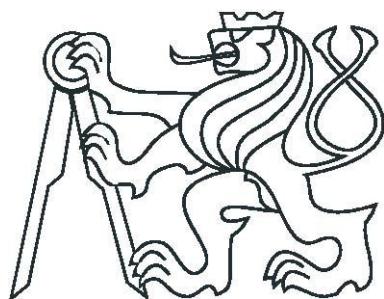


**ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE**  
**FAKULTA ELEKTROTECHNICKÁ**



**BAKALÁŘSKÁ PRÁCE**

Algoritmy a PLC pro inteligentní domy

Praha, 2011

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: **Algoritmy a PLC programy pro intelligentní domy**

### Pokyny pro vypracování:

1. Seznamte se se zásadami programování PLC podle standardu IEC EN 61131-3, s programovatelnými automaty Tecomat Foxtrot, s distribuovaným systémem Inels II, s vývojovým prostředím Mosaic.
2. Analyzujte funkce intelligentního domu z pohledu řízení, zabezpečení a diagnostiky. Vytipujte a popište standardně se vyskytující algoritmy.
3. Vytipované algoritmy realizujte programem PLC podle zásad normy IEC EN 61131-3.
4. Programy otestujte a vytvořte uživatelskou dokumentaci s metodikou použití.

### Seznam odborné literatury:

- Martinášková, M. - Šmejkal, L. : Řízení programovatelnými automaty III. Softwarové vybavení, ČVUT, Praha 2003  
Šmejkal, L. - Martinášková, M.: PLC automatizace 1. Základní pojmy a úvod do programování. BEN, Praha 1999  
Šmejkal, L.: PLC a automatizace 2. Sekvenční logické funkce minimum o fuzzy logice. BEN, Praha 2005  
Kol.: Programování PLC podle normy IEC 61 131-3, TECO, Kolín 2007  
[www.tecomat.cz](http://www.tecomat.cz)  
[www.elcom.cz](http://www.elcom.cz)

Vedoucí: Ing. Martin Hlinovský, Ph.D.

Platnost zadání: do konce zimního semestru 2011/2012

prof. Ing. Michael Šebek, DrSc.  
vedoucí katedry



prof. Ing. Boris Šimák, CSc.  
děkan

V Praze dne 3. 1. 2011

## **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**

Bakalářská práce popisuje inteligentní domy, jejich základní pojmy a široké možnosti jejich využití. Dále zde naleznete názorné ukázky programování pomocí PLC dle normy IEC EN 61131-3. Práce obsahuje souhrn nejvýznamnějších funkcí a funkčních bloků, které se používají v současných inteligentních domech. Zejména funkce a funkční bloky vyskytující se v oblasti řízení, diagnostiky a zabezpečení.

## **The Abstract**

The bachelor work describes the intelligent houses, their basic concepts and extensive possibilities of utilization. Then you can find vivid examples of programming using PLC according to the norm IEC EN 61131-3. This work includes the summary of the most significant functions and functional blocks used in the present intelligent houses. Especially functions and functional blocks occurring in the sphere of control, diagnostics and security are described.

# Obsah

1.	Teoretická část .....	1
1.1	Úvod .....	1
1.2	Základní pojmy a uspořádání .....	3
1.3	Uspořádání řídících systémů .....	5
1.3.1	Centralizované systémy .....	5
1.3.2	Decentralizované systémy .....	6
1.3.3	Hybridní systémy .....	8
1.4	Programovatelný automat - PLC .....	8
1.4.1	Princip vykonávání uživatelského programu .....	9
1.4.2	PLC Tecomat .....	9
1.4.3	Tecomat Foxtrot .....	10
1.4.4	Sběrnicový systém INELS .....	11
1.4.5	Vývojové prostředí MOSAIC .....	12
1.5	Mezinárodní norma IEC EN 61 131-3 .....	14
1.5.1	Společné prvky .....	14
1.5.2	Programovací jazyky .....	15
1.5.3	Programové organizační jednotky .....	17
1.5.4	Funkce a její syntaxe .....	17
1.5.5	Funkční bloky a syntaxe .....	17
1.5.6	Programy .....	18
1.5.7	Struktura programové organizační jednotky (POU) .....	18
1.6	Složitější sekvenční funkce .....	20
1.6.1	Jazyk SFC (Grafcet) .....	24
1.6.2	Základní koncepce .....	24
2.	Analýza typických funkcí .....	26
2.1	Analýza z pohledu zabezpečení .....	26
2.2	Analýza z pohledu diagnostiky .....	26
2.3	Analýza z pohledu řízení .....	27
3.	Realizace funkcí a funkčních bloků .....	28
3.1	Realizace funkcí .....	28
3.1.1	Funkce Paměťové funkce .....	28
3.1.2	Schodiště a chodby .....	30
3.2	Zabezpečovací technika .....	34
3.2.1	Majorita .....	34
3.2.2	Prahové funkce .....	36
3.2.3	Elementární symetrické funkce .....	36
3.3	Ovládání mechanismů .....	37
3.3.1	Ovladač roletových žaluzí SBC1 .....	37
4.	Návrh specifických bloků .....	39
4.1	Rozpoznání dlouhého a krátkého stisku .....	39
4.2	Funkce ovládání mechanismů .....	40
4.2.1	Synergie domu .....	41
4.3	Astroprogram .....	43
4.3.1	Množství energie dopadající na zemský povrch .....	43
5.	Závěr .....	49

# Seznam obrázků

Obrázek 1.1:	Schéma centralizovaného systému .....	6
Obrázek 1.2:	Schéma decentralizovaného systému.....	7
Obrázek 1.3:	Cyklus řešení uživatelského programu .....	9
Obrázek 1.4:	Základní modul Tecomat Foxtrot spolu s připojenými binárními moduly ..	11
Obrázek 1.5:	Ukázka vývojového prostředí Mosaic.....	13
Obrázek 1.6:	Ukázka logické funkce AND ve všech čtyřech jazycích .....	16
Obrázek 1.7:	Základní struktura POU .....	18
Obrázek 1.8:	Základní struktura POU programu (deklarační i výkonná část).....	19
Obrázek 1.9:	Automat typu Mealy .....	20
Obrázek 1.10:	Automat typu Moore .....	21
Obrázek 1.11:	a) Vývojový diagram (Mealy) b) Vývojový diagram (Moore) .....	21
Obrázek 1.12:	Ukázka jazyka SFC .....	22
Obrázek 1.13:	Ukázka simultánního větvení jazyka SFC.....	23
Obrázek 1.14:	a) Počáteční krok b) Přechod.....	24
Obrázek 1.15:	a) Divergentní AND b) Konvergentní AND .....	25
Obrázek 1.16:	a) Divergentní OR b) Konvergentní OR .....	25
Obrázek 3.1:	Jedno tlačítko s dvěma nestálými polohami.....	29
Obrázek 3.2:	Funkční blok T_FLOP .....	30
Obrázek 3.3:	Průběh FB T_FLOP .....	30
Obrázek 3.4:	Vzhled FB DMA1 .....	31
Obrázek 3.5:	Průběh časovače TP .....	32
Obrázek 3.6:	Průběh časovače TON.....	33
Obrázek 3.7:	Karnaughova mapa majority ze 3 .....	35
Obrázek 3.8:	FB majorita ze 3.....	35
Obrázek 3.9:	a) Prahová funkce 1 ze 4 b) Prahová funkce 3 ze 4 .....	36
Obrázek 3.10:	Vzhled FB SBC1 .....	37
Obrázek 4.1:	Vzhled bloku rozpoznání stisku tlačítka .....	39
Obrázek 4.2:	Ukázka LD programu rozpoznání stisku tlačítka .....	39
Obrázek 4.3:	Vzhled bloku mechanického ovládání .....	40
Obrázek 4.4:	Diagram mechanického ovládání SFC .....	40
Obrázek 4.5:	Ukázka simulace života v domě .....	42
Obrázek 4.6:	Světová mapa solárního záření .....	43
Obrázek 4.7:	Světová mapa solárního záření .....	44
Obrázek 4.8:	a) Výškový úhel b) Azimutální úhel .....	45
Obrázek 4.9:	a) Denní čáry b) Hodinové čáry.....	45
Obrázek 4.10:	Výškový a azimutální úhel .....	45
Obrázek 4.11:	Vzhled bloku astropogram .....	46
Obrázek 4.12:	Poloha slunce na obloze od zima-jaro.....	47
Obrázek 4.13:	Poloha slunce na obloze léto-podzim.....	48

# Seznam tabulek

Tabulka 1.1:	Přehled systémů a jejich komunikace .....	8
Tabulka 1.2:	Porovnání všech typů PLC automatů firmy Teco, a.s.....	10
Tabulka 3.1:	Popis funkce třívstupového majoritního členu .....	35

# Kapitola 1

## 1. TEORETICKÁ ČÁST

### 1.1 Úvod

Inteligentní domy, nebo spíše v anglické literatuře popisované smart home je dům vybavený kombinací elektronických a informačních technologií. V anglickém jazyce je používán výraz „domotic“ v přeneseném slova smyslu to znamená „domov se zabudovanou technologií“. Díky těmto technologiím se stává domov pro své obyvatele něčím jedinečným v podobě bezpečí, zábavy a komfortu. Adaptace technologie do životního prostoru přináší duševní i fyzické pohodlí. Můžeme také zoptymalizovat naši spotřebu energie, čímž šetříme přírodu i naše provozní náklady. Intelligentní dům se dá taktéž charakterizovat jako budova s integrovanými řídicími systémy. Systém domácí automatizace je velmi odlišný od konceptu tradiční elektroinstalace v domech. Za nemalého přispění automatizace lze ovládat nejnovější technologie, jako je ovládání světel, žaluzií, audiovizuální technika, klimatizace, kamerového systému či vzdáleného ovládání. Všechny tyto technologie se dnes stávají běžnou součástí našich domovů a díky vzájemné integraci a synchronizaci všech prvků pak dokážeme z „obyčejného“ domu udělat „intelligentní“ dům. Tento systém domovní automatizace dokáže vytvořit ideální podmínky pro bydlení a život v domě. Koncept budoucího intelligentního domu musí být velmi propracován. Zde se pak klade důraz na spolupráci jednotlivých systémů, jako je například naklopení žaluzií spolu se sepnutím alarmu. Nedílnou součástí systému je samozřejmě propojení těchto částí a komunikace mezi nimi. Jednotlivé systémy potřebují sdílet informace mezi sebou, aby byla jejich integrace co možná nejúčinnější. Propojení jednotlivých systémů většinou umožňuje datový kabel s obousměrnou komunikací (UTP – unshielded twisted pair).

Dnes již není intelligentní dům žádnou novinkou, proto mohou některé dílčí součásti připadat velmi jednoduše a samozřejmě, avšak díky synchronizaci všech těchto součástí se stává dům intelligentní. 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 řídící systém zajišťuje např. řízení garáže, skleníku, bazénu, otevírání vrat, závlahu záhonů a trávníků, zabezpečení venkovních prostor apod. [1] [15]

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 funkcí a funkčních bloků pro inteligentní domy. Navrhoji převážně takové funkční bloky, které jsou podle mého názoru nejfrequentovanější mezi typickými uživateli. Dále pak pro náročnější uživatele uvádím několik konkrétnějších příkladů využití PLC.

## 1.2 Základní pojmy a uspořádání

Pro orientaci a lepší porozumění textu zde uvedu několik důležitých pojmy, které se objevují v intelligentních 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 (Syncro 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 rychlosť 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, rychlosť 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). Využívána 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 Protocol/Internet Protocol) – 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 rychlosťí 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 Möller. Jádrem tohoto řešení je decentralizovaný sběrnicový systém belgické firmy Niko nazývaný

Nikobus. Pro svoji jednoduchost a uživatelskou dostupnost nachází vhodné uplatnění v méně rozlehlych 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

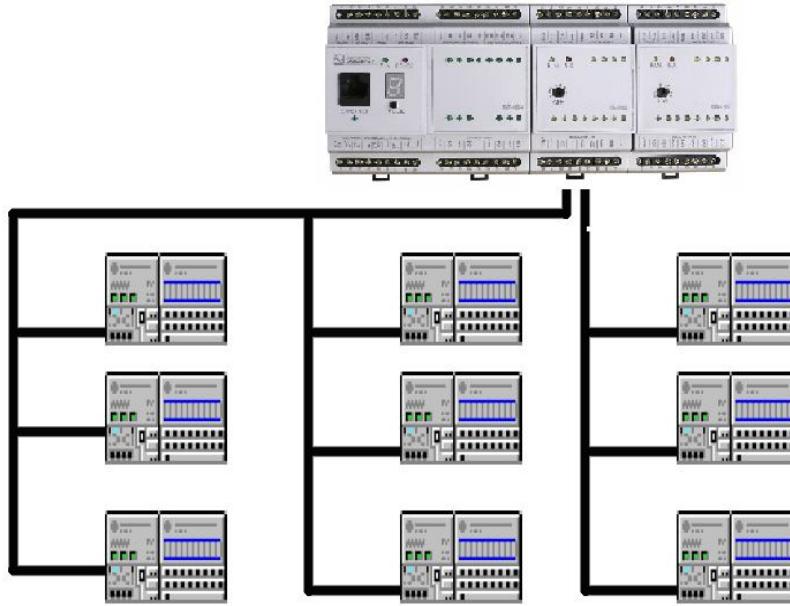
V současné době existují dva základní typy sběrnicových systémů nebo jejich kombinace. Jeden z nich se nazývá centralizovaný systém a druhý decentralizovaný. Kombinaci těchto dvou systémů nazýváme hybridním systémem. Stupeň centralizace závisí na počtu lokalit inteligentních prvků odpovídajících za řídící 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 využití dat o dění v objektu. Obrovskou nevýhodou je závislost jednotlivých systémů pouze na jedné jediné řídící jednotce. Jde o hlavní příčinu menší stability systému. Ryzé centralizované

systémy nejsou schopny plošně a efektivně pokrýt celou rozlohu budovy a díky vysokým nákladům výrazně převyšují výhody těchto systémů.



Obrázek 1.1: Schéma centralizovaného systému

#### **Výhody centralizovaného systému**

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

#### **Nevýhody centralizovaného systému**

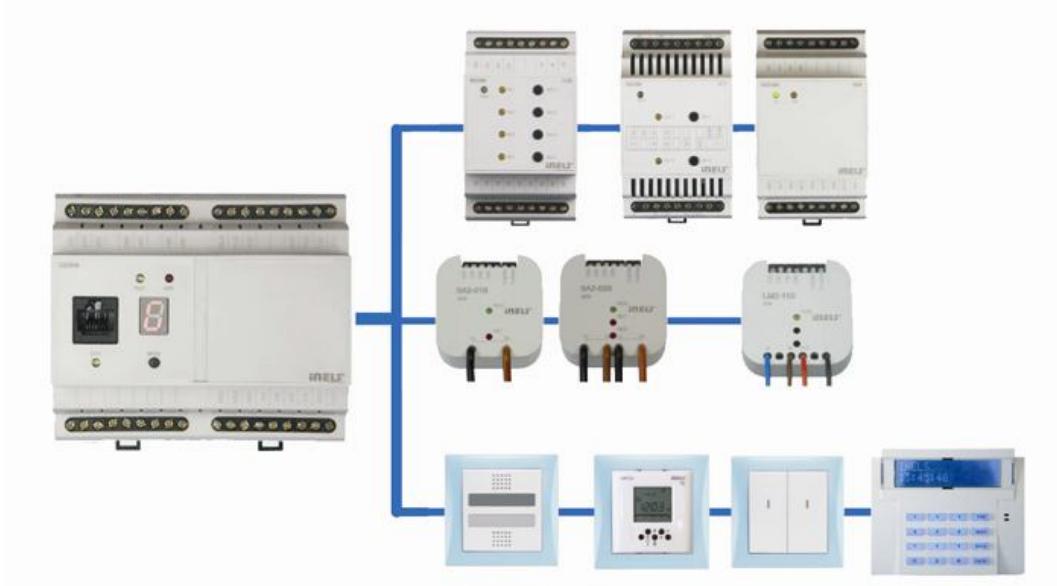
- Velké programy
- Výkon závisí na počtu I/O
- Závada jednoho PLC znamená zastavení celého programu [10]

### 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 využívání 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 1.2: 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

Otevřené systémy (standardKNX)			
	komunikace	počet přístrojů	uplatnění
ABB i-bus KNX/EIB	decentralizovaná	1 linie - 64 (256)	letiště, velké budovy
NikoBus	hybridní	256	rodinné domy
Inels	centralizovaná	64 až neomezeně	domovní instalace
Ego-n	centralizovaná	až 512	rekonstruované domy

Tabulka 1.1: Přehled systémů a jejich komunikace

### **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 komunikují navzájem a centrální jednotka je pouze sleduje.

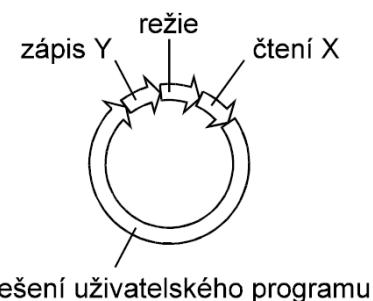
## **1.4 Programovatelný automat - PLC**

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 1.3: 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 vyzdvíženy 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 intelligentní 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
<b>TC400</b>	6/4	2/-	2 serial	externí	32 kB
<b>TC500</b>	20/20	4/4	2 serial	integrovaný	32 kB
<b>TC600</b>	48/44	24/8	3 serial	externí	32 kB
<b>TC650</b>	48/44	24/8	3 serial, 1 Ethernet	externí	64 + 64 kB
<b>Foxtrot</b>	až 134 DI/DO	až 80AI / 20AO	2× serial, 1×CIB, 1×Ethernet	externí	192 + 64 kB
<b>TC700</b>	přes 6500	až 3300/700	až 10 serial, 2× Ethernet, 1× USB	externí	192 + 64 kB

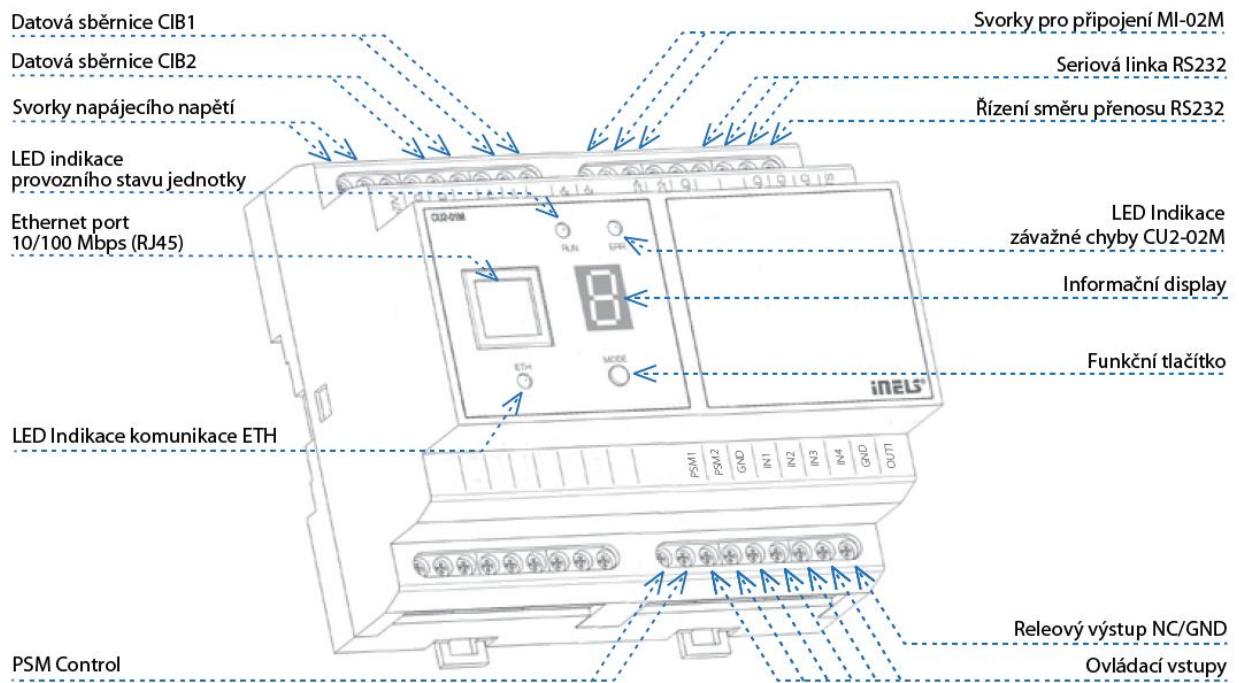
Tabulka 1.2: Porovnání všech typů PLC automatů firmy Teco, a.s.

#### 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 1.4: 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 instalací 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.

Po vyjmenování mnoha výhod přejdeme k reálným datům, jako je rychlosť 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. Rychlosť 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).

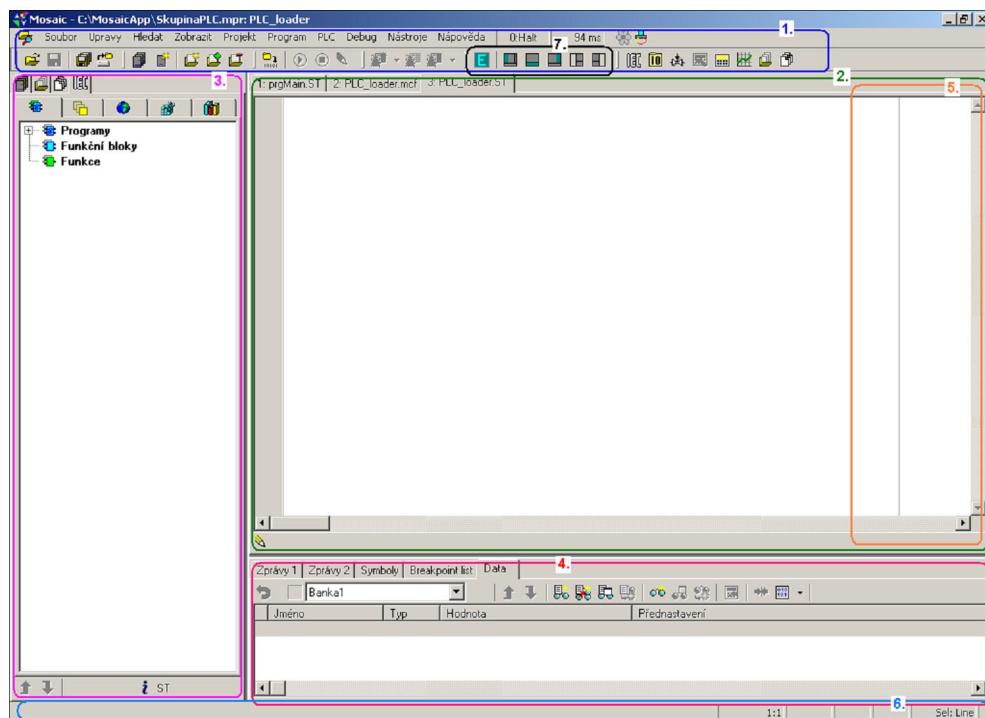
Programová forma je ve shodě s mezinárodní normou IEC EN 61 131-3 v textových jazyce IL (Instruction List = firemní mnemokód) a ST (Structured Text = strukturovaný text) a grafických jazycích LD (Ladder Diagram = reléové schéma) a FBD (Function Block Diagram = funkční bloky).

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 rádu 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 1.5: Ukázka vývojového prostředí Mosaic

#### Popis obrázku 1.4.3

1. V horní části hlavního okna je v rádku 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 rádku hlavního okna je informační rádek, kde se zobrazují informační texty a v pravé části informace z aktivního editoru, jako je číslo rádek : 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 zkušenosť 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 parametrem 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)

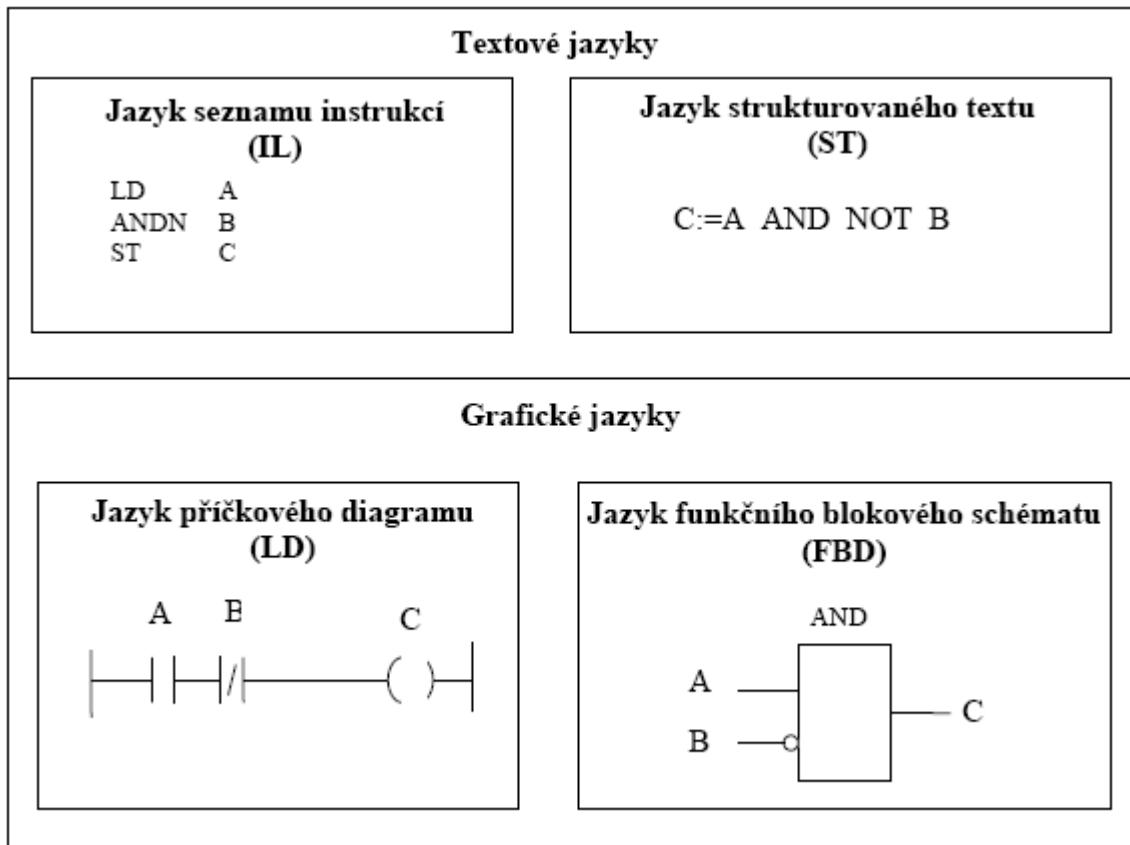
**FBD** - Function Block Diagram - jazyk funkčního blokového schématu

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 1.6: 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 POU). 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 opakováně.

### 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 opakováně 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.

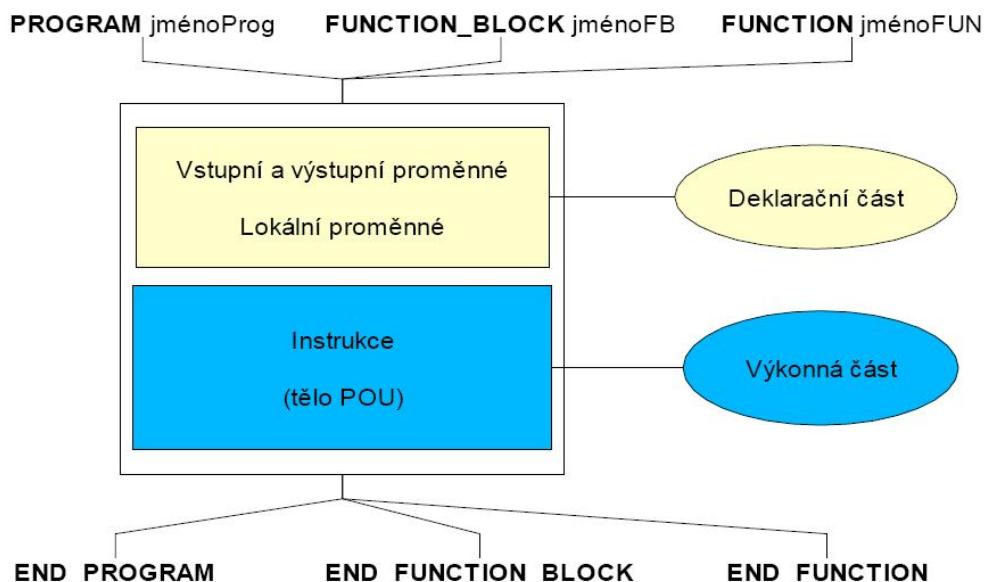
### 1.5.6 Programy

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.

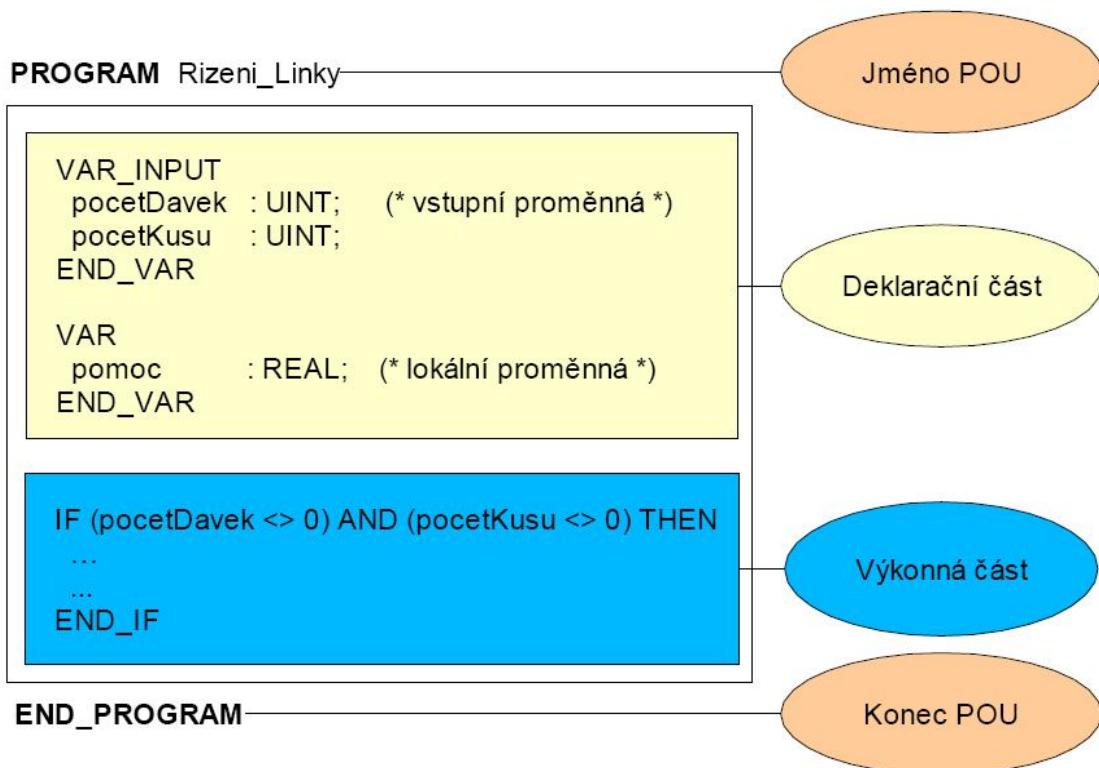


Obrázek 1.7: Základní struktura POU

#### 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 1.8: Základní struktura POU programu (deklarační i výkonná část)

#### 1.5.7.2. Výkonná část POU

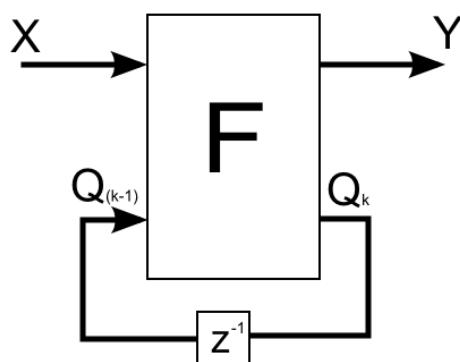
Výkonná část POU následuje za částí deklarační a obsahuje příkazy a instrukce, které jsou zpracovány centrální jednotkou PLC. Ve výjimečných případech nemusí definice POU obsahovat žádnou deklarační část a potom je výkonná část uvedena bezprostředně za definicí začátku POU. Příkladem může být POU, která 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.

## 1.6 Složitější sekvenční funkce

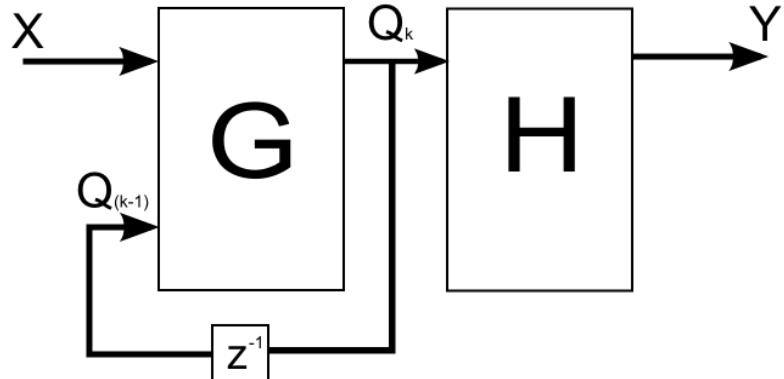
Funkční závislosti pro majoritní, prahové a elementární symetrické funkce mají ryze kombinační charakter a mohou tedy být realizovány jako POU typu funkce. V technice budov je ale mnoho logických funkčních závislostí, které mají výrazně sekvenční charakter, mnohdy poměrně komplikovaný. Pro ně je třeba 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.:1.10) 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 ( $Y = H(X, Q_{k-1})$ ).



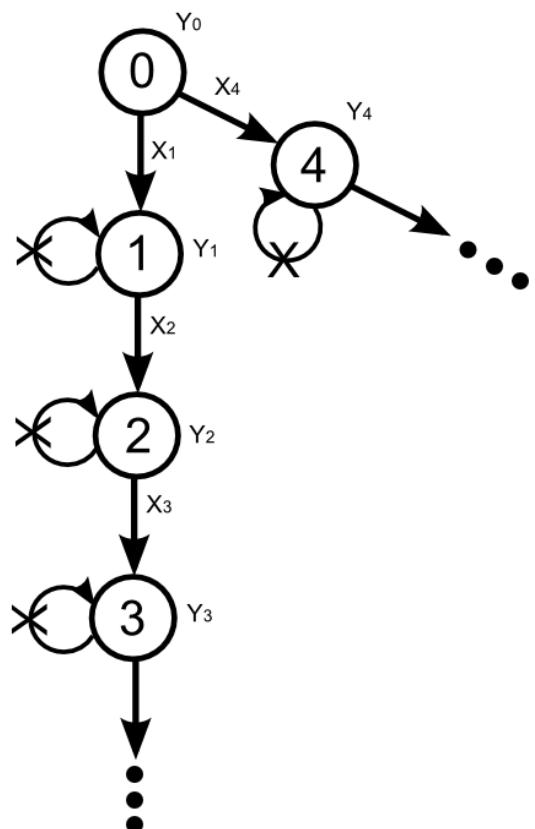
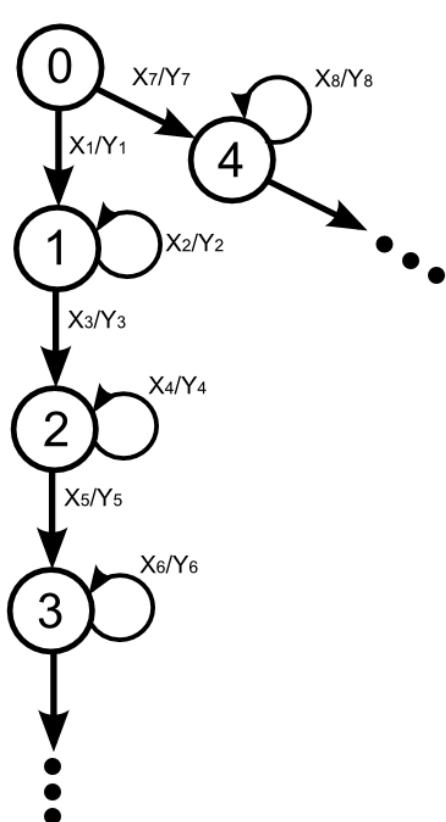
Obrázek 1.9: Automat typu Mealy

Automat **Mooreova typu** (Obr.: 1.11) má oddělenou přechodovou funkci ( $Q_k = G(X, Q_{k-1})$ ) a výstupní funkci ( $Y = H(Q_k)$ ), případně ( $Y = H(Q_{k-1})$ ). Argumentem výstupní funkce je jen stav (současný, případně minulý - rozdíl je jen o zpoždění jednoto výpočetního cyklu).



Obrázek 1.10: Automat typu Moore

Vhodným grafickým prostředkem pro popis sekvenčního chování je přechodový diagram - na Obr.: 1.12a pro automat Mealyho typu nebo na Obr.: 1.12b pro automat Mooreova typu.



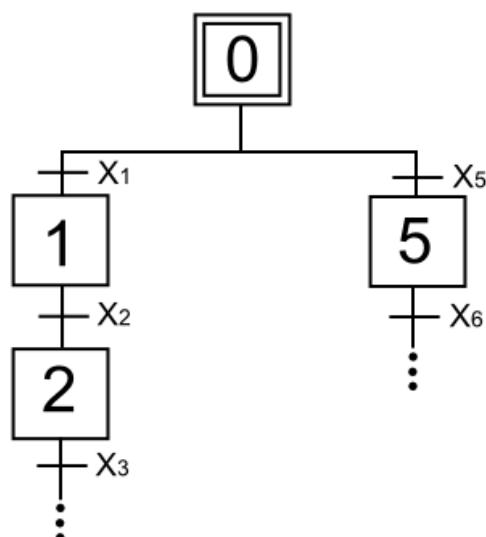
Obrázek 1.11: a) Vývojový diagram (Mealy) b) Vývojový diagram (Moore)

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.

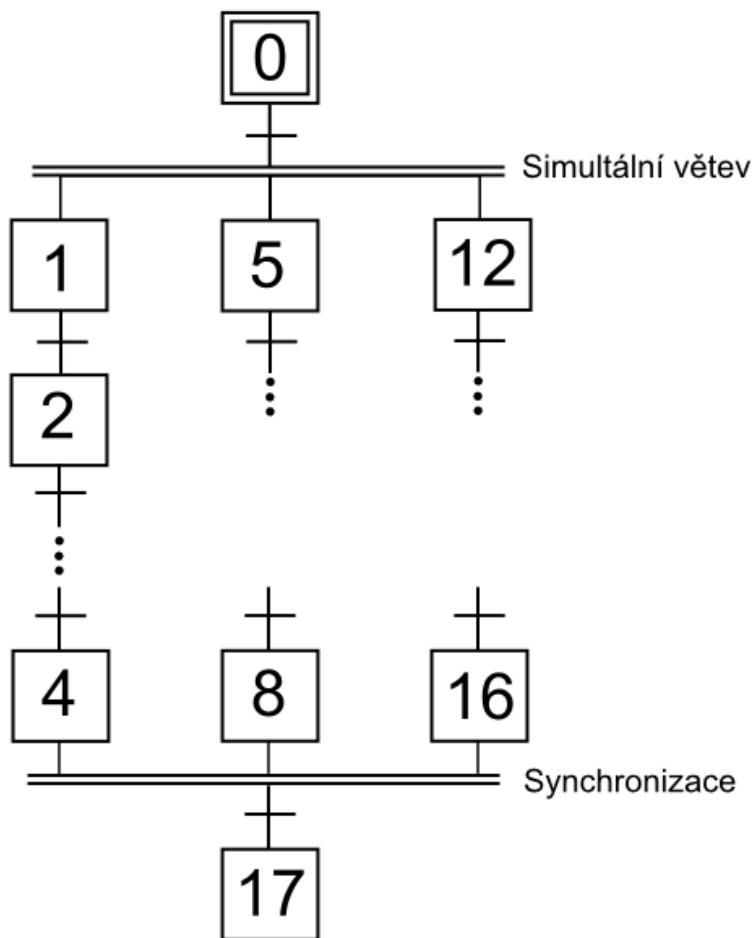
Model automatu Mooreova typu je proto vhodný pro popis chování většiny technologických procesů. Zvláštním případem automatu Mooreova typu je krokový řadič, jehož přechodový diagram má tvar lineárního řetězce stavů nebo jednoduché uzavřené smyčky - neobsahuje žádné větvení.

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 jeden ze stavů. Při větvení grafu „výlučným větvením“ (například Obr.: 1.13) 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 1.12: Ukázka jazyka SFC

Prostředek SFC ale nabízí další možnost, kterou se graf přibližuje vlastnostem jedné třídy Petriho sítí - je jím „simultánní větvení“ (Obr.: 1.14) 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 dvojitě „sběrnice“ a současně je splněna podmínka přechodu - s přechodem se počká na ukončení poslední větve grafu.



Obrázek 1.13: 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.

## 1.6.1 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 Francaise pour la Cybernetique Economique et Technique" a v roce 1987 se stal mezinárodním standardem ( International Electrotechnical Commission 848 - Preparation of function charts for control systems). [12]

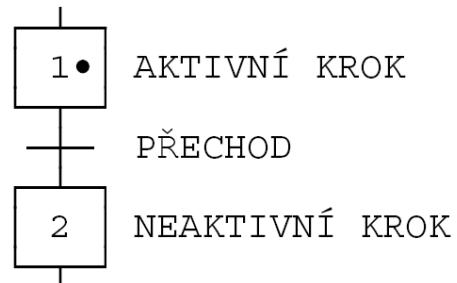
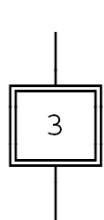
Jazyk Grafcet a SFC jsou velmi podobné, víceméně totožné. Jazyk SFC se vyvinul později, kdy převzal další z kompromisu.

## 1.6.2 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.: 1.15a).

**Přechod** je značen krátkou vodorovnou úsečkou (Obr.: 1.15b). Přechod spolu s krokem tvoří dvojici vrcholů bipartitního orientovaného grafu (hrana jde vždy z kroku do přechodu nebo z přechodu do kroku).

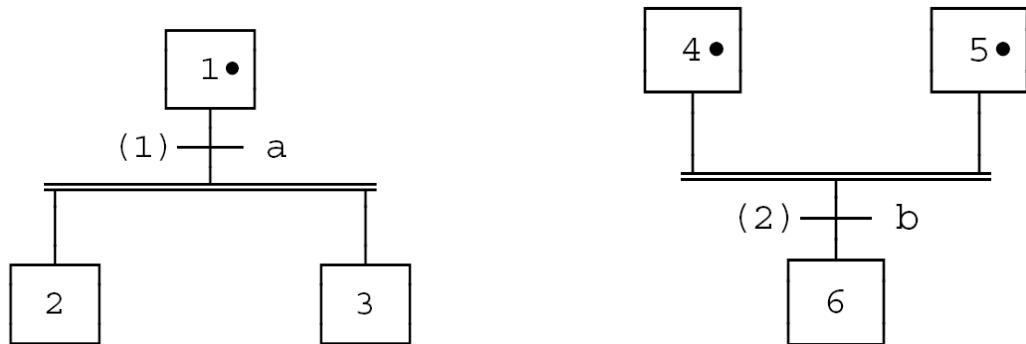


Obrázek 1.14:

a) Počáteční krok

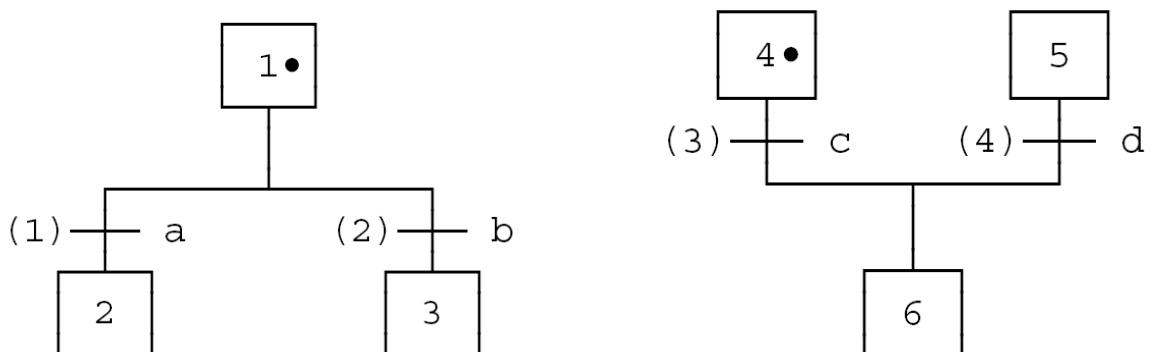
b) Přechod

Každý přechod může mít několik 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.: 1.16a) a konvergentní AND před a po přeskoku přechodu (Obr.: 1.16b)



Obrázek 1.15: 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 1.16: a) Divergentní OR b) Konvergentní OR

# Kapitola 2

## 2. ANALÝZA TYPICKÝCH FUNKCÍ

### 2.1 Analýza z pohledu zabezpečení

Budova jako taková musí být chráněna proti případným zlodějům, proto je potřeba dům zabezpečit proti krádeži. Inteligentní dům umožnuje zamknout (zabezpečit) jednotlivé sekce. Alarm, ale nemusí být jen hlasitý. Uvedu-li příklad, kdy do domu přijde opravář, který má opravit odpad v kuchyni, odblokuji mu jen jednotlivé místnosti a pokud navštíví například ložnici alarm mě upozorní pomocí SMS.

Nedílnou součástí zabezpečení je dnes monitorování uvnitř budovy a okolí pomocí kamer. Pokud kamery zachytí pohyb, informuje uživatele a ten se může pomocí internetu podívat co se v jeho domě děje. Nezbytné je také ukládání videozáznamů na pevný disk, kdy je poté možné materiál použít proti lupičům. Součástí zabezpečení jsou také čidla, detekující rozbití či otevření okna. Násilné vniknutí do garáže nebo otevření brány za naší nepřítomnosti. Detektory požáru (čidla CO<sub>2</sub>, teploty) a čidla úniku plynu či vodní hladiny.

Možná překvapivě bych se zde zmínil o zabezpečení domu v naší nepřítomnosti pomocí takzvané simulace života. Kdy systém simuluje chod domácnosti – zatahuje rolety, zapíná a vypíná světla a spotřebiče, pouští muziku (nahrané rozhovory členů domácnosti), také promítání osob pomocí projektoru. Bezpečnostní systém může identifikovat obyvatele pomocí bezdotykové vstupní karty nebo čtečky otisků prstů.

### 2.2 Analýza z pohledu diagnostiky

Diagnostiku domu můžeme rozdělit na vnitřní a venkovní. Uvnitř budovy můžeme diagnostikovat vzduchu (teplota, relativní vlhkost), osvětlení a akustiku. V případě vzduchu lze pomocí měření relativní vlhkosti a jejího dlouhodobého zvýšení zjistit vznik plísní. Měření se provádí pomocí termohydrografo. Realizace osvětlení pro domácností je relativně jednodušší oproti návrhu a požadavkům osvětlení v pracovním prostředí. Měření se provádí

kalibrovanými fotoelektrickými luxmetry a jasoměry. Zaměřil jsem se na diagnostiku základního osvětlení, jako je ovládání všech světel jedním tlačítek, synchronizace s alarmem a dalšími prvky „chytré“ elektroinstalace.

Pro venkovní diagnostiku můžeme využít například meteostanici, která měří sílu větru, teplotu, srážky a relativní vlhkost vzduchu. Díky informaci o síle větru může dům automaticky zatáhnout markýzy nebo okna. Všechna ovládání můžeme realizovat pomocí SMS, nebo dnes již spíše pomocí internetu v mobilních zařízeních.

## 2.3 Analýza z pohledu řízení

Řízení domu, jsem nakousl v předchozích dvou bodech. Jen lehce shrnu, například při alarmu dostanu automaticky SMS, možnost nastavení všech elektrických spotřebičů podle jednotlivých uživatelů, automatická reakce na změnu povětrnostních podmínek, teploty atd. Sdružování funkcí do takzvaných scén, kdy je provedeno několik kroků (např.: zatažení rolet a rozsvícení světel). Automatická regulace teploty, natočení markýz umožní pohodlně a efektivně využít všechny zdroje tepla. Tímto způsobem je možné ušetřit až třetinu nákladů, jelikož využijeme energie mnohem efektivněji.

# Kapitola 3

## 3. REALIZACE FUNKCÍ A FUNKČNÍCH BLOKŮ

Funkce a funkčních bloků jsem vytvořil ve vývojovém prostředí Mosaic v jazyce LD (Ladder Diagram) a jazyce ST (Structured Text). Některé podstatné funkce a funkční bloky uvedu v této kapitole, další funkce budou uvedeny v příloze.

### 3.1 Realizace funkcí

#### 3.1.1 Funkce 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 opakováního stisku tlačítka.

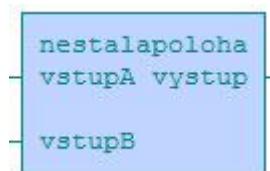
##### **3.1.1.1. Tlačítka s nestálými polohami**

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 „dvojitěho 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 zmínované 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 rozsvícení a zhasnání tří různých světidel v místnosti (lampa u křesla, osvětlení za televizorem a intenzivní světlo z lustru) a poslední tlačítko slouží k ovládání celé skupiny světidel naráz (stiskem do horní polohy se všechna svítidla aktivují, stiskem do dolní polohy se všechna vypnou).

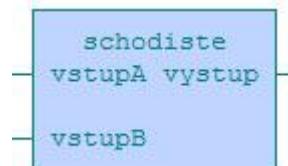
Znovu využijeme výstupní cívky s pamětí Set a Reset. Podobným postupem lze realizovat například centrální zhasnutí a vypnutí spotřebičů při odchodu z domu nebo naopak centrální zapnutí orientačního osvětlení při návratu.

### 3.1.2 Schodiště a chodby

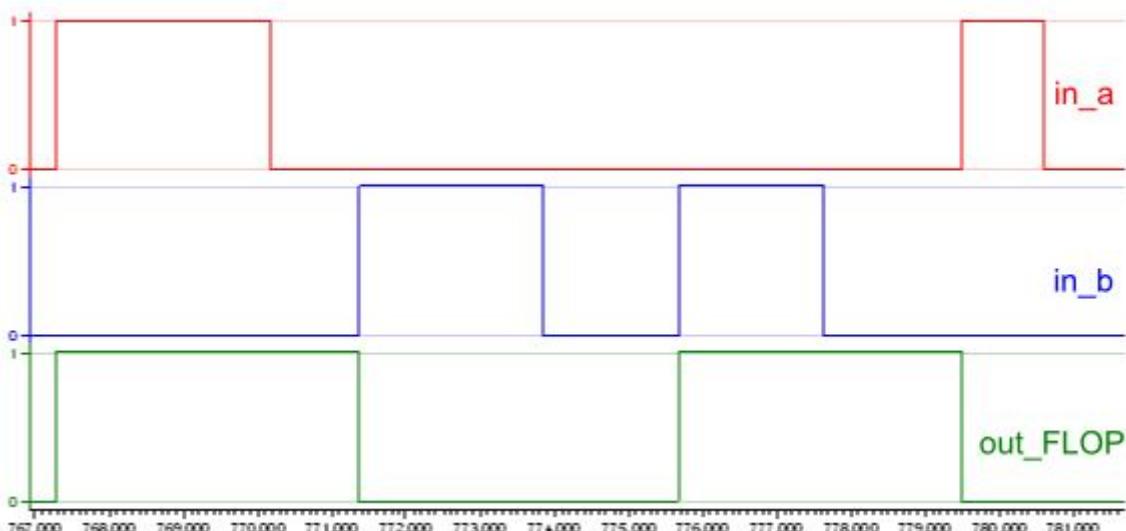
#### 3.1.2.1. T\_FLOP - Ovládání schodiště jedním tlačítkem (reverzibilní)

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 rozsvítí a na druhém konci zhasnu, třetím tlačítkem v mezipatře znova rozsvítí. 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 3.2: Funkční blok T\_FLOP



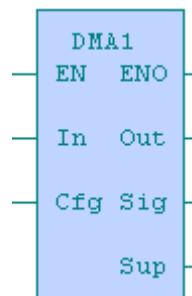
Obrázek 3.3: Průběh FB T\_FLOP

### 3.1.2.2. 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 kinosálech na začátku a konci promítání.

K tomuto bloku je široké spektrum možností obsluhy a požadavků obyvatel, proto uvede příklad, který by ocenilo i moje okolí. Světlo by má tři prahy intenzity světla. Nejméně intenzivní ke sledování televize, středně intenzivní pro četbu knihy a maximálně intenzivní pro osvětlení celého obytného prostoru. Variantou je také plynulé ovládání, které ocení nejspíše většina uživatelů, proto jej uvádím níže.

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 ( $<0.5\text{s}$ ) nabíhá na požadovanou úroveň jasu, dalším krátkým stiskem výstup vypíná. Dlouhým stiskem tlačítka ( $>0.5\text{s}$ ) dochází k plynulé regulaci jasu v rozsahu hodnot  $\text{MinOut} <-> \text{MaxOut}$ . 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. Rychlosť náběhu výstupu na požadovaný jas při zapnutí je dána parametrem  $Tup$ , rychlosť sestupu při vypnutí je dána parametrem  $Tdw$ . [7]



Obrázek 3.4: Vzhled FB DMA1

### 3.1.2.3. Ovládání osvětlení s časovou funkcí

Návrh 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.

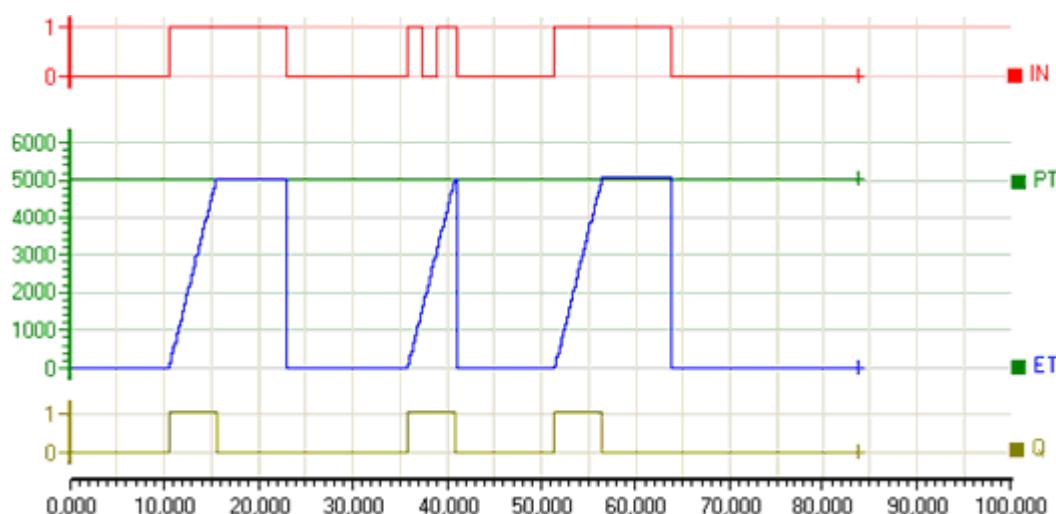
#### 3.1.2.4. Čidlo pohybu

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

Jako příklad vhodný pro využití čidla PIR je osvětlení WC, bohužel zde vzniká problém, kdy jste delší dobu v klidu a světlo zhasne. Proto bych raději volil variantu kde PIR čidlo je v kombinaci s dalším senzorem jako je světelná závora. Všeobecně pro veřejné záchody je vhodné využívat bezdotykové varianty všeho. Příkladem je samosplachovací toaleta, bezdotyková baterie...

#### 3.1.2.5. Vyvětrání místnosti

Požadavkem je, aby se po stisku tlačítka (nebo signálem z čidla PIR) spustil ventilátor a běžel nastavenou dobu. Tlačítko může současně sloužit i k rozsvícení nebo ke zhasnutí svítidla - podle toho, přejeme-li si aby ventilátor běžel současně se světlem (v naší přítomnosti) nebo až od okamžiku zhasnutí 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.

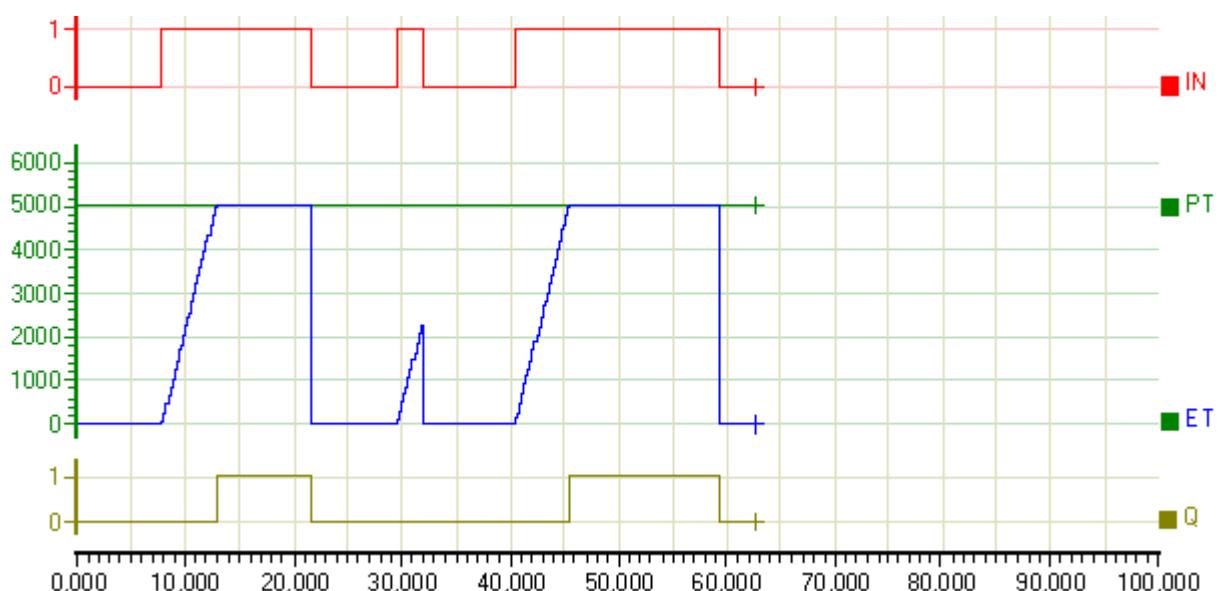


Obrázek 3.5: Průběh časovače TP

### 3.1.2.6. 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 3.6: Průběh časovače TON

## 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 jednoho ze senzorů nebo vzniku 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.

Nastavení alarmu definujeme na 3 úrovně:

1. senzor rozbití skla dává signál – muže to být zloděj, ale i míč kopnutý od sousedova kluka, proto nevolat bezpečnostní 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 bloků, zde uvedu dva, které jsou pro praktické využití nevhodnější a zbytek lze nalézt v knihovně.

### 3.2.1 Majorita

#### 3.2.1.1. Majorita ze 3

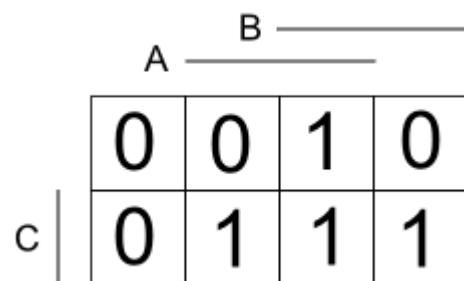
Funkce majority (většiny) se může definovat i pro jiný lichý počet operandů než tři a je pravdivá pro nadpoloviční počet pravdivých operandů. Majorita ze tří je funkce, která je pravdivá pokud nadpoloviční většina (dva nebo tři) z jejich operandů je pravdivých. Je obdobou hlasování tří rovnoprávných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

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

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

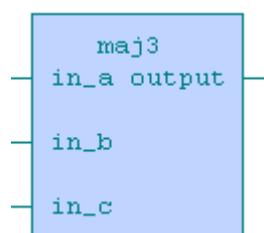
Tabulka 3.1: Popis funkce třívstupového majoritního členu



Obrázek 3.7: 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:

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



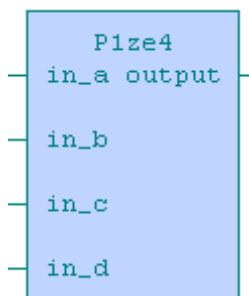
Obrázek 3.8: FB majorita ze 3

### 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*.

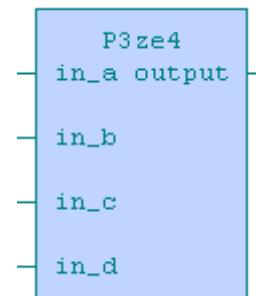
Obecně *pk\_n*, kde první číslo představuje hodnotu prahu a druhé je počtem operandů. Lze je definovat pro libovolný počet operandů a pro libovolné hodnoty prahu. Obecně platí, že jsou pravdivé, pokud počet pravdivých operandů je alespoň roven (roven nebo větší) prahu (prvnímu číslu z názvu funkce). 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 3.9: a) Prahová funkce 1 ze 4

**Prahová funkce 2 ze 4 (P2ze4)**



b) Prahová funkce 3 ze 4

### 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, jsou to všechny funkce popsané výše. Stručně je budeme označovat jako *S1ze3*, *S2ze3*, *S3ze3*, obecně *S k ze n*. Jsou pravdivé pokud počet pravdivých operandů je právě roven charakteristickému číslu - prvnímu číslu z označení funkce.

### 3.3 Ovládání mechanismů

V budovách je mnoho ovládacích mechanismů jako je automatické otevíraní oken, pohyb markýz či rolet, otevření a zavírání garážových a vjezdových vrat lze je definovat dvěma stabilními stavami. 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 nabíhat další. Jako je například mikro ventilace u oken, nebo náklon listů rolet v několika stupních dle požadavků zákazníka pro jeho maximální komfort.

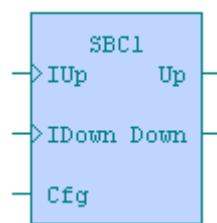
Narázíme zde na takzvané mezní polohy, kdy je potřeba kontroly chyb. Například zda-li se nám vstupní brána zavřela. To lze vyřešit jakoukoli časovou funkcí, kdy pro jednotlivé systémy (přístroje) definujeme čas, za který musí být brána v koncové poloze (zavřeno či otevřeno). Pokud tak nenastane, muže 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 3.10: Vzhled FB SBC1

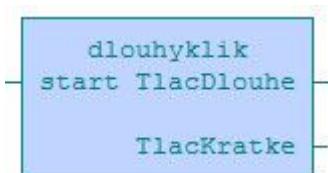
Tento funkční blok lze nalézt v knihovně BuildingLib pod názvem SBC1. Mě jako uživateli na tomto funkčním bloku chybělo několik věcí, které lze jistě široce využít při dalších konfiguracích pohodlného bydlení. Zde jen lehce naznačím, co bych ještě ocenil a v kapitole 4.1.3 uvedu podrobnosti a řešení. 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 vzloupáním nebo poruchou senzoru.

# Kapitola 4

## 4. NÁVRH SPECIFICKÝCH BLOKŮ

### 4.1 Rozpoznání dlouhého a krátkého stisku

Funkční blok slouží k rozpoznání dlouhého a krátkého stisku tlačítka. Po stisku tlačítka je aktivován čítač TON. Pokud čítač načítá jen velmi krátkou dobu ( $<5s$ ) a nepřijde sestupná hrana od vstupu zůstane vnitřní proměnná v 0 a aktivuje se výstup TlacDlouhe. Pokud, ale přijde sestupná hrana než čítač načítá dobu větší než pět vteřin, pak se vnitřní proměnná překlopí do 1 a zaktivuje výstup TlacKratke. Tento blok najde mnoho uplatnění, atž už v oblasti osvětlení, ovládání mechanických částí nebo zabezpečovacího systému.



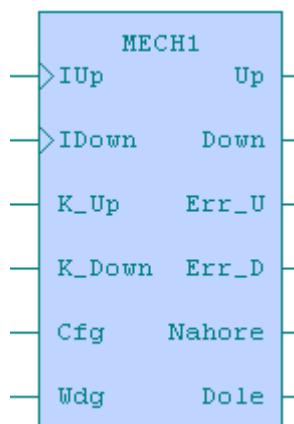
Obrázek 4.1: Vzhled bloku rozpoznání stisku tlačítka



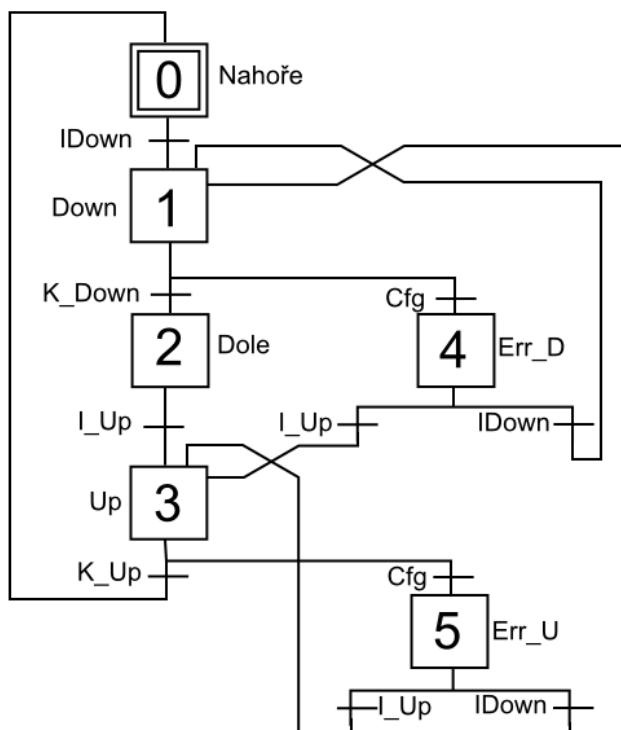
Obrázek 4.2: Ukázka LD programu rozpoznání stisku tlačítka

## 4.2 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 přednastavít.



Obrázek 4.3: Vzhled bloku mechanického ovládání



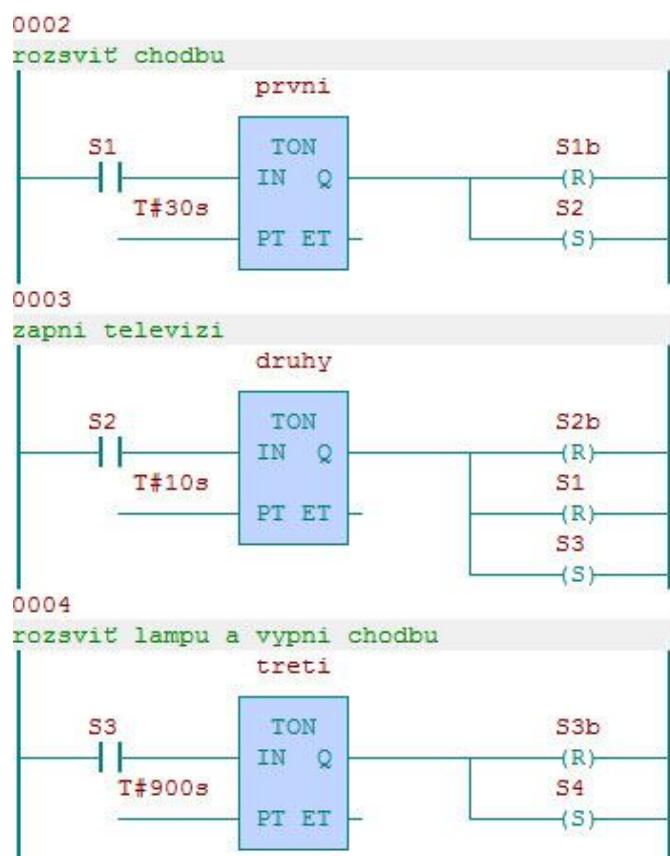
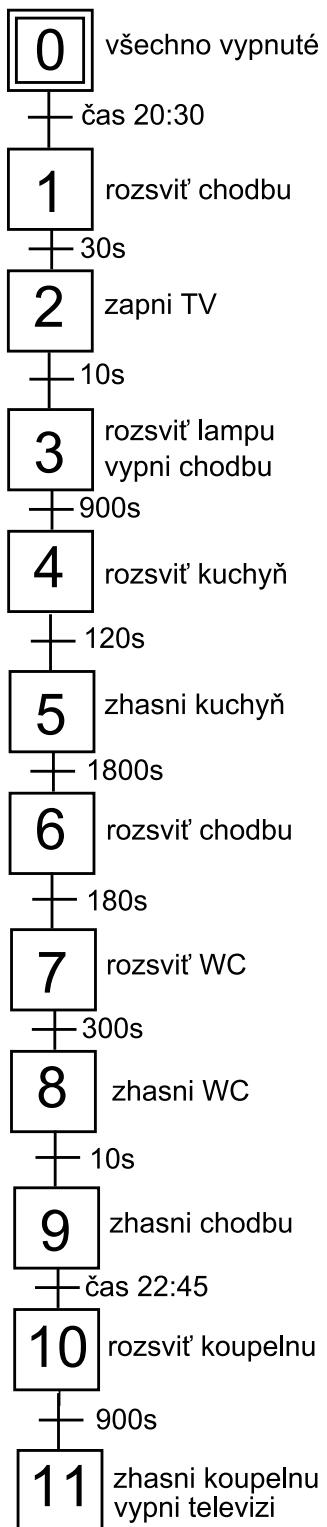
Obrázek 4.4: Diagram mechanického ovládání SFC

## 4.2.1 Synergie domu

Synergie domu (týž. spolupráce) v literatuře nazývané také scény jsou prvky inteligentního domu, které spolupracují navzájem, předávají si informace a diagnostikují okolí. Pro uživatele velmi zpříjemní ovládání celého domu, jelikož jeden jeho povl vyvolá hned několik akcí. Můžu uvést příklad režimu 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. A to vše pouze jedním tlačítkem, které je například vedle vašeho nočního stolku. Scén může být velmi mnoho, spektrum využití je velmi individuální. Dále uvedu příklady jako je režim romantický večer, návštěva, večírek nebo režim dovolená.

### 4.2.1.1. Simulace života v domě

Režim simulace života je a bude velmi využívaná scéna. Výhodný nese právě, když nejste přes víkend doma nebo trávíte čas na dovolené. Smyslem scény je simulace přítomnosti osob, která má působit na potencionální zloděje jako obydený dům (ve skutečnosti prázdný). Systém automaticky každý večer rozsvětí a zhasíná světla, pouští televizi nebo rádio. Večer zatahuje rolety a ráno je zase vytahuje. Toto se může každý den opakovat stále stejně. Pro intelligentnější řešení je možné nastavit tzv. pseudonáhodné kombinace různých přístrojů, která by měla případné zloděje odradit.



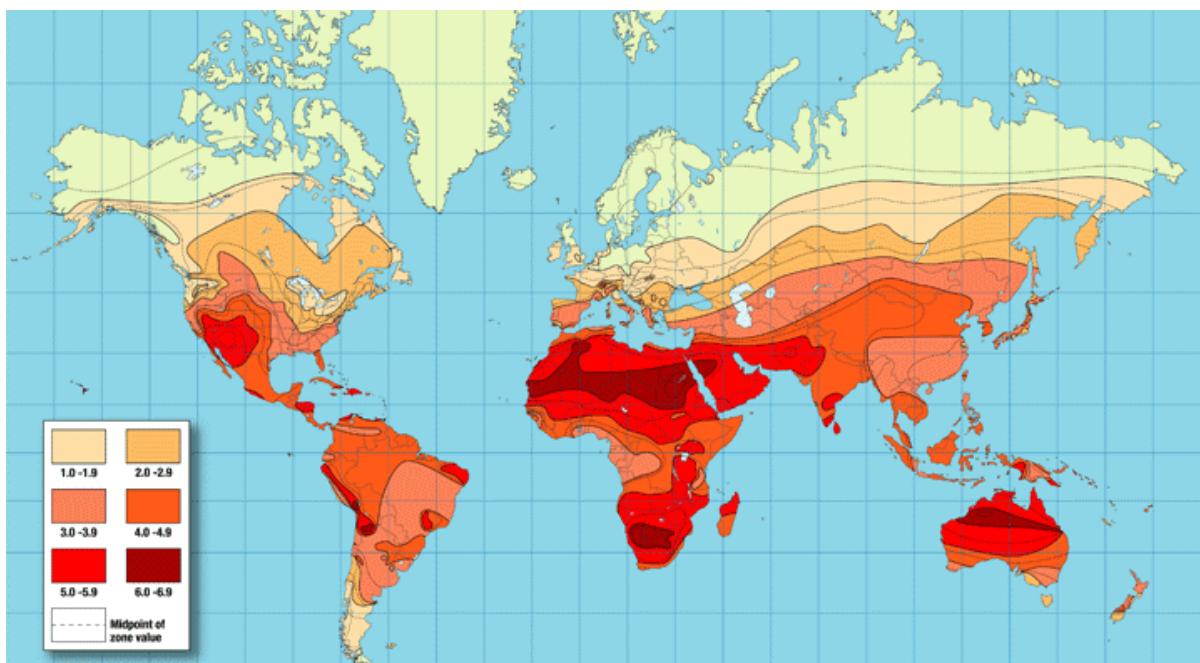
Obrázek 4.5: Ukázka simulace života v domě

## 4.3 Astropogram

Myšlenkou tohoto bloku je pomocí vstupních dat, jako je zeměpisná šířka a výška určit polohu slunce a jeho pohyb. Tyto data, lze poté využít například pro natočení solárních panelů, žaluzií a markýz. Díky tomuto, lze ušetřit mnoho financí za vytápění. Další z možností využití je ovládání veřejného osvětlení za přispění dalších senzorů a mnoho dalších.

### 4.3.1 Množství energie dopadající na zemský povrch

Energie, která dopadá na zemský povrch a lze jí využít je ovlivněna mnoha faktory. Jedním z nich je geografická poloha (viz. Obr 4.16), roční dobou, oblačností, sklonem a orientací solárního panelu. Nelze tedy určit přesně kolik energie dopadne na nějaké místo v určitý den a hodinu, toto lze ale určit pro větší časový úsek.



Obrázek 4.6: Světová mapa solárního záření

Nejprve začneme s měrným tokem, který dopadá na povrch atmosféry zvaným solární konstanta ( $I_0 = 1\ 360\ Wm^{-2}$ ), který je zmenšován průchodem ionosférou (atmosférické plyny), ozónosférou (UV záření) v těchto vrstvách zůstávají životu nebezpečné složky slunečního záření. Na energetickou hodnotu však nemají vliv. Vrstva následující asi 19% záření absorbuje a 34% odrazí zpět do meziplanetárního prostoru. Velikost součinitele znečištění, který lze podle vzorce 4.1 spočítat nebudeme pro naše účely využívat, ale pro

potřeby návrhu solárního zařízení zde uvedu. Velikost součinitele znečištění atmosféry se během roku v daném místě mění. Obecně lze říci, že s nadmořskou výškou hodnota součinitele znečištění klesá, naopak s nárůstem teploty zase stoupá (pro ČR 1,5 až 6).

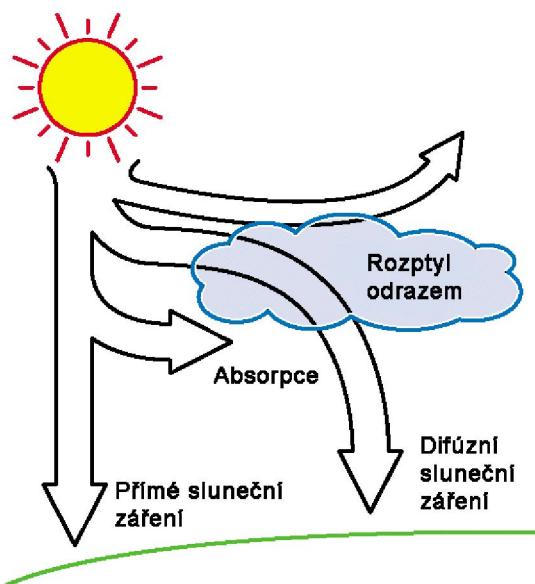
$$Z = \frac{\ln I_0 - \ln I_n}{\ln I_0 - \ln I_c}$$

$I_0$  je sluneční konstanta [ $\text{W} \cdot \text{m}^{-2}$ ]

$I_n$  intenzita záření dopadající na kolmou plochu vhledem ke slunečním paprskům při daném znečištění ovzduší [ $\text{W} \cdot \text{m}^{-2}$ ]

$I_c$  intenzita záření dopadající na kolmou plochu vhledem ke slunečním paprskům při dokonale čistém ovzduší [ $\text{W} \cdot \text{m}^{-2}$ ].

Pro určení množství energie je zapotřebí znát intenzitu záření. Je to základní veličina pro všechny výpočty týkající se dopadající energického účinku ze Slunce na povrch Země.

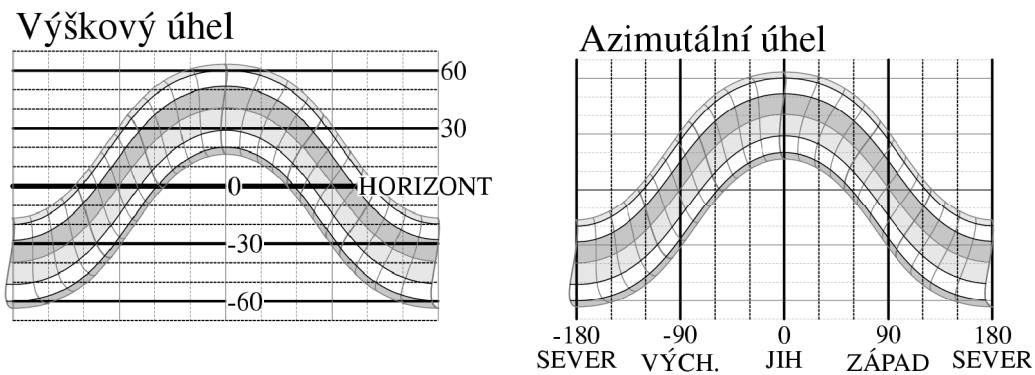


Obrázek 4.7: Světová mapa solárního záření

#### 4.3.1.1. Jak se určí poloha Slunce

Řekneme-li, že sluneční paprsky dopadají pod úhlem  $g$ , ten je dán vzájemnou polohou Slunce nad obzorem a plochou kam dopadají paprsky. Poloha slunce se samozřejmě mění v závislosti na čase, proto ji lze v každém okamžiku popsat výškou nad obzorem  $h$  a azimutem  $a$ . 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.

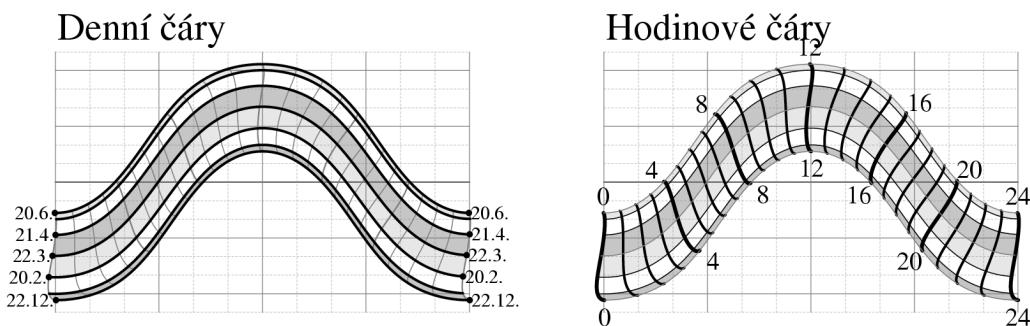


Obrázek 4.8: a) Výškový úhel

b) Azimutální úhel

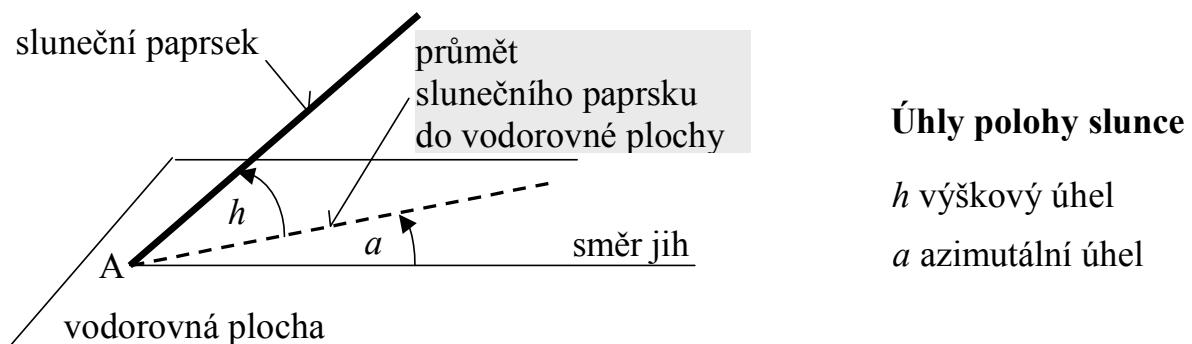
**Denní čára.** Množině poloh, kterými slunce prochází během celého kalendářního dne, odpovídá v grafu souvislá čára, denní čára chodu slunce. Pro různé dny v roce dostáváme různé denní čáry.

**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 4.9: a) Denní čáry

b) Hodinové čáry



$$\sin h = \sin \delta \sin \varphi + \cos \delta \cos \varphi \cos \tau \rightarrow h = \arcsin(\sin \delta \sin \varphi + \cos \delta \cos \varphi \cos \tau) \quad [^\circ]$$

$$\sin a = \frac{\cos \delta}{\cos h} \sin \tau \rightarrow a = \arcsin(\sin \tau) \quad [^\circ]$$

kde  $\delta$  je sluneční deklinace ,

$\varphi$  zeměpisná šířka (Praha přibližně  $50^\circ$ ),

$T$  časový úhel v obloukových stupních měřený od 12h, jedna hodina je  $15^\circ$  .

Azimut  $a$  je zde počítán od jihu ve smyslu otáčení hodinových ručiček jako kladná hodnota. Deklinace  $\delta$  je zeměpisná šířka, kde v daný den ve 12h je Slunce kolmo nad obzorem.

Vypočítá se ze vztahu:

$$\delta = 23,45^\circ \sin(0,98^\circ D + 29,7^\circ M - 109^\circ) \quad [^\circ]$$

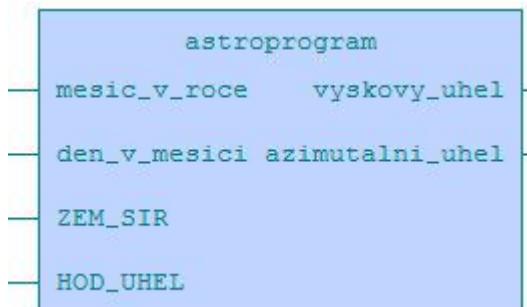
nebo

$$\delta = 23,45 \sin\left(360 \frac{284+n}{365}\right) \quad [^\circ]$$

kde  $D$  je den v měsíci,  $M$  pořadí měsíce v roce,  $n$  je pořadí dne v roce

Deklinace Slunce  $\delta$  udává úhel, který svírá sluneční paprsek směrující ke středu Země (Slunce je v zenithu) s rovinou rovníku. Tento úhel nabývá hodnot mezi  $-23,5^\circ$  (nejčastěji 21. prosince) a  $+23,5^\circ$  (nejčastěji 21. června). Právě díky tomuto sklonu zemské osy má Slunce 21. června maximální deklinaci (Slunce dosáhlo obratníku Raka) a od tohoto dne se vrací zpět k rovníku. Nejčastěji se využívá výpočet deklinací pro celý měsíc. Počítá se s deklinací pro charakteristický den v měsíci, pro který se počítají všechny parametry určující polohu Slunce včetně intenzity solárního záření. Takto vypočítaná intenzita je považována za průměrnou hodnotu dosaženou v celém daném měsíci.

Naším cílem je získat polohu slunce, tedy výškový úhel a azimutální úhel (v kartézských souřadnicích). Tedy na vstupu budeme v nejjednodušším případě zadávat den v roce. Pokud bychom chtěli určovat polohu jinde než v Praze, museli bychom zadávat i zeměpisnou polohu (v našem případě  $\varphi$ ).

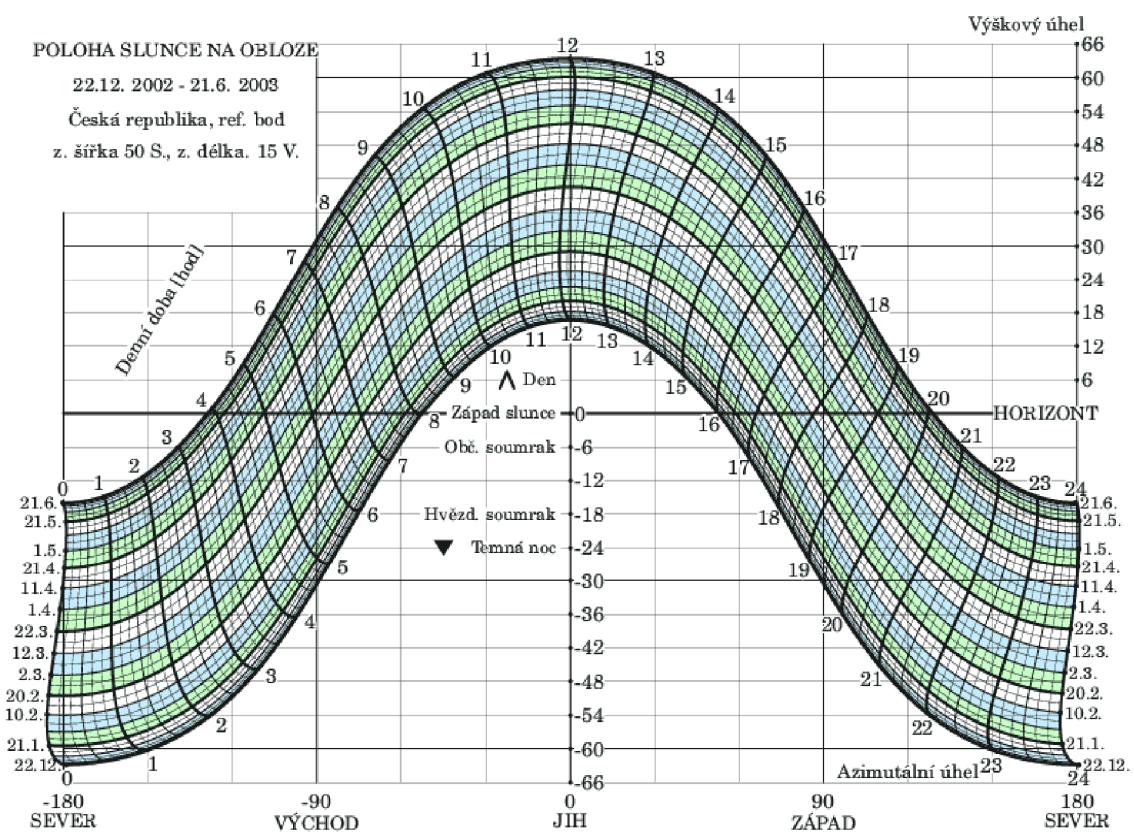


Obrázek 4.11: Vzhled bloku astroprogram

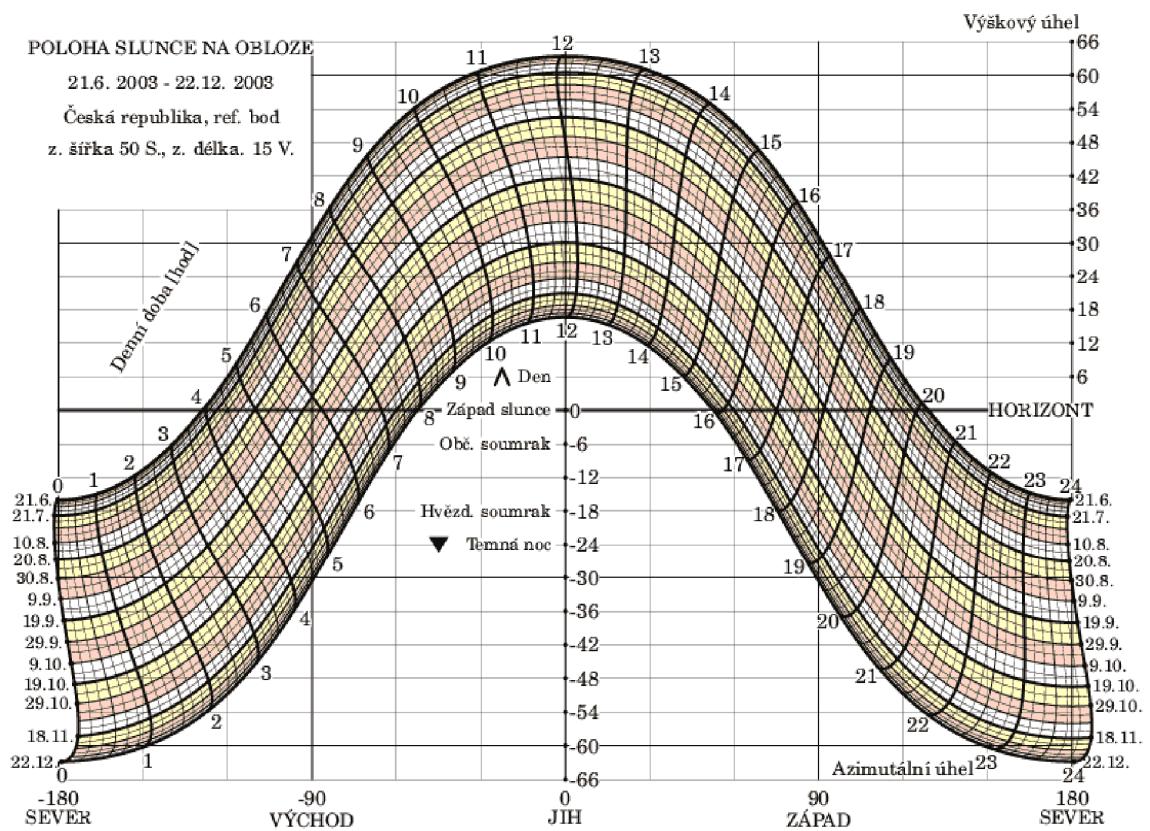
#### 4.3.1.2. Stanovení zeměpisné polohy místa

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  $15^{\circ}$  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 nebo pomocí GPS. [14]

#### 4.3.1.3. Dvojice základních grafů pro ČR a jejich využití



Obrázek 4.12: Poloha slunce na obloze od zima-jaro



Obrázek 4.13: Poloha slunce na obloze léto-podzim

# Kapitola 5

## 5. ZÁVĚR

V této práci jsem se zabýval tématem inteligentních budov, návrhem a tvorbou funkčních bloků. Téma inteligentních budov je nové téma, které se každým dnem vyvíjí. Até už jsou to nové okna, zdivo či solární panely. Cílem práce bylo navrhnout a realizovat funkce či funkční bloky, které lze do konkretního návrhu jednoduše zakomponovat pro budoucího uživatele. Také navrhnout pokročilejší řešení jako je synergie domu, simulace života v domě nebo astroprogram.

V první kapitole jsou základní informace o inteligentních domech, programovatelných automatech a mezinárodní normě IEC EN 61 131-3. Ve druhé kapitole jsem analyzoval zajímavé funkce z pohledu zabezpečení, diagnostiky a řízení. Třetí kapitolu jsem věnoval realizaci funkcí a funkčních bloků, jejich popisu či praktickému využití, popřípadě jejich kombinaci s dalšími funkcemi.

Ve čtvrté kapitole jsem posal jednotlivé funkce či funkční bloky pomocí jazyka SFC, LD, ST v kombinaci se slovním popisem. Jednu z podkapitol jsem věnoval popisu Astroprogramu, který může být v praxi mnoho využití. Poslední kapitola popisuje manuál knihovny, kterou jsem připravil za účelem praktické využití. Zde v bakalářské práci neuvádím všechny zdrojové kódy, které jsou k dispozici v přiloženém CD.

# 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](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](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
- [15] KMENT, J., *Bakalářská práce: Standardizace funkcí pro řízení inteligentních domů a návrh knihovny funkčních bloků*, Praha 2009
- [16] SETRNEBUDOVY [online], <http://www.setrnebudovy.cz/>

# **Přílohy**

**A – Manuál funkcí a funkčních bloků**

**B – CD s elektronickými dokumenty a knihovnou  
funkcí a funkčních bloků**

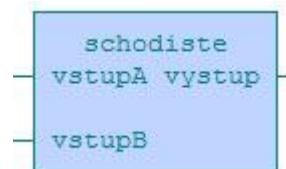
# **Manuál funkcí a funkčních bloků**

## **Obsah:**

1.	Tlačítko s nestálými polohami.....	I
2.	Ovládání schodiště jedním tlačítkem.....	II
3.	Tlačítko rozpoznání kliku .....	III
4.	Astroprogram.....	IV
5.	Vyvětrání místnosti .....	V
6.	Majorita ze tří .....	VI
7.	Prahové funkce .....	VII

# 1. Tlačítko s nestálými polohami

Funkční blok slouží k ovládání osvětlení. Nebo popřípadě k dalšímu využití. Blok má dvě vstupy a jeden výstup. Nejčastější využití nalezne pro schodišťe. Kliknutím do horní části (polohy) tlačítka aktivuji (rozsvítím), kliknutím do dolní polohy deaktivuji (zhasnu).



**Obr. 1** - Funkční blok nestálá poloha

Vstupem bloku je signál z osmi poloh tlačítka:

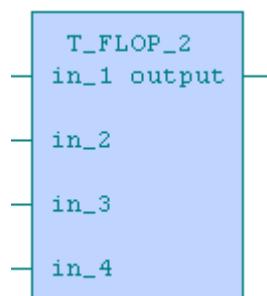
Tlačítko 1 má dvě polohy:      - horní (vstupA)  
    - dolní (vstupB)

Výstupem je signál vyslaný příslušnému svítidlu:

Výstup pro lampu:      vystup

## 2. Ovládání schodiště jedním tlačítkem

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

Vstupem bloku je signál ze čtyř tlačítek:

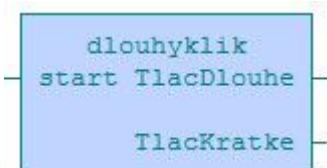
- |             |                |
|-------------|----------------|
| Tlačítko 1: | - horní (in_1) |
| Tlačítko 2: | - horní (in_2) |
| Tlačítko 3: | - horní (in_3) |
| Tlačítko 4: | - horní (in_4) |

Výstupem je signál vyslaný příslušnému svítidlu:

Výstup osvětlení:              output

### 3. Tlačítko rozpoznání délky kliku

Funkční blok slouží k rozpoznání dlouhého a krátkého stisku tlačítka. Po stisku tlačítka je aktivován čítač TON. Pokud čítač načítá jen velmi krátkou dobu ( $<5s$ ) a nepřijde sestupná hrana od vstupu zůstane vnitřní proměnná v 0 a aktivuje se výstup TlacDlouhe. Pokud, ale příjde sestupná hrana než čítač načítá dobu větší než pět vteřin, pak se vnitřní proměnná překlopí do 1 a zaktivuje výstup TlacKratke. Tento blok najde mnoho uplatnění, atž už v oblasti osvětlení, ovládání mechanických částí nebo zabezpečovacího systému.



Obr. 3 - Funkční blok rozpoznání kliku

Vstupem bloku je signál z tlačítka:

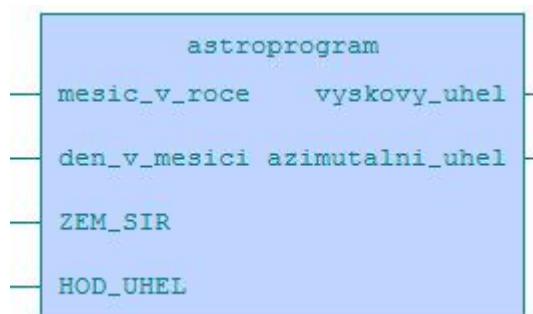
Tlačítko : - start

Výstupem je signál vyslaný příslušnému svítidlu:

Výstup 1 : - TlacDlouhe  
Výstup 2: - TlacKratke

## 4. Astroprogram

Tento blok pomocí vstupních dat, jako je zeměpisná šířka, číslo měsíce v roce, den v tomto měsíci a časový úhel v obloukových stupních měřený od 12h, jedna hodina je  $15^\circ$ . Tyto data, lze poté využít například pro natočení solárních panelů, žaluzií a markýz. Další z možností využití je ovládání veřejného osvětlení za přispění dalších senzorů.



Obr. 4 - Funkční blok Astroprogramu

Vstupem bloku je signál z tlačítek:

- Tlačítko 1: - mesic\_v\_roce
- Tlačítko 2: - den\_v\_mesici
- Tlačítko 3: - ZEM\_SIR
- Tlačítko 4: - HOD\_UHEL

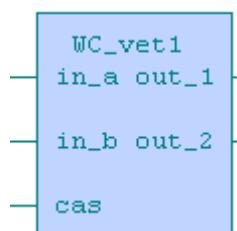
Výstupem je signál vyslaný příslušnému svítidlu:

- Výstup 1: - vyskovy\_uhel
- Výstup 2: - azimutalni\_uhel

## 5. Vyhýtrání místnosti

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. 5** - Funkční blok WC\_vet1

Vstupem bloku je signál z tlačítka:

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

Výstupem je signál vyslaný svítidlu a ventilátoru:

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

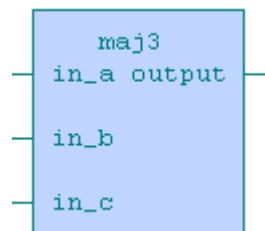
## 6. Majorita ze tří

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:

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



**Obr. 6** - Funkční blok maj3

Vstupem bloku je signál ze tří senzorů:

Senzor rozbití skla:      in\_a

Čidlo pohybu:            in\_b

Dveřní kontakt:          in\_c

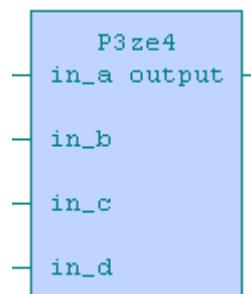
Výstupem je signál alarmu:

Výstup alarm:            output

## 7. 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. 7 - Funkční blok P3ze4**

Vstupem bloku je signál ze senzorů:

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

Výstupem je signál alarmu:

Výstup alarm:                  output