ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA ELEKTROTECHNICKÁ

BAKALÁŘSKÁ PRÁCE

Standardizace funkcí pro řízení inteligentních domů a návrh knihovny funkčních bloků

Praha, 2009

Autor: Jan KMENT

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra řídicí techniky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Jan Kment**

Studijní program: Elektrotechnika a informatika (bakalářský), strukturovaný
Obor: Kybernetika a měření

Název tématu: **Standardizace funkcí pro řízení inteligentních domů a návrh knihovny funkčních bloků**

Pokyny pro vypracování:

1. Seznamte se se zásadami programování podle standardu IEC EN 61131-3, s programovatelnými automaty Tecomat Foxtrot, s distribuovaným systémem Inels II, s vývojovým systémem Mosaic.
2. Analyzujte typické potřeby uživatelů inteligentních domů, zásady řízení prvků inteligentní instalace, jejich obsluhy a vytypujte ty, u kterých lze předpokládat opakované použit í v programech PLC.
3. Realizujte algoritmy pro vytypované opakovatelné funkce (algoritmy realizujte jako vhodné programové objekty - jako uživatelské funkční bloky, funkce nebo jen jako doporučené postupy a fragmenty programů podle zásad standardu IEC EN 61131-3 v prostředí Mosaic).
4. Vytvořte uživatelskou dokumentaci, případně metodickou příručku pro výuku.

Seznam odborné literatury:

Šmejkal, L. - Martinásková, M.: PLC automatizace 1. Základní pojmy a úvod do programování. BEN, Praha 1999.
Šmejkal, L.: PLC a automatizace 2. Sekvenčné logické funkce a minimum o fuzzy logice. BEN, Praha 2005.
Martinásková, M. - Šmejkal, L.: Řízení programovatelnými automaty III. Softwarové vybavení, ČVUT, Praha 2003.
kol.: Programování PLC podle normy IEC 61 131-3, Teco, Kolín 2007
www.tecomat.cz

Vedoucí: Ing. Martin Hlinovský, Ph.D.

Platnost zadání: do konce zimního semestru 2009/10

L.S.

prof. Ing. Michael Šebek, DrSc.
vedoucí katedry

doc. Ing. Boris Šimák, CSc.
děkan

V Praze dne 27. 2. 2009

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v přiloženém seznamu.

V Praze dne ……. …….

podpis

Poděkování

Na tomto místě bych chtěl poděkovat všem, jejichž rady a připomínky mě vždy posunovaly o krok vpřed. Chtěl bych také poděkovat všem těm, kteří přímo či nepřímo pomáhali při vzniku této práce a poskytli mi jakoukoli cennou radu. V nemalé míře bych chtěl poděkovat také své rodině, za podporu během celého studia.

Abstrakt

Tato bakalářská práce pojednává o současných inteligentních systémech používaných v inteligentních domech. V úvodu je popis inteligentních budov jako takových a popis některých základních pojmů, které se týkají automatizace budov.

Cílem této bakalářské práce je vytvořit souhrn nejvýznamnějších funkcí a ty prohlásit za standardní funkce, které se používají v současných inteligentních domech. Oblast použití funkcí je velmi široká od návrhu osvětlení, zabezpečovací techniky, až po ovládání žaluzií nebo vrat. V této práci jsem se zaměřil na typického uživatele a snažil jsem se, aby funkce či funkční bloky byly univerzální.

The Abstract

The bachelor work deals with the contemporary intelligent systems used for the intelligent houses. The introduction includes the description of the intelligent houses as well and the characterization of some basic terms related to the building automation.

The aim of this bachelor work is to create a summary of the most significant functions and then to declare them as the default functions which are applied in the intelligent houses at the present time. The application field is very wide from the lighting concept, safety technique to the sun-blind or the gate control. In this work I tried to focus on the typical user and I also tried to make the function blocks universal.

Obsah

1. Teoretická část ... 1
   1.1 Úvod ... 1
   1.2 Základní pojmy a uspořádání ... 3
   1.3 Uspořádání řídících systémů ... 6
      1.3.1 Centralizované systémy ... 6
      1.3.2 Decentralizované systémy ... 7
      1.3.3 Hybridní systémy ... 8
   1.4 Programovatelný automat - PLC ... 9
      1.4.1 Princip vykonávání uživatelského programu ... 9
      1.4.2 PLC Tecomat ... 10
      1.4.3 Tecomat Foxtrot ... 11
      1.4.4 Sběrnicový systém INELS ... 12
      1.4.5 Vývojové prostředí MOSAIC ... 13
   1.5 Mezinárodní norma IEC EN 61 131-3 ... 15
      1.5.1 Společné prvky ... 15
      1.5.2 Programovací jazyky ... 16
      1.5.3 Programové organizační jednotky ... 17
      1.5.4 Funkce a její syntaxe ... 18
      1.5.5 Funkční bloky a syntaxe ... 18
      1.5.6 Programy ... 19
      1.5.7 Struktura programové organizační jednotky (POU) ... 19
2. Typické požadavky uživatelů ... 22
   2.1 Elektroinstalace klasická versus inteligentní ... 22
      2.1.1 Inteligentní elektroinstalace na první pohled ... 24
   2.2 Přínosy uživatelům při použití inteligentní elektroinstalace oproti klasické ... 25
3. Realizace funkcí a funkčních bloků ... 27
   3.1 Osvětlení ... 27
      3.1.1 Paměťové funkce ... 27
      3.1.2 Schodiště a chodby ... 29
   3.2 Zabezpečovací technika ... 35
      3.2.1 Majorita ... 35
      3.2.2 Prahové funkce ... 38
      3.2.3 Elementární symetrické funkce ... 40
   3.3 Ovládání mechanismů ... 42
      3.3.1 Ovladač roletových žaluzií SBC1 ... 42
4. Návrh funkcí a funkčních bloků ... 44
   4.1 Složitější sekvenční funkce ... 44
      4.1.1 Jazyk SFC (Grafcet) ... 48
      4.1.2 Základní koncepce ... 48
      4.1.3 Funkce ovládání mechanismů ... 50
      4.1.4 Funkce rozpoznání dlouhého a krátkého stisku ... 51
      4.1.5 Funkce rozpoznání dvojkliku ... 52
      4.1.6 Scény ... 53
   4.2 Astroprogram ... 55
      4.2.1 Grafy poloh slunce – základní údaje ... 55
5. Závěr ... 61

Seznam obrázků

Obrázek 1.1: Ukázka inteligentního domu 2
Obrázek 1.2: Schéma centralizovaného systému 6
Obrázek 1.3: Schéma decentralizovaného systému 7
Obrázek 1.4: Cyklus řešení uživatelského programu 9
Obrázek 1.5: Základní modul Tecomat Foxtrot spolu s připojenými binárními moduly 11
Obrázek 1.6: Využití systému INELS se sběrnicí CIB v inteligentním domě 12
Obrázek 1.7: Ukázka vývojového prostředí Mosaic 14
Obrázek 1.8: Ukázka logické funkce AND ve všech čtyřech jazycích 17
Obrázek 1.9: Základní struktura POU 19
Obrázek 1.10: Základní struktura POU programu (deklarační i výkonná část) 20
Obrázek 2.1: Porovnání klasické a inteligentní elektroinstalace 22
Obrázek 2.2: Nástěnný ovladač inteligentní elektroinstalace 24
Obrázek 3.1: Jedno tlačítko s dvěma nestálými polohami 28
Obrázek 3.2: Čtyři tlačítka s dvěma nestálými polohami 29
Obrázek 3.3: Funkční blok T_FLOP 30
Obrázek 3.4: Průběh FB T_FLOP 30
Obrázek 3.5: Funkční blok T_FLOP_2 31
Obrázek 3.6: Funkční blok T_FLOP_4 31
Obrázek 3.7: Vzhled FB DMA1 32
Obrázek 3.8: Průběh časovače TP 33
Obrázek 3.9: Vzhled FB WC_vent1 33
Obrázek 3.10: Průběh časovače TON 34
Obrázek 3.11: Vzhled FB SV_DUM 34
Obrázek 3.12: Karnaughova mapa majority ze 3 36
Obrázek 3.13: FB majorita ze 3 36
Obrázek 3.14: FB majorita z 5 37
Obrázek 3.15: a) Prahová funkce 1 ze 4 b) Prahová funkce 2 ze 4 38
Obrázek 3.16: a) Prahová funkce 3 ze 4 b) Prahová funkce 4 ze 4 38
Obrázek 3.17: a) Prahová funkce 1 z 5 b) Prahová funkce 2 z 5 39
Obrázek 3.18: a) Prahová funkce 3 z 5 b) Prahová funkce 4 z 5 39
Obrázek 3.19: a) Symetrická funkce 1 ze 4 b) Symetrická funkce 2 ze 4 40
Obrázek 3.20: a) Symetrická funkce 3 ze 4 b) Symetrická funkce 4 ze 4 40
Obrázek 3.21: a) Symetrická funkce 1 z 5 b) Symetrická funkce 2 z 5 40
Obrázek 3.22: a) Symetrická funkce 3 z 5 b) Symetrická funkce 4 z 5 41
Obrázek 3.23: Symetrická funkce 5 z 5 41
Obrázek 3.24: Vzhled FB SBC1 42
Obrázek 4.1: Automat typu Mealy 44
Obrázek 4.2: Automat typu Moore 45
Obrázek 4.3: a) Vývojový diagram (Mealy) b) Vývojový diagram (Moore) 45
Obrázek 4.4: Ukázka jazyka SFC 46
Obrázek 4.5: Ukázka simultánního větvení jazyka SFC 47
Obrázek 4.6: a) Počáteční krok b) Přechod 48
Obrázek 4.7: a) Divergentní AND b) Konvergentní AND 49
Obrázek 4.8: a) Divergentní OR b) Konvergentní OR 49
Obrázek 4.9: Vzhled bloku mechanického ovládání 50
Obrázek 4.10: Diagram mechanického ovládání SFC 50
Obrázek 4.11: Vzhled bloku dlouhý a krátký stisk 51
Obrázek 4.12: Diagram rozpoznání tlačítka SFC 51

Obrázek 4.13: Vzhled bloku rozpoznání dvojkliku 52
Obrázek 4.14: Diagram rozpoznání dvojkliku SFC 52
Obrázek 4.15: Ukázka simulace života v domě 54
Obrázek 4.16: a) Výškový úhel b) Azimutální úhel 55
Obrázek 4.17: a) Denní řáry b) Hodinové čáry 56
Obrázek 4.18: Výškový a azimutální úhel 57
Obrázek 4.19: Poloha slunce na obloze od 22.12 do 21.6 57
Obrázek 4.20: Poloha slunce na obloze od 21.6 do 22.12 58
Obrázek 4.21: Poloha slunce na obloze 60

Seznam tabulek

Tabulka 1.1: Porovnání všech typů PLC automatů firmy Teco, a.s. 10
Tabulka 3.1: Popis funkce třívstupového majoritního členu 36
Tabulka 3.2: Popis funkce pětivstupového majoritního členu 37
Tabulka. 4.1: Výškový úhel 59
Tabulka. 4.2: Azimutální úhel 59

Kapitola 1

1. TEORETICKÁ ČÁST

1.1 Úvod

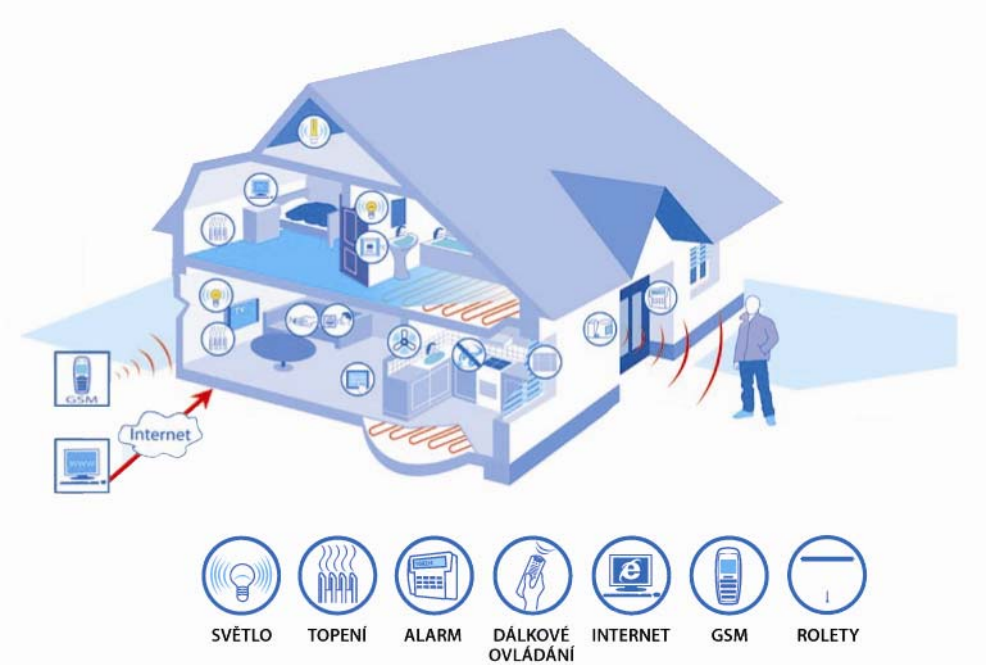
Inteligentní dům neboli chytrý dům je budova vybavená počítačovou a komunikační technologií. Tato technologie je schopna poskytovat svým obyvatelům komfort, zábavu a bezpečí. Inteligentní dům se dá taktéž charakterizovat jako budova s integrovanými řídicími systémy. Za nemalého přispění řídící techniky lze ovládat nejnovější technologie, jako je audiovizuální technika, klimatizace, kamerový systém či osvětlení. Vzájemnou integrací a synchronizací všech prvků pak dokážeme z „obyčejného“ domu udělat „inteligentní“ dům. Koncept inteligentního domu je velmi dobře propracován a obrovský důraz je kladen na to, aby jednotlivé systémy dokonale spolupracovaly. Klíčovou roli proto hraje komunikace, na které je celý systém závislý. Jednotlivé systémy potřebují sdílet informace mezi sebou, aby byla jejich integrace co možná nejúčinnější. Propojení jednotlivých systému většinou umožňuje datový kabel s obousměrnou komunikaci (UTP – unshielded twisted pair).

Inteligentní dům není dnes žádnou žhavou novinkou. Avšak díky technologickému vývoji, konkurenčnímu boji, snížení pořizovacích nákladů či zjednodušení ovládání pro případné zákazníky se objevuje „inteligentní“ instalace nejen v průmyslových budovách či bankách, ale právě také v rodinných domech.

Na rozvoji inteligentních domů má také velký podíl rozvoj komunikačních technologií (zásluhou komunikačních systémů a operátorů, vybavení PC, PLC), které mohou využívat rozhraní Ethernet, síť Internet, různé bezdrátové sítě a jsou kompatibilní s různými průmyslovými sběrnicemi. Významným faktorem rozvoje jsou i široké možnosti jednotlivých senzorů, akčních členů, prvky inteligentní elektroinstalace a zabezpečovacích systémů. Důležité jsou možnosti jejich sériové nebo bezdrátové komunikace, připojení k průmyslovým sběrnicím a v neposlední řadě i jejich cenová dostupnost. Rozvoj zaznamenávají i systémy pro řízení a monitorování.

Vzhledem k velmi rychlému rozvoji v oblasti řízení inteligentních domů je také samozřejmostí, že nejnovější vyvinuté komunikační a řídící technologie můžeme aplikovat na stávající domy bez výrazných stavebních úprav.

Donedávna byla technika inteligentních domů výsadou velkých administrativních budov, hotelů, obchodních center, jejich parkovišť, technického zázemí, osvětlení a zabezpečení venkovních prostor. V současnosti se tato technika stává obvyklou součástí i menších budov. Jako inteligentní budovy jsou často řešena komfortnější rodinná sídla a bytové domy, sídla a provozovny menších firem, zemědělské usedlosti, farmy, penziony, rekreační a sportovní objekty, úřady, školy a zdravotnická zařízení. Řízení domu často expanduje i do exteriéru a řídicí systém zajišťuje např. řízení garáže, skleníku, bazénu, otevírání vrat, závlahu záhonů a trávníků, zabezpečení venkovních prostor apod. [1]



Obrázek 1.1: Ukázka inteligentního domu

Realizace inteligentního domu nám přinese nejen již zmiňovaný komfort, ale usnadní a zpříjemní nám bydlení. Možná se už většině z nás po příchodu domů a pohodlném usazení se v křesle stalo, že jsme v domnění, že zapínáme vytouženou televizi, zmáčkli knoflík a k našemu „potěšení“ začalo hrát úplně jiné zařízení. A přesně tyto problémy řeší inteligentní dům. Je tu více možností řešení a jednou z nich je integrovat všechny funkce každého z ovladačů do jednoho ovladače. Další možností je dotykový displej, pomocí kterého můžeme ovládat televizor, klimatizaci, rolety i kamerový systém. Právě v tom je kouzlo zmíněného komfortu a pohodlí. Veškeré ovládání by mělo být intuitivní a velmi jednoduché. [5]

Inteligentní domy poskytují větší míru komfortu a bezpečnosti, především však vedou ke snížení energetické spotřeby v domech.

Nepřehlédnutelná je též velká úspora provozních nákladů, což v horizontu několika let dokáže vynulovat investice k realizaci inteligentní budovy. Paradoxně k rozšíření výstavby inteligentních a nízkoenergetických domů (přestože jsou náročnější na prvotní investiční náklady) napomáhá současná ekonomická krize. Developeři se na jejich stavbu orientují v důsledku příslibu nižších provozních nákladů dlouhodobého charakteru.

V bakalářské práci se zabývám výhradně návrhem funkčních bloků pro inteligentní domy. Navrhuji převážně takové funkční bloky, které jsou podle mého názoru nejfrekventovanější mezi typickými uživateli.

1.2 Základní pojmy a uspořádání

Pro orientaci a lepší porozumění textu zde uvedu několik důležitých pojmy, které se objevují v inteligentních budovách a také v tomto textu.

Konnex (dříve EIBA) je sdružení vedoucích firem v oblasti elektroinstalací v Evropě. Jeho cílem je nabízet jednotný systém instalační sběrnice pro řízení provozních procesů v budovách.

KNX – sběrnicový standard, který sdružuje 3 evropské technologie. Jedním z nich je francouzský ***Batibus***, evropsky normovaná komunikace pro domácí spotřebiče ***EHS*** a ***EIB*** (European Installation Bus). Byl definován v roce 2003 a stal se mezinárodním standardem pro automatizaci budov. Hlavním cílem bylo vytvoření standardu, kterým budou moci komunikovat výrobky od různých výrobců. Dnes má KNX mezinárodní nálepku standardu mezinárodně uznávaného pro automatizaci budov (ISO/IEC 14543-3). [2]

KNX RF – standard pro bezdrátovou komunikaci nabízené firmou Siemens (Synco living). Využívá kmitočtové pásmo 868 MHz až 870 MHz a je založen na modelu vrstev ISO-OSI. V budovách můžeme také vyžít kmitočtové pásmo 2,4 GHz. [3]

PROFIBUS (Process Field Bus) – průmyslová sběrnice pro komunikaci v oblasti domovní či procesní automatizaci. V roce 1996 byla přijata jako evropská norma EN 50170. Přenosová rychlost 9kbit/s až 12Mbit/s podle délky v rozsahu 1,2km až 100m a použité technologie (optika až 80km). Využívá přenosové technologie RS-485, optické vlákno nebo proudovou smyčku (IEC 1158-2). Přístup na sběrnici je řízen metodami token passing (předávání pověření v logickém kruhu) pro komunikaci mezi aktivními zařízeními, klient-server (centrálně řízené dotazování) pro komunikaci mezi aktivním a jemu přidělenými zařízeními nebo kombinací předcházejících metod. [4]

CIB – sběrnice nabízená firmou Teco, rychlost sběrnice činí 19,2kB/s, dvouvodičový systém do vzdálenosti až 1700m, odezva sběrnice dosahuje hodnot okolo 150ms

LonWorks - LON (Local Operating Network) - je průmyslová komunikační sběrnice nabízené firmou Siemens (Desing). Vyvíjena do roku 1992. Protokol užívaný pro komunikaci na této sběrnici se nazývá LonTalk. Využívá různá přenosová media jako je kroucená dvoulinka, optické vlákno, napájecí síť nebo bezdrátové spojení. Patří do skupiny decentralizovaných sběrnicových systémů. [4]

BACnet (Building Automation and Control Network) – standard pro operátorskou úroveň automatizace budov. Základní myšlenkou protokolu BACnet je formulace univerzálního popisu všech možných funkcí zařízení. Tento systém je celosvětovou normou a výkonným standardem automatizace budov. [4]

TCP/IP – Ethernet (Transmission Control Protokol/Internet Protokol) – protokol vyvinutý hlavně pro IT,tzv. sadou protokolů pro komunikaci v počítačové síti. Časem se však začala uplatňovat i v oblasti domovní automatizační techniky. Ethernet disponuje velkou přenosovou rychlostí až 100 Mbit/s. V průmyslu se používají kvalitnější materiály (konektory i kabely). Výhodou je úspora za údržbu a instalaci, jelikož v jednom systému kombinujeme řízení i správu. [4]

EHS (European Home Systems Protocol) – byl zaměřen na řízení domácích spotřebičů za podpory PLC. Sloučení se dvěma dalšími protokoly se stal součástí protokolu KNX, který je dnes standardem pro domovní elektroinstalaci.

EIB (European Installation Bus) – sběrnicový systém, který vznikl ze sběrnice INSTABUS ve firmě Siemens. EIB je inteligentní systém určený především pro regulaci, spínání, měření a předávání hlášení v budovách. Na sběrnici jsou připojena jednotlivá zařízení, která si mezi sebou vyměňují zprávy – telegramy. Každý telegram obsahuje informaci o úkonu, který je potřeba provést, a také adresu zařízení, která mají daný úkon vykonat. [2][3]

LCN (Local Control Network) - německý systému vyvíjen ve firmě Issendorff. Je to velmi jednoduchý a vysoce spolehlivý systém. Snahou bylo rozšířit se do zahraničí, bohužel konkurence v podobě KNX je obrovská. [2] [4]

Luxmate - promyšleným systémem pro řízení osvětlení od rakouské firmy Zumtobel. Systém je decentralizovaný rozdělený na tři odvětví: Basic, Emotion a Professional. Zajímavou funkcí je možnost instalace speciálního softwaru na osobní počítače jednotlivých uživatelů, přes které je následně možné přístupem k centrálnímu počítači po počítačové síti rovněž ovládat příslušné obvody osvětlení. [2]

Nikobus (Xcomfort) – systém domovní automatizace nabízené firmou Möeller. Jádrem tohoto řešení je decentralizovaný sběrnicový systém belgické firmy Niko nazývaný Nikobus. Pro svoji jednoduchost a uživatelskou dostupnost nachází vhodné uplatnění v méně rozlehlých stavbách.

Sekvenční logický obvod - je složen ze dvou částí kombinační a paměťové. Abychom mohli určit hodnotu výstupní proměnné, je potřeba u sekvenčních obvodů sledovat kromě vstupních proměnných ještě jeho vnitřní proměnné – vnitřní stav. Jsou to proměnné, které jsou uchovány v paměťových členech. Existence vnitřních proměnných způsobuje, že stejné hodnoty vstupních proměnných přivedené na vstup obvodu, nevyvolávají vždy stejnou odezvu na výstupu obvodu. [4]
Existují dva typy sekvenčních obvodů:

- **Mealyho typ** – hodnota výstupní proměnné je závislá jak na hodnotách vstupních proměnných, tak na vnitřních proměnných.
- **Moorův typ** – hodnota výstupní proměnné je přímým obrazem stavu vnitřních proměnných.

Sekvenční obvody dělíme na synchronní a asynchronní.

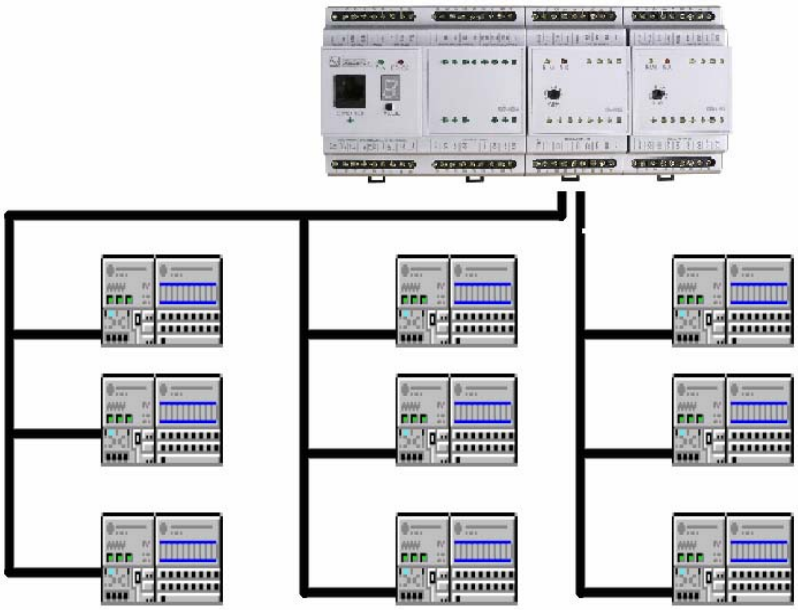
1.3 Uspořádání řídících systémů

V současné době existují dva základní typy sběrnicových systémů nebo jejich kombinace. Jeden z nich se nazývá centralizovaný systém a druhý decentralizovaný. Kombinaci těchto dvou systémů nazýváme hybridním systémem. Stupeň centralizace závisí na počtu lokalit inteligentních prvků odpovídajících za řídicí a monitorovací funkce.

1.3.1 Centralizované systémy

Centralizované systémy jsou koncipovány tak, že veškerou inteligenci a řídící jednotky mají spojeny do jednoho centrálního místa. Ve většině případů to je centrální jednotka, která je uložena na bezpečném místě, kde se minimalizují možnosti vzniku poruch či neočekávaných událostí. Tato centrální jednotka proto musí komunikovat s každým prvkem systému samostatně. Vysílá a přijímá signály, které určují co každá z jednotek má v daném okamžiku vykonat. K zajištění tohoto je nutné mít rozvod kabelů po celém domě. Tímto se nám ale zvyšují náklady na pořízení.

Hlavní výhodou plně centralizovaných systémů je možnost využití a vyhodnocování dat o dění v objektu. Obrovskou nevýhodou je závislost jednotlivých systémů pouze na jedné jediné řídící jednotce. Jde o hlavní příčinu menší stability systému. Ryze centralizované systémy nejsou schopny plošně a efektivně pokrýt celou rozlohu budovy a díky vysokým nákladům výrazně převyšují výhody těchto systémů.



Obrázek 1.2: Schéma centralizovaného systému

Výhody centralizovaného systému

- Všechny programy na jednom místě
- Lépe se řeší systémové problémy
- Rychlý přístup na I/O

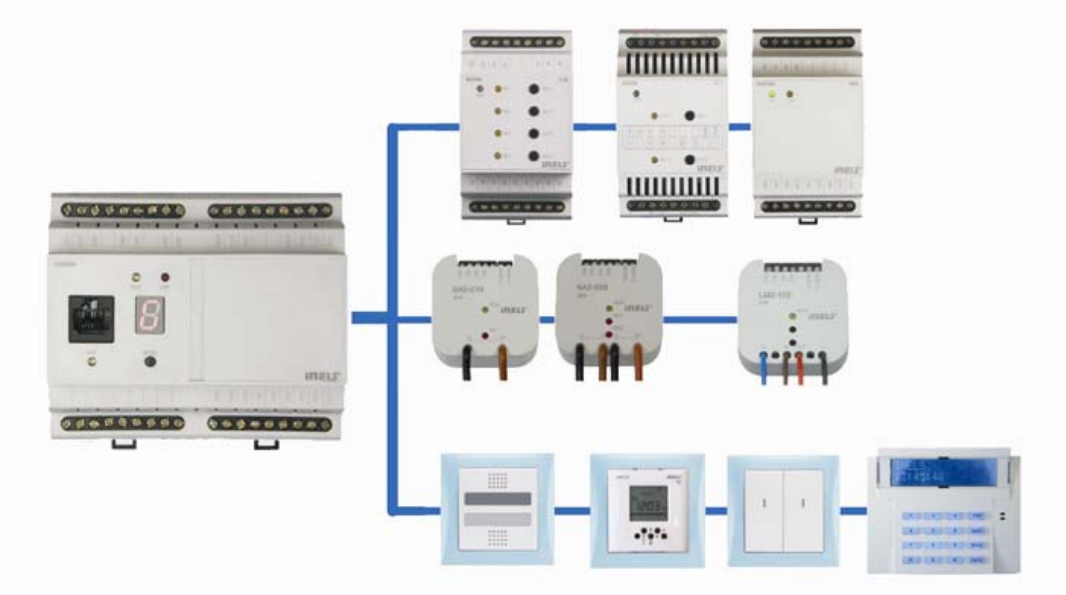
Nevýhody centralizovaného systému

- Velké programy
- Výkon závisí na počtu I/O
- Závada jednoho PLC znamená zastavení celého programu [10]

1.3.2 Decentralizované systémy

Decentralizované systémy jsou koncipovány tak, že veškerou inteligenci mají integrovanou v jednotlivých členech systému. Proto oproti centralizovanému systému nepotřebují žádnou centrální jednotku.

Inteligentní domy jsou pro nasazení decentralizovaných systémů vzhledem ke své prostorové rozlehlosti více než vhodné, tudíž jsou využívanější než centralizovaný systém. Má-li být zajištěn dokonalý přehled a kontrola nad jednotlivými technologiemi včetně jejich optimálního spolupůsobení, musí být lokální regulace organicky začleněna do systému řízení celého objektu. S decentralizovanými řídicími a informačními systémy se v poslední době setkáváme stále častěji. Zasloužil se o to prudký rozvoj polovodičových technologií, jejich neustálé zlevňování a poptávka po stále větším komfortu, bezpečnosti a ekonomické efektivitě.



Obrázek 1.3: Schéma decentralizovaného systému

V celosvětovém měřítku se u decentralizovaných systémů nejčastěji používají technologie: **LON** (Local Operating Network), **EIB** (European Installation Bus), který po konvergenci se systémy **EHS** a **Batibus** vytváří nově standard **KNX** (Konnex), a německý systém **LCN**. Pro řízení osvětlení má velký význam systém **Luxmate** společnosti Zumtobel. Zajímavé možnosti skýtá také systém **Xcomfort** společnosti Moeller. Plně decentralizovanou komunikaci využívají například zabezpečovací systémy (požární signalizace, zabezpečovací či docházkové systémy apod.). Na stejném principu funguje i standard pro sběr dat z měřičů spotřeby energií M-bus.

Vzniká zde několik problémů, které lze samozřejmě efektivně vyřešit, ale s ohledem na minimalizaci nákladů je mnohem výhodnější využití hybridních systémů, které dokáží spojit výhody centralizovaného a decentralizovaného řízení. [2] [5] [10]

Výhody decentralizovaného systému

- Rozdělení programu podle úloh
- Snáze se hledají lokální problémy
- Zvýšení výkonu na lokální síti

Nevýhody decentralizovaného systému

- Sběr dat
- Dálkové ovládání a vizualizace
- Řízení větších celků
- Vyšší cena

1.3.3 Hybridní systémy

Hybridní systémy jsou koncipovány jako kombinace výše zmíněných systému (centralizovaný a decentralizovaný) a mají být účelným kompromisem mezi centralizovaným a decentralizovaným řízením. V inteligentních domech je jejich nasazení méně časté, ale velice výhodné, protože spojují výhody obou typů řízení a minimalizují tak náklady na celý řídící systém. Systém tedy obsahuje centrální řídící jednotku a „chytré“ prvky domovní elektroinstalace. Jejich komunikace probíhá na bázi decentralizovaného systému a využívají přitom topologie, kdy prvky spolu komunikuji navzájem a centrální jednotka je pouze sleduje.

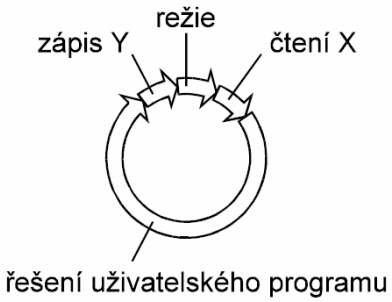
1.4 Programovatelný automat - PLC

Programovatelný automat, dále PLC (Programmable Logic Controller = programovatelný automat) je malý průmyslový počítač používaný pro automatizaci procesů v reálném čase. PLC se oproti normálnímu PC liší například tím, že vykonává program v cyklech. Jejich periferie jsou předem připraveny tak, aby mohly být propojeny s technologickými procesy.

1.4.1 Princip vykonávání uživatelského programu

Řídící algoritmus programovatelného automatu je zapsán jako posloupnost instrukcí v paměti uživatelského programu. Centrální jednotka postupně čte z této paměti jednotlivé instrukce, provádí příslušné operace s daty v zápisníkové paměti a zásobníku, případně provádí přechody v posloupnosti instrukcí, je-li instrukce ze skupiny organizačních instrukcí. Jsou-li provedeny všechny instrukce požadovaného algoritmu, provádí centrální jednotka aktualizaci výstupních proměnných do výstupních periferních modulů a aktualizuje stavy ze vstupních periferních modulů do zápisníkové paměti. Tento děj se stále opakuje a nazýváme jej cyklem programu (obr.1.4).

Jednorázová aktualizace stavů vstupních proměnných během celého cyklu programu odstraňuje možnosti vzniku hazardních stavů při řešení algoritmu řízení (během výpočtu nemůže dojít ke změně vstupních proměnných). [7]



Obrázek 1.4: Cyklus řešení uživatelského programu

čtení X - přepis hodnot ze vstupních modulů PLC do oblasti X v zápisníkové paměti
zápis Y - přepis hodnot vypočtených programem z oblasti Y do výstupních modulů PLC
režie - příprava centrální jednotky PLC k řešení dalšího cyklu programu

1.4.2 PLC Tecomat

Programové automaty Tecomat jsou určené pro řízení nejrůznějších strojů z oblasti průmyslu, jako je strojírenství nebo hutnictví. Dále také z oblasti chemického, vodohospodářského a potravinářského průmyslu apod. Tyto automaty patří mezi světové standarty PLC. Jelikož náplní této práce není porovnávání PLC automatů různých značek, musí být vyzdviženy ty přednosti, kterými disponuje firma Teco a její automaty. Jeden z hlavních faktorů proč doporučit tuto variantu je jednoznačně software, který firma poskytuje zdarma ke svým automatům, a možnost v tomto programovém vybavení jménem Mosaic programovat podle mezinárodního standartu IEC EN 61 131-3. Firma nabízí automaty pro široké využití, ať už jsou to automaty, které mají několik vstupů a výstupů (tzv. kompaktní systémy – TC400), přes větší kompaktní systémy TC500, TC600 a TC650, až po velké aplikace pro které jsou určeny PLC typu TC700 a NS950. Porovnání všech typů naleznete v tabulce 1.

Nejvíce populárním se, ať už dnes nebo v blízké budoucnosti, stává automat typu Foxtrot, který v porovnání s ostatními vykazuje malé rozměry při obrovské „síle“. V případě použití PLC pro inteligentní budovy bude využit ten systém, jehož mozkem bude právě již zmiňovaný Foxtrot.

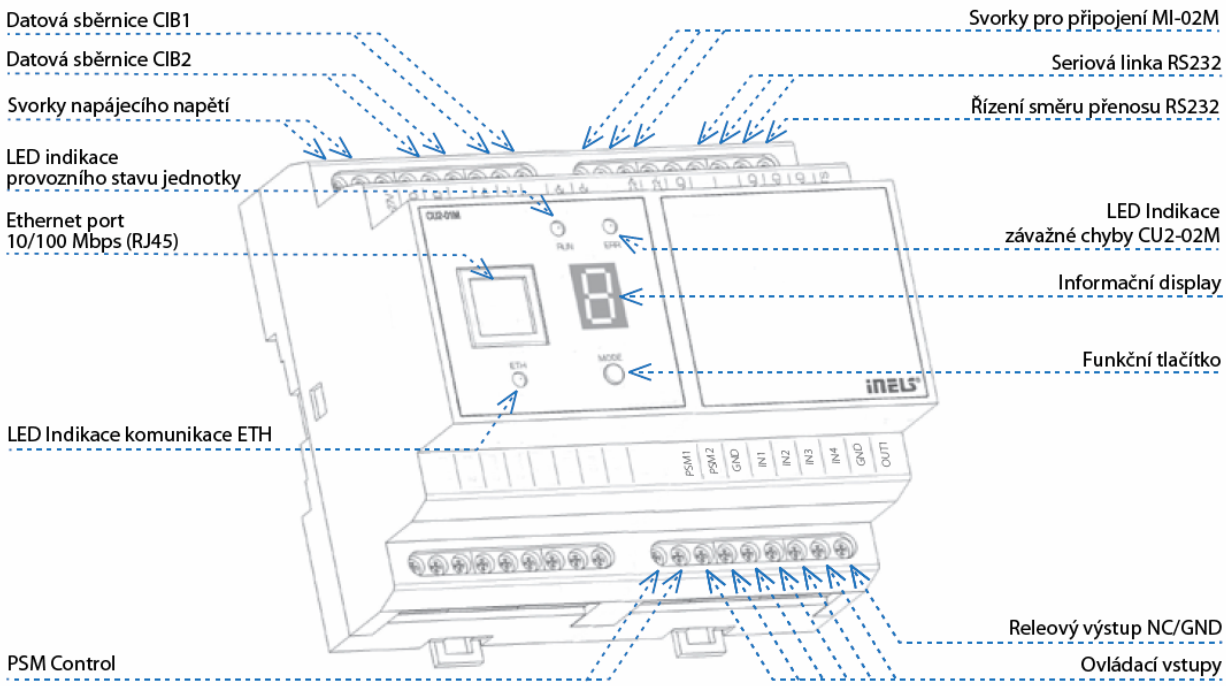
Typová řada	Binární I/O	Analogové I/O	Komunikační kanály	Displej/ klávesnice	Paměť programu
TC400	6/4	2/-	2 serial	externí	32 kB
TC500	20/20	4/4	2 serial	integrovaný	32 kB
TC600	48/44	24/8	3 serial	externí	32 kB
TC650	48/44	24/8	3 serial, 1 Ethernet	externí	64 + 64 kB
Foxtrot	až 134 DI/DO	až 80AI / 20AO	2× serial, 1×CIB, 1×Ethernet	externí	192 + 64 kB
TC700	přes 6500	až 3300/700	až 10 serial, 2× Ethernet, 1× USB	externí	192 + 64 kB

Tabulka 1.1: Porovnání všech typů PLC automatů firmy Teco, a.s.

1.4.3 Tecomat Foxtrot

K mým testovacím zařízením patří Tecomat Foxtrot, a proto zde popíši jeho funkce a základní údaje o něm. Tecomat Foxtrot je malý modulární řídící a regulační systém navržený (tzv. kompaktní PLC) pro snadnou instalaci do rozvaděčů na U-lištu. Je propojen sběrnicí TCL2. Tato sběrnice je tvořena linkou RS-485. Tvoří distribuovaný systém, kde se může poslední I/O modul nacházet ve vzdálenosti až 300m od centrálního modulu (použití optického kabelu až 1,5 km). Obsahuje 32 bitový procesor RISC s frekvencí 166 MHz, proto doba cyklu trvá 0,2 ms na 1000 logických instrukcí.

Paměť pro program činí 192 kB pro vlastní program a 64 kB pro tabulky, paměť flash EEPROM pro zálohování projektů 2 GB a nakonec je možnost k automatu přidat kartu MMC/SD do velikosti více než 4 GB. Základní modul obsahuje provedení analogové, digitální nebo univerzální vstupy, ethernetový konektor RJ-45 pro 100 Mb ethernet a jednu větev CIB sběrnice pro až 32 jednotek inteligentní elektroinstalace INELS. Dále má 2 komunikační kanály, první kanál CH1 je rozhraní RS-232 a druhý kanál CH2 je rozhraní volitelné submodelem. [8]



Obrázek 1.5: Základní modul Tecomat Foxtrot spolu s připojenými binárními moduly

1.4.4 Sběrnicový systém INELS

Sběrnicový systém INELS vznikl za vývojové a výrobní spolupráce firem Teco, a. s., a Elko EP, s. r. o. je založen na dvouvodičové instalační sběrnici CIB. Pomocí níž jsou propojeny všechny náležitosti systému.

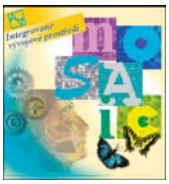
Sběrnice CIB (Common Installation Bus) se vyznačuje velice snadnou instalací. Ta je omezena na dvouvodičový kabel, který propojuje jednotlivé akční členy a postačí dodržení polarity vodičů. Jednou z výhod této sběrnice je také minimalizace počtu vodičů, přičemž napájení a data jsou vedena společně po již zmiňovaných dvou vodičích. Z toho důvodu není třeba řešit, jak napájet jednotlivé jednotky připojené na sběrnici. Jednou z dalších výhod je snadná rozšiřitelnost, konfigurovatelnost a velký dosah.

Obrázek 1.6: Využití systému INELS se sběrnicí CIB v inteligentním domě

Po vyjmenování mnoha výhod přejdeme k reálným datům, jako je rychlost sběrnice, která činí 19,2 kB/s. Na jednu větev lze připojit až 32 jednotek (lze rozšířit až na 288 jednotek), modul master lze umístit do vzdálenosti 300 m pomocí metalického vedení nebo optickým kabelem do vzdálenosti 1700 m. Odezva sběrnice dosahuje hodnot okolo 150 ms, tedy doby, kdy člověk není schopen zareagovat. Pokud by měl být schopen odezvy, musela by se tato hodnota pohybovat nad 300 ms. Rychlost 150 ms je velmi výhodná například při řízení osvětlení. Každá jednotka má svoji vlastní unikátní šestnáctibitovou adresu.

Jako poslední bych uvedl napájení sběrnice CIB. Její nominální hodnota je 24 V DC, ale je zde další možnost použití dobíjecí bateriové verze (2x12V), která zajistí chod centrální jednotky i se všemi jednotkami na sběrnici CIB i za výpadku sítě (zde se doporučuje napájení 27,2 V DC).[6]

1.4.5 Vývojové prostředí MOSAIC



Velká většina výrobců PLC má svůj vlastní software k programování automatů pomocí PC. Výjimkou není ani firma Teco a.s., která ovšem svůj software nabízí zcela zdarma. Vývojové prostředí nesoucí jméno Mosaic je uživatelsky velice přívětivé (skoro bych řekl „user friendly“) a poskytuje značný komfort při tvorbě vlastních programů, dokumentování, jejich ladění či diagnostice. Neméně důležitou vlastností prostředí Mosaic je možnost programovat a ladit PLC bez jeho fyzické přítomnosti (SimPLC).

Programová forma je ve shodě s mezinárodní normou IEC EN 61 131-3 v textových jazyce IL (Instruction List = firemní mnemokód) a ST (Structured Text = strukturovaný text) a grafických jazycích LD (Ladder Diagram = reléové schéma) a FBD (Function Block Diagram = funkční bloky).

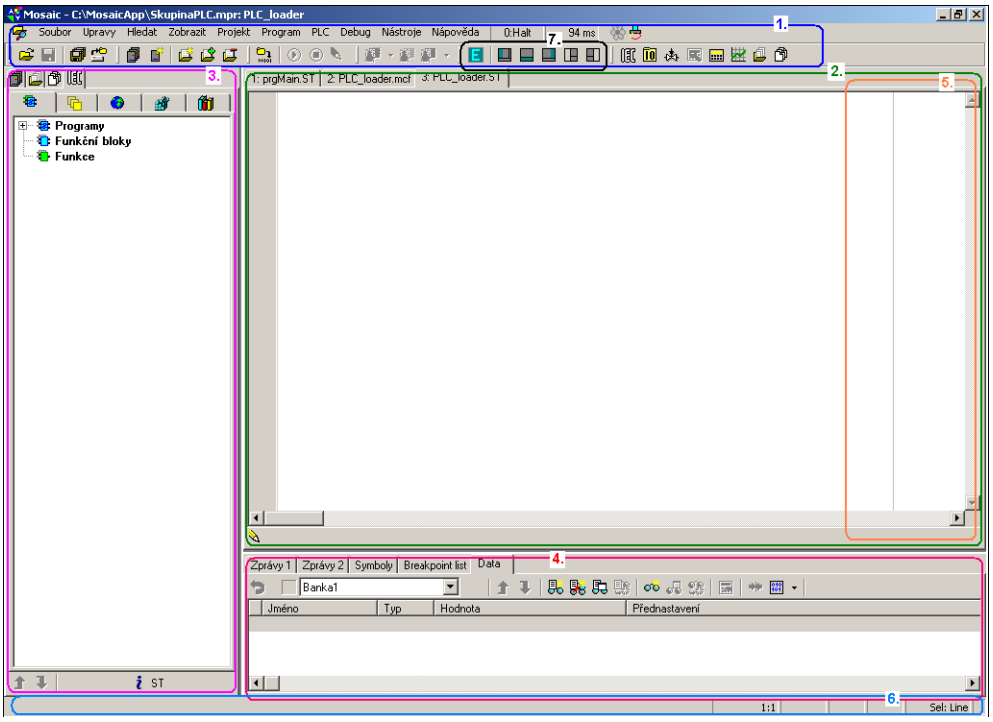
Program se skládá z programové organizační jednotky (POU = Program Organisation Unit). Těmito jednotkami jsou funkce, funkční bloky a nejvyšší jednotkou je program (IL, ST, LD nebo FBD). Tyto jazyky je možno při tvorbě programu mezi sebou kombinovat. Také jsou ve vývojovém prostředí integrovány standardní i uživatelské knihovny funkcí a funkčních bloků.

Mosaic také umožňuje komunikaci s řídícím systémem přes sériovou linku, Ethernet či USB. V prostředí je zahrnuta i podpora pro vytáčené připojení přes telefonní nebo GSM modem a poslední době oblíbené spojení přes Wi-Fi, která umožňuje dálkovou správu.

Stručný obsah nástrojů, které obsahuje vývojové prostředí Mosaic:

- PanelMaker
  - nástroj na tvorbu dialogů pro operátorské panely ID-07, ID-08 a PLC řady TC500 a TR200
  - program pro panel je součástí programu pro PLC
- PanelSim
  - simulátor operátorských panelů, dovoluje zkoušet dialogy vytvořené PanelMakerem bez připojení skutečného panelu, funguje jak s reálným, tak i simulovaným PLC
- PIDMaker
  - nástroj pro ladění a návrh PID regulátorů
  - nabízí interaktivní náhled na průběh regulace, usnadňuje správné nastavení parametrů regulátoru a generuje programový kód
  - součástí je simulace jednoduchých soustav do třetího řádu s dopravním zpožděním

- GraphMaker
    - nástroj pro podporu ladění a diagnostiku řízeného systému umožňuje zobrazení průběhů vybraných proměnných offline i v reálném čase
    - dva sledovací kurzory, nastavitelná perioda vzorkování, umožňuje ukládání dat na disk i export do DB programů
    - funkce digitálního osciloskopu (16 kanálů) a logického analyzátoru.



Obrázek 1.7: Ukázka vývojového prostředí Mosaic

Popis obrázku 1.4.3

1. *V horní části hlavního okna je v řádku hlavní nabídka, textové menu prostředí Mosaic a hned pod ním hlavní nástrojová lišta s grafickými ikonami.*
2. *Ve střední části je panel editoru. V horní části okna jsou záložky se jmény otevřených souboru.*
3. *V levé části hlavního okna je panel pomocných organizačních nástrojů.*
4. *V dolní části hlavního okna je panel informačních nástrojů.*
5. *V pravé části hlavního okna je náhled na oblasti paměti a proměnných v PLC.*
6. *V nejnižším řádku hlavního okna je informační řádek, kde se zobrazují informační texty a v pravé části informace z aktivního editoru, jako je číslo řádek : sloupec a pracovní režimy editoru.*
7. *Skupina ovládacích ikon pro rychlé vypínání/zapínání a změnu rozměru jednotlivých panelu*

1.5 Mezinárodní norma IEC EN 61 131-3

Norma IEC 61 131 pro programovatelné řídící systémy má pět základních částí a představuje souhrn požadavků na moderní řídící systémy. Je nezávislá na konkrétní organizaci či firmě a má širokou mezinárodní podporu. Jednotlivé části normy jsou věnovány jak technickému tak programovému vybavení těchto systémů.
V ČR byly přijaty jednotlivé části této normy pod následujícími čísly a názvy:
ČSN EN 61 131-1 (Všeobecné informace), ČSN EN 61 131-2 (Požadavky na zařízení a zkoušky), ČSN EN 61 131-3 (Programovací jazyky),ČSN EN 61 131-4 (Podpora uživatelů), ČSN EN 61 131-5 (Komunikace), ČSN EN 61 131-7 (Programování fuzzy řízení). V Evropské unii jsou tyto normy přijaty pod číslem EN IEC 61 131.

Na normu 61 131-3 je možné pohlížet z různých hledisek, např. tak, že je to výsledek náročné práce sedmi mezinárodních společností, které do vypracování normy vložily svoji desetiletou zkušenost na poli průmyslové automatizace. Výsledkem je specifikace syntaxe a sémantiky unifikovaného souboru programovacích jazyků, včetně obecného softwarového modelu a strukturujícího jazyka. Tato norma byla přijata jako směrnice u většiny významných výrobců PLC.

Podrobné informace o mezinárodní normě IEC EN 61131-3 jsou uvedeny v [7]. Moji snahou bude uvést pouze základní myšlenky a vlastnosti mezinárodní normy, které by mohli patřit k základním poznatkům nejen v mé práci, ale i pro náhodné čtenáře této práce.

1.5.1 Společné prvky

1.5.1.1. Typy dat

V rámci společných prvků jsou definovány typy dat. Definování datových typů napomáhá prevenci chyb v samém počátku tvorby projektu. Je nutné definovat typy všech použitých parametrů. Běžné datové typy jsou `BOOL`, `BYTE`, `WORD`, `INT` (Integer), `REAL`, `DATE`, `TIME`, `STRING` atd. Z těchto základních datových typů je pak možné odvozovat vlastní uživatelské datové typy, tzv. odvozené datové typy.

1.5.1.2. Proměnné

Proměnné mohou být přiřazeny explicitně k hardwarovým adresám (např. vstupům, výstupům) pouze v konfiguracích, zdrojích nebo programech. Tímto způsobem je dosaženo vysokého stupně hardwarové nezávislosti a možnosti opakovaného využití softwaru na různých hardwarových platformách.

Oblast působnosti proměnných je běžně omezena pouze na tu programovou organizační jednotku (POU), ve které byly deklarovány (proměnné jsou v ní lokální). To znamená, že jejich jména mohou být používána v jiných částech bez omezení. Tímto opatřením dojde k eliminaci řady dalších chyb. Pokud mají mít proměnné globální působnost, např. v rámci celého projektu, pak musí být jako globální deklarovány (blok `VAR_GLOBAL`). Aby bylo možné správně nastavit počáteční stav procesu nebo stroje, může být parametrům přiřazena počáteční hodnota při startu nebo studeném restartu.

1.5.2 Programovací jazyky

Jak již bylo zmíněno v předchozí části popisu vývojového prostředí Mosaic, tak v rámci standardu jsou definovány čtyři programovací jazyky. Jejich sémantika i syntaxe je přesně definována a neponechává žádný prostor pro nepřesné vyjadřování. Zvládnutím těchto jazyků se tak otevírá cesta k používání široké škály řídicích systémů, které jsou na tomto standardu založeny.

Programovací jazyky se dělí do dvou základních kategorií:

Textové jazyky
IL - Instruction List - jazyk seznamu instrukcí
ST - Structured Text - jazyk strukturovaného textu

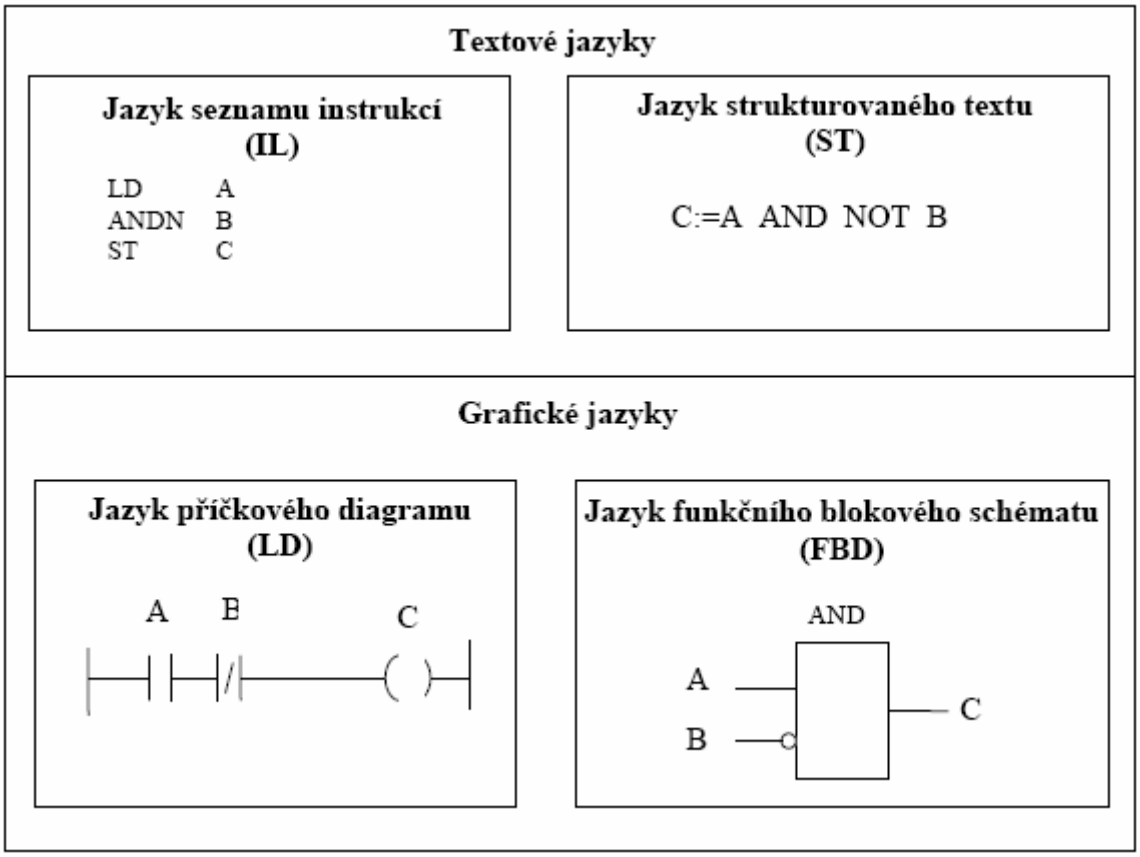
Grafické jazyky
LD - Ladder Diagram - jazyk příčkového diagramu (jazyk kontaktních schémat)
FBD - Function Block Diagram - jazyk funkčního blokového schématu

Volba programovacího jazyka je závislá na zkušenostech programátora, na typu řešeného problému, na úrovni popisu problému, na struktuře řídicího systému a na řadě dalších okolností, jako jsou např. typ odvětví průmyslu, zvyklosti firmy implementující řídicí systém, zkušenosti spolupracovníků v týmu apod. Všechny čtyři základní jazyky (IL, ST, LD a FBD) jsou vzájemně provázány, to znamená, že je možné část programu udělat v jazyce ST a zbytek doprogramovat například v LD.

Jazyk LD - jazyk příčkového diagramu má původ v USA a je založen na grafické reprezentaci reléové logiky. Naproti tomu jazyk IL - jazyk seznamu instrukcí - je jeho

evropský protějšek. Jako textový jazyk připomíná assembler.Předposlední z jazyků FBD - jazyk funkčního blokového schématu je velmi blízký procesnímu průmyslu. Vyjadřuje chování funkcí, funkčních bloků a programů jako soubor vzájemně provázaných grafických bloků, podobně jako v elektronických obvodových diagramech. Je to určitý systém prvků, které zpracovávají signály. A poslední jazyk ST - jazyk strukturovaného textu je velmi výkonný vyšší programovací jazyk, podobný Pascalu nebo C. Obsahuje všechny podstatné prvky moderního programovacího jazyka, včetně větvení IF, ELSE, CASE a iterační smyčky FOR, WHILE a REPEAT. Tyto prvky mohou být vnořovány. Tento jazyk je vynikajícím nástrojem pro definování komplexních funkčních bloků, které pak mohou být použity v jakémkoliv jiném programovacím jazyku.

Nejlépe vyjádřený popis každého z jazyků je uveden na obrázku 1.5.1



Obrázek 1.8: Ukázka logické funkce AND ve všech čtyřech jazycích

1.5.3 Programové organizační jednotky

Funkce, funkční bloky a programy jsou v rámci normy IEC 61 131-3 nazývány společně *programové organizační jednotky* (Program Organization Units, někdy se pro tento důležitý a často používaný pojem používá zkratka POUs). *)*. Jak vyplývá z názvu, POU je

nejmenší nezávislá část uživatelského programu. POU mohou být dodávány od výrobce řídícího systému nebo je může napsat uživatel. Každá POU může volat další POU a při tomto volání může volitelně předávat volané POU jeden nebo více parametrů.

Existují tři základní typy POU :

- *funkce (function, FUN)*
- *funkční blok (function block, FB)*
- *program (program, PROG)*

1.5.4 Funkce a její syntaxe

Nejjednodušší POU je ***funkce***, jejíž hlavní charakteristikou je to, že pokud je volána se stejnými vstupními parametry, musí produkovat stejný výsledek (funkční hodnotu). Funkce může vracet pouze jeden výsledek.

IEC 61 131-3 definuje standardní funkce a uživatelem definované funkce. Standardní funkce jsou např. `ADD` pro sčítání, `ABS` pro absolutní hodnotu, `SQRT` pro odmocninu, `SIN` pro sinus a `COS` pro cosinus. Jakmile jsou jednou definovány nové uživatelské funkce, mohou být používány opakovaně.

1.5.5 Funkční bloky a syntaxe

Dalším typem POU je ***funkční blok***, který si na rozdíl od funkce, může pamatovat některé hodnoty z předchozího volání (např. stavové informace). Ty pak mohou ovlivňovat výsledek. Hlavním rozdílem mezi funkcí a funkčním blokem je tedy schopnost funkčního bloku vlastnit paměť pro zapamatování hodnot některých proměnných. Tuto schopnost funkce nemají a jejich výsledek je tedy jednoznačně určen vstupními parametry při volání funkce. Funkční blok může také (na rozdíl od funkce) vracet více než jeden výsledek.

Jakmile je jednou funkční blok definován, může být používán opakovaně v daném programu, nebo v jiném programu, nebo dokonce i v jiném projektu. Je tedy univerzální a mnohonásobně použitelný. Funkční bloky mohou být zapsány v libovolném z jazyků definovaném v normě. Mohou být tedy plně definovány uživatelem. Odvozené funkční bloky jsou založeny na standardních funkčních blocích, ale v rámci pravidel normy je možno vytvářet i zcela nové zákaznické funkční bloky.

Na funkční bloky se můžeme dívat jako na integrované obvody, které reprezentují hardwarové řešení specializované řídicí funkce. Obsahují algoritmy i data, takže mohu

zachovávat informaci o minulosti, (tím se liší od funkcí). Mají jasně definované rozhraní a skryté vnitřní proměnné, podobně jako integrovaný obvod nebo černá skříňka. Umožňují tím jednoznačně oddělit různé úrovně programátorů nebo obslužného personálu. Klasickými příklady funkčního bloku jsou např. regulační smyčka pro teplotu nebo PID regulátor.

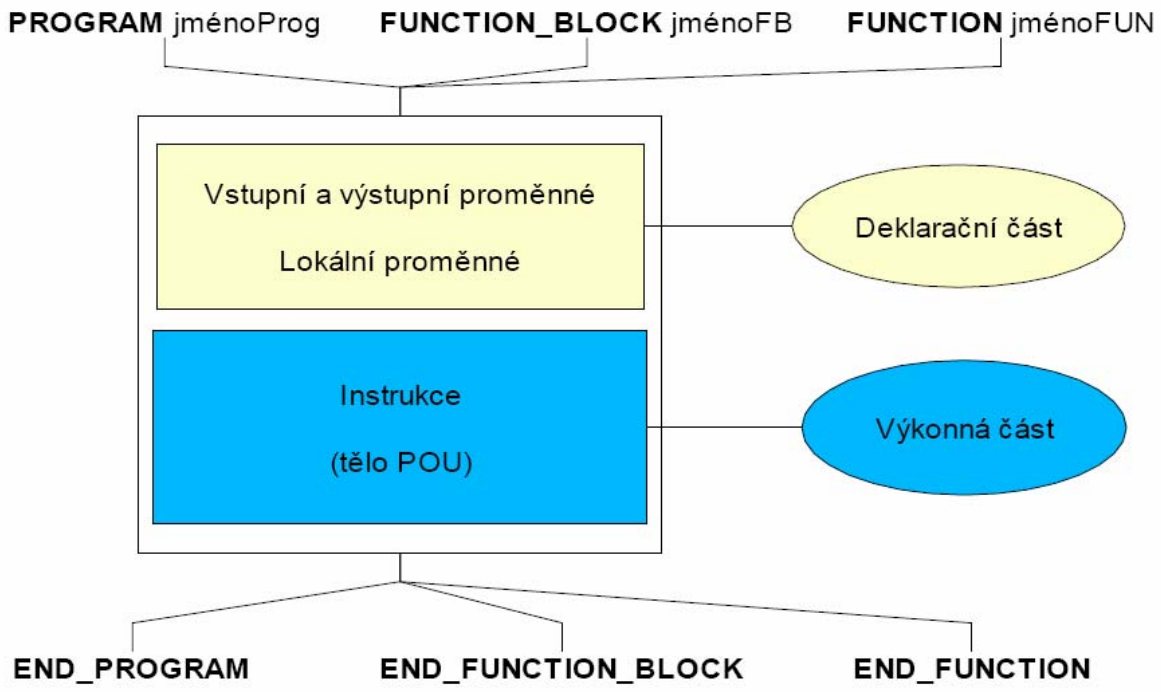
1.5.6 Programy

Posledním typem POU je ***program***, který představuje vrcholovou programovou jednotku v uživatelském programu. Centrální jednotka PLC může zpracovávat více programů a programovací jazyk ST obsahuje prostředky pro definice spouštění programů (v jaké periodě vykonávat program, s jakou prioritou, apod.).

Na základě výše uvedených definic lze říci, že program je vlastně sítí funkcí a funkčních bloků. Program může být zapsán v libovolném z jazyků definovaných výše.

1.5.7 Struktura programové organizační jednotky (POU)

Každá POU se skládá ze dvou základních částí : *deklarační* a *výkonné*, jak je vidět na Obr. 1.5.2. V deklarační části POU se definují proměnné potřebné pro činnost POU. Výkonná část pak obsahuje vlastní příkazy pro realizaci požadovaného algoritmu.



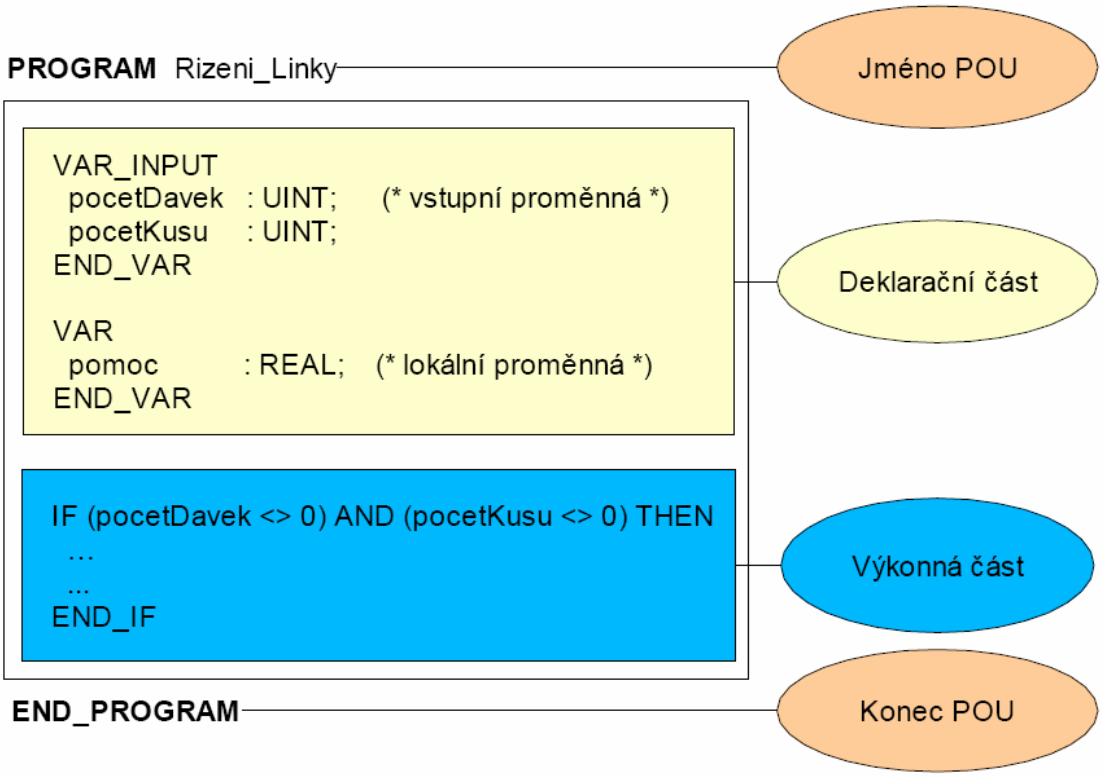
Obrázek 1.9: Základní struktura POU

1.5.7.1. Deklarační část POU

Deklarační část POU obsahuje definice proměnných potřebných pro činnost POU. Každá proměnná je definována jménem proměnné a datovým typem

Proměnné můžeme rozdělit podle použití na globální a lokální. Globální proměnné jsou definovány vně POU a mohou být použity v libovolné POU (jsou viditelné z libovolné POU). Lokální proměnné jsou definovány uvnitř POU a v rámci této POU mohou být používány (z ostatních POU nejsou viditelné). Proměnné lze rozdělit na 4 základní deklarační bloky jako je VAR_INPUT, VAR_OUTPUT, VAR a VAR_TEMP. Každý blok je ukončen END_VAR.

Na Obr. 1.5.3 je vidět, že vše začíná klíčovým slovem PROGRAM a je ukončena klíčovým slovem END_PROGRAM. Tato klíčová slova vymezují rozsah POU. Za klíčovým slovem PROGRAM je uvedeno jméno POU. Poté následuje deklarační část POU. Ta obsahuje definice proměnných uvedené mezi klíčovými slovy VAR_INPUT a END_VAR resp.VAR a END_VAR. Na závěr je uvedena výkonná část POU obsahující příkazy jazyka ST pro zpracování proměnných. Texty uvedené mezi znaky (* a *) jsou poznámky.



Obrázek 1.10: Základní struktura POU programu (deklarační i výkonná část)

1.5.7.2. Výkonná část POU

Výkonná část POU následuje za částí deklarační a obsahuje příkazy a instrukce, které jsou zpracovány centrální jednotkou PLC. Ve výjimečných případech nemusí definice POU obsahovat žádnou deklarační část a potom je výkonná část uvedena bezprostředně za definicí začátku POU. Příkladem může být POU, která pracuje pouze s globálními proměnnými, což sice není ideální řešení, ale může existovat. Na Obr. 1.5.3 si můžete všimnout modrého bloku jako ukázky výkonné části POU.

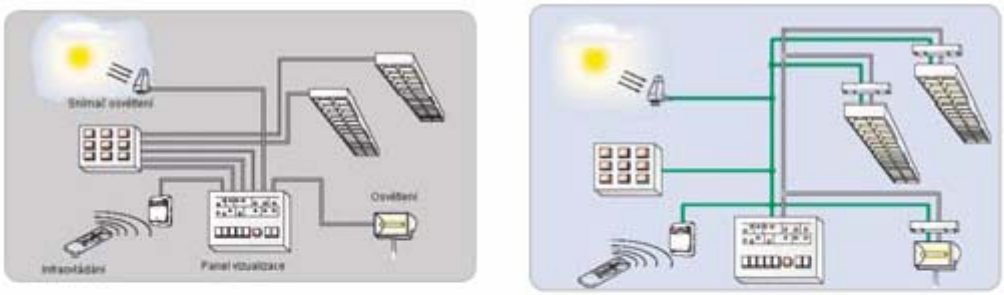
Výkonná část POU může obsahovat volání dalších POU. Při volání mohou být předávány parametry pro volané funkce respektive funkční bloky.

Kapitola 2

2. TYPICKÉ POŽADAVKY UŽIVATELŮ

2.1 Elektroinstalace klasická versus inteligentní

Budovy dneška jsou doslova prošpikovány moderní technikou. Kamerové systémy, osvětlení, vytápění, klimatizace či alarmy činí z elektroinstalace jednu z nejnáročnějších položek při rekonstrukci budovy nebo realizaci nové stavby. Hlavním cílem v oblasti domovní inteligentní elektroinstalace je řízení provozu budov jednoduše, bezpečně, s maximálním komfortem a hlavně levněji.



Uvedená ilustrace vyžaduje určité doplnění:
- Sběrnicové vedení je připojeno k napájecímu zdroji KNX/EIB a ke všem dalším přístrojům.
- Silové vedení 230 V (nebo 400 V) nevyžaduje ke svému ovládání přístroje (snímače).

Vedení sběrnice
Silové vedení

Je pouze vyžadováno pro silové napájení spotřebičů.
- Důsledkem jsou dva napájené systémy; jeden pro výkonový přenos a druhý pro přenos informací.

Obrázek 2.1: Porovnání klasické a inteligentní elektroinstalace

Můžeme si položit otázku, co vlastně uživatel chce, jaké výhody by mu „inteligentní“ elektroinstalace měla přinést? Jestliže porovnáváme „inteligentní“ a klasickou elektroinstalaci, můžeme jednoznačně říci, že klasická nás velmi omezuje. U klasické elektroinstalace je většinou pro každou funkci použit specializovaný řídicí systém. Každý vypínač má předem pevně přiřazenou funkci, která je dána kabelovými propojení. Například jeden konkrétní vypínač může ovládat jedno konkrétní světlo. Z hlediska montáže a provozních nákladů je toto řešení neefektivní a drahé. Každý si určitě řekne, že je to přirozené a úplně normální, že jeden vypínač disponuje jednou konkrétní funkcí, ale vůbec si neuvědomuje, jak je vlastně ochuzen, ba dokonce omezen. Co když je vypínač umístěn na nevhodném místě? Potom přicházejí na řadu zednické práce, sekání omítky, následná oprava a poté výmalba. Proto se většina lidí nechá touto nevýhodou omezit a raději si zvykne na polohu ovladače či způsob ovládání. Chybí také zvané pohodlné funkce jako ovládání všech světel v místnosti jedním jediným tlačítkem nebo ovládání rolet (pro jejich pohyb je nutné u klasické instalace tlačítko držet než dosáhne požadované polohy).

Naproti tomu systémy inteligentní elektroinstalace jsou navrženy tak, aby umožňovali snadné změny funkcí systému pomocí parametrizace bez zásahu do zapojení. Velkou výhodou při projektování těchto systémů je skutečnost, že není zapotřebí přesně určovat, která svítidla či jiné spotřebiče mají být z daného místa ovládány. Konkrétní konfigurace ovládání se nastavuje až při samotném oživování elektroinstalace. Všechny prvky systému, aktory (výkonová část) a senzory (řídící a ovládací část), spolu navzájem komunikují (bezdrátově či pomocí datové sběrnice) a umožňují tak efektivní ovládání topení, světel, zabezpečovacího systému, elektrických přístrojů, žaluzií či zavlažování zahrady. Například v oblasti topení lze tak dosáhnout úspor v řádu desítek procent (až 30%).

Při odchodu z domu uživatel nemusí přemýšlet, zda vypnul všechny spotřebiče či zavřel všechna okna – systém sám pomocí povelu z jediného tlačítka u vchodových dveří vypne elektrický proud v zásuvkách v celém domě (ne ovšem v těch, ke kterým jsou připojeny spotřebiče - lednice, pračka apod.), zhasne osvětlení a aktivuje zabezpečovací systém. Na uživateli tedy zůstane jen „povinnost“ neztratit klíče či nezapomenout heslo. [9]

2.1.1 Inteligentní elektroinstalace na první pohled

A jak poznáme inteligentní elektroinstalaci od elektroinstalace klasické? Jako základní a na první pohled nejsnáze pozorovatelný prvek je vypínač. Místo klasických kolébkových vypínačů (ty mají dvě polohy, stisk nahoru – zapnuto a stisk dolu – vypnuto) se používají tlačítka, která se po stisku sama vrací zpět do své původní polohy. Důvodem použití těchto ovladačů je nejspíše ten, že u inteligentní elektroinstalace vypínač neprovádí přímo ovládání světel či elektrického přístroje. Pouze posílá signál po sběrnici do centrálního zařízení.

Jestliže si zákazník přeje nemusí být instalovány jen mžikové ovládače, na binární vstupy lze připojit i kontakty tradičního (dvoustavového) vypínače, ale pak nelze realizovat mnohé „chytré“ funkce. Mžikový vypínač nemá vlastní paměť a nemůže sám o sobě realizovat čistě kombinační funkce, tak jako tradiční vypínač s mechanickou pamětí. Veškeré logické funkce, které jej využívají jako vstupní proměnnou mají proto sekvenční charakter, přinejmenším jako paměťová funkce R-S pro zapamatování a prodloužení požadavku. I další „chytré funkce“ mají sekvenční charakter.

V inteligentní elektroinstalaci je možné ovládat každé zařízení z několika míst. Toto by u elektroinstalace klasické byl problém (vypínače by se musely vracet do pozic vypnuto/zapnuto). U inteligentní instalace tento problém odpadá, jelikož stav vypnuto/zapnuto neexistuje.[1]

Obrázek 2.2: Nástěnný ovladač inteligentní elektroinstalace

2.2 Přínosy uživatelům při použití inteligentní elektroinstalace oproti klasické

Komfort

- funkce smívání (postupný náběh nebo doběh, světelné scény)
- ovládání přes dotykový displej, dálkovým ovladačem nebo hlasem (zabudovaný ve stěně, kompletní přehled)
- regulace teploty podle předem nastavených programů a to v každé místnosti samostatně
- nastavení ideálních podmínek pro spánek (postupné zhasínání, příjemná teplota, hlídání pohybu - babysiting)
- možnost ovládání přes mobilní telefon, počítač a internet

Automatizace

- nastaví elektrických spotřebičů individuálně podle uživatele
- informace o zvolené události (např. aktivace alarmu) pomocí SMS zprávou
- sdružování více funkcí už zde nazvat jako „scény“. Některé mají kombinační charakter (např. rozsvícení nebo zhasnutí několika světel současně, centrální zhasínání nebo rozsvěcování od vchodu), některé mají sekvenční charakter (např. „postupná vlna“ rozsvěcování a zhasínání pro očekávané trasy), simulace přítomnosti
- automatické reagování na povětrnostní vlivy (bouře, déšť, silný vítr, setmění, mráz, horko, rozednění)
- časové programy pro otop, klimatizaci a osvětlení, individuálně pro jednotlivé místnosti (IRC) v místnosti nebo v domě (může být i individuálně pro konkrétního člověka)
- automatické reagování na přítomnost člověka a na stav domu
- optimalizace spotřeby, využití levnějších zdrojů (světlo-rolety a žaluzie, regulace intenzity osvětlení, topení - oslunění, solární zdroj, klimatizace a vzduchotechnika - přirozené proudění, ...

Bezpečnost a diagnostika

- ochrana domu při špatném počasí (žaluzie při silném větru nebo bouřce), nečekaných událostech (poruchy v síti, přepětí, přetížení), živelných pohromách (senzor zaplavení, kouřový senzor)
- bioinstalace - vypnutí nepoužívaného elektrického okruhu (např. ložnice při spánku)
- dotykové části senzorů jsou napájené bezpečným napětím 24 V
- možnost centrálního vypínání okruhů zásuvek při odchodu z domu, nebo také signalizaci jejich stavu
- při odchodu z domu upozornění na zapomenutá otevřená okna či aktivní spotřebiče
- vícenásobné využití senzorů pro řízení a pro potřeby bezpečnosti a diagnostiky jako jsou senzory pohybu, otevření okna a dveří nebo koncentrace CO_2, využití údajů o spotřebě vody, plynu a elektřiny - zejména v případech, kdy má být nízká nebo nulová (např. zazimovaný rekreační objekt)
- reakce na mimořádné události jako je požár, riziko vloupání, loupež
- odhad možných budoucích rizik, informace o nich, popřípadě včasné zakročení proti nim
- možnost záložního napájení systémů a funkcí, nouzové LED osvětlení, zálohování PLC systému, komunikace a zabezpečení
- částečně redundantní řešení, každá z centrál distribuovaného systému „částečně dohlíží“ i na vedlejší podsystémy

Úspory

- regulace vytápění nebo klimatizace
- časové nebo časově omezené spínání
- regulace osvětlení (lze dosáhnout až 10 % úspor elektrické energie)
- závislé spínání (např. při soumraku, při nastavené teplotě …)
- blokování vybraných spotřebičů při vysokém tarifu elektroměru
- eliminace nechtěně zapnutých spotřebičů (např. při absenci pohybu vypne světlo)
- chráníte životní prostředí, šetříte peníze
- zúročení investice (stoupá hodnota domu při takovéto investici) [11]

Kapitola 3

3. REALIZACE FUNKCÍ A FUNKČNÍCH BLOKŮ

Knihovnu funkčních bloků jsem vypracoval ve vývojovém prostředí Mosaic v jazyce LD (Ladder Diagram = reléové schéma). Realizoval sem některé funkční bloky, které patří určitě pod pojem standardní, ale v textu jsem je neuvedl. Tyto funkční bloky jsem zahrnul do knihovny.

3.1 Osvětlení

V této kapitole uvedu několik funkčních bloků, které jsem zařadil do knihovny , které lze aplikovat na standardní inteligentní dům.

3.1.1 Paměťové funkce

Paměťové funkce jsou logické funkce, které na shodné hodnoty vstupů reagují různě, podle aktuálního stavu, v závislosti na předchozí posloupnosti hodnot vstupů nebo v závislosti na čase. Tato skupina logických funkcí se souhrnně nazývá jako sekvenční funkce.

V technice budov se nejčastěji setkáváme se sekvenčními funkcemi při realizaci časových závislostí, při obsluze mžikových kontaktů nebo krátkodobých senzorů, např. při rozlišení krátkého a dlouhého stisku tlačítka, při rozlišení jednoho stisku nebo opakovaného stisku tlačítka.

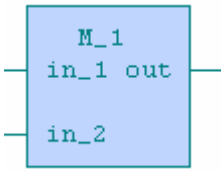
3.1.1.1. Tlačítka s nestálými polohami

Předpokladem pro úlohu bude použití tlačítka nebo skupiny tlačítek, z nichž každé má dvě polohy (horní a dolní), přičemž stálá poloha je střední (tzn. obě krajní polohy jsou nestálé - po uvolnění se tlačítko vrací do střední polohy, kdy není sepnutý žádný z kontaktů) Při stisku do jedné ze dvou krajních poloh spíná jeden nezávislý kontakt, který je přiveden do PLC. Díky další možnosti tlačítka jako je sepnutí kontaktu (pro obě polohy) buď na krátkou dobu (většinou mžikovou), anebo dlouhou dobu (přidržení tlačítka). Obsluha takovýchto tlačítek je sice logicky komplikovanější, ale poskytuje mnohem více možností. Jedním tlačítkem lze předat více informací (jako rozlišením jednoho či „dvojitého kliknutí“, krátkého nebo dlouhého stisku, případně jakákoli jejich kombinace.

Je to sice obrovská výhoda, ale na druhou stranu se nesmí nic přehánět. Důležitá je především názorná, přívětivá a spolehlivá obsluha, aby dům svou „přílišnou inteligencí“ nestresoval své obyvatele nebo jejich hosty.

Základní představou funkčního bloku je stisk tlačítka do jedné aktivní polohy (horní nebo dolní), kdy lze rozsvítit světlo, které má svítit do doby, než stiskneme totéž tlačítko do opačné aktivní polohy (dolní nebo horní).

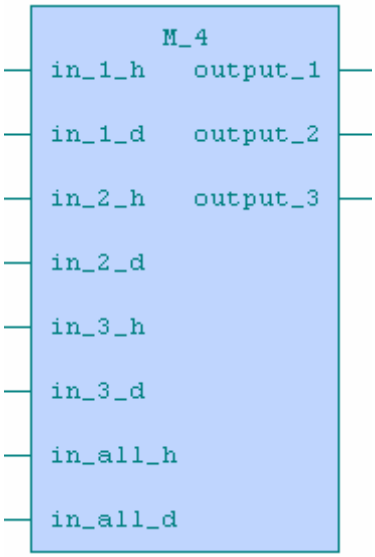
Řešit tento problém lze několika způsoby jako zpětnovazebním zapojení kontaktů nebo pomocí cívek Set a Reset. Asi nejjednodušším řešením je právě posledně zmiňované využití paměťového chování výstupních cívek typu Set a Reset. Může zde nastat jen problém tzv. spor, kdy je požadavek na zapnutí a vypnutí. Pak zde převáží poslední paměťová funkce v programu.



Obrázek 3.1: Jedno tlačítko s dvěma nestálými polohami

Jestliže chceme funkční blok použít pro reálný vypínač, musíme jej rozšířit na skupinu paměťových tlačítek. Nejvíce se v moderní elektroinstalaci využívají vypínače se čtyřmi mžikovými tlačítky. Můžeme například prvním třem přiřadit funkci rozsvěcení a zhasínání tří různých svítidel v místnosti (lampa u křesla, osvětlení za televizorem a intenzivní světlo z lustru) a poslední tlačítko slouží k ovládání celé skupiny svítidel naráz (stiskem do horní polohy se všechna svítidla aktivují, stiskem do dolní polohy se všechna vypnou).

Znovu využijeme výstupní cívky s pamětí Set a Reset. Podobným postupem lze realizovat například centrální zhasnutí a vypnutí spotřebičů při odchodu z domu nebo naopak centrální zapnutí orientačního osvětlení při návratu.



Obrázek 3.2: Čtyři tlačítka s dvěma nestálými polohami

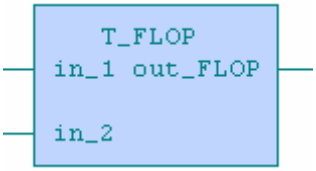
3.1.2 Schodiště a chodby

3.1.2.1. T_FLOP - Ovládání schodiště jedním tlačítkem (reverzibilní)

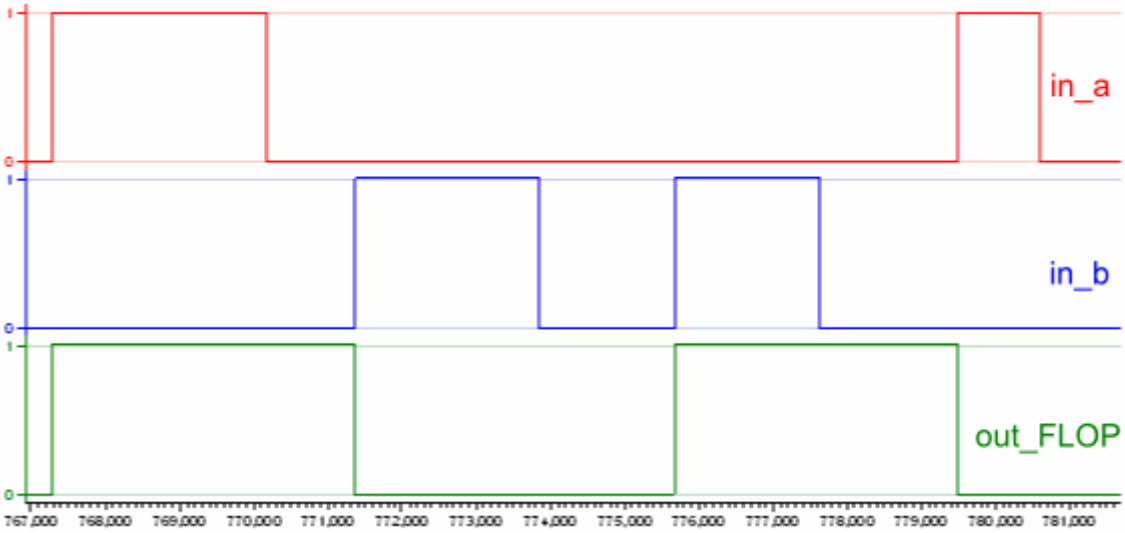
Ovládání osvětlení schodišť a chodeb je možné hned několika způsoby. Jednou z možností je ovládání pomocí tlačítek s nestálou polohou (více viz výše). Jiným požadavkem může být, aby světlo na chodbě či schodišti bylo ovládáno jedním kontaktem mžikových tlačítek tak, že na jednom konci chodby stiskem rozsvítím a na druhém konci zhasnu, třetím tlačítkem v mezipatře znovu rozsvítím. Dalším způsobem ovládání osvětlení tedy je, že každým stiskem se stav svítidla změní na opačný.

Realizace tohoto problému je velice vhodná pomocí náběžné hrany. To znamená, že ke změně stavu světa dojde v okamžiku stisku tlačítka (nikoliv při jeho uvolnění). Nejdříve je třeba vyhodnotit jednorázovou událost, právě když je stisknuto tlačítko (vyhodnocujeme náběžnou hranu vstupní proměnné). Jedná se o okamžik přechodu z hodnoty 0 na 1. Výsledkem je pak krátký impuls s hodnotou 1, který trvá právě dobu jednoho cyklu programu (obr.3.3). Jelikož jsem funkční blok realizoval v Mosaicu, mohl jsem využít blok R_TRIG

z knihovny STDLIB, který generuje impuls náběžné hrany. Jeho výstupem je krátký impuls jedničkové hodnoty po dobu jednoho cyklu programu.



Obrázek 3.3: Funkční blok T_FLOP

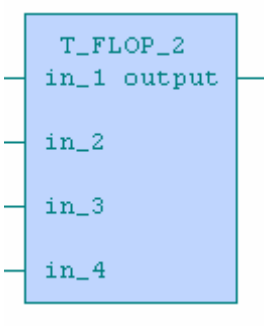


Obrázek 3.4: Průběh FB T_FLOP

3.1.2.2. Ovládání schodiště dvěma tlačítky

Jestliže chceme funkční blok použít pro reálný vypínač, musíme jej rozšířit. Proto budeme počítat se dvěma a více vstupy tak, aby se vždy v okamžiku stisku (s náběžnou hranou) kteréhokoliv z tlačítek změnil stav světla (obr.:3.5).

Řešit tento problém lze opět několika způsoby. Za nejvhodnější pokládám řešení s využitím R_TRIG_4 kaskádovitým způsobem. Zde se nejprve generuje impuls náběžné hrany pro každou vstupní proměnnou (stisk tlačítka) a teprve pak se tyto generované impulsy spojí logickým součtem XOR. Mohl by zde nastat jediný problém (použití OR místo XOR), a to když dva uživatelé stisknou dvě tlačítka současně během jednoho cyklu programu. To je ovšem v praxi téměř vyloučeno. Tomu jsem předešel použitím nejvhodnějšího zapojení.

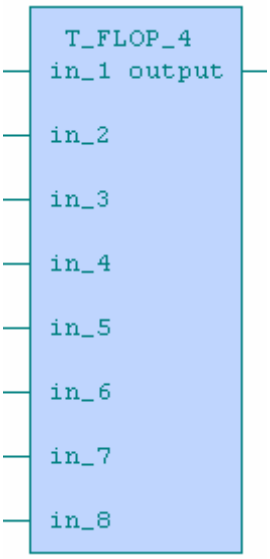


Obrázek 3.5: Funkční blok T_FLOP_2

3.1.2.3. Ovládání schodiště 4 a více tlačítky

Pro čtyři a více vstupů, kde musíme vždy v okamžiku stisku (s náběžnou hranou) kteréhokoliv z tlačítek změnil stav světla.

Lze řešení s dvěma tlačítky implementovat i pro více tlačítek. Kde jak je zmíněno implementace je stejná jen se zabalí do bloku, který se opakuje (tzv. kaskáda).

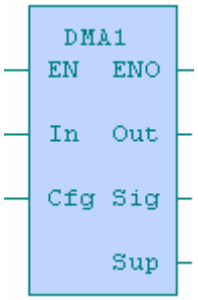


Obrázek 3.6: Funkční blok T_FLOP_4

3.1.2.4. Ovládání osvětlení – stmívání

Realizace toho funkčního bloku je velmi výhodná z hlediska šetrnosti obyvatele domu. Jestliže se obyvatel ráno probudí, není mu určitě příjemné jak září obyčejné světlo po sepnutí vypínače. Proto řešení takového funkčního bloku je velmi potřebné. Dále lze tento blok realizovat při romantických chvílích u večeře či televize nebo taktéž v promítacích kino sále na začátku a konci promítání.

Funkční blok stmívače DMA1 v knihovně BuildingLib je velmi užitečnou variantou. Slouží k plynulému ovládání jasu žárovek s možností postupného náběhu/doběhu po zapnutí/vypnutí stmívače. Při krátkém stisku (<0.5s) nabíhá na požadovanou úroveň jasu, dalším krátkým stiskem výstup vypíná. Dlouhým stiskem tlačítka (>0.5s) dochází k plynulé regulaci jasu v rozsahu hodnot *MinOut* <-> *MaxOut.* Po dosažení krajní hodnoty je regulace jasu zastavena. Po uvolnění tlačítka je intenzita jasu zapamatována a další krátké stisky tlačítka vypínají/zapínají výstup na tuto intenzitu. Rychlost náběhu výstupu na požadovaný jas při zapnutí je dána parametrem *Tup*, rychlost sestupu při vypnutí je dána parametrem *Tdw.* [7]



Obrázek 3.7: Vzhled FB DMA1

3.1.2.5. Ovládání osvětlení s časovou funkcí

Návrh knihovny by měl odpovídat praktickému využití funkcí, kde se často setkáváme s případy, kdy je třeba řídit objekt podle časových závislostí, vygenerovat impulsy potřebné délky nebo potřebnou časovou posloupnost signálů. Bývá potřebné měřit čas, dobu trvání nebo měřit délky impulzů či rozpoznat složitější události.

3.1.2.6. PIR čidlo pohybu

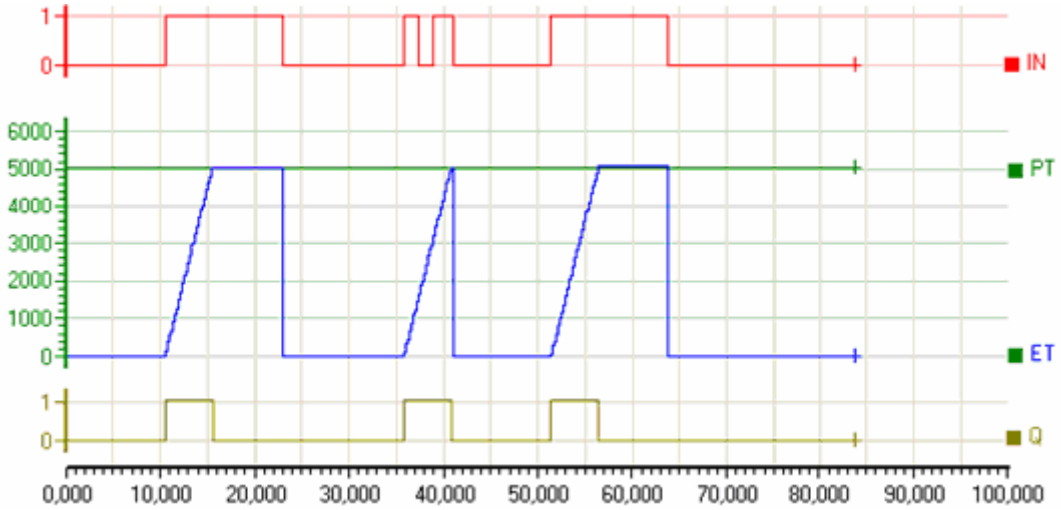
Ovládání osvětlení nemusí vždy probíhat jen pomocí tlačítek, ale lze využít také infračerveného senzoru pohybu PIR. Jedním z důvodů je, že osobu zaregistruje jen, když prochází „zorným polem“ snímače. Kdyby se zde osoba zastavila, přestane ji po určité době

registrovat, dokud se znovu nepohne. Jak bylo popsáno v kapitole 3.1.1 pro senzor PIR použijeme znovu paměťové funkce. Zjednodušeně lze říci, že PIR detektory jsou schopny zachytit pohyb těles, která mají jinou teplotu, než teplotu okolí. Pozor není to senzor přítomnosti! Jejich funkce je založena na zachycení změn vyzařování v infračerveném pásmu kmitočtového spektra elektromagnetického vlnění.

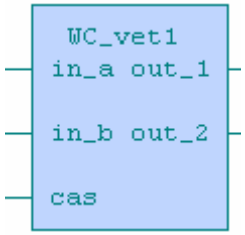
3.1.2.7. Vyvětrání místnosti

Jednoznačným požadavkem je, aby se po stisku tlačítka (nebo signálem z čidla PIR) spustil ventilátor a běžel nastavenou dobu (nastaví si zákazník sám, přednastaveny budou 2 minuty). Tlačítko může současně sloužit i k rozsvícení nebo ke zhasnutí svítidla - podle toho, přejeme-li si aby ventilátor běžel současně se světlem (patrně po dobu naší přítomnosti) nebo až od okamžiku zhasnutí (po našem odchodu) z místnosti. Nebo ještě lépe, aby po zhasnutí ventilátor běžel nastavenou dobu (přednastaveno 15 sekund) ještě sám.

Řešením tohoto problému je funkčního bloku SR a TP impulsního časovače, generátoru impulzů zadané délky uložen v knihovně STDLIB. Vytvořený funkční blok WC_vent1 je v knihovně IB_Lib obrázek 3.9.



Obrázek 3.8: Průběh časovače TP

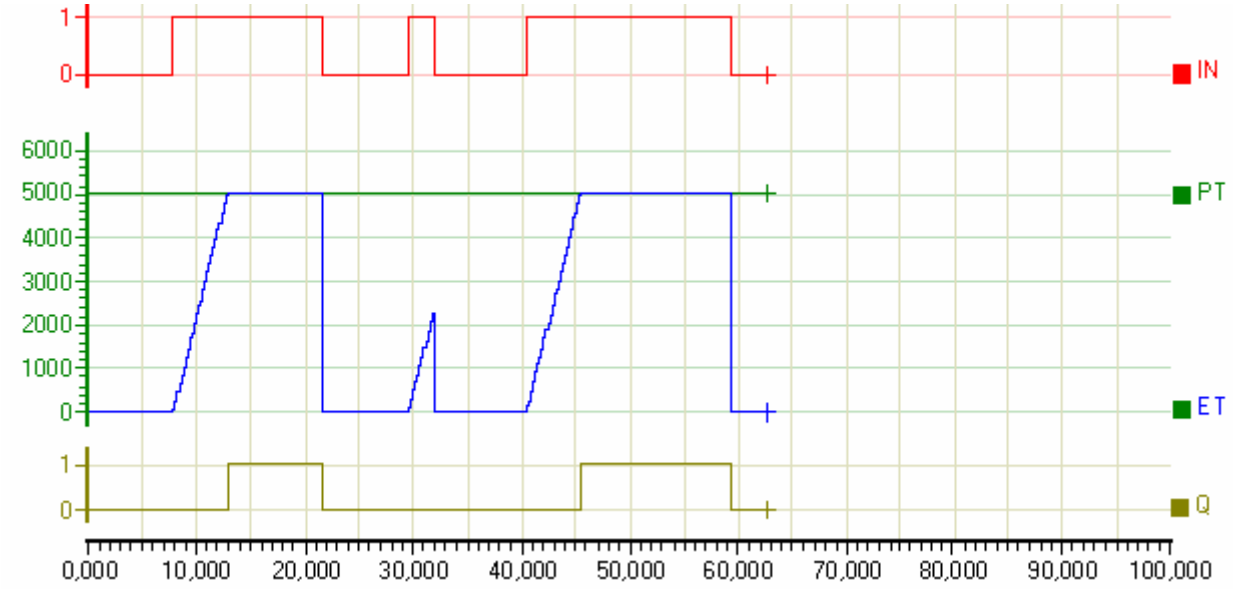


Obrázek 3.9: Vzhled FB WC_vent1

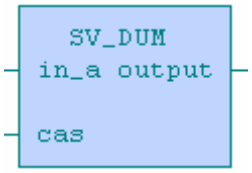
3.1.2.8. Osvětlení před domem

Světlo před domem je většinou ovládáno infračerveným senzorem PIR. Požadavek je, aby se světlo rozsvítilo až 30 sekund poté, co byl senzor aktivován (zaregistroval přítomnost osoby). Po skončení aktivace senzoru zhasne světlo okamžitě. Světlo tak nereaguje na chodce, kteří jen procházejí kolem vchodu ani na psy a menší živočichy. Rozsvítí se až při dlouhodobé aktivaci senzoru PIR.

Řešení odpovídá činnosti standardního funkčního bloku TON (Timer On Delay z knihovny STDLIB), tedy časovače, který zpožďuje náběžnou hranu řídicí proměnné - a tím současně odfiltruje vstupní impulzy, kratší než je doba předvolby.



Obrázek 3.10: Průběh časovače TON



Obrázek 3.11: Vzhled FB SV_DUM

3.2 Zabezpečovací technika

Zabezpečovací technika řeší většinou jiný problém, než ovládání osvětlení. Nejprve si uvedeme příklady, které se v praxi využívají, zejména typy senzorů. Nejprve bych uvedl nejdůležitější senzory rozbití skla, magnetické dveřní kontakty, pyrometrické detektory, sirény, detektory kouře a hořlavých plynů.

Alarm všeobecně bude spuštěn po vyhodnocení nadbytečných kontaktů, to znamená sepnutí několika kontaktů po sobě nebo naráz. Díky této realizaci se snižuje riziko neohlášených alarmů při selhání jedno ze senzorů nebo vznik falešných poplachů. Navíc lze také nastavit určité stupně naléhavosti hlášení alarmu či poplachu.

Samotná realizace bude tvořena například takzvanými prahovými funkcemi. Jako typický případ lze považovat inteligentní budovy, kde jsou čidlo pohybu, senzor rozbití skla a magnetický dveřní kontakt.

<u>Nastavení alarmu definujeme na 3 úrovně:</u>

1. senzor rozbití skla dává signál – muže to být zloděj, ale i míč kopnutý od sousedova kluka, proto nevolat bezpečností agenturu.
2. senzor rozbití skla a čidlo pohybu dávají signál – už je něco v nepořádku – proto pošli majiteli SMS
3. senzor rozbití skla, čidlo pohybu a dveřní kontakt dávají signál – dej signál bezpečnostní agentuře, nejspíše vykrádají dům

Zde je názorně vidět, že každý ze signálu může mít jinou váhu. A toto platí i při využití majority, obecně u symetrických funkcí.Navrhl jsem několik funkčních blok, zde uvedu dva, které jsou pro praktické využití nejvhodnější a zbytek lze nalézt v knihovně.

3.2.1 Majorita

3.2.1.1. Majorita ze 3

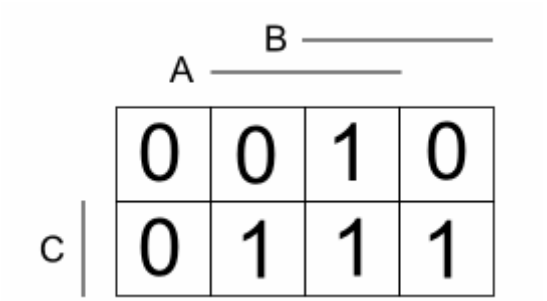
Funkce majority (většiny) se může definovat i pro jiný lichý počet operandů než tři a je pravdivá pro nadpoloviční počet pravdivých operandů. Majorita ze tří je funkce, která je pravdivá pokud nadpoloviční většina (dva nebo tři) z jejich operandů je pravdivých. Je obdobou hlasování tří rovnoprávných voličů.

Proto se jí někdy říká, *hlasovací funkce*. Je označena zkratkou *maj3* a je definována logickým výrazem: $maj3(a,b,c) = (a\ AND\ b)\ OR\ (a\ AND\ c)\ OR\ (b\ AND\ c)$

Nebo jí lze vyjádřit tabulkou (Tab.3.1) nebo karnaughovou mapou (Obr. 3.12)

Vstup A	Vstup B	Vstup C	maj3
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

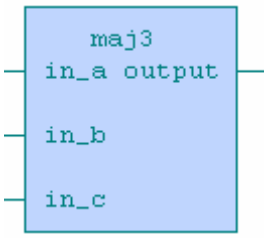
Tabulka 3.1: Popis funkce třívstupového majoritního členu



Obrázek 3.12: Karnaughova mapa majority ze 3

Její negace se někdy označuje jako funkce *minority ze tří*, která je pravdivá, pokud je pravdivá menšina (žádný nebo jeden) z jejich operandů. Platí pro ni logický výraz:

$$\overline{maj3(a,b,c)} = maj3(\overline{a},\overline{b},\overline{c}) = (\overline{a}\ AND\ \overline{b})\ OR\ (\overline{a}\ AND\ \overline{c})\ OR\ (\overline{b}\ AND\ \overline{c})$$



Obrázek 3.13: FB majorita ze 3

3.2.1.2. Majorita z 5

Majorita z pěti je funkce, která je pravdivá pokud nadpoloviční většina (tři nebo čtyři nebo pět) z jejich operandů je pravdivých. Je označena zkratkou *maj5* a je definována logickým výrazem:

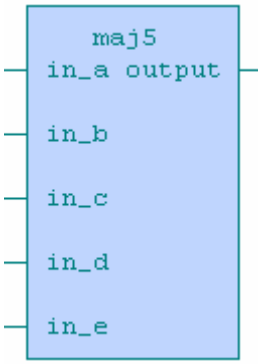
$$
\begin{aligned}
maj5(a,b,c,d,e) = {} & (a\ AND\ b\ AND\ c)\ OR\ (a\ AND\ b\ AND\ d)\ OR\ (a\ AND\ b\ AND\ e) \\
& OR\ (a\ AND\ c\ AND\ d)\ OR\ (a\ AND\ c\ AND\ e)\ OR\ (a\ AND\ d\ AND\ e) \\
& OR\ (b\ AND\ c\ AND\ d)\ OR\ (b\ AND\ c\ AND\ e)\ OR\ (b\ AND\ d\ AND\ e) \\
& OR\ (c\ AND\ d\ AND\ e)
\end{aligned}
$$

Nebo jí lze vyjádřit karnaughovou mapou (Tab. 3.2)

			A					
	0	1	1	1	0	1	0	0
	1	1	1	1	1	1	1	0
E	0	1	1	1	0	1	0	0
	0	0	1	0	0	0	0	0
		C		D			C	

B

Tabulka 3.2: Popis funkce pětivstupového majoritního členu



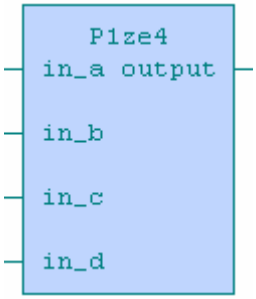
Obrázek 3.14: FB majorita z 5

3.2.2 Prahové funkce

Jsou to logické funkce, které jsou typu *alespoň k ze tří*, tedy *alespoň 1 ze tří* (shodná s logickým součtem OR), *alespoň 2 ze tří* (shodná s *maj3*) a *alespoň 3 ze tří* (shodná s logickým součinem AND). Souhrnně tyto funkce označujeme jako prahové funkce a zkrácené označení pro jednotlivé bloky je *p1ze3*, *p2ze3*, *p3ze3*.

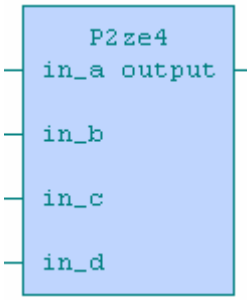
Obecně *pk_n*, kde první číslo představuje hodnotu prahu a druhé je počtem operandů. Lze je definovat pro libovolný počet operandů a pro libovolné hodnoty prahu. Obecně platí, že jsou pravdivé, pokud počet pravdivých operandů je alespoň roven (roven nebo větší) prahu (prvnímu číslu z názvu funkce). Zvláštním případem prahových funkcí je i logický součet OR (pro práh = 1, např. *p1_3*, obecně *p1_n*) a logický součin AND (pro práh rovný počtu operandů, např. *p3_3*, obecně *pn_n*). Pro lichý počet operandů sem patří i majoritní funkce např. *maj3* (práh = 2, *p2_3*), *maj5* (práh = 3, *p3_5*).

Prahová funkce 1 ze 4 (P1ze4)



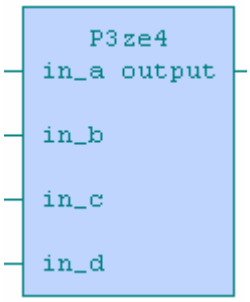
Obrázek 3.15: a) Prahová funkce 1 ze 4

Prahová funkce 2 ze 4 (P2ze4)



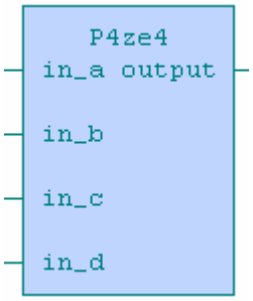
b) Prahová funkce 2 ze 4

Prahová funkce 3 ze 4 (P3ze4)



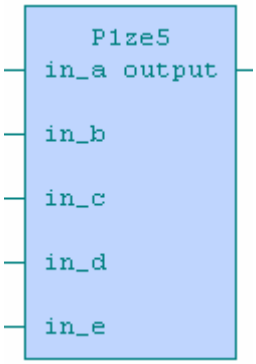
Obrázek 3.16: a) Prahová funkce 3 ze 4

Prahová funkce 4 ze 4 (P4ze4)



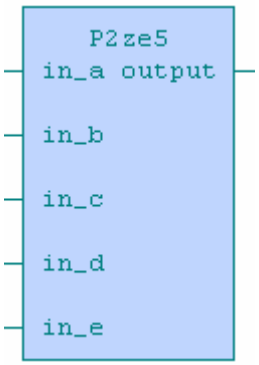
b) Prahová funkce 4 ze 4

Prahová funkce 1 z 5 (P1ze5)



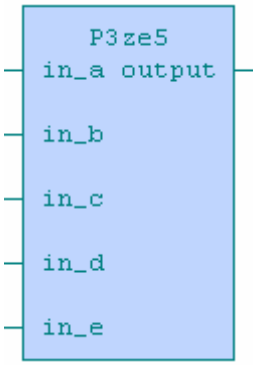
Obrázek 3.17: a) Prahová funkce 1 z 5

Prahová funkce 2 z 5 (P2ze5)



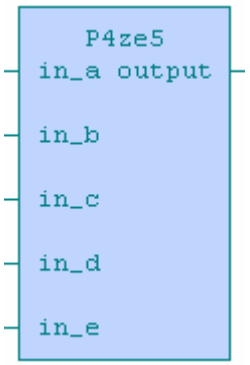
b) Prahová funkce 2 z 5

Prahová funkce 3 z 5 (P3ze5)



Obrázek 3.18: a) Prahová funkce 3 z 5

Prahová funkce 4 z 5 (P4ze5)



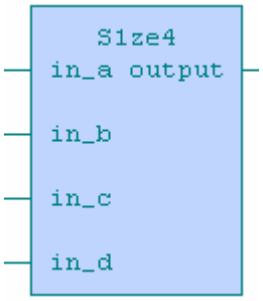
b) Prahová funkce 4 z 5

3.2.3 Elementární symetrické funkce

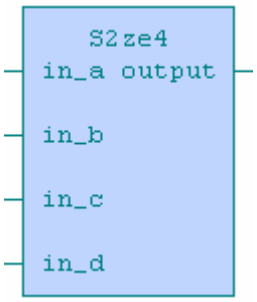
Jsou to logické funkce, které jsou slovně popsány *právě k ze tří*, konkrétně *právě 1 ze 3*, *právě 2 ze 3* a *právě 3 ze 3*.

Obecně pak *právě k z n*. Souhrnně tuto skupinu logických funkcí budeme nazývat jako elementární symetrické funkce. Stručně je budeme označovat jako *S1ze3*, S*2ze3*, S*3ze3*, obecně S k ze n. Jsou pravdivé pokud počet pravdivých operandů je právě roven charakteristickému číslu - prvnímu číslu z označení funkce.

Symetrická funkce 1 ze 4 (S1ze4)

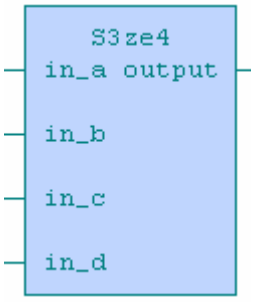


Symetrická funkce 2 ze 4 (S2ze4)

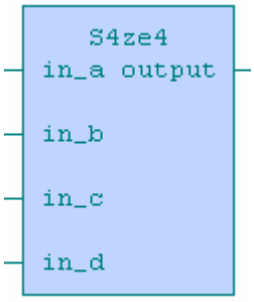


Obrázek 3.19: a) Symetrická funkce 1 ze 4 b) Symetrická funkce 2 ze 4

Symetrická funkce 3 ze 4 (S3ze4)

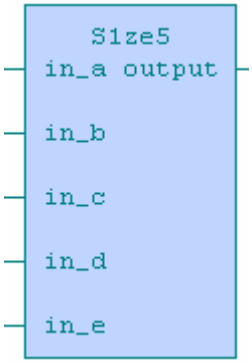


Symetrická funkce 4 ze 4 (S4ze4)

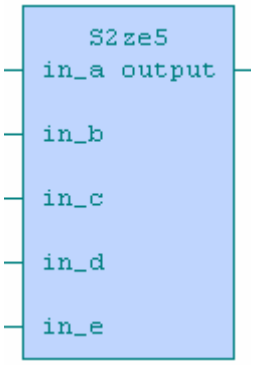


Obrázek 3.20: a) Symetrická funkce 3 ze 4 b) Symetrická funkce 4 ze 4

Symetrická funkce 1 z 5 (S1ze5)

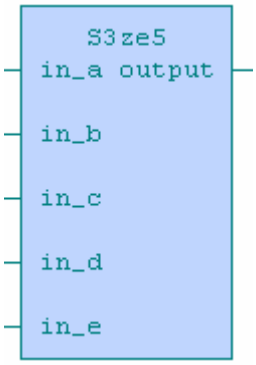


Symetrická funkce 2 z 5 (S2ze5)

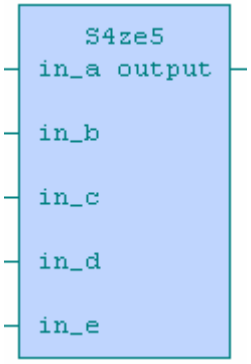


Obrázek 3.21: a) Symetrická funkce 1 z 5 b) Symetrická funkce 2 z 5

Symetrická funkce 3 z 5 (S3ze5)

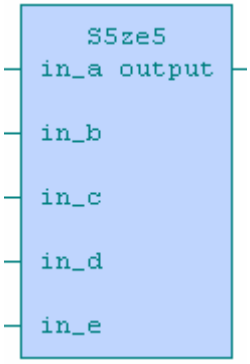


Symetrická funkce 4 z 5 (S4ze5)



Obrázek 3.22: a) Symetrická funkce 3 z 5 b) Symetrická funkce 4 z 5

Symetrická funkce 5 z 5 (S5ze5)



Obrázek 3.23: Symetrická funkce 5 z 5

3.3 Ovládání mechanismů

V budovách je mnoho ovládacích mechanismů jako je automatické otevíraní oken, pohyb markýz či rolet, otevírání a zavíraní garážových a vjezdových vrat lze je definovat dvěma stabilními stavy. Jedna poloha, kdy systém nebo jednotlivá část je v poloze otevřeno a druhá poloha, kdy je ve stavu zavřeno. Toto jsou dvě základní věci, na které můžeme nabalovat další. Jako je například mikro ventilace u oken, nebo náklon listů rolet v několika stupních dle požadavků zákazníka pro jeho maximální komfort.

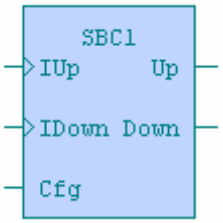
Narážíme zde na takzvané mezní polohy, kdy je potřeba kontroly chyb. Například zda-li se nám vstupní brána zavřela. To lze vyřešit jakoukoli časovou funkcí, kdy pro jednotlivé systémy (přístroje) definujeme čas, za který musí být brána v koncové poloze (zavřeno či otevřeno). Pokud tak nenastane, muže nám systém nahlásit chybu (brána není v požadované poloze).

3.3.1 Ovladač roletových žaluzií SBC1

Funkční blok slouží k ovládání pohonu roletových žaluzií. Ovládání je realizováno pomocí dvou tlačítek. Blok zajišťuje časovou kontrolu doby běhu pohonu a prodlevu při reverzaci chodu pohonu (ochrana pohonu z důvodů elektrických i mechanických).

Po stisku tlačítka *IUp* je aktivován výstup pro pohyb žaluzií směrem nahoru *Up*. Dalším stiskem kteréhokoliv tlačítka je výstup *Up* deaktivován. Po stisku tlačítka *IDown* je aktivován výstup pro pohyb žaluzií směrem dolů *Down*. Dalším stiskem kteréhokoliv tlačítka je výstup *Down* deaktivován.

Při aktivaci výstupů na dobu delší než *Tac* budou výstupy automaticky deaktivovány. Při reverzaci chodu pohonu je mezi aktivace jednotlivých výstupů vždy vložena časová prodleva *Twt*.



Obrázek 3.24: Vzhled FB SBC1

Tento funkční blok lze nalézt v knihovně BuildingLib pod názvem SBC1. Mě jako uživateli na tomto funkčním bloku chybělo několik věcí, které lze jistě široce využít při dalších konfiguracích pohodlného bydlení.

Zde jen lehce naznačím, co bych ještě ocenil a v kapitole 4.1.3 uvedu podrobnosti a řešení. Mezi vstupy bych upozornit potřebnost chybových výstupů, čas kdy mechanismus nedosáhl v limitu žádanou polohu, ať nahoře či dole a také ztrátu polohy, která mohla vzniknout vloupáním nebo poruchou senzoru.

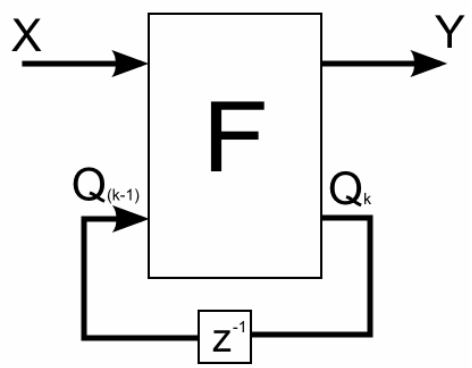
Kapitola 4

4. NÁVRH FUNKCÍ A FUNKČNÍCH BLOKŮ

4.1 Složitější sekvenční funkce

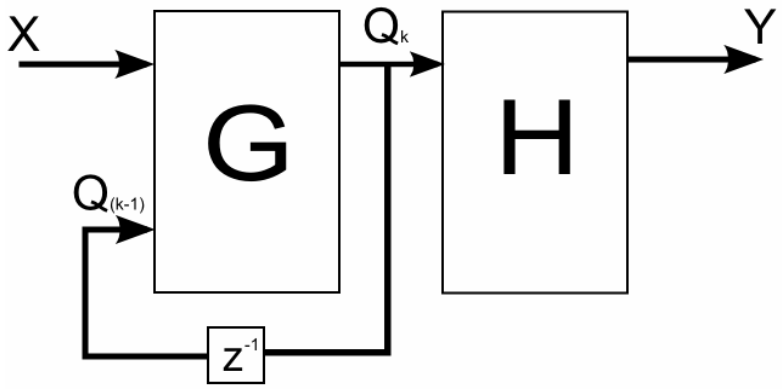
Funkční závislosti pro majoritní, prahové a elementární symetrické funkce mají ryze kombinační charakter a mohou tedy být realizovány jako POU typu funkce. V technice budov je ale mnoho logických funkčních závislostí, které mají výrazně sekvenční charakter, mnohdy poměrně komplikovaný. Pro ně je třeba zvolit POU typu funkčních bloků. Pro zadání a popis jejich chování je výhodné použít symboliku konečných automatů nebo grafického prostředku SFC (Sequential Function Chart) normy IEC 61131-3. V teorii konečných automatů se rozlišují dva typy (modely) konečných automatů.

Automat **Mealyho typu** (Obr.:4.1) zpracovává aktuální vstupní vektor (soubor aktuálních vstupů) a stav (soubor vnitřních proměnných) z minulého výpočetního cyklu a generuje z nich hodnoty výstupů (výstupního vektoru) a nového stavu. realizuje tak dvojici funkcí: přechodovou ($Q_k = G(X, Q_{k-1})$) a výstupní funkci ($Y = H(X, Q_{k-1})$).



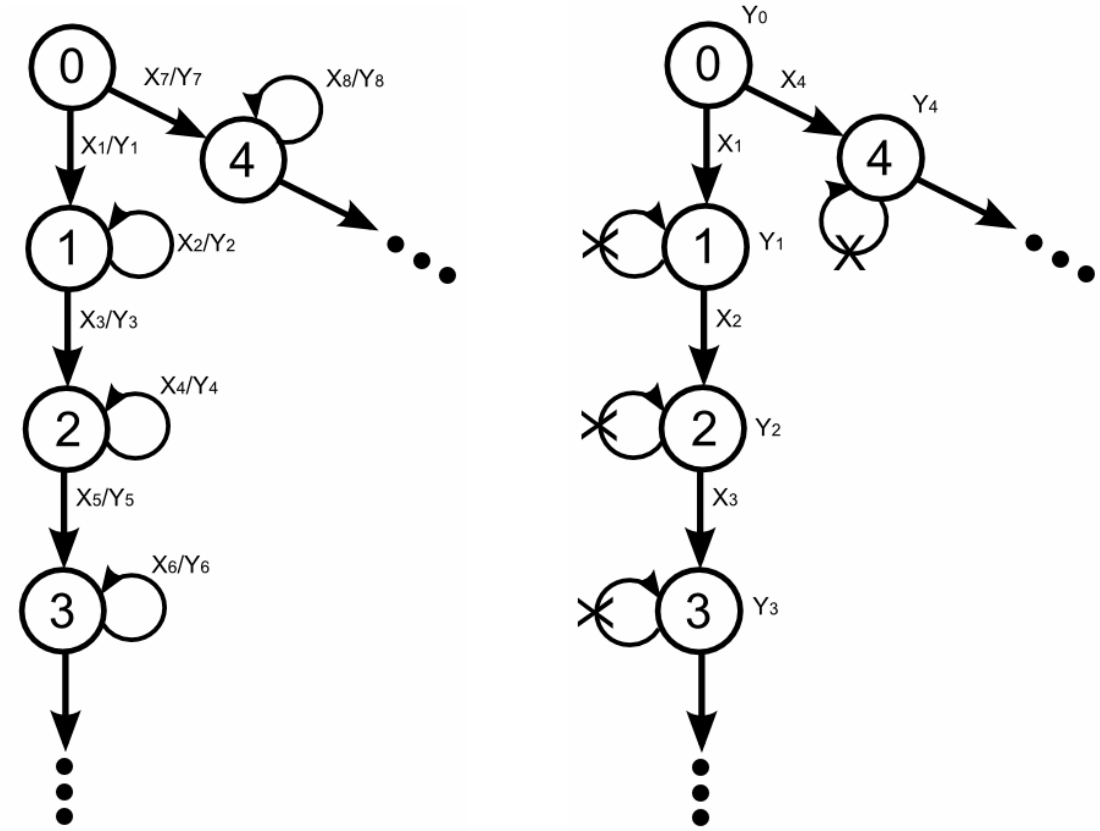
Obrázek 4.1: Automat typu Mealy

Automat **Mooreova typu** (Obr.: 4.2) má oddělenou přechodovou funkci ($Q_k = G(X, Q_{k-1})$) a výstupní funkci ($Y = H(Q_{k-1})$), případně ($Y = H(Q_k)$). Argumentem výstupní funkce je jen stav (současný, případně minulý - rozdíl je jen o zpoždění jednoto výpočetního cyklu).



Obrázek 4.2: Automat typu Moore

Vhodným grafickým prostředkem pro popis sekvenčního chování je přechodový diagram - na Obr.: 4.3 pro automat Mealyho typu nebo na Obr.: 4.4 pro automat Mooreova typu.



Obrázek 4.3: a) Vývojový diagram (Mealy) b) Vývojový diagram (Moore)

Oba typy automatů mají stejnou „výpočetní mohutnost“ a jsou navzájem převoditelné. K popisu stejné úlohy postačuje Mealyho automat s menším počtem stavů (je úspornější), než by potřeboval odpovídající automat Mooreova typu.

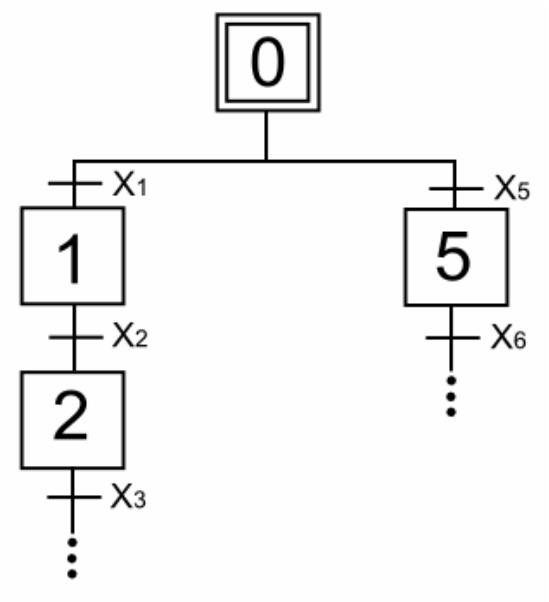
Automat Mealyho typu lépe popisuje dynamické systémy, protože i změnu stavu může být doprovázena změnou výstupů (viz přechodový diagram na Obr.: 4.3a). Naproti tomu

automat Mooreova typu popisuje spíše „stacionární“ chování, protože hodnota výstupu je zde jen kombinační funkcí stavu a po celou dobu setrvání v dosaženém stavu (do dalšího přechodu) se nemění ani hodnoty výstupů. V přechodovém diagramu se proto obvykle neuvádějí „krátké smyčky“ pro setrvání ve stavu. Změna stavu (přechod) je zde „bezvýznamnou mžikovou“ událostí, který se neprojeví na výstupech.

Model automatu Mooreova typu je proto vhodný pro popis chování většiny technologických procesů. Zvláštním případem automatu Mooreova tylu je krokový řadič, jehož přechodový diagram má tvar lineárního řetězce stavů nebo jednoduché uzavřené smyčky - neobsahuje žádné větvení.

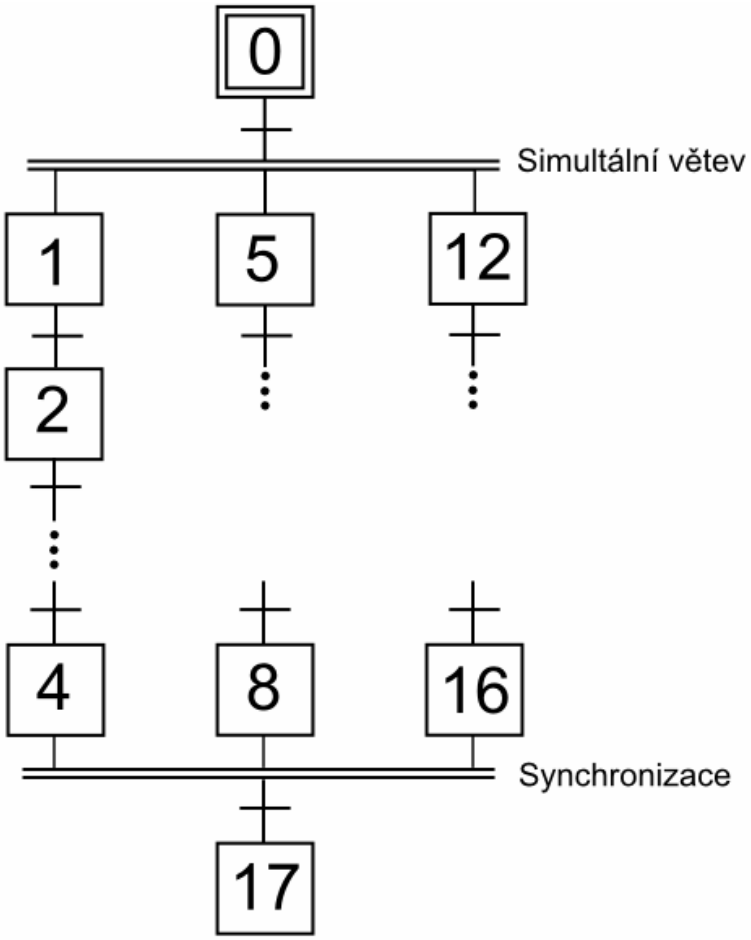
Norma IEC 61131-3 má k dispozici grafický prostředek SFC, který je obdobou přechodového diagramu automatu Mooreova typu. Stavům (zde se mluví o krocích) odpovídají obdélníky (počáteční stav má dvojité orámování), přechody se značí vodorovnou čárkou, která přetíná hranu mezi dvěma kroky. Podobně, jako u konečných automatů (přesněji u deterministických konečných automatů) může být v grafu SFC v daném okamžiku aktivní jediný se stavů.

Při větvení grafu „výlučným větvením“ (například Obr.: 4.4) je proto nutné zajistit, aby podmínka přechodu byla splněna jen pro jedinou větev - opačný případ je chyba programátora a při implementaci je třeba ji zvlášť ošetřit (například označením priority podmínky přechodu nebo definovaným pořadím vyčíslování podmínek, třeba zleva doprava).



Obrázek 4.4: Ukázka jazyka SFC

Prostředek SFC ale nabízí další možnost, kterou se graf přibližuje vlastnostem jedné třídy Petriho sítí - je jím „simultánní větvení“ (Obr.: 4.5) a synchronizace. Při splnění podmínky přechodu na vstupu simultánního větvení se současně aktivují všechny následující stavy, které se dále vyvíjejí podle dalších úseků grafu. Protějškem simultánního větvení je synchronizace. Přechod, který za ní následuje, se uskuteční, pokud jsou aktivní všechny stavy na vstupu dvojité „sběrnice“ a současně je splněna podmínka přechodu - s přechodem se počká na ukončení poslední větve grafu.



Obrázek 4.5: Ukázka simultánního větvení jazyka SFC

Ve své práci používám právě tuto symboliku pro popis sekvenčního chování funkčních bloků, a to i přesto, že vývojový systém Mosaic zatím nemá k dispozici implementaci SFC . Tento graf je totiž velmi názorný i pro laiky a je systematický. Převod grafu do formy programu PLC je poměrně nenáročnou rutinní prací.

4.1.1 Jazyk SFC (Grafcet)

Grafcet, ve francouzštině "Graphe Fonctionnel de Connexion Etapes Transitions", je grafický návrhový nástroj pro řídicí systémy. Popisuje pouze realizovanou funkci, neboli automat v matematickém smyslu, nezávisle na technologii a konečné realizaci.

Grafcet vychází z Petriho sítí. Grafcet je vhodný pro návrh algoritmů řízení PLC (programovatelných logických automatů).

Grafcet byl navržen v letech 1975-1977 francouzskou organizací AFCET "Association Francoise pour la Cybérnetique Economique et Technique" a v roce 1987 se stal mezinárodním standardem (International Electrotechnical Commission 848 - Preparation of function charts for control systems). [12]

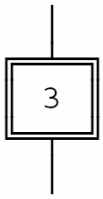
Jazyk Grafcet a SFC jsou velmi podobné, víceméně totožné. Jazyk SFC se vyvinul později, kdy převzal další z kompromisů.

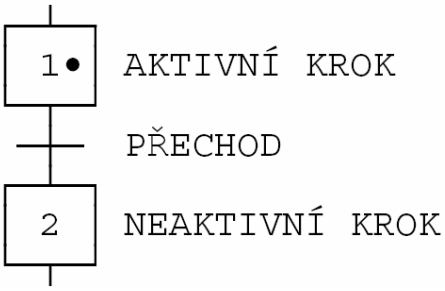
4.1.2 Základní koncepce

Grafcet obsahuje dva základní prvky : krok a přechod. Aktuální stav systému je v Grafcetu reprezentován aktivitou/neaktivitou všech kroků (neboli dosaženým značením). Každý krok (znázorněn čtverečkem s pořadovým číslem) se může vyskytovat pouze ve dvou základních stavech – buď může být aktivní anebo neaktivní . Aktivita kroku je v Grafcetu znázorněna značkou (tečka).

Kroky, které jsou aktivní na začátku řídicího procesu (popisují počáteční stav systému) jsou vyjádřeny symbolem **počáteční krok** (značen dvojitým čtverečkem - Obr.: 4.6a).

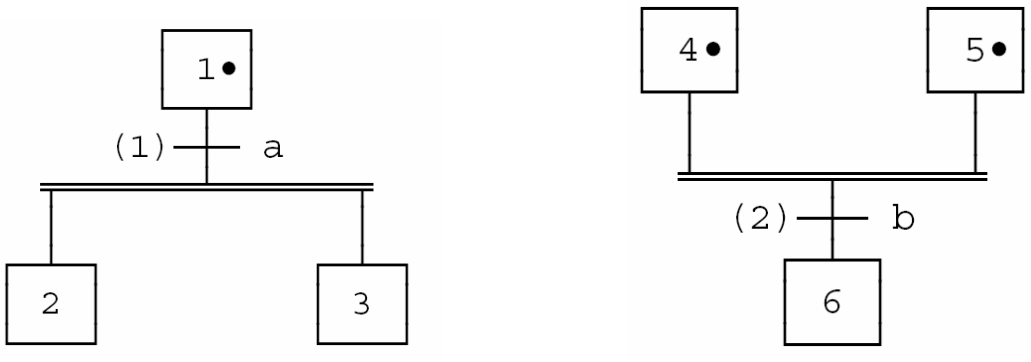
Přechod je značen krátkou vodorovnou úsečkou (Obr.: 4.6b). Přechod spolu s krokem tvoří dvojici vrcholů bipartitního orientovaného grafu (hrana jde vždy z kroku do přechodu nebo z přechodu do kroku).





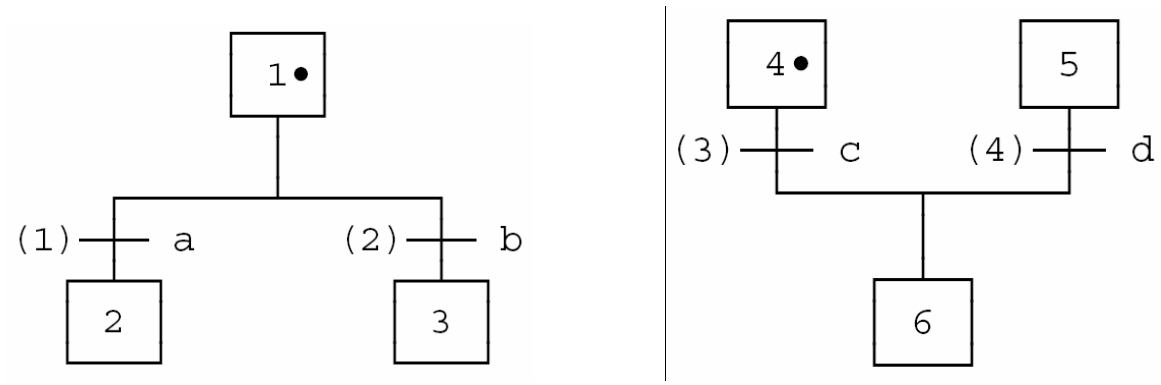
Obrázek 4.6: a) Počáteční krok b) Přechod

Každý přechod může mít několik kroků vstupních a několik kroků výstupních. Takové přechody nazýváme **konvergentní a divergentní AND** přechody. Umožňují znázornit počátek a konec dvou a více paralelních sekvencí. Graficky paralelismus značíme dvojitou čarou za přechodem, synchronizaci pak dvojitou čarou před přechodem. Následující obrázek představuje divergentní AND před a po přeskoku přechodu (Obr.: 4.7a) a konvergentní AND před a po přeskoku přechodu (Obr.: 4.7b)



Obrázek 4.7: a) Divergentní AND b) Konvergentní AND

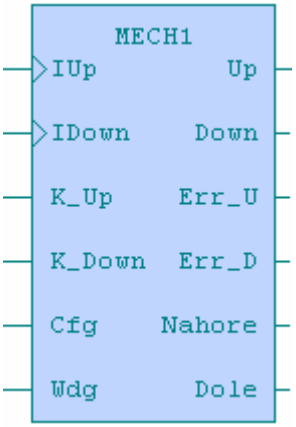
Každý krok může mít několik vstupních a několik výstupních přechodů. V těchto případech jsou přechody **nazývány konvergentní a divergentní OR** přechody. Divergentní OR přechod znázorňuje výběr mezi dvěma popřípadě více následujícími větvemi. Konvergentní OR přechod znázorňuje spojení dvou popřípadě více větví do jedné. [14] [13]



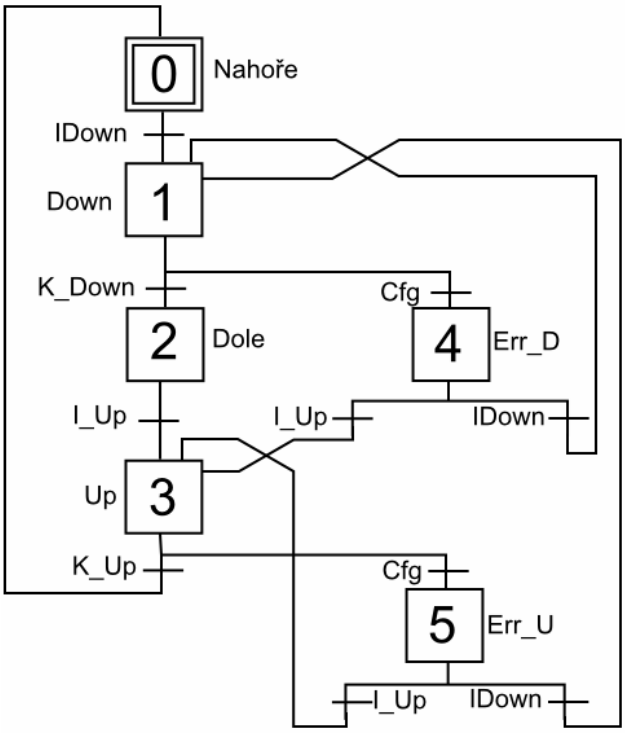
Obrázek 4.8: a) Divergentní OR b) Konvergentní OR

4.1.3 Funkce ovládání mechanismů

Funkční blok slouží k ovládání mechanismů (např.: pohonu roletových žaluzií). Ovládání je realizováno pomocí dvou tlačítek. Po stisku tlačítka *IUp* je aktivován výstup pro pohyb žaluzií směrem nahoru *Up* a přitom je spuštěn časovač TON, který je nastaven na optimální dobu pohybu rolety směrem nahoru. Po stisku tlačítka *IDown* je aktivován výstup pro pohyb žaluzií směrem dolů *Down* a přitom je spuštěn časovač TON, který je nastaven na optimální dobu pohybu rolety směrem dolů. Poté nastává kontrola *Cfg* (podmínky přerušení) nebo signálu ze senzoru *K_Down*. To samé nastává i u směru nahoru. Uživatel má možnost podmínky Cfg nastavit jak se mu zlíbí. Vstup *Wdg* je optimální čas pohybu rolety, který lze libovolně přednastavit.



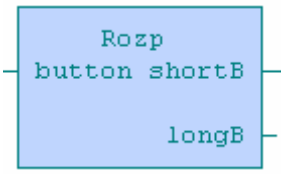
Obrázek 4.9: Vzhled bloku mechanického ovládání



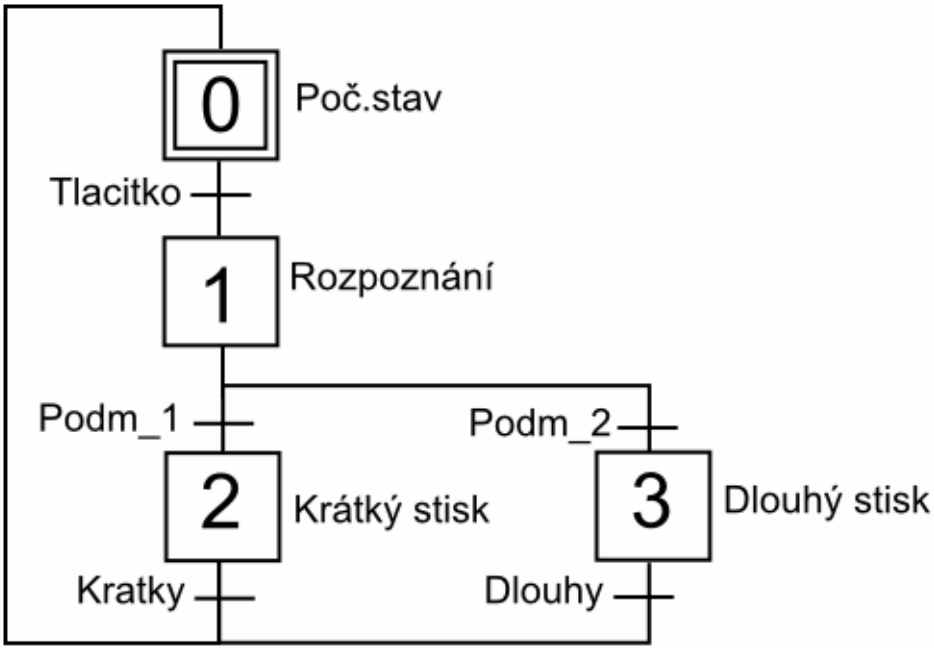
Obrázek 4.10: Diagram mechanického ovládání SFC

4.1.4 Funkce rozpoznání dlouhého a krátkého stisku

Funkční blok slouží k rozpoznání dlouhého a krátkého stisku tlačítka. Po stisku tlačítka *button* je aktivován čítač TON. Poté je splněna Podm_1, kdy čítač načítá jen velmi krátkou dobu (<5s) nebo Podm_2, kdy čítač načítá dobu větší než pět vteřin. Jestliže je splněna Podm_1 program vyšle signál výstupu *shortB*. Pokud je splněna podmínka Podm_2 program vyšle signál výstupu *longB*. Tento blok najde mnoho uplatnění, ať už v oblasti osvětlení, ovládání mechanických části nebo zabezpečovacího systému.



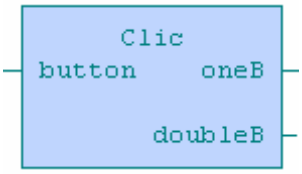
Obrázek 4.11: Vzhled bloku dlouhý a krátký stisk



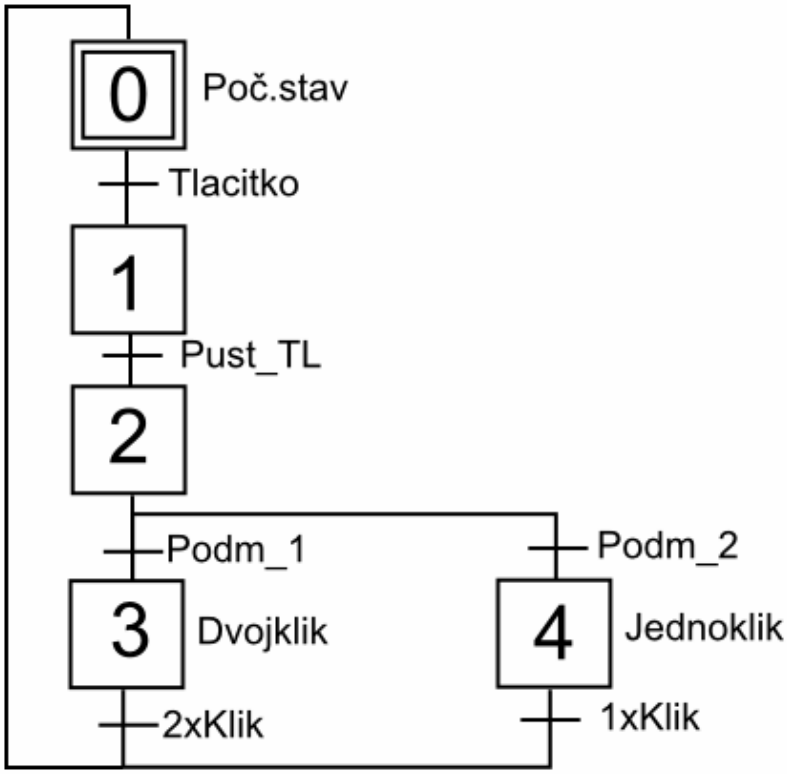
Obrázek 4.12: Diagram rozpoznání tlačítka SFC

4.1.5 Funkce rozpoznání dvojkliku

Funkční blok slouží k rozpoznání jednokliku a dvojkliku tlačítka. Po stisku tlačítka *button* je aktivován čítač TON. Poté co je splněna Podm_1, kdy čítač načítá jen velmi krátkou dobu (<5s) nebo Podm_2, kdy čítač načítá dobu větší než pět vteřin . Jestliže je splněna Podm_1 program vyšle signál výstupu *shortB*. Pokud je splněna podmínka Podm_2 program vyšle signál výstupu *longB*. Tento blok najde mnoho uplatnění, ať už v oblasti osvětlení, ovládání mechanických části nebo zabezpečovacího systému.



Obrázek 4.13: Vzhled bloku rozpoznání dvojkliku



Obrázek 4.14: Diagram rozpoznání dvojkliku SFC

4.1.6 Scény

Nebo taktéž režimy domy, jsou pro uživatele velmi zpříjemní ovládání celého domu. Režim bývá platný pro celý dům a zahrnuje nastavení všech potřebných systémů. Nejznámější je nejspíše režim spánku. Kdy se ztlumí topení na noční teplotu, v některých místnostech se zcela vypne. Zapomenutá rozsvícená světla, popřípadě elektronika se také vypne. Spustí se alarm v přízemí, zatáhnou se žaluzie a zamknou se dveře. Scén může být velmi mnoho, ještě pro příklad uvedu režim romantický večer, návštěva, večírek nebo režim dovolená. Právě poslední zmiňovanou popíšu podrobněji.

4.1.6.1. Simulace života v domě

Režim simulace života, je velmi výhodný právě když s rodinou vyrazíte někam na dovolenou. Smyslem scény je simulace přítomnosti osob, která má působit na potencionální zloděje jako obydlený dům – ačkoliv je ve skutečnosti prázdný. Systém automaticky každý večer rozsvěcí a zhasíná světla, pouští televizi nebo rádio. Večer zatahuje rolety a ráno je zase vytahuje. Toto se může každý den opakovat stále stejně, nebo je možné nastavit pseudonáhodné kombinace různých přístrojů.

Obrázek 4.15: Ukázka simulace života v domě

4.2 Astroprogram

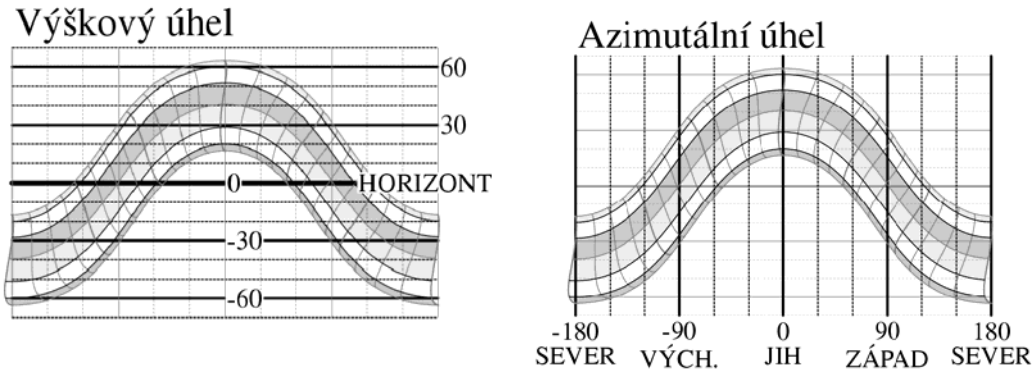
Jedním z dalších námětů, které vznikly díky požadavkům náročnějších uživatelů je například Astroprogram. Díky němuž lze určit polohu slunce, ze které je možno vypočíst i úhly natočení rolet nebo solárních panelů. Další z možností je ovládání veřejného osvětlení za přispění dalších senzorů. V publikacích je to prezentováno pro jedno místo (zeměpisnou šířku a délku, teoretickou rovinu) Na obrázku 4.19. je názorně vidět, že je to svazek křivek. Každá pro určitý den a na ní jsou vyznačen denní čas. V této práci jsem čerpal hlavně z materiálů uvedených v literatuře. [14]

Našim cílem je získat polohu slunce, tedy výškový úhel a azimutální úhel (v kartézských souřadnicích). Tedy na vstupu budeme v nejjednodušším případě zadávat *DATUM* a *DEN*. Pokud bychom chtěli určovat polohu kdekoli na Zemi museli bychom zadávat i zeměpisnou polohu. O té více v kapitole 4.2.1.4. Poté bychom museli přepočítat údaje z křivek a uložit do dvourozměrné datové struktury a z ní pak vybírat funkční hodnoty. Případně také využití interpolace. Nalézt funkční závislost a naprogramovat vzorec (pro datum a čas) a ten pak pro aktuální hodnoty vyčíslit. Bohužel přesný vzorec se mi nepodařilo objevit, jelikož je to velmi složité a přesahovali by to rámec bakalářské práce. Dále je zapotřební najít závislost pro proměnnou zeměpisnou polohu, závislost na letopočtu je pro námi vymezené účely zanedbatelné.

4.2.1 Grafy poloh slunce – základní údaje

Poloze slunce, přesněji poloze středu slunečního kotouče na obloze v nějakém okamžiku odpovídá v grafu bod.

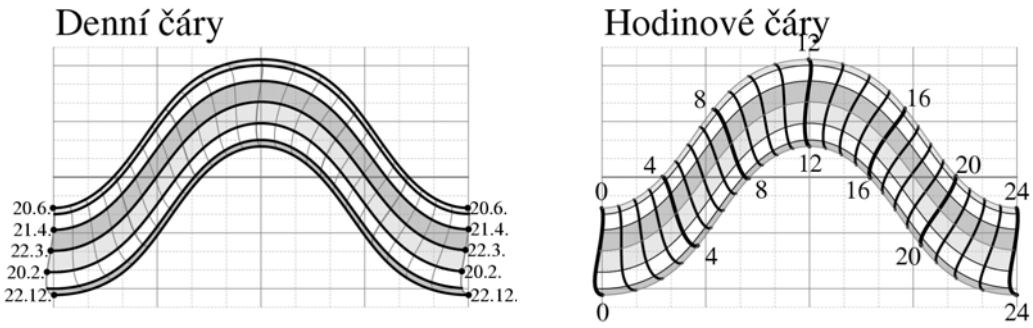
Výškový úhel je na svislé ose a ***azimutální úhel*** na ose vodorovnou.



Obrázek 4.16: a) Výškový úhel b) Azimutální úhel

Denní čára. Množině poloh, kterými slunce prochází během celého kalendářního dne, odpovídá v grafu souvislá čára, denní čára chodu slunce. Pro různé dny v roce dostáváme různé denní čáry.

Hodinová čára. Na každé denní čáře můžeme vymezit body odpovídající různým denním dobám. Spojíme-li na různých denních čarách body odpovídající stejné denní době, dostaneme souvislou čáru, kterou nazýváme hodinovou čarou.



Obrázek 4.17: a) Denní řáry b) Hodinové čáry

4.2.1.1. Výškový úhel h

Je úhel, který sluneční paprsek svírá s vodorovnou rovinou. Když je slunce na horizontu, je výškový úhel nulový, pro slunce nad horizontem je kladný.

Jedná se o výškovou polohu vzhledem k idealizovanému horizontu. Skutečný zemský povrch a skutečný horizont však bývá složitější (kopce, budovy). Idealizovanému modelu se dobře blíží situace, kdy bychom byli na moři daleko od pevniny.

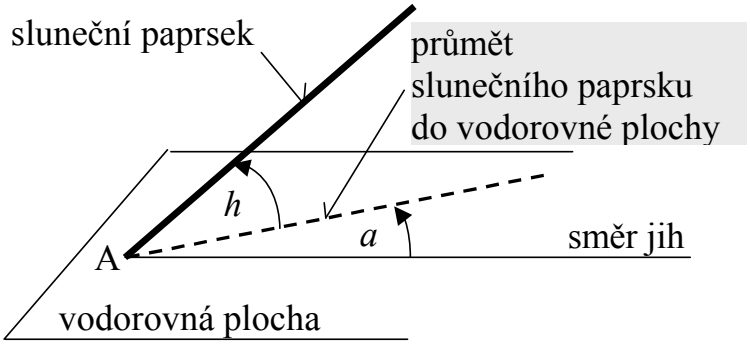
Důležité výškové úhly: 0°, západ slunce, střed slunce zapadá za horizont
-6°, mez občanského soumraku
-18°, mez hvězdářského soumraku

Úhlům 0° až –6° odpovídá ***občanský soumrak.*** Úhlům do –18° ***hvězdářský soumrak***. Při úhlech pod –18° je ***temná noc***; slunce je tak hluboko pod obzorem, že lze dobře provádět astronomická pozorování. Maximální výškový úhel 63,4° nastává v den letního slunovratu 21.6. Nejnižší denní maximum hodnoty výškového úhlu 16,6° nastává v den zimního slunovratu 22.12.

4.2.1.2. Azimutální úhel a

Je úhel sevřený průmětem slunečního paprsku do vodorovné roviny a polopřímkou ležící ve vodorovné rovině a směřující na jih; vymezuje, ze které světové strany slunce svítí. Pro slunce svítící z jihu je azimutální úhel roven nule, pro slunce na východě má hodnotu -90°, na západě +90°.

Azimutální úhel západu slunce se během roku mění, dne 21.6. je maximální $a = 130°$, dne 22.12. minimální $a = 53°$. V den letního slunovratu slunce zapadá téměř na severovýchodě.



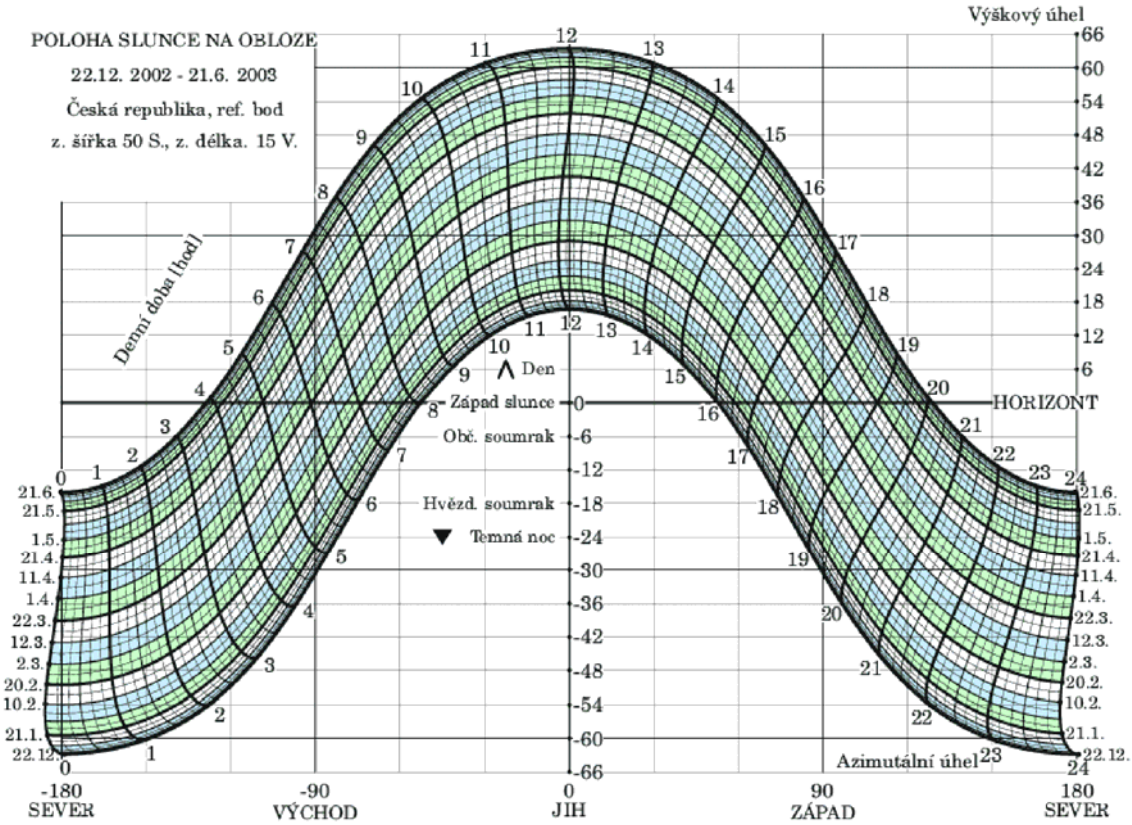
Úhly polohy slunce

h výškový úhel

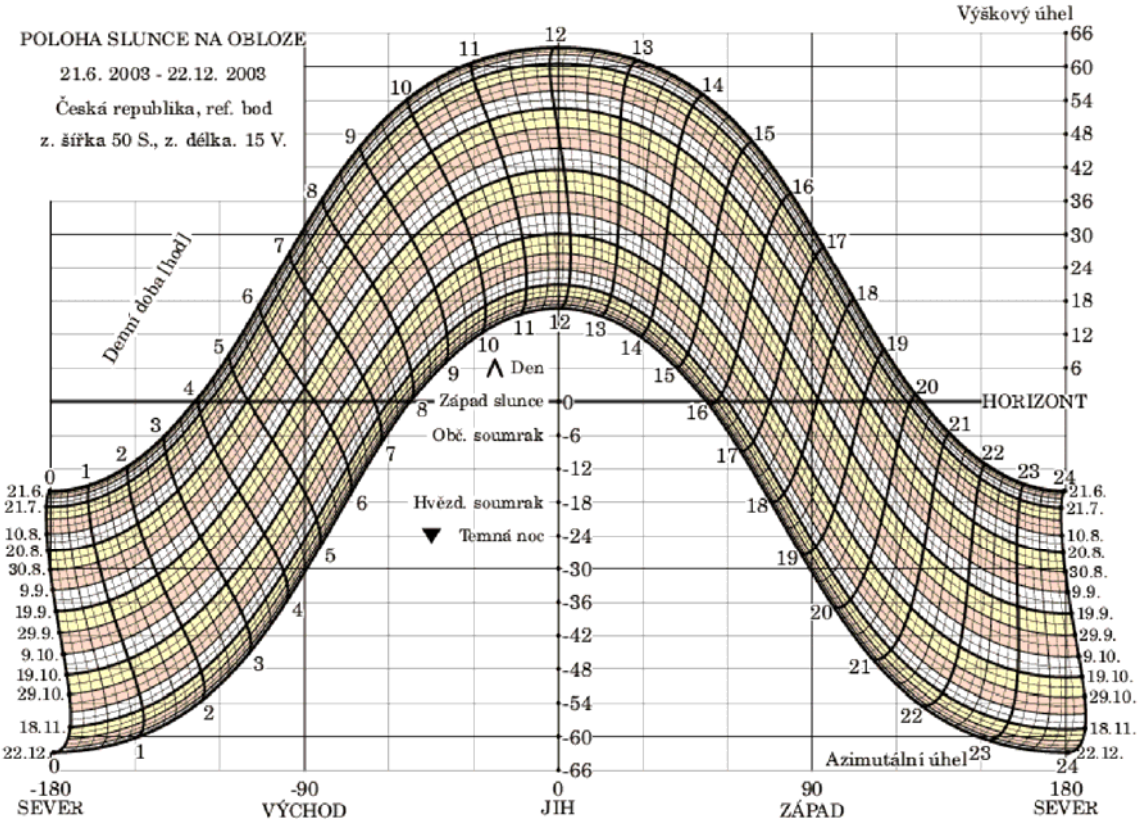
a azimutální úhel

Obrázek 4.18: Výškový a azimutální úhel

4.2.1.3. Dvojice základních grafů pro ČR a jejich využití



Obrázek 4.19: Poloha slunce na obloze od 22.12 do 21.6



Obrázek 4.20: Poloha slunce na obloze od 21.6 do 22.12

4.2.1.4. Stanovení zeměpisné polohy místa

Abychom pro nějaký bod zemského povrchu mohli vytvořit graf poloh slunce, potřebujeme pro toto místo zjistit ***zeměpisnou šířku***, ***zeměpisnou délku*** místa, ***referenční poledník časového pásma*** a ***nadmořskou výšku***.

V České republice a ve velké části Evropy se používá středoevropský čas, kterému odpovídá referenční poledník 15° východní délky. Pro města a obce ČR, i malé, lze zeměpisnou šířku a délku s přesností na 0,01" (0,2 m) najít na internetu. [14]

4.2.1.5. Naznačení řešení pomocí tabulek

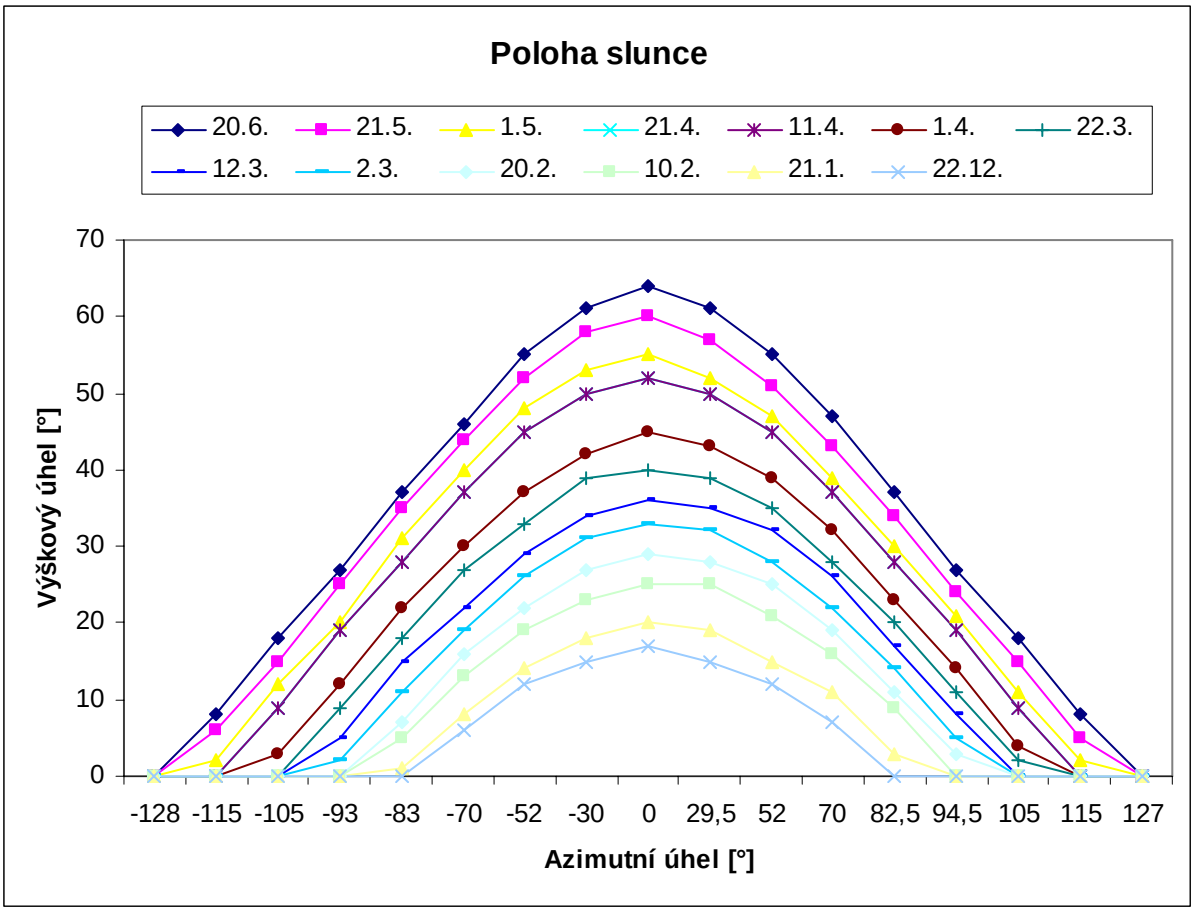
Řešením by mohlo být, jak již bylo výše uvedeno, pomocí dvou tabulek (Tab.:4.1 a Tab.:4.2). Průběh výsledné funkce je na Obr.: 4.21 a je s křivkou Obr.:4.19 téměř totožný

Výškový úhel																	
	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
20.6.	0	8	18	27	37	46	55	61	64	61	55	47	37	27	18	8	0
21.5.	0	6	15	25	35	44	52	58	60	57	51	43	34	24	15	5	0
1.5.	0	2	12	20	31	40	48	53	55	52	47	39	30	21	11	2	0
21.4.	0	0	9	19	28	37	45	50	52	50	45	37	28	19	9	0	0
11.4.	0	0	6	17	25	33	41	47	48	46	41	34	26	16	7	0	0
1.4.	0	0	3	12	22	30	37	42	45	43	39	32	23	14	4	0	0
22.3.	0	0	0	9	18	27	33	39	40	39	35	28	20	11	2	0	0
12.3.	0	0	0	5	15	22	29	34	36	35	32	26	17	8	0	0	0
2.3.	0	0	0	2	11	19	26	31	33	32	28	22	14	5	0	0	0
20.2.	0	0	0	0	7	16	22	27	29	28	25	19	11	3	0	0	0
10.2.	0	0	0	0	5	13	19	23	25	25	21	16	9	0	0	0	0
21.1.	0	0	0	0	1	8	14	18	20	19	15	11	3	0	0	0	0
22.12.	0	0	0	0	0	6	12	15	17	15	12	7	0	0	0	0	0

Tabulka. 4.1: Výškový úhel

Azimutální úhel																	
	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
20.6.	-128	-115	-105	-93	-83	-70	-52	-30	0	30	52	70	83	95	105	115	127
21.5.	0	-113	-103	-91	-77	-65	-48	-26	1	28	50	67	80	92	102	114	0
1.5.	0	-107	-100	-89	-75	-62	-44	-23	0	25	46	63	76	88	99	112	0
21.4.	0	-106	-98	-88	-74	-60	-42	-22	0	23	43	60	74	85	97	108	0
11.4.	0	0	-93	-86	-73	-57	-41	-22	0	22	41	57	71	83	94	0	0
1.4.	0	0	-91	-80	-72	-55	-39	-22	0	19	37	54	68	80	95	0	0
22.3.	0	0	-90	-78	-68	-54	-37	-22	-1	17	35	51	64	77	92	0	0
12.3.	0	0	0	-77	-66	-52	-36	-21	-2	16	33	48	62	74	89	0	0
2.3.	0	0	0	-75	-64	-51	-35	-21	-3	14	31	45	59	71	0	0	0
20.2.	0	0	0	0	-62	-49	-34	-20	-4	13	29	43	57	68	0	0	0
10.2.	0	0	0	0	-60	-48	-33	-20	-3	12	27	41	54	0	0	0	0
21.1.	0	0	0	0	-54	-44	-31	-18	-1	12	27	40	52	0	0	0	0
22.12.	0	0	0	0	-53	-41	-27	-14	0	14	27	40	52	0	0	0	0

Tabulka. 4.2: Azimutální úhel



Obrázek 4.21: Poloha slunce na obloze

Kapitola 5

5. ZÁVĚR

V této bakalářské práci jsem se zabýval tématem inteligentních budov, návrhem knihovny funkčních bloků a některými doporučenými postupy. Cílem bylo navrhnout a realizovat vytypované funkce, které budou uživatelé v praktickém životě nejvíce využívat. Zde v závěru bych zrekapituloval jednotlivé části bakalářské práce.

První kapitola obsahuje základní informace o inteligentních budovách, programovatelných automatech a mezinárodní normě IEC EN 61 131-3. V druhé kapitole jsem analyzoval typické potřeby uživatelů inteligentních domů.

Třetí kapitolu jsem věnoval realizaci vytypovaných funkcí a funkčních bloků. Zde jsem popsal jejich praktické využití, popřípadě jejich kombinaci s dalšími funkcemi.

Ve čtvrté kapitole jsem jednotlivé funkce či funkční bloky popsal pomocí jazyka SFC v kombinaci se slovním popisem. Jednu z podkapitol jsem věnoval popisu realizace programu Astroprogram, která může mít v praxi mnoho využití. Poslední kapitola popisuje manuál knihovny IB_Lib, kterou jsem připravil za účelem praktické využití. Zde v bakalářské práci neuvádím zdrojové kódy, které jsou k dispozici v přiloženém CD spolu s knihovnou IB_Lib.

Literatura

[1] VALEŠ, M. *Inteligentní dům*. Brno : Vydavatelství ERA, 2006.

[2] *TZB-INFO.CZ* [online], Portál zaměřený na technická zařízení budov, http://www.tzb-info.cz/

[3] *KNX.ORG* [online], Portál obsahující informace o standardu KNX a EIB, http://www.knx.org/

[4] *Wikipedia* [online], http://www.en.wikipedia.org/

[5] ZVOLÁNEK, M., *Bakalářská práce: Inteligentní řízení a zabezpečení domů s využitím informačních a komunikačních technologií*, Praha 2008

[6] KLABAN, J., *Inels a sběrnice CIB*. AUTOMA. 2008, č. 12, *http://www.odbornecasopisy.cz/index.php?id_document=38218*

[7] Teco DVD INFO 10/2008, Teco a. s., Kolín, 2008

[8] URBAN, L., Tecomat FOXTROT. AUTOMA. 2007, č. 10, *http://www.odbornecasopisy.cz/index.php?id_document=34290*

[9] *Comforthouse* [online], http://www.comforthouse.cz/

[10] ŠUSTA, R., *Přednášky X35LOR: Realizace řídících automatů*, Praha 2008

[11] *INELS* [online], http://www.inels.cz/

[12] HANZÁLEK Z., *Petriho sítě a GRAFCET*, CAK 2001

[13] *Přednášky X33SDU: Systémy diskrétních událostí*, Praha 2008

[14] POLÁČEK, L.: *Poloha slunce na zemské obloze*, Brno 2003

Přílohy

A – Manuál knihovny IB_Lib

B – CD s elektronickými dokumenty a knihovnou IB_Lib

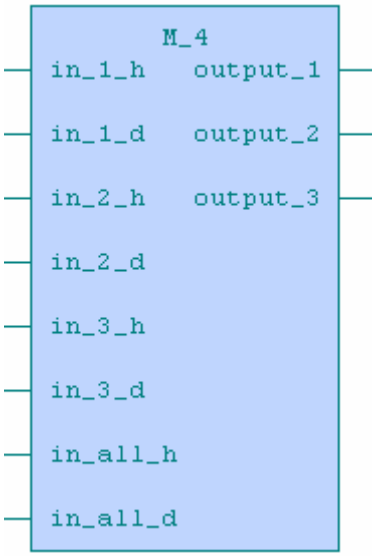
Manuál knihovny IB_Lib

Obsah:

1. Čtyř-tlačítko s nestálými polohami (M_4) I
2. Ovládání schodiště jedním tlačítkem (T-FLOP)..... II
3. Vyvětrání místnosti (WC_ven1)..... III
4. Osvětlení před domem (SV_DUM).....IV
5. Majorita ze tří (maj3)..... V
6. Prahové funkce VI
7. Elementární symetrické funkce VII

1. Čtyř-tlačítko s nestálými polohami (M_4)

Funkční blok slouží k ovládání osvětlení schodiště či místnosti. Ovladač má čtyři tlačítka z nichž třem přiřadíme funkci rozsvěcení a zhasínání tří různých svítidel v místnosti (lampa u křesla, osvětlení za televizorem a intenzivní světlo z lustru). Poslední tlačítko slouží k ovládání celé skupiny svítidel naráz (stiskem do horní polohy se všechna svítidla aktivují, stiskem do dolní polohy se všechna vypnou). Kliknutím do horní části (polohy) tlačítka aktivuji (rozsvítím), kliknutím do dolní polohy deaktivuji (zhasnu).



Obr. 1 - Funkční blok čtyř-tlačítko

<u>Vstupem bloku je signál z osmi poloh tlačítek:</u>

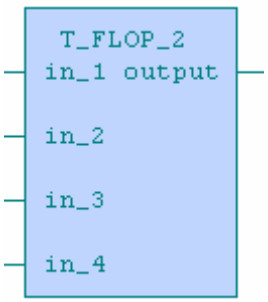
Tlačítko 1 má dvě polohy:	- horní (in_1_h)
	- dolní (in_1_d)
Tlačítko 2 má dvě polohy:	- horní (in_2_h)
	- dolní (in_2_d)
Tlačítko 3 má dvě polohy:	- horní (in_3_h)
	- dolní (in_3_d)
Tlačítko 4 má dvě polohy:	- horní (in_all_h)
	- dolní (in_all_d)

<u>Výstupem je signál vyslaný příslušnému svítidlu:</u>

Výstup pro lampu:	output_1
Výstup pro televizoru:	output_2
Výstup pro lustr:	output_3

2. Ovládání schodiště jedním tlačítkem (T-FLOP)

Funkční blok slouží k ovládání osvětlení schodišť a chodeb. Zde je požadavkem, aby světlo na chodbě či schodišti bylo ovládáno jedním kontaktem (např. na jednom konci chodby stiskem rozsvítím a na druhém konci zhasnu), tzn. ovládání osvětlení, kdy každým stiskem (s náběžnou hranou) se stav svítidla změní na opačný.



Obr. 2 - Funkční blok T-FLOP

<u>Vstupem bloku je signál ze čtyř tlačítek:</u>

Tlačítko 1: - horní (in_1)
Tlačítko 2: - horní (in_2)
Tlačítko 3: - horní (in_3)
Tlačítko 4: - horní (in_4)

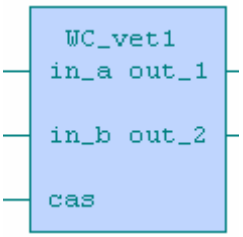
<u>Výstupem je signál vyslaný příslušnému svítidlu:</u>

Výstup osvětlení: output

3. Vyvětrání místnosti (WC_ven1)

Funkční blok WC_vet1 slouží k ovládání ventilátoru a příslušného světla. Po stisku tlačítka *in_a* (nebo signálem z čidla PIR) se spustí ventilátor a bude běžet nastavenou dobu (nastaví si zákazník sám, přednastaveny budou 2 minuty). Tlačítko může současně sloužit i k rozsvícení (nebo ke zhasnutí svítidla) - podle toho, přejeme-li si aby ventilátor běžel současně se světlem (patrně po dobu naší přítomnosti) nebo až od okamžiku zhasnutí (po našem odchodu) z místnosti.

Přednastaveno je, aby po zhasnutí ventilátor běžel nastavenou dobu (přednastaveno 15 sekund) ještě sám.



Obr. 3 - Funkční blok WC_vet1

<u>Vstupem bloku je signál z tlačítka:</u>

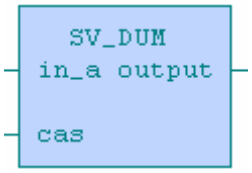
Tlačítko má dvě polohy:	- horní (in_a) - rozsvícení
	- dolní (in_b) – zhasnutí + ventilátor
Čas pro dobu větrání	- cas

<u>Výstupem je signál vyslaný svítidlu a ventilátoru:</u>

Výstup pro rozsvícení:	out_1
Výstup pro zhasnutí + ventilátor:	out_2

4. Osvětlení před domem (SV_DUM)

Funkční blok SV_DUM má za úkol rozsvítit světlo až 30 sekund poté, co senzor zaznamená aktivitu (zaregistroval přítomnost osoby). Po skončení aktivace senzoru zhasne okamžitě. Světlo tak nereaguje na chodce, kteří jen procházejí kolem vchodu ani na psy a menší živočichy. Rozsvítí se až při dlouhodobé aktivaci senzoru PIR.



Obr. 4 - Funkční blok SV_DUM

<u>Vstupem bloku je signál z čidla PIR:</u>

Senzor PIR: - in_a
Doba čekání: - cas

<u>Výstupem je signál:</u>

Výstup pro rozsvícení: - output

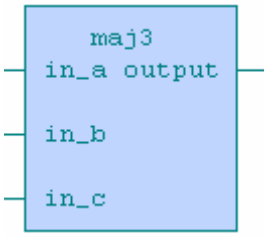
5. Majorita ze tří (maj3)

Funkce majority (většiny) se může definovat i pro jiný lichý počet operandů než tři a je pravdivá pro nadpoloviční počet pravdivých operandů.

Majorita ze tří je funkce, která je pravdivá pokud nadpoloviční většina (dva nebo tři) z jejich operandů je pravdivých.

Je definována logickým výrazem:

$$maj3(a,b,c) = (a\ AND\ b)\ OR\ (a\ AND\ c)\ OR\ (b\ AND\ c)$$



Obr. 5 - Funkční blok maj3

<u>Vstupem bloku je signál ze tří senzorů:</u>

Senzor rozbití skla:	in_a
Čidlo pohybu:	in_b
Dveřní kontakt:	in_c

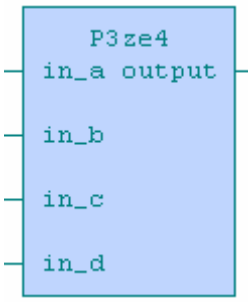
<u>Výstupem je signál alarmu:</u>

Výstup alarm:	output

6. Prahové funkce

Jsou to logické funkce, které jsou typu alespoň k ze tří, tedy alespoň 1 ze tří, , alespoň 2 ze tří, alespoň 3 ze tří , atd. Souhrnně tyto funkce označujeme jako prahové funkce a zkrácené označení pro jednotlivé bloky je P1ze3, P2ze3, P3ze3, P1ze4, P2ze4, P3ze4, P4ze4, P1ze5, P2ze5, P3ze5, P4ze5, P5ze5.

Obecně ***Pk_n***, kde první číslo představuje hodnotu prahu a druhé je počtem operandů. Lze je definovat pro libovolný počet operandů a pro libovolné hodnoty prahu. Obecně platí, že jsou pravdivé, pokud počet pravdivých operandů je alespoň roven (roven nebo větší) prahu (prvnímu číslu z názvu funkce).



Obr. 6 - Funkční blok P3ze4

<u>Vstupem bloku je signál ze senzorů:</u>

Senzor 1: in_a
Senzor 2: in_b
Senzor 3: in_c
Senzor 4: in_d
(popřípadě: Senzor 5: in_e)

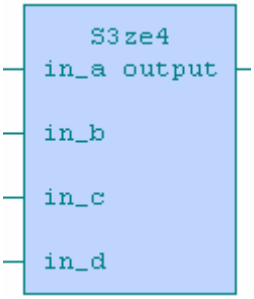
<u>Výstupem je signál alarmu:</u>

Výstup alarm: output

7. Elementární symetrické funkce

Jsou to logické funkce, které jsou slovně popsány právě k ze tří, konkrétně právě 1 ze 3, právě 2 ze 3 a právě 3 ze 3.

Obecně pak právě **k** z **n**. Stručně je budeme označovat jako S1ze3, S2ze3, S3ze3, S1ze4, S2ze4, S3ze4, S4ze4, S1ze5, S2ze5, S3ze5, S4ze5, S5ze5, obecně S k ze n. Jsou pravdivé pokud počet pravdivých operandů je právě roven charakteristickému číslu - prvnímu číslu z označení funkce.



Obr. 7 - Funkční blok S3ze4

<u>Vstupem bloku je signál ze senzorů:</u>

Senzor 1: in_a
Senzor 2: in_b
Senzor 3: in_c
Senzor 4: in_d
(popřípadě: Senzor 5: in_e)

<u>Výstupem je signál alarmu:</u>

Výstup alarm: output