



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV ELEKTROENERGETIKY

DEPARTMENT OF ELECTRICAL POWER ENGINEERING

EFEKTIVNÍ ŘÍZENÍ KOTLE NA TUHÁ PALIVA POMOCÍ PLC

EFFICIENT SOLID FUEL BOILER CONTROL BY PLC

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Josef Pirochta

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Branislav Bátora, Ph.D.

BRNO 2024



Diplomová práce

magisterský navazující studijní program **Elektroenergetika a komunikační technologie**

Ústav elektroenergetiky

Student: Bc. Josef Pirochta

ID: 211435

Ročník: 2

Akademický rok: 2023/24

NÁZEV TÉMATU:

Efektivní řízení kotle na tuhá paliva pomocí PLC

POKYNY PRO VYPRACOVÁNÍ:

- 1) Rešerše problematiky řízení kotlů a řídicích systémů
- 2) Návrh a realizace měření provozních parametrů kotle
- 3) Analýza naměřených dat a návrh řídicího algoritmu pro optimální výkon kotle
- 4) Implementace algoritmu do PLC a ladění v laboratorních podmínkách

DOPORUČENÁ LITERATURA:

doporučená literatura podle pokynů vedoucího závěrečné práce

Termín zadání: 5.2.2024

Termín odevzdání: 20.5.2024

Vedoucí práce: Ing. Branislav Bátora, Ph.D.

doc. Ing. Jaroslava Orságová, Ph.D.

předseda rady studijního programu

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Tato diplomová práce se zaměřuje na efektivní řízení kotle na tuhá paliva pomocí programovatelného logického automatu (PLC). Cílem je vyvinout a optimalizovat řídicí systém, který umožní bezpečný, spolehlivý a efektivní provoz kotle. Práce obsahuje teoretický rozbor bezpečnostních požadavků, popis parametrů kotle VERNER A251 a návrh regulace pomocí PID regulátorů. Výstupem je komplexní softwarový program navržený v prostředí Mosaic, který simuluje provoz kotle ve více scénářích a umožňuje jeho ladění a analýzu. Simulace byly navrženy tak, aby přesně reprezentovaly různé stavy kotle od zapálení až po chybové stavy, čímž poskytují ucelený pohled na celý proces řízení.

KLÍČOVÁ SLOVA

Řízení kotle, Foxtrot PLC, PID regulátor, simulace

ABSTRACT

This thesis focuses on the efficient control of a solid fuel boiler using a programmable logic controller (PLC). The objective is to develop and optimize a control system that will enable safe, reliable and efficient operation of the boiler. The thesis includes a theoretical analysis of the safety requirements, a description of the parameters of the VERNER A251 boiler and the design of the regulation using PID controllers. The output is a complex software program designed in Mosaic environment, which simulates the boiler operation in multiple scenarios and allows its adjustment and analysis. The simulations have been designed to accurately represent the various boiler states from ignition to fault conditions, providing a comprehensive view of the entire control process.

KEYWORDS

Boiler control, Foxtrot PLC, PID controller, simulation

PIROCHTA, Josef. *Efektivní řízení kotle na tuhá paliva pomocí PLC*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav elektroenergetiky, 2024, 79 s. Diplomová práce. Vedoucí práce: Ing. Branislav Bátora, Ph.D.

Prohlášení autora o původnosti díla

Jméno a příjmení autora: Bc. Josef Pirochta
VUT ID autora: 211435
Typ práce: Diplomová práce
Akademický rok: 2023/24
Téma závěrečné práce: Efektivní řízení kotle na tuhá paliva pomocí PLC

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

podpis autora*

*Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu mé diplomové práce, Ing. Branislavu Bátorovi, Ph.D., za jeho metodickou, pedagogickou a odbornou pomoc a cenné rady, které mi poskytl při zpracování tohoto projektu. Díky patří také Ing. Otakaru Štelcovi za umožnění přístupu ke kotli, na kterém probíhalo měření. Zvláštní poděkování také náleží Bc. Radomilu Holcovi za jeho odborné konzultace a rady během vývoje softwaru.

Obsah

Úvod	13
1 Bezpečnostní požadavky	14
1.1 Úvod	14
1.2 Bezpečnost proti zpětnému hoření	14
1.3 Tepelná vodivost	15
1.4 Šíření požáru do palivového potrubí nebo v integrovaném zásobníku	15
1.5 Prevence přenosu toxických složek	16
1.6 Zařízení pro regulaci a omezení teploty	16
1.6.1 Obecně	16
1.6.2 Regulační a omezovací zařízení teploty pro otevřené odvětrávací systémy	16
1.6.3 Regulační a omezovací zařízení teploty pro uzavřený odvětrávaný systém	17
1.6.4 Zařízení pro odvod přebytečného tepla	17
1.7 Příslušenství topných kotlů	17
1.7.1 Obecně	17
1.7.2 Elektrická bezpečnost Elektrická bezpečnost	18
2 Požadavky na výkon	19
2.1 Obecně	19
2.2 Energetická účinnost	20
2.2.1 Obecně	20
2.2.2 Tah	21
2.2.3 Doba spalování	21
2.2.4 Minimální tepelný výkon	21
2.3 Emisní limity	22
2.3.1 Emisní limity při jmenovitém tepelném výkonu a minimálním tepelném výkonu	22
3 Kotel VERNER A251	23
3.1 Kotel	23
3.2 Spalování	24
3.3 Pohon mechanických částí	24
3.4 Zásobník na palivo	24
3.5 Hasící zařízení	24
3.6 Přívod vzduchu	24

3.7	Turbulátory	25
3.8	Parametry kotle	25
4	Regulace	26
4.1	Ekvitermní regulace	27
4.2	Řízení kotle	28
4.2.1	Zážeh	28
4.2.2	Provoz	28
4.2.3	Odstavení kotle	29
4.2.4	Nastavení kotlového čerpadla	29
4.2.5	Ochranná zařízení	29
5	Výběr hardwaru	30
5.1	PLC Foxtrot CP-2000	30
5.2	PT-1000	31
5.3	EcoLambda ver. 2	32
6	PID regulátory	33
6.1	Princip funkce	33
6.2	Nastavení a ladění PID regulátoru	33
6.3	Účinky jednotlivých složek na systém	33
6.4	Příklady praktického použití	34
7	Měření a analýza dat	35
7.1	Výsledky měření	38
8	Program	41
8.1	Programovací prostředí Mosaic	41
8.2	Struktura	41
8.2.1	Hlavní cyklus PLC prgMain	42
8.2.2	CyclicInterrupt_400ms	43
8.3	Definování proměnných	44
8.4	Popis kódu	49
8.4.1	Program prgMain	49
8.4.2	Funkční blok Boiler_StateMachine	50
8.4.3	Funkční blok Boiler_Control	57
8.4.4	Program CyclicInterrupt_400ms	58
8.4.5	Funkční blok PID_C_Regulator	59

9 Simulace	63
9.1 Vývojové diagramy	63
9.2 Funkční blok Boiler_Simulation	66
9.3 Simulační prostředí	69
9.3.1 Automatické řízení kotle	70
9.3.2 Manuální řízení kotle	71
9.3.3 Nastavení časových intervalů	72
Závěr	73
Literatura	74
Seznam symbolů a zkratk	75
Seznam příloh	77
A Program pro řízení kotle	78
B Program pro týdenní měření	79

Seznam obrázků

4.1	Ekviterní křivka [2]	27
4.2	Ekviterní Regulace [8]	28
5.1	PLC Foxtrot CP-2000 [9]	31
5.2	Princip zapojení Lambda sondy [5]	32
7.1	Rozvaděč pro měření příkonu kotle	35
7.2	Přiřazení modulů CFOX C-EM-0300M v prostředí Mosaic	36
7.3	Struktura pro Datalog0	36
7.4	Okno kolekce	37
7.5	Okno signál	37
7.6	Průběh provozu kotle za dobu 20 minut	38
7.7	Průběh provozu kotle za dobu 14 hodin	39
7.8	Průběh provozu kotle za celý týden	39
8.1	Struktura programu	41
8.2	Průběh jednoho cyklu prgMain programu	43
8.3	Průběh jednoho cyklu CyclicInterrupt_400ms programu	44
8.4	Režimy a stavy kotle	45
8.5	Proměnné pro nastavení teplotních a časových parametrů	46
8.6	Proměnné pro sledování aktuálních hodnot	47
8.7	Manuální mód kotle	47
8.8	Výstupy a vstupy proměnných pro simulace	48
8.9	Globální proměnné	49
8.10	Hlavní program prgMain	50
8.11	Vstupní a výstupní proměnné	50
8.12	Definování časovačů	51
8.13	Podmínky pro časovače	52
8.14	Podmínky pro resetování	52
8.15	Aktuální krok = čekání	53
8.16	Aktuální krok = zapalování	53
8.17	Podmínka pro zapnutí zapalovací tyče	53
8.18	Zapalovací tyč zapnuta	54
8.19	Detekce zapálení	54
8.20	Vypnutí zapalovací tyče a časovačů	54
8.21	Logika časovačů při spuštění kotle	55
8.22	Kontrolní podmínky během provozu kotle	55
8.23	Zastavení kotle	56
8.24	Vypnutí ventilátoru po zpoždění	56
8.25	Chybový stav	56

8.26	Plnění kotle palivem	57
8.27	Odpopelování kotle	58
8.28	PID regulátor pro teplotu spalin	58
8.29	PID regulátor pro teplotu vody	58
8.30	PID regulátor - konstanty	59
8.31	Automatický mod pro PID regulaci	60
8.32	Výpočet mezí integrace pro PID regulator	60
8.33	Mechanismus anti-wind-up pro PID regulátor	61
8.34	Výstupní meze PID regulátoru	61
9.1	Obecný vývojový diagram pro řízení kotle	63
9.2	Vývojový diagram pro proces zapálení	64
9.3	Vývojový diagram pro proces provozu	65
9.4	Simulace doplňování paliva	66
9.5	Výpočet a nastavení časovačů	67
9.6	Simulace poklesu teploty	68
9.7	Simulace růstu teploty během zapalování kotle	68
9.8	Simulace změny teplot během běhu kotle	69
9.9	Režimy	70
9.10	Simulace automatického řízení kotle	71
9.11	Simulace manuálního řízení kotle	72
9.12	Nastavení simulací kotle	72

Seznam tabulek

2.1	Srovnání účinností jednotlivých technologií pro komerční použití [1]. . .	19
2.2	Přehled energetické účinnosti [1]	20
2.3	Emisní limity [1]	22
3.1	Parametry pro kotel A251 [10]	25

Úvod

Diplomová práce se zaměřuje na vývoj programu pro řízení kotle na tuhá paliva v prostředí Mosaic, což je platforma využívaná pro řízení chytrých domácností. Původní zadání práce se během tvorby změnilo z implementace programu do reálného provozu na simulaci provozu kotle pro různé scénáře.

Teoretická část práce obsahuje podrobnou rešerši o stávajících metodách řízení kotlů, která poskytuje důkladný základ pro pochopení potřebných aspektů pro vývoj řídicího systému. V této části jsou diskutovány různé přístupy k regulaci teploty, efektivitě spalování a bezpečnostním aspektům provozu kotlů.

Praktická část práce je zahájena sběrem dat o řízení kotle Verner A251, což umožňuje získání podrobných informací o jeho provozu a efektivitě. Tyto data jsou využity pro vyvinutí nového řídicího systému, který cílí na optimalizaci a zvýšení účinnosti provozu kotle. Protože implementace v reálném kotli není možná, jsou vyvinuty simulace, které simulují různé provozní stavy a umožňují ověřit funkčnost a efektivitu nového systému.

Tyto simulace poskytují cenné poznatky o chování kotle v různých provozních režimech a umožňují identifikaci potenciálních problémů a jejich řešení před reálným nasazením. Díky simulacím je možné testovat kotel pod různým zatížením a při různých teplotách, což přispěje k lepšímu porozumění jeho chování. Celkově tato práce přináší kompletní řešení pro efektivní řízení kotlů na tuhá paliva.

Diplomová práce tak poskytuje ucelený pohled na vývoj řídicího systému kotle v prostředí Mosaic. Od počáteční analýzy řízení kotle až po návrh simulace a softwarové implementace tato práce přináší podrobný návod na vytvoření efektivního systému pro řízení kotlů na tuhá paliva.

1 Bezpečnostní požadavky

1.1 Úvod

Bezpečnostní hlediska při provozu kotlů, včetně jejich roznětných systémů a příkladacích zařízení, jsou klíčové pro minimalizaci rizika možných nebezpečných situací. Zajištění bezpečnosti vyžaduje kombinaci konstrukčních prvků a použití bezpečnostních zařízení. Tyto prvky musí být navrženy tak, aby byla zachována bezpečnost i v případě poruchy samotných bezpečnostních zařízení.

Řídicí funkce jsou klasifikovány následovně:

Třída A: Funkce řízení, na které nelze spoléhat z bezpečnostního hlediska v aplikaci.

Třída B: Ovládací funkce, které mají zabránit vzniku nebezpečného stavu spotřebiče. Selhání řídicí funkce nevede přímo k nebezpečné situaci. U zařízení používaných v ovládací funkci třídy B se provádí hodnocení jediné poruchy zařízení včetně použití softwaru třídy B podle normy ČSN EN 60730-1.

Třída C: Řídicí funkce, které jsou určeny k prevenci zvláštních nebezpečí, jako je výbuch, nebo jejichž selhání by mohlo přímo způsobit nebezpečí ve spotřebiči. U zařízení používaných v řídicí funkci třídy C se vyžaduje druhé posouzení poruchy zařízení včetně použití softwaru třídy C podle normy ČSN EN 60730-1.

V případě bezpečnostních rutin realizovaných v programovatelném logickém řízení musí software splňovat požadavky příslušné třídy softwaru B nebo C. V posouzení rizik se každému zjištěnému nebezpečí musí přiřadit jedna z výše uvedených klasifikací řídicí funkce. Spuštění jakékoli řídicí funkce třídy B nebo C musí vést alespoň k přerušení přívodu paliva. V této práci bude algoritmus navržen tak, aby splňoval požadavky Třídy B.

Toto posouzení rizik musí zahrnovat alespoň následující: provoz kotle, dohled nad plamenem, průtok spalin, řízení potřeby tepla a řízení spalování [1].

1.2 Bezpečnost proti zpětnému hoření

Systémy automatického přikládání musí být navrženy tak, aby se zabránilo zpětnému hoření. Nebezpečí zpětného hoření je klasifikováno jako riziko odpovídající stupni bezpečnosti C. Souvisí s hnacími silami tepelnou vodivostí, zpětným prouděním zápalných plynů a zpětným šířením požáru. Zpětnému hoření se musí zabránit konstrukčními prostředky a zavedením jednoho nebo více bezpečnostních zařízení proti zpětnému hoření [1].

Toto posouzení rizik musí zahrnovat alespoň následující:

- Pracovat vždy na principu uzavřeného proudového okruhu

- Zabránit zpětnému hoření ve stavu ztráty napájení
- Zabránit zpětnému hoření ve stavu poruchy podavače paliva nebo přerušení dodávky paliva

1.3 Tepelná vodivost

Povrchová teplota podavače paliva do kotle (bez izolace) nebo integrovaného zásobníku nesmí v žádném provozním stavu nebo v případě poruchy překročit 85 °C. Pokud je toto kritérium splněno konstrukčními prostředky, není nutné žádné další bezpečnostní zařízení [1].

Přijatá řešení pro zabránění přehřátí dopravníku paliva v důsledku tepelné vodivosti jsou následující:

- Hasicí zařízení, např. vodní rozstřikovací systém, a bezpečnostní omezovač teploty nastavený na maximálně 95 °C.
- Nouzové vyprazdňovací zařízení, které vyprazdňuje dopravník paliva, aniž by došlo k přetopení kotle, který reaguje při teplotě nižší než 95 °C.
- Dopravník paliva, který je chlazen vodním okruhem a teplota vody je omezena pojistkou (např. vodní okruh je součástí kotlového okruhu).

Přijatá řešení pro zabránění přehřátí v integrovaném zásobníku paliva v důsledku tepelné vodivosti v kombinaci s přijatelnými řešeními pro zařízení pro přikládání jsou:

- Hasicí zařízení přímo v zásobníku paliva, např. vodní rozstřikovací systém a STB (Bezpečnostní omezovač teploty) nastavený na maximální teplotu 95 °C.
- Dostatečná izolace zásobníku paliva od horkých částí kotle.
- Přirozeně větraný prostor mezi zásobníkem paliva a kotlem.

1.4 Šíření požáru do palivového potrubí nebo v integrovaném zásobníku

Šíření požáru do palivového potrubí nebo do interní násypky musí být zabráněno v jakémkoli provozním stavu nebo v případě jakékoli poruchy. To nezahrnuje tepelnou reakci malého množství paliva na konci příkladacího zařízení, pokud nedojde k další reakci do palivového potrubí [1].

Mezi indikace významného šíření požáru může patřit např.:

- Zvýšení teploty v palivovém potrubí o více než 20 °C oproti normálnímu provozu.
- Teplota vyšší než 85 °C na povrchu palivového potrubí nebo integrovaného zásobníku.

- Nahromadění kouře v integrovaném zásobníku.

1.5 Prevence přenosu toxických složek

Složky spalin, které jsou toxické, nesmí v žádném provozním stavu kotle pronikat do provozního prostoru ani přecházet do palivového potrubí v nebezpečném množství. Konstrukční a provozní podmínky musí zajistit dostatečnou plynotěsnost kotle v kombinaci s větráním provozní místnosti a u kotlů s automatickým přikládáním musí být zabráněno jakémukoli významnému proudění spalin s možným obsahem CO nebo jiných toxických složek do palivového potrubí [1].

1.6 Zařízení pro regulaci a omezení teploty

1.6.1 Obecně

Regulační a bezpečnostní zařízení, jakož i příslušné možnosti instalace musí být k dispozici pro každý kotel v závislosti na typu spalovacího systému a typu ochrany zařízení, ve kterém má být kotel namontován. Zařízení požadovaná v každém případě dodává výrobce kotle spolu s kotlem. Pokud není vybavení dodáno, musí být v návodu k instalaci uvedeny přesné specifikace, zejména mezní hodnoty a časové konstanty pro bezpečnostní omezovač teploty. Zařízení pro omezení teploty musí zajistit, aby maximální teplota vody v kotli byla 110 °C [1].

1.6.2 Regulační a omezovací zařízení teploty pro otevřené od- vzdušňovací systémy

Při použití ve fyzicky chráněných topných zařízeních (teplota je omezena tlakem v instalaci) musí být k dispozici následující zařízení podle požadavků normy ČSN EN 12828:2012+A1:2014, bezpečnostní omezovače teploty podle normy ČSN EN 14597:2012:

- Regulátor teploty
- Bezpečnostní omezovače teploty (ruční resetování)

Bezpečnostní omezovač teploty je elektronický termostat sledující teplotu topných zařízení a strojů. Při překročení povoleného teplotního limitu nebo v případě poruchy se okamžitě vypne. Použití není nutné v případech, kdy spalovací systém není ani rychle, ani částečně odpojitelý. V těchto případech (např. u kotlů bez automatického nuceného tahu) je přebytečné teplo odváděno ve formě páry otevřeným odvětrávaným spojením s atmosférou [1].

1.6.3 Regulační a omezovací zařízení teploty pro uzavřený odvětrávaný systém

Při použití v termostaticky chráněných topných zařízeních musí být topný systém buď rychle, nebo částečně odpojitelný a teplo nebo zbytkový tepelný výkon, který není absorbován topným systémem, musí být spolehlivě odváděn pomocí bezpečnostního výměníku tepla nebo rovnocenných zařízení. V souladu s tím je třeba rozlišovat následující varianty zařízení podle požadavků normy ČSN EN 12828:2012+A1:2014, bezpečnostní omezovač teploty podle normy ČSN EN 14597:2012:

Systém spalování je rychle odpojitelný; nezbytné vybavení se skládá z:

- Regulátoru teploty
- Bezpečnostního omezovače teploty (ruční resetování)

Systém vypalování je částečně odpojitelný; nezbytné vybavení se skládá z:

- Regulátoru teploty
- Bezpečnostního omezovače teploty (ruční resetování)

Topný systém není odpojitelný a jmenovitý tepelný výkon je <100 kW; nezbytné vybavení se musí skládat z:

- Regulátoru teploty
- Bezpečnostního zařízení s termickým odváděním tepla v souladu s kapitolou 1.6.4 pro odvádění maximálního možného tepelného výkonu v případě poruchy [1].

1.6.4 Zařízení pro odvod přebytečného tepla

Bezpečnostní výměník tepla nebo jiná zařízení pro odvod přebytečného tepla musí zajistit, aby nebyla překročena maximální teplota vody v kotli 110 °C. K tomuto účelu se použije například bezpečnostní zařízení pro odvod tepla, jako je STW (Bezpečnostní hlídač teploty) typu Th (termostatický) podle EN 14597:2012, v kombinaci s výměníkem tepla integrovaným v kotli. Mezi přípustné výměníky tepla patří zásobníkové nebo oběhové ohřívače vody, pokud jsou navrženy a dimenzovány tak, aby bylo možné předávat teplo bez dalších pomocných zařízení a vnější energie. Pevně zabudované oběhové ohřívače vody nelze použít jako provozní ohřívače vody, ale pouze jako bezpečnostní výměníky tepla [1].

1.7 Příslušenství topných kotlů

1.7.1 Obecně

Pokud je kotel z výroby vybaven dalším příslušenstvím, které je třeba servisovat pro zajištění jeho správné funkce a bezpečnosti kotle, musí konstrukce zajistit snadný

přístup bez nutnosti rozsáhlých demontážních prací [1].

1.7.2 Elektrická bezpečnost Elektrická bezpečnost

Elektrická bezpečnost kotle musí odpovídat normě EN 60335-2-102. Pro abnormální provoz jako poruchový stav podle normy EN 60335-2-102:2016 (porucha integrovaných obvodů) se za relevantní považují pouze výstupní signály, které způsobují pouze jednu poruchu v jednom akčním členu. Kombinace výstupních signálů, které způsobují poruchu ve více než jednom akčním členu, se nepovažují za relevantní ve smyslu abnormálního provozu, protože je nepravděpodobné, že by mohla nastat nějaká nebezpečná situace [1].

2 Požadavky na výkon

2.1 Obecně

Následující výkonnostní požadavky se posuzují při zkouškách s použitím příslušných zkušebních paliv uvedených v tabulce 2.1. Tato paliva musí být vybrána tak, aby představovala doporučená paliva, o které může kotel spalovat. Jmenovitý tepelný výkon a rozsah tepelného výkonu se mohou lišit v závislosti na palivu. U kotlů se tepelný výkon, emise a účinnost kotle měří při jmenovitém a minimálním tepelném výkonu.

Tab. 2.1: Srovnání účinností jednotlivých technologií pro komerční použití [1].

	Černé uhlí	Hnědé uhlí (vč. briketového)		Dřevěné brikety	Piliny	
Vlhkost	< 11 %	< 20 %	< 15 % Brikety	< 10 %	35 % až 50 %	
Obsah popela	2 % až 7 %	5 % až 20 %		< 0,7 %	< 0,5 %	
Těžké látky	15 % až 30 %	>30 %	40 % až 50 %	50 % až 60 %	-	-
Výhřevnost NCV	>28 MJ/kg	>12,5 MJ/kg		>17 MJ/kg	>17 MJ/kg	
Hrubá výhřevnost GCV	>28 MJ/kg	>14 MJ/kg		>19 MJ/kg	>19 MJ/kg	

Vlhkost: Udává množství vody obsažené v palivu, vyjádřené jako procento hmotnosti. Nižší vlhkost znamená vyšší energetickou účinnost, protože méně energie se ztrácí při odpařování vody.

Obsah popela: Znamená podíl popela, který zůstane po úplném spálení paliva. Vyšší obsah popela vede k větší produkci zbytků a může způsobit častější čištění kotle.

Těkavé látky: Zahrnují organické složky v palivu, které se při zahřívání rychle odpařují nebo vypařují. Vysoký obsah těkavých látek obvykle znamená lepší zápalnost a vyšší produkci plamenů při spalování.

Výhřevnost NCV (Net Calorific Value, čistá výhřevnost): Udává množství tepla, které lze získat z paliva při spalování, pokud se nebere v úvahu kondenzace vodní páry. Je to energie skutečně dostupná k vytápění.

Hrubá výhřevnost GCV (Gross Calorific Value, hrubá výhřevnost): Zahrnuje celkové množství tepla, včetně kondenzace vodní páry. Poskytuje vyšší hodnotu než čistá výhřevnost, ale neodráží přesně energii, kterou lze přímo využít [1].

2.2 Energetická účinnost

2.2.1 Obecně

Energetická účinnost kotle je popsána různými charakteristikami, které se vypočítávají a měří, jak je uvedeno v tabulce 2.2.

Tab. 2.2: Přehled energetické účinnosti [1]

Charakteristika	Metoda stanovení	NCV/GCV	Účel
Účinnost kotle při jmenovitém a částečném zatížení	Měření QB a P	NCV	Informace v souladu s existujícími údaji kotlů a výpočet dalších charakteristik
Užitečná účinnost kotle při jmenovitém a částečném zatížení	Měření QB a P Přepočet QB na základě GCV	GCV počítáno nebo měřeno	Informační údaje pro design ECO a označení účinnosti
Sezónní vytápění prostor	Výpočet z užitečných účinností a dalších parametrů (pohotovostní režim, elektrický výkon, ...)	GVC	Požadavek na návrh ECO
Index energetické účinnosti			Štítek

Tato tabulka 2.2 poskytuje přehled o energetické účinnosti kotle a metodách stanovení.

Metoda stanovení

- **Měření QB a P:** QB je tepelný výkon kotle a P je tlak. Tyto hodnoty se měří pro určení účinnosti.
- **Měření QB a P přepočet QB na základě GCV:** Hrubá výhřevnost (GCV) poskytuje vyšší hodnoty, které se přepočítávají na skutečnou užitkovou účinnost.
- **Výpočet z užitkových účinností a dalších parametrů:** Zohledňuje další faktory, jako je režim pohotovosti a elektrický výkon.

NCV/GCV

- **NCV:** Net Calorific Value, čistá výhřevnost. Udává energii dostupnou při spalování bez kondenzace vodní páry.
- **GCV:** Gross Calorific Value, hrubá výhřevnost. Zahrnuje kondenzaci vodní páry a poskytuje vyšší hodnotu.

Účel

- **Informace v souladu s existujícími údaji kotlů:** Výpočet dalších charakteristik pro ekologický design.
- **Požadavek na návrh ECO:** Vytváření štítku pro splnění ekologických požadavků kotle.

2.2.2 Tah

Minimální tah spalin, uvedený v normě EN 13384-1, je důležitý pro zajištění správného provozu kotle. Tah je rozdíl v tlaku mezi komínem a okolním prostředím, který umožňuje přirozený pohyb spalin směrem ven přes komín.

Měření tahu spalin se provádí pomocí tlakoměru nebo speciálního přístroje navrženého k tomuto účelu. Tlakový rozdíl se obvykle vyjadřuje v pascálech (Pa) nebo milimetrech vodního sloupce (mm H₂O). Pro přesné měření se měřicí přístroj umísťuje do vhodného bodu komínu nebo kouřovodu a tlak se zaznamenává v průběhu provozu kotle, aby se zjistilo, zda je tah v souladu s doporučeními normy.

Sledovanými veličinami jsou především tlak (rozdíl tlaku mezi komínem a okolním prostředím), teplota spalin, která ovlivňuje tah a jeho změny, a také rychlost proudění spalin komínem. Výrobce má povinnost stanovit minimální hodnotu tahu pro optimální a bezpečný provoz kotle [1].

2.2.3 Doba spalování

Doba spalování u kotlů s ručním a automatickým přikládáním při jmenovitém tepelném výkonu musí být uvedena v technické dokumentaci. Minimální doba [1]:

- pro biogenní a jiná tuhá paliva musí být doba spalování 2 h.
- 4 h pro fosilní paliva.

2.2.4 Minimální tepelný výkon

U kotlů s automatickým přikládáním nesmí minimální tepelný výkon překročit 30 % jmenovitého tepelného výkonu. Tento požadavek na omezení tepelného výkonu musí být dosažen automaticky pomocí regulačního zařízení.

Automaticky roztápěné kotle se jmenovitým tepelným výkonem < 70 kW, které nejsou určeny pro provoz při 30 % jmenovitého tepelného výkonu nebo méně, musí být provozovány a zkoušeny s akumulací nádrží. Objem nádrže musí být uveden jako minimální objem v technické dokumentaci včetně schématu popisujícího požadavek na hydraulické připojení k akumulací nádrži. Kontrola přívodu paliva a/nebo vzduchu může být buď nepřetržitá, nebo přerušovaná [1].

2.3 Emisní limity

2.3.1 Emisní limity při jmenovitém tepelném výkonu a minimálním tepelném výkonu

Spalování musí být nízkoemisní. Tento požadavek je splněn, pokud nejsou překročeny emisní hodnoty uvedené v tabulce 2.3 při provozu při jmenovitém tepelném výkonu nebo v případě kotlů s rozsahem tepelného výkonu při provozu při jmenovitém tepelném výkonu a minimálním tepelném výkonu [1].

Tab. 2.3: Emisní limity [1]

Příkládání	Palivo	Jmenovitý tepelný výkon kW	Emisní limity								
			CO			OGC			Prach		
			mg/m ³ na 10 % O ₂								
			Třída	Třída	Třída	Třída	Třída	Třída	Třída	Třída	Třída
Automatické	Biomasa	≤ 50	3000	1000	500	100	30	20	150	60	40
		> 50 ≤ 150	2500			80			150		
		> 50 ≤ 500	1200			80			150		
	Fosilní	≤ 50	3000			100			125		
		> 50 ≤ 150	2500			80			125		
		> 50 ≤ 500	1200			80			125		

3 Kotel VERNER A251

Automatický teplovodní kotel VERNER se využívá pro vytápění rodinných domů, bytových jednotek, dílen, institucí a podobných objektů. Spaluje dřevěné a rostlinné pelety, zbytky zrn o průměru 6–14 mm (obiloviny) - pšenice, ječmen, oves, kukuřice.

Palivo se dopravuje z násypky do spadu pomocí šnekového dopravníku, pak padá do hořáku. Pro rovnoměrné spalování paliva se používá rozptylovač, nespalitelné zbytky se pomocí roštnic odstraní se do žlabu se šnekovým dopravníkem ten přesune popel do sběrného kontejneru. Výměník zajišťuje předání tepla spalin do topné vody. Primární vzduch potřebný pro spalování se dodává do paliva spárami v roštu, sekundární vzduch se dopravuje otvory v zadní stěně v hořáku a spadem paliva. K těmto účelům slouží přetlakový ventilátor.

Výhody automatického kotle VERNER:

Regulovatelnost kotle

Kotel je vybaven moderním elektronickým regulátorem pro řízení provozu kotle, distribuuje palivo a reguluje otáčky ventilátoru.

Vysoká účinnost

Kotel splňuje nejpřísnější emisní požadavky evropských norem. Vysoké účinnosti je dosaženo v důsledku pohyblivých turbulátorů a velké ploše spalinového výměníku.

Pohodlná obsluha

Ruční roztápění není zapotřebí, díky automatickému horkovzdušnému roztápění kotle. Kotel zajišťuje přes pokojový termostat požadovanou teplotu pro vytápění objektu a přes přídatné teplotní čidlo řídí dohřev boileru nebo akumulární nádrže.

Ke komfortu obsluhy přispívá i velký zásobník. Palivo proto stačí natankovat jednou za 1 až 3 dnů (v závislosti na spotřebované energii).

Kotel umožňuje připojení zařízení pro externí ovládání (např. přes mobilní telefon).

Kotel je možné doplnit o další zařízení na plnění a odstraňování popela [10].

3.1 Kotel

Šnekový podavač zajišťuje dodávku paliva do kotle, kde pomocí speciálního hořáku proběhne spalovací proces. Samočinné roštování umožňuje spalování paliv s vyšší spékavostí popela. Pomocí přetlakových ventilátorů je zajištěna dodávka spalovacího vzduchu.

Stěny kotle, které přijdou do styku s plamenem jsou dvojité z ocelových plechů o tloušťce 4 až 5 mm a chlazené vodou. Hořákový prostor se nachází se spodní části a žárotrubný spalinový výměník je v části horní. Spalinový výměník se skládá z 42 trubek o světlosti 50 mm [10].

3.2 Spalování

K obložení hořáku se používá speciální keramika. Pro správnou funkci je potřeba zajistit cirkulaci vzduchu, k tomu se využívají 3 otvory zapalovacího vzduchu, které se nachází ve spodní části a 2 otvory sekundárního vzduchu v horní části. Na dně je pohyblivý rošt, popel se vybírá otvorem ve vertikální stěně pod výstupem z hořáku. Kotel má automatické odpopelení, obsahuje šnekový dopravník a vybírací otvor je utěsněn samostatným víkem. V zadní části stropu spalovacího prostoru se nachází svislý spad přívodu paliva [10].

3.3 Pohon mechanických částí

V kruhové trubce je umístěn šnekový dopravník s násypným hrdlem. Pákový mechanismus pohonu roštnic se nachází pod šnekovým dopravníkem. Prostřednictvím kulových čepů je propojen s táhlem roštnic, na druhé straně je náboj s excentrem připojený k hřídeli dopravníku. Ochrana před opalování roštnic je zajištěna koncovým spínačem, který zastaví v roštnice v zadní poloze. Pohon mechanických částí zajišťuje převodovka s elektromotorem [10].

3.4 Zásobník na palivo

Uzavírá se vzduchotěsným víkem, které je opatřeno bezpečnostním spínačem. Při otevření dveří se kotel odstaví z provozu. Zásobník je svařen z ocelových plechů, ve spodní části zadní stěny je víčko, které umožňuje zásah v prostoru šnekového podavače [10].

3.5 Hasící zařízení

Je přichyceno na víčku zásobníku. Hadice z hasící nádoby vede do násypného hrdla dopravníku. Vosková zátka v nátrubku se v případě prohoření paliva do násypky roztaví a voda z nádoby zaplaví hrdlo podavače [10].

3.6 Přívod vzduchu

Přívod vzduchu zajišťuje obdélníkové těleso připojené ke kotli. Skládá se z ventilátoru se zpětnou klapkou, dvou regulačních clon primárního vzduchu a dvou regulačních clon sekundárního vzduchu a komory pro ohřev zapalovacího vzduchu, ve které je umístěna spirálová tyč [10].

3.7 Turbulátory

Nachází se ve spalínovém výměníku a jsou umístěny na pohyblivém držáku s ovládací pákou. Umožňují čištění trubkového výměníku z čela kotle. Turbulátory je nutné pootočit a povytáhnout, aby se vyčistily stěny trubek. Popílek vyhoří na cihlách katalyzátoru [10].

3.8 Parametry kotle

Tab. 3.1: Parametry pro kotel A251 [10]

Palivo	A251		
	Pelety dřevní průměr 6–14 mm	Pelety rostlinné průměr 6-14 mm	Obilí zrn (ječmen, oves žito, pšenice, kukuřice..)
Jmenovitý výkon	25 kW	25 kW	25 kW
Regulovatelnost			
- kontinuálním provozem	7,5-28 kW	7,5-28 kW	7,5-26 kW
- elektronicky řízeným odstávkovým režimem	0 – 7,5 kW	0 – 7,5 kW	0 – 7,5 kW
Účinnost	92 %	92 %	92%
Spotřeba paliva při jmenovitém výkonu	5,8 kg/h	6,3 kg/h	6,8 kg/h
Maximální provozní přetlak	3 bary (0,3 Mpa)		
Objem vodní náplně	85 litrů		
Objem násypky	240 dm ³		
Objem popelníku	18 dm ³		
Teplota spalin na výstupu při jmenovitém výkonu	160 °C		
Minimální teplota vratné vody v provozu	60 °C		
Připojovací nátrubky	G 5/4"i		
Celková hmotnost	575 kg		
Maximální hladina hluku	54 dB		
Předepsaný provozní tah komína	15 – 30 Pa		
Přívodní napětí	230 V / 50 Hz		
Maximální elektrický příkon (při zapalování)	1500 W		
Průměrný příkon při provozu	120W		
Druh krytí elektrických součástí	IP41		
Prostředí	základní AA5 / AB5 dle ČSN 332000-3		
Třída účinnosti	3	3	3
Emisní třída	3	3	3
Doba hoření 1 násypky při jmen. výkonu	30 hod.	28 hod	27 hod
Hmotnostní tok spalin při jmen. výkonu	0.016 kg/s	0.016kg/s	0.016kg/s

Konkrétní údaje o spotřebě elektrické energie kotle během zapalovacího procesu i při provozu jsou uvedeny v kapitole 7.

4 Regulace

Používá se k zajištění automatického provozu kotle, zajišťuje i roztápění a odstavení kotle. Nachází se na čelní stěně kotle společně s řídicí mikroprocesorovou jednotkou, vstupy měřených signálů, elektrickými výstupy, pojistkami a nezávislým bezpečnostním mechanickým termostatem se světelnou signalizací. K obsluze kotle je použit 4 řádkový podsvícený displej a tlačítkové ovládání [10].

Základní vstupy regulátoru:

- Teplota vody na výstupu z kotle
- Teplota spalin
- Signál z bezpečnostního čidla poklopu násypky
- Signál z koncového čidla roštovacího mechanismu

Základní výstupy regulátoru:

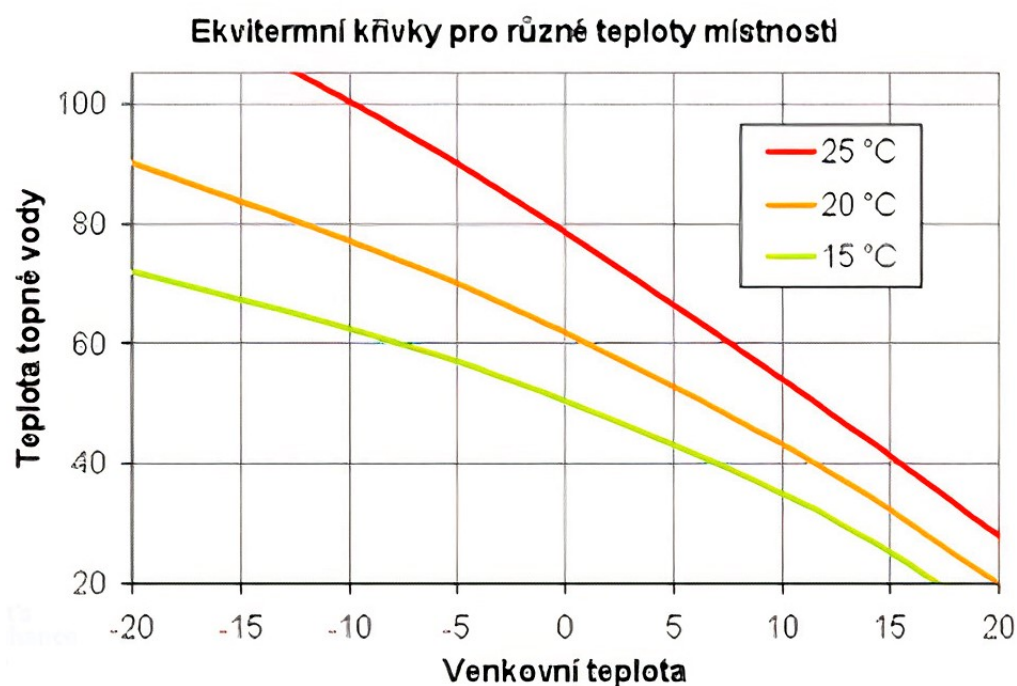
- Řízení elektromotoru pro funkci dopravníku
- Regulace otáček ventilátoru spalovacího vzduchu
- Zapalovací elektrická tyč
- Čerpadlo (topné soustavy a kotlového okruhu)

Regulátor umí pracovat s těmito typy doplňkového vybavení:

- Termostaty
- Čerpadla (systémové, kotlové)
- Čidlo teploty vody
- Lambda sonda
- Externí ovládání kotle (zapnutí, vypnutí, signalizace poruch)
- Odpopelovací zařízení
- Řídicí jednotka doplňovacího zařízení

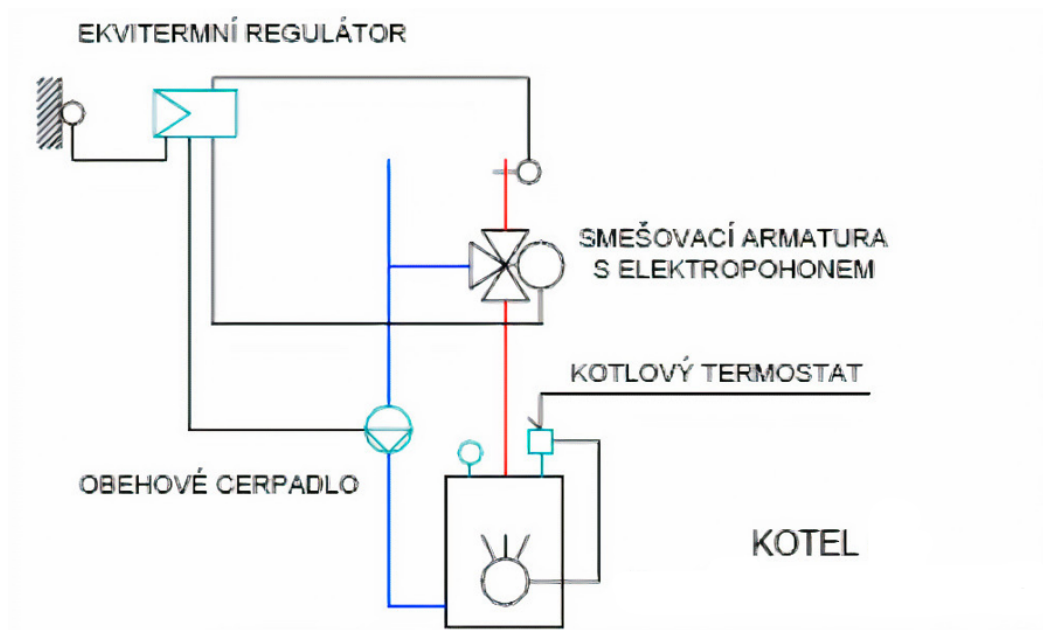
4.1 Ekvitermní regulace

Ekvitermní regulace je regulace teploty v místnosti na základě venkovní teploty. Aby teplota v místnosti zůstávala konstantní je důležité při nižší venkovní teplotě přivádět vodu o vyšší teplotě. Soustava ekvitermní křivek popisuje závislost venkovní teploty, místnosti a teploty topné vody, pro každou místnost se stanovuje zvlášť. Ekvitermní křivky se stanovují na základě nejnižší dosažené venkovní teploty v dané oblasti, dále na požadované teplotě v místnosti (typicky 15 až 25 °C) a typických průbězích závislosti teploty topné vody na venkovní teplotě [2].



Obr. 4.1: Ekvitermní křivka [2]

Regulátor na základě porovnání teplot (venkovní, oběhové vody a v místnosti) a ekvitermních křivek zasílá příkaz pohonu směšovací armatury k jejímu otevírání nebo přivírání [8].



Obr. 4.2: Ekviterní Regulace [8]

4.2 Řízení kotle

4.2.1 Zážeh

K zažehnutí dochází, pokud je teplota vody v kotli nižší než teplota nastavená na bojleru či pokojovém termostatu.

Při zážehu kotle se nejprve nahřívá zapalovací tyč a plnicí dopravník nasype do hořáku potřebnou dávku paliva, spustí se ventilátor. Tyč ohřívá vzduch a dojde ke zapálení paliva. Dojde ke zvýšení teploty v kotli a čidlo spalín vypne zapalovací tyč. Během provozu se pozvolna navyšuje plnění palivem až na hodnotu pro maximální výkon. Pokud je v hořáku dostatečné množství žnouceho zbytku paliva, regulátor nesepe zapalovací tyč.

V případě že snímač nezaznamená odpovídající nárůst teploty, kotel je odstaven a zobrazí hlášení „porucha nezapáleno“ [10].

4.2.2 Provoz

Při provozu regulátor upravuje otáčky ventilátoru a nastavuje časové intervaly pro dodáváním paliva do kotle pomocí plnicího dopravníku.

Intervaly roštování se nastavují pro jednotlivá paliva podle jejich vlastností. V nastavených intervalech dochází to ke spuštění popelavého dopravníku, který ústí do sběrného kontejneru [10].

4.2.3 Odstavení kotle

Regulátor kotel odstaví při odběru nižším, než je minimální výkon kotle nebo pokud je rozepnut pokojový termostat. Při doběhu je odstavená dodávka paliva, ventilátor je sepnut několik následujících minut a v hořáku dochází k vyhoření prchavých složek paliva. Po doběhu následuje dohořívací provoz, dojde k opakovanému roštování a následnému vyhoření paliva v hořáku, ventilátor ještě několik minut pracuje. Po vypnutí ventilátoru kotel přechází do režimu odstávky. Teplota vody v bojleru je nad nastavenou hodnotou a pokojový termostat je rozepnut.

Systémové čerpadlo zajišťuje rozvod vody v celém systému topení a kotlové čerpadlo slouží k distribuci ohřáté vody do bojleru [10].

Z režimu odstávka do plného provozu se kotel uvede automaticky za těchto podmínek:

- Po daném časovém intervalu (např. 1 hodina)
- Teplota vody v bojleru je o 20 °C nižší než nastavená teplota.
- Pokojový termostat je sepnut a teplota vody je 10 °C pod nastavenou hodnotou.

4.2.4 Nastavení kotlového čerpadla

Při sepnutém pokojovém termostatu a nepřipojeném čidle bojleru se čerpadlo spíná při teplotě vody vyšší, než je nastavená hodnota (obvykle 60 °C). Po snížení teploty o nastavenou hodnotu (obvykle 5 °C) se čerpadlo automaticky vypne.

Při rozepnutém pokojovém termostatu a připojeném čidle bojleru se čerpadlo automaticky spíná při teplotě vody v kotli větší než nastavená hodnota (obvykle 50 °C) a zároveň je teplota vody v kotli o 5 °C vyšší než na čidle bojleru. Po snížení teploty o nastavenou hodnotu (obvykle 5 °C) nebo pokud je teplota vody v kotli menší než na čidle bojleru, čerpadlo se automaticky vypne [10].

4.2.5 Ochranná zařízení

Z důvodu ochrany kotle před poškozením je při zvýšení teploty nad 93 °C je kotlové čerpadlo trvale zapnuto. K jeho vypnutí dojde až při poklesu na 90 °C.

K odstavení kotle dochází při teplotě vody vyšší než 95 °C, v případě překročení teploty vody nad 98 °C, havarijní termostat vypíná napájení ventilátoru, dopravníku a zapalovací tyče. Čerpadlo je nepřetržitě v provozu [10].

5 Výběr hardwaru

Tato kapitola obsahuje počáteční odhad hardwaru potřebného pro řízení kotle v reálném prostředí, avšak vzhledem k tomu, že řídicí systém nebylo možné implementovat do reálného provozu, zůstává kapitola pouze jako předběžný návrh pro potenciální budoucí realizaci.

Pro řízení kotle bude využito PLC Foxtrot CP-2000. Výběr hardwaru byl rozdělen do tří etap. V 1. etapě bude program vytvořen pro: dopravník paliva, lambda sondu pro měření kyslíku, PT-1000 pro měření teploty spalín a PT-1000 pro měření teploty kotle.

V 2. etapě bude připojená akumulární nádrž v návaznosti na to bude přidáno měření teploty vody před a za kotlem. Dále bude vytvořen program, který bude distribuovat vodu pomocí čerpadel po objektu a bude implementována ekvitermní regulace pro úsporu energie, k čemuž bude využit čtyřcestný ventil.

V poslední etapě se bude řešit bezdrátový způsob komunikace pro ovládání kotle pomocí systému od společnosti Teco.

5.1 PLC Foxtrot CP-2000

Jedná se o základní modul 2 generace s těmito vlastnostmi [9]:

- Dvě master sběrnice CIB s napájením a lze připojit až 64 modulů senzorů a aktorů
- TCL2 je masterem systémové rozšiřovací sběrnice
- Přes síť LAN (Ethernet) lze přidávat externí mastery do dalších větví CIB

Vstupy a výstupy:

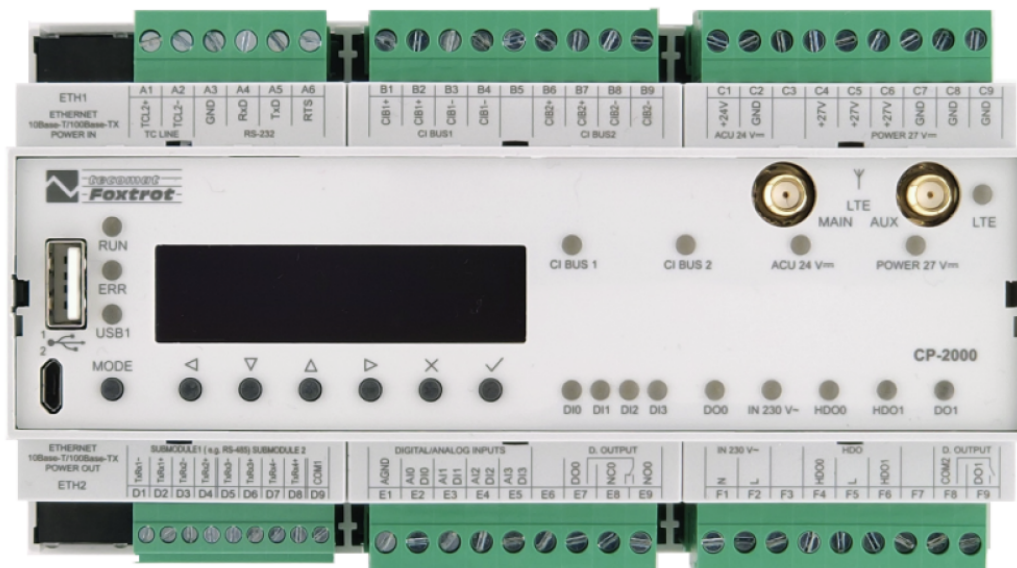
- Pro analogový nebo binární signál jsou využity 4 víceúčelové vstupy
- Vstup pro síťové napětí 230 V AC, který je monitorován
- Pro monitorování HDO slouží 2 binární vstupy na napěťové úrovni 230 V AC
- 2 reléové vstupy

Základní komunikační schopnosti modulu CP-2000:

- 2x Ethernet rozhraní s rychlostí přenosu 100 Mbit/s
- USB mikro pro programování
- USB A slouží pro Flash paměť nebo WIFI adaptér
- Zabudovaný LTE modem pro bezdrátové připojení do sítě
- 4 volitelná sériová rozhraní pro RS-485, RS-232

CP-2000 je osazen CPU řady L s pamětí:

- 1 MB pro uživatelský program
- 128 MB interního disku pro ukládání souborů s paralelním přístupem LAN k síťovému disku



Obr. 5.1: PLC Foxtrot CP-2000 [9]

- 16 MB rychlého RAM disku
- MicroSD karta až 1 TB
- Pro zabezpečení správy souborů slouží žurnálovací souborový systém
- Lokální komunikace s uživatelem přes OLED displej

5.2 PT-1000

Senzory Pt-1000 patří do skupiny teplotních senzorů nazývaných odporové teplotní detektory. K měření teploty využívají principu změny elektrického odporu s měnící se teplotou. Mezi další snímače, které tvoří tuto skupinu a které jsou možná známější, patří snímače teploty Pt-100.

První část názvu Pt-1000, "Pt", je chemická značka pro platinu a ukazuje, že snímač je založen na platině. Druhá část, 1000, udává jmenovitý odpor snímače při teplotě 0 °C. V tomto případě 1000 Ω.

Stejně jako u všech platinových odporových teploměrů je principem měření Pt-1000 změna odporu snímacího prvku s rostoucí nebo klesající teplotou. Tato hodnota odporu se měří pomocí Ohmova zákona a aplikací malého měřicího proudu. Nejčastěji používaná sonda Pt-1000 má základní interval 385 Ω pro nárůst teploty o 100 °C.

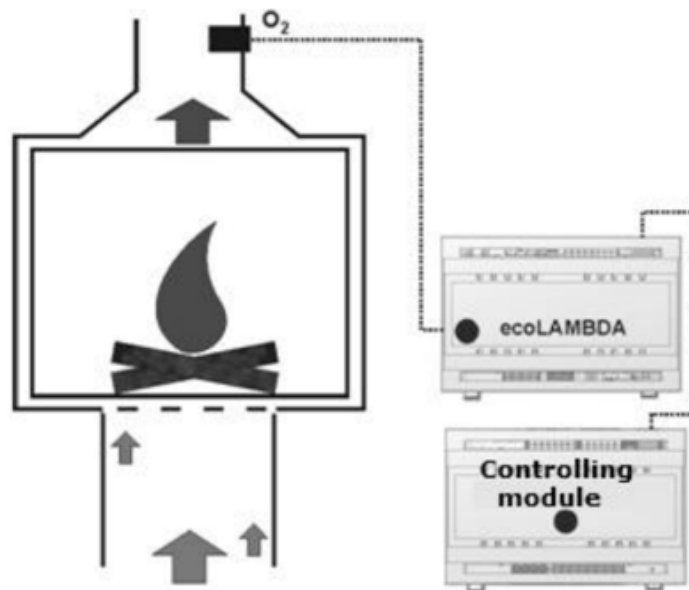
Jednou z hlavních vlastností odporového teplotního čidla Pt-1000 je relativně vysoký jmenovitý odpor a základní interval, což znamená, že odpor jakéhokoli prodlužovacího kabelu je ve srovnání s čidlem obecně zanedbatelný. To umožňuje použití

jednoduchého dvou vodičového měřicího obvodu téměř ve všech aplikacích. Teplotní sondy Pt-1000 s 5 metrovým kabelem by měly dodatečnou chybu menší než 0,1 °C. U Pt-100 by to bylo desetkrát horší, možná o 1 °C v závislosti na použitém kabelu [7].

5.3 EcoLambda ver. 2

Lambda Sonda je vhodná pro měření koncentrace kyslíku ve všech neagresivních prostředích až do teploty okolí 300 °C, tedy například pro regulaci primárního/ sekundárního vzduchu/ventilátoru spalin v kotlích. Při měření ve spalinách musí být velmi nízký obsah síry. Měření se přenášejí do řídicí jednotky prostřednictvím datové sběrnice.

EcoLAMBDA ver. 2 se používá k měření obsahu kyslíku ve výfukových plynech. Výsledky měření jsou odesílány prostřednictvím přenosu RS485 do řídicích jednotek spolupracujících s modulem, jejichž úkolem je udržovat obsah kyslíku a teplotu výfukových plynů na nastavené úrovni [5].



Obr. 5.2: Princip zapojení Lambda sondy [5]

6 PID regulátory

PID regulátor (Proporcionálně-Integračně-Derivační regulátor) je průmyslově nejběžnější řídicí algoritmus. Využívá tři základních složek pro přesnou regulaci procesu, a to proporcionální (P), integrační (I) a derivační (D). Tyto složky umožňují regulátoru poskytovat rychlou a stabilní odezvu na změny v systému [6].

6.1 Princip funkce

Proporcionální složka (P): Tato složka reaguje na aktuální chybu regulace, která je definována jako rozdíl mezi požadovanou a skutečnou hodnotou. Tato chyba je násobena konstantou K_p (proporcionální zesílení), čímž se získá proporcionální akce. P složka tedy poskytuje okamžitou odezvu, ale sama o sobě nemůže úplně eliminovat chybu [4].

Integrační složka (I): Sleduje kumulativní chybu v čase, což umožňuje postupně eliminovat rozdíl mezi požadovanou a skutečnou hodnotou. Tato složka zvyšuje účinek regulátoru při dlouhodobých odchylkách a díky ní lze dosáhnout nulové chyby. Nastavení této složky závisí na integrační konstantě K_i [6].

Derivační složka (D): Funguje jako předvídavá akce, protože reaguje na rychlost změny chyby a koriguje budoucí vývoj. Tato složka je obzvláště účinná při zamezení překmitům a oscilacím, protože tlumí náhlé změny. Nastavuje se pomocí konstanty K_d [4].

6.2 Nastavení a ladění PID regulátoru

Nastavení PID regulátoru je klíčové pro dosažení stabilní a přesné regulace. Při ladění se obvykle upravují konstanty K_p , K_i a K_d metodou pokus-omyl nebo na základě heuristických pravidel. Jedním z oblíbených přístupů je metoda Ziegler-Nichols, která poskytuje orientační hodnoty konstant [4].

6.3 Účinky jednotlivých složek na systém

Proporcionální akce: Zesílení K_p přímo ovlivňuje, jak rychle regulátor reaguje na změny, ale také může zvýšit velikost překmitů nebo způsobit nestabilitu. Integrační akce: Konstanty K_i je třeba nastavit pečlivě, aby nedošlo k příliš velkým překmitům nebo oscilacím. Derivační akce: Konstanty K_d pomáhají stabilizovat regulátor, ale pokud jsou příliš vysoké, mohou způsobit nežádoucí zpomalení [6].

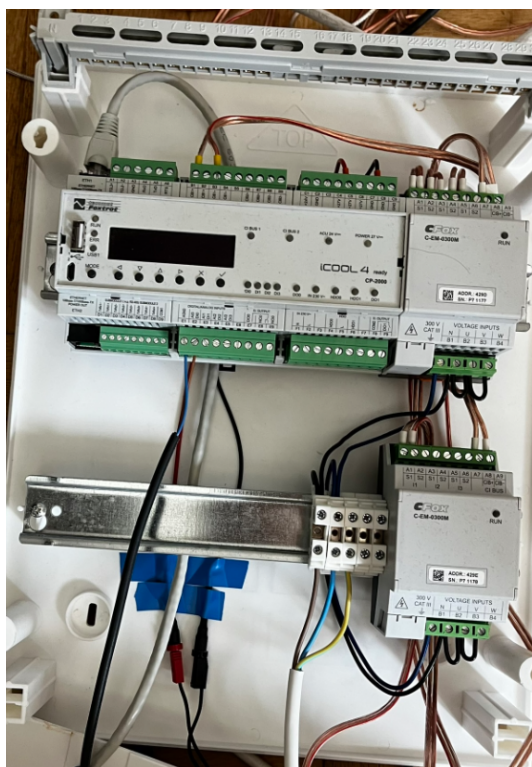
6.4 Příklady praktického použití

PID regulátory se široce používají v průmyslu pro regulaci teploty, tlaku, rychlosti a dalších parametrů. Jsou také využívány ve strojích a procesech, kde je potřeba přesně řídit požadovanou hodnotu a minimalizovat odchylky.

Kombinace proporcionální, integrační a derivační akce v PID regulátorech umožňuje udržet řízenou veličinu na požadované hodnotě s minimální chybou a překmity [4].

7 Měření a analýza dat

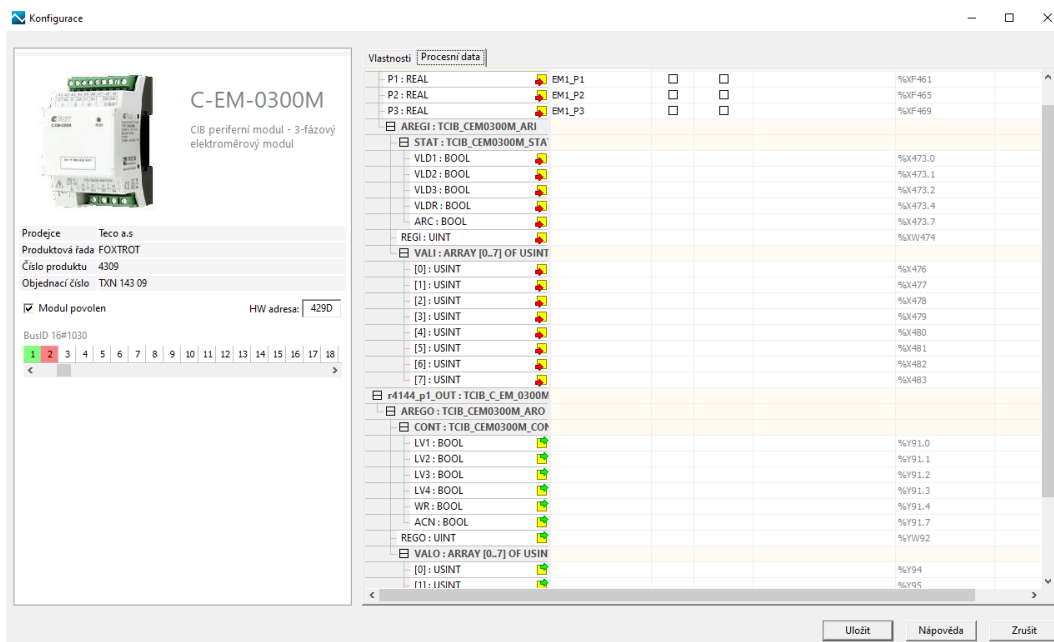
Pro vytvoření řídicího softwaru kotle bylo zapotřebí získání dat ze stávající řídicí jednotky kotle. Měření bylo provedeno pomocí proudových transformátorů. Jednotlivé proudové transformátory měřily proud na motoru pro dopravník paliva, motoru pro šnekový odpopelovač, ventilátoru a dále celkové proudové zatížení odběru kotle.



Obr. 7.1: Rozvaděč pro měření příkonu kotle

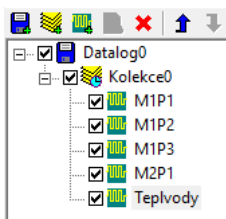
Proudové transformátory byly připojeny ke dvěma elektroměrům CFox C-EM-0300M, které byli napájeny nominálním napětím 230 V. Elektroměry, jenž byly připojeny na stejných potenciálových svorkách jako proudové transformátory, měřily velikost potenciálového fázového napětí. Následně elektroměr přepočítával fázové napětí a proud na příkon pro jednotlivé měřené části. Nakonec elektroměry posílaly data přes sběrnici CIB (Common Installation Bus) do programovatelného logického automatu (PLC). Tam byla data ukládána na externí flashdisk. Na obrázku 7.1 lze vidět zapojení dvou elektroměrů a PLC, kterými bylo měření prováděno.

Pro tyto měření byl v prostředí Mosaic vytvořen software, díky němuž bylo možné přepočítávat hodnoty fázového napětí a proudu na příkon kotle v reálném čase a ukládat data na flashdisk připojený do PLC. Komunikace mezi PLC Fox Trot CP2000 a elektroměry probíhala přes CIB (Common Installation Bus) sběrnice.



Obr. 7.2: Přiřazení modulů CFOX C-EM-0300M v prostředí Mosaic

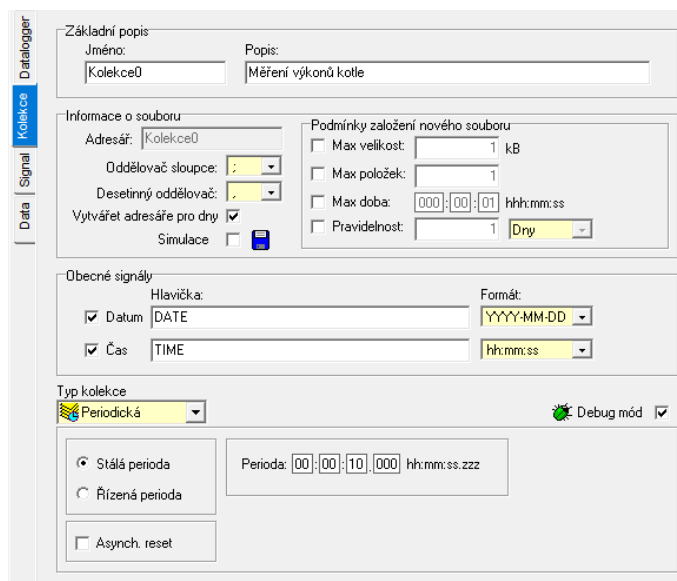
Na obrázku 7.2 je přiřazen modulu C-EM-0300M identifikátor BusID sběrnice, který umožňuje komunikaci s PLC. Dále byly definovány vstupy tohoto modulu jako proměnné typu REAL s názvy EM1_P1 (motor dopravníku), EM1_P2 (motor odpopelnění), EM1_P3 (ventilátor) a EM2_P1 (celkový příkon kotle).



Obr. 7.3: Struktura pro Datalog0

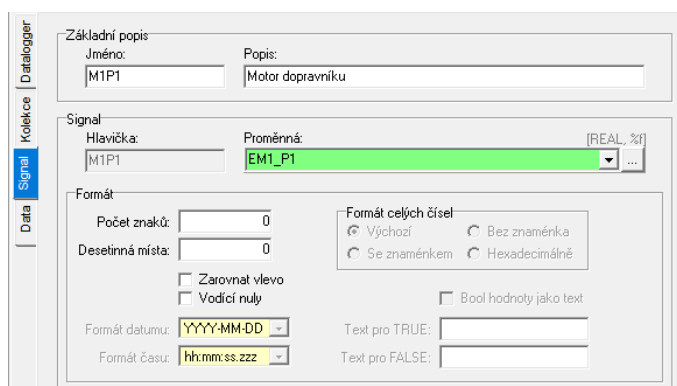
Poté byl vytvořen Datalog0 pro účely sběru dat. Do tohoto datového logu, viz obrázek 7.3, byla přidána kolekce, která obsahuje signály EM1_P1, EM1_P2, EM1_P3, EM2_P1 a také proměnnou pro měření teploty vody.

V kolekci, viz obrázek 7.4, pro Datalog0 byly nastaveny základní informace, stejně jako specifikace souboru pro ukládání dat. Byly definovány parametry, jako je oddělovač sloupců, způsob oddělení desetinných míst a časový interval, po kterém se má vytvořit nový soubor. Každé měření zaznamenává datum a čas. Interval sběru dat byl nastaven na 10 sekund, takže měření probíhá periodicky každých 10 sekund. Tímto způsobem se všechny vybrané proměnné ukládají do souboru na flashdisk.



Obr. 7.4: Okno kolekce

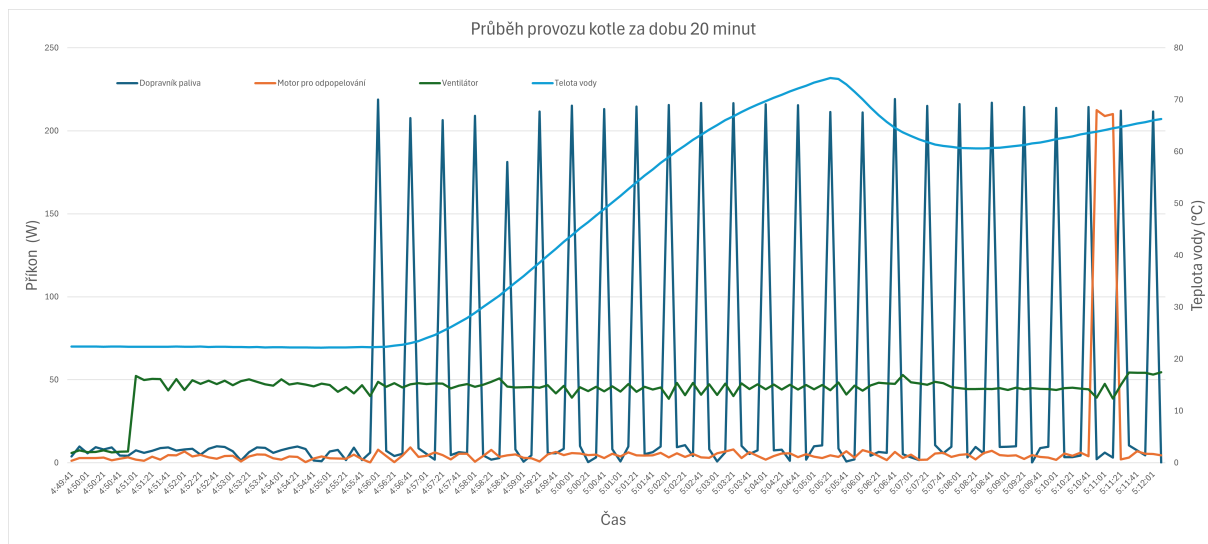
V okně signálu, viz obrázek 7.5, byla nastavená volaná proměnná a její jméno a popis.



Obr. 7.5: Okno signál

7.1 Výsledky měření

Aby bylo možné naměřená data verifikovat byly vytvořeny grafické závislosti po různé intervaly a následně byla provedená analýza jednotlivých činností kotle.

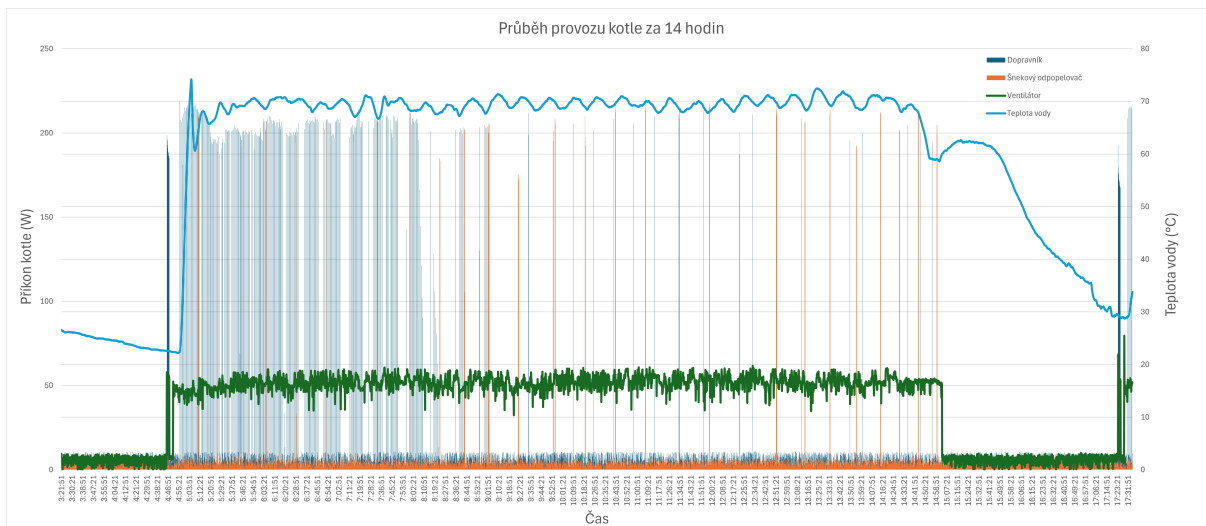


Obr. 7.6: Průběh provozu kotle za dobu 20 minut

V grafickém znázornění, viz obrázek 7.6, je vidět průběh provozu kotle ve dvaceti minutovém intervalu, ve kterém kotel sepnul pro dosažení teploty požadované termostatem. Průběh ukazuje proces po ukončení zapalovacího procesu a přechodu do provozního procesu. Před zobrazeným průběhem byla do kotle dodána zapalovací dávka paliva a sepnuta zapalovací tyč. Tento průběh začíná sepnutím ventilátoru (zelená křivka), který má pět stupňů rychlostí regulace.

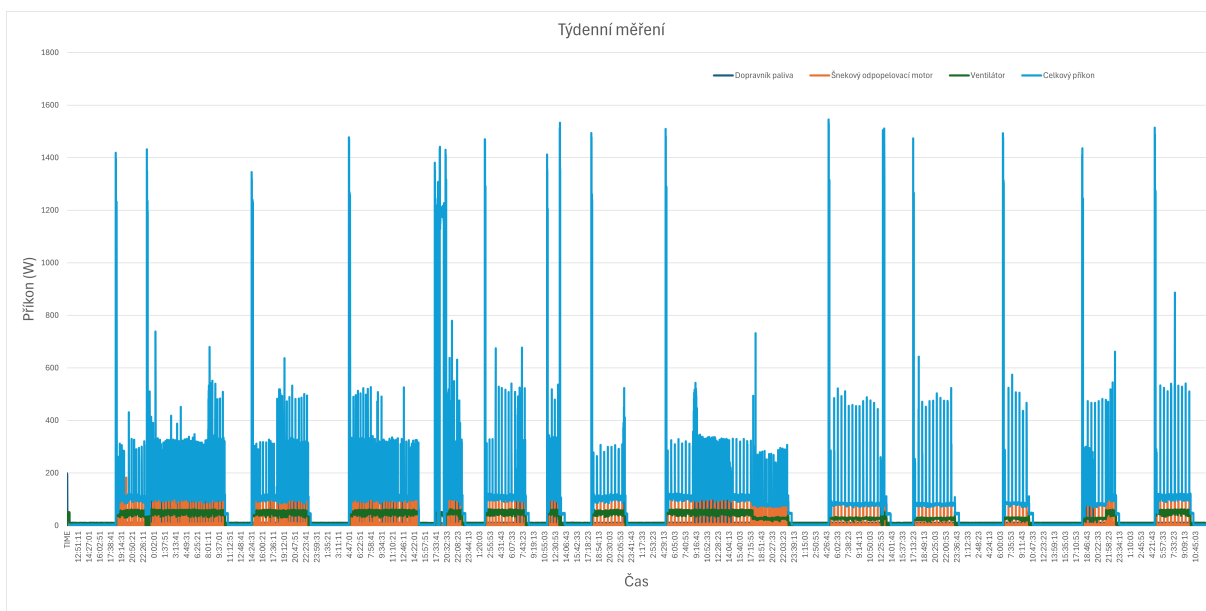
Po rozpálení zapalovací dávky, které trvalo 5 minut, se začne po 10vteřinových intervalech spínat dopravník paliva (tmavě modrá křivka). Světle modrá křivka znázorňuje teplotu vody na výstupu z kotle. Požadovaná teplota byla nastavena na 70 °C a je řízena pomocí regulátoru, který při překročení teploty vody sníží otáčky ventilátoru a množství paliva přivedeného do kotle pro doregulování na požadovanou teplotu.

Motor pro odpopelování (oranžová křivka) je sepnut v první fázi po rozběhu kotle po 15 minutách a následné intervaly sepnutí jsou po 20minutové pauze. Doba odpopelovacího procesu je nastavena na 40 vteřin.



Obr. 7.7: Průběh provozu kotle za dobu 14 hodin

V grafickém znázornění, viz obrázek 7.7, je zobrazen 14hodinový průběh kotle při běžném provozu. Teplota vody (světle modrá křivka) stoupla na požadovanou teplotu 70 °C a následně je udržována pomocí regulace. Na začátku byla voda v potrubí po objektu chladná, takže kotel musel mít větší výhřevnost, aby vyrovnal rozdíl teplot, a na výstupu dosáhl požadované teploty vody. Dopravník paliva (tmavě modrá křivka) tedy v prvních 3 hodinách dodával větší množství paliva a následně, při menším rozdílu teploty vstupní a výstupní, snižoval množství paliva.



Obr. 7.8: Průběh provozu kotle za celý týden

Na obrázku 7.8 jsou graficky znázorněny průběhy z celého týdenního měření. V době měření se venkovní teplota pohybovala na začátku týdne v rozmezí od 2 do 13 °C a od druhé poloviny týdne od 6,5 do 15 °C. V grafickém znázornění lze vidět, že v závislosti na venkovní teplotě byly v první půlce týdne intervaly sepnutého kotle delší než v druhé. Světle modrá křivka vykresluje celkový příkon kotle. Je to součet příkonu všech sepnutých prvků v kotli. Hodnoty dosahující až k 1400 Wattům jsou způsobeny sepnutím zapalování tyče na začátku procesu zapalování, která má odběr 1000 W.

8 Program

Program byl vytvořen pro řízení kotle Verner A251, který nakonec nebyl dostupný pro otestování správné funkce tohoto programu. Namísto odzkoušení programu v reálném provozu se program rozšířil o simulaci kotle, která simuluje proces rozběhu a provozu kotle.

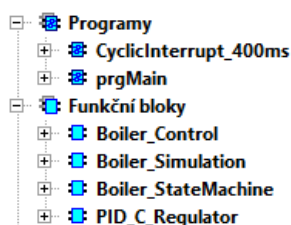
8.1 Programovací prostředí Mosaic

Program byl psán ve vývojovém prostředí Mosaic. Je to vývojový nástroj, který je využíván k tvorbě běžných i náročnějších aplikací systému Tecomat. Využívá se pro programování všech PLC (Programovatelný logický automat) jednotek od společnosti Teco. Jeho vlastnosti jsou tvorba programu a odladění, velké projekty mají mnoho řídicích systémů nebo vzdálené I/O moduly, pro které Mosaic využívá řadu moderních technologií. Prostředí i nástroje splňují normu IEC 61131-3 (Souhrn požadavků na moderní řídicí systémy) [11].

Programy je možné psát v textových jazycích, kterými jsou ST (Strukturovaný text) a IL (Instrukční list) nebo v grafických jazycích LD (Reléová schémata) a FBD (Funkční bloky). V případě této práce je program zpracováván v textovém jazyce ST (strukturovaný text), který je určen pro složité výpočty, algoritmy a jiné, kdy by byl zápis při použití instrukcí velmi složitý [3].

8.2 Struktura

Struktura programu, viz obrázek 8.1, je vytvořena následovně, programy `CyclicInterrupt_400ms` a `prgMain` slouží k řízení celého řídicího procesu kotle. Tyto Programy volají z funkčních bloků podprogramy pro správné řízení kotle. Funkční bloky byly využity pro lepší přehlednost a každý blok má svoji specifickou funkci, které jsou volány do Programů. Tato struktura obsahuje následující funkční bloky: `Boiler_Control`, `Boiler_Simulation`, `Boiler_StateMachine`, `PID_C_Regulator`.



Obr. 8.1: Struktura programu

8.2.1 Hlavní cyklus PLC prgMain

Hlavní cyklus PLC `prgMain` je nekonečný proces, který se stará o volání potřebných funkčních bloků k monitorování a ovládání funkcí kotle, viz obrázek 8.2. Cyklus `prgMain` je zpracováván v rámci každého cyklu PLC. V tomto případě trvá jeden cyklus PLC přibližně ± 100 ms.

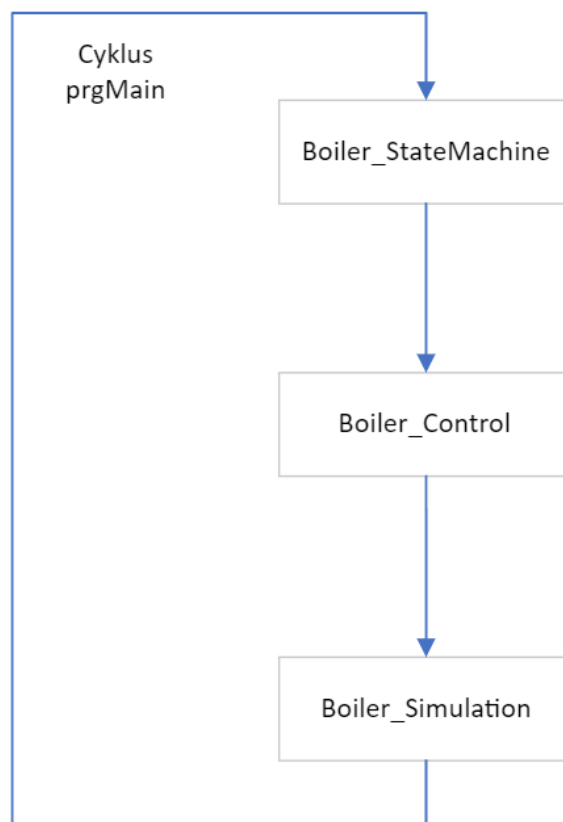
V první části programu je zjišťován stav režimu provozu kotle, zda je aktivní automatický nebo manuální režim provozu. V případě změny režimu se vygeneruje krátký impuls (přibližně 250 ms), který pomáhá v pozdějších částech programu pro odchyťování eventů při změně režimu z manuálního do automatického a naopak.

Následně je volán funkční blok `Boiler_StateMachine` s názvem instance `Kotel`. Tento funkční blok obsahuje stavový automat pro řízení kotle a interně zajišťuje ovládání jednotlivých kroků jako je zapálení, chod a odstávka.

Funkční blok `Boiler_Control` řídí různé aspekty provozu kotle, jako je zapalování, regulace dopravníku a ventilátoru spalin. Instance tohoto bloku s názvem `BoilerRegulation` primárně ovládá výstupní periferie, které jsou nezbytné pro správný chod kotle, včetně času a stavu dopravníku a zásobníku popela.

Funkční blok `Boiler_Simulation` simuluje chod kotle na základě poskytnutých vstupů a podmínek. Instance tohoto bloku, nazvaná `Boiler_Simulation`, umožňuje alespoň částečnou simulaci provozu kotle. Vzhledem k nedostupnosti reálného kotle pro testování funkčnosti programu je tato simulace klíčová pro ověření správnosti a účinnosti řešení.

Na konci zpracování hlavního cyklu se vyresetuje proměnná `Alarm_Reset`, a to z důvodu, kdyby se v HMI (Human-Machine Interface) nastavil stav této proměnné do logické jedničky (stav `True`) a nebyl by změněn zpět do logické nuly (stav `False`), zůstala by stále aktivní a resetovala by poruchové stavy do nekonečna. Touto jednoduchou logikou je zajištěno, že proměnná bude ve stavu logické jedničky pouze po dobu vykonání jednoho cyklu PLC.



Obr. 8.2: Průběh jednoho cyklu prgMain programu

8.2.2 CyclicInterrupt_400ms

Tento program, nazvaný `CyclicInterrupt_400ms`, je cyklicky volaný (přerušovaný) programový blok, který je procesorem volán každých 400 ms. Cyklický přerušovaný blok byl použit právě pro implementaci PID regulátorů a regulaci požadovaných procesů. Použití přerušovaného programového bloku je žádoucí pro přesnou a optimalizovanou činnost PID regulátoru, který pracuje v diskrétním čase, a tudíž použití v hlavním programovém bloku (`prgMain`) není žádoucí vzhledem k možnosti kolísání rychlosti zpracování jednoho cyklu PLC.

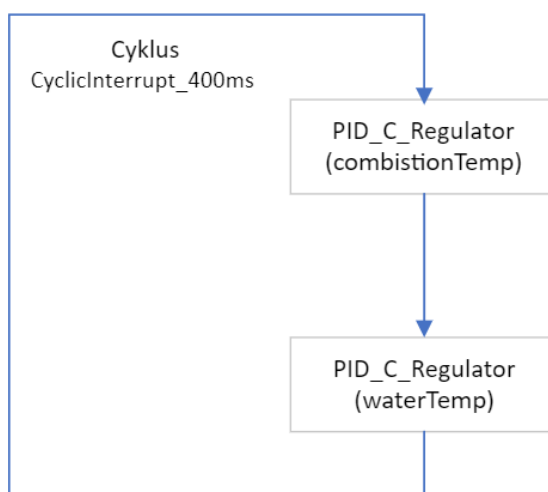
Programový blok obsahuje dva PID regulátory: `CombustionTempRegulator` pro regulaci teploty spalin a `WaterTempRegulator` pro regulaci teploty vody, viz obrázek 8.3. Každý PID regulátor má nastaveny odpovídající regulační konstanty, které odpovídají chování regulační soustavy.

Funkční blok `PID_C_Regulator` implementuje PID algoritmus pro řízení procesů na základě aktuálních hodnot, požadovaných hodnot (setpointů) a nastavených konstant. Má vstupy pro aktuální hodnotu procesu, požadovanou hodnotu (setpoint), režim regulace (automatický/manuální), manuálně nastavenou akční veličinu a sig-

nál pro restart regulátoru. Poskytuje výstupy obsahující vypočítanou akční veličinu a statusní slovo indikující stav regulace.

Implementuje PID algoritmus v automatickém i manuálním režimu. V automatickém režimu vypočítává akční veličinu na základě rozdílu mezi aktuální a požadovanou hodnotou a integruje minulé chyby pro eliminaci statické odchylky. V manuálním režimu pouze předává hodnotu manuálně nastavené akční veličiny.

Zajišťuje omezení akční veličiny na předem stanovené limity a ukládá aktuální hodnoty pro další výpočty v následujícím cyklu. Díky PID regulátorům je možné dosáhnout stabilního a přesného řízení teploty spalin a vody v kotli, což přispívá k efektivnímu a bezpečnému provozu tepelného systému.



Obr. 8.3: Průběh jednoho cyklu CyclicInterrupt_400ms programu

8.3 Definování proměnných

Definování globálních proměnných je klíčovým prvkem v prostředí Masaic, který umožňuje uživatelům efektivně spravovat informace a stav zařízení v jejich aplikacích. Globální proměnné jsou proměnné, které mají širší rozsah platnosti a jsou dostupné ve všech částech programu, bez ohledu na to, kde byly původně deklarovány. Tento koncept poskytuje jednoduchý způsob sdílení informací mezi různými částmi programu a umožňuje koordinaci jejich chování.

Ve struktuře definice globálních proměnných jsou zahrnuty různé datové typy, které usnadňují organizaci a manipulaci s daty v programu. Tyto datové typy jsou navrženy tak, aby poskytovaly uživatelům lepší orientaci a usnadnily práci při vývoji programu. Díky nim lze data jednoduše kategorizovat a spravovat, což vede k vyšší přehlednosti a efektivitě při programování.

V tomto programu jsou definovány různé typy dat, jako jsou výčty pro režimy doplňovacího dopravníku a ventilátoru spalování, stejně jako struktury pro nastavení teplot, času a dalších provozních parametrů kotle.

V první části, viz obrázek 8.4, byly definovány výčtové typy pro různé režimy a stavy kotle. Každý výčtový typ definuje konkrétní režimy kotle, které mohou být přiřazeny proměnným tohoto typu.

První definice, `Bolier_FillingConvMode_Enum`, obsahuje režimy plnicího dopravníku kotle: `OFF`, `ON_IGNITION` (zapnuto během zapalování) a `ON_REGULATION` (zapnuto v regulačním režimu).

Druhá definice, `Bolier_CombFanMode_Enum`, popisuje režimy ventilátoru spalování kotle: `OFF` a `ON_REGULATION` (zapnuto v regulačním režimu).

Třetí definice, `Boiler_ActualStatus_Enum`, identifikuje různé stavy aktivity kotle, včetně `WAITING` (čekání), `IGNITION` (zapalování), `HEATUP` (přehřívání), `RUNNING` (běžící), `STOPPING` (zastavování), `DOWNTIME` (mimo provoz) a `ERROR` (chyba).

```
TYPE //Definice pro různé režimy a stavy kotle
Bolier_FillingConvMode_Enum : (OFF, ON_IGNITION, ON_REGULATION);
Bolier_CombFanMode_Enum : (OFF, ON_REGULATION);
Boiler_ActualStatus_Enum : (WAITING, IGNITION, HEATUP, RUNNING, STOPPING, DOWNTIME, ERROR);
```

Obr. 8.4: Režimy a stavy kotle

Na obrázku 8.5 je kód obsahující definice struktur pro nastavení teplotních a časových parametrů kotle.

První struktura, `BoilerDb_Stp_Temps_Type`, slouží k definici teplotních parametrů a obsahuje několik prvků: `IgnitionCombTemp`, `MinRunningCombTemp`, `CombustionTemp`, `WaterTemp` a `MaxWaterTemp`, které určují teplotu spalin při rozběhu kotle, minimální provozní teplotu spalin, teplotu spalin při provozu a teplotu vody v kotli.

Druhá struktura, `BoilerDb_Stp_Time_Type`, definuje časové parametry a zahrnuje několik prvků, jako jsou `IngitionDose`, `CombFanDelay`, `MaxIngition`, `FillingDoseMaxPause`, `FillingDoseInterval`, `AshTrayPause` a `AshTrayInterval`.

Nakonec struktura `BolierDb_Setpoints_Type` spojuje teplotní a časové parametry dohromady a zahrnuje také proměnnou `ActualHeatSource`, která označuje aktuální zdroj tepla.

```

// Boiler TYPE - Setpoints

BoilerDb_Stp_Temps_Type : STRUCT //Struktura pro nastavení teplotních parametrů
  IgnitionCombTemp : REAL := 30; //Teplota spalin při rozběhu kotle
  MinRunningCombTemp : REAL := 50;
  CombustionTemp: real := 200; //Teplota spalin při provozu
  WaterTemp : real := 60; //Teplota vody
  MaxWaterTemp : REAL := 70; //Maximální teplota vody
END_STRUCT

BoilerDb_Stp_Time_Type : STRUCT //Struktura pro nastavení časových parametrů
  IngitionDose : TIME := t#30s; //Zapalovací dávka paliva
  CombFanDelay : TIME := t#45s; //Zpoždění ventilátoru
  CombFanStopDelay : TIME := t#20s; //Zastavení ventilátoru se zpožděním

  MaxIngition : TIME := t#60s; //Maximální doba zapalování

  FillingDoseMaxPause : TIME := t#10s; //Pauze dopravníku paliva
  FillingDoseInterval : TIME := t#1s; //Doba chodu dopravníku

  AshTrayPause : TIME := t#60s; //Pauza mezi roštováním
  AshTrayInterval : TIME := t#10s; //Doba roštování
END_STRUCT

BoilerDb_Setpoints_Type : STRUCT
  Temps: BoilerDb_Stp_Temps_Type;
  Times: BoilerDb_Stp_Time_Type;
  ActualHeatSource : INT := 1;
END_STRUCT

```

Obr. 8.5: Proměnné pro nastavení teplotních a časových parametrů

Tento kód na obrázku 8.6 obsahuje definice struktur pro sledování aktuálních hodnot a stavu kotle. Struktura `BoilerDb_Act_Temps_Type` obsahuje prvky `Water` a `Combustion`, které uchovávají aktuální teplotu vody a spalin v kotli.

Další struktury, `BoilerDb_Act_Conveyor_Type` a `BoilerDb_Act_Fan_Type`, obsahují informace o aktuálním režimu a výstupy regulace dopravníku paliva a ventilátoru spalin. Každá z těchto struktur má prvky pro režim (např. `FillingConveyorMode` a `CombustionFanMode`) a pro hodnotu regulace (např. `PID_Out_WaterReg` a `PID_Out_CombustionReg`).

Struktura `BoilerDb_Actual_Type` spojuje informace o aktuální teplotě a stavu kotle dohromady. Obsahuje proměnné typu `BoilerDb_Act_Temps_Type` pro teploty vody a spalin a proměnnou `ActualBoilerStatus` typu `Boiler_ActualStatus_Enum` pro sledování aktuálního stavu kotle.

Nakonec, struktura `BoilerDb_ControlVar_Type` uchovává informace o řízení kotle. Obsahuje informace o režimu dopravníku paliva a ventilátoru spalin (`FillingConv` a `CombustionFan`) a proměnnou `IgnitionRodEnable`, která indikuje, zda je možné povolit zapalovací tyč.

Celkově, struktura `BoilerDb_Type` spojuje všechny tyto informace dohromady, zahrnující nastavení, aktuální hodnoty a řídicí proměnné kotle.

```

// Boiler TYPE - Actual
BoilerDb_Act_Temps_Type : STRUCT //Aktuální teplota
  Water : REAL; //Voda
  Combustion : REAL; //Spaliny
END_STRUCT

BoilerDb_Act_Conveyor_Type : STRUCT //Aktuální hodnoty regulaci
  FillingConveyorMode : Bolier_FillingConvMode_Enum; //Mod dopravníku
  PID_Out_WaterReg : REAL; //Hodnota pro regulaci vody
END_STRUCT

BoilerDb_Act_Fan_Type : STRUCT //Aktuální hodnoty regulaci
  CombustionFanMode : Bolier_CombFanMode_Enum; //Mod ventilátoru
  PID_Out_CombustionReg : REAL; //Hodnota pro regulaci spalin
END_STRUCT

BoilerDb_Actual_Type : STRUCT
  Temps: BoilerDb_Act_Temps_Type;
  ActualBoilerStatus: Boiler_ActualStatus_Enum;
END_STRUCT

BoilerDb_ControlVar_Type : STRUCT
  FillingConv: BoilerDb_Act_Conveyor_Type;
  CombustionFan: BoilerDb_Act_Fan_Type;
  IgnitionRodEnable : bool;
END_STRUCT

BoilerDb_Type : STRUCT
  Setpoints : BoilerDb_Setpoints_Type;
  Actual : BoilerDb_Actual_Type;
  ControlVars : BoilerDb_ControlVar_Type;
END_STRUCT

```

Obr. 8.6: Proměnné pro sledování aktuálních hodnot

Tato struktura, viz obrázek 8.7, je pro manuální režim kotle, který umožňuje uživateli přímé ovládání různých prvků kotle. Každý prvek má svůj vlastní parametr, který určuje jeho stav (zapnutý/vypnutý) nebo nastavení (například výkon ventilátoru).

```

// Boiler TYPE - manual mode
BoilerDb_Manual_Type : struct
  IgnitionRodEnable : bool; //Zapalovací tyč On/Off

  CombustionFanEnable : bool; //Ventilátor On/Off
  CombustionFanOP : real; //Výkon ventilátoru

  FillingConveyorEnable : bool; //Dopravník On/Off
  FillingConveyorOP : real; //Výkon dopravníku

  AshTrayEnable : bool; //Roštování On/Off

  WaterPumpEnable : bool; //Čerpadlo vody On/Off
end_struct

```

Obr. 8.7: Manuální mód kotle

Tento kód na obrázku 8.8 obsahuje definici dvou struktur proměnných pro simulaci kotle. Struktury umožňují simulovat vstupy a výstupy kotle pro účely testování a ladění kódu.

První struktura, `BoiierSimulation_Inputs_Type`, obsahuje proměnné, které simulují vstupy do kotle, jako je požadovaná teplota vody (`WaterTemp`), požadovaná teplota spalin při provozu (`CombustionTemp`) a stav termostatu (`Termostat`), který indikuje, zda je termostat zapnutý nebo vypnutý.

Druhá struktura, `BoilerSimulation_Outputs_Type`, obsahuje proměnné, které simulují výstupy z kotle, jako je stav zapalovací tyče (`IgnitionRod`), stav ventilátoru spalín (`CombustionFan`), aktuální výkon ventilátoru spalín (`CombustionFanOP`), stav dopravníku paliva (`FillingConveyor`), aktuální výkon dopravníku paliva (`FillingConveyorOP`), stav roštování (`AshTray`) a stav čerpadla vody (`WaterPump`).

```
// Boiler TYPE - simulation inputs
BoilerSimulation_Inputs_Type : struct //Struktura proměnných pro simulaci
  WaterTemp : real; //Nastavení požadované teploty vody
  CombustionTemp : real; //Nastavení požadované teploty spalín při provozu

  Termostat : bool; //Termostat On/Off
end_struct

// Boiler TYPE - simulation outputs
BoilerSimulation_Outputs_Type : struct //Struktura proměnných pro simulaci
  IgnitionRod : bool; //Zapalovací tyč On/Off

  CombustionFan : bool; //Ventilátor On/Off
  CombustionFanOP : real; //Výkon ventilátoru

  FillingConveyor : bool; //Dopravník On/Off
  FillingConveyorOP : real; //Výkon dopravníku

  AshTray : bool; //Roštování On/Off

  WaterPump : bool; //Čerpadlo vody On/Off
end_struct
```

Obr. 8.8: Výstupy a vstupy proměnných pro simulace

Nakonec byly přidány definice globálních proměnných pro sledování a řízení stavu kotle, viz obrázek 8.9. Mezi tyto proměnné patří `Mode_AutomaticManual`, která určuje, zda je kotel v režimu automatického nebo manuálního ovládání. Dále je zde proměnná `DbBoilerManualMode`, která uchovává informace o manuálním ovládání kotle.

Pro simulaci vstupů a výstupů kotle jsou definovány proměnné `BoilerSimulationInputs` a `BoilerSimulationOutputs` odpovídajícím způsobem. `Mode_Change` indikuje změnu režimu kotle, zatímco `Alarm_Reset` uchovává informaci o resetování alarmu.

Proměnné označené jako `VAR_GLOBAL RETAIN` jsou udržovány i po vypnutí kotle. Jedná se o proměnnou `DbBoiler`, která je typu `BolierDb_Type` a obsahuje data o stavu kotle, včetně nastavení a aktuálních hodnot. Tyto proměnné jsou klíčové pro sledování a řízení kotle a umožňují efektivní simulaci a správné fungování zařízení.


```

VAR_GLOBAL //Globální proměnný

Mode_AutomaticManual : BOOL; //Proměnná pro manuální/automatický mod

DbBoilerManualMode : BoilerDb_Manual_Type; //Definuje proměnnou pro daný typ struktury

BoilerSimulationInputs : BoilerSimulation_Inputs_Type; //Definuje proměnnou pro daný typ struktury
BoilerSimulationOutputs : BoilerSimulation_Outputs_Type; //Definuje proměnnou pro daný typ struktury
Mode_Change : BOOL; //Změna režimu kotle
Alarm_Reset : BOOL; //Uchování informace o resetování

END_VAR

VAR_GLOBAL RETAIN //Proměnné zachovány i po vypnutí kotle
DbBoiler: BoilerDb_Type; //Data o stavu kotle včetně nastavení a aktuálních hodnot
END_VAR

```

Obr. 8.9: Globální proměnné

8.4 Popis kódu

8.4.1 Program prgMain

Na obrázku 8.10 je vidět program `prgMain`, který má za úkol řídit a monitorovat provoz kotle. Začíná se sledem událostí generovaných impulzními generátory `OnAutoModeChangePulse` a `OnManModeChangePulse`, které mají periodu 250 ms. Tyto generátory vytvářejí signály pro automatickou a manuální změnu režimu provozu kotle. Pokud alespoň jeden z těchto signálů vyšel (`OnAutoModeChangePulse.Q` nebo `OnManModeChangePulse.Q`), proměnná `Mode_Change` je nastavena na `true`, což signalizuje změnu režimu.

Následuje aktualizace teplot spalin a vody na hodnoty simulovaných teplot. Funkční blok `Boiler` (`Boiler_StateMachine`) je poté spuštěn s příslušnými vstupy a výstupy. Do bloku jsou předány informace o režimu provozu, případné změně režimu, stavu termostatu a stavu alarmu. Po spuštění bloku `BoilerRegulation` (`Boiler_Control`) následuje blok `BoilerSim` (`Boiler_Simulation`), který simuluje chod kotle.

Na závěr se kontroluje stav proměnné `Alarm_Reset`. Pokud je aktivní, což indikuje přítomnost alarmu, proměnná `Alarm_Reset` je resetována na `false`. Tím je zajištěno, že v případě, že by se alarm aktivoval a nebyl řádně resetován, nedojde k nekonečnému opakování alarmového stavu. Celkově tato část programu zajišťuje správné řízení a monitorování chodu kotle v různých režimech provozu.

```

// Mode change
OnAutoModeChangePulse(IN := Mode_AutomaticManual, PT := t#250ms);
OnManModeChangePulse(IN := not Mode_AutomaticManual, PT := t#250ms);

IF OnAutoModeChangePulse.Q OR OnManModeChangePulse.Q THEN
  Mode_Change := true;
ELSE
  Mode_Change := false;
END_IF;

DbBoiler.Actual.Temps.Combustion := BoilerSimulationInputs.CombustionTemp;
DbBoiler.Actual.Temps.Water := BoilerSimulationInputs.WaterTemp;

Boiler(
  IN_Automat := Mode_AutomaticManual
  ,IN_Manual := not Mode_AutomaticManual
  ,IN_ModeChange := Mode_Change
  ,IN_Termostat := BoilerSimulationInputs.Termostat
  ,IN_AlarmReset := Alarm_Reset
  ,OUT_ActualStatus => DbBoiler.Actual.ActualBoilerStatus
  ,OUT_IgnitionRod => DbBoiler.ControlVars.IgnitionRodEnable
  ,OUT_CombustionFanMode => DbBoiler.ControlVars.CombustionFan.CombustionFanMode
  ,OUT_FillingConveyorMode => DbBoiler.ControlVars.FillingConv.FillingConveyorMode
);

BoilerRegulation();

BoilerSim();

IF Alarm_Reset THEN
  Alarm_Reset := false;
END_IF
END_PROGRAM

```

Obr. 8.10: Hlavní program prgMain

8.4.2 Funkční blok Boiler_StateMachine

Kód představuje funkční blok nazvaný Boiler_StateMachine, který slouží k řízení provozu kotle pomocí stavového automatu. Blok je navržen tak, aby reagoval na různé vstupní podmínky a prováděl odpovídající akce v závislosti na aktuálním stavu kotle.

Na obrázku 8.11 je několik vstupních a výstupních proměnných, které ovlivňují jeho chování. Mezi vstupy patří signály pro automatický a manuální režim, změnu režimu, stav termostatu a stav alarmu. Výstupy pak indikují stav zapalovací tyče, režim ventilátoru spalin, režim dopravníku a aktuální stav kotle.

```

VAR_INPUT //Definuje vstupní proměnné
  IN_Automat : bool;
  IN_Manual : bool;
  IN_ModeChange : bool;
  IN_Termostat : bool;
  IN_AlarmReset : bool;
END_VAR

VAR_OUTPUT //Definuje výstupní proměnné
  OUT_IgnitionRod: bool;
  OUT_CombustionFanMode : Bolier_CombFanMode_Enum;
  OUT_FillingConveyorMode : Bolier_FillingConvMode_Enum;
  OUT_ActualStatus: Boiler_ActualStatus_Enum;
END_VAR

```

Obr. 8.11: Vstupní a výstupní proměnné

Na obrázku 8.12 jsou definovány časovače pro sledování různých časových intervalů, jako je doba dávky paliva pro zapalování, zpoždění spuštění ventilátoru spalin

a maximální doba zapalování. Kromě toho jsou zde proměnné pro zapalovací proces bez spuštěné zapalovací tyče, aktuální krok procesu a aktuální stav chybné funkce programu.

`ActualStep` určuje aktuální krok stavového automatu, zatímco `ActualStatus` označuje stav kotle podle výčtového typu `Boiler_ActualStatus_Enum`. Další proměnné, jako `IgnitionWithoutIgnitionRod`, `Timer_ShortTermShutdown` a `Timer_MaxIgnitionTime`, zajišťují bezpečný průběh zapalovacího procesu a zároveň poskytují zpětnou vazbu o celkovém chování řídicího programu. Proměnné `Error_Active` a `Error_Status` jsou klíčové pro detekci a správu chyb programu, což umožňuje optimalizaci řízení a bezpečnost provozu kotle.

```
VAR //Interní proměnné pro řízení stavového automatu
ActualStep: uint;
ActualStatus: Boiler_ActualStatus_Enum;

IgnitionWithoutIgnitionRod : BOOL;

Timer_ShortTermShutdown : TOF;
Timer_IgnitionDose : TON;
Timer_CombFanDelay : TON;
Timer_CombFanStopDelay : TON;

Timer_MaxIgnitionTime : TON;

Timer_DelayForDetectionToStop : TON;

Error_Active: bool;
Error_Status: word;
```

Obr. 8.12: Definování časovačů

Tento úsek kódu, viz obrázek 8.13, provádí několik operací souvisejících s časovači a změnou režimu provozu kotle. Pokud dojde k vypršení časovače `Timer_IgnitionDose`, což signalizuje, že doba pro dávku paliva pro zapálení skončila, časovač je vypnut a režim dopravníku paliva se nastaví na `OFF`.

Pokud časovač `Timer_CombFanDelay` vyprší, což indikuje, že došlo ke zpoždění spuštění ventilátoru spalin, režim tohoto ventilátoru se přepne na `ON_REGULATION`, což znamená, že je aktivní regulace.

Pokud je detekována změna režimu (`IN_ModeChange`), provedou se následující operace: Pokud nový režim je automatický (`IN_Automat`), aktuální krok se nastaví na 0. Výstupy pro zapalovací tyč, režim ventilátoru spalin a režim dopravníku paliva se nastaví na `OFF`. Stav kotle se nastaví na `WAITING` (čekání). Všechny časovače jsou vypnuty, aby se kotel připravil na nový režim provozu.

```

//Podmínky pro časovače
if(Timer_IgnitionDose.Q) then
  Timer_IgnitionDose.IN := false;
  OUT_FillingConveyorMode := OFF;
end_if;

IF Timer_CombFanDelay.Q then
  OUT_CombustionFanMode := ON_REGULATION;
ELSE
  OUT_CombustionFanMode := OFF;
END_IF;

IF IN_ModeChange THEN

  IF IN_Automat THEN
    ActualStep := 0;
  END_IF;

  OUT_IgnitionRod := false;
  OUT_CombustionFanMode := OFF;
  OUT_FillingConveyorMode := OFF;
  ActualStatus := WAITING;

  Timer_IgnitionDose.IN := false;
  Timer_CombFanDelay.IN := false;
  Timer_MaxIgnitionTime.IN := false;

END_IF;

```

Obr. 8.13: Podmínky pro časovače

Tento kód, viz obrázek 8.14 zajišťuje, že když je vstupní proměnná `IN_AlarmReset` aktivní, systém resetuje všechny hodnoty související s chybovým stavem. Nastaví se počáteční krok stavového automatu (`ActualStep := 0`) a aktuální stav na `WAITING`, což indikuje, že systém je v čekacím režimu. Proměnná `Error_Status` je vynulována a `Error_Active` je nastavena na `false`, čímž se zruší aktivní chyby. Nakonec je proměnná `OUT_ActualStatus` nastavena na aktuální stav systému pro sledování stavu z externích zařízení.

```

//Podmínka pro resetování alarmu
if IN_AlarmReset THEN
  ActualStep := 0;
  ActualStatus := WAITING;
  Error_Status := 0;

  Error_Active := false;
END_IF;

OUT_ActualStatus := ActualStatus;

```

Obr. 8.14: Podmínky pro resetování

V prvním kroku na obrázku 8.15 s názvem `Waiting` je kontrolováno, zda je aktuální krok roven nule. Pokud ano, je nastaven stav na `WAITING` a provádí se další testování podmínek. Pokud jsou podmínky splněny (`IN_Termostat` a `IN_Automat` jsou aktivní), krok se přesune na hodnotu 10.

```

// Waiting
if ActualStep = 0 then
  ActualStatus := WAITING;

  if IN_Termostat and IN_Automat then
    ActualStep := 10;
  end_if;

```

Obr. 8.15: Aktuální krok = čekání

Ve druhém kroku na obrázku 8.16 s názvem Ignition start (Začátek zapalování) je zahájeno zapalování kotle. Stav je nastaven na IGNITION a spouští se plnění zapalovací dávky a zpoždění ventilátoru spalování.

```

// Ignition start
// Začátek plnění zapalovací dávkou; zpoždění ventilátoru
if ActualStep = 10 then
  ActualStatus := IGNITION;

  OUT_FillingConveyorMode := ON_IGNITION;
  Timer_IgnitionDose.IN := true;

  Timer_CombFanDelay.IN := true;

  ActualStep := 11;
  return;
end_if

```

Obr. 8.16: Aktuální krok = zapalování

Ve třetím kroku, viz obrázek 8.17, řídicí jednotka rozhoduje, zda zapne zapalovací tyč. Pokud kotel byl aktivní před méně než třemi hodinami, předpokládá se, že na roštu jsou stále žhavé uhlíky a zátopová dávka chytne i bez zapalovací tyče. Tato optimalizace vede k úsporám odebíraného výkonu. Tedy pokud časovač Timer_ShortTermShutdown nevypršel, zapalovací tyč se nesepe.

```

// Rozhodnutí zapal. tyč ON/OFF
if ActualStep = 11 then

  IgnitionWithoutIgnitionRod := false;

  // Pokud časovač krátkodobého vypnutí nevypršel, nezapínej tyč zapalování.
  if (Timer_ShortTermShutdown.Q) then
    IgnitionWithoutIgnitionRod := true;
  end_if;

  ActualStep := 12;
  return;
end_if

```

Obr. 8.17: Podmínka pro zapnutí zapalovací tyče

Ve čtvrtém kroku, viz obrázek 8.18, je zapnuta zapalovací tyč, pokud nebylo rozhodnuto, že nebude zapnuta, a spouští se časovač Timer_MaxIgnitionTime.

```

// Zapalovací tyč ON
if ActualStep = 12 then

    if (not IgnitionWithoutIgnitionRod) then
        OUT_IgnitionRod := true;
    end_if;

    Timer_MaxIgnitionTime.IN := true;

    ActualStep := 13;
    return;
end_if

```

Obr. 8.18: Zapalovací tyč zapnuta

Ve pátém kroku, viz obrázek 8.19, je detekováno zapálení ohně. Pokud časovač Timer_MaxIgnitionTime (maximální čas zapnutí zapalovací tyče) vypršel, je vyvolána chyba a stav se nastaví na ERROR. Pokud je teplota spalín vyšší než 30 °C, proces pokračuje a krok se přesune na 14. Timer_MaxIgnitionTime.

```

// Detekce zapálení
if ActualStep = 13 then

    // Pokud dojde k vypršení časovače "Timer_MaxIgnitionTime",
    // nastavte alarm a přechod na krok poruchy.
    IF (Timer_MaxIgnitionTime.Q) then
        Timer_MaxIgnitionTime.IN := false;
        IF (IgnitionWithoutIgnitionRod) then
            IgnitionWithoutIgnitionRod := false;
            ActualStep := 12;
            return;
        else
            Error_Active := true;
            Error_Status := 16#F011;
            ActualStep := 9999;
            return;
        end_if;
    end_if;

    // Pokud je teplota spalín vyšší než 30 °C, pokračuj.
    IF (DbBoiler.Actual.Temps.Combustion >= DbBoiler.Setpoints.Temps.IgnitionCombTemp) then
        ActualStep := 14;
        return;
    end_if;
end_if

```

Obr. 8.19: Detekce zapálení

V šestém kroku, viz obrázek 8.20, se vypne zapalovací tyč a oba časovače Timer_IgnitionDose (zapalovací dávka) a Timer_MaxIgnitionTime jsou vypnuty.

```

// Vypnutí zapalovací tyče a časovačů
if ActualStep = 14 then

    OUT_IgnitionRod := false;
    Timer_IgnitionDose.IN := false;
    Timer_MaxIgnitionTime.IN := false;

    ActualStep := 20;
    return;
end_if

```

Obr. 8.20: Vypnutí zapalovací tyče a časovačů

V kroku označeném jako Running, viz obrázek 8.21, je stav systému nastaven na RUNNING. Dále je zapnut režim regulace pro doplňovací dopravník (Filling

Conveyor), aby mohl pracovat s optimálním nastavením. Časovač `Timer_ShortTermShutdown` je spuštěn pro detekci krátkodobého vypnutí.

Časovač `Timer_ShortTermShutdown` se aktivuje (`IN := true`), což znamená, že se sleduje, zda nedojde ke krátkodobému výpadku. Zároveň se aktivuje časovač `Timer_DelayForDetectionToStop`, který porovnává aktuální teplotu spalín s minimální požadovanou teplotou (`MinRunningCombTemp`). Pokud je aktuální teplota spalín nižší než tato minimální hodnota, časovač se spustí a indikace případné poruchy se nastaví podle další části kódu.

```
// Running
if ActualStep = 20 then
  ActualStatus := RUNNING;
  OUT_FillingConveyorMode := ON_REGULATION;
  Timer_ShortTermShutdown.IN := true; // Turn on

  Timer_DelayForDetectionToStop.IN := (DbBoiler.Actual.Temps.Combustion < DbBoiler.Setpoints.Temps.MinRunningCombTemp);
```

Obr. 8.21: Logika časovačů při spuštění kotle

Tato část kódu, viz obrázek 8.22, kontroluje kritické podmínky během provozu kotle. Pokud výstup časovače `Timer_DelayForDetectionToStop` signalizuje poruchu nebo je překročena maximální teplota vody, je nastavena příslušná chybová proměnná a proces kotle přechází do chybového stavu. Podmínka termostatu pak určuje, zda má kotel přejít do kroku pro snížení provozu nebo zastavení.

```
// Zastavení detekce časovače
IF(Timer_DelayForDetectionToStop.Q) then
  Error_Active := true;
  Error_Status := 16#F012;
  ActualStep := 9999;
  return;
end_if;

// Podmínka pro překročení Max. teploty vody
IF((DbBoiler.Actual.Temps.Water >= DbBoiler.Setpoints.Temps.MaxWaterTemp) ) then
  Error_Active := true;
  Error_Status := 16#F013;
  ActualStep := 9999;
  return;
end_if;

if not IN_Termostat then
  ActualStep := 30;
  return;
end_if;
return;
```

Obr. 8.22: Kontrolní podmínky během provozu kotle

V této části kódu, viz obrázek 8.23, je logika pro zastavení kotle. Když je krok `ActualStep` nastaven na 30, stav kotle se změní na `STOPPING`, což znamená vypnutí. Časovače se deaktivují a krok je nastaven na 31, což značí další fázi. Poté je aktivován časovač `Timer_CombFanStopDelay`, který zajišťuje vypnutí ventilátoru po stanovené době.

```

// Zastavení
if ActualStep = 30 then
    ActualStatus := STOPPING;
    OUT_FillingConveyorMode := OFF;
    Timer_ShortTermShutdown.IN := false; // Turn off
    Timer_DelayForDetectionToStop.IN := false;

    ActualStep := 31;
    return;
end_if

// Zastavení - počkat na vypnutí ventilátoru
if ActualStep = 31 then
    Timer_CombFanStopDelay.IN := true;

    ActualStep := 32;
    return;
end_if

```

Obr. 8.23: Zastavení kotle

V této části kódu, viz obrázek 8.24, je definováno vypnutí ventilátoru po určitém zpoždění. Jakmile časovač `Timer_CombFanStopDelay` signalizuje vypršení zpoždění, deaktivuje se časovač `Timer_CombFanDelay`, ventilátor se vypne (OFF) a aktuální krok se nastaví na 33. Následně se stavový automat vrací do výchozího stavu nastavením `ActualStep` na 0.

```

// Zastavení - vypnutí ventilátoru po zpoždění
if ActualStep = 32 then
    IF (Timer_CombFanStopDelay.Q) then
        Timer_CombFanDelay.IN := false;
        OUT_CombustionFanMode := OFF;
        Timer_CombFanStopDelay.IN := false;
        ActualStep := 33;
        return;
    end_if;
    return;
end_if

if ActualStep = 33 then
    ActualStep := 0;
    return;
end_if

```

Obr. 8.24: Vypnutí ventilátoru po zpoždění

```

// Chyba
if ActualStep = 9999 then
    ActualStatus := ERROR;

    OUT_IgnitionRod := false;

    Timer_IgnitionDose.IN := false;
    OUT_FillingConveyorMode := OFF;

    Timer_CombFanDelay.IN := false;
    OUT_CombustionFanMode := OFF;

    Timer_DelayForDetectionToStop.IN := false;

```

Obr. 8.25: Chybový stav

Tento kód, viz obrázek 8.25, nastavuje stav systému na `ERROR`, pokud je krok `ActualStep` roven 9999. Vypíná všechny prvky, jako je zapalovací tyč, dopravník

paliva, ventilátor spalin a odpovídající časovače, aby se systém bezpečně zastavil a nedošlo k poškození.

8.4.3 Funkční blok Boiler_Control

Funkční blok `Boiler_Control`, který je volán do programu `prgMain`, slouží k řízení některých aspektů kotle na základě různých podmínek. Tato kapitola se bude věnovat pouze části kódu zaměřené na řízení reálného kotle. Zbylá část kódu zaměřená na simulaci bude popsána v kapitole 9.

Kód na obrázku 8.26 řídí spínání dopravníku na základě časových intervalů. Nejprve je spočítána doba pauzy dopravníku (`FillingConvPauseTimeCalculation`) pomocí vzorce, který zohledňuje maximální dobu pauzy (`DbBoiler.Setpoints.Times.FillingDoseMaxPause`) a aktuální výkon dopravníku (`BoilerSimulationOutputs.FillingConveyorOP`).

Poté je časovač pauzy dopravníku (`FillingConvPauseTimer`) nastaven s časem trvání určeným výše vypočítanou dobou pauzy. Intervalový časovač dopravníku (`FillingConvIntervalTime`) je nastaven s dobou trvání určenou z nastavení kotle (`DbBoiler.Setpoints.Times.FillingDoseInterval`).

Nakonec jsou vstupy časovačů nastaveny na základě podmínek: časovač pauzy je aktivován, pokud výkon dopravníku je větší než nula a intervalový časovač dopravníku není aktivní. Intervalový časovač je spuštěn, pokud časovač pauzy je aktivní.

```
// Plnění kotle palivem časové dělení mezi pauzou a aktivním dopravníkem
FillingConvPauseTimeCalculation := REAL_TO_TIME(TIME_TO_REAL(DbBoiler.Setpoints.Times.FillingDoseMaxPause)
* ((100.0 - BoilerSimulationOutputs.FillingConveyorOP) / 100.0));
FillingConvPauseTimer(PT := FillingConvPauseTimeCalculation);
FillingConvIntervalTime(PT := DbBoiler.Setpoints.Times.FillingDoseInterval);
FillingConvPauseTimer.IN := BoilerSimulationOutputs.FillingConveyorOP > 0.0 AND NOT FillingConvIntervalTime.Q;
FillingConvIntervalTime.IN := FillingConvPauseTimer.Q;
```

Obr. 8.26: Plnění kotle palivem

První řádek na obrázku 8.27 definuje chování časovače pauzy popelníku (`AshTrayPauseTimer`). Jeho vstupní podmínka (IN) je nastavena na `TRUE`, pokud není aktivní výstupní signál intervalového časovače popelníku (`AshTrayIntervalTime.Q`) a zároveň je aktuální stav kotle (`DbBoiler.Actual.ActualBoilerStatus`) nastaven na `RUNNING` (běžící). Doba trvání časovače je určena podle hodnoty `AshTrayPause` z nastavení kotle (`DbBoiler.Setpoints.Times.AshTrayPause`).

Druhý řádek nastavuje chování intervalového časovače popelníku (`AshTrayIntervalTime`). Jeho vstupní podmínka (IN) je definována jako výstupní hodnota časovače

pauzy popelníku (`AshTrayPauseTimer.Q`). Doba trvání tohoto intervalového časovače je určena hodnotou `AshTrayInterval` z nastavení kotle (`DbBoiler.Setpoints.Times.AshTrayInterval`).

```
// Řízení odpopelování

AshTrayPauseTimer(IN := NOT AshTrayIntervalTime.Q AND DbBoiler.Actual.ActualBoilerStatus = RUNNING,
  PT := DbBoiler.Setpoints.Times.AshTrayPause);
AshTrayIntervalTime(IN := AshTrayPauseTimer.Q, PT := DbBoiler.Setpoints.Times.AshTrayInterval);

BoilerSimulationOutputs.AshTray := (Mode_AutomaticManual AND AshTrayPauseTimer.Q)
  OR ( NOT Mode_AutomaticManual AND DbBoilerManualMode.AshTrayEnable);
```

Obr. 8.27: Odpopelování kotle

8.4.4 Program `CyclicInterrupt_400ms`

Tento úsek kódu, viz obrázek 8.28, implementuje PID regulátor teploty spalin kotle. Funkce `CombustionTempRegulator` je volána s několika vstupy: aktuální teplotou spalin (`Input`), požadovanou hodnotou teploty spalin (`Setpoint`), režimem regulátoru (`Mode`), manuálně nastaveným výkonem ventilátoru spalin (`Man_OP`) a výstupem regulátoru, který je předán do proměnné `PID_Out_CombustionReg` v kontrolních proměnných kotle.

```
// PID regulator pro teplotu spalin
CombustionTempRegulator(
  Input := DbBoiler.Actual.Temps.Combustion
  ,Setpoint := DbBoiler.Setpoints.Temps.CombustionTemp
  ,Mode := (Mode_AutomaticManual AND (DbBoiler.ControlVars.CombustionFan.CombustionFanMode = ON_REGULATION))
  ,Man_OP := DbBoilerManualMode.CombustionFanOP
  ,Output => DbBoiler.ControlVars.CombustionFan.PID_Out_CombustionReg);
```

Obr. 8.28: PID regulátor pro teplotu spalin

Stejný princip se uplatňuje i v kódu pro regulaci teploty vody, viz obrázek 8.29. Regulátor přebírá požadovanou teplotu vody ze struktury `DbBoiler`, uloženou v položce `Setpoints.Temps.WaterTemp`, a následně od této hodnoty odečítá 5,0 stupňů. Tato upravená hodnota slouží jako cílová teplota pro regulaci vody. Úprava teploty o -5 °C je založena na doporučeních výrobce kotle Verner A251. Tato úprava umožňuje kotli jet na nižším výkonu, pokud předtím dosáhl požadované teploty (např. 60 °C). Výkon kotle je tedy snížen a topný systém je schopen teplo spotřebovat.

```
// PID regulator pro teplotu vody
WaterTempRegulator(
  Input := DbBoiler.Actual.Temps.Water
  ,Setpoint := DbBoiler.Setpoints.Temps.WaterTemp - 5.0
  ,Mode := (Mode_AutomaticManual AND (DbBoiler.ControlVars.FillingConv.FillingConveyorMode = ON_REGULATION))
  ,Man_OP := DbBoilerManualMode.FillingConveyorOP
  ,Output => DbBoiler.ControlVars.FillingConv.PID_Out_WaterReg);
```

Obr. 8.29: PID regulátor pro teplotu vody

Nakonec jsou definovány konstanty pro PID regulátory teploty spalín a vody, viz obrázek 8.30. Pro každý regulátor jsou nastaveny hodnoty parametrů proporcionálního zesílení (K_p), integračního členu (K_i) a derivativního členu (K_d), stejně jako hodnota časové konstanty (τ), délka cyklu (cycle) a mez výstupu (outLimMax a outLimMin).

Je důležité poznamenat, že tyto konstanty jsou pouze náhodně zvolené a pro reálné nasazení regulátorů je nutné provést nastavení a otestování na reálném kotli s ohledem na specifika provozního prostředí.

```
// PID regulator - Nastavení konstant
CombustionTempRegulator.PID_Const.Kp := 0.4;
CombustionTempRegulator.PID_Const.Ki := 0.01;
CombustionTempRegulator.PID_Const.Kd := 0.0;
CombustionTempRegulator.PID_Const.tau := 0.0;
CombustionTempRegulator.PID_Const.cycle := 0.400;
CombustionTempRegulator.PID_Const.outLimMax := 100.0;
CombustionTempRegulator.PID_Const.outLimMin := 0.0;

WaterTempRegulator.PID_Const.Kp := 1.0;
WaterTempRegulator.PID_Const.Ki := 0.1;
WaterTempRegulator.PID_Const.Kd := 0.0;
WaterTempRegulator.PID_Const.tau := 0.0;
WaterTempRegulator.PID_Const.cycle := 0.400;
WaterTempRegulator.PID_Const.outLimMax := 100.0;
WaterTempRegulator.PID_Const.outLimMin := 0.0;
```

Obr. 8.30: PID regulátor - konstanty

8.4.5 Funkční blok PID_C_Regulator

Tento kód představuje funkční blok PID regulátoru, což je algoritmus řízení používaný k udržování teploty na předem stanovené hodnotě (Setpoint). Regulátor porovnává aktuální hodnotu měřené veličiny s požadovanou hodnotou a na základě této chyby vypočítává výstupní signál, který ovlivňuje řízený proces. Algoritmus zahrnuje proporcionální, integrační a diferenciální akci, které společně přispívají k rychlé a stabilní regulaci.

První část kódu, viz obrázek 8.31, obsahuje podmínku, která ověřuje, zda je PID regulátor v automatickém režimu ($\text{Mode} == \text{true}$). Pokud ano, je vypočítávána aktuální chyba regulace (actualError), která je rozdílem mezi požadovanou hodnotou (Setpoint) a aktuálním vstupem (Input). Poté je vypočtena složka proporcionální akce (proportional) jako součin aktuální chyby a proporcionální konstanty (PID_Const.Kp). Dále je aktualizována hodnota integrátoru (integrator) pomocí metody diskrétního integrátoru s použitím Simpsonova pravidla.

```

if(Mode) then
// Automatic mod
actualError := Setpoint - Input;

proportional := actualError * PID_Const.Kp;

integrator := integrator + 0.5 * PID_Const.Ki * PID_Const.cycle * (actualError + previousError);

```

Obr. 8.31: Automatický mod pro PID regulaci

Tato část kódu vypočítává horní a dolní limity pro integrátorovou akci PID regulátoru, viz obrázek 8.32. Nejprve se porovnává hodnota maximálního limitu (`PID_Const.outLimMax`) s hodnotou proporcionálního členu (`proportional`). Pokud je maximální limit větší než proporcionální člen, pak je nastaven limit integrátoru na rozdíl mezi maximálním limitem a proporcionálním členem. Pokud je maximální limit menší nebo roven proporcionálnímu členu, je limit integrátoru nastaven na 0,0.

Stejný postup se použije i pro výpočet dolního limitu (`limitMinInt`). Nejprve se porovnává hodnota minimálního limitu (`PID_Const.outLimMin`) s proporcionálním členem. Pokud je minimální limit menší než proporcionální člen, je limit integrátoru nastaven na rozdíl mezi proporcionálním členem a minimálním limitem. Pokud je minimální limit větší nebo roven proporcionálnímu členu, je limit integrátoru nastaven na 0,0.

```

// Výpočet mezi integrátoru
if PID_Const.outLimMax > proportional then
  limitMaxInt := PID_Const.outLimMax - proportional;
else
  limitMaxInt := 0.0;
END_IF;

if PID_Const.outLimMin < proportional then
  limitMinInt := PID_Const.outLimMin - proportional;
else
  limitMinInt := 0.0;
END_IF;

```

Obr. 8.32: Výpočet mezi integrace pro PID regulator

Tato část kódu, viz obrázek 8.33, implementuje mechanismus anti-wind-up, který slouží k omezení efektu integrátoru v PID regulátoru a zabraňuje přetečení nebo podtečení integrátoru při velkých změnách vstupních hodnot. Pokud je regulátor v automatickém režimu (`Mode = true`), kontroluje se hodnota integrátoru (`integrator`) a porovnává se s vypočtenými limity (`limitMaxInt` pro horní limit a `limitMinInt` pro dolní limit). Pokud hodnota integrátoru překročí horní limit, je nastavena na hodnotu horního limitu. Pokud je hodnota integrátoru menší než dolní limit, je nastavena na hodnotu dolního limitu.

Poté je vypočtena hodnota diferenciátoru (`differentiator`) pomocí vzorce pro diskrétní derivaci. Nakonec je spočtena výsledná hodnota výstupu regulátoru (`local-`

Output) jako součet proporcionálního členu, integrátoru a diferenciátoru. Pokud je regulátor v automatickém režimu, je nastaven příznak `StatusWord.0` na `true`, což označuje, že regulátor pracuje v automatickém režimu. V opačném případě, pokud je regulátor v manuálním režimu (`Mode = false`), je hodnota výstupu nastavena na hodnotu manuálního nastavení (`Man_OP`) a příznak `StatusWord.0` je nastaven na `false`, což označuje, že regulátor pracuje v manuálním režimu.

```
// Anti-wind-up via integrator clamping
if integrator > limitMaxInt then
    integrator := limitMaxInt;
elseif integrator < limitMinInt then
    integrator := limitMinInt;
END_IF;

differentiator := -(2.0 * PID_Const.Kd * (Input - lastInputValue)) / PID_Const.cycle;

localOutput := proportional + integrator + differentiator;

StatusWord.0 := true;

else

// manual mode

localOutput := Man_OP;

StatusWord.0 := false;

end_if;
```

Obr. 8.33: Mechanismus anti-wind-up pro PID regulátor

```
// Výstupní meze
if localOutput >= PID_Const.outLimMax then
    localOutput := PID_Const.outLimMax;
    StatusWord.9 := true;
elseif localOutput <= PID_Const.outLimMin then
    localOutput := PID_Const.outLimMin;
    StatusWord.10 := true;
else
    StatusWord.9 := false;
    StatusWord.10 := false;
END_IF;

lastInputValue := Input;
previousError := actualError;

Output := localOutput;
```

Obr. 8.34: Výstupní meze PID regulátoru

Nakonec se implementuje omezení výstupní hodnoty regulátoru na definované meze (`PID_Const.outLimMin` a `PID_Const.outLimMax`), viz obrázek 8.34. Pokud vypočtená hodnota výstupu (`localOutput`) překročí horní mez (`PID_Const.outLimMax`), je nastavena na hodnotu horní meze a příznak `StatusWord.9` je nastaven na `true`, což označuje překročení horní meze. Pokud vypočtená hodnota výstupu klesne pod dolní mez (`PID_Const.outLimMin`), je nastavena na hodnotu dolní meze a příznak `StatusWord.10` je nastaven na `true`, což označuje pokles pod dolní mez. Pokud hodnota výstupu je v mezích, jsou příznaky `StatusWord.9` a `StatusWord.10` nastaveny

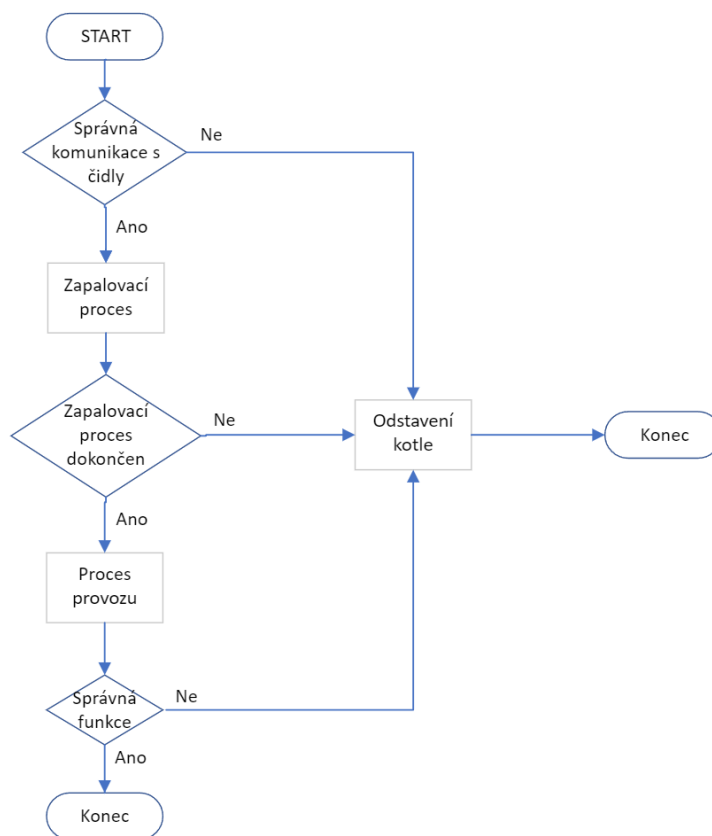
na `false`. Poté jsou aktualizovány hodnoty `lastInputValue` a `previousError` a vypočtená hodnota výstupu je přiřazena výstupní proměnné `Output`.

9 Simulace

Tato kapitola bude obsahovat popisy vývojových diagramů pro algoritmus řízení kotle. Následně bude popsán kód, který má obsáhlé jednotlivé proměnné pro simulované řízení, a poté bude popsáno samotné prostředí simulace pomocí výstřižků z vývojového prostředí Mosaic. Simulace jsou rozděleny na jednotlivé scénáře (proces zapalování, proces provozu, doběh, nesplnění podmínky pro proces zapalování, prázdný zásobník paliva, přerušení komunikace PLC – čidlo) a nakonec zaznamenány na videa a přidány do přílohy Diplomové práce.

9.1 Vývojové diagramy

Pro lepší pochopení funkce algoritmu, byly vytvořeny vývojové diagramy, na kterých budou popsány jednotlivé kroky řízení a rozhodování tohoto programu.



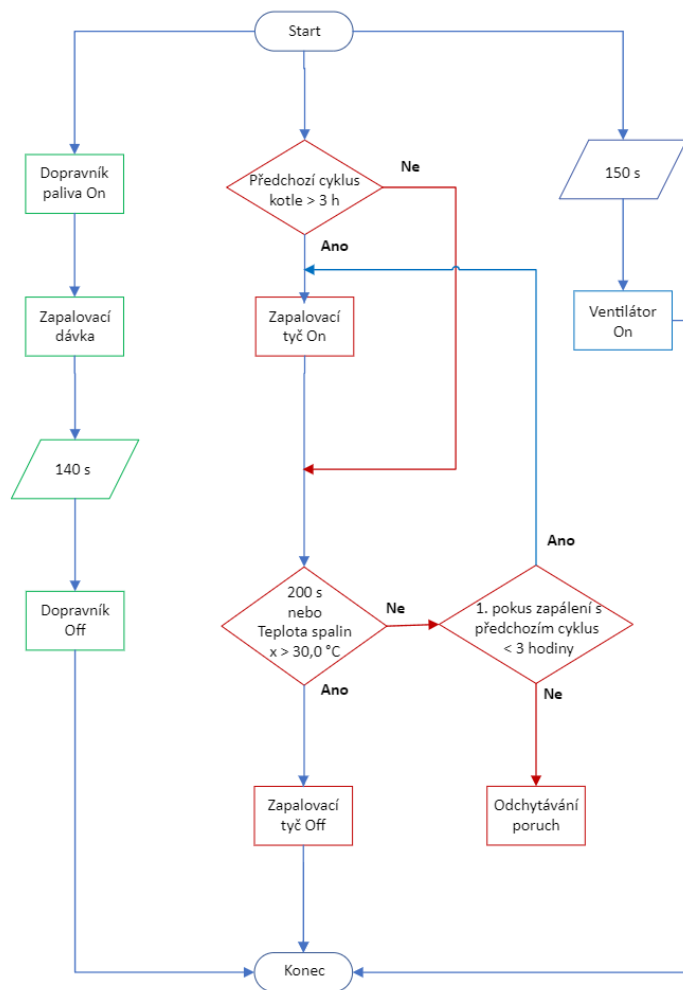
Obr. 9.1: Obecný vývojový diagram pro řízení kotle

Vývojový diagram na obrázku 9.1 zobrazuje proces řízení kotle na tuhá paliva, a popisuje jednotlivé kroky od spuštění až po možné ukončení činnosti. Proces začíná krokem „START“, po kterém následuje první rozhodovací krok: ověření správné

komunikace s čidly. Pokud je komunikace s čidly v pořádku, pokračuje se zapalovacím procesem. Pokud komunikace není správná, proces přechází přímo na odstavení kotle, aby se zabránilo jakémukoli nebezpečí způsobenému vadnými čidly.

V zapalovacím procesu je zajištěn rozběh kotle a příprava na provoz. Pokud zapalovací proces proběhne úspěšně, spustí se proces provozu kotle. Pokud zapalovací proces nebyl úspěšně dokončen, zařízení se odstaví.

Ve fázi provozu kotle se kontroluje správná funkce zařízení. Pokud vše funguje správně, proces končí úspěšným dokončením. Pokud však kotel nefunguje správně, opět se přechází k odstavení kotle. Odstavení kotle slouží jako bezpečnostní opatření, které zabrání jakýmkoli následkům způsobeným nesprávným provozem.



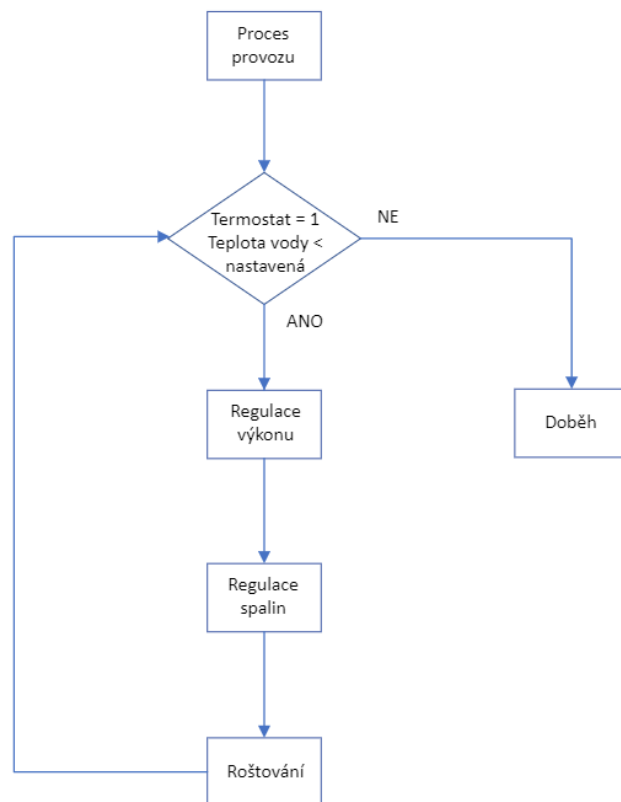
Obr. 9.2: Vývojový diagram pro proces zapálení

Tento vývojový diagram, viz obrázek 9.2, popisuje proces zapalování a obsahuje tři paralelní větve řízení.

První větev se zaměřuje na dopravník paliva. Na začátku procesu se zapne dopravník, který za 140 sekund naplní kotel zátopovou dávkou pelet. Po uplynutí této doby se dopravník automaticky vypne, což zajistí správné množství paliva pro zapálení.

Druhá větev se týká zapalovací tyče. Pokud poslední cyklus kotle (od jeho posledního vypnutí) trval déle než tři hodiny, zapalovací tyč se zapne. Následně systém zjišťuje, zda se během 200 sekund nebo při dosažení teploty spalin nad 30 °C zátopová dávka paliva úspěšně zapálila. Pokud byla některá z těchto podmínek splněna, zapalovací tyč se vypne. Pokud však ani po 200 sekundách není teplota spalin vyšší než 30 °C, přejde se k další rozhodovací podmínce. Ta určuje, zda se jedná o první pokus zapálení s předchozím cyklem kratším než tři hodiny. Pokud ano, proces se vrací k zapnutí zapalovací tyče a cyklus se opakuje. Pokud ne, systém hlásí poruchu.

Třetí větev je zaměřena na ventilátor. Ten má při spuštění zapalování zpoždění 150 sekund, aby se zapnul až po naplnění kotle zátopovou dávkou paliva. Tím se zajistí optimální podmínky pro účinné zapálení paliva a hoření.



Obr. 9.3: Vývojový diagram pro proces provozu

Tento vývojový diagram, viz obrázek 9.3, popisuje proces provozu kotle.

V první fázi se kontroluje stav termostatu a aktuální teplota vody. Pokud je termostat nastaven na hodnotu 1 a zároveň je teplota vody nižší než požadovaná, kotel přejde do fáze regulace výkonu. V opačném případě pokračuje do fáze doběhu.

V rámci regulace výkonu se v závislosti na aktuální teplotě vody upravuje množství paliva přiváděného do kotle. Je-li teplota vody příliš nízká, přísun paliva se zvýší, aby se dosáhlo vyššího výkonu. Pokud je naopak teplota vody příliš vysoká, přísun paliva se sníží. Tímto způsobem se udržuje optimální teplota vody a výkon kotle.

Regulace teploty spalin je prováděna změnou rychlosti otáček ventilátoru. Zvýšením nebo snížením rychlosti se upravuje odvod spalin a teplota spalování, což pomáhá udržovat efektivní provoz kotle.

Roštování je vždy prováděno po předem nastaveném časovém intervalu. Tento proces umožňuje pravidelné odstraňování popela z kotle, čímž je zajištěn nepřetržitý a efektivní provoz.

Pokud termostat je nastaven na hodnotu 0 nebo je teplota vody již dostatečně vysoká, kotel přejde přímo do fáze doběhu, kdy se kotel bezpečně odstaví.

Tento proces tak zajišťuje efektivní a automatizovaný provoz kotle na tuhá paliva díky regulaci přívodu paliva, spalin a pravidelnému roštování.

9.2 Funkční blok Boiler_Simulation

Tento kód je funkční blok s názvem `Boiler_Simulation`, který simuluje provoz kotle. Je navržen tak, aby věrně reprezentoval různé scénáře provozu, jako je zapálení kotle, přechod do provozního režimu a zastavení, stejně jako různé chybové stavy. Kód nabízí možnost simulovat nejen základní provozní stavy, ale také různé chybové scénáře, což umožňuje testování řízení kotle bez nutnosti fyzického zařízení.

Nejprve je definována instance detekce hrany nazvaná `FillingConveyorEdge`, viz obrázek 9.4. Parametr `CLK` je nastaven na signál `BoilerSimulationOutputs.FillingConveyor`. To znamená, že výstup `Q` této instance bude aktivní při vzestupné hraně signálu dopravníku.

```
// Aktivace dopravníku
FillingConveyorEdge(CLK := BoilerSimulationOutputs.FillingConveyor);

// Aktivace hmotnosti
IF FillingConveyorEdge.Q THEN
  WeightOfPeletsInBoiler := WeightOfPeletsInBoiler + WeightPerTakt;
END_IF;
```

Obr. 9.4: Simulace doplňování paliva

Následně na obrázku 9.4, pokud je výstup `FillingConveyorEdge.Q` aktivní (hodnota `true`), do proměnné `WeightOfPeletsInBoiler` (hmotnost pelet v kotli) se přidá konstanta `WeightPerTakt`, což simuluje doplnění paliva do kotle.

Tato část kódu, viz obrázek 9.5, definuje konstantu pro aktuální zdroj tepla a nastavuje dva časovače spolu s detekcí vzestupné hrany.

Nejprve se pomocí konstrukce `CASE` určuje hodnota proměnné `DbBoiler.Setpoints.ActualHeatSource`, která se porovnává s několika možnostmi. Podle hodnoty je přiřazena odpovídající konstanta. Pokud je hodnota 1, nastaví se proměnná `ActualHeatSourceConst` na 0.35. Pokud je hodnota 2, nastaví se na 0.25. Pokud je hodnota 3, nastaví se na 0.3. V ostatních případech je konstanta nastavena na 0.2.

Dva časovače, `Timer_1` a `Timer_2`, jsou nastaveny na periodu 250 ms. Každý z nich je aktivován, pokud druhý není. Toto časové posunutí vytváří signály, které lze použít ke generování vzestupné hrany.

Nakonec je detekce vzestupné hrany, `TimerRisingEdge`, nastavena se signálem `Timer_1.Q`. Tím je zajištěna detekce vzestupné hrany tohoto časovače. Jakmile je detekována vzestupná hrana, lze tento signál použít jako spouštěcí bod pro další procesy.

```

CASE DbBoiler.Setpoints.ActualHeatSource OF // Konstanty pro zdroj tepla
  1 : ActualHeatSourceConst := 0.35; // Pelety
  2 : ActualHeatSourceConst := 0.25; // Dřevo
  3 : ActualHeatSourceConst := 0.3; // Brikety
ELSE
  ActualHeatSourceConst := 0.2; // Bez definovaného typu paliva
END_CASE;

// Časovače pro generování vzestupných hran
Timer_1(IN := not Timer_2.Q, PT := t#250ms);
Timer_2(IN := Timer_1.Q, PT := t#250ms);

TimerRisingEdge(CLK := Timer_1.Q); // Detekce vzestupné hrany

```

Obr. 9.5: Výpočet a nastavení časovačů

Tato část kódu, viz obrázek 9.6, simuluje pokles teploty spalin a vody v kotli během stavů, kdy kotel čeká, zastavuje se, nebo je v chybovém stavu. Pokles teplot je řízen podle časovače s detekcí vzestupné hrany.

Nejprve podmínka `IF (DbBoiler.Actual.ActualBoilerStatus = WAITING OR DbBoiler.Actual.ActualBoilerStatus = STOPPING OR DbBoiler.Actual.ActualBoilerStatus = ERROR) AND TimerRisingEdge.Q THEN` ověřuje, zda je aktuální stav kotle `WAITING` (čekání), `STOPPING` (zastavování) nebo `ERROR` (chyba), a zda časovač signalizuje vzestupnou hranu.

Pokud jsou tyto podmínky splněny, kód sníží teplotu spalin (`BoilerSimulationInputs.CombustionTemp`) o 2,0 °C a teplotu vody (`BoilerSimulationInputs.WaterTemp`) o 0,5 °C.

Následně kód kontroluje, zda teploty neklesly pod minimální hranici 20,0 °C: - Pokud je teplota spalin nižší než 20,0 °C, nastaví se na 20,0 °C. - Pokud je teplota vody nižší než 20,0 °C, také se nastaví na 20,0 °C.

Tímto způsobem je zajištěno, že během stavů nečinnosti kotle teploty neklesnou pod určitou minimální hodnotu.

```
//Pokles teploty
IF (DbBoiler.Actual.ActualBoilerStatus = WAITING OR DbBoiler.Actual.ActualBoilerStatus
= STOPPING OR DbBoiler.Actual.ActualBoilerStatus = ERROR) AND TimerRisingEdge.Q THEN

    BoilerSimulationInputs.CombustionTemp := BoilerSimulationInputs.CombustionTemp - 2.0;
    BoilerSimulationInputs.WaterTemp := BoilerSimulationInputs.WaterTemp - 0.5;

    IF BoilerSimulationInputs.CombustionTemp < 20.0 then
        BoilerSimulationInputs.CombustionTemp := 20.0;
    END_IF

    IF BoilerSimulationInputs.WaterTemp < 20.0 then
        BoilerSimulationInputs.WaterTemp := 20.0;
    END_IF

END_IF;
```

Obr. 9.6: Simulace poklesu teploty

Tato část kódu, viz obrázek 9.7, simuluje růst teploty spalin a vody během zapalování kotle. Nejprve podmínka IF DbBoiler.Actual.ActualBoilerStatus = IGNITION AND TimerRisingEdge.Q THEN ověřuje, zda je aktuální stav kotle IGNITION (zapalování) a zda časovač signalizuje vzestupnou hranu.

Pokud je stav zapalování aktivní, kód ověřuje, zda není deaktivována simulace spalování při zapalování (DisableCombustionSimulationOnIgnite). Pokud simulace není deaktivována, teplota spalin (BoilerSimulationInputs.CombustionTemp) se zvýší. Zvýšení je vypočítáno jako součin aktuálního výkonu ventilátoru spalin (BoilerSimulationOutputs.CombustionFanOP) a konstanty zdroje tepla (ActualHeatSourceConst), snížený o požadavek na výkon (PowerDemand).

Teplota vody (BoilerSimulationInputs.WaterTemp) se pak také zvýší. Výpočet zahrnuje součin aktuální teploty spalin a koeficientu 0,03, a dále součin výkonu dopravníku paliva (BoilerSimulationOutputs.FillingConveyorOP) s koeficientem 0,1. Nakonec se výsledek sníží o požadavek na výkon (PowerDemand).

```
// Změny teploty spalin a vody
IF DbBoiler.Actual.ActualBoilerStatus = IGNITION AND TimerRisingEdge.Q THEN

    IF NOT DisableCombustionSimulationOnIgnite THEN
        BoilerSimulationInputs.CombustionTemp := BoilerSimulationInputs.CombustionTemp +
        (BoilerSimulationOutputs.CombustionFanOP * ActualHeatSourceConst) - PowerDemand;

        BoilerSimulationInputs.WaterTemp := BoilerSimulationInputs.WaterTemp +
        (BoilerSimulationInputs.CombustionTemp * 0.03) + (BoilerSimulationOutputs.FillingConveyorOP * 0.1) - PowerDemand;
    END_IF;

END_IF;
```

Obr. 9.7: Simulace růstu teploty během zapalování kotle

Tato část kódu, viz obrázek 9.8, simuluje změny teploty spalin a vody během běhu kotle (RUNNING), přičemž zohledňuje dostupnost paliva a možnost přehřátí.

Nejprve podmínka `IF DbBoiler.Actual.ActualBoilerStatus = RUNNING AND TimerRisingEdge.Q THEN` ověřuje, zda je kotel v běžícím stavu (RUNNING), a jestli časovač signalizuje vzestupnou hranu.

Pokud je palivo dostupné (`NOT NoFuelAvailable`), teplota spalin se zvýší. Výpočet je podobný jako při zapalování, tedy jako součin aktuálního výkonu ventilátoru spalin a konstanty zdroje tepla (`ActualHeatSourceConst`), snížený o požadavek na výkon (`PowerDemand`).

Pokud není povolena simulace přehřátí (`not OverheatSim`), teplota vody se vypočítává podobně jako při zapalování. Přidá se součin teploty spalin a koeficientu 0,03, a součin výkonu dopravníku paliva (`BoilerSimulationOutputs.FillingConveyorOP`) a koeficientu 0,1, snížený o požadavek na výkon.

Pokud je povolena simulace přehřátí, koeficient snížení výkonu (`PowerDemand`) je násoben 0,7, což simuluje vyšší účinnost ohřevu.

Pokud není palivo k dispozici (`ELSE` blok), teplota spalin a vody se postupně snižuje. Teplota spalin klesá o 2,0 °C a teplota vody o 0,5 °C.

Tímto způsobem je možné simulovat chování kotle během běhu, včetně reakce na situace s nedostatkem paliva nebo přehřátím.

```
//Změny teplot během běhu kotle
IF DbBoiler.Actual.ActualBoilerStatus = RUNNING AND TimerRisingEdge.Q THEN

  // Pokud je palivo k dispozici, začněte přidávat teplotu, pokud palivo není k dispozici, pouze snižte teplotu.
  IF NOT NoFuelAvailable then
    BoilerSimulationInputs.CombustionTemp := BoilerSimulationInputs.CombustionTemp +
      (BoilerSimulationOutputs.CombustionFanOP * ActualHeatSourceConst) - PowerDemand;

    if not OverheatSim then
      BoilerSimulationInputs.WaterTemp := BoilerSimulationInputs.WaterTemp +
        (BoilerSimulationInputs.CombustionTemp * 0.03) + (BoilerSimulationOutputs.FillingConveyorOP * 0.1) - PowerDemand;
    else
      BoilerSimulationInputs.WaterTemp := BoilerSimulationInputs.WaterTemp +
        (BoilerSimulationInputs.CombustionTemp * 0.03) + (BoilerSimulationOutputs.FillingConveyorOP * 0.1) - (PowerDemand*0.7);
    end_if
  end_if

ELSE
  BoilerSimulationInputs.CombustionTemp := BoilerSimulationInputs.CombustionTemp - 2.0;
  BoilerSimulationInputs.WaterTemp := BoilerSimulationInputs.WaterTemp - 0.5;
END_IF;
```

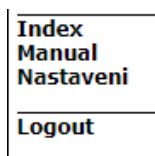
Obr. 9.8: Simulace změny teplot během běhu kotle

9.3 Simulační prostředí

Tato kapitola představuje prostředí pro simulace kotle, viz obrázek 9.9. Simulace lze provádět buď v prostředí Mosaic, nebo online prostřednictvím webového rozhraní na adrese 192.168.134.176, kde je vyžadováno přihlášení pomocí uživatelského jména a hesla k příslušnému PLC.

V simulačním prostředí jsou k dispozici tři hlavní režimy. Prvním režimem je automatické řízení kotle prostřednictvím PLC. Druhým je manuální režim, který

uživatelům umožňuje ruční spouštění jednotlivých částí kotle a jejich přímé ovládní. Třetí režim poskytuje možnosti pro úpravu časových intervalů souvisejících s dopravníkem paliva a roštováním.



Obr. 9.9: Režimy

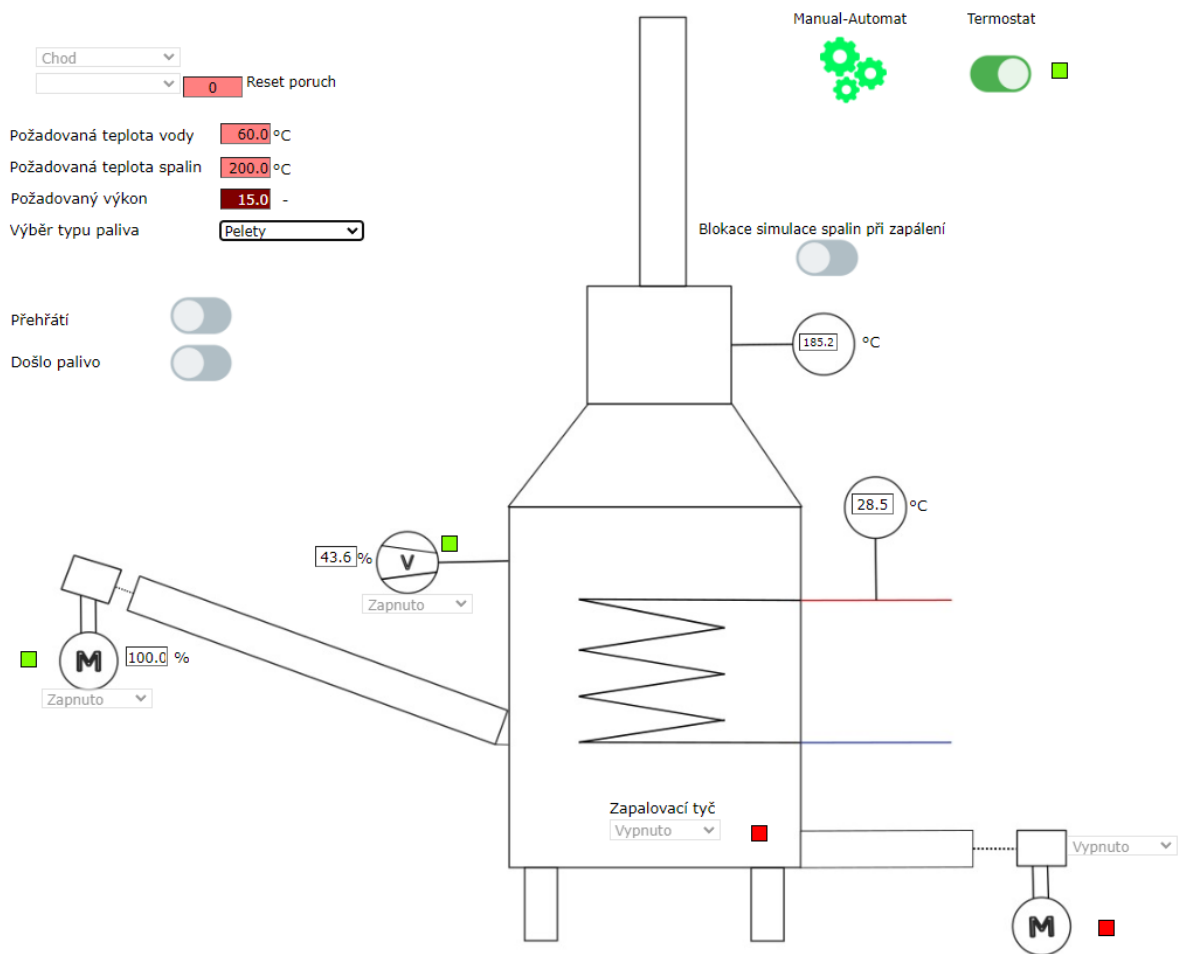
9.3.1 Automatické řízení kotle

Tato kapitola popisuje prostředí pro automatické řízení kotle, které zahrnuje simulace jednotlivých komponentů a procesů, viz obrázek 9.10. Na obrázku je zobrazena grafická reprezentace kotle a jeho jednotlivých částí, jako je motor dopravníku paliva, ventilátor pro regulaci teploty spalin, motor pro roštování popela a zapalovací tyč pro inicializaci zapalovací dávky. Výkonnost těchto prvků se zobrazuje procentuálně, zatímco zapalovací tyč pracuje ve dvou režimech: zapnuto a vypnuto. Výstupy z kotle, konkrétně teplota vody a spalin, jsou zobrazovány ve stupních Celsia.

V levém horním rohu je vidět stavový ukazatel procesu, který zobrazuje aktuální fázi simulace, jako například: čeká na start, vypíná, zapaluje, nahřívá, chod nebo porucha. Níže se nachází políčko, které zobrazuje možné chyby, jež mohou nastat během simulace, jako je chyba zapálení (F011), došlo palivo (F012) nebo přehřátí (F013).

Levý panel obsahuje také nastavení požadovaných hodnot kotle, kde uživatel může konfigurovat teplotu vody, teplotu spalin a výkon, který určuje rychlost simulace. Výchozí hodnota výkonu je nastavena na 15 (-), což je optimální rychlost umožňující detailní sledování jednotlivých prvků kotle. Dále je zde možnost výběru typu paliva, jako jsou pelety, dřevo nebo brikety. Každé palivo má odlišnou výhřevnost, což ovlivňuje chování simulace, například dřevo má nižší výhřevnost, což znamená, že dopravník paliva musí dodávat větší množství paliva a ventilátor se bude točit rychleji.

Na pravé straně nahoře jsou umístěna tlačítka pro spuštění simulace, jako je Termostat, který simuluje spuštění termostatu, a Manual-Automat, který přepíná režimy provozu. Další tlačítka, jako je Přehřátí, Blokace spalin při zapálení a Došlo palivo, simulují různé typy poruch pro testování správné funkčnosti řízení kotle.



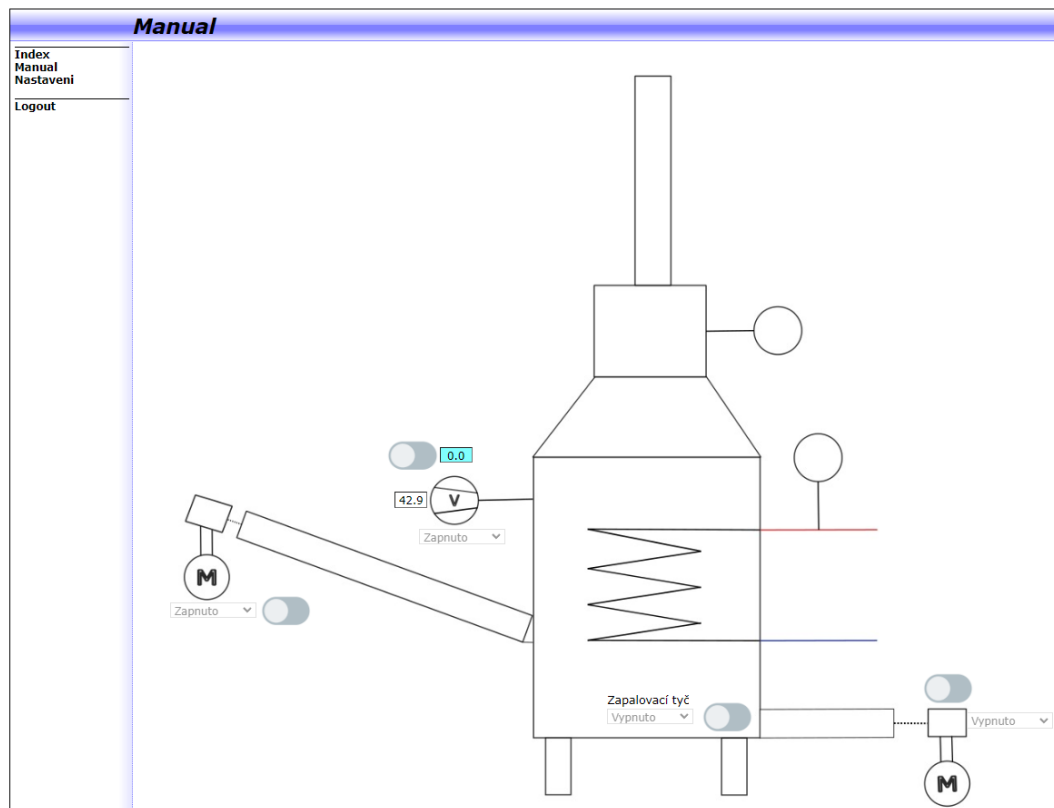
Obr. 9.10: Simulace automatického řízení kotle

9.3.2 Manuální řízení kotle

Manuální režim zobrazený na obrázku 9.11 umožňuje uživateli ručně ovládat jednotlivé části kotle. Uživatel má možnost samostatně zapínat nebo vypínat různé komponenty, například motor dopravníku, ventilátor nebo zapalovací tyč. Stav těchto komponent je zobrazen pomocí jednoduchých přepínačů a indikátorů, které signalizují, zda je zařízení aktivní, nebo kolik procent z celkového výkonu aktuálně využívá.

Uživatel může sledovat a ovlivňovat různé parametry, jako je procentuální zatížení ventilátoru a dopravníku, nebo teplotu vody a spalin v kotli. Vedle toho je k dispozici také vizuální indikace stavu zapalovací tyče, která zobrazuje, zda je aktuálně zapnutá nebo vypnutá.

Vlevo se nachází navigační panel s odkazy pro přechod mezi režimy Index, Manuál a Nastavení. V manuálním režimu má uživatel úplnou kontrolu nad jednotlivými procesy v kotli, což umožňuje přesné sledování reakce kotle na různé změny nastavení a simulace ručního řízení kotle.

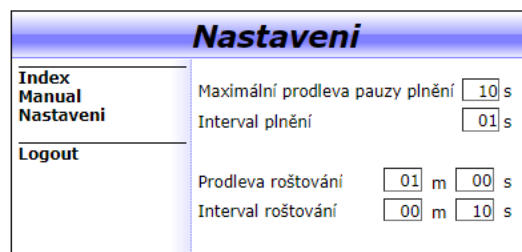


Obr. 9.11: Simulace manuálního řízení kotle

9.3.3 Nastavení časových intervalů

Obrázek 9.12 představuje prostředí nastavení simulací kotle. Zobrazené prvky umožňují uživatelům přizpůsobit různé časové intervaly a parametry související s plněním paliva a roštováním.

V prostředí lze nastavit Maximální prodleva pauzy plnění, která určuje nejdelší možnou dobu, po kterou se může zastavit plnění paliva před tím, než se dopravník znovu zapne. Interval plnění pak definuje, jak dlouho bude dopravník aktivní, aby dodal požadované množství paliva do kotle. Parametry související s roštováním fungují na stejném principu.



Obr. 9.12: Nastavení simulací kotle

Závěr

V závěru této diplomové práce je potřeba konstatovat fakt, že během jejího zpracování došlo ke změně původního zadání z implementace softwaru do reálného kotle na vytvoření simulací kotle. Tato změna umožnila vytvořit komplexní simulační prostředí, které dokáže věrně reprezentovat různé scénáře provozu kotle na tuhá paliva v reálném čase.

Vývojové prostředí Mosaic se ukázalo jako neintuitivní, například chybělo automatické formátování kódu a uživatel si musel vše formátovat sám. Další problém nastal při práci s WebMakerem kvůli omezenému počtu funkcí, což vyžadovalo hledání dalších zdrojů pro doplnění funkcionalit. Tyto nedostatky negativně ovlivnily efektivitu vývoje. Během týdenního měření a sběru dat došlo k výpadku kotle, což vyžadovalo opakování měření.

Program byl úspěšně navržen, vyvinut a otestován v simulačním prostředí a plní všechny požadované funkce. Je připraven k implementaci do reálného zařízení a parametrizaci v reálných provozních podmínkách. To umožňuje optimalizovat a efektivně řídit provoz kotle pro domácnosti a větší budovy, což povede k lepšímu využití paliva a k dosažení vyšší účinnosti systému.

Nakonec byly úspěšně vytvořeny jednotlivé scénáře simulací, které představují zapálení kotle, přechod do ustáleného chodu díky PID regulaci a různé chybové stavy, jako jsou problémy se zapálením, přehřátí kotle nebo nedostatek paliva. Tato diplomová práce ilustruje fungování a řízení systému, což umožňuje lepší pochopení jeho dynamiky a reakcí na různé provozní situace.

Simulace v diplomové práci umožňují důkladné pochopení jednotlivých funkcí kotle. To zahrnuje analýzu, jak kotel reaguje v různých provozních režimech, což je klíčové pro vývoj a úpravy řídicích systémů. Tyto simulace poskytují praktické využití nejen v průmyslovém nasazení, ale i pro studijní účely, kde mohou studenti lépe pochopit funkce PID regulátorů.

Literatura

- [1] ČSN EN 303-5:2021+A1, Heating boilers – for solid fuels, manually and automatically stoked, nominal heat output of up to 500 kW.
- [2] MATZ, Václav. Ekvitermní regulace – princip a využití v systémech regulace vytápění. Tzbinfo.
- [3] Mosaic - pro vývoj PLC programu dle standardu IEC 61131-3. Online. Programování podle IEC. 2007. Dostupné z: <<https://www.tecomat.cz/ke-staze-ni/software/mosaic/>>. [cit. 2024-05-06].
- [4] PID regulátor. Online. In: . Dostupné z: <<https://jvalter.cz/regulator-pid>>. [cit. 2024-05-07].
- [5] PLUM. MODULE ecoLAMBDA ver.2. Ver.2. 2021. Dostupné z: <<https://kipi.pl/wp-content/uploads/2021/07/User-manual-ecoLAMBDA.pdf>>
- [6] PRŮMYSLOVÉ PID REGULÁTORY: TEORIE PRO PRAXI. Online, vedoucí Miloš Schlegel. Plzeň: Fakulta aplikovaných věd ZČU v Plzni, katedra kybernetiky. Dostupné z: <<https://zcu.arcao.com/kky/zky/Prago1.pdf>>. [cit. 2024-05-07].
- [7] Processparameters: Pt1000 Temperature Sensors. Online. In: . Dostupné z: <<https://www.processparameters.co.uk/pt1000/>>. [cit. 2024-01-04]
- [8] ŘÍHA, Vladimír. Regulace malých otopných soustav s kotli. Tzbinfo.
- [9] TECO A.S. Dokumentace Foxtrot 2 - CP-2000. 2021.
- [10] VERNER A.S. Online. In: Projekční podklady pro automatické kotle VERNER A251. Sokolská 321, 549 41 Červený Kostelec, 2010, s. 26.
- [11] ZAČÍNÁME V PROSTŘEDÍ MOSAIC. Online. In: . Copyright © 1992 - 2010 Teco, s. 0-105. Dostupné z: <https://www.tecomat.cz/modules/DownloadManager/download.php?alias=txv00320_01_mosaic_progstart_cz>. [cit. 2024-05-06].

Seznam symbolů a zkratek

Zkratka	Význam
STB	Bezpečnostní omezovač teploty
STW	Bezpečnostní hlídač teploty
P	Tepelný výkon
CO	Oxid uhelnatý
OGC	Organické uhlovodíky
PM	Drobné částice pevného skupenství rozptýlené ve vzduchu
CIB	Common Installation Bus
AC	Střídavý proud
HDO	Hromadné dálkové ovládání
LTE	Long Term Evolution
USB	Universal Serial Bus
CPU	Centrální procesorová jednotka
RAM	Operační paměť
SD	Paměťová karta
OLED	Organic light-emitting diode
CO	Oxid uhelnatý
OGC	Organický plynný uhlík
PM	Suspendované částice
HMI	Human-Machine Interface
PLC	Programovatelný logický automat
ST	Strukturovaný text
IL	Instrukční list
LD	Reléová schémata

FBD	Funkční bloky
PID	Proporcionálně-Integračně-Derivační
η_K	Účinnost kotle

Seznam příloh

A Program pro řízení kotle	78
B Program pro týdenní měření	79

A Program pro řízení kotle

Soubor KotelSimulace.mpr spolu se složkou KotelSimulace tvoří kompletní řídicí program pro kotel.

B Program pro týdenní měření

Soubor Měření výkonu.mpr spolu se složkou Měření výkonu tvoří kompletní řídicí program pro kotel.