

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA ELEKTROTECHNICKÁ



DIPLOMOVÁ PRÁCE

Převodník pro připojení kotlů s řízením
OpenTherm k systémům Tecomat

Praha, 2009

Autor: Jan Malý

Prohlášení

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

V Praze dne _____

podpis

Poděkování

Rád bych poděkoval především vedoucímu diplomové práce Ing. Martinu Hlinovskému Ph.D. a kolektivu firmy Teco a.s. za cenné rady a připomínky při tvorbě této práce. Dále děkuji své rodině za podporu a zázemí po celou dobu studia.

Abstrakt

Tato diplomová práce se zabývá návrhem převodníku pro připojení kotlů využívajících komunikační protokol OpenTherm k automatům Tecomat se sběrnici TCL-2. V první části jsou představena obě komunikační rozhraní a použitý procesor ZiLOG Z8. Navazuje návrh a realizace hardware modulu a desek plošných spojů. V další části je popsán řídicí program procesoru Z8 v programovacím jazyce C. Poslední část se zabývá realizací a testováním prototypu převodníku.

Abstract

This diploma thesis is focused on the design of converter for the connection of boilers using OpenTherm communication protocol to the Tecomat PLCs with TCL-2 bus. In the first part, both communication interfaces and the ZiLOG Z8 processor are introduced. After that follows design and realization of the module and its printed circuit boards. In the next part the design of control software for the Z8 microcontroller in the C programming language is described. The last part is dedicated to realization and testing of the prototype converter module.

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Jan Malý**

Studijní program: Elektrotechnika a informatika (magisterský), strukturovaný
Obor: Kybernetika a měření, blok KM1 - Řídicí technika

Název tématu: **Převodník OpenTherm (pro připojení kotlů s řízením OpenTherm k systémům Tecomat)**

Pokyny pro vypracování:

1. Připravit studii a návrh řešení realizace převodníku tak, aby umožňoval připojení kotlů s protokolem OpenTherm k řídicím systémům Tecomat (prostřednictvím sběrnice TCL2).
2. Připravit návrh HW převodníku (mechanické řešení bude dodáno zadavatelem práce).
3. Připravit, oživit a odzkoušet rutiny pro obsluhu sběrnice OpenTherm.
4. Připravit, oživit a odzkoušet SW pro interface TCL2 (základní rutiny včetně dokumentace budou dodány zadavatelem práce).
5. Realizovat a oživit prototyp převodníku (vzniklého dle vlastního návrhu).
6. Všechny rutiny a funkce dle bodů 2 až 4 implementovat do prototypu přípravku, oživit, odladit.
7. Provést základní měření parametrů realizovaného převodníku a vypracovat stručnou verz i dokumentace pro uživatele.

Seznam odborné literatury:

Dodá vedoucí práce

Vedoucí: Ing. Martin Hlinovský, Ph.D.

Platnost zadání: do konce letního semestru 2009/10


prof. Ing. Michael Šebek, DrSc.
vedoucí katedry




doc. Ing. Boris Šimák, CSc.
děkan

V Praze dne 27. 2. 2009

Obsah

Seznam obrázků	vii
Seznam tabulek	ix
1 Úvod	1
2 Teoretický úvod	3
2.1 Komunikační protokol OpenTherm	3
2.1.1 Fyzická vrstva	3
2.1.2 Datová vrstva	5
2.1.2.1 OpenTherm Plus	5
2.1.2.2 OpenTherm Lite	6
2.1.3 Aplikační vrstva	7
2.2 Programovatelný automat Tecomat Foxtrot	9
2.2.1 Vývojové prostředí Mosaic	10
2.2.2 Sběrnice TCL-2	11
2.3 Mikrokontroler ZiLOG Z8 Encore! 64K	11
2.3.1 UART	13
2.3.2 DMA	15
2.4 Vývojové prostředí ZDS II	15
3 Hardwarový návrh modulu	17
3.1 Požadavky pro montáž	17
3.2 Návrh napájecího zdroje	19
3.3 Návrh vysílače a přijímače OpenTherm	20
3.3.1 Návrh vysílače	20
3.3.2 Návrh přijímače	22
3.4 Návrh vysílače a přijímače TCL-2	23

3.5	Návrh zapojení mikrokontroleru	24
3.6	Navržené desky plošných spojů	27
4	Návrh řídicího software modulu	31
4.1	Vysílání a příjem signálu po rozhraní OpenTherm	31
4.1.1	Vysílání signálu	32
4.1.2	Příjem signálu	33
4.2	Vysílání a příjem po sběrnici TCL-2	34
4.3	Formáty datových zpráv	35
4.3.1	Formát zpráv v módu OpenTherm Plus	35
4.3.2	Formát zpráv v módu OpenTherm Lite	36
4.4	Struktura navrženého řídicího programu	36
4.4.1	Kontrola funkčnosti komunikace TCL-2 a ošetření stavových zpráv	37
4.4.2	Obsluha výstupní datové oblasti TCL-2	37
4.4.3	Obsluha vstupní datové oblasti TCL-2	38
4.4.4	Požadavek na výměnu firmware	38
4.4.5	Odbavení požadavků na zařízení OpenTherm	39
4.5	Nastavení modulu v prostředí Mosaic	39
5	Realizovaný modul	43
5.1	Testování modulu	44
5.2	Fotodokumentace modulu	50
6	Závěr	53
	Literatura	55
A	Zprávy aplikační vrstvy OpenTherm	I
A.1	Kompletní popis zpráv protokolu OpenTherm	I
A.2	Testovací program modulu v prostředí Mosaic	VIII
B	Obsah přiloženého CD	XI

Seznam obrázků

2.1	Bity v kódování Manchester	4
2.2	Posloupnost několika bitů v kódování Manchester	5
2.3	Časový rozvrh komunikace	6
2.4	Programovatelný automat Tecomat Foxtrot	10
2.5	Blokové schéma procesoru ZiLOG Z8 Encore!	13
2.6	Blokové schéma kanálu UART	14
3.1	Rozměry použité krabičky 175/605-2	18
3.2	Rozmístění jednotlivých prvků na čelním panelu modulu	18
3.3	Rozmístění desek plošných spojů v krabičce	19
3.4	Schéma zapojení napájecího zdroje	20
3.5	Schéma zapojení vysílače a přijímače OpenTherm	23
3.6	Schéma zapojení vysílače/přijímače TCL-2	24
3.7	Schéma zapojení mikroprocesoru	26
3.8	Návrh desek plošných spojů - deska napájecího zdroje	27
3.9	Návrh desek plošných spojů - deska komunikačních rozhraní	28
3.10	Návrh desek plošných spojů - deska mikrokontroleru	29
5.1	Přípravek pro testování OpenTherm komunikace	44
5.2	Přípravek pro testování OpenTherm komunikace	45
5.3	Vyslaná zpráva OpenTherm[Modrá - výstup regulátoru, červená - úroveň na lince]	46
5.4	Vyslaná zpráva OpenTherm - detail	47
5.5	Přijatá zpráva OpenTherm	47
5.6	Přijatá zpráva OpenTherm - detail	48
5.7	Celková perioda komunikace[modrá - vysílání, zelená - příjem]	48
5.8	Modul v krabičce 175/605-2	50
5.9	Modul v krabičce - čelní panel	50

5.10	Prototyp modulu - rozmístění desek plošných spojů	51
5.11	Prototyp modulu - pohled zboku	51
5.12	Prototyp modulu - celkový pohled	52
5.13	Prototyp modulu - pohled zboku v krabici	52

Seznam tabulek

2.1	Logické úrovně protokolu OpenTherm	4
2.2	Typy zpráv	5
2.3	Rámec zprávy v protokolu OpenTherm	6
2.4	Minimální množina zpráv protokolu OpenTherm	8
3.1	Použité piny procesoru a jejich význam v aplikaci	25
4.1	Rámec datové zprávy mezi modulem a PLC	35
A.1	Třída 1 - Řídící zprávy a informace o stavu	I
A.2	Třída 2 - Informace o konfiguraci	III
A.3	Třída 3 - Dálkové ovládání	IV
A.4	Třída 4 - Informační a senzorová data	IV
A.5	Třída 5 - Specifické parametry kotle	VI
A.6	Třída 6 - Transparentní parametry kotle	VII
A.7	Třída 7 - Informace o historii chyb	VII
A.8	Třída 8 - Řízení speciálních aplikací	VIII

Kapitola 1

Úvod

S neustále rostoucími cenami energií vzniká stále větší tlak na úspornost provozu vytápění budov. Proto se vyvíjí stále nové metody, které kombinují různé principy vytápění (podlahové, radiátorové), zdroje tepla (kotle, tepelná čerpadla, solární ohřev) a způsoby řízení (intermní, ekvitermní regulace). Existuje mnoho přístupů k regulaci vytápění, od nejjednodušších prostorových termostatů pracujících na principu on/off, přes pokročilé způsoby regulace zdroje tepla po individuální řízení teploty v jednotlivých místnostech pomocí termoregulačních hlavic či servoventilů a kombinace všech těchto principů.

Díky požadavkům na sledování stále většího množství parametrů a řízení stále většího množství zařízení se začínají aplikace v odvětví managementu menších budov i bytových a rodinných domů blížit průmyslovým, přestávají stačit dřívější řešení a je třeba použít univerzálnější prostředky řízení, např. PLC nebo průmyslová PC. Ty umožňují připojení velkého množství vstupů a výstupů různých typů, svou podstatou nejsou omezeny jen na ovládání vytápění, čímž umožňují řídit i další oblasti správy budov jako například zabezpečení, tím integrovat různé systémy objektu a umožnit jejich snadnou interakci. Dále poskytují široké možnosti vizualizace, ovládání a komunikace s uživatelem ať již pomocí operátorských panelů, nebo po nejrozličnějších sítích, ať již rádiových, průmyslových, nebo internetu. Pojem *Inteligentní budova* je poté zcela na místě.

Aby bylo dosaženo opravdu optimálního řízení, je třeba sběr velkého množství různých dat, čímž je v inteligentních budovách kladen velký důraz na komunikaci. V posledních letech proto vzniklo několik nových komunikačních protokolů, které mají ambice standardizovat komunikaci komponent do inteligentních budov. Typickými zástupci jsou sběrnice pro inteligentní elektroinstalaci KNX/EIB, LON, CIB a další, dále například

specializovaný protokol komunikaci s kotli OpenTherm, kterému se budeme dále věnovat.

Tato práce se zabývá návrhem a realizací modulu, který umožní připojení kotlů komunikujících prostřednictvím protokolu OpenTherm k systému Tecomat Foxtrot, který je vhodný nejen na velké a průmyslové aplikace, ale i na úkoly domovní automatizace. Vyvinutý modul umožní řízení široké škály zařízení podporující tento protokol v celém rozsahu jeho definice a bude sloužit jako tlumočnick mezi rozhraním TCL-2, které je používáno automaty firmy Teco a.s., a protokolem OpenTherm. Spolu s možnostmi inteligentní elektroinstalace tak pokryje velké množství aplikací vyžadovaných v moderních stavbách.

Kapitola 2

Teoretický úvod

2.1 Komunikační protokol OpenTherm

Asociace OpenTherm vznikla v roce 1996 a první verzi specifikace protokolu odkoupila od firmy Honeywell za symbolickou 1£. Nyní sdružuje přes 40 členů včetně firem Honeywell a Siemens. Protokol je určen hlavně ke standardizaci komunikace s novými generacemi kotlů, dále přispívá k efektivnějšímu řízení, snadnému sběru dat a univerzálnosti při propojení zařízení od více výrobců. Protokol OpenTherm pracuje se třemi vrstvami ISO/OSI modelu. Fyzickou, datovou a aplikační. V datové vrstvě definuje dva typy komunikace. Kompletní specifikaci, označovanou jako OpenTherm Plus a verzi základní, označovanou jako OpenTherm Lite. Všechny přístroje se značkou OpenTherm by měly zvládat oba typy komunikace. Protokol se nezabývá podrobnostmi řízení kotle, regulace probíhá tak, že řídicí jednotka předá kotli údaj o požadované teplotě topné vody a již se neúčastní toho, jak kotel teploty dosáhne. Kotel zase nemá žádné informace o tom, jak řídicí jednotka údaj o teplotě určila. Následující popis vychází ze specifikace verze 2.2 dodané firmou Tecno a.s..

2.1.1 Fyzická vrstva

Protokol je založen na principu point-to-point komunikace řídicí jednotky (termostat, PLC, počítač) s kotlem, kdy řídicí jednotka vystupuje v komunikaci jako master a kotel jako slave. Jako médium je použita nekroucená dvoulinka, při nasazení v prostorech s elektromagnetickým rušením je doporučeno použití dvoulinky kroucené. Řídicí jednotka připojená na kotel nemusí mít žádné další napájení, může být napájena přímo po komu-

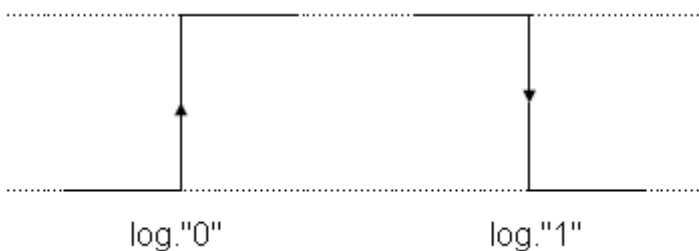
nikačních vodičích, poté na ni jsou ale kladeny požadavky v podobě napájecího napětí rovnému hodnotě při log."0" a nejvyššímu odběru proudu 5 mA.

Při samotné komunikaci řídicí jednotka vysílá tak, že mění napětí na komunikačních vodičích, kotel komunikuje změnou proudu procházejícím vodiči při úrovni napětí log."0". Příslušné logické úrovně jsou uvedeny v tabulce 2.1.

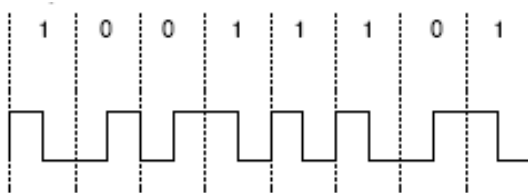
Tabulka 2.1: Logické úrovně protokolu OpenTherm

	log."0"	log."1"
master	max.7 V	15 - 18 V
slave	5 - 9 mA	17 - 23 mA

Protokol používá kódování Manchester, kdy každý bit je vyjádřen přechodem buď z log."0" do log."1" pro log."0", nebo přechodem z log."1" do log."0" pro log."1", viz. obrázek 2.1. Posloupnost několika bitů poté vypadá následovně, 2.2. Výhodou tohoto kódování je automatická synchronizace díky tomu, že uprostřed každého bitu musí být přechod mezi úrovněmi. To také usnadňuje odhalení chyby, která se projeví chybějícím přechodem. Perioda přenosu je 1 ms, okno pro změnu úrovně od začátku přenosu bitu je $500 \mu\text{s} -10 \% +15 \%$. Změna úrovně musí být provedena vždy v max. $50 \mu\text{s}$.



Obrázek 2.1: Bity v kódování Manchester



Obrázek 2.2: Posloupnost několika bitů v kódování Manchester

2.1.2 Datová vrstva

V datové vrstvě rozdělujeme OpenTherm na dvě části podle "intelligence" komunikujících zařízení. Lite verze protokolu je jednoduchá komunikace pomocí PWM signálu, proto je vhodná pro jednodušší aplikace. OpenTherm Plus zaručuje plnou funkčnost protokolu. Zjištění, který protokol se má použít, probíhá tak, že řídicí jednotka posílá rámce komunikace OpenTherm Plus a pokud kotel po dobu 20 s neodpovídá, přejde se na komunikaci v OpenTherm Lite.

2.1.2.1 OpenTherm Plus

Jednotlivé zprávy se skládají ze start bitu na začátku zprávy a stop bitu na jejím konci, přičemž oba mají hodnotu log."1" a obalují 32-bitový rámec složený z následujících částí:

1. Parita[1 bit] - hodnota bitu je taková, aby součet jedniček v rámci byl sudý.
2. Typ zprávy[4 bity] - Rozlišujeme 8 typů zpráv:

Tabulka 2.2: Typy zpráv

master		slave	
Kód zprávy	Význam	Kód zprávy	Význam
000	Čtení	100	Čtení potvrzeno
001	Zápis	101	Zápis potvrzen
010	Neplatná data	110	Neplatná data
011	Rezervováno	111	Neznámé Data ID

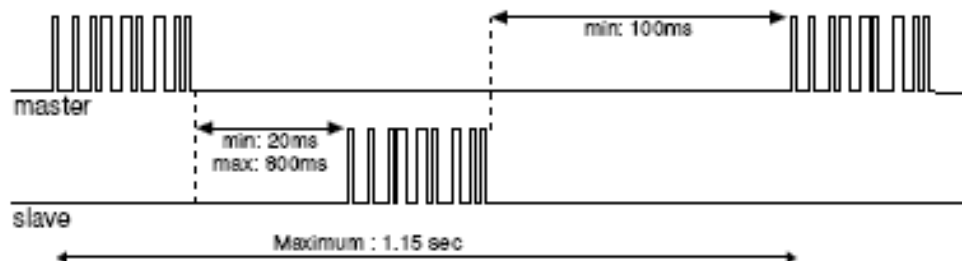
3. Volné bity[4].

4. Data ID[8 bitů] - Identifikátor dat z aplikační vrstvy.
5. Data[16 bitů] - Data související s identifikátorem(vyžádaná dotazem, určená k zápisu, nebo upřesňující při dotazu).

Tabulka 2.3: Rámec zprávy v protokolu OpenTherm

Parita[1 bit]	Typ zprávy[3 bity]	Volné bity[4bity]	Data ID[8 bitů]	Data[16 bitů]
---------------	--------------------	-------------------	-----------------	---------------

Komunikace probíhá tak, že master zašle požadavek a slave v daném časovém intervalu musí odpovědět. Master poté minimálně 100 ms čeká před dalším dotazem. Master by měl dotazovat nejméně každou 1 s, delší prodleva je brána jako chyba komunikace. Ukázka komunikace je na obrázku 2.3. V případě chyby je rámec označen za neplatný a komunikace pokračuje dalším rámcem, opravy nejsou podporovány.



Obrázek 2.3: Časový rozvrh komunikace

2.1.2.2 OpenTherm Lite

OpenTherm Lite je verze protokolu pro komunikaci s verzemi kotlů nepodporujícími rozhraní OpenTherm Plus. Je realizováno pomocí PWM signálu o periodě od 2 ms do 10 ms. Střída poté vyjádřuje procentuální teplotu topné vody v rozsahu, který je kotlem definován. Závislost střídy na teplotě je lineární. Pokud je střída menší než 5% je topení vypnuto. Kotel může signalizovat chybu tak, že proud z režimu log."0" přepne do režimu log."1".

2.1.3 Aplikační vrstva

Aplikační vrstva definuje celkem 256 druhů zpráv. Prvních 128(0-127) je určených protokolem, druhých 128 je volných pro členy asociace pro účely testování a diagnostiky. Budeme se tedy věnovat jen prvním 128 zprávám. Ty jsou rozděleny do 8 tříd:

- Třída 1 - Řídící zprávy a informace o stavu - Důležité kontrolní a informační ukazatele o provozu kotle. Obsahuje zprávu nastavení teploty topné vody ve °C. Dále chybový bit a nastavení základních parametrů kotle, jako zapnutí vytápění, chlazení, ohřevu topné vody apod.
- Třída 2 - Informace o konfiguraci - Informace o podporovaných funkcích kotle, verzi použitého protokolu, ID výrobce v asociaci, pokud je členem a verzi zařízení.
- Třída 3 - Dálkové ovládání - Reset chyby blokování, příkaz pro napouštění topné vody do soustavy, ostatní funkce rezervovány.
- Třída 4 - Informační a senzorová data - Nastavení data a času, informace o teplotách a tlacích v soustavě, dále počítadla zážehů a sepnutí čerpadel, celkový čas hoření plamene a běhu čerpadel.
- Třída 5 - Specifické parametry kotle - nastavení parametrů kotle definovaných výrobcem, např. maximální teploty topné vody a užitkové vody. Mohou, ale nemusí být přenastavitelné.
- Třída 6 - Transparentní parametry kotle(z hlediska master) - parametry, které nejsou definovány protokolem, tudíž jsou pro master transparentní, protože neví nic o jejich významu v aplikaci
- Třída 7 - Informace o historii chyb - zaznamenávají historii chyb kotle, umožňují čtení z chybového bufferu kotle.
- Třída 8 - Řízení speciálních aplikací - různé speciální aplikace, například ovládání chlazení, detekce při změně priority řízení teploty topné vody(manuálním zásahem nebo programem kotle).

Nyní zmíníme několik základních zpráv protokolu, které by mělo podporovat každé zařízení OpenTherm.

Tabulka 2.4: Minimální množina zpráv protokolu OpenTherm

Třída	Data ID	Typ zprávy	Význam zprávy	Popis významu bitů[0/1]
1	0	Čtení	HB: Master status	0:Vytápění[zakázáno/povoleno] 1:TUV[zakázáno/povoleno] 2:Chlazení[zakázáno/povoleno] 3:Kompenzace venkovní teploty [zakázáno/povoleno] 4:Vytápění 2.okruh [zakázáno/povoleno] 5-7:rezervováno
			LB: Slave status	0:Indikace chyby[ne/ano] 1:Vytápění[neaktivní/aktivní] 2:TUV[neaktivní/aktivní] 3:Stav plamene[neaktivní/aktivní] 4:Chlazení[neaktivní/aktivní] 5:Vytápění 2.okruh [neaktivní/aktivní] 6:Žádost o diagnostiku[ne/ano] 7:rezervováno
1	1	Zápis	Teplota topné vody	Číslo f8.8 0...100
2	3	Čtení	HB: Konfigurace slave	0:TUV přítomno[ne/ano] 1:Řízení[modulace/spínání] 2:Chlazení podporováno[ne/ano] 3:TUV konfigurace [nespecifikováno/nádrž] 4:Master Low-off&řízení čerpadla [povoleno/nepovoleno] 5:Vytápění 2.okruhu přítomno [ne/ano]
			LB:Členské ID výrobce	Číslo u8
Pokračování na další straně				

Pokračování z předchozí strany				
Třída	Data ID	Typ zprávy	Význam zprávy	Popis významu bitů[0/1]
8	14	Čtení	Maximální hodnota relativní modulace	Číslo f8.8 0...100
4	17	Čtení	Aktuální hodnota relativní modulace	Číslo f8.8 0...100
4	25	Čtení	Teplota topné vody z kotle	Číslo f8.8 -40...127

Zkratky a označení v tabulce:

- HB - horních 8 bitů dat.
- LB - dolních 8 bitů dat.
- TUV - teplá užitková voda.
- u8 - formát celého 8-bitového čísla bez znaménka(unsigned int8)
- f8.8 - formát 16-bitového čísla s pevnou řádovou čárkou(16 bit signed fixed point value), nejvyšší bit je znaménko, následuje 7 bitů celého čísla a 8 bitů čitatele zlomku /255. Nejnižší bit je tedy $\frac{1}{256}$.

Kompletní popis všech zpráv lze nalézt v příloze A.

2.2 Programovatelný automat Tecomat Foxtrot

Tecomat Foxtrot je malý modulární řídicí systém nové generace. Jeho základem je kompaktní PLC Foxtrot, které nabízí velké množství připojitelných periférií. Pomocí sběrnice TCL-2 je tedy možné tvořit systémy až o deseti rozšiřujících modulech. Dále je možné jednotlivá PLC propojovat mezi sebou nebo s PC prostřednictvím sítě Ethernet nebo EPSNET. Další typy rozhraní jsou podporovány volitelně. Taková koncepce umožňuje vytváření rozsáhlých distribuovaných systémů řízení. Konstrukčně je celý systém modulů i samotné PLC řešeno pro montáž do lišty rozvodných skříní, samotné

PLC má šířku 6 standartních jističů velikosti M, šířka modulů je 3M.

PLC Foxtrot se skládá z několika částí. Hlavním CPU je 32-bitový RISC procesor Freescale Coldfire s frekvencí 166 MHz a s dobou cyklu 0,2 ms na 1000 instrukcí. Dále je jednotka vybavena dvěma seriovými kanály, rozhraním Ethernet, SD/MMC kartou jako přídatnou pamětí, např. pro archivaci dat, a sběrnici TCL-2 pomocí které komunikuje s periferními moduly. Samotná jednotka podle typu obsahuje různé množství digitálních, případně analogových vstupů/výstupů. Další částí je procesor pro komunikaci se sběrnici CIB používanou k připojování inteligentní elektroinstalace Inels. Poslední volitelnou součástí podle typu jednotky je displej se šesti tlačítky ovládaný také vlastním procesorem.



Obrázek 2.4: Programovatelný automat Tecomat Foxtrot

2.2.1 Vývojové prostředí Mosaic

Programování PLC Foxtrot probíhá přes vývojové prostředí Mosaic. Jedná se o univerzální prostředí pro všechny typy PLC Tecomat. Umožňuje programování v jazyce instrukcí (mnemokód), systémy s 32-bitovými procesory (TECOMAT TC650, TC700 a Foxtrot) lze programovat také v jazycích podle IEC EN 61131-3 (IL - jazyk seznamu instrukcí, ST - jazyk strukturovaného textu, LD - příčkové diagramy, FBD - diagramy funkčních bloků). Dále obsahuje řadu nástrojů pro vývoj, jako například nástroje pro vytváření a simulaci operátorských panelů, tvorbu a ladění PID regulátorů, a pro ladění, jako například inspektor proměnných, nástroj pro záznam vývoje proměnných do grafů nebo kompletní simulátor, který umožňuje vyvíjet a testovat aplikaci na PC včetně vizualizace. Aplikace u systémů Tecomat TC650, TC700 a Foxtrot umí updatovat program v PLC bez nutnosti zastavit řízení technologie a umožňuje komunikovat s řídicím systémem

přes sériovou linku, Ethernet, či USB. V prostředí je zahrnuta i podpora pro vytáčené připojení přes telefonní nebo GSM modem a v poslední době oblíbené spojení přes Wi-Fi, čímž umožňuje dálkovou správu. Aplikace je k dispozici zcela zdarma na internetových stránkách firmy Teco a.s..

2.2.2 Sběrnice TCL-2

Sběrnice TCL-2 slouží ke spojení centrální jednotky PLC Foxtrot s jeho periferními moduly. Používá metalické vedení a fyzicky odpovídá standardu RS-485. Umožňuje připojit až 10 modulů sběrnicovou topologií. Ta se nastavuje v rozsahu 0-9 potenciometrem na horním panelu zařízení. Komunikační rychlost je 345,6 kBaud. Komunikační služby umožňují výměnu dat, inicializaci, testování a změnu stavu připojených modulů. Ke kontrole správnosti dat se využívá CRC součtu.

2.3 Mikrokontroler ZiLOG Z8 Encore! 64K

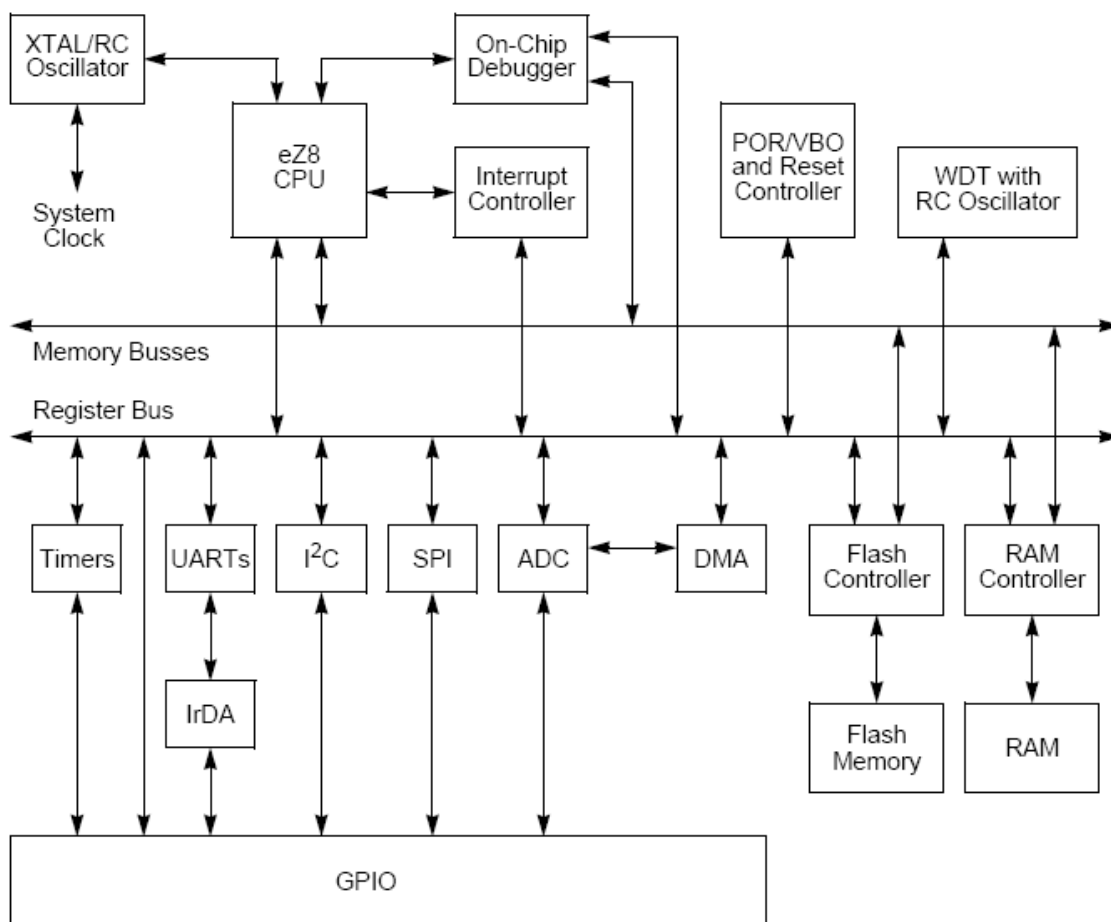
Osmibitové mikrořadiče Z8 Encore, nabízené firmou ZiLOG od roku 2003, jsou založeny na modernizovaném procesoru Z8, na čipu mají až 64KB paměti Flash, až 4kB paměti RAM, všechny obvyklé periferie (čítače, sériové komunikační obvody, paralelní vstupně / výstupní porty), oscilátor, 10-bitový AD převodník, 16-bitový časovač Watch-Dog.

Jádrem všech mikrořadičů Z8 je 8-bitový procesor s architekturou Harvard bez střadače, vykonávající všechny instrukce programu přímo nad registry RAM. Instrukční soubor umožňuje lineární adresování paměti RAM ve 4-bitovém, 8-bitovém a 12-bitovém adresním módu. 16-bitový čítač programu adresuje lineárně až 64kB paměti Flash. Pro snadný vývoj aplikací jsou tyto mikrokontrolery vybaveny hardwarovým debuggerem na čipu a sériovým rozhraním ZDI (ZiLOG Developer Interface).

Základní vlastnosti mikrokontroleru:

- Speciální funkce MCU:
 - Lineární adresování paměti Flash i paměti RAM.
 - Všechny operace nad registry RAM bez střadače.

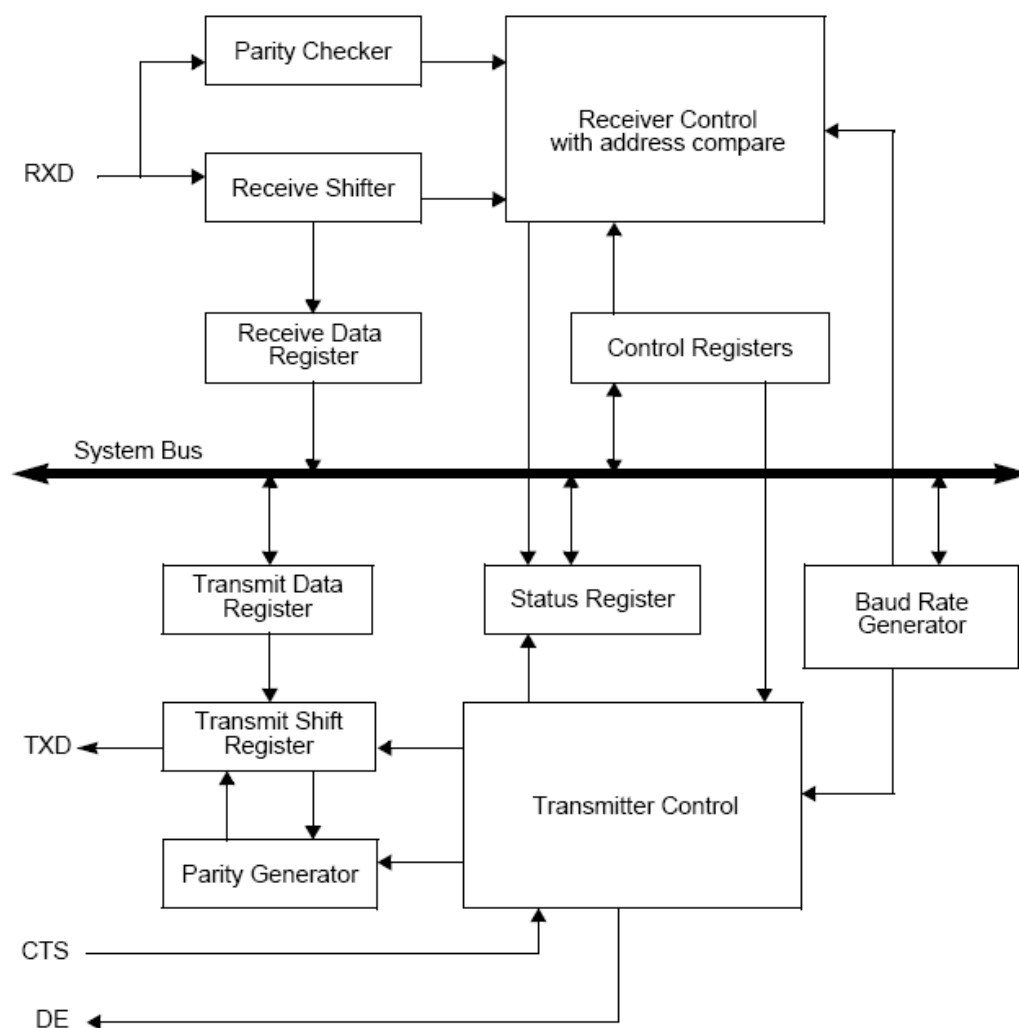
- Vstupy slučitelné s 5 V logikou.
 - 24 vektorů přerušení ve třech programovatelných prioritách.
 - Úsporné režimy HALT, STOP.
 - Programovatelný 16-bitový Watch-Dog časovač.
 - Sériové programování (ICSP) arduwarový debugger na čipu.
 - Napájecí napětí 3,0 - 3,6 V.
 - Taktovací kmitočet až 20MHz.
-
- Periferie:
 - 4 šestnáctibitové čítače (CCP).
 - Až 12-kanálový 10-bitový Sigma-Delta AD převodník.
 - 2 kanály UART, rozhraní SPI a I2C.
 - 3 kanály DMA (jeden pro AD převodník).



Obrázek 2.5: Blokové schéma procesoru ZiLOG Z8 Encore!

2.3.1 UART

V aplikaci se využívá ke komunikaci po sběrnici TCL-2 kanál UART1. Procesor Z8 Encore má k dispozici dva fullduplexní seriové kanály, schopné asynchronní komunikace. Podporují zprávy o délce 8 a 9 bitů, volitelnou paritu, jeden nebo dva stopbity ukončující zprávu, oddělené přerušení pro vysílání a příjem, signál Driver Enable, který ušetří čas procesoru při využití externího vysílače a nastavitelný generátor hodinového signálu pro vysílání a příjem dat. UART dále automaticky signalizuje pokus o přepsání validních(nezpracovaných dat) nebo přerušení spojení nastavením příslušných bitů stavovém byte.



Obrázek 2.6: Blokové schéma kanálu UART

ZiLOG Z8 umožňuje provozovat UART v tzv. multiprocesorovém režimu. V tomto režimu se vždy jako první byte posílá adresa cílového zařízení, přičemž tato zpráva má 9 bitů, kdy devátý bit nahrazuje paritu (poslední před stop bitem) a značí, že jde o adresní byte. Přijatou adresu lze poté automaticky porovnat s adresou zařízení uloženou v registru a tím ušetřit čas procesoru při komparaci adresy. Možné jsou tři režimy funkce, buď je vyvoláno přerušení při každém příjmu adresy a datových bytů, při příjmu korektní adresy a datových bytů, nebo jen při příjmu korektně adresovaných datových bytů, což je nejúspornější možnost.

2.3.2 DMA

DMA je označení funkce zvané přímý přístup do paměti. Tato funkce umožňuje automatický přesun dat mezi registry procesoru a perifériemi bez angažování procesoru. Z8 Encore obsahuje tři DMA kanály, přičemž první dva slouží k obousměrné obsluze periférií a třetí k obsluze AC převodníku. DMA operuje ve 4 krocích. Nejprve je registrován požadavek na DMA přesun dat, následně si DMA vyžádá přístup na adresní a datovou sběrnici procesoru, po umožnění přístupu vyšle jeden byte nebo word a vrátí řízení sběrnice zpět procesoru. Poté se porovná aktuální adresa s adresou koncovou, která je uložena v kontrolním registru používaného DMA a pokud souhlasí, je aktuální adresa nastavena na startovní hodnotu, pokud je povoleno, je vygenerováno přerušení, a přenos je ukončen. Buď se poté čeká na další spuštění, nebo je vynulován příznak povolující DMA(DEN) a DMA je vypnuto. Pokud adresa nesouhlasí, je inkrementována a pokračuje se další žádostí o přístup na sběrnice. V naší aplikaci bude DMA využito k naplnění vysílacího bufferu UART při komunikaci po sběrnici TCL-2.

2.4 Vývojové prostředí ZDS II

Integrované vývojové prostředí ZDS II slouží k vývoji a ladění aplikací pro všechny mikrokontrolery ZiLOG, zejména pak modely vybavené flash pamětí, tedy Z8 Encore, Z80 Acclaim a ZNEO. Distribuováno je zdarma na internetových stránkách firmy ZiLOG. Prostředí se skládá z několika součástí:

- Projektový manažer pro práci se všemi složkami projektu. Zobrazuje všechny zdrojové soubory projektu a veškeré externí a hlavičkové soubory, na které je z nich odkazováno.
- Textový editor pro psaní zdrojových souborů v assembleru nebo jazyce C. Editor má standardní vlastnosti pro tabelaci programových řádků a barevné odlišení položek programu.
- Assembler v jedné ze tří dostupných verzí (Z8Encore!, eZ80Acclaim! a ZNEO)
- Linker pro vytvoření spustitelného kódu, který obsahuje všechny informace pro ladění programu na úrovni zdrojového souboru.

- ANSI C překladač v jedné ze tří dostupných verzí (Z8Encore!, eZ80Acclaim! a ZNEO).
- Simulátor pro test programu bez mikrořadiče.
- Debugger pro ladění programu uloženého v paměti Flash

Mikrořadiče ZiLOG obsahují na čipu hardwarový debugger, se kterým komunikuje vývojové prostředí ZDS II po dvoudrátovém rozhraní ZDI (ZiLOG Developer Interface). Ladění probíhá na úrovni zdrojového souboru (v assembleru nebo v jazyce C), umožňuje manuální krokování programu, automatické krokování, tj. běh programu s rychlostí cca 1 instrukce za sekundu, RESET a skok na zvolenou část programu nebo na zarážku (breakpoint). Zarážek je možné vložit neomezený počet. Aktuální pozice v programu je označena kurzorem.

Na pracovní ploše je možné otevřít několik pomocných oken a zobrazit v nich obsah všech registrů, které program používá. Jde zejména o řídicí registry mikrokontroleru, také registry čítačů, časovačů a periférií. Dále je možné prohlížet stav lokálních i zvolených globálních proměnných v každém kroku programu a celý obsah paměti. Obsah všech oken je aktualizován po každém kroku programu, změna hodnot registrů je vyznačena červenou barvou. Jednou z dalších možností je schopnost zobrazení aktuálně vykonávaného kódu paralelně jak v jazyce C, tak v assembleru pomocí okna Disassembly.

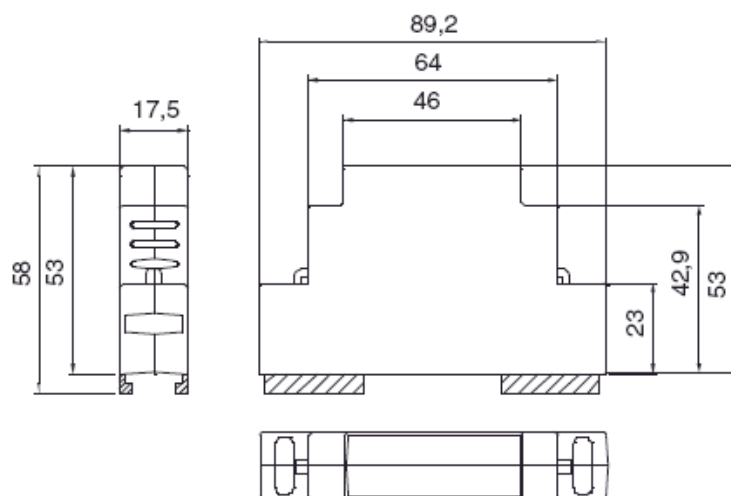
Kapitola 3

Hardwarový návrh modulu

Hardwarový návrh řešení modulu musí vycházet z několika základních požadavků. Celý modul je navržen do krabičky 175/605-2 určené pro montáž na M36 DIN lištu. Má vlastní 24 V napájení, které je vedeno spolu se sběrnici TCL-2 z PLC, není tedy využita možnost napájení řídicí jednotky kotlem, kterou zařízení protokolu OpenTherm podporují. Modul realizuje převod komunikace mezi protokolem OpenTherm a rozhraním TCL-2. Jeho částmi jsou tedy kromě nezbytných zdrojů napětí 5 a 3,3 V vysílač a přijímač signálů rozhraní OpenTherm, taktéž ze sběrnice TCL-2, a 8-bitový mikrokontroler ZiLOG Z8 Encore!, který bude aplikaci řídit.

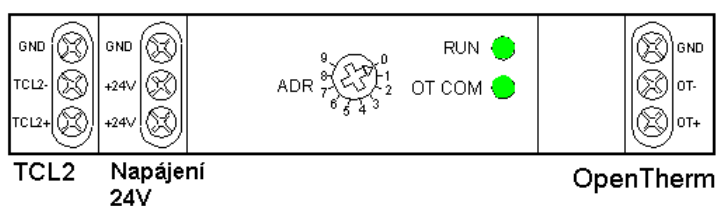
3.1 Požadavky pro montáž

Jak již bylo řečeno výše, modul je navržen k montáži na M36 DIN lištu. Dále obsahuje svorky pro připojení napájení 24 V, svorky pro připojení k zařízení OpenTherm a svorky pro připojení ke sběrnici TCL-2. Protože obě rozhraní jsou fyzicky realizována jako 2-vodičová, postačí nám třívodičové svorky. Poté našim požadavkům vyhovuje krabička 175/605-2, která je zobrazena na obrázku 5.9.



Obrázek 3.1: Rozměry použité krabičky 175/605-2

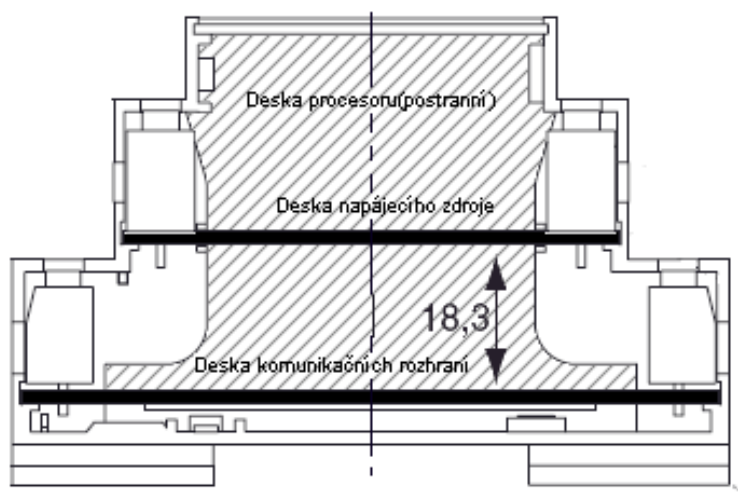
Další požadavky jsou dané standardizací modulu vzhledem k řadě již existujících periférií PLC Foxtrot. Na horním panelu modulu je vždy umístěn potenciometr umožňující krokové nastavení adresy modulu na sběrnici TCL-2 a dvě LED diody pro signalizaci stavu zařízení a sledování komunikace. V tomto případě bude druhá LED dioda signalizovat komunikaci se zařízením OpenTherm. Rozmístění svorek a prvků na čelním panelu je vyobrazeno ve schématu 3.2.



Obrázek 3.2: Rozmístění jednotlivých prvků na čelním panelu modulu

Kvůli rozměrům a pozici otvorů pro umístění svorek je nutné rozdělit návrh do tří částí. První částí je realizace zdroje napájení měničem z 24 V na 5 V. Druhou částí budou vysílače a přijímače pro komunikaci v obou rozhraních a poslední částí bude jednotka s mikrokontrolerem, stabilizátorem 5 V na 3,3 V a prvky čelního panelu. Tomuto rozvržení bude odpovídat i fyzické rozmístění součástek na deskách plošných spojů. Umístění jednotlivých desek v krabičce dokumentuje obrázek 3.3. Propojení desek stejně jako jejich

fixace v krabici bude realizováno pomocí pájivých plošek. Kvůli rozměrům desek jsou všechny součástky v pouzdrech pro povrchovou montáž SMD.



Obrázek 3.3: Rozmístění desek plošných spojů v krabici

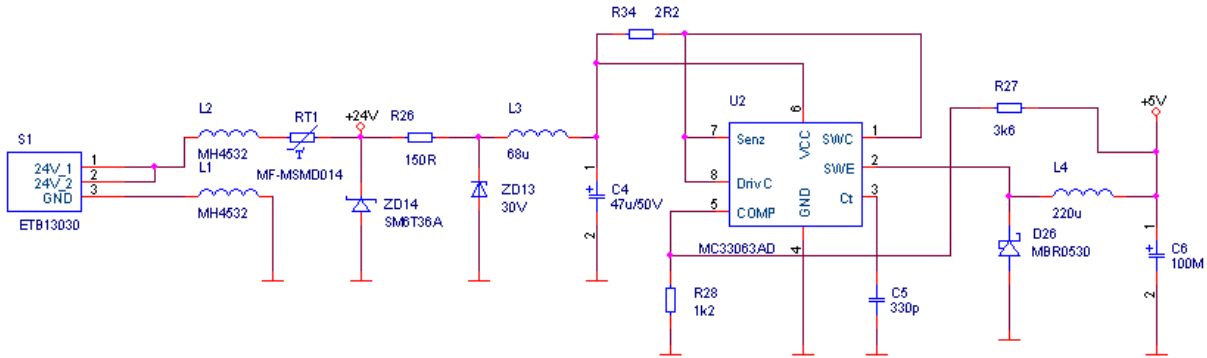
3.2 Návrh napájecího zdroje

Jádrum napájecího zdroje je řídicí obvod MC33063. Tento obvod umí pracovat v režimu step-up, step-down, nebo jako invertující měnič. Nabízí rozsah napětí 3 - 40 V, výstupní proud až 1,5 A, spínací frekvenci 100kHz, omezovač proudu, nízkou spotřebu v klidovém režimu a napětí interní reference je v rozsahu %2.

V naší aplikaci bude využit v typovém zapojení pro step-down DC-DC měnič. Samotný obvod se skládá z interní reference 1,25 V, komparátoru, řízeného oscilátoru pracovního cyklu s omezovačem proudu a výstupního proudového spínače s tranzistorem pro jeho ovládání. Obvod v zapojení step-down DC-DC měnič komparuje výstupní napětí s interní referencí, podle výsledku spíná/rozpíná výstupní napětí, přičemž kontroluje i velikost protékajícího proudu. Pro nastavení požadovaného napětí se umísťuje na vstup komparátoru napěťový dělič. Periodu oscilátoru lze měnit hodnotou kondenzátoru připojeného na pin CT.

Na vstup obvodu je připojen transil jako přepěťová ochrana a Zenerova dioda

ke stabilizaci vstupního napětí, dále tlumivky k vyhlazení proudu. Na výstup se zapojuje také tlumivka, proudový usměrňovač a elektrolytický kondenzátor k vyhlazení výstupního napětí a proudu. Přesné hodnoty použitých součástek a jejich zapojení je vidět v následujícím schématu 3.4.



Obrázek 3.4: Schéma zapojení napájecího zdroje

3.3 Návrh vysílače a přijímače OpenTherm

Požadavky na tuto část jsou specifikovány v Kapitole 1 v části věnované protokolu OpenTherm. Důležité jsou především parametry týkající se logických úrovní napětí u vysílače a úrovní proudu pro část přijímače, které jsou uvedeny zde 2.1. Řízení vysílače a snímání výstupu přijímače bude zajištěno mikrokontrolerem. Pro zajištění požadavku protokolu na možnost zapojení vodičů nezávisle na polaritě umístíme za vstupní svorky diodový můstek.

3.3.1 Návrh vysílače

Základní požadovanou vlastností vysílače má být schopnost měnit velikost napětí dodávaného připojeným zařízením. Hlavní částí vysílače je tedy říditelný napěťový regulátor pracující v rozsahu napětí 5 - 24 V. Přechod mezi úrovněmi musí být dost rychlý, aby neovlivnil komunikaci. Musí být tedy uskutečněn do 50 μ s. Tyto požadavky splňuje například regulátor TL431A.

Obvod TL431A realizuje zpětnovazební napěťový regulátor s interní referencí 2,5 V, který může pomocí dvojice externích rezistorů pracovat v rozsahu 2,5 - 36 V. Může pracovat i v režimu proudového regulátoru v rozmezí 0,1 - 1 A. Povolený ztrátový výkon pro menší pouzdra je 500 mW. Obvod zaručuje rychlou přechodovou charakteristiku při zapnutí, nízký výstupní šum a teplotní stabilitu v celém podporovaném rozsahu teplot -25 - 85°C. Vnitřní zapojení obvodu sestává z komparátoru, na jehož jeden vstup je připojena interní reference 2,5 V a na druhý referenční vstup obvodu. Na výstupu rozdílového zesilovače je připojen výkonový tranzistor, který pracuje ve své lineární oblasti. Realizuje tak proměnný rezistor, který udržuje napětí na požadované hodnotě. Napětí regulátoru se uspočte z následujícího vzorce:

$$U_{out} = U_{ref} \cdot \left(1 + \frac{R_1}{R_2}\right) + I_{ref} \cdot R_1 \quad (3.1)$$

Přičemž ve většině případů můžeme poslední člen vynechat. Získáme tedy následující vztah:

$$U_{out} = U_{ref} \cdot \left(1 + \frac{R_1}{R_2}\right) \quad (3.2)$$

Vysílač řídící jednotky OpenTherm přepíná mezi dvěma úrovněmi napětí. Ty jsou dány povolenými rozmezími definovanými v protokolu. Přepnutí mezi těmito dvěma úrovněmi dosáhneme změnou odporu připojených rezistorů R_4 , R_5 . Nejjednodušší způsob, jak dosáhnout požadovaného efektu je zapojit paralelně k rezistoru R_4 tranzistor ovládaný mikrokontrolerem, který bude pracovat jako spínač. Tím můžeme přepínat mezi dvěma pracovními body regulátoru a tím měnit napětí smyčky, což od vysílače požadujeme. Po aplikaci tohoto zapojení jsme schopni dosáhnout horní hranice U_H dané max.napětí smyčky, což je 24 V a dolní hranice $U_L = 5$ V použitím vztahu 3.2 a informace, že doporučený součet hodnot rezistorů je 10 kΩ.

$$5 = 2,5 \cdot \left(1 + \frac{R_1}{R_2}\right)$$

$$\frac{R_1}{R_2} = \frac{5}{2,5} - 1 = 1$$

Tedy odpory R_4 a R_5 jsou si rovny a každý má hodnotu 4,7 kΩ.

3.3.2 Návrh přijímače

Vysílání zařízení připojeného k naší jednotce bude spočívat ve změně hodnoty proudu protékajícího komunikační smyčkou. Tento proud proto musíme převést na napětí, zesílit a přivést na jednu ze vstupních bran procesoru. K převodu využijeme měřícího odporu R_3 . Změřený úbytek napětí z R_3 přivedeme na kladný vstup operačního zesilovače TLC272 zapojeného jako komparátor. Zde jej porovnáme s referenční hodnotou získanou z napájecího napětí operačního zesilovače děličem R_1 , R_2 a výstup již propojíme do samotného procesoru. Odpor R_3 volíme malý, aby zbytečně nezatěžoval komunikační smyčku. Referenci poté určíme tak, že spočteme úbytky napětí na R_3 pro obě proudové úrovně a zvolíme hodnotu přibližně uprostřed tohoto intervalu.

$$U_{R3H} = R_3 \cdot I_H = 4,7 \cdot 0,017 = 0,08V \quad (3.3)$$

$$U_{R3L} = R_3 \cdot I_L = 4,7 \cdot 0,009 = 0,0423V \quad (3.4)$$

Ted' již jednoduše určíme odpor rezistoru R_2 :

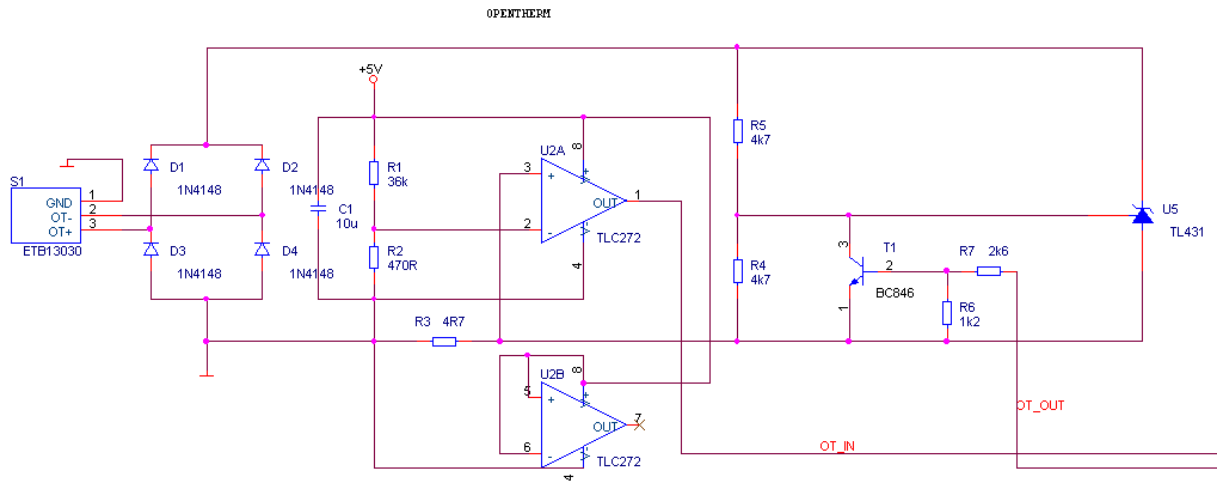
$$U_{R2} = U_{cc} \cdot \left(\frac{R_2}{R_1 + R_2} \right) \quad (3.5)$$

$$R_2 = R_1 \cdot \left(\frac{U_{R2}}{U_{cc} - U_{R2}} \right) \quad (3.6)$$

$$\begin{aligned} &= 36000 \cdot \left(\frac{0,611}{5 - 0,611} \right) \\ &= 445,272\Omega \end{aligned} \quad (3.7)$$

Nejbližší standardní hodnota rezistoru je 470Ω .

Výsledné zapojení poté kombinuje oba dva návrhy do jednoho celku. Tedy za vstupem je zapojen diodový můstek, poté následuje napěťová vysílací část s regulátorem TL431A a nakonec proudová přijímací část s operačním zesilovačem TLC 272 a měřícím odporem R_3 . Zapojení dokumentuje obrázek 3.5.

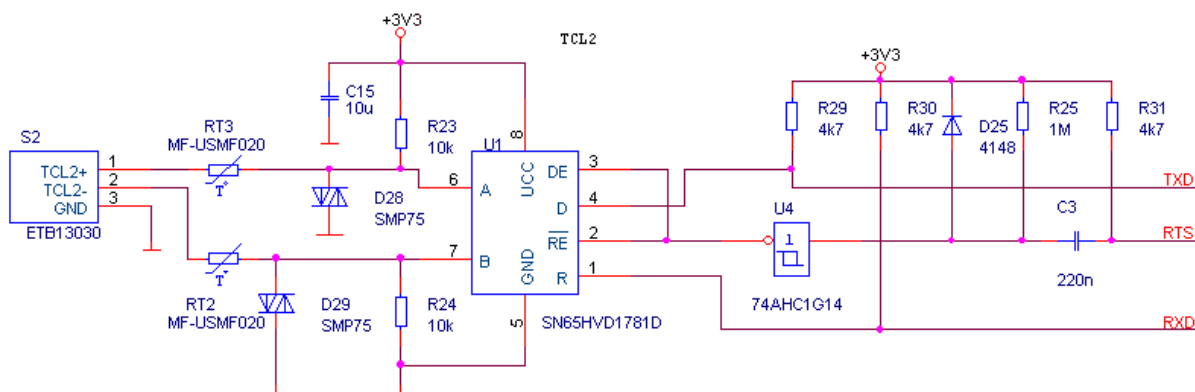


Obrázek 3.5: Schéma zapojení vysílače a přijímače OpenTherm

3.4 Návrh vysílače a přijímače TCL-2

Protože sběrnice TCL-2 vychází z rozhraní RS-485, bude vysílací/přijímací část s tímto rozhraním kompatibilní. Jako vysílač/přijímač je použit obvod SN65HVD1781. Obvod má tři vstupy, driver(D), driver enabled(DE) a read enabled($\overline{RE}$), dále tři výstupy. Read(R), což je datový výstup, kterým se z obvodu distribuuje přijatý signál a výstupy A a B, které reprezentují sběrnici RS-485. Vstup D a výstup R jsou datové, proto se připojí přes pull-up odpory na datové vodiče TxD a RxD.

Vstupy DE a $\overline{RE}$ slouží k aktivaci vysílací/přijímací části, přičemž se dají zapojit na jeden signální vodič, v našem případě (Ready To Send)RTS přicházející z procesoru. Ten přepne obvod do režimu vysílání. Po skončení přenosu se nastaví RTS do log.0 a obvod se přepne do režimu příjmu. Protože RTS je připojen na $\overline{CTS}$ pin procesoru, jehož funkce je dána kanálem UART a jeho úroveň nelze změnit programově, je před vstupem DE a $\overline{RE}$ zařazen negátor, jmenovitě obvod 74AHC1G14, který zajistí správné provázání vysílače s mikrokontrolerem. Na výstupech A a B jsou připojeny opět pull-up, resp. pull-down odpory tak, že vodič A je kladný a vodič B je zemněn, dále jsou zde připojeny jistící obvody proti přepětí a proudovým špičkám.



Obrázek 3.6: Schéma zapojení vysílače/přijímače TCL-2

3.5 Návrh zapojení mikrokontroleru

Mikrokontroler Z8 Encore a prvky čelního panelu, tzn. potenciometr pro nastavení adresy a dvě stavové LED diody, jsou umístěny na boční desce modulu spolu se svorkami pro připojení rozhraní ZDI při ladění a stabilizátorem napětí z 5 V na 3,3 V. Použité LED diody jsou umístěny v jednom pouzdře typu H201CBC a umožňují přepínání mezi dvěma barvami, zelenou a červenou. Tyto barvy se dají přepínat přivedením vysoké úrovně na jeden ze dvou pinů, které přísluší každé diodě. Jsme tak schopni snadno a přehledně signalizovat různé stavy modulu. Připojení prvků čelního panelu k procesoru se řídí hlavně jejich umístěním na desce plošných spojů, které je pevně dané požadavkem na unifiko- vaný vzhled všech modulů připojitelných k PLC Foxtrot.

Mikrokontroler budeme využívat k řízení komunikace po obou rozhraních, proto k němu musíme zavést všechny potřebné signály. Vysílání a příjem OpenTherm realizu- jeme pomocí dvou signálových vodičů, OT-IN a OT-OUT. Vodič OT-OUT je z hlediska procesoru výstup do vysílací části modulu, chová se jako standardní výstup a můžeme ho připojit takřka libovolně. OT-IN, což je vstup pro příjem přicházejícího vysílání, je třeba připojit na bránu, která je uzpůsobena na příjem 5 V signálů a podporuje přerušení při změně logické úrovně pinu. Proto volíme mezi branami PA a PD. Z hlediska rozložení na desce plošných spojů je výhodné vodič OT-OUT připojit na pin PA4 a vodič OT-IN na

pin PD4.

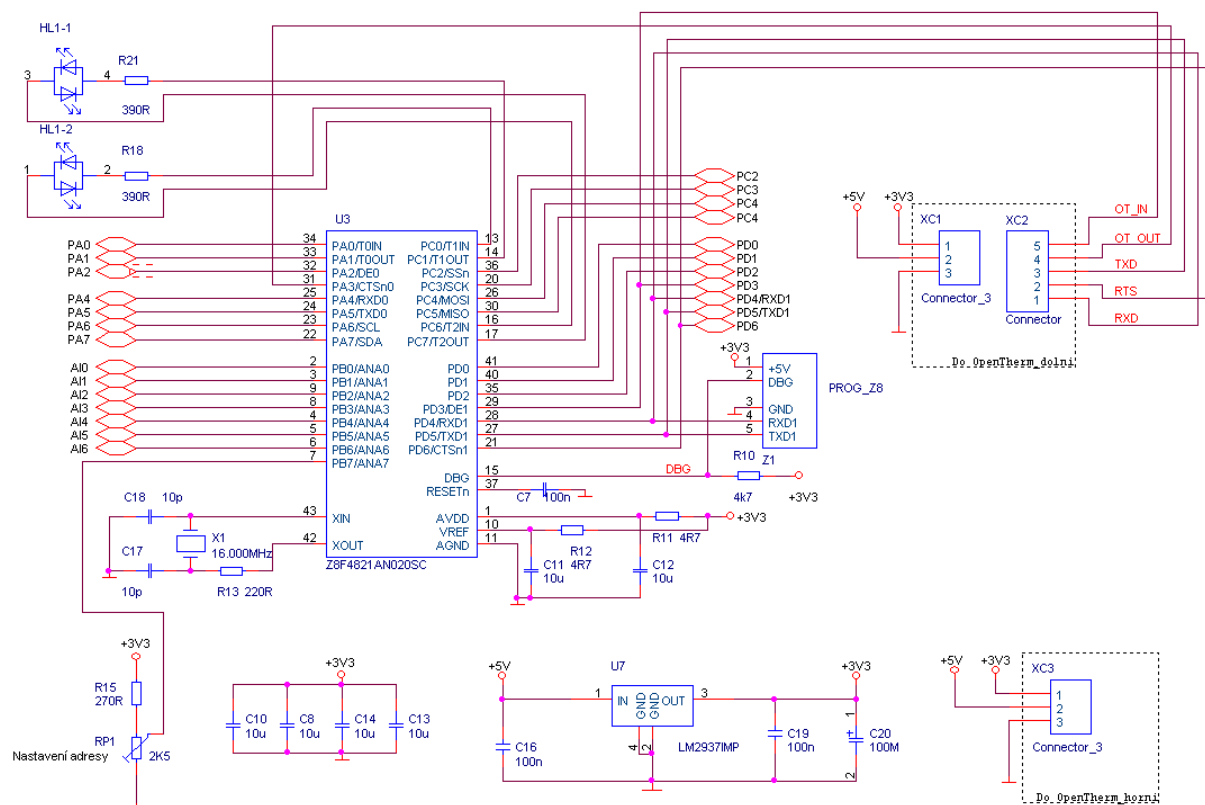
K vysílání a příjmu po sběrnici TCL-2 využijeme kanál UART, konkrétně kanál UART1. Hlavním důvodem této volby jsou rutiny vytvořené firmou Teco a dodané k této práci. Ty jsou totiž navrženy pro práci s tímto kanálem. Vysílací vodič TxD tedy připojíme na pin TxD PD4, přijímací vodič RxD na pin RxD PD5 a vodič RDS(Ready To Send - připraveno k odeslání) na vodič $\overline{CTS}$ PD6. Výměna dat mezi částí vysílače/přijímače a procesorem poté probíhá jako standardní sériová komunikace. Všechny použité piny a jejich význam jsou uvedeny v tabulce 3.1, nevyužité brány zůstaly nezapojeny.

Další částí návrhu této desky je stabilizátor 5 na 3,3 V, který zajišťuje napájecí napětí pro mikrokontroler. Jedná se o regulátor s fixním výstupem LM2937IMP ve verzi s výstupním napětím 3,3 V. Tento obvod je schopen dodávat výstupní napětí 3,3 V s tolerancí $\pm 5\%$ v rozsahu vstupního napětí 4,75 - 26 V při maximálním výstupním proudu 400 mA. Jeho zapojení je doplněno ještě o vstupní a výstupní blokovací kondenzátor a vyhlazovací kondenzátor na výstupu.

Celkové schematické znázornění návrhu je zobrazeno na obrázku 3.7.

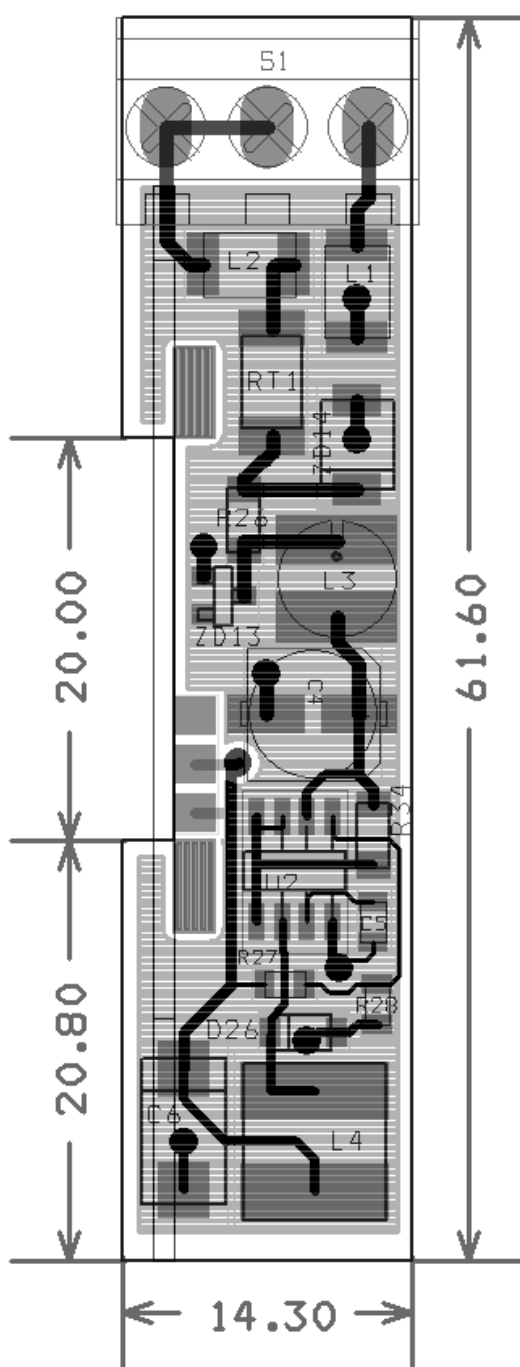
Tabulka 3.1: Použité piny procesoru a jejich význam v aplikaci

Označení pinu	Funkce v aplikaci
PA3	Vysílání OpenTherm(OT-OUT)
PB7	AD převodník adresy z potenciometru
PC0	Ovládání LED diody 2
PC1	Ovládání LED diody 1
PC6	Ovládání LED diody 2
PC7	Ovládání LED diody 1
PD3	Příjem OpenTherm(OT-IN)
PD4	RXD UART1 pro TCL-2
PD5	TXD UART1 pro TCL-2
PD6	CTS UART1 pro TCL-2
DBG	Připojení ZDI

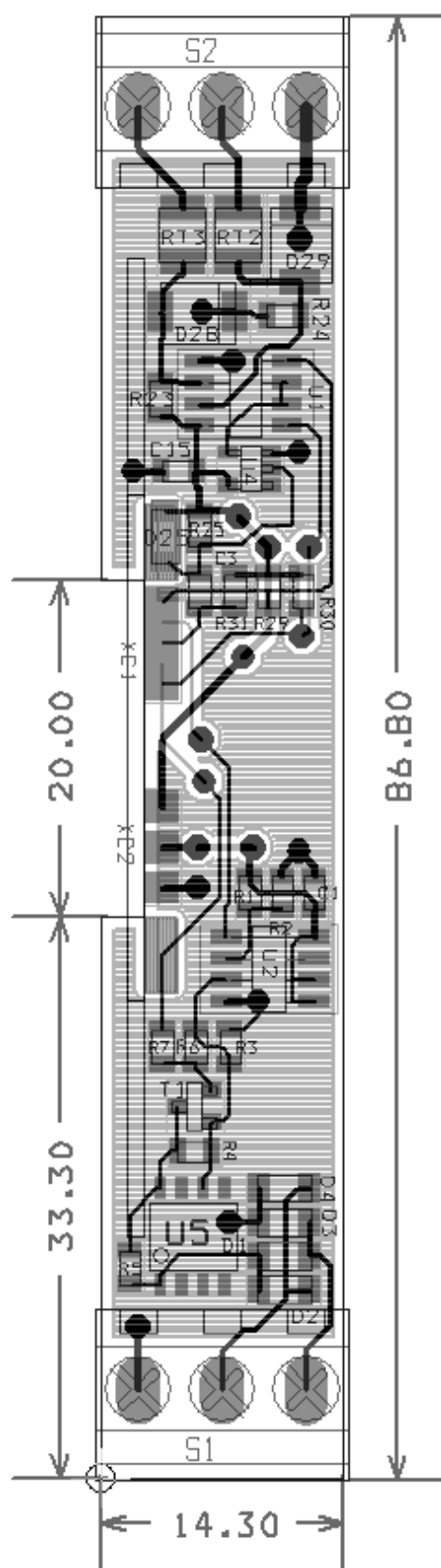


Obrázek 3.7: Schéma zapojení mikroprocesoru

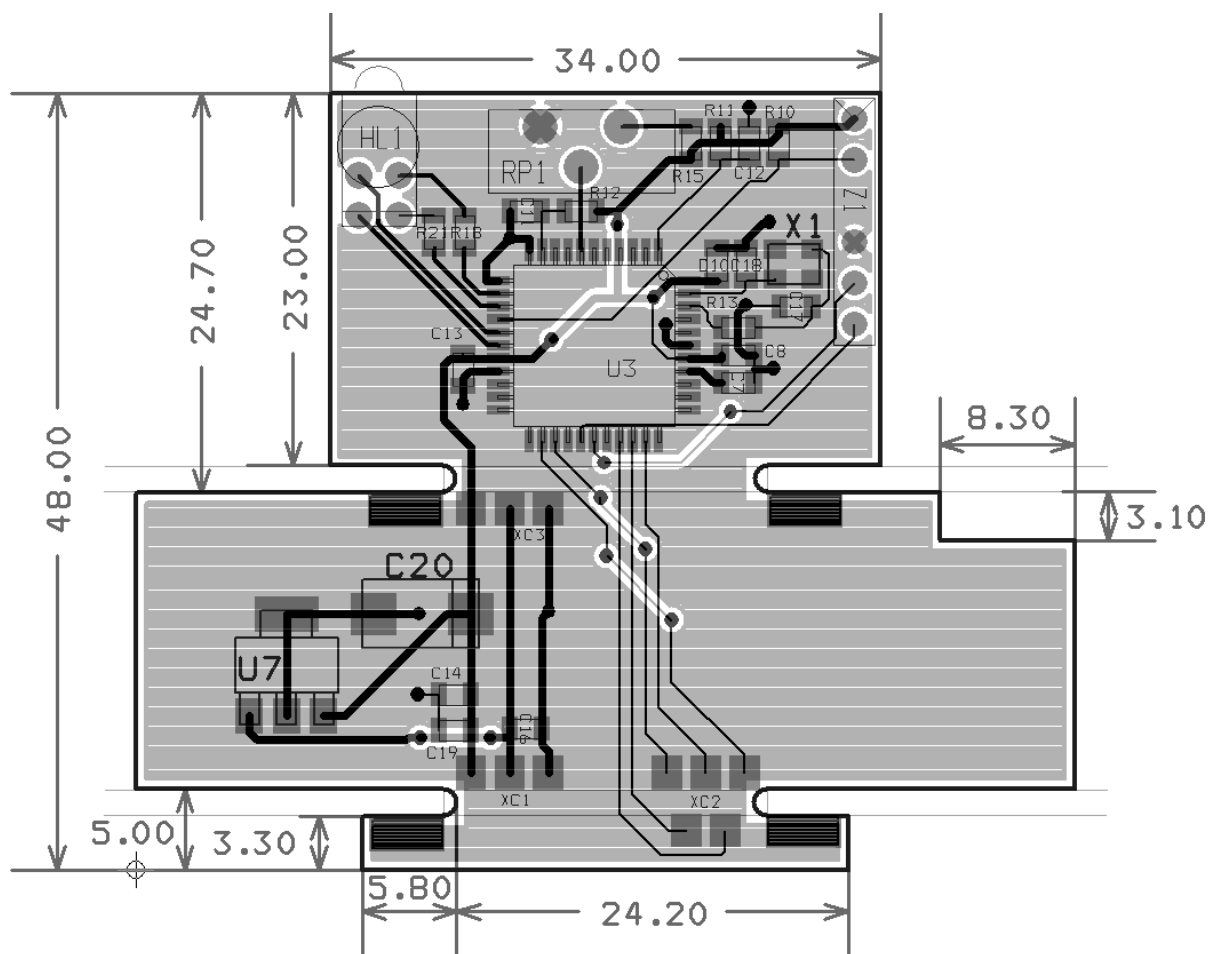
3.6 Navržené desky plošných spojů



Obrázek 3.8: Návrh desek plošných spojů - deska napájecího zdroje



Obrázek 3.9: Návrh desek plošných spojů - deska komunikačních rozhraní



Obrázek 3.10: Návrh desek plošných spojů - deska mikrokontroleru

Kapitola 4

Návrh řídicího software modulu

Řídicí software modulu je celý vytvořen v jazyce C v prostředí ZDS II. Je navržen pro mikrokontroler ZiLOG Z8 a stejně jako hardwarový návrh modulu sestává z několika menších celků. Rutiny těchto částí jsou poté podle oblasti, které se týkají, rozděleny do jednotlivých souborů. Cílem je navrhnout rutiny pro komunikaci podle protokolu OpenTherm, dále se seznámit s rutinami pro komunikaci s rozhraním TCL-2 a uzpůsobit je pro použití v software modulu. V poslední části poté spojit obě dvě komunikační části do jednoho celku a zajistit jeho funkčnost.

Z hlediska komunikace je prioritní výměna dat s procesorem, proto veškerá komunikace se sběrnici TCL-2 mají prioritu před obsluhou rozhraní OpenTherm. Vzhledem k rychlostem obou sběrnic, kdy OpenTherm komunikuje rychlostí 1000 b/s a TCL-2 345,6 kB/s, tedy nastává problém, protože v průběhu vysílání a příjmu OpenTherm zpráv bude docházet k průběžné obsluze sběrnice TCL-2. Protože komunikace na obou rozhraních není nijak synchronizována a nechceme, aby jedna z částí zbytečně blokovala procesor, je v programu klíčovým požadavkem rychlost a jednoduchost vykonávání kritických operací, jako je vysílání nebo příjem a dekodování zpráv (v případě TCL-2).

4.1 Vysílání a příjem signálu po rozhraní OpenTherm

Řízení rozhraní OpenTherm tvoří jednu samostatnou část návrhu software modulu. Jak již bylo řečeno v předchozí kapitole, navržené rozhraní OpenTherm je realizováno

jen dvěma datovými vodiči, OT-OUT, který data vysílá a OT-IN, který data přichází z připojeného zařízení přijímá. Nyní se budeme zabývat tvorbě těchto signálů na vysílací straně a jejich zpracováním na straně přijímače.

4.1.1 Vysílání signálu

Vysílání signálu realizujeme přímo změnou logických úrovní pinu PA3, nevyužíváme tedy žádné speciální funkce procesoru kromě časovačů, které zaručí správnou periodu komunikace a vysílání jednotlivých bitů zprávy. Komunikace po rozhraní OpenTherm je vždy inicializována řídicí jednotkou, tedy naším modulem, přičemž max. přijatelná perioda od konce vysílání do začátku následujícího je 1,15 s. Pro dodržení této periody je vhodné využít jeden časovač, v našem případě T2, resp. jeho přerušení jako spouštěcí signál začátku vysílání. Nastavíme tedy příslušný registr přerušení a globálně přerušení povolíme. Protože časovače našeho mikrokontroleru jsou 16-bitové a jejich časování je odvozeno od frekvence procesoru 16 MHz, přičemž maximální dělicí poměr frekvence je roven 128, jsme schopni dosáhnout maximální periody 0,52428 s. Proto nastavíme registry na periodu 0,5 s a budeme vysílat vždy až po dvou přerušeních od časovače. Tím dostaneme celkový rastr komunikace, kdy vysílání bude probíhat vždy po 1 s od začátku předchozího, čímž zachováme určitou rezervu od horní i dolní meze intervalu.

K vysílání i příjmu je využit časovač T1. Ten je při obsluze 2.přerušení časovače T2 nastaven příznakem *ot_send_enabled* na režim vysílání a spuštěn na dobu 500 μ s, což je polovina doby potřebné k vyslání jednoho bitu. Dále je v tomto okamžiku rozsvícena komunikační LED dioda na čelním panelu modulu. Samotné vyslání je tedy rozděleno na dvě části. V první části je po uplynutí tohoto intervalu vyvoláno přerušení, v rámci něhož je nastavena log.úroveň podle hodnoty vysílaného bitu a aktivován příznak *edge* pro identifikaci druhé části vysílání. Poté je opět spuštěn časovač T1. V druhé části, opět v obsluze přerušení časovače, je překlopena úroveň nastavená v první části a tím realizována hrana reprezentující vysílaný bit dle kódování Manchester. Opět je spuštěn časovač a vynulován příznak *edge*. Poté je inkrementována proměnná *bit_count_tx* čítající počet bitů a postup se opakuje. Takto je postupně odeslána celá zpráva. Nakonec je nastaven příznak úspěšného zápisu *write_flag*, příznak *ot_send_enabled* vynulován pro režim příjmu a zhasnuta komunikační LED dioda. Postup je díky implementaci v rámci obsluhy přerušení nenáročný na čas procesoru, protože je vykonáván jen v krátkých časových úsecích.

4.1.2 Příjem signálu

Příjem signálu lze rozdělit na dvě části. První částí je obsluha přerušení na pinu PD3, který slouží jako vstup z přijímací části rozhraní OpenTherm. Druhou částí je obsluha přerušení časovače T1 v režimu pro příjem. Přerušení pro příjem na pinu PD3 je nastaveno vždy až po úspěšném odeslání zprávy vysílačem. Je tomu tak kvůli špičkám, které vznikají na sběrnici při vysílání a mohly by být chybně považovány za odpověď.

Poté se čeká na příjem první hrany, která je díky faktu, že zpráva začíná vždy start bitem rovným 1 vždy vzestupná (aby mohl vysílač vyslat start bit, musí nejprve nastavit log. úroveň na sběrnici do log.1). Tato hrana je zachycena a úroveň na sběrnici uložena do proměnné *link_status*, poté je přepnuta detekce hrany přerušením na sestupnou hranu, nastaven příznak *first_read* a obsluha přerušení končí. Tato operace je provedena vždy jen jednou za zprávu. Po příjmu sestupné hrany je opět uložena úroveň na sběrnici tentokrát do proměnné *receive_bit* a porovnána s úrovní v *link_status*. Pokud se liší, je bit, který vyjadřuje vzniklou hranu, uložen do přijímací struktury, zapnut časovač T1 s periodou 700 μ s a obsluha končí.

Po uplynutí této doby je vyvoláno přerušení časovače T1 a v jeho obsluze v režimu přerušení uložena úroveň do již zmíněné proměnné *link_status*. Poté je očekávána hrana podle úrovně v *link_status*. Je-li uložena 1, bude hrana sestupná, je-li uložena 0, bude vzestupná. Poté je obsluha ukončena a čeká se na přerušení na pinu PD3, jehož obsluha poté probíhá stejným způsobem, jako v případě start bitu. Počet přijatých bitů je čítán proměnnou *bit_count_rx* a po přečtení všech bitů je nastaven příznak *read_flag* do hodnoty 2, což značí úspěšné čtení.

Nastalé chyby mohou vzniknout buď z důvodu neodbavení hodnoty *link_status*, což je detekováno příznakem *link_status_flag*, nebo z důvodu nezachycené, či špatně změřené (nezaregistrované) hrany. Tyto detekované chyby nastaví příznak *read_flag* do 1, což značí neúspěšné čtení a příjem zprávy je ukončen. Příjem zpráv je realizován na stejném principu jako vysílání, vykonání požadovaných instrukcí zabere jen krátký čas, což je výhodné zejména z hlediska minimalizace chyby ve čtení příchozí zprávy.

Po prvním úspěšném čtení je také nastaven příznak *lite_switch*. Ten slouží k detekci verze protokolu, kterým připojené zařízení komunikuje. Pokud není příznak aktivní, při

každém přerušení časovače T2 je inkrementována hodnota proměnné *lite_count*. Pokud modul nepřijme po dvaceti pokusech odpověď, přejde do Lite verze protokolu. Pokud již jednou zprávu úspěšně přijal, příznak zůstane nastaven a verze Lite je trvale neaktivní do restartu modulu. Taktéž z verze Lite do verze Plus lze přejít jen restartem.

4.2 Vysílání a příjem po sběrnici TCL-2

Komunikace po sběrnici TCL-2 probíhá stylem master-slave, tedy master cyklicky komunikuje s připojenými zařízeními. Komunikace s navrženým modulem je prováděna prostřednictvím kanálu UART1, což značně zjednoduší návrh komunikačního software. Kanál je nastaven do multiprocessorového režimu, což znamená, že první zpráva komunikačního rámce je vždy 9-bitová, nesoucí adresu cílového zařízení. Adresa je každým zařízením porovnána s hodnotou uloženou v registru, který byla při inicializaci vyčtena z adresního potenciometru. Pokud souhlasí, zařízení přijímá další zprávy rámce o délce 8 bitů, jinak zbytek rámce ignoruje.

Pokud je zpráva v pořádku přijata, je ověřen její CRC součet a poté dekodován její význam. Přijatá zpráva je podle výsledku dekodování buď uložena do vstupní datové zóny modulu, pokud jde o datovou zprávu, nebo je na ni odpovězeno, pokud jde o požadavek na data. Datové zprávy v naší aplikaci se týkají hlavně komunikace s připojeným kotlem či jiným OpenTherm kompatibilním zařízením, a proto se jedná většinou o méně či více upravné zprávy protokolu OpenTherm, na které je posléze požadována odpověď. Komunikace tedy vypadá tak, že kromě systémových a stavových zpráv přicházejí periodicky požadavky na zaslání výstupní datové oblasti, kterou je třeba naplnit požadovanými daty. Pokud jde o zprávu stavovou či systémovou, jsou vykonány instrukce, které zpráva obsahuje. Pomocí stavových a systémových zpráv lze například přepínat stav modulu, vyžádat reset modulu, dále může jít o dotaz na stav modulu, který je poté zobrazen v prostředí Mosaic, nebo o inicializaci k přehrání firmware modulu.

Vysílání probíhá naplněním výstupní zóny dat modulu novými daty. Poté je vypočten CRC součet zprávy, který je přidán na konec. Následně je aktivován pin RTS, který značí odesílání dat do externího RS-485 vysílače, a DMA kanál, který výstupní datovou oblast postupně přenesení do vysílacího bufferu. Z něj jsou data průběžně odesílána do externího

vysílače. Po skončení přenosu DMA vyvolá přerušování, v jehož obsluze je RTS nastaveno na příjem.

Specifikace komunikační sběrnice TCL-2 podléhá interním předpisům firmy TECO a.s., které nedovolují jeho zveřejnění. Proto zde podrobnější popis softwarové realizace tohoto rozhraní nelze uvést.

4.3 Formáty datových zpráv

4.3.1 Formát zpráv v módu OpenTherm Plus

V aplikaci se vyskytuje několik typů a úprav datových zpráv. Zprávy vždy vycházejí z tvaru definovaného protokolem OpenTherm. Pro komunikaci modulu s PLC je rámec modifikován pro lepší zpracování v prostředí Mosaic. Úpravy spočívají v tom, že zpráva, původně 4-bytová, je roztažena do 6 bytů. První byte je kontrolní. Je v něm umístěn na nejvyšší pozici alternační bit. Ten slouží k rozpoznání nové OpenTherm zprávy zaslané z centrální jednotky. Po každé nové přijaté zprávě je jeho hodnota změněna. Hodnota tohoto bitu je nastavena taktéž u odpovědi, která se odešle zpět PLC. Dají se tak snadno odlišit nově příchozí data od neaktuálních. První zpráva musí mít alternační bit roven 1, aby bylo možné zprávu správně identifikovat. Na nejnižší pozici tohoto byte pak modul signalizuje centrální jednotce výpadek OpenTherm komunikace.

Druhý byte má význam typu zprávy, který je shodný s protokolem OpenTherm. Jednotlivé typy jsou uvedeny v tabulce 2.2. Typ zprávy je umístěn ve spodních třech bitech. Zbytek zprávy je neobsazen. Čtvrtý byte je prázdný a významem odpovídá 4 volným bitům (spare bits) zprávy OpenTherm. Poslední tři byty již plně odpovídají svým protějškům v OpenTherm zprávě. Jedná se o data id, data high a data low byte. Struktura zprávy je zobrazena v následující tabulce.

Tabulka 4.1: Rámec datové zprávy mezi modulem a PLC

Kontrolní byte	Typ zprávy	Volný byte	Data ID	Data[2 byte]
----------------	------------	------------	---------	--------------

Další možná odchylka nastává v případě, že jde o zprávu, která využívá typ f8.8 - číslo s pevnou řádovou čárkou. Jedná se o zprávy obsahující číselný údaj o teplotě,

tlaku, či jinou neceločíselnou hodnotu. U těchto dat je nejprve provedena konverze datových typů v datové části zprávy. PLC Foxtrot využívá číslo s pevnou řádovou čárkou ve formátu signed integer posunuté o 1 desetinné místo doprava, rozsah je tedy -3276,7 až 3276,7. Popis typu f8.8 protokolu OpenTherm je uveden v kapitole 2 pod tabulkou 2.4. Konverze $\text{int} \rightarrow \text{f8.8}$ se provede tak, že číslo před řádovou čárkou se zachová, číslo mající význam desetin se vynásobí 255. Opačná konverze naopak spočívá v dělení čísla 255 a zaokrouhlení na jedno desetinné místo. Tato úprava se provádí jak na vstupu, tak na výstupu modulu vzhledem k centrální jednotce.

V samotné aplikaci je standardní datová zpráva, se kterou rutiny pracují, uchovávána ve formátu tohoto protokolu. Je tedy tvořena 4 byty, jejichž popis je v tabulce 2.3. K přehlednosti obsahu zprávy je definována struktura OT_MESSAGE, do které je zpráva uložena a z níž je postupně zpracovávána. Struktura sestává ze 4 částí tvořených jednotlivými byty. Jedná se o hlavičku(header), id zprávy(data id) a 2 byty dat(data high a data low). Přijatá zpráva může být tedy s minimálními úpravami ihned odeslána na rozhraní OpenTherm.

4.3.2 Formát zpráv v módu OpenTherm Lite

Pokud pracuje modul v režimu Lite, formát komunikační zprávy s centrálou se nezmění. Stále se používá alternační bit k aktualizaci požadavku, ale jediným požadavkem může být nastavení střídý signálu. Hodnota střídý je poté rovna hodnotě požadované teploty topné vody v rozsahu 0-100 uložené v posledních dvou bytech zprávy v řádu desetin. Pokud je tato hodnota menší než 50(5,0°C), je topení vypnuto. Ostatní byty jsou v režimu Lite ignorovány.

4.4 Struktura navrženého řídicího programu

Sestavený program slučuje obě komunikační části a umožňuje tedy komunikaci jak se zařízením OpenTherm, tak s centrální jednotkou Foxtrot. Po spuštění je inicializován a spuštěn watchdog časovač, který resetuje procesor v případě zacyklení. Poté je vypočteno CRC firmware, které musí souhlasit s hodnotou uloženou v paměti modulu. Pokud CRC nesouhlasí, je vykonávání programu zachyceno ve while smyčce a ukončeno resetem pro-

cesoru watchdog časovačem. Pokud CRC souhlasí, jsou inicializovány veškeré používané periferie procesoru, inicializovány používané brány a načtena adresa modulu v rozmezí 0 - 9, která je nastavena potenciometrem na čelním panelu. Poté jsou nastaveny vektory přerušení a program se dostává do nekonečné smyčky, která symbolizuje standardní chod modulu.

Vykonávání smyčky probíhá v následujících krocích:

1. Kontrola funkčnosti komunikace TCL-2 a ošetření stavových zpráv
2. Přesunutí dat do výstupní datové oblasti TCL-2
3. Přesunutí dat ze vstupní datové oblasti TCL-2
4. Kontrola požadavku na výměnu firmware
5. Naplnění struktury OpenTherm daty k odeslání

4.4.1 Kontrola funkčnosti komunikace TCL-2 a ošetření stavových zpráv

Ve smyčce probíhá nejprve kontrola spojení TCL-2, která je realizována příznakem *alarm*. Poté jsou ošetřeny žádosti na přechod mezi stavy. Modul se může nacházet ve stavu HALT, kdy nekomunikuje s periferií a přijímá zprávy po sběrnici TCL-2. Odstávka komunikace je realizována vypnutím časovače T2, který určuje celkový komunikační rámec OpenTherm. HALT je defaultním stavem, do kterého modul po inicializaci přejde. Tento mód je signalizován svítící stavovou LED diodou. Druhým režimem modulu je stav RUN, v němž je navázána komunikace se zařízením OpenTherm a modul je plně aktivní. Tento mód je indikován blikající stavovou LED diodou. Režimy HALT a RUN modulu jsou přímo závislé na režimu centrální jednotky. Modul nemůže samovolně mezi stavy přecházet, pokud nenastal výpadek komunikace TCL-2.

4.4.2 Obsluha výstupní datové oblasti TCL-2

Nejdůležitější částí pro standardní fungování modulu jsou části kódu smyčky, ve kterých se manipuluje s datovými oblastmi sběrnice TCL-2. Jde o dvě operace, vkládání dat určených PLC do výstupní datové oblasti, nebo naopak jejich vyčítání dat ze vstupní

datové oblasti.

Vkládání dat probíhá tehdy, pokud je registrován požadavek na jejich odeslání. Tento požadavek vyvolá příchozí systémová zpráva s požadavkem na obsah výstupní datové oblasti TCL-2. Připravená data, která v našem případě reprezentují odpověď kotle na zprávu řídicí jednotky, mohou být dvojího typu. Pokud byla zaregistrována nová platná zpráva z PLC směrem do zařízení OpenTherm, je nastaven příslušný příznak. Po úspěšném čtení, které je signalizováno nastavením příznaku *read_flag* do hodnoty 2, jsou nová data do oblasti zkopírována ze struktury OT_MESSAGE *ot_read*, kde je pouze paritní bit (nejvyšší bit 1. byte zprávy) nahrazen alternovaným bitem, který se překlápí po každém odeslání nových dat. V opačném případě se posílají poslední platná data uložená v oblasti.

4.4.3 Obsluha vstupní datové oblasti TCL-2

Přijdou-li nová data z PLC, jsou po dekódování zprávy uložena v definované vstupní datové oblasti a nastavením příznaku je vytvořen požadavek na jejich odbavení. To se děje v hlavní smyčce po porovnání jejich alternovaného bitu s poslední přijatou hodnotou a uloženou hodnotou, buď přesunutím těchto dat do struktury *ot_temp*, odkud se dále přesouvají ve vhodném okamžiku do struktury *ot_write*, nebo jejich ignorováním, pokud se bit neliší od předchozí hodnoty. Toto opatření je zavedeno z důvodu řádově jiné rychlosti TCL-2 a OpenTherm a tedy reálné možnosti příjmu velkého množství požadavků bez možnosti je obsloužit.

4.4.4 Požadavek na výměnu firmware

V další fázi smyčky je kontrolován případný požadavek na přehrání firmware modulu. Tento požadavek může vyvolat centrální jednotka na základě externího programu v připojeném PC. Po zaregistrování tohoto požadavku je do centrální jednotky nahrána nová verze řídicího programu, která je poté přenesena po sběrnici TCL-2 do samotného modulu. Zde je nový kód uložen do vyhrazené paměťové oblasti, odkud je po restartu modulu přehrán na místo původního programu. Nakonec je modul znovu restartován, načež již pokračuje podle nového programu.

4.4.5 Odbavení požadavků na zařízení OpenTherm

V poslední části běhu smyčky je realizována komunikace směrem k rozhraní OpenTherm. Zde je nejprve rozlišeno, zda modul pracuje v Lite či Plus verzi protokolu. Poté následuje pro Lite verzi generování obdélníkového signálu odpovídajícího přijaté hodnotě. Pro verzi Plus je vyčtena uložená zpráva ve struktuře *ot_temp* a uložena do struktury *ot_write*, odkud je po přetečení časovače T2 odeslána po rozhraní OpenTherm. Pokud je struktura *ot_temp* prázdná, je stále třeba udržovat komunikaci, proto je do přijetí další platné zprávy z PLC posílána neplatná zpráva (s hlavičkou Data invalid, viz. 2.2), rozlišení platných a neplatných dat je zajištěno příznakem *no_mess*, podle kterého se poté buď nazpět přijatá data z rozhraní OpenTherm odešlou zpět do PLC, nebo ignorují.

Do vykonávání smyčky dále vstupují přerušení k ošetření komunikace na obou rozhraních. Ty jsou způsobena buď příchodem nových dat, nebo přetečením časovačů, jak je popsáno v podkapitolách věnovaných softwarové realizaci obou rozhraní.

4.5 Nastavení modulu v prostředí Mosaic

Programovatelný automat Tecomat Foxtrot se programuje v prostředí Mosaic. Proto po připojení modulu k jednotce Foxtrot je třeba nakonfigurovat centrální jednotku tak, aby modul rozpoznala a mohla používat. Je proto třeba přidat do projektu soubor s příponou *.mos*, který obsahuje definici struktur využívaných k popisu modulu. Tento soubor je zapsán v jazyce ST.

Soubor obsahuje inicializační strukturu *tinit1204* použitou k standardnímu popisu vstupů a výstupů modulu:

```
#struct tinit1204
    word code,      ; 00 kód jednotky pro kontrolu platnosti dat */
    byte statd0,    ; 02 režim výměny dat */
    byte statd1,    ; 03 režim výměny dat */
    byte evari,     ; 04 povolení vstupních proměnných
    byte evaro      ; povolení výstupních proměnných
```

Struktura má definovaný tvar a obsahuje nejprve povinné prvky *code*, *statd0*, *statd1*, *evari*, *evaro*, jejichž význam je vysvětlen v popisu kódu. Modul může mít defi-

novány i další vstupy a výstupy, využívané například k účelům diagnostiky, nebo předávání pomocných dat do modulu. Tyto části struktury se definují nakonec a závisují na typu modulu.

Dále soubor obsahuje definici vstupních a výstupních registrů modulu:

```
#reg byte dataIN[4]          ; vstupni data
#reg byte dataOUT[4]         ; vystupni data
```

Následuje vytvoření instance struktury s defaultními parametry modulu:

```
#table tinit1204 UCinit =    1204,
                             $00,
                             $00,
                             $80,      ; vstupy povoleny
                             $80      ; výstupy povoleny
```

Nakonec je vytvořena instance modulu s upřesněním jeho verze, pozice v rámu, a přiřazením příslušných datových oblastí a inicializační tabulky:

```
#module TModule1 1, ; verze modulu
                1, ; číslo rámu
                0, ; pozice v rámu
                $0, ; lokalizace
                4, ; délka vstupních dat
                4, ; délka výstupních dat
                __offset(dataIN[0]), ; výstupní oblast modulu
                __offset(dataOUT[0]), ; vstupní oblast modulu
                __indx(UCinit) ; jméno inicializační tabulky
```

Poté je již modul připraven k použití. V uživatelském programu je třeba ještě deklarovat externí proměnné se stejnými jmény, jako mají námi definované datové oblasti. Způsob deklarace a struktura programu je ukázána na jednoduchém příkladu, který byl použit při testování modulu. Tento program je popsán v příloze A.

Po instalaci nejnovější verze prostředí Mosaic je usnadněno použití modulu v projektu. Modul byl zařazen mezi automaticky podporované externí moduly pro PLC Foxtrot pod označením UC-1204. Je tedy možné modul vybrat v HW konfiguraci v Manažeru projektu. Stačí zvolit pozici v rámu a přiřadit typ modulu. Poté je inicializace automatická,

modulu jsou přiřazeny vstupní a výstupní datové zóny a je možné sledovat jeho stav a hodnoty proměnných v části Nastavení V/V.

Kapitola 5

Realizovaný modul

Realizovaný modul ponese označení UC-1204. Splňuje veškeré komunikační parametry protokolu OpenTherm, a proto může být nasazen ke komunikaci s OpenTherm kompatibilními zařízeními. Komunikace s připojeným zařízením probíhá prostřednictvím programu v centrální jednotce Foxtrot, ke které musí být modul připojen a konfigurován. Komunikace poté z hlediska uživatele probíhá ve dvou datových zónách. Do vstupní datové zóny se vkládají zprávy specifikované protokolem OpenTherm, které jsou vypsány v příloze, upravené podle návodu v podkapitole 4.3.1. Do výstupní datové zóny jsou zasílány modulem odpovědi na tyto požadavky.

Modul sám udržuje komunikaci v případě, že neregistruje nevyřízené požadavky. Tato udržovací komunikace je označena jako neplatná, a proto nemá žádný vliv na funkci zařízení. Při programování ovládání připojeného zařízení je třeba brát v potaz komunikační rychlost rozhraní OpenTherm, která se pohybuje kolem 1 zprávy za vteřinu. Proto je nutné zprávy posílat s pokud možno alespoň 3-vteřinovým rozestupem, aby mělo připojené zařízení čas na odpověď.

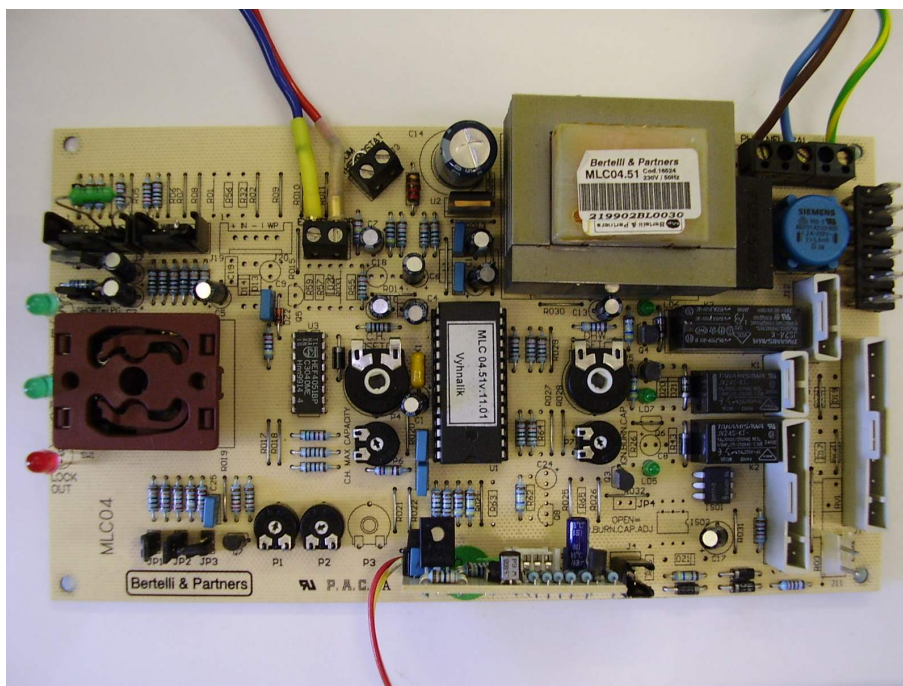
Modul může fungovat ve dvou módech, Plus a Lite. Modul se automaticky přepne do Lite módu, pokud neobdrží odpověď na 20 odeslaných zpráv. Poté se již dá předpokládat, že připojené zařízení nepodporuje protokol OpenTherm Plus. Tento mód je signalizován trvale svítící komunikační LED. Po tomto přepnutí již nelze přejít zpět do režimu OpenTherm Plus jinak než resetem modulu. Pokud obdrží na prvních dvacet požadavků alespoň jednu odpověď, je zařízení schopné komunikovat protokolem OpenTherm Plus a modul pokračuje v režimu Plus až do případného resetu.

Elektrické parametry modulu jsou následující:

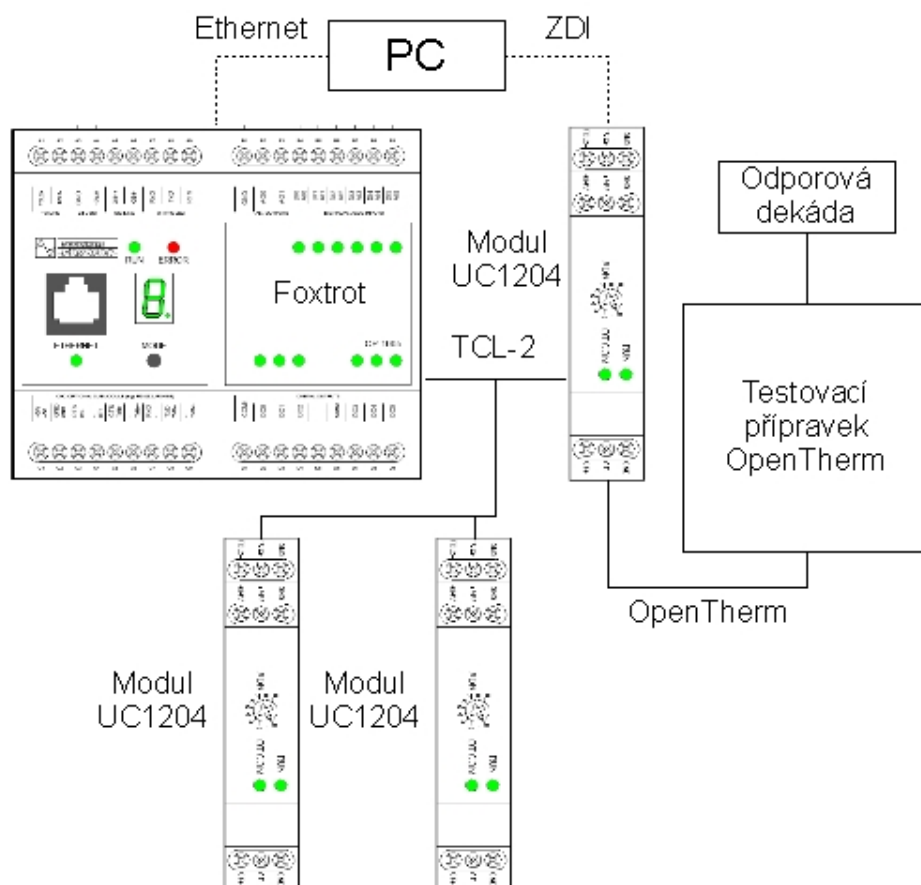
- Odběr 8 mA
- Napájení 24 V
- Příkon 192 mW

5.1 Testování modulu

K testování modulu byl firmou Teco zapůjčen přípravek simulující zařízení OpenTherm a umožňující základní komunikaci prostřednictvím protokolu OpenTherm. Přípravek umožňuje připojení čidel venkovní a vnitřní teploty, signalizuje komunikaci, zažehnutí plamene a běh čerpadla. Testování probíhalo v zapojení podle obrázku 5.2.



Obrázek 5.1: Přípravek pro testování OpenTherm komunikace

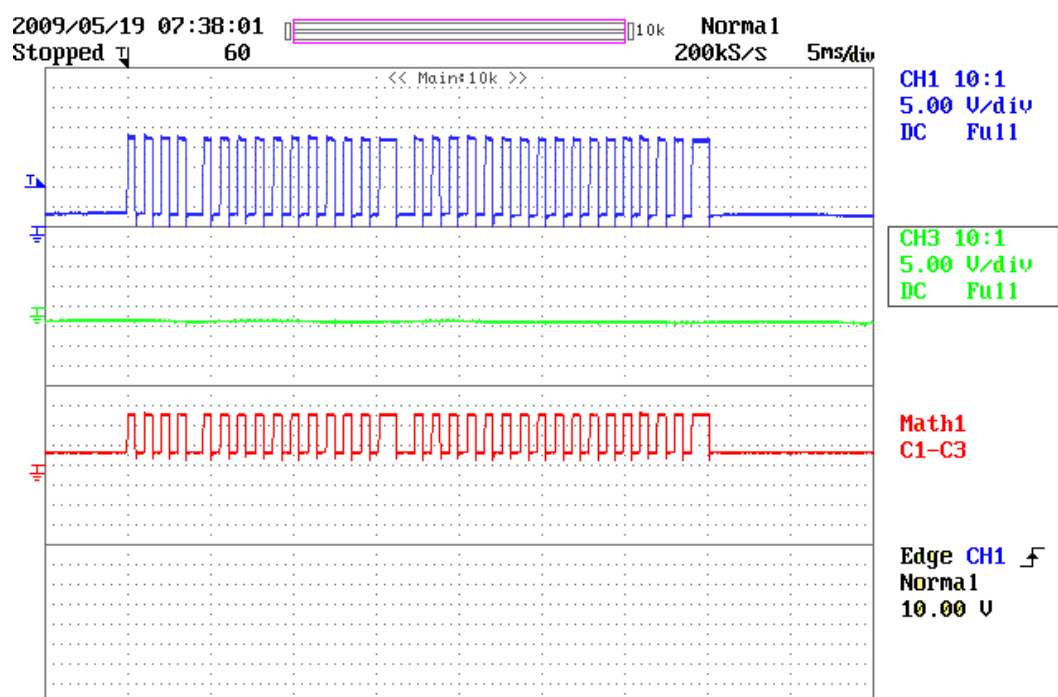


Obrázek 5.2: Přípravek pro testování OpenTherm komunikace

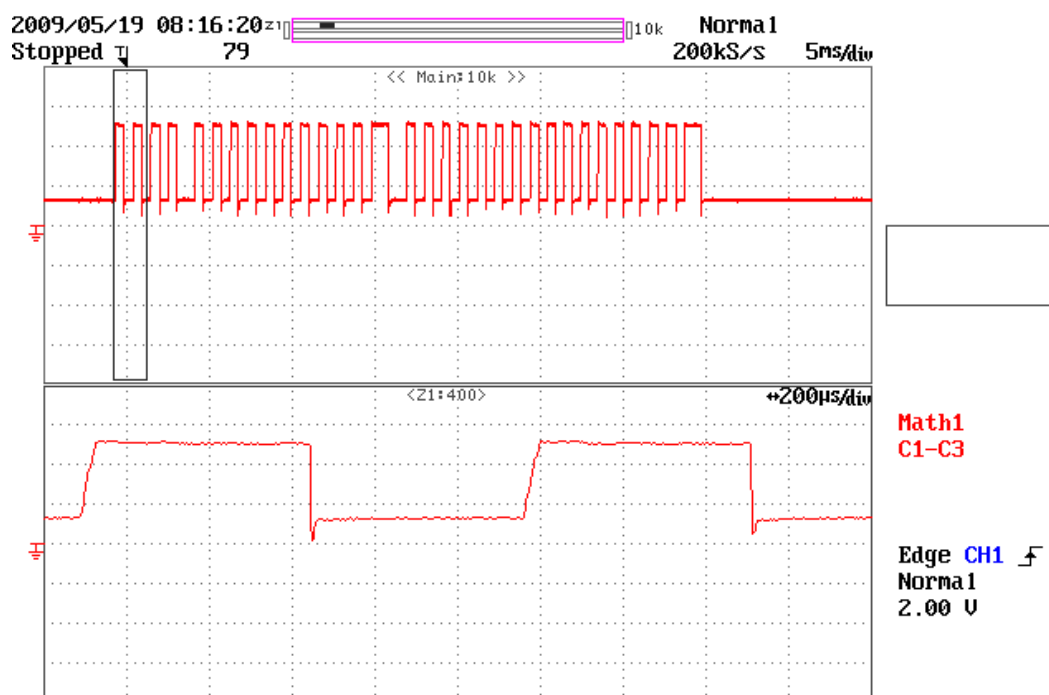
Základem je jednotka Foxtrot, která je připojena sběrnici TCL-2 k jednomu či více prototypům vyvíjeného modulu a pomocí sítě Ethernet k PC, kde běží testovací program v prostředí Mosaic. Modul je připojen pomocí rozhraní ZDI k PC, díky čemuž je možné provádět ladění programu přímo na čipu prostřednictvím vývojového prostředí ZDS. Dále je modul spojen rozhraním OpenTherm s testovacím přípravkem, který simuluje OpenTherm kompatibilní zařízení. K testovacímu přípravku je připojena odporová dekáda, která simuluje venkovní čidlo teploty.

Samotné testování ověřovalo funkčnost modulu na několika úrovních. Nejprve byla testována schopnost komunikace po rozhraní OpenTherm a sběrnici TCL-2. K tomu byl

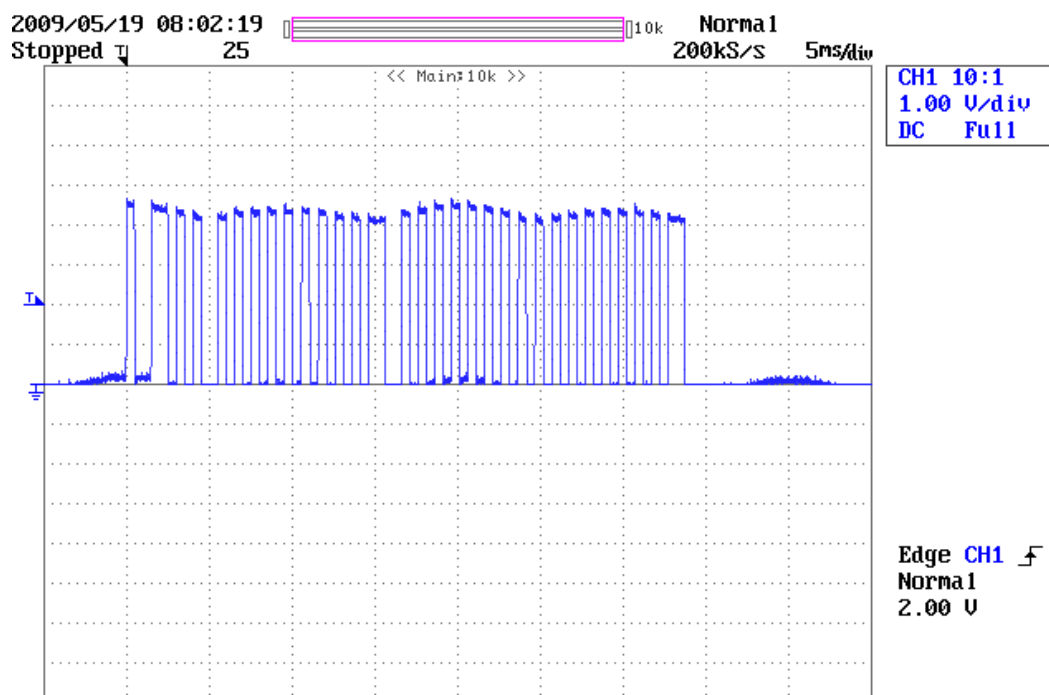
využit ještě osciloskop zapojený na svorky rozhraní OpenTherm, díky čemuž bylo možné ověřit, zda komunikace splňuje všechny požadavky specifikace protokolu. V další části byla ověřena schopnost inicializace a komunikace více modulů s centrálou Foxtrot na sběrnici TCL-2. V rámci tohoto testu byly na sběrnici zapojeny 3 prototypy vyvíjeného modulu, které se po restartu centrály úspěšně nainicializovaly a komunikovaly s centrálou. V průběhu testů rozhraní OpenTherm byly zaznamenány následující průběhy.



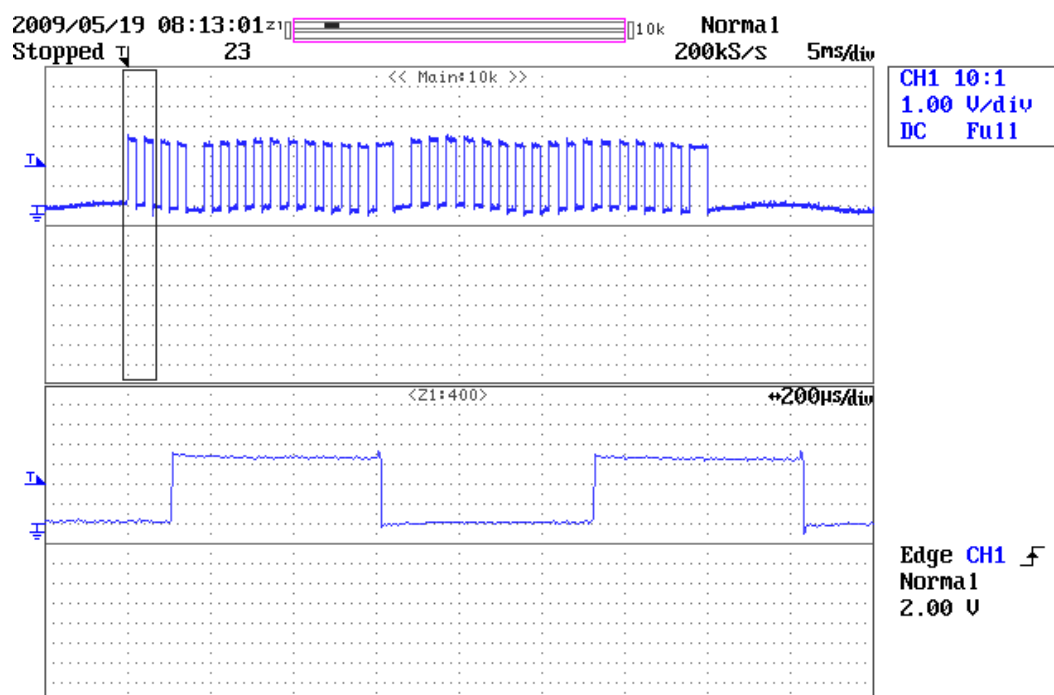
Obrázek 5.3: Vyslaná zpráva OpenTherm[Modrá - výstup regulátoru, červená - úrovně na lince]



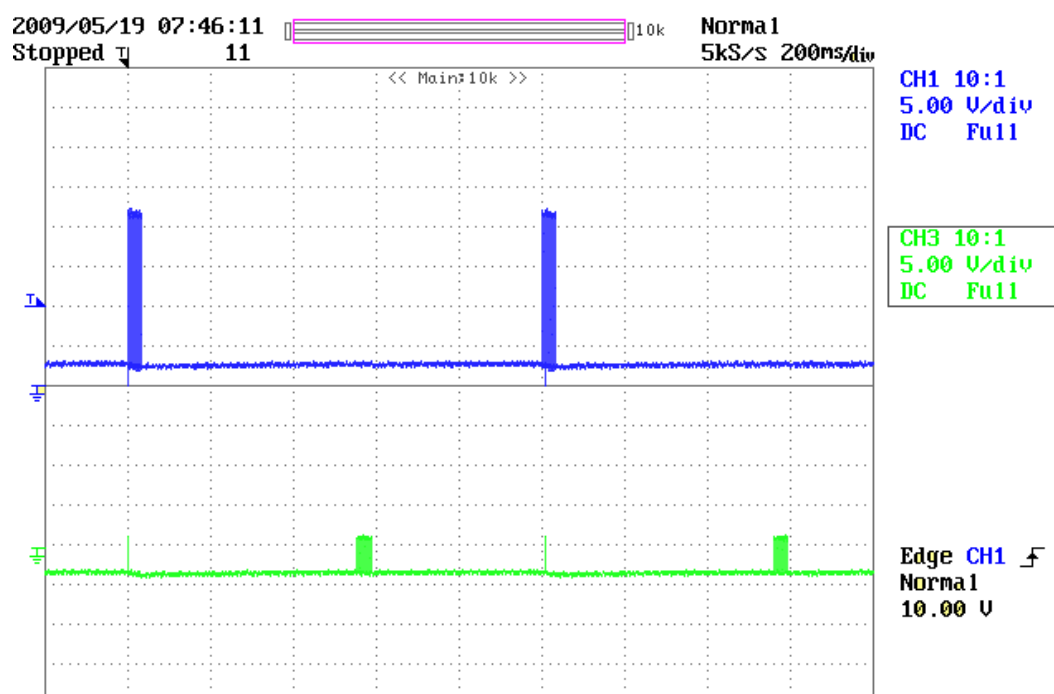
Obrázek 5.4: Vyslaná zpráva OpenTherm - detail



Obrázek 5.5: Přijatá zpráva OpenTherm



Obrázek 5.6: Přijatá zpráva OpenTherm - detail



Obrázek 5.7: Celková perioda komunikace[modrá - vysílání, zelená - příjem]

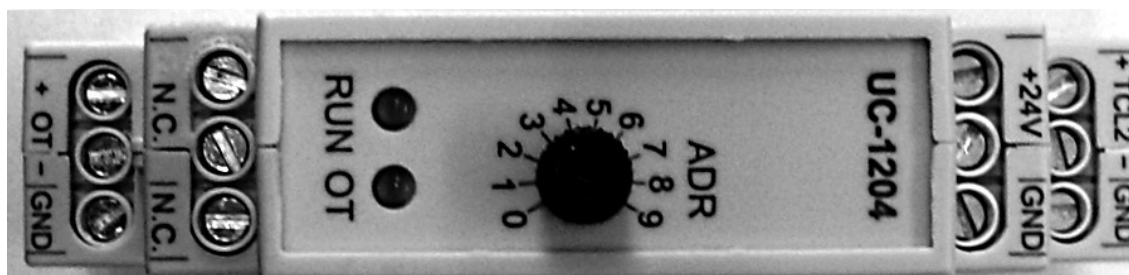
Jak je vidět z obrázku 5.4, náběžná hrana pulzu je přibližně $40 \mu s$, což splňuje specifikaci protokolu. Taktéž úrovně na lince jsou v zadaných normách. Na přijaté zprávě (obr. 5.5) je vidět, že na signál se moduluje usměrněná síťová frekvence. Ta může pocházet buď ze zdroje modulu, nebo pravděpodobněji ze zdroje testovacího přípravku simulujícího kotel.

Další ověření se týkalo obsahu komunikace po rozhraní OpenTherm. Zde byl úspěšně otestován příjem různých typů odpovědí na zprávy typu čtení, zápis a neplatná data s různými ID. Jako poslední část byla využita odporová dekáda k simulaci venkovního čidla a testování kompatibility převodu datových typů mezi zařízením OpenTherm a centrálou Foxtrot. Čidlo bylo simulováno v rozsahu kladných i záporných teplot, přičemž přijímaná data byla sledována v prostředí Mosaic. Mimoto byla otestována u všech realizovaných modulů v rámci jejich oživení schopnost přehrání různých verzí firmware z PC přes centrální jednotku Foxtrot pomocí specializovaného programu firmy Teco. Při testování bylo nalezeno několik větších či menších nedostatků, které byly posléze odstraněny. Jednalo se o několik chyb při návrhu hardware, které byly opraveny úpravou hodnot součástek či korekcí zapojení na desce plošných spojů, a dále o chyby v řídicím programu mikroprocesoru, které byly odstraněny úpravou zdrojového kódu. V další verzi modulu bude ještě doplněna přepětová ochrana v podobě transilu na 39 V a nadproudová ochrana tvořená termistorem stejného typu, jaký je použit na sběrnici TCL-2, tedy MF-USMF020.

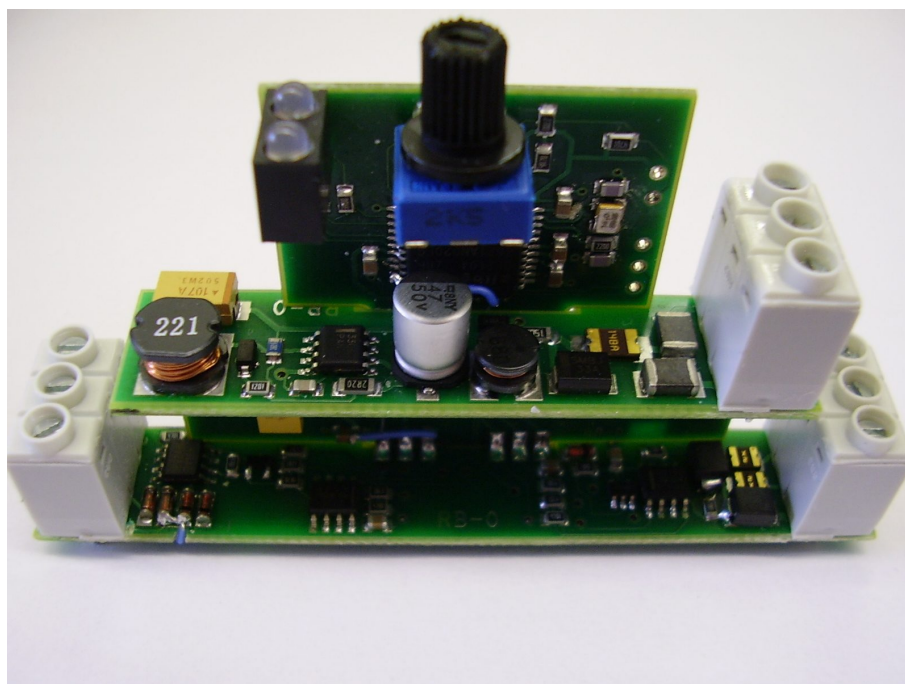
5.2 Fotodokumentace modulu



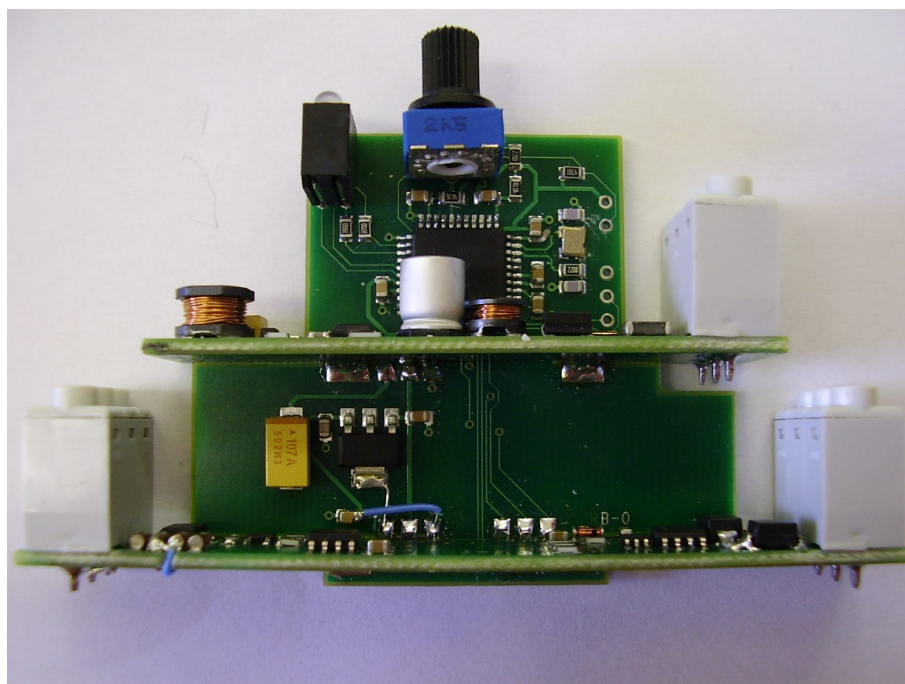
Obrázek 5.8: Modul v krabici 175/605-2



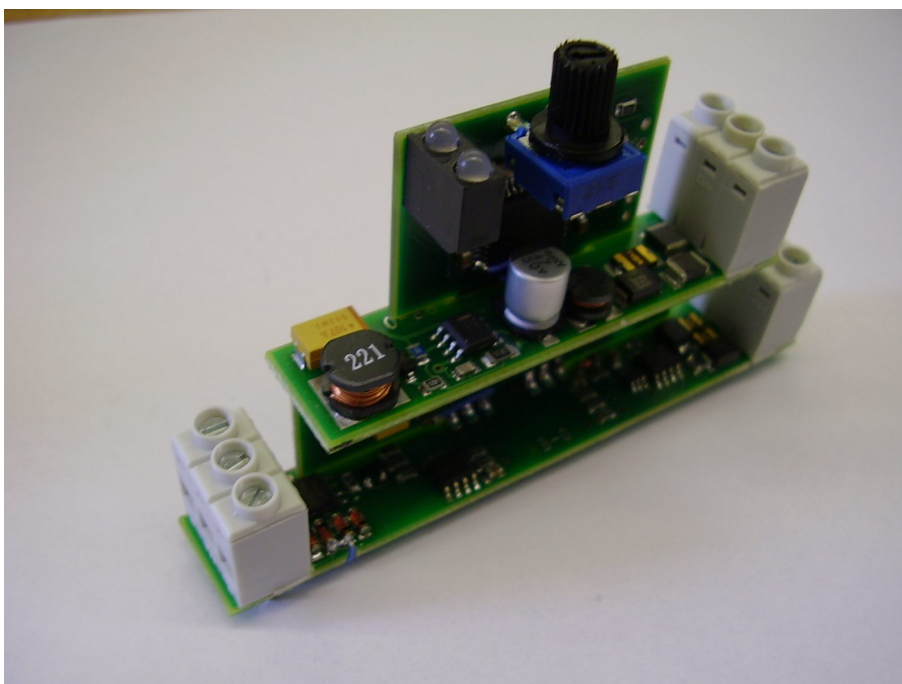
Obrázek 5.9: Modul v krabici - čelní panel



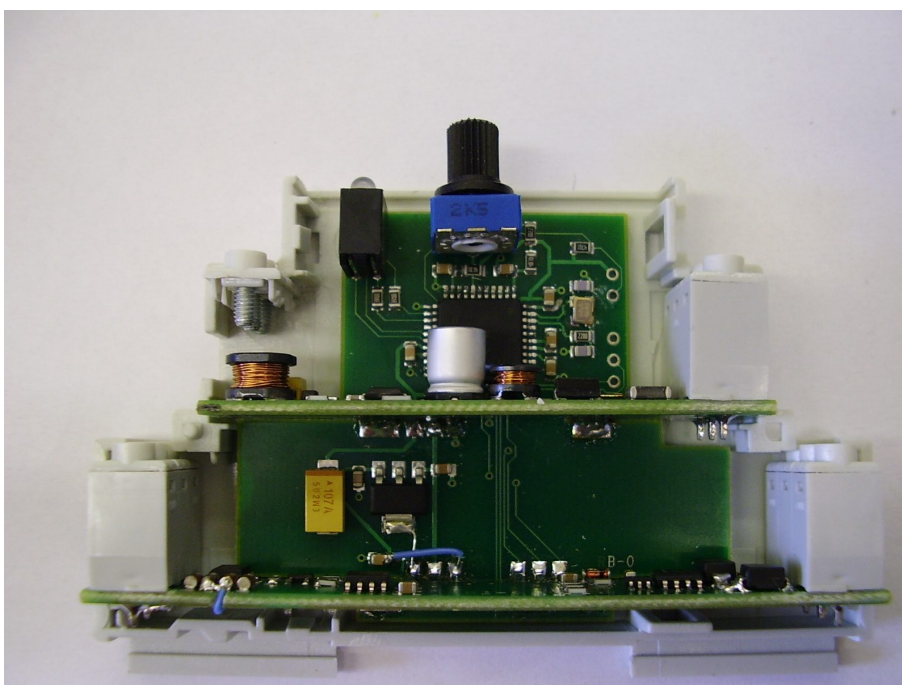
Obrázek 5.10: Prototyp modulu - rozmístění desek plošných spojů



Obrázek 5.11: Prototyp modulu - pohled z boku



Obrázek 5.12: Prototyp modulu - celkový pohled



Obrázek 5.13: Prototyp modulu - pohled z boku v krabičce

Kapitola 6

Závěr

Cílem práce byl kompletní návrh převodníku pro připojení kotlů s řízením OpenTherm k řídicím systémům Tecomat, konkrétně k centrální jednotce Tecomat Foxtrot. Práce byla vyvíjena ve spolupráci s firmou Teco a.s.. Nejprve jsem nastudoval problematiku komunikace pomocí protokolu OpenTherm, komunikační sběrnici TCL-2 a seznámil se s procesorem ZiLOG Z8 a modulárním systémem Foxtrot. Podle parametrů komunikace a požadavků obou komunikačních rozhraní byla navržena hardwarová realizace, která nejprve spočívala v sestavení schémat zapojení a po jejich ověření i v návrhu desek plošných spojů. Kompletní návrh byl proveden v prostředí OrCAD. Navržené desky plošných spojů byly poté vyrobeny a osazeny firmou Teco.

Další částí byl vývoj řídicího software pro mikroprocesor ZiLOG Z8. Celý software byl realizován v jazyce C v prostředí ZDS II. Zde byla nejprve odděleně realizována obě komunikační rozhraní, přičemž ovládací rutiny pro sběrnici TCL-2 dodala firma Teco, která si však nepřála jejich úplné zveřejnění. Proto jsou v práci uvedeny jen obecné principy komunikace na této sběrnici. Po návrhu komunikačních rutin pro protokol OpenTherm a přizpůsobení rutin TCL-2 aplikaci byl vytvořen řídicí program, který integruje obě komunikační části. Dále umožňuje řízení stavu modulu centrální jednotkou a možnost přehrávání firmware pomocí PC připojeného k PLC Foxtrot. Prototyp modulu byl poté odladěn a testován na vývojovém přípravku simulujícím OpenTherm kompatibilní zařízení a byly naměřeny jeho základní komunikační parametry, které odpovídají definici protokolu.

Modul nese označení UC-1204 a v současné době je realizován v několika prototypch, čímž splňuje zadání práce. Prototypy jsou plně funkční a podporují protokol

OpenTherm v celém rozsahu specifikace verze 2.2. V budoucnu bude provedeno ještě několik úprav, z nichž nejzásadnější je přeskupení součástek na deskách plošných spojů kvůli změně v rozmístění prvků na čelním panelu a doplnění napěťové a proudové ochrany realizované transilem a termistorem na vstup rozhraní OpenTherm. V nejbližší době bude prováděno další testování, a to jak laboratorní, tak přímo u zákazníků. Poté bude následovat nasazení modulu do sériové výroby. Rozšíří tak nabídku firmy Teco a.s. v oboru vytápění a domovní automatizace.

Literatura

- [1] Vít Záhlava (2004), *OrCAD 10*; Praha: Vydavatelství Grada Publishing a.s., ISBN 80-247-0904-X
- [2] Vít Záhlava (2005), *Návrh a konstrukce desek plošných spojů*; Praha: Nakladatelství ČVUT, ISBN 80-01-03351-1
- [3] Pavel Herout(2005), *Učebnice jazyka C 1.díl, IV.přepracované vydání*; České Budějovice: Nakladatelství KOPP, ISBN 80-7232-220-6
- [4] Pavel Herout(2004), *Učebnice jazyka C 2.díl, II.přepracované vydání*; České Budějovice: Nakladatelství KOPP, ISBN 80-7232-221-4
- [5] *OpenTherm protocol specification v2.2*
<http://www.opentherm.eu/>
- [6] *Oficiální stránky firmy Teco a.s.*
<http://www.tecomat.cz/>
- [7] *Oficiální stránky firmy ZiLOG*
<http://www.zilog.com/>

Příloha A

Zprávy aplikační vrstvy OpenTherm

A.1 Kompletní popis zpráv protokolu OpenTherm

Definice datových typů:

- u8 - 8-bitové celé číslo bez znaménka(unsigned 8-bit integer) 0...255.
- s8 - 8-bitové celé číslo se znaménkem(signed 8-bit integer), záporné číslo je dvojkový doplněk -127...128.
- f8.8 - 16-bitové číslo s pevnou řádovou čárkou(16 bit signed fixed point value), nejvyšší bit je znaménko, následuje 7 bitů celého čísla a 8 bitů čitatele zlomku /255. Nejnižší bit je tedy $\frac{1}{256}$. Záporné číslo je dvojkový doplněk.
- u16 - 16-bitové celé číslo bez znaménka(unsigned 16-bit integer) 0...65535.
- s16 - 16-bitové celé číslo se znaménkem(signed 16-bit integer) -32768...32767.

Tabulka A.1: Třída 1 - Řídící zprávy a informace o stavu

Data ID	Typ zprávy	Význam zprávy	Popis významu bitů[0/1]
0	Čtení	HB: Master status	0:Vytápění[zakázáno/povoleno] 1:TUV[zakázáno/povoleno] 2:Chlazení[zakázáno/povoleno] 3:Kompenzace venkovní teploty
Pokračování na další straně			

Pokračování z předchozí strany			
Data ID	Typ zprávy	Význam zprávy	Popis významu bitů[0/1]
			[zakázáno/povoleno] 4:Vytápění 2.okruh [zakázáno/povoleno] 5-7:rezervováno
		LB: Slave status	0:Indikace chyby[ne/ano] 1:Vytápění[neaktivní/aktivní] 2:TUV[neaktivní/aktivní] 3:Stav plamene[neaktivní/aktivní] 4:Chlazení[neaktivní/aktivní] 5:Vytápění 2.okruh [neaktivní/aktivní] 6:Žádost o diagnostiku[ne/ano] 7:rezervováno
1	Zápis	Teplota topné vody	Číslo f8.8 0-100
5	Čtení	HB:Specifické chyby aplikace	0:Požadavek na servis[ne/ano] 1:Uzamčení-reset [reset zakázán/reset povolen] 2:Nízký tlak vody[ne/ano] 3:Porucha plynu/hořáku[ne/ano] 4:Porucha tlaku vzduchu[ne/ano] 5:Přehřátí vody[ne/ano] 6-7:Rezervováno
		LB:OEM kód chyby	Číslo u8
8	Zápis	Teplota topné vody okruh 2	Číslo f8.8 0-100
115	Čtení	OEM kód diagnostiky/servisu	Číslo u16

Tabulka A.2: Třída 2 - Informace o konfiguraci

Data ID	Typ zprávy	Význam zprávy	Popis významu bitů[0/1]
2	Zápis	HB:Master konfigurace	0-8:Rezervováno
		LB:Členské ID Masteru	Číslo u8
3	Čtení	HB: Konfigurace slave	0:TUV přítomno[ne/ano] 1:Řízení[modulace/spínání] 2:Chlazení podporováno[ne/ano] 3:TUV konfigurace [nespecifikováno/nádrž] 4:Master Low-off&řízení čerpadla [povoleno/nepovoleno] 5:Vytápění 2.okruhu přítomno [ne/ano]
		LB:Členské ID slave	Číslo u8
124	Zápis	Verze protokolu Master	Číslo f8.8
125	Zápis	Verze protokolu Slave	Číslo f8.8
126	Zápis	Verze Masteru HB:Typ produktu	Číslo u8
127	Čtení	LB:Verze produktu	Číslo u8
		Verze Slave HB:Typ produktu	Číslo u8
		LB:Verze produktu	Číslo u8

Tabulka A.3: Třída 3 - Dálkové ovládání

Data ID	Typ zprávy	Význam zprávy	Popis významu bitů[0/1]
4	Zápis	HB:Kód příkazu	Číslo u8
		LB:Odpověď	Číslo u8

Podporované příkazy ve verzi 2.2 jsou:

- 1 - Reset chyby blokování
- 2 - Dopouštění topné vody

Ostatní čísla jsou v současné době nevyužita. Zařízení odpovídá pokud příkaz úspěšně proběhl číslem 0-127, v případě nesplnění příkazu číslem 128-255.

Tabulka A.4: Třída 4 - Informační a senzorová data

Data ID	Typ zprávy	Význam zprávy	Popis významu bitů[0/1]
16	Zápis	Nastavená teplota místnosti	Číslo f8.8 -40...127
17	Čtení	Aktuální hodnota relativní modulace	Číslo f8.8 0...100
18	Čtení	Tlak topné vody	Číslo f8.8 0...5(bar)
19	Čtení	Průtok TUV	Číslo f8.8 0...16(l/min)
20	Čtení Zápis	Den & čas HB:bity 7-5 Den bity 4-0 hodina LB:minuty	1...7 0...23 0...59
21	Čtení Zápis	Datum HB:Měsíc LB:Den v měsíci	1-12 1-31
Pokračování na další straně			

Pokračování z předchozí strany			
Data ID	Typ zprávy	Význam zprávy	Popis významu bitů[0/1]
22	Čtení	Rok	Číslo u16
23	Zápis	Nastavená teplota místnosti pro 2.okruh	Číslo f8.8 -40...127
24	Zápis	Změřená teplota místnosti	Číslo f8.8 -40...127
25	Čtení	Teplota topné vody z kotle	Číslo f8.8 -40...127
26	Čtení	Teplota TUV	Číslo f8.8 -40...127
27	Čtení	Venkovní teplota	Číslo f8.8 -40...127
28	Čtení	Teplota vratné vody	Číslo f8.8 -40...127
29	Čtení	Teplota solárního akumulátoru	Číslo f8.8 -40...127
30	Čtení	Teplota solárního kolektoru	Číslo s16 -40...250
31	Čtení	Teplota topné vody 2.okruhu	Číslo f8.8 -40...127
32	Čtení	Teplota TUV 2	Číslo f8.8 -40...127
33	Čtení	Teplota spalín	Číslo s16 -40...500
116	Čtení Zápis	Počet zážehů	Číslo u16
117	Čtení Zápis	Počet startů čerpadla topné vody	Číslo u16
118	Čtení Zápis	Počet startů čerpadla TUV	Číslo u16
119	Čtení Zápis	Počet zážehů TUV	Číslo u16
120	Čtení Zápis	Čas hoření plamene	Číslo u16 (h)
121	Čtení	Čas běhu	Číslo u16 (h)
Pokračování na další straně			

Pokračování z předchozí strany			
Data ID	Typ zprávy	Význam zprávy	Popis významu bitů[0/1]
	Zápis	čerpadla	
122	Čtení Zápis	Čas běhu čerpadla TUV	Číslo u16 (h)
123	Čtení Zápis	Čas hoření plamene TUV	Číslo u16 (h)

Tabulka A.5: Třída 5 - Specifické parametry kotle

Data ID	Typ zprávy	Význam zprávy	Popis významu bitů[0/1]
6	Čtení	HB:Příznaky povolení přenosu parametrů	0:Nastavení rozmezí teploty TUV[ne/ano] 1:Nastavení maximální teploty topné vody [ne/ano] 2-7:Rezervováno
		LB:Příznaky čtení/zápisu parametrů	0:Nastavení rozmezí teploty TUV [čtení/čtení-zápis] 1:Nastavení maximální teploty topné vody [čtení/čtení-zápis] 2-7:Rezervováno
48	Čtení	HB:Horní hranice nastavení TUV	Číslo u8
		LB:Dolní hranice nastavení TUV	Číslo u8
49	Čtení	HB:Horní hranice maximální nastavitelné	Číslo u8
Pokračování na další straně			

Pokračování z předchozí strany			
Data ID	Typ zprávy	Význam zprávy	Popis významu bitů[0/1]
		teploty topné vody	
		LB:Dolní hranice maximální nastavitelné teploty topné vody	Číslo u8
56	Čtení Zápis	Nastavení teploty TUV	Číslo f8.8 0...127
57	Čtení Zápis	Nastavení maximální teploty topné vody	Číslo f8.8 0...127

Tabulka A.6: Třída 6 - Transparentní parametry kotle

Data ID	Typ zprávy	Význam zprávy	Popis významu bitů[0/1]
10	Čtení	HB:Počet parametrů	Číslo u8
		LB:Rezervováno	
11	Čtení Zápis	HB:Index parametru	Číslo u8
		LB:Hodnota parametru	Číslo u8

Tabulka A.7: Třída 7 - Informace o historii chyb

Data ID	Typ zprávy	Význam zprávy	Popis významu bitů[0/1]
12	Čtení	HB:Velikost chybového zásobníku	Číslo u8
		LB:Rezervováno	
13	Čtení	HB:Index chyby	Číslo u8
		LB:ID chyby	Číslo u8

Tabulka A.8: Třída 8 - Řízení speciálních aplikací

Data ID	Typ zprávy	Význam zprávy	Popis významu bitů[0/1]
7	Zápis	Řízení chlazení	Číslo f8.8 0...100%
14	Zápis	Nastavení maximální hodnoty relativní modulace	Číslo f8.8 0...100%
15	Čtení	HB:Maximální výkon kotle	Číslo u8 0...255kW
		LB:minimální hodnota relativní modulace	Číslo u8 0...100%
9	Čtení	Dálkové přepsání nastavené teploty v místnosti	Číslo f8.8 0 = není% 1...30 = dálkové přepsání
3	Čtení	HB:Rezervováno	
		LB:Funkce při dálkovém přepisování	0:Manuální přepsání hodnoty [povolit/zakázat] 1:Programové přepsání hodnoty [povolit/zakázat] 2-7:Rezervováno

A.2 Testovací program modulu v prostředí Mosaic

Program cyklicky posílá zprávu k nastavení teploty topné vody na teplotu 0x108, tedy 26,4°C. Program poté pomocí dvou časovačů alternuje nejvyšší bit zprávy a tím odesílá každé 4 s požadavek na nastavení této teploty. Dále registruje chybová hlášení v přicházejících zprávách a počítá neúspěšně vyřízené požadavky.

VAR_EXTERNAL

dataOUT: ARRAY[0..5] OF BYTE;

dataIN: ARRAY[0..5] OF BYTE;

```
END_VAR
VAR_GLOBAL
ERR          : INT;
ERR_FLAG     : BOOL;
END_VAR
PROGRAM prgMain
VAR_INPUT
END_VAR
VAR
start1       : BOOL := 1;
start2       : BOOL;
timerTON1    : TP;
output1      : BOOL;
timerTON2    : TP;
output2      : BOOL;
POM          : BOOL;
POM2         : BOOL;
END_VAR
VAR_OUTPUT
END_VAR
VAR_TEMP
END_VAR

timerTON1(IN := start1 , PT :=T#4s , Q => output1 );
IF output1 THEN
dataOUT[0] := 128;
dataOUT[1] := 1;
dataOUT[2] := 0;
dataOUT[3] := 1;
dataOUT[4] := 1;
dataOUT[5] := 8;
POM := TRUE;
END_IF;

IF NOT output1 AND POM THEN
```

```
start1 := FALSE;
start2 := TRUE;
POM := FALSE;
timerTON1(IN := start1 , PT :=T#4s , Q => output1 );
END_IF;
```

```
timerTON2(IN := start2 , PT :=T#4s , Q => output2 );
IF output2 THEN
dataOUT[0] := 0;
dataOUT[1] := 1;
dataOUT[2] := 0;
dataOUT[3] := 1;
dataOUT[4] := 1;
dataOUT[5] := 8;
POM2 := TRUE;
END_IF;
IF NOT output2 AND POM2 THEN
start1 := TRUE;
start2 := FALSE;
POM2 := FALSE;
timerTON1(IN := start1 , PT :=T#4s , Q => output1 );
timerTON2(IN := start2 , PT :=T#4s , Q => output2 );
END_IF;
```

```
IF dataIN[2] AND 1 = 1 THEN
IF ERR_FLAG THEN
ERR := ERR + 1;
ERR_FLAG := 0;
END_IF;
END_IF;
IF dataIN[2] AND 1 = 0 THEN
ERR_FLAG := 1;
END_IF;
```

```
END_PROGRAM
```

Příloha B

Obsah přiloženého CD

- Diplomová práce ve formátu .pdf
- Vybrané zdrojové soubory řídicího programu
- Testovací projekt v prostředí Mosaic
- Bloková schémata navrženého modulu v prostředí OrCAD
- Návrh desek plošných spojů v prostředí OrCAD
- Fotodokumentace modulu