České vysoké učení technické v Praze
Fakulta strojní
Ústav přístrojové a řídicí techniky

Návrh a implementace systému zabezpečení prostoru částicového urychlovače

Diplomová práce

Bc. Martin Libenský

Magisterský program: Strojní inženýrství
Studijní obor: Přístrojová a řídicí technika
Vedoucí diplomové práce: Mgr. Ing. Jakub Jura, Ph.D.

Praha, Květen 2019

Vedoucí diplomové práce:

Mgr. Ing. Jakub Jura, Ph.D.
Ústav přístrojové a řídicí techniky
Fakulta strojní
České vysoké učení technické v Praze
Technická 1902/4
166 07 Praha 6
Česká Republika

Copyright © Květen 2019 Bc. Martin Libenský

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně s tím, že její výsledky mohou být dále použity podle uvážení vedoucího diplomové práce jako jejího spoluautora. Souhlasím také s případnou publikací výsledků diplomové práce nebo její podstatné části, pokud budu uveden jako její spoluautor.

Datum: ....

podpis

ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Libenský** Jméno: **Martin** Osobní číslo: **419964**

Fakulta/ústav: **Fakulta strojní**

Zadávající katedra/ústav: **Ústav přístrojové a řídící techniky**

Studijní program: **Strojní inženýrství**

Studijní obor: **Přístrojová a řídicí technika**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Návrh a implementace systému zabezpečení prostoru částicového urychlovače

Název diplomové práce anglicky:

Design and implementtion of electron accelerator security system

Pokyny pro vypracování:

1) Sestavte slovní zadání dle požadavků zadavatele a UJB a proveďte jeho analýzu. Výsledky znázorněte v UML.
2) Navrhněte a implementujte řízení hydraulicky ovládaných olověných dveří.
3) Navrhněte a implementujte systém dispečerského řízení včetně komunikace s další řídicí jednotkou urychlovače.
4) Realizujte sběr provozních dat dle požadavků UJB.

Seznam doporučené literatury:

Šimáně, Č., Vognar, M., Chvátil, D. The system for Control and Stabilization of the Beam Position in the Microtron MT25 in Prague, Acta Polytechnica Vol. 44, No. 1/2004, CTU in Prague, 2004, pp. 59-63
Krist, P. Řídicí systém mikrotronu – kruhového relativistického urychlovače elektronů. Praha: Ústav přístrojové a řídicí techniky, ČVUT Praha, 2012. 115 stran. Disertační práce, školitel Bíla, J.
Krist, P., Bíla, J. A mathematical model of the MT25 microtron. JINST 6, T10005, 2011
Firemní manuály výrobce řídicích systémů
B. Vogel-Heuser, D. Witsch a U. Katzke, “Automatic code generation from a UML model to IEC 61131-3 and system configuration tools”, in 2005 International Conference on Control and Automation, sv. 2, červ. 2005, 1034–1039 Vol. 2. doi: 10.1109/ICCA.2005.1528274.

Jméno a pracoviště vedoucí(ho) diplomové práce:

Ing. Mgr. Jakub Jura, Ph.D., U12110.3

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **26.04.2019** Termín odevzdání diplomové práce: **12.06.2019**

Platnost zadání diplomové práce: _____

Ing. Mgr. Jakub Jura, Ph.D. podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	prof. Ing. Michael Valášek, DrSc. podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

. _____ Datum převzetí zadání

_____ Podpis studenta

© ČVUT v Praze, Design: ČVUT v Praze, VIC

Abstrakt

Tato diplomová práce se zabývá návrhem a realizací systému řízení a zabezpečení urychlovače částic Mikrotron MT25. V práci je popsán postup analýzy požadavků na zabezpečení z hlediska ochrany před ionizujícím zářením a související syntéza objektového, dynamického a funkčního modelu. Detailní návrh je rozpracován do úrovně vývoje softwaru pro řídicí a komunikační systémy, který zajišťuje bezpečné provozování urychlovače a měření klíčových dozimetrických hodnot spolu s jejich automatickým ukládáním do provozního deníku. Pro pohodlné ovládání celého systému bylo vytvořeno uživatelské rozhraní (SCADA/HMI) v prostředí Reliance.

Klíčová slova

Mikrotron, ionizující záření, OMT analýza, UML diagramy, PLC, HMI

Abstract

This diploma thesis captures designing and realizing of control and security system of Mikrotron MT25 particle accelerator. The work describes the procedure of security requirements analysis in terms of protection against ionizing radiation and related synthesis of object, dynamic and functional model. The detailed design is elaborated to the level of software development for control and communication systems, which secure safe operating of the particle accelerator and dosimetric measurements together with their automatic saving to operating log. User interface (SCADA / HMI), created for effective control of the system, has been designed in Reliance.

Keywords

Microtron, ionizing radiation, OMT analysis, UML diagrams, PLC, HMI

Poděkování

Tímto bych chtěl poděkovat mému vedoucímu Mgr. Ing. Jakubovi Jurovi, Ph.D. za podporu a vedení mé diplomové práce. Ochotně pomohl vždy, když bylo třeba.

Velké poděkování patří také celému personálu urychlovače hlavně za ochotu a pomoc. Jmenovitě zejména Ing. Pavlovi Kristovi, Ph.D, Ing. Dušanovi Poklopovi a Ing. Václavovi Olšanskému.

Největší poděkování patří celé mé rodině, která mi poskytla prostředí pro práci po celou dobu mého studia.

Seznam použitých značek a symbolů

Zkratka	Význam
IZ	ionizující záření
SUJB	Státní úřad pro jadernou bezpečnost
HMI	human-machine interface
OMT	Object-modeling technique
UML	Unified modeling language
OMSE	Object-Managed Software Engineering
ST	Structured text
DFD	Data flow diagram
SW	software
HW	hardware

Obsah

Seznam použitých značek a symbolů . . . 1
1 Úvod . . . 3
2 Popis systému . . . 4
2.1 Popis systému pro účel analýzy . . . 4
2.2 Mikrotron MT 25 . . . 6
2.3 Popis pracoviště . . . 7
3 Teoretická část . . . 9
3.1 Ionizující záření . . . 9
3.2 Veličiny a jednotky v radiační fyzice . . . 9
3.2.1 Základní principy radiační ochrany . . . 11
3.2.2 Monitorování v radiační ochraně . . . 11
3.2.3 Evidence ozáření . . . 13
3.2.4 Detekce IZ . . . 13
3.2.5 Plynové detektory . . . 14
4 Praktická část . . . 17
4.1 Object-Modeling Technique (OMT) a Unified Modeling Language (UML) 17
4.1.1 Objektový model . . . 19
4.1.2 Dynamický model . . . 30
4.1.3 Funkční model . . . 34
4.2 Detail design . . . 37
4.2.1 Reliance 4 . . . 39
4.2.2 Mosaic . . . 39
4.2.3 Popis programu subsystému PLC . . . 39
4.2.4 HMI . . . 57
4.3 Technická realizace systému . . . 63
4.3.1 Tecomat foxtrot CP-1003 . . . 63
4.3.2 MS 2000 . . . 65
4.3.3 Komunikace . . . 67
4.3.4 Ovládací panel dveří . . . 69
5 Závěr . . . 70
Reference . . . 73
Přílohy . . . 77
A Hlavní program . . . 77
B Funkční blok pro ovládání čerpadla . . . 88
C Funkce pro změnu stavu systému . . . 89

1 Úvod

Práce si klade za cíl vytvořit zabezpečovací a ovládací systém kontrolního pásma elektronového urychlovače Mikrotronu MT 25. Fungování systému bylo určeno požadavky personálu urychlovače a dokumentem požadavků SUJB (Státní úřad pro jadernou bezpečnost), který mi byl poskytnut. Dokument specifikuje požadavky převážně na bezpečnost provozu a záznam klíčových bodů.

Stávající systém není řízen výpočetní technikou, ale kombinačně-sekvenčním logickým elektrickým obvodem. S možnostmi, které se dnes nabízejí v oblasti automatického řízení se tento způsob řízení jeví jako nevhodný a implementace dalších funkcí velice obtížná. Z toho důvodu je vhodné ho nahradit novým ovládacím systémem, který bude přehlednější a bude umožňovat jednodušší implementaci nových funkcí v případě potřeby.

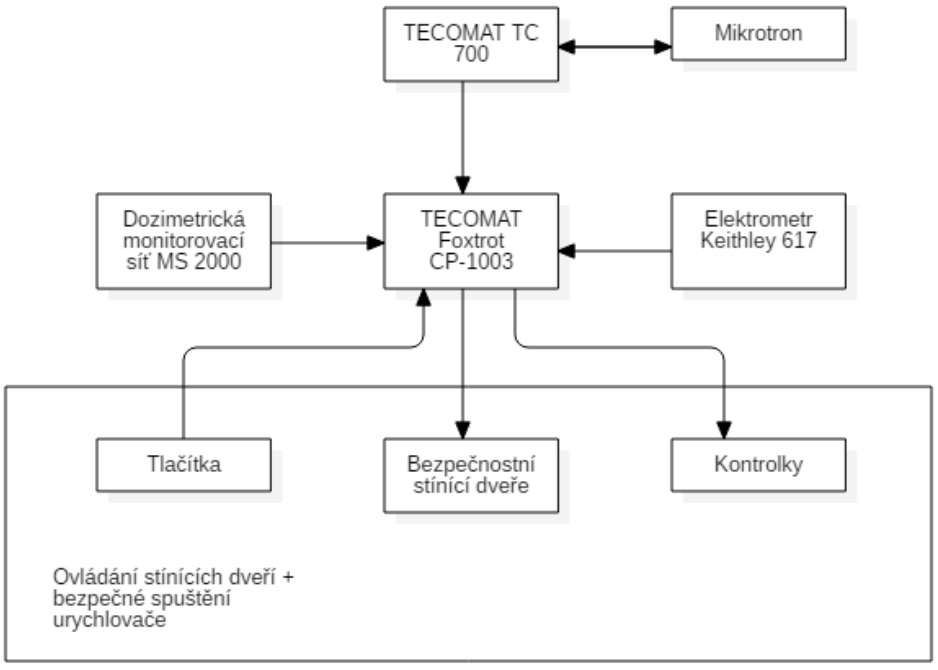
Účelem nového systému je zvýšení bezpečnosti provozu urychlovače, lepší přehlednost procesu z dispečerského stanoviště, zjednodušení některých úkonů, které nejsou dosud automatizované a implementace nových funkcí. Automatizace zjednoduší obsluhu urychlovače a také bude disponovat lepší diagnostikou s automatickým ukládáním záznamů o provedených činnostech.

2 Popis systému

Navrhněte systém pro automatické řízení a zabezpečení provozu urychlovače. Systém bude zajišťovat chod urychlovače, detekci havarijních stavů a kontrolu funkčnosti všech spustitelných prvků systému.

Systém je obsluhován pomocí PLC (Programable logic controler) TECOMAT Foxtrot CP-1003 a uživatelské rozhraní je vytvořeno v prostředí Reliance. Celý systéme se skládá ze tří částí. První část obsahuje bezpečnostní systém zajišťující ovládání stínících bezpečnostních dveří oddělující obsluhu urychlovače od ozařovací místnosti a bezpečné spuštění urychlovače. Druhá část slouží ke sběru dozimetrických údajů z monitorovací sítě MS 2000 dodaná fimou BQM. Monitorovací síť nepřetržitě sleduje stav radiace jak ozařovací místnosti, tak v ostatních prostorách laboratoře. Třetí část umožňuje automatizované měření dozimetrických hodnot dosažených v ozařovaném vzorku při běžném provozu urychlovače. PLC v tomto případě sbírá data z elektrometru Keithley 617.

Na obrázku č. 1 je zobrazené blokové schéma zapojení systému.



Obr. 1: Blokové schéma zapojení systému

2.1 Popis systému pro účel analýzy

Následující text je použitý pro účely lexikální analýzy při tvorbě systému. Text jsem vypracoval za účelem tvorby prvotního návrhu systému. Výsledný model byl následně mnohokrát upravován požadavky pracovníku mikrotronové laboratoře. Jedná

se tedy o prvotní představu systému a neobsahuje kompletní informace, které byly využity až v modifikaci finálních modelů. K jeho vypracování bylo nutné pochopit účel systému a prostředky, které byly vynaloženy či použity pro fungování nového systému.

Uvnitř kobky jsou 4 tlačítka: 2 stop tlačítka, poslední osoba, otevření dveří a dva senzory koncový spínač dveří pro otevřeno a pro zavřeno. Mimo kobku bude tlačítko otevřít, zavřít, stop motor, povolení k provozu, dále senzor koncový spínač (signalizace zavřeno) a kontrolky povolení k provozu, zavřené dveří a člověk v kobce. Stisknutí stop tlačítka způsobí při stisknutí havarijní stav, ve kterém se vypne urychlovač. Tlačítko poslední osoba indikuje absenci člověka v kobce. Tlačítko nouzové otevření dveří, které je fyzicky uvnitř kobky, způsobí otevření dveří za všech okolností. Senzory koncové spínače pro zavření dveří, které jsou uvnitř a vně kobky signalizují dovření dveří. Tlačítka otevření, resp. zavření způsobí otevírání, resp. zavírání dveří. Tlačítko otevřít bude možné stisknout pouze za podmínky, že neprobíhá urychlování.

Ovládání dveří bude realizováno pomocí 3 relé. První relé povolí/zakáže napájení čerpadla (jenž čerpá hydraulický olej pro otevírání/zavírání dveří), druhé relé bude přepínat směr čerpání čerpadla na otevírání a třetí bude přepínat směr čerpání čerpadla na zavírání.

V kontrolním stavu, který by se měl periodicky spouštět aspoň 1 krát za 3 měsíce, budou testovány tlačítka a senzory uvnitř kobky na funkčnost. Tento stav se bude spouštět z dispečerského pracoviště.

V provozním stavu bude systém kontrolovat podmínky a následně dávat signál vyhodnocení, což v případě splnění podmínek umožní spuštění urychlovače pomocí relé „povolení k provozu“. Podmínky pro udělení povolení k provozu jsou následující:

- dveře jsou zavřené (zavření indikují koncové spínače).
- nebylo stisknuto žádné ze stop tlačítek.
- nikdo není v kobce (indikováno tlačítkem „poslední osoba“).

V případě splnění podmínky zavřených dveří se rozsvítí kontrolka zavřených dveří. Pokud je stisknuté tlačítko „povolení k provozu“ a jsou splněny podmínky, rozsvítí se kontrolka „povolení k provozu“. Tlačítko „povolení k provozu“ bude možné stisknout paralelně v dispečerském pracovišti, což bude mít totožnou funkci, stačí stisknout jedno z těchto 2 tlačítek, aby bylo uděleno povolení k provozu. Kontrolka „člověk v kobce“ bude svítit v případě, že bylo stisknuté tlačítko stop či nebylo ještě stisknuto tlačítko poslední.

V případě stisknutí jakéhokoliv tlačítka "stop"po spuštění urychlovače, nastane havarijní stav, kde dojde k vypnutí urychlovače.

Systém bude komunikovat s PLC TECOMAT TC 700 v roli master-slave, kde TC

700 má roli slave. Fyzická vrstva komunikace bude probíhat pomocí RS-485. Od tohoto PLC bude systém získávat dozimetrické informace kontrolního a monitorovaného pásma, které bude možno ve SCADA (Supervisory Control And Data Acquisition", tedy "dispečerské řízení a sběr dat") systému zapsat do provozního deníku stisknutím příslušného tlačítka „uložit dozimetrické údaje“. Dále bude systém registrovat hodnoty proudu z elektrometru a ionizační komory, s jejichž pomocí bude počítat potřebnou dobu expozice daného předmětu.

Provozní deník bude vytvářet záznam při přechodu do havarijního stavu, úspěšný test tlačítek, stisknutí virtuálního tlačítka na zaznamenání dozimetrických údajů a záznam o každém spuštění a vypnutí urychlovače.

2.2 Mikrotron MT 25

Mikrotron MT 25 je relativistický elektronový urychlovač. Vynález mikrotronu je připisována ruskému fyzikovi Vladimiru Iosifoviči Vekslerovi v roce 1944.

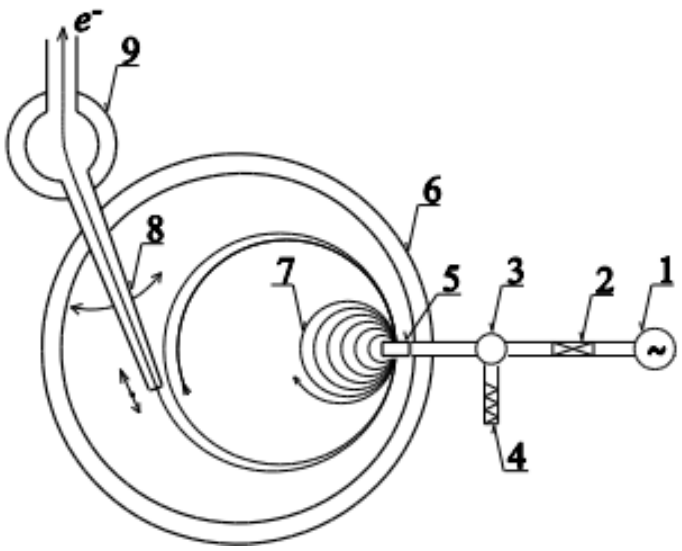
Urychlovač je zdrojem vysoko-energetických elektronů (resp. primárního elektronového záření), sekundárního brzdného záření, tedy vysoko-energetických fotonů a neutronů z fotojaderných reakcí. Svazek urychlených elektronů se pohybuje ve stacionárím homogenním magnetickém poli po kruhových drahách, jejichž průměr je dán jejich energii. Kinetická energie elektronů se zvyšuje při průchodu dutinovým rezonátorem, kde jsou tyto elektrony urychlovány elektrickým polem. Urychlující pole je pomocí magnetronu pulzně buzeno, s elektrickou složkou ve směru pohybu elektronu. Díky několikanásobnému oběhu v magnetickém poli urychlovače je svazek takto urychlených elektronů striktně monoenergetický (disperze energie řádově desítky keV). Manipulací nastavitelného vývodu svazku (část č. 8 na obr. 2) lze docílit vyvedení svazků elektronů odpovídajících různým energiím od 6 MeV do 25 MeV v závislosti na zvolené dráze. Dráhy elektronů jsou přibližně znázorněny na obr. 2. [1]

Urychlovač se využívá převážně k následujícím účelům:

- **Výroba radioizotopů** - Převážně pro lékařské účely v nukleární medicíně. V pozitronové emisní tomografii se pomocí radioizotopů určuje přítomnost nádorového onemocnění a jeho případný stagging (stádium). Dále se používá k diagostice a diferenciální diagnostice v lékařství, např. zjištění prokrvení myokardu, zjištění ventilace a perfuze plic, zjištění lokalizace krvácení v trávicím traktu. Přímo na mikrotronech se vytváří pouze radioizotop jódu ze xenonu, ostatní radioizotopy (nejčastěji 99m technecium nebo 111m Indium) se vytváří převážně na cyklotronech či reaktorech.
- **Testování rezistence materiálů na Ionizující záření** - Provádí se testy,

kdy se ozařují materiály (nejčastěji elektronika) a zjišťuje se odolnost proti ionizujícímu záření.

- **Určování složení materiálů** - Nedestruktivní metoda určování složení materiálů. Např. množství zlata či flóru v hornině nebo složení materiálu archeologických předmětů.
- **Kalibrace detektorů ionizujího záření (dozimetrů)**
- **Sterilizace** - Tento způsob se nejčastěji využívá pokud daný materiál "nesnese"vyšší teploty při tepelné sterilizaci.
- **Síťování polymerů** - Polymerům, které jsou vystaveny ionizujícímu záření, se zlepšují mechanické vlastnosti. Ionizující záření přeruší vazby v polymeru a dojde k "převazbení"jednotlivých polymolekul polymeru, což má za následek nižší míru mechanické anizotropie, zhuštění či ztvrdnutí polymeru.



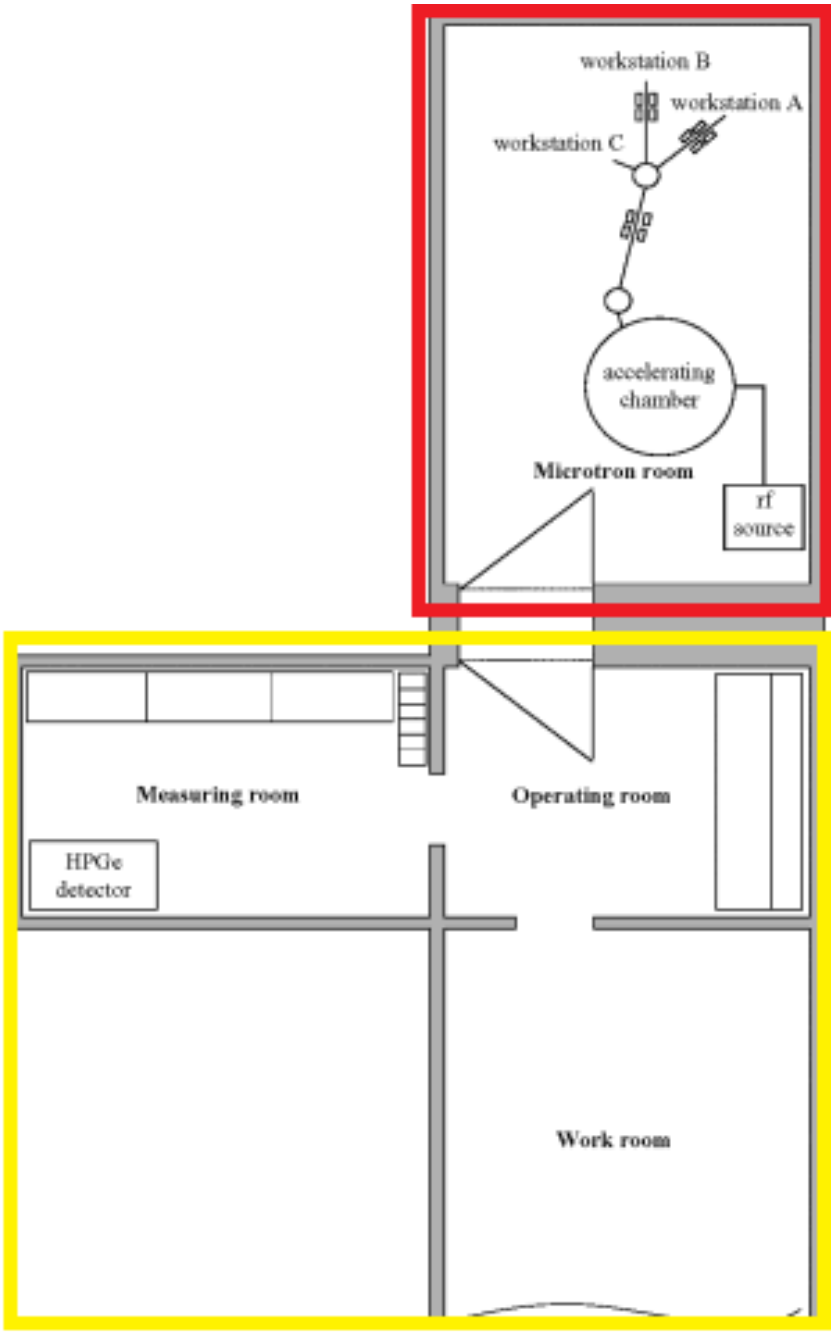
Obr. 2: Schéma mikrotronu MT25 [1]. 1 – magnetron, 2 – posouvač fáze, 3 – cirkulátor, 4 – vodní zátěž, 5 – urychlovací rezonátor, 6 – vakuová komora urychlovače, 7 – trajektorie urychlovaných elektronů, 8 – nastavitelný vývod svazku, 9 – první deflekční magnet

2.3 Popis pracoviště

Pracoviště urychlovače se nachází v Praze v podzemních prostorách vrchu Vítkov. Tunel, ve kterém se pracoviště nachází, má šířku 3 m, výšku 3, 5 m, délku 50 m a disponuje 7 místnostmi. Místnost, kde se nachází samotný urychlovač, (dále jen místnost urychlovače) je stíněná od okolních prostor stěnou (2, 5 m tlustou, vyplněnou litým betonem) s průchodem, které se skládá z 2 stínících dveří (každé z nich tlusté přibližně 0, 5 m, vylité betonem a po celé ploše pokryté 15 cm olova), ovládanými 2 hydraulickými písty. Ovládání těchto pístů bude součástí systému navrhovaného v

rámci této práce. Stěnou také prochází ventilační potrubí, které je situované a tvarované za účelem znemožnění přímému průchodu ionizujícího záření. [2]

Na obr. č. 3 je zobrazeno pracoviště urychlovače. Žlutě je označeno kontrolované pásmo, jehož radiace nesmí být vyšší než maximální přípustná pro personál. A červeně je označeno sledované pásmo (místnost s urychlovačem).



Obr. 3: Pracoviště urychlovače [3]

3 Teoretická část

Tato část se bude zabývat teorií v oblasti ionizujícího záření (dále pouze IZ). Navrhovaný systém v rámci této diplomové práce se zabývá mimo jiné zabezpečovacím systémem, jehož primární účel je ochránit personál před vysokými dávkami ionizujícího záření. Z toho důvodu považuji za nutné definovat základní veličiny, pojmy, vztahy a principy, vztahující se k radiační ochraně.

3.1 Ionizující záření

Ionizujícím zářením nazýváme záření, jehož kvanta nabývají tak vysoké hodnoty energie, že jsou schopny vyrazit elektrony z atomových obalů, čímž ionizují danou látku.

Touto základní charakteristikou disponují všechny typy ionizujícího záření. Každý typ ionizujícího záření má však specifický mechanismus interakce záření s hmotou.

3.2 Veličiny a jednotky v radiační fyzice

Vzhledem k faktu, že moje práce bude sloužit primárně k ochraně lidí v laboratoři, je nutné si vysvětlit základní principy a veličiny spojené s energií absorbovanou v látce a její projevy.

Většina vztahů, o kterých budu později psát vychází ze základní stochastické veličiny *sdělená energie* ϵ, což udává množství energie, která byla předána ionizujícím zářením látce v určitém objemu a lze vyjádřit vztahem č. 1 [4]

$$\epsilon = R_{in} + R_{out} + \sum Q \tag{1}$$

Kde R_{in} resp. R_{out} je zářivá energie, která vstoupila resp. vystoupila do daného objemu látky. Zářivou energií rozumíme sumu všech energií všech ionizujících částic s výjimkou klidových energií. $\sum Q$ je součet všech změn klidových energií jader a elementárních částic, ke kterým ve vztaženému objemu došlo při jakýchkoliv jaderných přeměnách. Změna klidové energie Q může nabývat záporných hodnot pro vzrůst klidové energie a naopak. Formálně by se do $\sum Q$ měly zahrnout i změny chemických vazeb, ty se však obvykle zanedbávají. Veličina ϵ má jednotku J, neboť se jedná o energii. [4]

Pokud *sdělenou energii* vztáhneme na jednotku hmotnosti m látky, které byla tato

energie předána, dostaneme *měrnou (sdělenou) energii* z:

$$z = \frac{\epsilon}{m} \tag{2}$$

Jednotkou této veličiny je Gray (Gy). Ze vztahu vyplývá, že tato jednotka má fyzikální význam $\mathrm{J\,kg^{-1}}$.

Uvažujeme-li střední měrnou energii a snižujeme objem, ve kterém je ukládána, dostaneme dnes nejužívanější dozimetrickou veličinu *absorbovaná dávka* (často pouze *dávka*). Dávku lze také vyjádřit tvarem diferenciální rovnice č. 3.

$$D = \lim_{m \to 0} \overline{z} = \frac{\mathrm{d}\overline{\epsilon}}{\mathrm{d}m} \tag{3}$$

Dávka vyjadřuje předání určitého množství energie. Pokud bychom chtěli znát velikost dávky za jednotku času, zavedeme veličinu *dávkový příkon*:

$$\dot{D} = \frac{\mathrm{d}D}{\mathrm{d}t} \tag{4}$$

Kde $\mathrm{d}D$ je dávkový příkon za jednotku času $\mathrm{d}t$ s jednotkou $\mathrm{Gy\,s^{-1}}$. [4]

Veličiny, které jsme si zavedli do této chvíle (dávka, dávkový příkon,..), nevystihují dostatečně efekty záření na živé organismy, kdy se nezohledňuje závislost lokální hustotě ionizace způsobené radiací. Z toho důvodu bylo nutné zavést veličinu dávkový ekvivalent H, která tyto faktory zohledňuje. Dávkový ekvivalent H, jenž byl stanoven jako součin dávky D ve tkáni a jakostního činitele Q, případně dalších modifikujících koeficientů. Jakostní činitel Q vyjadřuje rozdílné biologické účinky dle typu záření. Pro představu, radiační váhový faktor u fotonů a elektronů se pohybuje okolo 1 ale u α-částic či těžkých iontů je tento faktor roven přibližně 20. [4]

$$H = DQ \tag{5}$$

Dávkový ekvivalent má jednotku $\mathrm{J\,kg^{-1}}$ stejně jako dávka a aby se tyto dvě veličiny rozlišili, jednotka nese název sievert (Sv). Dávkový ekvivalent ve formě vztahu č. 5 je používán pro praktické měření na pracovištích a pro osobní měření je užíván osobní dávkový ekvivalent $H_p(d)$, což je dávkový ekvivalent v daném místě pod povrchem těla v hloubce d v měkké tkáni. Získává se přepočtem dávky na dozimetru pomocí jakostního činitele Q a absorbce v tkáni podle energie záření. [5]

Ekvivalentní dávka ve tkáni či orgánu H_T, jejíž vztah je označen č. 6. Tato veličina udává dávkový ekvivalent v daném místě pod povrchem těla. Hloubka bodu pod

povrchem těla závisí na druhu záření a místu těla či orgánu. [5]

$$H_T = \sum_R w_R D_{T,R}, \tag{6}$$

Kde $D_{T,R}$ je střední absorbovaná dávka ve tkáni či orgánu, způsobená zářením druhu R a w_R je příslušný tkáňový váhový faktor

V poslední řadě je tu veličina efektivní dávka E (vztah č. 7), která je dána součtem všech ekvivalentních dávek ve tkáních nebo orgánech T vážených tkáňovým váhovým faktorem w_T. Efektivní dávka zahrnuje senzitivitu jednotlivých tkání a orgánů z hlediska vzniku stochastických změn (zhoubných nádorů a genetických mutací). Jednotkou je opět Sievert. Nutno dodat, že součet váhových faktorů je 1. Tzn. vztah vyjadřuje podíl jednotlivých orgánů a tkání na vzniku stochastických změn při celotělním ozáření, což se jeví jako výhodné pro vyjádření radiační zátěže těla při nerovnoměrném ozáření. [5]

$$E = \sum_T w_R H_T, \tag{7}$$

3.2.1 Základní principy radiační ochrany

Existují 3 základní principy radiační ochrany, které lez odvodit ze vztahů výše zmíněných.

- **Minimalizace času** - ze vztahu č. 4 vyplývá přímá úměra délky expozice záření s velikostí přijaté dávky.
- **Vzdálenost** - ze vztahu č. 8, který vyjadřuje kermovou vydatnost, je zjevná kvadratická závislost vzdálenosti l na kermové vydatnosti $V_{K,\delta}$. Je-li v uvažovaném objemu rovnováha nabitých částic, pak kerma K_δ je rovna dávce. Veličina $\dot{K}_\delta$ se nazývá kermový příkon, který je možné vyjádřit jako $\frac{\mathrm{d}K_\delta}{\mathrm{d}t}$. [4]

$$V_{K,\delta} = \dot{K}_\delta l^2 \tag{8}$$

- **Stínění** - bariéra z materiálu, který absorbuje záření. Nejvhodnější jsou materiály s vysokým protonovým číslem a vysokou hustotou. Nejčastěji se používá olovo, beton, magnetit, ocel, wolfram a baryt.[4]

3.2.2 Monitorování v radiační ochraně

Pomocí radiačního monitorování se uskutečňuje splnění požadavků na limit ozáření osob a dalších požadavků na bezpečný provoz pracovišť se zdroji ionizujícího záření.

Monitorováním se rozumí měření veličin charakterizujících radiační pole, interpretace a hodnocení ozáření pracovníků a dalších osob. Obecně program monitorování obsahuje 4 části: [4]

- **monitorování pracoviště**
- **osobní monitorování**
- **monitorování výpustí**
- **monitorování okolí**

Monitorování okolí bylo vyřešeno tím, že se prostor urychlovače nachází v tunelu a před konstrukcí zařízení bylo odstínění zeminou označeno (podloženo výpočty) za dostačující a bezpečné. Stejným způsobem bylo vyřešeno monitorování výpustí.

Pracoviště je rozděleno na 2 prostory, které jsou definovány vyhláškou č. 422/2016 Sb. o radiační ochraně a zabezpečení radionuklidového zdroje.Bezprostřední prostor u urychlovače je klasifikován jako sledované pásmo, kde lze očekávat překročení obecných limitů, zde se provádí pouze monitorování pracoviště. Druhý prostor, je klasifikován jako kontrolované pásmo, kde lze očekávat překročení 0, 3 limitu pro radiační pracovníky (6 mSv/rok). Provádí se osobní monitorování dávek. [4]

V praxi se používají 3 typy monitorování: [4]

- **soustavné, nepřetržité** (kontinuální)
  Vztahuje se na danou praxi, účelem je potvrzení, že podmínky na pracovišti zůstávají bezpečné v souladu s požadavky povolení praxe. Tento typ monitorování se v mém případě realizuje pomocí dvoukanálového kontrolního panelu MS KP4. Toto měřící zařízení není pravidelně zkalibrováno, takže má spíše informativní charakter.
- **pravidelné**
  Měření se periodicky opakuje a má za cíl potvrdit, že podmínky na pracovišti zůstávají bezpečné v souladu s požadavky povolení praxe.
- **operativní**
  Měření se provádí při či před určitou činností s cílem zhodnotit a zajistit přijatelnost činnosti z hlediska systému limitování a podmínek povolení praxe. K tomuto účelu je na pracovišti urychlovače mobilní měřící zařízení, které je pravidelně kalibrované a vyhovuje podmínkám provozu urychlovače danými atomovým zákonem.

3.2.3 Evidence ozáření

Držitel, který získal od orgánu státního dozoru nad radiační ochranou povolení k dané praxi, je povinen vést evidenci s hodnocením ozáření pracovníků se zdroji záření, tzn. k nakládání se zdroji IZ či k činnosti vedoucí k ozáření. Právní předpisy ČR udávají povinnost archivovat tyto informace po dobu 50 let. Informace musí obsahovat program osobního monitorování včetně metodik, na které je navázán a údaje o osobních dávkách pracovníků a o ostatních osobách, které byly vystaveny záření, včetně doby vystavení. [4]

Program, o kterém budu více psát v praktické části práce, k těmto účelům využívá databázi s názvem "Provozní deník", do kterého zaznamenává mimo jiné dozimetrická data měření.[4]

3.2.4 Detekce IZ

Existuje mnoho způsobů rozdělení detektorů IZ. Mezi základní rozdělení detektorů dle principu detekce patří :

- **Fotografické** - jsou založené na fotochemických účincích záření (filmové dozimetry, RTG filmy, jaderné emulze), nebo využívající fotografické zobrazení stop částic v určitém látkovém prostředí (mlžné a bublinové komory). Jedná se o kumulativní (integrální) detektory, které se vyznačují tím, že po ukončení expozice záření stále vykazují hodnotu jako při expozici, a tudíž je lze vyhodnotit dodatečně. V mikrotronové laboratoři jsou běžně využívané, vyhodnocování je prováděno 1 krát za měsíc. [6]
- **Materiálové** - využívají dlouhodobější změny vlastností určitých látek (složení, barva) působením ionizujícího záření. Tyto detektory se vyznačují nízkou citlivostí a z toho důvodu se používají pro vysoké intenzity záření a pro kumulativní detekci. Do této kategorie patří např. stopové detektory či křemíkové diody s dlouhou bází určené na měření radiační dávky. [6]
- **Elektronické** - převádějí část absorbované energie ionizačního záření na elektrické proudy či impulsy (přímým nebo zprostředkovaným způsobem), které se zesilují a vyhodnocují v elektronických aparaturách. Jedná se o kontinuální detektory (nikoliv kumulativní), neboť detektor indikuje aktuální hodnotu záření. Do této kategorie patří plynové detektory, scintilační detektory, polovodičové detektory, mikrokalorimetrické detektory a magnetické spektrometry. [6]

Na obr. č. 4 je obecně zobrazeno schéma plynového detektoru IZ, které se používají v mikrotronových laboratořích. Jedná se o Gaiger-Millerův počítač a ionizační komoru. Podrobněji budou popsány v kapitole 3.2.5.

Obr. 4: Bodové schéma elektrického detektoru ionizujícího záření [6]

3.2.5 Plynové detektory

Plynový detektor je systém dvou navzájem izolovaných elektrod, obvykle umístěných ve válcovém plášti, který může sloužit jako jedna z elektrod. Prostor mezi těmito elektrodami je vyplněn plynem, obvykle argonem s příměsí dusíku či methanu, nebo vzduchem, který je za normálních podmínek nevodivý. K elektrodám se připojí zdroj stejnosměrného napětí.

Při expozici detektoru IZ dochází v plynu k ionizaci, což způsobí, že plyn se stává vodivým. Při ionizaci plynu dochází k rozpadu některých neutrálních částic na kladné, záporné ionty a elektrony. Tyto elektricky nabité částice a ionty putují k elektrodě opačné polarity a tím mezi elektrodami protéká slabý proud. Vzniklý elektrický proud se rovná elektrickému náboji odevzdanému ionty za jednu sekundu. Na obr. č. 6 je zobrazený graf závislosti počtu elektronů (dopadajících na elektrody) na hodnotě elektrického napětí.[5]

Graf je rozdělen na 6 oblastí označenými římskými číslicemi.

I. **Oblast platnosti Ohmova zákona** - Ionty vzniklé ionizací plynu spolu opět rekombinují, pravděpodobnost rekombinace klesá s rostoucí rychlostí iontů (napětím na elektrodách). Ionizační proud roste přibližně úměrně napětí na elektrodách. Oblast I se k detekci IZ nepoužívá. [5]

II. **Oblast nasyceného proudu** - Silnější elektrické pole způsobí, že se ionty pohybují natolik rychle, že nestihnou zrekombinovat a účastní se vedení proudu. Ionizační proud je proto nezávislý na napětí, ale pouze na intenzitě záření. V této oblasti pracují detektory na bázi ionizační komory. [5]

III. **Proporcionální oblast** - Ionty vzniklé ionizací od IZ (primární ionty) jsou urychlovány silným elektrickým polem do té míry, že při srážce s neutrálními molekulami či atomy plynu vytváří další ionty (sekundární ionty). Počet sekundárních iontů je přímo úměrný počtu primárních iontů. V této oblasti pracují proporční detektory.[5]

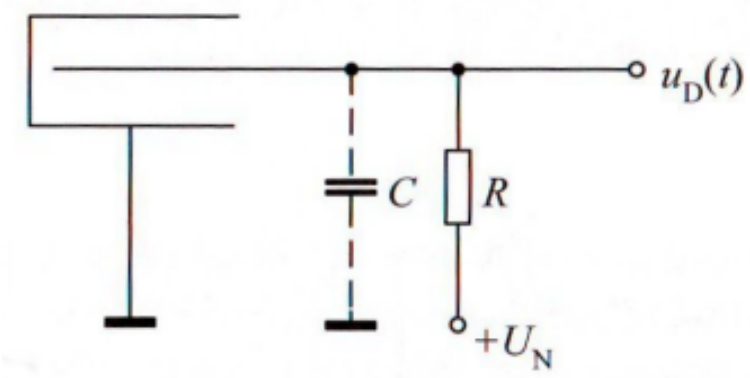
IV. **Oblast zmenšené proporcionality** - Ionty jsou silným elektrickým polem

urychleny do té míry, že dochází k exponenciálnímu nárůstu počtu sekundárních iontů.[5]

V. **Oblast Geiger-Müllerových detektorů** - Geiger-Müllerův (dále jen GM) detektor pracuje v impulzním režimu. V této oblasti již nezávisí na druhu či energii ionizující částice. Kvantum IZ ionizuje plyn, který se nachází mezi elektrodami, z něhož se stane vodič. Plyn má nižší tlak, než je atmosferický, tudíž při ionizaci je mezi atomy dostatečný prostor, aby ionty získaly v elektrickém poli dostatečnou kinetickou energii a srážkami s ostatními neutrálními částicemi tvořily sekundární ionty, které tento proces zopakují. Takto se vytvoří až 10^{10} sekundárních iontů z jednoho primárního.

   Tímto lavinovitým procesem vznikne mezi elektrodami samovolný výboj a obvodem projde poměrně silný proudový impulz. Kvantum IZ je tedy detekováno jako elektrický impuls a tyto impulsy se počítají (proto se detektor nazývá počítač).[5]

   Na obrázku č. 5 je zobrazeno zjednodušené schéma obvodu GM počítače. Na pracovním odporu R vzniká vysoký napěťový impuls (vyplývající z Ohmova zákona), který se vede přes oddělovací kondenzátor C ke zpracování k příslušné elektronické jednotce.



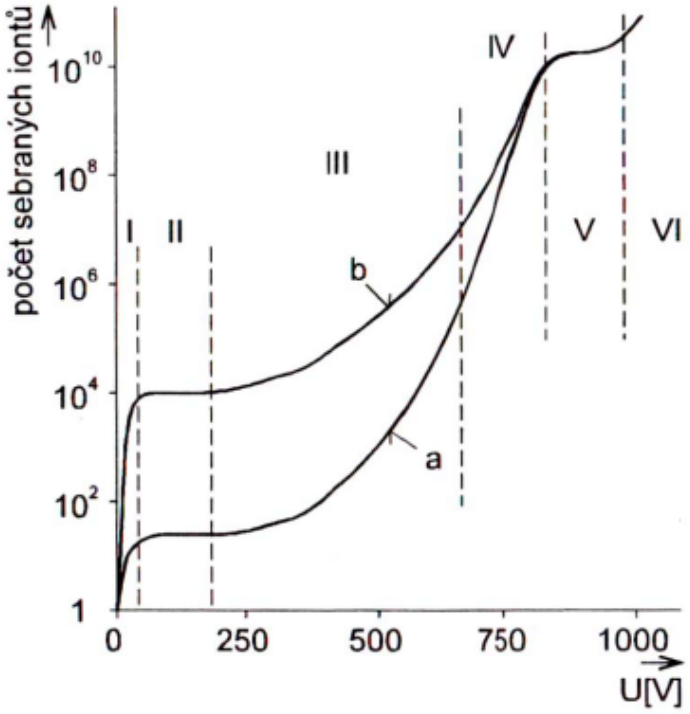
Obr. 5: Schematické zapojení GM detektoru [6]

   Aby nedošlo k poškození zařízení a kontinuitě měření, je nutno přerušit jednotlivé výboje. Toho se docílí úbytkem napětí na elektrodách pomocí vysokého pracovního odporu R, což sníží produkci sekundárních iontů. Rekombinací a deexcitací atomů dochází k tvorbě UV záření, což je také IZ a prodlužuje výboj. Z toho důvodu se do plynu také přidává tzv. zhášecí látka (často páry metylalkoholu, bromu, apod.), která absorbuje UV záření.

   Výhodou GM detektorů je velká účinnost při detekci nabitých částic a velká amplituda impulsů, která se pohybuje od jednotek do stovek V, což je důvod, proč GM nepotřebuje zesilovač. Důležitým faktorem je tzv. mrtvá doba, což je

doba trvání impulsu, po kterou detektor nedetekuje další částice, což ovlivňuje frekvenci vzorkování detektoru. [5]

VI. **Oblast jiskrového výboje** - IZ vyvolá mezi elektrodami přeskok jiskry, což způsobí silný elektrický impulz v obvodu, následovaný světelným a zvukovým projevem. V této oblasti pracují jiskrové detektory.[5]



Obr. 6: Počet sbíraných iontů jako funkce napětí [6]

4 Praktická část

4.1 Object-Modeling Technique (OMT) a Unified Modeling Language (UML)

Rumbaugh, Blaha, Premerlani, Eddy a Lorensen vytvořili v roce 1991 metodu OMT jako metodiku pro efektivní tvorbu modelu softwaru. Dle Rumbaugha má metoda 4 účely:

- testování entit, než se fyzicky vytvoří (simulace)
- komunikace se zákazníkem
- vizualizace (či prezentace)
- snížení komplexnosti

Celý proces vývoje OMT softwaru má 4 fáze.

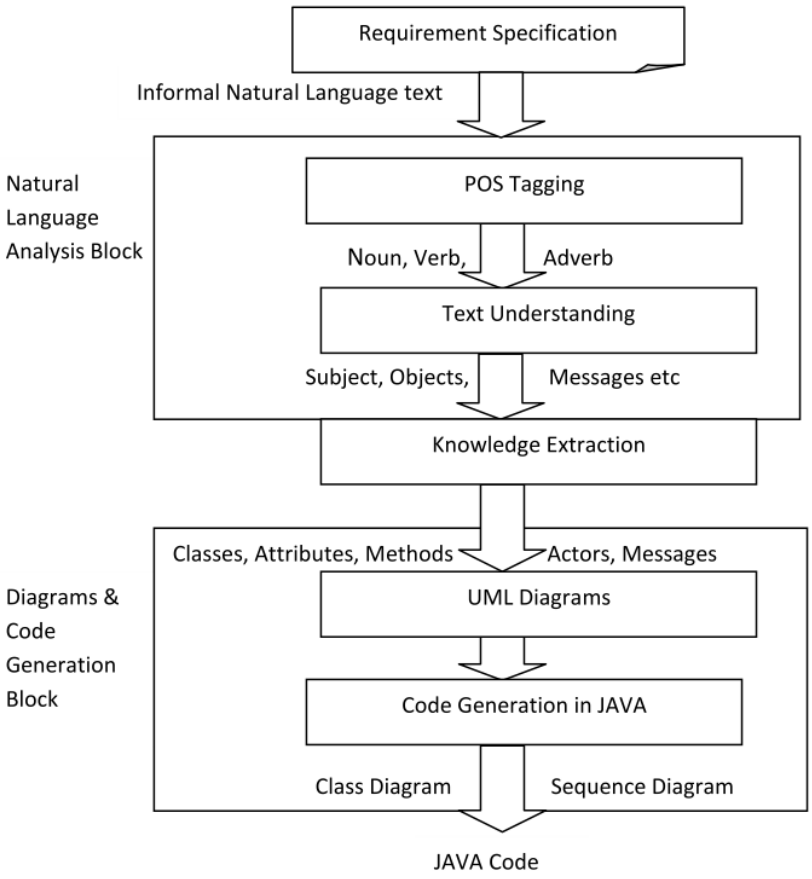
- Analýza
- Návrh systému
- Návrh objektů
- Implementace

Většina modelování systému je obsažena ve fázi analýzy, která obsahuje 3 modely (objektový 4.1.1, dynamický 4.1.2 a funkční 4.1.3). Obvykle je nejdůležitější objektový model, avšak v tomto případě je pro naše účely důležitější dynamický model. Všechny tyto 3 modely dohromady kompletně popisují systém. Tvorba systému je schématicky zobrazena na obrázku č. 7. Jediná změna v tomto schématu je generování kódu. Kód programu se nevygeneruje do jazyku, který potřebuji, takže jsem musel kód program vytvořit kompletně manuálně.

OMT disponuje silným nástrojem pro popis systému, který se nazývá lexikální analýza. Touto metodou lze velice strukturovaně sestavit diagramy na základě slovního popisu, který je uveden v kapitole 2.1. [7] [8]

Jazyk UML začali vyvíjet Grady Booch a Jim Rumbaugh společně se svojí firmou Rational Software, jenž se později stala součástí firmy IBM. Jazyk vychází z různých metodik jako například OMT či OMSE.

UML je jazyk pro specifikaci, vizualizaci, konstrukci a dokumentaci artefaktů softwarových systémů. Řadí se mezi grafické jazyky a používá se při navrhování velkých projektů. Mezi jeho výhody patří hlavně názorná grafická reprezentace, která má výhody při komunikaci mezi programátorem a zákazníkem. Zákazník často porozumí



Obr. 7: Metodika tvorby systému [8]

grafickému zobrazení snáze, než by tomu bylo v případě negrafického zobrazení. UML může být pro malé aplikace neefektivní. Mezi další výhody UML se řadí také možnost automaticky generování šablony, vytvořené pro daný projekt do různých programovacích jazyků.

UML se zakládá na 2 základních pilířích. Diagramy struktury a behaviorální. Behaviorální diagram zachycuje chování a diagram struktury zachycuje strukturu projektu. Tyto 2 pilíře se dělí na statické a dynamické. Statické diagramy neuvažují časovou složku. [9]

Pro tento projekt jsem použil program StarUML, který nabízí zdarma verzi s omezeními. Jedno z omezení byl vodoznak, který se vytvořil v pozadí vyexportovaného diagramu do formátu pdf. Tento fakt jsem zjistil až v době, kdy jsem měl hotové diagramy a chtěl je vložit do této diplomové práce. Důsledkem toho jsem nemohl vytvořit diagramy jako vektorovou grafiku ale ve formátu png, což má za následek mírně sníženou grafickou kvalitu diagramů.

4.1.1 Objektový model

Model, reprezentující statický, strukturální a datový aspekt systému. Zde jsou definovány třídy společně s jejich vztahy, atributy a operacemi. Objektový model je nejlépe reprezentován graficky třídovým diagramem zobrazeným na obrázcích č. 910. Objektový diagram obsahuje třídy. Každá třída reprezentuje sadu jednotlivých objektů. Tyto třídy jsou vzájemně propojeny spojnicemi, které reprezentují spojení objektů daných tříd s objekty jiných tříd. Lexikální analýza umožňuje tvorbu objektového modelu následnými kroky: [7]

1. **Vytvořit slovní popisy modelovaného problému**.
   V kapitole 2.1 (Popis systému pro účel analýzy) je uvedený text (dále jen popis systému), který obsahuje popis modelovaného systému a bude použitý pro tyto účely.
2. **Určit třídy objektů**.
   Následující tabulka č. 1 obsahuje výčet podstatných jmen a slovních spojení, které tvoří adepty na budoucí objekty a třídy.

Tab. 1: Seznam podstatných jmen a slovních spojení

Podstatná jména	**Podstatná jména**	**Podstatná jména**
zavření dveří	kobka	provozní stav
člověk v kobce	master-slave	relé
dispečerské obrazovky	napájení čerpadla	RS 485
dispečerském pracovišti	nouzové otevření dveří	řídící a zabezpečovací systém
doba expozice předmětu	otevírání	SCADA systém
dovření dveří	kontrolní stav	senzory
dozimetrické informace	ovládání dveří	směr čerpání čerpadla
fyzická vrstva komunikace	PLC TECOMAT TC 700	stop
havarijní stav	provozní deník	stop motor
hodnotu proudu z elektrometru	Podstatná jména	tlačítka
hodnotu radiace z dozimetru	poslední osoba	tvorba záznamu
koncový spínač zavřeno uvnitř	povolení k provozu	udělení povolení k provozu
koncový spínač zavřeno vně	povolení k provozu (kontrolka)	vypnutí urychlovače
		Pokračování na další straně

Tab. 1 – Pokračování z předchozí strany

Podstatná jména	**Podstatná jména**	**Podstatná jména**
kontrola funkčnosti	proud z ionizační komory	zavírání
kontrolky	provoz urychlovače	zavřené dveře

3. **Zrušit nepotřebné a chybné třídy**.
   Ze tabulky č. 1 z předchozího bodu jsem vyřadil následující slovní spojení z důvodů uvedených v názvu tabulky:

Tab. 2: Nadbytečné objekty a třídy

dispečerské obrazovky	dispečerské pracoviště	směr čerpání čerpadla

Tab. 3: Irelevantní třídy

doba expozice předmětu

Tab. 4: Neurčité objekty

kobka	provozní deník	relé

Tab. 5: Atributy

hodnotu proudu z elektrometru	hodnotu radiace z dozimetru
proud z ionizační komory	provoz urychlovače
stop motor	

Tab. 6: Operace

kontrola funkčnosti havarijních prvků systému	udělení povolení k provozu
napájení čerpadla	vypnutí urychlovače
otevírání	zavírání
otevření dveří	zavření dveří
tvorba záznamu	

Tab. 7: Implementační konstrukty

fyzická vrstva komunikace	PLC TECOMAT TC 700
master-slave	RS 485

Tab. 8: Stavy

havarijní stav	kontrolní stav	provozní stav

Po eliminaci výše zmíněných podstatných jmen a slovních spojení zbyly objekty a třídy uvedené v tabulce č. 9.

Tab. 9: Zbylá slovní spojení

zavřené dveře	stop
člověk v kobce	dozimetrické informace
dovření dveří	kontrolky
koncový spínač zavřeno uvnitř	ovládání dveří
koncový spínač zavřeno vně	řídící a zabezpečovací systém
nouzové otevření dveří	SCADA systém
poslední osoba	senzory
povolení k provozu	tlačítka
povolení k provozu (kontrolka)	

4. **Připravit slovník dat (data dictionary)**.
   Tabulka č. 10 přehledně popisuje slovní spojení, se kterými budu pracovat v rámci systému.

Tab. 10: Sémantický popis slovních spojení

Slovní spojení	**Popis**
člověk v kobce	kontrolka signalizující člověka v kobce
dovření dveří	signalizuje (pomocí koncových spínačů), že jsou dveře dovřené
dozimetrické informace	dozimetrické informace z přístrojů pracovního panelu
havarijní stav	zda je urychlovač v havarijním režimu
hodnotu proudu z elektrometru	hodnotu proudu z elektrometru
hodnotu radiace z dozimetru	hodnotu radiace z dozimetru
koncový spínač zavřeno uvnitř	senzor, signalizující dovření 1. části dveří
koncový spínač zavřeno vně	senzor, signalizující dovření 2. části dveří
kontrola funkčnosti havarijních prvků systému	aktualizování výsledku kontroly
kontrolky	obsluhování kontrolek
kontrolní stav	zda je urychlovač v kontrolním režimu
napájení čerpadla	zda je motor v běhu či nikoliv
nouzové otevření dveří	bezpodmínečné otevření dveří zevnitř kobky za všech okolností
otevírání	tlačítko pro otevření dveří mimo kobku
Pokračování na další straně	

Tab. 10 – Pokračování z předchozí strany

Slovní spojení	Popis
otevření dveří	motor způsobuje otevírání dveří
ovládání dveří	charakterizuje ovládání dveří
poslední osoba	tlačítko, které signalizuje, že v kobce není další člověk
povolení k provozu	tlačítko
povolení k provozu (kontrolka)	kontrolka
proud z ionizační komory	proud z ionizační komory
provoz urychlovače	informace, zda je rychlovač v běhu
provozní stav	zda je urychlovat v provozním režimu
řídící a zabezpečovací systém	obsahuje základní atributy a operace systému
SCADA systém	řízení systému z dispečerského pracoviště
senzory	všechny senzory
stop	havarijní tlačítko
stop motor	tlačítko pro zastavení otevírání/zavírání dveří
tlačítka	všechna tlačítka
tvorba záznamu	vytvoření záznamu o proběhlé havarijní události
udělení povolení k provozu	po splnění podmínek umožnění spuštění urychlovače
vypnutí urychlovače	způsobí vypnutí procesu urychlování
zavírání	tlačítko pro zavření dveří mimo kobku
zavřené dveře	kontrolka signalizující zavřené dveře
zavření dveří	motor způsobuje zavírání dveří

5. **Určit asociace mezi třídami**.

   V textu slovního popisu systému asociace (nebo také vazby) mezi třídami odpovídají slovesům a slovesným frázím. Vazby v objektovém modelu představují souvislosti, které více slouží k pochopení funkce interakce systému, než k samotnému vytváření“kódu". Obdobně jako v předchozích bodech jsem za tímto účelem vytvořil tabulku.

   Následující tabulka č. 11 obsahuje seznam sloves a slovesných frází, které které mohou znamenat asociace mezi třídami. Asociace obvykle zachycují 5 skutečností, které jsou uvedené ve výše zmíněné tabulce. Každá slovesná fráze je pomocí

hvězdičky rozřazena v tabulce. Tučně jsou označeny samotná slovesa a fráze. Slova v tabulce, které nejsou tučně označena, se v seznamu vyskytují pro zasazení sloves do kontextu a pro vyšší přehlednost.

Tab. 11: Slovesa a slovesné fráze

Slovesná fráze	**Umístění**				
		Cílené akce			
			Komunikace		
				Vlastnictví a dispozice	
					Rozdělení zodpovědnosti
Uvnitř kobky **budou**	*				
mimo kobku **bude**	*				
způsobí při stisknutí havarijní stav		*			
vypne urychlovač		*			
indukuje absenci osoby v kobce			*		
způsobí nouzové otevření dveří		*			
signalizují dovření dveří			*		
způsobí otevření		*			
způsobí zavření		*			
bude možné stisknout				*	
ovládání dveří **bude realizováno**					*
povolí/zakáže napájení		*			
bude přepínat na otevírání		*			
bude přepínat na zavírání		*			
kontrolní stav, který by se měl **spouštět**		*			
budou testovány					*
Kontrolní stav se **bude pouštět**	*				
systém **bude kontrolovat**					*
zavření **indikují**			*		
nebylo stisknuto			*		
není v kobce	*				
splnění podmínky zavřených dveří			*		
rozsvítí se kontrolka zavřených dveří		*			
rozsvítí kontrolka "povolení k provozu"		*			
nastane havarijní stav		*			

Pokračování na další straně

Tab. 11 – Pokračování z předchozí strany

Slovesná fráze	**Umístění**				
		Cílené akce			
			Komunikace		
				Vlastnictví a dispozice	
					Rozdělení zodpovědnosti
dojde k vypnutí		*			
získat dozimetrické informace			*		
bude možno **zapsat** do provozního deníku				*	
registrovat hodnoty proudu					*
sbírat hodnoty					*
vytvářet záznam		*			

6. **Zrušit nepotřebné a chybné asociace**.
   Z dále uvedených důvodů vyřadím chybné a nepotřebné asociace z důvodů uvedených v názvu tabulky:

Tab. 12: Vazby mezi zrušenými objekty

vypne urychlovač	Kontrolní stav se bude pouštět	nastane havarijní stav

Tab. 13: Nadbytečné vazby

zavření indikují	nebylo stisknuto
není v kobce	bude možno zapsat do provozního deníku
registrovat hodnoty proudu	

Tab. 14: Irelevantní vazby

Uvnitř kobky budou	Mimo kobku bude
indukuje absenci osoby v kobce	bude možné stisknout
Splnění podmínky zavřených dveří	

Tab. 15: Operace

způsobí otevření	budou testovány
způsobí zavření	systém bude kontrolovat
povolí/zakáže napájení	rozsvítí kontrolka“povolení k provozu"
bude přepínat na otevírání	dojde k vypnutí
bude přepínat na zavírání	získat dozimetrické informace
kontrolní stav, který by se měl spouštět	vytvářet záznam

Tab. 16: Implementační vazby

ovládání dveří bude realizováno

Tab. 17: Odvozené vazby

způsobí při stisknutí havarijní stav

Po eliminaci většiny slovesných frází z tabulky č. 11 zbyla následující slovesná fráze obsažená v tabulce č. 18.

Tab. 18: Zbývající slovesná fráze

způsobí nouzové otevření dveří
signalizují dovření dveří
rozsvítí se kontrolka zavřených dveří
sbírat hodnoty

7. **Určit atributy tříd**.
   Následující tabulka č. 19 obsahuje seznam přídavných jmen či jejich skupin, které by měly odpovídat vlastnostem objektů, jako například váha, velikost, rychlost, apod. a také doplňkovým skutečnostem, které z části určují objekty. V následující tabulce jsou samotná přídavná jména označena tučně.

Tab. 19: Adepty na atributy

Přídavná jména	**Přídavná jména**
koncový spínač	**zavřené** dveře
poslední osoba	**stisknuté** tlačítko
nouzové otevření	**fyzická** vrstva
první relé	**dozimetrické** informace
druhé relé	**kontrolního** pásmo
kontrolní stav	**monitorované** pásmo
dispečerského pracoviště	**provozní** deník
provozní stav	**havarijní** stav

8. **Zrušit nepotřebné a chybné atributy**.

Tab. 20: Atributy zrušených objektů

zavřené dveře	stisknuté tlačítko
první relé	druhé relé
kontrolní stav	provozní stav

Následující tabulka č. 21 obsahuje seznam zrušených atributů, které tvoří společně s následnými podstatnými jmény samostatné objekty.

Tab. 21: Objekty

koncový spínač	poslední osoba
nouzové otevření	fyzická vrstva
dozimetrické informace	kontrolního pásmo
monitorované pásmo	dispečerského pracoviště
provozní deník	první relé
druhé relé	

Tab. 22: Atributy vazeb

první relé	druhé relé

Z tabulky adeptů na atributy byly postupně vyřazeny všechny možnosti, což není nezvyklé.

9. **Určit vazby dědičnosti**.
   Vazby dědičnosti jsou dány vazbami typu generalizace v třídovém diagramu na obrázcích č. 9 a č. 10.
10. **Projít vše znova a určit nedostatky**.

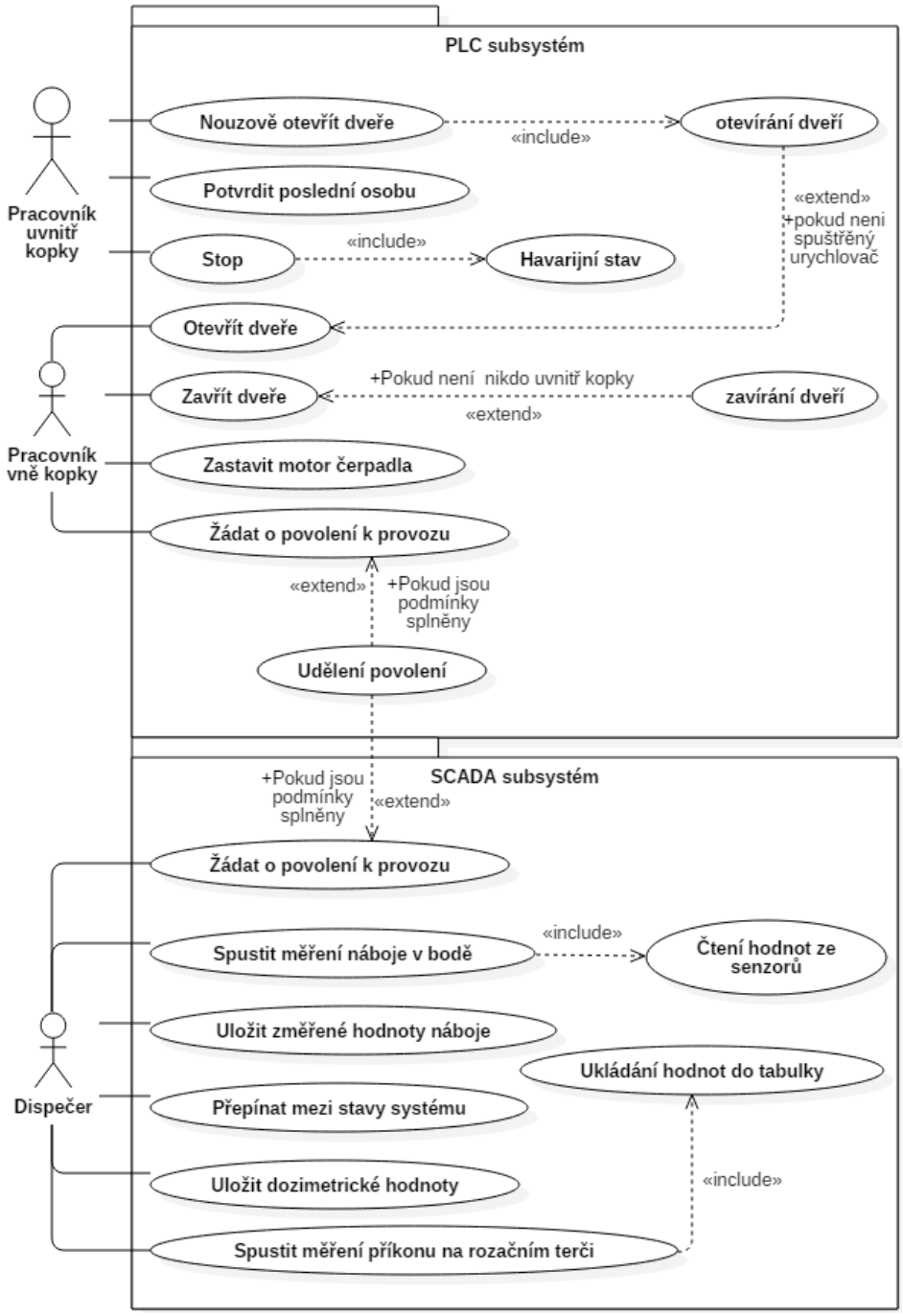
Pro základní pochopení chování systému z pohledu uživatele či pracovníka, který chce použít systém k jeho cílům, se nejlépe jeví UML diagram případu užití (často označováno jako use case diagram). Jedná se o staticky behaviorální diagram systému, který je zobrazen na obrázku č. 8.

Ve výše zmiňovaném diagramu vystupují dohromady tři aktéři. Každý z aktérů symbolizuje pracovníka na jiném místě. Projekt je rozdělen na dva subsystémy. Subsystémy mohou být definované jako skupiny tříd, asociací, operací, událostí a omezení, které jsou vzájemně propojené a mají dobře definované rozhraní. Neexistuje jasně daná hranice o velikosti či množství jednotlivých subsystémů. Vzájemný vztah mezi subsystémy může být charakterizovaný dvěma způsoby:

– klient-server (client-server)

– klient-klient (peer-to-peer)

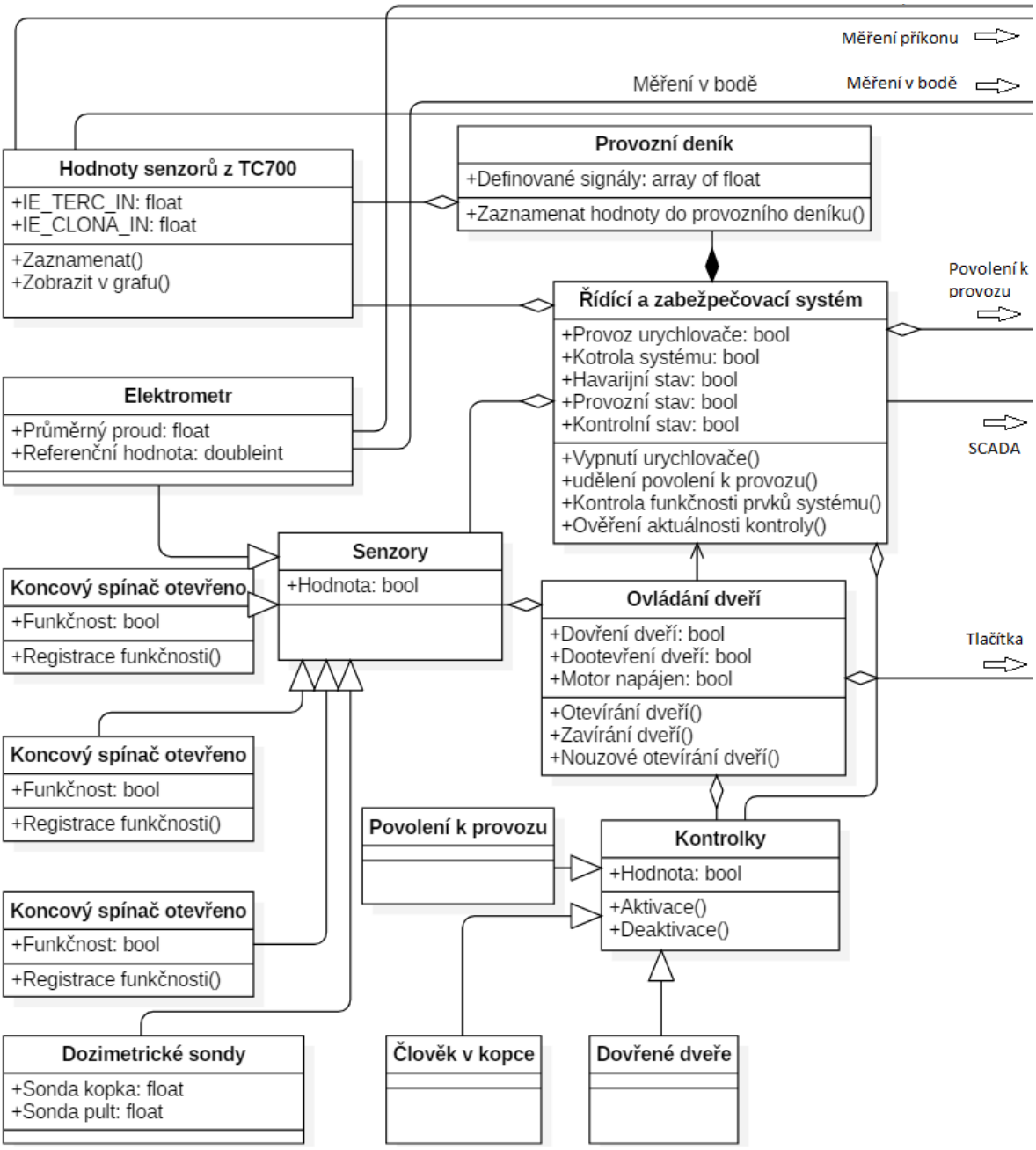
V tomto případě jsem systém rozdělil na základě vztahu klient-server. Klientem je SCADA subsystém a serverem je PLC subsystém. [7]



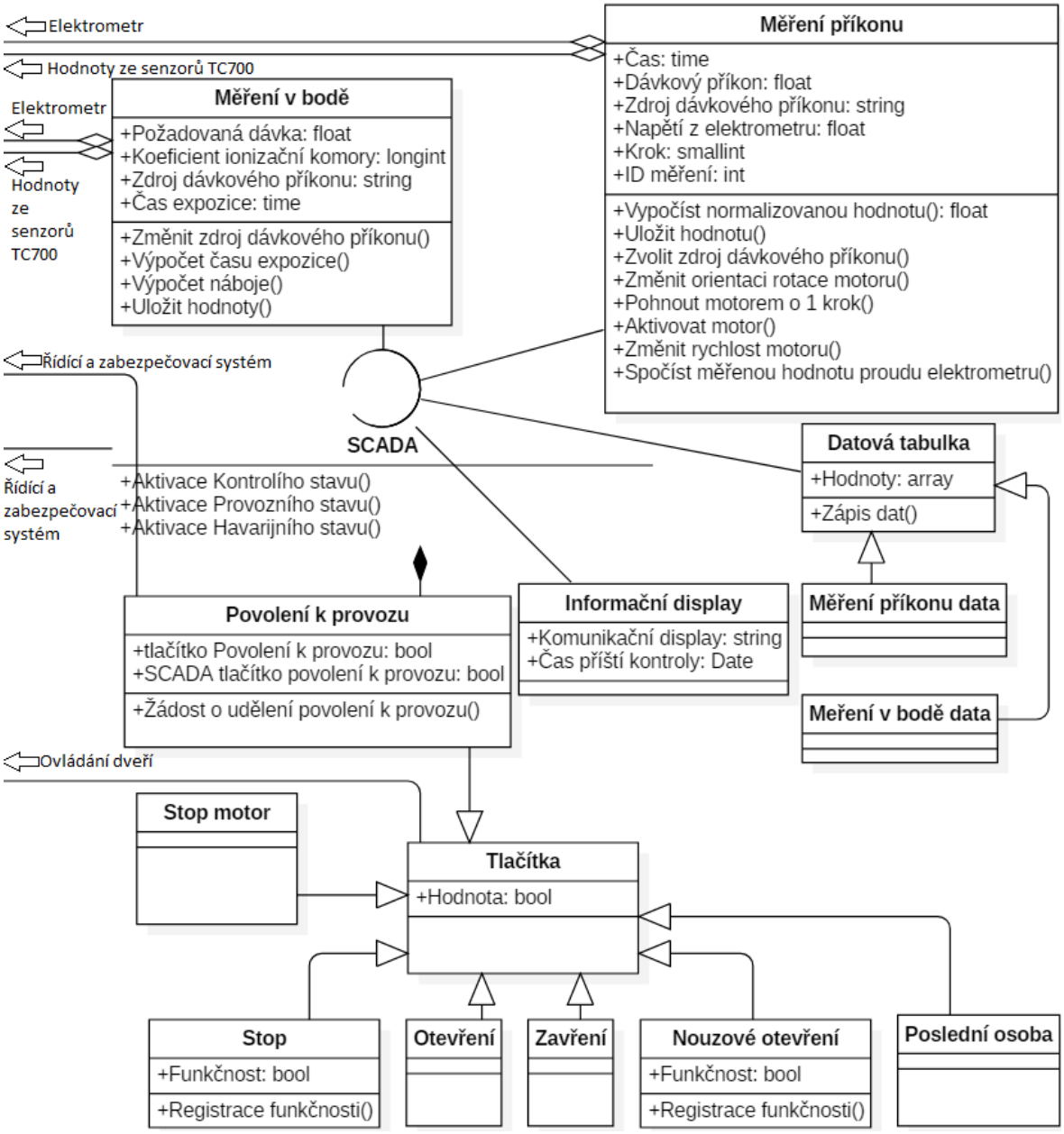
Obr. 8: Diagram případu užití

Jako celkový výstup lexikální analýzy je na obrázku č. 9 a obrázku č. 10 třídový diagram, ve kterém jsou zahrnuty i doplňující požadavky na fungování systému, které vznikaly v průběhu vykonávání mé diplomové práce ze strany zaměstnanců

mikrotronové laboratoře. Z důvodu velikosti diagramu a nemožnosti použít větší formát stránky jsem diagram rozdělil na dvě části a vazby vedoucí na jinou část diagramu jsou označeny šipkami se jménem třídy, ke které vedou.

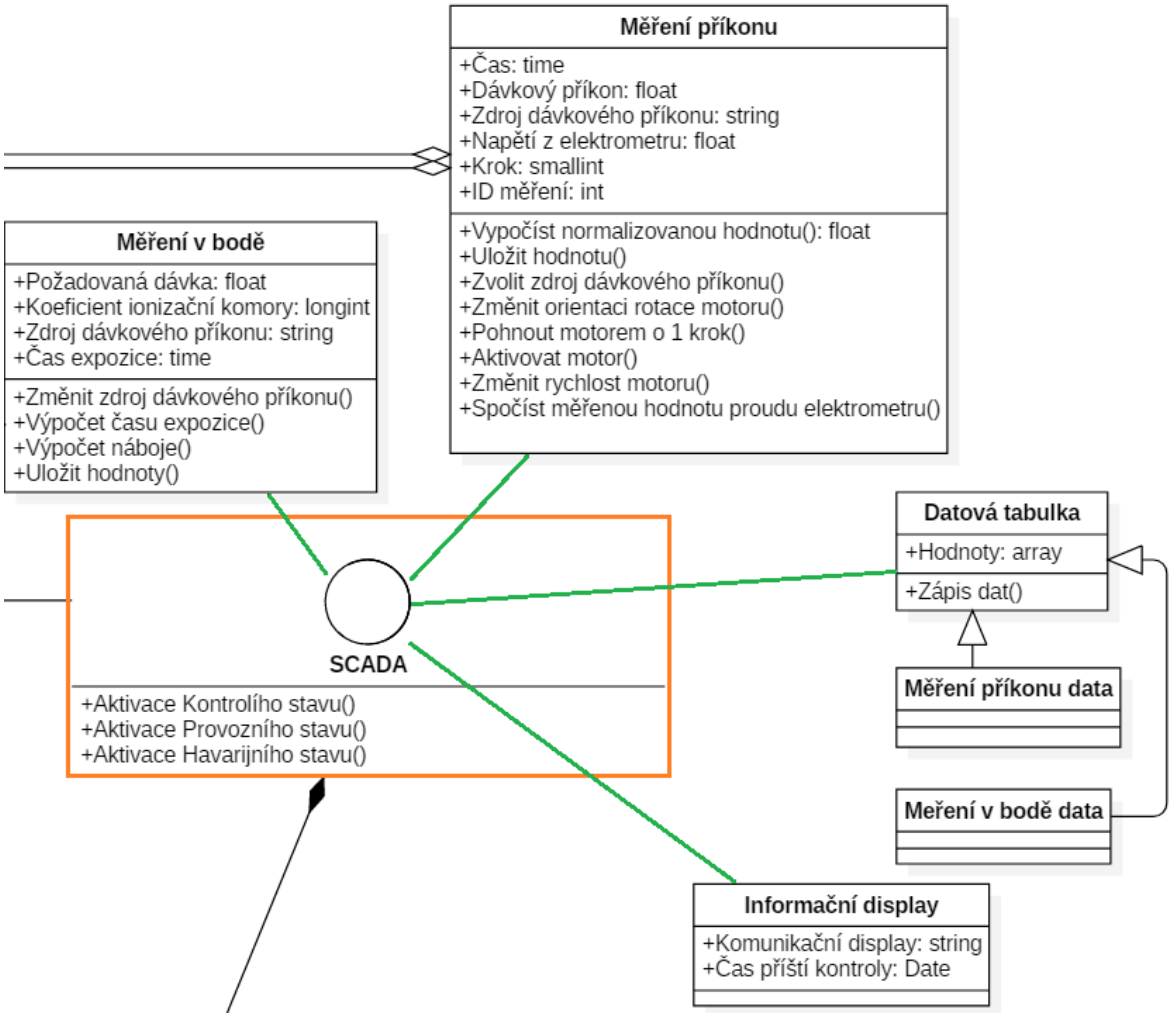


Obr. 9: Třídový diagram 1. část.



Obr. 10: Třídový diagram 2. část.

Program StarUML, ve kterém vznikly diagramy použité v této práci, má speciální typ třídy nesoucí název“interface class". Vzhledem k tomu, že se nejedná o základní typ třídy, je vhodné zdůraznit o jakou třídu se jedná. Na obrázku č. 11 je tato třída v oranžovém rámečku a zeleně jsou zvýrazněny vazby závislosti. Tento speciální typ třídy jsem použil pro třídu SCADA, protože SCADA tvoří HMI celého systému, tudíž se jedná o třída typu interface.



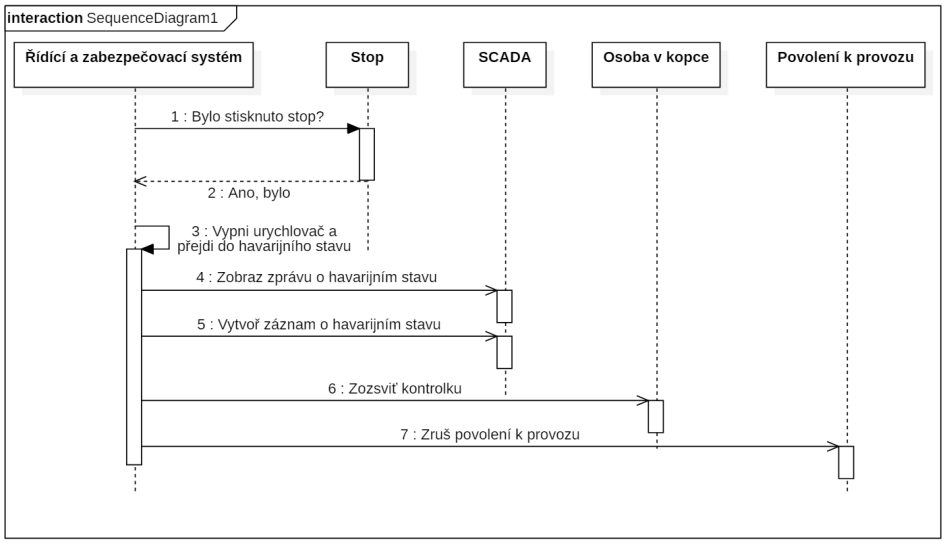
Obr. 11: Interface třída SCADA

4.1.2 Dynamický model

Dynamický model popisuje dočasný, behaviorální a řídící aspekt systému. Systém je v této podkapitole popsán UML diagramy. [7]

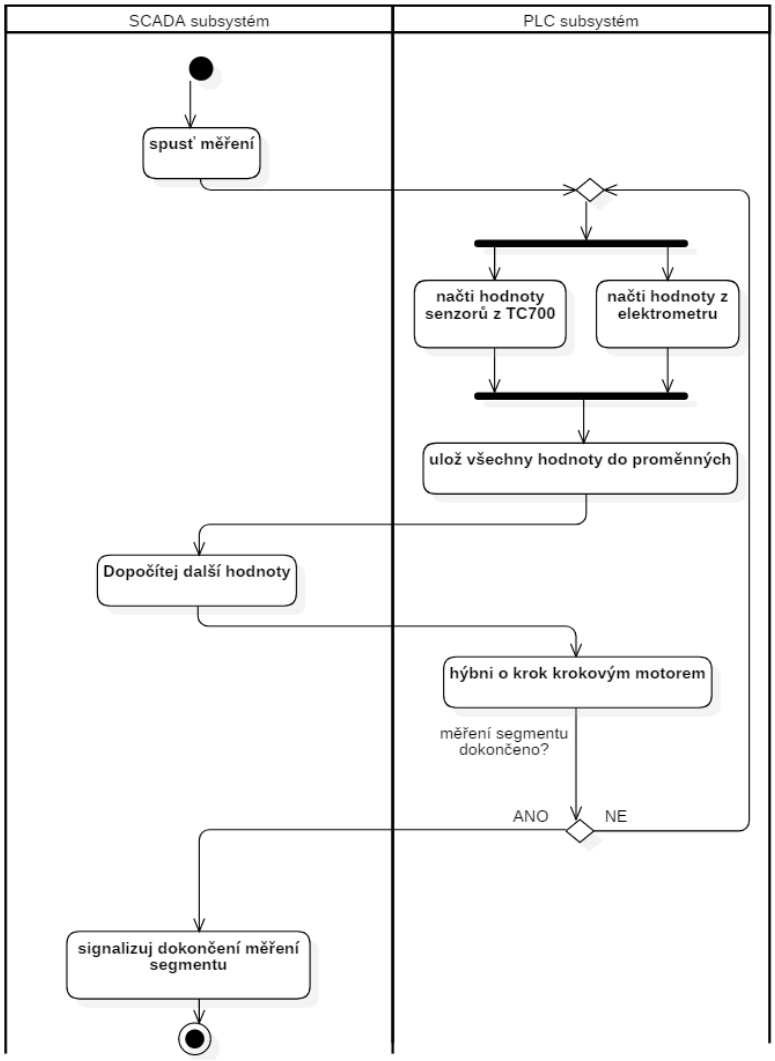
Tyto scénáře jsou nejlépe zachyceny pomocí sekvenčního diagramu. Na obrázku č. 12 je zobrazen sekvenční diagram, který zachycuje sekvenci úkonů, které se stanou, pokud bylo stisknuto jedno ze stop tlačítek. Pokud došlo ke stisknutí tlačítka, znamená to, že se v kobce vyskytuje člověk a tento mechanismus zajistí bezpečné vypnutí či zabránění spuštění urychlovače. Systém disponuje více pojistkami, které by této situaci měly předejít. Tato sekvence slouží jako poslední bezpečnostní pojistka.
Sekvenční diagram vyjadřuje interakci mezi třídami a slouží také jako dokumentace pro chování v rámci případu užití (use case). Reprezentuje podrobnější interakci mezi aktérem a systémem, nebo mezi kolaborujícími objekty v rámci daného časového bloku. Na rozdíl od diagramu aktivit hraje v sekvenčním diagramu větší roli časová posloupnost, díky čemuž je považován za dynamicky-behaviorální diagram UML.

Interakce mezi třídami je realizována pomocí zpráv, které jsou poslány mezi tzv. čárami života ("lifelines") daných objektů. Důležitou roli hraje časová posloupnost těchto zpráv. Čáry života tedy reprezentují časový tok.
Šipky mezi čárami života reprezentují dané zprávy a tyto zprávy mohou být různého typu. Zprávy mohou být synchronní a asynchronní. U asynchronních zpráv (značené klasickou šipkou) vysílající objekt nečeká na odpověď respondenta. Synchronní zprávy (značené šipkou zakončenou vyplněným trojúhelníkem) naopak nečekají na odpověď. Obdélník ležící na čáře života značí aktivitu třídy. [9]



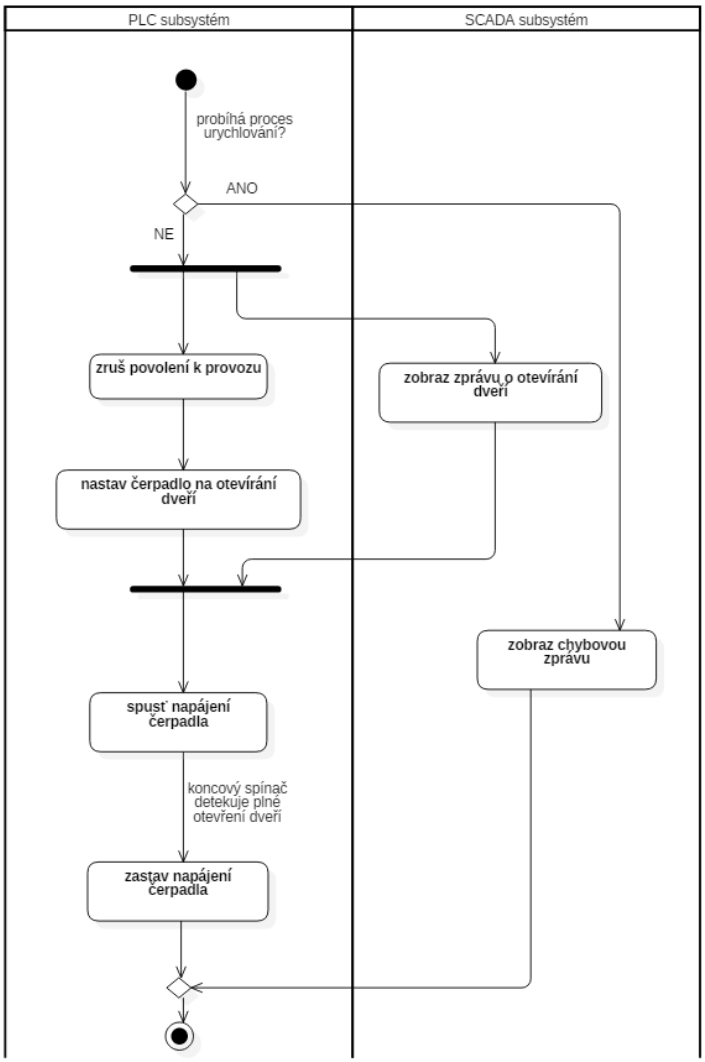
Obr. 12: Sekvenční diagram pro stisknutí havarijního stop tlačítka

Na obrázku č. 13 je znázorněný aktivity diagram pro měření příkonu. Tato aplikace vyžadovala otáčet krokovým motorem a zaznamenávat přitom sadu informací, přičemž bylo nutné, aby každá sada hodnot z různých senzorů byla zachycena v co nejmenším časovém úseku.



Obr. 13: Aktivity diagram pro měření příkonu na rotačním terči

Na obrázku č. 14 je aktivity diagram, který zobrazuje procedurální logiku otevírání dveří.

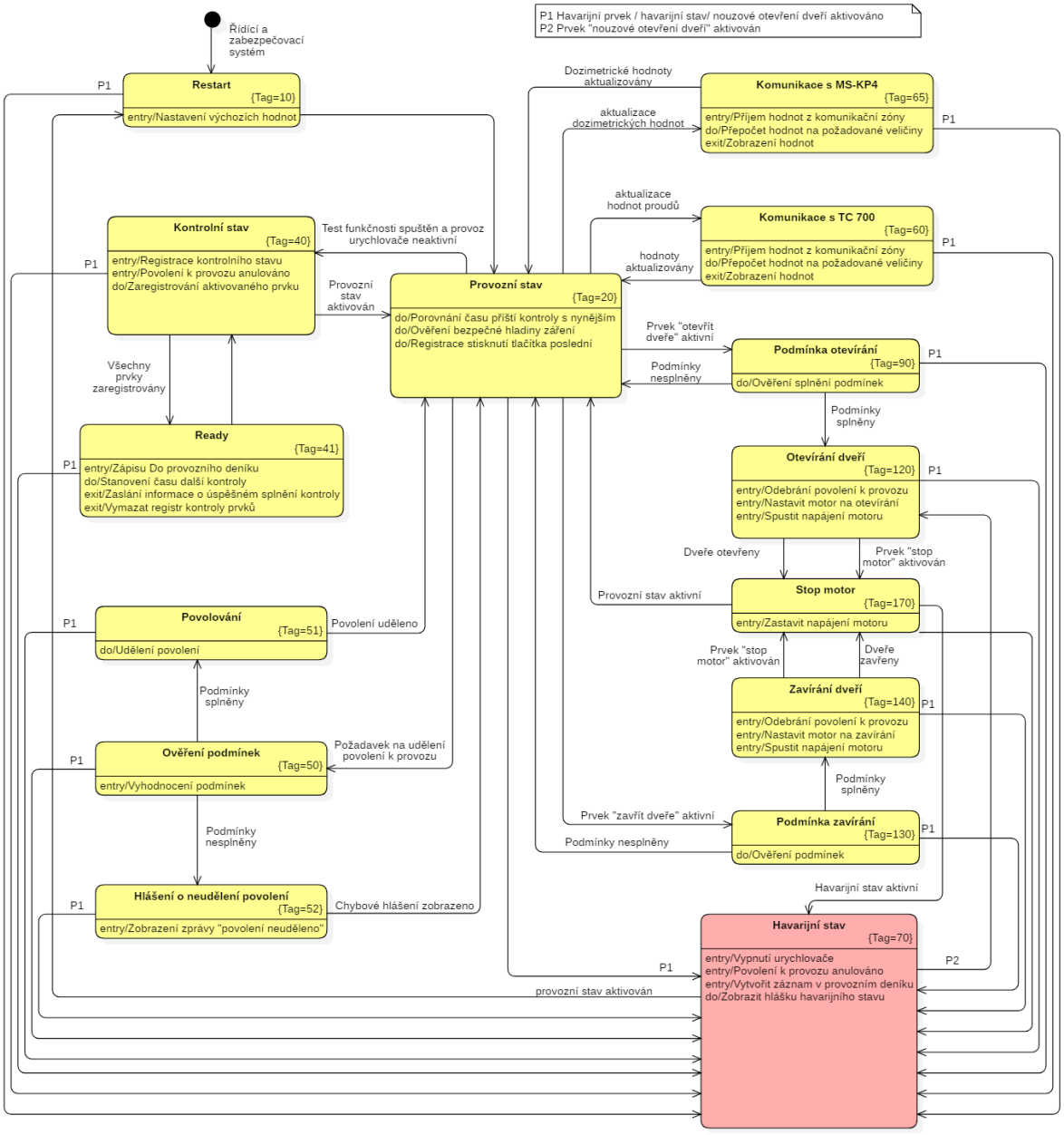


Obr. 14: Aktivity diagram otevření dveří

Na obrázku č. 15 je zobrazený stavový diagram, který nejlépe vystihuje dynamické chování třídy řídící a zabezpečovací systém.

Stavový diagram (anglicky“state machine diagram") popisuje chování objektů jako stavy, ve kterých by se mohl nacházet. Přechody mezi stavy charakterizuje chování, kterým systém reaguje na akce. Stavový diagram obsahuje všechny možné stavy, ve kterých se daný objekt může nacházet. Stavový diagram je dynamicky behaviorální diagram. V diagramu je tudíž zahrnuta časová složka a diagram je ideální pro modelování systému v reálném čase.

Vzhledem k implementaci tohoto systému je tento diagram velice užitečný. V kapitole detail design (4.2) bude uveden důvod tohoto tvrzení.



Obr. 15: Stavový diagram

4.1.3 Funkční model

Funkční model, někdy označovaný jako procesní model, popisuje transformační a funkční aspekt systému. Zachycuje procesy a činnosti zpracovávající data v systému. [7]

Nejprve je třeba sestavit tabulku vstupních a výstupních hodnot systému. Tabulka č. 23 obsahuje vstupní proměnné systému. První sloupec obsahuje názvy proměnných v programu a druhý sloupec pro přehlednost popisuje funkci dané proměnné. Obdobně vypadá tabulka č. 24 pro výstupy systému.

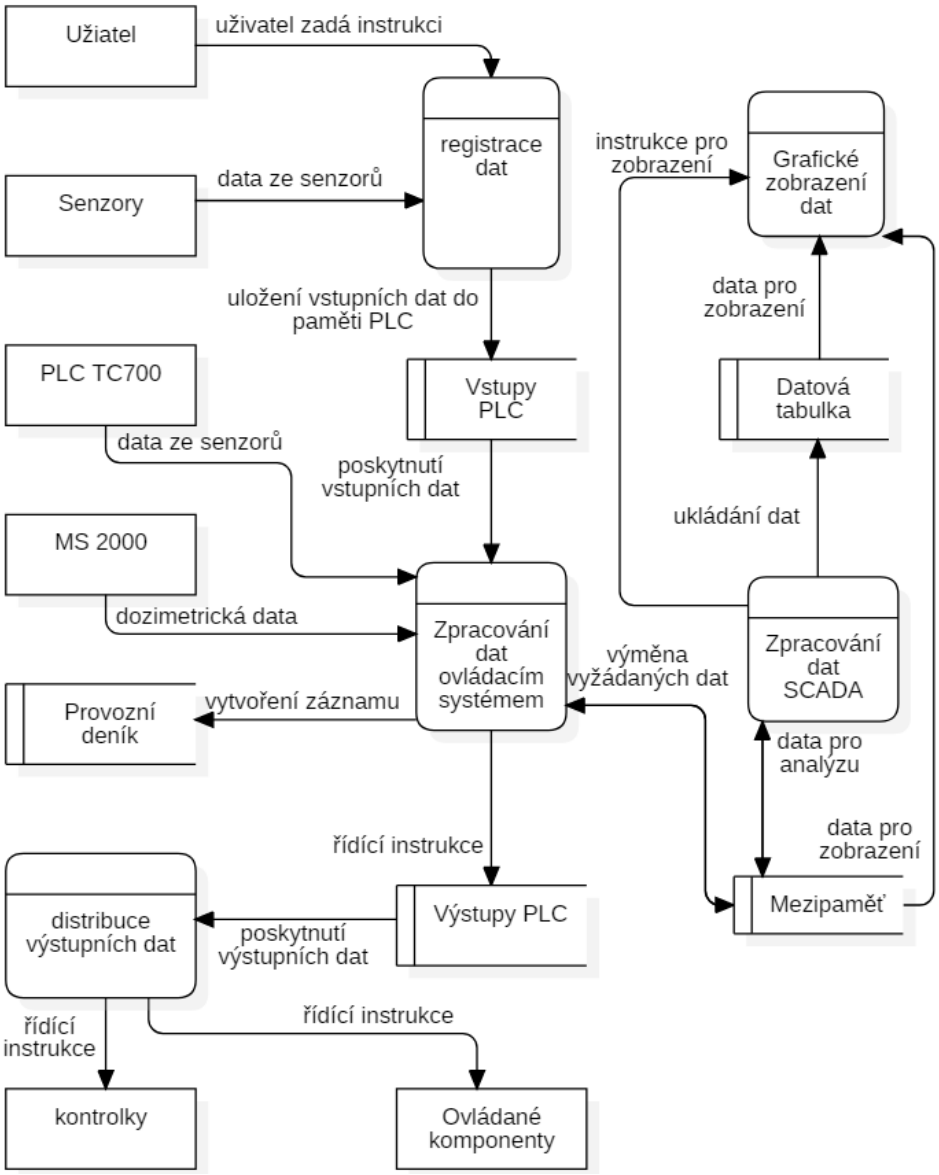
Tab. 23: Vstupy systému

Název proměnné v programu	**Popis funkce proměnné**
otevření	tlačítko pro standardní otevření dveří
zavření	tlačítko pro zavření dveří
poslední osoba	tlačítko dávající signál o přítomnosti osoby v kobce
stop motor	tlačítko dávající instrukci zastavení čerpadla
koncový spínač otevřeno	informace o otevření dveří
koncový spínač zavřeno 1	informace o zavření prvních dveří
koncový spínač zavřeno 2	informace o zavření druhých dveří
povolení k provozu (tlačítko)	tlačítko pro žádost o povolení k provozu
stop 1	první tlačítko stop
stop 2	druhé tlačítko stop
nouzové otevření	tlačítko pro nouzové otevření dveří
provoz urychlovače	signál, zda probíhá urychlování
průměrný proud	signál z elektrometru
TC700 zóna	data z druhého PLC (TC700)
DozimetrZona	data z dozimetrické sondy

Tab. 24: Výstupy systému

Název proměnné v programu	**Popis funkfce proměnné**
motor napájení	signál pro napájení čerpadla
motor zavíraní	signál pro ovládání smyslu čerpání čerpadla
motor otevíraní	signál pro ovládání smyslu čerpání čerpadla
povoleni k provozu	signál k udělení povolení k provozu
zavřené dveře kontrolka	kontrolka pro zavřené dveře
člověk v kobce kontrolka	kontrolka pro signalizaci osoby v kobce
zvuková signalizace	zvuková signalizace havarijního režimu
vypnutí urychlovače	signál pro vypnutí urychlovače
enable	signál enable pro krokový motor
dir	signál udávající smysl rotace krokového motoru
step	signál pro krokování krokového motoru
ena	signál pro zapnutí 1. bloku tranzistorů na PLC
enb	signál pro zapnutí 2. bloku tranzistorů na PLC

Na obrázku č. 16 je zobrazený diagram datových toků systému.



Obr. 16: Diagram datových toků (Data flow diagram)

Diagram toku dat (anglicky“data flow diagram"nebo DFD) je jeden z nejpoužívanějších nástrojů pro vytváření funkčních modelů informačních systémů. Diagram velmi dobře vyobrazuje návaznosti jednotlivých činností, udává zpracovatele těchto činností, zejména datové vstupy a výstupy obsažené v systému. Téměř není možné v tomto diagramu zachytit rozhodovací procesy. Ve větších informačních systémech se tento diagram obvykle vytváří ve více stupních detailnosti. Nejprve se kreslí jako celek a poté se rozkreslují detailněji jeho součásti.

Při jeho tvorbě je nutné řídit se několika pravidly. Diagram musí mít nutně vstupy i výstupy. Datové toky (značené šipkami, které mají směr podle směru datového toku) z externích entit (zobrazené v diagramu jako obdélník s ostrými rohy) vždy musí procházet přes proces. Externí entitou je myšlen objekt ve vnějším okolí systému

komunikující s procesy. Často jím bývá uživatel systému, senzory, či jiné externí přístroje či systémy. Správně by neměly být přímo propojeny datovými toky ani dva procesy (graficky zobrazené jako obdélníky s oblými rohy), neboť transport by měl vždy procházet přes jejich uložení v paměti. Toto pravidlo se však reálně moc neaplikuje protože vede k znepřehledňování celého diagramu. Jako příklad mohu uvést na následujícím DFD procesy“zpracování dat ovládacím systémem"a“zpracování dat SCADA", které si vyměňují data standardně jako SCADA systém a PLC. SCADA systém si reálně všechna data z PLC ukládá do mezipaměti, se kterou poté operuje. Paměť či úložiště se potrom značí jako upravený obdélník bez jedné strany. Grafické reprezentace se liší autory notací. [10]

Následující tabulka č. 25 obsahuje popis funkcí zmíněného DFD.

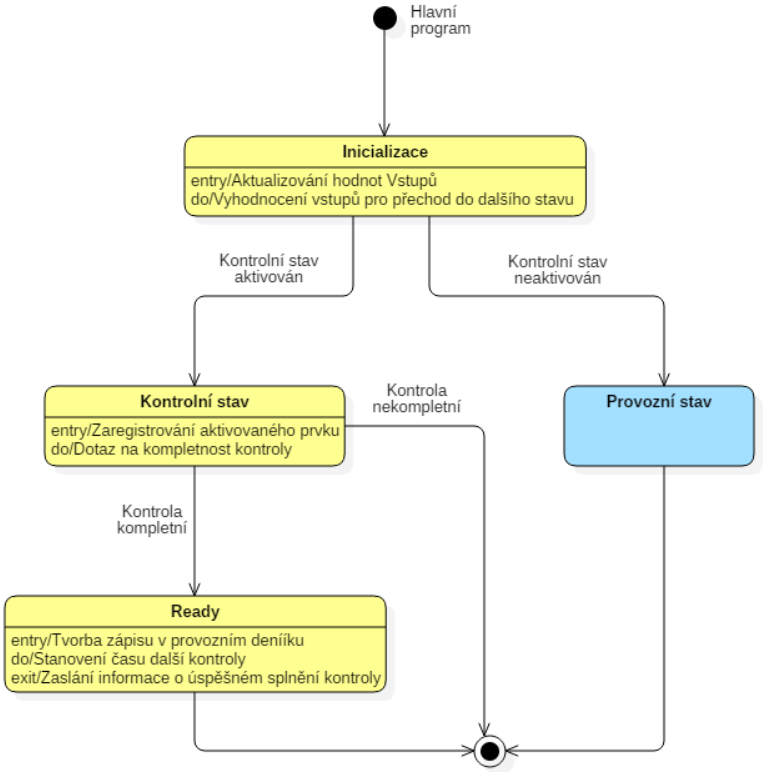
Tab. 25: funkce DFD

Název funkce v DFD	**Popis**
Registrace dat	vkládá do mezipaměti PLC data ze vstupů
Zpracování dat ovládacím systémem	symbolizuje procesy vykonávané řídícím a zabezpečujícím systémem
Zpracování dat SCADA	symbolizuje úkony vykonávané SCADA systémem
Distribuce výstupních dat	dle přijetých dat změní hodnoty na výstupech
Grafické zobrazení dat	SCADA systém zobrazuje grafy z datových zabulek a přímo zobrazuje některé aktuální hodnoty vstupů PLC

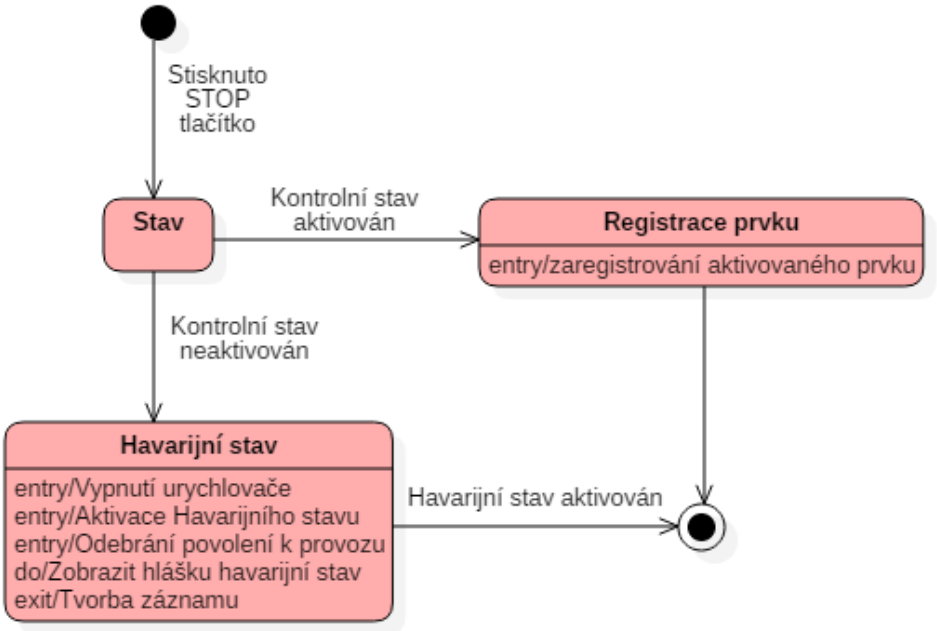
4.2 Detail design

Obrázek č. 17 zobrazuje velice zjednodušený prvotní návrh stavového diagramu. Vzhledem k tomu, že jsem u PLC kladl vysoký důraz na rychlost odezvy, tato varianta řešení se jevila jako neefektivní. Program by v tomto případě musel projít danou "větev" stavů každý cyklus, což by snižovalo rychlost reakce na případné změny na vstupech.

V popisu systému je uvedeno, že při stisknutí jakéhokoliv ze "stop" tlačítek musí nastat havarijní režim, kdy se musí bezprostředně po stisknutí urychlovač vypnout. To znamená, že reakce musí být rychlá z důvodu bezpečnosti personálu. Z toho důvodu se jevilo jako optimální použití přerušení od vstupů. Na obrázku č. 18 je zobrazen diagram znázorňující přerušení. Bohužel jsem po nějaké době zjistil, že PLC, které mi bylo poskytnuto, nedisponuje modulem přerušení od vstupů ale pouze časovým přerušením. Z toho důvodu jsem musel vymyslet a vypracovat jiný diagram. [11]



Obr. 17: Prvotní návrh stavového diagramu



Obr. 18: Přerušení

Program byl napsaný v prostředí Mosaic, o kterém bude více informací v kapitole č. 4.2.2. Personálem urychlovače jsem byl instruován k použití jazyku ST

(Structured text), který je podporovaný normou IEC 61131-3 a řadí se mezi "vyšší" programovací jazyky. Z toho důvodu se jevilo velice výhodné použít stavový diagram pro strukturování programu. Uživatelské rozhraní bylo vytvořeno v prostředí SW Reliance 4, kde je možné využít skriptovací jazyk Visual basic script.

4.2.1 Reliance 4

Reliance je profesionální SCADA/HMI systém určený pro monitorování a ovládání nejrůznějších průmyslových technologií a automatizaci budov. Data jsou získávána z řídicích nebo telemetrických systémů, ukládána do databází a prezentována koncovým uživatelům grafickou formou (schémata, grafy, tabulky atd.).

4.2.2 Mosaic

Mosaic je vývojové prostředí pro tvorbu a ladění programů pro programovatelné logické systémy (PLC, Programmable Logic Controller) TECOMAT ® a TECOREG ® z produkce firmy Teco a.s. Kolín. Program Mosaic je prodáván od roku 2000. Prostředí je vyvíjeno ve shodě s mezinárodní normou IEC EN-61131-3, která definuje strukturu programů a programovací jazyky pro PLC.

Jazyk ST disponuje mimo hlavní programovou jednotku také struktury funkční blok a funkce.

- **Funkce (Function)** Funkce je programová organizační jednotka, která po provedení vygeneruje vždy jeden datový element. Volání funkce se může použít v textových jazycích jako operand ve výrazu

  V jazyce ST lze funkci vyvolat zápisem jména funkce následovaným předávanými parametry v kulatých závorkách. Počet a datový typ předávaných parametrů musí odpovídat vstupním proměnným v definici funkce. Příloha C a obsahuje zdrojový kód funkce, která v závislosti na vstupních parametrech mění stav systému na jeden ze tří základních (provozní, kontrolní a havarijní).

- **Funkční blok (Function block)** Funkční blok je organizační jednotka programu, která po provedení vygeneruje jednu nebo více hodnot. Z funkčních bloků se dají vytvářet násobné pojmenované instance (kopie). Příloha A obsahuje funkční blok objektu Ovládání dveří.

4.2.3 Popis programu subsystému PLC

První částí programu je definice proměnných. Velice záleží, zda bude proměnná typu global (globální) a public (veřejné) . Aby byla možná komunikace se SCADA

systémem, je nutné, aby byly všechny proměnné typu global a public. Toto platí jen pro proměnné, ke kterým chceme, aby měly přístup oba subsystémy.

Hlavní programová jednotka je psána pomocí rozhodovacího příkazu CASE. Za dobu jednoho cyklu může být program pouze v jednom stavu zároveň. Ještě před příkazem CASE je část programu, která se vykonává vždy, nehledě na stav systému. Jsou zde například všechny časovace a funkce detekující náběžné a sestupné hrany proměnných. Také detekuje zakázané stavy (ve smyslu kombinací proměnných), které signalizují nestandardní situaci. Taková situace může nastat například při stisknutí tlačítka stop, kdy nastane havarijní režim a systém nehledě na momentální stav přejde do havarijního stavu. Tato část programuje uvedena zde:

```
vypnutiTOF (IN:=havarijni_stavR_TRIG.Q, PT:=T#1s,
Q=> vypnuti_urychlovace); //vypnutí urychlovače
 if stop1 or stop2 then
 //pokud bylo stisknut havarijní prvek
 if stav=40 then
 //pokud byl stisknutý v rámci kontroly
 if stop1 then
 kontrola_stop1:=true;
 end_if;
 if stop2 then
 kontrola_stop2:=true;
 end_if;
 else //pokud nef
 if posledni_pamet then
 prechod(havarijni:=true);
 end_if;
 end_if;
 end_if;
 if havarijni_stavR_TRIG.q then //přechod do havajijího stavu
 prechod(havarijni:=true); //stav => havarijní
 end_if;

```

Ve stavovám diagramu (obrázek č. 15 má každý stav číslo (označený slovem Tag), které koresponduje se stavy v zdrojovém kódu programu (zdrojový kód obsahuje Příloha A) Tato struktura má limity primárně v tom, že každý přechod trvá minimálně jeden cyklus PLC. Zde se pokusím popsat jednotlivé stavy:

- stav 0: inicializace (Příloha A, řádek 102-132)
  Tento stav nastane při studeném restartu PLC a při přechodu z havarijního stavu do provozního. Nastaví se výchozí hodnoty pro správný běh systému.
  Na konci tohoto stavu systém přejde do provozního stavu.
- stav 20: Provozní stav (příloha A, řádek 133-246)

V tomto stavu by měl být systém více než 99% veškerého času. Následující segment kódu má na starosti obsluhu krokového motoru. Rychlost krokového motoru se reguluje nastavením počtu cyklů PLC, než se vykoná jeden mikrokrok.

```
r_mereni(CLK:=rotacni_mereni);
r_mereni_not(CLK:=(not rotacni_mereni));
if r_mereni_not.q then
// pokud byl ukončený pokus měření na rotačním terči
  pokracovani:=false;
  krok:=0; //vynuluje se krok
end_if;
if r_mereni.q then
//nové měření na rotačním terči
  enable_pom:=true;
  ena_pom:=true;
  enb_pom:=true;
  cteni_dat:=false;
//vynulování hodnot
  k:=0;
  krok:=0;
//nové číslo měření
  ID_mereni:=ID_mereni+1;
  pokracovani:=true;
end_if;

if rotacni_mereni then
//pokud je měření zaplé
  step_pom:=true;
  r_pokr_rot(CLK:=pokracovani);
  if r_pokr_rot.q then
//pokud bylo stisknuto tlačítko
//pro měření dalšího segmentu
  if rotace=false then //nastavení smyslu rotace
    cil_krok:=krok+kroky_usek;
    dir_pom:=true;
  else
    cil_krok:=krok-kroky_usek;
    dir_pom:=false;
  end_if;

end_if;

if krok<cil_krok and (not rotace) then
//točim po měru hodinových ručiček
  k:=k+1;  //pomocná proměná pro regulaci
//rychlosti rotace krokového motoru
  if k>=rychlost_krokovani then
```

```
44      k:=0;
45 //ukládání měřených hodnot do globálních proměnných
46      krok_pom:=krok;
47      IS_IN_pom:=IS_IN;
48      IE_TERC_IN_pom:=IE_TERC_IN;
49      prumerny_proud_pom:=prumerny_proud;
50      krok:=krok+1;
51      step_pom:=false;
52      cteni_dat:=true;
53      end_if;
54    end_if;
55  //točim proti smětu hod. ručiček
56  if krok>cil_krok and rotace then
57    k:=k+1;
58    if k>=rychlost_krokovani then
59      k:=0;
60      krok_pom:=krok;
61      IS_IN_pom:=IS_IN;
62      IE_TERC_IN_pom:=IE_TERC_IN;
63      prumerny_proud_pom:=prumerny_proud;
64      krok:=krok-1;
65     step_pom:=false;
66     cteni_dat:=true;
67    end_if;
68  end_if;
69  if krok=cil_krok then
70          pokracovani:=false;
71  end_if;
72  end_if;
73
74 if posledni then //pokud bylo stisknuto poslední
75          posledni_pamet:=true;
76 end_if;
77
78  if kontrola <= dnesni_datum then //zda je kontrola aktuální
79    nutnost_kontroly:=1; // pokud není
80  end_if;
81
82  if aktualizace then
83  //aktualizace hodnot sond (IE Kanál, ..)
84   stav:=60; //přechod do stavu 60
85    pomocna_promena:=false;
86  end_if;
87
88  if aktualizace_dozi then //aktualizace dozimetrických hodnot
89    stav:=65; //přechod do stavu 65
90    pomocna_promena_3:=false;
```

```
91  end_if;
92
93  if kontrolni_stav then
94   //pokud byl z dispečerského stanoviště aktivován kontrolní stav
95    if provoz_urychlovace then
96      //kontrolu nelze spustit za běhu urychlovače
97      prechod(provozni:=true); // přechod do provozního stavu
98    else //pokud není urychlovač spuštěný
99      prechod(kontrolni:=true); //přechod do kontrolního stavu
100    //na počátku se vymaže paměť z předchozí kontroly
101     test_kompletni:=false;
102     kontrola_stop1:=false;
103       kontrola_stop2:=false;
104     kontrola_nouzove_otevreni:=false;
105     kontrola_koncovy_spinac1:=false;
106     kontrola_koncovy_spinac2:=false;
107   end_if;
108  end_if;
109
110  if (otevreni or otevreni_virtual) then
111   //pokud bylo stisknuto tlačítko na otevření dveří
112   stav:=90; //přechod do stavu 90
113  end_if;
114
115  if zavreni and (not kontrolni_stav) and not
116  ( koncovy_spinac1_zavreno and koncovy_spinac2_zavreno) then
117            //pokud bylo stisknuto tlačítko na zavření dveří
118            //a pokud ještě není zavřeno
119     stav:=130; //přechod do stavu 130
120  end_if;
121
122  if povoleni_k_provozu_virtual or povoleni_k_provozu_tlacitko then
123 //pokud někdo žádá o povolení k provozu
124     stav:=50; // přechod do stavu 50
125     povoleni_k_provozu_virtual:=false;
126  end_if;
127
```

– stav 40: Kontrolní stav (příloha A, řádek 250-295)
  V této čísti programu dochází k testování, zda jsou funkční havarijní prvky. Aktivované prvky se zapisují do mezipaměti, která se smaže příštím zapnutím kontroly. Pokud byly úspěšně aktivovány všechny havarijní prvky, systém přejde do stavu 41.

– stav 41: Ready (příloha A, řádek 295-309)
  Tento stav nastane pouze pokud je kontrola úspěšná

– stav 50: Ověření podmínek (příloha A, řádek 309-318)

Pokud jsou dveře dovřené, není v kobce žádná osoba a urychlovač už není v provozu, tak nastane stav č. 51, v opačném případě stav č. 52.

```
 if (koncovy_spinac1_zavreno) and (koncovy_spinac2_zavreno)
and posledni_pamet and (not provoz_urychlovace) then
   stav:=51;
 else
  stav:=52;
 end_if;

```

– stav 51: Povolení (příloha A, řádek 318-325)

```
\begin{lstlisting}[language=C++]
  povoleni_k_provozu_pom:=true; //povolení k provozu
  prechod(provozni:=true); //přechod do provozního stavu
  zprava := 'Povolení k provozu uděleno.';
  zaznamenanidozi:=1; //záznam hodnot do provozního deníku

```

– 52: // povolení neuděleno (příloha A, řádek 325-332)
Pokud nejsou splněny podmínky, zobrazí se zpráva dispečerovi a systém přejde do provozního stavu.

```
 prechod(provozni:=true);
 zprava := 'Povolení k provozu neuděleno, podmínky udělení nejsou splněny'

```

– 60: //Komunikace s TC700 (příloha A, řádek 325-332)
Jednou za 0.5 sekundy dochází k aktualizaci hodnot z indukčních snímačů, k tomuto účelu se pro každou proměnnou volá funkce, která bude rozepsaná v kapitole 4.2.3. Proměnná "TC700zona[..]" je pole hodnot formátu usint (0 až 255). Pole je výstup z komunikační zóny TC700zona.

```
 stav := 20;
 IE_KAnal_IN:=prevod_TC700(R0:=TC700zona[0], R1:=TC700zona[1],
      R2:=TC700zona[2], R3:=TC700zona[3]);
 IE_TERC_IN:=prevod_TC700(R0:=TC700zona[4], R1:=TC700zona[5],
      R2:=TC700zona[6], R3:=TC700zona[7]);
 IE_CLONA_IN:=prevod_TC700(R0:=TC700zona[8], R1:=TC700zona[9],
      R2:=TC700zona[10], R3:=TC700zona[11]);
 IS_IN:=prevod_TC700(R0:=TC700zona[12], R1:=TC700zona[13],
      R2:=TC700zona[14], R3:=TC700zona[15]);
 if plcerror>0 then error_komunikace:=true;
 else error_komunikace:=false;
 end_if;
```

```
13
```

– 65: Kontrolní panel radiační monitorovací sítě (příloha A, řádek 325-332)
Jednou za 3 sekundy dochází k aktualizaci hodnot dozimetrických sond. K tomuto účelu se pro každou proměnnou volá funkce, která bude rozepsaná v kapitole 4.2.3. Proměnná "DozimetrZona[..]" je pole hodnot formátu usint (0 až 255). Pole je výstup z komunikační zóny DozimetrZona.

```
1  if DozimetrZona[3]=72 then
2  dozi_pult:=prevod_dozi(H:=DozimetrZona[6], L:=DozimetrZona[7]);
3  dozi_kopka:=prevod_dozi(H:=DozimetrZona[10], L:=DozimetrZona[11]);
4  end_if;
5  if unierror>0 then error_komunikace_dozi:=true;
6  else error_komunikace_dozi:=false;
7  end_if;
8  stav:=20;
9
```

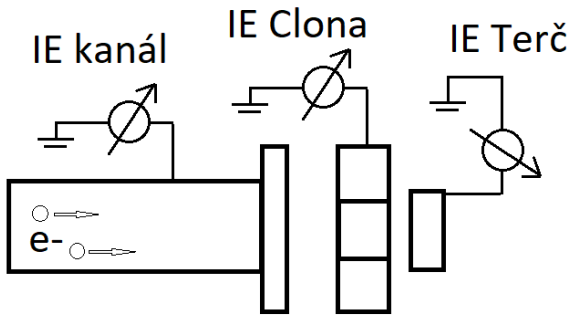
– 70: Havarijní stav (příloha A, řádek 325–332)
Do tohoto stavu se stav dostane při jakémkoliv ohrožení.

```
1  zvukova_signalizace_err:=true; //zvuková signalizace
2  zprava:='Havarijní stav aktivován, byl proveden zápis do
3         // provozního deníku.';
4  posledni_pamet:=false;
5  povoleni_k_provozu_pom:=false;
6
7  if kontrolni_stav then  //deaktivace hav. stavu
8    prechod(havarijni:=true);
9   zprava:='Přechod pouze do Provozního stavu';
10 end_if;
11
12 if provozni_stav then //deaktivace hav. stavu
13   prechod(inicializace:=true);
14  zprava:='Havarijní stav deaktivován';
15 end_if;
16
17 if otevreni or otevreni_virtual then
18  //pokud někdo chce otevřít dveře
19  stav:=100;
20 end_if;
21
22 if stop_motor then
23  //pokud někdo chce zastavit pohyb dveří
24  stav:=170;
25 end_if;
26
```

– 90: Povolení otevírání (příloha A, řádek 385–396)
  Pokud není urychlovač spuštěný, systém přejde do stavu 120 a dveře se začnou otevírat zavoláním funkce na ovládání dveří (příloha B). V opačném případě se zobrazí zpráva o neúspěšném pokusu otevřít dveře a systém přejde do provozního stavu.
– 120: Otevírání dveří (příloha A, řádek 398-415)
  Systém odebere povolení k provozu a anuluje platnost signalizace absence osoby v kobce. V případě, že je stisknuto tlačítko stop motor nebo koncový spínač detekuje plné otevření dveří, systém přejde do stavu 170.
– 130: Povolení zavírání (příloha A, řádek 415-428)
  Pokud bylo stisknuto tlačítko poslední osoba, voláním funkce na ovládání dveří (příloha B) se správnými parametry, se začnou zavírat dveře a systém přejde do stavu 140. V opačném případě systém přejde do provozního stavu a zobrazí zprávu o neúspěchu.
– 140: Zavírání (příloha A, řádek 428-439)
  Pokud se systém nachází v tomto stavu, setrvá v něm, dokud nejsou obě dveře zavřené. V případě, že je stisknuto tlačítko stop motor nebo koncové spínače detekují plné zavření dveří, systém přejde do stavu 170.
– 170: Stop motor (příloha A, řádek 439-448)

V tomto stavu dochází k přerušení napájení čerpadla, poté systém přejde do provozního stavu.

Komunikace s PLC TECOMAT TC700



Obr. 19: Senzory proudu. IE Kanál – proud urychlených elektronů zachycených na elektronovodu, IE Clona – proud urychlených elektronů zachycených na kolimátoru, IE Terč – proud urychlených elektronů procházející ozařovaným vzorkem

V Mikrotronové laboratoři se nachází PLC TECOMAT TC700, který slouží k

ovládání mikrotronu a měření jeho provozních parametrů. K tomuto PLC je připojeno mimo jiné několik ampérmetrů měřících proud urychlených elektronů, jak je znázorněno na obrázku č. 19. Ampérmetry detekují proud urychlených elektronů v klíčových místech urychlovače, což dává informaci množství urychlených elektronů.

Hodnoty napětí signálů "IE kanál", "IE Clona" a "IE Terč" jsou zasílány po fyzické vrstvě RS-485 do PLC, které programuji. Jsou zasílány ve speciálním formátu, který je zobrazen v tabulce č. 26.

Tab. 26: Posloupnost bytů

IE Kanál				IE Terč				IE Clona				IS IN			
a1	b1	c1	d1	a2	b2	c2	d2	a3	b3	c3	d3	a4	b4	c4	d4

Kde $a1$, $b1..d4$ jsou čísla formátu byte (hodnot 0..255). Pokud bychom každou čtveřici bytů binárně poskládali za sebe, vytvořilo by to číslo formátu integer (4 byty). Problém je v tom, že jazyk ST neumí takto binárně pracovat. Kdybychom převedli $a1$, $a2$ atd. na integer a potom tyto čísla sečetli, dostali bychom číslo maximální hodnoty 1020. Kdybychom tato čísla binárně zařadili za sebe, dostali bychom maximální číslo 2 147 483 647.

Abychom dostali správnou hodnotu, bylo nutné tyto operace provést poněkud složitěji. Integer může také nabývat záporných hodnot. Pokud je první bit 0 resp. 1, číslo bude kladné resp. záporné. Pro kladná čísla integer funguje jak jsme zvyklí. Ale pro záporná čísla je první bit 1 a pokud znegujeme ostatní bity, dostaneme záporné číslo, které je v absolutní hodnotě rovno kladnému číslu zvětšenému o jedna.

Tab. 27: Short integer

Dekadicky	**Binárně**															
32767	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
–32768	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Následující kód programu (jazyk ST) je funkční blok, který má účel přepočet 4 byteových čísel na číslo datového typu integer. Řádek číslo 54 až 57 obsahuje obdobný kód jako předchozí část, pouze pro záporná čísla. Pro změnu neguje zbylé bity (mimo prvního bitu nesoucí informaci o znaménku čísla) a od finálního čísla odečte číslo jedna.

```
1 FUNCTION prevod_TC700 : real
2  VAR_INPUT
3  R0, R1, R2, R3:usint; // 4 vstupní byty
4  END_VAR
5  VAR_IN_OUT
6  END_VAR
```

```
7  VAR
8  bitsplus: array[0..31] of bool; //pole 4 bytů
9  k:usint;
10 mantisa: uint;
11 zbytek : uint;
12 vysledek : udint;
13
14 END_VAR
15 VAR_TEMP
16 END_VAR
17
18 if R3<128 then //pokud je číslo kladné
19  mantisa:=R0;
20 for k:=0 to 7 do
21  //binárně zapsat číslo do pole
22  zbytek:=mantisa mod 2;
23  mantisa:=mantisa/2;
24  bitsplus[k]:=uint_to_bool(zbytek);
25 end_for;
26  mantisa:=R1;
27  for k:=8 to 15 do
28  //binárně zapsat číslo do pole
29  zbytek:=mantisa mod 2;
30  mantisa:=mantisa/2;
31  bitsplus[k]:=uint_to_bool(zbytek);
32 end_for;
33 mantisa:=R2;
34 for k:=16 to 23 do
35  //binárně zapsat číslo do pole
36  zbytek:=mantisa mod 2;
37  mantisa:=mantisa/2;
38  bitsplus[k]:=uint_to_bool(zbytek);
39 end_for;
40  mantisa:=R3;
41  for k:=24 to 31 do
42  //binárně zapsat číslo do pole
43  zbytek:=mantisa mod 2;
44  mantisa:=mantisa/2;
45  bitsplus[k]:=uint_to_bool(zbytek);
46 end_for;
47  //následně převést pole na integer kladné číslo
48  vysledek:= bool_to_udint(bitsplus[30])*1073741824
49  +bool_to_udint(bitsplus[29])*536870912
50  .
51  .
52  .
53   +bool_to_udint(bitsplus[3])*8
```

```
54     +bool_to_udint(bitsplus[2])*4
55    +bool_to_udint(bitsplus[1])*2
56    +bool_to_udint(bitsplus[0]);
57    prevod_TC700:=(udint_to_real(vysledek))/100;
58 else //totéž udělat s zápornými čísly
59    //bity negovat k výsledku přičíst 1
60    .
61    .
62    .
63    .
64 END_FUNCTION
```

Komunikační protokol EPSNET

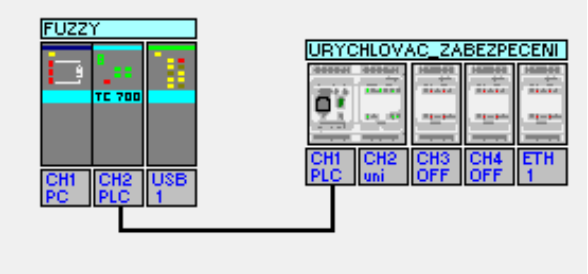
Přenos dat mezi TECOMAT Foxtrot CP-1003 a TECOMAT TC700 je realizován pomocí komunikačního protokolu EPSNET-F.

Komunikační kanál EPSNET-F slouží propojení více systémů TECOMAT a TECOREG za účelem rychlého vzájemného předávání dat. Komunikace probíhá protokolem EPSNET-F na sériovém rozhraní RS-485, které umožňuje vytvoření sítě. Ve všech systémech jsou vyhrazeny předávací zóny pro každého účastníka sítě na stejných registrech (předávací zóna dat z jednoho systému je vždy umístěna na stejných registrech ve všech ostatních systémech propojených do sítě). Výměnou za toto určité omezení je vysoká propustnost sítě, protože stejné rozmístění předávacích zón ve všech systémech dovoluje přenos jednoho bloku dat vždy do všech účastníků sítě zároveň. Tím dochází ke značné časové úspoře a nižším nárokům na strojový čas procesorů centrálních jednotek. Inicializace předávacích zón sítě je součástí uživatelských programů propojených systémů a musí být pro všechny systémy shodná (pokud jsou jednotlivé systémy součástí stejné skupiny projektů, požadavek zajistí vývojové prostředí Mosaic, jak tomu bylo i v tomto projektu). Příslušný komunikační kanál je třeba nastavit do režimu PLC, dále se nastavuje přenosová rychlost a adresa. [12]

Nastavování komunikace je tedy nutné provést jak na vysílací (TECOMAT TC700), tak na přijímací straně(TECOMAT Foxtrot CP-1003). Nejprve je nutné nastavit komunikaci v konfiguraci HW v prostředí Mosaic. Příslušný sériový kanál nastavíme do režimu PLC, nastavíme adresu komunikačního kanálu na hodnotu 0 a přenosovou rychlost na hodnotu 9600 Bd. Tato rychlos se jevila jako vhodná hodnota z hlediska dostatečné rychlosti a vysoké odolnosti vůči rušení. Po provedení překladu uživatelského programu se nastavení komunikací stane součástí uživatelského programu. Poté musíme všem PLC účastnících se komunikace nahrát projekty všech

ostatních PLC účastnících se komunikace a vytvořit totožnou skupinu projektů ve všech komunikujících PLC.

Dále je třeba v Mosaic Manažeru projektu otevřít logické propojení Sítě PLC a zobrazí se plocha. Na plochu vložíme všechny PLC ze skupiny projektů (tudíž i TC700), namalujeme sběrnici a propojíme na ni všechny kanály v režimu PLC, které chceme mít v jedné síti (obrázek č. 20).



Obr. 20: PLC síť

Dvojitým kliknutím na kterékoliv vyústění čar, symbolizující komunikační spojení 2 PLC na obr. č 20, se zobrazí okno na obr. č. 21. Zde jsem nastavil potřebnou délku zóny a vhodně nastavil adresu zóny, která bude odpovídat číslu registrů proměnných s danými daty ve všech PLC sítě. Poté jsem musel totéž nastavit v TC 700.

	Projekt	Kanál	Adresa ...	Adresa zóny	Délka zóny	Název zóny	Stavové slovo
1	fuzzy	CH2	3	4000	36	TC700zona	StatusCH2_32
2	Urychlov...	CH1	0	5410	36	ZonaCH1_33	StatusCH1_2

Obr. 21: Okno s parametry komunikace

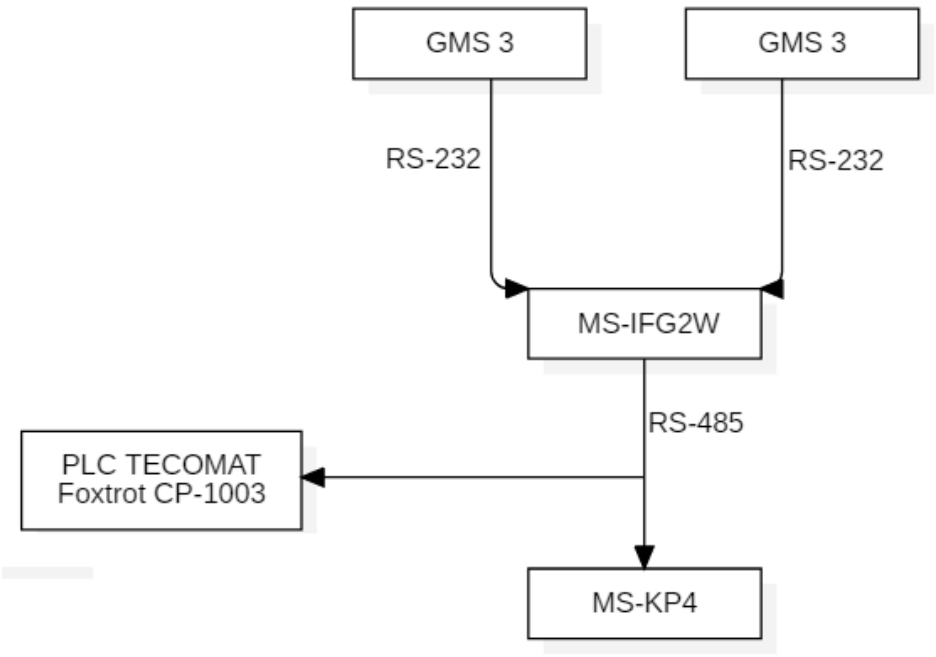
komunikace s monitorovací sítí MS 2000

Následující obrázek č. 22 zobrazuje schéma zapojení měřící sítě.

Pro úspěšnou komunikaci bylo třeba nastavit univerzální režim kanálu. Na obrázku č. 23 je zobrazeno okno s nastavitelnými parametry. Nastavení kanálu nebylo zcela jednoduché. Vycházel jsem částečně z popisu komunikace sériové linky dané výrobcem PLC, které vylo následně modifikováno dle potřeb netradičního komunikačního protokolu, který využívá dozimetrická monitorovací síť MS 2000. Analýza tohoto protokolu nebyla jednoduchá. Popis navázání úspěšné komunikace uvádím níže.

Z datasheetu dodaného personálem urychlovače jsem zjistil strukturu komunikace mezi MS-IFG2W a MS-KP4. Všechny zprávy mají shodný formát, liší se pouze počtem bytů, který je dán počtem parametrů povelu nebo odpovědi. Formát je následující:

@	Ap	Av	pb	povel	P1..Pn	Parametry/odpověď	CSUM



Obr. 22: Schéma monitorovací sítě MS 2000. GMS 3 – Gaiger-Millerův počítač (dozimetrická detekční sonda), MS-IFG2W – komunikační rozhraní, MS-KP4 – monitorovací panel

Kde parametry znamenají následující:

Parametr	**Význam**
@	synchronizační byte
Ap	adresa přijímače 0 až 31 binárně
Av	adresa vysílače 0 až 31 binárně
pb	počet bytů vysílané zprávy = n + 1
povel	kód povelu, většinou ascii znak velkého písmene
P1 ... Pn	případné parametry povelu nebo odpovědi
CSUM	kontrolní součet - osmibitový součet všech bytů od Ap do Pn včetně

Tab. 28: Příklad zprávy

Číslo bitu	0	1	2	3	4	5	6	7	8	9	10	11	12	...	20	21
Označení	@	Ap	Av	pb	povel	H	L	H	L	H	L	H	L	...	L	CSUM
Příklad	@	100	0	17	72	32	200	0	212	64	250	0		...	0	73
Význam		adresy				1. sonda				2. sonda				...		

V tabulce č. 28 je reálný příklad zprávy. Každá zpráva začíná znakem @, což je parametr v nastavení komunikačního kanálu s názvem "Počáteční znak". Nastavil jsem tedy jako počáteční znak číslo 64, což je označení v ASCII tabulce pro @.

Dále následují adresy přijímače a vysílače pod čísly bitů 1 a 2. V datasheetu je uvedeno, že moduly typu "posluchač" mají společnou adresu Ap rovno 100 a v

monitorovací síti nesmí odpovídat. Adresa vysílače modulu MS PM32 je vždy AV rovno nule.

Následuje číslo udávající délku zprávy zvětšené o jedna, protože zahrnuje i byte povelu, což není formálně obsah zprávy. V uvedeném příkladu je délak zprávy 17, tzn. 16 bytů s jedním bytem povelu. Povel má hodnotu 72, což v ASCII tabulce znamená písmeno "H". V datasheetu má tento povel parametry zobrazené v tabulce č. 29, kam jsem pro přehlednost přidal sloupec s číslem bytu v tabulce (28.

V poslední řadě už zbývá pouze byte číslo 21, což je kontrolní součet CSUM. Kontrolní součet je číslo vzniklé osmibitovým součtem všech bytů mimo synchronizační byte @.

Tab. 29: Parametry povelu H

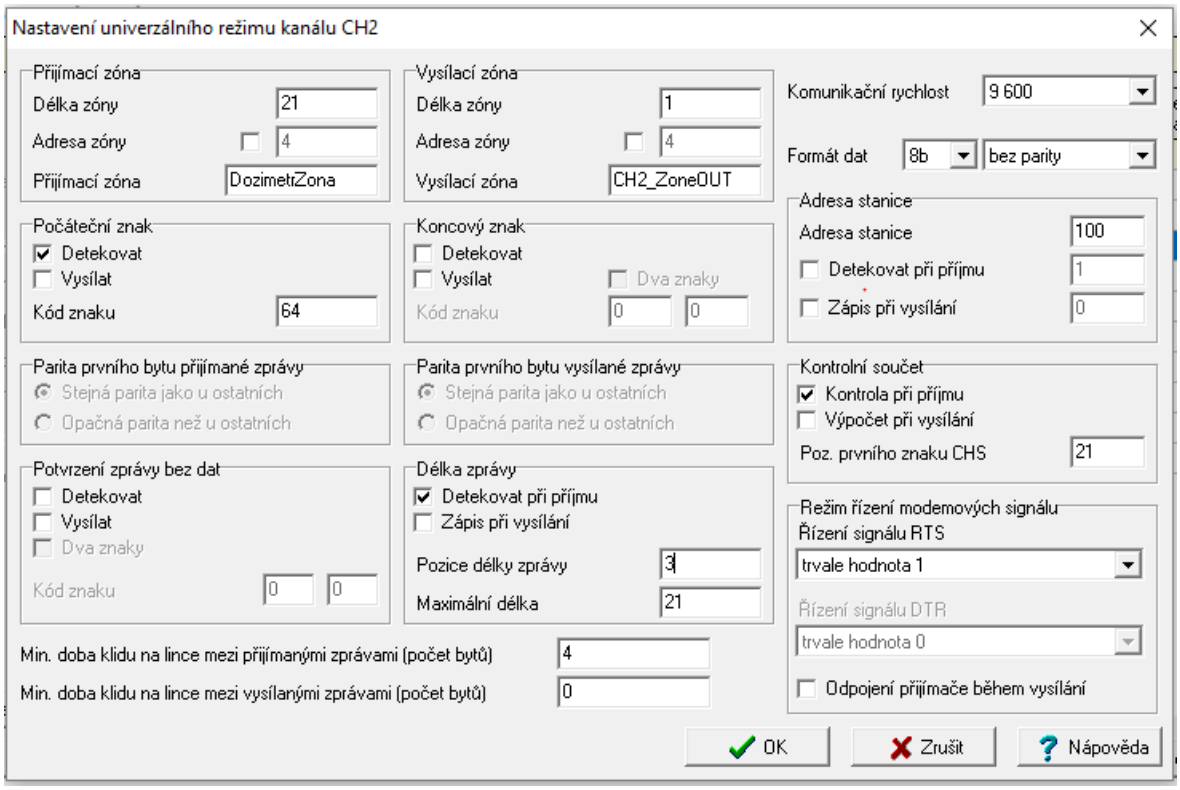
Číslo bytu	**Parametr**	**Hodnoty limitù a příkonu jednotlivých kanálù**
5	1. parametr	H-byte limitu 1. kanálu
6	2. parametr	L-byte limitu 1. kanálu
7	3. parametr	H-byte příkonu 1. kanálu
8	4. parametr	L-byte příkonu 1. Kanálu
9	5. parametr	H-byte limitu 2. kanálu
10	6. parametr	L-byte limitu 2. kanálu
11	7. parametr	H-byte příkonu 2. kanálu
12	8. parametr	L-byte příkonu 2. kanálu
13	9. parametr	H-byte limitu 3. kanálu
14	10. parametr	L-byte limitu 3. kanálu
15	11. parametr	H-byte příkonu 3. kanálu
16	12. parametr	L-byte příkonu 3. kanálu
17	13. parametr	H-byte limitu 4. kanálu
18	14. parametr	L-byte limitu 4. kanálu
19	15. parametr	H-byte příkonu 4. kanálu
20	16. parametr	L-byte příkonu 4. kanálu

Tab. 30: Zápis čísla do H a L bytů

Očíslování bitů	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Význam	řád				binární hodnota v rozmezí 0-999$[nSv/h]$											
byty	H (High)								L (Low)							
Příklad																
Dekadicky rozepsáno	32								200							
Bitově rozepsáno	0	0	1	0	0	0	0	0	1	1	0	1	0	1	0	0
Výsledek dekadicky	2				200											

Tabulka č. 30 vyobrazuje jakým způsobem lze z např. hodnoty limitu 1. kanálu získat limitní hodnotu 1. dozimetrické sondy. Informace obsahující 2 byty (číslo 32 a 200) lze přepočítat na hodnoty 2 a 200, kde číslo 2 udává řád a číslo 200 hodnotu v nSv/h. Hodnota finálního výsledku tedy vypadá následovně:

$$Hodnota = 10^2 * 200 = 20000nSv/h = 20\mu Sv/h \tag{9}$$



Obr. 23: Nastaveni univerzálního komunikačního kanálu

Vzhledem k tomu, že už znám formát a délku zprávy, mohu do kolonky "Přijímací zóna" (na obrázku č. 23) vyplnit číslo 21 bytů (počáteční znak do ní není započítaný). Vysílací zóna není důležitá, vzhledem k faktu že nevysíláme. Komunikační rychlost byla nastavena také podle informací na datasheetu na 9600 Bd bez parity. Koncový

znak není, zpráva končí kontrolním součtem, který je na 21. pozici. Adresu není třeba detekovat protože nic jiného v této síti nevysílá. Potvrzovat zprávu bez dat také není třeba, protože formát zprávy se nemění a zpráva bez dat by se v síti neměla vyskytovat.

Monitorovací síť MS 2000 obsahuje dvě sondy GMS 3, které komunikují s rozhraním MS-IFG2W. Toho rozhraní komunikuje s centrálním monitorovacím panelem po fyzické vrstvě RS-485. Toto schéma je zobrazeno na obrázku č. 22. Jak je vidět na schématu, komunikace je jednostranná, PLC Foxtrot pouze linku poslouchá. Následuje kód funkce, která z přijatých čísel (H-byte a L-byte) formátu byte vytváří jedno číslo ve formátu real.

```
FUNCTION prevod_dozi : real
 VAR_INPUT
 H, L :usint; //horní a dolní byte
 END_VAR
 VAR_IN_OUT
 END_VAR
 VAR
 bits : array[0..15] of bool;
 k:usint;
 mantisa, exponent:uint;
 zbytek : uint;
 END_VAR
 VAR_TEMP
 END_VAR
 mantisa:=H;
 for k:=0 to 7 do
    zbytek:=mantisa mod 2;
     mantisa:=mantisa/2;
    bits[k]:=uint_to_bool(zbytek);
 end_for;
     mantisa:=L;
 for k:=8 to 15 do
     zbytek:=mantisa mod 2;
     mantisa:=mantisa/2;
     bits[k]:=uint_to_bool(zbytek);
 end_for;
     exponent:=bool_to_uint(bits[4]) +bool_to_uint(bits[5])*2
      +bool_to_uint(bits[6])*4 +bool_to_uint(bits[7])*8;
     mantisa:= bool_to_uint(bits[1])*512 +bool_to_uint(bits[0])*256
     +bool_to_uint(bits[15])*128 +bool_to_uint(bits[14])*64
     +bool_to_uint(bits[13])*32 +bool_to_uint(bits[12])*16
     +bool_to_uint(bits[11])*8 +bool_to_uint(bits[10])*4
     +bool_to_uint(bits[9])*2 +bool_to_uint(bits[8])*1;

     prevod_dozi:=uint_to_real(mantisa)*10**(exponent)/1000;

```

```
37 END_FUNCTION
```

Měření v bodě

Tato aplikace je součástí SCADA subsystému a je vytvořená v prostředí Reliance s využitím Visual Basic script. Tato aplikace umožňuje měření a výpočet dávkového příkonu ionizujícího záření a času expozice při použití ionizační komory a elektrometru Keithley 617. Měření dávkového příkonu se provádí pouze v jednom specifikovaném bodě, ve kterém je umístěn ozařovaný vzorek.

Osoba, která chce použít tuto aplikaci musí nejprve zadat v kGy požadovanou dávku D, kterou chce dodat ozařovanému předmětu. Poté musí nastavit rozsah elektrometru stejný jako v uživalském prostředí. Hodnoty se velice mění v čase v závislosti na mnoha parametrech (hlavně na momentálním výkonu mikrotronu), takže rozsah se nastavuje dle předpokládaného dávkového příkonu. Dále se zadává koeficient ionizační komory, který závisí na používané ionizační komoře. Poté je nutné zvolit konkrétní ampérmetr, pomocí kterého měříme aktuálně urychlených elektronů, tzn. IE terč (Proud terčem) nebo IS IN (Indukční snímač IN). Následně jsou dopočteny zbylé hodnoty.

Tyto výpočty jsou realizovány pomocí Visual basic skriptu v SW Reliance.
Hodnota "Průměrný proud" Mu [pA] se spočte dle vztahu:

$$Mu = \frac{I_{mer} * Ic_{ref}}{100} \tag{10}$$

Kde I_{ref} je proud analogově měřený elektrometrem a Ic_{ref} je referenční proud, který jsme zadali podle elektrometru. Výpočet v příloze na řádku č. 22.

Průměrný dávkový příkon $\dot{D}$ [Gr/s] se spočte dle vztahu:

$$\dot{D} = Mu * 10^{-12} * koef \tag{11}$$

Kde $koef$ $[Gr/C]$znamená koeficient ionizační komory. Výpočet v příloze na řádku č. 23.
Následně se spočte přibližná doba expozice t [s]dle vztahu:

$$t = \frac{D}{\dot{D}} \tag{12}$$

Jako poslední hodnota se spočte velikost náboje Q [uC] dle vztahu:

$$Q = Mu * t \tag{13}$$

Tyto hodnoty je možné uložit do tabulky kliknutím na tlačítko "Uložit hodnoty do tabulky". Následuje kód Visual Basic skriptu s popisky, které tuto teorii realizuje.

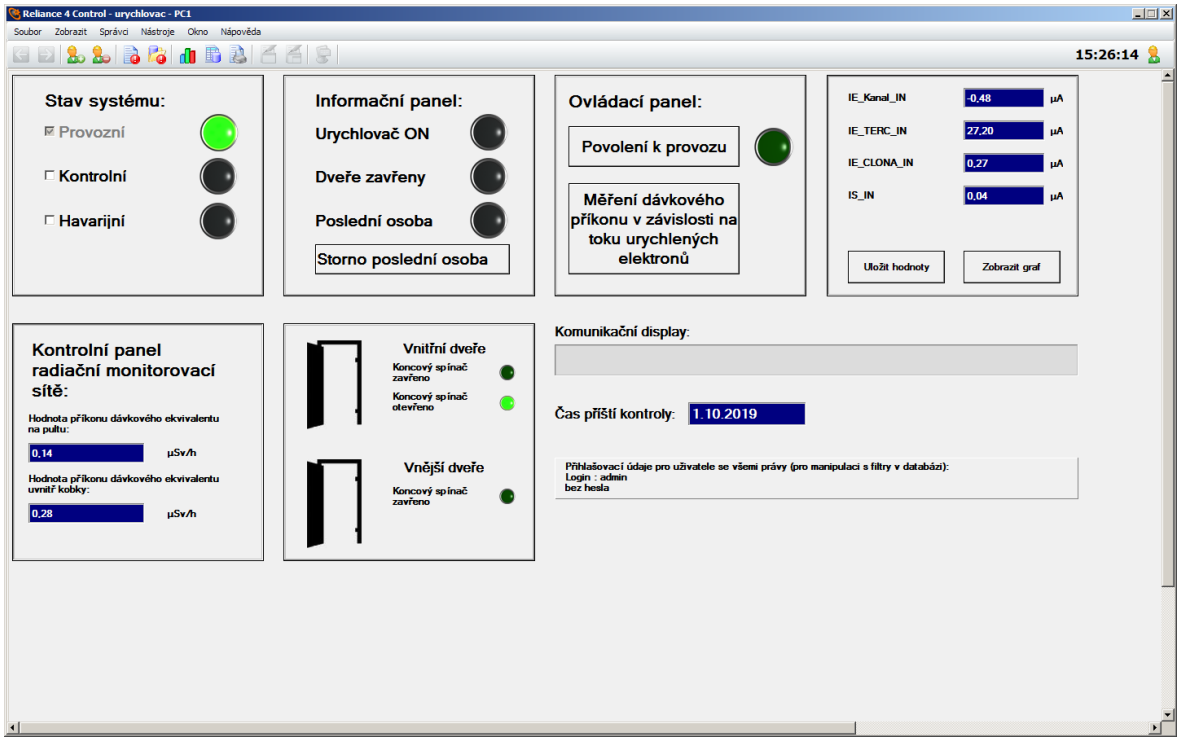
```
Option Explicit
rem dekladace proměnných
dim davka,icref,ieterc,ie_ind,proud,proud_poc,
davkovy_prikon,t_ir,Q,ss,mm,hh,pocetni_proud,koef
zbyvajicich_kroku,rotace,krok_displej,krok
rem definice a zápis reálných hodnot do dekladovaných proměnných
davka=rtag.GetTagValue(&quot;System&quot;,&quot;Davka&quot;)*1000
icref=rtag.GetTagValue(&quot;System&quot;,&quot;Ic_ref&quot;)
ieterc=rtag.GetTagValue(&quot;Teco1&quot;,&quot;IE_TERC_IN&quot;)
proud=rtag.GetTagValue(&quot;Teco1&quot;,&quot;prumerny_proud&quot;)
ie_ind=rtag.GetTagValue(&quot;Teco1&quot;,&quot;IS_IN&quot;)
pocetni_proud=rtag.GetTagValue(&quot;System&quot;,&quot;pocetni_proud&quot;)
koef=rtag.GetTagValue(&quot;System&quot;,&quot;koef&quot;)

rem v kapitoje Měření v bodě jsou uvedené následující vztahy
proud_poc=proud*icref/100 rem pikoampery
davkovy_prikon=proud_poc*10^(-12)*koef rem Gr/s
if davkovy_prikon&gt;0 then
rem zamezení děleí nulou
t_ir=davka / davkovy_prikon
rtag.SetTagValue &quot;System&quot;,&quot;t&quot;,t_ir

if pocetni_proud=false then
rem pokud jsme zvolili sondu IE Terč
Q=ieterc*t_ir
rem výpočet náboje
rtag.SetTagValue &quot;System&quot;,&quot;IS_IN_TERC1&quot;,&quot;TERC&quot;
rem uvedení správné meřící sondy do datové tabulky
else
rem pokud jsme zvolili sondu IS IN
Q=ie_ind*t_ir
rtag.SetTagValue &quot;System&quot;,&quot;IS_IN_TERC1&quot;,&quot;IS&quot;
rem uvedení správné meřící sondy do datové tabulky
end if
rem přepočet času v sekundách na větší jednotky
hh=t_ir\3600
t_ir=t_ir mod 3600
mm=t_ir\60
ss=t_ir mod 60
```

```
40
41 else
42 rem pokud bychom dělili nulou
43 ss = 0
44 mm=0
45 hh=0
46 Q = 0
47 end if
48 rem zápis proměnných skriptu do proměnných systému
49 rtag.SetTagValue &quot;System&quot;,&quot;Naboj&quot;,Q
50 rtag.SetTagValue &quot;System&quot;,&quot;cas_expozice_ss&quot;,ss
51 rtag.SetTagValue &quot;System&quot;,&quot;cas_expozice_mm&quot;,mm
52 rtag.SetTagValue &quot;System&quot;,&quot;cas_expozice_hh&quot;,hh
53 rtag.SetTagValue &quot;System&quot;,&quot;Davkovy_prikon&quot;,davkovy_prikon
54 rtag.SetTagValue &quot;System&quot;, &quot;proud_poc&quot;,proud_poc
```

4.2.4 HMI

Jak už bylo uvedeno, uživatelské prostředí je vytvořeno v SW Reliance. Zde jsem vytvořil všechny okna s grafickou reprezentací tlačítek, kontrolek, měřících přístrojů, ... a také jednotlivé skripty pro výše zmíněné výpočty.

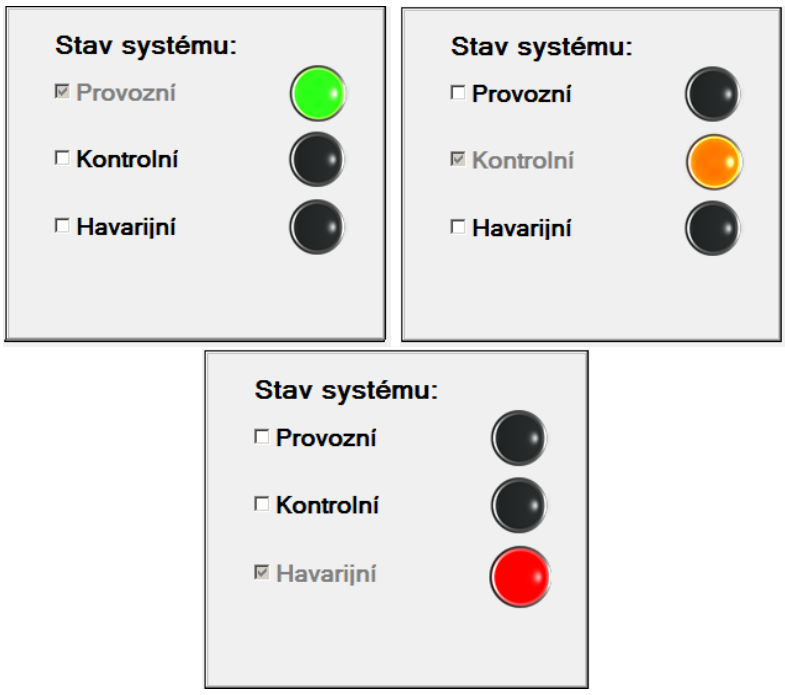
Na obrázku č. 24 je zobrazené grafické rozhraní SCADA systému je zobrazeno na . Tvoří jí 6 základních panelů a dalších segmentů, které následně specifikuji.



Obr. 24: Hlavní plocha

– **Stav systému**

Na obrázku č. 25 jsou vyobrazeny tři obrázky, které na dispečerské obrazovce zobrazují stav systému s možností stav změnit kliknutím na daný stav.



Obr. 25: Tři možné stavy systému

– **Informační panel**

První kontrolka informačního panelu zobrazuje, zda je urychlovač aktivní, tzn. jestli je zrovna "zapnutý".

Další kontrolka "Dveře zavřeny" dává signál o dovření dveří.

Kontrolka "Poslední osoba" signalizuje, zda je v kobce urychlovače osoba (zda poslední odoba opouštějící kobku stiskla tlačítko "Poslední osoba").

Tlačítko "Storno poslední osoby" je v případě, že je třeba anulovat signál o poslední osobě v kobce.

Obr. 26: Informační panel

– **Ovládací panel**

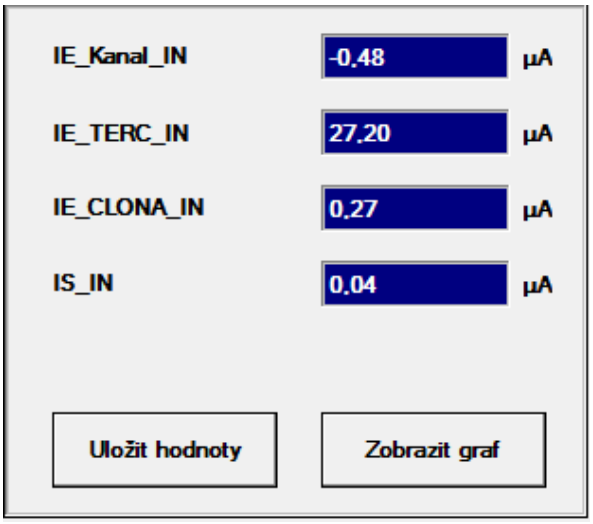
Tlačítko" Povolení k provozu" podává systému žádost o povolení k provozu. Systém žádost vyhodnotí a pokud jsou splněny všechny podmínky pro udělení povolení k provozu, tak se rozsvítí kontrolka vedle tlačítka.
Tlačítko "Měření dávkového příkonu v závislosti na toku urychlených elektronů" otevře nové okno (obrázek č. 34). O této funkci budu psát podrobněji později.

Obr. 27: Ovládací panel

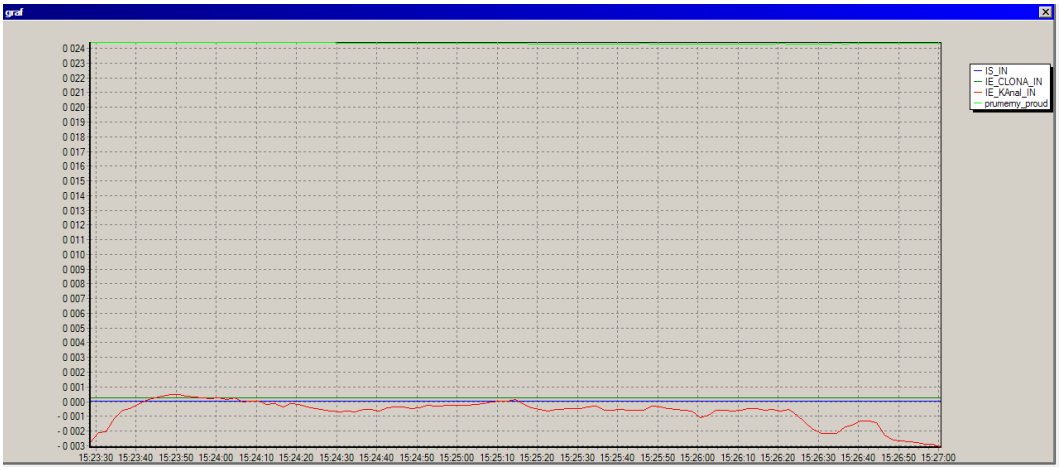
– **Hodnoty ze senzorů TC700**

Čtyři displeje na obrázku č. 28 zobrazují hodnoty, o kterých jsem psal v kapitole 4.2.3. Jsou to hodnoty čtyř indukčních snímačů. Tyto hodnoty lze v jakoukoliv chvíli uložit do datové tabulky stisknutím tlačítka "Uložit hodnoty". Při stisknutí tlačítka "Zobrazit graf" vyskočí dialogové okno zobrazené na obrázku

č. 29. Graf je plovoucí a je přehledně zobrazuje výše zmíněné hodnoty.

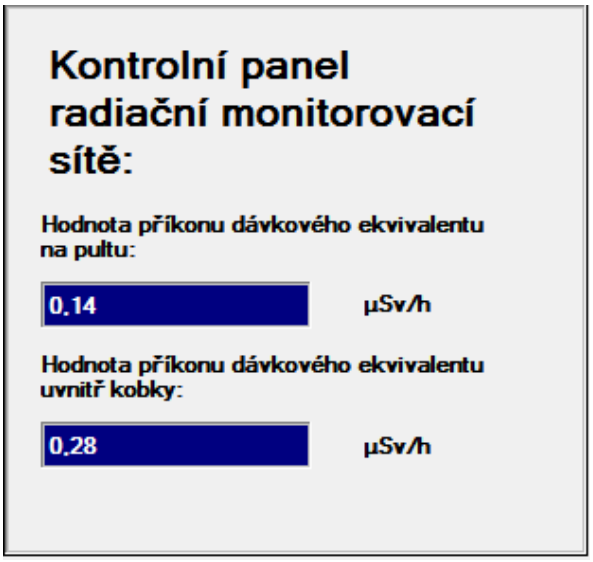


Obr. 28: Hodnoty napětí indukčních senzorů



Obr. 29: Plovoucí graf s hodnotami indukčních snímačů

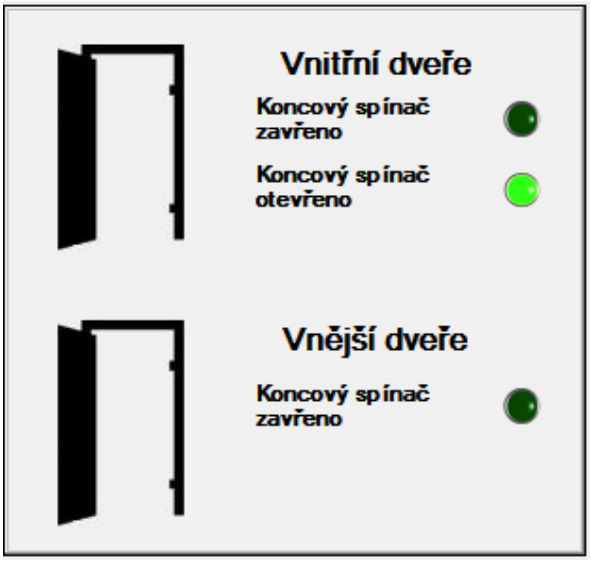
– **Kontrolní panel radiační monitorovací sítě**
Tento panel zobrazuje aktuální hodnotu dozimetrických sond. Jedná se o hodnoty, o kterých jsem psal v kapitole 4.2.3.



Obr. 30: Kontrolní panel radiační monitorovací sítě

– **Stav dveří**

Grafické zobrazení stavu dveří (zda jsou otevřeny, zavřeny či nedovřeny). Tři malé kontrolky, z nichž jedna na obrázku č. 31 svítí, signalizují přímo stav koncových snímačů.



Obr. 31: Signalizace stavu dveří

– **Komunikační displej**

Tento display slouží ke komunikaci systému s dispečerem. Například v něm dispečera informuje, že se zrovna otevírají dveře, že nastal havarijní stav, o úspěšnosti kontroly apod.

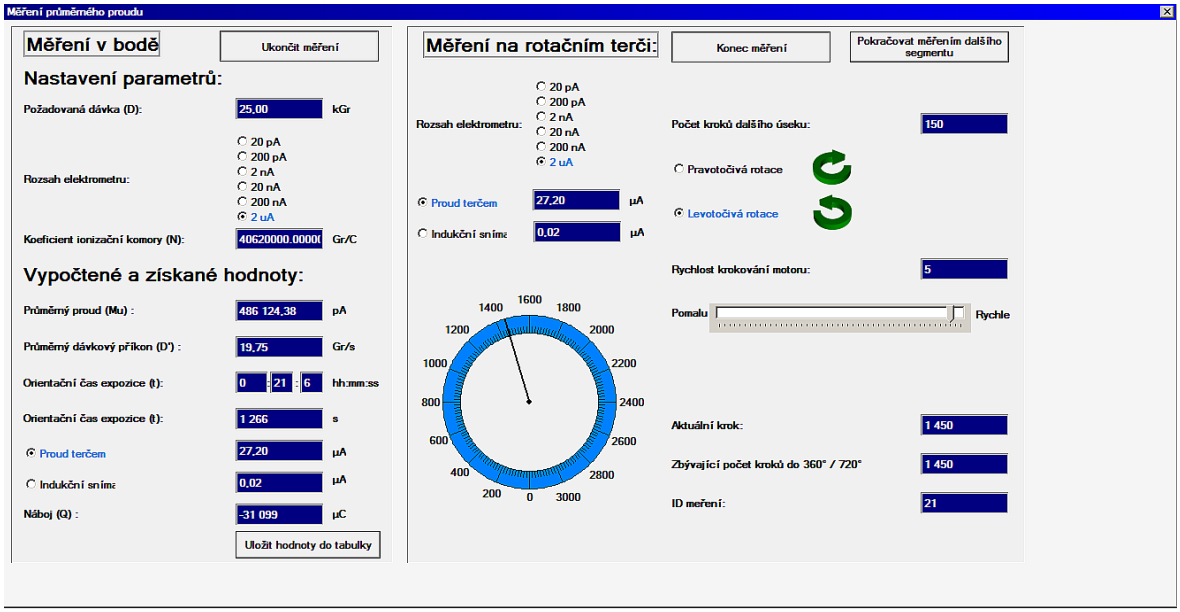
Obr. 32: Komunikační displej systému k uživateli

– **Čas příští kontroly**

  Jednou za 90 dní musí být provedena kontrola havarijních prvků v kobce (hlavně "stop" tlačítka, nouzového otevírání dveří, a některé koncové spínače). Tento display zobrazuje datum, kdy má být provedena kontrola. Pokud byla kontrola zrovna úspěšně provedena, připočte se k tomuto datu 90 dní. V případě, že je třeba udělat kontrolu, se za tímto displejem ukáže upozornění.

Obr. 33: Displej ukazující čas příští bezpečnostní kontroly

Na obr. 27 je tlačítko "Měření dávkového příkonu v závislosti na toku urychlených elektronů". Pokud stisknete toto tlačítko, zobrazí se nové okno zobrazené na obr. 34.



Obr. 34: okno "Měření příkonu"

Toto okno je opticky rozděleno na dvě části. Levá část nese jméno "Měření v bodě" a pravá část "Měření na rotačním terči". Levá část už byla popsána v kapitole 4.2.3.

Ovládání aplikace "Měření na rotačním terči" funguje následovně. Nejprve je nutné nastavit rozsah elektrometru, jako tomu bylo v případě měření v bodě, a zvolit správnou měřící sondu (IE Terč nebo IS IN). Krokový motor mí dohromady 3200 mikrokroků na

jednu celou otáčku. Nejprve zvolíme počet kroků, které se mají vykonat a poté smysl rotace motoru. V tu chvíli už stačí jen kliknout na tlačítko "Spustit měření" (pokud je stisknuto, mění se na "Konec měření") a začne měření.

Vzhledem k tomu, že měření probíhá při spuštěném urychlovači v kobce, není vidět v jaké fázi měření je. K tomu slouží na výše zmíněném obrázku modrý "budík", který zobrazuje v jaké poloze je terč a zda se ještě točí. Měření se musí provádět postupně po segmentech, protože se razantně mění proud a tím se musí manuálně zadávat nové referenční hodnoty elektrometru. Posuvná lišta umožňuje měnit rychlost rotace krokového motoru.

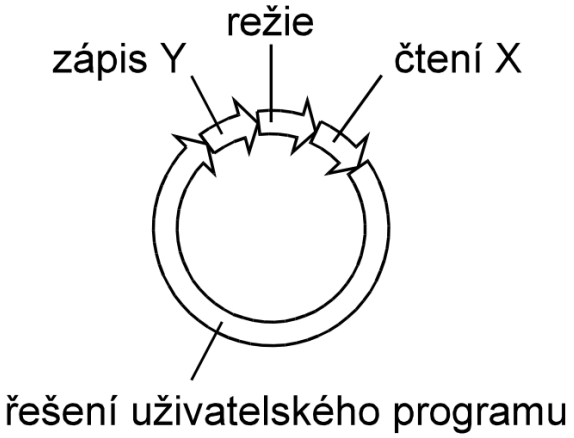
Čím rychleji rotor krokového motoru rotuje, tím je nižší vzorkování. Ukládání hodnot probíhá ve SCADA systému a ta má periodu vzorkování 0.5 sekundy, což je o poznání pomaleji než PLC, které si hodnoty v každém kroku ukládá do proměnných. Toto omezení by bylo možné eliminovat, ale toto vzorkování je pro tuto aplikaci postačující. Skript pro ukládání hodnot:

4.3 Technická realizace systému

Systém je nainstalovaný v mikrotronové laboratoři v tunelu pod Vítkovem, Praha-Karlín. Jádrem systému je PLC firmy TECO a.s. TECOMAT Foxtrot s centrální řídící jednotkou CP-1003.

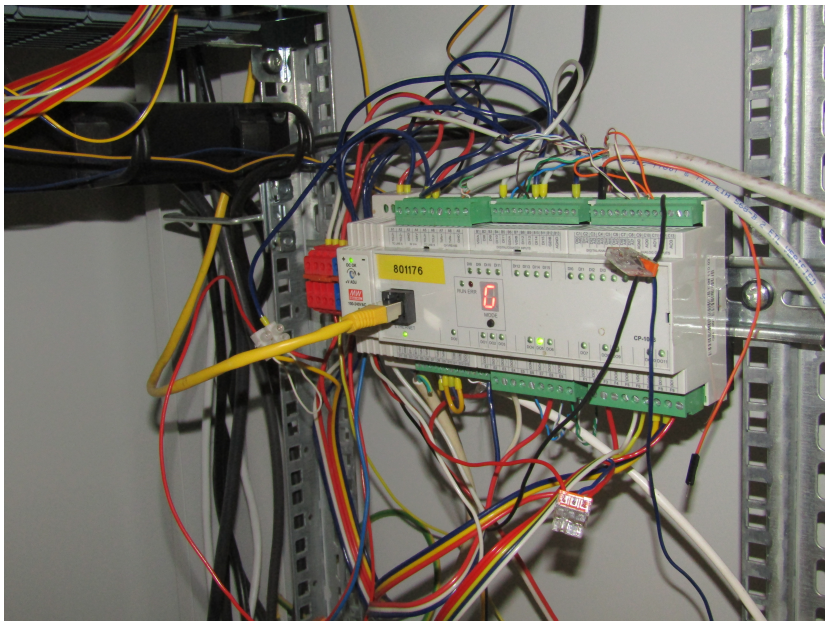
4.3.1 Tecomat foxtrot CP-1003

TECOMAT foxtrot CP-1003 je programovatelný automat (PLC). PLC je číslicový řídící elektronický systém určený pro řízení pracovních strojů a procesů v průmyslovém prostředí. PLC prostřednictvím číslicových nebo analogových vstupů a výstupů získává a předává informace z jednoho řízeného zařízení do jiného. Algoritmy řízení jsou uloženy v paměti uživatelského programu, který je cyklicky vykonáván. [13]



Obr. 35: Cyklus řešení uživatelského programu[13]

Na obrázku č. 35 je zobrazený zjednodušený cyklus PLC. 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. 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). Model CP-1003 jako většina dalších modelů od různých výrobců modulární, tzn. je možné na něco napojit širokou škálu modulů.[13] Na obrázku č. 36 je PLC v testovacím režimu s provizorní kabeláží.



Obr. 36: TECOMAT Foxtrot CP-1003

4.3.2 MS 2000

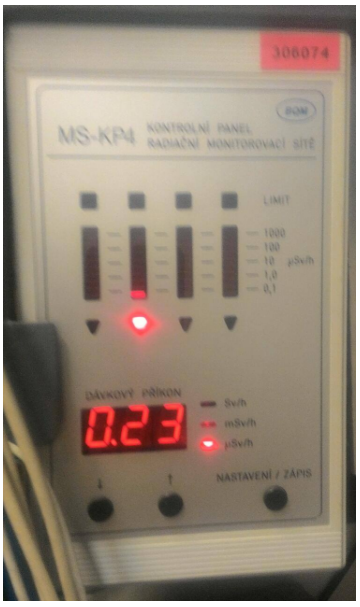
Monitorovací síť MS 2000 je určena k monitorování příkonu prostorového dávkového ekvivalentu a dalších fyzikálních veličin (teplota, koncentrace nebezpečných plynů atd.) na různých místech v reálném čase a k přenosu naměřených hodnot do jednoho řídícího nebo kontrolního místa.[14]

Obrázek č. 22 popisuje schéma zapojení této síťe.

V tomto schématu jsou dvě sondy GMS 3 připojené k MS IFG2W, jenž jim tvoří rozhraní. MS IFG2W přijímá informace z připojených sond a zasílá je do kontrolního panelu MS-KP4 po sériové lince RS-485. [14]

MS-KP4

Na obrázku č. 37 je zobrazen panel MS-KP4. Jedná se o zobrazovací zařízení, které na logaritmické stupnici zobrazuje hodnotu dávkového příkonu. Panel je schopen zobrazovat hodnoty až 4 sond. Disponuje displejem, který přibližně každou vteřinu zobrazí další sondu v pořadí. Panel také disponuje zvukovou signalizací, která alarmuje personál v případě, že by jedna ze sond vykazovala nadlimitní hodnoty. Tato limitní hodnota lze nastavit pro každou sondu zvlášť.



Obr. 37: Kontrolní panel MS-KP4

GMS 3

Na obrázku č. 38 je znázorněna sonda GMS 3. Jedná se o nízkorozsahovou dozimetrickou sondu pracující na principu Gaiger-Millerova počítače (popsán v kapitole 3.2.4). Sonda detekuje gama záření dávkového příkonu ekvivalentu v rozsahu 100nSv/h až 10mSv/h. [14]

Obr. 38: dozimetrická sonda GMS 3 [14]

Elektrometr Keithley 617

Jedná se o programovatelný elektrometr vyrobený firmou Keithley Instruments Inc. Výstup z elektrometru bylo napětí s referenční hodnotou 2 V, čemu poměrově

odpovídá hodnota měřeného proudu s vybranou referenční hodnotou proudu. Z toho důvodu jsou hodnoty referenčních proudů 20 pA, 200 pA apod.

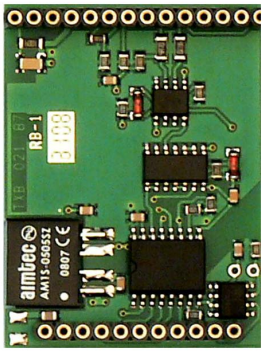
Elektrometr Keithley 617 je vysoce citlivý přístroj, vytvořený pro měření napětí, proudu, náboje a odporu. Měřící rozsah elektrometru je 10 μV až 200V pro měření napětí, 10 fA až 20 mA pro měření proudu a 0, 1 Ωaž 200 GΩ(10^{16} Ωpři použití vestavěného zdroje napětí) pro měření odporu a 10 fC až 20 nC pro měření náboje. Velmi vysoká vstupní impedance a extrémně nízký výstupní proudový offset umožňuje měření v situacích, kdy by měření jinými přístroji mělo škodlivý efekt na měřený obvod. [15]

Obr. 39: elektrometr

4.3.3 Komunikace

MR-0114

Komunikační modul MR-0114 slouží ke komunikaci po fyzické vrstvě RS-485. TECOMAT Foxtrot CP-1003 má vestavěný pouze jeden komunikační kanál RS-485. Pro účely tohoto systému byly třeba dva, tudíž byl dokoupen výše zmíněný modul (obrázek č. 40).



Obr. 40: Komunikační submodul MR 0114[13]

RS485

RS-485 je standard sériové komunikace definovaný v roce 1983 sdružením EIA. Standard RS-485 je navržen tak, aby umožňoval vytvoření dvouvodičového poloduplexního vícebodového sériového spoje.

Pomocí této sběrnice může komunikovat maximálně 32 vysílačů a 32 přijímačů, což je výrazné vylepšení oproti RS-422, kde mohl v danou chvíli existovat pouze jeden vysílač (tento způsob komunikace se nazývá multidrop, protože přenášená informace „spadne“ do více přijímačů) či oproti RS-232C (point-to-point). Funkčnost sběrnice je zaručena díky tomu, že všechny přijímače i neaktivní vysílače se v klidu musí nacházet ve stavu vysoké impedance, tj. nijak neovlivňují komunikující zařízení. Veškeré řízení přenosu i arbitráž sběrnice je ponechána na protokolu vyšší vrstvy.

Rozdíl mezi oběma napěťovými potenciály musí dosahovat hodnoty minimálně 0, 2 Voltů, typicky se však používají mnohem vyšší rozdíly, například 5 V, 7 V či 12 V.[16]

Fyzická vrstva RS-485 byla v rámci tohoto projektu dvakrát využita pro účely komunikace.

Prvním případem je přenos dat mezi PLC TECOMAT Foxtrot CP-1003 a PLC TECOMAT TC700 pomocí komunikačního protokolu EPSNET. Fyzické zapojení bylo realizováno propojením příslušných vstupů/výstupů (TxRx+ a TxRx-) na obou PLC.

V druhém případě byla použita pro získání dozimetrických hodnot měření. Fyzické zapojení je totožné jako v prvním případě s tím rozdílem, že bylo nutné přidat zakončovací odpor 120 Ω.

V obou případech byl použitý kabel s označením FTP cat.5E, což znamená, že se jedná o stíněnou kroucenou dvoulinku.

4.3.4 Ovládací panel dveří

Ovládací panel, zobrazený na obrázku č. 27 umožňuje manuální ovládání stínících dveří (obrázek č. 42) stanoviště a uvedení systému do stavu povolení k provozu.

Obr. 41: Ovládací panel stínicích dveří

Obr. 42: Stínicí dveře

5 Závěr

Tvorba systému pro mě znamenala výzvu. Nejprve jsem se musel podrobněji seznámit s objektovou modelovací technikou (OMT), která mi zabrala mnoho času. Ačkoliv se jedná o velice silný nástroj pro prvotní návrh struktury projektu, občas mi exaktní pravidla této metodiky činily problémy při návrhu systému. Navzdory tomu, jak je metoda stará, stále se jeví velice užitečně a efektivně, pokud je metodika správně aplikována.

Při tvorbě objektového, dynamického a funkčního modelu systému v rámci OMT, jsem se musel nastudovat metodiku tvorby UML modelů, které svoji grafickou ilustrací zpřehledňují strukturu tvořeného systému. Nejprve jsem se domníval, že je tvořím za účelem dokumentace, aby např. v případě potřeby úpravy systému mohl jiný člověk jednoduše pochopit strukturu a následně upravit systém. Poté jsem zjistil, že mě tvorba diagramů logicky nutí k zamyšlení nad systémem a v jeho důsledku jsem tyto diagramy mnohokrát přepracovával, než jsem si byl jistý, že takto navržený systém bude fungovat.

Vzhledem k faktu, že systém bude ovládat prvky v potenciálně nebezpečném zařízení a abych si byl jistý, že bude fungovat tak, jak byl navržený, považoval jsem za vhodné si nastudovat problematiku a nebezpečí ionizujícího záření. O tomto tématu pojednává teoretická část.

Pro samotné psaní kódu programu jsem se musel naučit programovací jazyk Structured text (ST), ve kterém je napsaný celý program pro PLC. SCADA systém zase pro své skripty používá jazyk Visual basic script, který je o dost jednodušší než ST. Ačkoliv ST vychází z jazyku Pascal, který jsem se učil, tak jsem se setkal s mnohými odlišnostmi a problémy, které měly primárně návaznost na hardware, tzn. na PLC, pro které je jazyk stvořený. Tato práce je psaná v online latex editoru Overleaf, což je více méně další jazyk, který jsem se učil, když jsem pracoval na tomto projektu. Nikdy před tím jsem tento způsob psaní textů nepoužíval.

Nemalou součástí designu a implementace projektu byla komunikace s externími zařízeními. Nejprve jsem musel najít způsob, jak "odposlouchávat"komunikaci po RS-485 v dozimetrické monitorovací síti. Když se podařilo hardwarově správně připojit PLC do sítě, čekalo mě nastavování univerzálního komunikačního kanálu v PLC. Tam jsem musel jít cestou částečně experimentálního způsobu nastavování parametrů.

Komunikace PLC Foxtrot CP-1003 se stávajícím PLC TECOMAT TC700, pomocí kterého se ovládá urychlovač, probíhala pomocí TECOMAT komunikačního protokolu EPSNET, což vyžadovalo úplně odlišný způsob nastavování než komunikace s dozimetrickou monitorovací sítí. Zde byl větší problém s transformací přijímaných zpráv na vhodný datový formát čísel, tudíž spíše programátorský problém.

V poslední řadě jsem vyvíjel aplikaci pro ovládání systému z dispečerského stanoviště v SW Reliance. V rámci dispečerského stanoviště jsem vytvořil několik skriptů pro měření dávkových příkonů v ozařovaných vzorcích. Jeden z terčů, na kterém byly vzorky ozařovány, byl rotován pomocí krokového motoru. Průběžně jsou hodnoty měřených veličin ukládány do tabulky, počítány jejich střední hodnoty. Tyto výstupy jsou dále používány pro optimální nastavení výkonu urychlovače při provozním ozařování.

Systém byl úspěšně otestován jak v simulovaném režimu, tak i za chodu urychlovače.Vzhledem k tomu, že je systém využíván při každodenním provozu urychlovače, kladl jsem veliký důraz na spolehlivost a funkčnost systému. Jakákoliv chyba by mohla velice zkomplikovat práci personálu a omezit provoz urychlovače. Z toho důvodu jsem cítil velikou zodpovědnost při tvorbě tohoto systému. V Mikrotronové laboratoři probíhá v současné době náhrada původního systému za nově vytvořený.

Reference

[1] KRIST, Pavel a Jiri BILA. A mathematical model of the MT 25 microtron: treaties and international agreements registered or filed and recorded with the Secretariat of the United Nations. Journal of Instrumentation. 2011, 1986, 6(10), T10005-T10005. DOI: 10.1088/1748-0221/6/10/T10005. ISSN 1748-0221. Dostupné také z: http://stacks.iop.org/1748-0221/6/i=10/a=T10005?key=crossref.3e77be55d411e83ba0dbed55e83346f8.

[2] CHVÁTIL, David. Dokumentace k nakládání se zdroji ionizujícího záření: Mikrotron MT 25. 18.3. 2019. 2019.

[3] KRIST, Pavel. Ústav jaderné fyziky AV ČR: Example of irradiation services [online]. In: . [cit. 2019-06-02]. Dostupné z: http://www.ujf.cas.cz/cs/oddeleni/oddeleni-urychlovacu/microtron/service.html.

[4] KLENER, Vladislav. Principy a praxe radiační ochrany. Praha: Státní uřad pro jadernou bezpečnost, 2000. ISBN 80-238-3703-6.

[5] FREITINGER SKALICKÁ, Zuzana, Jiří HALAŠKA, Renata HAVRÁNKOVÁ, et al. Radiobiologie: VELIČINY POUŽÍVANÉ V RADIAČNÍ OCHRANĚ. Radiobiologie [online]. [cit. 2019-05-15]. Dostupné z: http://fbmi.sirdik.org/1-kapitola/15/155.html.

[6] FUKÁTKO, Tomáš. Detekce a měření různých druhů záření: 5. díl edice Senzory neelektrických veličin. BEN-Technická literatura, 2007. ISBN 978-80-7300-1.

[7] Object Modeling Technique (OMT). In: NTNU: Norwegian University of Science and Technology [online]. [cit. 2019-05-07]. Dostupné z: https://www.idi.ntnu.no/grupper/su/publ/html/totland/ch0527.htm.

[8] K.SHINDE, Subhash, Varunakshi BHOJANE a Pranita MAHAJAN. NLP based Object Oriented Analysis and Design from Requirement Specification. International Journal of Computer Applications [online]. 2012, 47(21), 30-34 [cit. 2019-08-13]. DOI: 10.5120/7475-0574. ISSN 09758887. Dostupné z: http://research.ijcaonline.org/volume47/number21/pxc3880574.pdf.

[9] JAKUBEC, David. Využití softwarového jazyka UML pro programování řídicích systémů PLC. Praha, 2018. Bakalářská práce. České vysoké učení technické, Fakulta strojní. Vedoucí práce Ing. Mgr. Jakub Jura, Ph.D.

[10] POLÍVKA, Rostislav. DATOVÝ A FUNKČNÍ MODEL INFORMAČNÍHO SYSTÉMU. BRNO. Bakalářská práce. VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ. Vedoucí práce Doc. Ing. Miloš Koch, CSc.

[11] PLC Program State Diagram : CASE Structures in Structured Text:

What would a CASE structure for a Multi-State PLC Program Look Like?. Drives & System: PLC's, PAC's, Variable Speed Drives and Industrial Automation [online]. 30.6.2016 [cit. 2019-05-15]. Dostupné z: http://www.drivesandsystems.com/plc-program-sequences/.

[12] SÉRIOVÁ KOMUNIKACE PROGRAMOVATELNÝCH AUTOMATŮ TECOMAT - MODEL 32 BITŮ In: Teco a.s. [online]. [cit. 2019-08-14]. Dostupné z: https://www.tecomat.cz/modules/DownloadManager/download.php?alias=txv00403_01_comm_serial32_cz.

[13] PROGRAMOVATELNÉ AUTOMATY TECOMAT FOXTROT CP-1003, CP-1013: CP-1003. In: Teco a.s. [online]. [cit. 2019-08-14]. Dostupné z: https://www.tecomat.cz/modules/DownloadManager/download.php?alias=txv00433$_$01_foxtrot_cp_1003.

[14] Monitorovací síť MS 2000. In: Empos: Měřící technika a technika ionizujícího záření [online]. EMPOS® spol. s r.o. [cit. 2019-05-16]. Dostupné z: https://www.empos.cz/attachments/5/5869d5e8a7a0bd9139d26d1accbc568b.pdf.

[15] KEITHLEY INSTRUMENTS, INC. Model 617 Programmable Electrometer Instruction Manual [online]. 1984. Cleveland, Ohio, U.S.A.: Test Instrumentation Group, 1968 [cit. 2019-08-17]. ISBN 6179-0101.

[16] Sběrnice RS-422, RS-423 a RS-485: Elektrické charakteristiky sběrnice RS-485. ROOT.cz [online]. [cit. 2019-08-18]. Dostupné z: https://www.root.cz/clanky/sbernice-rs-422-rs-423-a-rs-485/.

Seznam obrázků, grafů a tabulek

Seznam obrázků

Obrázek 1 Blokové schéma zapojení systému 4
Obrázek 2 Schéma mikrotronu MT25 7
Obrázek 3 Pracoviště urychlovače [3] 8
Obrázek 4 Bodové schéma elektrického detektoru ionizujícího záření [6] 14
Obrázek 5 Schematické zapojení GM detektoru [6] 15
Obrázek 6 Počet sbíraných iontů jako funkce napětí [6] 16
Obrázek 7 Metodika tvorby systému [8] 18
Obrázek 8 Diagram případu užití 27
Obrázek 9 Třídový diagram 1. část. 28
Obrázek 10 Třídový diagram 2. část. 29
Obrázek 11 Interface třída SCADA 30
Obrázek 12 Sekvenční diagram pro stisknutí havarijního stop tlačítka 31
Obrázek 13 Aktivity diagram pro měření příkonu na rotačním terči 32
Obrázek 14 Aktivity diagram otevření dveří 33
Obrázek 15 Stavový diagram 34
Obrázek 16 Diagram datových toků (Data flow diagram) 36
Obrázek 17 Prvotní návrh stavového diagramu 38
Obrázek 18 Přerušení 38
Obrázek 19 Senzory proudu 46
Obrázek 20 PLC síť 50
Obrázek 21 Okno s parametry komunikace 50
Obrázek 22 Schéma monitorovací sítě MS 2000 51
Obrázek 23 Nastaveni univerzálního komunikačního kanálu 53
Obrázek 24 Hlavní plocha 57
Obrázek 25 Tři možné stavy systému 58
Obrázek 26 Informační panel 59
Obrázek 27 Ovládací panel 59
Obrázek 28 Hodnoty napětí indukčních senzorů 60
Obrázek 29 Plovoucí graf s hodnotami indukčních snímačů 60
Obrázek 30 Kontrolní panel radiační monitorovací sítě 61
Obrázek 31 Signalizace stavu dveří 61
Obrázek 32 Komunikační displej systému k uživateli 62
Obrázek 33 Displej ukazující čas příští bezpečnostní kontroly 62
Obrázek 34 okno "Měření příkonu" 62
Obrázek 35 Cyklus řešení uživatelského programu[13] 64

Obrázek 36 TECOMAT Foxtrot CP-1003 65
Obrázek 37 Kontrolní panel MS-KP4 66
Obrázek 38 dozimetrická sonda GMS 3 [14] 66
Obrázek 39 elektrometr 67
Obrázek 40 Komunikační submodul MR 0114[13] 68
Obrázek 41 Ovládací panel stínicích dveří 69
Obrázek 42 Stínicí dveře 69

Seznam tabulek

Tabulka 1 Seznam podstatných jmen a slovních spojení 19
Tabulka 2 Nadbytečné objekty a třídy 20
Tabulka 3 Irelevantní třídy 20
Tabulka 4 Neurčité objekty 20
Tabulka 5 Atributy 20
Tabulka 6 Operace 20
Tabulka 7 Implementační konstrukty 20
Tabulka 8 Stavy 20
Tabulka 9 Zbylá slovní spojení 21
Tabulka 10 Sémantický popis slovních spojení 21
Tabulka 11 Slovesa a slovesné fráze 23
Tabulka 12 Vazby mezi zrušenými objekty 24
Tabulka 13 Nadbytečné vazby 24
Tabulka 14 Irelevantní vazby 24
Tabulka 15 Operace 25
Tabulka 16 Implementační vazby 25
Tabulka 17 Odvozené vazby 25
Tabulka 18 Zbývající slovesná fráze 25
Tabulka 19 Adepty na atributy 25
Tabulka 20 Atributy zrušených objektů 26
Tabulka 21 Objekty 26
Tabulka 22 Atributy vazeb 26
Tabulka 23 Vstupy systému 35
Tabulka 24 Výstupy systému 35
Tabulka 25 funkce DFD 37
Tabulka 26 Posloupnost bytů 47
Tabulka 27 Short integer 47
Tabulka 28 Příklad zprávy 51
Tabulka 29 Parametry povelu H 52
Tabulka 30 Zápis čísla do H a L bytů 53

Přílohy

A Hlavní program

```
PROGRAM Hlavnisystem
  VAR_INPUT
  END_VAR
  VAR_OUTPUT
  END_VAR
  VAR

     aktualizace, aktualizace_cas, aktualizace_dozi : bool;
     pomocna_promena, pomocna_promena_2,pomocna_promena_3 :bool :=1;
     cas_aktualizace1, cas_aktualizace2, cas_aktualizace3 : TON;
     vypnutiTOF : TOF;
     dnesni_datum : date;
     pomocna_kontrola :DT;
     zaznmanenanikontrolaR_TRIG, zaznamenanidoziR_TRIG,
     havarijni_stavR_TRIG, r_mereni, r_mereni_not, r_pokr_rot,
     otev_trig : R_TRIG;
     dvere: ovladani_dveriFB;
     povoleni_k_provozu_pom : bool;
     k:usint;
     step_pom, dir_pom: bool;

  END_VAR
  VAR_TEMP
  END_VAR

  zaznamenanidozi:=0;
  zaznamenanikontrola:=0;

  if not kontrolni_stav  then
     if posledni_pamet then
        havarijni_stavR_TRIG(CLK:=havarijni_stav or stop1 or stop2);
     else
        havarijni_stavR_TRIG(CLK:=havarijni_stav);
     end_if;
     zaznamenanidoziR_TRIG(CLK:= vypnuti_urychlovace);
     if zaznamenanidoziR_TRIG.Q then     //záznam do provoz. děníku
        zaznamenanidozi:=1;
     end_if;
  end_if;

  vypnutiTOF (IN:=havarijni_stavR_TRIG.Q , PT:=T#1s ,
  Q=> vypnuti_urychlovace);     //vypnutí urychlovače
```

```
43  if stop1 or stop2 then     //pokud bylo stisknut havarijní prvek
44      if stav=40 then     //pokud byl stisknutý v rámci kontroly
45         if stop1 then kontrola_stop1:=true; end_if;
46         if stop2 then kontrola_stop2:=true; end_if;
47      else                                          //pokud ne
48         if posledni_pamet then
49           prechod(havarijni:=true);
50         end_if;
51      end_if;
52  end_if;
53  if havarijni_stavR_TRIG.q then     //přechod do havajijího stavu
54      prechod(havarijni:=true);     //stav => havarijní
55  end_if;
56
57  if (nouzove_otevreni and (not kontrolni_stav)) then
58     stav:=120;
59  end_if;
60  if stop_motor then
61  //pokud bylo stisknuto tlačítko na zastavení motoru dveří
62     stav:=170;
63     zprava:='Čerpadlo zastaveno';
64  end_if;
65
66  if povoleni_k_provozu_pom and (not koncovy_spinac1_zavreno or
67  not koncovy_spinac2_zavreno) then
68     if provoz_urychlovace then
69        prechod(havarijni:=true);
70     end_if;
71     povoleni_k_provozu_pom:=false;
72  end_if;
73
74  if koncovy_spinac1_zavreno and koncovy_spinac2_zavreno then
75     zavrene_dvere_kontrolka_pom:=true;
76  else
77     zavrene_dvere_kontrolka_pom:=false;
78     otev_trig(CLK:=koncovy_spinac1_zavreno=false and
79     koncovy_spinac2_zavreno=false);
80     if otev_trig.q then
81        posledni_pamet:=false;
82     end_if;
83  end_if;
84     //aktualizace dozimetrických hodnot
85  cas_aktualizace1(IN := pomocna_promena , PT := T#0.5s,
86  Q => aktualizace );
87  pomocna_promena:=true;
88
89  cas_aktualizace3(IN := pomocna_promena_3 , PT := T#1s,
```

```
90  Q => aktualizace_dozi);
91  pomocna_promena_3:=true;
92       //aktualizace času plc
93  cas_aktualizace2(IN := pomocna_promena_2 , PT := T#3600s,
94  Q => aktualizace_cas );
95  pomocna_promena_2:=true;
96         if aktualizace_cas then
97            dnesni_datum:=getdate();
98            pomocna_promena_2:=false;
99         end_if;
100
101 case stav of
102
103 0: //Inicializace
104 r0_p3_DO.DO0:=false;
105 r0_p3_DO.DO1:=false;
106 r0_p3_DO.DO2:=false;
107 r0_p3_DO.DO3:=false;
108 r0_p3_DO.DO4:=false;
109 r0_p3_DO.DO5:=false;
110 r0_p3_DO.DO6:=false;
111 r0_p3_DO.DO7:=false;
112 r0_p3_DO.DO8:=false;
113 r0_p3_DO.DO9:=false;
114 r0_p3_DO.DO10:=false;
115 r0_p3_DO.DO11:=false;
116 povoleni_k_provozu_pom:=false;
117 zvukova_signalizace_err:=false;
118
119 dnesni_datum:=getdate();
120 zaznamenanikontrola:=0;
121 zaznamenanidozi:=0;
122 povoleni_k_provozu_pom:=false;
123 povoleni_k_provozu_virtual := 0;
124
125 prechod(provozni:=true);
126 kontrola_stop1:=false;
127 kontrola_stop2:=false;
128 kontrola_nouzove_otevreni:=false;
129 kontrola_koncovy_spinac1:=false;
130 kontrola_koncovy_spinac2:=false;
131
132 prechod(provozni:=true);
133
134 20: //Provozni_stav
135
136     r_mereni(CLK:=rotacni_mereni);
```

```
137       r_mereni_not(CLK:=(not rotacni_mereni));
138       if r_mereni_not.q then
139         // pokud byl ukončený pokus měření na rotačním terči
140          pokracovani:=false;
141          krok:=0; //vynuluje se krok
142       end_if;
143
144       if r_mereni.q then
145         //nové měření na rotačním terči
146          enable_pom:=true;
147          ena_pom:=true;
148          enb_pom:=true;
149          cteni_dat:=false;
150          //vynulování hodnot
151          k:=0;
152          krok:=0;
153          //nové číslo měření
154          ID_mereni:=ID_mereni+1;
155          pokracovani:=true;
156       end_if;
157
158       if rotacni_mereni then
159       step_pom:=true;
160       r_pokr_rot(CLK:=pokracovani);
161       if r_pokr_rot.q then
162          if rotace=false then
163             cil_krok:=krok+kroky_usek;
164             dir_pom:=true;
165          else
166             cil_krok:=krok-kroky_usek;
167             dir_pom:=false;
168          end_if;
169
170       end_if;
171
172         if krok<cil_krok and (not rotace) then
173          k:=k+1;
174          if k>=rychlost_krokovani then
175             k:=0;
176             krok_pom:=krok;
177             IS_IN_pom:=IS_IN;
178             IE_TERC_IN_pom:=IE_TERC_IN;
179             prumerny_proud_pom:=prumerny_proud;
180             krok:=krok+1;
181             step_pom:=false;
182             cteni_dat:=true;
183          end_if;
```

```
184         end_if;
185
186         if krok>cil_krok and rotace then
187            k:=k+1;
188          if k>=rychlost_krokovani then
189             k:=0;
190             krok_pom:=krok;
191             IS_IN_pom:=IS_IN;
192             IE_TERC_IN_pom:=IE_TERC_IN;
193             prumerny_proud_pom:=prumerny_proud;
194             krok:=krok-1;
195             step_pom:=false;
196             cteni_dat:=true;
197          end_if;
198         end_if;
199        if krok=cil_krok then pokracovani:=false;
200        end_if;
201      end_if;
202
203      if posledni then posledni_pamet:=true; end_if;
204      //pokud bylo stisknuto poslední
205
206      if kontrola <= dnesni_datum then     //zda je kontrola aktuální
207         nutnost_kontroly:=1;
208      end_if;
209
210      if aktualizace then     //aktualizace dozimetrických hodnot
211         stav:=60;
212         pomocna_promena:=false;
213      end_if;
214
215      if aktualizace_dozi then     //aktualizace dozimetrických hodnot
216         stav:=65;
217         pomocna_promena_3:=false;
218      end_if;
219
220      if kontrolni_stav then
221         if provoz_urychlovace then
222          //kontrolu nelze spustit za běhu urychlovače
223             prechod(provozni:=true);
224         else
225             prechod(kontrolni:=true);
226             test_kompletni:=false;
227             kontrola_stop1:=false;
228             kontrola_stop2:=false;
229             kontrola_nouzove_otevreni:=false;
230             kontrola_koncovy_spinac1:=false;
```

```
231             kontrola_koncovy_spinac2:=false;
232          end_if;
233       end_if;
234
235       if (otevreni or otevreni_virtual) then
236       //pokud bylo stisknuto tlačítko na otevření dveří
237          stav:=90;
238       end_if;
239
240       if zavreni and (not kontrolni_stav) and not
241       ( koncovy_spinac1_zavreno and koncovy_spinac2_zavreno) then
242            //pokud bylo stisknuto tlačítko na zavření dveří
243          stav:=130;
244       end_if;
245
246       if povoleni_k_provozu_virtual or povoleni_k_provozu_tlacitko then
247            //pokus o povolení provozu
248          stav:=50;
249          povoleni_k_provozu_virtual:=false;
250       end_if;
251
252   40: //Kontrolni_stav
253
254       if provozni_stav = true then
255       //kdykoliv lze přejít zpět do provoz. stavu
256              zprava:='';
257              prechod(provozni:=true);
258              test_kompletni:=false;
259       else
260              if havarijni_stav then
261                 prechod(havarijni:=true);
262              end_if;
263
264              zaznmanenanikontrolaR_TRIG (CLK:=test_kompletni);
265              //pokud byl zrovna dokončen test
266              if zaznmanenanikontrolaR_TRIG.Q then
267                 zaznamenanikontrola:=1;
268              end_if;
269
270              if test_kompletni then     //pokud je dokončen test
271                  zprava := 'Kontrola úspěšná,
272                  proveden zápis v provozním deníku.' ;
273              else
274
275                  test_kompletni:=(kontrola_stop1 and kontrola_stop2 and
276                       kontrola_nouzove_otevreni and
277                       kontrola_koncovy_spinac1 and
```

```
278                      kontrola_koncovy_spinac2);
279                  if test_kompletni then
280                     stav:=41;
281                  end_if;
282                  zprava :='Probíhá kontrola funkčnosti prvků systému';
283                  if stop1 then
284                  kontrola_stop1:=true;
285                  end_if;
286                  if stop2 then
287                  kontrola_stop2:=true;
288                  end_if;
289                  if nouzove_otevreni then
290                  kontrola_nouzove_otevreni:=true;
291                  end_if;
292                  if koncovy_spinac1_zavreno then
293                       kontrola_koncovy_spinac1:=true;
294                       end_if;
295                  if koncovy_spinac2_zavreno then
296                       kontrola_koncovy_spinac2:=true;
297                       end_if;
298             end_if;
299      end_if;
300
301  41: //Ready
302      pomocna_kontrola := date_to_dt(dnesni_datum);
303      pomocna_kontrola := ADD_DT_TIME( IN1 := pomocna_kontrola,
304      IN2 := T#540:00:00);
305      pomocna_kontrola := ADD_DT_TIME( IN1 := pomocna_kontrola,
306      IN2 := T#540:00:00);
307      pomocna_kontrola := ADD_DT_TIME( IN1 := pomocna_kontrola,
308      IN2 := T#540:00:00);
309      pomocna_kontrola := ADD_DT_TIME( IN1 := pomocna_kontrola,
310      IN2 := T#540:00:00);
311      kontrola:=dt_to_date(pomocna_kontrola);
312      nutnost_kontroly := false;
313      stav:=40;
314
315  50: //Overeni_podminek
316
317      if (koncovy_spinac1_zavreno) and (koncovy_spinac2_zavreno) and
318      posledni_pamet and (not provoz_urychlovace) then
319         stav:=51;
320      else
321         stav:=52;
322      end_if;
323
324  51: //Povolovani
```

```
325
326         povoleni_k_provozu_pom:=true;
327         prechod(provozni:=true);
328         zprava := 'Povolení k provozu uděleno.';
329         zaznamenanidozi:=1;
330
331     52: // povolení neuděleno
332
333
334         prechod(provozni:=true);
335         zprava := 'Povolení k provozu neuděleno, podmínky udělení
336         nejsou splněny';
337
338     60: //Komunikace_s_TC700          //
339
340         stav := 20;
341             IE_KAnal_IN:=prevod_TC700(R0:=TC700zona[0],R1:=TC700zona[1],
342             R2:=TC700zona[2],R3:=TC700zona[3]);
343             IE_TERC_IN:=prevod_TC700(R0:=TC700zona[4],R1:=TC700zona[5],
344             R2:=TC700zona[6],R3:=TC700zona[7]);
345             IE_CLONA_IN:=prevod_TC700(R0:=TC700zona[8],R1:=TC700zona[9],
346             R2:=TC700zona[10],R3:=TC700zona[11]);
347             IS_IN:=prevod_TC700(R0:=TC700zona[12],R1:=TC700zona[13],
348             R2:=TC700zona[14],R3:=TC700zona[15]);
349             if plcerror>0 then error_komunikace:=true;
350             else error_komunikace:=false;
351             end_if;
352     65: //Komunikace panel
353
354         if DozimetrZona[3]=72 then
355            dozi_pult:=prevod_dozi(H:=DozimetrZona[6],L:=DozimetrZona[7]);
356            dozi_kopka:=prevod_dozi(H:=DozimetrZona[10],L:=DozimetrZona[11]);
357         end_if;
358         if unierror>0 then error_komunikace_dozi:=true;
359         else error_komunikace_dozi:=false;
360         end_if;
361         stav:=20;
362
363     70: //Havarijni_stav
364
365         zvukova_signalizace_err:=true;
366         zprava:='Havarijní stav aktivován,
367         byl proveden zápis do provozního deníku.';
368         posledni_pamet:=false;
369         povoleni_k_provozu_pom:=false;
370
371         if kontrolni_stav then      //deaktivace hav. stavu
```

```
372           prechod(havarijni:=true);
373           zprava:='Přechod pouze do Provozního stavu';
374        end_if;
375
376        if provozni_stav then      //deaktivace hav. stavu
377
378           prechod(inicializace:=true);
379           zprava:='Havarijní stav deaktivován';
380        end_if;
381
382        if otevreni or otevreni_virtual then
383           stav:=100;
384        end_if;
385
386        if stop_motor then
387           stav:=170;
388        end_if;
389
390
391     90: //Povoleni_otevirani
392
393        if provoz_urychlovace=false then
394           stav:=120;
395           otevreni_virtual:=false;
396           dvere(nastavit_otevirani:=true, napajeni:=true,
397           nastavit_zavirani:=false);
398        else
399           prechod(provozni:=true);
400           zprava:='Nelze otevřít dveře, podmínky nejsou
401           splněny (urychlovač aktivní).';
402        end_if;
403
404     120: //Otevirani dveri
405          povoleni_k_provozu_pom:=false;
406          posledni_pamet:=false;
407          if stop_motor then
408            stav:=170;
409          end_if;
410
411          if havarijni_stav then
412                prechod(havarijni:=true);
413          end_if;
414
415          zprava:='Dveře se otevírají.';
416          if koncovy_spinac1_otevreno then     //jakmile jsou dveře otevřeny
417             stav:=170;
418             zprava:='Dveře otevřeny.';
```

```
419         end_if;
420
421     130: //Povoleni_zavirani
422
423         if posledni_pamet then
424            dvere(nastavit_otevirani:=false,
425            napajeni:=true, nastavit_zavirani:=true);
426            stav:=140;
427         else
428             stav:=20;
429             zprava:='Nebylo stisknuto tlačítko poslední,
430             nebo je aktivován kontrolní stav.';
431         end_if;
432
433
434     140: //Zavirani dveri
435         zprava:='Dveře se zavírají';
436         if (koncovy_spinac1_zavreno and koncovy_spinac2_zavreno) then
437            zprava:='Dveře jsou zavřené';
438            stav:=170;
439         end_if;
440
441         if stop_motor then
442           stav:=170;
443         end_if;
444
445     170: //Stop_motor
446         dvere(nastavit_otevirani:=false, napajeni:=false,
447         nastavit_zavirani:=false);
448         if havarijni_stav then
449               prechod(havarijni:=true);
450         else
451               prechod(provozni:=true);
452         end_if;
453
454     end_case;
455
456     povoleni_k_provozu:=povoleni_k_provozu_pom;
457     zvukova_signalizace:=zvukova_signalizace_err;
458     motor_napajeni:=motor_napajeni_pom;
459     motor_otevirani:=motor_otevirani_pom;
460     motor_zavirani:=motor_zavirani_pom;
461     zavrene_dvere_kontrolka:=zavrene_dvere_kontrolka_pom;
462     clovek_v_kopce_kontrolka := not posledni_pamet;
463     step:=step_pom;
464     dir:=dir_pom;
465     enable:=enable_pom;
```

```
466   ena:=ena_pom;
467   enb:=enb_pom;
468
469 END_PROGRAM
```

B Funkční blok pro ovládání čerpadla

```
1   VAR_INPUT
2   napajeni : bool;
3   nastavit_otevirani : bool;
4   nastavit_zavirani : bool;
5   END_VAR
6   VAR_OUTPUT
7   END_VAR
8   VAR_IN_OUT
9   END_VAR
10  VAR
11  END_VAR
12  VAR_TEMP
13  END_VAR
14
15  if napajeni then
16     motor_napajeni_pom:=true;
17  else
18     motor_napajeni_pom:=false;
19  end_if;
20
21  if nastavit_otevirani then
22     motor_zavirani:=false;
23     motor_otevirani:=true;
24  else
25     motor_otevirani:=false;
26  end_if;
27
28  if nastavit_zavirani then
29     motor_otevirani:=false;
30     motor_zavirani:=true;
31  else
32     motor_zavirani:=false;
33  end_if;
34
35
36 END_FUNCTION_BLOCK
```

C Funkce pro změnu stavu systému

```
FUNCTION Prechod :bool
  VAR_INPUT
  havarijni:bool;
  kontrolni:bool;
  provozni:bool;
  inicializace:bool;
  END_VAR
  VAR_IN_OUT
  END_VAR
  VAR
  END_VAR
  VAR_TEMP
  END_VAR

      if havarijni then
         havarijni_stav:=true;
         provozni_stav:=false;
         kontrolni_stav:=false;
         stav:=70;
      end_if;
      if provozni then
         havarijni_stav:=false;
         provozni_stav:=true;
         kontrolni_stav:=false;
         stav:=20;
      end_if;
      if kontrolni then
         havarijni_stav:=false;
         provozni_stav:=false;
         kontrolni_stav:=true;
         stav:=40;
      end_if;
      if inicializace then
         havarijni_stav:=false;
         provozni_stav:=false;
         kontrolni_stav:=false;
         stav:=0;
      end_if;
      prechod:=true;

END_FUNCTION
```