oprávnění uživatele. Použití v události *Valid*.

***if Buffer(x) then***

Vyhodnotí zaplnění archivu událostí, je-li archiv zaplněn více než definuje parametr *x* (v procentech) je výsledek pravdivý. Použití ve všech událostech.

***if Byte1=x then***

Porovná zadanou hodnotu *x* s hodnotou uloženou ve sloupci *Byte1* načtené karty. Při jejich shodě je výsledek pravdivý. Použití v událostech *Valid* a *Invalid*.

***if Byte2=x then***

Porovná zadanou hodnotu *x* s hodnotou uloženou ve sloupci *Byte2* načtené karty. Při jejich shodě je výsledek pravdivý. Použití v událostech *Valid* a *Invalid*.

***if Group(x) then***

Vyhodnotí příslušnost načtené karty ke skupině s číslem *x*. Použití v události *Valid*.

***if GroupEmpty(x) then***

Vyhodnotí nulovost čítače přítomných osob u přístupové skupiny číslo *x*. Použití ve všech událostech.

***if KeyCode(x) then***

Porovná kód stisknuté klávesy na čtečce s důvodovou klávesnicí se zadaným kódem *x*. Použití v události *Valid*.

***if ModuleLost(x) then***

Vyhodnotí stav komunikace síťového modulu s adresou *x*. Pokud modul nekomunikuje, je výsledek pravdivý. Použití ve všech událostech.

***if Present then***

Vyhodnotí příznak *PRESENT* načtené karty. Použití v události *Valid*.

***if TimePlan(x) then***

Vyhodnotí platnost časového plánu definovaného číslem *x*. Použití ve všech událostech.

***if User(x) then***

Vyhodnotí rovnost čísla *x* a uživatelského kódu přístupové skupiny načtené karty. Použití v události *Valid*.

### ***if UserEmpty(x) then***

Vyhodnotí nulovost čítače přítomných osob u skupin s nastaveným uživatelským kódem *x*. Použití ve všech událostech.

### ***if x then ... endif / if x then ... else ... endif / if not x then ... endif / if not x then ... else ... endif***

Logická podmínka. Vyhodnotí podmínku *x*. Výsledek vyhodnocení lze negovat operátorem *not*. Podmínky je možné vnořovat až do 20 úrovní. Použití ve všech událostech.

### ***Invert (x)***

Invertuje stav logického výstupu definovaného jménem *x*. Použití ve všech událostech.

### ***Mark(x)***

Uloží do archivu událostí uživatelskou značku definovanou jménem *x*. Použití ve všech událostech.

### ***SetByte1(x)***

Nastaví načtené kartě hodnotu *x* do sloupce *Byte1* v paměti řídicího modulu. Použití v událostech *Valid* a *Invalid*.

### ***SetByte2(x)***

Nastaví načtené kartě hodnotu *x* do sloupce *Byte2* v paměti řídicího modulu. Použití v událostech *Valid* a *Invalid*.

### ***SetModePIN(x)***

Přepne čtecí modul definovaný jménem *x* do režimu identifikace "karta + PIN". Takto lze přepínat jen čtečky s klávesnicí pro PIN kód, ostatní typy nastavení ignorují. Použití ve všech událostech.

### ***SetModePINFree(x)***

Přepne čtecí modul definovaný jménem *x* do režimu identifikace "jen karta". Takto lze přepínat jen čtečky s klávesnicí pro PIN kód, ostatní typy nastavení ignorují. Použití ve všech událostech.

### ***SetPresent***

Nastaví načtené kartě příznak *PRESENT* a inkrementuje odpovídající čítače přítomných osob u skupiny a uživatelského kódu. Použití v události *Valid*.

**Signal(x) / Signal(ReaderN,X)**

Spustí akustický signál na čtečce jejíž makro je prováděno, resp. na čtečce definované názvem *ReaderN*. Význam parametru *x* je následující:

- 1 ... jedno krátké pípnutí,
- 2 ... dvě krátká pípnutí,
- 3 ... tři krátká pípnutí,
- 4 ... jedno dlouhé pípnutí,
- 5 ... pět krátkých pípnutí,
- 6 ... příkaz pro smazání paměti karet v paměti čtecího modulu.

## 5.4 Přehled systémových archivních značek

IDUdalosti	Popis	IDUdalosti	Popis	IDUdalosti	Popis
<b>Archivní značky řídicího modulu (IDModulu=0)</b>					
4	Input1 On	40	Rezervováno	132	Relay 3 On
5	Input1 Off	41	Rezervováno	133	Relay 3 Off
6	Input2 On	42	Rezervováno	134	Relay 4 On
7	Input2 Off	43	Rezervováno	135	Relay 4 Off
8	Tamper	45	Global door ajar	136	Relay 5 On
16	Input3 On	46	Global forced door	137	Relay 5 Off
17	Input3 Off	47	Global door ajar OK	138	Relay 6 On
18	Input4 On	48	Timer 1 sec	139	Relay 6 Off
19	Input4 Off	49	Timer 10 sec	140	Relay 7 On
20	Input5 On	50	Init	141	Relay 7 Off
21	Input5 Off	51	Global tamper	142	Relay 8 On
22	Input6 On	52	Global tamper ok	143	Relay 8 Off
23	Input6 Off	53	Global comm lost	144	Rezervováno
24	Input7 On	54	Global comm restore	145	Rezervováno
25	Input7 Off	55	PSU On	146	Rezervováno
26	Input8 On	56	PSU Off	147	Rezervováno
27	Input8 Off	60	User event 1	148	Rezervováno
28	Rezervováno	61	User event 2	149	Rezervováno
29	Rezervováno	62	User event 3	150	Rezervováno
30	Rezervováno	63	User event 4	151	Rezervováno
31	Rezervováno	64	User event 5	152	Rezervováno
32	Rezervováno	65	User event 6	153	Rezervováno
33	Rezervováno	66	User event 7	154	Rezervováno
34	Rezervováno	67	User event 8	155	Rezervováno
35	Rezervováno	68	Global forced door ok	156	Rezervováno
36	Rezervováno	128	Relay 1 On	157	Rezervováno
37	Rezervováno	129	Relay 1 Off	158	Rezervováno
38	Rezervováno	130	Relay 2 On	159	Rezervováno
39	Rezervováno	131	Relay 2 Off	160	LineBusy
<b>Archivní značky programování z PC (IDModulu=253)</b>					
0	Connected	2	Parametres upload	-	-
1	Script upload	3	Access data upload	-	-
<b>Archivní značky síťového čtecího modulu (IDModulu=HW adresa modulu)</b>					
1	Valid	7	Input2 Off	66	Comm. lost
2	Invalid	8	Tamper	67	Comm. restored
3	Unknown	9	DoorAjar	161	AuxOutput On
4	Input1 On	10	Forced Door	162	AuxOutput Off
5	Input1 Off	64	Relay1 On	249	Tamper OK
6	Input2 On	65	Relay1 Off	250	Invalid PIN

Obr. 5.28: Archivní značky systému

Archivní značky síťového zdroje napájecího napětí (IDModulu=HW adresa modulu)					
8	Tamper	72	Battery low	77	Battery disconnected
66	Comm. lost	73	Battery OK	78	DC On
67	Comm. restored	74	AC Lost	79	DC Off
70	DC overload	75	AC Restored	249	Tamper OK
71	DC OK	76	Battery connected		
Archivní značky ústředny EZS (IDModulu=100)					
4	Alarm signal off	35	Rezervováno	148	Rezervováno
5	Trouble signal off	36	Rezervováno	149	Rezervováno
66	Comm. lost	37	Rezervováno	150	Rezervováno
67	Comm. restored	38	Rezervováno	151	Rezervováno
68	NX lost	39	Rezervováno	152	Rezervováno
69	NX restored	40	Rezervováno	153	Rezervováno
28	Rezervováno	41	Rezervováno	154	Rezervováno
29	Rezervováno	42	Rezervováno	155	Rezervováno
30	Rezervováno	43	Rezervováno	156	Rezervováno
31	Rezervováno	144	Rezervováno	157	Rezervováno
32	Rezervováno	145	Rezervováno	158	Rezervováno
33	Rezervováno	146	Rezervováno	159	Rezervováno
34	Rezervováno	147	Rezervováno	-	-
Archivní značky oblasti EZS (IDModulu=100+číslo oblasti)					
4	Armed On	7	Fire Off	10	AlarmOff
5	Armed Off	8	Alarm signal Off	-	-
6	Fire On	9	Alarm On	-	-

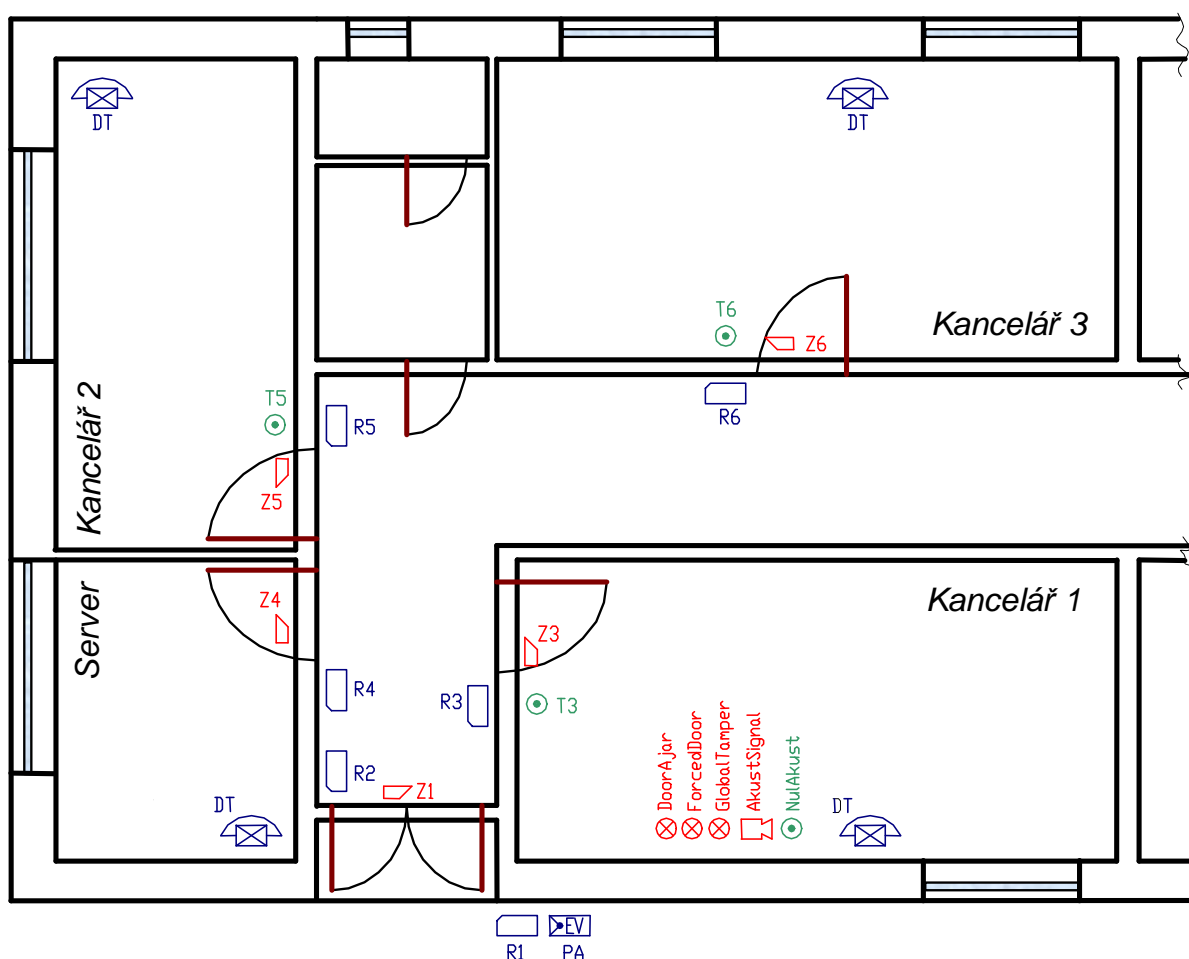
Obr. 5.28: Archivní značky systému (pokračování)

## 5.5 Ucelený příklad aplikace systému APS 400

### Požadovaná funkce systému

Schéma na *obr. 5.29* znázorňuje standardní aplikaci přístupového systému pro řízení přístupu do objektu a do jednotlivých kanceláří. Na systém jsou kladeny následující požadavky:

- Otevírání hlavního vstupu má být kombinováno s využitím funkce domácího telefonu,
- má být umožněna evidence aktuální přítomnosti pracovníků,
- stavy „Vyražené dveře“ a „Narušení“ mají být opticky a akusticky signalizovány, akustická signalizace má být aktivní nastavitelnou dobu po vzniku příslušného stavu, může být předčasně ukončena i stiskem tlačítka nebo z nadřazeného PC,
- stav „Dlouho otevřené dveře“ má být signalizován pouze opticky.



Obr. 5.29: Zjednodušené schéma aplikace přístupového systému

### Hardwarové řešení

Z důvodu požadavku evidence aktuální přítomnosti pracovníků v objektu je hlavní vchod osazen čtecími moduly oboustranně (R1, R2). Dveře mohou být ovládány např. motorickým zámkem. Otevírání hlavních dveří pomocí domácího telefonu slouží pouze pro návštěvy. Dveře do místnosti počítačového serveru jsou osazeny čtecím modulem z chodby (R4), který ovládá např. elektromechanický zámek s kontaktem kliky. Dveře do kanceláří mají v zárubních osazené otvírače a jsou ovládány z vnější strany čtecími moduly, z vnitřní, odchozími tlačítky.



Zapojení vstupů řídicího modulu:

- Výstupní relé domácího telefonu ... vstup 1,
- tlačítko pro vypnutí akustické signalizace ... vstup 2.

Zapojení výstupů řídicího modulu:

- Siréna ... relé 1,
- indikátor stavu „Dlouho otevřené dveře“ ... relé 3,
- indikátor stavu „Vyražené dveře“ ... relé 4,
- indikátor stavu „Narušení“ ... relé 5.

Vstupy a výstupy čtecích modulů budou zapojeny standardně, viz [http://www.techfass.cz/aplikace\\_cz.html](http://www.techfass.cz/aplikace_cz.html)

Poznámka 1: V každém případě je při instalaci přístupového systému nutno zajistit nouzový únik z vnitřních prostorů nezávisle na funkci přístupového systému dle platných předpisů.

### Softwarové řešení

Do aplikace v prostředí APS Config vložíme řídicí modul MCA 167(8) (pojmenovaný `RidiciModul`) a šest síťových čtecích modulů (pojmenované `R1..R6`). Pro větší přehlednost pro uživatele systému definujeme následující popisy modulů (vlastnost `Description`):

<code>RidiciModul</code>	... Řídicí modul
<code>R1</code>	... Vchod do objektu
<code>R2</code>	... Odchod z objektu
<code>R3</code>	... Kancelář 1
<code>R4</code>	... Server
<code>R5</code>	... Kancelář 2
<code>R6</code>	... Kancelář 3

Dvojice čtecích modulů `R1` a `R2`, osazená u vstupních dveří do objektu, má nastavenou rozdílně vlastnost `typ`. Čtecí modul `R2` ovládá dveře přímo a generuje poplachové stavy, má tedy nastaven `typ Door`, čtecí modul `R1` ovládá dveře pouze zprostředkovaně, má tedy nastaven `typ General`. Má-li být možno zjišťovat aktuální přítomnost osob v objektu, je nutné nastavit příchozí a odchozí čtečku v komunikačním software, viz manuál: [http://www.techfass.cz/files/m\\_aps\\_400\\_iserver\\_cz.pdf](http://www.techfass.cz/files/m_aps_400_iserver_cz.pdf).

Čtečka `R4` má nastaven `typ DoorWithHandle`, ostatní čtečky `typ Door`.

Výpis programu pro řídicí modul:

```
define timer TMR1                ; softwarový časovač pro sirénu
define parameter PAR1=30         ; doba signalizace vyražených dveří
define parameter PAR2=30         ; doba signalizace narušení

Event RidiciModul.GlobalDoorAjar
    Relay3=ON                    ; optická signalizace dlouho otevřených dveří
EventEnd

Event RidiciModul.GlobalForcedDoor
    Relay4=ON                    ; optická signalizace vyražených dveří
    TMR1=PAR1                    ; start časování akustické signalizace
EventEnd

Event RidiciModul.GlobalTamper
    Relay5=ON                    ; optická signalizace narušení
    TMR1=PAR2                    ; start časování akustické signalizace
EventEnd

Event RidiciModul.GlobalTamperOK
    TMR1=0                       ; stop akustické signalizace
    Relay5=OFF                   ; stop optické signalizace
EventEnd
```

```

Event RidiciModul.GlobalDoorOK
    Relay3=OFF           ; stop optické signalizace
    Relay4=OFF           ; stop optické signalizace
    TMR1=0               ; stop akustické signalizace
EventEnd

Event RidiciModul.UserEvent1      ; vypnutí sirény z PC
    TMR1=0
EventEnd

Event RidiciModul.Timer1sec
    If TMR1>0 then
        Invert(Relay1)      ; takto bude siréna přerušovaně spínána
    Endif
    If TMR1=0 then          ; vypnutí sirény od časovče
        Relay1=OFF
    Endif
EventEnd

Event RidiciModul.Input2On        ; vypnutí sirény tlačítkem
    TMR1=0
EventEnd

Event RidiciModul.Input1On        ; otevření hlavních dveří z DT
    Door(R2)
EventEnd

Event R1.Valid
    Door(R2)
EventEnd

Event R2.Valid
    Door()
EventEnd

Event R3.Valid
    Door()
EventEnd

Event R3.Input2On
    Door()
EventEnd

Event R4.Valid
    Door()
EventEnd

Event R5.Valid
    Door()
EventEnd

Event R5.Input2On
    Door()
EventEnd

Event R6.Valid
    Door()
EventEnd

Event R6.Input2On
    Door()
EventEnd

```

Konec výpisu.

### 5.6 Programovací protokol

Všechny strany programovacího protokolu obsahují identické záhlaví a zápatí s pořadovým číslem strany. Levou část záhlaví sestavy je možné modifikovat nastavením parametrů tiskových sestav – viz. kapitola 2.6: „Nastavení programu“.

Vlastní tisková sestava je rozdělena na tři části:

- *Obecné parametry* ... obsahuje obecné informace o aplikaci, např. jméno souboru, datum a čas, způsob připojení k řídicímu modulu apod.
- *Seznam modulů* ... seznam všech modulů, včetně podřízených zařízení. U každého modulu, resp. zařízení, je uveden úplný výpis hodnot všech vlastností (normálním písmem) a popisů archivních značek (kurzívou).
- *Script* ... úplný výpis scriptu.